By
Syed Azhar Hussain
Synametrics Technologies

Collaborator's Tool

Concurrent Versions System

In the Name of Allah, The Most Gracious, and Most Merciful

"Collaborator's Tool: Concurrent Versions System"

Syed Azhar Hussain
azhar[at]synametrics[dot]com

v1.0, 20 September 2003

Abstract: *"This document covers brief information about CVS and its practical knowledge; how to configure and use CVS. Targeted readers are collaborative programmers and project managers. CVS configurations used in this document have been implemented and tested using CVS Server on RedHat9 (or any *inx[1] distribution) and clients either on *inx[1], Windows (WinCVS) or Collaborator's Visual Programming tool (JBuilder8). This document's key purpose is to show collaborators, a heaven path where digital data involved."*

Hypothesis: Reader knows basic *inx[1], and already installed its (any) distribution.

What is CVS?

Generically, I define CVS (Concurrent Versions System) as, "A tool for collaborators to systematize their electronic data."

CVS is an authoritative tool which allows simultaneous and interrelated development of software by multiple users to be done in an organized and consistent manner. Using it, you can maintain the history of changes in your source files.

For example, bugs sometimes creep in when software is modified, and you might not detect the bug until a long time after you make the modification. With CVS, you can easily retrieve older Versions to see exactly which change caused the bug. This can at sometimes, be a big help.

CVS is very helpful for managing releases and controlling the concurrent editing of source files among multiple authors. Instead of providing Versions control for a collection of files in a single directory, CVS provides Versions control for a hierarchical collection of directories consisting of revision controlled files. It can be used for storing "C", "C++", Java, Perl, HTML and other files.

[1] *inx mean either Unix or Linux

CVS short-cut details:

- Versions control
- Connecting multiple developers/persons together on one project
- Document management and archiving
- Graceful way to carry out one project with multiple machines and still manage the Versions
- Allows free experimenting on project
- Following the growth of the project
- Automatic, constant and forced backup
- Gives freedom to choose afterwards, when the program is ready
- Saves all the Versions for later use
- Clean way of saving only the necessary files and managing projects
- Gives freedom to develop on multiple machines simultaneously
- Multi-user support and synchronization (file locking, etc.)
- Keyword substitutions
- Development of several Versions at the same time
- Multiplatform-support (works even on obscure platforms like Windows)
- One important reason, very useful tool without any cost

Deep thinking, CVS Resources

If you want to learn CVS seriously, Hmm I like your concerns, following links will save your googling[2] time.

- Official CVS web site: www.cvshome.com
- Free CVS book, learn CVS A to Z: www.cvsbook.red-bean.com
- Official manual: www.cvshome.org/docs/manual/ recommended for speedy in-depth reading

Essential Terms

- Repository: The master copy where CVS stores a project's full revision history. Each project has exactly one repository.

- Working copy: The copy in which you actually make changes to a project. There can be many working copies of a given project. Generally, each developer owns a separate working copy.

- Check out: To request a working copy from the repository (code or files storage area). Your working copy reflects the state of the project as of the moment you checked it out; when you and other developers make changes, you must use the commit and update commands to "publish" your changes and view others' changes.

[2] I mean searching using google.com

- Commit: To send changes from your working copy into the central repository. Also known as check in.

- Conflict: The situation when two developers try to commit changes to the same region of the same file. CVS notices and points out conflicts, but the developers must resolve them.

- Log message: A comment you attach to a revision when you commit it, describing the changes. Others can page through the log messages to get a summary of what's been going on in a project.

- Revision: A committed change in the history of a file or set of files. A revision is one "snapshot" in a constantly changing project.

- Update: To bring others' changes from the repository into your working copy and to show whether you're working copy has any uncommitted changes. Be careful not to confuse this with the commit operation described above; Commit and Update are complementary, not identical, operations. Here's a mnemonic to help you remember: update brings your working copy up to date with the repository copy.

