

Part I

Movie Review Web Site

Chapter 1: Configuring Your Installation

Chapter 2: Creating PHP Pages Using PHP6

Chapter 3: Using PHP with MySQL

Chapter 4: Using Tables to Display Data

Chapter 5: Form Elements: Letting the User Work with Data

Chapter 6: Letting the User Edit the Database

Chapter 7: Manipulating and Creating Images with PHP

Chapter 8: Validating User Input

Chapter 9: Handling and Avoiding Errors

1

Configuring Your Installation

We assume that since you've spent your hard-earned money to purchase this book, you undoubtedly know the enormous benefits of using Apache, MySQL, and PHP (AMP) together to create and deliver dynamic web sites. But just in case you found this book on your desk one Monday morning with a sticky note from your boss reading "Learn this!," this chapter looks at what makes the "AMP" combination so popular. This chapter also walks you through installing and configuring all three components of the AMP platform on Windows (installation and configuration for Linux-based platforms can be found in Appendix I).

Projects in This Book

You will develop two complete web sites and a few "side projects" over the course of this book:

- ❑ **Movie Review web site:** By developing this site, you will be introduced to the necessary skills to write a PHP program, work with variables and include files, and use data from MySQL. Using PHP and MySQL together makes your site truly dynamic as pages are created on the fly for your visitors. You will also get experience in validating user input while working on this site.
- ❑ **Comic Book Fan web site:** While creating this web site, you'll learn how to build and normalize databases from scratch, manipulate images and send e-mails from PHP. You'll also learn about authenticating users, managing content through a Content Management System, creating a mailing list, setting up an e-commerce section and developing and customizing a discussion forum.

This book also covers how to learn about your visitors through the use of log files and how to troubleshoot common mistakes or problems you will undoubtedly encounter while programming. The appendixes in this book will provide you with the necessary reference materials you'll need to assist you in your web development journey once you complete the book and propose tools to help make you a more efficient coder. After reading this book, you will be able to create a well-designed, dynamic web site using freely available tools.

A Brief Introduction to Apache, MySQL, PHP, and Open Source

There are many open source projects from address books to full-fledged operating systems. Apache, MySQL, and PHP are all open source projects that can be installed on a wide variety of platforms. They are most popular on Linux (giving the acronym “LAMP”) although Windows-based Apache, MySQL and PHP installations are becoming increasingly popular, especially for developers.

The open source movement is a collaboration of some of the finest minds in programming and development, which make up the open source community. The open source movement is defined by the efforts of the community to make for easier development and standardization of systems, applications and/or programs. By allowing the open exchange of source code and other information, programmers from all over the world contribute to making truly powerful and efficient pieces of software usable by everyone. This is the opposite of the closed source model, which is more commonly referred to as “proprietary.” Bugs get fixed, improvements are made, and a good software program can become a great program through the contributions of many people to publicly available source codes.

A Brief History of Open Source Initiatives

The term *open source* was coined in 1998 after Netscape decided to publish the source code for its popular Navigator browser. This announcement prompted a small group of software developers who had been long-time supporters of the soon-to-be open source ideology to formally develop the Open Source Initiative (OSI) and the Open Source Definition. An excerpt taken from the OSI’s web site (www.opensource.org) briefly defines the organization’s objectives:

The OSI is a non-profit corporation formed to educate about and advocate for the benefits of open source and to build bridges among different constituencies in the open-source community. [. . .] One of our most important activities is as a standards body, maintaining the Open Source Definition for the good of the community.

Although the OSI’s ideology was initially promoted in the hacker community, a global base of programmers began to offer suggestions and supply fixes to improve Netscape’s performance upon Netscape’s release of the browser’s source code. The OSI’s mission was off and running and the mainstream computing world began to embrace the idea.

Linux was the first operating system that could be considered open source (although BSD distributed from the University of California Berkeley in 1989 was a close runner-up), and many programs followed soon thereafter. Large software corporations such as Corel began to offer versions of their programs that worked on Linux machines. People soon had entire open source systems, from the operating system right up to the applications they used every day.

Although there are now numerous classifications of OSI open source licenses, any software that bears the OSI certification seal can be considered open source because it has passed the Open Source Definition test. These programs are available from a multitude of web sites; the most popular is www.sourceforge.net, which houses more than 175,000 open source projects!

