Rational Software Corporation®

RATIONAL® CLEARCASE® LT

INTRODUCTION

UNIX/WINDOWS EDITION

VERSION: 2002.05.00 AND LATER

PART NUMBER: 800-025074-000

Rational®
the software development company

Introduction
Document Number 800-025074-000 October 2001
Rational Software Corporation 20 Maguire Road Lexington, Massachusetts 02421

IMPORTANT NOTICE

Copyright

Copyright © 1992, 2001 Rational Software Corporation. All rights reserved.
Copyright 1989, 1991 The Regents of the University of California
Copyright 1984–1991 by Raima Corporation

Permitted Usage

THIS DOCUMENT IS PROTECTED BY COPYRIGHT AND CONTAINS INFORMATION PROPRIETARY TO RATIONAL. ANY COPYING, ADAPTATION, DISTRIBUTION, OR PUBLIC DISPLAY OF THIS DOCUMENT WITHOUT THE EXPRESS WRITTEN CONSENT OF RATIONAL IS STRICTLY PROHIBITED. THE RECEIPT OR POSSESSION OF THIS DOCUMENT DOES NOT CONVEY ANY RIGHTS TO REPRODUCE OR DISTRIBUTE ITS CONTENTS, OR TO MANUFACTURE, USE, OR SELL ANYTHING THAT IT MAY DESCRIBE, IN WHOLE OR IN PART, WITHOUT THE SPECIFIC WRITTEN CONSENT OF RATIONAL.

Trademarks

Rational, Rational Software Corporation, the Rational logo, Rational the e-development company, Rational Suite ContentStudio, ClearCase, ClearCase MultiSite ClearQuest, Object Testing, Object-Oriented Recording, Objectory, PerformanceStudio, PureCoverage, PureDDTS, PureLink, Purify, Purify'd, Quantify, Rational Apex, Rational CRC, Rational PerformanceArchitect, Rational Rose, Rational Suite, Rational Summit, Rational Unified Process, Rational Visual Test, Requisite, RequisitePro, RUP, SiteCheck, SoDA, TestFactory, TestMate, TestStudio, The Rational Watch, among others 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.

Sun, Solaris, and Java are trademarks or registered trademarks of Sun Microsystems, Inc.

Microsoft, the Microsoft logo, the Microsoft Internet Explorer logo, Windows, the Windows logo, Windows NT, the Windows Start logo are trademarks or registered trademarks of Microsoft Corporation in the United States and other countries.

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

Patent

U.S. Patent Nos. 5,193,180 and 5,335,344 and 5,535,329 and 5,574,898 and 5,649,200 and 5,675,802 and 5,835,701. Additional patents pending.

Government Rights Legend

Use, duplication, or disclosure by the U.S. Government is subject to restrictions set forth in the applicable Rational License Agreement and in DFARS 227.7202-1(a) and 227.7202-3(a) (1995), DFARS 252.227-7013(c)(1)(ii) (Oct 1988), FAR 12.212(a) 1995, FAR 52.227-19, or FAR 52.227-14, as applicable.

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.

Technical Acknowledgments

This software and documentation is based in part on BSD Networking Software Release 2, licensed from the Regents of the University of California. We acknowledge the role of the Computer Systems Research Group and the Electrical Engineering and Computer Sciences Department of the University of California at Berkeley and the Other Contributors in its development.

This product includes software developed by Greg Stein <gstein@lyra.org> for use in the mod_dav module for Apache (http://www.webdav.org/mod_dav/).

Contents

Preface vii
About This Manual vii
Typographical Conventions vii
Online Documentation viii
Technical Support ix

1. Overview of ClearCase LT 1
1.1 Benefits of Using ClearCase LT 1
1.2 Basic Terminology of ClearCase LT 2
1.3 Working with ClearCase LT 5
Using Unified Change Management 5
UCM Terminology 6
Creating a Project in UCM 9
Joining the Project 11
Delivering Work to the Integration Stream 11
Rebasing Private Work Areas 12
Implementing Development Policies 13
Integrating UCM with ClearQuest 14
Using Base ClearCase LT 14
Base ClearCase LT Terminology 14
Setting Up a Project in Base ClearCase LT 17
Planning a Branching Strategy 17
Using a Private Branch 18
Merging Work Between Branches 19
Creating Standard Config Specs 20
Creating Labels 20
Using Metadata to Implement Policies 21

2. Starting to Use ClearCase LT 23
2.1 First Steps for Project Managers and Administrators 23
Using UCM 23
Using Base ClearCase LT 24
2.2 First Steps for Developers on Windows 24
Starting Rational ClearCase LT Explorer 25
Completing the ClearCase LT Tutorial 25
Setting Up Work Areas and Starting Your Work 25
2.3 First Steps for Developers on UNIX 26
Completing the ClearCase LT Tutorial 26
Setting Up Work Areas and Starting Your Work 26
Using UCM 26
Using Base ClearCase LT 27
2.4 Finding Out More About ClearCase LT 28
ClearCase LT Information Set 28
Getting Information About Developer Tasks 28
Getting Information About Project Manager and Administrator Tasks 29
Getting Information About Administrator Tasks 29

Figures

Figure 1 Elements and Their Versions 2
Figure 2 Storing Elements in a Versioned Object Base (VOB) 3
Figure 3 Using the Check Out/Check In Model 4
Figure 4 Using a View to See Shared Elements 4
Figure 5 An Activity 6
Figure 6 A Stream 7
Figure 7 A Baseline 8
Figure 8 Composite Baseline 9
Figure 9 Private Work Areas 10
Figure 10 Shared Work Area 11
Figure 11 Delivering Work to the Integration Stream 12
Figure 12 Rebasing the Development Stream 13
Figure 13 Branches 15
Figure 14 Version Label 16
Figure 15 Selecting Versions Used in Development Work 16
Figure 16 Selecting Versions Included in a Particular Release 17
Figure 17 Branches in a File System Directory Tree 18
Figure 18 Working on a Private Branch 19
Figure 19 Merging Branches 20

Preface

Rational ClearCase LT is a configuration management (CM) system that manages multiple variants of evolving software systems. ClearCase LT maintains a complete version history of all software development artifacts, including code, requirements, models, scripts, test assets, and directory structures. It enforces site-specific development policies, offers multiple developer workspaces, and provides advanced support for parallel development. ClearCase LT includes Unified Change Management (UCM), an optional, out-of-the-box process for organizing software development teams and their work products.

About This Manual

This manual is intended for evaluators and users of ClearCase LT. It provides an overview of the product and describes how to get started using ClearCase LT and where to get more information. For information on installing ClearCase LT, see the *Installation Guide* for Rational ClearCase LT.

Typographical Conventions

This manual uses the following typographical conventions:

- **Bold** is used for names the user can enter; for example, all command names, file names, and branch names.
- *Italic* is used for variables, document titles, glossary terms, and emphasis.
- `A monospaced font` is used for examples. Where user input needs to be distinguished from program output, **bold** is used for user input.

- Nonprinting characters are in small caps and appear as follows: **<EOF>**, **<NL>**.
- Key names and key combinations are capitalized and appear as follows: SHIFT, CTRL+G.
- [ ] Brackets enclose optional items in format and syntax descriptions.
- { } Braces enclose a list from which you must choose an item in format and syntax descriptions.
- | A vertical bar separates items in a list of choices.

Online Documentation

The ClearCase graphical interface includes an online help system.

There are three basic ways to access the online help system: the **Help** menu, the **Help** button, or the F1 key. **Help > Contents** provides access to the complete set of ClearCase online documentation. For help on a particular context, press F1. Use the **Help** button on various dialog boxes to get information specific to that dialog box.

