如果要编写一个类似的脚本，可能还希望传输一些其他的数据。例如，如果要连接到一个外部URL，可能需要传输一些用户输入参数。如果打算这样做，使用函数urlencode()是一个好主意。此函数将接受一个字符串并将其转换为适合URL的格式。例如，将空格转换为“+”符号。可以按如下方式调用此函数：

```
$encodedparameter=urlencode($parameter);
```

这种方法的问题是获得信息的源站点可能会改变其数据格式，这样就使脚本无法正常运行。

20.4　使用网络查找函数

PHP提供一组查找函数，这些函数可以用来检查主机名称、IP地址、邮件交换等信息。例如，如果要创建一个目录站点，例如Yahoo！，当用户提交一个新的URL时，我们可能会自动检查URL所指向的主机和此站点的联系信息是否有效。这样，当一个访问者浏览一个网站并发现此网站不存在或电子邮件无效时，可以节省进一步的操作。

程序清单20-2给出了目录提交的HTML表单。

程序清单20-2　directory_submit.html——提交表单的HTML

```
<html>
<head>
<title>Submit your site</title>
</head>
<body>
<h1>Submit site</h1>
<form method=post action="directory_submit.php">
URL:<input type=text name="url"size=30 value="http://”><br/>
Email contact:<input type=text name="email'size=23><br/>
<input type="submit"name="Submit site">
</form>
</body>
</html>
```

这是一个非常简单的表单，图20-2所示的就是带有输入样本数据的表单。



图　20-2　目录提交通常要求提交URL和某些详细的联系信息，这样可以方便目录管理员在添加了站点以后通知我们

当点击提交按钮时，我们首先要检查该URL是否对应一台真实的主机，其次，还要检查电子邮件地址的主机部分是否也对应真实的机器。我们已经编写了这些检查脚本，它的输出结果如图20-3所示。



图　20-3　此版本的脚本显示对URL主机名和电子邮件地址的检查结果——一个实际的产品版本可能不会显示这些结果，但看看检查返回的信息是非常有趣的

执行这些检查的脚本使用了两个来自PHP网络函数集中的函数——gethostbyname()和dns_get_mx()。全部脚本如程序清单20-3所示。

程序清单20-3　directory_submit.php——用于检查URL和电子邮件地址的脚本

```
<html>
<head>
<title>Site submission results</title>
</head>
<body>
<h1>Site submission results</h1>
```

```
<?php
//Extract form fields
$url=$_REQUEST['url'];
$email=$_REQUEST['email'];
//Check the URL
$url=parse_url($url);
$host=$url['host'];
if(!($ip=gethostbyname($host)))
{
echo'Host for URL does not have valid IP';
exit;
}
echo"Host is at IP$ip<br>";
//Check the email address
$email=explode('@',$email);
$emailhost=$email[1];
//note that the dns_get_mx()function is*not implemented*in
//Windows versions of PHP
if(!dns_get_mx($emailhost,$mxhostsarr))
{
echo'Email address is not at valid host';
exit;
}
echo'Email is delivered via:';
foreach($mxhostsarr as$mx)
echo"$mx";
//If reached here,all ok
echo'<br>All submitted details are ok.<br>';
echo'Thank you for submitting your site.<br>'
.'It will be visited by one of our staff members soon.'
//In real case,add to db of waiting sites...
?>
</body>
</html>
```

现在，我们来了解一下这个脚本的有趣部分。

首先，获取URL并将其作为函数parse_url()的参数。该函数将返回包含URL不同部分的相关数组。该数组的可用信息部分分别是“模式”、“用户”、“传递”、“主机”、“端口”、“路径”、“查询”和“代码段”（scheme、user、pass、host、port、path、query、fragment）。一般情况下，并不需要所有这些信息。如下所示的例子说明它们是如何组成一个有效的URL。

假设有一个如下所示的URL：http://nobody:secret@example.com:80/script.php?variable=value#anchor，数组的每一个元素值分别是：

■scheme:http

■user:nobody

■pass:secret

■host:example.com

■port:80

■path:/script.php

■query:variable=value

■fragment:anchor

在directory_submit.php脚本中，我们只需要主机信息，因此可以按如下操作从数组中取出：

```
$url=parse_url($url);
$host=$url['hos'];
```

在完成以上操作后，如果主机是在域名服务（DNS）中，可以获得此主机的IP地址。我们可以通过函数gethostbyname()获得一个主机的

IP地址。如果主机存在，此函数就返回其IP地址，如果不存在，函数将返回false：

```
$ip=gethostbyname($host);
```

也可以使用函数gethostbyaddr()来实现，此函数以IP作为参数并返回主机名。如果顺序地调用这个函数，最后获得的主机名将与开始时的主机名不同。这可能意味着网站正在使用一个虚拟主机服务。在这个服务中，一个物理主机和IP地址具有多个域名。

如果URL是有效的，接下来就需要检查电子邮件地址。首先，通过调用函数explode()将邮件地址分割成用户名和主机名两部分：

```
$email=explode('@',$email);
$emailhost=$email[1];
```

当得到主机部分的地址后，可以通过函数dns_get_mx()检查是否有邮件可以到达的确切地方：

```
dns_get_mx($emailhost,$mxhostsarr);
```

该函数将返回一个邮件地址的一组邮件交换（Mail Exchange,MX）记录，该地址由数组$mxhostarr提供。

MX记录存储在DNS中，而且其查找方式类似于主机名的查找方式。MX列出的机器不一定是邮件最终到达的机器。相反，它是一台知道邮件发送路由的机器（可能不止一台，因此这个函数返回一个数组而不是一个主机名字符串）。如果DNS中没有MX记录，那么该邮件就没有可发送的目的地。

请注意，在PHP的Windows版本中，并没有实现dns_get_mx()函数。如果使用Windows，必须使用PEAR::Net_DNS包（http://pear.php.net/package/NET_DNS），该包包含了具有此功能的函数。

如果这些检查都正确，可以将这个表单的数据输入到数据库，以便工作人员以后复查。

除了刚才已经使用的函数外，我们还可以使用更普通的函数checkdnsrr()，这个函数以主机名作为参数，如果在DNS中有其记录，函数将返回true。

20.5　备份或镜像一个文件

文件传输协议（FTP）是用来在网络主机之间传输文件的协议。就像使用HTTP连接一样，使用PHP，可以在FTP中使用fopen()函数和其他不同的文件函数来连接一个FTP服务器，并在客户端和FTP服务器之间传送文件。但是，标准的PHP安装也提供了一整套专门适用于FTP的函数。

在默认情况下，这些函数并没有内置在PHP的标准安装中。要在UNIX下使用这些函数，必须运行带有--enable-ftp命令选项的configure程序，然后再次运行make文件。如果使用标准的Windows安装，FTP函数将被自动启用。（关于如何配置PHP的详细信息，请参阅附录A“安装PHP和MySQL”。）

20.5.1　使用FTP备份或镜像一个文件

对于从一台主机到另一台主机之间的文件移动和复制来说，FTP函数是非常有用的。FTP最常见的应用可能就是在另一个位置备份Web站点或者镜像文件。接下来，我们来了解一个使用FTP函数镜像文件的简单例子，如程序清单20-4所示。

程序清单20-4　ftp_mirror.php——一个从FTP服务器下载新版本文件的脚本

```
<html>
<head>
<title>Mirror update</title>
</head>
<body>
<h1>Mirror update</h1>
<?php
//set up variables-change these to suit application
$host='ftp.cs.rmit.edu.au';
$user='anonymous';
$password='me@example.com';
```

```
$remotefile='/pub/tsg/teraterm/ttssh14.zip';
$localfile='/tmp/writable/ttssh14.zip';
//connect to host
$conn=ftp_connect($host);
if(!$conn)
{
echo'Error:Could not connect to ftp server<br/>';
exit;
}
echo"Connected to$host.<br/>";
//log in to host
$result=@ftp_login($conn,$user,$pass);
if(!$result)
{
echo"Error:Could not log on as$user<br/>";
ftp_quit($conn);
exit;
}
echo"Logged in as$user<br/>";
//check file times to see if an update is required
echo'Checking file time...<br/>';
if(file_exists($localfile))
{
$localtime=filemtime($localfile);
echo'Local file last updated';
echo date('G:i j-M-Y',$localtime);
echo'<br/>';
}
else
$localtime=0;
$remotetime=ftp_mdtm($conn,$remotefile);
if(!($remotetime>=0))
{
//This doesn't mean the file's not there,server may not support mod
time
echo'Can\'t access remote file time.<br/>';
```

```
$remotetime=$localtime+1;//make sure of an update
}
else
{
echo'Remote file last updated';
echo date('G:i j-M-Y',$remotetime);
echo'<br/>';
}
if(!($remotetime>$localtime))
{
echo'Local copy is up to date.<br/>';
exit;
}
//download file
echo'Getting file from server...<br/>';
$fp=fopen($localfile,'w');
if(!$success=ftp_fget($conn,$fp,$remotefile,FTP_BINARY))
{
echo'Error:Could not download file';
ftp_quit($conn);
exit;
}
fclose($fp);
echo'File downloaded successfully';
//close connection to host
ftp_quit($conn);
?>
</body>
</html>
```

在特定情况下，运行以上脚本的输出结果如图20-4所示。

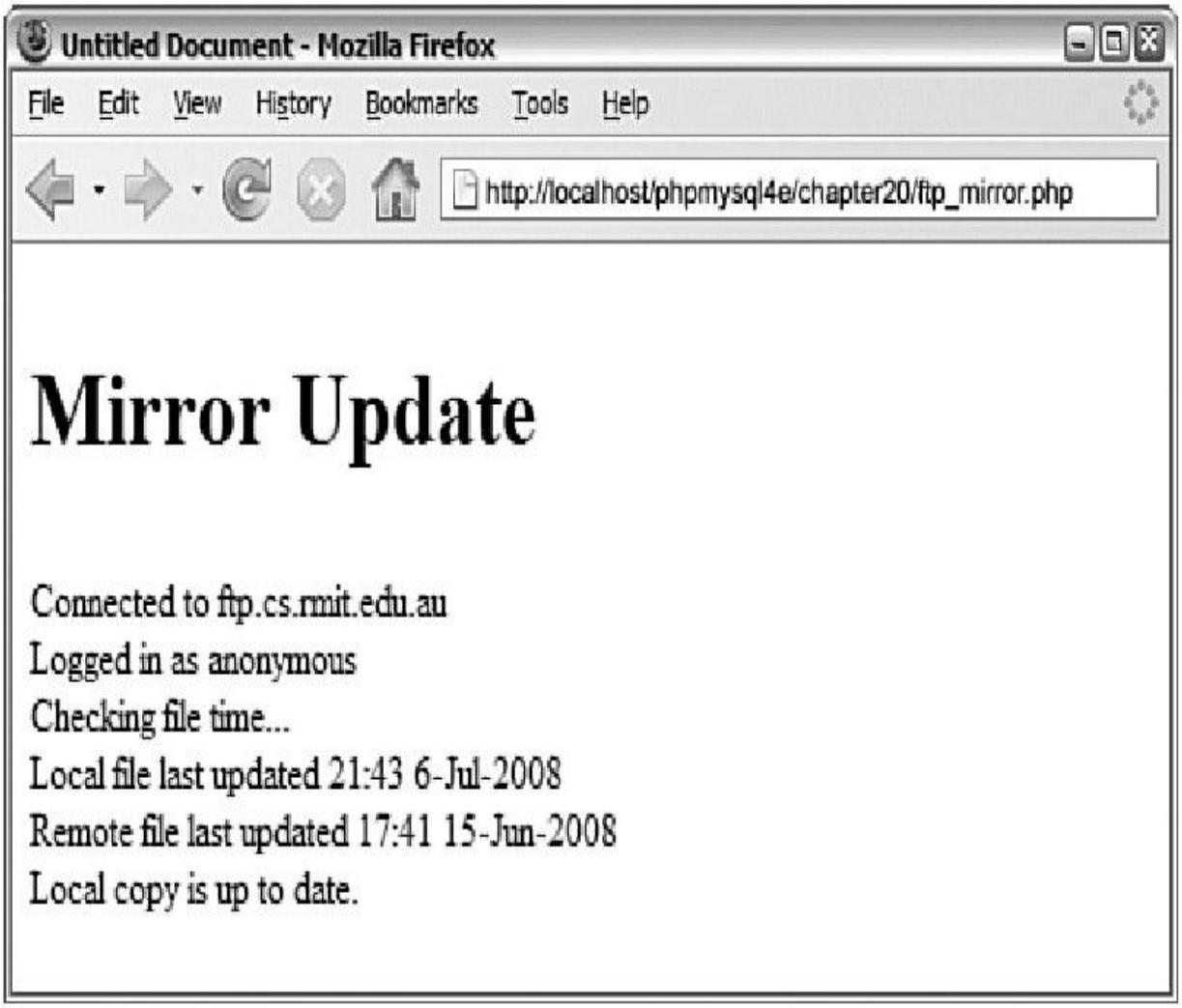


图　20-4　该FTP镜像脚本将检查文件的本地版本是否是最新版本，如果不是，就下载一个新的版本

这是一个非常普通的脚本。可以看到，在此脚本的开始处设置了一些变量：

```
$host='ftp.cs.rmit.edu.au';
$user='anonymous';
$password='me@example.com';
$remotefile='/pub/tsg/teraterm/ttssh14.zip';
$localfile='/tmp/writable/ttssh14.zip';
```

变量$host包含了即将连接的FTP服务器的名称，$user和$password对应于登录的用户名和密码。

许多FTP网站都支持匿名登录（anony-mous），匿名登录是指任何人都可以使用这个用户名来连接FTP服务器。匿名登录不需要密码，但通常的做法是把电子邮件作为密码，这样管理员能够知道用户来自什么地方。在这里，我们沿袭了这个惯例。

变量$remotefile包含我们将要下载的文件的路径。在这个例子中，我们要下载并镜像Tera Term SSH的一个本地拷贝，以及一个适用于Windows平台的SSH客户端（SSH表示安全Shell。它是Telnet的加密形式）。

变量$localfile包含了下载文件存储在机器上的本地路径。在这个例子中，我们创建了/tmp/writable目录，该目录的权限设置允许PHP对其执行写操作，也就是可以写入一个文件。无论操作系统是什么，必须创建这个目录，这样这个脚本才能正确运行。如果操作系统具有严格的权限机制，必须确认它们允许这个脚本执行写操作。我们应该能够根据需要修改这些变量和脚本。

该脚本的基本执行步骤与在命令行中手动使用FTP传输文件一样：

1.连接远程FTP服务器。

2.登录（通过用户名或匿名登录）。

3.检查远程文件是否已经更新。

4.如果更新过，下载此文件。

5.关闭FTP连接。

下面，让我们依次讨论这些步骤。

1.连接远程FTP服务器

这一步与在Windows或者UNIX平台的命令提示符下输入如下命令等价：

```
ftp hostname
```

在PHP中，通过如下所示代码完成此步骤：

```
$conn=ftp_connect($host);
if(!$conn)
{
echo'Error:Could not connect to ftp server<br/>';
exit;
}
echo"Connected to$host.<br/>";
```

在这里，我们调用的函数是ftp_connect()。此函数以主机名作为参数并返回连接的句柄或者false（如果不能建立连接）。这个函数也能够以要连接的主机端口号作为可选的第二个参数（这里没有用到）。如果不指定一个端口号，此函数将使用默认值21（FTP的默认端口）。

2.登录到FTP服务器

下一步是使用一个使用特定用户名和密码登录到FTP服务器上。可以通过函数ftp_login()来完成此步骤：

```
$result=@ftp_login($conn,$user,$pass);
if(!$result)
{
echo"Error:Could not log on as$user<br/>";
ftp_quit($conn);
exit;
}
echo"Logged in as$user<br/>";
```

这个函数需要3个参数：一个FTP连接句柄（通过函数ftp_connect()获得）、用户名和密码。如果用户能够登录，此函数将返回true，如果

不能登录，函数将返回false。注意我们在这一行的开始处放置一个“@”符号来抑制错误。这是因为如果用户不能登录，他将在浏览器窗口中获得一个PHP警告信息。通过测试$result变量的内容，可以捕获这个错误，并可以自己定制更多的、更友好的用户出错信息。

请注意，如果登录尝试失败，实际上将通过ftp_quit()函数关闭FTP连接；稍后将详细介绍该函数。

3.检查文件更新时间

假设我们打算更新文件的本地副本，比较明智的做法是首先检查文件是否需要更新，因为如果文件是最新的，就无须重新下载此文件，特别是当它是一个很大的文件的时候。这将避免不必要的网络通信量。现在，让我们来查看实现检查文件更新时间的代码。

文件时间是我们使用FTP函数而不是其他更简单的文件函数的原因。文件函数可以很容易读，而且在某些情况下，也可以通过网络接口写文件，但是大多数状态函数，例如filemtime()，无法执行远程操作。这一点将在以后发生变化。

要确定是否需要下载一个文件，可以通过函数file_exists()确认一个文件的本地副本是否存在。如果没有，显然需要下载此文件。如果此文件存在，通过函数filemtime()获得文件的最后修改时间，并把它存储在变量$localtime中。如果此文件不存在，设置变量$localtime为0，这样，此文件将会比任何可能的远程文件修改时间都要“老”。

```
echo'Checking file time...<br/>';
if(file_exists($localfile))
{
$localtime=filemtime($localfile);
echo'Local file last updated';
echo date('G:i j-M-Y',$localtime);
echo'<br/>';
}
else
$localtime=0;
```

（在第2章“数据的存储与检索”和第19章“与文件系统和服务器的交互”中，我们已经分别介绍了函数file_exists()和filemtime()的使用）。

解决了本地时间的问题后，我们需要获得远程文件的修改时间。可以通过函数ftp_mdtm()获得此信息，如下所示：

```
$remotetime=ftp_mdtm($conn,$remotefile);
```

这个函数带有两个参数（FTP连接句柄和远程文件的路径），并返回文件最后修改时间的UNIX时间戳或者-1（如果存在某种错误的话）。不是所有的FTP服务器都支持这种特性，因此通过这个函数可能得不到一个有用的结果。在这种情况下，可以选择手动设置，将变量$localtime加1，使变量$remotetime比$localtime“新”。这样，就可以确保能够下载文件，如下代码所示：

```
if(!($remotetime>=0))
{
//This doesn't mean the file's not there,server may not support mod time
echo'Can't access remote file time.<br>';
$remotetime=$localtime+1;//make sure of an update
}
else
{
echo'Remote file last updated';
echo date('G:i j-M-Y',$remotetime);
echo'<br>';
}
```

当我们拥有二者的时间时，就可以对它们进行比较，确认是否需要下载这个文件：

```
if(!($remotetime>$localtime))
{
echo'Local copy is up to date.<br>';
```

```
exit;
}
```

4.下载文件

到这一步，我们就可以从服务器上下载文件。

```
echo'Getting file from server...<br>';
$fp=fopen($localfile,'w');
if(!$success=ftp_fget($conn,$fp,$remotefile,FTP_BINARY))
{
echo'Error:Could not download file';
fclose($fp);
ftp_quit($conn);
exit;
}
fclose($fp);
echo'File downloaded successfully';
```

正如我们在前面代码所见到的，我们通过函数fopen()打开一个本地文件。在成功打开该文件后，调用函数ftp_fget()，该函数将下载这个文件并存储到本地文件中。ftp_fget()函数带有4个参数。前3个非常简单：FTP连接句柄、本地文件句柄和远程文件路径。第4个参数是FTP模式。

对于FTP传输，有两种模式：ASCII和二进制。ASCII模式用于传输文本文件（也就是说，文件全部由ASCII字符组成），二进制模式用于传输所有其他类型的文件。二进制模式在传输一个文件的过程中不会修改这个文件，而ASCII模式将回车换行字符转换成适用于系统的特定字符（在UNIX下为\n，Windows下为\r\n，而Macintosh下为\r）。

PHP的FTP库有两个预定义的常量，FTP_ASCII和FTP_BINARY，它们分别代表这两种模式。我们需要确定哪种模式适合文件，并将相应的常量作为第4个变量传给函数ftp_fget()。在这个例子中，传输的是一个压缩文件，因此我们使用了FTP_BINARY模式。

如果所有的这些操作都是正常执行，ftp_fget()函数将返回true；如果遇到错误，此函数将返回false。函数执行结果存储在变量$success中，这样用户可以知道此函数的运行情况。

在文件下载完成之后，需要调用函数fclose()关闭本地文件。

除了ftp_fget()函数外，还可以使用函数ftp_get()来实现此目的，此函数具有如下所示的原型：

```
int ftp_get(int ftp_connection,string localfile_path,
string remotefile_path,int mode)
```

此函数在许多方面与函数ftp_fget()相同，但是它不需要打开本地文件。传递给此函数的参数为本地文件的系统文件名而不是文件句柄。

请注意，PHP中还没有与FTP命令mget等价的函数，此命令能够一次下载多个文件。我们必须多次调用ftp_fget()函数或者ftp_get()函数来代替mget命令。

5.关闭连接

在完成FTP连接之后，应该通过函数ftp_quit()来关闭此连接。

```
ftp_quit($conn);
```

你必须将FTP连接的句柄传递给此函数。

20.5.2 上传文件

如果想执行下载文件的反向操作，也就是要将文件从自己的服务器上复制到远程的机器上，可以使用两个函数，它们的功能基本上与函数ftp_fget()和ftp_get()相反。这两个函数就是ftp_fput()和ftp_put()。其函数原型如下所示：

```
int ftp_fput(int ftp_connection,string remotefile_path,int fp,int mode)
int ftp_put(int ftp_connection,string remotefile_path,
string localfile_path,int mode)
```

这些参数与_get函数的参数一样。

20.5.3 避免超时

当通过FTP传输文件时，可能会遇到的一个问题就是超过最大可执行时间。这种情况是否会发生是可以预知的，因为PHP将给出一个出错信息。如果服务器运行在一个非常慢或拥挤的网络上，或者正在下载一个很大的文件（如电影剪辑），这种情况就很可能发生。

配置文件php.ini定义了所有的PHP脚本最大可执行时间的默认值。此默认值设置为30s。

这样，就可以避免脚本失去控制地运行。然而，当正在通过FTP传输文件时，如果到其他网址的连接速度非常慢，或者文件很大时，文件传输需要的时间可能就会比这个默认值要大。

幸运的是，对于一个特定的脚本，我们可以通过函数set_time_limit()来修改这个最大可执行时间。调用这个函数将重新设置脚本所允许的最大可执行时间（以s为单位），计时是从调用此函数的时候开始。例如，如果调用函数：

```
set_time_limit(90);
```

这样，这个脚本将从这个函数的调用开始重新运行90s。

20.5.4　使用其他的FTP函数

在PHP中，还有许多其他非常有用的FTP函数。函数ftp_size()能够显示远程服务器上一个文件的大小。此函数原型如下所示：

```
int ftp_size(int ftp_connection,string remotefile_path)
```

此函数返回远程文件的字节数或者-1（如果有错误的话）。但是，该函数并不被所有的服务器所支持。

函数ftp_size()的一个实际应用是计算出在特定传输速率一次文件传输所需的最大可执行时间。如果知道文件大小和连接速度，可以估算出传输可能需要的时间，并可以根据实际情况调用函数set_time_limit()。

通过如下所示代码，可以获得并显示远程FTP服务器特定目录上的文件列表：

```
$listing=ftp_nlist($conn,dirname($remotefile));
foreach($listing as$filename)
echo"$filename＜br＞";
```

此代码通过函数ftp_nlist()的调用获得特定目录中的文件名的列表。

对于FTP函数，凡是可能通过FTP命令行完成的事情，几乎都可以通过FTP函数来完成。

在PHP在线指南中，可以找到对应于每个FTP命令的特殊函数，其URL如下所示：http://php.net/manual/en/ref.ftp.php。

一个例外是命令mget（multiple get，也就是获得多个文件），但是可以通过函数ftp_nlist()获得文件列表并根据需要选择文件。

20.6 进一步学习

在本章中，我们介绍了许多基础知识。关于这些主题，还有许多内容。有关个别协议的信息和它们的工作原理，可以参考相应的RFC标准，其网址如下所示：

http://www.rfc-editor.org/.

也可以在W3C组织找到关于某些协议的信息：

http://www.w3.org/Protocols/.

还可以参考有关TCP/IP的书籍，例如，由Andrew Tanenbaum编著的《计算机网络》。

20.7 下一章

我们可以进入第21章的学习了。在第21章“日期和时间的管理”中，我们将讨论PHP日期和日历函数。我们将了解如何把用户输入的格式转换为PHP格式，再转换为MySQL格式，以及进行相互的转换。

第21章　日期和时间的管理

在本章中，我们将介绍日期和时间的检查和格式化，以及日期在不同格式之间的转换。

当要在MySQL日期格式和PHP日期格式、UNIX日期格式与PHP日期格式，以及用户输入HTML表单的日期格式之间进行转换时，这是非常重要的。

在本章中，我们将主要介绍以下内容：

■在PHP中获取日期和时间

■在PHP日期格式和MySQL日期格式之间进行转换

■计算日期

■使用日历函数

21.1　在PHP中获取日期和时间

在第1章“PHP快速入门教程”中，我们讨论了使用date()函数来获取并格式化PHP中的日期和时间。在本章中，我们仍将讨论该函数，并且更详细地介绍PHP中其他的时间和时间函数。

21.1.1　使用date()函数

我们能够回忆起来，date()函数带有两个参数，其中有一个是可选的。第一个是格式字符串，第二个（即可选的一个）是UNIX时间戳。如果没有指定时间戳，在默认的情况下，date()函数将返回当前的日期和时间。该函数可以返回一个格式化后表示适当日期的字符串。

date()函数的常见调用方式如下所示：

```
echo date('jS F Y');
```

以上调用将返回格式为"19th June 2008"的日期。表21-1给出PHP中所支持的日期格式代码。

表21-1 PHP的date()函数所支持的格式代码

代　码	描　述
a	上午或下午，两个小写字符表示，“am”或“pm”
A	上午或下午，两个大写字符表示，“AM”或“PM”
B	Swatch Internet时间，一种统一的时间模式。在http://www.swatch.com/可以找到关于该时间模式的详细信息
c	ISO 8601日期。其日期用YYYY-MM-DD表示。大写T用来间隔日期和时间。时间用HH:MM:SS来表示。最后，时区是用格林威治时间（GMT）的偏差来表示的，例如，2008-06-26T21:04:42+11:00（PHP 5.1.0已经引入这个格式代码）
d	两位数字表示的月份中的日期，带前导0，从“01”到“31”
D	3个缩略字符表示的星期，文本格式，从“Mon”到“Sun”
e	时区识别器（PHP 5.1.0）

（续）

代　码	描　述
F	年中的月份，全写，从“January”到“December”
g	日期中的时间，12小时制，无前导0，从“1”到“12”
G	日期中的时间，24小时制，无前导0，从“0”到“23”
h	日期中的时间，12小时制，带前导0，从“01”到“12”
H	日期中的时间，24小时制，带前导0，从“00”到“23”
i	小时中的分钟，带前导0，从“00”到“59”
I	夏令时制，以布尔值表示，若为夏令时，返回“1”，否则返回“0”
j	月份中的日期，数字型，无前导0，从“1”到“31”
l	星期，全称，从“Sunday”到“Saturday”
L	闰年，以布尔值表示，如果日期所在年是闰年，返回“1”，否则返回“0”
m	以两位数字表示的月份，带前导0，从“01”到“12”
M	以3个缩略字符表示的月份，文本格式，从“Jan”到“Dec”
n	年中的月份，以数字表示，无前导0，从“1”到“12”
o	ISO-8601的年份数。与“Y”具有相同值。不同点在于，如果ISO周数（W）属于上一年或下一年，将使用新的年份数（在PHP 5.1.0版本引入）
O	当前时区与格林威治时间之间小时时差。例如，+1600
r	RFC822格式的日期和时间。例如，Wed, 1 Jul 2008 18:45:30 + 1600（这是在PHP 4.0.4中新增加的）
s	带有前导0的秒钟时间数，其取值为“00”到“59”
S	序数，日期后缀，以两个字符表示，包括“st”、“nd”、“rd”或“th”，具体取决于日期数字后面的数字是什么
t	月份的天数，从“28”到“31”
T	服务器的时间区域设置，例如，“EST”
U	从1970年1月1日到某时刻总的秒数；也叫该日期的UNIX时间戳
w	星期，以单个数字表示，从“0”（星期日）到“6”（星期六）
W	一年当中的星期数，与ISO-8601格式相兼容（在PHP 4.1.0中新添加的）
y	两位数字表示的年份，例如，“08”
Y	四位数字表示的年份，例如，“2008”
z	数字表示的日期，从“0”到“365”
Z	与当前时区的时区差，单位为s（秒）。从“-43200”到“43200”

21.1.2　使用UNIX时间戳

date()函数的第二个参数是UNIX时间戳。我们对时间戳的意义做一个简单介绍。大多数UNIX系统保存当前日期和时间的方法是：保存格林威治标准时间从1970年1月1日零点起到当前时刻的秒数，以32位整列表示，其中1970年1月1日零点也叫UNIX纪元。如果对它不熟悉，这看起来有点深奥，但它是一个标准，而且整数适用于计算机处理。

UNIX时间戳是保存时间的一种紧凑简洁的方法，同时，它不会遭遇千年虫（Y2K）问题，该问题可以影响其他的紧凑或缩略的日期格式。然而，它们具有类似的问题，因为时间是通过一个32位的整数来表示的。如果软件需要处理1902年以前或2038年以后的事件，将会遇到一些问题。

在某些系统中（包括Windows），其范围更加有限。时间戳不能为负数，因此1970年以前的时间戳无法使用。要使代码具有可移植性，必须记住这一点。

我们不需要担心软件是否可以在2038年以后使用。时间戳没有固定的大小；它们是C语言的长整型，至少有32位。如果软件在2038年还会继续使用，那你的系统将肯定会使用了更大的类型。

虽然这是标准的UNIX惯例，但是即使在Windows服务器中运行PHP，这个格式仍然被date()函数和许多的PHP其他函数使用。唯一的不同就是，对于Windows来说，时间戳必须是正数。

如果要将一个日期和时间转变成UNIX时间戳，可以使用mktime()函数。该函数原型如下所示：

```
int mktime([int hour[,int minute[,int second[,int month[,
int day[,int year[,int is_dst]]]]]]])
```

除了最后一个参数is_dst外，其他参数的含义都很容易理解。参数is_dst表示该日期所示的时间是否是夏令时。如果是，可以将其设置为1，如果不是，设置为0（默认值），如果不知道，则设置为-1（默认

值）。在使用-1的情况下，PHP将根据所运行的系统来确定它。该参数是可选的，很少用到。

使用该函数需要避免的一个主要陷阱是其参数顺序非常不直观。参数的顺序不允许漏掉一个时间参数。如果对具体时刻不在乎，可以将0传给hour、minute和second参数。

可以从参数列的右边开始遗漏参数的值。如果参数为空，将默认为当前时间。因此，如下所示的调用：

```
$timestamp=mktime();
```

将返回当前日期和时间的UNIX时间戳。当然，也可以通过如下所示的调用获取当前的UNIX时间戳：

```
$timestamp=time();
```

time()函数不需要任何参数，而且通常返回当前日期和时间的UNIX时间戳。

另一个选项就是date()函数，正如我们已经讨论的。格式字符串"U"要求一个时间戳。

如下语句等价于上两个语句：

```
$timestamp=date("U");
```

可以将用2位或4位数字表示的年份数传递给mktime()函数。从0～69的2位数字表示的年份可以解释成2000年到2069年，而从70～99的年份解释成1970年到1999年。

如下所示的代码是说明mktime()函数使用的其他例子：

```
$time=mktime(12,0,0);
```

该语句将给出今天日期的中午时间。

```
$time=mktime(0,0,0,1,1);
```

该语句将给出当前年的1月1日。请注意，在小时参数中，我们使用了0（而不是24）来表示午夜。

也可以在简单日期算法中使用mktime()函数。如下所示：

```
$time=mktime(12,0,0,$mon,$day+30,$year);
```

虽然（$day+30）通常都会大于一个月的日期，该语句将在指定的日期基础上增加30天。

要消除冬令时和夏令时之间的问题，可以使用12点来代替0点。如果在25小时日中增加（24*60*60）s，将停留在同一天。在中午时间增加相同的秒数，将给出11am时间，但是至少是在正确的一天。

21.1.3　使用getdate()函数

能够确定当前时间的另一个很实用的函数是getdate()函数。该函数原型如下所示：

```
array getdate([int timestamp])
```

它以时间戳作为可选参数，返回一个相关数组，表示日期和时间的各个部分，如表21-2所示。

表21-2　getdate()函数返回的相关数组中的关键字-值对

关键字	值	关键字	值
seconds	秒钟，数字	year	年份，数字
minutes	分钟，数字	yday	年份中的日期，数字
hours	小时，数字	weekday	星期，全写
mday	月份中的日期，数字	month	月份，全写
wday	星期，数字	0	时间戳，数字
mon	月份，数字		

在数组中定义了以上数据后，可以很方便地将它们转换成任何所需的格式。数组的0元素（时间戳）可能没有什么用途，但是如果调用

没有参数的getdate()函数，它将返回当前的时间戳。

```
<?php
$today=getdate();
print_r($today);
?>
```

produces something similar to the following output:

```
Array(
[seconds]=>45
[minutes]=>6
[hours]=>20
[mday]=>14
[wday]=>3
[mon]=>3
[year]=>2007
[yday]=>72
[weekday]=>Wednesday
[month]=>March
[0]=>1173917205
)
```

21.1.4　使用checkdate()函数检验日期有效性

可以调用checkdate()函数来检验日期是否有效。这对检查用户输入的日期来说是非常有用的。checkdate()函数的原型如下所示：

```
int checkdate(int month,int day,int year)
```

它将检查年份数是否为介于0～32 767的一个整数，月份是否介于1～12，以及日期是否存在于特定的月份。当判断一个日期是否有效时，该函数同样会考虑闰年。

例如：

```
checkdate(2,29,2008)
```

将返回true，而：

```
checkdate(2,29,2007)
```

则返回false。

21.1.5　格式化时间戳

使用strftime()函数，你可以根据系统的locale（地域，Web服务器的本地设置）来格式化一个时间戳。这个函数具有如下所示的原型：

```
string strftime(string$format[,int$timestamp])
```

$format参数是定义了如何显示时间戳的格式化代码。$timestamp参数是传递给该函数的时间戳。这个参数是可选的。如果没有传递时间戳参数，本地系统的时间戳（脚本运行时的时间戳）将被返回。如下代码所示：

```
<?php
echo strftime('%A<br/>');
echo strftime('%x<br/>');
echo strftime('%c<br/>');
echo strftime('%Y<br/>');
?>
```

以上代码用四种不同格式显示了系统的当前时间戳。这段代码将产生类似于如下的输出：

Friday

03/16/07

03/16/07 21:17:24

2007

表21-3给出了strftime()函数格式化代码的完整列表。

需要注意的是，表21-3中列出的格式化代码中，如果它包含了标准格式，其值将被Web服务器的本地相关设置所代替。Strftime()函数对于

用不同的格式显示日期和时间是非常有用的，而且有助于改进页面的用户友好体验。

表21-3 **strftime()**函数的格式化代码

代 码	描 述
%a	星期几（英文缩写）
%A	星期几
%b或%h	月份（英文缩写）
%B	月份
%c	标准格式的日期和时间
%C	公元
%d	月份内的日期（从01～31）
%D	缩写格式的日期（mm/dd/yy）
%e	月份内的日期（两个字符组成的字符串，“1”～“31”）
%g	根据周数的年份数，两位数字
%G	根据周数的年份数，四位数字
%H	小时数（从00～23）
%I	小时数（从1～12）
%j	年内的日期（从001～366）
%m	月份（从01～12）
%M	分钟数（从00～59）
%n	换行（\n）
%p	am或pm（或与地域风俗等价）
%r	使用a.m./p.m.风格的时间表示
%R	使用24小时风格的时间表示
%S	秒数（从00～59）
%t	制表符（\t）
%T	hh:ss:mm格式的时间
%u	星期内的日期（从1-Monday到7-Sunday）
%U	周数（一年中第一个周日将作为第一周的第一天）
%V	周数（一年的第一周将至少有4天，这周才会当作一周）
%w	星期内的日期（从0-Sunday到7-Saturday）
%W	周数（一年的第一个周一作为该周的第一天）
%x	标准格式的日期（没有时间）
%X	标准格式的日期（没有日期）
%y	年份数（两位数字）
%Y	年份数（四位数字）
%z或%Z	时区

21.2　在PHP日期格式和MySQL日期格式之间进行转换

MySQL中的日期和时间是以ISO8601标准处理的。从其中获取时间是相对正常的，但是ISO8601期望输入的日期要首先输入年。例如，2008年3月29日应该输入2008-03-29或08-03-29。在默认的情况下，从MySQL获取日期的顺序也是如此。

根据目标用户不同，我们可能会发现这个函数并不是用户友好的。通常，要在PHP和MySQL之间通信，需要进行日期格式的转换。这可以在其中任一端进行。

当从PHP将日期输入到MySQL时，可以调用前面介绍的date()函数轻松地将其转换为适当的格式。需要注意的一个小问题是，在进行操作时应该使用带有前导0格式的日期和月份，这样可以避免在MySQL中造成混乱。可以使用两位数字的年份，但是使用4位年份是一个不错的想法。如果希望在MySQL端进行转换，可以使用两个有用的函数，它们分别是DATE_FORMAT()和UNIX_TIMESTAMP()。

DATE_FORMAT()函数与PHP的同名函数类似，只是使用不同的格式代码。通常，我们最希望做的事情就是以MM-DD-YYYY的形式格式化日期，而不是采用MySQL中固有的ISO格式，也就是YYYY-MM-DD格式。可以通过如下所示的查询代码来完成它：

```
SELECT DATE_FORMAT(date_column,'%m%d%Y')
FROM tablename;
```

格式代码%m表示2位数字的月份；%d表示2位数字的日期；而%Y表示4位数字的年份。表21-4列出了更多MySQL支持的实用格式代码。

表21-4 MySQL的DATE_FORMAT()函数的格式代码

代 码	描 述
%M	月份，全称
%W	星期，全称
%D	月份中的日期，数字，带文本后缀（例如，1 st）
%Y	年份，数字，4位数字
%y	年份，数字，2位数字
%a	星期，3字符
%d	月份中的日期，数字，带前导0
%e	月份中的日期，数字，不带前导0
%m	月份，数字，带前导0
%c	月份，数字，不带前导0
%b	月份，文本，3个字符表示
%j	年中的日，数字
%H	小时，24小时制，带前导0
%k	小时，24小时制，不带前导0
%h或%I	小时，12小时制，带前导0
%l	小时，12小时制，不带前导0
%i	分钟，数字，带前导0
%r	时刻，12小时制（hh:mm:ss[AM[PM]]）
%T	时刻，24小时制（hh:mm:ss）
%S或%s	秒钟，数字，带前导0
%p	AM或PM
%w	星期，数字，从0（星期日）到6（星期六）

UNIX_TIMESTAMP()函数功能与之类似，但是它可以将一列转换为一个UNIX时间戳。例如：

```
SELECT UNIX_TIMESTAMP(date_column)
FROM tablename;
```

将返回已经被格式化成UNIX时间戳的日期。这样，就可以像在PHP中一样处理它。

使用UNIX时间戳，可以很方便执行日期计算和比较操作。但是请记住，时间戳通常可以表示1902年至2038年之间的日期，而MySQL日期类型具有更大的时间范围。

作为一条重要的规则，当只是保存和显示日期的时候，应该使用UNIX时间戳来计算日期和作为标准日期格式。

21.3　在PHP中计算日期

在PHP中，计算两个日期之间长度的最简单方法就是通过计算两个UNIX时间戳之差来获得。程序清单21-1所示的脚本中就使用了这种方法。

程序清单21-1　calc_age.php——根据某人的生日计算年龄

```
<?php
//set date for calculation
$day=18;
$month=9;
$year=1972;
//remember you need bday as day month and year
$bdayunix=mktime(0,0,0,$month,$day,$year);//get ts for then
$nowunix=time();//get unix ts for today
$ageunix=$nowunix-$bdayunix;//work out the difference
$age=floor($ageunix/(365*24*60*60));//convert from seconds to years
echo"Age is$age";
?>
```

在以上脚本中，我们设置了用以计算年龄的日期。在一个实际的应用程序中，该信息很可能来自一个HTML表单。首先，我们调用了mktime()函数分别计算生日的时间戳和当前时间的时间戳：

```
$bdayunix=mktime(0,0,0,$month,$day,$year);
$nowunix=time();//get unix ts for today
```

因为这些日期具有相同的格式，因此，我们可以直接将它们相减。

```
$ageunix=$nowunix-$bdayunix;
```

现在，来处理一个有点棘手的问题——将这个时间段转化为更为友好的时间度量单位。这并不是一个时间戳，而是一个用秒钟量度的人的年龄。通过用一年的秒数来除当前以秒度量的年龄，将其转化为以年来度量。这样，我们就可以使用floor()函数对所得结果进行取整处理，以20岁为例，到他20岁那年为止：

```
$age=floor($ageunix/(365*24*60*60));//convert from seconds to years
```

但是值得注意的是，该方法是有缺陷的，它受UNIX时间戳（通常是32位整型）范围的限制。生日计算并不是时间戳的很好应用。这个例子只适用于在所有平台下计算1970年以后出生的人的生日。Windows无法管理1970年以前的时间戳。即使这样，这种计算通常也不是非常准确的，因为它不支持闰年，并且如果某人的生日刚好是冬令时和夏令时（或夏令时和冬令时）进行切换的午夜，这种计算也会出现错误。

21.4　在MySQL中计算日期

PHP没有提供更多内置的日期操作函数。很明显，我们可以编写自己的函数，但是务必考虑闰年和时令切换的时间。另一个选择是下载别人的函数。在PHP手册中，可以找到许多用户编写的函数，但是只有很少的一部分才是考虑全面的。

提示　在PHP 5.3版本中，增加了一些日期计算函数，包括date_add()、date_sub()和date_diff()函数。这些日期操作函数消除了必需使用MySQL来提供PHP以前版本所缺少的日期操作函数。

一个并不是非常好的选择是使用MySQL。MySQL提供了大量的日期操作函数，这些函数适用于UNIX时间戳以外的可供日期范围。必须连接MySQL服务器来运行一个MySQL查询，但是不使用数据库的数据。

如下所示的查询在1700年2月28日的基础上增加了一天，并且返回了结果日期：

```
select adddate('1700-02-28',interval 1 day)
```

1700年不是闰年，因此其结果是1700-03-01。

在MySQL手册中，可以找到大量描述和修改日期和时间的语法，网址为：http://www.mysql.com/doc/en/Date_and_time_functions.html。

不幸的是，要获得两个日期之间的年数并不是一件容易的事情，因此生日例子还存在一些问题。我们可以很容易获得以天为单位的某人年龄，程序清单21-2所示的代码将年龄转换为年份，这并不是非常准确的。

程序清单21-2　mysql_calc_age.php——使用MySQL来计算某人基于生日的年龄

```
<?php
//set date for calculation
$day=18;
$month=9;
$year=1972;
//format birthday as an ISO 8601 date
$bdayISO=date("c",mktime(0,0,0,$month,$day,$year));
//use mysql query to calculate an age in days
$db=mysqli_connect('localhost','user','pass');
$res=mysqli_query($db,"select datediff(now(),'$bdayISO')");
$age=mysqli_fetch_array($res);
//convert age in days to age in years(approximately)
echo"Age is".floor($age[0]/365.25);
?>
```

在将生日格式化成一个ISO时间戳后，可以将如下所示的查询提交给MySQL：

```
select datediff(now(),'1972-09-18T00:00:00+10:00')
```

MySQL的now()函数通常将返回当前的日期和时间。MySQL的datediff()函数（在PHP 4.1.1版本引入）将两个日期相减并返回日期的差。

以上脚本的执行并不需要从一个表格选择数据，甚至选择一个数据库，但是必须使用有效的用户名和密码登录到一台MySQL服务器。

由于没有特定的内置函数可以用于这种计算，因此用来计算确切年份的SQL查询就比较复杂。这里，我们采用了捷径，用日期年龄除以365.25来获得年龄。如果对某人的生日进行如此计算，根据这个人一生可能经历的闰年数不同，年份生日可能会出现一年的偏差。

21.5　使用微秒

对于某些应用程序来说，以s（秒）来计量时间不够精确。如果希望以更短的时间段来计量时间，例如运行所有PHP脚本所需的时间，必须使用microtime()函数。

在PHP 5中，调用microtime()并且将参数get_as_float设置为true。当给出了这个可选参数后，这个调用将返回浮点数的时间戳。该时间戳与mktime()函数、time()函数或date()返回值相同，但是还有小数部分。

如下所示的语句：

```
echo number_format(microtime(true),10,'.','');
```

将生成类似于1174091854.84的输出。

在早期版本中，无法请求浮点数类型的输出。它是以字符串形式提供的。没有给出参数的microtime()函数调用将返回一个字符串，类似于“0.34380900 1174091816”。第一个数字是小数点部分，而第二个数字是整个秒数，该秒数是1970年1月1日以后的所有秒数。

与处理字符串相比，处理数字更加方便。因此在PHP 5中，最简单的方法就是调用参数为true的microtime()函数。

21.6 使用日历函数

PHP提供了一组日历函数，这些函数可以实现日期在不同的日历系统之间的转换。我们使用的主要日历有Gregorian、Julian和Julian Day Count。

Gregorian日历是大多数西方国家目前所使用的历法。Gregorian中的日期1582年10月15日，1582与Julian日历中的1582年10月5日等效。而在此日期以前，Julian日历是人们更常用的历法。不同的国家将原日历转换为Gregorian日历的时期不同，有些国家甚至在20世纪早期才转换。

除了这两个日历之外，我们可能还没有听说过Julian Day Count日历。该日历与UNIX时间戳有许多相似之处。它是从大约公元前4000年起的某个日期开始计算的日子数，自身并不是特别有意义，但是它对于格式之间的转换却非常有用。要将一个日历格式转换到另一个日历格式，我们首先要转换成Julian Day Count，然后再将其转换成要输出的日历。

要在UNIX下使用这些函数，必须已经在PHP中编译了日历扩展库，通过--enable-calendar选项实现。这些日历扩展库已经内置在Windows系统的安装中。

要体验这些函数，我们首先要了解这些可能用来将日期从Gregorian日历转换到Julian日历的函数原型：

```
int gregoriantojd(int month,int day,int year)
string jdtojulian(int julianday)
```

要转换一个日期，需要调用这两个函数：

```
$jd=gregoriantojd(9,18,1582);
echo jdtojulian($jd);
```

以上代码将以MM/DD/YYYY格式显示Julian日期。

这些函数的变体可以实现日期格式在Gregorian、Julian、French以及Jewish日历和UNIX时间戳之间转换。

21.7 进一步学习

如果要了解PHP和MySQL中更多的时间与日期函数，可以参阅PHP在线指南的相关部分，网址如下所示：http://php.net/manual/en/ref.datetime.php 以及 http://dev.mysql.com/doc/refman/5.0/en/date-and-timefunctions.html。

如果要在日历之间进行转换，可以查看PHP日历函数的手册页：http://php.net/manual/en/ref.calendar.php。

21.8 下一章

使用PHP可以实现的独特而有意义的事情之一就是创建动态图像。在第22章“创建图像”中，我们将讨论如何使用图形库函数来获得一些有趣而又有用的效果。

第22章　创建图像

使用PHP能够完成的一件有意义的事情之一就是创建动态图像。PHP提供了一些内置的图像信息函数，也可以使用GD2函数库创建新图像或处理已有的图像。本章将介绍如何使用这些图像函数得到一些有趣而又有用的效果。

在本章中，我们将主要介绍以下内容：

■在PHP中设定图像支持

■理解图像格式

■创建图像

■在其他页面中使用自动生成的图像

■用文本和字体创建图像

■绘制图像与用图表描绘数据

特别地，我们将介绍两个例子：创建网站动态按钮和使用来自MySQL数据库中的数字绘制一个条形图。

在这里，我们将使用GD2函数库，但是还有一个颇受欢迎的PHP图像库。ImageMagick库并不是标准PHP的一部分，但是通过PHP扩展类库（PECL）很容易安装这个函数库。ImageMagick和GD2都有许多类似的特性，但是在某些方面，ImageMagick的功能更丰富一些。如果希望创建GIF（甚至是动画GIF），就应该使用ImageMagick。如果希望使用真彩图像或制造透明效果，应该比较这两个函数库所提供的功能。

访问如下站点可以找到ImageMagick的PHP下载扩展类库——PECL：

http://pecl.php.net/package/imagick.

关于ImageMagick主要功能的介绍以及详细文档，请参阅如下站点：

http://www.imagemagick.org.

22.1 在PHP中设置图像支持

在PHP中，有些图像函数是可以直接使用的，但是大多数函数需要安装GD2函数库。关于GD2的详细信息，请访问站点：http://www.boutell.com/gd/。

从PHP的4.3版本开始，PHP捆绑了自己版本的GD2库，这是由PHP开发团队实现的。这个版本的GD2库更容易安装，因此我们可以使用这个版本。在Windows平台下，只要注册了php_gd2.dll扩展，PNG和JPEG是自动支持的。注册php_gd2.dll非常简单，只要在PHP的安装目录（\ext子目录）找到该文件并复制到系统目录（如果使用Windows XP，就是C:\Windows\system）。此外还需要在php.ini文件中取消如下一行指令的注释（删除该行指令开始处的“;”），如下所示：

```
extension=php_gd2.dll
```

如果使用UNIX而又希望使用PNG，必须安装libpng库和zlib库，可以从如下站点分别获得它们：http://www.libpng.org/pub/png/、http://www.gzip.org/zlib/。

需要使用如下命令行选项对PHP进行配置：

```
--with-png-dir=/path/to/libpng
--with-zlib-dir=/path/to/zlib
```

如果使用UNIX并且希望使用JPEG，必须下载jpeg-6b库，然后重新编译GD库，使其包括对JPEG的支持。可以从以下站点下载该库：ftp://ftp.uu.net/graphics/jpeg/。

此外，还应该使用如下命令行选项重新配置和编译PHP：

```
--with-jpeg-dir=/path/to/jpeg-6b
```

如果希望在图像中使用TrueType字体，还需要FreeType库。这个函数库也是在从PHP 4版本开始捆绑的。当然，也可以从以下站点下载该库：http://www.freetype.org/。

如果希望使用PostScript Type 1字体，必须下载t1lib库，可以从如下站点下载该库：

ftp://sunsite.unc.edu/pub/Linux/libs/graphics/.

需要使用如下所示的命令行选项运行PHP的配置程序：

```
--with-t1lib[=path/to/t1lib]
```

最后使用--with--gd配置PHP。

22.2 理解图像格式

GD库支持JPEG、PNG和WBMP格式。但它不再支持GIF格式。下面，让我们简要地介绍一下这些格式。

22.2.1 JPEG

JPEG（发音为"jay-peg"）是联合图像专家组（Joint Photographic Experts Group，JPEG）的缩写，它是一种压缩标准的名字，我们提到JPEG文件的时候，其文件格式实际上是JFIF，它对应于JPEG组织发布的标准之一。

简单地说，JPEG通常是用来存储照片或者存储具有丰富色彩和色彩层次的图像。这种格式使用了有损压缩，也就是说，为了将图形压缩成更小的文件，图像质量有所破坏。因为JPEG压缩后应该包含基本类似的图像和颜色的层次，所以人眼可以忍受这些图像质量的损失。正是由于这个原因，该格式不适合绘制线条、文本或颜色块。

可以从官方网站了解更多关于JPEG/JFIF的信息：http://www.jpeg.org/。

22.2.2 PNG

PNG（发音为"ping"）是可移植的网络图像（Portable Network Graphics,PNG）的缩写。该文件格式可以看作是图形文件交换格式（Graphics Interchange Format,GIF）的替代品，其原因我们将在后续内容详细介绍。PNG网站将其描述为“一种强壮的图像格式，并且是无损压缩”。正因为它是无损压缩，所以该图像格式适合包含文本、直线和简单颜色块的图像，例如，Web站点标题和按钮——所有可以用GIF图像来达到的效果。通常，一个相同图像的PNG压缩版本大小与GIF压缩版本的大小相当。PNG还提供了可变的透明度、微细修正和二维空间交错。但是，它不支持动画——要实现动画，必须采用其扩展格式MNG，该格式仍在开发之中。

无损压缩模式对于图解来说是非常不错的，但是通常并不适合保存大图像，因为它们通常很大。

可以从官方网站了解更多关于PNG的信息：http://www.libpng.org/pub/png/。

22.2.3 WBMP

WBMP全称为Wireless Bitmap（无线位图）。它是专门为无线通信设备设计的文件格式，但是并没有得到广泛应用。

22.2.4 GIF

GIF是图形文件交换格式（Graphics Interchange Format）的缩写，它是无损压缩格式，广泛应用于网络，用来存储包含文本、直线和单块颜色的图像。

GIF格式使用了来自24位RGB颜色空间的256种不同颜色的调色板。它还支持动画，允许每一帧使用不同的256色调色板。颜色的限制使得GIF格式不适用于产生高画质以及需要扩展颜色的图像，但是它还是非常适合简单图像，例如具有特定颜色区域的图形或徽标。

GIF使用LZW无损数据压缩技术进行压缩，这样就减少了文件大小，而又不会降低可视质量。

22.3　创建图像

在PHP中，创建一个图像应该完成如下所示的4个基本步骤：

1）创建一个背景图像，以后的操作都将基于此背景图像。

2）在背景上绘制图形轮廓或输入文本。

3）输出最终图形。

4）清除所有资源。

下面，我们先来了解一个非常简单的创建图像脚本。该脚本如程序清单22-1所示。

程序清单22-1　simplegraph.php——输出带标签"Sales"的简单直线图形

```
<?php
//set up image
$height=200;
$width=200;
$im=imagecreatetruecolor($width,$height);
$white=imagecolorallocate($im,255,255,255);
$blue=imagecolorallocate($im,0,0,64);
//draw on image
imagefill($im,0,0,$blue);
imageline($im,0,0,$width,$height,$white);
imagestring($im,4,50,150,'Sales',$white);
//output image
Header('Content-type:image/png');
imagepng($im);
//clean up
imagedestroy($im);
?>
```

运行以上脚本，其输出结果如图22-1所示。

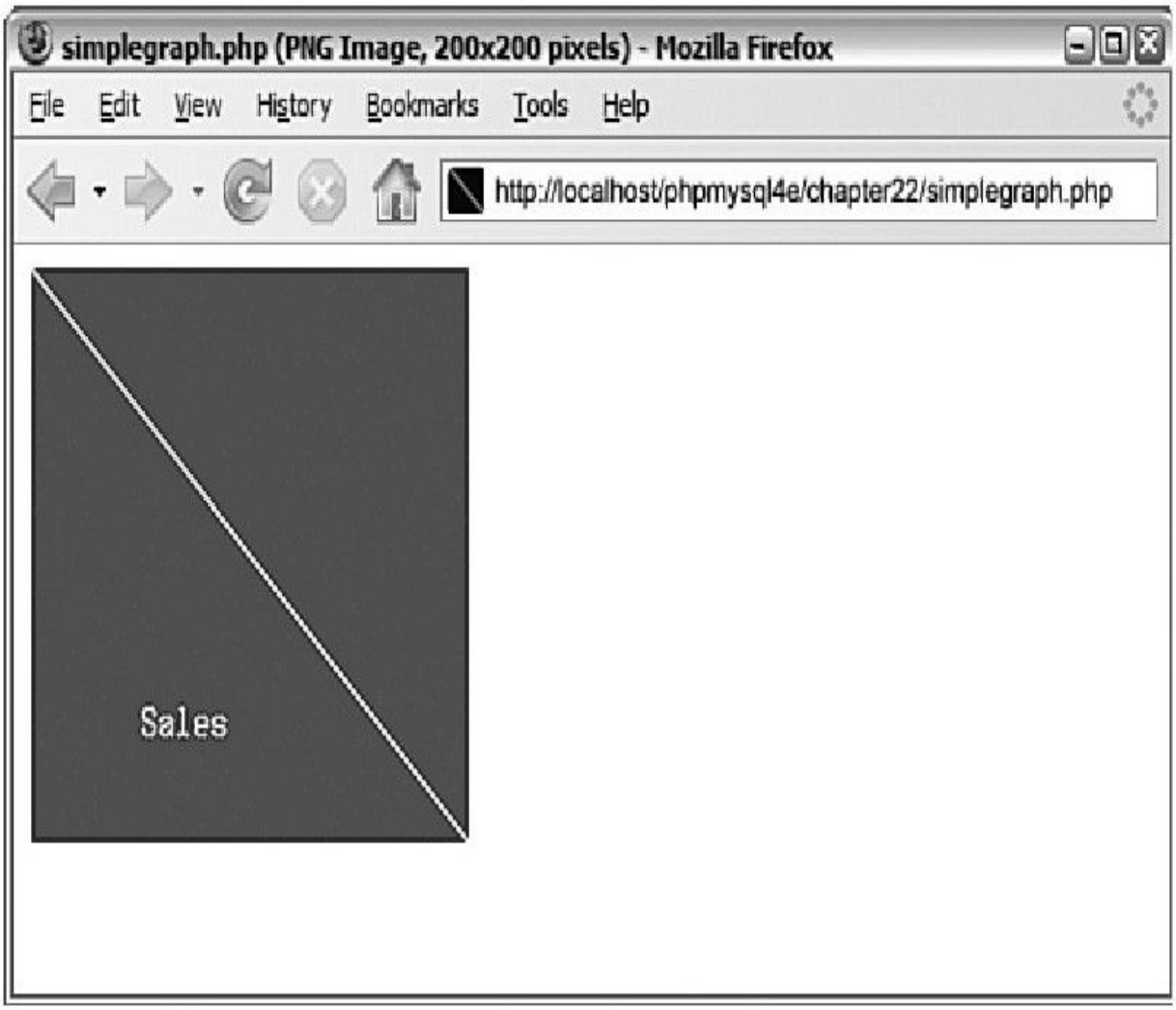


图　22-1　该脚本绘制黑色背景，并在上面添加了一条直线和一个文本标签

接下来，我们将一步一步地讲述该图像的创建过程。

22.3.1　创建一个背景图像

要在PHP中开始创建或修改一个图像，必须首先建立一个图像标识符。有两种方法可以实现图像标识符的创建。一种方法是创建一个空白的背景，这可以通过调用函数ImageCreate-TrueColor()来实现，如以下脚本所示：

```
$im=imagecreatetruecolor($width,$height);
```

我们需要为这个函数传递两个参数。第一个是新图像的宽度，另一个是新图像的高度。该函数将返回新图像的标识符。（它非常类似于文件句柄。）

另一种方法是读入一个已有的图像文件，可以对图像进行过滤，改变其大小或在上面添加其他图像。根据所读入的文件格式不同，可以通过调用下面3个函数之一来实现：Imagecreatefrompng()、Imagecreatefromjpeg()或Imagecreatefromgif()。这3个函数都是以文件名作为参数的，如下所示：

```
$im=imagecreatefrompng('baseimage.png');
```

本章后续内容所示的一个例子说明了如何使用已有的图像来创建动态按钮。

22.3.2　在图像上绘图或打印文本

在一个图像上绘图或打印文本需要两个步骤。首先，必须选择希望绘制的颜色。我们知道，在计算机显示器上显示的颜色是由不同数量的红色、绿色和蓝色组成的。图像格式使用一个调色板，它包含所有3种颜色的可能组合的特定子集。要使用一种颜色绘制一个图像，必须将此颜色添加到图像的调色板上。我们必须对每一种要使用的颜色进行相同的处理，即使它是白色或黑色。

也可以通过调用Imagecolorallocate()函数为图像选择颜色。需要为该函数传递图像标识符和希望使用的颜色，而颜色由红、绿和蓝（RGB）值的组合决定。

在程序清单22-1中，我们使用了两种颜色：黑色和白色。通过调用以下函数分配这两种颜色。

```
$white=imagecolorallocate($im,255,255,255);
$blue=imagecolorallocate($im,0,0,64);
```

以上函数将返回一个可供以后使用的颜色标识符。

其次，要真正将颜色绘制到图像中，还需要使用许多其他不同的函数，对这些函数的使用取决于要绘制的内容——直线、弧、多边形或文本。

通常，绘图函数需要下列参数：

■图像标识符

■需要绘制内容的起始坐标和结束坐标

■图像使用的颜色

■对于文本，需要字体信息

在这个例子中，我们使用了3个绘画函数。下面，我们逐一了解这些函数。

首先，调用ImageFill()函数绘制了一个用以在上面绘画的黑色背景：

```
imagefill($im,0,0,$blue);
```

该函数以图像标识符、绘画区域的起始坐标（x和y）以及颜色作为参数。

提示　需要注意的一点是，图像的起始坐标从左上角开始，该点坐标为x=0、y=0。图像右下角的坐标是x=$width、y=$height。这是计算机图形的常识，但是这与常规的作图习惯是相反的。

接下来，我们从左上角（0，0）开始画一条线，直到图像的右下角（$width,$height）：

```
imageline($im,0,0,$width,$height,$white);
```

该函数以图像标识符、直线的起始点的x和y坐标、终点以及颜色作参数。

最后，我们在该图中添加了一个标签：

```
imagestring($im,4,50,150,'Sales',$white);
```

Imagestring()函数所需的参数与Imageline()有些不同。其原型是：

```
int imagestring(resource im,int font,int x,int y,string s,int col)
```

它以图像标识符、字体、文本的起始坐标x与y以及颜色作为参数。

字体参数值是1～5的数字。它们表示一组以latin2为编码的内置字体，参数值越高，对应的字体越大。也可以选择特殊字体，例如，可以选择TrueType字体，或PostScript Type 1字体。这些字体都有相应的函数设置。在下一个例子中，我们将使用TrueType字体的函数。



图　22-2　常规的文本表现出交错的锯齿，特别是字体很大的时候，而反别名

使用可选的字体函数组的一个原因就是由函数Imagestring()或相关函数如Imagechar()（写一个字符到图像）写出的文本是可用别名的，而TrueType和PostScript函数生成反别名的文本。

如果不能确认它们到底有什么区别，请参阅图22-2。请注意，字母中的曲线或折线的地方，别名文本将表现出交错的锯齿。而曲折和拐弯的地方则是通过使用"staircase"效果而获得的。在反别名的图像中，文本中曲线或折线的地方以及背景色和文本色之间的颜色趋于平滑。

22.3.3 输出最终图形

可以将一个图像直接输出到浏览器或者一个文件。

在这个例子中，我们将图像直接输出到浏览器。这包括两个步骤。首先，需要告诉Web浏览器我们输出的是一个图像而不是文本或HTML。这可以通过调用Header()函数指定图像的MIME类型来完成：

```
Header('Content-type:image/png');
```

通常，当在浏览器中接收一个文件时，Web服务器首先发送的内容是MIME类型。对于一个HTML或PHP页面（执行后），最先发送的是：

```
Content-type:text/html
```

它将告诉浏览器如何解释后续的数据。

在这个例子中，我们要告诉浏览器将发送的是一个图像而不是常规的HTML输出。可以调用函数Header()来实现它。该函数我们尚未讨论过。

该函数将发送一个HTML标题字符串。该函数的另一个典型应用是实现HTTP重定向，告诉浏览器来加载一个不同的页面，而不是被请求的那个页面。它们通常应用于删除页面的时候。例如：

```
Header('Location:http://www.domain.com/new_home_page.html');
```

值得注意的是，在使用Header()函数时，如果该页的HTTP标题已经被发送了，那么该函数将不会执行。一旦输出任何东西到浏览器，PHP将自动发送一个HTTP标题。因此，如果使用了任何echo语句，或者甚至是在打开PHP标记之前有任何空白区域，该标题都将被发送过去，并且在试图调用Header()的时候，会从PHP接收到一个警告信息。

然而，你却可以在同一个脚本里多次调用Header()函数，从而发送多个HTTP标题，尽管它们都必须出现在向浏览器发送任何输出之前。

在我们发送标题数据之后，将通过调用如下函数输出图像数据：

```
imagepng($im);
```

该函数以PNG格式将输出内容发送到浏览器。如果希望以不同的格式发送，可以调用Imagejpeg()函数（如果系统支持JPEG的话）。仍需首先发送相应的标题，也就是：

```
Header('Content-type:image/jpeg');
```

当然，也有第二选择，那就是：将图像发送到一个文件而不是浏览器。这可以通过将可选的第二个参数加到ImagePNG()函数（或者，与之类似的支持其他格式的函数）来实现：

```
imagepng($im,$filename);
```

请记住，在这里，所有PHP的写文件规则都适用（例如，正确地设置文件权限）。

22.3.4　清理

当完成对一个图像的处理的时候，应该通过销毁图像标识符将所占用的所有资源返回给服务器。可以通过调用Imagedestroy()函数来完成：

```
imagedestroy():
imagedestroy($im);
```

22.4　在其他页面中使用自动生成的图像

因为标题只可以发送一次，而这是告诉浏览器正在发送图像数据的唯一方法，所以在普通页面里嵌入动态图像会遇到一些麻烦。可以通过以下3种方法实现：

1.正如我们在前面的例子中，可以拥有一个由图像输出组成的页面。

2.正如前面所提到的，可以将图像写到一个文件中，然后用＜IMG＞标记指向它。

3.可以将图像创建脚本置于一个图像标记中。

我们已经介绍过前两种方法。下面简要介绍一下第三种方法。要采用这种方法，需要通过图像标记包含一个HTML内嵌图像，如下所示：

```
＜img src="simplegraph.php'height="200"width="200"alt="Sales going down"/＞
```

除了直接将PNG、JPEG或GIF图像加入到IMG标记，还可以在SRC属性中使用能够生成图像的PHP脚本。该图像可以被检索，其输出是内嵌式的，如图22-3所示。

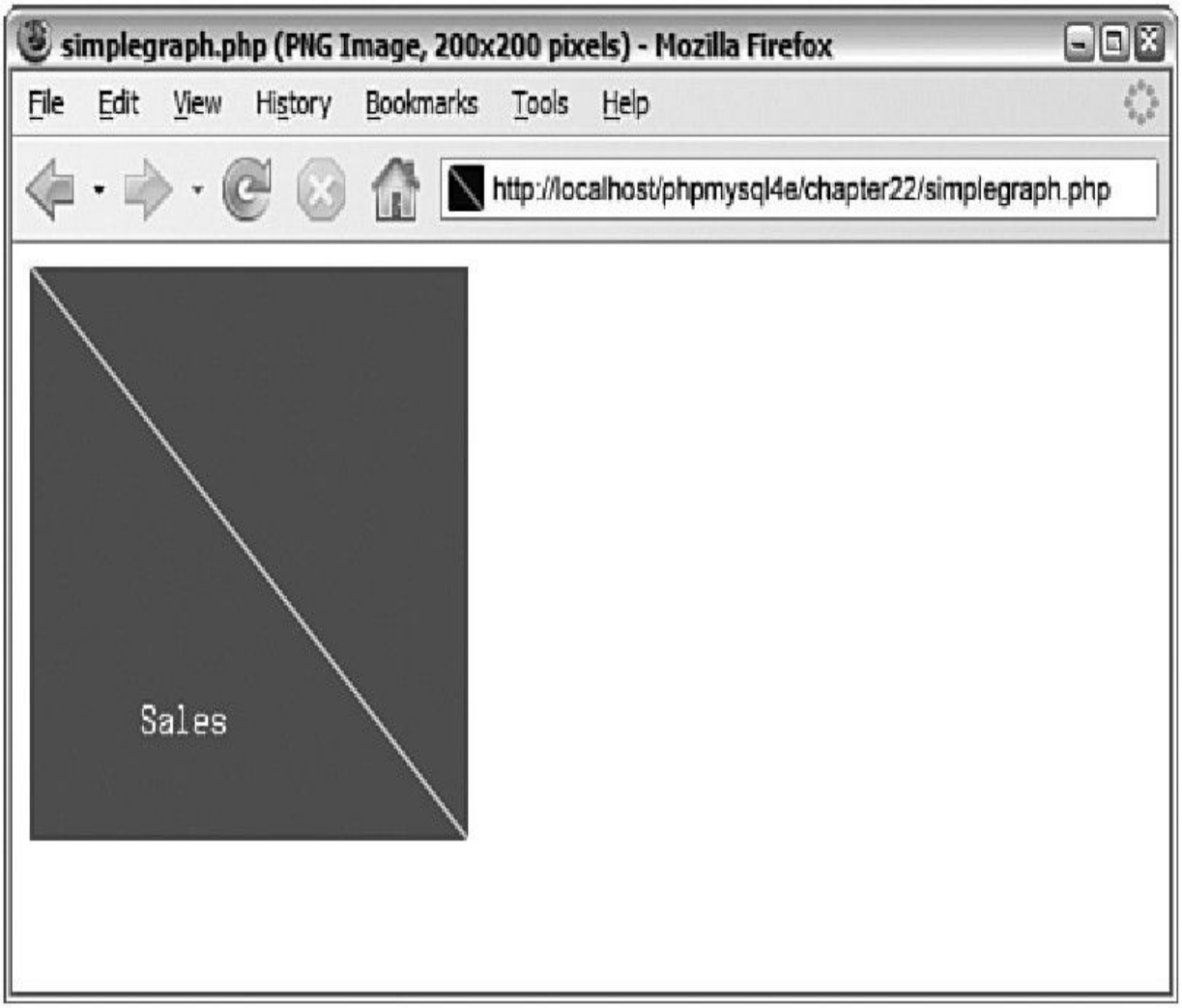


图　22-3　动态产生的内嵌图像对终端用户来说看起来像普通的图像

22.5 使用文本和字体创建图像

我们来看一个关于创建图像更复杂的例子。如果能够在网站里自动创建一个按钮或其他图像，那将会是很有趣的。可以使用我们已经介绍的技术在基于一个四边形的背景颜色上创建一个简单的按钮。通过编程，还可以实现更加复杂的效果，但是通常使用一个画板程序更容易完成这些操作。这也使得我们让一个艺术家来绘制图形而程序员来完成编程工作。

在这个例子中，我们将使用一个空白按钮模板来创建多个按钮，这样，通过该按钮模板创建的按钮具有相同的特征（如羽化边缘等），这些特征在使用Photoshop、GIMP或其他图形工具创建时容易得多。使用PHP中的图像库，我们可以从一个基本图像开始，在此图像上绘制。

我们也使用TrueType字体以便可以使用反别名的文本。TrueType字体函数有它们自己的特性，关于这些内容，我们将在以后讨论。

基本的处理过程是接收一些文本并产生包含此文本的按钮，该文本位于按钮的中央（水平方向和垂直方向都处于中央），并被赋以适合按钮的最大字体。

我们已经创建了用于测试和试验的按钮生成器的前台，其界面如图22-4所示。（因为这非常简单，我们没有为表单包含进HTML，但是可以在本书附带的文件找到，在design_button.html文件中。）

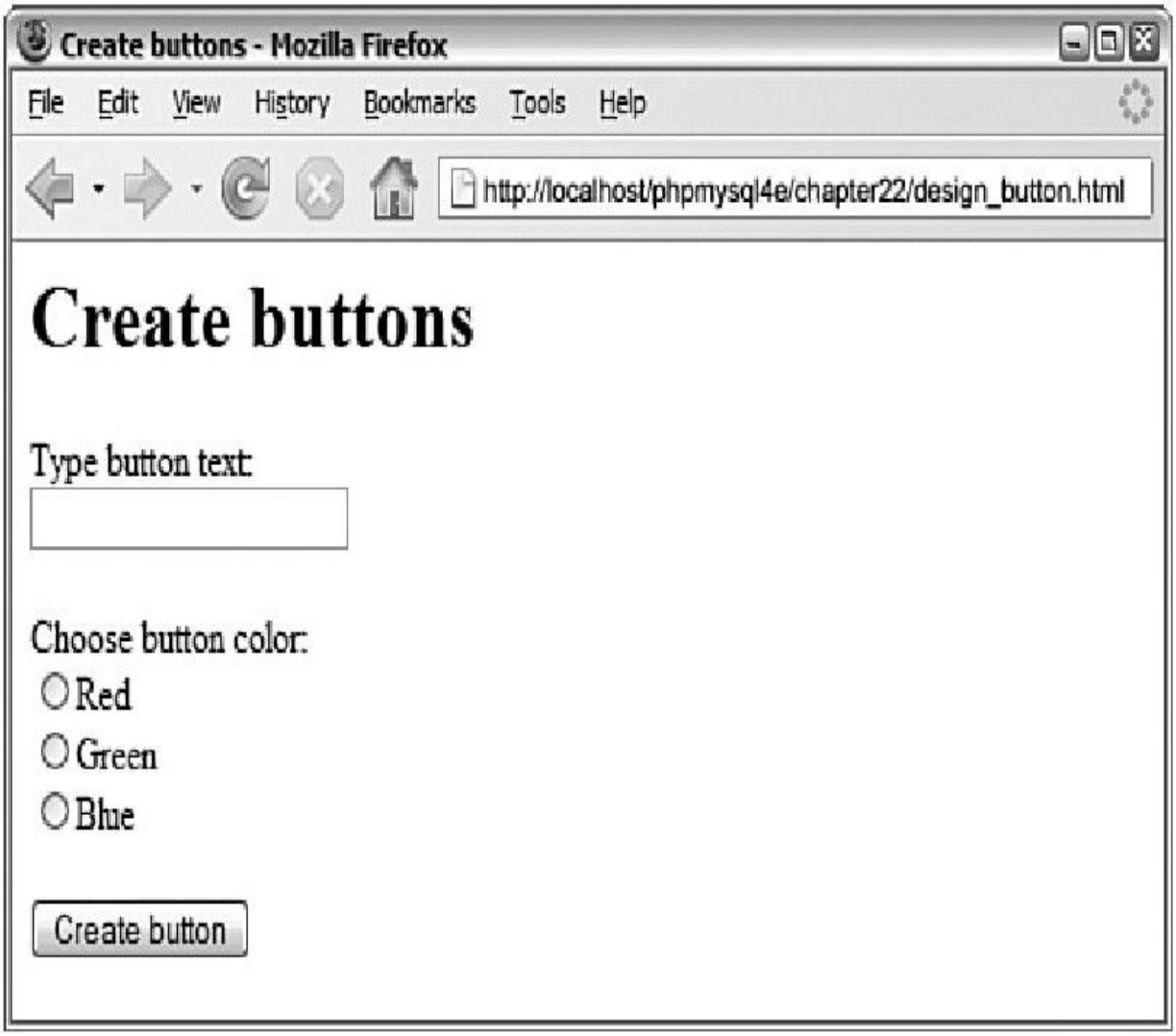


图　22-4　前台允许用户选择的请求文本中按钮的颜色和类型

可以使用这种界面让程序自动创建网站。也可以通过联机的方式调用该脚本来创建网站所有的动态按钮，但是这可能需要缓冲机制，防止其占用太多的处理器时间。

该脚本的输出结果如图22-5所示。



图　22-5　make_button.php脚本生成的按钮

该按钮由make_button.php脚本生成。脚本如程序清单22-2所示。

程序清单22-2　make_button.php——该脚本可以从design_button.html的表单或一个HTML图像标签中调用

```
<?php
//check we have the appropriate variable data
//variables are button-text and color
$button_text=$_REQUEST['button_text'];
$color=$_REQUEST['color'];
if(empty($button_text)||empty($color))
{
echo'Could not create image-form not filled out correctly';
exit;
}
//create an image of the right background and check size
$im=imagecreatefrompng($color.'-button.png');
$width_image=imagesx($im);
$height_image=imagesy($im);
//Our images need an 18 pixel margin in from the edge of the image
```

```
$width_image_wo_margins=$width_image-(2*18);
$height_image_wo_margins=$height_image-(2*18);
//Work out if the font size will fit and make it smaller until it does
//Start out with the biggest size that will reasonably fit on our buttons
$font_size=33;
//you need to tell GD2 where your fonts reside
putenv('GDFONTPATH=C:\WINDOWS\Fonts');
$fontname='arial';
do
{
$font_size--;
//find out the size of the text at that font size
$bbox=imagettfbbox($font_size,0,$fontname,$button_text);
$right_text=$bbox[2];//right co-ordinate
$left_text=$bbox[0];//left co-ordinate
$width_text=$right_text-$left_text;//how wide is it?
$height_text=abs($bbox[7]-$bbox[1]);//how tall is it?
}
while($font_size>8&&
($height_text>$height_image_wo_margins||
$width_text>$width_image_wo_margins)
);
if($height_text>$height_image_wo_margins||
$width_text>$width_image_wo_margins)
{
//no readable font size will fit on button
echo'Text given will not fit on button.<br/>';
}
else
{
//We have found a font size that will fit
//Now work out where to put it
$text_x=$width_image/2.0-$width_text/2.0;
$text_y=$height_image/2.0-$height_text/2.0;
if($left_text<0)
$text_x+=abs($left_text);//add factor for left overhang
```

```
$above_line_text=abs($bbox[7]);//how far above the baseline?
$text_y+=$above_line_text;//add baseline factor
$text_y-=2;//adjustment factor for shape of our template
$white=imagecolorallocate($im,255,255,255);
imagettftext($im,$font_size,0,$text_x,$text_y,$white,$fontname,
$button_text);
Header('Content-type:image/png');
imagepng($im);
}
imagedestroy($im);
?>
```

这是我们所看到的最长的脚本了。我们将逐步地讨论脚本的各个部分。开始是一些基本的错误检查，然后是创建画布并在此画布上操作。

22.5.1 创建基本画布

在程序清单22-2中，我们并不是从零开始作图，而是以一个已有的按钮图像为开始的。基本按钮的颜色存在3种选择：红（red-button.png）、绿（green-button.png）和蓝（blue-button.png）。

用户选择的颜色将保存在来自表单的变量color中。首先我们将从超级全局变量$_REQUEST中获得颜色，并且创建一个基于适当按钮的新图像标识符：

```
$color=$_REQUEST['color1];
...
$im=imagecreatefrompng($color.'-button.png');
```

函数Imagecreatefrompng()以一个PNG文件名作为参数，并且返回一个包含该PNG图像拷贝的图像标识符。请注意，这里并没有对基本的PNG进行任何修改。如果已经安装了适当的支持程序，就可以使用Imagecreatefromjpeg()和ImageCreateFromGIF()函数。

提示　对Imagecreatefrompng()函数的调用只是在内存中创建图像。要将该图像保存到一个文件或输出到浏览器，必须调用ImagePNG()函数。后面我们将讨论它，此前我们需要先对图像进行一些其他处理。

22.5.2　将文本调整到适合按钮

我们已经接收到了用户键入的一些文本，这些文本保存在$button_text变量中，我们要做的是将它以适合按钮的最大字体打印在按钮上。通过多次调整可以达到此目标；严格地说，是经过反复的试验。

首先设置一些相关变量。首要的两个变量是按钮图像的高度和宽度：

```
$width_image=imagesx($im);
$height_image=imagesy($im);
```

其次，设置两个表示按钮边缘的宽度的变量。按钮图像是有斜边的，因此应在文本周围为斜边留有空白。如果使用的图像与此不同，这个数字将是不同的！在这个例子中，每边的空白大概是18个像素：

```
$width_image_wo_margins=$width_image-(2*18);
$height_image_wo_margins=$height_image-(2*18);
```

我们还设置了初始化字体大小。起始字体大小是32（实际上是33，但是我们会马上减小它），因为它大约是可以适合该按钮的最大字体：

```
$font_size=33;
```

使用GD2库，必须通过设置环境变量GDFONTPATH告诉脚本字体所在的位置，如下所示：

```
putenv('GDFONTPATH=C:\WINDOWS\Fonts');
```

我们还需要设置希望使用的字体名称。我们将在TrueType函数中使用这个字体，这将在以上字体路径中查找字体文件，而且将在文件名称后添加.ttf扩展名（TrueType字体）。

```
$fontname='arial';
```

请注意，根据操作系统的不同，可能要在字体名称后添加".ttf"。

如果系统没有安装Arial（我们在这个例子中使用的字体）字体，可以将其改变为其他的TrueType字体。

现在我们开始执行循环，每次字体大小递减1，如此重复，直到提交的文本基本适合按钮：

```
do
{
$font_size--;
//find out the size of the text at that font size
$bbox=imagettfbbox($font_size,0,$fontname,$button_text);
$right_text=$bbox[2];//right co-ordinate
$left_text=$bbox[0];//left co-ordinate
$width_text=$right_text-$left_text;//how wide is it?
$height_text=abs($bbox[7]-$bbox[1]);//how tall is it?
}
while($font_size>8&&
($height_text>$height_image_wo_margins||
$width_text>$width_image_wo_margins)
);
```

以上代码通过查看所谓的文本边框测试文本的大小。通过调用ImagegetttfBox()函数完成测试，该函数是TrueType字体函数之一。在计算出字体大小之后，我们将用TrueType字体（在这个例子中，我们使用的是Arial字体，但也可以使用任何你所喜欢的字体），以及调用Imagettftext()函数将文本打印在按钮上。

一段文本的边框是围绕该文本的最小可能边框。边框的一个示例如图22-6所示。



图　22-6　边框的坐标是相对于基线给出的。坐标原点是图中的（0,0）

要获取该框的大小，可以调用如下函数：

```
$bbox=imagettfbbox($font_size,0,$fontname,$button_text);
```

以上调用的意思是，“对给定的字体大小$font_size，其文本倾斜0°，它使用TrueType字体Arial，请告诉我$button_text变量中文本的大小。”

请注意，实际上还需要将包含该字体文件的路径传递给该函数。在这个例子中，其目录与脚本所在目录相同（在默认情况下），因此不用指定其他更长的路径。

该函数将返回一个包含边框各个角的坐标的数组。数组内容如表22-1所示。

表22-1 边框数组的内容

数组索引	内　容	数组索引	内　容
0	x坐标，左下角	4	x坐标，右上角
1	y坐标，左下角	5	y坐标，右上角
2	x坐标，右下角	6	x坐标，左上角
3	y坐标，右下角	7	y坐标，左上角

要记住数组的内容，只要记住坐标开始于边框的左下角，并且坐标系是逆时针方向的。

应当注意的是，从ImageTTFBBox()函数返回的值。这些值都是坐标值，也就是与原点的距离的相对值。然而，它们不像图像坐标，图像坐标相对于图像左上角而定，而它们是根据基线而定的。

让我们回头再看看图22-6。可以看到，我们沿大部文本的底部画了一条线。该线称为基线。一些字母在行方向上低于基线，例如，在这个例子中的字母"y"。我们称之为下行字母。

基线的左边定义为起始量度——也就是，x坐标为0，y坐标也为0。坐标位于x之上称x坐标为正，位于x之下称x坐标为负。

除此之外，事实上，文本的坐标值还有位于边框之外的可能情况。例如，文本可能实际上从x坐标值-1的地方开始。

这些事实都使我们用这些数字进行计算的时候要加倍小心。

我们将按如下步骤计算文本的宽度和高度：

```
$right_text=$bbox[2];//right co-ordinate
$left_text=$bbox[0];//left co-ordinate
$width_text=$right_text-$left_text;//how wide is it?
$height_text=abs($bbox[7]-$bbox[1]);//how tall is it?
```

完成这些操作后，我们将测试循环条件。

```
}while($font_size>8&&
($height_text>$height_image_wo_margins||
$width_text>$width_image_wo_margins)
);
```

在这里，我们测试了两组条件。首先是字体是否仍然可读——使用小于8号的字体已经没有意义，因为人的肉眼几乎不能看清按钮上的文本。第二组条件是测试文本是否适合我们为其绘制的空间。

下一步，将检查循环计算是否找到了可以接受的字体大小。如果没有找到，将报告错误：

```
if($height_text>$height_image_wo_margins||
$width_text>$width_image_wo_margins)
{
//no readable font size will fit on button
echo'Text given will not fit on button.<br/>';
}
```

22.5.3 放置文本

如果前面所有的工作都做好了，下面就可以开始计算文本的基本位置。计算的是有效区域的中点。

```
$text_x=$width_image/2.0-$width_text/2.0;
$text_y=$height_image/2.0-$height_text/2.0;
```

因为使用相对基线的坐标系统比较复杂，我们需要添加一些矫正因子：

```
if($left_text<0)
$text_x+=abs($left_text);//add factor for left overhang
$above_line_text=abs($bbox[7]);//how far above the baseline?
$text_y+=$above_line_text;//add baseline factor
$text_y-=2;//adjustment factor for shape of our template
```

因为我们的图形太复杂，矫正因子允许采用基线和微调。

22.5.4　将文本写到按钮上

此后的事情就非常简单了。将文本颜色设置为白色：

```
$white=ImageColorAllocate($im,255,255,255);
```

然后，可以调用ImageTTFText()函数将文本写到按钮上：

```
imagettftext($im,$font_size,0,$text_x,$text_y,$white,$fontname,
$button_text);
```

该函数需要许多输入参数。依次是：图像标识符、用像素表示的字体大小、文本倾斜角度、文本的起始x和y坐标、文本颜色字体，以及写到按钮的实际文本。

提示　字体文件需要保存在服务器上可供读取的位置，它不能从客户端机器上获得，因为客户端的机器将它看作一个图像。

22.5.5 完成

最后，可以将该按钮输出到浏览器：

```
Header('Content-type:image/png');
imagepng($im);
```

然后，要清除占用的资源并且结束脚本：

```
imagedestroy($im);
```

这就完成了！如果一切顺利，应当看到已经有一个按钮显示在浏览器中了，它看起来如图22-5所示。

22.6　绘制图像与用图表描绘数据

在上一个应用程序中，我们已经使用了已有图像和文本。我们还没有了解绘图的例子，现在就介绍它。

在这个例子中，我们将在本网站中进行一次民意测验，让用户为虚构的选举进行投票。投票结果保存到一个MySQL数据库中，并用图像函数绘制出条形图，以表示投票结果。

这些函数的另一个主要应用是绘制图形。还可以用图形来表示任何需要表示的数据——交易量、网站点击数和任何我们所喜欢的。

对于这个例子，我们花了几分钟建立了一个名为poll的MySQL数据库。该数据库包含一个名为poll_results的表，该表的candidate列保存了候选人的名字，num_votes列保存了候选人收到的投票数。我们还为该数据库创建了一个名为poll的用户，其密码为poll。这个表的创建过程简单明了，可以通过运行如程序清单22-3所示的SQL脚本完成。可以使用如下命令，以root用户的身份进行登录：

```
mysql-u root-p＜pollsetup.sql
```

当然，也可以以具有适当权限的用户身份进行登录。

程序清单22-3　pollsetup.sql——创建Poll数据库

```
create database poll;
use poll;
create table poll_results(
candidate varchar(30),
num_votes int
);
insert into poll_results values
('John Smith',0),
('Mary Jones',0),
```

```
('Fred Bloggs',0)
;
grant all privileges
on poll.*
to poll@localhost
identified by'poll';
```

该数据库包含了三个候选人。我们通过vote.html页面提供一个投票界面。该页面代码如程序清单22-4所示。

程序清单22-4　vote.html——用户可以在此投票

```
<html>
<head>
<title>Polling</title>
<head>
<body>
<h1>Pop Poll</h1>
<p>Who will you vote for in the election?</p>
<form method="post"action="show_poll.php">
<input type="radio"name="vote"value="John Smith">John Smith<br/>
<input type="radio"name="vote"value="Mary Jones">Mary Jones<br/>
<input type="radio"name="vote"value="Fred Bloggs">Fred Bloggs<br/><br/>
<input type="submit"value="Show results">
</form>
</body>
</html>
```

该页面的输出结果如图22-7所示。

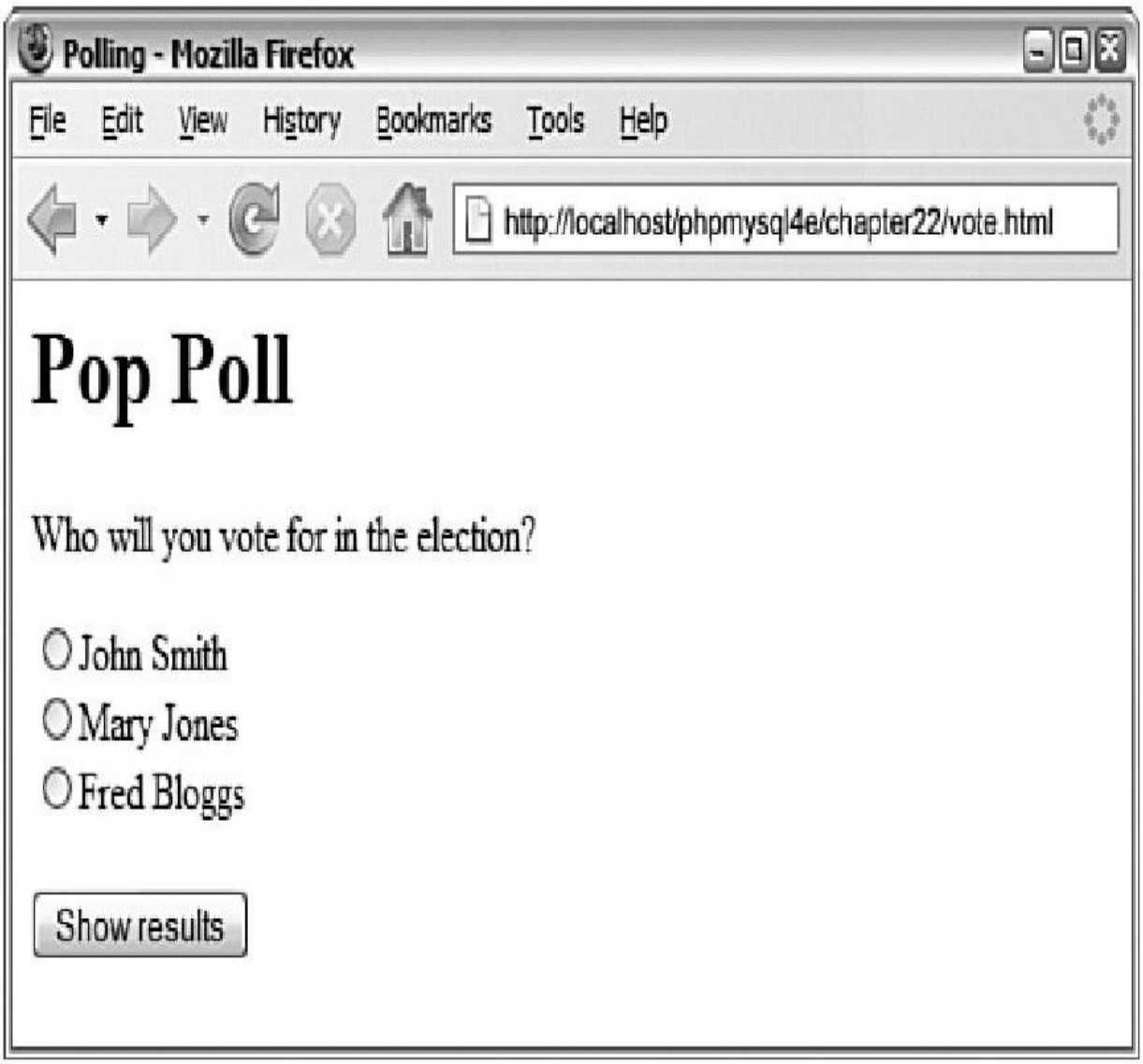


图　22-7　用户可以在此投票，点击提交按钮将显示当前投票结果

通常的想法是，当用户点击提交按钮时，将他们的投票添加到数据库中，然后再将数据库中所有的投票取出，绘制当前结果的条形图。

在用户投票之后的条形图输出如图22-8所示。

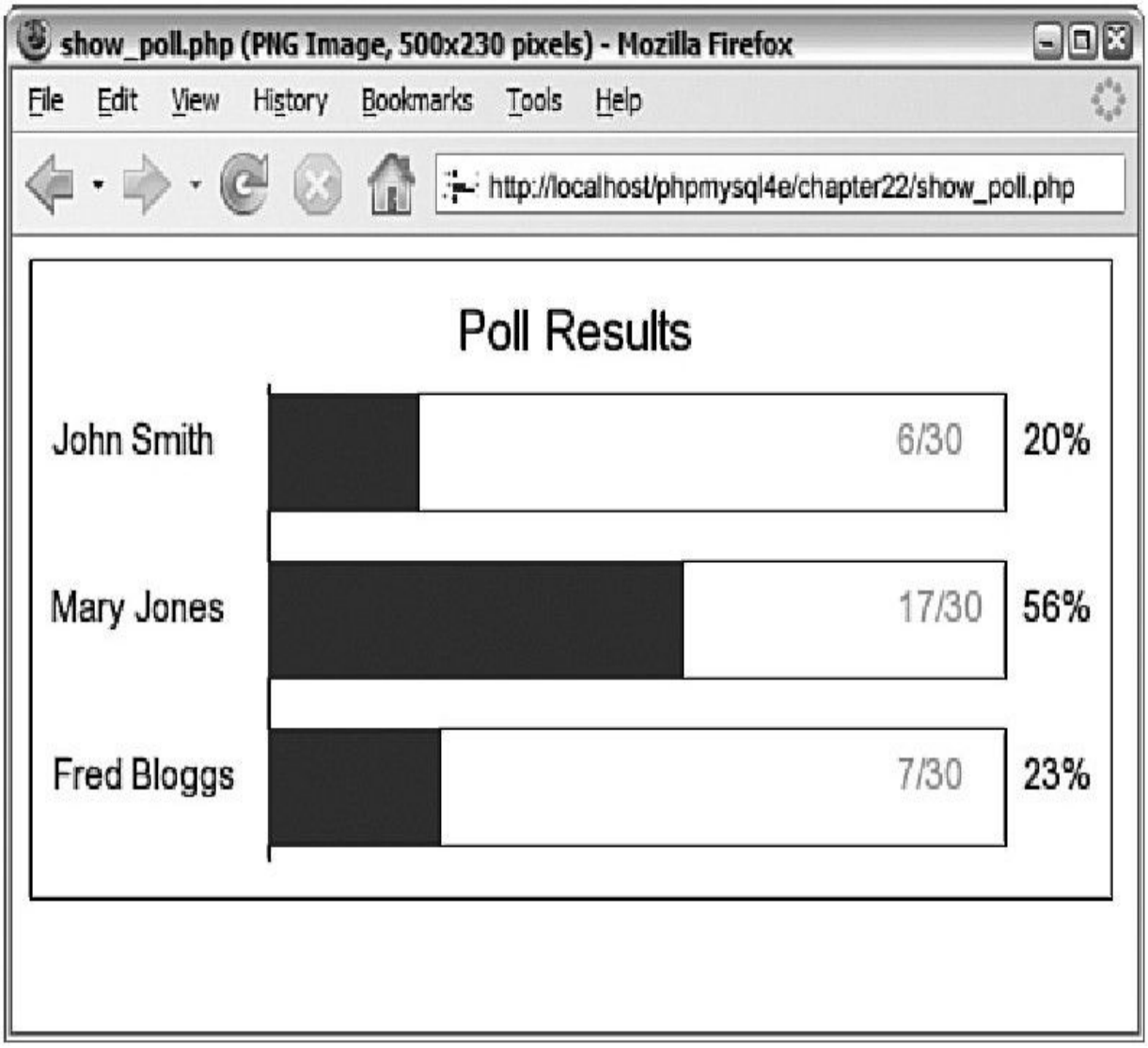


图　22-8　投票结果通过将一系列线条、矩形和文本绘制到背景图来实现

创建该图像的脚本非常长。我们将它分为4部分，并分别加以讨论。这些脚本大多数都是我们非常熟悉的；我们已经看过许多类似的MySQL示例。我们已经了解了如何用单一的颜色绘制一个背景画布，以及如何在背景图上打印文本标签。

与前面所介绍的脚本相比较，该脚本新的部分是绘制线条和矩形。我们将重点讨论这些部分。第1部分（脚本1个部分的第1部分）如程序清单22-5-1所示。

第1部分，如程序清单22-5-1所示，将连接到MySQL数据库，根据用户的投票更新投票数，获得新票数。在获得新票数信息后，就可开始通过计算绘制图表。第2部分代码如程序清单22-5-2所示。

程序清单22-5-1　show_poll.php——脚本的第1部分将更新投票数据库并检索新的投票结果

```
<?php
/*****************************************
Database query to get poll info
*****************************************/
//get vote from form
$vote=$_REQUEST['vote'];
//log in to database
if(!$db_conn=new mysqli('localhost','poll','poll','poll'))
{
echo'Could not connect to db<br/>';
exit;
}
if(!empty($vote))//if they filled the form out,add their vote
{
$vote=addslashes($vote);
$query="update poll_results
set num_votes=num_votes+1
where candidate='$vote'";
if(!($result=@$db_conn->query($query)))
{
echo'Could not connect to db<br/>';
exit;
}
}
//get current results of poll,regardless of whether they voted
$query='select*from poll_results';
if(!($result=@$db_conn->query($query)))
{
echo'Could not connect to db<br/>';
```

```
exit;
}
$num_candidates=$result->num_rows;
//calculate total number of votes so far
$total_votes=0;
while($row=$result->fetch_object())
{
$total_votes+=$row->num_votes;
}
$result->data_seek(0);//reset result pointer
```

