## Installing the Apache 2 Web Server

This document describes how to install the version of the Apache 2 web server that is located on the class website, version 2.2.22. You may obtain this version either [by clicking here](http://cs-server.usc.edu:45678/download/httpd-2.2.22.tar) or by going to the class web page and clicking on Download Course Software. If you choose to use this version of Apache, you may follow the detailed instructions below. Otherwise you may download another version from [the Apache web site,](http://www.apache.org/) but in that case you will have to follow the installation instructions included with the software. If you need assistance you should read the documentation that comes with the Apache package (see the htdocs subdirectory) or go to the online documentation at<http://www.apache.org/docs/>.

The following procedure describes how to build and install the Apache 2.2.22 source distribution. Once again, this procedure works ONLY FOR the version of Apache located on the course web site, namely 2.2.22.

**Please run your server on cs-server.usc.edu; DO NOT run your server on nunki.usc.edu or aludra.usc.edu, otherwise, your account will be suspended.**

**Download and Unpack Apache 2 Package**

Step 1: Download the 2.2.22 source version of the Apache web server from the class website by clicking [here](http://cs-server.usc.edu:45678/download/httpd-2.2.22.tar) or go to the course home page and click on Download Course Software. Upload the file to your class user account home directory.

Assume your user account home directory is **/home/scf-22/myname/**, and you want to unpack the apache files there.

Step 2: unpack the tar package by running "tar xvf httpd-2.2.22.tar"

Step 3: after several minutes, a directory tree **/home/scf-22/myname/httpd-2.2.22** will be created.

Step 4: you should delete httpd-2.2.22.tar which is no longer needed by running "rm httpd-2.2.22.tar"

**Install Apache 2.2.x Binary**

Connect to the Apache source distribution directory just created and do a "pwd" command to get the full path to the apache directory, as in:

cs-server.usc.edu(2): **cd httpd-2.2.22**

cs-server.usc.edu(3): **pwd**

/home/scf-22/csci571b/httpd-2.2.22

cs-server.usc.edu(4):

**Part 1: Source creation and configuration.**

Then type the following command to create the source files and configure them to your "hardware" environment:

./configure --prefix=**/absolute/path/to/apache/directory**

Note: The directory **/absolute/path/to/apache/directory/** SHOULD BE a new directory, e.g. /home/scf-22/myname/apache2, so that you can erase the distribution directory /home/scf-22/myname/httpd-2.2.22 at the end of this installation process. Therefore we strongly recommend that you use a **new directory** for this installation.

During installation, which could last as long as 10 minutes, you will see hundreds of output lines. Many will look like **checking for \_name\_ yes/no**. At the beginning and end of the source code creation, you should see output similar to the one below (most of the "checking" and "creating" lines have been omitted):

cs-server.usc.edu(5): **./configure --prefix=/home/scf-22/csci571b/apache2**

checking for chosen layout... Apache

checking for working mkdir -p... yes

checking build system type... sparc-sun-solaris2.10

checking host system type... sparc-sun-solaris2.10

checking target system type... sparc-sun-solaris2.10

Configuring Apache Portable Runtime library ...

. . . . . . . . . lots of other messages . . . . . . .

Configuring Apache Portable Runtime Utility library...

Configuring Apache httpd ...

Applying OS-specific hints for httpd ...

Restore user-defined environment settings...

Construct makefiles and header files...

creating Makefile

creating test/Makefile

config.status: creating support/split-logfile

config.status: creating build/rules.mk

config.status: creating build/pkg/pkginfo

config.status: creating build/config\_vars.sh

config.status: creating include/ap\_config\_auto.h

config.status: executing default commands

cs-server.usc.edu(6):

**Part 2: Source compilation and linking.**

Do **not** change directory. Then type the following command to compile all source files and create (i.e. "make" in UNIX lingo) the **httpd** binary file:

**make**

The compilation will **take a long time** (as long a 20 minutes).

**Part 3: Library Linking and installation.**

Once compilation is finished, you can install the **httpd** binary using:

**make install**

This phase will last about 3 minutes. You will see several lines of the type "Making..." and "mkdir...". With many of those lines removed, you will see output similar to the one below:

cs-server.usc.edu(7): **make install**

Making install in srclib

Making install in apr

mkdir /home/scf-22/csci571b/apache2

mkdir /home/scf-22/csci571b/apache2/lib

mkdir /home/scf-22/csci571b/apache2/bin

mkdir /home/scf-22/csci571b/apache2/build

mkdir /home/scf-22/csci571b/apache2/lib/pkgconfig

mkdir /home/scf-22/csci571b/apache2/include

----------------------------------------------------------------------

Libraries have been installed in:

/home/scf-22/csci571b/apache2/lib

If you ever happen to want to link against installed libraries

in a given directory, LIBDIR, you must either use libtool, and

specify the full pathname of the library, or use the `-LLIBDIR'

flag during linking and do at least one of the following:

- add LIBDIR to the `LD\_LIBRARY\_PATH' environment variable

during execution

- use the `-RLIBDIR' linker flag

See any operating system documentation about shared libraries for

more information, such as the ld(1) and ld.so(8) manual pages.

----------------------------------------------------------------------

Installing configuration files

Installing HTML documents

Installing icons

Installing CGIs

mkdir /home/scf-22/csci571b/apache2/cgi-bin

Installing header files

Installing build system files

Installing man pages and online manual

cs-server.usc.edu(8):

If you have reached this point successfully, you now have a ready-to-customize binary. Congratulations!

**Configure/Customize Apache**

Apache 2 uses several configuration files. We will need to change two (2) of these configuration files before starting the Apache 2 web server.

