How-to-install Subversion for MD++ version control in linux computers

Keonwook Kang

October 9, 2007

1 Installation

1.1 Download and Extract

1. Download the latest version of Subversion from http://subversion.tigris.org. Currently (May 31 2007), v.1.4.3 is the latest one.

2. Extract the downloaded file.

```
# mv subversion-1.4.3.tar.gz /usr/local/src
# cd /usr/local/src
# tar -zxvf subversion-1.4.3.tar.gz
```

3. Note that this how-to document is written based on the author's experience with Fedora core 4 linux computer. Check your machine.

```
# uname -a
OS : linux fedora core 4 kernel release 2.6.15-1.1831_FC4
```

1.2 Required Programs or Libraries

1. According to INSTALL file, the prerequisites are listed below.

   (a) APR (Apache Protable Runtime 0.9.7) library
   (b) APR-util 0.9.7 library
   (c) autoconf 2.50 or newer
   (d) libtool 1.4 or newer
   (e) Neon library 0.25.x or 0.26.x
   (f) Berkeley DB 4.X
   (g) Apache Web Server 2.0.49 or newer
   (h) Python 2.0

In the list, Berkeley DB can be replaced with FSFS, “a versioned filesystem implementation that uses the native OS filesystem directly”. According to Subverion book[1], FSFS gives more flexibility but the author just happened to decide to use BDB. The author also chose serf package, a library for HTTP and WebDAV, instead of Neon.

2. Check the version of each required programs.

   ```
   # rpm -qa apr
   apr-0.9.6-3.5
   # rpm -qa apr-util
   apr-util-0.9.6-2
   # rpm -qa libtool
   libtool-1.5.16.multilib2-3
   # berkeley_db_svc -V
   Sleepycat Software: Berkeley DB 4.3.27: (April 25, 2005)
   # apachectl -v
   Server version: Apache/2.0.54
   Server built:   May 23 2005 08:12:24
   # rpm -qa python
   python-2.4.1-2
   ```

   According to the version check, the author had to install APR/APR-util and serf packages.

1.3 Install APR and APR-util

1. Download from http://apr.apache.org/. there are two parallel versions possible (v.1.2.8 and v.0.9.13). What we need is v.0.9.13.[1]

   ```
   # mv apr-0.9.13.tar.gz /usr/local/src/
   # cd /usr/local/src/
   # tar -zxvf apr-0.9.13.tar.gz
   # cd apr-0.9.13
   # ./configure
   # make
   # make install
   ```

2. You will see a directory `/usr/local/apr` generated. There is a binary file `/usr/local/apr/bin/apr-config`, which I made a symbolic link to `/usr/bin`.

[1] Make sure that you do not download and install v.1.2.X, which gave me an error message, “`ELinks: Error reading from socket`” when I started apache web server. Subversion installation guide also says that Subversion needs 0.9.x version of APR. People said this error could result from PHP bug, but I didn't have any problem like this previously when I ran apache 2.0.54 together with php 5.0.4. Even after I installed and ran apache 2.0.59 and php 5.2.2, the problem were not solved. I might screw up something else while I tried to install.

```
# mv /usr/bin/apr-config /usr/bin/apr-config-v.0.9.6
# ln -s /usr/local/apr/bin/apr-config /usr/bin
```

Check the version of installed APR

```
# apr-config --version
0.9.13
```

3. Install APR-util

   ```
   # mv apr-util-0.9.13.tar.gz /usr/local/src/
   # cd /usr/local/src/
   # tar -zxvf apr-util-0.9.13.tar.gz
   # cd apr-util-0.9.13
   # ./configure --with-apr=/usr/local/apr --with-berkeley-db=/usr/bin
   # make
   # make install
   ```

   Here I had to explicity specify the direcotory of APR and Berkeley-db. Otherewise, running `./configure` gave an error message “`configure: error: APR could not be located. Please use the --with-apr option.`”

4. After installing APR-util, you can see a binary file `/usr/local/apr/bin/apu-config` created. I also made a symbolic link to `/usr/bin`

   ```
   # mv /usr/bin/apu-config /usr/bin/apu-config-v.0.9.6
   # ln -s /usr/local/apr/bin/apu-config /usr/bin/
   ```

   Actually, people may not need to download and install APR/APR-util separately, because Subversion website distributes additional dependency package, which has APR/APR-util v.0.9.12 in it. Because I'd like to use the same APR libraries for the Apache web server, I decided to install them separately.