程序清单22-5-2　show_poll.php——脚本的第2部分设置绘画所需要的所有变量

```
/*****************************************
Initial calculations for graph
*****************************************/
//set up constants
putenv('GDFONTPATH=C:\WINDOWS\Fonts');
$width=500;//width of image in pixels-this will fit in 640x480
$left_margin=50;//space to leave on left of graph
$right_margin=50;//ditto right
$bar_height=40;
$bar_spacing=$bar_height/2;
$font='arial';
$title_size=16;//point
$main_size=12;//point
$small_size=12;//point
$text_indent=10;//position for text labels from edge of image
//set up initial point to draw from
$x=$left_margin+60;//place to draw baseline of the graph
$y=50;//ditto
$bar_unit=($width-($x+$right_margin))/100;//one"point"on the graph
//calculate height of graph-bars plus gaps plus some margin
$height=$num_candidates*($bar_height+$bar_spacing)+50;
```

脚本的第2部分创建了一些用来实际绘制图形的变量。

计算出这些类型的变量值是冗长乏味的，但是想想完成之后的图像是什么模样，会使这个过程变得轻松一点。在这里，我们使用的值是通过在纸上拟定设计效果，然后再估计所要求的比例而得到的。

宽度变量$width是背景总宽度。我们也设置了左边缘和右边缘的宽度（分别使用$left_margin和$right_margin变量），每条长方形的厚度和它们之间的间距（$bar_height和$bar_spacing），以及字体、字体大小和标签的位置（$font、$title_size、$main_size、$small_size和$text_indent）。

在给定这些基本的值后，就可以进行一些基本的计算了。我们希望绘制一条基线，所有的横条都从该线伸展开。通过在左边缘加上文本标签的宽度可以得到基线的x坐标位置，然后根据边框线估计基线的y坐标位置。如果灵活性非常重要，还可以根据最长的名称获得确切的宽度。

我们还需要计算两个重要的值：第一，在图像上表示一个单元的宽度：

```
$bar_unit=($width-($x+$right_margin))/100;//one"point"on the graph
```

这是长方形条的最大宽度——从基线到右边空白处——分为100份，因为我们的图形要显示百分值。

第二个值是背景总高度：

```
$height=$num_candidates*($bar_height+$bar_spacing)+50;
```

这基本上是每一条长方形的高度乘以其数据，再加上标题所需的高度。脚本的第3部分如程序清单22-5-3所示。

程序清单22-5-3　show_poll.php——脚本的第3部分建立图形并准备添加数据

```
/*****************************************
Set up base image
*****************************************/
//create a blank canvas
$im=imagecreatetruecolor($width,$height);
//Allocate colors
$white=imagecolorallocate($im,255,255,255);
$blue=imagecolorallocate($im,0,64,128);
$black=imagecolorallocate($im,0,0,0);
$pink=imagecolorallocate($im,255,78,243);
$text_color=$black;
$percent_color=$black;
$bg_color=$white;
$line_color=$black;
$bar_color=$blue;
$number_color=$pink;
//Create"canvas"to draw on
imagefilledrectangle($im,0,0,$width,$height,$bg_color);
//Draw outline around canvas
imagerectangle($im,0,0,$width-1,$height-1,$line_color);
//Add title
$title='Poll Results';
$title_dimensions=imagettfbbox($title_size,0,$font,$title);
$title_length=$title_dimensions[2]-$title_dimensions[0];
$title_height=abs($title_dimensions[7]-$title_dimensions[1]);
$title_above_line=abs($title_dimensions[7]);
$title_x=($width-$title_length)/2;//center it in x
$title_y=($y-$title_height)/2+$title_above_line;//center in y gap
imagettftext($im,$title_size,0,$title_x,$title_y,
$text_color,$font,$title);
//Draw a base line from a little above first bar location
//to a little below last
```

```
imageline($im,$x,$y-5,$x,$height-15,$line_color);
```

在第3部分，我们将建立基本图像，分配颜色，然后开始绘制图形。

首先，我们将填充图形的背景颜色，使用如下所示函数：

```
imagefilledrectangle($im,0,0,$width,$height,$bg_color);
```

顾名思义，ImageFilledRectangle()函数将绘制一个中间填充了颜色的矩形。该函数的第一个参数与其他图形函数一样都是图像标识符。然后我们将此矩形的起点和结束点的x与y坐标传递给该函数。其起始点和结束点分别对应矩形的左上角和右下角。在这个例子中，我们还在整个背景中填充了背景颜色，该颜色是最后一个参数：白色。

然后调用：

```
imagerectangle($im,0,0,$width-1,$height-1,$line_color);
```

如上语句将绘制出围绕画布边沿的黑色轮廓线。该函数绘制了一个轮廓矩形而不是实心矩形。而其参数与前面的函数是一样的。请注意，我们绘制的矩形坐标是一直绘制到$width-1和$height-1——从（0，0）一直到画布的宽度和高度。如果我们绘制的是$width和$height，矩形将超出画布区域。

在这里，我们使用了前一个示例所介绍的逻辑和函数，将标题写到图像中，并将其置于图像中央。

最后，我们为每个长方形图绘制了基线：

```
imageline($im,$x,$y-5,$x,$height-15,$line_color);
```

ImageLine()函数可以在我们所指定的图像（$im）上画一条线，我们指定了该直线的起点坐标（$x，$y-5）和终点坐标（$x,$height-

15），并通过变量$line_color指定颜色。

在这个例子中，我们在第一条长方形图的上面一点绘制了基线，其终点是背景的底端一点。

现在，我们准备将数据填充到该图形上。脚本的第4部分如程序清单22-5-4所示。

程序清单22-5-4　show_poll.php——脚本的第4部分将实际的数据绘制到该图像上，并结束整个程序

```
/****************************************
Draw data into graph
****************************************/
//Get each line of db data and draw corresponding bars
while($row=$result->fetch_object())
{
if($total_votes>0)
$percent=intval(($row->num_votes/$total_votes)*100);
else
$percent=0;
//display percent for this value
$percent_dimensions=imagettfbbox($main_size,0,$font,$percent.'%');
$percent_length=$percent_dimensions[2]-$percent_dimensions[0];
imagettftext($im,$main_size,0,$width-$percent_length-$text_indent,
$y+($bar_height/2),$percent_color,$font,$percent.'%');
//length of bar for this value
$bar_length=$x+($percent*$bar_unit);
//draw bar for this value
imagefilledrectangle($im,$x,$y-2,$bar_length,$y+$bar_height,$bar_color);
//draw title for this value
imagettftext($im,$main_size,0,$text_indent,$y+($bar_height/2),
$text_color,$font,"$row->candidate");
//draw outline showing 100%
imagerectangle($im,$bar_length+1,$y-2,
($x+(100*$bar_unit)),$y+$bar_height,$line_color);
```

```
//display numbers
imagettftext($im,$small_size,0,$x+(100*$bar_unit)-50,$y+($bar_height/2),
$number_color,$font,$row->num_votes.'/'.$total_votes);
//move down to next bar
$y=$y+($bar_height+$bar_spacing);
}
/*****************************************
Display image
*****************************************/
Header('Content-type:image/png');
imagepng($im);
/*****************************************
Clean up
*****************************************/
imagedestroy($im);
?>
```

第4部分脚本将在数据库中逐个检索候选人，计算各自投票百分比，并为每位候选人绘制条形图并写上标签。

这里，我们又调用了ImageTTFText()函数来为图像添加标签。我们调用ImageFilled-Rectangle()函数绘制一个实心条形图：

```
imagefilledrectangle($im,$x,$y-2,$bar_length,$y+$bar_height,$bar_color);
```

并使用ImageRectangle()函数绘制了标志100%的轮廓：

```
imagerectangle($im,$bar_length+1,$y-2,
($x+(100*$bar_unit)),$y+$bar_height,$line_color);
```

在完成所有这些条形图之后，调用ImagePNG()函数输出了图像，并调用ImageDestroy()函数清理资源。

这是一段比较长的代码，但是它合乎需要，或者适合通过一个界面自动产生投票结果。该脚本一个重要的缺陷是它缺乏反欺骗机制。用户很快就可以发现他们可以重复投票而使投票结果变得毫无意义。

我们可以使用类似的方法绘制直线图形，甚至如果擅长数学的话，还可以用这样的方法来绘制杂乱的流程图。

22.7　使用其他图像函数

除了本章中用到的图像函数之外，PHP还提供了许多其他的图像函数。通过编程来绘制图形将要花费很长时间，经过多次尝试和遇到很多错误。绘图的时候我们通常应该先绘制出其轮廓，然后查阅联机手册以寻找所需的函数。

22.8 进一步学习

许多可供阅读的资料可以通过在线方式获得。如果在使用这些图像函数时遇到了问题，查看GD库的源代码文档可能很有帮助，因为PHP函数只是封装了该库中的函数。关于GD库的文档，可以从以下站点获取：http://www.libgd.org/Documentation。

但是，请记住，GD2的PHP版本只是主函数库的一个派生库，因此许多实现细节是不同的。

还有许多关于特定类型的图形应用的优秀教程。尤其是在Zend和Devshed网站，其URL如下所示：http://www.zend.com and http://devshed.com,respectively。

创建本章示例条形图应用程序的灵感也是来自Steve Maranda所写的动态图像脚本。该脚本可以在Devshed站点找到。

22.9 下一章

在下一章中，我们将讨论在PHP中使用方便的会话控制功能。

第23章　在PHP中使用会话控制

在本章中，我们将讨论PHP的会话控制功能。

在本章中，我们将主要介绍以下内容：

■什么是会话控制

■cookie

■创建一个会话的步骤

■会话变量

■会话和身份验证

23.1　什么是会话控制

我们可能曾经听说过“HTTP是无状态的协议”。这是说，HTTP协议没有一个内建机制来维护两个事务之间的状态。当一个用户在请求一个页面后再请求另外一个页面时，HTTP将无法告诉我们这两个请求是来自同一个用户。

会话控制的思想是指能够在网站中根据一个会话跟踪用户。如果我们可以做到这点，就可以很容易地做到对用户登录的支持，并根据其授权级别和个人喜好显示相应的内容。我们可以根据会话控制记录该用户的行为，还可以实现购物车。

在PHP 4及其以后版本中，PHP自身包含了会话控制函数。自从超级全局变量概念的引入，会话控制方法就发生了一些变化。如今可以使用$_SESSION超级全局变量。

23.2 理解基本的会话功能

PHP的会话是通过唯一的会话ID来驱动的，会话ID是一个加密的随机数字。它由PHP生成，在会话的生命周期中都会保存在客户端。它可以保存在用户机器里的cookie中，或者通过URL在网络上传递。

会话ID就像一把钥匙，它允许我们注册一些特定的变量，也称为会话变量。这些变量的内容保存在服务器端。会话ID是客户端唯一可见的信息。如果在一次特定的网站连接中，客户端可以通过cookie或URL看到会话ID，那么我们就可以访问该会话保存在服务器上的会话变量。在默认情况下，会话变量保存在服务器上的普通文件中。（如果愿意编写自己的函数，我们可以改变这种情况，使用数据库来保存这些变量——在“配置会话控制”一节将详细介绍该内容。）

我们可能曾经使用过一些网站，它们将会话ID保存到URL中，如果在URL中有一串看起来像随机数字的字符串，可能它就是某种形式的会话控制。

cookie是与会话不同的解决方法，它也解决了在多个事务之间保持状态的问题，同时，它还可以保持一个整洁的URL。

23.2.1 什么是cookie

cookie其实就是一小段信息，它可以由脚本在客户端机器保存。可以通过发送一个包含特定数据并且具有如下格式的HTTP标题头，从而在用户端机器设置一个cookie：

```
Set-Cookie:NAME=VALUE;[expires=DATE;][path=PATH;]
[domain=DOMAIN_NAME;][secure]
```

这会创建一个名为NAME，值为VALUE的cookie。除了该参数，其他参数都是可选的。expires域设置该cookie失效日期（请注意，如果失效日期不设置，cookie将永远有效，如果不手动将其删除的

话）。path和domain域合起来指定URL或与cookie相关的URL。secure关键字的意思是在普通的HTTP连接中不发送cookie。

当浏览器连接一个URL时，首先要搜索当地保存的cookie。如果有任何与正在连接的URL相关的cookie，浏览器将它提交到服务器。

23.2.2 通过PHP设置cookie

可以使用setcookie()函数在PHP中手动设置cookie。该函数原型如下所示：

```
bool setcookie(string name[,string value[,int expire[,string path
[,string domain[,int secure]]]]])
```

参数和前述的Set-Cookie标题对应。

如果按如下方式设置一个cookie：

```
setcookie('mycookie','value');
```

当用户访问站点中的下一页（或者重载当前页）的时候，可以通过名为$_COOKIE['mycookie']来实现。

可以调用setcookie()来删除一个cookie，setcookie()中的参数为要删除的cookie名称和已经过去的到期时间。也可以使用前面已经给出的cookie语法并通过header()函数来手动设置cookie。需要注意的一点是，cookie标题头必须在发送其他标题头之前发送，否则就无效（这是cookie的限制，而不是PHP的限制）。

23.2.3　在会话中使用cookie

使用cookie也存在一些问题：一些浏览器不接受cookie，一些用户可能将其浏览器的cookie功能关闭了。这也就是PHP会话控制使用cookie/URL双模式的原因之一（我们将在稍后的内容中详细介绍它）。

当使用PHP会话的时候，不必手动设置cookie。会话函数可以完成该操作。

可以使用函数session_get_cookie_params()来查看由会话控制设置的cookie内容。它将返回包含元素lifetime、path、domain和secure的相关数组。

也可以使用：

```
session_set_cookie_params($lifetime,$path,$domain[,$secure]);
```

来设置会话cookie的参数。

如果希望了解更多关于cookie的内容，可以参考Netscape网站中关于cookie的规范说明：http://wp.netscape.com/newsref/std/cookie_spec.html。

（我们很可能会忽略这个事实，那就是Netscape将其称为“基础规范”——该规范在1995年就已经出现了。它其实是类似于标准的文档，虽然还不是标准，但是大家都将其看作标准。）

23.2.4　存储会话ID

在默认情况下，PHP将在会话中使用cookie。如果可能，可以设置一个cookie用来存会话ID。

另一个使用会话的方法是将会话ID添加到URL中。如果在php.ini文件设置了session.use_trans_sid指令，则可以自动实现它。当然，启用该设置后，在使用时应该非常小心。因为它增加了站点的安全风险。如果将该指令设置为ON，用户就可以将包含该会话ID的URL通过电子邮件发送给其他人，而该URL就可以保存在一个公众可以访问的计算机中，或者在一个公众可以访问计算机的浏览器历史记录或书签。或者，可以手动将会话ID嵌入到链接中，这样它就可以一起传递到用户端。会话ID保存在常量SID中。要手动将其传递过去，需将它添加到一个与GET参数类似的链接末尾：

```
<A HREF="link.php?<?php echo strip_tags(SID);?>">
```

（在这里，使用strip_tags()函数可以避免一些跨站点的攻击。）

通常，如果可能的话，选中--enable-trans-sid选项对PHP进行编译会更容易一些。

23.3 实现简单的会话

使用会话的基本步骤如下：

1.开始一个会话

2.注册会话变量

3.使用会话变量

4.注销变量并销毁会话

请注意，这些步骤不一定都要发生在同一个脚本中，其中的一些步骤可以在多个脚本中发生。接下来，我们将依次讨论这些步骤。

23.3.1 开始一个会话

在使用会话功能前，必须开始一个会话。可以通过如下介绍的两种方法来开始一个会话。

第一种方法，也是最简单的方法，就是以调用session_start()函数开始一段脚本：

```
session_start();
```

该函数将检查是否有一个会话ID存在。如果不存在，就创建一个，并且使其能够通过超级全局数组$_SESSION进行访问。如果已经存在，将这个已经注册的会话变量载入以便使用。

你必须在使用会话的脚本开始部分调用session_start()函数。如果没有调用这个函数，所有保存在该会话的信息都无法在脚本中使用。

第二种方法是将PHP设置成当有用户访问网站的时候就自动启动一个会话。可以使用php.ini文件中的session.auto_start选项完成该设置——我们在讨论配置的时候再详细介绍它。这种方法有一个很大的缺

点：启用auto_start设置导致无法使用对象作为会话变量。这是因为该对象的类定义必须在创建该对象的会话开始之前载入。

23.3.2　注册一个会话变量

最近，在PHP中，会话变量的注册已经有所改变。自从PHP 4.1版本以后，会话变量保存在超级全局数组$_SESSION中。要创建一个会话变量，只需在这些数组中设置一个元素，如下所示：

```
$_SESSION['myvar']=5;
```

以上代码创建会话变量只有在会话结束或手动重置它时才会失效。根据php.ini文件对会话gc_maxlifetime指令设置，该会话也可能会过期。该指令将确定会话的持续时间（秒为单位），超过时间，该会话将被垃圾回收。

23.3.3 使用会话变量

要使一个会话变量在某个范围内可以使用，必须首先使用session_start()函数启动一个会话。这样，就可以通过$_SESSION超级全局数组访问这个变量了。例如，$_SESSION['myvar']。

当使用对象作为会话变量时，在调用session_start()函数重新载入会话变量之前，必须包含该类对象的定义。这样，PHP就知道如何构建这个会话对象。

相反，在检查会话变量是否已经被设置时（例如，通过isset()函数或empty()函数），必须非常小心。请记住，变量可以被用户通过GET或POST设置。可以通过检查$_SESSION数组来确定一个变量是否是注册的会话变量。可以通过如下所示的代码来检查：

```
if(isset($_SESSION['myvar'])...
```

23.3.4　注销变量与销毁会话

当使用完一个会话变量后，可以将其注销。通过注销$_SESSION数组的适当元素，可以直接注销该变量，如下所示：

```
unset($_SESSION['myvar']);
```

请注意，session_unregister()函数和session_unset()函数不再是必须的和推荐的方法。这些函数是在引入$_SESSION数组之前才使用的。

我们不能销毁整个$_SESSION数组，因为这样将禁用会话功能。要一次销毁所有的会话变量，可以使用如下所示语句：

```
$_SESSION=array();
```

当使用完一个会话后，首先应该注销所有的变量，然后再调用：

```
session_destroy();
```

来清除会话ID。

23.4　创建一个简单的会话例子

上面的内容看起来可能还有些抽象，下面让我们来看一个例子。我们将实现3个页面。

在第1页里，我们启动了一个会话并注册了$_SESSION['sess_var']变量。代码如程序清单23-1所示。

程序清单23-1　pagel.php——启动一个会话并注册一个变量

```
<?php
session_start();
$_SESSION['sess_var']="Hello world!";
echo'The content of$_SESSION[\'sess_var\']is'
.$_SESSION['sess_var'].'<br/>';
?>
<a href="page2.php">Next page</a>
```

我们已经注册了该变量并设置了它的值。该脚本输出如图23-1所示。

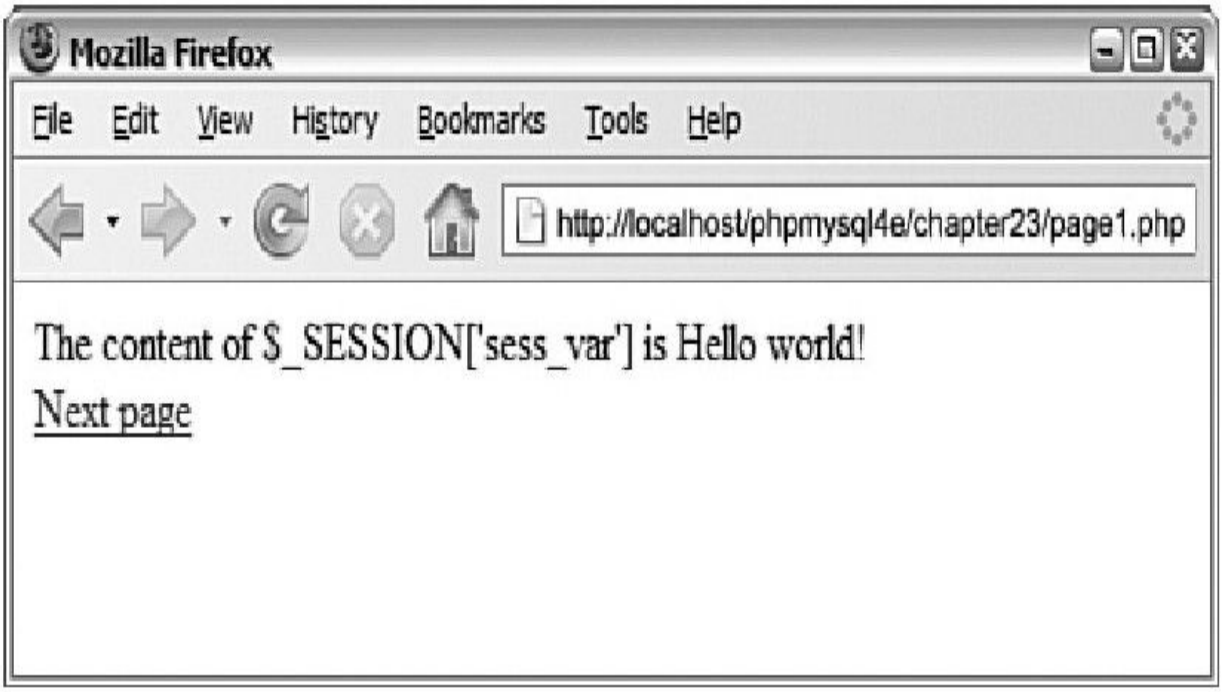


图　23-1　pagel.php显示出的会话变量初始值

在该页面上，该变量的最终值可以被后续页面使用。在该脚本末尾，会话变量被序列化了，或者说被冻结了，直到再次调用session_start()函数后，该变量才会被载入。

因此，在接下来的脚本中，我们将调用session_start()函数。该脚本如程序清单23-2所示。

程序清单23-2　page2.php——访问一个会话变量并注销它

```
<?php
session_start();
echo'The content of$_SESSION[\'sess_var\']is'
.$_SESSION['sess_var'].'<br/>';
unset($_SESSION['sess_var']);
?>
<a href="page3.php">Next page</a>
```

在调用session_start()之后，可以获取$_SESSION['sess_var']变量和前面所存储的值，如图23-2所示。

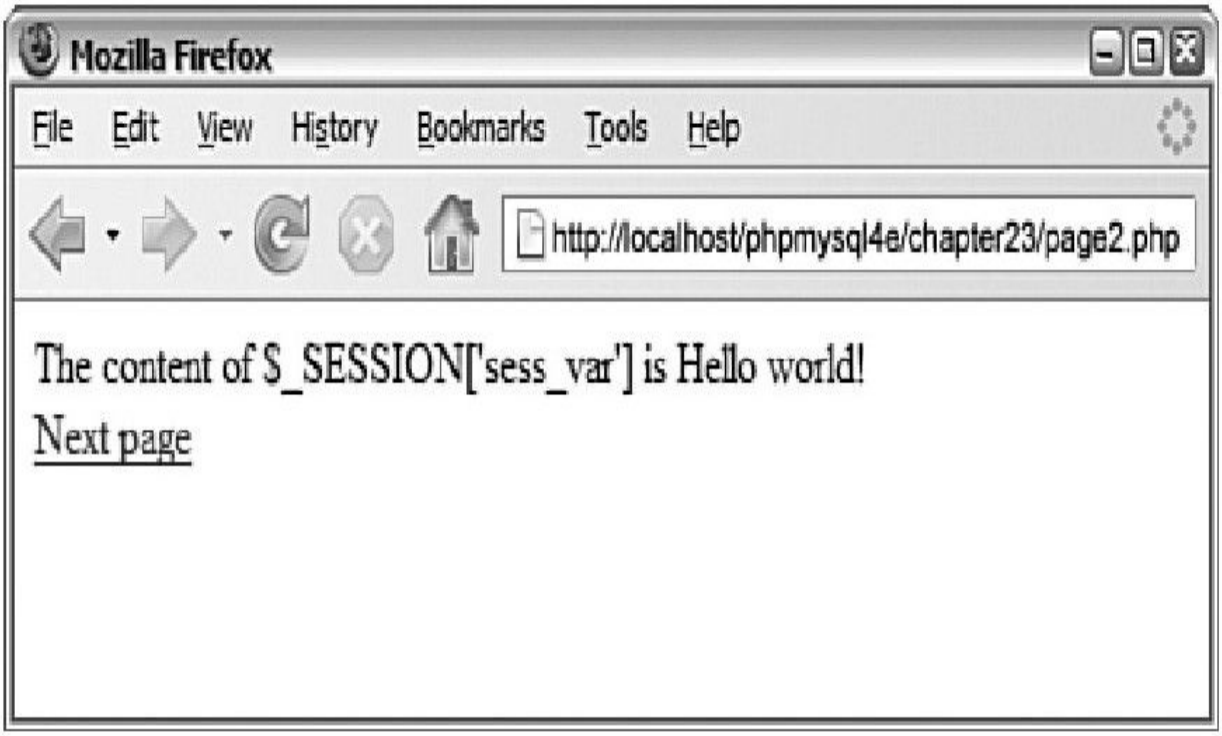


图　23-2　会话变量的值已经通过会话ID传递到page2.php

在使用了该变量之后，必须注销该变量。这样，会话仍然存在，但是变量$_SESSION['sess_var']不再是一个注册变量。

最后，我们将进入到page3.php页面，它也是本例最后一页。该脚本的代码如程序清单23-3所示。

程序清单23-3　page3.php——结束会话

```
<?php
session_start();
echo'The content of$_SESSION[\'sess_var\']is'
.$_SESSION['sess_var'].'<br/>';
session_destroy();
?>
```

可以从图23-3看出，我们将无法再访问$_SESSION['sess_var']变量值。

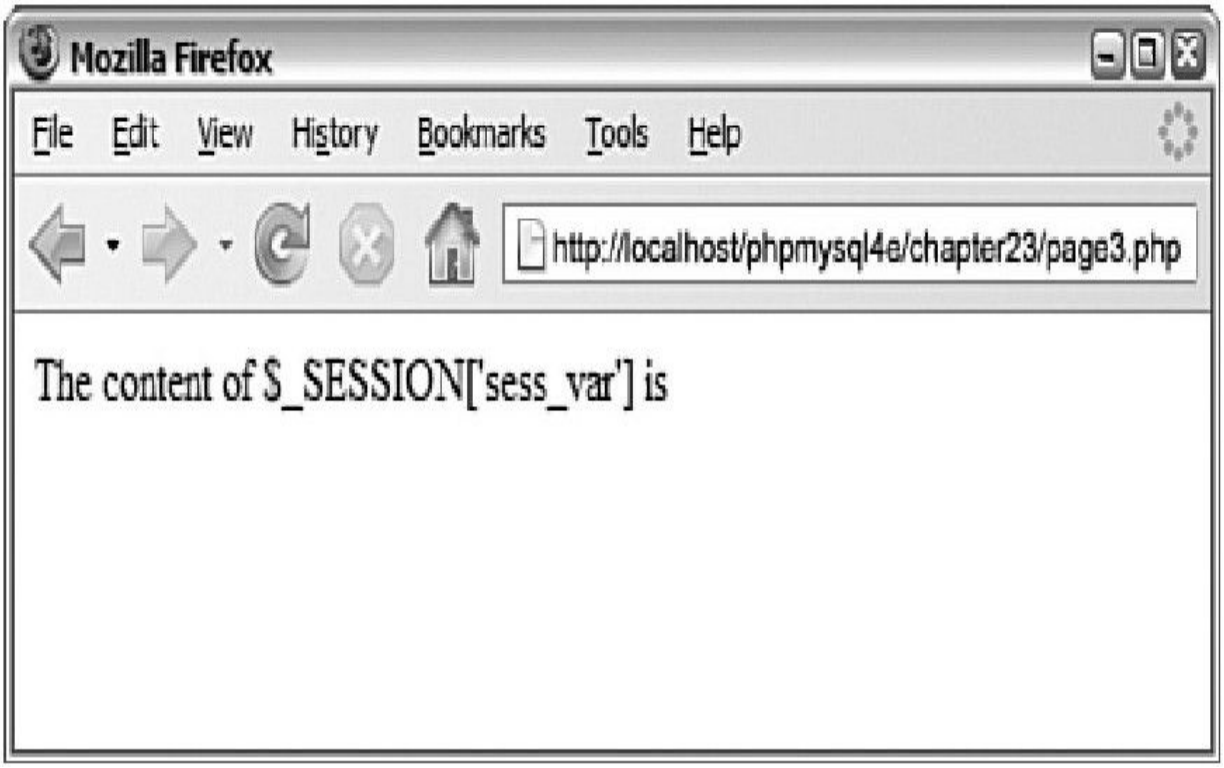


图　23-3　注销的变量不再有效

在PHP 4.3以前的版本中，当尝试注销$HTTP_SESSION_VARS数组或$_SESSION数组中的变量时，可能会遇到一个问题。如果发现无法注销元素（也就是说，它们还是处于注册状态），可以使用session_unregister()函数来清除这些变量。

通过调用session_destroy()函数清除会话ID。

23.5 配置会话控制

php.ini配置文件中有一组会话配置选项，可以对其进行设置。一些有意义的选项及其描述如表23-1所示。

表23-1 会话配置选项

选项名	默认	效果
session.auto_start	0（被禁用）	自动启动会话
session.cache_expire	180	为缓存中的会话页设置当前时间，精确到分钟
session.cookie_domain	none	指定会话cookie中的域
session.cookie_lifetime	0	cookie会话ID将在用户的机器上延续多久。默认值0表示延续到浏览器关闭
session.cookie_path	/	在会话cookie中要设置的路径

（续）

选项名	默　认	效　果
session.name	PHPSESSID	会话的名称，在用户系统中用作会话名
session.save_handler	files	定义会话数据保存的地方。可以将其设置为指向一个数据库，但是需要编写自己的函数
session.save_path	""	会话数据存储的路径。更通常的情况下，传递给存储的参量由session.save_handler函数来处理和定义
session.use_cookies	1（允许使用）	配置在客户端使用cookie的会话
session.cookie_secure	0（被禁用）	确定是否应该在安全连接中发送Cookie
session.hash_function	0（MD5）	允许指定用来生成会话ID的哈希算法。“0”表示使用MD5（128位）；“1”表示SHA-1（160位）。这种设置是在PHP 5.0版本中引入的。

23.6 通过会话控制实现身份验证

最后，我们将介绍使用会话控制的实际应用例子。

会话控制最常见的用法可能就是在用户通过一个登录机制验证后跟踪该用户的行为。在这个例子中，我们将结合MySQL身份验证功能和会话控制功能的使用来实现该功能。该功能将作为我们在第27章“建立用户身份验证机制和个性化设置”中实现的项目的基础，将在那个项目中再次使用。我们将再次使用在第17章“使用PHP和MySQL实现身份验证”中创建的身份验证数据库。可以查看程序清单17-3了解该数据库的详细信息。

本例包含3个简单的脚本。第一个是auth-main.php，它提供了一个登录表单，并且为站点成员提供了身份验证。第二个是members_only.php，只向成功登录的成员显示信息。第三个是logout.php，用户退出登录。

要了解这些是如何工作的，请查阅图23-4，它是由authmain.php显示的初始页面。



图　23-4　因为用户尚未登录，所以显示的是一个登录页面

该页给出了用户登录的地方。如果用户没有登录而访问成员区，将得到如图23-5所示的信息。



图　23-5　用户没有登录就不能看到网站内容；网站将显示警告信息

然而，如果用户先登录了（用户名：testuser；密码password。在第16章中，我们已经创建了这个用户）然后尝试浏览成员页面，看到的页面如图23-6所示。



图　23-6　用户已经登录，可以访问成员区的内容

下面，我们看一下该应用程序的代码。该程序的大部分代码在authmain.php中，该脚本如程序清单23-4所示。我们将详细介绍它。

程序清单23-4　authmain.php——身份验证应用程序的主体部分

```
<?php
session_start();
if(isset($_POST['userid'])&&isset($_POST['password']))
{
//if the user has just tried to log in
$userid=$_POST['userid'];
$password=$_POST['password'];
$db_conn=new mysqli('localhost","webauth','webauth','auth');
if(mysqli_connect_errno()){
echo'Connection to database failed:'.mysqli_connect_error();
```

```
exit();
}
$query='select*from authorized_users'
."where name='$userid'"
."and password=sha1('$password')";
$result=$db_conn->query($query);
if($result->num_rows)
{
//if they are in the database register the user id
$_SESSION['valid_user']=$userid;
}
$db_conn->close();
}
?>
<html>
<body>
<h1>Home page</h1>
<?
if(isset($_SESSION['valid_user']))
{
echo'You are logged in as:'.$_SESSION['valid_user'].'<br/>';
echo'<a href="logout.php">Log out</a><br/>';
}
else
{
if(isset($userid))
{
//if they"ve tried and failed to log in
echo'Could not log you in.<br/>';
}
else
{
//they have not tried to log in yet or have logged out
echo'You are not logged in.<br/>';
}
//provide form to log in
```

```
echo'<form method="post"action="authmain.php">";
echo'<table>';
echo'<tr><td>Userid:</td>';
echo'<td><input type="text"name="userid"></td></tr>';
echo'<tr><td>Password:</td>';
echo'<td><input type="password"name="password"></td></tr>';
echo'<tr><td colspan="2"align="center">";
echo'<input type="submit"value="Log in"></td></tr>';
echo'</table></form>';
}
?>
<br/>
<a href="members_only.php">Members section</a>
</body>
</html>
```

因为它显示了登录表单，所以该脚本包含了一些比较复杂的逻辑；表单的行为也比较复杂，因为它包含了成功和失败登录操作的HTML代码。

脚本的执行是围绕会话变量valid_user展开的。其基本思想是如果某人成功登录，我们将注册一个$_SESSION['valid_user']的会话变量，该变量包含用户的ID。

在该脚本中，我们所做的第一件事就是调用session_start()函数。如果会话变量valid_user已经创建，这个函数调用将载入该变量。

在第一次执行脚本的时候，if条件均不成立，用户将直接进入脚本的末尾，脚本告诉用户尚未登录并提供一个表单以便登录。

```
echo'<form method="post"action="authmain.php">";
echo'<table>';
echo'<tr><td>Userid:</td>';
echo'<td><input type="text"name="userid"></td></tr>';
echo'<tr><td>Password:</td>';
echo'<td><input type="password"name="password"></td></tr>';
```

```
echo'<tr><td colspan="2"align="center">";
echo'<input type="submit"value="Log in"></td></tr>';
echo'</table></form>';
```

当用户点击表单上的提交按钮后，这个脚本将从顶端开始重新执行。这次，我们有了用以验证的用户名和密码，它们分别存储在$_POST['userid']和$_POST['password']中。如果用户设置了这些变量，我们就可以进入身份验证模块：

```
if(isset($_POST['userid'])&&isset($_POST['password']))
{
//if the user has just tried to log in
$userid=$_POST['userid'];
$password=$_POST['password'];
$db_conn=new mysqli('localhost","webauth','webauth','auth');
if(mysqli_connect_errno()){
echo'Connection to database failed:'.mysqli_connect_error();
exit();
}
$query='select*from authorized_users'
."where name='$userid'"
."and password=sha1('$password')";
$result=$db_conn->query($query);
```

我们连接到一个MySQL数据库并检查用户ID和密码。如果它们与数据库中的数据匹配，将创建变量$_SESSION['valid_user']，它包含该用户的ID，因此我们就知道谁登录进来了，并对其进行跟踪。

```
if($result->num_rows>0)
{
//if they are in the database register the user id
$_SESSION['valid_user']=$userid;
}
$db_conn->close();
```

```
}
```

由于我们知道该用户是谁，就不必再向他显示登录表单。取而代之，告诉他我们知道他是谁，并提供退出的选项：

```
if(isset($_SESSION['valid_user']))
{
echo'You are logged in as:'.$_SESSION['valid_user'].'<br/>';
echo'<a href="logout.php">Log out</a><br/>';
}
```

如果试图让用户登录，但是由于某种原因失败了，我们将拥有一个用户ID而不是$_SESSION['valid_user']变量，因此我们为该用户提供一个错误信息。

```
if(isset($userid))
{
//if they"ve tried and failed to log in
echo'Could not log you in.<br/>';
}
```

以上就是主要的脚本逻辑。现在，我们来了解成员页面。成员页面的代码如程序清单23-5所示。

程序清单23-5　members_only.php——网站的有效用户检查部分，可以确定是否是有效成员

```
<?php
session_start();
echo'<h1>Members only</h1>';
//check session variable
if(isset($_SESSION['valid_user']))
{
echo'<p>You are logged in as'.$_SESSION['valid_user'].'</p>';
```

```
echo'<p>Members only content goes here</p>';
}
else
{
echo'<p>You are not logged in.</p>';
echo'<p>Only logged in members may see this page.</p>';
}
echo'<a href="authmain.php">Back to main page</a>';
?>
```

以上代码非常简单。它所做的只是启动一个会话，并且通过检查$_SESSION['valid_user']变量是否被设置来检查当前的会话是否包含一个注册的用户。如果用户登录进入我们的网站，我们显示成员内容；否则告诉他尚未通过身份验证。

最后，我们将讨论logout.php脚本，该脚本可以让用户退出登录。脚本代码如程序清单23-6所示。

程序清单23-6　logout.php——该脚本注销会话变量并销毁会话

```
<?php
session_start();
//store to test if they*were*logged in
$old_user=$_SESSION['valid_user'];
unset($_SESSION['valid_user']);
session_destroy();
?>
<html>
<body>
<h1>Log out</h1>
<?php
if(!empty($old_user))
{
echo'Logged out.<br/>';
}
```

```
else
{
//if they weren't logged in but came to this page somehow
echo'You were not logged in,and so have not been logged out.<br/>';
}
?>
<a href="authmain.php">Back to main page</a>
</body>
</html>
```

源代码十分简单，当然，它也只完成了一些简单的工作。在这里，我们启动了一个会话，保存用户的旧用户名，注销valid_user变量，销毁了会话。然后，我们给用户发送一个消息，该消息在用户退出或者尚未登录的情况下的具体内容是不同的。

以上简单的脚本是在后续章节中要完成的许多工作的基础。

23.7 进一步学习

关于cookie的更多信息，请访问如下URL：

http://home.netscape.com/newsref/std/cookie_spec.html.

23.8 下一章

本篇内容就此结束。在开始示例项目之前，我们还要先简要讨论一下本书其他部分没有介绍的PHP中的一些零碎但实用的功能。

第24章　其他有用的特性

PHP中有一些非常实用的功能和特征，它们不属于任何特定的类型。本章将专门介绍这些内容。

在本章中，我们将主要介绍以下内容：

■使用eval()函数对字符串求值

■中断执行：die和exit

■序列化变量和对象

■获取PHP环境信息

■暂时改变运行时环境

■加亮源代码

■在命令行中使用PHP

24.1　使用eval()函数对字符串求值

函数eval()可以计算出PHP代码字符串的值。例如：

```
eval("echo'Hello World';");
```

将接收字符串的内容并执行它。这行代码将产生与如下语句运行结果相同的输出：

```
echo'Hello World';
```

eval()函数可以在许多不同的情况下使用。例如，我们可以在数据库里存储一段代码，以便此后可以检索它们并求值。也可能在一个循

环里生成代码，然后使用eval()来执行它。

eval()函数的最常见用法是系统模板化。我们可以从数据库中载入HTML、PHP和纯文本的混合。模板系统可以对这些内容格式化并且通过eval()函数来执行任何PHP代码。

可以使用eval()函数有效地更新和校正已有代码。如果我们知道一堆脚本需要在什么地方进行修改，那么可以写一段新的脚本代码，将老的脚本代码加载到一个字符串中（效率较低），运行regexp进行修改，然后用eval()执行修改过的脚本。

我们甚至可以想像，某人对系统非常信任时，可能会希望在浏览器端输入PHP代码，然后在服务器端执行。

24.2　终止执行：die()和exit()

在本书的前面内容中，我们已经介绍了使用语言结构exit来终止一段脚本的执行。回顾一下，该语句的使用只有一行代码，如下所示：

```
exit;
```

它不会返回任何值。也可以使用该函数的别名函数die()。

要实现一个功能性更强的终止操作，可以向exit()函数传递一个参数。这可以在终止代码执行之前输出一个错误信息或者执行特定的函数。这对于Perl程序员来说是非常熟悉的。例如：

```
exit('Script ending now');
```

更常见的是，这两个语句OR（与）一个可能失败的语句一起使用，主要用来实现推出错误。例如，打开一个文件或者连接数据库：

```
mysql_query($query)or die('Could not execute query');
```

如果不希望以上代码只是打印出一条简单的错误信息，可以在脚本中断之前调用一个函数：

```
function err_msg()
{
return'MySQL error was:'.mysql_error();
}
mysql_query($query)or die(err_msg());
```

这样做可以使用户了解程序失败的原因或关闭HTML元素的方法，或在输出缓冲中清除一个已经完成部分的页面，这对用户来说是

非常有帮助的。

或者，可以将错误信息以电子邮件的方式发送给自己，这样可以了解程序是否出现重要错误，还可以将错误信息添加到日志文件中。

24.3 序列化变量和对象

序列化是将可以保存在PHP变量或对象中的任何数据转换成字节流的处理过程，这个字节流可以存储在数据库中，或者通过URL在网页之间传递。如果不使用这个处理过程，很难存储和传递整个数组或对象的内容。

随着session（会话）控制的引入，序列化的作用有了一定的降低，数据的序列化现在可以使用会话控制来实现。事实上，为了将会话变量在HTTP请求之间存储，可以使用会话控制函数来序列化会话变量。

但是，我们可能还希望将PHP数组或对象保存到一个文件或数据库中。如果要这样做，就必须了解如何使用这两个函数：serialize()和unserialize()。

可以按如下方式调用serialize()函数：

```
$serial_object=serialize($my_object);
```

如果希望知道序列化都完成了哪些操作，可以通过查看序列化后返回的结果来获知。显而易见，它可以将对象或数组的内容转换为字符串。

例如，我们可以查看对一个简单的employee对象上执行序列化操作的结果。该对象的定义和初始化如下：

```
class employee
{
var$name;
var$employee_id;
}
$this_emp=new employee;
$this_emp->name='Fred';
```

```
$this_emp->employee_id=5324;
```

如果对其进行序列化操作并显示到浏览器，其输出结果为：

```
O:8:"employee":2:{s:4:"name";s:4:"Fred";s:11:"employee_id";i:5324;}
```

可以很容易看出原始对象数据和序列化后的数据之间的关系。

由于序列化后的数据只有文本，你可以将它写到数据库或其他位置。请注意，与往常的操作一样，应该使用mysql_real_escape_string()函数对将要写入数据库的数据进行处理，这样可以转义任何特殊字符。注意上述序列化字符串中的引号，我们就知道这是必要的。

要恢复原来的对象，可以调用unserialize()函数，如下所示：

```
$new_object=unserialize($serial_object);
```

显然，如果在将对象保存到数据库之前调用addslashes()函数，那么也需要在反序列化字符串之前调用stripslashes()函数。

当序列化类或使用它们作为会话变量时需要注意一点：在PHP能够重新实例化一个类之前，它必须知道类的结构。因此，必须在调用session_start()或unserialize()函数之前包含该类的定义文件。

24.4 获取PHP环境信息

在PHP中，有许多函数可以用来获取PHP的环境信息。

24.4.1 找到所加载的PHP扩展部件

使用get_loaded_extensions()函数和get_extension_funcs()函数，可以方便地了解哪些函数集合是可用的，以及在每个函数集合中又有哪些函数是可用的。

get_loaded_extensions()函数将返回PHP当前版本中可供使用的函数集合数组。如果给定特定函数集合名称或扩展部件名称，get_extension_funcs()返回该集合所包含的函数数组。

程序清单24-1中的脚本使用了这两个函数来获知当前安装的PHP所包含的函数。

程序清单24-1 list_functions.php——该脚本列出了PHP所有可供使用的扩展部件，每一个扩展所包含的函数

```
<?php
echo'Function sets supported in this install are:<br/>';
$extensions=get_loaded_extensions();
foreach($extensions as$each_ext)
{
echo"$each_ext<br/>";
echo'<ul>';
$ext_funcs=get_extension_funcs($each_ext);
foreach($ext_funcs as$func)
{
echo"<li>$func</li>";
}
echo'</ul>';
}
?>
```

请注意，get_loaded_extensions()函数不带任何参数，而get_extension_funcs()函数只带一个参数——扩展部件的名称。

如果要查询是否成功地安装了某个扩展部件，或者正在编写能够在安装过程中生成有用的诊断信息的可移植代码，这些信息是非常有用的。

24.4.2　识别脚本所有者

通过调用get_current_user()函数，可以识别当前处于运行状态的脚本的所有者。如下所示：

```
echo get_current_user();
```

以上语句可在解决权限问题的时候用到。

24.4.3　确定脚本最近修改时间

将最近修改时间写到站点的每一页面上是非常普遍的做法。

可以使用getlastmod()函数（注意函数名中间没有下划线）可以确定脚本的最近修改时间，如下所示：

```
echo date('g:i a,j M Y',getlastmod());
```

该函数将返回UNIX时间戳，它作为date()函数的参数（如上），从而产生可读的日期。

24.5　暂时改变运行时环境

可以在php.ini文件中查看指令集，或者也可以在一个脚本的运行周期中修改它们。这个特性是非常有用的。例如，如果我们知道脚本需要运行一段时间，可以通过设置max_execution_time指令来限制脚本的最长运行时间。

使用ini_get()函数和ini_set()函数，可以访问和修改指令。程序清单24-2显示了使用这些函数的简单脚本。

程序清单24-2　iniset.php——重置php.ini文件中的变量的脚本

```
<?php
$old_max_execution_time=ini_set('max_execution_time',120);
echo"old timeout is$old_max_execution_time<br/>";
$max_execution_time=ini_get('max_execution_time');
echo"new timeout is$max_execution_time<br/>";
?>
```

ini_set()函数需要两个参数。第一个参数是文件php.ini中的配置指令的名称，它是要修改的对象，第二个参数是修改后的值。该函数将返回配置指令的前一个值。

在这个例子中，我们将脚本最长运行时间的默认值30s（或任何在php.ini文件的设置值）修改为最大值120s。

ini_get()函数只检查特定配置指令的值。指令名称需要通过字符串形式传给该函数。在这个例子中，我们只是用它来检查指令值是否已经改变。

并不是所有的INI选项都是可以通过这种方式进行设置的。每一个选项都有允许设置的级别。可能的级别包括：

■PHP_INI_USER——可以通过ini_set()函数在脚本中改变这些值。

■ PHP_INI_PERDIR—— 如果使用Apache，可以改变php.ini、.htaccess或httpd.conf文件的设置值。能够在.htaccess中修改这些设置值意味着可以在目录级别修改这些值。

■PHP_INI_SYSTEM——可以修改php.ini或httpd.conf文件中的设置值。

■PHP_INI_ALL——可以使用以上任意一种方法修改这些值——也就是，在一个脚本中，在.htaccess文件，或者httpd.conf、php.ini文件中。

ini选项的完整集合和可以设置的级别可以在PHP手册中找到：http://www.php.net/ini_set。

24.6 源代码加亮

与许多IDE一样，PHP内置一个默认的语法加亮器。这项功能在与别人共享代码或在Web页面上讨论并展示这些代码时非常有用。

函数show_source()和highlight_file()是一样的（确切地说，show_source()函数是highlight_file()的别名函数）。这两个函数的输入参数都是一个字符串，表示文件名称（这个文件必须是PHP文件，否则加亮的结果将会是没有意义的）。例如：

```
show_source('list_functions.php');
```

执行以上代码后，该文件的代码将显示在浏览器窗口中，该文件内容将会以不同的颜色显示，而这些颜色是根据文本类型决定的，例如是否是字符串、注释、关键词或HTML。浏览器具有背景颜色。不属于这些类型的文本内容将以默认颜色显示。

highlight_string()函数的工作原理类似，但是它的参数是字符串，并且以语法加亮格式显示在浏览器中。

可以在php.ini文件中设置加亮语法的颜色。该文件中，设置语法加亮颜色的部分如下所示：

```
;Colors for Syntax Highlighting mode
highlight.string=#DD0000
highlight.comment=#FF9900
highlight.keyword=#007700
highlight.bg=#FFFFFF
highlight.default=#0000BB
highlight.html=#000000
```

加亮颜色是用标准的HTML RGB格式来表示的。

24.7　在命令行中使用PHP

通常，我们可以编写或下载许多小程序，并且在命令行下运行它们。在UNIX系统中，这些程序通常都是用shell脚本语言或Perl编写的。在Windows平台上，它们通常都是一个批处理文件。

对于一个Web应用程序，我们可能会首选PHP，但是使其成为一个优秀的Web语言的相同文本处理工具是它的命令行工具程序。

在命令行下，有3种方法来执行PHP脚本：通过一个文件、管道或直接在命令行下。

要在一个文件中执行PHP，请确认PHP可执行文件（php或php.exe，取决于操作系统）位于路径下，并且以脚本名称作为参数调用它。如下所示示例：

```
php myscript.php
```

myscript.php文件只是一个常规的PHP文件，因此它包含了任何常规的PHP语法和PHP标记。

要通过管道传递代码，可以运行任何能够生成有效的PHP脚本的程序，并且以该程序作为输出，导出到php可执行文件。如下示例使用了echo程序给出了一个单行的程序：

```
echo'<?php for($i=1;$i<10;$i++)echo$i;?>'|php
```

需要再次指出的是，以上的PHP代码封闭在PHP标记中（<?php和?>）。还需要注意的是，这是命令行程序，不是PHP的语言结构。

单行程序的特性使得该程序很容易在命令行下直接传递，如下例所示：

```
php-r'for($i=1;$i<10;$i++)echo$i;'
```

这里，情况有所不同。在这个字符串中传递的PHP代码并没有封闭在PHP标记中。如果将该字符串封闭在PHP标记中，将遇到一个语法错误。

为命令行使用而编写的PHP程序是不受限制的。我们可以为PHP应用程序编写安装程序。可以在将文本文件导入到数据库之前使用一个脚本来重新格式化这个文本文件。甚至还可以编写一个能够执行任何需要重复的任务，而这些任务需要在命令行下执行；一个很好的例子就是，可以编写一个PHP脚本，在开发用途的Web服务器中将所有PHP文件、图像和MySQL表格结构复制到产品服务器中。

24.8 下一章

本书的第五篇“创建实用的PHP和MySQL项目”，将介绍一些使用PHP和MySQL构建相对复杂的实用项目。这些项目为我们可能面对的任务提供了很好的例子，此外，它们还说明了如何在大型项目中应用PHP和MySQL。

第25章“在大型项目中使用PHP和MySQL”讨论了大型项目中使用PHP编程可能遇到的问题。这些问题包括软件工程的一些基本原则，例如软件设计、文档管理和变更管理。

第五篇　创建实用的PHP和MySQL项目

第25章　在大型项目中使用PHP和MySQL

在本书的前几篇中，我们已经讨论了PHP和MySQL各个不同的组件及其应用。尽管我们试图列举出一些有趣且互相联系的例子，但实际上它们非常简单，只包含一两个脚本，每个脚本最多只有100行左右代码。

当创建真正的Web应用程序时，编写代码就不会这么简单了。几年前，“交互式”网站还处于表单邮件的阶段。如今，Web站点已经变成了Web应用程序——也就是说，符合一定规范的软件已经整个分布在Web网站。这种改变清楚地表明了网站规模的改变。网站已经从几行脚本代码发展到如今的成千上万行代码。如此规模的项目需要像一般软件开发那样进行规划和管理。

在我们开始介绍本书这部分将要涉及的项目之前，先浏览一下管理大型Web项目需要使用到的一些技术。这可以说是一门新兴艺术，要正确掌握它非常困难。这一点在软件市场中不难发现。

在本章中，我们将主要介绍以下内容：

■在Web开发中应用软件工程

■规划和运行Web应用程序项目

■重用代码

■编写可维护代码

■实现版本控制

■选择一个开发环境

■项目的文档化

■建立原型

■分离逻辑、内容和外观：PHP、HTML和CSS

■优化代码

25.1　在Web开发中应用软件工程

我们都知道，软件工程是软件开发过程中系统化的、可以量化的开发方法。也就是说，它是工程原则在软件开发中的应用。

很明显，软件工程在当今Web项目中的应用是很少的。这主要有两个原因：

首先，网站开发的管理经常与书面报告的管理方法相同。它是文档结构、图形设计和产品化的运用。它是一个面向文档的模式。这对中小型静态网站来说是很适宜的，也很有效。

但是随着网站动态内容的增加，网站的功能由原来简单地提供文档发展到提供服务，这样的模式就不再适应了。很多人根本就没有想到将软件工程实践应用到Web项目中。

其次，软件项目之所以未能在Web项目中应用，还在于Web应用程序的开发与一般的应用程序的开发存在多方面的区别。对网站的设计，我们没有什么规划时间，但却面临很大压力，要求马上就要建好。对软件项目则要求有计划、有秩序地开发，并且会在规划上花费不少时间。因此，对网站设计来说，通常的感觉是没有时间规划。

当Web项目设计失败的时候，它出现的问题和任何软件设计失败时出现的问题是一样的：错误百出的应用程序、超过时间期限以及源代码晦涩难懂。

然而，关键是寻找适用于这种Web应用程序开发新规则的软件工程部分，而忽略非适用部分。

25.2 规划和运行Web应用程序项目

Web项目没有最好的设计方法或项目周期，但是，我们却必须为自己的项目仔细考虑一些问题。在这里，我们将给出这些问题，在后续内容中将详细讨论它们。这里列出的问题是按照一定顺序的，但是这个顺序如果不适合具体项目，可以不按照这个顺序。重要的是注意这些问题，选择适合具体项目的技术。

■在项目开始之前，必须明确要创建什么，创建的最终目标是什么。要考虑谁将使用这个Web应用程序；也就是说，谁是目标用户。从技术上说，许多Web项目都是很完美的，但是最终失败了，因为没有人考察是否有人对他们的应用程序感兴趣。

■要尝试将应用程序分成几个部分。应用程序有哪些部分或处理步骤？这些部分是如何工作的？它们之间能够相互协调吗？草拟其工作场景或举出一些例子，对于回答和解决这些问题是有效的。

■列出各个部分之后，看看哪些部分已经存在了。如果一个以前完成的模块已经具有了该部分的功能，我们需要考虑是否可以直接使用它。不要忘了在公司内外查找现存的代码。特别是在源代码开放的社区，许多现存的代码组件是可以免费使用的。决定哪些代码必须得从头编写，并且大略估计一下工作量。

■对工作进程问题作出决定。Web项目中这些问题太容易忽略。这里所说的工作进程问题，是指代码编写标准、目录结构、版本控制管理、开发环境、文档化级别和标准，以及对小组成员的任务分配。

■基于所有已经获得的信息构建一个原型，展示给用户。并反复修改和展示。

■请记住，对于所有这些，将应用程序的内容和逻辑分开是非常重要的。在后续的内容中，我们将详细解释这些观点。

■对系统进行必要的优化。

■像其他软件开发项目一样，一边开发，一边测试。

25.3 重用代码

通常，编程人员容易犯的一个错误是重写已经存在的代码。当知道需要什么应用程序组件或函数之后，应该在开发之前看看哪些组件和函数是可从别的地方获取的。

作为编程语言，PHP的一个优点就是它有大量内置的函数库。我们应该经常查看是否已经存在一个能够完成所需功能的函数。通常，找到所需的函数并不是很困难的。查找函数的一个好方法就是浏览按照函数组分类的手册。

有时候，编程人员重写函数是因为他们并没有查看手册，寻找是否有现存的函数已经提供了他们需要的功能。应该将手册页面作为书签保存在浏览器中，但是请注意，在线手册更新频繁。带注释的手册是一种非常好的资源，它包含了评注、建议和其他用户编写的示例代码。通常，这些资源还包含了问题报告和问题的解决方法。

如下站点提供了PHP手册的英文版本：http://www.php.net/manual/an/。

一些有着不同语言背景的编程人员可能会编写一些封装函数，并且使用这些封装函数完全改变PHP原函数的名称以适应他们熟悉的语言。这种做法有时候也称为“语法糖果”。它是一个糟糕的主意，这将使得别人很难读懂这些代码，从而带来代码维护问题。如果正在学习一门新的语言，需要学会怎样正确使用它。另外，增加一层函数调用也会使代码执行速度下降。考虑到这些因素，这种做法应该尽量避免。

如果发现所需的功能在PHP主函数库里没有提供，那么有两个选择。如果需要的功能非常简单，可以选择自己编写该函数或对象。然而，如果需要创建一个相当复杂的功能——例如购物车、Web邮件系统或Web论坛——我们会发现这些东西别人已经做好了。在开放源代码社区工作的一个动力就是像这些应用程序组件的代码经常是免费的。如果发现某个组件与需要构建的相似甚至相同，那么就可将这些源代码作为基础，在此基础上修改或创建自己的组件。

如果已经开发完毕自己的函数或组件，应该认真考虑将这些函数或组件公布到PHP社区。只有大家都遵循这种原则，才能使PHP开发者社区继续是一个有帮助的、充满活力的、知识丰富的群体。

25.4　编写可维护代码

通常，代码的可维护性问题在Web应用中很容易被忽略，尤其是当网站创建得非常匆忙的时候。有时候，开始编写代码并快速完成它看起来比最初进行规划更重要。但是，在开始工作以前投入一点时间可以节省接下来的许多时间，当在编写的过程中重复编写应用程序的时候，就会发现这一点。

25.4.1　编码标准

大多数IT组织都制定了编码标准——选择文件名、变量名以形成内部编码准则、注释代码的准则、代码缩进的准则等。

因为文档模式经常应用于网站开发，因此在这里，代码标准经常会被忽略。对于独立编码或者在一个较小的团队里编码的人来说，容易低估代码标准化的重要性。当小组和项目不断发展的时候，若还不实行代码标准化的话，不但自己忙得手忙脚乱，而且许多编程人员也会对现存的代码摸不着头脑。

1.定义命名惯例

定义一个命名规范的目的是为了：

■使代码易读。如果在定义变量和函数时使用的名称是非常直观的，就可以像阅读英语句子一样阅读代码，至少可以猜出它们的意思。

■使标识符易记。如果标识符格式统一，就更容易记住使用了什么变量，调用了什么函数。

变量名称应该能够描述其所包含的数据。如果使用变量表示一个人的姓氏，就可以给它取名$surname。也可以在变量名称的长度和可读性之间找一个平衡点。例如，将表示名字的变量取名为$n可以使其类型简单，但是这样的变量不具有良好的可读性。而如果取名为$surname_of_the_current_user，虽然可以提供详细的信息，但是对数据类型来说太长了（容易导致输入错误），并且不具有很大的价值。

还需要决定在什么时候使用大写。正如前面所提到的，变量名称在PHP中是区分大小写的。需要决定变量名称是否全部为小写、全部为大写或者两者混合，例如，将单词的第一个字母大写。我们倾向于全部使用小写，因为它们最容易记忆。

将常量和变量通过大小写区分是一个好主意——通常的模式是，变量都用小写表示（例如，$result），常量都用大写表示（例如，PI）。

一些编程人员喜欢将两个变量使用同一个名字，但是使用不同的大小写来区分，例如，$name和$Name，显而易见，这是非常糟糕的想法。

此外，最好还要避免一些可笑的大小写模式，例如，$WaReZ。没有人会记得它代表什么。

还要考虑多单词变量名称的命名模式。例如，我们已经见过如下所示的模式：

```
$username
$user_name
$userName
```

以上都是我们曾经见到过的命名模式。选择哪一种规范都是没有问题的，但关键是要一直保持使用这种规范。另外，对于多单词变量名，一个名字应最多包含2～3个单词。

函数名称的命名规范需要考虑的因素与变量名称命名规范有许多相似之处，当然也有其自身的特点。通常，函数名是面向动词的。PHP内置的函数，例如，addslashes()或mysqli_connect()清楚地描述了它们的功能或它们要传递的参数。这大大提高了代码的可读性。请注意，这两个函数虽然都是多单词函数名，但是使用了不同的命名模式。从这点来说，PHP的命名是不一致的。原因之一可能就是因为参与的开发人员太多。但是最主要的原因是许多函数名称是直接从不同的语言和API拿过来的，而未经过改动。

此外，还要记住，在PHP中函数名称是不区分大小写的。我们应该一直坚持一种特定的格式，以避免造成混乱。

我们可能希望在许多PHP模块中使用模块命名模式，也就是说，在函数名前都加上模块名称作为前缀。例如，所有的MySQL函数都以mysqli_开头。所有的IMAP函数都以imap_开头。如果代码里有一个购物车模块，则可在该模块中的函数名前加上cart_。

然而，请注意，PHP5提供了过程接口和面向对象接口，在这两种接口中，函数名称是不同的。通常，过程式接口使用下划线（my_function()），而面向对象接口使用一种名为studlyCaps的模式（myFunction()）。

最后，编写代码的时候使用哪种命名惯例或标准对程序本身并没有多大的影响。只要遵循和使用一致的方针。

2.对代码进行注释

所有的程序都应该有一定程度的注释语句。那么使用何种级别的注释是适当的呢？通常，应该考虑为如下每个项目添加注释：

■文件，无论是完整的脚本还是包含文件——每个文件应该有一个注释，说明该文件的名称、功能、作者和更新日期。

■函数——函数的注释应指明函数的功能，输入参数和返回值。

■类——注释应描述类的用途。类方法应该具有与函数描述相同的注释。

■脚本或函数里面的一大段代码——在脚本之前添加一段伪码样式的注释然后再编写代码是很好的。因此，初始化脚本可能是这样：

```
<?
//validate input data
//send to database
//report results
?>
```

这种注释模式是非常方便的，因为在函数调用或其他代码编写之前，代码的注释已经做好了。

■复杂的代码——当整天围绕某件事，或者花费了很长时间来解决这个问题时，应该编写一段注释解释那样做的原因。这样，当下次看到这段代码时，就不会皱着眉头想，“究竟我为什么会那样做呢？”

另一个通常要遵循的原则是一边写代码一边做注释。有人希望在完成项目之后回头注释自己的代码。我敢保证，除非他没有惩罚性的开发时间表并且自律力比我们强，否则这样的事情不会发生。

3.代码缩进

就像在所有的编程语言中一样，我们应该清楚一致地缩进代码。就像个人简历或商业信件的布局一样。缩进使代码更易读，更快地让人理解。

通常，任何属于某个控制结构的程序块都应该缩进于周围的代码。缩进的深度要明显（也就是说，大于一个空格）但不能太大。通常，要避免使用tabs键。虽然容易敲入，但是在许多人的屏幕上它占了太多的空间。我们在所有的项目中都使用了2～3个空格的缩进。

大括号的布局也是一个问题。最常用的两种布局模式如下所示：

模式1：

```
if(condition){
//do something
}
```

模式2：

```
if(condition)
{
```

```
    //do something else
    }
```

使用哪一种模式由自己决定。但是一旦选择一个，就要在整个项目中保持一致，以免造成混淆。

25.4.2　分解代码

单段庞大的代码是非常可怕的。有人会编写这样一个庞大的脚本，它将所有的操作在一段主代码中完成。最好将它分解成多个函数或类，并将相关的函数和类保存到包含文件中。例如，可以将所有与数据库相关的函数放到一个名为dbfunctions.php的文件中。

将代码分解成清晰的代码块，原因如下所示：

■它使源代码易读、易懂。

■它可以提高源代码的重用性，将代码冗余减少到最小。例如，对前述的dbfunctions.php文件，可以在需要连接数据库的每个脚本中重用它。如果需要改变一种方式，只要在一个地方改变它即可。

■方便团队工作。如果代码分成多个部分，就可将这些部分的编程责任分配给不同的团队成员。这就意味着，可以避免一个编程人员在等待另一个编程人员完成大块脚本GiantScript.php这样的情形，前者可以继续干他自己的工作。

在一个项目开始的时候，应该花些时间考虑如何将一个项目分成有计划的部分。这需要将功能区域分成多个部分。但也不要一味地陷进去，因为项目开始后功能区域的划分可能会变化。同时，还要决定哪些部分需要首先完成，哪些组件要基于另外的部分，还有开发所有组件的时间规划。

即使团队所有的工作人员分别编写各部分代码，通常将每个组件的主要责任分配给特定的人员是一个好办法。最终，如果某人的任务出了问题，该责任就可以由责任人承担。当然，还需要有人承担项目的管理工作——也就是说，他将确保各个部分工作正常并且能与其他部分配合工作。通常，这个人还要管理源代码的版本控制——在本章的后续内容中，我们将详细介绍版本控制。此人可以是项目经理，或者可以为他分配独立的责任。

25.4.3　使用标准的目录结构

在开始一个项目的时候，必须考虑组件结构如何反映到网站目录结构中。就像用一个宠大的脚本实现所有的功能是十分糟糕的一样，用一个大目录包含所有的东西也是非常糟糕的做法。决定如何按照组件、逻辑、内容和源代码库将目录分成多个部分。对目录结构进行文档化处理，并确认开发本项目的每一位工作人员都有一个副本，以确保他们可以从中查找需要的东西。

25.4.4　文档化和共享内部函数

在开发函数库的时候，应该允许开发队伍的每一位工作人员都可使用这些函数。通常，队伍中的每一个人都编写他们自己的一套数据库、日期和调试函数。事实上，这是一种时间浪费。应该让别人可以获取我们的函数和类。

请记住，即使代码存在于可以访问的区域或目录中，如果不告诉他们，他们也不会知道。开发一个能够文档化内部函数库的系统，并使开发队伍中每一个编程人员可以获取它。

25.5 实现版本控制

版本控制是一门“艺术”，它适用于软件开发中的并发变更管理。通常，版本控制系统用作中央信息库或存档，它为访问和共享别人代码（或文档）提供一个可以控制的接口。

设想这样的情形，我们尝试去改进一些代码，但是却不幸将它弄坏，而且无论怎样也恢复不到原来的样子。或者，我们或客户认为网站早期的版本更好。或者，因为法律的原因必须回到早期版本。

设想另一种情形，开发队伍的两个成员希望对同一个文件进行修改。它们可能同时打开并编辑该文件，覆盖对方的修改。它们可能都工作于当地副本上，但是以不同的方式修改。如果可能发生这样的事情，必须让一位编程人员等待另一位编程人员完成对该文件的修改。

使用版本控制系统，可以解决所有这些问题。该系统可以记录信息库里每一个文件的修改情况，这样，不仅可以看到对它现在的描述，也可以清楚地了解到过去任何时候它的样子。版本控制系统的这个特性可将弄乱的代码恢复到已知的可工作的版本。也可以对一系列的特定文件设置标签，将其作为要发布的版本，这就意味着可以在这些代码的基础上继续开发，也可以随时获得已发表版本的副本。

版本控制还有助于解决多个编程人员合作编写程序的问题。每个编程人员可以在信息库里获取代码的一个副本（称为checking it out），并且当他们完成修改后，将这些修改提交到信息库（称checked in或committed）。版本控制系统也因此能够跟踪谁对某系统做了什么修改。

通常，这些系统还有管理并发更新的优点。并发更新是指两个编程人员可能在同一时刻修改同一个文件。例如，假设John和Mary都获取了他们所做项目的最近发布的一个副本。John完成了对某一文件的修改并将其提交到信息库。Mary也修改了那个文件，也试图提交。如果他们修改的不是文件的同一部分，版本控制系统就会将两个版本的文件合并。如果他们的修改互相冲突，系统将通知Mary并显示两种不同的版本。她就可以调整自己的代码以避免冲突。

大多数UNIX开发人员和开放源代码开发人员使用的版本控制系统是CVS。CVS是并发版本系统的缩写。CVS也是开放源代码。它与每个版本的UNIX捆绑在一起，也可以将其用于运行DOS或Windows的PC机和Mac机。它支持客户端-服务器模式，因此如果CVS服务器在网上是可见的，也可通过互联网进行连接，并从任何机器提交或获取代码。它用于PHP、Apache和Mozilla以及其他项目的开发，部分是因为这个原因。

从CVS的主页，可以为系统下载合适的CVS：http://ximbiot.com/cvs/wiki/。

尽管基本CVS系统是一个命令行工具，然而不同的附加软件给它提供了美观的前台，包括基于Java的前台和Windows前台。这些也可以从CVS主页访问到。

Bitkeeper是一个非常有竞争力的版本控制产品，它为几个高性能的开放源代码项目所使用，其中包括MySQL和Linux内核。它对开放源代码项目来说是免费的：http://www.bitkeeper.com/。

当然，也可以选择商业化的CVS软件。其中之一就是运行在大多数常见平台上的perforce，它支持PHP。请注意，虽然它是商业软件，但是可以在开放源代码项目主页中找到其免费的许可：http://www.perforce.com/。

25.6 选择一个开发环境

前面关于版本控制的讨论就会带来更多关于开发环境选择的话题。我们真正需要的是一个文本编辑器和一个用来测试的浏览器，但是大多数编程人员还是非常喜欢集成开发环境（IDE），通常他们在IDE下的开发效率更高。

目前，有许多适用于专门的PHP IDE的免费项目，包括KPHPDevelop，该开发环境适用于Linux平台的KDE桌面环境，可以从如下站点获得它：http://kphpdev.sourceforge.net/。

然而，目前最好的PHP IDE都是商业性的。例如，zend.com公司的Zend Studio，active-state.com公司的Komodo，以及nusphere.com公司的PHPEd，都提供了功能全面的集成开发环境。所有这些产品都提供了可供下载免费试用的版本，但要长期使用，必须付费。Komodo提供了一个非常便宜的非商业许可。

25.7 项目的文档化

我们可以为正在开发的项目制作许多种文档，这些文档包括但是不局限于下面这些：

■设计文档

■技术文档/开发指南

■数据词典（包含类文档）

■用户指南（尽管大多数Web应用程序是自我解释的）

在这里，我们并不是要讲解如何写技术文档，而是建议自动化这些过程，从而减轻开发工作的负担。

在一些语言中，有一些自动生成这些文档的方法——特别是技术文档和数据词典。例如，javadoc可以生成HTML格式的树型结构文件目录，该树型结构就包含了Java程序中类成员的原型及其描述。

对于PHP，也有一些可以提供该功能的工具。其中包括如下这些：

■phpdoc，可以从如下站点获取：http://www.phpdoc.de/。

它是PEAR用来对代码执行文档化处理的系统。请注意，术语phpDoc用于描述几个这种类型的项目，这就是其中之一。

■ PHPDocumentor，可以从如下站点获取：http://phpdocu.sourceforge.net。

PHPDocumentor可以提供非常类似于javadoc的输出，而且稳定性非常好。看上去，PHPDocumentor的开发团队比我们这里介绍的其他两个开发团队更主动。

■ phpautodoc，可以从如下站点获取：http://sourceforge.net/projects/phpautodoc/。

phpautodoc的输出结果也非常类似于javadoc的输出结果。

查找更多的这种类型（和常用的PHP组件）应用软件的一个好地方是SourceForge：http://sourceforge.net。

SourceForge主要用于UNIX/Linux社区，但是这里也有许多运行于其他平台的项目。

25.8 建立原型

通常，建立原型是一个适用于开发Web应用程序的开发周期。一个原型是对于获得用户需求很有意义的工具。通常，原型是应用程序的简化的一部分工作版本，可以用来与客户进行讨论，并作为最终系统的基础。对原型的多次反复讨论将最终产生应用程序。这种方法的好处就是它让我们更紧密地同客户或者终端用户一起工作，从而产生一个他们喜欢并且有主人翁感的系统。

为了能够较快地将一个原型“凑到一起”，需要一些特别的技术和工具。这是一个基于组件的方法发挥很好作用的地方。如果可以访问一系列已经存在的组件，这些组件在内部和公共地方都可以访问，我们将能够更快地实现它。另一个有用的快速开发原型的工具是模板。我们将在下一节详细介绍它。

使用建立原型的方法有两个主要问题。我们必须意识到这些问题到底是什么，从而避免它们，使这种方法发挥最大潜力。

第一个问题是编程人员通常会发现他们会因为某种原因很难丢弃他们所编写的代码。原型的编写通常会很快。可是过后，我们会发现自己没有以理想的或者近似理想的方法建立原型。代码的部分可以修改，但是如果整体结构错误，那就麻烦了。问题是Web应用程序经常要在巨大的时间压力下创建，因而没有时间来修改它。这样我们就被一个极难维护的糟糕设计缠住了。

可以像本章前面讨论的那样做一些计划来避免这个问题。但是也要记住，有时候，从头开始再次写新代码比修改原代码还容易。尽管这可能看起来是没有时间来完成的，但是在以后它可能会减少许多麻烦。

建立原型的第二个问题是一个系统可能会以一个无休止的原型结束。每次认为完成了，雇员又将提出更好的改进意见或者附加功能或网站更新。这种发展的蔓延可能让我们没完没了地构建一个项目。

要避免这些问题，需要草拟一份设计计划，规定项目的反复次数，设定某个日期之后如果不重新计划、预算和规划就不能添加新功能。

25.9 将逻辑和内容分离

我们可能对使用HTML描述一个网络文档结构，和用级联样式表单（CSS）描述其外观比较熟悉。这种将外观从内容分离出来的想法可以扩展到脚本编码中去。一般来说，如果将逻辑从内容中分离、内容从外观分离，那么网站将更易长期使用和维护。这就归结为PHP和HTML的分离。

对于只有行数不多的代码或脚本的简单项目，使用分离法是得不偿失的。但是当项目扩大的时候，就必须找到将逻辑和内容分离的方法。如果不这样分离，代码将越来越难维护。如果我们决定将某一新的设计应用于网站，要在原来的代码中嵌入许多代码，那么做这样的修改简直就是一场噩梦。

要分离逻辑和内容，有3种基本的方法，如下所示：

■用包含文件保存不同部分的内容。这种方法虽然过分单纯，但是如果网站主要是静态的，它很有效。在第5章“代码重用与函数编写”的TLA咨询公司例子中，我们已经解释了这种方法。

■用一个函数或者带有一组成员函数的类API将动态内容插入到静态网页模板中。在第6章“面向对象的PHP”中，我们已经介绍了这种方法。

■使用模板系统。模板系统可以用来解析静态模板，并通过正则表达式动态数据代替占位符标记。这样做的主要好处是，如果别人设计模板，例如图形设形师，他就根本不需要懂PHP代码。应该使提供的模板只需做最小的修改。

许多模板系统是可以从网站或其他地方获取的。最常用的可能是Smarty，可以从如下站点获得：http://smarty.php.net/。

25.10 优化代码

如果没有Web编程经验，代码的优化看起来真的很重要。当使用PHP的时候，用户等待Web应用程序的大部分时间都是来自连接和下载的次数。而优化代码对这些时间可能没有什么影响。

25.10.1 使用简单优化

可以进行一些简单的优化，使得数据库连接和下载时间有所改善。这里介绍的大多数优化都与使用PHP代码来集成数据库（如MySQL）的应用程序有关。这些优化包括：

■减少数据库连接。通常，连接数据库是所有脚本中最慢的部分。可以通过使用持久稳固的连接来解决这个问题。

■加速数据库查询。减少所做的查询数量，并且确保这些查询得到了优化。对一个复杂查询（因此比较慢）来说，通常可以使用几种方法来实现。在数据库的命令行界面运行查询，试验使用不同的方法来加速查询。在MySQL中，可以使用EXPLAIN语句来查看查询可能在什么地方绕弯路（该语句的使用已经在第12章中介绍过）。通常的原则是尽可能减少连接，增加索引。

■使PHP中生成的静态内容减少到最小。如果生成的每段HTML都来自echo或print()语句，就要花多得多的时间（这是前述的分离逻辑和内容的论点之一）。这也适用于动态生成的图像按钮——可以使用PHP来生成一次按钮之后就能在需要的时候重用它。如果每次页面载入的时候需要从函数或模板生成纯静态网页，可以考虑运行一次函数或使用一次模板，然后保存结果。

■尽可能使用字符串函数代替正则表达式。前者速度更快。

25.10.2 使用Zend产品

Zend科技公司拥有在PHP 4版本及其以后版本中使用的PHP脚本引擎（开放源代码）。除了基本引擎外，还可以下载Zend优化程序。这是一个多策略的优化程序，它可以优化代码，提高脚本运行速度从40%～100%不等。需要PHP 4.0.2或更高版本来运行这个优化程序。虽然源代码不是开放的，但是可以从Zend的站点免费下载它：http://www.zend.com。

这个插件是通过优化由脚本运行时编译产生的代码来实现的。其他Zend产品包括Zend Studio、Zend Accelerator、Zend Encoder，都需要商业性的许可协议。

25.11 测试

审查和测试代码是软件工程中另一个基本要点，然而在Web开发设计中，它常常被忽略。通常，测试者会满足于运行系统，测试两三个例子，然后说，“哦，是的，它工作得很好。”这是经常犯的一个错误。在准备将项目作为产品发布之前，要确保它已经在各种情况下经过了广泛的测试和审查。

我们推荐使用两种方法来降低代码错误率（永远不可能清除所有错误，但肯定可以排除大部分错误或将错误减到最小）。

首先，采用代码审查。也就是让另一位编程人员或另一组编程人员查看代码，并提出改进意见。通常经过这样的分析，我们能够对下面的问题提出建议：

■可能忽略的错误

■原程序设计者没有想到的测试用例

■代码优化

■安全性的提高

■可以使用已有组件改进某一段代码

■附加功能

即使一个人独立开发，我们建议也要找一个“代码查错者”（“查错者”和开发者情况差不多，可互相审查代码）。

第二个建议是让最终用户作为Web应用程序的测试人员。Web应用程序和桌面应用程序的主要区别就是每个人都将使用Web应用程序。我们不能假定每个人都能熟练使用计算机，也不可能给他们提供厚厚的手册或快速参考卡片。必须让应用程序自动存档而且自己说明自己。此外，还得考虑用户希望如何使用应用程序。可用性是极为重要的。

对于有经验的编程人员或网上冲浪者来说，很难理解没有经验的用户所遇到的问题。解决这个问题的方法之一就是请代表典型用户的人做测试员。

过去用作测试的一种方法是只基于beta版发布Web应用程序。当认为Web应用程序的主要错误已经消除时，可以将应用程序公布于小部分测试用户，并为网站提供小流量通信。可以许诺为前100位反馈网站使用意见的用户提供免费服务。我可以确信他们会带来一些开发人员原先并未想到的数据和用法及其组合。如果是在为某客户公司开发Web站点，通常，可以通过让公司的员工使用站点而提供许多没有经验的用户。（这也有内在的好处，它可以提高员工在网站中的主人翁感。）

25.12 进一步学习

这个领域有太多的内容。基本上，我们只是主要讨论了软件工程的科学，而关于软件工程，有许多图书已经介绍了。

解释了文档式网站和应用程序式网站这两种对立观点的著作是由Thomas A.Powell编写的《Web站点工程学：不仅仅是设计Web页面》（Web Site Engineering:Beyond Web Page Design）。任何自己喜欢的软件工程图书也都可留下来作为后备图书。

要了解版本控制的信息，可以访问CVS网站：http://ximbiot.com/cvs/wiki/。

关于版本控制的书并不多（但是，它却非常重要，这让人惊讶），但是可以参考由Karl Franz Fogel编著的《使用CVS进行开放源代码开发》（Open Source Development with CVS）或由Gregor N.Purdy编著的《CVS袖珍参考》（CVS Pocket Reference）。

如果要查找PHP组件、IDE或文档系统，可以试试SourceForge网站：http://sourceforge.net。

在本章中，我们介绍的许多内容都在Zend网站中的文章中讨论到了。可以访问Zend网站，查找与本章内容相关的信息。也可以考虑从该站点下载优化器：http://www.zend.com。

如果对本章内容感兴趣，还可了解极限编程（Extreme Programming），它是针对软件需求频繁改变的领域（例如，We b开发领域）而研究的软件开发方法。极限编程的网站：http://www.extremeprogramming.org。

25.13 下一章

在第26章“调试”中，我们将讨论不同类型的程序错误、PHP错误信息和查找错误的技术。

第26章　调试

在本章中，我们将主要讨论如何调试PHP脚本。完成了本书前面所介绍的一些例子或者以前曾经用过PHP，我们已经掌握了一些关于调试的技巧和技术。随着项目更加复杂，调试也变得更加困难。虽然技能提高了，但错误也可能牵涉到更多的文件，或者在多个程序员编写的代码之间相互作用。

在本章中，我们将主要介绍以下内容：

■编程语法，运行时以及逻辑错误

■出错信息

■错误级别

■触发自定义错误

■巧妙地处理错误

26.1　编程错误

无论使用什么语言进行编程，通常，都会遇到如下所示的3种错误类型：

■语法错误

■运行时错误

■逻辑错误

在开始讨论检查、处理、避免和解决错误的策略之前，我们先简要地介绍这3种类型。

26.1.1　语法错误

语言有一系列规则，称之为语法，语句只有符合语法才是有效的。自然语言（例如，英语）和编程语言（例如，PHP）都是如此。如果语句不符合这些语言规则，就被称为有语法错误。通常，语法错误在讨论解释性语言（例如，PHP）的时候，也经常称为解析错误；在讨论编译性语言（如C或Java语言）的时候也称为编译错误。

如果我们打破英语的语法规则，人们仍然很可能知道我们的意思是什么。但在编程语言中，情况就不是这样。如果脚本不遵守PHP的语法规则——也就是说，如果它含有语法错误——PHP解析器将不能处理其中的一部分或者全部。人们善于从片面或相互冲突的数据中推测出信息，但是计算机却不能。

在许多其他的规则中，PHP语法要求语句以分号结束，字符串包含在引号内，传给函数的参数用逗号分开并且包括在圆括号内。如果不遵守这些规则，PHP脚本就不可能正常工作，并在我们首次运行这些脚本的时候产生出错信息。

PHP的最大的一个优点就是在出错的时候会给出出错信息，而且其出错信息十分有用。通常，一个PHP出错信息将告诉我们出现了什么错误，在哪个文件出错，以及错误在哪一行。

如下所示的就是一个示例的出错信息：

```
Parse error:parse error,unexpected'''in
/home/book/public_html/phpmysql4e/chapter26/error.php on line 2
```

该错误由下列脚本产生：

```
<?php
$date=date(m.d.y');
?>
```

可以看到，我们试图将一个字符串传递给date()函数，但是忘了在字符串开始前添加引号。

通常，这样的简单语法错误最容易发现。但是，我们可能还会产生一个类似的却更难发现的错误——忘记了字符串结束的后引号，例如：

```
<?php
$date=date('m.d.y);
?>
```

以上代码将产生如下所示的错误：

```
Parse error:parse error,unexpected$end in
/home/book/public_html/phpmysql4e/chapter26/error.php on line 2
```

很明显，脚本代码只有3行，错误不可能真的出现在第4行。当我们打开了什么，却没有将它关闭时出现的错误可能就会像上述错误信息一样。在使用单引号和双引号，以及使用不同形式的括号时，很容易遇到这个问题。

如下所示的脚本将产生类似的语法错误：

```
<?php
if(true){
echo'error here';
?>
```

如果是多个文件的组合产生的这些错误，那么它们就很难找出来。错误出现在一个大文件里也很难查找。当遇到一个1000行的文件出现了"parse error on line 1001"的错误，恐怕要花一天才能找出错误来。作为一个提醒，我们应该尽量编写模块化的代码。

然而一般来说，语法错误是最容易找出的错误类型。如果出现语法错误并且尝试执行一个代码块，PHP会给出错信息并告诉我们在哪里查找错误。

26.1.2 运行时错误

运行时错误更难检测到，也更难修改。一个脚本可能包含语法错误，也可能没有语法错误。如果脚本包含一个语法错误，解析器能够检测到它。而运行时错误则不会只由脚本的内容导致。它们可能出现在脚本交互的过程中或者其他的事件或条件下。

如下所示的语句：

```
require('filename.php');
```

这是一个完全有效的PHP语句，它没有任何语法错误。

然而，上述语句却可能产生运行时错误。如果执行该语句时，filename.php并不存在或者运行脚本的用户对该文件没有“读”权限，就可能出现如下所示的出错信息：

```
Fatal error:main()[function.require]:Failed opening required'filename.php'
(include_path='.:/usr/local/lib/php')in
/home/book/public_html/phpmysql4e/chapter26/error.php on line 1
```

尽管代码没有任何错误，但是由于运行代码所依赖的文件在代码运行的不同时刻可能存在，也可能不存在，所以就可能产生运行时的错误。

如下所示的3个语句都是有效的PHP语句。不幸的是，当3个语句结合到一起的时候，它们是不可能得到运行结果的——因为出现了被0除的现象:

```
$i=10;
$j=0;
$k=$i/$j;
```

该代码将产生如下所示的警告：

```
Warning:Division by zero in
/home/book/public_html/phpmysql4e/chapter26/div0.php on line 3
```

这种错误很容易修改。没有人会故意编写被0除的代码，但是忽略检查用户输入却经常会导致这种错误类型。

如下所示的代码有时候将产生相同的错误，但是可能会很难隔离和修改，因为它只发生在某些时候：

```
$i=10;
$k=$i/$_REQUEST['input'];
```

这是在测试代码时最经常看到的运行时错误之一。

通常，以下操作易导致运行时的错误：

■调用不存在的函数

■读写文件

■与MySQL或其他数据库的交互

■连接到网络服务

■检查输入数据失败

在接下来的内容中，我们将简单介绍以上每一种容易导致错误的原因。

1.调用不存在的函数

在偶然的情况下，调用不存在的函数的操作比较容易发生。此外，内置函数也经常会出现命名不一致的情况。为什么strip_tags()函数名称中间有一个下划线，但是stripslashes()却没有呢？

不但在当前脚本里容易调用自己编写的、不存在的函数，而且在其他地方也容易这样调用。如果代码包含了对一个不存在函数的调用，例如：

```
nonexistent_function();
```

或

```
mispeled_function();
```

将出现如下所示的出错信息：

```
Fatal error:Call to undefined function:nonexistent_function()
in/home/book/public_html/phpmysql4e/chapter26/error.php on line 1
```

相似地，如果调用一个存在的函数，但是使用的参数个数不对，也将遇到一个警告。

函数strstr()要求输入两个字符串：haystack用来搜索，needle用来查找。如果我们按如下方式调用它：

```
strstr();
```

将得到如下所示警告：

```
Warning:Wrong parameter count for strstr()in
/home/book/public_html/phpmysql4e/chapter26/error.php on line 1
```

如下所示的语句同样是错误的：

```
<?php
```

```
if($var==4){
strstr();
}
?>
```

除了极少的情况（也就是，变量$var值为4）外，对strstr()的调用将不会发生，警告也不会出现。PHP解释器不会将时间花费在解析那些不必要在当前的执行脚本中执行的代码。我们必须确认进行了仔细的测试。

不正确的函数调用很容易发生，但是因为导致的出错信息可以确切地识别出错误行和导致错误的函数，所以它们都比较容易纠正。只有在测试过程非常糟糕并且没有测试所有条件执行的代码时，才很难查出错误。进行测试的时候，测试的目标之一是代码的每一行都要执行一次。另一个目标是测试所有的临界条件和各种类型的输入。

2.读写文件

尽管在一定程度上说，程序的使用周期里什么错误都可能出现，但是一些错误还是比其他错误更容易出现。由于访问文件错误是经常出现的，因此非常有必要巧妙地处理它们。硬盘驱动器出错或写满，人为操作错误导致目录权限改变等。

通常，与fopen()函数一样，可能偶尔失败的函数都有一个返回值指示出现的错误。对于fopen()函数来说，返回值为false表示失败。

对提供失败指示的函数，我们需要小心检查每次调用的返回值，并根据返回值进行接下来的操作。

3.与MySQL或其他数据库的交互

连接和使用MySQL可能产生许多错误。函数mysqli_connect()就可能至少产生下列一些错误：

■Warning:mysqli_connect()[function.mysqli-connect]:Can't connect to MySQL server on'localhost'(10061)（无法连接到"localhost"上的MySQL

服务器）

■ Warning:mysqli_connect()[function.mysqli-connect]:Unknown MySQL Server Host'hostname'(11001)（未知的MySQL服务器主机"hostname"）

■ Warning:mysqli_connect()[function.mysqli-connect]:Access denied for user:'username'@'localhost'(Using password:YES)

（用户username@localhost的访问被拒绝，使用密码：YES）

mysqli_connect()函数在出现错误的时候将返回值false。这就意味着用户可以很容易地检测和处理这类常见的错误。

如果希望不终止脚本的正常执行而解决这些错误，脚本将继续与数据库交互。没有一个有效的MySQL连接就试图运行查询和得到结果，将导致访问者看到满屏幕的有失专业水准的错误信息。

许多其他常用的、与MySQL相关的PHP函数，例如mysqli_query()，也将返回false来表示函数调用出现错误。

如果出现错误，可以通过调用函数mysqli_error()来获得错误信息文本，或者调用函数mysqli_errno()获得函数调用的错误代码。如果MySQL函数的调用没有出现错误，mysqli_error()将返回空字符串，mysqli_errno()将返回0。

例如，假设我们已经连接到服务器，并且选择了一个要使用的数据库，如下所示的代码段：

```
$result=mysqli_query($db,'select*from does_not_exist');
echo mysqli_errno($db);
echo'<br/>';
echo mysqli_error($db);
```

的可能输出结果是：

```
1146
Table'dbname.does_not_exist'doesn't exist
```

请注意，这些函数的输出是指最后执行的MySQL函数（而不是mysqli_error()或mysqli_errno()的结果）。如需要知道命令的结果，需确认在运行其他命令之前检查它。

就像文件交互错误一样，数据库交互造成的错误也会时常发生。即使在完成某个服务的开发和测试之后，也会偶尔发现MySQL后台程序（mysqld）崩溃或不能正常连接问题。如果服务器运行于另一台物理机器，那么它还要依赖于另一套硬件和软件组件，而这些组件可能出错——Web服务器和数据库服务器之间的网络连接、网卡、路由器等。

请记住，在试图使用这些结果之前需检查一下数据库请求是否成功。连接数据库失败之后试图检索结果是没有意义的，而检索失败后试图提取和处理检索的结果也是没有意义的。

在这里，值得注意的是，检索失败与仅仅是没有返回任何数据，也没有对任何记录行产生影响的检索是有区别的。

一个包含SQL语法错误的SQL检索，或者对不存在的数据库、表和列的访问也可能会导致错误。如下所示的查询：

```
select*from does_not_exist;
```

也可能会失败，因为表名称并不存在，这样将产生一个错误号和错误消息，可以通过mysqli_errno()函数或mysqli_error()函数获得。

通常，一个SQL查询如果只是在语法上是合法的，而且只对存在的数据库、表和列进行访问，它是不会产生错误的。但是，如果它检索一个空的数据库表或检索一个不存在的数据，它将返回空值。假设已经成功地连接了数据库，并且存在表t1和列c1，如下所示的检索：

```
select*from t1 where c1='not in database';
```

该代码将成功执行但是不返回任何结果。

在使用检索结果之前，要检查检索是否失败和返回空结果。

4.连接到网络服务

尽管系统中的设备和其他程序偶尔可能出错，但如果质量不是很差，它们应该很少出错。当在网络上连接其他机器和这些机器上的软件时，应该意识到系统的某些部分会经常出错。从一台机器连接到另一机器依赖于许多我们无法控制的设备和服务。

尽管操作可能重复，但我们还是要仔细检查试图与网络服务连接的函数的返回值。

如下所示的函数调用：

```
$sp=fsockopen('localhost'',5000);
```

如果它无法正确连接localhost机器的端口5000，该函数将给出一个警告信息。但是该警告信息将以默认的格式给出，同时不会为代码提供能够巧妙处理该警告的选择。

将以上函数调用重写为：

```
$sp=@fsockopen('localhost',5000,&$errorno,&$errorstr);
if(!$sp){
echo"ERROR:".$errorno.":".$errorstr;
}
```

以上代码将抑制内置错误消息，并且通过函数的返回值检查函数调用是否出现错误，同时使用自己的代码来处理错误消息。运行以上代码后，它将显示一个可以帮助解决问题的错误消息。在这个例子中，将产生如下所示的输出：

```
ERROR:10035:A non-blocking socket operation could not be completed immediately.
```

与语法错误相比较，运行时错误更难消除，因为在代码首次运行的时候解析器并不能指示出错误所在。因为运行时错误通常出现在一连串的事件响应中，因此很难检查到，也很难解决。解析器并不能自动指出哪一行会产生错误。测试的时候需要提供一个能够产生该类错误的情形。

解决运行时错误需要一定程度的预见性；不仅要检查所有可能出现的不同类型的错误，然后采取适当的措施；也需要仔细检测和模拟可能出现的每一类运行时错误。

这并不表示必须模拟每一个可能出现的错误。例如，MySQL可以提供大约200种不同的错误编号和错误信息，我们需要的是在每一个可能导致错误的每个函数调用中模拟函数错误，每种不同类型的错误将由特定的代码来解决。

5.检查输入数据失败

通常，我们会对用户可能输入的数据进行一些假设。如果输入的数据不是原先所预料的数据类型，它就可能会导致错误，该错误或者是运行时错误，或者是逻辑错误（将在下一节里详细介绍）。

典型的运行时错误的例子是在处理用户输入数据的时候忘记对该数据执行addslashes()函数。这样，如果有一个叫O'Grady的用户，名字中包含一个撇号，当在数据库插入语句中通过单引号使用这个输入，那么数据库函数将产生一个错误。

在接下来的一节中，我们将介绍由于对用户输入数据不同假设所导致的错误。

26.1.3 逻辑错误

逻辑错误是最难发现和清除的错误类型。这类错误类型的代码是完全正确的，而且它也是按照正确的程序逻辑执行的，但是这种工作却不是编程人员所期望的。

逻辑错误可能出现于一个简单的键盘输入错误，例如：

```
for($i=0;$i<10;$i++);
{
echo'doing something<br/>';
}
```

以上这一小段代码是完全合法的。它遵循PHP语法。也不依赖任何外部服务，因此也不可能出现运行时错误。但是如果不仔细检查，它的运行结果可能就不是我们所期望的或程序员打算要的。

粗看起来，它好像是用for语句循环10次，每次执行都回显"doing something"。在第一行for语句结尾后的分号表示循环对下面的语句行没有任何影响。for语句将循环10次，没有任何输出，然后echo语句只执行一次。

因为这段代码完全合法，但是却效率低下，这样的代码将导致无法预见的结果，解析器也不会有任何警告或出错信息。计算机擅长做某些事情，但是它没有任何常识和智能。计算机只能做我们让它所做的事情。必须确认它所做的就是我们希望的。

逻辑错误不是由任何一种代码失败所导致的，而是程序员没有正确地编写代码以命令计算机做人们所期望的事情。结果，错误就不能自动检查到。计算机也不能告诉编程者出现了错误，更不会给他一段代码告诉他问题出在什么地方。逻辑错误只有通过适当的测试才能检测到。

如上所述的微小逻辑错误我们容易犯，也容易纠正，因为在首次运行源代码看到的输出结果并不是我们所期望的输出。但是，大多数

逻辑错误的隐蔽性更强一些。

通常，麻烦的逻辑错误产生于开发人员的错误假设。在第25章“在大型项目中使用PHP和MySQL”中，我们已经建议请其他开发人员来审查代码并提出他们的测试方案，并请目标用户而不是开发人员作为测试人员进行测试。用户只输入某几种类型数据是很容易的，但是如果自己做测试容易漏掉一些没有检测到的情况。

举个例子，在某商务网站有一个Order Quantity文本框。开发人员是不是会假设用户只输入正数呢？如果一个用户输入-10，软件是否会将10倍于该物品价格的钱退还到用户信用卡里呢？

假设有一个文本输入框用来输入美元总数。它允许用户输入的总数后面带美元符号吗？允许满1000就用逗号分开吗？一些错误可以在客户端（例如，使用JavaScript）检查出来而无须由服务器来检查。

如果正在将信息传送到另一页面，可能有一些字符在一个URL中有特殊重要的意义，例如传输的字符串中包含空格。你遇到过这样的情况吗？

逻辑错误有许多。没有自动检测的方法。唯一的办法是，首先要消除暗含在脚本中的假设；其次是彻底地测试每种可能有效或无效的输入，确认这些输入都得到了期望结果。

26.2　使用变量帮助调试

当项目变得更复杂的时候，使用一些实用程序代码帮助确定错误的原因就很有意义了。程序清单26-1所示的代码就可能是很有用的。该代码回显传到页面的变量的值。

程序清单26-1　dump_variables.php——该代码可以包含在页面中，它可以将特定变量的内容打印出来，有助于调试

```
<?php
//these lines format the output as HTML comments
//and call dump_array repeatedly
echo"\n<!--BEGIN VARIABLE DUMP-->\n\n";
echo"<!--BEGIN GET VARS-->\n";
echo"<!--".dump_array($_GET)."-->\n";
echo"<!--BEGIN POST VARS-->\n";
echo"<!--".dump_array($_POST)."-->\n";
echo"<!--BEGIN SESSION VARS-->\n";
echo"<!--".dump_array($_SESSION)."-->\n";
echo"<!--BEGIN COOKIE VARS-->\n";
echo"<!--".dump_array($_COOKIE)."-->\n";
echo"\n<!--END VARIABLE DUMP-->\n";
//dump_array()takes one array as a parameter
//It iterates through that array,creating a single
//line string to represent the array as a set
function dump_array($array){
if(is_array($array)){
$size=count($array);
$string="";
if($size){
$count=0;
$string.="{";
//add each element's key and value to the string
foreach($array as$var=>$value){
```

```
$string.=$var."=".$value;
if($count++<($size-1)){
$string.=",";
}
}
$string.="}";
}
return$string;
}else{
//if it is not an array,just return it
return$array;
}
}
?>
```

