XMI Toolkit
for Rhapsody®

User Guide

IBM®

Notices

       1 Rogers Street

       Cambridge, Massachusetts 02142 U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Table of contents

1. INTRODUCTION

The XMI Toolkit purpose is to convert information between Rhapsody © projects and UML® models (versions 1.3, 2.1 and 2.2) stored in the XML Metadata Interchange (XMI®) format. With the XMI format, your models can be exchanged with other tools, like repositories or XML tools.

The import and export processes read and write XMI files that conform to the Object Management Group ™ (OMG™) XMI 1.0, 1.1, and 1.2 standards for the UML 1.3 implementation and XMI 2.1 for the UML 2.1, UML 2.2 and UML 2.3 implementation.

2. LINKS

- XMI Standard http://www.omg.org
- OMG (XMI, UML) http://www.omg.org
- XML, W3C http://www.w3.org/XML
- Rhapsody © from IBM http://www.ibm.com/software/awdtools/rhapsody/

3. REQUIRED SOFTWARE

The Toolkit requires Rhapsody version 7.6 or higher.

3.1. Windows platforms
The Toolkit runs on Windows platforms, with any **Java 1.5** virtual machine (JSDK or JRE 1.5.x or higher). A Java Virtual Machine can be downloaded at http://java.sun.com/j2se/.

3.2. Linux platforms
The Toolkit runs on RedHat 4 SE, with **Java Version 1.5.x** or higher.
It requires XulRunner 1.8.x - 1.9.x or 3.6.x.

4. USER INTERFACE

The toolkit can be launched directly from the Rhapsody Tools menu (« Import XMI into Rhapsody » and « Export XMI from Rhapsody »).
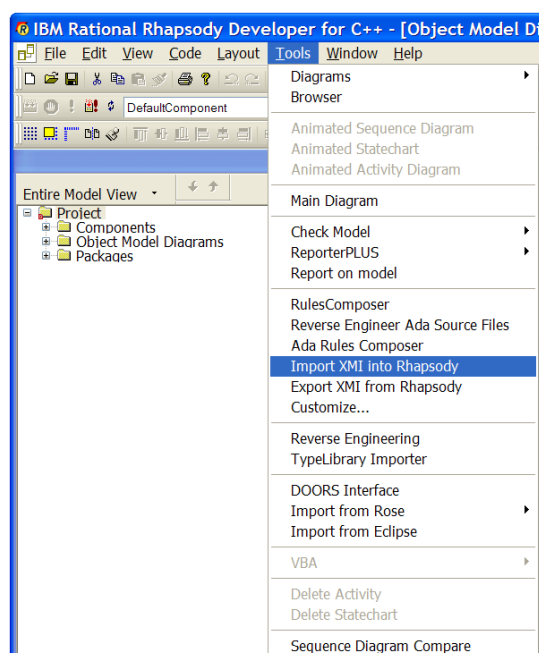
4.1. Export – User Interface

The export wizard is composed of several pages.

Each page offers the possibility to display the help document list, by clicking on the button ⑦. It will extend the current page with a help zone on the right side.

XMI Toolkit Documentation
XMI Toolkit changes
XMI Toolkit user guide
XMI Toolkit mapping rules
Tau - Rhapsody Integration Documentation
Tau importer process
Tau importer mapping rules

© Sodius SAS, 2009
version 7.5.1.1
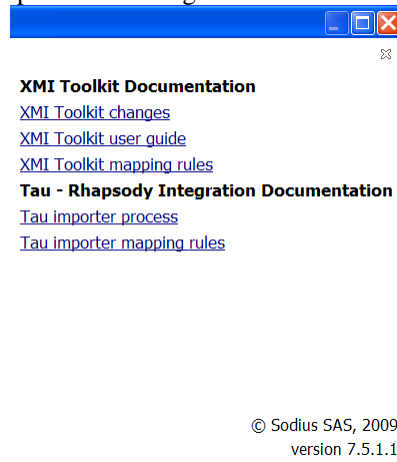
At any time you can go back to the previous page for changing the options.