ClearCase also provides access to full "reference pages" (detailed descriptions of ClearCase commands, utilities, and data structures) with the **cleartool man** subcommand. Without any argument, **cleartool man** displays the **cleartool** overview reference page. Specifying a command name as an argument gives information about using the specified command. For example:

cleartool man *(display the cleartool overview page)*

cleartool man man *(display the cleartool man reference page)*

cleartool man checkout *(display the cleartool checkout reference page)*

ClearCase's **–help** command option or **help** command displays individual subcommand syntax. Without any argument, **cleartool help** displays the syntax for all **cleartool** commands. **help checkout** and **checkout –help** are equivalent.

cleartool uncheckout –help

```
Usage: uncheckout | unco [-keep | -rm] [-cact | -cwork ] pname ...
```

Additionally, the online *ClearCase Tutorial* provides a step-by-step tour through ClearCase's most important features. To start the tutorial:

- On Windows, choose **Tutorial** in the appropriate ClearCase folder off the **Start** menu.

➤ On UNIX, type **hyperhelp cc_tut.hlp**.

Technical Support

If you have any problems with the software or documentation, please contact Rational Technical Support via telephone, fax, or electronic mail as described below. For information regarding support hours, languages spoken, or other support information, click the **Technical Support** link on the Rational Web site at **www.rational.com**.

Your Location	Telephone	Facsimile	Electronic Mail
North America	800-433-5444 toll free or 408-863-4000 Cupertino, CA	408-863-4194 Cupertino, CA 781-676-2460 Lexington, MA	**support@rational.com**
Europe, Middle East, and Africa	+31-(0)20-4546-200 Netherlands	+31-(0)20-4546-201 Netherlands	**support@europe.rational.com**
Asia Pacific	61-2-9419-0111 Australia	61-2-9419-0123 Australia	**support@apac.rational.com**

1 Overview of ClearCase LT

This chapter introduces the features of Rational ClearCase LT and describes the different models for using the product.

1.1 Benefits of Using ClearCase LT

Rational ClearCase LT is an extensive set of tools that you can use to manage and track software resources, whether they are Web pages, mission-critical data, documentation, or source code. Like other source control tools, ClearCase LT helps you do the following with your software resources:

- Create new versions of a software resource
- Compare versions of a software resource
- Merge changes from one version into another version
- Control simultaneous changes to a software resource
- Mark certain versions as stable sources to be used in builds
- Determine the who, when, and why of a particular change

ClearCase LT also provides some unique advantages:

- Supports team leaders who need to coordinate the activity of people developing products together

- Gives project managers control over the extent and frequency with which team members synchronize their work

- Supports parallel development: developers work in private areas and do not affect the work of team members
- Assists integrators in combining the efforts of the team in a controlled manner

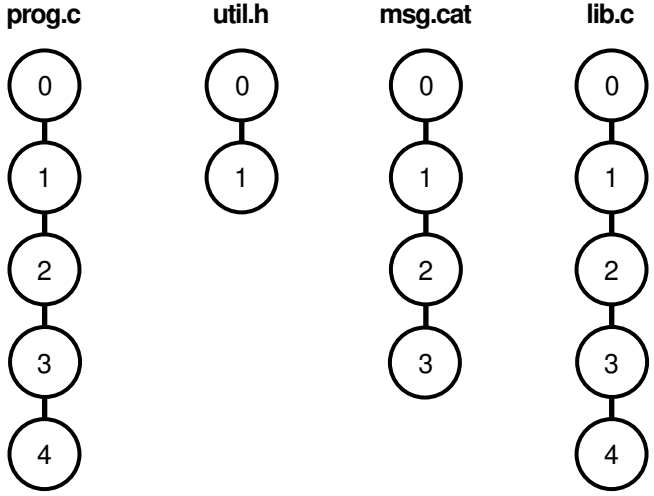
1.2 Basic Terminology of ClearCase LT

Before you learn about the most important features of ClearCase LT, it is helpful for you to understand the following terms.

A *file element* is a file that contains software source code, a document, HTML code, XML code, or other data that can be stored in a file system. A *directory element* contains file elements and other directory elements.

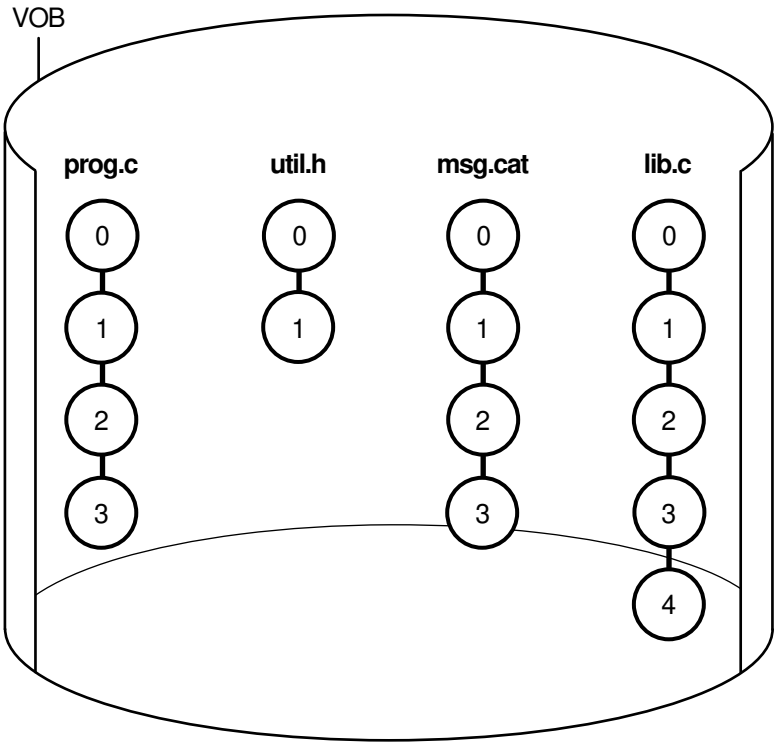
A *version* is a specific revision of an element. For instance, instead of overwriting the same copy of your draft each time you work on it, you store a copy of the first version, the second version, and so on (Figure 1).

Figure 1 Elements and Their Versions



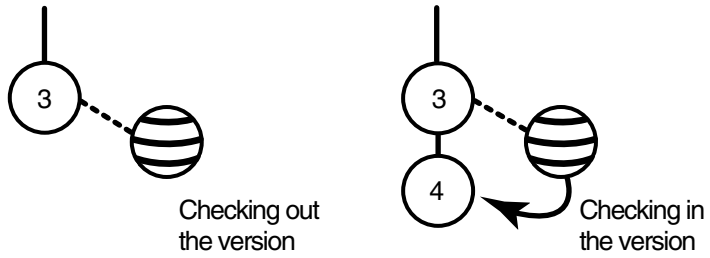
A *versioned object base*, or *VOB*, is a repository that stores versions of file elements and directory elements (Figure 2).

Figure 2 Storing Elements in a Versioned Object Base (VOB)



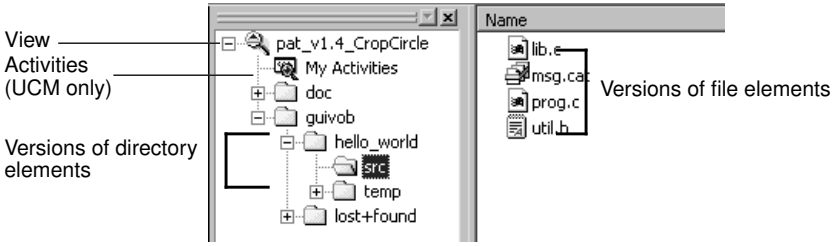
A *check out-edit-check in model* manages changes to your project. When you check out an element, ClearCase LT creates an editable copy in your view. When you check in an element, a new version of it is added to the VOB. For instance, you check out a chapter of a book from the documentation VOB. You work on a copy of the chapter in your view. When you want your changes to become a part of a new version of the chapter, you save the changes and check in the chapter (Figure 3).

