

# Release Notes

RATIONAL ROSE® REALTIME

VERSION: 2002.05.00

PART NUMBER: 800-025113-000

WINDOWS/UNIX



**IMPORTANT NOTICE**

**COPYRIGHT**

Copyright ©1993-2001, Rational Software Corporation. All rights reserved.

Part Number: 800-025113-000

Version Number: 2002.05.00

**PERMITTED USAGE**

THIS DOCUMENT CONTAINS PROPRIETARY INFORMATION WHICH IS THE PROPERTY OF RATIONAL SOFTWARE CORPORATION (“RATIONAL”) AND IS FURNISHED FOR THE SOLE PURPOSE OF THE OPERATION AND THE MAINTENANCE OF PRODUCTS OF RATIONAL. NO PART OF THIS PUBLICATION IS TO BE USED FOR ANY OTHER PURPOSE, AND IS NOT TO BE REPRODUCED, COPIED, ADAPTED, DISCLOSED, DISTRIBUTED, TRANSMITTED, STORED IN A RETRIEVAL SYSTEM OR TRANSLATED INTO ANY HUMAN OR COMPUTER LANGUAGE, IN ANY FORM, BY ANY MEANS, IN WHOLE OR IN PART, WITHOUT THE PRIOR EXPRESS WRITTEN CONSENT OF RATIONAL.

**TRADEMARKS**

Rational, Rational Software Corporation, Rational the e-development company, ClearCase, ClearCase Attache, ClearCase MultiSite, ClearDDTS, ClearQuest, ClearQuest MultiSite, DDTS, Object Testing, Object-Oriented Recording, ObjecTime & Design, Objectory, PerformanceStudio, ProjectConsole, PureCoverage, PureDDTS, PureLink, Purify, Purify'd, Quantify, Rational, Rational Apex, Rational CRC, Rational Rose, Rational Suite, Rational Summit, Rational Visual Test, Requisite, RequisitePro, RUP, SiteCheck, SoDA, TestFactory, TestFoundation, TestMate, The Rational Watch, AnalystStudio, ClearGuide, ClearTrack, Connexis, e-Development Accelerators, ObjecTime, Rational Dashboard, Rational PerformanceArchitect, Rational Process Workbench, Rational Suite AnalystStudio, Rational Suite ContentStudio, Rational Suite Enterprise, Rational Suite ManagerStudio, Rational Unified Process, SiteLoad, TestStudio, VADS, among others, are either 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.

Microsoft, the Microsoft logo, Active Accessibility, Active Channel, Active Client, Active Desktop, Active Directory, ActiveMovie, Active Platform, ActiveStore, ActiveSync, ActiveX, Ask Maxwell, Authenticode, AutoSum, BackOffice, the BackOffice logo, BizTalk, Bookshelf, Chromeffects, Clearlead, ClearType, CodeView, Computing Central, DataTips, Developer Studio, Direct3D, DirectAnimation, DirectDraw, DirectInput, DirectMusic, DirectPlay, DirectShow, DirectSound, DirectX, DirectXJ, DoubleSpace, DriveSpace, FoxPro, FrontPage, Funstone, IntelliEye, the

IntelliEye logo, IntelliMirror, IntelliSense, J/Direct, JScript, LineShare, Liquid Motion, the Microsoft eMbedded Visual Tools logo, the Microsoft Internet Explorer logo, the Microsoft Office Compatible logo, Microsoft Press, the Microsoft Press logo, Microsoft QuickBasic, MS-DOS, MSDN, Natural, NetMeeting, NetShow, the Office logo, One Thumb, OpenType, Outlook, PhotoDraw, PivotChart, PivotTable, PowerPoint, QuickAssembler, QuickShelf, Realimation, RelayOne, Rushmore, SourceSafe, TipWizard, TrueImage, TutorAssist, V-Chat, VideoFlash, Virtual Basic, the Virtual Basic logo, Visual C++, Visual FoxPro, Visual InterDev, Visual J++, Visual SourceSafe, Visual Studio, the Visual Studio logo, Vizact, WebBot, WebPIP, Win32, Win32s, Win64, Windows, the Windows CE logo, the Windows logo, Windows NT, the Windows Start logo, and XENIX are trademarks or registered trademarks of Microsoft Corporation in the United States and other countries.

FLEXIm and GLOBEtrotter are trademarks or registered trademarks of GLOBEtrotter Software, Inc. Licensee shall not incorporate any GLOBEtrotter software (FLEXIm libraries and utilities) into any product or application the primary purpose of which is software license management.

Portions Copyright ©1992-20xx, Summit Software Company. All rights reserved.

**PATENT**

U.S. Patent Nos. 5,193,180 and 5,335,344 and 5,535,329 and 5,835,701. Additional patents pending.

Purify is licensed under Sun Microsystems, Inc., U.S. Patent No. 5,404,499.

**GOVERNMENT RIGHTS LEGEND**

Use, duplication, or disclosure by the U.S. Government is subject to restrictions set forth in the applicable Rational Software Corporation license agreement and as provided in DFARS 277.7202-1(a) and 277.7202-3(a) (1995), DFARS 252.227-7013(c)(1)(ii) (Oct. 1988), FAR 12.212(a) (1995), FAR 52.227-19, or FAR 227-14, as applicable.

**WARRANTY DISCLAIMER**

This document and its associated software may be used as stated in the underlying license agreement. 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 or fitness for a particular purpose or arising from a course of dealing, usage, or trade practice.

# Contents

<b>Preface</b> .....	<b>21</b>
Audience .....	21
Other Resources .....	21
Contacting Rational Technical Publications .....	21
Contacting Rational Technical Support .....	22
<b>Introduction</b> .....	<b>1</b>
Contents .....	1
Overview .....	1
What's New .....	2
PLEASE READ FIRST .....	3
You Must Install License Keys to Run this Product .....	3
Notice To Customers Running Earlier Rose RealTime Releases on Windows ..	3
New Rational Developer Network Menu Option .....	4
<b>Platform and Toolchain Requirements</b> .....	<b>5</b>
Contents .....	5
Referenced Configuration Requirements for Windows NT .....	5
Referenced Configuration Requirements for Windows 2000 .....	6
Referenced Configuration Requirements for Windows XP Pro .....	6
Referenced Configuration Requirements for UNIX .....	7
Toolchain Requirements .....	7
Help Viewer (Windows Platforms Only) .....	7
Compiler .....	8
Real-Time Operating System .....	8
Changes to Referenced Configurations .....	8

<b>Licensing Requirements</b> .....	<b>11</b>
Contents .....	11
Checking the Validity of Your License Keys .....	11
License Usage .....	11
License Location .....	12
<b>Integration Notes</b> .....	<b>13</b>
Contents .....	13
Microsoft Development Environment .....	13
Configuration for Operation with Rational ClearCase .....	13
Integration with Rational Robot .....	13
Rational ClearCase on a UNIX Server and Clients on Both NT and UNIX ..	14
Rational SoDA .....	14
Rational RequisitePro .....	14
Rational Purify .....	15
Adding Options to Purify on UNIX .....	15
Using OSE and Rose RealTime with the Code Warrior Compiler .....	15
<b>Transition Notes</b> .....	<b>17</b>
Contents .....	17
Migrating from Rational Rose and ObjecTime Developer .....	17
Using the New Installation .....	18
Installation Process .....	18
Installation on Windows .....	18
Updating Batch Files .....	18
Additional settings .....	19
Installation on UNIX .....	19
Installation Overview .....	19
Licensing .....	21
Installing Source Files .....	21
Environment Variables for UNIX .....	21
Upgrading .....	22
Upgrading to New Version Only (Deleting Earlier Versions) - UNIX .....	22
Upgrading to Rose RealTime 2002.05.00 and While Maintaining Rose RealTime 2001A.04.00 - UNIX .....	23
Reinstalling .....	24

<b>Known Problems, Limitations, and Updates</b> . . . . .	<b>25</b>
Contents . . . . .	25
Installation and Startup Issues . . . . .	25
File Association for Compiled Scripts . . . . .	25
Suite Objects.dll Not Found . . . . .	25
Start-up Problems . . . . .	26
Uninstall . . . . .	26
General Issues . . . . .	26
Unique Ids . . . . .	27
Probes . . . . .	27
ReliableService.ebx Crashes When Running -runScriptAndQuit. . . . .	28
License Manager . . . . .	28
Changes to Code that Uses Default Arguments . . . . .	28
Spaces in Directory Names . . . . .	29
Too Many Files Open . . . . .	30
Executing Component Instances . . . . .	30
Sequence Diagrams . . . . .	30
Build Dependencies on Case-Insensitive File Systems (NT only) . . . . .	30
Using the Debugger-xxgdb Tool and Running your Component Instance . . . . .	31
Running Multiple Sessions on the Same Workstation . . . . .	31
Use of C and C++ Add-ins . . . . .	31
Symbolic Links with TargetRTS . . . . .	31
McAfee VirusScan Affects Startup and Shutdown . . . . .	31
Creating Capsules with the Same Name in Different Packages . . . . .	32
Code Generator Runs Out of Memory When Generating Very Large Models. . . . .	32
Window Order Policy . . . . .	32
Running Model Examples . . . . .	33
Using Get Method and Set Method in the Attribute and Operation Tools . . . . .	33
Integrations Issues . . . . .	33
Error Loading Large Models from ClearCase - Windows. . . . .	33
RequisitePro . . . . .	34
SoDA . . . . .	34
Online Help Issues . . . . .	35
Problems Accessing Rose Help when Concurrently Running Rose RealTime on Windows NT . . . . .	35
Problem . . . . .	35
Explanation . . . . .	35
Workaround . . . . .	35

Java Reference . . . . .	36
Navigating through the QuickStart Tutorial Online Help . . . . .	36
Viewing Animated Demonstrations . . . . .	36
Windows 2000 Issues . . . . .	36
UNIX Issues . . . . .	37
Executing the Current Version with a Previous Version . . . . .	37
Refresh Problems with Exceed. . . . .	38
HP-UX Make Dependency Filename Restriction . . . . .	38
Non-GUI-based External Editors . . . . .	38
Case Sensitivity within Paths . . . . .	39
Troubleshooting Rose RealTime Toolset Freezes on UNIX . . . . .	39
Toolset freezing on startup . . . . .	39
Online Help . . . . .	40
Context-sensitive help on UNIX may not invoke the correct help page. . . . .	40
Cannot open some links with online help . . . . .	40
RQA-RT . . . . .	40
Limitations . . . . .	40
Additional documentation on create messages and sequence diagrams. . . . .	41
Additional documentation on send message specification data field format for Java .42	42
Updates . . . . .	42
Code Generator Updates . . . . .	42
Model File Syntax . . . . .	44
.jar Files . . . . .	44
rtsetup.pl File . . . . .	44
Hot Link for Call Message Operations . . . . .	45
Problems Addressed . . . . .	46
<b>Documentation Updates . . . . .</b>	<b>51</b>
<b>Technical Support . . . . .</b>	<b>53</b>
Contents . . . . .	53
Submitting Problem Reports . . . . .	53
Submitting Feature Requests. . . . .	54
Submitting Support Requests . . . . .	55
Contacting Rational Customer Service by Email or Telephone . . . . .	56
License Support Contact Information . . . . .	57
<b>Index . . . . .</b>	<b>59</b>



# Preface

This manual provides the information on the following key topics:

- *What's New* on page 2
- *PLEASE READ FIRST* on page 3
- *Referenced Configuration Requirements for Windows NT* on page 5
- *Licensing Requirements* on page 11
- *Integration Notes* on page 13
- *Known Problems, Limitations, and Updates* on page 25
- *Documentation Updates* on page 51
- *Technical Support* on page 53

## Audience

---

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

## 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 **Rose RealTime Online Documentation** from the **Start** menu.
- For more information on training opportunities, see the Rational University Web site: <http://www.rational.com/university>.

