

# Apache 2.0 Installation Guide

**Ryan Spangler**

[spanglerrp22@uww.edu](mailto:spanglerrp22@uww.edu)  
<http://ceut.uww.edu>

May 2002



Department of Business Education/  
Computer and Network Administration

Copyright © Ryan Spangler 2002

## TABLE OF CONTENTS

Introduction .....	1
Installation Options .....	1
Download the Software .....	2
What you need .....	2
Binary Installation .....	2
RPM Installation .....	3
Build from Source .....	4
Starting Apache .....	5
Customize .....	5
Restarting Apache .....	6

# Installing Apache

This guide explains what Apache HTTP (Web) server is and how to install it. We will be talking about the installation and configuration on a Linux system.

## Introduction

The Apache Web server is arguably the most popular Web server in use on the Internet today. Here are some of the reasons why Apache is so popular; you don't have to be running Windows to run Apache. It was developed on various Unix/Linux/BSD platforms, and then recently ported to Win32. Internet Information Server, a Web server made by Microsoft for the Windows NT platform, is made for use in the "Windows-only" world. While IIS has many features, it's lack of portability limits it's market share.

Another reason for Apache's widespread acceptance is its overall stability. While you can slow down an Apache Web server, you can rarely, if ever, kill one. The Apache Web server service is near bulletproof.

Lastly, it's relatively fast. I say "relatively" as it's relative to what you're doing with it. If you're hosting a pretty plain Web site with mostly static content, Apache is a fireball. If you throw tons of CGI scripts at it, while making database calls at the same time, you're going to slow it down. Though much of the slowdown will come from your scripts themselves, and not from Apache.

Apache evolved out of the NCSA httpd project to its current status as one of the fastest, most efficient, and most functional web servers in existence today.

Apache has been shown to be substantially faster, more stable, and more feature-full than many other web servers. Although certain commercial servers have claimed to surpass Apache's speed (it has not been demonstrated that any of these "benchmarks" are a good way of measuring WWW server speed at any rate). The developers of Apache feel that it is better to have a mostly-fast free server than an extremely fast server that costs thousands of dollars. Apache is run on sites that get millions of hits per day, and they have experienced no performance difficulties.

Apache is run on over 10 million Internet servers (as of May 2002). It has been tested thoroughly by both developers and users. The Apache Group maintains rigorous standards before releasing new versions of their server, and their server runs without problems on over one half of all WWW servers available on the Internet. When bugs do show up, they release patches and new versions as soon as they are available.

## Installation Options

I'll cover three methods of installing Apache under Linux. The three methods are:

- *Binary installation*
- *Using an RPM (Red Hat Package Manager)* – recommended for people running Red Hat Linux
- *Building from source*

I recommend using the RPM installation method. This is the easiest way to install Apache for people that are new to Linux.

## Download the Software

You can download Apache from the Apache Software Foundation web site located at <http://www.apache.org>, in source and binary forms. While your downloading Apache, you may want to browse over the documentation.

## What you need

To install Apache, you will need the following things:

1. A computer running Linux
2. Root access on this computer
3. For binary and source installations, the `tar` and `gunzip` Unix utilities

## Binary Installation

A binary is pre-configured, which means someone else has gone to the trouble of configuring and building the software for you. There are, however, a few things you should keep in mind:

- Binaries are compiled for a particular operating system. In other words, you must use a binary built specifically for FreeBSD on your FreeBSD machine and a Linux binary on your Linux machine. You need to be sure to grab the correct binary; if you don't see a binary for your particular operating system, you must choose a different method of installation.
- Apache binaries are usually a version or two behind the current source distribution. This means you don't reap the benefits of the latest bug fixes and feature enhancements.
- Because binaries are pre-configured, you don't have much opportunity to alter the way the software works. If you're a newcomer, you may not care about this loss of flexibility. Fortunately most Apache binaries include a full source distribution, providing you with the best of both worlds -- play now, learn later.

Now let's install a binary. Point your browser at <http://www.apache.org/dist/httpd/binaries/> and download the binary for your operating system (in our case, Linux). You'll most likely be presented with a directory containing multiple versions of Apache in various compressed forms. For the purposes of this guide, I'll assume you've downloaded the gzip'd form of the latest 2.0.x Apache binary (currently that's `httpd-2.0.35-i686-pc-linux-rh72.tar.gz`). If there is a README associated with the file you're downloading, you may want to review it for any interesting installation tidbits or possible bugs.

If you can't find a binary for your operating system, choose either the RPM Installation (if you are running Red Hat Linux) or Build from Source method.

Now let's uncompress the archive using the handy combination of `gunzip` and `tar`. You should replace the "`httpd-2.0.35-i686-pc-linux-rh72.tar.gz`" text below with the name of the `gzip`'d file you downloaded.

```
gunzip < httpd-2.0.35-i686-pc-linux-rh72.tar.gz | tar xvf -
```

Some of you may be lucky enough to have a version of `tar` that is capable of taking care of both tasks.

```
tar xvzf httpd-2.0.35-i686-pc-linux-rh72.tar.gz
```

Either way, you should end up with an `httpd-2.0.x` directory, with `x` being the particular sub-version of Apache you downloaded. Move into the newly created directory.

```
cd httpd-2.0.x
```

As of Apache 1.3.11, binary distributions contain an install script called `install.bindist.sh`. If your binary does not seem to contain such an install script, take a look at the `README.bindist` and/or `INSTALL.bindist` documents for further information; if these documents don't outline a simple installation method, you'll probably want to use one of the other methods I cover in this guide.

