

CVS Command Summary

File Status

```
$ cvs status -v [files...]
```

Displays the status of the working copy/tree. The `-v` switch is recommended as this causes all the available tags (regular and branch) to be displayed.

```
$ cvs update [files...]
```

Brings the working copy/tree in sync with the repository. This reconciles your work with any revisions committed to the source repository since your last checkout or update.

```
$ cvs diff [files...]
```

Show differences between your working copy/tree and the repository, or between different revisions in the repository.

File watching

```
$ cvs watch [on|off|add|remove] [files...]
```

- Turn on/off read-only checkouts of files.

- Add/remove notifications of actions on files (edit, commit, unedit)

```
$ cvs watchers [files...]
```

See who is watching a file.

```
$ cvs editors [files...]
```

See who is editing a watched file.

Editing files and committing changes

```
$ cvs edit [files...]
```

Get ready to edit a watched file. All of the files in the CRTM tree are watched and are thus checked out read-only. This comand make the working copy editable.

```
$ cvs commit [files...]
```

Check any changes in your working copy/tree into the repository. This will make watched files read-only.

```
$ cvs unedit [files...]
```

Undo an edit command on a watched file, i.e. abandon all changes and revert to most recent repository revision. Makes file read-only.

```
$ cvs add [files...]
```

Add a new file or directory to the repository. In the case of files, to make the changes permanent in the repository, you must then commit the file.

```
$ cvs log [files...]
```

Display history information for files.

File tags and branches

```
$ cvs tag tagname [files...]
```

Adds a symbolic tag to checked out version of a file.

```
$ cvs tag -F tagname [files...]
```

Move the symbolic tag *tagname*, if it exists, to the latest checked out version of a file.

```
$ cvs tag -b tagname [files...]
```

Creates a branch named *tagname*. Note that the branch is created in the repository, not your working copy/tree.

```
$ cvs update -r tagname [files...]
```

Checkout the revisions specified by the tag *tagname*. Note that this makes the tag “sticky.” For branch tags, this command switches the working copy to the specified branch.

Deleting files

```
$ cvs remove [files...]
```

Remove an entry from the repository. File must already be deleted. Use commit to put removed file in the CVS “attic.”

```
$ cvs release directory
```

Indicate that a directory is no longer in use. This should be done if you want to delete the directory.