

Revision Control

Dr. Hyrum Carroll

February 1, 2016

Revision Control: Motivation

The typical problem

`http://www.phdcomics.com/comics.php?n=1531`

Does this look familiar?

```
-rw-r--r-- 1 usr1 Sep 19 16:53 simulation.f
-rw-r--r-- 1 usr1 Sep 19 16:53 #simulation.f#
-rw-r--r-- 1 usr1 Sep 19 14:38 simulation.f~
-rw-r--r-- 1 usr1 Sep 17 12:01 simulation.f.old
-rw-r--r-- 1 usr1 Feb 21 2014 simulation.f.bak
-rw-r--r-- 1 usr1 Sep 16 2014 simulation.f.orig
-rw-r--r-- 1 usr1 Dec 13 2010 simulation.f.from-BYU
```

Or maybe this?

```
-rw-r--r-- 1 usr1 Sep 19 16:53 simulation.f
-rw-r--r-- 1 usr1 Sep 17 12:01 simulation.f.2014.09.17
-rw-r--r-- 1 usr1 Feb 21 2010 simulation.f.2010.02.21
```

Revision Control: Motivation

Wouldn't it be nice if we had a system that:

- ▶ stored revisions of a file on demand
- ▶ stored comments on each revision
- ▶ allowed checking out any previous revision
- ▶ displayed the differences between any pair of revisions
- ▶ manage multiple lines of development
- ▶ resolve access conflicts between developers
- ▶ automatically identifies compiled programs with embedded revision numbers
- ▶ doesn't require changes to the way you edit/compile your files

This has been widely available since the 1980s, as both free and commercial software.

So use it!

(© 2007 Dartmouth College, Source)

Revision Control: History

Some of the players

- ▶ SCCS (Source Code Control System) - AT&T proprietary software, distributed with many commercial versions of Unix
- ▶ RCS (Revision Control System) - Walter Tichy; Purdue University (1985)
 - ▶ Like SCCS, but open source (stores differences more efficiently)
 - ▶ Still on every Unix/Linux system (in Xcode for OSX)
 - ▶ Very mature code
 - ▶ Strict locking to prevent conflicts
 - ▶ Great for individual files and small projects
- ▶ CVS (Concurrent Versions System)
 - ▶ Layered on top of RCS (originally)
 - ▶ Relaxes strict locking and uses a modify and merge model
 - ▶ Introduces a client-server model for distributed operation
- ▶ SVN (Subversion)
 - ▶ Complete rewrite of CVS, but sticking to the same goals
 - ▶ Tracks directories, not just files
 - ▶ Tracks arbitrary metadata
 - ▶ Repository is a real database
 - ▶ All access, local or remote, is through the client-server model

Revision Control: Vocab

Basic Setup

- ▶ Repository (repo): The database storing the files.
- ▶ Server: The computer storing the repo.
- ▶ Client: The computer connecting to the repo.
- ▶ Working Copy: Your local directory of files, where you make changes.
- ▶ Trunk/Main: The primary location for code in the repo.
Think of code as a family tree — the trunk is the main line.

(Source: betterexplained.com)

Revision Control: Vocab

Basic Actions

- ▶ Add: Put a file into the repo for the first time, i.e. begin tracking it with Version Control.
- ▶ Revision: What version a file is on (v1, v2, v3, etc.).
- ▶ Head: The latest revision in the repo.
- ▶ Check out: Download a file from the repo.
- ▶ Check in: Upload a file to the repository (if it has changed). The file gets a new revision number, and people can “check out” the latest one.
- ▶ Checkin Message: A short message describing what was changed.
- ▶ Changelog/History: A list of changes made to a file since it was created.
- ▶ Update/Sync: Synchronize your files with the latest from the repository. This lets you grab the latest revisions of all files.
- ▶ Revert: Throw away your local changes and reload the latest version from the repository.