Figure 3 Using the Check Out/Check In Model



A *view*, which is represented as a directory, provides access to a specific version of one or more elements in a VOB. It allows you to select a set of versions of elements without having to specify the versions explicitly. It provides a workspace in which you can work on assignments in isolation from other developers (Figure 4). For example, as you work on the latest version of a Web page, no one can see the changes you've made until you've checked in that work to the VOB.

Figure 4 Using a View to See Shared Elements



1.3 Working with ClearCase LT

Configuration and change management is a system or process that ensures each stage of project development (planning, editing, testing, building, and deploying) can be controlled and reproduced easily.

ClearCase LT offers two configuration management interfaces:

- Unified Change Management (UCM), which is an out-of-the-box, activity-based change management process
- Base ClearCase LT, which is a set of tools that you can use to create a configuration management solution tailored to your development environment

UCM is layered on base ClearCase LT, which allows you to work in UCM without having to master the basic tool set. In UCM, the project team can track and manage project work at the activity level. Process support for UCM is installed when you install ClearCase LT.

When you use base ClearCase LT, your work is recorded at the file or element level. ClearCase LT records when each file is created or modified. Changes to files that make up a development task, such as a bug fix, are not recorded as a group. To reconstruct the history of a development task, you must examine the history of individual files.

You can extend either configuration management interface by integrating it with Rational ClearQuest, a tool that tracks defects and change requests.

The following sections provide overviews of working in ClearCase LT with and without UCM. For more information about creating and managing projects with UCM, see *Managing Software Projects*. For information on joining a project and working in UCM, see the online help.

Using Unified Change Management

UCM structures the work of a project team into a defined, repeatable process. This section provides an overview of the key concepts and tasks involved in using UCM.

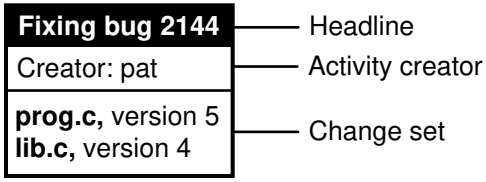
UCM Terminology

A project is a specific product of a development effort, such as a corporate Web site or an order fulfillment process for an e-business. In UCM, a *project* is an object that contains the configuration information (for example, components, activities, policies) needed to manage and track the work on a product. A UCM project in ClearCase LT consists of one shared work area and many private work areas (typically one per developer).

A *component* is any group of source code and other relevant elements or files, such as a license module or a customer GUI, that the team develops, integrates, and releases as a unit. Components constitute parts of a project, and projects often share components.

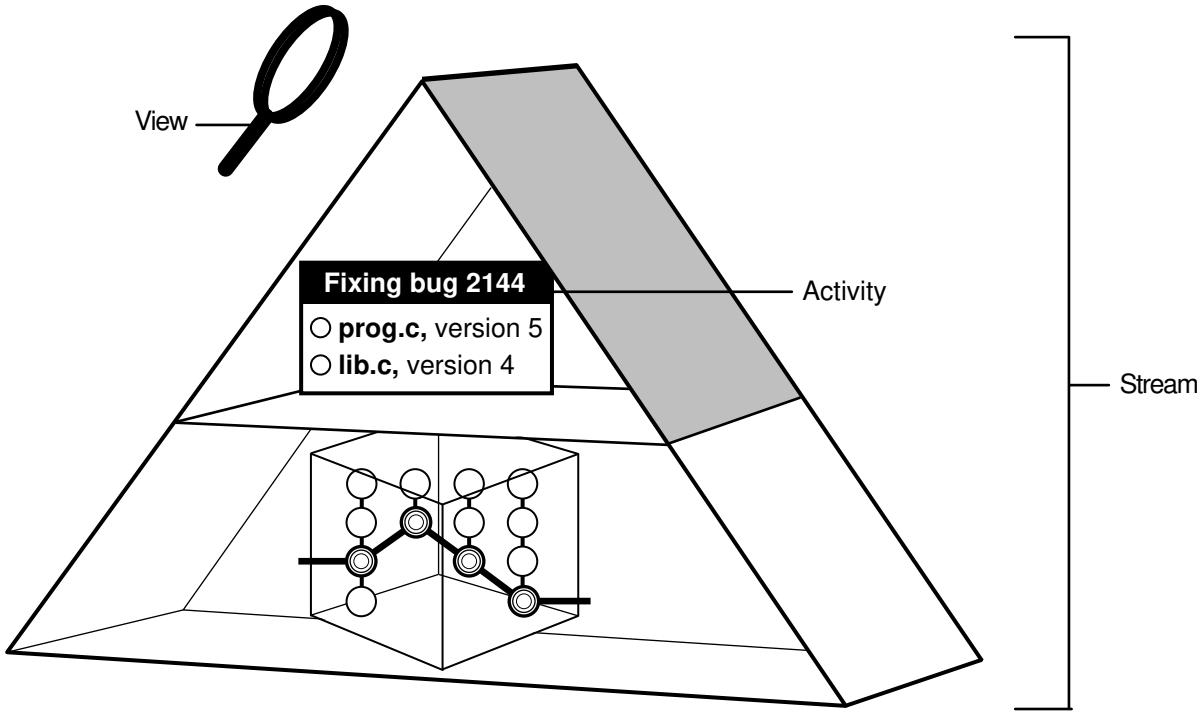
An *activity* is an object that records the set of files (*change set*) that a developer creates or modifies to complete and deliver a development task, such as a bug fix (Figure 5). Examples of other activities could include an update to an online help file or the addition of a menu item to a GUI component.

Figure 5 An Activity



A work area consists of a view and a stream. As discussed in *Basic Terminology of ClearCase LT* on page 2, a *view* is a directory tree that shows a single version of each file in your project. A *stream* is a ClearCase object that maintains a list of activities and baselines and determines which versions of elements appear in your view (Figure 6).

Figure 6 A Stream

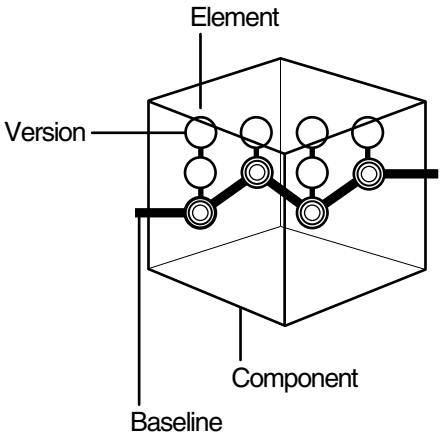


A project contains one *integration stream*, which records the project's baselines and enables access to versions of the project's shared elements. The integration stream and a corresponding integration view represent the project's shared work area.

Each developer on a project has a private work area, which consists of a development stream and a corresponding development view. The *development stream* maintains a list of the developer's activities and determines which versions of elements appear in the developer's view.

A *baseline* identifies one version of each element in a component that represents the integrated or merged work of team members (Figure 7). It represents a version of a component at a particular stage in project development, such as the first draft of a book, a beta release, or a final product release. Throughout the project cycle, the project manager creates and recommends baselines and changes their promotion level attributes to reflect project milestones.