Format Selection
The first one asks you to choose the export format, between UML 2.1 and UML 2.2
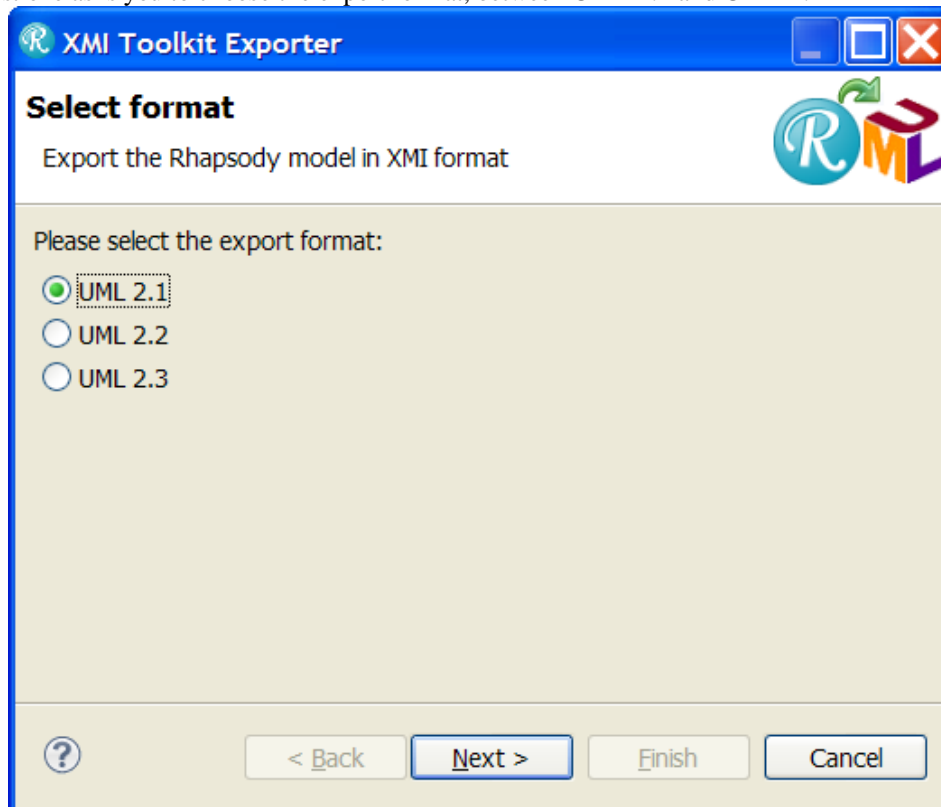
Export – Format Selection

SysML Option
In case of a SysML Model, a new field "Embed SysML Profile" appears in the window.
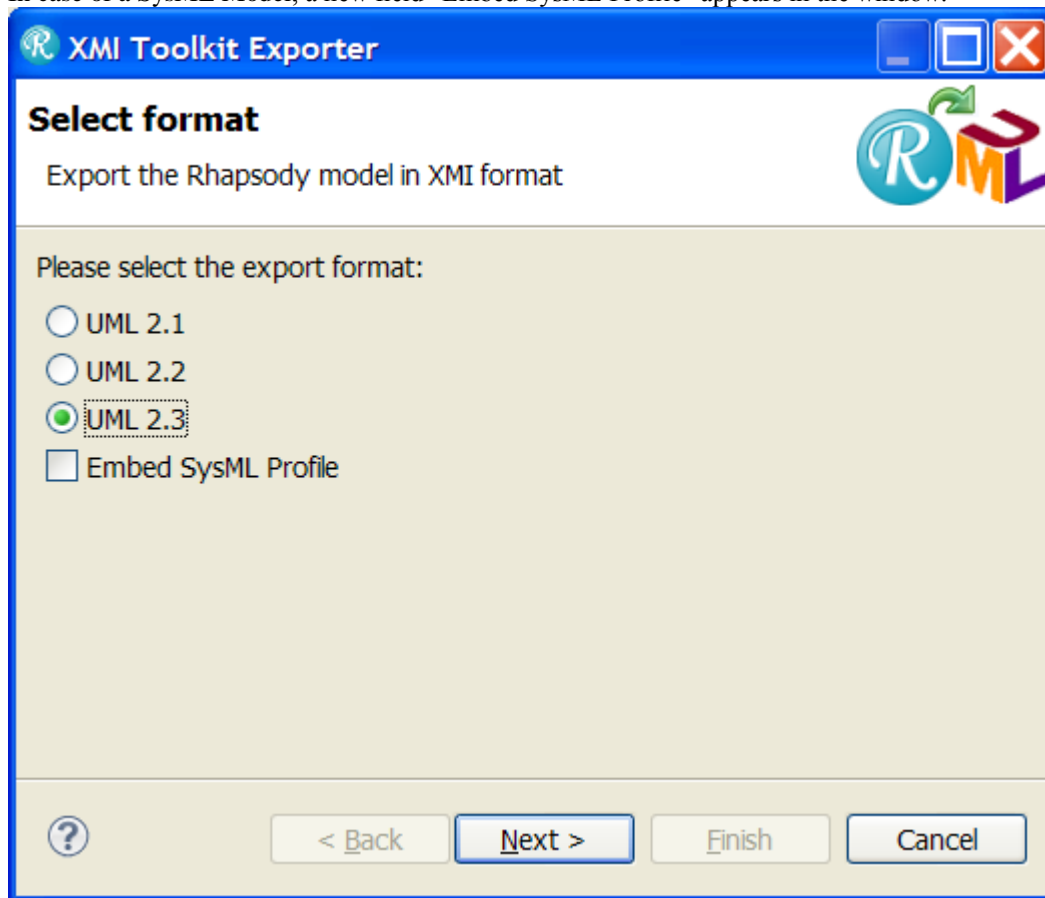
If selected:
The SysML Profile definition is embedded into the same xmi than the exported model.
This feature is useful for importing the xmi in a tool which does not manage SysML 1.2.

