# Table of Contents

## 1 mwCVS Overview and Setup
- 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
- 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
- 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://cvs gui.sourceforge.net/
mwCVS Overview and Setup

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>
<thead>
<tr>
<th>Host</th>
<th>mwCVS Files</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows</td>
<td>• mwCVS.dll</td>
</tr>
<tr>
<td></td>
<td>• pMWCVS.dll</td>
</tr>
<tr>
<td></td>
<td>• cvs.exe</td>
</tr>
<tr>
<td></td>
<td>• cwspawn.exe</td>
</tr>
<tr>
<td>Macintosh</td>
<td>• mwCVS</td>
</tr>
<tr>
<td></td>
<td>• mwCVS Panel</td>
</tr>
<tr>
<td></td>
<td>• CVSAAdapter.bundle</td>
</tr>
</tbody>
</table>

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\CodeWarriorDirPath\Bin\Plugins\version control\%

2. Place the file mwCVS.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)\n
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

<table>
<thead>
<tr>
<th>Scope</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global</td>
<td>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).</td>
</tr>
<tr>
<td>Project</td>
<td>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.</td>
</tr>
</tbody>
</table>

### 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

<table>
<thead>
<tr>
<th>Scope</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global</td>
<td>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.</td>
</tr>
<tr>
<td>Project</td>
<td>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. 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.</td>
</tr>
</tbody>
</table>

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:

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

<table>
<thead>
<tr>
<th>Option</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Make new working files read-only (-r)</td>
<td>Select to prevent changes to new working files that you check out from the CVS repository.</td>
</tr>
<tr>
<td>Use gZip level 9 compression (-z)</td>
<td>Select to use gZip level 9 (high compression) for files transmitted between the IDE and the CVS repository.</td>
</tr>
<tr>
<td>Command Output</td>
<td>Select the level of information that you want mwCVS to output for CVS operations.</td>
</tr>
<tr>
<td>Compatible with WinCVS 1.2</td>
<td>Select to prevent Daylight Saving Time issues from affecting CVS operations with WinCVS 1.2.</td>
</tr>
</tbody>
</table>
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`.
Configuring mwCVS

Integrating mwCVS with ssh

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 cvsowner/.ssh/authorized_keys or cvsowner/.ssh/authorized_keys2 on the CVS repository host, where cvsowner is the name of the user account that owns CVSROOT.
Configuring mwCVS

Integrating mwCVS with ssh

NOTE  Never share your private key (id_dsa).

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

   ssh -i path/to/key -l cvs/owner server/addr

   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, server/addr is the address of the CVS repository host, and path/to/cvsroot is the path to the CVSROOT of the CVS repository host:

   CVS_RSH="ssh"
   CVSROOT=":ext:user@server/addr:/path/to/cvsroot"
   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://sshwindows.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.
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 `cvsowner/.ssh/authorized_keys` or `cvsowner/.ssh/authorized_keys2` on the CVS repository host, where `cvsowner` 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, `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 /home/loginname/\.ssh/id_dsa -l cvsowner 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.
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.
Using mwCVS
Project Window

Table 3.1  CVS status icons

<table>
<thead>
<tr>
<th>Icon</th>
<th>File is in the CVS Repository</th>
<th>File is not in the CVS Repository</th>
<th>File isWritable</th>
<th>File is Read Only</th>
<th>File has changed since the last Update</th>
<th>File has not changed since the last Update</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Icon" /></td>
<td><img src="image2" alt="Icon" /></td>
<td><img src="image3" alt="Icon" /></td>
<td><img src="image4" alt="Icon" /></td>
<td><img src="image5" alt="Icon" /></td>
<td><img src="image6" alt="Icon" /></td>
<td><img src="image7" alt="Icon" /></td>
</tr>
<tr>
<td><img src="image8" alt="Icon" /></td>
<td><img src="image9" alt="Icon" /></td>
<td><img src="image10" alt="Icon" /></td>
<td><img src="image11" alt="Icon" /></td>
<td><img src="image12" alt="Icon" /></td>
<td><img src="image13" alt="Icon" /></td>
<td><img src="image14" alt="Icon" /></td>
</tr>
<tr>
<td><img src="image15" alt="Icon" /></td>
<td><img src="image16" alt="Icon" /></td>
<td><img src="image17" alt="Icon" /></td>
<td><img src="image18" alt="Icon" /></td>
<td><img src="image19" alt="Icon" /></td>
<td><img src="image20" alt="Icon" /></td>
<td><img src="image21" alt="Icon" /></td>
</tr>
<tr>
<td><img src="image22" alt="Icon" /></td>
<td><img src="image23" alt="Icon" /></td>
<td><img src="image24" alt="Icon" /></td>
<td><img src="image25" alt="Icon" /></td>
<td><img src="image26" alt="Icon" /></td>
<td><img src="image27" alt="Icon" /></td>
<td><img src="image28" alt="Icon" /></td>
</tr>
<tr>
<td><img src="image29" alt="Icon" /></td>
<td><img src="image30" alt="Icon" /></td>
<td><img src="image31" alt="Icon" /></td>
<td><img src="image32" alt="Icon" /></td>
<td><img src="image33" alt="Icon" /></td>
<td><img src="image34" alt="Icon" /></td>
<td><img src="image35" alt="Icon" /></td>
</tr>
</tbody>
</table>
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**.
Using mwCVS
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

