

Configuring Rational Suite

VERSION: 2001.03.00

PART NUMBER: 800-023720-000

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

Rational®
the e-development company™

IMPORTANT NOTICE

COPYRIGHT

Copyright ©1998-2000, Rational Software Corporation. All rights reserved.

Part Number: 800-023720-000

Version Number: 2001.03..00

PERMITTED USAGE

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

TRADEMARKS

Rational, Rational Software Corporation, the Rational logo, Rational the e-development company, ClearCase, ClearCase Attache, ClearCase MultiSite, ClearDDTS, ClearQuest, DDTS, Object Testing, Object-Oriented Recording, Objectory, PerformanceStudio, PureCoverage, PureDDTS, PureLink, Purify, Purify'd, Quantify, Rational Apex, Rational CRC, Rational Rose, Rational Summit, Rational Visual Test, Requisite, RequisitePro, SiteCheck, SoDA, TestMate, and The Rational Watch are trademarks or registered trademarks of Rational Software Corporation in the United States and in other countries. All other names are used for identification purposes only, and are trademarks or registered trademarks of their respective companies.

Microsoft, the Microsoft logo, Active Accessibility, Active Channel, Active Client, Active Desktop, Active Directory, ActiveMovie, Active Platform, ActiveStore, ActiveSync, ActiveX, Ask Maxwell, Authenticode, AutoSum, BackOffice, the BackOffice logo, BizTalk, Bookshelf, Chromeffects, Clearlead, ClearType, CodeView, Computing Central, DataTips, Developer Studio, Direct3D, DirectAnimation, DirectDraw, DirectInput, DirectMusic, DirectPlay, DirectShow, DirectSound, DirectX, DirectXJ, DoubleSpace, DriveSpace, FoxPro, FrontPage, Funstone, IntelliEye, the IntelliEye logo, IntelliMirror, IntelliSense, J/Direct, JScript, LineShare, Liquid Motion, the Microsoft eMbedded Visual Tools logo, the Microsoft Internet Explorer logo, the Microsoft Office Compatible logo, Microsoft Press, the Microsoft Press logo, Microsoft QuickBasic, MS-DOS, MSDN, Natural, NetMeeting, NetShow, the Office logo, One Thumb, OpenType, Outlook, PhotoDraw, PivotChart, PivotTable, PowerPoint, QuickAssembler, QuickShelf, Realmation, RelayOne, Rushmore, SourceSafe, TipWizard, TrueImage, TutorAssist, V-Chat, VideoFlash, Virtual Basic, the Virtual Basic logo, Visual C++, Visual FoxPro, Visual InterDev, Visual J++, Visual SourceSafe, Visual Studio, the Visual Studio logo, Vizact, WebBot, WebPIP, Win32, Win32s, Win64, Windows, the Windows CE logo, the Windows logo, Windows NT, the Windows Start logo, and XENIX are trademarks or registered trademarks of Microsoft Corporation in the United States and other countries. FLEXIm and GLOBEtrotter are trademarks or registered trademarks of GLOBEtrotter Software, Inc. Licensee shall not incorporate any GLOBEtrotter software (FLEXIm libraries and utilities) into any product or application the primary purpose of which is software license management.

Portions Copyright ©1992-2000, Summit Software Company. All rights reserved.

PATENT

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

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

GOVERNMENT RIGHTS LEGEND

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

WARRANTY DISCLAIMER

This document and its associated software may be used as stated in the underlying license agreement. Rational Software Corporation expressly disclaims all other warranties, express or implied, with respect to the media and software product and its documentation, including without limitation, the warranties of merchantability or fitness for a particular purpose or arising from a course of dealing, usage, or trade practice.

Contents

Preface	ix
Audience	ix
Other Resources	ix
Rational Suite Documentation Roadmap	x
Contacting Rational Technical Support	xi
1 Welcome to Rational Suite	13
Rational Suite Editions	13
2 Migration and Conversion	15
Before You Proceed	15
Upgrade Order	15
Overview of Upgrade and Conversion in This Release	16
Rational ClearQuest Designer	16
Rational Administrator	16
Rational RequisitePro	17
Rational RequisiteWeb	17
Understanding a Rational Project	18
Unified Change Management	18
Configuring Rational Projects and Datastores	18
Upgrading Your Rational Components	19
Rational ClearCase VOB Storage	19
Backing Up Your Information	19
Backing Up a Rational ClearCase VOB	19
Backing Up Your Rational ClearQuest Database and Rational RequisitePro Project	20
Upgrading your Rational Administrator Repository	21
For More Tools Information	22
3 Rational Suite Administration	23
Installing Rational Suite Software	23
Licensing Rational Suite	23
Before You Plan	23
Developing a Database Plan	25

Creating and Managing a Rational Project	25
Security Issues	25
Writing Synchronizer Rules	26
Customizing Extended Help.	26
4 Creating a Rational Project	27
What is a Rational Project?	27
Planning the Project.	27
Which Rational Tools Will the Team Use?	27
Which Commercial Databases Will You Use?	28
Will You Use Existing Rational Data?	28
Using the Rational Administrator	29
Setting Up the Project	29
High-Level Steps for Setting Up a Rational Project	30
Upgrading Existing Data.	30
Upgrading ClearQuest Data	30
Upgrading RequisitePro Data	31
Upgrading Rational Test Data	31
Upgrading Rose Data	31
Installing and Configuring Databases.	31
Deciding Where to Locate the Project	32
Creating a Project.	32
Adding Users and Groups	32
Rational Test	32
ClearQuest	32
RequisitePro	33
Attaching Existing Rose Models	33
Placing Data under Configuration Control	33
Using Rational Suite without a Repository.	33
5 Configuring the Integration for ClearQuest and RequisitePro. . .	35
Overview: Setting Up the Integration	35
Initial Integration Setup	36
Creating a Rational Administrator Project	36
Out-of-the-Box Tools.	37
Collecting Setup Information	37
Editing the ASCQISetup.bat file	39
Executing the Batch File.	41
ASCQISetup.exe Command Argument Descriptions	42

Upgrading Your Existing Integration	44
Upgrading the ClearQuest Schema Database	44
Upgrading the ClearQuest User Database	45
Executing the Batch File	45
Customizing Your Integration	46
Changing the Default RequisitePro Requirement Types	46
Changing the Default RequisitePro Project Attributes	46
Changing the Default ClearQuest Records	47
Adding your Customizations to the ASCQISetup.bat file	47
6 Rational Synchronizer	49
Overview of Using the Synchronizer	49
Example	49
Creating and Updating Items	50
In this Chapter	50
Using the Synchronizer	50
Starting the Synchronizer	51
Startup Context: What to Expect	51
Working with the Synchronizer	51
Administration for the Synchronizer	52
Importing Rules	52
How Rules are Stored	52
Using Rational Suite: Importing Rules	53
Working without Rational Suite: Importing Rules	53
Writing Synchronizer Rules	54
Understanding Your Development Process	54
What Can You Specify in a Rule?	54
Sample Rule without Annotations	55
Sample Rule with Annotations	55
Sample Rules Supplied with the Synchronizer	59
The Synchronizer Rule Template File	60
Synchronizer Rule Grammar	60
Converting the RequisitePro Database	62
Preparing the RequisitePro Database for Synchronization	64
Converting Rose Models	66
Editing the Rules	66
For More Information	66

7	Working with Extended Help	67
	Using Extended Help	67
	Starting Extended Help	67
	Changing Context	67
	Adding or Removing Databases	68
	Making a Database Active or Inactive	68
	Adding or Deleting an Extended Help Database	68
	Administering Extended Help	69
	Upgrading or Converting an Extended Help Database	69
	Converting from a Microsoft Access Database to a .csv File	70
	Creating an Extended Help Database	71
	High-level Process	71
	Populating the Database	71
	Valid Product and Subtool Values	72
	Working with Translation Strings	73
	Defining Translation Strings on User Systems	74
	Deleting a Translation String	74
8	Rational Suite Out-of-the-Box Tools	75
	Rational RequisitePro Project Templates	75
	Creating a Project Based on the Use-Case Template	75
	The Rational Rose RUP Model	77
	Tool Mentors	77
9	Additional Resources	79
	Online Release Notes	79
	Error Message	79
	EHRegister	79
	Index	81

Tables

Table 1	Commercial Database Choices	28
Table 2	ClearQuest – RequisitePro Integration Configuration	36
Table 3	Variables Describing Your Environment	37
Table 4	Rational ClearQuest Related ASCQISetup.exe Parameters	42
Table 5	Rational RequisitePro Related ASCQISetup.exe Parameters	43
Table 6	Sample Rules	59
Table 7	Synchronizer Rule Syntax	60
Table 8	RequisitePro Use-Case Project Template	76

Preface

This manual describes how to set up Rational Suite for use by a development team.

Rational Suite delivers a comprehensive set of integrated tools that embody software engineering best practices and span the entire software development life cycle.

Rational Suite's unparalleled level of integration improves communication both within teams and across team boundaries, reducing development time and improving software quality.

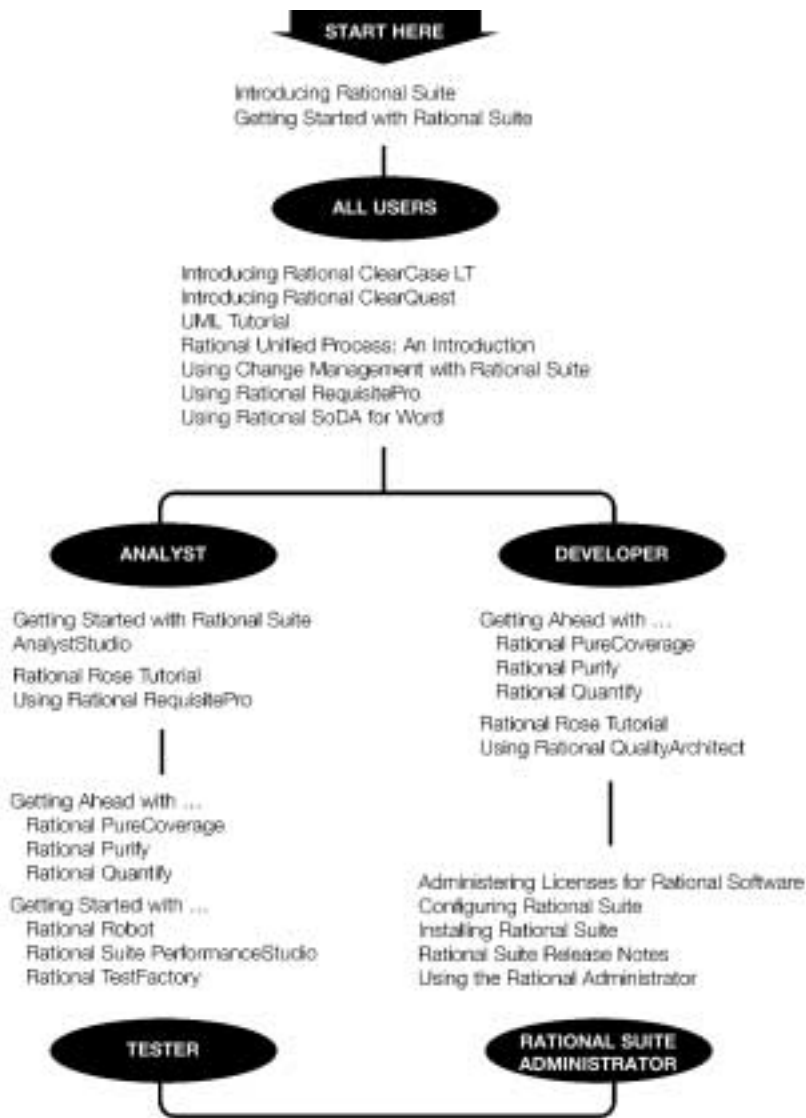
Audience

This guide is intended for Rational Suite administrators.

Other Resources

- Online Help is available for Rational Suite.
- From a Suite tool, select an option from the **Help** menu.
- All manuals are available online, either in HTML or PDF format. The online manuals are on the Rational Solutions for Windows Online Documentation CD.
- To send feedback about documentation for Rational products, please send e-mail to techpubs@rational.com.
- For more information about Rational Software technical publications, see: <http://www.rational.com/documentation>.
- For more information on training opportunities, see the Rational University Web site: <http://www.rational.com/university>.

Rational Suite Documentation Roadmap



Contacting Rational Technical Support

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

Your Location	Telephone	Facsimile	E-mail
North America	(800) 433-5444 (toll free) (408) 863-4000 Cupertino, CA	(781) 676-2460 Lexington, MA	support@rational.com
Europe, Middle East, Africa	+31 (0) 20-4546-200 Netherlands	+31 (0) 20-4545-201 Netherlands	support@europe.rational.com
Asia Pacific	+61-2-9419-0111 Australia	+61-2-9419-0123 Australia	support@apac.rational.com

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

- Your name, telephone number, and company name
- Your computer's make and model
- Your 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)

Welcome to Rational Suite

1

Rational Suite is an integrated set of software development products. It combines best-in-class tools with the industry's best practices encapsulated by the Rational Unified Process.

