Rational the e-development company™

WINDOWS

Configuring Rational Suite®

VERSION: 2001A.04.00

PART NUMBER: 800-024439-000

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

Rational®
the e-development company™

IMPORTANT NOTICE

COPYRIGHT

Part Number: **800-024439-000**

Version Number: **2001A.04.00**

PERMITTED USAGE

TRADEMARKS

Apache Software Foundation Notice

PATENT

Portions covered by U.S. Patent Nos.5,193,180 and 5,335,334 and 5,535,329 and 5,835,701 and 5,574,898 and 5,649,200 and 5,675,802.

U.S. Patents Pending.

International 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

This manual describes how to set up a Rational Suite environment 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 lifecycle. Rational Suite's unparalleled level of integration improves communication both within teams and across team boundaries, reducing development time and improving software quality.

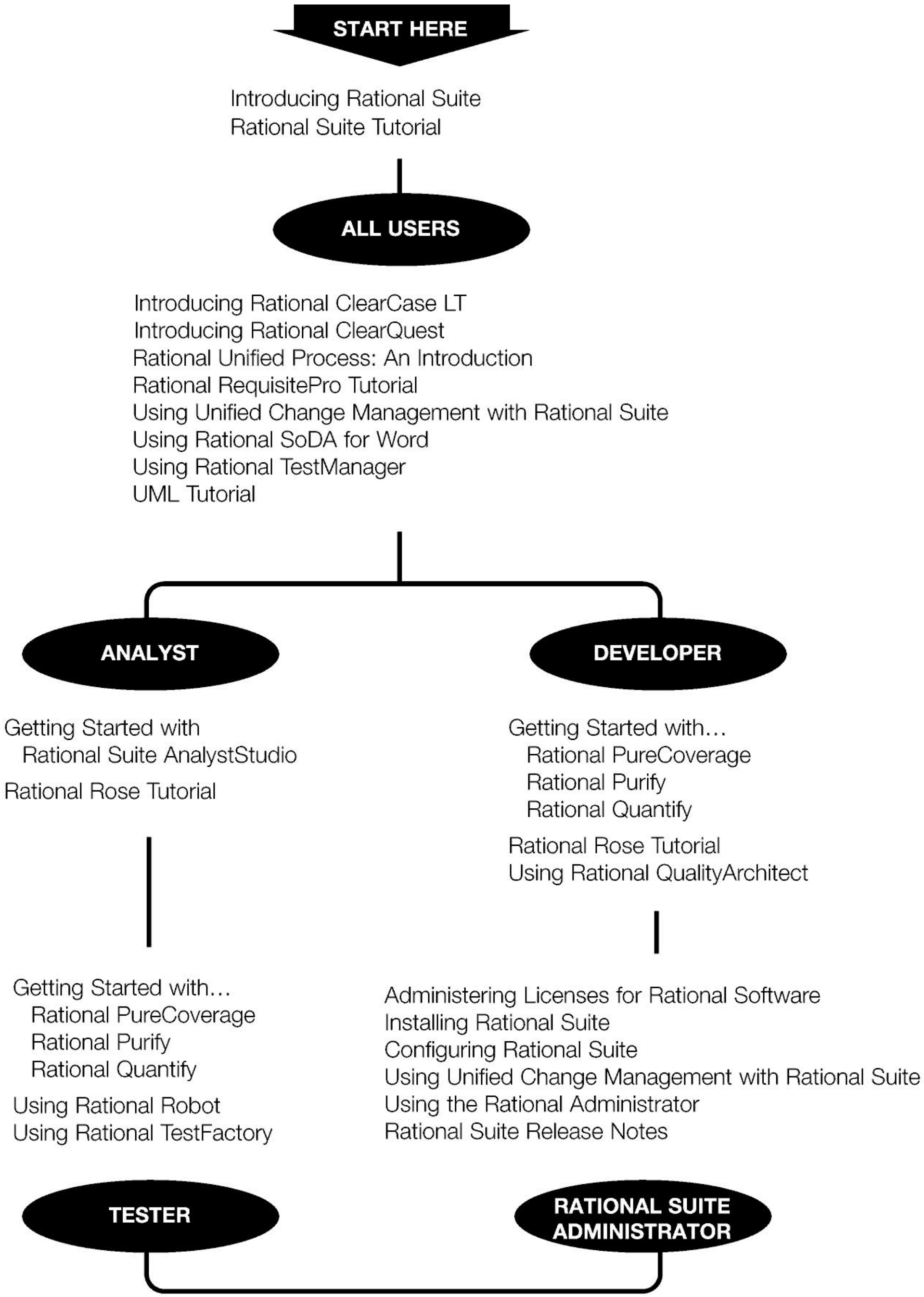
Audience

This guide is intended for Rational Suite administrators.