Why Open Source Rocks

Open source programs are very cool because:

- ❑ **Open source programs are free:** The greatest thing about open source software is that its source code is free of charge and widely available to the general public. This makes it easy for software developers and programmers to volunteer their time to improve existing software and create new programs. Open source software cannot, by definition, require any sort of licensing or sales fees that restrict access to its source code.
- ❑ **Open source programs are cross-platform and “technology-neutral”:** By requiring open source software to be non-platform specific, the open source community has ensured that the programs are usable by virtually everyone. According to the Open Source Definition provided by the OSI at <http://opensource.org/docs/definition.php>, open source programs must not be dependent on any “individual technology or style of interface” and must be “technology-neutral.” As long as the software can run on more than one operating system, it meets that criteria.
- ❑ **Open source programs must not restrict other software:** This means that if an open source program is distributed along with other programs, those other programs are free to be open source or proprietary in nature. This gives software developers maximum control and flexibility.
- ❑ **Open source programs embrace diversity:** The diversity of minds and cultures simply produce a better result. For this reason, open source programs cannot discriminate against any person or group, nor can they discriminate against any field of endeavor. For example, a program designed for use in the medical profession cannot be limited to that profession if someone in another field wants to take the program’s source code and modify it to fit his or her needs.

For a complete list of criteria a piece of software must meet before it can be considered “open source,” or for more information about the OSI and the open source community, visit the OSI web site at www.opensource.org.

How the AMP Pieces Work Together

Now that you’ve learned about some of the spirit and history of open source, it’s important to understand the role Apache, MySQL and PHP play in creating your web site.

Imagine for a moment that your dynamic web site is like a fancy restaurant. Hungry diners come to your place and each one wants something different and specific. They don’t worry so much about how the food is prepared so long as it looks great and tastes delicious. Unlike a buffet spread where everything is laid out and your patrons just pick and choose from what’s available, a nice restaurant encourages interaction between the patron and waiter and complete customization of any meal to meet any specific dietary needs. Similarly, your web site shouldn’t be a static page with little interaction from its visitors; it should be a dynamic site where visitors can choose what they want to see.

Continuing with this scenario, you can characterize the components of the AMP platform as follows:

- ❑ **PHP:** Whatever people ask for, your highly trained master of culinary arts, the chef, prepares it without complaint. She is quick, flexible, and able to prepare a multitude of different types of foods. PHP acts in much the same way as it mixes and matches dynamic information to meet the request for fresh web pages.
- ❑ **MySQL:** Every chef has a well-stocked stockroom of ingredients. In this case, the ingredients used by PHP are records of information stored in MySQL's databases.
- ❑ **Apache:** This is the waiter. He gets requests from the patron and relays them back to the kitchen with specific instructions about how the meal should be prepared. Then he serves the meal once it is complete.

When a patron (web site visitor) comes to your restaurant (web site), he or she sits down and orders a meal with specific requirements (requests a particular page or resource), such as a steak served medium well. The waiter (Apache) takes those specific requirements back to the kitchen and passes them off to the chef (PHP). The chef then goes to the stockroom (MySQL) to retrieve the ingredients (data) to prepare the meal and presents the final dish (web page) back to the waiter, who in turn serves it to the patron exactly the way he or she ordered it.

You can choose to install one, two or all three of the AMP components based on your specific needs. Each is a powerful application in its own right. But the reason the Apache, MySQL, and PHP combination has become so popular is that they work incredibly well together. We obviously recommend that you install all three. You can even benefit from installing them on your development system that is separate from your hosting server. This way, you can develop and test your site in the comfort of your own workspace without having to upload scripts up to the hosting server to test every little change. It also gives you a safe environment to test your code without breaking a live web site.

Installing Apache, MySQL, and PHP on Windows

After following these instructions, you will have successfully installed Apache, MySQL, and PHP on your Windows system. We cover installing them on Windows XP- and Windows Vista-based systems. You should review each component's web site if you want more detailed installation instructions or information on other supported platforms.

- ❑ **Apache:** <http://httpd.apache.org/docs/2.2/platform/windows.html>
- ❑ **MySQL:** <http://dev.mysql.com/doc/refman/5.1/en/windows-installation.html>
- ❑ **PHP:** www.php.net/install.windows

Installing Apache

As your web server, Apache's main job is to listen to any incoming requests from a browser and return an appropriate response. Apache is quite powerful and can accomplish virtually any task that you as a webmaster require.