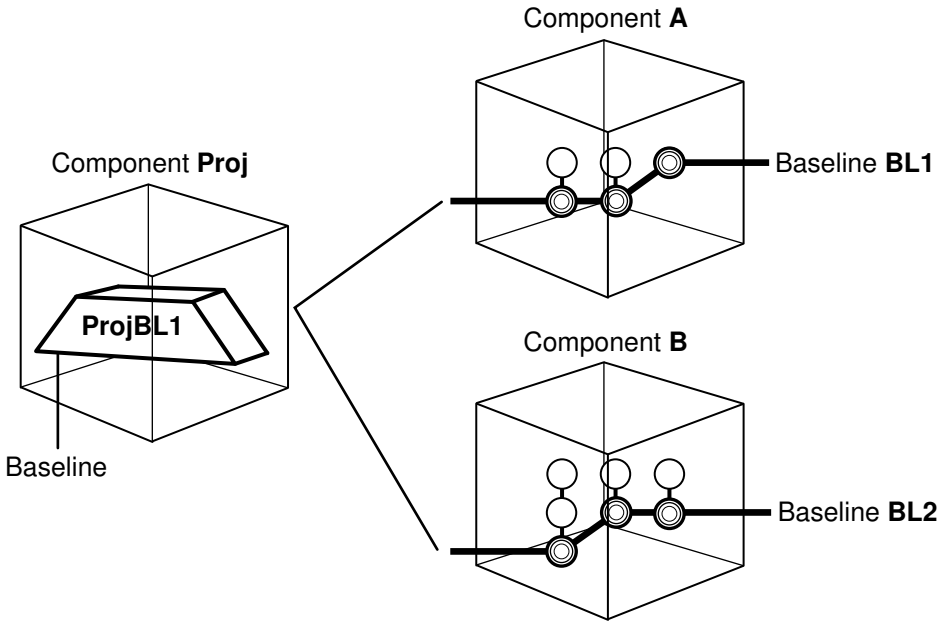
Figure 7 A Baseline



When developers join the project, they populate their work areas with the versions of directory and file elements represented by the project's recommended baselines. Alternatively, developers can join the project at a feature-specific development stream level, in which case they populate their work areas with the development stream's recommended baselines. This practice ensures that all members of the project team start with the same set of files.

If your project team works on multiple components, you many want to use a composite baseline. A *composite baseline* is a baseline that selects baselines in other components. In Figure 8, the **ProjBL1** composite baseline selects baselines **BL1** and **BL2** of components **A** and **B**, respectively. The **Proj** component does not contain any elements of its own. Its sole purpose is to contain the composite baseline that selects the recommended baselines of the project's components. By using a composite baseline in this manner, you can identify one baseline to represent the entire project.

Figure 8 Composite Baseline



Creating a Project in UCM

Before creating a project, the project manager maps files and directories in the product architecture to a set of UCM components. Using utilities provided in ClearCase LT, the project manager or administrator converts existing files and directories from third-party version control software systems to elements. See *Managing Software Projects with ClearCase* and *Administering ClearCase LT* for details about importing existing files and directories into components.

The project manager creates a project, along with its baselines and integration stream, with the Project Explorer. You can create a new project from scratch or populate a new project with baselines from an existing project.

NOTE: When you install ClearCase LT Server, the ClearCase LT Server Set Up wizard creates a *project VOB* (PVOB), an initial component VOB, an initial project, and an initial baseline. On Windows, it also asks whether you want to import data into ClearCase LT and runs the Import Wizard, if appropriate.

The project includes a private work area for each developer (Figure 9) and one shared work area (Figure 10). The project manager manages the shared work area. These areas are explained in more detail in *Joining the Project* and *Delivering Work to the Integration Stream*.

Figure 9 Private Work Areas

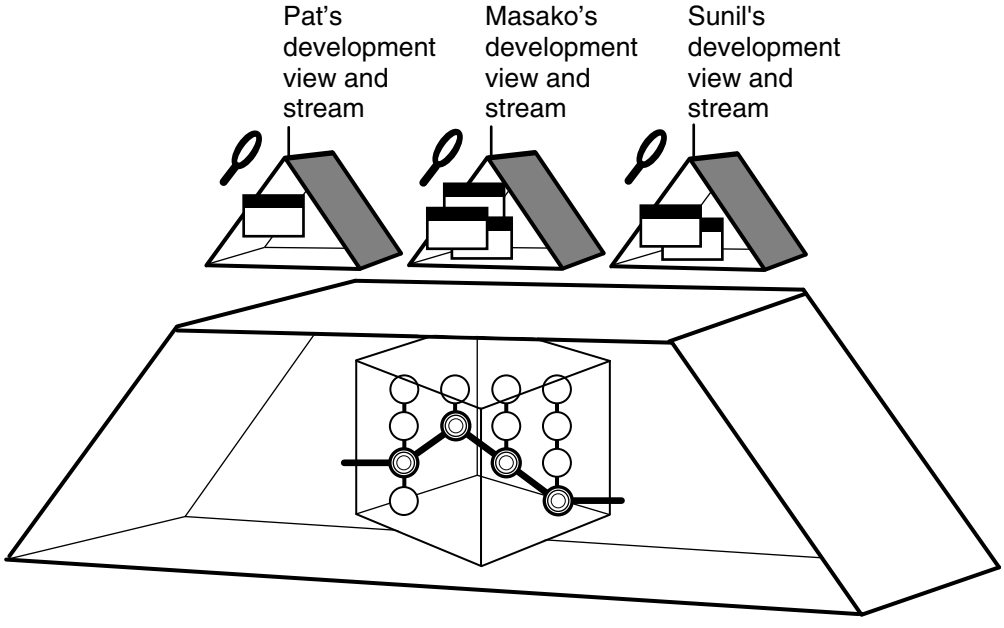
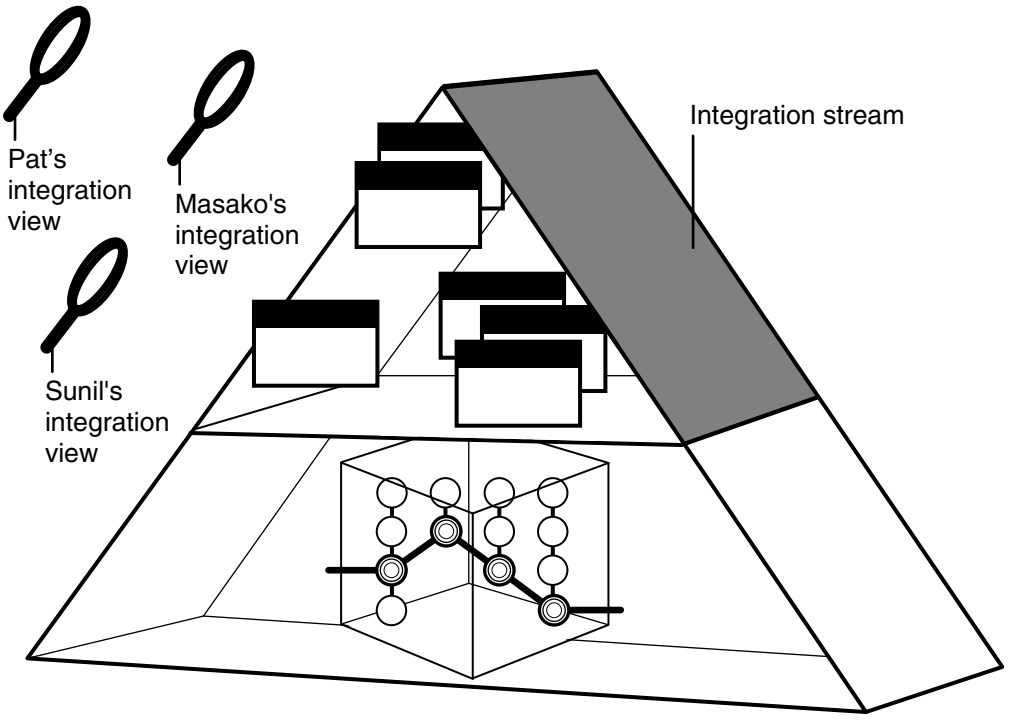


Figure 10 Shared Work Area



Joining the Project

To join the project, you use the Join Project Wizard to create a *development stream*, a *development view,* and an *integration view.*

The development stream and development view make up the private work area, where you work on activities. You use development views to access and change elements. The development stream determines which versions of elements appear in each development view.