该代码将页面所接收到的4个数组变量的元素内容通过循环的方式打印出来。如果某页面由GET变量、POST变量、cookie或会话变量来调用，就将打印出这些变量的内容。

在这里，我们已经将其输出加入到HTML注释，这样可以看到它们，而且不会影响浏览器对可见页面元素的显示方式。这是创建调试信息的一种好方法。就像程序清单26-1所示，在注释中隐藏调试信息将允许在最后一分钟以前仍然保留调试代码。我们使用dump_array()函数作为print_r()函数的封装器。dump_array()函数只是过滤了任何HTML的注释结束字符。

确切的输出取决于传给页面的变量，但是如果将以上代码加入到程序清单23-4中（第23章“在PHP中使用会话控制”中的一个身份验证的例子），它将下面这些行添加到脚本生成的HTML中：

```
<!--BEGIN VARIABLE DUMP-->
<!--BEGIN GET VARS-->
<!--Array
(
)
```

```
-->
<!--BEGIN POST VARS-->
<!--Array
(
[userid]=>testuser
[password]=>password
)
-->
<!--BEGIN SESSION VARS-->
<!--Array
(
)
-->
<!--BEGIN COOKIE VARS-->
<!--Array
(
[PHPSESSID]=>b2b5f56fad986dd73af33f470f3c1865
)
-->
<!--END VARIABLE DUMP-->
```

可以看到，以上脚本显示了POST变量的内容——userid和password，该变量是前一页面中的登录表单发送过来的。此外，它也显示了会话变量，该变量用来隐藏用户的名字：valid_user。正如第23章中所述，PHP用一个cookie将会话与特定的用户相关联。我们的脚本显示了伪随机码：PHPSESSID，该码将保存在cookie里，从而识别特定的用户。

26.3 错误报告级别

PHP允许设置对错误的重视程度。用户可以修改以确定哪类事件将产生错误消息。在默认情况下，PHP将报告所有除了通知之外的错误。

错误报告级别是通过一些预定义的常量来设置的，如表26-1所示。

表26-1 错误报告常量

值	名 称	意 义
1	E_ERROR	报告在运行时的致命错误
2	E_WARNING	报告运行时的非致命错误
4	E_PARSE	报告解析错误
8	E_NOTICE	报告通告、注意，表示所做的事情可能是错误的
16	E_CORE_ERROR	报告PHP引擎启动失败
32	E_CORE_WARNING	报告PHP引擎启动时非致命错误
64	E_COMPILE_ERROR	报告编译错误
128	E_COMPILE_WARNING	报告编译时出现的非致命错误
256	E_USER_ERROR	报告用户触发的错误
512	E_USER_WARNING	报告用户触发的警告
1024	E_USER_NOTICE	报告用户触发的通告
6143	E_ALL	报告所有的错误和警告（不包括E_STRICT所报告的）
2048	E_STRICT	报告不赞成的用法和不推荐的行为；不包括在E_ALL中，但对代码重构很有帮助。为交互方式提供修改意见
4096	E_RECOVERABLE_ERROR	报告所有可以捕获的非致命错误

每个常量都表示一种错误类型，该错误可以被报告也可以被忽略。例如，如果指定错误报告级别为E_ERROR，那么只有出现致命错误的时候才报告。这些常量可以用二进制算法将其结合起来，产生不同的错误级别。

默认的错误级别是报告除了通知之外的所有错误，由如下所示语句指定：

```
E_ALL & ~E_NOTICE
```

以上表达式由前面说明的两个常量组成，它们通过位算法操作符“&”结合。“&”操作符表示位操作符AND，而符号“~”表示操作符NOT。上述表达式可读作E_ALL AND NOT E_NOTICE。

E_ALL本身就是除E_STRICT以外的所有错误类型的有效结合。它可以用其他错误级别通过位操作符“|”相“或”得到，例如：

```
E_ERROR|E_WARNING|E_PARSE|E_NOTICE|E_CORE_ERROR|E_CORE_WARNING|
E_COMPILE_ERROR|E_COMPILE_WARNING|E_USER_ERROR|E_USER_WARNING|
E_USER_NOTICE
```

类似地，默认的错误报告级别也可通过除了"E_NOTICE"之外的所有错误级别相“或”而得到：

```
E_ERROR|E_WARNING|E_PARSE|E_CORE_ERROR|E_CORE_WARNING|E_COMPILE_ERROR|
E_COMPILE_WARNING|E_USER_ERROR|E_USER_WARNING|E_USER_NOTICE
```

26.4　改变错误报告设置

可以通过php.ini文件或在每个脚本中进行错误报告设置。

在默认情况下，要改变所有脚本的错误报告，可以修改php.ini文件里的下面4行。

```
error_reporting=E_ALL＆～E_NOTICE
display_errors=On
log_errors=Off
track_errors=Off
```

默认的全局设置如下：

■报告除了通知之外的所有错误

■以HTML形式将出错信息输出到标准输出接口

■不将错误信息作为日志记录到磁盘

■不跟踪错误，将错误保存在变量$php_errormsg中

最可能要改变的是将错误报告级别提高到E_ALL|E_STRICT。这将导致报告脚本中可能出现的所有通知，这些通知可能指示一个错误，也可能是因为程序员想利用PHP弱类型的特性，还可能因为自动将变量初始化为0这样的事实。

在调试过程中，我们会发现提高error_reporting的级别是有意义的。在产品代码中，如果为自己提供了有用的错误信息，在将error_reporting的级别提高的同时，需将display_errors选项关闭，并将log_errors选项开启，这样会使代码看起来更加专业一些。如果出现错误，可以在日志里看到错误的详细信息。

将track_errors选项打开可以帮助解决用户自己代码中的错误，而不是让PHP提供其默认的功能。尽管PHP提供了实用的出错信息，但

当出错的时候，它的默认行为却非常糟糕。

在默认情况下，当脚本出现一个致命错误的时候，PHP将输出如下所示的信息：

```
<br>
<b>Error Type</b>:error message in<b>path/file.php</b>
on line<b>lineNumber</b><br>
```

并且停止执行脚本。对于非致命错误，也会输出同样的信息，但是脚本正常执行。

该HTML的输出将直接发送给错误的标准输出，但是看起来很乏味。这样的出错信息不可能与网站其他部分的外观相协调。如果页面的整个内容显示在一个表格里的话，它也可能导致Netscape用户根本看不到输出。这是因为打开了的HTML没有关闭表格元素，例如：

```
<table>
<tr><td>
<br>
<b>Error Type</b>:error message in<b>path/file.php</b>
on line<b>lineNumber</b><br>
```

以上HTML代码将在某些浏览器中显示为一个空屏。

我们没有必要保留PHP的默认错误处理行为，甚至对所有的文件使用同样的设置。要改变当前脚本的错误报告级别，可以调用error_reporting()函数。

传递一个错误报告常量或几个错误报告常量的结合，与php.ini中类似的指令一样，可以用来设置错误级别。error_reporting()函数将返回前一个错误级别。该函数的常见使用方式如下所示：

```
//turn off error reporting
```

```
$old_level=error_reporting(0);
//here,put code that will generate warnings
//turn error reporting back on
error_reporting($old_level);
```

这小段代码将关闭错误报告，允许我们执行一些可能产生警告的代码，而我们并不希望看到这些警告。

永久关闭出错报告是非常糟糕的想法，因为这样很难发现错误并纠正它。

26.5 触发自定义错误

函数trigger_error()可以用来触发错误。用这种方式创建的错误可以像PHP的常规错误那样处理它。

该函数要求给出出错信息，此外，错误类型是该函数的可选参数。错误类型应该是E_USER_ERROR、E_USER_WARNING或E_USER_NOTICE中的一种。如果没有指定这些类型，默认值是E_USER_NOTICE。

如下所示的例子说明了如何使用trigger_error()函数：

```
trigger_error('This computer will self destruct in 15 seconds',E_USER_WARNING);
```

26.6 巧妙地处理错误

对于有C++或Java编程经历的人来说，可能会很习惯使用异常处理。异常处理允许函数发出信号通知错误出现，并将错误处理交给一个专门的异常处理程序。异常机制是在大型项目中处理错误的好方法。关于异常处理，我们已经在第7章“错误和异常处理”中详细介绍，这里就不再重复。

你已经了解如何触发自己的错误。你也可以提供错误处理程序来捕获错误。

当用户级别的错误、警告和通知发生时，set_error_handler()函数可以提供一个可供调用的函数，可以将用作错误处理程序的函数名称作为set_error_handler()的参数。

错误处理函数必须带有两个参数：错误类型和错误信息。基于这两个参数变量，函数可以决定如何解决这个错误。其中，错误类型应该是已经定义好的错误类型常量。错误信息则是一个描述性字符串。

如下所示的是set_error_handler()函数的调用：

```
set_error_handler('my_error_handler');
```

以上调用将告诉PHP调用一个名为my_error_handler()的函数来进行错误处理操作，此外，还要提供该函数的具体实现方法。该函数原型如下所示：

```
My_error_handler(int error_type,string error_msg
[,string errfile[,int errline[,array errcontext]]]))
```

但是，该函数的具体实现可以由自己来设定。

传递给处理程序函数的参数如下：

■错误类型

■错误消息

■发生错误的文件

■发生错误的行号

■符号表——也就是，发生错误时所有变量及其值的集合

该函数的逻辑行为可能包括：

■显示系统提供的错误信息

■将信息存储到日志文件

■将错误以电子邮件形式发送到一个邮箱

■调用一个exit，中断脚本执行

程序清单26-2所示的脚本声明了一个错误处理程序，并调用函数set_error_handler()设置它，然后生成了一些错误。

程序清单26-2　handle.php——该脚本声明和定制了一个错误处理程序并生成不同的错误

```
<?php
//The error handler function
function myErrorHandler($errno,$errstr,$errfile,$errline){
echo"<br/><table bgcolor=\"#cccccc\"><tr><td>
<p><strong>ERROR:</strong>".$errstr."</p>
<p>Please try again,or contact us and tell us
that the error occurred in line".$errline."of
file".$errfile."</p>";
if(($errno==E_USER_ERROR)||($errno==E_ERROR)){
echo"<p>This error was fatal,program ending</p>
</td></tr></table>";
```

```
//close open resources,include page footer,etc
exit;
}
echo"</td></tr></table>";
}
//Set the error handler
set_error_handler('myErrorHandler');
//trigger different levels of error
trigger_error('Trigger function called',E_USER_NOTICE);
fopen('nofile','r');
trigger_error('This computer is beige',E_USER_WARNING);
include('nofile');
trigger_error('This computer will self destruct in 15 seconds',E_USER_ERROR);
?>
```

以上脚本的输出结果如图26-1所示。

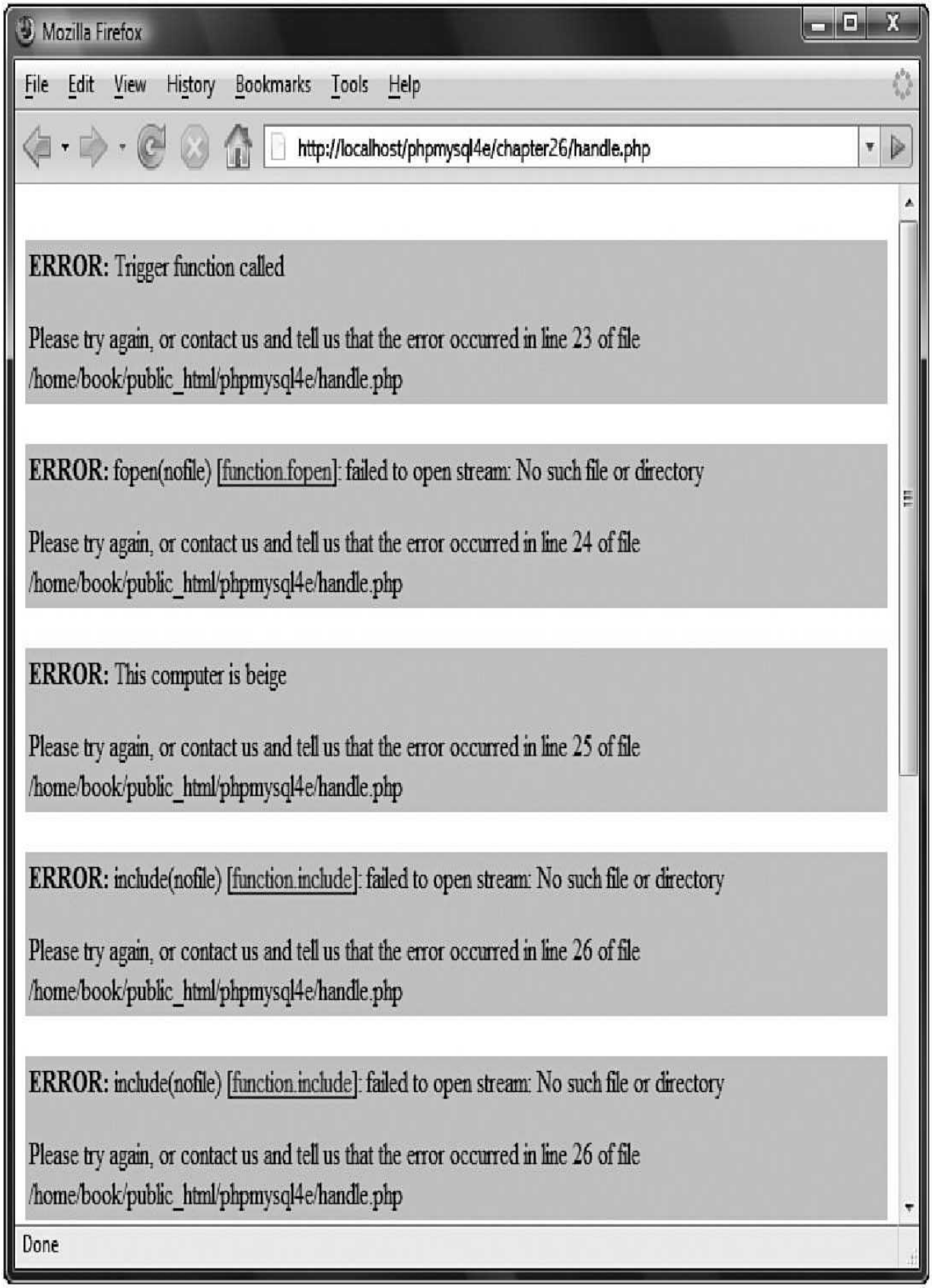

Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/phpmysql4e/chapter26/handle.php
ERROR: Trigger function called
Please try again, or contact us and tell us that the error occurred in line 23 of file /home/book/public_html/phpmysql4e/handle.php
ERROR: fopen(nofile) [function.fopen]: failed to open stream: No such file or directory
Please try again, or contact us and tell us that the error occurred in line 24 of file /home/book/public_html/phpmysql4e/handle.php
ERROR: This computer is beige
Please try again, or contact us and tell us that the error occurred in line 25 of file /home/book/public_html/phpmysql4e/handle.php
ERROR: include(nofile) [function.include]: failed to open stream: No such file or directory
Please try again, or contact us and tell us that the error occurred in line 26 of file /home/book/public_html/phpmysql4e/handle.php
ERROR: include(nofile) [function.include]: failed to open stream: No such file or directory
Please try again, or contact us and tell us that the error occurred in line 26 of file /home/book/public_html/phpmysql4e/handle.php
Done

图　26-1　如果使用自己的错误处理程序，可以设置比PHP更友好的错误信息

以上自定义的错误处理程序的功能并不比默认的错误处理程序强很多。因为这段代码是由我们自己实现的，所以可以让它做任何事情。通过它可以在出错的时候选择性地告诉访问者一些有用的信息，以及如何显示该信息，这样就可以与网站其他部分的内容相协调。更重要的是，我们可以灵活地决定发生什么事情：脚本是否应该继续执行？信息是否应该记录到日志或将它显示出来吗？要对技术支持发出警告吗？

需要注意的一点是，自定义错误处理函数将无法处理所有的错误类型。有些错误，例如，解析错误和致命的运行时错误，还是会触发默认的错误处理函数。如果需要考虑到这种情况，需确认在将参数传递给能够产生致命错误，或触发自定义的E_USER_ERROR级别错误之前，仔细检查这些参数。

这里再介绍一个有用的新特征：如果错误处理程序返回了false值，PHP的内置错误处理程序将被调用。这样，可以处理E_USER_*错误，并由内置处理程序处理常见错误。

26.7 下一章

在第27章“建立用户身份验证机制和个性化设置”中，我们将开始本书的第一个项目。在这个项目中，我们将讨论如何识别再次访问网站的用户，以及如何对网站内容进行适当的调整。

第27章　建立用户身份验证机制和个性化设置

在这个项目中，我们将讨论如何让用户在我们的站点注册。当完成注册之后，我们能跟踪他们所感兴趣的事物并向他们显示适当的内容。这个过程叫做用户个性化设置。

这个项目允许用户建立一组网页书签，并根据他们以前的操作向他们建议其他可能感兴趣的链接。更常见的是，用户个性化设置可用于所有基于Web的应用程序，以用户希望的格式显示他们感兴趣的内容。

在这个项目和接下来的其他项目中，我们将开始了解一组需求，这个需求类似于从客户那里获得的需求。我们会将这些需求开发为一组解决方案组件，并建立一个将这些组件联系到一起的设计，然后再应用每个组件。

在这个项目中，我们将实现如下功能：

■用户登录和验证用户

■管理密码

■记录用户的个人喜好

■个性化内容

■基于已有的用户信息为用户推荐他们可能感兴趣的内容

27.1　解决方案的组成

对于这个项目，我们希望为在线书签系统建立一个原型，称之为PHPbookmark，它与在Backflip网站上可以访问到的系统类似（但在功能上更有限）：http://backflip.com。

我们的系统应该能够让用户登录进来，保存他们的书签，并基于他们的个人喜好推荐给他们可能喜欢的其他站点。

这些解决方案需求分为3个主要部分。

首先，需要识别每个用户。我们应该有验证他们身份的方法。

其次，需要保存单个用户的书签。用户应该能够添加和删除书签。

再次，需要根据对他们的了解，向用户建议他们可能感兴趣的站点。

现在，我们已经对项目有了基本了解，可以开始设计解决方案及其组件了。下面，我们开始了解上述3个需求各自的解决方案。

27.1.1 用户识别和个性化设置

用户身份验证有许多可以选择的办法，这些在本书的其他部分已经讨论过。因为我们希望将用户和一些个性化信息联系起来，所以就要将用户的登录名和密码保存在一个MySQL数据库中，验证的时候用它做比较。

如果要让用户以用户名和密码登录，需要如下组件。

■用户能够注册一个用户名和密码。我们需要限制用户名和密码的长度和格式。为安全起见，应该将密码加密之后再保存。

■在注册过程中，用户应该能够看到他们提供的详细信息。

■用户完成网站访问之后能够登出。从隐私角度考虑看，如果用户是使用自己家中的PC，这并不是很重要，但是在共用的PC上这就相当重要了。

■网站能够检测用户是否登录，并为登录的用户访问数据。

■为了安全起见，用户可以修改密码。

■用户应该能够在不需要开发者帮助的前提下重置他们的密码。一个常用的方法是将密码发送到用户注册时提供的邮箱。这意味着要在注册的时候保存他们的邮箱地址。因为密码是以加密的形式进行保

存的，而且其他人不能够解码。因此实际上只能为用户设置一个新密码，并将它发给用户。

我们要为所有这些功能编写函数。这些函数大多数是可重用的，或者可以从其他项目中拿过来，只要经过较小的修改就可以重用。

27.1.2 保存书签

要保存一个用户的书签，需要在MySQL数据库中开辟一定的空间。我们必须实现如下功能：

■用户应该能够取回、浏览他们的书签

■用户应该能够增添加新的书签，我们要检查这些URL是否有效

■用户应该能够删除书签

我们要为以上每个功能编写函数。

27.1.3 推荐书签

我们可以采取不同的方式向用户推荐书签，可以推荐最流行的或者在某个主题上最流行的站点。对于这个项目，我们要实现一个“相似意向”建议系统，该系统可以查找与登录用户具有相同书签的用户，并将他们的其他书签推荐给用户。为了避免推荐任何个人的书签，我们只推荐几个用户同时拥有的书签。

我们同样需要为这个功能编写一个函数。

27.2　解决方案概述

在纸上绘制草图之后，我们提出了系统流程图，如图27-1所示。

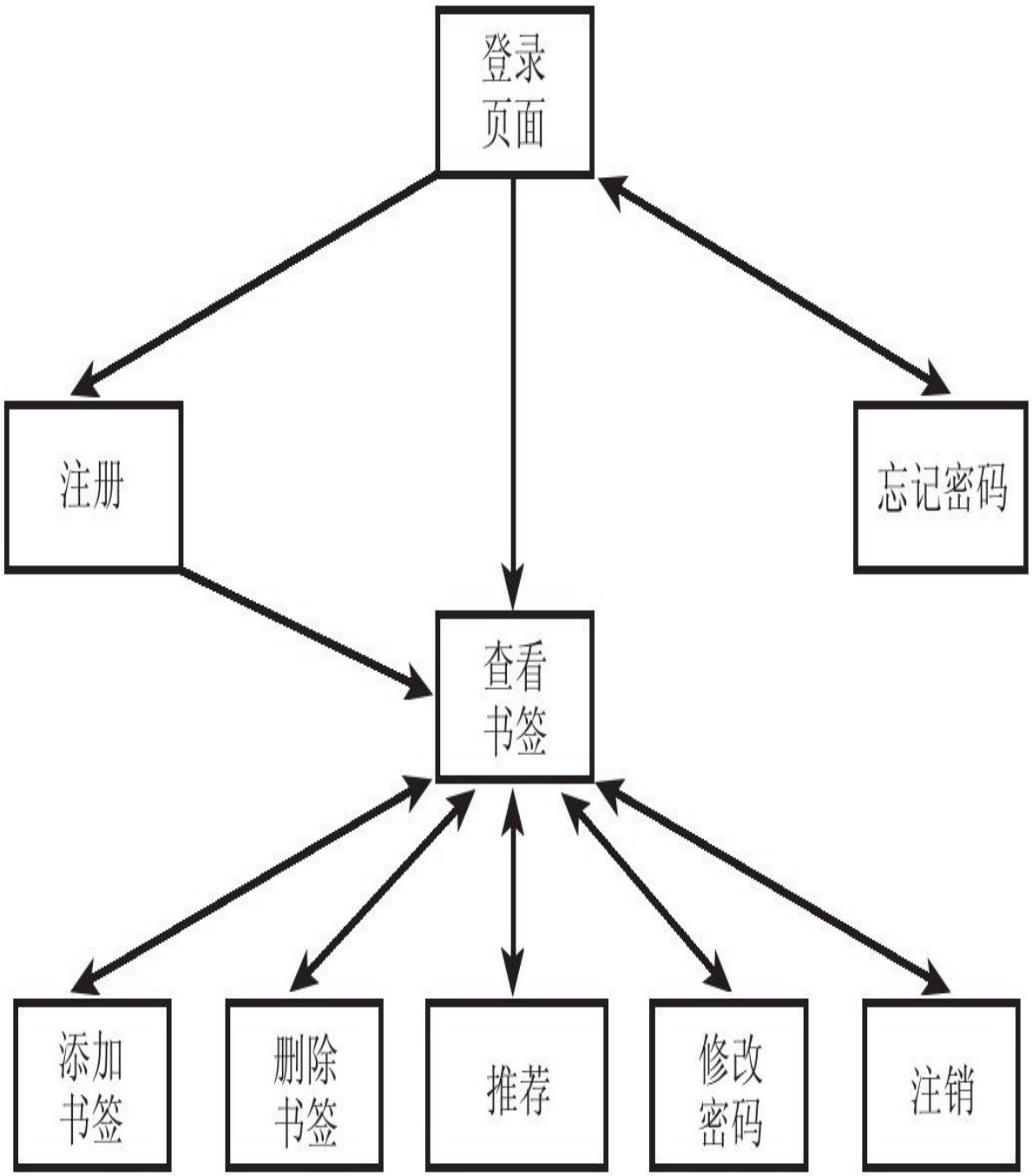


图　27-1　该图显示了PHPbookmark系统的各种工作流程

我们将为该表中每一部分创建一个模块；其中一些模块需要一段脚本，而另外一些可能需要两脚本段。我们还要建立函数库，函数库作用如下：

■用户身份验证

■书签保存与检索

■数据验证

■数据库连接

■输出到浏览器

我们将把所有HTML输出都限制到该函数库，确保在网站里所有可视的外观是和谐一致的（这就是函数API方法用来将逻辑和内容分离的功能）。

我们还需要建立一个系统后台数据库。

我们只介绍这个解决方法的部分详细内容，该应用程序的代码都可以在随书附带文件的chapter27目录下找到。该目录下所包含文件的摘要如表27-1所示。

表27-1 PHPbookmark应用程序中的文件

文件名	描述
bookmarks.sql	创建PHPbookmark数据库的SQL语句
login.php	包含系统登录表单的页面
register_form.php	系统中用户注册表单
register_new.php	处理新注册信息的脚本
forgot_form.php	用户忘记密码后需要填写的表单
forgot_passwd.php	重新设置遗忘密码的脚本
member.php	用户的主页面，包含该用户所有的当前书签
add_bm_form.php	添加书签的表单
add_bms.php	将书签真正添加到数据库中的脚本
delete_bms.php	从用户的书签列表中删除选定书签的脚本
recommend.php	基于用户以前的操作，推荐用户可能感兴趣的书签
change_passwd_form.php	用户修改密码时要填的表单
change_passwd.php	修改数据库中用户密码的表单
logout.php	将用户注销的脚本
bookmark_fns.php	应用程序的包含文件集合
data_valid_fns.php	确认用户输入数据有效的函数
db_fns.php	连接数据库的函数
user_auth_fns.php	用户身份验证的函数
url_fns.php	增加和删除书签的函数
output_fns.php	以HTML形式格式化输出的函数
bookmark.gif	PHPbookmark的logo图标

首先，我们将讨论应用程序中MySQL数据库的实现，因为它实际上是实现其他功能所必需的。

然后，我们将以编写代码的顺序详细研究代码。从首页开始，到用户验证，到书签保存和检索，最后书签建议。这个顺序是很有逻辑性的——正好是一个解决依赖性的问题，也就是依次创建下一个模块所需要的模块。

提示　要该应用能够正常工作，需要一个支持JavaScript的浏览器来查看应用程序。

27.3 实现数据库

对于PHPbookmark数据库来说，只需要一个非常简单的数据库模式。在程序中，我们要保存用户名和他们的邮箱地址以及用户密码，还要保存书签的URL。一个用户可能有许多书签，许多用户也可能注册了同一个书签。因此，我们需要两个表，user表和bookmark表，如图27-2所示。

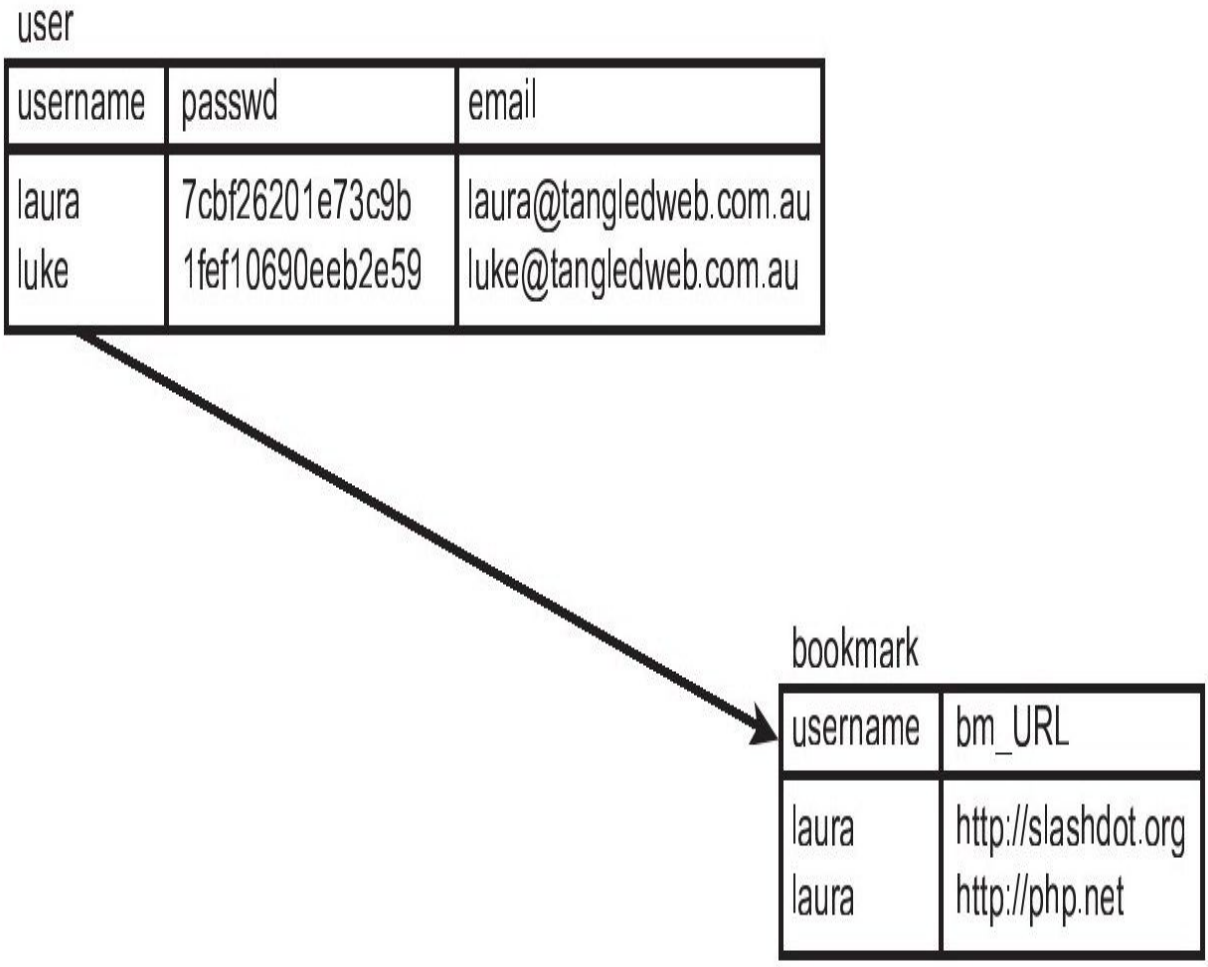


图 27-2 PHPbookmark系统的数据库模式

user表用来保存用户的username（该表单的主键）、password和email。bookmark表用来保存username和bm_URL对。表中的username指向user表格中的username。

创建本数据库的SQL脚本，以及创建一个用户用来通过网络连接到数据库的脚本，如程序清单27-1所示。注意如果要将该代码应用于自己的系统，需要进行一定的修改：改变用户密码，使其更安全！

程序清单27-1　bookmarks.sql——建立Bookmark数据库的SQL文件

```
create database bookmarks;
use bookmarks;
create table user(
username varchar(16)not null primary key,
passwd char(40)not null,
email varchar(100)not null
);
create table bookmark(
username varchar(16)not null,
bm_URL varchar(255)not null,
index(username),
index(bm_URL),
primary key(username,bm_URL)
);
grant select,insert,update,delete
on bookmarks.*
to bm_user@localhost identified by'password';
```

可以以MySQL root用户的身份运行以上这些命令在系统中建立数据库。可以在系统的命令行下输入如下所示命令：

```
mysql-u root-p＜bookmarks.sql
```

系统将要求输入root用户的密码。

数据库建好之后，我们可以继续网站建设的其他基本步骤。

27.4　实现基本的网站

我们需要创建的第一页称为login.php，因为它向用户提供了登录系统的机会。该页面的代码如程序清单27-2所示。

程序清单27-2　login.php——PHPbookmark系统的首页

```
<?php
require_once('bookmark_fns.php');
do_html_header('');
display_site_info();
display_login_form();
do_html_footer();
?>
```

这段代码看起来非常简单，它大部分是调用我们将要为这个应用程序所创建的函数API。

稍后，我们就可看到这些函数的详细信息。从这些文件我们可以看出已经包含了一个文件（也就是包含了一些函数），调用一些函数绘制一个HTML标题，显示一些内容，然后再绘制一个HTML页脚。

以上脚本的输出如图27-3所示。

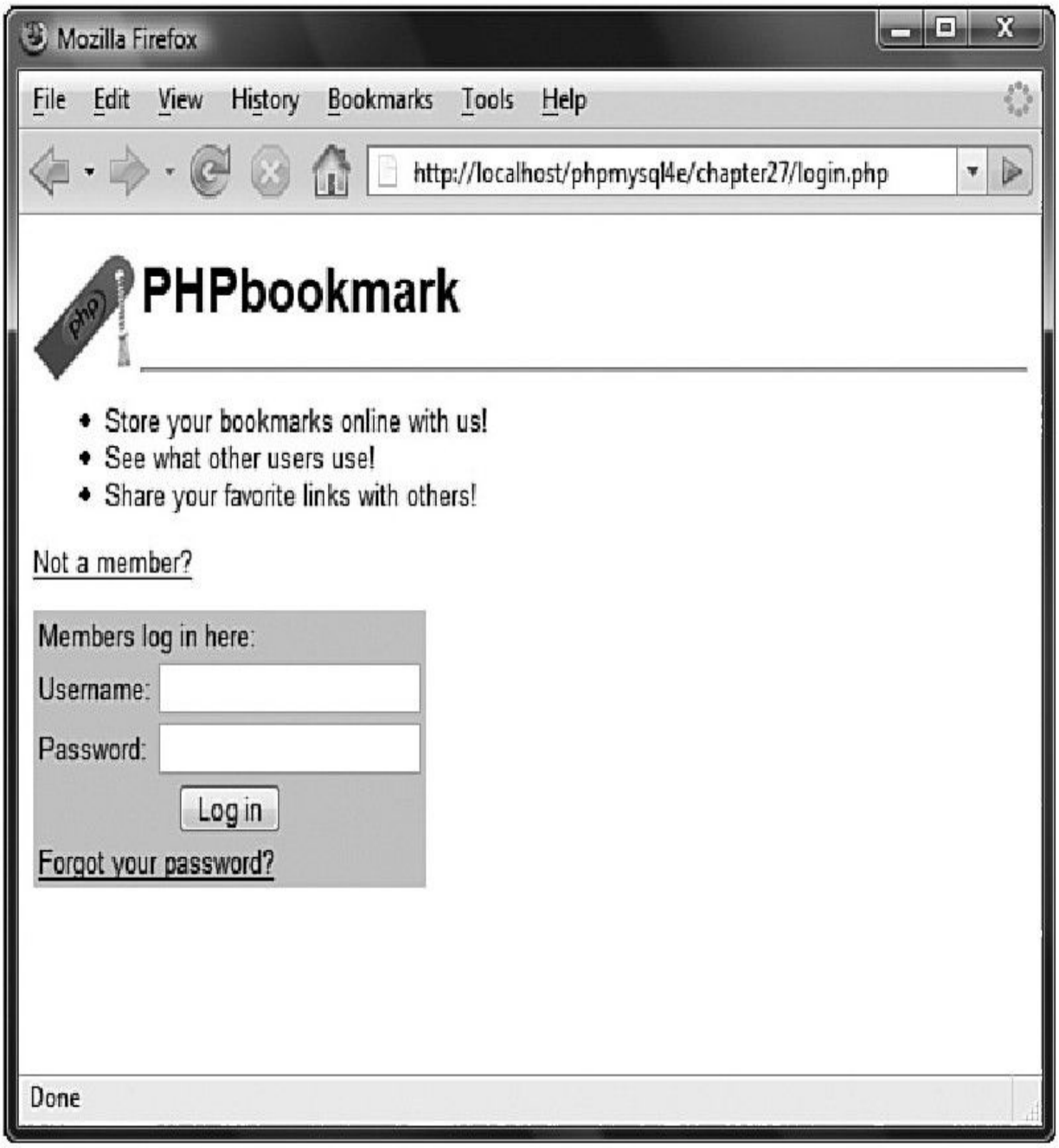


图　27-3　PHPbookmark系统的首页由login.php中的绘制HTML的函数创建

系统中的函数都包含在bookmark_fns.php文件中，如程序清单27-3所示。

程序清单27-3　bookmark_fns.php——包含Bookmark应用程序函数的文件

```
<?php
//We can include this file in all our files
//this way,every file will contain all our functions and exceptions
require_once('data_valid_fns.php');
require_once('db_fns.php');
require_once('user_auth_fns.php');
require_once('output_fns.php');
require_once('url_fns.php');
?>
```

可以看到，该文件就是本应用程序中将要使用到的5个其他包含文件的“容器”。设计成这样的结构是因为函数在逻辑上可以分为几组。这些组中的一些函数可能对其他项目有用处，因此我们将每个函数组保存不同的文件，这样就可以在再需要它们的时候知道到哪里去查找。创建bookmark_fns.php文件是因为在大部分脚本里都要用到这5个函数文件。在每个脚本里包含这一个文件而不是使用5个require语句，这样会更容易一些。

在这个例子中，我们使用的函数来自output_fns.php文件。这些函数相当直观，它们的输出都是非常明了的HTML。该文件包含了我们在login.php中使用的4个函数，即do_html_header()、display_site_info()、display_login_form()和do_html_footer()等。

本书不再深入讨论所有这些函数，只举例说明其中一个函数。do_html_header()函数源代码如程序清单27-4所示。

程序清单27-4　output_fns.php文件中的函数do_html_header()——

该函数输出在本应用程序的每个页面中都将出现的标准标题

```
function do_html_header($title){
//print an HTML header
?>
<html>
<head>
```

```
<title><?php echo$title;?></title>
<style>
body{font-family:Arial,Helvetica,sans-serif;font-size:13px}
li,td{font-family:Arial,Helvetica,sans-serif;font-size:13px}
hr{color:#3333cc;width=300px;text-align:left}
a{color:#000000}
</style>
</head>
<body>
<img src="bookmark.gif"alt="PHPbookmark logo"border="0"
align="left"valign="bottom"height="55"width="57"/>
<h1>PHPbookmark</h1>
<hr/>
<?php
if($title){
do_html_heading($title);
}
}
```

可以看到，函数中唯一的逻辑是在页面中添加适当的标题。login.php中使用的其他函数与该函数类似。display_site_info()函数添加一些关于网站的文本；display_login_form()显示如图27-3所示的灰色表单；do_html_footer()为页面添加一个标准的HTML页脚。

关于从主逻辑流中分离或删除HTML的意义，我们已经在第25章“在大型项目中使用PHP和MySQL”详细讨论过。在本章中，我们将介绍使用函数API方法。

在图27-3中，可以看到该页面有3个选择——用户可以注册、登录（如果已经注册）和修改密码（如果忘记了密码）。要实现这些模块，必须先了解下一节的用户身份验证。

27.5 实现用户身份验证

用户身份验证模块包括4个主要的元素：用户注册、登录和登出、修改密码以及重置密码。我们将按顺序详细讨论每个元素。

27.5.1 注册用户

要注册一个用户，需要通过一个表单获得用户的详细信息，并且将这些信息保存到数据库中。

当用户点击login.php页面上的"Not a member?"链接时，就会出现一个由register_form.php产生的注册表单。该脚本如程序清单27-5所示。

程序清单27-5 register_form.php——该表单让用户在PHPbookmark系统中注册

```
<?php
require_once('bookmark_fns.php');
do_html_header('User Registration');
display_registration_form();
do_html_footer();
?>
```

可以看到，该页非常简单，只调用了来自output_fns.php的输出库中的函数。该脚本输出如图27-4所示。



图　27-4　注册表单获取了数据库需要的用户详细信息。密码要求输入两次，以防输入错误

该页中的灰色表单是由display_registration_form()函数输出的，该函数也包含在output_fns.php中。当用户点击"Register"按钮时，register_new.php脚本将运行。该脚本如程序清单27-6所示。

程序清单27-6　register_new.php——该脚本验证新用户的数据，并将其存入数据库

```
<?php
//include function files for this application
require_once('bookmark_fns.php');
//create short variable names
$email=$_POST['email'];
$username=$_POST['username'];
$passwd=$_POST['passwd'];
$passwd2=$_POST['passwd2'];
//start session which may be needed later
//start it now because it must go before headers
session_start();
try{
//check forms filled in
if(!filled_out($_POST)){
throw new Exception('You have not filled the form out correctly-
please go back and try again.');
}
//email address not valid
if(!valid_email($email)){
throw new Exception('That is not a valid email address.
Please go back and try again.');
}
//passwords not the same
if($passwd!=$passwd2){
throw new Exception('The passwords you entered do not match-
please go back and try again.');
}
//check password length is ok
//ok if username truncates,but passwords will get
```

```
//munged if they are too long.
if((strlen($passwd)<6)||(strlen($passwd)>16)){
throw new Exception('Your password must be between 6 and 16 characters.
Please go back and try again.');
}
//attempt to register
//this function can also throw an exception
register($username,$email,$passwd);
//register session variable
$_SESSION['valid_user']=$username;
//provide link to members page
do_html_header('Registration successful');
echo'Your registration was successful.Go to the members page to start
setting up your bookmarks!';
do_html_url('member.php','Go to members page');
//end page
do_html_footer();
}
catch(Exception$e){
do_html_header('Problem:');
echo$e->getMessage();
do_html_footer();
exit;
}
?>
```

这是该项目中我们看到的第一个比较复杂的脚本。该脚本的起始部分包含了应用程序函数文件并启动了一个会话（用户注册的时候，我们将他的用户名创建为会话变量，正如我们在第23章中“在PHP中使用会话控制”所介绍的那样）。

脚本的主体有一个try语句块，因为需要检查许多条件。如果任何一个条件失败，执行将进入catch语句块，我们将在稍后详细介绍。

接下来验证用户输入的数据。在此过程中，我们要测试许多条件，如下所示。

■检查表单是否完全填写。调用filled_out()函数测试，如下所示：

```
if(!filled_out($_POST))
```

这个函数是我们自己编写的函数之一。它位于data_valid_fns.php函数库。稍后，我们将详细介绍这个函数。

■检查邮件地址是否有效。测试如下所示：

```
if(valid_email($email))
```

这个函数也是我们自己编写的函数之一。它也位于data_valid_fns.php函数库。

■验证用户两次输入的密码是否一致，如下所示：

```
if($passwd!=$passwd2)
```

■验证密码长度是否在规定范围之内，如下所示：

```
if((strlen($passwd)<6)
```

和

```
if((strlen($passwd)>16)
```

在我们的例子中，密码至少为6个字符，这样以防别人容易猜出，同时用户名要少于17个字符，以适合存放到数据库中。请注意，密码的最大长度并不局限于此，因为它是以SHA1哈希值保存的，因此通常是40个字符，而不受密码长度的限制。

本例用到的数据验证函数filled_out()和valid_email()，分别如程序清单27-7和程序清单27-8所示。

程序清单27-7　data_valid_fns.php文件中的filled_out()函数——该函数检查表单是否完全填写

```
function filled_out($form_vars){
//test that each variable has a value
foreach($form_vars as$key=>$value){
if((!isset($key))||($value==='')){
return false;
}
}
return true;
}
```

程序清单27-8　data_valid_fns.php文件中的valid_email()函数——该函数检查邮件地址是否有效

```
function valid_email($address){
//check an email address is possibly valid
if(ereg('^[a-zA-Z0-9_\.\-]+@[a-zA-Z0-9\-]+\.[a-zA-Z0-9\-\.]+$',$address)){
return true;
}else{
return false;
}
}
```

函数filled_out()需要传递一个数组变量，通常是$_POST或$_GET变量数组。它检查表单是否完全填写，如果完全填写则返回true，否则返回false。

valid_email()函数使用了在第4章“字符串操作与正则表达式”中介绍的正则表达式来验证邮件地址。如果地址是有效的，就返回true，否则返回false。

在验证了输入数据之后，我们就可以尝试注册该用户了。如果回头看看程序清单27-6，会发现我们是按如下方式实现的：

```
register($username,$email,$passwd);
//register session variable
$_SESSION['valid_user']=$username;
//provide link to members page
do_html_header('Registration successful');
echo'Your registration was successful.Go to the members page to start
setting up your bookmarks!';
do_html_url('member.php','Go to members page');
//end page
do_html_footer();
```

可以看到，我们使用用户输入的用户名、邮件地址和密码作为参数调用了register()函数。如果函数执行成功，我们就将用户名注册为会话变量，并为用户提供一个指向成员主页的链接（如果函数执行失败，它将抛出一个可以在catch语句块中捕获的异常）。其输出如图27-5所示。



图　27-5　注册成功，用户可以访问成员页

register()函数包含在user_auth_fns.php函数库中。函数代码如程序清单27-9所示。

程序清单27-9　user_auth_fns.php文件中的register()函数——该函数试图将用户信息提交到数据库

```
function register($username,$email,$password){
//register new person with db
//return true or error message
//connect to db
$conn=db_connect();
//check if username is unique
$result=$conn->query("select*from user where username='".$username."'");
if(!$result){
throw new Exception('Could not execute query');
}
if($result->num_rows>0){
throw new Exception('That username is taken-go back and choose another
one.');
}
//if ok,put in db
$result=$conn->query("insert into user values
('".$username."',sha1('".$password."'),'".$email."')");
if(!$result){
throw new Exception('Could not register you in database-please try again
later.');
}
return true;
}
```

在这个函数中，也没有特别新的内容，只是将它连接到前面已经建立的数据库中。如果选定的用户名已经存在，或者数据库不能被更新，它将抛出一个异常。否则，它将更新数据库并返回true。

需要注意的是，我们使用了自己编写的函数db_connect()来执行数据库连接操作。该函数只提供了一个连接到数据库的地址，而该地址同时还保存着用户名和密码。这样，如果要修改数据库中的密码，只

需改变应用程序中的一个文件即可。db_connect()函数如程序清单27-10所示。

程序清单27-10　db_fns.php文件中的db_connect()函数——该函数连接MySQL数据库

```
<?php
function db_connect(){
$result=new mysqli('localhost','bm_user','password','bookmarks');
if(!$result){
throw new Exception('Could not connect to database server');
}else{
return$result;
}
}
?>
```

在用户注册之后，可以通过正规的登录或登出页面登录和退出网站。接下来，我们就将实现它。

27.5.2 登录

如果用户将他们的信息输入到login.php（如图27-3所示）的表单中，并提交给系统，系统将运行member.php脚本。如果用户信息正确，该脚本将允许用户登录，同时显示与登录用户所有相关的书签。这是本应用程序剩余部分的核心。该脚本如程序清单27-11所示。

程序清单27-11　member.php——该脚本是本应用程序的主体部分

```
<?php
//include function files for this application
require_once('bookmark_fns.php');
session_start();
//create short variable names
$username=$_POST['username'];
$passwd=$_POST['passwd'];
if($username&&$passwd){
//they have just tried logging in
try{
login($username,$passwd);
//if they are in the database register the user id
$_SESSION['valid_user']=$username;
}
catch(Exception$e){
//unsuccessful login
do_html_header('Problem:');
echo'You could not be logged in.
You must be logged in to view this page.';
do_html_url('login.php','Login');
do_html_footer();
exit;
}
}
do_html_header('Home');
check_valid_user();
```

```
//get the bookmarks this user has saved
if($url_array=get_user_urls($_SESSION['valid_user'])){
display_user_urls($url_array);
}
//give menu of options
display_user_menu();
do_html_footer();
?>
```

可以看出脚本中的逻辑：我们在脚本中重用了第23章中的一些思想。

首先，检查用户是否是从前面的页面链接过来的——也就是说，他是否填了登录表单——然后尝试将其登录，如下所示：

```
if($username&&$passwd){
//they have just tried logging in
try{
login($username,$passwd);
//if they are in the database register the user id
$_SESSION['valid_user']=$username;
}
```

可以看出，我们试图通过login()函数调用将用户登录进来。我们已经在user_auth_fns.php库中定义了这个函数。稍后，我们将详细介绍其源代码。

如果用户登录成功，我们就将注册他的会话，正如前面所做的一样。并将用户名保存到会话变量valid_user中。

如果一切顺利，为该用户显示成员页：

```
do_html_header('Home');
check_valid_user();
//get the bookmarks this user has saved
```

```
if($url_array=get_user_urls($_SESSION['valid_user'])){
display_user_urls($url_array);
}
//give menu of options
display_user_menu();
do_html_footer();
```

该页面也是通过输出函数创建的。注意在这里我们使用了几个新函数。它们分别是user_auth_fns.php文件中的check_valid_user()函数、url_fns.php文件中的get_user_urls()函数，以及来自output_fns.php文件的display_user_urls()函数。check_valid_user()函数将检查当前用户是否拥有一个注册的会话。这是针对还没有登录却处于会话当中的用户。get_user_urls()函数将从数据库中获得用户的书签，而display_user_urls()函数将以表格的形式在浏览器中输出用户的书签。稍后，我们将详细介绍check_valid_user()函数，而其他两个函数将在书签存储和检索一节中介绍。

member.php脚本通过调用display_user_menu()函数显示一个菜单来结束本页面。图27-6显示了member.php的输出结果。

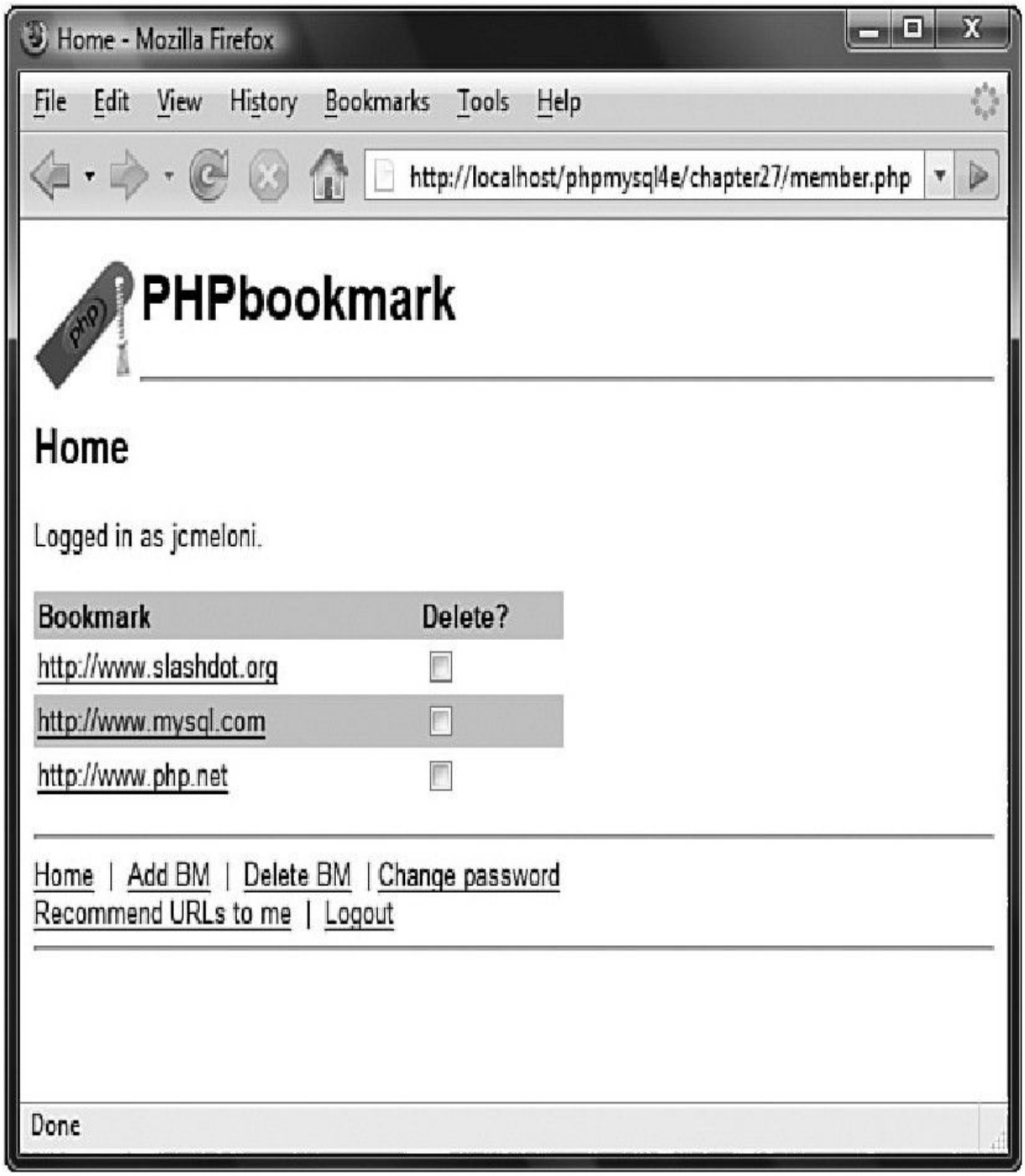


图　27-6　member.php脚本检查用户是否登录；取回并显示他的书签；给出一个选项菜单

我们将进一步讨论login()和check_valid_user()函数。login()函数如程序清单27-12所示。

程序清单27-12　user_auth_fns.php文件中的login()函数——该函数将用户信息与数据库中保存的信息进行比较

```
function login($username,$password){
//check username and password with db
//if yes,return true
//else throw exception
//connect to db
$conn=db_connect();
//check if username is unique
$result=$conn->query("select*from user
where username='".$username."'
and passwd=sha1('".$password."')");
if(!$result){
throw new Exception('Could not log you in.');
}
if($result->num_rows>0){
return true;
}else{
throw new Exception('Could not log you in.');
}
}
```

可以看到，该函数连接数据库并且检查数据库中是否有与本用户名和密码相匹配的用户。如果有，返回true，如果没有，或者本用户的信息不能被检查，将抛出一个异常。

函数check_valid_user()不再连接数据库，但是它将检查该用户是否有注册过的会话，也就是说，该用户是否已经登录。该函数如程序清单27-13所示。

程序清单27-13　user_auth_fns.php文件中的check_valid_user()函数——该函数检查用户是否有有效的会话

```
function check_valid_user(){
```

```
//see if somebody is logged in and notify them if not
if(isset($_SESSION['valid_user'])){
echo"Logged in as".$_SESSION['valid_user']."．＜br/＞";
}else{
//they are not logged in
do_html_heading('Problem:');
echo'You are not logged in.＜br/＞';
do_html_url('login.php','Login');
do_html_footer();
exit;
}
}
```

如果用户尚未登录，该函数将告诉他必须在登录之后才能浏览本页，并提供一个指向登录页的链接。

27.5.3　登出

我们可以已经注意到了，在图27-6所示的菜单选项上有一个标有"logout"的链接。该链接指向脚本logout.php。程序清单27-14给出了它的源代码。

程序清单27-14　logout.php——该脚本将结束一个用户会话

```
<?php
//include function files for this application
require_once('bookmark_fns.php');
session_start();
$old_user=$_SESSION['valid_user'];
//store to test if they*were*logged in
unset($_SESSION['valid_user']);
$result_dest=session_destroy();
//start output html
do_html_header('Logging Out');
if(!empty($old_user)){
if($result_dest){
//if they were logged in and are now logged out
echo'Logged out.<br/>';
do_html_url('login.php','Login');
}else{
//they were logged in and could not be logged out
echo'Could not log you out.<br/>';
}
}else{
//if they weren't logged in but came to this page somehow
echo'You were not logged in,and so have not been logged out.<br/>';
do_html_url('login.php','Login');
}
do_html_footer();
?>
```

我们会发现这段代码有些熟悉。因为它是根据第23章中编写的源代码而修改的。

27.5.4　修改密码

如果一个用户点击"Change Password"这个菜单选项，系统将打开如图27-7所示的表单。



图　27-7　change_passwd_form.php脚本为用户提供一个修改密码的表单

表单是由脚本change_passwd_form.php生成的。该脚本比较简单，只是调用了输出库中的一些函数，因而在这里，我们没有介绍它的源代码。

在用户提交该表单之后，将触发change_passwd.php脚本，如程序清单27-15所示。

程序清单27-15　change_passwd.php——该脚本将改变用户密码

```
<?php
require_once('bookmark_fns.php');
session_start();
do_html_header('Changing password');
//create short variable names
$old_passwd=$_POST['old_passwd'];
$new_passwd=$_POST['new_passwd'];
$new_passwd2=$_POST['new_passwd2'];
try{
check_valid_user();
if(!filled_out($_POST)){
throw new Exception('You have not filled out the form completely.
Please try again.');
}
if($new_passwd!=$new_passwd2){
throw new Exception('Passwords entered were not the same.
Not changed.');
}
if((strlen($new_passwd)>16)||(strlen($new_passwd)<6)){
throw new Exception('New password must be between 6 and 16 characters.
Try again.');
}
//attempt update
change_password($_SESSION['valid_user'],$old_passwd,$new_passwd);
```

```
echo'Password changed.';
}
catch(Exception$e){
echo$e->getMessage();
}
display_user_menu();
do_html_footer();
?>
```

以上脚本将检查用户是否已经登录（使用check_valid_user()函数），是否已经填好密码表单（使用filled_out()函数），新密码是否一致以及其长度是否符合规定。这些都不是新内容，我们已经在以前介绍过了。如果所有这些都已经完成，可以调用change_password()函数，如下所示：

```
change_password($_SESSION['valid_user'],$old_passwd,$new_passwd);
echo'Password changed.';
```

该函数来自user_auth_fns.php函数库。其源代码如程序清单27-16所示。

程序清单27-16　user_auth_fns.php文件中的change_password()函数——该函数更新数据库中的用户密码

```
function change_password($username,$old_password,$new_password){
//change password for username/old_password to new_password
//return true or false
//if the old password is right
//change their password to new_password and return true
//else throw an exception
login($username,$old_password);
$conn=db_connect();
$result=$conn->query("update user
set passwd=sha1('".$new_password."')
```

```
where username='".$username."'");
if(!$result){
throw new Exception('Password could not be changed.');
}else{
return true;//changed successfully
}
}
```

该函数调用了前面所介绍的login()函数来判断用户输入的旧密码是否正确。如果正确，函数将连接到数据库并将它更新为新密码。

27.5.5 重设遗忘的密码

除了修改密码，我们还需要解决用户忘记其密码的情形。请注意，在首页login.php中，我们专门为此情形提供了一个链接，"Forgotten your password?"，该链接指向脚本forgot_form.php，该脚本调用输出函数来显示如图27-8所示的表单。



图　27-8　forgot_form.php脚本提供了一个表单，在这里，用户可以请求重置他们的密码并发送给他们

该脚本非常简单，只是调用了一些输出函数，因此，我们将不再详细介绍它。当提交表单时，它将调用forgot_passwd.php脚本，这段代码更加有趣。其脚本如程序清单27-17所示。

程序清单27-17　forgot_passwd.php——该脚本将用户密码

```
<?php
require_once("bookmark_fns.php");
do_html_header("Resetting password");
//creating short variable name
$username=$_POST['username'];
try{
$password=reset_password($username);
notify_password($username,$password);
echo'Your new password has been emailed to you.<br/>';
}
catch(Exception$e){
echo'Your password could not be reset-please try again later.';
}
do_html_url('login.php','Login');
do_html_footer();
?>
```

重置为一个随机值并将新密码发送到用户的邮箱

可以看到，该脚本使用两个主要函数来实现此功能：reset_password()和notify_password()。我们将逐一介绍它们。

reset_password()函数将产生一个随机密码并将它保存到数据库。该函数的代码如程序清单27-18所示。

程序清单27-18　user_auth_fns.php文件中的reset_password()函数——该脚本将用户密码重置为随机值并将其发送到该用户邮箱

```
function reset_password($username){
```

```
//set password for username to a random value
//return the new password or false on failure
//get a random dictionary word b/w 6 and 13 chars in length
$new_password=get_random_word(6,13);
if($new_password==false){
throw new Exception('Could not generate new password.');
}
//add a number between 0 and 999 to it
//to make it a slightly better password
$rand_number=rand(0,999);
$new_password.=$rand_number;
//set user's password to this in database or return false
$conn=db_connect();
$result=$conn->query("update user
set passwd=sha1('".$new_password."')
where username='".$username."'");
if(!$result){
throw new Exception('Could not change password.');//not changed
}else{
return$new_password;//changed successfully
}
}
```

该函数通过从字典里获取随机单词来生成一个随机密码。调用get_random_word()函数并在得到的单词后面添加一个0～999之间的随机数作后缀。get_random_word()函数也包含在user_auth_fns.php库中。其脚本如程序清单27-19所示。

程序清单27-19　user_auth_fns.php文件中的get_random_word()函数——该函数从词典中获取一个随机单词，以生成新密码

```
function get_random_word($min_length,$max_length){
//grab a random word from dictionary between the two lengths
//and return it
//generate a random word
```

```
$word='';
//remember to change this path to suit your system
$dictionary='/usr/dict/words';//the ispell dictionary
$fp=@fopen($dictionary,'r');
if(!$fp){
return false;
}
$size=filesize($dictionary);
//go to a random location in dictionary
$rand_location=rand(0,$size);
fseek($fp,$rand_location);
//get the next whole word of the right length in the file
while((strlen($word)<$min_length)||(strlen($word)>$max_length)||
(strstr($word,"'")){
if(feof($fp)){
fseek($fp,0);//if at end,go to start
}
$word=fgets($fp,80);//skip first word as it could be partial
$word=fgets($fp,80);//the potential password
}
$word=trim($word);//trim the trailing\n from fgets
return$word;
}
```

要使该函数能够正常工作，我们需要一个词典。如果使用的是UNIX系统，其内置的拼写检查器ispell就带有单词词典，通常它位于/usr/dict/words或/usr/share/dict/words目录下。如果在以上两个位置都没有找到，在大多数系统上，可以使用如下命令找到一个词典：

```
$locate dict/words
```

如果使用的是其他的系统而且不愿安装ispell，不用担心，可以下载ispell使用的单词列表，其下载地址如下所示：http://wordlist.sourceforge.net/。

该网站也有许多其他语言的词典，因此如果喜欢其他任意一种语言，例如，Norwegian或Esperanto的单词，也可下载这些词典。所有这些文件的格式都是每个单词一行，每行通过换行符分开。

要从该文件中获取一个随机单词，首先应选取一个介于0到文件长度之间的位置，并从此位置开始读文件。如果从该随机位置开始一行一行地读，获取的很可能是单词的一部分，因此，我们通过两次调用fgets()函数，跳过开始的随机行，而将下面的一个单词作为需要的单词。

该函数有两处设计很巧妙。第一，如果在查找单词的时候到了文件结尾，可以从头开始，如下代码所示：

```
if(feof($fp)){
fseek($fp,0);//if at end,go to start
}
```

第二，可以搜索特定长度的单词：我们搜索从词典中抽出的每个单词，如果它的长度没有介于$min_length和$max_length之间，就继续搜索。同时，我们还将过滤带有单引号的单词。当使用该词时，我们过滤这些字符，但是获得下一个单词会更容易一些。

回到reset_password()函数，在生成了一个新密码之后，需要更新数据库以体现密码已被修改，并将新密码返回到主脚本；然后又将它传到notify_password()这个函数，该函数将新密码发送到用户邮箱。下面，让我们来了解notify_password()函数，如程序清单27-20所示。

程序清单27-20　user_auth_fns.php文件中的notify_password()函数——该函数将新密码以电子邮件方式发送给用户

```
function notify_password($username,$password){
//notify the user that their password has been changed
$conn=db_connect();
$result=$conn->query("select email from user
where username='".$username."'");
```

```
if(!$result){
throw new Exception('Could not find email address.');
}else if($result->num_rows==0){
throw new Exception('Could not find email address.');
//username not in db
}else{
$row=$result->fetch_object();
$email=$row->email;
$from="From:support@phpbookmark\r\n";
$mesg="Your PHPBookmark password has been changed to".$password."\r\n"
."Please change it next time you log in.\r\n";
if(mail($email,'PHPBookmark login information',$mesg,$from)){
return true;
}else{
throw new Exception('Could not send email.');
}
}
}
```

在这个函数中，给定一个用户名和密码，我们只需要在数据库中查找该用户的邮箱地址，调用PHP的mail()函数将其发送给该用户。

给用户一个真正随机的密码是更保险的——该密码是任何小写字母、大写字母、数字和标点符号的组合——而不只是如上设计的随机单词和数字的组合。但是，像"zigzag487"这样的密码，用户更易阅读和输入，这比真正随机数好。因为用户通常容易混淆字符串中的0和O（数字0和大写O），以及1和1（数字1和小写1）。

在我们的系统中，词典文件包含了45 000个单词记录。如果一个黑客知道我们是如何创建密码的，而且知道用户名称，就可以在尝试22 500 000次获得一个用户的密码。看上去，这种安全级别对于这种类型的应用程序是足够的，即使我们的用户没有按照电子邮件中的建议再次修改密码。

27.6　实现书签的存储和检索

在实现了与用户账户相关的功能后，现在，我们开始讨论如何保存、检索和删除书签。

27.6.1　添加书签

用户点击用户菜单上的"Add BM"链接可以添加书签。该链接将用户带到如图27-9所示的页面。

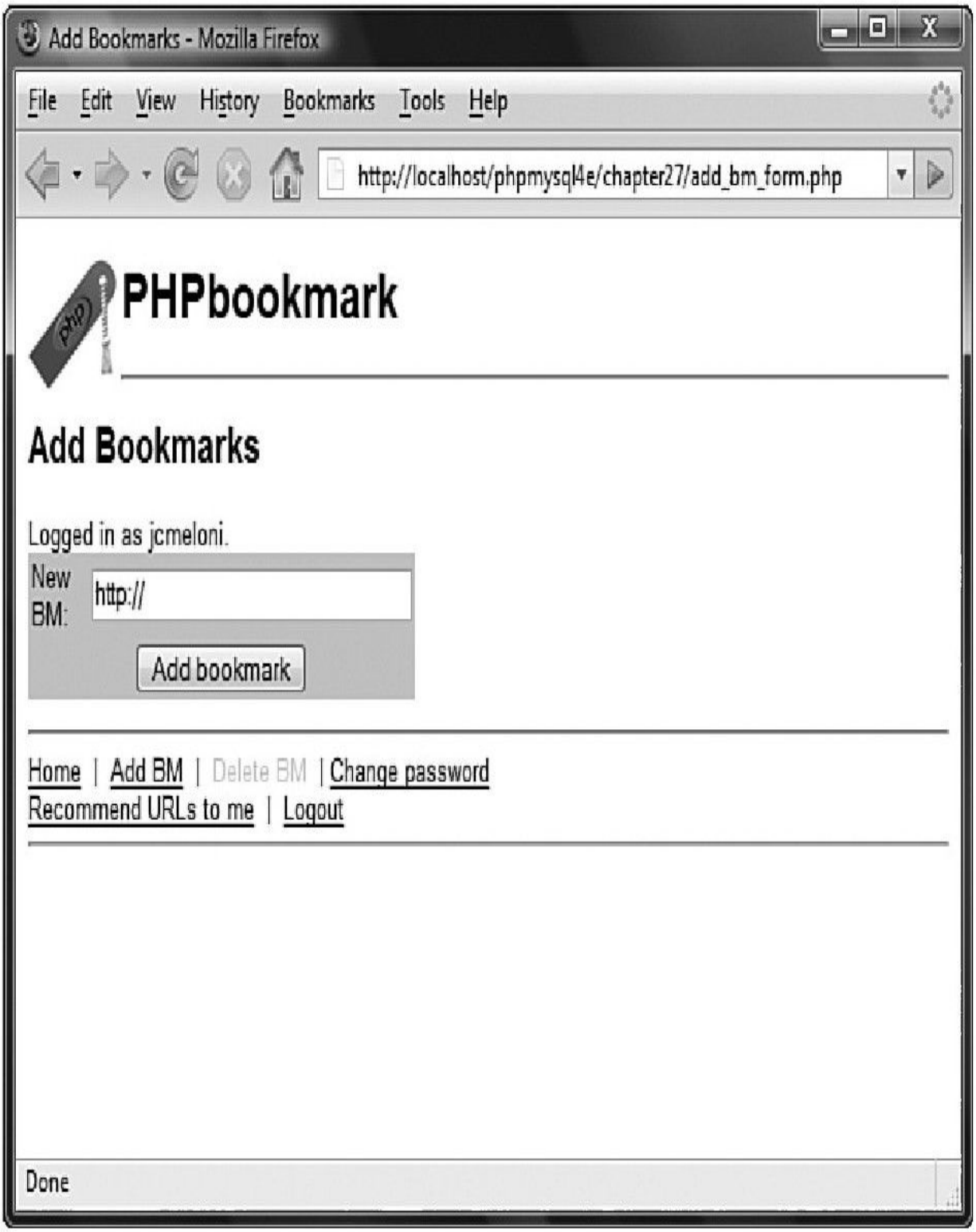


图　27-9　add_bm_form.php脚本将提供一个表单，用户在此可以将书签添加到他们的书签页中

同样，由于这段脚本也是非常简单的，并且只调用了一些输出函数，因此，我们也不深入讨论。当表单被提交的时候，系统调用add_bms.php脚本，该脚本如程序清单27-21所示。

程序清单27-21　add_bms.php——该脚本添加新书签到用户的个人页面

```
<?php
require_once('bookmark_fns.php');
session_start();
//create short variable name
$new_url=$_POST['new_url'];
do_html_header('Adding bookmarks');
try{
check_valid_user();
if(!filled_out($_POST)){
throw new Exception('Form not completely filled out.');
}
//check URL format
if(strstr($new_url,'http://')===false){
$new_url='http://'.$new_url;
}
//check URL is valid
if(!(@fopen($new_url,'r'))){
throw new Exception('Not a valid URL.');
}
//try to add bm
add_bm($new_url);
echo'Bookmark added.';
//get the bookmarks this user has saved
if($url_array=get_user_urls($_SESSION['valid_user'])){
display_user_urls($url_array);
}
}
catch(Exception$e){
echo$e->getMessage();
```

```
}
display_user_menu();
do_html_footer();
?>
```

这段脚本也遵循了验证、数据库输入和输出的模式。

要验证用户身份，我们应该首先调用filled_out()函数检查该用户是否完全填写表单。然后，再执行两项检查URL的操作。首先，调用strstr()函数，检查URL是否以http://开头。如果不是，我们就将其添加到URL的开头。完成此操作后，就可确切地检查该URL是否存在。回顾一下第20章“使用网络函数和协议函数”，我们可以调用fopen()函数打开一个以http://开头的URL。如果可以打开这个文件，就假定该URL是有效的，并调用add_bm()函数将其添加到数据库中。

本函数和其他与书签相关的函数都保存在函数库url_fns.php中。程序清单27-22显示了add_bm()函数的代码。

程序清单27-22　url_fns.php文件中的add_bm()函数——该函数将用户提交的新书签添加到数据库中

```
<?php
require_once('bookmark_fns.php');
session_start();
//create short variable name
$new_url=$_POST['new_url'];
do_html_header('Adding bookmarks');
try{
check_valid_user();
if(!filled_out($_POST)){
throw new Exception('Form not completely filled out.');
}
//check URL format
if(strstr($new_url,'http://')===false){
$new_url='http://'.$new_url;
}
```

```
//check URL is valid
if(!(@fopen($new_url,'r'))){
throw new Exception('Not a valid URL.');
}
//try to add bm
add_bm($new_url);
echo'Bookmark added.';
//get the bookmarks this user has saved
if($url_array=get_user_urls($_SESSION['valid_user'])){
display_user_urls($url_array);
}
}
catch(Exception$e){
echo$e->getMessage();
}
display_user_menu();
do_html_footer();
?>
```

该函数也很简单。它检查用户是否在数据库中已经有了该书签（尽管他们不可能两次输入同一个书签，但很可能要更新该页）。如果书签是新的，它就被添加到数据库中。

回头看看add_bm.php函数库，可以看出，它最后执行的操作是调用get_user_urls()函数和display_user_urls()函数，这与member.php是相同的。我们将在接下来的内容中继续讨论这些函数。

27.6.2　显示书签

在member.php脚本和add_bm()函数中，我们使用了函数get_user_urls()和display_user_urls()。它们分别从数据库中检索用户的书签和显示这些书签。get_user_urls()函数包含在url_fns.php库中。而display_user_urls()函数包含在output_fns.php库中。

get_user_urls()函数如程序清单27-23所示。

我们简要介绍一下该函数的执行步骤，它以用户名作为参数，从数据库中取回该用户的书签，返回一组URL；或者如果书签获取失败，返回false值。

程序清单27-23　url_fns.php文件中的get_user_urls()函数——该函数从数据库中取回用户书签

```
function get_user_urls($username){
//extract from the database all the URLs this user has stored
$conn=db_connect();
$result=$conn->query("select bm_URL
from bookmark
where username='".$username."'");
if(!$result){
return false;
}
//create an array of the URLs
$url_array=array();
for($count=1;$row=$result->fetch_row();++$count){
$url_array[$count]=$row[0];
}
return$url_array;
}
```

get_user_urls()函数将返回一组可以传给函数display_user_urls()的URL。该函数也是一个简单的HTML输出函数，它可将用户的URL以美观的表格形式显示在浏览器中，在这里，我们也不详细讨论。回到图27-6，看看输出是什么。实际上函数将URL输出到一个表单。而每个URL右边是一个复选框，用以选定要删除的书签。接下来，我们就将讨论它。

27.6.3 删除书签

当用户将一些书签标记为删除并点击菜单选项中的"Delete BM"时，就将提交一个包含URL的表单。每个复选框由display_user_urls()函数中的如下代码产生：

```
echo"<tr bgcolor=\"".$color."\"><td><a
➥href=\"".$url."\">".htmlspecialchars($url)."</a></td>
<td><input type=\"checkbox\"name=\"del_me[]\"
value=\"".$url."\"/></td>
</tr>";
```

每个输入的名称是del_me[]。这就是说，在该表单激活的PHP脚本中，我们可以访问名为$del_me的数组，该数组包含所有要删除的书签。

点击"Delete BM"就触发了delete_bms.php脚本，该脚本源代码如程序清单27-24所示。

程序清单27-24　delete_bms.php——该脚本从数据库中删除书签

```
<?php
require_once('bookmark_fns.php');
session_start();
//create short variable names
$del_me=$_POST['del_me'];
$valid_user=$_SESSION['valid_user'];
do_html_header('Deleting bookmarks');
check_valid_user();
if(!filled_out($_POST)){
echo'<p>You have not chosen any bookmarks to delete.<br/>
Please try again.</p>';
display_user_menu();
do_html_footer();
```

```
exit;
}else{
if(count($del_me)>0){
foreach($del_me as$url){
if(delete_bm($valid_user,$url)){
echo'Deleted'.htmlspecialchars($url).'.<br/>';
}else{
echo'Could not delete'.htmlspecialchars($url).'.<br/>';
}
}
}else{
echo'No bookmarks selected for deletion';
}
}
//get the bookmarks this user has saved
if($url_array=get_user_urls($valid_user)){
display_user_urls($url_array);
}
display_user_menu();
do_html_footer();
?>
```

在脚本的开始，我们执行了常规的确认操作。当确定用户已经删除选中的书签时，将通过如下的循环将其删除：

```
foreach($del_me as$url){
if(delete_bm($valid_user,$url)){
echo'Deleted'.htmlspecialchars($url).'.<br/>';
}else{
echo'Could not delete'.htmlspecialchars($url).'.<br/>';
}
}
```

可以看到，delete_bm()函数并没有执行从数据库中删除书签的实际工作。该函数如程序清单27-25所示。

程序清单27-25　url_fns.php文件中的delete_bm()函数——该函数从用户的书签列表中删除一个书签

```
function delete_bm($user,$url){
//delete one URL from the database
$conn=db_connect();
//delete the bookmark
if(!$conn->query("delete from bookmark where
username='".$user."'and bm_url='".$url."'")){
throw new Exception('Bookmark could not be deleted');
}
return true;
}
```

可以看到，这也是一个相当简单的函数。它试图从数据库中删除特定用户的书签。需要注意的是，我们要删除的是“用户名-书签”对。其他用户可能仍然拥有此书签URL。

在系统中运行删除脚本的输出示例如图27-10所示。

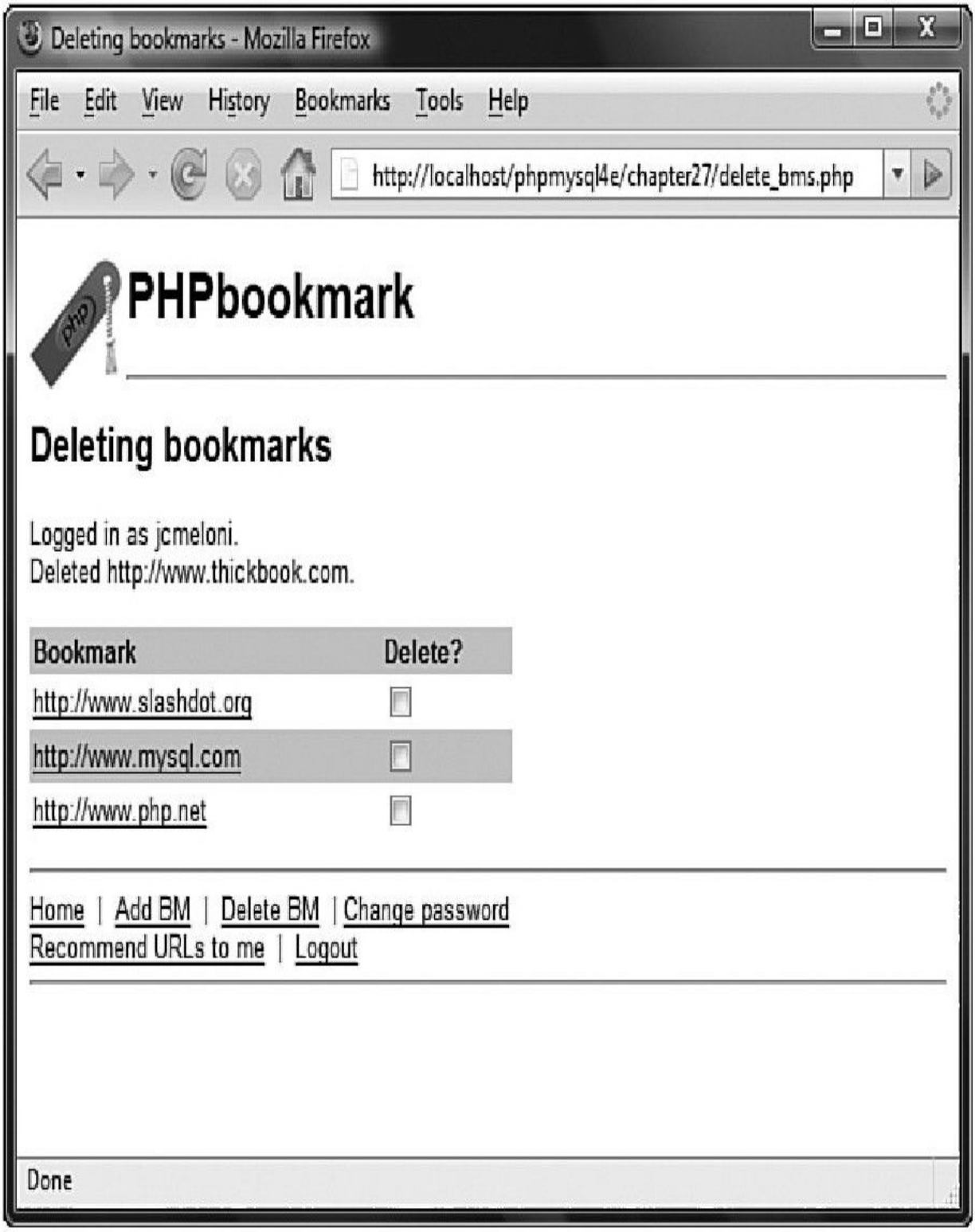


图　27-10　删除脚本将通知用户已删除了书签，然后显示其余的书签

与在add_bms.php脚本中的操作类似，当数据库被修改之后，我们将调用get_user_urls()函数和display_user_urls()函数显示新的书签。

27.7 实现书签推荐

最后，我们将讨论书签推荐脚本recommend.php。我们可以通过许多方法实现书签推荐。在此，我们决定应用“相似意向”的推荐。该推荐的含义是，查找与给定用户至少有一个相同书签的其他用户。其他用户的其他书签也对给定的用户有吸引力。

将“相似意向”应用到SQL查询最简单的方法是使用子查询。第一个子查询如下所示：

```
select distinct(b2.username)
from bookmark b1,bookmark b2
where b1.username='".$valid_user."'
and b1.username!=b2.username
and b1.bm_URL=b2.bm_URL)
```

这个查询使用别名将数据库表bookmark进行自身连接——这是一个很奇怪但是有时候又非常有用的概念。假设有两个书签表，b1和b2。在b1中，查询当前用户及其书签。在另一个表中，查询所有其他用户的书签。我们需要查找的是用户书签中有一个URL与当前用户相同（b1.bm_URL=b2.bm_URL）的其他用户（b2.username）。其他用户不包括当前用户（b1.username!=b2.username）。

该查询将给出一个与当前用户意向相似的人的列表。得到了这个用户列表后，可以用下面的查询搜索他们的其他书签了：

```
select bm_URL
from bookmark
where username in
(select distinct(b2.username)
from bookmark b1,bookmark b2
where b1.username='".$valid_user."'
and b1.username!=b2.username
and b1.bm_URL=b2.bm_URL)
```

可以添加第二个子查询来过滤当前用户的书签；如果用户已经有了这些书签，就不必再将该书签推荐给他。最后，对$popularity变量进行书签过滤。我们不希望推荐太个性化的URL，因此只将一定数量的其他用户做了书签的URL推荐给用户。最终的查询如下所示：

```
select bm_URL
from bookmark
where username in
(select distinct(b2.username)
from bookmark b1,bookmark b2
where b1.username='".$valid_user."'
and b1.username!=b2.username
and b1.bm_URL=b2.bm_URL)
and bm_URL not in
(select bm_URL
from bookmark
where username='".$valid_user."')
group by bm_url
having count(bm_url)>".$popularity;
```

如果我们期望许多用户使用我们的系统，可以调整变量$popularity，只推荐许多其他用户做了书签的URL。许多人做了书签的URL可能质量更高，这样的书签当然比一般的页面更大众化、更具吸引力。

实现书签推荐的完整脚本如程序清单27-26和程序清单27-27所示。推荐的主脚本称为recommend.php（请参阅程序清单27-26），它调用来自url_fns.php函数库（请参阅程序清单27-27）的函数recommend_urls()。

程序清单27-26　recommend.php——推荐某一用户可能喜欢的书签

```
<?php
require_once('bookmark_fns.php');
```

```
session_start();
do_html_header('Recommending URLs');
try{
check_valid_user();
$urls=recommend_urls($_SESSION['valid_user']);
display_recommended_urls($urls);
}
catch(Exception$e){
echo$e->getMessage();
}
display_user_menu();
do_html_footer();
?>
```

程序清单27-27　url_fns.php文件中的recommend_urls()函数——该脚本做出实际的推荐

```
function recommend_urls($valid_user,$popularity=1){
//We will provide semi intelligent recommendations to people
//If they have an URL in common with other users,they may like
//other URLs that these people like
$conn=db_connect();
//find other matching users
//with an url the same as you
//as a simple way of excluding people's private pages,and
//increasing the chance of recommending appealing URLs,we
//specify a minimum popularity level
//if$popularity=1,then more than one person must have
//an URL before we will recommend it
$query="select bm_URL
from bookmark
where username in
(select distinct(b2.username)
from bookmark b1,bookmark b2
where b1.username='".$valid_user."'
```

```
and b1.username!=b2.username
and b1.bm_URL=b2.bm_URL)
and bm_URL not in
(select bm_URL
from bookmark
where username='".$valid_user."')
group by bm_url
having count(bm_url)>".$popularity;
if(!($result=$conn->query($query))){
throw new Exception('Could not find any bookmarks to recommend.');
}
if($result->num_rows==0){
throw new Exception('Could not find any bookmarks to recommend.');
}
$urls=array();
//build an array of the relevant urls
for($count=0;$row=$result->fetch_object();$count++){
$urls[$count]=$row->bm_URL;
}
return$urls;
}
```

---

recommend.php的输出示例如图27-11所示。

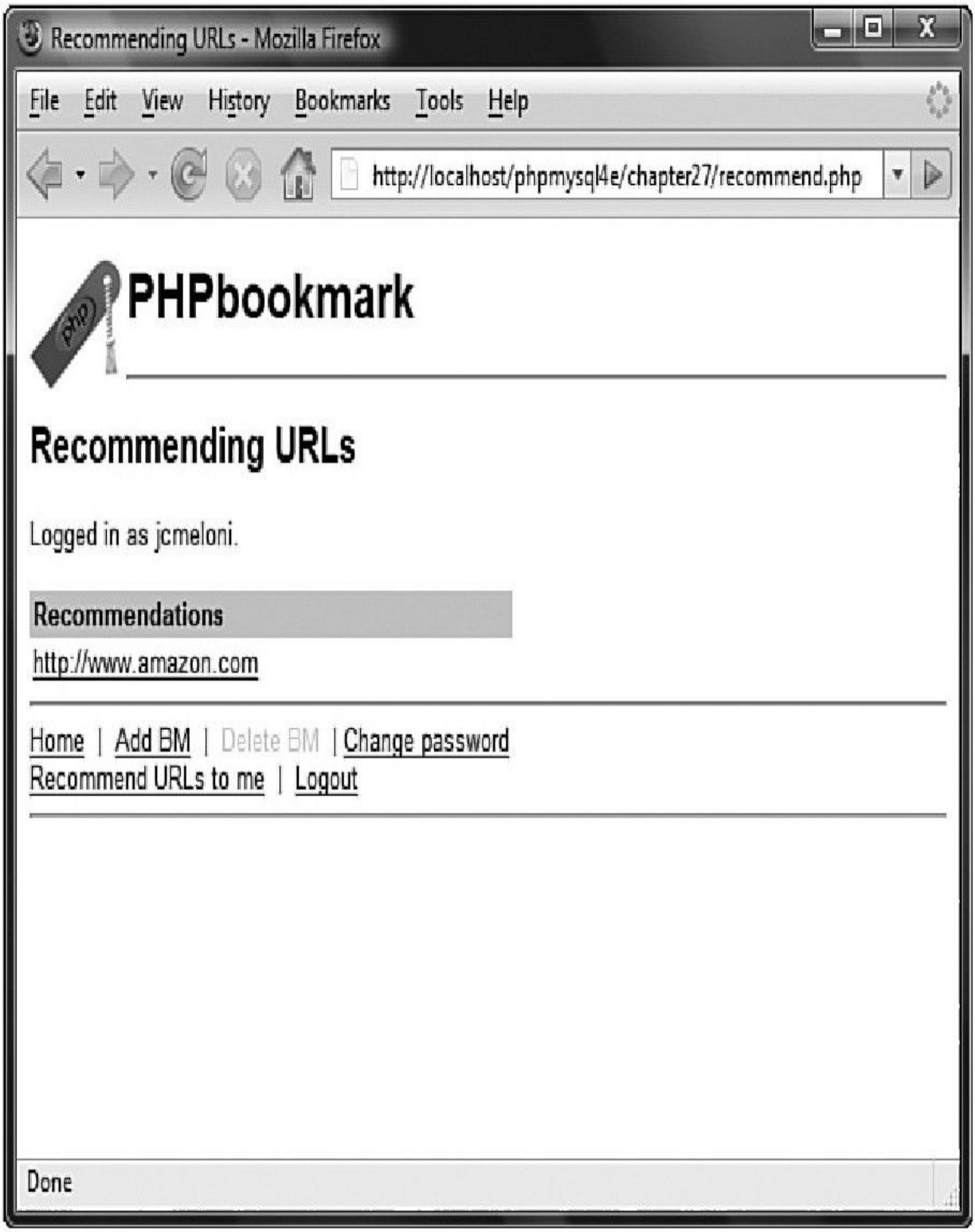


图　27-11　脚本向该用户推荐了他可能喜欢的amazon.com，数据库中至少有两位用户将amazon.com作为他们的书签

27.8 考虑可能的扩展

在以上的内容中，我们介绍了PHP-bookmark应用程序的基本功能。它还有许多可以扩展的地方。例如，可以考虑添加：

■按主题分类的一组书签

■书签推荐功能中的“将此URL添加到我的书签”的链接

■基于数据库中最流行URL的推荐，或者基于某一特定主题的推荐

■一个管理界面，用以创建、管理用户和书签

■使推荐书签更智能化或更快的方法

■附加的用户输入错误检查

实践！这是最好的学习方法。

27.9 下一章

在下一个项目中，我们将创建一个购物车，使用该购物车可以让用户浏览我们的网站，并在浏览的时候购买商品，直到最后结账并使用电子的方式付款。

第28章　创建一个购物车

在本章中，我们将学习如何创建一个基本的购物车。我们将此购物车添加到Book-O-Rama数据库之上，该数据库是我们在第二篇“使用MySQL”实现的。此外，我们还将探讨另外一种选择：创建和使用一个已有的、开放源代码的PHP购物车。

如果以前还没有听说过这些内容，那让我们先了解一下术语“购物车”。“购物车”（有时也叫购物篮）用来描述一种特定的在线购物机制。当浏览一个在线目录的时候，可以将一些商品添加到购物车中。在完成浏览之后，要与在线商店结账——也就是买下购物车内的商品。

为了实现购物车，我们需要完成下面的功能：

■在线出售商品的数据库

■一个在线产品目录，按商品种类分类

■一个能记录用户打算购买商品的购物车

■结账脚本，处理付款和运送细节

■一个管理界面

28.1　解决方案的组成

回忆一下在本书第二篇创建的Book-O-Rama数据库。在本项目中，我们将建立Book-O-Rama在线商店并运行它。该解决方案的组成主要包括如下所示。

■我们需要找到一种将数据库连接到用户的浏览器的方法。用户能够按目录浏览商品。

■用户应该能够从商品目录中选取商品以便此后购买。我们也要能够记录他们选中的物品。

■当用户完成购买，要合计他们的订单，获取运送商品细节，并处理付款。

■我们还应该为Book-O-Rama网站创建一个管理界面，以便管理员在上面添加、编辑图书和目录。

现在，我们已经了解了项目的需求，开始开始设计解决方案及其组成。

28.1.1 创建一个在线目录

我们已经有了一个用来创建Book-O-Rama目录的数据库。但是，要实现该目录，还需要修改和添加一些功能。其中之一就是要添加图书的目录，如需求中所述。

我们还要为现存的数据库添加一些关于商品运送地址、付款细节等的信息。我们已经知道了如何使用PHP创建一个连接到MySQL数据库的界面，因此创建管理界面这一步就很简单了。

在完成顾客订单的时候，还应该使用事务。要使用事务，必须将Book-O-Rama表格转换成使用InnoDB存储引擎。这个处理也是非常简单和直观的。

28.1.2　在用户购买商品的时候记录购买行为

记录用户所购买的商品有两种基本方法。一种是将用户的选择存入数据库中，另一种是使用会话变量。

使用会话变量逐页记录用户的方法是很容易实现的，因为它不要求我们不停地查询数据库。这样做也避免了在结束的时候留下许多垃圾数据，这些数据来自于那些只是浏览或不停地改变主意的用户。

因此，我们需要设计一个或一组会话变量来保存用户的选择。当用户完成购买并付款之后，将此信息送到数据库作为一个事务处理的记录。

我们还可以使用该数据给出一个当前购物车的摘要描述，将其显示在页面的某个位置，以便用户在任何时候都知道花费是多少。

28.1.3 实现一个付款系统

在本项目中，我们要合计用户的订单总价，并获取送货详细信息。实际上，我们并不处理任何付款。如今，有许多的付款系统可供使用，但是，对于每一个系统应用，付款系统又各不相同。我们将编写一个dummy函数，该函数可以用所选定系统的界面代替。

尽管有一些不同的付款网关可以使用，但是，对于这些付款网关的接口来说，实时信用卡处理接口的功能都是类似的。你必须在银行开通一个商业账户，确定能够接受的信用卡类型——通常，你的银行会针对你所选择的支付系统给出一个推荐的信用卡提供商列表，你的付款系统提供商会给出该付款系统所需的参数以及如何传递这些参数。大多数付款系统都有PHP版本的示例代码，这样就便于替代本章所创建的示例函数。

当使用付款系统时，系统将你的数据发送给银行并且返回成功代码或任何一个错误代码类型。在数据交换的时候，付款网关将收取设置费用或年费，以及基于交易金额的手续费。某些提供商甚至会收取取消交易的费用。

不过，你的付款系统至少需要来自客户的信息（例如，信用卡），标识你的信息（指定交易款项的商业账户），以及交易的总金额。

我们可以通过一个用户的购物车会话变量计算出订单的总量。将订单的最终信息记录到数据库，并在此时销毁会话变量。

28.1.4　创建一个管理界面

除了这些，我们将创建一个管理员界面。在此界面上，可以添加、删除和编辑数据库中的图书及目录。

通常，我们将要用到的一个编辑功能是修改某一物品的价格（例如，要举行一次特别的出售活动或甩卖）。这就意味着在保存用户订单的时候，也要保存他所买商品的价格。如果我们已经保存的记录只有每个用户所订购的物品，而按照每个商品的当前价格来计费，可以想像，那会是一场“噩梦”。这也意味着如果客户要退回或更换商品，我们也可以正确地计算商品价格。

在本项目中，我们并不打算创建一个可履行和记录订单的界面。我们可以根据需要在这个基本系统中添加此功能。

28.2 解决方案概述

现在，让我们将各个部分整合到一起。该系统有两个基本视图：用户视图和管理员视图。在仔细考虑需要的功能之后，我们提出了两个流程设计，一个视图对应一个流程设计。它们分别如图28-1和图28-2所示。

