

Mac OS X Installation Guide

This is version 1.1 (30 June 2004) of the Mac OS X Installation guide. It is updated version that replaces Appendix C in the 2nd edition (16 May 2004) of “Web Database Applications with PHP and MySQL” by Hugh E. Williams and David Lane. This appendix is copyright O'Reilly Media Inc. We thank Jeanne Pickering, Scott Brown, Matt Starks, and Bill Griffiths for feedback, errata, and suggestions.

This appendix is a guide to installing the software used in the book on an Apple MacIntosh OS X platform. The instructions are designed for administrators, so you'll need to be able to login as the root user. The instructions were written and tested on Mac OS X Panther (Version 10.3.1) with kernel version Darwin 7.0.0.

After showing you how to get PHP, Apache, and MySQL running, we then present a short guide to downloading and installing the PHP script and database examples used in this book. We also show you how to set up tools that are needed by examples in the book but aren't included in the default Apache, PHP, and MySQL installations.

The Mac OS X environment, PHP, Apache, MySQL, and our code examples can all change over time. This means that this guide may not work perfectly when you use it. To get the latest installation guide in HTML and PDF formats, along with changes and corrections to this guide, visit <http://www.webdatabasebook.com/install-guides>.

Getting Started

There are two approaches to working with MySQL, Apache, and PHP:

- Use the software that's installed with Mac OS X and add the missing components. Panther includes Apache 1.3.28 and PHP, but not MySQL. This is the easiest approach, but is prone to the problem of the software being out-of-date and, of course, you still need to install and configure some components.

- Obtain and build the software from source code. This is the most difficult approach, but it has the advantage that the latest software is installed and the configuration layout and options are controlled in the process.

This appendix focuses on the first approach: we use the Apache that comes with Mac OS X, upgrade PHP, and add MySQL.

Before we can begin, you need to unlock your root user. Do the following:

1. Open NetInfo Manager, which is located in Applications > Utilities.
2. Click on the Security menu, and then on Enable Root User. Enter a password and record it for later use.
3. Quit NetInfo manager.

When you've finished the installation steps in this appendix, you can choose to disable the root user by repeating the steps above but clicking on Disable Root User instead .

Installing MySQL

This section shows you how to install and configure your MySQL 4 DBMS server. When you've completed this section, continue on to *Section Starting MySQL* and *Section Configuring MySQL*.

1. Download MySQL. Using a web browser, visit <http://www.mysql.com/downloads/>. Under the heading *MySQL database server & standard clients*, click on the the link that's marked as the production release. On the release page, find the heading *Mac OS X Package Installer downloads* (not Mac OS X downloads!). Click on download next to the Standard option. Save the file to the desktop.

At the time of writing, choosing which release to use is difficult. We recommend Server version 4.0, as you can use all of the book except some of the new features of MySQL 4.1 that are discussed in *Chapter 14*.

If you choose Server version 4.1 or later, then the material in this book that relies on the standard MySQL library can't be used at the time of writing. The standard library is discussed in *Chapter 6*, and the standard library supports the PEAR DB package described in *Chapter 7* and used in the example application in *Chapters 15 to 19*. It's likely that this will be remedied in the future, and that downloading 4.1 will become a good option. See *Appendix F* for more information.

1. After the download completes, double click on the downloaded file that's stored on the desktop; the file has name such as *mysql-standard-4.0.17.dmg*. This mounts the image file: a dialog will show progress and then the Finder will appear showing two packages and a *ReadMe.txt* file.