Configure CVS - Starts Real Action

CVS deployed in two parts (good practice);

CVS server either installed on Unix/Linux (www.cvshome.org), Microsoft Windows (www.cvsnt.org)

CVS client either installed on Unix/Linux (www.cvshome.org), Microsoft Windows (www.wincvs.org) or Macintosh (www.wincvs.org)

STEP "A": [I don't know CVS is installed or not]

Perform the following steps to check whether CVS is already installed in your box:

1. Now, You can quickly check with

   ```
   user@linux# whereis cvs
   ```

   If the command replies with a line containing /usr/bin/cvs; it means that CVS is installed and you jump to Step "B".

   If it doesn't; you'll have to install it first. (see Step# 2)

2. Install CVS includes

   a. When installing RedHat9 + select custom package install, and select packages:
      i. CVS
      ii. Webmin, this tool is integrated with CVS, using this tool you setup CVS very quickly and also browse your repository through a webpage. I personally recommend it with CVS. Mostly in collaborative environments people use windows or other OS, they watch repository using their native browsers. Isn't cool?

   OR

   b. Find CVS and Webmin RPMs from www.rpmfind.net then download and install.

STEP "B": [CVS already installed]

We are going to setup:

- CVS Server at RedHat9
- CVS Client at ReadHat9 using commands, Windows using WinCVS and Visual Programming tool like JBuilder8

There are different ways to setup CVS server[3] (which is available over network); the option discussed in this document is "pserver", CVS password authenticated server. The CVS client connect to the CVS pserver using a CVS password.

Configuration of CVS Server/Client divided into two tasks:

1. Administrative/Project Manager task [setup server + initial project]

   a. Setup Repository & Server: Create repository with this command

   ```
   user@linux#cvs -d /usr/local/cvsroot init
   ```

   This will create a default empty repository at the path specified. The latest Version of Webmin (1.110 at the time of writing, www.webmin.com) has a CVS setup function. Once you have the repository in place you have to run the CVS server from xinetd so that it can be accessed from the outside world (your local LAN). To do these simply press the "Setup CVS Server" button on the CVS Server page (Using any browser go to Webmin Control Panel http://YourLinuxBox:10000, enter root password and go to Servers

[3] For CVS server setup on Microsoft platform, visit www.cvsnt.org, or visit http://www.cvsnt.org/wiki/HelpContents for detailed information.

and clicks CVS server.). All this actually does is; making an entry in the file /etc/xinetd.conf of the form

```
service cvspserver
{
     port= 2401
     socket_type = stream
     protocol = tcp
     user = root
     wait = no
     type = UNLISTED
     server = /usr/bin/cvs
     server_args = -f --allow-root /usr/loca/cvsroot pserver
     disable = no
}
```

It is simply a service definition. Note that this could also be included as a separate file in the /etc/xinetd.d directory.

b. Environment Variable CVSROOT: It is probably also a good idea to include the environment variable CVSROOT. I find that it is the best to included CVSROOT in /etc/profile. This seems to be no security problems associated (with all the users knowing where the repository is), because only permitted users are able to access repository.

   Add these few lines to the /etc/profile file after the "export PATH USER LOGNAME MAIL HOSTNAME HISTSIZE INPUTRC" line. You will need to log out and log back in to cause the root process to re-read the file. Alternatively you can "su - your_username" with every new console window.

```
#CVS
CVSROOT=/usr/local/cvsroot
export CVSROOT
```

c. Adding Users: CVS allows you to add users independent of the underlying OS installation which is at the same time, a good thing and a bad thing. Probably the easiest way from a management point of view is to use the *inx (Unix/Linux) users for CVS as well although this does have a few minor security issues since this is just a home network we will grin and bear it.

   From the CVS set up page in Webmin select the options to keep users synchronize with the *inx users. This is useful to keep the record-keeping at minimum. Since there is no option to bring across existing users you have to add them manually. Click "Add new CVS User" and then enter the details. Set the password for the CVS account to something other than the *inx account's password as it is a security risk due to a very basic password encryption technique used at the client machine. For the most part you can't actually use the login password since it is generally stored in MD5 format and can’t be read by CVS. However, it doesn’t stop you from setting a password manually that are the same as the account password though.

