

CodeWarrior™

Version Control Plug-in

1.3

for CVS

Revised: 23 April 2008



Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. CodeWarrior is a trademark or registered trademark of Freescale Semiconductor, Inc. in the United States and/or other countries. All other product or service names are the property of their respective owners.

Copyright © 2003–2008 by Freescale Semiconductor, Inc. All rights reserved.

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including “Typicals”, must be validated for each customer application by customer's technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

How to Contact Us

Corporate Headquarters	Freescale Semiconductor, Inc. 7700 West Parmer Lane Austin, TX 78729 U.S.A.
World Wide Web	http://www.freescale.com/codewarrior
Technical Support	http://www.freescale.com/support

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	11
2	Configuring mwCVS	13
	mwCVS Panel	13
	mwCVS Panel Options	15
	Make new working files read-only	15
	Use gZip level 9 compression	15
	Command Output	15
	Compatible with WinCVS 1.2	16
	Integrating mwCVS with ssh	16
3	Using mwCVS	21
	Project Window	21
	Context Menus	23
	IDE Version Control System (VCS) Commands	24
	Synchronize Status	24
	Synchronize Selection/Document	25
	File Commands	26
	Update	26
	Commit	30
	Status	32
	Difference	33
	Tag	36
	Add	39
	Remove	40
	Log	41

Table of Contents

Login	43
Project Commands	44
Update	44
Commit	48
Status	50
Tag	51
Add	54
Remove	55
Log	56
Recursive Commands	58
Update	59
Commit	63
Checkout	66
Status	70
Tag	71
Log	74
4 mwCVS Tutorial	77
Part One: Turning on mwCVS	78
Part Two: Accessing the Repository	78
Windows	78
Macintosh	80
Part Three: Checking out a Module	81
Part Four: Modifying a File	84
Part Five: Committing the Modified File	87
Index	91

mwCVS Overview and Setup

This chapter is an overview of the CodeWarrior™ 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 11](#)

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

<http://cvsgui.sourceforge.net/>

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:

NOTE These directions apply to version 1.3 of the mwCVS plug-in.

1. Find the path to your CodeWarrior directory. Subsequent steps refer to this path as the *CodeWarriorDirPath*.

TIP On most Windows hosts, this is the default path to the CodeWarrior directory:
C:\Program Files\Freescale\CodeWarrior\

2. Place the file mwcv.s.dll in:
CodeWarriorDirPath\Bin\Plugins\version control\
3. Place the file pMWCVS.dll in:
CodeWarriorDirPath\Bin\Plugins\Preference Panel\
4. Place the file cvs.exe in:
CodeWarriorDirPath\(Helper Apps)\

mwCVS Overview and Setup

mwCVS Installation

5. Place the file `cwspawn.exe` in:
`CodeWarriorDir\ (Helper Apps) \`

Installing mwCVS on a Macintosh Host

NOTE Unlike the Windows version, the Mac OS X version of mwCVS does not include its own version of the `cvs` executable file. 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 the path to your CodeWarrior folder. Subsequent steps refer to this path as the *CodeWarriorFldrPath*.

NOTE On Macintosh hosts, this is the default path to the CodeWarrior folder, where *StartupDiskName* is the name of your startup disk, and *VersionNumber* is the version number of your CodeWarrior product:
`StartupDiskName:Applications:Freescale CodeWarrior`
`VersionNumber:Freescale CodeWarrior:.`

2. Place the file `mwCVS` in:
`CodeWarriorFldrPath:CodeWarrior Plugins:Version Control:`
3. Place the file `mwCVS Panel` in:
`CodeWarriorFldrPath: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

You can turn on mwCVS in the IDE for different scopes of operation. [Table 1.2](#) explains these scopes.

Table 1.2 mwCVS Scopes

Scope	Explanation
Global	To turn on this scope, first close all projects in the IDE. Configuring mwCVS in the global scope allows individual projects to inherit their VCS settings from the global settings that you specify (when you check the Use global settings as default checkbox).
Project	To turn on this scope, first bring forward a project. Configuring mwCVS in the project scope allows each project to have different VCS settings. These project-level VCS settings override the global settings that you specify.

Turning On mwCVS in the Global Scope

Follow these steps to turn on mwCVS in the global scope:

1. Close all open projects in the IDE.
After you close all projects, you can modify mwCVS settings in the global scope.
2. Choose **Edit > Version Control Settings**.
The **VCS Settings** window appears.
3. Select **VCS Setup** in the VCS Settings Panels list.
The VCS Setup panel appears.

NOTE Only three options in the VCS Setup panel apply to mwCVS operation:
Use Version Control, **Method**, and **Use global settings as default**.

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.

NOTE If you want projects to inherit their VCS settings from the global settings that you specify, check the **Use global settings as default** checkbox.

mwCVS Overview and Setup

Turning On mwCVS

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 a Macintosh host.

Turning On mwCVS in the Project Scope

Follow these steps to turn on mwCVS in the project scope:

1. Bring forward the project for which you want to turn on mwCVS.

The mwCVS settings you specify apply to this project only.

2. Choose **Edit > Version Control Settings**.

The **VCS Settings** window appears.

3. 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**.

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 a Macintosh host.

Turning Off mwCVS

You can turn off mwCVS in the IDE for different scopes of operation. [Table 1.3](#) explains these scopes.

Table 1.3 mwCVS Scopes

Scope	Explanation
Global	To turn off this scope, first close all projects in the IDE. Turning off mwCVS in the global scope prevents individual projects from inheriting global VCS settings.
Project	<p>To turn off this scope, first bring forward a project. Turning off mwCVS for one project does not affect the mwCVS state of other projects.</p> <p>You can turn on mwCVS globally, but turn it off for individual projects. This configuration is useful when you want to keep the global VCS settings you specified, but disable them in an individual project.</p>

Turning Off mwCVS in the Global Scope

Follow these steps to turn off mwCVS in the global scope:

1. Close all open projects in the IDE.
After you close all projects, you can modify mwCVS settings in the global scope.
2. Choose **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 **None** from the **Method** list box.
The mwCVS item disappears from the VCS Settings Panels list.
5. Clear the **Use Version Control** option.
6. Click **Apply** or **Save** to confirm your changes.
7. Close the VCS Settings window.

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

Turning Off mwCVS in the Project Scope

Follow these steps to turn off mwCVS in the project scope:

NOTE If you checked the **Use global settings as default** checkbox in the global scope, you can keep mwCVS turned on globally, but turn it off for a given project. This way, other projects can still inherit the global settings, while the project for which you turn off mwCVS does not inherit the global settings.

1. Bring forward the project for which you want to turn off mwCVS.
The mwCVS settings you specify apply to this project only.
 2. Choose **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 **None** from the **Method** list box.
The mwCVS item disappears from the VCS Settings Panels list.
 5. Clear the **Use Version Control** option.
 6. Click **Apply** or **Save** to confirm your changes.
 7. 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. Also, this chapter explains how to configure mwCVS to work together with a secure shell (ssh).

This chapter has these topics:

- [“mwCVS Panel” on page 13](#)
- [“mwCVS Panel Options” on page 15](#)
- [“Integrating mwCVS with ssh” on page 16](#)

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 14](#) briefly explains the panel options. See [“mwCVS Panel Options” on page 15](#) for more detailed explanations.

NOTE You do not specify the local path in the **mwCVS** panel. Instead, mwCVS uses the local path of each and every project file.

Configuring mwCVS

mwCVS Panel

Figure 2.1 mwCVS panel

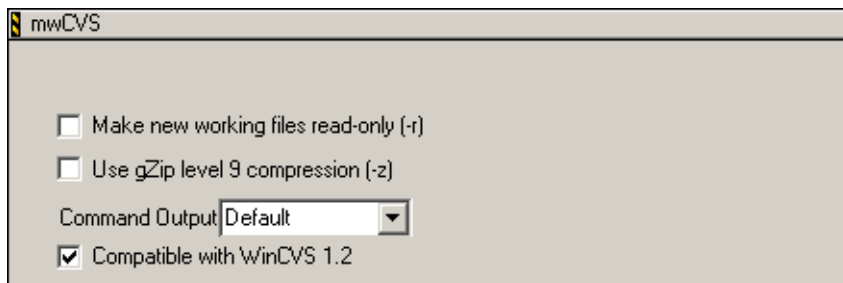


Table 2.1 mwCVS panel options

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 (-z)	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.
Compatible with WinCVS 1.2	Select to prevent Daylight Saving Time issues from affecting CVS operations with WinCVS 1.2.

mwCVS Panel Options

This section explains mwCVS panel options in detail. 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 `-z`.

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`.

Compatible with WinCVS 1.2

Select this option to prevent Daylight Saving Time issues from affecting CVS operations with WinCVS version 1.2.

NOTE If you use WinCVS version 1.3 beta, or a newer version, clear this option.

Integrating mwCVS with ssh

This section explains how to set up mwCVS to work together with a secure shell to access a CVS repository on a Linux-based computer. To follow the steps in this section, you should be familiar with `ssh` and its use of public- and private-key encryption.

Integrating mwCVS with ssh on a Linux Host

Follow these steps to integrate mwCVS with `ssh`:

1. Generate an `ssh` public and private key pair by entering this command line in a terminal window, where the `-t` flag specifies the key type and the `-C` flag specifies a key comment (such as your user name and host name):

```
ssh-keygen -t dsa -C user@host
```
2. Press the Enter key at the prompt `Enter file in which to save the key.`
3. Press the Enter key at the prompt `Enter passphrase.`
The public key `id_dsa.pub` and the private key `id_dsa` appear in `~/.ssh`.

NOTE For security, set the permissions on `~/.ssh` to 0600 (readable and writable by you only).

4. Add the content of your *public* key (`id_dsa.pub`) to the `authorized_keys` file on the computer that hosts the CVS repository. The content of your public key is the line that begins with the text `ssh-dss`.

NOTE If the CVS repository uses SSH2, add the content of your public key to the `authorized_keys2` file instead of the `authorized_keys` file.

The content of your public key should become a new line at the end of `cvswowner/.ssh/authorized_keys` or `cvswowner/.ssh/authorized_keys2` on the CVS repository host, where `cvswowner` is the name of the user account that owns `CVSROOT`.

NOTE *Never share your private key (id_dsa).*

5. Try accessing the CVS repository by entering this command line in a terminal window, where `pathtokey` is the local path to your private key, `cvsowner` is the name of the user account on the CVS repository host that owns CVSROOT, and `serveraddr` is the address of the CVS repository host:

```
ssh -i pathtokey -l cvsowner serveraddr
```

After you successfully use `ssh` to access the CVS repository, continue with the remainder of this procedure.

