

Chapter 1

Building Your Application

In This Chapter

- Understanding PHP and MySQL versions
 - Installing applications files from the CD
 - Setting up your programming environment
 - Customizing the applications in the book
 - Planning your application
-

You know PHP. Or at least you've been introduced and have spent some quality time together. You know PHP syntax, control structures, and some built-in functions. You can display a form and retrieve the information from it. You can interact with a database. You have the basics down.

Or, perhaps you're an expert programmer in another language. You've been using C for years. You know the basics of programming. You don't know exactly how the familiar programming features are implemented in PHP, but you believe you can understand quickly from seeing examples. After all, a `for` loop is a `for` loop and an `if` statement is an `if` statement. Other programmers have told you how easy PHP is and how similar it is to C.

Now, you want to write a practical application. You need an application quickly. Perhaps you need to provide a login application to protect a Web site or part of a Web site. Perhaps you need to provide an online catalog for a store. Perhaps you need to implement a forum on your Web site where your customers can interact.

This book provides complete applications. Chapters 3 through 8 provide all the code for six popular applications. An additional bonus chapter on the CD provides a seventh application. You can copy the code from the CD to your Web site and have a working application. Of course, nothing is ever quite that simple. You probably need to modify the application; you might need to make a small modification, such as adding your company logo, or a larger modification, such as removing or adding features to an application. Thus, I provide

explanations with the code so that you can more easily modify it. The applications are

- ✓ **User authentication:** The user authentication application uses HTTP (Hypertext Transfer Protocol) authentication. This feature is built in and useful for simple user/password authentication. It is quick and easy, but also limited and not very flexible. (See Chapter 3.)
- ✓ **User login:** In the user login application, the user/password authentication is written from scratch in PHP. This application allows users to register and set up their own user IDs and passwords, as well as log in to the Web site. (See Chapter 4.)
- ✓ **Online catalog:** Displays product information stored in a MySQL database on a Web site where customers can view it. (See Chapter 5.)
- ✓ **Shopping cart:** This application allows customers to purchase the products that they find in an online catalog. (See Chapter 6.)
- ✓ **Content management system:** This application allows users to post, delete, and edit information on a Web site. (See Chapter 7.)
- ✓ **Web forum:** This application functions as a public bulletin board. Users can read the posted messages and post messages of their own or responses to current messages. (See Chapter 8.)
- ✓ **Mailing list management:** This application allows users to subscribe to one or more mailing lists. An authorized administrator can use the application to create new mailing lists. (See the Bonus Chapter on the CD.)

You can copy an application from the CD to your Web site and have a working application instantly — well, assuming you have the correct versions of PHP and MySQL. In the first section (“Understanding PHP and MySQL Versions”), you find out more information about the versions that I use in this book. You also have to put the application files in the correct place, and I tell you how to do that in the “Using the Application Source Code” section.

Understanding PHP and MySQL Versions

Because PHP and MySQL are open-source software, new versions are released often and sometimes without much warning. Sometimes new releases include changes in the way the software works or the installation procedure that require changes to your application — not often, but occasionally. The software developers try to maintain *backward compatibility* (meaning old programs can run on the new versions), but sometimes it’s just not possible. Consequently, you need to be aware of versions and keep informed about PHP and MySQL versions, changes, and problems.

MySQL

Currently, MySQL offers three versions: MySQL 4.0, MySQL 4.1, and MySQL 5.0. At this time, MySQL 5.0 is a developmental version, not recommended for production uses. It's fine for testing or experimenting, but if you have a Web site that users are accessing, I recommend not using a developmental version.

MySQL 4.0 and 4.1 are stable versions, recommended for use on active Web sites. MySQL is maintaining and improving both versions. The current versions are MySQL 4.0.24 and 4.1.11.

Version 4.1 added many new features and is the currently recommended version. If you don't have an existing MySQL server, install MySQL 4.1.

If you upgrade from version 4.0 to version 4.1, one change, starting with version 4.1.1, is longer passwords for MySQL accounts. That is, when you set a password for a new account using `SET PASSWORD`, `PASSWORD()`, or `GRANT`, the password is longer (and more secure) in 4.1 than in 4.0. Therefore, after you upgrade, you need to run the `mysql_fix_privilege_tables` script that is provided with the MySQL installation. This script changes the tables in MySQL that hold the account and password information, making the password column wider to hold the new, longer passwords. In addition, you need to access the database with a client that understands MySQL 4.1 passwords, such as the `mysql` client provided with MySQL version 4.1. (See <http://dev.mysql.com/doc/mysql/en/password-hashing.html> for more information on passwords in version 4.1.)

