

April 17, 2007
S.D. Peckham

Notes on How to Use Subversion

Introduction

SVN is a version control program that uses a "copy-modify-merge" approach instead of a "lock-modify-unlock" approach, as explained in the Subversion User's Guide in Chapter 2 (p. 9 - 13). It can be used to track changes in a directory tree, either to individual files (called *file changes*), or to the tree itself (*tree changes*). The latter would include addition, deletion or renaming of files or subdirectories. Subversion can track changes for any collection of files, but is best-suited to tracking changes in text files, such as program source code or documentation. Subversion is a free, open-source program that is distributed under an Apache/BSD-style open-source license. (See <http://subversion.tigris.org>.) The actual title of the Subversion User's Guide is Version Control with Subversion: For Subversion 1.3.

This document is meant to be used as a very basic tutorial on how to perform common tasks with Subversion. Explicit examples are provided for a software project called TopoFlow.

SVN is set up on our server "mysticplum", so you need to log in there before you can execute the SVN subcommands discussed below.

Check Which Version of SVN is Installed

```
svn --version    (currently version 1.4 on mysticplum)
```

Get Help on Any SVN Subcommand

```
svn help <subcommand>
```

Create a New Repository

```
sudo chown -Rh nobody:nobody /usr/local/SVN/<name of repo>
```

All repositories must reside in /usr/local/SVN. Files in the directory /usr/local/SVN/conf need to be kept away from the users in general. One is the svn-auth-file that contains all of the usernames and the associated passwords of users that need access to the repository.

What about this method from the Subversion User's Guide?

```
svnadmin create /path/to/repos  
ls /path/to_repos
```

Copy Files Into a New Repository

Copy the contents of a directory called "TF_tree" under "TF_Directory" to create a new repository

```
svn import TF_tree file://TF_Directory -m "Initial import"
```

List Contents of a Repository

svn list <https://mp.colorado.edu/svn/TopoFlow> (for TopoFlow)
svn list <https://mp.colorado.edu/svn/Main> (for SedFlux)

(or just point a browser to the URL)

Check Out Files in a Repository

To check out the TopoFlow source code and copy it to "TF_Checkout" in current directory:

```
svn checkout https://mp.colorado.edu/svn/TopoFlow/trunk/code TF_Checkout
```

```
A TF_Checkout/version-1.5
A TF_Checkout/version-1.5/GUI_gw.pro
A TF_Checkout/version-1.5/COMPILE_TF
A TF_Checkout/version-1.5/LIB_filepath.pro
A TF_Checkout/version-1.5/GUI_precip.pro
A TF_Checkout/version-1.5/GUI_stop.pro
A TF_Checkout/version-1.5/LIB_path_sep.pro
A TF_Checkout/version-1.5/utlis_TF.pro
A TF_Checkout/version-1.5/GUI_snow.pro
A TF_Checkout/version-1.5/GUI_utlis_tf.pro
A TF_Checkout/version-1.5/smooth_DEM.pro
A TF_Checkout/version-1.5/route.pro
A TF_Checkout/version-1.5/richards.pro
A TF_Checkout/version-1.5/channel_grids.pro
A TF_Checkout/version-1.5/tests.pro
A TF_Checkout/version-1.5/methods.pro
A TF_Checkout/version-1.5/LIB_modifyct.pro
A TF_Checkout/version-1.5/Qnet_file.pro
A TF_Checkout/version-1.5/pixel_size.pro
A TF_Checkout/version-1.5/getvars.pro
A TF_Checkout/version-1.5/GUI_chan.pro
A TF_Checkout/version-1.5/plot.pro
A TF_Checkout/version-1.5/GUI_sed.pro
A TF_Checkout/version-1.5/postprocess.pro
A TF_Checkout/version-1.5/preprocess.pro
A TF_Checkout/version-1.5/diversions.pro
A TF_Checkout/version-1.5/init_depth.pro
A TF_Checkout/version-1.5/formulas.pro
A TF_Checkout/version-1.5/GUI_main.pro
A TF_Checkout/version-1.5/LIB_xmanager.pro
A TF_Checkout/version-1.5/cascade_rain.pro
A TF_Checkout/version-1.5/LIB_loadct.pro
A TF_Checkout/version-1.5/GUI_evap.pro
A TF_Checkout/version-1.5/GUI_infil.pro
A TF_Checkout/version-1.5/distrib_tool2.pro
Checked out revision 1.
```

Add a New File to a Repository

It is not enough to simply add a file to a checked-out directory. You must use the "add" subcommand to tell Subversion about the new file. Change to the working directory where the checked-out files have been saved (e.g. `cd ~peckhams/TF_Checkout`) and then type

```
svn add <new_file>
```

```
svn add save_load.pro    (example)
svn add LIB_strsplit.pro (example)
```

```
A    save_load.pro
A    LIB_strsplit.pro
```

Delete a File from a Repository

It is not enough to simply delete the file from a checked-out directory. You must use the "delete" subcommand to tell Subversion about the change. Change to the working directory where the checked-out files have been saved, and then type

```
svn delete <old_file>
svn delete distrib_tool.pro (?)
svn delete tests.pro      (?)
```

View/Save Changes to the Repository

Change to the working directory where the checked-out files have been saved, and then type

```
svn diff                    (view all of the changes, diff-style)
svn diff > Revision1.txt    (save changes to text file in diff-style)
```

