Release Notes

Rational Rose

VERSION: 2002.05.20 FOR SOLARIS

NOTE: THESE RELEASE NOTES APPLY TO THE SOLARIS RELEASE. FIXES LISTED IN THIS RELEASE NOTE FOR VERSION 2002.05.01 AND 2002.05.20 APPLY ONLY TO ROSE UNIX ON SOLARIS. THE HP-UX 10.20 AND HP-UX 11.00 RELEASES INCLUDED IN THIS SUITE RELEASE REMAIN AS VERSION 2002.05.00 AND THEIR PREVIOUS RELEASE NOTES APPLY (INCLUDED, ONLINE IN THE RELEASE - IN THE "DOCS" DIRECTORY).

PART NUMBER: 800-025343-001

Rational®
the software development company

Contents

Preface

Rational Rose for UNIX/Linux® is a comprehensive, integrated programming environment that supports the development of complex software systems. These release notes contain information about last-minute changes to Rose and the Rose Add-Ins.

Scope

These Rational Rose release notes provide:

- System and software requirements
- New features and enhancements
- Any defects fixed since the last release
- Current known defects and defect workarounds

The electronic versions of the release notes that were packaged with your software contain the most up-to-date information available at release time.

Updated versions of the release notes will periodically be posted on the World Wide Web. Please check the Rational Web site at http://www.rational.com/support/documentation/release/ to ensure that you are using the latest version of this document.

When you are ready to install Rational Rose, be sure to check the Web for the latest available downloads:

1 From http://www.rational.com, click **Support**.

2 Go to the **Download Center** and navigate to **Rational Rose Upgrades and Patches.**

Also remember to have the *Administering Licenses* manual available to provide all the information you need to complete your installation.

Contacting Rational Technical Publications

To send feedback about documentation for Rational products, please send e-mail to techpubs@rational.com.

Contacting Rational Technical Support

Contact Rational Customer Support if you have questions about installing, using, or maintaining this Rational Software product. The following table describes how to contact Rational Customer Support.

Your Location	Telephone	Fascimile	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) 20-4546-202 Netherlands	+31 (0) 20-4545-201	support@europe.rational.com
Asia Pacific	+61-2-9419-0111 Australia	+61-2-9419-0123 Australia	support@apac.rational.com

For up-to-date contact numbers and addresses visit the support contacts web page at http://www.rational.com/support/contact.

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 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)

Contacting Rational Licensing Technical Support

If you have questions about acquiring license keys for your Rational Software products, contact Rational Licensing Support as follows:

Your Location	Telephone	Facsimile	E-mail
Asia Pacific	+61-2-9419-0111	+61-2-9419-0123	license@apac.rational.com
Asia Pacific (Japan)	+61-2-9419-0111	+61-2-9419-0123	license@japan.rational.com
Asia Pacific (Korea)	+82-2-556-9420	+82-2-556-9426	license@apac.rational.com
Asia Pacific (Mainland China, Hong Kong, and Taiwan)	+61-2-9419-0111	+61-2-9419-0123	license@china.rational.com
Europe Middle East Africa	+31 20 4546 200	+31 20 4546 202	license@europe.rational.com
North American (East Coast) South America	800-433-5444 (toll free)	781-676-2460	license@rational.com
North America (West Coast) Central America	800-433-5444 (toll free)	408-863-4001	license@rational.com

Introduction

1

Thank you for selecting Rational Rose for UNIX/Linux. The release notes provide last-minute changes and reference information about Rational Rose and all of the Rational Rose Add-Ins. The release notes contain a listing of enhancements, known problems, and other information pertaining to the operation and functionality of Rational Rose.

Updates for Registered Owners

From time to time, service packs and additional utilities may be made available for download to registered owners. Please periodically check the Rational Rose web page at www.rational.com/rose/.

Rational Rose for UNIX/Linux Documentation Set

The complete documentation set is supplied in electronic format with this release.

For a complete printed documentation set, contact your Rational Sales office.

Installation and Licensing 2

Please refer to the *Rose for UNIX/Linux Installation Guide* that is provided as your primary guide for installing and setting up licensing for this product.

If you encounter any problems while installing or licensing the Rose product, please check here first to see if it is a known problem. If a problem is not yet documented, please contact Rational Technical Support so we can investigate it, provide you with a workaround, and track the problem for future action.

This chapter contains the following sections:

General Install Information

As a precaution, always make back ups of any Rose model files that you will be accessing with this software.

This product includes FLEXlm, a licensing product from Globetrotter, Inc. A startup license key that allows you to start using your product immediately is shipped with this product. You must obtain a separate permanent license from Rational in order to continue using this product beyond this startup key expiration date. Please refer to

the licensing section of the *Installation Guide* for detailed instructions on obtaining and installing permanent licenses. Also, see *Licensing Information* on page 19 for any last minute licensing notes.

Note: Rose 98i, 2000e, 2001,2001a, and 2002 for UNIX licenses are valid for this release. You will not need to utilize the startup licenses or obtain new permanent licenses if you are already utilizing Rose 98i, 2000e, 2001, or 2001a for UNIX licenses.

Pre-Installation Information

You may want to back up your Registry prior to running the installation program. Specifically, we recommend you make backups of the following files:

- You should always have a backup of all your model files. These can include files that end with .mdl,.cat,.ptl, .red, and .sub.

- Custom property files, which can include files that end with .pty and .prp

  Note: There is no need to save .pty files that ship with Rose since these will be re-installed.

- Custom script files, which can include files that end with .ebs and .ebx

  Note: There is no need to save .ebs or .ebx files that ship with Rose, since these will be re-installed.

- Path Maps (For Rose 98i, 2000e, 2001, and 2001a path map variables are saved in the registry.)

Follow these steps to back up path maps:

1 Run **regedit** and navigate to **HKEY_CURRENT_USER\Software\Rational Software\Rose\Virtual Path Map**

2 Click **Registry > Export Registry File...**

Follow these steps to restore path maps:

1 Run **regedit** and navigate to **HKEY_CURRENT_USER\Software\Rational Software\Rose\Virtual Path Map**

2 Click **Registry > Import Registry File...** and import the file you exported.

System Requirements

Before installing this software, please verify that your system meets the minimum system requirements listed below. If the requirements are not met, the product may not install and/or operate correctly.

Note: All platforms may not be available at this time. Contact Rational technical support for platform availability information.

The following table contains the hardware requirements for installation and use of this product. Note that the recommended amount of memory is only the suggested minimum amount. Additional memory may improve performance. Your memory requirements will also be larger if you are constructing large models.

Table 1   Hardware Requirements

Hardware Item	Recommendation	For
Memory	256 + (32 * N) MB of RAM (where N is the number of users running Rose simultaneously on a given workstation)	Use
Server	A midrange UNIX/Linux server or client UNIX/Linux workstation (Sun servers: Sparc Station 20 and above)	Use
Client	An AIX, Solaris, HP-UX, SGI IRIX, or Tru64 Unix workstation running the required operating system level as described in the following table under operating system software requirements. The UNIX/Linux workstation must be capable of displaying X Windows or a PC running eXceed version 6.1or higher to display X windows or for Rose on Linux x86, a PC running with Xfree and Xi Graphics X servers on Linux. It is recommended to use more than 256 colors in your graphics card/monitor configuration.	Use
Disk space	400 MB for loading release + 1-3 MB for each Rose model	Installation, Use
CD-ROM	Mounted as a UNIX/Linux file system	Installation
Display	Color display	Use

The next table contains the software requirements for installation and use of this product. Note that this is also only a suggested amount of swap space. Your requirements may be larger if you expect to construct very large models. If all of your models are small, you may need less than the amount shown.

Table 2   Software Requirements

Software Item	Requirement	For
Operating system[a]	AIX 4.3.2 Solaris 2.5.1, 2.6, 7, 8 (kernel revision January 2000 and later) HP-UX[b] 10.20 (ACE release[c]), 11.00, and 11.11 SGI IRIX 6.5.5 Tru64 Unix 4.0f Linux x86 Red Hat 6.2, 7.0 and 7.1	Use

C++ Compiler #include files	AIX CSet ++ 3.6.6
	Solaris SC 5.0
	HP-UX 10.20:
	HP aC++ B3910B A.01.21
	HP aC++ B3910B A.01.19.02 Language Support Library
	
	HP-UX 11.00:
	  HP aC++B3910B A.03.13
	  HP aC++B3910B X.03.11.10 Language Support Library
	
	IRIX C++ 7.3.1.1m
	Tru64 for Unix C++ 6.2
	Linux x86 g++ 2.95.2
		Analyzer

JDK, JFC, and J2EE	JDK 1.1.6 JFC 1.1 (swing classes for 1.1) JDK 1.2 (based on 1.2.1) J2SE 1.2 (based on 1.2.2) J2SE 1.3 J2EE 1.2 or any of the major Java IDEs	

Table 2   Software Requirements

Software Item	Requirement	For
Java	Rose J supports the following Java IDE's. Use the JDK appropriate to your IDE.[d] - Forte for Java Internet Edition 2.0 & 3.0(Solaris & Linux only) - Forte for Java Community Edition 2.0 & 3.0(Solaris & Linux only) - JBuilder 4.0, 5.0, and 6.0 Professional Edition (Solaris & Linux only) - JBuilder 4.0, 5.0, and 6.0 Foundation Edition (Solaris & Linux only)	
Port map daemon	Must be running	Installation, Use
TCP/IP	Must be running	Installation, Use
Host names	Must be configured	Installation, Use
Page/Swap space	Generally three times the space in swap as RAM available	Use
X Window System	Native X11R5 or X11R6 (except HP 10.20) X news on Sun with OpenWindows 3 PC X server eXceed version 6.1 or higher In addition, when running Rose for Linux x86, XFree and Xi Graphics X servers on Linux	Use

Table 2   Software Requirements

Software Item	Requirement	For
Window Manager	MWM (Motif Window Manager) version 1.2	Use
	vuewm on HP-UX	
	4dwm on SGI IRIX	
	dtwm (CDE (Desk Top) version 1.0)	
	olwm (Open Look Window Manager) on SUN is supported, but icons of Rose for UNIX/Linux may not integrate well with other icons on the screen	
	In addition to those listed above, Rose for Linux x86 will run with the GNOME desktop environment with the Enlightenment window manager and the KDE desktop environment with the lcwm window manager.	
	The following Window Managers are not supported:	
	olvwm (olwm with virtual screens) Rose for UNIX/Linux will not remain on original virtual desktop, but rather will follow when other desktops selected.	
	twm (shipped with X11)	
	tvtwm (twm with virtual screens)	
	fwm	
	gwm	
	fvwm	

a. All platforms may not be available at this time. Contact technical support for availability information.
b. HP9000 Series 700.
c. Rose requires that an ACE release be installed to run properly on HP-UX 10.20. To determine if an ACE release is installed, run **/usr/sbin/swlist | grep ACE**.
d. See the Rose_IDE_Link_README file in the release for Rose UNIX/Linux IDE Link Integration setup information, prior to using the IDE Link integration. The Rose_IDE_Link_README file will also list the supported / tested Java versions to utilize with IDE Links and J Pattern features.

