University of Texas at El Paso
Computational Science (CPS)

A short Subversion (SVN) tutorial

1 Introduction

An important element of the modern software development process is **source control (or version control)**. Subversion is a software source code system for collaborative development. Cooperating developers commit their changes incrementally to a common source repository, which allows them to collaborate on code without resorting to crude file-sharing techniques (shared drives, email). It maintains a history of file and directory versions. The files and directories are checked out of the repository and into your local project work area. This is called **working directory**. Changes are made to files in your working directory. After changes are made to the create the next SVN working version, the files are **checked into** the subversion repository.

This document quickly explains how to install, configure, and use subversion locally. There is plenty of well written documentation on this subject, that far exceeds my knowledge of the tool. I wrote this document with the only idea of help you to started quickly and for those like me that just like to jump in with both feet into new things and realize get some knowledge first is always useful.

2 References

```
Subversion                              : http://subversion.apache.org/source-code.html
Download sourse code                    : http://subversion.apache.org/download/
Version Control with Subversion (book): http://svnbook.red-bean.com/
A short and more complete tutorial    : http://www.tutorialspoint.com/svn/
```

3 Basics SVN Commands

In Subversion (often abbreviated SVN), code is stored in a repository, which is located somewhere on the network.

3.1 Windows

Please check the follow web-page the procedure with the windows installation

```
http://svn.spears.at/
http://www.wikihow.com/Install-Subversion-on-Windows-7
```

3.2 Linux

- If you are using Debian based **GNU/Linux** then use apt command for installation.

```
henry@bluebottle:~$ sudo apt-get update

henry@bluebottle:~$ sudo apt-get install subversion
```

- **Which version of SVN**

```
henry@bluebottle:~$  svn --version
svn, version 1.7.5 (r1336830)
compiled Jun 21 2013, 22:11:49
```

- **Which and Where is SVN**

```
henry@bluebottle:~$ which svn
/usr/bin/svn

henry@bluebottle:~$  whereis svn
svn: /usr/bin/svn /usr/bin/X11/svn /usr/share/man/man1/svn.1.gz
```

4 Checking Out the Repository For The First Time

Each repository has a URL from which it can be initially checked out; once that's done, the **URL** is remembered for future operations. Here's how to check out the repository: **("co" is short for "checkout").**

4.1 Create your Workspace :

Let us start by create a folder for the semester `UTEP_FALL_2015` and class `CPS_5401`

```
henry@bluebottle:~$ cd /Desktop/
henry@bluebottle:~/Desktop/mkdir UTEP_Fall_2015
henry@bluebottle:~/Desktop/cd UTEP_Fall_2015
henry@bluebottle:~/Desktop/UTEP_Fall_2015$ mkdir CPS_5401
henry@bluebottle:~/Desktop/UTEP_Fall_2015$ cd CPS_5401
henry@bluebottle:~/Desktop/UTEP_Fall_2015/CPS_5401/$
```

Let me cut the prompt so I do not need to write the long extent show above. Instead, I will use the following one

```
~/CPS_5401/$
```

4.2 For CPS 5401 :

The following are the URL for the SVN repository.

```
http://repository.cs.utep.edu/svn/cps5401_f15/cpsXX
```

Here is your username and password

Name	UTEP Email	SVN Username	Password
Henry	`hrmoncadalopez@miners.utep.edu`	cps01	HP$777789lp

4.3 Let start with SVN:

```
~/CPS_5401/$ svn checkout  http://repository.cs.utep.edu/svn/cps5401_f15/cpsXX
```

or

```
~/CPS_5401/$ svn co http://repository.cs.utep.edu/svn/cps5401_f15/cpsXX
```

The `XX` on the `cpsXX` should be replaced by a number. Look on your SVN username.

```
~CPS_5401$ svn co https://repository.cs.utep.edu/svn/cps5401_f15/cps01
Authentication realm: <https://repository.cs.utep.edu:443> EOS Subversion Authorization
Password for 'henry': HP$777789lp
Authentication realm: <https://repository.cs.utep.edu:443> EOS Subversion Authorization
Username: cps01
Password for 'cps01': HP$777789lp
```