<table>
<thead>
<tr>
<th>Option</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overwrite locally modified file</td>
<td>Select to replace a locally modified file with the version stored on the CVS repository. Clear to preserve the locally modified file.</td>
</tr>
<tr>
<td></td>
<td>Selecting this option is equivalent to using the GNU CVS command-line switch <code>-C</code>.</td>
</tr>
<tr>
<td>Reset sticky tags</td>
<td>Select to remove any sticky tags associated with the selected file after the IDE completes the update operation. Clear to preserve existing sticky tags.</td>
</tr>
<tr>
<td></td>
<td>Selecting this option is equivalent to using the command-line switch <code>-A</code>.</td>
</tr>
<tr>
<td>Retrieve before date</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>Enter the date according to GNU CVS conventions.</td>
</tr>
<tr>
<td></td>
<td>Sample dates are:</td>
</tr>
<tr>
<td></td>
<td>• 1 month ago</td>
</tr>
<tr>
<td></td>
<td>• last Monday</td>
</tr>
<tr>
<td></td>
<td>• 3/31/02 10:00:04 PST</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>Retrieve revision</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>Selecting this option is equivalent to using the command-line switch <code>-r NumberOrTag</code>, where <code>NumberOrTag</code> is the revision number or tag to use for this operation.</td>
</tr>
</tbody>
</table>
## Table 3.2 Update dialog box (for files)—options (continued)

<table>
<thead>
<tr>
<th>Option</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Merge From Branch</td>
<td>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. Selecting this option is equivalent to using the command-line switch <code>–j NumberOrTag</code>.</td>
</tr>
<tr>
<td>and this Revision</td>
<td>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. 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. Selecting this option is equivalent to using the command-line switch <code>–j NumberOrTag –j NumberOrTag</code>.</td>
</tr>
<tr>
<td>Read-Only</td>
<td>Select to make the file read-only on the disk after issuing the Update 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>.</td>
</tr>
</tbody>
</table>


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

<table>
<thead>
<tr>
<th>Option</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Do not change any files</td>
<td>Select to see the result of the update operation without modifying the CVS repository. Clear to modify the CVS repository to reflect the update operation. Selecting this option is equivalent to using the command-line switch (-n).</td>
</tr>
<tr>
<td>Command Output</td>
<td>Select the level of information that you want mwCVS to output during CVS operations:&lt;br&gt;  * Default—output informational messages  &lt;br&gt;  * Quiet—output non-informational messages ((-q)}&lt;br&gt;  * Very Quiet—output only messages for serious errors ((-Q)}&lt;br&gt; This option supersedes the setting in the mwCVS panel of the VCS Settings window.</td>
</tr>
</tbody>
</table>
Using mwCVS
File Commands

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)
### Using mwCVS

#### File Commands

#### Table 3.3 Commit dialog box (for files)—options

<table>
<thead>
<tr>
<th>Option</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Log message</td>
<td>Enter a log message for 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 <code>Text</code> is the log message to use for this operation.</td>
</tr>
<tr>
<td>Commit to revision</td>
<td>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 <code>NumberOrTag</code> is the revision number or tag to use for this operation.</td>
</tr>
<tr>
<td>Read-Only</td>
<td>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>.</td>
</tr>
<tr>
<td>Do not change any files</td>
<td>Select to see the result of the commit operation without modifying the CVS repository. Clear to modify the CVS repository to reflect the commit operation. Selecting this option is equivalent to using the command-line switch <code>-n</code>.</td>
</tr>
</tbody>
</table>
| Command Output        | Select the level of information that you want mwCVS to output during CVS operations:  
  * **Default**—output informational messages  
  * **Quiet**—output non-informational messages (-q)  
  * **Very Quiet**—output only messages for serious errors (-Q)  
This option supersedes the setting in the mwCVS panel of the VCS Settings window. |
Status

Select a file in the project window, then select VCS > Status to open the Status dialog box. Figure 3.5 shows the Status dialog box for a file.

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

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

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

Figure 3.5 Status dialog box (for files)
Difference

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

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

Figure 3.6 Diff dialog box (for files)

Table 3.4 Diff dialog box (for files)—options

<table>
<thead>
<tr>
<th>Option</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>CodeWarrior Diff</td>
<td>Select to display difference results in a File Compare Results window. This</td>
</tr>
<tr>
<td></td>
<td>window graphically depicts difference information. If you select this option,</td>
</tr>
<tr>
<td>GNU Diff</td>
<td>select the GNU Diff option.</td>
</tr>
<tr>
<td>Diff Options</td>
<td></td>
</tr>
<tr>
<td></td>
<td>loginUser With same revision in CVS</td>
</tr>
<tr>
<td></td>
<td>loginUser With a specific revision(s)</td>
</tr>
<tr>
<td></td>
<td>loginUser With Two remote revisions</td>
</tr>
<tr>
<td>Contextual</td>
<td></td>
</tr>
<tr>
<td>Universal</td>
<td></td>
</tr>
</tbody>
</table>

Version Control Plug-in 1.3 for CVS
## Using mwCVS

### File Commands

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

<table>
<thead>
<tr>
<th>Option</th>
<th>Explanation</th>
</tr>
</thead>
</table>
| Diff local file with same revision in CVS | Select to display difference results between the local working file and the corresponding revision of that file in the CVS repository.  
If you select this option, you cannot select the Diff local file with a specific revision(s) or Diff two remote revisions options. |
| Diff local file with a specific revision(s) | 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:  
- **Revision/Tag**—Enter a revision number or tag to use for the difference operation. This option is equivalent to using the command-line switch `-r NumberOrTag`, where `NumberOrTag` 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 `-D Date`, where `Date` is the date to use for this operation.  
The IDE displays difference results between the local working file and the specified revisions of that file in the CVS repository.  
If you select this option, you cannot select the Diff local file with same revision in CVS or Diff two remote revisions options. |
Using mwCVS
File Commands

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

<table>
<thead>
<tr>
<th>Option</th>
<th>Explanation</th>
</tr>
</thead>
</table>
| Diff two remote revisions | Select to enter information to use for the difference operation. In this case, you select two revisions of the file in the CVS repository.  
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.  
The second pair of fields corresponds to the part of this command-line expression enclosed in braces:  
\[
([-r \text{ NumberOrTag}] | [-D \text{ Date}]) \\
\{([-r \text{ NumberOrTag}] | [-D \text{ Date}]) \}
\]  
The IDE displays difference results between the specified revisions of the file in the CVS repository.  
If you select this option, you cannot select the **Diff local file with same revision in CVS**. |
| Contextual | Select to generate a contextual-format difference for use with the UNIX `patch` command. Clear to generate a non-contextual format.  
Selecting this option is equivalent to using the command-line switch `-c`.  
Select the **GNU Diff** option to turn on the **Contextual** option. |
| Universal | Select to generate a universal-format difference. Clear to generate a non-universal format.  
Selecting this option is equivalent to using the command-line switch `-u`.  
Select the **GNU Diff** option to turn on the **Universal** option. |
Using mwCVS
File Commands