## Contacting Rational Technical Publications

---

To send feedback about documentation for Rational products, please send e-mail to our Technical Documentation Department at [techpubs@rational.com](mailto:techpubs@rational.com).

## Contacting Rational Technical Support

---

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

Your Location	Telephone	Fax	E-mail
North America	(800) 433-5444 (toll free) (408) 863-4000 Cupertino, CA	(781) 676-2460 Lexington, MA	support@rational.com
Europe, Middle East, Africa	+31 (0) 20-4546-200 Netherlands	+31 (0) 20-4546-202 Netherlands	support@europe.rational.com
Asia Pacific	+61-2-9419-0111 Australia	+61-2-9419-0123 Australia	support@apac.rational.com

**Note:** When you contact Rational Technical Support, please be prepared to supply the following information:

- Your name, telephone number, and company name
- Your computer's make and model
- Your computer's operating system and version number
- Product release number and serial number
- Your case ID number (if you are following up on a previously-reported problem)

## Contents

---

This chapter is organized as follows:

- *Overview* on page 1
- *What's New* on page 2
- *PLEASE READ FIRST* on page 3
- *New Rational Developer Network Menu Option* on page 4

## Overview

---

Thank you for selecting **Rational Rose RealTime**, the real-time industry's leading environment that unifies software teams by integrating with best-in-class Rational products including Rational ClearCase, Rational RequisitePro, and Rational SoDA.

These release notes describe:

- *What's New* on page 2
- *PLEASE READ FIRST* on page 3
- *Referenced Configuration Requirements for Windows NT* on page 5
- *Licensing Requirements* on page 11
- *Integration Notes* on page 13
- *Known Problems, Limitations, and Updates* on page 25
- *Documentation Updates* on page 51
- *Technical Support* on page 53

Please read these Release Notes before you install or use Rational Rose RealTime.

**Note:** In some cases, you may note discrepancies between the printed documentation and the online documentation. In these cases, the online documentation is the most up-to-date and should be followed.

Please visit the Rational Web site for the latest Release Notes, patches, and information:

<http://www.rational.com/support>

If you encounter any problems while installing or running Rose RealTime, please check here first to see if you encountered a known problem. If you find a problem that is not yet documented, please call Rational Technical Support so we can investigate it, provide you with a workaround, and track the problem for further action.

Contact information for this release is in *Technical Support* on page 53.

## What's New

---

These are the new features included in this release of Rational Rose RealTime.

- Support for Additional hosts
  - Windows XP
- Online Help Improvements
  - A larger, more comprehensive index.
  - Viewlets - animated demonstrations on a variety of How Do I... topics.
  - Improved tutorials.
- Workflow enhancements
  - Aggregation Tool - enables you to quickly create, modify, and manage aggregate and composite associations.
  - Attribute Tool - enables you to quickly create and set options for an attribute.
  - Operation Tool - enables you to quickly create and set options for an operation.
  - Ability to add class operations directly from a Sequence diagram.
  - Ability to use the target build specification when building a model - see `rtsetup.pl`.
- Debugging
  - Source debugging integration with Microsoft Embedded Visual Tools (EMVT).
  - Basic model debugging with Microsoft Windows CE.
- Test Models
  - Used to validate your target configuration after you make a configuration change.

- Installation Changes
  - Rational Rose RealTime for UNIX is now included on the Rational Suite DevelopmentStudio for UNIX CD. This new rs\_install process includes many enhancements to quickly get you started with Rational Rose RealTime.

## PLEASE READ FIRST

---

Please read the following notes before you install Rational Rose RealTime.

### **You Must Install License Keys to Run this Product**

With your software shipment, you received an Welcome letter containing your Start-up License Key Certificates for this product. You need these keys in order to run your software. The start-up keys will expire a few weeks after shipment. Ensure that you request your permanent license keys as soon as they are available. The availability date for your permanent keys is indicated in the top section of your Start-up License Key Certificate.

**Please check the *Rational Rose RealTime Installation Guide* for detailed instructions on how to install your license keys and how to request permanent license keys.**

**Note:** If you are installing license keys on a Unix platform, please refer to the Installation Guide for Instructions. **Do not** follow the instructions on the Rational Start-up License Certificate or on the envelope in which the certificate is delivered to you.

### **Notice To Customers Running Earlier Rose RealTime Releases on Windows**

You cannot run earlier versions of Rose RealTime on Windows NT, Windows 2000, and Windows XP Pro with the latest version of Rose RealTime at the same time.

## New Rational Developer Network Menu Option

---

The Rational Developer Network provides guidance to implement and deepen your knowledge of Rational tools and best practices. It includes immediate access to white papers, artifacts, code, discussions, training, and documentation. For more information, see <http://www.rational.net>.

From Rational Rose RealTime, click

**Help > Rational on the Web > Rational Developer Network** to open the Rational Developer Network.

# Platform and Toolchain Requirements

# 2

## Contents

---

This chapter is organized as follows:

- *Referenced Configuration Requirements for Windows NT* on page 5
- *Referenced Configuration Requirements for Windows 2000* on page 6
- *Referenced Configuration Requirements for Windows XP Pro* on page 6
- *Referenced Configuration Requirements for UNIX* on page 7
- *Toolchain Requirements* on page 7

This chapter describes the platform and toolchain requirements for running Rational Rose RealTime.

**Note:** Rational Rose RealTime is not supported on Windows 98, or Windows XP Home.

## Referenced Configuration Requirements for Windows NT

---

The minimum supported configuration for running Rose RealTime on Windows NT is:

- Windows NT 4.0 with service pack 6a
- Minimum Pentium 150 MHz. We recommend 500 MHz or faster CPU
- Minimum 128 MB of RAM. We recommend 256 MB of RAM
- Minimum 325 MB of disk space for the Rose RealTime installation
- Minimum display 1024 X 768. We recommend 1280 X 1024 or better
- Postscript printer for printing
- Browser requirement - Internet Explorer 5.01 or 5.5 or Netscape Navigator 4.7 or 6.0. We recommend Internet Explorer 5.01 or 5.5

## Referenced Configuration Requirements for Windows 2000

---

The minimum supported configuration for running Rose RealTime on Windows 2000 is:

- Windows 2000 Professional, with service pack 1 and service pack 2
- Minimum Pentium 150 MHz. We recommend 500 MHz or faster CPU
- Minimum 128 MB of RAM. We recommend 256 MB of RAM
- Minimum 325 MB of disk space for the Rose RealTime installation
- Minimum display 1024 X 768. We recommend 1280 X 1024 or better
- Postscript printer for printing
- Browser requirement - Internet Explorer 5.01 or 5.5 or Netscape Navigator 4.7 or 6.0. We recommend Internet Explorer 5.01 or 5.5

## Referenced Configuration Requirements for Windows XP Pro

---

The minimum supported configuration for running Rose RealTime on Windows XP Pro is:

- Minimum Pentium 300 MHz. We recommend 500 MHz or faster CPU
- Minimum 128 MB of RAM. We recommend 256 MB of RAM
- Minimum 325 MB of disk space for the Rose RealTime installation
- Minimum display 1024 X 768. We recommend 1280 X 1024 or better
- Postscript printer for printing
- Browser requirement - Internet Explorer 5.01 or 5.5 or Netscape Navigator 4.7 or 6.0. We recommend Internet Explorer 5.01 or 5.5



## Referenced Configuration Requirements for UNIX

---

The minimum supported configuration for running Rose RealTime on UNIX is:

- Solaris 2.6, Solaris 2.7, Solaris 2.8, or HP-UX 10.20
  - For Solaris operation, the minimum workstation is an UltraSparc 10. We recommend an UltraSparc 60 with 512 MB of RAM. We recommend the Solaris 2.8 operating system.
  - For HP-UX operation, we support installation of the HP 700 series architecture
  - See the Rational Rose RealTime web site (<http://www.rational.com/support>) for a list of the required UNIX patches applicable to your operating system.
- Minimum 256 MB of RAM. We recommend 512 MB of RAM with approximately three times this amount in swap space
- Minimum 370 MB of disk space for the Rose RealTime installation

## Toolchain Requirements

---

For Toolchain requirements see the following:

- *Help Viewer (Windows Platforms Only)* on page 7
- *Compiler* on page 8
- *Real-Time Operating System* on page 8
- *Changes to Referenced Configurations* on page 8

### Help Viewer (Windows Platforms Only)

The Help Viewer requires that Microsoft Internet Explorer (version 3.02 or later) be configured on a user's computer. It is not required that Internet Explorer be used as the system's default browser, or that the Internet Explorer icon be visible on the user's desktop.

If you choose not to have Internet Explorer as the default browser, you will need to run Hhupd.exe. This file is the distribution executable that installs the run-time components needed for an HTML Help Project, such as Hh.exe, Hhctrl.ocx, Itss.dll, and Itircl.dll. Hhupd.exe is in the HTML Help Workshop/Redist folder.

## Compiler

You must have a C++ compiler installed on your system to make use of the code generation and execution capabilities for Rose RealTime. Different compilers are required for host workstation and for embedded system targets. The list of supported compilers and targets is provided in the *Rational Rose RealTime Installation Guide*.

## Real-Time Operating System

If you are planning to deploy your model on a real-time operating system, your operating system, hardware, and tool line-up must be one of the referenced configurations listed in the *Rational Rose RealTime Installation Guide*. If you do not have one of the referenced configurations, you may be able to obtain support for your configuration from a Rational RoseLink partner, or by customizing the Rose RealTime Services Library for your target. See the *C++ Reference* or *C Reference* for instructions on customizing the Services Library and compiling for new target configurations.

## Changes to Referenced Configurations

Table 1 shows the referenced host configurations and targets for Rational Rose RealTime.

**Table 1**    **Host configurations**

<b>Toolset Host</b>
Solaris 2.6
Solaris 2.7
Solaris 2.8
Windows NT 4.0 (Service Pack SP6a)
Windows 2000 (Service Packs SP1 and SP2)
Windows XP Pro
HPUX 10.20

If you use a referenced configuration other than that tested by Rational and listed in the topic "*Referenced Configurations and Toolchain Requirements*", in the *Rational Rose RealTime Installation Guide*, standard support will cover problems encountered by customers only to the extent that the problem is reproducible on the configuration listed in the *Rational Rose RealTime Installation Guide*.

**Note:** The following target RTS is not included in the Referenced Configurations and Targets list. Although Rational Rose RealTime will continue to include its build configuration, testing was not performed

Host Platform	Target RTOS	Compiler/Processor	RTS	DCS
N/C - native compilation only	IRIX 6.	gnu 2.8.1	C++	-



## Contents

---

This chapter is organized as follows:

- *Checking the Validity of Your License Keys* on page 11
- *License Usage* on page 11

Please refer to the *Rational Rose RealTime Installation Guide* for instructions on installing a startup license and obtaining a permanent license.

**Note:** If you are installing licenses on a UNIX platform, **do not** follow the instructions on the Rational Start-up License Certificate or on the envelope in which the certificate is delivered to you.

## Checking the Validity of Your License Keys

---

If you upgrade to Rational Rose RealTime 6.4 from Rose RealTime releases 6.0, 6.0.1, or 6.0.2, your license keys are not valid. For information on obtaining new license keys, see "Requesting License Keys" in the *Rational Rose RealTime Installation Guide*.

If you upgrade to Rational Rose RealTime 6.4 from Rose RealTime releases 6.1, 6.1.1, 6.2, or 6.3, your license keys are valid.

