

CodeWarriorTM

Development Tools

mwCVS Plug-in User

Guide

Revised 2002/8/12



Metrowerks, the Metrowerks logo, and CodeWarrior are registered trademarks of Metrowerks Corp. in the US and/or other countries. All other tradenames and trademarks are the property of their respective owners.

Copyright © Metrowerks Corporation. 2002. ALL RIGHTS RESERVED.

The reproduction and use of this document and related materials are governed by a license agreement media, it may be printed for non-commercial personal use only, in accordance with the license agreement related to the product associated with the documentation. Consult that license agreement before use or reproduction of any portion of this document. If you do not have a copy of the license agreement, contact your Metrowerks representative or call 800-377-5416 (if outside the US call +1 512-997-4700). Subject to the foregoing non-commercial personal use, no portion of this documentation may be reproduced or transmitted in any form or by any means, electronic or mechanical, without prior written permission from Metrowerks.

Metrowerks reserves the right to make changes to any product described or referred to in this document without further notice. Metrowerks makes no warranty, representation or guarantee regarding the merchantability or fitness of its products for any particular purpose, nor does Metrowerks assume any liability arising out of the application or use of any product described herein and specifically disclaims any and all liability. **Metrowerks software is not authorized for and has not been designed, tested, manufactured, or intended for use in developing applications where the failure, malfunction, or any inaccuracy of the application carries a risk of death, serious bodily injury, or damage to tangible property, including, but not limited to, use in factory control systems, medical devices or facilities, nuclear facilities, aircraft or automobile navigation or communication, emergency systems, or other applications with a similar degree of potential hazard.**

USE OF ALL SOFTWARE, DOCUMENTATION AND RELATED MATERIALS ARE SUBJECT TO THE METROWERKS END USER LICENSE AGREEMENT FOR SUCH PRODUCT.

How to Contact Metrowerks

Corporate Headquarters	Metrowerks Corporation 9801 Metric Blvd. Austin, TX 78758 U.S.A.
World Wide Web	http://www.metrowerks.com
Ordering & Technical Support	Voice: (800) 377-5416 Fax: (512) 997-4901

Table of Contents

1	mwCVS Overview and Setup	5
	Differences Between GNU CVS and mwCVS	5
	System Requirements	6
	Learning More About CVS	6
	mwCVS Installation	7
	Turning On mwCVS	9
	Turning Off mwCVS.	10
2	Configuring mwCVS	11
	mwCVS Panel	11
	mwCVS Panel Options	12
	Make New working files read-only	12
	Use gZip level 9 compression	12
	Command Output	13
3	Using mwCVS	15
	Project Window.	15
	Context Menus	17
	IDE Version Control System (VCS) Commands	18
	Synchronize Status	18
	Synchronize Selection	18
	File Commands	19
	Update	19
	Commit	22
	Status	24
	Difference	25
	Tag	28
	Add	30
	Remove	31
	Log	33
	Login	35

Table of Contents

Project Commands	36
Update	36
Commit	39
Status	41
Tag	42
Add	45
Remove	46
Log	48
Recursive Commands	50
Update	51
Commit	54
Checkout	56
Status	60
Tag	61
Log	64
4 mwCVS Tutorial	67
Part One: Turning on mwCVS	68
Part Two: Accessing the Repository	69
Windows	69
Macintosh	70
Part Three: Checking out a Module	72
Part Four: Modifying a File	75
Part Five: Committing the Modified File	78
Index	81

mwCVS Overview and Setup

This chapter is an overview of the Metrowerks™ Concurrent Versions System (mwCVS) plug-in and its setup. Use mwCVS together with the CodeWarrior™ IDE to perform CVS operations.

This chapter has these topics:

- [“Differences Between GNU CVS and mwCVS” on page 5](#)
- [“System Requirements” on page 6](#)
- [“Learning More About CVS” on page 6](#)
- [“mwCVS Installation” on page 7](#)
- [“Turning On mwCVS” on page 9](#)
- [“Turning Off mwCVS” on page 10](#)

The mwCVS plug-in adds some GNU is Not UNIX (GNU) CVS commands to the CodeWarrior IDE. This plug-in implements a command-line wrapper for common CVS commands. You can use mwCVS in addition to using a separate CVS application to issue those commands.

Differences Between GNU CVS and mwCVS

Most of the mwCVS implementation of CVS commands closely follows the GNU CVS implementation. The mwCVS implementation of the `remove` command, however, differs from the GNU CVS implementation:

- mwCVS `remove` deletes the file on disk (local file) as part of executing the command (`remove -f`). This behavior differs from the default GNU CVS implementation, which does not remove the local file as part of executing the command (`remove`).

- mwCVS `remove` commits the file deletion as part of executing the command. This behavior differs from the default GNU CVS implementation, where you must explicitly commit removed files. mwCVS automatically commits the file removal.

System Requirements

Your computer system must meet these minimum requirements before you can use the mwCVS plug-in with the IDE:

- CodeWarrior IDE version 5.0 or higher
- On a Windows host: Microsoft® Windows® XP, Windows® 2000, Windows NT®, or Windows® 98 operating system
- On a Macintosh host: Apple® Mac® OS X 10.1.3 or higher operating system

Using mwCVS with the IDE involves adding files to your CodeWarrior installation. See [“mwCVS Installation” on page 7](#) for more information.

NOTE Some versions of the CodeWarrior IDE ship with the mwCVS plug-in and its support files pre-installed.

Learning More About CVS

This *mwCVS Plug-in User Guide* explains how to use the mwCVS plug-in with the CodeWarrior IDE. If you are not familiar with general CVS concepts, consult the web sites listed in this section to learn more about CVS.

For information about frequently asked questions, documentation, and mailing lists for CVS, see this site:

<http://www.cvshome.org/>

For information about graphical user-interface (GUI) tools for CVS, see this site:

<http://www.cvsgui.org/>

For information about mwCVS downloads and updates, see this site:

<http://www.metrowerks.com/MW/Develop/Desktop/mwCVS.htm>

For information about mwCVS bug-report submissions, see this site:

<http://www.metrowerks.com/MW/Support/Contact+Support/bug.htm>

mwCVS Installation

This section explains how to install the mwCVS plug-in and its support files into a CodeWarrior installation. [Table 1.1](#) lists these files by host.

NOTE Some versions of the CodeWarrior IDE ship with the mwCVS plug-in and its support files. If you upgrade mwCVS, however, follow the instructions in this section to ensure that you upgrade all files correctly.

Table 1.1 mwCVS files

Host	mwCVS Files
Windows	<ul style="list-style-type: none">• mwCVS.dll• pMwCVS.dll• cvs.exe• cwspawn.exe
Macintosh	<ul style="list-style-type: none">• mwCVS• mwCVS Panel• CVSAdapter.bundle

Installing mwCVS on a Windows Host

Follow these steps to install mwCVS and its support files:

- 1 Find your CodeWarrior directory. Subsequent steps in this task list paths relative to this directory.

For example, if you installed CodeWarrior at `C:\Program Files\Metrowerks\CodeWarrior\`, then this path represents your CodeWarrior directory.

TIP On most Windows hosts, the CodeWarrior installer places the CodeWarrior directory in the path `C:\Program Files\Metrowerks\CodeWarrior\`.

- 2 Place the file `mwcvcs.dll` in this path of your CodeWarrior directory:
`Bin\Plugins\version control\`

For example, if your CodeWarrior directory is `C:\Program Files\Metrowerks\CodeWarrior\`, then place `mwcvcs.dll` in
`C:\Program Files\Metrowerks\CodeWarrior\Bin\Plugins\version control\`.

- 3 Place the file `pMWCVS.dll` in this path of your CodeWarrior directory:
`Bin\Plugins\Preference Panel\`
- 4 Place the file `cvs.exe` in this path of your CodeWarrior directory:
`(Helper Apps)\`
- 5 Place the file `cwspawn.exe` in this path of your CodeWarrior directory:
`(Helper Apps)\`

Installing mwCVS on a Macintosh Host

NOTE The Mac OS X version of mwCVS does not include its own version of `cvs`, like the Windows version. Instead, mwCVS for Mac OS X uses the default CVS shell application `/usr/bin/cvs`.

Follow these steps to install mwCVS and its support files:

- 1 Find your CodeWarrior directory. Subsequent steps in this task list paths relative to this directory.

For example, if you installed CodeWarrior at
`StartupDiskName:Applications:Metrowerks CodeWarrior Version Number:Metrowerks CodeWarrior:`, then this path represents your CodeWarrior directory. *StartupDiskName* is the name of the startup disk of your computer and *VersionNumber* is the version number of the CodeWarrior software,

TIP On most Macintosh hosts, the CodeWarrior installer places the CodeWarrior directory in the path
`StartupDiskName:Applications:Metrowerks CodeWarrior VersionNumber:Metrowerks CodeWarrior:.`

- 2 Place the file mwCVS in this path of your CodeWarrior directory:
CodeWarrior Plugins:Version Control:

For example, if your CodeWarrior directory is

`StartupDiskName:Applications:Metrowerks CodeWarrior Version
Number:Metrowerks CodeWarrior:`, then place mwcvS in
`StartupDiskName:Applications:Metrowerks CodeWarrior Version
Number:Metrowerks CodeWarrior:CodeWarrior Plugins:
Version Control:.`

- 3 Place the file mwcvS Panel in this path of your CodeWarrior directory:
CodeWarrior Plugins:Preference Panels:
- 4 Place the file CVSAdapter.bundle in your CodeWarrior directory, at the same level as the CodeWarrior IDE application.

Turning On mwCVS

Follow these steps to turn on mwCVS in the IDE:

- 1 Choose **Edit > Version Control Settings**.
The **VCS Settings** window appears.
- 2 Select **VCS Setup** in the VCS Settings Panels list.
The VCS Setup panel appears.

NOTE Only two options in the VCS Setup panel apply to mwCVS operation:
Use Version Control and **Method**.

- 3 Select the **Use Version Control** option.
- 4 Select **mwCVS** from the **Method** list box.
The mwCVS item appears in the VCS Settings Panels list.
- 5 Click **Apply** or **Save** to confirm your changes.
- 6 Close the VCS Settings window.

Enabling mwCVS causes the VCS menu to appear in the menu bar. You use this menu to issue CVS commands from the CodeWarrior IDE.



This icon represents the VCS menu on a Macintosh host.

Turning Off mwCVS

Follow these steps to turn off mwCVS in the IDE:

- 1 Choose **Edit > Version Control Settings**.

The **VCS Settings** window appears.

- 2 Select **VCS Setup** in the VCS Settings Panels list.

The VCS Setup panel appears.

- 3 Select **None** from the **Method** list box.

The mwCVS item disappears from the VCS Settings Panels list.

- 4 Clear the **Use Version Control** option.

- 5 Click **Apply** or **Save** to confirm your changes.

- 6 Close the VCS Settings window.

Disabling mwCVS causes the VCS menu to disappear from the menu bar.

Configuring mwCVS

This chapter explains how to configure the options in the mwCVS panel. The options correspond to GNU CVS command-line switches. These options determine how mwCVS operates with the CodeWarrior™ IDE.

This chapter has these topics:

- [“mwCVS Panel” on page 11](#)
- [“mwCVS Panel Options” on page 12](#)

NOTE The **VCS Setup** panel in the **VCS Settings** window has two options for configuring mwCVS: **Use Version Control** and **Method**. See [“Turning On mwCVS” on page 9](#) for more information about these options. The other options in the VCS Setup panel do not apply to configuring mwCVS.

mwCVS Panel

Use the **mwCVS** panel, shown in [Figure 2.1](#), to configure default mwCVS options. This panel appears in the VCS Settings window after you turn on mwCVS in the **VCS Setup** panel. [Table 2.1 on page 12](#) briefly explains the panel options. See [“mwCVS Panel Options” on page 12](#) for more detailed explanations.

Figure 2.1 mwCVS panel

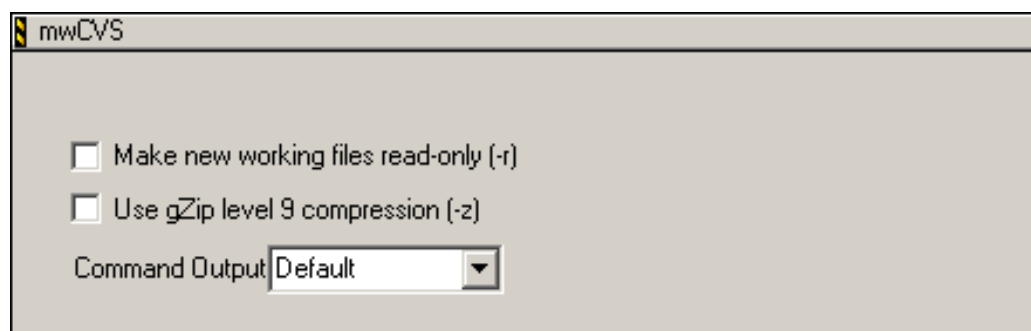


Table 2.1 mwCVS panel options

Option	Explanation
Make New working files read-only (-r)	Select to prevent changes to new working files that you check out from the CVS repository.
Use gZip level 9 compression (-z9)	Select to use gZip level 9 (high compression) for files transmitted between the IDE and the CVS repository.
Command Output	Select the level of information that you want mwCVS to output for CVS operations.

mwCVS Panel Options

This section explains in detail the options in the mwCVS panel. These options configure the default settings for various CVS operations.

Make New working files read-only

Select this option to make the file read-only on the disk after issuing the CVS command. Clear this option to make the file read/write.

Selecting this option is equivalent to using the GNU CVS command-line switch `-r`. The option also overrides the `CVSREAD` environment variable.

Use this option to manage changes to source code that you retrieve from the CVS repository. For example, select this option to preserve the files that you retrieve from the repository so that you cannot easily change them.

Use gZip level 9 compression

Select this option to compress files using gZip level 9 during transmission between the IDE and the CVS repository. Clear this option to transmit files without using compression.

Selecting this option is equivalent to using the command-line switch `-z9`.

Use this option to manage network and bandwidth issues when transmitting data between your computer and the CVS repository. For example, select this option to reduce file size and shorten transmission time.

Command Output

Select from this list box the level of information that you want mwCVS to output during CVS operations. The output appears in an editor window.

Output choices include:

- **Default**—Select this option to output informational messages.
- **Quiet**—Select this option to output non-informational messages. This option is equivalent to the command-line switch `-q`.
- **Very Quiet**—Select this option to output only messages for serious errors. This option is equivalent to the command-line switch `-Q`.

Using mwCVS

With the mwCVS plug-in installed, configured, and started, the CodeWarrior™ IDE's version-control features use GNU CVS. This chapter explains how the IDE works with GNU CVS.

This chapter has these topics:

- [“Project Window” on page 15](#)
- [“Context Menus” on page 17](#)
- [“IDE Version Control System \(VCS\) Commands” on page 18](#)
- [“File Commands” on page 19](#)
- [“Project Commands” on page 36](#)
- [“Recursive Commands” on page 50](#)

Project Window

The project window in the CodeWarrior IDE displays CVS status icons after you turn on mwCVS in the VCS Settings window. These icons indicate the status of each file in the project.

[Figure 3.1 on page 16](#) shows a sample project window with CVS status icons. [Table 3.1](#) explains these icons.

Table 3.1 CVS status icons




Icon	File is in the CVS Repository	File is not in the CVS Repository	File is Writable	File is Read Only	File has changed since the last Update	File has not changed since the last Update
		.	.		Not applicable	Not applicable
		.		.	Not applicable	Not applicable
	.			.		.

Table 3.1 CVS status icons (*continued*)



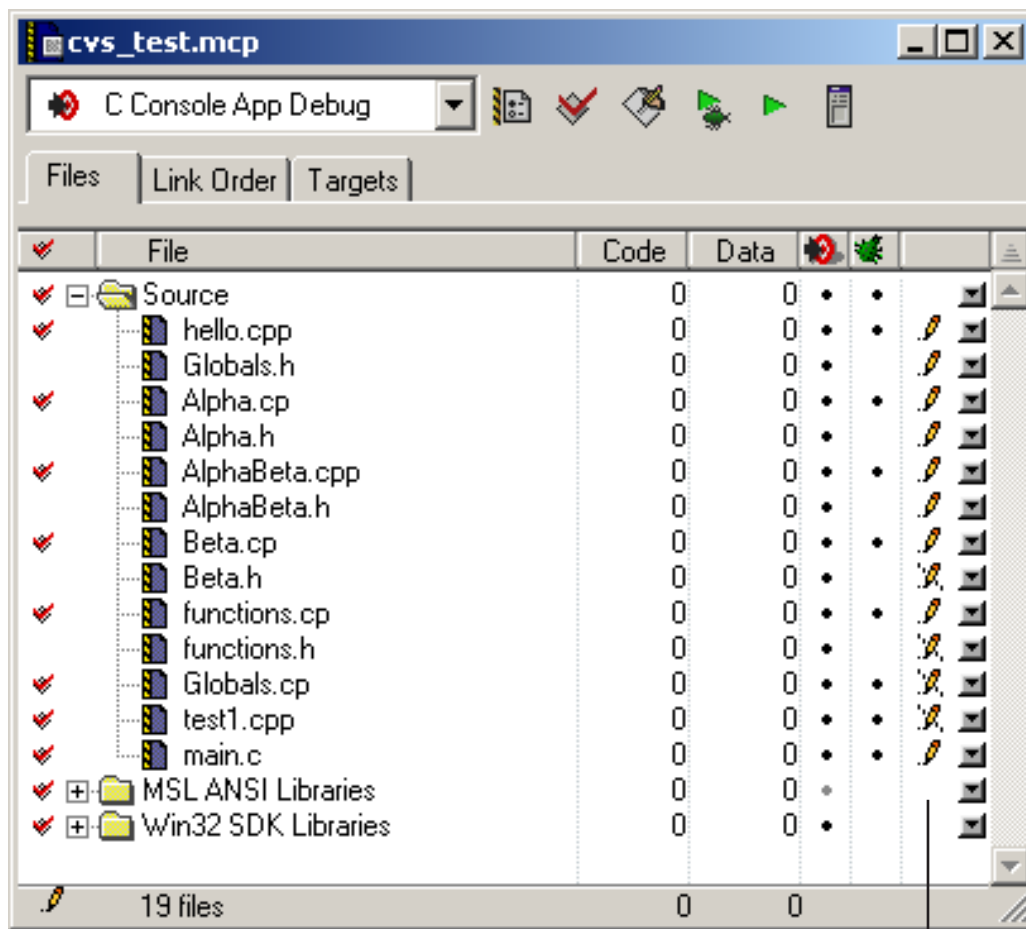
Icon	File is in the CVS Repository	File is not in the CVS Repository	File is Writable	File is Read Only	File has changed since the last Update	File has not changed since the last Update
	.		.			.
	.		.		.	

