

SubVersion Howto Guide

By Ashley Chew

Last revision 22072008

This document is a quick induction into the use of SubVersion via the command line ie on Linux but is the same as on Windows provided the command line tools are installed.

Subversion or commonly known as SVN in most communities, is a version control system to manage and maintain versions of files between multiple users much like Concurrent Version System known as CVS and is largely seen as the successor to CVS for its features and simplicity.

First thing that needs to be done is to decide on a common area which is accessible by you and your members for the project which is commonly accessible.

This documentation is based for the unit CITS3200 which has the default assumptions. The default machine used for the repository is **cits3200.csse.uwa.edu.au** and the default common shared area is **/home/cits3200/cits3200XX** where **XX** is group number or letter (Substitute accordingly for machine name and common share area accordingly)

For this example, it will show step how to setup a SVN repository, checking in/out and updating the repository.

I am currently sitting on `uggp.csse.uwa.edu.au`, so I'm going to remotely connect to the `cits3200.csse.uwa.edu.au`.

```
[ashley@uggp]/home/staff/ashley% ssh ashley@cits3200.csse.uwa.edu.au
```

```
The authenticity of host 'cits3200 (130.95.1.45)' can't be established.
```

```
RSA key fingerprint is 52:03:a9:20:52:75:f0:1e:29:80:47:73:da:e6:19:12.
```

```
Are you sure you want to continue connecting (yes/no)? yes
```

```
ashley@pc3200's password:
```

```
Last login: Tue Jul 22 11:23:58 2008 from linux-vm2.csse.uwa.edu.au
```

```
*****
***                               ***
***           Welcome to CITS3200           ***
***                               ***
***           Centos 5.1                   ***
***                               ***
***           Reserved exclusively for CITS3200 Students           ***
***                               ***
***           Authorised users only!       ***
***                               ***
*****
```



Now that I have connected, I will traverse down to the shared area ie

```
[ashley@cits3200]/% cd /home/cits3200/cits3200xx
```

```
[ashley@cits3200]/home/cits3200/cits3200xx%
```

Ask the machines what is its name.

```
[ashley@cits3200]/home/cits3200/cits3200xx% hostname
```

```
cits3200
```

Tell me where I am on the filesystem

```
[ashley@cits3200]/home/cits3200/cits3200xx% pwd
```

```
/home/cits3200/cits3200xx
```

Show me a listing of the files and directory

```
[ashley@cits3200]/home/cits3200/cits3200xx% ls -al
```

```
total 24
drwxrwxr-x 6 apache staff 4096 Jul 10 14:24 .
drwxr-xr-x 29 root root 4096 Jul 22 11:27 ..
drwxrwxr-x 2 apache staff 4096 Jul 10 12:28 public_html
drwxrwxr-x 7 apache staff 4096 May 9 11:40 svn
drwxrwxr-x 9 apache staff 4096 Jul 10 14:24 trac
drwxrwxr-x 2 apache staff 4096 Jul 10 15:07 weblogs
```

What groups do I belong to?

```
[ashley@cits3200]/home/cits3200/cits3200xx% groups
staff systems
```

Now from the above commands you can see that I've changed directory into /home/cits3200/cits3200xx and the default directory/files are owned by apache and the group staff which I am a member of which both have read, write and execute

permissions. If these permissions are not set correctly ie for the group it will defeat the purpose of the repository as they can't get access to the files/directories in the area

Now that we are in the common group area ie /home/cits3200/cits3200xx, we need to create Subversion Repository Area called "svnroot" (Can actually called anything but it seems like most people call the root repository area svnroot so I'm just sticking with the default syntax) and create a project called project_example to check in/out files and directories.

```
[ashley@cits3200]/home/cits3200/cits3200xx% pwd
```

```
/home/cits3200/cits3200xx
```

Create a directory svnroot.

```
[ashley@cits3200]/home/cits3200/cits3200xx% mkdir svnroot
```

List the contents of svnroot.

```
[ashley@cits3200]/home/cits3200/cits3200xx% ls -al svnroot
```

```
total 8
drwxr-xr-x 2 ashley staff 4096 Jul 22 11:36 .
drwxrwxr-x 7 apache staff 4096 Jul 22 11:36 ..
```

(We can see that the directory is empty)

