SVN Commands

INTRODUCTION:

What is Subversion (SVN):

Subversion (SVN) is a version control system. It is used to maintain current and historical versions of files such as source code, web pages, and documentation.

Subversion is a free/open-source version control system. That is, Subversion manages files and directories, and the changes made to them, over time. This allows you to recover older versions of your data, or examine the history of how your data changed.

Subversion is a centralized system for sharing information. At its core is a repository, which is a central store of data. The repository stores information in the form of a file system tree—a typical hierarchy of files and directories. Any numbers of clients connect to the repository, and then read or write to these files.

The files and directories are checked out of the repository and into your local project work area. This called your "working directory". Changes are made to files in your "working directory". After changes are made, to create the next working version, the files are checked into the Subversion CM repository.

Why to use SVN:

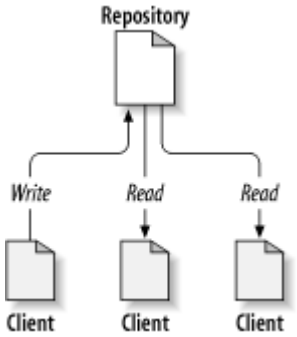
- It helps to track changes
- Multiple users can work together on same project

How to use SVN:

- Module wise commit
- Proper commenting while commit so when needed it helps to revert old changes

SVN Architecture:

- SVN is using Client-server architecture



COMMANDS:

svn --help — List Subversion commands

svn help — Help on given "subcommand"

Usage:

svn help [subcommand]
svn ? [subcommand] [PATH]
svn h [subcommand]

Example:

- svn help add

svn checkout — Check out a working copy from a repository.

Usage:

svn checkout URL[@REV]... [PATH]
svn co URL[@REV]... [PATH]

Description:

Check out a working copy from a repository. If PATH is omitted, the basename of the URL will be used as the destination. If multiple URLs are given each will be checked out into a subdirectory of PATH, with the name of the subdirectory being the basename of the URL.

Example:

- svn checkout http://repos/svn/trunk/sourcepath destinationpath

This creates:
path/file1
path/file2

- svn checkout http://repos/svn/trunk/sourcepath .

This creates same out put as above, "." means current directory

- svn co -r 100 http://repos/svn/trunk/path .

This checkouts the repository from revision number 100

svn add — Add files, directories, or symbolic links.

Usage:

svn add [filename or directory]

Description:

Add files, directories, or symbolic links to your working copy and schedule them for addition

to the repository. They will be uploaded and added to the repository on your next commit. If you add something and change your mind before committing, you can unschedule the addition using **svn revert**.

Example:

- svn add file (it will adds file in svn)
- svn add dir (it will add all files from dir and dir itself)
- svn add . /* (recursively adds all files and dirs into svn)

svn info — Print information about PATHs.

Usage:

svn info [PATH]

Description:

Display information about file or directory specified in PATH. (Date modified, author, revision, path in repository, Last Changed Author, Last Changed Revision, Last Changed Date, etc.)

svn commit — Send changes from your working copy to the repository.

Usage:

svn commit [PATH...]
svn commit --message "Message goes here." [PATH...]
svn commit -m "Message goes here." filename
svn ci filename1 filename2 filename3
svn ci .

Description:

Check-in (commit) local "working" file, files or directory and contents (recursively) into Subversion repository. Atomic, i.e. all committed or none, no incomplete check-in.

svn log — Display commit log messages.

Usage:

svn log [PATH]

Description:

The default target is the path of your current directory. If no arguments are supplied, **svn log** shows the log messages for all files and directories inside of (and including) the current working directory of your working copy. You can refine the results by specifying a path, one or more revisions, or any combination of the two. The default revision range for a local path is BASE: 1.

Examples:

svn log filename

svn log .
svn log http://URL/path/file
svn log -v .
svn log -r RevisionNumber http://URL/path/file

svn delete — Delete an item from a working copy or the repository.

Usage:

svn delete [PATH]
svn del [PATH]
svn remove [PATH]
svn rm [PATH]

Description:

Delete files or directories specified by PATH from repository. Files (and directories that have not been committed) are immediately removed from the working copy. The command will not remove any unversioned or modified items; use the --force switch to override this behavior.