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

<table>
<thead>
<tr>
<th>Option</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tag name</td>
<td>Enter the name of the tag.</td>
</tr>
<tr>
<td>Tag revision</td>
<td>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 (-r) <code>NumberOrTag</code>, where <code>NumberOrTag</code> is the revision number or tag to use for this operation.</td>
</tr>
<tr>
<td>Option</td>
<td>Explanation</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Tag as of date</td>
<td>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. 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.</td>
</tr>
<tr>
<td>Create branch</td>
<td>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. Selecting this option is equivalent to using the command-line switch <code>-b</code>. If you select this option, you cannot select the <strong>Delete tag</strong> option.</td>
</tr>
<tr>
<td>Delete tag</td>
<td>Select to permanently delete the specified tag from the CVS repository. Clear to add the tag to the specified file revision. Selecting this option is equivalent to using the command-line switch <code>-d</code>. If you select this option, you cannot select the <strong>Create branch</strong> or <strong>Move tag if it exists</strong> options.</td>
</tr>
<tr>
<td>Move tag if it exists</td>
<td>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. Selecting this option is equivalent to using the command-line switch <code>-f</code>. If you select this option, you cannot select the <strong>Delete tag</strong> option.</td>
</tr>
<tr>
<td>Tag latest head revision if tag/date is not found</td>
<td>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>.</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Option</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read-Only</td>
<td>Select to make the file read-only on the disk after issuing the Tag command. Clear to make the file read/write.</td>
</tr>
<tr>
<td></td>
<td>This option supersedes the setting in the mwCVS panel of the VCS Settings window.</td>
</tr>
<tr>
<td></td>
<td>Selecting this option is equivalent to using the command-line switch <code>-r</code>.</td>
</tr>
<tr>
<td>Do not change any files</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>Selecting this option is equivalent to using the command-line switch <code>-n</code>.</td>
</tr>
<tr>
<td>Command Output</td>
<td>Select the level of information that you want mwCVS to output during CVS operations:</td>
</tr>
<tr>
<td></td>
<td>• Default—output informational messages</td>
</tr>
<tr>
<td></td>
<td>• Quiet—output non-informational messages (<code>-q</code>)</td>
</tr>
<tr>
<td></td>
<td>• Very Quiet—output only messages for serious errors (<code>-Q</code>)</td>
</tr>
<tr>
<td></td>
<td>This option supersedes the setting in the mwCVS panel of the VCS Settings window.</td>
</tr>
</tbody>
</table>
**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>
<thead>
<tr>
<th>File type:</th>
<th>Text</th>
<th>Binary</th>
</tr>
</thead>
</table>

Table 3.6 Add dialog box (for files)—options

<table>
<thead>
<tr>
<th>Option</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Log message</td>
<td>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.</td>
</tr>
<tr>
<td>Text</td>
<td>Select this file type if you are adding a text file to the CVS repository.</td>
</tr>
<tr>
<td>Binary</td>
<td>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>.</td>
</tr>
</tbody>
</table>
Using mwCVS
File Commands

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.

Figure 3.9 Remove dialog box (for files)

Table 3.7 Remove dialog box (for files)—options

<table>
<thead>
<tr>
<th>Option</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remove local file</td>
<td>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 -f.</td>
</tr>
<tr>
<td>Commit removal immediately</td>
<td>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.</td>
</tr>
</tbody>
</table>
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)**

![Log dialog box](image)

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

<table>
<thead>
<tr>
<th>Option</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default branch revisions</td>
<td>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>.</td>
</tr>
<tr>
<td>Headers</td>
<td>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>.</td>
</tr>
<tr>
<td>RCS file names</td>
<td>Select to list file names that comply with Revision Control System (RCS) naming conventions. Clear to exclude RCS file names from the log. Selecting this option is equivalent to using the command-line switch <code>-R</code>.</td>
</tr>
</tbody>
</table>
## Table 3.8 Log dialog box (for files)—options (continued)

<table>
<thead>
<tr>
<th>Option</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Do not list tags</td>
<td>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>.</td>
</tr>
<tr>
<td>Revisions</td>
<td>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.</td>
</tr>
<tr>
<td>Date</td>
<td>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.</td>
</tr>
<tr>
<td>User(s)</td>
<td>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.</td>
</tr>
<tr>
<td>State(s)</td>
<td>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>.</td>
</tr>
</tbody>
</table>
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.

![Login dialog box](image)

Table 3.9 Login dialog box—options

<table>
<thead>
<tr>
<th>Option</th>
<th>Explanation</th>
</tr>
</thead>
</table>
| CVSRoot   | Enter in this field the root of the remote CVS repository. Use the general construction 
:protocol:Login@RepstryAddr:RepstryRoot 
where 
- protocol is the protocol you use to access the repository (such as local or pserver). 
- Login@RepstryAddr is your login name (Login), followed by @, followed by the Internet Protocol (IP) or Domain Name System (DNS) address of the remote repository (RepstryAddr) 
- RepstryRoot is the path to the root level of the remote repository 
The default entry that appears in this field corresponds to the last used CVSRoot entry. |
| 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