6. Configure local CVS environment variables to have mwCVS use `ssh`. Add these lines to your `~/ .bashrc` file, where `user` is your CVS login name, `serveraddr` is the address of the CVS repository host, and `pathtocvsroot` is the path to the CVSROOT of the CVS repository host:

```
CVS_RSH="ssh"
```

```
CVSROOT=":ext:user@serveraddr:/pathtocvsroot"
```

```
export CVS_RSH CVSROOT
```

7. Try checking out a module from the CVS repository by entering this command line in a terminal window, where `module` is the path to the module that you want to check out:

```
cvs -d $CVSROOT checkout module
```

After you successfully check out a module in the last step, you have completed integrating mwCVS with `ssh`. See [“mwCVS Overview and Setup” on page 5](#) for more information on turning mwCVS on and off.

Integrating mwCVS with ssh on a Windows Host

Follow these steps to integrate mwCVS with `ssh`:

1. Install OpenSSH for Windows. The installer is available at this URL:

<http://sshhwindows.sourceforge.net/download/>

NOTE If you do not plan to use `ssh` to log into your Windows computer from another computer, you can ignore the installation message regarding the `passwd` file.

2. Select **Start > Programs > Accessories > Command Prompt**.

A terminal window appears.

Configuring mwCVS

Integrating mwCVS with ssh

3. Generate an ssh public and private key pair by entering this command line in the terminal window, where the `-t` flag specifies the key type and the `-C` flag specifies a key comment (such as your user name and host name):

```
ssh-keygen -t dsa -C user@host
```

4. Press the Enter key at the prompt `Enter file in which to save the key.`
5. Press the Enter key at the prompt `Enter passphrase.`

The public key `id_dsa.pub` and the private key `id_dsa` appear in this directory, where `loginname` is the name you use to log in to your Windows computer:

```
C:\Documents and Settings\loginname\.ssh
```

NOTE For security, set the permissions on the `.ssh` directory so that only you can read and write to it.

6. Add the content of your *public* key (`id_dsa.pub`) to the `authorized_keys` file on the computer that hosts the CVS repository. The content of your public key is the line that begins with the text `ssh-dss`.

NOTE If the CVS repository uses SSH2, add the content of your public key to the `authorized_keys2` file instead of the `authorized_keys` file.

The content of your public key should become a new line at the end of `cvswowner/.ssh/authorized_keys` or `cvswowner/.ssh/authorized_keys2` on the CVS repository host, where `cvswowner` is the name of the user account that owns `CVSROOT`.

NOTE *Never* share your *private* key (`id_dsa`).

7. Try accessing the CVS repository by entering this command line in a terminal window, where `loginname` is the name you use to log in to your Windows computer, `cvswowner` is the name of the user account on the CVS repository host that owns `CVSROOT`, and `serveraddr` is the address of the CVS repository host:

```
ssh -i /home/loginname/.ssh/id_dsa -l cvswowner serveraddr
```

After you successfully use `ssh` to access the CVS repository, continue with the remainder of this procedure.

8. Select **Start > Settings > Control Panel**.
9. Open the **System** control panel.
The **System Properties** window appears.
10. Click the **Advanced** tab of the System Properties window.

11. Click the **Environment Variables** button.
The **Environment Variables** window appears.
 12. In the **System variables** list, click the **New** button.
The **New System Variable** dialog box appears.
 13. Enter CVS_RSH in the **Variable name** text box.
 14. Enter ssh in the **Variable value** text box.
 15. Click the **OK** button.
 16. In the **System variables** list, click the **New** button again.
 17. Enter CVSROOT in the **Variable name** text box.
 18. In the **Variable value** text box, enter following text, where user is your CVS login name, serveraddr is the address of the CVS repository host, and pathtocvsroot is the path to the CVSROOT of the CVS repository host:
`:ext:user@serveraddr:/pathtocvsroot`
 19. Click the **OK** button.
 20. Click the **OK** button in the Environment Variables window.
 21. Click the **OK** button in the System Properties window.
 22. Try using the CodeWarrior IDE to check out a module from the CVS repository.
- After you successfully check out a module in the last step, you have completed integrating mwCVS with ssh. See [“mwCVS Overview and Setup” on page 5](#) for more information on turning mwCVS on and off.

Configuring mwCVS

Integrating mwCVS with ssh

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 21](#)
- [“Context Menus” on page 23](#)
- [“IDE Version Control System \(VCS\) Commands” on page 24](#)
- [“File Commands” on page 26](#)
- [“Project Commands” on page 44](#)
- [“Recursive Commands” on page 58](#)

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 23](#) shows a sample project window with CVS status icons. [Table 3.1 on page 22](#) explains these icons.

Table 3.1 CVS status icons






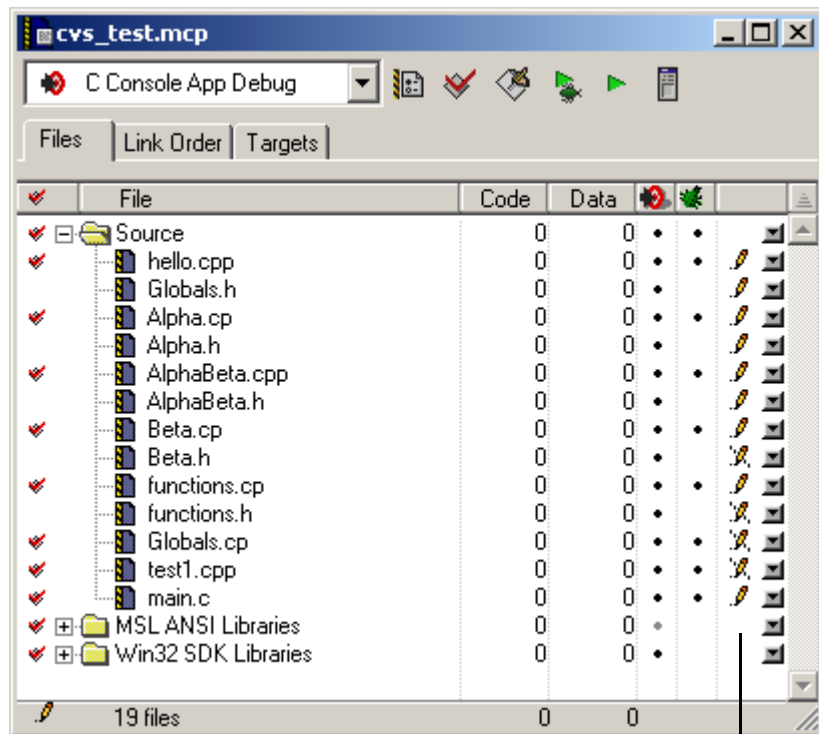
Icon	File is in the CVS Repository	File is not in the CVS Repository	File is Writable	File is Read Only	File has changed since the last Update	File has not changed since the last Update
		•	•		Not applicable	Not applicable
		•		•	Not applicable	Not applicable
	•			•		•
	•		•			•
	•		•		•	

Figure 3.1 CVS status icons in the project window



CVS Status Icons

Context Menus

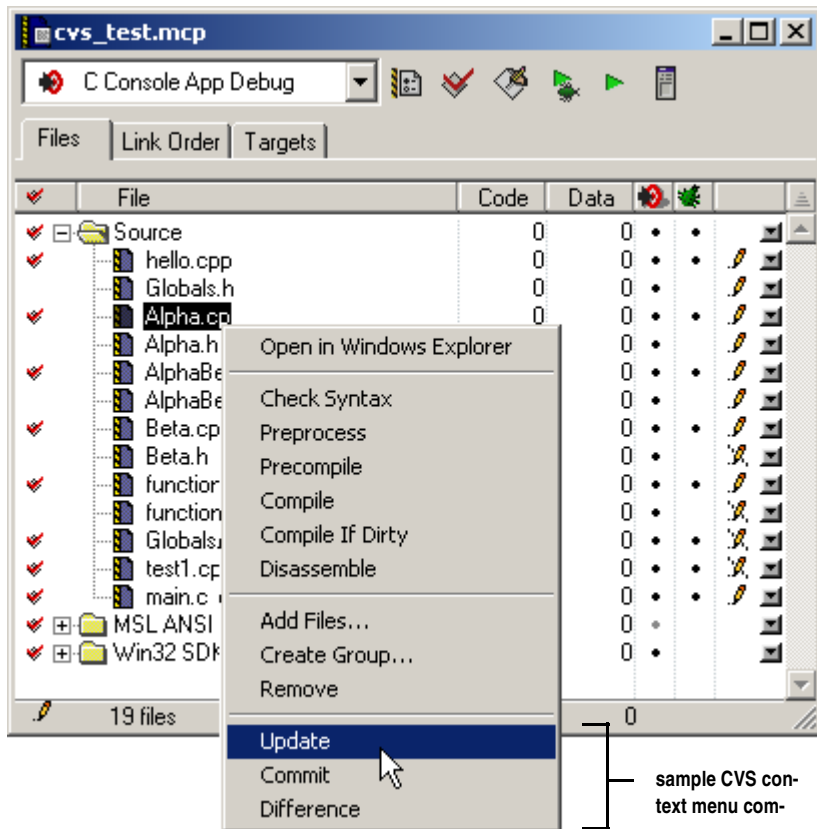
mwCVS adds CVS commands to the IDE context menu. The commands that appear in the context menu depend on the items that you select. Use these CVS context-menu commands as shortcuts for common commands in the VCS menu.

[Figure 3.2 on page 24](#) shows a sample IDE context menu that has CVS commands.

Using mwCVS

IDE Version Control System (VCS) 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 22](#) for more information about the CVS status icons that appear in the project window.

Synchronize Selection/Document

Select **VCS > Synchronize Selection** to refresh the CVS status icon of the selected items in the project window. If you select a group, the operation applies to all items in that group. Select **VCS > Synchronize Document** to refresh the CVS status icon of the active editor window. After you select this command, mwCVS analyzes CVS information about the selected items. 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 icons that appear in the project window.

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

See [Table 3.1 on page 22](#) for more information about the CVS status icons that appear in the project window.

NOTE (Macintosh) Hold down the Shift key in order to select **VCS > Synchronize Selection** or **VCS > Synchronize Document**.

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 27](#) 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 15](#) for more information.