Other Resources

- 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`.
- For articles, discussion forums, and Web-based training courses on developing software with Rational Suite products, join the Rational Developer Network by selecting **Start > Rational Suite > Logon to the Rational Developer Network**.

Rational Suite Documentation Roadmap



START HERE

Introducing Rational Suite
Rational Suite Tutorial

ALL USERS

Introducing Rational ClearCase LT
Introducing Rational ClearQuest
Rational Unified Process: An Introduction
Rational RequisitePro Tutorial
Using Unified Change Management with Rational Suite
Using Rational SoDA for Word
Using Rational TestManager
UML Tutorial

ANALYST

Getting Started with
Rational Suite AnalystStudio

Rational Rose Tutorial

DEVELOPER

Getting Started with…
Rational PureCoverage
Rational Purify
Rational Quantify

Rational Rose Tutorial
Using Rational QualityArchitect

Getting Started with…
Rational PureCoverage
Rational Purify
Rational Quantify

Using Rational Robot
Using Rational TestFactory

TESTER

Administering Licenses for Rational Software
Installing Rational Suite
Configuring Rational Suite
Using Unified Change Management with Rational Suite
Using the Rational Administrator
Rational Suite Release Notes

RATIONAL SUITE ADMINISTRATOR

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, company name, telephone number, and e-mail address
- Your operating system, version number, and any service packs or patches you have applied
- Product name and release number
- Your case ID number (if you are following up on a previously reported problem)

Overview of Rational Suite Administration

1

This chapter provides an overview of the typical tasks that an administrator needs to perform to set up a Rational Suite environment. The remainder of this book describes these topics in more detail and provides pointers to other resources.

Installing Rational Suite for the First Time

When you install Rational Suite for the first time, you perform these tasks:

- Plan network resources.
- Configure databases.
- Configure licenses.
- Install Rational Suite software on servers.
- Configure servers and files.
- Configure integrations.
- Install Rational Suite software on clients.

See *Setting Up a Rational Suite Environment* on page 17 for more information about these topics.

Upgrading from an Earlier Release

When you upgrade from an earlier release of Rational Software, you need to perform updates to related software, including databases and datastores for Rational products. 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.
3. Update the data contained in the database.

Perform these upgrades as part of installing a new version of Rational Suite software. For more information, see *Installing Rational Suite*, which contains chapters about upgrade procedures. *Configuring the Integration for ClearQuest and RequisitePro* on page 27 also contains some information about upgrading that is specific to the integration between ClearQuest and RequisitePro.

Creating a Rational Project

A Rational project establishes connections among some of the products in Rational Suite and the data in those products. After you establish these connections, you can take advantage of many of the integrations among Rational Suite products.

You use the Rational Administrator to create a Rational project.

See *Creating a Rational Project* on page 23 for more information about this topic.

Configuring Integrations

You need to perform some additional work to set up or customize certain integrations:

- Integration between Rational RequisitePro and Rational ClearQuest

  The integration between ClearQuest and RequisitePro allows you to associate enhancement requests or defects with their related requirements. See *Configuring the Integration for ClearQuest and RequisitePro* on page 27.

- 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, see *Working with and Configuring Extended Help* on page 43.

- Rational Synchronizer

  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, see *Working with and Configuring the Rational Synchronizer* on page 53.

Troubleshooting

For pointers to troubleshooting information of interest to administrators, see *Troubleshooting* on page 67.

Setting Up a Rational Suite Environment 2

This chapter describes the tasks you need to perform to set up a Rational Suite environment when:

- You are installing Rational Suite for the first time on your network.
- You are upgrading from an earlier version of Rational Suite.

This chapter does not provide instructions for performing each task; rather, it gives you an overview of the tasks and tells you where you can find more information.

Installing Rational Suite for the First Time

When you install Rational Suite for the first time, you perform these tasks:

- Plan network resources.
- Configure databases.
- Configure licenses.
- Install Rational Suite software on servers.
- Configure servers and files.
- Configure integrations.
- Install Rational Suite software on clients.

Planning Network Resources

During planning, you work in these areas, as described in this section:

- Planning for UCM (Unified Change Management)
- Designating Servers
- Developing a database plan

Planning for UCM

Unified Change Management is Rational Software's approach to managing change in software system development from requirements to release. Configuring a network to work with UCM and Rational Suite requires special considerations. If you plan to use UCM in your development environment, see *Using Unified Change Management with Rational Suite*.