(Source: betterexplained.com)

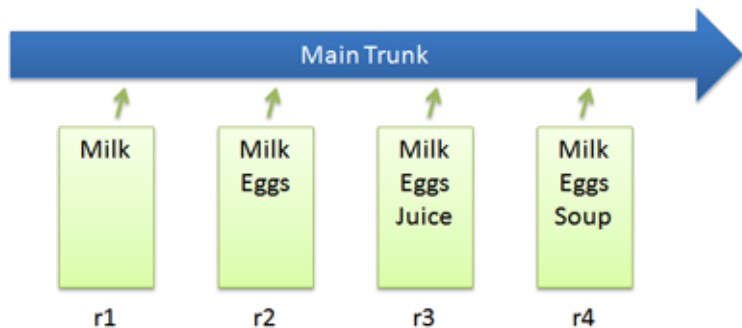
Revision Control: Vocab

Advanced Actions

- ▶ Branch: Create a separate copy of a file/folder for private use (bug fixing, testing, etc). (both a verb and a noun).
- ▶ Diff: Finding the differences between two files. Useful for seeing what changed between revisions.
- ▶ Merge (or patch): Apply the changes from one file to another, to bring it up-to-date.
- ▶ Conflict: When pending changes to a file contradict each other (both changes cannot be applied).
- ▶ Resolve: Fixing the changes that contradict each other and checking in the correct version.
- ▶ Locking: Taking control of a file so nobody else can edit it until you unlock it. Some version control systems use this to avoid conflicts.
- ▶ Breaking the lock: Forcibly unlocking a file so you can edit it.

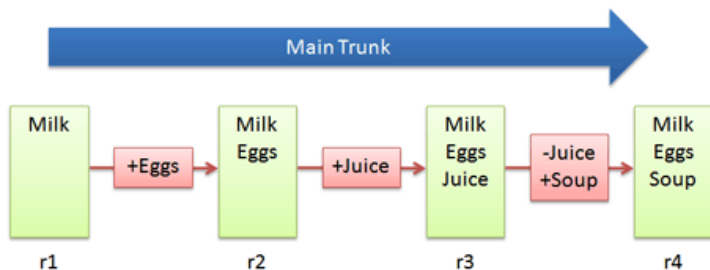
(Source: betterexplained.com)

Basic Checkins



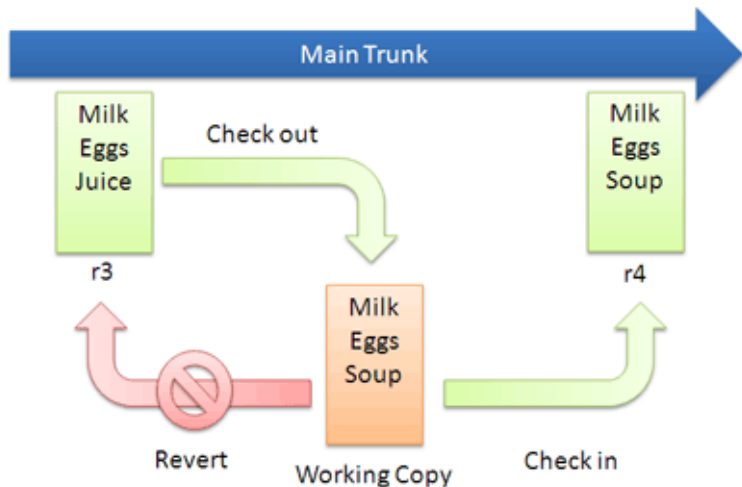
(Source: betterexplained.com)

Basic Diffs



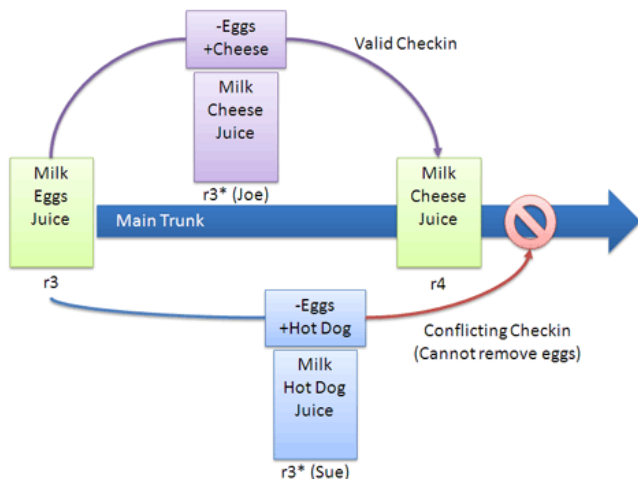
(Source: betterexplained.com)