Figure 3.3 Update dialog box (for files)

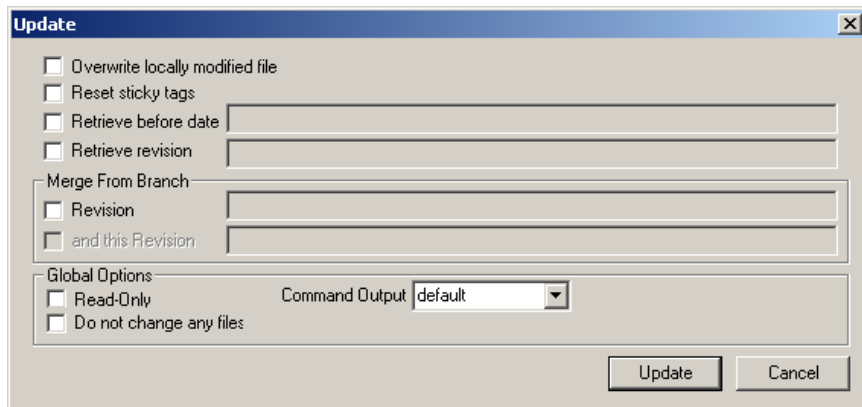


Table 3.2 Update dialog box (for files)—options

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 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 3.2 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	<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

Select a file in the project window, then select **VCS > Commit** to open the Commit dialog box. [Figure 3.4](#) shows the Commit dialog box for a file. [Table 3.3 on page 31](#) 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 15](#) for more information.

Figure 3.4 Commit dialog box (for files)

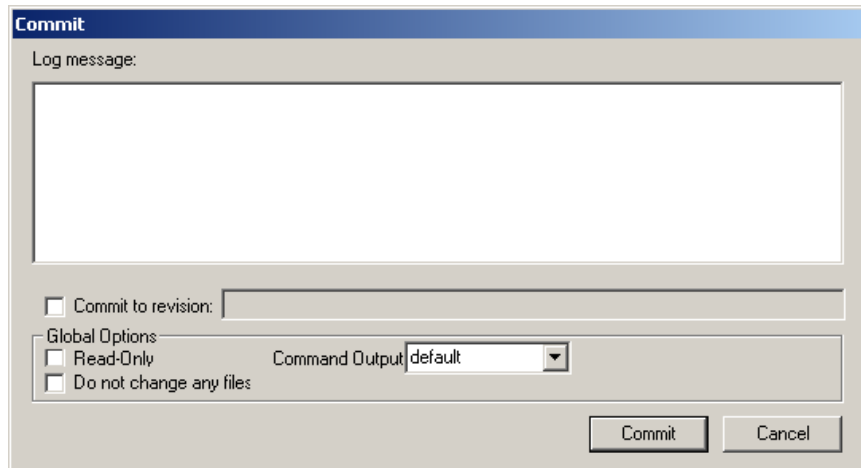


Table 3.3 Commit dialog box (for files)—options

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>
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

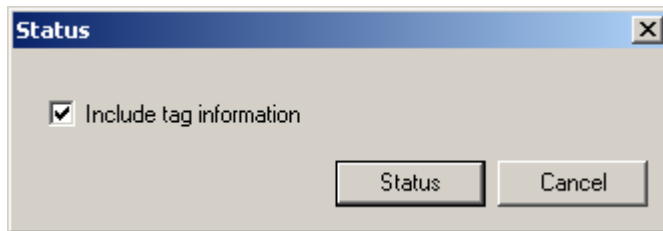
Select a file in the project window, then select **VCS > Status** to open the Status dialog box. [Figure 3.5](#) shows the Status dialog box for a file.

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)

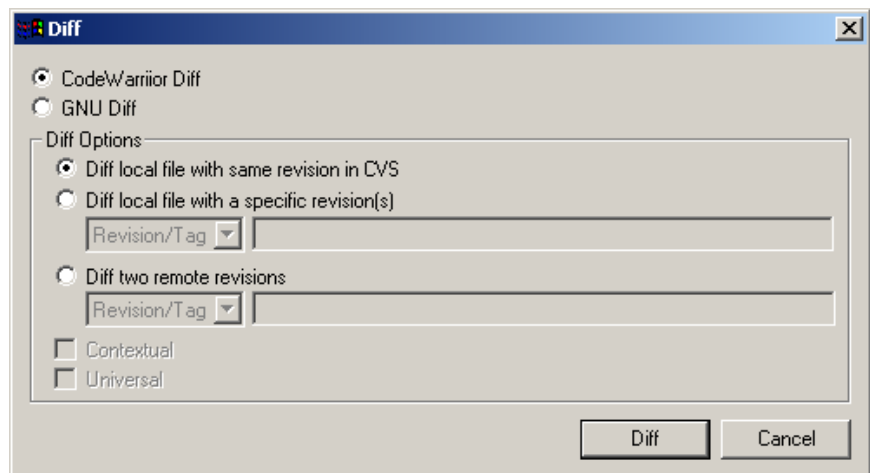


Table 3.4 Diff dialog box (for files)—options

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 NumberOrTag</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 Date</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:</p> <pre>[[[-r NumberOrTag] [-D Date]] { [[[-r NumberOrTag] [-D Date]] }</pre> <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 15](#) for more information.

Figure 3.7 Tag dialog box (for files)

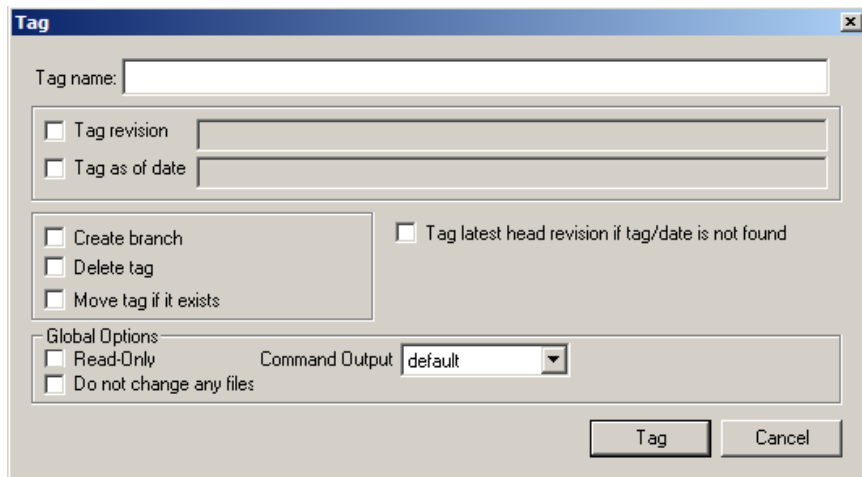


Table 3.5 Tag dialog box (for files)—options

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 <code>NumberOrTag</code> 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
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>
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

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](#) shows the Add dialog box for a file. [Table 3.6](#) 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)

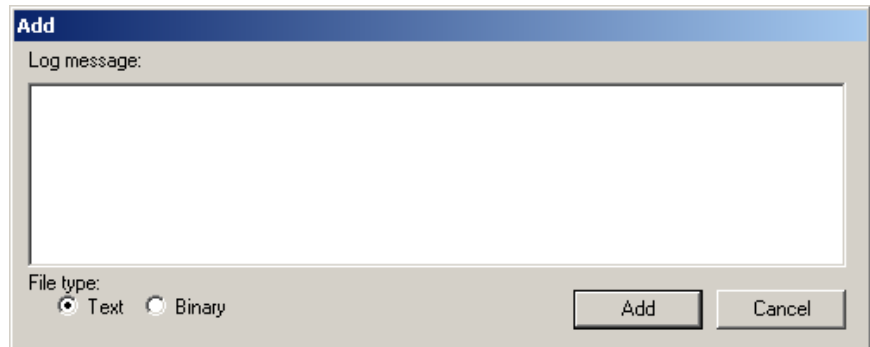


Table 3.6 Add dialog box (for files)—options

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 Text</code> , where <code>Text</code> 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)

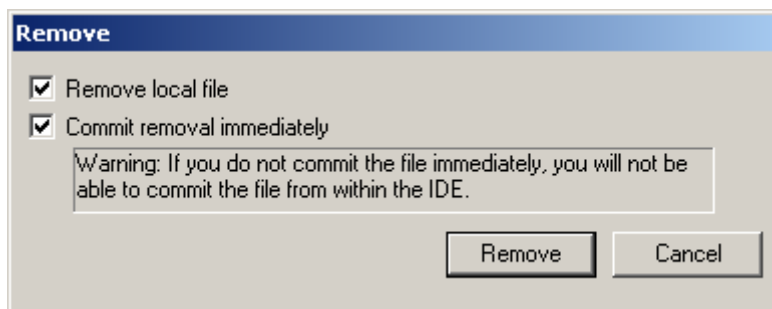


Table 3.7 Remove dialog box (for files)—options

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)

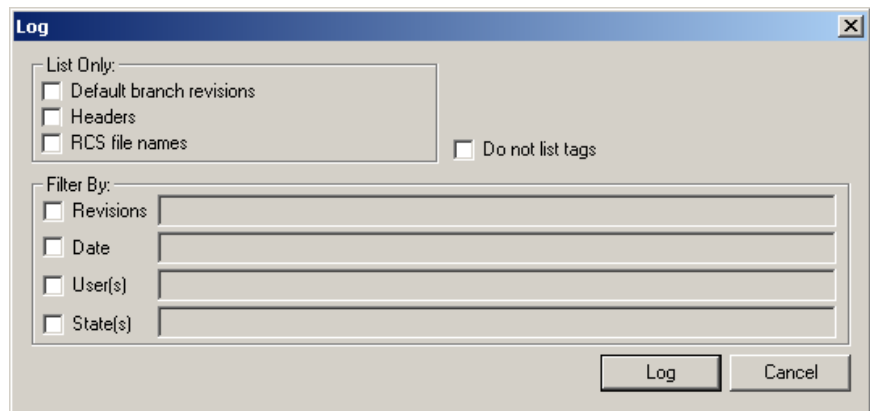


Table 3.8 Log dialog box (for files)—options

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

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

Option	Explanation
Do not list tags	Select to exclude tag information from the log. Clear to include the tag information. Selecting this option is equivalent to using the command-line switch <code>-N</code> .
Revisions	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. Selecting this option is equivalent to using the command-line switch <code>-r NumberOrTag:NumberOrTag</code> , where <code>NumberOrTag</code> is the revision number or tag to use for this operation.
Date	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. Selecting this option is equivalent to using the command-line switch <code>-d Date</code> , where <code>Date</code> is the date to use for this operation.
User(s)	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. Selecting this option is equivalent to using the command-line switch <code>-w LoginName1, LoginName2,...</code> , where <code>LoginName</code> is the login name to use for this operation.
State(s)	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. Selecting this option is equivalent to using the command-line switch <code>-s</code> .