Figure 3.1 CVS status icons in the project window



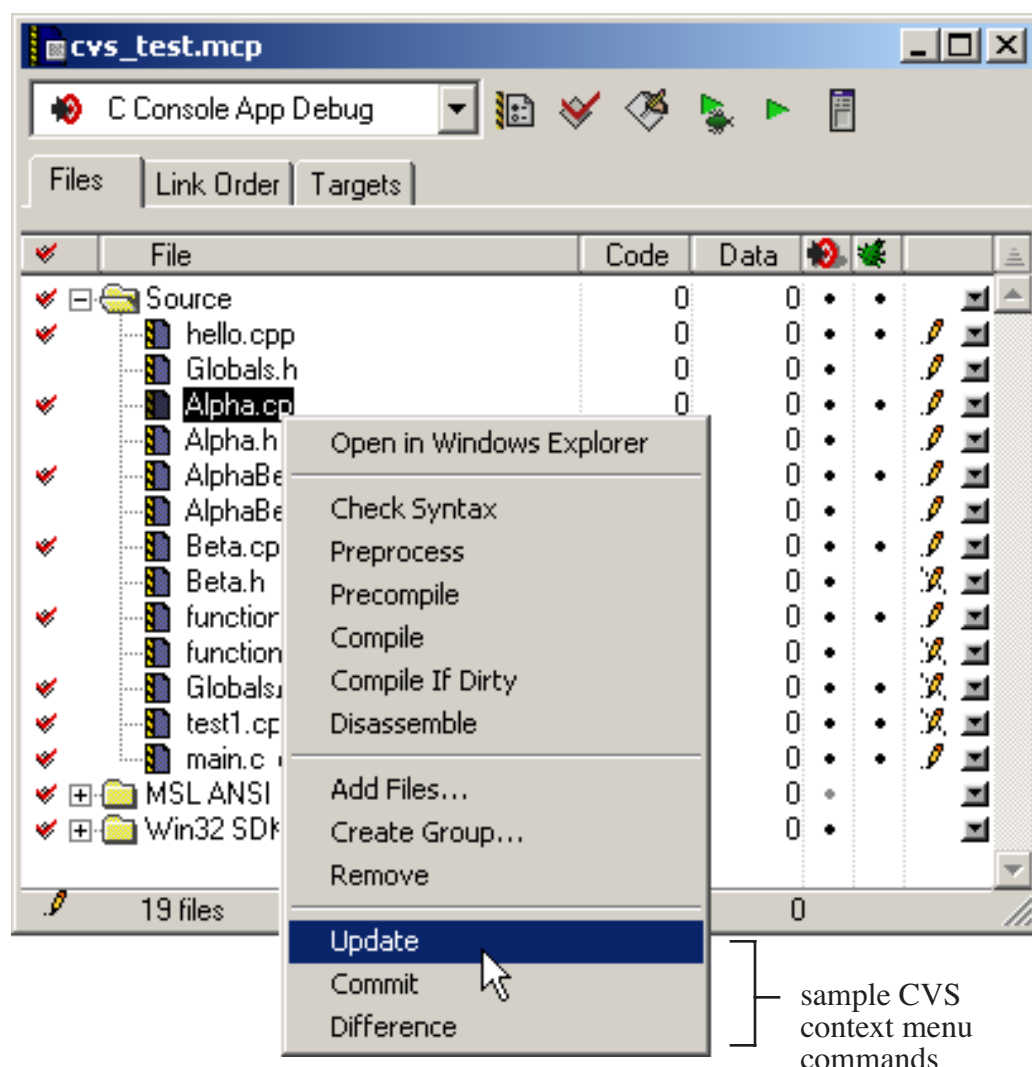
CVS status icons

Context Menus

mwCVS adds CVS commands to the IDE context menu. The commands that appear in the context menu depend on the items that you select. Use these CVS context-menu commands as shortcuts for common commands in the VCS menu.

[Figure 3.2](#) shows a sample IDE context menu that has CVS commands.

Figure 3.2 Sample CVS context menu in the project window



IDE Version Control System (VCS) Commands

This section explains the IDE commands for version-control systems.

Synchronize Status

Select **VCS > Synchronize Status** to refresh the CVS status icons that appear in the project window. After you select this command, mwCVS analyzes CVS information about each file in your project. It compares this information against the information stored in the CVS data folder that resides in your project folder. The IDE uses the results of these comparisons to update each status icon in the project window.

After you select this command, a progress indicator shows synchronization progress.

See [Table 3.1 on page 15](#) for more information about the CVS status icons that appear in the project window.

Synchronize Selection

NOTE This command does not appear in the Macintosh-hosted IDE.

Select **VCS > Synchronize Selection** to refresh the CVS status icon of the selected file in the project window. After you select this command, mwCVS analyzes CVS information about the selected file. It compares this information against the information stored in the CVS data folder that resides in your project folder. The IDE uses the results of this comparison to update the status icon in the project window.

After you select this command, a progress indicator shows synchronization progress.

See [Table 3.1 on page 15](#) for more information about the CVS status icons that appear in the project window.

File Commands

This section explains the mwCVS commands for files.

Update

Select a file in the project window, then select **VCS > Update** to display the Update dialog box. [Figure 3.3](#) shows the Update dialog box for a file. [Table 3.2 on page 20](#) explains the dialog-box options.

After you click **Update**, a progress indicator appears.

NOTE You can change the default settings for the [Read-Only](#) and [Command Output](#) options. Use the **mwCVS** panel in the **VCS Settings** window to configure the default settings. See [“mwCVS Panel Options” on page 12](#) for more information.

Figure 3.3 Update dialog box (for files)

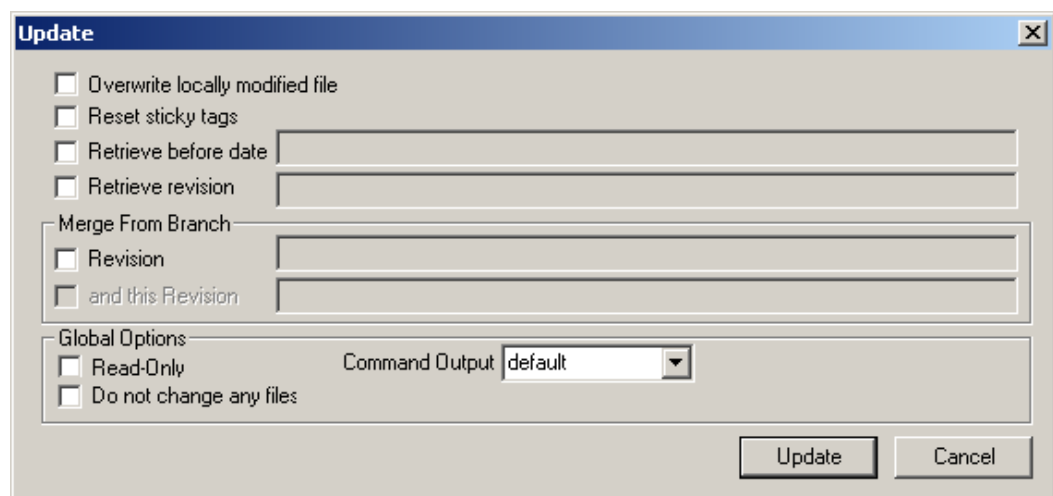


Table 3.2 **Update dialog box (for files)—options**

Option	Explanation
Overwrite locally modified file	<p>Select to replace a locally modified file with the version stored on the CVS repository. Clear to preserve the locally modified file.</p> <p>Selecting this option is equivalent to using the GNU CVS command-line switch <code>-C</code>.</p>
Reset sticky tags	<p>Select to remove any sticky tags associated with the selected file after the IDE completes the update operation. Clear to preserve existing sticky tags.</p> <p>Selecting this option is equivalent to using the command-line switch <code>-A</code>.</p>
Retrieve before date	<p>Select to enter in the corresponding field the date before which you want to retrieve a version of the file in the CVS repository. Clear to disregard date information.</p> <p>Enter the date according to GNU CVS conventions. Sample dates are:</p> <ul style="list-style-type: none">• 1 month ago• last Monday• 3/31/02 10:00:04 PST <p>Selecting this option is equivalent to using the command-line switch <code>-D <i>Date</i></code>, where <i>Date</i> is the date to use for this operation.</p>
Retrieve revision	<p>Select to enter in the corresponding field the revision tag of the file that you want to retrieve from the CVS repository. Clear to disregard revision-tag information.</p> <p>Selecting this option is equivalent to using the command-line switch <code>-r <i>NumberOrTag</i></code>, where <i>NumberOrTag</i> is the revision number or tag to use for this operation.</p>

Table 3.2 **Update dialog box (for files)—options (*continued*)**

Option		Explanation
Merge From Branch	Revision	<p>Select to enter in the corresponding field the revision tag or branch name of the file that you want to retrieve from the CVS repository. The IDE retrieves this file and merges its contents with the contents of your local file. Clear to preserve the contents of the local file.</p> <p>Selecting this option is equivalent to using the command-line switch <code>-j NumberOrTag</code>.</p>
	and this Revision	<p>Select to enter in the corresponding field a revision tag or branch name that, together with the Revision entry, specifies a version range over which to merge file contents. Clear to preserve the contents of the local file.</p> <p>The IDE interprets the combination of Revision and this Revision differently from Revision alone. CVS merges into the local file any changes between the version specified by Revision and the version specified by and this Revision.</p> <p>Selecting this option is equivalent to using the command-line switch <code>-j NumberOrTag -j NumberOrTag</code>.</p>
Read-Only		<p>Select to make the file read-only on the disk after issuing the Update command. Clear to make the file read/write.</p> <p>This option supersedes the setting in the mwCVS panel of the VCS Settings window.</p> <p>Selecting this option is equivalent to using the command-line switch <code>-r</code>. Clearing this option is equivalent to using the command-line switch <code>-w</code>.</p>

Table 3.2 **Update dialog box (for files)—options (*continued*)**

Option	Explanation
Do not change any files	Select to see the result of the update operation without modifying the CVS repository. Clear to modify the CVS repository to reflect the update operation. Selecting this option is equivalent to using the command-line switch <code>-n</code> .
Command Output	Select the level of information that you want mwCVS to output during CVS operations: <ul style="list-style-type: none">• Default—output informational messages• Quiet—output non-informational messages (<code>-q</code>)• Very Quiet—output only messages for serious errors (<code>-Q</code>) This option supersedes the setting in the mwCVS panel of the VCS Settings window.

Commit

Select a file in the project window, then select **VCS > Commit** to open the Commit dialog box. [Figure 3.4 on page 23](#) shows the Commit dialog box for a file. [Table 3.3 on page 23](#) explains the dialog-box options.

After you click **Commit**, a progress indicator shows commit progress.

NOTE You can change the default settings for the [Read-Only](#) and [Command Output](#) options. Use the **mwCVS** panel in the **VCS Settings** window to configure the default settings. See [“mwCVS Panel Options” on page 12](#) for more information.

Figure 3.4 Commit dialog box (for files)

Table 3.3 Commit dialog box (for files)—options

Option	Explanation
Log message	<p>Enter a log message for the file that you will commit.</p> <p>Entering a message in this field is equivalent to using the GNU CVS command-line switch <code>-m Text</code>, where <i>Text</i> is the log message to use for this operation.</p>
Commit to revision	<p>Select to enter in the corresponding field a numeric or symbolic revision tag to which you want to commit the file. Clear to commit the file to the latest revision.</p> <p>Selecting this option is equivalent to using the command-line switch <code>-r NumberOrTag</code>, where <i>NumberOrTag</i> is the revision number or tag to use for this operation.</p>
Read-Only	<p>Select to make the file read-only on the disk after issuing the Commit command. Clear to make the file read/write.</p> <p>This option supersedes the setting in the mwCVS panel of the VCS Settings window.</p> <p>Selecting this option is equivalent to using the command-line switch <code>-r</code>. Clearing this option is equivalent to using the command-line switch <code>-w</code>.</p>

Table 3.3 Commit dialog box (for files)—options (*continued*)

Option	Explanation
Do not change any files	Select to see the result of the commit operation without modifying the CVS repository. Clear to modify the CVS repository to reflect the commit operation. Selecting this option is equivalent to using the command-line switch <code>-n</code> .
Command Output	Select the level of information that you want mwCVS to output during CVS operations: <ul style="list-style-type: none">• Default—output informational messages• Quiet—output non-informational messages (<code>-q</code>)• Very Quiet—output only messages for serious errors (<code>-Q</code>) This option supersedes the setting in the mwCVS panel of the VCS Settings window.

Status

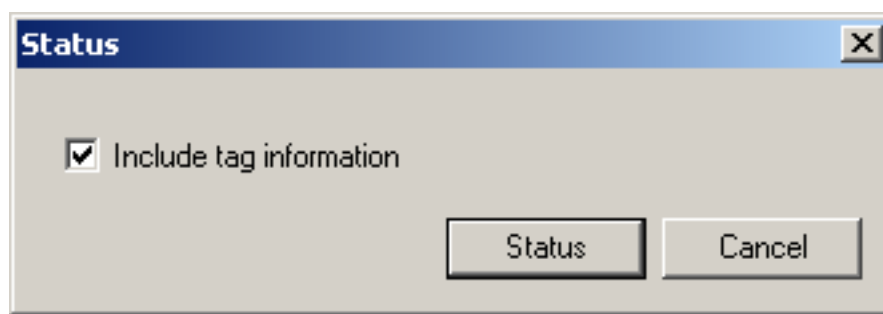
Select a file in the project window, then select **VCS > Status** to open the Status dialog box. [Figure 3.5](#) shows the Status dialog box for a file.

Select **Include tag information** to include in the status report information about tags applied to the file revisions. Clear the option to exclude the tag information from the status report. Selecting the option is equivalent to using the command-line switch `-v`.

Notice that the default mwCVS status behavior is to include tag information, which is equivalent to using the GNU CVS command line `status -v`.

After you click **Status**, a progress indicator appears. After the IDE completes the status operation, an editor window appears. This editor window shows the status information for the selected file.

Figure 3.5 Status dialog box (for files)



Difference

Select a file in the project window, then select **VCS > Difference** to open the Diff dialog box. [Figure 3.6](#) shows the Diff dialog box for a file. [Table 3.4](#) explains the dialog-box options.

After you click **Diff**, a progress indicator shows difference progress. After the IDE completes the difference operation, either the **File Compare Results** window or an editor window appears, depending on your settings. These windows contain the difference information for the selected files.

Figure 3.6 Diff dialog box (for files)

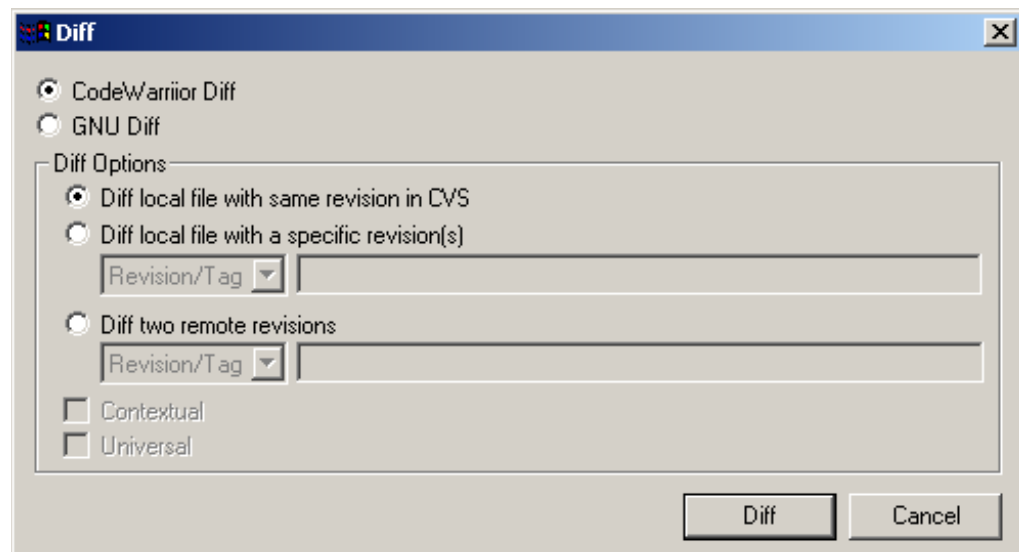


Table 3.4 Diff dialog box (for files)—options

Option	Explanation
CodeWarrior Diff	Select to display difference results in a File Compare Results window. This window graphically depicts difference information. If you select this option, you cannot select the GNU Diff option.
GNU Diff	Select to display difference results in an editor window. The difference information appears in textual form in the editor window. If you select this option, you cannot select the CodeWarrior Diff option. Selecting this option also turns on the Contextual and Universal options.

Table 3.4 **Diff dialog box (for files)—options (*continued*)**

Option	Explanation
Diff local file with same revision in CVS	<p>Select to display difference results between the local working file and the corresponding revision of that file in the CVS repository.</p> <p>If you select this option, you cannot select the Diff local file with a specific revision(s) or Diff two remote revisions options.</p>
Diff local file with a specific revision(s)	<p>Select to enter information to use for the difference operation. Select an item in the list box, then supply information for that item in the corresponding field:</p> <ul style="list-style-type: none">• Revision/Tag—Enter a revision number or tag to use for the difference operation. This option is equivalent to using the command-line switch <code>-r <i>NumberOrTag</i></code>, where <i>NumberOrTag</i> is the revision number or tag to use for this operation.• Date—Enter a date to use for the difference operation. This option is equivalent to using the command-line switch <code>-D <i>Date</i></code>, where <i>Date</i> is the date to use for this operation. <p>The IDE displays difference results between the local working file and the specified revisions of that file in the CVS repository.</p> <p>If you select this option, you cannot select the Diff local file with same revision in CVS or Diff two remote revisions options.</p>

Table 3.4 **Diff dialog box (for files)—options (*continued*)**

Option	Explanation
Diff two remote revisions	<p>Select to enter information to use for the difference operation. In this case, you select two revisions of the file in the CVS repository.</p> <p>Use the first pair of fields under the Diff local file with a specific revision(s) option to specify the first revision that you want to use in the difference operation. Use the second pair of fields under the Diff two remote revisions option to specify the second revision that you want to use in the difference operation.</p> <p>The second pair of fields corresponds to the part of this command-line expression enclosed in braces: <code>[[-r NumberOrTag] [-D Date]]</code> <code>{ [[-r NumberOrTag] [-D Date]] }</code></p> <p>The IDE displays difference results between the specified revisions of the file in the CVS repository.</p> <p>If you select this option, you cannot select the Diff local file with same revision in CVS.</p>
Contextual	<p>Select to generate a contextual-format difference for use with the UNIX <code>patch</code> command. Clear to generate a non-contextual format.</p> <p>Selecting this option is equivalent to using the command-line switch <code>-c</code>.</p> <p>Select the GNU Diff option to turn on the Contextual option.</p>
Universal	<p>Select to generate a universal-format difference. Clear to generate a non-universal format.</p> <p>Selecting this option is equivalent to using the command-line switch <code>-u</code>.</p> <p>Select the GNU Diff option to turn on the Universal option.</p>

Tag

Select a file in the project window, then select **VCS > Tag** to open the Tag dialog box. [Figure 3.7](#) shows the Tag dialog box for a file. [Table 3.5](#) explains the dialog-box options.

After you click **Tag**, a progress indicator shows tag progress.

NOTE You can change the default settings for the [Read-Only](#) and [Command Output](#) options. Use the **mwCVS** panel in the **VCS Settings** window to configure the default settings. See [“mwCVS Panel Options” on page 12](#) for more information.