<table>
<thead>
<tr>
<th>Option</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overwrite locally modified file</td>
<td>Select to replace a locally modified project with the version stored on the CVS repository. Clear to preserve the locally modified project. Selecting this option is equivalent to using the GNU CVS command-line switch <code>-C</code>.</td>
</tr>
<tr>
<td>Reset sticky tags</td>
<td>Select to remove any sticky tags associated with the current project 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>.</td>
</tr>
</tbody>
</table>
| Retrieve before date          | 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. Enter the date according to GNU CVS conventions. Sample dates are:  
  - 1 month ago  
  - last Monday  
  - 3/31/02 10:00:04 PST  
  Selecting this option is equivalent to using the command-line switch `-D Date`, where `Date` is the date to use for this operation. |
| Retrieve revision             | 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. Selecting this option is equivalent to using the command-line switch `-r NumberOrTag`, where `NumberOrTag` is the revision number or tag to use for this operation. |
Using mwCVS
Project Commands

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

<table>
<thead>
<tr>
<th>Option</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Merge From Branch</td>
<td>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. Selecting this option is equivalent to using the command-line switch <code>-j NumberOrTag</code>.</td>
</tr>
<tr>
<td>and this Revision</td>
<td>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. 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. Selecting this option is equivalent to using the command-line switch <code>-j NumberOrTag -j NumberOrTag</code>.</td>
</tr>
<tr>
<td>Read-Only</td>
<td>Select to make the file read-only on the disk after issuing the Update 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>-c</code>. Clearing this option is equivalent to using the command-line switch <code>-w</code>.</td>
</tr>
<tr>
<td>Option</td>
<td>Explanation</td>
</tr>
<tr>
<td>------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Do not change any files</td>
<td>Select to see the result of the update operation without modifying the CVS repository. Clear to modify the CVS repository to reflect the update operation. Selecting this option is equivalent to using the command-line switch -n.</td>
</tr>
<tr>
<td>Command Output</td>
<td>Select the level of information that you want mwCVS to output during CVS operations:</td>
</tr>
<tr>
<td></td>
<td>• <strong>Default</strong>—output informational messages</td>
</tr>
<tr>
<td></td>
<td>• <strong>Quiet</strong>—output non-informational messages (-q)</td>
</tr>
<tr>
<td></td>
<td>• <strong>Very Quiet</strong>—output only messages for serious errors (-Q)</td>
</tr>
<tr>
<td></td>
<td>This option supersedes the setting in the mwCVS panel of the VCS Settings window.</td>
</tr>
</tbody>
</table>
Using mwCVS
Project Commands

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

<table>
<thead>
<tr>
<th>Option</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Log message</td>
<td>Enter a message to log with the project 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 <code>Text</code> is the log message to use for this operation.</td>
</tr>
<tr>
<td>Commit to revision</td>
<td>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. Selecting this option is equivalent to using the command-line switch <code>\(-r \) NumberOrTag</code>, where <code>NumberOrTag</code> is the revision number or tag to use for this operation.</td>
</tr>
<tr>
<td>Read-Only</td>
<td>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>.</td>
</tr>
<tr>
<td>Do not change any files</td>
<td>Select to see the result of the commit operation without modifying the CVS repository. Clear to modify the CVS repository to reflect the commit operation. Selecting this option is equivalent to using the command-line switch <code>\(-n \)</code>.</td>
</tr>
</tbody>
</table>
| Command Output          | Select the level of information that you want mwCVS to output during CVS operations:  
  * **Default**—output informational messages  
  * **Quiet**—output non-informational messages (`\(-q \)`  
  * **Very Quiet**—output only messages for serious errors (`\(-Q \)`  
  This option supersedes the setting in the mwCVS panel of the VCS Settings window. |
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

<table>
<thead>
<tr>
<th>Option</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tag name</td>
<td>Enter in this field the name of the tag.</td>
</tr>
<tr>
<td>Tag revision</td>
<td>Select to enter in the corresponding field the project revision that you want to tag. Clear to apply the tag to the latest revision.</td>
</tr>
<tr>
<td></td>
<td>Selecting this option is equivalent to using the GNU CVS command-line switch --r NumberOrTag, where NumberOrTag is the revision number or tag to use for this operation.</td>
</tr>
</tbody>
</table>
### Table 3.12 Tag dialog box (for projects)—options (continued)

<table>
<thead>
<tr>
<th>Option</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tag as of date</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>Selecting this option is equivalent to using the command-line switch (-D) \textit{Date}, where \textit{Date} is the date to use for this operation.</td>
</tr>
<tr>
<td>Create branch</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>Selecting this option is equivalent to using the command-line switch (-b).</td>
</tr>
<tr>
<td></td>
<td>If you select this option, you cannot select the \textbf{Delete tag} option.</td>
</tr>
<tr>
<td>Delete tag</td>
<td>Select to permanently delete the specified tag from the CVS repository. Clear to add the tag to the specified project revision.</td>
</tr>
<tr>
<td></td>
<td>Selecting this option is equivalent to using the command-line switch (-d).</td>
</tr>
<tr>
<td></td>
<td>If you select this option, you cannot select the \textbf{Create branch} or \textbf{Move tag if it exists} options.</td>
</tr>
<tr>
<td>Move tag if it exists</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>Selecting this option is equivalent to using the command-line switch (-F).</td>
</tr>
<tr>
<td></td>
<td>If you select this option, you cannot select the \textbf{Delete tag} option.</td>
</tr>
<tr>
<td>Tag latest head revision if tag/date is not found</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>Selecting this option is equivalent to using the command-line switch (-f).</td>
</tr>
</tbody>
</table>
### Table 3.12 Tag dialog box (for projects)—options (continued)