Now we create a SVN project called project_example.

```
[ashley@cits3200]/home/cits3200/cits3200xx% svnadmin create svnroot/project_example
```

(We just initialised a project called project_example with the necessary files and structure)

```
[ashley@cits3200]/home/cits3200/cits3200xx% ls -al svnroot
```

```
total 12
drwxr-xr-x 3 ashley staff 4096 Jul 22 11:37 .
drwxrwxr-x 7 apache staff 4096 Jul 22 11:36 ..
drwxr-xr-x 7 ashley staff 4096 Jul 22 11:37 project_example
```



```
[ashley@cits3200]/home/cits3200/cits3200xx% ls -al svnroot/project_example
```

```
total 36
```

```
drwxr-xr-x 7 ashley staff 4096 Jul 22 11:37 .  
drwxr-xr-x 3 ashley staff 4096 Jul 22 11:37 ..  
drwxr-xr-x 2 ashley staff 4096 Jul 22 11:37 conf  
drwxr-xr-x 2 ashley staff 4096 Jul 22 11:37 dav  
drwxr-sr-x 5 ashley staff 4096 Jul 22 11:37 db  
-r--r--r-- 1 ashley staff  2 Jul 22 11:37 format  
drwxr-xr-x 2 ashley staff 4096 Jul 22 11:37 hooks  
drwxr-xr-x 2 ashley staff 4096 Jul 22 11:37 locks  
-rw-r--r-- 1 ashley staff 229 Jul 22 11:37 README.txt
```

As you can see it just populated the project_example directory, but the group permissions does not have all the permissions set for the group ie missing write permissions so let set the permissions for the entire directory and files recursively just in case.

```
[ashley@cits3200]/home/cits3200/cits3200xx% pwd
```

```
/home/cits3200/cits3200xx
```

```
[ashley@cits3200]/home/cits3200/cits3200xx% chgrp -R staff svnroot
```

```
[ashley@cits3200]/home/cits3200/cits3200xx% chmod -R g+rwX svnroot
```

```
[ashley@cits3200]/home/cits3200/cits3200xx% ls -al svnroot
```

```
total 12
```

```
drwxrwxr-x 3 ashley staff 4096 Jul 22 11:37 .  
drwxrwxr-x 7 apache staff 4096 Jul 22 11:36 ..  
drwxrwxr-x 7 ashley staff 4096 Jul 22 11:37 project_example
```

```
[ashley@cits3200]/home/cits3200/cits3200xx% ls -al svnroot/project_example
```

```
total 36
```

```
drwxrwxr-x 7 ashley staff 4096 Jul 22 11:37 .  
drwxrwxr-x 3 ashley staff 4096 Jul 22 11:37 ..  
drwxrwxr-x 2 ashley staff 4096 Jul 22 11:37 conf  
drwxrwxr-x 2 ashley staff 4096 Jul 22 11:37 dav  
drwxrwsr-x 5 ashley staff 4096 Jul 22 11:37 db  
-r--rwxr-- 1 ashley staff  2 Jul 22 11:37 format  
drwxrwxr-x 2 ashley staff 4096 Jul 22 11:37 hooks
```

```
drwxrwxr-x 2 ashley staff 4096 Jul 22 11:37 locks
-rw-rwxr-- 1 ashley staff 229 Jul 22 11:37 README.txt
```

As you can see now all files are owned the group staff which has read, write and execute permissions.

You can the set default permissions of files being create by setting the umask variable in your shell.

That is basically if for setting up a SVN repository, the next part is actually checking in/out files from the SVN repository remotely via ssh.

Lets say I'm working on `uggp.csse.uwa.edu.au` and I want to check out the contents of `project_example` (Although the `project_example` at the moment is empty). So I am going to create a directory called `svn_checkout` as a work area for me to checkout SVN projects .

```
ashley@uggp:~:667> hostname
uggp
```

```
ashley@uggp:~:667> pwd
/home/staff/ashley
```

```
ashley@uggp:~:668> mkdir svn_checkout
ashley@uggp:~:669> ls -al svn_checkout
```

```
total 24
drwxr-xr-x  2 ashley staff 4096 2008-07-22 15:17 ./
drwx--x--x 141 ashley staff 20480 2008-07-22 15:17 ../
```