Note, when you add a user like this Webmin doesn't know the there is already a user on the system with that name. In /usr/loca/cvsroot/CVSROOT/passwd update the users to indicate that they are actually known system users rather than just CVS users. This is done by adding the name of the system account you want the cvs account to run as at the end of the line. Each line is made up of three parts

```
<cvs_acc_name>:<cvs_password>:<sys_acc_name>
```

d. <u>Adding “cvs” Group:</u> Now set up a user group called “cvs” or some such and add your users to this group. Give this group permission to write to the /usr/local/cvsroot directory and all subdirectories.

   This can be done using KUser and Konqueror or by following commands, but logged as a super user [root]:

```
root@linux# chgrp -R cvs $CVSROOT
root@linux# chmod o-rwx $CVSROOT
root@linux# chmod ug+rwx $CVSROOT
```

   If using KUser and Konqueror, then select the /usr/local/cvsroot directory, right click and select properties. In the permissions tab select Write Entries and Special (Set GID). Change the Group Ownership to be what ever the group was that you created above. Check "Apply changes to all subdirectories and their contents" and then press ok. Once this is done the CVS users "own" the “cvs” repository and can from now on create lock files in it.

   To keep things simple at this stage we are giving all users the ability to read and write the repository. As things change you will probably want to change this. Allowing everybody access is the default set up.

e. <u>Setup initial Project files/directory:</u> For a new project, the easiest thing to do is probably to create an empty directory structure, like this:

```
user@linux# mkdir SpamMarshall
user@linux# mkdir SpamMarshall/doc
user@linux# mkdir SpamMarshall/src
```

   Put or create some files in these directories. After that, you use the import command to create the corresponding (empty, if no file) directory structure inside the repository:

```
user@linux# cd SpamMarshall
```

Add or Create file in the directory “SpamMarshall”.[4]

```
user@linux# cvs import –m "Create Initial Project Files"
ITANetworks/SpamMarshall ITA start
```

Then, use “add” to add files (and new directories) as they appear. Check that the permissions CVS sets on the directories inside $CVSROOT are reasonable.

```
user@linux# ls –l $CVSROOT
```

```
User group displayed must be “cvs” and user from “cvs”
group only, if not then run commands again:
```

```
root@linux # chgrp -R cvs $CVSROOT
root@linux # chmod o-rwx $CVSROOT
root@linux # chmod ug+rwx $CVSROOT
```

f. Create project module: One repository contains one or more projects, one projects contains different modules. In our case, we created a project name “SpamMarshall”. This step is to define the module in the `` `modules' `` file at $CVSROOT/CVSROOT/modules. This is not strictly necessary, but modules can be convenient in grouping together related files and directories. Note: Its also recquired for Jbuilder8, beacause Jbuilder8 scan modules from CVS server.

In simple cases these steps are sufficient to define a module:

a. Get a working copy of the modules file.

```
user@linux# cd SpamMarshall
user@linux# cvs checkout CVSROOT/modules
user@linux# cd CVSROOT
```

b. Edit the file and insert a line that defines the module. You can use the following line to define the module `` `SpamMarshall' ``:

```
SpamMarshall  ITANetworks/SpamMarshall
```

c. Commit your changes to the modules file.

```
user@linux# cvs commit -m "Add the module" modules
```

[4] For testing, create file “readme.txt”, enter text and close it.
user@linux# vi readme.txt

d. Release the modules module.

```
user@linux# cd ..
user@linux# cvs release -d CVSROOT
```

e. Get modules listing from modules database.

```
user@linux# cvs checkout -c
```

Make sure module is installed or not, simply open a file $CVSROOT/CVSROOT/modules using any editor, you find a line at bottom of file `”SpamMarshall  ITANetworks/SpamMarshall“.`

g. Now, project module is ready to use; using CVS client, WOW!

2. Collaborative Programmer task

Depending on what you are intending to do with CVS depends a little on which tools you will want to use on the client side to interact with your repository. I use JBuilder as my development environment of choice for Java. JBuilder has built in CVS functionality that works very well. When, I need a little more control, I use WinCVS which is a very good Windows based CVS browser. When, I need more control, simply telnet CVS server machine.