If you've not already done so, you'll need to become root. You can become root by typing `su`, then the root password. Then go ahead and run the install script.

```
./install.bindist.sh
```

This command should install the various bits of the Apache distribution into the appropriate locations; the default is usually to install everything under `/usr/local/apache` (confirm this by consulting your `README.bindist` and/or `INSTALL.bindist` documents).

That's all there is to it on installing binaries.

## RPM Installation

Those of you running Red Hat Linux may want to take advantage of Red Hat's RPM ("Red Hat Package Manager") system. Almost identical to a binary, an RPM is further customized to play nicely with other RPMs and provide a consistent interface to installing, updating, and removing binaries. For Linux newcomers or when installing a small standard component, RPMs are simple and reliable.

Bear in mind that an Apache RPM may already be installed on your system depending on how Linux was originally installed on your computer. To find out, at the shell prompt, type:

```
rpm -qa | grep apache
```

If you see something like `apache-1.3.9xxx`, an Apache RPM has already been installed. You can also type that command typing `httpd` instead of `apache` to see if it's installed.

If you don't have an Apache RPM, you must obtain one. Red Hat 7.3 ships `apache-1.3.23-11.src.rpm` in the `RedHat/RPMS` directory on the installation CD. Or, point your browser at <ftp://ftp.redhat.com/pub/redhat/redhat-7.3-en/os/i386/RedHat/RPMS> and download it. If you've not already done so, you'll need to become root. Navigate to the same directory as the `.rpm` file

you obtained, and then type the following command, substituting the name of the .rpm you're using for example: `apache-1.3.23-11.src.rpm`.

```
rpm -ivh apache-1.3.23-11.src.rpm
```

RPM should grind away, displaying its progress with a primitive ##### progress bar. Barring any errors, you're done.

## Build from Source

Building Apache from source may seem like a daunting task to newcomers, but the Apache developers have done a wonderful job of making the task about as simple as could be. Just three more commands than a binary installation and you skip the arduous task of figuring out which binary is the right one for your particular operating system.

Point your browser at <http://www.apache.org/dist/httpd/> and download the gzip'd form of the current version of Apache (2.0.36 at the time of this writing).

Now let's uncompress that archive using `gunzip` and `tar`. You should replace the `httpd-2.0.36.tar.gz` below with the name of the gzip'd file you downloaded.

```
gunzip < httpd-2.0.36.tar.gz | tar xvf -
```

You should end up with an `httpd-2.0.x` directory, `x` being the particular sub-version of Apache you downloaded. Move into the newly created directory.

```
cd httpd-2.0.x
```

Now we'll use the `configure` and `make` commands to configure, make, and install Apache. If you've not already done so, now would be the time to become root.

```
./configure
```

Your screen should look something like:

```
# ./configure
checking for chosen layout... Apache
checking for working mkdir -p... yes
checking build system type... i686-pc-linux-gnu
checking host system type... i686-pc-linux-gnu
checking target system type... i686-pc-linux-gnu

Configuring Apache Portable Runtime library ...
...
config.status: executing default commands
```

Unless errors were reported (not warnings), your Apache installation is now configured and we can move on. This is where things get a bit ugly. Make'ing Apache produces screenfulls of output.

```
make
```

Your screen should look something like:

```
# make
Making all in srclib
make[1]: Entering directory `/home/ryan/dl/apache_guide/httpd-2.0.36/srclib'
Making all in apr
make[2]: Entering directory `/home/ryan/dl/apache_guide/httpd-2.0.36/srclib/apr'
...
make[1]: Leaving directory `/home/ryan/dl/apache_guide/httpd-2.0.36'
#
```

Finally, you're ready to install your Apache build.

```
# make install
```

Now Apache is installed.

## Starting Apache

Let's take your new Apache installation out for a spin. If you installed Apache using a binary or from scratch, as root, type:

```
/usr/local/apache/bin/apachectl start
```

If you used an RPM, as root, type:

```
/sbin/service httpd start
```

Point your browser at your brand new Web server, <http://localhost/>. If everything worked you should see the default home page.

## Customize

Apache uses some rather easy to understand text files for configuration. On a Red Hat system, you'll find them in `/etc/httpd/conf`. Quite a few Linux distributions place them in this same place, but if you can't find such a directory, do a search for "httpd.conf". Once you find these, you've found the main config files. If you're new to Linux, and need help finding this file, here's how you can find it.

- 1.Login as root
- 2.Type: `cd /`
- 3.Type: `find -name httpd.conf`

Now you should see where the file is located. When you move into the directory containing `httpd.conf`, you should see these three files:

- *httpd.conf* – This has the settings for the overall configuration for the server.
- *access.conf* – This file contains all the security settings for Apache.
- *srvm.conf* – This file contains the MIME definitions and default document names for files on the server.

## **Restarting Apache**

Whenever you make changes to the server configuration files, such as `httpd.conf`, they won't take effect until the server is restarted. In Linux, Apache can be restarted depending on how you installed it. If you installed Apache using a binary or from scratch, as root, type:

```
/usr/local/apache/bin/apachectl start
```

If you used an RPM, as root, type:

```
/sbin/service httpd start
```

After being restarted the changes will have taken effect.