According to the Netcraft web site (www.netcraft.com), Apache is running over 83.5 million Internet servers, more than Microsoft, Sun ONE, and Zeus combined at the time of this writing. Its flexibility, power, and, of course, price make it a popular choice. It can be used to host a web site for the general public, a company-wide intranet or for simply testing your pages before they are uploaded to a secure server on another machine.

Follow these steps to download and install Apache on your Windows machine (installation instructions can be found in Appendix I):

1. Go to www.apache.org, and click the HTTP Server link in the Apache Projects list. The Apache Software Foundation offers many different software packages, though the HTTP Server is the only one we are concerned with.
2. Click the Download link under the most recent version of Apache.
3. Click the Win 32 Binary (MSI Installer) link to download the installation package. Whether you choose the download without `mod_ssl` or the one that includes OpenSSL depends on your local laws, needs and personal preferences. We do not use any of the functionality offered by `mod_ssl` in this book, so if you want to lean towards the safe side feel free to download the package without `mod_ssl`.

If you experience problems downloading this file, you can try downloading from a different mirror site. Select an available mirror from the drop-down box near the top of the download page.

4. You should be able to double-click the MSI file to initiate the installation wizard for Apache once it has finished downloading, but you may experience some issues depending on what security policies Windows has in effect. We recommend running the installer with administrative privileges from within a console window.

To open a console as an Administrator in Windows XP, navigate through Start ⇨ All Programs ⇨ Accessories, right-click on Command Prompt and select the Run As option. In Windows Vista, navigate through Start ⇨ All Programs ⇨ Accessories, right-click on Command Prompt, and select the Run as Administrator option.

5. Use the `cd` command to navigate to where you downloaded the installer file, and then run the installer using `msiexec -i`. The Installation Wizard will open.

```
cd C:\Users\Timothy\Downloads\  
msiexec -i apache_2.2.9-win32-x86-no_ssl-r2.msi
```

6. After accepting the installation agreement, you will see a screen that is equivalent to a `readme.txt` file — it gives basic information about the Apache software and where to go to find more information. We highly recommend that you read this.
7. Enter the following information on the Server Information screen:
 - Domain name: For example, `example.com`
 - Server name: For example, `www.example.com`
 - Net administrator's e-mail address
 - Whether to install Apache for all users or only the current user.

We recommend the default option, which is to install Apache for all users on port 80 as a service.

8. At the Setup Type screen, the Typical installation option is recommended for beginners and will suffice for most of your needs. If you have special circumstances or are an advanced user, feel free to choose the Custom setup option.
9. You can specify which directory Apache will be installed in on the Destination Folder screen. Again, we recommend the default (C:\Program Files\Apache Software Foundation\Apache2.2), although you may want to change this depending on your needs and your system's configuration.
10. How long it takes for Apache to be installed on your system depends on many factors, but typically it shouldn't take longer than a couple minutes. The wizard will tell you when it has finished, and you can click the Finish button to close the window.
11. To close the console window from which you launched the installation file, you can either click the X in the window's top-right corner or enter `exit` at the prompt.
12. Next, bring up the System Properties window. In Windows XP, this is done by right-clicking on the My Computer icon on your desktop and selecting Properties. In Windows Vista, this is done by right-clicking on the Computer icon on your desktop, selecting Properties, and then selecting Advanced System Settings.
13. Select the Advanced tab, and then click the Environment Variables button. Select PATH from the System variables section and then Edit. Add the path to Apache's bin directory to the end of the existing list (C:\Program Files\Apache Software Foundation\Apache2.2\bin by default). This will allow you to run Apache's utilities from the command line without having to explicitly type the full path each time.

Starting and Stopping Apache

Apache runs as a service waiting for web requests and handling them in the background; you don't interact with it on the desktop like other applications. Instead, you set Apache's options with its configuration file. There are three ways to start and stop the server:

- ❑ **Windows Service Manager:** Go to Start ⇒ Control Panel ⇒ Administrative Tools, and click on the Services icon. Alternatively, you can go to Start ⇒ Run and execute `services.msc`. If you installed Apache as a Windows service for all users (the suggested installation type), then you will see its entry in the listing of services. Just highlight the entry and click the desired action (start, stop or restart).
- ❑ **The `net` command:** The `net` command is used to monitor network related services. Open up a command window that has administrative privileges, and type `net start apache2.2` to start Apache and `net stop apache2.2` to stop Apache.
- ❑ **Apache Service Monitor:** The Apache Service Monitor is installed by default with Apache and typically can be found running in your system tray. If it's not there, then you can find it by going to Start ⇒ All Programs ⇒ Apache HTTP Server 2.2 ⇒ Monitor Apache Servers. Just highlight the server's entry and click the desired action button (start, stop or restart).

