

## **A. Installing Apache, PHP, and MySQL**

Apache, PHP, and MySQL are available for many combinations of operating systems and web servers. In this appendix, we explain how to set up Apache, PHP, and MySQL on a UNIX platform from (almost) scratch, and offer pointers for installing these technologies on Windows and Mac OS X.

Key topics covered in this appendix include

Running PHP as a CGI interpreter or as a module

Installing Apache, SSL, PHP, and MySQL under UNIX

Installing Apache, PHP, and MySQL using all-in-one installation packages

Installing PEAR

Considering other web server configurations

### **Note**

Detailed information for adding PHP to Microsoft Internet Information Server or other web servers is not included in this appendix. We recommend using the Apache web server when possible. However, pointers to other web server configurations can be found at the end of this appendix.

Our goal in this appendix is to provide you with an installation guide for a web server that will enable you to host multiple websites. Some sites that you create will require Secure Sockets Layer (SSL) for e-commerce solutions, and most sites are driven via scripts to connect to a database (DB) server to extract and process data. Therefore, we include instructions for the common setup of PHP, MySQL, and Apache with SSL on a UNIX machine.

Many PHP users never need to install PHP on a machine, which is why this material is in an appendix rather than [Chapter 1, “PHP Crash Course.”](#) The easiest way to get access to a reliable server with a fast connection to the Internet and PHP already installed is to simply sign up for an account at one of the thousands of hosting services or hosting service resellers around the globe. If this is the route you take, be sure your hosting provider is using up-to-date versions of Apache, PHP, and MySQL, otherwise you will be susceptible to security issues that you cannot control (not to mention you will not be able to use the latest and greatest features within the technologies).

Depending on why you are installing PHP on a machine, you might make different decisions. For example, if you have a machine permanently connected to the network that you intend to use as a live server, then performance will be important to you. If you are building a development server where you can build and test your code, then having a similar configuration to the live server will be the most important consideration.

### **Note**

The PHP interpreter can be run as either a module or as a separate common gateway interface (CGI) binary. Generally, the module version is used for performance reasons. However, the CGI version is sometimes used for servers where a module version is not available or because it enables Apache users to run different PHP-enabled pages under different user IDs.

In this appendix, we primarily cover the module option as the method to run PHP.

## **Installing Apache, PHP, and MySQL Under UNIX**

Depending on your needs and your level of experience with UNIX systems, you might choose to do a binary install or compile the programs directly from their source. Both approaches have their advantages.

A binary install will take an expert minutes and a beginner not much longer, but it will result in a system that is probably a minor version or two behind the current releases and one that is configured with somebody else's choices of options. If you have read the subsequent release changelogs and know what you're missing, or if the build options used by the maintainer of the binary distribution meet your needs, then by all means perform a binary installation.

Although a source install will take additional time to download, install, and configure, and such an approach may be intimidating the first few times you do it, it does give you complete control over the configuration of the technologies. When performing a source installation, you may choose exactly what to install, which version to use, and have complete control over the configuration directives that can be set.

## **Binary Installation**

Most Linux distributions include a preconfigured Apache Web Server with PHP built in. The details of what is provided out of the box depend on your chosen distribution and version.

One disadvantage of binary installs is that you rarely get the latest version of a program. Depending on how important the last few bug fix releases are, getting an older version might not be a problem for you. However, we recommend that if you intend to use the preconfigured binary installations of PHP, MySQL, Apache, and any ancillary libraries, then before you begin using them be sure to update the packages using your distribution's typical update method (e.g., using `apt-get`, `yum`, or other package managers).

The biggest issue with binary installations is that you do not get to choose what options are compiled into your programs. The most flexible and reliable path to take is to compile all the programs you need from their sources. This path will take a little more time than installing packages using a package manager, so we understand you might choose to use binary packages when available, and in fact if you just want a basic configuration, it is likely that the official pre-packaged binaries for your system will meet your needs.

## **Source Installation**

To install Apache, PHP, and MySQL from source under a UNIX environment, the first step is to decide which extra modules you will load under the trio. Because some of the examples covered in this book show the use of a secure server for web transactions, you should install an SSL-enabled server.

For purposes of this book, the PHP configuration is more or less the default setup but also covers ways to enable the `gd2` library under PHP.

The `gd2` library is just one of the many libraries available for PHP. We included this installation step so that you can get an idea of what may be required to build from source and enable extra libraries within PHP. Compiling most UNIX programs follows a similar process, but as you can see as we move through this section, there are times when it is just easier to install precompiled packages anyway.

You usually need to recompile PHP after installing a new library, so if you know what you need in advance, you can install all required libraries on your machine and then begin to compile the PHP module.

Here, we describe installation from source on a Ubuntu server, but the description is generic enough to apply to other UNIX servers as well.

Start by gathering the required files for the installation. You need these items:

Apache (<http://httpd.apache.org/>)—The web server

OpenSSL (<http://www.openssl.org/>)—Open source toolkit that implements the Secure Sockets Layer

MySQL (<http://www.mysql.com/>)—The relational database