Hardware/Software Patch Requirements

Please see "Operating System Runtime and/or Patch Requirements" in the *Installation Guide* for system patch requirements.

When installing and running Rose for UNIX/Linux for the first time, you will be alerted if hardware or software patches are necessary. If running on a Solaris or HP-UX server, Rose for UNIX/Linux will attempt to automatically determine if patches are necessary on your server and/or Solaris display, and will advise accordingly.

You may also run the following anytime to determine if your computer requires operating system patches to run Rose for UNIX/Linux.

- SUN Solaris Server (computer running Rose for UNIX/Linux)

  cd *rational_dir*/**releases/rose.***I.J.K*/**bin**

  ./check_rose_reqs sol2_server rose

- SUN Solaris Display (computer displaying Rose for UNIX/Linux)

  Check that the DISPLAY environment variable is set to your display, and then:

  cd *rational_dir*/**releases/rose.***I.J.K*/**bin**

  ./check_rose_reqs sol2_client rose

- HP-UX, IBM, or SGI Server

  cd *rational_dir*/**releases/rose.***I.J.K*/**bin**

  for HP-UX - **./check_rose_reqs hp_runtime rose**

  for IBM - **./check_rose_reqs aix_runtime rose**

  for SGI - **./check_rose_reqs sgi_runtime rose**

  for Tru64 Unix - **./check_rose_reqs axp_runtime rose**

Running and Displaying on SGI IRIX - Hang Can Occur

This issue typically occurs when running on an SGI IRIX multi-processor machine and displaying to any SGI machine. If you encounter a Rose Hang, under these circumstances, install the patched "libX11.so" library in your Rose release as described below.

If you encounter any problems, after installing this patch contact Rational Technical Support.

1 Make sure your Rose process is no longer running. (Us**e ps** to determine if the process is running, and "kill -TERM [pid]" if it is necessary to kill the process.)

2 As the owner of these files, **cd** *rational_dir*/**releases/rose.I.J.K/install/patches/mips_irix_n32**

3 **cp libX11.so** *rational_di***r/releases/rose.I.J.K/mips_irix_n32/lib/.**

4 Run **rose_cleanup.**

5 Run **rose**.

Configuring Window Manager So Pop-Up Windows Will Remain On Top of their Parent Windows

Configure your Window Manager such that pop-up windows will remain on top of their parent windows, in compliance with the Windows API. Under X11, Window Managers may implement different strategies for maintaining the Z-order of pop-up windows.

In order to keep pop-ups on top, if possible:

- For MWM - Motif Window Manager

There is no specific configuration to keep pop-up (transient) windows on top. Pop-up windows are kept on top by default.

- For DTWM - Desk Top Window Manager (shipped with CDE)

Depending on your system's configuration, the default behavior of dtwm may not keep pop-up (transient) windows on top of their owner.

For example, it may be that the window stacking behavior for dialogs is specific, such as "Dtwm*secondariesOnTop = False". In such a case, here are some possible solutions for keeping the pop-ups on top:

  a Change Dtwm*secondariesOnTop to True in the system-wide dtwm configuration file. On Solaris, the dtwm configuration file is located at **/usr/dt/app-defaults/C/Dtwm**. On platforms other than Solaris, the path to the dtwm configuration file may be different.

  b Add

```
Dtwm*secondariesOnTop: True
```

    to your **.Xdefaults** file.

  c Be more specific with the program name, and add, for example:

```
Dtwm*ssexp*secondariesOnTop: True
```

    to your **.Xdefaults** file.

    Note: The **.Xdefaults** file needs to be reloaded with:

    xrdb ~/.Xdefaults

- For OLWM -- Open Look Window Manager

Add the following line to your **.Xdefaults** file:

```
OpenWindows.KeepTransientsAbove: True
```

Running Multi-threaded GUI Applications on Multi-processor Solaris 2.6 Computers

If Rose for UNIX/Linux hangs when running on a multi-processor Solaris 2.6 computer, verify that the Solaris 2.6 server and display (if applicable) includes the Sun Solaris XSun Patch 105633-32 or higher. This patch includes a fix for Sun Defect 4233280, Sun SO#3918581 (multi-threaded applications crash/hang on 2.6 multiprocessor computers). To check for this patch, run

showrev -p | grep 105633

on both the server running Rose and the display (if display Solaris 2.6).

When installing or running the application for the first time, Rose will advise if this patch is installed or required.

Displaying on Ultrabook, Using PGX Graphics Card

A CDE crash may occur when displaying Rose for UNIX/Linux on a Ultrabook using a PGX Graphics Card, if SUN Solaris patch 105362-26 is installed.

If using a PGX Graphics Card, we recommend that SUN Solaris patch 105362-20 be installed as this is the latest version tested and determined to be compatible with this release.

If you encounter the CDE crash, uninstall patch 105362-26 and if possible install "105362-20". If SUN Solaris patch "105362-20" is not installed, the following may occur:

- GUI Hanging Issues
- Paint Problems (i.e. black/white regions may appear in menus, dialog boxes, toolbars, etc.)

If you uninstall patch 105362-26 and encounter any of the issues described above, contact Rational Technical Support if patch 105362-20 is not available.

Running Rose for UNIX/Linux on HP-UX 11.00 with HP-UX OS Patch PHSS_21493

Rose for UNIX/Linux requires HP-UX OS Patch PHSS_17327 (X/Motif 2.1 Dev Kit Mar99 Periodic Patch). This Patch requires PHSS_17326 (X/Motif2.1 Runtime Mar99 Periodic Patch). PHSS_17326 has been superseded by HP-UX OS Patch PHSS_23823.

It is important to use PHSS_23823 as Rose for UNIX/Linux will not start (display) if an earlier version of this patch is installed (i.e. PHSS_21493).

If it is necessary to have a Motif Xlib runtime patch prior to PHSS_23823 and you encounter this issue, refer to the following instructions to workaround this problem:

1   Make sure your Rose process is no longer running. (Use **ps** to determine if the process is running, and "kill -TERM [pid]" if it is necessary to kill the process.)

2   As the owner of these files,

    cd *rational_dir***/releases/rose.I.J.K/install/patches/hppa_hpux/hp11**

3   **cp libX11.3** *rational_dir***/releases/rose.I.J.K/hppa_hpux/hp11/lib/.**

4   **chmod 555 rational_dir/releases/rose.I.J.K/hppa_hpux/hp11/lib/libX11.3**

5   Run **rose_cleanup**.

6   Run **rose**.

MainWin Runtime Components

When Rose for UNIX/Linux runs for the first time, an **$MWUSER_DIRECTORY** directory is created which includes MainWin files and directories. Rose is created utilizing MainWin (developed by Mainsoft) and requires specific MainWin components to operate correctly. **$MWUSER_DIRECTORY** may be set to serve as the windows directory. For example, **$MWUSER_DIRECTORY** may be set to **$HOME/.windows**. If **$MWUSER_DIRECTORY** is not set, the default location will be **$HOME/windows**.

Within the **$MWUSER_DIRECTORY** directory, you will find the following type of files and directories:

- **mwfc**-* - Font caches, are created incrementally over the duration of the application process; activating when a user chooses a new font. The font cache is built each time Rose for UNIX/Linux is run on a new X server, or if the server's font path is changed. Each font cache file has a unique name:

  mwfc-*xxxx* (where *xxxx* is dependent on the specific display used)

  This file can be removed if it is suspected of interfering with font operations. If removed, it will be regenerated the next time Rose is invoked.

  When Rose for UNIX/Linux is invoked by the user on the same X server, the application can quickly look up the font cache file and select the closest X font available.

- **registry** directory - This directory includes a binary "registry" file which is created the first time the user runs Rose for UNIX/Linux (for each version of the Operating System). For example, a "registry.5.6" registry is created for a user running the release on a SUN Solaris 2.6 computer. The registry stores Rose for UNIX/Linux configuration and initialization data.

  The registry directory also includes **rose_reg_check**, an ascii file which includes the version of Rose for UNIX/Linux and it's associated add-ins. This file is created the first time a user runs Rose for UNIX/Linux.

  A specific registry file may be utilized by setting the following environment variable:

  setenv USE_MWREGISTRY *path/myregistry*

- **help** - This directory includes files that get created and stored every time the user runs help and creates the help keyword database.

- **win.ini** file - This is a default configuration file. It includes such information as print device/port settings.

  During a Rose session, Rose for UNIX/Linux *.**ini** files may also be created in the user's "windows" directory. These files include Add-In specific settings.

Rational Rose for UNIX/Linux includes "mw" directory

Rational uses MainWin (by Mainsoft) to rehost Rose for UNIX/Linux platforms. Specific MainWin runtime components are included in Rose. Below is a description of the type of MainWin files included in Rose for UNIX/Linux under the *rational_dir*/**rel***/**rose***/**mw** directory.

- **afm** directory - Includes support fonts.

- **bin** directory - Includes programs which perform such functions as determining hardware configuration, removing unreleased X resources, showing X resources, and converting old registries to new.

- **bin**-<**os**>_**optimized** directory - Includes binaries designed to register libraries and controls, provides registry browser and support, and type library browser, and font server.

- **fonts** directory - Includes release fonts and font files compatible on both PC and UNIX/Linux displays. See *Installing and Utilizing Fonts Compatible on Both PC and UNIX/Linux Displays* on page 14.

- **fontserver.cfg** file - Font configuration file used by font server.

- **helpfile** directory - Includes help files provide by Mainsoft.

- **lib** directory - Includes scripts to order library paths correctly, and to setup MainWin configuration.