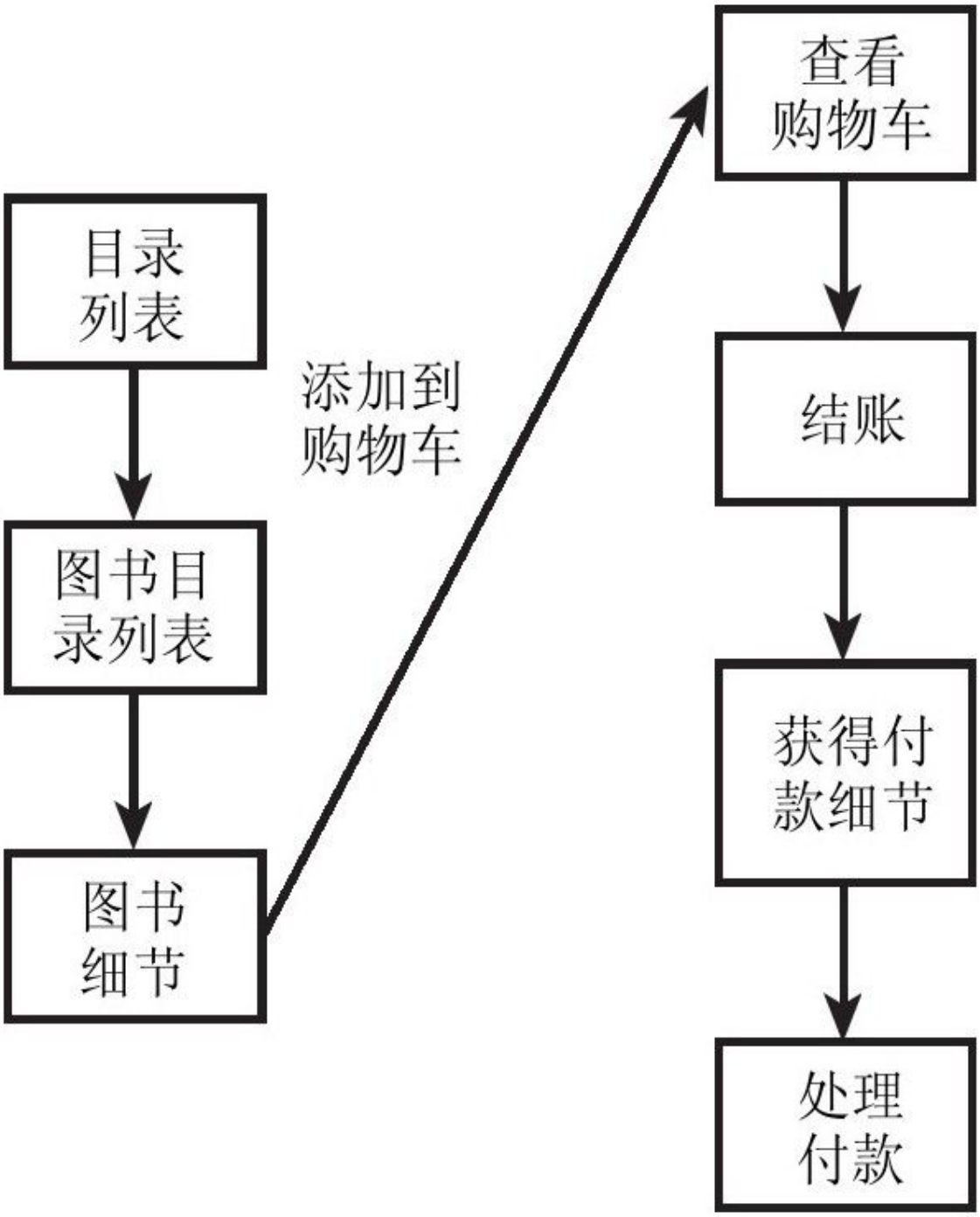


图　28-1　Book-O-Rama系统的用户视图可使得用户按照目录查看图书，浏览图书的详细信息，将书添加到购物车，以及购买它们

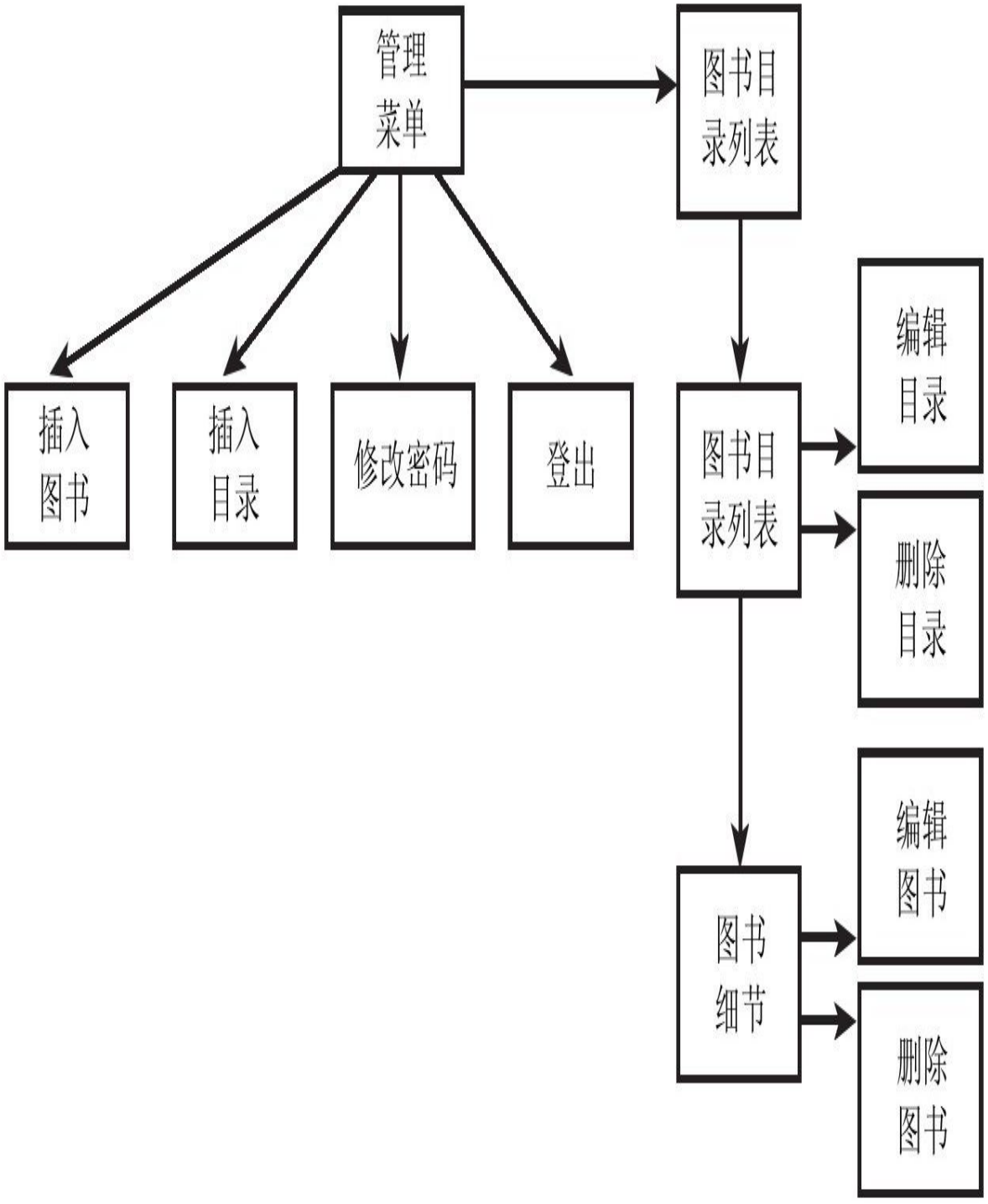


图　28-2　Book-O-Rama系统管理员视图允许管理员插入、编辑和删除图书及目录

在图28-1中，我们显示了网站中用户部分的脚本之间的主要链接。用户将首先进入主页面，该页面给出了网站中所有图书的目录。从该页面开始，用户可以进入特定的图书目录，从该目录又可以进入某一本书的详细介绍。

我们将为用户提供一个链接以添加特定的图书到购物车。根据购物车中的商品，用户可以与该在线商店结账付款。

图28-2所示的是管理员界面——它需要更多的脚本，但是新脚本却并不是很多。这些脚本允许管理员登录并插入图书和目录记录。

实现编辑、删除图书和目录最简单的方法是为管理员提供一个界面，该界面与用户界面略有不同。管理员仍然能够浏览目录和图书，但与用户访问购物车不同的是，管理员能够进入特定的图书和目录，并且编辑和删除该书和目录。通过设计适用一般用户和管理员的同样脚本，我们可以节省时间和精力。

该应用程序由如下所示的三个主要代码模块组成：

■目录

■购物车和订单处理（我们将此二者捆绑在一起是因为它们之间的联系非常紧密）

■管理

与第27章“建立用户身份验证机制和个性化设置”介绍的项目一样，我们也要创建和使用一组函数库。在这个项目中，我们将使用一个函数API，它与上一个项目的函数API类似。我们试图把输出HTML的代码放到一个函数库中，以支持分离逻辑和内容的原则，更重要的是，这样做可以使代码更易阅读和维护。

针对这个项目，我们也需要对Book-O-Rama数据库作一些小修改。我们已经重新命名了数据库book_sc（购物车），以区分在本书第二篇所创建的数据库。

本项目所需的所有代码都可以在本书附带的文件中找到。表28-1给出了本应用程序中用到的这些文件的摘要。

表28-1 购物车应用程序中用到的文件

名　称	模　块	描　述
index.php	目录	网站首页，显示系统中的图书目录
show_cat.php	目录	显示特定目录包含的所有图书
show_book.php	目录	显示特定图书的详细信息
show_cart.php	购物车	显示用户购物车的内容。也用来向购物车中添加图书
checkout.php	购物车	向用户显示所有的订单细节。获取商品运送细节
purchase.php	购物车	从用户获取付款细节
process.php	购物车	处理付款细节，将订单添加到数据库
login.php	管理	允许管理员登录进行修改
logout.php	管理	管理员退出
admin.php	管理	主管理菜单
change_password_form.php	管理	允许管理员修改密码的表格
change_password.php	管理	修改管理员密码
insert_category_form.php	管理	允许管理员向数据库中添加一个目录的表格
insert_category.php	管理	向数据库中插入新目录
insert_book_form.php	管理	管理员添加新书到系统的表单
insert_book.php	管理	将新书插入到数据库
edit_category_form.php	管理	管理员编辑目录的表单
edit_category.php	管理	更新数据库中的目录
edit_book_form.php	管理	管理员编辑图书信息的表单
edit_book.php	管理	更新数据库中的图书信息
delete_category.php	管理	从数据库中删除一个目录
delete_book.php	函数	从数据库中删除一本书
book_sc_fns.php	函数	该应用程序的包含文件集合
admin_fns.php	函数	管理脚本使用的函数集合
book_fns.php	函数	用以保存和获取图书数据的函数集合
order_fns.php	函数	用以保存和获取订单数据的函数集合
output_fns.php	函数	输出HTML的函数集合
data_valid_fns.php	函数	验证用户输入数据的函数集合
db_fns.php	函数	连接book_sc数据库的函数集合
user_auth_fns.php	函数	授权管理员用户的函数集合
book_sc.sql	SQL	创建book_sc数据库的SQL
populate.sql	SQL	插入样本数据到book_sc数据库中的SQL

现在，让我们开始了解每个模块的具体实现。

提示　该应用程序的代码量非常大。其中要实现的许多功能我们都已经见过（特别是在上一章中），例如，将数据保存到数据库，从数据库中取回数据，授予用户管理权限。在这方面的代码，我们只简要地看一下，而将大部分时间花在购物车函数上。

28.3　实现数据库

正如前面所提到的，我们已经对在本书第二篇创建的数据库进行了少量修改。创建book_sc数据库的SQL代码如程序清单28-1所示。

程序清单28-1　book_sc.sql——创建book_sc数据库的SQL代码

```
create database book_sc;
use book_sc;
create table customers
(
customerid int unsigned not null auto_increment primary key,
name char(60)not null,
address char(80)not null,
city char(30)not null,
state char(20),
zip char(10),
country char(20)not null
)type=InnoDB;
create table orders
(
orderid int unsigned not null auto_increment primary key,
customerid int unsigned not null references customers(customerid),
amount float(6,2),
date date not null,
order_status char(10),
ship_name char(60)not null,
ship_address char(80)not null,
ship_city char(30)not null,
ship_state char(20),
ship_zip char(10),
ship_country char(20)not null
)type=InnoDB;
create table books
```

```
(
isbn char(13)not null primary key,
author char(100),
title char(100),
catid int unsigned,
price float(4,2)not null,
description varchar(255)
)type=InnoDB;
create table categories
(
catid int unsigned not null auto_increment primary key,
catname char(60)not null
)type=InnoDB;
create table order_items
(
orderid int unsigned not null references orders(orderid),
isbn char(13)not null references books(isbn),
item_price float(4,2)not null,
quantity tinyint unsigned not null,
primary key(orderid,isbn)
)type=InnoDB;
create table admin
(
username char(16)not null primary key,
password char(40)not null
);
grant select,insert,update,delete
on book_sc.*
to book_sc@localhost identified by'password';
```

尽管最初的Book-O-Rama界面并没有错误，但要在线使用，必须满足一些其他的新需求。

对最初数据库进行的修改如下所示：

■增加更多的用户地址域。既然要建立实际的应用程序，这就很重要了。

■为每个订单增加一个运送地址。顾客的联系地址可能并不是商品运送地址，特别当他在该网站上给别人买礼物的时候。

■增加一个categories表并在books表中增加目录标识符（catid）。将图书分类会使网站更易浏览。

■增加item_price列到order_items表，因为某项商品的价格可能改变。我们要知道顾客购买时候的商品价格。

■增加一个admin表以保存管理员登录名和密码的详细信息。

■删除评论表。可以以该项目扩展特性的形式增加评论功能。作为替代，每本书有一个描述域，该域对本书作简要的介绍。

■将存储引擎修改为InnoDB。这样处理后，就可以使用外键，同时还可以在输入顾客订单信息的时候使用事务。

要在系统中建立数据库，必须以root的身份在MySQL中运行book_sc.sql脚本，如下所示：

```
mysql-u root-p<book_sc.sql
```

（需要提供root用户密码）

在进行以上操作之前，最好修改book_sc用户的密码，修改后的密码应该比原始密码"password"好。请注意，如果在book_sc.sql脚本中修改密码，还需要在db_fns.php脚本中修改它（稍后，我们将详细介绍）。

我们还要包含一个示例数据文件。该文件名为populate.sql。可以使用上述方法运行该sql语句将示例数据添加到数据库。

28.4　实现在线目录

在这个应用程序中，将使用3个目录脚本：主页、目录页和图书详细信息页。

本站点的首页由脚本index.php产生。该脚本输出如图28-3所示。

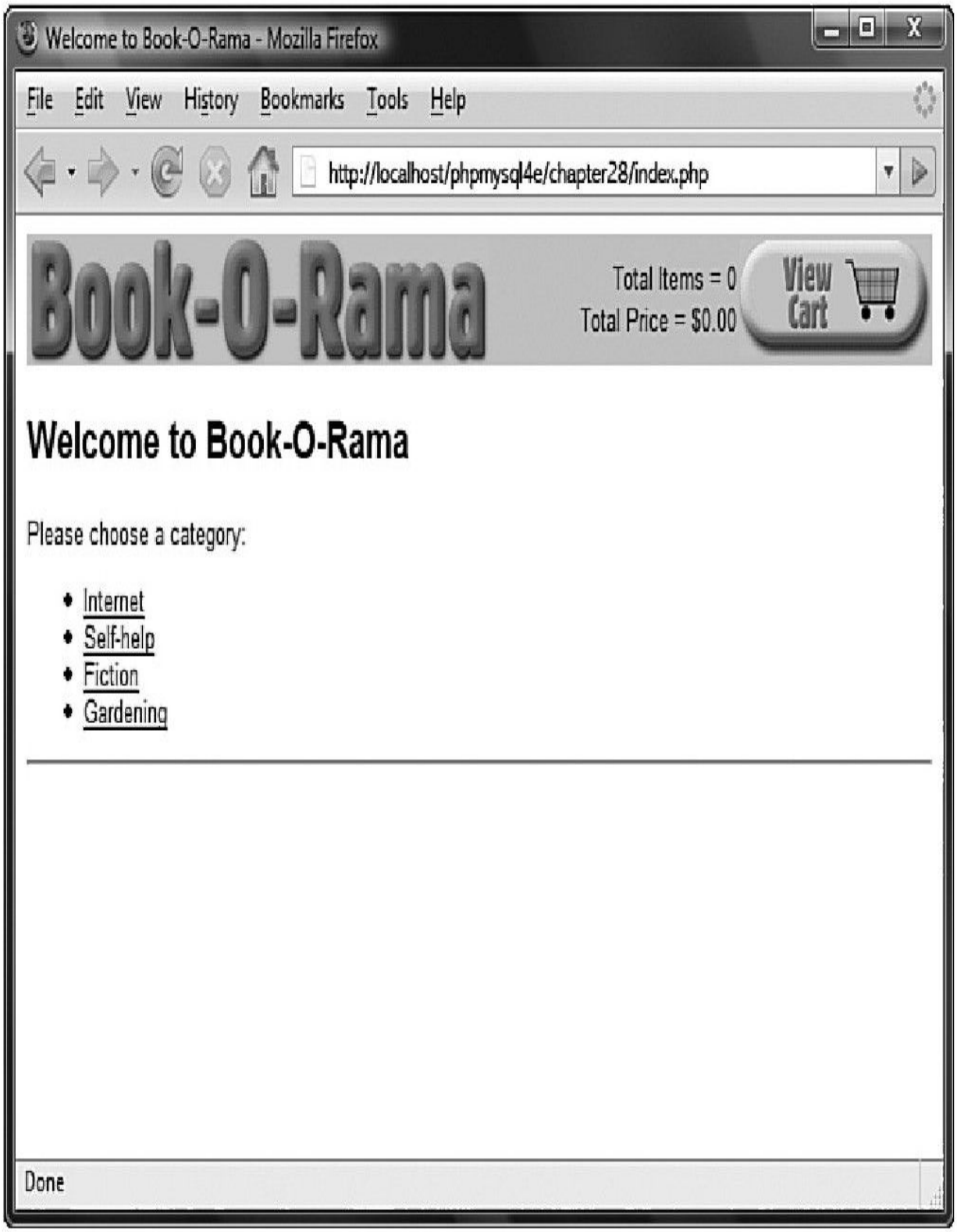


图　28-3　该页列出了可在线购买图书的目录

注意在站点中，除了给出了一个目录列表外，屏幕的右上角还有一个购物车链接，以及对该购物车的简要描述。在用户浏览站点和购买商品的时候，它会一直显示在每一个页面上。

如果用户点击其中的一个目录链接，它将进入一个目录页面，该页由脚本show_cat.php产生。Internet图书的目录页面如图28-4所示。

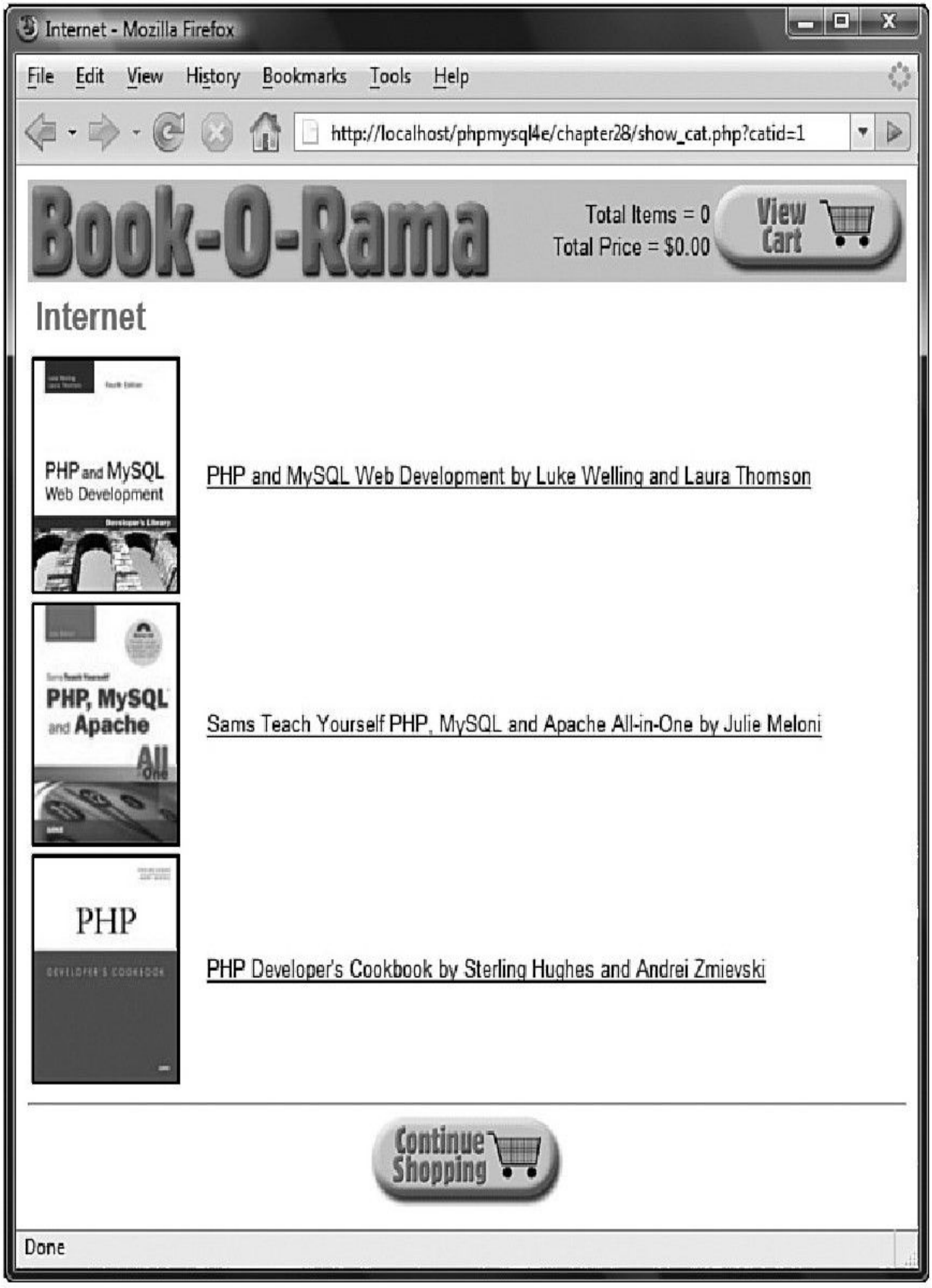

Internet - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/phpmysql4e/chapter28/show_cat.php?catid=1
Book-O-Rama
Total Items = 0
Total Price = $0.00
View Cart
Internet
PHP and MySQL Web Development
PHP and MySQL Web Development by Luke Welling and Laura Thomson
PHP, MySQL and Apache
Sams Teach Yourself PHP, MySQL and Apache All-in-One by Julie Meloni
PHP
PHP Developer's Cookbook by Sterling Hughes and Andrei Zmievski
Continue Shopping
Done

图　28-4　列出目录中的每本书，旁边附有一张图片

所有包含在Internet目录的图书以链接形式列出。如果用户点击其中一个链接，将进入该书的详细信息页。一本书的详细信息页如图28-5所示。

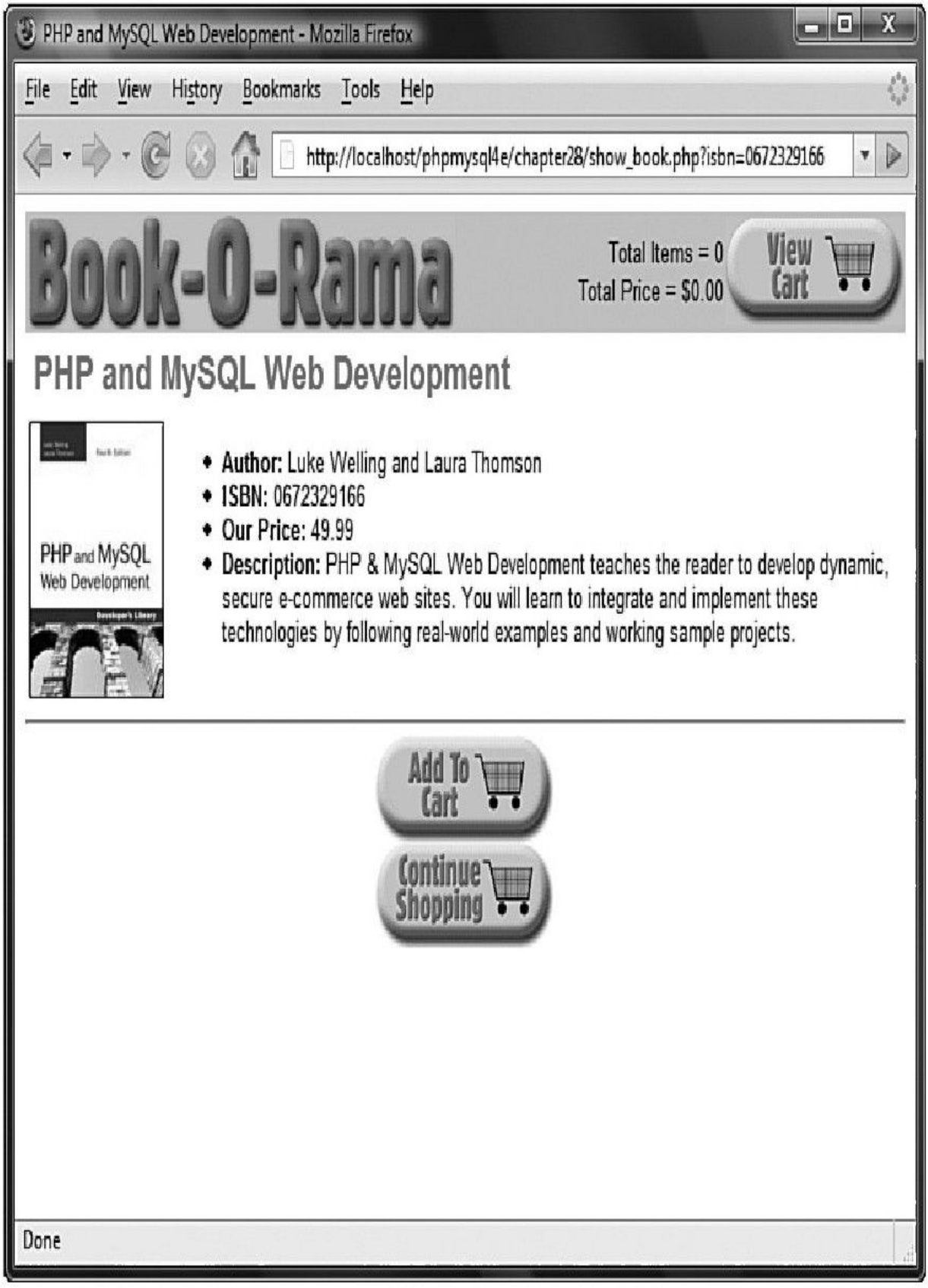

PHP and MySQL Web Development - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/phpmysql4e/chapter28/show_book.php?isbn=0672329166
Book-O-Rama
Total Items = 0
Total Price = $0.00
View Cart
PHP and MySQL Web Development
PHP and MySQL Web Development
Author: Luke Welling and Laura Thomson
ISBN: 0672329166
Our Price: 49.99
Description: PHP & MySQL Web Development teaches the reader to develop dynamic, secure e-commerce web sites. You will learn to integrate and implement these technologies by following real-world examples and working sample projects.
Add To Cart
Continue Shopping
Done

图　28-5　每本书有一个详细信息页，该页详细地描述了该书

在详细信息页中，除了"View Cart"链接外，我们还提供了"Add to Cart"链接，用户点击该链接可以选中一个物品。在讨论了如何创建购物车之后，我们再详细介绍它。

下面，让我们来了解这3个脚本。

28.4.1　列出目录

在本项目中使用的第一个脚本——index.php列出了数据库中的所有目录。该脚本如程序清单28-2所示。

程序清单28-2　index.php——该脚本产生本站首页

```
<?php
include('book_sc_fns.php');
//The shopping cart needs sessions,so start one
session_start();
do_html_header("Welcome to Book-O-Rama");
echo"<p>Please choose a category:</p>";
//get categories out of database
$cat_array=get_categories();
//display as links to cat pages
display_categories($cat_array);
//if logged in as admin,show add,delete,edit cat links
if(isset($_SESSION['admin_user'])){
display_button("admin.php","admin-menu","Admin Menu");
}
do_html_footer();
?>
```

在以上脚本的开始处，包含了book_sc_fns.php脚本，该文件包含该应用程序所有的函数库。

然后，我们必须开始一个会话。在购物车有了一个会话后，它才能正常工作。本网站中的每一页面都使用了会话。

index.php脚本还必须包含一些对HTML输出函数的调用，例如，do_html_header()和do_html_footer()（二者都包含在文件output_fns.php中）。该脚本还需要包含一些代码来检查用户是否以管理员的身份登录。如果是，就为它提供一些不同的导航选项。我们将在管理函数部分讨论它。

以上脚本最重要的部分是：

```
//get categories out of database
$cat_array=get_categories();
//display as links to cat pages
display_categories($cat_array);
```

函数get_categories()和display_categories()分别来自函数库book_fns.php和output_fns.php。函数get_categories()将返回系统中的一组目录，我们将该组目录传递到display_categories()函数。下面，我们来看一下get_categories()函数的代码，如程序清单28-3所示。

程序清单28-3　book_fns.php文件中的函数get_categories()——该函数从数据库中取回一个目录列表

```
function get_categories(){
//query database for a list of categories
$conn=db_connect();
$query="select catid,catname from categories";
$result=@$conn->query($query);
if(!$result){
return false;
}
$num_cats=@$result->num_rows;
if($num_cats==0){
return false;
```

```
}
$result=db_result_to_array($result);
return$result;
}
```

可以看到，该函数连接到数据库并获得所有目录的标识符（ID）和名称。我们已经编写和使用了名为db_result_to_array()的函数，该函数包含在db_fns.php中。其代码如程序清单28-4所示。它以一个MySQL搜索结果标识符作为参数，返回一组数字索引的行，其中每一行是一个相关数组。

程序清单28-4　db_fns.php文件中的db_result_to_array()函数——该函数将一个MySQL结果标识符转换为结果数组

```
function db_result_to_array($result){
$res_array=array();
for($count=0;$row=$result->fetch_assoc();$count++){
$res_array[$count]=$row;
}
return$res_array;
}
```

在这个例子中，我们自始至终将这个数组返回给index.php，在该文件中，我们将该数组传递给output_fns.php的函数display_categories()。display_categories()函数以链接形式显示每个目录，此链接指包含该目录所有图书的页面。本函数源代码如程序清单28-5所示。

程序清单28-5　output_fns.php文件中的display_categories()函数——该函数以一列指向目录链接的形式显示一组目录

```
function display_categories($cat_array){
if(!is_array($cat_array)){
echo"<p>No categories currently available</p>";
return;
```

```
}
echo"<ul>";
foreach($cat_array as$row){
$url="show_cat.php?catid=".($row['catid']);$title=$row['catname'];
echo"<li>";
do_html_url($url,$title);
echo"</li>";
}
echo"</ul>";
echo"<hr/>";
}
```

该函数将数据库中的每一个目录转换成一个链接。每个链接导致下一个脚本的执行——show_cat.php——但是都有不同的参数，参数为目录的标识符或catid（这是一个唯一的标识数字，由MySQL生成，用以识别目录）。

这个将要传递给下一个脚本的参数将确定我们需要查看哪一个目录。

28.4.2　列出一个目录中的所有图书

列出一个目录中的所有图书的处理过程与前面类似。实现此功能的函数称为show_cat.php，其源代码如程序清单28-6所示。

程序清单28-6　show_cat.php——该脚本显示特定目录中包含的所有图书

```
<?php
include('book_sc_fns.php');
//The shopping cart needs sessions,so start one
session_start();
$catid=$_GET['catid'];
$name=get_category_name($catid);
do_html_header($name);
//get the book info out from db
$book_array=get_books($catid);
display_books($book_array);
//if logged in as admin,show add,delete book links
if(isset($_SESSION['admin_user'])){
display_button("index.php","continue","Continue Shopping");
display_button("admin.php","admin-menu","Admin Menu");
display_button("edit_category_form.php?catid=".$catid,
"edit-category","Edit Category");
}else{
display_button("index.php","continue-shopping","Continue Shopping");
}
do_html_footer();
?>
```

该脚本在结构上与index页面非常类似，不同的是，在这里，我们获取的是图书信息而不是目录信息。

与列出目录信息示例类似，我们调用session_start()函数开始一个会话，然后调用函数get_category_name()将目录标识符转换为目录名，如下所示：

```
$name=get_category_name($catid);
```

该函数从数据库中查询目录名。其代码如程序清单28-7所示。

程序清单28-7　book_fns.php文件中的get_category_name()函数——该函数将一个目录标识符转换为一个目录名

```
function get_category_name($catid){
//query database for the name for a category id
$conn=db_connect();
$query="select catname from categories
where catid='".$catid."'";
$result=@$conn->query($query);
if(!$result){
return false;
}
$num_cats=@$result->num_rows;
if($num_cats==0){
return false;
}
$row=$result->fetch_object();
return$row->catname;
}
```

获得目录名称后，我们可以完成HTML标题并进而从数据库中获取和列出该选中目录的图书。如下所示：

```
$book_array=get_books($catid);
display_books($book_array);
```

函数get_books()和display_books()分别与函数get_categories()和display_categories()类似。因此，在这里我们将不再详细讨论。它们唯一的不同是前者从books表中获取信息，而后者是从categories表中获取。

display_books()函数通过show_book.php脚本为目录中的每一本书提供了链接。而且，每个链接后面都有一个参数。在本项目中，该参数是当前图书的ISBN。

在show_cat.php脚本的末尾，可以看到，如果是管理员登录的话，有一些函数将会显示一些附加的功能。我们将在管理函数部分讨论这些。

28.4.3　显示图书详细信息

show_book.php脚本将ISBN作为参数，取回并显示该书的详细信息。该脚本的代码如程序清单28-8所示。

程序清单28-8　show_book.php——该脚本显示特定图书的详细信息

```
<?php
include('book_sc_fns.php');
//The shopping cart needs sessions,so start one
session_start();
$isbn=$_GET['isbn'];
//get this book out of database
$book=get_book_details($isbn);
do_html_header($book['title']);
display_book_details($book);
//set url for"continue button"
$target="index.php";
if($book['catid']){
$target="show_cat.php?catid=".$book['catid'];
}
//if logged in as admin,show edit book links
if(check_admin_user()){
display_button("edit_book_form.php?isbn=".$isbn,"edit-item","Edit Item");
display_button("admin.php","admin-menu","Admin Menu");
display_button($target,"continue","Continue");
}else{
display_button("show_cart.php?new=".$isbn,"add-to-cart",
"Add".$book['title']."To My Shopping Cart");
display_button($target,"continue-shopping","Continue Shopping");
}
do_html_footer();
?>
```

对于这段脚本，我们还是做了与前两个脚本非常相似的操作。首先，我们将开始一个会话，然后使用如下语句：

```
$book=get_book_details($isbn);
```

从数据库中获取图书的信息，再使用如下语句：

```
display_book_details($book);
```

以HTML形式输出数据。

在这里，需要注意的是，display_book_details()函数将为每本书寻找一个图像，例如，images/".$book['isbn'].".jpg,。这里，该文件名称是该图书的ISBN号加上".jpg"的文件扩展名。如果该图像文件不存在，就不显示任何图像。脚本其余的部分将建立一个导航。普通用户将会有"Continue shopping"的选项，该选项将他们带回到目录页；此外还有"Add to Cart"选项，这将允许用户将图书添加到购物车。如果用户以管理员的身份登录，选项将有所不同，我们将在介绍管理部分时详细介绍它。

以上代码只是完成了目录系统的基本功能。下面，让我们继续讨论实现购物车功能。

28.5 实现购物车

购物车功能都是围绕名为cart的会话变量展开的。该变量是一个相关数组，它以ISBN作为主键，以图书的数量作为值。例如，如果在购物车中添加了一本图书，那么该数组将包含：

```
0672329166=>1
```

也就是说，这是ISBN为0672329166的一本书。当我们将该书添加到购物车的时候，此信息也加到了该数组。当浏览购物车的时候，就是使用cart数组查看数据库中存储该物品的详细信息。

我们还使用了另外两个会话变量来控制标题栏的显示，该标题栏显示了全部物品数和总价格。这两个变量分别是items和total_price。

28.5.1 使用show_cart.php脚本

通过了解show_cart.php脚本，可以明白购物车代码是如何实现其功能的。如果我们点击"View Cart"或"Add to Cart"链接的话，show_cart.php脚本将显示我们要访问的页面。如果不使用任何参数来调用show_cart.php，将看到购物车的内容；如果用一个ISBN作为参数，该ISBN对应的物品被将添加到购物车中。

要完全理解这些，请参阅图28-6。

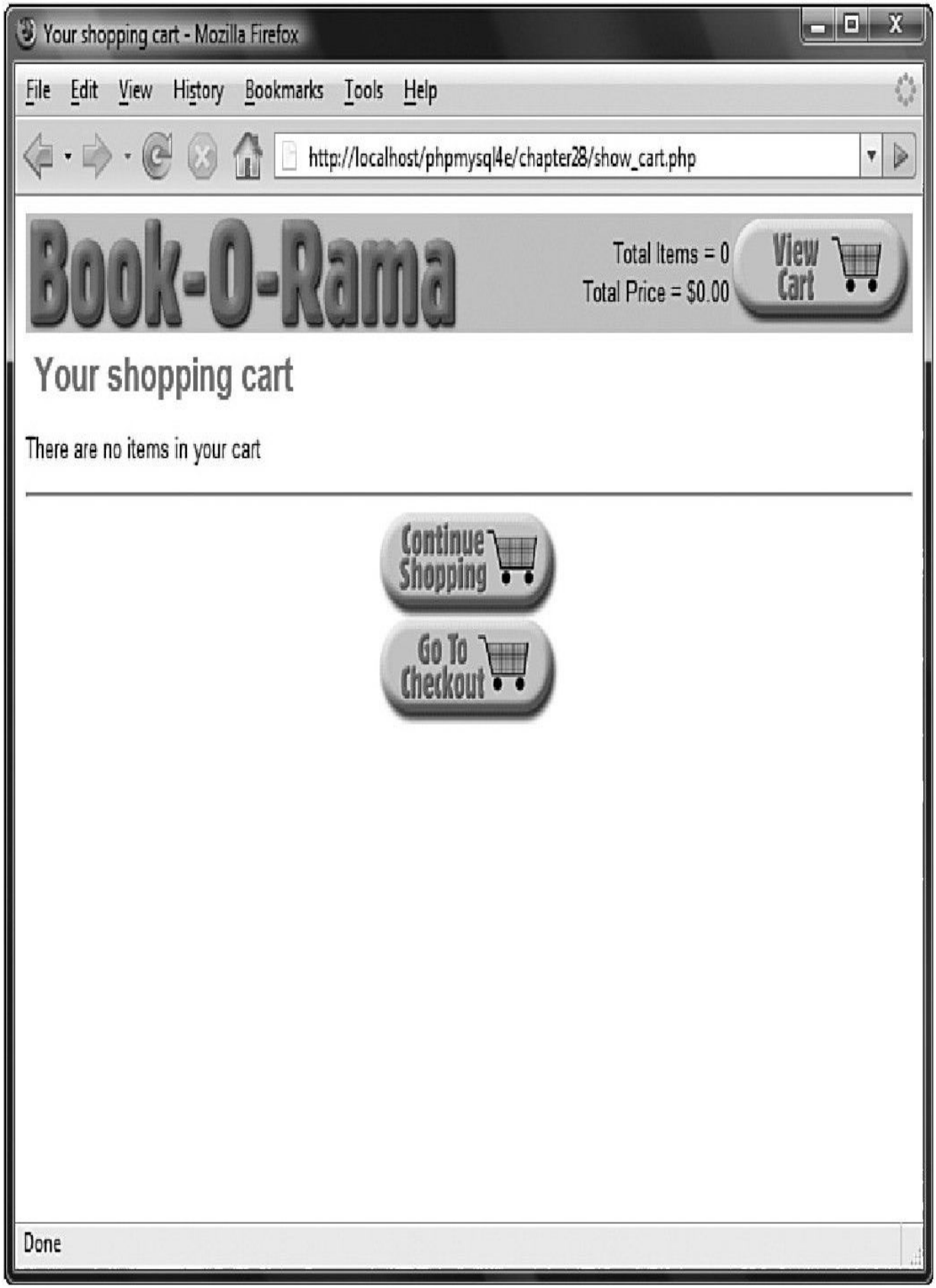

Your shopping cart - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/phpmysql4e/chapter28/show_cart.php
Book-O-Rama
Total Items = 0
Total Price = $0.00
View Cart
Your shopping cart
There are no items in your cart
Continue Shopping
Go To Checkout
Done

图　28-6　不使用参数，show_cart.php脚本只显示购物车的内容

在这个例子中，我们点击了"View Cart"链接，此时购物车为空；也就是说，我们还没有选中任何要买的物品。

图28-7进一步显示了用户的购物车的记录，此时我们已经选中了两本要买的书。在这种情况下，我们是通过点击该书在show_book.php页面上的"Add to Cart"链接而进入本页面的，该书为《PHP和MySQL Web开发》。如果仔细查看URL地址栏，可以看到这次我们使用了一个参数来调用该脚本。该参数为new，其值为067232976X——即刚刚添加到购物车的图书的ISBN。

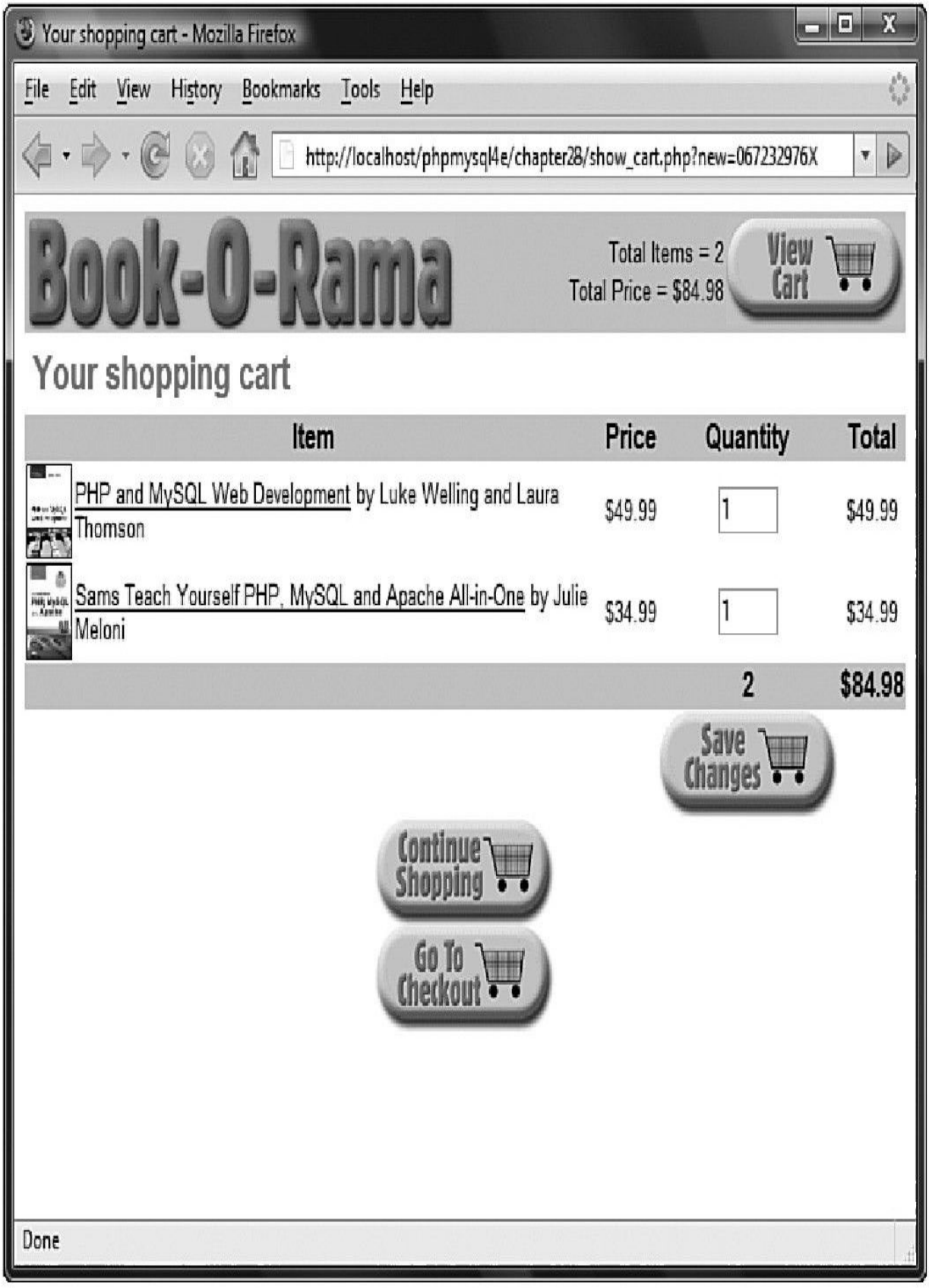

Your shopping cart - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/phpmysql4e/chapter28/show_cart.php?new=067232976X
Book-O-Rama
Total Items = 2
Total Price = $84.98
View Cart
Your shopping cart
Item
Price
Quantity
Total
PHP and MySQL Web Development by Luke Welling and Laura Thomson
$49.99
1
$49.99
Sams Teach Yourself PHP, MySQL and Apache All-in-One by Julie Meloni
$34.99
1
$34.99
2
$84.98
Save Changes
Continue Shopping
Go To Checkout
Done

图　28-7　带有参数new的show_cart.php脚本将添加一个物品到购物车

从这个页面，可以看到我们已经多了两个其他选项。一个是"Save Changes"按钮，它可以用来修改购物车中物品的数量。要修改数量，我们可直接改变物品的数量并点击"Save Changes"按钮。它实际上是一个提交按钮，可以将我们带到show_cart.php脚本以更新购物车。

除此之外，该页面中还有一个"Go To Checkout"按钮，当用户准备离开的时候，可以点击此按钮。稍后，我们将详细介绍它。

从现在开始，我们来了解show_cart.php脚本的代码。该脚本如程序清单28-9所示。

程序清单28-9　show_cart.php——该脚本控制购物车

```
<?php
include('book_sc_fns.php');
//The shopping cart needs sessions,so start one
session_start();
@$new=$_GET['new'];
if($new){
//new item selected
if(!isset($_SESSION['cart'])){
$_SESSION['cart']=array();
$_SESSION['items']=0;
$_SESSION['total_price']='0.00';
}
if(isset($_SESSION['cart'][$new])){
$_SESSION['cart'][$new]++;
}else{
$_SESSION['cart'][$new]=1;
}
$_SESSION['total_price']=calculate_price($_SESSION['cart']);
$_SESSION['items']=calculate_items($_SESSION['cart']);
}
if(isset($_POST['save'])){
```

```
foreach($_SESSION['cart']as$isbn=>$qty){
if($_POST[$isbn]=='0'){
unset($_SESSION['cart'][$isbn]);
}else{
$_SESSION['cart'][$isbn]=$_POST[$isbn];
}
}
$_SESSION['total_price']=calculate_price($_SESSION['cart']);
$_SESSION['items']=calculate_items($_SESSION['cart']);
}
do_html_header("Your shopping cart");
if(($_SESSION['cart'])&&(array_count_values($_SESSION['cart']))){
display_cart($_SESSION['cart']);
}else{
echo"<p>There are no items in your cart</p><hr/>";
}
$target="index.php";
//if we have just added an item to the cart,continue shopping in that category
if($new){
$details=get_book_details($new);
if($details['catid']){
$target="show_cat.php?catid=".$details['catid'];
}
}
display_button($target,"continue-shopping","Continue Shopping");
//use this if SSL is set up
//$path=$_SERVER['PHP_SELF'];
//$server=$_SERVER['SERVER_NAME'];
//$path=str_replace('show_cart.php','',$path);
//display_button("https://".$server.$path."checkout.php",
//"go-to-checkout","Go To Checkout");
//if no SSL use below code
display_button("checkout.php","go-to-checkout","Go To Checkout");
do_html_footer();
?>
```

该脚本主要由3个部分组成：显示购物车、添加物品到购物车以及保存购物车的修改结果。接下来，我们将在下面3个小节里分别讨论它们。

28.5.2 浏览购物车

不论我们从哪个页面进入，都需要显示购物车的内容。在这个最基本的例子中，当用户点击"View Cart"按钮后，系统将只执行如下所示的代码：

```
if(($_SESSION['cart'])&&(array_count_values($_SESSION['cart']))){
display_cart($_SESSION['cart']);
}else{
echo"<p>There are no items in your cart</p><hr/>";
}
```

可以从这段代码看出，如果购物车里有一些物品，我们就调用display_cart()函数。如果购物车是空的，会给用户发送一条消息，告诉他们购物车为空。

display_cart()函数只以可读的HTML格式打印出购物车的内容，这点可从图28-6和图28-7看出。尽管函数的代码可以在output_fns.php中找到，但我们还是给出了output_fns.php的源代码，如程序清单28-10所示。尽管该函数只是一个显示函数，但是它比较复杂，因此在这里我们讨论一下。

程序清单28-10　output_fns.php文件中的display_cart()函数——该函数将格式化并打印购物车的内容

```
function display_cart($cart,$change=true,$images=1){
//display items in shopping cart
//optionally allow changes(true or false)
//optionally include images(1-yes,0-no)
echo"<table border=\"0\"width=\"100%\"cellspacing=\"0\">
<form action=\"show_cart.php\"method=\"post\">
<tr><th colspan=\"".(1+$images)."\"bgcolor=\"#cccccc\">Item</th>
<th bgcolor=\"#cccccc\">Price</th>
<th bgcolor=\"#cccccc\">Quantity</th>
```

```
<th bgcolor=\"#cccccc\">Total</th>
</tr>";
//display each item as a table row
foreach($cart as$isbn=>$qty){
$book=get_book_details($isbn);
echo"<tr>";
if($images==true){
echo"<td align=\"left\">";
if(file_exists("images/".$isbn.".jpg")){
$size=GetImageSize("images/".$isbn.".jpg");
if(($size[0]>0)&&($size[1]>0)){
echo"<img src=\"images/".$isbn.".jpg\"
style=\"border:1px solid black\"
width=\"".($size[0]/3)."\"
height=\"".($size[1]/3)."\"/>";
}
}else{
echo" ";
}
echo"</td>";
}
echo"<td align=\"left\">
<a href=\"show_book.php?isbn=".$isbn."\">".$book['title']."</a>
by".$book['author']."</td>
<td align=\"center\">\$".number_format($book['price'],2)."</td>
<td align=\"center\">";
//if we allow changes,quantities are in text boxes
if($change==true){
echo"<input type=\"text\"name=\"".$isbn."\"value=\"".$qty."\"
size=\"3\">";
}else{
echo$qty;
}
echo"</td>
<td align=\"center\">\$".number_format($book['price']*$qty,2)."</td>
</tr>\n";
```

```
}
//display total row
echo"<tr>
<th colspan=\"".(2+$images)."\"bgcolor=\"#cccccc\"> </td>
<th align=\"center\"bgcolor=\"#cccccc\">".$_SESSION['items']."</th>
<th align=\"center\"bgcolor=\"#cccccc\">
\$".number_format($_SESSION['total_price'],2)."
</th>
</tr>";
//display save change button
if($change==true){
echo"<tr>
<td colspan=\"".(2+$images)."\"> </td>
<td align=\"center\">
<input type=\"hidden\"name=\"save\"value=\"true\"/>
<input type=\"image\"src=\"images/save-changes.gif\"
border=\"0\"alt=\"Save Changes\"/>
</td>
<td> </td>
</tr>";
}
echo"</form></table>";
}
```

该函数的基本流程如下所示：

1.对购物车中所有物品执行循环，将每个物品的ISBN传递给get_book_details()函数，这样，可以总结每本书的详细信息。

2.如果有图片存在，可以为每本书提供一个图片。我们使用HTML图像的高度标记和宽度标记将图片改得小了一点。这意味着这些图片可能有一些扭曲，但是因为它们很小，扭曲一点不会有太大问题。（如果看起来还是不舒服，可以使用第22章“创建图像”中介绍的gd库来修改它的大小，或者手动为每个产品生成一个不同大小的图片。）

3.让每个购物车包含一个指向适当图书的链接，也就是以ISBN作为show_book.php的参数显示图书。

4.如果以change作参数调用函数，而且该change设置为true（或不设置——其默认值是true），则看到的是以表单形式显示的购物车中所有物品的数量，点击"Save Changes"按钮结束（在结账之后再使用该函数的时候，我们希望用户不能修改订单）。

该函数中没有特别复杂的逻辑，但是它完成了许多操作，因此，仔细阅读它，会觉得它非常有用。

28.5.3 将物品添加到购物车

如果用户点击"Add to Cart"按钮进入show_cart.php页面，在显示其购物车内容之前，我们要做一些工作。特别地，需要将适当的物品添加到购物车中，如下所示。

首先，如果该用户此前没有在购物车中添加任何物品，那么该用户没有一个购物车，需要为其创建一个购物车：

```
if(!isset($_SESSION['cart'])){
$_SESSION['cart']=array();
$_SESSION['items']=0;
$_SESSION['total_price']='0.00';
}
```

初始状态下，购物车是空的。

其次，建立了一个购物车后，可以将物品添加到购物车内：

```
if(isset($_SESSION['cart'][$new])){
$_SESSION['cart'][$new]++;
}else{
$_SESSION['cart'][$new]=1;
}
```

在这里，我们检查了该物品是否已经在购物车中存在，如果是，我们将该物品的数量增1，否则，添加新物品到购物车。

再次，我们必须计算购物车中所有物品的总价格和数量。要完成这些操作，我们使用了calculate_price()函数和calculate_items()函数，如下所示：

```
$_SESSION['total_price']=calculate_price($_SESSION['cart']);
```

```
$_SESSION['items']=calculate_items($_SESSION['cart']);
```

这些函数位于book_fns.php函数库中。其代码如程序清单28-11和程序清单28-12所示。

程序清单28-11　book_fns.php文件中的calculate_price()函数——计算和返回购物车中物品的总价格

```
function calculate_price($cart){
//sum total price for all items in shopping cart
$price=0.0;
if(is_array($cart)){
$conn=db_connect();
foreach($cart as$isbn=>$qty){
$query="select price from books where isbn='".$isbn."'";
$result=$conn->query($query);
if($result){
$item=$result->fetch_object();
$item_price=$item->price;
$price+=$item_price*$qty;
}
}
}
return$price;
}
```

可以看到，calculate_price()函数工作的时候查询了数据库中每个物品的价格。该操作可能会有些慢，因此如果没有必要，应该避免这样的计算。我们将保存此价格（还有物品总数）到会话变量中，当购物车改变的时候才重新计算。

程序清单28-12　book_fns.php文件中的calculate_items()函数——该函数计算并返回购物车中物品的总数

```
function calculate_items($cart){
//sum total items in shopping cart
$items=0;
if(is_array($cart)){
foreach($cart as$isbn=>$qty){
$items+=$qty;
}
}
return$items;
}
```

---

calculate_items()函数相当简单；它只是扫描了购物车，使用array_sum()函数将每个物品的数量加起来得到总物品数量。如果还没有数组（或者购物车为空），它将返回0。

28.5.4 保存更新后的购物车

如果我们通过点击"Save Changes"按钮进入show_cart.php脚本，那么它的处理过程与前面有些不同。在这种情况下，我们是通过提交一个表单进入的。如果仔细查看这些代码，可以发现"Save Changes"按钮是一个表单的提交按钮。该表单包含了隐含变量save。如果该变量被赋了值，我们就可以知道是从"Save Changes"按钮的点击事件进入此页面的。这就意味着，该用户可能已经编辑了购物车中物品的数量值，我们需要更新它们。

在 output_fns.php 脚本中，如果查看 display_cart() 函数 "Save Changes"表单部分的文本框，你将发现它们的命名是与ISBN相关联的，如下所示：

```
echo"<input type=\"text\"name=\"".$isbn."\"value=\"".$qty."\"size=\"3\">";
```

现在，让我们来看看保存修改部分的脚本：

```
if(isset($_POST['save'])){
foreach($_SESSION['cart']as$isbn=>$qty){
if($_POST[$isbn]=='0'){
unset($_SESSION['cart'][$isbn]);
}else{
$_SESSION['cart'][$isbn]=$_POST[$isbn];
}
}
$_SESSION['total_price']=calculate_price($_SESSION['cart']);
$_SESSION['items']=calculate_items($_SESSION['cart']);
}
```

可以看出，我们通过购物车完成了工作，对购物车中的每一个isbn，检查了该变量名称的POST变量。这些都是"Save Changes"表单的表单域。

如果将任何一个域设置为0，我们将使用unset()函数将购物车中该物品完全删除。否则，就更新该购物车，使之与该表单域匹配，如下所示：

```
if($_POST[$isbn]=='0'){
unset($_SESSION['cart'][$isbn]);
}else{
$_SESSION['cart'][$isbn]=$_POST[$isbn];
}
```

在完成购物车更新之后，我们又调用calculate_price()函数和calculate_items()函数计算出会话变量total_price和items的新值。

28.5.5　打印标题栏摘要

网站中每一页的标题栏都简要总结了购物车中的物品。这是通过打印会话变量total_price和items的值来实现的。函数do_html_header()能够完成此功能。

当用户第一次访问show_cart.php页面的时候，我们就注册了这些变量。我们还需要一些代码来解决用户未访问该页的情形。这个逻辑同样包含在do_html_header()函数中，如下代码所示：

```
if(!$_SESSION['items']){
$_SESSION['items']='0';
}
if(!$_SESSION['total_price']){
$_SESSION['total_price']='0.00';
}
```

28.5.6 结账

当用户点击购物车的"Go to Checkout"按钮时，将触发checkout.php脚本。该结账脚本及其后面的脚本必须通过SSL访问到，但是，我们给出的示例应用程序并不要求这么做（要了解更多关于SSL的信息，请参阅第18章“通过PHP和MySQL实现安全事务”）。

结账页面如图28-8所示。

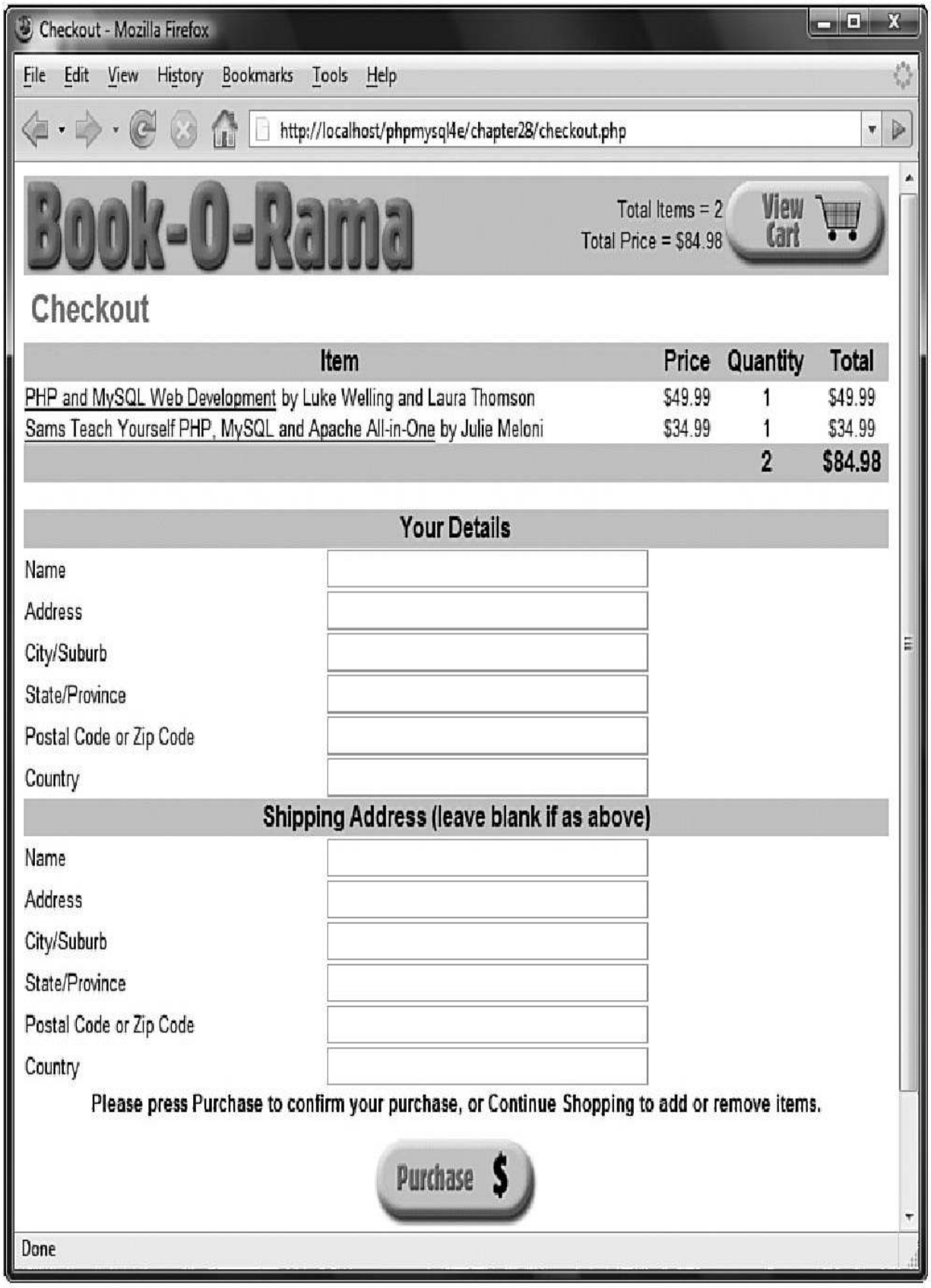

Checkout - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/phpmysql4e/chapter28/checkout.php
Book-O-Rama
Total Items = 2
Total Price = $84.98
View Cart
Checkout
Item
Price
Quantity
Total
PHP and MySQL Web Development by Luke Welling and Laura Thomson
$49.99
1
$49.99
Sams Teach Yourself PHP, MySQL and Apache All-in-One by Julie Meloni
$34.99
1
$34.99
2
$84.98
Your Details
Name
Address
City/Suburb
State/Province
Postal Code or Zip Code
Country
Shipping Address (leave blank if as above)
Name
Address
City/Suburb
State/Province
Postal Code or Zip Code
Country
Please press Purchase to confirm your purchase, or Continue Shopping to add or remove items.
Purchase $
Done

图　28-8　checkout.php脚本获取顾客的详细信息

结账脚本要求顾客输入地址（如果运送地址与该地址不同，需要输入运送地址）。该脚本非常简单，其代码如程序清单28-13所示。

程序清单28-13　checkout.php——该脚本获取用户的详细信息

```
<?php
//include our function set
include('book_sc_fns.php');
//The shopping cart needs sessions,so start one
session_start();
do_html_header("Checkout");
if(($_SESSION['cart'])&&(array_count_values($_SESSION['cart']))){
display_cart($_SESSION['cart'],false,0);
display_checkout_form();
}else{
echo"<p>There are no items in your cart</p>";
}
display_button("show_cart.php","continue-shopping","Continue Shopping");
do_html_footer();
?>
```

在以上脚本中，并没有什么奇怪的事情发生。如果购物车是空的，脚本将通知该用户；否则，它将显示如图28-8所示的表单。

如果用户继续点击该表格下面的"Purchase"按钮，将进入purchase.php脚本。在图28-9中，可以看到该脚本的输出。

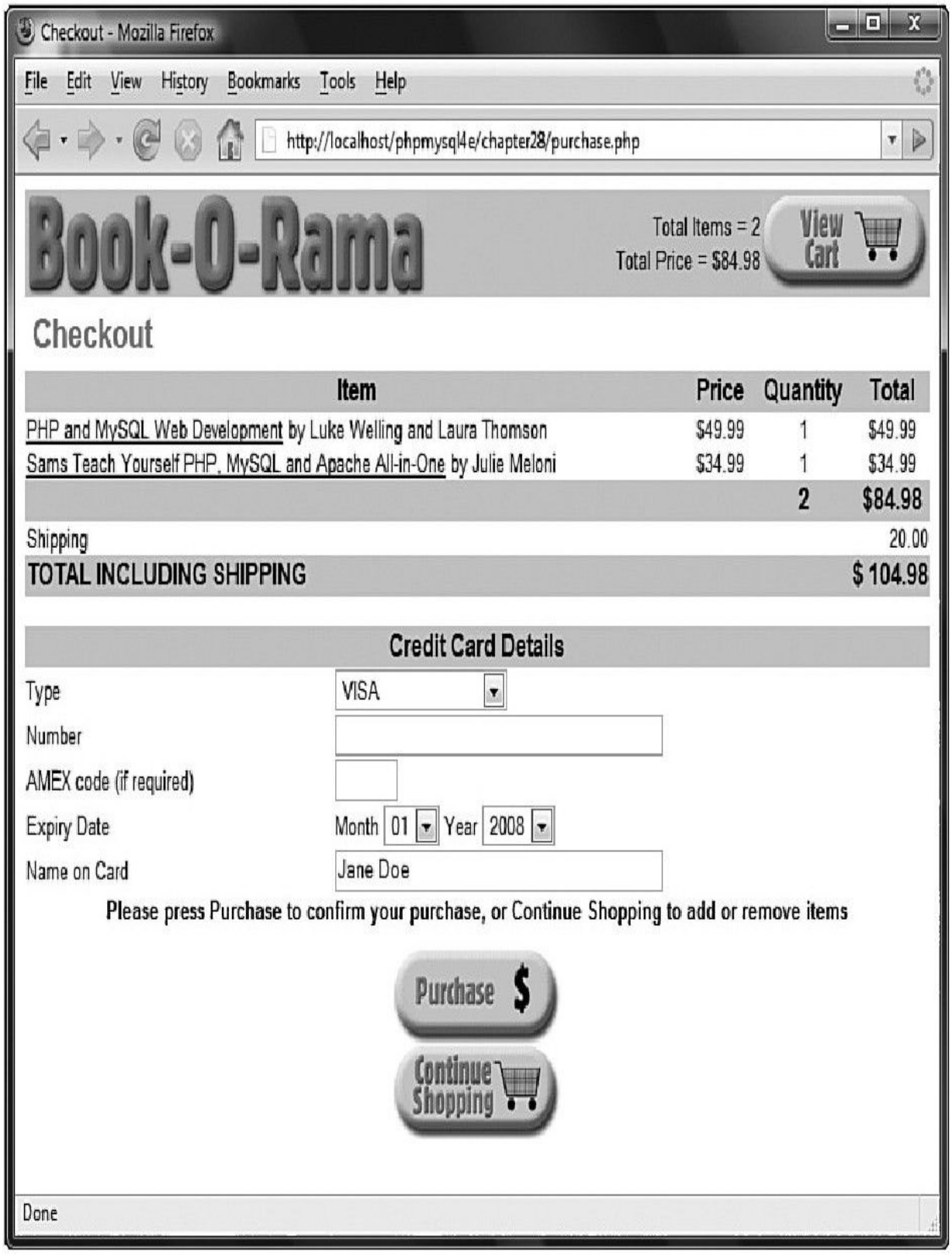

Checkout - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/phpmysql4e/chapter28/purchase.php
Book-O-Rama
Total Items = 2
Total Price = $84.98
View Cart
Checkout
Item
Price
Quantity
Total
PHP and MySQL Web Development by Luke Welling and Laura Thomson
$49.99
1
$49.99
Sams Teach Yourself PHP, MySQL and Apache All-in-One by Julie Meloni
$34.99
1
$34.99
2
$84.98
Shipping
20.00
TOTAL INCLUDING SHIPPING
$ 104.98
Credit Card Details
Type
VISA
Number
AMEX code (if required)
Expiry Date
Month 01 Year 2008
Name on Card
Jane Doe
Please press Purchase to confirm your purchase, or Continue Shopping to add or remove items
Purchase $
Continue Shopping
Done

图　28-9　purchase.php脚本计算客户购买的商品数量和最终的订单价格，并且获得客户的付款详细信息

以上脚本代码比checkout.php脚本逻辑稍微复杂一些，如程序清单28-14所示。

程序清单28-14　purchase.php——该脚本将用户订单的详细信息保存到数据库中并获取付款细节

```
<?php
include('book_sc_fns.php');
//The shopping cart needs sessions,so start one
session_start();
do_html_header("Checkout");
//create short variable names
$name=$_POST['name'];
$address=$_POST['address'];
$city=$_POST['city'];
$zip=$_POST['zip'];
$country=$_POST['country'];
//if filled out
if(($_SESSION['cart'])&&($name)&&($address)&&($city)
&&($zip)&&($country)){
//able to insert into database
if(insert_order($_POST)!=false){
//display cart,not allowing changes and without pictures
display_cart($_SESSION['cart'],false,0);
display_shipping(calculate_shipping_cost());
//get credit card details
display_card_form($name);
display_button("show_cart.php","continue-shopping","Continue Shopping");
}else{
echo"<p>Could not store data,please try again.</p>";
display_button('checkout.php','back','Back');
}
}else{
```

```
echo"<p>You did not fill in all the fields,please try again.</p><hr/>";
display_button('checkout.php','back','Back');
}
do_html_footer();
?>
```

以上代码的逻辑简单明了：它首先检查用户是否填好了表单，并调用insert_order()函数，这是一个简单的函数，将用户填写的详细信息插入数据库。其代码如程序清单28-15所示。

该函数相当长，因为我们需要插入用户所有的细节，包括订单细节以及他们要买的每一本书的细节。

程序清单28-15　order_fns.php文件中的insert_order()函数——该函数将所有的用户订单详细信息插入到数据库中

```
<?php
function process_card($card_details){
//connect to payment gateway or
//use gpg to encrypt and mail or
//store in DB if you really want to
return true;
}
function insert_order($order_details){
//extract order_details out as variables
extract($order_details);
//set shipping address same as address
if((!$ship_name)&&(!$ship_address)&&(!$ship_city)
&&(!$ship_state)&&(!$ship_zip)&&(!$ship_country)){
$ship_name=$name;
$ship_address=$address;
$ship_city=$city;
$ship_state=$state;
$ship_zip=$zip;
$ship_country=$country;
```

```
}
$conn=db_connect();
//we want to insert the order as a transaction
//start one by turning off autocommit
$conn->autocommit(FALSE);
//insert customer address
$query="select customerid from customers where
name='".$name."'and address='".$address."'
and city='".$city."'and state='".$state."'
and zip='".$zip."'and country='".$country."'";
$result=$conn->query($query);
if($result->num_rows>0){
$customer=$result->fetch_object();
$customerid=$customer->customerid;
}else{
$query="insert into customers values
('','".$name."','".$address."','".$city."',
'".$state."','".$zip."','".$country."')";
$result=$conn->query($query);
if(!$result){
return false;
}
}
$customerid=$conn->insert_id;
$date=date("Y-m-d");
$query="insert into orders values
('','".$customerid."','".$_SESSION['total_price']."',
'".$date."','".PARTIAL."','".$ship_name."',
'".$ship_address."','".$ship_city."',
'".$ship_state."','".$ship_zip."',
'".$ship_country."')";
$result=$conn->query($query);
if(!$result){
return false;
}
$query="select orderid from orders where
```

```
customerid='".$customerid."'and
amount>(".$_SESSION['total_price']."-.001)and
amount<(".$_SESSION['total_price']."+.001)and
date='".$date."'and
order_status='PARTIAL'and
ship_name='".$ship_name."'and
ship_address='".$ship_address."'and
ship_city='".$ship_city."'and
ship_state='".$ship_state."'and
ship_zip='".$ship_zip."'and
ship_country='".$ship_country."'";
$result=$conn->query($query);
if($result->num_rows>0){
$order=$result->fetch_object();
$orderid=$order->orderid;
}else{
return false;
}
//insert each book
foreach($_SESSION['cart']as$isbn=>$quantity){
$detail=get_book_details($isbn);
$query="delete from order_items where
orderid='".$orderid."'and isbn='".$isbn."'";
$result=$conn->query($query);
$query="insert into order_items values
('".$orderid."','".$isbn."',".$detail['price'].",$quantity)";
$result=$conn->query($query);
if(!$result){
return false;
}
}
//end transaction
$conn->commit();
$conn->autocommit(TRUE);
return$orderid;
}
```

```
?>
```

插入操作的不同部分被包括在一个事务中，它以如下语句开始：

```
$conn->autocommit(FALSE);
```

以如下语句结束：

```
$conn->commit();
$conn->autocommit(TRUE);
```

这是应用程序中唯一需要使用事务的地方。如何避免在其他情况下也使用它呢？请查看db_connect()函数的代码，如下所示：

```
function db_connect(){
$result=new mysqli('localhost','book_sc','password','book_sc');
if(!$result){
return false;
}
$result->autocommit(TRUE);
return$result;
}
```

很明显，以上代码与在其他章节中使用的相同函数名称的代码不同。在创建了到MySQL的连接后，必须开启自动提交模式。

正如我们前面介绍的，这可以确保每一个SQL语句将自动被提交。当需要使用一个多语句事务时，必须关闭自动提交模式，执行一系列的插入操作，提交数据，然后再重新启用自动提交模式。

然后我们根据顾客地址计算出运送费用，并通过如下所示代码告诉他们需要的费用：

```
display_shipping(calculate_shipping_cost());
```

在这里，我们使用的calculate_shipping_cost()函数将始终返回$20。当真正建立一个购物网站的时候，必须选择一个送货方式，计算出不同的目的地需要的运送费用，并据此计算出总费用。

接下来，我们将向每个用户显示一个表单，用来获得信用卡详细信息，此过程中，我们调用了output_fns.php函数库中的display_card_form()函数。

28.6 实现付款

当用户点击"Purchase"按钮时，我们将调用process.php脚本来处理付款细节。可以在图28-10中看到成功付款的结果。

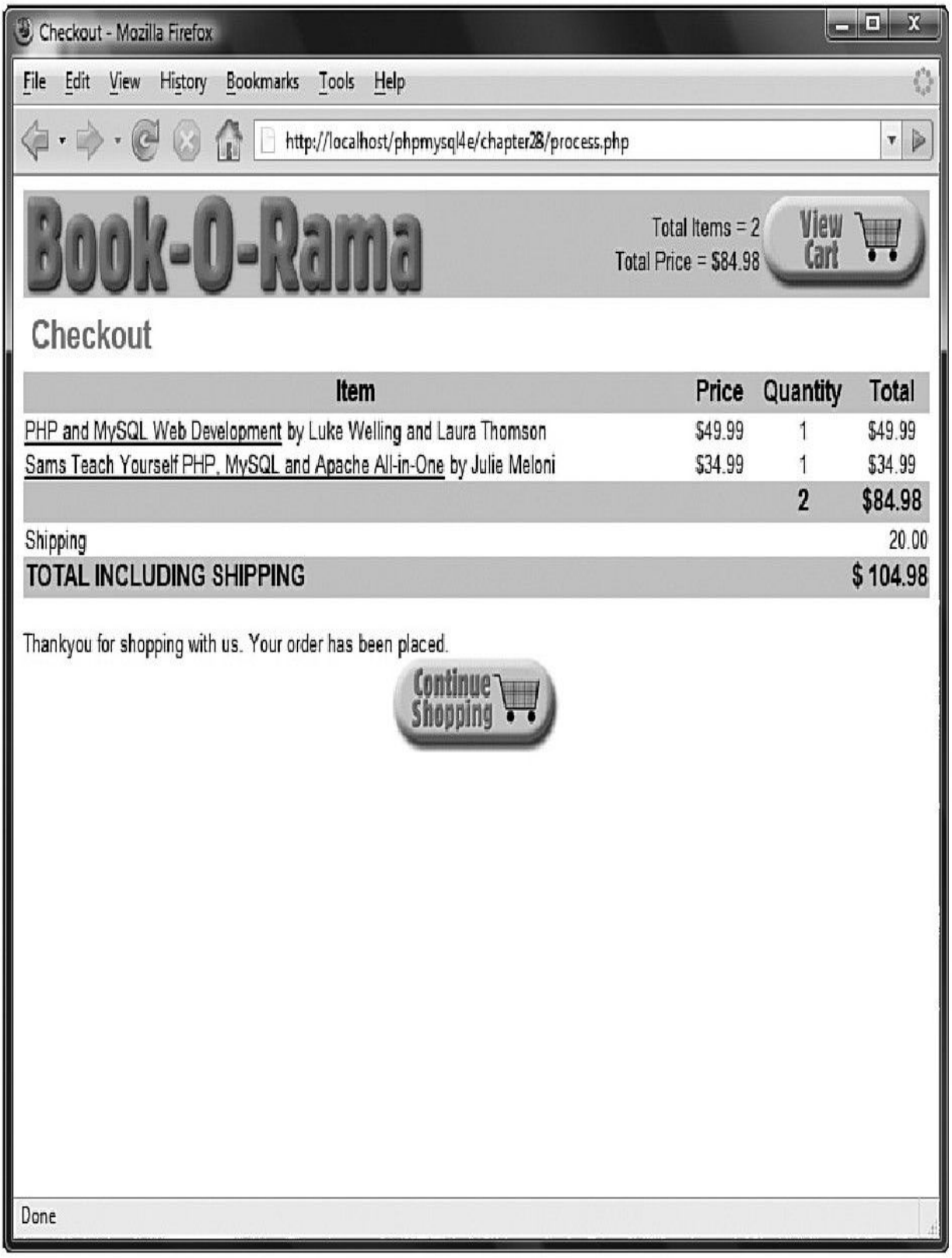


图　28-10　付款操作成功，我们将发送顾客订购的商品

process.php脚本如程序清单28-16所示。

我们将处理用户的信用卡，如果所有操作成功完成，再销毁用户的会话。

我们编写的信用卡处理函数将简单地返回true。在具体实现过程中，可能需要先执行一些验证（检查信用卡的过期日期是否仍然有效以及卡号是否正确），然后再执行真正的付款操作。

程序清单28-16　process.php——该脚本处理顾客的付款并显示处理结果

```
<?php
include('book_sc_fns.php');
//The shopping cart needs sessions,so start one
session_start();
do_html_header('Checkout');
$card_type=$_POST['card_type'];
$card_number=$_POST['card_number'];
$card_month=$_POST['card_month'];
$card_year=$_POST['card_year'];
$card_name=$_POST['card_name'];
if(($_SESSION['cart'])&&($card_type)&&($card_number)&&
($card_month)&&($card_year)&&($card_name)){
//display cart,not allowing changes and without pictures
display_cart($_SESSION['cart'],false,0);
display_shipping(calculate_shipping_cost());
if(process_card($_POST)){
//empty shopping cart
session_destroy();
echo"<p>Thank you for shopping with us.Your order has been placed.</p>";
display_button("index.php","continue-shopping","Continue Shopping");
}else{
echo"<p>Could not process your card.Please contact the card
issuer or try again.</p>";
display_button("purchase.php","back","Back");
```

```
}
}else{
echo"<p>You did not fill in all the fields,please try again.</p><hr/>";
display_button("purchase.php","back","Back");
}
do_html_footer();
?>
```

实际建立站点时，必须决定采用哪种事务清除机制。我们可以：

■与一个事务清理提供商签约。根据所在地的不同，可以选择许多服务商。一些服务商提供实时的数据清理，一些服务商则不是。是否需要实时的数据清理取决于向用户提供的服务。如果提供的是在线服务，可能最希望实时清理；如果发送货物，实时处理就不是很重要了。无论选择哪种方式，这些服务商都承担了保存信用卡号码的责任。

■通过加密邮件将信用卡号码发送给自己，加密算法可以采用PGP或GPG等，这些内容已经在第18章中详细介绍。当接收并解密这些邮件之后，就可以手动处理这些事务。

■在数据库中保存信用卡号。如果没有慎重考虑这样做对系统安全性能的潜在危害，那么我们不推荐采用这种选择。可以参阅第18章的详细介绍，明白为什么这是一个糟糕的主意。

以上就是购物车模块和付款模块。

28.7　实现一个管理界面

我们实现的管理界面非常简单。创建一个网站界面，该页面可以连接数据库，并且执行一些前台的身份验证。这些代码与第27章中用到的代码非常类似。出于完整性考虑，我们还是在这里给出了这些代码，但对它不做具体的介绍。

管理界面需要用户通过login.php文件进行登录，该文件会将用户带入到管理菜单admin.php中。登录界面如图28-11所示。（为了简洁起见，我们没有给出login.php的代码——该文件与第27章中介绍的几乎一样，如果要查看这个文件，可以在附带的文件中找到这个文件。）管理菜单如图28-12所示。

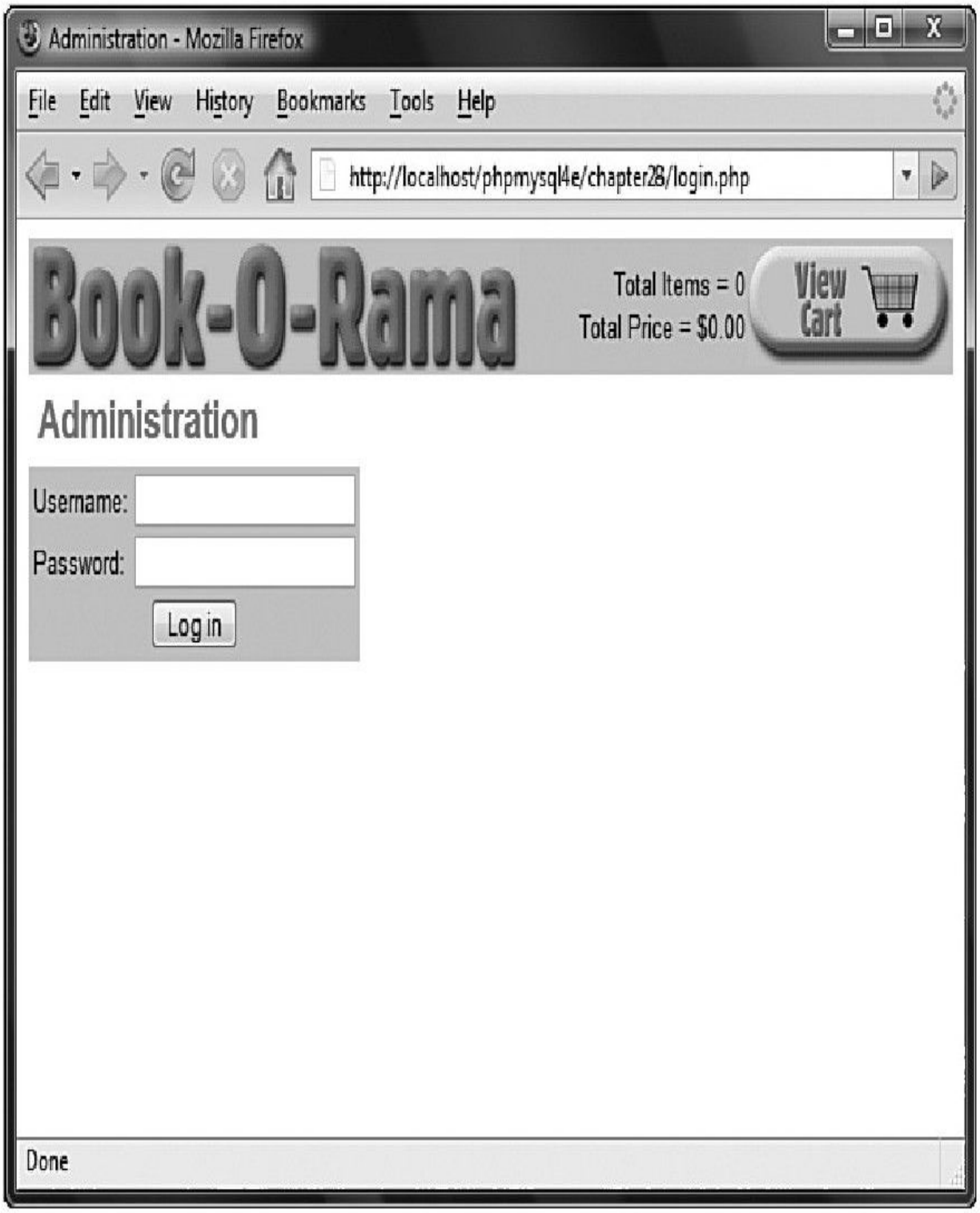


图　28-11　用户通过登录页才能访问管理功能

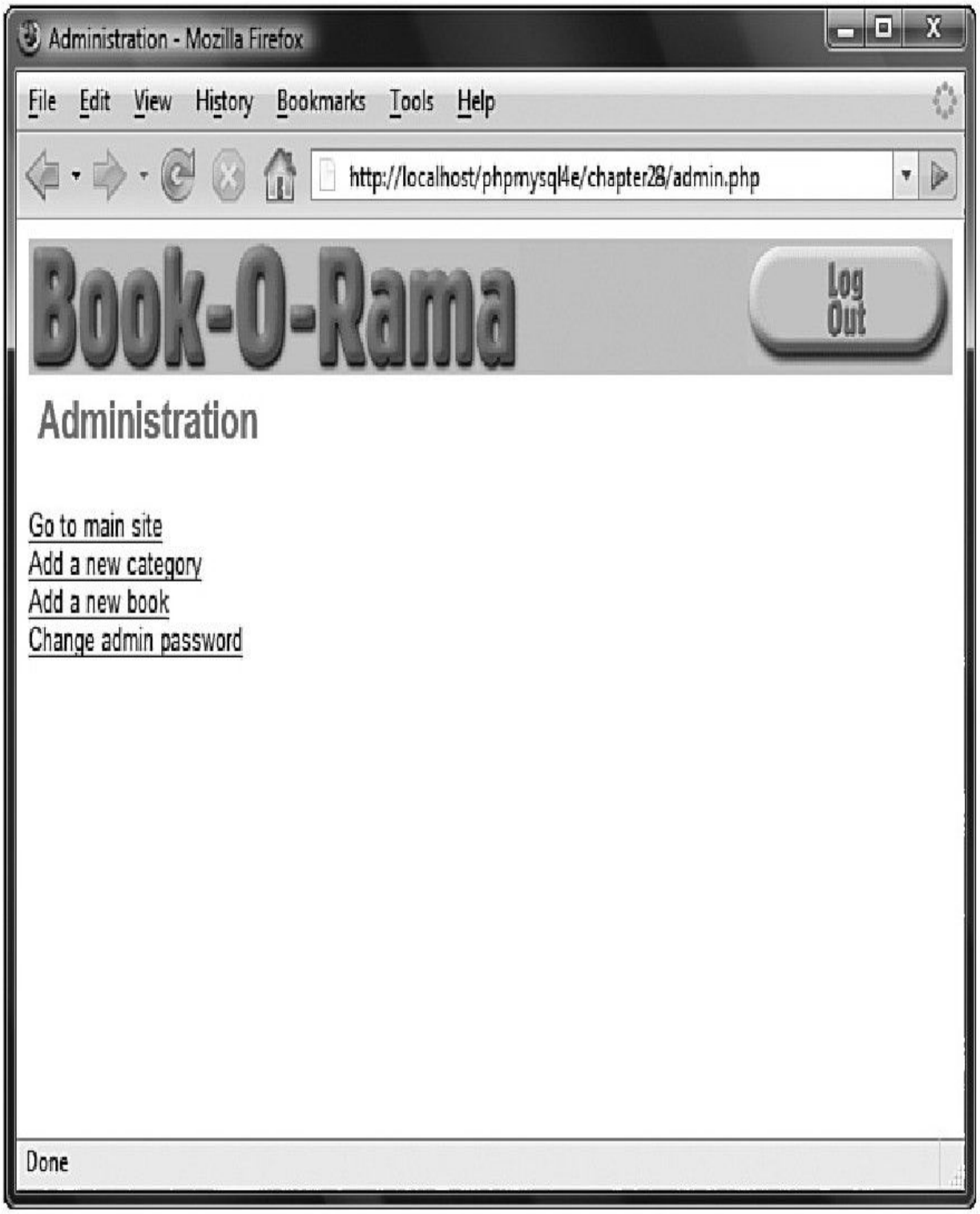


图　28-12　管理菜单允许用户访问管理函数

管理菜单的代码如程序清单28-17所示。

程序清单28-17　admin.php——该脚本将验证管理员身份并允许他访问管理功能

```
<?php
//include function files for this application
require_once('book_sc_fns.php');
session_start();
if(($_POST['username'])&&($_POST['passwd'])){
//they have just tried logging in
$username=$_POST['username'];
$passwd=$_POST['passwd'];
if(login($username,$passwd)){
//if they are in the database register the user id
$_SESSION['admin_user']=$username;
}else{
//unsuccessful login
do_html_header("Problem:");
echo"<p>You could not be logged in.<br/>
You must be logged in to view this page.</p>";
do_html_url('login.php','Login');
do_html_footer();
exit;
}
}
do_html_header("Administration");
if(check_admin_user()){
display_admin_menu();
}else{
echo"<p>You are not authorized to enter the administration area.</p>";
}
do_html_footer();
?>
```

这段代码看起来有点熟悉；它与第27章中的一段脚本类似。管理员到达这个页面之后，可以修改密码或者退出——该代码与第27章中

的代码完全一样，因此在这里不再介绍。

我们在管理员登录之后通过会话变量admin_user和check_admin_user()函数来识别其身份。该函数和其他被管理员脚本调用的函数一样，都可以在函数库admin_fns.php中找到。

如果管理员选择添加一个新的目录或图书，可以根据具体情况进入insert_category_form.php或insert_book_form.php文件。每个脚本都会向管理员提供一个表单，该表单必须由管理员填写。每个表单都由相应的脚本来处理（insert_category.php和insert_book.php），这些脚本将检查表单是否填好并将数据插入到数据库中。在这里，我们只讨论添加图书的脚本，因为添加图书的脚本和添加目录的脚本类似。

insert_book_form.php脚本的输出结果如图28-13所示。

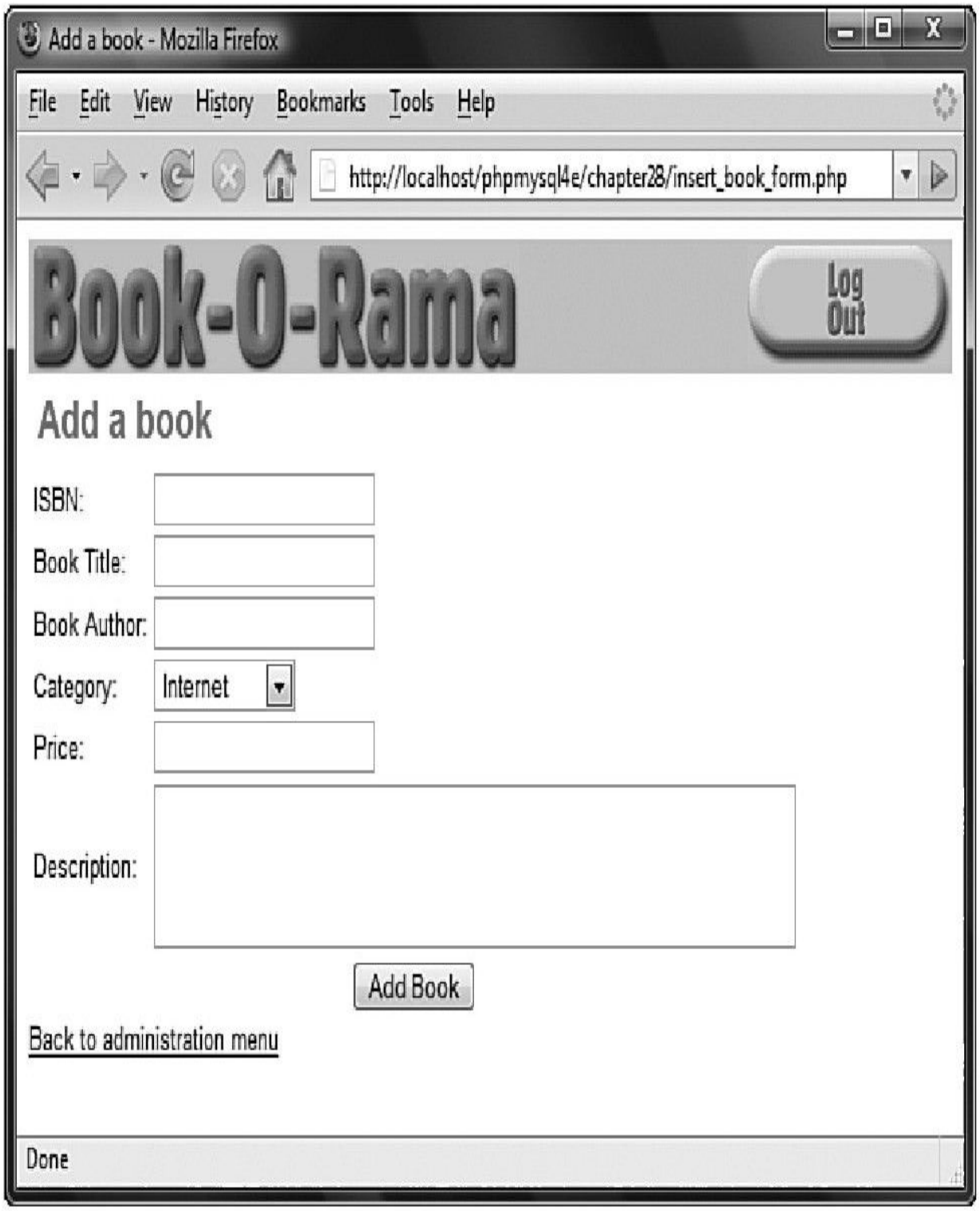


图　28-13　该表单允许管理员输入新书到在线目录

注意图书的目录域是一个HTML的SELECT元素。SELECT的选项来自对get_categories()函数的调用，前面我们已经介绍了该函数。

点击"Add Book"按钮之后，将触发insert_book.php脚本的执行。该脚本的代码如程序清单28-18所示。

程序清单28-18　insert_book.php——该脚本将验证新书数据并将它添加到数据库

```
<?php
//include function files for this application
require_once('book_sc_fns.php');
session_start();
do_html_header("Adding a book");
if(check_admin_user()){
if(filled_out($_POST)){
$isbn=$_POST['isbn'];
$title=$_POST['title'];
$author=$_POST['author'];
$catid=$_POST['catid'];
$price=$_POST['price'];
$description=$_POST['description'];
if(insert_book($isbn,$title,$author,$catid,$price,$description)){
echo"<p>Book<em>".stripslashes($title)."</em>was added to the
database.</p>";
}else{
echo"<p>Book<em>".stripslashes($title)."</em>could not be
added to the database.</p>";
}
}else{
echo"<p>You have not filled out the form.Please try again.</p>";
}
do_html_url("admin.php","Back to administration menu");
}else{
echo"<p>You are not authorised to view this page.</p>";
}
```

```
do_html_footer();
?>
```

可以看到，该脚本调用了函数insert_book()。该函数和其他管理脚本调用的函数一样，都可以在函数库admin_fns.php中找到。

除了添加新目录和新书，管理员还可以编辑和删除它们。我们已经重用了尽可能多的脚本来实现这些功能。当管理员点击管理菜单页面中的"Go to Main Site"链接时，将回到index.php中的目录索引，并且与普通用户一样使用同样的脚本接受该索引的导航。

然而，管理导航之间不完全相同：管理员将看到不同的选项，这些选项是根据他们已经注册的会话变量admin_user来确定的。例如，如果我们查看一下本章前面介绍的show_book.php页，将看到一些不同的菜单选项，如图28-14所示。

PHP and MySQL Web Development - Mozilla Firefox

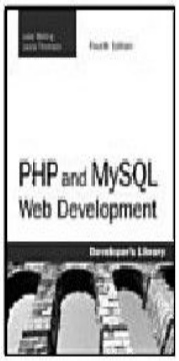
File Edit View History Bookmarks Tools Help

http://localhost/phpmysql4e/chapter28/show_book.php?isbn=0672329166

Book-O-Rama

Log Out

PHP and MySQL Web Development



- **Author:** Luke Welling and Laura Thomson
- **ISBN:** 0672329166
- **Our Price:** 49.99
- **Description:** PHP & MySQL Web Development teaches the reader to develop dynamic, secure e-commerce web sites. You will learn to integrate and implement these technologies by following real-world examples and working sample projects.