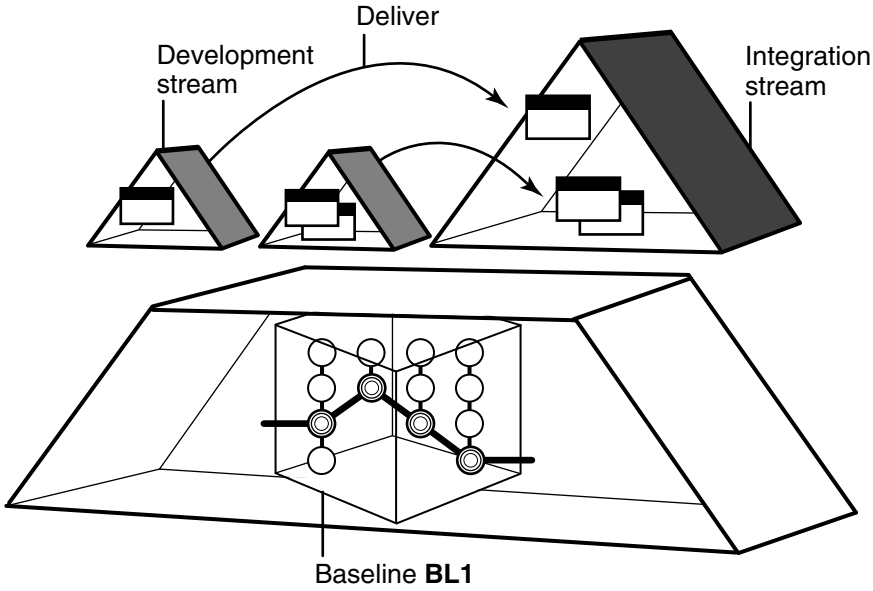
The integration view gives developers and project managers access to the work that has been delivered to the shared work area.

Delivering Work to the Integration Stream

When the project was created, the project manager created *one* shared work area, or *integration stream*.

When you are ready to integrate your work with the work of the team, you *deliver* that work from the development stream to the integration stream. The deliver operation may involve a merge if other team members have already delivered versions of one or more of the same elements (Figure 11).

Figure 11 Delivering Work to the Integration Stream



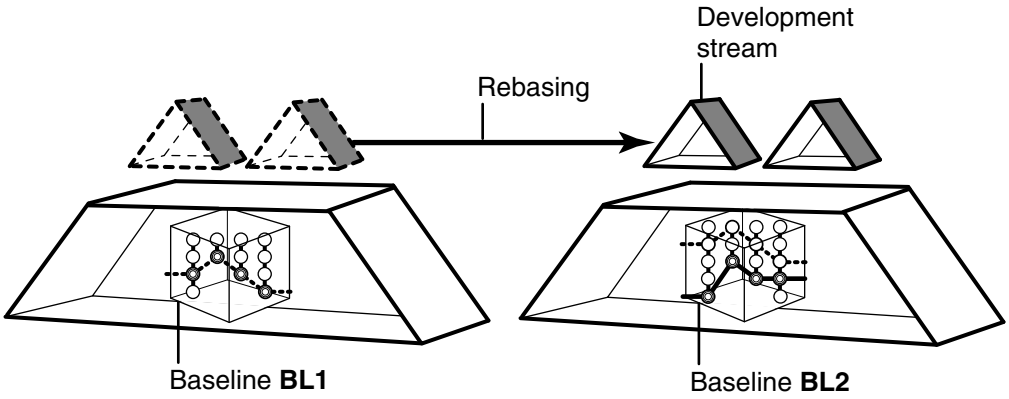
When the work is delivered and merged, you use the integration view to build and test work in the integration stream. The integration stream determines which versions of delivered elements appear in the integration view.

Rebasing Private Work Areas

Periodically, the project manager creates a new baseline, which includes the work delivered since the last baseline was created. When this new baseline passes a certain level of testing, the project manager promotes it, designating it as the project's *recommended baseline*.

As shown in Figure 12, you update, or *rebase*, your development stream to use the new recommended baselines. The rebase operation updates the development view so that it includes the versions from the new baselines. If the baselines include new versions of the same element that you checked in from a private work area (that is, another developer checked in and delivered a new version of the same element to the integration stream), the rebase operation merges the new version into your development stream.

Figure 12 Rebasing the Development Stream



Implementing Development Policies

UCM includes a set of policies that you can use on projects and streams to enforce development practices among members of the project team. By setting policies, you can improve communication among project team members and minimize the problems you may encounter when integrating their work. For example, you can set a policy that requires developers to update their work areas with the project's latest recommended baseline before they deliver work to the integration stream. This practice reduces the likelihood that developers will need to work through complex merges when they deliver their work. For a description of all policies that you can set in UCM, see *Managing Software Projects*.

In addition to the set of policies that UCM provides, you can create triggers on UCM operations to enforce customized development policies. A *trigger* is a monitor that specifies one or more standard programs or built-in actions to be executed whenever a certain ClearCase operation is performed. See *Managing Software Projects* for details about creating triggers.

Integrating UCM with ClearQuest

Rational ClearQuest is a change-request management application. Using ClearQuest, team members can submit change requests for their product, view and modify existing change requests, and create and run user- or site-specific queries and reports to determine project status.

The UCM-ClearQuest integration provides additional activity management capabilities by linking project activities in UCM to corresponding records in a ClearQuest user database. The integration lets the project team track an activity's progress from assignment through completion. Project managers can assign activities to developers, and developers can use ClearQuest queries to find their assignments. Because activities have corresponding records in a ClearQuest user database, project managers can use all the query, reporting, and charting utilities in ClearQuest to help them manage projects.

For information about planning and setting up the UCM-ClearQuest integration, see the online help and *Managing Software Projects.* For information about working on activities in a project that has been linked to a ClearQuest user database, see the online help.

Using Base ClearCase LT

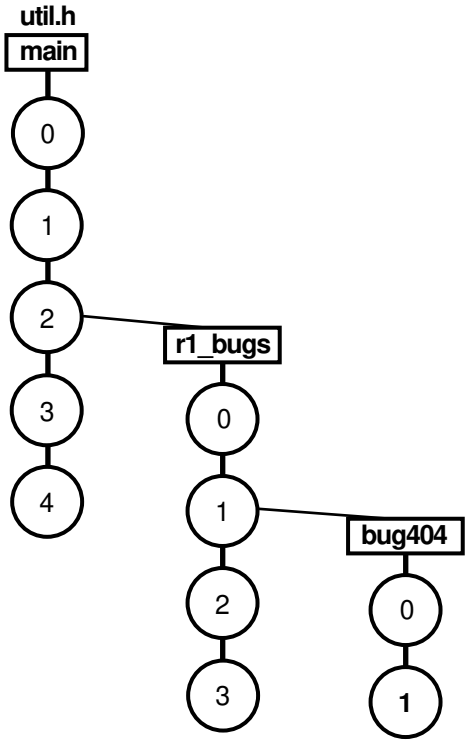
A project team can use ClearCase LT tools to develop its own system of managing and tracking software resources. This section describes the basic steps involved in setting up a work environment in base ClearCase LT.

For more information about creating a custom configuration management solution, see the online help, *Managing Software Projects,* and the *Administrator's Guide* for Rational ClearCase LT.