<table>
<thead>
<tr>
<th>Option</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Read-Only</strong></td>
<td>Select to make the file read-only on the disk after issuing the Tag command. Clear to make the file read/write.</td>
</tr>
<tr>
<td></td>
<td>This option supersedes the setting in the mwCVS panel of the VCS Settings window.</td>
</tr>
<tr>
<td></td>
<td>Selecting this option is equivalent to using the command-line switch <code>-r</code>.</td>
</tr>
<tr>
<td><strong>Do not change any files</strong></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>Selecting this option is equivalent to using the command-line switch <code>-n</code>.</td>
</tr>
<tr>
<td><strong>Command Output</strong></td>
<td>Select the level of information that you want mwCVS to output during CVS operations:</td>
</tr>
<tr>
<td></td>
<td>• <strong>Default</strong>—output informational messages</td>
</tr>
<tr>
<td></td>
<td>• <strong>Quiet</strong>—output non-informational messages (<code>-q</code>)</td>
</tr>
<tr>
<td></td>
<td>• <strong>Very Quiet</strong>—output only messages for serious errors (<code>-Q</code>)</td>
</tr>
<tr>
<td></td>
<td>This option supersedes the setting in the mwCVS panel of the VCS Settings window.</td>
</tr>
</tbody>
</table>
Using mwCVS
Project Commands

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)

![Add dialog box](image)

Table 3.13 Add dialog box (for projects)—options

<table>
<thead>
<tr>
<th>Option</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Log message</td>
<td>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 -m Text, where Text is the log message to use for this operation.</td>
</tr>
<tr>
<td>Text</td>
<td>Select this file type if you are adding a text-formatted project to the CVS repository.</td>
</tr>
<tr>
<td>Binary</td>
<td>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 -kb.</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Option</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remove local file</td>
<td>Select to delete the local project as part of the mwCVS remove operation. Clear to preserve the local project. Selecting this option is equivalent to using the GNU CVS command-line switch -f.</td>
</tr>
<tr>
<td>Commit removal immediately</td>
<td>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. 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.</td>
</tr>
</tbody>
</table>
Using mwCVS
Project Commands

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>
<thead>
<tr>
<th>Option</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default branch revisions</td>
<td>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 -b.</td>
</tr>
<tr>
<td>Headers</td>
<td>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 -h.</td>
</tr>
<tr>
<td>RCS file names</td>
<td>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 -R.</td>
</tr>
</tbody>
</table>
### Table 3.15 Log dialog box (for projects)—options (continued)

<table>
<thead>
<tr>
<th>Option</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Do not list tags</td>
<td>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 (-N).</td>
</tr>
<tr>
<td>Revisions</td>
<td>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. Selecting this option is equivalent to using the command-line switch (-r \text{NumberOrTag}:\text{NumberOrTag}), where NumberOrTag is the revision number or tag to use for this operation.</td>
</tr>
<tr>
<td>Date</td>
<td>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. Selecting this option is equivalent to using the command-line switch (-d \text{Date}), where Date is the date to use for this operation.</td>
</tr>
<tr>
<td>User(s)</td>
<td>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. Selecting this option is equivalent to using the command-line switch (-u \text{LoginName1, LoginName2,...}), where LoginName is the login name to use for this operation.</td>
</tr>
<tr>
<td>State(s)</td>
<td>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. Selecting this option is equivalent to using the command-line switch (-s).</td>
</tr>
</tbody>
</table>
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

![Directory-selection dialog box](image)
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

<table>
<thead>
<tr>
<th>Option</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Update folder</td>
<td>This field holds the path to the directory that you selected in the directory-selection dialog box. Click <strong>Select</strong> to choose a different directory. Alternatively, enter a full path directly into the field.</td>
</tr>
<tr>
<td>Recursive</td>
<td>Select to update both the directory itself and any directories nested within it. Clear to update only the directory itself.</td>
</tr>
<tr>
<td>Overwrite locally modified file</td>
<td>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 (-c).</td>
</tr>
<tr>
<td>Reset sticky tags</td>
<td>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 (-A).</td>
</tr>
</tbody>
</table>
| 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:  
  • 1 month ago  
  • last Monday  
  • 3/31/02 10:00:04 PST  
Selecting this option is equivalent to using the command-line switch \(-D Date\), where *Date* 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 \(-r NumberOrTag\), where *NumberOrTag* is the revision number or tag to use for this operation. |
Table 3.16  Update dialog box (recursive)—options (continued)

<table>
<thead>
<tr>
<th>Option</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Merge From Branch</td>
<td>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. Selecting this option is equivalent to using the command-line switch <code>-j NumberOrTag</code>.</td>
</tr>
<tr>
<td>and this Revision</td>
<td>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. 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. Selecting this option is equivalent to using the command-line switch <code>-j NumberOrTag</code> <code>-j NumberOrTag</code>.</td>
</tr>
<tr>
<td>Read-Only</td>
<td>Select to make the file read-only on the disk after issuing the Update 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>.</td>
</tr>
</tbody>
</table>
### Using mwCVS

*Recursive Commands*

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

<table>
<thead>
<tr>
<th>Option</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Do not change any files</td>
<td>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. 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>-n</code>.</td>
</tr>
</tbody>
</table>
| Command Output          | Select the level of information that you want mwCVS to output during CVS operations:  
  - **Default**—output informational messages  
  - **Quiet**—output non-informational messages (-q)  
  - **Very Quiet**—output only messages for serious errors (-Q)  
  This option supersedes the setting in the mwCVS panel of the VCS Settings window. |
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

<table>
<thead>
<tr>
<th>Option</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Commit folder</td>
<td>This field holds the path to the directory that you selected in the directory-selection dialog box. Click <strong>Select</strong> to choose a different directory. Alternatively, enter a full path directly into the field.</td>
</tr>
<tr>
<td>Log message</td>
<td>Enter in this field a message to log with the file that you will commit.</td>
</tr>
<tr>
<td></td>
<td>Entering a message in this field 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.</td>
</tr>
<tr>
<td>Recursive</td>
<td>Select to commit both the directory itself and any directories nested within it. Clear to commit only the directory itself.</td>
</tr>
<tr>
<td>Commit to revision</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>Selecting this option is equivalent to using the command-line switch <code>-r NumberOrTag</code>, where <code>NumberOrTag</code> is the revision number or tag to use for this operation.</td>
</tr>
<tr>
<td>Read-Only</td>
<td>Select to make the file read-only on the disk after issuing the Commit command. Clear to make the file read/write.</td>
</tr>
<tr>
<td></td>
<td>This option supersedes the setting in the mwCVS panel of the VCS Settings window.</td>
</tr>
<tr>
<td></td>
<td>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>.</td>
</tr>
</tbody>
</table>
### Table 3.17 Commit dialog box (recursive)—options (continued)

