Subversion Cheat Sheet

Basic Work Cycle

Update to the latest revision	`checkout, update`
Do your work, change files	`add, remove, copy, move`
Examine changes	`status, diff, revert`
Merge your work	`update, resolved`
Commit	`commit -m "your comment"`

The Repository

The repository (repo) manages all versions (revisions) of your project. You can't work on files of a repo, you do a checkout to get a working copy from a repository, do your work there, and commit your changes back to the repo.

Create a repository	`svnadmin create /path/to/repo`

The Working Copy

This is where your work is done. You may work with the files as with standard files, but leave those ".svn" directories alone. Renaming, moving, copying, adding or deleting files and directories has to be done through svn additionally to creating them, to tell the versioning system that those files are also part of the project (e.g. `touch info.txt && svn add info.txt`). Else they will be missing after or recreated with the next checkout.
You most likely want to execute these commands with the working copy as your current directory (PWD).

Get a working copy (= checkout)	`svn (checkout|co) [-r REV] url://repo/path/ path/to/project`
Update your working copy *1	`svn (update|up) [-r REV] [path/to/project]`

Add a file/directory to the project (!= import)	`svn add path/to/fileOrDirectory`
Copy a file/directory and mark it as added	`svn copy path/to/fileOrDirectory path/to/newCopy`
Rename/move a file/directory	`svn (move|mv|rename|ren) path/to/dir path/to/newDir`
Delete a file/directory	`svn (delete|del|remove|rm) path/to/fileOrDirectory`
Lock (or unlock) a file/directory for editing	`svn (lock|unlock) path/to/fileOrDirectory`

Check the status of your local file(s) *2	`svn status [path/to/fileOrDirectory]`
Compare the status to the repo *2	`svn status -u [path/to/fileOrDirectory]`
Do a diff of your file(s) and the repo	`svn diff [-r REV] [path/to/fileOrDirectory]`
Revert to last revision (last checkout or update)	`svn revert [path/to/fileOrDirectory]`
Resolve conflicts (after you reviewed them!)	`svn resolved [path/to/fileOrDirectory]`
Commit and create a new revision	`svn (commit|ci) [path/to/project] -m "your comment"`

Import unversioned files/folders *3	`svn import path/to/original url://repo/path/`
Show a log	`svn log [-r REV] [path/to/fileOrDirectory]`
Concatenate a specific file	`svn cat [-r REV] path/to/file`

*1) Update Status Descriptions

6 possible status can occur after you do a "update", per file:

U	unchanged
A	added
D	deleted
R	replaced
G	merged
C	conflict!

If **C** happens 3 copies of the file will be created: `file.c.mine`, `file.c.rOLD` and `file.c.rNEW`. `file.c` will have conflict markers added. You will only be able to commit again when those 3 files have been removed. This can be done automatically by running `svn resolved file.c`, which removes those 3 files.

*2) Extended Status Descriptions

```
  L       some_dir               # svn left a lock in the .svn area of some_dir
M         bar.c                  # the content in bar.c has local modifications
 M        baz.c                  # baz.c has property but no content modifications
X         3rd_party              # dir is part of an externals definition
?         foo.o                  # svn doesn't manage foo.o
!         some_dir               # svn manages this, but it's missing or incomplete
                                 # do a svn delete to remove it from the repo
~         qux                    # versioned as file/dir/link, but type has changed
I         .screenrc              # svn doesn't manage this, and is set to ignore it
A  +      moved_dir              # added with history of where it came from
M  +      moved_dir/README       # added with history and has local modifications
D         stuff/fish.c           # file is scheduled for deletion
A         stuff/loot/bloo.h      # file is scheduled for addition
C         stuff/loot/lump.c      # file has textual conflicts from an update
 C        stuff/loot/glub.c      # file has property conflicts from an update
R         xyz.c                  # file is scheduled for replacement
    S     stuff/squawk           # file or dir has been switched to a branch
      K   dog.jpg                # file is locked locally; lock-token present
      O   cat.jpg                # file is locked in the repository by other user
      B   bird.jpg               # file is locked locally, but lock has been broken
      T   fish.jpg               # file is locked locally, but lock has been stolen
```

See:
http://svnbook.red-bean.com/en/1.4/

*3) Import Procedure

1. Create a repo (`svnadmin create`)
2. Import your project (`svn import`)
3. Do your first checkout (`svn checkout`)
4. Work on checked out working copy