

## How to install svnserv on Snow Leopard

### Abstract:

Source Code Management (**SCM**) is very important in Software Development. If two or more developers are working on the same project at the same time, then its very important that they should maintain one version control system to sync with each other to avoid some simple but irritable problems during the software development.

Here I am going to explain how to configure **svnserv** on server machine and how to create a repository for maintaining source code and how to configure SCM for a project and how to do check ins/checkouts ..etc on MAC OS 10.6 Snow Leopard.

### System Requirements:

- MAC OS 10.6 Snow Leopard
- Xcode (already installed on the machine)

### svnserv download link:

- <http://www.open.collab.net/downloads/community/>

### Before going to install svnserv:

- Install Xcode on the machine
- Download [Subversion-1.6.12-10.6.x.dmg](#) file

## How to configure svnserv:

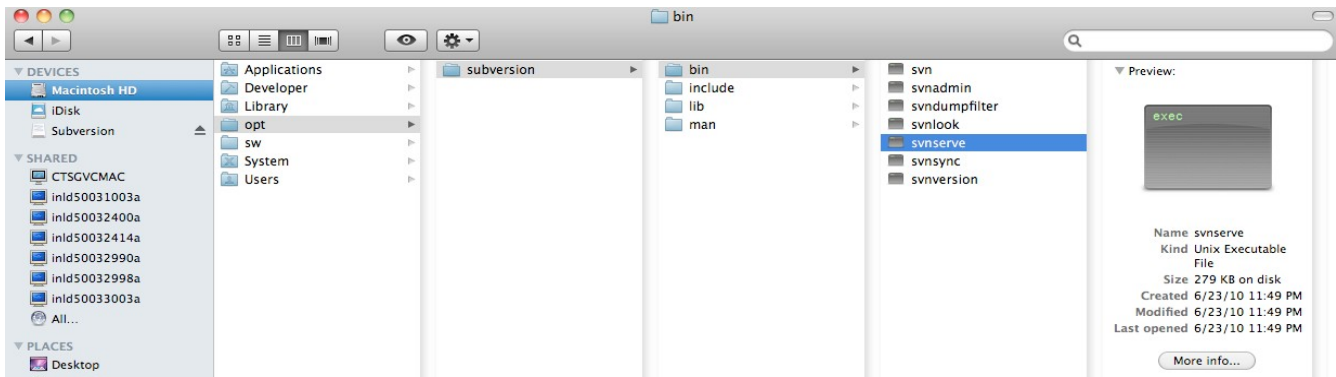
### #1.

Double click on Subversion-1.6.12-10.6.x.dmg file. Then below window will be open. Double click on Subversion-1.6.12 in the below window.



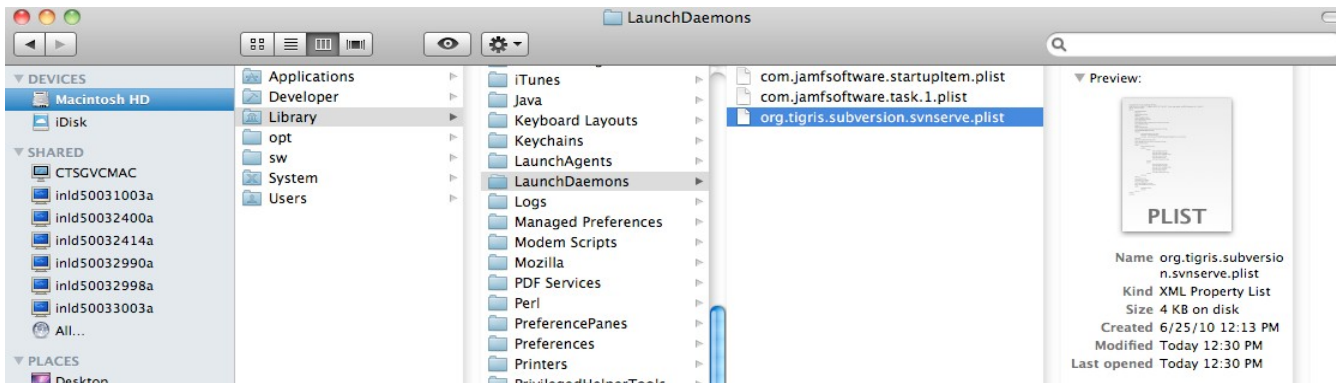
### #2.

