



*Getting Started with
Rational Rose
RealTime*

Copyright © 1993-1999 Rational Software Corporation. All rights reserved

Copyright © 1999 ObjecTime Limited. All rights reserved.

Portions Copyright © 1992-1999 Summit Software Company. All rights reserved.

Part Number: 800-012304-000

Revision 6.01, May 1999

This document is subject to change without notice.

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 227.7202-1(a) and 227.7202-3(a) (1995), DFARS 252.227-7013(c)(1)(ii) (Oct 1988), FAR 12.212(a) (1995), FAR 52.227-19, or FAR 52.227-14, as applicable.

Rational, the Rational logo, Rational Rose, ClearCase, and Rational Unified Process are trademarks or registered trademarks of Rational Software Corporation in the United States and in other countries. All other names are used for identification purposes only and are trademarks or registered trademarks of their respective companies.

ObjecTime (and clock design) and ObjecTime are registered trademarks of ObjecTime Limited in Canada, the United States and in other countries.

Microsoft, Windows, Windows NT, and Visual SourceSafe are trademarks or registered trademarks of Microsoft Corporation.

T Quatrani VISUAL MODELING WITH RATIONAL ROSE AND UML, (pages 3, 4, 29, 39, 73, 142, 153). © 1998 Addison Wesley Longman Inc., -Reprinted by permission of Addison Wesley Longman. All rights reserved.



Contents

Chapter 1	Introduction	7
	Overview	7
	Product Information	7
	Overview of Rose RealTime capabilities	8
	Year 2000 compliance	9
	Contact information	10
	When contacting Rational technical support	10
	Rational web site	10
	How to contact Rational Customer support	10
	Evaluation and ordering information numbers	11
	The telephone numbers for Rational Software Corporation's Technical Support are listed below:	11
	Installation keys	12
	Model examples	12
	Accessing the online help system	12
	Configuring your browser	12
	Latest updates	13
Chapter 2	Model Upgrades and Conversions	15
	Opening models from Rational Rose	15
	Limitations and restrictions	16
	Importing Rational Rose Generated Code	17

	Limitations and restrictions	18
	Opening models from ObjecTime Developer 5.2/5.2.1	18
	Limitations and restrictions	19
Chapter 3	Supported Platforms	21
Chapter 4	Documentation Roadmap	25
	Hardcopy and online information	25
	Tours and tutorials	26
Chapter 5	Development Environment Integration	27
	Configuration management (CM) tools integration	27
	Requirements management tools integration	28
	RequisitePro REI emulation start-up	28
	Directory contents	29
Chapter 6	Starting Rational Rose RealTime	31
	Starting Rose RealTime under WindowsNT:	31
	Starting Rose RealTime under UNIX:	31
	Start-up options for UNIX:	32
	-regedit	32
	-startuplicense	32
	-license	32
	-recreate_registry	32
	-q -quiet	32
	-v -verbose	32
	-cleanup	32
	Running Rose RealTime in batch mode	33
Chapter 7	Troubleshooting	35
	CD read errors	35
	Install/Uninstall problems	35
	Install will not proceed for non-Administrators.	35
	Uninstall leaves incrementally installed Rose RealTime files on the disk.	35
	Online help system	36
	Mailing exception files	37

Icon display	37
CM Library problems	38
Compilation problems	39
Compile fails on valid C++ models with VC++ 5.0 or VC++ 6.0	39
Error linking Capsule (“error from nmake”)	39
Windows NT Compilation Command Line Limits	39
Source File Compilation	40
Linking	40
Known limitations and restrictions	40



Chapter 1

Introduction

Overview

Product Information

Rational Rose RealTime is a comprehensive visual development environment that delivers a powerful combination of notation, processes, and tools to meet the challenges of real-time software development. Through the industry-standard Unified Modeling Language (UML), real-time design constructs, code generation, and model execution capabilities, Rational Rose RealTime addresses the complete lifecycle of a project: from early use case analysis, through to design, implementation, and testing.

Rational Rose RealTime is designed for simple insertion into your software development environment, processes, and workflows. Rose RealTime includes seamless integration with other Rational products and support for a variety of commercial real-time operating systems.