2. Double click the install package in the Finder. This is the package named, for example, *mysql-standard-4.0.17.pkg*. The Installer will run.
3. The first step in the installation is an Introduction page. Click Continue.
4. The second step is viewing the Read Me file. This has few useful hints, so you might find it useful to save the file somewhere for later reference but there's no requirement to do so. Click on Continue.
5. The third step is the license. Click Continue.
6. The fourth step is to select a destination volume (it defaults to Mac OS X). Unless you have a reason to change it, click Continue.
7. The fifth step is the Easy Install page. Click Install. You may then need to enter the root user password you setup in the previous section. Now, wait while the software installs.
8. The final step is the Finish up page. Click Close.
9. Install the StartupItem package. This configures your machine so that the MySQL server will start when your machine is turned on. To do this, double click the StartupItem package in the Finder; this is the package named, for example, *MySQLStartupItem.pkg*. The Installer will run. Now repeat the six steps from Steps 6 to 11 for the StartUpItem.
10. MySQL is now installed but isn't yet configured. To configure it, start by opening Terminal, which is located in Applications > Utilities. This opens a terminal window which we recommend maximizing. In the following steps that require the Terminal, we show commands that are entered next to a prompt, `%`. To enter these commands, type them into your Terminal window and press Enter.
11. Log in as the root user. You can login as root by typing:

```
% su
```

and then supplying the root user password that you set when enabling the root user in the previous section.
12. Run the *mysql_install_db* script to initialize the system databases used by MySQL. To do this, change directory to where MySQL is installed:

```
% cd /usr/local/mysql
```

Type the following to run the installation script:

```
% ./scripts/mysql_install_db
```
16. Change the owner of the MySQL program files to be the `root` user:

```
% chown -R root /usr/local/mysql
```
17. Change the owner of the MySQL databases and log files to be the `mysql` user:

```
% chown -R mysql /usr/local/mysql/data
```

This user already exists in your Panther installation.

18. Change the group of the MySQL installation files to be the `mysql` group:

```
% chgrp -R mysql /usr/local/mysql
```

This group already exists in your Panther installation.

19. Copy the default medium-scale parameter configuration file to the default location of `/etc`. These parameters are read when MySQL is started. The copy command is:

```
% cp support-files/my-medium.cnf /etc/my.cnf
```

20. Edit the configuration file and adjust the default number of maximum connections to match the default value for the maximum Apache web server connections. Also, let's turn on the query cache. To do this, we'll use the pico editor that comes installed with Mac OS X; we don't use the popular TextEdit editor because it has trouble saving files when you're the root user and when the file exists.

To edit the configuration file with pico, type:

```
% pico /etc/my.cnf
```

Using the down arrow key, scroll down through the file until you find section beginning with the following text:

```
# The MySQL server
[mysqld]
```

In this section, type the following lines:

```
set-variable = max_connections=150
query_cache_type = 1
```

Save the file by holding down the `ctrl` (Control) key and pressing the `o` key (we refer to this as `ctrl-o` throughout this appendix). Then press Enter. Quit pico by holding `ctrl` and pressing `x` (`ctrl-x`).

21. The installation of MySQL is now complete, but it isn't running. To get it running, restart your machine by clicking on the Apple menu and then Restart.

MySQL isn't completely configured. The steps in the next section set it up further.

Configuring MySQL

The following steps set a password for the root user and create a new user for the MySQL installation that is used in PHP scripts to access the DBMS. The steps assume you've run Terminal, logged in as the root user, and that MySQL is running; see the previous section for more information.

The new user can carry out all actions required from *Chapter 6* to *Chapter 19* on the *winstore* database but has no access to other databases and can't change database access privileges. The new user also can't access the DBMS from a remote server. If you're creating your own application, replace *winstore*

with the name of your database. More information on user privileges can be found in *Chapter 14*.

The steps are as follows:

1. Choose and set a password for root user access to the MySQL DBMS. To set a password of **secret**, type into your Terminal window:

```
| % /usr/local/mysql/bin/mysqladmin -uroot password secret
```

Record the password for later use.

2. Start the MySQL command line interpreter using the password defined in the previous step:

```
| % /usr/local/mysql/bin/mysql -uroot -psecret
```

This displays a **mysql>** prompt.

3. Add a new user by typing the following into the command interpreter. Choose a username to replace **username** and a password to replace **password** in the following command:

```
|mysql> GRANT SELECT, INSERT, UPDATE, DELETE, LOCK TABLES ON winestore.* TO  
| username@localhost IDENTIFIED BY 'password';
```

It doesn't matter that the **winestore** database doesn't exist yet.