If not selected (only available on UML 2.3)
The SysML Profile is referenced to the Standard OMG URI
http://www.omg.org/spec/SysML/20100301/SysML-profile
This is the suggested option for importing the xmi in a SysML 1.2 compliant tool.

Units Selection
This page offers the possibility to export a fragment of your model.
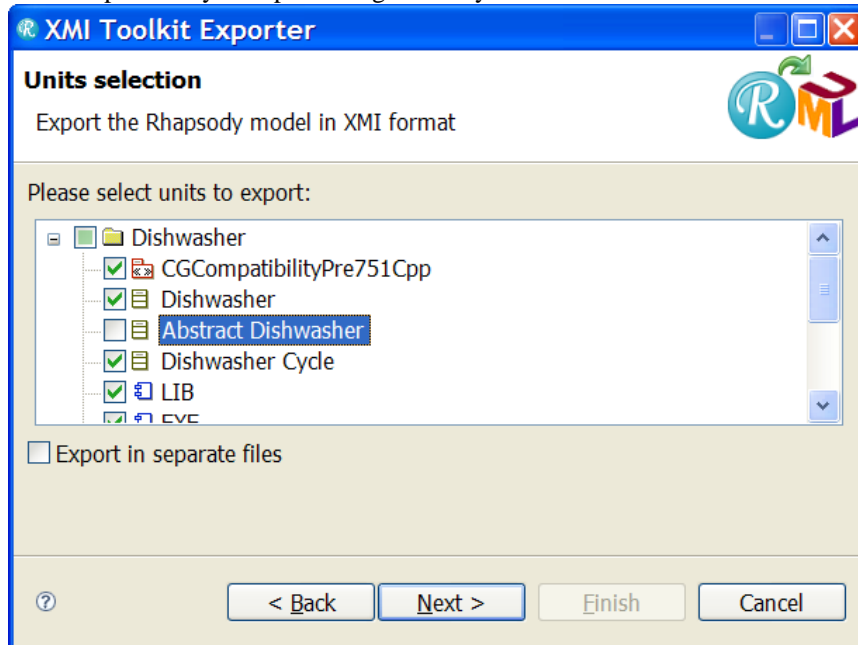
Export – Units Selection

- Unit tree
  This tree displays the list of units existing in your model, following their hierarchy.
  Select or unselect a Unit by clicking on its box
- Export in separate files
  - Checked : this option exports the model into several files following the unit fragmentation of your Model.
  - Unchecked : the selected model is exported into one file.

File/Folder selection
Select the name of the exported file.
In case of the option "Export in separate files", only a folder has to be defined.
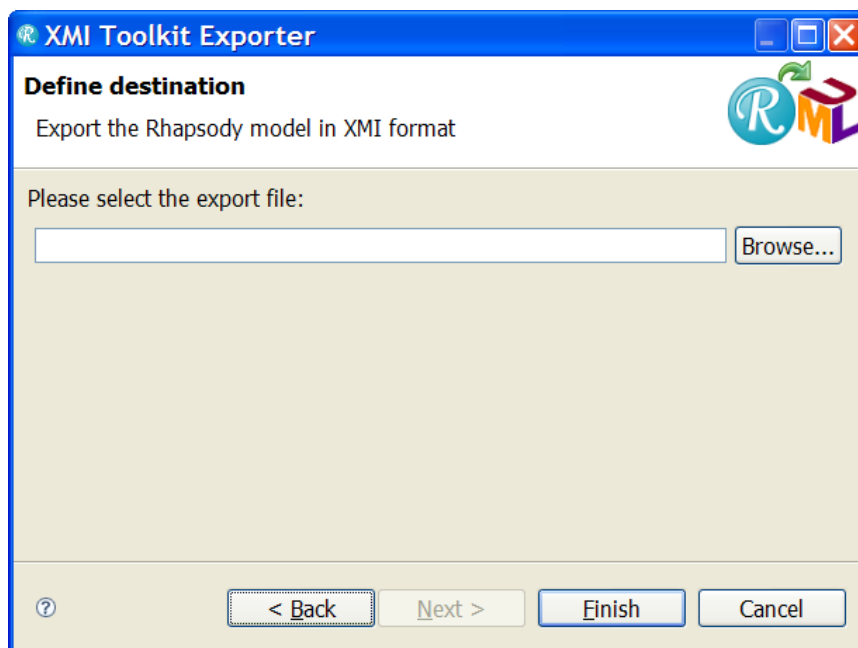
Export – File selection