Rational Rose RealTime is packaged as two variants:

- Rational Rose RealTime, Developer Edition — UML for Real-Time (RT) modeling, host and target execution:
 - Ability to produce standalone C++ applications for target operating systems.
 - Distribution license packaged with the product
 - Source code for toolset host and target Service Libraries
 - Target ports for Tornado I, pSOS, VRTX & OSE

- Rational Rose RealTime, Modeler Edition — UML for RT modeling and host execution only:
 - No distribution licenses.
 - No service library source code.
 - Precompiled service libraries for toolset hosts only.
 - Toolset must be running for generated executables to run.

Overview of Rose RealTime capabilities

Modeling:

- Use Case Modeling
- Class Modeling
- Collaboration (role) Modeling
- Interaction Modeling (sequence diagrams)
- Component Modeling
- Deployment Modeling

Application Generation:

- C++ Language Support
- Data Class Code Generation

Visual Execution:

- Host Execution
- Target Execution
- Model Visualization (Animation)
- Model Debugging (Tracing, Injection, Inspection)

Tools interworking:

- Rational ClearCase
- Microsoft Visual SourceSafe (Windows NT only)
- SCCS (UNIX only)
- RCS (UNIX only)
- Rational SoDA (requires Rational Rose RealTime domain)
- Rational Req Pro

Model Documentation:

- Report Generation (WindowsNT only)

Year 2000 compliance

Rational Software Corporation ("Rational") warrants the media on which the Product is furnished to you to be free of defects in materials and workmanship and that the Product will substantially operate according to specifications published by Rational for a period of ninety (90) days from the date of purchase.

The latest revisions of all of Rational's currently available product family on Windows, Windows 95, Windows NT, and supported variants of UNIX rely on standard mechanisms for all operations related to computing and storing the date and time, which encode the year such that the transition to the year 2000 is handled properly. In addition, the Product accurately processes date data (including, but not limited to calculating, comparing, and sequencing) from, into and between the twentieth and twenty-first centuries, including leap-year calculations, when used in accordance with the Product documentation provided. To the extent that the hardware and underlying operating system correctly handle the calendar shift from 1999 to 2000, Rational warrants that the latest revisions of all of Rational's currently available product family on Windows, Windows 95, Windows NT, and on supported variants of UNIX platforms are free from the 'Year 2000 problem', and that all operations involving date and time will remain accurate when the calendar shifts from 1999 to 2000. Rational Products will remain Year 2000 compliant provided the Product remains under current Customer Support Services contract and the updates provided thereunder are applied to the Product.

If there are any additional questions regarding this subject, please contact your local Rational Sales Office.

Contact information

This section describes procedures for interacting with Rational Software Corporation's technical support services.

When contacting Rational technical support

When contacting technical support for Rose RealTime, please be prepared to supply the following information:

- Name, telephone number, and company name.
- Product version number (found in the Help > About dialog).
- Computer make and model.
- Make and version of the operating system.
- Make and version of development tools (configuration management program, compiler, linker, RTOS, requirements management program).
- Your log number (if you're calling about a previously reported problem).

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

Rational web site

You can contact technical support and obtain the latest product information through our web site at:

<http://www.rational.com/products/rosert>

How to contact Rational Customer support

Telephone support is available Monday through Friday (except holidays) from 6:00 a.m. to 6:00 p.m. Pacific time in the United States. International telephone support hours span the normal local business hours.

Evaluation and ordering information numbers

United States and Canada

Rosebud@rational.com

1-800-728-1212

Other Worldwide locations

Rosebud@rational.com

+1-408-863-9900

The telephone numbers for Rational Software Corporation's Technical Support are listed below:

United States and Canada

support@rational.com

1-800-433-5444

Europe

support@europe.rational.com

+31-23-569-4310

Scandinavia

support@europe.rational.com

+46-8-566-28200

Asia-Pacific

support@apac.rational.com

+61-2-9419-0111

Other Worldwide locations

support@rational.com

+1-408-863-5000

Installation keys

Included with your Rose RealTime shipment will be an envelope which contains the installation keys necessary to install the software from the CD media. These keys are unique to your order and the envelope and contents should be kept in a safe place to facilitate future installations should a re-install become necessary. If these keys are misplaced please send a request for replacement keys to Rose RealTime Support (see the Contact Information section for telephone numbers and emails addresses of the Rational Support representatives for your region).

Model examples

A small set of examples is provided in the ModelExamples directory in your Rose RealTime installation. These examples are referenced from the online help.

