WHITE PAPER

Setting up Subversion on Mac OS X

Software engineers working in a professional environment typically have a source code repository that helps them keep track of changes. When working on their personal projects, though, many of these same engineers often trust in few repositories for their hard work.

A source code repository is not that hard to set up, and it can save you from disaster. We will discuss setting up the extremely popular Subversion source code control system on a default MacOS X Tiger machine.

If you have not already done so, Install the Apple developer tools, available from connect.apple.com to any developer who has registered. (The online account is free, and gives access to the tools.). Once that is done you are ready to begin.

1. Download the pre-built subversion binaries from <http://metissian.com/projects/macosx/subversion/> and install them.

2. Create the repository

Decide where on your system you wish your subversion archive to live. It will have to be a directory that is always available, so do not put it in a FileVault-protected home directory. If you are planning on using it for the projects for just one user account, put the repository in that user's home directory. The rest of these instructions assume you chose 'subversion-repository' under /Users/work/Documents/coding.

Create the parent, e.g.

```
mkdir /Users/work/Documents/coding
```

Create the repository under the parent

```
svnadmin create --fs-type fsfs /Users/work/Documents/coding/subversion-repository
```

This will create a repository that stores its own files in the file system. You do not have to worry about database locks, freezes, and other database admin problems, but you pay a speed cost under certain conditions. For a personal repository, however, there probably will be no speed problems with a fsfs repository.

3. Edit authz in /Users/work/Documents/coding/subversion-repository/conf

```
vi /Users/work/Documents/coding/subversion-repository/conf
```

You will want this file to contain at least one valid user. To allow anonymous access, and to give the 'alodar' user read and write privileges, make the [/] group within the file look like:

```
[/]
alodar = rw
* = r
```

If, on the other hand, you want to keep anonymous prying eyes away from your code, use

```
        * =
```

4. Edit the passwd file in /Users/work/Documents/coding/subversion-repository/conf

```
vi /Users/work/Documents/coding/subversion-repository/conf/passwd
```

You will need to define a password for each account in this file in the ‘Users’ section:

```
[users]
alodar = alodar-secret-password
```

Please use a password that has mixed case, numbers, misspelled words, and other obfuscations - Keychain Access’ Password Assistant can help suggest decent passwords.

5. Edit the svnserve.conf file in /Users/work/Documents/coding/subversion-repository/conf

```
vi /Users/work/Documents/coding/subversion-repository/conf/svnserve.conf
```

This file describes your environment to the svnserve daemon that will be handling requests. Make the general section look like:

```
[general]
anon-access = read
auth-access = write
password-db = passwd
authz-db = authz
```

If you want your code to be hidden from prying eyes, set

```
anon-access = none
```

6. Create your .subversion directory

```
mkdir ~/.subversion
```

7. Edit the config file as needed

```
vi ~/.subversion/config
```

In your config file, you might want to set the global-ignores to skip over the ‘build’ directory. This is done in the [miscellany] section

In that same section, you should specify to write last-committed timestamps on all files, and to enable automatic property setting on new files added or imported.

```
[miscellany]
global-ignores = *.o *.lo *.la #*# .*.rej *.rej .*~ *~ .#* .DS_Store *.pbxuser *.mode1 *~.nib build
use-commit-times = yes
enable-auto-props = yes
```

You will likely need to add some entries in the automatic property set list.

```
[auto-props]
*.m = svn:eol-style=native;svn:keywords=Author Date URL Rev Id
*.mm = svn:eol-style=native;svn:keywords=Author Date URL Rev Id
*.h = svn:eol-style=native;svn:keywords=Author Date URL Rev Id
*.pch = svn:eol-style=native;svn:keywords=Author Date URL Rev Id
*.pbxuser = svn:mime-type=application/octet-stream
*.java = svn:eol-style=native;svn:keywords=Author Date URL Rev Id
*.plist = svn:eol-style=native;svn:keywords=Author Date URL Rev Id
*.template = svn:eol-style=native;svn:keywords=Author Date URL Rev Id
```

8. Create the xinetd.d startup file.

```
sudo vi /etc/xinetd.d/svn
```

Give it the following content:

```
service svn
{
        disable = no
        socket_type     = stream
        wait            = no
        user            = work
        server          = /usr/local/bin/svnserve
        server_args     = --inetd --root=/Users/work/Documents/coding/subversion-repository
        groups          = yes
        flags           = REUSE IPv6
}
```

While xinetd has been superseded by launchd, you have not yet moved my subversion starter to launchd. We invite anyone who gets it working to drop us a line with the details.

9. Start subversion

```
sudo /sbin/service svn start
```

You should then be able to perform commands on your repository, like

```
svn ls svn://localhost
```

If you want to make changes to your configuration file, or if you need to eradicate the repository and start afresh, be sure to stop the server with

```
sudo /sbin/service svn stop
```

Acknowledgements:

Metissian <www.metissian.com>
The Subversion project <subversion.tigris.org>
Sourceforge, for hosting several subversion repositories for projects that outgrew one machine and one developer.

About Us

Alodar Systems is a custom software and integration consulting firm. We are skilled Java and Cocoa developers with experience in many languages and tools. We can help, whether your challenges are coming from design, architecture, data management, methodology, or good old fashioned code.
We work with some of the biggest names in our industry: Apple, Oracle, BEA, JBoss, ActiveMQ, Sonic, and JetBrains, and we can find the best tools and technologies to solve your problems.

We work with some of the biggest names in our industry: Apple, Oracle, BEA, JBoss, ActiveMQ, Sonic, and JetBrains, and we can find the best tools and technologies to solve your problems.