Follow the instructions there in the installer and finally **svnserv** will be installed in /opt/subversion folder. See below pic.



### #3.

Create a new file **org.tigris.subversion.svnserve.plist** in **/Library/LaunchDaemons** folder.



### #4:

Open the above file with **TextEdit** and copy the below text into the file and save it.

-----copy from below -----

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"
"http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
<key>Debug</key>
<false/>
<key>Disable</key>
<false/>
<key>GroupName</key>
<string>staff</string>
<key>Label</key>
<string>org.tigris.subversion.svnserve</string>
<key>OnDemand</key>
<true/>
<key>Program</key>
<string>/opt/subversion/bin/svnserve</string>
<key>ProgramArguments</key>
<array>
<string>svnserve</string>
<string>--inetd</string>
<string>--root=/Users/PresentUser/Documents/Repositories</string>
```

```

</array>
<key>ServiceDescription</key>
<string>SVN Version Control System</string>
<key>Sockets</key>
<dict>
    <key>Listeners</key>
    <array>
        <dict>
            <key>SockFamily</key>
            <string>IPv4</string>
            <key>SockServiceName</key>
            <string>svn</string>
            <key>SockType</key>
            <string>stream</string>
        </dict>
        <dict>
            <key>SockFamily</key>
            <string>IPv6</string>
            <key>SockServiceName</key>
            <string>svn</string>
            <key>SockType</key>
            <string>stream</string>
        </dict>
    </array>
</dict>
<key>Umask</key>
<integer>2</integer>
<key>UserName</key>
<string>PresentUser</string>
<key>inetdCompatibility</key>
<dict>
    <key>Wait</key>
    <false/>
</dict>
</dict>
</plist>

```

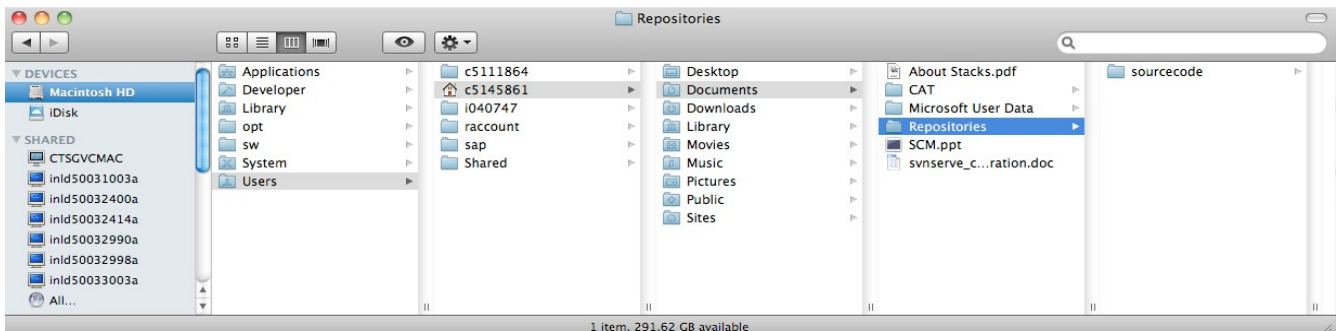
-----copy till here -----

#### Notes:

- The highlighted PresentUser is the present user of the system.
- /Users/PresentUser/Documents/Repositories is the place where **svnserv** expect us to create our repositories
- If Repositories folder is not there in /Users/PresentUser/Documents folder, then create a "Repositories" folder there.

#### #5:

Create an actual repository inside Repositories by executing the below command.  
**svnadmin create /Users/PresentUser/Documents/Repositories/sourcecode**



## #6:

Open `/Users/PresentUser/Documents/Repositories/sourcecode/conf/svnserve.conf` file in TextEdit.

By default the anon-access filed is written as `anon-access = read`. Modify it to `anon-access = none`.