For the latest examples to get you started with Rose RealTime, please visit our website at:

<http://www.rational.com/products/rosert>

Accessing the online help system

Online help and documentation for Rational Rose RealTime is provided in HTML format. In order for the help system to function correctly, Microsoft's Internet Explorer 4.0 or higher, alternatively Netscape Navigator 4.51 or higher must be installed your system.

Both Microsoft Internet Explorer and Netscape Communicator can be downloaded from the web at the following sites:

<http://www.microsoft.com/products/ie/default.htm>

<http://www.netscape.com/computing/download/index.html>

Configuring your browser

The Rational Rose RealTime Help system is configured to use the default browser when opening the HTML help pages.

Latest updates

For the latest product and documentation updates, please for the Rose RealTime website at **<http://www.rational.com/products/rosert>**



Chapter 2

Model Upgrades and Conversions

Opening models from Rational Rose

Rose RealTime can open files saved with Rational Rose 98 and 98i (.mdl files). To open models saved in more recent versions of Rose—for example, Rose98i Service Pack 1—ensure they are saved in Rose 98 or Rose 98i format before they are opened in Rose RealTime.

Fixing unresolved references

When importing a model from Rose 98/98i into Rose RealTime, you are encouraged to resolve any model errors in Rose98 (Tools > Check Model) before trying to import the model. In particular it is important to resolve any unresolved references. In general, Rose98 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.

Tasks

To open a Rational Rose model in Rose RealTime:

1. Select File > Open and choose Rose Model (.mdl) from the Files of Type pull-down menu.
2. Select the file to open and click Open.

Files from Rose versions older than Rose 98 will have to be opened in Rose 98 and saved first.

Note: *Opening a new model will discard any existing model that you have. The tool will prompt for you to save changes first.*

Import log messages

The following messages may appear in the Log after a Rose98 model has been imported.

Message: Warning: Renamed elementClass “oldElementName” to “newElementName”.

Description: A loaded model element has been renamed to conform with Rose RealTime's naming requirements. Double-clicking on the warning in the log may (or may not) display the renamed element.

Message: Error: Unresolved reference from ... to ... by

Description: The toolset was unable to resolve a reference between two model elements. This is usually the result of loading an incomplete model, for instance when the user has updated only part of a model from CM. The rest of the model needs to be loaded in order for the reference to be resolved. However, in some instances (where toolset stability is an issue) the unresolved model element is removed from the model. If this is the case, the deletion is also recorded in the log window.

Message: Error: Error reading file fileName at line lineNumber or Error message detail.

Description: The error message detail may contain validation errors originating from the internal meta-model, which are not covered here. Possible error message details that originate from the petal reader are listed below.

Message: Invalid syntax.

Description: The file contents cannot be read by the toolset. The user should send the file to customer support with a description of what they were doing when the file was created. Example: Imported a Rose98 model, made some changes to the Component View, now the file won't reload in RoseRT.

Limitations and restrictions

When a Rose model is opened in Rose RealTime, the following elements will not be converted:

- State diagrams

The model will open, but the state diagrams will not be present in Rose RealTime.

- Importing Rose98 models containing controllable units is not supported

If the Rose98 model file contains controllable units. The user should export the model from Rose98 into a single .ptl petal file (File > Export Model) which can then be opened with Rose RealTime (File > Open, and select All Files... in the combo box to display .ptl files.)

- Three-tier class diagrams are not supported in Rose RealTime.

If the Rose98 model file contains a three-tier class diagram, which are not supported in Rose RealTime, the user should create a copy of the Rose98 model that does not contain a three-tier diagram to import into Rose RealTime.

Note: *The conversion of models is supported in one direction only: once models are brought into Rose RealTime, they cannot be converted back to Rose without losing significant amounts of data. Working in a mixed Rose RealTime/Rose98 environment is not supported. Generated code is not compatible between the two tools.*

Importing Rational Rose Generated Code

Source code that has been generated from a Rose model and has been edited within the preserved regions may be imported.

To import Rose generated code:

1. Verify that the Rose .mdl file is not newer than the generated code. If so, regenerate the code.
2. Open the Rose model (see Opening Rose Models).
3. From the tools menu select “Import Code”.

If code was generated from this model using Rational Rose and the model was saved after the code generation was performed, a “Rose Code Import” window is displayed. Otherwise, a “There are no cpp or h files available for import” message is displayed.