Designating Servers

Before you install Rational Suite, you need to determine which computers will be servers. Servers need to be highly available, and may need extra memory and disk space. Table 1 describes types of servers you may need. Note that you may be able to combine several servers on one computer.

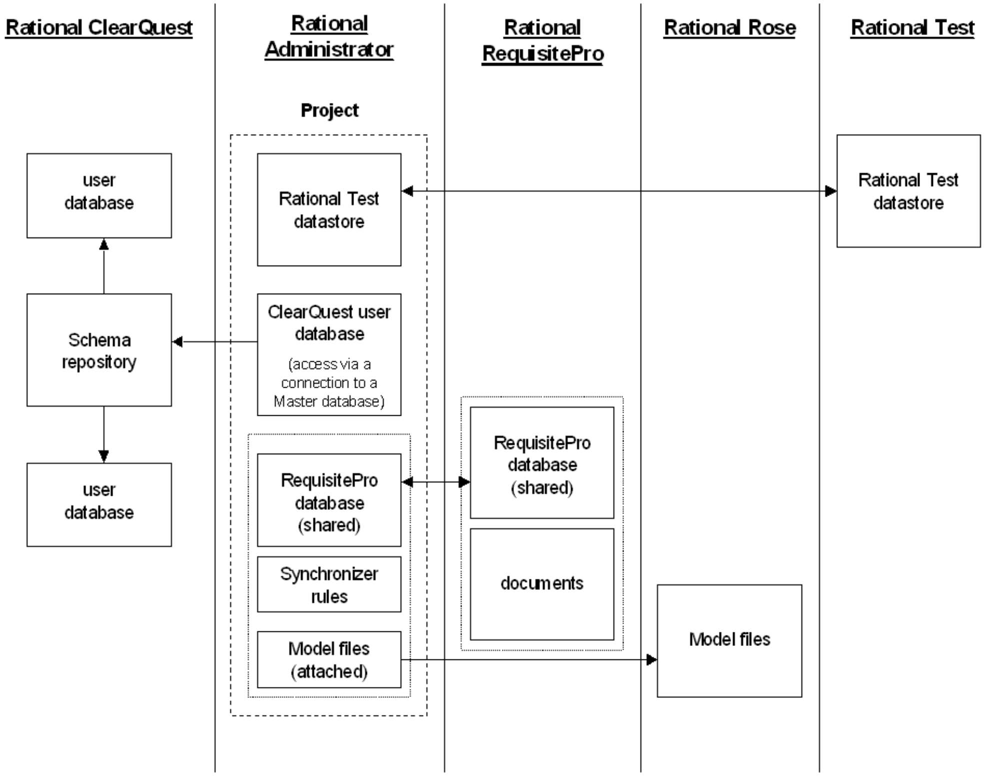
Table 1 Servers Used with Rational Suite

Type of Server	Description
License server	If you use floating licenses, you need to set up a server to administer the licenses. For more information, see *Administering Licenses for Rational Software.*
Web server	Some Rational tools include a Web component. You can set up a Web server for these tools: ▪ ClearCase LT. See *Administering Rational ClearCase LT.* ▪ ClearQuest. See *Installing Rational ClearQuest*. ▪ RequisitePro. See *Installing Rational RequisitePro.* ▪ Rational Test. See *Installing Rational Testing Products*.
Database server	If you are using commercial databases with Rational Suite products, you probably want to install those databases on dedicated servers. See *Understanding Database Relationships* on page 19.
ClearCase LT server	ClearCase LT is a client/server product. To use a ClearCase LT client, a ClearCase LT server must be installed in the same network domain as the clients. A client can work with exactly one server. For guidelines about selecting a ClearCase LT server system, see *Installing Rational ClearCase LT.*
ClearQuest administrator's system	To use ClearQuest, you need to designate one computer as the administrator's system. See *Installing Rational ClearQuest*.

Understanding Database Relationships

Before setting up Rational Suite on a network, take the time to plan how you will configure databases and datastores to hold the data produced by Rational products. Figure 1 describes the logical relationships among the databases used by Rational Suite products.

Figure 1 Rational Suite Database Relationships



Notes concerning the figure:

- Under ClearQuest, the arrows between the schema repository (Master database) and the two user databases indicate logical connections; the information in the user databases is a subset of information in the schema repository.
- Under Rational Administrator, the arrow from the ClearQuest schema repository to the ClearQuest user database indicates access to the user databases through the schema repository.

- The arrow between the two RequisitePro databases (between Rational Administrator and RequisitePro) indicates that the two boxes represent the same database.
- The arrow between the two Rational Test datastores (between Rational Administrator and Rational Test) indicates that the two boxes represent the same datastore.
- The arrow between the two model files (between Rational Administrator and Rational Rose) indicates that the two boxes represent the same file.

