

Everything You Didn't Want To Know About CVS

Daniel Bradley
ARC Centre in Bioinformatics

What is CVS

- CVS (Concurrent Versions System) is a derivative of RCS (Revision Control System)
- An implementation of a *revision control* system

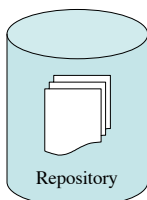
What does CVS do

- Keeps a record of all changes to files it manages
- Individual (or groups of) files can be tagged at specific versions for later retrieval

How is it used

- Undoubtedly a reason why CVS is so popular is its flexibility
- CVS can operate locally on a computer
- CVS can operate over a shared file-system (NFS, Windows shares)
- CVS can operate over the Internet (using SSH)
- CVS can operate in a client-server mode

Commands: init



- The repository is where CVS maintains file histories
- A repository can be created anywhere, by anyone
- A repository is just a standardized directory layout
- By convention the top-level directory of a cvs repository is called "cvsroot"

```
$ cvs -d /home/daniel/cvsroot init
```

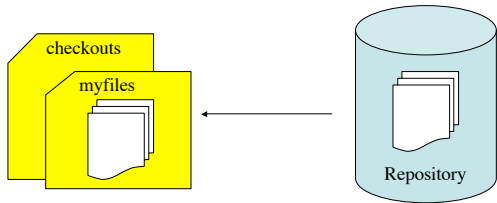
Commands: import



```
~:~$ cd myfiles
```

```
~/myfiles:$ cvs -d /home/daniel/cvsroot import itee start
```

Commands: checkout



```
~:$ mkdir checkouts; cd checkouts
~/checkouts:$ cvs -d /home/daniel/cvsroot co myfiles
```

Your checkout

- The *checkout* refers to your local copy of the files in the repository.
- Each directory will also have a “CVS” directory that contains the files:
 - Entries - list of directories and files (with current version) in the parent directory
 - Repository - position of parent directory relative to root
 - Root - location of the CVS root

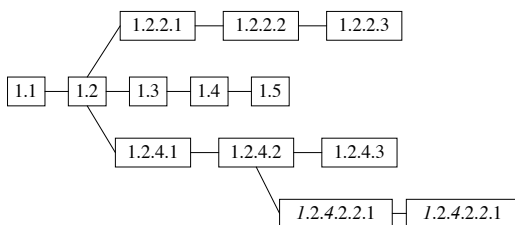
Your repository

- The directory structure of the repository is similar to that of your checkouts directory except
 - Within the top-level directory (cvsroot) is a directory CVSROOT, which stores configuration information regarding the repository
 - Each file imported into the repository is converted into a versioned file with a “v” suffix.

Versioned files

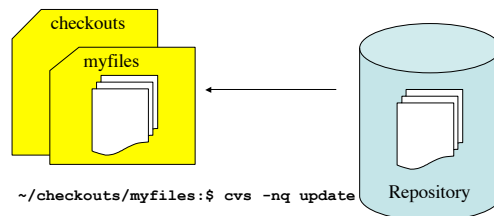
- The format of versioned files is similar as for RCS (see rcsfile manpage), it keeps track of changes using a tree structure based upon version identifiers
- Each file contains the latest version (the head) in full and diffs of each version increment
- The main trunk is identified by versions with two numbers eg. 1.1, 1.2, 1.3, branches are identified by their parent version + branch version + increment version eg 1.2.2.1, 1.2.2.2, 1.2.2.3

File Example



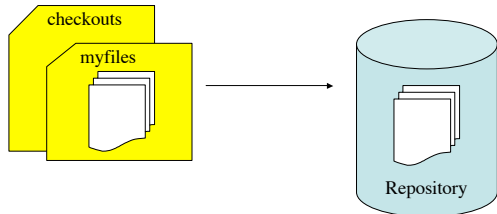
General form of version
 <branch>.<increment>.<branch>.<increment>...