- **lib**-**<arch>_<os>** directory - Includes an nls directory which holds *.**nls** (Nation Language Support) files that are used upon application startup. It also includes **mwperl** (an executable called upon by the MainWin scripts).

- **lib**-**<os>** directory - Includes files to manage the font cache, provide support for type library browser and registry browser.

- **lib**-**<os>_optimized** directory - Includes MainWin support libraries and resources.

- **system** directory - Includes default binary registry and system MainWin registry files.

- **setmw*** and **setup*** files - These files provide appropriate MainWin environment settings.

- **win.ini** is the default Windows configuration file (which gets copied in **$MWUSER_DIRECTORY** the first time Rose is run).

Installing and Utilizing Fonts Compatible on Both PC and UNIX/Linux Displays

Rose for UNIX/Linux now includes the following release fonts which are compatible and may be used on both the PC and UNIX/Linux displays:

- MS Sans Serif

- Arial

- Times New Roman

- Courier New

- Tahoma

As it now is possible to display these same fonts on UNIX/Linux and the PC - models, etc. created on the PC (using these fonts) should display easily and accurately in Rose for UNIX/Linux on PC and UNIX/Linux displays. Furthermore, utilization of these fonts will provide the following additional enhancements:

- Improved dialog scaling on all platforms

- Resolves defects when these PC fonts were not always available on UNIX/Linux displays. Previously, a model created on the PC would utilize an Arial font (not available on UNIX/Linux). Opening the same model on a UNIX/Linux display could produce unpredictable results when the font would map to "Application".

- Resolves defects when users were creating/sharing models to be displayed on PC and UNIX/Linux displays. Display difficulties could occur when models created or edited on PC display (i.e. using eXceed) were then opened on a UNIX/Linux display.

Font files (in *.pcf - portable compiled format), are now included in the *rational_dir*/**releases/rose.***I.J.K***/mw/fonts** directory of your release. This directory also includes the following fonts* files:

- **fonts.dir** - The X11 fonts dir file.

- **fonts.mwd** - MainWin fonts dir file. This file is accessed by this release and contains, for each font, the WIN32 metrics, the X11 fontname, a mask stating whether it is a "stock" font, and a "used on unix" flag.

A **fontserver.cfg** file is included in the *rational_dir*/**releases/rose.***I.J.K***/mw** directory. This file is used when starting the font server to provide the new fonts to the Rose application. A font server (xfs) is also provided in the *rational_dir*/**releases/rose.***I.J.K***/mw/bin_*<os>*_optimized** directory.

Making The Release Fonts Available on Your $DISPLAY

Prior to running Rose for UNIX/Linux for the first time, run *rational_dir*/**releases/rose.***I.J.K***/bin/rose_cleanup**. Upon starting Rose for UNIX/Linux, you will notice some new messages in relation to the availability of the new fonts. Rose will attempt to determine if the new fonts are available for your $DISPLAY, by running the new Rose for UNIX/Linux **rose_fonts** utility.

UNIX/Linux DISPLAY with visibility to Rose fonts

rose_fonts will make the fonts available to your DISPLAY by including the path to *rational_dir*/**releases/rose.***I.J.K***/mw/fonts/** in your DISPLAY's font path.

DISPLAY other than UNIX/Linux (i.e. PC using eXceed) or UNIX/Linux DISPLAY without visibility to Rose fonts

In this case, fonts must be made available by using a font server and including the font server in the DISPLAY's font path.

rose_fonts will determine if a font server is providing access to the Rose fonts, as described below:

- Determines if a font server is already running (on the server used to run Rose). There may already be a font server running on the designated font server's port# 7100. On SUN systems 2.5.1 and above it is likely that a font server is already

running and defined in the **/etc/services** file, as "xfs" or "fs". The existing font server's default **fontserver.cfg** file will not include a path to the Rose fonts (unless it has been configured to do so).

- If a font server is running, **rose_fonts** determines whether the existing font server is already providing access to the **rose_fonts** (i.e. it determines if the fontserver.cfg includes the path to the Rose fonts provided in the release).

- If a font server is not running, **rose_fonts** attempts to start a font server "xfs".

- If a font server is running and does provide access to the Rose fonts, **rose_fonts** adds the font server to the DISPLAY's font path.

If Necessary To Configure Existing Font Server

If a font server is running on the Rose server and it is not already providing access to the Rose fonts, contact your System Administrator to perform this setup or refer to the man pages for xfs(1). The following advises how to configure an existing font server on SUN Solaris.

Note: Configuring an existing font server to provide access to the Rose fonts may produce less than optimal font results, based on the font paths included in the fontserver.cfg's catalogue setting. This is due to a potential Rose 2000e for UNIX/Linux font mapping issue that is currently under investigation. If you elect to continue with the following instructions, and you encounter poor font results please notify Rational Technical Support and remove the Rose font support from your fontserver.cfg file.

If you elect not to add the Rose font support (as described next) or elect to remove the Rose font support from an existing font server Rose will continue to operate correctly, however the font enhancements listed in *Installing and Utilizing Fonts Compatible on Both PC and UNIX/Linux Displays* on page 14 will not be available and you may encounter the following defects:

Defect 10386
Defect 10387

Adding Rose fonts to Existing Font Server on SUN Solaris

To add Rose font support to an existing font server on SUN (it may be necessary to su to root to perform these actions):

- Edit the default fontserver.cfg file "/usr/openwin/lib/X11/fontserver.cfg" to add the location of the Rose fonts in the "catalogue" path (being careful not to add or remove extra spaces).

A catalogue path may appear as:

/usr/openwin/lib/X11/fonts/F3bitmaps/,/usr/openwin/lib/X11/fonts/Type1/

Add the Rose font path, by prepending the path to the catalogue path.

For example:

rational_dir/releases/rose.*I.J.K*/ mw/fonts/,/usr/openwin/lib/X11/fonts/F3bitmaps/,...

Do not include spaces. Each path is separated by a colon.

- Use **ps** to determine the process ID of the "fs" or "xfs" process, and then stop (kill) the "fs" or "xfs" process, if running.

- Set the DISPLAY font path to include the path to the existing font server by running:

  xset +fp tcp/<*fontserver*>**:7100**

  and then

  xset fp rehash

- Determine if the Rose fonts are available and restart font server:

  fslsfonts -**server** <*fontserver*>**:7100** | **grep** -**i mainwin**

  The font server must be the current host for it to restart the font server ("fs" or "xfs"). In many cases these x* utilities are located in /usr/openwin/bin.

Rose (upon starting) should indicate that the Rose fonts are now available.

Upon performing this setup, it will be important to update the **fontserver.cfg** file if Rose is uninstalled or moved to a different location (path changes).

Starting a Font Server

If a font server is not running (on the server running Rose), **rose_fonts** will start a font server provided in the release, if necessary. This is necessary if the DISPLAY is not a UNIX/Linux display, or if the display does not have direct visibility to the fonts provided in the Rose release.

X Font Commands Utilized by rose_fonts

The following commands are typically located in one of the following directories: /usr/openwin/bin, /usr/bin/X11, /usr/X11R6/bin, /usr/openwin/lib/X11, or /usr/openwin/include/X11. The font server "xfs" is provided in your release in *rational_dir*/releases/rose.*I.J.K*/mw/bin-*$arch*_optimized.

To Start a Font Server

xfs -**config** *rational_dir*/**releases/rose.***I.J.K***/mw/fontserver.cfg** -**port 7100**

To Set the Font Path

xset +fp *rational_dir***/releases/rose.***I.J.K***/mw/fonts/** (direct path to fonts in release)

xset +fp tcp/$host:7100 (font server path)

To Determine if Rose Fonts Available

xlsfonts | grep -i mainwin

xlsfonts may hang if a font path includes a path to a font server which is not running. **rose_fonts** will not let **xlsfonts** run for longer than a specified period of time (typically 30 seconds), to safeguard against this event. If an **xlsfonts** hang occurs, **rose_fonts** will provide status with recommendations.

rose_fonts Overview

In summary, **rose_fonts** will provide the Rose fonts to your DISPLAY by either setting the font path to point to the fonts in the release (UNIX/Linux display with visibility to the fonts), or by setting the font path to a running font server if the font server is providing access to the Rose fonts. **rose_fonts** will use "xset" to set the font path, "xfs" to start a font server (if necessary) and "xlsfonts" to determine if the Rose fonts are available after the setup has been performed. If a font server is already running and does not already provide support for the Rose fonts, manual setup is necessary (i.e. see *Adding Rose fonts to Existing Font Server on SUN Solaris* on page 16) to complete the Rose font setup. In this case, manual set up is required as it is often necessary to su to root to perform this action. **rose_fonts** will provide messages describing the current status each time Rose is started.

Contact Rational Technical Support if you should have any questions or encounter any difficulties accessing or utilizing the new Rose fonts.

If Rose Fonts Not Available

There may be occasions when the Rose fonts will not be available. **rose_fonts** will alert you if, when starting Rose, if the Rose fonts are not available. If this occurs, Rose will continue operate correctly, however the font enhancements listed in *Installing and Utilizing Fonts Compatible on Both PC and UNIX/Linux Displays* on page 14 will not be available and you may encounter the following defects:

Defect 10386

Defect 10387

To Eliminate Output from "rose_fonts" When Running Rose

Each time that you start Rose, **rose_fonts** determines whether the Rose Fonts are available, attempting to make the fonts available when necessary. Each time Rose is started, **rose_fonts** displays messages advising current Rose Font status. In order to eliminate output from **rose_fonts**, set the following environment variable prior to running Rose.

setenv ROSE_NO_FONT_MSG 1

Setting this environment variable will allow **rose_fonts** to run, but will refrain from printing Rose Font status messages.

Licensing Information

There are three license types supported by Rose for UNIX/Linux:

- Startup - This is a time-locked temporary license to allow you to use the software until you receive a production key.

- Node-locked - This allows you to run the software on a single node (computer).

- Floating - This is server-based licensing, where a client logs onto a license server to be assigned a key.

For a full description of license types, installation and usage, please consult the *Rational Rose for UNIX/Linux Installation Guide.*

The software you receive includes a temporary startup license. You must install the software and follow the steps detailed in the *Installation Guide* to acquire node-locked or floating license for your product. For users who are upgrading from Rose **98i** for UNIX/Linux or after, you may use your existing license key to run Rose for UNIX/Linux.