Developing a Database Plan

For each Suite product you are using, you need to decide which database management system or systems to use with Rational Suite products. You can use different databases for different Suite products. Table 2 summarizes the available choices based on the size of the user group planned for each product; refer to product documentation for more details about each choice.

Table 2 Commercial Database Choices

	ClearQuest	**RequisitePro**	**Rational Test**
Microsoft Access	Small groups, fewer than 5 users	Small or medium groups, fewer than 10 users	Small groups, fewer than 5 users
Sybase SQL Anywhere Server	Small to medium groups, 5-20 users	N/A	N/A
Microsoft SQL Server	Large groups, more than 20 users	Large groups, more than 20 users	N/A
Oracle	Large groups, more than 20 users	Small to medium groups, 10-20 users	N/A
IBM DB2	Large groups, more than 20 users	N/A	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 computer. (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.

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

Configuring Databases

For the Rational products that work with commercial databases, install and configure those databases before installing Rational products. *Installing Rational Suite* provides high-level descriptions of working with commercial databases, and lists other tasks to perform before you install Rational products.

Configuring Licenses

Rational Suite uses Rational Common Licensing Services. Before installing Rational software, decide what types of licenses you want to use and, if necessary, configure licenses for the software you will install. For more information about licensing, see *Administering Licenses for Rational Software*, included with your Rational Suite media kit.

Installing Rational Suite Software on Servers

Use the Rational Software Setup program to install Rational Suite software on your network’s servers. The Rational Software Setup program allows you to install Rational Suite and other Rational Software products. The Setup program uses a consistent and familiar installation wizard.

Installing Rational Suite, included with your Rational Suite media kit, contains details about using the Setup program, and also describes prerequisites, tasks to perform before installing, and tasks to perform after installing.

For assistance while using the program, see the Rational Windows Setup online Help.

Configuring Servers and Files

After you install Rational software, perform other tasks to configure the servers, including Web servers, and files that you will use with Rational tools. Table 1 on page 18 lists the types of servers that you may need to configure. *Installing Rational Suite* provides procedures to follow after you install Rational software.

Configuring Integrations

Configure integrations that you plan to use, as described in Table 3.

Table 3 Rational Suite Integrations

Integration	Comment
Rational project	A Rational project associates datastores from Rational products, enabling integrations across Rational Suite. For an overview, see *Creating a Rational Project* on page 23. For a more in-depth treatment, see *Using the Rational Administrator.*
UCM	UCM is a process related to configuration management. Tool support for UCM is included in Rational ClearCase and Rational ClearQuest. See *Using Unified Change Management with Rational Suite.*
UCM and ClearQuest	To learn about requirements for adding ClearQuest to a UCM-enabled project, see the "Enabling ClearQuest for UCM" chapter in *Administering Rational ClearQuest*.
RequisitePro and ClearQuest	This integration allows you to link defect and enhancement requests to requirements, thus allowing you to trace back to a report's origin. See *Configuring the Integration for ClearQuest and RequisitePro* on page 27.
Extended Help	Extended Help, available from all Rational Suite tools, allows you to view relevant topics in the Rational Unified Process, depending on the context from which you start. You can also customize Extended Help to include information relevant to your team. See *Working with and Configuring Extended Help* on page 43.
Synchronizer	The Synchronizer helps you keep items in Rose and RequisitePro consistent with each other in an automated way. See *Working with and Configuring the Rational Synchronizer* on page 53.
Integrations with RequisitePro	To learn about configuring integrations that work with RequisitePro, see the "Integrating with Other Products" chapter in *Using Rational RequisitePro*.

Installing Rational Suite Software on Client Computers

After you set up the server environment for Rational Suite, you can install Rational software on client computers. Follow the steps in *Installing Rational Suite*, paying attention to steps to follow before and after you install.

Creating a Rational Project 3

A Rational project is a logical collection of databases and datastores that associates the data you use when working with Rational Suite. A Rational project associates Rational Test datastores, RequisitePro projects, ClearQuest databases, and Rose models, and optionally places them under configuration management.

After you create a Rational project, you can take advantage of many of the integrations among Rational Suite tools, for example:

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

This chapter describes how to use the Rational Administrator to create a Rational project.

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 the following items:

- Zero or one RequisitePro project
- Zero or one Test datastore
- Zero or one ClearQuest database
- Zero to seven Rose model files

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.

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**. The Rational Administrator appears.