4.2. Import – User Interface

File Selection
The first page asks you to select the file to import.
The XMI Toolkit automatically detects the file format.
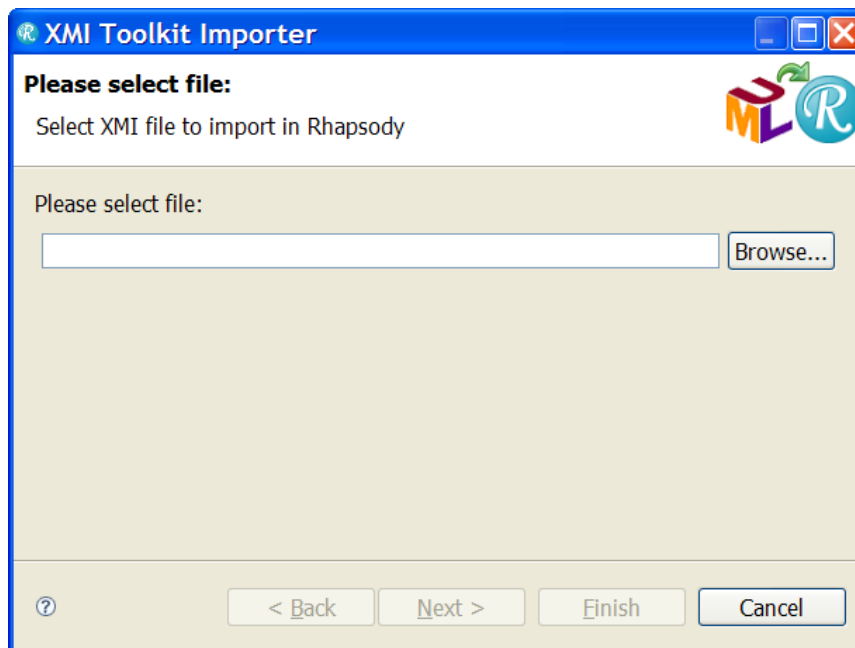
Import – File selection

Units Selection
This page offers the possibility to import a fragment of the file.
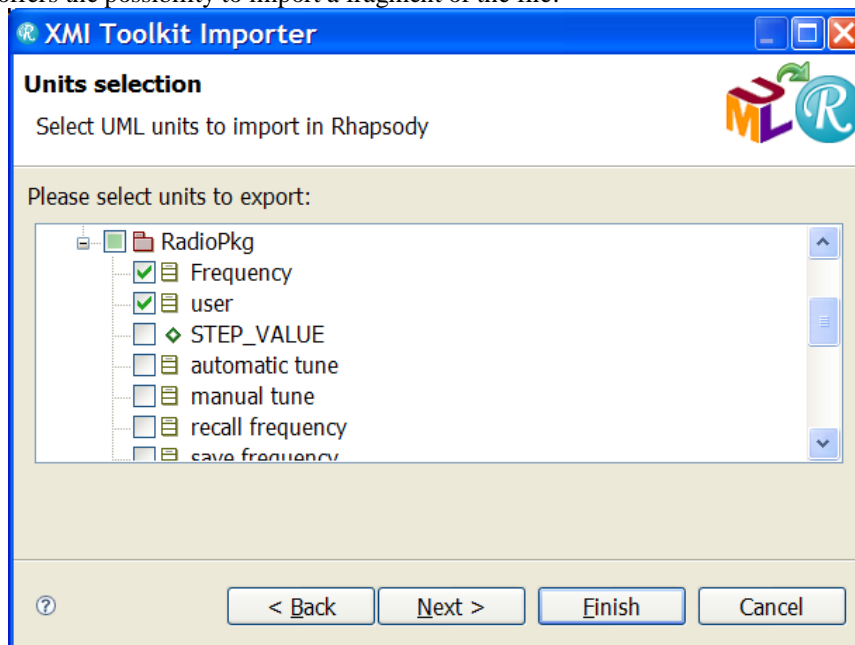
Import – Unit selection

Import method selection (Only for UML 2.x)
The XMI Toolkit offers three ways to import
- Write in current Project
  The uml file is directly written into your current project.
  Existing elements are never modified.
  Conflictual packages are resolved by a duplication name (ie : myPackage_duplicate)

- Automatic merging
  In case of conflict the imported file always contain the correct version : each conflictual element is replaced by the new one. All cross references are kept.
- Manual merging
  The Diffmerge Tool is launched and offers all the possibilities of merging.