The Rose Code Import Window lists all the .cpp and .h files that were generated from the model and lets you select all or a subset of the files. It also displays the classes that will be affected by each file that is selected. Once a file has been imported it will not be listed if code importation is repeated.

4. Once you have complete importation and are satisfied with the results save the model.

Limitations and restrictions

- No action will be taken on empty preserved regions. As a result, constructors, destructor and operators that are generated by Rose, which have empty preserved regions, will not be added to the model.
- Use of the Code Name properties for classes and operations may cause inconsistent naming in the generated code. The inconsistencies may cause compile time errors that can be resolved manually.

Opening models from ObjecTime Developer 5.2/5.2.1

Note: *If you plan to move models from ObjecTime Developer to Rose RealTime, contact ObjecTime support for the latest information and patches.*

Rose RealTime can only import Linear Form files from ObjecTime Developer 5.2/5.2.1. Other kinds of files, such as binary .update or .context files cannot be imported directly into Rose RealTime.

Note: *ObjecTime 5.2/5.2.1 users must apply a patch to their toolset in order to export models from ObjecTime that can be read by Rose RealTime. See the ObjecTime support website.*

To open an ObjecTime Developer 5.2/5.2.1 model:

1. The ObjecTime Developer project file must be saved as a Linear Form file (.lf)
2. To open an ObjecTime Developer model from Rose RealTime, select File > Open and choose Linear Form (.lf) from the Files of Type pull-down menu. Select the file to open and click Open.

Files from versions of ObjecTime older than ObjecTime Developer 5.2 will have to be opened in ObjecTime Developer 5.2 and saved as project files first.

Note: *Opening a new model will discard any existing model that you have. The tool will prompt for you to save changes first.*

Importing requirements

Requirements captured in ObjecTime Developer models can be converted through a requirements-specific patch for 5.2 and 5.2.1. An HTML file will be generated that will contain the actual requirements from the OTD models. Links to these requirements will be converted when the actual model is imported into Rose RealTime. The HTML requirements file is stored outside of the Rose RealTime toolset. Place the file in your configuration management library for storage purposes.

Note: *Contact ObjecTime Support for details.*

Limitations and restrictions

When an ObjecTime Developer model is opened in Rose RealTime, the following elements may not be converted:

- dependencies — the dependencies list for classes in ObjecTime Developer is not converted. Dependencies must be recreated using the **Build > Add Class Dependencies...** command. This runs a script which checks the model elements for dependencies and adds them. It will not, however, find references that exist only in detailed code. For example, data class references declared in transition will result in a compile error.

Note: *Models brought into Rose RealTime from ObjecTime Developer cannot be brought back into ObjecTime Developer. Only one-way conversion is supported.*

- ObjecTime batch mode is not supported. Use the RRTEI to perform batch mode operations.



Chapter 3

Supported Platforms

Table 1 shows the supported host platforms for Rational Rose RealTime 6.01.

Table 1 Host Platforms

Toolset Host
Solaris 2.5.1
Solaris 2.6
Windows NT 4.0
HPUX 10.20

A pre-defined set (ports) of the Rose RealTime Services Libraries are delivered as part of the Rational Rose RealTime product. The Services Library is what allows standalone executable models to be executed on target operating systems. These ports are fully tested by Rational, and are covered by standard Rational support. A standard port can be used to facilitate a port to your environment of choice.

Note: For a more detailed description of the Services Library, refer to the Rational Rose RealTime C++ Programmer's Guide.

A port is based on the following specifics (often called the toolchain line-up):

- OS version
- Compiler version
- Processor type

Chapter 3 Supported Platforms

If you are using a line-up other than the one tested by Rational and listed in this guide, standard support will cover problems encountered by you only to the extent that the problem is reproducible on the line-up listed in this guide.

Table 2 shows the supported platforms and targets for Rational Rose RealTime 6.01.