Apache only reads its main configuration file once when it starts up, so you will need to restart Apache any time you make changes to its configuration file for those changes to be active.

Testing Your Installation

To test the installation of your Apache server, open a web browser and type the following URL:

```
http://localhost/
```

If the installation was successful then you will see an Apache “success” page in your browser. If not, check your error log by opening the `error.log` file, which you can find in the `logs` subdirectory of Apache’s installation directory (`C:\Program Files\Apache Software Foundation\Apache2.2\logs` by default). By searching through the log file, you can find issues, or maybe an indication of where your installation may have experienced a problem. For a more in-depth discussion of logs, please refer to Chapter 17.

If you had installation problems, note that you might experience some errors such “no services installed” if Apache is trying to share port 80 with another web server or application, such as a locally installed firewall application. To fix this, you can tell Apache to use a different port. Open your `httpd.conf` file in the `conf` subdirectory (`C:\Program Files\Apache Software Foundation\Apache2.2\conf` by default) and locate the following lines:

```
# Listen: Allows you to bind Apache to specific IP addresses and/or
# ports, instead of the default. See also the <VirtualHost>
# directive.
#
# Change this to Listen on specific IP addresses as shown below to
# prevent Apache from glomming onto all bound IP addresses (0.0.0.0)
#
#Listen 12.34.56.78:80
Listen 80
```

Change the last line of this block to read:

```
Listen 8080
```

Then, locate the following lines:

```
# ServerName gives the name and port that the server uses to identify itself.
# This can often be determined automatically, but we recommend you specify
# it explicitly to prevent problems during startup.
#
# If this is not set to valid DNS name for your host, server-generated
# redirections will not work. See also the UseCanonicalName directive.
#
# If your host doesn't have a registered DNS name, enter its IP address here.
# You will have to access it by its address anyway, and this will make
# redirections work in a sensible way.
#
ServerName www.example.com:80
```

Change the last line of this section to the following:

```
ServerName www.example.com:8080
```

Now restart Apache and retest the installation with the following:

```
http://localhost:8080/
```

If you are still experiencing problems, the Apache Foundation has provided a nifty document about some other issues that may arise during installation. You can view it by going to <http://httpd.apache.org/docs/2.2/platform/windows.html>.

Installing PHP

PHP is a server-side scripting language that allows your web site to be truly dynamic. PHP stands for *PHP: Hypertext Preprocessor* (and, yes, we're aware PHP is a "recursive acronym" — probably meant to confuse the masses). Its flexibility and relatively small learning curve (especially for programmers who have a background in other programming languages like C, Java and Perl) make it one of the most popular scripting languages used today. PHP's popularity continues to increase as businesses and individuals everywhere embrace it as an alternative to Microsoft's ASP.NET languages. According to Netcraft, PHP code can be found running on approximately 21 million web sites.

There are several different installation methods for PHP, though we strongly recommend you follow the manual installation process. At the time of publication, the automated installer is not complete, secure or intended for use on live servers. Follow these steps to install PHP on your system:

1. Go to the PHP web site at www.php.net.
2. Click on the Download link to go to the site's downloads page.
3. Scroll down to the Windows Binary section, and click on the appropriate link to download the latest PHP .zip package.
4. Click any of the mirror sites to begin the download. If you have difficulties downloading from one mirror, then try a different mirror that may be closer to you.
5. Once the Zip file has been downloaded, extract its contents using any standard unzip program and save it to the directory of your choice. We recommend a directory named `C:\PHP`.

Both Windows XP and Windows Vista have built-in capabilities to extract files from Zip archives. If you are on a different version of Windows or prefer to use a dedicated compression tool, we recommend 7-Zip available at www.7-zip.org. It is a free application that can work with many different compression formats, including Zip.

6. It is advised to run PHP with a `php.ini` file. By default, the PHP installation provides two copies of the file with common configuration values: `php.ini-dist` and `php.ini-recommended`. Rename the configuration file of your choice to `php.ini`.

The `php.ini-dist` file is meant to be used for development purposes while `php.ini-recommended` has additional security measures and should be used when your site goes live. Depending on your reason for using PHP, choose the `php.ini` file that best suits your needs. For the purposes of this book, we are going to be using the `php.ini-dist`. Feel free to switch to the `php.ini-recommended` file as your default once you are more familiar with how PHP behaves.