In case of merging import, Rhapsody may ask you to save your model. This is required in case of a manual diffmerge.
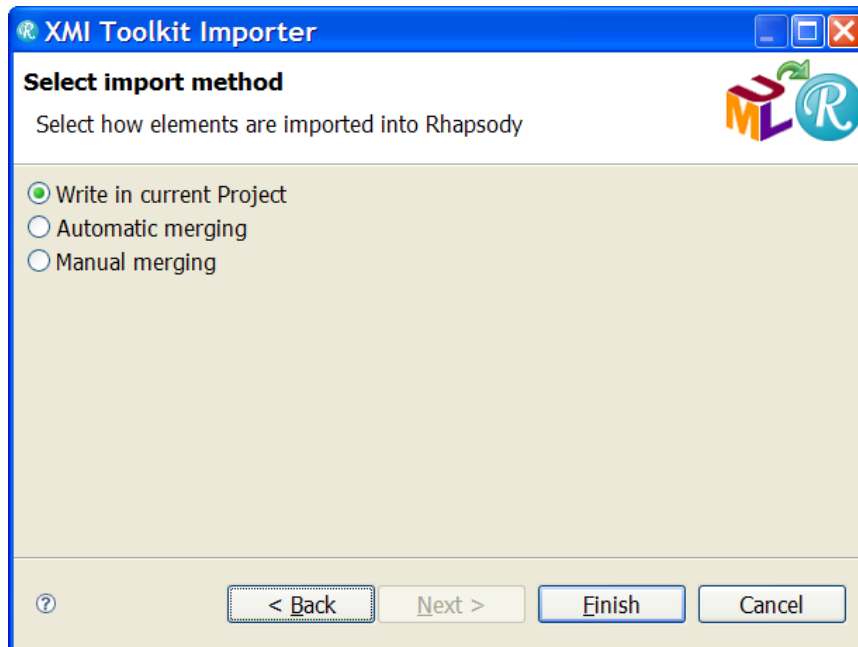
Import – Import method selection

5. RHAPSODY PROPERTIES

Rhapsody model elements can have properties, for code generation puposes for example.

If you want to generate properties in the XMI file, you will have to specify the ones to export in the *Properties.ini* file, located in the toolkit main directory. This file will contain the list of properties to export, each line specifying a property name.

The syntax is :
```
<rhapsodyMetaClass>.<propertyName> [ = <defaultValue>]
```

where :
- `<rhapsodyMetaClass>` is the name of the object type *(Actor, Class, Attribute, Package, Model, Relation*, etc.). This name must correspond to a Rhapsody COM interface name (see the Rhapsody documentation for more details).

- `<propertyName>` consists of three qualifying fields, `<subject>.<metaclass>.<name>` (see the Rhapsody documentation for more details). You can find the property names in the menu "Properties" on a model element in Rhapsody.

- The default value is optional. When you specify the default value of a property, the toolkit will only export the element properties with a value different from the default one.

Here is a small example of a *Properties.ini* :

```
# Lines starting with a '#' are ignored (= comment).
```

```
# classes properties I want to export
Class.CG.Class.UseAsExternal = False
Class.CG.Class.FileName
Class.CG.Class.Concurrency = sequential

# attributes properties I want to export
Attribute.CG.Attribute.Mutator

# actors properties I want to export
Actor.CG.Class.UseAsExternal = False
```

In this example, the properties CG.Class.Concurrency will be exported only if the value is different from *sequential*.

6. RHAPSODY TAGS

Rhapsody model elements can also have tags, which behave quite like properties except that they are not inheritable.

If you want to generate tags in the XMI file, you will have to specify the ones to export in the *Tags.ini* file, located in the toolkit main directory. This file will contain the list of tags to export, each line specifying a tag name.

The syntax is :

-    If a profile exists and if the tag is created from this profile
<profileName>.<rhapsodyMetaClass>.<tagName> [= <defaultValue>]

-    If a profile exists and the tag is not created from this profile or if there is no profile
<rhapsodyMetaClass>.<tagName> [= <defaultValue>]

where :

–   <profileName> is the name of the profile the tag is issued from

–   <rhapsodyMetaClass> is the same as for properties

–   <tagName> is the name of the tag

-    The default value is optional. When you specify the default value of a tag, the toolkit will only export the element tags with a value different from the default one.

Here is a small example of a *Tags.ini* :

```
# Lines starting with a '#' are ignored (= comment).

AdaCodeGeneration.Class.generatePragmaPure
AdaCodeGeneration.Package.generatePragmaElaborateBody
Package.tag_0
```

7. TOOLKIT AS A RHAPSODY HELPER

You can launch the Toolkit as a Rhapsody helper (launch the tool directly from Rhapsody menu). See Rhapsody help page "Helper Tool Dialog".

The XMI toolkit is directly registered as a Rhapsody helper during the installation. If this registering process failed, you can do it manually by creating two menu items:

Import XMI into Rhapsody
command: The toolkit EXE path (ex : C:\Rhapsody\Sodius\XMI_Toolkit\XMI4Rhapsody.exe)
arguments: -mode IMPORT

Export XMI from Rhapsody
command: The toolkit EXE path (ex : C:\Rhapsody\Sodius\XMI_Toolkit\XMI4Rhapsody.exe)
arguments : -mode EXPORT

8. SILENT MODE

8.1. General

The XMI Toolkit has the capability to run into silent mode, where all required parameters are passed in command line and where no user interface is displayed. You just have to execute the following command line :
```
XMI4Rhapsody -silent true <parameters>
```

The XMI Toolkit requires some extra parameters in order to run. Without all the required parameters, the toolkit simply output an error message (with all the expected parameters).

