Instructions

for

UML Interface 2.40

Rational Rose

by

Alexander Egyed

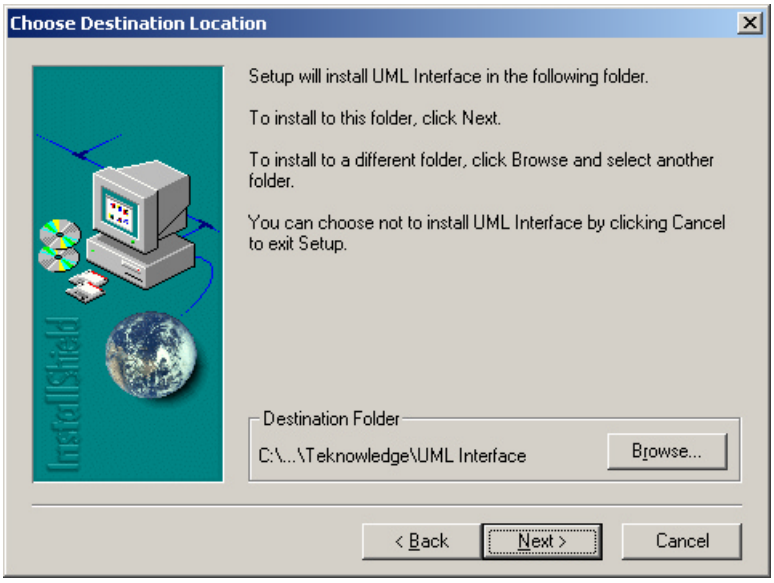
UML Interface 2.40 (and higher) integrates Rational Rose and the UML 1.3. This release has three components: (1) the interface itself, (2) a UML model browser, and (3) a change notification logger. This document describes how to install, test, and use those components.

Created: 9 September 2001
Revised: 15 October 2001
Revised: 4 December 4 2001
Revised: 5 February 2002
Revised: 15 April 2002
Revised: 1 July 2002
Revised: 25 October 2002
Revised: 11 December 2003
Revised: 1 June 2004
Revised: 1 November 2004

1. Installation

To install, please close all running applications; especially Rational Rose. The system requirements are Windows NT, Windows 2000, or Windows XP; and about 12MB of disk space.

The UML Interface is distributed in form of a zipped file (i.e., “UML_Interface.zip”). Unzip that file into a temporary directory and execute its “SETUP.EXE” program. SETUP will guide you through the installations process. Please read and acknowledge the license agreement, select the target directory (referred to as <TARGETDIR> later), and select the program group (UML Interface by default). After installation, please reboot your computer.



WARNING: If you have a previous version of the UML Interface installed then uninstall it prior to installing this version.

Please do NOT distribute the UML Interface to third parties without our approval.

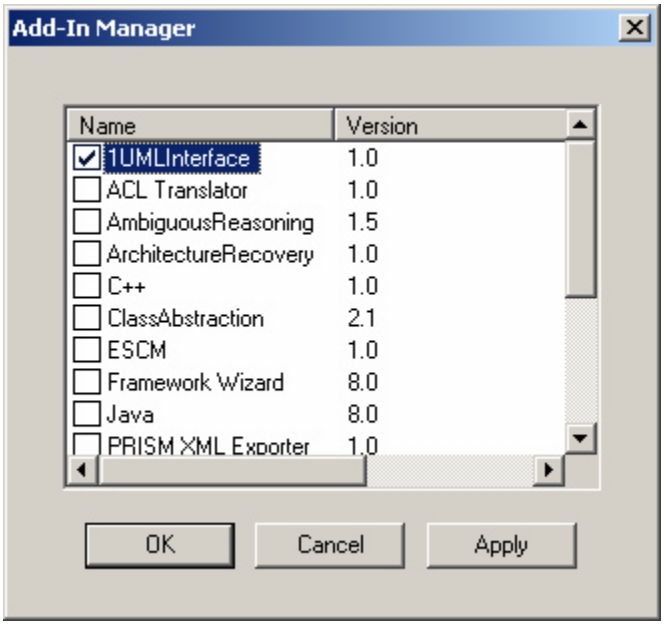
Contact aegyed@teknowledge.com for more information.

