

# Subversion, WebDAV, and Apache HTTP Server 2.0

---

*Justin R. Erenkrantz*

*University of California, Irvine*

*justin@erenkrantz.com*

*Slides: <http://www.erenkrantz.com/oscon/>*

# What is Subversion?

---

- *Did you miss ‘Subversion: Version Control Rethought’?*
- <http://subversion.tigris.org/>
- *A compelling replacement to CVS*
- *0.25.0 was released this morning*
- *Expect a release every two-three weeks*

# What is Subversion?

---

- *Currently alpha*
  - *Extremely conservative approach*
  - *Subversion is self-hosting*
- *But, don't take our word for it*
  - *<http://subversion.tigris.org/svn-repositories.html>*
- *Expect 1.0 ‘soon-ish’ - beta in 2-3 months*

# What is WebDAV?

---

- *A mechanism for distributed authoring*
- *Specified in RFC 2518*
- *A lot of software now supports WebDAV*
- *Forgot about versioning in RFC 2518*
  - *DeltaV - RFC 3253*
  - *Subversion partially implements DeltaV*

# Overview

---

- *mod\_dav\_svn*
- *mod\_authz\_svn*
- *Autoversioning*
- *ra\_svn basics*
- *svnadmin basics*
- *Bindings and Hooks*

# Subversion + WebDAV

---

- “*Why do what others have done?*”
- *Portability used by using APR*
- *Subversion includes an httpd 2.x module*
  - *Built on top of mod\_dav*
  - *Can browse repository over HTTP*
    - *Need WebDAV client to make changes*
  - *Simply add mod\_ssl to use SSL*

# Subversion Remote Access

---

- *Ways to talk to a repository in Subversion*
- *ra\_local - Local file-based*
  - *file://*
- *ra\_svn - Custom protocol*
  - *svn:// or svn+ssh://*
- *ra\_dav - WebDAV-based*
  - *http:// or https://*

# mod\_dav\_svn

---

- *Apache module for Subversion*
- *Requires httpd-2.0 (or greater)*
- *Uses mod\_dav included with 2.0*
- *Use --with-apxs=/path/to/httpd/apxs when configuring Subversion*
- *Clients require Neon to use ra\_dav client*

# mod\_dav\_svn/ra\_dav Protocol

---

- *Uses a subset of WebDAV/DeltaV*
- *Each file is retrieved via GET*
- *Properties retrieved via PROPFIND*
- *Transactions done via MKACTIVITY*
- *Finally MERGE back*
- *See notes/webdav-general-summary*

# mod\_dav\_svn Example

---

```
# Must be after mod_dav.so if it exists as a DSO
LoadModule dav_svn_module      modules/mod_dav_svn.so

<Location /svn/repos>
  DAV svn
  SVNPath /absolute/path/to/repository
</Location>
```

If you want authentication and a group of repositories:

```
<Location /svn/repos>
  DAV svn
  SVNParentPath /absolute/path/to
  AuthType Basic
  AuthName "Subversion repository"
  AuthUserFile /my/svn/user/passwd/file
</Location>
```

# mod\_authz\_svn

---

- *Path-based authorization*
  - Allow user *Joe* to access */joe* only
  - Allow group *committers* write-access
  - Inherit permissions from parent
- *File-based configuration*
- *Other RA layers will not use this now!*

# mod\_authz\_svn Syntax

---

```
[groups]
<groupname> = <user>[,<user>...]
...
```

```
[<path in repository>]
@<group> = [rw|r]
<user> = [rw|r]
* = [rw|r]
```

**\* = Everyone**

**r = GET, PROPFIND, REPORT, OPTIONS**

**w = MKCOL, DELETE, PUT, PROPPATCH,  
CHECKOUT, MERGE, MKACTIVITY...**

**COPY and MOVE require at least w on destination**

# mod\_authz\_svn Example

---

Add to your Location block:

```
AuthzSVNAccessFile /path/to/access/file  
Require valid-user
```

/path/to/access/file contains:

```
[groups]  
branchonly=john,mary,alice  
writers=bob,jones,scotty  
[/]  
*=r  
@writers=rw  
[/branches]  
@branchonly=rw
```

# Autoversioning

---

- *Not all clients will be Subversion clients*
- *Allows WebDAV-only clients to modify*
  - *No need for custom clients!*
- *Generic log message used on commits*

```
<Location /svn/repos>
  DAV svn
  SVNPath /absolute/path/to/repository
  SVNAutoversioning on
</Location>
```

# Autoversioning Example

---

- *See subversion/notes/autoversion-compatibility.txt*

Add to your Location block:

```
SVNAutoversioning On
```

Tested Clients:

Nautilus 2.x

Windows WebFolders

Mac OS X Finder

Linux davfs2

Macromedia Dreamweaver

DAVExplorer

# mod\_dav\_lock

---

- Some WebDAV clients are ‘class 2’
  - MacOS X’s Finder
  - Require *LOCK* support from repository
- Subversion does not have this directly
  - Only *httpd 2.1* includes *mod\_dav\_lock*
  - Use *DavGenericLockDB* directive

# Full configuration example

---

```
<Location /svn/repos>
  DAV svn
  SVNPath /absolute/path/to/repository
  SVNAutoversioning on
  DAVGenericLockDB logs/repos-locks
  AuthType Basic
  AuthName "Subversion repository"
  AuthUserFile /my/svn/user/dir/passwd

  AuthzSVNAccessFile /my/svn/user/dir/access
  Require valid-user
</Location>
```

# ra\_svn

---

- *Not everyone trusts Apache!*
- *ra\_svn was introduced*
- *Allows for tunneling*
  - *May require local accounts*
- *svnservice daemon*
  - *Anonymous/read-only access*

# ra\_svn Tunnels

---

```
% svn checkout svn+ssh://hostname/usr/local/svn/repository
```

- *Opens a remote connection*
- *Executes svnserve -t on the other end*
- *Uses ‘svn+<tunnel>’*
  - *Configured in ~/.subversion/config*
  - *Default of svn+ssh exist*

# svnadmin

---

- Allows for management of repository
  - Recover
  - Backup/Restore
  - Archive
- Only works with local file access
  - No remote administration yet

# Subversion and Berkeley DB

---

- *Built on top of Berkeley DB 4.x*
- *All data is stored in Berkeley DB tables*
- *Ensures the integrity of the system (ACID)*
- *However, may require some maintaining*

# Recovering ‘corrupted’ database

---

- *You may see, “Need to run recovery”*
- *Chances are it is not really corrupt*
- *Run ‘svnadmin recover /path/to/repos’*
- *Ensure that there are no open processes*
  - *httpd, svnserv, etc.*

# Berkeley DB log files

---

- *Berkeley DB creates log files*
  - *Allows complete playback*
  - *Can grow very large*
- *Usually, not all log files are needed*
- *As of 0.25.0, ‘svnadmin archive’ cleans up*
- *‘db\_archive | xargs rm’ also works*

# Subversion dump format

---

- *Subversion has its own archive format*
- *Plain text format with MIME-like headers*
- *Berkeley DB is not really portable*
  - *If switch endianness, need to recreate*
- *Required on repository format updates*
  - *‘dump/load cycle’*

# svnadmin dump

---

- ‘*svnadmin dump /path/to/repos*’
- *Defaults to stdout*
- *Options*
  - -rX:Y - *specify ranges to dump*
  - --incremental
    - *Loading into created repository*

# svnadmin load

---

- ‘*svnadmin load /path/to/repos*’
- *Reads from stdin*
- *Repository must already be created*
- *UUIDs*
  - *For future distributed repositories*
  - *Use, ignore, or create on load*

# svndumpfilter

---

- Ability to tweak a dump file
- Reads dump from stdin
- New file on stdout
- Exclude or include paths in dump
- Renumber ‘missing’ revisions
  - Poor man’s ‘svn obliterate’

# Bindings

---

- *C and SWIG-centric bindings*
- *Subversion has lots of little libraries*
- *Utilize SWIG to target multiple languages*
  - *Python is the main supported target*
  - *Automatically in-sync with C API*
- *mailer.py and ViewCVS use SWIG binding*

# Hooks

---

- *Each repository has a hooks directory*
- *Scripts can be placed in there*
- *Can be called at various points*
  - *Server-side only*
- *commit: pre-, start-, post- hooks*
- *revprop changes: pre-, post-*

# ViewCVS and mailer.py

---

- *Current development snapshot of ViewCVS supports Subversion*
- *Allows viewing of revisions other than the latest one*
- *mailer.py*
  - *Post-commit hook that sends diff emails*

# Conclusion

---

- *Subversion plays well with WebDAV*
- *Path-based authorization available*
- *Autoversioning allows for non-Subversion clients to access the repository*
- *svnadmin is an essential tools for admins*

# The Final Word

---

- *Upcoming O'Reilly Book*
- *Subversion: The Definitive Guide*
- *See <http://svnbook.red-bean.com/>*
- *Included with Subversion source, too*
- *Real-time help: irc.freenode.net #svn*
- *Mailing list: dev@subversion.tigris.org*