1.4 Install Serf

Download `serf-0.1.1.tar.bz2` from http://code.google.com/p/serf/ and install it.

```
# mv serf-0.1.1.tar.bz2 /usr/local/src/
# cd /usr/local/src
# tar -jxvf serf-0.1.1.tar.bz2
# cd serf-0.1.1
# ./configure
# make
# make install
```

You will see a new directory /usr/local/serf.

1.5 Install Apache Web server and PHP

Initially I didn't expect that I needed to install Apache web server and PHP. To make sure the same APR/APR-util is used in web server and to enable DAV mode, the Apache web server needs to be reinstalled and because of that PHP also has to be reinstalled.

1.5.1 Install Apache web server

1. Download from http://httpd.apache.org/. The latest version is 2.2.4 currently, but I found that the subversion install instruction assumes httpd v.2.0.58 and I didn't want to lose any dependency. So I decided to install httpd 2.0.59.

   ```
   # mv httpd-2.0.59.tar.gz /usr/local/src
   # cd /usr/local/src
   # tar -zxvf /usr/local/src/httpd-2.0.59.tar.gz
   # cd httpd-2.0.59
   # ./configure --enable-dav --enable-so
   # make
   # make install
   ```

   When you configure, the first argument says to build "mod_dav" which is needed to Subversion Web/DAV interface and the 2nd argument enables shared module necessary for loading both PHP module and dav module.

2. After installation, you will see a directory `/usr/local/apache2` made. I made symbolic links for some of binaries in `/usr/local/apache2/bin` such as

   ```
   # mv /usr/sbin/apachectl /usr/sbin/apachectl-v.2.0.54
   # ln -s /usr/local/apache2/bin/apachectl /usr/sbin/
   ```

3. The configuration file `/usr/local/apache2/conf/httpd.conf` is modified such as

   ```
   ServerAdmin root@localhost
   DocumentRoot "/var/www/html"
   <Directory "/var/www/html">
   ```

4. To check the Apache server is working, type

   ```
   # /usr/sbin/apachectl start
   ```

   If you see any error, read the log file `/usr/local/apache2/logs/error_log` to get more detailed about the error.

1.5.2 Install PHP

1. Download PHP 5.2.2 from http://www.php.net/ and install it.

   ```
   # mv php-5.2.2.tar.gz /usr/local/src
   # cd /usr/local/src
   # tar -zxvf php-5.2.2.tar.gz
   # cd php-5.2.2
   # ./configure --prefix=/usr/local/php5
          --with-apxs2=/usr/local/apache2/bin/apxs --with-mysql
   # make
   # make install
   ```

   When configuring, the 2nd argument is needed because otherwise PHP looked old version of apxs. The 3rd argument is needed for MySQL.

2. If you see an error related with a dynamic shared object `libphp5.so` when you try to run the Apache web server, it's prbably because a new `libphp5.so` was not created and PHP still use old one. You can check whether the old version of PHP is running by loading a test php file that has `phpinfo()` from the website. In this case, what you need to do is to clean everying by typing

   ```
   # make clean
   ```

   before configure and to make sure you add `--prefix` when you configure.

3. Copy `php.ini` file

   ```
   # cp /usr/local/src/php-5.2.2/php.ini-dist /usr/local/php5/lib/php.ini
   ```

4. You will see the following line added to `httpd.conf` file.

   ```
   LoadModule php5_module modules/libphp5.so
   ```

   You may want to add the following lines to `httpd.conf`.

