

Subversion

9 Most Used SVN Commands with Examples.

Author:Sudarshan

SVN stands for Subversion.

Subversion is a free/open-source version control system. Subversion manages files and directories over time. A tree of files is placed into a central repository. The repository is much like an ordinary file server, except that it remembers every change ever made to your files and directories. This allows you to recover older versions of your code, or examine the history of how your code was changed.

This document explains some basic SVN commands with examples.

SVN Working Copy

SVN is a repository that holds all our versioned data, which is also called as SVN server. SVN client program which manages local reflections of portions of that versioned data which is called as working copy. SVN client can access its repository across networks. Multiple users can access the repository at the same time.

1. SVN Checkout - Create working copy

Checkout command is used to download sources from SVN repository to working copy. If you want to access files from the SVN server, checkout is the first operation you should perform.

SVN checkout creates the working copy, from where you can do edit, delete, or add contents. You can checkout a file, directory, trunk or whole project. To checkout you should know URL of the components you want to checkout.

Syntax:

```
svn checkout/co URL PATH
```

URL is the URL of the components to checkout

```
svn co https://www.thegeekstuff.com/project/branches/release/migration/data/cfg  
/home/sasikala/cfg/
```

```
A  /home/sasikala/cfg/ftp_user.cfg  
A  /home/sasikala/cfg/inventory.cfg  
A  /home/sasikala/cfg/email_user.cfg  
A  /home/sasikala/cfg/svn-commands  
Checked out revision 811.
```

2. SVN Commit - Save changes to the repository

Whenever you do changes to the working copy, it will not reflect in SVN server. To make the changes permanent, you need to do SVN commit.

Syntax:

```
svn commit -m "log messages"
```

Explain why you are changing the file in the -m option.

Put the log message in "" , so that u can write sentences.

```
svn commit -m "Making the file empty" welcome.html
```

```
Sending      welcome.html
Transmitting file data .
Committed revision 813.
```

After this whenever you update your working copy or checkout, the changes will appear in the server.

3. SVN List - Lists directory entries

svn list is useful when you want to view the content of the SVN repository, without downloading a working copy.

Syntax:

```
svn list
```

The following example lists all the files available in the given URL in the repository without downloading a working copy. When you execute svn list command with -verbose option it displays the following information.

- Revision number of the last commit
- Author of the last commit
- Size (in bytes)
- Date and time of the last commit

```
svn list --verbose https://www.thegeekstuff.com/project/branches/release/migration/data/bin
16 sasikala 28361 Apr 16 21:11 README.txt
21 sasikala 0 Apr 18 12:22 INSTALL
22 sasikala Apr 18 10:17 src/
```

4. SVN Add - Add a new file to SVN repository

When you want to add a new file (or directory) to the repository you need to use SVN add command. The repository will have newly added file, only when you do SVN commit. Now let us add a new file called "hello.html" to our repository.

1. Create a file in local working copy
2. Add the file into SVN repository

svn add filename will add the files into SVN repository.

```
svn add hello.html
A      hello.html
```

- Commit the added the file

- Until you commit, the added file will not be available in the repository.

```
svn commit -m "Adding a file hello.html by Sravya" hello.html
Adding      hello.html
Transmitting file data .
Committed revision 814.
```

5. SVN Delete - Removing a file from repository

SVN delete command deletes an item from the working copy (or repository). File will be deleted from the repository when you do a SVN commit.

```
Syntax:
svn delete URL
```

Now let us remove the recently created file called "hello.html".

```
svn delete hello.html
D      hello.html

$ svn commit -m "Removing the file hello.html by Lalitha" hello.html
Deleting    hello.html
Committed revision 814.
```

Now you can do svn list and check whether the file was deleted from the repository.

6. SVN Diff - Display the difference

SVN diff displays the differences between your working copy and the copy in the SVN repository. You can find the difference between two revisions and two paths etc.,

```
Syntax:
svn diff filename

svn -r R1:R2 diff filename
```

The above example compares the filename@R1 and [filename@R2](#).

I edited the content of hello.html file from testing to tester, which is shown below using the svn diff command.

```
svn diff hello.html
Index: hello.html
=====
--- hello.html    (revision 815)
+++ hello.html    (working copy)
@@ -1,1 @@
-testing
+tester
```

7. SVN Status - Status of the working copy

Use svn status command to get the status of the file in the working copy. It displays whether the working copy is modified, or its been added/deleted, or file is not under revision control, etc.

```
Syntax:

svn status PATH
```

The following example shows the status of my local working copy,

```
svn status /home/sasikala/hello.html
M    /home/sasikala/cfg/welcome.cfg
M    /home/sasikala/cfg/test.html
```

‘M’ represents that the item has been modified. “svn help status” command will explain various specifiers showed in SVN status command.

8. SVN Log - Display log message

As we discussed in the beginning of this article, SVN remembers every change made to your files and directories. To know all the commits made in a file or directory, use SVN log command.

Syntax:

```
svn log PATH
```

The following displays all the commits made on hello.html file

```
svn log hello.html
```

```
-----  
r815 | sasikala | 2011-04-16 05:14:18 -0700 (Sat, 16 Apr 2011) | 1 line
```

```
Adding a file hello.html
```

Since we made only one commit in the file hello.html, it shows only one log message with the details.

9. SVN Update - Update the working copy.

svn update command brings changes from the repository into your working copy. If no revision is specified, it brings your working copy up-to-date with the HEAD revision. Otherwise, it synchronizes the working copy to the revision given in the argument. Always before you start working in your working copy, update your working copy. So that all the changes available in repository will be available in your working copy. i.e latest changes.

Syntax:

```
svn update PATH
```

In case some other user added/deleted file in URL,

<https://www.thegeekstuff.com/project/branches/release/migration/data/cfg>, your working copy will not have those files by default, until you update your working copy.

```
svn update  
A new/usercfg  
A new/webcfg  
Updated to revision 819.
```

In the above svn update command output, A represents that this file is “Added” to the working copy.