Rational Software Corporation

Addendum

RATIONAL ROSE® REALTIME

VERSION: 2003.06.00

WINDOWS/UNIX/LINUX

Rational®
the software development company

support@rational.com
http://www.rational.com

Legal Notices

Part Number: 800-026121-000
Version Number: 2003.06.00

Rational, Rational Software Corporation, the Rational logo, Rational Developer Network, AnalystStudio, ClearCase, ClearCase Attache, ClearCase MultiSite, ClearDDTS, ClearGuide, ClearQuest, ClearTrack, Connexis, e-Development Accelerators, DDTS, Object Testing, Object-Oriented Recording, ObjecTime, ObjecTime Design Logo, Objectory, PerformanceStudio, PureCoverage, PureDDTS, PureLink, Purify, Quantify, Rational Apex, Rational CRC, Rational Process Workbench, Rational Rose, Rational Suite, Rational Suite ContentStudio, Rational Summit, Rational Visual Test, Rational Unified Process, RUP, RequisitePro, ScriptAssure, SiteCheck, SiteLoad, SoDA, TestFactory, TestFoundation, TestStudio, TestMate, VADS, and XDE, among others, are trademarks or registered trademarks of Rational Software Corporation in the United States and/or in other countries. All other names are used for identification purposes only, and are trademarks or registered trademarks of their respective companies.

Portions covered by U.S. Patent Nos. 5,193,180 and 5,335,344 and 5,535,329 and 5,574,898 and 5,649,200 and 5,675,802 and 5,754,760 and 5,835,701 and 6,049,666 and 6,126,329 and 6,167,534 and 6,206,584. Additional U.S. Patents and International Patents pending.

U.S. GOVERNMENT RIGHTS. All Rational software products provided to the U.S. Government are provided and licensed as commercial software, subject to the applicable license agreement. All such products provided to the U.S. Government pursuant to solicitations issued prior to December 1, 1995 are provided with “Restricted Rights” as provided for in FAR, 48 CFR 52.227-14 (JUNE 1987) or DFARS, 48 CFR 252.227-7013 (OCT 1988), as applicable.

WARRANTY DISCLAIMER. This document and its associated software may be used as stated in the underlying license agreement. Except as explicitly stated otherwise in such license agreement, and except to the extent prohibited or limited by law from jurisdiction to jurisdiction, Rational Software Corporation expressly disclaims all other warranties, express or implied, with respect to the media and software product and its documentation, including without limitation, the warranties of merchantability, non-infringement, title or fitness for a particular purpose or arising from a course of dealing, usage or trade practice, and any warranty against interference with Licensee's quiet enjoyment of the product.

Third Party Notices, Code, Licenses, and Acknowledgements

Licensee shall not incorporate any GLOBEtrotter software (FLEXlm libraries and utilities) into any product or application the primary purpose of which is software license management.

Additional legal notices are described in the legal_information.html file that is included in your Rational software installation.

Contents

Preface

This addendum provides you with important information regarding this specific release of Rational Rose RealTime. The information in this addendum supersedes the information found in the online Help and hardcopy documentation included in Rational Rose RealTime version 2003.06.00.

This addendum is organized as follows:

- *Installing Rational Rose RealTime on Linux* on page 11
- *General Issues* on page 25
- *Using Globally Unique Identifiers (GUIDs)* on page 29

Audience

This guide is intended for all readers including managers, project leaders, analysts, developers, and testers.

This guide is specifically designed for software development professionals familiar with the target environment they intend to port to.

Other Resources

- Online Help is available for Rational Rose RealTime.

  Select an option from the **Help** menu.

  All manuals are available online, either in HTML or PDF format. To access the online manuals, click **Rational Rose RealTime Documentation** from the **Start** menu.

- To acces Rational Rose RealTime support online, see: **http://www.ibm.com/support/us** .

- To send feedback about documentation for Rational products, please send e-mail to **rational-techpubs/Lexington/IBM@IBMUS.com**.

- For more information about Rational Software technical publications, see: **http://www.rational.com/support/documentation**.

- For more information on training opportunities, see the Rational University Web site: **http://www.rational.com/university**.
- For articles, discussion forums, and Web-based training courses on developing software with Rational Suite products, join the Rational Developer Network by selecting **Start > Programs > Rational Suite > Logon to the Rational Developer Network.**

Rational Rose RealTime Integrations With Other Rational Products

Integration	Description	Where it is Documented
Rational Rose RealTime–ClearCase	You can archive Rational Rose RealTime components in ClearCase.	▪ *Toolset Guide: Rational Rose RealTime* ▪ *Guide to Team Development: Rational Rose RealTime*
Rational Rose RealTime–UCM	Rose RealTime developers can create baselines of Rose RT projects in UCM and create Rose RealTime projects from baselines.	▪ *Toolset Guide: Rational Rose RealTime* ▪ *Guide to Team Development: Rational Rose RealTime*
Rose RealTime–Purify	When linking or running a Rational Rose RealTime model with Purify installed on the system, developers can invoke the Purify executable using the **Build > Run with Purify** command. While the model executes and when it completes, the integration displays a report in a Purify Tab in Rational Rose RealTime.	▪ *Rational Rose RealTime Help* ▪ *Toolset Guide: Rational Rose RealTime* ▪ *Installation Guide: Rational Rose RealTime*
Rational Rose RealTime–RequisitePro	You can associate RequisitePro requirements and documents with Rational Rose RealTime elements.	▪ *Addins, Tools, and Wizards Reference: Rational Rose RealTime* ▪ *Using RequisitePro* ▪ *Installation Guide: Rational Rose RealTime*

Integration	Description	Where it is Documented
Rational Rose RealTime–SoDa	You can create reports that extract information from a Rose RealTime model.	▪ Installation Guide: Rational Rose RealTime ▪ *Rational SoDA User's Guide* ▪ SoDA Help

Contacting Rational Customer Support

If you have questions about installing, using, or maintaining this product, contact Rational Customer Support.