Table 2 Supported platforms and targets

Toolset Host	Target Library Name	Target Services Library
HPUX 10.20	HPUX10T.hppa-gnu-2.8.1	Supplied
	HPUX10T.hppa-HPC++-10.11	Supplied
Solaris 2.5.1 Solaris 2.6	SUN5T.sparc-gnu-2.8.1	Supplied
	SUN5T.sparc-SunC++-4.2	Supplied
WinNT 4.0	NT40T.x86-VisualC++-5.0	Supplied
	NT40T.x86-VisualC++-6.0	Supplied
Solaris 2.5.1 Solaris 2.6 WindowsNT 4.0	PSOS25T.ppc603-Diab-4.2b (pRISM+: Diab/SDS tools for pSOS 2.5 for PowerPC)	Generate
Solaris 2.5.1 Solaris 2.6 HPUX 10.20	VRTX4T.ppc603-Microtec-1.3C (SPECTRA: Microtec tools for VRTX 4.AB on PowerPC)	Generate
WinNT 4.0	VRTX4T.ppc603-Microtec-1.4 (SPECTRA: Microtec tools for VRTX 4.Baa on PowerPC)	Generate
Solaris 2.6 Solaris 2.5.1 WinNT 4.0	OSE32T.ppc603-Diab-4.1a (Diab 4.1a, SDS 7.1.1 tools for OSE 3.2 for PowerPC)	Generate

Toolset Host	Target Library Name	Target Services Library
HPUX 10.2 0 Solaris 2.5.1 Solaris 2.6 WinNT 4.0	TORNADO101T.m68040-cygnus- 2.7.2-960126 (Tornado 1.0.1: cygnus tools for VxWorks 5.3.1 on 68K)	Generate
	TORNADO101T.x86-cygnus- 2.7.2-960126 (Tornado 1.0.1: cygnus tools for VxWorks 5.3.1 on x86)	Generate
	TORNADO101T.ppc-cygnus- 2.7.2-960126 (Tornado 1.0.1: cygnus tools for VxWorks 5.3.1 on PowerPC)	Generate
	TORNADO101T.i960-cygnus- 2.7.2-960126 (Tornado 1.0.1: cygnus tools for VxWorks 5.3.1 on PowerPC)	Generate

Note: *The following applies to the Tables in this chapter:*

T = Multi-threaded

Supplied = supplied as part of the Rose RealTime installation.

Generate = not supplied as part of the Rose RealTime installation, but can be generated from the target source code that is supplied, if you have the Developer license.



Chapter 4

Documentation Roadmap

Hardcopy and online information

The Rose RealTime documentation set includes both hardcopy and online documentation. In general, you will find that the online documentation is more thorough and more up to date than the hardcopy documentation. We suggest you use the online help system as your primary source of information.

There are many important concepts in Rose RealTime, and the tool is very powerful. It is important that you get off to the right start by learning the underlying concepts so that you know what it is you are trying to accomplish with the tool. We suggest that you start by reading the Introduction and Modeling Language chapters of the Rose RealTime User's Guide. Once you have digested some of the background concept material, you can begin exploring the toolset.

The Rose RealTime C++ Programmer's Guide gives detailed information on how to use the Rose RealTime services library operations, and how to generate the code to implement various modeling constructs in C++. You need not read the C++ Programmer's Guide from cover to cover, but you should read the first few chapters that provide an overview of some of the sophisticated code generation and run-time capabilities available to you.

Finally, if you have the Developer license, you will also have a Rose RealTime C++ Target Guide, which contains detailed internal information about the services library and how to customize it for your application and target.

The online help also includes an Extensibility Interface book, which is not provided in hardcopy. This provides details on the scripting interface for adding new capabilities to the tool, for writing scripts to perform batch operations, and for extracting model information for analysis or reporting.

Tours and tutorials

Included in the online help are a number of tours and tutorials. These are intended to help new users orient themselves to the different capabilities provided by Rose RealTime and the underlying concepts behind the tool.

Tours provide an overview of concepts and tool features aimed at audiences with different backgrounds.

Tutorials provide a very quick look at certain concepts, then give detailed how-to information on how to use different tool features, and then provide hands-on exercises to try.



Chapter 5

Development Environment Integration

Rational Rose RealTime can coexist on the same workstation with any Rational or ObjecTime product. In addition Rational Rose RealTime is shipped with “out-of-the-box” integrations with several popular development tools. It will simplify tool-chain complexity by providing teams with seamless integration to leading real-time operating systems, compilers, symbolic debuggers, and other market-leading Rational Software products. For a list of supported platform ‘line-ups’ see the Supported Platforms section.

Configuration management (CM) tools integration

The following CM tools are supported with integration for Rose RealTime. For more information on integrating these tools refer the User's Guide Configuration Management chapter.

