SUBVERSION

Tutorial

CUSEC 2005

Christopher Ness

McMaster University

nesscg@mcmaster.ca

Outline

- What is Revision Control & Subversion
- Advantages/Disadvantages of Subversion
- Subversion Concepts
- Two Developer Usage Demonstration
- Advanced Topics
- Questions

What is Revision Control?

- Revision Control (RC) can be thought of like a time machine which is constrained to travel only between the present and when the repository was created.
  - Every change you make is logged
  - Every file/directory can be viewed at any stage during it's development

Revision Control Model

- There are two main models for RC.
  - **Copy; Modify; Merge [CMM]**
  - Lock; Modify; Unlock [LMU]
- Subversion <= 1.1.x only supports CMM
- Developers are working on LMU for release in 1.2 according to the SVN website. http://subversion.tigris.org/project_status.html

What Is Subversion?

- http://subversion.tigris.org/ says:
  - "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. The software is released under an Apache/BSD-style open source license."
- Subversion is often shortened to SVN, which is the name of the command line client.

* Not a "drop-in replacement"

SVN Features/Advantages

- Actively Developed
- Atomic Commits
- Cheap copies and moves
- Supports renames
- Hook Scripts
- Portability (Apache Portable Runtime)
- Improved handling of different EOL styles
- Scriptable through bindings (C, Python, Perl, Java)
- Parseable output
- Similar UI to CVS

SVN Disadvantages

- Similar UI to CVS
- Requires more processing power than CVS
  - Diff files sent from Server <--> Client
  - Advantage if you are on a slow WAN
- In the past, was rejected because it did not scale to ∞ as well as CVS.
- Like C, SVN is very flexible and assumes users know best.

Who Uses Subversion?

- Subversion was self-hosting on 30 August, 2001. [r12635]
  - http://svn.collab.net/viewcvs/svn/
- Apache Software Foundation [r124662]
  - http://svn.apache.org/repos/asf/
- Others include: Samba, Mono, PuTTY, Debian, GNUe, Ethereal...

Working Copy vs. Repository

- The repository holds all the changes to the project that have been ***committed***.
- A working copy (wc) can hold many files which do not *have* to be in the repository.
  - Unversioned files
  - Mixed revisions
  - Uncommitted changes
- Unlimited number of workings copies

SVN Uses URLs – Not Modules

- CVS uses modules:

```
$ cvs -d \
  ":pserver:anonymous@anoncvs.gnome.org:/c
  vs/gnome" checkout module
```

- SVN uses URLs:

```
$ svn checkout \
  https://anoncvs.gnome.org/trunk/module/
```

- Once you have a working copy, URLs are rarely required any more.

Suggested Repository Layout

```
/              <- root {http[s],svn,file}
 trunk/        <- Main development line
  foo.c
 branches/     <- Does not effect main line
  devb/
   feature13/  <- Feature based on bug 13
   bugfix56/   <- Bug 56 fixing in progress
   bugfix19/
 tags/         <- Snapshots in time
  0.3/
  0.4/
  0.5/         <- Version 0.5 code as shipped
  0.5alpha/
  0.5beta/
  1.0/
```

Two Developer Demo

- I have always found it best to learn by diving in feet first.
- I will now use two different local users on my laptop to demonstrate some basic uses of Subversion.
- A log of the commands and repository will be available with the slides online.

svn:externals

- Allow you to construct a working copy made of many sub-working copies without having to check out each repository by hand.
- Once this property is set, all developers will automatically check out the externals as well.
- An example from the SVN book explains.

```
$ svn propget svn:externals calc
third-party/sounds             http://sounds.red-bean.com/repos
third-party/skins              http://skins.red-bean.com/repositories/skinproj
third-party/skins/toolkit -r21 http://svn.red-bean.com/repos/skin-maker
```

svn:externals Output

```
$ svn checkout http://svn.example.com/repos/calc
A  calc
A  calc/Makefile
A  calc/integer.c
A  calc/button.c
Checked out revision 148.

Fetching external item into calc/third-party/sounds
A  calc/third-party/sounds/ding.ogg
A  calc/third-party/sounds/dong.ogg
A  calc/third-party/sounds/clang.ogg
…
A  calc/third-party/sounds/bang.ogg
A  calc/third-party/sounds/twang.ogg
Checked out revision 14.

Fetching external item into calc/third-party/skins
…
```

Vendor Branches

- An external dependency, where you make local modifications or enhancements and wish to merge the vendors new releases with your modifications.
- You basically build a patch of the differences between 1.0 and 1.1 of the vendor's code and apply it against your modifications.
- It *can* be difficult, but is possible.

Questions?

- Slides, repository and log will be made available on the CUSEC website in a tarball.
- I'm usually lurking on the SVN users mailing list: users@subversion.tigris.org
- http://subversion.tigris.org for more info.
- http://svnbook.red-bean.com was referenced heavily in creating this presentation.