

CVS How-to

helpdesk@stat.rice.edu

17th July 2003

This how-to gives introduction on how to set up and use cvs (Concurrent revision system) on stat network.

Briefly, cvs system maintains a center repository and let you check out, check in files conveniently both locally and remotely. It allows you to work on the most current version of your files from any machine. The usual scenario of using cvs system is:

```
cvs checkout project      # check out a module
cd project
... work on the project ...
cvs commit                # commit changes, check in
cd ..
cvs release -d myproject  # delete local copy. optional
```

Set up

Server set up

1. Log on to stat007
2. `mkdir ~/CVS`
3. Add the following two lines to your `.cshrc`

```
setenv CVSROOT ~/CVS
setenv CVSEEDITOR vi # or your favorite editor
```

4. Source `.cshrc` to activate `$CVSROOT` and `$CVSEEDITOR`
5. `cvs init`
6. If you are going to use `rsh`, edit or create `~/rhosts`, add machines/usernames that you will be working on. '-' can be used as any machine or user. For example:

```
your-laptop.stat.rice.edu yourusername
- yourusername
```

Note:

- (a) Only machine within Rice network can `rsh` to `stat007`.
- (b) Your home machine can `rsh` via a VPN connection but the machine appears to `stat007` as something like `vpnrice-10-2.rice.edu` (dynamically). If you would like to `rsh` from home, you will have to use the second form which allows `rsh` from any rice machine to `stat007`.
- (c) The second line is useful but dangerous. Do not use it if possible.
- (d) The most current version of `cvs` is under `/home/helpdesk/newbin`

Client set up

1. Add the following lines to your `.cshrc`

```
setenv CVSROOT ":ext:username@stat007.stat.rice.edu:/home/username/CVS"

setenv CVSEEDITOR vi # or your favorite editor
setenv CVS_RSH rsh
```

Note:

- (a) Substitute `username` with your real user name on `stat007`.
- (b) Use `setenv CVS_RSH ssh` if the machine is off campus or is not listed in `.rhosts`. You can also use `ssh` for in campus connections but you will have to enter password for each `cvs` command. (See the last part of this howto if you would like to avoid repeated password.) In general, `rsh` is easy to use but not secure and can only be used within Rice (or via VPN). `ssh` is secure and universal but need to input password.

2. Source `.cshrc` to activate these variables.
3. test your `rsh` connect if your set `CVS_RSH` to `rsh`

```
rsh stat007.stat.rice.edu 'echo $path'
```

4. If not successful, check your `.rhosts` on `stat007`.
5. You should be able to run all `cvs` commands.

CVS Usage

Helpdesk has written some shell scripts to speed up the use of `cvs` system. However, you would better read some tutorials and understand how basic `cvs` commands work.

Module are just directories in `cvs` system. It contains files or subdirectories. `cvs` allows you to define modules within a module but that is out of the range of this introduction.

create a module

by importing a folder

```
cd stat410
cvs import stat410 vendor release
```

Note:

1. You would better clean your folder and leave only text files you would like to import.
2. vendor release are two parameters cvs import need. You can change them if you understand what they mean.
3. You will be prompted to enter some comment. Use `cvs import -m "stat course" stat410 vendor release` to avoid it.

by helpdesk cvsadd

```
cvsadd dotfiles .cshrc .muttrc .emacs
```

will add these three files to module dotfiles. If dotfiles does not exist, you will be prompted to create it.

list files on center repository or some of the modules

Use helpdesk cvsls

```
cvsls # list module names
cvsls -a # list all modules
cvsls module1 module2 # list files of module1 module2
```

Read `cvsls` to see how it works.

Add files to existing module

usual cvs sequence

```
cvs co module
cd module cp files .
cvs add files
cvs commit
```

use helpdesk cvsadd

```
cvsadd module files
```

It is basically an automation of the above procedure.