Commands: fake update



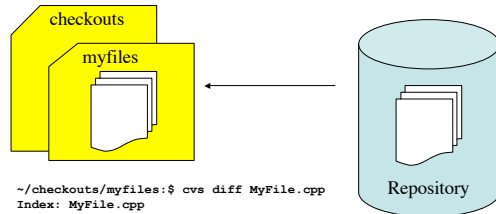
```
~/checkouts/myfiles:$ cvs -nq update
? MyNewFile.cpp
U FredsFile.cpp
M MyFile.cpp
```

Commands: add



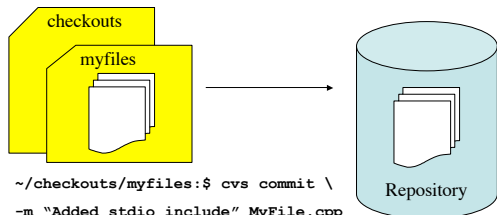
```
~/checkouts/myfiles:$ cvs add MyNewFile.cpp
cvs add: scheduling file `libpthread.so.link' for addition
Cvs add: use `cvs commit' to add this file permanently
```

Commands: diff



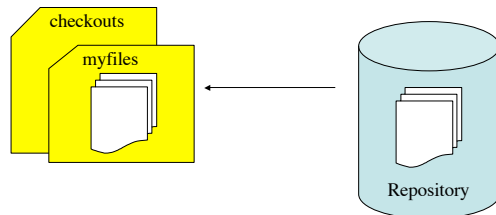
```
~/checkouts/myfiles:$ cvs diff MyFile.cpp
Index: MyFile.cpp
=====
RCS file: /Volumes/daniel/cvsroot/myfiles/MyFile.cpp,v
retrieving revision 1.1
diff -r1.1 CellularControl.cpp
10a11
> #include <stdio.h>
```

Commands: commit



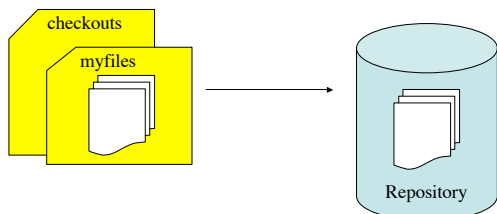
```
~/checkouts/myfiles:$ cvs commit \
-m "Added stdio include" MyFile.cpp
...
~/checkouts/myfiles:$ cvs commit \
-m "Initial add" MyNewFile.cpp
```

Commands: update



```
~/checkouts/myfiles:$ cvs -q update
U FredsFile.cpp
```

Commands: tag



```
~/checkouts/myfiles:$ cvs tag release_1_0
...
```

Benefits of revision control

- A file can be compared against any version in the repository
- You can return to a previous version of a file
- You can have concurrent versions of files
- You can easily collaborate with others
- You can merge a branch into another branch

Benefits of CVS

- Users do not need administrators to install a revision control service (either locally or on a server) as CVS is a user application
- The Repository is simple and is not susceptible to binary corruption
- The Repository can be rearranged without affecting the accessibility of files

Pitfalls of CVS

- Versioning is only relevant to individual files
 - Although tagging enables the manipulation of groups of files. Actions only have meaning in the context of single files.
 - No transaction support (commit/don't commit)
- No history of file meta data such as position

CVS kludges

- Moving/renaming files
 - Commit changes, tag release, delete checkout
 - Duplicate repository directory
 - Move file in repository
 - Checkout new repository
 - ?
 - Profit

Some other revision control systems

- Free (including Open Source)
 - Arch
 - Git (used by linux kernel developers and xorg)
 - Subversion (now available on Source Forge)
- Commercial
 - Bitkeeper (previously used by linux kernel developers)
 - StarTeam, Borland
 - Visual SourceSafe, Microsoft

References

1. http://en.wikipedia.org/wiki/Concurrent_Versions_System
2. http://en.wikipedia.org/wiki/Revision_Control_System
3. CVS Manual (cederqvist). <http://ximbiot.com/cvs/manual>
4. CVS Website. <http://www.nongnu.org/cvs>
5. http://en.wikipedia.org/wiki/List_of_revision_control_software