   ```
   AddType application/x-httpd-php .php .phtml
   AddType application/x-httpd-php-source .phps
   ```

5. Make a symbolic link.

   ```
   # mv /usr/bin/php /usr/bin/php-v.5.0.4
   # ln -s /usr/local/php5/bin/php /usr/bin/
   ```

6. Restart the Apache web server.

1.6 Install Subversion

1. Finally, you are ready to install Subversion. Type the following commands in order.

   ```
   # cd /usr/local/src/subversion-1.4.3
   # sh ./autogen.sh
   # rm /usr/local/lib/libsvn* /usr/lib/libsvn*
   # ./configure --prefix=/usr/local/svn-1.4.3 --with-apr=/usr/local/apr/bin
             --with-apr-util=/usr/local/apr/bin --with-berkeley-db=/usr/bin
             --with-serf=/usr/local/serf --without-neon
             --with-apxs=/usr/local/apache2/bin/apxs
   # make clean
   # make
   # make install
   ```

   The 2nd command in the above will check if every necessary component is available. If you have any old version of Subversion, remove any library files by the 3rd command. Otherwise, you will meet the following error when running ViewVC, a browser interface for CVS and Subversion version control.

   ```
   Exception: Version requirement not met (needs 1.2.0 or better)
   ```

2. You will see Subversion shared libraries (`mod_authz_svn.so` and `mod_dav_svn.so`) created in `/usr/local/apache2/modules/`. Also check the following lines are added to `httpd.conf` file.

   ```
   LoadModule dav_svn_module     modules/mod_dav_svn.so
   LoadModule authz_svn_module   modules/mod_authz_svn.so
   ```

3. Add the following at the bottom of `httpd.conf` file and modify the location directive appropriately.

   ```
   <Location /svn/repository>
       DAV svn
       SVNPath /absolute/path/to/repository
   </Location>
   ```

   For example, the locattion directive is `<Location /svn/MD++>` in our Subversion configuration. You can declare a parenth path so that you have multiple repositories under the parent direcotry as shown below.

   ```
   <Location /svn>
       DAV svn
       # any "/svn/foo" URL will map to a repository /usr/local/svn/foo
       SVNParentPath /var/www/svn
   </Location>
   ```

4. Also change user/group in `httpd.conf` file like

   ```
   #User nobody
   #Group #-1
   User apache
   Group apache
   ```

   and `chown` and `chgrp` of repository files (and directories, too) to “apache” once you create a repository folder later. Otherwise, you will see the following error when you try to check-in your change.

   ```
   $ svn ci -m "message describing your change"
   svn: Commit failed (details follow):
   svn:
   Can’t create directory ’/var/www/svn/XXX/db/transactions/1-1.txn’:
                                 Permission denied
   ```

1.7 Start Subversion

1. Restart the Apache web server.

   ```
   # apachectl restart
   ```

2. Assume that you want to constuct a repository for a local directory “myproject” and all the files in it. First, you create a repository, *e.g.* `test-repo`.

   ```
   # svnadmin create /var/www/svn/test-repo
   ```

   Here I assumed that I already have a directory `/var/www/svn`.

3. Secondly, I import “myproject” directory.

   ```
   # svn import /tmp/myproject file:///var/www/svn/test-repo/
                              -m "initial import"
   Adding         /tmp/myproject/trunk
   Adding         /tmp/myproject/trunk/sumin_clip.html
   Adding         /tmp/myproject/trunk/topframe2.htm
   Adding         /tmp/myproject/trunk/index.html
   Adding         /tmp/myproject/trunk/sumin.html
   Adding         /tmp/myproject/branches
   Adding         /tmp/myproject/tags

   Committed revision 1.
   ```

4. Change owner and group of repository direcotry.

```
# cd /var/www/svn/test-repo
# chown -R apache ./*
# chgrp -R apache ./*
```

Note that the original `/tmp/myproject` directory will be unchanged no matter what change are done in the subversion copy; Subversion is unaware of it. In fact, you can even delete that directory if you wish. All the data (files and directories) will be databased and cannot be seen by directly peeking into repository.

