DEVELOPING SOFTWARE WITH CLEARCASE

Release 4.1 and later

Windows Edition

Rational®
the e-development company™

800-023554-000

Developing Software with ClearCase
Document Number 800-023554-000 August 2000
Rational Software Corporation 20 Maguire Road Lexington, Massachusetts 02421

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 software and documentation is based in part on software written by Victor A. Abell while at Purdue University. We acknowledge his role 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/).

Rational®
the e-development company™

Preface

Rational ClearCase is a comprehensive configuration management (CM) system that manages multiple variants of evolving software systems. ClearCase maintains a complete version history of all software development artifacts, including code, requirements, models, scripts, test assets, and directory structures. It performs audited system builds, enforces site-specific development policies, offers multiple developer workspaces, and provides advanced support for parallel development. ClearCase 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 guides software developers through everyday development tasks using either UCM or customizable features of ClearCase.

Organization

The manual is divided into two parts:

- *Working in UCM*. Read this part only if your development team uses UCM to implement its development process.

- *Working in Base ClearCase*. Read this part only if your development team uses customizable ClearCase features to implement its own development process.

Recommended Reading Paths

Use this manual as a guide during your first few weeks of developing software with ClearCase. We recommend this sequence for proceeding:

- Skim through this manual to understand at what points in the development cycle you'll need to use ClearCase. Consider skipping the sections titled *Under the Hood*, which describe advanced concepts and technical details.
- Read through this manual carefully and follow the procedures.
- As you become familiar with ClearCase, read the sections titled *Under the Hood*.

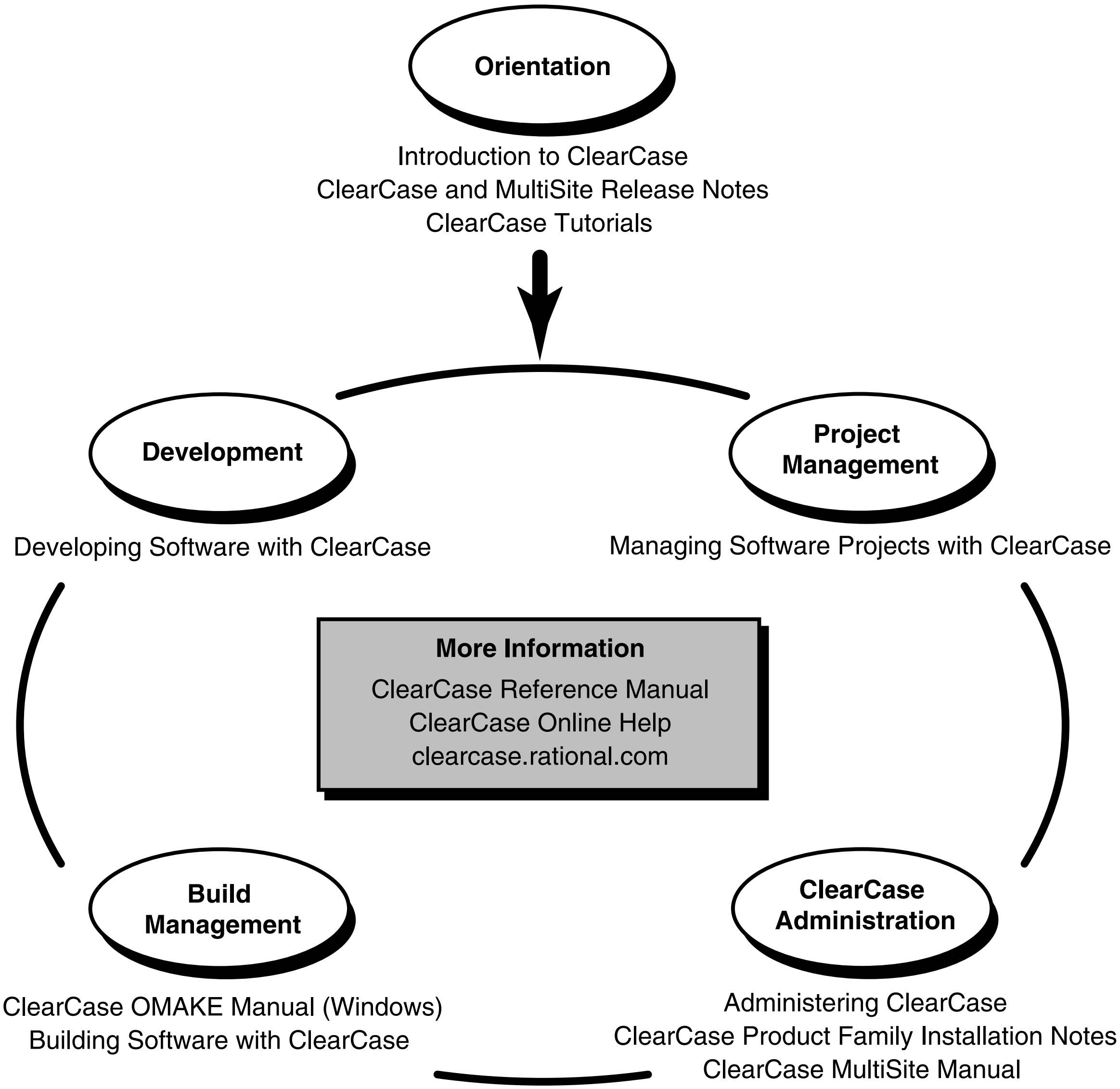
The manual describes using ClearCase primarily from ClearCase Explorer. In addition to the ClearCase Explorer interface, ClearCase is integrated with Windows Explorer (via shortcut menus for ClearCase objects) and many authoring and development environments. Read this manual to understand the basics of using ClearCase; then refer to ClearCase online help for information on using ClearCase from your authoring or development environment. For a complete list of ClearCase integrations, refer to the Rational Web site at **www.rational.com**.

Assumptions

This manual assumes:

- ClearCase has been installed and configured on your computer. For information on installing ClearCase, see *ClearCase Product Family Installation Notes*.
- One or more *versioned object bases* (VOBs), which are the repositories for ClearCase data, exist in your organization.
- If your development team uses ClearCase UCM, your project manager has set up a UCM *project*.
- Your organization has established some other development strategy using base ClearCase as the configuration-management system, and you are familiar with this strategy.

ClearCase Documentation Roadmap



Typographical Conventions

This manual uses the following typographical conventions:

- *ccase-home-dir* represents the directory into which the ClearCase Product Family has been installed. By default, this directory is **/usr/atria** on UNIX and **C:\Program Files\Rational\ClearCase** on Windows.

- *attache-home-dir* represents the directory into which ClearCase Attache has been installed. By default, this directory is **C:\Program Files\Rational\Attache**, except on Windows 3.*x*, where it is **C:\RATIONAL\ATTACHE**.

- **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.

- ... In a syntax description, an ellipsis indicates you can repeat the preceding item or line one or more times. Otherwise, it can indicate omitted information.

  NOTE: In certain contexts, ClearCase recognizes "**...**" within a pathname as a wildcard, similar to "*" or "?". See the **wildcards_ccase** reference page for more information.

- If a command or option name has a short form, a "medial dot" (·) character indicates the shortest legal abbreviation. For example:

  lsc·heckout

  This means that you can truncate the command name to **lsc** or any of its intermediate spellings (**lsch**, **lsche**, **lschec**, and so on).

Online Documentation

The ClearCase graphical interface includes a standard Windows help system.

There are three basic ways to access the online help system: the **Help** menu, the **Help** button, or the F1 key. **Help>Help Topics** 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 lsprivate –help**

```
Usage: lsprivate [-tag view-tag] [-invob vob-selector] [-long | -short]
                 [-size] [-age] [-co] [-do] [-other]
```

Additionally, the online *ClearCase Tutorial* provides important information on setting up a user's environment, along with a step-by-step tour through ClearCase's most important features. To start the *ClearCase Tutorial*, choose **Tutorial** in the ClearCase folder off the **Start** menu.

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**

Working in UCM

Contents

Working in UCM

Figures

1 The UCM Workflow

Unified Change Management (UCM) structures the efforts of your software development team into a defined, repeatable process. This chapter describes the UCM workflow and how it affects your work as a developer.

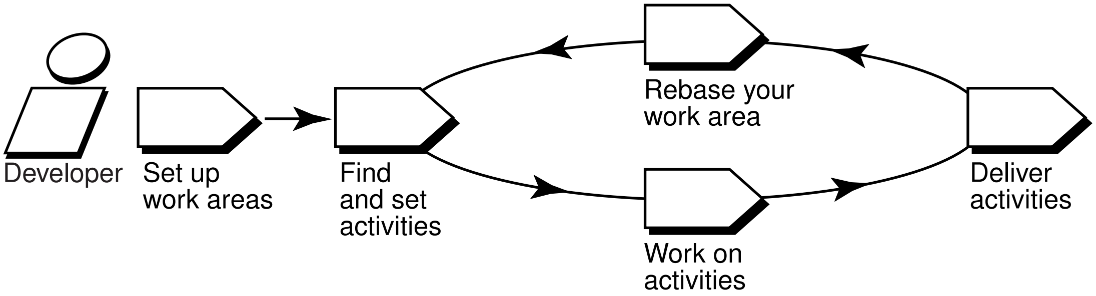
Recommended Reading Paths

Read this chapter first. Then, if you want to start working immediately, use online help to learn as you go. Or, if you prefer a more structured approach, use the remainder of *Working in UCM* as a guide through the development cycle.

Until you are familiar with using Rational ClearCase, consider skipping the sections titled *Under the Hood*, which describe advanced concepts and technical details.

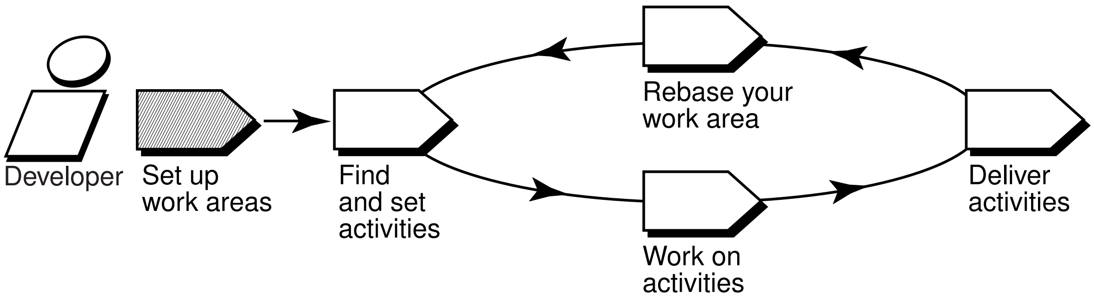
1.1 The UCM Workflow

When your project uses UCM, your work as a software or information developer follows a cycle.



To access your project's source files, you set up work areas. As you modify source files, you use activities to organize and identify your work. Other developers on the project do not see your work until you deliver your activities to a shared work area. Periodically, the project manager incorporates activities in the shared work area into baselines, which are sets of activities that represent a significant change. Then, you synchronize (or rebase) your work area with the activities in the new baseline.

Setting Up Work Areas

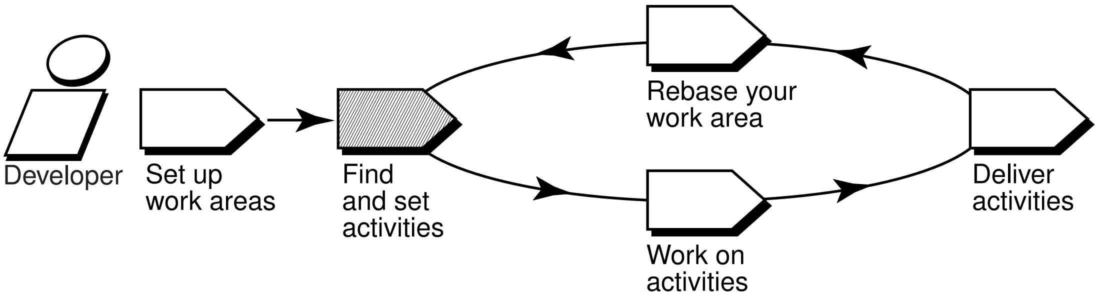


To contribute work to a UCM project, you must set up two work areas: a development area for working on activities in isolation and an integration area for testing the activities you deliver.

Each work area consists of a view, which provides a directory tree of source files for you to modify, and a stream, a ClearCase administrative object that configures the view and keeps track of the activities you create and deliver. For more information, see *Views and Streams* on page 11.

You can access your views and activities through ClearCase Explorer, your primary interface with ClearCase.

Finding and Setting Activities



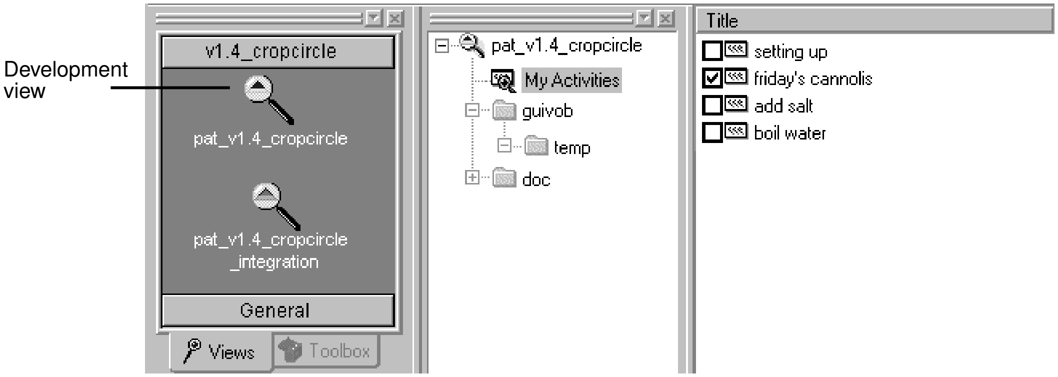
After you set up your work areas, find any activities that your project manager or other team members have assigned to you; if necessary, you can add new ones.

Finding Activities

Clicking the **MyActivities** icon in ClearCase Explorer shows the activities that are in your work area (Figure 1). If your project uses Rational ClearQuest as its change-request management system, you can use ClearQuest queries such as MyToDoList to find activities that are assigned to you but are not yet in your work area. For more information, see *UCM-Enabled Record Types and ClearCase Activities* on page 18 and *Queries* on page 19.

When you're ready to work on an activity, you must set it in your development view.

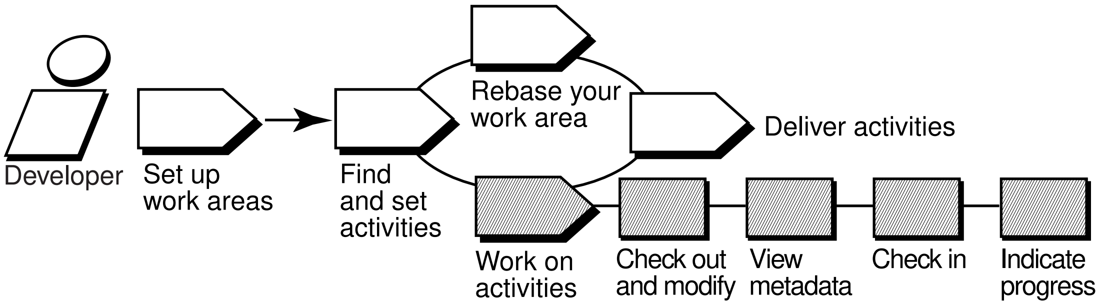
Figure 1 MyActivities in ClearCase Explorer



Setting Activities

Setting your view to an activity instructs ClearCase to assign any versions of source files you create to the activity's change set. You must set your view to an activity before you can check out and modify source files.

Working on Activities



To work on activities, you check out and modify source files and test your modifications. You may also need to enter information in your change-request management system to communicate your progress.

Checking Out and Modifying Source Files

Checking out a file makes it writable in your view. Then you can use any editing tool to modify it.

Viewing Metadata

Metadata is information that ClearCase and ClearQuest keep about your activities and source files. While working with checkouts, you may want use ClearCase and ClearQuest to view the following types of metadata:

- ➤ The revision history of an element
- ➤ A comparison between versions of an element
- ➤ A list of versions that are checked out to your work area

Checking In and Testing Source Files

When you want to keep a record of a file's current state, check it in. Any work you check in from your development view is not available to other team members until you deliver it.

Build and test the work in your development view, especially after you rebase the stream and before you deliver activities, to reduce the amount of merging needed when you deliver your work.

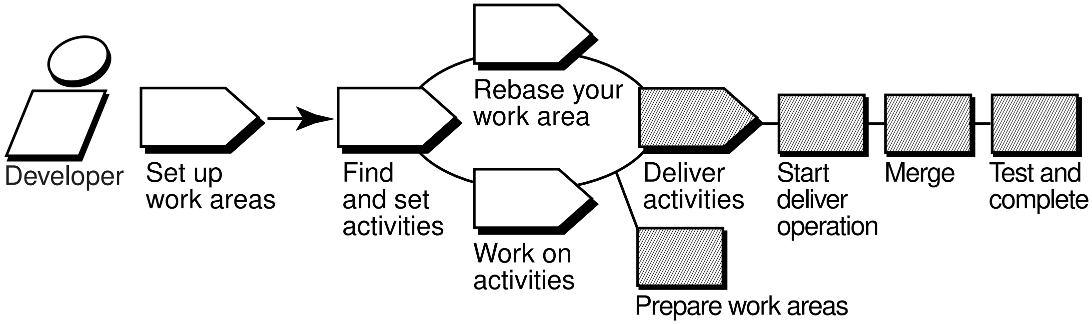
Indicating Your Progress

If your project is enabled for ClearQuest, you can indicate your progress on an activity in several ways:

- ➤ Modifying information in ClearQuest
- ➤ Closing an activity
- ➤ Using schema-specific actions

You may need to do reassign or delete an activity, regardless of whether your project is enabled for ClearQuest. For more information, see *Indicating Your Progress* on page 67. When you're ready to make your work available to the rest of the project, deliver it.

Delivering Activities



When you're ready to make one or more activities available to the project team, prepare your work areas. Then start the deliver operation.

Preparing Your Work Areas

Do the following to prepare your work areas:

➤ If your project manager has created a new recommended baseline since you last rebased, rebase your development area.

➤ Find, compare, and check in the work you want to deliver.

Starting the Deliver Operation

After preparing your work areas, start the deliver operation.

Merging

As part of the deliver operation, ClearCase merges the work in your development stream with the work in the integration stream. It completes trivial merges for you. If it encounters merge conflicts, it prompts you to resolve the conflicts.

Testing

Your integration view contains the merge results. To make sure that your delivered work is compatible with the work in the integration stream, build and test the files in the integration view.

In addition to building and testing, you may need to do the following:

- Edit the checked-out versions to resolve build errors.
- Check out and edit additional files.

  NOTE: We recommend that you do not check in any of your changes until you are ready to complete the operation. Checking in complicates efforts to undo the deliver operation.

- Update your integration view to see the work other developers have delivered since you started the deliver operation.

Completing the Operation

When you are satisfied with your test builds, complete the delivery. ClearCase checks in your modifications and changes the state of your stream.

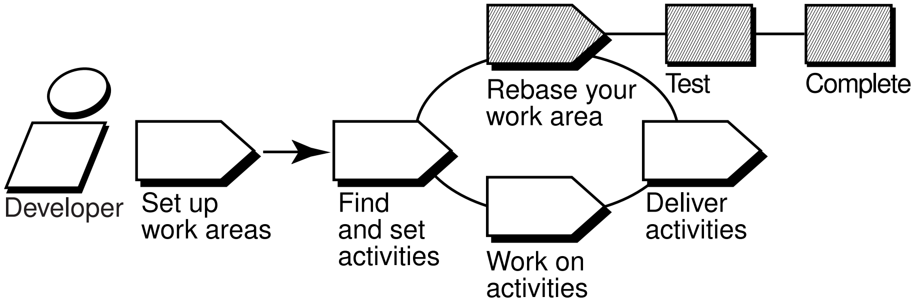
Delivering with MultiSite

If your project uses MultiSite to share source data with developers in other geographical locations, you may use a different method for delivering activities. If a different site is responsible for controlling your project's source data, you do the following:

- Post a list of activities that are ready to be delivered.
- Notify the project manager at the site that controls your project's source data.

Then the project manager at the other site merges your activities to the integration stream and tests your work.

Rebasing Your Development Work Area



The project manager organizes delivered activities into baselines. Usually, baselines go through a cycle of testing and fixing bugs until they reach a satisfactory level of stability. When a baseline reaches this level, your project manager designates it as a recommended baseline.

To work with the set of versions in the recommended baseline, you rebase your development work area. To minimize the amount of merging necessary when you deliver activities, rebase your development work area with each new recommended baseline as it becomes available.

After you rebase, be sure to build and test the source files in your development view to verify that your undelivered activities build successfully with the versions in the baseline.

1.2 Under the Hood: ClearCase UCM Concepts

This section describes fundamental ClearCase UCM concepts:

- Projects
- Components and baselines
- Activities
- Views and streams
- Delivering, creating baselines, and rebasing

Reading this section isn't necessary to start working on a UCM project, but it does provide a basis for understanding how ClearCase implements UCM and may be helpful if you need to diagnose problems.

Projects

In ClearCase, *project* refers to an object that defines a set of development policies and a set of configurations used in a significant development effort. Your organization may create a project for each product it develops, for a group of products, for a subset of a product's functionality, or for a product release.

A project's policies govern how developers access and modify sets of source files and directories (called components). To record and configure the development work that proceeds on components, projects use the following ClearCase objects:

- Baselines
- An integration stream
- Development streams
- Activities

Because ClearCase supports *parallel development*, different projects can work concurrently with different versions of the same set of source files.

ClearCase stores projects in a data repository called a *PVOB* (project versioned object base).

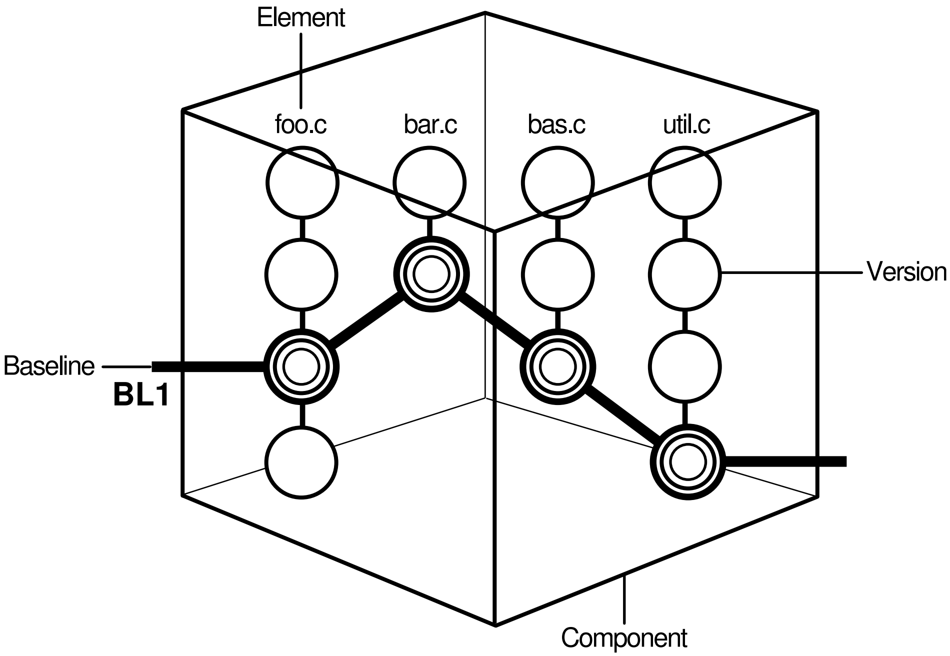
Components and Baselines

Your project manager organizes files and directories that are under source control into components. Usually, components can be built into functional units of software.

Files and directories in a component are called elements, and each checked-in revision of an element is called a version.

To keep track of different configurations of versions, your project manager creates a baseline, which records one version of each element in a component. In effect, a baseline is a version of a component (Figure 2).

Figure 2 Elements, Components, and Baselines



ClearCase stores elements (and their versions) in data repositories called *versioned object bases* (VOBs); it stores component and baseline definitions in PVOBs.

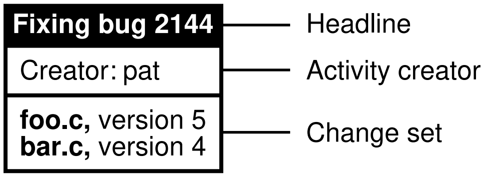
Activities

To create a version in a ClearCase UCM project, you must first assign it to an activity. An activity is a ClearCase object that identifies the versions created to complete a development task. For example, the versions you create to fix a defect that is stored in your change request management system may constitute an activity. Your organization determines the scope of its activities.

The ClearCase activity object includes a text headline, which describes the task, the user ID of the activity creator, and a change set, which identifies all versions that you create or modify while working on the task (Figure 3). (For more information on an activity's headline, see *Under the Hood: Different Ways to Refer to Objects in a Project* on page 27.)

NOTE: In a project that is enabled for ClearQuest, ClearQuest includes a field to describe the activity owner. This activity owner and the ClearCase activity creator are two different data points; the former is stored in ClearQuest and the latter in ClearCase.

Figure 3 Activity Object



An activity object belongs to a single stream and cannot be moved from one stream to another. (If you assign one or more versions to the wrong activity or if you create new activities to better represent your work, you can assign the versions to a different activity. For more information, see *Moving Versions to a Different Activity* on page 62.)

When you deliver an activity, ClearCase merges the versions in the activity's change set to the integration stream, but it does not move the activity object to the integration stream. Instead, it creates an integration activity to identify the versions created as a result of the merge. For more information, see *Under the Hood: Integration Activities and Baselines* on page 80.

In a project that is enabled for ClearQuest, ClearCase links its activity objects to UCM-enabled records in ClearQuest. For more information, see *UCM-Enabled Record Types and ClearCase Activities* on page 18.

Views and Streams

To create new versions of elements in a component, you use a view; to keep track of the versions you create, you use a stream.