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

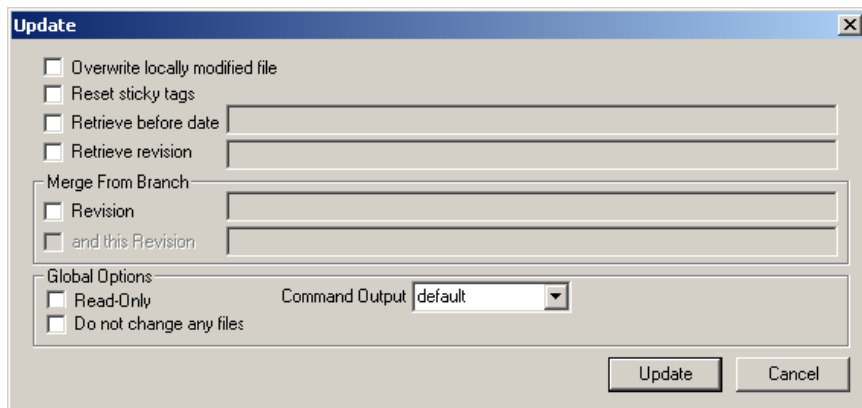
This section explains the mwCVS commands for projects. Most of the project commands appear identical to the file commands, but project commands apply to the current project while the file commands apply to a selected file or group of files. The [Difference](#) command does not apply to projects.

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 45](#) 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 15](#) 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](#) shows the Commit dialog box for a project. [Table 3.11 on page 49](#) 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 15](#) for more information.

Figure 3.13 Commit dialog box (for projects)

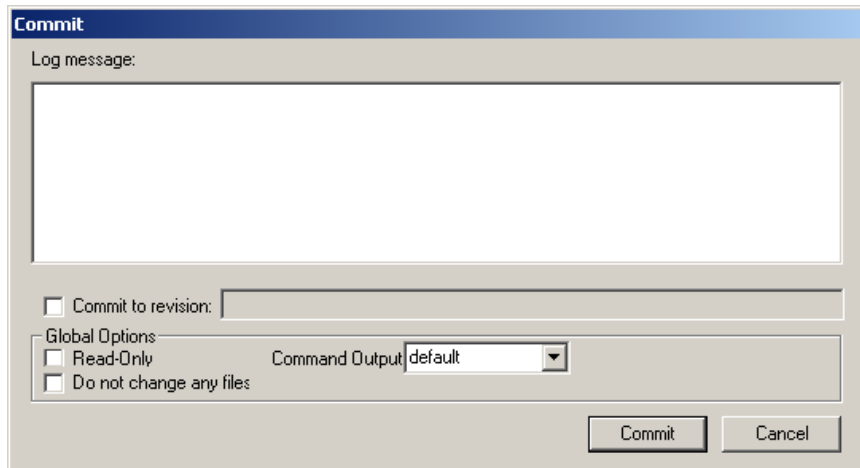


Table 3.11 Commit dialog box (for projects)—options

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>
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

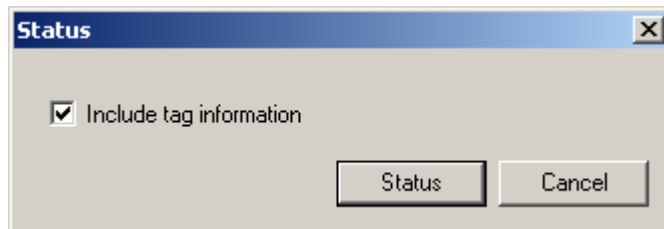
Bring forward a project window, then select **VCS > Project > Status** to open the Status dialog box.

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](#) 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 15](#) for more information.

Figure 3.15 Tag dialog box (for projects)

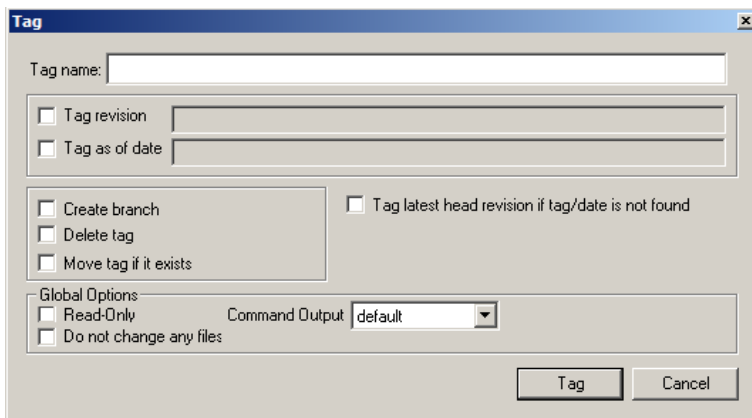


Table 3.12 Tag dialog box (for projects)—options

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 NumberOrTag</code>, where <i>NumberOrTag</i> is the revision number or tag to use for this operation.</p>

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

Option	Explanation
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 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 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>
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>

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

Option	Explanation
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>
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](#) shows the Add dialog box for a project. [Table 3.13](#) 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)

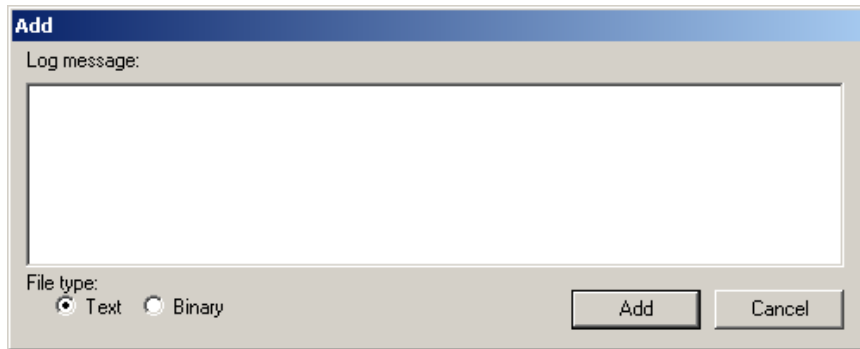


Table 3.13 Add dialog box (for projects)—options

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 Text</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](#) shows the Remove dialog box for a project. [Table 3.14](#) explains the dialog-box options.

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

Figure 3.17 Remove dialog box (for projects)

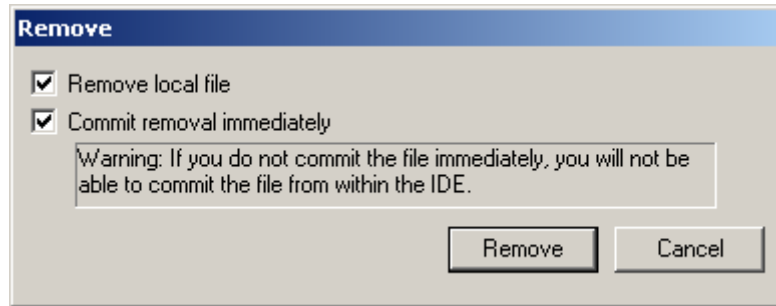


Table 3.14 Remove dialog box (for projects)—options

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)

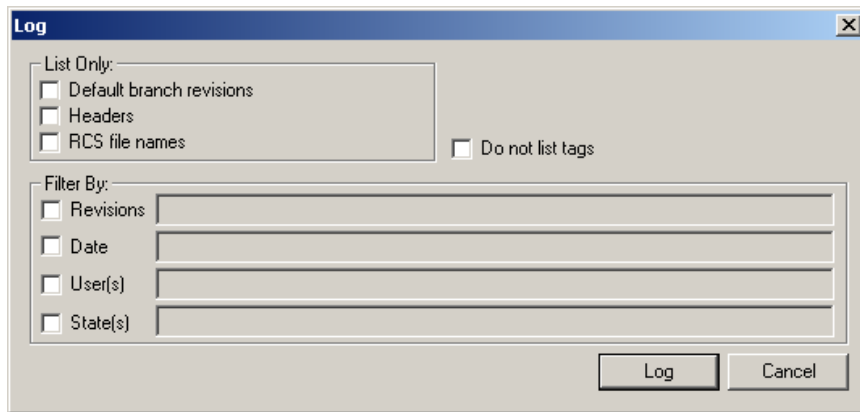


Table 3.15 Log dialog box (for projects)—options

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> .
RCS file names	Select to list project names that comply with Revision Control System (RCS) naming conventions. Clear to exclude RCS names from the log. Selecting this option is equivalent to using the command-line switch <code>-R</code> .

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

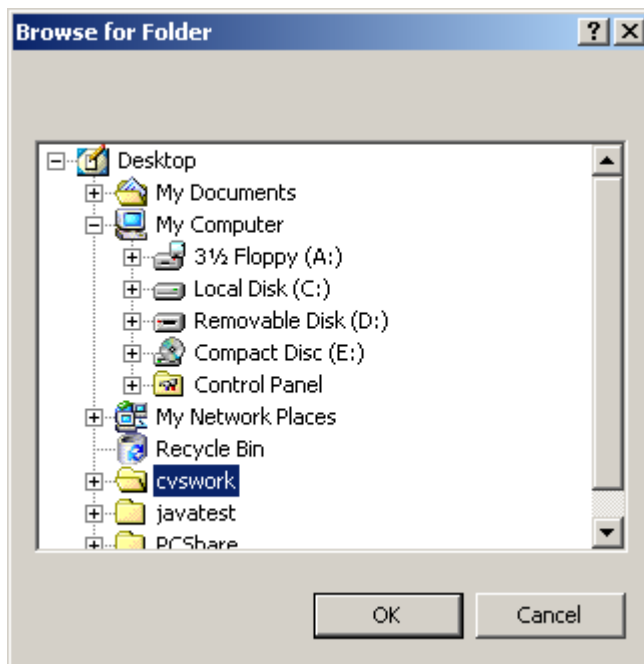
Option	Explanation
Do not list tags	<p>Select to exclude tag information from the log. Clear to include the tag information.</p> <p>Selecting this option is equivalent to using the command-line switch <code>-N</code>.</p>
Revisions	<p>Select to enter in the corresponding field the project revisions for 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 <i>NumberOrTag</i>:<i>NumberOrTag</i></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 <i>Date</i></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 <i>LoginName1</i>, <i>LoginName2</i>,...</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 15](#) for more information.

Figure 3.20 Update dialog box (recursive)

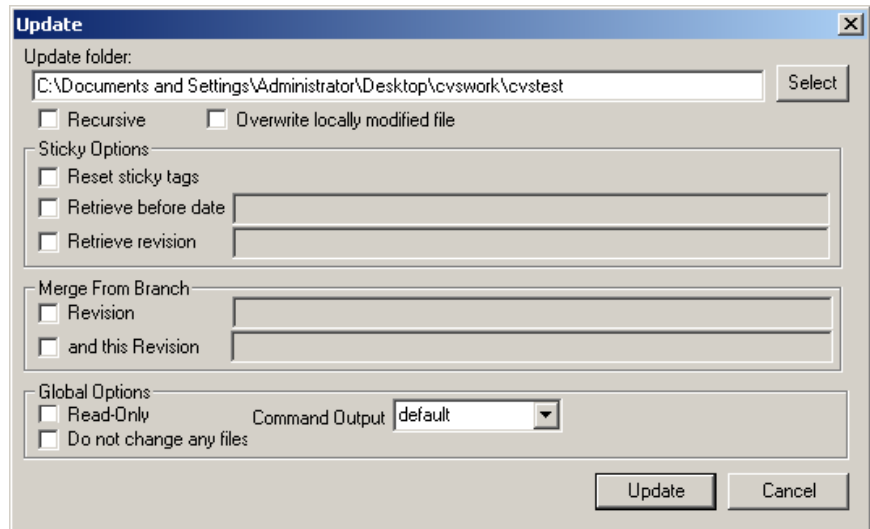


Table 3.16 Update dialog box (recursive)—options

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.
Overwrite locally modified file	Select to replace a locally modified file with the version stored on the CVS repository. Clear to preserve the locally modified file. Selecting this option is equivalent to using the GNU CVS command-line switch <code>-C</code> .
Reset sticky tags	Select to remove any sticky tags associated with the selected file after the IDE completes the update operation. Clear to preserve existing sticky tags. Selecting this option is equivalent to using the command-line switch <code>-A</code> .
Retrieve before date	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. Enter the date according to GNU CVS conventions. Sample dates are: <ul style="list-style-type: none">• 1 month ago• last Monday• 3/31/02 10:00:04 PST 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.
Retrieve revision	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. 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 3.16 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</code> <code>-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](#) shows the Commit dialog box for a directory. [Table 3.17](#) 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 15](#) for more information.