7. Bring up the System Properties window. In Windows XP, this is done by right-clicking on the My Computer icon on your desktop and selecting Properties. In Windows Vista, this is done by right-clicking on the Computer icon on your desktop, selecting Properties and then Advanced System Settings.

8. Select the Advanced tab, and then click the Environment Variables button. Add the directory to which you extracted PHP to your System's PATH variable (C:\PHP in our configuration). Also create a new System variable PHPRC with the same directory as its value. This allows other applications (such as Apache) to find PHP without your having to copy files into your System directory.

Configuring PHP to Use MySQL

MySQL support was included in earlier versions of PHP by default, but starting with PHP version 5 you now have to specifically enable this. For PHP to play nice with MySQL, you need to make two changes to your `php.ini` file. Open the file using your text editor and locate the following lines:

```
; Directory in which the loadable extensions (modules) reside.  
extension_dir = "./"
```

Change the line to:

```
extension_dir = "C:\PHP\ext"
```

Then locate the following line:

```
;extension=php_mysql.dll
```

The semicolon is what denotes a comment within this file and will be ignored. Simply remove the semicolon at the beginning of the line to uncomment it:

```
extension=php_mysql.dll
```

Finally, copy the file `libmysql.dll` from your C:\PHP directory into your C:\Windows\System32 or C:\WINNT\System32 directory.

Configuring Apache to Use PHP

Now that both Apache and PHP are installed, there are a few more customizable options that need to be adjusted. To configure Apache to recognize a PHP file as one that needs to be parsed with the PHP engine, you need to first locate the following lines in your `httpd.conf` file:

```
# AddType allows you to add to or override the MIME configuration  
# file specified in TypesConfig for specific file types.  
#  
#AddType application/x-gzip .tgz  
#  
# AddEncoding allows you to have certain browsers uncompress  
# information on the fly. Note: Not all browsers support this.  
#  
#AddEncoding x-compress .Z  
#AddEncoding x-gzip .gz .tgz  
#  
# If the AddEncoding directives above are commented-out, then you  
# probably should define those extensions to indicate media types:  
#  
AddType application/x-compress .Z  
AddType application/x-gzip .gz .tgz
```

Part I: Movie Review Web Site

Then add the following lines:

```
AddType application/x-httpd-php .php
AddType application/x-httpd-php-source .phps
PHPIniDir "C:\PHP"
```

If you installed PHP in a location other than the recommended C:\PHP, then make sure your path matches the location of the directory.

Next, you need to add the PHP module into your `httpd.conf` program so that Apache can properly coordinate with PHP to serve the dynamically generated pages PHP will produce. In your configuration file, locate the following lines:

```
# Dynamic Shared Object (DSO) Support
#
# To be able to use the functionality of a module which was built as a DSO you
# have to place corresponding 'LoadModule' lines at this location so the
# directives contained in it are actually available before they are used.
# Statically compiled modules (those listed by 'httpd -l') do not need
# to be loaded here.
#
# Example:
# LoadModule foo_module modules/mod_foo.so
#
LoadModule actions_module modules/mod_actions.so
LoadModule alias_module modules/mod_alias.so
LoadModule asis_module modules/mod_asis.so
LoadModule auth_basic_module modules/mod_auth_basic.so
#LoadModule auth_digest_module modules/mod_auth_digest.so
...
#LoadModule usertrack_module modules/mod_usertrack.so
#LoadModule version_module modules/mod_version.so
#LoadModule vhost_alias_module modules/mod_vhost_alias.so
```

Add the following line:

```
LoadModule php6_module "C:\PHP\php6apache2_2.dll"
```

Again, make sure your path matches the location of the `php6apache2_2.dll` file if you did not install PHP in the recommended directory.

Oh, and remember to restart Apache after you've saved your modifications to `httpd.conf` or else Apache will not be aware of your changes!

Testing the Configuration

To ensure that both PHP and Apache have been configured to work together, let's write a short test program. Open notepad and type the following program:

```
<html>
  <head>
    <title>PHP Testing</title>
  </head>
  <body>
    <?php
    echo "<p>If you see this then we did it right!</p>";
    ?>
  </body>
</html>
```

Save this file as `test.php` in Apache's `htdocs` directory. By default it is at `C:\Program Files\Apache Software Foundation\Apache2.2\htdocs`. Then, open your web browser and visit `http://localhost/test.php`. You should see the screen shown in Figure 1-1.