Your Location	Telephone	E-mail
North, Central, and South America	+1 (800) IBM-SERV (toll free)	sw_support@us.ibm.com
Europe, Middle East, Africa	+1 (800) IBM-SERV (toll free)	sw_support_emea@nl.ibm.com
Asia Pacific	+1 (800) IBM-SERV (toll free)	sw_support_ap@au.ibm.com

Note: If you encounter any difficuly with the information contained in the table above, please contact IBM General Services at 1-800-426-4968.

When you contact Rational Customer Support, please be prepared to supply the following information:

- Your IBM customer number (ICN)
- Your name, company name, telephone number, and e-mail address
- Your operating system, version number, and any service packs or patches you have applied
- Product name and release number
- Your Service Request number (PMR#) if you are following up on a previously reported problem

When sending e-mail concerning a previously-reported problem, please include in the subject field:

[PMR XXXXXXX YYY ZZZZ]

where XXXXXXX YYY ZZZZ is the request number of the issue. For example:

[[PMR 1234567 999 0000] - New data on Rational Rose RealTime install issue

Installing Rational Rose RealTime on Linux

1

Contents

This chapter is organized as follows:

Before You Install

Before you install Rational Rose RealTime on Linux, refer to the items in Table 1 to direct you to information that can help you perform pre-installation tasks.

Table 1 Linux Pre-installation Tasks

License your Rational software	See *Specifying the Rational License Server* on page 29 of the *Installation Guide, Rational Rose RealTime,* and *UNIX Licenses* on page 87 of the *Installation Guide, Rational Rose RealTime.*
Ensure that your system meets the minimum or recommended system and software requirements	*Requirements for Linux* on page 11

Note: When using Linux with Exceed, the BACKSPACE key does not function. Instead of using the BACKSPACE key, use CTRL + h.

Requirements for Linux

The minimum supported configuration for running Rational Rose RealTime on Linux is:

- Red Hat 7.3 and 8.0.
- For Linux operation, the minimum workstation is a 450 MHz Pentium III.
- The minimum is 256 MB of RAM. We recommend 512 MB of RAM with approximately three times this amount of swap space.

- Minimum 450 MB of free disk space for the Rational Rose RealTime installation.
- Please see the Rational Rose RealTime web site (http://www.rational.com/support) for a list of the required Linux patches applicable to your operating system, or run the **check_rose_reqs** script in the $ROSERT_HOME/bin folder.

Installing in Secure Environments

Problems may occur when trying to perform a remote installation of Rational Rose RealTime for Linux in a secure environment (for example, remote access to other machines is through **ssh**) if the environment does not have access to **rsh** or **remsh**. To install Rational Rose RealTime for Linux in this situation, perform a local installation of the software rather than a remote installation. If you experience further problems, contact Rational Customer Support.

Installing Multiple Versions of Rational Rose RealTime for Linux

If you wish to install different versions of Rational Rose RealTime for Linux on the same file server, we recommend that you install them in different rational directories (referred to as <***rational_dir***>). If you install them into the same Rational directory, you will not be able to uninstall a specific version later because the uninstall script removes all versions that reside in the same Rational directory.

Stopping and Restarting an Installation

You can stop an installation by entering **q** to quit the installation. If you choose **q**, most of your input is saved to a user defaults file located in <rational_dir>/config/defaults. The file name is in the following format:

rs_install.*release_name*.*user_name*

The defaults file contains general purpose defaults that relate to the *username* and the license server that you configure. It also keeps track of the product-specific information for the installation of this specific product and version.

Note: If you enter **q!**, only some of your entries are saved to the user defaults file.

You can restart the installation by running **rs_install** again. Many of your entries appear as the default value. Press the **ENTER** key to continue with the installation.

Installation Instructions

Unless otherwise specified, your system administrator will perform these steps.

Note: For environments where there is more than one user of Rational Rose RealTime for Linux, we strongly recommend that you install the Rational Rose RealTime files on a centralized file server.

During the installation process, the default values are prefixed with the following notation:

-->

To accept the default value and continue with the installation, press **ENTER**.

To Install Rational Rose RealTime on Linux:

Note: The directory and file names indicated in the following steps are for example purposes only.

1 Log on to the install client. This may be any Linux computer that:

- Gives you access to a CD-ROM drive
- Mounts the file system into which you will load the Rational Rose RealTime for Linux release
- Runs the operating system specified on the *Rational Rose RealTime for Linux* CD (Red Hat 7.3, or 8.0)

2 Place the *Rational Rose RealTime for Linux* CD in the CD-ROM drive.

Note: If the CD-ROM drive is not mounted, mount the CD-ROM drive.

As the root user, create a directory (if one does not already exist) to be the mount point for the CD-ROM drive. The following examples for each platform use the directory **/cdrom**. Ensure that you know the device name of the CD-ROM drive. If you do not know the device name, consult your system administrator.