Edit Item

Admin Menu

Continue

Done

图　28-14　show_book.php脚本为管理员用户生成不同的输出

管理员可以访问该页面中的两个新增选项："Edit Item"和"Admin Menu"。注意在页面的右上角我们看不到购物车——取而代之的是一个"Log Out"按钮。

这些代码都在程序清单28-8中列出，如下所示：

```
if(check_admin_user()){
display_button("edit_book_form.php?isbn=".$isbn,"edit-item","Edit Item");
display_button("admin.php","admin-menu","Admin Menu");
display_button($target,"continue","Continue");
}
```

如果回头看看show_cat.php脚本，将发现它也有这些内置选项。

如果管理员点击"Edit Item"按钮，将进入edit_book_form.php脚本。该脚本的输出如图28-15所示。

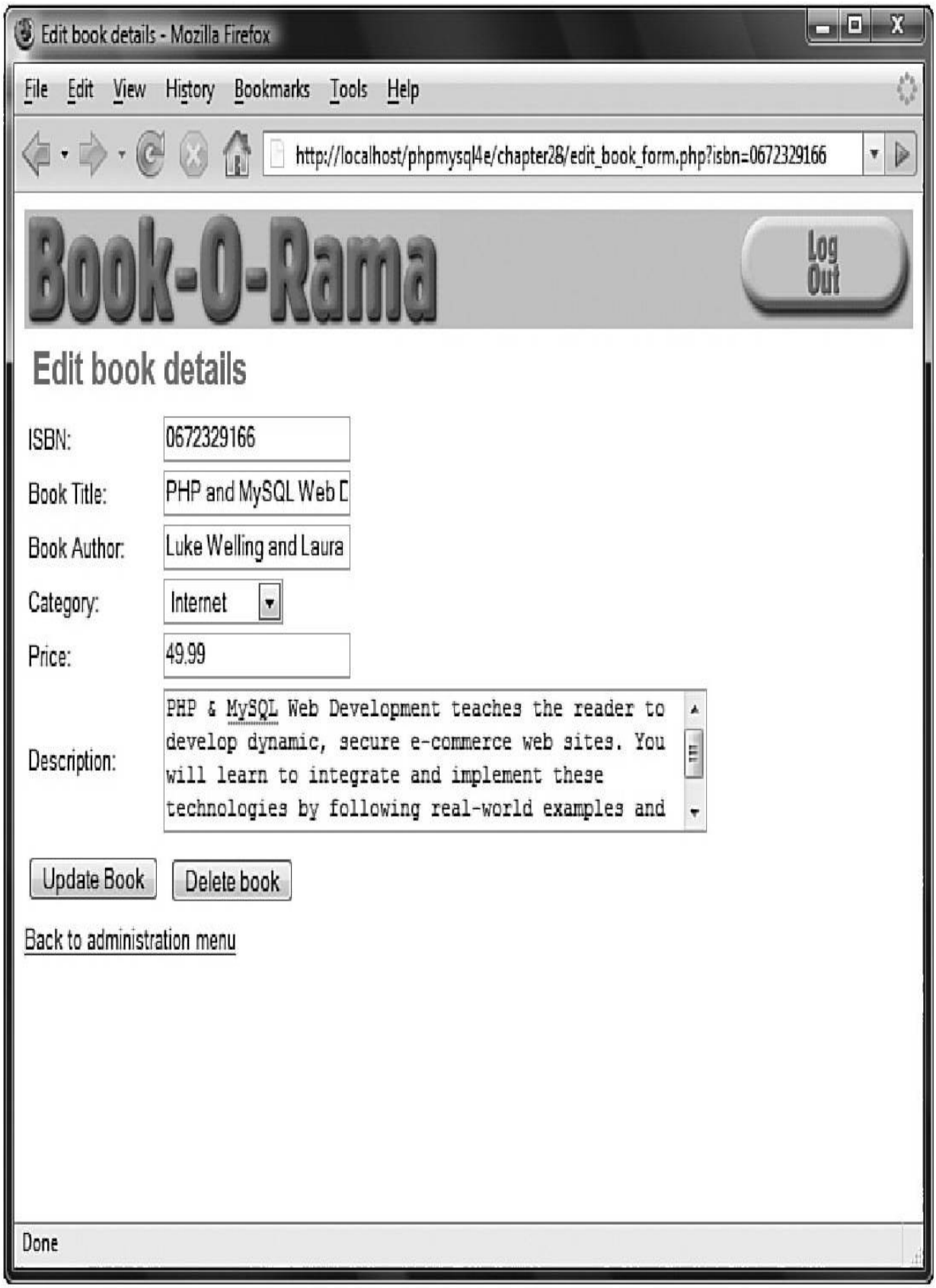

Edit book details - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/phpmysql4e/chapter28/edit_book_form.php?isbn=0672329166
Book-O-Rama
Log Out
Edit book details
ISBN:
0672329166
Book Title:
PHP and MySQL Web D
Book Author:
Luke Welling and Laura
Category:
Internet
Price:
49.99
Description:
PHP & MySQL Web Development teaches the reader to develop dynamic, secure e-commerce web sites. You will learn to integrate and implement these technologies by following real-world examples and
Update Book
Delete book
Back to administration menu
Done

图　28-15　edit_book_form.php脚本使管理员可以编辑图书的详细信息或者删除一本书

实际上，这是我们前面用来获取图书详细信息的同一个表单。我们建立了一个指向该表单的选项用来传入和显示现存的图书数据。对于目录，同样如此。要明白为什么目录操作也是如此，请参阅程序清单28-19。

程序清单28-19　admin_fns.php文件中的display_book_form()函数——该表单完成了两项工作：插入图书和编辑图书

```
function display_book_form($book=''){
//This displays the book form.
//It is very similar to the category form.
//This form can be used for inserting or editing books.
//To insert,don't pass any parameters.This will set$edit
//to false,and the form will go to insert_book.php.
//To update,pass an array containing a book.The
//form will be displayed with the old data and point to update_book.php.
//It will also add a"Delete book"button.
//if passed an existing book,proceed in"edit mode"
$edit=is_array($book);
//most of the form is in plain HTML with some
//optional PHP bits throughout
?>
<form method="post"
action="<?php echo$edit?'edit_book.php':'insert_book.php';?>">
<table border="0">
<tr>
<td>ISBN:</td>
<td><input type="text"name="isbn"
value="<?php echo$edit?$book['isbn']:'';?>"/></td>
</tr>
<tr>
<td>Book Title:</td>
<td><input type="text"name="title"
```

```
value="<?php echo$edit?$book['title']:'';?>"/></td>
</tr>
<tr>
<td>Book Author:</td>
<td><input type="text"name="author"
value="<?php echo$edit?$book['author']:'';?>"/></td>
</tr>
<tr>
<td>Category:</td>
<td><select name="catid">
<?php
//list of possible categories comes from database
$cat_array=get_categories();
foreach($cat_array as$thiscat){
echo"<option value=\"".$thiscat['catid']."\"";
//if existing book,put in current catgory
if(($edit)&&($thiscat['catid']==$book['catid'])){
echo"selected";
}
echo">".$thiscat['catname']."</option>";
}
?>
</select>
</td>
</tr>
<tr>
<td>Price:</td>
<td><input type="text"name="price"
value="<?php echo$edit?$book['price']:'';?>"/></td>
</tr>
<tr>
<td>Description:</td>
<td><textarea rows="3"cols="50"
name="description">
<?php echo$edit?$book['description']:'';?>
</textarea></td>
```

```
</tr>
<tr>
<td<?php if(!$edit){echo"colspan=2";}?>align="center">
<?php
if($edit)
//we need the old isbn to find book in database
//if the isbn is being updated
echo"<input type=\"hidden\"name=\"oldisbn\"
value=\"".$book['isbn']."\"/>";
?>
<input type="submit"
value="<?php echo$edit?'Update':'Add';?>Book"/>
</form></td>
<?php
if($edit){
echo"<td>
<form method=\"post\"action=\"delete_book.php\">
<input type=\"hidden\"name=\"isbn\"
value=\"".$book['isbn']."\"/>
<input type=\"submit\"value=\"Delete book\"/>
</form></td>";
}
?>
</td>
</tr>
</table>
</form>
<?php
}
```

如果我们传入一个包含图书数据的数组，该表单将显示为编辑模式，并用数组中的数据填充其中的域，如下所示：

```
<input type="text"name="price"
value="<?php echo$edit?$book['price']:'';?>"/><
```

我们甚至可以得到一个不同的提交按钮。实际上，对编辑表单，我们有两个脚本——一个用来更新图书，而另一个用来删除图书。这两个脚本分别称为edit_book.php和delete_book.php，它们都要相应地更新数据库。

从工作原理来说，目录版本与图书版本的脚本是相同的，但有一点不同的是，当管理员删除一个目录的时候，如果该目录仍然包含有图书，那么该目录不可删除（这需要通过数据库查询进行检查），这避免了任何不正常删除的问题。在第8章中我们讨论过这样的问题。在这种情况下，如果一个目录还有图书包含在内，就被直接删除，那么这些书将成为“孤儿”。我们不知道它们属于哪个目录，也不知道如何查找它们！

以上就是对管理界面的概述。要了解更多信息，请参阅代码——它们都可在附带的文件里找到。

28.8 扩展该项目

本项目创建了一个相当简单的购物车系统。我们还可以对它进行许多改进和提高：

■在真正的在线商店，可能必须建立一些订单记录和实施系统——在我们这个系统中，用户无法看到已经预订了的订单。

■顾客希望在不必与我们联系的前提下就能检查到他们的订单处理情况。我们觉得重要的是用户不必登录进来就能看到这些信息。但是，为已有客户提供一种验证身份的方法可以使他们能够查看自己以前的订单，并且也可以将操作与个人情况紧密地结合起来。

■如今，图书的图片可以通过FTP传输到该网站的图像目录并给它们取一个合适的名字。可以把文件上载到图书插入页，以使该操作方便一些。

■可以添加用户登录、个性化设置以及书目推荐、在线评论、会员制度、库存级别检查等。可能要添加的物品是无穷无尽的。

28.9 使用一个已有系统

如果希望得到一个很有特点而且运行速度也很快的购物车，可以尝试使用一个已有的购物车系统。一个著名的使用PHP编写的开放源代码购物车叫FishCartSQL，可以通过如下站点获得：http://www.fishcart.org/。

该系统有很多高级的特性，例如，用户记录、即时交易、多语言支持、信用卡处理以及支持在一个服务器上部署多个在线商店。当然，当使用一个已有系统的时候，经常会发现有许多功能自己不需要使用，而需要使用的功能可能没有。开放源代码产品的好处就是可以随时进入源代码改掉不喜欢的东西。

28.10 下一章

在下一章中，我们将讨论如何构建一个能够收发电子邮件的Web界面，这个界面将使用IMAP协议收发电子邮件。

第29章　创建一个基于Web的电子邮件服务系统

当今，越来越多的站点希望为用户提供一个基于Web的电子邮件系统。本章将介绍如何使用PHP IMAP库为一个已有的邮件服务器实现Web界面。可以用它通过Web页面来检查自己已有的邮箱，或者将它扩展，使其像基于Web的大规模电子邮件系统一样支持多用户，例如，Gmail、Yahoo!Mail和Hotmail。

在这个项目中，我们将建立一个电子邮件客户端，Warm Mail，它允许用户：

■连接到POP3或IMAP邮件服务器上的账户

■阅读邮件

■发送邮件

■回复邮件消息

■转发邮件消息

■从用户账户中删除邮件

29.1　解决方案的组成

为了使用户可以阅读邮件，需要找到一种方法连接到邮件服务器。通常，这种机制与连接Web服务器的机制不同。我们需要一种方法来与用户的邮箱进行交互，检查收到的邮件消息，并单独处理每一条消息。

29.1.1　电子邮件协议：POP3和IMAP

邮件服务器需要支持两个主要的协议：POP3和IMAP。通过这两个协议，用户能够查看邮箱。如果可能，我们应该同时支持二者。POP3表示邮局协议（Post Office Protocol，POP）第三版，IMAP表示互联网消息访问协议（Internet Message Access Protocol，IMAP）。

二者主要的不同是：POP3（也是较为常见的情况）是为了那些仅从服务器下载或删除邮件而短时间连接网络的人所设计的。而IMAP出于在线使用的目的，它可以使用户与邮件进行交互，而且邮件也可以永久保存在远程服务器上。在这里，我们将不会使用IMAP的一些更高级的特性。

如果对这两个协议之间的差别感兴趣，可以参考RFC标准（RFC 1939版本3和RFC 3501 IMAP版本4修改稿1）。可以在如下站点找到关于二者相比较的优秀文章：

http://www.imap.org/papers/imap.vs.pop.brief.html.

这两个协议都不是为发送邮件设计的，因此，我们必须使用简单电子邮件传输协议（Simple Mail Transfer Protocol，SMTP）来发送邮件，在本书的前面内容中，当我们介绍如何在PHP中使用mail()函数时，我们介绍了这个协议。该协议在RFC 821中进行了描述。

29.1.2 PHP对POP3和IMAP的支持

虽然PHP对IMAP和POP3有着不错的支持，但它都是通过IMAP函数库所提供的功能来实现的。为了使用本章所提供的代码，需要安装IMAP库。可以通过查看phpinfo()函数的输出来判断是否已经安装该库。

如果使用的是Linux或UNIX并且没有安装它，则需要下载该库。可以通过FTP从如下站点得到最新的版本：ftp://ftp.cac.washington.edu/imap/。

在UNIX平台下，你可以下载源代码并根据操作系统对其进行编译。

你必须在系统包含目录中创建一个保存IMAP文件的目录，例如imap（不要只是将这些文件复制到这个基本的包含目录中，因为这可能会导致一些冲突）。在新目录中，创建imap/lib和imap/include两个子目录。将安装中涉及的所有*.h文件复制到imap/include/目录。当完成了编译，将创建一个c-client.a文件。将其重命名为libc-client.a并且复制到imap/lib目录。

然后必须运行PHP的配置脚本，将--with-imap=dirname指令添加到所使用的任何其他参数，（这里，dirname是所创建的目录名称）并重新编译PHP。

要在Windows下使用IMAP扩展，可以打开php.ini配置文件，注释掉如下代码行：

```
extension=php_imap.dll
```

然后再重新启动Web服务器。

通过运行phpinfo()函数，可以确认IMAP扩展是否已经安装。函数输出将显示关于IMAP的内容。

需要注意的一件有趣的事情是：虽然这些函数都叫做IMAP函数，但它们同样能够支持POP3和网络新闻传输协议（Network News Transfer Protocol，NNTP）。在这个例子中，我们准备将它们用于IMAP和POP3，但Warm Mail应用程序可以很容易扩展，因此除了邮件客户端外，它还可以使用NNTP并且成为一个新闻阅读程序。

该函数库中包含一些函数，但为了实现该程序的功能，我们仅使用其中少数几个。具体使用函数时，我们会详细介绍这些函数，但是要注意有很多这样的函数。针对具体需求的不同，或者想为程序增加特别的特性，请查看相关资料。

仅使用部分的内置函数，就可以建立一个相当有用的邮件应用程序。这意味着只需要研究部分文档。在本章中，我们将使用的IMAP函数包括：

■imap_open()

■imap_close()

■imap_headers()

■imap_header()

■imap_fetchheader()

■imap_body()

■imap_delete()

■imap_expunge()

为了让用户能够读取邮件，需要获得用户邮件服务器和用户账户的详细信息。为了不必每次都从用户那里获得这些详细信息，我们将建立一个有用户名和密码的数据库以便存储这些资料。

通常，人们会有多个电子邮件账户（比如一个用于家庭，另一个用于工作），我们应该允许他们连接其中任何一个。因此，应该允许他们在数据库中有多条记录。

除了让用户能够发送新邮件外，还应该让用户能够阅读、回复、转发和删除已存在的邮件。我们可以使用IMAP或POP3完成所有的阅读功能，使用SMTP的mail()函数完成所有发送功能。

下面，让我们看看如何实现这些功能。

29.2 解决方案概述

这个基于Web的系统的大体流程图与其他电子邮件客户端没有明显差别。该系统流程和模块如图29-1所示。

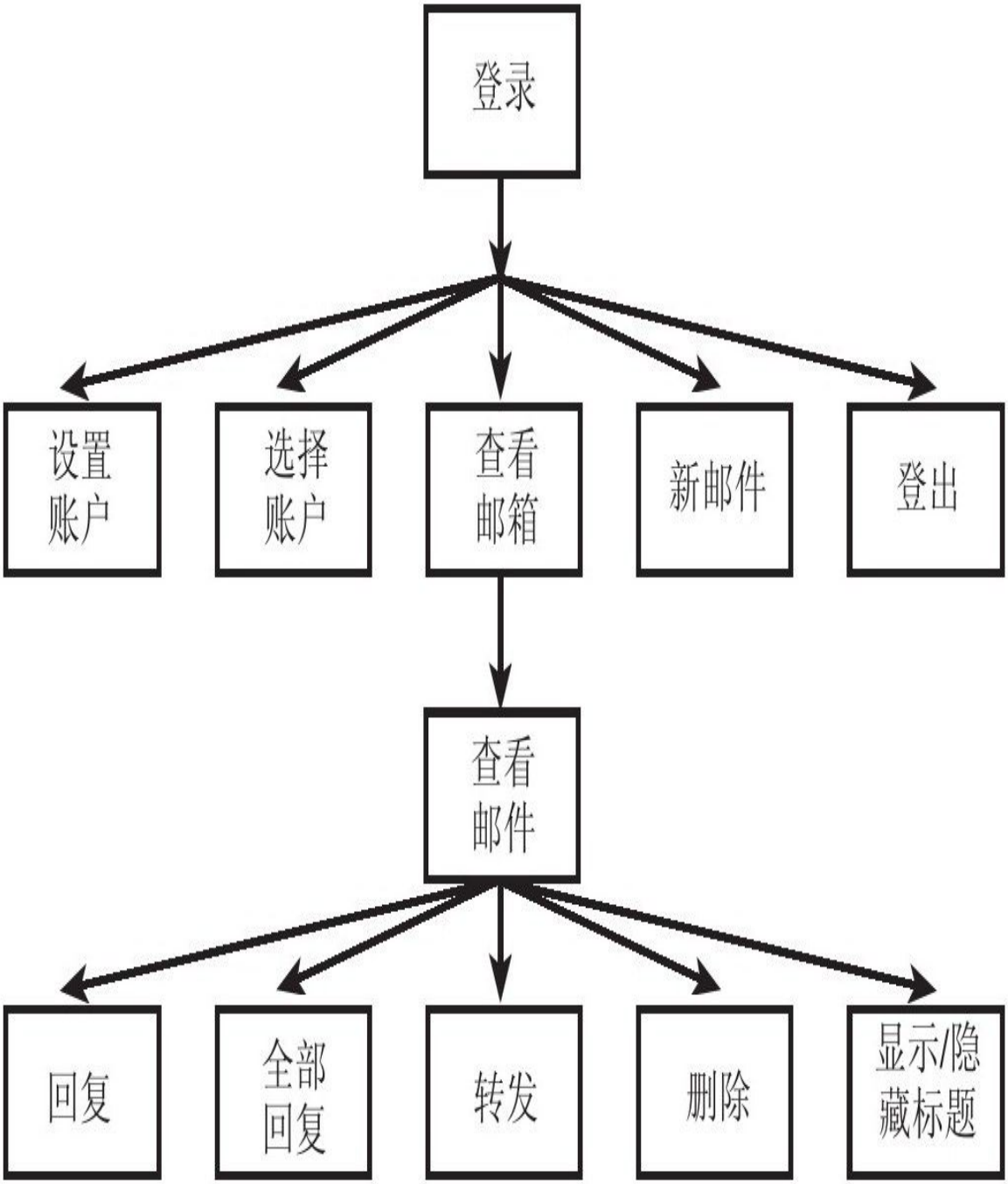


图　29-1　Warm Mail界面为用户提供邮箱级的功能和消息级的功能

可以看到，我们首先要求用户登录，然后给他一个选项。用户可以选择创建一个新的邮件账户，或选择一个已有的账户来使用。用户也可以查看收到的来信——回复、转发或者删除它——还可以发送新邮件。

我们也给予用户查看特定消息详细标题的选项。查看完整标题可以显示更多关于消息的内容。我们可以看出该邮件来自哪一台机器——它是记录垃圾邮件的非常有用的工具。可以看出哪台机器转发了邮件，以及在什么时间到达每一台主机——这对于认清谁该对延时邮件负责是非常有用的。也可以看到发送者使用的是哪一种邮件客户端程序，如果应用程序将可选的信息加入到标题中的话。

在这个项目中，我们使用了一个稍有不同的应用程序架构。不是使用一组脚本，每一个脚本对应于一个模块，而是有一个稍长的脚本——index.php，该脚本就像由GUI驱动的事件循环一样工作。在站点上，我们通过点击按钮所触发的每一个操作都会将我们返回至index.php，但对应有一个不同的参数。根据参数的不同，脚本将调用不同的函数为用户提供适当的输出。这些函数照样都在函数库当中。

这种架构适合于像这个项目一样的小型应用程序。它适合那些基于事件驱动的应用程序，即用户动作触发某功能的应用程序。简单地使用事件处理不适合那些由整个团队人员操作的大型架构或项目。

Warm Mail项目所包含的文件概要如表29-1所示。

下面，让我们继续了解这个应用程序。

表29-1 Warm Mail应用程序中的文件

文件名	类型	描述
index.php	应用程序	运行整个应用程序的主要脚本
include_fns.php	函数	该应用程序包含文件的集合
data_valid_fns.php	函数	使输入数据生效的函数的集合
db_fns.php	函数	连接到mail数据库的函数集合
mail_fns.php	函数	打开邮箱、阅读邮件与邮件相关的函数集合
output_fns.php	函数	输出HTML函数的集合
user_auth_fns.php	函数	验证用户的函数的集合
create_database.sql	SQL	用来建立book_sc数据库和建立用户的SQL

29.3 建立数据库

Warm Mail的数据库相当简单，因为实际上我们并不打算在其中保存任何电子邮件。

我们需要保存系统用户的信息，我们将为每个用户保存如下所示的字段信息：

■用户名——用户期望出现在Warm Mail中的用户名。

■密码——用户期望出现在Warm Mail中的用户密码。

■地址——用户期望出现在From field中的电子邮件地址，该地址将出现在从该系统发出去的电子邮件的From域。

■显示名称——他们想在发送给其他人的邮件中显示的名称。

我们也需要为那些检查该系统的用户存储账户信息。对于每一个账户，必须保存以下信息：

■用户名——该账户所属的Warm Mail用户。

■服务器——该账户所在的机器，例如，本地机或mail.tangledweb.com.au，或其他Domain（域）。

■端口——当使用该账户时连接的端口号。通常，对于POP3服务器来说，这个端口是110，而对于IMAP服务器来说，则是143。

■类型——用来连接到该服务器的协议，POP3或者IMAP。

■远程用户——连接到邮件服务器的用户名。

■远程口令——连接到邮件服务器的口令。

■账户ID——区别账户的唯一标识符。

可以通过运行程序清单29-1中所示的SQL语句来创建这个应用程序的数据库。

程序清单29-1　create_database.sql——用来创建邮件数据库的SQL

```
create database mail;
use mail;
create table users
(
username char(16)not null primary key,
password char(40)not null,
address char(100)not null,
displayname char(100)not null
);
create table accounts
(
username char(16)not null,
server char(100)not null,
port int not null,
type char(4)not null,
remoteuser char(50)not null,
remotepassword char(50)not null,
accountid int unsigned not null auto_increment primary key
);
grant select,insert,update,delete
on mail.*
to mail@localhost identified by'password';
```

请记住，我们可以通过输入如下所示的命令来执行这个SQL语句：

```
mysql-u root-p＜create_database.sql
```

我们必须提供root用户密码。此外，还应该在运行以上命令之前修改create_database.sql和db_fns.php中的邮件用户密码。

在本书附带的文件中，我们提供了一个叫做populate.sql的SQL文件。在这个应用程序中，我们没有创建用户注册或管理进程。如果要在一个更大规模的系统上使用该软件，可以自己添加一个。但是如果要作为个人使用，只需要将自己的账户信息插入到数据库中。populate.sql脚本为做这些提供了一个模板，以此将详细信息插入到其中并运行它以便将自己设置为用户。

29.4　了解脚本架构

正如我们前面提到的，该应用程序使用一个脚本控制所有操作。该脚本叫做index.php，如程序清单29-2所示。该脚本比较长，我们逐段详细介绍。

程序清单29-2　index.php——Warm Mail系统的框架

```
<?php
//This file is the main body of the Warm Mail application.
//It works basically as a state machine and shows users the
//output for the action they have chosen.
//*************************************************************************
//Stage 1:pre-processing
//Do any required processing before page header is sent
//and decide what details to show on page headers
//*************************************************************************
include('include_fns.php');
session_start();
//create short variable names
$username=$_POST['username'];
$passwd=$_POST['passwd'];
$action=$_REQUEST['action'];
$account=$_REQUEST['account'];
$messageid=$_GET['messageid'];
$to=$_POST['to'];
$cc=$_POST['cc'];
$subject=$_POST['subject'];
$message=$_POST['message'];
$buttons=array();
//append to this string if anything processed before header has output
$status='';
//need to process log in or out requests before anything else
if($username||$password){
```

```
if(login($username,$passwd)){
$status.="<p style=\"padding-bottom:100px\">Logged in
successfully.</p>";
$_SESSION['auth_user']=$username;
if(number_of_accounts($_SESSION['auth_user'])==1){
$accounts=get_account_list($_SESSION['auth_user']);
$_SESSION['selected_account']=$accounts[0];
}
}else{
$status.="<p style=\"padding-bottom:100px\">Sorry,we could
not log you in with that username and password.</p>";
}
}
if($action=='log-out'){
session_destroy();
unset($action);
$_SESSION=array();
}
//need to process choose,delete or store account before drawing header
switch($action){
case'delete-account':
delete_account($_SESSION['auth_user'],$account);
break;
case'store-settings':
store_account_settings($_SESSION['auth_user'],$_POST);
break;
case'select-account':
//if have chosen a valid account,store it as a session variable
if(($account)&&(account_exists($_SESSION['auth_user'],$account))){
$_SESSION['selected_account']=$account;
}
break;
}
//set the buttons that will be on the tool bar
$buttons[0]='view-mailbox';
$buttons[1]='new-message';
```

```
$buttons[2]='account-setup';
//only offer a log out button if logged in
if(check_auth_user()){
$buttons[4]='log-out';
}
//*****************************************************************************
//Stage 2:headers
//Send the HTML headers and menu bar appropriate to current action
//*****************************************************************************
if($action){
//display header with application name and description of page or action
do_html_header($_SESSION['auth_user'],"Warm Mail-".
format_action($action),
$_SESSION['selected_account']);
}else{
//display header with just application name
do_html_header($_SESSION['auth_user'],"Warm Mail",
$_SESSION['selected_account']);
}
display_toolbar($buttons);
//*****************************************************************************
//Stage 3:body
//Depending on action,show appropriate main body content
//*****************************************************************************
//display any text generated by functions called before header
echo$status;
if(!check_auth_user()){
echo"<p>You need to log in";
if(($action)&&($action!='log-out')){
echo"to go to".format_action($action);
}
echo".</p>";
display_login_form($action);
}else{
switch($action){
//if we have chosen to setup a new account,or have just added or
```

```
//deleted an account,show account setup page
case'store-settings':
case'account-setup':
case'delete-account':
display_account_setup($_SESSION['auth_user']);
break;
case'send-message':
if(send_message($to,$cc,$subject,$message)){
echo"<p style=\"padding-bottom:100px\">Message sent.</p>";
}else{
echo"<p style=\"padding-bottom:100px\">Could not send message.</p>";
}
break;
case'delete':
delete_message($_SESSION['auth_user'],
$_SESSION['selected_account'],$messageid);
//note deliberately no'break'-we will continue to the next case
case'select-account':
case'view-mailbox':
//if mailbox just chosen,or view mailbox chosen,show mailbox
display_list($_SESSION['auth_user'],
$_SESSION['selected_account']);
break;
case'show-headers':
case'hide-headers':
case'view-message':
//if we have just picked a message from the list,or were looking at
//a message and chose to hide or view headers,load a message
$fullheaders=($action=='show-headers');
display_message($_SESSION['auth_user'],
$_SESSION['selected_account'],
$messageid,$fullheaders);
break;
case'reply-all':
//set cc as old cc line
if(!$imap){
```

```
$imap=open_mailbox($_SESSION['auth_user'],
$_SESSION['selected_account']);
}
if($imap){
$header=imap_header($imap,$messageid);
if($header->reply_toaddress){
$to=$header->reply_toaddress;
}else{
$to=$header->fromaddress;
}
$cc=$header->ccaddress;
$subject="Re:".$header->subject;
$body=add_quoting(stripslashes(imap_body($imap,$messageid)));
imap_close($imap);
display_new_message_form($_SESSION['auth_user'],
$to,$cc,$subject,$body);
}
break;
case'reply':
//set to address as reply-to or from of the current message
if(!$imap){
$imap=open_mailbox($_SESSION['auth_user'],
$_SESSION['selected_account']);
}
if($imap){
$header=imap_header($imap,$messageid);
if($header->reply_toaddress){
$to=$header->reply_toaddress;
}else{
$to=$header->fromaddress;
}
$subject="Re:".$header->subject;
$body=add_quoting(stripslashes(imap_body($imap,$messageid)));
imap_close($imap);
display_new_message_form($_SESSION['auth_user'],
$to,$cc,$subject,$body);
```

```
}
break;
case'forward':
//set message as quoted body of current message
if(!$imap){
$imap=open_mailbox($_SESSION['auth_user'],
$_SESSION['selected_account']);
}
if($imap){
$header=imap_header($imap,$messageid);
$body=add_quoting(stripslashes(imap_body($imap,$messageid)));
$subject="Fwd:".$header->subject;
imap_close($imap);
display_new_message_form($_SESSION['auth_user'],
$to,$cc,$subject,$body);
}
break;
case'new-message':
display_new_message_form($_SESSION['auth_user'],
$to,$cc,$subject,$body);
break;
}
}
//*****************************************************************************
//Stage 4:footer
//*****************************************************************************
do_html_footer();
?>
```

这个脚本使用一个事件处理的方法。它包含对于每个事件应该调用哪个函数的知识或逻辑。在这种情况下，事件是由用户点击站点内不同的按钮触发的，每一个按钮对应一种动作。大多数按钮都是由display_button()函数产生的，但是如果用户点击的按钮是提交按钮时，则是调用display_form_button()函数。这些函数都在output_fns.php中。这些都转至表单的URL：

```
index.php?action=log-out
```

当调用index.php时，action变量值决定哪一个事件处理程序被触发。

该脚本由如下所示的4个主要部分组成：

1）在将页面标题发送到浏览器之前要做一些必要的处理，例如，开始一个会话，执行用户选择的动作所必须的预处理操作，并确定页面标题的内容。

2）为用户选择的动作处理和发送适当的标题和菜单条。

3）选择要执行哪一块代码，取决于用户所选的动作。不同的动作将触发不同的函数调用。

4）发送页面的页脚。

如果粗略地看一下代码，会发现这四个部分都用注释作了标记。

为了完全理解脚本程序，让我们来真正使用这个站点。

29.5　登录与登出

当用户载入index.php页面时，将看到如图29-2所示的输出页面。


Warm Mail - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/phpmysql4e/chapter29/index.php
Warm Mail
View Mailbox
New Message
Account Setup
You need to log in.
Please Log In
Username:
Password:
Log In
Done

图　29-2　Warm Mail系统的登录页面询问用户名与密码

这是该应用程序的默认动作。如果没有选择$action动作，并且也没有提供登录细节，将执行代码的下面部分。

在预处理阶段，PHP将首先执行如下所示的代码：

```
include('include_fns.php');
session_start();
```

以上代码将开始一个会话，该会话可以用来保存会话变量$auth_user和$selected_account，这两个变量后面会用到。

在其他应用程序中，我们创建了短的变量名称。从第1章“PHP快速入门教程”开始，我们就在与每一个表单相关的脚本中使用短的变量名称，因此除了action变量外，我们不需要再提这一点。根据应用程序功能模块的不同，它可能是一个GET或POST变量。因此，可以从$_REQUEST数组获得这个变量。对于account变量也是如此，因此通常它可以通过GET进行访问，但是当要删除一个账户时，需要通过POST进行访问。

为了在定制用户界面时减少工作量，可以使用一个数组来控制出现在工具条中的按钮。我们声明了一个如下所示的空数组：

```
$buttons=array();
```

并将按钮设置为我们希望在页面看到的那些按钮：

```
$buttons[0]='view-mailbox';
$buttons[1]='new-message';
$buttons[2]='account-setup';
```

如果用户是以管理员身份登录的，还需要在这个数组中添加更多按钮。

在标题阶段，我们打印了一个普通的vanilla标题：

```
do_html_header($_SESSION['auth_user'],"Warm Mail",
$_SESSION['selected_account']);
...
display_toolbar($buttons);
```

这些代码将打印标题和标题头，然后可以看到如图29-2所示的按钮工具条。这些函数可以在output_fns.php函数库中找到，但由于可以轻易地在图中看出他们的运行结果，因此在这里，我们不再详细介绍它们。

现在，我们来了解代码的主体部分：

```
if(!check_auth_user()){
echo"<p>You need to log in";
if(($action)&&($action!='log-out')){
echo"to go to".format_action($action);
}
echo".</p>";
display_login_form($action);
}
```

check_auth_user()函数来自于user_auth_fns.php库。我们已经在前面的某些项目中使用过与此非常相似的代码——它检查用户是否登录。在这里，作为一种情况，如果用户没有登录，我们将显示一个登录表单，如图29-2所示。我们使用output_fns.php中的display_login_form()函数绘制该表单。

如果用户正确填写了表单并点击"Log In"按钮，将看到如图29-3所示的输出页面。


Warm Mail - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/phpmysql4e/chapter29/index.php?action=
Warm Mail
View Mailbox
New Message
Account Setup
Log Out
Logged in successfully.
Done

图　29-3　在成功登录后，用户可以开始使用该应用程序

在这个脚本的执行过程中，我们将触发代码的不同部分。登录表单有两个字段，$username和$password。如果填好了这两个字段，如下所示的预处理代码将会被触发：

```
if($username||$password){
if(login($username,$passwd)){
$status.="<p style=\"padding-bottom:100px\">Logged in successfully.</p>";
$_SESSION['auth_user']=$username;
if(number_of_accounts($_SESSION['auth_user'])==1){
$accounts=get_account_list($_SESSION['auth_user']);
$_SESSION['selected_account']=$accounts[0];
}
}else{
$status.="<p style=\"padding-bottom:100px\">Sorry,we could not log you
in with that username and password.</p>";
}
}
```

可以看到，以上代码将调用login()函数，该函数与第27章“建立用户身份验证机制和个性化设置”和第27章“创建一个购物车”中的相似。如果一切正常，我们将在会话变量auth_user中注册该用户名。

除了设置当用户没有登录时所看到的按钮外，也可以添加另外的按钮让用户登出，如下所示：

```
if(check_auth_user()){
$buttons[4]='log-out';
}
```

可以在图29-3中看到这个"Log Out"按钮。

在“标题”阶段，我们再次显示了标题和按钮。在正文阶段，显示了以前设置的状态信息：

```
echo$status;
```

接下来，我们将打印页脚并等待用户的下一步操作。

29.6 建立账户

当用户第一次使用Warm Mail系统时，需要建立一些电子邮件账户。如果用户点击"Account Setup"按钮，将会把action变量的值设为account-setup并再次调用index.php脚本。用户将看到如图29-4所示的输出页面。



图　29-4　用户在可以阅读电子邮件前必须建立电子邮件账户

现在，让我们回过头来看看程序清单29-2。这一次，由于$action值的不同，我们看到不同的行为。我们将得到稍有不同的标题，如下所

示：

```
do_html_header($_SESSION['auth_user'],"Warm Mail-".
format_action($action),
$_SESSION['selected_account']);
```

更重要的是，我们将得到一个不同的正文，如下所示：

```
case'store-settings':
case'account-setup':
case'delete-account':
display_account_setup($_SESSION['auth_user']);
break;
```

这是一个典型的模式：每个命令都将调用一个函数。在这个例子中，我们调用了display_account_setup()函数。该函数的代码如程序清单29-3所示。

程序清单29-3　来自output_fns.php文件的display_account_setup()函数——获得并显示账户详细信息的函数

```
function display_account_setup($auth_user){
//display empty'new account'form
display_account_form($auth_user);
$list=get_accounts($auth_user);
$accounts=sizeof($list);
//display each stored account
foreach($list as$key=>$account){
//display form for each accounts details.
//note that we are going to send the password for all accounts in the HTML
//this is not really a very good idea
display_account_form($auth_user,$account['accountid'],$account['server'],
$account['remoteuser'],$account['remotepassword'],
$account['type'],$account['port']);
```

```
}
}
```

当我们调用该函数时，它将显示一个空白表单来添加新账户，接着是包括该用户所有当前邮件账户的可编辑文本框。display_account_form()函数将显示如图29-4所示的表单。可以发现，在这里我们通过两种不同的方法来使用它：如果不带参数，将显示一个空表单，如果带有一组完整的参数，则显示一条已有的记录。这个函数在output_fns.php库中；它只是简单地输出HTML，因此我们这里不详细介绍它。

检索任何已存在账户的函数是get_accounts()，它出自mail_fns.php库。函数代码如清单29-4所示。

程序清单29-4　来自mail_fns.php文件的get_accounts()函数——为某个特定用户检索所有的账户细节

```
function get_accounts($auth_user){
$list=array();
if($conn=db_connect()){
$query="select*from accounts where username='".$auth_user."'";
$result=$conn->query($query);
if($result){
while($settings=$result->fetch_assoc()){
array_push($list,$settings);
}
}else{
return false;
}
}
return$list;
}
```

可以看到，该函数将连接数据库，检索特定用户的所有账户并以数组的形式返回。

29.6.1　创建一个新账户

如果用户填写了账户表单并点击"Save Changes"按钮，store-settings动作将会被触发。下面，让我们来看看在index.php脚本中该动作的事件处理代码。在预处理阶段，执行如下所示的代码：

```
case'store-settings':
store_account_settings($_SESSION['auth_user'],$_POST);
break;
```

store_account_settings()函数将一个新账户的详细信息写入数据库。该函数的代码如程序清单29-5所示。

程序清单29-5　mail_fns.php文件的store_account_settings()函数——为用户保存一个新账户详细信息的函数

```
function store_account_settings($auth_user,$settings){
if(!filled_out($settings)){
echo"<p>All fields must be filled in.Try again.</p>";
return false;
}else{
if($settings['account']>0){
$query="update accounts set server='".$settings[server]."',
port=".$settings[port].",type='".$settings[type]."',
remoteuser='".$settings[remoteuser]."',
remotepassword='".$settings[remotepassword]."'
where accountid='".$settings[account]."'
and username='".$auth_user."'";
}else{
$query="insert into accounts values('".$auth_user."',
'".$settings[server]."','".$settings[port]."',
'".$settings[type]."','".$settings[remoteuser]."',
'".$settings[remotepassword]."',NULL)";
}
if($conn=db_connect()){
```

```
$result=$conn->query($query);
if($result){
return true;
}else{
return false;
}
}else{
echo"<p>Could not store changes.</p>";
return false;
}
}
}
```

可以看到，这个函数中有两个选项分别对应于插入一个新账户和更新一个已有的账户。函数执行相应的查询语句来保存账户详细信息。

在保存账户详细信息后，我们将返回到index.php的主体阶段：

```
case'store-settings':
case'account-setup':
case'delete-account':
display_account_setup($_SESSION['auth_user']);
break;
```

可以看到，我们像以前一样执行display_account_setup()函数来列出用户的账户详细信息。新添加的账户就被包括进来了。

29.6.2 修改已有账户

修改已有账户的过程也是非常相似的。用户可以修改账户的详细信息并点击"Save Changes"按钮。这将再次触发store-settings动作，但是，这一次，它将更新账户详细信息而不是插入账户信息。

29.6.3 删除账户

要删除一个账户，可以点击显示在每个账户列表下面的"Delete Account"按钮。这会触发delete-account动作。

在index.php脚本的预处理阶段，该操作将执行如下所示的代码：

```
case'delete-account':
delete_account($_SESSION['auth_user'],$account);
break;
```

这段代码将调用delete_account()函数。该函数的代码如程序清单29-6所示。由于可能用到的账户包含在标题中，因此删除账户操作必须在“标题”阶段以前处理。在账户列表可以正确地列出之前需要进行更新。

程序清单29-6　mail_fns.php文件的delete_account()函数——删除一个账户详细信息的函数

```
function delete_account($auth_user,$accountid){
//delete one of this user's accounts from the DB
$query="delete from accounts where accountid='".$accountid."'
and username='".$auth_user."'";
if($conn=db_connect()){
$result=$conn->query($query);
}
return$result;
}
```

当程序执行返回到index.php时，主体阶段将运行如下所示的代码：

```
case'store-settings':
```

```
case'account-setup':
case'delete-account':
display_account_setup($_SESSION['auth_user']);
break;
```

请注意，我们会发现这与前面运行的代码一样，它仅仅显示了用户账户的列表。

29.7 阅读邮件

在用户建立一些账户后，我们转移到重点问题上来：连接这些账户并读取邮件。

29.7.1 选择账户

我们需要选取一个用户账户以便从中读取邮件。当前选择的账户存储在会话变量$selected_account中。

如果该用户在系统中仅注册了一个账户，那么登录时会自动选取该账户，如下所示：

```
if(number_of_accounts($_SESSION['auth_user'])==1){
$accounts=get_account_list($_SESSION['auth_user']);
$_SESSION['selected_account']=$accounts[0];
}
```

来自mail_fns.php文件中的number_of_accounts()函数用来判断用户是否有多个账户。该函数源代码如程序清单29-7所示。get_account_list()函数将检索用户账户ID的数组。在这种情况下，用户仅有一个账户，因此也就是数组的第0个值。

程序清单29-7 mail_fns.php文件中的number_of_accounts()函数——用来计算某个用户注册了多少个账户的函数

```
function number_of_accounts($auth_user){
//get the number of accounts that belong to this user
$query="select count(*)from accounts where
username='".$auth_user."'";
if($conn=db_connect()){
$result=$conn->query($query);
if($result){
$row=$result->fetch_array();
```

```
return$row[0];
}
}
return 0;
}
```

---

get_account_list()函数与我们以前看到的get_accounts()函数类似，不同之处在于它仅仅检索账户名称。

如果用户注册了多个账户，需要选择使用其中的一个。在这种情况下，标题中将包括一个列表框，列出所有可供使用的邮箱。选择一个适当的邮箱后系统会自动为用户显示该邮箱内容，如图29-5所示。

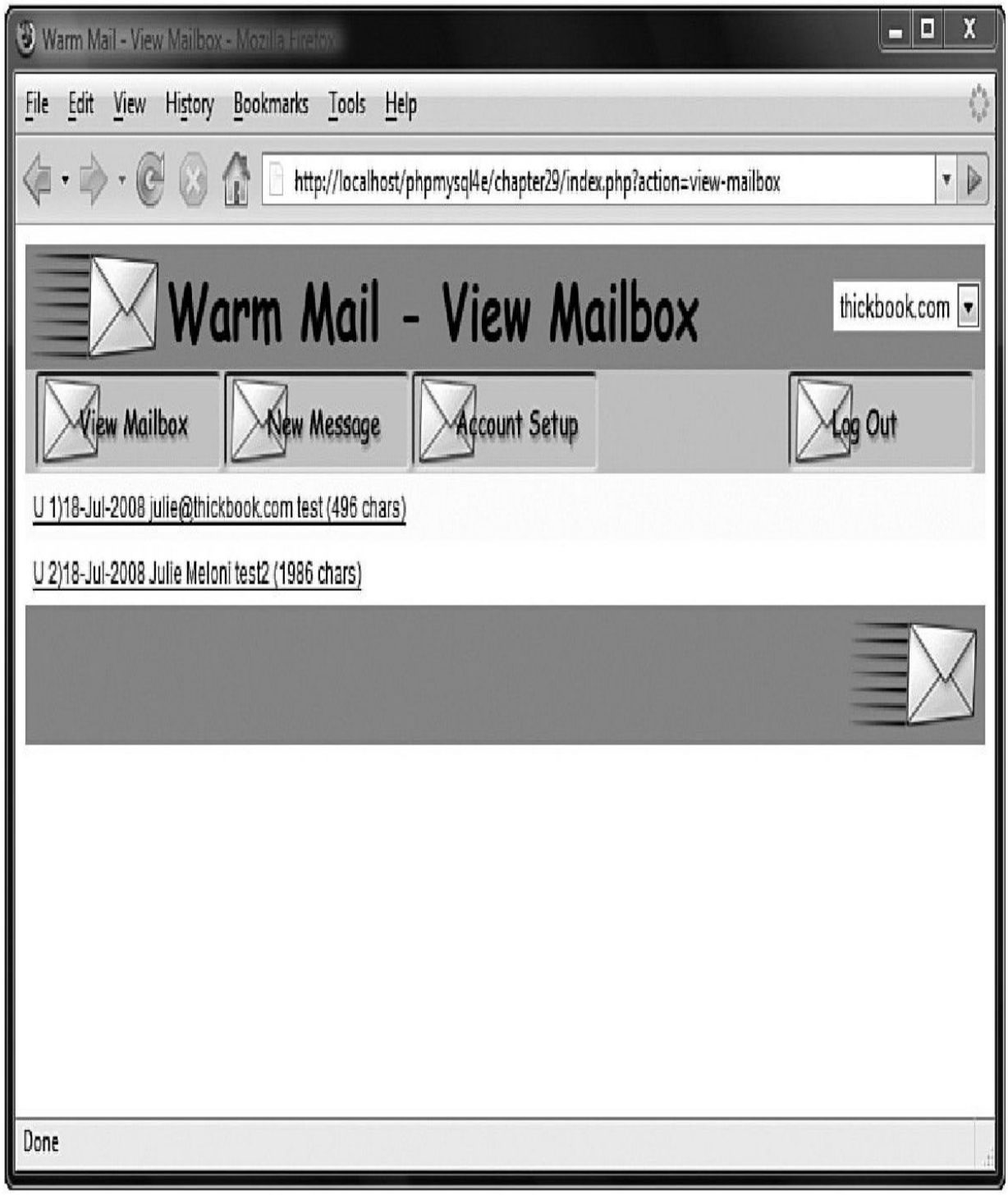


图　29-5　从列表框中选定一个账户后，该账户的邮件将被下载并显示出来

列表框的选项由output_fns.php文件中的do_html_header()函数生成，如下代码段所示：

```
//include the account select box only if the user has more than one account
if(number_of_accounts($auth_user)>1){
echo"<form action=\"index.php?action=open-mailbox\"method=\"post\">
<td bgcolor=\"#ff6600\"align=\"right\"valign=\"middle\">";
display_account_select($auth_user,$selected_account);
echo"</td>
</form>";
}
```

在本书的例子中，我们一直尽量避免讨论HTML，但是由函数display_account_select()生成的HTML值得我们仔细查看一下。

根据当前用户账户的不同，display_account_select()函数将生成如下所示的HTML代码：

```
<select
onchange="window.location=this.options[selectedIndex].value
name=account">
<option
value="index.php?action=select-account&account=4"selected>
thickbook.com
</option>
<option
value="index.php?action=select-account&account=3">
localhost
</option>
</select>
```

其中，大部分代码仅仅是一个HTML的select元素，当然，它也包括一些JavaScript。与PHP生成HTML的方式相同，JavaScript也可以用来生成客户端脚本。

对于该元素来说，无论什么时候发生了变化事件，JavaScript就将window.location选项赋值为当前的选项值。如果用户选择了列表框中的

第一项，window.location将被设置为"index.php?action=select-account &account=10"。这样就会使浏览器载入这个URL。很明显，如果用户使用的浏览器不支持JavaScript或禁用了JavaScript，这段代码将不会产生任何结果。

output_fns.php文件中的display_account_select()函数可以用来获得特定用户的所有可用账户列表并显示列表框。它也使用前面讨论过的get_account_list()函数。

在列表框中选择一个选项将触发select_account事件。观察图29-5中所示的URL，可以看到该事件和选中的账户ID都被添加到URL的末尾了。

在URL末尾添加这些参数产生两个作用。首先，在index.php的预处理阶段，选中的账户ID将被存储在会话变量$selected_account中，如下所示：

```
case'select-account':
//if have chosen a valid account,store it as a session variable
if(($account)&&(account_exists($_SESSION['auth_user'],
$account))){
$_SESSION['selected_account']=$account;
}
break;
```

其次，当执行到该脚本的主体部分时，将运行如下所示代码：

```
case'select-account':
case'view-mailbox':
//if mailbox just chosen,or view mailbox chosen,show mailbox
display_list($_SESSION['auth_user'],
$_SESSION['selected_account']);
break;
```

可以看到，在这里，我们执行了相同的动作，如同用户选择了查看邮箱选项一样。下面我们接着讨论。

29.7.2　查看邮箱内容

使用display_list()函数，可以查看邮箱的内容。该函数将显示邮箱中所有消息的列表。该函数的代码如程序清单29-8所示。

程序清单29-8　output_fns.php文件中的display_list()函数——显示所有邮箱消息

```
function display_list($auth_user,$accountid){
//show the list of messages in this mailbox
global$table_width;
if(!$accountid){
echo"<p style=\"padding-bottom:100px\">No mailbox selected.</p>";
}else{
$imap=open_mailbox($auth_user,$accountid);
if($imap){
echo"<table width=\"".$table_width."\"cellspacing=\"0\"
cellpadding=\"6\"border=\"0\">";
$headers=imap_headers($imap);
//we could reformat this data,or get other details using
//imap_fetchheaders,but this is not a bad summary so we
//just echo each
$messages=sizeof($headers);
for($i=0;$i<$messages;$i++){
echo"<tr><td bgcolor=\"";
if($i%2){
echo"#ffffff";
}else{
echo"#ffffcc";
}
echo"\"><a href=\"index.php?action=view-message&messageid="
.($i+1)."\">";
echo$headers[$i];
echo"</a></td></tr>\n";
```

```
}
echo"</table>";
}else{
$account=get_account_settings($auth_user,$accountid);
echo"<p style=\"padding-bottom:100px\">Could not open mail
box".$account['server']."</p>";
}
}
}
```

在这个函数中，我们实际上已开始使用了PHP的IMAP函数。该函数的两个关键部分是打开邮箱和阅读消息标题。

我们通过调用mail_fns.php文件中的open_mailbox()函数来打开用户的账户邮箱。函数代码如程序清单29-9所示。

程序清单29-9　mail_fns.php文件中的open_mailbox()函数——这个函数连接到用户邮箱

```
function open_mailbox($auth_user,$accountid){
//select mailbox if there is only one
if(number_of_accounts($auth_user)==1){
$accounts=get_account_list($auth_user);
$_SESSION['selected_account']=$accounts[0];
$accountid=$accounts[0];
}
//connect to the POP3 or IMAP server the user has selected
$settings=get_account_settings($auth_user,$accountid);
if(!sizeof($settings)){
return 0;
}
$mailbox='{'.$settings[server];
if($settings[type]=='POP3'){
$mailbox.='/pop3';
}
```

```
$mailbox.=':'.$settings[port].'}INBOX';
//suppress warning,remember to check return value
@$imap=imap_open($mailbox,$settings['remoteuser'],
$settings['remotepassword']);
return$imap;
}
```

实际上，我们是使用imap_open()函数来打开邮箱的。函数原型如下：

```
int imap_open(string mailbox,string username,string password[,int options])
```

该函数所需的参数如下所示：

■mailbox——这个字符串应包括服务器名和邮箱名，以及可选的端口号和协议。该字符串的格式如下所示：

```
{hostname/protocol:port}boxname
```

如果没有指定协议，则默认为IMAP。在我们编写的代码中，可以看到，如果用户为某一特定账户指定了POP3协议，我们就会指定为POP3。

例如，要使用默认端口从本地机器读取邮件，对于IMAP协议，我们使用如下所示的邮箱：

```
{localhost:143}INBOX
```

对于POP3协议，我们使用如下所示的邮箱：

```
{localhost/pop3:110}INBOX
```

■username——该账户的用户名。

■password——该账户的密码。

也可通过提交可选的标记来指定某些选项，例如，以只读方式打开邮箱。

值得注意的是，我们在将邮箱名称字符串传递给imap_open()函数打开之前，是一段一段地将它组织起来的。因此，在构造该串时，必须仔细一些，因为在PHP中包含“{$”的字符串会引起问题。

如果邮箱可以成功打开，该函数调用将返回一个IMAP流，如果不能成功打开则返回false。

当完成对一个IMAP流的操作时，可以调用imap_close(imap_stream)函数关闭它。在这个函数中，IMAP流被传回至主程序。接着，我们使用imap_headers()函数获得要显示的电子邮件标题：

```
$headers=imap_headers($imap);
```

这个函数将返回我们已经连接的邮箱所有消息的标题信息。这些信息以数组的形式返回，每条消息占一行。我们并没有对这些消息进行格式化处理。每条消息在输出中将占用一行，可从图29-5看到其输出形式。

可以通过使用imap_headers()函数来得到关于电子邮件标题的更多信息，该函数与imap_header()名称类似，因此易发生混淆。在本示例中，imap_headers()函数给出的详细信息对我们的项目目标来说已经足够了。

29.7.3 阅读邮件消息

我们已经将前面的display_list()函数返回的消息数组中的每一条消息连接到了特定的邮件消息。每个连接的格式如下所示：

```
index.php?action=view-message&messageid=6
```

messageid是我们前面在标题中得到的索引号。请注意，IMAP消息索引是从1开始的，而不是0。

如果用户点击了其中一个链接，将看到如图29-6所示的输出。

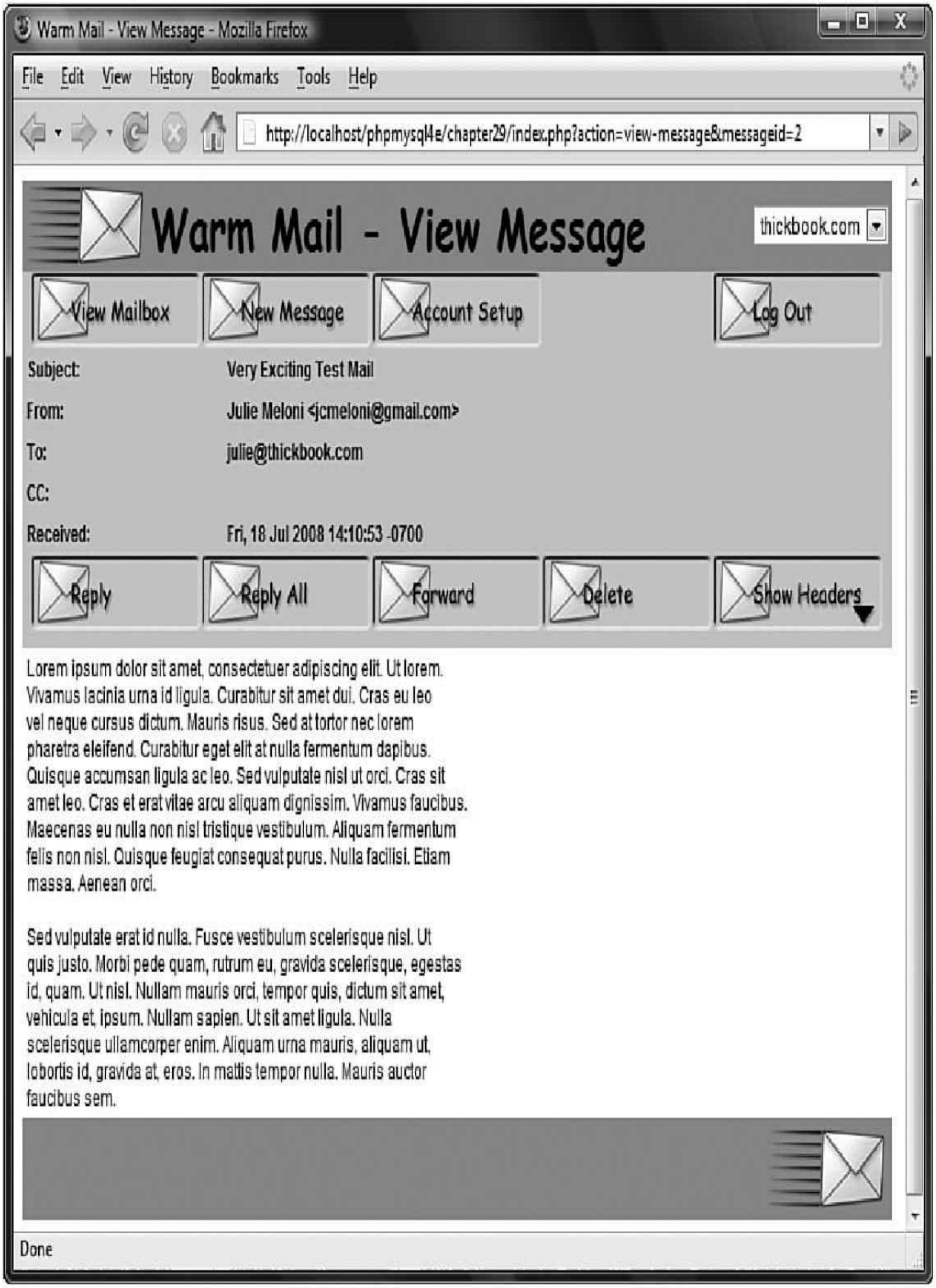

Warm Mail - View Message - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/phpmysql4e/chapter29/index.php?action=view-message&messageid=2
Warm Mail - View Message
thickbook.com
View Mailbox
New Message
Account Setup
Log Out
Subject: Very Exciting Test Mail
From: Julie Meloni <jcmeloni@gmail.com>
To: julie@thickbook.com
CC:
Received: Fri, 18 Jul 2008 14:10:53 -0700
Reply
Reply All
Forward
Delete
Show Headers
Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut lorem. Vivamus lacinia urna id ligula. Curabitur sit amet dui. Cras eu leo vel neque cursus dictum. Mauris risus. Sed at tortor nec lorem pharetra eleifend. Curabitur eget elit at nulla fermentum dapibus. Quisque accumsan ligula ac leo. Sed vulputate nisl ut orci. Cras sit amet leo. Cras et erat vitae arcu aliquam dignissim. Vivamus faucibus. Maecenas eu nulla non nisl tristique vestibulum. Aliquam fermentum felis non nisl. Quisque feugiat consequat purus. Nulla facilisi. Etiam massa. Aenean orci.
Sed vulputate erat id nulla. Fusce vestibulum scelerisque nisl. Ut quis justo. Morbi pede quam, rutrum eu, gravida scelerisque, egestas id, quam. Ut nisl. Nullam mauris orci, tempor quis, dictum sit amet, vehicula et, ipsum. Nullam sapien. Ut sit amet ligula. Nulla scelerisque ullamcorper enim. Aliquam urna mauris, aliquam ut, lobortis id, gravida at, eros. In mattis tempor nulla. Mauris auctor faucibus sem.
Done

图　29-6　用view-message动作来显示一个特定消息。在这个例子中，该消息是一封垃圾邮件

当我们将这些参数引入到index.php脚本后，将执行如下所示代码：

```
case'show-headers':
case'hide-headers':
case'view-message':
//if we have just picked a message from the list,or were looking at
//a message and chose to hide or view headers,load a message
$fullheaders=($action=='show-headers');
display_message($_SESSION['auth_user'],
$_SESSION['selected_account'],
$messageid,$fullheaders);
break;
```

这里，我们检查了$action的变量值是否等于"show-headers"。在这个例子中，该逻辑判断值为false，所以$fullheaders变量值被设置为false。稍后，我们再详细介绍"show-headers"动作。

如下代码行：

```
$fullheaders=($action=='show-headers');
```

如果按如下方式编写，可能会更繁琐一些，但是，却可能更清楚：

```
if($action=='show-headers'){
$fullheaders=true;
}else{
$fullheaders=false;
}
```

接下来，我们将调用display_message()函数。该函数的输出大部分都是纯HTML，因此在这里，我们将不做详细介绍。调用retrieve_message()函数从邮箱中检索适当的消息：

```
$message=retrieve_message($auth_user,$accountid,$messageid,$fullheaders);
```

retrieve_message()函数在mail_fns.php库中，其代码如程序清单29-10所示。

程序清单29-10　mail_fns.php文件的retrieve_message()函数——该函数将从邮箱中取得指定的消息

```
function retrieve_message($auth_user,$accountid,$messageid,
$fullheaders){
$message=array();
if(!($auth_user&&$messageid&&$accountid)){
return false;
}
$imap=open_mailbox($auth_user,$accountid);
if(!$imap){
return false;
}
$header=imap_header($imap,$messageid);
if(!$header){
return false;
}
$message['body']=imap_body($imap,$messageid);
if(!$message['body']){
$message['body']="[This message has no body]\n\n\n\n\n\n";
}
if($fullheaders){
$message['fullheaders']=imap_fetchheader($imap,$messageid);
}else{
$message['fullheaders']='';
}
```

```
$message['subject']=$header->subject;
$message['fromaddress']=$header->fromaddress;
$message['toaddress']=$header->toaddress;
$message['ccaddress']=$header->ccaddress;
$message['date']=$header->date;
//note we can get more detailed information by using from and to
//rather than fromaddress and toaddress,but these are easier
imap_close($imap);
return$message;
}
```

我们将再次使用open_mailbox()函数打开用户的邮箱。但是，这次我们是查找一个指定的消息，使用该函数库，我们可以分别下载消息标题与消息正文。

在这里，我们使用的3个IMAP函数分别是imap_header()、imap_fetchheader()和imap_body()。请注意，这两个标题函数不同于我们前面使用的imap_headers()。它们的名字在某种程度上容易造成混淆。简而言之：

■imap_headers()——返回邮箱中所有消息的标题摘要。每条消息作为数组的一个元素，并且以数组形式返回。

■imap_header()——以对象的形式返回一条特定消息的标题。

■imap_fetchheader()——以字符串的形式返回一条特定消息的标题。

在这个例子中，我们使用imap_header()函数来填充指定的标题域，用imap_fetchheader()函数来显示用户请求的完整标题（稍后，我们还将详细介绍这一点）。

我们使用imap_header()函数和imap_body()函数建立一个包括我们感兴趣的消息的所有元素的数组。我们以如下方式调用imap_header()函数：

```
$header=imap_header($imap,$messageid);
```

可以从这个对象中获得所请求的每一个域：

```
$message['subject']=$header->subject;
```

调用imap_body()函数可以将消息体添加到该数组，如下所示：

```
$message['body']=imap_body($imap,$messageid);
```

最后，我们将调用imap_close()函数关闭邮箱并返回已经创建的数组。display_message()函数可以在表单中显示消息的域，如图29-6所示。

29.7.4　查看消息标题

如图29-6所示，可以看到一个"Show Headers"按钮。这个按钮可以触发"show-headers"选项，它会将该消息的完整标题加入到消息显示中。如果用户点击此按钮，可以看出如图29-7所示的页面。

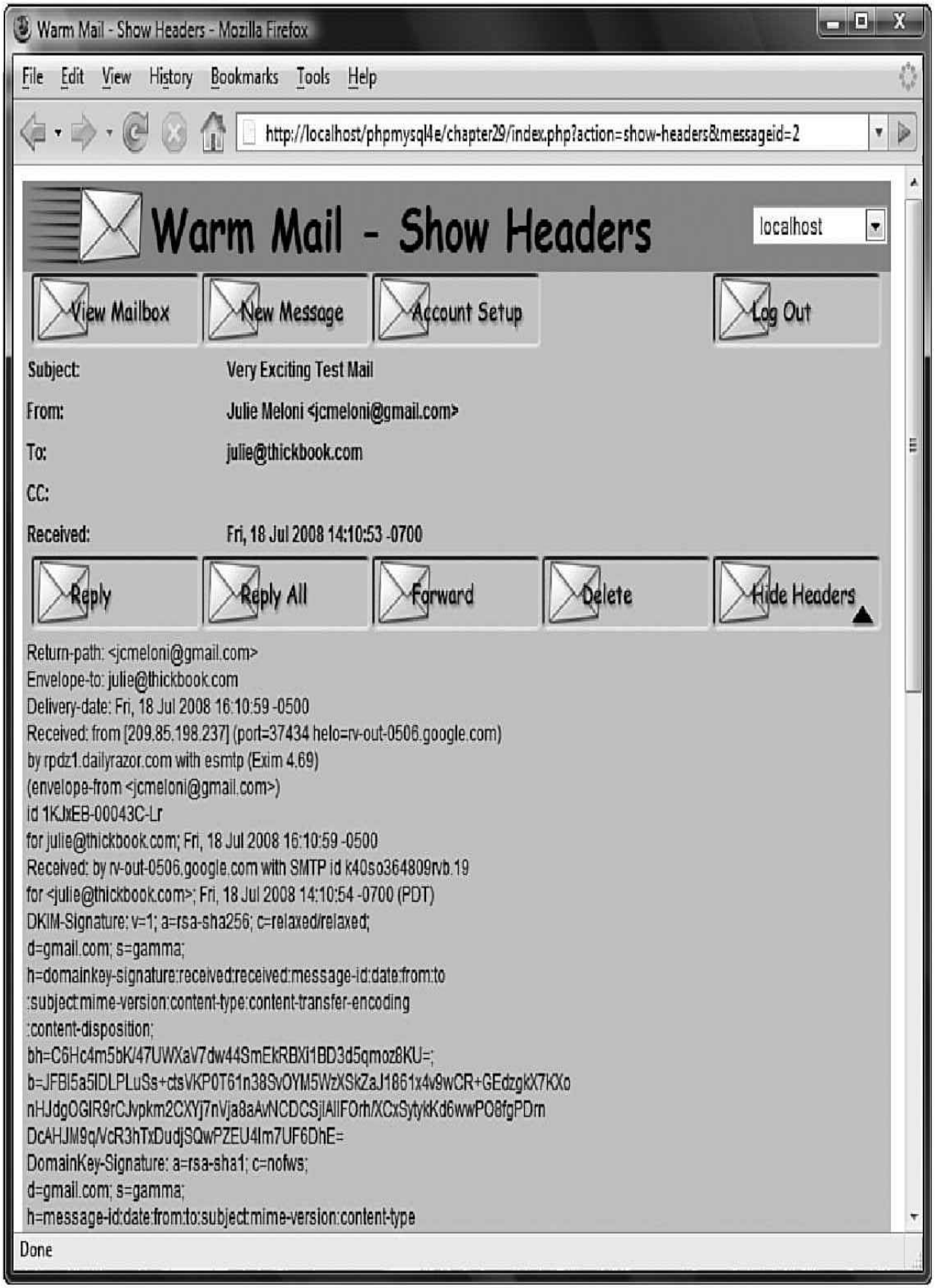

Warm Mail - Show Headers - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/phpmysql4e/chapter29/index.php?action=show-headers&messageid=2
Warm Mail - Show Headers
localhost
View Mailbox
New Message
Account Setup
Log Out
Subject: Very Exciting Test Mail
From: Julie Meloni <jcmeloni@gmail.com>
To: julie@thickbook.com
CC:
Received: Fri, 18 Jul 2008 14:10:53 -0700
Reply
Reply All
Forward
Delete
Hide Headers
Return-path: <jcmeloni@gmail.com>
Envelope-to: julie@thickbook.com
Delivery-date: Fri, 18 Jul 2008 16:10:59 -0500
Received: from [209.85.198.237] (port=37434 helo=rv-out-0506.google.com)
by rpdz1.dailyrazor.com with esmtp (Exim 4.69)
(envelope-from <jcmeloni@gmail.com>)
id 1KJxEB-00043C-Lr
for julie@thickbook.com; Fri, 18 Jul 2008 16:10:59 -0500
Received: by rv-out-0506.google.com with SMTP id k40so364809rvb.19
for <julie@thickbook.com>; Fri, 18 Jul 2008 14:10:54 -0700 (PDT)
DKIM-Signature; v=1; a=rsa-sha256; c=relaxed/relaxed;
d=gmail.com; s=gamma;
h=domainkey-signature:received:received:message-id:date:from:to
:subject:mime-version:content-type:content-transfer-encoding
:content-disposition;
bh=C6Hc4m5bKJ47UWXaV7dw44SmEkRBXi1BD3d5qmoz8KU=;
b=JFBl5a5IDLPLuSs+ctsVKP0T61n38SvOYM5WzXSkZaJ1861x4v9wCR+GEdzgkX7KXo
nHJdgOGIR9rCJvpkm2CXYj7nVja8aAvNCDCSjIAlFOrh/XCxSytykKd6wwPO8fgPDrn
DcAHJM9q/VcR3hTxDudjSQwPZEU4Im7UF6DhE=
DomainKey-Signature: a=rsa-sha1; c=nofws;
d=gmail.com; s=gamma;
h=message-id:date:from:to:subject:mime-version:content-type
Done

图　29-7　使用"show-headers"选项查看消息的完整标题将帮助用户跟踪垃圾邮件的来源

注意view-message的事件处理也包括show-headers（和与之对应的hide-headers）。如果选择了该选项，会和前面一样执行相同的操作。但在retrieve_message()函数中，也可得到标题的完整文本，如下所示：

```
if($fullheaders){
$message['fullheaders']=imap_fetchheader($imap,$messageid);
}
```

然后我们可以将这些标题显示给用户。

29.7.5　删除邮件

如果用户点击某一特定邮件上的“删除”按钮，将激活"delete"操作。该操作执行index.php脚本中的如下代码：

```
case'delete':
delete_message($_SESSION['auth_user'],
$_SESSION['selected_account'],$messageid);
//note deliberately no'break'-we will continue to the next case
case'select-account':
case'view-mailbox':
//if mailbox just chosen,or view mailbox chosen,show mailbox
display_list($_SESSION['auth_user'],
$_SESSION['selected_account']);
break;
```

可以看到，我们使用delete_message()函数删除消息，然后，像前面讨论的那样显示删除消息后的邮箱。delete_message()函数代码如程序清单29-11所示。

程序清单29-11　来自mail_fns.php的delete_message()函数——从邮箱中删除一个指定消息的函数

```
function delete_message($auth_user,$accountid,$message_id){
//delete a single message from the server
$imap=open_mailbox($auth_user,$accountid);
if($imap){
imap_delete($imap,$message_id);
imap_expunge($imap);
imap_close($imap);
return true;
}
return false;
}
```

可以看到，该函数使用了几个IMAP函数。其中，我们还没有使用过的函数是imap_delete()和imap_expunge()。请注意，imap_delete()函数只是将邮件标记为删除。我们可以将任意数量的邮件标记为删除。imap_expunge()函数的调用才真正删除消息。

29.8 发送邮件

最后，我们来介绍如何发送邮件。该脚本程序中有多种方法可以完成此操作：用户可以发送一则新的邮件消息，回复或者转发邮件。我们来看看它们是如何工作的。

29.8.1 发送一则新消息

点击"New Message"按钮，用户可以选择发送一则新的邮件消息。这将触发"new-message"操作，该操作将执行index.php/case'New-Message'脚本中的如下代码：

```
index.php/case'new-message':
display_new_message_form($_SESSION['auth_user'],
$to,$cc,$subject,$body);
break;
```

新消息表单仅仅是一个供发送邮件用的表单，如图29-8所示。实际上，该图显示的是回复邮件而非新建邮件，但是用到的表单都是相同的。接下来，我们将了解如何转发和回复邮件。

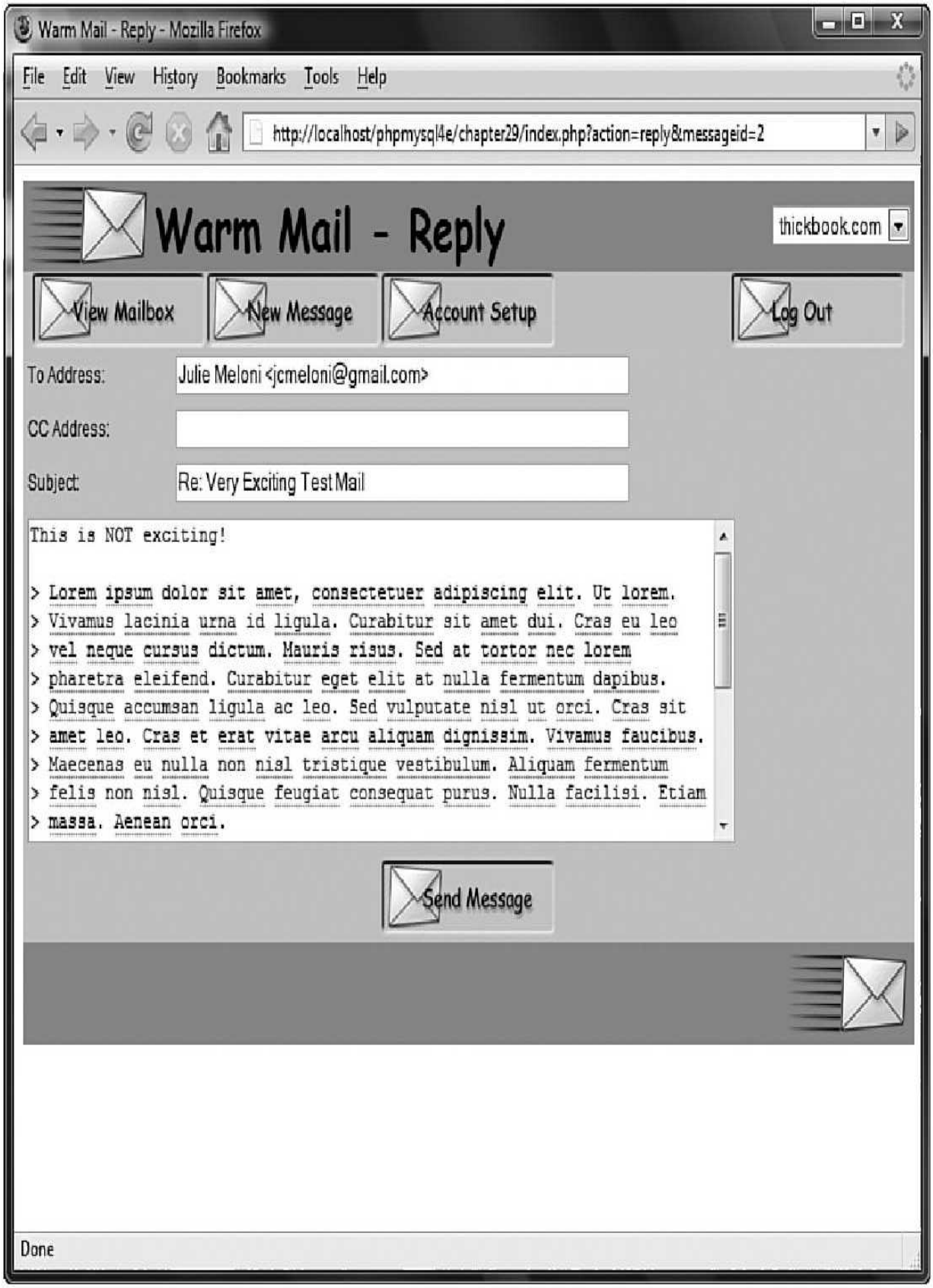

Warm Mail - Reply - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/phpmysql4e/chapter29/index.php?action=reply&messageid=2
Warm Mail - Reply
thickbook.com
View Mailbox
New Message
Account Setup
Log Out
To Address:
Julie Meloni <jcmeloni@gmail.com>
CC Address:
Subject:
Re: Very Exciting Test Mail
This is NOT exciting!
> Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut lorem.
> Vivamus lacinia urna id ligula. Curabitur sit amet dui. Cras eu leo
> vel neque cursus dictum. Mauris risus. Sed at tortor nec lorem
> pharetra eleifend. Curabitur eget elit at nulla fermentum dapibus.
> Quisque accumsan ligula ac leo. Sed vulputate nisl ut orci. Cras sit
> amet leo. Cras et erat vitae arcu aliquam dignissim. Vivamus faucibus.
> Maecenas eu nulla non nisl tristique vestibulum. Aliquam fermentum
> felis non nisl. Quisque feugiat consequat purus. Nulla facilisi. Etiam
> massa. Aenean orci.
Send Message
Done

图　29-8　你可以将邮件回复或转发给其他用户

点击"Send Message"按钮将调用"send-message"操作，该操作将执行index.php脚本中的如下代码：

```
case'send-message':
if(send_message($to,$cc,$subject,$message)){
echo"<p style=\"padding-bottom:100px\">Message sent.</p>";
}else{
echo"<p style=\"padding-bottom:100px\">Could not send message.</p>";
}
```

以上代码调用了send_message()函数，该函数真正执行邮件发送的操作。函数如程序清单29-12所示。

程序清单29-12　mail_fns.php的send_message()函数——发送用户键入的消息的函数

```
function send_message($to,$cc,$subject,$message){
//send one email via PHP
if(!$conn=db_connect()){
return false;
}
$query="select address from users where
username='".$_SESSION['auth_user']."'";
$result=$conn->query($query);
if(!$result){
return false;
}else if($result->num_rows==0){
return false;
}else{
$row=$result->fetch_object();
$other='From:'.$row->address;
if(!empty($cc)){
$other.="\r\nCc:$cc";
```

```
}
if(mail($to,$subject,$message,$other)){
return true;
}else{
return false;
}
}
}
```

可以看到，该函数使用了mail()函数来发送邮件，但是，首先该函数将从数据库中载入用户的电子邮件地址，并将其设置在待发邮件的From域。

29.8.2 回复或转发邮件

"Reply"、"Reply All"和"Forward"功能发送邮件的方法跟发送新邮件的方法相同。其不同之处在于，在将邮件发送表单显示给用户之前，系统已经自动填写好了某些域。回顾图29-8。我们要回复的消息正文已经用“>”符号缩进了，主题行以"Re:"为开始。同样，"Forward"和"Reply All"会预先填好接收者、主题行和缩进的消息正文。

完成这些操作的代码都是由index.php脚本的主体部分触发，如下所示：

```
case'reply-all':
//set cc as old cc line
if(!$imap){
$imap=open_mailbox($_SESSION['auth_user'],
$_SESSION['selected_account']);
}
if($imap){
$header=imap_header($imap,$messageid);
if($header->reply_toaddress){
$to=$header->reply_toaddress;
}else{
$to=$header->fromaddress;
}
$cc=$header->ccaddress;
$subject="Re:".$header->subject;
$body=add_quoting(stripslashes(imap_body($imap,$messageid)));
imap_close($imap);
display_new_message_form($_SESSION['auth_user'],
$to,$cc,$subject,$body);
}
break;
case'reply':
```

```
//set to address as reply-to or from of the current message
if(!$imap){
$imap=open_mailbox($_SESSION['auth_user'],
$_SESSION['selected_account']);
}
if($imap){
$header=imap_header($imap,$messageid);
if($header->reply_toaddress){
$to=$header->reply_toaddress;
}else{
$to=$header->fromaddress;
}
$subject="Re:".$header->subject;
$body=add_quoting(stripslashes(imap_body($imap,$messageid)));
imap_close($imap);
display_new_message_form($_SESSION['auth_user'],
$to,$cc,$subject,$body);
}
break;
case'forward':
//set message as quoted body of current message
if(!$imap){
$imap=open_mailbox($_SESSION['auth_user'],
$_SESSION['selected_account']);
}
if($imap){
$header=imap_header($imap,$messageid);
$body=add_quoting(stripslashes(imap_body($imap,$messageid)));
$subject="Fwd:".$header->subject;
imap_close($imap);
display_new_message_form($_SESSION['auth_user'],
$to,$cc,$subject,$body);
}
break;
```

可以看到，每一个选项都将建立相应的标题，必要时还会对标题进行格式化处理，然后再调用display_new_message_form()函数来建立表单。

这就是我们的Web邮件阅读系统的整个功能。

29.9 扩展这个项目

我们可以对该项目做出很多扩展和改进。我们可以观察平常使用的邮件阅读器并从中受到启发，下面是一些非常有用的补充。

■为用户增加在该站点注册的功能。（关于用户注册功能，可重复利用第27章“建立用户身份验证机制和个性化设置”的一些代码。）

■为用户提供具有多个邮件地址的功能。很多用户有多个电子邮件地址；例如可能一个用于个人，一个用于工作。通过把存储的电子邮件地址从用户表转移到账户表，用户能够使用多个地址。我们还需要在其他代码中进行有限的修改。发送邮件表单需要有一个下拉列表框来选择使用哪一个地址。

■增加发送、接收、查看附件的功能。如果用户想要发送附件，就必须增加如第19章“与文件系统和服务器的交互”中讨论过的文件上传功能。而发送带有附件的邮件则在第30章“创建一个邮件列表管理器”中讨论。

■增加地址簿功能。

■增加网络新闻阅读功能。使用IMAP函数从一个NNTP服务器阅读新闻和从邮箱阅读邮件几乎是相同的。只需要在imap_open()函数调用中指定一个不同的端口号和协议。我们不是命名一个像INBOX一样的邮箱，而是命名一个想要获得新闻的新闻组。我们可以将此特性与第31章“创建一个Web论坛”项目中用到的创建主题功能结合起来，创建一个基于Web的线程化新闻阅读器。

29.10 下一章

在下一章中，我们将创建另外一个与电子邮件相联系的项目——在这个项目中，我们将创建一个支持向多个主题发送新闻信件的应用程序，它将新闻发送给站点的订阅用户。

第30章　创建一个邮件列表管理器

站点建立了相当数量的订阅客户群之后，可以通过给他们发送新闻信件保持联系。在本章中，我们将实现一个邮件列表管理器（MLM）的前台程序。有些MLM允许每个订阅者给其他订阅者发送消息。我们在本章所创建的程序将是一个新闻信件系统，在这个系统中，只有列表管理员才能发送消息。我们把这个系统叫做金字塔式MLM。

这个系统与市场上其他已有系统相似。如果想了解系统的目标，可以访问如下站点：http://www.topica.com。

我们的程序允许管理员创建多个邮件列表，而且可以单独将新闻信件发送给每一个列表。该程序使用文件上载让管理员可以上载脱机创建的文本或HTML格式的新闻信件。这就意味着管理员可以使用任何自己喜欢的软件来创建新闻信件。

用户可以订阅站点上的任何一个列表，并且选择以文本形式还是HTML格式接收新闻信件。

在本章中，我们将主要介绍以下内容：

■多个文件的文件上载

■Mime编码的电子邮件附件

■HTML格式的电子邮件

■不需要人工交互的用户密码管理方法

30.1　解决方案的组成

我们希望建立一个在线新闻信件创作与发送系统。这个系统应该允许创作各种各样的新闻信件发送给用户，并且允许用户订阅这些新闻信件中的一种或多种。

特别地，该解决方案将满足如下所示的目标。

■管理员应该能够建立和修改邮件列表。

■管理员应该能够将文本或HTML格式的新闻信件发送给一个列表中的所有订阅者。

■用户应该能够通过注册使用一个站点，并且可以进入并修改他们的个人资料。

■用户应该能够订阅该站点的任意一个列表的新闻信件。

■用户应该能够取消一个邮件列表的订阅。

■用户应该能够据个人喜好以HTML格式或纯文本格式存储新闻信件。

■出于安全的原因，用户应该不能将邮件发送到列表，或者不能看见其他用户的邮件地址。

■用户和管理员应该能够查看有关邮件列表的信息。

■用户和管理员应该能够查看过去已经发送给某个列表（存档文件）上的新闻信件。

到这里，我们已经对项目有了基本了解，就可以开始设计解决方案及其组成部分，例如，建立一个列表、订阅者和存档新闻信件的数据库，上载脱机创建的新闻信件；以及发送带附件的邮件。

30.1.1 建立列表和订阅者数据库

在这个项目中，我们除了记录系统用户订阅的列表外，还要记录用户的用户名和密码。当然，也要记录每个用户的个人选项：接收文本格式还是HTML格式的邮件，从而能够以适当的格式向用户发送新闻信件。

管理员是一个特殊的用户，他拥有创建新邮件列表和将新闻信件发送给该列表用户的权限。

对过去的新闻信件进行存档也是该系统一个很好的功能。订阅者可能没有保存以前的新闻信件，但是订阅者可能想要查找其中的某些东西。存档文件还可以作为新闻信件的一种市场推广工具，因为订阅者可以看到以前的新闻信件的主要内容。

在MySQL中建立该数据库，并且在PHP中创建访问MySQL的界面是非常简单的，没有什么新内容，也没有什么困难。

30.1.2 上载新闻信件

正如前面提到的，我们需要一个管理员界面来发送新闻信件。目前，我们还没有讨论过的内容是管理员如何创建新闻信件。系统可以为管理员提供一个表单，在这个表单中，管理员可以输入或者粘贴新闻信件内容。然而，让管理员在他喜爱的编辑器中创建新闻信件然后再上载到Web服务器，将增加系统的用户友好性。这也会使得管理员将图像加入到HTML新闻信件中变得更加简单。因此，我们可以使用在第19章“与文件系统和服务器的交互”中讨论的文件上载功能。

我们使用一个比前面项目中所用过的更加复杂的表单。对于这个项目，我们要求管理员上载文本格式和HTML格式的新闻信件，同时包括嵌入到HTML中的所有图像。

上载新闻信件后，需要建立一个界面，以便管理员在发送之前浏览该新闻信件。这样可以确保所有的文件都被正确地上载了。

请注意，系统也可以将所有这些文件保存在一个存档目录中，这样用户可以重复阅读新闻信件。这个目录必须具有Web服务器用户的写入权限。上载脚本尝试将新闻信件写入./archive/目录，这样我们必须确保创建了该目录并且正确设置了该目录的权限。

30.1.3　发送带附件的邮件

在这个项目中，我们想要根据用户喜好，既能够向用户发送纯文本格式的新闻信件，也能够发送“别致的”HTML格式的新闻信件。

要发送一个嵌入了图像的HTML文件，需要找到一种方法来发送附件。PHP简单的mail()函数并不能直接支持附件的发送。取而代之的是，我们将使用PEAR中的Mail_Mime包（该包最初由Richard Heyes编写）。它可以处理HTML附件，而且可以自动附加任何包含在HTML文件中的图像。

在附录A“安装PHP和MySQL”的“PEAR安装”一节中，给出了该包的安装说明。

30.2　解决方案概述

正如我们在第29章中所做的一样，对于这个项目，我们将再次使用事件驱动的方法来编写代码。

我们仍然将以系统流程图的绘制开始，绘制了用户使用系统的可能方式。在这个例子中，我们绘制了3幅代表用户与系统可能进行的3种不同交互方式的流程图。当用户没有登录时，用户拥有不同的权限。而当用户以普通用户或管理员身份登录时，他们又具有不同的权限。这些权限分别如图30-1、图30-2和图30-3所示。

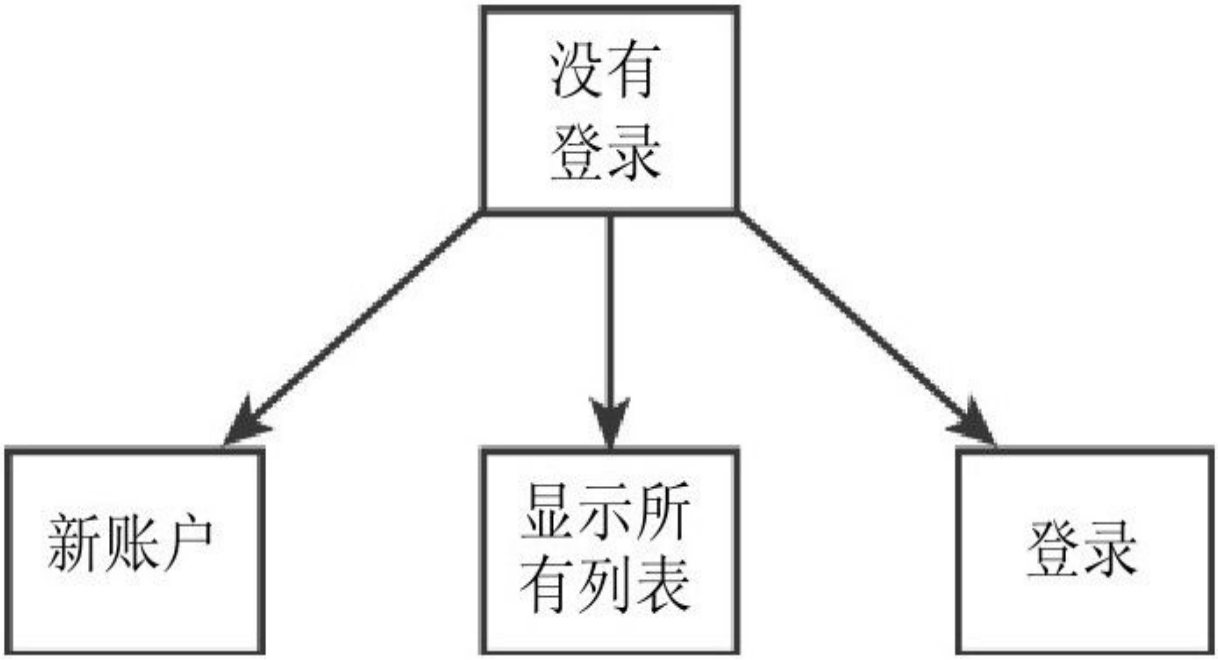


图　30-1　当一个用户没有登录时，他只能进行有限的操作

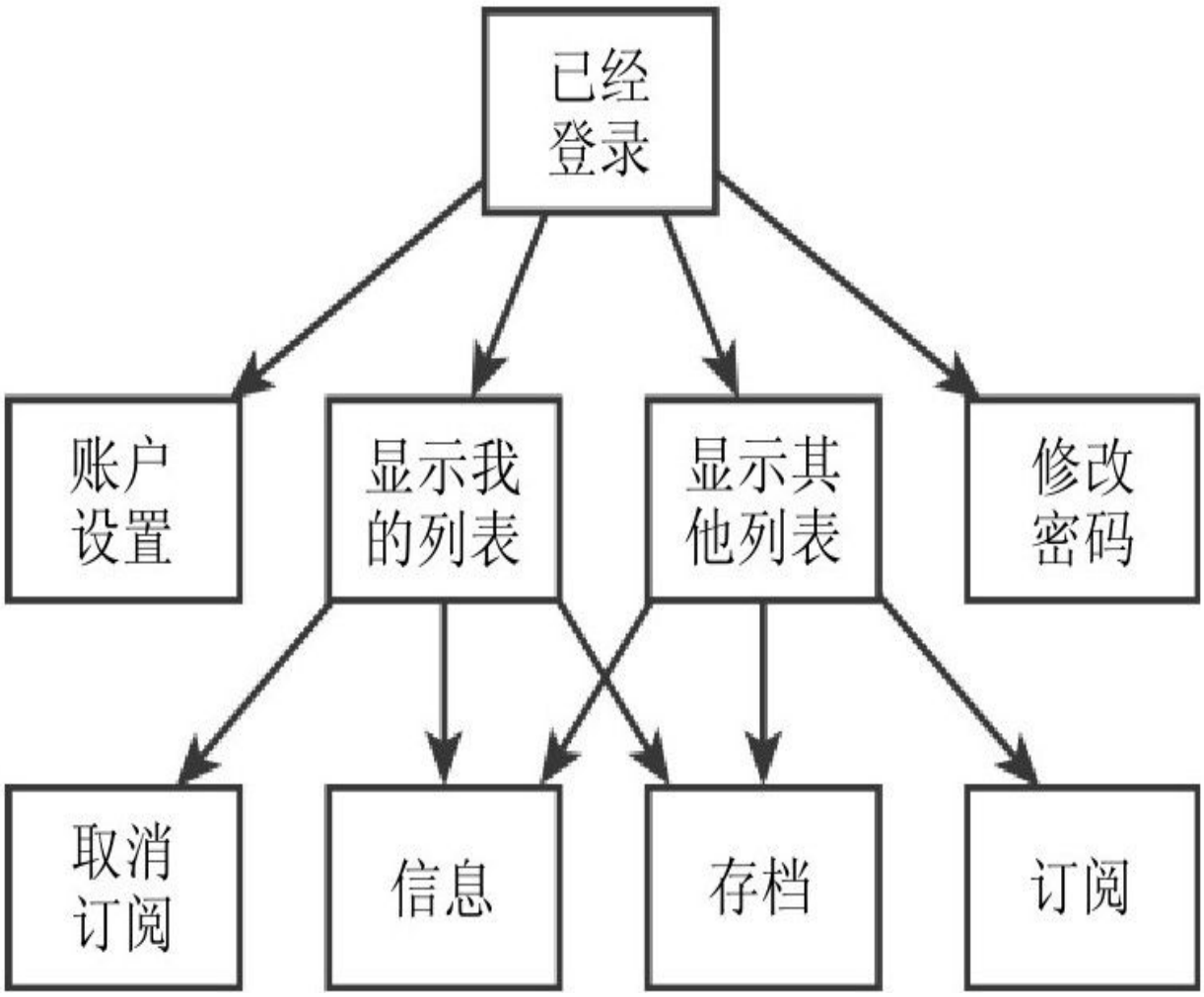


图　30-2　用户登录后，可以通过不同的选项修改个人喜好

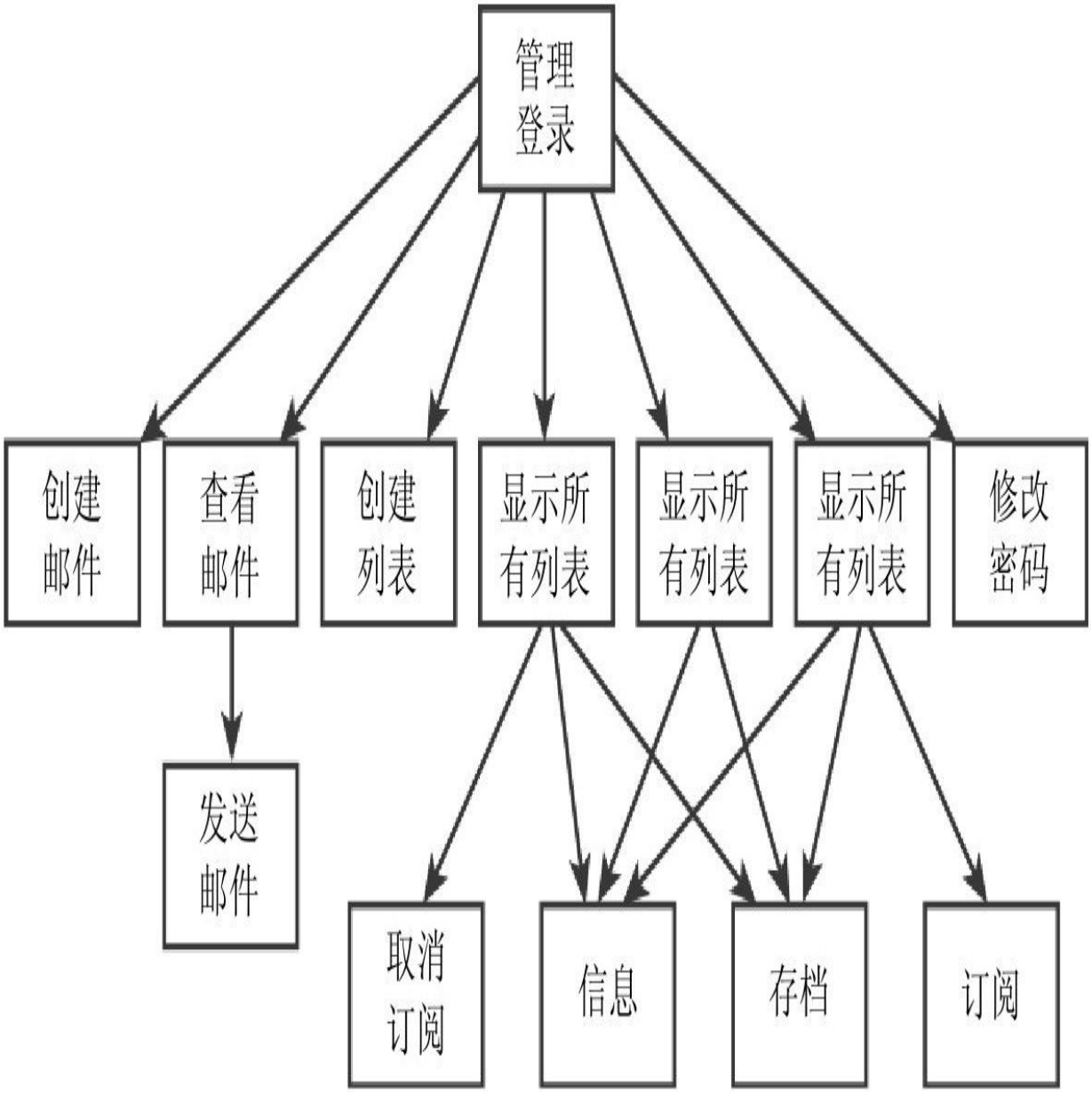


图　30-3　管理员拥有专用的操作

在图30-1中，可以看到没有登录的用户所能进行的操作。可以看到，该用户可以登录（如果已经有一个账户），创建一个账户（如果还没有账户），或者查看可供订阅的邮件列表（作为一种市场营销策略）。

图30-2显示了用户登录后可以进行的操作。用户可以修改账户设置（邮件地址和个人喜好），更改密码，修改所订阅的邮件列表。

图30-3显示了系统管理员登录后可以进行的操作。可以看到，系统管理员除了拥有大多数普通用户拥有的功能，还具有一些额外选项。管理员还可以创建新的邮件列表，通过上载文件为某个邮件列表创建新邮件，并且在发送这些邮件前对它们进行预览。

由于我们再次使用了事件驱动的方法，因此应用程序的架构包含在文件index.php中，由它调用一系列函数库。该程序用到的文件总览如表30-1所示。

我们将以自己的方式来完成项目，以创建存储订阅者和列表信息的数据库开始。

表30-1　邮件列表管理器应用程序所用到的文件

文件名称	类　型	描　述
index.php	应用程序	运行整个程序的主要脚本
include_fns.php	函数	该程序包含文件的集合
data_valid_fns.php	函数	验证输入数据的函数集合
db_fns.php	函数	连接到mlm数据库的函数集合
mlm_fns.php	函数	该应用程序专用的函数集合
output_fns.php	函数	输出HTML的函数集合
upload.php	组件	该脚本对管理员使用的文件上载组件进行管理。单独列出来以简化安全机制
user_auth_fns.php	函数	用户验证的函数集合
create_database.sql	SQL	用来设置mlm数据库，设置Web用户和管理员用户的SQL

30.3　建立数据库

对于这个应用程序，需要保存如下所示的细节信息。

■Lists（列表）：可供订阅的邮件列表。

■Subscribers（订阅者）：系统用户和他们的个人喜好信息。

■Sublists（订阅列表）：记录哪些用户订阅了哪些邮件列表（多对多的关系）。

■Mail（邮件）：已经发送的邮件消息的记录。

■Images（图像）：由于我们想要发送由多个文件组成的邮件消息，必须跟踪每个邮件附带的文本、HTML和图像。

用来创建该数据库的SQL如程序清单30-1所示。

程序清单30-1　create_database.sql——创建mlm数据库的SQL

```
create database mlm;
use mlm;
create table lists
(
listid int auto_increment not null primary key,
listname char(20)not null,
blurb varchar(255)
);
create table subscribers
(
email char(100)not null primary key,
realname char(100)not null,
mimetype char(1)not null,
password char(40)not null,
admin tinyint not null
```

```
);
#stores a relationship between a subscriber and a list
create table sub_lists
(
email char(100)not null,
listid int not null
);
create table mail
(
mailid int auto_increment not null primary key,
email char(100)not null,
subject char(100)not null,
listid int not null,
status char(10)not null,
sent datetime,
modified timestamp
);
#stores the images that go with a particular mail
create table images
(
mailid int not null,
path char(100)not null,
mimetype char(100)not null
);
grant select,insert,update,delete
on mlm.*
to mlm@localhost identified by'password';
insert into subscribers values
('admin@localhost','Administrative User','H',sha1('admin'),1);
```

注意可以通过输入如下命令来执行这个SQL脚本：

```
mysql-u root-p＜create_database.sql
```

必须提供root用户密码（当然，也可以通过任何具有适当权限的MySQL用户来执行该脚本；这里，我们为了简便仅使用root用户）。在运行这个脚本之前，应该修改该脚本中的mlm用户和管理员的密码。

该数据库的某些字段需要稍作解释，我们简单地看一下。lists表包括listid字段和listname字段。它还包括一个blurb字段，它描述了该列表的主要内容。

subscribers表包括电子邮件地址（E-mail）字段和订阅者的名字（realname）字段。该表还保存了用户密码和一个用来表明该用户是否是管理员的标记（admin）字段。我们还要在mimetype字段中存储用户想要接收的邮件类型。这里，可以用H代表HTML或T代表文本。

sublists表包括关于来自subscribers表的邮件地址（E-mail）和来自lists表的listids。

mail表包括关于通过系统发送的每条邮件消息的信息。它保存了唯一标识（mailid）、邮件从何处发送而来的地址（E-mail）、邮件的主题行（subject）和已经发送的或将要被发送的列表listid。消息的实际文本或HTML版本可能是一个大文件，因此我们在数据库之外存储实际消息的存档文件。我们还保存一些常规状态信息：消息是否已经被发送了（status），什么时候发送的（sent），以及显示该记录最后一次修改的时间戳（modified）。