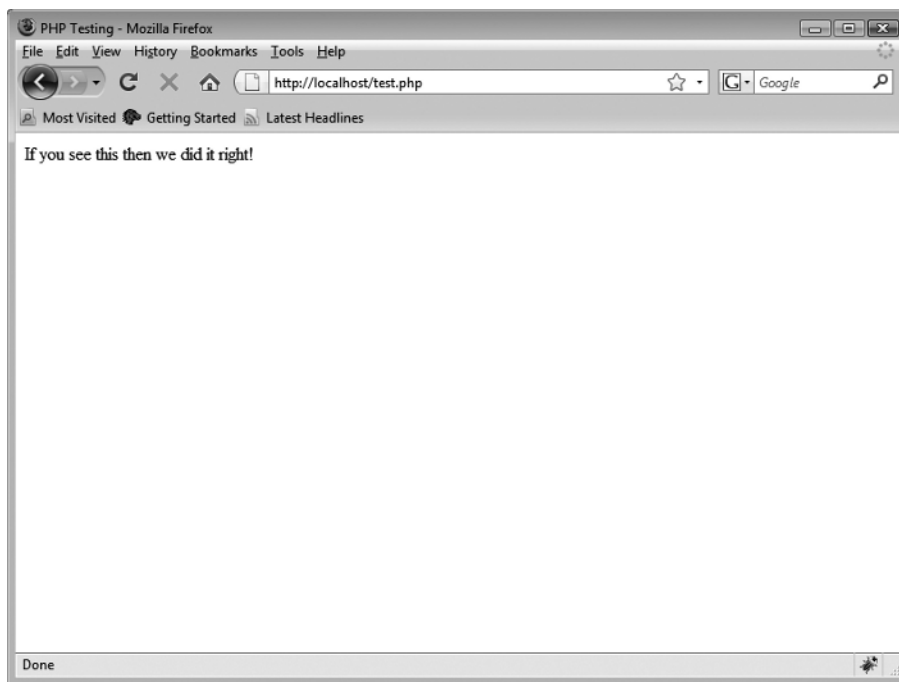


Figure 1-1

Installing MySQL

Another open source favorite, MySQL, is the database construct that enables PHP and Apache to work together to access and display data in a readable format to a browser. It is a Structured Query Language (SQL) server designed for heavy loads and processing of complex queries. As a relational database system, MySQL allows many different tables to be joined together for maximum efficiency and speed.

MySQL is the perfect choice for providing data via the Internet because of its ability to handle heavy loads, its advanced security measures and (depending on who you ask) it's easier to manage than some of the other open source database servers available. Follow these steps to install MySQL on your Windows system:

1. Go to the MySQL web site at `www.mysql.com`, and click the Developer Zone tab. Then, click the Downloads link on the navigation bar just under the tabs.
2. Scroll down and click on the link for the latest General Availability version of MySQL.
3. Scroll down to the Windows section of the downloadable files and click Pick a Mirror next to the Windows Essentials package.
4. Select the download from a nearby mirror and the download will begin.
5. As was with Apache, you should be able to double-click the MSI file to initiate the Installation Wizard, but you may experience some issues depending on what security policies Windows has in effect. We recommend running the installer with administrative privileges from within a console window.

To open a console as an Administrator in Windows XP, navigate through Start ⇒ All Programs ⇒ Accessories, right-click on Command Prompt, and select the Run As option. In Windows Vista, navigate through Start ⇒ All Programs ⇒ Accessories, right-click on Command Prompt, and select the Run as administrator option.

6. Use the `cd` command to navigate to where you downloaded the installer file, and then run the installer using `msiexec -i`. The Installation Wizard will open.

```
cd C:\Users\Timothy\Downloads\  
msiexec -i mysql-essential-5.1.26-rc-win32.msi
```

7. At the Setup Type screen, the Typical installation option is recommended for beginners and will suffice for most of your needs. If you have special circumstances or are an advanced user, feel free to choose the Complete or Custom setup options instead.
8. How long it takes for MySQL to be installed on your system depends on many factors, but the installation should proceed quickly and shouldn't take longer than a couple of minutes. After the wizard has finished installing the appropriate files, its final screen will simply indicate the installation is complete. Click Finish to end the wizard.
9. To close the console window from which you launched the installation file, you can either click the X in the window's top-right corner or enter `exit` at the prompt.