1.8 Subversion Client-side Configuration

Once you check out repository from Subversion server, Subversion creats a hidden directory `.subversion` in user's home directory. In `.subversion` directory, there are two files `servers` and `config` that a user can configure. For the security reason, uncomment the following line in ~`/.subversion/config`.

```
store-passwords = no
```

For details about per-user configuration, see the Subversion book.[1]

1.9 Install ViewVC

ViewVC is "a browser interface for CVS and Subversion version control repositories". The requrement for Subversion support

- Python 2.0 or later
- Subversion, Version Control System, 1.2.0 or later

and the optional requirement is

- a web server capable of running CGI programs, e.g. Apache
- MySQL 3.22 and MySQLdb 0.9.0 or later to create a commit database
- Enscript, code colorizer
- Highlight, code colorizer, 2.2.10 or later required, 2.4.5 or later. Recommended for reliable line numbering

Between Enscript and Highlight, the author decided to use Enscript which was already installed in Fedora Core 4.

1. Donwload ViewVC from http://www.viewvc.org/ and install it.

   ```
   # mv viewvc-1.0.4.tar.gz /usr/local/src
   # cd /usr/local/src
   # tar -zxvf viewvc-1.0.4.tar.gz
   # cd viewvc-1.0.4
   # ./viewvc-install
   ```

2. Configure `viewvc.conf`. Edit `/usr/local/viewvc-1.0.4/viewvc.conf`.

   ```
   root_parents = /var/www/svn/ : svn
   svn_path = /usr/local/bin/
   address = <a href="mailto:root@localhost">ADMIN</a>
   ```

3. Edit the web browser configuration file `httpd.conf`, which is usually in `/usr/local/apache2/conf/`, to run ViewVC cgi file `/usr/local/viewvc-1.0.4/bin/cgi/viewvc.cgi`. Add the following two lines to `httpd.conf`.

   ```
   ScriptAlias /viewvc "/usr/local/viewvc-1.0.4/bin/cgi/viewvc.cgi"
   ScriptAlias /query "/usr/local/viewvc-1.0.4/bin/cgi/query.cgi"
   ```

4. Create MySQL checkin database and account.

   ```
   # mysql --user=root
   mysql> create database ViewVC;
   mysql> grant all privileges on ViewVC.* to \
          ’XXXXXXXXX’@’localhost’ identified by \
          ’pass’ with grant option;
   mysql> exit
   ```

5. Run the `make-database` script, which creates the database and tables in MySQL used by the ViewVC checkin database. You will be prompted for: database user, database user password, and database name.

   ```
   # cd /usr/local/viewvc-1.0.4/
   # bin/make-database
   MySQL User:
   MySQL Password:
   ViewVC Database Name [default: ViewVC]:
   ```

6. You will then need to set the appropriate parameters in your `viewvc.conf` file under the [`cvsdb`] section.

   ```
   [cvsdb]

   enabled = 1
   host = localhost
   port = 3306
   database_name = ViewVC
   user = XXXXXXXXX
   passwd = pass
   ```

7. Publish Subversion commits to the database.

   ```
   # bin/svndbadmin rebuild /var/www/svn/test-repo
   ```

8. For real-time update, add a post-commit hook.

   ```
   # cd /var/www/svn/test-repo/hooks
   # cp post-commit.tmpl post-commit
   # vi post-commit
   ```

   `post-commit` file should look like

   ```
   #------------------------------------------------------------
   REPOS="$1"
   REV="$2"
   /usr/local/viewvc-1.0.4/bin/svndbadmin rebuild "$REPOS" "$REV"
   #------------------------------------------------------------
   ```

9. Enalbe syntax colorizing. Enscript and Highlight are two programs that can colorize source code for a lot of languages. ViewVC can be configured to use either one. Set either the 'use_enscript' or 'use_highlight' options in `viewvc.conf` to 1.

10. Restart the Apache server. If you see some error related with svn library, try this.

    ```
    # cd /usr/local/svn-1.4.3/lib/svn-python
    # cp -r ./libsvn/ /usr/lib/python2.4/site-packages/
    # cp -r ./svn/ /usr/lib/python2.4/site-packages/
    ```

2 Authentification and Authorization of Access to Subversion Server

2.1 htaccess

A simple way of protecting a directory on your web server is using `htpasswd` utility that comes with Apache.