2. Basic Tests

This section presents some basic tests through which you can verify the correct installation and operation of the UML Interface and its components.

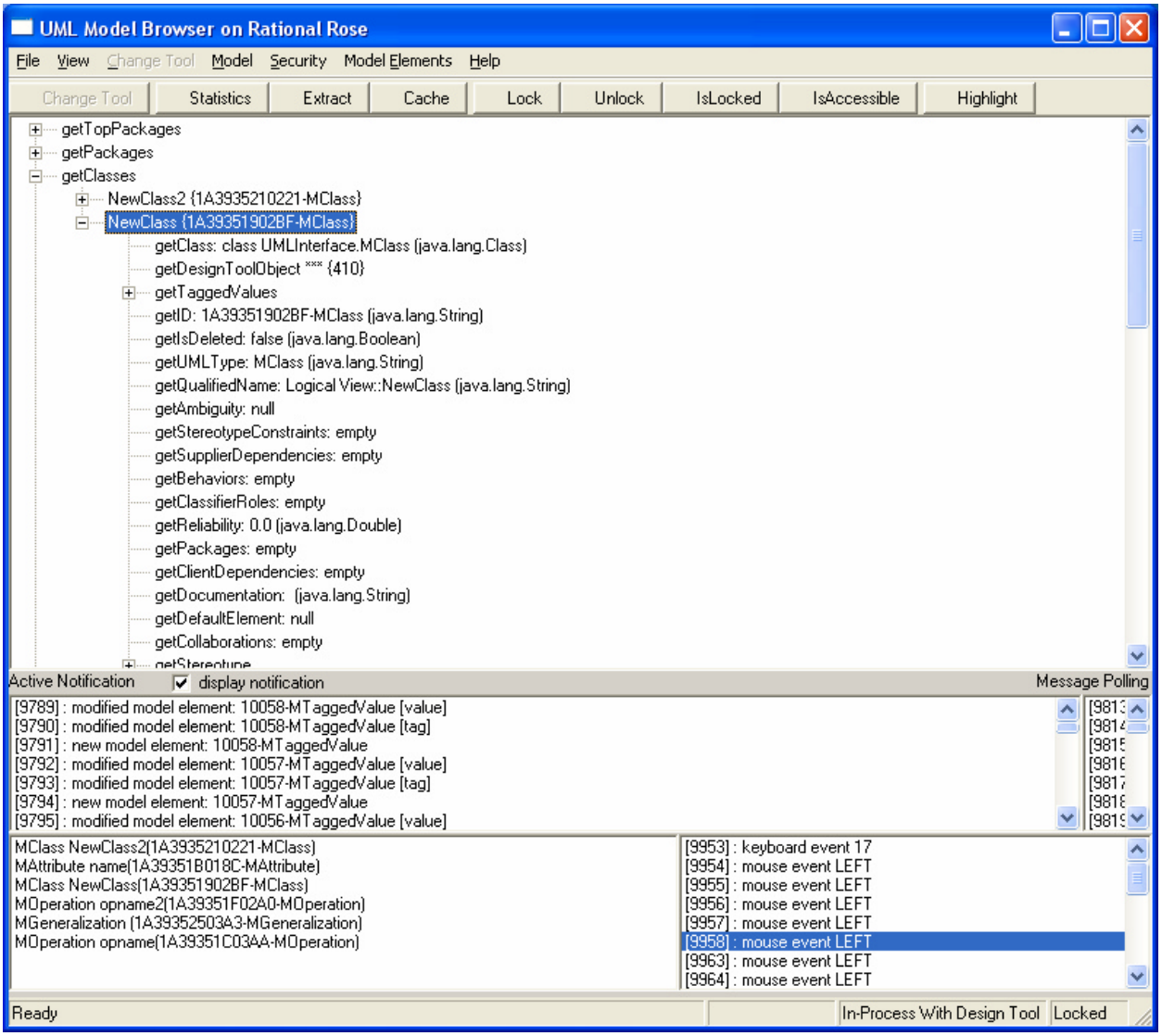
2.1 COM Test in Rational Rose

Start up Rational Rose and make sure the "1UMLInterface" component is checked. To do this, open the Add-In Manager in Rational Rose by selecting *Add-Ins::Add-In Manager* from the menu.