For more information on license keys, see *Installing License Keys* on page 61.

## License Usage

---

A toolset launched manually will require a license.

A second toolset session launched manually on the same platform will require a second license.

A toolset launched as a COM Automation server will NOT require a license unless it is explicitly made visible by setting the RRTEI Application object's Visible property to True. For more information on RRTEI, see the *Extensibility Interface Reference* in the Rational Rose RealTime online Help.

A toolset launched with a compiled script passed as a command line argument will require a license since the toolset will be made visible before running the script.

A toolset launched with both the **-runScriptAndQuit** and a compiled script passed as a command line argument, will not require a license since the toolset will not be made visible when running the script.

**Note:** A toolset session, regardless of how it is initiated, will require a license if the toolset is made visible.

## License Location

We recommended that you save your license files in a safe location.

## Contents

---

This chapter describes how to integrate Rational Rose RealTime with other tools. This chapter is organized as follows:

- *Microsoft Development Environment* on page 13
- *Configuration for Operation with Rational ClearCase* on page 13
- *Integration with Rational Robot* on page 13
- *Rational ClearCase on a UNIX Server and Clients on Both NT and UNIX* on page 14
- *Rational SoDA* on page 14
- *Rational RequisitePro* on page 14
- *Rational Purify* on page 15

## Microsoft Development Environment

---

We recommend that you install the latest service packs available from Microsoft for Visual Studio or Visual C++.

## Configuration for Operation with Rational ClearCase

---

For more information on this integration, refer to the *Guide to Team Development for Rational Rose RealTime*.

## Integration with Rational Robot

---

Installing the 2002.05.00 release of Rational Rose RealTime will interfere with the operation of the 6.1 release of Rational Robot.

We recommend that you upgrade to the 2001A.04.00 release of Rational Robot.

## Rational ClearCase on a UNIX Server and Clients on Both NT and UNIX

---

You can access a ClearCase server on UNIX with Rose RealTime clients running on both NT and UNIX workstations. For more information on integrating these tools, refer to the *Guide to Team Development for Rational Rose RealTime*.

## Rational SoDA

---

Both Rational SoDA and Rose RealTime must be properly installed and licensed before starting. SoDA may be installed either as part of a suite or as an individual product.

Please refer to the product support page at:

<http://www.rational.com/support>

for the latest updates on SoDA integration.

**Note:** To generate a report using SoDA, the Rose RealTime model must be saved at least once. If the Rose RealTime model has never been saved, it will be untitled. An untitled model causes SoDA to generate errors.

## Rational RequisitePro

---

Both Rational RequisitePro and Rose RealTime must be properly installed and licensed before starting. RequisitePro may be installed either as part of a suite or as an individual product. See "RequisitePro" in the *Rational Rose RealTime Installation Guide*.

Please refer to the product support page at:

<http://www.rational.com/support>

for the latest updates on RequisitePro integration.

**Note:** The Rose RealTime Requisite Pro integration does not support the association of a Rose RealTime package with a RequisitePro project. Use Case and Model association is supported.



## Rational Purify

---

### Adding Options to Purify on UNIX

The toolset looks for an installation of Purify by checking for an environment variable named `PURE_HOME`. This environment variable is not set up by installing Rational Purify.

You must set this environment variable manually. The variable need not point to a directory containing Purify, nor is it required to point to a directory. The variable may contain anything, but must be set.

Occasionally, you may need to add options during a Purify'd build on UNIX. For example, Purify on HP needs to know the name of the linker or collector used by **Gnu g++**.

Options can be added by changing `PURIFY_OPTIONS` in the **CompilationMakeInsert** field of the executable component.

The default value of `PURIFY_OPTIONS` (generated in the makefile by the code generator) is:

```
PURIFY_OPTIONS = -log-file=$(BUILD_TARGET).txt -windows=no
```

To accommodate using `g++` on HP, you can add the following, or similar:

```
PURIFY_OPTIONS = -log-file=$(BUILD_TARGET).txt -windows=no  
-collector=/usr/lib/gcc-ld -g++=yes
```

Where the path of to the collector, `gcc-ld` in most cases, should be the path that is specific to your environment.

For proper integration of Purify when running the Purify'd executable from the toolset, you should preserve the default options.

For an explanation of Purify options, see *Running a component instance with Purify* in the *Toolset Guide*.

## Using OSE and Rose RealTime with the Code Warrior Compiler

---

To use OSE and Rational Rose RealTime with the Code Warrior compiler, Rose RealTime expects a `make` file that provides the name of the compiler, linker, and archiver, as well as any flags. This file, which will be called `config.mk`, needs to reside in a compiler specific directory that you create, for example, `ARM7TDMI-SOC-ADS1.1`. The name is comprised of the CPU type and the compiler. You do not have to use this name. This `make` file is used to compile the Rose RealTime

models as well as to build the TargetRTS. Information on how to create these directories and files, as well as how to compile the TargetRTS, is in the Target Porting guides.

### **Overview of the Process**

Create a libset directory (used to describe the compiler) and add a config directory for the RTOS-Compiler. The files in the config directory can be copied (unchanged) from an existing ose3.x config. Copy the make file in to the newly created arm-codewarrior libset directory. The existing OSE port uses PPC-Diab4.1a. This PPC-Diab tools' libset has the three files: ld.pl, libset.mk, and RTLlibSet.h. You can use these as a starting point for deriving an arm-code-warrior libset. The ld.pl is probably not be required for this compiler, or it may be usable as is. The RTLlibSet.h is used to capture any target specific global includes or compile directives (like header file pragmas) and probably is not required.

After the config.mk file is created, you must compile the target libraries. If you chose a compiler configuration name called **ARM7TDMI-SOC-ADS1.1**, then the entire target configuration will look like **OSE33T.ARM7TDMI SOC-ADS1.1**.

### **Why Rational Does Not Ship all Possible Configurations?**

Rational cannot provide all possible out-of-the-box reference configurations because there are too many possible target configurations. Alternatively, our code generators generate target independent code, and we provide a simple adaptable mechanism to make that code run on any target. We attempt to deliver the most common referenced configurations. Operating system vendors may also be interested in performing an adaptation for you.

We do not restrict your deployment to the referenced configurations identified in the Installation Guide. You can modify the following:

- Change the version of the compiler
- Choose different compiler options
- Choose different TargetRTS options to remove features such as debugging
- Compile for a different CPU
- Choose a different CPU
- Compile using a different compiler
- Compile for a different CPU architecture
- Compile for a different RTOS
- Customize the TargetRTS for error logging, start-up, shutdown, and so on.

We deliver all source code for a Rose RealTime application, and this source code, along with the target adaptations, provides the basis for support.

## Contents

---

This chapter is organized as follows:

- *Migrating from Rational Rose and ObjecTime Developer* on page 17
- *Using the New Installation* on page 18
- *Upgrading* on page 22

## Migrating from Rational Rose and ObjecTime Developer

---

To migrate models into Rose RealTime from either Rational Rose or ObjecTime Developer where models were previously stored in a configuration management system, the model must be loaded into Rational Rose or ObjecTime Developer and written out to a single file. For additional information, see the chapter called *Migration* in the *Rational Rose RealTime Installation Guide*.

When importing a model from Rose into Rose RealTime, you are encouraged to resolve any model errors in Rose (**Tools > Check Model**) and fix any unresolved references. In general, Rose is not concerned with unresolved references; however, they are very important in Rose RealTime as they can result in incomplete code generation and compilation errors.

To export the ObjecTime model in a format readable by Rose RealTime, a patch must be applied to the 5.2 or 5.2.1 toolset to format the file in to a single linear form file with all the required information. The patch is available from Rational Customer Support for both the 5.2 and 5.2.1 product release only. Please contact the Rose RealTime Technical Support group at Rational Customer Service for further information.

After the model is imported into Rose RealTime, it can then be stored in the configuration management system.

## Using the New Installation

---

When installing Rational Rose RealTime, review the following:

- *Installation Process* on page 18
- *Installation on Windows* on page 18
- *Installation on UNIX* on page 19
- *Licensing* on page 21
- *Installing Source Files* on page 21
- *Environment Variables for UNIX* on page 21
- *Upgrading to New Version Only (Deleting Earlier Versions) - UNIX* on page 22
- *Upgrading to Rose RealTime 2002.05.00 and While Maintaining Rose RealTime 2001A.04.00 - UNIX* on page 23
- *Reinstalling* on page 24

### Installation Process

You will find that the installation of Rational Suite DevelopmentStudio is more involved than the Rational Rose RealTime installation; however, it simplifies the configuration of the Rational Suite and Licensing controls, as well as getting you started quickly with Rational Rose RealTime.

### Installation on Windows

#### Updating Batch Files

If you use a batch file to start Rational Rose RealTime, after you install, you must modify the environment variables to use the new mapped drive. To successfully launch Rose RealTime, you must specify a fully qualified path, including the drive letter. For example, you want to update the `ROSERT_HOME` variable, as well as the launch command path:

```
set ROSERT_HOME=C:\Program Files\Rational\Rose RealTime
set ROSERT_HOST=win32
set ROSERT_LICENSE_FILE=%ROSERT_HOME%\license\license.dat
set path=%ROSERT_HOME%\bin\%ROSERT_HOST%;c:\Program
Files\Microsoft Visual
Studio\Common\VSS\win32;c:\DevStudio\VSS\win32;%PATH%
"C:\Program Files\Rational\Rose RealTime\bin\win32\RoseRT"
```

## Additional settings

For additional environment variables and startup options, see the *Rational Rose RealTime Toolset Guide*.

## Installation on UNIX

The installation process creates sourced setup files that create all necessary links, mapping and environment variables for Rational Rose RealTime. These source setup files should be added to the client .csh or .sh setup files. Administrators can view the setup files in the following locations after installation:

```
source <rational_install_directory>/config/rosert_setup.csh
```

or

```
.<rational_install_directory>/config/rosert_setup.sh
```

**Note:** In the setup files, the install path is a hard coded path. We recommend that you install Rational Suite Development Studio for UNIX to a new directory. Typically, Rational products which build databases use hard-coded UNC paths based on installed directories, registry items, or environment variables.

## Installation Overview

The following provides an overview of the installation process and show the installed UNIX directories and files.

**Note:** Directory and file names are for example purposes only.

**rs\_install** welcome, license agreement phase

sets up install directory structures under an entered directory name  
other directories created

**install options and license server phase**

1. *Use existing server*  
license file or server already setup  
e.g. <port>@<servername>
2. *Permanent or TLA license*  
imports license\_XXX.upd into a rational.dat file and stops & restarts the license server

3. *Temporary license*

installs applications  
based on selection, installs application into appropriate directories

**rs\_install** automatically creates setup scripts which should be sourced in the .cshrc or .profile or .login as appropriate

**RoseRT environment variables**  
RoseRT depends on environment variables which are set up by sourcing the setup files created during install

Use the appropriate setup script for the shell being used and the installed setup file

e.g. for csh  
source <rational\_dir>/rosert\_setup.csh

e.g. for ksh

Entered dir name < rational\_dir >  
e.g. /ratlserver/Rel2002/RRT.2002

/base  
/config  
/releases  
/DevelopmentStudioUNIX.2002.05.00

1. modifies setup files when installed

2. /config/rational.dat created

3. /config/temporary.dat created

created and filled  
/releases/RoseRT.2002.05.00  
/releases/purify.2002.05.00  
/releases/RUP\_Gen.2002.05.00

rs\_setup.csh, rs\_setup.sh (suite install)  
or  
rosert\_setup.csh, rosert\_setup.sh  
(RoseRTonly) install

rs\_setup.xxx  
ROBERT\_HOME  
ROBERT\_LICENSE\_FILE  
ROBERT\_HOST  
Purify, RUP and other variables env settings are also set

rosert\_setup.xxx  
ROBERT\_HOME  
ROBERT\_LICENSE\_FILE  
ROBERT\_HOST

## Licensing

The installation process uses a new version of FLEXlm (7.0). Licensing files and licensing servers are set up as part of the installation process and are stored under Rational Suite Development Studio directories.