Assuming you have followed the instructions above, the configuration files are contained in the folder **/home/scf-22/myname/apache2/conf** and its subfolders. Change to that directory:

cs-server.usc.edu(9): **cd /home/scf-22/csci571b/apache2/conf**

**Part 1: Edit main configuration file, httpd.conf**

Open the file **httpd.conf** using a text editor. **PLEASE** examine the directives shown below and make sure they are properly set. If you followed the instructions above, you will not need to modify these **ServerRoot** and **DocumentRoot** settings. However, you must verify the paths in the configuration file are correct.

o ServerRoot **/absolute/path/to/apache/directory**

o DocumentRoot **your\_document\_root**

[The default setting is /absolute/path/to/apache/directory/htdocs/, e.g. /home/scf-22/myname/apache2/htdocs/. However, you can change it to any other directory you like.]

The following set of directives must be changed:

o Listen **port\_selected** (for port\_selected replace port 80 using the last 5 digits of your class ID as the base for this value)

o ServerAdmin **your\_email**

o Include conf/extra/httpd-mpm.conf *(You need to "uncomment" this Include line i.e., remove the '#' sign).*

**Part 2: Edit the MPM configuration file, conf/extra/httpd-mpm.conf**

Several of the directives in this file are quite important as ISD has requested that each of your servers take up only a minimum amount of resources. This file is located in the **extra**sub-folder of the **config** directory [for example, /home/scf-22/csci571b/apache2/conf/extra]. Open the file with a text editor, and make sure to make all of the following changes:

o MinSpareServers 2

o MaxSpareServers 2

o StartServers 2

o MaxClients 2

After the changes, the section of the httpd-mpm.conf configuration file will look as follows:

# StartServers: number of server processes to start

# MinSpareServers: minimum number of server processes which are kept spare

# MaxSpareServers: maximum number of server processes which are kept spare

# MaxClients: maximum number of server processes allowed to start

# MaxRequestsPerChild: maximum number of requests a server process serves

<IfModule mpm\_prefork\_module>

StartServers 2

MinSpareServers 2

MaxSpareServers 2

MaxClients 2

MaxRequestsPerChild 0

</IfModule>

You have now completed all the steps to configure, create, install and customize Apache 2. **Congratulations!**

**Starting and Stopping Apache**

Students MUST run their server on cs-server.usc.edu. To start the server, change to the Apache 2 *bin* directory (e.g., /home/scf-22/csci571b/apache2/bin, in the example above) and type **apachectl -k start**. The following ouput should be displayed:

cs-server.usc.edu(10): **apachectl -k start**

cs-server.usc.edu(11):

It is a good practice to check the error log file, e.g. /home/scf-22/csci571b/apache2/logs/error\_log, to ensure your server is properly running. You may see something similar to this:

[Thu Jul 26 11:07:42 2012] [notice] Apache/2.2.22 (Unix) configured -- resuming normal operations

You can also check that the Apache 'httpd' processes are running, using the 'ps' UNIX command:

cs-server.usc.edu(11): **ps -gx**

PID TT S TIME COMMAND

2354 ? S 0:01 /usr/lsd/openssh/default/sbin/sshd -R

25167 ? S 0:00 **/home/scf-22/csci571b/apache2/bin/httpd -k start**

25168 ? S 0:00 **/home/scf-22/csci571b/apache2/bin/httpd -k start**

25169 ? S 0:00 **/home/scf-22/csci571b/apache2/bin/httpd -k start**

2356 pts/70 S 0:00 -tcsh

25389 pts/70 O 0:00 ps -gx

cs-server.usc.edu(12):

If Apache 2 started successfully, you will see running the 3 highlighted processes above.

Similarly, to stop the server, in director *bin*, type **apachectl stop**. The following output should be displayed:

cs-server.usc.edu(12): **apachectl stop**

**Testing Your Server**

By issuing the URL

http://cs-server.usc.edu:**port\_selected**/

you should see the Apache default home page below:

# It works!

You should also test one of the sample files in the cgi-bin directory. Both files, printenv (a Perl script) and test-cgi (a shell script), are shipped with incorrect file protection. Remember to allow EVERYONE to have EXECUTE rights (permissions) to your cgi-bin directory and scripts (e.g., chmod go+x test-cgi). You can test it by issuing the URL:

http://cs-server.usc.edu:**port\_selected**/cgi-bin/test-cgi

The output should be similar to this:

CGI/1.0 test script report:

argc is 0. argv is .

SERVER\_SOFTWARE = Apache/2.2.22 (Unix)

SERVER\_NAME = cs-server.usc.edu

GATEWAY\_INTERFACE = CGI/1.1

SERVER\_PROTOCOL = HTTP/1.1

SERVER\_PORT = 33559

REQUEST\_METHOD = GET

HTTP\_ACCEPT = text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8

PATH\_INFO =

PATH\_TRANSLATED =

SCRIPT\_NAME = /cgi-bin/test-cgi

QUERY\_STRING =

REMOTE\_HOST =

REMOTE\_ADDR = 159.83.137.209

REMOTE\_USER =

AUTH\_TYPE =

CONTENT\_TYPE =

CONTENT\_LENGTH =

**Finishing Up Your Web Server Setup**

If you installed Apache 2 in a directory other than the source distribution, as we recommended, you can now safely delete the distribution directory, its sub-directories and all its files with:

cs-server.usc.edu(12): **/usr/bin/rm -r ~/httpd-2.2.22**

**Useful Links for Compiling, Installing and Configuring the Apache Web Server**

Apache 2 Installation - <http://httpd.apache.org/docs/2.2/install.html>   
Apache 2 Server Documentation - <http://www.apache.org/docs/>