This book shows you how to configure Rational Suite, including how to set up databases for Rational Suite products, how to use and customize the Synchronizer and Extended Help, and how to use predefined projects to get started quickly.

Additional information for each Rational Suite product is located in each product's online Help and in printed documentation. See also *Tool Mentors* on page 77, which describes features to help you use the Rational Unified Process with Rational Suite.

Rational Suite Editions

Rational Suite is available in these editions, tailored for each member of your team:

- Rational Suite AnalystStudio — optimized for the analyst
- Rational Suite DevelopmentStudio — optimized for the developer
- Rational Suite DevelopmentStudio – RealTime Edition — optimized for the developer of real-time, embedded systems.
- Rational Suite TestStudio — optimized for the quality engineer
- Rational Suite Enterprise — complete set of software development tools

The following table lists the products in each Rational Suite edition.

Rational Tool	Analyst Studio	Development Studio (Windows or UNIX)	Development Studio - RealTime Edition	Test Studio	Team Unifying Platform	Enterprise
Rational Unified Process	X	X	X	X	X	X
Rational RequisitePro	X	X	X	X	X	X
Rational ClearQuest	X	X	X	X	X	X
Rational SoDA for Word (Windows) or for FrameMaker (UNIX)	X	X	X	X	X	X
Rational ClearCase LT	X	X (Windows only)	X	X	X	X
Rational TestManager	X	X	X	X	X	X
Rational Rose	Data Modeler Edition	Enterprise Edition	RealTime Edition			Enterprise Edition
Rational Robot				X		X
Rational TestFactory				X		X
Rational PureCoverage		X	X	X		X
Rational Purify		X	X	X		X
Rational Quantify		X	X	X		X
Rational Quality Architect		X (Windows only)				X

If you are working with a Rational Suite product, and you plan to use your existing datastores after you convert to a new release, you must upgrade your Rational Suite product in a specific order.

This chapter discusses how to effectively and efficiently convert data that was developed in a previous version of a Rational Suite product.

Before You Proceed

Warning: Back up your current repository before you upgrade any Rational product. Failing to back up your data can cause you to lose your data during the upgrade.

Warning: Back up your ClearCase LT VOB storage, your ClearQuest databases and your RequisitePro databases simultaneously before you upgrade. See the section, *Backing Up Your Information* on page 19.

Read the *Rational Suite Release Notes* and the release notes for each individual Rational product you plan to upgrade before you proceed with your upgrade. These documents may contain information that became available only after this book went to press. To access the Release Notes, click on **Start > Programs > Rational Suite [suite name] > Readme for Rational Suite** (or **Readme for all products** to link to individual products' release notes).

Upgrade Order

You must upgrade your Rational Suite components in the following order:

- 1 If you do not use Rational ClearCase for controlling versions of your Rational Software artifacts, proceed now to Step 2.

If you do use Rational ClearCase and have any files, Rose models, RequisitePro projects, or other Rational Suite artifacts checked into your VOB, you must check those items out of the VOB before you can upgrade them. When you have finished your upgrade, you will check the artifacts back in.

- 2 Update Rational ClearQuest Designer
- 3 Update Rational RequisitePro and RequisiteWeb
- 4 Update Rational Administrator

Overview of Upgrade and Conversion in This Release

The following is an overview of the expected behavior of particular products:

Rational ClearQuest Designer

With many integrations between Rational ClearQuest and other Rational products, you may be required to upgrade Rational ClearQuest packages so the integration will continue to work. Rational ClearQuest Designer provides this upgrade functionality. It is important that the ClearQuest package application upgrade occur before upgrades to other Rational datastores such as the Rational Repository, Rational TeamTest, and Rational RequisitePro datastores. Please consult the Rational ClearQuest release notes for the latest upgrade information and procedure.

Rational Administrator

The Rational Administrator manages the datastores and databases that store your software testing and development information. If you used a previous release of a Rational product, your software testing and development information was stored in a repository with one or more projects in it.

The current release of Rational software uses a project to store Rational Test datastores, RequisitePro projects, ClearQuest databases, and Rose models.

If you used a previous release of Rational software, you need to use the Administrator to convert your repository to a project. When you use the Administrator to convert your repository to a project, the Rational Administrator conversion tool:

- Informs you what Rational ClearQuest packages must be upgraded; you must manually run the upgrade procedure. See the section *Upgrading Your Rational Administrator Repository* on page 21 for more information.
- Tries to determine that the proper packages have been installed in the associated Rational ClearQuest database before proceeding with the upgrade.
- Runs the Rational RequisitePro project upgrade tool via an API. This API does not do the checking listed below under the next subsection, *Rational RequisitePro*.
- Converts each project in a repository into a new Rational Administrator project. It also maintains any links that you created in a repository project to a Rational ClearQuest database, Rational RequisitePro project, or Rose model.

See the books, *Using the Rational Administrator* and *Installing Rational Suite* for further information. Also see the *Rational Software Setup Online Help* for more information.

Rational RequisitePro

When you open a stand-alone Rational RequisitePro project, RequisitePro detects whether an upgrade is required or not and informs you of its result. If you associated a Rational RequisitePro project with a Rational Administrator repository or project, you must use the Rational Administrator to upgrade the associated RequisitePro project. (See *Upgrading Your Rational Components* on page 19 of this section.) If you created a Rational RequisitePro project in the Rational Administrator, you must convert the repository to a project. When you convert the repository to a project, the RequisitePro project converts automatically. The following table summarizes when to upgrade a RequisitePro project or convert a repository to a project.

If you	then	For more information, see
Created a RequisitePro project and did not associate it with a Rational repository	RequisitePro upgrades your project automatically when you open a RequisitePro project.	<i>Using Rational RequisitePro</i>
Created a RequisitePro project using Rational Administrator	Use the Rational Administrator to convert a Rational repository to a Rational project. The RequisitePro project converts when the repository converts. In Administrator, click File > Convert Rational Repository into Rational Projects .	<i>Using the Rational Administrator</i>

Rational RequisiteWeb

When upgrading Rational RequisiteWeb, uninstall any earlier version of RequisiteWeb before you install the current release on your server. To uninstall RequisiteWeb, you must do the following:

- Stop the IIS Admin Service and all application server processes.
- Remove the RequisitePro application from your RequisitePro requirements server using Windows Add/Remove program in the Control Panel.
- Start the World Wide Publishing service.

Note: When upgrading, do not remove your existing RequisiteWeb virtual directory; instead, in the RequisiteWeb virtual directory Properties dialog box, change the **Local Path** by browsing to the location of the new RequisiteWeb components in the directory <Local Path>\webapps\ReqWeb.

Microsoft Word 2000 is now required on your RequisiteWeb requirements server. If you upgrade from Word 97 to Word 2000, be sure to delete the contents of the c:\inetpub\wwwroot\ReqWebDoc\GenDoc directory before attempting to view RequisitePro documents in RequisiteWeb.

For more information, refer to the section, *Installing and Configuring Rational RequisiteWeb*, in the *Installing Rational RequisitePro* manual, or to *ReqWebSetup.html* included in your RequisiteWeb installation.

Understanding a Rational Project

A Rational project stores software testing and development information. The Rational Administrator lets you manage a project that includes Rational Test datastores, RequisitePro projects, ClearQuest databases, and Rose models. If you used a previous release of a Rational product, your software testing and development information was stored in a repository. You need to convert your repository to a project to manage your software testing and development information with the current version of the Administrator.

You need to convert repositories to projects if you used prior versions of Rational Suite, or Rational Testing products such as Rational Robot 7.x.

Unified Change Management

UCM is the Rational Software approach to managing change to requirements, design models, documentation, components, test cases, source code, and so on. The Rational Administrator enables Unified Change Management (UCM) for managing RequisitePro projects, test datastores and Rose models.

You can bring the information in a Rational project under configuration management using Rational ClearCase by associating a Rational project with a UCM project. Use ClearCase to create a UCM project first, and then create a Rational project and UCM-enable it. ClearCase LT ships with the Rational Administrator.

For more information, see the manual, *Using Unified Change Management with Rational Suite*.

Configuring Rational Projects and Datastores

When you create a Rational project, you also must configure it. Configuring a project associates Rational Test datastores, RequisitePro projects, ClearQuest databases, and Rose models with the project and optionally brings them under configuration management.

See *Using the Rational Administrator* and the *Rational Test Release Notes* for more information on the Rational Administrator and how to upgrade your Rational products in the Administrator.

Upgrading Your Rational Components

Rational ClearCase VOB Storage

Before you upgrade, determine whether the users of your system use Rational ClearCase's VOB storage for RequisitePro or ClearQuest elements. If you do not do this, you need do nothing about VOB storage during your upgrade.

If they do use ClearCase, check out all elements (files, Rose models, RequisitePro projects, etc.) before you proceed with the upgrade, then perform your upgrade normally (the Installer detects that ClearCase is present on your system and does not install ClearCase LT over it). When the upgrade is complete, check out the elements from the VOB and upgrade them as described in this chapter, then check them back into the VOB. See the section *Backing Up a Rational ClearCase VOB* on page 19 and the book *Administering ClearCase* for more information.

Backing Up Your Information

Before you upgrade your Rational components, you must back them up. This section provides a high-level overview of how to back up your components.

You should back your Rational components up regularly. We recommend you back up your data at least once per day.

Caution: You must plan your backups to synchronize with one another. Not synchronizing your backups can cause serious problems with restoring your data. Thus, if you are using any two Rational Suite elements together (such as ClearCase LT and RequisitePro or RequisitePro and ClearQuest, and so on) you must back those datastores up simultaneously.

Note: You must have exclusive control over the data in order to back it up. The first step in each procedure below gives you that exclusive control. Backups usually take no longer than 15 minutes, depending on your particular system.

Backing Up a Rational ClearCase VOB

Be sure you back up all synchronized Rational Suite data simultaneously.

- 1 Lock the VOB.
- 2 If your backup software can be configured to capture files that are open for write access:
 - a Back up the VOB storage directory.
 - b Unlock the VOB.
- 3 If your backup software cannot process files that are open for write access, you can:
 - Copy the VOB storage directory, unlock the VOB, and back up your copy.
 - Copy the VOB database subdirectory to an on-disk location, back up the database copy plus the resto of the storage directory in place, and unlock the VOB.
 - Use the semi-live backup strategy described in the manual *Administering Rational ClearCase* in the section entitled *Standard or Semi-Live Backup?*
 - Stop ClearCase LT on the VOB host, back up the VOB storage directory, unlock the VOB, and restart ClearCase.

For more information, see the *Administering Rational ClearCase* manual.

Backing Up Your Rational ClearQuest Database and Rational RequisitePro Project

Be sure you back up all synchronized Rational Suite data simultaneously.

Be sure you are logged on with an ID with administrator privileges.

Back up your Rational ClearQuest data using the backup tool associated with your database of choice.

- 1 Become the exclusive user on each of the databases you will back up and on all directories and flat files associated with the database (including RequisitePro RQS and source directories, and Rational TeamTest project directories).
- 2 Back up your data. Consult your database vendor's backup procedures for instructions on backing up your data.
- 3 Remove your exclusive user designation.

Upgrading your Rational Administrator Repository

To upgrade all Rational components that are part of a Rational Administrator repository, you must convert the repository to a Rational Administrator project.

Note: To upgrade a RequisitePro project or convert a repository, you must have administrator permission for each component of the repository. We also recommend that you back up your current repository before you convert to a project. You will not be able to use it as a repository after you convert it to a project.

- 1 Upgrade the schema packages and your user database in Rational ClearQuest using the ClearQuest Designer Package Upgrade tool. Use the latest versions of the following upgrade packages: Repository, RequisitePro, RequisitePro Supplement, and TeamTest. See *Administering Rational ClearQuest* for more information.
- 2 Upgrade your RequisitePro project. The following table tells you how to convert a RequisitePro project created with Rational software previous to the current version

If you	then	For more information, see
Created a RequisitePro project and did not associate it with a Rational repository	RequisitePro upgrades your project automatically when you open a RequisitePro project.	<i>Using Rational RequisitePro</i>
Created a RequisitePro project using Rational Administrator	Use the Rational Administrator to convert a Rational repository to a Rational project. See step 3 below.	<i>Using the Rational Administrator</i>

- 3 Convert your Rational Administrator repository to a project:

- a In the Rational Administrator, click **File > Convert Rational Repository into Rational Projects**, and follow the steps in the wizard.

Caution: When you upgrade a RequisitePro project that uses a SQL Server or Oracle database, the Rational Administrator upgrades all RequisitePro projects in that database. You will be asked to enter an administrative password for the database in order to proceed.