Rational Rose for UNIX/Linux

<div style="text-align: right; font-size: 3em;">3</div>

Rational Rose, the world's leading visual modeling tool, allows you to define and communicate a software architecture, resulting in accelerated development, improved quality, and increased visibility and predictability.

The Rational Rose portion of the release notes describes new features and enhancements, known problems and limitations, and fixed defects since the last release. You will also find the Rose Extensibility Interface information in this section.

Note: Fixes listed in this Release Note for version 2002.05.01 and 2002.05.20 apply only to Rose UNIX on Solaris. If you require the fixes made for Solaris on another architecture (i.e. HP-UX 10.20, HP-UX 11.00 or Linux x86 (Red Hat)), please contact Rational Technical Support.

New Rational Rose Features and Enhancements

This release of Rational Rose includes these new features and enhancements:

- Rose 2002.05.20 creates models in a new petal file format for Rose UNIX on Solaris due to the inclusion of a new 'quidu' element designed to enhance performance. See Defect #79513, under Fixed Defects in Core Rose and Extensibility.

Rose.ini file replaced by rose.reg (Rose 2001a and after)

Any settings previously in the Rose.ini file are now in the registry file, rose.reg. This new registry file is located in the rational_dir/releases/rose.I.J.K/registry directory of the release. The settings in this file are entered in your registry, after running "rose_cleanup" and then "rose". Any changes made to the release rose.reg file will apply to all users running rose. If you wish to make changes specific for your environment, a rose.reg.template file is available in rational_dir/releases/rose.I.J.K/registry. Copy this file to your $HOME directory and name it $USER.reg. Edit this file with the settings you prefer, run rose_cleanup and then rose. Rose will automatically register your $HOME/$USER.reg file and your settings will be available in your Rose session.

Known Core Rose and Extensibility Problems and Limitations

The following table contains known Rose problems and limitations.

Defect	Description
1719 16674	F1 Help for keywords does not work in the Script Editor. Use the online Help table of contents or index to find the help you require.
1859	Fonts change when sharing units with team members using different default fonts. Default font settings are currently stored with the model, and not with the .cat files. Items will only store font settings in a .cat file if the font is different from the default. As a result, text for model elements in units will be formatted, by default, according to the settings of the model. If one team member formats diagrams relying on a specific default color and font, that formatting will not be rendered for other team members with different default settings. Workaround: Explicitly format text using a font that is different than the default setting.
1977	A business actor loses (changes) its fill color if it is moved around in the browser. For example, change the fill color of the business actor from yellow to green and then move the business actor to a different package.
1981	**Report > Show Usage** does not work correctly.
2090	**Print Preview** does not work reliably for all diagram types.
2101	Print Specifications omits classes from the output report, without apparent reason or pattern.
2242	Drag and Drop of multiple selection from browser not supported. The browser can be used to select multiple items, and the context menu can be used for that selection, but drag and drop of multiple items from the browser to a diagram is not supported.
2746	Rose hangs when checking out a controlled unit named with a special character.
3057	If you change the font size (through **Tools > Options > Diagram** tab) of a model element that is already on a diagram, the model element may not correctly resize. This happens frequently on activity and sequence diagrams. Workaround - You can clean up the diagram using the **Edit > Refresh** command.
3078	Rose is inconsistent in how it handles user attempts to overload element names in different contexts.

3093	It is not possible to drag the "History" icon from the browser to the diagram.
3728	CDE window menu options do not appear when clicking in the top left hand corner menu of Rose. Workaround: Minimize Rose and right click on the icon. CDE options, such as "Occupy Workspace" will then appear.
3861	Scroll bar in Language Properties dialog box may disappear. Workaround: If this occurs, clicking in the scroll bar area will cause the scroll bar to reappear.
3877	Having a <return> character in the language properties **Tools > Options** "language tab" property fields can create a code generation problem. In addition, it is difficult to determine if a <return> character exists in language properties as it does not display.
6397	CDE crash and technicolor affect when running on RDI Ultrabook if PGX graphics patch 105362-20 or higher installed. (May occur with previous PGX patches as well.)
6924	REI: RoseModel.FindItems() does not find attributes or operations.
7156	Encapsulated postscript file created by Rose which includes either multiple diagrams or diagram which exceeds one page size may print on one page creating distorted images.
7871	Text within the brackets on messages are lost on all sequence diagrams when the Apply button is clicked after changing the Message Signature option.
8718	The Fit to Page print feature will occasionally not leave enough space at the bottom of the page for the last element on a diagram. This happens when the diagram, as normally displayed, is just a little too large to fit on one page.
8845	The Test Dialog option in the Dialog Editor is not supported in this version of Rose for UNIX/Linux. Selecting the F5 shortcut key or the Test Dialog Icon will have no affect.
9168	Some custom stereotype icons added-in to previous versions of Rational Rose do not display correctly in Rose for UNIX/Linux.
9229	To change the ROSE_CPP path map variable, you must first manually remove its entry from the registry. In the Windows Registry, this pathmap variable is located at two different locations: • HKEY_CURRENT_USER\Software\Rational Software\Rose\Virtual Path Map • HKEY_LOCAL_MACHINE\Software\Rational Software\Rose\Virtual Path Map

9535	Collaboration diagram numbering is thrown off by deleting and undeleting a link message. Workaround: Turn Collaboration Numbering off and then on again to reset the numbering. (Select **Tools > Options**. Go to the Diagram tab and click Collaboration Numbering off. Click Apply. Click Collaboration Numbering back on and click Apply again.)
10388	Arial font size not consistent. Certain icons with Arial font style does not display the correct font size. If 14pt Arial font size is selected, certain icons (i.e: usecase/business/interface) icons may display in 12pt size.
10622	Print output may not appear the same as the screen display. This issue only occurs when printing from a UNIX/Linux display. This does not occur when printing from a PC display using Exceed.
11133	Fit/Undo Fit in Window may result in undesirable layout. Undo Fit in Window may not result to its original display/layout.
11416	Font scaling difficulties when using Rose fonts. Problem with scaling occurs when doing print preview.
11615	Help button on title bar missing. The **?** button on the title bar of various dialogs (located left of the **X** button) does not appear initially, unless: 1 The area where the **?** button should appear is clicked on. 2 The window area containing the spot where the **?** button should appear is moved off screen, and then moved back on screen.

11800	You may encounter an unresolved model warning if there is a view of an item on a diagram, but the underlying item is not loaded or could not be found in the current model.
	Items and relations are correctly displayed and reported by Check Model as unresolved references when the view to the item is in a diagram that is in the current model but the item resides in an unloaded package (controlled unit). The reference becomes resolved and the unresolved indicator is removed when the unit containing the item is loaded.The unresolved model warning occurs occasionally during model loading if the model was created by a previous version of Rose and that version of Rose allowed an invalid model to be created. Some known cases are:
	Circular Generalize/Realize Relationship Combinations (allowable in Rose2000e but no longer valid in this release). Note that if you attempt to create a circular generalize/realize relationship combination, you will get an error and the relationship will not be created.Circular generalize/realize relationship combinations that exist in models files created with earlier versions of Rose are detected on model load and one of the relationships involved in the circularity is left unresolved.
	You may delete the unresolved view and then add new ones as appropriate. Running **Tools > Check Model** will provide in a list of unresolved views in the Rose log. If the relationship has no view on a diagram in the model, then an "unresolved relation" warning is placed in the Rose log. The warning is also generated by **Tools >Check Model**.
	To correct the unresolved relations:
	1 Load the model into the Rose
	2 Check the log file for "Unresolved relation" warnings
	3 Open the spec of the "from" class to the relations tab
	4 Note that the unresolved relation (generalize or realize) is the one with the class name in brackets in the name column
	5 Determine which classes are involved in the circularity
	6 Delete the undesired relationship via the class spec relations tab
	Shared Objects - Shared activity diagram objects (allowable in Rose2000e but no longer valid in this release). A shared object is an object that belongs to one state machine and has a view on an activity diagram belonging to another state machine. The drag and drop of the object is now disabled if the object does not belong to the same state machine as the diagram.In general, the unresolved object can be deleted from the diagram and a new object can be created on that diagram. The model does not lose integrity or validity for doing this since an object is not actually a model item - it doesn't define anything, it is only an instance of a class used for visualizing state or behavior.

11810	Printed diagrams can differ from what is displayed on the user's screen. For example, some longer operations do not print completely.
	If the text on the screen is:
	`provideWorkflowClosure(WorkItemClosureData) : void`
	the printout might only show:
	`provideWorkflowClosure(WorkItemClosureData) : v`
11879	Display problem with Message\More dialog and long names.
14249	A new **Creates a Form** class diagram toolbar is available, but you must manually add it to the toolbar.
14739	If you use the inline-editing capability on a diagram to rename an operation parameter name, then any OVERRIDDEN model properties associated with that parameter are reset to the DEFAULT value.
	Note that if you rename the RoseParameter via REI or through the specification dialogs, then the model properties are not reset.
15037	Role Name will not remain on diagram if context menu is used to create it.
	Workaround: Define the Role Name using the Association Specification dialog.
17061	If a state or activity diagram contains a view of an element from another state or activity diagram/model, the **Query** > **Expand** command will not work for that element.
18168	Deleting a package with a class attached to a read-only object in an activity diagram will cause Rose to crash.
18229	Encapsulated Postscript %%Page and %%BoundingBox not conforming to standard.
18733	Infrequently, a modal dialog may move behind Rose. In this event, you will need to move Rose to access the modal dialog.
	Contact Rational Technical Support if you encounter this issue. See *Contacting Rational Technical Support* on page viii.
18734	It is not possible to delete an unloaded controlled unit from a model. To delete a unit from a model, first load that unit and then delete it.
18890	Rose may crash when Web Publishing a model if the following environment variables are not set, *and* csh **unlimit** has been run, prior to running Rose:
	▪ MWVISUAL_CLASS=TrueColor
	▪ MWVISUAL_DEPTH=24
	Workaround: If this occurs, unsetenv MWVISUAL_CLASS and MWVISUAL_DEPTH prior to running Rose.
25404	Some diagrams can lose or have additional vertical toolbars.