All the outputs are redirected to the "system_out.txt" and "system_err.txt" files in the toolkit "log" directory.

8.2. Parameters

The required command line parameters are :
-   mode <IMPORT|EXPORT> : determines wheter the toolkit runs into import or export mode
-   format <uml13|uml21> : specifies the metamodel to be used
-   xmi <file> : the file where to read or write data
-   writer <1.0|1.1|1.2|2.1> : specifies the xmi format to use for writing

The optional command line parameters are :
-   [automaticMerge <boolean>] : if true, indicates a automatic merging import. If false, it writes directly to the current project
-   [checkDTD <boolean>] : controls consistency check of XMI data with declared DTD
-   [copyDTD <boolean>] : controls copying of the DTD in the generated XMI file directory
-   [propertiesFile <file>] indicates the file listing the properties to export
-   [separateFile <boolean>] : indicates if the export should be done in separate files or not.
-   [tagsFile <file>] : indicates the file listing the tags defined in profiles to export
-   [units "qualifiedName1,QualifiedName2,…"] : on *import*, indicates the list of uml units to import, separate by comas. The qualified name is in the form myModel::myPackage::myUnit
-   [units "qualifiedName1#MetaClass1,QualifiedName2#MetaClass2,…"] : on *export*, indicates the list of units to export. Each qualified name should also define the metaclass of the element.

Or :
-   silentFile <file> : a Java property file which contains all the above parameters

Note
slashes are doubled in a Java property file (see java.util.Property documentation for more details).

8.3. Examples

1) All parameters as command line
```
XMI4Rhapsody -silent true -mode EXPORT -format uml13 -xmi
c:\myModel.xmi -copyDTD true
```

2) Parameters specified in a property file
```
XMI4Rhapsody -silent true -silentFile c:\silentFile.ini
```
Where the silent file contains :
```
# Silent parameters
mode = IMPORT
format = uml13
```

```
xmi = c:\\myModel.xmi
checkDTD = true
```

9. UML 2 STANDARD PROFILES IMPORT EXPORT

The XMI Toolkit offers the possibility to use Rhapsody for creating a model applying a UML2 profile as a standard profile.
This functionality is developed in three steps:
- Importing the UML2 XMI containing the Profile into Rhapsody
- Using the resulting profile
- Exporting the model

9.1. Importing a UML 2 Profile into Rhapsody

XMI Format
The XMI Toolkit detects the Standard Profile Import if the XMI is in the following format:
- XMI defined in UML 2.1
- Contains a Profile
- Does not contain a Model

Import Process
Because the XMI Toolkit detects automatically a Standard Profile Import, the import process is equivalent to any UML 2 Import.

Result in Rhapsody
The profile imported into Rhapsody contains two specific tags:
- original_UML_namespace : defines the namespace of the original Profile into the XMI
- original_UML_XMIUri : defines the XMI's URI

They will be use for founding the original Profile in the export process.
It is possible to modify them if the XMI is moved, or the Profile into the XMI is renamed.

9.2. Modeling into Rhapsody

After the importing step, it is already possible to use this new Profile as any Rhapsody Profile into a model.

This Profile can be save in a separate Unit in order to be apply it on other models, without needing to call again the UML 2 Profile import.

9.3. Exporting to UML 2

The XMI Toolkit automatically detects the presence of a standard profile into a Rhapsody Model. The export process is equivalent to any UML2 export.

If the profile URI is accessible, the export creates a UML2 XMI containing the mapped Rhapsody Project applying the original UML2 Profile.

9.4. Limitations
- In case of conflictual naming, as a name containing spaces or a name equals to a Rhapsody reserved word, the import modifies names of the profile elements.
- References from Model's elements to Profile's elements with modified names will be not exported.

10. PLUGIN MODE

The Plugin Mode is a specific mode which launches the XMI Toolkit embedded into the Rhapsody kernel. This mode can drastically reduce time needed for the XMI Toolkit processing. This mode is only accessible on a Windows plateform.

10.1. Activating Plugin Mode
The Plugin Mode is not activated by default. A batch is provided for installing the XMI Toolkit as a plugin. The process is :

1. Check that no Rhapsody is currently running
2. Into the folder */RhapsodyInstall/Sodius/XMI_Toolkit/bin*
   Double click on *plugin_rhp_ini_update.bat*