- 4 To upgrade your ClearQuest and RequisitePro integration:

- a Locate your ASCQISetup.bat file, which you used for your original integration setup. If you no longer have your edited batch file, edit the ASCQISetup.bat file that is included in your Rational Software installation. (If you installed

following the default settings, you will find the file in the C:\Program Files\Rational\common\ directory.) For more information, see the section *Configuring the Integration for ClearQuest and RequisitePro* on page 35. See also the book, *Using Rational ClearQuest* for more information.

- b After you've prepared ASCQISetup.bat, log on as an administrator in the ClearQuest User database.
- c Delete all RequirementMap records (a RequirementMap defines a link between a RequisitePro project and your ClearQuest database record).
 - i Be sure you are logged in as the ClearQuest database administrator.
 - ii Create a query to show all files with a record type of RequirementMap.
 - iii Delete the records shown by that query.
- d Run the ASCQISetup.bat file.

Note: Rational Rose models that are included in your Rational Administrator repository do not need to be upgraded for the Rational Administrator project.

For More Tools Information

See the *Using the Rational Administrator* manual for information on how to convert a repository to a project. See the *Using Rational RequisitePro* manual for more information about the Project Upgrade utility. See the *Using Rational ClearQuest* manual for more information about the Package Upgrade utility.

This chapter provides an overview of the tasks typically performed by a Rational Suite administrator.

Installing Rational Suite Software

The Rational Software Setup program installs Rational Suite and other Rational Software products, using a consistent and familiar installation wizard.

You can find additional details about Rational Software installation in *Installing Rational Suite*, included with your Rational Suite media kit. See the Rational Windows Setup online Help for assistance while using the program.

Licensing Rational Suite

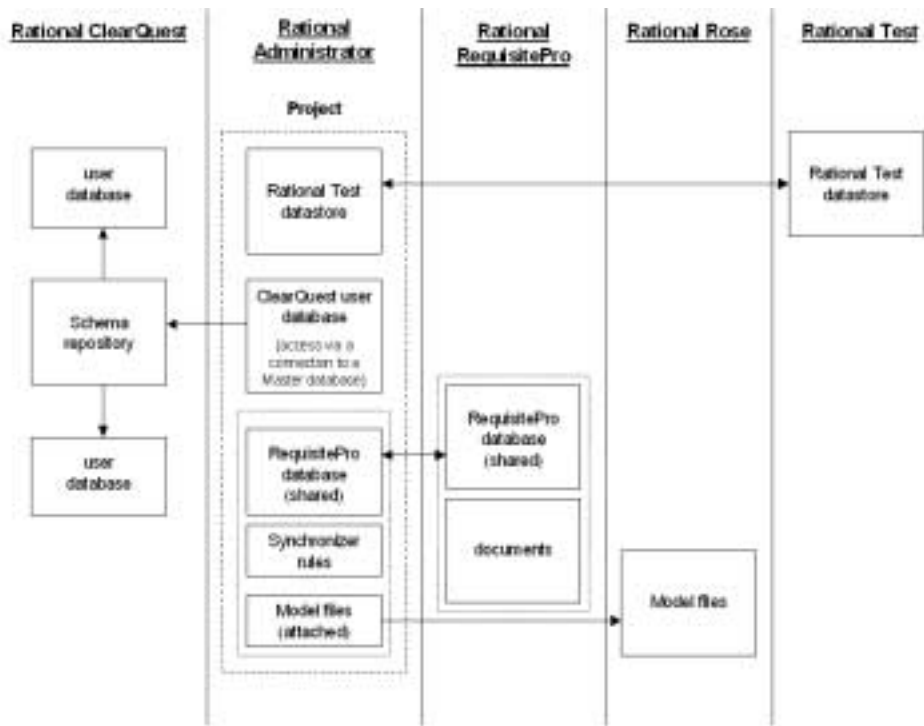
Rational Suite uses FLEXlm licensing services from GLOBEtrouter, Inc. For more information about licensing, please refer to one of the following sources:

- *Administering Licenses for Rational Suite*, included with your Rational Suite media kit.
- Online Help for the Rational License Key Administrator. The Help provides step-by-step instructions for installing startup license keys and requesting permanent client and server license keys.

Before You Plan

Before you set up your data stores and repositories, you should plan. Before you can plan, though, you need to understand logical relationships among the databases used by Rational Suite products, as shown in the following figure.

Figure 1 Rational Suite Structure Relationships



Some notes concerning the figure:

- The arrows between the Master database and the two user databases indicate logical connections; information in the user databases are subsets of information in the Master database.
- The arrow between the Master database and the ClearQuest user database indicates access to the user databases through the Master database.
- The arrow between the two RequisitePro databases indicate that both databases are actually the same database.
- The arrow between the two Rational Test datastores indicates that the two datastores are the same.
- The arrow between the two model files indicate that both files are actually the same file.

Developing a Database Plan

As an administrator, you need to decide which database management system(s) (DBMS) to use with Rational Suite products.

Creating a Rational Project on page 27 provides more information about planning for and installing databases.

Installing Rational Suite, included with your Rational Suite media kit, describes the installation, configuration, and administration of different database management systems, particularly for use with ClearQuest.

Creating and Managing a Rational Project

A Rational datastore is a logical collection of datastores that associates the data you use on a Rational Suite project. Once you create a project, you can take advantage of integrations among Rational Suite tools.

Creating a Rational Project on page 27 provides more information about creating and configuring a project.

The *Using Rational Administrator* manual, included with your Rational Suite media kit, describes how to manage a datastore.

Security Issues

You can set some security parameters through the Rational Administrator. However, you must set most parameters through the individual point product (such as Rational Rose or Rational ClearQuest) by setting up the appropriate users and groups. See the documentation for the individual product for more information on how to set security parameters. You must do this in each of the products.

The ClearQuest administrator sets up the ClearQuest list of users and then set security on the project. See *Using Rational ClearQuest* for more information.

The RequisitePro administrator sets up the RequisitePro users and groups, then sets security on the project. See *Using Rational RequisitePro* for more information.

You can use the Rational Administrator to set up users and groups for test. See *Using the Rational Administrator* for more information.

To put security on Rational Rose, which does not support user lists, secure the directory..

Writing Synchronizer Rules

The Rational Synchronizer helps maintain project consistency, resulting in improved team communication. The Synchronizer works by creating and updating a project's related items based on the existence of other items in your project.

As an administrator, you may be responsible for analyzing your team's development process and using the results of the analysis to develop Synchronizer rules.

For more information about this process, please see *Rational Synchronizer* on page 49.

Customizing Extended Help

Extended Help provides a connection between Rational Suite tools and the Rational Unified Process. You can augment Extended Help with your own team's process descriptions.

For more information about customizing Extended Help, please see *Working with Extended Help* on page 67.

Creating a Rational Project

4

This chapter describes how to create a Rational project for use with Rational Suite.

What is a Rational Project?

A Rational project is a logical collection of datastores that associates the data you use on a Rational Suite project. A Rational project consists of:

- One Rational Test database
- Zero or one ClearQuest database
- One or more projects containing:
 - One Rational Suite Synchronizer rules file
 - One RequisitePro database
 - Zero, one, or more Rose model files

Once you create a Rational project, you can take advantage of product integrations. For example, a project allows you to:

- From the LogViewer window of the Rational TestManager, generate a defect in ClearQuest.
- Work with the Rational Synchronizer to create or update project data based on the existence of other data.

The amount of integration you can achieve often depends on the ClearQuest database schema you use; see the book, *Administering Rational ClearQuest* for more information.

Planning the Project

Before you create a project, you need to do some initial planning, as described in the remainder of this section.

Which Rational Tools Will the Team Use?

Your team may want to use all the products in Rational Suite or just a subset of the products. Determine which Suite products your group plans to use.

Which Commercial Databases Will You Use?

For each Suite product you are using, decide which commercial database to use. You can use different databases for different Suite products. The following table summarizes the choices based on the size of the user group planned for each product; please refer to product documentation for more details about each choice.

Table 1 Commercial Database Choices

	ClearQuest	RequisitePro	Rational Test
Microsoft Access	Small groups	Small or medium groups	Small groups
SYBASE SQL Anywhere Server	Medium or large groups	N/A	Medium or large groups
Microsoft SQL Server	Large groups	Large groups	N/A
Oracle	Large groups	Large groups	N/A

Database performance depends on factors such as: number of users, network response time, number of records in the database, and database tuning.

Microsoft Access Runtime is automatically installed when you install at least one Rational Suite product on your machine. (The Access development environment is not installed.) Also, Sybase SQL Anywhere is available on the product CD, but it is configured to work only with Rational products.

Will You Use Existing Rational Data?

Does your project include existing data created with earlier versions of Rational Software? If so, collect information about which data you want to use and where it's located. For example, if you've used earlier versions of Rational tools, you might want to continue using the following types of data:

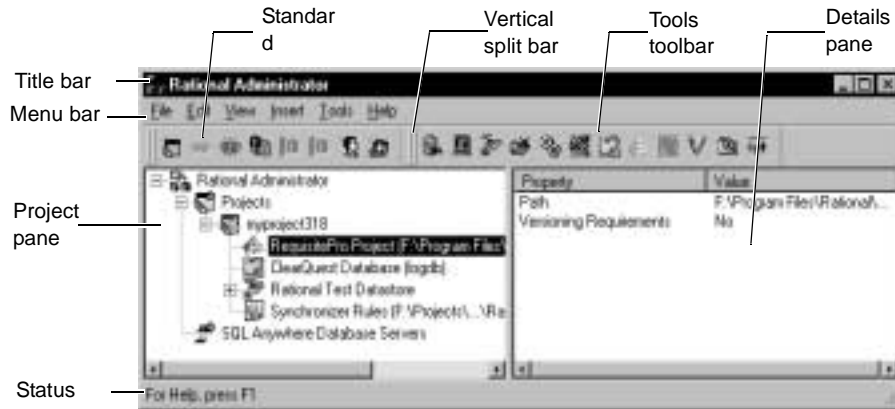
- ClearQuest change requests or schema repositories
- RequisitePro requirements
- Rose models

We recommend that you upgrade existing data before you start production work on your project; see *Upgrading Existing Data* on page 30.

Using the Rational Administrator

To create or manage a Rational project, you use the Rational Administrator. To start the Administrator, click the **Start** button, then click **Programs > Rational Suite > Rational Administrator**.

Figure 2 The Rational Administrator



This chapter does not provide complete information about the Administrator. For more information, refer to the Administrator's online Help or to the book, *Using the Rational Administrator*.

Setting Up the Project

This section describes, at a high level, how to set up a Rational project. As you set up a project, if you need more details, please see the documentation for your commercial database, for an individual Suite product, or for the Rational Administrator.

High-Level Steps for Setting Up a Rational Project

To set up a Rational project, perform the following steps, which are described in the remainder of this section:

- 1 Upgrade any existing data.
- 2 Install and configure databases.
- 3 Decide where to locate the project.
- 4 Create the project.
- 5 Add one or more projects.
- 6 Add users, groups, and computers.
- 7 Attach existing Rose models.
- 8 Place data under configuration control.

Upgrading Existing Data

If you want your group's project to use data you've created with earlier versions of Rational tools, then you first need to upgrade the existing data. You may also need to upgrade the commercial database you are using. General steps are as follows:

- 1 Back up existing data. Make sure that you keep a safe copy of your data so that you can revert to it if you run into problems during the upgrade.
- 2 If necessary, upgrade the databases used for your data. Depending on the database you have decided to use, there may be a newer version available. For your convenience, we recommend that you upgrade your database before you create the project.

The remainder of this section describes steps to take for specific products.

Upgrading ClearQuest Data

To continue using data created in an earlier version of ClearQuest, upgrade by using the ClearQuest Maintenance tool. This tool:

- 1 Upgrades the schema repository.
- 2 Upgrades the user database.
- 3 Optionally moves the schema repository to a new database.

To start the ClearQuest Maintenance tool, click the **Start** button, then click **Programs > Rational Suite > Rational ClearQuest > Rational ClearQuest Maintenance Tool**. Refer to *Installing Rational Suite* for additional ClearQuest upgrade and configuration information.

Upgrading RequisitePro Data

To continue using data created in an earlier version of RequisitePro, upgrade by using the RequisitePro upgrade wizard. The wizard:

- Works with existing databases from RequisitePro 2.x and later.
- Makes a backup copy of the original database.
- Upgrades in place, overwriting the existing database (but not the backup copy).

The wizard starts when you open a RequisitePro project that contains data created with an older version of RequisitePro.

Upgrading Rational Test Data

The Rational Administrator provides a wizard which allows you to upgrade from Version 7 to Version 8; however, there is no conversion facility from Version 6.x to Version 8. To convert from Version 6.x to Version 8, you must convert to Version 7 and upgrade to Version 8 from there.

To convert from Version 6.x to Version 7, you must ensure that:

- Rational Suite 6.x is installed on the machine where you are creating the project.
- The SQA Suite repository is accessible from the machine where you are creating the Version 7 repository.