For complete information about using the Rational Administrator, 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, 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, described in the remainder of this section:

1. Decide where to locate the project.
2. Create the project.
3. Configure security.

Deciding Where to Locate the Project

Determine where the project will reside, following these guidelines:

- The Rational project contains pointers to the 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.
- If you are using UCM, there may be special considerations for locating the Rational project. See *Using Unified Change Management with Rational Suite*.

Creating a Project

To start creating a project, from the Rational Administrator menu, select **File > New Project**. A wizard appears, guiding you through the creation process. When you specify the project location, make sure that you use a UNC pathname, which is a pathname in the format \\*computer**share*.

When you have finished creating the Rational project, you have the option of configuring it. During configuration, you associate Rational datastores with the project.

For more details about creating a project, see *Using the Rational Administrator.*

Configuring Security

You can set some security parameters through the Rational Administrator. However, you must set most parameters through the individual point products (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.

Table 1 provides an overview of how to set up security in each Rational tool.

Table 1 Summary: Setting Up Security for Rational Tools

Tool	**Comments**
Rational ClearQuest	The ClearQuest administrator uses the ClearQuest Designer to set up user and group access to the ClearQuest database. See *Using Rational ClearQuest*.
Rational RequisitePro	The RequisitePro administrator sets up the RequisitePro users and groups by using project security for the RequisitePro database. See *Using Rational RequisitePro.*
Rational Rose	To place security on Rational Rose, which does not support user lists, secure the directory containing the Rose models.
Rational Test	Use the Rational Administrator to add users and groups to the test datastore. See *Using the Rational Administrator.*

Configuring the Integration for ClearQuest and RequisitePro

4

This chapter provides the steps for configuring the integration of Rational ClearQuest and Rational RequisitePro. The majority of the administrative tasks involve configuration of ClearQuest, so we strongly recommend that the Rational tools administrator have ClearQuest administration experience.

When configuring the integration, you perform the following steps:

- Configure a ClearQuest schema
- Configure a RequisitePro project
- Configure a Rational project
- Set up the integration mapping

If you are enabling the integration for the first time, Rational provides components to get you up and running quickly. These preconfigured components are:

- Rational ClearQuest database schemas
- Rational RequisitePro project templates

In addition, this chapter presents instructions for integrating customized ClearQuest schemas and RequisitePro projects.

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

Configuring a ClearQuest Schema

No schema configuration is necessary if you used one of the following Rational Suite schemas supplied with this release of your software when creating your ClearQuest user database:

- Rational Suite AnalystStudio schema
- Rational Suite DevelopmentStudio schema
- Rational Suite Enterprise schema
- Rational Suite TestStudio schema

If you used one of these schemas to create your user database, skip to *Configuring a RequisitePro Project* on page 32.

Upgrading the ClearQuest Schema Database

If your schema is not based on one of the Rational Suite schemas provided in the current release, you must apply the RequisitePro package in order to enable the ClearQuest/RequisitePro integration. This package adds fields, forms, and scripts to the schema to support the integration.

You can only apply this package to schemas that were created in ClearQuest release 2000.02.10 or later. For earlier versions, you must first upgrade to at least release 2000.02.10.

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 project

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

After you upgrade, continue with the steps in this section. If you have already performed a step as part of the upgrade, you do not need to repeat that step when you encounter it in this section.

Verifying Previous Packages

Before applying the package, you must verify which packages have been applied to your schema.

1. Launch the Rational ClearQuest Designer from the **Start** menu. Log on to your schema database with an administrator profile. The Open Schema window appears. Click **Cancel**.
2. Click **View > Schema Summary**. The Schema Summary dialog box appears.
3. Open the appropriate version of the schema in the folder.

4 In the **Associated Packages** list, note whether the RequisiteProSupplement and Repository packages are present:

- If the RequisiteProSupplement package is *not* present, you must apply the **RequisitePro 1.5** package in the following procedure.
- If any version of RequisiteProSupplement package is present, you must apply the **RequisitePro 1.5 Upgrade** package in the following procedure.
- If the Repository package is not present, you must apply the packages **Repository 0.2,** and then the latest version of the **Repository** package in the following procedure.
- If the Repository package is present, you must apply the latest version of the **Repository** package in the following procedure.

5 Click **OK** to close the Schema Summary dialog box.

Applying the Repository and RequisitePro Packages

To apply the Repository and RequisitePro packages:

1 From the ClearQuest Designer menu, click **Package > Package Wizard**. The Package Wizard window appears.

2 In the **Select a Package** list, locate the latest version of **Repository** package.

Note: If it is not present in the list, click **More Packages**. Click **OK** to return to the **Select a Package** list.