You view your repository online through any web browser. Login at Webmin, go to CVS server page then click repository and enjoy.

There are three ways (commonly), collaborator use CVS server: using telnet, any GUI tool or any Visual Programming IDE. I am discussing briefly all three below.

a. Checkout project module specified by Project Manager

o Linux tty (telnet):

Connect using any telnet client to your CVS server, following are the basic commands used for normal CVS session:

Checkout:

```
user@linux# mkdir test
user@linux# cd test
user@linux# cvs checkout ITANetworks/SpamMarshall
```

Working copy created.

```
user@linux# ls –l
user@linux# cd SpamMarshall
```

```
user@linux# ls –l
```

Change file “readme.txt”, we created for testing. Modify this file.

Ask CVS, any change in project:

```
user@linux# cvs diff
```

This command displays the difference.

Commit this change:

```
user@linux# cvs commit
```

Get latest update from CVS server:

```
for particular module:

user@linux# cvs update ITANetworks/SpamMarshall

OR for whole repository:

user@linux# cvs update ITANetworks

OR generically, (must be in directory of working copy):

user@linux# cvs update
```

Note: In above work, we connected to CVS server where we already setup repository, but if you want to use CVS client from another Linux box (having CVS rpm already installed) to connect to CVS server, you still use the same commands; moreover, you need to do some extra effort. You will have to setup environment variable, then using “cvs login” command (this command prompt for password, enter password then) logged into CVS server.

```
#Environment variables export
CVSROOT=:pserver:user@cvs.hostname.org:usr/loca/cvsroot/
CVS_RSH="ssh"
```

Then, finally login using.

```
user@linux# cvs login
```

- WinCVS:

  WinCVS Graphical user interface helps to utilize full power of CVS (for experts) and quickly learn basics (for beginners). Download and Install WinCVS from www.wincvs.org. Open WinCVS select Admin then Preferences. In the CVSROOT text area enters the path to your

CVSROOT. Typically this will be something like azhar@myserver.com:/usr/local/cvsroot which tells WinCVS which server to login to which repository and who is logging in e.g.

```
<user_name>@<server_name>:<path_to_repository>
```

Now try logging into the server Admin>Login; this should hopefully ask you for your password which when given logs you in. You will probably now want to check out a module from the server and get a local copy for working on. To do this simply select Create>Checkout Module and enter the module name. The module names correspond to the name of the directory in the CVS repository you can enter either a root level project or one further into the tree.

- JBuilder8:

  Just pull out project from CVS server, select "pserver" and enter hostname, user ID, repository path "/usr/loca/cvsroot/", password when prompted scan module and select main branch then click Finish button. You see the magic.

  Note: Please make sure that CVS server and client (JBuilder's CVS client) are same (means same Version).

  ```
  user@linux# cvs Version
  ```

  Note for JBuilder8 users, if your CVS server Version is 1.11 or later then you will have to update your JBuilder8's CVS client. How? Tricky question. Go to JBuilder8 directory <home_directory>/bin/cvs.exe, you will have to update this cvs.exe (because it contain old CVS server protocol). Hmm.. How? Another tricky question. Download (latest Version) and Install WinCVS then copy cvs.exe from its home directory to JBuilder8 bin directory. Now, JBuilder8's CVS client updated.

  You want to learn in-depth about JBuilder8 with CVS, visit http://info.borland.com/techpubs/jbuilder/jbuilder8/teamdev/tutcvs/cvs_tutorial.html

- Visual Studio:

  Visit www.kryptonians.net/cvs/wincvs_and_ide.html, CVS Integration Add-in for Microsoft Visual Studio.

  Visit www.jalindi.com/igloo/, CVS SCC plug-in for Microsoft Visual Studio and other compliant IDEs.

b. Do work in your working copy
c. If any change(s) from other collaborators, update working copy from server, commit your change
d. Feel relieved, Enjoy and, say “Big Woo! for Open Source & CVS”

After Everything Else

I would welcome your opinion(s). I'm not really a Connoisseur with CVS, so there may be much easier ways to do some of these things too.