As a first test, open the "UML Model Browser in Java" from the *Tools* menu (select *Tools::UML Model Browser in Java* from the Rose menu). A window like the one below should appear (without information in the panels). The UML Model Browser is a sample application that allows you to navigate among the Application Program Interface we created for Rational Rose (see also *The API to the UML Interface* manual). You can start this application from within Rose ("In-Process" will appear in upper part of the window) or you can start this application as standalone from the Program menu ("Out-of-Process").

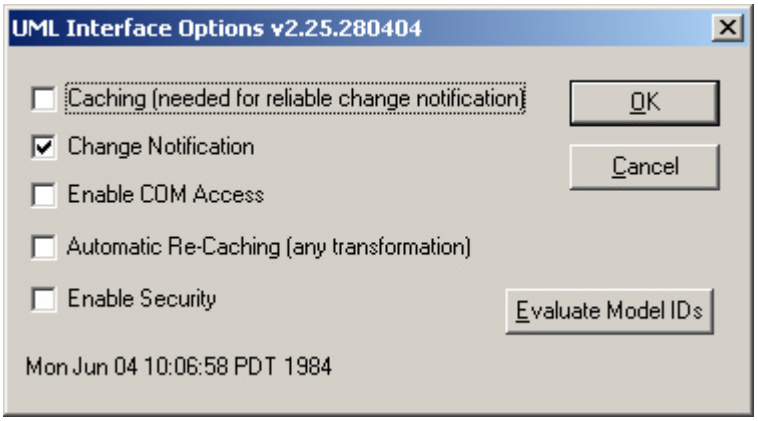
Since we have not yet loaded a Rose model, the contents of the main panel of the UML Model Browser window will be fairly blank. Nonetheless, a Rose model always contains the basic packages *Logical View* and *Use Case View*; both of which should be listed as sub-nodes in *getPackages* in the main window. By double-clicking on one of those packages (i.e., *Logical View*), more information is retrieved from Rational Rose about this package. Note that you can continue double clicking on nodes in the UML Model Browser window to navigate among other Rose model elements.



The UML Model Browser also receives changes as they occur in Rational Rose. Very simple notifications are mouse clicks. Since we navigated among some model elements in the main window, new entries in one of the panels of the UML Model Browser window should have occurred every time a mouse click or keyboard strike happened. For instance, the entry “mouse event LEFT_DOUBLECLICK” must have occurred when you double-clicked on the *Logical View* item in the model browser. To receive other events, try loading new or existing models in Rational Rose (select *File::New* or *File::Open*). Note that with the loading of new models, the UML Model Browser also refreshes the main panel of its window because the old contents got obsolete.

2.2 Change Notification Test

Through the interface, Rational Rose will notify other tools about changes that occur within it. Our change notification mechanism works in real time. Make sure it is selected by pressing the F12 key in Rose or by selected *Tools::UML Interface Options*.



- Caching: if checked then all design data in Rose is downloaded to the UML Interface (note: the UML Interface maintains a copy of the data in Rose for automatic change detection which is based on the comparison of current Rose data with previously cached Rose data)
- Change Notification: if checked then third-party tools (your tools) are notified of changes to design data in Rose. Consul the documentation for information on how to build and register observers for change detection.
- Enable COM Access: If you access the UML Interface API through a language other than Java then please check. This ensures that design data is translated into a COM-compatible format.
- Automatic Re-Caching: The UML Interface is always aware of changes inside Rose. This implies that its cached copy is always consistent with the Rose design. It is possible that bugs exist. If you feel that the UML Interface does not export the latest design then check this box to ensure that a full re-initialization is done before ACL translation and XML export.
- Enable Security: If checked then the Windows locking and unlocking mechanism is enabled. This mechanism is needed only if multiple add-ins access Rose concurrently.

Check the change notification items and press OK. Changes will now be reported in the UML Model Browser as they occur in Rose. For example, create a class, open a model, or delete a relationship and observe the notifications in the UML Model Browser Tool. Consult the Rose manual on how to do these things.

Please be aware that caching and change notification requires additional memory and processor resources.