Configuring MySQL

The MySQL Server Instance Configuration Wizard is used on Windows to install MySQL as a service and establish a basic configuration. Go to Start ⇒ All Programs ⇒ MySQL ⇒ MySQL Server Instance Configuration Wizard to launch the utility.

You can choose either a Detailed Configuration or Standard Configuration, and we recommend the Standard Configuration option unless you are an advanced user.

We also recommend you check the option to include the bin directory in Window's PATH variable. This will allow you to run MySQL's utilities from the command line without having to explicitly type the full path each time and without having to manually configure the PATH variable as you did with Apache and PHP.

Last, we recommend you specify a password for MySQL's root user so your installation isn't left vulnerable to unauthorized access.

The Configuration Wizard will prepare a basic configuration file at `C:\Program Files\MySQL\MySQL Server 5.1\my.ini` by default and install MySQL as a windows service. You can fine-tune your installation by modifying the directives in `my.ini`. MySQL only reads this file once when it starts up, so you will need to restart MySQL any time you make changes to it for them to be active.

Starting and Stopping MySQL

Like Apache, MySQL runs as a background service without much interaction on the desktop. You can start and stop the server in one of two ways:

- ❑ **Windows Service Manager:** Go to Start ⇒ Control Panel ⇒ Administrative Tools, and click on the Services icon. Alternatively, you can go to Start ⇒ Run and execute `services.msc`. Highlight the MySQL entry, and click the desired action (start, stop, or restart).
- ❑ **The net command:** Open up a command window that has administrative privileges, and type `net start mysql` to start MySQL and `net stop mysql` to stop MySQL.

Testing Your Installation

As with the other applications, it's a good idea to test your MySQL installation. MySQL comes with a command-line client that you can use to connect to the MySQL server that is running and execute queries against your databases. Go to a command line and run:

```
mysql.exe -u root -p test
```

MySQL should prompt you for a password, and you need to enter whatever you set it as for the root user when you configured the MySQL installation.

The `-u` option provides the username that you're using to connect to MySQL, in this case `root`, and `-p` tells MySQL to prompt you for the password. `test` is the name of the database you will be using. The `test` database is installed by default.

The prompt will change to `mysql ⇒` and whatever you enter will be sent to the MySQL server. See what database tables have been set up by default. Type the following:

```
SHOW DATABASES;
```

You should see three existing databases, `information_schema`, `mysql` and `test`. To see what tables there are in the `mysql` database, type the following:

```
SHOW TABLES FROM mysql;
```

To exit the MySQL client, type `exit`.

Setting Up Dedicated MySQL User Accounts

Malicious hackers can be quite crafty in the ways in which they break into your system, especially if you are directly connected to the Internet. MySQL allows you to pick and choose what user is allowed to perform what function based on the privileges that you establish. All user privilege information is stored in a database called `mysql`.

If you're the only one accessing the MySQL database, you may not have to worry about adding users. You have already used MySQL's root account to verify you can communicate with the running instance of the MySQL server. However, use of the root account should be limited for administrative tasks only. What if you have, say, an Aunt Edna who is going to help you out by inputting some backlogged information? You want her to be able to go into the tables and look at things, and even insert some information. But you probably don't want her to be able to delete your entire database. By restricting her privileges as a user, you help to protect your data.

Connect to MySQL using the MySQL command-line client as the `root` user as previously discussed. Then follow these steps:

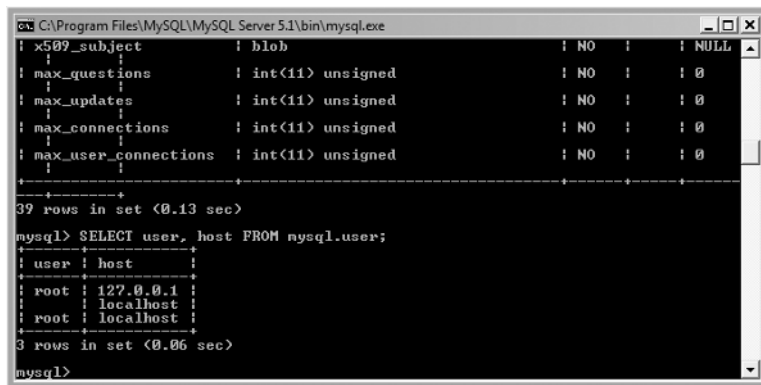
1. If you would like to see all the privileges that can be assigned, you can type the following:

```
SHOW COLUMNS FROM user FROM mysql;
```

