MICRO FOCUS®

PVCS Version Manager
Getting Started Guide

Table of Contents

Chapter 1

Getting Started with the Version Manager Desktop Client

Starting Version Manager

Once you install Version Manager, you are ready to begin using the product. The installation process automatically configures your workstation for your specific platform.

Starting the Desktop Client

The first time you start Version Manager, the Welcome to Version Manager dialog box appears, which provides options for you to:

- Create a new project database, unless this feature has been restricted.

- Open the sample project database.

- Open the project root that was last opened in Version Manager.

The next time you start Version Manager, the Welcome to Version Manager dialog box provides options for you to open the previously opened project database(s).

The Welcome to Version Manager dialog box will not show the next time you start Version Manager if you select the **Don't show this dialog again** check box.

To start the desktop client:

1   Do one of the following:

- **On Windows:** From the Start menu, select Programs | Serena | Version Manager *X.X.X* | Version Manager.

- **On UNIX:** Navigate to the /usr/serena/vm/<*os*>/bin directory and run pvcsvmux.

    The -d option of pvcsvmux writes debug information to the console.

    Version Manager starts and the Welcome to Version Manager dialog box opens.

2   Verify that the **Open the sample project database** check box is selected.

    Optionally, to create a new project database, select the **Create a new project database** check box. Version Manager creates a project database using default configuration values. You can add your workfiles to it as a quick way to start using Version Manager, or you can use the new project database along with the sample project database to practice Version Manager procedures. The creation of a new project database may be restricted by the administrator.

3   Click **OK**.

The Login dialog box appears.
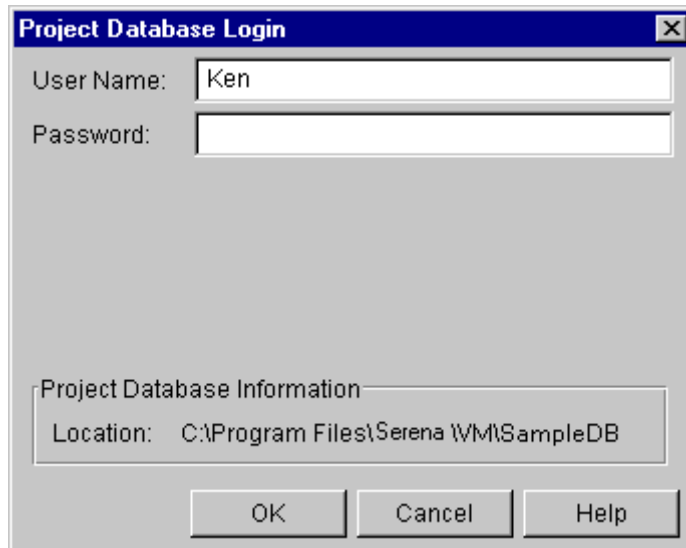
4 Log in to the sample project database by entering Admin as the User Name, leaving the password blank, and clicking **OK**.

Version Manager opens and displays the project databases, depending on the options you selected in Step 2, in the Project pane below.
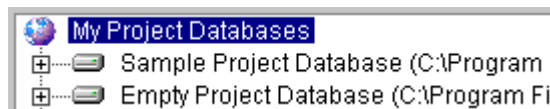
5 Expand the contents of the sample project database by doing one of the following:

- Click the + signs.

—*or*—

- Highlight the project database and press the right arrow key.
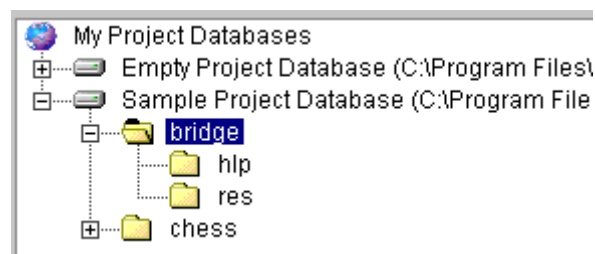
The sample project database contains sample files that you can work with in the section,

Exploring the Version Manager Desktop

Now that you have started Version Manager and opened the sample project database, you can use the sample files to explore the desktop. This section identifies and describes some of the key features that comprise the interface.

Tool Bar

The tool bar provides shortcut buttons for performing commonly used functions, such as check in, check out, lock, and unlock. Administrators can add buttons to the toolbar that invoke other applications. Tool bar information is stored in a tool bar configuration file, which administrators associate with project databases. When a tool bar configuration file is associated with a project database, the tool bar loads automatically whenever the project database is opened or selected. For more information, refer to "Adding Custom Tools" in the *Administrator's Guide*.

The default tool bar buttons include the following:

Click this button...	To do this...
	Create a new project.
	Add workfiles to project or project database.
	Set the workfile location for the currently selected item.
	Check in a workfile.
	Check out a revision.
	Get a revision.
	Unlock a revision.
	Lock a revision.
	Assign a version label.
	Promote to next promotion group.
	Show history report for item.
	Show differences for item.
	Toggle (show or hide) revision pane.
	Launch Tracker.

Click this button...	To do this...
	Indicates that the File pane is displaying the versioned files, projects, and subprojects contained within the selected project or project database.
	Indicates that the File pane is displaying only the versioned files within the selected project, subproject, or project database.

Status Bar

The status bar is located at the bottom of the window, as shown below.
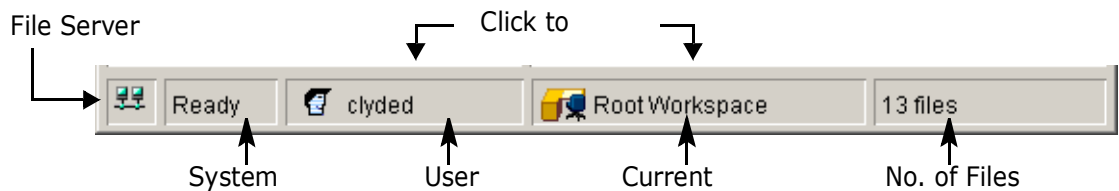
The status bar contains the following information and shortcuts:

- File Server Status—indicates the status of the Version Manager File Server (click to refresh status and display more information):

    - No file server is configured.

    - One or more configured file servers are not responding.

    - All configured file servers are working correctly.

- System Message – Either displays information about the current status of Version Manager, such as "Ready" or "Loading...", or a message specific to the current action, such as "Checking In".

- User ID – Displays the name of the user logged in to the selected project database. Click this field to access the Switch User dialog box.

- Workspace – Displays the name of the current workspace set on the selected project database. There are two types of workspaces, public and private. Generally, a public workspace is created by the administrator and is indicated by a desk icon. A private workspace is generally created by an individual Project Team Member and is indicated by a desk icon with a figure. Click this field to access the Set Workspace dialog box.

- Files Selected – Displays the number of files selected. When you select a project, the total number of files in the project is displayed. When you select one or more files, the total number of files selected is displayed.

Standard Menus and Context Menus

Two types of menus are available in the Version Manager desktop client:

- Standard menus, which are available from the menu bar.

- Context menus, which appear when you right-click an item.

From the standard menus of the menu bar, you can access all Version Manager features and functions, including all tasks and the online help.

Optionally, your administrator can configure a custom Tools menu that you can access from the menu bar. From this Tools menu, you can access the applications your administrator has configured for the project database. If the administrator has not configured any user tools for the project database, the Tools menu is not available.

Quick access to common tasks

From context menus, you can access the most common tasks performed on the item you have selected.

Standard Menu	
Get...	Ctrl+G
Check Out...	Ctrl+O
Check In...	Ctrl+I
Unlock...	
Lock...	
Version Label	▶
Promotion Group	▶
Show History...	Ctrl+H
Show Differences...	Ctrl+D
Show Merge...	Ctrl+M
Show Journal...	Ctrl+J
Delete Revision...	

Example of the Actions menu when a revision is selected.

Context Menu
View File...
Edit File...
Get...
Check Out...
Check In...
Unlock...
Lock...
Assign Version Label...
Promote To Next Group...
Show History...
Delete
Properties
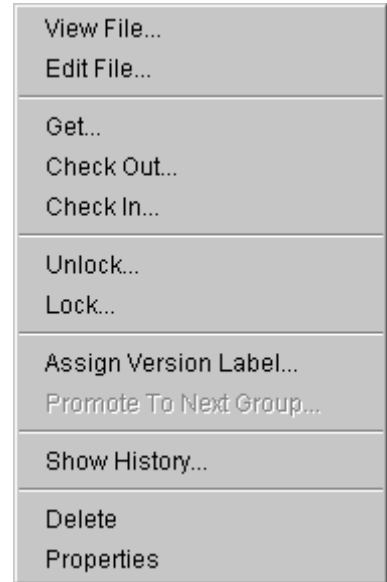
Example of the context menu when a revision is selected.

Functions that are grayed out in standard menus and context menus are unavailable because the function:

■ Is not applicable to the item you currently have selected

■ Has been disallowed by your administrator

Header Bar

The header bar contains the following three columns:

■ All Projects – Contains the Project pane. You can access open project databases and the projects and subprojects they contain from the Project pane.

■ Contents of – Displays the contents of the item selected in the Project pane. If you set a filter, the name of this field changes to "Filtered Contents of".

■ Workfile Location – Displays the workfile location defined for the selected project database, project, subproject.

Project Pane

The Project pane displays open project databases, projects, and subprojects.

Inherit settings from project databases

A project database is a named, hierarchical collection of projects, subprojects, and versioned files. Project databases store configuration and access-control database information, which affects how you work with the projects within it.

Project databases are represented by the following icons:

This icon...	Represents...
	An open project database. When you open a project database, the name and location of the project database are shown next to the icon. When you open a project database that was created using an earlier release of Version Manager, only the location of the project database is shown.
	A selected project database. When you select a project database, you can perform an action on the contents of the entire project database.

A project is a logical grouping of versioned files and subprojects. Projects inherit configuration settings from the project database; however, you can also apply configuration settings to individual projects. Projects are represented by the following icons:

This icon...	Represents...
	A project or subproject (yellow).
	A configured project (yellow). These configuration settings override configuration settings inherited from the project database.
	A project assigned an access control database that does not contain your user ID (yellow). You do not have privileges to access this project.
	A 5.3/6.0 project or folder (blue).
	A configured 5.3/6.0 project (blue). These configuration settings override configuration settings inherited from the master configuration file of the project root.
	A 5.3 project assigned an access control database that does not contain your user ID (blue). You do not have privileges to access this project.

From the Project pane, you can:

■ Expand and collapse the view of project databases, projects, and subprojects.

■ Right-click My Project Databases to access a context menu, which provides a quick way to create or open a project database.

■ Right-click a project database or project to access a context menu, which provides access to frequently used functions.

■ Resize the width or height of the Project pane.

■ Select multiple projects on which to perform an action.

- Drag and drop projects to other project databases or projects. Dragging and dropping projects enables you to copy projects and share the archives between project databases.

  • If you drag and drop a project within the same project database, Version Manager moves the project.

  • If you drag and drop a project to a different project database, Version Manager copies the project.

  • If you drag and drop a project within the same project database while holding down the CTRL key, Version Manager copies the project.

- Drag and drop Version Manager 5.3/6.0 projects into project databases to copy the projects into the project databases. Copying 5.3/6.0 projects into project databases upgrades the projects to the new format and enables you to take advantage of the functions not available in Version Manager 5.3/6.0.

File Pane

The File pane provides access to the versioned files contained within a selected project database, project, or subproject. A versioned file is a file placed under version control. The File pane contains the following fields:

This field...	Identifies...
Name	The name of the versioned file.
Date Checked In	The most recent date and time the versioned file had a revision checked in on the trunk. If no revisions exist, then the date and time the archive was created are displayed.
Locked By	The user(s) who holds a lock on the item.
Checked Out to Location	The path(s) the file was checked out to, if a revision is locked and checked out.

Files are represented by the following icons:

This icon...	Represents a...
📄	Versioned file that does not contain any locked revisions.
📄	Versioned file that contains a locked revision.
❌	Versioned file that has lost the association with its archive. This occurs when the physical location of an archive is moved. Importing the archive restores the association.

From the File pane, you can:

- Click a column header to sort the files by that column.

- Click a sort order button in a column header to reverse the sort order.

- Right-click a file to access a context menu, which provides quick access to frequently used functions.

- Double-click a file to view or edit a revision.

- Drag and drop files into other projects or project databases. Dragging and dropping files into another project within the current project database moves the files. Dragging and dropping files into a different project database copies the files.

- Resize the width of the columns displayed in the File pane.

- Scroll horizontally or vertically.