2.5 Change Notification Logger

The change notification logger writes changes onto a log file as they occur. It can be invoked by pressing F9 or by selecting *Tools::Change Notification Logger*. It asks for a log name and then

displays a stop logging button on the top left corner. Changes will be written into the log file until the stop logging button is pressed.

The following shows an example log file. The first parameter is the time stamp, the second parameter is the tool name and the third parameter is the notification message. An optional forth parameter lists message details. For instance, the “deleted model element” message is given the unique ID of that element.

975892612364 Rational Rose: keyboard event: 13
975892612364 Rational Rose: mouse event: LEFT
975892612364 Rational Rose: mouse event: RIGHT
975892612364 Rational Rose: mouse event: LEFT
975892612364 Rational Rose: new model element: 3A2AF084016C-MClass (null)
975892612364 Rational Rose: modified model element: 36BBA7E60039-MModel [ownedElements] (Logical View)
975892612364 Rational Rose: mouse event: LEFT
975892612364 Rational Rose: mouse event: LEFT
975892612364 Rational Rose: mouse event: LEFT
975892612364 Rational Rose: mouse event: LEFT
975892612364 Rational Rose: keyboard event: 16
975892612364 Rational Rose: keyboard event: 67
975892612364 Rational Rose: keyboard event: 76
975892612364 Rational Rose: keyboard event: 65
975892612364 Rational Rose: keyboard event: 83
975892612364 Rational Rose: keyboard event: 83
975892612364 Rational Rose: keyboard event: 49
975892612364 Rational Rose: modified model element: 3A2AF084016C-MClass [name] (Logical View::Class1)
975892612364 Rational Rose: keyboard event: 13
975892612364 Rational Rose: mouse event: RIGHT
975892612364 Rational Rose: mouse event: LEFT
975892612364 Rational Rose: new model element: 3A2AF084016D-MStateMachine (null)
975892612364 Rational Rose: new model element: 3A2AF084016D-MCompositeState (null)
975892612364 Rational Rose: modified model element: 3A2AF084016C-MClass [behaviors] (Logical View::Class1)
975892612364 Rational Rose: modified model element: 1-MBase [stateMachines] (MBase)
975892612364 Rational Rose: mouse event: LEFT
975892612364 Rational Rose: mouse event: LEFT
975892612364 Rational Rose: mouse event: RIGHT
975892612364 Rational Rose: mouse event: LEFT
975892612364 Rational Rose: new model element: 3A2AF084016F-MSimpleState (null)
975892612364 Rational Rose: modified model element: 3A2AF084016D-MCompositeState [subvertices] (State/Activity Model)
975892612364 Rational Rose: mouse event: LEFT
975892612364 Rational Rose: keyboard event: 16
975892612364 Rational Rose: keyboard event: 83
975892612364 Rational Rose: keyboard event: 84
975892612364 Rational Rose: keyboard event: 65
975892612364 Rational Rose: keyboard event: 84
975892612364 Rational Rose: keyboard event: 69
975892612364 Rational Rose: keyboard event: 49
975892612364 Rational Rose: modified model element: 3A2AF084016F-MSimpleState [name] (State1)
975892612364 Rational Rose: mouse event: RIGHT
975892612364 Rational Rose: mouse event: LEFT
975892612364 Rational Rose: new model element: 3A2AF0840170-MSimpleState (null)
975892612364 Rational Rose: modified model element: 3A2AF084016D-MCompositeState [subvertices] (State/Activity Model)
975892612364 Rational Rose: mouse event: LEFT
975892612364 Rational Rose: keyboard event: 16
975892612364 Rational Rose: keyboard event: 83
975892612364 Rational Rose: keyboard event: 84
975892612364 Rational Rose: keyboard event: 65
975892612364 Rational Rose: keyboard event: 84
975892612364 Rational Rose: keyboard event: 69
975892612364 Rational Rose: keyboard event: 50
975892612364 Rational Rose: modified model element: 3A2AF0840170-MSimpleState [name] (State2)
975892612364 Rational Rose: mouse event: RIGHT

975892612364 Rational Rose: mouse event: LEFT
975892612364 Rational Rose: new model element: 3A2AF0840171-MPseudostate (null)
975892612364 Rational Rose: modified model element: 3A2AF084016D-MCompositeState [subvertices] (State/Activity Model)
975892612364 Rational Rose: mouse event: LEFT
975892612364 Rational Rose: mouse event: LEFT
975892612364 Rational Rose: mouse event: LEFT
975892612364 Rational Rose: mouse event: LEFT_DOUBLECLICK
975892612364 Rational Rose: mouse event: LEFT

3. Using the Java/COM API of the UML Interface

Please consult the manual *The API Of The UML Interface* for information on how to use the programmatic interface.