Mounting commands for different operating systems are as follows:

- **Linux with Volume Management**

  Linux with volume management mounts to the **/cdrom** directory automatically when you load the CD-ROM drive. You have volume management if the **vold** daemon is running on the system.

- **Linux Without Volume Management**

  ```
  # mkdir /cdrom
  # mount -t iso9660 -r /dev/cdrom /mnt/cdrom
  ```

3 From a shell window, change directory to the root level of the mounted CD-ROM device. For example: **cd /cdrom**, and press **ENTER.**

4 To run the setup script, type the following:

rs_install

The **rs_install** command is a complete installer that includes licensing setup, license checking, product installation, and product setup. Rational recommends that you follow the menus and prompts and allow **rs_install** to guide you through the installation process.

Note: You can invoke **rs_install** with a number of options. For example, you can use the **-no_log (-nl)** option to stop **rs_install** from creating a log file. To see a listing of all available options, type **rs_install -help**.

5 After the **Using RS Install** script displays, press **ENTER** to continue.

In the **Enter Install Location** script, the installation process searches for Rational directories.

6 Press **ENTER** to continue.

Next, you will specify the directory in which to install Rational Rose RealTime for Linux.

Note: An arrow (**- - >**) opposite a number indicates the default used for this installation. Press **ENTER** to select the default.

Dedpending on your installation:

- **First time installation** - If you are installing Rational Rose RealTime for Linux for the first time, you are automatically prompted to specify a directory for the installation, and then press **ENTER**.
- **Existing single installation** - If the installation process detects an existing Rational Rose RealTime directory, you can press **ENTER** to select that directory, or type **0** (zero) to specify a new directory.

- **Multiple existing installations** - If the Rational Rose RealTime installation detects multiple Rational Rose RealTime directories, type **0** to specify a new directory, or type a value associated with a listed directory, then press **ENTER**.

If you specify a new directory, **rs_install** copies the Rational files to this location. The directory name must be specified as an absolute path name. The directory must be visible on all computers from which you want to run Rational Rose RealTime, and must be writable by the installer's user name.

Next, the license agreements appear and you are prompted to accept or reject the license agreements. You must accept both license agreements to proceed with the installation.

7 If you agree with the terms of the Rational Rose RealTime License Agreement, type **Y** and press **ENTER**.

8 If you agree with the terms of the Third Party License Agreement, type **Y** and press **ENTER**.

Note: If you do not agree with the terms of the license agreements, the installation should be aborted. All software and documentation should be returned to IBM Rational Software.

9 On the **Product and License Configuration** menu, type the number associated with **Rational Rose RealTime for Linux**, then press **ENTER**.

Note: If the installation process detects any existing license configurations, you can specify their use and continue with the installation prompts. Otherwise, you must obtain and specify a valid license and continue with Step 10. For additional information on licenses, see *Specifying the Rational License Server* on page 29 of the *Installation Guide, Rational Rose RealTime*, and *UNIX Licenses* on page 87 of the *Installation Guide, Rational Rose RealTime*.

10 On the **Rational Rose RealTime - Licensing Options** menu, select a licensing option.

Option	Description
1	Use an existing Rational license (FLEXlm) file or a server that is already configured.

2	Set up a permanent or term license(s). - Request Node-Locked or floating keys through **AccountLink (https://licensing.rational.com/accountlink/transactionType)**. - After you request Node-Locked key(s) from **AccountLink**, you will receive an e-mail from Rational that contains an attachment (a .upd file). You must save this file to a secure location on your workstation.
3	Set up a temporary license file.

Depending on the licensing option you select, answer the questions and follow the directions.

11 After licensing, on the **Rational Rose RealTime - Product Customization** menu, verify that Rational Rose RealTime for Linux will be installed, and that you have enough space to install it.

12 Press **f** (the default) to continue.

13 Select an install option. **Typical** installs all components, and **Custom** allows you to specify only those components that you want installed.

14 In **Rational Rose RealTime - Enter Install Mode** menu, indicate how you want **rs_install** to deal with components that are already installed.

15 Press **ENTER** to continue.

rs_install installs Rational Rose RealTime for Linux. It there is not sufficient disk space, the installation process will stop.

16 After the installation process completes, press **ENTER** to continue.

After You Install

After you install, follow these steps:

- *Source the Setup Script* on page 17
- *Unmounting the CD-ROM Drive* on page 17
- *Set the Connexis Variable* on page 19
- *Verify the Connexis Installation* on page 19

Installing GNU 3.2

The GNU libraries included with Red Hat Linux version 7.3 are not current enough for this version of Rational Rose RealTime. To run Rational Rose RealTime on Red Hat Linux version 7.3, you must first install GCC 3.2 or later, because Rational Rose RealTime depends on newer run-time libraries included with the new versions of GCC (GNU Compiler Collection).

You can download GCC from **http://gcc.gnu.org/**.

Note: After you complete the GCC installation, ensure that the **lib** directory from the GCC installation is included in **LD_LIBRARY_PATH**. If the libraries are missing, you will receive errors similar to the following:

```
RoseRT: error while loading shared libraries: libstdc++.so.5:
cannot open shared object file: No such file or directory.
```

Source the Setup Script

After you install Rational Rose RealTime for Linux, you should **source** your **rs_setup** script to automatically set your environment variables.

- For the Rational Rose RealTime point product, type the following:

`source <`***rational_dir***`>/rosert_setup.`***csh or***
`. <`***rational_dir***`>/rosert_setup.`***sh***

Unmounting the CD-ROM Drive

For CD-ROM installations, unmount the CD-ROM drive with the following command:

```
umount /Name
```

where *Name* is the name of the device or resource.