```

### This file controls the configuration of the svnserve daemon, if you
### use it to allow access to this repository. (If you only allow
### access through http: and/or file: URLs, then this file is
### irrelevant.)

### Visit http://subversion.tigris.org/ for more information.

[general]
### These options control access to the repository for unauthenticated
### and authenticated users. Valid values are "write", "read",
### and "none". The sample settings below are the defaults.
# anon-access = none
# auth-access = write
### The password-db option controls the location of the password
### database file. Unless you specify a path starting with a /,
### the file's location is relative to the directory containing
### this configuration file.
### If SASL is enabled (see below), this file will NOT be used

```

## #7:

Open `/Users/PresentUser/Documents/Repositories/sourcecode/conf/svnserve.conf` file in TextEdit.

By default the password-db filed is like below

`# password-db = passwd`

Change it to like below (delete #, delete space between # and password-db)

`password-db = passwd`

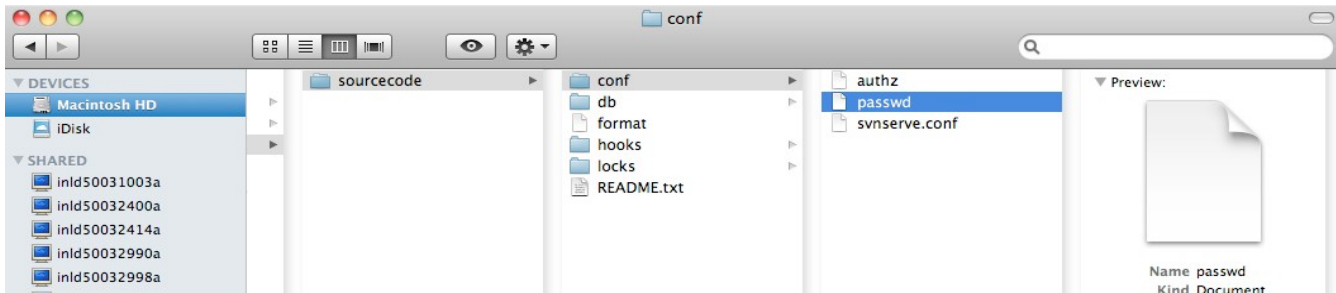
```

### These options control access to the repository for unauthenticated
### and authenticated users. Valid values are "write", "read",
### and "none". The sample settings below are the defaults.
# anon-access = none
# auth-access = write
### The password-db option controls the location of the password
### database file. Unless you specify a path starting with a /,
### the file's location is relative to the directory containing
### this configuration file.
### If SASL is enabled (see below), this file will NOT be used.
### Uncomment the line below to use the default password file.
password-db = passwd
### The authz-db option controls the location of the authorization
### rules for path-based access control. Unless you specify a path
### starting with a /, the file's location is relative to the the
### directory containing this file. If you don't specify an
### authz-db, no path-based access control is done.

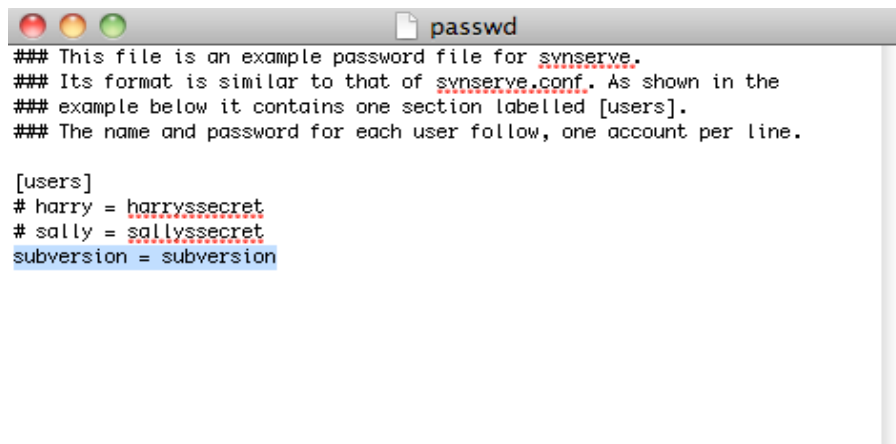
```

### #8:

Open passwd file in TextEdit from ../Repositories/sourcecode/conf/ folder



Simply replace those entries with *username = password* pairs of your own and save your changes, and you should be able to access your repository and the username/password pairs you wrote in this file.



### #9:

Restart the system

### #10:

Execute the below line on Terminal