Check the Status of the Repository

Single-character flags let you know the status of each file in the repository:
A=Added, D=Deleted, C=Conflicts, R=Replace (sheduled), L=Locked,
I=Ignore (set to not be managed by svn), ?=Not managed by svn,
M=Modified (locally)

Note that files in the repository that were not modified do not show up in the listing from "svn status" (e.g. GUI_sed.pro, init_depth.pro)

```
svn status
```

```
?  Revision1_diff.txt
M  GUI_gw.pro
M  COMPILE_TF
M  GUI_precip.pro
M  GUI_stop.pro
M  utils_TF.pro
M  GUI_snow.pro
M  GUI_utils_tf.pro
A  save_load.pro
M  smooth_DEM.pro
M  route.pro
M  methods.pro
M  Qnet_file.pro
M  getvars.pro
M  GUI_chan.pro
M  plot.pro
M  preprocess.pro
M  diversions.pro
M  formulas.pro
M  GUI_main.pro
```

```

M  GUI_evap.pro
M  GUI_infil.pro
A  LIB_strsplit.pro

```

To see when each file in a repository was last revised, you can set the verbose flag to see revision numbers. A repository usually contains "mixed revisions" as shown below, since not every file gets changed in a new revision.

```
svn status --verbose
```

```

?
?
Revision1_diff.txt
TF_Update_Apr0507.txt
1  1 peckhams .
2  2 peckhams GUI_gw.pro
2  2 peckhams COMPILE_TF
1  1 peckhams LIB_filepath.pro
2  2 peckhams GUI_precip.pro
2  2 peckhams GUI_stop.pro
1  1 peckhams LIB_path_sep.pro
2  2 peckhams utils_TF.pro
2  2 peckhams GUI_snow.pro
2  2 peckhams GUI_utils_tf.pro
2  2 peckhams save_load.pro
2  2 peckhams smooth_DEM.pro
2  2 peckhams route.pro
1  1 peckhams richards.pro
1  1 peckhams channel_grids.pro
1  1 peckhams tests.pro
2  2 peckhams methods.pro
1  1 peckhams LIB_modifyct.pro
2  2 peckhams Qnet_file.pro
2  2 peckhams getvars.pro
1  1 peckhams pixel_size.pro
2  2 peckhams GUI_chan.pro
2  2 peckhams plot.pro
1  1 peckhams GUI_sed.pro
1  1 peckhams postprocess.pro
2  2 peckhams diversions.pro
2  2 peckhams preprocess.pro
1  1 peckhams init_depth.pro
1  1 peckhams LIB_xmanager.pro
2  2 peckhams GUI_main.pro
2  2 peckhams formulas.pro
1  1 peckhams cascade_rain.pro
1  1 peckhams LIB_loadct.pro
2  2 peckhams GUI_evap.pro
2  2 peckhams GUI_infil.pro
2  2 peckhams LIB_strsplit.pro
1  1 peckhams distrib_tool2.pro

```

Commit Changes to a Repository

Change to the working directory where the checked-out files have been saved, and then type

```
svn commit one_file_only.pro      (example)
svn commit --message "This is the version of April 5, 2007."
svn commit --file Log_File.txt    (example)
svn commit --file TF_Update_Apr0507.txt
```

```
Sending      version-1.5/COMPILE_TF
Sending      version-1.5/GUI_chan.pro
Sending      version-1.5/GUI_evap.pro
Sending      version-1.5/GUI_gw.pro
Sending      version-1.5/GUI_infil.pro
Sending      version-1.5/GUI_main.pro
Sending      version-1.5/GUI_precip.pro
Sending      version-1.5/GUI_snow.pro
Sending      version-1.5/GUI_stop.pro
Sending      version-1.5/GUI_utils_tf.pro
Adding       version-1.5/LIB_strsplit.pro
Sending      version-1.5/Qnet_file.pro
Sending      version-1.5/diversions.pro
Sending      version-1.5/formulas.pro
Sending      version-1.5/getvars.pro
Sending      version-1.5/methods.pro
Sending      version-1.5/plot.pro
Sending      version-1.5/preprocess.pro
Sending      version-1.5/route.pro
Adding       version-1.5/save_load.pro
Sending      version-1.5/smooth_DEM.pro
Sending      version-1.5/utils_TF.pro
Transmitting file data .....
Committed revision 2.
```

Examine History of Changes to the Repository

Use the revision and verbose flags to examine changes that were made in particular revision. If a log file was specified via "svn commit --file Log_File.txt", then the entire log file is also printed (even if the verbose flag is not set).

```
svn log foo.c      (examine changes to a single file)
svn log --revision 0:3
svn log -r 0:3     (same as previous line)
svn log -r 2 -v    (verbose version, for 2nd revision only)
```

Update Your Working Copy of the Repository

This brings your working copy into sync with the latest revision in the repository. As explained in Chapter 2 (p. 18) of the Subversion User's Guide, Updates and Commits are separate. You can do either without doing the other.

```
svn update
```

Add a New User

```
cd /usr/local/SVN/conf
sudo usr/local/apache/bin/htdigest svn-auth-file Subversion <username>
```

It is not necessary for them to have an account on the server,

they only have to have a username and password set in the /usr/local/SVN/conf/svn-auth-file. Users should choose good passwords and to have a password that is different from the one(s) they use to access other systems here on campus.