Note: You cannot eject the CD if you are at the directory **/cdrom** or **/cdrom/cdrom0**. If you receive a "Device busy" error, change your directory location to a location other than the CD-ROM and repeat the above commands.

ClearCase Workstation Setup

The following setup must take place on all workstations that will be accessing a VOB or view. For Linux, this includes all machines that are view servers.

These steps will also need to run on all machines that act as view servers for the ClearCase views used by Rational Rose RealTime. If you use ClearCase MultiSite, you will need to do this at all the sites where the VOBs containing the Rose RealTime elements are replicated.

You can determine which machines are view servers by typing:

```
cleartool lsview
```

in a command window. The second item on each output line indicates the machine name where the view server is running. For example, if you see the following line in the output of the **lsview** command:

```
myview \\mymachine\vws\myview.vws
```

then "*mymachine*" is the name of the machine where the view server for **myview** exists.

For further details, see your ClearCase administrator.

Command-line Access to the Source Control Tool

For any user that wants to use the Rational Rose RealTime integration with ClearCase, **cleartool** must be accessible from the command prompt.

Element type setup: type manager

The following steps are required for making ClearCase clients aware of the new element type.

Linux

Use the $ROSERT_HOME/bin/$ROSERT_HOST/cc/mi_typeman script to install the type manager in each ClearCase installation. To configure the extensions and tool mappings, the user executing the script must have write access to the following directories in the ClearCase installation:

<CC_HOME>/lib/mgrs

<CC_HOME>/config/ui/icons

<CC_HOME>/config/ui/bitmaps

<CC_HOME>/config/magic

Use the following command-line to configure the proper file extensions and tool invocations:

```
<ROSERT_HOME>/bin/<ROSERT_HOST>/cc/mi_typeman.sh install-server
```

Configure the ClearCase Repository

Each VOB must be configured to allow files of the new element type to be created. Follow the steps that apply to your platform below for each VOB that will be storing Rational Rose RealTime files.

Linux

Use the $ROSERT_HOME/bin/$ROSERT_HOST/cc/mi_typeman script to register the **rosert_unit** element type in each VOB using the following syntax:

<ROSERT_HOME>/bin/<ROSERT_HOST>/cc/mi_typeman.sh install-eltype -vob <vob_path>

Test the Type Manager

To determine if the **rosert_unit** element type has been successfully registered in the VOB, perform the following command from a command prompt after changing to a directory contained in the VOB:

cleartool lstype -long eltype:rosert_unit

A listing of the type details will verify that it is correctly registered.

Set the Connexis Variable

After you install Rational Rose RealTime, you must set the environment variable for **CONNEXIS_HOME** to the appropriate location, such as:

setenv CONNEXIS_HOME $ROSERT_HOME/Connexis

Note: Set this environment variable after **$ROSERT_HOME** is created (either by `setenv` **ROSERT_HOME** or in a **rs_install** setup), and then type:

source <rational_dir>/rosert_setup.csh

or

. <rational_dir>/rosert_setup.sh

Verify the Connexis Installation

To increase efficiency and eliminate improper installation and/or setup misconfiguration, you are strongly encouraged to verify your installation. Additionally, verify your Connexis installation by using the BasicTest model provided with Connexis in `$CONNEXIS_HOME/Connexis`. This model uses the CDM transport.

Verifying your Installation using BasicTest

You can easily verify your installation by using the BasicTest model provided with Connexis in:

```
$ROSERT_HOME/Connexis/C++/examples
```

Host Configuration Installation Verification

The following instructions are for the Linux host platform.

To verify your host configuration installation:

Note: Use the information in Table 2 and Table 3, when completing the following steps:

1 Start Rational Rose RealTime.

2 Load the BasicTest model from $ROSERT_HOME/Connexis/C++/examples.

3 From the **Component View**, expand the component package corresponding to your host platform.

4 Select the client component and from its item menu, click **Build > Rebuild All** to recompile it.

5 Select the server component and from its item menu, click **Build > Rebuild All** to recompile.

6 In the **Deployment View** package, expand the processor that corresponds to your host platform.

7 The client will use port 9100 and the server will use port 9900. If these ports are being used by other another application on your workstation, you will need to change them. Open the server component instance's specification sheet and change the 9900 in the **-CNXep** startup parameter to an available port number. Open the client **Component Instance Specification** dialog box and change 9900 specified in the **-s** argument to the server's port number. Change the 9100 in the **-CNXep startup** parameter to an available port number, and then save your changes.

8 Select the server component instance and click **Run**. On the **RunTime View tab** of the instance, click **Start** to execute the server.

9 On the **Model View** tab, select the client component instance and click **Run** from its item menu. On the **Runtime View** tab of the instance, click **Start** to execute the client.

10 Verify that your output for client and server looks similar to the output shown in sections *BasicTest Server Output* on page 21 and *BasicTest Client Output* on page 22.