<table>
<thead>
<tr>
<th>Option</th>
<th>Explanation</th>
</tr>
</thead>
</table>
| Do not change any files | Select to see the result of the commit operation without modifying the CVS repository. Clear to modify the CVS repository to reflect the commit operation.  
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 `-n`. |
| Command Output          | Select the level of information that you want mwCVS to output during CVS operations:  
- **Default**—output informational messages  
- **Quiet**—output non-informational messages (`-q`)  
- **Very Quiet**—output only messages for serious errors (`-Q`)  
This option supersedes the setting in the mwCVS panel of the VCS Settings window. |
Using mwCVS
Recursive Commands

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 5Output 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](image)

![Table 3.18 Checkout dialog box—options](table)

<table>
<thead>
<tr>
<th>Option</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Checkout to</td>
<td>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.</td>
</tr>
<tr>
<td>Module</td>
<td>Enter in this field the path to the module that you want to check out from the CVS repository.</td>
</tr>
</tbody>
</table>
### Table 3.18 Checkout dialog box—options (continued)

<table>
<thead>
<tr>
<th>Option</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CVSRoot</strong></td>
<td>Enter in this field the root of the CVS repository. Use the general construction &lt;br&gt;protocol: Login@RepstryAddr: RepstryRoot &lt;br&gt;where &lt;br&gt;• protocol is the protocol you use to access the repository (such as local or pserver). &lt;br&gt;• Login@RepstryAddr is your login name (Login), followed by @, followed by the Internet Protocol (IP) or Domain Name System (DNS) address of the remote repository (RepstryAddr) &lt;br&gt;• RepstryRoot is the path to the root level of the remote repository &lt;br&gt;The default entry that appears in this field corresponds to the last used CVSRoot entry.</td>
</tr>
<tr>
<td><strong>Use date</strong></td>
<td>Select to check out files with a sticky date. Clear to disregard date when checking out files. &lt;br&gt;Selecting this option is equivalent to using the GNU CVS command-line switch -D Date, where Date is the date to use for this operation.</td>
</tr>
<tr>
<td><strong>Use revision</strong></td>
<td>Select to check out files with a sticky revision tag. Clear to disregard revision tag when checking out files. &lt;br&gt;Selecting this option is equivalent to using the command-line switch -r NumberOrTag, where NumberOrTag is the revision number or tag to use for this operation.</td>
</tr>
<tr>
<td><strong>Recursive</strong></td>
<td>Select to check out both the directory itself and any directories nested within it. Clear to check out only the directory itself. &lt;br&gt;Clearing this option is equivalent to using the command-line switch -l.</td>
</tr>
<tr>
<td><strong>Prune empty directories</strong></td>
<td>Select to avoid checking out empty directories from the CVS repository. Clear to check out all directories, regardless of empty state. &lt;br&gt;Selecting this option is equivalent to using the command-line switch -p.</td>
</tr>
</tbody>
</table>
Using mwCVS
Recursive Commands

Table 3.18 Checkout dialog box—options (continued)

<table>
<thead>
<tr>
<th>Option</th>
<th>Explanation</th>
</tr>
</thead>
</table>
| Reset sticky tags               | Select to remove any sticky tags associated with the selected file after the IDE completes the checkout operation. Clear to preserve existing sticky tags.  
                                | Selecting this option is equivalent to using the command-line switch \(-A\).                                                             |
| Don’t run module program        | Select to see the result of the commit operation without modifying the CVS repository. Clear to modify the CVS repository to reflect the commit operation.  
                                | Selecting this option is equivalent to using the command-line switch \(-n\).                                                            |
| Force head revision if date/ tag not found | 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.  
                                | Selecting this option is equivalent to using the command-line switch \(-f\).                                                             |
| Read-Only                       | Select to make the file read-only on the disk after issuing the Checkout 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 \(-r\). Clearing this option is equivalent to using the command-line switch \(-w\). |

Version Control Plug-in 1.3 for CVS
Using mwCVS
Recursive Commands

Table 3.18 Checkout dialog box—options (continued)

<table>
<thead>
<tr>
<th>Option</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Do not change any files</td>
<td>Select to see the result of the commit operation without modifying the CVS repository. Clear to modify the CVS repository to reflect the commit operation. Selecting this option is equivalent to using the command-line switch <code>-n</code>.</td>
</tr>
</tbody>
</table>
| Command 5Output               | Select the level of information that you want mwCVS to output during CVS operations:  
• **Default**—output informational messages  
• **Quiet**—output non-informational messages (-q)  
• **Very Quiet**—output only messages for serious errors (-Q)  
This option supersedes the setting in the mwCVS panel of the VCS Settings window. |
Using mwCVS
Recursive Commands

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

<table>
<thead>
<tr>
<th>Option</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status folder</td>
<td>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.</td>
</tr>
<tr>
<td>Include tag information</td>
<td>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 -v.</td>
</tr>
<tr>
<td>Recursive</td>
<td>Selecting this option is equivalent to using the command-line switch -r. Clearing this option is equivalent to using the command-line switch -l.</td>
</tr>
</tbody>
</table>
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)

Table 3.20 Tag dialog box (recursive)—options

<table>
<thead>
<tr>
<th>Option</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tag folder</td>
<td>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.</td>
</tr>
<tr>
<td>Tag name</td>
<td>Enter in this field the name of the tag.</td>
</tr>
</tbody>
</table>
### Table 3.20 Tag dialog box (recursive)—options (continued)

