Subversion: Powerful New Toys

Justin R. Erenkrantz

Joost

http://www.erenkrantz.com/oscon/

justin@erenkrantz.com

Why should I pay attention?

- *Subversion committer since 2001*
- *Committer to Apache HTTP Server & APR*
- *Helped lead apache.org SVN deployment*
- *President, The Apache Software Foundation*
- *Completing Ph.D. at UC Irvine*
- *Senior Software Engineer, Joost*

“Praise” for Subversion

Linus Torvalds: “I see Subversion as being the most pointless project ever started - because the whole slogan for Subversion for a while was 'CVS done right.' If you start with that kind of slogan, there is nowhere you can go.”

Aim: Explore the features added since 1.0 and discuss what is in store for the future.

Subversion

The goal of the Subversion project is to build a version control system that is a compelling replacement for CVS in the open source community.

Since 1.0: Be faster and more applicable to a wider range of situations than CVS could possibly have been.

Subversion Chronology

- *Founded by CollabNet in April 2000*
- *First milestone in October 2000*
- *Subversion 1.0.0 on Feb 23, 2004*
- *Subversion Corp. founded in April 2006*
- *Subversion 1.4.4 on June 8, 2007*
- *Subversion 1.5 ‘soon’*

Subversion Key Features

- *A centralized repository isn’t always wrong*
- *Cheap copies with branches and tags*
- *Supports renames (but not atomic...)*
- *Supports WebDAV protocol*
- *Stable, well-documented C interfaces*
- *Platform/language support & 3rd party apps*

New repository backend: FSFS

- *When: Introduced in 1.1; default in 1.2*
- *How:*
  - *Server: svnadmin create --fs-type fsfs <path>*
  - *Client: Not available - transparent to the end-user*
- *Why: Original Berkeley DB database repository format did not scale*

FSFS Repository Format

- *Berkeley DB fell over repeatedly on apache.org installation - just could not handle scale gracefully*
    - *Deadlocks, infinite loops, corruption*
- *FSFS: flat-file format on top of basic portable FS semantics*

Corruption bug w/revprops fixed in 1.4.4

Optional Locking

- *When: Introduced in 1.2*
- *How:*
  - *Server: svnadmin lslocks/rmlocks*
  - *Client: svn lock / svn unlock*
- *Why: Allow people to indicate when they are working on a file that will not merge well (i.e. binary file)*

Optional locking: Details

- *Useful for Word docs, images, etc.*
- *Proper terminology is "advisory" lock*
    - *Locks are breakable via --force*
    - *svn lock --force / svn unlock --force*
- *svn:needs-lock property on a file requires lock before commit (read-only w/o lock!)*

WebDAV Auto-versioning

- *When: Introduced in 1.2*
- *How:*
  - *Server: SVNAutoversioning On in httpd.conf*
  - *Client: Not accessible via normal SVN client*
- *Why: Allows DeltaV clients (such as Mac OS X Finder, Linux WebDAV FS, etc.) to commit to the repository*

Auto-versioning Caveat

- *While versions are technically preserved with DeltaV auto-versioning, the automatic versions often look ‘weird’*
  - *Often ‘delete’ and ‘add’ the same file*
  - *Plays havoc when looking at diffs*
  - *“Feature” of the WebDAV clients - not of Subversion*

Path-based authorization

- *When: Intro in 0.24.0 (DAV) / 1.3 (svnserve)*
- *How:*
  - *Server:*
    - *WebDAV: AuthzSVNAccessFile <authz> in httpd conf*
    - *ra_svn: authz-db = authzfile in svnserve.conf*
  - *Client: Not available.*
- *Why: Allow administrators to partition access to the repository*

Example authorization file

```
[groups]
admins=alice,bob
users=carol
[/]
*=r
@admins=rw
[/path]
@users=rw
[/carol]
*=
carol=rw
```

Authorization Performance

- *Authorization can be quite slow*
- *For log (or recursive) op., descend into each subdirectory to ensure access - safe!*
- *Overhead is in httpd - not Subversion*
- *Prior to SVN 1.5: SVNPathAuthz off*
  - *Danger! Unsafe if mixed authorization!*
- *1.5 (safer): SVNPathAuthz short-circuit*

Operational Logging

- *When: Introduced in 1.3 (WebDAV only)*
- *How:*
  - *Server: Add to httpd.conf:*

    ```
    CustomLog logs/svn_log "%t %u %{SVN-ACTION}e" env=SVN-ACTION
    ```
  - *Client: not available.*
- *Why: Allow server admins to track commits, updates, checkouts, etc.*

Improved language bindings

- *When: Introduced in 1.3*
- *How:*
  - *Server: Not available.*
  - *Client: New Python, Ruby, Perl SWIG bindings*
- *Why: No build requirement of SWIG; more language-friendly interfaces*