MySQL responds with:

```
|Query OK, 0 rows affected (0.00 sec)
```

Record the username and password.

If the statement doesn't work for your release, try it again but omit the **, LOCK TABLES** component.

4. Quit the MySQL command interpreter with the command:

```
|mysql> quit
```

MySQL responds with:

```
|Bye
```

5. Test the user by running the MySQL command interpreter using the **username** and **password**:

```
| % /usr/local/mysql/bin/mysql -uusername -psecret
```

6. MySQL responds with a message beginning:

```
|Welcome to the MySQL monitor.
```

7. Quit the MySQL interpreter again with:

```
|mysql> quit
```

The MySQL DBMS is now configured with a user who can access the **winestore** database from the database server machine **localhost**. The **winestore** database can't be tested yet as it isn't loaded. The database is loaded in *Section What's Needed for this Book*.

Setting-up Apache and PHP

Apache Version 1 is already installed under Panther and doesn't need to be upgraded to use this book; you don't need the latest version, Apache 2. However, by default, it isn't running, it won't serve PHP requests, and it only serves standard, non-secure requests. In this section, we discuss how it's set up on your machine, how to get it started, and how to modify it so that it can serve secure and PHP requests.

PHP 4.3.2 is installed with Panther. It includes most of the features you need to work with this book, except that the PEAR installer doesn't work without an additional step. In this section, we show you how to upgrade to the latest stable release using a package. Fixing the PEAR installer is discussed in *Section What's Needed for this Book*.

Starting Apache

Let's start by getting Apache running. To get it started, you can do one of two things:

Automatically start Apache each time your machine starts.

To do this, launch System Preferences from Applications > System Preferences. Then, select Sharing from beneath the *Internet & Network* heading. At the left of the window, select Personal Web Sharing by ticking the box. Quit System Preferences. Apache will now start and will also start each time you start up your computer.

Manually start Apache using Terminal.

Start by opening Terminal, which is located in Applications > Utilities. In the Terminal window, login as the root user and start Apache using:

```
% su
% apachectl start
```

You should see a message such as:

```
[/usr/sbin/apachectl start: httpd started
```

Quit the Terminal program. When you restart your machine, you need to repeat this process to restart Apache.

To test that your Apache is running, use a web browser to retrieve the URL *http://localhost/*. You should see an Apache test page.

The Apache and PHP Setup

Your Apache is installed differently to how it's configured for most other Unix variants:

- You'll find the Document Root (where the documents are stored that are retrieved when you request *http://localhost/*) in the directory

/Library/WebServer/Documents. (On other Unix variants, the directory is */usr/local/apache2/htdocs/*.)

- You'll find the Apache configuration file *httpd.conf* in the directory */etc/httpd*. (On other Unix variants, it's usually in */usr/local/apache2/conf*.)
- The script used to start and stop Apache, *apachectl*, is located in */usr/sbin*. (On other Unix variants, it's usually in */usr/local/apache2/bin/*.)
- You'll find the PHP *php.ini* configuration file in the directory */usr/local/php/lib*.

In addition, while your Apache does have the PHP module available to it, it isn't configured to serve PHP requests. To set it up so that it will serve PHP requests you can do one of two things:

Upgrade your PHP module

By using Marc Liyanage's upgrade package to get the latest release of PHP, your Apache will be automatically configured to serve PHP requests. If you plan to do this, then follow the instructions in the next section now and, when you've done that, return to Step 9 in this section to test everything is working correctly.

Modify the *httpd.conf* configuration

If you don't want to upgrade your PHP module, then you can manually modify the configuration of your Apache so that it serves PHP requests. This is outlined in the steps described next.

We recommend upgrading your PHP module to the latest release.