Table 2 Components for Referenced Configurations

	Component Package	Client Component	Server Component
Linux	REDHAT73-X86-gnu-3-2	basicTestClient_43	basicTestServer_43
Linux	REDHAT80-X86-gnu-3-2	basicTestClient_44	basicTestServer_4

Table 3 Component Instances for Referenced Configurations

	Component Package	Client Component	Client Component Instance	Server Component Instance
Linux	Red Hat 7.3	MyRedHat73Workstation	basicTestClient_43 Instance	basicTestServer_43 Instance
Linux	Red Hat 8.0	MyRedHat80Workstation	basicTestClient_44 Instance	basicTestServer_44 Instance

BasicTest Server Output

```
Rational Rose RealTime C++ Target Run Time System
Release 6.50.B.82 (+c)
Copyright (c) 1993-2003 Rational Software
rosert: observability listening at tcp port 30399


*******************************************************************
*                Please note: STDIN is turned off.                *
*   To use the command line, telnet to the above mentioned port.  *
* The _output_ of any command will be displayed in _this_ window. *
*******************************************************************


Rational Software Corp. Connexis(tm) - Distributed Connection
Service (dcs)
Release 6.50.B.82
Copyright (c) 1999-2003 Rational Software Corporation
```

```
dcs: CRM Transport : enabled
dcs: CDM Transport : enabled
dcs: CRM listening at [crm://192.139.251.167:2005]
dcs: CDM listening at [cdm://192.139.251.167:9900]
dcs: target agent enabled
dcs: locator service not available
dcs: metric service enabled

BasicTest-Server-started:

Server : Received simple greeting message... sending it back

Server : test cycle completed, received rtunbound !

Server : Received simple greeting message... sending it back
Server : test cycle completed, received rtunbound !
```

Note: The above represents a partial listing of the BasicTest Server Output.

BasicTest Client Output

```
Rational Rose RealTime C++ Target Run Time System
Release 6.50.B.82 (+c)
Copyright (c) 1993-2003 Rational Software
rosert: observability listening at tcp port 30380
*******************************************************************
*                Please note: STDIN is turned off.                *
*   To use the command line, telnet to the above mentioned port.  *
* The _output_ of any command will be displayed in _this_ window. *
*******************************************************************
Rational Software Corp. Connexis(tm) - Distributed Connection
Service (dcs)
```

```
Release 6.50.B.82
Copyright (c) 1999-2003 Rational Software Corporation


BasicTest-Client-started:

dcs: CRM Transport : enabled
dcs: CDM Transport : enabled
dcs: CRM listening at [crm://192.139.251.167:2010]
dcs: CDM listening at [cdm://192.139.251.167:9100]
dcs: target agent enabled
dcs: locator service not available
dcs: metric service enabled

Client : sending a greeting message...

->Client: received message:
RTString"Hello, Welcome to the Connexis world!"

Client : unbound received

Client : reregistering SAP

Client : sending a greeting message...

->Client: received message:
RTString"Hello, Welcome to the Connexis world!"

Client : unbound received
```

Note: The above represents a partial listing of the BasicTest Client Output.

Starting Rational Rose RealTime on Linux

To start Rational Rose RealTime on Linux configurations, run the command displayed at the end of the **rs_install** process. For example:

```
/myInstall/Rose RealTime/bin/RoseRT
```

Note: The installation process creates **rosert_setup.csh** or **rosert_setup.sh**.

You can source the setup file to help you start the programs from this installation. If you are installing Rational Rose RealTime for users in addition to yourself, you will want to have them add one of these commands to their login environment:

- Users of **csh**, **tcsh** and other **csh**-compatible shells must add the following command:

  ```
  source /myInstall/Rose RealTime/rosert_setup.csh
  ```

- Users of **sh**, **ksh**, **bash** and other bourne-compatible shells must add the following command:

  ```
  . /myInstall/Rose RealTime/rosert_setup.sh
  ```

General Issues

2

Contents

This chapter is organized as follows:

Service Pack Requirement Update for Windows

Rational Rose RealTime now supports Window service pack SP4; however, Windows service pack SP2 is no longer supported.

Using the C++ Analyzer on Linux

Although Rational Rose RealTime for Linux supports Red Hat versions 7.3 and 8.0, the C++ Analyzer is not available for these versions.

Rational Rose RealTime uses the C++ Analyzer for reverse engineering and the Code Import feature. Consequently, these features will not function in this version of Rational Rose RealTime for Linux. Alternatively, you can launch the C++ Analyzer from a UNIX or Windows version of Rational Rose or Rational Rose RealTime, and then import the code from the Linux version of Rational Rose RealTime.

Displaying the Version Tree on Linux

Rational Rose RealTime uses the ClearCase version tree feature to display a version tree from the toolset. Currently, this feature will not display the version tree because ClearCase 2002.05.00 on Linux Red Hat 8.0 does not support Unicode.

To display the version tree in Rational Rose RealTime, use a non-Unicode locale like C, from a Linux command window, and then type the following:

setenv LC_ALL C

Using Connexis Viewer on Linux

When using Connexis Viewer with RedHat Linux versions 7.3 and 8.0, the data is not legible. Alternatively, you can launch Connesis Viewer using Exceed as your Xserver, or run Connexis Viewer on Linux and use a Solaris computer as your Xserver.

You may observe errors after shutting down the Connexis Viewer and Rose RealTime.

Updating GNU Libraries on Linux 7.3

The GNU libraries included with Red Hat Linux version 7.3 are not current enough for this version of Rational Rose RealTime. To run Rational Rose RealTime on Red Hat Linux version 7.3, you must first install GCC 3.2 or later, because Rational Rose RealTime depends on newer run-time libraries included with the new versions of GCC (GNU Compiler Collection).

You can download GCC from **http://gcc.gnu.org/**.

Note: After you complete the GCC installation, ensure that the **lib** directory from the GCC installation is included in **LD_LIBRARY_PATH**. If the libraries are missing, you will receive errors .

