SVN = Common Home Dir Goodness

branson matheson
edward.b.matheson@nasa.gov

Topics

- Common Home Directory Methodologies
- Common Home Dir Pros vs. Cons
- My Solution: SVN
- Planning
- Setup
- Usage
- Support
- Q&A

CHD Methodologies

- NFS/AFS/CIFS mounted /home

- Automounting using /ump (Sarah et al.)

- AFS/DFS

- Apple AFS + Autofsd

- SMB/CIFS + Windows

- Replication using rsync/rdist

- Initial Buildout using cfengine/puppet

Common Home Dir (CHD)
Pros vs Cons

- Unified environment
- Improved capabilities
- Simpler management
- Transparent config
- Files are everywhere
- Changes are instant-ish

- Cross-platform issues
- Local changes affect globally
- Size of dataset
- Latency (speed)
- Latency (replication)

My Solution: subversion

- Replicate only what you want
- Change control == Good™
- Simple and effective
- Secure
- Scales well
- Can use on an individual basis or site basis

My Solution: subversion

- Really became possible with subversion 1.6

- Integrates well with OSX, unix, linux and windows

- Can be used for LOTS of other things

    - Code Management

    - System Config Management (better than RCS!)

- Distributed backups!

    - Each user has several backups of their config at each place they check it out!

My Solution: subversion Pros

- ## Inplace edits!
  - – User doesn't stomp changes by checkout like RCS does.
  - – File perms and such are intact after checkin
- ## Changes are managed at dir level instead of file level
  - – Dependent files can all be edited at once and checked in together (inetd + services)

My Solution: Subversion Cons

- Updates are manual .. '`svn up`'

    – Mainly because of security

- Central server goes down, can't get updates

- New files have to be manually added

    – Tho you can add on a directory if you use '`svn propset`' to eliminate files you never want (/etc/passwd)

- Can't have multiple files based on ${var}

    – So can't have .zshenv for osx and .zshenv for linux

Planning
Personal Management

- Simple .. create a repo off a central machine

- Use a simple naming scheme cause you're gonna have to type it occasionally

- Example

```
svn+ssh://peter.larc.nasa.gov/~branson/repo/home
```

Planning
Site Management

- Plan what you might store
    - system config
    - source trees
    - documentation

- Plan **GOOD BACKUPS!!!** of the repo over short intervals. (I `svndump` to a usb fob)

- Plan storage type (Berkley DB vs. text)

- Plan consistency checking of the data

Planning
Site Management

- Plan Remote Access scheme

    - `svn://` - requires a service and configuration

    - `http://` - requires an http server with WebDAV

    - `svn+ssh://` - requires just a user account and normal unix filesystem controls

- Remember that authentication to the entire repo is by normal authentication controls (eg. `/etc/passwd`)

- Remember that access to different parts of the repo is configurable.. but an extra step.

Planning
Site Suggested Hierarchy

```
/repo - top level directory

/repo/site - system stuff

/repo/site/home/{UID} - user home

/repo/site/{domain}/{machine} - system cfg

/repo/src - programming source code

/repo/doc - documentation
```

Setup - Personal

```
# see if you have subversion installed and version

> which svn

> svn --version


# Create the Repository

> cd ~

> svnadmin create ~/repo


# Create the place to put home

> svn mkdir file:///Users/LA1936AJ/repo/home
```

Setup - Site

```
> cd /
```

Create the Repository

```
> sudo svnadmin create /repo
```

Or for Berkley DB

```
> sudo svnadmin create --fs-type=bdb /repo
```

Fix permissions

```
> sudo find /repo -type d -exec chmod 2775 {}\;
> sudo chgrp -R {users-group} /repo
```

Setup - Site

Create the dir hierarchy

```
# svn mkdir file:///repo/site

# svn mkdir file:///repo/site/home

# svn mkdir file:///repo/site/larc.nasa.gov

# svn mkdir file:///repo/site/doc

# svn mkdir file:///repo/site/src
```

- Note: if you have multiple sites, you can have separate home directories under each site.