To manually set up your Apache to serve PHP requests, do the following:

1. Start by opening Terminal, which is located in Applications > Utilities. Maximize the Terminal window. In the Terminal window, login as the root user by typing:

```
| % su
```

Supply the password you set when unlocking the root user.
2. In the Terminal window, type:

```
| % pico /etc/httpd/httpd.conf
```

This launches the pico editor and opens the *httpd.conf* file.
3. Search in the file for the string **php**. You can do this by holding the **ctrl** (Control) key and pressing the **w** key (**ctrl-w**), typing **php**, and pressing Enter. You should see the following line:

```
| #LoadModule php4_module libexec/httpd/libphp4.so
```
4. From the line in Step 3, remove the # so that it reads:

```
| LoadModule php4_module libexec/httpd/libphp4.so
```
5. Search again for the string **php** by pressing **ctrl-w** and then Enter. You should see the following line:

```
|#AddModule mod_php4.c
```

6. From the line in Step 5, remove the # so that it reads:

```
|AddModule mod_php4.c
```

7. Save the file by **ctrl-o** and Enter. Quit pico by typing **ctrl-x**.
8. Restart your Apache so that it re-reads its configuration by typing into your Terminal window:

```
;% /usr/sbin/apachectl stop  
;% /usr/sbin/apachectl start
```

(For some reason, `/usr/sbin/apachectl restart` doesn't work.)

9. To test the PHP module, change the directory to the Apache document root using your Terminal window:

```
;% cd /Library/WebServer/Documents
```

10. Create a file with the name *phpinfo.php* using pico. Type the following into your Terminal window:

```
;% pico phpinfo.php
```

The pico editor runs and an empty window appears. Into the window, type:

```
|<?php phpinfo(); ?>
```

Save the file using **ctrl-o** and pressing Enter. Then quit pico using **ctrl-x**.

11. Test the newly created PHP script by retrieving with a browser the following URL <http://localhost/phpinfo.php>. You should see a page of information about Apache and PHP.

Upgrading PHP

PHP 4.3.2 is installed with Panther. To upgrade your PHP to the latest stable release, do the following:

1. Using a web browser, visit Marc Liyanage's website, <http://www.entropy.ch/software/macosx/php/>.
2. Scroll down to the *Installation Instructions* heading. Next to item 1 in the list, click on the link to the installation package. This is labelled, for example, *PHP 4.3.4 (entropy.ch Release 1)*. Save the downloaded file.
3. After the download is complete, double-click on the disk image file that's been downloaded. It has a file name such as *Entropy-PHP-4.3.4-1.dmg*. This mounts the disk image file and the Finder appears.
4. In the Finder, double-click on the package and the Installer runs; the package has a name such as *php-4.3.4.pkg*. On the Installer welcome screen, click Continue. On the Destination screen, click on the destination volume that is home to your Apache server and click Continue. On the Easy Install screen, click Upgrade; you may need to then provide your root user password. The package will install. On the Finish Up screen, click Close.

The PHP upgrade is complete -- you don't even need to restart your Apache!

Installing a Secure Apache Server

This section describes how to configure a secure version of the Apache web server so that you can support *https://* requests. If you don't need a secure server, skip this section. You can find out more about secure web servers in *Chapter 11*.

There are three major differences encountered when configuring Apache to use SSL versus using Apache normally:

Secure Sockets Layer software is required.

There are several sources of Secure Sockets Layer software. The OpenSSL is probably the most-commonly used with Apache, and it's already installed with Panther. We use it in this section.

A site certificate needs to be obtained and configured.

A free, self-signed certificate can be created (and that's what we do in this section). You need to replace it with a purchased certificate from a Certification Authority when an application goes live.

You need to start Apache manually

Because you'll add a passphrase in the steps in this section, Apache can't start automatically without you providing the passphrase. Therefore, when you restart your machine, you'll need to start it manually through a terminal window.

Creating a Key and Certificate

For Apache to use SSL, it needs a private key and a certificate. Once the key and certificate have been created, they need to be configured into Apache. These steps show you how to do this:

1. Start by opening Terminal, which is located in Applications > Utilities. Maximize the Terminal window. In the Terminal window, login as the root user by typing:

```
| % su
```

Supply the password you set when unlocking the root user.

1. Create the key. Into the Terminal window, type:

```
| % openssl genrsa -des3 1024 > /etc/httpd/localhost.key
```

If you have an actual domain for your server, replace **localhost** with the full domain name. Decide on a password and enter it twice; record it for future use. You've now created the private key.