When you create the Version 7 repository, the Administrator converts the Rational 6.x data to its new format.

You can now run the wizard to convert the Version 7 data to Version 8.

After you create the project, you can remove SQA Suite 6.x from the system containing the project. For more information, please see “Upgrading an SQA 6.x Repository” in *Using the Rational Administrator*.

Upgrading Rose Data

To continue using data created in Rose 3.x, use the current version of Rose to open the model. Rose converts the model. Make sure that you save the model before closing Rose.

Installing and Configuring Databases

In the planning phase, you chose the commercial databases to work with (see Table 1 on page 28). It is now time to install and set up those databases, following these guidelines:

- This step is mandatory if you are using Sybase SQLAnywhere, Microsoft SQL Server, or Oracle. Install the database, then perform any set-up activities (such as setting privileges and tuning the database) appropriate for that database. For more information, refer to the documentation for the database and to documentation for the product with which you will use the database.
- If you are using Microsoft Access on a server, install it on the server before creating the project. Alternately, installing Rational Suite on any machine automatically installs Access Runtime. For Access, you do not have to perform database set-up activities before you create the project.

Deciding Where to Locate the Project

Determine where the project will reside, following these guidelines:

- The Test datastore must reside in the datastore directory structure. The disk that will contain the datastore must have enough space to contain at least the Test database.
- The project contains pointers to the other tools' databases. Therefore, these databases can reside anywhere on your network system.
- Before proceeding, create the top-level directory of the project. Make sure the directory is empty and make it a shared directory. If you do not share the directory, the project will not be available to members of your team.

Creating a Project

For information on how to create a project, see the book, *Administering Rational Suites*.

Adding Users and Groups

Configure Rational Test, ClearQuest, and RequisitePro as described in the following sections.

Rational Test

Use the Rational Administrator to add users and groups to the Rational Test datastore. If you initialized the Test datastore with data from a prior repository, the users, groups, and computers defined in the existing data appear in the new project.

ClearQuest

If you are using ClearQuest, use the ClearQuest Designer to set up user and group access to the ClearQuest database. Refer to the ClearQuest Designer online Help for more information.

RequisitePro

If you are using RequisitePro, you can use project security for the RequisitePro database. Refer to the RequisitePro online Help for more information.

Attaching Existing Rose Models

You can attach Rose models to a project after you create the project. This feature allows you to create models after you create the datastore.

- 1 From the Rational Administrator, right-click on a project.
- 2 On the shortcut menu, click **Attach Rose Model**.
- 3 Use the File Browse dialog box to select the model to use.

You cannot share a Rose model among more than one project. You can attach up to seven Rose models to a datastore project.

Placing Data under Configuration Control

We recommend that you use a configuration control program such as ClearCase to manage the datastore and the databases associated with the datastore.

Using Rational Suite without a Repository

We recommend that you use a repository to work with Rational Suite. It is possible to use some editions of Rational Suite, specifically those editions that do not use Rational Test, without a repository. Rational Test requires a repository.

To use Rational Suite AnalystStudio or Rational Suite DevelopmentStudio without a repository:

- 1 Set up ClearQuest, RequisitePro, and Rose models as if you were using them in stand-alone mode.
- 2 Use Rational Synchronizer without a repository, as detailed in *Rational Synchronizer* on page 49

Configuring the Integration for ClearQuest and RequisitePro

5

The integration between ClearQuest and RequisitePro allows you to associate qualified enhancement requests or defects with related requirements. After you collect stakeholder feedback as enhancement requests or defects, you validate, or *qualify*, the requests you want to implement. Then use the integration to establish associations between the originating enhancement request or defect and related requirements.

Working in ClearQuest, you can associate or create new requirements from enhancement requests or defects. Working in RequisitePro, you can view the original enhancement requests or defects on which the requirement is based, as well as establish new associations, and create new enhancement requests.

The integration allows you to track stakeholder input on an ongoing basis. This input reflects an important source of the change that occurs while we build our systems. Tracking these changes ensures that all qualified stakeholder input is addressed.

Overview: Setting Up the Integration

This section provides a summary of the steps that a Rational administrator should follow. These steps describe configuring Rational ClearQuest and RequisitePro to enable the integration. Once you perform these steps, the integration is accessible from each user machine. The majority of the administrative tasks involve configuration of Rational ClearQuest, so it is strongly recommended that the Rational tools administrator have ClearQuest administration experience.

The chapter describes three possible setup sections:

- **Initial Integration Setup**
If you have never used the Rational ClearQuest - RequisitePro integration, start with this section.
- **Upgrading Your Existing Integration**
Read this section if you have enabled the integration for previous versions of Rational tools.
- **Customizing the Integration**

Read this section to learn how to customize your integration.

Note: This chapter assumes that you installed Rational software on the C drive. Substitute your actual installation drive name, as needed.

Table 2 summarizes the steps for setting up the ClearQuest and RequisitePro integration. The remainder of this chapter describes these steps in detail.

Table 2 ClearQuest – RequisitePro Integration Configuration

Action	Comment
Create a Rational Administrator project	The ClearQuest/RequisitePro integration requires that you create a project in Rational Administrator. The project points to the ClearQuest database and the RequisitePro project that you are integrating.
Configure the ClearQuest database	No configuration is necessary if you used the out-of-the-box tools when creating your Rational Administrator project. For information on the out-of-the-box tools, see the related topic in the following section.
Configure the RequisitePro project or projects	No configuration is necessary if you used the out-of-the-box tools when creating your Rational Administrator project. For information on the out-of-the-box tools, see the related topic in the following section.
Edit and run the integration setup file	This file contains a description of your Rational environment, including file locations and names.
Upgrade the Associations	Perform this step only if you used the integration in a previous release and need to upgrade the Rational Administrator project, ClearQuest, and RequisitePro for logon information.

Initial Integration Setup

Creating a Rational Administrator Project

The ClearQuest database and the RequisitePro project must be contained in the same Rational Administrator project before the integration can be enabled. For more information on setting up a Rational Administrator project, see *Creating a Rational Project* on page 27.

Out-of-the-Box Tools

If you are enabling the integration for the first time, Rational provides some out-of-the-box tools that are included with your installation package to get you up and running quickly. Those tools include:

- A pre-configured Rational RequisitePro project structure
- Pre-configured Rational ClearQuest database schemas, including:
 - Rational Suite AnalystStudio schema
 - Rational Suite DevelopmentStudio schema
 - Rational Suite Enterprise schema
 - Rational Suite TestStudio schema.

These tools contain the unique artifacts that must be present to enable the integration. If you use the Rational-provided RequisitePro project template and one of the ClearQuest schemas when creating your Rational Administrator project, then the project configurations are complete. For more information on the out-of-the-box components, see *Rational Suite Out-of-the-Box Tools* on page 75.

Collecting Setup Information

To enable the integration in your specific environment, you must edit the ASCQISetup batch file. This file contains the command arguments shown in Table 4 below. Use this table to record the values of each variable for your project. You will use this information to edit the batch file. For a listing of all command arguments and descriptions used in the batch file, please see *ASCQISetup.exe Command Argument Descriptions* on page 42.

Note: If your installation spans multiple machines, you need to use Universal Naming Conventions (UNC) paths, as noted in the table below.

Table 3 Variables Describing Your Environment

Batch Command	Description	Must Edit	Your Values
set CQDatabaseName =	The name of the ClearQuest User database.	You must edit this entry.	

Table 3 Variables Describing Your Environment

Batch Command	Description	Must Edit	Your Values
set H =	The hard drive letter where RequisitePro is installed, followed by a colon (for example C:). This refers to the Administrator's installation, not the client installation.	The default value is "C". Only edit this entry if you installed RequisitePro on a drive other than the "C" drive.	
set U =	The Super User logon name for the ClearQuest database, the RequisitePro project, and the Rational Administrator project.	The default value for the Super User logon is "admin". If you are using a logon for RequisitePro, ClearQuest, or the Rational Administrator project other than "admin", edit this field. (The Super User logon refers to ClearQuest user privileges.)	
set P =	The path to the administrator's local Rational installation directory.	You must edit this field to display your administrator's installation location.	
set RP =	The name of the RequisitePro project as displayed in the RequisitePro Project > Open dialog.	You must edit this field to display the name of your RequisitePro project that you will integrate.	
set RQS =	The full directory path to the RequisitePro *.rqs file.	You must edit this field to indicate the correct path of the RequisitePro project. (If this is located on a different machine, use the UNC name in the path.)	

Table 3 Variables Describing Your Environment

Batch Command	Description	Must Edit	Your Values
set Help=	The full path to the ClearQuest - RequisitePro Integration help file (Program Files\Rational\RequisitePro\Help\CQI.hlp)	You must edit this field if you have located the CQI.hlp file in a location other than the default location listed in the ASCQISetup.bat file. (If this is located on a different machine, use the UNC name in the path.)	

Editing the ASCQISetup.bat file

Now use the information in the preceding table to adapt the ASCQISetup.bat file to your specific environment. The file is in the directory <Install>Program Files\Rational\Common. The ASCQISetup.bat file is designed to be run once *only* by the administrator and all settings in this file should reflect those of the administrator's installation. After the file has been run, it does not need to be run again unless one of the above variables changes. Edit and run this file once for each RequisitePro and ClearQuest project that you want to associate.

Begin by copying the ASCQISetup.bat file to another name, such as Original.bat, to preserve the original file. Edit the ASCQISetup batch file as follows:

- 1 Locate it in Windows Explorer,
- 2 Right-click the file, and select **Edit** from the context menu. A text editor opens, displaying the file.
- 3 Locate each `set` command and set the value to reflect the project-specific information you supplied in Table 3.

The following example of a modified ASCQISetup.bat file has been edited to work with the integration for the *Rational Suite Tutorial* data, which is used throughout the document *Getting Started with Rational Suite AnalystStudio*. This sample batch file reflects a local installation of RequisitePro and an unmodified help file. Remember to use UNC paths when necessary to reflect your installation environment.

Note: Fields that you must edit are in **bold** and are surrounded by angle brackets (<< >>). When editing to reflect your project specifications, replace the entire command including the angle brackets, with information about your administrator's project installation.

Note: The **REM** command precedes a comment.

```
REM Before you launch this batch file, please update the following
information:
```

```
REM **** INFORMATION TO UPDATE ****
```

```
REM Update the name of the ClearQuest database
```

```
set CQDatabaseName=<<CLASIC>>
```

```
REM Update the hard drive letter where your local Rational products are
installed
```

```
set H=<<C:>>
```

```
REM Update the SuperUser login name for ClearQuest database and
RequisitePro
```

```
set U=<<admin>>
```

```
REM Update the path to your local Rational installation directory
```

```
set P=<<\\<Server_Name>\Program Files\Rational\>>
```

```
REM Update the name of your RequisitePro project
```

```
set RP=<<Order Processing and Fulfillment System>>
```

```
REM Update the full name to your RequisitePro .rqs file.
```

```
REM Use a UNC path for a project on a non local machine
```

```
set RQS=<<\\<Server_Name>\Program Files\Rational\Classics Demo\
ClassicsRepo\Project\ClassicsPOS\Requisite\requisite.rqs>>
```

```
REM Update the path to the cqi.hlp file
```



```

REM Use a UNC path for a project on a non local machine
set Help=<<\\<Server_Name>\Program Files\Rational\RequisitePro
\help\cqi.hlp>>
REM **** BEGINNING OF THE SCRIPT ****

%H%
cd %P%\common

REM Create the association between the 'Defect' ClearQuest record type
and the RequisitePro attribute 'Defect' of the Feature requirement
type

ASCQISetup.exe -c %U% %CQDatabaseName% Defect Requirements_List
Defects_list RepoProject Modify "Associate Defects" "%Help%" 2 "%RP%"
"%RQS%" FEAT Defect 3

REM Create the association between the 'EnhancementRequest' ClearQuest
record type and the attribute 'EnhancementRequest' of the Feature
requirement type

ASCQISetup.exe -c %U% %CQDatabaseName% EnhancementRequest
Requirements_List EnhancementRequests_list RepoProject Modify
"Associate Enhancement Requests" "%Help%" 2 "%RP%" "%RQS%" FEAT
EnhancementRequest 3

pause

```

Note: If you have an unsecured RequisitePro project, it must contain the admin login specified in the ASCQISetup.bat file.

Executing the Batch File

Run the ASCQISetup batch file. As the batch file runs, it pauses every time it creates a setup record and prompts you to Press any key to continue. Press RETURN until the batch file ends.

Edit and run this file once for each RequisitePro project and ClearQuest database that you want to associate.

Your integration has now been enabled and you are ready to associate ClearQuest enhancement requests and defect records with feature requirements in RequisitePro.

ASCQISetup.exe Command Argument Descriptions

At the end of the ASCQISetup.bat file, there are two calls to the routine ASCQISetup.exe. The following tables list the parameters of ASCQISetup.exe in the order in which they appear in the command. Table 4 describes the Rational ClearQuest parameters; Table 5 describes Rational RequisitePro parameters.