Views

A view provides a directory tree of one version of each file in one or more components. In the view, you modify source files, compile them into object modules for testing purposes, format them into documents, and so on. Figure 4 shows a view named **pat_v1.4_cropcircle** as seen from ClearCase Explorer.

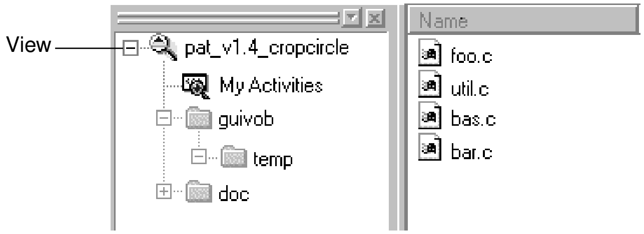
ClearCase offers two kinds of views:

➤ *Snapshot views*, which copy files from VOBs to your computer.

➤ *Dynamic views*, which provide immediate, transparent access to the data in VOBs. (Dynamic views are not available on all platforms. For more information, see the ClearCase online help.)

An associated stream determines which versions of elements are in the view.

Figure 4 A View as Seen from ClearCase Explorer

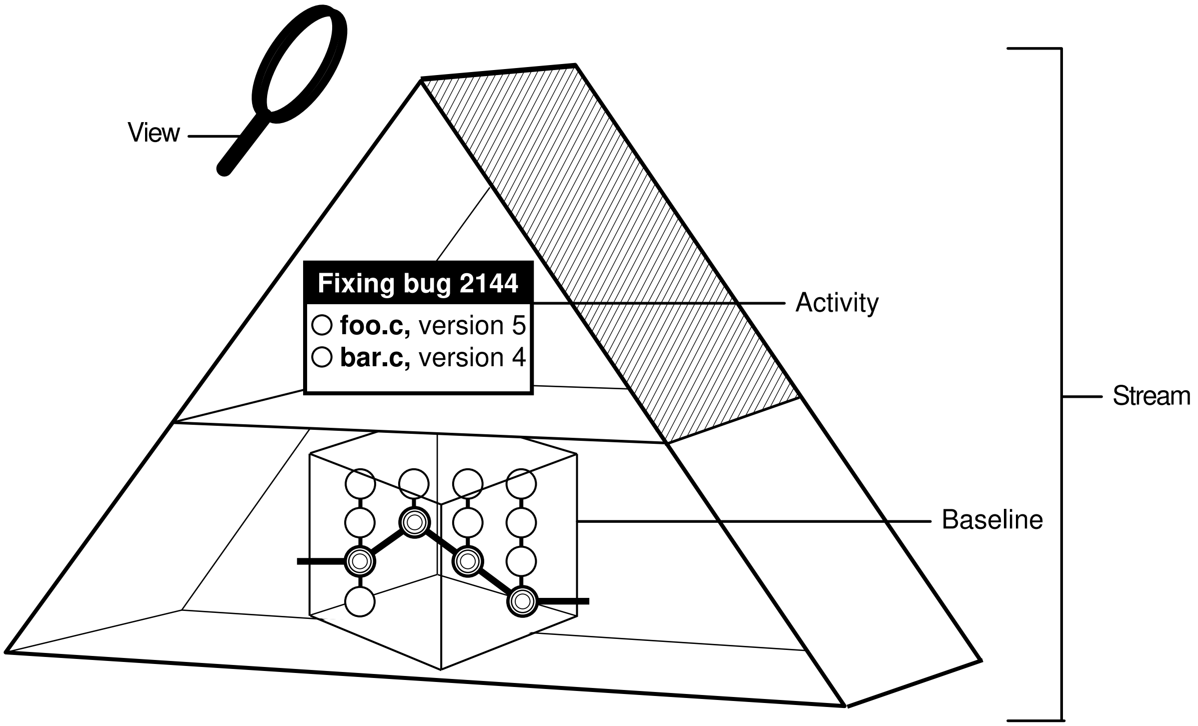


Streams

A stream is a long-lived ClearCase object. It is a member of a single UCM project and is a mechanism for creating and recording configurations. A stream identifies the exact set of versions currently available for you to view, modify, or build.

UCM uses baselines and activities to encapsulate a stream's configuration. When you create a stream, its original configuration is the same as a baseline (that is, it identifies a single version of each element in a component). When you create new versions of elements, you assign the new versions to one or more activities. Hence, a stream's configuration is a given baseline plus one or more activities (Figure 5).

Figure 5 A Stream's Configuration



The following actions modify a stream's configuration:

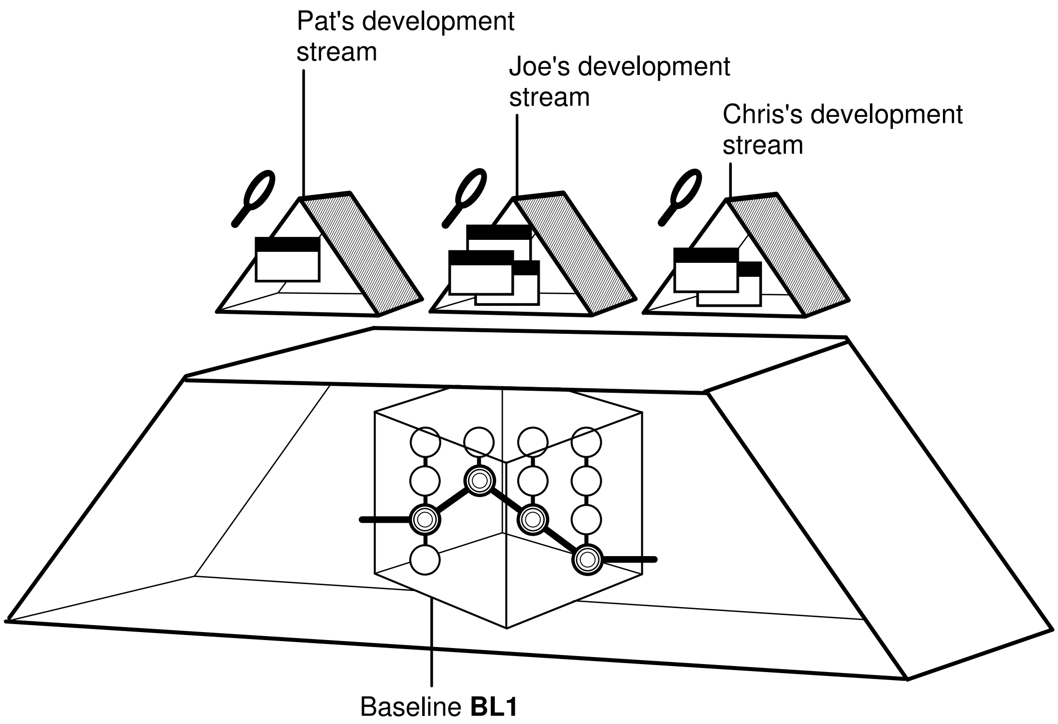
- Checking in versions from an associated view. (A stream may be associated with several views.)
- Rebasing, which replaces the baseline in the stream's configuration with a more recent one.
- Delivering activities, which changes the configuration of the integration stream by adding activities that were previously available only to the contributing development stream. (Delivering activities does not modify the development stream.)

A project includes two kinds of streams: development streams and integration streams.

Development Stream

Typically, each project includes many development streams (one for each developer on the project), all of which start from a baseline and evolve separately as developers add activities (Figure 6.)

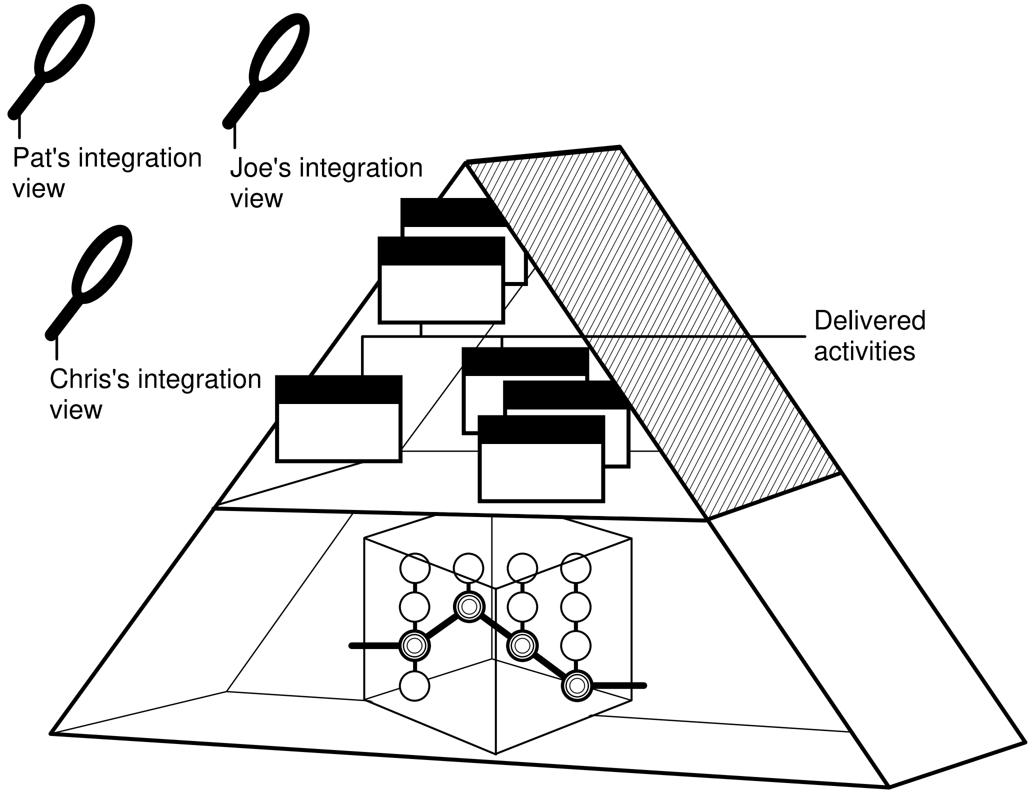
Figure 6 Many Development Streams



The Integration Stream

Projects always include a single integration stream to collect the work team members deliver from their development streams (Figure 7).

Figure 7 Integration Stream

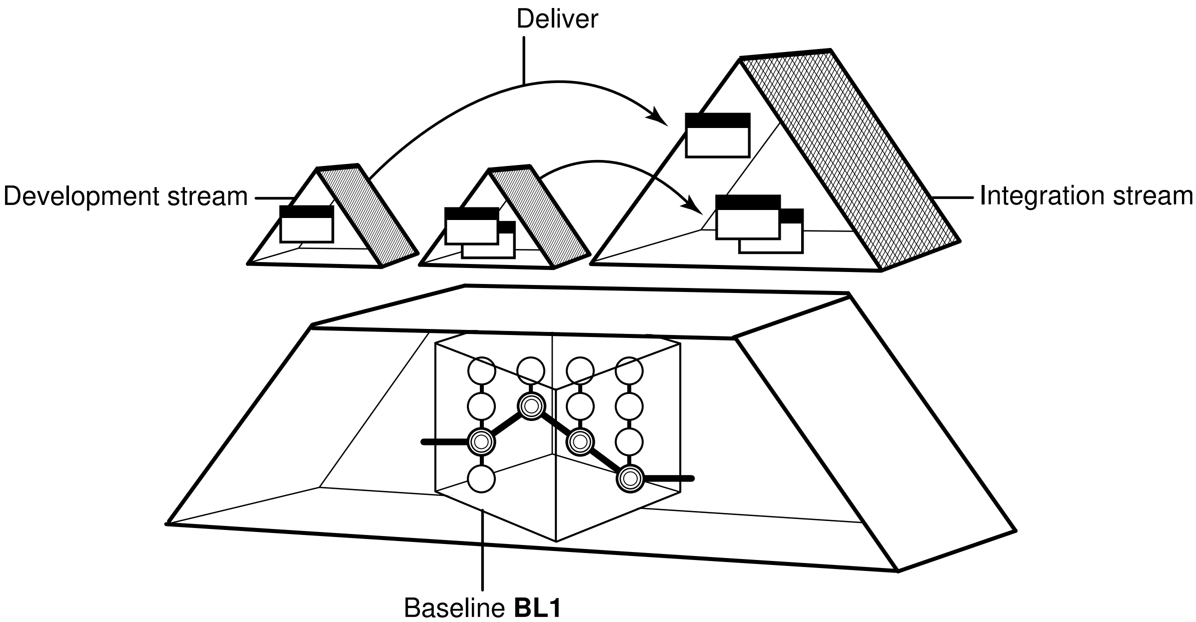


To facilitate delivering activities, each team member associates a separate integration view with the project's integration stream. The integration views show the baseline and all delivered activities.

Delivering, Creating Baselines, and Rebasing

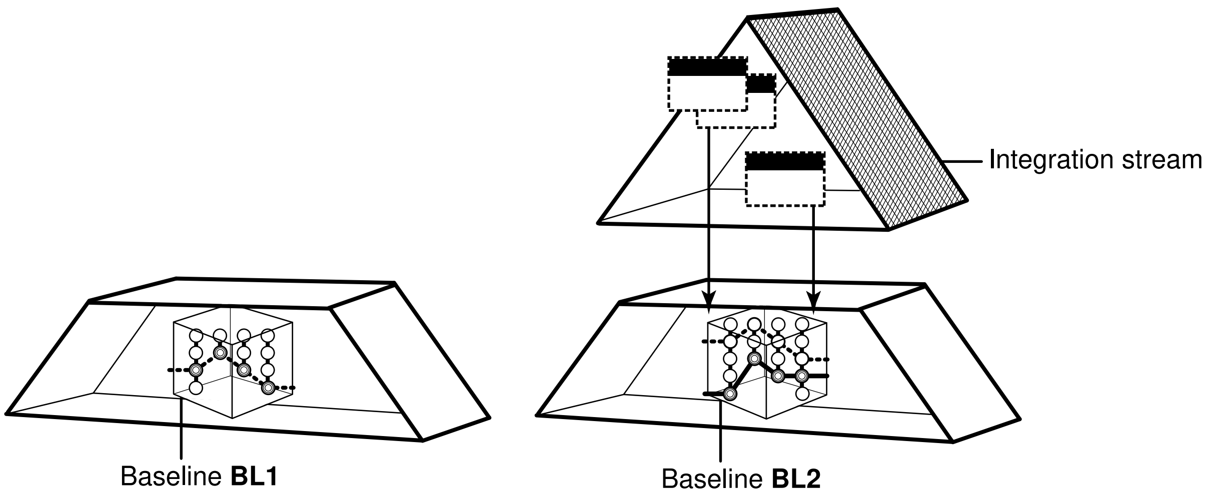
Developers on a project deliver activities from their development streams to a single integration stream, which identifies a shared set of versions to be used for projectwide building and testing (Figure 8).

Figure 8 Delivering to the Integration Stream



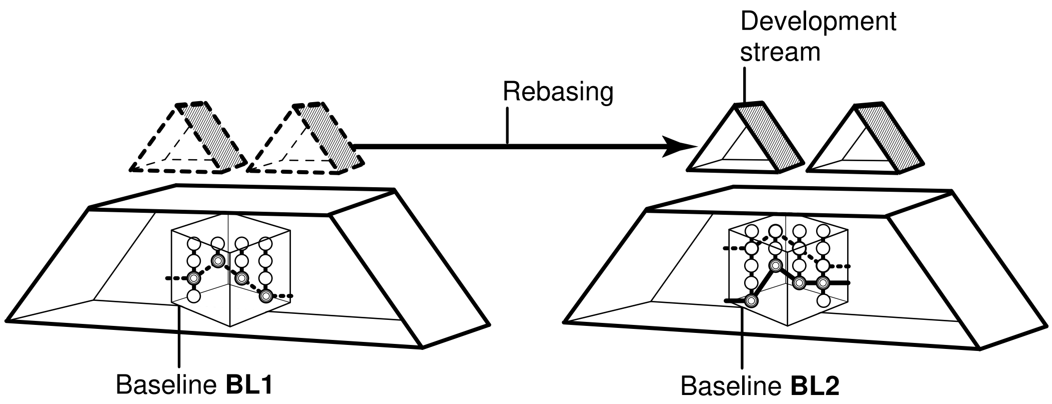
Project managers collect the activities delivered to the integration stream into baselines (Figure 9.) They characterize specific baselines that include stable and significant changes as *recommended*.

Figure 9 Creating a Baseline



Developers rebase their development work areas to see the new work identified by the recommended baselines. After rebasing, a development work area shows the versions specified by the baseline, plus any of your undelivered activities (Figure 10.)

Figure 10 Rebasing Work Areas



1.3 Under the Hood: UCM ClearQuest Concepts

Rational ClearQuest is a change-request management system that can integrate with UCM projects to provide extended functionality. This section describes fundamental concepts for the UCM-ClearQuest integration:

- ClearQuest schema
- UCM-enabled record types and ClearCase activities
- Queries
- State types and state transitions

Reading this section isn't necessary to start working on a UCM project that is enabled for ClearQuest, but it does provide a basis for understanding how ClearQuest implements the integration and may be helpful if you need to diagnose problems.

The UCM-ClearQuest Schema

ClearQuest stores change-requests as records in a database. A *schema* defines the types of records in the database, the states available to each type of record, and other attributes of the database.

ClearQuest supplies two UCM-enabled schemas, Unified Change Management and Enterprise; your project manager can set up a custom UCM-enabled schema. To support the UCM-ClearQuest integration, a ClearQuest database must use one of these UCM-enabled schemas.

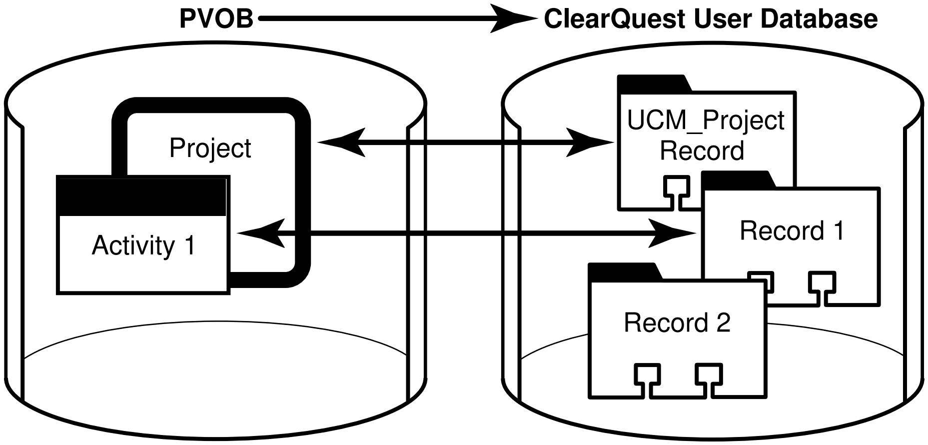
ClearQuest stores all schemas in a schema repository, which is a master database of schemas. The schema repository does not contain any user-owned data or change request data. Change request data is stored in a user database.

UCM-Enabled Record Types and ClearCase Activities

A UCM-enabled record type is a template for ClearQuest records. It includes definitions for a set of fields that store information about ClearCase activities. Your ClearQuest database may include different UCM-enabled record types for different purposes (such as tracking defects and enhancement requests), and it may include other record types that are not UCM enabled.

In a project that uses the UCM-ClearQuest integration, records based on a UCM-enabled record type can be linked with ClearCase activity objects (Figure 11). This link enables ClearQuest to display information about the ClearCase activity (such as its change-set, its stream, and whether it is currently set in any view). It also enables policies governing when you can deliver an activity in ClearCase and when you can close an activity in ClearQuest. Because of the close association between linked UCM-enabled records and ClearCase activities, the UCM documentation usually refers to both linked entities as *activities*.

Figure 11 Activities and UCM-Enabled Records



At any point in a project, your ClearQuest database may contain UCM-enabled records that are not linked to a ClearCase activity object. For example, a newly created record may not be linked to a ClearCase activity. You must explicitly complete an action (for example, by clicking **Action>WorkOn** in ClearQuest) to link a UCM-enabled record to a ClearCase activity.

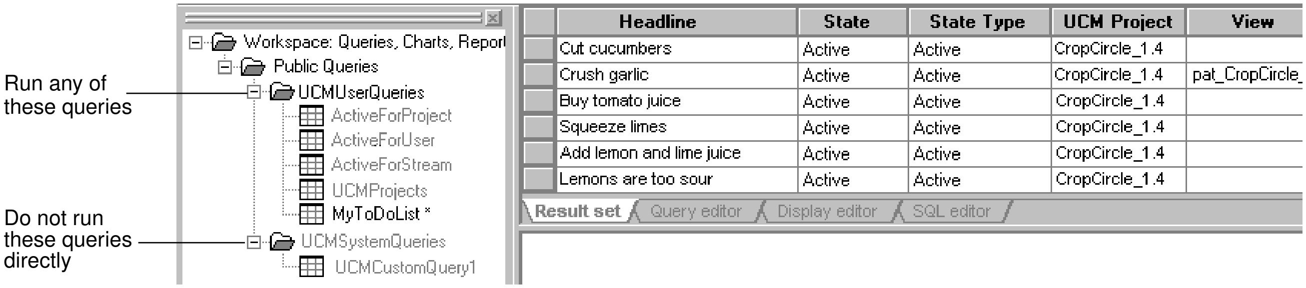
Each ClearCase activity in a UCM project that is enabled for ClearQuest must be linked to a ClearQuest record. You cannot create a ClearCase activity object without linking it to a UCM-enabled record in a ClearQuest database.

Queries

Queries are the vehicle for navigating through the ClearQuest database. Before viewing or modifying records, you must query the database to find the records you are interested in. For example, you may want to see only activities that are assigned to you or only those activities that are associated with your project, or you may want to see activities created before or after a particular date.

A UCM-enabled schema includes a set of queries that you can use to find UCM-enabled records. ClearQuest places the queries into two categories in the ClearQuest Public Queries folder (Figure 12).

Figure 12 UCM-Enabled Queries in ClearQuest



You can run queries in the UCMUserQueries folder and modify them when necessary. You can save any public query you modify in your Private Queries folder.

Do not run queries in the UCMSystemQueries folder from ClearQuest; they are intended to be run by the ClearQuest and ClearCase system only.

This section describes the following queries:

➤ MyToDoList
➤ UCMCustomQuery1
➤ Other queries

In addition to the queries included in the database schema, you and your project manager can create other queries.

MyToDoList

The MyToDoList query finds UCM-enabled records that all of match the following criteria:

➤ They are assigned to you.
➤ They are in a state whose state type is Ready or Active.

You or your project manager can modify this query.

UCMCustomQuery1

The UCMCustomQuery1 query finds UCM-enabled records that match any of the following criteria:

- In a state whose state type is **Ready** and match all of the following criteria:
  - Are assigned to you
  - Have not yet been assigned to a UCM project
  - Have not yet been worked on

- In a state whose state type is Ready or Active and match all of the following criteria:
  - Are assigned to you
  - Have been assigned to the UCM project associated with the current view
  - Have not yet been worked on

- In a state whose state type is Active and match all of the following criteria:
  - Are assigned to you
  - Have been assigned to the UCM project associated with the current view
  - Have been worked on already in the stream associated with the current view

MyActivities in ClearCase Explorer and the **Check Out**, **Check In**, and **Add to Source Control** dialog boxes in ClearCase present the activities that UCMCustomQuery1 finds. Although you can also see this query from ClearQuest, it is not designed to be run outside the context of a ClearCase view.

Your project manager can modify this query.

Other Queries

A UCM-enabled schema also supplies the queries described in Table 1. You can run any of these queries.

Table 1 Other Queries in the Unified Change Management Schema

Query	Description
ActiveForProject	For one or more specified projects, selects all activities in an active state type.
ActiveForStream	For one or more specified streams, selects all activities in an active state type.
ActiveForUser	For one or more specified developers, selects all assigned activities in an active state type.
UCMProjects	Selects all UCM-project records in ClearQuest user database.

State Types and State Transitions

Change requests move through a pattern, or life cycle, from submission through resolution. In ClearQuest, each stage in this life cycle is called a *state*, and each movement from one state to another is called a *state transition*.

As with record types and records, a state type is a template that defines actions and other attributes associated with a state. The states in a UCM-enabled schema must be based on one of the following state types:

- Waiting
- Ready
- Active
- Complete

Your project manager may give the states in your UCM-enabled schema different names and may create multiple, different states based on the same state type. For example, your UCM-enabled schema may contain the states Scheduled and Deferred, both of which are based on the Active state type but have different associated actions and meanings.

State Transitions

Schemas include rules for changing records from one state type to the next. In a UCM-enabled schema, records must follow these state-transition rules (Figure 13):

- **Waiting**. Creating an activity places it in a state based on this state type to indicate, for example, that the work is waiting for someone to resolve dependencies and that it is not ready to be scheduled. Activities in this state do not appear in MyActivities or MyToDoList; they may or may not contain a value in the owner field and may or may not contain a value in the UCM project field. The schema in Figure 13 bases its Submitted and Postpone states on the Waiting state type.

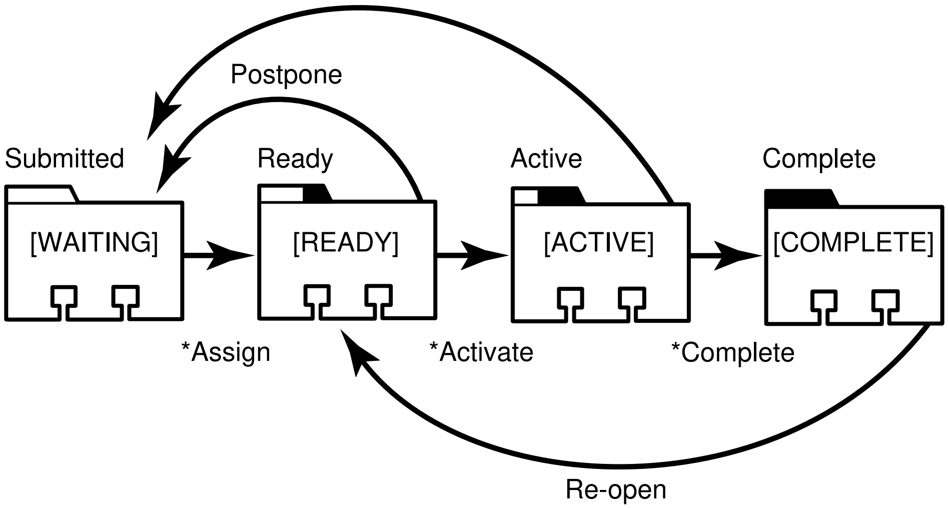
- **Ready**. Assigning an activity usually places it in a state based on this state type to indicate that the work is pending. Activities in this appear in the activity owner's MyActivities and MyToDoList, but are not part of the owner's stream. The schema in Figure 13 bases its Ready and Re-open states on the Ready state type.