3. Create the certificate request by typing:

```
| % openssl req -new -key /etc/httpd/localhost.key > \
```

```
| /etc/httpd/localhost.csr
```

If you have an actual domain for your server, replace `localhost` with the full domain name. The process asks for several fields including country, state, organization name, and email address; answer these as best you can, but it doesn't matter if you omit some answers by pressing Enter. The script produces a file that contains the certificate signing request.

4. Now, create the self-signed certificate by typing :

```
| % openssl req -x509 -days 90 -key \  
| /etc/httpd/localhost.key \  
| -in /etc/httpd/localhost.csr > \  
| /etc/httpd/localhost.crt
```

You need to provide the password you used to create your private key.

5. Activate your SSL module in your Apache installation. To do this, type:

```
| % apxs -e -a -n ssl /usr/libexec/httpd/libssl.so
```

6. You need to instruct Apache how to process secure requests. In the Terminal window, type:

```
| % pico /etc/httpd/httpd.conf
```

This launches the pico editor and opens a window containing the `httpd.conf` file.

7. Scroll to the end of the file using the down arrow key. At the end of the `httpd.conf` file, add the following lines:

```
<IfModule mod_ssl.c>  
  
    Listen 80  
    Listen 443  
  
    SSLRandomSeed startup builtin  
    SSLRandomSeed connect builtin  
  
    <VirtualHost _default_:443>  
        SSLEngine on  
        DocumentRoot "/Library/WebServer/SecureDocuments"  
  
        SSLCertificateFile /etc/httpd/localhost.crt  
        SSLCertificateKeyFile /etc/httpd/localhost.key  
    </VirtualHost>  
  
</IfModule>
```

These instructions configure Apache to serve secure documents from the directory `/Library/WebServer/SecureDocuments`. If you don't want to use that directory, you can replace it with your choice; we assume our choice in the next few steps. If you have an actual domain for your server, also replace `localhost` with the full domain name.

Save the file using `ctrl-o` and pressing Enter, and quit using `ctrl-x`.

6. Create the directory from which you want to serve secure files. Type:

```
| % mkdir /Library/WebServer/SecureDocuments
```

7. For testing, create a simple `index.html` file in the new directory. Type:

```
| % pico /Library/WebServer/SecureDocuments/index.html
```

Into the file, type:

```
| <html>Secure hello!</html>
```

Save the file using **ctrl-o** and pressing Enter. Quit pico using **ctrl-x**. Ensure the file can be accessed by typing:

```
| % chmod a+rx /Library/WebServer/SecureDocuments/  
| % chmod a+r /Library/WebServer/SecureDocuments/index.html
```

8. Stop and start Apache so it re-reads its configuration. Use the following commands:

```
| % apachectl stop  
| % apachectl start
```

You need to provide the password you used in creating your key. A secure Apache is now running and serving requests on port 443 (the default HTTPS port) via SSL and also serving regular HTTP requests on port 80. You can test it by requesting the resources <https://localhost/> and <http://localhost/> with a web browser running on the same machine as the web server. You should see the sample page you created in Step 7, and your regular pages respectively.

When a resource such as <https://localhost/> is requested with a browser, the browser alerts the user to an unknown certificate. To obtain a certificate that will be trusted by users, you need to send your certificate request to a Certification Authority to be signed using their authoritative certificates. There is a fee for this service. While the Apache configuration allows both the key and the certificate to be placed in a single file, the private key should not be sent to anyone, not even the Certification Authority. More documentation can be found at <http://www.openssl.org/docs/apps/openssl.html>.

What's Needed for this Book

This section shows you how to download and install our example database, all of the book code examples, the sample online winestore application, and the additional packages that are used throughout the book.

Installing PEAR Packages

The following optional PEAR package is required:

HTML_Template_IT

This is needed to work with *Chapter 7* and all later chapters.

Detailed instructions to install optional packages can be found in *Chapter 7*. You'll also find a discussion there about how to find about, install, and upgrade packages.