Table 4 Rational ClearQuest Related ASCQISetup.exe Parameters

Command Line Argument	Description	Comments
%U%	Name of Super User in ClearQuest, a user in RequisitePro, and a user in the Rational Administrator project to which your ClearQuest database and RequisitePro project belong.	This variable is defined in the ASCQISetup.bat file. If the Super User has different accounts in each tool, you will be prompted to log into each tool as necessary.
%CQDatabaseName%	Name of the ClearQuest database in which you want to create associations to requirements.	This variable is defined in the ASCQISetup.bat file.
Defect	Name of the ClearQuest Record Type whose records you want to associate with requirements.	
Requirements_List	Name of the ClearQuest Record Type Field that stores references to associated requirements. This field must be of type Reference_List and reference the 'Requirement' Record Type.	
Defects_List	Name of the Back Reference Field defined for the previous field. This field automatically stores references of ClearQuest records associated with a requirement.	

Table 5 Rational RequisitePro Related ASCQISetup.exe Parameters

Command Line Argument	Description	Comments
RepoProject	Name of the ClearQuest Record Type Field that stores the Rational Administrator project containing the requirements you want to associate with the current ClearQuest record. This field must be of type Reference and refer to the Requirement Record Type.	
Modify	Name of an action of type MODIFY that operates on the ClearQuest_Record_Type_Name.	
"Associate Defects"	Title of the association dialog box that appears when the integration is called from RequisitePro. This title can mention, for example, the Record Type Name of the ClearQuest records that will be associated.	
%Help%	Full path to the help file used by both association dialog boxes, which are called when creating associations in either RequisitePro or ClearQuest.	This variable is defined in the ASCQISetup.bat file.
2	Context ID for the help topic that is called when a user clicks the Help button of the association dialog box that appears when the integration is called from RequisitePro.	
"%RP%"	Name of the RequisitePro project in which you want to create associations to ClearQuest records.	This variable is defined in the ASCQISetup.bat file.
"%RQS%"	Full path to the RequisitePro project file in which you want to create associations.	This variable is defined in the ASCQISetup.bat file.
FEAT	Prefix of the Requirement Type you want to associate with ClearQuest_Record_Type_Name.	
Defect	Name of the Attribute Field Type in which the list of associated ClearQuest record IDs is stored.	

Command Line Argument	Description	Comments
3	Context ID of the help topic that is called when a user clicks the Help button of the association dialog box that appears when the integration is called from RequisitePro.	

Upgrading Your Existing Integration

Each ClearQuest user database is associated with a schema database (also called a master database). Both databases must be configured to use the packages specific to the integration between ClearQuest and RequisitePro.

If you used the integration in an earlier version, you need only upgrade your ClearQuest databases to include additional packages.

Upgrading the ClearQuest Schema Database

To upgrade your existing schema database, begin by consulting the section on upgrading the database in the guide, *Administering Rational ClearQuest*. This document provides detailed recommendations about upgrading your schema database.

Note: Before upgrading, we strongly recommend that you back up the following data:

- ClearQuest schema database
- ClearQuest user database
- The entire tree of directories and files belonging to the Rational Administrator Project

To upgrade the ClearQuest schema database:

- 1 Use the ClearQuest Designer to log on to your schema database with an administrator profile. The Open Schema window appears. Click **Cancel**.
- 2 From the ClearQuest Designer menu, click **Package > Package Wizard**. The Package Wizard window appears.
- 3 In the Select a Package list, identify the latest version of the **RequisitePro** package (the package with the highest version number).
- 4 Click **More Packages**. If there is a **RequisitePro** package with a higher version number than the one you identified in Step 3, select it. Click **OK** to return to the Select a Package list.
- 5 Select the latest version of the **RequisitePro** package. Click **OK**.

- 6 In the Package Wizard, click **Next**. Select your schema. Click **Next** again.
- 7 Place a check mark in all record types that you will use in the integration. We recommend that you select at least **Defect** and **EnhancementRequest**. Click **Finish**.
- 8 From the ClearQuest menu, select **File > Checkin** to check this new package into your schema. Click **OK** after verifying your choice. Click **OK** again in the ClearQuest checkin confirmation dialog box.
- 9 Repeat this procedure to add the latest version of the **RequisiteProSupplement** upgrade package.

Upgrading the ClearQuest User Database

To upgrade the ClearQuest user database:

- 1 Back up your database before upgrading it.
- 2 From the ClearQuest Designer menu, select **Database > Upgrade Database** and click **Yes** to continue.
- 3 In the Upgrade Database window, select the database instance that is included in your Rational Administrator project and that has a previously integrated schema. Click **Next**.
- 4 Select the latest version of the schema (the schema with the highest version number). Click **Finish**.
- 5 Click **OK** and close the ClearQuest Designer.

Your ClearQuest databases are now upgraded to use the current ClearQuest/RequisitePro integration.

Executing the Batch File

The final step in upgrading the ClearQuest/RequisitePro integration is to run the ASCQISetup batch file.

- 1 Determine if you have the ASCQISetup.bat file that you customized for your previous RequisitePro/ClearQuest integration. If you have it, skip to step 3.
- 2 If you do not have a previous version of the ASCQISetup.bat file, edit the version that is supplied with this release, as described in "Editing the ASCQISetup.bat file" on page 39.
- 3 Log on to the ClearQuest User database as an administrator.
- 4 Delete all RequirementMap records.

- 5 Run the ASCQISetup batch file as described in "Executing the Batch File" on page 45.

Customizing Your Integration

The ClearQuest and RequisitePro integration is configured with the following default types:

- "Defect" and "EnhancementRequest" ClearQuest record types and
- "Feature" RequisitePro requirement type

You can include additional types to fit your development process by modifying the ClearQuest database schema and the RequisitePro project structure, and then editing the ASCQISetup.bat file to contain your new values in the command arguments. The following components can be customized:

- The RequisitePro requirement types
- The ClearQuest record types
- The RequisitePro requirement attribute labels

Changing the Default RequisitePro Requirement Types

The ASCQISetup.bat file contains a default requirement type of FEAT (Feature) to use in associating a requirement with a ClearQuest record. If your development team uses a different requirement type to track the product features, you can change this requirement type. For instructions on adding a new requirement type, see the RequisitePro online Help.

Changing the Default RequisitePro Project Attributes

If you add a new requirement type for use by the integration, you need to ensure that the requirement type contains a ClearQuest Integration attribute. This attribute displays the ClearQuest record ID associated with the requirement in RequisitePro.

To add the ClearQuest Integration attribute to a RequisitePro requirement type:

- 1 Back up your RequisitePro project before beginning the upgrade.
- 2 Launch RequisitePro and open the related Rational Administrator project.
- 3 From the RequisitePro Tool Palette click **Project > Properties**.
- 4 On the Attributes tab, from the Requirement Type list, click the requirement type you have selected to use for the integration. Click **Add**. The Add Attribute dialog box appears.

- 5 In the Label field, enter the name of your new attribute. (Remember this must exactly match the name of your customized ClearQuest record type.)
- 6 From the Type list, select **ClearQuest integration**. Click **OK**.
- 7 Repeat steps 3-6 to add an attribute for each ClearQuest record type that you want to associate in the integration.
- 8 Save and close the RequisitePro project.

The new attributes in RequisitePro are populated with ClearQuest record IDs when a requirement is associated to a ClearQuest record of that type. This makes it possible to track the origin of the associated requirement.

Changing the Default ClearQuest Records

The ASCQISetup.bat file contains two default ClearQuest records that the integration uses when the batch file is run:

- EnhancementRequest
- Defect

Both these records can be changed to suit your development environment. Begin by changing the record type names in ClearQuest, or by adding your own new record types. For more information on adding or changing ClearQuest records, please refer to the guide *Administering Rational ClearQuest* on your documentation CD.

Adding your Customizations to the ASCQISetup.bat file

After customizing your RequisitePro requirement types and attributes and your ClearQuest records, edit the entire ASCQISetup.exe command and update the requirement type and the attribute name. Replace **FEAT** with your requirement type, and **Defect** or **EnhancementRequest** with your customized attribute label or record. Copy and update this command for each requirement type and attribute name to include in the integration. Then run the ASCQISetup.bat file to activate the new values.

For more information about command line arguments for ASCQISetup.exe, please see *ASCQISetup.exe Command Argument Descriptions* on page 42.

The Rational Synchronizer helps maintain project consistency, resulting in improved team communication. The Synchronizer works by creating and updating a project's related items based on the existence of other items in your project.

For example, the Synchronizer might create a class based on the existence of a feature requirement. Or it might create an actor based on the existence of an actor requirement.

The Synchronizer works with the requirements (Rational RequisitePro) and modeling (Rational Rose) domains.

Overview of Using the Synchronizer

Generally, when you start the Synchronizer, it displays the Find Wizard. The Find Wizard helps you narrow the scope of items to work with. When you finish using the Find Wizard, the Synchronizer presents a list of items it proposes to create, update, or delete. You select only those items that you want to change. The Synchronizer carries out the changes for the items you chose.

The Synchronizer works by using rules to determine which changes are needed. Rational provides some sample rules to get you started. Your administrator or project leader analyzes your development environment to decide which rules to use, and whether to write new rules or to edit existing ones.

Example

For example, say that you want to ensure that every actor requirement in Rational RequisitePro has a corresponding Actor in Rational. The Synchronizer can enforce this constraint with a rule. When you run the Synchronizer it finds all actor requirements for which there is no Rose Actor. When you instruct the Synchronizer to proceed, it creates and initializes the test script.

As a result, new developments in the requirements domain are communicated accurately and promptly to the team responsible for testing.

Creating and Updating Items

You can use the Synchronizer to:

- **Create** items in one domain that are related to items in another domain or in the same domain.

When the Synchronizer creates items, it uses the work of one team to jump-start another team's efforts. For example, the Synchronizer can create test requirements in RequisitePro that correspond to Rose interfaces. As a result, the testing team is automatically notified of new work they need to reflect in test planning.

- **Copy** the values of attributes between items, allowing changes in one application to be visible in another. For example, the Synchronizer can cause a name change in a Rose use case to trigger a corresponding change to a related requirement in RequisitePro.

Copying attribute values ensures that changes in one item are propagated to related items. This feature eliminates error-prone manual updating.

In this Chapter

The remainder of this chapter discusses:

- Using the Synchronizer (read this section if you plan to use the Synchronizer to create or update items)
- Administration for the Synchronizer (read this section if you plan to set up a project to use the Synchronizer or if you plan to edit Synchronizer rules)
- Comparison with previous integrations (read this section if you have used earlier versions of Rational software)

Using the Synchronizer

You have the Rational Synchronizer if you have installed any edition of Rational Suite or any of the following tools:

- RequisitePro 4.0 or later
- Rose 98i, Rose 2000, or later

Starting the Synchronizer

As a user, start the Synchronizer by taking one of the following actions:

- From the RequisitePro menu, click **Tools > Rational Synchronizer**.
- From the Rose menu, click **Tools > Synchronize**.
- From the Windows **Start** menu, click **Programs > Rational Suite > Rational Synchronizer**.

As an administrator, you can also start the Synchronizer as follows:

- In the Rational Administrator, select a Rational project and click **Tools > Rational Synchronizer**.

Startup Context: What to Expect

If you have installed Rational Suite, your project administrator has probably used the Rational Administrator to define a *Rational project* for your team. A Rational project associates the following items:

- One Rational Test database
- Zero or one Rational ClearQuest databases
- One Rational Suite Synchronizer rules file
- One Rational RequisitePro database
- Zero, one, or more Rational Rose model files

If your work is part of a Rational project, in most cases when you start the Synchronizer, the Find Wizard immediately appears.

In some cases, the Setup Context dialog box appears first, prompting you to identify the Rational project to use. Log in to the project, using information supplied by your project administrator. The Find Wizard appears.

If you are not working with a Rational project, you cannot permanently identify the set of rules to use when synchronizing. When you start the Synchronizer, the Setup Context dialog box allows you either to use rules that are already defined or to create new rules.

Working with the Synchronizer

To use the Synchronizer, you first run the Find Wizard which helps you identify the items to work with. The Synchronizer displays a list of items based on the selections you make. Each of these items represents an action to create or update part of your project. You review the list to understand the impact of the changes. You select the items you want synchronized. The Synchronizer creates or updates, causing the selected items to become consistent across your project.

For more information about working with the Synchronizer, please see online Help for the Synchronizer.

Administration for the Synchronizer

If your team is using Rational Suite, administration tasks for the Rational Synchronizer consist of the following activities:

- Creating a Rational project for your team. (For a brief description of a Rational project, see page 51. For more detailed information about working with Rational projects, see the manual *Using the Rational Administrator*.)
- Importing rules for your project to use.
- Analyzing your project's development process to determine which Synchronizer rules you need.
- Editing existing rules or creating new rules.