31538	On Solaris displays, double clicking on the top bar of the IEditor can cause the IEditor to disappear. This defect does not occur on other UNIX displays or on PC displays, running Exceed.
57200	'x' is disabled in the 'error Box' when drawing a 'Realize' reationship from 'class' to 'package'.
58064	IEditor Horizontal scroll bar may not appear. Workaround - Bring up the IEditor by selecting View:Editor prior to browsing code. If the IEditor is already running, the horizontal scroll bar will always correctly appear.
66268	Context menu arrow may not display properly when the class is placed at the edge of the window.
66926	Diagram items not dragging correctly after using 'birds eye view' utility.
71347	Including ":" Within the package name of a controlled unit can cause Rose to crash. It is important not to use the ":" character within a package name.
79812	Rose requires Internet Explorer version 5.01 SP 2 (rather than Internet Explorer 4.01, as Rose Release Notes incorrectly indicate).
80298	Containment property (role) documentation changes not updated in online Help. The possible values for the collectionValid values shoud be listed as: * Unspecified = 0 * ByValue = 1 * ByReference = 2

Fixed Defects in Core Rose and Extensibility

Fixed Defects in This Release

Defect	Description
1998	Problems dragging association point on a diagram.
3881	If a Java or Corba specification dialog is displayed and the Rose browser is not docked, Rose will lose focus when the specification dialog is closed. To proceed the user must terminate the Rose process and all unsaved data will be lost.
56047	File : "Save subunits of <...>" does not work correctly, when the Rose window is maximized. Workaround: This will always work correctly when the Rose window is minimized or upon selecting View : Toolbars.
58660	Fully qualified names for types in operation return and argument types are not updated correctly when the class is relocated.
67865	Compartment items not maintained after copying and pasting operations, after saving *.cat file.
70964	Transitions may no longer be properly linked to a Decision, after resizing the Decision.
71257	Rose may crash upon copying and pasting an activity diagram which contains swimlanes and object flows.
72192	Focus of Control issues in Sequence Diagrams.
72654	Messages may not get updated correctly upon moving operations between classes.
74412	Rose may crash upon performing an undo delete of class and interface in a controlled unit.
75510	Unable to assign a custom size to a Focus of Control.
76525	Stereotype display icon (decoration) not always displaying correctly.
76945	Rose may crash on second attempt to delete a write-protected association.
77042	File types such as 'dtdFile dtd' type (.dtd extension) may become associated with Rational Rose.
79513	Rose 2001a and 2002 models may not be compatible, due to the introduction of a new 'quidu' element designed to enhance performance. Upon using Rose 2002 to create an attribute of an array of some other type in the model, Rose 2002 will store the internal reference 'quidu' to the base type. Upon loading this model in Rose 2001a, the array type is lost as the type is based on the quidu (which, Rose 2001a does not recognize. Rose 2002.05.20 introduces a new petal version format (version: 46) to resolve this issue.

Defect	Description
80557	Crash may occur when state is moved and transition text label is automatically repositioned.
81785	Manipulating the contents of a Sequence Diagram (i.e. deleting elements, adding new objects) and then creating a message between two objects, may cause Rose to crash.
82309	Connections can be lost between messages and operations in Sequence Diagram.
82755/ 83306	Selecting an operation from the **More** context menu, by right-clicking on a message in a Sequence Diagram will cause Rose to crash. Workaround: Use the **Specification** Dialog to select the operation when several operations exist and the **More** menu must be utilized.

Fixed Defects in Service Release v.2002.05.01

Defect	Description
12541	Objects in a Sequence Diagram can overlap when performing dragging operations.
66331	Class icon 'auto resize' not functioning correctly.
66855	Scrolling in diagram may cause crash on call to: :__1cHCObjectIIsKindOf6kMpknNCRuntimeClass__i_.
67057	Moving objects in Sequence Diagrams may cause unexpected behavior to construction messages.
68146	Allow Rose splash screen to go behind other applications.
69961	When attaching a file to an instance of an object on an activity diagram, an absolute path is saved rather than saving the file reference relative to the closest PATHMAP variable.
70855	Moving messages in Sequence Diagram can cause Rose to crash.

Rose Extensibility Interface (REI) Release Notes

Rose Extensibility Interface Features and Enhancements

The Rational Rose Extensibility Interface (REI) provides several ways for you to extend and customize Rose's capabilities to meet your specific software development needs. Using REI capabilities, you can:

- Customize Rational Rose menus

- Automate manual Rational Rose functions with Rational Rose Scripts (for example, diagram and class creation, model updates, document generation)

- Execute Rational Rose functions from within another application by using the Rational Rose Automation object (RoseApp).

- Access Rational Rose classes, properties and methods right within your software development environment by including the Rational Rose Extensibility Type Library in your environment.

- Activate Rational Rose add-ins using the Add-In Manager

There are no new features or enhancements for this release of the Rose Extensibility Interface (REI).

Netscape V4.7x Issues

Netscape 4.7*x* is provided in the *rational_dir*/base/cots/netscape.4.7*x*/*$arch* directory of this Release. **rs_help** uses Netscape to display Installation Guide and Release Note information. Rose for UNIX/Linux will occasionally use a browser. Rose will first use the browser defined in the users "BROWSER_PATH" environment variable or Netscape included in this release, if BROWSER_PATH is not defined.

If you have not used Netscape 4.7*x*, we recommend you consult the Netscape README file located at

rational_dir/base/cots/netscape.4.7*x*/*$arch*/README

This file includes a warning, recommending that an existing **$HOME/.netscape** file or directory be renamed prior to running Netscape 4.7. Otherwise, data may be lost (i.e. old preferences, bookmarks, cookies, etc.). The README file includes other helpful information such as platform specific issues as well.

ANSI C++ Add-In

<div align="right">

4

</div>

Rose ANSI C++ is the Rational Rose language add-in in support of the C++ programming language.

Because of its many benefits, users are encouraged to convert their models from Rose C++ to Rose ANSI C++. Rose ANSI C++ is designed to be powerful enough to handle large projects, with particular emphasis on scalability and completeness of language support.

A model converter is included as an add-in with this release, and the ANSI C++ online help provides instructions for this easy conversion.

Rose ANSI C++ Features and Enhancements

Rose ANSI C++ provides:

- User can specify the file extension for all generated classes · Source code file can be named differently than the class.

- Source and body files can be generated into different directories.

- Support for modeling of exception specifications.

- Provides support for default sets on the **Component Specification** style page.

- Support default sets for component styles.

Rose ANSI C++ provides:

- Design, modeling and visualization of all C++ constructs including classes, templates, namespaces, inheritance, and class members functions.

- Support for the C++ language, without being restricted to a single vendor's C++ compiler.

- Support for large frameworks.

- Automatic generation and reverse engineering of C++ source code.

- User-controlled code generation using patterns of default constructors, destructors and class members. One-step reverse engineering from Rose, without a separate import step.

- Style sheet mechanism to allow custom formatting of generated code.

- Simplified, yet complete round-trip engineering support, which synchronizes models and generated C++ code across multiple iterations. After assigning files or classes to a project, you can forward or reverse engineer the entire project using a single context menu command.

- Update-in-place mechanism to determine the differences between model and code, and make only those changes required to maintain consistency between the two.

Known ANSI C++ Defects

Defect	Description
8350	Reverse engineering with the ANSI C++ add-in does not create a diagram that contains all reverse engineered classes. Although they do not appear on a diagram, the reverse engineered classes do exist in the model. You can see them in the browser and can add them to a diagram with drag and drop, or by using the Query	Add Classes menu item.
9084	The ANSI C++ add-in will create two unidirectional associations on reverse engineering when the model originally has a bi-direction association between two classes and both roles of the association are named.
10776	When generating code, Rose ANSI C++ does not make backup copies of files it changes, nor does it provide an undo mechanism for such changes. You are strongly encouraged to backup your source files or have them under an effective source code control mechanism before generating code.

11115	If a typedef defines more than one name, the ANSI C++ add-in does not reverse engineer the typedef.
	
	For example, reverse engineering the following code completes without error:
	>>>
	typedef struct myStruct {
	    DWORD Version;
	    DWORD Size;
	} myName, * lp_myName;
	<<<
	
	However, the struct myStruct appears in the model, but the typedef does not.

14630	The ANSI C++ add-in ignores comments in source code.
15120	After you convert a class or model from Classic C++ using the ANSI C++ Model Converter, you generate code for the class or model before attempting reverse engineering. Attempting to reverse engineer before generating code will cause this message to be displayed: Class cannot be reverse engineered because it does not exist in the project files.
16714	**Apply** button on component dialog is never enabled. Workaround: "OK" button is available and enabled.
17114	When a template class is deleted from the code, and then the code is reverse engineered, the corresponding template class in the model is deleted; however, anonymous classes that are instantiations of that template are not deleted.

Defect	Description
17513	When indicating "vi" as the editor of choice in the **Tools > ANSI C++ > Preferences** dialog, it is necessary to invoke vi within the context of it's own window, by entering the following command in the preferences. **xterm -e vi command_line_args** Entering "vi", instead of the command listed above will start vi if an xterm window is currently open and will display the source but it then receives an 'input read error' and exits vi. emacs and gvim will work as expected.
18360	Creation order of classes affects code generation from class (but not from component). If a class containing a data member is created before the class being contained and code is generated by selecting both classes on a class diagram and choosing **Tools > ANSI C++ > Generate Code**, the #include for the contained class is not generated. If code is generated from the component (not the classes), the #include is generated. If the contained class is created first and code is generated from the class diagram, the #include is generated.

20639	Types nested within template instantiations may not be used as base classes or the ends of assocations, since such types are not modeled. Reverse engineering such code can result in an incorrect model.
	
	For example, if the following code is reverse engineered, the base class of class c will be "a<b>", not "a<b>::z".
	
	<pre>template <class t> class a {
	    typedef t z;
	};
	class b { };
	class c : public a<b>::z { };</pre>

21809	Automatically generated #include directives are always relative to the source file root directory. There is no easy way to force them to always be simple names or to make them relative to some other directory.
24206	With AutoSync enabled, when a class in the model is changed from being at package scope to being a nested class, its new definition is correctly added to the code of its containing class, but its old definition in the code is not deleted.
76415	Converting to ANSI C++ from C++ may cause loss of association relationships in diagram.