1. First, you have to generate a password file. For example, let's assume that you are going to generate a password file for the first time for Harry. Then type

   ```
   # htpasswd -cm /usr/local/apache2/passwd/svn-password Harry
   New password: *****
   Re-type new password: *****
   Adding password for user Harry
   ```

   In the command, `-c` is used to create the password file. Next time, you shouldn't add `-c` because it will overwrite preexisting file and you will lose every password stored in the file. `-m` is used for MD5 encryption of the password, which is more secure. Now if you want to add a password for Sally, you type

```
$ htpasswd -m /usr/local/apache2/passwd/svn-passwd Sally
New password: *******
Re-type new password: *******
Adding password for user Sally
```

2. Create a group file

   ```
   # vi /usr/local/apache2/passwd/groups
   ```

   The file looks like

   ```
   GroupName: Harry Sally Puppy
   ```

3. Edit your `httpd.conf`.

   ```
   <Location /svn/test-repo>
       Dav svn
       SVNPath /var/www/svn/test-repo
       AuthType Basic
       AuthName "By Invitation Only"
       AuthBasicProvider file
       AuthUserFile  /usr/local/apache2/passwd/svn-passwd
       AuthGroupFile /usr/local/apache2/passwd/groups
       Require group GroupName
   </Location>
   ```

2.2 Secure HTTP

As an alternate of `.htaccess`, secure HTTP provides a strong encryption via SSL.

1. Check if `mod_ssl` is compiled.

   ```
   # apachectl -l | grep ssl
   ```

   The command `apachectl -l` lists all the compiled modules. If there is no `mod_ssl`, you need to recompile Apache web server to enable ssl-related features.

2. Locate `openssl`.

   ```
   # locate openssl
   /usr/bin/openssl
   ```

3. Reinstall Apache HTTPD.

```
# cd /usr/local/src/httpd-2.0.59
# make clean
# ./configure --enable-dav --enable-so --enable-ssl --with-ssl=/usr/bin
# make
# make install
```

4. Create a server certificate.[2] Here we are going to generate a self-signed certificate and we skip the part creating a key pair.

```
# cd /usr/local/apache2/conf
# openssl req -new -nodes -x509 -days 365 -out server.cert
 -keyout server.key -subj
"/C=US/ST=CA/L=Stanford/CN=micro.stanford.edu/emailAddress=root@localhost"
```

5. Comment out the following in `httpd.conf`

```
<IfModule mod_ssl.c>
    Include conf/ssl.conf
</IfModule>
```

and edit `ssl.conf` such that

```
SSLCertificateFile /usr/local/apache2/conf/ssl.crt/server.cert
SSLCertificateKeyFile /usr/local/apache2/conf/ssl.key/server.key
```

6. Add `SSLRequireSSL` to the location section `<Location /svn/test-repo>` in `httpd.conf` If you have ViewVC configured in `httpd.conf`, modify it such that

```
<Directory /usr/local/viewvc-1.0.4>
    AuthName "Subversion repository"
    AuthType Basic
    AuthUserFile  /usr/local/apache2/passwd/svn-passwd
    AuthGroupFile /usr/local/apache2/passwd/groups
    # Deny access when SSL is not used for the HTTP request
    SSLRequireSSL
    # only authenticated users may access the repository
    Require valid-user
</Directory>
```

7. Start the Apache server[2]

[2] If you compiled and installed Apache yourself, in many of the vendor configuration files, you can see that the SSL directives are surrounded by an `<IfDefine SSL>` block. That allows for conditional starting of the server in SSL mode. If you start the httpd server binary directly, you can pass it the -DSSL flag at startup. if you always want to start Apache with SSL support, you can just remove the ¡ifDefine¿ section and start Apache in the usual way.

```
# apachectl -k start -DSSL
```

8. Make sure that you type `https://` instead of `http://` when you connect to Subversion server.

```
$ svn checkout https://micro.stanford.edu/svn/test-repo/trunk
```

References

[1] http://svnbook.red-bean.com/en/1.4/index.html

[2] http://www.samspublishing.com/articles/article.asp?p=30115&seqNum=4