Subversion (SVN)

Brian Magill

September 9, 2009

Projects Using SVN:

- Apache Software Foundation
- GCC
- Python
- Ruby
- PHP
- GNOME
- KDE

Evolutionary Progression:

RCS
Versioning Files
Rudimentary
Tagging

➡

CVS
Versioning Files
Some Tagging and
Branching

➡

SVN
Version Entire
Repository

Features

- Like CVS, uses a client/server model
- Versions repository as a whole
- Atomic Commits
- Renamed files retain full revision history
- Directories and symbolic links are versioned
- Branching and Tagging are cheap operations
- Automatically versions binary files
- Status and update are separate commands
- Current development paths can be automatically viewed via an Apache web server (if it is set up).

Limitations

- No "Obliterate" command
- Refactoring Merge Issues

Creating a New Repository

➢**cd** *<parent directory>*
➢**svnadmin create** *<repository name>*
➢**svn mkdir -m** *<comment> <URL name>*

Example:

$: cd /home/Smith/
$: svnadmin create svnrepo
$: svn mkdir -m "Creating directory for CERES_misc project" \
                    file:///home/Smith/svnrepo/CERES_misc/
$: svn mkdir -m "Creating trunk directory for CERES_misc project" \
                    file:///home/Smith/svnrepo/CERES_misc/trunk/
$: svn mkdir -m "Creating tags for CERES_misc project" \
                    file:///home/Smith/svnrepo/CERES_misc/tags/

Repository Notes

- svnadmin creates the SVN repository
- svn mkdir creates all SVN directories in a repository. (These entities can only be accessed via SVN commands.)
- Can have multiple repositories on a disk
- A change to any directory or file in a repository increments its revision number.

SVN URLs

- Can be either http:// or file:/// (also an SVN protocol available)
- Correspond to directory file structure down to repository level
- Need to SVN tools or (if set up) Apache web server to view structure for projects and branches
- Not an issue for http checkouts or any commit operations

Viewing SVN Structures

- **svn list** *<URL>* - Lists files or SVN paths under the given URL

- **svnlook tree** *<actual repository path>* - Lists all directories and files in repository in a tree structure

- **svnlook –h** - Lists other commands associated with svnlook. (Essentailly svnlook is for SVN system administrators.)

Initial Import and Checkout

svn import *<directory of files for CM>   <SVN URL>* **-m** "initial import"
svn co *<SVN URL> <new configured directory (working copy)>*

For Example:

svn import Unconfigured/   http://www.temp-site.org/CERES_misc/trunk \
             -m "initial import"

svn co http://www.temp-site.org/CERES_misc/trunk   Configured/

SVN Directory Commands

- **svn status** - Shows changes to local (working) copy
- **svn update** – Updates local copy with any newer files in repository
- **svn diff** *<file name>* - Displays differences between a file and the configured version
- **svn add** *<list of filenames>* - Schedules files to be added to SVN repository
- **svn delete** *<list of filenames>* - Schedules files to be removed from SVN repository
- **svn commit –m** *<comment>* - Commits changes to SVN repository

Branches and Tags

Same command:

svn copy *<source URL>* *<destination URL>* **-m** *<message>*

Example:

svn copy http://www.temp-site.org/CERS_misc/trunk/ \
       http://www.temp-site.org/CERES_misc/tags/milestone1 \
          -m ”first milestone”

svn copy http://www.temp-site.org/CERES_misc/trunk \
      http://www.temp-site.org/CERES_misc/branches/versionA \
       -m “first working version of project”

Summary

- Open Source
- Client/Server Model
- Mature Technology
- Substantial Improvements over CVS

References

- Official web site, http://subversion.tigris.org
- "Version Control with Subversion," Ben Collins-Sussman et al., http://svnbook.red-bean.com/
- "Dispelling Subversion FUD," Ben Collins-Sussman, http://www.red-bean.com/sussman/svn-anti-fud.html
- Subversion Project's Repository, http://svn.collab.net/repos/svn
- PHP Project's Repository, http://svn.php.net/repository/