最后，我们用images表来记录任何与HTML消息有关的图像。同样，这些图像可能非常大，因此出于效率的考虑，我们将它们保存在数据库之外。我们需要记录与之相关联的mailid、图像实际存储位置的路径和图像的MIME类型（mimetype），例如image/gif。

前面所示的SQL脚本还将为PHP创建一个用户，通过这个用户，PHP可以连接数据库，同时，该用户还是系统的管理用户。

30.4　定义脚本架构

与前一个项目一样，对于这个项目，我们也采用了事件驱动的方法。该程序的架构保存在文件index.php中。该脚本有4个主要的部分，它们分别如下。

1.执行预处理：标题发送前必须完成的所有处理工作。

2.建立和发送标题：创建与发送HTML页的开始部分。

3.执行一个动作：响应已经传入的事件。正如前一个例子一样，事件包含在$action变量中。

4.发送页脚。

几乎所有的应用程序处理都是在这个文件中完成的。该程序还使用了表30-1中列出的函数库，正如前面所介绍的。

index.php脚本的所有代码如程序清单30-2所示。

程序清单30-2　index.php——金字塔式MLM的主要应用程序文件

```
<?php
/*********************************************************************
*Section 1:pre-processing
*********************************************************************/
include('include_fns.php');
session_start();
$action=$_GET['action'];
$buttons=array();
//append to this string if anything processed before header has output
$status='';
//need to process log in or out requests before anything else
if(($_POST['email'])&&($_POST['password'])){
$login=login($_POST['email'],$_POST['password']);
if($login=='admin'){
```

```
$status.="<p style=\"padding-bottom:50px\">
<strong>".get_real_name($_POST['email'])."</strong>
logged in successfully as
<strong>Administrator</strong>.</p>";
$_SESSION['admin_user']=$_POST['email'];
}else if($login=='normal'){
$status.="<p style=\"padding-bottom:50px\">
<strong>".get_real_name($_POST['email'])."</strong>
logged in successfully.</p>";
$_SESSION['normal_user']=$_POST['email'];
}else{
$status.="<p style=\"padding-bottom:50px\">Sorry,we could
not log you in with that email address
and password.</p>";
}
}
if($action=='log-out'){
unset($action);
$_SESSION=array();
session_destroy();
}
/**********************************************************************
*Section 2:set up and display headers
**********************************************************************/
//set the buttons that will be on the tool bar
if(check_normal_user()){
//if a normal user
$buttons[0]='change-password';
$buttons[1]='account-settings';
$buttons[2]='show-my-lists';
$buttons[3]='show-other-lists';
$buttons[4]='log-out';
}else if(check_admin_user()){
//if an administrator
$buttons[0]='change-password';
$buttons[1]='create-list';
```

```
$buttons[2]='create-mail';
$buttons[3]='view-mail';
$buttons[4]='log-out';
$buttons[5]='show-all-lists';
$buttons[6]='show-my-lists';
$buttons[7]='show-other-lists';
}else{
//if not logged in at all
$buttons[0]='new-account';
$buttons[1]='show-all-lists';
$buttons[4]='log-in';
}
if($action){
//display header with application name and description of page or action
do_html_header('Pyramid-MLM-'.format_action($action));
}else{
//display header with just application name
do_html_header('Pyramid-MLM');
}
display_toolbar($buttons);
//display any text generated by functions called before header
echo$status;
/**********************************************************************
*Section 3:perform action
**********************************************************************/
//only these actions can be done if not logged in
switch($action){
case'new-account':
//get rid of session variables
session_destroy();
display_account_form();
break;
case'store-account':
if(store_account($_SESSION['normal_user'],
$_SESSION['admin_user'],$_POST)){
$action='';
```

```
}
if(!check_logged_in()){
display_login_form($action);
}
break;
case'log-in':
case'':
if(!check_logged_in()){
display_login_form($action);
}
break;
case'show-all-lists':
display_items('All Lists',get_all_lists(),'information',
'show-archive','');
break;
case'show-archive':
display_items('Archive For'.get_list_name($_GET['id']),
get_archive($_GET['id']),'view-html',
'view-text','');
break;
case'information':
display_information($_GET['id']);
break;
}
//all other actions require user to be logged in
if(check_logged_in()){
switch($action){
case'account-settings':
display_account_form(get_email(),
get_real_name(get_email()),get_mimetype(get_email()));
break;
case'show-other-lists':
display_items('Unsubscribed Lists',
get_unsubscribed_lists(get_email()),'information',
'show-archive','subscribe');
break;
```

```
case'subscribe':
subscribe(get_email(),$_GET['id']);
display_items('Subscribed Lists',get_subscribed_lists(get_email()),
'information','show-archive','unsubscribe');
break;
case'unsubscribe':
unsubscribe(get_email(),$_GET['id']);
display_items('Subscribed Lists',get_subscribed_lists(get_email()),
'information','show-archive','unsubscribe');
break;
case'':
case'show-my-lists':
display_items('Subscribed Lists',get_subscribed_lists(get_email()),
'information','show-archive','unsubscribe');
break;
case'change-password':
display_password_form();
break;
case'store-change-password':
if(change_password(get_email(),$_POST['old_passwd'],
$_POST['new_passwd'],$_POST['new_passwd2'])){
echo"<p style=\"padding-bottom:50px\">OK:Password
changed.</p>";
}else{
echo"<p style=\"padding-bottom:50px\">Sorry,your
password could not be changed.</p>";
display_password_form();
}
break;
}
}
//The following actions may only be performed by an admin user
if(check_admin_user()){
switch($action){
case'create-mail':
display_mail_form(get_email());
```

```
break;
case'create-list':
display_list_form(get_email());
break;
case'store-list':
if(store_list($_SESSION['admin_user'],$_POST)){
echo"<p style=\"padding-bottom:50px\">New list added.</p>";
display_items('All Lists',get_all_lists(),'information',
'show-archive','');
}else{
echo"<p style=\"padding-bottom:50px\">List could not be
stored.Please try again.</p>";
}
break;
case'send':
send($_GET['id'],$_SESSION['admin_user']);
break;
case'view-mail':
display_items('Unsent Mail',get_unsent_mail(get_email()),
'preview-html','preview-text','send');
break;
}
}
/**********************************************************************
*Section 4:display footer
**********************************************************************/
do_html_footer();
?>
```

在以上程序清单中，可以看出代码4个部分清楚地标记出来了。在预处理阶段，我们将建立会话，并且完成在标题发送前必要的处理，在这个例子中，包括登录和登出。

在标题阶段，我们设置用户将会见到的菜单按钮，并且使用output_fns.php中的函数do_html_header()来显示相应的标题。该函数仅显示标题栏和菜单，我们将不详细介绍。

在脚本的主要部分，我们根据用户选择的操作做了相应的响应。这些操作分为3个子集：用户没有登录时可以进行的操作、普通用户可以进行的操作，以及管理员用户可以进行的操作。使用check_logged_in()和check_admin_user()函数来检查用户是否允许进行后两组操作。这3个函数都位于user_auth_fns.php函数库中。这些函数以及check_normal_user()函数的代码如程序清单30-3所示。

程序清单30-3　user_auth_fns.php中的函数——这些函数检查用户是否登录以及以什么级别登录

```
function check_normal_user(){
//see if somebody is logged in and notify them if not
if(isset($_SESSION['normal_user'])){
return true;
}else{
return false;
}
}
function check_admin_user(){
//see if somebody is logged in and notify them if not
if(isset($_SESSION['admin_user'])){
return true;
}else{
return false;
}
}
function check_logged_in(){
return(check_normal_user()||check_admin_user());
```

可以看到，这些函数使用会话变量normal_user和admin_user来检验用户是否登录。稍后，我们将讨论如何设置这些会话变量。

在该脚本的最后部分，我们使用output_fns.php中的函数do_html_footer()来发送HTML页脚。

我们简要地了解一下该系统中可能发生的动作，如表30-2所示。

表30-2 邮件列表管理器应用程序中可能的操作

动 作	执 行 者	描 述
log-in	任何人	为用户给出一个登录表单
log-out	任何人	结束一个会话
new-account	任何人	为用户创建一个新账户
store-account	任何人	保存账户详细信息
show-all-lists	任何人	显示可供使用的邮件列表的清单
show-archive	任何人	显示特定列表的存档新闻信件
information	任何人	显示特定列表的基本信息
account-settings	登录用户	显示用户账户设置
show-other-lists	登录用户	显示用户没有订阅的邮件列表
show-my-lists	登录用户	显示用户已经订阅的邮件列表
subscribe	登录用户	用户订阅特定列表
unsubscribe	登录用户	取消用户对特定列表的订阅
change-password	登录用户	显示修改密码表单
store-change-password	登录用户	在数据库中更新用户密码
create-mail	管理员	显示上载新闻信件表单
create-list	管理员	显示创建新邮件列表表单
store-list	管理员	在数据库中保存邮件列表的详细信息
view-mail	管理员	显示已经上载但仍未发送的新闻信件
send	管理员	向订阅者发送新闻信件

在这个表中，一个值得注意的省略是关于store-mail行的选项，即管理员通过create-mail动作上载输入新闻信件的操作。事实上，这个单独的功能是在另一个不同的文件upload.php中实现的。我们把它放在单独的一个文件中是为了减轻程序员对安全问题的注意。

在接下来的内容中，我们将讨论表30-2中给出的3组操作的实现，也就是，未登录用户的操作、登录用户的操作和管理员的操作。

30.5 实现登录

当一个新用户访问站点的时候，我们希望他完成3件事：首先，查看我们所提供的服务；其次，在网站进行注册；最后，登录。我们将依次讨论这3件事。

在图30-4中，可以看到用户每一次光临站点时我们所提供的页面。

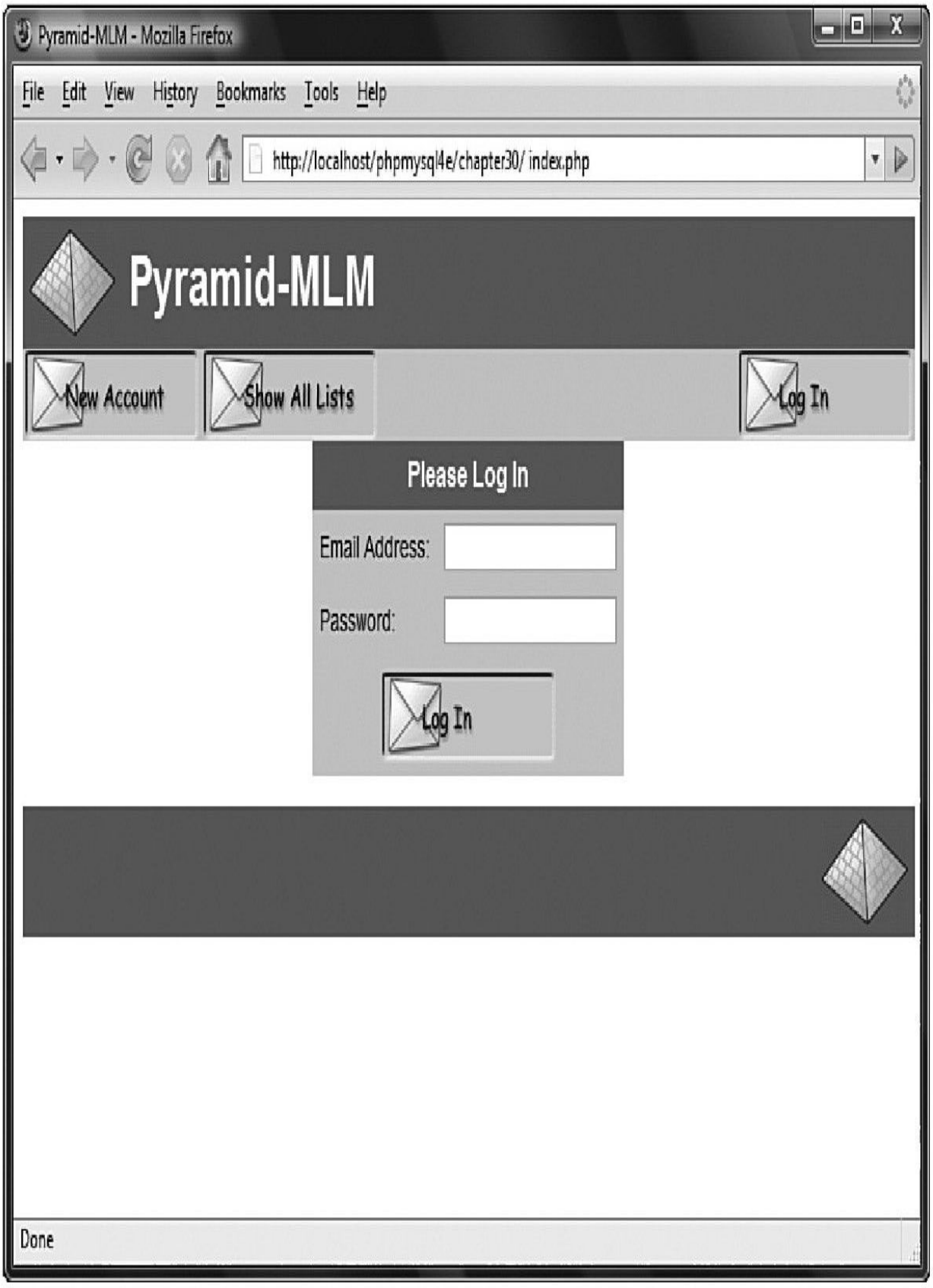

Pyramid-MLM - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/phpmysql4e/chapter30/ index.php
Pyramid-MLM
New Account
Show All Lists
Log In
Please Log In
Email Address:
Password:
Log In
Done

图　30-4　访问该站点后，用户可以创建新账户，查看可用的邮件列表或进行登录

现在，我们先来了解新账户的创建和用户登录。在本章稍后的“用户函数的实现”和“管理函数的实现”部分中，我们将回顾列表细节。

30.5.1　新账户的创建

如果用户选择"New Account"菜单选项，将触发new-account动作。这样就会触发index.php中的如下代码段：

```
case'new-account':
//get rid of session variables
session_destroy();
display_account_form();
break;
```

如果用户当前处于登录状态，这段代码将会登出用户，并将显示如图30-5所示的账户详细信息表单。


Pyramid-MLM - New Account - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/phpmysql4e/chapter30/index.php?action=new-account
Pyramid-MLM - New Account
New Account
Show All Lists
Log In
New Account
Real Name:
Email Address:
Requested Email Format:
Text Only
Password:
Save Changes
Done

图　30-5　新账户创建表单允许用户输入详细信息

该表单是由output_fns.php文件的display_account_form()函数生成的。这个函数除了在这里使用外，还在account-settings动作中使用，因为它可以用来显示一个创建新账户的表单。如果该函数由account-settings动作触发，表单会预先填好当前已有的用户数据。在这个例子中，表单是空的，准备接受新的账户信息。由于该函数只输出HTML，在这里，我们不做详细介绍。

以上表单中的提交按钮将触发store-account动作。该动作的代码如下所示：

```
case'store-account':
if(store_account($_SESSION['normal_user'],
$_SESSION['admin_user'],$_POST)){
$action='';
}
if(!check_logged_in()){
display_login_form($action);
}
break;
```

store_account()函数将用户的账户信息写入数据库中。程序清单30-4给出了该函数的代码。

程序清单30-4　mlm_fns.php脚本中的store_account()函数——这些函数可以添加一个新用户或者修改数据库中一个已有的用户

```
//add a new subscriber to the database,or let a user modify their data
function store_account($normal_user,$admin_user,$details){
if(!filled_out($details)){
echo"<p>All fields must be filled in.Try again.</p>";
return false;
}else{
if(subscriber_exists($details['email'])){
```

```
//check logged in as the user they are trying to change
if(get_email()==$details['email']){
$query="update subscribers set
realname='".$details[realname]."',
mimetype='".$details[mimetype]."'
where email='".$details[email]."'";
if($conn=db_connect()){
if($conn->query($query)){
return true;
}else{
return false;
}
}else{
echo"<p>Could not store changes.</p>";
return false;
}
}else{
echo"<p>Sorry,that email address is already registered here.</p>
<p>You will need to log in with that address to
change its settings.</p>";
return false;
}
}else{
//new account
$query="insert into subscribers
values('".$details[email]."',
'".$details[realname]."',
'".$details[mimetype]."',
sha1('".$details[new_password]."'),
0)";
if($conn=db_connect()){
if($conn->query($query)){
return true;
}else{
return false;
}
```

```
}else{
echo"<p>Could not store new account.</p>";
return false;
}
}
}
}
```

该函数首先检查用户是否填好了所需的资料。如果这些信息已经填好，函数接着将创建新用户，或者如果该用户已经存在，函数将更新该账户资料。用户只能更新当前登录用户的账户资料。这是通过调用get_email()函数来检查的，该函数将得到当前登录用户的邮件地址。在稍后的内容中，我们还将接触这个函数，因为它使用了用户登录时设置的会话变量。

30.5.2 登录

如果用户填写了前面图30-4中给出的登录表单，并且点击"Log In"（登录）按钮后，他将进入index.php脚本，同时email和password变量也将被传递到这个脚本。这将触发登录代码，登录代码是整个脚本的预处理阶段，如下所示：

```
//need to process log in or out requests before anything else
if(($_POST['email'])&&($_POST['password'])){
$login=login($_POST['email'],$_POST['password']);
if($login=='admin'){
$status.="<p style=\"padding-bottom:50px\">
<strong>".get_real_name($_POST['email'])."</strong>
logged in successfully as
<strong>Administrator</strong>.</p>";
$_SESSION['admin_user']=$_POST['email'];
}else if($login=='normal'){
$status.="<p style=\"padding-bottom:50px\">
<strong>".get_real_name($_POST['email'])."</strong>
logged in successfully.</p>";
$_SESSION['normal_user']=$_POST['email'];
}else{
$status.="<p style=\"padding-bottom:50px\">Sorry,we could
not log you in with that email address
and password.</p>";
}
}
if($action=='log-out'){
unset($action);
$_SESSION=array();
session_destroy();
}
```

可以看到，我们首先使用user_auth_fns.php库中的login()函数让用户进行登录。这个函数与我们在其他地方用到的登录函数有所不同，因此我们对其进行分析。该函数的代码如程序清单30-5所示。

程序清单30-5　user_auth_fns.php库中的login()函数——该函数检查用户登录资料

```
function login($email,$password){
//check username and password with db
//if yes,return login type
//else return false
//connect to db
$conn=db_connect();
if(!$conn){
return 0;
}
$query="select admin from subscribers
where email='".$email."'
and password=sha1('".$password."')";
$result=$conn->query($query);
if(!$result){
return false;
}
if($result->num_rows<1){
return false;
}
$row=$result->fetch_array();
if($row[0]==1){
return'admin';
}else{
return'normal';
}
}
```

在前面讨论过的登录函数中，如果登录成功，将返回true，否则将返回false。在这个例子中，登录失败仍然返回false，但是如果登录成功，将返回用户类型，用户类型可以是“管理员”或者“普通用户”。通过检索存储在subscribers表中的admin列，我们可以判断由邮件地址和密码组成一个特定组合的用户类型。如果没有返回结果，则返回false，如果用户是管理员，返回值为1（true），这样返回"admin"。否则，返回"normal"。

回到以上代码执行的主要部分，可以看到，我们注册了一个用于跟踪该用户身份的会话变量。如果该用户是管理员，该变量将是admin_user；如果是普通用户，该变量则是normal_user。无论我们设置的是哪一个变量，该变量都将包括用户的邮件地址。为了简化用户邮件地址的检查，我们使用前面介绍的get_email()函数。该函数代码如程序清单30-6所示。

程序清单30-6　user_auth_fns.php中的get_email()函数——返回登录用户的邮件地址

```
function get_email(){
if(isset($_SESSION['normal_user'])){
return$_SESSION['normal_user'];
}
if(isset($_SESSION['admin_user'])){
return$_SESSION['admin_user'];
}
return false;
}
```

回到主程序中，我们将报告用户是否已经登录，以及以什么级别登录。图30-6所示的是一个登录操作的输出。

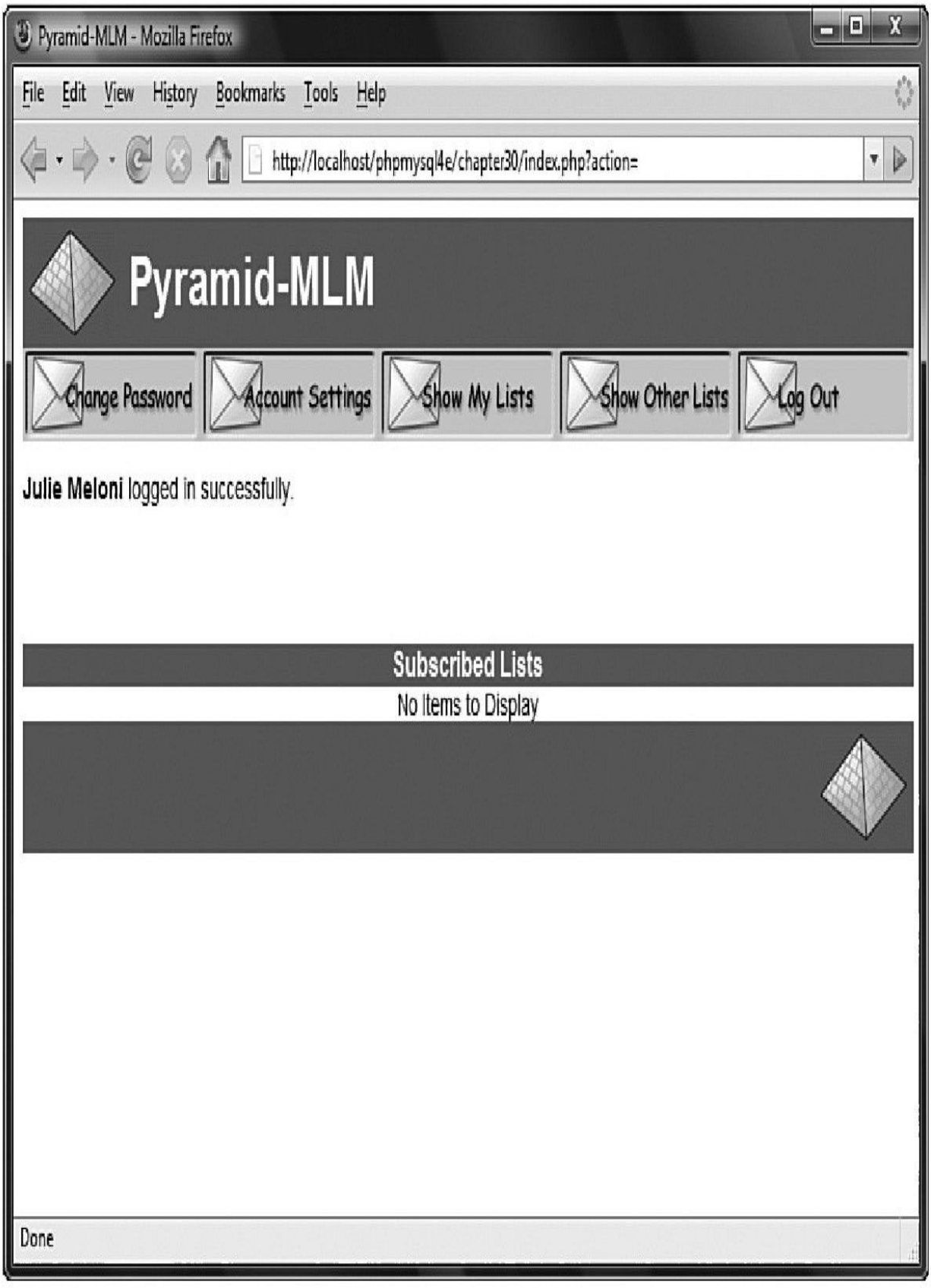

Pyramid-MLM - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/phpmysql4e/chapter30/index.php?action=
Pyramid-MLM
Change Password
Account Settings
Show My Lists
Show Other Lists
Log Out
Julie Meloni logged in successfully.
Subscribed Lists
No Items to Display
Done

图　30-6　系统报告用户登录成功

现在我们已经完成了一个用户的登录，下面讨论用户函数。

30.6 用户函数的实现

用户登录后，我们希望他们能够完成如下所示的5件事：

■查看可供订阅的邮件列表

■订阅或取消订阅邮件列表

■修改建立账户的方法

■更改密码

■登出

在图30-6中，可以看到这些选项的大部分。接下来，我们介绍每个选项的具体实现。

30.6.1 查看列表

在这个项目中，我们将实现很多选项，这些选项可以用来查看可供使用的列表及列表细节。在图30-6中，可以看到其中两个选项："Show My Lists"，该选项用来获得该用户订阅的邮件列表；"Show Other Lists"，来获得该用户没有订阅的邮件列表。

如果回头看看图30-4，会发现还有另一个选项——"Show All Lists"，该选项可以获得系统中所有可供使用的邮件列表。为了使系统真正可以升级，我们应增加分页功能（例如，每页显示10个查询结果）。出于简单的考虑，我们没有使用这项功能。

这3个选项将分别触发show-all-lists、show-other-lists和show-my-lists动作。我们可能已经意识到，所有这些动作的工作方式十分相似。以上3种动作的代码如下所示：

```
case'show-all-lists':
display_items('All Lists',get_all_lists(),'information',
'show-archive','');
```

```
break;
case'show-other-lists':
display_items('Unsubscribed Lists',
get_unsubscribed_lists(get_email()),'information',
'show-archive','subscribe');
break;
case'':
case'show-my-lists':
display_items('Subscribed Lists',get_subscribed_lists(get_email()),
'information','show-archive','unsubscribe');
break;
```

可以看到，所有这些操作都将调用output_fns.php库中的display_items()函数，但是每一个操作对该函数的调用都有不同的参数。它们也将使用前面提到的get_email()函数来获得用户的电子邮件地址。

图30-7所示的是这个函数的功能。


Pyramid-MLM - Show Other Lists - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/phpmysql4e/chapter30/index.php?action=show-other-lists
Pyramid-MLM - Show Other Lists
Change Password
Account Settings
Show My Lists
Show Other Lists
Log Out
Unsubscribed Lists
PHP Tipsheet
Information
Show Archive
Subscribe
Pyramid Newsletter
Information
Show Archive
Subscribe
Done

图　30-7　display_items()函数用来列出用户未订阅的列表

以上就是"Show Other Lists"页面。

下面，让我们来看看display_items()函数的代码，如程序清单30-7所示。

程序清单30-7　output_fns.php库中的display_items()函数——该函数用来显示相关操作的条目列表

```
function display_items($title,$list,$action1='',$action2='',
$action3=''){
global$table_width;
echo"<table width=\"$table_width\"cellspacing=\"0\"
cellpadding=\"0\"border=\"0\">";
//count number of actions
$actions=(($action1!='')+($action2!='')+($action3!=''));
echo"<tr>
<th colspan=\"".(1+$actions)."\"bgcolor=\"#5B69A6\">"
.$title."</th>
</tr>";
//count number of items
$items=sizeof($list);
if($items==0){
echo"<tr>
<td colspan=\"".(1+$actions)."\"align=\"center\">No
Items to Display</td>
</tr>";
}else{
//print each row
for($i=0;$i<$items;$i++){
if($i%2){
//background colors alternate
$bgcolor="#ffffff";
}else{
$bgcolor="#ccccff";
```

```
}
echo"<tr>
<td bgcolor=\"".$bgcolor."\"
width=\"".($table_width-($actions*149))."\">";
echo$list[$i][1];
if($list[$i][2]){
echo"-".$list[$i][2];
}
echo"</td>";
//create buttons for up to three actions per line
for($j=1;$j<=3;$j++){
$var="action".$j;
if($$var){
echo"<td bgcolor=\"".$bgcolor."\"width=\"149\">";
//view/preview buttons are a special case as they link to a file
if(($$var=='preview-html')||($$var=='view-html')||
($$var=='preview-text')||($$var=='view-text')){
display_preview_button($list[$i][3],$list[$i][0],$$var);
}else{
display_button($$var,'&id='.$list[$i][0]);
}
echo"</td>";
}
}
echo"</tr>\n";
}
echo"</table>";
}
}
```

该函数将输出一个条目表，每个条目都具有3个相关的操作按钮。该函数具有5个参数，它们依次如下所示。

■$title是显示在表顶部的标题——在图30-7中，我们提交的标题为"Unsubscribed Lists"，正如我们前面讨论过的"Show Other Lists"的那段代码中所显示的。

■$list是表中每行显示的条目组成的数组。在这个例子中，它是该用户没有订阅的邮件列表组成的数组。我们将在get_unsubscribed_lists()函数中构建该数组（在这个例子中），我们将在稍后的内容中介绍这些函数。这是一个多维数组，该数组中的每一行包括4块数据。依次如下。

■$list[n][0]包含条目标识符，通常是行号。该序号为按钮给出了将要操作的行号。在我们的例子中，使用数据库中的ID——稍后将详细介绍。

■$list[n][1]包含该条目名称。为特定条目显示的文本。例如，在图30-7中，表中第一行的条目名是PHP Tipsheet。

■$list[n][2]和$list[n][3]是可选的。使用它们来表示还有其他信息。它们分别对应更多的信息文本和更多的信息ID。我们在讨论“管理函数的实现”部分的"View Mail"动作时了解使用这两个参数的例子。

■该函数的第3、第4、第5个参数用来传递3个动作，这些动作将会显示在相应条目的按钮上。在图30-7中，3个动作按钮分别是"Information"、"Show Archive"和"Subscribe"。

通过传递动作名称，即information、show-archive和subscribe，可以获得"Show All Lists"页面上的这3个按钮。通过display_button()函数，这些动作将被转换成带有文字的按钮，并且赋予它们相应的动作。

可以看到，在这些动作中，每一个Show动作都将以不同的方法调用display_items()函数。除了具有不同的标题和动作按钮外，这3个动作还会使用不同的函数建立要显示的条目数组。"Show All Lists"使用get_all_lists()函数。"Show Other Lists"使用get_unsubscribed_lists()函数。而"Show My Lists"将使用get_subscribed_lists()。所有这些函数的工作方式类似。这些函数都来自mlm_fns.php函数库。

下面，我们将了解get_unsubscribed_lists()函数，因为它是我们已经使用的例子。该函数的代码如程序清单30-8所示。

程序清单30-8　mlm_fns.php库中的get_unsubscribed_lists()函数——该函数用来建立用户没有订阅的邮件列表数组

```
function get_unsubscribed_lists($email){
$list=array();
$query="select lists.listid,listname,email from lists
left join sub_lists on lists.listid=sub_lists.listid
and email='".$email."'where email is NULL
order by listname";
if($conn=db_connect()){
$result=$conn->query($query);
if(!$result){
echo'<p>Unable to get list from database.</p>';
return false;
}
$num=$result->num_rows;
for($i=0;$i<$num;$i++){
$row=$result->fetch_array();
array_push($list,array($row[0],$row[1]));
}
}
return$list;
}
```

可以看到，该函数要求传递一个邮件地址作为参数。这应该是我们正与之交互的订阅者的邮件地址。get_subscribed_lists()函数也要一个邮件地址作为参数，显然，get_all_lists()不需要任何参数。

给出订阅者邮件地址后，我们可以连接数据库并从中取出该订阅者没有订阅的所有邮件列表。可以使用LEFT JOIN来查找不匹配条目。通过遍历这个结果集，我们可以使用array_push()内置函数来逐行建立数组。

现在我们已经了解了列表的产生过程，下面，我们将介绍与这些显示相关的动作按钮。

30.6.2　查看邮件列表信息

在图30-7中，"Information"按钮将触发information动作，如下所示：

```
case'information':
display_information($_GET['id']);
break;
```

图30-8所示的是display_information()函数的输出结果。

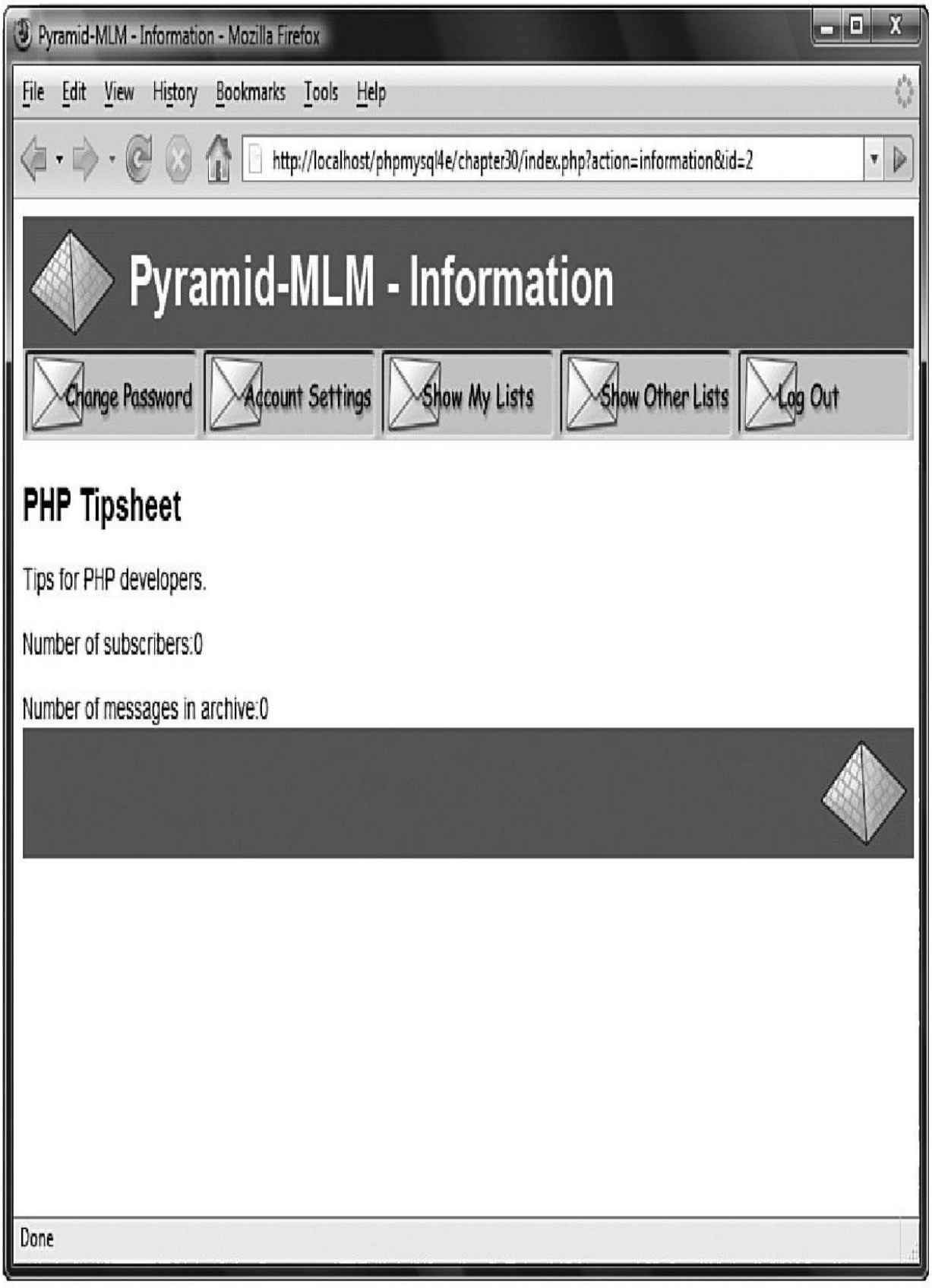

Pyramid-MLM - Information - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/phpmysql4e/chapter30/index.php?action=information&id=2
Pyramid-MLM - Information
Change Password
Account Settings
Show My Lists
Show Other Lists
Log Out
PHP Tipsheet
Tips for PHP developers.
Number of subscribers:0
Number of messages in archive:0
Done

图　30-8　display_information()函数显示邮件列表的介绍

该函数将显示特定邮件列表的常规信息，此外，还列出了该列表的订阅人数和将要发送给该列表上用户，以及在存档文件（稍后将进一步讨论）中可用的新闻信件数目。该函数的代码如程序清单30-9所示。

程序清单30-9　output_fns.php库中的display_information()函数——显示邮件列表信息的函数

```
//diplay stored information about each list
function display_information($listid){
if(!$listid){
return false;}
$info=load_list_info($listid);
if($info){
echo"<h2>".pretty($info[listname])."</h2>
<p>".pretty($info[blurb])."
</p><p>Number of subscribers:".$info[subscribers]."
</p><p>Number of messages in archive:"
.$info[archive]."</p>";
}
}
```

display_information()函数调用其他两个函数来协助完成存档Web任务：load_list_info()函数和pretty()函数。load_list_info()函数实际上是从数据库中获取数据。而pretty()函数只是通过去除数据中的斜杠，并且执行一些替换操作来格式化来自数据库中的数据，例如，将换行转变成HTML的行标记等。

下面，我们简单介绍load_list_info()函数。该函数保存在mlm_fns.php函数库中。代码如程序清单30-10所示。

程序清单30-10　mlm_fns.php库中的load_list_info()函数——该函数建立邮件列表信息的数组

```
function load_list_info($listid){
if(!$listid){
return false;
}
if(!($conn=db_connect())){
return false;
}
$query="select listname,blurb from lists where listid=
'".$listid."'";
$result=$conn->query($query);
if(!$result){
echo"<p>Cannot retrieve this list.</p>";
return false;
}
$info=$result->fetch_assoc();
$query="select count(*)from sub_lists where listid=
'".$listid."'";
$result=$conn->query($query);
if($result){
$row=$result->fetch_array();
$info['subscribers']=$row[0];
}
$query="select count(*)from mail where listid='".$listid."'
and status='SENT'";
$result=$conn->query($query);
if($result){
$row=$result->fetch_array();
$info['archive']=$row[0];
}
return$info;
}
```

该函数将运行3个数据库查询语句，分别用来收集lists表中邮件列表的名称和简介，以及sub_lists表中订阅者的数目和由mail表发送的邮件数目。

30.6.3　查看邮件列表存档

除了可以查看邮件列表简介外，用户还可以通过点击"Show Archive"按钮查看发送到邮件列表上的所有邮件。这样将触发show-archive动作，其代码如下所示：

```
case'show-archive':
display_items('Archive For'.get_list_name($_GET['id']),
get_archive($_GET['id']),'view-html',
'view-text','');
break;
```

需要再次提到的是，该函数也用到了display_items()函数，它使用这个函数列出已经发送到列表上的各种条目。这些条目可以通过调用mlm_fns.php中的get_archive()函数来获得。该函数如程序清单30-11所示。

程序清单30-11　mlm_fns.php库中的get_archive()函数——该函数可以给出特定列表中的存档新闻信件数组

```
function get_archive($listid){
//returns an array of the archived mail for this list
//array has rows like(mailid,subject)
$list=array();
$listname=get_list_name($listid);
$query="select mailid,subject,listid from mail
where listid='".$listid."'and status='SENT'
order by sent";
if($conn=db_connect()){
$result=$conn->query($query);
if(!$result){
echo"<p>Unable to get list from database.</p>";
return false;
}
```

```
$num=$result->num_rows;
for($i=0;$i<$num;$i++){
$row=$result->fetch_array();
$arr_row=array($row[0],$row[1],
$listname,$listid);
array_push($list,$arr_row);
}
}
return$list;
}
```

该函数将从数据库中获得所需的信息——在这个例子中，所需的信息是已经发送出去的邮件详细信息——并且构建一个适合传递给display_items()函数的数组。

30.6.4　订阅与取消订阅

在图30-7所示的邮件列表中，每个邮件列表都有一个按钮可供用户订阅。与此类似，如果用户选择"Show My Lists"选项查看已经订阅的邮件列表，他们将看到在每个邮件列表后面都有一个"Unsubscribe"按钮。

这些按钮可以触发订阅或取消订阅的动作，这两个动作分别触发下面两段代码：

```
case'subscribe':
subscribe(get_email(),$_GET['id']);
display_items('Subscribed Lists',
get_subscribed_lists(get_email()),
'information','show-archive','unsubscribe');
break;
case'unsubscribe':
unsubscribe(get_email(),$_GET['id']);
display_items('Subscribed Lists',
get_subscribed_lists(get_email()),
'information','show-archive','unsubscribe');
break;
```

在每一种情况中，我们都将调用一个函数（subscribe()或unsubscribe()）并调用display_items()函数重新显示该用户订阅的邮件列表。

subscribe()和unsubscribe()函数如程序清单30-12所示。

程序清单30-12　mlm_fns.php函数库中的subscribe()和unsubscribe()函数——这些函数可以为一个用户增加或删除邮件列表的订阅

```
//subscribe this email address to this list
function subscribe($email,$listid){
```

```
if((!$email)||(!$listid)||(!list_exists($listid))
||(!subscriber_exists($email))){
return false;
}
//if already subscribed exit
if(subscribed($email,$listid)){
return false;
}
if(!($conn=db_connect())){
return false;
}
$query="insert into sub_lists values('".$email."',$listid)";
$result=$conn->query($query);
return$result;
}
//unsubscribe this email address from this list
function unsubscribe($email,$listid){
if((!$email)||(!$listid)){
return false;
}
if(!($conn=db_connect())){
return false;
}
$query="delete from sub_lists where email='".$email."'and
listid='".$listid."'";
$result=$conn->query($query);
return$result;
}
```

---

subscribe()函数将在对应的sub_lists表中增加一行；而unsubscribe()函数则删除一行。

30.6.5 更改账户设置

点击"Account Settings"按钮时，将触发account-settings动作。该动作的代码如下所示：

```
case'account-settings':
display_account_form(get_email(),
get_real_name(get_email()),get_mimetype(get_email()));
break;
```

可以看到，我们再次使用了最初用来创建账户的函数display_account_form()。但是这次，我们向这个函数传递了用户当前的资料，这些资料会显示在表单中以便用户编辑。用户点击该表单中的提交按钮时，就会像前面讨论过的一样触发store-account动作。

30.6.6　更改密码

点击"Change Password"按钮将触发change-password动作，该动作触发如下所示的代码：

```
case'change-password':
display_password_form();
break;
```

display_password_form()函数（来自output_fns.php函数库）只是显示一个可供用户修改密码的表单。该表单如图30-9所示。

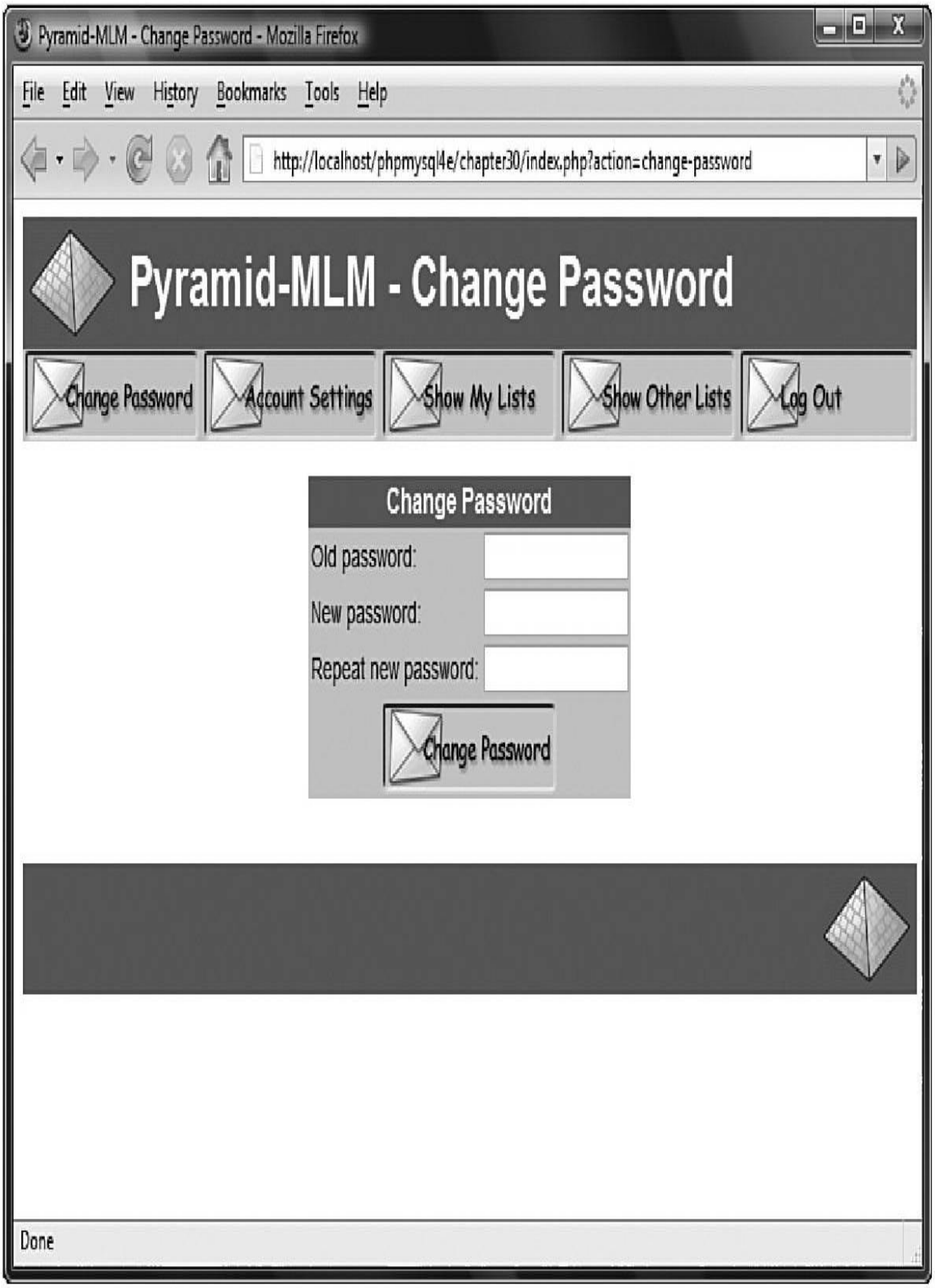

Pyramid-MLM - Change Password - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/phpmysql4e/chapter30/index.php?action=change-password
Pyramid-MLM - Change Password
Change Password
Account Settings
Show My Lists
Show Other Lists
Log Out
Change Password
Old password:
New password:
Repeat new password:
Change Password
Done

图　30-9　display_password_form()函数允许用户更改密码

当用户点击表单底部的"Change Password"按钮时，将触发store-change-password动作。该动作代码如下所示：

```
case'store-change-password':
if(change_password(get_email(),$_POST['old_passwd'],
$_POST['new_passwd'],$_POST['new_passwd2'])){
echo"<p style=\"padding-bottom:50px\">OK:Password
changed.</p>";
}else{
echo"<p style=\"padding-bottom:50px\">Sorry,your
password could not be changed.</p>";
display_password_form();
}
break;
```

可以看到，以上代码将使用change_password()函数来更改密码并向用户报告密码更改是否成功。change_password()函数可以在user_auth_fns.php函数库中找到。该函数的代码如程序清单30-13所示。

程序清单30-13　user_auth_fns.php函数库中的change_password()函数——该函数可以校验并更新用户密码

```
function change_password($email,$old_password,$new_password,
$new_password_conf){
//change password for email/old_password to new_password
//return true or false
//if the old password is right
//change their password to new_password and return true
//else return false
if(login($email,$old_password)){
if($new_password==$new_password_conf){
if(!($conn=db_connect())){
return false;
```

```
}
$query="update subscribers
set password=sha1('".$new_password."')
where email='".$email."'";
$result=$conn->query($query);
return$result;
}else{
echo"<p>Your passwords do not match.</p>";
}
}else{
echo"<p>Your old password is incorrect.</p>";
}
return false;//old password was wrong
}
```

该函数与我们前面所看到的密码设置和修改函数类似。它比较用户输入的两个密码，判断是否一致，如果一致，则在数据库中更改该用户的密码。

30.6.7 登出

当用户点击"Log Out"按钮时，将触发log-out动作。在主脚本中，该动作所执行的代码实际上主要在预处理阶段，如下所示：

```
if($action=='log-out'){
unset($action);
$_SESSION=array();
session_destroy();
}
```

以上代码段将清除会话变量并结束该会话。请注意，它还将清除action变量——这意味着进入主case语句而不发生任何动作，这将触发下面的代码：

```
default:
if(!check_logged_in()){
display_login_form($action);
}
break;
```

这将使另一个用户登录，或者当前用户以其他身份登录。

30.7　管理功能的实现

如果某人以管理员身份登录，将看到一些额外的菜单选项，如图30-10所示。

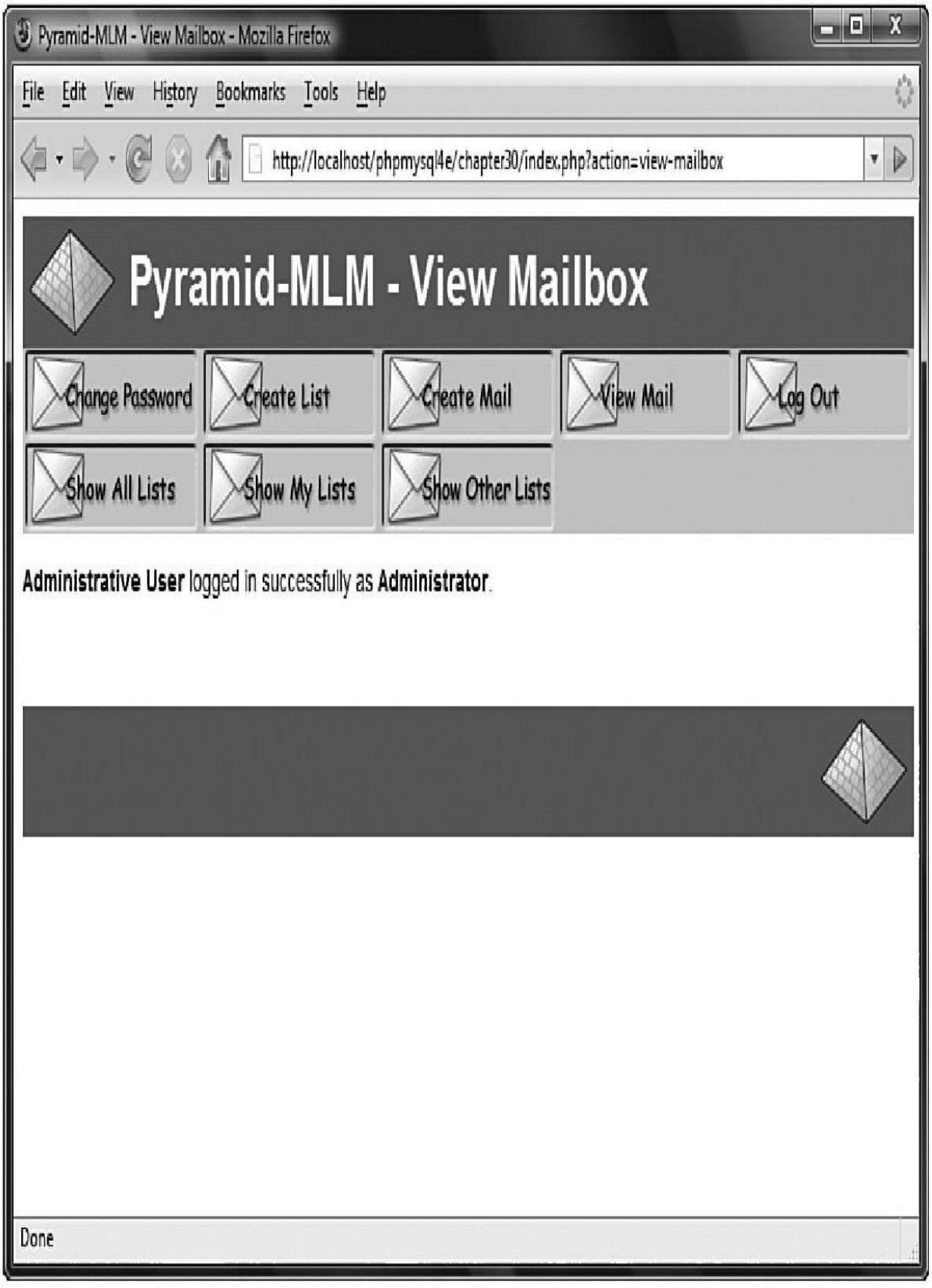

Pyramid-MLM - View Mailbox - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/phpmysql4e/chapter30/index.php?action=view-mailbox
Pyramid-MLM - View Mailbox
Change Password
Create List
Create Mail
View Mail
Log Out
Show All Lists
Show My Lists
Show Other Lists
Administrative User logged in successfully as Administrator.
Done

图　30-10　管理员菜单选项允许创建和维护邮件列表

管理员身份独有的选项是"Create List"（创建新的邮件列表）、"Create Mail"（创建新的新闻信件）和"View Mail"（查看并发送那些已经创建但还没发出去的新闻信件）。下面，我们将依次讨论它们。

30.7.1　创建新的邮件列表

如果管理员点击"Create List"按钮来建立新的邮件列表，将触发create-list动作，该动作与如下所示代码相关：

```
case'create-list':
display_list_form(get_email());
break;
```

display_list_form()函数将显示一个可供管理员输入新邮件列表详细信息的表单。该函数保存在output_fns.php函数库中。它只是用来输出HTML，因此在这里，我们不深入讨论它。该函数的输出如图30-11所示。

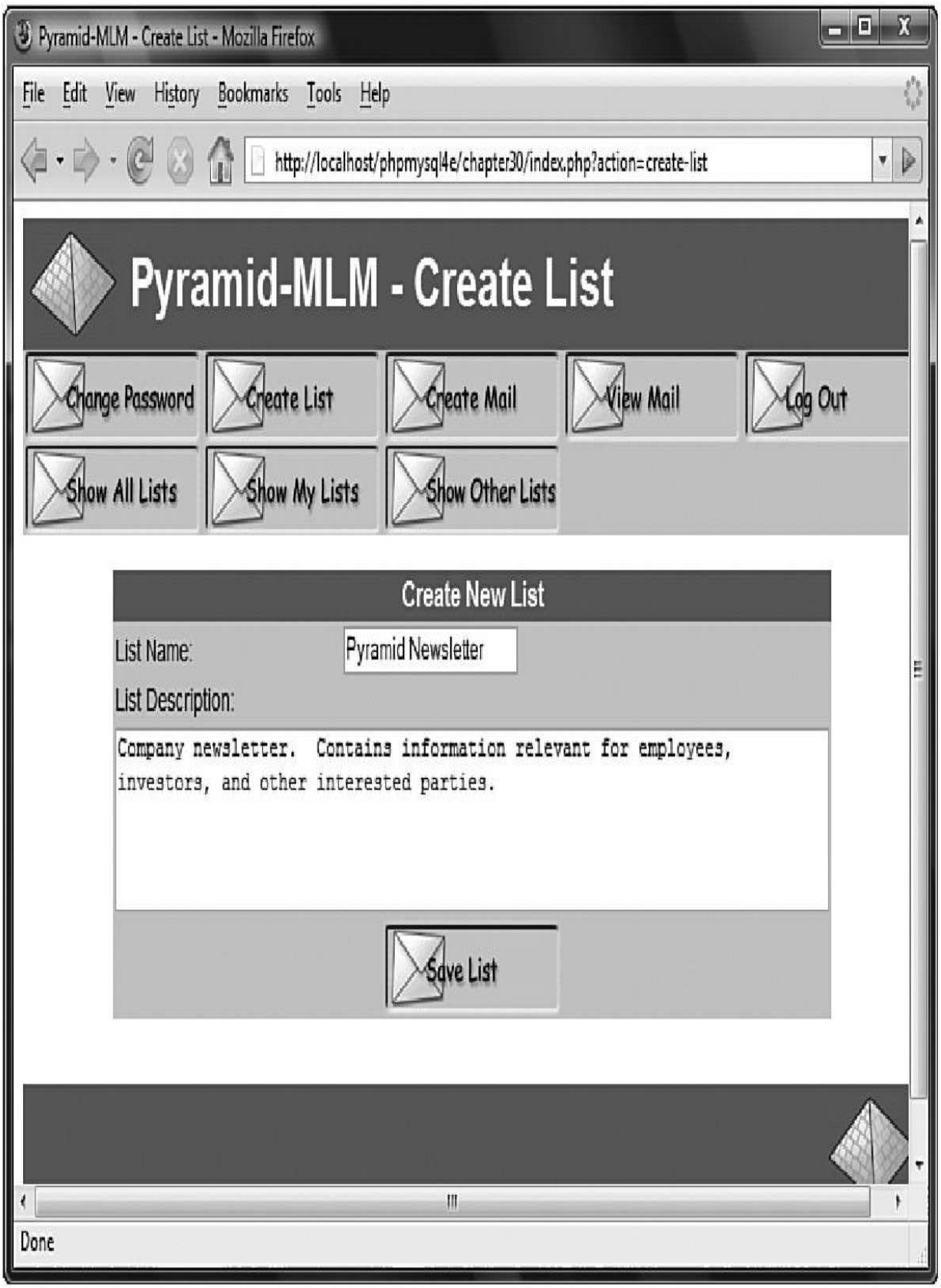

Pyramid-MLM - Create List - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/phpmysql4e/chapter30/index.php?action=create-list
Pyramid-MLM - Create List
Change Password
Create List
Create Mail
View Mail
Log Out
Show All Lists
Show My Lists
Show Other Lists
Create New List
List Name:
Pyramid Newsletter
List Description:
Company newsletter. Contains information relevant for employees, investors, and other interested parties.
Save List
Done

图　30-11　"Create List"选项要求管理员为新的邮件列表输入名称和描述（或简介）

当管理员点击"Save List"按钮时，将触发store-list动作，而该动作将触发index.php中的如下所示代码：

```
case'store-list':
if(store_list($_SESSION['admin_user'],$_POST)){
echo"<p style=\"padding-bottom:50px\">New list added.</p>";
display_items('All Lists',get_all_lists(),'information',
'show-archive','');
}else{
echo"<p style=\"padding-bottom:50px\">List could not be
stored.Please try again.</p>";
}
break
```

可以看到，以上代码将保存新的列表详细信息，并显示新的列表。列表详细信息是通过store_list()函数来保存的。该函数的代码如程序清单30-14所示。

程序清单30-14　mlm_fns.php函数库的store_list()函数——该函数在数据库中插入一个新的邮件列表

```
function store_list($admin_user,$details){
if(!filled_out($details)){
echo"<p>All fields must be filled in.Try again.</p>";
return false;
}else{
if(!check_admin_user($admin_user)){
return false;
//how did this function get called by somebody not logged in as admin?
}
if(!($conn=db_connect())){
return false;
```

```
}
$query="select count(*)from lists where listname='".$details['name']."'";
$result=$conn->query($query);
$row=$result->fetch_array();
if($row[0]>0){
echo"<p>Sorry,there is already a list with this name.</p>";
return false;
}
$query="insert into lists values(NULL,
'".$details['name']."',
'".$details['blurb']."')";
$result=$conn->query($query);
return$result;
}
}
```

该函数在写入数据库之前进行一些有效性检查：它将检查是否已经提供了所有需要的详细信息、当前的用户是否是管理员、以及该列表的名称是否是唯一的。如果所有检查都正确，该列表将被添加到数据库中的lists表。

30.7.2　上载新的新闻信件

最后，我们再来看看该应用程序的关键部分：上载并发送新闻信件到邮件列表。

当管理员点击"Create Mail"按钮时，将触发create-mail动作，如下所示：

```
case'create-mail':
display_mail_form(get_email());
break;
```

管理员将看到如图30-12所示的表单。


Pyramid-MLM - Create Mail - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/phpmysql4e/chapter30/index.php?action=create-mail
Pyramid-MLM - Create Mail
Change Password
Create List
Create Mail
View Mail
Log Out
Show All Lists
Show My Lists
Show Other Lists
List:
Pyramid Newsletter
Subject:
Text Version:
Browse...
HTML Version:
Browse...
Images: (optional)
Image 1
Browse...
Image 2
Browse...
Image 3
Browse...
Image 4
Browse...
Image 5
Browse...
Image 6
Browse...
Image 7
Browse...
Image 8
Browse...
Image 9
Browse...
Image 10
Browse...
Done

图　30-12　"Create Mail"选项提供了可供管理员上载新闻信件文件的界面

请记住，对于该程序，我们假设管理员已经在脱机环境中创建了HTML格式和文本格式的新闻信件，而且在发送该新闻信件之前，需要上载这两个版本的文件。我们选择用这种实现方法是为了让管理员可以使用他们最喜欢的软件创建新闻信件。这使得该程序更容易使用。

该表单有几处地方需要由管理员填写。在表单的最上方是一个下拉列表框，可供管理员选择邮件列表。管理员还必须填写新闻信件的主题——这就是最终邮件的主题行。

所有其他的表单域都是文件上载域，可以通过它们右边的"Browse"按钮分辨出来。要发送新闻信件，管理员必须给出该邮件的文本和HTML两种版本（可以根据需要进行修改）。此外，表单中还有很多可选的图像域，在这里，管理员可以上载任何嵌入到HTML的图像。这些文件必须明确指定并分别上载。

该表单与普通的上载表单相似，只是在这个例子中，我们使用它来上载多个文件。这就必然使得在表单语法与我们在另一端处理上载文件的方法上会存在一些差异。

display_mail_form()函数的代码如程序清单30-15所示。

在这里，需要注意的是我们要在一系列文本输入框中输入上载文件的名称，以及相应的文件类型。这些文件名称在程序中的命名为userfile[0]到userfile[n]。事实上，我们使用与处理复选框一样的方法来处理这些表单域，并使用数组命名惯例对其命名。

程序清单30-15　output_fns.php函数库中的display_mail_form()函数——该函数将显示文件上载表单

```
function display_mail_form($email,$listid=0){
//display html form for uploading a new message
global$table_width;
$list=get_all_lists();
$lists=sizeof($list);
```

```
?>
<table cellpadding="4"cellspacing="0"border="0"
width="<?php echo$table_width;?>">
<form enctype="multipart/form-data"action="upload.php"method="post">
<tr>
<td bgcolor="#cccccc">List:</td>
<td bgcolor="#cccccc">
<select name="list">
<?php
for($i=0;$i<$lists;$i++){
echo"<option value=\"".$list[$i][0]."\"";
if($listid==$list[$i][0]){
echo"selected";
}
echo">".$list[$i][1]."</option>\n";
}
?>
</select>
</td>
</tr>
<tr>
<td bgcolor="#cccccc">Subject:</td>
<td bgcolor="#cccccc">
<input type="text"name="subject"
value="<?php echo$subject;?>"
size="60"/></td>
</tr>
<tr>
<td bgcolor="#cccccc">Text Version:</td>
<td bgcolor="#cccccc">
<input type="file"name="userfile[0]"size="60"/></td>
</tr>
<tr><td bgcolor="#cccccc">HTML Version:</td>
<td bgcolor="#cccccc">
<input type="file"name="userfile[1]"size="60"/></td>
</tr>
```

```
<tr><td bgcolor="#cccccc"colspan="2">Images:(optional)
<?php
$max_images=10;
for($i=0;$i<10;$i++){
echo"<tr><td bgcolor=\"#cccccc\">Image".($i+1)."</td>
<td bgcolor=\"#cccccc\"><input type=\"file\"
name=\"userfile[".($i+2)."]\"size=\"60\"/></td>
</tr>";
}
?>
<tr><td colspan="2"bgcolor="#cccccc"align="center">
<input type="hidden"name="max_images"
value="<?php echo$max_images;?>">
<input type="hidden"name="listid"
value="<?php echo$listid;?>">
<?php display_form_button('upload-files');?>
</td>
</form>
</tr>
</table>
<?php
}
```

如果想通过PHP脚本上载多个文件并且以数组方式轻松地处理它们，必须遵循这个命名惯例。

在处理该表单的脚本中，我们实际上以3个数组结束。下面，让我们来看一下这个脚本。

30.7.3　多文件上载的处理

文件上载代码是一个单独的文件upload.php，该文件的完整代码如程序清单30-16所示。

程序清单30-16　upload.php——上载新闻信件所需的所有文件脚本

```
<?php
//this functionality is in a separate file to allow us to be
//more paranoid with it
//if anything goes wrong,we will exit
$max_size=50000;
include('include_fns.php');
session_start();
//only admin users can upload files
if(!check_admin_user()){
echo"<p>You do not seem to be authorized to use this page.</p>";
exit;
}
//set up the admin toolbar buttons
$buttons=array();
$buttons[0]='change-password';
$buttons[1]='create-list';
$buttons[2]='create-mail';
$buttons[3]='view-mail';
$buttons[4]='log-out';
$buttons[5]='show-all-lists';
$buttons[6]='show-my-lists';
$buttons[7]='show-other-lists';
do_html_header('Pyramid-MLM-Upload Files');
display_toolbar($buttons);
//check that the page is being called with the required data
if((!$_FILES['userfile']['name'][0])||
(!$_FILES['userfile']['name'][1])||
(!$_POST['subject']||!$_POST['list'])){
```

```
echo"<p>Problem:You did not fill out the form fully.
The images are the only optional fields.
Each message needs a subject,text version
and an HTML version.</p>";
do_html_footer();
exit;
}
$list=$_POST['list'];
$subject=$_POST['subject'];
if(!($conn=db_connect())){
echo"<p>Could not connect to db.</p>";
do_html_footer();
exit;
}
//add mail details to the DB
$query="insert into mail values(NULL,
'".$_SESSION['admin_user']."',
'".$subject."',
'".$list."',
'STORED',NULL,NULL)";
$result=$conn->query($query);
if(!$result){
do_html_footer();
exit;
}
//get the id MySQL assigned to this mail
$mailid=$conn->insert_id;
if(!$mailid){
do_html_footer();
exit;
}
//creating directory will fail if this is not the first message archived
//that's ok
@mkdir('archive/'.$list,0700);
//it is a problem if creating the specific directory for this mail fails
if(!mkdir('archive/'.$list.'/'.$mailid,0700)){
```

```
do_html_footer();
exit;
}
//iterate through the array of uploaded files
$i=0;
while(($_FILES['userfile']['name'][$i])&&
($_FILES['userfile']['name'][$i]!='none')){
echo"<p>Uploading".$_FILES['userfile']['name'][$i]."-".
$_FILES['userfile']['size'][$i]."bytes.</p>";
if($_FILES['userfile']['size'][$i]==0){
echo"<p>Problem:".$_FILES['userfile']['name'][$i].
"is zero length";
$i++;
continue;
}
if($_FILES['userfile']['size'][$i]>$max_size){
echo"<p>Problem:".$_FILES['userfile']['name'][$i]."is over"
.$max_size."bytes";
$i++;
continue;
}
//we would like to check that the uploaded image is an image
//if getimagesize()can work out its size,it probably is.
if(($i>1)&&(!getimagesize($_FILES['userfile']['tmp_name'][$i]))){
echo"<p>Problem:".$_FILES['userfile']['name'][$i].
"is corrupt,or not a gif,jpeg or png.</p>";
$i++;
continue;
}
//file 0(the text message)and file 1(the html message)are special cases
if($i==0){
$destination="archive/".$list."/".$mailid."/text.txt";
}else if($i==1){
$destination="archive/".$list."/".$mailid."/index.html";
}else{
$destination="archive/".$list."/".$mailid."/"
```

```
.$_FILES['userfile']['name'][$i];
$query="insert into images values('".$mailid."',
'".$_FILES['userfile']['name'][$i]."',
'".$_FILES['userfile']['type'][$i]."')";
$result=$conn->query($query);
}
if(!is_uploaded_file($_FILES['userfile']['tmp_name'][$i])){
//possible file upload attack detected
echo"<p>Something funny happening with"
.$_FILES['userfile']['name'].",not uploading.";
do_html_footer();
exit;
}
move_uploaded_file($_FILES['userfile']['tmp_name'][$i],
$destination);
$i++;
}
display_preview_button($list,$mailid,'preview-html');
display_preview_button($list,$mailid,'preview-text');
display_button('send',"&id=$mailid");
echo"<p style=\"padding-bottom:50px\"> </p>";
do_html_footer();
?>
```

下面，让我们来分析一下上载文件的步骤。首先，开始一个会话并检验用户是否以管理员身份登录——我们不希望其他的用户能够上载文件。

严格地讲，我们可能还需要对list和mailid变量进行检验，检查这些变量是否包含不希望的字符，在这里，我们只是从简化代码的角度考虑，忽略了该功能。

接下来，我们建立并发送页面的标题，并检验该表单已经正确填写。这非常重要，因为对用户来说，它是一个相当复杂的表单。

接着，需要在数据库中为该邮件创建一条记录，并且在存档目录中创建一个用来保存邮件的目录。

接下来就是脚本程序的主要部分，该部分将检查并转移上载的文件。在上载多个文件时，该部分还有一点不同。这里我们需要处理4个数组。这些数组分别是$_FILES['userfile']['name']、$_FILES['userfile']['tmp_name']、$_FILES['userfile']['size']和$_FILES['userfile']['type']。它们对应单文件上载时与之命名相似的变量，只是在这里它们都是数组。表单中第一个文件的详细内容在$_FILES['userfile']['tmp_name'][0]、$_FILES['userfile']['name'][0]、$_FILES['userfile']['size'][0]和$_FILES['userfile']['type'][0]中。

针对这3个数组，我们将进行常规的安全检查并将文件转移到存档中。

最后，我们为管理员提供几个按钮，在发送新闻信件前，这些按钮可以用来预览已经上传的新闻信件文件。图30-13所示的是upload.php的输出。

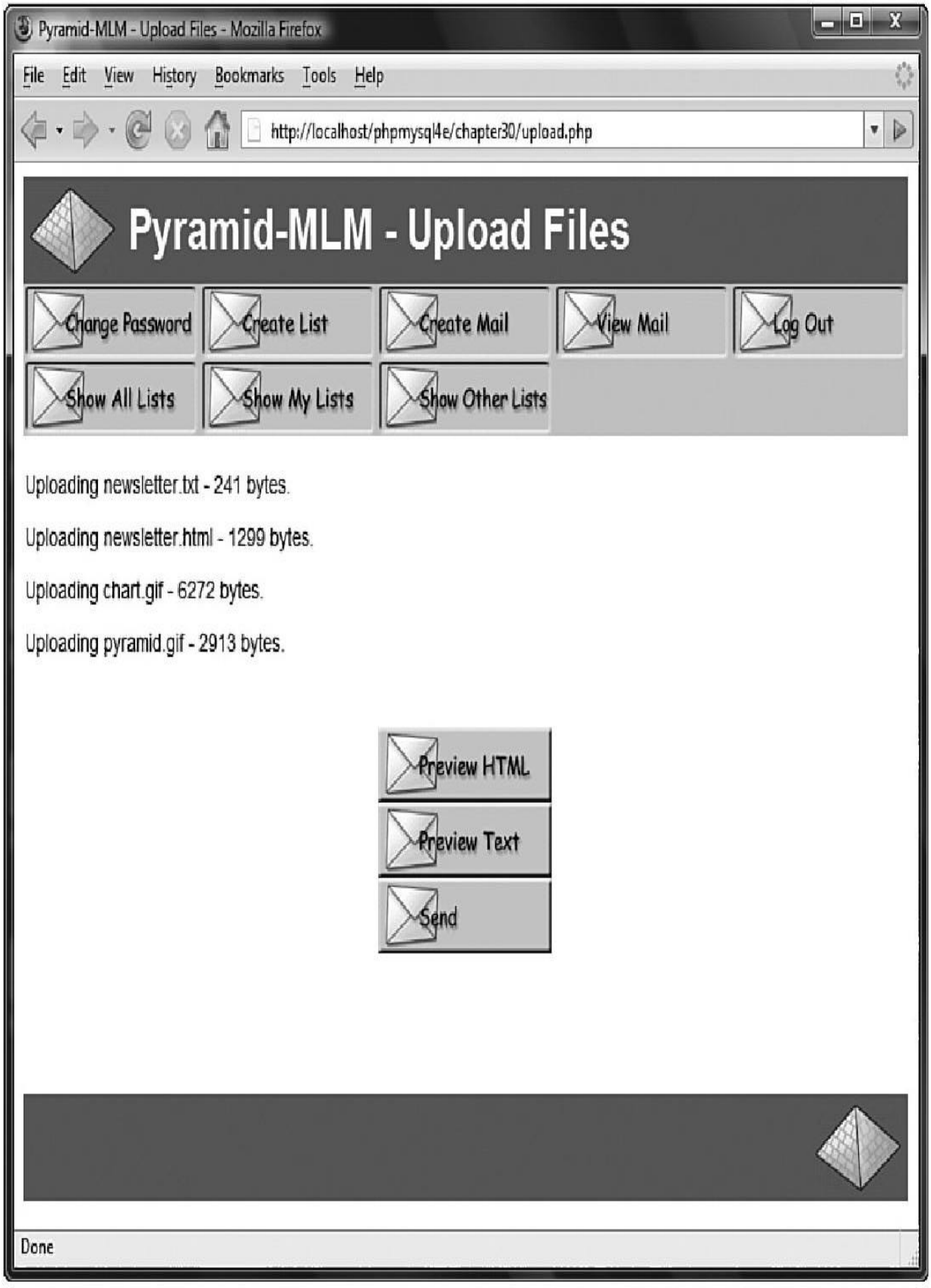

Pyramid-MLM - Upload Files - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/phpmysql4e/chapter30/upload.php
Pyramid-MLM - Upload Files
Change Password
Create List
Create Mail
View Mail
Log Out
Show All Lists
Show My Lists
Show Other Lists
Uploading newsletter.txt - 241 bytes.
Uploading newsletter.html - 1299 bytes.
Uploading chart.gif - 6272 bytes.
Uploading pyramid.gif - 2913 bytes.
Preview HTML
Preview Text
Send
Done

图　30-13　上传脚本将报告所上载的文件及其大小

30.7.4　预览新闻信件

在发送一个新闻信件前，管理员可以通过两种方法来预览新闻信件。如果希望上载后立即预览，可以从上载页面上使用预览功能。如果管理员希望预览邮件并在以后再发送，可以通过点击"View Mail"按钮，该按钮将显示系统中所有未发送的新闻信件。"View Mail"按钮将触发view-mail动作，触发如下代码：

```
case'view-mail':
display_items('Unsent Mail',get_unsent_mail(get_email()),
'preview-html','preview-text','send');
break;
```

可以看到，在preview-html、preview-text和send动作中，我们再次使用了display_items()函数。

请注意，"Preview"按钮实际上并不触发任何动作，而是直接连接到存档中的新闻信件。如果回顾程序清单30-7及程序清单30-16，可以看到，我们使用了display_preview_button()函数来创建这些按钮，而不是使用display_button()函数。

display_button()函数将创建一个指向特定脚本的图像链接，该函数必须给出一个GET参数；而display_preview_button()函数给出一个指向存档的普通链接。该链接将会显示在一个新的弹出窗口中，这可以通过在HTML的anchor标记中设置target=new属性来实现。我们将看到如图30-14所示的HTML版本的新闻信件。

Newsletter - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost/phpmysql4e/chapter30/archive/6/1/index.html

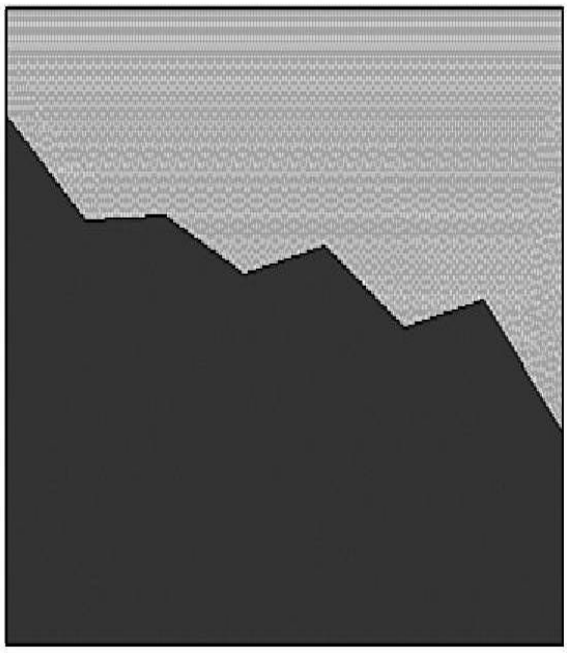
Pyramid-MLM

Press Release

According to data released today, Pyramid MLM Pty Ltd is not losing money as fast as some other companies.

Pyramid MLM lost only 22 million dollars this quarter.

This loss is only 10% more than the same period last year.



Done

图　30-14　对一个内嵌有图像的HTML格式新闻信件进行预览

30.7.5　发送邮件

点击新闻信件的发送按钮将触发发送动作，该动作将触发如下代码：

```
case'send':
send($_GET['id'],$_SESSION['admin_user']);
break;
```

该代码将调用send()函数，该函数可以在mlm_fns.php函数库中找到。这是一个相当长的函数。它也是我们使用Mail_mime类的地方。

该函数的代码如程序清单30-17所示。

程序清单30-17　mlm_fns.php函数库中的send()函数——该函数最终将发送一个新闻信件

```
//create the message from the stored DB entries and files
//send test messages to the administrator,or real messages to the whole list
function send($mailid,$admin_user){
if(!check_admin_user($admin_user)){
return false;
}
if(!($info=load_mail_info($mailid))){
echo"<p>Cannot load list information for message".$mailid."</p>";
return false;
}
$subject=$info['subject'];
$listid=$info['listid'];
$status=$info['status'];
$sent=$info['sent'];
$from_name='Pyramid MLM';
$from_address='return@address';
$query="select email from sub_lists where listid='".$listid."'";
```

```
$conn=db_connect();
$result=$conn->query($query);
if(!$result){
echo$query;
return false;
}else if($result->num_rows==0){
echo"<p>There is nobody subscribed to list number".$listid."</p>";
return false;
}
//include PEAR mail classes
include('Mail.php');
include('Mail/mime.php');
//instantiate MIME class and pass it the carriage return/line feed
//character used on this system
$message=new Mail_mime("\r\n");
//read in the text version of the newsletter
$textfilename="archive/".$listid."/".$mailid."/text.txt";
$tfp=fopen($textfilename,"r");
$text=fread($tfp,filesize($textfilename));
fclose($tfp);
//read in the HTML version of the newsletter
$htmlfilename="archive/".$listid."/".$mailid."/index.html";
$hfp=fopen($htmlfilename,"r");
$html=fread($hfp,filesize($htmlfilename));
fclose($hfp);
//add HTML and text to the mimemail object
$message->setTXTBody($text);
$message->setHTMLBody($html);
//get the list of images that relate to this message
$query="select path,mimetype from images where
mailid='".$mailid."'";
$result=$conn->query($query);
if(!$result){
echo"<p>Unable to get image list from database.</p>";
return false;
}
```

```
$num=$result->num_rows;
for($i=0;$i<$num;$i++){
//load each image from disk
$row=$result->fetch_array();
$imgfilename="archive/$listid/$mailid/".$row[0];
$imgtype=$row[1];
//add each image to the object
$message->addHTMLImage($imgfilename,$imgtype,
$imgfilename,true);
}
//create message body
$body=$message->get();
//create message headers
$from='"'.get_real_name($admin_user).'"<'.$admin_user.'>';
$hdrarray=array(
'From'=>$from,
'Subject'=>$subject);
$hdrs=$message->headers($hdrarray);
//create the actual sending object
$sender=&Mail::factory('mail');
if($status=='STORED'){
//send the HTML message to the administrator
$sender->send($admin_user,$hdrs,$body);
//send the plain text version of the message to administrator
mail($admin_user,$subject,$text,
'From:"'.get_real_name($admin_user).'"<'.$admin_user.'>');
echo"Mail sent to".$admin_user."";
//mark newsletter as tested
$query="update mail set status='TESTED'where
mailid='".$mailid."'";
$result=$conn->query($query);
echo"<p>Press send again to send mail to whole list.
<div align=\"center\">";
display_button('send','&id='.$mailid);
echo"</div></p>";
}else if($status=='TESTED'){
```

```
//send to whole list
$query="select subscribers.realname,sub_lists.email,
subscribers.mimetype
from sub_lists,subscribers
where listid=$listid and
sub_lists.email=subscribers.email";
$result=$conn->query($query);
if(!$result){
echo"<p>Error getting subscriber list</p>";
}
$count=0;
//for each subscriber
while($subscriber=$result->fetch_row()){
if($subscriber[2]=='H'){
//send HTML version to people who want it
$sender->send($subscriber[1],$hdrs,$body);
}else{
//send text version to people who don't want HTML mail
mail($subscriber[1],$subject,$text,
'From:"'.get_real_name($admin_user).'"
<'.$admin_user.'>');
}
$count++;
}
$query="update mail set status='SENT',sent=now()
where mailid='".$mailid."'";
$result=$conn->query($query);
echo"<p>A total of$count messages were sent.</p>";
}else if($status=='SENT'){
echo"<p>This mail has already been sent.</p>";
}
}
```

该函数将完成几项不同的操作。在将新闻信件发送出去之前，该函数测试性地将它发送给管理员。它通过记录数据库中邮件的状态来

保存这些信息。当上载程序脚本程序上载一封邮件时，该函数将该邮件的初始状态设置为"STORED"。

如果send()函数发现某邮件状态为"STORED"，则将它更改为"TESTED"并将它发送给管理员。"TESTED"状态表示该新闻信件已经被发送并经过管理员测试了。如果send()函数遇到某邮件状态为"TESTED"，则把它变成"SENT"并将它发送给列表上所有用户。这意味着每封邮件必须经过两次发送：一次是测试模式；一次是实际模式。

该函数还可以发送两种不同的邮件：通过PHP的mail()函数发送的文本版本，和用Mail_mime类发送的HTML版本。在本书中，我们已经在多处用到了mail()函数，因此在这里我们了解一下如何使用Mail_mime类。但是，我们并不会全面地讨论该类，只解释如何在这个典型的程序中使用它。

首先，我们将类文件包含进来并创建Mail_mime类的一个实例：

```
//include PEAR mail classes
include('Mail.php');
include('Mail/mime.php');
//instantiate MIME class and pass it the carriage return/line feed
//character used on this system
$message=new Mail_mime("\r\n");
```

请注意，我们在这里包含了两个类文件。我们还将使用PEAR软件包中的常规Mail类，正是这个类真正完成了邮件发送功能。这个类随同PEAR软件包一起安装。

Mail_mime类可以用来创建将要发送的MIME格式消息。

接下来，我们将读入邮件的文本和HTML版本，并且将它们添加到Mail_mime类：

```
//read in the text version of the newsletter
```

```
$textfilename="archive/".$listid."/".$mailid."/text.txt";
$tfp=fopen($textfilename,"r");
$text=fread($tfp,filesize($textfilename));
fclose($tfp);
//read in the HTML version of the newsletter
$htmlfilename="archive/".$listid."/".$mailid."/index.html";
$hfp=fopen($htmlfilename,"r");
$html=fread($hfp,filesize($htmlfilename));
fclose($hfp);
//add HTML and text to the mimemail object
$message->setTXTBody($text);
$message->setHTMLBody($html);
```

完成邮件读入后，该脚本循环地从数据库中载入每一个图片细节，并将其添加到需要发送的邮件中：

```
$num=$result->num_rows;
for($i=0;$i<$num;$i++){
//load each image from disk
$row=$result->fetch_array();
$imgfilename="archive/".$listid."/".$mailid."/".$row[0];
$imgtype=$row[1];
//add each image to the object
$message->addHTMLImage($imgfilename,$imgtype,$imgfilename,true);
}
```

在这里，传递给addHTMLImage()函数的参数分别是：图像文件的名称（或者我们可以传递图像数据）、图像的MIME类型、文件名称，以及用来标记第一个参数是图像文件名称而不是文件数据的true（如果想传递原始图像数据，可以传递数据、MIME类型、空参数和false）。这些参数有些繁琐。

在这一步，我们需要创建邮件正文。在设置邮件标题前，必须先创建邮件正文。我们按如下代码创建邮件正文：

```
//create message body
$body=$message->get();
```

接下来，可以调用Mail_mime类的headers()函数创建该邮件标题：

```
//create message headers
$from='"'.get_real_name($admin_user).'"<'.$admin_user.'>';
$hdrarray=array(
'From'=>$from,
'Subject'=>$subject);
```

最后，在设置完邮件后，就可以发送它了。在发送该邮件前，还需要初始化PEAR Mail类，并且将其传递给我们所创建的邮件。初始化该类的代码如下所示：

```
//create the actual sending object
$sender=&Mail::factory('mail');
```

（在这里，'mail'参数只是告诉Mail类使用PHP的mail()函数来发送邮件。很明显，也可以使用'sendmail'或'smtp'作为该参数值。）

接下来，我们将该邮件发送给每一个订阅者，通过检索该列表的每一个订阅者，并且根据用户的MIME类型，使用Mail的send()函数或常规的mail()函数来发送邮件：

```
if($subscriber[2]=='H'){
//send HTML version to people who want it
$sender->send($subscriber[1],$hdrs,$body);
}else{
//send text version to people who don't want HTML mail
mail($subscriber[1],$subject,$text,
'From:"'.get_real_name($admin_user).'"
<'.$admin_user.'>');
```

```
}
```

$sender- > send()函数的第一个参数应该是用户的电子邮件地址，第二个参数是邮件标题，而第三个参数是邮件正文。

这样就完成了邮件列表应用程序的构建。

30.8 扩展这个项目

就像这些常规项目一样，通常有许多种扩展功能的方法。我们可能希望做到以下几点。

■与订阅者确认成员关系，这样那些没有得到许可的人不能订阅。通常，可以通过向这些账户发送邮件并且删除那些没有回复的用户来实现。该方法也可以从数据库中清除任何不正确的邮件地址。

■给予管理员批准或拒绝申请订阅列表的用户的权限。

■增加开放列表功能以使任何成员都向列表发送电子邮件。

■对于某特定的邮件列表仅注册成员可以看到存档。

■允许用户搜索满足特定条件的列表。例如，用户可能对高尔夫球新闻信件感兴趣。一旦用户数量超过一定的大小，搜索指定的邮件的功能将会很有用。

■如果希望创建一个大型的邮件列表，PHP应用程序可能不是发送这些邮件的理想选择。一个实用的邮件列表管理器（例如，ezmlm）的效率会比在PHP中调用mail()函数高得多。当然，我们还可以使用PHP来构建前台程序，而使用ezmlm来处理繁重的工作。

30.9 下一章

在下一章中，我们实现一个Web论坛程序，该论坛允许用户联机讨论，这些讨论可以是按照主题和对话内容进行组织的。

第31章　创建一个Web论坛

吸引网站用户的一个好方法是建立Web论坛。它可以用于哲学讨论组和产品技术支持等多种用途。在本章中，我们将使用PHP来实现Web论坛。当然，也可以选择已有的软件包，例如Phorum，来建立自己的论坛。

Web论坛有时候又叫做讨论公告牌或主题讨论组。论坛的想法就是允许论坛中的用户发表文章或提出问题，而其他人可以阅读并回复这些问题。在论坛中，每个讨论的主题叫做一个话题（thread）。

在这个项目中，我们要实现一个名为blah-blah的论坛，该论坛具有下面的功能。用户可以：

■通过发表文章开始新的讨论话题

■发表文章来回复已有的文章

■查看发表的文章

■查看论坛中的交谈话题

■查看文章之间的关系，也就是，查看哪篇文章是另外文章的回复

31.1　理解流程

建立论坛事实上是一个相当有趣的流程。我们需要一些方法来在数据库中存储文章的作者、标题、发表日期和内容信息。初看觉得这似乎与Book-O-Rama数据库没有多大不同。

然而，大多数主题讨论软件都以如下方式工作：给用户显示文章的同时，还显示出这些文章之间的关系。也就是说，可以看出哪些文章是回复另外一些文章的（以及它们是哪篇文章的回复），以及哪些文章是新讨论的话题。

我们可以看到在许多地方看到实现这些功能的论坛例子，包括Slashdot:http://slashdot.org。

需要仔细考虑的地方是确定如何显示这些文章之间的关系。对于目前这个系统来说，用户应该能够查看单个的消息，一个显示有相互关系的谈论话题，或者系统中所有话题。

用户还必须能发表新主题的文章或者进行回复。这只是简单的部分。

31.2 解决方案的组成

正如我们前面介绍的，保存及检索作者和消息文本是非常容易的。该应用程序最困难的部分是如何找到一种保存所需信息的数据库结构，以及一种高效率浏览该结构的方法。

如图31-1所示就是一个可能的正在讨论中的文章结构。

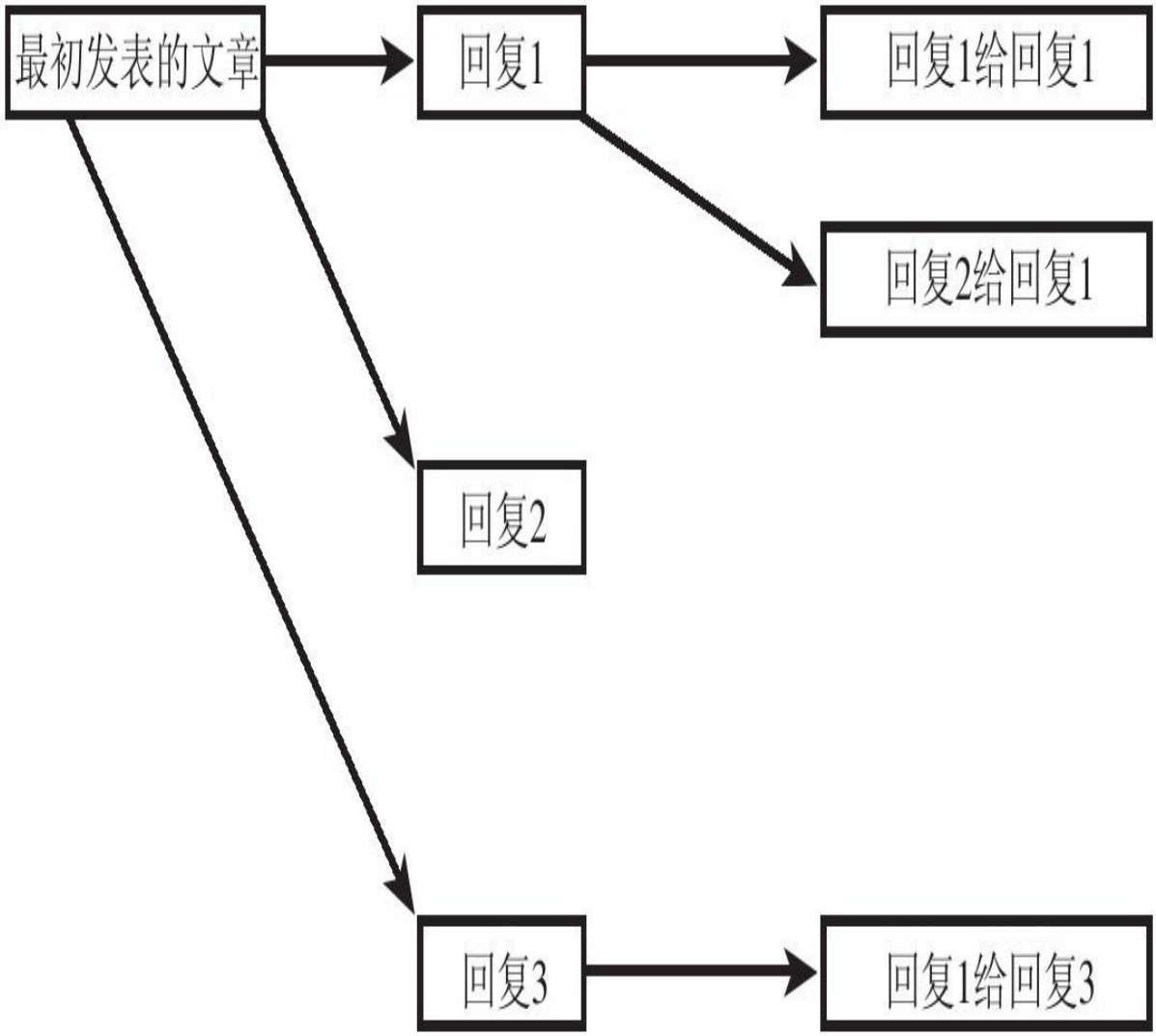


图　31-1　主题讨论中的一个文章可能是一个新话题的第一篇文章，也可能是对其他文章的响应

在图31-1中，可以看到关于某主题的一个初始发表的文章，该文章具有3篇回复文章。这些回复文章中的某些又有回复文章。这些文章还可能回复等。

仔细查看图31-1，可以得到如下启示，我们应该如何保存并检索文章数据及文章间的连接。图中显示了一种树形结构。如果我们做过很多编程工作，会发现这是一种很常用的数据结构。在图31-1中，有节点（或说文章）和连接（或文章之间的关系），正如任何树形结构一

样。（如果对树形数据结构不熟悉，不用担心——我们会在开始主题讨论之前介绍这些基本知识。）

掌握这些知识的诀窍是：

1)找到一种将这些树形结构映射到物理存储器的方法——在这个例子中，就是映射到MySQL数据库。

2)找到一种能够按要求重建数据的方法。

对于这个项目，我们将以实现一个能够保存使用当中的文章的MySQL数据库为开始。我们将创建一些简单的界面来实现文章的保存。

当载入待显示的文章列表时，我们会将每篇文章的标题放进一个tree_node的PHP类。每一个tree_node将包含一个文章标题以及对该文章的回复。

这些回复将保存在一个数组中，而每个回复本身又是一个tree_node，它可以包含由该文章的回复组成的一个数组，而这些回复本身同样又是tree_node，如此继续下去。直到遇到我们所说的叶子节点（leaf node），叶子节点本身不再包含回复。这样，我们就有了一个如图31-1所示的树形结构。

在这里，我们先介绍一些术语：要回复的消息称作当前节点的父节点。该消息的任何回复可以称作当前节点的子节点。如果把该树形结构想象成家庭树，就很容易记住这一点。

该树形结构的第一篇文章（没有父节点的那篇）有时候叫做根节点。

提示　将第一篇文章称作根节点可能不太直观，因为通常我们都将根节点画在图的顶部，不像实际的树根。

要建立并显示这个树形结构，我们将编写一些递归函数。（在第5章中，我们已经讨论过递归函数。）

我们决定使用一个类来表示这种数据结构，因为对于这个应用程序来说，这样更易于建立一种复杂的、动态可扩展的数据结构。它也意味着，我们可以使用简单而别致的代码来完成相当复杂的工作。

31.3　解决方案概述

为了真正理解在该项目中我们所做的工作，阅读其代码是一个不错的方法。接下来，我们就研究代码。与其他项目比起来，该项目包含冗长程序不多，都是简短的代码，但是代码却更复杂。

在该程序中，实际上仅有3个页面。它有一个主索引页，显示论坛中所有文章并指向这些文章的链接。从这里，可以增加新文章，查看已列出的文章，或通过展开和折叠树枝来改变查看文章的方法。（稍后将详细介绍）。在文章视图页面中，可以回复该文章或查看已存在的回复。在新文章页面中，可以发表一篇新文章、或者一篇对已存在文章的回复、或者一篇无关的新消息。

该系统的流程图如图31-2所示。

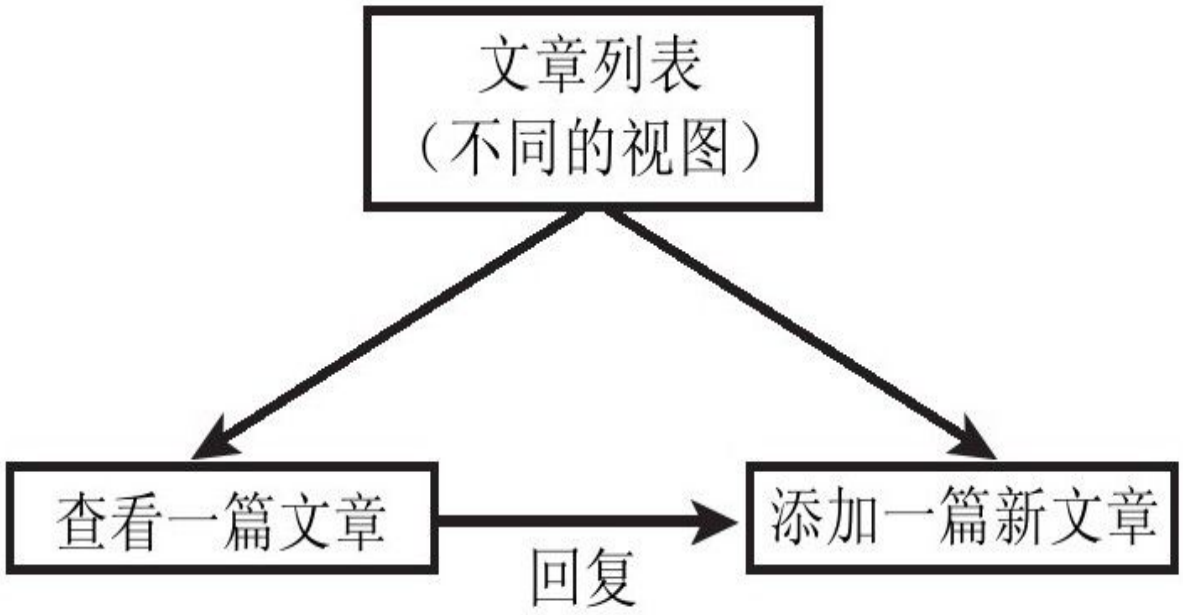


图　31-2　在blah-blah论坛系统中的3个主要部分

该程序使用到的文件汇总如表31-1所示。

表31-1 Web论坛应用程序中用到的文件

文件名称	类　型	描　述
index.php	应用程序	用户进入站点看到的主页。包括该站点中所有文章的可扩展及删除的列表
new_post.php	应用程序	用来发表新文章的表单
store_new_post.php	应用程序	保存已输入到new_post.php表单中的文章
view_post.php	应用程序	显示一篇单独的文章及回复它的文章的列表
treenode_class.php	函数库	包含节点类，我们将要用它来显示文章的继承关系
include_fns.php	函数库	将该程序中用到的所有其他函数库放到一起（其他库类型文件在这里列出）
data_valid_fns.php	函数库	数据检验函数
db_fns.php	函数库	数据库连接函数
discussion_fns.php	函数库	处理存储及重新获得发表的文章的函数
output_fns.php	函数库	输出HTML的函数
create_database.sql	SQL	建立该程序所需要数据库的SQL脚本