Base ClearCase LT Terminology

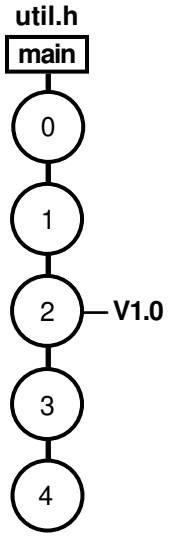
A *branch* is an object that specifies a sequence of versions of an element. Every element has one *main branch*, which represents the principal line of development, and may have multiple *subbranches*, each of which represents a separate line of development (Figure 13).

Figure 13 Branches



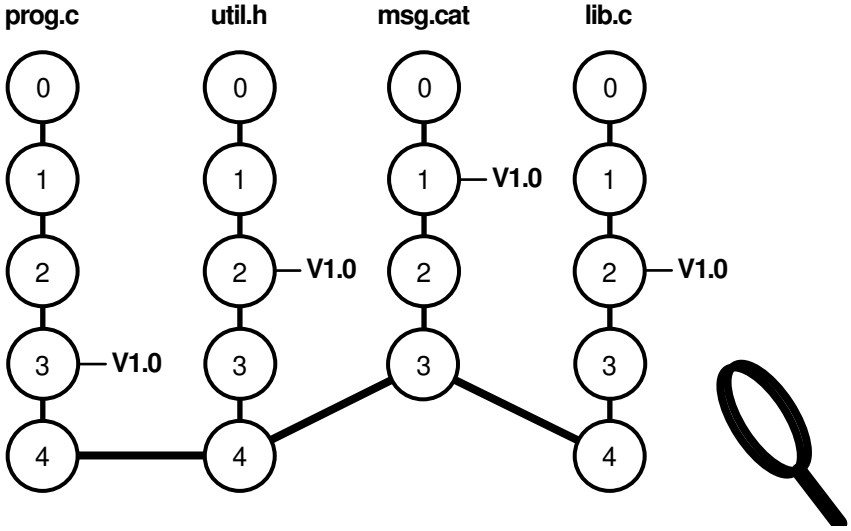
A *version label* can be attached to any version of an element to identify that version in an easy to remember way. A single version of an element can have several different labels. Labels are usually applied to a set of elements to mark important project milestones or the proposed starting point of a branch (Figure 14).

Figure 14 Version Label



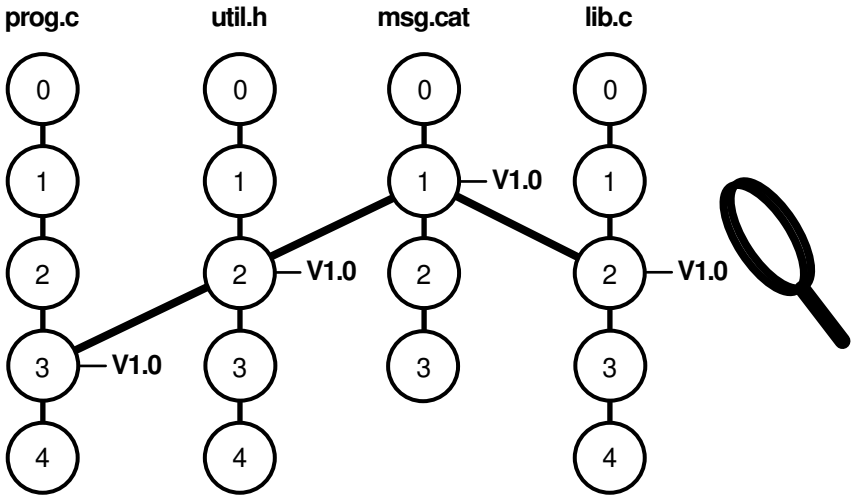
A *configuration specification*, or config spec, contains the rules used by a view to select versions of elements. The rules are very flexible, and you can use various specifiers to indicate which versions to select. For example, a config spec for a view used in ongoing development would contain rules that select the latest version on the development branch (Figure 15).

Figure 15 Selecting Versions Used in Development Work



To examine the versions that were included in a particular release, you would use a config spec that uses a label rule to select the versions that were labeled for that release (Figure 16).

Figure 16 Selecting Versions Included in a Particular Release



Setting Up a Project in Base ClearCase LT

When project managers set up projects in base ClearCase LT, they map files and directories in the product architecture to a set of ClearCase LT VOBs (just as project managers map files and directories to ClearCase LT components in UCM). For instructions on creating VOBs and populating them with files and directories, see the *Administrator's Guide* for Rational ClearCase LT.

After the files and directories are imported into VOBs, the project manager usually attaches labels to a set of versions of elements in those VOBs to designate the versions from which development should start.

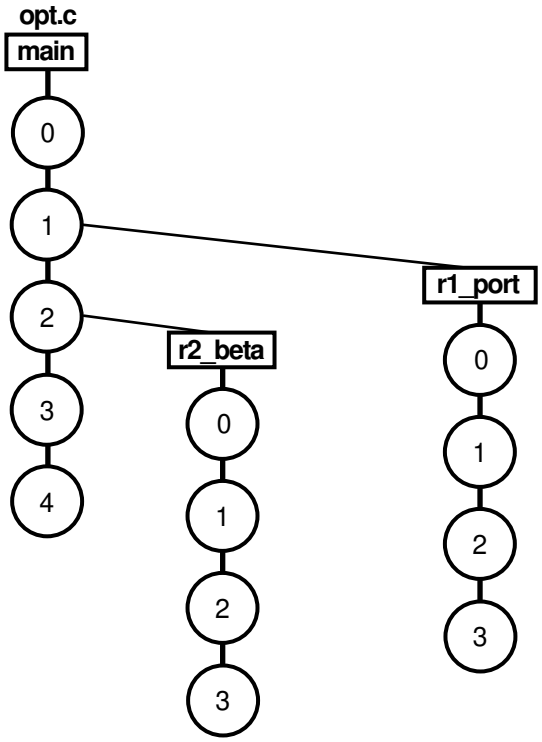
Planning a Branching Strategy

When you use base ClearCase LT, you must use branches directly to implement parallel development. (UCM uses streams to manage branches for you.)

This section provides a simple example of how to define a branching strategy. As you read the section, keep in mind that the possible branching and merging combinations are almost infinite.

A project team can use many branches concurrently. In Figure 17, the **main** branch represents the integration branch, an **r1_port** subbranch is a port to a new platform, an **r2_beta** subbranch contains work for the beta version of the next release, and so on.

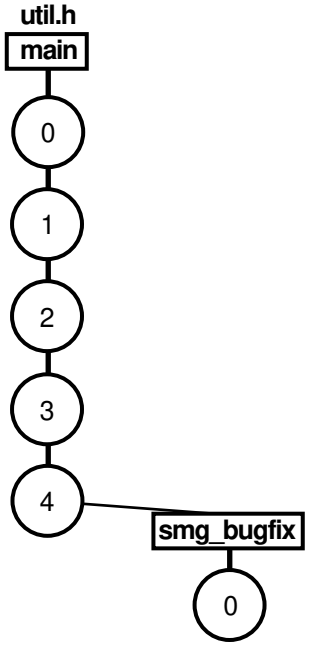
Figure 17 Branches in a File System Directory Tree



Using a Private Branch

At times, you may want to create a private branch based on the project branch. For example, you want to work on a bug fix without affecting the work that other developers are doing. To do this, choose the version from which you want to start. Then you can create a second config spec for a second view that will create a private branch starting at that version (Figure 18).