Fixed ANSI C++ Defects

Fixed ANSI C++ Defects in This Release

Defect	Description
22935	Changing stereotype to <<parameterizedClass>> does not display until another code update occurs.
30368	Incomplete entry is made when attempting to generate code when a type has a "$" in it's name.
31580	ANSI C++ does not generate code for ALL fully qualified assocations if target language not ANSI C++. Potential Workaround - Model the data member as an attribute. The attribute may be instead of the association, or in addition to the association (since the association will be ignored by ANSI C++).
55134	Codename property not saved when converting from Classic C++.
55269	OperationIsConst property not converted to ANSI C++ as stereotype for operation.
55272	OperationKind C++ property not converted to ANSI C++ stereotypes.
56191	ANSI C++ Reverse Engineer of Composition displays as Aggregation.
56256	Inline property of operation lost while converting from Classic C++ to ANSI C++.
56258	OperaionIsExplicit property not converted to ANSI C++ as stereotype for constructor.
58326	Reverse Engineering of complex roles does not produce correct results.
66172	ANSI C++ code generated ('Role A General' Tab) incorrectly when autosync enabled.
66567	Associations disappear when performing C++ Model Conversion with autosync enabled.
68936	Macro defining namespace keyword causes entire namespace not to be reversed.
71656	Upon reverse engineering, class is not updated to parameterized class correctly.
71890	Converting to ANSI C++ from C++, attribute name is not generated correctly.
74289	ANSI C++ Converter crashes when converting a model with incorrect (inconsistent) C++ code.
74305	ANSI C++ Converter does not maintain association that was not code generated.
74487	Generation of ANSI C++ code for multiple modules functions incorrectly.

Defect	Description
74615	Upon converting from C++ to ANSI C++, association relationships no longer appear in the diagram but still exist in the class specification.
74785	Reverse engineering and generating code on the Orion component looses code.
76485	Converting to ANSI C++ from C++ fails if class is assigned to C++ Package Body and Spec components.
77974	Generalization to instantiated class is not maintained upon ANSI C++ reverse engineering.
79459	Template instantiation is not reverse engineered correctly, after code is updated.
79688	Relationship not maintained, upon converting C++ to ANSI C++.
81467	Converting to ANSI C++ from C++ may cause crash if a class is not assigned to ANSI C++ component.
81636	No code generated when attempting to generate code on a generalization class after adding attributes.
81653/ 22950	Source code not updated correctly upon changing parameterized class to a regular class in model and regenerating code.
81785	Creating a message between two objects on a Sequence Diagram, after performing other manipulations in the Sequence Diagram may cause Rose to crash.
82000	Rose crashes if add attribute (or operation) to a class with dependency to a C++ class and regenerating code.
82093	Function body not maintained in code, upon converting a class to a template and re-generating code.

Fixed ANSI C++ Defects in Service Release v.2002.05.01

Defect	Description
21101	Conversion from C++ to ANSI C++ fails if class is assigned to C++ package body and spec components.
31580	ANSI C++ should generate code for all fully qualified associations.
56191	ANSI C++ reverse engineering of composition displays incorrectly as aggregation.
68559	ANSI C++ crash upon recreation of association after deleting it.
65155	When a function with an exception specification returns a pointer to the array, parser incorrectly handles as a data member. When such a function is static and has a definition, a crash occurs.

Defect	Description
68321	Generating code on an association role name that has a specified type and then immediately reverse engineering, the association is automatically converted to an attribute instead of correctly remaining an association.
69839	Assertion failure may occur upon selecting 'Get' and 'Set' check boxes when performing class customization on an attribute that is an instantiated class.

Rose C++ Add-In

(Not Available for Linux)

5

Note: This addin is no longer supported. The addin is still shipped with Rose but is disabled by default. If the addin is needed, it can be enabled using the addin manager available from the Rose top level menu.

Rose J Add-In

<div align="right">6</div>

The Rose J Add-In supports true round-trip engineering and has support for large frameworks.

The following sections provide information on the Rose J Add-In:

New Rose J Add-In Features and Enhancements

The Rose J add-in supports true roundtrip engineering and provides support for large frameworks. This release of the Rose J add-in provides the following new capability:

- Added support for Remote Interfaces.

Rose J Design Pattern Support (Not Available on HP-UX 10.20)

When you apply a design pattern to your model, Rose J adds it in the form of new classes and components for each participant in the design pattern. Rose J places new Rose elements (classes, methods, packages, relations, etc.) in the Logical View and adds a component in the Component View for every added or updated class. The component name also appears in parentheses beside the class name in the Logical View.

When you forward engineer your model into Java code, Rose J populates the design pattern participant classes with methods that have complete code bodies, as well as methods with empty code bodies to be filled in by you.

Rose J includes support for the following design patterns covered in the book, *Design Patterns, Elements of Reusable Object-Oriented Software* (ISBN 0-201-63361-2), by Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides, often referred to as the "The Gang of Four."

- Creational Patterns
  - Abstract Factory

- □ Factory Method
- □ Prototype
- □ Singleton
- Structural Patterns
  - □ Adapter
  - □ Bridge
  - □ Composite
  - □ Decorator
  - □ Facade
  - □ Flyweight
  - □ Proxy
- Behavioral Patterns
  - □ Chain of Responsibility
  - □ Command
  - □ Iterator
  - □ Observer
  - □ State
  - □ Strategy
  - □ Template Method
  - □ Visitor

Known Rose J Add-In Problems and Limitations

The following table contains known Rose J problems and limitations in Rose for UNIX/Linux.

Defect	Description
09370	In Operation specification Window under Code tab the read only field is not greyed out.
10988	Get/set operations are not visible in Sequence Diagrams.

Defect	Description
11721	Unable to create directory on an NFS mounted disk when defining source directory using Java or CORBA project specification.
12950	Rose J uses a code generation property to indicate if an operation is abstract, rather than the model property used by the Rose specifications. For this reason when you specify an operation as abstract in the Standard Specification this is not reflected in the Custom dialog box. The reverse is also true.
13935	Utilizing tabs in the Rose J style specification causes errors when round trip engineering. Workaround - Utilize spaces instead of tabs.
14628	Reverse engineering of very large Java files, can cause memory issues.
15420	Code generating for EJB's does not prompt user for all classes at the same time (only prompts for one at a time).
15481	No warning dialog to indicate there are warnings in log if in Java Auto Sync Mode.
15487	Java code gen does not begin when class is created with Auto Sync already enabled.
15554	When reverse engineering .cab files (which include .java and/or .class files), the method body contents of the source files is missing.
15891	Reverse engineering of .cab files may produce unpredictable results. When reverse engineering .cab files (which include .java and/or .class files), the resulting Rose model may not be correct. The Rose log records errors, but does not alert the user.
16021	No context sensitive help for object finders labels in EJB Persistence Properties window.
16037	If you have a package in your model that has a space in its name, for example "w p" and a class attribute references a type defined in that package, the generated code will include the space and will be incorrect.
16295	JavaDoc tags are not generated correctly.
17256	Unable to create new EJB/Servlet from statechart icon.
18195	The Java code generation OK prompt (modal dialog) can infrequently go behind Rose. If this occurs, it will be necessary to move Rose to respond to the modal dialog - before proceeding with Rose activities.
20257	Pattern Support -> Singleton - Singleton interface should implement java.lang.RunTimeException when exception is thrown.
21096	For J2EE, EJB classes and the diagrams that contain them must be in the same package in order for the EJB **Update** to work.

Defect	Description
31374	Unable to delete Finalizer method from browser after turning on Autosync.
57390	Rose may crash when switching EJB type with Forte IDE.
64316	Rose may crash after saving a message EJB in the editor and bringing up the EJB
74071	No documentation in online Help about the **Reference** checkbox placed in the specification of a class. The Reference checkbox is used to mark a class as a reference in order to prevent modifying the source code of the class by generating code. When the Reference checkbox in the specification of a class is checked, Rose J cannot generate code (forward engineer) for this class. The Reference checkbox can be checked in two ways. Open the class specification and checked the checkbox. Reverse engineer a class from a Reference classpath. When the class is created in Rose, it is automatically marked as reference.

Fixed Rose J Defects

Fixed Rose J Defects in This Release

The following table contains Rose J Add-In defects that have been fixed since the previous release.

Defect	Description
16593	EJB DocComment - extra newline erroneously inserted and inconsistent indentation in code. To reproduce: Create a class and convert it to EJB. Enter multiline comments in the DocComment fields for the class, for the operation and attributes. Generate code. Extra new lines are inserted between consecutive lines of comment in the model. The indentation is also inconsistent.
24832	New EJB CREATE method cannot be created.
30858	Error when reverse engineering java files with comment in if / else statement (in multiline statement).
61221	Error message (Enumerated value public does not exist) may occur upon reverse engineering a servlet.
66132	Parser error may occur when comment (i.e., //) is before closing brace.
66786	Generic servlet icons are available, but is missing the "GS" text.
67314	Comments below Model ID are removed after Round Trip Engineering.

Defect	Description
69341	Convert to EJB assumes class diagram is open and is currently the diagram in focus, when attempting to display the newly created EJB.
69564	EJB*Local*Interface options not available from Stereotype selection list.
70075	Code generation for package names (i.e. those including spaces) may not be handled correctly (i.e. the space may be deleted).
70076	Signature of operation are not preserved when commented out at the end of the file.
70206	Java Component Specification Imports tree sorting not sorting correctly.
70359	Reverse engineering can give a reference to a class in the Java or Rose notation.
70901	Autosync does not update a class when drawing a generalize from the interface it implements to another interface with an operation.
71212	Java file is not moved into new directory when moving the package that contains it to another package.
71676	EJB Local and LocalHome suffix fields lose their default values.
72794	Code generation may cause loss of data if both the *.sub and *.mdl files are read only. If only the *.cat and/or *.java files are read only - the problem will not occur. This is resolved by checking whether the file for the model file of the component is read only, prior to modifying the *.java file.
73587	Operation in nested class with argument not fully qualified will reverse engineer incorrectly.
75052	Reverse engineering the rt.jar file may produce the error - "Wrong file format… [path to *.class file[".
75210	Incorrect code generated for arrays in EJB business methods.
75873	Create EJB version 1.1 utilizes EJB 2.0 dtd in the descriptor file.
77011	Support for the Java SDK 1.4 assert statement.
77050	Error (Error resolving qualified name */IOException occurred in File C:\path\x.java) on comment between throws and exception, if comment contains a comma.
77112	Error (Parser Error: near token) on line n occurred in File C:\path\x.java) on comment inside an empty operation parameter list.
77113	Error (Parser Error: near token if on line n occurred in File C:\path\x.java) on comment between else and if
77361	Error (Parser Error: near token { on line n occurred in File C:\path\x.java) on comment for an anonymous inner class.