Tools	Version
Rational ClearCase	3.2
Microsoft Visual SourceSafe (NT only)	5.0 and 6.0
RCS (Unix only)	5.7
SCCS (Unix only)	1.2

Requirements management tools integration

The following tools are supported for integration with Rose RealTime.

Tools	Version
Rational SoDA	(requires RoseRT 1.0 domain)
Rational RequisitePro	4.0

RequisitePro REI emulation start-up

In order to use RequisitePro with Rose RealTime, the Rose RealTime toolset must be started up from outside of RequisitePro (the start Rose button in RequisitePro will not work).

In addition, the Rose RealTime toolset has to be started with the `-emulateREI` command line parameter. This parameter is necessary to start Rose RealTime to register a version of the REI which is compliant with the Rose98 REI. This allows applications written to use Rose98 REI, to run with Rose98. The registry for the emulated REI will be overwritten the next time Rose98 starts.

On a command prompt:

```
RoseRT.exe -emulateREI
```

As the properties of a shortcut:

```
"C:\Program Files\Rational\RoseRT.exe -emulateREI"
```

If you run Rose98 at the same time as Rose RealTime, do not use the `-emulate REI` parameter. If you need to run RequisitePro with Rose RealTime, then you cannot run Rose 98i at the same time.

Directory contents

After installation of the main Rose RealTime files has been completed, the directory structure should be as follows. Please ensure that the installation directory `$(ROBERT_HOME)` and all its files are readable, and not writable, by all users of Rose RealTime. The directory and its sub-directories contain all the individual files that comprise the particular release. Some of the files and directories included here are:

`$(ROBERT_HOME)`

This is the top level directory.

Help

This directory and its sub-directories contains the online Help.

`$(ROBERT_HOME)/bin`

The bin directory holds the Rose RealTime executable and various scripts. `$(ROBERT_HOME)/bin` also contains subdirectories for each of the supported workstation platforms `$(ROBERT_HOST)`.

license

This directory contains various files containing encrypted information and is used by the License Manager in order to ensure that Rose RealTime is being executed according to the End-User License Agreement.

Scripts

This directory contains various Rose RealTime scripts.

`C++` or `<language>`

This directory contains the libraries, header files, scripts relating to code generation, and for Rose RealTime Developer, source files, for the Services Library. For more information regarding the Services Library see the User's Guide and Programmer's Guide.

AddIns

This directory contains the configuration information required by Rose RealTime add-ins.

Note: For the latest integration information and line-up details please consult the Release Notes which accompanied the product.



Chapter 6

Starting Rational Rose RealTime

Starting Rose RealTime under WindowsNT:

To start Rose RealTime on WindowsNT, open the Start menu, and under the Programs folder, select Rational Rose RealTime 6.0. You must first install license keys before running Rose RealTime 6.01.

Temporary license keys can be found in the product package. Installation details can be found in the Installation Guide which accompanied the product.

Starting Rose RealTime under UNIX:

Before starting Rose RealTime on UNIX make sure your environment is set up correctly, according to the instructions provided earlier in this guide and in the *Install Guide*. If not, your path may not be defined properly to find the various programs required to run Rational Rose RealTime.

Rose RealTime is started from a UNIX command shell by typing:

```
% rosert
```

Start-up options for UNIX:

-regedit

To edit the internal registry that maintains mappings of directory names and other information required by the Rose RealTime tool. This registry mimicks the function of the WindowsNT registry, except that on UNIX the registry is maintained directly by Rose RealTime.

-startuplicense

Create a startup license.

-license

Obtain a permanent or temporary license.

-recreate_registry

To create a default registry, throwing away any changes made through the `-regedit` option.

-q | -quiet

Limit the output of the tool on startup.

-v | -verbose

Provide verbose output on startup.

-cleanup

Kills all applications running using Mainwin and then cleans up the x-server resources.

This is especially useful for the Visual Differencing tool. You should be very careful with this command as it will kill all mainwin applications running under your id.

Running Rose RealTime in batch mode

Rose RealTime can be programmed to automatically perform a wide variety of tasks. This programming is done through the Rose RealTime Extensibility Interface (RRTEI). The RRTEI provides a scripting language for interacting with the tool. This interface can be used to create Add-ins and developer scripts. Rose RealTime also supports the Rose Extensibility Interface (REI) for backwards compatibility. The complete documentation for the RRTEI is included in the Rose RealTime Online Help System. See the Extensibility Interface folder.