If your team is not using Rational Suite, an administrator may identify which rules to use, but users need to import those rules every time they start the Synchronizer.

Importing Rules

How Rules are Stored

Synchronizer rules are text descriptions of Synchronizer actions. You can store one or more rules in a *rule definition file*, a text file with file type .rsd. The Synchronizer ships with sample rule definition files, located in:

<Install Path>\Common\OutOfTheBox\synchronizer rules\.

For your project, you will probably create new rules which you may base on the sample rules. You can edit a rule definition file with Notepad or another text editor. For more information, see *Writing Synchronizer Rules* on page 54.

To use a rule in the Synchronizer, you first import it into a *Synchronizer Rules Project*, a file with file type .rsr. Importing a rule converts it (compiles it) into the internal representation used by the Synchronizer. The Synchronizer Rules Project stores rules in their internal representation. If you edit a rule, you need to re-import it to make your changes effective.

Using Rational Suite: Importing Rules

If your team is working with Rational Suite, we recommend that you start from the Rational Administrator to import rules for your entire project. A placeholder for a Synchronizer Rules Project (a container for imported rules) is created automatically when you create a new project in the Rational Administrator. The file is actually created the first time you import rules and save the file. The Rules Project is named Rational Synchronizer Rules.rsr. It is created in the folder *project name*\Project*current project*\, where *project name* is the Rational project and *current project* is the project of interest.

To start the Synchronizer from Rational Administrator, click **Tools > Rational Synchronizer**. For each project, the first time you start the Synchronizer from the Rational Administrator, the Synchronizer displays the following prompt: `There are no rules defined. Would you like to import rule definitions? To import rules, select Yes.`

Working without Rational Suite: Importing Rules

If you do not have an edition of Rational Suite installed, you can still use the Synchronizer between domains. You first create the Rules Project file, then import rules into it. When you next start the Synchronizer, use the Setup Context dialog box to open the same Rules Project. In more detail:

Start the Synchronizer (from the Windows **Start** menu, click **Programs > Rational Suite > Rational Synchronizer**).

The Rational Synchronizer - Setup Context dialog box appears.

Select **Synchronize without using the Rational Project**, then select **Create a Synchronizer Rules Project**. Click **OK**.

In the Create Synchronizer Rules Project dialog box, specify a directory and file name for the Rules Project. Click **Save**.

The Synchronizer appears. Also, a message appears asking whether you want to import rule definitions. If you click **Yes**, you are prompted to select the rules to import, and then to save the rules project.

When you next start the Synchronizer, the Setup Context dialog box again appears. To use the same Rules Project, select **Synchronize without using the Rational Project**, then **Open an existing Synchronizer Rules Project**.

Writing Synchronizer Rules

This section describes how to design and write Synchronizer rules.

Understanding Your Development Process

Before you start to write rules, you need to analyze your development process to understand your group's needs. The following steps provide an overview of the analysis:

- 1 Write a narrative description of how your group works with items such as requirements, model elements, and test scripts.
- 2 From the description, identify the item types you work with and the domain they come from.
- 3 Select only those items that you want to relate to other items.
- 4 For each pair of items you want to relate, specify the action to take: create or update. Also describe the properties you want to synchronize, the direction of data flow, the timing of the synchronization, and conditions under which to synchronize.

From this information, you can create tables that serve as specifications for the rules you need to write.

A white paper, *Using the Rational Synchronizer*, elaborates on these steps and provides an example. This paper is available from your Rational technical representative.

What Can You Specify in a Rule?

You can write rules to perform the following actions:

- *item creation* – These actions create an item (for example, a RequisitePro requirement or Rose use case) that corresponds to another item.
- *data synchronization* – These actions transfer information between two corresponding items. You can use data synchronization actions only for two items that are already related.

You can write a rule that both creates an item and causes data synchronization. You can also write a rule that just creates an item.

This section presents an annotated Synchronizer rule. This rule is part of the Sample rule set supplied with the Synchronizer, described in the next section. For reference, we present the rule first without annotations.

Sample Rule without Annotations

```
# File:      actor_to_actor_requirement.rsd
# Purpose:   A sample rule for creating an Actor requirement in a Rational
#           RequisitePro domain project that corresponds to an actor in
#           a synchronized Rational Rose project.

#####
#
# Create an actor requirement from a Rational Rose actor element.
#
begin rule "Create Actor requirement from Rose actor"

description
{
Create an actor requirement in RequisitePro that corresponds to a Rational
Rose actor element.
}

    projects Rose to RequisitePro
    items Actor to "Actor Requirement Type"
    properties
        source.Name to target.Name
        source.Documentation to target."Brief Description"

end rule
```

Sample Rule with Annotations

```
# File:      actor_to_actor_requirement.rsd
# Purpose:   A sample rule for creating an Actor requirement in a Rational
#           RequisitePro domain project that corresponds to an actor in
#           a synchronized Rational Rose project.
```

You can insert comments anywhere in the rule definition file. A comment extends from a # character to the end of the line. Good housekeeping: Ensure that the file name describes the rule. Include the file name and an English-language description at the beginning of the file.

```
begin rule "Create Actor requirement from Rose actor"
```

Every rule starts with a begin clause. Give your rule a meaningful name so that it will be easy to remember what it does. If you use spaces in the name, enclose the name in quotes.

```
description
```

```
{  
Create an actor requirement in RequisitePro that corresponds to a Rational  
Rose actor element.  
}
```

Optionally provide a description of your rule. The description appears in the Synchronizer session window when you display detailed information about the rule. The description may be any length, but try to keep it short for easy viewing. Describe the purpose of the rule and the data transfer that it causes. Notice that the text of the description is enclosed in braces { }.

```
projects Rose to RequisitePro
```

Identify the projects to which this rule applies. In this example, Rose is the source project and RequisitePro is the target project. The projects are from one of the supported domains: modeling (Rational Rose) or requirements (Rational RequisitePro).

The source and target project may be the same, for example, if you are synchronizing use case requirements to test requirements within Requisite Pro. If the project name contains spaces, enclose the name in quotes.

items Actor to "Actor Requirement Type"

Specify the item types that this rule will synchronize. If an item name contains spaces, enclose the name in quotes. Item types are case sensitive. Spell them exactly as shown in the following table or as they appear in the domain project.

Project Type	Item Types
Rose	"Use Case" Actor Package "Sequence Diagram" Class
RequisitePro	Any requirement type. Type the name of the requirement (not the tag) exactly as shown in the Requirement Tab of the project properties display in RequisitePro.

properties

source.Name to target.Name

source.Documentation to target."Brief Description"

The properties statement describes one or more data transfers. There is one properties clause per rule.

Each transfer is defined by a pair of properties, or attributes, one in the source item and one in the target item. The data transfer may be from the source to the target or from the target to the source. In some rules, there may be data transfers in both directions.

For each transfer, the first property is the source of the data (the source item is called the broadcaster for this transfer). The second property is the target of the data (the target item is called the receiver for this transfer).

Attributes are case-sensitive. Spell them exactly as shown in the following table, including quotes, or as they appear in the domain project.

Item Type	Valid Properties
Rose "Use Case"	Name, Documentation, Stereotype, Rank
Rose Actor	Name, Documentation, Stereotype
Rose Package	Name, Documentation, Stereotype, Global
Rose "Sequence Diagram"	Name, Documentation
Rose Class	Name, Documentation, Stereotype, "Export Control", Cardinality, Space, Persistency, Concurrence, Abstract
RequisitePro Requirement	Any attribute for the requirement type. Enclose in quotes if the attribute contains spaces. At a minimum, all requirements have the following attributes: <ul style="list-style-type: none"> ▪ Text ▪ Tag ▪ Key (see Note below)

Note: For these attributes, Tag and Key are read-only; they cannot be set by a Synchronizer rule. You can set the Text attribute only if the requirement is database-based (not document-based).

Specify properties using the format **source.<property>** or **target.<property>** where source and target correspond to the definitions in the items clause. To use quotes, enclose just the property in quotes, for example, **target."Planned Iteration"**.

The keyword to between the two properties means that on the first application of the rule, the receiving field, will be created and initialized with the source data. On subsequent synchronizations the data will be re-transferred if the values are different.

The equivalent keywords **init** or **initialize** cause the data transfer to occur once, when the target item is initially created.

You can optionally create a traceability link from the source to the target or from the target to the source if allowed in the appropriate domains. Both the source and target items must be in the requirements domain. In this case, you can include one of these

- statements in the rule:
- trace source to target
- trace target to source

end rule

Every rule ends with an end clause.

Sample Rules Supplied with the Synchronizer

The Synchronizer includes sample rules which you can use as a starting point for your project. You may be able to use the rules as shipped or you can customize them to support your project's needs.

The sample rules are in the directory:

Install Path\Common\OutOfTheBox\synchronizer rules\.

These rules work with the out-of-the-box RequisitePro domain project. If you use a different RequisitePro project schema you will probably have to modify the sample rules or write your own.

Table 6 describes the rules supplied with the Rational Synchronizer.

Table 6 Sample Rules

File Name	Purpose
actor_requirement_to_actor.rsd Create a Rational Rose actor from an actor requirement	Create a Rational Rose actor model element that corresponds to a RequisitePro actor requirement. The name and documentation in the Rational Rose element are initialized from the requirement. After initialization, the name and documentation in the Rational Rose element are the master fields and are synchronized to the requirement on subsequent synchronizations.
actor_to_actor_requirement.rsd Create an actor requirement from a Rational Rose actor element	Create an actor requirement in RequisitePro that corresponds to a Rational Rose actor element.
class_to_tr_requirement.rsd Create Test Requirement from Rose Class	For each class model element in the modeling (Rose) domain, create a test requirement object in the requirements (RequisitePro) domain.
supl_requirement_to_tr_requirement.rsd Create Test requirement from Supplemental requirement	For a supplemental requirement, create a test requirement for use by TestStudio. This rule applies when the Test attribute on the SUPPL requirement is True.
uc_requirement_to_tr_requirement.rsd Create Test requirement from Use Case requirement	For a use case requirement, create a test requirement for use by TestStudio. This rule applies when the Test attribute on the UC requirement is True.

The Synchronizer Rule Template File

A Synchronizer rule template file provides a starting point for writing rules. The template describes rule syntax and gives guidelines for writing rules.

The Synchronizer Rule Template File is located in

<Install Path>\Common\OutOfTheBox\synchronizer rules\synchronizer_rule_template.rsd

Synchronizer Rule Grammar

For completeness, Table 7 presents the Synchronizer rules grammar. **Bold** text designates terminal symbols; regular text designates non-terminal symbols.

Table 7 Synchronizer Rule Syntax

Left Side	Right Side
RulesFile:	Specifications
Specifications:	Specification Specification Specifications
Specification:	ProjectSpecification RuleSpecification
ProjectSpecification:	Empty ProjectBegin ProjectClauses ProjectEnd
ProjectBegin:	begin project ProjectName
ProjectClauses:	ProjectClause ProjectClauses
ProjectClause:	Empty ApplicationSpec PathSpec
ApplicationSpec:	application AppName
PathSpec:	path PathName
ProjectEnd:	end project
ProjectName:	String
AppName:	String
PathName:	String
RuleSpecification:	Empty RuleBegin RuleClauses RuleEnd

Table 7 Synchronizer Rule Syntax (continued)

Left Side	Right Side
RuleClauses:	RuleClause RuleClauses
RuleClause:	Empty RuleDesc ProjectSpec ItemSpec PropertySpec Applies Trace
Empty:	
RuleBegin:	begin rule RuleName
RuleName:	String
RuleDesc:	Empty DescKeyword {Any-text-except-quote}
DescKeyword:	'desc' 'descr' 'description'
ProjectSpec:	projects ProjectName ProjectName projects ProjectName to ProjectName
ProjectName:	String
ItemSpec:	items ItemType ItemType items ItemType to ItemType
PropertySpec:	properties PropertyTransferList
PropertyTransferList:	PropertyTransfer PropertyTransfer PropertyTransferList
PropertyTransfer:	Property TransferMode Property
Property:	PropertyLocation '.' String
PropertyLocation:	source target

Table 7 Synchronizer Rule Syntax (continued)

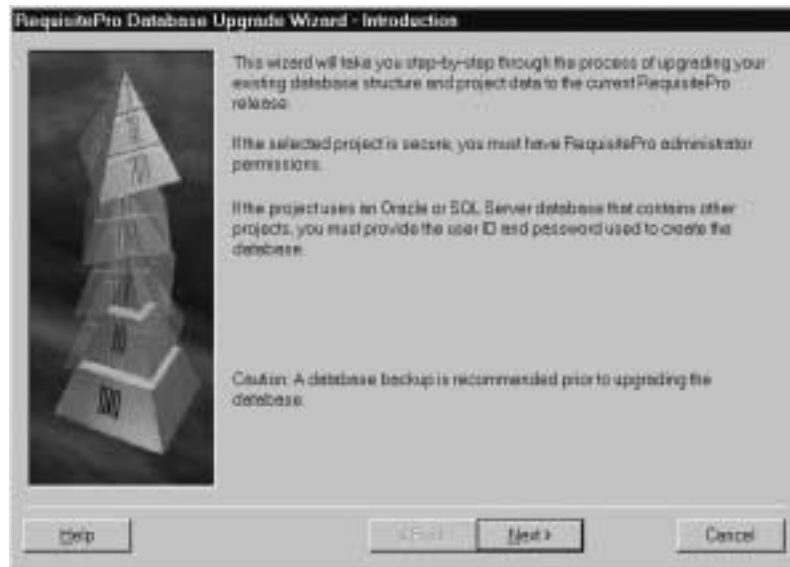
Left Side	Right Side
TransferMode:	init initialize to
Applies:	applies SourcePropertyName '=' PropertyValue applies when SourcePropertyName '=' PropertyValue
Trace:	trace source to target trace target to source
SourcePropertyName:	String
PropertyValue:	String
String:	Identifier "Any-text-except-quote"
Trace:	'trace source to target' 'trace target to source'
RuleEnd:	end rule

Converting the RequisitePro Database

You must convert older RequisitePro databases before you can use them with the new Suite Synchronizer.