## Installing Source Files

The Rational Suite Development Studio - RealTime UNIX suite has two types of configurations: HP-UX and Solaris. Rational Rose RealTime installs HP-UX and Solaris executables to a separate directory. If you have a specific configuration, you must install from the appropriate source. If you have two configurations, we recommend that you install the source to different directories.

## Environment Variables for UNIX

With older versions of Rational Rose RealTime, you had to specify environment variables, such as ROSERT\_HOME, ROSERT\_HOST, and others. Now, you only need to set LM\_LICENSE\_FILE.

The Rational Suite DevelopmentStudio - RealTime (UNIX) installation process creates two files in *<rational\_dir>* which should be sourced to set up the environment variables for Rose RealTime. Depending on a Rational Suite or Point Product installation, the files created are:

### **Rational Suite:**

rs\_setup.csh and rs\_setup.sh.

### **Point Product:**

rosert\_setup.csh and rosert\_setup.sh file.

These files automatically configure the Rational Rose RealTime environment and Rational Rose RealTime Suite environment settings.

The rosert\_setup.csh or rosert\_setup.sh file configures environment variables, including those for the other Rational Suite products. The sourced setup files set the following:

- ROSERT\_HOME
- ROSERT\_HOST
- ROSERT\_LICENSE\_FILE
- Other products' environment variables (SodA, Reqpro, Purify environment variables are set by the setup files according to the setup specified)

You will have to set the environment variable for `CONNEXIS_HOME`. For additional information, see the *Rational Rose RealTime Installation Guide*.

When starting Rational Rose RealTime, it looks for the `$ROSET_LICENSE_FILE` first. If it does not find the files, it then looks for the `$LM_LICENSE_FILE`.

## Upgrading

---

When upgrading from an older version of Rational Rose RealTime:

For UNIX, we recommend removing existing setups to minimize possible conflicts between a Rational Rose RealTime version 2001A.04.00 installation and the new Rational Suite DevelopmentStudio - RealTime version 2002.05.00 installation.

You will need to create a directory for the installation path (*<rational\_dir>*).

For additional information on licensing, see the *Licensing* chapter in the *Rational Rose RealTime Installation Guide*.

### Upgrading to New Version Only (Deleting Earlier Versions) - UNIX

You cannot install Rational Suite Development Studio -Rose RealTime to the same directory as an older installation of Rose RealTime.

To uninstall an existing version, see the chapter "Uninstalling Rational Rose RealTime" for your specific platform.

**Note:** Ensure that you make copies of any files added to, or modified in the existing Rational Rose RealTime installation prior to uninstalling. The uninstall process may remove these files.

You can load models created in earlier versions of Rose RealTime directly into 6.4 (also referred to as 2002.05.00). To convert your existing Rational Rose and ObjecTime Developer models, see *Migrating from ObjecTime Developer 5.2/5.2.1* on page 79.

**Note:** Do not attempt to load workspaces created in earlier versions of Rose RealTime, as they are not compatible with the new release.

If you are upgrading Rose RealTime on any of the UNIX platforms, you must do one of the following:

- Manually delete your `~/registry` directory before you run the new version for the first time
- or
- Add the `"-recreate_registry"` command line option the first time you run the new version.



## Checking the Validity of Your License Keys

If you upgrade to Rational Rose RealTime 6.4 from Rose RealTime releases 6.0, 6.0.1, or 6.0.2, your license keys are not valid. For information on obtaining new license keys, see "Requesting License Keys" in the *Rational Rose RealTime Installation Guide*.

If you upgrade to Rational Rose RealTime 6.4 from Rose RealTime releases 6.1, 6.1.1, 6.2, or 6.3, your license keys are valid.

For more information on license keys, see *Installing License Keys* in the *Rational Rose RealTime Installation Guide*.

## Upgrading to Rose RealTime 2002.05.00 and While Maintaining Rose RealTime 2001A.04.00 - UNIX

Your UNIX environment can continue to have a Rational Rose RealTime 2002.05.00 installation and an earlier release of Rational Rose RealTime that uses UNIX environment variables. Refer to the following pseudo code to set up your environment to use both releases of Rational Rose RealTime (.csh or .sh setup):

if your current softlink is set to an old version

set up the following environment variables

```
ROBERT_HOME
ROBERT_HOST
ROBERT_LICENSE_FILE
```

else

```
source <rational_dir>/rosert_setup.xxx
    (where .xxx represents .csh or .sh)
```

set up the following

```
CONNEXIS_HOME to $ROBERT_HOME/Connexis
```

If you maintain older versions of Rational Rose RealTime after installing the latest version:

- update the path for RoseRT executable
- depending on a Rational Suite or Point Product installation, the files created for the Rational Suite are:

```
rs_setup.csh and rs_setup.sh.
```

and for the Point Product:

```
rosert_setup.csh and rosert_setup.sh file.
```

## Reinstalling

When re-installing, the user account originally used to install Rational Rose RealTime must be used because there was a local `.rs_install.<name>` file created for the initial installation which retains any install settings. These original install settings are retained in the original user `.rs_install.<name>` file paths.

## Contents

---

This chapter is organized as follows:

- *Installation and Startup Issues* on page 25
- *General Issues* on page 26
- *Integrations Issues* on page 33
- *Online Help Issues* on page 35
- *UNIX Issues* on page 37
- *RQA-RT* on page 40
- *Updates* on page 42
- *Problems Addressed* on page 46

## Installation and Startup Issues

---

The installation and startup issues are:

- *File Association for Compiled Scripts* on page 25
- *Suite Objects.dll Not Found* on page 25
- *Start-up Problems* on page 26
- *Uninstall* on page 26

### File Association for Compiled Scripts

Windows NT: Rose RealTime does not install a file association for compiled scripts (.ebx). This means that they cannot be automatically run by double-clicking on the file from Windows Explorer.

### Suite Objects.dll Not Found

When installing on Windows NT, or attempting to run Rose RealTime, an error message sometimes occurs indicating that "suite objects.dll" cannot be found in the path.

Adding "C:\Program Files\Rational\common" to the path will fix this problem.

This problem only occurs on Windows NT systems that have had a previous version of Rose RealTime installed.

## Start-up Problems

If Rose RealTime has problems starting up, look at the task bar and use the Task Manager to see if there are running copies of Rose RealTime. Terminating all running copies allows new copies of the tool to start properly.

**Note:** More than one copy of Rose RealTime can run at the same time; however, if startup problems exist, find and terminate any runaway processes.

## Uninstall

Occasionally, files are left behind after an Uninstall. For example, if a model was saved in one of the Rose RealTime subdirectories, the subdirectory and its parents will not be removed. You must remove these manually if you wish to return your system to a clean state.

## General Issues

---

This topic describes the following issues:

- *Unique Ids* on page 27
- *Probes* on page 27
- *ReliableService.ebx Crashes When Running -runScriptAndQuit* on page 28
- *License Manager* on page 28
- *Changes to Code that Uses Default Arguments* on page 28
- *Spaces in Directory Names* on page 29
- *Too Many Files Open* on page 30
- *Executing Component Instances* on page 30
- *Sequence Diagrams* on page 30
- *Build Dependencies on Case-Insensitive File Systems (NT only)* on page 30
- *Using the Debugger-xxgdb Tool and Running your Component Instance* on page 31
- *Running Multiple Sessions on the Same Workstation* on page 31
- *Use of C and C++ Add-ins* on page 31
- *Symbolic Links with TargetRTS* on page 31
- *McAfee VirusScan Affects Startup and Shutdown* on page 31
- *Creating Capsules with the Same Name in Different Packages* on page 32
- *Code Generator Runs Out of Memory When Generating Very Large Models* on page 32
- *Window Order Policy* on page 32
- *Running Model Examples* on page 33
- *Using Get Method and Set Method in the Attribute and Operation Tools* on page 33

## Unique Ids

Unique ids are unique internal names associated with model elements. They are used internally by Rational Rose RealTime, and not all model elements require unique ids. Rational Rose RealTime includes a feature that helps Model Integrator by generating unique ids for those model elements that would otherwise not require them, for internal use. For Model Integrator, an element with a unique id is easier to merge.

For information on **how** and **when** to enable the Unique ids, see “Model Specification” in the online Help.

RRTEI users will find traceability easier when they set this option. Unique ids improve the traceability of model elements of other tool integrations that use RRTEI.

It is necessary to plan and choose when to incorporate the new unique ids into the project model since virtually all controlled units will be modified implicitly. Additionally, the generated new ids are dependent on time and location. For example, generating unique ids for a given model at different times, or on different machines, produces different ids.

**Caution:** We strongly recommend any team involved in parallel development use this option.

**Caution:** Setting this option creates unique ids for model elements that currently do not have them. This typically affects most of the model, so you will be prompted to check out those parts when setting this option.

When saving the model, the size of the affected file increases by approximately 20%, and the time to load the model also increases.

**Caution:** Do not set this option in multiple streams, otherwise, objects with similar characteristics will be treated differently since their unique id’s will differ.

For more information about when and how to use this feature, see the *Guide to Team Development for Rose RealTime*.

**Caution:** *If you clear this option, your merge results will not be as reliable.*

## Probes

When a message hit causes a probe to halt the execution of a running model, there is no way of finding out which probe in the model causes the model to stop executing. To determine which probe execution is halted on, open trace windows for each enabled probe.

## ReliableService.ebx Crashes When Running -runScriptAndQuit

When attempting to run the example RQA-RT model, called **ReliableService.rtm**, from the Command Prompt with the RRTEI script called **ReliableService.ebs**, and using the startup option **-runScriptAndQuit**, the model execution will abort. The example model runs well if you run the script from within Rose RealTime

Do not use the **VerifyBehavior** command from RRTEI with the **-runScriptAndQuit** option because the verification process may not complete in this mode.

For example, if you use the **-runScriptAndQuit** option, Rose RealTime runs the script specified at the Command Prompt. Immediately after the script finishes, it sends a **WM\_CLOSE** message to itself, which is equivalent to selecting **File > Close**. However, the RRTEI command **RunVerifyBehavior** works asynchronously from the toolset; it returns control to Rose RealTime before verification completes. We recommend that you make this command the last command in the script.

**Note:** Asynchronous mode is required for RQA-RT to function properly.

### Solution:

Do not use the option **-runScriptAndQuit** with RQA-RT. You can continue to specify the name of the **ReliableService.ebs** script at the Command Prompt.

## License Manager

When there is no License manager running, a pop-up window appears indicating that the toolset cannot obtain a license at that time.