Checkout and Edit



Revision Control

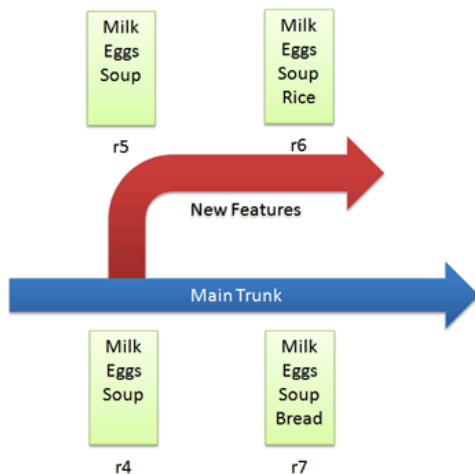
Conflicts



(Source: betterexplained.com)

Revision Control

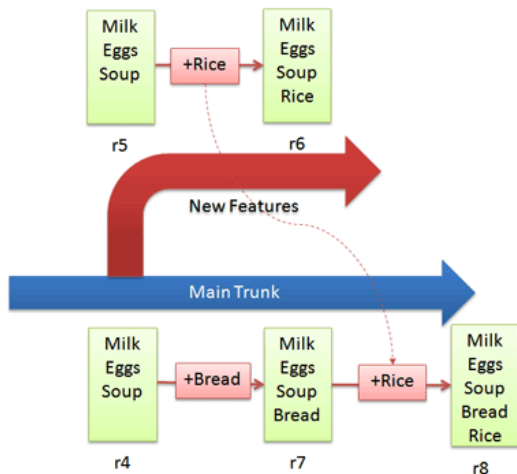
Branching



(Source: betterexplained.com)

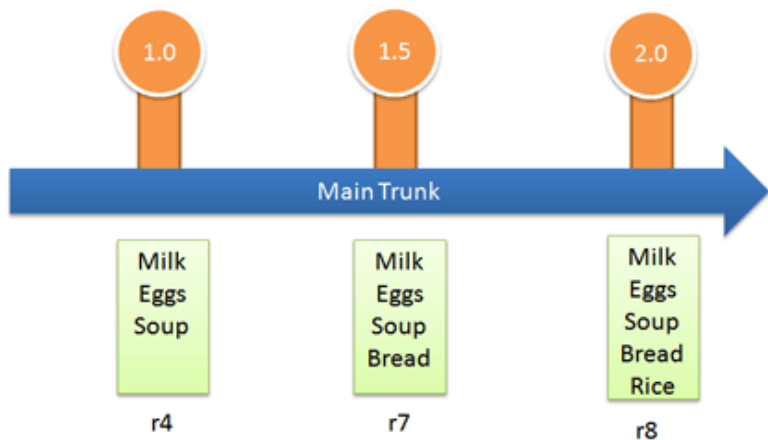
Revision Control

Merging



(Source: betterexplained.com)

Tagging



(Source: betterexplained.com)

SVN Setup Instructions

On the server (*i.e.*, ranger), in `~/public_html/`:

1. `svnadmin create newProj`
2. `chmod 700 newProj/`
3. `cd newProj/conf/`
4. `echo '[users]' > passwd`
5. `echo 'user1' = ToPsEcReT >> passwd`
6. `chmod go-rwx passwd`
7. `cp ~/cs/public_html/share/svnserve.conf .`
either the above command or the next four
8. `echo "[general]" > svnserve.conf`
9. `echo "anon-access = none" >> svnserve.conf`
10. `echo "auth-access = write" >> svnserve.conf`
11. `echo "password-db = passwd" >> svnserve.conf`

SVN New Repository Instructions

On YOUR machine, (e.g., a laptop):

1. `mkdir mortgageCalculator`
2. `mkdir mortgageCalculator/trunk/`
3. `emacs mortgageCalculator/trunk/assignment5.f90`
4. `emacs mortgageCalculator/trunk/Makefile`
5. `svn import --username user1 mortgageCalculator/ \
svn://svn.cs.mtsu.edu/$USER/public_html/newProj \
-m "Initial import"`

Authentication realm: <svn://svn.cs.mtsu.edu:3690> 22d56e8c

Password for 'user1':

```
Adding      mortgageCalculator/trunk
Adding      mortgageCalculator/trunk/assignment5.f90
Adding      mortgageCalculator/trunk/Makefile
```


SVN Check Out Repository Instructions

On the YOUR machine, (e.g., a laptop):

1. `svn checkout --username user1
svn://svn.cs.mtsu.edu/$USER/public_html/newProj`

A newProj/trunk

A newProj/trunk/assignment5.f90

A newProj/trunk/Makefile

Checked out revision 1.