- **Active**. Setting your view to an activity usually places the activity in a state based on this state type to indicate that the work is in progress. Setting your view to an activity also links the ClearQuest UCM-enabled record to the ClearCase activity. After the activity is linked, you can change the ClearQuest record's owner but you cannot change the ClearCase stream that contains the activity. For more information, see *Reassigning Activities* on page 70.

  The **ucm_stream** field in a UCM-enabled record displays the stream to which an activity is linked, and the **ucm_view** field indicates whether an activity is currently set in a view. (Figure 19 on page 43) The schema in Figure 13 bases its Active state on the Active state type.

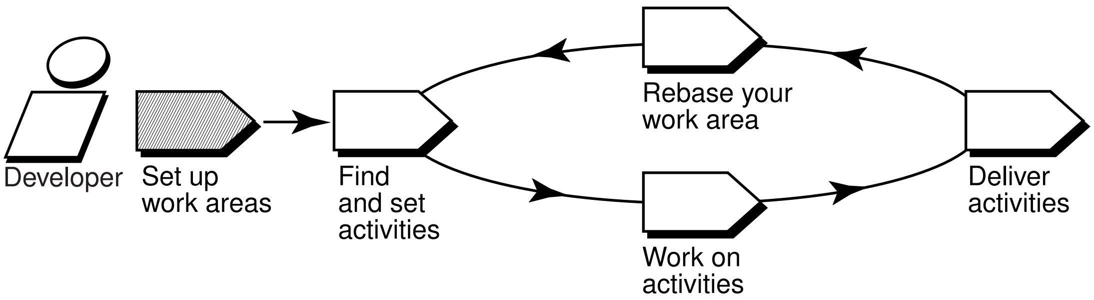
- **Complete**. When you have completed and delivered an activity, you change it to a state based on the Complete state type. Activities in this state do not appear in MyToDoList, but do appear in **MyActivities** and remain a permanent part of your stream. The schema in Figure 13 bases its Complete state on the Complete state type

Figure 13 Example of ClearQuest State Transitions



2

Setting Up Work Areas



To set up work areas, you must complete the following tasks:

- Starting the Join Project Wizard
- Choosing a project
- Verifying stream names
- Setting up views
- Choosing components to load into snapshot views
- Accessing your development view
- Connecting to the UCM-ClearQuest integration (only for projects that use the UCM-ClearQuest integration)

2.1 Starting the Join Project Wizard

The Join Project Wizard assists you in each step of joining a project. Start the Join Project Wizard and use this chapter to complete the steps.

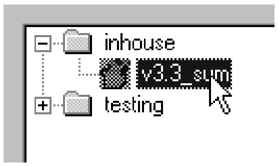
To Start the Join Project Wizard

1. Start ClearCase Explorer by clicking the shortcut on your desktop or, if you did not install the shortcut, by clicking **Start>Programs>ClearCase>ClearCase Explorer**.
2. In the ClearCase Explorer Shortcut pane, click **Toolbox**. Then click **UCM>Join Project**.

2.2 Choosing a Project

As its first step, the Join Project Wizard prompts you to choose a project. Navigate the list of projects and click the one you want to join (Figure 14).

Figure 14 List of Projects



To see a description of a project, select it and click **Properties**.

2.3 Verifying Stream Names

After you choose a project, the Join Project Wizard presents the two streams that ClearCase uses to keep track of your work on this project. Take note of the stream titles when the wizard presents them; they are permanent objects that keep track of your work on the project.

Development Stream

The Join Project Wizard gives your development stream a title based on the following convention: *userID_project-name*. For example, Pat's development stream for the 1.4 release of the Cropcircle project is titled **pat_1.4_cropcircle**.

If ClearCase cannot determine your user ID, it substitutes the name of the ClearCase host from which you are creating the stream.

Although you can change the title of your development stream on this step of the Join Project Wizard, we recommend that you do not, unless your project manager has requested it. Following consistent conventions facilitates project management.

Integration Stream

Your project manager creates the integration stream as part of creating the project. The Join Project Wizard presents the title of the project's integration stream for informational purposes only.

Under the Hood: Different Ways to Refer to Objects in a Project

Project objects, such as streams and activities, have two types of identifiers:

➤ Titles or headlines provide a way to refer to the object that is easy to remember and meaningful to you. Titles (which apply to projects, streams, and folders) and headlines (which apply to *activities*) do not need to be unique among all objects in a PVOB.

➤ Object-selectors provide an identifier that is unique among all objects in all PVOBs in your organization. An object-selector uses the following syntax:

  [*object-type*:] *alphanumeric-string* [@*PVOB-tag*]

  The alphanumeric string must be unique to the current PVOB; ClearCase provides one if you do not. For example, the selector ClearCase provides for the **pat_1.4_cropcircle** stream is **stream: pat_1.4_cropcircle_02442@\projects**.

When you use ClearCase and ClearQuest GUIs, you need to know only an object's title or headline. For example, when the Project Explorer lists the streams and activities for a given project, it displays the stream titles and the activity headlines. These identifiers may not be unique, but are usually sufficient for the context in which they are displayed.

When you use the command-line interface (CLI), which gives you greater flexibility, you must use the object-selector to refer to streams and activities. If you plan to work primarily from the command line, we recommend that you supply meaningful alphanumeric strings when you create and work on objects, rather than use the ClearCase default string, which is unique, but may not be as easy to remember.

Setting Activities in Projects that Are Enabled for ClearQuest

When using the command line to set activities in projects that are enabled for ClearQuest, you must use the record ID of the associated ClearQuest record (for example, SAMPL123456) instead of the ClearCase object-selector.

2.4 Setting Up Your Views

In the next steps of the Join Project Wizard, you set up your development view and integration view by specifying the following information:

- Which type of view to create for your development view and integration view
- Locations for snapshot views
- Drive letters for dynamic views (optional)
- Locations for dynamic view storage directories

We recommend that you accept the defaults the Join Project Wizard presents until you are familiar with UCM. For example, the default naming conventions for your views are intended to distinguish your development view from your integration view.

Determining the View Types

As described in *Views* on page 11, you can use either a snapshot view or a dynamic view to create a directory tree of source files. Your project manager determines the default view types that the Join Project Wizard presents.

Work in a snapshot view when any of these conditions is true:

➤ Your computer does not support dynamic views.

➤ You want to optimize build performance to achieve native build speed.

➤ You want to work with source files under ClearCase control when you are disconnected from the network that hosts the VOBs.

➤ You want to access a view from a computer that is not a ClearCase host.

➤ Your development project doesn't use the ClearCase *build auditing* and *build avoidance* features.

Work in a dynamic view when any of these conditions is true:

➤ Your development project uses build auditing and build avoidance.

➤ You want to access elements in VOBs without copying them to your computer.

Because dynamic views can show changes in its associated stream at all times without requiring an update, we recommend that you use dynamic views for your integration views whenever they are available.

The two view types behave slightly differently, so remember the type of view you create.

Your project manager, or someone else in your organization, determines the types of views the Join Project Wizard creates by default based on network configuration and performance needs. For a detailed comparison of snapshot views and dynamic views, see the **view** reference page in *ClearCase Reference Manual*.

Locations for Snapshot Views

ClearCase loads, or copies, versions of source files from VOBs into the snapshot view at the location you specify.

When choosing a location, consider these constraints:

- The view's root directory must be located on a disk with enough space for the files loaded into the view and any other files you add. Files and directories that you add to the view but do not add to source control are called view-private files and view-private directories.

- Your organization may restrict where you can create a view. For example, you may be required to use a disk that is part of a data-backup scheme.

If your makefiles or other files require absolute pathnames with a specific drive letter, you can assign the view to a drive letter. See *Assigning Snapshot Views to Drive Letters* on page 108.

Under the Hood: A Snapshot View Storage Directory

Every snapshot view has a *view storage directory* in addition to the directory tree of source files that it loads from VOBs. ClearCase uses the snapshot view storage directory to keep track of such information as which files are loaded into your view and which versions are checked out to it. The view storage directory is for ClearCase administrative purposes only. Do not modify anything in it.

For every 1,000 elements loaded into the view, ClearCase uses about 400 KB of disk space for the view storage directory.

Locations for Snapshot View Storage Directories

Usually, your ClearCase administrator sets up a storage location, which is a directory on a ClearCase server host on UNIX or Windows, and by default ClearCase locates snapshot view storage directories in the storage location. If your ClearCase administrator sets up more than one storage location, ClearCase selects one of these locations as the default when you create a view.

If your ClearCase host is set up to store view storage directories (which happens when you install ClearCase), you can override the default and locate the view storage directory under the root directory of the snapshot view. If you choose to locate the view storage directory under the root directory of the view, be aware of the following recommendations:

- Do not choose this configuration if you use the view when disconnected from the network. You can corrupt the data in the view storage directory if you disconnect it from the network while the view's **view_server** process is running.
- Make sure the view storage directory is accessible to any data backup schemes your organization institutes.

If your ClearCase administrator does not set up storage locations, ClearCase locates the view storage directory under the root directory of the snapshot view.

NOTE: If you plan to work while disconnected from the network, or if your ClearCase host is not set up to store view storage directories, your administrator must set up storage locations.

To Override the Default Location for a Snapshot View Storage Directory

1. While creating a view, on the step of the wizard that asks you to choose a location for a view, click **Advanced Options**.
2. In the **Advanced Options** dialog box, do one of the following:
   - Select **Use Server Storage Location**. If your administrator created more than one location, ClearCase selects one for you. You can choose a different one if you prefer.
   - If your computer is set up to store view storage directories and you want to locate the view storage directory in the root directory of the snapshot view, select **Co-locate Under Snapshot View Root**. Do not select this option if you plan to use the view while disconnected from the network.

Drive Letters for Dynamic Views

If your makefiles or other files require absolute pathnames, assign your view to a drive letter. When you use a wizard to create a view, ClearCase prompts you to assign the dynamic view to a drive letter. After creating a view, if you want to change a drive-letter assignment or assign a drive letter to a team member's view, you can create or modify the assignment while adding a view shortcut to ClearCase Explorer.

In addition, you can create or modify drive-letter assignments from Windows Explorer. Any changes you make to a view's drive-letter assignments outside ClearCase Explorer will invalidate the view's shortcut in ClearCase Explorer.

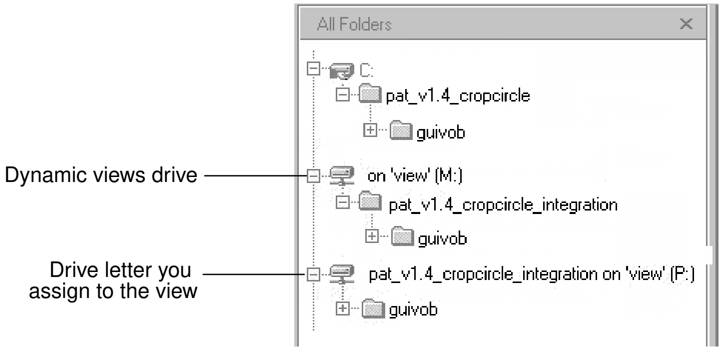
Pathname Differences

You can access any dynamic view that is started on your computer from the *dynamic-views-drive* (**M:** by default) in Windows Explorer. However, when you access a view from the dynamic-views drive, its pathname includes one more component than when you access a view from an assigned drive letter (Figure 15).

For example, both of the following pathnames are available to an element in the **pat_v1.4_cropcircle dynamic** view that is assigned to the **P:** drive:

```
M:\pat_v1.4_cropcircle\guivob\foo.c
P:\guivob\foo.c
```

Figure 15 Dynamic Views Started on a Host



Under the Hood: A Dynamic View Storage Directory

The first time you create a dynamic view, the wizard prompts you to provide a pathname for the view storage directory. The location you choose becomes the default for other dynamic views that you create. Every view has a view storage directory. For dynamic views, ClearCase uses this directory to keep track of which versions are checked out to your view and to store view-private objects. The view storage directory is for ClearCase administrative purposes only. Do not modify anything in it.

The size of the view storage directory depends on the following factors:

- Whether you use the **clearmake** or **omake** *build auditing* and *build avoidance* features
- The size and number of view-private files

For more information, refer to the **clearmake**, **omake,** and **view** reference pages in *ClearCase Reference Manual*.

Valid Locations for Dynamic View Storage Directories

If your computer is set up to store view storage directories, ClearCase reduces communication over the network by locating the view storage directory on your computer.

Consider the following restrictions when choosing a dynamic view storage directory location:

- The directory must be a subdirectory of a shared network resource on a ClearCase host on Windows NT. View processes (specifically, **view_server** processes) run on the machine that physically stores the view storage directory, and only ClearCase hosts on Windows NT can run view processes. The pathname for the directory must not use a Windows special share name. Special share names usually include the dollar sign ($), such as the driveletter$ share name that allows an administrator to gain access to a drive over the network. For example, **\\bread\c$\view\pat_1.4_cropcircle.vws** is not a valid pathname.

- To maintain data integrity, the view storage directory must remain connected to the network. For example, do not locate the view storage directory on a removable storage device.

- If you locate the view storage directory on a laptop and then disconnect the laptop from the network, all of the following restrictions apply:

  - You cannot use the dynamic view.

  - Team members who try to start your view from their hosts will receive error messages from ClearCase.

  - Any **clearmake** or **omake** process that attempts to wink in a derived object from your view will spend some amount of time trying to contact your view. If it cannot contact your view, it will not consider derived objects in your view as *winkin* candidates for 60 minutes. (You can change the amount of time by setting the **CCASE_DNVW_RETRY** environmental variable.) For more information, see the **clearmake** reference page.

- If your ClearCase administrator sets up storage locations (which are directories on ClearCase server hosts), you can locate your dynamic view storage directory in a storage location. However, for best performance, we recommend that you locate dynamic view storage directories on your local host.

We recommend that you make the view storage directory accessible to any data backup schemes your organization institutes.

To Choose a Location for a Dynamic View Storage Directory

1. On the step of the wizard that asks you to choose a location for a view, click **Advanced Options**.

2. In the **Advanced Options** dialog box, do one of the following:

   ➢ Select **Use Explicit Path** and provide a UNC pathname to a shared directory on a ClearCase host on Windows NT. For best performance, we recommend this option.

   The first time you create a view with the wizard, ClearCase presents a template: `\\`*current-hostname*`\<Share>\`*view-tag*`.vws`

   To locate the view storage directory on your computer (recommended), replace `<Share>` with the name of a shared directory on your computer. For example, **\\bread\view_storage\pat_v1.4_cropcircle.vws**

   ➢ Select **Use Server Storage Location**. If your administrator created more than one location, ClearCase selects one for you. You can choose a different one if you prefer.

Under the Hood: Activating and Deactivating VOBs

Dynamic views provide transparent access only to the VOBs that are active on your computer. ClearCase activates the VOBs associated with your project when you finish the Join Project Wizard, each time you log on, and when you start ClearCase Explorer.

You can deactivate VOBs you don't use frequently:

1. In the system taskbar, click **Start>Programs**.

2. Click **ClearCase Administration>Unmount VOB**.

To reactivate the VOB, click **Mount VOB** on the **ClearCase Administration** menu.

2.5 Choosing Components to Load into Snapshot Views

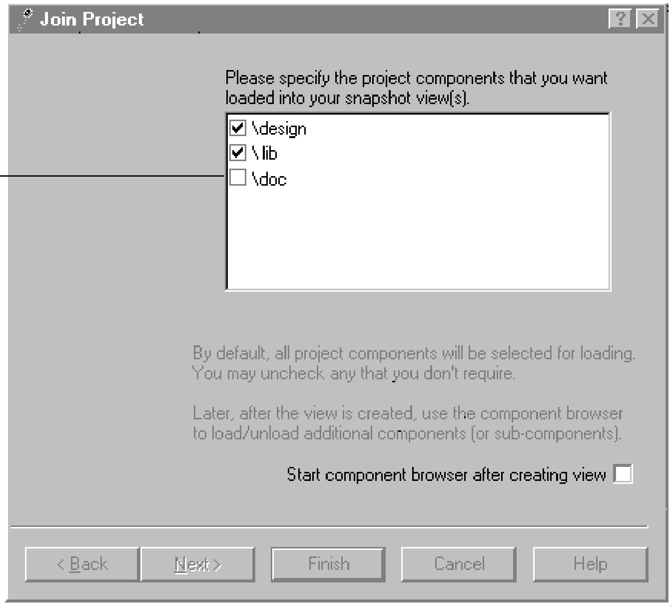
If you choose to create any snapshot views, the Join Project Wizard prompts you to choose the components to load.

Refining the List of Source Files

To save disk space, to reduce the time needed for the initial loading operation, and to reduce time needed for rebase operations, clear the check box for any components that you do not need to complete your work (Figure 16).

Figure 16 Clear Check Boxes for Unused Components

Clear the check box for any component you do not want loaded into your snapshot view



To further refine the list of files and directories that are loaded into your view, select **Start component browser after creating the view**. With the Component Browser, you can specify individual directory and file elements to load.

UNIX VOBs and Case-Sensitivity

If you're selecting elements from VOBs located on a UNIX host, you may encounter problems due to case-sensitivity. Because native file systems for UNIX are case-sensitive, it is possible to create two elements from a UNIX host whose names differ only in capitalization. For example, in a UNIX VOB, these two elements are distinct:

```
/design/func_specs/bas
/design/func_specs/Bas
```

However, Windows does not support case-sensitive file lookups and does not distinguish the two elements in the previous example. If you were to load these two elements, only one of them would have the correct data when copied into the view; duplicated files are reported as *hijacked*. (Hijacked files are discussed throughout Appendix A, *Working in a Snapshot View While Disconnected from the Network*.)

Administering ClearCase discusses case-sensitivity issues in depth. We suggest that you not use mixed-case names for elements that you store in VOBs.

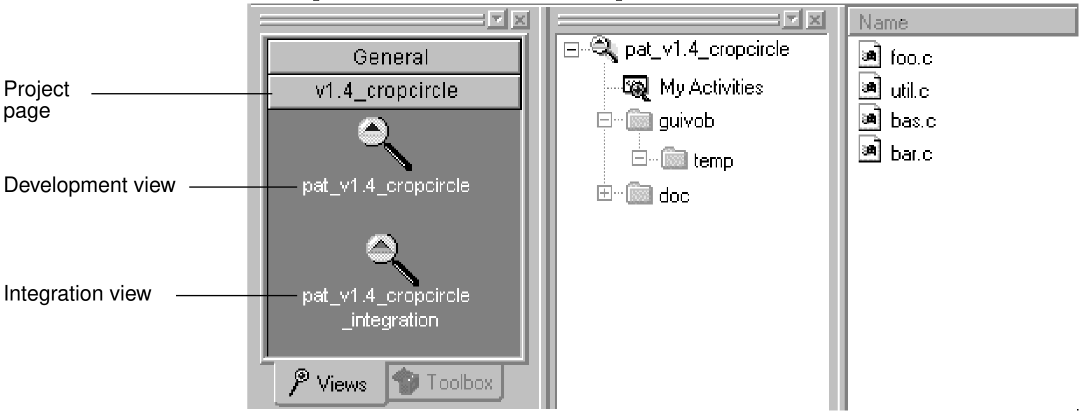
Loading Elements

After you choose components to load into the view, the Join Project Wizard displays your choices. When you confirm your choices and finish the wizard, ClearCase loads the views with the elements you selected. As ClearCase loads files and directories into each view, it shows a progress indicator in the **Snapshot View Update** dialog box. The complete list of files copied into your view appears in the Snapshot View Update window.

2.6 Accessing Your Development View

The **Views** tab in ClearCase Explorer provides access to the views that you create with the Join Project Wizard or View Creation Wizard (Figure 17).

Figure 17   Development View in ClearCase Explorer



ClearCase Explorer creates a page for each project that you join and places view shortcuts on the associated project's page. By convention, your development view is named *username_project-name*, and your integration view is named *username_project-name*_**integration**.

The ClearCase Explorer title bar shows the location of the view in your computer's namespace. Any command you issue from ClearCase Explorer that requires a pathname uses the pathname displayed in the title bar. To assign a different pathname to a view (for example, to change the drive letter that is assigned to a view), modify the view's shortcut.

To access other views from ClearCase Explorer (for example, to access a team member's view), add a view shortcut to the **Views** tab. For information on creating or modifying shortcuts in ClearCase Explorer, refer to ClearCase Explorer online help.

To access your development view outside ClearCase Explorer, refer to the following sections.

Accessing a Snapshot View from Windows Explorer

A snapshot view is a directory tree in a standard file system (plus some hidden, administrative files). You can access it through Windows Explorer as you would access any other directory tree in a file system. For information on assigning a snapshot view to a drive letter, see *Assigning Snapshot Views to Drive Letters* on page 108.

Accessing Someone Else's Snapshot View from Windows Explorer

You can access someone else's snapshot view as you would access any other directory on another computer. Assuming that the directory's owner has shared the directory and set up the proper permissions, use Network Neighborhood to access the view.

If you want to perform ClearCase operations in the view without adding a shortcut to the view in ClearCase Explorer, see *Registering a Snapshot View* on page 112. If you want to assign the view to a drive letter, see *Assigning Snapshot Views to Drive Letters* on page 108.

Accessing Dynamic Views from Windows Explorer

You can access any view that you have started on your computer from the *dynamic-views-drive* (**M:** by default) in Windows Explorer. If you assign the view to a drive letter, you can also access it from the drive letter in Windows Explorer.

For information on assigning dynamic views to drive letters, see ClearCase online help.

Starting Dynamic Views and Activating VOBs

To access dynamic views from Windows Explorer, you must start the view. In a UCM project, ClearCase activates all VOBs that contain components for any of the projects you have joined. For more information, see *Activating or Deactivating VOBs for Dynamic Views* on page 107. If you want to access files in other VOBs, you must mount VOBs. For more information, see ClearCase online help.

Starting the view initiates a **view_server** process, which maps the data in the VOBs that are activated on your computer into a directory tree. VOBs that you activate appear as subdirectories of a view. Then you browse the VOB as you would any other directory.

If you assign a view to a drive letter, ClearCase starts the view when you log on to your computer. If you do not assign a view to a drive letter but do create a ClearCase Explorer shortcut, ClearCase starts the view when you click the shortcut. Otherwise, you must start the view.

To start a dynamic view that was created from a UNIX host, you must first use the Region Synchronizer to import the view's view-tag into your Windows network region.

If you are unable to start a dynamic view that is on another host, check with your administrator to make sure that you can access the view's view storage directory. For more information, see *Administering ClearCase*.

2.7 Connecting to the UCM-ClearQuest Integration

If your project uses the UCM-ClearQuest integration, complete the following tasks to set up your computer for the integration:

- Connecting to a UCM-enabled schema repository. (Your ClearQuest administrator may have done this for you.)
- Logging in to a ClearQuest user database.

To determine whether your project is enabled for ClearQuest:

1. In the ClearCase Explorer Shortcut pane, click **Toolbox**. Then click **UCM>Project Explorer**.
2. In Project Explorer, select your project and click **File>Properties**.
3. In the project's **Properties** dialog box, click the **ClearQuest** tab. If the **Project is ClearQuest-enabled** check box is selected, the project is enabled for ClearQuest.

Connecting to a UCM-enabled Schema Repository

A schema repository is a master database that contains data associated with ClearQuest schemas. To use the UCM-ClearQuest integration, you must connect ClearQuest to a schema repository that contains UCM-specific data. (For more information, see *The UCM-ClearQuest Schema* on page 18.)

Your ClearQuest administrator can create a database profile that simplifies the process of connecting. If such a profile exists, you must know its name and location. If there is no profile, you must know the properties of the database designated as the schema repository (location, name, and vendor type, as well as password information, and so on).

To Connect to a Schema Repository

1. On your Windows desktop, click **Start>Programs>Rational ClearQuest>Rational ClearQuest Maintenance Tool**.

2. In ClearQuest Maintenance Tool, click **Connect to Existing Schema Repository**. Then click **Next**.

3. Do **one** of the following:

   ➢ If your ClearQuest administrator created a database profile, click **Load Profile**. In the **Open** dialog box, select the profile and click **Open**.

   ➢ In the **Vendor** box, select the database vendor for the repository and enter the database properties.

4. Complete any remaining steps in the wizard.

Logging In to a ClearQuest User Database

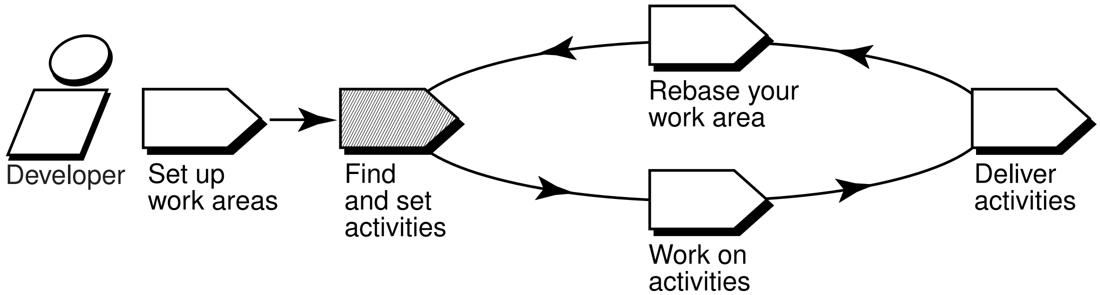
The first time you perform a ClearCase operation that involves the UCM-ClearQuest integration, such as setting your view to an activity, ClearQuest prompts you to enter your ClearQuest user ID and password.