```
sudo launchctl load /Library/LaunchDaemons/org.tigris.subversion.svnserve.plist
```

If the above command is not working or getting any errors then try the below command

```
sudo launchctl load -F /Library/LaunchDaemons/org.tigris.subversion.svnserve.plist
```

### #11:

Execute the below command

```
sudo launchctl start org.tigris.subversion.svnserve
```

**Now, svnserve is configured.**

## How to configure the SCM repositories:

### #12:

Open **Xcode**

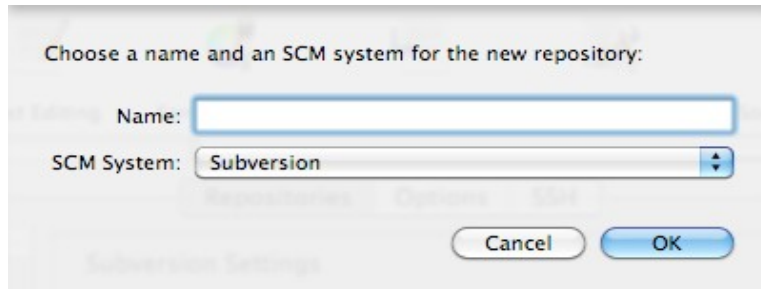
### #13:

Go to **SCM** in the Xcode menu

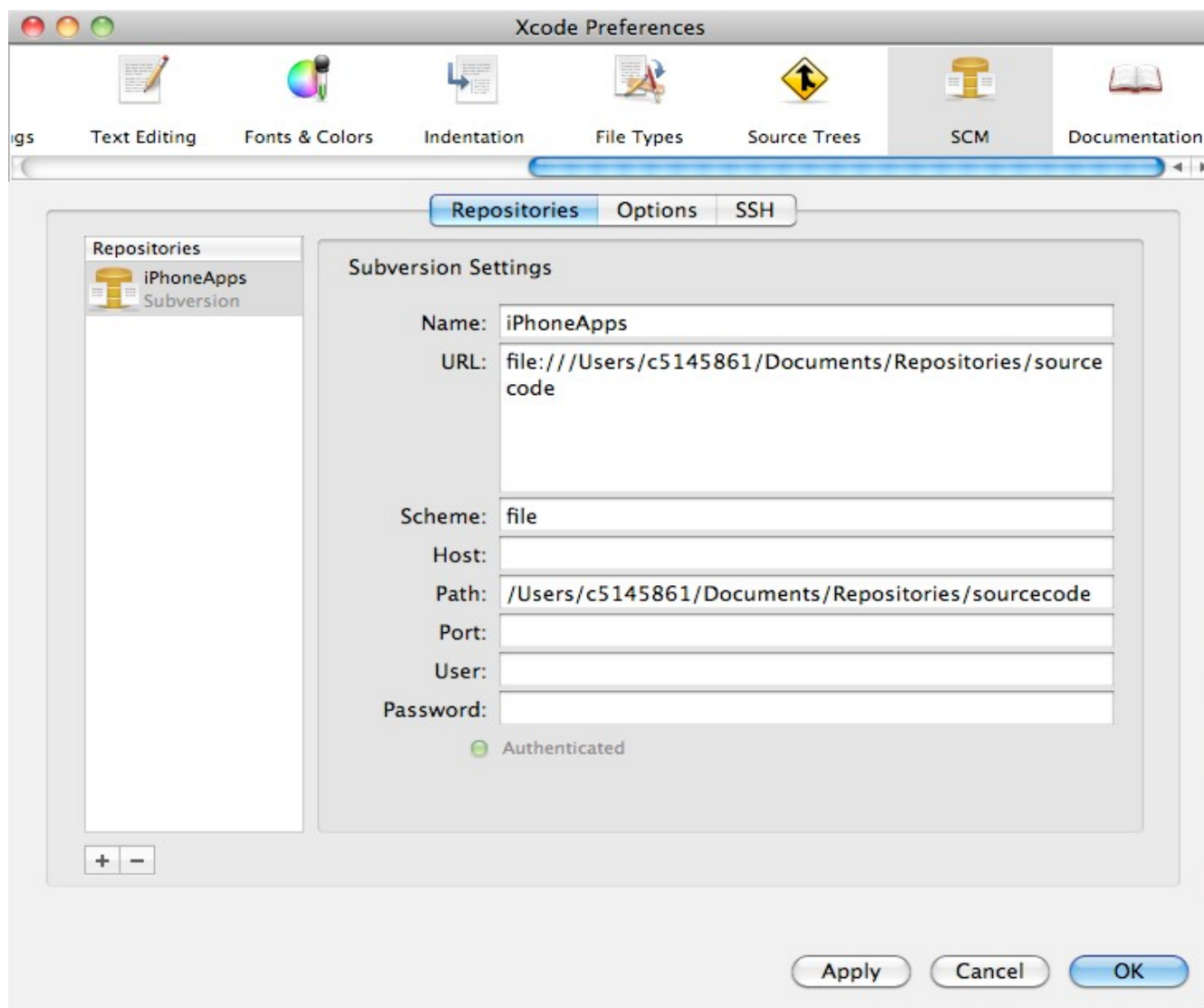


**#14:**

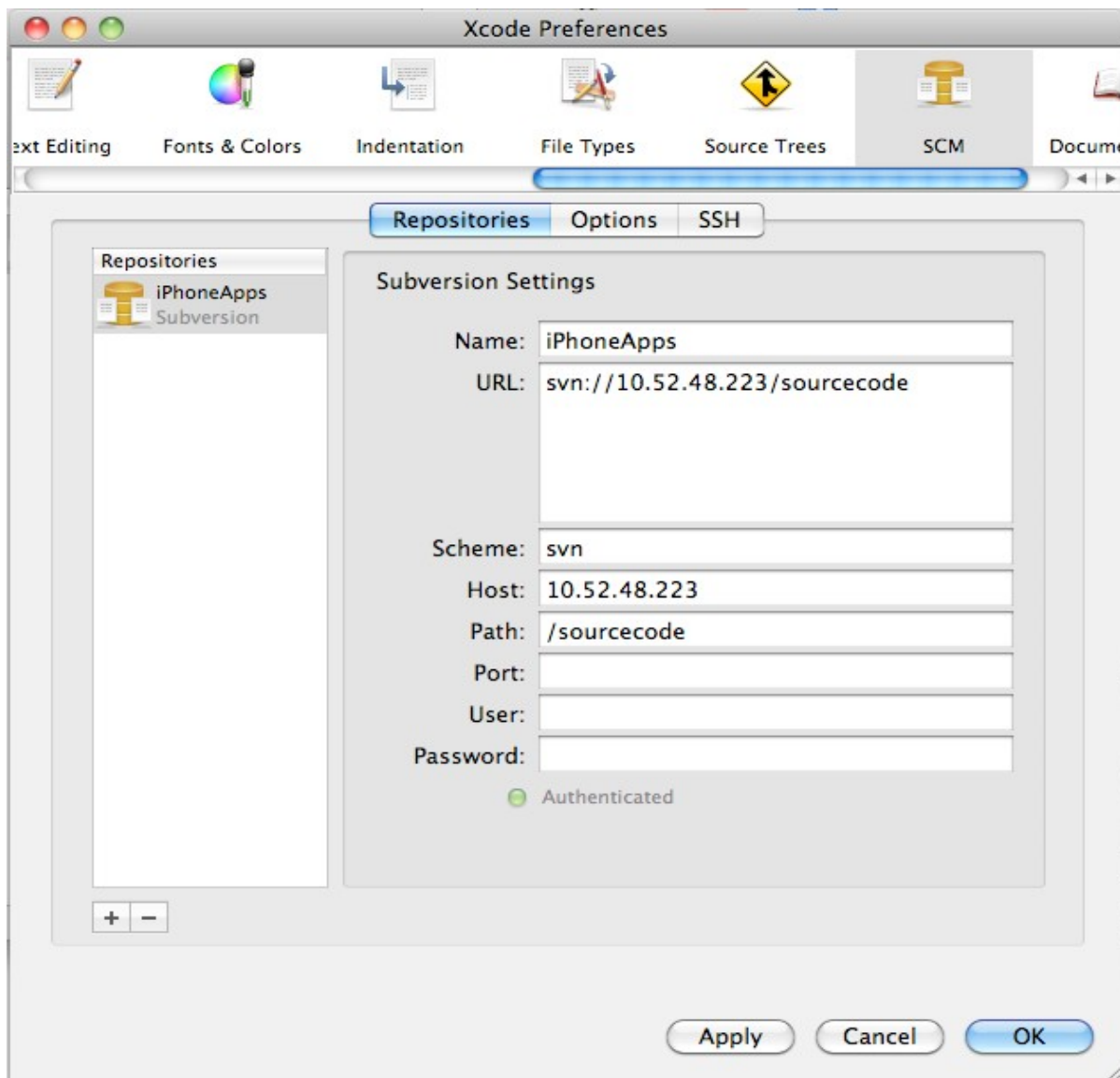
Click on Configure SCM Repositories. Below window will be opened.  
Give the repository name (ex: iPhoneApps) and click on OK button.



**#15: Below window will be opened after clicking on OK button in the above window.