

In Fundies 2, we use Subversion for collaboration, version control, and homework submission. Subversion keeps versions of your files in a “repository.” We provide a repository on a remote server for you to use. This enables you to retrieve (“checkout,” “update”) and

store (“commit”) your files from multiple locations and enables us to access your code for grading. It also provides backup and a means for reverting to older revisions if necessary.

Getting Subversion

If you use Windows, you will need to install Subversion, but if you have Linux or Mac OS X, the command-line Subversion client is likely pre-installed. On the NEU CCS Windows computers, the GUI TortoiseSVN client is already installed.

For Windows, TortoiseSVN may be downloaded at <http://tortoisesvn.tigris.org/>. The command-line Subversion client may be downloaded at <http://subversion.tigris.org/>.

Command Reference

SVN Book

There is a comprehensive book about Subversion available at:

<http://svnbook.red-bean.com/>

Chapter 1 (*Fundamental Concepts*) gives a high-level description of how Subversion works. Chapter 2 (*Basic Usage*) covers all the commands you will need for this course.

Essential Commands

A brief description of some essential commands are given below:

`help [command]`

Displays usage information for the command *command*.

`checkout repository-url [new-directory]`
`[--username UNIX-username]`

Makes a new working copy from a repository. You only have to do this when you start working on a new machine. The optional *new-directory* parameter specifies a local directory in which to store repository files. The optional *UNIX-username* parameter allows you to specify your remote UNIX account username.

`add [files or directories]`

Places files or directories under svn’s control. Use this command when first creating a file. (This has no effect on the repository until you commit! See also: `rm`, `mkdir`, `mv`, and `cp`.)

`status [files or directories]`

Displays information about changes since your last commit. (See also: `log` and `diff`.)

`update [files or directories]`

Grabs the latest version from the repository and merges with your working copy. Occasionally, there may be a merge conflict that must be resolved manually. The SVN book describes how to do this. However, since you are pair programming in this class, merge conflicts should be rare.

`commit [files or directories]`

Saves changes from the working copy to a new revision in the repository. Use the `-m` switch to add a comment that describes the commit. Attempting to commit will fail if the repository has changed since your last update. You will need to update your working copy first.

SVN in Fundies 2

Each pair has been assigned a Subversion repository such as pair052. In the following examples, pairXXX stands in for your pair repository name.

Your Subversion repository URL is <https://trac.ccs.neu.edu/svn/cs2510s11/pairXXX/>. If you’re using the command-line client, check it out using the `checkout` command as described above.

Each homework assignment [01, 02,...,NN] must have a corresponding hand-in directory [Assignment-01, Assignment-02,..., Assignment-NN] in the root of your repository. You will need to create (see `svn help mkdir`) the directories before starting each assignment, and each one (except for the first one) should make up an Eclipse Project.

Problems after assignment 1 will correspond to Java files within the `src` directory of each assignment, e.g., Assignment-02/src. **You submit your homework by plac-**

ing the required files in the appropriate hand-in directory and committing them to the repository (see `svn help commit`). We collect your homework by grabbing a snapshot of whatever you have submitted in the hand-in directory as of the deadline (Tuesday 22:00 unless otherwise stated).

You must use SVN throughout the development of the homework solution. It is the most convenient, reliable and safe way to backup and manage your work and share your files with your homework partner. Commit your work as often as possible. **You must commit after every programming session, and should commit more often than that.** This way you can have access to it from different machines without using any physical storage or email and without the fear of unintentionally overwriting your files and losing your work. **Do not use email or a USB key to move your work between computers or to transmit it to your partner. Use Subversion.**