CS5010 (PDP) Subversion Lab Demo

Fall 2009

About Subversion (and some terminology)

Subversion (SVN) is a version control system. Subversion is used in this class primarily to submit homework assignments but it is also useful as a backup system and as a way to sync and transport files when working on multiple computers. In Subversion, files are stored in a *repository*, which is located on a remote server. You communicate with the server using a Subversion client, which must be installed on your computer.

A client is used to *checkout* a *working copy* of the repository onto your local machine. When you want to change or add a file to the repository, you must first change or add the file to your working copy. After you do this, you *commit* the changes in your working copy to the remote repository.

In this class, the repository is located at the following URL:

```
https://trac.ccs.neu.edu/svn/cs5010fall2009
```

Everyone will work from the same repository but each pair will be assigned a subdirectory in the repository that is accessible by the pair members only (using CCS accounts). The subdirectories are named `pairXXX`, where the `XXX` is a three digit number corresponding to the team number that was assigned in lab. So the complete URL for each team is:

```
https://trac.ccs.neu.edu/svn/cs5010fall2009/pairXXX
```

Setting up Subversion

To access the repository, you will need a Subversion client installed on your local machine. Linux and Mac machines come pre-installed with a command-line SVN client. The Windows CCS machines in the lab have both a command-line (via Cygwin) and a graphical client (TortoiseSVN) installed. On your home computer, you will need to install a client yourself. If you have a Windows machine, TortoiseSVN can be downloaded for free from the internet.

This demo will focus on the command-line version of Subversion but the graphical client uses the exact same commands. You will just need to find them in the GUI. If you are using Linux or a Mac, open a terminal window. On Windows CCS machines, open a Cygwin shell prompt (should be accessible from Start->Department Applications).

Checking out a working copy

To check out a working copy onto your local machine, type the following command:

```
svn checkout https://trac.ccs.neu.edu/svn/cs5010fall2009/pairXXX
```

This downloads a copy of every file in the `pairXXX` directory in the repository and stores it on your local machine in a directory called `pairXXX`. You can also add an optional parameter:

```
svn checkout https://trac.ccs.neu.edu/svn/cs5010fall2009/pairXXX localdir
```

where `localdir` is the name of a local directory where you want to store the working copy. If you do not specify a `localdir`, then `pairXXX` will be used as the local directory name.

By default, SVN will use the username of the account that is logged into the computer. To use a different account, you can use the `--username` parameter:

```
svn checkout https://trac.ccs.neu.edu/svn/cs5010fall2009/pairXXX localdir --username otheruser
```

You will then be prompted for the password of `otheruser`. On subsequent commits, SVN will use the credentials of the last user. So on the next commit, SVN will automatically commit as `otheruser`, unless otherwise specified.

Adding a file to a working copy

Once you have a working copy, you can add files to it. To add a file to a working copy, make sure the file you want to add is in your working copy folder and type (also make sure you change the directory to your working copy first):

```
svn add filename
```

where `filename` is the name of the file you want to add. Now the file has been added to the working copy. However, **the file has not been uploaded to the repository yet**.

Committing files to the repository

When you are done making changes and adding files to a working copy, you can upload the changes to the remote repository. First, to see what will happen on the next commit, type:

```
svn status
```

You will see a list of files printed out, each with a letter next to the filename. An explanation of some of the letters is below:

- `?` File has not been added to the working copy (it exists locally only).
- `A` File has been newly added to the working copy and will be uploaded to the repository on the next commit.
- `M` File already exists in the repository but has been modified in the working copy. The modified version will be uploaded to the repository on the next commit.

If a file does not show up in the status list (but does show up in the working copy), then it means that the file is in the repository but is unchanged so nothing will happen to the file on the next commit.

To perform the actual commit, type:

```
svn commit -m "message"
```

where `"message"` is a description of this commit. The message can be useful when you are trying to find a previous version of a file. **Make sure to commit early and commit often.**

To check that your files have been uploaded to the repository, you can enter the url of your subdirectory in a web browser (note the trailing "/" after the `pairXXX` subdirectory name):

```
https://trac.ccs.neu.edu/svn/cs5010fall2009/pairXXX/
```

Updating a working copy

Say someone else has uploaded changes to the repository. This means that your working copy is now out of date. To grab the latest files from the repository, use the `update` command. To try this, first checkout a second working copy of your repository subdirectory (which should now have a file in it) into a different local directory :

```
svn checkout https:://trac.ccs.neu.edu/svn/cs5010fall2009/pairXXX pairXXXcopy
```

Now make a change to a file in one of your working copies and commit the change to the repository. Go to your other working copy and type:

```
svn update
```

and make sure that the changes are downloaded to the other working copy.

Note that if changes exist in both the repository and the working copy, you must do an `update` first before you can commit changes in your working copy. SVN is usually smart enough to automatically merge files but occasionally a manual merge is necessary. This situation should be rare but if you happen to encounter it, you can find out what to do from the free Subversion book located here:

```
http://svnbook.red-bean.com/en/1.5/svn-book.pdf
```

Chapter 1 describes how Subversion works. Chapter 2 introduces some basic commands. For this class, you will not need to know anything that is introduced after Chapter 2.