**



**For svn (different machine)**



### Enter the following details ...

- Scheme: svn/file
- Host: server IP address (on which repository is created)
- Path: Repository path
- Port:
- Username: username to access the svn server
- Password: password to access the svn server

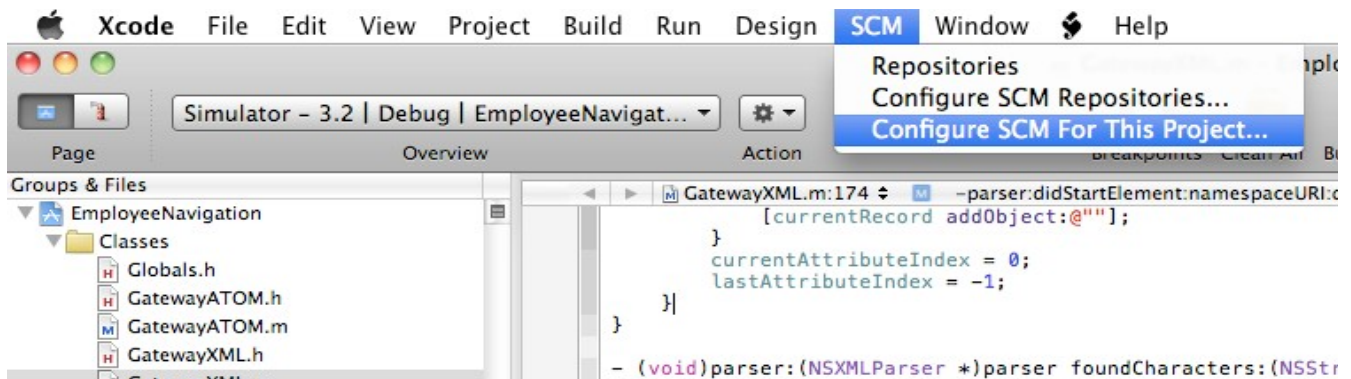
Click on Apply button and see whether the connection is established or not. If the status is "Authenticated" then connection is established. Otherwise there is some error in the settings of the above fields.

