

Subversion has an excellent manual available at <http://svnbook.red-bean.com>. You should look over it, particularly the first few sections if you're not sure what the point is in version control at all. The debian build in netlab is 1.5.x, current production is 1.6., The BIT web repository is 1.4. There are major changes between some of these revisions!

There are two papers available for you to enjoy ☺

The first is a relatively old article from Dr. Dobbs Journal.
The second is from a Ganssele book – Embedded Systems World Class Designs. The first is on my module web page, the second has ben handed out on paper.

Getting started.

Split into groups. I'd suggest your partner from the MARCO assignment (or any other), but it would be useful to have some code to manage!

Choose a group member to own your repository. Only they will execute these commands. Keep in mind that an empty repository will be about 1MB.

You need only create your repository in a place accessible to your group. I'd recommend something hidden and inoffensive, such as `./mysvn`.
Create the repository:

```
umask 002
svnadmin create --fs-type fsfs ~/.mysvn
```

The umask is to ensure the repository is group writable. You should also check that your directory has group execute permission.

Now you need something to add.

In your home directory, create a base directory holding your code. I'm playing with the xv-3 source distro here.

```
mkdir repos
cp -a xv-3.10a repos
svn commit repos file:///home/irj/.mysvn/ -m "Initial
Import"
```

```
Adding      repos/xv-3.10a
Adding      repos/xv-3.10a/xvrle.c
Adding      repos/xv-3.10a/xvtext.c
Adding      repos/xv-3.10a/xvpds.c
Adding      repos/xv-3.10a/jpeg
Adding      repos/xv-3.10a/jpeg/filelist.doc
Adding      (bin)  repos/xv-3.10a/jpeg/testimg.gif
```

```
...
...
Adding      repos/xv-3.10a/c2b
Adding      repos/xv-3.10a/vprintf.c
Committed revision 1.
```

Note the comment after `-m`. If this is omitted `svn` starts `$EDITOR` for you to enter your comment.

You should ALWAYS comment repository changing events.

Everyone using the repository will need access permissions. If you want to try the apache managed one on the faculty server the repository is CSITEST and password TSETISC. This repo is a) backed up b) password protected (if desired) c) accessible globally. Ask at the help desk for a repo for your group for projects or assignments.

Also note that unix file permissions will need to be correct for multiple users to access the same repo. It may well be worth aliasing `svn` to set `umask` first for the repo owner.

```
svn co file:///home/irj/.mysvn/ xv-3.10a
A    xv-3.10a/xv-3.10a
A    xv-3.10a/xv-3.10a/xvrle.c
...
...
Checked out revision 1.
```

The above command will check out (`co`) the specified file or directory (relative to `repos`). Note the space between the required directory and the repo URL.

```
cd xv-3.10a/
touch foo
svn add foo
```

Note: as you're in a working copy, subversion understands the context of the command so there is no need to specify which project or repository you mean.

```
svn status
```

Will tell you the current status of the working copy.

```
svn ci foo -m "The worksheet said so - so there"
```

Will commit the change

If other users go into their working copy and type

```
svn update
```

Then you will all be in sync!

```
svn log
```

Will list the revision history

Using the web based repository

1. Check out the current repository:

```
svn co http://164.11.131.73/svn/CSITEST <your target  
working directory>
```

2. Add a file to the working directory. Lets see who gets this far by giving it your username.

```
touch irjohnso
```

```
svn add irjohnso
```

```
svn ci irjohnso -m "My username file"
```

3. Browse the repository with the web browser of your choice!

4. To ensure you're in sync:

```
svn update
```

5. To see revision history

```
svn log
```