Figure 3.21 Commit dialog box (recursive)

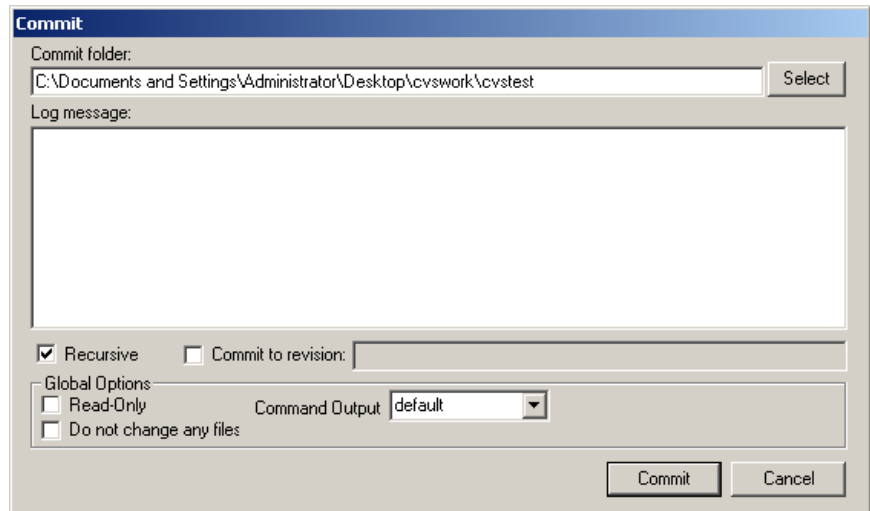


Table 3.17 Commit dialog box (recursive)—options

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.
Read-Only	Select to make the file read-only on the disk after issuing the Commit 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> . Clearing this option is equivalent to using the command-line switch <code>-w</code> .

Table 3.17 Commit dialog box (recursive)—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>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](#) shows the Checkout dialog box for a directory. [Table 3.18](#) 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 Output](#) options. Use the **mwCVS** panel in the **VCS Settings** window to configure the default settings. See [“mwCVS Panel Options” on page 15](#) for more information.

Figure 3.22 Checkout dialog box

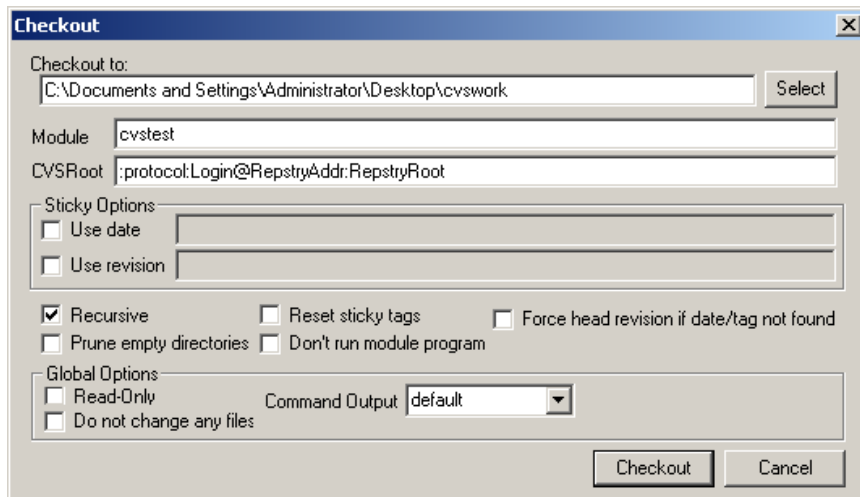


Table 3.18 Checkout dialog box—options

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 <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>
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 <code>-D Date</code>, 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 <code>-r NumberOrTag</code>, 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 <code>-l</code>.</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 <code>-P</code>.</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	<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 5Output	<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

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](#) explains the dialog-box options.

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

Figure 3.23 Status dialog box (recursive)

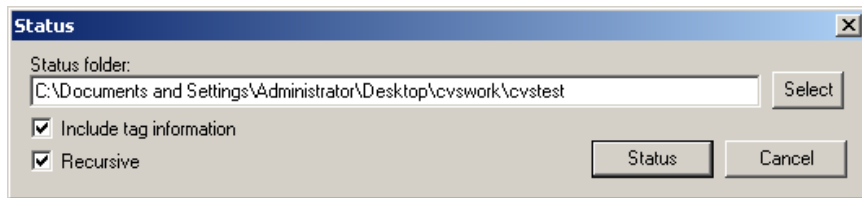


Table 3.19 Status dialog box (recursive)—options

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	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. Selecting this option is equivalent to using the GNU CVS command-line switch <code>-v</code> .
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](#) shows the Tag dialog box for a directory. [Table 3.20](#) 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 15](#) for more information.

Figure 3.24 Tag dialog box (recursive)

The screenshot shows the 'Tag' dialog box with the following fields and options:

- Tag folder:** A text field containing 'C:\Documents and Settings\Administrator\Desktop\cvswork\cvstest' and a 'Select' button.
- Tag name:** An empty text field.
- Tag revision:** A checkbox and an empty text field.
- Tag as of date:** A checkbox and an empty text field.
- Create branch:** A checkbox.
- Delete tag:** A checkbox.
- Move tag if it exists:** A checkbox.
- Recursive:** A checked checkbox.
- Tag latest head revision if tag/date is not found:** An unchecked checkbox.
- Global Options:**
 - Read-Only:** An unchecked checkbox.
 - Do not change any files:** An unchecked checkbox.
 - Command Output:** A dropdown menu set to 'default'.
- Buttons:** 'Tag' and 'Cancel' at the bottom right.

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.

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

Option	Explanation
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 NumberOrTag</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 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>

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

Option	Explanation
Recursive	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.
Tag latest head revision if tag/date is not found	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. Selecting this option is equivalent to using the command-line switch <code>-f</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.

Log

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

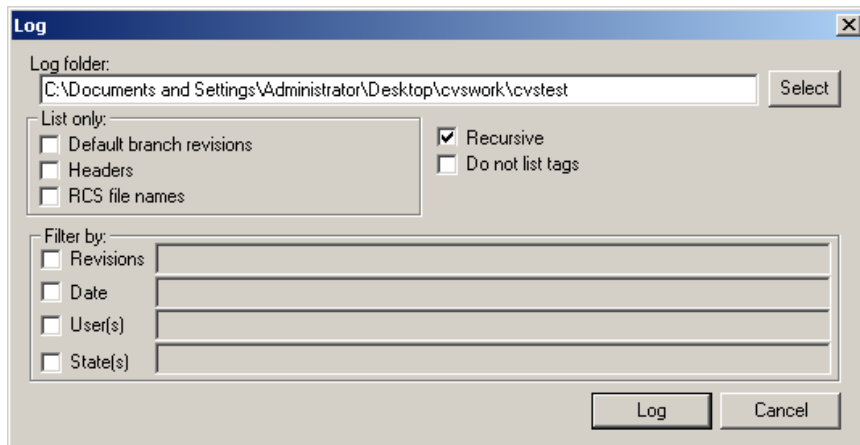


Table 3.21 Log dialog box (recursive)—options

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> .

Table 3.21 Log dialog box (recursive)—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>
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>

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

Option	Explanation
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 <i>LoginName1, LoginName2,...</i></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 `cvstest.mcp` and the associated files for that project.
- All files in the `cvstest.mcp` project are under version control.

This chapter has these topics:

- [“Part One: Turning on mwCVS” on page 78](#)
- [“Part Two: Accessing the Repository” on page 78](#)
- [“Part Three: Checking out a Module” on page 81](#)
- [“Part Four: Modifying a File” on page 84](#)
- [“Part Five: Committing the Modified File” on page 87](#)

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.

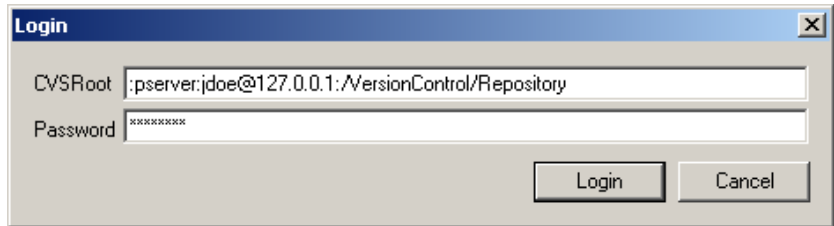
Windows

Follow the instructions in this section if you use mwCVS with the Windows-hosted IDE. See [“Macintosh” on page 80](#) if you use mwCVS with the Macintosh-hosted IDE.

1. Select **VCS > Login**.

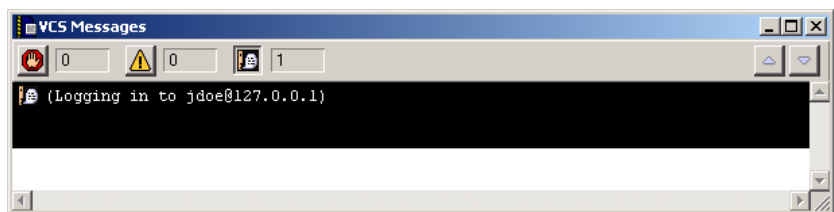
The Login window appears. Use this window, shown in [Figure 4.1 on page 79](#), to enter CVS root information and your password to access the repository.

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 78](#) 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](#) 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 on page 80](#)). 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`.