Working copy improvements

- *When: Introduced in 1.4*
- *How:*
  - *Server: Not available.*
  - *Client: Automatically updates on ‘svn update’/etc.*
- *Why: Major performance penalties with XML backing store; now use flat-file.*

New working copy format

```
<entry
   committed-rev="25830"
   name=""
   committed-date="2007-07-24T17:25:05.512383Z"
   url="http://svn.collab.net/repos/svn/trunk/notes"
   last-author="cmpilato"
   kind="dir"
   uuid="65390229-12b7-0310-b90b-f21a5aa7ec8e"
   repos="http://svn.collab.net/repos/svn"
   revision="25834"/>
```

12091 bytes

```
dir
25834
http://svn.collab.net/repos/svn/trunk/notes
http://svn.collab.net/repos/svn



2007-07-24T17:25:05.512383Z
25830
cmpilato
```

6296 bytes

Repository replication

- *When: Introduced in 1.4*
- *How:*
  - *Master server: Must be running SVN 1.4+*
  - *Slave server: svnsync init/synsync sync*
- *Why: Allow third parties to easily mirror or replicate an entire repository*

New binary diff algorithm

- *When: Introduced in 1.4*
- *How:*
  - *Server: Dump/load cycle required (not transparent)*
  - *Client: Not available.*
- *Why: Substantial space savings (50% smaller in many cases) and faster ops*

ra_serf

- *When: Introduced in 1.4 (optional)*
- *How:*
  - *Server: Not available.*
  - *Client: Transparent to end-user when enabled.*
- *Why: ra_neon requires custom uncachable HTTP requests; ra_serf uses basic HTTP requests*

Serf features

- *ra_neon used to do GET for every file*
  - *Simpler, allows for proxy caching*
  - *neon doesn't support pipelining - slow!*
  - *ra_neon switched to custom HTTP reports*
- *serf restores this by using HTTP pipelining*
- *serf also uses four concurrent connections*

The near future: 1.5

1.5 has been discussed for almost a year. However, we've decided to wait for true merge-tracking to be completed before 1.5 is released. Here is an overview of some of the already-completed features and finally a look at merge tracking...

WebDAV transparent mirroring

- *When: 1.5 (complete; dav-mirror branch)*
- *How:*
  - *Master Server: Have a post-commit hook to distribute updates to slaves via rsync/svnsync*
  - *Slave servers: SVNMasterURI http://master.example.com/repos/master in httpd.conf*
  - *Client: Check out from the slave server*
- *Why: Geographically distributed mirrors*

Incomplete checkouts

- *When: 1.5 (in trunk; almost finished)*
- *How:*
  - *Server: Server should be upgraded for best performance*
  - *Client: svn checkout --depth files*
- *Why: True non-recursive checkouts - partial checkouts of a particular directory*

Interactive conflict resolution

- *When: 1.5 (command-line client)*
- *How:*
  - *Server: Not available.*
  - *Client: svn update / merge - when a conflict arises*
- *Why: Conflicts weren't resolved right away; can bring up editor and resolve immediately before update completes*

Merge tracking

Linus Torvalds: "Merging in Subversion is a complete disaster. They have a plan and their plan sucks too. It is incredible how stupid these people are."

Yes, merge tracking in prior releases was non-existent. Sorry. Now, as to the plan...

What is merge tracking?

- *A ‘stable’ branch and an ‘unstable’ trunk*
- *Make changes first in trunk*
- *Then, after testing, want to merge to stable*
- *One-off merges are fine...but what about subsequent merges? Oops...*
- *Subversion wouldn’t record the merges...*
- *Merge tracking is all about remembering*

Merge Tracking: svnmerge.py

- *When: Introduced in 1.3*
- *How:*
  - *Server: Not available.*
  - *Client: svnmerge.py avail/merge/block*
- *Why: Similar to how SVK does merge tracking - client-side merge tracking*

Merge Tracking: The Future

- *When: Currently in 1.5 (in-development)*
- *How:*
  - *Server: Server requires upgrade (needs SQLite)*
  - *Client: svn merge*
- *Why: Integrated merge tracking solution into the core Subversion libraries*

Some svnmerge.py features may not make it into 1.5

Thoughts for the distant future

- *No one knows, but here's a few ideas...*

Fitz’s list

- *Faster. Subversion does need to be faster for many ops.*
- *Offline commits.*
- *Local branches.*
- *Karl Fogel adds: “Better merging”*
  - *(Duh.)*

Justin’s Wild Guesses...

- *Subdirectory detachability*
- *New repository formats (cf. Hg revlog)*
- *Clients migrate towards “thin” Python apps*
- *Atomic renames (multiple attempts so far!)*
- *Distributed repositories (cross-repos copies)*

Subversion: Powerful New Toys

Justin R. Erenkrantz
Joost
http://www.erenkrantz.com/oscon/
justin@erenkrantz.com