Panther has a problem: the PEAR installer doesn't work and you need to carry out an extra step to get it going. We show you how to install, upgrade, and fix the problem in the following steps:

1. Start by opening Terminal, which is located in Applications > Utilities. Maximize the Terminal window. In the Terminal window, login as the root user by typing:

```
| % su
```

Supply the password you set when unlocking the root user.

2. Let's fix the PEAR installer's configuration. To do this, you need to set the environment variable `PHP_PEAR_INSTALL_DIR` to point to the PEAR directory `/usr/local/php/lib/php`. However, when you use Terminal to type commands, you're using a shell program, and how you set environment variables is dependent on the shell you use.

Most users use the bash shell. To set an environment variable in bash, type into your Terminal window:

```
| % export PHP_PEAR_INSTALL_DIR=/usr/local/php/lib/php
```

If you know you're using the csh or tcsh shell, then type:

```
| % setenv PHP_PEAR_INSTALL_DIR /usr/local/php/lib/php
```

If you know how to add environment settings to your shell resource file (such as `.bashrc` or `.cshrc`), then add the command you just typed to the end of that file. If you don't do this, then each time you want to use the PEAR installer, you need to repeat this step first.

3. If you're not already, connect to the Internet.
4. Type the following to install the optional package:

```
| % /usr/local/php/bin/pear install HTML_Template_IT
```

You need to include the path to the Pear installer, because there's another Pear installer program installed in `/usr/bin` that we don't want to use.
5. Type the following to check if upgrades of the core packages used in this book are available:

```
| % /usr/local/php/bin/pear upgrade PEAR  
| % /usr/local/php/bin/pear upgrade Date  
| % /usr/local/php/bin/pear upgrade DB  
| % /usr/local/php/bin/pear upgrade Mail
```

Installing the Code Examples

The example PHP scripts are available from our book web site, <http://www.webdatabasebook.com>. In this section, we show you how to install them for use on your machine. The winestore application isn't installed in this section; see the *Section Installing the Winestore Application* for instructions.

To install the example scripts, follow these steps:

1. Using a browser, download the file <http://www.webdatabasebook.com/examples.zip>. If you're using Safari, you'll find that the file is usually automatically saved on the Desktop, unzipped, and shown as a folder labelled *wda*. If you're not using Safari, then save the file on your Desktop.

1. Open Terminal, which is located in Applications > Utilities. Maximize the Terminal window. In the Terminal window, login as the root user by typing:

```
% su
```

Supply the password you set when unlocking the root user.

2. You need to configure your Apache web server so that it won't show users the content of the *.inc* include files if they're retrieved with a web browser. There are several ways to do this, but the simplest is to edit your *httpd.conf* configuration file. You'll find the file in */etc/httpd/*.

To edit the file, login as the root user and edit the file by typing the following into a Terminal window:

```
% su
% pico /etc/httpd/httpd.conf
```

Scroll to the end of the file using the down arrow key. To the end of the file, add:

```
<Files ~ "\.inc$">
    Order allow,deny
    Deny from all
    Satisfy All
</Files>
```

Save the file by pressing **ctrl-o** and then Enter. Quit pico using **ctrl-x**. You now need to restart your Apache server. You can do this by typing:

```
% apachectl stop
% apachectl start
```

3. If you used Safari in step 1, then move the *wda* folder from the Desktop to the web server's document root:

```
% mv /Users/username/Desktop/wda /Library/WebServer/Documents
```

Replace **username** with your Mac OS X login name.

If you didn't use Safari in step 1, then create a new *wda* folder and move the examples into it:

```
% mkdir /Library/WebServer/Documents/wda
% mv /Users/username/Desktop/examples.zip /Library/WebServer/Documents/wda
```

Replace **username** with your Mac OS X login name.