A `cps01` folder must be created. This `cps01` folder is now link to the SVN repository.

```
~/CPS_5401/$ ll
total 52
drwxrwxr-x 13 henry henry 4096 Sep  3 12:05 ./
drwxrwxr-x  4 henry henry 4096 Sep  3 14:05 ../
drwxrwxr-x  3 henry henry 4096 Sep  3 12:05 cps01/
```

5 Example: Submit a test file

Follow the procedure to build and submit a test file,

- Now, You need to move into your SVN folder

  ```
  ~/CPS_5401/$ cd cps01
  ~/CPS_5401/cps01$
  ```

- This is just a quickly way to build a text file. You can use an editor to do the same.

```
~/CPS_5401/cps01$ echo "I finish my Hw1"> HW1_Henry_oncada.txt
```

You can check that was create using

```
~/CPS_5401/cps01$ vi HW1_Henry_moncada.txt
```

or

```
~/CPS_5401/cps01$ ls
total 24
drwxrwxr-x  3 henry henry 4096 Sep  3 12:05 ./
drwxrwxr-x 13 henry henry 4096 Sep  3 12:05 ../
-rw-rw-r--  1 henry henry   15 Sep  3 12:05 HW1_Henry_oncada.txt
drwxrwxr-x  6 henry henry 4096 Sep  3 12:09 .svn/
```

- **Adding a new file:** Subversion can tell automatically when you change an existing file from the repository, but it doesn't know when you want to add a new file or directory to source control. For this you need to use svn add:

```
~/CPS_5401/cps01$ svn add HW1_Henry_Moncada.txt
```

At this point, you can commit this change like any other.

- **Committing your changes:** Now you're ready to commit your changes the the repository so your teammate(s) can see and build on them. When you're ready to check in your changes, run svn commit on the files or directories you've changed:

```
~/CPS_5401/cps01$ svn commit HW1_Henry_Moncada.txt
```

You'll note that Subversion responds to your checkin with a message:

```
Adding         vis15/HW1_Henry_moncada.txt
Transmitting file data .
Committed revision 23.
```

6 Useful Commands

- **Look around:** You'll frequently want to check to see what you've edited and what you haven't, often before a svn commit command.

```
~/CPS_5401/cps01$ svn status
```

You'll get a list of the files in the current directory, with flags (such as M for modified) next to each file describing its status in your local copy.

- **Updating your sources:** Usually, before you make a checkin, it's a good idea to check to make sure nobody else committed changes while you weren't looking.

```
~/CPS_5401/cps01$ svn update
```

- **Committing your changes (update previous file):** Now you're ready to commit your changes the the repository. When you're ready to check in your changes, run svn commit on the files or directories you've changed:

```
~/CPS_5401/cps01$ svn commit -m HW1_Henry_Moncada.txt
```

You'll note that Subversion responds to your checkin with a message:

```
Adding         vis15/HW1_Henry_moncada.txt
Transmitting file data .
Committed revision 23.
```

- **Delete a file:** Using svn to delete a file from your working copy merely schedules it to be deleted. When you commit, the file is deleted in the repository.

```
~/CPS_5401/cps01$ svn delete myfile
D         myfile

[@Ubuntu]$  svn commit -m "Deleted file 'myfile'."
Deleting       myfile
Transmitting file data .
Committed revision 14.
```

7 Example: Submit Folder

Follow the procedure to submit a test Folder,

- **Adding a New file:** Subversion can tell automatically when you change an existing file from the repository, but it doesn't know when you want to add a new file or directory to source control. For this you need to use svn add:

```
~/CPS_5401/cps01$ svn add New_Folder
```

- **Committing your changes (a folder):** `HW1_Henry_M` is a folder, Commit a simple modification to a file with the commit message on the command line and an implicit target of your current directory ("."):

```
~/CPS_5401/cps01$ svn commit -m "Add New folder to the project."
Transmitting file data .
Committed revision 3.
```

  You'll note that Subversion responds to your checking with a message:

```
Adding         vis15/New_Folder
Transmitting file data .
Committed revision 23.
```