ClearQuest keeps its own database of user IDs and passwords. (ClearCase uses your standard network user ID and password.) Make sure your ClearQuest administrator has added a user ID and password to the ClearQuest user database for you.

When ClearQuest prompts you, be sure to enter your ClearQuest user ID in this dialog box; ClearQuest stores your login information as part of your Windows user account so it doesn't need to prompt you again during ClearCase operations.

Finding and Setting Activities

3



Before you start work on the project, find any *activities* that your project manager or other team members have already assigned to you; if necessary, add new ones to indicate the scope and amount of work scheduled for the project. This chapter describes the following tasks:

- Finding activities
- Creating activities
- Assigning activities in a project that is enabled for ClearQuest
- Setting activities

NOTE: Both ClearCase and ClearQuest provide a Web interface through which you can complete some of the tasks described in this and subsequent chapters. (You cannot set an activity or issue other ClearCase commands from the ClearQuest Web interface.) Ask your system administrator how to access the Web interfaces. Then use the Web interface's online help for more information.

3.1 Finding Activities

UCM provides several locations from which to find activities:

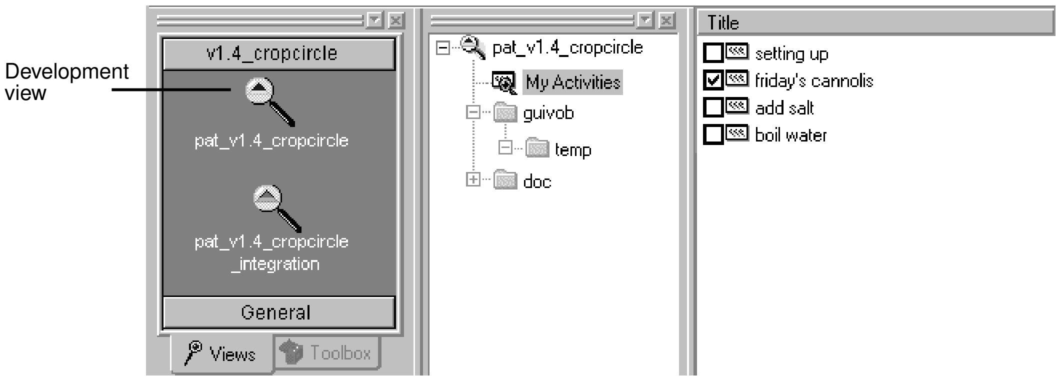
- MyActivities in ClearCase Explorer
- **MyToDoList** and other ClearQuest queries (only for projects that are enabled for ClearQuest)
- Project Explorer
- ClearCase dialog boxes

Finding Activities with MyActivities

MyActivities in ClearCase Explorer shows the activities that are in your work area (Figure 18).

In a project that is enabled for ClearQuest, MyActivities shows the results of the UCMCustomQuery1 query. (For more information, see *UCMCustomQuery1* on page 21.) In a project that does not use the UCM-ClearQuest integration, MyActivities shows the list of activity objects that are in your stream.

Figure 18 MyActivities in ClearCase Explorer



To Open MyActivities

1. In ClearCase Explorer, click the **Views** tab.
2. Click the page for your project. Then select your development view.
3. In the Folder Window, click **MyActivities**.

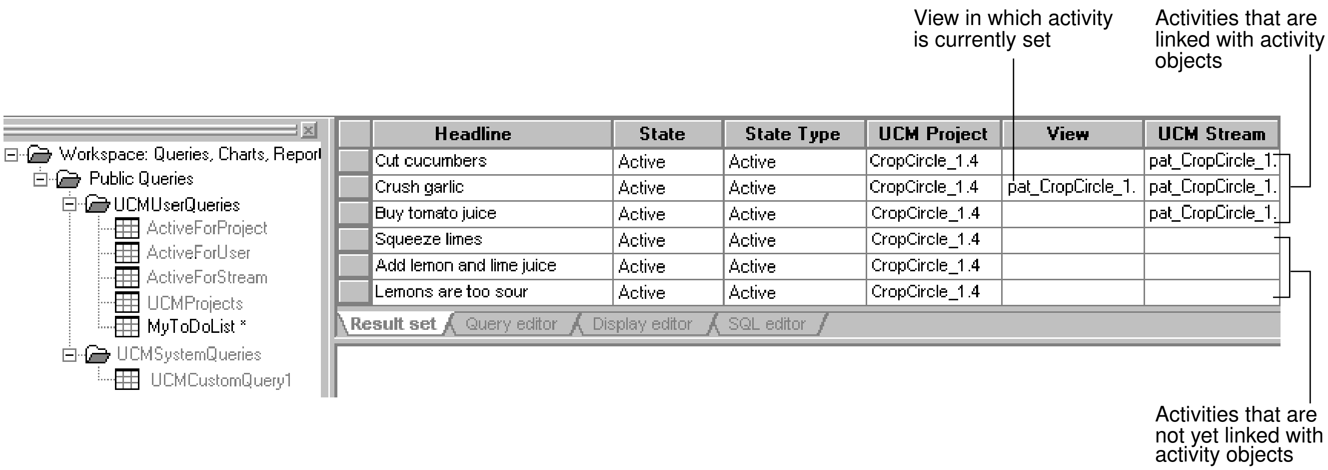
Finding Activities with MyToDoList and Other ClearQuest Queries

If your project is enabled for the UCM-ClearQuest integration, you can use ClearQuest to find activities.

Using MyToDoList

The MyToDoList query finds all UCM-enabled records that are assigned to you in ClearQuest, even if they have not been linked with activity objects or if they have been linked with activity objects in someone else's stream (Figure 19). For more information, see *Queries* on page 19 and *UCM-Enabled Record Types and ClearCase Activities* on page 18.

Figure 19 Linked Activities in ClearQuest MyToDoList



To Open MyToDoList

1. On the Windows desktop, click **Start>Programs>Rational ClearQuest>ClearQuest**.

2. In ClearQuest, in the left pane, click **Workspace>Public Queries>UCMUserQueries**. Then run the **MyToDoList** query.

   ClearQuest displays the query results in the Query results pane (Figure 19).

3. To arrange the list by information displayed in a column, right-click the column heading and select a sort order from the shortcut menu. For example, to arrange activities alphabetically,

right-click **Headline** (the first column's heading) and on the shortcut menu, click **Sort Ascending**.

4. To see more information about an item on your to do list, select it. ClearQuest displays the record's details in the Record form pane.

Creating Your Own ClearQuest Query

If none of the existing queries suits your needs, you can create a new one. When creating a query to find UCM-enabled records, note the following:

➤ To search across all UCM-enabled record types, in the **Choose a Record** dialog box, select **All_Ucm_Activities**. This selection limits your query to fields common to all UCM-enabled record types.

➤ You cannot use a ClearQuest query to find ClearCase change-set information.

For more information on creating a query, refer to the ClearQuest online help.

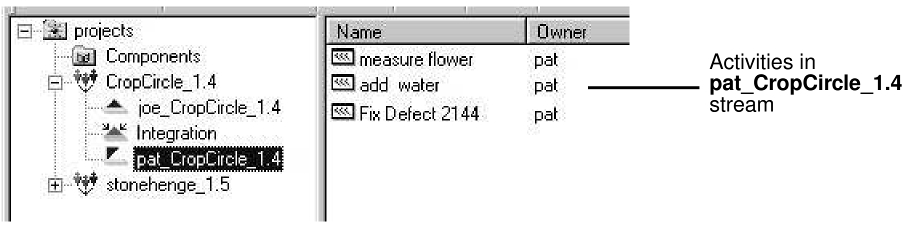
Finding Activities with Project Explorer

The Project Explorer arranges activities hierarchically by stream and by project. It shows only the ClearCase activity objects that are in the currently selected stream.

In a project that is enabled for ClearQuest, each activity in a stream is linked to a UCM-enabled record in ClearQuest (Figure 20). For more information, see *UCM-Enabled Record Types and ClearCase Activities* on page 18.

To start the Project Explorer, in the ClearCase Explorer Shortcut pane, click **Toolbox**. Then click **UCM>Project Explorer**.

Figure 20 Activities in Project Explorer



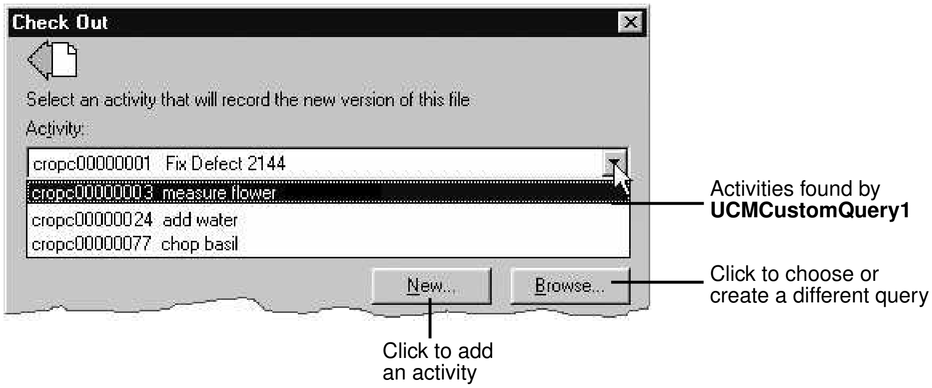
Finding Activities with ClearCase Dialog Boxes

During the check in, check out, and add to source control operations, ClearCase opens a dialog box that prompts you to enter a version creation comment and to set an activity.

In a project that is enabled for ClearQuest, the dialog boxes display the names of activities found by the UCMCustomQuery1 query. (For more information, see *UCMCustomQuery1* on page 21.) You can click **Browse** to use or create a different query (Figure 21).

In projects that do not use the UCM-ClearQuest integration, the dialog boxes show the names of activity objects in the stream that is associated with the view from which you issued the command. You can click **Browse** to see the activities that other team members have created in your stream.

Figure 21 Activities in the Check Out Dialog Box (Project That Is Enabled for ClearQuest)



3.2 Creating Activities

Your organization may have policies regarding the scope of an activity. For example, these policies may require that all activities refer to a change request in your change request management system. The process for creating activities differs, depending on whether your project uses the UCM-ClearQuest integration.

This section describes the following tasks:

- Creating activities in a project that is enabled for ClearQuest
- Creating activities in a project that is not enabled for ClearQuest

Creating Activities in a Project That Is Enabled for ClearQuest

In a project that is enabled for ClearQuest, *activity* usually refers to two objects that are linked together: a ClearCase activity object and a UCM-enabled record in a ClearQuest database. The process of creating an activity in a project that is enabled for ClearQuest entails the creation of two objects in a specific order:

1. You create ClearQuest records (based on a UCM-enabled record type). You can create ClearQuest records from ClearCase Explorer, ClearQuest, or ClearCase dialog boxes.

2. When you set a view to an activity (which you can do by clicking **Actions>WorkOn** for a ClearQuest record), ClearCase creates an activity object in the stream that is associated with the view and links the activity object to the UCM-enabled record in ClearQuest. You do not create ClearCase activity objects directly.

To Create Activities from ClearQuest

NOTE: You can create and use a template of default values for the ClearQuest Submit form. This can save time and be helpful when you are entering multiple, related records. For information on using the template, refer to ClearQuest online help.

1. On the Windows desktop, click **Start>Programs>Rational ClearQuest>ClearQuest**.

2. In ClearQuest, click **Actions>New**.

3. In the **Choose a Record Type** dialog box, select an appropriate UCM-enabled record type. (Your project leader sets up the record types that are in your ClearQuest database.) Then click **OK**.

   ClearQuest highlights in red the fields that are required and the fields that contain invalid values. If a field contains an invalid value, you can right-click the field and then click **Error Message** to see any message your project manager may have included. Your project manager can include explanations for each field. If you have questions about what type of information a field requires, right-click the field and then click **Help** on the shortcut menu.

4. Complete the required fields in the ClearQuest Submit form.

5. In the **UCM Project** list, select your project. (Your project manager determines the location for this list when she creates the corresponding record type.) Then click **OK**.

After you create an activity from ClearQuest, you must complete an additional procedure to make the activity appear in MyToDoList or MyActivities. (See *To Make Activities Appear in MyActivities and MyToDoList* on page 49.)

To Create Activities in a ClearQuest-Enabled Project from ClearCase Dialog Boxes

This procedure accomplishes three tasks:

➤ Creates a UCM-enabled record in the ClearQuest database and assigns the activity to you.

➤ Creates an activity object in the stream that is associated with the view and links the activity object to the UCM-enabled record in ClearQuest.

➤ Sets your view to the activity.

To create activities:

1. In the **Check Out**, **Check In**, or **Add to Source Control** dialog box (which opens as part of the corresponding operation), click **New**.

2. From the **New** menu, select a record type. (All record types on the **New** menu are UCM-enabled.)

3. In the **Submit** dialog box, complete the required fields, which are highlighted in red.

4. Click **OK** to create a UCM-enabled record and assign the activity to yourself.

5. To add this activity to your stream and set your view to the activity, select it in the **Check Out**, **Check In**, or **Add to Source Control** dialog box and click **OK**.

Creating Activities in a Project That Is Not ClearQuest-Enabled

In UCM projects that do not use the UCM-ClearQuest integration, you create activity objects directly in a specific stream.When you add an activity, it remains in your stream unless you use **rmactivity** to remove it.

Non-ClearQuest-Enabled Project: To Create Activities from ClearCase Dialog Boxes

1. In the **Check Out**, **Check In**, or **Add to Source Control** dialog box (which open as part of the corresponding operation), click **New**.
2. In the **New Activity** dialog box, enter the name of the activity.
3. Click **OK**.

When you add an activity from the **Check Out**, **Check In**, or **Add to Source Control** dialog box, ClearCase sets your view to the activity.

Non-ClearQuest-Enabled Project: To Add Activities from the View Properties Dialog Box

1. Open a view's properties dialog box. One way to display view properties is as follows:
   a. In ClearCase Explorer, right-click the root directory of the view, and from the shortcut menu choose **Properties of View**.
   b. In the **Edit View Properties** dialog box, select a view and click **OK**.
2. On the **General** tab of the Properties dialog box, click **Change**.
3. In the **Select Activity** dialog box, click **New**.
4. In the **New Activity** dialog box, enter an activity name and click **OK**.

3.3 Assigning Activities in a Project That Is Enabled for ClearQuest

The concept of assigning an activity applies only to UCM-enabled records. In general, *assigning* an activity means making a person responsible for some portion of an activity's development. An activity assignment may change several times during a development cycle. For example, a project manager may take initial responsibility for an activity. When she sets the schedule for a given project, she assigns the activity to a developer on the project. After a developer begins work on an activity, the activity assignment usually doesn't change.

ClearQuest records include fields to keep track of an activity's current owner. Your organization may restrict who can change the value in the owner field.

NOTE: ClearCase activity objects exist in a given stream, and they record the activity creator, but these data points cannot be changed and are not intended to express the concept of assignment.

To Make Activities Appear in MyActivities and MyToDoList

By default, the MyToDoList query and MyActivities find records that are both in a Ready or Active state type and assigned to you. To make an activity appear in MyActivities or MyToDoList, you must change its state type and owner.

1. In ClearQuest, run a query to find an activity. Then click the activity in the Query editor pane.
2. In the Record form pane, click **Actions** and choose a command that moves the activity to a state based on the Ready state type.
3. In the **Owner** list, select your user ID.
4. To see the activity in MyToDoList, in the left pane, click **Workspace>Public Queries >UCMUserQueries**. Then run the MyToDoList query.

   To see the activity in MyActivities, start ClearCase Explorer and select your view from the Shortcut pane. Then click MyActivities in the Folder pane.

3.4 Setting Activities

To start working on an activity, you must set your development view to the activity.ClearCase assigns all work in your view to the currently set activity, adding each version you create in your view to the activity's change set.

You can set activities from the following locations:

- ClearCase Explorer
- MyToDoList or the results of any other ClearQuest query (only for projects that are enabled for ClearQuest)
- Dialog boxes that open during checkout, checkin, and add to source control operations

This section also describes the following tasks:

- Unsetting an activity
- Seeing which activity is currently set in the view

Setting Activities from ClearCase Explorer

1. In ClearCase Explorer, click the **Views** tab.
2. Click the page for your project. Then select your development view.
3. In the Folder Window, click **MyActivities**.
4. In the Details pane, select the check box next to the activity you want to set.

Setting Activities from ClearQuest

1. In ClearQuest, do **one** of the following:
   - Display MyToDoList by clicking **Start>Programs>Rational ClearQuest>ClearQuest**. Then in the left pane, click **Workspace>Public Queries>UCMUserQueries** and run the MyToDoList query.
   - Use another query to find the activity you want to set.
2. In the Query results pane, click the activity you want to set.
3. In the Record form pane, if it isn't already selected, select your project in the **UCM Project** list. (Your project manager determines the location for this list when she creates the corresponding record type.)
4. Click **Actions>WorkOn**.

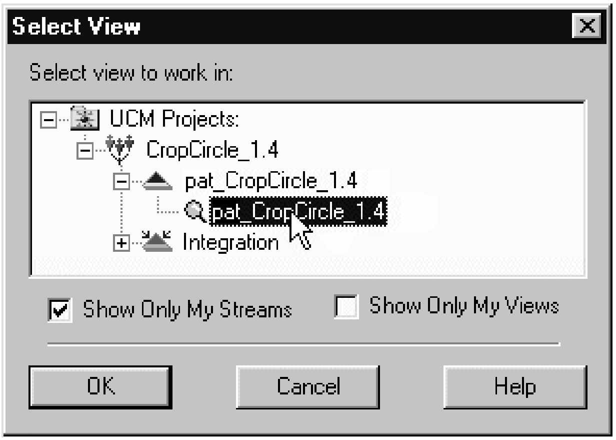
5. In the **Select View** dialog box, select **Show Only My Streams**.

6. Click the plus sign (+) next to your stream; then select your development view and click **OK**.

The status bar in ClearQuest indicates that your view is being set to the activity.

To start working on your project's source files, navigate to your view and check out the source files.

Figure 22 Select Your View



Setting Activities from ClearCase Dialog Boxes

During the checkin, checkout, and add to source control operations, ClearCase opens a dialog box that prompts you to enter a version creation comment and to set an activity.

In a project that is enabled for ClearQuest, the dialog boxes list the names of activities found by the UCMCustomQuery1 query. (For more information, see *UCMCustomQuery1* on page 21.) You can click **Browse** to use or create a different query (Figure 21).

In projects that do not use the UCM-ClearQuest integration, the dialog boxes show the names of activity objects in the stream that is associated with the view from which you issued the command. You can click **Browse** to see the activities that other team members have created in your stream.

Unsetting Your View from an Activity

You can change any view to have no currently set activities, for example, if you need to remove the view or remove the activity currently set in the view.

To Unset Activities from ClearCase Explorer

1. In ClearCase Explorer, display your work area and click **MyActivities**.
2. In the Details pane, clear the check box next to the currently set activity.

To Unset Activities from a Command Prompt

1. At a command prompt, change to any directory in your view.
2. Enter the following command

   cleartool setactivity –none

To See Which Activity Is Currently Set

From ClearCase Explorer

In ClearCase Explorer, display your work area and click **MyActivities**.

The Details pane places a checkmark next to the currently set activity.

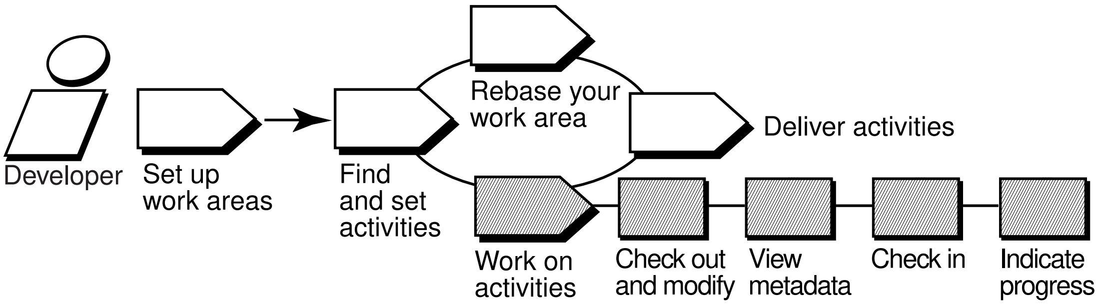
From a Command Prompt

1. At a command prompt, change to any directory in your view.
2. Enter the following command:

   cleartool setactivity –none

Working on Activities

4



Working on activities involves the following tasks:

- Checking out files
- Working with checkouts
- Canceling checkouts
- Checking in files
- Testing your work
- Indicating your progress

The work in your development work area is unavailable to the project team until you deliver it.

4.1 Checking Out Files

To modify files under ClearCase control, you must check them out and make sure that you have set your view to an *activity*. Checking out does not add a version to the activity's change set. (Checking in creates a new version, which ClearCase adds to the activity's change set.)

NOTE: For information on accessing your development view, see *Accessing Your Development View* on page 36.

To Check Out Files and Set Activities

1. In ClearCase Explorer, navigate to the directory in your development view that contains the files you want to check out. Then select the files.

2. Right-click one of the selected files. On the shortcut menu, click **Check Out**.

   ClearCase opens the **Check Out** dialog box with the currently set activity selected in the Activity list.

3. To set a different activity (or, if you have not set the view to an activity), do the following:

   c. Select one from the **Activity** list. To filter the list, click **Browse**.

   d. If an appropriate activity doesn't exist, create and set a new one by clicking **New**. If your project is enabled for ClearQuest, select a record-type from the shortcut menu that appears when you click **New**. ClearQuest changes the activity state to an **Active** state type, and ClearCase adds the version you check out to the activity's change set.

4. Describe the modifications you plan to make in the **Checkout Comment** box. Use comments to describe modifications to individual versions within an activity's change set. Your comments are displayed as part of the element's history and the version's properties.

5. Click **OK**.

For information on checking out VOB links in a snapshot view, see *Loading Elements* on page 36.

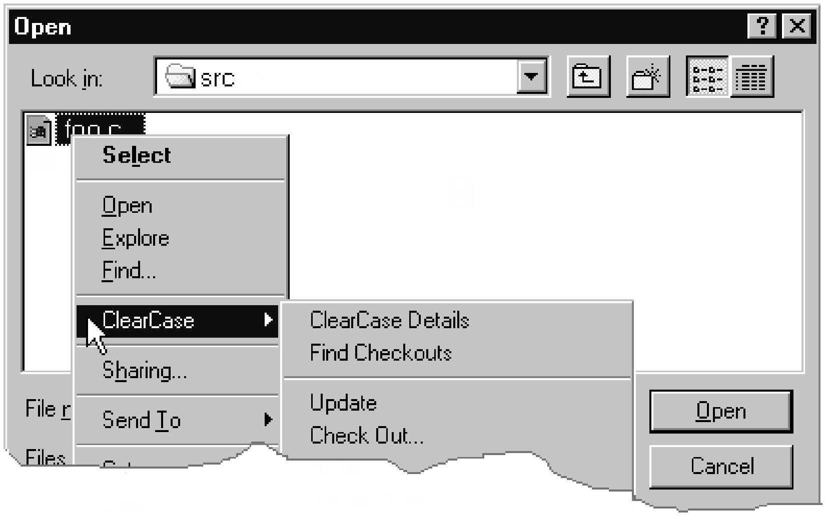
Using the Open Dialog Box

The **Open** dialog box that many applications use is actually part of ClearCase Explorer. As illustrated in Figure 23, right-clicking a loaded version of a ClearCase element in the **Open** dialog box displays the ClearCase shortcut menu.

To check out a file from an **Open** dialog box:

1. In an application such as Microsoft Notepad, click **File>Open**.
2. In the **Open** dialog box, access the file or directory that is under source control.
3. Right-click the file; then click **ClearCase>Check Out**.

Figure 23 The ClearCase Shortcut Menu from the Open Dialog Box



Checking Out Directories

Directories, as well as files, are under ClearCase source control, yet you rarely need to check out a directory explicitly. ClearCase checks out and checks in a file's parent directory when you add the file to source control.

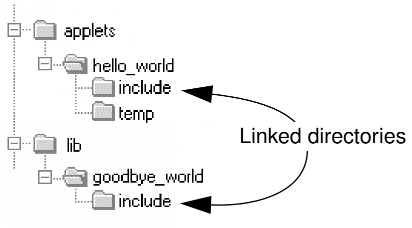
What does it mean for a directory to be under source control? In a version-controlled directory, ClearCase creates a new version of the directory when you add or rename a file element under source control. Having versions of directories can be helpful if, for example, you rename a source file used in a particular release and then modify your makefile to reflect this change. If you need to rebuild a previous release, you can set up your view to select the version of the directory that contains the file under its previous name.

NOTE: When you issue commands from a command-line interface (CLI), such as an MS-DOS command prompt, ClearCase does not check out directories automatically. When using a CLI to change a directory element, you need to check out the directory explicitly. For more information on checking out files and directories from a CLI, see the **checkout** reference page in *ClearCase Reference Manual*.

Under the Hood: VOB Links

A VOB link makes a file element or directory element accessible from more than one location in the VOB namespace. There are two kinds of VOB links: *symbolic links*, which are available for file and directory elements, and *hard links*, which are available for file elements only. We recommend that you use VOB symbolic links instead of VOB hard links whenever possible. In Figure 24, the directory element **include** is linked.

Figure 24 VOB Link



You use the **cleartool ln** command to create VOB links. See the **ln** reference page in *ClearCase Reference Manual* for more information.

Symbolic Links and Hard Links in Dynamic Views

In *dynamic views* (which use the MVFS, or multiversion file system), VOB links behave similarly to symbolic links or hard links in a UNIX file system: symbolic links point to a file or directory element in a different location, and hard links are alternate names for a single file element.

You cannot check out a VOB symbolic link; you must check out the symbolic link target.