4. If you didn't use Safari in Step 1, then unzip the *examples..zip* file in their new directory by doing this:

```
% cd /Library/WebServer/Documents/wda
% unzip examples.zip
```

5. Follow the instructions in the file */Library/WebServer/Documents/wda/README*, which you can open using

pico. The key steps are to modify the *db.inc* files you'll find in the chapter directories, using (for example) the following approach:

```
% cd /Library/WebServer/Documents/wda/ch06
% pico db.inc
```

Modify the lines beginning with `$username` and `$password` so that `fred` and `shhh` are replaced with the username and password you chose for your user when installing MySQL.

```
<?php
    $hostName = "localhost";
    $databaseName = "winestore";
    $username = "fred";
    $password = "shhh";
```

Save the file by pressing `ctrl-o` and Enter. Quit pico using `ctrl-x`.

6. You may also need to set the file permissions so that examples are accessible through your web browser. To do this, use:

```
% chmod a+rx /Library/WebServer/Documents/wda
% chmod a+r /Library/WebServer/Documents/wda/*
```

7. You should now be able to load the example list by requesting the following URL with a web browser running on the same machine as the web server: <http://localhost/wda/>. Test the examples from *Chapters 2 to 4*, and they should work.

You'll find that some of the examples from *Chapter 6* onwards run only if the *winestore* database has been loaded into the MySQL DBMS by following the instructions in the next section. In addition, most examples from *Chapter 7* onwards work only if you've installed the PEAR package `HTML_Template_IT`.

Installing the PDF PHP library

If you haven't installed the examples in the previous section, and you want to work with the PDF PHP library in *Chapter 13*, you need to download the class files. If you have installed our examples, you'll find the PDF-PHP library files in the `/Library/WebServer/Documents/wda/ch13` directory.

If you haven't installed our examples and want the library, follow these steps:

1. Visit the website <http://ros.co.nz/pdf/>.
2. From the Downloads section, choose the link to the Zip file that contains the class, sample, and the required font metric files. If you're using Safari, the file will automatically be saved on your Desktop, unzipped, and a new folder will appear. If you're not using Safari, then save the file on your Desktop. These instructions assume the file you've downloaded is *pdfClassesAndFonts_009e.zip*.
3. If you're using Safari, move the new folder into a shared location:

```
% mv /Users/username/Desktop/pdfClassesAndFonts_009e /usr/local/src/
```

Replace `username` with you Mac OS X login name.

If you're not using Safari, create a new folder, move the Zip file there, and unzip it:

```
% mkdir /usr/local/src/pdfClassesAndFonts_009e
% mv /Users/username/Desktop/pdfClassesAndFonts_009e.zip \
  /usr/local/src/pdfClassesAndFonts_009e
% cd /usr/local/src/pdfClassesAndFonts_009e
% unzip pdfClassesAndFonts_009e.zip
```

Replace **username** with your Mac OS X login name.

4. Edit your *php.ini* file with `pico` and include the directory */usr/local/src/pdfClassesAndFonts_009e* at the end of the `include_path` directive. This allows you to include the path in a PHP script using only the class name and without specifying the directory.
5. Restart your Apache server.

Loading the Winestore Database

A local copy of the *winestore* database is required to test the SQL examples in *Chapter 5*, to test some of the web database application examples in *Chapters 6 to 13*, and to use the sample winestore application described in *Chapters 15 to 19*. In addition, MySQL must be installed and configured before the *winestore* database can be loaded.

The steps to load the *winestore* database are as follows:

1. Using a web browser, download the file <http://www.webdatabasebook.com/database.zip>. If you're using Safari, the file will automatically be saved on your Desktop, unzipped, and a new *winestore.data* file will appear. If you're not using Safari, then save the file on your Desktop.
2. Open Terminal, which is located in Applications > Utilities. Maximize the Terminal window. In the Terminal window, login as the root user by typing:
| `su`
Supply the password you set when unlocking the root user.
3. If you're not using Safari, then uncompress the *winestore* database by typing the following
| `cd /Users/username/Desktop`
| `unzip database.zip`
Replace **username** with your Mac OS X login name.
4. Run the MySQL command-line interpreter using the root username and the **password** you set, and load the *winestore* database:

```
% /usr/local/mysql/bin/mysql -uroot -ppassword < \  
  /Users/username/Desktop/winestore.data
```

Replace **username** with your Mac OS X login name.