- Type a character to jump the display to the first file that starts with that character. (Click anywhere in the File pane first.)

- Filter the file display by file type (wild card), locks, version labels, or promotion groups using the Filter drop-down menu.

- Filter the file display nonrecursively by toggling the file filter button. A nonrecursive display lists only the versioned files that meet the filter criteria for the selected project in the Project pane; it does not show the versioned files in the project's subprojects. This filter is the default.

- Filter the file display recursively by toggling the File Filter icon. A recursive display lists all of the subprojects and projects in the File pane that are contained in the project selected in the Project pane, along with versioned files that meet the filter criteria.

Revision Pane

Use the Revision pane to view the revision details of a versioned file. To open the Revision pane, select View | Show Revisions or click the Show Revisions tool bar icon. The Revision pane is only displayed if a file is selected. If more than one file is selected, then no information is displayed in the Revision pane. If the Revision pane is showing, then the Hide Revisions option is displayed as a menu option.

Revision	Author	Date Checked In	Locked By	Change Description
1.0	Admin	5/18/98 03:37 PM	ken	Initial revision.

| **Revisions** | Version Labels | Promotion Groups |

Revisions tab

The Revision pane changes dynamically to reflect the information specific to the selected versioned file. Depending on the versioned file, the Revision pane can display up to three tabs:

- Revisions tab

- Version Labels tab

- Promotion Groups tab

The Revision pane always contains a Revisions and a Version Labels tab. If a promotion model is in effect, the Promotion Group tab is also displayed.

Revisions are represented by the following icons:

This icon...	Represents a...
	Revision
	Locked revision
	Revision assigned to a promotion group
	Locked revision assigned to a promotion group
	Revision assigned a version label
	Locked revision assigned a version label
	Locked revision assigned a version label and a promotion group
	Trunk (the main line of development)
	Branch (a separate line of development that diverges from the trunk or another branch)

From the Revision pane, you can:

- Click a branch to review information about the branch revisions.

- Click the Promotion Groups, Version Labels, or Revisions tab to view different types of information about the revision.

- Click a column header to sort the files by that column.

- Click a sort order button in a column header to reverse the sort order.

- Right-click a revision, version label, or promotion group to access a context menu, which provides quick access to frequently used functions.

- Resize the width of the fields displayed on each tab.

- Resize the width or height of the Revision pane.

- Scroll horizontally or vertically.

About the Revisions Tab

By default, the Revisions tab is at the front of the Revision pane. The Revisions tab lists all of the revisions associated with the selected versioned file. For each revision, the tab includes the revision number, the author of the revision, the date the revision was checked in, the users who have the revision locked, and the change description.

You can also tell whether a revision is part of a branch, belongs to a promotion group, is locked, or has a version label assigned to it by looking at the icon.

The Revisions tab contains the following fields:

This field...	Identifies the...
Revision	Number assigned to the revision.
Author	User ID of the author of the revision.
Date Checked In	Date and time the revision was checked in.
Locked By	User(s) holding a lock on the revision.
Change Description	Description of the modifications contained in the revision.

About the Version Labels Tab

Click the Version Labels tab to move it to the front.
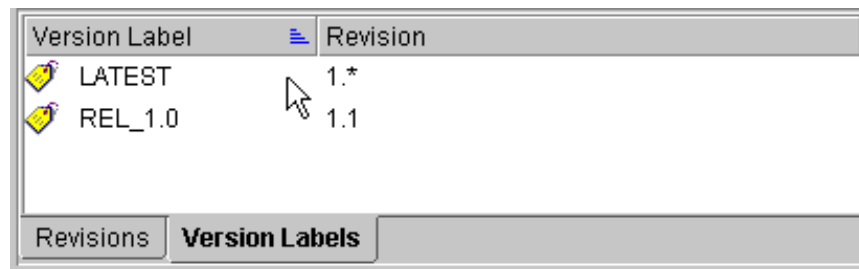
The Version Labels tab identifies the version label(s) assigned to each revision number. If no version labels have been assigned to the Versioned file, this display is blank.

About the Promotion Groups Tab

Promotion groups assigned to the versioned file can be reviewed in the Promotion Groups tab. If a promotion model has not been defined for the project or project database, this tab does not appear. Click the tab to move it to the front.
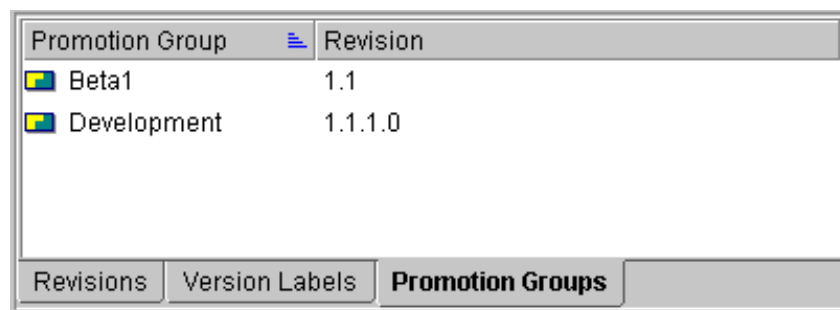
The Promotion Groups tab identifies the promotion group assigned to each revision number.

Reviewing Branches

If the selected versioned file contains a branch, the Revisions tab shows the details of the branch, as shown below.

Main line of
development

Branch

Revision	≡	Author	Date Checked In
⊟ 🔼 1.* (Trunk)			
📄 1.3		Admin	5/18/98 05:03 PM
📄 1.2		Admin	5/18/98 04:49 PM
📄 1.1		Admin	5/18/98 04:36 PM
📄 1.0		Admin	5/18/98 03:37 PM
⊞ 🔀 1.2.1.*			

Revisions	Version Labels
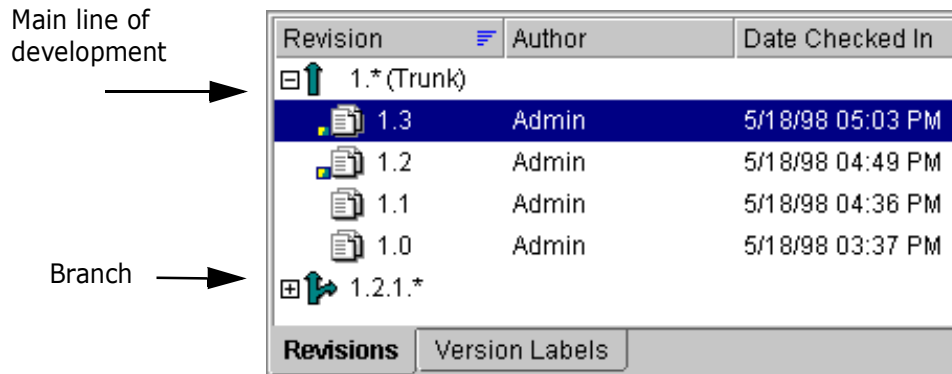
When a branch exists, the Revisions tab displays the revisions of the main line of development (trunk) and its branches. You can expand or collapse the view of the branch or main line of development by clicking the + or – signs. For more information about branching and merging, refer to the *Administrator's Guide*.

Filter

Perform actions on
groups of files

By defining a filter, you can modify the display of versioned files in the File pane and find the versioned files you need faster. By narrowing the file display to only those you need, you can perform actions, such as check out, on the entire group of versioned files by selecting all (CTRL+A) of the filtered files.

You can filter your view to display only versioned files that:

- Are locked by a specific user(s)

- Match a wildcard filename search

- Match a specified promotion group or version label

- Differ or match in terms of revision number, modification time, or revision contents between:

  • Two version labels

  • Two promotion groups

  • A version label and a promotion group

You can quickly set or define a filter using the filter drop-down menu located in the upper right corner of the desktop, as shown below.

Wild Card '.cpp'

mple Proji ... c:\Progra Current File Filter M\Samp
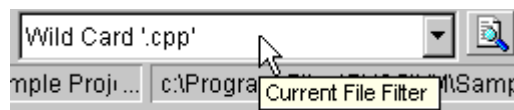
Version Manager saves the last four filters you've defined. You can access these saved filters from the View | Filter menu or the Filter drop-down menu.

 — Recursive filter button
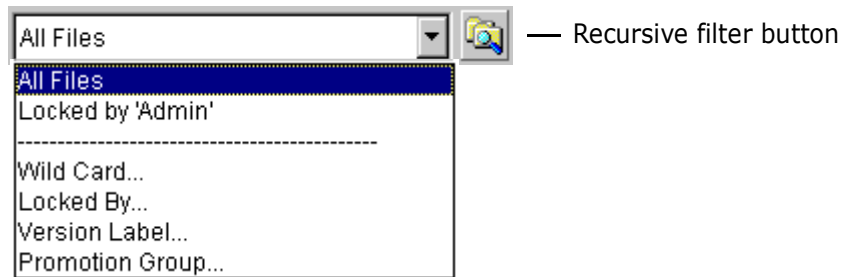
Filter recursively
You can also change the display by using a recursive filter, which filters the current project and its subprojects, as shown below.

To set a nonrecursive (not including projects) or recursive (including projects) filter, use the following tool bar buttons:

Click this button...	To do this...
	Filter only the current project, excluding subprojects.
	Filter the current project or project database, including all subprojects.

For more information about filtering the file display, refer to the *User's Guide*.

Sort

By default, files are sorted alphabetically by name. However, you can modify the sort order to display files according to the lock status, modification date, and check out location.

There are two ways you can sort the display of files:

- Select View | Sort, as shown below.

■ Click a column heading, such as Name, Date Checked In, Locked By, or Check Out to Location, to sort on that field.

Reverse the sort order to modify the display
You can also reverse the sort order by clicking the sort order button (▤). The reverse sort order inverts the display of files. For example, if you sort the files by Date Checked In, the files most recently modified appear at the bottom of the display. To display the files most recently modified at the top of the File pane, click the sort order button.

Sorted by date

Name	Date Checked In▤	Locked By	Ch
📁 /chess/client			
📄 DrawDlg.java	5/17/98 10:03 PM		
📄 chessviewer.html	5/18/98 03:46 AM		
📄 ChessViewer.jpr	5/18/98 03:48 AM		
📄 BoardOptions.java	5/18/98 03:51 AM		
📄 ChessRules.java	5/18/98 03:53 AM		

Sorted by date in reverse order

Name	Date Checked In▤	Locked By	Ch
📁 /chess/client			
📄 lst	11/23/99 04:04 PM		
📄 WannaPlay.java	5/18/98 04:07 AM		
📄 ServerMessage.ja...	5/18/98 04:07 AM		
📄 readme.html	5/18/98 04:07 AM		
📄 PlayerListListene ...	5/18/98 04:07 AM		

Shortcut and Mnemonic Keys

Shortcut and mnemonic keys provide quick methods of navigating through the desktop client interface. These keys provide access to functions listed on menus or dialog boxes.

When a function can be completed using shortcut keys, the keys are listed on the menu next to the function, as shown below.

Function   Shortcut keys

Get...	Ctrl+G
Check Out...	Ctrl+O
Check In...	Ctrl+I

To use shortcut keys, press and hold the CTRL key and press the corresponding shortcut key. Shortcut keys are available for the following functions.

Function:	**Shortcut Keys:**
Add Workfiles	CTRL+F
Check In	CTRL+I
Check Out	CTRL+O
Get	CTRL+G
Refresh	CTRL+R
Select All	CTRL+A
Show Differences	CTRL+D

Function:	Shortcut Keys:
Show History	CTRL+H
Show Merge	CTRL+M
View Properties	CTRL+ENTER
Set WorkFile Location	CTRL + W
Create Project	CTRL + P
Show Journal	CTRL + J

NOTE On Windows, press ALT to access the menu bar.

Mnemonic Keys

Mnemonic keys are available in most message boxes. Mnemonic keys allow you to use your keyboard to make basic selections, by pressing the underlined character, for example, Y for Yes, N for No or 'o' in Format.

Turning Off Dialog Boxes

Streamline your work process

To streamline your work process, you can turn off dialog boxes for certain actions, such as the Check Out dialog box, so that when you perform an action, Version Manager completes the task without prompting you for more information. Instead, Version Manager uses the values entered in the dialog box the last time the dialog box was used.

Chapter 2
Using Version Manager

Working with the Desktop Client

In this section, you will create a new project and work with the sample project database files using the desktop client.