Figure 3.7 Tag dialog box (for files)

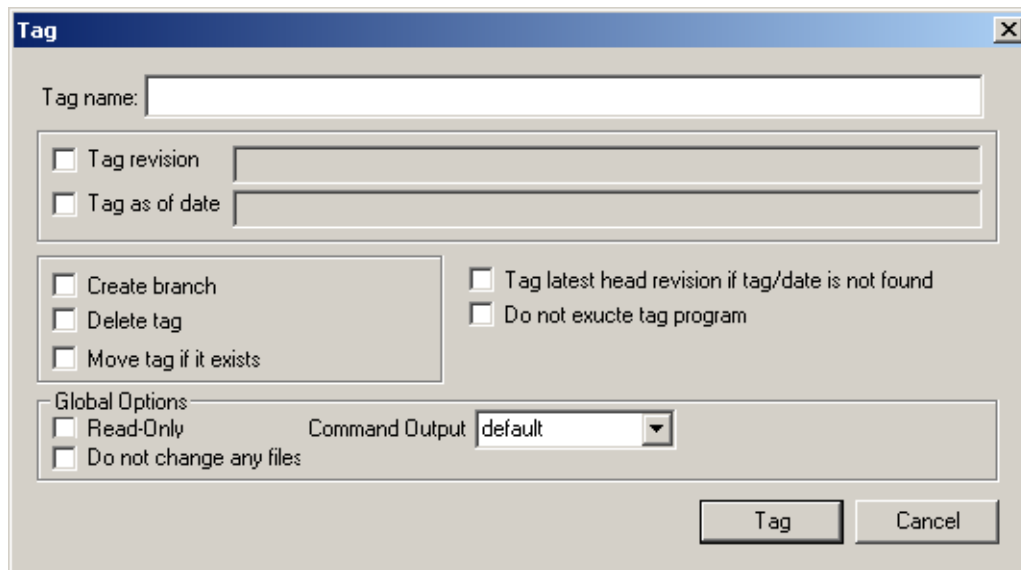


Table 3.5 Tag dialog box (for files)—options

Option	Explanation
Tag name	Enter the name of the tag.
Tag revision	Select to enter in the corresponding field the file revision that you want to tag. Clear to apply the tag to the latest revision. Selecting this option is equivalent to using the GNU CVS command-line switch <code>-r NumberOrTag</code> , where <i>NumberOrTag</i> is the revision number or tag to use for this operation.

Table 3.5 Tag dialog box (for files)—options (*continued*)

Option	Explanation
Tag as of date	<p>Select to enter in the corresponding field the date of the file revision that you want to tag. Clear to apply the tag to the latest revision date.</p> <p>Selecting this option is equivalent to using the command-line switch <code>-D Date</code>, where <i>Date</i> is the date to use for this operation.</p>
Create branch	<p>Select to use the tag to denote a branch of the file revision in the CVS repository. Clear to use the tag to denote a label for a particular file revision.</p> <p>Selecting this option is equivalent to using the command-line switch <code>-b</code>.</p> <p>If you select this option, you cannot select the Delete tag option.</p>
Delete tag	<p>Select to permanently delete the specified tag from the CVS repository. Clear to add the tag to the specified file revision.</p> <p>Selecting this option is equivalent to using the command-line switch <code>-d</code>.</p> <p>If you select this option, you cannot select the Create branch or Move tag if it exists options.</p>
Move tag if it exists	<p>Select to have the IDE relocate existing tags with the same name as the tag you specify. Clear to have the IDE display an error message if it encounters an existing tag with the same name as the tag you specify.</p> <p>Selecting this option is equivalent to using the command-line switch <code>-F</code>.</p> <p>If you select this option, you cannot select the Delete tag option.</p>
Tag latest head revision if tag/date is not found	<p>Select to have the IDE apply the tag to the latest file revision on the CVS repository if it does not encounter a revision with the specified tag or date. Clear to have CVS report an error if it fails to find a matching tag.</p> <p>Selecting this option is equivalent to using the command-line switch <code>-f</code>.</p>

Table 3.5 Tag dialog box (for files)—options (*continued*)

Option	Explanation
Do not execute tag program	Select to bypass running a tag program as part of the tag operation. Clear to run the program. Selecting this option is equivalent to using the command-line switch <code>-n</code> .
Read-Only	Select to make the file read-only on the disk after issuing the Tag command. Clear to make the file read/write. This option supersedes the setting in the mwCVS panel of the VCS Settings window. Selecting this option is equivalent to using the command-line switch <code>-r</code> .
Do not change any files	Select to see the result of the tag operation without modifying the CVS repository. Clear to modify the CVS repository to reflect the tag operation. Selecting this option is equivalent to using the command-line switch <code>-n</code> .
Command Output	Select the level of information that you want mwCVS to output during CVS operations: <ul style="list-style-type: none">• Default—output informational messages• Quiet—output non-informational messages (<code>-q</code>)• Very Quiet—output only messages for serious errors (<code>-Q</code>) This option supersedes the setting in the mwCVS panel of the VCS Settings window.

Add

Select a file in the project window that does not currently reside in the CVS repository, then select **VCS > Add** to open the Add dialog box. [Figure 3.8 on page 31](#) shows the Add dialog box for a file. [Table 3.6 on page 31](#) explains the dialog-box options.

After you click **Add**, a progress indicator appears.

NOTE The current version of GNU CVS, when using the `pserver` connection method, ignores the **Log Message** of the **Add** command. This is a known

bug in GNU CVS. For more information, refer to the file BUGS in the CVS source distribution.

Figure 3.8 Add dialog box (for files)

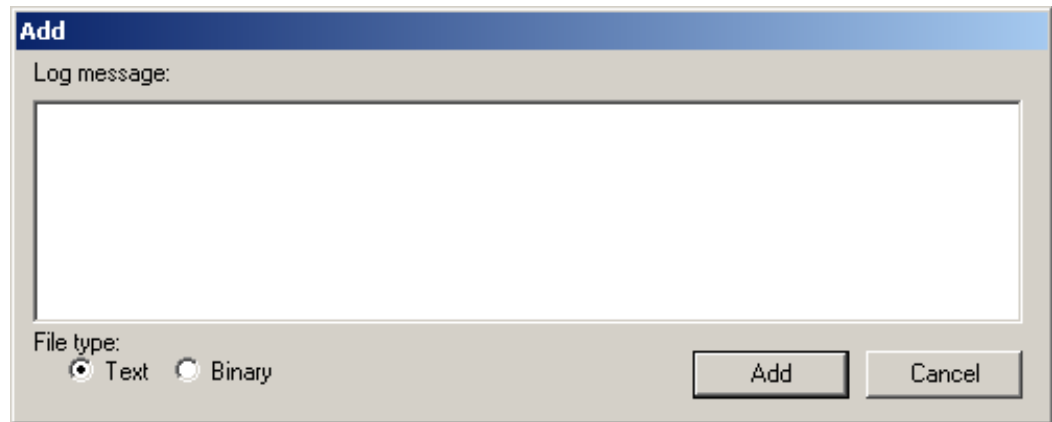


Table 3.6 Add dialog box (for files)—options

Option	Explanation
Log message	Enter in this field a message to log with the file that you will add. This option is equivalent to using the GNU CVS command-line switch <code>-m <i>Text</i></code> , where <i>Text</i> is the log message to use for this operation.
Text	Select this file type if you are adding a text file to the CVS repository.
Binary	Select this file type if you are adding a binary file (non-text file) to the CVS repository. Selecting this option is equivalent to using the command-line switch <code>-kb</code> .

Remove

Select a file in the project window, then select **VCS > Remove** to open the Remove dialog box.

mwCVS implements the `remove` command differently from GNU CVS. The default mwCVS behavior for a remove command is to delete the local file from disk and immediately commit that removal to the CVS repository. This behavior allows the

IDE to commit the file removal even though the file no longer exists on your computer.

[Figure 3.9](#) shows the Remove dialog box for a file. [Table 3.7](#) explains the dialog-box options.

After you click **Remove**, a progress indicator shows remove progress.

Figure 3.9 Remove dialog box (for files)

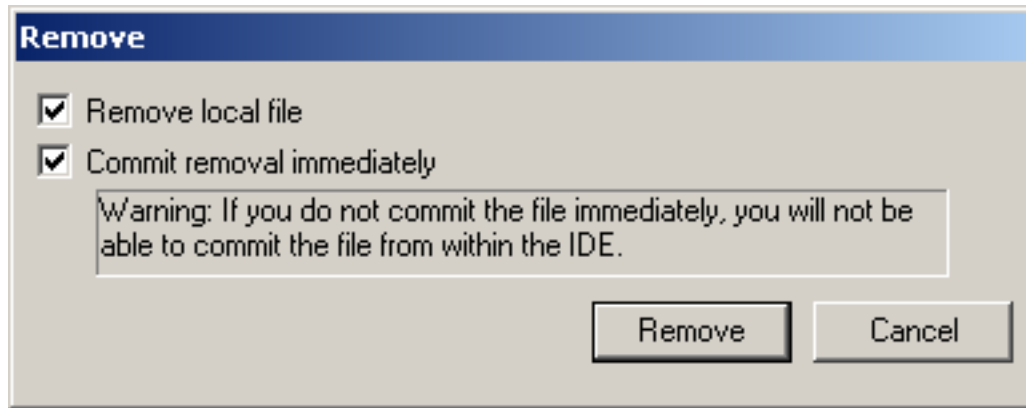


Table 3.7 Remove dialog box (for files)—options

Option	Explanation
Remove local file	Select to delete the local file as part of the mwCVS remove operation. Clear to preserve the local file. Selecting this option is equivalent to using the GNU CVS command-line switch <code>-f</code> .
Commit removal immediately	Select to have the IDE immediately commit the file removal as part of the remove operation. Clear to use a separate CVS application to commit the file removal. If you do not select the Remove local file option, you must use a separate CVS application to commit the file removal.

Log

Select a file in the project window, then select **VCS > Log** to open the Log dialog box. [Figure 3.10](#) shows the Log dialog box for a file. [Table 3.8](#) explains the dialog-box options.

After you click **Log**, a progress indicator appears. After the IDE completes the log operation, an editor window appears. This window shows the log information for the selected file.

Figure 3.10 Log dialog box (for files)

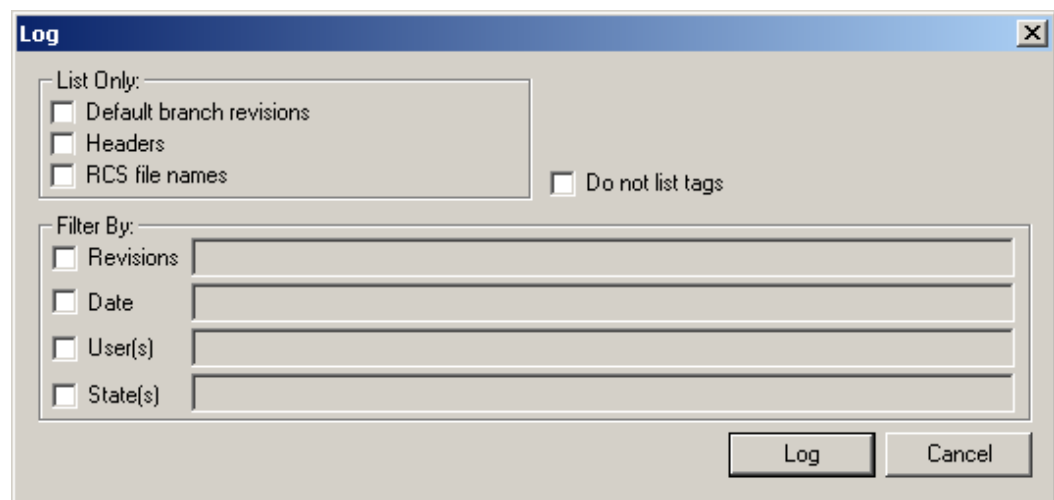


Table 3.8 Log dialog box (for files)—options

Option	Explanation
Default branch revisions	Select to list default branch revisions in the log. Clear to list all branch revisions. Selecting this option is equivalent to using the GNU CVS command-line switch <code>-b</code> .
Headers	Select to list header information in the log. Clear to exclude header information from the log. Selecting this option is equivalent to using the command-line switch <code>-h</code> .

Table 3.8 Log dialog box (for files)—options (*continued*)

Option	Explanation
RCS file names	<p>Select to list file names that comply with Revision Control System (RCS) naming conventions. Clear to exclude RCS file names from the log.</p> <p>Selecting this option is equivalent to using the command-line switch <code>-R</code>.</p>
Do not list tags	<p>Select to exclude tag information from the log. Clear to include the tag information.</p> <p>Selecting this option is equivalent to using the command-line switch <code>-N</code>.</p>
Revisions	<p>Select to enter in the corresponding field the file revisions for which you want to retrieve log information. Clear to retrieve log information for all file revisions.</p> <p>Selecting this option is equivalent to using the command-line switch <code>-r NumberOrTag: NumberOrTag</code>, where <i>NumberOrTag</i> is the revision number or tag to use for this operation.</p>
Date	<p>Select to enter in the corresponding field the dates of file revisions for which you want to retrieve log information. Clear to retrieve log information for all file-revision dates.</p> <p>Selecting this option is equivalent to using the command-line switch <code>-d Date</code>, where <i>Date</i> is the date to use for this operation.</p>
User(s)	<p>Select to enter in the corresponding field the users for whose file revisions you want to retrieve log information. Clear to retrieve log information for all users who revised the file.</p> <p>Selecting this option is equivalent to using the command-line switch <code>-w LoginName1, LoginName2, ...</code>, where <i>LoginName</i> is the login name to use for this operation.</p>
State(s)	<p>Select to enter in the corresponding field the state information of files for which you want to retrieve log information. Clear to retrieve log information for all file states.</p> <p>Selecting this option is equivalent to using the command-line switch <code>-s</code>.</p>

Login

Select **VCS > Login** to open the Login dialog box. [Figure 3.11](#) shows the Login dialog box. [Table 3.9](#) explains the dialog-box options.

NOTE The Macintosh-hosted version of mwCVS does not support the Login dialog box. Instead, use a command-line utility, such as the **Terminal** application, to log into a repository by using the CVS `login` command. After you log in to the repository, you can use the IDE to issue CVS commands.

Figure 3.11 Login dialog box

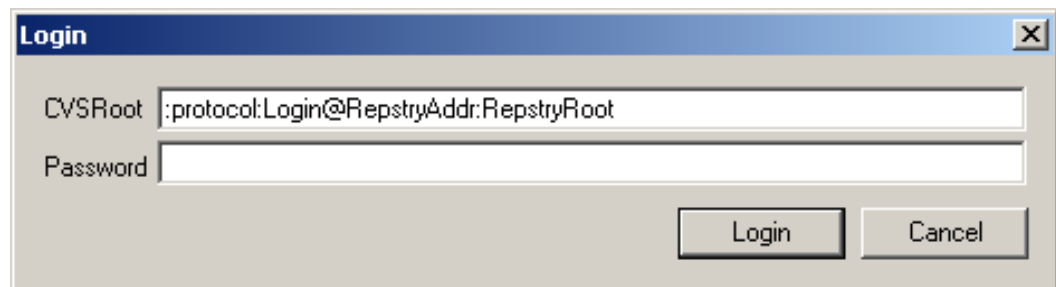


Table 3.9 Login dialog box—options

Option	Explanation
CVSRoot	<p>Enter in this field the root of the remote CVS repository. Use the general construction</p> <p><i>:protocol:Login@RepstryAddr:RepstryRoot</i></p> <p>where</p> <ul style="list-style-type: none">• <i>protocol</i> is the protocol you use to access the repository (such as <code>local</code> or <code>pserver</code>).• <i>Login@RepstryAddr</i> is your login name (<i>Login</i>), followed by <code>@</code>, followed by the Internet Protocol (IP) or Domain Name System (DNS) address of the remote repository (<i>RepstryAddr</i>)• <i>RepstryRoot</i> is the path to the root level of the remote repository <p>The default entry that appears in this field corresponds to the last used CVSRoot entry.</p>
Password	Enter the password that you use to log into the CVS repository.

After you click **Login**, a progress indicator shows login progress. After the IDE completes the login operation, the **VCS Messages** window appears. This window notifies you of the login status. The message (Logging into *server*), where *server* is the name or address of the server, confirms successful login to the CVS repository.

Project Commands

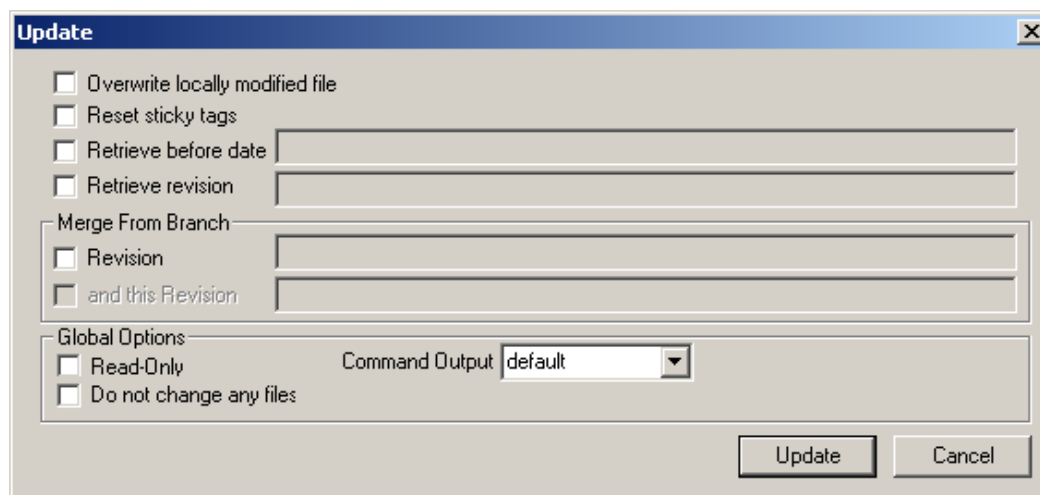
This section explains the mwCVS commands for projects. Most of the project commands appear identical to the file commands, but project commands apply to the current project while the file commands apply to a selected file or group of files. The [Difference](#) command does not apply to projects.

Update

Bring forward a project window, then select **VCS > Project > Update** to open the Update dialog box. [Figure 3.12](#) shows the Update dialog box for a project. [Table 3.10 on page 37](#) explains the dialog-box options.

After you click **Update**, a progress indicator appears.

Figure 3.12 Update dialog box (for projects)



NOTE You can change the default settings for the [Read-Only](#) and [Command Output](#) options. Use the **mwCVS** panel in the **VCS Settings** window to

configure the default settings. See [“mwCVS Panel Options” on page 12](#) for more information.

