CVS Lecture

Victor Mircea

November 4, 2004

These are notes from a lecture on CVS, please try to keep these handy throughout the Intranet2 development so that when you have an issue, or a question you can refer to this first.

1. **So what is CVS?** CVS is a very nice versioning system that allows developers to work together using one central source tree, with each use having a local copy of the master tree. CVS provides for easy conflict resolution between code changes, and easy version tracking, as well as emails sent to a mailing list whenever a change is made.

2. **So how do I start using CVS?** To import an existing directory into CVS, you would do the following (assuming that cvs is already installed on the computer).

   ```
   root@xeon [~]# mkdir /root/dev
   root@xeon [~]# cd /root/dev/
   root@xeon [~/dev]# CVSROOT=~/dev;export CVSROOT
   ```

   This sets the directory that you want to use for your CVS storage directory. Then to import the files, assuming they are in /root/project:

   ```
   root@xeon [~/project]# cvs import -m "Sample Program" project sample start
   N project/testfile
   N project/testfile2
   ```

3. **So I have a project setup, now what?** After you have a project all setup, which will be the case with Intranet 2, to checkout a copy to some local dir you need to do the following:

```
root@xeon [~/project]# cd
root@xeon [~]# cvs checkout project
cvs checkout: Updating project
U project/testfile
U project/testfile2
```

After this is done, you will have the most recent of the code that the repository had (excluding any code that conflicted with modifications that you made).

4. **I made a change, how do I save it?** After you have made a change to a file in your local directory, or added any new files, or directories, you will need to do:

```
root@xeon [~/project]# cvs commit -m "added some stuff" testfile
Checking in testfile;
/root/dev/cvsroot/project/testfile,v <-- testfile
new revision: 1.2; previous revision: 1.1
done
```

5. **Some other commands...** Another very useful command is cvs status, assume that a developer has updated a file in the repository, and you would like to edit that file locally. You can use cvs status as used belowed to find out if a file has been changed in the repository:

```
root@xeon [~/project]# cvs status testfile
===================================================================
File: testfile          Status: Needs Patch

   Working revision:    1.1.1.1 Thu Nov  4 02:32:46 2004
   Repository revision: 1.2 /root/dev/project/testfile,v
   Sticky Tag:          (none)
   Sticky Date:         (none)
   Sticky Options:      (none)
```

This shows that the file has been modified and it should be patched before you modify it. ALWAYS check the cvs status of a file before modifying it in order to avoid headaches, but CVS can deal with multiple changes to a file as well. The statuses that a file can have are:

(a) Up-to-date: The file is identical with the latest revision in the repository.

(b) Locally Modified: You have edited the file, and not yet committed your changes.

(c) Needing Patch: Someone else has committed a newer revision to the repository.

(d) Needs Merge: Someone else has committed a newer revision to the repository, and you have also made modifications to the file.

You can avoid almost ALL instances of having to merge a file by checking to see if someone has modified it before you modify it. But in the event that two people do modify a file, CVS will attempt to smart merge it, if the two (or more) modifications were in different parts of the file, then they will both happen, and there will be no conflict. However, if they were close together, CVS will show you a diff like output and ask you what it should use.

6. **How do I add and remove files from CVS?** The last thing that you really need to know is how to add and remove files from CVS. Look at the following examples:

```
root@xeon [~/project]# touch test
root@xeon [~/project]# cvs add test
cvs add: scheduling file ‘test’ for addition
cvs add: use ’cvs commit’ to add this file permanently
root@xeon [~/project]# cvs commit test

root@xeon [~/project]# rm test
rm: remove regular empty file ‘test’? y
root@xeon [~/project]# cvs remove test
cvs remove: removed ‘test’
root@xeon [~/project]# cvs commit test
```