When databases from previous versions are opened in with the new Rational RequisitePro application, RequisitePro automatically prompts the user to run the Database Converter Wizard.

Figure 3 Database Upgrade Wizard



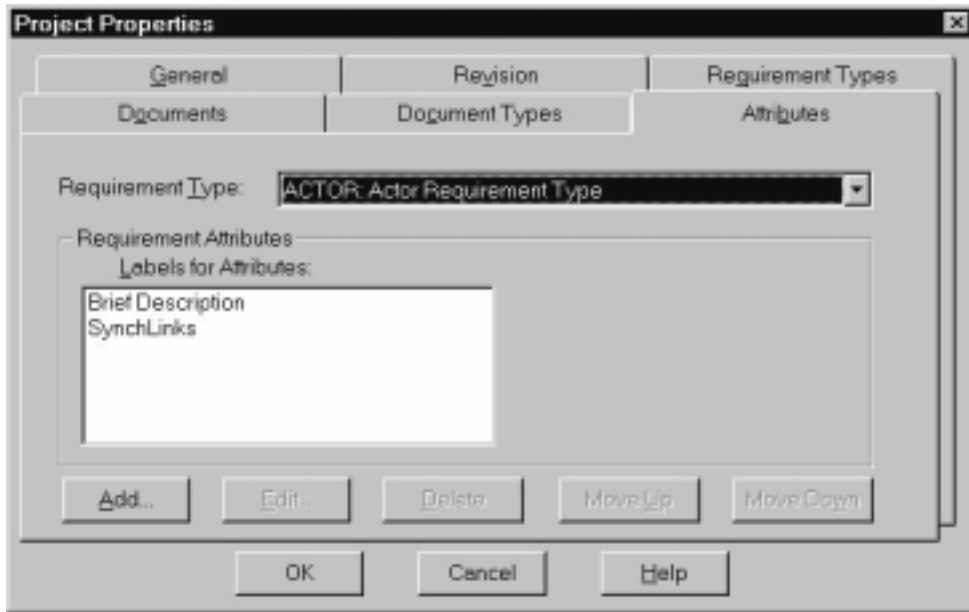
After the RequisitePro database is converted, you need to rename the SynchLinks attribute to SynchronizerSynchLinks:

Select **File->Properties** and select the **Attributes** tab

Edit the label for the SynchLink attribute for each Requirement Type that you want to Synchronize.

If the Requirement Type does not have a SynchLink attribute, create a SynchronizerSynchLink with type Text.

Figure 4 Project Properties Wizard



You need do nothing to convert the data within the SynchLink attribute. The data is converted automatically the first time you attempt a Synchronization.

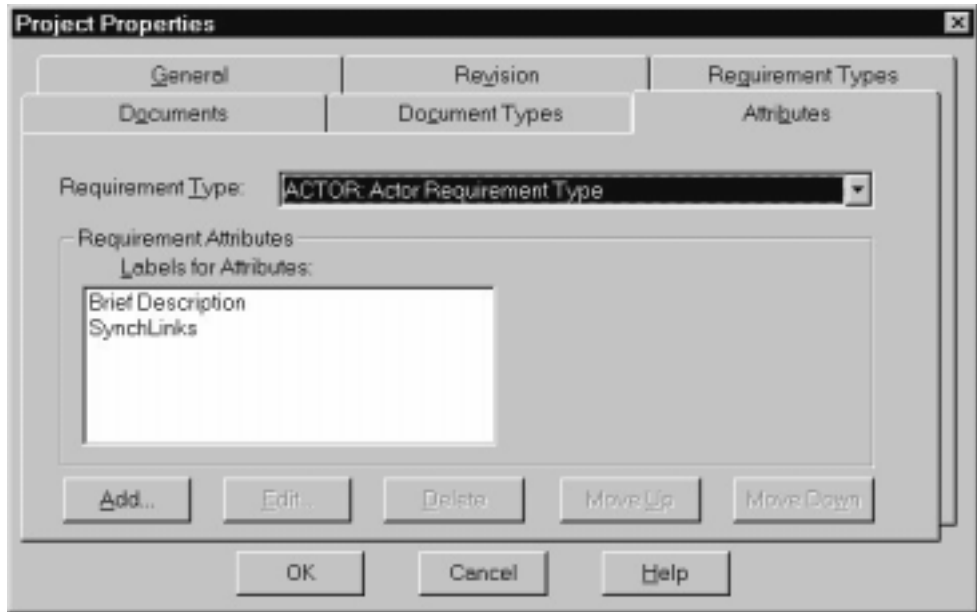
Preparing the RequisitePro Database for Synchronization

The Suite Synchronizer tracks links to and from requirements by storing data in a requirement attribute called SynchronizerSynchLinks. This attribute must be present in any requirement type used as a source or a target of a synchronization rule.

RequisitePro project templates already create this attribute in the most commonly synchronized requirement types but it is occasionally necessary to add the attribute manually.

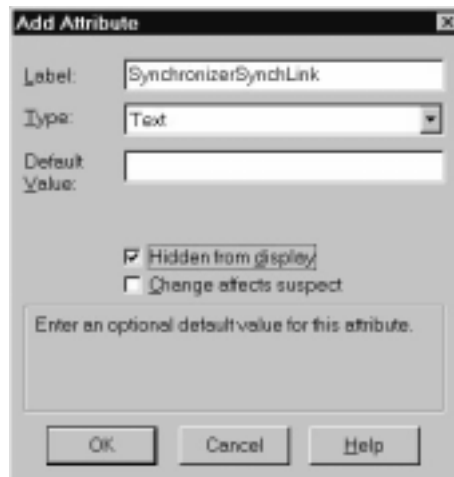
To access requirement types, select the **File->Properties** menu item, then select the **Attributes** tab. Use the **Requirement Type** drop down menu to select the requirement type that needs to be edited, and select **Add** to insert the SynchronizerSynchLink.

Figure 5 Project Properties Wizard



The SynchronizerSynchLink should be set to type Text with no default value. Normally, you should be sure the Hidden from display modifier is set.

Figure 6 Add Attribute Dialog



Converting Rose Models

You need not do anything to convert Rose models for use by the new Suite synchronizer.

Editing the Rules

You may also need to edit and reimport rules to remove references to Test Manager databases which the Suite Synchronizer no longer supports.

For More Information

See the chapter on *Migration and Conversion* on page 15 for high-level information on migrating and converting data from the prior releases to the current release of Rational Suite products.

See *Using Rational RequisitePro* and *Using the Rational Administrator* for more detailed information on converting and migrating data.

Extended Help gives you access to the Rational Unified Process and other information databases (including ones that you create). You typically start Extended Help from a Rational Suite tool, so you can get information about process and policy without interrupting your work flow. You can customize Extended Help to include topics related to your own process or environment.

The remainder of this chapter tells you how to use Extended Help and how to customize it and perform other administrative tasks.

Using Extended Help

This section describes how to get the most out of using Extended Help. It describes how to start Extended Help, how to change context, and how to add or remove databases.

Starting Extended Help

To use Extended Help:

- 1 Go to a Rational tool and select **Help > Extended Help** from the tool's menu.
The Extended Help browser appears, listing topics in a tree browser on the left.
- 2 To view a topic, navigate to, and double-click on, a topic title in the tree browser.

Changing Context

In Extended Help, the context determines how many, and which, topics Extended Help can display. You can check your current context by looking at the right side of the status bar. For example:

- The context `(Any, Any)` means that Extended Help is displaying all topics for the databases that are currently active.
- The context `(Rose, Any)` filters the topics to just those that are relevant to Rose.
- The context `(Rose, Use Case Diagram)` filters topics further so that they are relevant to Use Case Diagrams within Rose.

When you start Extended Help, a context is established by determining what tool and sub-area you started from.

Once you start Extended Help, you can reset the context to view more, fewer, or different topics. For example, say that Extended Help starts with the context (Rose, Use Case Diagram). You might want to change the context to (Rose, Activity Diagram), (ReqPro, Any), or even (Any, Any). To do so:

- 1 From the Extended Help menu, select **Edit > Context**.
- 2 Make selections from the **Product** list and the **Subtool** list. Notice that the value of **Product** affects the options listed under **Subtool**.
- 3 Click **OK**.

Adding or Removing Databases

A database contains the topics that you view in Extended Help. You can make more than one database available at any time. Changing the databases available changes the list of topics available to you. There are two ways to add or remove databases from Extended Help:

- Make a database active or inactive
- Add or delete Extended Help databases

Making a Database Active or Inactive

The effect of making a database active in Extended Help is to load all its topics into the Extended Help browser. Similarly, making a database inactive unloads its topics.

To make a database active or inactive:

- 1 From the Extended Help menu, select **Edit > Options**.
- 2 In the list of databases, locate the one you want to change.
- 3 Clear or select the check box in the **Active** column.

The change takes effect the next time you start Extended Help.

Adding or Deleting an Extended Help Database

When you add a database to Extended Help, you make its topics potentially available for viewing. (The database must also be active for you to see its topics.) Similarly, when you delete a database, you make its topics unavailable for viewing, but you don't physically delete the database from the disk. You can always re-add a deleted database back into Extended Help.

To add a database:

- 1 If you used this database in an earlier version of Rational Suite, or if you have not yet prepared the database for use with Extended Help, perform the steps listed in *Upgrading or Converting an Extended Help Database* on page 69.
- 2 From the Extended Help menu, select **Edit > Options**.
- 3 Click the **Add** button, then fill in the table's cells:
 - a Determine whether the database should be active, and if so, set the **Active** check box.
 - b In the **Name** column, enter a human-readable and unique name that will help you remember the purpose of the database.
 - c In the **Path** column, accurately enter the path to the database.

To delete a database:

- 1 From the Extended Help menu, select **Edit > Options**.
- 2 Select the row containing the database you want to delete.
- 3 Click the **Delete** button.
- 4 Click **OK**.

Administering Extended Help

This section describes administrative tasks to perform with Extended Help:

- How to upgrade or convert a database used with an earlier version of Rational Suite
- How to create an Extended Help database (this allows you to customize the information displayed by Extended Help)
- How to use translation strings to make database information more portable across multiple systems

Upgrading or Converting an Extended Help Database

Before you add a database to Extended Help, ensure that it is in the correct format. You need to perform the tasks described in this section if:

- You are creating a new database (described on page 71)
- You are upgrading from an earlier version of Rational Suite

To prepare a database for use with Extended Help:

- 1 Start with a properly structured file in comma-separated variable format (CSV):
 - See *Creating an Extended Help Database* on page 71 to learn how to structure this file.
 - If you are starting with a Microsoft Access database, read about *Converting from a Microsoft Access Database to a .csv File* on page 70.
- 2 Start the EHelpUpdate program. In Windows Explorer, go to *install directory\common*, where by default, install directory is *c:\Program Files\Rational*.
- 3 Double-click EHelpUpdate.bat. The EHelpUpdate window appears.
- 4 In the **Input** file field, type or browse to (click the ... button) the CSV file you've prepared.
- 5 In the **Output** file field, type or browse to a file whose type is .ehd, for example, *c:\temp\myExtendedHelp.ehd*
- 6 Click **OK**.
- 7 Continue with the steps in *Adding or Removing Databases* on page 68.

Converting from a Microsoft Access Database to a .csv File

If you used an earlier version of Rational Suite, you may have an existing Extended Help database. This database is most likely written in Microsoft Access. For this version of Rational Suite, you need to convert the Access database to a .csv file (a file in comma-separated values format) before you can use it.

To convert an Extended Help database from Access to a .csv file:

- 1 Go to a view of the database fields by opening the Access database and then opening the Help Table.
- 2 From the Access menu, select **File > Export**.
- 3 In the Export Table dialog box:
 - a Save the table to a filename with a .csv extension.
 - b Clear the **Save Formatted** check box.
 - c Click **Save All**.
- 4 The Export Text Wizard appears. On the first page of the wizard, select the **Delimited** export format.