下面，我们继续来了解程序的实现。

31.4 数据库的设计

我们需要保存一些关于发表到论坛中的文章的属性，其中包括：文章的作者（称作poster）、文章的标题、发表时间和文章正文。因此我们需要一个文章表，还必须为每一篇文章创建一个唯一的ID，称作postid。

每篇文章需要一些关于它在继承关系中所属位置的信息。我们可以将关于文章的子文章的位置随该文章一起保存起来。然而，每篇文章都会有许多回复，因此这可能会在数据库的结构上导致某些问题。由于每篇文章只能是一篇文章的回复，因此保存父文章的引用可能会更容易些，父文章即当前文章回复的那篇文章。

这样我们需要为每篇文章保存以下数据：

■postid：每篇文章的唯一的ID

■parent：父文章的postid

■poster：该文章的作者

■title：该文章的标题

■posted：该文章发表的时间和日期

■message：该文章的正文

当然，我们还需要对以上数据进行一些优化。

当需要判断一篇文章是否有回复时，必须运行一个SQL查询来确定是否有任何的文章将该文章作为其父文章。我们需要为每一篇所列出文章生成这些信息。需要运行的查询语句越少，代码运行的速度就越快。我们可以通过增加一个表示是否有任何的回复的字段来去除这种查询的必要。我们把这个字段叫做children并将它设为查询效率较高的Boolean类型——如果节点有"children"，则为1，否则为0。

当然，优化总是要付出代价的。这里，我们选择了存储冗余数据。由于我们以两种方式保存这些数据，必须保证两种表示方法互相匹配。当增加子节点时，必须更新其父节点。如果删除子节点，需要更新父节点确保数据库的一致性。在该项目中，我们并不打算添加删除文章的功能，因此可以避免一半的问题。如果决定扩展该代码，请记住这个问题。

在这个项目中，我们还需要完成一些其他优化。我们将消息体与其他数据分离，并且将这些消息体保存在一个单独的表格中。这样做可以使消息体具有MySQL的text属性。在一个表中使用这种类型的数据将降低该表的查询速度。由于我们要完成许多小查询来建立树形结构，因此这会使该系统的速度下降很多。将消息正文放在单独的表中，我们可以在用户需要查看某个特定消息时，将它从数据库中取出。

MySQL查询固定大小记录的速度比查询可变大小记录的速度要快。如果想使用可变大小的数据，可以通过对用来检索数据库的字段上创建索引来实现。当然对于某些项目来说，将文本字段与其他字段一样放在相同的记录中，并为所有要基于这些字段的检索指定索引可能也会好些。创建索引要花费时间，而且论坛数据可能经常会发生变化，因此我们必须经常重新建立索引。

我们还增加一个area属性以备以后决定在一个程序中实现多个论坛。在这里，我们不实现这个功能，但是该方法可以保留为将来使用。

基于以上所有这些考虑，创建论坛数据库的SQL脚本如程序清单31-1所示。

程序清单31-1　create_database.sql——创建论坛数据库的SQL脚本

```
create database discussion;
use discussion;
create table header
(
parent int not null,
```

```
poster char(20)not null,
title char(20)not null,
children int default 0 not null,
area int default 1 not null,
posted datetime not null,
postid int unsigned not null auto_increment primary key
);
create table body
(
postid int unsigned not null primary key,
message text
);
grant select,insert,update,delete
on discussion.*
to discussion@localhost identified by'password';
```

可以在MySQL中运行如下所示的脚本来创建这个数据库结构：

```
mysql-u root-p＜create_database.sql
```

需要提供root用户密码。此外，还应该更改我们为论坛用户设置的密码。

为了更好地理解该结构如何保持文章及它们相互之间存在的关系，请参阅图31-3。

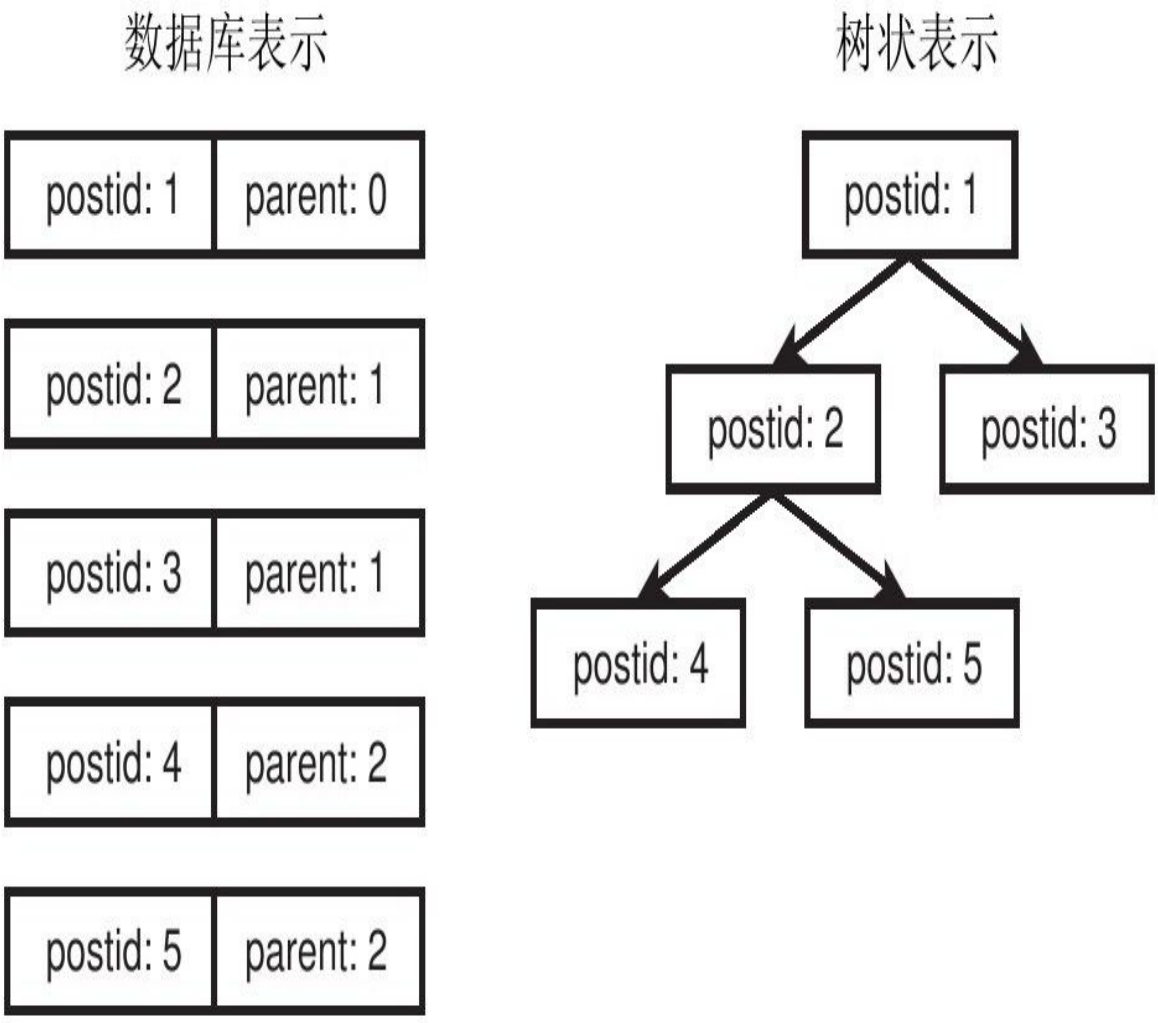


图　31-3　以一个平面关系表格方式保存的树形结构数据

可以看到，数据库中每篇文章的父字段保存了树形结构中它上面的文章的postid。父文章是将被回复的文章。

我们还可以看到，根节点postid 1没有父节点，讨论组所有新主题都在这个位置。对于这种类型的文章，我们在数据中将它们的父节点保存为0。

31.5　查看文章的树形结构

接下来，我们需要一种方法将信息从数据库中取出，并将其在树形结构中表示出来。我们在主页面index.php中完成这项工作。为了将这个过程解释得更加清楚，我们通过文章粘贴脚本程序new_post.php和store_new_post.php输入了一些样本帖子。我们将在下一节介绍这两个程序。

这里，我们首先讨论文章列表，因为它是整个站点的主体。然后其他内容就容易理解了。

如图31-4所示的是用户将看到的该站点初始的文章视图。

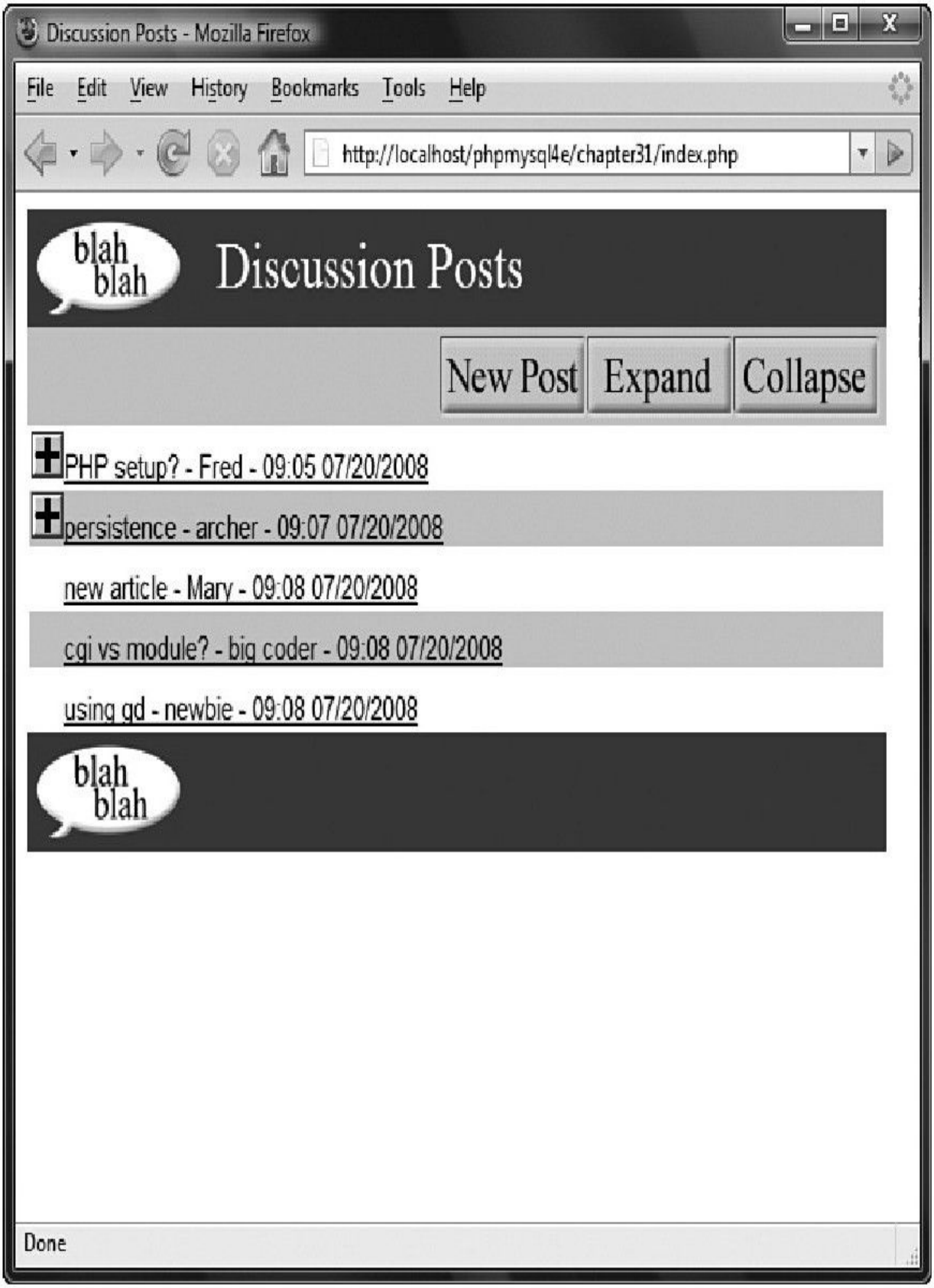

Discussion Posts - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/phpmysql4e/chapter31/index.php
blah blah
Discussion Posts
New Post
Expand
Collapse
PHP setup? - Fred - 09:05 07/20/2008
persistence - archer - 09:07 07/20/2008
new article - Mary - 09:08 07/20/2008
cgi vs module? - big coder - 09:08 07/20/2008
using gd - newbie - 09:08 07/20/2008
blah blah
Done

图　31-4　文章列表的初始视图显示了在折叠表单中的文章

我们这里看到的都是最初的文章，它们都没有回复文章；它们都是某特定主题的第一篇文章。

在这个例子中，还可以看到有很多选项。有一个菜单栏，我们可以用它来添加一篇新文章以及展开或折叠文章的视图。

为了理解这一点，请看这些发表的文章。其中有一些文章前面有“+”号，这表示这些文章已经被回复过。如果要查看特定文章的回复，可以点击该加号标志。如图31-5所示的就是点击这些符号产生的一个结果。

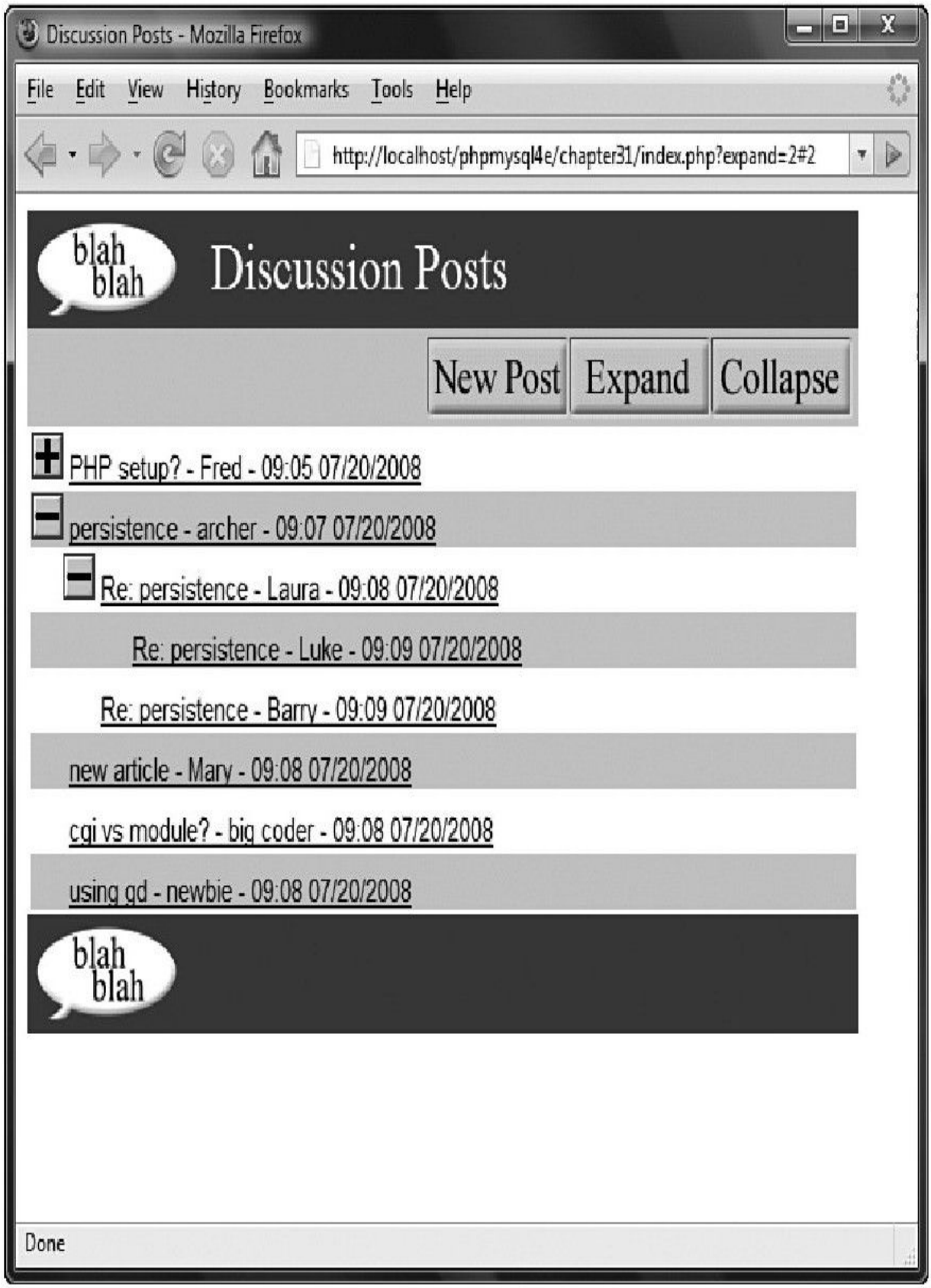

Discussion Posts - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/phpmysql4e/chapter31/index.php?expand=2#2
blah blah
Discussion Posts
New Post
Expand
Collapse
PHP setup? - Fred - 09:05 07/20/2008
persistence - archer - 09:07 07/20/2008
Re: persistence - Laura - 09:08 07/20/2008
Re: persistence - Luke - 09:09 07/20/2008
Re: persistence - Barry - 09:09 07/20/2008
new article - Mary - 09:08 07/20/2008
cgi vs module? - big coder - 09:08 07/20/2008
using gd - newbie - 09:08 07/20/2008
blah blah
Done

图　31-5　关于PHP持续性的讨论话题已被展开

可以看到，点击加号后会显示出第一篇文章的回复。而加号则变成了减号。如果我们再点击它，在这个话题里的所有文章都将会折叠起来，又返回至初始视图。

可以看到，其中的一篇回复前面也有个加号。这表示这篇回复也有回复。如此重复操作，可以一直点击下去，通过点击相应的加号来查看每篇回复。

在菜单栏中，有两个菜单选项，Expand和Collapse，分别为展开和折叠所有可能的话题。点击Expand按钮的结果如图31-6所示。

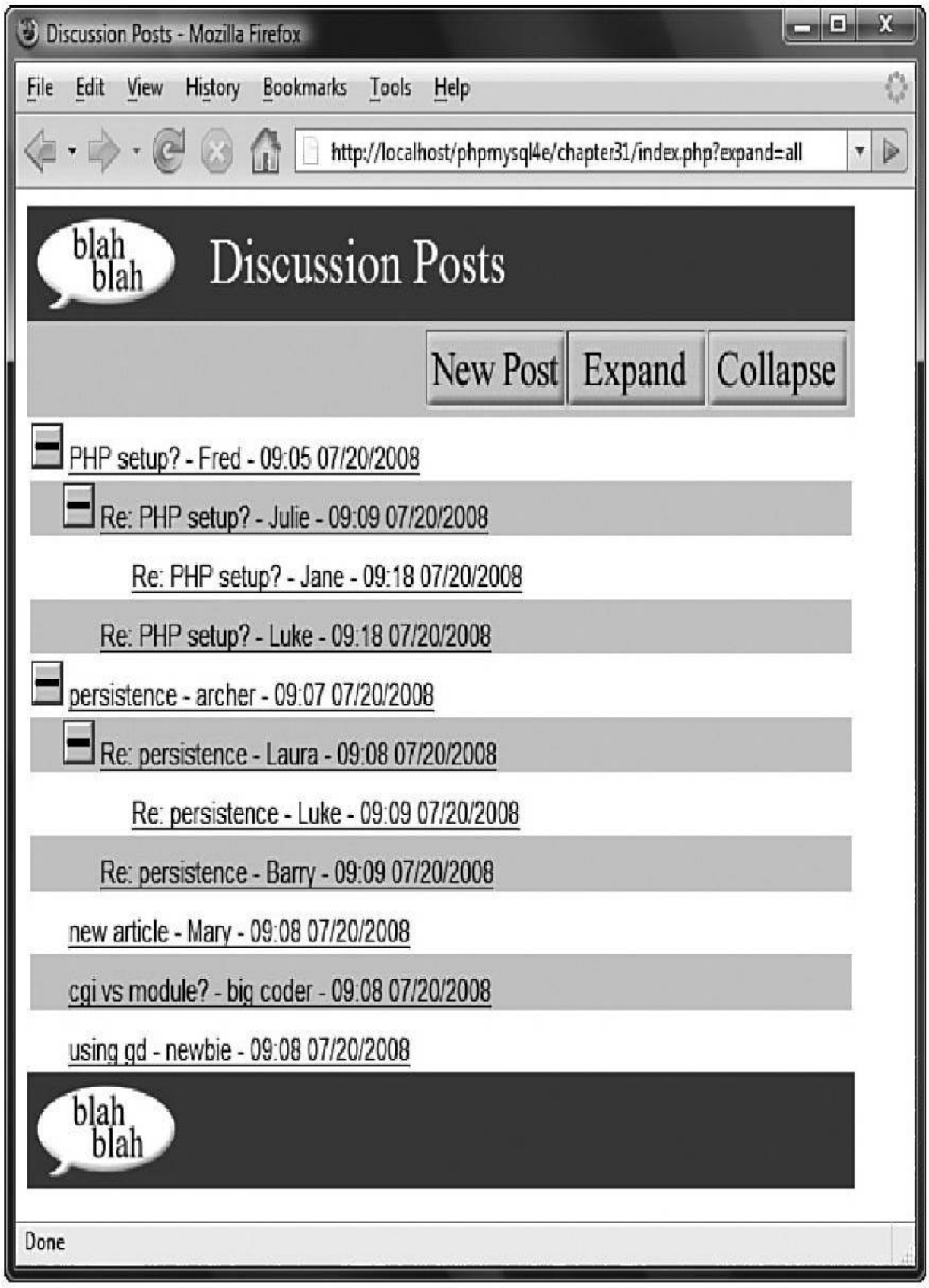

Discussion Posts - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/phpmysql4e/chapter31/index.php?expand=all
blah blah
Discussion Posts
New Post
Expand
Collapse
PHP setup? - Fred - 09:05 07/20/2008
Re: PHP setup? - Julie - 09:09 07/20/2008
Re: PHP setup? - Jane - 09:18 07/20/2008
Re: PHP setup? - Luke - 09:18 07/20/2008
persistence - archer - 09:07 07/20/2008
Re: persistence - Laura - 09:08 07/20/2008
Re: persistence - Luke - 09:09 07/20/2008
Re: persistence - Barry - 09:09 07/20/2008
new article - Mary - 09:08 07/20/2008
cgi vs module? - big coder - 09:08 07/20/2008
using gd - newbie - 09:08 07/20/2008
blah blah
Done

图　31-6　所有话题都已被展开

如果你仔细查看图31-5和图31-6，可以看到我们在命令行中将一些参数传回了index.php。在图31-5中，URL框显示了如下所示字符串：

```
http://localhost/phpmysql4e/chapter31/index.php?expand=2#2
```

以上脚本将上面这行解释为“展开postid为2的节点”。“#”号仅是个HTML标记，用来滚动页面直到刚才展开的那个节点。

```
http://localhost/phpmysql4e/chapter31/index.php?expand=all
```

在图31-6中，URL是点击"Expand"按钮将传递一个值为all的expand参数。

31.5.1　展开和折叠

要了解如何创建文章视图，让我们来研究index.php脚本，如程序清单31-2所示。

程序清单31-2　index.php——在应用程序主页面创建文章视图的脚本

```
<?php
include('include_fns.php');
session_start();
//check if we have created our session variable
if(!isset($_SESSION['expanded'])){
$_SESSION['expanded']=array();
}
//check if an expand button was pressed
//expand might equal'all'or a postid or not be set
if(isset($_GET['expand'])){
if($_GET['expand']=='all'){
```

```
expand_all($_SESSION['expanded']);
}else{
$_SESSION['expanded'][$_GET['expand']]=true;
}
}
//check if a collapse button was pressed
//collapse might equal all or a postid or not be set
if(isset($_GET['collapse'])){
if($_GET['collapse']=='all'){
$_SESSION['expanded']=array();
}else{
unset($_SESSION['expanded'][$_GET['collapse']]);
}
}
do_html_header('Discussion Posts');
display_index_toolbar();
//display the tree view of conversations
display_tree($_SESSION['expanded']);
do_html_footer();
?>
```

以上脚本使用3个变量来实现其功能。它们是：

■会话变量expanded，用来记录那些已经被展开的记录。该变量可以在不同的视图之间使用，因此我们可展开多个话题。该变量是一个数组，它包含了与那些需要展开并显示其回复文章相关的postid数组。

■参数expand，告诉脚本哪个新话题需要展开。

■参数collapse，通知脚本哪个话题需要折叠。

当点击加号或减号，或者点击Expand或Collapse按钮时，它们将再次调用index.php脚本，不过将带有新的参数expand和collapse。我们使用expanded变量在页面间记录哪些话题在任何给定的视图中应该展开。

该脚本以初始化一个会话开始，并增加expanded变量作为会话变量，如果这个操作还没有完成的话。之后，该脚本将检查是否传递了expand或collapse参数，并相应地修改expanded数组。请看以下关于expand参数的代码：

```
if(isset($_GET['expand'])){
if($_GET['expand']=='all'){
expand_all($_SESSION['expanded']);
}else{
$_SESSION['expanded'][$_GET['expand']]=true;
}
}
```

如果点击了Expand按钮，该操作将调用expand_all()函数，这样所有有回复的话题都将加入到expanded数组中（稍后将详细介绍）。

如果想要扩展特定话题，可以通过expand变量传递一个postid。因此，我们可以在expanded数组中增加一条新记录来反映它。

expand_all()函数如程序清单31-3所示。

程序清单31-3　discussion_fns.php函数库中的expand_all()函数——对$expanded数组进行操作以展开论坛中所有话题

```
function expand_all(&$expanded){
//mark all threads with children as to be shown expanded
$conn=db_connect();
$query="select postid from header where children=1";
$result=$conn->query($query);
$num=$result->num_rows;
for($i=0;$i<$num;$i++){
$this_row=$result->fetch_row();
$expanded[$this_row[0]]=true;
}
}
```

该函数运行一个数据库查询语句来找出论坛中哪些话题有回复，如下所示：

```
select postid from header where children=1
```

每一篇返回的文章都将被添加到expanded数组中。我们运行查询语句是为了节省时间。也可以简单地将所有文章加入到expanded列表中，但是尝试处理那些不存在的回复相当浪费时间。

折叠文章节点的操作与此非常相似，不过是以相反的方法进行的，如下所示：

```
if(isset($_GET['collapse'])){
if($_GET['collapse']=='all'){
$_SESSION['expanded']=array();
}else{
unset($_SESSION['expanded'][$_GET['collapse']]);
}
}
```

可以通过重置该数组，从该数组中删除所有条目。如果整个页面都需要折叠，就删除需要折叠的话题或重置整个数组。

以上所有这些操作都只是预处理，由此我们可以知道哪些文章应该显示出来而哪些不应该显示。该脚本的主要部分是对display_tree($_SESSION['expanded'])；函数的调用，由它实际产生要显示的文章树形结构。

31.5.2　显示文章

下面，我们来了解display_tree()函数，如程序清单31-4所示。

程序清单31-4　output_fns.php函数库中的display_tree()函数——建立树形结构的根节点

```
function display_tree($expanded,$row=0,$start=0){
//display the tree view of conversations
global$table_width;
echo"<table width=\"".$table_width."\">";
//see if we are displaying the whole list or a sublist
if($start>0){
$sublist=true;
}else{
$sublist=false;
}
//construct tree structure to represent conversation summary
$tree=new treenode($start,'','','',1,true,-1,$expanded,
$sublist);
//tell tree to display itself
$tree->display($row,$sublist);
echo"</table>";
}
```

该函数的主要作用是创建树形结构的根节点。我们用它来显示整个索引，并在view_post.php页面上建立回复文章的子树。可以看到，它带有3个参数。第1个参数是$expanded，它是要以展开形式显示的文章列表。第2个参数是$row，是一个行号，用来计算在列表中该行的交替颜色。

第3个参数是$start，告知函数在哪里开始显示文章。这是将被创建并显示的树的根节点的postid。如果我们打算显示所有文章，如主页

面一样，该变量将是0，表示显示所有没有父节点的文章。如果该参数是0，我们将$sublist变量设为false，并显示整个树。

如果该参数大于0，我们将它作为该树的根节点来显示，将$sublist变量设为true，可以建立和显示该树的一部分（我们将在view_post.php脚本程序中用到它）。

该函数最重要的一个功能就是初始化一个treenode类的实例，它代表该树的根节点。它实际上并不是一篇文章，但它相当于所有第一级文章的父文章，而事实上第一级文章是没有父节点的。树建立以后，我们将调用显示函数来显示文章列表。

31.5.3　使用treenode类

treenode类的代码如程序清单31-5所示。（在这里，我们会发现它非常有用，查阅第6章回忆它是如何工作的。）

程序清单31-5　treenode_class.php函数库中的treenode类——该应用程序的主体架构

```
<?php
//functions for loading,constructing and
//displaying the tree are in this file
class treenode{
//each node in the tree has member variables containing
//all the data for a post except the body of the message
public$m_postid;
public$m_title;
public$m_poster;
public$m_posted;
public$m_children;
public$m_childlist;
public$m_depth;
public function__construct($postid,$title,$poster,$posted,
$children,$expand,$depth,$expanded,$sublist){
//the constructor sets up the member variables,but more
//importantly recursively creates lower parts of the tree
$this->m_postid=$postid;
$this->m_title=$title;
$this->m_poster=$poster;
$this->m_posted=$posted;
$this->m_children=$children;
$this->m_childlist=array();
$this->m_depth=$depth;
//we only care what is below this node if it
//has children and is marked to be expanded
```

```
//sublists are always expanded
if(($sublist||$expand)&&$children){
$conn=db_connect();
$query="select*from header where
parent='".$postid."'order by posted";
$result=$conn->query($query);
for($count=0;$row=@$result->fetch_assoc();$count++){
if($sublist||$expanded[$row['postid']]==true){
$expand=true;
}else{
$expand=false;
}
$this->m_childlist[$count]=new treenode($row['postid'],
$row['title'],$row['poster'],$row['posted'],
$row['children'],$expand,$depth+1,$expanded,
$sublist);
}
}
}
function display($row,$sublist=false){
//as this is an object,it is responsible for displaying itself
//$row tells us what row of the display we are up to
//so we know what color it should be
//$sublist tells us whether we are on the main page
//or the message page.Message pages should have
//$sublist=true.
//On a sublist,all messages are expanded and there are
//no"+"or"-"symbols.
//if this is the empty root node skip displaying
if($this->m_depth>-1){
//color alternate rows
echo"<tr><td bgcolor=\"";
if($row%2){
echo"#cccccc\">";
}else{
echo"#ffffff\">";
```

```
}
//indent replies to the depth of nesting
for($i=0;$i<$this->m_depth;$i++){
echo"<img src=\"images/spacer.gif\"height=\"22\"
width=\"22\"alt=\"\"valign=\"bottom\"/>";
}
//display+or-or a spacer
if((!$sublist)&&($this->m_children)&&
(sizeof($this->m_childlist))){
//we're on the main page,have some children,and they're expanded
//we are expanded-offer button to collapse
echo"<a href=\"index.php?collapse=".
$this->m_postid."#".$this->m_postid."\"><img
src=\"images/minus.gif\"valign=\"bottom\"
height=\"22\"width=\"22\"alt=\"Collapse Thread\"
border=\"0\"/></a>\n";
}else if(!$sublist&&$this->m_children){
//we are collapsed-offer button to expand
echo"<a href=\"index.php?expand=".
$this->m_postid."#".$this->m_postid."\"><img
src=\"images/plus.gif\"valign=\"bottom\"
height=\"22\"width=\"22\"alt=\"Expand Thread\"
border=\"0\"/></a>\n";
}else{
//we have no children,or are in a sublist,do not give button
echo"<img src=\"images/spacer.gif\"height=\"22\"
width=\"22\"alt=\"\"valign=\"bottom\"/>\n";
}
echo"<a name=\"".$this->m_postid."\"><a href=
\"view_post.php?postid=".$this->m_postid."\">".
$this->m_title."-".$this->m_poster."-".
reformat_date($this->m_posted)."</a></td></tr>";
//increment row counter to alternate colors
$row++;
}
//call display on each of this node's children
```

```
//note a node will only have children in its list if expanded
$num_children=sizeof($this->m_childlist);
for($i=0;$i<$num_children;$i++){
$row=$this->m_childlist[$i]->display($row,$sublist);
}
return$row;
}
}
?>
```

该类包含主应用程序中用来驱动树形视图的功能。

treenode类的一个实例包括关于帖子及到所有对该帖子回复的帖子的链接细节信息。这样，我们就必须为该类定义如下所示的成员变量：

```
public$m_postid;
public$m_title;
public$m_poster;
public$m_posted;
public$m_children;
public$m_childlist;
public$m_depth;
```

请注意，treenode并不包含文章的正文。在用户访问view_post.php脚本之前，没有必要载入文章的正文。我们需要尽量使这些运行得相对较快，因为在显示树型列表时，我们还需要完成许多的数据操作，而且在页面刷新时或者按下某个按钮时需要重新计算。

这些变量的命名方法遵循在面向对象应用程序中普遍使用的命名规则——变量名称以m_开始提醒我们它们是类的成员变量。

这些变量的大部分直接对应于数据库中header表的数据行。$m_childlist和$m_depth变量是例外。我们用变量$m_childlist来保存该

文章的回复。而变量$m_depth保存我们深入树的层次数——这个信息可以用来创建视图。

该类的构造函数将设置所有变量的值，如下所示：

```
public function__construct($postid,$title,$poster,$posted,$children,
$expand,$depth,$expanded,$sublist){
//the constructor sets up the member variables,but more
//importantly recursively creates lower parts of the tree
$this->m_postid=$postid;
$this->m_title=$title;
$this->m_poster=$poster;
$this->m_posted=$posted;
$this->m_children=$children;
$this->m_childlist=array();
$this->m_depth=$depth;
```

当我们在主页面的display_tree()函数中构造根treenode节点时，实际上创建了一个虚节点，没有文章与之相关。我们将传递一些初始值：

```
$tree=new treenode($start,'','','',1,true,-1,$expanded,$sublist);
```

以上代码将创建一个$postid为0的根节点。它可用来找出所有第一级的文章，因为它们有一个postid为0的父节点。我们将它的深度设为-1是因为该节点实际上并不是视图的一部分。所有第一级的帖子的深度都为零，处于屏幕的最左端。接下来的深度慢慢向右扩展。

在该构造函数中，最重要的一点是当前节点的子节点被初始化了。该过程中我们首先检查是否需要扩展子节点。仅当某节点有子节点时才执行该过程，并已选择了显示它们：

```
if(($sublist||$expand)&&$children){
$conn=db_connect();
```

接下来，我们将连接到数据库，并取出所有子文章，如下所示：

```
$query="select*from header where parent='".$postid."'order by posted";
$result=$conn->query($query);
```

然后，使用treenode类的实例填充$m_childlist数组，该数组将包含存储在该节点中所有的回复，如下所示：

```
for($count=0;$row=@$result->fetch_assoc();$count++){
if($sublist||$expanded[$row['postid']]==true){
$expand=true;
}else{
$expand=false;
}
$this->m_childlist[$count]=new treenode($row['postid'],$row['title'],
$row['poster'],$row['posted'],$row['children'],$expand,
$depth+1,$expanded,$sublist);
}
```

最后一行代码将创建新的treenode节点，它完全遵照前面讨论过的过程，然而这是对于该树的下一个层次来说的，这里是递归部分。一个父节点调用treenode的构造函数，将自身的postid作为父节点传递，并在传递前将自身的深度加1。

依次创建每个树节点及它的子节点，直到不再有回复或者到达我们想要扩展的深度为止。

完成以上操作后，我们将调用根节点的显示函数（回顾display_tree()函数），如下所示：

```
$tree->display($row,$sublist);
```

display()函数首先检查它是否是虚根节点：

```
if($this->m_depth>-1)
```

通过该方法，可以将虚节点从视图中去除。然而我们并不希望完全跳过根节点。虽然我们不显示它，但它需要通知它的子节点来显示它们自己。

该函数接下来将绘制包含文章的表。它使用取模操作符（%）来判断该行的背景色应该是什么（因此两种背景色需要交替变化）：

```
//color alternate rows
echo"<tr><td bgcolor=\"";
if($row%2){
echo"#cccccc\">";
}else{
echo"#ffffff\">";
}
```

接下来，该函数使用$m_depth成员变量来计算当前条目应该缩进多少。回顾前面的图例，可以看到，回复的层次越深，缩进得越多。这是通过下面的代码来完成的：

```
//indent replies to the depth of nesting
for($i=0;$i<$this->m_depth;$i++){
echo"<img src=\"images/spacer.gif\"height=\"22\"
width=\"22\"alt=\"\"valign=\"bottom\"/>";
}
```

接下来，该函数将判断是否应提供加号或减号或者什么都没有：

```
//display+or-or a spacer
if(!$sublist&&$this->m_children&&sizeof($this->m_childlist)){
```

```
//we're on the main page,have some children,and they're expanded
//we are expanded-offer button to collapse
echo"<a href=\"index.php?collapse=".
$this->m_postid."#".$this->m_postid."\"><img
src=\"images/minus.gif\"valign=\"bottom\"height=\"22\"
width=\"22\"alt=\"Collapse Thread\"border=\"0\"/></a>\n";
}else if(!$sublist&&$this->m_children){
//we are collapsed-offer button to expand
echo"<a href=\"index.php?expand=".
$this->m_postid."#".$this->m_postid."\"><img
src=\"images/plus.gif\"valign=\"bottom\"height=\"22\"
width=\"22\"alt=\"Expand Thread\"border=\"0\"/></a>\n";
}else{
//we have no children,or are in a sublist,do not give button
echo"<img src=\"images/spacer.gif\"height=\"22\"width=\"22\"
alt=\"\"valign=\"bottom\"/>\n";
}
```

完成以上操作后，将显示该节点的详细内容：

```
echo"<a name=\"".$this->m_postid."\"><a href=
\"view_post.php?postid=".$this->m_postid."\">".
$this->m_title."-".$this->m_poster."-".
reformat_date($this->m_posted)."</a></td></tr>";
```

我们改变下一行的颜色：

```
//increment row counter to alternate colors
$row++;
```

之后，有一些代码将被包括根节点在内的所有节点执行，如下所示：

```
//call display on each of this node's children
//note a node will only have children in its list if expanded
$num_children=sizeof($this->m_childlist);
for($i=0;$i<$num_children;$i++){
$row=$this->m_childlist[$i]->display($row,$sublist);
}
return$row;
```

在这里，又是一个递归函数调用，它将调用每一个节点的子节点来显示它们自身。将当前行颜色传递给它们，当完成显示操作后再将它传递回来，这样就可以保存交替颜色的记录。

以上就是这个类的所有功能。其代码相当复杂。可以通过运行该程序来认识这个类。当对它的功能感到满意时，再返回来研究这些代码。

31.6　查看单个的文章

display_tree()函数的调用将会给出一组文章的链接。如果我们点击其中的某篇文章，将进入view_post.php脚本程序，同时传递给该脚本这篇文章的postid。该脚本的示例输出如图31-7所示。



图　31-7　现在，我们可以看到该帖子的消息正文

view_post.php脚本如程序清单31-6所示，除了显示该消息的回复外，还可以显示该消息的正文。可看到这些回复也是以树形结构显示

出来的，不过这个树形结构已经被完全展开了，并且不带有加号或减号按钮。这是$sublist开关变量起作用的结果。

程序清单31-6　view_post.php——显示单个消息正文

```
<?php
//include function libraries
include('include_fns.php');
$postid=$_GET['postid'];
//get post details
$post=get_post($postid);
do_html_header($post['title']);
//display post
display_post($post);
//if post has any replies,show the tree view of them
if($post['children']){
echo"<br/><br/>";
display_replies_line();
display_tree($_SESSION['expanded'],0,$postid);
}
do_html_footer();
?>
```

该脚本主要调用3个函数来实现其功能：get_post()、display_post()和display_tree()函数。get_post()函数将一条消息细节从数据库中取出来。该函数的代码如程序清单31-7所示。

程序清单31-7　discussion_fns.php函数库中的get_post()函数——从数据库中取出一则消息

```
function get_post($postid){
//extract one post from the database and return as an array
if(!$postid){
return false;
}
```

```
$conn=db_connect();
//get all header information from'header'
$query="select*from header where postid='".$postid."'";
$result=$conn->query($query);
if($result->num_rows!=1){
return false;
}
$post=$result->fetch_assoc();
//get message from body and add it to the previous result
$query="select*from body where postid='".$postid."'";
$result2=$conn->query($query);
if($result2->num_rows>0){
$body=$result2->fetch_assoc();
if($body){
$post['message']=$body['message'];
}
}
return$post;
}
```

该函数根据一个给定的postid执行两个查询语句，为该文章取出消息的标题和正文，并将它们一起放在一个相关的数组中返回。

接下来，该函数返回结果将传递给output_fns.php函数库中的display_post()函数，该函数仅以某种HTML格式将数组打印出来，因此这里我们不再讨论。

最后，view_post.php脚本将检查该文章是否有回复并调用display_tree()函数以子列表的格式显示它们——即不带加号或减号并且完全展开的形式。

31.7　添加新文章

了解了以上所有操作后，我们现在可以来看看如何将一篇新文章加入到论坛中去。用户可以使用两种方法来完成该项操作：第一，通过在索引页面点击"New Post"按钮，第二，通过在view_post.php页面点击"Reply"按钮。

这些动作都将触发相同的脚本，new_post.php，只是带有不同的参数。图31-8显示通过点击"Reply"按钮而得到的new_post.php脚本的输出结果。

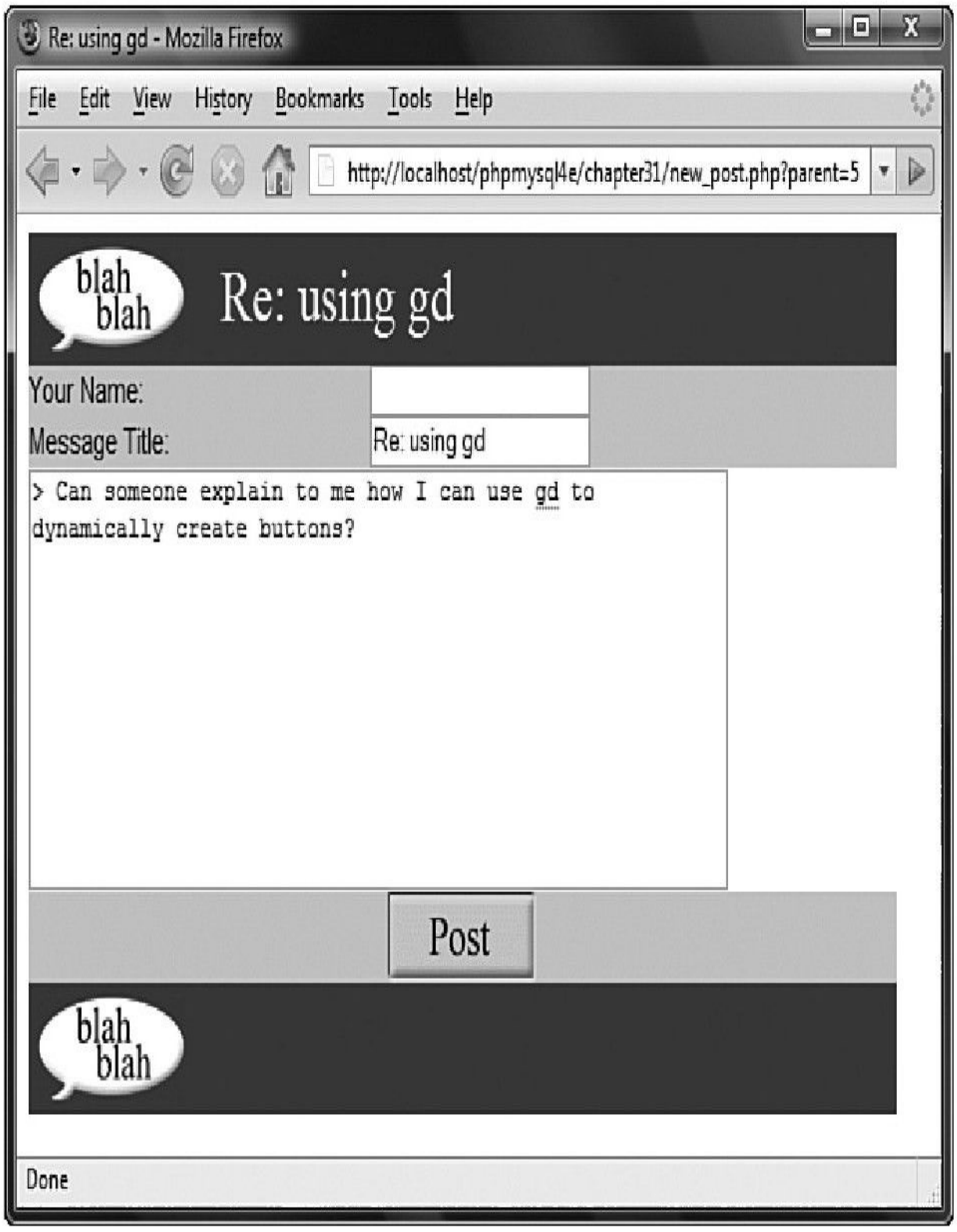


图　31-8　在回复文章中，父文章的文本将自动插入和标记

首先，让我们查看上图所示的URL：

```
http://localhost/phpmysql4e/chapter31/new_post.php?parent=5
```

传递给parent参数值应该是新文章父节点的postid。如果点击了"New Post"按钮而不是"Reply"按钮，将在URL中得到parent=0的字符串。

其次，可以看到对于一个回复，最初文章的文本被插入进来，而且前面都用“＞”字符标记了，这一点与大多数的邮件和新闻阅读程序一样。

第三，可以看到回复消息的默认标题为最初标题加上前缀"Re:"。下面，我们来看看产生该输出的代码，如程序清单31-8所示。

程序清单31-8　new_post.php——允许用户对一个已有帖子输入或创建一篇新文章

```
＜?php
include('include_fns.php');
$title=$_POST['title'];
$poster=$_POST['poster'];
$message=$_POST['message'];
if(isset($_GET['parent'])){
$parent=$_GET['parent'];
}else{
$parent=$_POST['parent'];
}
if(!$area){
$area=1;
}
if(!$error){
if(!$parent){
$parent=0;
if(!$title){
```

```
$title='New Post';
}
}else{
//get post name
$title=get_post_title($parent);
//append Re:
if(strstr($title,'Re:')==false){
$title='Re:'.$title;
}
//make sure title will still fit in db
$title=substr($title,0,20);
//prepend a quoting pattern to the post you are replying to
$message=add_quoting(get_post_message($parent));
}
}
do_html_header($title);
display_new_post_form($parent,$area,$title,$message,$poster);
if($error){
echo"<p>Your message was not stored.</p>
<p>Make sure you have filled in all fields and
try again.</p>";
}
do_html_footer();
?>
```

在完成一些初始化设置之后，该脚本将检查父节点是否为0。如果是0，则表示该文章是一个新主题，而接下来需要完成的操作就相对少些。

如果是一个回复（$parent变量是某篇已有文章的postid），以上脚本程序将继续执行并且设置标题和最初消息的文本，如下所示：

```
//get post name
$title=get_post_title($parent);
//append Re:
```

```
if(strstr($title,'Re:')==false){
$title='Re:'.$title;
}
//make sure title will still fit in db
$title=substr($title,0,20);
//prepend a quoting pattern to the post you are replying to
$message=add_quoting(get_post_message($parent));
```

这里用到的函数包括get_post_title()、get_post_message()和add_quoting()。这些函数都来自discussion_fns.php函数库。它们分别如程序清单31-9、程序清单31-10和程序清单31-11所示。

程序清单31-9　discussion_fns.php函数库中的get_post_title()函数——从数据库中得到某消息的标题

```
function get_post_title($postid){
//extract one post's name from the database
if(!$postid){
return'';
}
$conn=db_connect();
//get all header information from'header'
$query="select title from header where postid='".$postid."'";
$result=$conn->query($query);
if($result->num_rows!=1){
return'';
}
$this_row=$result->fetch_array();
return$this_row[0];
}
```

程序清单31-10　discussion_fns.php函数库中的get_post_message()函数——从数据库中获得某消息的正文

```
function get_post_message($postid){
//extract one post's message from the database
if(!$postid){
return'';
}
$conn=db_connect();
$query="select message from body where postid='".$postid."'";
$result=$conn->query($query);
if($result->num_rows>0){
$this_row=$result->fetch_array();
return$this_row[0];
}
}
```

这两个函数分别从数据库中获得特定文章的标题和正文。

程序清单31-11　discussion_fns.php函数库中的add_quoting()函数——用“>”符号缩进一则消息的正文

```
function add_quoting($string,$pattern='>'){
//add a quoting pattern to mark text quoted in your reply
return$pattern.str_replace("\n","\n$pattern",$string);
}
```

add_quoting()函数将重新格式化字符串，用一个符号开始原始文本的每一行，在默认情况下，该符号为“>”。

用户在回复文章中完成所需输入并且点击"Post"按钮后，将进入store_new_post.php脚本程序。图31-9所示的是该脚本的一个示例输出。

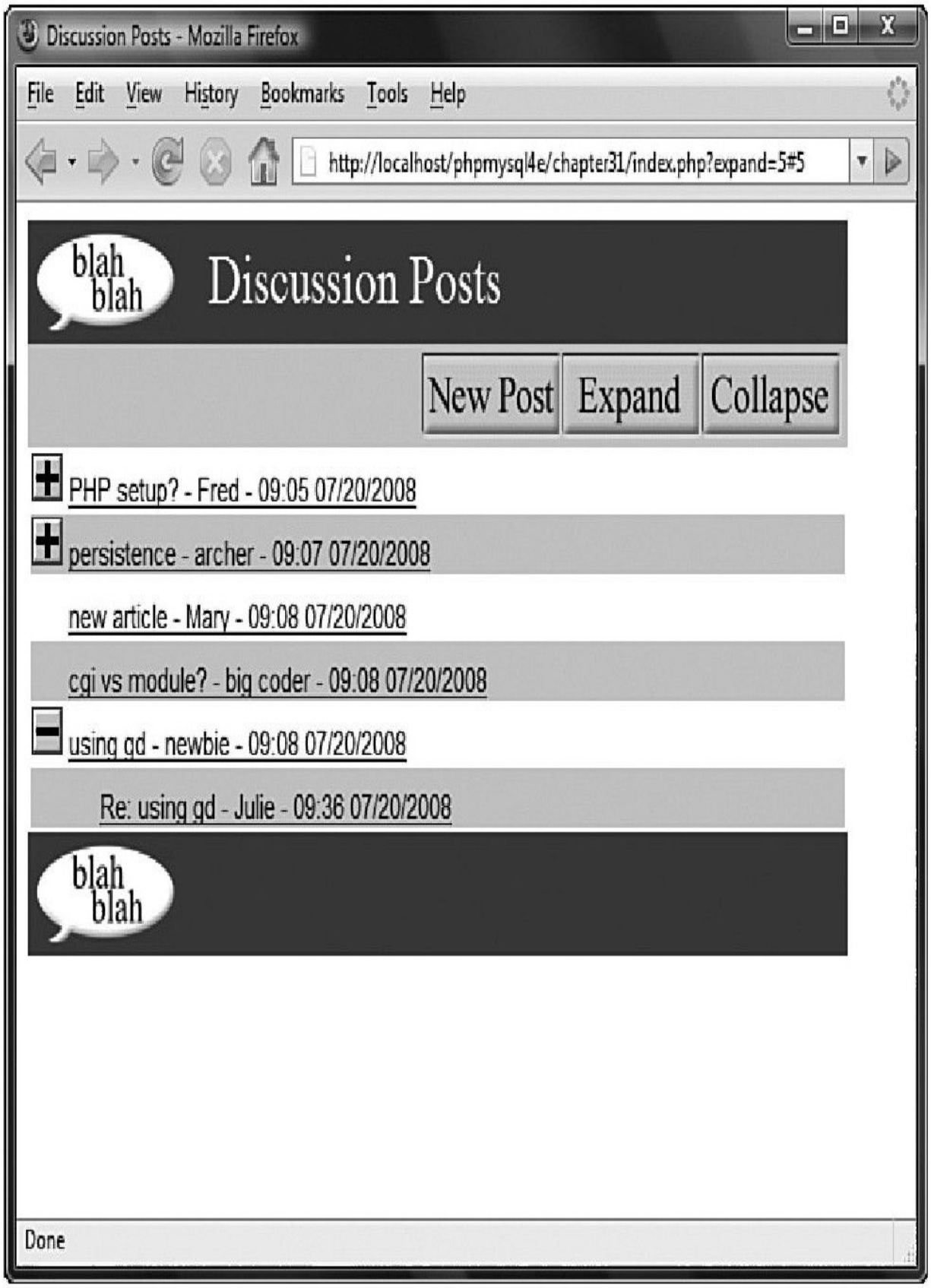

Discussion Posts - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/phpmysql4e/chapter31/index.php?expand=5#5
blah blah
Discussion Posts
New Post
Expand
Collapse
PHP setup? - Fred - 09:05 07/20/2008
persistence - archer - 09:07 07/20/2008
new article - Mary - 09:08 07/20/2008
cgi vs module? - big coder - 09:08 07/20/2008
using gd - newbie - 09:08 07/20/2008
Re: using gd - Julie - 09:36 07/20/2008
blah blah
Done

图　31-9　新发表的文章在树形结构中显示出来

在上图中，新发表的文章在这行下面Re：using gd?-Laura-Julie-09:36 07/20/2008。不仅如此，该页面看起来还像普通的index.php页面。

下面，我们来看看store_new_post.php的代码，如程序清单31-12所示。

程序清单31-12　store_new_post.php——在数据库中存入新的文章

```
<?php
include('include_fns.php');
if($id=store_new_post($_POST)){
include('index.php');
}else{
$error=true;
include('new_post.php');
}
?>
```

可以看到，这是一段很短的脚本程序。它的主要任务是调用store_new_post()函数。该页面没有它自身的可见内容。如果保存成功，我们会看到索引页面。否则，将回到new_post.php页，这样，用户可以重试。store_new_post()函数如程序清单31-13所示。

程序清单31-13　discussion_fns.php函数库中的store_new_post()函数——在数据库中保存并检验新发表的文章

```
function store_new_post($post){
//validate clean and store a new post
$conn=db_connect();
//check no fields are blank
if(!filled_out($post)){
return false;
```

```
}
$post=clean_all($post);
//check parent exists
if($post['parent']!=0){
$query="select postid from header where
postid='".$post['parent']."'";
$result=$conn->query($query);
if($result->num_rows!=1){
return false;
}
}
//check not a duplicate
$query="select header.postid from header,body where
header.postid=body.postid and
header.parent=".$post['parent']."and
header.poster='".$post['poster']."'and
header.title='".$post['title']."'and
header.area=".$post['area']."and
body.message='".$post['message']."'";
$result=$conn->query($query);
if(!$result){
return false;
}
if($result->num_rows>0){
$this_row=$result->fetch_array();
return$this_row[0];
}
$query="insert into header values
('".$post['parent']."',
'".$post['poster']."',
'".$post['title']."',
0,
'".$post['area']."',
now(),
NULL
)";
```

```
$result=$conn->query($query);
if(!$result){
return false;
}
//note that our parent now has a child
$query="update header set children=1 where postid='".$post['parent']."'";
$result=$conn->query($query);
if(!$result){
return false;
}
//find our post id,note that there could be multiple headers
//that are the same except for id and probably posted time
$query="select header.postid from header left join body
on header.postid=body.postid
where parent='".$post['parent']."'
and poster='".$post['poster']."'
and title='".$post['title']."'
and body.postid is NULL";
$result=$conn->query($query);
if(!$result){
return false;
}
if($result->num_rows>0){
$this_row=$result->fetch_array();
$id=$this_row[0];
}
if($id){
$query="insert into body values
($id,'".$post['message']."')";
$result=$conn->query($query);
if(!$result){
return false;
}
return$id;
}
}
```

这是个很长的函数，但它并不太复杂。它之所以长，是因为插入一篇文章就意味着要标题表和正文表进行插入操作，并要在标题表中更新父文章行以表明它现在有子文章了。

以上就是该Web论坛应用程序的所有代码。

31.8　添加扩充

可以在该项目中增加如下所示的几项扩充功能：

■可以在视图选项中增加导航功能，这样就可以从一篇文章跳到下一篇、前一篇、下一个话题的文章，或前一个话题的文章。

■可以增加一个界面来建立新的论坛和删除旧的文章。

■可以增加用户验证以确保只有注册用户可以发表文章。

■可以增加某些限制和审查机制。

学习一些已有的系统，将从中受到一些启发。

31.9 使用一个已有的系统

Phorum是一个值得注意的已有系统，它是开放源代码的Web论坛项目。它与我们的系统有不同的导航和语义，但是它的结构相对来说易于定制，从而适合我们的站点要求。Phorum的一个显著特性是它能由用户自主确定配置，从而决定是以集中视图还是普通视图来显示文章。可以在下面的站点找到更多关于它的内容：http://www.phorum.org。

31.10 下一章

在第32章中，我们将用PDF格式来发布极具吸引力的、排列一致的、并且可以在某种程度上防止破坏的文档。这对于基于服务范畴的应用程序来说是相当有用的，比如生成在线合同。

第32章　生成PDF格式的个性化文档

在以服务为驱动的站点中，我们有时候需要发布个性化文档，这些文档是根据访问者的输入而生成的。这个特性可以用来实现表单的自动填写或用来生成个性化文档，例如法律文档、信件或证书等。

在本章中，我们所介绍的例子将为用户提供一个在线技术评估页面并生成一个证书。

在本章中，我们主要介绍以下内容：

■如何使用PHP字符串处理功能将用户数据与模板结合起来，从而创建一个RTF格式的文档。

■如何使用类似的方法生成PDF格式的文档。

■如何使用PHP的PDFlib函数来生成类似的PDF文档。

32.1　项目概述

对于这个项目，我们希望能为访问者提供一个包括许多问题的测验。如果他们正确回答了足够多的问题，则给他们一个证书证明通过了该测验。

因为计算机能够比较容易地对问题进行打分，所以问题采用多项选择形式，它由一个问题和多个可能的答案组成。每个问题仅仅有一个答案是正确的。

如果用户在这些问题上获得了及格的分数，将获得一个证书。

在理想状况下，我们所颁布的证书文件格式应该是：

■容易设计。

■能够包含多种不同的元素，例如位图和向量图。

■能产生一个高质量的打印输出。

■只需要下载一个小文件。

■几乎立即就能生成结果。

■成本低。

■可以在多种操作系统上工作。

■难以伪造及更改。

■不需要任何特殊的软件来浏览或打印。

■对所有接收者都能够一致地显示和打印。

像我们有时候做很多决定一样，当选择一种传送格式来满足以上特性时，需要进行一些折中。

评估文档格式

需要做出的最重要决定是用什么格式来传送证书。可选项包括纸张、ASCII文本、HTML、Microsoft Word或其他字处理格式，诸如：RTF、PostScript和PDF等。根据上面列出的特性，我们可以考虑并比较其中的一些选项。

1.纸张

用纸张传送证书具有许多明显的优点。我们可以完全控制处理过程。在将证书发送出去之前，可以很清楚地看到输出证书的外观，从而不必担心软件或带宽，并且证书可以使用反伪造技术进行印制。

除了“几乎立即就能生成结果”和“成本低”项要求外，它可以适应其他所有要求。但是纸质证书不能快速制造并传送。根据我们及接收者邮政投递系统的不同，这可能需要花费几天或几周的时间。

每一张证书都要花费几分到几美元的印刷和邮寄费用，而在实际处理中可能需要更多费用。相比较，自动电子投递可能要便宜些。

2.ASCII

以ASCII或纯文本格式来投递文档具有一些优势。首先，兼容性就不会成为问题。带宽要求也会相当小，因此费用会相当低。最终结果的简单性使得设计起来非常简单，而且可以使用脚本快速生成。

但是，如果以ASCII文件的形式向访问者提供证书，那么对证书的外观几乎没有任何控制。我们不能控制字体或者分页。我们仅能包含文本，而几乎对格式没有任何控制。我们也不能控制接收者对文档的复制和更改。这种方法使得接收者易于欺骗性地更改证书。

3.超文本置标语言（HTML）

在Web上传送文档，HTML是一个显而易见的选择。超文本置标语言就是专门为该目的而设计的。毫无疑问，HTML包括格式控制，支持包括多种对象（比如图像）的语法，并且与多种操作系统和软件是兼容的（包含一些变化）。它非常简单，因此既易于设计又可以快速地由脚本程序生成和传送。

在本程序中使用HTML的缺陷包括：对打印相关格式的有限支持，比如分页、在不同的平台及程序上的输出不一致性，以及不稳定的打印质量。此外，虽然HTML可以包含任何外部元素的类型，但是对于不常见的类型，浏览器显示或使用这些元素的能力不能得到保证。

4.字处理器格式

对于企业内部网的项目，提供字处理器格式的文档是尤其有意义的。但是，对于Internet项目，使用专有的字处理器格式会排斥某些访问者，但由于其市场优势，对Microsoft Word来说这样做可能有意义。多数用户会用Word或者某种字处理器（例如，OpenOffice Writer）来阅读Word文件。

没有Word的Windows用户可以从如下网址下载免费软件WordViewer：http://office.microsoft.com/en-us/downloads/ha010449811033.aspx。

生成像Microsoft Word文档一样的文档有几点好处。只要有一份Word的副本，设计一个文档就变得相当容易。我们可以很好地控制文档的打印效果，文档内容也极具灵活性。当然，也可以通过要求密码的方式来使得接收者难以修改其内容。

不幸的是，Word文件可能会很大，特别是当它们包含图像或其他复杂元素时。而且没有很好的办法使用PHP动态生成这种文档。这种格式是经过文档化的，但是它是一种二进制格式，而且这种格式的文档化要求得到许可条件。使用COM对象生成Word文档是可能的，但是这相当复杂。

可以考虑的另一个可能方法是使用OpenOffice Writer软件，这个软件具有非专利软件和使用基于XML的文件格式的双重优点。

如今，Office 2003和2007也支持自身定义的XML文件格式。Word和其他Office产品的文档类型定义（DTD）可以从Microsoft.com站点下载。请查阅“Office XML参考模式”（Office XML Reference Schemas）。这是一个不错的选择，但不是最简单的。

5.RTF格式

RTF（Rich Text Format）具有Word的大多数优点，但是RTF文件更容易生成。在打印页的排列和格式方面仍然具有灵活性，仍然可以包含诸如向量图或位图之类的元素。此外还可以确保当用户浏览或打印文档时，他们看到的结果与我们看到的会十分相似。

RTF是Microsoft Word的文本格式。它是为一种在不同程序间传送文档所使用的交互格式而设计的。在某种程度上，它与HTML类似。它使用语法和关键字而不用二进制数据来表达格式化信息，因此相对更容易阅读一些。

这种格式的文档非常丰富和翔实，其规范可以免费获得，用关键字"RTF Specification"搜索Microsoft.com就可找到。

生成一个RTF文档的最简单的方法是在字处理器中选择Save As RTF选项。由于RTF文件仅包含文本，因此直接生成和修改已有的文件也十分简单。

因为这种格式的文档非常丰富而且可以免费使用，与Word的二进制格式比起来，RTF可以被更多的软件读取。请注意，用户如果使用旧版本的Word或其他字处理器软件打开一个复杂的RTF文件，经常会产生不同的显示结果。每种新版本的Word都为RTF引进新的关键字，因此旧版本经常忽略它们无法解释的控件或那些已经没有选择实现的控件。

从我们最初的要求列表来看，RTF证书易于使用Word或其他字处理器设计；可以包含多种不同的元素，如向量图和位图；具有高质量的打印输出；可以简单且快速地生成；并且以低廉的费用通过电子化方式传送。

它可以在多种应用程序及操作系统上使用，尽管结果略微有些变化。不利的一面是：RTF文档可以轻易地被任何人随时随意更改，这对于证书和某些其他文档来说是一个问题。对于复杂文档，其文件可能稍微大一些。

RTF对于许多文档分发程序来说是一个不错的选择，因此我们这里把它作为一个可选项。

6.PostScript

Adobe公司的PostScript是一种页面描述语言。它是一种强大且复杂的编程语言，试图以与设备无关的方式表示文档，也就是说，一段描述可以在不同的设备，比如打印机、显示器上产生一致的输出结果。该格式的文档也非常丰富。除了很多Web站点外，至少有3本图书全面地介绍了这种格式。

PostScript文档可以包含非常精确的格式、文本、图像、嵌入式字体和其他元素。可以通过在应用程序中将PostScript文档打印到PostScript打印机驱动程序，很容易地生成一个PostScript文档。如果对这些感兴趣，可以学习用它直接编辑。

PostScript文档具有很好的可移植性。在不同的设备及操作系统下，它们都可以给出一致的高质量的打印输出。

用PostScript发布文档有两个重大缺陷：文件大小可能很大以及许多用户必须下载额外的软件来使用它。

大多数UNIX用户可以处理PostScript文件，但Windows用户通常需要下载一个浏览器，比如GSview，该浏览器使用Ghostscript PostScript解释器。该软件适合多种类型的平台。虽然它是自由软件，但我们不希望强迫用户下载更多的软件。

可以在如下网址获得更多关于Ghostscript的文章：http://www.ghostscript.com/。

也可以在如下网址下载：http://www.cs.wisc.edu/～ghost/。

对于我们当前的程序，PostScript在具有一致性和高质量的输出方面表现十分出色，而在其他要求方面却表现得相当糟糕。

7.可移植文档格式（PDF）

可喜的是，有一种格式既具有PostScript的大多数功能，又具有很多其他优点。PDF（Portable Document Format，也出自Adobe公司）是出于以下目的来设计文档发布方法的：这种文档在不同的平台表现一致，且可在屏幕或纸上产生可预见的高质量输出。

Adobe公司将PDF称为“面向世界范围开放的电子文档发布的事实标准”。Adobe的PDF是一种统一文件格式，它预留了所有字体、格式、颜色和任何源文档的图形，不论它是在何种应用程序和平台上创建的。PDF文件是简洁的，拥有免费Adobe Acrobat阅读器的人可以随心所欲地共享、浏览、操作和打印。

PDF是一种开放格式，除了其他许多Web站点和正式的手册外，也可以在如下网址找到相关文档：http://partners.adobe.com/asn/tech/pdf/specifications.jsp。

从我们要求的特性来看，PDF非常令人满意：PDF可以产生一致的、高质量的输出，可以包含诸如位图或向量图之类的元素，可以使用压缩方法创建较小的文件，可以廉价地通过电子方式进行传送，可以在多数操作系统上使用，而且还包括安全控制。

依靠PDF来工作是基于这样一个事实，那就是大多数用来创建PDF文档的软件都是商业性的。浏览PDF文件需要一个阅读器，但Adobe公司发布的Acrobat Reader的Windows、UNIX及Macintosh版本都是可以免费使用的。我们站点的许多访问者应该已经相当熟悉.pdf的扩展名，而且他们可能已经安装了阅读器。

PDF文件是一种易于发布的、可打印文档的好方法，特别是对那些不希望接收者轻易更改的情况。

接下来，我们将要通过两种不同的方法来生成PDF证书。

32.2 解决方案的组成

为了使系统能够工作，我们需要测试用户已经掌握的知识（假设他们通过了测试），并生成一个证书来报告成绩。我们通过3种方法来完成证书的生成：使用RFT模板，使用PDF模板以及通过程序创建新的PDF。

下面，我们仔细了解各组成部分的要求。

32.2.1 问题与回答系统

提供一个灵活的在线评估系统，用以支持不同的问题类型，支持信息的不同媒体类型，对错误回答给出有效反馈，以及支持智能的统计信息搜集和报告，这本身就是一个非常复杂的任务。

在本章中，我们主要兴趣在于生成可以通过Web发布的用户自定义文档，因此只创建一个非常简单的测试系统。这个测试系统不依赖于任何特殊的软件，它使用HTML表单来提出问题，使用PHP脚本来处理用户的回答。事实上，我们从第1章就开始了PHP脚本的编写。

32.2.2 文档生成软件

在Web服务器上，我们不需要额外的软件就可以从模板中生成RTF或PDF格式的文档。但是需要使用某些软件来建立模板。为了使用PHP PDF创建函数，必须将PDF支持编译到PHP中。（我们将在稍后的内容详细讨论。）

1.创建RTF模板的软件

可以使用我们选择的字处理器软件来生成RTF文件。我们使用Microsoft Word来创建证书模板。证书模板包含在附带文件的第32章的目录下。

如果希望使用其他字处理软件，最好是先在Word下测试其输出效果，因为这是网站访问者最经常使用的软件。

2.创建PDF模板的软件

PDF文档生成起来稍微难些。最简单的方法就是购买Adobe Acrobat软件。该软件允许我们从各种应用程序中创建高质量的PDF。对于该项目，我们将使用Acrobat来创建模板文件。

为了创建该文件，可以使用Microsoft Word来设计一个文档。Acrobat包中的一个工具是Adobe Distiller。在Distiller中，需要选择一些非默认的选项。该文件必须以ASCII格式保存，而压缩选项必须关闭。设置完这些选项后，创建一个PDF文件就同打印一样简单。

可以在如下网址找到更多关于Acrobat的内容：http://www.adobe.com/products/acrobat/。

我们可以通过在线方式或者从普通软件零售商那里购买该软件。

创建PDF的另一个方法是使用转换程序ps2pdf，正如其名所示，该程序将PostScript文件转换成PDF文件。该程序是免费的，但是对于那些包含图像或非标准字体的文档并不总是具有好的输出。ps2pdf转换器包是前面提到的Ghostscript包附带的。

很明显，如果想通过这种方法创建PDF文件，必须先创建一个PostScript文件。通常，UNIX用户会使用a2ps或dvips工具来创建PostScript文件。

如果在Windows环境下工作，也可以在不使用Adobe Distiller的情况下创建PostScript文件，虽然创建过程可能稍复杂一些。需要安装一个PostScript打印机驱动程序。例如，可以使用Apple LaserWriter IINT驱动程序。如果没有安装PostScript驱动程序，可以从Adobe公司的网址下载一个：http://www.adobe.com/support/downloads/product.jsp?product=44＆platform=Windows。

要创建PostScript文件，必须选择该打印机和“打印到文件”选项，通常，可以在打印对话框里找到该选项。

大多数Windows应用程序将产生一个.prn扩展名的文件。这应该是一个PostScript文件。需要将它重命名为.ps文件。使用GSview或其他的PostScript浏览器，可以浏览它，或者可以通过ps2pdf工具创建一个PDF文件。

请注意，不同的打印机驱动程序将产生不同质量的PostScript输出。我们可能会发现生成的某些PostScript文件在ps2pdf工具中运行时会产生错误。遇到这种情况，我们建议使用不同的打印机驱动程序。

如果只想创建少量的PDF文件，Adobe在线服务能够满足要求。每个月只需要9.99美元，可以上传多种格式的文件并下载转换后的PDF文件。这个服务很适合我们的证书，但是，它却不允许我们选择某些对项目很重要的选项。生成的PDF将作为一个二进制文件保存并要经过压缩。这使得修改它变得非常困难。

该服务可以在如下网址找到：https://createpdf.adobe.com/。

如果希望试用这个服务，也可以找到该服务的免费试用选项。此外，如果你要创建的PDF文件数量少于5个，你也可以Adobe的免费服务：http://www.acrobat.com。

在Net Distillery网站上，还有一个基于FTP的ps2pdf下载界面：http://www.babinszki.com/distiller/。

最后一个选项就是以XML格式对证书进行编码，然后使用XML风格样式单转换器（XSLT）将其转换成PDF格式和任何其他所需的格式。这种方法需要对XSLT的很好理解，在这里不做介绍。

3.用软件通过编程方式创建PDF

PHP本身就提供了对创建PDF文档的支持。PHP的PDFlib函数使用PDFlib函数库。该函数库可以从http://www.pdflib.com/products/pdflibfamily/下载。PDFlib提供了生成PDF文档所需的API。

PDFlib并不是免费的；它要求商业许可。PDFlib Lite是免费的，而且是开放源代码的，但是只有在特定条件下，例如用于非商业用途。

此外，还有一些免费的函数库，例如FPDF。FPDF使用PHP语言编写（而不是用C语言编写的PHP扩展），它的性能要慢于其他两个库。你可以从http://www.fpdf.org/下载FPDF。

在本章中，我们将使用PDFlib，因为它可能是使用的最为广泛的创建PDF的扩展。

检查phpinfo()函数的输出，你可以检查系统是否已经安装PDFlib。在PDF标题部分，你可以找到PDFlib支持是否已经启用以及所使用的PDFlib版本。

如果要在PDF文档使用TIFF或JPEG图像，你还需要安装TIFF函数库（可以从http://www.libtiff.org/下载），以及JPEG函数库（可以从ftp://ftp.uu.net/graphics/jpeg/下载）。

PDFlib扩展并没有内置在PHP中；你必须从PECL（PHP扩展组织函数库）获得这些文件并且手动安装这些扩展。

在非Windows系统中，你可以http://pecl.php.net/package/pdflib处下载这些扩展文件并使用pecl命令安装。请参阅http://www.php.net/manual/en/install.pecl.pear.php处的说明。

在Windows系统中，从://pecl4win.php.net/ext.php/php_pdflib.dll下载预编译扩展（php_pdf.dll）或者从PHP.net下载页面下载完整的已编译PECL扩展。下载后，将php_pdflib.dll文件保存在PHP扩展目录（通常是PHP安装路径下的ext目录），并且在php.ini文件添加如下代码：extension=php_pdf.dll。

32.3 解决方案概述

在这个项目中，我们将设计一个具有3种可能输出的系统。如图32-1所示，我们将提出测验问题，评估用户的回答，然后用以下3种方法之一来生成证书：

- 用一个空白模板生成RTF文档。
- 用一个空白模板生成PDF文档。
- 通过PDFlib库用程序生成PDF文档。

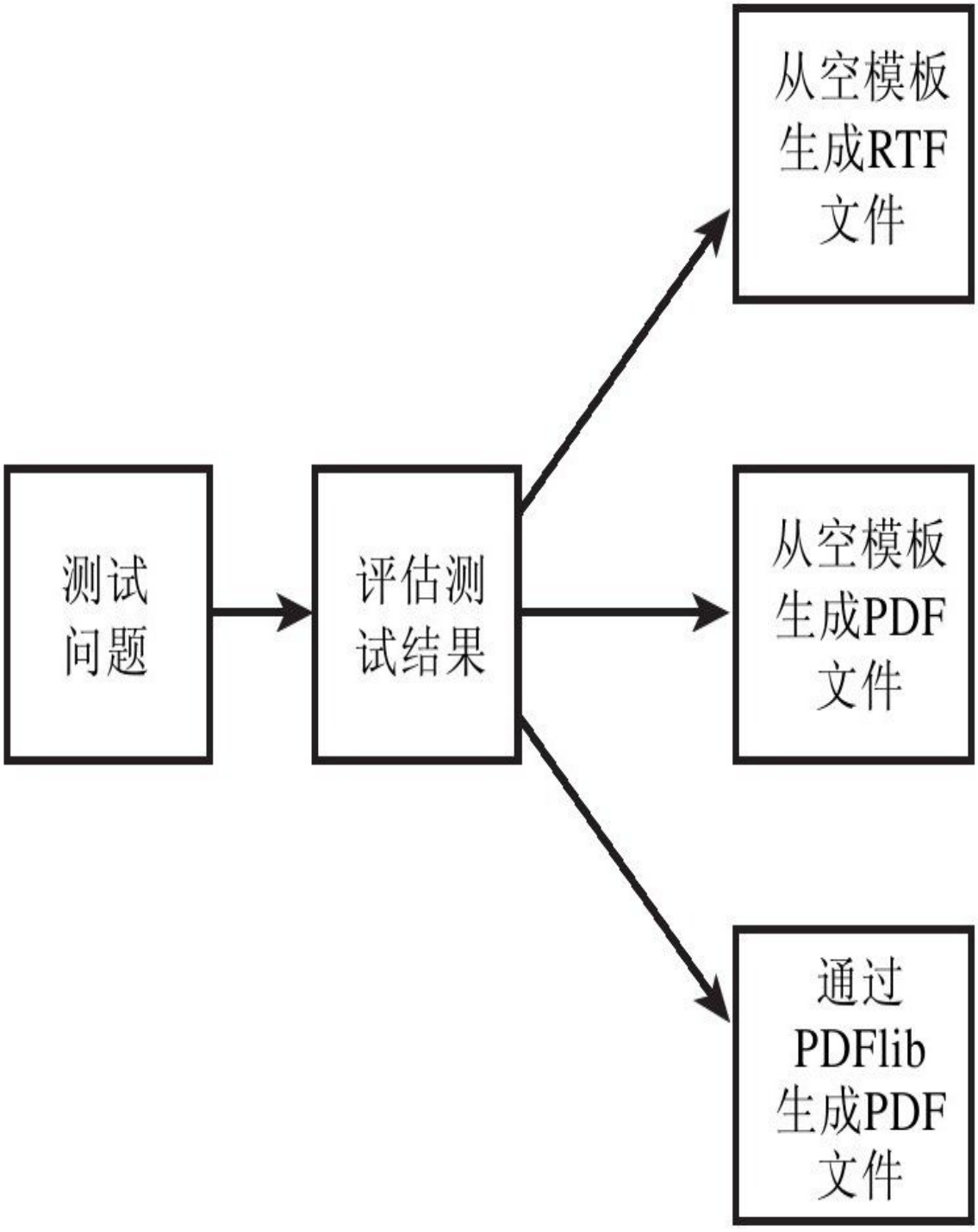


图　32-1　证书系统将生成3种不同证书中的一种

表32-1给出了该证书项目将用到的所有文件。

表32-1 证书应用程序将用到的所有文件

文件名	类 型	描 述
index.html	HTML页	包括测验题目的HTML表单
score.php	应用程序	评估用户回答的脚本程序
rtf.php	应用程序	用模板生成RTF的脚本程序
pdf.php	应用程序	用模板生成PDF的脚本程序
pdflib.php	应用程序	用PDFlib库生成PDF证书的脚本程序
signature.png	图像	包含在PDFlib证书中的签名的位图图像
PHPCertification.rtf	RTF	RTF证书模板
PHPCertification.pdf	PDF	PDF证书模板

下面，我们继续学习该程序。

32.3.1 提问

index.html文件非常直观。它必须包含一个HTML表单来询问用户姓名及一些问题的答案。在一个真正的评估应用程序中，很可能是从数据库中取出这些问题。在这里，我们将集中讨论产生证书的问题，因此只随意编造几个题目“硬编码”在HTML中。

姓名域是一个文本输入框。每个题目有3个单选按钮以供用户选择答案。该表单还有一个带有图像的提交按钮。

该页的代码如程序清单32-1所示。

程序清单32-1　index.html——包含测验问题的HTML页

```
<html>
<body>
<h1><p align="center">
<img src="rosette.gif"alt="">
Certification
<img src="rosette.gif"alt=""></p></h1>
<p>You too can earn your highly respected PHP certification
from the world famous Fictional Institute of PHP Certification.</p>
<p>Simply answer the questions below:</p>
<form action="score.php"method="post">
<p>Your Name<input type="text"name="name"/></p>
<p>What does the PHP statement echo do?</p>
<ol>
<li><input type="radio"name="q1"value="1"/>
Outputs strings.</li>
<li><input type="radio"name="q1"value="2"/>
Adds two numbers together.</li>
<li><input type="radio"name="q1"value="3"/>
Creates a magical elf to finish writing your code.</li>
</ol>
<p>What does the PHP function cos()do?</p>
<ol>
<li><input type="radio"name="q2"value="1"/>
Calculates a cosine in radians.</li>
<li><input type="radio"name="q2"value="2"/>
Calculates a tangent in radians.</li>
<li><input type="radio"name="q2"value="3"/>
It is not a PHP function.It is a lettuce.</li>
</ol>
```

```
<p>What does the PHP function mail()do?</p>
<ol>
<li><input type="radio"name="q3"value="1"/>
Sends a mail message.
<li><input type="radio"name="q3"value="2"/>
Checks for new mail.
<li><input type="radio"name="q3"value="3"/>
Toggles PHP between male and female mode.
</ol>
<p align="center"><input type="image"src="certify-me.gif"border="0"></p>
</form>
</body>
</html>
```

如图32-2所示的是在浏览器中打开index.html的结果。

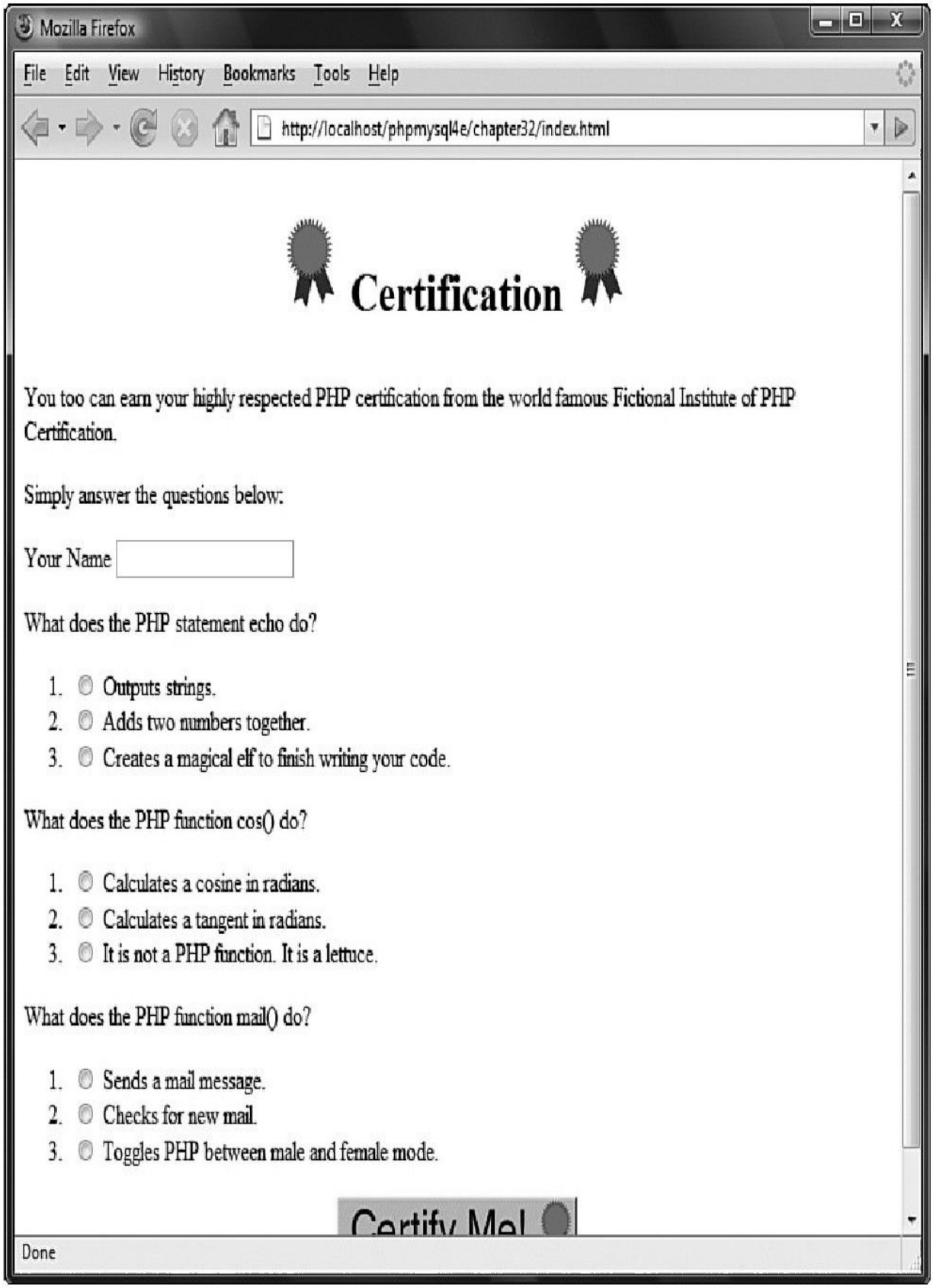

Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/phpmysql4e/chapter32/index.html
Certification
You too can earn your highly respected PHP certification from the world famous Fictional Institute of PHP Certification.
Simply answer the questions below:
Your Name
What does the PHP statement echo do?
1. Outputs strings.
2. Adds two numbers together.
3. Creates a magical elf to finish writing your code.
What does the PHP function cos() do?
1. Calculates a cosine in radians.
2. Calculates a tangent in radians.
3. It is not a PHP function. It is a lettuce.
What does the PHP function mail() do?
1. Sends a mail message.
2. Checks for new mail.
3. Toggles PHP between male and female mode.
Certify Me!
Done

图　32-2　index.html要求用户回答测试问题

32.3.2　给答题评分

当用户在index.html中提交了题目的答案后，我们需要对答案进行评估，并且计算得分。这项工作是由score.php脚本程序完成的。该脚本代码如程序清单32-2所示。

程序清单32-2　score.php——评阅测验的脚本

```
<?php
//create short variable names
$q1=$_POST['q1'];
$q2=$_POST['q2'];
$q3=$_POST['q3'];
$name=$_POST['name'];
//check that all the data was received
if(($q1=='')||($q2=='')||($q3=='')||($name=='')){
echo"<h1>
<p align=\"center\">
<img src=\"rosette.gif\"alt=\"\"/>
Sorry:
<img src=\"rosette.gif\"alt=\"\"/></p></h1>
<p>You need to fill in your name and answer all questions.</p>";
}else{
//add up the scores
$score=0;
if($q1==1){
//the correct answer for q1 is 1
$score++;
}
if($q2==1){
//the correct answer for q2 is 1
$score++;
}
if($q3==1){
//the correct answer for q3 is 1
```

```
$score++;
}
//convert score to a percentage
$score=$score/3*100;
if($score<50){
//this person failed
echo"<h1>
<p align=\"center\">
<img src=\"rosette.gif\"alt=\"\"/>
Sorry:
<img src=\"rosette.gif\"alt=\"\"/></p></h1>
<p>You need to score at least 50%to pass the exam.</p>";
}else{
//create a string containing the score to one decimal place
$score=number_format($score,1);
echo"<h1 align=\"center\">
<img src=\"rosette.gif\"alt=\"\"/>
Congratulations!
<img src=\"rosette.gif\"alt=\"\"/></h1>
<p>Well done".$name.",with a score of".$score."%,
you have passed the exam.</p>";
//provide links to scripts that generate the certificates
echo"<p>Please click here to download your certificate as
a Microsoft Word(RTF)file.</p>
<form action=\"rtf.php\"method=\"post\">
<div align=\"center\">
<input type=\"image\"src=\"certificate.gif\"border=\"0\">
</div>
<input type=\"hidden\"name=\"score\"value=\"".$score."\"/>
<input type=\"hidden\"name=\"name\"value=\"".$name."\"/>
</form>
<p>Please click here to download your certificate as
a Portable Document Format(PDF)file.</p>
<form action=\"pdf.php\"method=\"post\">
<div align=\"center\">
<input type=\"image\"src=\"certificate.gif\"border=\"0\">
```

```
</div>
<input type=\"hidden\"name=\"score\"value=\"".$score."\"/>
<input type=\"hidden\"name=\"name\"value=\"".$name."\"/>
</form>
<p>Please click here to download your certificate as
a Portable Document Format(PDF)file generated with PDFLib.</p>
<form action=\"pdflib.php\"method=\"post\">
<div align=\"center\">
<input type=\"image\"src=\"certificate.gif\"border=\"0\">
</div>
<input type=\"hidden\"name=\"score\"value=\"".$score."\"/>
<input type=\"hidden\"name=\"name\"value=\"".$name."\"/>
</form>";
}
}
?>
```

如果用户没有回答完所有问题，或者得分低于规定的及格线，该脚本显示一条消息。

如果用户成功地回答了问题，可以获得一个证书。一个成功测试的输出如图32-3所示。

Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost/phpmysql4e/chapter32/score.php

Congratulations!

Well done Jane Doe, with a score of 100.0%, you have passed the exam.

Please click here to download your certificate as a Microsoft Word (RTF) file.



Please click here to download your certificate as a Portable Document Format (PDF) file.



Please click here to download your certificate as a Portable Document Format (PDF) file generated with PDFLib.