This book avoids the use of complex SQL queries, making the applications as easy to read and understand as possible. All SQL queries used in the applications in this book can run with either version 4.0 or 4.1. However, the functions used in PHP might or might not run correctly. See the following section for information on PHP versions.

PHP

Currently, PHP is maintaining two versions: PHP 4 and PHP 5. The current versions are PHP 4.3.11 and PHP 5.0.4.

PHP 5 is a major change from PHP 4. Enhancing object-oriented programming features was an important goal in the development of PHP 5. The creation and use of objects runs much faster, many object-oriented features have been added, and exceptions are introduced. Programmers who prefer object-oriented programming will be much happier with PHP 5. Most object-oriented programs that run with PHP 4 can run under PHP 5.

With PHP 5, the directory structure was changed. The executable programs have different names. The extension files are located in a different directory. Functions were added or enhanced. (For a complete list of new functions, see www.php.net/manual/en/migration5.functions.php.)

Each application provides procedural scripts and object-oriented programs. The procedural scripts in this book run with either PHP 4 or PHP 5, with the exception of the MySQL function calls. See the following section, “PHP and MySQL together,” for further information on the MySQL function calls. The object-oriented programs in this book run only with PHP 5.

PHP and MySQL together

PHP interacts with MySQL by using built-in functions. Currently, PHP provides two sets of functions for use when accessing MySQL databases: the MySQL extension and the MySQL Improved extension. The MySQL Improved extension was made available with PHP 5 for use with MySQL 4.1.

When you install PHP, you activate either the MySQL or the MySQL Improved extension. PHP 4 activates MySQL automatically during installation. You don’t need to activate it yourself. PHP 4 activates the MySQL extension. The MySQL Improved extension isn’t available with PHP 4. You can use the MySQL extension with MySQL 4.1; you just can’t use some of the new version 4.1 features.

PHP 5 doesn’t activate MySQL automatically. You need to enable MySQL support yourself either by using the installation option — with-mysql or with-mysqli — on Linux/Mac or by uncommenting one of the following lines in `php.ini`:

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

In general, it’s best to use `mysql` with MySQL version 4.0 and `mysqli` with MySQL version 4.1.

To access MySQL from a PHP script, you use the appropriate functions, depending on which extension you enabled. The functions are similar to the following:

```
$cxn = mysql_connect($host,$userid,$password);  
$cxn = mysqli_connect($host,$userid,$password);
```

The applications in this book use the `mysqli` functions. Consequently, you must use PHP 5 to run these scripts in their current format. However, if you need to run the applications with PHP 4, you just need to use the `mysql` function calls instead of the `mysqli` calls. If you revise the script and change the `mysqli` functions to `mysql`, you need to change the format of some of the functions.

In the preceding `mysql_connect` functions, the syntax of the two function calls is the same. However, many of the function calls differ slightly, such as the following:

```
$db = mysql_select_db("dbname");  
$db = mysqli_select_db($cxn, "dbname");
```

The `mysqli` function requires a database connection parameter before the database name. Other functions require similar minor changes. Appendix C shows the differences between `mysql` and `mysqli` syntax for the functions used in this book.

Using the Application Source Code

All the code for the applications in this book is provided on the CD. Each application is in its own directory. If you copy all the files from a specific directory to your Web space, you can run the application in your browser.

Choosing a location

Copy all the files from the CD directory to your Web space. You might want to put all the files into a subdirectory in your Web space, such as `c:\program files\apache group\apache\http\catalog`. The files include three types of files:

- ✓ **PHP scripts:** The files contain the scripts with the PHP code that provides the application functionality. PHP script files end with a `.php` extension.
- ✓ **Include files:** The files are called by using `include` statements in the PHP scripts. Include files end with a `.inc` extension.
- ✓ **Classes:** The files contain class definitions for object-oriented programs. The files are called at the beginning of the PHP scripts using `include` statements. Class files end with a `.class` extension.

If all the files are together in a single directory, the application runs. However, you might want to organize the files by putting them in subdirectories. If you put the files in subdirectories, you need to modify the script to use the correct path when including or calling the files.

