CSE 331, Autumn 2011
Section 3 cheat sheet

Subversion (SVN)

Version Control

Software that tracks and manages changes to a set of files and resources.
Importance: Version control is incredibly helpful while working individually on an assignment. It is absolutely vital when working with a team.

Version control provides...

- a shared copy of all code files that all users can access
- current versions of all files, and backups of past versions
- the ability to see what files others have modified and view the changes
- a way to manage conflicts when multiple users modify the same file
- the ability to retrieve accidentally deleted files
- uses beyond source code; it can be used for papers, photos, etc.

Subversion

Installing in Ubuntu: $ sudo apt-get install subversion
Installing in Fedora: System → Admin → Add/Remove Software- “subversion”

Terminology

Repositories: Central location storing a copy of all files.

Repositories can be created anywhere. They can be on the same computer that you are working on. However, this should only be done with personal projects where you just want rollback protection.

For larger, group projects you want a computer that is up and running 24/7 and which has a redundent file system (i.e. RAID). `attu` satisfies both of these requirements and is likely where you should be creating your repositories.

Commit: Checking in a new version of a file that was checked out
Revert: Undoing changes to a file that was checked out
Update: Downloading the latest versions of all files that have been committed by other users.
Conflict: Conflicts come up when two people have made changes to the same file. They require you to resolve in some way. Be very careful when asked to resolve a conflict that you don’t override any changes made by others. The choices for resolving conflicts are: fully accepting changes from one of the versions or merging the files to incorporate all changes.

SVN commands:

command	**description**
svn add *files*	schedule files to be added at next commit.
svn delete *files*	schedule files to be deleted at next commit.
svn ci	commit / check in changed files
svn co *repo*	check out
svn merge *source1 source2*	merge changes
svn revert *files*	restore local copy to repo’s version
svn resolve *files*	resolve merging conflicts
svn update *[files]*	update local copy to latest version
svnadmin create ***repopath***	creating the overall repository on `attu`

Installing Subclipse

To install Subclipse, load Eclipse and click **Help, Install New Software**. In the dialog box that appears, type the following URL into the **Work with:** text box.

- http://subclipse.tigris.org/update_1.6.x

Check the box for Subclipse and click Next. The rest of the installation is straightforward; check appropriate boxes and click Next or Finish.

Creating your project

To create a project, create a New Java Project as usual. You can name the project anything you want (that is appropriate and descriptive). Right click it in Eclipse's Package Explorer window and choose **Team**, then **Share Project...**. In the dialog box that pops up, choose the following:

- repository type: SVN (Next)
- Create a new repository location
- URL: <location of the repository>
- use project name as folder name
- Finish

If you did this properly, your Eclipse project should now be in the repository. To add a file to the system, create a new Java class within the project and save it. Then right-click the project and choose **Team, Commit...** and check in all of your files.

Accessing an existing project

To check out a local copy of a project...

- Choose File, New, **Project...**, and instead of choosing a new Java Project, scroll down to the SVN category and choose **Checkout Projects from SVN**.
- In the next dialog, choose **Create a new repository location**. In the URL field type the location of the repository.
- In the "Select Folder" dialog box that now appears, click the [+] sign next to the top svn+ssh item to show the project inside it. Click on that project to select it, and then choose **Finish**.

If you did it properly, you will now have the project in your Eclipse Package Explorer and can use it like everyone else. Don't forget that when you create new files, you must add them to the repository by right-clicking them and choosing **Team, Add to Version Control**.

In addition to Subclipse and the command-line, there are a couple of other ways that you can try out managing Subversion. These include, but are not limited to, TortoiseSVN (for Windows) and NautilusSVN (for Linux).

Some helpful links:

```
subclipse.tigris.org
web.eecs.utk.edu/ cs360/svn.html
svnbook.red-bean.com/nightly/en/svn-book.html
```