Running Rose RealTime in batch mode requires a license.



Chapter 7

Troubleshooting

CD read errors

If you are installing from a CD-ROM drive across the network and you are using a fast CD-ROM device, you may see some tar read errors during the installation process. To avoid the problem, either copy the CD contents to a local disk drive and run the installation from there, or run `setup.sh` from the machine to which the CD-ROM drive is connected.

Install/Uninstall problems

Install will not proceed for non-Administrators.

The user doing the install must be in the Administrator group to run Rational Rose RealTime install. There is a concept of Administrator privileges on the System in NT, rather than network administrator. To add the user administrator privileges, you have to login as an administrator for the system (not a network administrator) and run "User Manager" utility in Start Menu\Programs\Administrative Tools(Common). Select Administrators group and add the user to the group. Refer to Windows NT documentation for further details.

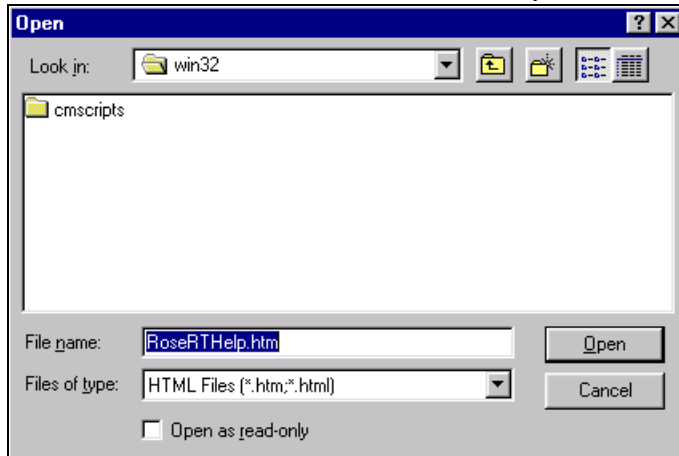
Uninstall leaves incrementally installed Rose RealTime files on the disk.

Always run the uninstall program before re-installing Rose RealTime. If components have been incrementally installed they will not be removed by the uninstall but must be removed manually.

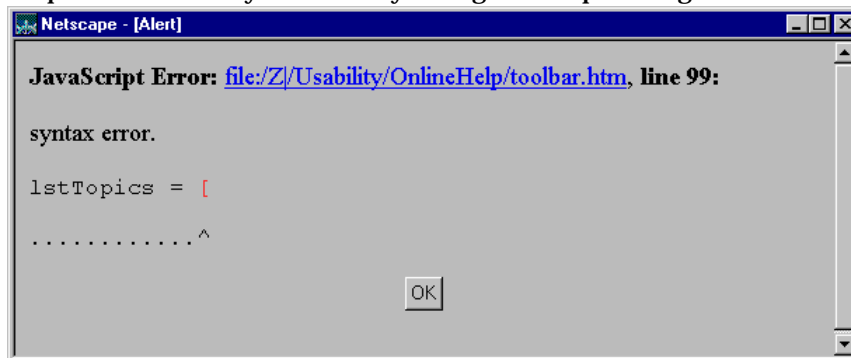
Online help system

If a default browser has not been configured, or the help files cannot be found, a dialog will appear when Help is requested from within Rose RealTime. Ensure that the \$ROSERT_HOME environment variable is properly set.

You can find this address under Tools > Options > General tab.



If a version of Netscape Navigator 4.03 and below is used with the help system, you may encounter several Javascript errors when loading the help. Ensure that you are only using Netscape Navigator 4.51 or up.