One of the include files, named `Vars.inc`, contains the sensitive information needed to access the MySQL database. You should secure this file by putting it into your *include directory* — a directory where PHP looks for the files specified in an `include` statement. The include directory can be located outside your Web space, where visitors to your Web page cannot access it. You set up your include directory in the `php.ini` file. Look for the `include_path` setting. If the line starts with a semicolon (;), remove the semicolon. Add the path to the directory you want to use as your include directory. For example, you could use one of the following statements:

```
include_path=".;c:\include";           #Windows
include_path=".: /include";             #Linux
```

Both of these statements specify two directories where PHP looks for include files. The first directory is dot (meaning the current directory), followed by the second directory path. You can specify as many include directories as you want, and PHP searches through them for the include file in the order in which they are listed. The directory paths are separated by a semicolon for Windows and a colon for Linux.

If you don't have access to `php.ini`, you can set the path in each individual script by using the following statement:

```
ini_set("include_path", "c:\hidden");
```



This statement sets the `include_path` to the specified directory only while the program is running. It doesn't set the directory for your entire Web site.

The catalog application in the book includes images, but the images aren't included on the CD. Any catalog you implement will need specific product pictures. The application expects to find image files in a subdirectory named `images`.

Understanding the PHP code

The PHP code in the applications consists of only basic PHP statements. It doesn't use advanced PHP concepts or statements. Anyone who has a basic understanding of PHP can understand the code in the applications. You don't need to be an expert.

In the application, most of the code is included in the main PHP script(s). When building PHP scripts for an application, good programming practice dictates that you look for opportunities to use functions. Any time you find yourself using the same code more than once, you can place the code in a function and call the function at the appropriate locations in the script.

In the applications in this book, I don't use functions nearly as often as I could (or should). I believe that you can understand the code and follow its flow more easily when the code is in a single file, rather than when you must jump from page to page and back again, looking for the listing of functions. So, I present the code in the listings in a less disjointed manner — in fewer files showing the code in a top-down listing. In the explanation of the code, I point out locations where functions would be better coding style.

After each listing, I explain the code. Numbers in the explanation refer to line numbers shown in the code listing. I assume you know how control structures work in PHP and can follow the program flow. I provide some general description and some detailed description for more difficult or complex coding blocks.

Procedural versus object-oriented programs

Each application in this book is built with both procedural code and object-oriented code. That means that the CD contains two sets of independent programs for each application in the book. The mailing list application, described in the bonus chapter on the CD, however, is provided only with procedural code.

I am providing both types of code with the intention of producing a useful book for the following readers:

- ✓ **Inexperienced PHP programmers who have written only procedural code and who need to build an application for a real-world Web site:** You can install and use the procedural version of the application.
- ✓ **Programmers experienced with procedural programs in PHP who want to find out how to write object-oriented code in PHP:** You can compare the two versions to understand how to build object-oriented code. Appendixes A and B provide the concepts and syntax of object-oriented programming.
- ✓ **Programmers experienced in writing object-oriented code in another language who want to build an object-oriented application in PHP:** You can install and use the object-oriented version of the application. Appendix B describes the syntax of object-oriented programming in PHP.

Procedural and object-oriented methods are more than simply different syntax. As I describe in Appendix A, object-oriented programming is a different way of approaching programming projects. In the object-oriented approach, the programming problem is modeled with objects that represent the components of the programming problem. The objects store information and can perform needed tasks. The code that defines the object is stored in a class, which can then be used anywhere in the application that it's useful. The programmer using the class doesn't need to know anything about what's happening inside the class or how the class performs its tasks. The programmer can just use it. Thus, one programmer can develop a class that works in programs for many other programmers.

Developing really large, complex applications, involving several programmers or teams of programmers, is pretty difficult without using object-oriented programming. With object-oriented programming, programmers can develop their parts of the application independently. In addition, if something needs to be changed later, only the class with the change is affected. The other components of the application need not change. For the same reasons, maintenance of the application is much easier.

Modifying the Source Code

In most cases, you need to modify the application code. For one thing, the Web page design is very plain. Nothing in the page design will excite visitors or win you that Designer of the Year award. So, you undoubtedly want to customize the look and feel of the page. If you're adding one of these applications to an existing Web site, you can modify these pages to look like the existing page. Or, you might want to design something creative to impress your customers. If nothing else, you surely want to add your logo.

Because the source code provided with this book is a simple text file, you can use your favorite text-editing tool to modify the PHP source code files. You wouldn't be the first person to create scripts with vi, Notepad, or WordPad. However, you can find tools that make script editing much easier.