10.2. Using XMI Toolkit in Plugin Mode
After the activation of the XMI Toolkit Plugin Mode, two new options appear in the "Tools" menu of Rhapsody :
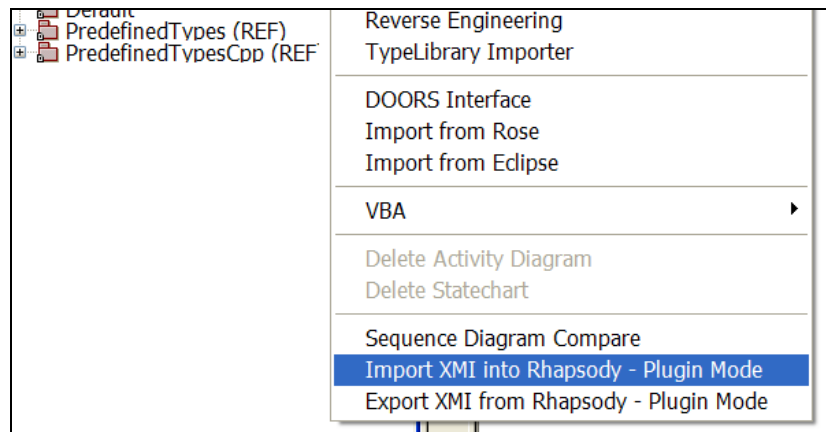
Options activated for XMI Toolkit in Plugin Mode

- Import XMI into Rhapsody – Plugin Mode
- Export XMI from Rhapsody – Plugin Mode

10.3. Disactivating Plugin Mode
A batch is provided for disactivating the XMI Toolkit as a plugin. The process is the following:

3. Check that no Rhapsody is currently running
4. Into the folder */RhapsodyInstall/Sodius/XMI_Toolkit/bin*
   Double click on *plugin_un_rhp_ini_update.bat*

10.4. RhapsodyCL Mode
RhapsodyCL is Rhapsody in Command Line.
It allows the control of Rhapsody through scripts and launch plugins with arguments.
If the XMIToolkit is installed as plugin, it is possible to call it in RhapsodyCL.
Two plugins are usable:
- Import_XMI_into_Rhapsody_PluginMode : for importing a file into Rhapsody
- Export_XMI_from_Rhapsody_PluginMode : for exporting a file from Rhapsody

Both plugins allow all the arguments defined in silent mode.
For sending arguments to the plugin, all commands need to be separate by ";"
For instance:

```
open E:\programs\rhapsody\Samples\CSamples\Elevator\elevator.rpy
call "Export_XMI_from_Rhapsody_PluginMode -silent;true;-format;uml21;-
xmi;E:\tmp\out.xmi"
exit
```

RhapsodyCL Limitations

RhapsodyCL does not handle diagrams.

Export limitations:

- Containment of Swimlanes is unknown
- Containment of CombinedFragments is unknown

Import limitations:

- Diagram layout is lost
- Sequence Diagrams are lost

11. INTEROPERABILITY BETWEEN TOOLS

Guides dedicated for each tools are published with the XMIToolkit:
Rhapsody_Enterprise_Architect_Integration
Rhapsody_RSART_Integration
Rhapsody_Tau_Integration

Please read them for complete details.

11.1. Coverage in uml1.3 import

Element	Import	Comments
Activity Diagram	7.1	
Actor	X	
Association Class	6.0	
Attribute	X	
Blocks	5.0.1 MR2	Only available with Rhapsody 5.0.1 MR2 or higher.
CallOperation	7.1	
Class	X	
Collaboration		
CombinedFragment		
Comment	5.0.1 MR2	Only available with Rhapsody 5.0.1 MR2 or higher.
Component	X	
ComponentInstance	4.0	Only available with Rhapsody 4.0 or higher.
Configuration		Not UML standard.
Constraint	X	
Dependency	X	
Diagram		Only available with Rhapsody 5.0.1 MR2 or higher.
Event	X	
EventReception	X	
File	5.2 MR1	Files in a component are imported. File or folder elements that are packages are not imported. Files in a package (only exist in Rhapsody in C and C ++) are imported. When imported into other Rhapsody languages, the files are imported as <<Rhapsody_File>> classes.
Hyperlink	X	
Implicit object	X	
Instance	5.0.1	Import : only available with Rhapsody 5.0.1 or higher.
InteractionOccurence		
Interface	X	
Generalization	X	
Operation	X	Rhapsody API does not allow to import constructors, destructors and event receptions for actors and use cases.
Link		Only available with Rhapsody 4.1 or higher.
Node	4.0	Only available with Rhapsody 4.0 or higher.
Package	X	
Port	7.1	Only avaible with UML 2.1
Project	X	
Properties	X	

Relation	X	Rhapsody API does not allow to import relations between classes and use cases.
Requirement	5.0.1 MR2	
Sequence	4.0.1	Only available with Rhapsody 4.0 or higher (4.0.1 for import). On import, a diagram view is automaticaly built.
SendAction	7.1	
Statechart	4.0.1 MR2	Rhapsody API does not allow to import flow charts. Diagram views are lost, but Rhapsody is able to rebuild a statechart view.
Stereotype	X	
Subsystem	X	Imported as <<Subsystem>> Class.
Swimlane		Only available with Rhapsody 4.1 or higher.
Template	X	
Type	X	
Typedef	X	
Transition	4.0.1 MR2	
UseCase	X	

12. MANAGING HUGE MODELS

In case of huge models, the XMI Toolkit needs more memory for its process. The solution for a majority of models is to increase the maximum memory allowed to the Java Virtual Machine.
All modification needs to be done with Rhapsody and XMI Toolkit closed, and after the creation of a Backup for each file modified.