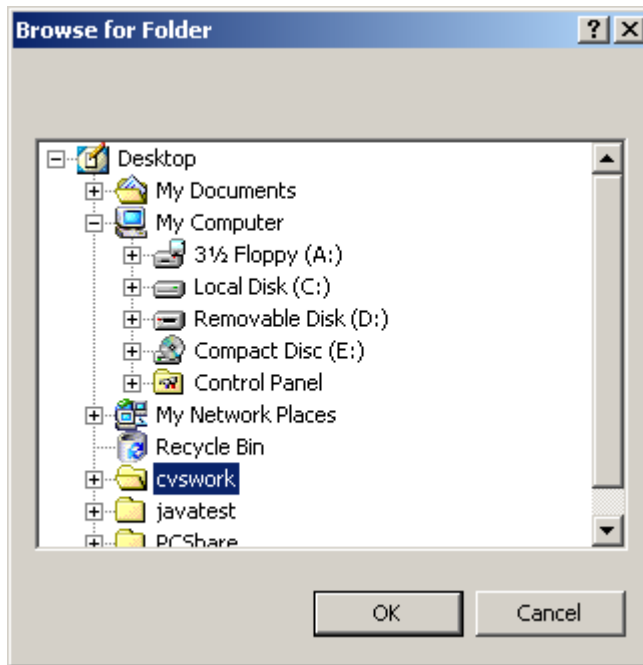
1. Select **VCS > Recursive > Checkout**.

A directory-selection dialog box like the one shown in [Figure 4.4 on page 82](#) appears.

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

[Figure 4.4 on page 82](#) shows `cvswork` as the selected directory. This directory becomes the working directory for the checkout command.

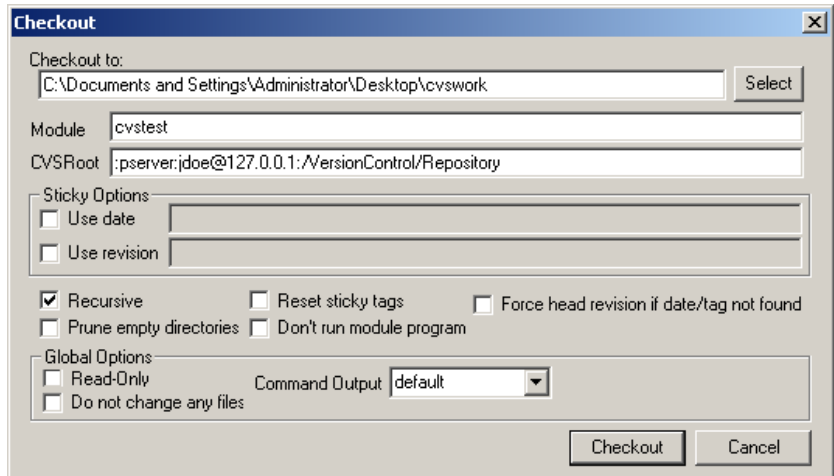
Figure 4.4 Directory-selection dialog box



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

Use the Checkout window, shown in [Figure 4.5 on page 83](#), to enter information about the repository and the modules that you want to check out. [Figure 4.5 on page 83](#) 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.

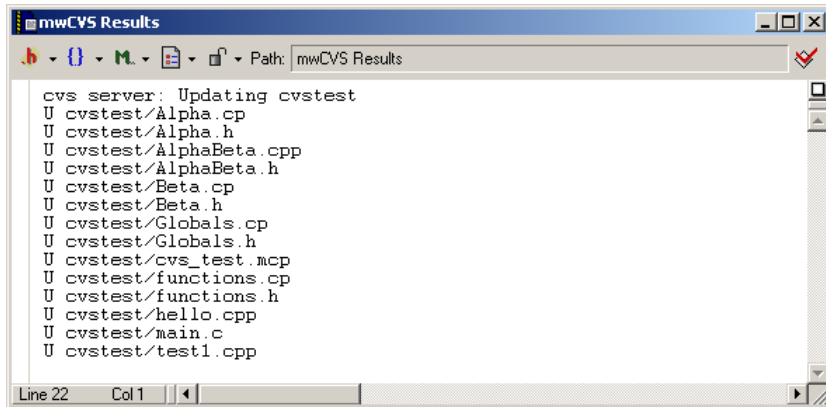
6. Click **Checkout**.

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 on page 84](#) 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.

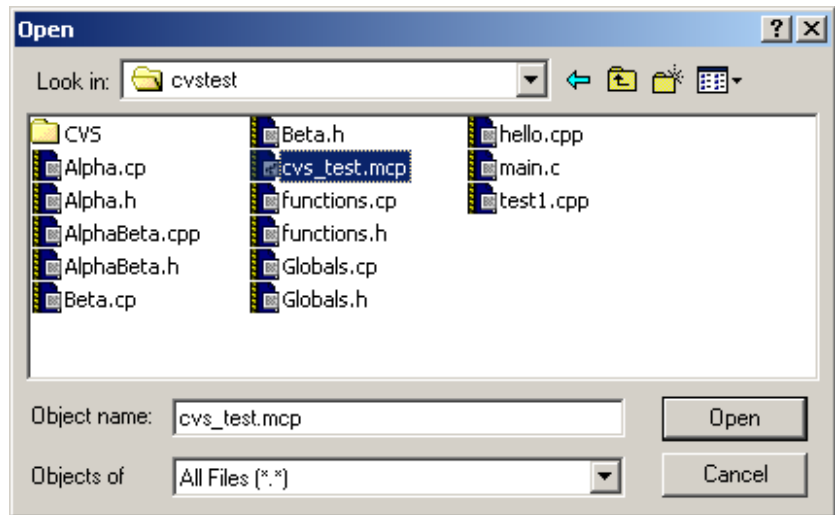
Part Four: Modifying a File

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 `cvs_test.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 `cvs_test.mcp` project inside the folder, as shown in [Figure 4.7 on page 85](#).

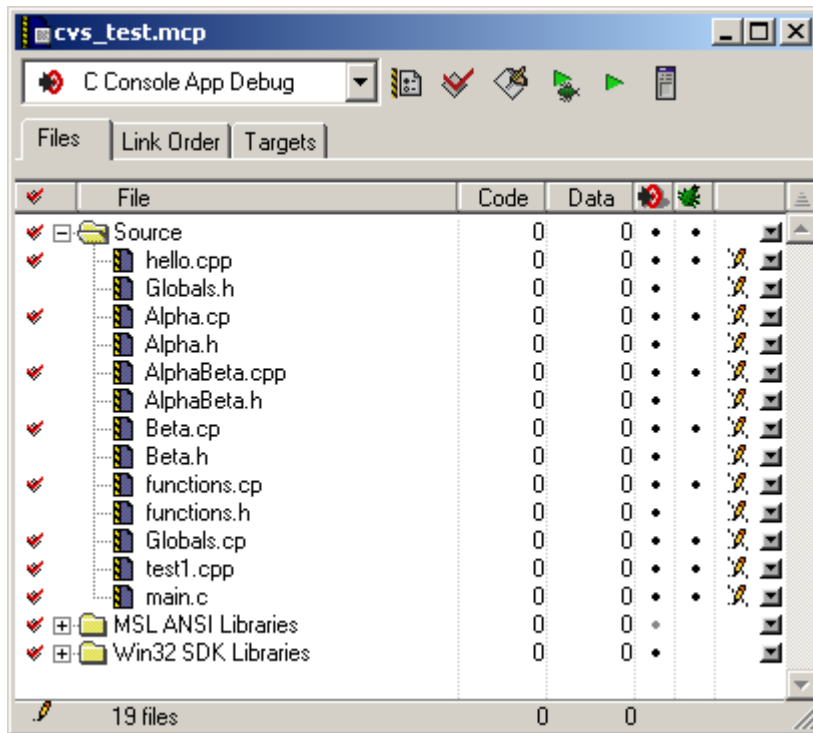
Figure 4.7 Opening the project in the checked-out module



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

[Figure 4.8 on page 86](#) shows the project window for `cvs_test.mcp`. Notice the VCS status icon  that appears next to the header file `Beta.h` in the Source group of the project window.

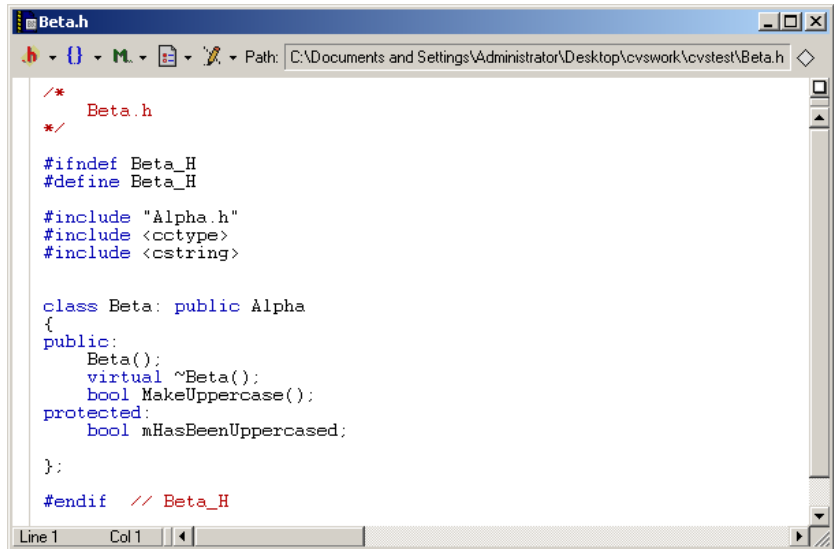
NOTE [Table 3.1 on page 22](#) explains the VCS status icons.

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 on page 87](#).

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 22](#) explains the VCS status icons.

Part Five: Committing the Modified File

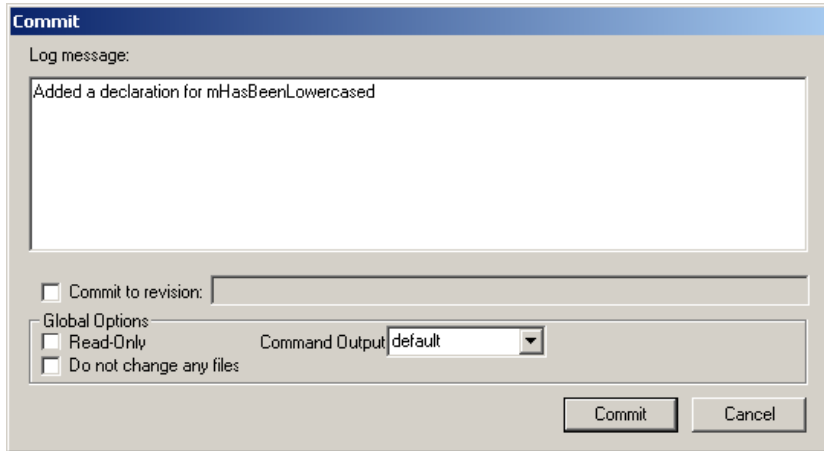
This part explains how to commit the changes that you made to a file. In this tutorial, you commit the `Beta.h` file that you modified earlier.