Defect	Description
77639	Reverse engineering error (Parser Error: near token char on line occurred in File C:\path\x.java) on a comment in front of a method argument list.
78100	Code generation performed incorrectly on array parameter (two brackets instead of one will be generated).
78454	Round trip engineering will cause data loss if ")" is included in code.
79333	J2EE Framework model references J2SE 1.2 components instead of correctly referencing the J2SE 1.3 components.
79480	Incorrect code generated upon reverse engineering code when there are brackets in one line of code.
79733	Warning message added to log when Rose J Add-in does not update getter and setter code.
80317	Significant performance degradation when attempting to reverse engineer Java files.
80697	Location for ejb-jar_2_0.dtd should be updated in EJB 2.0 deployment descriptor.
81861	Unable to reverse engineer xerces.jar. Errors may occur, i.e. Cannot assign a public class to more than one component). Rose may crash.

Fixed Rose J Defects in Service Release v.2002.05.01

Defects	Description
27765	Rose cannot automatically update fully qualified paths for argument types of classes relocated from one package to another.
54211	Rose J handles white space and "&" characters in Logical View package names inconsistently.
57137	JBuilder class in package is not deleted in Rose even with autosync on (JBuilder 6 is required for this fix to take effect).
58432	Generating Java code for setters places custom source code in comments. Upon fixing this defect, the following will apply: The Rose J Add-In will only add default code body for getter/setter if and only if the method body in the code file is EMPTY. In this way, any user-defined code will remain. This is also a consistent behavior with the "generate default return line" feature. If there is ever a case you change the attribute from one bean property type (i.e. Simple) to another (i.e. Bound) and would like a new getter/setter method to be generated in the code file, you may either: - remove the getter/setter method in the code file; OR - remove the method body in the getter/setter method in the code file

Defects	Description
58944	Support for roundtrip engineering using the JBuilder "Folder" specific feature.
60169	Rose J loss of implementation when Logical and Component views differ.
64897	When reverse engineering a parser error occurs when there is a comment between the variable initializers.
65574	Turning on GenerateFullyQualifiedTypes for attributes from model properties causes Rose to stop using default.
66704	Creating a final argument in a servlet operation (while autosync is active) will generate the code in the argument name twice.
68506	Rose unable to correctly locate the JDK that JBuilder installs.
68949	Crash upon activating autosync and deleting component.
69539	Error occurs ("an unnamed file was not found") when relocating a module, while utilizing the JBuilder IDE Link.
70059	Reverse engineering Java class with association to itself, presents a class with an association to an unspecified class.
70215	Static initializers blocks may be deleted during code generation.
70753	Rose may crash when changing the name of an entity EJB Primary Key class.
70755	J2EE EJB - Converting to EJB can create duplicate attributes.
71523	When round -trip engineering, if the class contains an operation with arguments referencing other classes without being fully qualified - method bodies may be deleted.
71785	Java parser errors occur when trying to reverse engineer a java interface that has "//" style comments on the line preceding the close brace.
71877	Dragging a unidirectional association from a class to a servlet may leave rose process running after exiting Rose.
72290	Slow performance when reverse engineering.
73819	When using JBuilder integration, Rose may erroneously indicate the project does not contain any source files.
74262	Round trip engineering Http Servlet may cause a loss of implementation code.

Rose Ada Add-In

7

The Rose Ada add-in allows you to produce Ada source code from the information contained in a Rose model.

The code generated for each selected model component is a function of that component's specification and code-generation properties, and the model's properties. These properties provide the language-specific information required to map your model onto Ada.

Rose/Ada preserves user-supplied declarations and statements from one iteration to the next. You place such user-supplied code in protected code regions which are preserved whenever code is re-generated.

The Rose/Ada Code Generator:

- Substantially reduces the elapsed time between design and execution

- Produces uniformly-structured source code files, promoting consistent coding and commenting styles with minimal typing

- Supports both Ada 83 and Ada 95

Rose Ada Add-In New Features and Enhancements

There are no new Rose Ada features or enhancements for this version release of Rose.

Rose Ada Fixes With No Defect Associated

Rose SGI: Remove dependency on an OS call, the CodeMapper interpreter is returning a spurious status code. The OS call was for performing a file compare to determine whether an update was necessary.

For a role, where RecordFieldImplementation is set to Discriminant, the container declaration occurred after the type definition.

Known Rose Ada Defects

Defect	Description
15927	Apex & Rose/Ada problems can occur when performing code generation on checked-in units.
25926	Rose Ada uses Handle before it's Definition for public types in a reflexive association.
62201	Private constant not generated correctly.

Rose Ada Defects Fixed in This Release

Defect	Description
14965	Rose/Apex needlessly reports circularities among nested packages,
16142	Private Only visibility for code generation of operations required.
69859	Ability to specify a text file in the Copyright notice property required.
71327	Ability to control the order of code generation in a package required. A new (attribute,has,role) property Declare=(Before,After) can be used to control where the static declaration should be declared - either before or after the class type.
74511	Wrong signature generated in the Generic Subprogram Body.
75639	Rose Ada: System Error: line 10811, pc 27607 Null Handle in request for property "inherits".

Fixed Defects in Service Release v.2002.05.01

Defect	Description
29061	Rose/Ada does not support RUP component view mapping.

Rose Web Publisher Add-In

8

The Rational Rose Web Publisher Add-In allows you to publish Rational Rose models in HTML format. The Web Publisher Add-In exports your Rose model to HTML (including diagrams) so you can view them using a standard browser such as Netscape Communicator. The Web Publisher also produces meta tags for use by search engines.

New Web Publisher Add-In Features and Enhancements

There are no new features or enhancements for this version of Rose Web Publisher.

Browser Data

In order for Netscape to find the correct Java jar files, you must have the browser on your PATH, or set the environment variable BROWSER_PATH to the absolute path of your browser.

The Web Publisher on UNIX supports only .jpg file format.

The Web Publisher has only been tested with Netscape versions 4.6 and 4.7*.

Rose Web Publisher Known Defects in This Release

Defect	Description
18890	Rose may crash when Web Publishing a model if the following environment variables are not set, *and* csh **unlimit** has been run, prior to running Rose: ■ MWVISUAL_CLASS=TrueColor ■ MWVISUAL_DEPTH=24 Workaround: If this occurs, unsetenv MWVISUAL_CLASS and MWVISUAL_DEPTH prior to running Rose.

XML DTD Add-In

9

The Rational Rose XML DTD Add-in provides visualization, modeling, and code generation to create XML document type definitions (DTD) for use in your application. You can reverse engineer an existing XML DTD to visualize a DTD's structure using a Rose class diagram. After modeling an XML DTD, you can use the Syntax Checker to validate your XML DTD. From the valid XML DTD model, you generate code to create a new XML DTD.

In addition to reverse engineering an existing XML DTD, you can model a DTD using Rose class stereotypes for XML elements, element attributes, entities, and notations. By modeling an XML DTD, you can visualize the structure of the document to see which element definitions to change or remove. You can also share your XML DTD model with members of your development team to verify that you have captured the XML document requirements.

An XML DTD syntax checker finds errors in the XML DTD model, allowing you to make corrections before the DTD before is implemented. Use the Generate Code feature to produce an XML DTD that you can test or implement in your environment.

To support mapping XML to UML, Rational Rose extends UML with stereotypes for XML elements, element attribute lists, entities, and notations. Stereotypes also represent XML operator symbols, sequence lists, choice lists, and element and element attribute multiplicity.

Note: DTDElementANY cannot be nested. Use this DTDElement at the root level.

Known XML DTD Add-In Problems and Limitations

The following table lists known XML DTD problems or limitations in Rose for UNIX/Linux.

Defect	Description
09129	Apply button not enabled for Component Spec window changes.
10584	Expanding tree structure in XML reverse engineering dialog can be slow in UNIX. This can be due to accessing NFS mounted disks.
11382	File extension case sensitive on UNIX after generating code. Reverse engineering and then generating code on *.DTD file, does not generate code to existing *.DTD but creates a new *.dtd file (case sensitive) on UNIX.

XML DTD Add-In Defects Fixed

There are no new fixed defects in this release.

Additional Add-Ins

10

This chapter contains information on the following add-ins:

- *Rose Model Integrator Add-In* on page 55
- *ClearCase Add-In* on page 57
- *DDL Add-In* on page 58
- *CORBA Add-In* on page 59
- *Framework Add-In* on page 60
- *Apex Add-In (Not available for Linux)* on page 61

Rose Model Integrator Add-In

The Rose Model Integrator Add-In lets you compare model elements from up to seven contributor files, discover their differences, and merge them into the base model. There are no new features for this release of Rose Model Integrator.

Known Model Integrator Add-In Problems and Limitations

The following table contains known Model Integrator problems and limitations in Rose for UNIX/Linux.

Defect	Description
1236 11454	Model Integrator does not support ClearCase elements. Model Integrator does support ClearCase versions for primary units specified on the command line or in the Contributors dialog, but it does not correctly parse subunit file names to create version-enhanced file names used by ClearCase. The workarounds are: 1. Merge subunit files individually using ClearCase. 2. Create separate views in ClearCase which contain all the proper versions of your model's subunits and then merge the models independently of ClearCase (start Model Integrator in the normal way and enter the model file names in the Contributors dialog. Check the resulting merged model back into ClearCase when you are finished. You will need to create 3 views: the base view containing the previously checked in version, the latest version, and the version you want to merge.
1614	Project specs in Java and CORBA. Directories specified for code generation may not be merged properly into the merged model. The workaround is to reenter these directories in Rose.
2169	Model Integrator allows user to select more than 7 contributors. Model Integrator only allows 7 contributors, but it allows the user to select an 8th contributor and then issues an error message.
7469	Model Integrator creates duplicate model elements. In a situation where one contributor has renamed a model element to new name and another contributor has added the same new name to the model without renaming the old model element, Model Integrator may produce a model with two model elements having the same name. The workaround for this is that when the model is loaded into Rose, Rose will report an error and rename one of the model elements automatically. Visit these elements in the Rose browser window and correct the names of the elements.
18159	In the Contributors dialog, the checkbox for the option "Compare/Merge against Base Model" should be grayed out until two contributors have been added.
25249	**Select All** does not select the main node in the tree view.