Using Shortcut Keys to Cycle Through Open Specification Dialog Boxes on Linux

When using Red Hat Linux 7.3, the SHIFT+ALT+RIGHT (next Specification dialog box) and SHIFT+ALT+LEFT (previous **Specification** dialog box) shortcut keys do not cycle between open Specification dialogs. Instead, specifying these shortcuts cycles through Linux workspaces.

Using Web Publisher on Linux

When using Web Publisher with RedHat Linux versions 7.3 and 8.0, if the publishing process fails, change the screen display to use a lower resolution. For example, if the current screen display is set to 24 bits, change the value to 16 bits.

When using Web Publisher with Exceed, Rational Rose RealTime crashes.

You cannot view Web Publisher diagrams on RedHat Linux 7.3. You can view the published diagrams using Exceed on Windows or an Xserver on Solaris as your display.

Using Rational Rose RealTime with a Node Locked License Does Not Warn User About Expiration

Previously, Rational Rose RealTime prompted users before the expiration date for their node locked license. Now, Rational Rose RealTime does not warn users prior to license expiration. When the license expires, you will receive the following error message:

```
Unable to Obtain License.
```

Using Type Descriptor Functions

The TargetRTS Services Library uses type descriptors when it manipulates objects of a class type. The five generated type descriptor functions for a given class are **init**, **copy**, **destroy**, **encode**, and **decode**.

When the object is...	**The following functions are used...**
Output to the System Log, trace window, or watch window	encode
Modified at run-time in the watch window	init, copy, destroy, decode
Injected in a message	init, copy, destroy, decode
Sent by value (without Connexis)	copy, destroy
Sent by value (with Connexis)	init, copy, destroy, decode, encode
Sent by value (with Connexis, but in the same process)	copy, destroy

If the class is a subclass of **RTDataObject** or used in **RTSequenceOf**, or **RTWrapper** operations, the five type descriptor functions may also be used.

Note: For ObjecTime models converted into Rational Rose RealTime, only use the **RTDataObject**, **RTSequenceOf**, and **RTWrapper** classes for backwards compatibility.

Building on Solaris 2.6

You may observe the build failing with the message, "Error: make terminated by signal 10". This is due to a problem in the make utility supplied with Solaris 2.6.

Use a later version of make or install Rose RealTime in a directory with a shorter path name.

Using Globally Unique Identifiers (GUIDs) 3

Contents

This chapter is organized as follows:

Advanced Handling of Globally Unique Identifiers (GUIDs)

Rational Rose RealTime assigns Globally Unique Identifiers (GUIDs) to primary model elements (such as a class, package, component, and use case). These unique identifiers allow Model Integrator to easily identify model changes (such as when a model element is renamed or moved) and merge elements from contributor models.

In previous releases of Rational Rose RealTime, all GUIDs were computed using a time-based algorithm to ensure that the GUIDs were unique. GUIDs for additional element properties (such as operations, attributes, states, transitions, and triggers) are optional and, by default, are not set. Enabling and then disabling optional GUIDs resulted in removal of any previously existing optional GUIDs from the model.

Most model elements (such as classes, capsules, and protocols) always have GUIDs generated; however, the generation of GUIDs for a number of other elements can be enabled or disabled through the user interface. Rational Rose RealTime generates GUIDs for these model elements when you select the option **Generate for all elements** in the **Unique Identifiers** area on the **General** tab of the **Model Specification for Model** dialog box for the top model file.

Problem

Before enabling the feature to turn GUIDs on in a multi-stream or multi-model environment, it was necessary to collapse all the models and streams into a single model. If a model element with a unique ID was imported into a model with this feature not set, the optional GUIDs were removed. If a model element without a unique ID was imported into a model with this feature set, a unique ID was added. Because the unique ID was not always preserved, it was important to ensure that all models had this feature enabled to ensure that unique identifiers were preserved.

Resolution

The current GUIDs feature provides a controlled mechanism for propagating optional Globally Unique Identifiers into a multi-stream model environment. This feature supports a one-time upgrade of a model to enable GUIDs without having to create a single master model. It introduces an alternate position-based, not time-based, algorithm for GUID generation, which guarantees that the optional GUID for a property will be the same in all models where the element property has the same name and resides in the same location. For example, **NewClass1** in model A and B has a state machine. Each state machine has a state S1. The old behavior of GUID generation would have assigned a time-based GUID in each model causing merge conflicts. The new algorithm uses a name-position algorithm to assign the same GUID in each model.

If a model element property has a GUID and it is loaded into a model with optional GUIDs disabled, the GUID will be preserved. If a model element property without a GUID is loaded into a model with optional GUIDs enabled, a GUID will not be added. It is possible to have optional GUIDs enabled but GUIDs not present in the model. This new feature provides an easy method to determine if optional GUIDs are missing from a model, and it also provides you with an easy method to add additional alternate GUIDs if optional GUIDs are not present. In rare circumstances, the alternate GUID algorithm can generate a GUID which conflicts with a time-based GUID. For these situations, the new feature supports the generation of a single GUID. To activate the GUIDS feature for optional model elements, see *Managing GUIDs* on page 32.

Note: By default, all models created from Rational Rose RealTime Frameworks will have the GUIDs feature activated. We strongly recommend that you do not change the default settings for generating GUIDs in your models to ensure optimal Model Integrator merge sessions.

Generating GUIDs

Previously, before you could enable the **Generate for all model elements** option, all development streams had to be merged into a single stream. Rational Rose RealTime now includes a new method of GUID generation which attempts to generate the same GUIDs for the same elements across development streams.