2. You only want to look at what users are already there, so type the following:

```
SELECT user, host FROM mysql.user;
```

You should see what is depicted in Figure 1-2.



```
C:\Program Files\MySQL\MySQL Server 5.1\bin\mysql.exe
+-----+-----+-----+-----+-----+
| x509_subject | blob | NO | NULL |
+-----+-----+-----+-----+
| max_questions | int(11) unsigned | NO | 0 |
+-----+-----+-----+-----+
| max_updates | int(11) unsigned | NO | 0 |
+-----+-----+-----+-----+
| max_connections | int(11) unsigned | NO | 0 |
+-----+-----+-----+-----+
| max_user_connections | int(11) unsigned | NO | 0 |
+-----+-----+-----+-----+
39 rows in set (0.13 sec)

mysql> SELECT user, host FROM mysql.user;
+-----+-----+
| user | host |
+-----+-----+
| root | 127.0.0.1 |
| root | localhost |
| root | localhost |
+-----+-----+
3 rows in set (0.06 sec)

mysql>
```

Figure 1-2

3. Because you want to set up a secure service, you want to remove the blank user for the `localhost` host. Type the following:

```
DELETE FROM mysql.user WHERE Host="localhost" AND User="";
```

You will get a response from MySQL that states:

```
Query OK, 1 row affected (0.02 sec)
```

Note the time it takes to process the query may differ based on the speed of your computer, but the important thing here is that you see “Query OK.”

4. Now you’re going to GRANT Aunt Edna some privileges as a new user, so type the following:

```
GRANT SELECT, INSERT, UPDATE ON *.*  
TO edna@localhost  
IDENTIFIED BY "ednapass";
```

You’ll notice how the prompt changed to `->` on the second line. MySQL will not run the query until it encounters the terminating semicolon. This allows you to enter longer queries on multiple lines.

You have now established `edna` as a valid user will be allowed access to your MySQL system provided two things:

- She attempts her connection from the localhost — not a different connection from a remote computer.
- She supplies the correct password: `ednapass`.

Your Aunt Edna will now be allowed to select information from the database, insert new information in the database, and update old information in the database. By giving her access to all the tables in the database (via the use of `ON *.*`), you have allowed her to modify any table in existence.

As you become more familiar with working with tables and MySQL commands, modifying privileges or user information will become easier for you because the information is all stored in a table (just like everything else in MySQL).

A complete list of privileges that you can grant is available at the MySQL web site, <http://dev.mysql.com/doc/refman/5.1/en/privileges-provided.html>.

Where to Go for Help and Other Valuable Resources

Although we’ve certainly tried to make this as easy as possible for you, there are so many different variables in computers and their setups that it is virtually impossible to cover every possible situation. Anyone who works with computers on a regular basis is surely aware that, while in theory everything seems relatively simple, things don’t always go as planned (or as you think they should). To your advantage, there are several avenues for help should you find yourself in a difficult situation.

Help within the Programs

Before getting online and searching for help, you can try looking for answers to your problems within the programs themselves.

In Apache, the manual was installed with the standard installation and can be accessed in `C:\Program Files\Apache Software Foundation\Apache2.2\manual`. A check of your error log will be most helpful as well (`C:\Program Files\Apache Software Foundation\Apache2.2\logs\error.log`).

With the MySQL client, you can see some information and command-line arguments by typing the following at your command prompt:

```
mysql.exe --help
```

This provides a multitude of commands that will help you find what you need, or at the very least, a valuable “cheat sheet” for administering your MySQL server. In addition, this will allow you to see the current settings for your server at a glance, so you can potentially troubleshoot any problem spots.

Source Web Sites

You undoubtedly know where to find these by now, but just in case, the web sites associated with each of our three components have incredibly detailed information to help you work out any issues or report any bugs you may find:

- ❑ For Apache questions and information: www.apache.org
- ❑ For PHP questions and information: www.php.net
- ❑ For MySQL questions and information: www.mysql.com

Summary

By now, you should have an idea of what AMP is and how it fits into the world of open source software. You know that the abbreviation AMP refers to Apache, MySQL, and PHP, all of which work together to help you develop dynamic web sites.

Now you’ve installed, configured and tested the installation for Apache, MySQL, and PHP, you should be ready to start making a web site! You’ll get your hands dirty in the next chapter starting with lessons on PHP code and the creation of your movie review web site.