(The `svn_checkout` directory we created should be empty)

```
ashley@uggp:~:670> cd svn_checkout
```

```
ashley@uggp:~/svn_checkout:671> pwd

/home/staff/ashley/svn_checkout
```

```
ashley@uggp:~/svn_checkout:672> svn co
svn+ssh://ashley@cits3200.csse.uwa.edu.au/home/cits3200/cits3200xx/svnroot/project_example
```

```
ashley@cits3200.csse.uwa.edu.au's password:
```

(Prompts me for my Linux password , if successful you should see something like this)

Checked out revision 0.

```
ashley@uggp:~/svn_checkout:673> cd ..
```

```
ashley@uggp:~:674> ls -al svn_checkout
```

```
total 28
```

```
drwxr-xr-x  3 ashley staff  4096 2008-07-22 15:18 ./
```

```
drwx--x--x 141 ashley staff 20480 2008-07-22 15:17 ../
```

```
drwxr-xr-x  3 ashley staff  4096 2008-07-22 15:18 project_example/
```

As you can see it checked out or rather it made a copy of the project_example area onto my home area which I can work on but remember there no files within the project_example as of yet.

```
ashley@uggp:~:675> ls -al svn_checkout/project_example
```

```
total 12
```

```
drwxr-xr-x  3 ashley staff  4096 2008-07-22 15:18 ./
```

```
drwxr-xr-x  3 ashley staff  4096 2008-07-22 15:18 ../
```

```
drwxr-xr-x  6 ashley staff  4096 2008-07-22 15:18 .svn/
```

Although there are zero files in the project area it created a directory called .svn which has information about the project itself.

Lets create a couple of files and directories in our locally checkout version of project_example

```
ashley@uggp:~/svn_checkout:680> pwd
```

```
/home/staff/ashley/svn_checkout
```

```
ashley@uggp:~/svn_checkout:681> cd project_example
```

```
ashley@uggp:~/svn_checkout/project_example:683> mkdir sample
```

Created a sample directory under the project_example directory

```
ashley@uggp:~/svn_checkout/project_example:684> touch fileA.txt
```

Created a empty file called fileA.txt in project_example directory

```
ashley@uggp:~/svn_checkout/project_example:685> touch fileB.txt
```

Created a empty file called fileB.txt in project_example directory

```
ashley@uggp:~/svn_checkout/project_example:686> touch sample/fileC.txt
```

Created a empty file called fileC.txt in a subdirectory called sample under the project_example directory.

```
ashley@uggp:~/svn_checkout/project_example:687> echo "Demo" >> sample/fileC.txt
```

Injected some contents ie the word Demo into fileC.txt file.

```
ashley@uggp:~/svn_checkout:698> cat project_example/sample/fileC.txt
```

```
Demo
```

To prove that I injected the word Demo in the file fileC.txt

```
ashley@uggp:~/svn_checkout/project_example:688> cd ..
```

Now we have made changes by creating several files in the locally check out version of the project_example in our own home area. We have to tell SVN to check it back into the repository you can check individual files or entire directories.

In this example, I am explicitly tell SVN that I just want to check in one file in this case fileA.txt

```
ashley@uggp:~/svn_checkout:680> pwd
```

```
/home/staff/ashley/svn_checkout
```

Shows my current working directory on `uggp.csse.uwa.edu.au`

```
ashley@uggp:~/svn_checkout:700> cd project_example
```

```
ashley@uggp:~/svn_checkout/project_example:701> ls
```

```
fileA.txt fileB.txt sample/
```

As before there is fileA.txt, fileB.txt and a sample directory which has fileC.txt in it.

```
ashley@uggp:~/svn_checkout/project_example:703> svn add fileA.txt
```

```
A    fileA.txt
```

I'm telling SVN explicitly that fileA.txt is part of the project and needs to be committed to the repository on `cits3200.csse.uwa.edu.au`

```
ashley@uggp:~/svn_checkout/project_example:704> svn ci --message "Revision 1"
```

```
ashley@cits3200.csse.uwa.edu.au's password:
```

```
Adding    fileA.txt
Transmitting file data .
Committed revision 1.
```

Subversion has committed fileA.txt to subversion repository `project_example` that I originally checked out from `cits3200.csse.uwa.edu.au`.

