Subversion Best Practices

Brian W. Fitzpatrick
Software Engineer
Google

June 28th, 2006

Introduction

Who says these are

“Best Practices”?

Overview

- These are just my opinions
- This is not a tutorial!
- Not everything is Subversion-specific
- Feel free to ask questions at any time

Overview

- Server Best Practices
- Client Best Practices
- Common Use Cases

Server Best Practices

Which Server to Use?

- Depends on the situation
- Do you have to comply with your hosting service?
- `svnserve:` faster, lighter, good for simple setups
- `svn+ssh:` great if you already depend on sshd
- Apache HTTP Server:
  - More points of integration
  - Straight web browsing of repository
  - Ability to mount repository as network share
  - custom logging (coming soon for svnserve)

One Repository or Many?

- It should always be one repository...
  - Shared users
  - Shared code
  - Reduce maintenance burden
- ...except when it should be more than one
  - Radically different access policies
  - Radically different data types

Authorization Policy

- **None** if at all possible!
- Encourage a culture of trust
- Remember, you can’t really delete anything from Subversion

Repository Browsing Tools

- Should you set one up?
- Ones we like:
  - ViewVC
  - Trac

Hook Scripts

- Pre-commit hooks
  - Do not attempt to modify the transaction
  - Hooks we like:
    - `check-case-insensitive.py`
- Post-commit hooks
  - Run them in the background &
  - Hooks we like:
    - `mailer.py`
    - CIA bot

Locking/Reserved Checkouts

- Don’t lock everything all the time (like VSS)
- Only lock non-mergeable files
- Use `svn:needs-lock` property for communication
- Real life examples

Autoversioning

- Good for non-coding projects
- Good for non-techies
- Bad for traditional coding projects
  - no log messages
  - potential email spam
  - empty revisions
- Good candidate for separate repository!

Repository Maintenance

- Backup: dump vs. hotcopy
- History obliteration should be avoided
  - Takes a long time
  - Invalidates working copies
  - `svndumpfilter` has limitations
- If you must obliterate, try selectively dumping

Client Best Practices

Encourage Code Review

- Commit often
- Commit in small, discrete chunks
- Use consistent log messages
- Send commit emails to team

Branches

- Don’t fear them!
- Useful types of branches
  - short-lived task branch
  - medium-lived feature branch
  - long-lived release branch
- Have a release policy

Merge Tracking

- Needs to be managed by humans
- Describe merges in the log messages
- `svnmerge.py`
- Subversion 1.5: real merge tracking?

Standardize on One Locale

- All filenames and log messages stored as UTF-8
- Choose one locale and stick with it... or else

Use Autoprops

- No, the server can’t transmit them to clients
- Useful autoprops:
  - `svn:mime-type`
  - `svn:eol-style`
  - `svn:needs-lock`

Cool Client Tricks

- Switching to a branch in mid-flight
- In-place “import”

```
$ cd dataset/
$ svn mkdir URL
$ svn checkout URL .
$ svn add *
$ svn commit
```

Common Use-Cases

Mixing and Matching Components

- `svn:externals`
- ‘`svn switch`’ on empty directories

Managing a website in Subversion

- serve site from a working copy
- disable httpd access to `.svn/`
- write post-commit hook to update working copy

Use svnversion

- `$Revision$` doesn’t do what you think
- `svnversion` designed to work with your build system

Use a Template

- Dealing with a “mostly standardized” file:
  - Commit a template of file to repository
  - Have build system copy it to unversioned file
  - Users edit unversioned file

Q&A

Brian W. Fitzpatrick
fitz@google.com

http://subversion.tigris.org/