When you check out a hard-linked element from a given pathname, ClearCase considers other pathnames for the element as “checked out but removed.” That is, to prevent you from modifying the element from multiple pathnames, ClearCase executes standard checkout behavior at only one pathname (the one from which you entered the checkout command), but does not create view-private files at other pathnames. For information about standard checkout behavior, see the **checkout** reference page in *ClearCase Reference Manual*.

Symbolic Links in Snapshot Views

Snapshot views created from a Windows host do not support links. ClearCase approximates VOB symbolic link behavior in the following ways:

- If a *load rule* selects a symbolic link, ClearCase copies the link target into the view at the link's pathname.

  We recommend that you do not load a linked directory more than once in your view unless it is necessary for build purposes. Although ClearCase keeps track of multiply loaded elements accurately, you may become confused and modify the wrong copy of a file, losing information upon checkin. For more information, see *Caution: Losing Data Because of VOB Hard Links*.

- You cannot check out a file element from a symbolic link pathname; you must check out the link target. In ClearCase Explorer, the shortcut menu for a symbolic link includes the **Explore Link Target** command, which displays the link target's parent directory. Use this command to find and check out the link target.When you check in the modified link target, ClearCase updates all associated symbolic links loaded in your view. Unlike directory link targets, you can check out a file link target only if it is loaded in your view.

- The **Add to Source Control** command (available from ClearCase GUIs) checks out, modifies, and checks in the link target directory whether you issue the command from a directory symbolic link or from the link target (and whether or not the target directory is loaded into the view).

  If you issue the **cleartool checkout** command from a symbolic link directory, you must use the following syntax:

  cleartool checkout .

  A **checkout** command issued for the current directory checks out the link target whether or not the target directory is loaded into the view.

  However, if you use the **cleartool checkout** *dirname* form of the command to check out a different directory, *dirname* must be a link target.

  When either you or ClearCase checks in a link target directory, ClearCase updates the associated symbolic links that are loaded in your view.

- If you *hijack* a file symbolic link, the Update Tool detects it. However, you cannot check out the hijacked symbolic link. To add your hijacked changes to the VOB, you must check out and modify the link target.

Hard Links in Snapshot Views

Each time a load rule selects a hard link, ClearCase loads the element into the view as a standard file.

Caution: Losing Data Because of VOB Hard Links

If you load multiple instances of a hard-linked element into a snapshot view, you must be careful to check out, modify, and check in only one instance of the file.When you check in a hard-linked file (or a file below a symbolic-linked directory), ClearCase updates all other instances in your view, which could result in loss of data if you modified multiple copies of the same file. (Note that, when updating instances of files because of a checkin, ClearCase renames any *hijacked* file to *filename*.**keep** before updating it.)

For example, the following sequence of events will lead to lost data:

1. You check out the hard-linked file **foo\util.c**.
2. ClearCase removes the read-only attribute from **util.c** in the **foo** directory only (which is the location from which you issued the **checkout** command).
3. You modify **foo\util.c** but do not check it in.
4. Later, you lose track of which file you checked out. You then remove the read-only attribute and modify **util.c** in the **bar** directory.
5. You check in **bar\util.c**. Even though you checked out and modified **foo\util.c**, ClearCase does not prevent you from checking in **bar\util.c**; with a VOB hard link, **bar\util.c** is just another name for **foo\util.c**.
6. Any changes you made to **foo\util.c** are lost upon checkin because ClearCase updates all copies of duplicated files when you check in an element. Note that ClearCase does not consider any copy of **util.c** to be hijacked (even if you change attributes), because you checked out the element in the VOB.

4.2 Working with Checkouts

After you check out a file, you do not need to interact with ClearCase until you're ready to check it in. However, you may want to use some ClearCase tools for the following tasks:

- Viewing an element's history
- Comparing versions of elements
- Tracking checked-out versions
- Viewing an activity's change set
- Moving versions to a different activity

Viewing an Element's History

The History Browser displays the history of an element's modifications, including version-creation comments (entered when someone checks out or checks in an element).

To View an Element's History

In ClearCase Explorer, right-click an element; on the ClearCase shortcut menu, click **History**.

Comparing Versions of Elements

As you modify source files, you may want to compare versions to answer such questions as these:

- What changes have I made in my checked-out version?

- How does my checked-out version differ from a particular historical version or from the version being used by one of my colleagues?

To Compare with a Predecessor

In ClearCase Explorer, right-click a file. On the ClearCase shortcut menu, click **Compare with Previous Version**.

To Compare with a Version Other Than the Predecessor

1. In ClearCase Explorer, right-click a file and click **Version Tree**.

2. In the Version Tree Browser, right-click a version and click **Compare>with Another Version**.

   The cursor changes to a target icon.

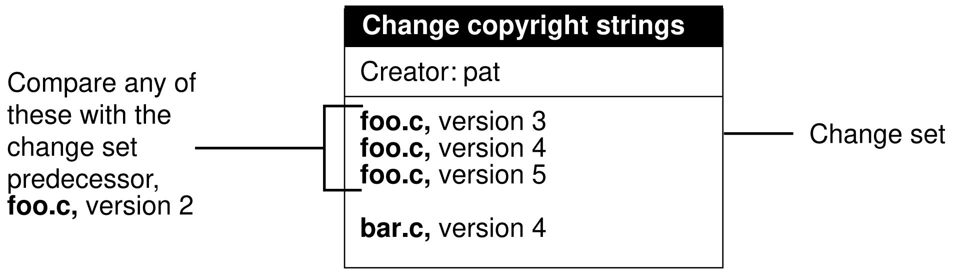
3. Click the version you want to compare.

Comparing with a Change-Set Predecessor

A change set lists all versions you have created while working on a specific activity. In some cases, you create more than one version of an element for a given activity. For example, for the activity **Change copyright strings** you check in a version of **foo.c** and later realize that you missed a string in the file. You then set your view to the activity **Change copyright strings**, check out **foo.c**, modify the string, and check in the file.

To see the changes you have made to an element while working on an activity, you can compare a version in the activity's change set with the version that immediately precedes the change set. In Figure 25, the change set predecessor of **foo.c** is **version 2**. You can compare any version of **foo.c** in the change set with **version 2**.

Figure 25 Compare with Change Set Predecessor



To Compare with a Change-Set Predecessor

1. Display an activity's properties sheet. For example:

   a. In ClearCase Explorer, select your work area and click **MyActivities**.

   b. Right-click the activity and click **Properties**.

2. In the activity's properties sheet, click the **Change Set** tab.

3. Right-click the version you want to compare and, from the shortcut menu, click **Compare with Change Set Predecessor**.

Tracking Checked-Out Versions

Depending on how you work, you may forget exactly how many and which files are checked out. To list all the files and directories you currently have checked out to your view:

1. In the ClearCase Explorer Folder window (middle), right-click a view root and click **Find Checkouts**.

2. In the **Find Criteria** dialog box, edit the path from the root directory of the snapshot view in the **Search folder** box.

3. Click the **Include subfolders** option.

4. Click **OK**.

Viewing an Activity's Change Set

You can see which versions are in an activity's change set from ClearQuest, ClearCase Explorer, or Project Explorer.

To View an Activity's Change Set from ClearQuest

1. Use a query to find the activity. For example, to use MyToDoList:

   a. Click **Start>Programs>Rational ClearQuest>ClearQuest.**

b. In the left pane, click **Workspace>Public Queries>UCMUserQueries** and run the **MyToDoList** query.

   ClearQuest displays MyToDoList list in the Query builder pane. (Figure 19 on page 43.)

2. In the Query builder pane, select a record.

3. In the Record form pane, display the **Change Set** list. Your project manager determines the location of this list when she designs the record's record type.

To View an Activity's Change Set from ClearCase Explorer

1. From MyActivities in ClearCase Explorer, right-click an activity.

2. On the shortcut menu, click **Properties**.

To View an Activity's Change Set from Project Explorer

1. In the ClearCase Explorer Shortcut pane, click **Toolbox**. Then click **UCM>Project Explorer**.

2. In Project Explorer, select a stream and right-click an activity in the stream; on the shortcut menu, click **Properties**.

3. In the **Properties** dialog box, click the **Change Set** tab.

Moving Versions to a Different Activity

You can move versions from a ClearCase properties sheet or from ClearQuest if your project is enabled for the UCM-ClearQuest integration.

To Move Versions with ClearCase

1. Display an activity's change set. (For more information, see *Viewing an Activity's Change Set* on page 61.)

2. On the **Change Set** tab, select one or more elements.

3. Right-click a selected element, and on the shortcut menu, click **Move to Activity**.

To Move Versions with ClearQuest

1. Display an activity's change set in ClearQuest. For more information, see *To View an Activity's Change Set from ClearQuest* on page 61.
2. Select one or more elements from the **Change Set** list.
3. Right-click a selected element, and on the shortcut menu, click **Move to Activity**.
4. Follow the ClearCase prompts.

4.3 Canceling Checkouts

If you check out a file but don't want to check in your changes or want to start with a fresh copy, you can cancel the checkout as follows:

1. In the ClearCase Explorer Details pane, select one or more checkouts. Then right-click.
2. On the shortcut menu, click **Undo checkout**.
3. Click **Yes** in the **Confirm Undo Checkout** dialog box. You can choose to save any of your changes in the file *filename***.keep**.

Under the Hood: Canceling Checkouts

When you cancel the checkout of a file element, ClearCase handles the request as follows:

1. It prompts you to rename the file in your view to *filename***.keep**.
2. It notifies the VOB that you no longer have the version checked out in your view.
3. In a snapshot view, it copies from the VOB the version that was in your view when you performed the checkout operation.

   In a dynamic view, it uses the config spec's version-selection rules to select a version.

Canceling Directory Checkouts

Although you rarely need to check out a directory explicitly, if you do and then cancel the checkout, ClearCase notifies the VOB that you no longer have the version of the directory checked out to your view. ClearCase does not prompt you to rename a canceled directory checkout to *directory-name***.keep**.

If you cancel a directory checkout after changing its contents, any changes you made with **cleartool rmname**, **mv**, and **ln** are lost. Any new elements you created with **mkelem** or **mkdir** become orphaned. (When you create elements from the GUI with the **Add to source control** command, ClearCase checks out the directory, adds the element, and checks in the directory, avoiding the creation of orphaned elements.) ClearCase moves orphaned elements (and any data that exists in the view at the pathname of the new element) to the VOB's **lost+found** directory under names of this form:

element-name.UUID

In such cases, **uncheckout** displays this message:

```
cleartool: Warning: Object "foo.c" no longer referenced.
cleartool: Warning: Moving object to vob lost+found directory as
"foo.c.5f6815a0a2ce11cca54708006906af65".
```

In a snapshot view, ClearCase does not remove *view-private objects* or start the update operation for the directory in the view. To return the directory in your view to its state before you checked it out, you must start the Update Tool. For information on starting the Update Tool, see ClearCase online help.

In a dynamic view, ClearCase does not remove view-private objects, but it does revert the view to its previous state.

To move an element from the **lost+found** directory to another directory within the VOB, use the **cleartool mv** command. To move an element from the **lost+found** directory to another VOB, use the **relocate** command. For more information about moving elements to another VOB, see the **relocate** reference page in *ClearCase Reference Manual*.

To permanently delete an element in the **lost+found** directory, take note of the orphaned element's name and use this command:

cleartool rmelem *VOB-pathname***\lost+found***orphaned-element-name*

For example, from a dynamic view:

> **cleartool rmelem \guivob\lost+found\foo.c.5f6815a0a2ce11cca54708006906af65**

From a snapshot view:

> **cd c:\pat_v1.4_cropcircle_sv**
> **cleartool rmelem guivob\lost+found\foo.c.5f6815a0a2ce11cca54708006906af65**

NOTE: In a snapshot view, ClearCase treats the **lost+found** directory, which is located immediately below the root directory of a VOB, as any other directory. To load the directory in your view, you must use a load rule that specifies either the element's parent directory or the directory itself. However, as with any other directory in a snapshot view, you do not need to load the **lost+found** directory to issue ClearCase commands for elements in the directory.

Canceling the Checkout of a Deleted File

If you check out a file element and then delete that file from your view (by using, for example, the **Delete** command in ClearCase Explorer), use this procedure to cancel the checkout:

1. In ClearCase Explorer, right-click the directory containing the deleted file.

2. On the shortcut menu, click **Find Checkouts**.

3. Complete the **Find Criteria** dialog box.

4. In the Find Checkouts window, select the file whose checkout you want to cancel and click **Undo Checkout**.

4.4 Checking In Files

Check in when you want to keep a record of the file in its current state. When you check in from your development view, ClearCase records the new version in your development stream; even though the new version is a permanent part of the element, the modifications are unavailable to the rest of the development team until you deliver them to the integration stream.

To Check In Files

1. In ClearCase Explorer, select one or more files.
2. Right-click a selected file. On the shortcut menu, click **Check In**.
3. In the **Check In** dialog box, ClearCase displays the comments you entered when you checked out the file. You can reuse these comments or modify them.

Snapshot View: Checking In VOB Links

When you issue a checkin command from a snapshot view, ClearCase handles the request as follows:

1. It copies your modifications to the VOB as a new version.

   The version you check in remains in the view, regardless of the view's config spec.

2. It removes write permission for the file.

For any other instance of the file loaded into a snapshot view, ClearCase copies the new version from the VOB into your view. (If your load rules specify an element that appears in more than one VOB location, the element is copied into each of the appropriate locations in your view's directory tree.)

When you check in a symbolic-linked directory from a snapshot view, ClearCase does not update any other instances of the directory loaded in your view. As you add file elements to source control, ClearCase adds a copy of the element to all instances of a parent directory loaded in your view.

4.5 Testing Your Work

To reduce the amount of merging needed when you deliver your work, you should build and test the modifications in your development stream, especially after you rebase the stream and before you deliver activities. If your organization uses **clearmake** or **omake**, you can use these ClearCase build tools for your test builds; however, the *build auditing* and *build avoidance* features are available only from dynamic views.

For more information about using **clearmake** or **omake**, refer to their reference pages in *ClearCase Reference Manual* and *Building Software with ClearCase*.

4.6 Indicating Your Progress

If your project is enabled for ClearQuest, you can complete the following tasks to indicate your progress on an activity:

- ➤ Modifying information in ClearQuest
- ➤ Closing an activity
- ➤ Using schema-specific actions

You may need to reassign or delete an activity regardless of whether your project is enabled for ClearQuest.

Modifying Information in ClearQuest

Your ClearQuest activities may include fields for keeping additional information. For example, an activity record form may contain an **Attachments** tab to link to supporting documents written in other applications. The form may also contain fields to describe a defect's symptoms. Adding or modifying such information does not change the activity's state or have any effect on the corresponding activity in your stream.

For information on adding information to multiple activities, see ClearQuest online help.

To Modify ClearQuest Information

1. Use a query to find the activity. For example, to use MyToDoList:

   a. Click **Start>Programs>Rational ClearQuest>ClearQuest.**

   b. In the left pane, click **Workspace>Public Queries>UCMUserQueries** and run the MyToDoList query.

      ClearQuest displays MyToDoList in the Query builder pane. (Figure 19 on page 43.)

2. In the Query builder pane, select a record.

3. In the Record form pane, click **Actions>Modify**.

   Your project manager can include help for the record form. To see information about a field, right-click it and then click **Help** on the shortcut menu.

4. In the Record form, make your changes. (Changing the Owner field reassigns the activity; for information on reassigning an activity, see *Reassigning Activities* on page 70.)

5. To save your changes, click **Apply**. To clear changes you made without saving them, click **Revert**.

Closing Activities

Your project manager may have set up a policy for your project that closes activities in ClearQuest when you deliver them. If the policy is not in effect for your project, close your activity when you have completed the associated work (which is usually when you deliver it).

Completed activities do not appear on MyToDoList, but they remain in your stream and appear in your stream's properties even after you deliver them.

To Close a ClearQuest Activity

Your project manager may have customized your database schema to implement various ways to move an activity to a state based on the Complete state type. The following procedure assumes that your database schema includes an action to close activities:

1. Use a query to find the activity. For example, to use MyToDoList:

   a. Click **Start>Programs>Rational ClearQuest>ClearQuest.**

   b. In the left pane, open **Workspace>Public Queries>UCMUserQueries** and run the **MyToDoList** query.

      ClearQuest displays MyToDoList in the Query builder pane. (Figure 19 on page 43.)

2. In the Query builder pane, select a record.

3. In the Record form pane, click **Actions** and select an action that closes activities.

   Your project manager can include help for the record form. To see information about a field, right-click it and then click **Help** on the shortcut menu.

4. Click **Apply**.

Using Schema-Specific Actions

Your ClearQuest database schema, which your project manager can customize (see *The UCM-ClearQuest Schema* on page 18), may support these common change-request management actions:

- Marking activities as duplicate
- Reopening closed activities
- Postponing activities

Your database schema may provide various ways to use these actions. For example, record types that can be postponed may have a **Postpone** command that you can use after clicking **Actions** in the record's Record form pane.

Keep the following UCM-ClearQuest integration concepts in mind while using any schema-specific actions:

- Your project manager can set up each UCM-enabled record type to behave differently; some actions may not be available for all record types.

- You cannot move an activity from one development stream to another. However, you can move versions in an activity's change set to another activity in the same stream (see *To Move Versions with ClearCase* on page 62) or to an activity in a different stream (see *Moving Work from a Development Stream to Another Project* on page 110).

- You cannot delete an activity that is either currently set in a view or that contains versions in its change set. The **UCM View** field in the ClearQuest Query results pane indicates whether an activity is currently set. (Figure 19 on page 43.) The **Change Set** list, which is available from the ClearQuest Record form pane or from the activity's properties sheet, displays the activity's change set.

- We recommend against working in a development stream that contains activities you do not intend to deliver. Keeping undelivered activities in your development stream increases the risk that, after a rebase operation, your development stream does not contain a cohesive set of versions. If you begin working on an activity and then postpone it until the next release, we recommend that you move the work to the new project, and then create and work in a new development stream. (For more information, see *Moving Work from a Development Stream to Another Project* on page 110.)

Reassigning Activities

You can assign an activity on MyToDoList to another project team member by changing the activity's owner, unless your project manager has enabled the UCM policy **Check Before Work On**, which prevents you from assigning your work to someone else.

If the ClearQuest activity has been linked with a corresponding activity in your stream (which occurs when you set your view to the activity), you can change the owner in ClearQuest, but you cannot move the ClearCase activity to the new owner's development stream. The **UCM Stream** field in the ClearQuest Query results pane indicates whether an activity is linked with a stream. (Figure 19 on page 43.)

To reassign an activity that is linked with a stream, change the activity's owner in ClearQuest, and then do one of the following:

➤ If you have not created versions for the activity (that is, the activity's change set is empty), you can remove the activity in your development stream. (See *To Delete a ClearCase Activity* on page 71.) Then, when the new activity owner sets his view to the activity, ClearCase creates a new activity in the new owner's stream. The activity's **Change Set** list, which is available from the ClearQuest Record form pane or from an activity's properties sheet, displays its change set.

➤ If the activity's change set is not empty, the new owner can create and work in a view attached to the stream containing the activity. For information on attaching a view to an existing stream, see ClearCase online help.

To Change an Activity's Owner

1. Use a query to find the activity. For example, to use **MyToDoList**:

   a. Click **Start>Programs>Rational ClearQuest>ClearQuest.**

   b. In the left pane, click **Workspace>Public Queries>UCMUserQueries** and run the **MyToDoList** query.

      ClearQuest displays MyToDoList in the Query builder pane. (Figure 19 on page 43.)

2. In the Query builder pane, select an activity.

3. In the Record form pane, click **Actions>Modify**.

4. In the **Owner** field, select the new owner.

5. Click **Apply**.

Deleting Activities

NOTE: Your project manager determines whether developers can delete ClearQuest records of any given record type.

If a ClearQuest activity is linked with a ClearCase activity (which happens if you set a view to the activity) and if the ClearCase activity contains versions in its change set, you must delete the ClearCase activity first.

The **UCM Stream** field in the ClearQuest Query results pane indicates whether an activity is linked with a stream. (Figure 19 on page 43.) The activity's Change Set list, which is available from the ClearQuest Record form pane or from the activity's properties sheet, displays its change set.

To Delete a ClearCase Activity

NOTE: You cannot delete an activity if the change set contains one or more versions. We recommend that you use some other action for an activity with a nonempty change set. For example, you can deliver all other activities in the stream, abandon the stream, and create a new one to keep track of your work on the project. But if no other action is appropriate, you must first use **cleartool rmversion** to remove the versions from the VOB. Note that removing versions from the VOB is irreversible and may have unintended consequences, for example, if there are dependencies on the versions that you remove.

1. In ClearCase Explorer, from the Shortcut pane, click **Toolbox>UCM>Project Explorer**.

2. In Project Explorer, navigate to your project.

3. Select the stream that contains the activity.

4. In the right pane, select the activity and click **File>Delete**. If the activity is linked with a record in ClearQuest, you must delete the ClearQuest record from the ClearQuest GUI.

To delete an activity from a command prompt, use **cleartool rmactivity**. For more information about **rmactivity**, see *ClearCase Reference Manual*.

To Delete a ClearQuest Activity

1. Use a query to find the activity. For example, to use **MyToDoList**:

   a. Click **Start>Programs>Rational ClearQuest>ClearQuest.**

   b. In the left pane, click **Workspace>Public Queries>UCMUserQueries** and run the MyToDoList query.

      ClearQuest displays MyToDoList in the Query builder pane. (Figure 19 on page 43.)

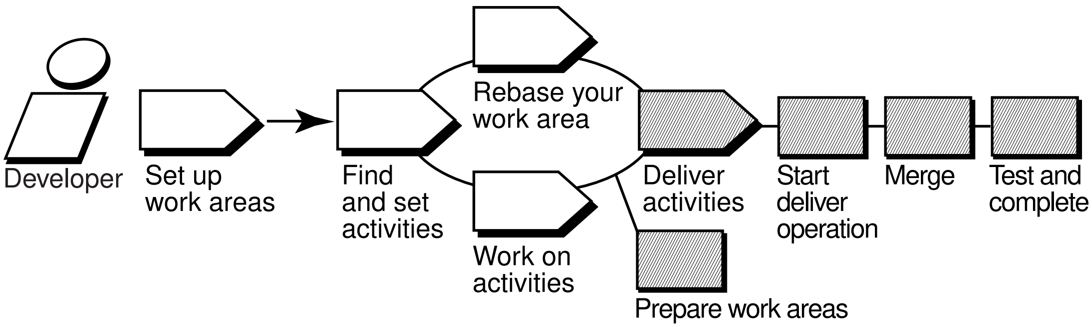
2. In the Query builder pane, select an activity.

3. In the Record form pane, click **Actions>Delete**. If a ClearQuest activity is linked with a ClearCase activity (and the activity's change set is empty), ClearQuest deletes the activity in the stream as well.

4. Click **Apply**.

Delivering Activities

5



When you're ready to make the activities in your development stream available to the rest of the project team (for example, to contribute to official project builds), you deliver your work. Delivering activities involves the following tasks:

- Preparing your work areas
- Starting the deliver operation
- Selecting activities
- Merging versions
- Testing your work
- Completing the deliver operation

NOTE: If your project uses MultiSite to share source data with developers in other geographical locations and if the project's streams are mastered at a different site, you complete only the first three tasks; the project integrator completes the remaining tasks. For more information, see *MultiSite: Posting Work to Deliver* on page 86.

5.1 Preparing Your Work Areas

Before starting the deliver operation, prepare your work areas as follows:

- If your project manager has created a new *recommended baseline* since you last rebased, rebase your development work area to minimize the amount of merging needed during the deliver operation. For information on rebasing, see Chapter 6, *Rebasing Your Work Area*.
- In your development work area, find, compare, and check in all the work you want to deliver.
- If your integration view is a snapshot view, update it and resolve any *hijacked files*.

Finding, Comparing, and Checking In Work from Your Development View

ClearCase delivers only checked-in changes. In addition, your project manager may have set up your UCM project to require that you check in all files and directories in your development view before starting the deliver operation.

To Find, Compare, and Check In Your Work

1. In the ClearCase Explorer Folder window (middle), right-click the view's root directory and then click **Find Checkouts**.
2. In the **Find Criteria** dialog box, select these criteria:
   - **Include subfolders**
   - **Search only for your checkouts**
   - **Search only for checkouts in the current view**
3. Then click **OK**.

   Find Checkouts produces a list of file or directory versions checked out to your development view and indicates which activity each checked-out version belongs to.

4. To see which changes are in a checked-out version:
   - To see the differences between the previously checked-in version and the current version, click **Tools>Compare with Previous Version**.
   - To see checkin comments, click **Tools>History**.

5. Do **one** of the following:

   ➢ To deliver changes for a given file, check it in by selecting it and clicking **Tools>Check In**.

   ➢ To cancel your changes for a file, undo its checkout by selecting it and clicking **Tools>Undo Checkout**.

   During the cancel checkout operation, you can choose to save your changes in a *view-private file*.

6. To keep your changes without delivering a file, leave it checked out. If you leave a file checked out, ClearCase delivers the predecessor version in the activity's change set. If the checked-out version is the only version in the change set, ClearCase does not deliver any version of the element.

Updating Your Integration View

During the deliver operation, use your integration view to make sure that your delivered work builds successfully with the work other team members have delivered.

If your integration view is a snapshot view, update it and resolve any *hijacked files* to ensure that it contains the latest versions delivered to the integration stream. For more information, see ClearCase online help and the **view** reference page in *ClearCase Reference Manual*.

To Update a Snapshot View and Resolve Hijacked Files

1. In ClearCase Explorer, right-click the root directory of the view and click **Update View**.

2. In the **Start Update** dialog box, click **Advanced** and select an option for resolving *hijacked files*. (Refer to the online help for information on each option.)

3. To see the actions needed to update the view without actually updating, click **Preview only** on the **View** tab.

4. Click **OK**.

   ClearCase displays its progress as it updates the view (or previews the update).