You can do this for multiple files or directories ie I have not checked in fileB.txt nor the fileC.txt in the subdirectory `sample`.

```
ashley@uggp:~/svn_checkout/project_example:737> pwd
```

```
/home/staff/ashley/svn_checkout/project_example
```

```
ashley@uggp:~/svn_checkout/project_example:738> svn add fileB.txt
```

```
A    fileB.txt
```

Added fileB.txt to be checked into cits3200.csse.uwa.edu.au the next time I do a SVN commit (ci)

```
ashley@uggp:~/svn_checkout/project_example:739> svn add sample
```

```
A    sample
```

```
A    sample/fileC.txt
```

Added the subdirectory to be checked into cits3200.csse.uwa.edu.au the next time I do a SVN commit (ci). Notice that it found the subdirectory file fileC.txt without me explicitly mentioning it.

```
ashley@uggp:~/svn_checkout/project_example:740> svn ci --message "Revision 2"
```

```
ashley@cits3200.csse.uwa.edu.au's password:
```

```
Adding    fileB.txt
```

```
Adding    sample
```

```
Adding    sample/fileC.txt
```

```
Transmitting file data ..
```

```
Committed revision 2.
```

As you can see now its committed the two files into the project_example. SVN will only do comiit files into the repository if it knows about it be either doing a svn add to a file or direcotory. SVN will only commit files if they are different and it knows about it ie

```
ashley@uggp:~/svn_checkout/project_example:743> pwd
```

```
/home/staff/ashley/svn_checkout/project_example
```

```
ashley@uggp:~/svn_checkout/project_example:744> echo "Hello" >> fileA.txt
```

I've injected the word "Hello" into fileA.txt

```
ashley@uggp:~/svn_checkout/project_example:745> cat fileA.txt
```

Hello

Show you the contents of fileA.txt

```
ashley@uggp:~/svn_checkout/project_example:746> svn ci --message "Revision 3 - Changed FileA.txt"
```

```
ashley@cits3200.csse.uwa.edu.au's password:
```

```
Sending fileA.txt
```

```
Transmitting file data .
```

```
Committed revision 3.
```

As you can see the only changes was to fileA.txt so it committed only the changes that it is aware of.

Now lets delete the files that I've checked out on uggp.csse.uwa.edu.au and recheck the files out to see if they have been committed.

```
ashley@uggp:~/svn_checkout:755> pwd
```

```
/home/staff/ashley/svn_checkout
```

```
ashley@uggp:~/svn_checkout:756> ls -al
```

```
total 28
```

```
drwxr-xr-x  3 ashley staff 4096 2008-07-22 15:18 ./
```

```
drwx--x--x 141 ashley staff 20480 2008-07-22 15:17 ../
```

```
drwxr-xr-x  4 ashley staff 4096 2008-07-22 15:39 project_example/
```

Lets delete the entire contents of the project_examples area

```
ashley@uggp:~/svn_checkout:757> rm -rf project_example
```

Show me the contents of directory

```
ashley@uggp:~/svn_checkout:758> ls -al
```

```
total 24
drwxr-xr-x  2 ashley staff  4096 2008-07-22 16:36 ./
drwx--x--x 141 ashley staff 20480 2008-07-22 15:17 ../
```

Which is empty as I expected it.

```
ashley@uggp:~/svn_checkout:759> pwd
```

```
/home/staff/ashley/svn_checkout
```

Lets recheck the contents out from cits3200.csse.uwa.edu.au on the project_example.

```
ashley@uggp:~/svn_checkout:761> svn co
svn+ssh://ashley@cits3200.csse.uwa.edu.au/home/cits3200/cits3200xx/svnroot/project_example
```

ashley@cits3200.csse.uwa.edu.au's password:

```
A project_example/fileB.txt
A project_example/sample
A project_example/sample/fileC.txt
A project_example/fileA.txt
Checked out revision 3.
```

```
ashley@uggp:~/svn_checkout:762> cat project_example/fileA.txt
Hello
```

```
ashley@uggp:~/svn_checkout:763> cat project_example/sample/fileC.txt
Demo
```

As you can see it has checked out the entire contents plus the changes and that conclude the crash course on Subversion.

Last piece of advice read the inline help tools for subversion ie on a linux terminal type "man svn" or "svn help" which is probably the better one for explanations. It will give a breakdown on what it can do ie add, checkout (co), delete, checkin (ci), merge and host of of other operations associated with SVN.