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



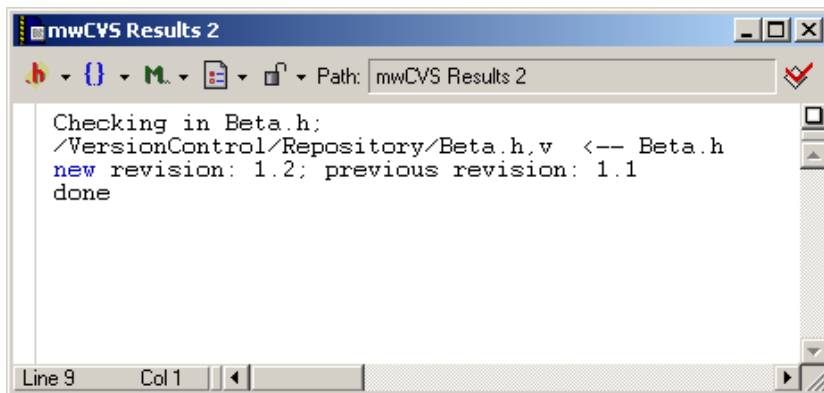
4. Click **Commit**.

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 27, 45, 60, 68
 - b
 - in Log dialog box 41, 56, 74
 - in Tag dialog box 37, 52, 72
 - C 27, 45, 60
 - c 35
 - D
 - in Checkout dialog box 67
 - in Tag dialog box 37, 52, 72
 - in Update dialog box 27, 45, 60
 - d
 - in Log dialog box 42, 57, 75
 - in Tag dialog box 37, 52, 72
 - F 37, 52, 72
 - f
 - in Checkout dialog box 68
 - in Remove dialog box 40, 55
 - in Tag dialog box 37, 52, 73
 - h 41, 56, 74
 - j 28, 46, 61
 - kb 39, 54
 - l 67, 70, 75
 - m 31, 39, 49, 54, 64
 - N 42, 57, 75
 - n
 - in Checkout dialog box 68, 69
 - in Commit dialog box 31, 49, 65
 - in Tag dialog box 38, 53, 73
 - in Update dialog box 29, 47, 62
 - P 67
 - Q
 - in Checkout dialog box 69
 - in Commit dialog box 31, 49, 65
 - in mwCVS Panel 15
 - in Tag dialog box 38, 53, 73
 - in Update dialog box 29, 47, 62
 - q
 - in Checkout dialog box 69
 - in Commit dialog box 31, 49, 65
 - in mwCVS Panel 15
 - in Tag dialog box 38, 53, 73
 - in Update dialog box 29, 47, 62
 - R 41, 56, 75
 - r 70
 - in Checkout dialog box 67, 68
 - in Commit dialog box 31, 49, 64
 - in Log dialog box 42, 57, 75
 - in mwCVS Panel 15
 - in Tag dialog box
 - Read-only 38, 53, 73
 - Tag revision 36, 51, 72
 - in Update dialog box
 - Read-only 28, 46, 61
 - Retrieve revision 27, 45, 60
 - s 42, 57, 76
 - u 35
 - v 32, 50, 70
 - w
 - in Checkout dialog box 68
 - in Commit dialog box 31, 49, 64
 - in Log dialog box 42, 57, 76
 - in Update dialog box 28, 46, 61
 - z 15
(Logging into remoteserver) 44
- ## A
- A 27, 45, 60, 68
 - about
 - CVS 6
 - mwCVS 5
 - accessing the repository, tutorial 78
 - Add dialog box
 - Binary 39, 54
 - Log message 39, 54
 - Text 39, 54
 - Add menu command 39, 54
 - and this Revision 28, 46, 61
- ## B
- b
 - in Log dialog box 41, 56, 74
 - in Tag dialog box 37, 52, 72

Binary 39, 54

button, Select

- in Checkout dialog box 66

- in Commit dialog box 64

- in Log dialog box 74

- in Status dialog box 70

- in Tag dialog box 71

- in Update dialog box 60

C

-C 27, 45, 60

-c 35

checking out a module, tutorial 81

Checkout dialog box

- Checkout to 66

- Command Output

 - Default 69

 - Quiet 69

 - Very Quiet 69

- CVSRoot 67

- Do not change any files 69

- Don't run module program 68

- Force head revision if date/tag not found 68

- Module 66

- Prune empty directories 67

- Read-Only 68

- Recursive 67

- Reset sticky tags 68

- Use date 67

- Use revision 67

Checkout menu command 66

Checkout to 66

CodeWarrior Diff 33

Command Output

- Default

 - in Checkout dialog box 69

 - in Commit dialog box 31, 49, 65

 - in mwCVS Panel 15

 - in Tag dialog box 38, 53, 73

 - in Update dialog box 29, 47, 62

- in Commit dialog box 31, 49, 65

- in mwCVS Panel 15

- in Tag dialog box 38, 53, 73

- in Update dialog box 29, 47, 62

Quiet

- in Checkout dialog box 69

- in Commit dialog box 31, 49, 65

- in mwCVS Panel 15

- in Tag dialog box 38, 53, 73

- in Update dialog box 29, 47, 62

Very Quiet

- in Checkout dialog box 69

- in Commit dialog box 31, 49, 65

- in mwCVS Panel 15

- in Tag dialog box 38, 53, 73

- in Update dialog box 29, 47, 62

commands

- Add 39, 54

- Checkout 66

- Commit 30, 48, 63

- context 23

- Difference 33

- for files 26

- for projects 44

- for the version control system 24

- Log 41, 56, 74

- Login 43

- recursive 58

- Remove 40, 55

- remove 5, 40, 55

- Status 32, 50, 70

- Synchronize Document 25

- Synchronize Selection 25

- Synchronize Status 24

- Tag 36, 51, 71

- Update 26, 44, 59

Commit dialog box

- Command Output 31, 49, 65

 - Default 31, 49, 65

 - Quiet 31, 49, 65

 - Very Quiet 31, 49, 65

- Commit folder 64

- Commit to revision 31, 49, 64

- Do not change any files 31, 49, 65

- Log message 31, 49, 64

- Read-Only 31, 49, 64

- Recursive 64

Commit folder 64

- Commit menu command 30, 48, 63
- Commit removal immediately 40, 55
- Commit to revision 31, 49, 64
- committing the modified file, tutorial 87
- Compatible with WinCVS 1.2
 - in mwCVS Panel 16
- concepts of CVS 6
- configuring mwCVS 13, 21
- context menus for CVS commands 23
- Contextual 35
- Create branch 37, 52, 72
- CVS
 - context menu commands 23
 - learning more about 6
 - status icons in project window 21
- CVSREAD 15
- CVSROOT 83
- CVSRoot 43, 67

D

- D
 - in Checkout dialog box 67
 - in Tag dialog box 37, 52, 72
 - in Update dialog box 27, 45, 60
- d
 - in Log dialog box 42, 57, 75
 - in Tag dialog box 37, 52, 72
- Date 42, 57, 75
- date formatting 27, 45, 60
- Daylight Saving Time issues, and resolution 16
- Default 15
- Default branch revisions 41, 56, 74
- Delete tag 37, 52, 72
- Diff dialog box
 - CodeWarrior Diff 33
 - Contextual 35
 - Diff local file with a specific revision(s) 34
 - Date 34
 - Revision/Tag 34
 - Diff local file with same revision in CVS 34
 - Diff two remote revisions 35
 - Date 34
 - Revision/Tag 34
 - GNU Diff 33

- Universal 35
- Diff local file with a specific revision(s) 34
 - Date 34
 - Revision/Tag 34
- Diff local file with same revision in CVS 34
- Diff two remote revisions 35
 - Date 34
 - Revision/Tag 34
- Difference menu command 33
- disabling mwCVS in the global scope 11
- disabling mwCVS in the project scope 12
- Do not change any files
 - in Checkout dialog box 69
 - in Commit dialog box 31, 49, 65
 - in Tag dialog box 38, 53, 73
 - in Update dialog box 29, 47, 62
- Do not list tags 42, 57, 75
- Don't run module program 68
- DST issues, and resolution 16

E

- enabling mwCVS in the global scope 9
- enabling mwCVS in the project scope 10

F

- F 37, 52, 72
- f
 - in Checkout dialog box 68
 - in Remove dialog box 40, 55
 - in Tag dialog box 37, 52, 73
- file commands
 - Add 39
 - Commit 30
 - Difference 33
 - Log 41
 - Login 43
 - Remove 40
 - Status 32
 - Synchronize Document 25
 - Synchronize Selection 25
 - Synchronize Status 24
 - Tag 36
 - Update 26
- file options

- and this Revision 28
- Binary 39
- CodeWarrior Diff 33
- Command Output 29, 31, 38
 - Default 29, 31, 38
 - Quiet 29, 31, 38
 - Very Quiet 29, 31, 38
- Commit removal immediately 40
- Commit to revision 31
- Contextual 35
- Create branch 37
- CVSRoot 43
- Date 42
- Default branch revisions 41
- Delete tag 37
- Diff local file with a specific revision(s) 34
 - Date 34
 - Revision/Tag 34
- Diff local file with same revision in CVS 34
- Diff two remote revisions 35
 - Date 34
 - Revision/Tag 34
- Do not change any files 29, 31, 38
- Do not list tags 42
- GNU Diff 33
- Headers 41
- Include tag information 32
- Log message 31, 39
- Move tag if it exists 37
- Overwrite locally modified file 27
- Password 43
- RCS file names 41
- Read-Only 28, 31, 38
- Remove local file 40
- Reset sticky tags 27
- Retrieve before date 27
- Retrieve revision 27
- Revision 28
- Revisions 42
- State(s) 42
- Tag as of date 37
- Tag latest head revision if tag/date is not found 37
- Tag name 36

- Tag revision 36
- Text 39
- Universal 35
- User(s) 42
- files, commands for 26
- Force head revision if date/tag not found 68
- formatting of dates 27, 45, 60

G

- general CVS information 6
- GNU Diff 33

H

- h 41, 56, 74
- Headers 41, 56, 74
- how to
 - disable mwCVS in the global scope 11
 - disable mwCVS in the project scope 12
 - enable mwCVS in the global scope 9
 - enable mwCVS in the project scope 10
 - install mwCVS on a Macintosh host 8
 - install mwCVS on a Windows host 7
 - integrate mwCVS with ssh on a Linux host 16
 - integrate mwCVS with ssh on a Windows host 17
 - turn off mwCVS in the global scope 11
 - turn off mwCVS in the project scope 12
 - turn on mwCVS in the global scope 9
 - turn on mwCVS in the project scope 10