About the Sample Project Database and Scenarios

By default, when you install Version Manager, the sample project database is installed to the following locations:

- **On UNIX:** /usr/serena/vm/common/sampledb

- **On Windows:** C:\Users\All Users\Serena\VM\SampleDB

If you modify the default Version Manager installation location, the sample project database is installed in a location relative to the vm directory. In this location, there are sample files for three games: Chess, Checkers, and Bridge. However, only the files in the Chess and Bridge projects have been archived. The Chess and Bridge projects are shown below.
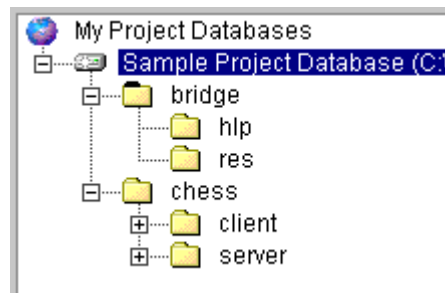
The Checkers files have not been archived—you will do this yourself in the following exercises. You will also perform some of the basic tasks that a Developer might perform while developing a new application.

While working through these exercises, assume the following:

- You work as a Developer for a large company that designs computer games.

- Your task is to begin work on a new game: Checkers.

- Your administrator has already set up a project database for the games under development.

- Your administrator assigned you a user name (Ken) that you'll need to use when logging in to the sample project database.

Using Default Settings

In each dialog box, Version Manager supplies a set of default options to help you perform the action faster and easier. The exercises in this chapter use these default settings. These default options are defined by the configuration settings applied to the project database. If your administrator modifies the default configuration settings, the options set in each dialog box match those set by your administrator.

For example, you could quickly check in a workfile by selecting the versioned file, selecting Actions | Check In, entering a description, and then clicking OK. If you (or your

administrator) have not changed any configuration settings, Version Manager automatically:

- Checks in the workfile from the location shown in the Check In From field and assigns the workfile a sequential revision number.

- Retains a read-only copy of the workfile.

- Does not delete the workfile or keep the revision locked after checking in the workfile.

- Checks in only the files of the selected project and does not include the files of the project's subprojects.

- Prompts you if the workfile is unchanged.

- Retains the promotion group currently assigned to the revision, if a promotion model is in effect.

- Does not assign a version label or create a branch.

Automatically Dismissing the Results Dialog Box

The Results dialog box appears after many tasks you perform, such as adding workfiles, checking in workfiles, and checking out revisions. The Results dialog box either verifies that the action was successful or, if the action failed, provides an error message. By default, you must click Close to dismiss the Results dialog box, as shown below. However, to speed your work, you can automatically dismiss the Results dialog box when an action is successful, so that you don't have to click Close.
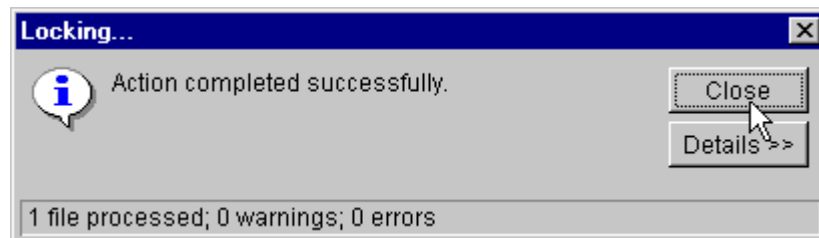
To automatically dismiss the Results dialog box, select the **Automatically dismiss result dialogs when action is successful** check box on the Dialog Behavior tab of the Options dialog box.

For more information about setting dialog box options, refer to the *User's Guide*.

Project Databases

A project database is a hierarchical representation of a group of projects, subprojects, and versioned files. A project database is not a relational database; instead, a project database stores the configuration settings for an entire collection of projects, subprojects, and versioned files.

Project database structure on an operating system

On your operating or file system, a project database is a collection of project files, a pvcsuser directory, project directories, a lib directory, and possibly an archives and a work directory. You can change the default locations of the archive and work directories when you create the project database.

Project database
structure within
the desktop client

From within Version Manager, a project database contains projects and subprojects, as shown below. They also contain versioned files, which appear in the File pane (not shown).

Project Database

Project

Subproject
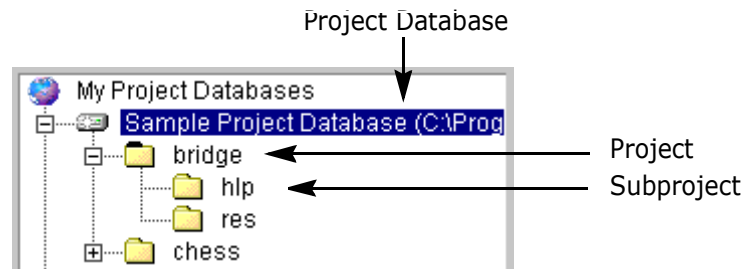
How users interact with projects depends on how the project database is set up:

- The configuration settings of a project database are inherited by the projects within it, but can also be overridden by a project's configuration file if the administrator has not disallowed this behavior in the project database.

- The access control database defined for the project database is inherited by the projects and identifies the users who are authorized to perform actions on the project database and identifies the actions each user can perform. An access control database can be created for individual projects and will override the access control database of the project database.

- The workspace set on a project database defines the workfile location assigned to each project and versioned file.

- The tool bar configuration file defined for the project database defines the custom tools and functions available in the custom tool bar.

Subprojects

Subprojects are projects that are nested within other projects.
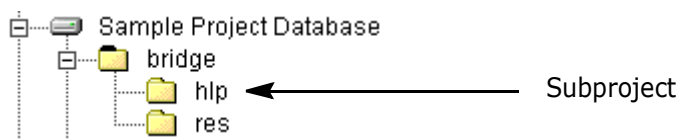
Subproject

Use subprojects to organize your versioned files just as you organize your workfiles on an operating or file system. When you add workfiles, Version Manager automatically creates subprojects to match the subdirectories of your workfile location. Alternatively, you can create subprojects and organize your versioned files to reflect some other organizational structure.

Explore It

1 Expand the sample project database.

2 Expand the chess project. The chess project contains two subprojects, client and server. Both of these subprojects contain additional subprojects, which have been nested to reflect a client/server application work structure.

Workspaces

A *workspace* is a collection of work settings defined for a project database. Only one workspace can be set on a project database by a user at any given time, and this workspace defines the work settings for all of the projects, subprojects, and versioned files within the project database.

Work settings include:

- The workfile locations defined for the project database and its projects, subprojects, and versioned files.

- The default version, which is the revision Version Manager will automatically operate on when you don't specify a revision number, version label, or promotion group.

- The base version and branch version used to facilitate automatic branching.

- The default promotion group, which is valid only if a promotion model is in effect. A default promotion group is a lowest-level promotion group of a promotion model. By default, Version Manager associates the default promotion group you specify with revisions when checking out revisions, locking revisions, and adding workfiles. By defining a default promotion group, you eliminate the need for a user to specify which lowest-level promotion group to use for these actions.

Why use workspaces?

Workspaces provide an effective way of managing work environments when multiple functional groups share a centrally located project database.

Typically, an administrator defines several workspaces, such as QA, DEV, and Production, for a project database if the project database is shared between multiple functional groups. Each of these workspaces would define different work settings. For instance, users in the QA group would have workfile locations defined within the K:\QA\Checkers location, whereas users in the Dev group might have workfile locations defined within the S:\Dev\Checkers location. Once these workspaces are defined, users can select the appropriate predefined workspace to use the workfile locations, default versions, base versions, and branch versions that are appropriate for their functional group.

Explore It

1   Select the sample project database.

2   Select File | Set Workspace.

3   Select Dev and click **OK**.

Notice how the path displayed in the Workfile Location field of the header bar changes to the Development workfile location: SampleDb\work\dev. The Dev workspace name also appears in the status bar.

Adding Workfiles

A *workfile* is a source file located on your operating system, from which you create a versioned file. *Versioned files* are files you place under version control. You can now add an entire directory of workfiles, create the versioned files, create the projects, and create the archives in one easy step. Version Manager automates the process and creates archive directories, using the existing structure of the workfile directories. When you add a workfile, you are actually checking in the initial revision of the file.

Explore It

1   Right-click the chess project.

2   Select Add Workfiles.

3   In the Add Workfiles dialog box, use the Browse button to locate the SampleDB\work\bridge directory and click **OK**.

4   In the Description text box, enter my files.

5   Select the **Use description for all** check box, and click **OK**. Version Manager creates the Bridge project, adds the workfiles, creates the versioned files, and creates the archives.

Getting and Viewing Files

You can get and view a read-only copy of a file without checking it out of an archive. When you get a revision, Version Manager leaves the revision in the state that it is in (locked or unlocked) and creates a read-only workfile in the location currently set as the workfile location.

Viewing a revision is basically the same as Getting a revision: it creates a read-only copy of the default revision. However, when you view a revision, Version Manager:

- Copies the workfile into a temporary directory. On Windows, the temp directory is defined by the TEMP environment variable. Version Manager creates a /pvcs directory in your temp directory the first time you view a file (for example, /temp/pvcs). On UNIX, this is defined by the pvcsvmux script, which sets it to /tmp/pvcs.

- Launches the application associated with the file type and opens the file. The editor Version Manager uses to display the file is either defined by the default Windows associations or by entering an editor location in the View | Options dialog box.

Because Version Manager does not lock the file or copy the file to the workfile location, you do not need to worry about obstructing other users' access to the file while you view it.

Explore It

1   Select the sample project database.

2   Expand the Chess project.

3   Highlight the server subproject.

4   In the File pane, double-click the server.bat file.

5   Select View and click **OK**.

6   In the Get File dialog box, notice that the path displayed in the **Copy To** field is a combination of the temp directory defined for your system and a /pvcs directory, which Version Manager automatically creates. Click **OK**. The file opens in a text editor.

Moving Projects and Versioned Files

If you need to reorganize your work, you can move projects and versioned files to new locations within the same project database by dragging and dropping the projects and versioned files to the new destinations. You can also move items by selecting the items and then selecting Edit | Move.

What happens when you move a project?
When you move a project, the project's workspace settings are retained if the workspace settings *are not* inherited from the project database. If you haven't changed the workspace settings, and the settings *are* inherited from the project database, the workspace settings of the project you move will be replaced by the workspace settings of the destination project.

Explore It

1 Expand the sample project database.

  a Expand the Bridge project.

  b Drag the hlp project over the res project and drop it.

  c At the prompt, click Yes. The hlp project is moved beneath the res project.

Copying Projects and Files

Drag and drop copying
You can copy an entire project and its contents or selected versioned files by dragging the item from one project database and dropping it into a second project database or project. You can also copy a project or versioned file by selecting Edit | Copy.

When you copy a project, workspace settings are not retained. The project assumes the workspace settings defined by the project database to which it was copied.

Merge Tool

*Supports *n*-way merging*
The Version Manager desktop client provides a powerful difference and merge tool that enables you to perform *n*-way merges on Windows platforms and two-way merges on UNIX platforms. On Windows, there is no limit to the number of files you can merge.

There are many options you can customize for both the difference and merge functions, including:

- Whether or not case is ignored

- Which colors denote differences and conflicts

- Which font styles to use

- Whether or not to ignore blank spaces and tabs

- Whether or not to use line numbers

Many of these options, including file selection, can be saved in a configuration file for later use.

Locating Archives

In Version Manager you define the archive location when you first create a project database. By default, Version Manager automatically creates the archive location using a path relative to the project database location. However, you can also modify this path to use a different location.

Project Information Locations

On an operating or file system, the project database location contains information about the project database, such as configuration options and user information. These files are stored in directories that are created by Version Manager in the locations specified in the Create Project Database dialog box. The directories include:

- An *archives* directory.

- A *work* directory.

- A *pvcsuser* directory, which is automatically created within the project database location.

- A *lib* directory, which is automatically created within the project database location.

- Several Version Manager project database files that identify the location as a project database location. These data files are stored at the root of the project database directory.

User Workflow for the Desktop Client

The most basic tasks a typical user performs using the Version Manager desktop client are shown in the following graphic. The numbered exercises in the following sections follow this workflow.

As the user, you:

1 Open the project database created by the administrator

2 Set a workspace for the project database

Your administrator first sets up the project databases you'll be working with...