## Changes to Code that Uses Default Arguments

ObjecTime Developer models which used the **RTTimespec** constructor with only one parameter, as in the following code:

```
timer.informIn(RTTimespec(2));
```

will result in a compile error after conversion of the model to Rational Rose RealTime. The compile error will appear something like:

```
..\rtg\Driver.cpp(67) : error C2440: 'type cast' : cannot convert from  
'const int' to 'struct RTTimespec'
```

```
No constructor could take the source type, or constructor overload  
resolution was ambiguous.
```

The reason is that in ObjecTime Developer, the `RTTimespec` constructor included default arguments, that is, `RTTimespec (long=0, long=0)`. The default constructor values are not supported with `RTTimespec` in Rose RealTime. Any code that made use of the default arguments needs to be changed to supply both constructor arguments. For example:

```
OTD = RTTimespec (2);
```

must be changed to:

```
RRT = RTTimespec (2, 0);
```

## Spaces in Directory Names

For Windows NT, to allow the use of cross-compilers that do not allow spaces in the path names, use the `subst` command and map a drive to the value of `%ROBERT_HOME%` after the installation. For example, if you wish to use the K: drive, and your Rose RealTime installation directory is:

```
ROBERT_HOME=C:\Program Files\Rational\Rose RealTime
```

you must map this directory to the drive by running the following commands from a console window:

```
subst K: "%ROBERT_HOME%"
```

```
set ROBERT_HOME=K:
```

Spaces in directory names can cause problems with the following *operating system.compiler library set.development platform*: systems:

OSE411T.ppc603-Diab-4.3f.NT4

VRTX4T.ppc603-Microtec-1.4.NT40

TORNADO2T.ppc630-GreenVX-1.8.9.NT40

TORNADO2T.ppc630-GreenVX-2.0.NT40

TORNADO2T.m68040-cygnus-2.7.2-960126.NT40

TORNADO2T.ppc-cygnus-2.7.2-960126.NT40

TORNADO2T.ppc860-cygnus-2.7.2-960126.NT40

For Tornado configurations, you should also use the substituted drive to point to the load script directory when running Target Observability.

On the detail tab of the processor specification, the `Load Script` path should not contain any spaces. If spaces are present, when attempting to load the executable in Basic or Debugger-Tornado[2] modes, the error message, **Unable to Execute** appears. This does not occur when loading the executable in manual mode.

Spaces in directory names can also cause problems with ClearMake.

## Too Many Files Open

You can have a maximum of 1024 file pointers open simultaneously.

## Executing Component Instances

If you receive the message “Unable to connect to target” when attempting to connect to a target (both host and embedded), change the value for the **Connect Delay** box on the Component Instance specification by increasing the value by two or more seconds.

## Sequence Diagrams

There are a few conditions under which the Sequence Diagram will incorrectly draw Messages / FOCs. If a Message or FOC appears to be incorrectly drawn, select the Message (or the Message that starts the FOC) and, using the center 're-orient' handle, slightly move the Message. This will force the Diagram to recalculate the correct display values for that message.

## Build Dependencies on Case-Insensitive File Systems (NT only)

During a build, Rose RealTime detects and records build dependencies for comparison during subsequent builds; this is done to facilitate build-avoidance by only regenerating or recompiling targets when a build dependency changes. These build dependencies preserve the case of the filenames involved, even when the underlying file-system (for example, NTFS) is case-insensitive. This may cause problems when using names which are distinct within the toolset, and distinct on case-sensitive drives, but indistinct on case-insensitive drives. In most cases, the toolset or code-generator will identify and avoid or warn against potential case-insensitive name collisions.

However, some case-insensitive filename collisions cannot be detected. For example, if a component is renamed to something that is case-insensitive-identical (for example, renaming from "myComponent" to "MyComponent"), a build may incorrectly reuse all previous build results, since the underlying build dependencies will be indistinct according to the file-system.



Users are advised to use case-sensitive file-systems if possible. Otherwise, users should avoid case-insensitive name collisions when they create or rename classes, components or controllable units.

## Using the Debugger-xxgdb Tool and Running your Component Instance

When using the xxgdb tool, run your component instance, add any breakpoints, and then restart it to enable the breakpoints in **gdb**.

## Running Multiple Sessions on the Same Workstation

If you experience instability when running multiple Rose RealTime sessions on the same workstation, limit your usage to a single session at any given time.

## Use of C and C++ Add-ins

You cannot use C and C++ Add-ins at the same time in Rational Rose RealTime.

## Symbolic Links with TargetRTS

When using LynxOS 3.1.0, do not install Visual Lynx 3.1.0 for Windows NT on a network (NFS) disk. It should only be installed on a local NTFS drive; otherwise, symbolic links to some **include** directories will not work properly. This will cause compilation errors if you re-compile the TargetRTS.

If you have installed Visual Lynx 3.1.0 on a network disk, and if you see compilation errors stating that **include** directories **netinet/in.h** or **net/if.h** are not found, locate the entries **net** and **netinet** in the following directory:

```
$LYNX_HOME/usr/lynx/3.1.0/ppc/usr/include
```

If these entries are text files containing the following text: **!<symlink>bsd**

rename these files and create new symbolic links called **'net'** and **'netinet'** that both point to the directory **bsd**.

## McAfee VirusScan Affects Startup and Shutdown

On Startup or shutdown, McAfee VirusScan will scan the RoseRT.ini file approximately 426 times, causing a very slow startup or shutdown.

To enable a much faster shutdown, exclude the INI extension from the McAfee VirusScan Program File Extensions list.

## Creating Capsules with the Same Name in Different Packages

While Rose RealTime allows users to create capsules with the same name in the different packages, Target Observability for both C++ and Java may confuse such capsules.

As a temporary workaround, we recommend either renaming such capsules to have unique names (preferred method), or placing the capsules to be observed directly under the **Logical View**.

## Code Generator Runs Out of Memory When Generating Very Large Models

When you attempt to generate a very large model, the code generator may run out of memory.

To improve memory usage when generating very large models, we recommend that you control all the controllable elements in your model as individual units.

### To control elements in your model:

- 1 In the **Model View** tab in the browser, select the model.
- 2 Right-click and click either **File > Control Units** or **File > Control Child Units**.
- 3 When prompted to control all child units recursively, click **Yes**. Also click **Yes** on the subsequent confirmation dialog.

Controlling all the controllable elements partitions your model into individual files; one file for each controllable element. For details on controlled units, see the topic, “What is a controllable Element?” in the *Guide to Team Development - Rational Rose RealTime*.

## Window Order Policy

When using the CDE window manager, to ensure that the proper Secondary and Transient window policy is in effect, in .Xdefaults, set the following environment variable:

```
Dtwm*secondariesOnTop: True
```

Setting this variable to True ensures that an opened secondary window in Rational Rose RealTime for UNIX, such as an external editing window, does not get caught behind the main primary window. Since the secondary window is the active window, you are unable to regain focus of this secondary window.

If `Dtwm*secondariesOnTop` is not set to `True`, since the secondary window is the active window, you may not be able to regain focus of this secondary window when it goes behind the primary window.

## Running Model Examples

When trying to compile some example model installed with Rational Rose RealTime, such as `SocketInterfaceExample`, you must have the Rational Rose RealTime Companion Products installed to successfully compile example models.

## Using Get Method and Set Method in the Attribute and Operation Tools

In the **Attribute** and **Operations** tools, you can de-select the **Get method** and **Set method** boxes. If you want to prevent deletion of modified code for the Get and Set methods, do the following:

- 1 Open the specification for an **Attribute** or **Operation**.
- 2 Click the **General** tab.
- 3 In the **Documentation** box, delete the text

```
//GENERATED BY ATTRIBUTE TOOL
```

or

```
//GENERATED BY OPERATION TOOL
```

Deleting this text prevents deletion of the modifications you made to the Get or Set methods if you de-select **Get method** and **Set method**.

## Integrations Issues

---

This topic describes the following issues:

- *Error Loading Large Models from ClearCase - Windows* on page 33
- *RequisitePro* on page 34
- *SoDA* on page 34

### Error Loading Large Models from ClearCase - Windows

If you have problems loading large models from ClearCase - that is, it takes a long time or an empty model appears - check your error messages. It may be because there are too many files open.

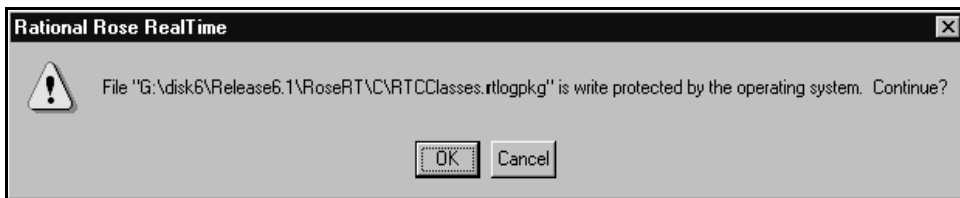
### To resolve this problem:

- 1 Go to the Control Panel, and double-click on the **ClearCase** applet.  
The **ClearCase Properties** dialog appears.
- 2 Click the **MVFS Performance** tab.
- 3 Under **Mnodes**, click the **Override** boxes for both **Maximum number of mnodes to keep on the free list** and **Maximum number of mnodes to keep for cleartext free list**, and set the value for both to **800**.
- 4 Click **OK** to apply the changes and close the dialog.
- 5 Reboot the Windows computer.

## RequisitePro

When the RequisitePro add-in is enabled in Rose RealTime, there is a known problem with **File > Save As**.

There will be numerous prompts similar to the following:



as RequisitePro attempts to save the units of RTClasses and RTCClasses. The number of prompts will be less if only one of C or C++ is enabled. Clicking **OK** to all the prompts causes no harm. Other possible workarounds are disabling the RequisitePro add-in for the **Save As** (ensure the model is saved before disabling) or removing the RT packages if they are not required at this time.

Please refer to the product support page at:

<http://www.rational.com/support>

for the latest updates on RequisitePro integration.

## SoDA

SoDA can be installed without Rose 2001 and will continue to function with Rose RealTime. However, when run from Rose RealTime, the menu of templates presented are from Rose, not Rose RealTime. Some of these templates will issue errors.

Please refer to the product support page at:

<http://www.rational.com/support>

for the latest updates on SoDA integration.

**Note:** To install the Rose RealTime domain for SoDA, you need to select the Full/Customer install option during your SoDA installation.

## Online Help Issues

---

This topic describes the following issues:

- *Problems Accessing Rose Help when Concurrently Running Rose RealTime on Windows NT* on page 35
- *Java Reference* on page 36
- *Navigating through the QuickStart Tutorial Online Help* on page 36
- *Viewing Animated Demonstrations* on page 36

### Problems Accessing Rose Help when Concurrently Running Rose RealTime on Windows NT

#### Problem

Running both Rose RealTime and Rose may cause Rose not to access its help files.

#### Explanation

To enable the help for the Rose add-ins to be accessible when used from within the Rose RealTime toolset, Rose RealTime temporarily replaces a registry key value installed by Rational Rose. This registry key is located in [HKEY\_LOCAL\_MACHINE\Software\Rational Software\Rose] and is named "HelpFileDir".

When the toolset starts, it substitutes this key's value for its own Help directory. The value installed by the Rational Rose installation program is backed up and restored when the last instance of the running toolset exits.

#### Workaround

You can disable this behavior by setting the value of the key located in [HKEY\_LOCAL\_MACHINE\Software\Rational Software\Rose RealTime\6.4] whose name is "ReplaceRoseHelpDir" to "No".

## Java Reference

The formatting of some text (including code) may not appear correctly in the Java Reference online Help book. Please refer to either the `jguide.pdf` file or the *Java Reference* book to verify the content.

## Navigating through the QuickStart Tutorial Online Help

You can either select a topic in the **Contents** tab or click the **Next** or **Previous** buttons, and the topic appears in the main window.

**Note:** If you click **Previous**, although the previous topic appears in the main window, the incorrect topic may be highlighted in the **Contents** tab. However, this does not affect the navigating to the selected topic.

## Viewing Animated Demonstrations

Rational Rose RealTime includes animated demonstrations of various How Do I... topics, called Viewlets. Viewlets can be viewed on a number of platforms; however, your browser must have Java 1.1 and JavaScript support to play a Viewlet. Additionally, Java must be enabled in your browser.

