What's New in Subversion 1.9?

About Our Presenters

C. Michael Pilato

Mike has been an open source software developer for over 14 years, and has been one of the core designers and developers of Apache™ Subversion® (http://subversion.apache.org/). Additionally, he is the primary maintainer of ViewVC (http://viewvc.org/) and co-author of *Version Control With Subversion*.

Within CollabNet, Mike is a member of the Subversion, TeamForge and CloudForge teams, where he focuses on the integration of Subversion and related software into CollabNet's other primary product portfolio offerings.

Bob Jenkins

With over 28 years in providing software solutions to a varied set of markets and product offerings, Bob has an abundance of first-hand experience with version control systems like Subversion.

At CollabNet, he is the Director of Version Control Systems and defines and executes both consulting and training services for customers moving to and using either Subversion or Git.

Agenda

- Client Enhancements
  - Improvements to Working Copy Operation
  - Improved Conflict Resolution
  - New Command Options
  - New Commands
- Server Enhancements
  - Modified Administration Commands and Utilities
  - New Administration Command and Utility
- Experimental Enhancement
  - FSX Backend
- Migration to 1.9
  - Client and Server Migration
  - Why Should You Consider Moving/Upgrading to Subversion 1.9?

Client Enhancements

Improvements to Working Copy Operation

- Local tracking was added in 1.8 for the copy and delete pair of operations related to a move/rename, but 1.9 addresses corner cases and avoids unnecessary tree conflicts.

- Working copy library has been improved to increase performance and stability.

Improved Conflict Resolution

- File conflict markers now show the base version in addition to the working copy and repository versions.

```
Unconflicted lines
<<<<<<< .mine
"Hello hello, world!“
||||||| .r1
"Hello, world!“
=======
"Hello, ${name}!“
>>>>>>> .r2
Unconflicted lines
```

- GUI clients will not be directly impacted.

New Blame Option

Blame now can show not only when the last change to each line in a file was, but also when the next change will be. This functionality is invoked with the revision range is reversed.

```
$ svn blame –r HEAD:3 README.txt
```

Changes to Command Options

- New global option: --trust-server-cert-failures
  - Arguments: unknown-ca, cn-mismatch, expired, not-yet-valid, other
- Info command: --show-item and –no-newline options
- Propget command: --no-newline option
- Cleanup command: --remove-unversioned, –remove-ignored and --include-externals options
- Copy command: --pin-externals option

New Command

Auth – view cached authentication credentials
showing related information for all such
credentials or using patterns, those that match
the patterns

--remove option results in removing the matching authentication credentials

```
$ svn auth
------------------------------------------------------------------------
Credential kind: svn.simple
Authentication realm: <https://ctf.open.collab.net:443> Authorization Realm
Password cache: wincrypt
Password: [not shown]
Username: jbond
```

Server Enhancements

Modified Administration Commands and Utilities

- The svnadmin verify command now has a *--keep-going* option to instruct svnadmin to continue with the next revision so that multiple issues can be uncovered in a single execution rather than just reporting on the first one found.

- The svnadmin verify command also has a *--metadata-only* option to skip the expensive internal consistency checks and reconstruction of all user data.

```
$ svnadmin pack –M 1000 /svn/my-repos
```

New Administration Command

The svnadmin utility now supports an info command to report detailed format information about a specific repository.

```
$ svnadmin info /svn/my-repos
Path: /svn/my-repos
UUID: ac336b0e-000b-11e0-b354-23d019ddd9ed
Repository Format: 5
Compatible With Version: 1.8.0
Repository Capability: mergeinfo
Filesystem Type: fsfs
Filesystem Format: 6
FSFS Sharded: yes
FSFS Shard Size: 1000
FSFS Shards Packed: 1631/1631
FSFS Logical Addressing: no
Configuration File: /svn/my-repos/db/fsfs.conf
```

Experimental Enhancement

EXPERIMENTAL Backend (DO NOT USE FOR PRODUCTION)

- FSX is intended to ultimately overcome limitations in the current FSFS backend.
- Here some of the intended (not yet fully accomplished) improvements:
  - 90% reduction in metadata overhead.
  - Efficient handling of very large files.
  - Higher overall compression rates, in particular for office-style documents.
  - Information required for log and merge operations is more readily available.
  - O(1) handling of large directories.
  - Versioned revision properties.
  - Partitionable storage.
  - Arbitrary meta data storage and indexing facilities.