3 Create new projects of versioned files by adding directories of workfiles

4 Get and view revisions

5 Check out revisions for editing

6 Check in workfiles to create new revisions
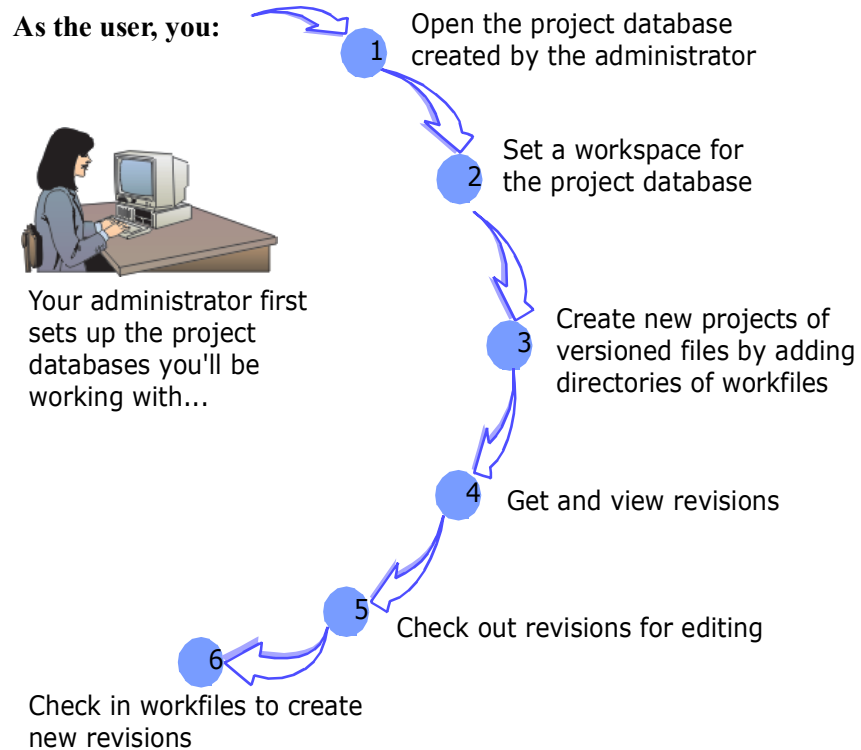
1 Opening a Project Database

Objective: To open a project database using the File menu.

Scenario: Although your administrator already set up a project database for several sample game applications, you must add a project for the new Checkers project you've been assigned to work on. Before you can do this, you must open the project database.

> **NOTE** If you already opened the sample project database from the Welcome to Version Manager dialog box, you will need to close it first before completing the rest of this exercise.

To close a project database:

1 If the sample project database is already open, select it in the Project pane.

2 Select File | Close Project Database. The sample project database closes, and it is removed from the Project pane.

To open the sample project database:

1 Select File | Open Project Database. The Open Project Database dialog appears.

Indicates directory is a project database →
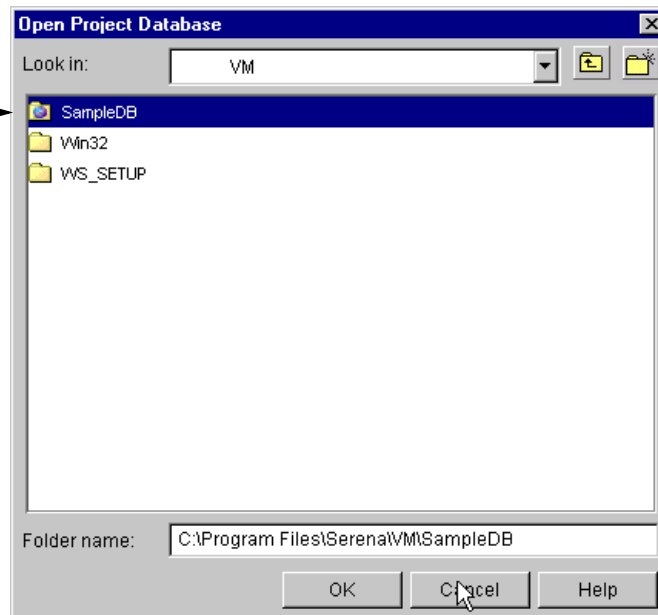
2 Select the location of the sample project database. It is relative to the Version Manager installation directory. Unless you modified the default installation directory, the sample project database is installed to one of the following locations:

- **On UNIX:** /usr/serena/vm/common/sampledb

- **On Windows:** C:\Users\All Users\Serena\VM\SampleDB

TIP If the project database hosted on a Version Manager file server:

a Select **File Servers** in the **Look in** field.

b Double-click on **Path Map View** or **Project Database View**.

c Navigate to, and select, the project database.

3 Click **OK**. If this is the first time you have opened the sample project database, the Login dialog box appears.
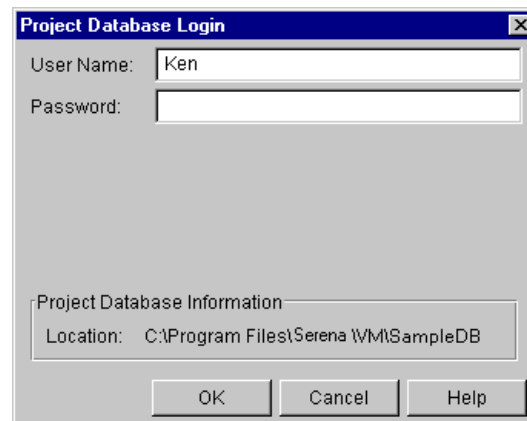
4 Using the information your administrator supplied you, enter Ken as the user name. The administrator did not assign a password, so you do not need to enter a value in the Password field.

5 Click **OK**. Version Manager logs you in to the sample project database, which now appears in the Project pane.

Now you have learned:

■ How to close a project database.

■ How to open the sample project database.

■ How to log in to a project database that has been configured to use the Login Dialog login source.

Some notes about project databases...

■ Expand the view of the project database by clicking the plus sign next to the project database name.

■ Collapse the view of the project database by clicking the minus sign next to the project database name.
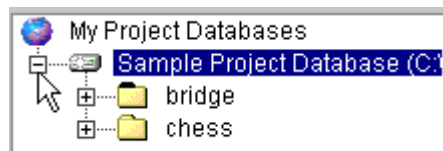
For more information Refer to "Working with Project Databases" in the *User's Guide* or "Setting Up Version Manager Using the Desktop Client" in the *Administrator's Guide*.

2 Setting a Workspace

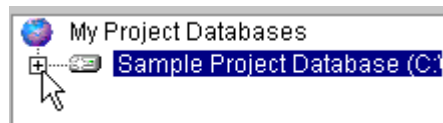**Objective:** To set a workspace and define a default workspace.

A *workspace* is a collection of work settings that enable you to customize your environment according to your work preferences. A workspace stores information about where your workfiles are located, how you select revisions (using a revision number, version label, or promotion group), and how or if you work with automatic branching.

A *default workspace* is the workspace that Version Manager automatically sets when you open the project database.

Scenario: Your administrator set up five workspaces for the sample project database:

Workspace	Workfile Location
Root	C:\Users\All Users\Serena\VM\SampleDB\work\
QA	C:\Users\All Users\Serena\VM\SampleDB\work\qa
Dev	C:\Users\All Users\Serena\VM\SampleDB\work\dev
Production	C:\Users\All Users\Serena\VM\SampleDB\work\prod
UNIX	/usr/serena/vm/common/sampledb/work

Because your role is that of a Developer, you will use the Dev workspace on the sample project database.

To set the Dev workspace:

1 You are not sure which workspace is currently set on the project database, so you select the sample project database and then check the status bar to see which workspace is current, as shown in the following graphic.
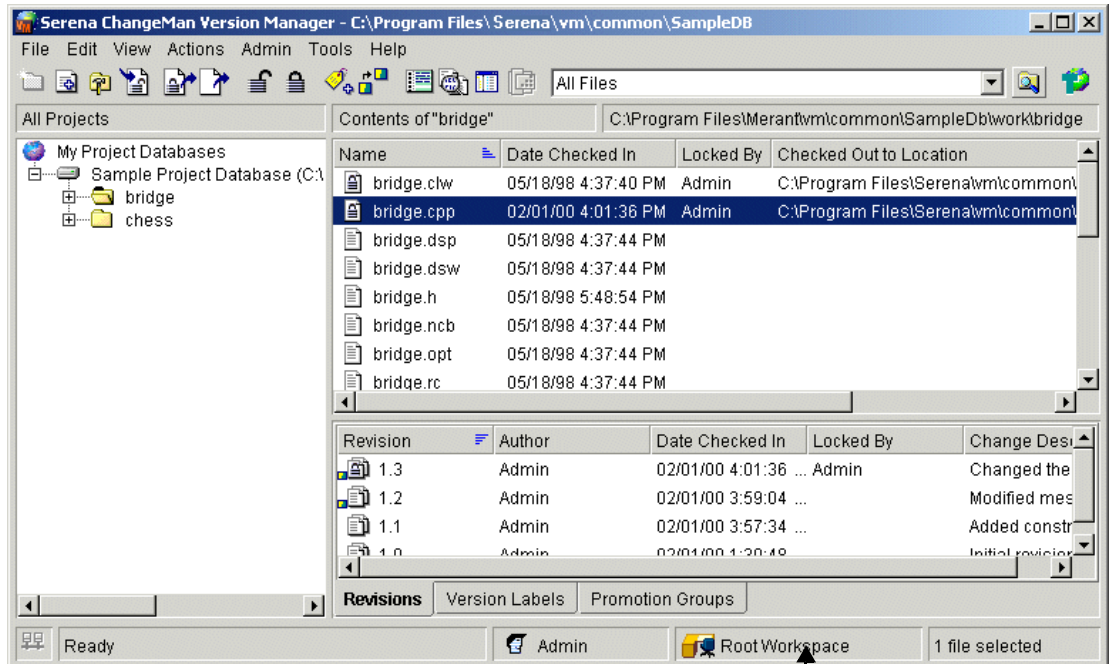
Name of current workspace

You see that the Root Workspace is currently set, so you need to change to the Dev workspace.

2   Click the workspace name in the status bar or select File |
Set Workspace. The Set Workspace dialog box appears, showing Root Workspace as
the default and current workspace.

Default and
current workspace
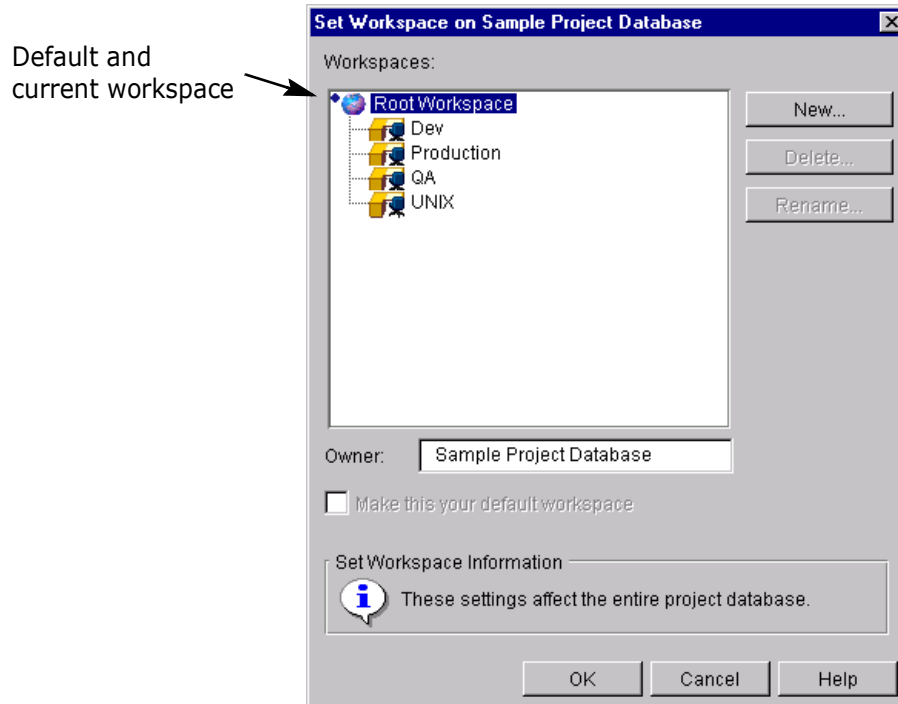
The default workspace is identified by a diamond. The current workspace is
highlighted.

3   Select Dev and then select the **Make this your default workspace** check box.

Version Manager automatically sets the workspace defined as the default workspace
whenever you open a project database.

4   Click **OK**.

Version Manager associates the Dev workspace with your user information for the
sample project database. Whenever you open the sample project database, Version
Manager automatically sets this Dev workspace. However, when other users open this
same project database, Version Manager sets the workspace that they defined as the
default workspace, which might not be the same default workspace you defined.

Now you have learned:

■   How to identify the current workspace.

■   How to set a workspace.

■   How to define a default workspace so that Version Manager automatically sets the
workspace each time you open the project database.