Fixed Model Integrator Defects in This Release

Defect	Description
58683	Model Integrator does not handle unnamed objects properly.
66188	Model Integrator performs merge on unrelated diagrams, which results in a corrupt model. In this case, the diagrams being merged were located in different state machines, different names and UID.

Fixed Model Integrator Defects in Service Release v.2002.05.01

Defect	Description
67981	Rose may crash when selecting 'NewDiagram' once an .mdl file merged by Model Integrator is loaded in Rose.

ClearCase Add-In

Note: Prior to using the Rose for UNIX/Linux ClearCase Add-In, it is necessary to review and follow the steps outlined in the README file. Please see the *rational_dir*/releases/rose.I.J.K/addins/clearcase/README file.

The ClearCase Add-In provides a tight integration between Rational Rose and the Rational ClearCase version control system. This add-in is intended as a replacement for the ClearCase capabilities provided through the previous Rose/ClearCase integration. This has and will continue to allow Rational to provide more ClearCase specific customization to this add-in.

When performing ClearCase actions on a unit in a Rose model, the ClearCase Add-In will now perform CM status checking on all ClearCase elements within the model. This provides an opportunity to perform the same ClearCase action on any element which applies. When the ClearCase dialog is displayed, you will have an opportunity to perform the selected ClearCase action on any or all of the ClearCase elements presented in the dialog. The CM status checking which is performed to provide this new capability can at times seem lengthy if working with a large model. In this case, you may elect not to utilize this feature by placing the following in either your $USER.reg (applies to single user) or the rational_dir/releases/rose.I.J.K/registry/rose.reg file (applies to anyone using the release):

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Rational
    /Software\Rose\AddIns\ClearCase\Custom Settings]
"ShowAll"="No"
```

Once the registry file is edited the user(s) should run rational_dir/releases/rose/I.J.K/bin/rose_cleanup and then run 'rose' for the change to take effect.

Known ClearCase Add-In Problems and Limitations

The following table contains known ClearCase problems and limitations in Rose for UNIX/Linux.

Defect	Description
15035	When using Rose/ClearCase Integration in Rose for UNIX/Linux, ClearCase output will appear in the terminal used to run Rose in addition to the ClearCase dialogs.
16647	If a user has selected the Rose option of creating back up files but has not selected to create the back up files by copy, Rose may not detect when a newer version of a file has been made available from ClearCase. To project against this, make sure that the Update by Copy option has been selected.

Fixed ClearCase Defects

The following table lists ClearCase Add-In defects fixed since the last release.

Defect	Description
63576	Selecting multiple items and right clicking in the diagram to display context menu, may cause Rose to hang. Workaround: After selecting multiple items in the diagram, use the Tools menu to select option or make selections and right click in the browser.
78647	Rose may hang upon opening ClearCase LT Explorer. Workaround: Close ClearCase LT Explorer in order to use Rose.

DDL Add-In

The Data Definition Language (DDL) Add-In release note contains known problems and limitations as well as fixed defects since the last release.

The gap between object technology and relational databases is closed by providing a mapping interface to relational databases through generation of persistent classes to Structured Query Language (SQL) and Data Definition Language (DDL).

You can find DDL Add-In release note information in the following locations:

- Known DDL Add-In Problems and Limitations

Known DDL Add-In Problems and Limitations

There are no known problems and limitations for the DDL Add-In.

CORBA Add-In

Rose CORBA Features and Enhancements

The Rational Rose CORBA add-in allows you to forward engineer Rose model elements into CORBA-compliant IDL code and reverse engineer CORBA IDL code into a Rose model.

In support of the CORBA 2.3 Specification, Rose CORBA implements the use of value types. Value types allow you to pass objects by value rather than by reference, which is particularly useful when an object's primary purpose is to encapsulate data, or when you want to explicitly make a copy of an object. Two CORBA stereotypes make use of this enhancement:

- CORBAValue
- CORBAFixed

Known CORBA Add-In Problems and Limitations

The following table contains known CORBA Add-In problems and limitations:

Defect	Description
1770	You can create a nested class in CORBA typedef. You are not supposed to use nested classes in a typedef. This is described in the online documentation for CORBA.
1771	Reverse Engineering reports parse error on comment before }; CORBA reverse engineering reports an error when a comment is added to the last attribute of a struct.
1773	Some Include statements are generated incorrectly.
3881	Rose may hang if a Rose J or CORBA operation and attribute specification dialog box is viewed while the browser is undocked. Rose is unable to gain focus, once the specification dialog is closed.
6478	Nested classes in a CORBA Typedef mnay cause problems during round-trip engineering and are no longer supported. Contact Rational Technical Support if you have any questions.

9664	Syntax Checker should check to see whether a valid modifier is selected before reverse engineering idl.
11721	Unable to create directory over NFS in Java or CORBA Project Specification.

Unable to create directory on an NFS mounted disk when defining source directory using Java or CORBA project specification. |
| 16853 | Browse source <ctrl+H> or <ctrl+K> shortcut key on CORBA classes fails. It will be necessary to use the **CORBA > Browse Code** menu option. |
| 57198 | Exit sign 'x' disabled when the position of the Rose/CORBA Add-In dialog box is moved. |

Fixed CORBA Defects in This Release

The following table shows the CORBA Add-In defects that have been fixed since the last release.

Defect	Description
65666	Incorrect package may be created under component view when RE a file that has include statement to other file.
70948	Incorrect IDL code generation for enums.

Fixed CORBA Defects in Service Release v.2002.05.01

Defect	Description
68178	Inheritance may be lost upon reverse engineering.

Framework Add-In

The Framework Add-In provides a library of frameworks that can be used as templates when creating new models. There are no new features or enhancements for this version of the Framework Add-In.

Apex Add-In (Not available for Linux)

The Apex Add-In of Rose for UNIX/Linux provides direct access to the Compile and Control commands in Apex. It also works with language add-ins for Rose, including Rose C++, Ada 95, and Ada 83 Add-Ins, to create Apex subsystems and views during the code generation from the model to the source files of the specified language.

- It is required to run Rose inside an Apex shell window to use the Apex Add-Ins.

- If the **Apex** submenu is not in the **Tools** menu, select **Add-Ins > Add-In Manager** to make sure the Apex Add-In is activated.

The following information is also located in your Apex Add-In online documentation.

Apex Compile and Control Commands

The Compile and Control commands of Apex are directly accessible from Rose through the **Tools > Apex** menu. The Compile commands compile and link the source files, generated into Apex views via Apex Add-In, that correspond to the selected items in the diagram. The Control commands perform Apex source code control operations on the source files generated in the same way.

To use these commands, select **Tools > Apex**, and then select Compile or Control to get to the list of commands, respectively.

For the descriptions on these Apex commands, please consult the online help in the Apex windows.

Code Generation With Apex Add-In

In general, the code generation operation for a regular language Add-In is based on a directory structure where top-level packages, nested packages, and classes (or modules) are mapped, respectively, into directories, subdirectories, and source files in a hierarchical way. The operation can create such directories or subdirectories if they do not already exist, and then generate source files into these (sub)directories. This works with the regular source code control system and build management.

Apex has a different style of source code control and build management. An Apex project has a set of subsystems which contain the source files and subdirectories of source files for development. Each developer has his own views, one for each subsystem, which collectively reflect the source base the developer uses for the project. For this reason, Apex-style code generation is different from the regular style. From a Rose model, the top level packages will be mapped into subsystems and the associated views for the developers; the packages nested inside another package will

be mapped into the subdirectories inside an Apex view, or a subdirectory already enclosed by a view; the classes (or modules) will be mapped into the source files inside the views or the subdirectories.

The Apex Add-In provides a mechanism for the user to select which of the two code generation styles to use. When the property CreateApexSubsystemAndView of the Apex tool set is set to "Yes", the code generation command for a language add-in will automatically create Apex subsystems and views for the top-level packages selected in the model, followed by the generation of subdirectories and source files into the views. If the property is not set to "Yes", the code generation command does the regular style of code generation.

The language add-ins that work with Apex Add-In include C++, Ada 95, and Ada 83.

Properties for Apex Add-In

Refer to the online documentation for a list of the properties used by the Apex Add-In for Apex-style code generation. Some are properties for the project, and the others for the subsystems in the model. The code generation command will use the combination of these properties when it needs to generate code into Apex subsystems and views.

All the properties except CreateApexSubsystemAndView will take effect only when the CreateApexSubsystemAndView property is set to "Yes" and when the selected item is a top-level package. When any of these properties are specified differently from their default value, the Directory property for the top-level package should remain as the default value.

Utilizing Apex Add-In on HP-UX and SGI IRIX

In order for Apex to generate subsystems correctly, it is necessary for "tmp-mnt" to be placed in the automount field of the Virtual Pathmap dialog box. The "/tmp-mnt" entry should be removed from the Virtual Pathmap dialog box when not using Apex.

Utilizing Apex Add-In on AIX

When using the Apex Add-In to create Apex subsystems and views on AIX, Rose for UNIX/Linux might run into a hanging state. If that happens, look for an Apex "Job Queue" window. In that window, select all the entries and click **File > Run Job**. This will resolve the "hanging" state and resume code generation.

Known Apex Add-In Problems and Limitations

The following table shows known defects in the Apex Add-In.

Defect	Description
3853	CM commands do not show on the context menu for packages in the browser. When this occurs, you are unable to use Apex CM integration on a package that does not appear in the diagram. Workaround: Add the package to the diagram.

Rose Ada Defects Fixed in This Release

Defect	Description
14965	Rose/Apex needlessly reports circularities among nested packages,
16142	Private Only visibility for code generation of operations required.
69859	Ability to specify a text file in the Copyright notice property required.
71327	Ability to control the order of code generation in a package required. A new (attribute,has,role) property Declare=(Before,After) can be used to control where the static declaration should be declared - either before or after the class type.
74511	Wrong signature generated in the Generic Subprogram Body.
75639	Rose Ada: System Error: line 10811, pc 27607 Null Handle in request for property "inherits".

Fixed Defects in Service Release v.2002.05.01

Defect	Description
29061	Rose/Ada does not support RUP component view mapping.

Index