Check out programming editors and Integrated Development Environments before creating your PHP scripts. These tools offer features that can save you enormous amounts of time when building your application. So download some demos, try out the software, and select the one that suits you best. You can take a vacation on the time you save later.

Programming editors

Programming editors offer many features specifically for writing programs. The following features are offered by most programming editors:

- ✓ **Color highlighting:** Highlight parts of the script — such as HTML tags, text strings, keywords, and comments — in different colors so they're easy to identify.
- ✓ **Indentation:** Automatically indent inside parentheses and curly braces to make scripts easier to read.
- ✓ **Line numbers:** Add temporary line numbers. This is important because PHP error messages specify the line where the error was encountered. It would be cumbersome to have to count 872 lines from the top of the file to the line that PHP says is a problem.
- ✓ **Multiple files:** You can have more than one file open at once.
- ✓ **Easy code inserting:** Buttons for inserting code, such as HTML tags or PHP statements or functions are available.
- ✓ **Code library:** Save snippets of your own code that can be inserted by clicking a button.

Many programming editors are available on the Internet for free or for a low price. Some of the more popular editors include the following:

- ✓ **Arachnophilia:** This multiplatform editor is written in Java. It's CareWare, which means it doesn't cost any money.
www.arachnoid.com/arachnophilia
- ✓ **BEdit:** This editor is designed for use on a Mac. BEdit sells for \$199.00. Development and support have been discontinued for BEdit Lite, which is free, but you can still find it and legally use it.
www.barebones.com/products/bbedit/index.shtml
- ✓ **EditPlus:** This editor is designed for use on a Windows machine. EditPlus is shareware, and the license is \$30.
www.editplus.com
- ✓ **Emacs:** Emacs works with Windows, Linux, and UNIX, and it's free.
www.gnu.org/software/emacs/emacs.html

- ✓ **HomeSite:** HomeSite is designed for use with Windows and will run you \$99.00.

www.macromedia.com/software/homesite

- ✓ **HTML-Kit:** This is another Windows editor that you can pick up for free.

www.chami.com/html-kit

- ✓ **TextWrangler:** This editor is designed for use on a Mac. It's developed and published by the same company that sells BBEdit. TextWrangler has fewer features than BBEdit, but has most of the major features useful for programmers, such as syntax highlighting and automatic indenting. And it's much cheaper than BBEdit — as in *free*.

www.barebones.com/products/textwrangler/index.shtml

- ✓ **Vim:** These free, enhanced versions of vi can be used with Windows, Linux, UNIX, and Mac OS.

www.vim.org

Integrated Development Environment (IDE)

An *Integrated Development Environment (IDE)* is an entire workspace for developing applications. It includes a programming editor as well as other features. Some features included by most IDEs are the following:

- ✓ **Debugging:** Has built-in debugging features.
- ✓ **Previewing:** Displays the Web page output by the script.
- ✓ **Testing:** Has built-in testing features for your scripts.
- ✓ **FTP:** Has built-in ability to connect, upload, and download via FTP. It also keeps track of which files belong in which Web site and keeps the Web site up to date.
- ✓ **Project management:** Organizes scripts into projects, manages the files in the project, and includes file checkout and check-in features.
- ✓ **Backups:** Makes automatic backups of your Web site at periodic intervals.

IDEs are more difficult to get familiar with than programming editors. Some are fairly expensive, but their wealth of features can be worth it. IDEs are particularly useful when several people will be writing scripts for the same application. An IDE can make project coordination much simpler and make the code more compatible.

The following are popular IDEs:

- ✓ **Dreamweaver MX:** This IDE is available for the Windows and Mac platforms. It provides visual layout tools so you can create a Web page by dragging elements around and clicking buttons to insert elements. Dreamweaver can write the HTML code for you. It includes the HomeSite editor so you can write code. It also supports PHP. Dreamweaver will set you back \$399.00.

www.macromedia.com/dreamweaver

- ✓ **Komodo:** Komodo is offered for the Linux and Windows platforms. It's an IDE for open-source languages, including Perl and Python, as well as PHP. It's offered for \$29.95 for personal or educational use, and \$295.00 for commercial use.