Some notes about workspaces...

■   To verify which workspace is set, look at the status bar. When you select a project
database, the name of the current workspace appears in the status bar to the right of
the current user name. This workspace applies to all projects and subprojects within
the project database. Each project database has a different workspace.

- To verify a workfile location set for a project database, project, or subproject, select the item and look in the Workfile Location field above the File pane.

- You can modify the settings of the current workspace by changing the workfile location defined for a versioned file, project, subproject, or the project database. To change a workfile location, select the item and select File | Set Workfile Location.

- To verify or change the workspace settings defined for a project database, project, subproject, or versioned file, select the item and select File | Properties. The Workspace Settings tab contains the values of the workspace settings.

For more information   Refer to "Using Workspaces" in the *User's Guide* or "Managing Your Environment" in the *Administrator's Guide*.

3 Adding Workfiles

Objective: To add an entire directory of workfiles and automatically create a project, the versioned files, and the archives. When you add workfiles, you check in the initial revisions of the file.

For every workfile you add, Version Manager creates a versioned file and an archive. The versioned file is created in the project selected in the window when you added the workfile. The archive is created on your operating or file system in the archive location defined for the project.

When you add an entire directory (including subdirectories) of workfiles, Version Manager creates a subproject for each subdirectory of workfiles. Version Manager also creates a subdirectory for the archives in the archive location defined for the project. The subproject and the archive directory on your operating or file system share the same name.

Scenario: While working on the Checkers application, you have created numerous files. Your Project Leader has asked you to place these files under version control and assign each file a description identifying it as a component of the Checkers application. To place the files under version control, you must create a project and add the workfiles to it. You can do this in one step by adding the entire directory of Checkers workfiles.
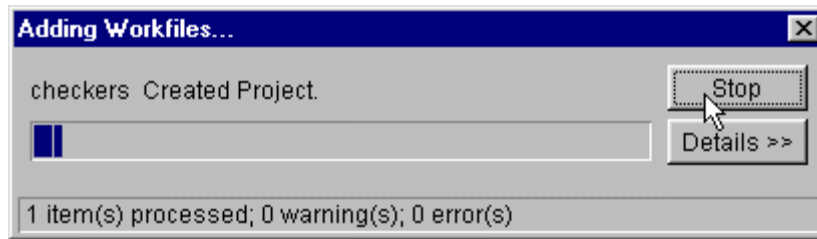
To add the Checkers workfiles:

1 In the Project pane, select the sample project database.

2 Select File | Add Workfiles. The Add Workfiles dialog box appears.
By default, the location displayed in this dialog box is the location defined as the
workfile location for the project database.
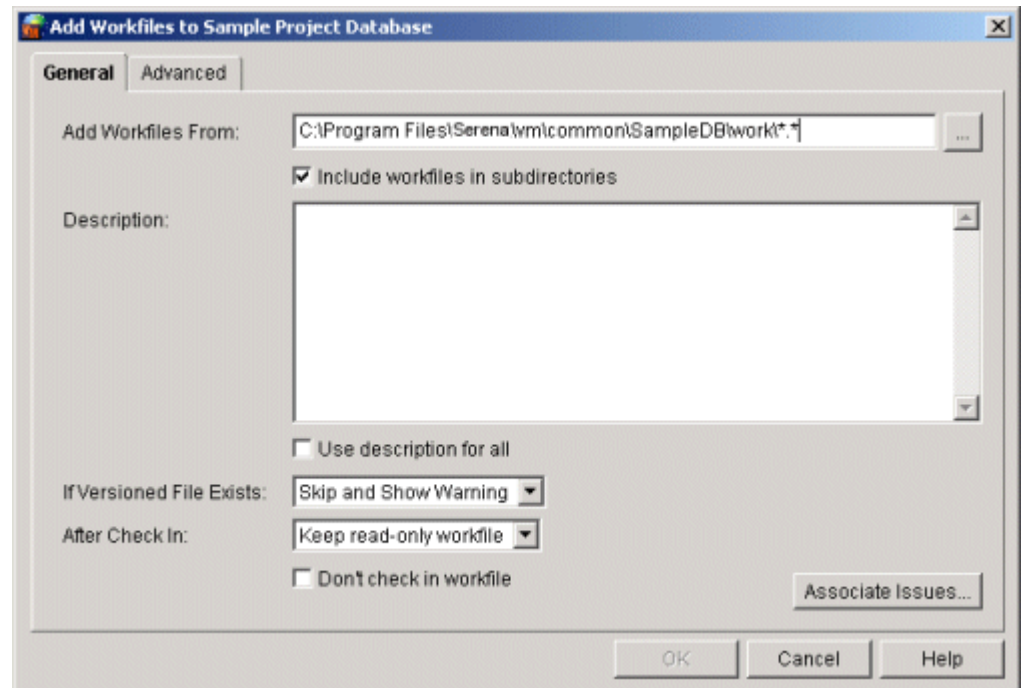
3 Do the following:

- In the **Add Workfiles From:** edit box, navigate to one of the following locations,
  which contain all of the workfiles you need to place under version control:

  - **On Unix:**
    /usr/serena/vm/common/sampledb/work/dev/checkers

  - **On Windows:**
    C:\Users\All Users\Serena\VM\SampleDB\work\dev\checkers

- Verify that *.* (Windows) or * (UNIX) appears at the end of the path.

4 The Project Leader asked you to assign each file a description identifying it as a
component of the Checkers application, so you need to add a description to each
archive. To do this, you:

- Enter Checkers Project in the **Description** field.

- Select the **Use description for all** check box.

5 Because the folder containing the Checkers files contains two subdirectories—Client
and Server—you retain the remaining default options, including the option to include
workfiles in subdirectories. This last option instructs Version Manager to automatically
create subprojects for any subdirectories within the selected directory of workfiles and
automatically add the workfiles in the subdirectories.

When you click **OK**, Version Manager adds the files, and the progress dialog box appears.

While the progress is monitored, Version Manager:

- Creates the Checkers project and adds it to the sample project database.

- Creates two subprojects—Client and Server—to match the subdirectories in the Checkers workfile directory.

- Adds a versioned file to the new Checkers project for every workfile in the Checkers directory and subdirectories. Each versioned file has one revision (the initial revision).

- Creates an archive for each versioned file, using the default archive suffix: –arc. The archives are created in the archive location defined for the sample project database (sampledb/archives or sampledb\archives). Within this archive location, Version Manager creates a new checkers directory, and stores the archives using the same hierarchy used in the checkers workfile directory.

- Assigns the existing location of the Checkers workfiles as the workfile location for the new Checkers project.

- Assigns the description *Checkers Project* to each new archive.

- Assigns the default revision description *Initial revision* to the first revision of each versioned file.

Now you have learned:

- How to add workfiles to a project database for the first time.

- Where Version Manager stores the archives for the versioned files.

- How to create a project structure that replicates your workfile structure, by including the workfiles in subdirectories.

- How to enter descriptions for archives.

Some notes about adding workfiles...

- To view information about the revisions that were created, select View | Show Revisions, if the Revision pane is not already open. Expand a project in the Project pane and select a versioned file in the File pane. Information about the revisions within the selected versioned file displays in the Revision pane.

- To modify the default options that pertain to workfile actions, such as what happens when you check in a workfile, select View | Options and change the values in the Check In/Out Options group box.

For more information Refer to "Adding Workfiles" in the *User's Guide* or "Setting Up Version Manager Using the Desktop Client" in the *Administrator's Guide*.

4 Getting and Viewing Revisions

Objective: To get a revision and view its contents without locking or checking out the revision.

When you get a revision, Version Manager leaves the revision in the state that it is in (locked or unlocked) and creates a read-only workfile in the location currently set as the workfile location.

Viewing a revision provides the same basic function as the Get function: it creates a read-only copy of the default revision. By viewing the file, you can decide whether or not you want to check out the revision to edit it. However, when you view a revision, Version Manager:

■ Copies the workfile into a temporary directory. On Windows, the temp directory is defined by the TEMP environmental variable. Version Manager creates a /pvcs directory in your temp directory the first time you view a file (for example, /temp/pvcs). On UNIX, this is defined by the pvcsvmux script, which sets it to /tmp/pvcs.

■ Launches the application associated with the file type and opens the file. The editor Version Manager uses to display the file is either defined by the default Windows associations or by entering an editor location in the View | Options dialog box.

Because Version Manager does not lock the file or copy the file to the workfile location, you do not need to worry about obstructing other users' access to the file while you view it.

Scenario: The first issue you need to address in the Checkers application is a problem with the server. Another Developer mentioned that he entered some information about the problem in the shutdown.txt file, so you need to get this file and read it before you make any changes to the code. Since the shutdown.txt workfile was in the Server subdirectory on your operating system, you know that in Version Manager the shutdown.txt versioned file exists in the Server subproject.

To get the latest revision of shutdown.txt:

1   Expand the Checkers project and select the Server subproject. The shutdown.txt versioned file is listed in the File pane, as shown below.
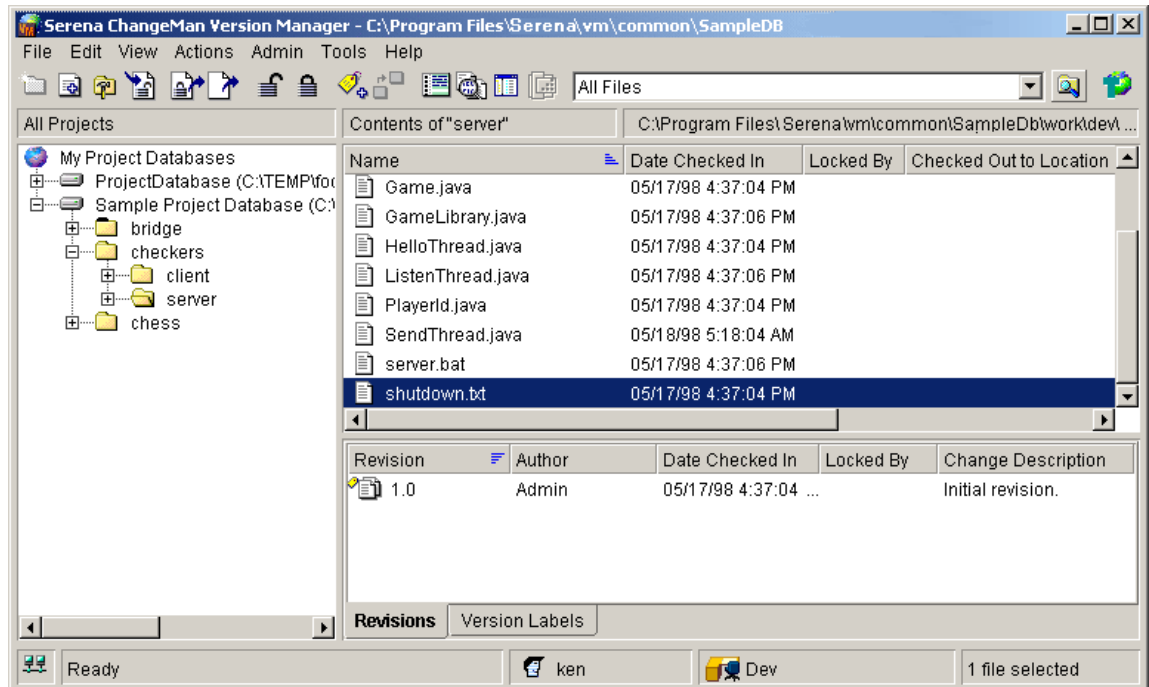

2   Double-click the shutdown.txt file. The View or Edit prompt appears.
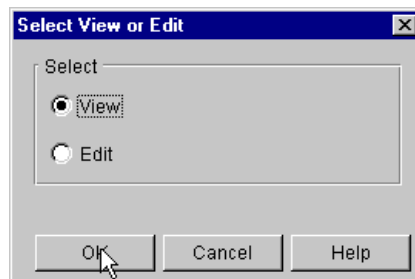
NOTE The behavior associated with double-clicking may be changed by selecting View | Options and accessing the Applications dialog box. The Default Double-Click Action option may be set to Ask, View File or Edit file. The default setting is Ask.