Recommended Steps for Enabling GUIDs in Multi-Stream Development

To avoid conflicts between time-based GUIDs and alternate GUIDs, you can enable optional GUIDs in a single model in a single stream. If you have a multi-stream development environment where GUIDs are not enabled, we recommend that you merge all streams into a single stream before enabling optional GUIDs. (For a detailed explanation, see the topic *To set the Generate unique identifiers for all elements option* in the online Help.)

For those cases where merging into a single stream is not practical, this feature also supports the incremental introduction of GUIDs into a multi-stream environment. When optional GUIDs are added incrementally in a multi-stream environment, there is a higher risk for a GUID collision (where different model elements share the same GUID).

To minimize the change and impact of collisions:

You want to ensure that there are no duplicate GUIDs (different model elements with the same GUID) in any of the streams because duplicate GUIDs will complicate the process.

1. Open each model from the development streams and select **Add/turn on unique identifiers**.

   Note: We recommend adding optional GUIDs one model at a time and then propagating the results across a team before starting the next model.

2. Open Model Integrator and perform a trial merge of the models.

3. Load the resulting model you created from the trial merge process into Rational Rose RealTime.

   Note: If you receive the following error message

   > Warning: This model has multiple objects with the same unique id (XXXXXXXXXXXX).

   take note of the duplicate unique IDs, and then identify them in the stream in which they appear.

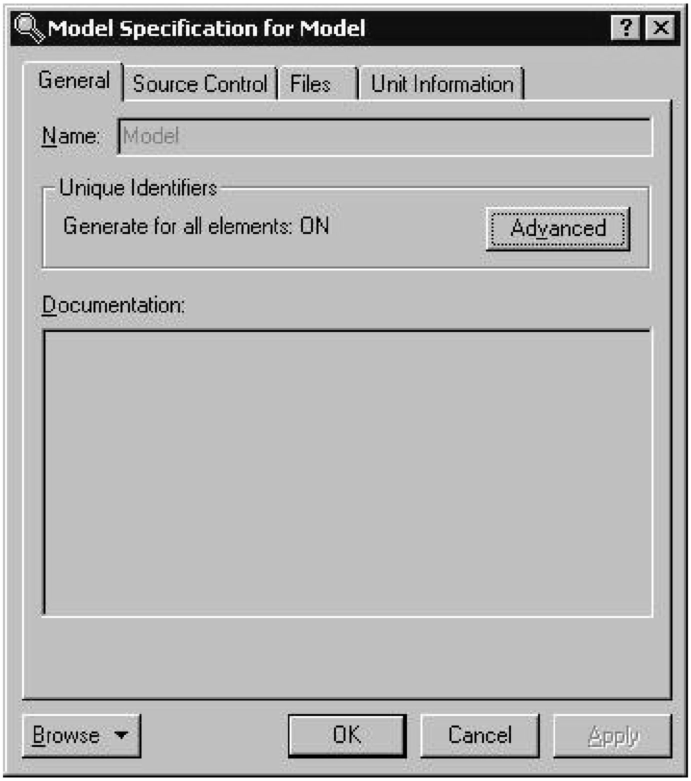
4. For each of the duplicate IDs you encounter, open the model corresponding to the ID and then select **Regenerate unique identifier** to re-generate it.

   Note: This scenario is of low probability and is not likely to occur.

5. Perform a merge again to ensure that all duplicate GUIDs have been resolved. If there are no errors, save the result as the new baseline.

Managing GUIDs

The **General** tab on the **Model Specification for Model** dialog box includes the **Generate for all elements** option that specifies the current state of a model for generating unique identifiers for the model: **ON** or **OFF**. By default, this option is set to **ON**.



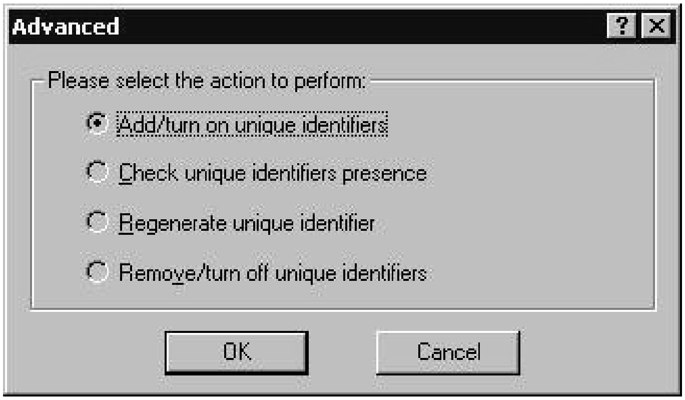
Name: Model

Unique Identifiers

Generate for all elements: ON

Documentation:

Click **Advanced** to select an action to perform with the unique identifiers.