3 Select the appropriate version of the **Repository** package. Click **OK**.

4 In the Package Wizard, click **Next**. Select your schema. Click **Next** again**.**

5 Place a check mark in all record types that you will use in the integration.

Note: If a previous version of the Repository package has been applied, some record types in your schema may not appear because they have already been selected. We recommend that you select at least **Defect** and **EnhancementRequest**, if they are present.

6 Click **Finish**.

7 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.

8 Repeat this procedure for additional versions of the **Repository** package, if necessary, and the appropriate **RequisitePro** package (as determined in the previous procedure *Verifying Previous Packages*).

Adding Required Schema Entries

The ClearQuest/RequisitePro integration requires a Back Reference field on your Requirements_List field for each of the record types you selected when you applied the RequisitePro package, such as Defect and Enhancement Request.

To add a Back Reference field to your schema, do the following:

1. In the ClearQuest Designer, click **File** > **Open Schema**. The Open Schema dialog box appears.
2. Select the option **Checkout a schema to edit**.
3. Select your schema from the list and click **Next**. Click **Finish**.
4. In the schema browser, open the Record Types folder.
5. In a record type you selected when you applied the RequisitePro package, double-click **Fields**. The list of fields appears.
6. Right-click the **Requirements_List** field name and select **Field Properties**. The properties dialog box appears.
7. At the **Back Reference** field, type a unique name, such as "<Name_of_Record_Type>_List". For some schemas, this field may already contain a value.

   Note: If you enter a name at this field, you must edit the Back Reference field name in the ASCQISetup.exe command, as described in *Adding Your Customizations to the ASCQISetup.bat file* on page 39.
8. Close the properties dialog box.
9. Repeat from step 5 for each of the record types you selected when you applied the RequisitePro package.
10. Click **File > Check In**. ClearQuest validates the schema and displays a Check In dialog box.
11. Type a comment describing the schema changes and click **OK**.

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 project. 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 database is now upgraded to use the current ClearQuest/RequisitePro integration.

Maintaining Record Visibility

The ClearQuest record visibility feature allows you to limit access to certain ClearQuest records. This limited access applies to selected ClearQuest user groups, and is based on criteria defined by an administrator. The criteria specify that records are visible for a user group depending on the value of a specific ClearQuest reference field. To use the ClearQuest/RequisitePro integration, RequisitePro users must have record visibility throughout the life of all integrated records.

We recommend that you define record visibility for user groups based on the value of the RAProject field, which is required for integration with RequisitePro. If you use this criterion, you preserve the ClearQuest/RequisitePro integration, because you cannot change the value of the RAProject field without first removing all associations to RequisitePro requirements. In addition, this criterion ensures that all RequisitePro project users have visibility for the same ClearQuest records.

Using the Integration with ClearQuest MultiSite

When you add a ClearQuest database to a Rational project, the Rational Administrator creates a record in the ClearQuest database of the type *RAProject*. This record contains the information that is necessary for locating the Rational project from ClearQuest. In ClearQuest MultiSite, this record does not identify whether the Rational project can be reached from the database, because all records are replicated.

In a MultiSite environment, tracking changes and preventing data corruption is accomplished with an exclusive-right-to-modify scheme, called mastership. The database that has mastership of this RAProject record is considered to be “local” to the Rational project. Do not move the mastership of a RAProject record from one database replica to another.

When associating RequisitePro requirements with ClearQuest records in a ClearQuest MultiSite environment, users can only associate the requirements with ClearQuest records in a database replica that is local to the Rational project.

For more information, see *Administering Rational ClearQuest MultiSite* on the Rational Solutions for Windows Products Online Documentation CD.

Configuring a RequisitePro Project

No RequisitePro project configuration is necessary if you created your project from one of the following project templates provided with the current release of RequisitePro:

- Composite Template
- Traditional Template
- Use-Case Template

If you used one of these templates to create your RequisitePro project, skip to *Configuring a Rational Project*, below. If not, you must configure RequisitePro requirement types attributes to enable the integration. Refer to *Changing the Default RequisitePro Project Attributes* on page 38.

Configuring a Rational Project

If your ClearQuest database and RequisitePro project are already associated with a single Rational project, skip to *Setting Up the Integration Mapping*, below.

If you are converting a Rational repository from version 2000.02.10, you first must convert the repository to a Rational project. For more information, read about upgrading in *Installing Rational Suite*.

If you are not upgrading a Rational repository or project, use the Rational Administrator to create a Rational project. When configuring the Rational project, select the ClearQuest database and the RequisitePro project that you’ve configured for the integration. For more information, see *Creating a Rational Project* on page 23.