www.activestate.com/Products/Komodo

- ✓ **Maguma:** Maguma is available for Windows only. It's an IDE for Apache, PHP, and MySQL on Windows and comes in two versions at different costs: Maguma Studio Desktop and Maguma Studio Enterprise, which offers features for huge sites with multiple servers. Maguma Studio for PHP is a free version with support for PHP only.

www.maguma.com

- ✓ **PHPedit:** This free IDE is available only for Windows.

www.phpedit.net/products/PHPedit

- ✓ **Zend Studio:** Zend Studio is offered for the Linux and Windows platforms. This IDE was developed by the people who developed the Zend engine, which is the engine under the hood of PHP. These people know PHP extremely well. Zend Studio will cost you \$195.00.

www.zend.com/store/products/zend-studio.php

Planning Your Application

Planning is an essential part of building your application. The application design is the blueprint for building your application. Your plan should be complete enough to keep your project on track toward its goal and to ensure that all the needed elements and features are included in the plan.

Even if you're using one of the applications in this book, you need to develop your own plan first. With your plan as a guide, you can see whether the application meets all your needs as is or whether you need to modify the application, adding or removing features so the application fits your needs perfectly.



The larger and more complex your application is, the more planning is required. An application that displays `Hello World` on the screen, with five lines in the script, built by one person, requires little planning. The Amazon Web site requires mammoth planning.

Planning the software

Planning the application software includes the following steps:

- 1. Identify the goal or goals of the application.**

Is the application intended to collect information from users? Sell products to users? Entertain users? Create a community of users?

- 2. Develop a list of tasks that the application needs to perform in order to meet the goal.**

For instance, if the goal is to sell products, the application needs to, at the least, display the products information, provide a means for the customer to select a product, collect the information needed to fill the order, and charge the customer for the product.

- 3. Plan the database.**

Decide what information needs to be stored. Decide how to store it for quick and easy access.

- 4. Develop a detailed plan for the methods to use in carrying out the general behavior tasks that you develop in Step 2.**

For instance, “collect the information needed to fill the order” can expand to:

- a. Display a form.
- b. Verify the information submitted in the form.
- c. Store the information in a database.

- 5. Plan the Web pages.**

How many Web pages need to be displayed? For instance, do you need a form and a feedback page? A product information page? A page that looks like a chess board? Design the look and feel of the Web pages.

Additional planning

The application plan is a basis for other project planning. You need to develop a schedule for your project. You also need to develop a resource plan.

Developing a schedule

The most important date for your project is the date the application goes live. That is, the date when outside users can begin using the application. In some cases, you are given the date, and you need to determine the resources you need to meet the date. In other cases, you have finite resources and you must estimate the date when the application will be ready to go live.

You can use the application plan to estimate the number of man hours needed to build the application. The calendar time required depends on how many programmers are working on the application. A project that takes 100 hours will take one programmer 2½ weeks to finish, assuming the programmer makes optimum use of every hour of a 40-hour week. Two programmers can finish the application (theoretically) in 1¼ weeks.

When scheduling, be sure to allow some time for those tasks required by Murphy's Law — rewriting vanished code, time lost due to bubonic plague, electric outages caused by lightening, and so forth. Also, be sure to allow time for testing and writing documentation.

When planning your timeline, remember that some tasks can proceed simultaneously, but other tasks must proceed sequentially. For instance, if one of your tasks is to buy a computer, the programming can't start until after the computer arrives.



Project management software can be useful when developing a schedule. It keeps track of the tasks, the resources, and the critical events along the way. It charts the tasks into a timeline, showing the *critical path* — the series of tasks that must be completed on time in order for the project to be completed on time.

Planning the resources

Resources include both human and material resources. Your software plan and the project delivery date determine the human resources required. Your plan needs to schedule the availability of the human resources. If you need to hire more people, include that time in your schedule. If you plan to use existing staff, plan the project time into their schedules.

Make sure that material resources are available when they're needed. For instance, if you need to buy a new computer for the project, you need to start the purchasing process so that the computer will arrive before it's needed. For the applications in this book, you need PHP and MySQL, so you need to plan their availability. Is the software currently installed? Does it need upgrading? If it's not installed, who will install and administer it? When can the administrator have it available?

Include a list of resources needed, both human and material, as part of your project plan. For projects such as the applications in this book, personnel and computers are required resources. However, for your specific project, many other resources might be needed. For instance, artwork or photos of products may be required. Written copy for an online catalog might be needed. You might want a reference book or two. A list of resources can help prevent dead time spent waiting for needed resources.