Advanced

Please select the action to perform:

● Add/turn on unique identifiers

○ Check unique identifiers presence

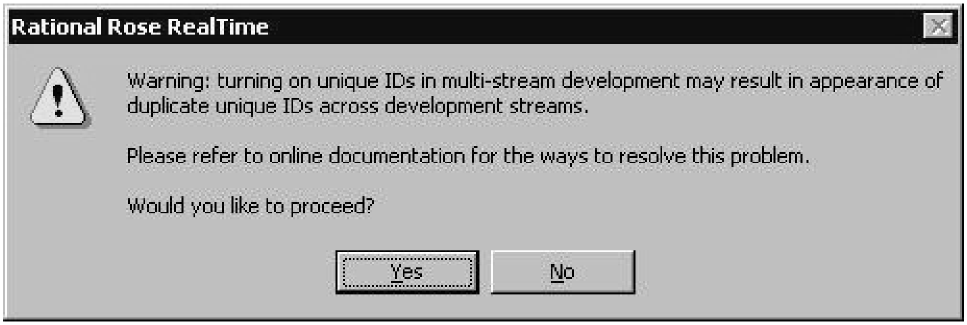
○ Regenerate unique identifier

○ Remove/turn off unique identifiers

Add/turn on unique identifiers

Activates the GUIDs feature and adds optional GUIDs to the elements that support optional GUIDs, but do not currently have a GUID.

When you select this option, the following dialog box appears:



Rational Rose RealTime

Warning: turning on unique IDs in multi-stream development may result in appearance of duplicate unique IDs across development streams.

Please refer to online documentation for the ways to resolve this problem.

Would you like to proceed?

Click **Yes** to start the process.

Note: To see details in the log, ensure that the toolset option for command logging is enabled. Click **Tools > Options**, then set the **Log commands** option on the **General** tab. The **Log** tab in the **Output** window shows the number of GUIDs generated using a message similar to the following:

Generated NNN hash GUIDs.

Where **NNN** represents the actual number of GUIDs generated in the current model by Rational Rose RealTime.

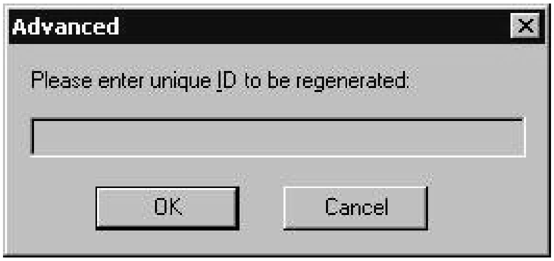
Check unique identifiers presence

Rational Rose RealTime will report the number of model elements which support optional GUIDs that do not currently have optional GUIDs assigned. If Rational Rose RealTime encounters any model elements that do not have optional GUIDs assigned, you must click the **Add/turn on unique identifiers** option.

Regenerate unique identifier

When this option is selected, you can regenerate an alternate GUID for a model element.

When you select this option, the following dialog box appears:



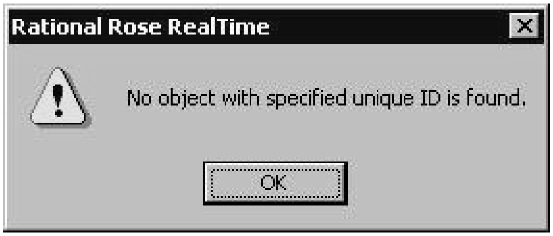
Advanced

Please enter unique ID to be regenerated:

When specifying the GUID to regenerate, enter it exactly as it appears in the model file or in the duplicate ID error message. For example:

3EC115960177

If a GUID cannot be found in the model, the model will not be modified and the following error message appears:



Rational Rose RealTime

No object with specified unique ID is found.

If a valid GUID was found in the model, a GUID is regenerated for the first model element found that matches the supplied GUID, and references to the old GUID are replaced with references to the new GUID.

If the model contains more than one element with the specified GUID, the GUID is regenerated for the model element that has an optional GUID.

Note: To see details in the log, ensure the toolset option for command logging is enabled. Click **Tools > Options**, then set the **Log commands** option on the **General** tab. After the process of regenerating GUIDs completes, check the log for messages.

Remove/turn off unique identifiers

This option deactivates the GUIDs feature and removes optional GUIDs from the model.

Note: We recommend that optional GUIDs always be enabled. Use this feature to remove GUIDs in situations where optional GUIDs are already enabled and you wish to replace them with alternate GUIDs.

Known Issues with GUIDs

- Adding or removing unique identifiers might force Rational Rose RealTime to check out controlled units from your source control tool that will not be modified. Typically, those would be units that use elements with changing GUIDs, but do not use that GUID to reference the element.
- When using alternate GUIDs, it is possible to have more than one object with the same GUID, referred to as a collision. We estimate that the number of collisions is approximately one collision per 10MB of model for each development stream. This feature automatically resolves collisions in a single model, however, you can expect GUID collisions when enabling GUIDs in different streams. See *Recommended Steps for Enabling GUIDs in Multi-Stream Development* on page 31.
- Naming of junction points on State Diagrams and connectors on Collaboration and Structure Diagrams:
  - Previously, junction points on State Diagrams, and connectors on Collaboration and Structure Diagrams were named **Junction *n***, and connectors were named **C*n*** (where *n* represents a number assigned to the element). Now, these elements are named **J<GUID>** for junction points and **C<GUID>** for connectors, where GUID is replaced with the GUID of the object. This change affects UI operations only and will not change existing models.
  - The new naming of junction points on State Diagrams and connectors on Collaboration and Structure Diagrams prevents Model Integrator merge conflicts. For example **contributor 1** adds a transition **t1** from state **S1** to state **S2**. **Contributor 2** adds a transition **t2** from state **S1** to **S3**. This should be a non-conflicting change, but previous versions of Rational Rose RealTime

named the originating junction in **S1** for **t1** and **t2** with the same name (for example, **JUNCTION1**) which results in a merge conflict requiring user intervention during the merge process. Using the new naming convention, the merge would be a non-conflicting change and Model Integrator is capable of automatically resolving the changes.

Index