Setup - Site

Setup backup (on a unix machine)

> sudo crontab -e

… add these lines …

Run a verify pass every night at midnight

0 0 * * * svnadmin verify -q /repo | mail -S 'svnverify' root@localhost

Run a hotcopy every hour to the backup

0 * * * * svnadmin hotcopy /repo /backup/repo

Usage – Home Dir

- First .. make a directory for your home dir on the repo

```
> svn mkdir {path_to_repo}/home/ematheso
```

- Next Check it out on top of your home

```
> cd ~
> svn co -f {path_to_repo}/home/ematheso .
```

Usage – Home Dir

Now add the files you care about

```
 > svn add .z??* .bash* .csh* .tcsh*
.forward .plan .profile .xinitrc .perltidyrc
```

- Once you checkout the repo.. it knows the path.

- These files will not be altered by default

  – If you want Id .. see svn help propset

- Make sure they have permissions you want when you add them, they should persist

Usage – Home Dir

Now add the directories you want .. you can do this non-recursively (thanks 1.6!)

```
> svn add -N .ssh
```

Or Recursively

```
> svn add bin bin/scripts
```

Now add the files you might care about from those directories

```
> svn add .ssh/authorized_keys .ssh/config
```

Note: do NOT add files that are

autoupdated like known_hosts!!!

Usage – Home Dir

Ok .. check everything in

```
> svn ci
Adding              .ssh/config
Adding              .ssh/id_rsa.pub
Adding              bin/
Adding              bin/scripts
Adding              bin/scripts/vi
Transmitting file data …
Committed revision 3.
```

Usage – Home Dir

On another host .. you merely have to:

```
> cd ~
> svn co --force \
    svn+ssh://{host}/{path_to_repo}/{UID} .
```

Use `--force` to overwrite otherwise it will fail if it finds an existing file like say .. .login

To Update your home dir... REALLY SIMPLE!

```
> svn up
```

Usage – System Directories

- Same as Home Dir .. just deeper in hierarchy
- Can apply config control to
    - /etc
    - /etc/sysconfig
    - /usr/local/etc
    - /opt/etc
    - /var/www
    - /Library/LaunchDaemons

Usage – System Directories

- Avoid any files that change automagically!!!
  - `/etc/(passwd|shadow|master.passwd)`
  - `/var/log/*`
  - `/tmp/*`
  - `/var/run/*`

- Avoid binaries that the system will update
  - `/(bin|sbin|libexec)`
  - `/usr/(bin|sbin|libexec)`
  - `/usr/local/(bin|sbin|libexec)`

Usage – Tips and tricks

- ## Use ssh_keys!!

  - Put `ssh-agent` in your default login script

  - Use `ssh-add` on your default

  - Enable ForwardAgent in your .ssh/config

- ## Use an svn-aware editor

  - I have a vi script that detects svn and asks if you wanna ci after editing a file under management... also does sudo ;)

  - I'll give it out

Usage – Tips and Tricks

- ## Make your .files OS aware

```
#-- per OS commands
os=`uname -s`
case $os in
  AIX)
    alias gps="ps -elf | grep -v grep | grep -i"
    ;;
  FreeBSD)
    alias gps="ps -auxwww | grep -v grep | grep -i"
    ;;
```

Usage – Tips and Tricks

- A Useful Alias

```
alias svnst='svn status -q'
```

- Make your login svn aware

```
# Put this in your .login
svnstatus=`svn status -q`
if [ ${svnstatus} ]
then
    echo "You have unsaved svn changes. "
    echo "Please 'svn co'"
    echo $svnstatus
fi
```

Usage – Tips and Tricks

- Want to check-out with out adding all those .svn directories?

  ```
  > svn export svn+ssh://{host}/{path}
  ```

  - This will not enable any tracking of files

  - This is good for things like websites!

  - This will not delete any files that exist unless you use `--force`

SVN Support Documentation

- The 'redbook' - http://svnbook.red-bean.com/

- Homepage - http://subversion.tigris.org/

- Find me on #unix @ irc.larc.nasa.gov

- Or Find me on #nohelp @irc.bsdnet.org

SVN Support
Other svn aware tools

- On OSX .. textmate and Xcode

- On Linux .. emacs and command-line

- On Windows .. TortiseSVN

- Eclipse + subclipse == editor goodness

- Over 500 Tools that work directly with svn

SVN = Common Home Dir Goodness

branson matheson
edward.b.matheson@nasa.gov
4-9054

Q&A