After **svn delete**, must perform a "commit" to update the repository and local working directory with the changes.

Examples:

- svn delete filename
- svn delete directory

svn revert — Undo all local changes.

Usage:

svn revert [PATH]

Description:

Reverts any local changes to a file or directory and resolves any conflicted states. **svn revert** will not only revert the contents of an item in your working copy, but also any property changes. It behaves like undo.

Examples:

- svn revert filename
- svn revert directory

svn status — Print the status of working copy files and directories.

Usage:

svn status [PATH]
svn stat [PATH]
svn st [PATH]

Description:

Show the status of working copy files and directories. With no arguments, it prints only locally modified items (no repository access).
Show out of date file info: svn status --show-updates
(equivalent: svn status -u)
-u: Determines status by comparing your local repository with the server repository. Without this option, the status shown will only be the changes you have made in your local repository.

-q: Quiet. Do not print "?: File/directory not under version control" or "!: File/directory missing" extraneous information.

The first five columns in the output are each one character wide, and each column gives you information about different aspects of each working copy item.

The first column indicates that an item was added, deleted, or otherwise changed.

' ' : No modifications.

'A' : Item to be added.

'C' : Item is in conflict with updates received from the repository.

'D' : Item to be deleted.

'G' : Item to be merged with updates from server

'M' : Item has been modified.

'R' : Item has been replaced in your working copy.

'X' : Item is related to an externals definition.

'I' : Item is being ignored (e.g. with the svn: ignore property).

'?' : Item is not under version control.

'!' : Item is missing (e.g. you moved or deleted it without using svn). This also indicates that a directory is incomplete (a checkout or update was interrupted).

'~': Item is versioned as one kind of object (file, directory, link), but has been replaced by different kind of object.

The second column tells the status of a file's or directory's properties.

The third column is populated only if the working copy directory is locked.

The fourth column is populated only if the item is scheduled for addition-with-history.

The fifth column is populated only if the item is switched relative to its parent.

The out-of-date information appears in the eighth column (only if you pass the --show-updates switch).

svn resolved — Remove “conflicted” state on working copy files or directories.

Usage:

svn resolved [PATH]

Description:

Remove “conflicted” state on working copy files or directories. This routine does not semantically resolve conflict markers; it merely removes conflict-related artifact files and allows PATH to be committed again; that is, it tells Subversion that the conflicts have been “resolved”.

Run this command after resolving merge conflicts. Next "commit" your changes.

svn cleanup — Recursively clean up the working copy.

Usage:

svn cleanup [PATH]

Description:

Recursively clean up the working copy, removing locks resuming unfinished operations. If you ever get a “working copy locked” error, run this command to remove stale locks and get your working copy into a usable state again.

svn list — List directory entries in the repository.

Usage:

svn list [PATH]

Description:

List file or directory of files in repository. Used to browse repository before checkout.

svn update — Update your working copy.

Usage:

svn update [PATH]

Description:

It brings changes from the repository into your working copy. If no revision given, it brings your working copy up-to-date with the HEAD revision. Otherwise, it synchronizes the working copy to the revision given by the --revision switch.

This command brings changes from the repository into your working copy. (recursively for all files in the current directory and all below it).

If no revision given, it brings your working copy up-to-date with the HEAD revision. Otherwise, it synchronizes the working copy to the revision given by the --revision switch.

If there have been updates to the svn repository since you downloaded the files, subversion will give you the opportunity to merge. Status of files will use the coding as stated above for "status". Files marked with a "C" (conflict) should be merged of reverted. If merged then one can perform a "resolve" and then a "check-in".
If a file name is specified, only that file is updated.

Examples:

svn update
svn update filename
svn update -r458 filename
svn update --ignore-externals ./

svn diff — Display the differences between two paths.

Usage:

svn diff
svn diff [-r N [:M]]

Description:

Show file diffs between SVN repository and your local file changes when no path specified.

Examples:

- svn diff -r rev1:rev2 (It compares diff between revision rev1 and rev2 in current repository)
- svn diff -r rev1:rev2 filename (it compares diff for that particular file only)

patch — Generate patch file with the difference.