**Note:** It is normal for some larger Viewlets to take 10 to 15 seconds to load. Ensure that your browser cache is not disabled; otherwise, some Viewlets may take longer to load.

If launching a Viewlet takes a very long time, or a gray box displays instead of running a Viewlet, try running the Viewlet in your browser by doing the following:

- 1 In your browser, navigate the Rational Rose RealTime /Help directory.
- 2 Select `WatchThisViewlet.htm`.
- 3 Select a Viewlet to run that demonstration.

## Windows 2000 Issues

---

You will encounter problems running Rational Rose RealTime 2002.05.00 on Windows 2000 if you log on using User. The types of problems include the following:

- When opening Rational Rose RealTime, you may see the error message, "Failed to Update the System Registry. Try using REGEDIT."
- The configuration of the Add-in manager cannot be restored.
- The two menu items **Add Class Dependencies** and **Component Wizard** are missing from the **Build** menu.

- When creating a component, you are not able to define the **Environment in Component Specification** for the **C++ TargetRTS**.
- When selecting **Rebuild** from the **Build** menu, or clicking the **Build** tool from the Toolbar, there is no activity.
- After selecting **Run** from the **Build** menu, then clicking **Yes** to select **Build the component**, you will receive the error message **Operation not allowed**.
- When clicking Help > About, there is no Version or Company information.
- You will not be allowed to set a Top Capsule in the component specification.

To run Rational Rose RealTime on Windows 2000, you must log on with Administrative or Power User privileges.

## UNIX Issues

---

This topic describes the following issues:

- *Executing the Current Version with a Previous Version* on page 37
- *Refresh Problems with Exceed* on page 38
- *HP-UX Make Dependency Filename Restriction* on page 38
- *Non-GUI-based External Editors* on page 38
- *Case Sensitivity within Paths* on page 39
- *Troubleshooting Rose RealTime Toolset Freezes on UNIX* on page 39
- *Online Help* on page 40

### Executing the Current Version with a Previous Version

The current and previous toolsets do not work together by default, due to sharing the `MWRPC_ENDPOINT` and `MWSHM_KEY` environment variable settings. These settings are only calculated if the environment does not already have them set.

**To execute a current and a previous toolset from the same host or shell:**

- 1 Invoke the first toolset with the `-env` parameter (and `&` to have it run in the background)
- 2 Find the `MWRPC_ENDPOINT` environment variable and note the value, for example, `10633`

- 3 Set the *MWRPC\_ENDPOINT* environment variable for the second toolset to be one greater than the previous:

```
setenv MWRPC_ENDPOINT 10634
```

*MWRPC\_ENDPOINT* values must be between 10000 and 65000 to be valid.

- 4 Set the *MWSHM\_KEY* to the same value as the *MWRPC\_ENDPOINT*

```
setenv MWSHM_KEY 10634
```

*MWSHM\_KEY* environment variable values must be integers less than 1048576, to be valid. We recommend that you use the RPC value for convenience.

- 5 Invoke the second toolset with the `-env` parameter to confirm environment settings.

## Refresh Problems with Exceed

The screen sometimes does not refresh completely when running the UNIX version of Rose RealTime and displaying it on a PC using Hummingbird Exceed.

### To changing your Exceed settings:

- 1 Start Xconfig.
- 2 Open the **Performance** dialog.
- 3 Use the following settings:

**Save Unders** = No

**Maximum Backing Store** = When Mapped

**Default Backing Store** = None

**Minimum Backing Store** = None.

## HP-UX Make Dependency Filename Restriction

On the HP-UX platform, the HP-UX Make restricts dependency filenames to 160 characters. We recommend the use of Gnu-Make to work around this problem. Alternatively, you may shorten the path names to your class files to avoid this issue.

## Non-GUI-based External Editors

On UNIX, the toolset will freeze when the user specifies `"/bin/vi"` as the external editor and then tries to launch the external editor from the code edit pane. If you are using a non-GUI-based external editor, this should be specified using `"xterm -e /bin/vi"` so that the editor has a terminal (tty) to display to.



## Case Sensitivity within Paths

The UNIX temporary directory name is translated to all lower case. If you set the environment variable TEMP, make sure the path name is all lower case or the directory will not be found. This will cause problems when Help is started.

## Troubleshooting Rose RealTime Toolset Freezes on UNIX

Occasionally, the Rose RealTime toolset freezes (does not respond to user input, window will not refresh, etc.). We have been unable to fully characterize these problems.

### If you encounter a toolset freeze, please complete the following steps:

- 1 Run "top" from your UNIX shell and save the output to a file.
- 2 If on Solaris, run `/usr/proc/bin/pstack <Rose RealTime pid>` and save the output to a file.
- 3 Kill the Rose RealTime process.
- 4 Run `Rose RT -cleanup`.
- 5 Attempt to reproduce the problem.
- 6 Note Rose RealTime build information (Help / About Rational Rose RealTime) and the operating system information.
- 7 If you are using ClearCase, were you in a ClearCase view at the time of the freeze and what version of ClearCase are you using?
- 8 Please send all the above information to Technical Support (see *Technical Support* on page 53 for contact information), and use "Rose RealTime" in the subject of your email message.

## Toolset freezing on startup

Under certain configurations, when starting Rose RealTime on Solaris and displaying to an exceed X-server, the "Welcome to ..." window appears momentarily, then disappears. After this, the main application GUI is locked out and freezes.

### The workaround is as follows:

- 1 Kill the Rose RealTime process.
- 2 Edit the `RoseRT.ini` file in the `~/registry.2002.05.00` directory and change `ShowStartupDialog=Yes` to `ShowStartupDialog=No`.

3 Run *RoseRT -cleanup*.

4 Run *RoseRT*.

## Online Help

### Context-sensitive help on UNIX may not invoke the correct help page

Context-sensitive help on UNIX may not invoke the correct help page. The workaround is to use the advanced search capabilities of the online-help to find the appropriate topic.

### Cannot open some links with online help

Some links to multimedia content do not work on UNIX.

The online Help browser on UNIX does not support hyperlinks to external files. Instead, use the Contents tab in the online Help; it contains the .pdf books in online format.

## RQA-RT

---

This topic describes the following:

- *Limitations* on page 40
- *Additional documentation on create messages and sequence diagrams* on page 41
- *Additional documentation on send message specification data field format for Java* on page 42

### Limitations

States in a specification sequence diagram are not referenced to states in a model (they are independent).

RQA-RT is not supported for C models.

Only leaf node instances in the interaction/sequence diagram can be specified as a driver/stub.

Within a sequence diagram, each interaction instance without a specified role must have a unique name. Only one interaction instance per test set can be left unnamed.

Within a sequence diagram, if you run multiple tests in a single pass, a port from a CUT can be used by only one interaction instance with an unspecified role.

As it is impossible to specify cardinality index for the interaction instances with no roles specified, care should be taken then using such objects with capsules under test and/or ports with cardinality greater than 1.

If the generated trace sequence diagram is directly compared with the specification diagram, it is important to remember that such comparison is asymmetric – sequence diagram should be used in place of sequence diagram; trace diagram should be used in place of trace diagram.

When running multiple specifications in a test in sequence, ensure that you complete each sequence when using a CUT, so that the CUT is self-contained. This prevents a CUT from replying to a message from a CUT in the previous sequence diagram.

In a controlled model when you generate a new harness with affected model elements not checked out, many warnings will be generated.

In RQA-RT Java, there may be some problems when specifying deeply nested replicated subcapsules as drivers/stubs. The RQA-RT test harness will not be able to distinguish particular driver instances if driver cardinality is greater than 1. This might lead to unexpected behavior as settings of the first driver instance (like minimum run time) will be used for all driver instances. This limitation does not apply to the drivers present on the top level of driver selection dialog and drivers with cardinality equal to 1.

## **Additional documentation on create messages and sequence diagrams**

The **Item Properties** in the Create Message Specification dialog, with the exception of **THREAD**, only apply if you are creating a Capsule Under Test (CUT). **THREAD** applies in every case. The **Item Properties** entries are as follows:

- Capsule class - A capsule name. Enter a value here only if you want to override the default capsule class associated with the role that is associated with the interaction instance being created.
- Initial data - Enter a value here only if your Capsule Under Test (CUT) requires data on startup. To provide data you must specify an attribute in the driver sending the create message with the appropriate initial values. You cannot provide initial data in create messages sent from the environment.
- Data Descriptor - If you provided initial data, you must specify the type descriptor of the type of data that you specified. This is in the format **RTType\_<type of initial data>**.
- Thread name.

## Additional documentation on send message specification data field format for Java

- For the data to be passed with the messages, the data type should be specified as `<data_type> <constructor_arguments>`. For example, to pass Integer object referring to the number 5, place the following line: `Integer 5` or `java.lang.Integer 5`.
- If `<data_type>` is omitted, `Integer` is assumed.
- The `<constructor_arguments>` should contain the exact line to be passed as an argument to constructor. For example, for the `MyObject(Integer, String)` the following line can be used: `MyObject 5, "Acme"`.

**Note:** No additional brackets or quotes need to be placed around `<constructor_arguments>`.

## Updates

---

Rational Rose RealTime includes the following updates:

- *Code Generator Updates* on page 42
- *Model File Syntax* on page 44
- *.jar Files* on page 44
- *rtsetup.pl File* on page 44
- *Hot Link for Call Message Operations* on page 45

## Code Generator Updates

### Description

The code generator will issue a warning message when the action code for an inherited transition is duplicated in child capsule code.

### Background

The action of a transition or expression of a choice point becomes the body of a C++ operation with two parameters, `rtdata` and `rtport`.

For example:

```
void MyCapsule_Actor::transition2_t1( const MyData * rtdata,
                                     MyProtocol::Base * rtport )
{
    // transition action code
}
```

**rtdata:** The **rtdata** parameter points to the data in a message. It is a pointer to the lowest common denominator superclass of the possible data classes in all the signals (regardless of the port) that trigger the transition. The argument type must be the most common base class, or **void \*** if there is none.

**rtport:** This represents the port on which the current message was received. It is a pointer to the common base class of all protocol roles that trigger a transition.

For example, if we have two triggers for a transition, one with portA (ProtocolA - Base) and one based on portB (ProtocolB), the type for **rtport** would be the lowest common denominator in the protocol class hierarchy, which if ProtocolA and ProtocolB were not related by inheritance would be **RTProtocol \*** (the base class for the **TargetRTS**).

## Result

When a transition is inherited from its parent and the action code is not overridden, in most cases, the transition function is not regenerated in the derived (child) capsule's code.

In certain cases, the transition function will be regenerated in the child capsule's code. Additional triggers can introduce new data types or protocol roles to what must be supported by the action code. The result is that the **rtdata** and **rtport** parameters may need to be more general (higher in the class inheritance hierarchy). When the transition function is regenerated with this different signature, your code may no longer compile. The code generator will issue a warning if the inherited action code is not overridden and the context is more general. The warning is:

- For a transition:  
"A duplicate function is generated for this action because its triggers are incompatible with the inherited function."
- For a choice point:  
"A duplicate function is generated for this choice point because its triggers are incompatible with the inherited function."

For example, if the base capsule class has a transition triggered by a signal which conveys data of type **RTString**, the transition in the base class has the formal argument:

```
const RTString * rtdata
```

Suppose a derived capsule class modifies that transition so that it is also triggered by another signal which conveys data of type **RTInteger**, and the action code is inherited. If the function generated for the base class is reused in the derived class, you have a problem: the action code may depend on the data being of type **RTString**, but we cannot be sure of this type. In this example, the action is generated as the body of a new function in the derived class, and the warning is displayed.