Table 3.10 Update dialog box (for projects)—options

Option	Explanation
Overwrite locally modified file	<p>Select to replace a locally modified project with the version stored on the CVS repository. Clear to preserve the locally modified project.</p> <p>Selecting this option is equivalent to using the GNU CVS command-line switch <code>-C</code>.</p>
Reset sticky tags	<p>Select to remove any sticky tags associated with the current project after the IDE completes the update operation. Clear to preserve existing sticky tags.</p> <p>Selecting this option is equivalent to using the command-line switch <code>-A</code>.</p>
Retrieve before date	<p>Select to enter in the corresponding field the date before which you want to retrieve a version of the project in the CVS repository. Clear to disregard date information.</p> <p>Enter the date according to GNU CVS conventions. Sample dates are:</p> <ul style="list-style-type: none"> • 1 month ago • last Monday • 3/31/02 10:00:04 PST <p>Selecting this option is equivalent to using the command-line switch <code>-D Date</code>, where <i>Date</i> is the date to use for this operation.</p>
Retrieve revision	<p>Select to enter in the corresponding field the revision tag of the project that you want to retrieve from the CVS repository. Clear to disregard revision-tag information.</p> <p>Selecting this option is equivalent to using the command-line switch <code>-r NumberOrTag</code>, where <i>NumberOrTag</i> is the revision number or tag to use for this operation.</p>

Table 3.10 **Update dialog box (for projects)—options (*continued*)**

Option		Explanation
Merge From Branch	Revision	<p>Select to enter in the corresponding field the revision tag or branch name of the project that you want to retrieve from the CVS repository. The IDE retrieves this project and merges its contents with the contents of your local project. Clear to preserve the contents of the local project.</p> <p>Selecting this option is equivalent to using the command-line switch <code>-j NumberOrTag</code>.</p>
	and this Revision	<p>Select to enter in the corresponding field a revision tag or branch name that, together with the Revision entry, specifies a version range over which to merge project contents. Clear to preserve the contents of the local project.</p> <p>mwCVS interprets the combination of Revision and this Revision differently from Revision alone. CVS merges into the local project any changes between the version specified by Revision and the version specified by and this Revision.</p> <p>Selecting this option is equivalent to using the command-line switch <code>-j NumberOrTag -j NumberOrTag</code>.</p>
Read-Only		<p>Select to make the file read-only on the disk after issuing the Update command. Clear to make the file read/write.</p> <p>This option supersedes the setting in the mwCVS panel of the VCS Settings window.</p> <p>Selecting this option is equivalent to using the command-line switch <code>-r</code>. Clearing this option is equivalent to using the command-line switch <code>-w</code>.</p>

Table 3.10 Update dialog box (for projects)—options (*continued*)

Option	Explanation
Do not change any files	<p>Select to see the result of the update operation without modifying the CVS repository. Clear to modify the CVS repository to reflect the update operation.</p> <p>Selecting this option is equivalent to using the command-line switch <code>-n</code>.</p>
Command Output	<p>Select the level of information that you want mwCVS to output during CVS operations:</p> <ul style="list-style-type: none"> • Default—output informational messages • Quiet—output non-informational messages (<code>-q</code>) • Very Quiet—output only messages for serious errors (<code>-Q</code>) <p>This option supersedes the setting in the mwCVS panel of the VCS Settings window.</p>

Commit

Bring forward a project window, then select **VCS > Project > Commit** to open the Commit dialog box. [Figure 3.13 on page 40](#) shows the Commit dialog box for a project. [Table 3.11 on page 40](#) explains the dialog-box options.

After you click **Commit**, a progress indicator shows commit progress.

NOTE You can change the default settings for the [Read-Only](#) and [Command Output](#) options. Use the **mwCVS** panel in the **VCS Settings** window to configure the default settings. See [“mwCVS Panel Options” on page 12](#) for more information.

Figure 3.13 Commit dialog box (for projects)

Table 3.11 Commit dialog box (for projects)—options

Option	Explanation
Log message	<p>Enter a message to log with the project that you will commit.</p> <p>Entering a message in this field is equivalent to using the GNU CVS command-line switch <code>-m Text</code>, where <i>Text</i> is the log message to use for this operation.</p>
Commit to revision	<p>Select to enter in the corresponding field a numeric or symbolic revision tag to which you want to commit the project. Clear to commit the project to the latest revision.</p> <p>Selecting this option is equivalent to using the command-line switch <code>-r NumberOrTag</code>, where <i>NumberOrTag</i> is the revision number or tag to use for this operation.</p>
Read-Only	<p>Select to make the file read-only on the disk after issuing the Commit command. Clear to make the file read/write.</p> <p>This option supersedes the setting in the mwCVS panel of the VCS Settings window.</p> <p>Selecting this option is equivalent to using the command-line switch <code>-r</code>. Clearing this option is equivalent to using the command-line switch <code>-w</code>.</p>

Table 3.11 Commit dialog box (for projects)—options (*continued*)

Option	Explanation
Do not change any files	<p>Select to see the result of the commit operation without modifying the CVS repository. Clear to modify the CVS repository to reflect the commit operation.</p> <p>Selecting this option is equivalent to using the command-line switch <code>-n</code>.</p>
Command Output	<p>Select the level of information that you want mwCVS to output during CVS operations:</p> <ul style="list-style-type: none"> • Default—output informational messages • Quiet—output non-informational messages (<code>-q</code>) • Very Quiet—output only messages for serious errors (<code>-Q</code>) <p>This option supersedes the setting in the mwCVS panel of the VCS Settings window.</p>

Status

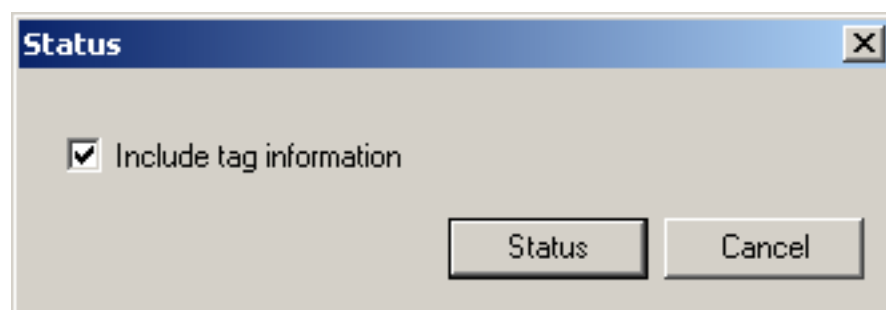
Bring forward a project window, then select **VCS > Project > Status** to open the Status dialog box.

Select **Include tag information** to include information about tags applied to the project revisions in the report. Clear the option to exclude the tag information from the report. Selecting the option is equivalent to using the command-line switch `-v`.

Notice that the default mwCVS status behavior is to include tag information, which is equivalent to using the GNU CVS command line `status -v`.

After you click **Status**, a progress indicator appears. After the IDE completes the status operation, an editor window appears. This editor window shows the status information for the current project.

Figure 3.14 Status dialog box (for projects)



Tag

Bring forward a project window, then select **VCS > Project > Tag** to open the Tag dialog box. [Figure 3.15](#) shows the Tag dialog box for a file. [Table 3.12 on page 43](#) explains the dialog-box options.

After you click **Tag**, a progress indicator appears.

NOTE You can change the default settings for the [Read-Only](#) and [Command Output](#) options. Use the **mwCVS** panel in the **VCS Settings** window to configure the default settings. See [“mwCVS Panel Options” on page 12](#) for more information.

Figure 3.15 Tag dialog box (for projects)

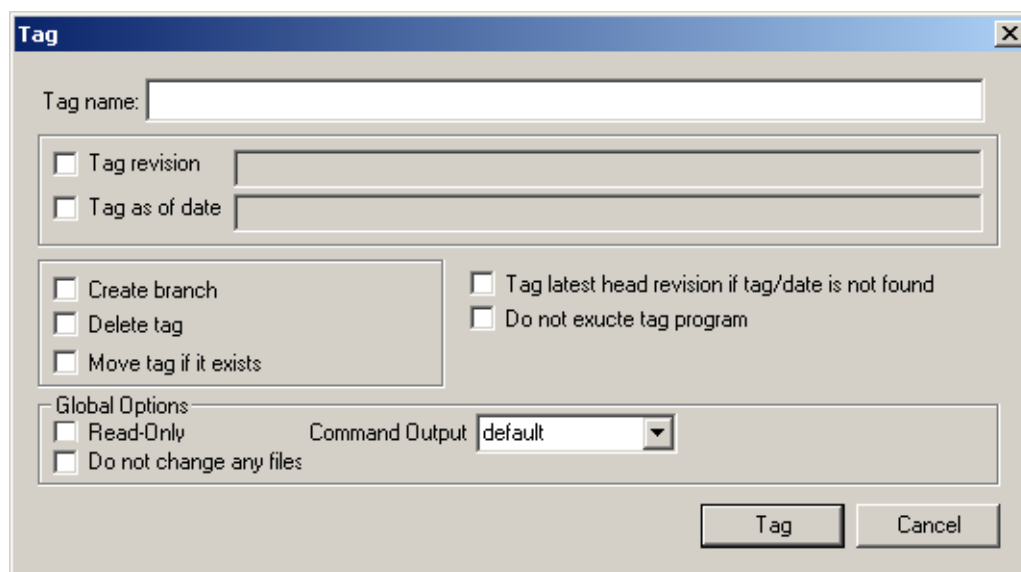


Table 3.12 Tag dialog box (for projects)—options

Option	Explanation
Tag name	Enter in this field the name of the tag.
Tag revision	<p>Select to enter in the corresponding field the project revision that you want to tag. Clear to apply the tag to the latest revision.</p> <p>Selecting this option is equivalent to using the GNU CVS command-line switch <code>-r <i>NumberOrTag</i></code>, where <i>NumberOrTag</i> is the revision number or tag to use for this operation.</p>
Tag as of date	<p>Select to enter in the corresponding field the date of the project revision that you want to tag. Clear to apply the tag to the latest revision date.</p> <p>Selecting this option is equivalent to using the command-line switch <code>-D <i>Date</i></code>, where <i>Date</i> is the date to use for this operation.</p>
Create branch	<p>Select to use the tag to denote a branch of the project revision in the CVS repository. Clear to use the tag to denote a label for a particular project revision.</p> <p>Selecting this option is equivalent to using the command-line switch <code>-b</code>.</p> <p>If you select this option, you cannot select the Delete tag option.</p>
Delete tag	<p>Select to permanently delete the specified tag from the CVS repository. Clear to add the tag to the specified project revision.</p> <p>Selecting this option is equivalent to using the command-line switch <code>-d</code>.</p> <p>If you select this option, you cannot select the Create branch or Move tag if it exists options.</p>

Table 3.12 Tag dialog box (for projects)—options (*continued*)

Option	Explanation
Move tag if it exists	<p>Select to have the IDE relocate existing tags with the same name as the tag you specify. Clear to have the IDE display an error message if it encounters an existing tag with the same name as the tag you specify.</p> <p>Selecting this option is equivalent to using the command-line switch <code>-F</code>.</p> <p>If you select this option, you cannot select the Delete tag option.</p>
Tag latest head revision if tag/date is not found	<p>Select to have the IDE apply the tag to the latest project revision on the CVS repository if it does not encounter a revision with the specified tag or date. Clear to have CVS report an error if it fails to find a matching tag.</p> <p>Selecting this option is equivalent to using the command-line switch <code>-f</code>.</p>
Do not execute tag program	<p>Select to see the result of the tag operation without modifying the CVS repository. Clear to modify the CVS repository to reflect the tag operation.</p> <p>Selecting this option is equivalent to using the command-line switch <code>-n</code>.</p>
Read-Only	<p>Select to make the file read-only on the disk after issuing the Tag command. Clear to make the file read/write.</p> <p>This option supersedes the setting in the mwCVS panel of the VCS Settings window.</p> <p>Selecting this option is equivalent to using the command-line switch <code>-r</code>.</p>

Table 3.12 Tag dialog box (for projects)—options (*continued*)

Option	Explanation
Do not change any files	<p>Select to see the result of the tag operation without modifying the CVS repository. Clear to modify the CVS repository to reflect the tag operation.</p> <p>Selecting this option is equivalent to using the command-line switch <code>-n</code>.</p>
Command Output	<p>Select the level of information that you want mwCVS to output during CVS operations:</p> <ul style="list-style-type: none"> • Default—output informational messages • Quiet—output non-informational messages (<code>-q</code>) • Very Quiet—output only messages for serious errors (<code>-Q</code>) <p>This option supersedes the setting in the mwCVS panel of the VCS Settings window.</p>

Add

Bring forward the project window of a project that does not currently reside in the CVS repository, then select **VCS > Project > Add** to open the Add dialog box. [Figure 3.16 on page 46](#) shows the Add dialog box for a project. [Table 3.13 on page 46](#) explains the dialog-box options.

After you click **Add**, a progress indicator appears.

NOTE The current version of GNU CVS, when using the `pserver` connection method, ignores the **Log Message** of the **Add** command. This is a known bug in GNU CVS. For more information, refer to the file `BUGS` in the CVS source distribution.

Figure 3.16 Add dialog box (for projects)

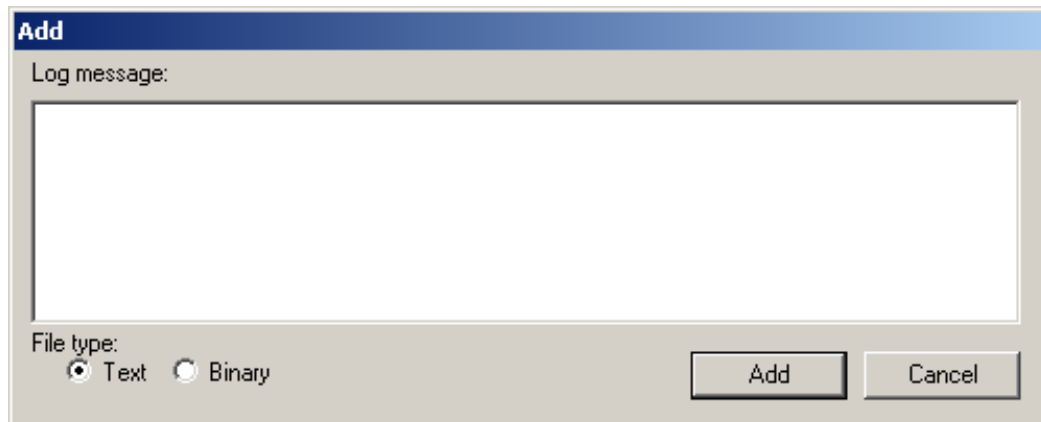


Table 3.13 Add dialog box (for projects)—options

Option	Explanation
Log message	Enter in this field a message to log with the project that you will add. This option is equivalent to using the GNU CVS command-line switch <code>-m <i>Text</i></code> , where <i>Text</i> is the log message to use for this operation.
Text	Select this file type if you are adding a text-formatted project to the CVS repository.
Binary	Select this file type if you are adding a binary (non-text)-formatted project to the CVS repository. Selecting this option is equivalent to using the command-line switch <code>-kb</code> .

Remove

Bring forward a project window, then select **VCS > Project > Remove** to open the Remove dialog box.

mwCVS implements the `remove` command differently from GNU CVS. The default mwCVS behavior for a remove command is to delete the local project from disk and immediately commit that removal to the CVS repository. This behavior allows the IDE to commit the project removal even though the project no longer exists on your computer.

[Figure 3.17 on page 47](#) shows the Remove dialog box for a project. [Table 3.14 on page 47](#) explains the dialog-box options.

After you click **Remove**, a progress indicator appears.

Figure 3.17 Remove dialog box (for projects)

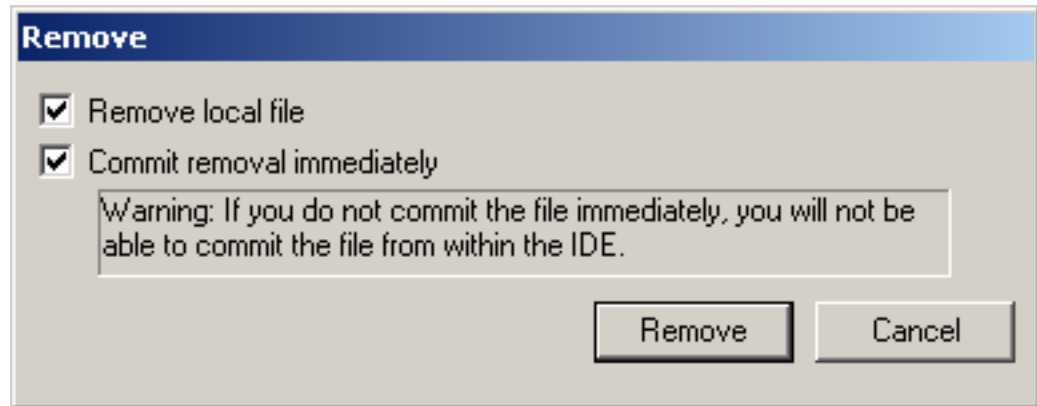


Table 3.14 Remove dialog box (for projects)—options

Option	Explanation
Remove local file	<p>Select to delete the local project as part of the mwCVS remove operation. Clear to preserve the local project.</p> <p>Selecting this option is equivalent to using the GNU CVS command-line switch <code>-f</code>.</p>
Commit removal immediately	<p>Select to have the IDE display the Commit dialog box immediately after the tag command. Use this dialog box to specify commit parameters or cancel the commit operation. Clear to use a separate CVS application to commit the project removal.</p> <p>If you select the Remove local file option, you should select this option if you want to continue working within the IDE. Selecting this option lets you avoid switching to a separate CVS application to commit the project removal.</p>

Log

Bring forward a project window, then select **VCS > Project > Log** to open the Log dialog box. [Figure 3.18](#) shows the Log dialog box for a file. [Table 3.15](#) explains the dialog-box options.

After you click **Log**, a progress indicator appears. After the IDE completes the log operation, an editor window appears. This window shows the log information for the current project.

Figure 3.18 Log dialog box (for projects)

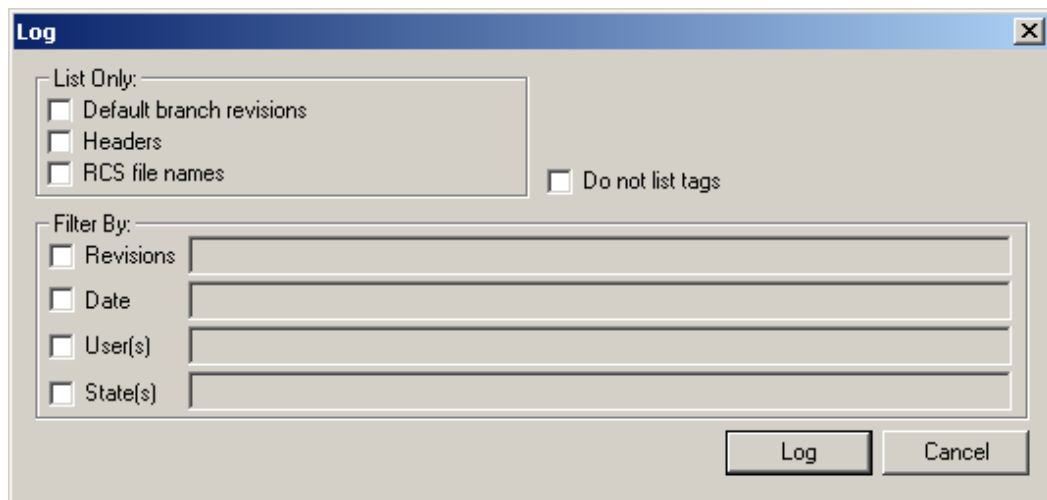


Table 3.15 Log dialog box (for projects)—options

Option	Explanation
Default branch revisions	Select to list default branch revisions in the log. Clear to list all branch revisions. Selecting this option is equivalent to using the GNU CVS command-line switch <code>-b</code> .
Headers	Select to list header information in the log. Clear to exclude header information from the log. Selecting this option is equivalent to using the command-line switch <code>-h</code> .