<table>
<thead>
<tr>
<th>Option</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tag revision</td>
<td>Select to enter in the corresponding field the file revision that you want to tag. Clear to apply the tag to the latest revision. Selection</td>
</tr>
<tr>
<td></td>
<td>ing this option is equivalent to using the GNU CVS command-line switch <code>-r NumberOrTag</code>, where NumberOrTag is the revision number or tag to use for this operation.</td>
</tr>
<tr>
<td>Tag as of date</td>
<td>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. Selection</td>
</tr>
<tr>
<td></td>
<td>this option is equivalent to using the command-line switch <code>-D Date</code>, where Date is the date to use for this operation.</td>
</tr>
<tr>
<td>Create branch</td>
<td>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. Selection</td>
</tr>
<tr>
<td></td>
<td>this option is equivalent to using the command-line switch <code>-b</code>.</td>
</tr>
<tr>
<td></td>
<td>If you select this option, you cannot select the <strong>Delete tag</strong> option.</td>
</tr>
<tr>
<td>Delete tag</td>
<td>Select to permanently delete the specified tag from the CVS repository. Clear to add the tag to the specified file revision. Selection</td>
</tr>
<tr>
<td></td>
<td>this option is equivalent to using the command-line switch <code>-d</code>.</td>
</tr>
<tr>
<td></td>
<td>If you select this option, you cannot select the <strong>Create branch</strong> or <strong>Move tag if it exists</strong> options.</td>
</tr>
<tr>
<td>Move tag if it exists</td>
<td>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. Selection</td>
</tr>
<tr>
<td></td>
<td>this option is equivalent to using the command-line switch <code>-F</code>.</td>
</tr>
<tr>
<td></td>
<td>If you select this option, you cannot select the <strong>Delete tag</strong> option.</td>
</tr>
</tbody>
</table>
### Table 3.20 Tag dialog box (recursive)—options (continued)

<table>
<thead>
<tr>
<th>Option</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recursive</td>
<td>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.</td>
</tr>
<tr>
<td>Tag latest head revision if tag/date is not found</td>
<td>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>.</td>
</tr>
<tr>
<td>Read-Only</td>
<td>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>.</td>
</tr>
<tr>
<td>Do not change any files</td>
<td>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>.</td>
</tr>
<tr>
<td>Command Output</td>
<td>Select the level of information that you want mwCVS to output during CVS operations:&lt;br&gt;  - Default—output informational messages&lt;br&gt;  - Quiet—output non-informational messages (<code>-q</code>)&lt;br&gt;  - Very Quiet—output only messages for serious errors (<code>-Q</code>)&lt;br&gt;This option supersedes the setting in the mwCVS panel of the VCS Settings window.</td>
</tr>
</tbody>
</table>
Using mwCVS
Recursive Commands

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

<table>
<thead>
<tr>
<th>Option</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Log folder</td>
<td>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.</td>
</tr>
<tr>
<td>Default branch revisions</td>
<td>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 -b.</td>
</tr>
<tr>
<td>Headers</td>
<td>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 -h.</td>
</tr>
</tbody>
</table>
### Using mwCVS

#### Recursive Commands

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

<table>
<thead>
<tr>
<th>Option</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RCS file names</td>
<td>Select to list file names that comply with Revision Control System (RCS) naming conventions. Clear to exclude RCS file names from the log.</td>
</tr>
<tr>
<td></td>
<td>Selecting this option is equivalent to using the command-line switch (-R).</td>
</tr>
<tr>
<td>Recursive</td>
<td>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. Clearing this box is equivalent to using the command-line switch (-l).</td>
</tr>
<tr>
<td>Do not list tags</td>
<td>Select to exclude tag information from the log. Clear to include the tag information.</td>
</tr>
<tr>
<td></td>
<td>Selecting this option is equivalent to using the command-line switch (-N).</td>
</tr>
<tr>
<td>Revisions</td>
<td>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 (-r\ $\ NumberOrTag).</td>
</tr>
<tr>
<td></td>
<td>$\ NumberOrTag; NumberOrTag$, where $\ NumberOrTag$ is the revision number or tag to use for this operation.</td>
</tr>
<tr>
<td>Date</td>
<td>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 (--d\ $\ Date) where $\ Date$ is the date to use for this operation.</td>
</tr>
</tbody>
</table>
### Using mwCVS

**Recursive Commands**

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

<table>
<thead>
<tr>
<th>Option</th>
<th>Explanation</th>
</tr>
</thead>
</table>
| 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 `-w LoginName1, LoginName2,...`, where `LoginName` 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 `-s`. |
mwCVS Tutorial

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

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

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

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

Assume these facts:

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

This chapter has these topics:

- “Part One: Turning on mwCVS” on page 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.

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.
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:

   \[ \text{protocol} : \text{Login@RepositoryAddress : RepositoryRoot} \]

   where \text{protocol} is the protocol you use to access the repository, \text{Login} is your
   login name for the repository, \text{RepositoryAddress} is the Internet Protocol
   (IP) or Domain Name System (DNS) address of the repository, and
   \text{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 4.1 Login window**

![Login window](image)

**Figure 4.2 VCS Messages window - login**

![VCS Messages window](image)
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