Add directories to a existing module

It is easier to import a directory as a new module. If you would like to add a directory to an existing module:

usual cvs sequence

```
cvs co module
cd module
cp -r directory . # or mkdir and add files
cvs add directory
cd directory
cvs add * # and keep going on.
```

helpdesk cvsadd

```
cvsadd module directory
```

cvsadd does the adding procedure recursively.

Check in and check out

Usually, you work in this way

```
cvs co project
cd project
...work on it ...
cvs commit
```

After a while, if you have modified some of the files from other places

```
cvs update # read the newest version from center repository.
```

If you would like to remove local copy,

```
cd .. cvs release -d project
```

Note: You can

```
cd ..
rm -rf project
```

directly. cvs release will check if you have un-committed changes before removal.

Remove file or directory from an existing module

usual cvs sequence

```
cvs co module
cd module
cvs remove -f file # tell cvs to remove
cvs commit # really remove
```

helpdesk cvsremove

```
cvsremove module files-or-directories
```

automation of above procedure.

Retrieve files or directories from repository

usually `cvs`

```
co module cp module/file somewhere com -rf module
```

By **helpdesk cvsget**

```
cvsget module files
```

Dealing with binary files

add `-kb` option to `cvs add` or `cvsadd` command.

Dealing with keywords

Keywords are words such as `Log`, `Date`, that are delimited by `$ $` and will be managed by `cvs` system. I can not write it here since it will be replaced by `cvs` and will not be recognized. See the last part of this file how `Date`, `Log` are replaced by `cvs`. See `/home/helpdesk/bin/cvs*` for examples.

Tired with repeated SSH password?

It is not easy, but you can use "public-key" authorization instead of "password" authorization so you do not have to enter password each time. However, you are under the risk that:

**ANYONE WOULD BE ABLE TO LOG ON TO STAT007
IF THEY HAVE ACCESS TO YOU LOCAL ACCOUNT!**

Anyway, if you will work off-campus a lot and would like to take the risk, here is how:

1. On your LOCAL machine, run `ssh-keygen2`. Enter return for the two questions. If 'command not found', you are running `ssh1` and can not use this method. The following instruction will assume that you get two files: `~/.ssh2/id_dsa_2048_a` and `~/.ssh2/id_dsa_2048_a.pub`. Change file names accordingly if you get different keys.
2. On you LOCAL machine, run

```
cd ~/.ssh2  
echo "IdKey id_dsa_2048_a" > identification
```

3. Transfer `id_dsa_2048_a.pub` to `stat007:~/ssh2`. You can use `sftp` to do this.

```
sftp stat007.stat.rice.edu
...
> cd .ssh2
> put id_dsa_2048_a.pub
> quit
```

`scp` can be used too.

```
scp id_dsa_2048_a.pub stat007.stat.rice.edu:~/ssh2
```

4. Log on to `stat007`. Run

```
cd .ssh2 echo "Key id_dsa_2048_a.pub" > authorization
```

5. Log off and try

```
ssh stat007.stat.rice.edu
```

You should not be prompted for a password. All cvs operations will also be executed without asking a password.

If you are using `Openssh` through Linux or Windows/Cygwin and would like to connect to `stat007` without password, here is a undocumented trick from internet:

1. Make sure you are using `openssh` $\geq 2.2.x+$
2. Run

```
ssh-keygen -d -f $HOME/.ssh/id_dsa
cd $HOME/.ssh
ssh-keygen -x -f $HOME/.ssh/id_dsa > ssh2.pub
echo "IdKey id_dsa" > identification
```

3. Copy `ssh2.pub` to `stat007:~/ssh2`
4. Use `ssh2.pub` in authorization file

```
echo "Key ssh2.pub" > authorization
```

Warning

cvs scripts are under development. Helpdesk is not responsible for data lose or any consequences of using them. I will appreciate your help if you can report any bug to me.

The public-key method is dangerous. Helpdesk is not responsible for any data lose or invasion to your account due to its use.

helpdesk@stat.rice.edu

\$Date: 2003/07/17 02:49:21 \$