5. In the Update window, check out or undo any hijacked files. For more information, refer to ClearCase online help.

5.2 Starting the deliver Operation

If your project uses the UCM-ClearQuest integration, your project manager may enable a ClearQuest policy called **Check Before ClearCase Delivery**. This policy runs a customized ClearQuest script when you start the deliver operation. For example, your project manager may create a script that checks for dependencies between ClearQuest records. Before you can continue the deliver operation, you must satisfy any requirements established by the **Check Before ClearCase Delivery** policy.

To Start Delivery from ClearCase Explorer

1. In the ClearCase Explorer Shortcut pane, click **Toolbox**. Then click **UCM>Deliver from Stream**.

2. Select your development stream and click **OK**.

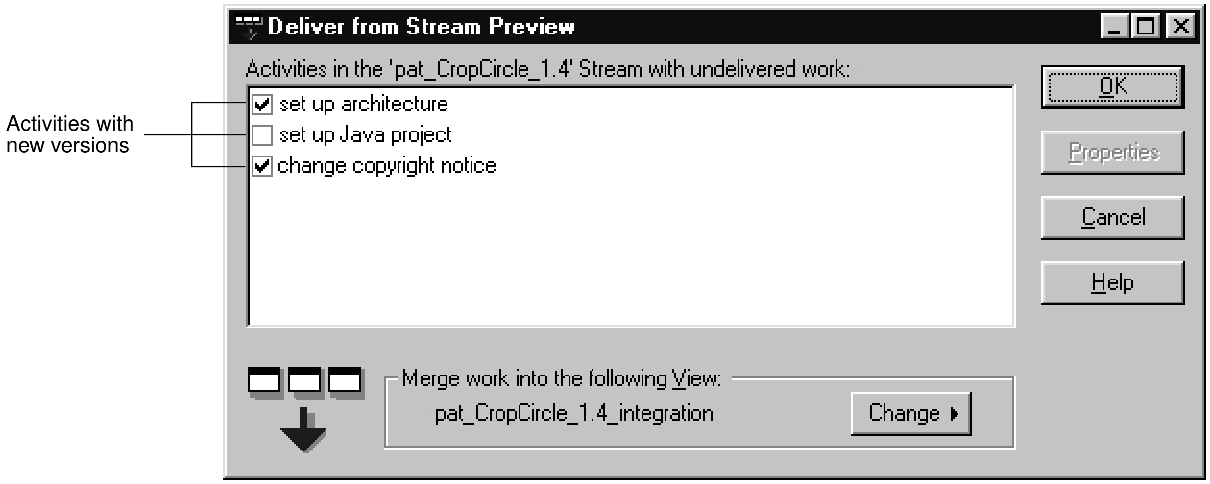
To Start Delivery from Windows Explorer

Right-click the root directory of your development view and then click **ClearCase>Deliver from Stream**.

5.3 Selecting Activities

After you start the deliver operation, ClearCase opens the **Deliver from Stream Preview** dialog box to display the activities in your development stream that have never been delivered or that include versions created since the last deliver operation (Figure 26).

Figure 26 Undelivered Activities



You can select all activities in the dialog box or a subset. If you select a subset of the activities, ClearCase checks for dependencies between versions in the change sets before continuing the operation. If your selection does not meet dependency requirements, ClearCase prompts you to change it.

To see an activity's change set, select the activity, click **Properties**, and click the **Change Set** tab in the **Properties** dialog box.

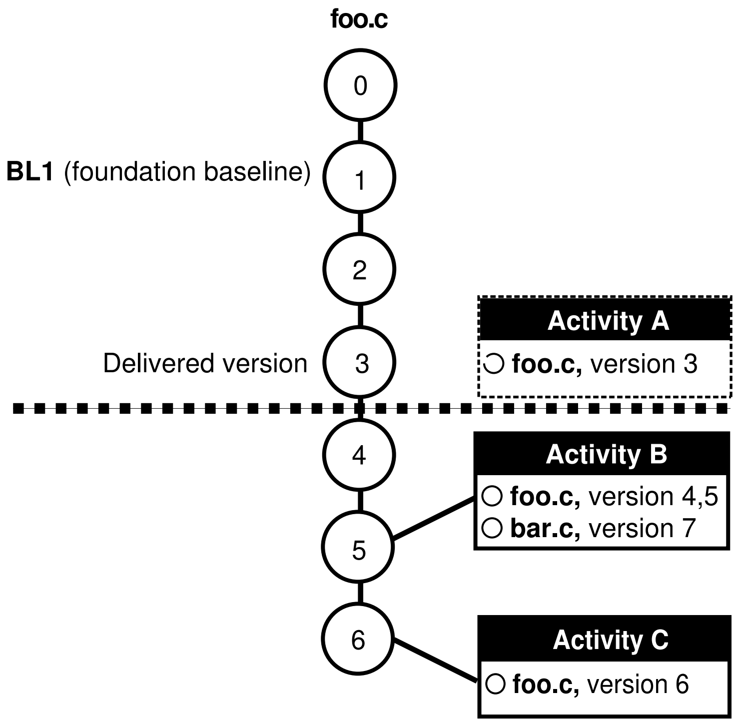
Activity Dependencies in the deliver Operation

For any given element in an activity's change set, you must deliver all versions in your stream in the range between the version in the change set and either the version most recently delivered or the version in the stream's *foundation baseline* (whichever is later). For example, in Figure 27, because **version 3** of **foo.c** was delivered in a previous deliver operation, the dependencies between **Activities B** and **C**, which contain versions of **foo.c**, are as follows:

- You can deliver **Activity B** without delivering **Activity C**.
- To deliver **Activity C**, which contains **version 6**, you must also deliver **Activity B**, which contains the versions between **6** and the most recent delivery of **foo.c**.
- Because **Activity B** also contains versions of **bar.c**, you may be required to deliver other activities to satisfy dependencies for **bar.c**. For example, if **Activity D** (not shown in

Figure 27) contained an undelivered **version 6** of **bar.c**, delivering **Activity B** would require you to deliver **Activity D** as well.

Figure 27 Activity Dependencies



5.4 Merging Versions

NOTE: If your project uses MultiSite to share source data with developers in other geographical locations, and if the project's streams are mastered at a different site, you do not merge versions. For more information, see *MultiSite: Posting Work to Deliver* on page 86.

After selecting activities, click **OK** in the **Deliver from Stream Preview** dialog box to start merging versions in your development view with versions in the integration view. As it merges files and directories, ClearCase displays its progress in the **Delivering to View** dialog box.

For each new version delivered to the integration view, ClearCase merges all nonconflicting differences. As it merges versions, ClearCase changes the value in the Merged column of the

Delivering to View dialog box to **Yes**. If versions in your development view contain changes that conflict with the corresponding versions in the integration view, ClearCase starts Diff Merge, a tool that helps you resolve conflicting differences.

Under the Hood: Concurrent Deliver Operations

ClearCase allows developers to deliver activities to the integration stream concurrently. However, if another team member's deliver operation has already checked out a given element, you cannot deliver any changes to that element until that deliver operation is completed or canceled.

When delivering activities, ClearCase follows a specific order:

1. It evaluates each directory sequentially, checking out and merging changes to the integration stream when needed. If ClearCase encounters a reserved directory checkout in the integration stream, it stops the deliver operation (but does not undo any successful directory merges).

2. It then attempts to check out each element that requires a merge. If ClearCase encounters an element that it cannot check out (for example because another team member's deliver operation has already checked out the element), it skips the element and attempts to check out the remaining elements.

   If ClearCase is unable to check out an element, it does not proceed to the following step. In this case, you have two options:

   - Wait until the other developer completes or cancels her deliver operation, and then restart your deliver operation. Note, however, that with a deliver operation in progress, any new versions you create in your development stream are not delivered until the next deliver operation.

   - Undo your deliver operation and continue working on your activities. (See *To Undo a deliver Operation* on page 85 for more information.)

3. After successfully checking out all elements, it merges each element, starting Diff Merge and requesting your input when it encounters merge conflicts.

Under the Hood: Integration Activities and Baselines

ClearCase uses a special kind of activity, an integration activity, to keep track of the changes that occur during deliver and rebase operations. ClearCase creates integration activities and adds versions to them; you do not need to create or maintain them. You may want to view an integration activity's properties to see exactly which versions were merged and created during a deliver or rebase operation. You can view an integration activity's properties from the Project Explorer; the activity name is of the form *deliver.stream-name.date-stamp*.

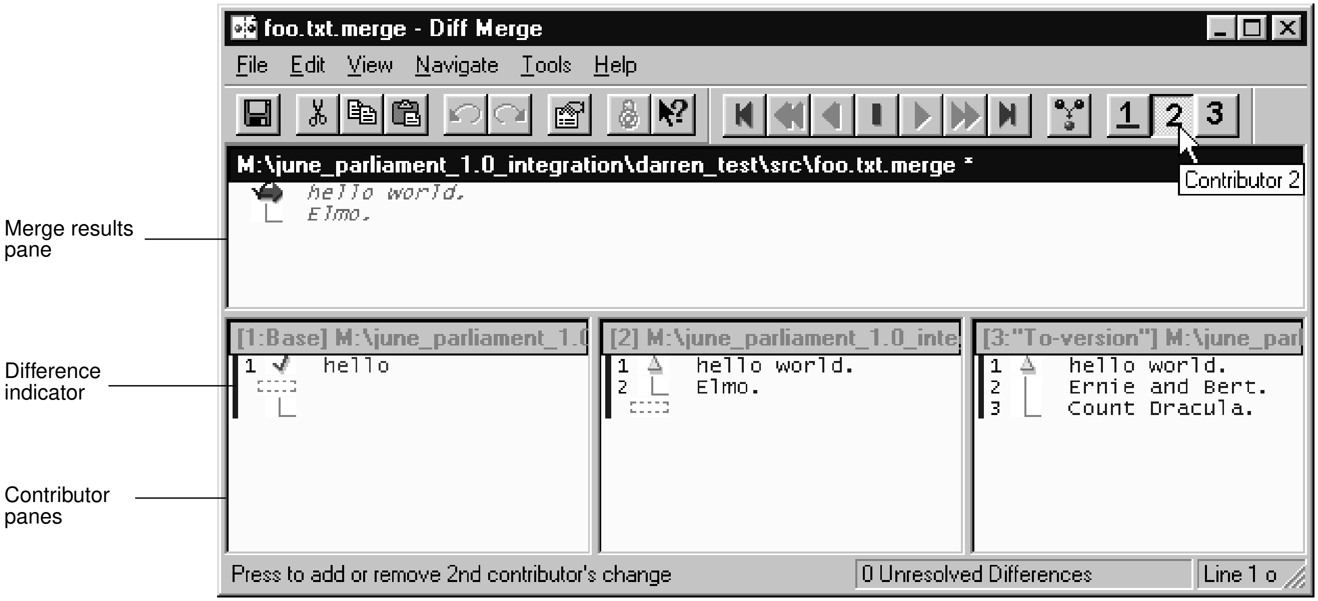
If your project uses the UCM-ClearQuest integration, ClearQuest links a UCM-enabled record based on the UCMUtilityActivity record type with the ClearCase activity. During the deliver or rebase operation, the activity is in a state based on the Active state type. After the operation, ClearQuest moves the activity to a Complete state type.

As an intermediate step in the delivery process, ClearCase creates a baseline in your development stream that identifies the state of your stream at the time of delivery. You can browse this baseline and its properties in the Component Browser. For more information, see *Managing Software Projects with ClearCase*.

Using Diff Merge to Resolve Differences

When Diff Merge starts, it indicates that it has resolved all nonconflicting differences. Then, it highlights the first unresolved difference with a vertical blue line at the left edge of the contributor panes (Figure 28).

Figure 28 Diff Merge Window



To Resolve Differences

1. In Diff Merge, compare the differences in the contributor panes and click the numbered button on the navigation toolbar that corresponds to the contributor pane number containing the resolution you prefer.

   For example, Figure 28 shows that clicking **2** moves the difference region in contributor pane 2 to the results pane.

   In some cases, you can click more than one numbered button to include changes from multiple contributor panes. In addition, after you select a contributor to resolve the difference, you can edit the results pane directly.

2. Click **Navigate>Next Unresolved Difference**.

3. Repeat Step 1 and Step 2 until you have resolved all unresolved differences.

4. Click **File>Save** and then click **File>Exit**. When you complete the deliver operation, ClearCase checks in the merge results.

NOTE: For more information on using Diff Merge, choose a command from the Diff Merge **Help** menu.

After you exit Diff Merge, ClearCase continues merging the versions in your activities.

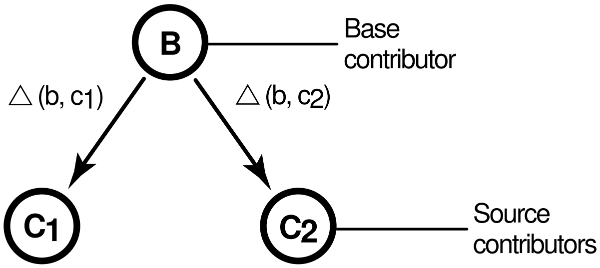
Under the Hood: How ClearCase Merges Files and Directories

A merge combines the contents of two or more versions into a single new version. During a deliver operation, the ClearCase merge algorithm involves the following versions:

- The source contributors: one version from your development stream and one version from the integration stream.
- The base contributor: the common ancestor of the source versions.
- The destination version: the merge output, which becomes a new version in the integration stream.

As illustrated in Figure 29, ClearCase starts with the base contributor and adds changes that are present in the source contributors.

Figure 29 ClearCase Merge Algorithm



Destination version = B + △ (b, c_1) + △ (b, c_2)

If the two source contributors contain conflicting changes (for example, if each contains a line that is different from the base contributor), ClearCase starts the Diff Merge tool so you can choose which change to add to the destination version.

For more information on the ClearCase merge algorithm, see the **merge** reference page in *ClearCase Reference Manual*.

Handling a Binary File in a deliver Operation

If you have a binary file in any of your activities when you perform the deliver operation, and your organization does not have a type manager for that file type, ClearCase generates an error when it encounters the binary file during the delivery. The error occurs because there is no type manager that can merge the version of the file in your development stream with the version of the file in the integration stream. Your project manager can overcome this problem by creating a special element type for the binary file type and associating that element type with the binary file in the VOB. This special element type can tell ClearCase not to attempt merging the file. See *Managing Software Projects with ClearCase* for more information.

Stopping the deliver Operation

You can stop the deliver operation temporarily by clicking **Stop** in the **Delivering to View** dialog box. Resume the operation by clicking **Resume** or by right-clicking the root directory of the view in ClearCase Explorer and clicking **Deliver from Stream**.

5.5 Testing Your Work

At this point in the deliver process, the destination versions (the new versions resulting from the merges) are checked out and visible only in your integration view.

In addition to your delivered activities, your integration view shows the work other team members have delivered. To make sure that your work is compatible with this, we recommend that you test the work in your integration view before completing the deliver operation.

To go to your integration view, click **OK** in the **Merges Complete** dialog box. ClearCase opens your integration view in ClearCase Explorer.

An additional dialog box, **Deliver Status**, remains open. Consider leaving it open to facilitate completing the deliver operation after you build and test.

If your organization uses **clearmake** or **omake**, you can use these ClearCase build tools for your builds; however, the *build auditing* and *build avoidance* features are available only from dynamic views.

For more information about **clearmake** or **omake**, refer to their reference pages in *ClearCase Reference Manual* and to *Building Software with ClearCase*.

In addition to building and testing, you may need to do the following:

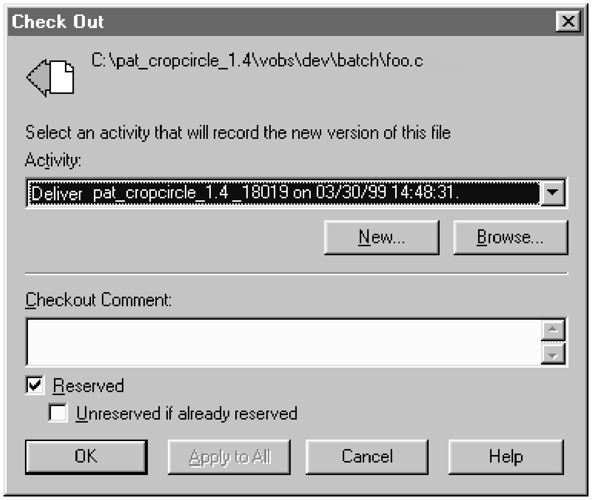
- ➤ Edit the versions checked out to your integration view to resolve build errors
- ➤ Check out and edit additional elements to resolve build errors
- ➤ Update your integration view
- ➤ Undo a deliver operation

Checking Out Versions

In the process of building and testing, you may need to check out versions in your integration view and modify them to resolve build errors. As described in *Under the Hood: Integration Activities and Baselines* on page 80, ClearCase adds any modifications you make during the deliver operation to an integration activity (Figure 30).

NOTE: We recommend that you postpone checking in your modifications until you complete the delivery. Checking in complicates efforts to undo the deliver operation, if you find it necessary to do so.

Figure 30 Integration Activity



Under the Hood: Checking Out Versions from Snapshot Views

If your project allows concurrent deliveries and if your integration view is a snapshot view, the view becomes out of date when other team members deliver to the integration stream. If you try to check out a version that is not the latest in the integration stream, ClearCase prompts you to check out the latest version in the integration stream and loads it into your view.

Undoing a deliver Operation

At any time before completing the deliver operation, you can undo changes made as a result of the operation.

This procedure is valid only if you have started a deliver operation and have not completed it.

To Undo a deliver Operation

1. If you started the Diff Merge tool, close it.
2. If the **Deliver Status** dialog box is open, click **Close**.
3. In ClearCase Explorer, right-click the view's root directory and click **Deliver from Stream**.
4. In the dialog box, click **Undo the delivery** and click **OK**.

ClearCase cancels all checkouts in the integration view and removes all *merge arrows* created during the current deliver operation. (ClearCase uses merge arrows in VOBs to keep a history of merges.)

If you checked in any versions to the integration view, clicking **Cancel** is not sufficient to undo the deliver operation. You must either complete the operation (recommended) or use **cleartool rmver –hlink** to remove the versions you checked in before canceling the operation. Because **rmver** erases part of your organization's development history, your organization may not allow individual developers to use it. Because it may have unintended consequences, be very conservative in using this command, especially with the **–hlink** option. For information on using **cleartool rmver**, see the **rmver** reference page in *ClearCase Reference Manual*.

5.6 Completing the deliver Operation

Because the deliver operation is not instantaneous (for example, you may need to build, test, and correct errors several times during a deliver operation), ClearCase imposes a simple state model: deliveries are either *in progress* or *complete*, and you cannot start a new deliver operation from a development stream when one is already in progress for the stream.

To Complete a deliver Operation

1. If it isn't open, open the **Deliver Status** dialog box as follows:

   In ClearCase Explorer, right-click the view's root directory and click **Deliver from Stream**.

2. In the dialog box, click **Complete the delivery** and click **OK**.

What Happens When You Complete a deliver Operation

When you issue the **complete** command for a deliver operation, ClearCase does the following:

1. It checks in the merge results in the integration stream.

2. If your project is enabled for ClearQuest and your project manager has enabled the DoClearQuestActionAfterDelivery policy, ClearCase executes a ClearQuest script. If the script returns an error, ClearCase stops the deliver operation. You must correct the error before ClearCase can continue.

3. It changes the state of the operation to **Complete**.

5.7 MultiSite: Posting Work to Deliver

If your organization uses MultiSite to distribute development among multiple geographical sites, your project may share source files with developers at other sites. Each site has its own replica of project *components*, and developers work in site-specific replicas. Each replica controls (masters) a set of streams, and only developers at that replica's site can modify them. If your site does not master a stream, you cannot deliver work from or to that stream. Complete the following tasks to make your work on a remote stream available to the rest of the project:

➤ Prepare your work areas (see *Preparing Your Work Areas* on page 74)
➤ Start the deliver operation (see *Starting the deliver Operation* on page 76)
➤ Select activities (see *Selecting Activities* on page 76)
➤ Post work to be delivered

After you post your work, the project integrator at the site that masters your project's integration stream completes the deliver operation.

Posting Work To Be Delivered

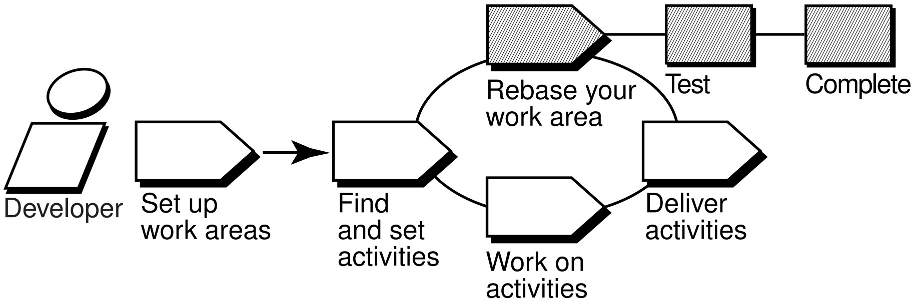
The **deliver** operation determines whether your development and integration streams are mastered at different replicas. If they are, after you select activities to deliver, ClearCase changes mastership of the development stream object to match that of the integration stream object, and it notifies you that it has assigned the development stream the *posted* status.

After posting your work, notify the project integrator at the site that masters the integration stream's replica. The project integrator can then continue the operation or cancel it to return the development stream to its previous state. Note that, after you post your work, only someone at the master site can cancel or complete the operation.

You can create activities and perform checkins and checkouts for your development stream while the remote deliver is in process. However, you cannot add, remove, or create baselines; add or remove components; or rebase the development stream.

Rebasing Your Work Area

6



Periodically, your project manager groups delivered activities into baselines. Some of these baselines constitute a stable and significant source configuration, and your project manager will recommend that you rebase your development work area to the recommended configuration.

To reduce the number of merges and possible build errors while integrating your work, rebase your development work area each time your project manager creates a new recommended baseline.

Rebasing involves the following tasks:

- Preparing your development view
- Starting the rebase operation
- Testing your development work area
- Completing the rebase operation

6.1 Preparing Your Development View

Before starting, you must check in all files and directories in your development view. You cannot start a rebase operation from a view that contains checkouts.

To Prepare Your Development View

1. In the ClearCase Explorer Folder window (middle), right-click the view's root directory and then click **Find Checkouts**.

2. In the **Find Criteria** dialog box, select these criteria:

   ➢ **Include subfolders**
   ➢ **Search only for your checkouts**
   ➢ **Search only for checkouts in the current view**

3. Then click **OK**.

   Find Checkouts produces a list of file or directory versions checked out to your development view and indicates which activity each checked-out version belongs to.

4. To see which changes are in a checked-out version:

   ➢ To see the differences between the previously checked-in version and the current version, click **Tools>Compare with Previous Version**.

   ➢ To see checkin comments, click **Tools>History**.

5. Do **one** of the following:

   ➢ To deliver changes for a given file, check it in by selecting it and clicking **Tools>Check In**.

   ➢ To cancel your changes for a file, undo its checkout by selecting it and clicking **Tools>Undo Checkout**.

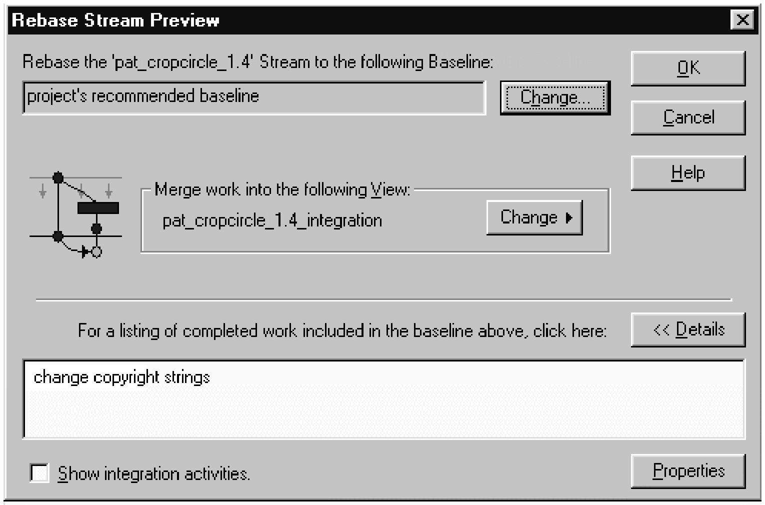
   During the cancel checkout operation, you can choose to save your changes in a *view-private file*.

6.2 Starting the rebase Operation

After you prepare your development view (see *Preparing Your Development View* on page 90), you can start the rebase operation. To start the rebase operation, right-click the root directory of the development view in ClearCase Explorer and click **Rebase Stream**.

ClearCase opens the **Rebase Preview** dialog box (Figure 31).

Figure 31 Rebase Preview Dialog Box



By default, the **Rebase Preview** dialog box presents the latest recommended baseline. To choose a different baseline, click **Change**. You cannot choose a baseline that is older than the baseline currently in your development stream.

You can also do the following in the **Rebase Preview** dialog box:

➤ Verify that the dialog box displays your development view under **Merge work into the following view**. If you associated multiple views with your development stream, you must choose one view to use for the rebase operation. You cannot issue ClearCase commands from other views attached to the development stream until you complete the operation.

➤ To see the changes that will be available in your development stream after rebasing to the displayed baseline, click **Details**. The dialog box expands to show the list of activities in the baseline. To see more information about an activity, select it and click **Properties**.