   ```bash
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.
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.
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.
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:

\[ \text{protocol} : \text{Login}@\text{RepositoryAddress} : \text{RepositoryRoot} \]

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.
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.

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
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 '_SCHEDULED' 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.
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.
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 ![VCS Status](image), 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
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
  - in Log dialog box 41, 56, 74
  - 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
  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
<table>
<thead>
<tr>
<th>J</th>
<th>learning about CVS 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>J</td>
<td>Log dialog box</td>
</tr>
<tr>
<td>J</td>
<td>Date 42, 57, 75</td>
</tr>
<tr>
<td>J</td>
<td>Default branch revisions 41, 56, 74</td>
</tr>
<tr>
<td>J</td>
<td>Do not list tags 42, 57, 75</td>
</tr>
<tr>
<td>J</td>
<td>Headers 41, 56, 74</td>
</tr>
<tr>
<td>J</td>
<td>Log folder 74</td>
</tr>
<tr>
<td>J</td>
<td>RCS file names 41, 56, 75</td>
</tr>
<tr>
<td>J</td>
<td>Recursive 75</td>
</tr>
<tr>
<td>J</td>
<td>Revisions 42, 57, 75</td>
</tr>
<tr>
<td>J</td>
<td>State(s) 42, 57, 76</td>
</tr>
<tr>
<td>J</td>
<td>User(s) 42, 57, 76</td>
</tr>
<tr>
<td>J</td>
<td>Log folder 74</td>
</tr>
<tr>
<td>J</td>
<td>Log menu command 41, 56, 74</td>
</tr>
<tr>
<td>J</td>
<td>Log message 31, 39, 49, 54, 64</td>
</tr>
<tr>
<td>J</td>
<td>Logging into remoteserver 44</td>
</tr>
<tr>
<td>J</td>
<td>Login dialog box</td>
</tr>
<tr>
<td>J</td>
<td>CVSRoot 43</td>
</tr>
<tr>
<td>J</td>
<td>Password 43</td>
</tr>
<tr>
<td>J</td>
<td>Login menu command 43</td>
</tr>
<tr>
<td>K</td>
<td>Macintosh VCS menu icon 10, 78</td>
</tr>
<tr>
<td>K</td>
<td>Make new working files read-only 15</td>
</tr>
<tr>
<td>K</td>
<td>menu commands</td>
</tr>
<tr>
<td>K</td>
<td>Add 39, 54</td>
</tr>
<tr>
<td>K</td>
<td>Checkout 66</td>
</tr>
<tr>
<td>K</td>
<td>Commit 30, 48, 63</td>
</tr>
<tr>
<td>K</td>
<td>Difference 33</td>
</tr>
<tr>
<td>K</td>
<td>Log 41, 56, 74</td>
</tr>
<tr>
<td>K</td>
<td>Login 43</td>
</tr>
<tr>
<td>K</td>
<td>Remove 40, 55</td>
</tr>
<tr>
<td>K</td>
<td>Status 32, 50, 70</td>
</tr>
<tr>
<td>K</td>
<td>Synchronize Document 25</td>
</tr>
<tr>
<td>L</td>
<td>Synchronize Selection 25</td>
</tr>
<tr>
<td>L</td>
<td>Synchronize Status 24</td>
</tr>
<tr>
<td>L</td>
<td>Tag 36, 51, 71</td>
</tr>
<tr>
<td>L</td>
<td>Update 26, 44, 59</td>
</tr>
<tr>
<td>L</td>
<td>menu, context 23</td>
</tr>
<tr>
<td>L</td>
<td>modifying a file, tutorial 84</td>
</tr>
<tr>
<td>L</td>
<td>Module 66</td>
</tr>
<tr>
<td>L</td>
<td>more information about CVS 6</td>
</tr>
<tr>
<td>L</td>
<td>Move tag if it exists 37, 52, 72</td>
</tr>
<tr>
<td>L</td>
<td>mwCVS</td>
</tr>
<tr>
<td>L</td>
<td>about 5</td>
</tr>
<tr>
<td>L</td>
<td>configuring 13, 21</td>
</tr>
<tr>
<td>L</td>
<td>disabling in the global scope 11</td>
</tr>
<tr>
<td>L</td>
<td>disabling in the project scope 12</td>
</tr>
<tr>
<td>L</td>
<td>enabling in the global scope 9</td>
</tr>
<tr>
<td>L</td>
<td>enabling in the project scope 10</td>
</tr>
<tr>
<td>L</td>
<td>file commands 26</td>
</tr>
<tr>
<td>L</td>
<td>installation 7</td>
</tr>
<tr>
<td>L</td>
<td>installing on a Macintosh host 8</td>
</tr>
<tr>
<td>L</td>
<td>installing on a Windows host 7</td>
</tr>
<tr>
<td>L</td>
<td>integrating with ssh 16</td>
</tr>
<tr>
<td>L</td>
<td>options</td>
</tr>
<tr>
<td>L</td>
<td>Command Output 15</td>
</tr>
<tr>
<td>L</td>
<td>Compatible with WinCVS 1.2 16</td>
</tr>
<tr>
<td>L</td>
<td>Make new working files read-only 15</td>
</tr>
<tr>
<td>L</td>
<td>Use gZip level 9 compression 15</td>
</tr>
<tr>
<td>M</td>
<td>overview 5</td>
</tr>
<tr>
<td>M</td>
<td>project commands 44</td>
</tr>
<tr>
<td>M</td>
<td>recursive commands 58</td>
</tr>
<tr>
<td>M</td>
<td>system requirements 6</td>
</tr>
<tr>
<td>M</td>
<td>turning off in the global scope 11</td>
</tr>
<tr>
<td>M</td>
<td>turning off in the project scope 12</td>
</tr>
<tr>
<td>M</td>
<td>turning on in the global scope 9</td>
</tr>
<tr>
<td>M</td>
<td>turning on in the project scope 10</td>
</tr>
<tr>
<td>M</td>
<td>tutorial 77</td>
</tr>
<tr>
<td>M</td>
<td>accessing the repository 78</td>
</tr>
<tr>
<td>M</td>
<td>checking out a module 81</td>
</tr>
<tr>
<td>M</td>
<td>committing the modified file 87</td>
</tr>
<tr>
<td>M</td>
<td>modifying a file 84</td>
</tr>
<tr>
<td>M</td>
<td>turning on mwCVS 78</td>
</tr>
<tr>
<td>M</td>
<td>version control system commands 24</td>
</tr>
<tr>
<td>M</td>
<td>mwCVS remove 5, 40, 55</td>
</tr>
<tr>
<td>M</td>
<td>mwCVS Results window 66, 83, 88</td>
</tr>
</tbody>
</table>
Version Control Plug-in 1.3 for CVS

N
-\text{n} 42, 57, 75
-\text{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
-\text{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
-\text{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
-\text{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
-\text{R} 41, 56, 75
-\text{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
Version Control Plug-in 1.3 for CVS

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

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
-\texttt{z} 15