Even if your code compiles, you may have a problem at run-time. For example, if CapsuleA contains an inherited transition from its parent which calls its parent's transition code (using SUPERMETHOD).

CapsuleB is a subclass of CapsuleA and adds a trigger to this transition. If the transition code needed to be regenerated in CapsuleB, then this would lead to the transition code in CapsuleA being called twice.

## Model File Syntax

The syntax for model files has changed in Rational Rose RealTime version 2002.05.00. As a result, when opening a model in Rational Rose RealTime 2001A.04.00 that was created in Rational Rose RealTime 2002.05.00, you may receive syntax errors.

## .jar Files

Classes for RTJava applications are now bundled into a single .jar file rather than directory. You may need to modify CLASSPATH in your environment, as well as other locations where Java code is compiled or run outside the toolset.

## rtsetup.pl File

There is a new Perl script, called `rtsetup.pl`, which you can use to recreate the environment used to build the TargetRTS. The primary usage will be in the **CompilationMakeCommand**: you want to prefix the existing command with the following:

```
rtperl -S rtsetup.pl
```

For example, if the command was

```
$defaultMakeCommand
```

it becomes

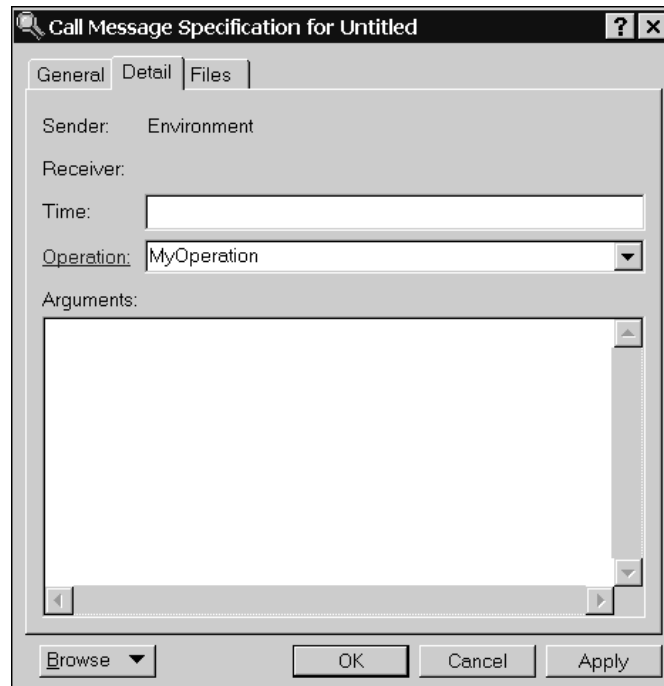
```
rtperl -S rtsetup.pl $defaultMakeCommand
```

The script examines the generated makefile to determine the **TargetConfiguration** specified for the current component, executes `$RTS_HOME/config/<TargetConfiguration>/setup.pl` to recreate the environment required to build the TargetRTS, and then, in that environment, executes the command that follows.

It is useful for designers who must build components for different targets which depend on conflicting environment variable definitions.

## Hot Link for Call Message Operations

For a Sequence diagram, if you open the **Call Message Specification** dialog and select an **Operation** from the **Detail** tab, the label opposite the **Operation** box become a hot link to that operation.



## Problems Addressed

---

The following customer originated cases have been addressed in this release:

**Table 2 Problems addressed for Rose RealTime**

Defect Number	Description
rose00050338	RRTEI: From documentation cannot figure how to modify TopCapsule property
rose00048995	Pressing F1 in RoseRT brings up help system but there are no index entries
rose00054912	Shutdown of the RTS panel should trigger the Verify Behaviour session to end
rose00049539	Documentation is missing descriptions of RTFRAME configuration options
rose00050271	Services Library Class Reference in C++ Language Guide has organizational defects
rose00054995	Build reuse restrictions with RQA-RT
rose00051269	RQA-RT should not be specifying use of RTInteger
rose00055082	RQA-RT Search Tool Does not look in Sequence Diagrams
rose00055090	RQA-RT Problem Converting OTD MSCs when multiple variables of the same name
rose00054000	Importation is not supported in RQA-RT
rose00053965	UNIX Exception: 1cHCsocketSCancelBlockingCall6M_v_
rose00054514	RQA-RT Converts as RPL code even if blank
rose00054377	RQA-RT: improve error messaging for protected driver ports
rose00047825	Online help is missing an image
rose00047895	Missing keywords in Help index
rose00047960	Inconsistency in operator== depending on use of RTDataObject as a base class
rose00048370	Unix Exception: _doprnt
rose00048550	Wrong string.h is included when codegendir is not defined (using STL classes)
rose00048638	Change defaults... button doesn't work in dependency specification



**Table 2 Problems addressed for Rose RealTime**

Defect Number	Description
rose00048741	Exception Services examples need to be corrected in C++ book & online help
rose00048831	Add limitations of opening Rose models & the C++ Analyzer to documentation
rose00048850	Single Threaded T.O. not working -- add note to documentation
rose00048924	Should be note about how to deregister REGISTER_LAYER functions
rose00048949	Card Game Tutorial - provide more info on build/not build component at run-time
rose00048950	Card Game Tutorial - Add info on use of the "Step" button in the RTS browser
rose00049526	On-line Help Documentation Typo
rose00049584	Omission of step of script code in RRTEI Tutorial
rose00049705	Help-> Index is blank in RoseRT on Unix
rose00049789	No documentation on how to start Model Integrator from command line on Unix
rose00050268	RRTEI Menu statement does not work
rose00050346	Source debugging on Tornado 2, XxSim, Solaris does not bring up debugger automatically
rose00050347	Qualified name for capsule roles, transition, ... does not give full model path
rose00050386	Quickstart Tutorial instructs to add a log port even if using C language
rose00050401	Installation Guide and Release Notes mention docs that are not sent with package
rose00050437	UNIX Exception: __1cjCTitleFTSFQdDuery6MpwLppnMIITResultSet_pnPCFullTextSearch_pnGCUWait_i_l_
rose00050527	C++ Guide: RTTimespec basic comparison operators gives inappropriate example
rose00050796	Tornado: RTS panel stuck in Initial state for simso using PPP instead of ULIP
rose00050807	frame.incarnate prototypes do not match documentation
rose00051000	RTMI: Model Integrator fails to report conflict in Capsule and Class specs

**Table 2      Problems addressed for Rose RealTime**

<b>Defect Number</b>	<b>Description</b>
rose00051116	RoseRT for C, possible to bind SAP and SPP based on incompatible protocols
rose00051118	RoseRT C, able to register 2 SPPs with same name
rose00051119	RoseRT C, buffered SAP registrations not bound when other SAP deregisters
rose00051148	Error in script CreateClasses.ebs on Solaris 2.6
rose00051166	Error in script CreateProtocol.ebs on Solaris 2.6
rose00051170	CheckOut and UnCheckOut have no "Do not retrieve" option
rose00051181	RoseRT 6.x Debugger limit
rose00051188	Code generation for class scoped public attribute in C incorrect
rose00051209	RTMI - 62-Toolset-01.win32 - corrupted merge
rose00051214	Error in IntegratingData model in copy constructor of PointerInts
rose00051231	Model process consumes full CPU - source code debugging C Model on NT VC++6
rose00051232	Release Notes, RTD_CONNEXIS_BUILD in pdf file and not online help
rose00051238	RQART: Cardinality editor not working properly
rose00051249	releasenotes.pdf has Asia-Pacific support contact information wrong
rose00051250	RRTEI: Automating Model Validation
rose00051251	RRTEI: SaveLogAs not documented
rose00051258	RQART: Problem selecting harness generation and result location under CM
rose00051264	Unix Exception - __1cHDiagramLupdateTitle6M_v_
rose00051276	Customizing DCS library section should state what needs to be done for a patch
rose00051283	Exception: 1cHObjectIsKindOf6kMpknNCRuntimeClass__i_
rose00051286	Incorrect Target Observability command line parameters
rose00051290	EXCEPTION_ACCESS_VIOLATION at 001B:00491571 when deleting capsule

**Table 2 Problems addressed for Rose RealTime**

Defect Number	Description
rose00051291	EXCEPTION_ACCESS_VIOLATION in module <?> after Undo of inheritance
rose00051304	EXCEPTION_ACCESS_VIOLATION in module RoseRT.exe when running script
rose00051338	RQART: "Error: Ports specified in the message are not connected"
rose00051359	C TargetRTS: layer registration table re-indexed after SAP deregister
rose00051378	Incorrect example of frame.incarnate(), passing data to a specific thread
rose00051395	Unix exception: __1cOCOTreeListCtrlTUUpdateChildControls6M_v_
rose00051408	Unix: Alt-graph key combinations not working
rose00051409	RoseRT timing inaccuracy
rose00053478	Unix Exception: __1cLCompareKeys6Fpc0pv_i_ (on-line help)
rose00053863	OTD -> RoseRT conversion problem Sending Base instead of Sub-class
rose00053929	Missing information in MSCs converted by RQA-RT conversion tool
rose00054204	Doc error - explains C addin has synchronous messaging but this is not supported
rose00054316	Codegen fatal error: parse error while reading return type [friend ostream &]
rose00054323	Error Creating Attribute using the RQA-RT Conversion Tool with parametrized constructors
rose00054329	Error Creating Attribute using the RQA-RT Conversion Tool with pointers as an attribute
rose00054522	RQA-RT Conversion Issue on Unix - problems with new lines
rose00055108	RQA-RT has unnecessary modal operations
rose00055171	Duplicating a collaboration diagram with a sequence overlay causes duplicated signals
rose00055297	NT Exception: 001B:6C372550
rose00055730	RoseRT crashes when a public port is introduced to a capsule - KERNEL32.dll, SetUnhandledExceptionFilter()+92 byte(s)
rose00056206	RoseRT: Windows toolset crash, Unix crash :__1cKOrdCollectKgetIndexOf6kMpknKRoseObject__i_

**Table 2      Problems addressed for Rose RealTime**

<b>Defect Number</b>	<b>Description</b>
rose00056266	RQA-RT cannot handle "return" even if it is in a string
rose00056466	indexTo returns -1 when other end of replicated port is not bound
rose00056694	"Error: Parameter 1 contains an invalid value." not very informative
rose00056744	Unix Exception: 1cSCOleDispatchDriverNInvokeHelperV6MIHHpvpkC1_v_
rose00057156	Possible bug in RTFormat::_double
rose00057157	Incorrect mantissa width calculation in RTFormat::_double
rose00057618	ReliableService.ebx crashes when running -runScriptAndQuit
rose00058254	Documentation: Topics/Links under "Concepts Tutorials" are missing
rose00058926	RT Java: Unable to inject messages with patch 6.3.114.0
rose00059041	RQA-RT Toolset Crash during Verify Behaviour.
rose00059140	Cut+Paste in RoseRT option tab caused a EXCEPTION_ACCESS_VIOLATION in module <UNKNOWN>
rose00059260	Codegen change to signals that use typedef (pointer) as data
rose00059783	RQA-RT Toolset Crash during Verify Behaviour.

# Documentation Updates

# 7

For the most recent documentation updates, please visit the Product Support section of the Rational Rose RealTime web site at:

<http://www.rational.com/support>



## Contents

---

This chapter is organized as follows:

- *Submitting Problem Reports* on page 53
- *Submitting Feature Requests* on page 54
- *Submitting Support Requests* on page 55
- *Contacting Rational Customer Service by Email or Telephone* on page 56

This chapter describes how to submit problem reports, feature requests and support requests to Rational Customer Service.

