
SUBVERSION - A VERSION CONTROL SYSTEM

Vincent Ho

vince at fluffyspider.com.au

March 2004

AREN'T WE ALREADY USING CVS?

CVS is:

- open source
- relatively well understood
- ubiquitous - used by GNOME^a, KDE^b, SourceForge^c etc.

^awww.gnome.org

^bwww.kde.org

^cwww.sourceforge.net

Problems with CVS:

Unfortunately, CVS also has a number of drawbacks. These include:

- tracks changes only on a per-file basis
- can't handle directories, or file metadata
- no renaming or moving of files possible
- lack of atomic commits
- tagging and branching are very slow operations
- poor handling of branch merging

SO WHAT'S SUBVERSION?

Enter our hero:

Subversion is a new system designed to fix the most critical problems of CVS but stay similar from a user's point of view.

`http://subversion.tigris.org`

Design:

- store repository in Berkeley DB
- basic idea is tracking *trees*, not files
- revisions are per tree, not per file

Eventually the stable 1.0 released was announced. As of this date, the latest release is 1.0.1.

Features:

- atomic commits
- able to copy/rename
- tags and branches are fast and cheap

Architecture:

→ server is Apache2 with a WebDAV SVN module

This provides a reliable server with access control, and “free” support for viewing the repository in a web browser. But Apache is also complex and perhaps a bit heavyweight. A lot of people think that using Apache as a Subversion server sucks.

But it's not all bad...

→ there's also `svnserve`, a custom server

You can run this:

→ from `inetd`, or

→ tunneled over `ssh`

What about the client?

→ client uses a library with well-defined API, `libsvn`

This allows clients to be written in other languages, including:

- python2.3-subversion (Python binding)
- libsvn-core-perl (Perl binding)
- RapidSVN - a graphical Subversion client

The cvs2svn converter script is written using the Python binding.

But enough info! Tell me how to use it!

CHECKING OUT FROM A REPOSITORY

Subversion is meant to be similar to CVS. So most things will be the same except for the CVSROOT being replaced by the repository URL:

```
$ svn co svn+ssh://manta/trunk/hpm/src
A  src/lowplugin
A  src/lowplugin/client.c
.
.
A  src/doxygen
A  src/doxygen/Makefile
Checked out revision 2487.
```

COMMITTING CHANGES

So you've made your changes, now what? Committing is just like CVS:

```
$ svn commit -m ' "My First SVN Commit", by Vince' Makefile
Sending          Makefile
Transmitting file data .
Committed revision 2488.
```

REVISION NUMBERS

Let's take a look at the current revision numbers of a couple of files:

```
$ svn stat -v Rules.make Makefile
      2487      2126 akumria      Rules.make
      2488      2488 vince      Makefile
```

```
$ svn update Rules.make
```

At revision 2488.

```
$ svn stat -v Rules.make
      2488      2126 akumria      Rules.make
```

Rules.make is now at revision 2488 (r2488), but hasn't changed!

COPYING

Now let's try a copy, in the project 'test', which I've checked out as well, even though you didn't see me do it.

```
$ svn cp kim.c kashi.c
```

```
A          kashi.c
```

```
$ svn commit -m "kashi == kim2" kashi.c
```

```
Adding          kashi.c
```

```
Committed revision 2490.
```

We can also copy from one URL to another, such as from
`svn+ssh://manta/trunk/test/kim.c` to
`svn+ssh://manta/trunk/test/kashi.c`.

This applies instantly and would save us doing the commit.
We don't even need a checkout to do this.

In all cases, copies retain full history.

MOVING/RENAMING

We can also move files at last (hurrah!).

```
$ svn mv kashi.c kim2.c
```

```
A      kim2.c
```

```
D      kashi.c
```

```
$ svn ci -m "renamed kashi to kim2" kim2.c
```

```
Adding      kim2.c
```

```
Committed revision 2491.
```

Once again, moved files retain history.

TAGGING AND BRANCHING

There is no tagging or branching command in Subversion..
it's all done via copying!

```
$ svn cp svn+ssh://manta/trunk/test \
      svn+ssh://manta/tags/test-0.1
Committed revision 2492.
```

Copies are cheap in both space and time, so tags are too.

OTHER STUFF

There is also the *merge* command, which is similar to doing a `cvs checkout -j REV1 -j REV2`. This is designed to make it easier to merge changes from branches.

SO WHO'S USING SUBVERSION?

- Subversion itself, naturally
- Debian maintainers for XFree86 and GNOME
- Xiph.org Foundation (for Ogg, Vorbis and Theora)
- linux1394 (Firewire) project
- libCACA

SUMMING UP

- Subversion (svn) is a new revision control system
- designed to be a CVS “work-alike”
- it tracks *trees*, not files
- has atomic commits, native copying and moving
- fast and cheap branching and tagging

QUESTIONS, BOUQUETS, BRICKBATS?

This space intentionally left blank.