3 Select View and click **OK**. The Get File dialog box appears.
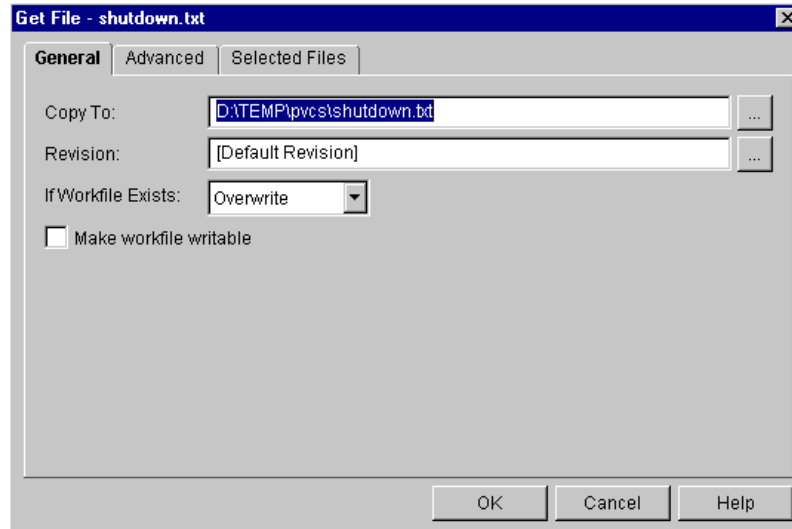
Note that the path defined in the **Copy To** field is a combination of the temp directory defined for your system and a pvcs directory, which Version Manager automatically creates.

4 Click **OK** without changing the value of any other field. This retains the default settings and Version Manager does the following:

- Retrieves a read-only copy of the revision specified in the **Revision** field and copies it to the pvcs directory beneath the temp directory defined for your system.

- Opens the read-only workfile using the editor associated with text files (or the editor defined using View | Options).

You can now review the contents of the file.

5 After reviewing the information, you realize you'll need to review a different file before you can understand the problem, so you exit the text editor to close the file.

Now you have learned:

- How to retrieve a read-only copy of a revision for viewing.

- Where Version Manager creates the read-only copy of the revision.

Some notes about getting revisions...

- By default, when you get a revision, Version Manager retrieves a temporary, read-only copy of the file, which is created in a pvcs directory beneath the temp directory defined for your system.

- If you plan on editing a file, you should check out the file with a lock instead of just getting it.

- You can get a file by selecting Actions | Get, but Version Manager only launches the file in an editor if you view or edit the file (double-click the versioned file in the File pane or select View or Edit from the Edit menu or context menu).

For more information
Refer to "Getting Revisions" in the *User's Guide*.

5 Checking Out and Editing Revisions

Objective: To check out and automatically open a revision for editing in the appropriate editor.

When you check out a revision, Version Manager places a lock on the revision using your user ID. Typically, no one else can check in changes until you remove the lock by checking in your changes or unlocking the revision. The only exception to this rule is when your administrator configures the project so that users can place multiple locks on revisions for branch development.

Scenario: Because the shutdown.txt file didn't contain any useful information, you decide to check out a batch file to see if there is a problem with the server commands it contains. Because this is also a server file, you're pretty sure this file is also stored in the Server subproject.

To check out and edit the batch file:

1   If it isn't already expanded, expand the Checkers project and select the Server subproject. In the File pane, you see the server.bat file you need.

2   In the File pane, double-click the server.bat versioned file.

3   At the prompt, select Edit and click **OK**. The Check Out dialog box appears.
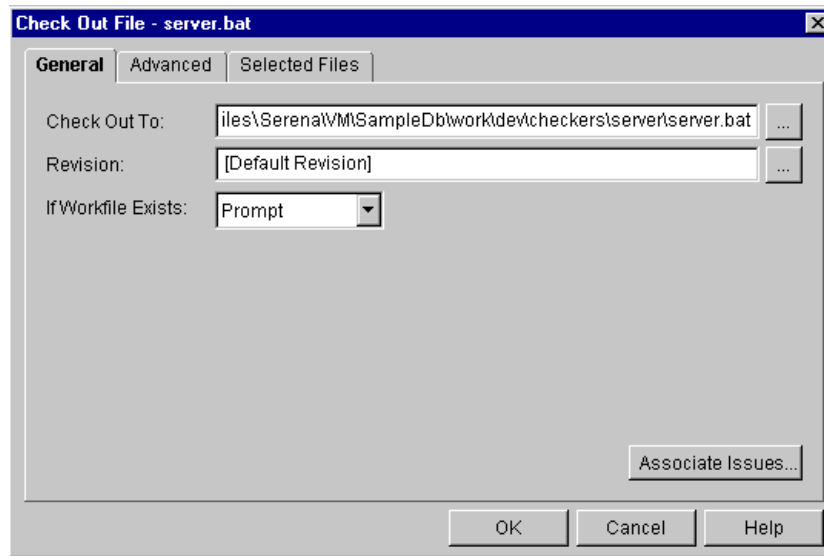
4   Click **OK** without changing the values of any other fields. The default settings are retained and Version Manager does the following:

   ■   Retrieves a writable copy of the server.bat file and copies it to the path identified in the **Check Out To** field.

   ■   Places a lock on the revision with your user name.

   ■   Opens a writable copy of server.bat in a text editor, enabling you to edit the contents of the file.

5   You see that there are several references to "chess" in the file, which need to be changed to "checkers," so you make the changes and save the file.

Now you have learned:

- How to check out a file with a lock.

- How to open the file in an editor.

Some notes about checking out a revision...

- By default, when you check out a revision, Version Manager locks the revision and provides you with a writable copy of the workfile.

- You can check out a file by selecting Actions | Check Out, but Version Manager only launches the file in an editor if you view or edit the file (double-click the versioned file in the File pane or select View or Edit from the Edit menu or context menu).

For more information   Refer to "Checking Out Revisions" in the *User's Guide*.

6 Checking In Workfiles

Objective: To check in a workfile that contains changes.

When you check in a workfile, Version Manager stores it in an archive and creates a revision. You cannot check in a workfile unless you have already locked the revision.

Scenario: Because you fixed the problem in server.bat, you want to archive your changes. To do this, you'll need to check the workfile into Version Manager.
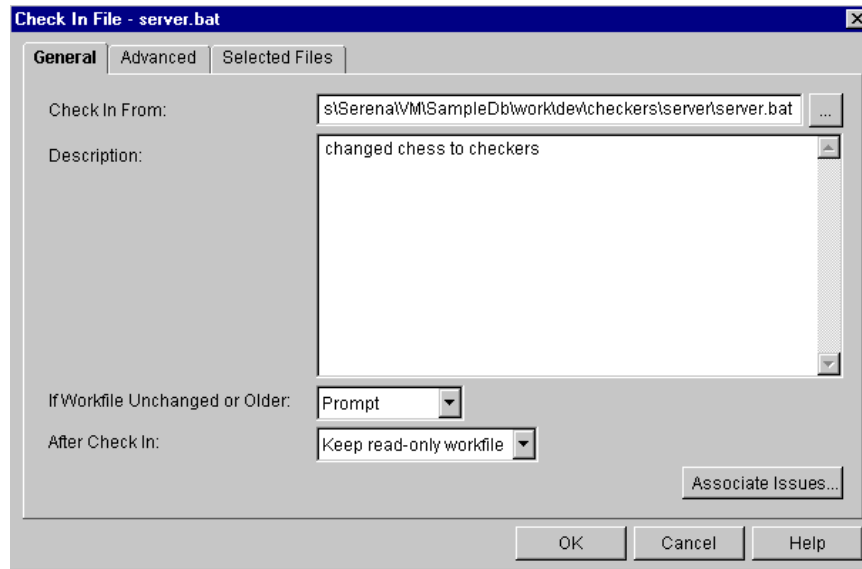
To check in server.bat:

1   If it isn't already open, expand the Checkers project and select the Server subproject.

2   Select the server.bat versioned file, which now shows a lock icon, as shown in the graphic below.

SendThread.java	05/18/98 5:18:04 AM	
server.bat	05/17/98 4:37:06 PM	ken
shutdown.txt	05/17/98 4:37:04 PM	

3   Right-click the versioned file and select Check In. The Check In dialog box appears.

4 You decide to add information about the change you made, so you enter changed chess to checkers in the **Description** field.

5 Click **OK** without changing the values of any other fields. The default settings are retained and Version Manager does the following:

- Checks in server.bat from the location shown in the **Check In From** field.

- Retains a read-only copy of the server.bat workfile.

- Does not delete the workfile or retain a lock on the revision.

- Assigns the description to the revision.

Now you have learned:

- How to check in a file and create a new revision of the versioned file.

- How to enter a description for the changes contained in the new revision.

Some notes about checking in workfiles...

- You can check in entire directories of workfiles at once, including workfiles in subdirectories.

- You can perform more advanced check in actions, such as forcing a branch, using the Advanced tab of the Check In dialog box.

For more information   Refer to "Checking In Items" in the *User's Guide*.
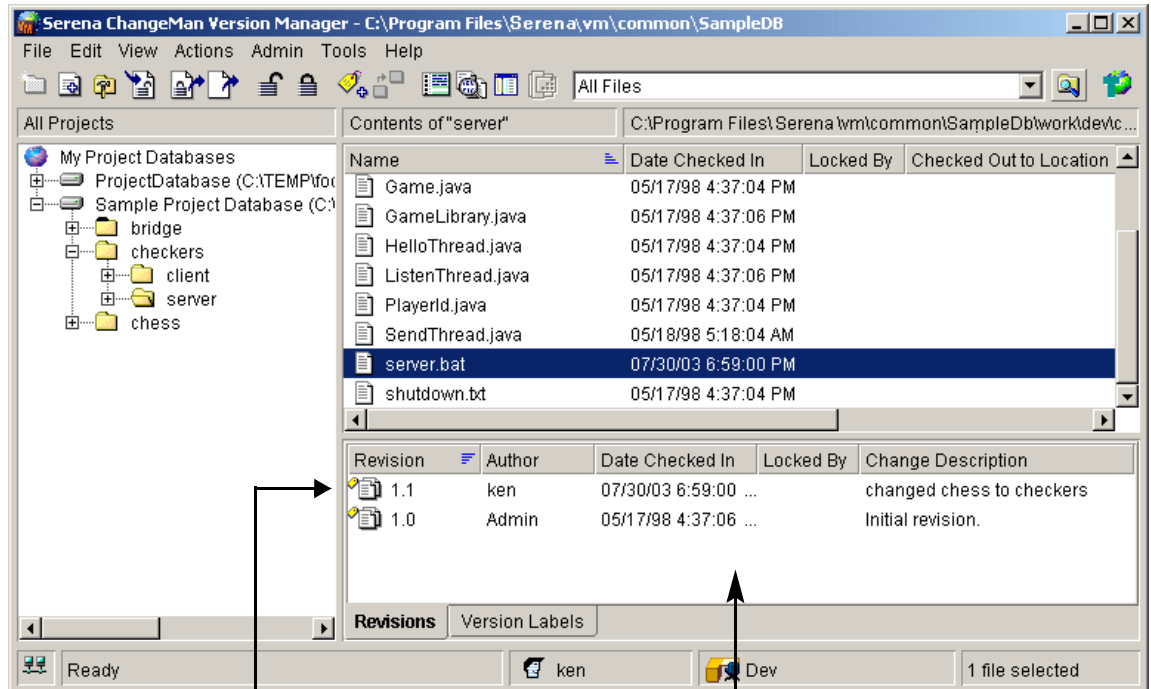
Reviewing Revisions

Objective: To verify that the versioned file contains a new revision by reviewing the contents of the Revision pane.

Scenario: Now that you checked in a workfile, you want to review the changes that were saved to the archive.

To review your changes:

1 If it isn't already open, expand the Checkers project and select the Server subproject.

2 If the Revision pane isn't already open in the lower-right corner of the Version Manager desktop, select View | Show Revisions. The Revision pane appears, as shown in the next graphic.

New revision
you added

Revision pane

3 Select the server.bat versioned file. The revision you added to the archive is shown in the pane, along with the change description you entered.

Now you have learned:

How to verify revision information by reviewing the contents of the Revision pane.

Working with the Command-Line Interface

In this section, you will work with the sample project database files using the command-line interface. The files you will work with are the same files provided in the sample project database, described in "About the Sample Project Database and Scenarios" on page 28. However, because the command-line interface does not recognize projects or project databases, the goal of these exercises is to demonstrate how you can use the command-line interface to modify files contained within a project database that was created using the desktop client interface.

NOTE If you have already completed the section, "Working with the Desktop Client" on page 28, you must delete the contents of the sampledb\archives\checkers\server directory before continuing with the following procedures.

About the Sample Project Files and Scenarios

By default, when you install Version Manager, the sample project database is installed to the following locations:

- **On UNIX:** /usr/serena/vm/common/sampledb

- **On Windows:** C:\Users\All Users\Serena\VM\SampleDB

If you modify the default installation location, the sample project database is installed in a location relative to the Version Manager installation directory.