Table 3.15 Log dialog box (for projects)—options (*continued*)

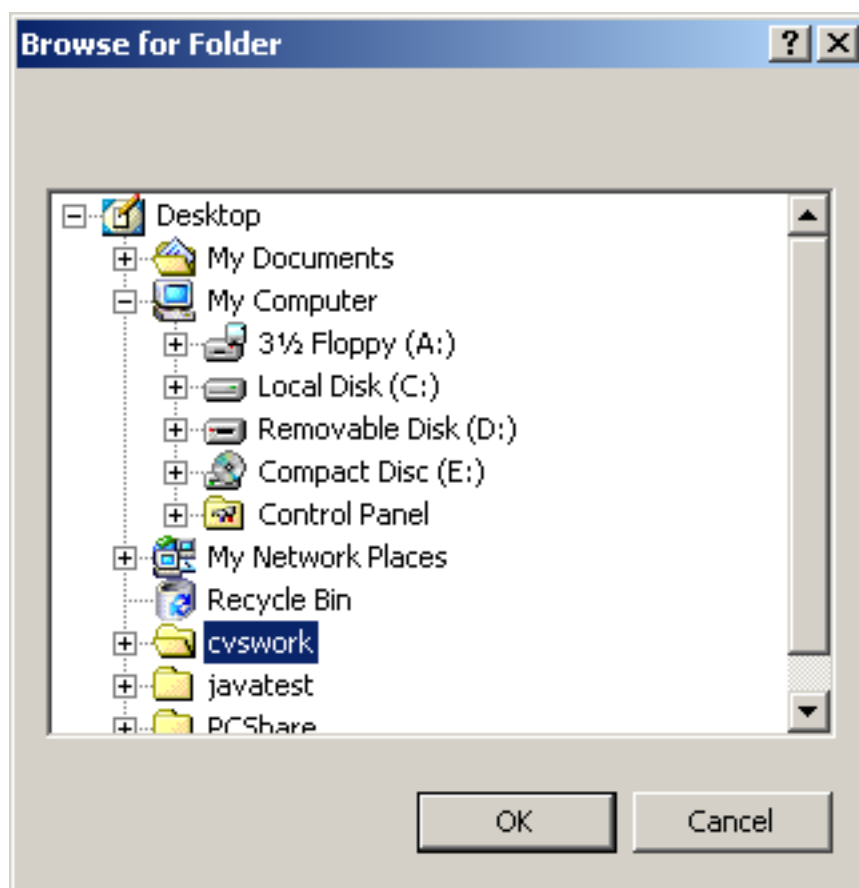
Option	Explanation
RCS file names	<p>Select to list project names that comply with Revision Control System (RCS) naming conventions. Clear to exclude RCS names from the log.</p> <p>Selecting this option is equivalent to using the command-line switch <code>-R</code>.</p>
Do not list tags	<p>Select to exclude tag information from the log. Clear to include the tag information.</p> <p>Selecting this option is equivalent to using the command-line switch <code>-N</code>.</p>
Revisions	<p>Select to enter in the corresponding field the project revisions for which you want to retrieve log information. Clear to retrieve log information for all project revisions.</p> <p>Selecting this option is equivalent to using the command-line switch <code>-r NumberOrTag: NumberOrTag</code>, where <i>NumberOrTag</i> is the revision number or tag to use for this operation.</p>
Date	<p>Select to enter in the corresponding field the dates of project revisions for which you want to retrieve log information. Clear to retrieve log information for all project-revision dates.</p> <p>Selecting this option is equivalent to using the command-line switch <code>-d Date</code>, where <i>Date</i> is the date to use for this operation.</p>
User(s)	<p>Select to enter in the corresponding field the users for whose project revisions you want to retrieve log information. Clear to retrieve log information for all users who revised the project.</p> <p>Selecting this option is equivalent to using the command-line switch <code>-w LoginName1, LoginName2, ...</code>, where <i>LoginName</i> is the login name to use for this operation.</p>
State(s)	<p>Select to enter in the corresponding field the state information of projects for which you want to retrieve log information. Clear to retrieve log information for all project states.</p> <p>Selecting this option is equivalent to using the command-line switch <code>-s</code>.</p>

Recursive Commands

This section explains the recursive mwCVS commands. These commands let you specify a directory (folder) as the argument to the CVS command. After you choose one of these recursive commands, a directory-selection dialog box like the one shown in [Figure 3.19](#) appears. Use this dialog box to select the directory to which you want to issue the CVS command.

Click **Cancel** in the directory-selection dialog box or in subsequent dialog boxes to close the dialog box without making any modifications. Cancelling the operation in the directory-selection dialog box also cancels the CVS command.

Figure 3.19 Directory-selection dialog box



Update

Select **VCS > Recursive > Update** and select a directory in the directory-selection dialog box to open the Update dialog box. [Figure 3.20](#) shows the Update dialog box for a directory. [Table 3.16](#) explains the dialog-box options.

After you click **Update**, a progress indicator appears.

NOTE You can change the default settings for the [Read-Only](#) and [Command Output](#) options. Use the **mwCVS** panel in the **VCS Settings** window to configure the default settings. See [“mwCVS Panel Options” on page 12](#) for more information.

Figure 3.20 Update dialog box (recursive)

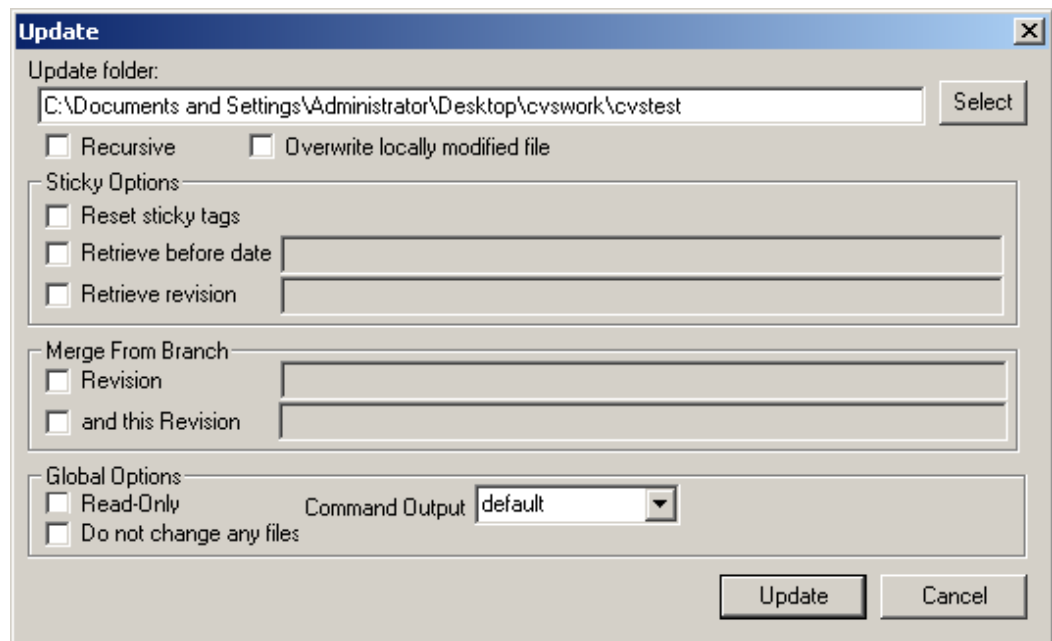


Table 3.16 Update dialog box (recursive)—options

Option	Explanation
Update folder	This field holds the path to the directory that you selected in the directory-selection dialog box. Click Select to choose a different directory. Alternatively, enter a full path directly into the field.
Recursive	Select to update both the directory itself and any directories nested within it. Clear to update only the directory itself.

Table 3.16 Update dialog box (recursive)—options (*continued*)

Option	Explanation
Overwrite locally modified file	<p>Select to replace a locally modified file with the version stored on the CVS repository. Clear to preserve the locally modified file.</p> <p>Selecting this option is equivalent to using the GNU CVS command-line switch <code>-C</code>.</p>
Reset sticky tags	<p>Select to remove any sticky tags associated with the selected file after the IDE completes the update operation. Clear to preserve existing sticky tags.</p> <p>Selecting this option is equivalent to using the command-line switch <code>-A</code>.</p>
Retrieve before date	<p>Select to enter in the corresponding field the date before which you want to retrieve a version of the file in the CVS repository. Clear to disregard date information.</p> <p>Enter the date according to GNU CVS conventions. Sample dates are:</p> <ul style="list-style-type: none">• 1 month ago• last Monday• 3/31/02 10:00:04 PST <p>Selecting this option is equivalent to using the command-line switch <code>-D Date</code>, where <i>Date</i> is the date to use for this operation.</p>
Retrieve revision	<p>Select to enter in the corresponding field the revision tag of the file that you want to retrieve from the CVS repository. Clear to disregard revision-tag information.</p> <p>Selecting this option is equivalent to using the command-line switch <code>-r NumberOrTag</code>, where <i>NumberOrTag</i> is the revision number or tag to use for this operation.</p>

Table 3.16 Update dialog box (recursive)—options (*continued*)

Option		Explanation
Merge From Branch	Revision	<p>Select to enter in the corresponding field the revision tag or branch name of the file that you want to retrieve from the CVS repository. The IDE retrieves this file and merges its contents with the contents of your local file. Clear to preserve the contents of the local file.</p> <p>Selecting this option is equivalent to using the command-line switch <code>-j NumberOrTag</code>.</p>
	and this Revision	<p>Select to enter in the corresponding field a revision tag or branch name that, together with the Revision entry, specifies a version range over which to merge file contents. Clear to preserve the contents of the local file.</p> <p>The IDE interprets the combination of Revision and this Revision differently from Revision alone. CVS merges into the local file any changes between the version specified by Revision and the version specified by and this Revision.</p> <p>Selecting this option is equivalent to using the command-line switch <code>-j NumberOrTag -j NumberOrTag</code>.</p>
Read-Only		<p>Select to make the file read-only on the disk after issuing the Update command. Clear to make the file read/write.</p> <p>This option supersedes the setting in the mwCVS panel of the VCS Settings window.</p> <p>Selecting this option is equivalent to using the command-line switch <code>-r</code>. Clearing this option is equivalent to using the command-line switch <code>-w</code>.</p>

Table 3.16 Update dialog box (recursive)—options (*continued*)

Option	Explanation
Do not change any files	<p>Select to see the result of the update operation without modifying the CVS repository. Clear to modify the CVS repository to reflect the update operation.</p> <p>This option supersedes the setting in the mwCVS panel of the VCS Settings window.</p> <p>Selecting this option is equivalent to using the command-line switch <code>-n</code>.</p>
Command Output	<p>Select the level of information that you want mwCVS to output during CVS operations:</p> <ul style="list-style-type: none">• Default—output informational messages• Quiet—output non-informational messages (<code>-q</code>)• Very Quiet—output only messages for serious errors (<code>-Q</code>) <p>This option supersedes the setting in the mwCVS panel of the VCS Settings window.</p>

Commit

Select **VCS > Recursive > Commit** and select a directory in the directory-selection dialog box to open the Commit dialog box. [Figure 3.21 on page 55](#) shows the Commit dialog box for a directory. [Table 3.17 on page 55](#) explains the dialog-box options.

After you click **Commit**, a progress indicator appears.

NOTE You can change the default settings for the [Read-Only](#) and [Command Output](#) options. Use the **mwCVS** panel in the **VCS Settings** window to configure the default settings. See [“mwCVS Panel Options” on page 12](#) for more information.

Figure 3.21 Commit dialog box (recursive)

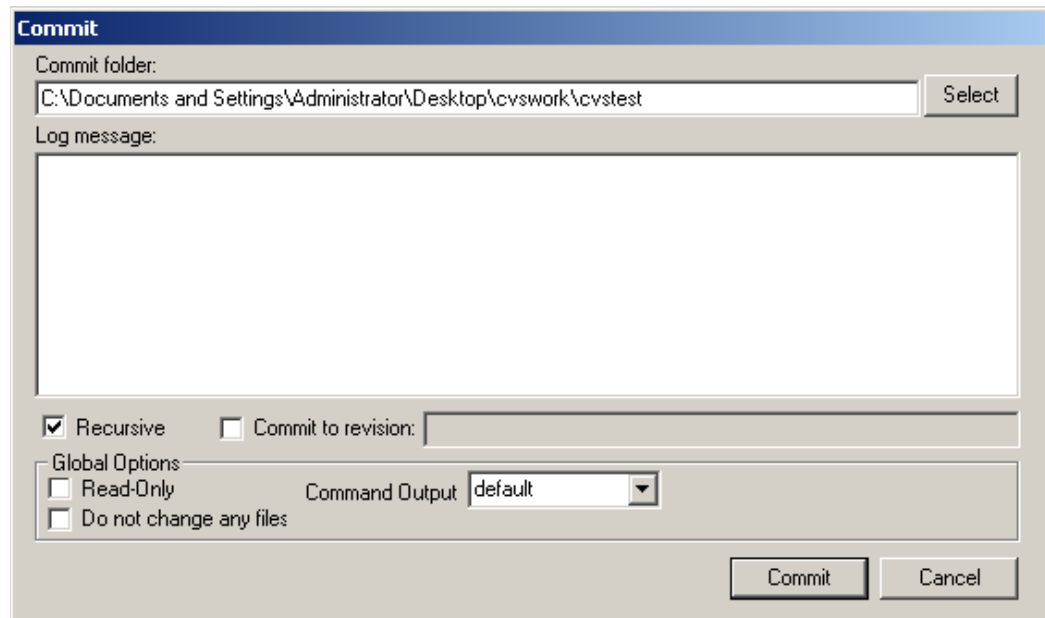


Table 3.17 Commit dialog box (recursive)—options

Option	Explanation
Commit folder	This field holds the path to the directory that you selected in the directory-selection dialog box. Click Select to choose a different directory. Alternatively, enter a full path directly into the field.
Log message	Enter in this field a message to log with the file that you will commit. Entering a message in this field is equivalent to using the GNU CVS command-line switch <code>-m Text</code> , where <i>Text</i> is the log message to use for this operation.
Recursive	Select to commit both the directory itself and any directories nested within it. Clear to commit only the directory itself.
Commit to revision	Select to enter in the corresponding field a numeric or symbolic revision tag to which you want to commit the file. Clear to commit the file to the latest revision. Selecting this option is equivalent to using the command-line switch <code>-r NumberOrTag</code> , where <i>NumberOrTag</i> is the revision number or tag to use for this operation.

Table 3.17 Commit dialog box (recursive)—options (*continued*)

Option	Explanation
Read-Only	<p>Select to make the file read-only on the disk after issuing the Commit command. Clear to make the file read/write.</p> <p>This option supersedes the setting in the mwCVS panel of the VCS Settings window.</p> <p>Selecting this option is equivalent to using the command-line switch <code>-r</code>. Clearing this option is equivalent to using the command-line switch <code>-w</code>.</p>
Do not change any files	<p>Select to see the result of the commit operation without modifying the CVS repository. Clear to modify the CVS repository to reflect the commit operation.</p> <p>This option supersedes the setting in the mwCVS panel of the VCS Settings window.</p> <p>Selecting this option is equivalent to using the command-line switch <code>-n</code>.</p>
Command Output	<p>Select the level of information that you want mwCVS to output during CVS operations:</p> <ul style="list-style-type: none">• Default—output informational messages• Quiet—output non-informational messages (<code>-q</code>)• Very Quiet—output only messages for serious errors (<code>-Q</code>) <p>This option supersedes the setting in the mwCVS panel of the VCS Settings window.</p>

Checkout

Select **VCS > Recursive > Checkout** and select a directory in the directory-selection dialog box to open the Checkout dialog box. [Figure 3.22 on page 57](#) shows the Checkout dialog box for a directory. [Table 3.18 on page 57](#) explains the dialog-box options.

After you click **Checkout**, a progress indicator appears. After the progress indicator closes, the **mwCVS Results** window appears.

NOTE You can change the default settings for the [Read-Only](#) and [Command 5Output](#) options. Use the **mwCVS** panel in the **VCS Settings** window to

configure the default settings. See [“mwCVS Panel Options” on page 12](#) for more information.

Figure 3.22 Checkout dialog box

Table 3.18 Checkout dialog box—options

Option	Explanation
Checkout to	This field holds the path to the directory that you selected in the directory-selection dialog box. Click Select to choose a different directory. Alternatively, enter a full path directly into the field.
Module	Enter in this field the path to the module that you want to check out from the CVS repository.

Table 3.18 Checkout dialog box—options (*continued*)

Option	Explanation
CVSRoot	<p>Enter in this field the root of the CVS repository. Use the general construction</p> <p><i>:protocol:Login@RepstryAddr:RepstryRoot</i></p> <p>where</p> <ul style="list-style-type: none">• <i>protocol</i> is the protocol you use to access the repository (such as <i>local</i> or <i>pserver</i>).• <i>Login@RepstryAddr</i> is your login name (<i>Login</i>), followed by <i>@</i>, followed by the Internet Protocol (IP) or Domain Name System (DNS) address of the remote repository (<i>RepstryAddr</i>)• <i>RepstryRoot</i> is the path to the root level of the remote repository <p>The default entry that appears in this field corresponds to the last used CVSRoot entry.</p>
Use date	<p>Select to check out files with a sticky date. Clear to disregard date when checking out files.</p> <p>Selecting this option is equivalent to using the GNU CVS command-line switch <i>-D Date</i>, where <i>Date</i> is the date to use for this operation.</p>
Use revision	<p>Select to check out files with a sticky revision tag. Clear to disregard revision tag when checking out files.</p> <p>Selecting this option is equivalent to using the command-line switch <i>-r NumberOrTag</i>, where <i>NumberOrTag</i> is the revision number or tag to use for this operation.</p>
Recursive	<p>Select to check out both the directory itself and any directories nested within it. Clear to check out only the directory itself.</p> <p>Clearing this option is equivalent to using the command-line switch <i>-l</i>.</p>
Prune empty directories	<p>Select to avoid checking out empty directories from the CVS repository. Clear to check out all directories, regardless of empty state.</p> <p>Selecting this option is equivalent to using the command-line switch <i>-P</i>.</p>

Table 3.18 Checkout dialog box—options (*continued*)

Option	Explanation
Reset sticky tags	<p>Select to remove any sticky tags associated with the selected file after the IDE completes the checkout operation. Clear to preserve existing sticky tags.</p> <p>Selecting this option is equivalent to using the command-line switch <code>-A</code>.</p>
Don't run module program	<p>Select to see the result of the commit operation without modifying the CVS repository. Clear to modify the CVS repository to reflect the commit operation.</p> <p>Selecting this option is equivalent to using the command-line switch <code>-n</code>.</p>
Force head revision if date/tag not found	<p>Select to have the IDE force a head revision on files checked out from the CVS repository if it does not encounter a revision with the specified tag or date. Clear to have the IDE cancel the checkout operation if it does not encounter a revision with the specified tag or date.</p> <p>Selecting this option is equivalent to using the command-line switch <code>-f</code>.</p>
Read-Only	<p>Select to make the file read-only on the disk after issuing the Checkout command. Clear to make the file read/write.</p> <p>This option supersedes the setting in the mwCVS panel of the VCS Settings window.</p> <p>Selecting this option is equivalent to using the command-line switch <code>-r</code>. Clearing this option is equivalent to using the command-line switch <code>-w</code>.</p>

Table 3.18 Checkout dialog box—options (*continued*)

Option	Explanation
Do not change any files	Select to see the result of the commit operation without modifying the CVS repository. Clear to modify the CVS repository to reflect the commit operation. Selecting this option is equivalent to using the command-line switch <code>-n</code> .
Command 5Output	Select the level of information that you want mwCVS to output during CVS operations: <ul style="list-style-type: none">• Default—output informational messages• Quiet—output non-informational messages (<code>-q</code>)• Very Quiet—output only messages for serious errors (<code>-Q</code>) This option supersedes the setting in the mwCVS panel of the VCS Settings window.