12.1. For XMI Toolkit in Normal Mode
The file */RhapsodyInstall/Sodius/XMI_Toolkit/bin/XMI4Rhapsody.bat* contains the JVM properties.
Default sample :

```
XMIJDK%\bin\javaw.exe" -Xss4M -Xmx512m -Djava.library.path…
```

You can increase the maximum memory allowed to the JVM by changing the option "Xmx"
1024 Mb Sample:

```
XMIJDK%\bin\javaw.exe" -Xss4M -Xmx1024m -Djava.library.path…
```

12.2. For XMI Toolkit in Plugin Mode
The file /RhapsodyInstall/rhapsody.ini defines the JVM properties for plugins. Please find the [JVM] section wrote in this file
Default sample :

```
[JVM]
JavaLocation=C:\Program Files\Java\jdk1.5.0_09\
Options=ClassPath,LibPath
ClassPath=-Djava.class.path=C:\...\Share\JavaAPI\rhapsody.jar
LibPath=-Djava.library.path=C:\...\Share\JavaAPI
```

You can increase the maximum memory allowed to the JVM by adding a specific option as followed:
512 Mb Sample :

```
[JVM]
JavaLocation=C:\Program Files\Java\jdk1.5.0_09\
Options=ClassPath,LibPath,MaxMemory
MaxMemory=-Xmx512m
ClassPath=-Djava.class.path=C:\...\Share\JavaAPI\rhapsody.jar
LibPath=-Djava.library.path=C:\...\Share\JavaAPI
```

Note that a too high MaxMemory risks to prevent from the JVM execution: The XMI Toolkit in Plugin Mode would not be launched, without any error message.

13. ADDITIONAL ARGUMENTS

XMItoolkit contains additional arguments in file resources/toolkit.ini:

Import options:
- ruleset.transition.label.limit.size
  value=int
  Shrink imported transition label to this size limit. Only in case of Diagram layout import
- ruleset.transition.label.limit.char
  value=String
  Shrink imported transition label to the first existing string. Only in case of Diagram layout import

14. LIMITATIONS

14.1. Global UML 1.3 limitations
- Relations to implicit objects are not imported in UML 1.3 Importer

14.2. UML 2.1 RoundTrip Limitations

Here a list of limitation which appears in a roundtrip process: exporting a Rhapsody model to UML, reimporting this file in a new model.

- Reimport of Diagrams is limited :
  - Static diagrams are lost
  - Behavioral diagrams are redrawn to generic display
- Configurations are lost
- Statecharts and ActivityDiagrams are never inherited (always considered as overridden)
- Hyperlinks are lost
- Flows are never bidirectional

14.3. UML 2.1 Partial Import/Export Limitations

The usecase
The global use case is one central model in a specific uml tool, managed by a project leader. This model is split and shared to developers for updating/adding components. Then the developer returns the updated xmi which is reimported in the central model.

- The central model is in Rhapsody
The project leader will split the model by an export using the option "export in separate files". The export follows the unit hierarchy of the rhapsody model, and each unit is export in one xmi file. Then those files can be shared to developers.

- The central model is in another uml tool and a developer works in Rhapsody
The developer imports the shared xmi file(s) for working on his project.
When he finishes his work he will export partially his model in order to select only the "shared" part.
Here he can export without the option "export in separate files": he is free to select the name of the XMI in order to fill it with a file name required by the specific uml tool.

Limitations
A model exported in multiple files is consistent only if option "export in seperate files" is selected. Without the option, all relationships between files are lost:
When an exported element of 1.xmi references an element not exported, the XMIToolkit cannot know that the future filename of this referenced element will be 2.xmi. So when reimporting 1.xmi in Rhapsody, it references an unknown element and the linkage cannot be done.

Notes

- When dealing with partial export/import, the linkage with existing rhapsody elements can only be done when you select "automatic merging" or "manual merging" in the import method.

14.4. Troubleshooting

Linux: *The XMIToolkit crashes when opening help section – or the error reference XulRunner*
A compatible XulRunner is probably not available on your machine.
We suggest installing a compatible XulRunner, for instance:
http://ftp.mozilla.org/pub/mozilla.org/xulrunner/releases/1.8.1.3/contrib/linux-i686/xulrunner-1.8.1.3.en-US.linux-i686-20080128.tar.gz

1. Unzip the file into the XMIToolkit installation path (RHP_INSTALL/Sodius/XMI_Toolkit
2. Uncomment lines declared for this troubleshooting in file toolkit.sh (RHP_INSTALL/Sodius/XMI_Toolkit/toolkit.sh)