## Submitting Problem Reports

---

With Rational Rose RealTime, you can email problem reports to the Rational Software Customer Service department that services your location. When you email a problem report directly from the Rose RealTime application, a wizard guides you through the process, ensuring that you provide the correct information to the Rational Technical Support team. This information includes contact and location information, and a detailed description of the problem that you are reporting.

### To submit a problem report:

- 1 From the **Help** menu, click **Email Technical Support**.

A submenu appears, providing you with three options.

- 2 Click **Problem Report**.

The General Information dialog appears.

- 3 Type your contact and location information in the text areas provided and click **Next**.

The Problem Report - Additional Information dialog appears.

- 4 In the **Defect Title** text area, type a detailed name for the problem that your are reporting.

- 5 Select the type of problem that you are reporting from the appropriate list boxes.
- 6 Describe the problem, using the categories provided in the **Details** area.
- 7 Click **Next**.  
The Email Summary dialog appears.
- 8 Ensure that the information that appears in the Email Summary dialog is accurate.  
**Note:** The email information displayed in the Technical Support Email Address is chosen based on the location information that you provided in the General Information dialog. It is not recommended that you edit the email address.
- 9 If you want to save or print a copy of the email, click the appropriate button.
- 10 Click **Send Email** to send your email.

## Submitting Feature Requests

---

With Rational Rose RealTime, you can email feature requests to the Rational Software Customer Service department that services your location. When you email a feature request directly from the Rose RealTime application, a wizard guides you through the process, ensuring that you provide the correct information to the Rational Software Customer Service department. This information includes contact and location information, and a detailed description of the feature that you are requesting.

### To submit a feature request:

- 1 From the **Help** menu, click **Email Technical Support**.  
A submenu appears, providing you with three options.
- 2 Click **Feature Request**.  
The General Information dialog appears.
- 3 Type your contact and location information in the text areas provided and click **Next**.  
The Feature Request - Additional Information dialog appears.
- 4 In the **Request Title** text area, type a detailed name for the Feature that you are requesting.
- 5 Select the level of urgency for the feature that you are requesting.
- 6 Describe the feature, using the categories provided in the **Details** area.



- 7 Click **Next**.

The Email Summary dialog appears.

- 8 Ensure that the information that appears in the Email Summary dialog is accurate.

**Note:** The email information displayed in the Technical Support Email Address is chosen based on the location information that you provided in the General Information dialog. It is not recommended that you edit the e-mail address.

- 9 If you want to save or print a copy of the email, click the appropriate button.

- 10 Click **Send Email** to send your email.

## Submitting Support Requests

---

With Rational Rose RealTime, you can email Support requests to the Rational Software Customer Service department that services your location. When you email a Support request directly from the Rose RealTime application, a wizard guides you through the process, ensuring that you provide the correct information to Rational Software Technical Support department. This information includes contact and location information, and a detailed description of the support request that you are submitting.

### To submit a support request:

- 1 From the **Help** menu, click **Email Technical Support**.

A submenu appears, providing you with three options.

- 2 Click **Support Request**.

The General Information dialog appears.

- 3 Type your contact and location information in the text areas provided and click **Next**.

The Support Request - Additional Information dialog appears.

- 4 In the **Request Title** text area, type a detailed name for the request that you require.

- 5 Select the level of urgency for the question with which you need help.

- 6 Type your question in the Question text area.

- 7 Click **Next**.

The Email Summary dialog appears.

- 8 Ensure that the information that appears in the Email Summary dialog is accurate.  
**Note:** The email information displayed in the Technical Support Email Address is chosen based on the location information that you provided in the General Information dialog. It is not recommended that you edit the e-mail address.
- 9 If you want to save or print a copy of the email, click the appropriate button.
- 10 Click **Send Email** to send your email.

## **Contacting Rational Customer Service by Email or Telephone**

---

When contacting Rational Customer Service by email or by telephone, please be prepared to supply the following information:

- Name, telephone number, and company name
- Product name and version number
- Operating system and version number (for example, Windows NT 4.0, Windows 2000, Windows XP, Solaris 2.6/2.7/2.8, or HP-UX 10.20)
- Computer make and model
- Your case id (if you are calling about a previously reported problem)
- A summary description of the problem, related errors, and how it was made to occur

If your organization has a designated, on-site support person, please try to contact that person before contacting Rational Customer Service.

You can obtain technical assistance by sending electronic mail to the appropriate email address. Electronic mail is acknowledged immediately and is usually answered within one working day of its arrival at Rational. When sending an email place "Rational Rose RealTime" in the subject line, and in the body of your message include a description of your problem.

When sending email concerning a previously-reported problem, please include in the subject field: "CaseID: v0XXXXX", where XXXXX is the caseid number of the issue. For example:

```
CaseID: v0176528 New data on rational rose realtime install issue
```

Sometimes Rational Customer Service engineers will ask you to fax information to help them diagnose problems. You can also report a technical problem by fax if you prefer. Please mark faxes "**Attention: Technical Support**" and add your fax number to the information requested above.

Telephone, fax, and email information for Rational Customer Service are Table 3. If you have problems or questions regarding licensing, please see *License Support Contact Information* on page 57.

**Table 3 Telephone and fax and email information**

Your Location	Telephone	Fax	E-mail
North America	(800) 433-5444 (toll free) (408) 863-4000 Cupertino, CA	(781) 676-2460 Lexington, MA	support@rational.com
Europe, Middle East, Africa	+31 (0) 20-4546-200 Netherlands	+31 (0) 20-4546-202 Netherlands	support@europe.rational.com
Asia Pacific	+61-2-9419-0111 Australia	+61-2-9419-0123 Australia	support@apac.rational.com

## License Support Contact Information

If you have a problem or questions regarding the licensing of your Rational Software products, please contact the Licensing Support office nearest you.

Telephone numbers for license support are listed in the following table. Ask for, or select, Licensing Support.

**Table 4 License Support Telephone and Fax**

Region	Telephone Number	Fax Number
Americas	800-433-5444	781-676-2510
Europe, Israel, and Africa	+31 (0)20 4546 200	+31 23 554 10 69
North Asia Pacific (Mainland China, Hong Kong, Taiwan)	+852 2143 6382	+852 2143 6018
Korea	+82 2 556 9420	+82 2 556 9426
South Asia Pacific Australia, New Zealand, Malaysia, Singapore, Indonesia, Thailand, The Philippines, Vietnam, Guam and India	+61-2-9419-0111	+61 2 9419 0123
Japan	+81 3 5423 3611	+81 3 5423 3622

Email addresses for license support are listed in the following table.

**Table 5 License Support Email**

<b>Region</b>	<b>Email Address</b>
Americas	lic_americas@rational.com
Europe, Israel, and Africa	lic_europe@rational.com
North Asia Pacific Mainland China, Hong Kong, Taiwan, and Korea	lic_apac@rational.com
South Asia Pacific Australia, New Zealand, Malaysia, Singapore, Indonesia, Thailand, The Philippines, Vietnam, Guam and India	lic_apac@rational.com
Japan	lic_japan@rational.com

# Index

## A

- accessing
  - Rose Help when Concurrently Running Rose RealTime on Windows NT 35

## B

- build
  - Dependencies on Case-Insensitive File Systems 30

## C

- capsules
  - creating with the same name in different packages 32
- Case Sensitivity within Paths 39
- Case-Insensitive File Systems 30
- changes to Code that Uses Default Arguments 28
- code generator
  - running out of memory for large models 32
  - warnings 43
- code generator updates 42
- Code that Uses Default Arguments 28
- CompilationMakeCommand 44
  - rtsetup.pl 44
- configuration
  - Operation with Rational ClearCase 13
- configuration requirements
  - UNIX 7
  - Windows 2000 6
  - Windows NT 5
  - Windows XP Pro 6
- contacting Rational technical publications 21
- contacting Rational technical support 22

- creating
  - Capsules with the Same Name in Different Packages 32
  - messages and sequence diagrams 41

## D

- debugger
  - xxgdb (issues) 31
- Default Arguments 28
- directory names
  - spaces in 29
- documentation feedback 21
- Documentation Updates 51

## E

- Environment Variables 21
- External Editors 38

## F

- Feature Requests
  - submitting 54
- File Association for Compiled Scripts 25
- Files
  - too many opened 30

## H

- Host configurations 8
- HP-UX Make Dependency Filename Restriction 38

## I

- installation issues 25

- installing
  - on Unix 22
  - upgrade information for Unix 22
- Installing Source Files 21
- Integration Issues 33
- integration Notes 13, 17

## J

- .jar Files 44
- Java
  - send message specification 42

## K

- known problems 25

## L

- License
  - location 12
- License Keys 3
- License Manager 28
- License Support Contact Information 57
- License Usage 11
- Licensing 21
- Limitations 25
- LM\_LICENSE\_FILE 21

## M

- McAfee VirusScan
  - affects of 31
- Microsoft Development Environment 13
- migrating
  - from Rational Rose and ObjecTime Developer 17
- model
  - unique Id 27
- models
  - code generator may run out of memory 32
- Multiple Sessions on the Same Workstation 31

## O

- Objects.dll Not Found 25
- online Help Issues 35

## P

- packages
  - issues when creating capsule names 32
- paths
  - case sensitivity 39
- platforms (see referenced configurations) 8
- probes 27
- Problem Reports
  - submitting 53
- problems addressed 46
- PURIFY\_OPTIONS 15

## Q

- QuickStart Tutorial Online Help 36

## R

- Rational ClearCase
  - configuration 13
  - loading large models from 33
  - on a UNIX Server and Clients on Both NT and UNIX 14
- Rational Purify 15
- Rational RequisitePro 14
- Rational Robot
  - integration with 13
- Rational SoDA 14
- Rational technical publications
  - contacting 21
- Rational technical support
  - contacting 22
- referenced configuration requirements
  - Windows XP Pro 6
- referenced configurations and targets 8
- requirements
  - referenced configurations 6
  - toolchain 7

- UNIX 7
- Windows 2000 6
- Windows NT 5
- Windows XP Pro 6
- RequisitePro 34
- RQA-RT 40
  - Limitations 40
- rtsetup.pl File 44

## S

- send message specification data field format for
  - Java 42
- Sequence Diagrams 30
- SoDA 34
- Source Files
  - installing 21
- spaces in Directory Names 29
- Startup Issues 25
- Start-up Problems 26
- submitting
  - Feature Requests 54
  - Problem Reports 53
- submitting Support Requests 55
- Suite Objects.dll Not Found 25
- Support Requests
  - submitting 55
- Symbolic Links with TargetRTS 31

## T

- TargetRTS
  - rtsetup.pl 44
  - symbolic Links 31
- targets
  - supported 8
- Technical Support
  - contacting 56
- Toolchain requirements 7
  - Changes to Referenced Configurations 8
  - Compiler 8
  - Help Viewer 7
  - Real-Time Operating System 8
- toolset freezing on startup 39

- troubleshooting
  - Rose RealTime Toolset Freezes on UNIX 39
  - warnings from code generator 43

## U

- Uninstall 26
- unique Id's 27
- unique Ids 27
- UNIX
  - additional settings 19, 24
  - Environment Variables 21
  - executing the Current Version with a Previous Version 37
  - Rational ClearCase 14
  - Refresh Problems with Exceed 38
  - requirements 7
  - Toolset freezing on startup 39
  - troubleshooting when toolset freezes 39
- UNIX Issues 37
- unning
  - Multiple Sessions on the Same Workstation 31
- using
  - C and C++ Add-ins 31

## W

- what's new 2
- Windows 2000
  - requirements 6
- Windows 2000 Issues 36
- Windows NT
  - requirements 5
- Windows XP Pro
  - requirements 6

## X

- xxgdb 31