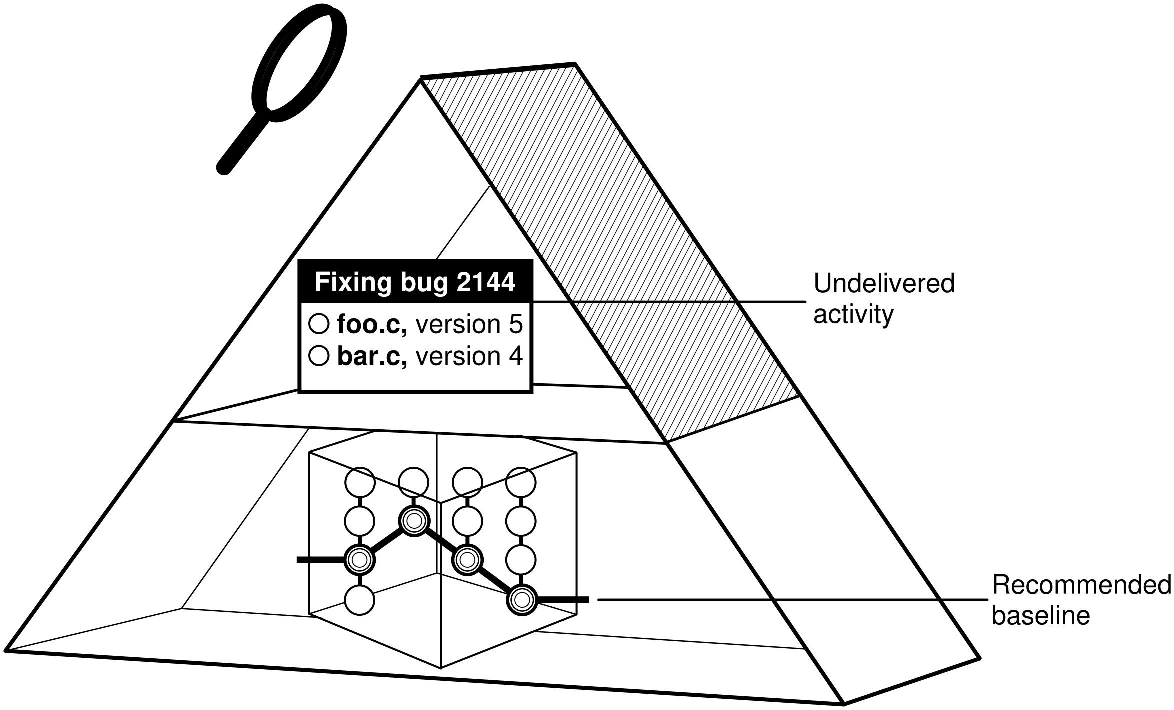
To continue the rebase operation, click **OK**.

Rebasing Your Development Work Area

To rebase your work area, ClearCase takes the following actions:

- It changes your development stream's configuration to select the new recommended baseline. The stream's configuration continues to select any undelivered activities (Figure 32).
- For development views that are *snapshot views*, it starts an update operation to copy versions in the new baseline into your view. If there are other snapshot views attached to your development stream, you must update them so that they contain the changes in the stream. Development views that are dynamic views show versions in the new baseline without requiring an update operation.

Figure 32 Development Work Area After Rebasing



As it rebases your development work area, ClearCase displays its progress in the **Rebasing Stream** dialog box.

You can stop the rebase operation temporarily by clicking **Stop** in the **Rebasing Stream** dialog box. To resume the operation, click **Resume** in the **Rebasing Stream** dialog box or right-click the root directory of the view in ClearCase Explorer and then click **Rebase Stream**.

Merging Versions

For any version you modify in your development stream, if another team member has modified and delivered a version of the same element, you must merge when you rebase to a baseline that contains the delivered version.

For example:

1. As part of working on an activity, you check in a version of **foo.c** from your development view.
2. Another developer checks in and delivers a new version of **foo.c**.
3. Your project manager adds the new version to the recommended baseline.
4. You rebase your development stream. As part of the rebase operation, you must merge the other developer's version into your development stream.

As it does during the deliver operation, ClearCase merges all nonconflicting differences. If versions in the baseline contain changes that conflict with the corresponding versions in your development work area, ClearCase prompts you to start Diff Merge, a tool with which you resolve conflicting differences. For information on using Diff Merge to resolve conflicting differences, see *Using Diff Merge to Resolve Differences* on page 80.

6.3 Testing Your Development Work Area

After ClearCase changes your stream's configuration and completes any needed merges, verify that any undelivered work builds successfully with the versions in the new baseline by initiating a test build in your development view. If your organization uses **clearmake** or **omake**, you can use these ClearCase build tools for your builds; however, the *build auditing* and *build avoidance* features are available only from dynamic views.

For more information about **clearmake** or **omake**, refer to their reference pages in *ClearCase Reference Manual* and to *Building Software with ClearCase*.

In addition, you may need to do the following:

- Check out elements
- Undo a rebase operation

Checking Out Elements

In the process of building and testing, you may need to check out elements in your development view and modify them. As described in *Under the Hood: Integration Activities and Baselines* on page 80, ClearCase creates an integration activity to record any modifications you make during the rebase operation.

NOTE: We recommend that you postpone checking in your modifications until you complete the rebase. Checking in complicates efforts to undo the rebase operation, if you find it necessary to do so.

Undoing a rebase Operation

At any time before completing the rebase operation, you can undo changes made as a result of the operation.

This procedure is valid only if you have started a rebase operation and have not completed it.

To Undo a rebase Operation

1. If you started the Diff Merge tool, close it.

2. If the **Rebase Status** dialog box is open, click **Close**.

3. In ClearCase Explorer, right-click the view's root directory and click **Rebase Stream**.

4. In the **Rebase Stream** dialog box, select **Undo the rebase** and click **OK**.

ClearCase cancels all checkouts in the development view and removes all *merge arrows* created during the current operation. (ClearCase uses merge arrows in VOBs to keep a history of merges.)

If you checked in any versions to the development view, you must remove them using the **cleartool rmver** command. Because **rmver** erases part of your organization's development history, your organization may not allow individual developers to use it. Be very conservative in using this command.

For information on using **cleartool rmver**, refer to the **rmver** reference page in *ClearCase Reference Manual*.

6.4 Completing the rebase Operation

As with the deliver operation, ClearCase imposes a simple state model: rebase operations are either *in progress* or *complete*, and you cannot start a new operation for a development stream when one is in progress.

Completing a rebase operation consists of two tasks: checking in any merge results and changing the state of the operation to complete.

After testing your work, click **Complete** in the **Rebase Status** dialog box. ClearCase checks in any versions checked out to your development view and notifies your development stream that the rebase operation is complete.

If you closed the **Rebase Status** dialog box before completing the operation, complete it as follows:

1. In ClearCase Explorer, right click the view's root directory and click **Rebase Stream**.
2. In the dialog box, click **Complete the rebase** and click **OK**.

7 Other Development Tasks

Chapter 4, *Working on Activities*, describes tasks that you perform daily or weekly. You may need to perform some of these tasks less often:

- Adding files and directories to source control
- Moving, removing, and renaming elements
- Accessing elements not loaded into a snapshot view
- Adjusting the scope of a view
- Assigning a snapshot view to a drive letter
- Moving work from a development stream to another project
- Registering a snapshot view
- Moving views
- Access views and VOBs across platform types
- Regenerating the **view.dat** file

7.1 Adding Files and Directories to Source Control

You can add files or directories to source control at any time during the development cycle. Usually, you add a few files to a directory that is already under source control. Less frequently, you may need to add an entire directory tree to source control.

This section explains the following tasks:

- Adding files and directories to an existing directory
- Adding a directory tree to source control

Adding to an Existing Directory Tree

To add files and directories to source control to an existing directory tree:

1. In ClearCase Explorer, navigate to the view used for your development task.
2. Navigate to the parent directory to which you want to add the files or directories.
3. If the files or directories are not present, drag them to the parent directory.
4. Select the files and directories you want to add to source control.
5. Right-click one of the selected objects. On the shortcut menu, click **Add to Source Control**.

   We recommend you select items that are the furthest from the root of the directory tree: the **Add to Source Control** command for any given file or directory also adds any parent directories (up to the VOB root directory) that are not already *elements*.

   ClearCase opens the **Check Out** dialog box with the currently set activity selected in the **Activity** list.

6. To set a different activity (or, if you have not set the view to an activity), do the following:

   a. Select one from the **Activity** list. To filter the list, click **Browse**.

   b. If an appropriate activity doesn't exist, create and set a new one by clicking **New**. If your project is enabled for ClearQuest, choose a record-type from the shortcut menu that appears when you click **New**. ClearQuest changes the activity to an Active state type, and ClearCase adds the version you check out to the activity's change set.

7. Click **OK**.

Under the Hood: What Happens When You Add a File or Directory to Source Control

The **mkelem** or **Add to Source Control** command always creates an element and initializes its version tree by creating a single branch (named **main**) and a single, empty version (version 0) on that branch. The following arguments for the **mkelem** command or options in the **Add to Source Control** dialog box determine optional ClearCase behavior:

- Selecting the **Keep checked out** check box or using **mkelem** with no arguments checks out the element. Any view-private data that corresponds to the element pathname remains in your view only and is added to version 1 in the VOB when you check in (Figure 33).

- Clearing the **Keep checked out** check box or using **mkelem –ci** checks in the element, using any existing view-private data that corresponds to the element pathname as the content for version 1. Your view's config spec determines the branch on which ClearCase creates version 1.

- Using **mkelem –nco** suppresses automatic checkout; **mkelem** creates the new element, along with the **main** branch and version **\main\0**, but does not check it out. If *element-pathname* exists, it is moved aside to a **.keep** file.

- (Replicated VOBs only) Selecting the **Make current replica the master of all newly created branches** check box or using **mkelem –master** assigns to your *current replica* mastership of all branches created during element creation. You will be able to create new versions on the branches.

  Clearing this check box or using **mkelem** without the **–master** option assigns mastership of a new branch to the VOB replica that masters the associated branch type. If this replica is not your current replica, you cannot create new versions on the branch.

  You can set the default for mastership assignment in the **User Options** dialog box:

  a. In ClearCase Explorer, click **Tools>Options**.

  b. In the **User Options** dialog box, click **ClearCase Options**.

  c. In the **ClearCase User Options** dialog box, click the **Operations** tab.

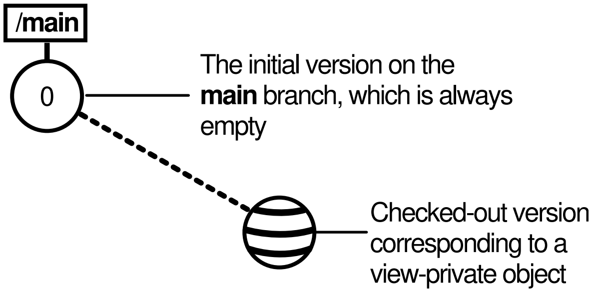
  d. Click **Advanced Options**.

e. Select the **When creating an element in a replicated VOB, make current replica the master of all newly created branches** check box.

f. Click **OK**.

Other views do not see the element until you check in the element's parent directories (the **Add to Source Control** command does this for you) and check in the file or directory.

Figure 33 Creating an Element



NOTE: *VOB links* make it possible to have more than one copy of a directory in a snapshot view. When you add a file or directory to a snapshot view, ClearCase updates all other instances of the parent directory that are loaded into your view.

Adding Elements for a New Development Task

NOTE: This procedure assumes that a VOB exists. For information on creating VOBs, see *Managing Software Projects with ClearCase*.

1. In ClearCase Explorer, navigate to the view you created for the development task.

2. If you work in a snapshot view, create a folder in the view that corresponds to each VOB to which you want to add files and directories.

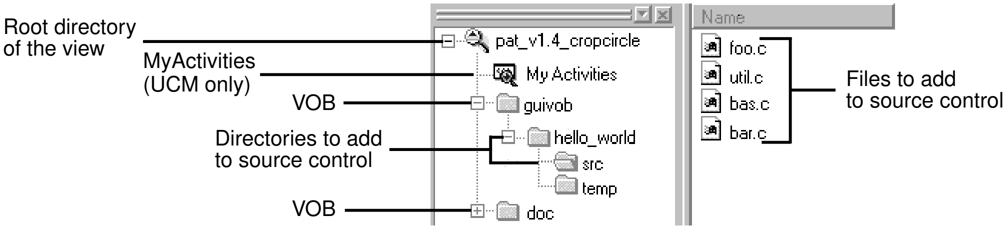
   If you work in a dynamic view, make sure that each VOB to which you want to add files and directories is accessible from your view. If a VOB isn't accessible, activate it on your computer by clicking **Start>Programs>ClearCase Administration>Mount VOB**.

3. Under each VOB, do one of the following:

➢ Create a directory structure.
➢ If files and directories already exist, copy them into the appropriate location within the directory tree.

When you are finished, the view resembles Figure 34.

Figure 34 Directory Structure in a Snapshot View



4. In the right upper pane of ClearCase Explorer, select files and directories.

We recommend that you select items that are the farthest from the root of the directory tree: the **Add to Source Control** command for any given file or directory also adds any parent directories (up to the VOB root directory) that are not already *elements*. For example, in Figure 34, if neither **\guivob\hello_world** nor **\guivob\hello_world\src** are elements, you can add both directories to source control by adding **\guivob\hello_world\src** to source control.

5. Right-click one of the selected objects. On the shortcut menu, click **Add to Source Control**.

Importing Files

If you're adding a large number of files and directories to source control, use the **clearexport_ffile** command (or other **clearexport** commands) and **clearimport** command. For more information, see the **clearexport_ffile** and **clearimport** reference pages in *ClearCase Reference Manual*.

7.2 Moving, Removing, and Renaming Elements

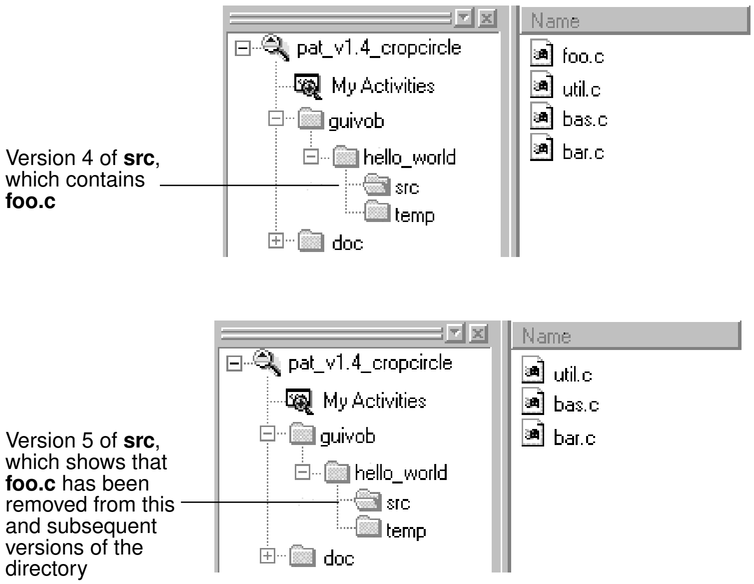
This section explains how to move, remove, and rename elements.

Moving and Removing Elements

Because directories as well as files are under ClearCase control, you can move or remove elements from specific versions of directories without affecting the element itself. Moving or removing elements creates new versions of the parent directories to record the modifications.

For example, version 4 of **\gui_vob\hello_world\src** contains an element named **foo.c**. If you remove **foo.c** from the **hello_world\src** directory, ClearCase creates version 5 of **\gui_vob\hello_world\src**, which does not contain the **foo.c** file element. The element **foo.c** itself is not modified (Figure 35).

Figure 35 Removing an Element Name from a Directory



Before you move or remove an element name from a directory, verify with your project manager that your changes will not adversely affect other team members or break projectwide builds.

To Move an Element Within a VOB

Do **one** of the following:

- ➤ In the ClearCase Explorer Details pane, drag an element to its destination directory in the Folder Window.
- ➤ In the ClearCase Explorer Details pane, right-click an element and click **Cut**. In the Details pane, right-click the destination directory and click **Paste**.

To Move an Element to Another VOB

Use the **cleartool relocate** command.

WARNING: The **relocate** command makes irreversible changes to at least two VOBs and their event histories. We recommend that you not use it for minor adjustments. Furthermore, you are advised to stop VOB update activity before and during a relocate operation. Check with your project manager and ClearCase administrator before using the **relocate** command.

To Remove an Element Name from a Directory

In the ClearCase Explorer Details pane, right-click an element and click **Delete**.

Other Methods for Removing Elements

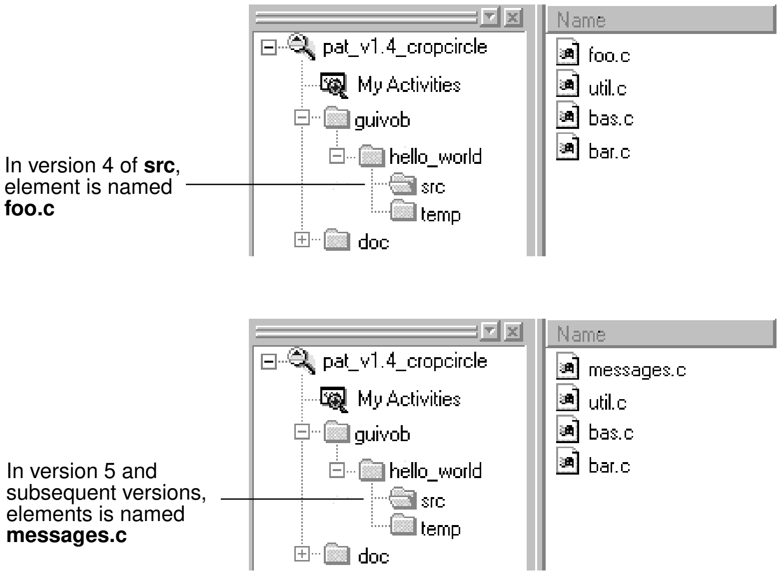
Removing an element from its parent directory does not affect the element itself, but two other types of a removal operation do irrevocably affect an element, and we recommend that you be very conservative in using these operations:

- ➤ Removing a version from an element's version tree. For more information, see the **rmver** reference page in *ClearCase Reference Manual*.
- ➤ Removing an element from a VOB. For more information, see the **rmelem** reference page in *ClearCase Reference Manual*.

Renaming Elements

Renaming an element creates a new version of the parent directory to catalog the new element name. The element uses its new name in subsequent versions of its parent directory, but previous versions of the parent directory refer to the element by its previous name (Figure 36).

Figure 36 Renaming an Element



Before you move or remove an element name from a directory, verify with your project manager that your changes will not adversely affect other team members or break project builds.

To Rename an Element

1. In the ClearCase Explorer Details pane, right-click an element and click **Rename**.
2. In the Details pane, type a new name for the element
3. Press ENTER.

7.3 Accessing Elements Not Loaded into a Snapshot View

While working with source files in a snapshot view, you may need to see the contents of elements that are not loaded into the view or see ClearCase information about these nonloaded elements. For example, you may have chosen not to load a VOB that contains functional-specification documents. However, you might want to check periodically whether the functional specifications have been modified by reviewing the element's ClearCase history.

Listing All Elements in the VOB Namespace

You can set up ClearCase Explorer to provide a directory listing of nonloaded file and directory elements. ClearCase Explorer displays the version of the element that would be selected if the element were loaded in the view. The Version column in the Details pane indicates which version the view is selecting. Then, you can display the nonloaded element's history, version tree, and properties sheet, and compare versions of the element.

To See Nonloaded Elements from ClearCase Explorer

1. In ClearCase Explorer, click **Tools>Options**.
2. In the **Options** dialog box, under **Snapshot View Options**, select **Show Unloaded Snapshot View Elements**.
3. Click **OK**.

   The Details pane uses an icon and the State column to indicate which versions are not loaded.

To See Metadata for Nonloaded Versions

Metadata is information that ClearCase keeps about elements under source control. To see metadata for a nonloaded version, click the version in the Details pane and select commands from the **Tools** menu.

Viewing the Contents of a Nonloaded Version

To see the contents of a nonloaded version, you can do either of the following:

- ➤ Use the **Send To** command from the version's shortcut menu in the Version Tree Browser. The **Send To** command uses the set of shortcuts in your Send To folder, which is a function that the Windows operating system provides. For information about your Send To folder, refer to Windows online help.
- ➤ Copy the version into your view with the **cleartool get** command.

You can view nonloaded files or copy them into your view for build purposes, but you cannot check them out. Only file elements that are loaded into the view can be checked out.

NOTE: You cannot use the **Send To** command or **cleartool get** for directory elements.

To Copy a Nonloaded Version with cleartool get

1. In the ClearCase Explorer Details pane, in the Version column, note the *version-extended pathname* of the unloaded element, which is displayed in the Version column.

2. Right-click the parent directory and from the shortcut menu, click **Command Prompt** to display a **cleartool** prompt.

3. Type the **cleartool get** command using the following syntax:

   get –to *filename version-extended-pathname*

   For example, **get –to foo.c.previous.version foo.c@@\main\v3.1_fix\10** copies **foo.c@@\main\v3.1_fix\10** into your view under the name of **foo.c.previous.version**.

7.4 Adjusting the Scope of a View

At any time during a development cycle, you may need to change the set of elements available to your view. For example, if you work in a snapshot view, you can reduce the amount of time required for deliver and rebase operations by narrowing your view's scope to include only the set of elements related to your activities. (The fewer elements in your snapshot view, the less time needed to update it.) Note, however, that to test your work during the deliver operation, your view must include any files required to satisfy build dependencies.

To adjust the scope of a view, you can do the following:

- Change which elements are loaded into a snapshot view
- Activate or deactivate VOBs for dynamic views

Adjusting the scope of a view does not affect the view's stream or any other views that are attached to the stream. For example, if a recommended baseline includes a new set of files and directories, rebasing your development stream makes the new files and directories available to your stream, but they may not be visible in your development view, depending on the view's scope.

To Change Which Elements Are Loaded into a Snapshot View

1. In the ClearCase Explorer Shortcut pane, click **Toolbox.** Then click **Base ClearCase>Edit View Properties**.

2. In the **Edit View Properties** dialog box, select the view and click **Edit**.

3. In the **Properties** dialog box, click the **Load Rules** tab.

4. On the **Load Rules** tab, click **Edit Load Rules**.

5. In the **Choose Elements to Load** dialog box, click **Add** to load another element into the view, or click **Remove** to unload a file or directory from the view.

6. Click **OK**.

Activating or Deactivating VOBs for Dynamic Views

Activating a VOB on your computer makes its files and directories available to your dynamic views. Deactivating a VOB frees your computer's mount points.

To Activate VOBs

NOTE: Usually, ClearCase mounts all VOBs that contain versions of elements in a project baseline when you start a dynamic view attached to a project stream. However, you may need to follow this procedure if you deactivate a VOB and then want to mount it again.

1. In the system taskbar, click **Start**. Then click **Programs>ClearCase Administration>Mount VOB**.
2. In the **Mount VOB** dialog box, select the VOBs containing your source files.
3. Select **Reconnect at Logon** to activate the VOBs when you log on.
4. Click **OK**.

To Deactivate VOBs

1. In the system taskbar, click **Start**. Then click **Programs>ClearCase Administration>Unmount VOB**.
2. In the **Unmount** dialog box, select one or more VOBs.
3. Click **Unmount**.

7.5 Assigning Snapshot Views to Drive Letters

If your makefiles or other files require absolute pathnames with a specific drive letter, assign your view to a drive letter.

Depending on your computer's configuration, two methods are available:

- Make the snapshot view a shared directory and then assign it to a drive letter.
- Use the **subst** command.

Assign Only the Root Directory. You must assign only the snapshot view's root directory to a drive letter. The root directory of the snapshot view contains a hidden file named **view.dat**. ClearCase searches for this file to determine whether the current directory is in a snapshot view.

If you assign a directory below the view's root to a drive letter, ClearCase cannot find the view's **view.dat** file and assumes that the current directory is not a snapshot view.

Creating a Shared Directory and Assigning It to a Drive Letter

From Windows NT hosts, you can create a shared directory and assign it to a drive letter. With this method, you can access a view through Network Neighborhood, but the performance is slightly slower than if you use the **subst** command to assign the view to a drive letter.

From Windows 95 or Windows 98 hosts, you can assign a shared directory to a drive letter only if the directory is on some other host; for example, you can use this method to assign someone else's view to a drive letter. To assign your own Windows 95 or Windows 98 view to a drive letter, you must use the **subst** command.

To Create a Shared Directory

NOTE: If you want to share a directory on a Windows 95 or Windows 98 computer (for example, so team members can assign your view to a drive letter on their computers), you must enable file sharing before using this procedure.

1. In Windows Explorer, click the snapshot view's root directory; then click **File>Sharing**.
2. On the **Sharing** tab, click **Shared As**.
3. For the sake of consistency, accept the default value in the **Share Name** box. (Make the share name the same as the leaf name of the snapshot view's root directory.)
4. Click **OK**.

To Assign the Root Directory to a Drive Letter

NOTE: If you prefer to use the command line, you can use the **net use** command instead of the following procedure.

1. In Network Neighborhood, navigate to the view's root directory.
2. Right-click the folder; on the shortcut menu, select **Map Network Drive**.
3. In the **Map Network Drive** dialog box, select a drive letter from the **Drive** box.
4. Click **OK**.

Using the subst Command

Assigning a view to a drive letter with the **subst** command provides slightly better performance than making the snapshot view a shared directory; however, only shared directories are accessible through Network Neighborhood.

1. Open a command shell.
2. Enter **subst** *driveletter*: *view's-path*. For example, the command

   subst Y: C:\library\pat_v1.4_cropcircle

   maps the **pat_v1.4_cropcircle** directory to the **Y:** drive. Assign only the view's root directory to a drive letter.

For more information on the **subst** command, type **help subst** in a command shell.

7.6 Moving Work from a Development Stream to Another Project

Under normal circumstances, project managers use baselines (which reside on a project's integration stream) to share work between projects. But if you start work on an activity in your development stream and then determine that you cannot complete the work for the current project, we recommend that you move the work to the project in which you intend to complete it. To facilitate the merge process, move the work to the new project as soon as you determine that you cannot complete it in the original project. (The longer you wait to merge the work, the less you may remember about it and the greater the potential for merge conflicts.)

We recommend against working in a development stream that contains activities you do not intend to complete for the current project because it may prevent the rebase operation from synchronizing your development stream with the work in the baseline.

To Move Work from a Development Stream to Another Project

1. Get the activity's *activity-selector*:

   a. Find the activity. For example, use **MyActivities** in ClearCase Explorer.

   b. Right-click the activity and on the shortcut menu, click **Properties**.