Figure 18 Working on a Private Branch



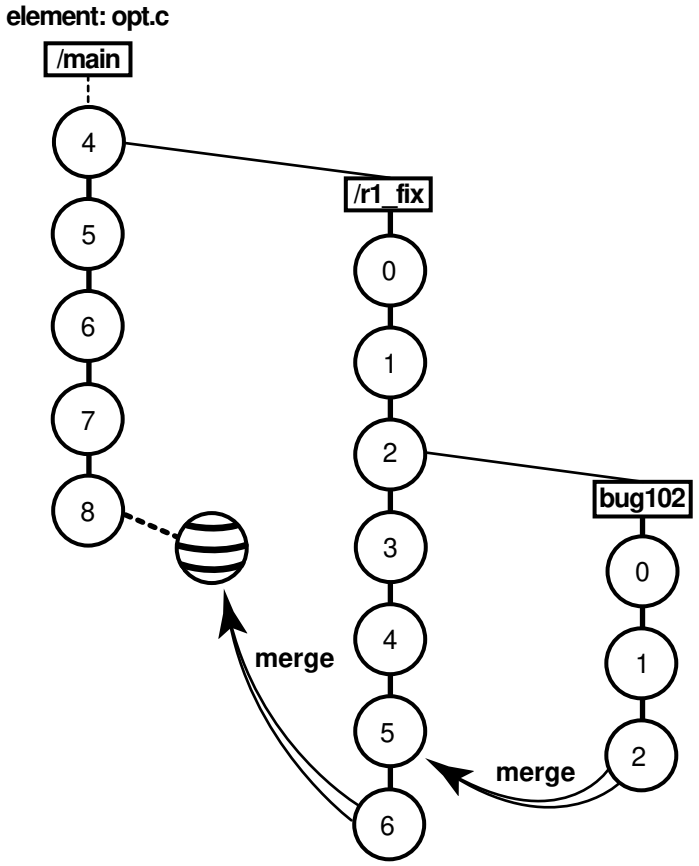
To incorporate your work into the project, merge the changes on your private branch to the project branch. If you decide to abandon your work, do not merge the changes. In either case, when you want to resume your work on the project branch, change your view back to use the project branch view.

For more information about creating private branches for development work and merging your work back to the project branch, see the online help.

Merging Work Between Branches

To integrate work from one branch to another, you *merge* work from one subbranch to another branch. In Figure 19, all work must be merged to the **main** branch before it can be included in a product release. After the fix on the **bug102** branch is tested and approved for integration, you first merge the work to the **r1_fix** branch. At some point, you test all the work on the **r1_fix** branch; when that work is ready to be incorporated into the main project, you merge from that branch to the **main** branch.

Figure 19 Merging Branches



Creating Standard Config Specs

To ensure that team members are working on the correct branches (appropriate directory and file versions), the project manager creates a standard config spec that team members use in their views. For more information on creating config specs, see *Managing Software Projects*.

Creating Labels

A *label* in base ClearCase LT is a user-defined name that can be attached to a version. Labels are a powerful tool for project managers. By applying labels to groups of elements, you can define and preserve the relationship of a set of file and directory versions to each other at a given point

in the development lifecycle. For example, you can apply a label to all versions considered stable after integration and testing. Use this baseline label as the foundation for new work.

Using Metadata to Implement Policies

In base ClearCase LT, project managers use *metadata* to define project policies and annotate different objects in the VOB. Metadata include the following:

- Attributes
- Branch types
- Hyperlinks
- Locks
- Triggers
- Version labels

NOTE: You can also use attributes, hyperlinks, locks, and triggers when you use UCM.

To annotate objects in the VOB, a project manager can use attributes, hyperlinks, and version labels:

- Attributes are name-value pairs that are attached to objects. One use for attributes is to attach bug numbers to versions; another is to identify which versions in a baseline have been tested.

- Hyperlinks connect two VOB objects. For example, you can define a hyperlink to connect a design specification to its associated executable.

- Labels identify specific versions of elements. Labels can be used in config spec rules to identify branching points or to mark all versions that were used in a build.

To define and enforce project policy, project managers can use locks and triggers. For example, an integration branch can be locked temporarily to prevent developers from merging work onto it. A project manager could also create a trigger that prompts developers to enter the bug number associated with a checkin and attaches that number to the version.

For details about using ClearCase LT metadata, see *Managing Software Projects* and the *Administrator's Guide* for Rational ClearCase LT.

Starting to Use ClearCase LT

2

This chapter describes what to do after installing Rational ClearCase LT and where to find more information about using it. If you are a project manager or administrator, read *First Steps for Project Managers and Administrators*. If you are a developer, read *First Steps for Developers on Windows* on page 24 or *First Steps for Developers on UNIX* on page 26.

For information on installing ClearCase LT, see the installation guide for your platform.

2.1 First Steps for Project Managers and Administrators

After you install ClearCase LT Server on the server computer and install ClearCase LT Client on your desktop computer, you need to plan and create project structures for the developers.

Using UCM

The ClearCase LT Server Set Up program creates a *project VOB* (PVOB), an initial component VOB, an initial project, and an initial baseline. On Windows, it also asks whether you want to import data into ClearCase LT and runs the Import Wizard, if appropriate.

After the Server Set Up program finishes, you can start development. You can also create additional projects, components, and baselines, but it is not necessary.

NOTE: A project must contain one or more components, and projects can share components.

For information on getting started using UCM, see the online help, *Managing Software Projects*, and the *Administrator's Guide*.for Rational ClearCase LT.

Using Base ClearCase LT

The ClearCase LT Server Set Up program creates a VOB and an administrative VOB. An *administrative VOB* is a VOB containing global type objects that are copied to client VOBs on an as-needed basis when users wish to create instances of the type objects in the client VOBs. On Windows, the ClearCase LT Server Set Up program also asks whether you want to import data into ClearCase LT and runs the Import Wizard, if appropriate.

After the Server Set Up program finishes, you can create additional VOBs. Each VOB you create is linked to the administrative VOB. If you choose not to use the administrative VOB, you can ignore these links.

You must also create these additional items:

- Branch types, to represent the different lines of development that occur in your organization.
- Configuration specification (config spec) rules, to control the versions that are selected by the view.
- Label types and labels, to identify the starting point of a branch or mark project milestones.

For more information, see the online help, *Managing Software Projects*, and the *Administrator's Guide* for Rational ClearCase LT.

2.2 First Steps for Developers on Windows

After you install ClearCase LT on your computer, do the following:

- Start Rational ClearCase LT Explorer
- Complete the ClearCase LT tutorial
- Set up work areas and start your work

The following sections describe these tasks.

Starting Rational ClearCase LT Explorer

Rational ClearCase LT Explorer is the interface to your development environment. To start Rational ClearCase LT Explorer, double-click the **Rational ClearCase LT** icon on your desktop or click **Start > Programs > Rational ClearCase LT Client > Rational ClearCase LT Explorer**.

Completing the ClearCase LT Tutorial

The tutorial introduces you to the development tasks you perform using ClearCase LT. Before starting the tutorial, ask your project manager or administrator whether you will be using UCM.

To start the tutorial:

1. Start Rational ClearCase LT Explorer.
2. In the left pane, click the **Toolbox** tab and click **Getting Started**.
3. Click **Tutorial**.
4. Read the introductory information and then click the appropriate tutorial button.

Setting Up Work Areas and Starting Your Work

To find the instructions for setting up your work areas:

1. In Rational ClearCase LT Explorer, click the **Toolbox** tab in the left pane.
2. Click **Getting Started**.
3. Click the appropriate Developing Software button.

Follow the instructions in the information pane of Rational ClearCase LT Explorer.

The menus, toolbars, and tabs in Rational ClearCase LT Explorer contain the commands you'll use to do your work. To learn more about the task flow in ClearCase LT, use the information in the information pane of Rational ClearCase LT Explorer.