4. License Agreement

The copyright of the UML Interface belongs to Teknowledge Corporation. Please contact Alexander Egyed (aegyed@teknowledge.com) for questions.

Teknowledge Corporation,
4640 Admiralty Way, Suite 1010,
Marina Del Rey, CA 90292, USA
+1 310 578 5350

5. Release Notes

The UML Interface is still under construction. The current release implements UML interfaces for large portions of UML Statechart, UML Class Diagrams, UML Collaboration Diagrams (Sequence diagrams), and Use Case Diagrams.

The UML Interface can work with multiple instances of Rational Rose. In-process applications (tools) can only access the model of that process. Standalone applications (tools) can select from the range of available applications.

Note that the de-selection of the UML Interface in Rose causes an immediate shutdown of Rose itself. We recommend saving Rose files before doing this.

Please report any bugs or wishes to aegyed@teknowledge.com.

2.25.060104
- bug fixes (COM interface)

2.22.171003:
- bug fixes(statecharts)
- syntax check for tagged values
- adding GENERATED to MoBIES Translator generated tagged values
- adding ERROR to syntactically wrong tagged values
- painting classes with syntactically wrong tagged values red

2.00.030600

complete change notification access Rose class, sequence, state, use-case, and package diagrams.
write access to Rose class model elements
read and write access to Rose class diagrams

1.00.030000

complete read access Rose class, sequence, state, use-case, and package diagrams.

0.92.021025:

bug fixes (receptacle and dependencies)
changed API

0.92.020911:

PRISM XML EXPORTER interface added on top of UML Interface (MoBIES specific)
Discontinued the use of Unisys and Saxon for XML exporting
Created new XML Exporter based on PRISM XML EXPORTER
Upgraded Tagged Values and MoBIES Property Translator to ACL 1.0

0.91.020623:

Use case diagrams added (read only/change notification)

0.91.020415:

Sequence and Collaboration Diagrams (read only/ change notification buggy)

0.91.020415:

Optimized set language (speed improvement)
Upgrade to Unisys 1.3.3
Upgrade to Rational Rose 2002
Cleanup after XML Export (Unisys XML Exporter gives dummy names to some classes)
MoBIES Translator also takes input from documentation field (tag : value)

0.90.020205:

Improved notification about mouse and keyboard events
Use of progress bar (i.e., during caching) to help during time consuming activities
XML Export now has the option of exporting without filtering; a new dialog box exists
MoBIES Properties/Translator conform to ACL 1.0
MoBIES Translator can now be invoked for selected or all elements and supports reports
Manual Re-Caching now a menu item in Rose's tool menu

0.90.0212:

Complete rose access for class and statechart diagrams
Matured change notification but limited to changes in browser or diagram windows

0.90.0112:

Complete Implementation of the UML Meta-Model v1.3
Back-End COTS tools are still limited to Rational Rose
More mature change notification mechanism

0.82.0110:

Added Some Change Notification capability for Rational Rose added
Added Change Notification Logger
Instrumented Unisys XML Export 1.3.2
Instrumented Saxon 6.4.4
Upgraded MoBIES XML Export and Filtering

0.81.0003:

Grouped java files into packages
Upgraded ACL_Translator to support about 30 properties

0.80.0002:

Replaced non-disclosure agreement with simple copyright statement

0.80.0001:

First Release of class/statechart interface
ModelBrowser for Java and VB
MoBIES Translator (only conversion of uuid field)
Use of “Parallel State” stereotype to emulate UML isConcurrent in Composite States

6. Known Issues

The UML Interface provides full access to UML 1.3 elements but it is only partially integrated with Rational Rose. The UML Interface may be used to read use case, class, package, sequence, and statechart diagrams. Change notifications were tested primarily for class, package, use case, and statechart diagrams.

Activity and component diagrams are not supported.