Status

Select **VCS > Recursive > Status** and select a directory in the directory-selection dialog box to open the Status dialog box. [Figure 3.23](#) shows the Status dialog box for a directory. [Table 3.19 on page 61](#) explains the dialog-box options.

After you click **Status**, a progress indicator appears.

Figure 3.23 Status dialog box (recursive)

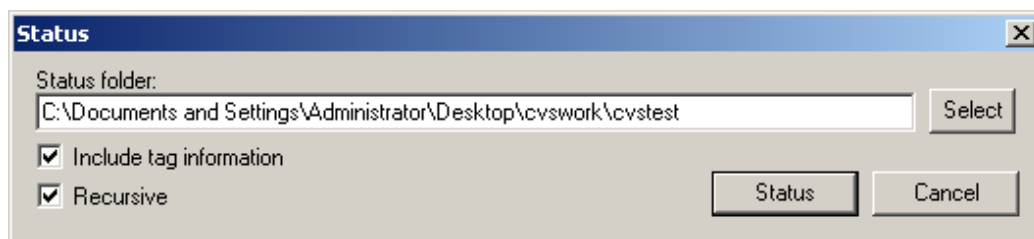


Table 3.19 Status dialog box (recursive)—options

Option	Explanation
Status folder	This field holds the path to the directory that you selected in the directory-selection dialog box. Click Select to choose a different directory. Alternatively, enter a full path directly into the field.
Include tag information	<p>Select to include in the status report information about tags applied to the file revisions. Clear to exclude the tag information from the status report.</p> <p>Selecting this option is equivalent to using the GNU CVS command-line switch <code>-v</code>.</p>
Recursive	Selecting this option is equivalent to using the command-line switch <code>-r</code> . Clearing this option is equivalent to using the command-line switch <code>-l</code> .

Tag

Select **VCS > Recursive > Tag** and select a directory in the directory-selection dialog box to open the Tag dialog box. [Figure 3.24 on page 62](#) shows the Tag dialog box for a directory. [Table 3.20 on page 62](#) explains the dialog-box options.

After you click **Tag**, a progress indicator appears.

NOTE You can change the default settings for the [Read-Only](#) and [Command Output](#) options. Use the **mwCVS** panel in the **VCS Settings** window to configure the default settings. See [“mwCVS Panel Options” on page 12](#) for more information.

Figure 3.24 Tag dialog box (recursive)

Table 3.20 Tag dialog box (recursive)—options

Option	Explanation
Tag folder	This field holds the path to the directory that you selected in the directory-selection dialog box. Click Select to choose a different directory. Alternatively, enter a full path directly into the field.
Tag name	Enter in this field the name of the tag.
Tag revision	<p>Select to enter in the corresponding field the file revision that you want to tag. Clear to apply the tag to the latest revision.</p> <p>Selecting this option is equivalent to using the GNU CVS command-line switch <code>-r <i>NumberOrTag</i></code>, where <i>NumberOrTag</i> is the revision number or tag to use for this operation.</p>
Tag as of date	<p>Select to enter in the corresponding field the date of the file revision that you want to tag. Clear to apply the tag to the latest revision date.</p> <p>Selecting this option is equivalent to using the command-line switch <code>-D <i>Date</i></code>, where <i>Date</i> is the date to use for this operation.</p>

Table 3.20 Tag dialog box (recursive)—options (*continued*)

Option	Explanation
Create branch	<p>Select to use the tag to denote a branch of the file revision in the CVS repository. Clear to use the tag to denote a label for a particular file revision.</p> <p>Selecting this option is equivalent to using the command-line switch <code>-b</code>.</p> <p>If you select this option, you cannot select the Delete tag option.</p>
Delete tag	<p>Select to permanently delete the specified tag from the CVS repository. Clear to add the tag to the specified file revision.</p> <p>Selecting this option is equivalent to using the command-line switch <code>-d</code>.</p> <p>If you select this option, you cannot select the Create branch or Move tag if it exists options.</p>
Move tag if it exists	<p>Select to have the IDE relocate existing tags with the same name as the tag you specify. Clear to have the IDE display an error message if it encounters an existing tag with the same name as the tag you specify.</p> <p>Selecting this option is equivalent to using the command-line switch <code>-F</code>.</p> <p>If you select this option, you cannot select the Delete tag option.</p>
Recursive	<p>Select to apply the tag to both the directory itself and any directories nested within it. Clear to apply the tag to only the directory itself.</p>
Tag latest head revision if tag/date is not found	<p>Select to have the IDE apply the tag to the latest file revision on the CVS repository if it does not encounter a revision with the specified tag or date. Clear to have CVS report an error if it fails to find a matching tag.</p> <p>Selecting this option is equivalent to using the command-line switch <code>-f</code>.</p>

Table 3.20 Tag dialog box (recursive)—options (*continued*)

Option	Explanation
Read-Only	Select to make the file read-only on the disk after issuing the Tag command. Clear to make the file read/write. This option supersedes the setting in the mwCVS panel of the VCS Settings window. Selecting this option is equivalent to using the command-line switch <code>-r</code> .
Do not change any files	Select to see the result of the tag operation without modifying the CVS repository. Clear to modify the CVS repository to reflect the tag operation. Selecting this option is equivalent to using the command-line switch <code>-n</code> .
Command Output	Select the level of information that you want mwCVS to output during CVS operations: <ul style="list-style-type: none">• Default—output informational messages• Quiet—output non-informational messages (<code>-q</code>)• Very Quiet—output only messages for serious errors (<code>-Q</code>) This option supersedes the setting in the mwCVS panel of the VCS Settings window.

Log

Select **VCS > Recursive > Log** and select a directory in the directory-selection dialog box to open the Log dialog box. [Figure 3.25 on page 65](#) shows the Log dialog box for a directory. [Table 3.21 on page 65](#) explains the dialog-box options.

After you click **Log**, a progress indicator appears. After the IDE completes the log operation, an editor window appears. This window shows the log information for the selected file.

Figure 3.25 Log dialog box (recursive)

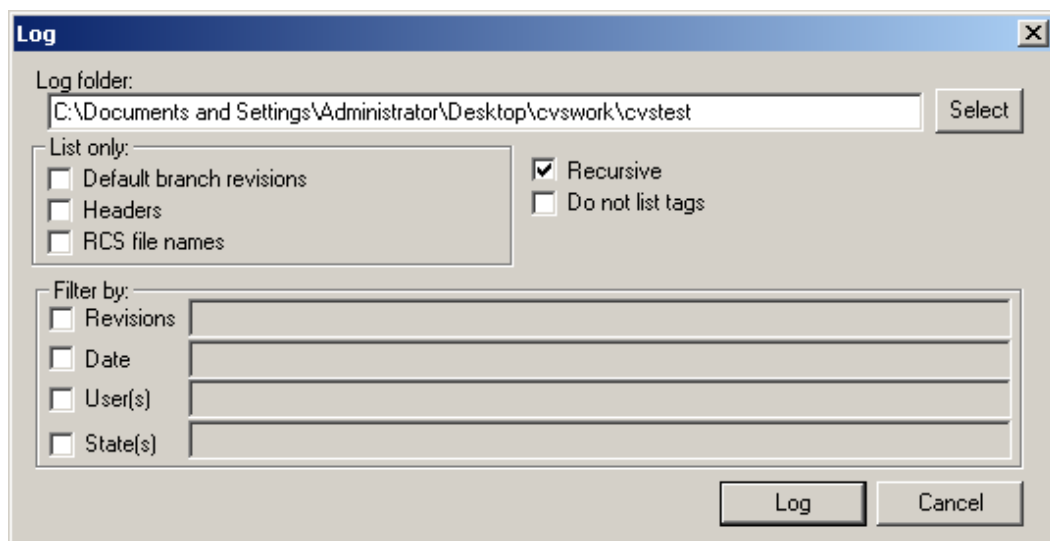


Table 3.21 Log dialog box (recursive)—options

Option	Explanation
Log folder	This field holds the path to the directory that you selected in the directory-selection dialog box. Click Select to choose a different directory. Alternatively, enter a full path directly into the field.
Default branch revisions	Select to list default branch revisions in the log. Clear to list all branch revisions. Selecting this option is equivalent to using the GNU CVS command-line switch <code>-b</code> .
Headers	Select to list header information in the log. Clear to exclude header information from the log. Selecting this option is equivalent to using the command-line switch <code>-h</code> .
RCS file names	Select to list file names that comply with Revision Control System (RCS) naming conventions. Clear to exclude RCS file names from the log. Selecting this option is equivalent to using the command-line switch <code>-R</code> .

Table 3.21 Log dialog box (recursive)—options (*continued*)

Option	Explanation
Recursive	<p>Select to retrieve log information for both the directory itself and any directories nested within it. Clear to retrieve log information for only the directory itself.</p> <p>Clearing this box is equivalent to using the command-line switch <code>-l</code>.</p>
Do not list tags	<p>Select to exclude tag information from the log. Clear to include the tag information.</p> <p>Selecting this option is equivalent to using the command-line switch <code>-N</code>.</p>
Revisions	<p>Select to enter in the corresponding field the file revisions for which you want to retrieve log information. Clear to retrieve log information for all file revisions.</p> <p>Selecting this option is equivalent to using the command-line switch <code>-r NumberOrTag: NumberOrTag</code>, where <i>NumberOrTag</i> is the revision number or tag to use for this operation.</p>
Date	<p>Select to enter in the corresponding field the dates of file revisions for which you want to retrieve log information. Clear to retrieve log information for all file-revision dates.</p> <p>Selecting this option is equivalent to using the command-line switch <code>-d Date</code>, where <i>Date</i> is the date to use for this operation.</p>
User(s)	<p>Select to enter in the corresponding field the users for whose file revisions you want to retrieve log information. Clear to retrieve log information for all users who revised the file.</p> <p>Selecting this option is equivalent to using the command-line switch <code>-w LoginName1, LoginName2, ...</code>, where <i>LoginName</i> is the login name to use for this operation.</p>
State(s)	<p>Select to enter in the corresponding field the state information of files for which you want to retrieve log information. Clear to retrieve log information for all file states.</p> <p>Selecting this option is equivalent to using the command-line switch <code>-s</code>.</p>

mwCVS Tutorial

The tutorial in this chapter is a general example of using mwCVS with the CodeWarrior™ IDE to interact with a CVS repository.

The goal of this tutorial is to help you become familiar with basic use of mwCVS. After completing the tutorial, you will understand how to use mwCVS to

- check out a module from a CVS repository
- modify a source-code file, and
- commit your source-code changes to the CVS repository.

Because there are many ways to set up and use a CVS repository, this tutorial uses a trivial example to help clarify your understanding of mwCVS.

Assume these facts:

- You are familiar with basic CVS use, but you want to better understand how to use mwCVS.
- You created a CVS repository on a UNIX system.
- You previously initialized the repository, configured a `pserver`, and imported source-code files into the repository.
- The root level of the CVS repository is `/VersionControl/Repository`.
- The `Repository` subdirectory has a module called `cvstest`.
- The `cvstest` module has a CodeWarrior project called `cvs_test.mcp` and the associated files for that project.
- All files in the `cvs_test.mcp` project are under version control.

This chapter has these topics:

- [“Part One: Turning on mwCVS” on page 68](#)
- [“Part Two: Accessing the Repository” on page 69](#)
- [“Part Three: Checking out a Module” on page 72](#)
- [“Part Four: Modifying a File” on page 75](#)
- [“Part Five: Committing the Modified File” on page 78](#)

Part One: Turning on mwCVS

This part explains how to turn on mwCVS so that you can use the IDE to issue CVS commands.

- 1** Launch the CodeWarrior IDE.
- 2** Select **Edit > Version Control Settings**.
The **VCS Settings** window appears.
- 3** Select **VCS Setup** in the VCS Settings Panels list.
The VCS Setup panel appears.
- 4** Select the **Use Version Control** option.
- 5** Select **mwCVS** from the **Method** list box.
The mwCVS item appears in the VCS Settings Panels list.
- 6** Click **Apply** or **Save** to confirm your changes.
- 7** Close the VCS Settings window.

Enabling mwCVS causes the VCS menu to appear in the menu bar. You use this menu to issue CVS commands from the CodeWarrior IDE.



This icon represents the VCS menu on the Macintosh-hosted IDE.

Part Two: Accessing the Repository

This part explains how to use mwCVS to log in, such as through a password server (pserver), to access the contents of a CVS repository.

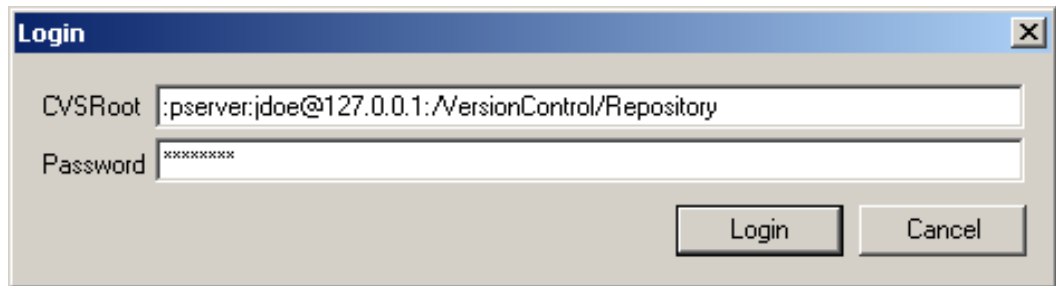
Windows

Follow the instructions in this section if you use mwCVS with the Windows-hosted IDE. See [“Macintosh” on page 70](#) if you use mwCVS with the Macintosh-hosted IDE.

- 1 Select **VCS > Login**.

The Login window appears. Use this window, shown in [Figure 4.1](#), to enter CVS root information and your password to access the repository.

Figure 4.1 Login window



- 2 Enter in the **CVSRoot** field the path to the repository root.

mwCVS requires specific syntax for the contents of this field. The general form is `:protocol:Login@RepositoryAddress:RepositoryRoot`

where *protocol* is the protocol you use to access the repository, *Login* is your login name for the repository, *RepositoryAddress* is the Internet Protocol (IP) or Domain Name System (DNS) address of the repository, and *RepositoryRoot* is the path to the root level of the repository.

For example, in this tutorial you use a password server to access the repository on a UNIX host, so you would enter the information shown in [Figure 4.1](#) into the **CVSRoot** field.

- 3 Enter in the **Password** field the password associated with the login name that you specified in the **CVSRoot** field.

This password is the password that *Login* uses to access the repository.

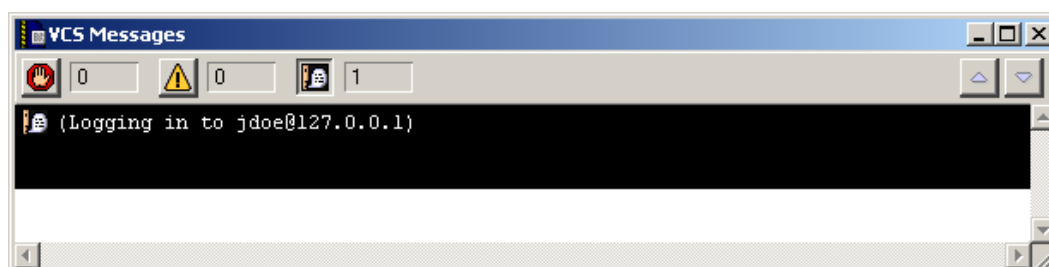
4 Click Login.

The Login window closes and a progress indicator appears. mwCVS logs in to the repository with your account information. The progress indicator closes after the login operation completes.

5 View login information in the VCS Messages window that appears.

This window shows whether mwCVS successfully logged you into the repository. [Figure 4.2](#) shows the successful login to the repository.

Figure 4.2 VCS Messages window - login



Macintosh

Follow the instructions in this section if you use mwCVS with the Macintosh-hosted IDE. See [“Windows” on page 69](#) if you use mwCVS with the Windows-hosted IDE.

1 Launch a command-line utility.

For example, launch the Terminal application in Mac OS X. ([Figure 4.3 on page 71](#) shows Terminal after you use it to successfully log in to a CVS repository.)

2 At the prompt, enter the command line

```
cvs -d :protocol:Login@RepositoryAddr:RepositoryRoot login
```

where *protocol* is the protocol you use to access the repository, *Login* is your login name for the repository, *RepositoryAddr* is the Internet Protocol (IP) or Domain Name System (DNS) address of the repository, and *RepositoryRoot* is the path to the root level of the repository.

For example, in this tutorial you use a password server to access the repository on a UNIX host, so you would enter the command line shown in [Figure 4.3](#).

Figure 4.3 Using the Terminal application to log in to a CVS repository



- 3 Press Return to execute the command line.
CVS prompts you to enter your password.
- 4 Enter the password associated with the *Login* that you specified in the first command line.
This password is the password that *Login* uses to access the repository.
- 5 Press Return to execute the command line.
After you successfully log in to the repository, a new prompt appears in the command-line utility (as shown in [Figure 4.3](#)). Otherwise, error messages appear in the command-line utility.
- 6 After you log in to the repository, return to the CodeWarrior IDE.
You can now use mwCVS to interact with the repository.

Part Three: Checking out a Module

This part explains how to check out a module from the CVS repository. In this tutorial, you check out a sample module named `cvstest` into a working directory named `cvswork`.

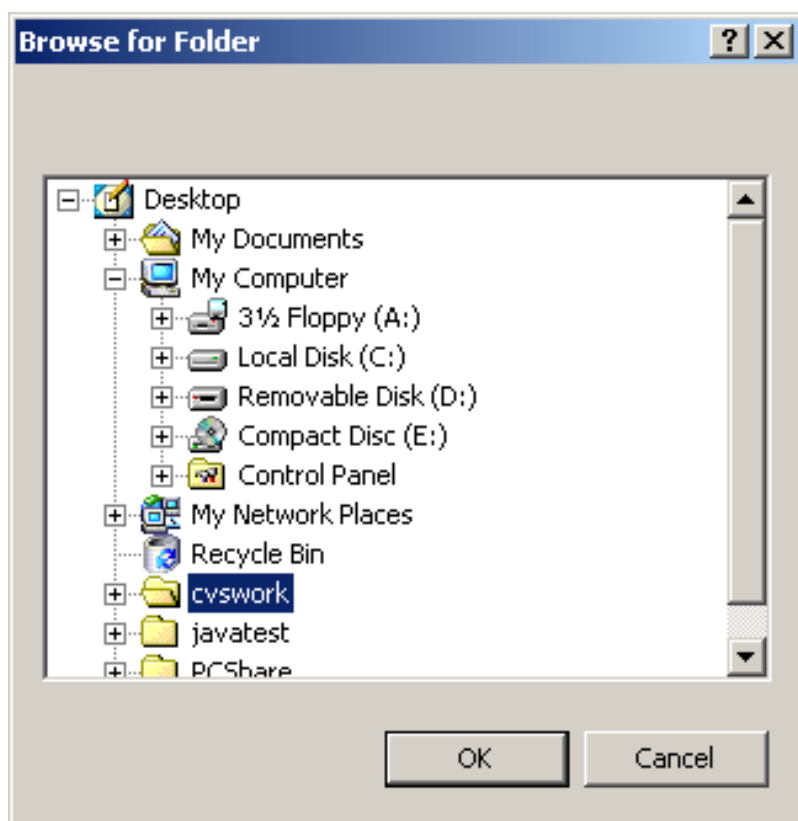
- 1 Select **VCS > Recursive > Checkout**.