I

- icon for VCS menu on a Macintosh host 10, 78
- Include tag information 32, 50, 70
- information about CVS 6
- installation of mwCVS 7
- installing mwCVS
 - on a Macintosh host 8
 - on a Windows host 7
- integrating mwCVS with ssh on a Linux host 16
- integrating mwCVS with ssh on a Windows host 17

J

-j 28, 46, 61

K

-kb 39, 54

L

-l 67, 70, 75

learning about CVS 6

Log dialog box

 Date 42, 57, 75

 Default branch revisions 41, 56, 74

 Do not list tags 42, 57, 75

 Headers 41, 56, 74

 Log folder 74

 RCS file names 41, 56, 75

 Recursive 75

 Revisions 42, 57, 75

 State(s) 42, 57, 76

 User(s) 42, 57, 76

Log folder 74

Log menu command 41, 56, 74

Log message 31, 39, 49, 54, 64

(Logging into remoteserver) 44

Login dialog box

 CVSRoot 43

 Password 43

Login menu command 43

M

-m 31, 39, 49, 54, 64

Macintosh VCS menu icon 10, 78

Make new working files read-only 15

menu commands

 Add 39, 54

 Checkout 66

 Commit 30, 48, 63

 Difference 33

 Log 41, 56, 74

 Login 43

 Remove 40, 55

 Status 32, 50, 70

 Synchronize Document 25

 Synchronize Selection 25

 Synchronize Status 24

 Tag 36, 51, 71

 Update 26, 44, 59

menu, context 23

modifying a file, tutorial 84

Module 66

more information about CVS 6

Move tag if it exists 37, 52, 72

mwCVS

 about 5

 configuring 13, 21

 disabling in the global scope 11

 disabling in the project scope 12

 enabling in the global scope 9

 enabling in the project scope 10

 file commands 26

 installation 7

 installing on a Macintosh host 8

 installing on a Windows host 7

 integrating with ssh 16

 options

 Command Output 15

 Compatible with WinCVS 1.2 16

 Make new working files read-only 15

 Use gZip level 9 compression 15

overview 5

project commands 44

recursive commands 58

system requirements 6

turning off in the global scope 11

turning off in the project scope 12

turning on in the global scope 9

turning on in the project scope 10

tutorial 77

 accessing the repository 78

 checking out a module 81

 committing the modified file 87

 modifying a file 84

 turning on mwCVS 78

version control system commands 24

mwCVS remove 5, 40, 55

mwCVS Results window 66, 83, 88

N

-N 42, 57, 75

-n

- in Checkout dialog box 68, 69
- in Commit dialog box 31, 49, 65
- in Tag dialog box 38, 53, 73
- in Update dialog box 29, 47, 62

O

overview of mwCVS 5

Overwrite locally modified file 27, 45, 60

P

-P 67

Password 43

password server 78

project commands

- Add 54
- Commit 48
- Log 56
- Remove 55
- Status 50
- Tag 51
- Update 44

project options

- and this Revision 46
- Binary 54
- Command Output 47, 49, 53
 - Default 47, 49, 53
 - Quiet 47, 49, 53
 - Very Quiet 47, 49, 53
- Commit removal immediately 55
- Commit to revision 49
- Create branch 52
- Date 57
- Default branch revisions 56
- Delete tag 52
- Do not change any files 47, 49, 53
- Do not list tags 57
- Headers 56
- Include tag information 50
- Log message 49, 54
- Move tag if it exists 52

Overwrite locally modified file 45

RCS file names 56

Read-Only 46, 49, 53

Remove local file 55

Reset sticky tags 45

Retrieve before date 45

Retrieve revision 45

Revision 46

Revisions 57

State(s) 57

Tag as of date 52

Tag latest head revision if tag/date is not
found 52

Tag name 51

Tag revision 51

Text 54

User(s) 57

project window, CVS status icons in 21

projects, commands for 44

Prune empty directories 67

pserver 78

Q

-Q

- in Checkout dialog box 69
- in Commit dialog box 31, 49, 65
- in mwCVS Panel 15
- in Tag dialog box 38, 53, 73
- in Update dialog box 29, 47, 62

-q

- in Checkout dialog box 69
- in Commit dialog box 31, 49, 65
- in mwCVS Panel 15
- in Tag dialog box 38, 53, 73
- in Update dialog box 29, 47, 62

Quiet 15

R

-R 41, 56, 75

-r 70

- in Checkout dialog box 67, 68
- in Commit dialog box 31, 49, 64
- in Log dialog box 42, 57, 75
- in mwCVS Panel 15

-
- in Tag dialog box
 - Read-only 38, 53, 73
 - Tag revision 36, 51, 72
 - in Update dialog box
 - Read-only 28, 46, 61
 - Retrieve revision 27, 45, 60
 - RCS 41, 56, 75
 - RCS file names 41, 56, 75
 - Read-Only
 - in Checkout dialog box 68
 - in Commit dialog box 31, 49, 64
 - in Tag dialog box 38, 53, 73
 - in Update dialog box 28, 46, 61
 - Recursive
 - in Checkout dialog box 67
 - in Commit dialog box 64
 - in Log dialog box 75
 - in Status dialog box 70
 - in Tag dialog box 73
 - in Update dialog box 60
 - recursive commands 58
 - Checkout 66
 - Commit 63
 - Log 74
 - Status 70
 - Tag 71
 - Update 59
 - recursive options
 - and this Revision 61
 - Checkout to 66
 - Command Output 62, 65, 73
 - Default 62, 65, 69, 73
 - Quiet 62, 65, 69, 73
 - Very Quiet 62, 65, 69, 73
 - Commit folder 64
 - Commit to revision 64
 - Create branch 72
 - CVSRoot 67
 - Date 75
 - Default branch revisions 74
 - Delete tag 72
 - Do not change any files 62, 65, 69, 73
 - Do not list tags 75
 - Don't run module program 68
 - Force head revision if date/tag not found 68
 - Headers 74
 - Include tag information 70
 - Log folder 74
 - Log message 64
 - Module 66
 - Move tag if it exists 72
 - Overwrite locally modified file 60
 - Prune empty directories 67
 - RCS file names 75
 - Read-Only 61, 64, 68, 73
 - Recursive
 - in Checkout dialog box 67
 - in Commit dialog box 64
 - in Log dialog box 75
 - in Status dialog box 70
 - in Tag dialog box 73
 - in Update dialog box 60
 - Reset sticky tags 60, 68
 - Retrieve before date 60
 - Retrieve revision 60
 - Revision 61
 - Revisions 75
 - State(s) 76
 - Status folder 70
 - Tag as of date 72
 - Tag folder 71
 - Tag latest head revision if tag/date is not found 73
 - Tag name 71
 - Tag revision 72
 - Update folder 60
 - Use date 67
 - Use revision 67
 - User(s) 76
 - remove command, differences from GNU CVS implementation 5, 40, 55
 - Remove dialog box
 - Commit removal immediately 40, 55
 - Remove local file 40, 55
 - Remove local file 40, 55
 - Remove menu command 40, 55
 - requirements for using mwCVS 6
 - Reset sticky tags 27, 45, 60, 68
-

resources for CVS 6

Retrieve before date 27, 45, 60

Retrieve revision 27, 45, 60

Revision 28, 46, 61

Revisions 42, 57, 75

S

-s 42, 57, 76

sample date formatting 27, 45, 60

Select button

in Checkout dialog box 66

in Commit dialog box 64

in Log dialog box 74

in Status dialog box 70

in Tag dialog box 71

in Update dialog box 60

settings

Command Output 15

Compatible with WinCVS 1.2 16

Make new working files read-only 15

Use gZip level 9 compression 15

ssh

integrating with mWCVS 16

State(s) 42, 57, 76

Status dialog box

Include tag information 32, 50, 70

Recursive 70

Status folder 70

Status folder 70

status icons in project window 21

Status menu command 32, 50, 70

Synchronize Document menu command 25

Synchronize Selection menu command 25

Synchronize Status menu command 24

system requirements 6

T

Tag as of date 37, 52, 72

Tag dialog box

Command Output 38, 53, 73

Default 38, 53, 73

Quiet 38, 53, 73

Very Quiet 38, 53, 73

Create branch 37, 52, 72

Delete tag 37, 52, 72

Do not change any files 38, 53, 73

Move tag if it exists 37, 52, 72

Read-Only 38, 53, 73

Recursive 73

Tag as of date 37, 52, 72

Tag folder 71

Tag latest head revision if tag/date is not
found 37, 52, 73

Tag name 36, 51, 71

Tag revision 36, 51, 72

Tag folder 71

Tag latest head revision if tag/date is not
found 37, 52, 73

Tag menu command 36, 51, 71

Tag name 36, 51, 71

Tag revision 36, 51, 72

tasks

disabling mWCVS in the global scope 11

disabling mWCVS in the project scope 12

enabling mWCVS in the global scope 9

enabling mWCVS in the project scope 10

installing mWCVS on a Macintosh host 8

installing mWCVS on a Windows host 7

integrating mWCVS with ssh on a Linux
host 16

integrating mWCVS with ssh on a Windows
host 17

turning off mWCVS in the global scope 11

turning off mWCVS in the project scope 12

turning on mWCVS in the global scope 9

turning on mWCVS in the project scope 10

Text 39, 54

turning off mWCVS in the global scope 11

turning off mWCVS in the project scope 12

turning on mWCVS in the global scope 9

turning on mWCVS in the project scope 10

turning on mWCVS, tutorial 78

tutorial 77

accessing the repository 78

Macintosh 80

Windows 78

checking out a module 81

committing the modified file 87

modifying a file 84
turning on mwCVS 78

U

-u 35
Universal 35
Update dialog box
 and this Revision 28, 46, 61
 Command Output 29, 47, 62
 Default 29, 47, 62
 Quiet 29, 47, 62
 Very Quiet 29, 47, 62
 Do not change any files 29, 47, 62
 Overwrite locally modified file 27, 45, 60
 Read-Only 28, 46, 61
 Recursive 60
 Reset sticky tags 27, 45, 60
 Retrieve before date 27, 45, 60
 Retrieve revision 27, 45, 60
 Revision 28, 46, 61
 Update folder 60
Update folder 60
Update menu command 26, 44, 59
Use date 67
Use global settings as default 9, 12
Use gZip level 9 compression 15
Use revision 67
User(s) 42, 57, 76
using ssh with mwCVS 16

V

-v 32, 50, 70
valid date formatting 27, 45, 60
VCS menu on a Macintosh host 10, 78
VCS Messages window 44, 79
version control system, commands for 24
Very Quiet 15

W

-w
 in Checkout dialog box 68
 in Commit dialog box 31, 49, 64
 in Log dialog box 42, 57, 76

 in Update dialog box 28, 46, 61
WinCVS version 1.2, resolving Daylight Savings
 Time issues in 16
windows
 mwCVS Results
 for Checkout 66, 83
 for Commit 88
VCS Messages 79

Z

-z 15

