

CVS How To

CVS needs to know where the "repository" is kept:

```
CVSR00T=/usr/local/cvs/repository; export CVSR00T
```

Creating a new project (importing)

Go to the top level of your project

```
% cd myproj
% ls
README.txt a-subdir/ b-subdir Hello.java
% cvs import -m "initial import into cvs" my_project iris_dmc
start
```

IMPORTANT: This will suck ALL files under myproj into the repository. You should do a clean before importing.

Checking files out of the repository. Make sure to do this after the initial import.
This is what you do if you want to make changes to files...

```
% cd ~/work
% cvs checkout my_project
% ls
my_project
% cd my_project
ls
CVS/ README.txt a-subdir/ b-subdir Hello.java
```

Exporting files from the repository.
This is what you do if you just want to view files

```
% cd ~/work2
% cvs export -D now my_project
% ls
my_project
% cd my_project
ls
README.txt a-subdir/ b-subdir Hello.java
```

NOTE: there's no "CVS" directory. You can also specify a tag or a date instead of "now"

Making a change to a file

```
% cd ~/work/my_project
vi hello.java...
% cvs update
M hello.java
blah blah blah
    The "M" tells you that the file hello.c has been modified
```

```
% cvs diff
```

Prints a bunch of stuff telling you what you've changed in hello.c
There are lots of different ways of running cvs diff

```
% cvs commit -m "fixed a bug in hello.java" hello.java
```

This stuffs the changes in hello.c the repository.

You can also do this recursively from any level in your project:

```
% cvs commit -m "fixed numerous bugs"
```

Tagging a build

```
% cd ~/work/my_project  
% cvs tag Release_1_0
```

You can export this build now like this:

```
% cvs ~/work2  
% cvs export -r Release_1_0 my_project  
Adding a text file  
% cd ~/work/my_project  
% ls  
CVS/ README.txt a-subdir/ b-subdir Hello.java Foo.java  
% cvs add Foo.java  
% cvs commit -m "added blah" Foo.java
```

You can do the same thing with directories.

Adding a binary file

CVS is configured to handle certain file types as binary. Look at the file
/usr/local/cvs/CVSROOT/cvswrappers to determine
which files are treated as binary

```
% tail /usr/local/cvs/repository/CVSROOT/cvswrappers  
####  
*.jar -k 'b'  
*.zip -k 'b'  
*.class -k 'b'  
*.jpg -k 'b'  
*.gif -k 'b'
```

If your file type is not in the list, use the -kb option to the add command

```
% cd ~/work/my_project  
% ls  
CVS/ README.txt a-subdir/ b-subdir Hello.java foo.jar
```

```
% cvs add -kb foo.jar
% cvs commit -m "added blahh" foo.jar
```

Running CVS through SSH

You can access the repository via ssh by doing the following:

```
% CVS_RSH=ssh; export CVS_RSH
%
CVSROOT=:ext:yourname@machinename:/usr/local/cvs/repository; export
CVSROOT
```

e.g. Machinename = gravity.usc.edu (any machine that can get to \$CVSROOT)