Setting Up the Integration Mapping

To enable the integration of ClearQuest and RequisitePro, an association must be created between ClearQuest record types and RequisitePro requirement types. This mapping is performed by editing and running a batch file. The following sections describe the steps required for setting up your integration mapping:

- collecting data related to your ClearQuest database and RequisitePro project environment
- editing the ASCQISetup.bat batch file
- executing the edited ASCQISetup.bat batch file

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 1. 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, see *ASCQISetup.exe Command Argument Descriptions* on page 39.

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

Table 1 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.	
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”. Edit this entry only if you installed RequisitePro on a drive other than the “C” drive.	

Table 1 Variables Describing Your Environment (continued)

Batch Command	Description	Must Edit	Your Values
set U =	The Superuser logon name for the ClearQuest database, the RequisitePro project, and the Rational Project	The default value for the Superuser logon is “admin”. If you are using a logon for RequisitePro, ClearQuest, or the Rational Project other than “admin”, edit this field. (The Super user logon refers specifically 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 box.	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. Use the path defined in your RequisitePro .rqs file. If your installation spans multiple computers, use a UNC path.	
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 the CQI.hlp file is in a location other than the default location listed in the ASCQISetup.bat file. If your installation spans multiple computers, use a UNC path.	

Editing the ASCQISetup.bat file

Now use the information in the preceding table to modify the ASCQISetup.bat file for your specific environment. The file is in the directory *Install*\Program Files\Rational\Common. The ASCQISetup.bat file is designed to be run *only* by the administrator and all settings in this file should reflect those of the administrator's installation. After the file runs, it does not need to 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 integrate.

Begin by saving a copy of the ASCQISetup.bat file with a name that corresponds to the Rational project you configured in the section *Configuring a Rational Project* on page 32. For example, you might create a copy named ASCQI_*RA_project_name*.bat. Edit this copy to include the data that is specific to your ClearQuest database and RequisitePro project environment as follows:

1 Locate the file 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 provide the value containing your project-specific information you supplied in Table 1.

The following example of a modified ASCQISetup.bat file was 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=<<CLSIC>>
```

```
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 project.
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 nonlocal computer.
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 nonlocal computer.
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 RAProject Modify "Associate Defects" "%Help%" 1 "%RP%"
"%RQS%" FEAT Defect 2


pause
```

```
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 RAProject Modify "Associate
Enhancement Requests" "%Help%" 1 "%RP%" "%RQS%" FEAT
EnhancementRequest 2