- 5 On the next page, select the **Comma** delimiter to separate fields. Under **Text Qualifier**, select **{none}** or the quotation mark ("). Click **Finish**.
- 6 Use Microsoft Excel to open the resulting .csv file.
- 7 Select the third and fourth columns -- Columns **C** and **D**. These columns should be empty. Delete them by selecting **Edit > Delete** from the Access menu.
- 8 Save the file again, keeping the same .csv extension.
- 9 Continue with the instructions in *Upgrading or Converting an Extended Help Database* on page 69.

Creating an Extended Help Database

If you create an Extended Help database, users at your site will have easy, desktop access to information important to your project, for example, policies and procedures. In fact, you can create an Extended Help database containing any information accessible by a URL. In some cases, individual users create Extended Help databases to provide a quick reference to important information.

High-level Process

The high-level view of the creation process is:

- 1 Populate the database: use Microsoft Excel to create and enter pointers to the data (discussed in this section).
- 2 Add the database to Extended Help or tell your users to add it (described on page 68).
- 3 Tell your users how to use translation strings (described on page 73).

Populating the Database

To populate an Extended Help database:

- 1 Start Microsoft Excel.
- 2 For each database record, enter the following information in the first six columns:

Column	Field Name	Description
A	product	The name of the Rational product from which a query is invoked. See list of valid values on page 72.
B	subtool	The specific context within a product. See list of valid values on page 72.

C	query	This value becomes a folder name in the Extended Help browser. Think of it as a category for the information pointed to by this record, for example, "Requirements Traceability". You can use any value, up to 100 characters, but we recommend that you keep the length relatively short.
D	target	A URL pointing to the information for this record. The URL can be: <ul style="list-style-type: none"> ▪ a URL using an absolute value (example: http://www.rational.com) ▪ a URL using a translation string – described on page 73 (example: <corpinfo>\requirements\traceability.html) . ▪ a file name (example: file:d:\myfile.html) The maximum length of this field is 255 characters.
E	short_description	The description that appears for the record under the folder in the Extended Help browser. The maximum length of this field is 50 characters.
F	description	A longer description that appears when the cursor is held over the item (the information appears in a tool tip). The maximum length of this field is 255 characters.

- 3 Save the file in the comma-separated values (.csv) format (use **File > Save as**).
- 4 Continue by adding the database to Extended Help (described on page 68).

Valid Product and Subtool Values

The following table describes valid combinations for values of Product and Subtool. A database record must have one of these valid combinations in order for Extended Help to find the record.

Note: You can use the value **Any** for the subtool or for the product and subtool. A value of **Any** indicates that any context matches. For example, the values **(ReqPro, Any)** match **(ReqPro, Toolbar)** or **(ReqPro, View)**.

Product	Subtool
Rational ClearQuest	client
Rational ClearQuest	designer

Rational Test	Administrator
Rational Test	LoadTest
Rational Test	LogViewer
Rational Test	Robot
Rational Test	TestFactory
Rational Test	TestManager
Rational RequisitePro	Toolbar
Rational RequisitePro	View
Rational Rose	Activity Diagram
Rational Rose	Class Diagram
Rational Rose	Collaboration Diagram
Rational Rose	Deployment Diagram
Rational Rose	Other Tool
Rational Rose	Package Diagram
Rational Rose	Sequence Diagram
Rational Rose	State Diagram
Rational Rose	Use Case Diagram
Rational SoDA	tool

You can also use these products with the subtool, **Any** (for example, (*product, Any*)):

- Purify
- VisualPurify
- VisualQuantify

Working with Translation Strings

A translation string is a variable that stands for part of a URL. It makes your Extended Help database portable to multiple systems.

Say that in a geographically distributed project, there exist local copies of the HTML pages pointed to by the Extended Help database. If you use absolute URLs, each set of HTML pages must be saved in exactly the same location. By using translation strings, however, you can store the HTML pages in a location appropriate to each site.

For example, you might have a set of HTML pages that are all under a single directory, say `\myProject`. You could define a translation string, `<project>`. On various systems, `<project>` might point to:

- `c:\program files\myProject`
- `d:\temp\HTML pages\projects\myProject`
- `h:\myVOB\projects\myProject`

Then, when you create records in an Extended Help database, you can use a translation string to refer to a URL, for example: `<project>\index\index.htm`

Defining Translation Strings on User Systems

On a user's system, define a translation string as follows:

- 1 From the Extended Help menu, choose **Edit > Options**.
- 2 Click the **Translation Strings** tab.
- 3 Click **Add**. A new blank line appears.
- 4 Under **String**, type the name of the translation string, for example, `<rup>`.
- 5 Under **Replacement**, type the path that the string should translate to, for example, `file:c:\Program Files\Rational\RationalUnifiedProcess`. Make sure that you do not include a final slash.
- 6 Click **OK**.

Deleting a Translation String

To delete a translation string:

- 1 From the Extended Help menu, choose **Edit > Options**.
- 2 Click the **Translation Strings** tab.
- 3 Select a translation string to delete.
- 4 Click **Delete**, then click **OK**.

Rational Suite provides sample material to help you get started with Rational RequisitePro, Rational Rose, and the Rational Synchronizer.

- Rational RequisitePro contains project templates for creating new projects. These templates include requirement types and attributes that are required for some Rational Suite tool integrations.
- When you install Rational Suite, the installer places a sample Rose model in the directory *Install_Path*\Common\OutOfTheBox\rose frameworks. By default, the *Install_Path* is c:\Program Files\Rational.
- The OutOfTheBox directory also contains Synchronizer rules, which are described in *Rational Synchronizer* on page 49.

The remainder of this chapter describes the RequisitePro and Rose items in more detail. It also includes a brief description of the tool mentors that are included in the Rational Unified Process.

Rational RequisitePro Project Templates

RequisitePro provides project templates that include complete RequisitePro project structures that are consistent with the Rational Unified Process and enable you to integrate your RequisitePro project with other Rational Suite tools. You can use any of these project templates as a starting point for new RequisitePro projects.

Creating a Project Based on the Use-Case Template

We recommend that you use the Use-Case Template project template for creating your RequisitePro project. This template is ideal for implementing the Rational Unified Process. It is also designed for RequisitePro projects that utilize the RequisitePro integration with Rational Rose use cases and ClearQuest enhancement requests or other ClearQuest records.

The project template provides requirement types, suggested attributes and attribute values (including default values), and it defines related document types. You can use the template as provided, or modify it to meet your project's needs. The following table describes the structure of the project template:

Table 8 RequisitePro Use-Case Project Template

Document Type	Associated Document Outline	Default Requirement Type
Vision	RUP Vision Document	FEAT
Glossary	RUP Glossary	TERM
Use Case Specification	RUP Use Case Specification	UC
Supplementary Requirements Specification	RUP Supplementary Specification	SUPL
Requirements Management Plan	RUP Requirements Management Plan	RMP
Test Plan	RUP Test Plan	TC
Test Case Specification	Functional Test Case Specification	TC
Stakeholder Request	RUP Stakeholder Requests	STRQ

To create a project using the Use-Case Template, follow these steps:

- 1 Open the Create RequisitePro Project dialog box by doing one of the following:
 - From RequisitePro, click **Project > New** on the Tool Palette.
 - From Rational Administrator, in the Configure Project dialog box, click the **Create** button at the **Requirement Asset** field.
- 2 Select the **Use-Case Template** project template and click the **Create** button.
- 3 Refer to the online help, as needed, for entering the properties for the new project.

The Rational Rose RUP Model

The rose frameworks directory contains a Rational Rose model, called rup framework.mdl (a framework in Rational Rose is a set of predefined model elements needed to model a certain kind of system; for further information, see *Using Rose*). To use this model, copy and rename the model file as follows:

- 1 Locate the model in the installed directory,
Install Path\Common\OutOfTheBox\rose frameworks.
- 2 Copy the model into one of your project directories.
- 3 Rename the file as appropriate.

The model is designed to be self-explanatory. Each diagram and model element is documented in the documentation window, attached notes, or both. The structure of the model is designed to facilitate use of the Rational Unified Process.

Tool Mentors

Tool Mentors provide a link between the Rational Unified Process and Rational tools. A Tool Mentor tells you how to perform a Unified Process activity using a Rational tool. Tool Mentors are part of the Rational Unified Process and are available from both the Unified Process and from Extended Help.

Rational Suite provides you with additional resources and assistance, including technical documentation and support information.

Online Release Notes

Release notes for Rational Suite are located in the *Install Path\doc* directory.

Each product within Rational Suite provides its own set of release notes, included in *Install Path\doc*.

Error Message

EHRegister

```
There was an error opening the registry key
'HKEY_LOCAL_MACHINE\SOFTWARE\Rational Software\Rational
Suite\1.0\Extended Help\databases.' This program is unable to
continue.
```

You do not have Rational Suite installed properly on your system. Review the installation of Rational Suite and retry EHRegister.

Index

A

- Actor
 - Rose 58, 59
- Actor element 59
- actor requirement 59
- administration
 - Synchronizer 52
- attribute 75

B

- back up 19
- Backing up 15

C

- Class
 - Rose 58
- ClearQuest 16
- ClearQuest database 16, 18, 51
- configuration management 18
- convert 18
- convert RequisitePro database 62

D

- database
 - add or remove from Extended Help 68
 - convert 62
 - creating in Extended Help 71
 - make active or inactive in Extended Help 68
 - populate in Extended Help 71
 - upgrading or converting in Extended Help 69
- database relationships in Rational Suite 24
- datastore 16, 18
- DBMS (Database Management System) 25
- development process 52, 54

- domain 54

E

- element
 - Actor 59
 - model 59
- error messages 79
- Extended Help 67

I

- import rules 53
- importing rules 53
- item
 - relate 54
- item type 54

M

- messages, error 79
- model element 59
- model elements 54
- modeling domain 49

O

- Out-of-the-box
 - creating a RequisitePro project 75
 - Synchronizer rules 59
 - tools 75
- Overview of tasks performed by Rational Suite administrator 23

P

- Package Upgrade utility 22
- planning

- what to do before you plan 23
- planning your database 25
- project 21
- project schema 59
- project template 75
- Project Upgrade utility 22
- projects
 - Rational 51

R

- Rational Administrator 18, 19, 21, 51, 53
 - using 29
- Rational Administrator project 16, 17
- Rational Administrator RepoProject 17
- Rational ClearCase 18
- Rational ClearCase LT 18
- Rational ClearQuest database 18
- Rational ClearQuest schema package 21
- Rational project 27, 51
- Rational repository
 - planning the 27
 - setting up, high-level 29
 - what is 27
- Rational RequisitePro database 51
 - convert 62
- Rational RequisitePro project 18, 75
- Rational Rose model file 51
- Rational Rose RUP model 77
- Rational Suite Administrator tasks 23
- Rational Suite Synchronizer rules file 51
- Rational Test 16, 18
- Rational Test database 51
- Rational Test datastore 18
- Rational Unified Process 75
- relate items 54
- RepoProject 16
- repositories 16, 18
- repository 17, 21
 - adding one or more projects 32
 - adding users, groups, and computers 32
 - attaching existing Rose models 33
 - creating the 32
 - installing and configuring databases 31

- setting up, high-level 29
- upgrading ClearQuest data 30
- upgrading existing data 30
- upgrading Rational Rose data 31
- upgrading Rational test data 31
- upgrading RequisitePro data 31
- using Rational Suite without one 33
- requirement
 - actor 59
 - RequisitePro 58
- requirement type 58, 75
- requirements domain 49
- RequisitePro 16, 17
 - out-of-the-box project 75
- RequisitePro project 16, 18, 21
- RequisitePro requirement 58
- resources
 - online documentation 79
 - online release notes 79
 - support information 79
- Rose 16
- Rose Actor 58, 59
- Rose Class 58, 59
- Rose model 18, 75
- Rose models 16, 18
- rule definition file 52
- rules 52, 66
 - importing 53
 - sample 59
 - Synchronizer 49, 52
 - template file 60
 - writing 54
- running the Synchronizer 51

S

- schema
 - project 59
- schema package
 - Rational ClearQuest 21
- synchronized backup 19
- Synchronizer
 - administration of 52
 - out-of-the-box rules 59

- overview of using 49
- Rule template file 60
- rule template file 60
- rules 52
- rules files 59
- rules grammar 60
- running 51
- starting 51
- writing rules 54
- Synchronizer administration 52
- Synchronizer rules 49

T

- technical support 79
- template
 - Synchronizer rule file 60
- test scripts 54

- Tool Mentors 77
- translation string 74

U

- UCM 18
- UCM project 18
- Unified Change Management 18
- upgrade 15, 17, 21
- upgrade repositories to projects 18
- use case 58, 75
- using Rational Suite without a repository 33

V

- VOB storage 19