Click on OK button.

### How to configure SCM for the current project:

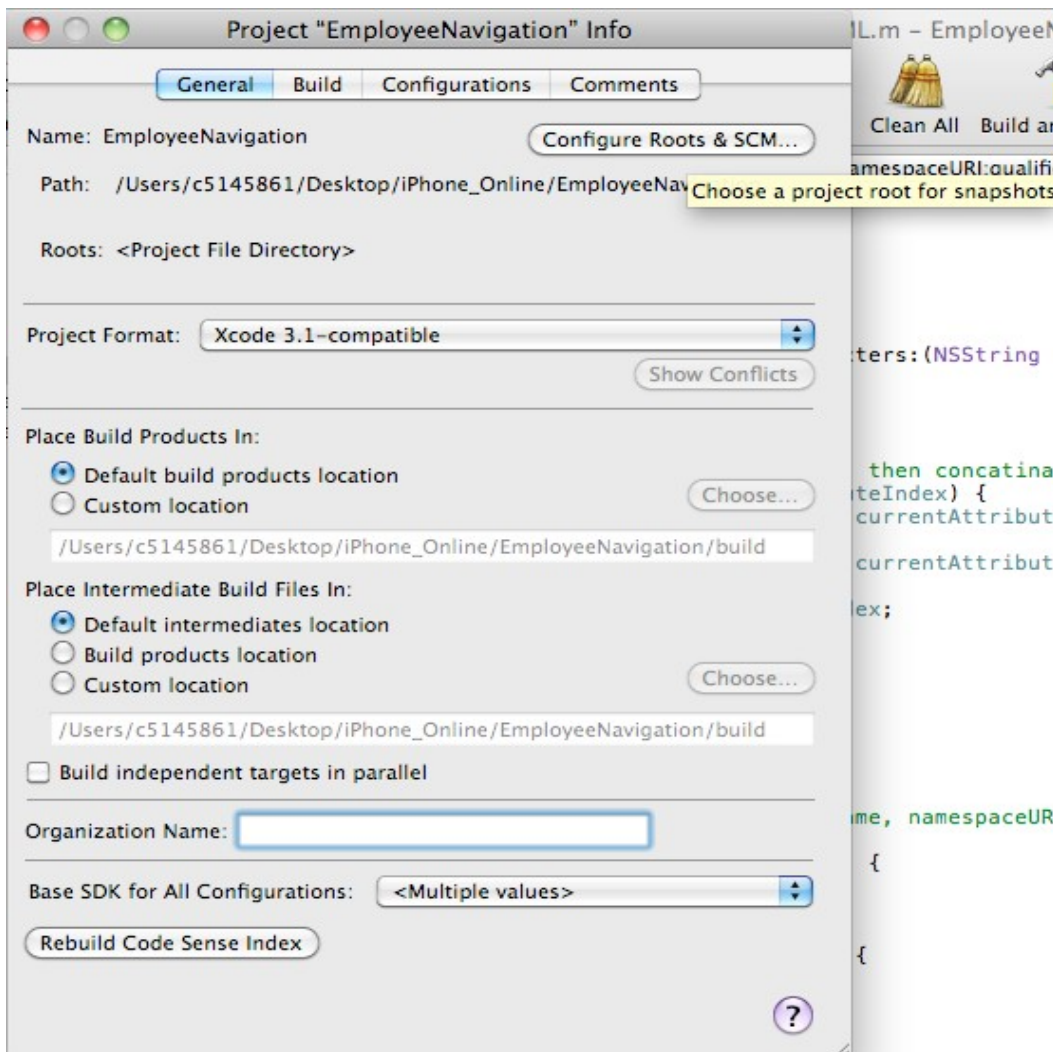
#### #16:

Open any project and Click on **SCM -> Configure SCM For This Project**. See below Pic.



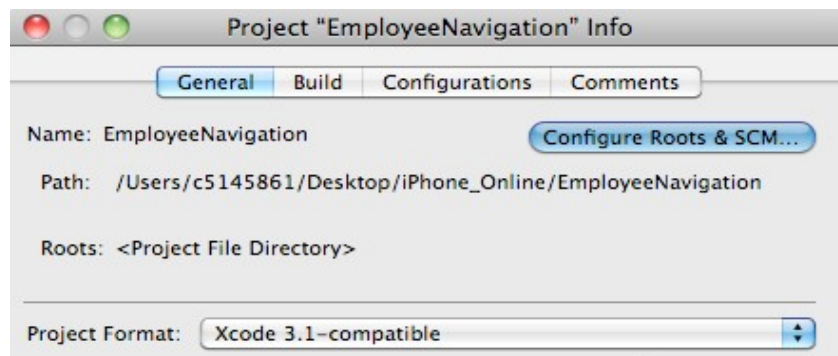
**#17:**

Below window will be opened by clicking on "Configure SCM For This Project" in the above window.