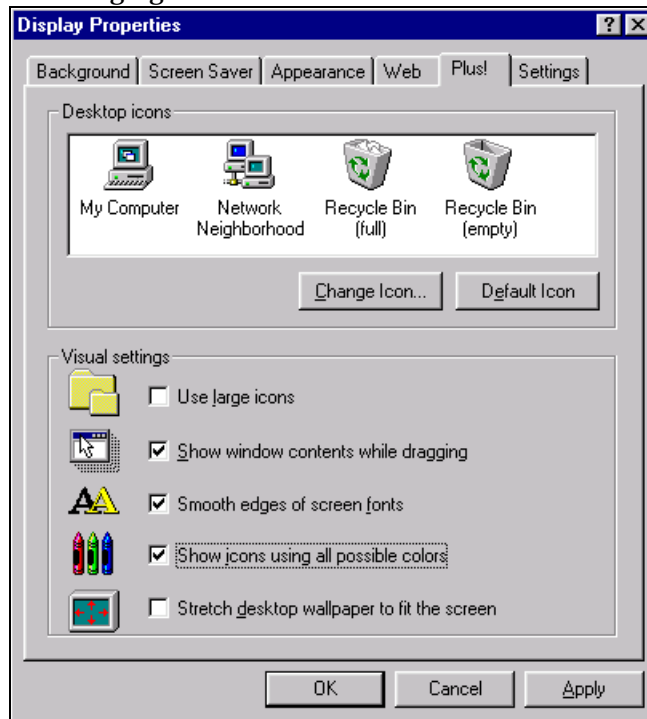


Mailing exception files

Windows Messaging must be installed before either exception files or comment files are automatically mailed. The format supported is SMTP (internet format addresses).

Icon display

If the Rose Realtime icon which is displayed under Windows NT does not match the documentation, you may need to increase the number of colors which Windows NT uses to display the program icons. In order to correct, you will need to bring up Display Properties from the Desktop of your NT workstation and check the box as displayed in the following figure.



CM Library problems

If problems are encountered when enabling source control for a model (see the *User's Guide* for instructions on integrating with a CM system) you will get the following dialog:



Possible cause:

Rose RealTime scripts directory does not contain valid CM scripts.

Solution:

In order for Rose RealTime to recognize a directory as a valid library to be accessed with the script interface, the directory must contain valid interface scripts. By default interface scripts for Visual SourceSafe, ClearCase, RCS, and SCCS, are located in $\$(ROSE_HOME)/\$(ROSE_HOST)/cmscripts$. Open the Source Control tab on the model specification dialog, and chose an appropriate directory.

Possible cause:

CM system binaries are not on the path.

Solution:

Ensure that the CM systems binaries directory is added to the path and then restart Rose RealTime.

Possible cause:

You are not configured as a CM system user.

Solution:

For some CM systems (e.g., Visual SourceSafe) each user has to be configured, before using the CM system ensure that you have an account setup within the CM system. For example with Visual SourceSafe you can setup accounts through SourceSafe Administrator. Request from your CM system administrator add you as a user and then restart Rose RealTime.

Compilation problems

Compile fails on valid C++ models with VC++ 5.0 or VC++ 6.0

The \$INCLUDE and \$LIB environment variables may not be properly set Ensure that your compiler binaries are on the path and that the \$INCLUDE and \$LIB environment variables are set (for example, they could be set for the user who installed VC++, but not set for another user). Set the environment variables. Refer to the VC++ documentation for further details.

Error loading Capsule (“could not spawn process”)

If the executable (capsule1.exe) is stored on an NFS server then the NFS client must be configured to have execute permission set.

Error linking Capsule (“error from nmake”)

If the executable (capsule1.exe) is stored on an NFS server then the NFS client must be configured to have execute permission set.

Windows NT Compilation Command Line Limits

If you encounter a compilation error message that complains about the command line being too long, the cause may be that the length of your compile or linker has exceeded a limit.

Windows NT compilation has command line limits in two areas: source compilation and linking. Both limits have been explored for the Visual C++ 5.0, VRTX PPC Microtec 1.4 and Tornado 1.0.1 PPC Cygnus 2.7.2 compilers.

Source File Compilation

The variables in source compilation are the update name, the \$ROSERT_HOME path, compilation options, the local working directory and include directories. The only compiler that has a measurable limit is VRTX. The command line limit is 768 characters.

A workaround for the problem is to reduce the number of include directories by combining include files. Other solutions are to shorten paths and names for the variables listed in the previous paragraph.

Linking

The variables in linking are the update name, the \$ROSERT_HOME path, the link options, the number and name length of libraries, the library search paths and the local working directory. The link limits are shown below:

- Visual C++ 5.0: more than 20875 characters
- VRTX PPC Microtec 1.4: 4147 characters
- Tornado 1.0.1 PPC Cygnus 2.7.2: 4150 characters
- HPUX 10.20: 16384 characters (make: couldn't load shell.stop)

A workaround for the problem is to shorten paths and names for the variables listed in the previous paragraph.

Known limitations and restrictions

For the current list of know limitations and restrictions please visit the Rational Rose RealTime home page at:

<http://www.rational.com/products/rosert>