

Using subversion

COMP 2400

Prof. Chris GauthierDickey

What is subversion?

- A set of programs that help you keep track of changes to “stuff”
 - code
 - documents
- A replacement for cvs
- A set of tools for collaboration so that multiple people can work on the same “stuff”

svn Basics

- At the heart of svn is the repository
 - A centralized store of data, organized like a filesystem tree
- svn allows you to see previous changes to the repository
 - When was the last time the file contained something I'm looking for?

- **svn has two basic parts:**
 - **The server: hosts the repository--all changes go through the server**
 - **The client: an application we use to interact with the server**
 - **commit changes, get the recent version, search, etc**
- **svn keeps track of your local repository through 'hidden' files and directories: .svn**

The working copy

- While the server maintains the main repository, each user has one or more 'working copies'
 - These copies hold your own personal set of changes to the files
 - When you commit your changes, the server determines if any conflicts exist
 - You have to fix those conflicts in order to finally commit!

svn Syntax

- From the command line, we use the 'svn' command
 - It's an svn client that knows how to talk to a server
- The first command you should learn:
 - `svn help`
 - This will give you a list of commands you can get help on

svn import

- The 2nd command you must learn: **import**
 - **import** copies a set of directories or files to the repository to place them under version control
 - **svn import mydev <http://svn.cs.du.edu/courses/f2008/comp2400/myuser/mydev> -m 'initial import'**
- This will recursively copy everything to your personal space in the **comp2400** directory under **mydev**

How to start working

- Note that once you do an 'svn import,' you still do not have a working copy!
- You only import once!
 - Do not modify the stuff under mydev and try to import again
- You need to learn the 3rd svn command:
 - svn checkout

svn checkout

- **svn checkout will create a working copy of your repository on your machine**
 - **svn checkout <https://svn.cs.du.edu/courses/f2008/comp2400/myuser/mydev> .**
 - **This will create a working copy in your current directory**
 - **You work and edit from inside the working copy**
 - **Yes, the working copy is a set of directories**

SVN COMMIT

- After you've made your changes, you must use the 4th important command
 - `svn commit -m 'lil message about what changes were'`
 - This commits any changes from your working copy to your repository
 - You use `-m` to specify what you changed: it's entirely up to you what you say here, but something useful is better than nothing!

svn commit messages

- You'll notice when you commit that you get several letters on the left for each file that has been committed
 - **U**: the file was successfully updated
 - **G**: You had changes but they were merged successfully with the repository version
 - **C**: conflict! Uh oh! We'll get to that later...

How to keep working

- Now that you have a working copy:
 - **svn add myfile**
 - if myfile is a file, it will be added to the repository on the next commit
 - if myfile is a directory, it will add the directory recursively
 - **svn delete myfile**
 - If it's a file, it's deleted immediately from the working copy and will be deleted in the repository on the next commit
 - If it's a directory, it will be deleted on the next commit

- **svn copy existingfile newfile**
 - copies the existing file to the new file, will be added on commit
- **svn move oldfile newfile**
 - renames oldfile to newfile, will occur on commit
- **svn mkdir newdir**
 - creates newdir in the working copy and adds it on commit

Repository and Working Copy info

- After you've made changes, you can use 'svn status' to see the status of your working copy
 - You'll get three columns, the first a letter meaning:
 - A: item is scheduled for addition to the repository
 - C: item has conflicts with the latest version in the repository
 - D: item is scheduled to be deleted from the repository
 - M: item has local changes not in the repository

svn diff

- If you have an item that is in conflict or that's modified and you want to examine in details, use 'svn diff'
- **svn diff output is a bit odd at first: it only shows the differences between two files**
 - A line prefaced with '-' means that it's been removed
 - A line prefaced with '+' means that you've added it

svn revert

- Let's say that you realized you don't like the changes you made:
 - Use 'svn revert myfile'
 - This will overwrite your working copy with the latest version from the repository
 - Yes, you can just delete it and do 'svn update', but that's more typing!

svn conflicts

- What does svn do when it detects a conflict?
 - It puts 3 unversioned files in your directory:
 - filename.mine: your file as it existed in the working copy
 - filename.rOLDREV: the version in the repository when you checked it out
 - filename.rNEWREV: the version in the repository when you tried to check it in

No, you can't commit!

- **svn won't let you commit while the file is in conflict**
 - **You can make the changes by hand: edit the copy in vi**
 - **You'll see parts marked <<<<<< .mine, which are your changes, and >>>>>> .rXXX which are the latest revision in the repository**
 - **You get to decide what stays and goes!**
 - **You can 'svn revert' the file if you don't want your changes**

svn resolved

- Once you've resolved the conflicts, run 'svn resolved myfile'
 - This removes the temporary unversioned files
 - It informs your working directory that your local copy is now the corrected version
- Finally, 'svn commit' to upload your corrected copy to the server

Checking the svn History

- **svn log** will give you a log of all changes to your repository
- **svn diff** will show you the details of a change
- **svn cat** will output a particular file to your screen
- **svn list** will list the contents of a repository
- **svn info** will give you information about the repository, including its location and version #

Where to go from here

- <http://subversion.tigris.org>
 - The main subversion website--you can find clients for (almost) all OSes here
- <http://svnbook.red-bean.com>
 - The online manual for svn--many more details than presented here, if you get stuck, read this!

Getting your repository set up

- `htpasswd -n $USER`
 - `$USER` is interpreted by your shell to be your user name
 - You **MUST** have a Unix account set up already
 - `PASSWORD` is the password you will use for subversion
 - It's not the same as your unix password, it can be whatever
 - It's ***NOT*** `PASSWORD`, please don't make yourself look dumb by using that password ;-)

- Once you execute that command, it will produce an MD5 hash of your password, which is cryptographically secure
 - You email the returned value from that command to grothoff@cs.du.edu and tell him you are in COMP 2400
- He's going to create the initial repository for you which will live under:
 - <https://svn.cs.du.edu/courses/f2008/comp2400/<your user name>>

Exercises

- **Add an unrevised directory to your repository**
- **Add files and directories to it**
- **List it from the repository**
- **Delete something from your working copy the correct way (not using rm!)**