You can now start to use ClearCase LT as part of your development work. *Finding Out More About ClearCase LT* on page 28 describes how to get more information about ClearCase LT.

2.3 First Steps for Developers on UNIX

After you install ClearCase LT on your computer, do the following:

- Complete the ClearCase LT tutorial
- Set up work areas and start your work

The following sections describe these tasks.

Completing the ClearCase LT Tutorial

The tutorial introduces you to the development tasks you perform using ClearCase LT. Before starting the tutorial, ask your project manager or administrator whether you will be using UCM.

To start the tutorial, type the following command on the command line:

hyperhelp cc_tut.hlp

Setting Up Work Areas and Starting Your Work

The ClearCase LT File Browser (**xclearcase**) is the graphical user interface to your ClearCase LT development environment. You can also use ClearCase LT commands on the command line to do your work.

Using UCM

To get started:

1. Type **xclearcase** at a command prompt.
2. Join a project:
   a. Click **Project > Join Project**.
   b. Follow the prompts in the Join Project Wizard.
3. Navigate to your snapshot view directory by typing its pathname in the directory box at the top of the File Browser.

To check out a file or directory, select it and click **Versions > Checkout > Prompt for options**.

For more information on an available button or command in the File Browser, right-click the button or command.

Using Base ClearCase LT

To get started:

1. Type **xclearcase** at a command prompt.
2. Create a view:
   a. Click **View > Create**.
   b. Type a comment in the Create View window and click **OK**.
3. Add load rules to your view's config spec to select the elements you want to load into the view:
   a. Click **Admin > View**.
   b. Select the view-tag.
   c. Click **ConfigSpec > Edit**.
   d. Add load rules. For example, if you want to load all elements in the **/dev** VOB and all elements in and under the **/doc/design** directory, add these lines:

      load /dev
      load /doc/design

   e. Save and close the edit window.
   f. Click **Yes**.

      Files and directories are loaded into your snapshot view.

   g. Click **Dismiss**.
   h. Click **File > Close window** to close the View Tag Browser.
4. Navigate to your snapshot view directory by typing its pathname in the directory box at the top of the File Browser.

To check out a file or directory, select it and click **Versions > Checkout > Prompt for options**.

For more information on an available button or command in the File Browser, right-click the button or command.

2.4 Finding Out More About ClearCase LT

This section describes the information available about ClearCase LT. You can get more information about Rational documentation at **www.rational.com/support/documentation**.

ClearCase LT Information Set

The ClearCase LT information set includes the following documentation:

- *Installation Guide* for Rational ClearCase LT
- *Release Notes* for Rational ClearCase LT (online only)
- *Introduction* to Rational ClearCase LT (this manual)
- *Managing Software Projects*
- *Administrator's Guide* for Rational ClearCase LT
- *Command Reference*
- Task help, available online
- Tutorial, available online

Getting Information About Developer Tasks

In UCM, typical development tasks include joining a project, working on and delivering activities, and rebasing the work area. For overview and procedural information for these tasks, see the online help.

In base ClearCase LT, typical development tasks include creating and updating a view, checking out and checking in files and directories, and merging files. For overview and procedural information for these tasks, see the online help.

Getting Information About Project Manager and Administrator Tasks

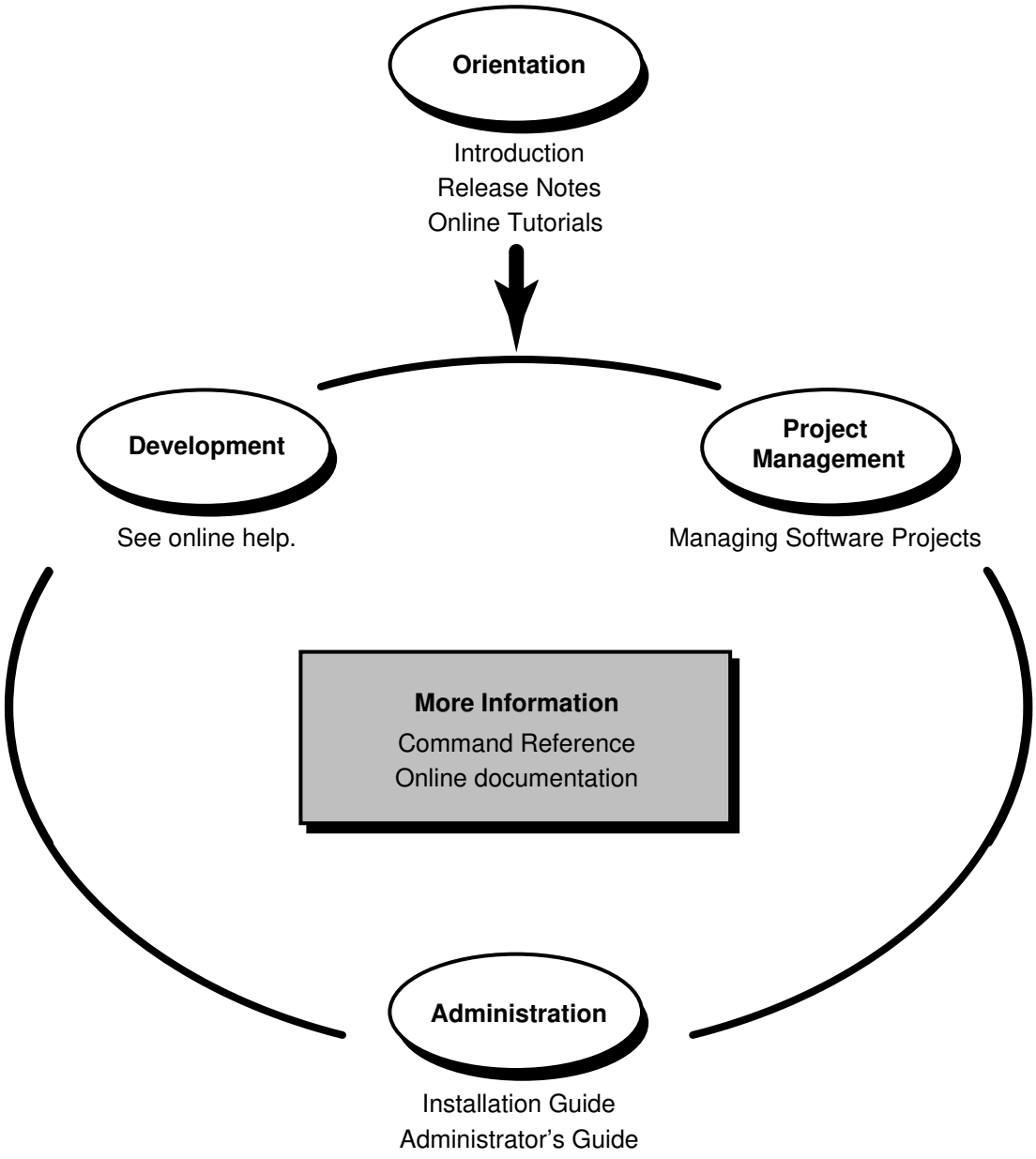
In UCM, project managers and administrators set up projects and components, determine policy, establish and promote baselines, and manage the integration stream.

In base ClearCase LT, project managers and administrators determine policy and create branch types, config specs, triggers, and labels to enforce the policies.

For more information, see the online help and *Managing Software Projects*.

Getting Information About Administrator Tasks

Administrators create and maintain VOBs and manage VOB and view storage, licensing, and the registries for ClearCase LT. For more information about administrative tasks, see the online help and the *Administrator's Guide* for Rational ClearCase LT.


Orientation
Introduction
Release Notes
Online Tutorials
Development
See online help.
Project Management
Managing Software Projects
More Information
Command Reference
Online documentation
Administration
Installation Guide
Administrator's Guide