   On the **General** tab of the **Properties** dialog box, the **ID** box displays the activity's *activity-selector*.

   c. Write down the activity-selector or copy it to the clipboard.

   d. Close the Properties Browser.

2. Join the project in which you intend to complete the work. (See Chapter 2, *Setting Up Work Areas*.)

3. Decide whether to add the work to the new project's integration stream or to your development stream in the new project.

4. From your view in the new project, create an activity to identify the work you are moving from the original project. Then set the view to the activity. (See *Creating Activities* on page 46 and *Setting Activities* on page 49.)

5. Open a command prompt and change to your view in the project.

6. Use the **cleartool findmerge** command with the following syntax:

   cleartool findmerge *activity-selector* **–fcsets –merge –gmerge**

   Supply the activity-selector for the activity from which you want to move versions, that is, the activity-selector from Step #c.

   For example:

   cleartool findmerge change_copyright_notice@\projects\cropcircle
   ^ –fcsets fl–merge –gmerge

7. If ClearCase encounters merge conflicts, it starts Diff Merge. In Diff Merge, resolve the conflict, save your merge results, and exit.

8. In your development stream for the original project, complete work for and deliver all other activities.

9. Discontinue work in your original development stream and create another one for the original project. If your organization's policies require you to work in only one development stream per project, consult your project manager.

   It is possible, but not recommended, to remove the versions in the original activity (using **cleartool rmver –hlink**) and then to remove the activity using **cleartool rmactivity**. However, removing versions effectively erases your organization's development history and should be done only when necessary and only when you and your project manager are certain that removing the work does not affect other work. By removing the versions and *merge arrows* (with the **–hlink** option), ClearCase has no record of the merge (although the element's history remains intact).

7.7 Registering a Snapshot View

When you create a snapshot view or click on a view shortcut in ClearCase Explorer, ClearCase registers it in your Windows User Profile. The integration with Windows Explorer recognizes files and directories below a registered snapshot view as ClearCase objects.

If a view is not registered in your Windows User Profile and you access the view through Network Neighborhood, the Windows Explorer integration does not recognize the files or directories as ClearCase objects. If you want to perform ClearCase operations in this view, you must either create a view shortcut in ClearCase Explorer or, if you want to access the view only through Windows Explorer, register the view.

NOTE: You cannot register a snapshot view that was created from a UNIX host.

For example, if you use Windows Explorer to navigate to a team member's view on a different computer and right-click a file in the view, ClearCase options are not available. To perform ClearCase operations on the files in a team member's view, register the view as follows:

1. In Windows Explorer, navigate to the root directory of the view you want to register.

2. Right-click the root directory of the view.

   When you right-click the root directory of the view, ClearCase registers the view by adding an entry to your Windows User Profile.

7.8 Moving Views

This section discusses the following tasks:

- Changing the physical location of a snapshot view's directory tree
- Moving a view storage directory

For information on changing a view-tag, see the **mktag** reference page in *ClearCase Reference Manual*.

Changing the Physical Location of a Snapshot View

If the snapshot view storage directory is in a storage location, you can use standard Windows commands (such as **Cut** and **Paste** commands or drag and drop operations in Windows Explorer) to move the snapshot view's directory tree of loaded elements and view-private files. You can move the view to a different computer, but the computer must run a Windows operating system.

CAUTION: If the view storage directory is located below the root directory of the view, **do not use** standard Windows commands to move the snapshot view. Instead, see *Moving a View Storage Directory*.

To Find the Location of the View Storage Directory

1. In ClearCase Explorer, right-click the view's root directory.

2. In the **View Properties** dialog box, click **Advanced**.

   The **Host Path** box displays the pathname for the view storage directory.

Update After Moving

After moving a snapshot view, you must update any view shortcuts in ClearCase Explorer that use the old pathname to the view. If you did not create a ClearCase Explorer shortcut for the view and you move a view to a different computer, you must register its new location. Until you register its new location, the ClearCase user interface continues to present the old location.

Moving a View Storage Directory

Each dynamic view and snapshot view includes a view storage directory, which ClearCase uses to maintain the view. **Do not use** standard Windows commands (such as **Cut** and **Paste** commands or drag and drop operations) to move a view storage directory for the following reasons:

- The view storage directory includes a database. Moving the database without first shutting down the view's **view_server** process can corrupt the database.
- ClearCase stores the location of view storage directories in its own set of registries. The information in these registries must be correct for you to perform ClearCase operations in your views. In a dynamic view, the location in ClearCase registries must be correct for you to access any file or directory in the view.

We suggest that you ask your ClearCase administrator to move view storage directories because it may affect other, potentially many other, ClearCase users at your site. *Administering ClearCase* describes the procedure for moving view storage directories.

CAUTION: You will lose data (including view-private files in a dynamic view) if you move a view storage directory without following the procedure described in *Administering ClearCase*.

7.9 Accessing Views and VOBs Across Platform Types

ClearCase supports environments in which some ClearCase hosts use a Microsoft Windows operating system and others use a UNIX operating system.

This section discusses the following topics:

- Creating views across platform types
- Accessing VOBs across platform types
- Developing software across platform types

Creating Views Across Platform Types

Your administrator can set up storage locations on Windows and UNIX server hosts. Any snapshot view that you create can use one of these storage locations, regardless of the platform type of the server host. For more information about storage locations, see the **mkstgloc** reference page in *ClearCase Reference Manual*.

For a dynamic view, the view storage directory must be located on a host of the same platform type as the host from which you create the view: if you create a dynamic view from a UNIX host, you must locate the view storage directory on a ClearCase host on UNIX; if you create a dynamic view from a Windows host, you must locate the view storage directory on a Windows NT host that is set up to store view storage directories. We recommend that you locate dynamic view storage directories on the host from which you most often use the view.

Snapshot View Characteristics and Operating-System Type

For snapshot views, the operating system type from which you create the view determines view characteristics; the operating system type that hosts the files and processes related to a snapshot view do not affect the view's behavior.

For example, it is possible to create a snapshot view from a Windows host and locate the view directory tree and the view storage directory on a ClearCase host on UNIX (assuming that you use third-party software to access UNIX file systems from Windows computers). Even though all files related to the view are on a UNIX workstation, because you created the view from a Windows host, the view behaves as if its files are located on a Windows computer: it does not create symbolic links if the load rules encounter a VOB symbolic link, and you can issue ClearCase commands for the view only from Windows hosts (ClearCase hosts on UNIX will not recognize the directory tree as a snapshot view).

Accessing Views Across Platform Types

Accessing UNIX Snapshot Views from Windows Hosts

ClearCase supports a set of third-party products for accessing UNIX file systems from Windows computers. If your organization uses one of these products, you can access UNIX snapshot views from Windows Explorer (or a command prompt) just as you would access any other directory tree on a UNIX workstation.

You can access snapshot views across platforms, but you cannot issue ClearCase commands across platforms. For example, you cannot check out files in UNIX snapshot views from Windows hosts nor can you create shortcuts to UNIX snapshot views from ClearCase Explorer.

If, from a Windows host, you hijack a file in a UNIX snapshot view, ClearCase detects the hijack when you update the view from a ClearCase host on UNIX.

Accessing Windows Snapshot Views from UNIX Hosts

ClearCase does not support accessing Windows file systems from UNIX workstations.

Accessing UNIX Dynamic Views from Windows Hosts

ClearCase supports a set of third-party products for accessing UNIX file systems from Windows computers. If your organization uses one of these products, you can complete the following tasks to access UNIX dynamic views from Windows computers:

1. Create the UNIX view with the proper text mode. For more information, see *Developing Software Across Platform Types*.
2. Import the UNIX view's view-tag into your Windows network region.
3. Start the dynamic view or add a shortcut to the view in ClearCase Explorer.

Accessing Windows Dynamic Views from UNIX Hosts

ClearCase does not support products for accessing Windows file systems from UNIX workstations. You cannot access Windows views from UNIX hosts.

Accessing VOBs Across Platform Types

Your administrator sets up VOBs on Windows or UNIX hosts and creates *VOB-tags* in each ClearCase network region that needs to access the VOBs. (See *Administering ClearCase* for information on registering UNIX VOB-tags in a Windows network region.) Then, from any ClearCase host on Windows or UNIX you can create snapshot views to load elements from VOBs that have tags in your network region.

From a ClearCase host on Windows that supports dynamic views, you can access VOBs on Windows and UNIX from dynamic views as well as snapshot views. To access VOBs on UNIX from Windows dynamic views, you must use third-party software that provides access to UNIX file systems from Windows computers. From a ClearCase host on UNIX, you cannot access VOBs on Windows from dynamic views. Table 2 summarizes your options for accessing VOBs across platform types.

Table 2 Accessing ClearCase VOBs Across Platform Types

Platform Type of Your ClearCase Host	**Platform Type on Which VOB Is Located**	**View from Which You Can Access Source Files**
Windows computer	Windows computer or UNIX workstation	Snapshot view or dynamic view
UNIX workstation	Windows computer	Snapshot view
UNIX workstation	UNIX workstation	Snapshot view or dynamic view

Developing Software Across Platform Types

If developers check in source files from views created on both Windows and UNIX hosts, consider creating your views in insert_cr or strip_cr text mode. The text modes change how a view manages line terminator sequences. For more information about view text modes, refer to *Administering ClearCase* or ClearCase online help.

7.10 Regenerating the view.dat File

The root directory of a snapshot view contains a hidden file, **view.dat**. If you delete this file inadvertently, ClearCase no longer identifies the view as a ClearCase object, and you can no longer perform ClearCase operations on files or directories loaded in the view.

To Regenerate the view.dat File

1. Open a command shell.

2. Type this command:

   > **ccperl** *ccase-home-dir***etc\utils\regen_view_dot_dat.pl**
   ^ [**–tag** *snapshot-view-tag*] *snapshot-view-pathname*

   For example:

   > **ccperl c:\atria\etc\utils regen_view_dot_dat.pl**
   ^ –tag pat_v1.4_cropcircle
   ^ c:\pat_v1.4_cropcircle

   If the view storage directory is under the root directory of the view, you do not need to use the **–tag** *snapshot-view-tag* argument.

A Working in a Snapshot View While Disconnected from the Network

If you need to work with your source files from a computer that is disconnected from the network of ClearCase hosts and servers, you can set up a snapshot view for disconnected use.

This chapter describes the following tasks:

- ➤ Setting up a view for your hardware configuration
- ➤ Preparing the view
- ➤ Disconnecting the view
- ➤ Working in the view
- ➤ Re-connecting to the network
- ➤ Using the Update Tool

NOTE: While disconnected from the network, you cannot access ClearCase information about the files in your view or issue most ClearCase commands. If you want to work from a remote location and continue to access ClearCase information and issue ClearCase commands, consider using the ClearCase Web interface. Talk to your ClearCase administrator to see whether the ClearCase Web interface has been configured at your site and what URL you need to supply to your Web browser to access it. For information about using the Web interface, see the Web interface online help.

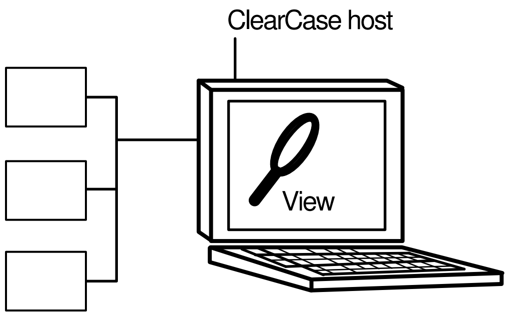
A.1 Setting Up a View for Your Hardware Configuration

You can use one of several hardware configurations to work in a snapshot view that is disconnected from the network.

This chapter describes the following recommended configurations:

- ➤ Creating and using the view on a laptop computer that periodically connects to the network (Figure 37).

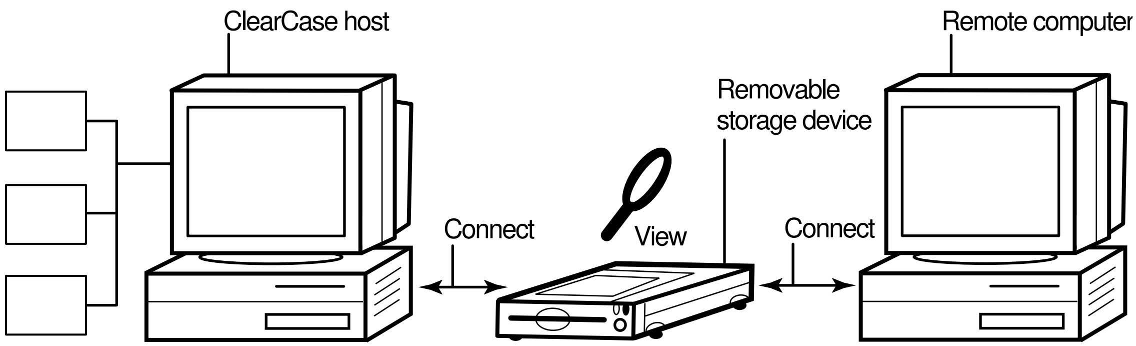
Figure 37 View on a Laptop



NOTE: The laptop computer must run a Windows operating system.

- ➤ Creating and using the view on a removable storage device such as an external hard drive or some other device (such as a Jaz™ drive) that provides satisfactory read/write performance (Figure 38).

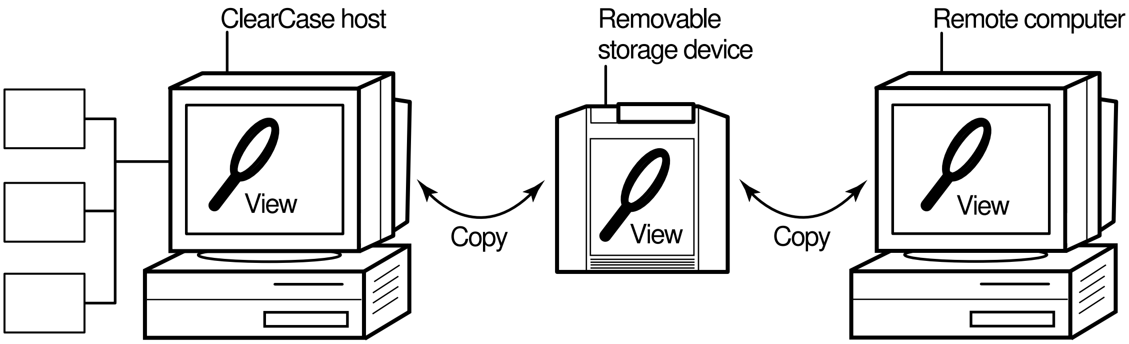
Figure 38 View On a Removable Storage Device



NOTE: The remote computer must run a Windows operating system.

➤ Copying the view from a ClearCase host to a temporary, removable storage device such as a diskette or a tape drive, which usually does not provide satisfactory read/write performance, and then copying the view from the storage device to a computer that is disconnected from the network (Figure 39).

Figure 39 Copy the View



NOTE: The remote computer must run a Windows operating system.

Under the Hood: Location of the View Storage Directory in Disconnected-Use Configurations

In all the configurations recommended for disconnected use, the snapshot view storage directory is in a server storage location. We recommend (and in the case of a removable storage device, we enforce) this configuration because a view's **view_server** process runs on the host that contains the view storage directory. A **view_server** is a long-lived process that manages activity for a specific view. If the view storage directory is in the root directory of the snapshot view and you disconnect the view from the network while the **view_server** process is writing to the storage directory, you can corrupt the data ClearCase uses to maintain your view.

A.2 Preparing the View

Before you disconnect the view from the network, complete these tasks:

- Update the view to establish a checkpoint. (For information on updating the view, see *Updating the View* on page 129.)
- Check out the files you expect to modify. After you're disconnected from the network, you cannot check out files, although there are workarounds. (See *Hijacking a File* on page 124.)

When you are no longer connected to the network you cannot use most ClearCase commands. At this point, the disconnected computer does not distinguish a snapshot view directory from any other directory in the file system.

A.3 Disconnecting the View

How you disconnect the view depends on which hardware configuration you use. This section describes disconnecting the view for the following hardware configurations:

- View on a laptop
- View on a removable storage device
- Copying the view

Hardware Configuration: View on a Laptop

If the view is located on a laptop, you must deactivate the ClearCase integrations with Windows Explorer and other developer tools before you disconnect the view. Most ClearCase operations communicate with various ClearCase servers. Even if you install view server software on your host, almost any time you start a ClearCase operation (even an operation as simple as displaying the ClearCase shortcut menu from Windows Explorer) your host must connect to the ClearCase license server. The ClearCase integration with Microsoft Visual Studio attempts to connect to the license server periodically. If you are disconnected from the network, the application that attempts to connect to the ClearCase license server locks for several seconds until it returns an error message. Although the application doesn't fail, such interruptions are unnecessary.

To Deactivate ClearCase Integrations

1. In Control Panel, click **ClearCase**.

2. Click the **Options** tab and clear the **Enable ClearCase network operations** check box.

3. Disconnect the laptop from the network.

Deactivating the integrations removes ClearCase commands from the Windows Explorer shortcut menus.

Hardware Configuration: View on a Removable Storage Device

If the view is located on a removable storage device, disconnect the removable storage device from the network computer and reconnect the media to the remote computer.

Hardware Configuration: View Copied to a Storage Device

If you do not have a storage device with satisfactory read/write performance, use the **xcopy** command to copy files from your view to the storage media and from the storage media to the remote computer as follows:

xcopy *path**snapshot-view-directory* *destination* [**/D** [**:***m-d-y*] | **/M**] **/S /K /R**

For example, **xcopy c:\Library\pat_v1.4_cropcircle e:\temp /D /S /K /R**

The command options accomplish the following:

➤ The **/D** option instructs **xcopy** to copy only the files that have changed, which is useful if you frequently alternate between the network computer and the nonnetwork computer. Use **/D** if you configured the Update Tool to set the time stamp of updated files to the time at which the files are loaded into the view. You can view or change the Update Tool's time stamp option on the **Advanced** tab of the **Start Update** dialog box. For more information, refer to Step #3 of *Updating the View* on page 129.

  If you set the Update Tool to create time stamps based on the version-creation time, consider using the **/M** option, which causes **xcopy** to copy the files with the archive attribute set. The **/M** option is less reliable than the **/D** option, because some applications alter the archive attribute without writing the file.

- The **/S** option instructs **xcopy** to copy all subdirectories of the snapshot view.
- The **/K** option keeps the read-only file attribute as established by ClearCase.
- The **/R** option allows **xcopy** to overwrite read-only files.

A.4 Working in the View

You cannot use most ClearCase commands when disconnected from the network. Yet you may need to work on files that you did not check out or locate files you have modified. This section provides workarounds for these ClearCase operations.

Hijacking a File

If you need to modify a loaded file element that you have not checked out, you can *hijack* the file. ClearCase considers a file hijacked when you modify it without checking it out. For more information, see *Under the Hood: How ClearCase Determines Whether a File is Hijacked* on page 129.

When you reconnect to the network, you use the Update Tool to find the files you hijacked. You can do the following with a hijacked file:

- Check out the file. You can then continue to modify it and, when you're ready, check in your changes.
- Undo the hijack. For more information, see *Undoing a Hijack* on page 128.

To Hijack a File

1. In Windows Explorer, right-click the file you want to hijack.
2. On the shortcut menu, click **Properties** to display the file's property sheet.
3. On the **General** tab, clear the **Read-Only** check box.

Finding Modified Files While Disconnected

You can use Windows Explorer to find all files that have been modified after a specified date.

1. In Windows Explorer, click **Tools>Find>Files or Folders**.
2. In the **Find: All Files** dialog box, type the path to the view or to a specific directory in the view in the **Look In** box.
3. Use the **Date Modified** tab to define your search criteria.
4. Click **Find Now**.

   The **Find: All Files** dialog box expands to display the files and directories it finds.

A.5 Connecting to the Network

When you return to the office and can access the network directly, do one of the following, depending on your hardware configuration.

Hardware Configuration: View on a Laptop

If you're using the view on a laptop, connect the laptop to the network. If ClearCase is installed on the laptop, activate the ClearCase integrations as follows:

1. In Control Panel, click **ClearCase**.
2. Click the **Options** tab and select the **Enable ClearCase network operations** check box.

Hardware Configuration: View on a Removable Storage Device

If you're using the view on a removable storage device, connect the removable storage device to the computer on the network.

Hardware Configuration: View Copied to a Storage Device

If you copied the view onto removable media, use **xcopy** with the [**/D** | **/M**] **/S /K /R** options to copy files back to the original location on the network computer.

A.6 Using the Update Tool

When you're connected to the network, use the Update Tool for the following tasks:

- ➤ Determine how to handle hijacked files
- ➤ Update the view

Determining How to Handle Hijacked Files

Handling hijacked files involves the following tasks:

- ➤ Finding hijacked files
- ➤ Comparing a hijacked file to the version in the VOB
- ➤ Checking out a hijacked file
- ➤ Undoing a hijack
- ➤ Choosing other ways to handle hijacked files

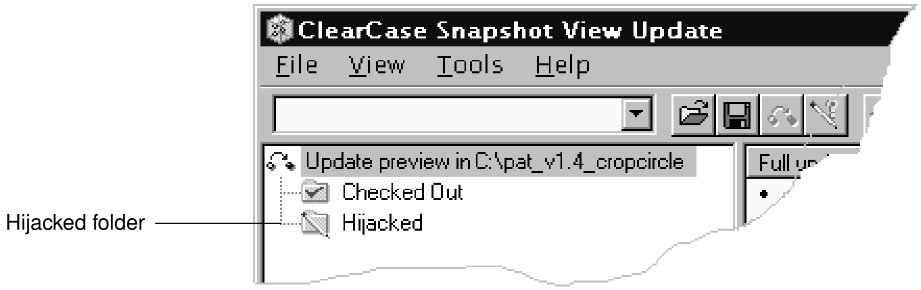
To Find Hijacked Files

1. In ClearCase Explorer, right-click the root directory of the snapshot view.

2. On the shortcut menu, click **Find Modified Files**.

3. If any hijacked files are in your view, the ClearCase Snapshot View Update window displays a folder in the left pane titled **Hijacked** (Figure 40).

Figure 40 Hijacked Files in the Results Window



To Compare a Hijacked File to the Version in the VOB

You can use the Diff Merge tool to see how the hijacked file differs from the checked-in version of the file:

1. In the right pane of the ClearCase Snapshot View Update window, right-click a hijacked file.
2. On the shortcut menu, click **Compare with Original Version**. For information on using the Diff Merge tool, see the online help.

Checking Out a Hijacked File

To keep the modifications in a hijacked file, check out the file:

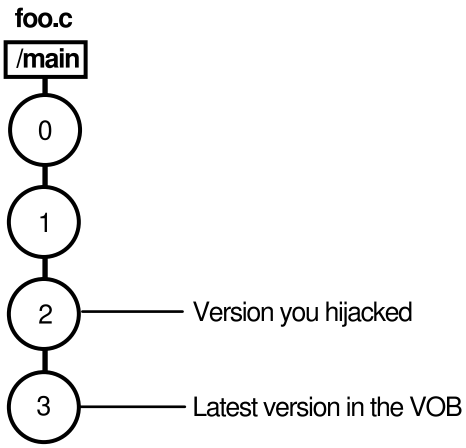
1. In the right pane of the ClearCase Snapshot View Update window, right-click a hijacked file.
2. On the shortcut menu, click **Check Out**.
3. ClearCase treats a checked-out hijacked file as it does any other checkout.

When you're ready, you can check in the file or, if necessary, merge your changes with a version in the VOB.

You May Be Prompted to Merge

If you're working with a shared set of versions and someone has checked in a newer version of the file while it was hijacked in your view (Figure 41), you have to merge the hijacked file with the newer version in the VOB at checkout.

Figure 41 Hijacked Version May Not Be the Latest Version



To Merge with the Latest Version

When you issue the **Check In** command for a nonlatest version, ClearCase opens the dialog box to ask whether you want to merge the file now. If you choose to merge, ClearCase attempts to merge automatically, starting the Diff Merge tool if it needs your input to complete the merge. For information on using Diff Merge, refer to ClearCase online help. After the merge, ClearCase prompts you to check in the file.

Undoing a Hijack

If, for specific hijacked files, you want to discard your changes and get a fresh copy of the version from the VOB, you can undo the hijack.

1. In the right pane of the ClearCase Snapshot View Update window, select one or more hijacked files.

2. Right-click the selected files, and on the shortcut menu, click **Undo Hijack**.

ClearCase overwrites the hijacked file with the version that was loaded in the view. If you want to overwrite hijacked files with the versions the config spec selects in the VOB, refer to Step #3 in *Updating the View* on page 129.

Under the Hood: How ClearCase Determines Whether a File is Hijacked

To keep track of file modifications in a snapshot view, ClearCase stores a loaded file's size and last-modified time stamp (as reported by the Windows file system). ClearCase updates these values each time you check out a file, check in a file, or load a new version into the view.

To determine whether a file is hijacked, ClearCase compares the current size and last-modified time stamp of a non-checked-out file with the size and time stamp recorded in the view database. If either value is different from the value in the view database, ClearCase considers the file hijacked.

Changing a non-checked-out file's read-only attribute alone does not necessarily mean ClearCase considers the file hijacked.

Other Ways to Handle Hijacked Files

While updating the view, you can handle hijacked files in any of the following ways:

- Leave hijacked files in place
- Rename the hijacked files and load the version from the VOB
- Overwrite hijacked files with the version the config selects in the VOB

See *Updating the View* for more information.

Updating the View

1. In ClearCase Explorer, select the snapshot view's root directory.

2. Right-click to display the shortcut menu and click **Update View**.

3. To configure the Update Tool for handling hijacked files, in the **Start Update** dialog box click the **Advanced** tab and select a method for handling the remaining hijacked files. You have three choices:

   - Leave hijacked files in place
   - Rename the hijacked files and load the version from the VOB
   - Overwrite hijacked files with the version the config spec selects in the VOB

4. To start the update, click **OK**.

Index

D

E