A directory-selection dialog box like the one shown in [Figure 4.4](#) appears.

- 2 Use the directory-selection dialog box to select a working directory.

[Figure 4.4](#) shows `cvswork` as the selected directory. This directory becomes the working directory for the checkout command.

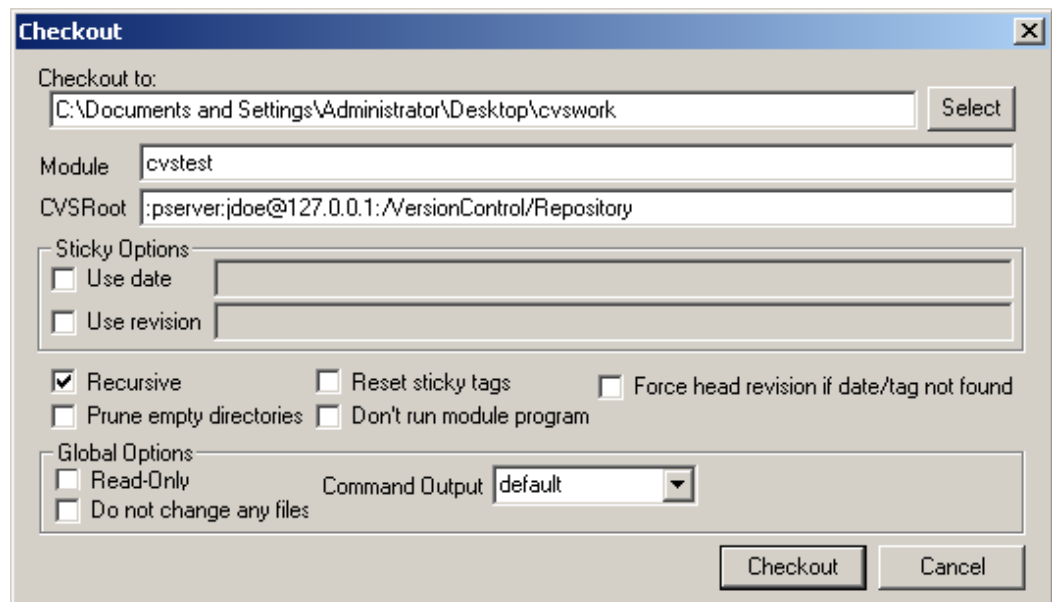
Figure 4.4 Directory-selection dialog box



- 3 Enter repository and module information in the Checkout window that appears.

Use the Checkout window, shown in [Figure 4.5](#), to enter information about the repository and the modules that you want to check out. [Figure 4.5](#) also shows that mwCVS automatically placed the path to the working directory (`cvswork`) in the **Checkout to** field.

Figure 4.5 Checkout window



- 4 Enter in the **Module** field the module that you want to check out.

In this tutorial, the module is `cvstest`.

- 5 Enter in the **CVSRoot** field the CVSROOT setting.

mwCVS requires specific syntax for the contents of this field. The general form is

`:protocol:Login@RepositoryAddress:RepositoryRoot`

where *protocol* is the protocol you use to access the repository, *Login* is your login name for the repository, *RepositoryAddress* is the Internet Protocol (IP) or Domain Name System (DNS) address of the repository, and *RepositoryRoot* is the path to the root level of the repository.

For example, in this tutorial you use a password server to access the repository on a UNIX host, so you would enter the information shown in [Figure 4.5](#) into the **CVSRoot** field.

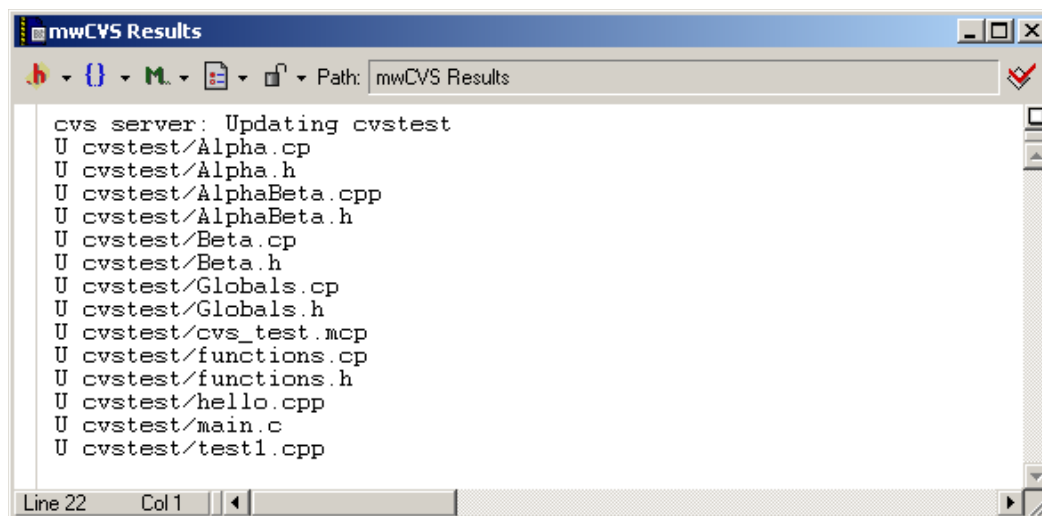
6 Click Checkout.

The Checkout window closes and a progress indicator appears. mwCVS places a copy of the module in your working directory. The progress indicator closes after this operation completes.

7 View checkout information in the **mwCVS Results window that appears.**

This window shows the result of the checkout operation. In this tutorial, [Figure 4.6](#) shows that mwCVS successfully checked out a collection of files for the `cvstest` module.

Figure 4.6 mwCVS Results window - checkout



8 mwCVS checks out the module to the working directory.

In this tutorial, the checkout operation created a new folder, `cvstest`, inside the `cvswork` folder. The `cvstest` folder has the files checked out from the repository.

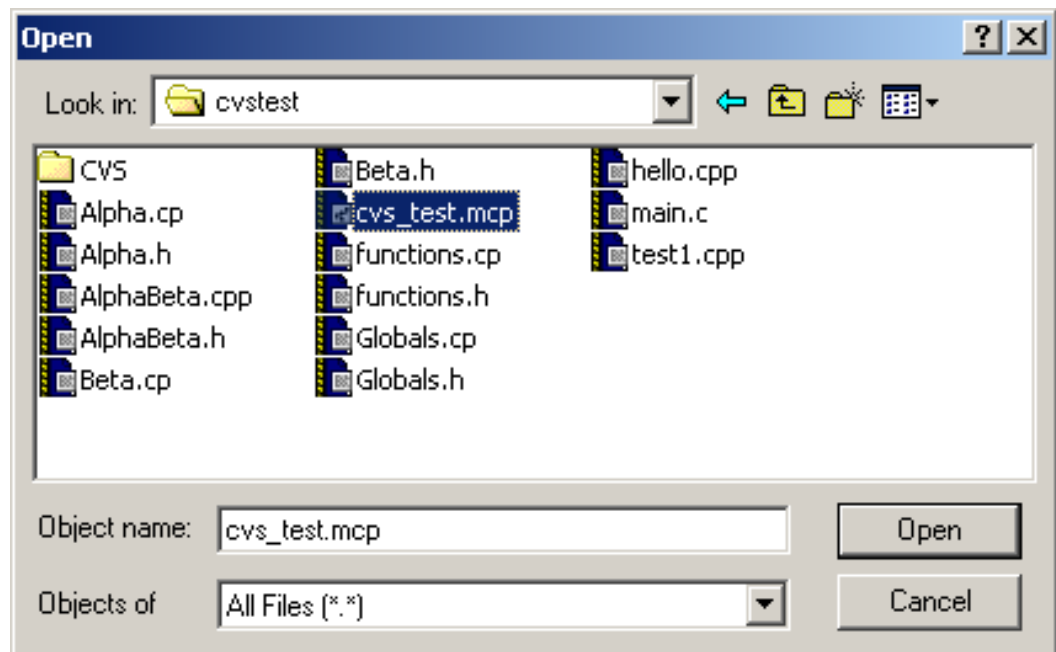
Part Four: Modifying a File

This part explains how to modify a file in a module checked out from the CVS repository. In this tutorial, you modify a file named `Beta.h` that you checked out from a module named `cvstest`. The `Beta.h` file is part of a CodeWarrior project named `cvstest.mcp`.

- 1 Open the project in the checked-out module.

Select **File > Open** to display an Open dialog box. Use this dialog box to open the project. In this tutorial, navigate to the `cvstest` folder and open the `cvstest.mcp` project inside the folder, as shown in [Figure 4.7](#).

Figure 4.7 Opening the project in the checked-out module

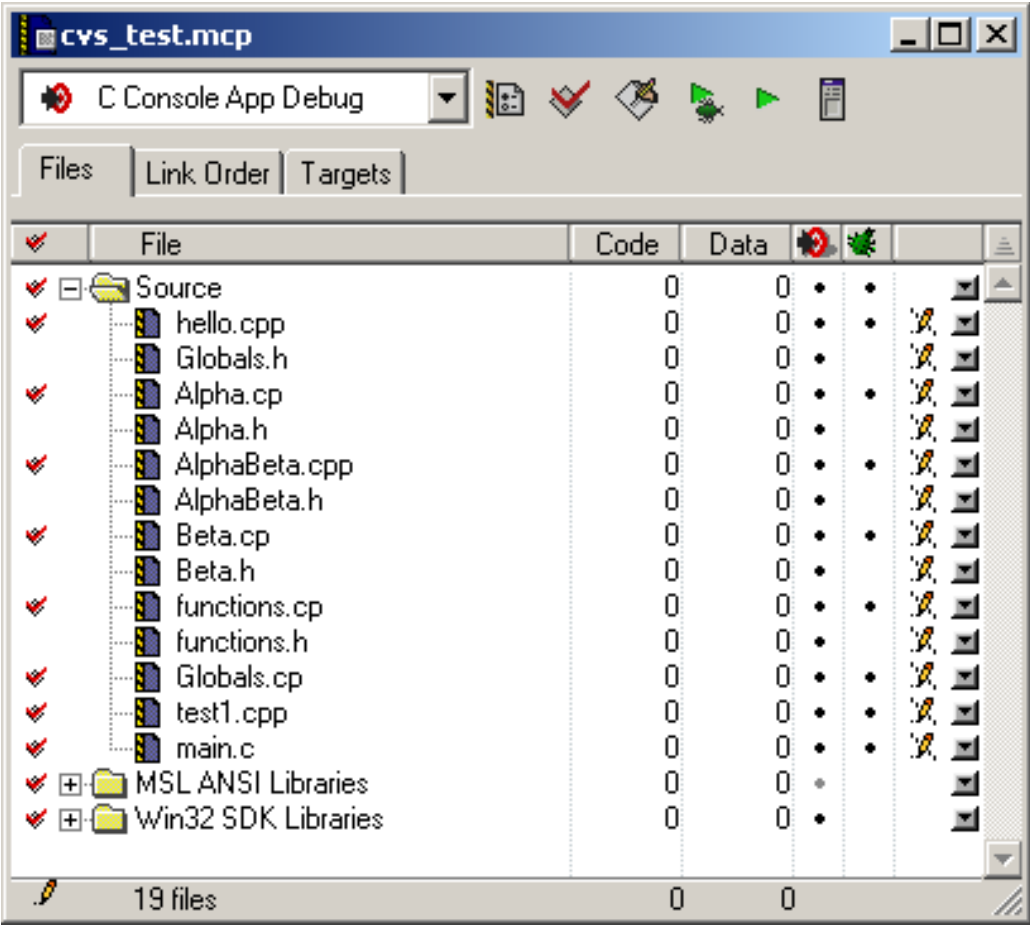


- 2 View the project and its files in the project window that appears.

Figure 4.8 shows the project window for `cvstest.mcp`. Notice the VCS status icon  that appears next to the header file `Beta.h` in the Source group of the project window.

NOTE [Table 3.1 on page 15](#) explains the VCS status icons.

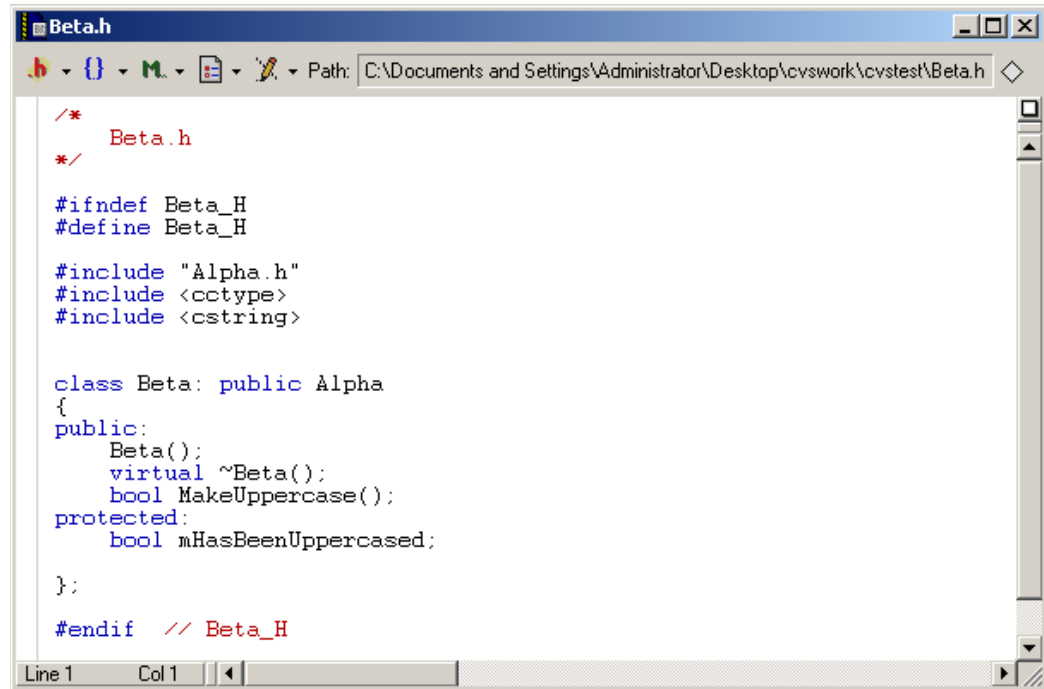
Figure 4.8 Project window



- 3 Open the file that you want to modify.

In this tutorial, open the file named `Beta.h` that appears in the Source group of the project window. This header file appears in an editor window, shown in [Figure 4.9](#).

Figure 4.9 Header file



```
/*
 * Beta.h
 */


#ifndef Beta_H
#define Beta_H

#include "Alpha.h"
#include <cctype>
#include <cstring>

class Beta: public Alpha
{
public:
    Beta();
    virtual ~Beta();
    bool MakeUppercase();
protected:
    bool mHasBeenUppercased;
};

#endif // Beta_H
```

- 4 Modify the file.
- 5 Save your changes.
- 6 Close the editor window.

In the project window, notice that the VCS status icon next to `Beta.h` changes to , indicating that your local copy of that file differs from the version stored in the repository.

NOTE [Table 3.1 on page 15](#) explains the VCS status icons.

Part Five: Committing the Modified File

This part explains how to commit the changes that you made to a file. In this tutorial, you commit the `Beta.h` file that you modified earlier.

- 1 Bring forward the project window that has the edited file.

In this tutorial, bring forward the `cvs_test.mcp` project window.

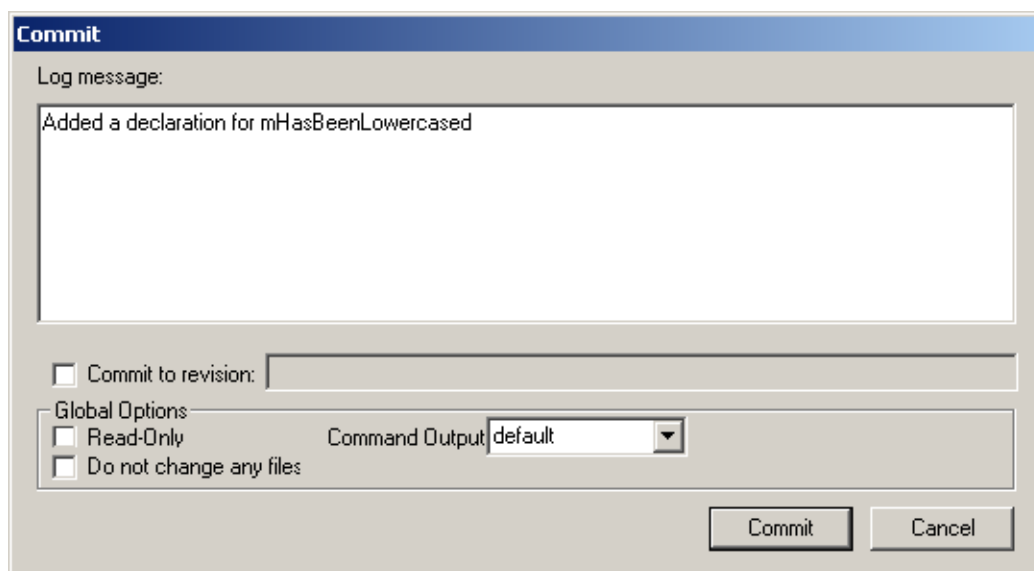
- 2 Select the edited file in the project window.

In this tutorial, select the `Beta.h` header file.

- 3 Select **VCS > Commit**.

Enter a log message in the **Commit** window that appears, as shown in [Figure 4.10](#).

Figure 4.10 Commit window



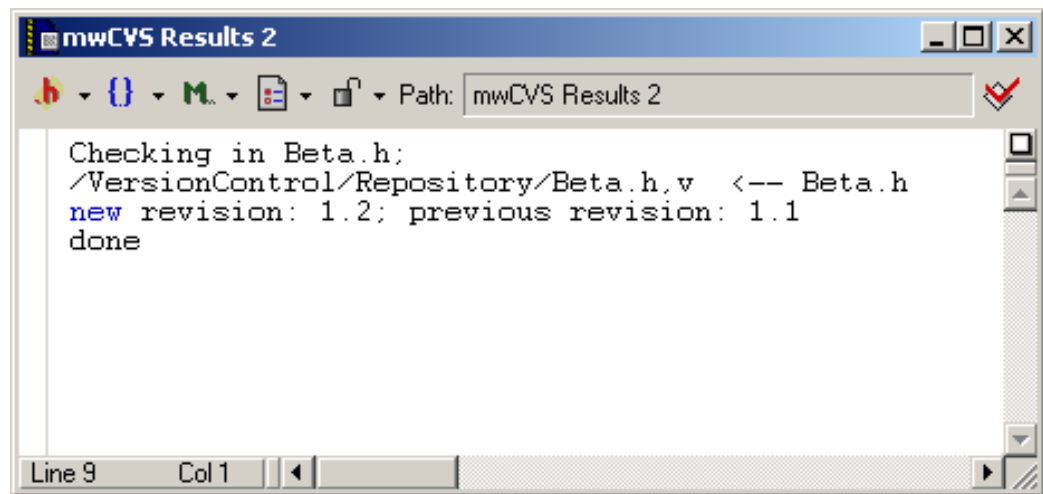
- 4 Click **Commit**.

The Commit window closes and a progress indicator appears. The progress indicator closes after the commit operation completes.

- 5 View commit information in another **mwCVS Results** window that appears.