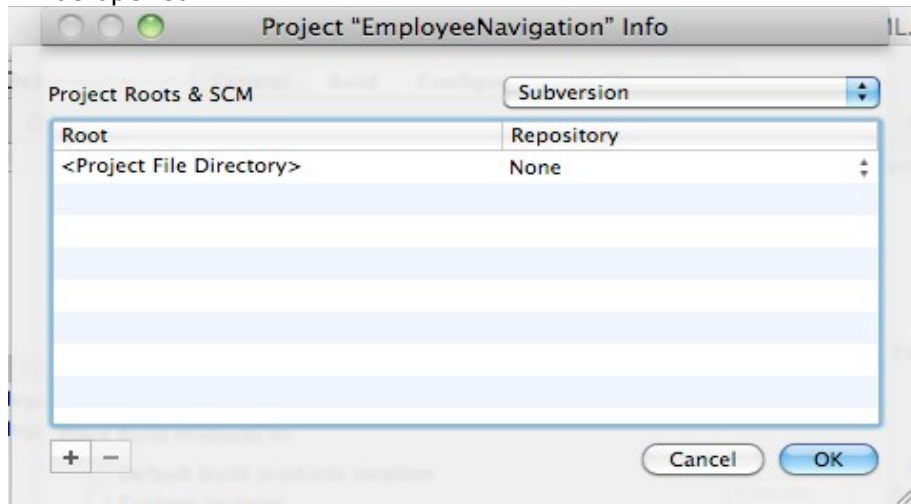


**#18:**

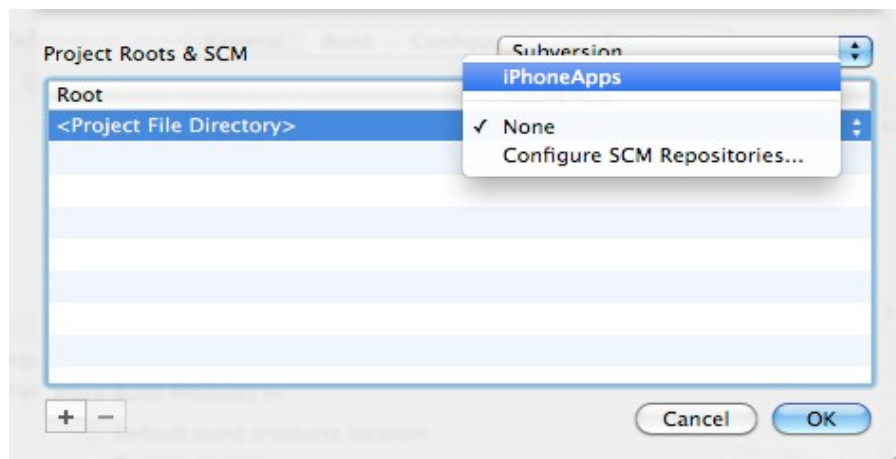
Click on **Configure Roots & SCM.**



**#19:**  
Below window will be opened.



**#20:**  
Click on Repository section (i.e., click on None combo box). Combo Box will open and select already created repository there. In this case, select iPhoneApps. See below Pic.



**Click on OK button.**  
**Now current project is successfully check-in into the repository.**