At the sample project database location, there are sample files for three games: Chess, Checkers, and Bridge. Only the files in the Chess and Bridge projects have been archived. The Checkers files have not been archived—you will do this yourself in the following exercises. You will also perform some of the basic tasks that a Developer might perform while developing a new application.

While working through these exercises, assume the following:

- You work as a Developer for a large company that designs computer games. Your task is to begin work on a new game: Checkers.

- Your administrator has already set up a project database for the sample games under development using the Version Manager desktop client by doing the following:

  - She created an archive location for the Checkers project.

  - She added HOST to the login sources for the project database, so that you can access the archives from the command-line interface.
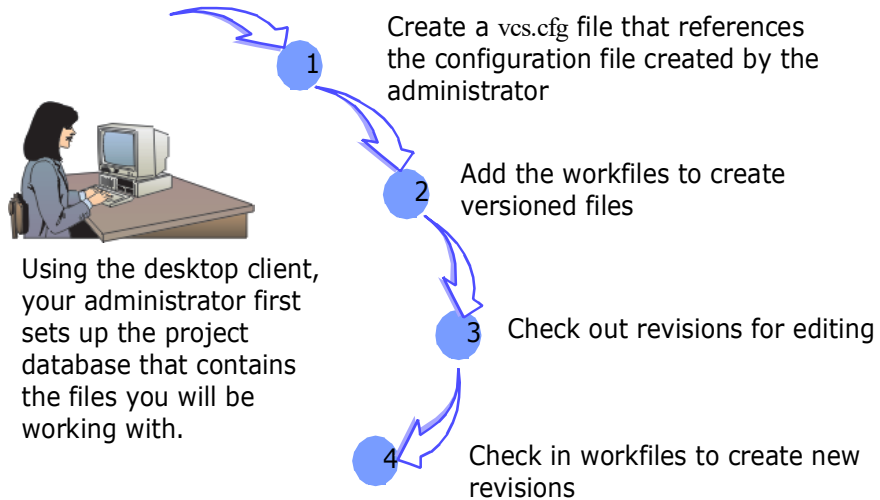
Now you must archive the Checkers files using the command-line interface.

User Workflow for the Command-Line Interface

The most basic tasks a user will perform using the Version Manager command-line interface are shown in the graphic below. Note that this workflow differs considerably from the desktop client workflow shown on page because the command-line interface does not recognize projects or project databases. Instead, when you use this interface, you

work directly with the archives. The numbered exercises in the following sections follow this workflow.

As the user, you:

1. Create a vcs.cfg file that references the configuration file created by the administrator

2. Add the workfiles to create versioned files

3. Check out revisions for editing

4. Check in workfiles to create new revisions

Using the desktop client, your administrator first sets up the project database that contains the files you will be working with.

1 Creating a vcs.cfg File

Objective: To create a vcs.cfg file.

A vcs.cfg file is an ASCII text file that contains directives. These directives control how Version Manager commands operate on archives. To use Version Manager from the command line, you must do one of the following:

■ Create a vcs.cfg file in your workfile directory (the directory from which you check in files and to which you check out files).

■ Set a vcscfg environment variable that specifies the name and location of the project database's configuration file, as it was defined in the desktop client.

Scenario: Your administrator has already set up a new project database using Version Manager. When she set up the project database, she also created an archive and work directory for a new Checkers application, which you've been assigned to develop.

Because you primarily work from the command prompt, you'd like to check files into and out of archives using the Version Manager command-line interface. To do this, you first need to create a vcs.cfg file and copy it to the appropriate work directory.

To create a vcs.cfg file:

1 In a text editor, create one of the following ASCII files:

On Windows: enter the following three lines:
INCLUDE C:\Users\All Users\Serena\VM\SampleDB\archives\basecfg.cfg
NoAccessControl
NODELETEWORK

On UNIX: enter the following three lines:
INCLUDE /usr/serena/vm/common/sampledb/archives/basecfg.cfg
NoAccessControl
NODELETEWORK

What it means:

The INCLUDE directive specifies where the configuration file for the project database that contains the files you'll be working with is located. To determine this path, start Version Manager and view the Properties of the appropriate project database. Under the Properties tab, the path of the configuration file is listed in the **Configuration File** field.

The configuration file defines the archive location. Since the vcs.cfg you just created references the project database configuration file, it does not need to define the archive location because the configuration file already defines the location.

NOTE If there are blank spaces in the path to the configuration file, use quotation marks around the path. For example, if the sample project database is installed to the default location, define the path as:

include "C:\Users\All Users\Serena\VM\SampleDB\archives\basecfg.cfg"

The NoAccessControl directive instructs Version Manager to ignore the access control database defined for the sample project database. This directive can only be used if your administrator has not modified the configuration of the sample project database by disallowing this directive. If the directive is disallowed, you will not be able to complete the command-line interface tutorial unless the administrator adds your user ID to the access control database associated with the sample project database. Typically, administrators will disallow this directive in actual project databases so that users cannot override access control databases.

For more information about setting up security, refer to the *Administrator's Guide*.

The NODELETEWORK directive instructs Version Manager to not delete the workfiles after you check them into the archives. This directive is optional.

2 Save the file as vcs.cfg and copy it to the Server workfile location, which is located in one of the following directories by default:

- **On UNIX:**

   /usr/serena/vm/common/sampledb/work/dev/checkers/server

- **On Windows:**

   C:\Users\All Users\Serena\VM\SampleDB\work\dev\checkers\
       server

   All of the files in this directory will be archived in C:\Users\All Users\Serena\VM\SampleDB\archives\checkers\server.

3 If you will be working with other work directories that map to other archive locations, create additional vcs.cfg files for those directories.

Now you have learned:

- How to create a vcs.cfg file.

- What information to put in the vcs.cfg file.

- Where to locate the vcs.cfg file.

For more information Refer to "Setting Up Version Manager Using the Command-Line Interface" in the *Administrator's Guide*.

2 Adding Workfiles

Objective: To add a workfile for the first time, which creates the archive, a versioned file, and the first revision.

Scenario: When you were assigned the Checkers project, you were given some code that another Developer had begun to write. These files exist in the work/dev/checkers/server directory. To safeguard these original files, you want to add the workfiles to archives.

To add workfiles:

1   Open a command prompt window, and from the command prompt, navigate to the workfile location:

   ■   **On UNIX:**

      /usr/serena/vm/common/sampledb/work/dev/checkers/server

   ■   **On Windows:**

      C:\Users\All Users\Serena\VM\SampleDB\work\dev\checkers\server

2   Add all of the workfiles to the archive location by issuing one of the following commands:

   On UNIX:

      put /usr/serena/vm/common/sampledb/archives/checkers/server\(*.*\)

   On Windows:

      put "C:\Users\All Users\Serena\VM\SampleDB\archives\checkers\
         server(*.*)"

> **NOTE** If there are blank spaces in the path, use quotation marks around the path. For example, if the sample project database is installed to the default location, define the path as:
>
> put "C:\Users\All Users\Serena\VM\SampleDB\
> archives\checkers\server(*.*)"

3   At the prompt, enter for the checkers project and press CTRL+Z to assign the description to the first archive.

4   At each prompt to use the same description, enter Y and press ENTER.

   Version Manager does the following:

   ■   Creates one archive for each workfile and enters *for the checkers project* as the description of each archive.

   ■   Creates the first revision in each archive and enters *initial revision* as the description of each revision.

   ■   Attaches the same archive suffix (–arc) to the new archives that were defined in Version Manager, ensuring access to the archives using either the desktop client or command-line interface.

Now you have learned:

■   How to add a workfile for the first time.

■ How to enter a description for the archive, which is created by Version Manager the first time you add a workfile.

Some notes about the command-line interface...

■ To access versioned files in the Version Manager desktop client that you added as workfiles from the command-line interface, you must first import the archives in the desktop client. When you import an archive, Version Manager creates the graphical representation of the versioned file in the desktop client and establishes an association between the versioned file and the archive.

For more information   Refer to "Setting Up Version Manager Using the Command-Line Interface" in the *Administrator's Guide* or "Put" in the *Command-Line Reference Guide.*

3 Checking Out Revisions

Objective: To check out a revision for editing.

When you check out a revision, Version Manager places a lock on the revision with your user ID. Typically, no one else can check in changes until you release the lock by checking in your changes or unlocking the revision. The only exception to this rule is when your administrator enables multiple locking for branch development.

Scenario: To get started on the Checkers project, you need to access the CheckersServer.html file, which contains information about previous changes that were made to the Checkers application. You also might need to modify the contents of this file, so you want to lock the revision, too. To do this, you must check out the latest revision of the file with a lock.

To check out CheckersServer.html:

1   Navigate to the Checkers workfile location:

■ **On UNIX:**

/usr/serena/vm/common/sampledb/work/dev/checkers/server

■ **On Windows:**

C:\Users\All Users\Serena\VM\SampleDB\work\dev\checkers\server

2   Enter get -l CheckersServer.html

Version Manager retrieves the latest version of the file, locks the revision using your HOST ID, and copies the workfile to the working directory.

Now you have learned:

How to check out a file with a lock.

For more information   Refer to "Get" in the *Command-Line Reference Guide.*

4 Checking In Revisions

Objective: To check in a workfile that contains changes.

When you check in a workfile, Version Manager stores the changes in the workfile's archive and creates another revision. Before you check in a workfile, you must have a lock on the revision in Version Manager. Only the person who locked a revision can check in a

new version of it, unless your administrator has configured the project to allow multiple locks.

Scenario: Now that you have reviewed and made a few changes to the CheckersServer.html file, you'd like to check in your modifications. To do this, you'll need to check the workfile into Version Manager.

To check in CheckersServer.html:

1 Navigate to the Checkers working directory:

- **On UNIX:**

  /usr/serena/vm/common/sampledb/work/dev/checkers/server

- **On Windows:**

  C:\Users\All Users\Serena\VM\SampleDB\work\dev\checkers\server

2 Enter put CheckersServer.html

Version Manager checks the file into its archive and then releases the lock that was set with your HOST ID. Your changes become the next revision in the archive, which other users can now access. You can import these archives to the Version Manager desktop client, and then you can review the changes you archived. For more information, refer to "Reviewing Revisions" on page 48.

Now you have learned:

How to check in a file.

For more information Refer to "Put" in the *Command-Line Reference Guide.*

Chapter 3

Strategies For Working With Version Manager

Introduction

This chapter introduces strategies for using Version Manager to best meet the demands of your work environment.

Whether you want to work from a remote location, automate processes with scripts, or perform source control actions from within a development environment, Version Manager has the features you need.

Ways to Use the Desktop Client

For both administrators and users, the Version Manager desktop client provides the most comprehensive functionality.

Reviewing Properties

You can review the properties (attributes) of any item in the desktop client, including a project database, a project, a subproject, or a versioned file. The types of properties you can review depends upon the selected item.

To review the properties of an item, right-click the item and select Properties. For more information about properties, refer to the *User's Guide*.

> **NOTE** You can also review properties from a Version 5.3/6.0 folder.

Hiding Dialog Boxes

To turn off one or more dialog boxes, select View | Options and select the Dialog Behavior tab, which is shown below.
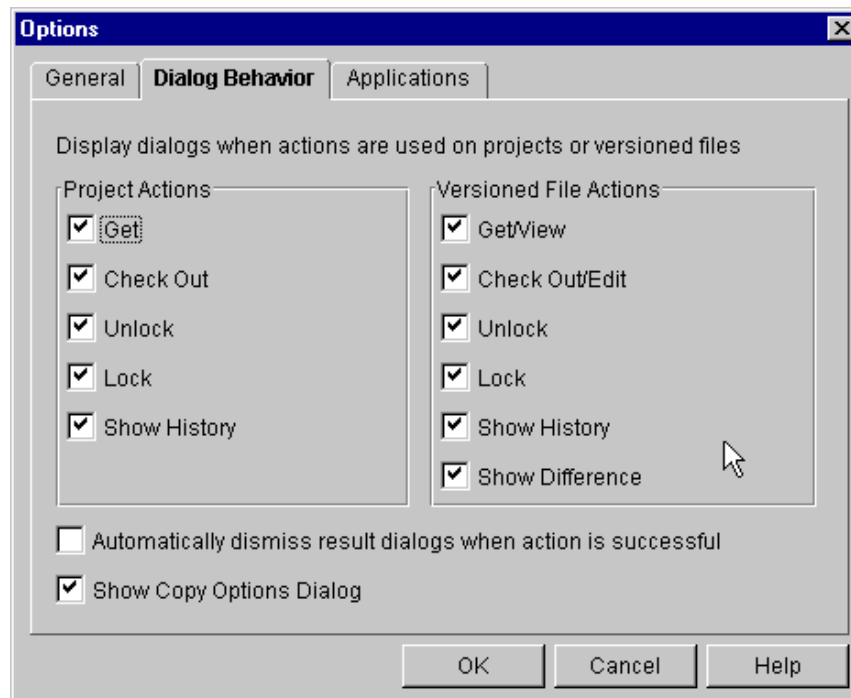
For more information about setting dialog box options, refer to the *User's Guide*.