pause
```

Note: For more information on the ASCQISetup.exe command arguments, refer to the section *ASCQISetup.exe Command Argument Descriptions* on page 39.

Note: If you have a RequisitePro project that does not have security enabled, your RequisitePro default project logon user name must be set to the SuperUser login name specified in the ASCQISetup.bat file. This setting is available through the **RequisitePro Tools > Options** menu command.

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 ENTER until the batch file ends.

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

Customizing Your Integration

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

- “Defect” and “EnhancementRequest” ClearQuest record types
- “Feature” RequisitePro requirement type

You can use your own 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 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 a name for your new attribute.
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.

Changing the Default ClearQuest Records

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

- EnhancementRequest
- Defect

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, refer to the manual, *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 in the **ASCQISetup.bat** batch file as follows:

1. Replace **FEAT** with your requirement type, and **Defect** or **EnhancementRequest** with your customized attribute label or record.
2. Copy and update this command for each requirement type and attribute name to include in the integration.
3. Verify that the Back Reference field name, which is the fifth argument in the ASCQISetup.exe command, matches the name you used for the corresponding record type in your ClearQuest schema.
4. Run the ASCQISetup.bat file to activate the new values.

For more information about command line arguments for ASCQISetup.exe, see *ASCQISetup.exe Command Argument Descriptions* on page 39.

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 arguments of ASCQISetup.exe in the order in which they appear in the command. Table 2 describes the Rational ClearQuest arguments; Table 3 describes Rational RequisitePro arguments.

Table 2 Rational ClearQuest Related ASCQISetup.exe Arguments

Command Line Argument	Description	Comments
%U%	Name of Super User in ClearQuest, a user in RequisitePro, and a user in the Rational 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 3 Rational RequisitePro Related ASCQISetup.exe Arguments

Command Line Argument	Description	Comments
RAProject	Name of the ClearQuest Record Type Field that stores the Rational 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.	

Command Line Argument	Description	Comments
“Associate Defects”	Title of the association dialog window 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 windows (either when the Integration is called from RequisitePro or ClearQuest) in which you want to create associations.	This variable is defined in the ASCQISetup.bat file.
2	Context ID in the help file. This file is called from the Help button of the association dialog window, which 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.	
3	Context ID in the help file. This file is called from the Help button of the association dialog window, which appears when the integration is called from RequisitePro.	

Getting Started with ClearQuest and RequisitePro

For more information about using the ClearQuest – RequisitePro integration, refer to the document *Getting Started with Rational Suite AnalystStudio* contained on your documentation CD, as well as the Let’s Go AnalystStudio wizard provided with your software.

Working with and Configuring Extended Help

5

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.

After 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 46.
2. From the Extended Help menu, select **Edit > Options**.
3. Click the **Add** button, then fill in the table's cells, pressing **Enter** or **Tab** after entering a value in each column:
   - **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.
4. Click **OK**.

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**.

The change will take effect the next time you start Extended Help.

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 47)
- You are upgrading from an earlier version of Rational Suite

To prepare a database for use with Extended Help:

1. Start with a correctly structured file in comma-separated variable format (CSV):
   - See *Creating an Extended Help Database* on page 47 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 46.
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 44.

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 46.

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 (described in this section).
2. Add the database to Extended Help or tell your users to add it (described on page 44).
3. Tell your users how to use translation strings (described on page 50).

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 49.
B	subtool	The specific context within a product. See list of valid values on page 49.
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: ▪ a URL using an absolute value (example: http://www.rational.com) ▪ a URL using a translation string – described on page 50 (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 44).

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**
ClearQuest	Client
ClearQuest	Designer
RationalTest	Administrator
RationalTest	LoadTest
RationalTest	LogViewer
RationalTest	Robot
RationalTest	TestFactory
RationalTest	TestManager
ReqPro	Toolbar
ReqPro	View
Rose	Activity Diagram
Rose	Class Diagram
Rose	Collaboration Diagram
Rose	Deployment Diagram
Rose	Other Tool
Rose	Package Diagram
Rose	Sequence Diagram
Rose	Statechart Diagram
Rose	Use Case Diagram
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**.

Working with and Configuring the Rational Synchronizer

6

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 Rose class based on the existence of a feature requirement in RequisitePro. Or it might create an actor in Rose based on the existence of an actor requirement in RequisitePro.

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 choose.

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 Rose. 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 Actors.

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

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

  For example, the Synchronizer can cause a name change in a Rose use case to trigger a corresponding change to a use case requirement in RequisitePro.

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)

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:

- Zero or one Rational Test database
- Zero or one Rational ClearQuest databases
- Zero or one Rational Suite Synchronizer rules file
- Zero or 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, 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:

- Upgrading the Synchronizer environment from a previous release of Rational Suite to the current release. For more information, see *Upgrading the Rational Synchronizer Environment* in *Installing Rational Suite.*
- Creating a Rational project for your team. (For a brief description of a Rational project, see page 55. 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 58.

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 *Rational project*\Project\rules project\, where:

- *Rational project* is the name of the Rational project you created with the Rational Administrator
- *rules project* is the path containing the .rsr file.

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 to exchange information 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:

1. Start the Synchronizer: from the Windows **Start** menu, click **Programs > Rational Suite > Rational Synchronizer**.

   The Rational Synchronizer - Setup Context dialog box appears.

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

3. 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.

4. 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 process:

1. Write a narrative description of how your group works with items such as requirements, model elements.
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: ▪ Text ▪ Tag ▪ Key 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 (but not if it is 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 retransferred 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.

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 1 describes the rules supplied with the Rational Synchronizer.

Table 1 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 a 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 a Supplemental requirement	For a supplemental requirement, create a test requirement. This rule applies when the Test attribute on the SUPL requirement is True.
uc_requirement_to_tr_requirement.rsd Create Test requirement from a Use Case requirement	For a use case requirement, create a test requirement. 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 in *Install Path*\Common\OutOfTheBox\synchronizer rules\synchronizer_rule_template.rsd

Synchronizer Rule Grammar

For completeness, Table 2 presents the Synchronizer rules grammar. **Bold** text designates terminal symbols; regular text designates nonterminal symbols.

Table 2 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 2 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 2 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**

Troubleshooting 7

This chapter provides pointers to information about troubleshooting, available throughout the Rational documentation set.

Table 1 Where to Find Troubleshooting Information

Topic	**Source of Troubleshooting Information**
Rational Suite licenses	“Troubleshooting Licenses” chapter of *Administering Licenses for Rational Software*
Rational Suite installations	“Troubleshooting” chapter of *Installing Rational Suite*
ClearCase LT Administration	“Troubleshooting” chapter of *Administering ClearCase LT*
RequisitePro installation	“Troubleshooting” chapter of *Installing Rational RequisitePro*
RequisitePro use	“Troubleshooting” chapter of *Using Rational RequisitePro*

Index