Mad Scientist

Migration to 1.9

Client Migration

- The working copy revision has not changed so no upgrade of existing working copies is required if you are already at 1.8.x (if a working copy is at an older release, then an upgrade is required).

- There are no new server version dependencies.
  - With 1.9 the general rule that 1.*x* clients always work against 1.*y* servers, regardless of *x* and *y*, holds true.

Server Migration

- There is no requirement for a dump and load of existing repositories.

- In order to take advantage of FSFS backend enhancements, the format associated with existing repositories must be upgraded.
  - See the table on the next slide

Repository Service Quality

Feature	Format 6 (SVN 1.8) or older	Upgraded to format 7 from older formats	Created as format 7 not packed	Created as format 7 packed
Reduction in dynamic memory usage[1]	yes	yes	yes	yes
Saturate 10Gb networks from SVN caches[2]	yes	yes	yes	yes
Saturate 1Gb networks from OS caches[3]	yes	yes	yes	yes
svnadmin pack does not block commits	no	yes	yes	yes
Full checksum coverage of revision data[4]	no	no	yes	yes
Quick verification to find external corruption[5]	no	no	yes	yes
Fast access to cold data on disk[6]	no	no	no	yes

[1] Where feasible, temporary buffers have a fixed maximum size now. Other temporary containers have been reduced in memory consumption.
[2] If almost all requests, i.e. well over 90%, can be served from SVN fulltext caches etc., an 8-core server running Apache can saturate a 10Gb network with uncompressed data. It will take 20+ concurrent checkout or export requests to generate that load.
[3] If virtually all requests (>95%), can be served from the OS file cache, a 4-core server running Apache can saturate a 1Gb network with uncompressed data. It will take 2 or more concurrent checkout or export requests to generate that load.
[4] Not only user file contents, directories and properties are protected by checksums but also the meta-data tying them together. This only detects external corruption, caused by rogue scripts, hard disk failure etc. and will not help against internal corruption caused by faulty SVN logic.
[5] Verifies a repository at several 100MB/s and does not slow down with increasing number of revisions. This allows for a much faster health check after system failure.
[6] Core feature of format 7. Revision data is read about twice as fast as with older formats. Assuming reading data from disk being 10x slower than from OS caches and a mere 10% OS cache misses, this translates into 30% higher overall throughput with format 7 over previous formats.

Subversion 1.9 Compatibility

New Feature	Minimum Client1	Minimum Server	Minimum Repository	Notes
Prospective blame	1.9	1.8	any	
FSFS format 7	any	1.9	1.9	Older formats remain supported.
Lock HTTP pipelining	1.9	1.2	1.2	
Commit unlocking	1.2	1.9	1.2	
Locking multiple files	1.3	1.9	1.2	Over the svn:// protocol.
FSX	any	1.9	1.9	Will not be compatible with 1.10

[1]Reminder: when using the file:// repository access method, the Subversion program is both the client and the server.

Why should you consider moving/upgrading to Subversion 1.9?

- If you are at release prior to 1.8, then you are not supported.
- Better performance and stability are part of 1.9 client and server binaries.
- It is relatively painless and quick.
  - Upgrades don't require synchronized client and server versions.
  - Client upgrade doesn't require working copy upgrade if upgrading from 1.8.
  - Upgrades fundamentally only require binary replacement.
  - Repository upgrades are optional, but with svnsync, they can also be relatively quick and painless.

Thank you!