SVN (Subversion) cheat sheet

For UNIX command line

(Note: I've seen Tortoise SVN recommended as a Windows client, but I haven't tried it)

Get help:

svn help – list svn commands

svn help *commandname* – get help with a particular command

online documentation: http://svnbook.red-bean.com/

svn website: http://subversion.apache.org/

Basic workcycle:

svn co *projectpathinrepository localpath* – checkout a directory in the repository for the first time

make changes to files and directories – be sure to use svn commands to move, delete, or copy files, otherwise svn won't keep track of them

svn add *file1 file2 dir1 dir2 dir3 …* – add new files or directories to version control

svn delete *fileordir1 fileordir2 …* - delete local file and remove from version control

svn copy *SRC DST* - copy files or folders. Both SRC and DST can be either a local or repository path

svn move *SRC DST* – move files, basically a copy then delete of original

svn mkdir *PATH* – make a new directory: either make locally and add to version control, or make in repository if PATH is a repository path

if at any point you want to know how your working copy is different from the repository, use **svn status** to look at the change status of files and **svn diff** to see what has changed in a particular file. Note that this does not compare to the current repository, but to the last time you referenced the repository using **svn update**.

svn update – update your local working copy with any changes from the repository. This will automatically merge changes with your local changes if it can, and flag conflicts if it can't. You should also use this if you haven't used your working copy for a while. If any conflicts are flagged, you need to manually modify the files to choose which changes to keep, then do **svn resolved** to let svn know that it is allowed to continue.

svn commit – commit changes in your local working copy into the repository. Use the "-m" option to write a message about what you changed, or SVN will open your default editor (specified by the environmental variable SVN_EDITOR) to write a message.

Getting information:

svn ls *path* – List contents of directory. Locally, same as a normal ls, but in repository shows actual directory structure. Note that doing a normal bash ls in the repository is meaningless.

svn status – get status of files, i.e. how your working copy compares to the last time you checked the repository. By default, only lists files that have been changed since last update

svn diff *path* – print differences between local files in path and repository, in format for **patch** program

svn log – get log entries, i.e. the message from each commit

Going back to older versions:

svn revert – undo local edits, revert to last image of repository

svn update -r *revisionnumber* – update working directory to a particular version number

svn co -r *revisionnumber projectpathinrepository localpath* – checkout a particular version

Referencing repository paths:

file:///pathtorepository - reference a repository on a local machine.

svn+ssh://user@host/pathtorepository – use ssh to reference a repository on a remote machine. Note that this will require you to enter your password for every repository action, unless you set up ssh so that it isn't required

To put a new project under version control:

Method 1 (recommended):

1. create a new directory in the repository for the new project using: **svn mkdir *repositorypath/newproject***
2. cd to the location of your project
3. checkout the empty folder in the repository: **svn co *repositorypath/newproject*** .
4. add project files and directories to version control: **svn add *file1 file2 dir1 dir2 dir3 …***
5. upload the files and directories to the repository: **svn commit**

Method 2:

1. **svn import *[projectpath] repositorypath/newproject***
2. this adds the project to the repository but does not put the current directory under version control, so you must checkout a new working copy