This window shows the result of the commit operation. In this tutorial, [Figure 4.11](#) shows that mwCVS successfully committed the `Beta.h` file.

Figure 4.11 mwCVS Results window - commit



Congratulations! You successfully completed these tasks with mwCVS, the IDE, and a CVS repository:

- turned on mwCVS
- accessed the repository
- checked out a module from the repository
- modified a file in the checked-out module
- committed the file modifications to the repository

Index

Symbols

- A 20, 37, 52, 59
 - b
 - in Log dialog box 33, 48, 65
 - in Tag dialog box 29, 43, 63
 - C 20, 37, 52
 - c 27
 - D
 - in Checkout dialog box 58
 - in Tag dialog box 29, 43, 62
 - in Update dialog box 20, 37, 52
 - d
 - in Log dialog box 34, 49, 66
 - in Tag dialog box 29, 43, 63
 - F 29, 44, 63
 - f
 - in Checkout dialog box 59
 - in Remove dialog box 32, 47
 - in Tag dialog box 29, 44, 63
 - h 33, 48, 65
 - j 21, 38, 53
 - kb 31, 46
 - l 58, 61, 66
 - m 23, 31, 40, 46, 55
 - N 34, 49, 66
 - n
 - in Checkout dialog box 59, 60
 - in Commit dialog box 24, 41, 56
 - in Tag dialog box 30, 44, 45, 64
 - in Update dialog box 22, 39, 54
 - P 58
 - Q
 - in Checkout dialog box 60
 - in Commit dialog box 24, 41, 56
 - in mwCVS Panel 13
 - in Tag dialog box 30, 45, 64
 - in Update dialog box 22, 39, 54
 - q
 - in Checkout dialog box 60
 - in Commit dialog box 24, 41, 56
 - in mwCVS Panel 13
 - in Tag dialog box 30, 45, 64
 - in Update dialog box 22, 39, 54
 - R 34, 49, 65
 - r 61
 - in Checkout dialog box 58, 59
 - in Commit dialog box 23, 40, 55, 56
 - in Log dialog box 34, 49, 66
 - in mwCVS Panel 12
 - in Tag dialog box
 - Read-only 30, 44, 64
 - Tag revision 28, 43, 62
 - in Update dialog box
 - Read-only 21, 38, 53
 - Retrieve revision 20, 37, 52
 - s 34, 49, 66
 - u 27
 - v 24, 41, 61
 - w
 - in Checkout dialog box 59
 - in Commit dialog box 23, 40, 56
 - in Log dialog box 34, 49, 66
 - in Update dialog box 21, 38, 53
 - z9 12
(Logging into remoteserver) 36
- ## A
- A 20, 37, 52, 59
 - about
 - CVS 6
 - mwCVS 5
 - accessing the repository, tutorial 69
 - Add dialog box
 - Binary 31, 46
 - Log message 31, 46
 - Text 31, 46
 - Add menu command 30, 45
 - and this Revision 21, 38, 53
- ## B
- b
 - in Log dialog box 33, 48, 65
 - in Tag dialog box 29, 43, 63
 - Binary 31, 46
 - button, Select
 - in Checkout dialog box 57
 - in Commit dialog box 55
 - in Log dialog box 65

- in Status dialog box 61
- in Tag dialog box 62
- in Update dialog box 51

C

- C 20, 37, 52

- c 27

- checking out a module, tutorial 72

- Checkout dialog box

 - Checkout to 57

 - Command Output

 - Default 60

 - Quiet 60

 - Very Quiet 60

 - CVSRoot 58

 - Do not change any files 60

 - Don't run module program 59

 - Force head revision if date/tag not found 59

 - Module 57

 - Prune empty directories 58

 - Read-Only 59

 - Recursive 58

 - Reset sticky tags 59

 - Use date 58

 - Use revision 58

- Checkout menu command 56

- Checkout to 57

- CodeWarrior Diff 25

- Command Output

 - Default

 - in Checkout dialog box 60

 - in Commit dialog box 24, 41, 56

 - in mwCVS Panel 13

 - in Tag dialog box 30, 45, 64

 - in Update dialog box 22, 39, 54

 - in Commit dialog box 24, 41, 56

 - in mwCVS Panel 13

 - in Tag dialog box 30, 45, 64

 - in Update dialog box 22, 39, 54

 - Quiet

 - in Checkout dialog box 60

 - in Commit dialog box 24, 41, 56

 - in mwCVS Panel 13

 - in Tag dialog box 30, 45, 64

 - in Update dialog box 22, 39, 54

 - Very Quiet

 - in Checkout dialog box 60

 - in Commit dialog box 24, 41, 56

 - in mwCVS Panel 13

 - in Tag dialog box 30, 45, 64

 - in Update dialog box 22, 39, 54

- commands

 - Add 30, 45

 - Checkout 56

 - Commit 22, 39, 54

 - context 17

 - Difference 25

 - for files 19

 - for projects 36

 - for the version control system 18

 - Log 33, 48, 64

 - Login 35

 - recursive 50

 - Remove 31, 46

 - remove 5, 31, 46

 - Status 24, 41, 60

 - Synchronize Selection 18

 - Synchronize Status 18

 - Tag 28, 42, 61

 - Update 19, 36, 51

- Commit dialog box

 - Command Output 24, 41, 56

 - Default 24, 41, 56

 - Quiet 24, 41, 56

 - Very Quiet 24, 41, 56

 - Commit folder 55

 - Commit to revision 23, 40, 55

 - Do not change any files 24, 41, 56

 - Log message 23, 40, 55

 - Read-Only 23, 40, 56

 - Recursive 55

- Commit folder 55

- Commit menu command 22, 39, 54

- Commit removal immediately 32, 47

- Commit to revision 23, 40, 55

- committing the modified file, tutorial 78

- concepts of CVS 6

- configuring mwCVS 11, 15

- context menus for CVS commands 17

- Contextual 27

- Create branch 29, 43, 63

- CVS

 - context menu commands 17

 - learning more about 6

 - status icons in project window 15

- CVSREAD 12

CVSROOT 73
CVSRoot 35, 58

D

-D
 in Checkout dialog box 58
 in Tag dialog box 29, 43, 62
 in Update dialog box 20, 37, 52
-d
 in Log dialog box 34, 49, 66
 in Tag dialog box 29, 43, 63
Date 34, 49, 66
date formatting 20, 37, 52
Default 13
Default branch revisions 33, 48, 65
Delete tag 29, 43, 63
Diff dialog box
 CodeWarrior Diff 25
 Contextual 27
 Diff local file with a specific revision(s) 26
 Date 26
 Revision/Tag 26
 Diff local file with same revision in CVS 26
 Diff two remote revisions 27
 Date 26
 Revision/Tag 26
 GNU Diff 25
 Universal 27
Diff local file with a specific revision(s) 26
 Date 26
 Revision/Tag 26
Diff local file with same revision in CVS 26
Diff two remote revisions 27
 Date 26
 Revision/Tag 26
Difference menu command 25
disabling mwCVS 10
Do not change any files
 in Checkout dialog box 60
 in Commit dialog box 24, 41, 56
 in Tag dialog box 30, 45, 64
 in Update dialog box 22, 39, 54
Do not execute tag program 30, 44
Do not list tags 34, 49, 66
Don't run module program 59

E

enabling mwCVS 9

F

-F 29, 44, 63
-f
 in Checkout dialog box 59
 in Remove dialog box 32, 47
 in Tag dialog box 29, 44, 63
file commands
 Add 30
 Commit 22
 Difference 25
 Log 33
 Login 35
 Remove 31
 Status 24
 Synchronize Selection 18
 Synchronize Status 18
 Tag 28
 Update 19
file options
 and this Revision 21
 Binary 31
 CodeWarrior Diff 25
 Command Output 22, 24, 30
 Default 22, 24, 30
 Quiet 22, 24, 30
 Very Quiet 22, 24, 30
 Commit removal immediately 32
 Commit to revision 23
 Contextual 27
 Create branch 29
 CVSRoot 35
 Date 34
 Default branch revisions 33
 Delete tag 29
 Diff local file with a specific revision(s) 26
 Date 26
 Revision/Tag 26
 Diff local file with same revision in CVS 26
 Diff two remote revisions 27
 Date 26
 Revision/Tag 26
 Do not change any files 22, 24, 30
 Do not execute tag program 30
 Do not list tags 34
 GNU Diff 25

- Headers 33
- Include tag information 24
- Log message 23, 31
- Move tag if it exists 29
- Overwrite locally modified file 20
- Password 35
- RCS file names 34
- Read-Only 21, 23, 30
- Remove local file 32
- Reset sticky tags 20
- Retrieve before date 20
- Retrieve revision 20
- Revision 21
- Revisions 34
- State(s) 34
- Tag as of date 29
- Tag latest head revision if tag/date is not found 29
- Tag name 28
- Tag revision 28
- Text 31
- Universal 27
- User(s) 34

files, commands for 19

Force head revision if date/tag not found 59

formatting of dates 20, 37, 52

G

general CVS information 6

GNU Diff 25

H

-h 33, 48, 65

Headers 33, 48, 65

how to

- disable mwCVS 10
- enable mwCVS 9
- install mwCVS on a Macintosh host 8
- install mwCVS on a Windows host 7
- turn off mwCVS 10
- turn on mwCVS 9

I

icon for VCS menu on a Macintosh host 10, 68

Include tag information 24, 41, 61

information about CVS 6

installation of mwCVS 7

installing mwCVS

on a Macintosh host 8

on a Windows host 7

J

-j 21, 38, 53

K

-kb 31, 46

L

-l 58, 61, 66

learning about CVS 6

Log dialog box

- Date 34, 49, 66
- Default branch revisions 33, 48, 65
- Do not list tags 34, 49, 66
- Headers 33, 48, 65
- Log folder 65
- RCS file names 34, 49, 65
- Recursive 66
- Revisions 34, 49, 66
- State(s) 34, 49, 66
- User(s) 34, 49, 66

Log folder 65

Log menu command 33, 48, 64

Log message 23, 31, 40, 46, 55

(Logging into remoteserver) 36

Login dialog box

- CVSRoot 35
- Password 35

Login menu command 35

M

-m 23, 31, 40, 46, 55

Macintosh VCS menu icon 10, 68

Make New working files read-only 12

menu commands

- Add 30, 45
- Checkout 56
- Commit 22, 39, 54
- Difference 25
- Log 33, 48, 64
- Login 35
- Remove 31, 46
- Status 24, 41, 60
- Synchronize Selection 18

- Synchronize Status 18
- Tag 28, 42, 61
- Update 19, 36, 51
- menu, context 17
- modifying a file, tutorial 75
- Module 57
- more information about CVS 6
- Move tag if it exists 29, 44, 63
- mwCVS
 - about 5
 - configuring 11, 15
 - disabling 10
 - enabling 9
 - file commands 19
 - installation 7
 - installing on a Macintosh host 8
 - installing on a Windows host 7
 - options
 - Command Output 13
 - Make New working files read-only 12
 - Use gZip level 9 compression 12
 - overview 5
 - project commands 36
 - recursive commands 50
 - system requirements 6
 - turning off 10
 - turning on 9
 - tutorial 67
 - accessing the repository 69
 - checking out a module 72
 - committing the modified file 78
 - modifying a file 75
 - turning on mwCVS 68
 - version control system commands 18
- mwCVS remove 5, 31, 46
- mwCVS Results window 56, 74, 79

N

- N 34, 49, 66
- n
 - in Checkout dialog box 59, 60
 - in Commit dialog box 24, 41, 56
 - in Tag dialog box 30, 44, 45, 64
 - in Update dialog box 22, 39, 54

O

- overview of mwCVS 5

- Overwrite locally modified file 20, 37, 52

P

- P 58
- Password 35
- password server 69
- project commands
 - Add 45
 - Commit 39
 - Log 48
 - Remove 46
 - Status 41
 - Tag 42
 - Update 36
- project options
 - and this Revision 38
 - Binary 46
 - Command Output 39, 41, 45
 - Default 39, 41, 45
 - Quiet 39, 41, 45
 - Very Quiet 39, 41, 45
 - Commit removal immediately 47
 - Commit to revision 40
 - Create branch 43
 - Date 49
 - Default branch revisions 48
 - Delete tag 43
 - Do not change any files 39, 41, 45
 - Do not execute tag program 44
 - Do not list tags 49
 - Headers 48
 - Include tag information 41
 - Log message 40, 46
 - Move tag if it exists 44
 - Overwrite locally modified file 37
 - RCS file names 49
 - Read-Only 38, 40, 44
 - Remove local file 47
 - Reset sticky tags 37
 - Retrieve before date 37
 - Retrieve revision 37
 - Revision 38
 - Revisions 49
 - State(s) 49
 - Tag as of date 43
 - Tag latest head revision if tag/date is not found 44
 - Tag name 43
 - Tag revision 43

- Text 46
- User(s) 49
- project window, CVS status icons in 15
- projects, commands for 36
- Prune empty directories 58
- pserver 69

Q

- Q
 - in Checkout dialog box 60
 - in Commit dialog box 24, 41, 56
 - in mwCVS Panel 13
 - in Tag dialog box 30, 45, 64
 - in Update dialog box 22, 39, 54
- q
 - in Checkout dialog box 60
 - in Commit dialog box 24, 41, 56
 - in mwCVS Panel 13
 - in Tag dialog box 30, 45, 64
 - in Update dialog box 22, 39, 54
- Quiet 13

R

- R 34, 49, 65
- r 61
 - in Checkout dialog box 58, 59
 - in Commit dialog box 23, 40, 55, 56
 - in Log dialog box 34, 49, 66
 - in mwCVS Panel 12
 - in Tag dialog box
 - Read-only 30, 44, 64
 - Tag revision 28, 43, 62
 - in Update dialog box
 - Read-only 21, 38, 53
 - Retrieve revision 20, 37, 52
- RCS 34, 49, 65
- RCS file names 34, 49, 65
- Read-Only
 - in Checkout dialog box 59
 - in Commit dialog box 23, 40, 56
 - in Tag dialog box 30, 44, 64
 - in Update dialog box 21, 38, 53
- Recursive
 - in Checkout dialog box 58
 - in Commit dialog box 55
 - in Log dialog box 66
 - in Status dialog box 61

- in Tag dialog box 63
- in Update dialog box 51
- recursive commands 50
 - Checkout 56
 - Commit 54
 - Log 64
 - Status 60
 - Tag 61
 - Update 51
- recursive options
 - and this Revision 53
 - Checkout to 57
 - Command Output 54, 56, 64
 - Default 54, 56, 60, 64
 - Quiet 54, 56, 60, 64
 - Very Quiet 54, 56, 60, 64
 - Commit folder 55
 - Commit to revision 55
 - Create branch 63
 - CVSRoot 58
 - Date 66
 - Default branch revisions 65
 - Delete tag 63
 - Do not change any files 54, 56, 60, 64
 - Do not list tags 66
 - Don't run module program 59
 - Force head revision if date/tag not found 59
 - Headers 65
 - Include tag information 61
 - Log folder 65
 - Log message 55
 - Module 57
 - Move tag if it exists 63
 - Overwrite locally modified file 52
 - Prune empty directories 58
 - RCS file names 65
 - Read-Only 53, 56, 59, 64
 - Recursive
 - in Checkout dialog box 58
 - in Commit dialog box 55
 - in Log dialog box 66
 - in Status dialog box 61
 - in Tag dialog box 63
 - in Update dialog box 51
 - Reset sticky tags 52, 59
 - Retrieve before date 52
 - Retrieve revision 52
 - Revision 53
 - Revisions 66

- State(s) 66
- Status folder 61
- Tag as of date 62
- Tag folder 62
- Tag latest head revision if tag/date is not found 63
- Tag name 62
- Tag revision 62
- Update folder 51
- Use date 58
- Use revision 58
- User(s) 66
- remove command, differences from GNU CVS implementation 5, 31, 46
- Remove dialog box
 - Commit removal immediately 32, 47
 - Remove local file 32, 47
- Remove local file 32, 47
- Remove menu command 31, 46
- requirements for using mwCVS 6
- Reset sticky tags 20, 37, 52, 59
- resources for CVS 6
- Retrieve before date 20, 37, 52
- Retrieve revision 20, 37, 52
- Revision 21, 38, 53
- Revisions 34, 49, 66

S

- s 34, 49, 66
- sample date formatting 20, 37, 52
- Select button
 - in Checkout dialog box 57
 - in Commit dialog box 55
 - in Log dialog box 65
 - in Status dialog box 61
 - in Tag dialog box 62
 - in Update dialog box 51
- settings
 - Command Output 13
 - Make New working files read-only 12
 - Use gZip level 9 compression 12
- State(s) 34, 49, 66
- Status dialog box
 - Include tag information 24, 41, 61
 - Recursive 61
 - Status folder 61
- Status folder 61
- status icons in project window 15

- Status menu command 24, 41, 60
- Synchronize Selection menu command 18
- Synchronize Status menu command 18
- system requirements 6

T

- Tag as of date 29, 43, 62
- Tag dialog box
 - Command Output 30, 45, 64
 - Default 30, 45, 64
 - Quiet 30, 45, 64
 - Very Quiet 30, 45, 64
 - Create branch 29, 43, 63
 - Delete tag 29, 43, 63
 - Do not change any files 30, 45, 64
 - Do not execute tag program 30, 44
 - Move tag if it exists 29, 44, 63
 - Read-Only 30, 44, 64
 - Recursive 63
 - Tag as of date 29, 43, 62
 - Tag folder 62
 - Tag latest head revision if tag/date is not found 29, 44, 63
 - Tag name 28, 43, 62
 - Tag revision 28, 43, 62
- Tag folder 62
- Tag latest head revision if tag/date is not found 29, 44, 63
- Tag menu command 28, 42, 61
- Tag name 28, 43, 62
- Tag revision 28, 43, 62
- tasks
 - disabling mwCVS 10
 - enabling mwCVS 9
 - installing mwCVS on a Macintosh host 8
 - installing mwCVS on a Windows host 7
 - turning off mwCVS 10
 - turning on mwCVS 9
- Text 31, 46
- turning off mwCVS 10
- turning on mwCVS 9
- turning on mwCVS, tutorial 68
- tutorial 67
 - accessing the repository 69
 - Macintosh 70
 - Windows 69
 - checking out a module 72

committing the modified file 78
modifying a file 75
turning on mwCVS 68

VCS Messages 70

Z

-z9 12

U

-u 27

Universal 27

Update dialog box

and this Revision 21, 38, 53

Command Output 22, 39, 54

Default 22, 39, 54

Quiet 22, 39, 54

Very Quiet 22, 39, 54

Do not change any files 22, 39, 54

Overwrite locally modified file 20, 37, 52

Read-Only 21, 38, 53

Recursive 51

Reset sticky tags 20, 37, 52

Retrieve before date 20, 37, 52

Retrieve revision 20, 37, 52

Revision 21, 38, 53

Update folder 51

Update folder 51

Update menu command 19, 36, 51

Use date 58

Use gZip level 9 compression 12

Use revision 58

User(s) 34, 49, 66

V

-v 24, 41, 61

valid date formatting 20, 37, 52

VCS menu on a Macintosh host 10, 68

VCS Messages window 36, 70

version control system, commands for 18

Very Quiet 13

W

-w

in Checkout dialog box 59

in Commit dialog box 23, 40, 56

in Log dialog box 34, 49, 66

in Update dialog box 21, 38, 53

windows

mwCVS Results

for Checkout 56, 74

for Commit 79