Usage:

svn diff filename > patch-file
svn diff [-r N [: M]] > patch-file

Description:

Generates the difference and store in patch file which is then useful if you want to import difference in other repository.

Examples:

- svn diff -r rev1:rev2 > file.patch

svn copy — Copy a file or directory in a working copy or in the repository.

Usage:

svn copy source destination
svn cp source destination

Description:

Copy a file in a working copy or in the repository. SRC and DST can each be either a working copy (WC) path or URL:

WC -> WC : Copy and schedule an item for addition (with history).
WC -> URL : Immediately commit a copy of WC to URL.
URL -> WC : Check out URL into WC, and schedule it for addition.
URL -> URL: Complete server-side copy. This is usually used to branch and tag.

Create New Branch using copy command
svn copy http://repos/svn/project/trunk http://repos/svn/project/branches -m "Branch Created”

Examples:

- svn copy foo.txt bar.txt
- svn copy near.txt http://repos/svn/project/trunk -m "Remote copy."
- svn copy http://repos/svn/project/trunk http://repos/svn/project/tags/TagName-1.4.5 -m "Tag Release 1.4.5" (Tag a release. Takes a snapshot of the repository and assigns a name. This can be performed at any directory branch.)

svn switch — Update working copy to a different URL.

Usage:

svn switch URL [PATH]
svn switch --relocate http://server/old-path http://server/new-path

Description:

Switch your local working copy to mirror a new repository branch instead of main trunk or previous branch. Also allows you to point your repository to a new path on the server if the server path changes since you performed a check-out.

Examples:

- svn switch http://repos/svn/project/branches (switches current working copy to branch)

svn merge — Apply the differences between two sources to a working copy path.

Usage:

svn merge sourceURL1[@N] sourceURL2[@M] [WCPATH]
svn merge -r N:M SOURCE[@REV] [WCPATH]
svn merge file1@revJ file2@revK
svn merge -r Rev1: Rev2 sourceurl [WCPATH]
svn merge -r Rev:HEAD file-name

Description:

Merge directory changes into your current working directory or merge a file in Subversion into the file in your working directory. If target is not specified, the identical base name or current directory is assumed.

While merging if a conflict happens then edit conflicts and resolve it: svn resolve file-name or abort changes: svn revert file-name and then commit the changes.

Test merge: Before merge command by –dry-run you can check what files/dirs will be affected by merge, so in this No changes are made to your local working copy but shows Subversion feedback as if merge was performed.

svn merge --dry-run -r 414:413 http://url/path

svn mkdir — Create a new directory under version control.

Usage:

svn mkdir [PATH]

Description:

Create a directory with a name given by the final component of the PATH or URL.

Examples:

- svn mkdir newdir
- svn mkdir -m "Making a new dir." http://repos/svn/trunk/newdir

svn export — Export a clean directory tree.

Usage:

svn export [PATH]
svn export -r Rev-Number [PATH]

Description:

Export directory tree to your file system but it will not be a "working directory" under SVN control, means the repository without .svn.

In second, export directory tree of specified version and create local directory tree and files not under SVN control.

svn import — Recursively commit a copy of PATH to URL.

Usage:

svn import [PATH] URL

Description:

Recursively commit a copy of PATH to URL. If PATH is omitted “.” is assumed. Parent directories are created in the repository as necessary.

Examples:

- svn import -m "New import" myproj http://repos/trunk/test

svn move — Move a file or directory.

Usage:

svn move SRC DST

Description:

This command moves a file or directory in your working copy or in the repository.

This command is equivalent to a svn copy followed by svn delete.

WC -> WC: Move and schedule a file or directory for addition (with history).

URL -> URL: Complete server-side rename.

Examples:

- svn move foo.c bar.c

- svn move -m "Move a file" http://repos/foo.c http://repos/bar.c

svn blame — Show author and revision information in-line for the specified files or URLs.

Usage:

svn blame TARGET

Description:

Show author and revision information in-line for the specified files or URLs. Each line of text is annotated at the beginning with the author (username) and the revision number for the last change to that line.

svn cat — Output the contents of the specified files or URLs.

Usage:

svn cat [PATH]

Description:

List contents of file under Subversion control.

Reference: http://svnbook.red-bean.com/en/1.1