Using the Revision Pane to View Change History

The Revision pane provides quick access to the complete history of a versioned file. For each revision of a versioned file, the Revision pane shows the author of the revision, the date it was last checked in, the users who hold a lock on it, the change description entered for it, the version labels assigned to it (if any), the promotion group assigned to it (if any), and the branch it belongs to if a branch exists.

The Revision pane is shown below.
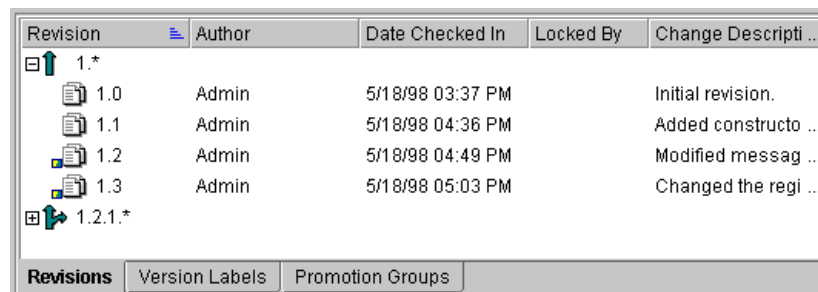
To view the Revision pane, select View | Show Revisions.

For more information about the Revision pane, refer to the *User's Guide*.

Using Workspaces to Change Drives

Customize work settings

When working in a project database shared by multiple users, you can use workspaces to customize the work environment for each user or each group of users. For instance, if everyone in your organization uses the same archives, but the Quality Assurance engineers work on one drive, such as *Z:* or */home/build/work*, and the development engineers work on another drive, such as *Y:* or */home/test/work*, you can set up a separate workspace for each group.

For more information about workspaces, refer to the *User's Guide*.
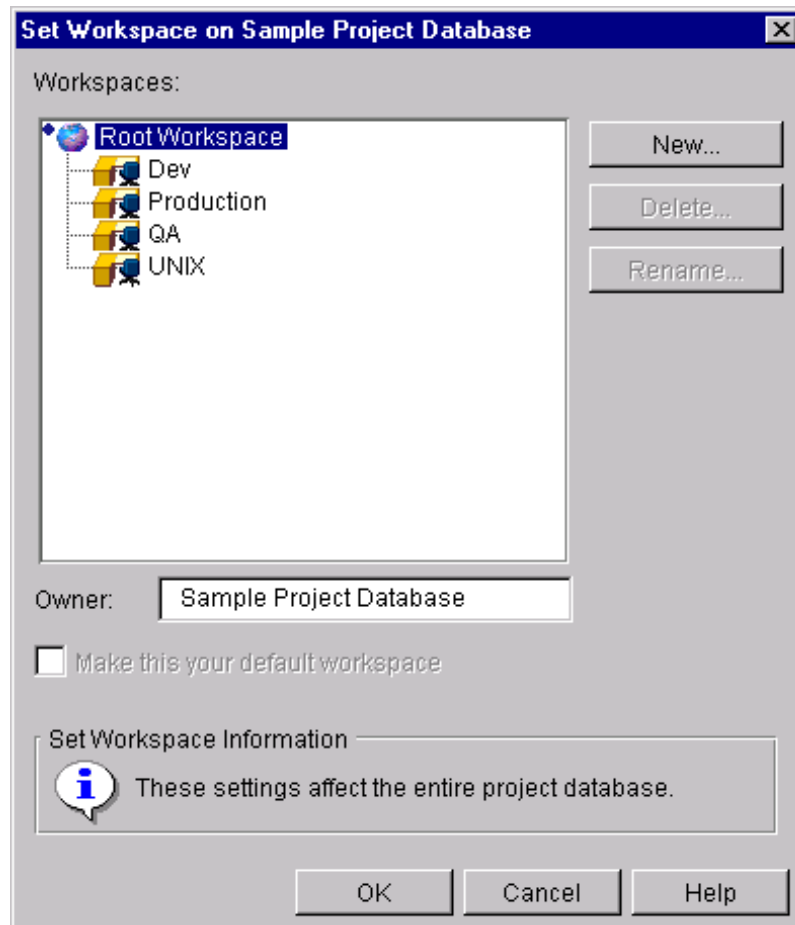
Using Subprojects to Organize Files

Organize versioned files like workfile directories

Using subprojects, you can set up a project database in Version Manager that matches the actual working directories on your operating system. The sample project database, shown below, demonstrates this new subproject concept.
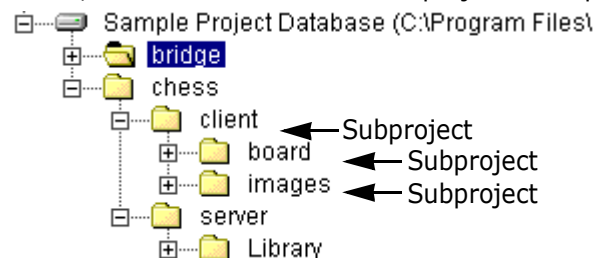
You can use subprojects to organize related versioned files and other subprojects. For more information about projects and subprojects, refer to the *User's Guide*.

Defining Archive Locations

Organize archive
directories to
match workfile
directories

Archive locations are created on your operating system by Version Manager automatically, relative to the location in which you created the project database. By default, Version Manager creates an archive structure that mirrors your project database. For example, if this is your project database structure:
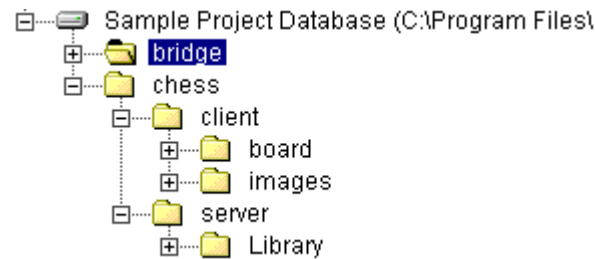
Then this is the corresponding archive directory structure
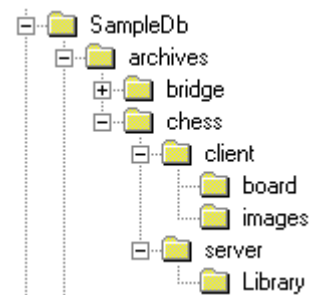
Because the archive structure in project databases supports nested directories, it's possible to use the exact same structure of your workfile directories for your project database and archive locations. By default, Version Manager automatically creates the structure for you.

For more information about creating project databases and defining archive locations, refer to the *Administrator's Guide*.

Multiple Project Selection

Version Manager supports the multiple selection of projects in a project database and multiple selection of folders in a 5.3/6.0 project root. Multiple projects or folders can be selected in the project pane if they all have the same parent project or parent project database. Multiple projects may not be selected in a 5.3/6.0 project root.

To select more than one project/folder, click the items while holding down the SHIFT key to select a series of items, or click the items while holding down the CTRL key to select non-adjacent items.

When multiple projects/folders are selected, all of the files contained within each selected item are shown in the file pane. A header line identifying the project/folder that the files are contained in is displayed for each item selected, followed by the files contained within the item. If you have the recursive file filter turned on (View | Filter | Recursive), then all of the subprojects and their files are also displayed.

Within the file pane, you can select a subset of the files to perform an action on. The selected files do not need to have the same parent project/folder. For more information about multiple project selection, refer to the *User's Guide*.

Working with Version Manager 5.3/6.0 Project Roots

You can perform a subset of Version Manager functionality when working with 5.3/6.0 project roots. For more information about working with Version Manager 5.3/6.0 project roots, see the *User's Guide*.

Using the Command-Line Interface

The project command-line interface lets you work with project databases and projects using a command-line interface in addition to using the Version Manager desktop client. For more information, refer to the next section "Using the Project Command-Line Interface with Version Manager."

The command-line interface works with archives directly. Therefore, function-related features that do not exist in the command-line interface, such as workspaces, projects, and project databases, are not supported. The command-line interface can be used with the PCLI to access these features.

To ensure that the same default archive suffix template is used for any archive that you access via both the desktop client and command-line interfaces, you must use the same configuration file regardless of which interface you use to modify an archive. Information about these issues is provided in this section.

Using the Project Command-Line Interface (PCLI)

The Version Manager project command-line interface (PCLI) allows you to work with project databases and projects using a command-line interface in addition to using the Version Manager graphical user interface. For example, using the PCLI, you can:

- Create project databases, projects, and workspaces.

- Set workfile locations, archive locations, and configuration files.

- Get the current workfile locations, archive locations, and configuration file names and locations.

- List project databases, versioned files, and the attributes of various entities.

- Add workfiles to projects.

- Import archives to projects.

Workspace Paths

In PCLI project commands, you can specify a workspace. For example, the workfile location for a project can vary depending on which workspace is set. Therefore, you may want to specify the workspace from which to obtain the workfile location. The default is the user's default workspace.

Direct Versus Batched Execution

Direct execution   You can execute PCLI commands one at a time by typing the command directly at the command line.

Batched execution   To speed multi-step and repetitive operations, you can write PCLI scripts that execute a sequence of PCLI commands. You can then execute all of the commands in the script by typing one command.

Using the Command-Line Interface with the PCLI

You can use commands provided in the PCLI in combination with the commands provided in the command-line interface (CLI). The PCLI uses a project path to identify files; the CLI uses both the workfile location and the archive location to identify files. For more information on using the PCLI, see the *PCLI User's Guide and Reference*.

Using the Version Manager IDE Client

The Version Manager IDE client allows you to access Version Manager projects from within numerous integrated development environments (IDEs). Using the IDE client, you can perform Version Manager functions, such as checking files in and out of archives, assigning version labels, and generating reports, without leaving your development application.

The Version Manager IDE client is certified for use with a number of IDEs, including Microsoft Visual C++, Microsoft Visual Basic, Microsoft Visual InterDev, Sybase PowerBuilder, Macromedia, Allaire ColdFusion Studio, and Microsoft FrontPage. Supported functions vary across IDEs. See the *IDE Client Implementation Guide*.

Working with IDE Client Projects in Version Manager

Follow these guidelines when working with a Version Manager IDE client project from the Version Manager desktop client rather than from an IDE:

- You can check in, check out, lock, unlock, label, and force a branch on a file. These changes will be available in the project when you open it from your IDE.

- You can create new Version Manager projects and subprojects, and then associate your development projects with the new Version Manager projects from within your IDE.

- You cannot modify workfile locations for IDE client projects. When you add a project to source control from within your IDE, the location of your project files determines the Version Manager workfile location.

Working in Development Applications

When you perform any of the functions listed below from a development application, you can view the changes in the Version Manager desktop client when you open the project database that contains the IDE client projects:

- Add new projects.

- Add new files.

- Get, check in, check out, and undo check out.

- Assign, rename, or delete a version label.

- Increment a revision number.

- Force a branch.

Using the Version Manager Web Client

The Version Manager web client is an optional web browser interface to Version Manager, which is enabled when the Version Manager administrator installs and configures the Version Manager Server. The Version Manager web client works with Version Manager project databases and 5.3/6.0 project roots. With the Version Manager web client, users can perform daily tasks such as getting revisions, checking out/in files, adding workfiles, assigning version labels, and setting workspaces.

You must use Version Manager to perform all administrative tasks on project databases or roots. Administrative tasks can include creating and configuring project databases or roots, defining security, and setting up promotion models.

Using WebDAV Server

WebDAV Server allows you to automatically check out and check in files from within a supported WebDAV client, such as Microsoft Office or Macromedia Dreamweaver. Implementing and extending the WebDAV protocol, WebDAV Server prevents work from being lost by saving each new version of a file as a revision in Version Manager. You can access previous revisions through Version Manager desktop client or any of the other Version Manager interfaces.

Using a supported WebDAV client, you can perform these tasks on the project database configured to work with WebDAV Server:

- Get, check out, check in, and add files.

- Create new projects.

- Delete, copy, move, and rename versioned files and projects.

Index