Be patient, this may take a while.

5. After the loading is complete the database can be tested by running a query. To do this, use the **username** and **password** you created when installing and configuring MySQL:

```
| % /usr/local/mysql/bin/mysql -uusername -ppassword
```

Now type:

```
|mysql> SELECT * FROM region;
```

This should produce the following list of wine regions as output:

region_id	region_name
1	All
2	Goulburn Valley
3	Rutherglen
4	Coonawarra
5	Upper Hunter Valley
6	Lower Hunter Valley
7	Barossa Valley
8	Riverland
9	Margaret River
10	Swan Valley

The *winstore* database has now been loaded and tested.

Installing the Winestore Application

The sample online winestore application is available from our book web site, <http://www.webdatabasebook.com>. In this section, we show you how to install it on your machine. We assume you've installed the *winstore* database by following the instructions in the previous section.

Follow these steps:

1. Using a browser, download the file <http://www.webdatabasebook.com/wda2-winstore.zip>. If you're using Safari, the file will automatically be saved on your Desktop, unzipped, and a new *wda2-winstore* folder will appear. If you're not using Safari, then save the file on your Desktop.
2. If you haven't configured your Apache web server so that it won't show users the content of the *.inc* include files, follow Steps 2 and 3 in *Section Installing the Code Examples*.
3. Open Terminal, which is located in Applications > Utilities. Maximize the Terminal window. In the Terminal window, login as the root user by typing:

```
| % su
```

Supply the password you set when unlocking the root user.

4. If you used Safari in step 1, then move the *wda2-winstore* folder from the Desktop to the web server's document root:

```
% mv /Users/username/Desktop/wda2-winstore /Library/WebServer/Documents
```

Replace **username** with your Mac OS X login name.

If you didn't use Safari in step 1, then move the examples and unzip them:

```
% mv /Users/username/Desktop/wda2-winstore.zip \  
/Library/WebServer/Documents/
```

Replace **username** with your Mac OS X login name. Then, unzip the *wda2-winstore.zip* code to create the subdirectory *wda2-winstore*:

```
% cd /Library/WebServer/Documents  
% unzip wda2-winstore.zip
```

5. Edit the file *wda2-winstore/includes/db.inc* using pico by typing:

```
% cd /Library/WebServer/Documents  
% pico wda2-winstore/includes/db.inc
```

6. Modify the lines beginning with **\$username** and **\$password** so that **fred** and **shhh** are replaced with the username and password you selected when configuring MySQL:

```
<?php  
$hostname = "localhost";  
$databasename = "winstore";  
$username = "fred";  
$password = "shhh";
```

Save the file by pressing **ctrl-o** and Enter. Exit pico using **ctrl-x**.

7. Edit the file *wda2-winstore/includes/winstore.inc* by typing:

```
% pico wda2-winstore/includes/winstore.inc
```

Locate the following three lines:

```
// define("D_INSTALL_PATH", "c:/progra~1/easyph~1/www");  
// define("D_INSTALL_PATH", "/Library/WebServer/Documents");  
define("D_INSTALL_PATH", "/usr/local/apache2/htdocs");
```

Uncomment the second line and comment-out the third so that you have the following:

```
// define("D_INSTALL_PATH", "c:/progra~1/easyph~1/www");  
define("D_INSTALL_PATH", "/Library/WebServer/Documents");  
// define("D_INSTALL_PATH", "/usr/local/apache2/htdocs");
```

Save the file by pressing **ctrl-o** and Enter. Quit the editor using **ctrl-x**.

8. You may also need to set the file permissions so that examples are accessible through your web browser. To do this, use:

```
% chmod a+rx /Library/WebServer/Documents/wda2-winstore  
% chmod a+r /Library/WebServer/Documents/wda2-winstore/*
```

9. You should now be able to load the application index by requesting the following URL with a web browser running on the same machine as the web server: *http://localhost/wda2-winstore/*. Click on the Use the Application link and you should be able to use the application.