Done

图　32-3　score.php向成功回答问题的用户展示3种可供选择证书的选项

在这里，用户有3个选项。可以得到一个RTF证书，或者两个PDF证书之一。下面，我们分别介绍相应的脚本。

32.3.3 生成RTF证书

我们可以通过将ASCII文本写入文件或字符串变量的办法来生成一个RTF文档，但这意味着我们要学习另一套语法。

如下所示的是一个非常简单的RTF文档示例：

```
{\rtf1
{\fonttbl{\f0 Arial;}{\f1 Times New Roman;}}
\f0\fs28 Heading\par
\f1\fs20 This is an rtf document.\par
}
```

该文档设置一个包含两种字体的字体表：Arial，对应于f0，Times New Roman，对应于fl。然后以28号字（14 point），f0（Arial）字体书写"Heading"。\par控制符表示一个段落分隔符。接着，用20号字（10 point）的f1（Times New Roman）字体书写"This is an rtf document"。

我们可以像这样手动生成一个RTF文档，但在PHP中，并没有任何简便的函数来简化这些困难的工作，比如，合并图形。幸运的是，在许多文档中，结构、类型和大部分文本都是静态的，仅有一小部分存在个体差异。使用模板是生成文档的一种更高效的方法。

我们可以使用字处理器方便地建立一个复杂的文档，如图32-4所示。

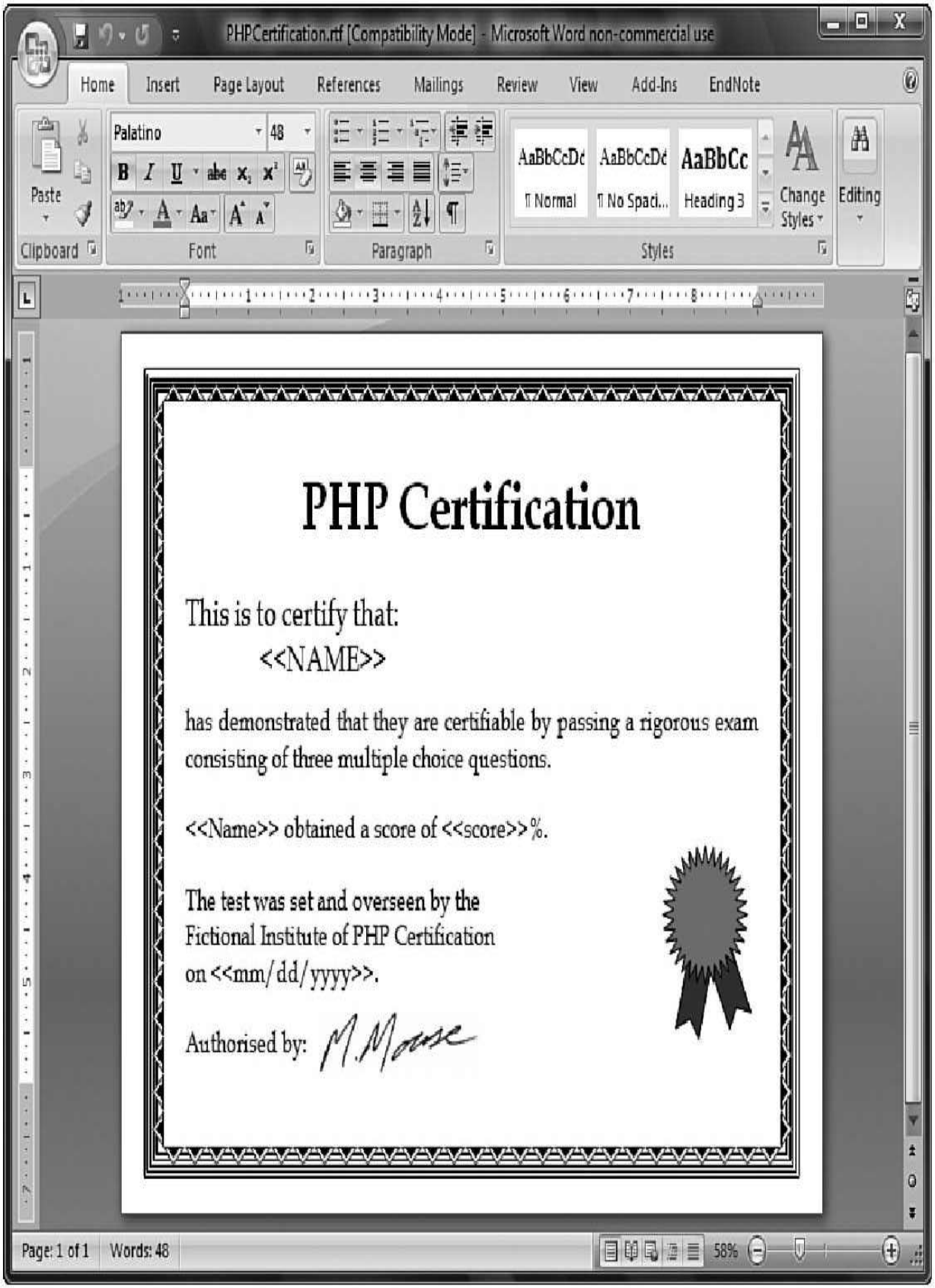

PHPCertification.rtf [Compatibility Mode] - Microsoft Word non-commercial use
Home
Insert
Page Layout
References
Mailings
Review
View
Add-Ins
EndNote
Palatino
48
Paste
Clipboard
Font
Paragraph
Styles
Change Styles
Editing
PHP Certification
This is to certify that:
<<NAME>>
has demonstrated that they are certifiable by passing a rigorous exam consisting of three multiple choice questions.
<<Name>> obtained a score of <<score>>%.
The test was set and overseen by the
Fictional Institute of PHP Certification
on <<mm/dd/yyyy>>.
Authorised by: M. Mouse
Page: 1 of 1
Words: 48
58%

图　32-4　使用字处理器，可以容易地创建一个复杂而又有用的模板

模板包括占位符，如＜＜NAME＞＞，用来标记该处将插入动态数据。这些占位符是什么样子并不重要。我们在两组尖括号之间使用一个有意义的字符串来描述它。选择一个不可能在文档其余部分显示的占位符是相当重要的。当占位符与替代它们的数据的长度大致相同时，它有助于排列模板格式。

这个文档中的占位符包括＜＜NAME＞＞、＜＜Name＞＞、＜＜score＞＞和＜＜mm/dd/yyyy＞＞。请注意，我们既使用了NAME又使用了Name，因为我们想用区分大小写的方法来替换它们。

现在我们已经有了模板，需要一个脚本来让它个性化。该脚本是rtf.php，代码如程序清单32-3所示。

程序清单32-3　rtf.php——产生一个个性化RTF证书的脚本

```
＜?php
//create short variable names
$name=$_POST['name'];
$score=$_POST['score'];
//check we have the parameters we need
if(!$name||!$score){
echo"＜h1＞Error:＜/h1＞
＜p＞This page was called incorrectly＜/p＞";
}else{
//generate the headers to help a browser choose the correct application
header('Content-type:application/msword');
header('Content-Disposition:inline,filename=cert.rtf');
$date=date('F d,Y');
//open our template file
$filename='PHPCertification.rtf';
$fp=fopen($filename,'r');
//read our template into a variable
$output=fread($fp,filesize($filename));
fclose($fp);
```

```
//replace the place holders in the template with our data
$output=str_replace('<<NAME>>',strtoupper($name),$output);
$output=str_replace('<<Name>>',$name,$output);
$output=str_replace('<<score>>',$score,$output);
$output=str_replace('<<mm/dd/yyyy>>',$date,$output);
//send the generated document to the browser
echo$output;
}
?>
```

该脚本将执行一些基本的错误检查，从而确保已提交了用户的所有资料，然后转而进行生成证书的工作。

该脚本的输出是一个RTF文件而不是HTML文件，因此我们必须提醒用户浏览器知道这个事实。这个步骤很重要，这样浏览器才能以正确的应用程序打开这个文件，或者在不能辨认.rtf扩展名的情况下能给出一个Save As...类型的对话框。

使用PHP的header()函数，可以指定将要输出文件的MIME类型以发送相应的HTTP页头，如下代码所示：

```
header('Content-type:application/msword');
header('Content-Disposition:inline,filename=cert.rtf');
```

第一个页头通知浏览器正在发送一个Microsoft Word文件（其实并不完全正确，但是它是最有可能打开该RTF文件的应用程序）。

第二个页头通知浏览器自动显示文件内容，并告知推荐文件名为cert.rtf。如果用户想在浏览器中保存该文件，这个名字是用户看到的默认文件名称。

在发送页头之后，我们打开并将该模板RTF文件读入到$output变量中，通过str_replace()函数，用我们希望显示的实际数据替换占位符。该行代码如下所示：

```
$output=str_replace('<<Name>>',$name,$output);
```

用变量$name的内容替换任何出现占位符 < < Name > > 的地方。

替换完成之后，接下来就只需将输出回送到浏览器中。该脚本的示例输出结果如图32-5所示。

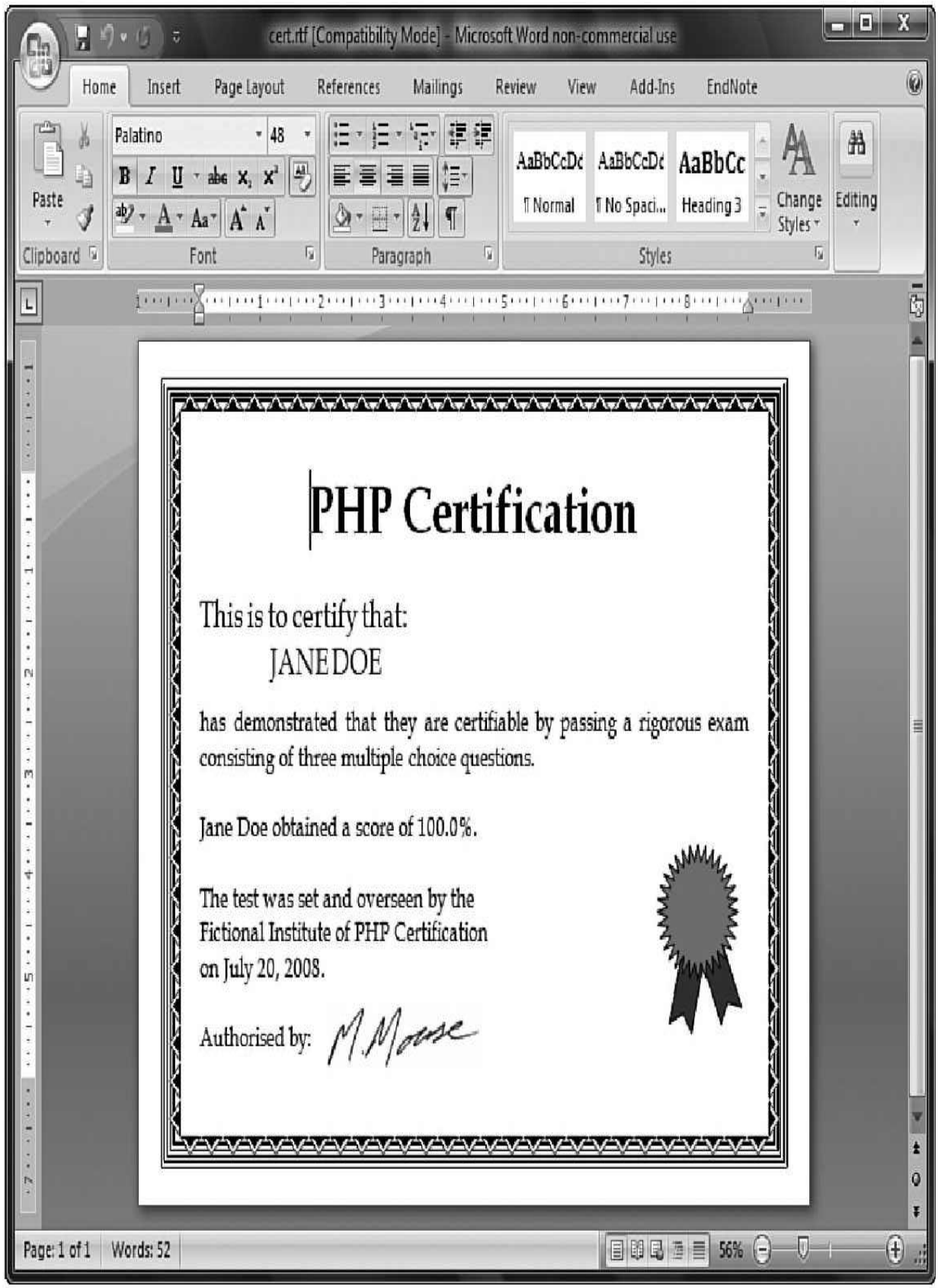

cert.rtf [Compatibility Mode] - Microsoft Word non-commercial use
Home
Insert
Page Layout
References
Mailings
Review
View
Add-Ins
EndNote
Paste
Clipboard
Palatino
48
Font
Paragraph
AaBbCcDc
¶ Normal
AaBbCcDc
¶ No Spaci...
AaBbCc
Heading 3
Change Styles
Editing
Styles
PHP Certification
This is to certify that:
JANE DOE
has demonstrated that they are certifiable by passing a rigorous exam consisting of three multiple choice questions.
Jane Doe obtained a score of 100.0%.
The test was set and overseen by the Fictional Institute of PHP Certification on July 20, 2008.
Authorised by:
M. Mouse
Page: 1 of 1
Words: 52
56%

图　32-5　rtf.php根据RTF模板生成一个证书

这个方法效果很好。即使在模板和$output变量的内容非常长的情况下，函数str_replace()的调用执行速度也很快。从这个应用程序的角度来看，主要的问题就是用户将在他自己的字处理器软件中载入证书并打印。这可能会导致用户能够修改证书。RTF不允许我们创建一个只读文档。

32.3.4　从模板生成PDF证书

从模板生成PDF证书的过程非常相似。主要的差异在于当创建PDF文件时，某些占位符可能会被解释为一些格式化代码，这将根据所使用的Acrobat版本不同而不同。例如，如果我们查看所创建的证书模板（使用一个文本编辑器），可以看到占位符将变成：

```
<<N)-13(AME)-10(>)-6(>
<<Na)-9(m)0(e)-18(>>
<)-11(<)1(sc)-17(or)-6(e)-6(>)-11(>
<)-11(<)1(m)-12(m)0(/d)-6(d)-19(/)1(yy)-13(yy)-13(>>
```

如果阅读整个文件，将发现与RTF不同，这不是一个我们能够轻易理解的格式。

提示　根据所使用的Acrobat版本或其他PDF生成工具的不同，PDF模板文件也会有所不同。在以上示例所提供的代码可能无法适用于你所生成的模板。请检查模板并且相应地修改其中的代码。如果还存在问题，可以使用本章稍后给出的PDFlib示例。

对于这种情况，我们可以使用几种不同的办法来解决问题。可以检查每个占位符并删除这些格式化符号。事实上，删除这些格式化符号最终将不会太多改变文档外观，因为嵌入到前面模板中的代码表示了我们将要替代的占位字符之间需要留出多少空间。然而，如果采取这种方法，必须在每次修改或更新文件的时候，仔细检查并手动编辑PDF文件。如果仅有4个占位符的时候这也许并不是大问题，但是当需要处理有许多占位符的多个文档，并决定要修改所有文档的信头时，这将会是一件非常可怕的事情。

我们可以通过使用其他技巧来避免此问题。可以用Adobe Acrobat来创建一个PDF表单——与带有空白命名域的HTML表单相似。然后可以用PHP脚本创建所谓的FDF（Forms Data Format）文件，该文件主要是一些将要合并到模板中的数据。可以用PHP的FDF函数库创建FDF：fdf_create()函数创建一个文件：fdf_set_value()函数设置表单域的值；fdf_set_file()函数设置相关的模板表单文件。可以用适当的MIME类型

将该文件返回到浏览器中（在这个例子中，MIME类型是vnd.fdf），并且用浏览器的Acrobat Reader插件程序来替换数据并填充到表单中。

这是一个非常巧妙的方法，但是，它也有两点局限性。首先，它需要一份Acrobat专业版的副本（完整版本，不是免费的阅读器，也不是标准版）。其次，在行内的文本而非表单域中进行文本替换是相当困难的。这可能是一个问题，但也可能不是问题，这是由具体操作来决定的。我们已在需要成行地替换许多文本的地方大量使用PDF生成功能来生成字符。FDF不能很好地完成这个功能。例如，如果自动填充一个在线税收表单，这不会成为一个问题。

在Adobe站点，可以看到更多关于FDF格式的介绍：http://www.adobe.com/devnet/acrobat/fdftoolkit.html。

如果打算使用这种方法的话，还应该查看PHP手册中的FDF文档：http://www.php.net/manual/en/ref.fdf.php。

现在，我们回到解决前面问题的PDF方案。

如果我们可以确认附加的格式代码只是由单个的连字符、阿拉伯数字及圆括号组成，并且能够通过正则表达式进行匹配，那么我们仍然可在PDF文件中找出并替换这些占位符。我们已经写了一个pdf_replace()函数来自动地生成某个占位符的匹配正则表达式，并能用适当的文本替代该占位符。

请注意，在Acrobat的某些版本中，占位符以纯文本格式出现。正如前面所介绍的，可以用str_replace()函数来替换它们。

除了这点外，通过PDF模板生成证书的代码与RTF版本的也非常相似，该脚本程序如程序清单32-4所示。

程序清单32-4　pdf.php——通过模板产生个性化PDF证书的脚本程序

```
<?php
set_time_limit(180);//this script can be slow
//create short variable names
```

```
$name=$_POST['name'];
$score=$_POST['score'];
function pdf_replace($pattern,$replacement,$string){
$len=strlen($pattern);
$regexp='';
for($i=0;$i<$len;$i++){
$regexp.=$pattern[$i];
if($i<$len-1){
$regexp.="(\)\-{0,1}[0-9]*\(){0,1}";
}
}
return ereg_replace($regexp,$replacement,$string);
}
if(!$name||!$score){
echo"<h1>Error:</h1>
<p>This page was called incorrectly</p>";
}else{
//generate the headers to help a browser choose the correct application
header('Content-Disposition:filename=cert.pdf');
header('Content-type:application/pdf');
$date=date('F d,Y');
//open our template file
$filename='PHPCertification.pdf';
$fp=fopen($filename,'r');
//read our template into a variable
$output=fread($fp,filesize($filename));
fclose($fp);
//replace the place holders in the template with our data
$output=pdf_replace('<<NAME>>',strtoupper($name),$output);
$output=pdf_replace('<<Name>>',$name,$output);
$output=pdf_replace('<<score>>',$score,$output);
$output=pdf_replace('<<mm/dd/yyyy>>',$date,$output);
//send the generated document to the browser
echo$output;
}
?>
```

该脚本产生PDF文档的一个自定义版本。如图32-6所示，该文档在多数系统中可以稳定地打印出来，且接收者无法修改其内容。我们会发现图32-6中的PDF文档看起来很像图32-5中的RTF文档。

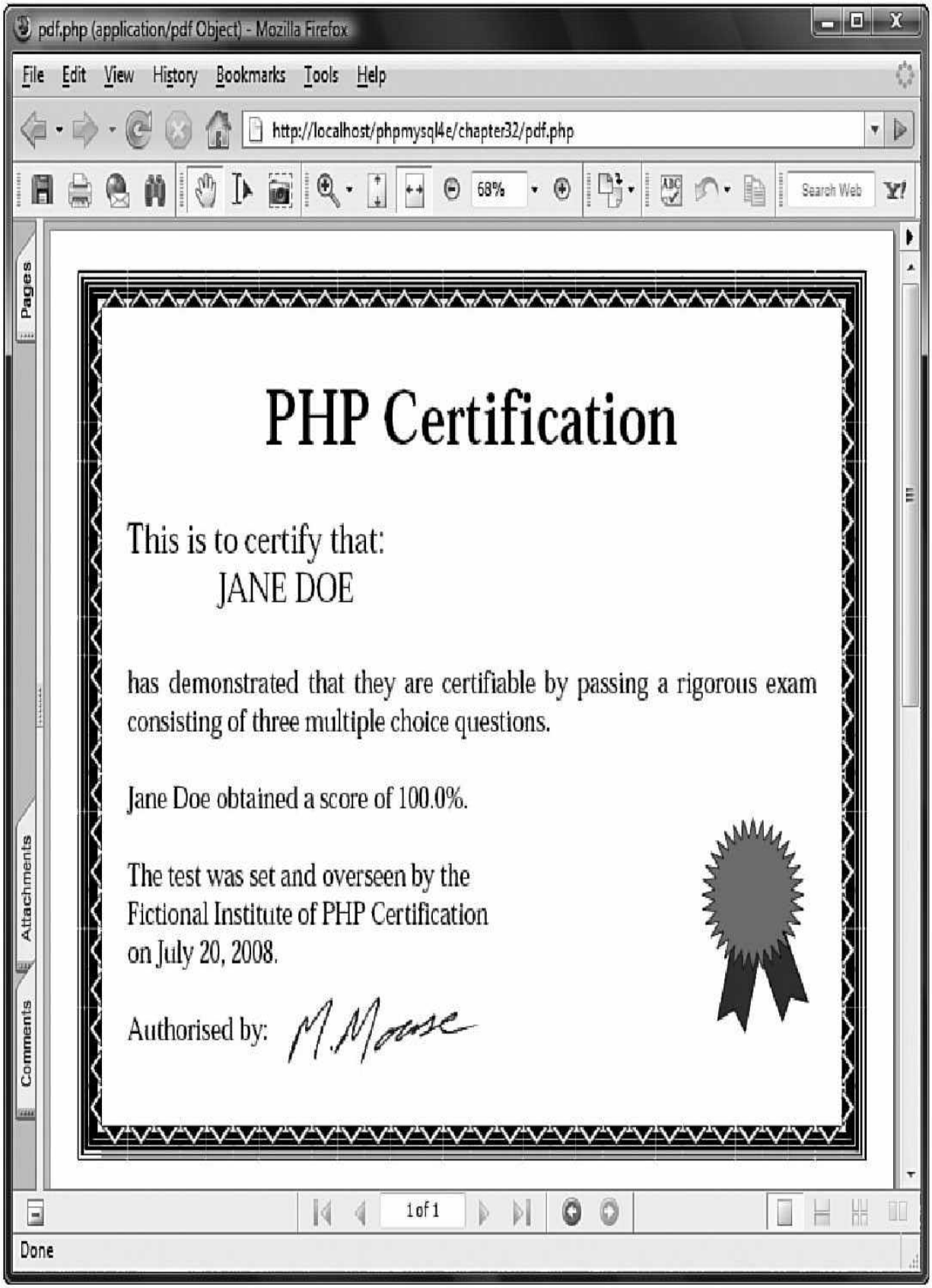

pdf.php (application/pdf Object) - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/phpmysql4e/chapter32/pdf.php
68%
Search Web
Pages
Attachments
Comments
PHP Certification
This is to certify that:
JANE DOE
has demonstrated that they are certifiable by passing a rigorous exam consisting of three multiple choice questions.
Jane Doe obtained a score of 100.0%.
The test was set and overseen by the Fictional Institute of PHP Certification on July 20, 2008.
Authorised by: M. Mouse
1 of 1
Done

图　32-6　pdf.php用PDF模板生成证书

该方法的一个问题是由于正则表达式的匹配使得该代码的运行速度非常慢。正则表达式比在RTF版本中用到的str_replace()函数要慢得多。

如果要在相同的服务器上匹配大量的占位符，或者试图生成许多这样的文档，可能需要寻求其他方法。而对于简单的模板，这个问题稍小一些。该文件中的大部分数据都是表示图像。

32.3.5 使用PDFlib生成PDF文档

PDFlib是为能通过Web生成动态PDF文档而设计的。严格地说，它并不是PHP的一部分，而是一个单独的库，它提供了大量函数供多种编程语言调用。绑定的编程语言有C、C++、Java、Perl、Python、Tcl和ActiveX/COM。

PDFlib由PDFlib GmbH正式支持。这就意味着你可以参考http://www.php.net/en/manual/ref.pdf.php处的PHP文档或者从http://www.pdflib.com下载正式文档。

32.3.6　使用PDFlib的一个"Hello World"程序

在拥有并以启用PDFlib的方式安装了PHP之后，可以用一个简单的程序（例如Hello World）来测试它，如程序清单32-5所示。

程序清单32-5　testpdf.php——通过PHP使用PDFlib编写的Hello World典型示例

```
<?php
//create a pdf document in memory
$pdf=pdf_new();
pdf_open_file($pdf,"");
pdf_set_info($pdf,"Author","Luke Welling and Laura Thomson");
pdf_set_info($pdf,"Title","Hello World(PHP)");
pdf_set_info($pdf,"Creator","testpdf.php");
pdf_set_info($pdf,"Subject","Test PDF");
//US letter is 11"x 8.5"and there are 72 points per inch
pdf_begin_page($pdf,8.5*72,11*72);
//add a bookmark
pdf_add_bookmark($pdf,'Page 1',0,0);
$font=pdf_findfont($pdf,'Times-Roman','host',0);
pdf_setfont($pdf,$font,24);
pdf_set_text_pos($pdf,50,700);
//write text
pdf_show($pdf,'Hello,world!');
pdf_continue_text($pdf,'(says PHP)');
//end the document
pdf_end_page($pdf);
pdf_close($pdf);
$data=pdf_get_buffer($pdf);
//generate the headers to help a browser choose the correct application
header('Content-Type:application/pdf');
header('Content-Disposition:inline;filename=testpdf.pdf');
header('Content-Length:'.strlen($data));
```

```
//output PDF
echo$data;
?>
```

如果该脚本执行失败，最有可能看到的出错信息如下所示：

```
Fatal error:Call to undefined function pdf_new()
in C:\Program Files\Apache Software
Group\Apache2.2\htdocs\phpmysql4e\chapter32\testpdf.php on line 4
```

这表明我们没有将PDFlib扩展库的支持编译到PHP中。

安装过程相当直观，但一些细节的修改将根据所使用的PHP和PDFlib的不同而有所不同。一个寻求具体细节意见的好地方就是注释过的PHP手册中PDFlib页上用户添加的注释。

当准备好了该脚本并在系统中运行后，可以开始了解它如何工作了。如下所示的代码：

```
$pdf=pdf_new();
pdf_open_file($pdf,"");
```

将在内存中初始化一个PDF文档。

pdf_set_info()函数允许使用一个“主题；标题；创建者；作者”这些关键字列表以及对应的用户自定义域来标记一个文档。

在这里，我们设置了创建者、作者和标题以及主题。注意这6个信息字段都是可选的：

```
pdf_set_info($pdf,"Author","Luke Welling and Laura Thomson");
pdf_set_info($pdf,"Title","Hello World(PHP)");
pdf_set_info($pdf,"Creator","testpdf.php");
pdf_set_info($pdf,"Subject","Test PDF");
```

一个PDF文档由很多页组成。要开始一个新页面，必须调用pdf_begin_page()函数。pdf_begin_page()函数的参数除了要求有pdf_open()函数返回的标识符外，还要求有页面大小的参数。文档中的每一页都可以有不同的大小，但是应该尽量使用相同的页面大小，除非有充足的理由使用不同大小的页面。

PDFlib是以点（point）为度量单位的，无论是对于页面大小还是在页面中定位坐标位置都是这样的。作为参考，A4纸的大小大概是595×842点，美国信函纸大小为612×792。如下代码行：

```
pdf_begin_page($pdf,8.5*72,11*72);
```

表示在文档中创建了一个美国信函大小的页面。

一个PDF文档不只限于具有打印功能的文档。在文档中，还可以包含许多PDF特征，比如超链接和书签等。pdf_add_outline()函数可以将书签添加到文档提纲中。文档中的书签将以单独的窗格式出现在Acrobat阅读器中，这样允许我们直接跳到重要的章节。

如下代码行：

```
pdf_add_bookmark($pdf,'Page 1',0,0);
```

将增加一条带有标签Page1的大纲记录，该记录指向当前页。

系统中可用字体将根据操作系统不同而不同，甚至对于不同的机器也不一样。为了保证结果的一致性，PDF阅读器包含了一组核心字体。这14个核心字体分别是：

■Courier

■Courier-Bold

■Courier-Oblique

■Courier-BoldOblique

■Helvetica

■Helvetica-Bold

■Helvetica-Oblique

■Helvetica-BoldOblique

■Times-Roman

■Times-Bold

■Times-Italic

■Times-BoldItalic

■Symbol

■ZapfDingbats

文档中也可以包含以上字体集合以外的字体。但这可能会增加文件的大小，而且可能不被某些特殊的字体许可所支持。我们可以选择字体、大小和字符编码方式，如下所示：

```
$font=pdf_findfont($pdf,'Times-Roman','host',0);
pdf_setfont($pdf,$font,24);
```

字体大小是按点计算的。在这个例子中，我们选择了主机字符编码。这种编码允许的取值有winansi、builtin、macroman、ebcdic或host。不同取值的含义如下所示：

■winansi——由Microsoft在ISO8859-1字符集的基础上再添加的特殊符号，比如欧元符号。

■builtin——使用字体内置的编码。通常，用于非拉丁字体和符号。

■macroman——Mac Roman编码。默认的Macintosh字符集。

■ebcdic——用在IBM AS/400系统中的EBCDIC。

■host——基于Mac的系统自动选择macroman，基于EBCDIC的系统选择ebcdic，所有其他系统选择winansi。

如果不需要包含特殊字符，那么这个编码选项并不重要。

PDF文档不像HTML文档或字处理器文档。在默认情况下，文本并不是简单地从左上方开始，接下来是其他行。我们必须选择在哪里放置每一行文本。正如前面已经提到的，PDF用点来指定位置。原点（X，Y坐标为[0，0]）位于页面的左下角。

假定页面大小是612×792点，点（50，700）大约距离页面的左边2/3英寸，距离顶部1/3英寸。要将文本位置设置到该点，可以使用如下所示的代码：

```
pdf_set_text_pos($pdf,50,700);
```

最后，设置好页面后，可以在上面书写文本了。我们用pdf_show()函数在当前点以当前字体添加文本。

如下代码行：

```
pdf_show($pdf,'Hello,world!');
```

将"Hello world!"添加到文档中。

要移到下一行并接着写入更多文本，可以使用pdf_continue_text()函数。要在文档中增加字符串"（says PHP）"，可以使用如下所示语句：

```
pdf_continue_text($pdf,'(says PHP)');
```

该字符串的确切显示位置将由所选的字体及大小来确定。

如果还要插入其他行、短语和段落，pdf_show_boxed()函数非常有用。它允许声明一个文本框并在其中输入文本。

完成将元素添加到页面后，要以如下方式调用pdf_end_page()函数：

```
pdf_end_page($pdf);
```

生成整个PDF文档后，还需要调用pdf_close()函数关闭该文档。当要生成一个文件时，也需要关闭这个文件。

如下代码行：

```
pdf_close($pdf);
```

将完成Hello World整个文档。

现在，我们可以将所完成的PDF文档发送给浏览器：

```
$data=pdf_get_buffer($pdf);
//generate the headers to help a browser choose the correct application
header('Content-Type:application/pdf');
header('Content-Disposition:inline;filename=testpdf.pdf');
header('Content-Length:'.strlen($data));
//output PDF
echo$data;
```

我们也可以将该数据写入磁盘。通过以文件名称作为pdf_open_file()函数的第二个参数，PDFlib允许将数据写入磁盘中。

请注意，在PDFlib的某些版本中，PHP手册中给出的某些可选的PDFlib函数参数是必需的。关于证书的文档相对要复杂一些，它包含边界、一幅向量图和一幅位图。使用其他两种技巧，我们可以用字处理器来添加这些元素。如果使用PDFlib，我们则必须手动添加。

32.3.7　用PDFlib生成证书

在这个项目中，为了使用PDFlib，我们需要做一些折中。虽然精确复制前面用到的证书是肯定可以的，但仍需要努力地手动生成及定位每一个元素，而不能使用像Microsoft Word这样的工具来协助编排文档。

我们使用与前面相同的文本，包括红色的圆花饰及位图签名，但我们不打算复制复杂的边界。该脚本的完整代码如程序清单32-6所示。

程序清单32-6　pdflib.php——用PDFlib生成证书

```
<?php
//create short variable names
$name=$_POST['name'];
$score=$_POST['score'];
if(!$name||!$score){
echo"<h1>Error:</h1>
<p>This page was called incorrectly</p>";
}else{
$date=date('F d,Y');
//create a pdf document in memory
$pdf=pdf_new();
pdf_open_file($pdf,"");
//set up name of font for later use
$fontname='Times-Roman';
//set up the page size in points and create page
//US letter is 11"x 8.5"and there are approximately
//72 points per inch
$width=11*72;
$height=8.5*72;
pdf_begin_page($pdf,$width,$height);
//draw our borders
$inset=20;//space between border and page edge
$border=10;//width of main border line
```

```
$inner=2;//gap within the border
//draw outer border
pdf_rect($pdf,$inset-$inner,
$inset-$inner,
$width-2*($inset-$inner),
$height-2*($inset-$inner));
pdf_stroke($pdf);
//draw main border$border points wide
pdf_setlinewidth($pdf,$border);
pdf_rect($pdf,$inset+$border/2,
$inset+$border/2,
$width-2*($inset+$border/2),
$height-2*($inset+$border/2));
pdf_stroke($pdf);
pdf_setlinewidth($pdf,1.0);
//draw inner border
pdf_rect($pdf,$inset+$border+$inner,
$inset+$border+$inner,
$width-2*($inset+$border+$inner),
$height-2*($inset+$border+$inner));
pdf_stroke($pdf);
//add heading
$font=pdf_findfont($pdf,$fontname,'host',0);
if($font){
pdf_setfont($pdf,$font,48);
}
$startx=($width-pdf_stringwidth($pdf,'PHP Certification',
$font,'12'))/2;
pdf_show_xy($pdf,'PHP Certification',$startx,490);
//add text
$font=pdf_findfont($pdf,$fontname,'host',0);
if($font){
pdf_setfont($pdf,$font,26);
}
$startx=70;
pdf_show_xy($pdf,'This is to certify that:',$startx,430);
```

```
pdf_show_xy($pdf,strtoupper($name),$startx+90,391);
$font=pdf_findfont($pdf,$fontname,'host',0);
if($font)
pdf_setfont($pdf,$font,20);
pdf_show_xy($pdf,'has demonstrated that they are certifiable'.
'by passing a rigorous exam',$startx,340);
pdf_show_xy($pdf,'consisting of three multiple choice questions.',
$startx,310);
pdf_show_xy($pdf,"$name obtained a score of$score".'%.',$startx,260);
pdf_show_xy($pdf,'The test was set and overseen by the',$startx,210);
pdf_show_xy($pdf,'Fictional Institute of PHP Certification',
$startx,180);
pdf_show_xy($pdf,"on$date.",$startx,150);
pdf_show_xy($pdf,'Authorised by:',$startx,100);
//add bitmap signature image
$signature=pdf_load_image($pdf,'png','/Program Files/Apache Software
Foundation/Apache2.2/htdocs/phpmysql4e/chapter32/signature.png','');
pdf_fit_image($pdf,$signature,200,75,'');
pdf_close_image($pdf,$signature);
//set up colors for rosette
pdf_setcolor($pdf,'both','cmyk',43/255,49/255,1/255,67/255);//dark
blue
pdf_setcolor($pdf,'both','cmyk',1/255,1/255,1/255,1/255);//black
//draw ribbon 1
pdf_moveto($pdf,630,150);
pdf_lineto($pdf,610,55);
pdf_lineto($pdf,632,69);
pdf_lineto($pdf,646,49);
pdf_lineto($pdf,666,150);
pdf_closepath($pdf);
pdf_fill($pdf);
//outline ribbon 1
pdf_moveto($pdf,630,150);
pdf_lineto($pdf,610,55);
pdf_lineto($pdf,632,69);
pdf_lineto($pdf,646,49);
```

```
pdf_lineto($pdf,666,150);
pdf_closepath($pdf);
pdf_stroke($pdf);
//draw ribbon 2
pdf_moveto($pdf,660,150);
pdf_lineto($pdf,680,49);
pdf_lineto($pdf,695,69);
pdf_lineto($pdf,716,55);
pdf_lineto($pdf,696,150);
pdf_closepath($pdf);
pdf_fill($pdf);
//outline ribbon 2
pdf_moveto($pdf,660,150);
pdf_lineto($pdf,680,49);
pdf_lineto($pdf,695,69);
pdf_lineto($pdf,716,55);
pdf_lineto($pdf,696,150);
pdf_closepath($pdf);
pdf_stroke($pdf);
pdf_setcolor($pdf,'both','cmyk',1/255,81/255,81/255,20/255);//red
//draw rosette
draw_star(665,175,32,57,10,$pdf,true);
//outline rosette
draw_star(665,175,32,57,10,$pdf,false);
//finish up the page and prepare to output
pdf_end_page($pdf);
pdf_close($pdf);
$data=pdf_get_buffer($pdf);
//generate the headers to help a browser choose the correct application
header('Content-type:application/pdf');
header('Content-disposition:inline;filename=test.pdf');
header('Content-length:'.strlen($data));
//output PDF
echo$data;
}
function draw_star($centerx,$centery,$points,$radius,
```

```
$point_size,$pdf,$filled){
$inner_radius=$radius-$point_size;
for($i=0;$i<=$points*2;$i++){
$angle=($i*2*pi())/($points*2);
if($i%2){
$x=$radius*cos($angle)+$centerx;
$y=$radius*sin($angle)+$centery;
}else{
$x=$inner_radius*cos($angle)+$centerx;
$y=$inner_radius*sin($angle)+$centery;
}
if($i==0){
pdf_moveto($pdf,$x,$y);
}else if($i==$points*2){
pdf_closepath($pdf);
}else{
pdf_lineto($pdf,$x,$y);
}
}
if($filled){
pdf_fill_stroke($pdf);
}else{
pdf_stroke($pdf);
}
}
?>
```

使用以上脚本生成的证书如图32-7所示。可以看到，它与其他证书非常相似，只是边界简单一些，星状图案看起来稍有不同。这是因为我们这里是将它绘制到文档中的，而不是使用一个已存在的剪切美术文件。

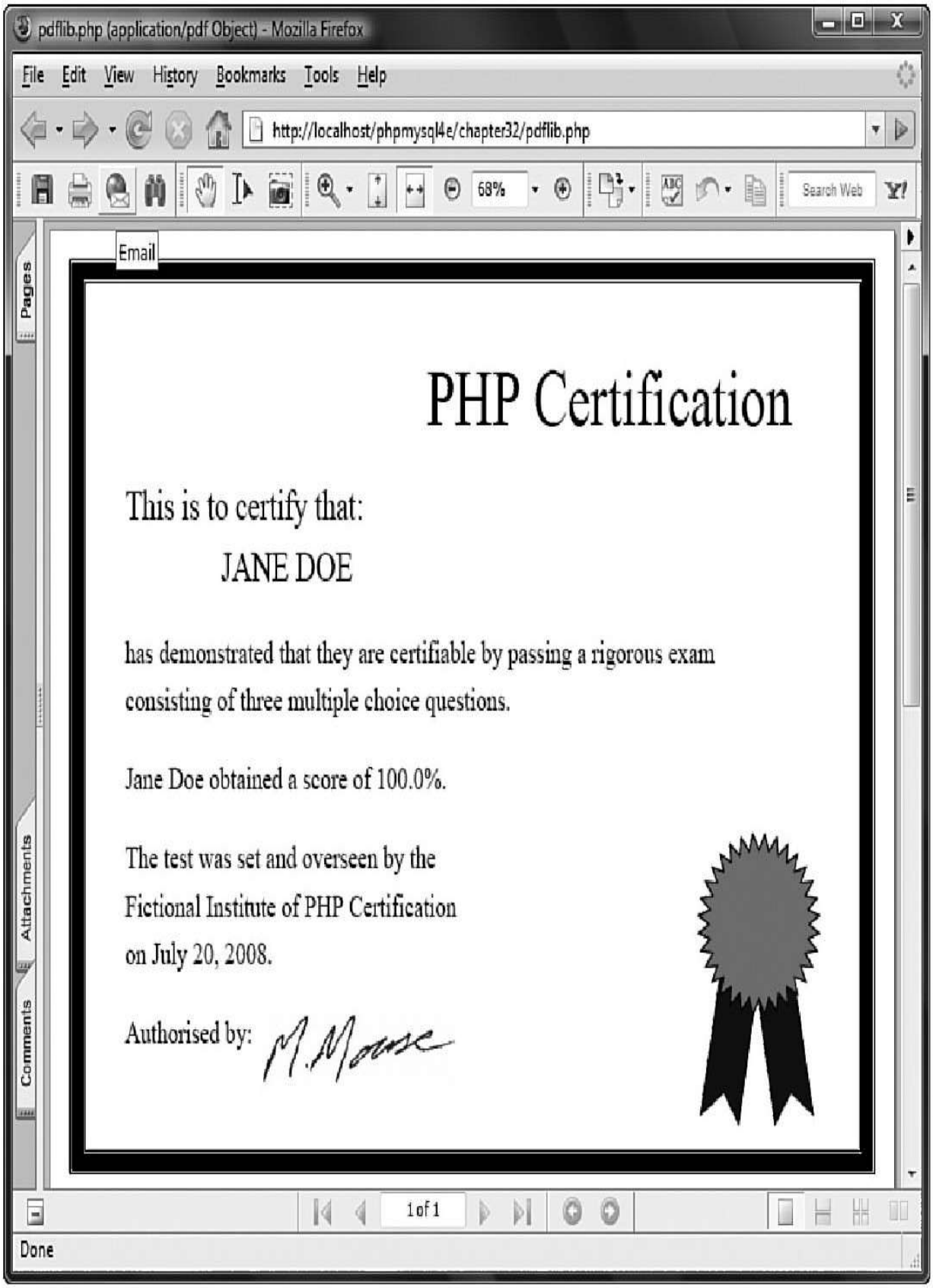

pdflib.php (application/pdf Object) - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/phpmysql4e/chapter32/pdflib.php
68%
Search Web
Email
Pages
Attachments
Comments
PHP Certification
This is to certify that:
JANE DOE
has demonstrated that they are certifiable by passing a rigorous exam consisting of three multiple choice questions.
Jane Doe obtained a score of 100.0%.
The test was set and overseen by the Fictional Institute of PHP Certification on July 20, 2008.
Authorised by: M. Mouse
1 of 1
Done

图　32-7　pdflib.php将证书画到一个PDF文档中

现在，我们介绍一下该脚本与前面的例子不同的部分。

由于访问者需要在证书中获得关于自己的资料，因此我们将在内存中创建该文档而不是在文件中。如果将它写入一个文件，需要考虑能够生成唯一文件名的机制，并且能够防止人们偷偷进入其他人的证书，以及选择一种删除旧文件以释放服务器上硬盘空间的机制。

为了在内存中创建一个文档，我们在调用了pdf_open_file()函数后再调用不带参数的pdf_new()函数。如下所示：

```
$pdf=pdf_new();
pdf_open_file($pdf,"");
```

简化的边界将包含3个条纹：一条粗边框和两条细边框，其中一个在主边框里面，另一个在外面。我们将以矩形方式绘制它们。

为了以这种方法确定边界的位置以便轻易调整页面的大小或边界的外观，我们把所有边界位置建立在已有的变量$width和$height，以及新变量$inset、$border和$inner的基础上。用$inset指定位于页面边缘的边界有多少点宽，$border来指定主边框的厚度，以及使用$inner来指定主边框与两条细边框之间的缝隙有多宽。

如果我们使用过其他的图形API来绘制图形，在使用PDFlib绘制图形时，就不会有太多的惊讶。第22章对于这项工作是有帮助的，因为用gd库绘制图形与用PDFlib绘制图形非常相似。

细边框是比较简单的。为了创建一个长方形，可以使用pdf_rect()函数。该函数要求提供如下参数：PDF文档标识符、长方形的左下角的X和Y坐标以及宽和高。因为我们希望排列能够灵活些，因此用已设置的变量来计算这些：

```
pdf_rect($pdf,$inset-$inner,
$inset-$inner,
$width-2*($inset-$inner),
```

```
$height-2*($inset-$inner));
```

对pdf_rect()数的调用可以设置一块长方形的区域。为了绘制出这个形状，需要调用pdf_stroke()函数。如下所示：

```
pdf_stroke($pdf);
```

为了绘制主边框，需要指定线宽。在默认的情况下，线宽为1个点。如下代码中，对pdf_setlinewidth()函数的调用将它设为$border（在这个例子中，设置为10）点：

```
pdf_setlinewidth($pdf,$border);
```

在设置好宽度后，我们再次调用pdf_rect()函数来创建一个矩形并调用pdf_stroke()函数将它绘制出来：

```
pdf_rect($pdf,$inset+$border/2,
$inset+$border/2,
$width-2*($inset+$border/2),
$height-2*($inset+$border/2));
pdf_stroke($pdf);
```

在绘制好一条宽线后，我们必须记住用如下所示代码将宽度重新设置为1：

```
pdf_setlinewidth($pdf,1.0);
```

我们使用pdf_show_xy()函数来定位证书中每一行文本的位置。对于大多数文本行，我们使用可配置的左边距（$startx）作为X坐标。如果想要将标题置于页面的中央，那么需要知道它的宽度来确定左面的位置。我们可以使用pdf_stringwidth()函数来得到其宽度。如下所示的函数调用：

```
pdf_stringwidth($pdf,'PHP Certification',$font,'12')
```

将以当前的字体与大小返回字符串"PHP Certification"的宽度。

对于其他版本的证书，需要嵌入一个扫描后的位图签名。如下所示的3个语句：

```
$signature=pdf_load_image($pdf,'png','/Program Files/Apache Software
Foundation/Apache2.2/htdocs/phpmysql4e/chapter32/signature.png','');
pdf_fit_image($pdf,$signature,200,75,'');
pdf_close_image($pdf,$signature);
```

将打开一个包含签名的GIF文件，将图像添加到页面的指定位置并关闭该GIF文件。我们也可以使用其他类型的文件。

提示　当通过pdf_load_image()函数载入一个图像时，使用该文件的完整路径。在这个例子中，signature.png的完整路径是Windows系统的路径。

使用PDFlib添加到证书中的元素最困难的是那个圆饰物。我们不能自动地打开已有的包含该圆饰物的Windows meta文件，但可以任意绘制任何想要的图形。

为了绘制一个填充的图形，例如其中一个锯齿，我们编写了如下所示的代码。在这里，条纹和直线的颜色都设为黑色，填充或内部颜色设为暗蓝色：

```
pdf_setcolor($pdf,'fill','rgb',0,0,.4,0);//dark blue
pdf_setcolor($pdf,'stroke','rgb',0,0,0,0);//black
```

这里，我们建立了一个五边形作为其中一个锯齿并填充它：

```
pdf_moveto($pdf,630,150);
pdf_lineto($pdf,610,55);
```

```
pdf_lineto($pdf,632,69);
pdf_lineto($pdf,646,49);
pdf_lineto($pdf,666,150);
pdf_closepath($pdf);
pdf_fill($pdf);
```

就像多边形的轮廓一样，我们需要用相同的方法再一次设置，但是这次是调用pdf_stroke()函数而不是pdf_fill()函数。

由于多点星形是一种复杂形状，我们编写了一个函数来计算该形状沿途路径的位置。该函数叫做draw_star()，它要求中点的X和Y坐标、角点的数目、半径、角点的长度、PDF文件标识符和一个布尔值来表示该星形应填充还是仅绘制其轮廓。

draw_star()函数用到了一些基本的三角学来计算形成一颗星的一系列点的位置。对于星形具有的每个点，我们在星形的半径上找出一个点，在外圆周的更小的圆$point_size上再找出一个点，并在它们之间画一条直线。一个值得注意的地方是PHP的三角函数如cos()和sin()是以弧度而不是以度为单位的。

利用一个函数和一些数学知识，我们可以精确地生成一个复杂的重复性图形。如果希望以复杂的模式画页面边界，也可以使用相似的方法。

当产生了所有页面元素后，需要关闭该页面及文档。

32.4 处理标题的问题

在所有这些脚本中，需要注意的一个细小问题是需要告诉浏览器将发送给它什么类型的数据。我们通过发送一个内容类型的HTTP标题来实现该操作，如下所示：

```
header('Content-type:application/msword);
```

或者

```
header('Content-type:application/pdf');
```

一个值得注意的地方是浏览器处理这些标题的方法并不一致。特别是Internet Explorer经常选择忽略MIME类型，而试图自动检测文件类型（这个特殊的问题似乎已经在Internet Explorer的最新版本中得到解决，因此，如果还遇到这样的问题，最简单的解决方法是升级浏览器）。

一些标题似乎会在会话控制标题中出现问题。如今，对于这个问题，已经有几种解决方法。我们发现通过使用GET参数而不是POST或会话变量参数可以避免该问题。

另一种方法，正如Hello World PDFlib示例中一样，不使用内嵌的PDF，而让用户下载它。

编写两种不同版本的代码也可以避免此类问题，一个适用于Netscape浏览器，另一个适用于Internet Explorer浏览器。

32.5 扩展该项目

显然，在测试中添加一些更现实的评测任务可以扩展该项目，但是，在本书中，该项目只是作为一个发布自己文档的方法示例。

通过在线传送方式发布的自定义文档可以包括法律文档、部分填充的订单或申请表单，以及政府部门需要的表单。

32.6 下一章

在下一章中，我们将介绍PHP 5的XML新功能以及使用PHP通过REST和SOAP协议访问Amazon的Web服务API。

第33章　使用XML和SOAP来连接Web服务

在最近几年中，XML（可扩展的置标语言）已经成为通信的重要方法。在本章中，我们使用亚马逊公司最新的Web服务接口在本地Web站点中创建一个以Amazon为后台的购物车（我们将这个应用程序命名为Tahuayo，它是亚马逊河区印第安人的名称）。我们将使用两种不同的方法来创建购物车：SOAP和REST。REST也就是基于HTTP协议之上的XML。我们将使用PHP内置的SimpleXML库和NuSOAP库来实现这两种方法。

在本章中，我们将主要介绍以下内容：

■理解XML和Web服务的基础知识

■使用XML与Amazon进行交互

■使用PHP的SimpleXML库解析XML

■缓存响应

■使用NuSOAP与Amazon进行交互

33.1　项目概述：使用XML和Web服务

在这个项目中，我们要实现两个目标：第一，理解什么是XML和SOAP并且掌握如何在PHP中使用它们。第二，使用这两种技术与外界进行通信。我们之所以选择Amazon的Web服务程序作为例子，是因为它对我们自己的Web站点将会非常有帮助。

很早以前，Amazon就提供了允许在Web站点给其产品做广告的相关程序。用户可以点击这些链接进入到Amazon站点上每一个产品的页面。如果某些用户通过某站点访问Amazon站点并且成功购买其产品，该站点将获得一定的现金奖励。

Amazon的这些Web服务程序允许将它作为一个引擎：我们可以搜索Amazon站点并且通过自己的站点显示搜索结果，或者在用户浏览我

们站点时将用户选择的产品直接放入购物车中。换句话说，客户在付费之前可以一直使用我们的站点，就像在Amazon站点上的购买操作一样。

我们和Amazon站点之间的交互可以按两种可能的方式进行。第一种方法是使用基于HTTP的XML，也就是表示状态转换（REST）。例如，如果我们希望使用这种方法执行一个搜索操作，可以向Amazon发送一个关于所查询信息的普通HTTP请求，Amazon将用包含请求信息的XML文档响应查询。于是，我们可以使用PHP的XML库对这个XML文档进行解析并且使用我们所选择的接口向终端用户显示搜索结果。通过HTTP发送和接收数据的过程非常简单，但是解析结果文档的难易程度则是由文档的复杂度决定的。

第二种方法是使用SOAP。SOAP是Web服务的标准协议之一。它是简单对象访问协议的缩写，但是最近该协议变得不再简单，因此其名称就有一点名不符实。最后的结果是该协议仍然叫做SOAP，但是它不再是一个缩写词了。

在这个项目中，我们将创建一个可以向Amazon SOAP服务器发送请求并获得响应的SOAP客户端。这些将包含我们使用基于HTTP的XML方法从Amazon服务器获得的响应信息相同的信息，但是我们将使用不同的方法来获取数据，也就是NuSOAP库。

在这个项目中，我们最终的目标是创建自己的、使用Amazon作为后台的图书销售Web站点。我们将创建两个版本：一个是使用REST，而另一个是使用SOAP。

在真正进入这个项目之前，我们需要熟悉XML和Web服务的常规结构和使用。

33.1.1　理解XML

下面，如果我们不熟悉XML和Web服务的概念，那么花一些时间来了解它们。

正如前面已经提到的，XML是可扩展的置标语言。在W3C的站点上，提供了其规范。在W3C的XML站点上，可以找到大量关于XML

的信息。W3C的XML站点URL如下所示：http://www.w3.org/XML/。

XML源自SGML（标准通用置标语言）。如果我们已经了解了HTML（假设我们的确已经了解了），这对快速理解XML概念是非常有帮助的。

XML是针对文档的，基于标记的文本格式。作为XML文档的一个例子，程序清单33-1给出了Amazon站点返回的、针对一个基于特定请求参数的HTTP XML请求的响应内容。

程序清单33-1　描述了本书第一版的XML文档

```
<?xml version="1.0"encoding="UTF-8"?>
<ItemLookupResponse
xmlns="http://webservices.amazon.com/AWSECommerceService/2005-03-23">
<Items>
<Request>
<IsValid>True</IsValid>
<ItemLookupRequest>
<IdType>ASIN</IdType>
<ItemId>0672317842</ItemId>
<ResponseGroup>Similarities</ResponseGroup>
<ResponseGroup>Small</ResponseGroup>
</ItemLookupRequest>
</Request>
<Item>
<ASIN>0672317842</ASIN>
<DetailPageURL>http://www.amazon.com/PHP-MySQL-Development-Luke-Welling/
dp/0672317842%3F%26linkCode%3Dsp1%26camp%3D2025%26creative%3D165953%26crea
tiveASIN%3D0672317842
</DetailPageURL>
<ItemAttributes>
<Author>Luke Welling</Author>
<Author>Laura Thomson</Author>
<Manufacturer>Sams</Manufacturer>
<ProductGroup>Book</ProductGroup>
```

```
<Title>PHP and MySQL Web Development</Title>
</ItemAttributes>
<SimilarProducts>
<SimilarProduct>
<ASIN>1590598628</ASIN>
<Title>Beginning PHP and MySQL:From Novice to Professional,
Third Edition(Beginning from Novice to Professional)</Title>
</SimilarProduct>
<SimilarProduct>
<ASIN>032152599X</ASIN>
<Title>PHP 6 and MySQL 5 for Dynamic Web Sites:
Visual QuickPro Guide</Title>
</SimilarProduct>
<SimilarProduct>
<ASIN>B00005UL4F</ASIN>
<Title>JavaScript Definitive Guide</Title>
</SimilarProduct>
<SimilarProduct>
<ASIN>1590596145</ASIN>
<Title>CSS Mastery:Advanced Web Standards Solutions</Title>
</SimilarProduct>
<SimilarProduct>
<ASIN>0596005431</ASIN>
<Title>Web Database Applications with PHP&amp;amp;MySQL,
2nd Edition</Title>
</SimilarProduct>
</SimilarProducts>
</Item>
</Items>
```

该文档是以如下语句行为开始的：

```
<?xml version="1.0"encoding="UTF-8"?>
```

这是一个标准的声明，它告诉我们接下来的文档将会是使用UTF-8字符编码的XML。

让我们看看该文档的正文。整个文档由打开和关闭的标记对组成，例如，你可以从以下代码的开放和关闭的Item标记看到：

```
<Item>
...
```

</Item>是一个元素，就像其在HTML中一样。而且，就像在HTML中，我们可以嵌套元素，例如，以上代码的ItemAttributes元素，它为Item元素内，同样，Author元素却位于ItemAttributes元素内：

```
<ItemAttributes>
<Author>Luke Welling</Author>
<Author>Laura Thomson</Author>
<Manufacturer>Sams</Manufacturer>
<ProductGroup>Book</ProductGroup>
<Title>PHP and MySQL Web Development</Title>
```

还像HTML中的一样，元素可以具有属性。就像这个例子中，例如：

Details元素只有一个属性url。由于URL非常长，在这里，它被分成了3行。

当然，XML和HTML还是有一些不同的。首先，在XML中，所有开始的标记都必须有一个对应的结束标记。这条规则的例外就是空元素，空元素的开始和结束标记都在一个标记内，因为它没有包括任何文本。如果我们熟悉XHTML，将会看到，我们可以在使用
的地方使用
标记。此外，所有元素必须正确地嵌套。在HTML解析器中，Text可能是合法的，但是在XML或XHTML中，它却是不合法的，这些标记必须正确地嵌套，Text。

注意XML和HTML之间最主要的区别在于我们可以在文档中设置自己的标记！这就是XML的灵活性。我们可以根据希望保存的数据来设计文档的结构。我们可以通过编写一个DTD（文档类型定义）或XML模式来规范化XML文档的结构。这两个文档都是用来描述一个给定的XML文档的结构。如果你愿意，我们可以将DTD或模式看作是一个类声明，而XML文档则是类的实例。在这个例子中，没有使用DTD或模式。

可以通过如下URL给出的Web服务查阅Amazon站点的XML模式：http://webservices.amazon.com/AWSECommerceService/AWSECommerceService.xsd。

你应该可以直接在浏览器中打开这个XML模式。

需要注意的是，除了最初的XML声明外，文档的所有正文都是包含在ItemLookupResponse元素中。这个元素就叫做该文档的根元素。下面，我们仔细查看这个元素：

```
<ItemLookupResponse
xmlns="http://webservices.amazon.com/AWSECommerceService/2005-03-23">
```

该元素具有一些特殊的属性。这些属性是XML的名称空间（namespace）。我们并不需要理解名称空间，因为在这个项目中，这并不重要，但是这将会是非常有用的。该元素的基本功能是确保元素和属性名称的正确性，这样当处理来自不同组织的文档时，这些通用名称不会相互冲突。

如果想要了解更多关于名称空间的内容，可以在如下URL找到“XML标准建议的名称空间”文档：http://www.w3.org/TR/REC-xml-names/。

如果想要了解更多关于XML的内容，在Internet上存在了大量的资料。W3C站点就是了解XML的一个非常不错的入门站点，在该站点上，介绍了关于XML和Web入门的海量图书。ZVON.org提供了关于XML的最佳入门教程。

33.1.2　理解Web服务

Web服务是通过Internet可供使用的应用程序接口。我们可以将Web服务看作是一个通过Web开放了其公有方法的类。现在，Web服务随处可见，而且商业领域的一些著名企业也正在通过Web服务逐步开放它们的功能。

例如，Google、Amazon、eBay和PayPal现在就可以提供一系列的Web服务。在学习了本章关于如何创建一个能够访问Amazon接口的客户端后，我们会发现创建Google的客户端也是非常简单而又直观的。可以在http://code.google.com/apis/找到更多信息。

在如下站点给出了不断发展的公用Web服务列表：http://www.xmethods.net。

在这种远程函数调用方法中，还涉及了一些核心协议。最重要的两个协议是SOAP和WSDL。

1.SOAP

SOAP是一个以请求和响应为驱动的消息传递协议，它允许客户端调用Web服务，服务器对客户端的调用进行响应。每一个SOAP消息，无论是请求的还是响应的，都是一个简单的XML文档。程序清单33-2给出了一个我们可能发送给Amazon的SOAP示例请求。事实上，这个请求产生的XML响应如程序清单33-1所示。

程序清单33-2　一个基于ASIN的SOAP搜索请求

```
<SOAP-ENV:Envelope>
<SOAP-ENV:Body>
<m:ItemLookup>
<m:Request>
<m:AssociateTag>webservices-20</m:AssociateTag>
<m:IdType>ASIN</m:IdType>
<m:ItemId>0672317842</m:ItemId>
```

```
<m:AWSAccessKeyId>0XKKZBBJHE7GNBWF2ZG2</m:AWSAccessKeyId>
<m:ResponseGroup>Similarities</m:ResponseGroup>
<m:ResponseGroup>Small</m:ResponseGroup>
</m:Request>
</m:ItemLookup>
</SOAP-ENV:Body>
```

SOAP消息以一个XML文档的声明为开始。所有SOAP消息的根元素是SOAP“信封”。在这个“信封”中，我们可以找到包含真正请求的Body元素。

这个请求是一个ItemLookup，在这个例子中，它请求Amazon服务器在其数据库中基于ASIN（表示Amazon.com标准条目号）搜索特定项。这个号码是Amazon数据库中每一个产品的唯一标识符。

我们可以将ItemLookup看作是对一个远程计算机的函数调用，而且包含在该元素中的所有元素和属性就是我们传递给这个函数的参数。在这个例子中，通过IdType元素，我们传递了"ASIN"参数，而真正的ASIN值（0672317842）通过ItemId元素进行了传递。这个ASIN是本书第一版的唯一标识符。我们还需要传递另一个参数，那就是AssociateTag,，这是你的Amazon会员ID；此外，还需要的参数有：希望的响应类型（通过ResponseGroup元素）；以及AWSAccessKeyId——Amazon分配的开发人员令牌。

这个请求的响应非常类似于程序清单33-1中的XML文档，但是它是封闭在一个SOAP信封中的。

当使用SOAP时，无论使用何种编程语言，通常要生成SOAP请求并且使用SOAP库通过程序来解释响应。这样做是非常不错的，因为它可以节省大量手动构建SOAP请求并解释响应所需的操作。

2.WSDL

WSDL是Web服务描述语言的缩写（通常，这个词的发音是"wiz-dul"）。这个协议是用来描述特定Web站点上可供使用的接口的。如果想了解用来描述本章中使用的Amazon站点所提供Web服务的WSDL文

档，可以访问http://soap.amazon.com/schemas2/AmazonWebServices.wsdl。

如果点击这个链接，将发现WSDL文档明显比SOAP消息要复杂。如果让我们选择的话，肯定会选择使用程序来生成请求并解释响应。

如果希望了解更多关于WSDL的内容，可以访问如下URL：http://www.w3.org/TR/wsdl20/。

33.2 解决方案的组成

实现这个解决方案需要不同的组件。最明显的部分是一个能够向顾客显示的购物车接口和通过REST或SOAP连接到Amazon的代码，除此之外，还需要一些辅助的部分。在接收一个XML文档后，代码必须解析它，并且提取购物车将要显示的信息。要满足Amazon的要求并且提高性能，必须考虑缓存机制。最后，由于付费结账操作必须在Amazon完成，需要一些能够向Amazon提交用户购物车内容以及将用户提交给该服务的功能。

很明显，我们需要创建一个购物车作为系统的前台。在第28章中已经介绍了如何创建一个购物车。由于购物车并不是该项目的主要部分，我们将使用一个简化的应用程序。我们只需要提供一个基本的购物车，这样就可以记录客户想要购买哪些产品，并且在用户结账时将这些产品报告给Amazon。

33.2.1 使用Amazon的Web服务接口

要使用Amazon（亚马逊）的Web服务接口，我们需要注册一个开发人员令牌。可以在http://aws.amazon.com站点完成开发人员的注册。当请求到达Amazon站点时，这个令牌可以用在站点上标识开发人员本身。

我们可能还会希望注册一个Amazon会员ID。当任何客户通过我们的接口在Amazon购买产品时，通过这个ID，Amazon会给我们返回一些现金奖励。

http://developer.amazonwebservices.com/ 提供的Amazon Web服务（AWS）开发人员资源中心包含了大量关于使用SOAP和REST连接Amazon所有Web服务的文档、教程以及示例代码。本章给出的示例程序将生成一个可用的系统，并且提供连接AWS并获取信息的基本知识，但是如果你打算构建一个类似本章的应用，你就应该花些时间阅读这些文档。例如，你可能希望通过浏览和直接搜索接口搜索并获得不同商品的信息。根据所需元素的不同，所返回的数据也有不同的结构。AWS开发人员指南给出了所有这些信息。

提示　另一个有价值的资源是AWSZone.com（http://www.awszone.com）。在这个网站上，你可以测试SOAP和REST查询，并且了解请求以及响应的结构，这样你就可以知道如何处理返回的数据。此外，测试响应可以有助于确定确切的ResponseGroup对象，从而获得最佳的速度。

在注册开发人员令牌时，你必须同意这个许可协议。这个协议值得一读，因为它不是普通的软件协议。在实现时，许可的某些条件还是非常重要的，这些条件如下所示：

■作为客户端，每秒钟不能发出多个请求。

■必须缓存来自Amazon的数据。

■可以24小时缓存大多数数据，而一些稳定属性可以缓存至3个月。

■如果缓存价格或库存信息超过1小时，必须提供一些声明。

■必须将本地所拥有的所有Amazon数据链接到Amazon站点的页面，严禁将从Amazon下载的文本或图形链接至其他商业网站。

由于没有一个拼写简单的域名，没有推广政策以及其他使用Tahuayo.com的明显理由，所以我们还是直接使用Amazon网站，我们不需要采取任何特殊的措施来保障每秒不超过一个的请求。

在这个项目中，我们已经实现了缓存技术来保障协议条件的第2至4点。我们可以将图像缓存24小时，而产品数据（包含了价格信息和库存信息）则缓存1小时。

我们的应用程序也遵循该协议的第5点。我们希望主页上的产品链接到我们站点的详细信息页面，而且只有当完成交易时才链接到Amazon站点。

33.2.2　XML的解析：REST响应

Amazon为其Web服务提供的第一个最受欢迎的接口是通过REST实现的。这个接口可以接受一个普通的HTTP请求并且返回一个XML文档。要使用这个接口，需要解析Amazon返回给我们的XML响应。可以通过PHP的SimpleXML库来实现XML的解析。

33.2.3 在PHP中使用SOAP

另一种可以提供相同Web服务的接口是SOAP。要使用SOAP访问这些服务，我们需要使用许多不同的PHP SOAP库中的其中之一。PHP内置有SOAP库，但是由于该库并不是一直可用的，因此可以使用NuSOAP库。由于NuSOAP是用PHP编写的，它并不需要进行编译。它只是一个可以通过require_once()进行调用的文件。

在http://dietrich.ganx4.com/nusoap上，可以找到NuSOAP。NuSOAP可以在Lesser的开放源代码项目中找到，也就是说，我们可以在任何应用程序中使用它，包括非免费的应用程序。

33.2.4 缓存

正如我们前面提到的，亚马逊对开发人员提出的条件和约束之一就是必须缓存通过Web服务下载的数据。在我们的解决方案中，仍然需要找到一个保存和重复使用所下载数据的方法，当然必须在这些数据失效之前。

33.3 解决方案概述

对于这个项目，我们还将使用事件驱动的方法来编写代码，正如第29章和第30章所介绍的。在这个例子中，我们不会绘制系统流程图，因为系统中只出现非常少的几个界面，而且这些页面之间的链接是非常简单的。

如图33-1所示的是用户访问Tahuayo时看到的主页面。

Tahuayo.com - An Amazonian Tributary - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost/phpmysql4e/chapter33/index.php

tahuayo.com.

home | about | cart

Search For Books

Search Now

Your Cart $0.00

Details Checkout

Selected Categories

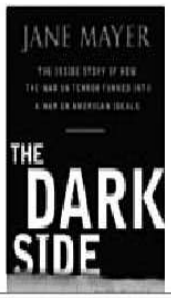
- Business & Investing
- Computers & Internet
- Food & Wine
- Literature and Fiction
- Non Fiction
- PHP
- Reference
- Romance
- Science
- Science Fiction
- Travel
- Web Development

Current Best Sellers in Non Fiction



Three Cups of Tea: One Man's Mission to Promote Peace . . . One School at a Time

by Greg Mortenson and David Oliver Relin
Published: 2007-01-30
$7.05 (list price ~~$15.00~~)
ISBN: 0143038257
Customer Rating: ★★★★★
Add to Cart Details



The Dark Side: The Inside Story of How The War on Terror Turned into a War on American Ideals

by Jane Mayer
Published: 2008-07-15
$14.97 (list price ~~$27.50~~)
ISBN: 0385526393

Done

图　33-1　Tahuayo的第一个页面显示了该站点的所有主要特性：目录浏览、搜索和购物车

可以看到，该站点的主要特性就是选中目录的显示以及显示这些目录中的物品。在默认情况下，第一页上显示当前销售情况最好的产品目录。如果一个用户点击其他目录，将看到该目录下相似的显示。

在我们进一步开始项目之前，需要掌握一些简单的术语：Amazon将目录当作是浏览节点。在贯穿整个代码和正式的文档中，我们都将发现这种表达方式。

文档提供了所有流行的浏览节点列表。此外，如果希望看到特定的节点，可以浏览Amazon站点并且从URL中读入它们，你可以在http://www.browsernodes.com/获得Browse Nodes资源。

奇怪的是，某些重要的目录，例如销售最佳的图书，无法通过浏览节点进行访问。

在这个页面的下方，还有许多图书和链接，但是无法从上图中看到（上图只是屏幕的截图）。我们在每一页上显示10本图书，同时显示链接到30多本相关图书。这种每页显示10本图书的设置是由Amazon确定的。而每页显示30本图书的设置是我们自己的选择。

从这个页面，用户可以点击并查看每一本图书的详细信息。这个页面如图33-2所示。

Tahuayo.com - An Amazonian Tributary - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost/phpmysql4e/chapter33/index.php?action=detail&ASIN=0143038257

tahuayo.com.

home | about | cart

Search For Books

Search Now

Your Cart: $0.00

Details Checkout

Selected Categories

- Business & Investing
- Computers & Internet
- Food & Wine
- Literature and Fiction
- Non Fiction
- PHP
- Reference
- Romance
- Science
- Science Fiction
- Travel
- Web Development



Add to Cart

Three Cups of Tea: One Man's Mission to Promote Peace . . . One School at a Time

by Greg Mortenson and David Oliver Relin
Published: 2007-01-30
Publisher: Penguin (Non-Classics)
Our Price: $7.05
List price: $15.00 **SAVE $7.95**
ISBN: 0143038257
Customer Rating: ★★★★★
Sales Rank: 13
Availability: Usually ships in 24 hours

Similar Products

- Three Cups of Tea
- The Difference a Day Makes: 365 Ways to Change Your World in Just 24 Hours by Karen M. Jones
- Catching Genius by Kristy Kiernan

Done

图　33-2　详细信息页面显示了关于特定图书的详细信息，同时还包括类似产品和概述

虽然无法在一个截图中显示整个页面，但是我们已经尽量显示了更多的内容，但是并不是全部的，因为在这个页面上显示的信息是一个heavy查询得来的。我们选择过滤了图书以外的产品，以及不适合图书目录的其他产品列表。

如果点击图书的封面，可以看到一个扩大后的图书封面。

我们可能已经注意到了位于上图右上方的搜索文本框。这个搜索操作可以对站点的关键字进行搜索，同样，它也可以通过Web服务接口搜索Amazon的目录。一个搜索操作的输出结果示例如图33-3所示。

Tahuayo.com - An Amazonian Tributary - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost/phpmysql4e/chapter33/index.php?action=search&search=batman

tahuayo.com.

home | about | cart

Search For Books

Search Now

Your Cart: $0.00

Details Checkout

Selected Categories

- Business & Investing
- Computers & Internet
- Food & Wine
- Literature and Fiction
- Non Fiction
- PHP
- Reference
- Romance
- Science
- Science Fiction
- Travel
- Web Development

Search Results For batman



Batman: The Dark Knight Returns

by Frank Miller
Published: 1997-05-01
$7.82 (list price ~~$14.99~~)
ISBN: 1563893428
Customer Rating: ★★★★½
Add to Cart Details



Batman: The Killing Joke

by Alan Moore and Brian Bolland
Published: 2008-03-19
$9.23 (list price ~~$17.99~~)
ISBN: 1401216676

Done

图　33-3　搜索batman的结果输出

虽然我们只是列出了少量的目录，但是客户可以通过这个搜索工具找到任何想要的图书，并且浏览特定的图书。

每一本图书都有一个“添加到购物车（Add to Cart）”的链接。点击购物车汇总中的这个链接或“详细信息（Details）”链接，将显示购物车中的详细内容，如图33-4所示。

Tahuayo.com - An Amazonian Tributary - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost/phpmysql4e/chapter33/index.php?action=addtocart&ASIN=0143038583

tahuayo.com.

home | about | cart

Search For Books

Search Now

Your Cart $33.13

The Omnivore's Dilemma: A Natural His...
Batman: The Killing Joke
Batman: The Dark Knight Returns

Details Checkout

Your Shopping Cart

Checkout When you have finished shopping press checkout to add all the items in your Tahuayo cart to your Amazon cart and complete your purchase.

Empty Cart If you have finished with this cart, you can empty it of all items.

Cart Contents



Batman: The Dark Knight Returns

by Frank Miller
Published: 1997-05-01
$7.82 (list price ~~$14.99~~)
Quantity in Cart: 2
Total Price: $15.64
ISBN: 1563893428
Customer Rating: ★★★★½

Delete Details



Batman: The Killing Joke

by Alan Moore and Brian Bolland
Published: 2008-03-19

Done

图　33-4　在购物车页面，可以删除物品，清空购物车或付账

最后，当用户点击其中任何一个“付账（Checkout）”链接，都会将购物车中的详细信息发送给Amazon并且转到Amazon站点。这样，客户就将看到如图33-5所示的页面。

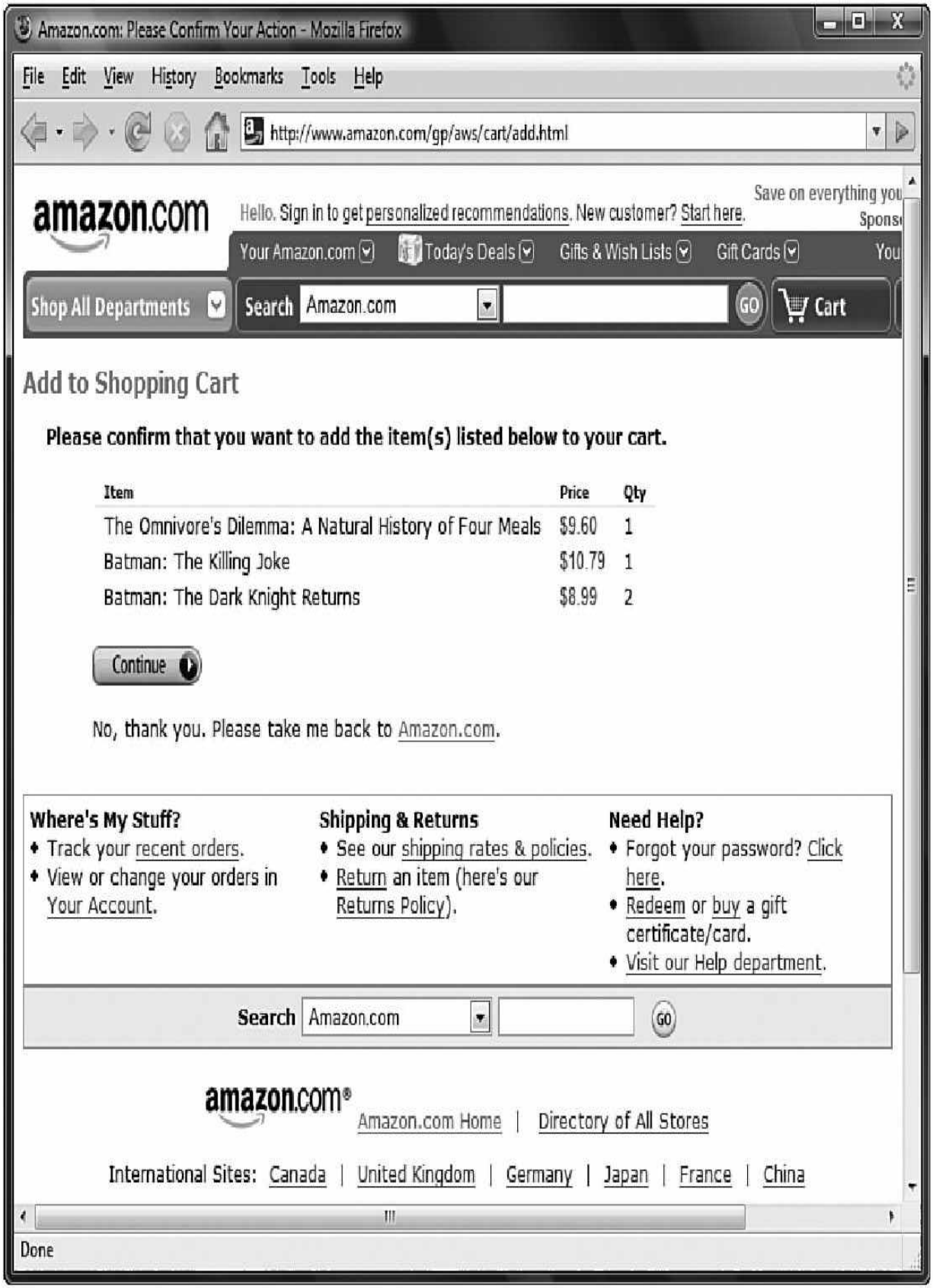

Amazon.com: Please Confirm Your Action - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://www.amazon.com/gp/aws/cart/add.html
amazon.com
Hello. Sign in to get personalized recommendations. New customer? Start here.
Save on everything you
Your Amazon.com
Today's Deals
Gifts & Wish Lists
Gift Cards
Shop All Departments
Search
Amazon.com
GO
Cart
Add to Shopping Cart
Please confirm that you want to add the item(s) listed below to your cart.
Item
Price
Qty
The Omnivore's Dilemma: A Natural History of Four Meals
$9.60
1
Batman: The Killing Joke
$10.79
1
Batman: The Dark Knight Returns
$8.99
2
Continue
No, thank you. Please take me back to Amazon.com.
Where's My Stuff?
Track your recent orders.
View or change your orders in Your Account.
Shipping & Returns
See our shipping rates & policies.
Return an item (here's our Returns Policy).
Need Help?
Forgot your password? Click here.
Redeem or buy a gift certificate/card.
Visit our Help department.
Search
Amazon.com
GO
amazon.com®
Amazon.com Home | Directory of All Stores
International Sites: Canada | United Kingdom | Germany | Japan | France | China
Done

图　33-5　在将客户选购物品保存在Amazon购物车之前，系统将确认该交易，并显示Tahuayo购物车中的所有物品

通过构建我们自己的前台应用程序和使用Amazon作为后台，可以理解本项目的含义。

由于这个项目仍然使用了事件驱动的方法，所以这个应用程序中的核心程序逻辑都是在一个文件中实现的——index.php。在该应用程序中，所用到的文件概述如表33-1所示。

表33-1 Tahuayo应用程序中用到的文件

文件名称	类 型	描 述
index.php	应用程序	主要应用程序文件
about.php	应用程序	显示“关于”页面
constants.php	包含文件	设置一些全局变量
topbar.php	包含文件	生成每一个页面和CSS之间的信息栏
bottom.php	包含文件	生成每一个页面的注脚
AmazonResultSet.php	类文件	包含保存每一个Amazon查询结果的PHP类
Product.php	类文件	包含保存特定图书信息的PHP类
bookdisplayfunctions.php	函数	包含用来显示图书和图书列表的函数
cachefunctions.php	函数	包含执行Amazon所需的缓存操作相关的函数
cartfunctions.php	函数	包含与购物车相关的函数
categoryfunctions.php	函数	包含用来检索和显示一个目录的函数
utilityfunctions.php	函数	包含在整个应用程序中要使用到的工具函数的集合

我们还需要前面所介绍的nusoap.php文件，因为在以上文件中，这个文件是必需的。NuSOAP文件可以在附带的文件中找到，具体目录位于chapter33中，但是也可以从http://dietrich.ganx4.com/nusoap/index.php中找到该文件的最新版本（如果发布了新版本）。

下面，我们开始了解核心应用程序index.php文件。

33.3.1 核心应用程序

程序清单33-3所示的就是核心应用程序index.php文件。

程序清单33-3 index.php——核心应用程序文件

```
<?php
//we are only using one session variable'cart'to store the cart contents
session_start();
require_once('constants.php');
require_once('Product.php');
require_once('AmazonResultSet.php');
require_once('utilityfunctions.php');
require_once('bookdisplayfunctions.php');
require_once('cartfunctions.php');
require_once('categoryfunctions.php');
//These are the variables we are expecting from outside.
//They will be validated and converted to globals
$external=array('action','ASIN','mode','browseNode','page','search');
//the variables may come via Get or Post
//convert all our expected external variables to short global names
foreach($external as$e){
if(@$_REQUEST[$e]){
$$e=$_REQUEST[$e];
}else{
$$e='';
}
$$e=trim($$e);
}
```

```
//default values for global variables
if($mode==''){
$mode='Books';//No other modes have been tested
}
if($browseNode==''){
$browseNode=53;//53 is bestselling non-fiction books
}
if($page==''){
$page=1;//First Page-there are 10 items per page
}
//validate/strip input
if(!eregi('^[A-Z0-9]+$',$ASIN)){
//ASINS must be alpha-numeric
$ASIN='';
}
if(!eregi('^[a-z]+$',$mode)){
//mode must be alphabetic
$mode='Books';
}
$page=intval($page);//pages and browseNodes must be integers
$browseNode=intval($browseNode);
//it may cause some confusion,but we are stripping characters out from
//$search it seems only fair to modify it now so it will be displayed
//in the heading
$search=safeString($search);
if(!isset($_SESSION['cart'])){
session_register('cart');
$_SESSION['cart']=array();
}
//tasks that need to be done before the top bar is shown
if($action=='addtocart'){
addToCart($_SESSION['cart'],$ASIN,$mode);
}
if($action=='deletefromcart'){
deleteFromCart($_SESSION['cart'],$ASIN);
}
```

```
if($action=='emptycart'){
$_SESSION['cart']=array();
}
//show top bar
require_once('topbar.php');
//main event loop.Reacts to user action on the calling page
switch($action){
case'detail':
showCategories($mode);
showDetail($ASIN,$mode);
break;
case'addtocart':
case'deletefromcart':
case'emptycart':
case'showcart':
echo"<hr/><h1>Your Shopping Cart</h1>";
showCart($_SESSION['cart'],$mode);
break;
case'image':
showCategories($mode);
echo"<h1>Large Product Image</h1>";
showImage($ASIN,$mode);
break;
case'search':
showCategories($mode);
echo"<h1>Search Results For".$search."</h1>";
showSearch($search,$page,$mode);
break;
case'browsenode':
default:
showCategories($mode);
$category=getCategoryName($browseNode);
if(!$category||($category=='Best Selling Books')){
echo"<h1>Current Best Sellers</h1>";
}else{
echo"<h1>Current Best Sellers in".$category."</h1>";
```

```
}
showBrowseNode($browseNode,$page,$mode);
break;
}
require('bottom.php');
```

下面，我们用自己的方法来了解这个文件。首先，我们创建了一个会话。就像前面所介绍的，将客户的购物车保存为一个会话变量。

接着，我们包括并引入了几个文件。这些文件都是下面将要介绍的函数，但是在介绍它们之前，我们必须先介绍第一个被包括进来的文件。constants.php文件定义了一些重要的常量，这些常量将在整个应用程序中使用。程序清单33-4给出了constants.php的所有代码。

程序清单33-4　constants.php——声明重要的全局常量和变量

```
<?php
//this application can connect via REST(XML over HTTP)or SOAP
//define one version of METHOD to choose.
//define('METHOD','SOAP');
define('METHOD','REST');
//make sure to create a cache directory an make it writable
define('CACHE','cache');//path to cached files
define('ASSOCIATEID','XXXXXXXXXXXXXX');//put your associate id here
define('DEVTAG','XXXXXXXXXXXXXX');//put your developer tag here
//give an error if software is run with the dummy devtag
if(DEVTAG=='XXXXXXXXXXXXXX'){
die("You need to sign up for an Amazon.com developer tag at
<a href=\"https://aws.amazon.com/\">Amazon</a>
when you install this software.You should probably sign up
for an associate ID at the same time.Edit the file constants.php.");
}
//(partial)list of Amazon browseNodes.
$categoryList=array(5=>'Computers&Internet',3510=>'Web Development',
295223=>'PHP',17=>'Literature and Fiction',
```

```
3=>'Business&Investing',53=>'Non Fiction',
23=>'Romance',75=>'Science',21=>'Reference',
6=>'Food&Wine',27=>'Travel',
16272=>'Science Fiction'
);
```

这个应用程序可以使用REST或SOAP进行开发。修改METHOD常量值，可以选择所使用的开发方法。

CACHE常量定义了保存我们从Amazon站点上下载的数据的路径。可以修改该常量值，使其指向系统上任何希望的地方。

ASSOCIATEID常量定义了会员ID。如果是在事务中发送会员ID，将得到现金奖励。修改该常量，使其保存会员ID。

DEVTAG常量定义了Amazon在注册时为我们分配的开发人员令牌。必须将其修改为我们自己的开发人员令牌，否则该应用程序将无法正常运行。可以在如下URL注册一个开发人员令牌：http://aws.amazon.com。

现在，让我们回头看看index.php文件。它包含一些初始设置，以及主要的事件循环。首先，我们将从通过GET或POST方法传递进来的$_REQUEST超级全局变量中获取所需的变量。接着，将为这些标准的全局变量设置默认值，这些全局变量可以确定以后的页面显示，如下所示：

```
//default values for global variables
if($mode==''){
$mode='Books';//No other modes have been tested
}
if($browseNode==''){
$browseNode=53;//53 is bestselling non-fiction books
}
if($page==''){
$page=1;//First Page-there are 10 items per page
```

在以上代码中，我们将mode变量设置为"books"。Amazon支持许多模式（产品的类型），但是对于这个应用程序，我们只考虑图书。修改本章代码使其适用于其他产品目录不会太困难，只要重新设置$mode变量就可以了。此外，可能还需要查看Amazon文档，确认对于非图书类产品是否还具有其他属性，同时从用户界面上去除只与图书相关的文字。

browseNode变量用来指定要显示的图书种类。如果用户点击了"Selected Categories"链接，就可以自动设置图书种类。如果还没有设置（例如，当用户第一次来到该站点）我们仍然需要对其进行设置，设置为53。Amazon的浏览节点都是整数，用整数来标识一个种类。53表示非科幻图书类别，该类别也是一个非常不错的节点，就像那些出现在初始页面的类别，虽然有些最好的常见目录（例如，最佳销售）无法通过浏览节点进行访问。

page变量用来告诉Amazon我们希望在给定的种类中要显示的结果子集。page 1包含了1～10的结果，page 2包含了11～20的结果等。Amazon可以设置一个页面上的结果数，我们不用对其进行控制。当然，也可以在一页上显示Amazon站点上两页或更多页上的数据，但是对Amazon和我们的站点来说，10是都可以接受的，而且这样可以使我们的站点与Amazon保持一致。

接下来，我们将整理接收到的输入数据，可以通过搜索文本框或者是GET和POST参数：

```
//validate/strip input
if(!eregi('^[A-Z0-9]+$',$ASIN)){
//ASINS must be alpha-numeric
$ASIN='';
}
if(!eregi('^[a-z]+$',$mode)){
//mode must be alphabetic
$mode='Books';
}
$page=intval($page);//pages and browseNodes must be integers
$browseNode=intval($browseNode);
```

```
//it may cause some confusion,but we are stripping characters out from
//$search it seems only fair to modify it now so it will be displayed
//in the heading
```

以上代码没有什么新内容。safeString()函数源自utilityfunctions.php函数库。它只是通过一个正则表达式的替换操作，从输入字符串中删除任何非字母字符。由于我们在前面已经介绍它，这里不再进行描述。

在应用程序中，我们对用户输入数据进行校验的主要原因在于，我们将使用客户的输入在缓存中创建文件名称。如果允许客户在输入中使用“..”或“/”，可能会遇到非常严重的问题。

接下来，如果客户还没有购物车的话，将为其设置一个购物车：

```
if(!isset($_SESSION['cart'])){
session_register('cart');
$_SESSION['cart']=array();
```

在页面最上方的信息栏中显示信息之前（参阅图33-1），还需要完成一些操作。购物车将出现在每一个页面最上方的信息栏中。因此，在显示购物车内容之前，保持购物车变量为最新的是非常重要的：

```
//tasks that need to be done before the top bar is shown
if($action=='addtocart'){
addToCart($_SESSION['cart'],$ASIN,$mode);
}
if($action=='deletefromcart'){
deleteFromCart($_SESSION['cart'],$ASIN);
}
if($action=='emptycart'){
$_SESSION['cart']=array();
```

在这里，我们将在显示购物车之前添加或删除一些物品。讨论购物车和付账时，我们还将介绍这些函数。如果想现在就了解这些函数，可以在cartfunctions.php文件中找到它们。现在，我们先将它们放到一边，因为首先必须理解Amazon的接口。

下一步，我们将包含并引入topbar.php文件。这个文件只包含了HTML和样式单，以及一个对ShowSmallCart()函数的调用（源自cartfunctions.php）。这个函数将显示购物车的总结信息，我们将在每个页面右上方看到它们。介绍购物车函数时，我们还将介绍这些函数。

最后，我们介绍事件处理的主循环。表33-2给出了可能出现的事件总结。

可以看到，表33-2中的前4个事件都与获取和显示信息相关。而后4个事件都与管理购物车相关。

从Amazon获取数据的事件具有相似的工作方式。我们将以browsenode（种类）事件为例介绍如何获取关于图书的数据。

表33-2 可能出现的事件主循环

事　件	描　述
browsenode	显示特定种类中的图书。这是默认的事件
detail	显示特定图书的详细信息
image	显示图书封面的图像
search	显示用户搜索结果
addtocart	将一个物品添加到用户的购物车中
deletefromcart	从购物车中删除一个物品
emptycart	清空购物车
showcart	显示购物车中的物品

33.3.2　显示特定种类的图书

当browsenode（查看一个种类）事件被执行时，将运行如下代码：

```
showCategories($mode);
$category=getCategoryName($browseNode);
if(!$category||($category=='Best Selling Books')){
echo"<h1>Current Best Sellers</h1>";
}else{
echo"<h1>Current Best Sellers in".$category."</h1>";
}
```

showCategories()函数将显示选中种类的列表，我们可以在页面的最上方看到这个列表。getCategoryName()函数返回给定browsenode号条件下的当前种类名称。showBrowseNode()函数将在一个页面中显示该目录下的图书。

下面，我们先来分析showCategories()函数。该函数代码如程序清单33-5所示。

程序清单33-5　categoryfunctions.php函数库中的showCategories()函数——种类列表

```
//display a starting list of popular categories
function showCategories($mode){
global$categoryList;
echo"<hr/><h2>Selected Categories</h2>";
if($mode=='Books'){
asort($categoryList);
$categories=count($categoryList);
$columns=4;
$rows=ceil($categories/$columns);
echo"<table border=\"0\"cellpadding=\"0\"cellspacing=\"0\"
```

```
width=\"100%\"><tr>";
reset($categoryList);
for($col=0;$col<$columns;$col++){
echo"<td width=\"".(100/$columns)."%\"valign=\"top\"><ul>";
for($row=0;$row<$rows;$row++){
$category=each($categoryList);
if($category){
$browseNode=$category['key'];
$name=$category['value'];
echo"<li><span class=\"category\">
<a href=\"index.php?action=browsenode&browseNode="
.$browseNode."\">".$name."</a></span></li>";
}
}
echo"</ul></td>";
}
echo"</tr></table><hr/>";
}
```

以上函数使用categoryList数组将browsenode号映射到种类名称，该数组在constants.php函数库中声明。所需的browsenode将被硬编码到这个数组中。该函数还将对这个数组进行排序，并且显示不同的种类。

在主事件循环中，接下来调用的getCategoryName()函数可以用来查询当前正在查看的browsenode名称，这样我们就可以在页面上显示一个标题，例如，“Current Best Sellers in Business & Investing（当前商业和投资领域中最热销的产品）”。它将在前面提到的categoryList数组中查询这个关键字。

在这里，真正有趣的地方是从调用showBrowseNode()函数开始的。这个函数如程序清单33-6所示。

程序清单33-6　bookdisplayfunctions.php函数库中的showBrowseNode()函数——种类列表

```
//For a particular browsenode,display a page of products
function showBrowseNode($browseNode,$page,$mode){
$ars=getARS('browse',array('browsenode'=>$browseNode,
'page'=>$page,'mode'=>$mode));
showSummary($ars->products(),$page,$ars->totalResults(),
$mode,$browseNode);
```

这个函数将实现两个功能。首先，它将调用cachefunctions.php函数库中的getARS()函数。getARS()函数将获得并返回一个AmazonResultSet对象（稍后将详细介绍这个对象）。接下来，该函数将调用bookdisplayfunctions.php函数库中的showSummary()函数来显示所获得的信息。

getARS()函数绝对是整个应用程序的关键部分。如果我们按部就班了解其他操作（查看详细信息、图像以及搜索）我们将发现这个函数经常用到。

33.3.3　获得一个AmazonResultSet类

现在，我们来仔细了解getARS()函数，如程序清单33-7所示。

程序清单33-7　cachefunctions.php函数库中的getARS()函数——一个查询的结果集

```
//Get an AmazonResultSet either from cache or a live query
//If a live query add it to the cache
function getARS($type,$parameters){
$cache=cached($type,$parameters);
if($cache){
//if found in cache
return$cache;
}else{
$ars=new AmazonResultSet;
if($type=='asin'){
$ars->ASINSearch(padASIN($parameters['asin']),$parameters['mode']);
}
if($type=='browse'){
$ars->browseNodeSearch($parameters['browsenode'],
$parameters['page'],$parameters['mode']);
}
if($type=='search'){
$ars->keywordSearch($parameters['search'],$parameters['page'],
$parameters['mode']);
}
cache($type,$parameters,$ars);
}
return$ars;
}
```

这个函数是用来驱动从Amazon获得数据的操作。它可以以两种方式完成数据获取：从缓存，或者从Amazon实时获得。由于Amazon要

求开发人员缓存已经下载的数据，因此这个函数将首先查看保存在缓存中的数据。我们将在稍后的内容详细介绍缓存。

如果我们还没有执行特定的查询操作，就必须从Amazon站点实时获得数据。通过创建一个AmazonResultSet类的实例，并且调用该类中对应于我们需要执行的查询的方法，可以获得实时数据。查询的类型是由$type参数决定的。以种类（或者浏览节点）查询为例，我们将browse传递给该参数——参阅程序清单33-6。如果我们希望执行一个针对特定图书的查询，就应该传递该图书的asin，而如果希望执行一个关键字查询，该参数就必须设置为search。

该参数的每一个不同值都将调用AmazonResultSet类中的不同方法。单个物品搜索将调用ASINSearch()方法。种类搜索将调用browseNodeSearch()方法。而关键字搜索将调用keywordSearch()方法。

下面，我们来仔细了解AmazonResultSet类。该类的所有代码如程序清单33-8所示。

程序清单33-8　AmazonResultSet.php——处理与Amazon站点连接的类

```
<?php
//you can switch between REST and SOAP using this constant set in
//constants.php
if(METHOD=='SOAP'){
include_once('nusoap/lib/nusoap.php');
}
//This class stores the result of queries
//Usually this is 1 or 10 instances of the Product class
class AmazonResultSet{
private$browseNode;
private$page;
private$mode;
private$url;
private$type;
private$totalResults;
```

```
private$currentProduct=null;
private$products=array();//array of Product objects
function products(){
return$this->products;
}
function totalResults(){
return$this->totalResults;
}
function getProduct($i){
if(isset($this->products[$i])){
return$this->products[$i];
}else{
return false;
}
}
//Perform a query to get a page full of products from a browse node
//Switch between XML/HTTP and SOAP in constants.php
//Returns an array of Products
function browseNodeSearch($browseNode,$page,$mode){
$this->Service="AWSECommerceService";
$this->Operation="ItemSearch";
$this->AWSAccessKeyId=DEVTAG;
$this->AssociateTag=ASSOCIATEID;
$this->BrowseNode=$browseNode;
$this->ResponseGroup="Large";
$this->SearchIndex=$mode;
$this->Sort='salesrank';
$this->TotalPages=$page;
if(METHOD=='SOAP'){
$soapclient=new nusoap_client(
'http://ecs.amazonaws.com/AWSECommerceService/AWSECommerceService.wsdl',
'wsdl');
$soap_proxy=$soapclient->getProxy();
$request=array('Service'=>$this->Service,
'Operation'=>$this->Operation,'BrowseNode'=>$this->BrowseNode,
'ResponseGroup'=>$this->ResponseGroup,'SearchIndex'=>
```

```
$this->SearchIndex,'Sort'=>$this->Sort,'TotalPages'=>
$this->TotalPages);
$parameters=array('AWSAccessKeyId'=>DEVTAG,
'AssociateTag'=>ASSOCIATEID,'Request'=>array($request));
//perform actual soap query
$result=$soap_proxy->ItemSearch($parameters);
if(isSOAPError($result)){
return false;
}
$this->totalResults=$result['TotalResults'];
foreach($result['Items']['Item']as$product){
$this->products[]=new Product($product);
}
unset($soapclient);
unset($soap_proxy);
}else{
//form URL and call parseXML to download and parse it
$this->url="http://ecs.amazonaws.com/onca/xml?".
"Service=".$this->Service.
"&Operation=".$this->Operation.
"&AssociateTag=".$this->AssociateTag.
"&AWSAccessKeyId=".$this->AWSAccessKeyId.
"&BrowseNode=".$this->BrowseNode.
"&ResponseGroup=".$this->ResponseGroup.
"&SearchIndex=".$this->SearchIndex.
"&Sort=".$this->Sort.
"&TotalPages=".$this->TotalPages;
$this->parseXML();
}
return$this->products;
}
//Given an ASIN,get the URL of the large image
//Returns a string
function getImageUrlLarge($ASIN,$mode){
foreach($this->products as$product){
if($product->ASIN()==$ASIN){
```

```
return$product->imageURLLarge();
}
}
//if not found
$this->ASINSearch($ASIN,$mode);
return$this->products(0)->imageURLLarge();
}
//Perform a query to get a products with specified ASIN
//Switch between XML/HTTP and SOAP in constants.php
//Returns a Products object
function ASINSearch($ASIN,$mode='books'){
$this->type='ASIN';
$this->ASIN=$ASIN;
$this->mode=$mode;
$ASIN=padASIN($ASIN);
$this->Service="AWSECommerceService";
$this->Operation="ItemLookup";
$this->AWSAccessKeyId=DEVTAG;
$this->AssociateTag=ASSOCIATEID;
$this->ResponseGroup="Large";
$this->IdType="ASIN";
$this->ItemId=$ASIN;
if(METHOD=='SOAP'){
$soapclient=new nusoap_client(
'http://ecs.amazonaws.com/AWSECommerceService/AWSECommerceService.wsdl',
'wsdl');
$soap_proxy=$soapclient->getProxy();
$request=array('Service'=>$this->Service,'Operation'=>
$this->Operation,'ResponseGroup'=>$this->ResponseGroup,
'IdType'=>$this->IdType,'ItemId'=>$this->ItemId);
$parameters=array('AWSAccessKeyId'=>DEVTAG,
'AssociateTag'=>ASSOCIATEID,'Request'=>array($request));
//perform actual soap query
$result=$soap_proxy->ItemLookup($parameters);
if(isSOAPError($result)){
return false;
```

```
}
$this->products[0]=new Product($result['Items']['Item']);
$this->totalResults=1;
unset($soapclient);unset($soap_proxy);
}else{
//form URL and call parseXML to download and parse it
$this->url="http://ecs.amazonaws.com/onca/xml?".
"Service=".$this->Service.
"&Operation=".$this->Operation.
"&AssociateTag=".$this->AssociateTag.
"&AWSAccessKeyId=".$this->AWSAccessKeyId.
"&ResponseGroup=".$this->ResponseGroup.
"&IdType=".$this->IdType.
"&ItemId=".$this->ItemId;
$this->parseXML();
}
return$this->products[0];
}
//Perform a query to get a page full of products with a keyword search
//Switch between XML/HTTP and SOAP in index.php
//Returns an array of Products
function keywordSearch($search,$page,$mode='Books'){
$this->Service="AWSECommerceService";
$this->Operation="ItemSearch";
$this->AWSAccessKeyId=DEVTAG;
$this->AssociateTag=ASSOCIATEID;
$this->ResponseGroup="Large";
$this->SearchIndex=$mode;
$this->Keywords=$search;
if(METHOD=='SOAP'){
$soapclient=new nusoap_client(
'http://ecs.amazonaws.com/AWSECommerceService/AWSECommerceService.wsdl',
'wsdl');
$soap_proxy=$soapclient->getProxy();
$request=array('Service'=>$this->Service,'Operation'=>
$this->Operation,'ResponseGroup'=>$this->ResponseGroup,
```

```
'SearchIndex'=>$this->SearchIndex,'Keywords'=>$this->Keywords);
$parameters=array('AWSAccessKeyId'=>DEVTAG,
'AssociateTag'=>ASSOCIATEID,'Request'=>array($request));
//perform actual soap query
$result=$soap_proxy->ItemSearch($parameters);
if(isSOAPError($result)){
return false;
}
$this->totalResults=$result['TotalResults'];
foreach($result['Items']['Item']as$product){
$this->products[]=new Product($product);
}
unset($soapclient);
unset($soap_proxy);
}else{
$this->url="http://ecs.amazonaws.com/onca/xml?".
"Service=".$this->Service.
"&Operation=".$this->Operation.
"&AssociateTag=".$this->AssociateTag.
"&AWSAccessKeyId=".$this->AWSAccessKeyId.
"&ResponseGroup=".$this->ResponseGroup.
"&SearchIndex=".$this->SearchIndex.
"&Keywords=".$this->Keywords;
$this->parseXML();
}
return$this->products;
}
//Parse the XML into Product object(s)
function parseXML(){
//suppress errors because this will fail sometimes
$xml=@simplexml_load_file($this->url);
if(!$xml){
//try a second time in case just server busy
$xml=@simplexml_load_file($this->url);
if(!$xml){
return false;
```

```
}
}
$this->totalResults=(integer)$xml->TotalResults;
foreach($xml->Items->Item as$productXML){
$this->products[]=new Product($productXML);
}
}
}
```

这个类是非常有用的。它以非常好的黑盒方式封装了与Amazon站点的接口。在这个类中，与Amazon站点的连接可以通过REST方法或SOAP方法来实现。所使用的方法通过在最开始的constants.php文件设置的全局常量METHOD来确定。

下面，让我们回到前面的种类搜索例子。我们按如下方式使用AmazonResultSet：

```
$ars=new AmazonResultSet;
$ars->browseNodeSearch($parameters['browsenode'],
$parameters['page'],
$parameters['mode']);
```

这个类没有构造函数，因此我们直接介绍browseNodeSearch()方法。这里，我们将向该方法传递3个参数：我们所感兴趣的browsenode号（相应地，例如，"Business & Investing" 或者 "Computers & Internet"）；表示要获得多少记录的页数；以及表示我们所感兴趣的商业类型的模式。该方法的代码如程序清单33-9所示。

程序清单33-9　browseNodeSearch()方法——执行一个种类查询

```
//Perform a query to get a page full of products from a browse node
//Switch between XML/HTTP and SOAP in constants.php
//Returns an array of Products
function browseNodeSearch($browseNode,$page,$mode){
```

```
$this->Service="AWSECommerceService";
$this->Operation="ItemSearch";
$this->AWSAccessKeyId=DEVTAG;
$this->AssociateTag=ASSOCIATEID;
$this->BrowseNode=$browseNode;
$this->ResponseGroup="Large";
$this->SearchIndex=$mode;
$this->Sort="salesrank";
$this->TotalPages=$page;
if(METHOD=='SOAP'){
$soapclient=new nusoap_client(
'http://ecs.amazonaws.com/AWSECommerceService/AWSECommerceService.wsdl',
'wsdl');
$soap_proxy=$soapclient->getProxy();
$request=array('Service'=>$this->Service,
'Operation'=>$this->Operation,'BrowseNode'=>$this->BrowseNode,
'ResponseGroup'=>$this->ResponseGroup,'SearchIndex'=>
$this->SearchIndex,'Sort'=>$this->Sort,'TotalPages'=>
$this->TotalPages);
$parameters=array('AWSAccessKeyId'=>DEVTAG,
'AssociateTag'=>ASSOCIATEID,'Request'=>array($request));
//perform actual soap query
$result=$soap_proxy->ItemSearch($parameters);
if(isSOAPError($result)){
return false;
}
$this->totalResults=$result['TotalResults'];
foreach($result['Items']['Item']as$product){
$this->products[]=new Product($product);
}
unset($soapclient);
unset($soap_proxy);
}else{
//form URL and call parseXML to download and parse it
$this->url="http://ecs.amazonaws.com/onca/xml?".
"Service=".$this->Service.
```

```
"&Operation=".$this->Operation.
"&AssociateTag=".$this->AssociateTag.
"&AWSAccessKeyId=".$this->AWSAccessKeyId.
"&BrowseNode=".$this->BrowseNode.
"&ResponseGroup=".$this->ResponseGroup.
"&SearchIndex=".$this->SearchIndex.
"&Sort=".$this->Sort.
"&TotalPages=".$this->TotalPages;
$this->parseXML();
}
return$this->products;
}
```

根据METHOD常量值的不同，这个方法将执行通过基于SOAP或REST的查询。但是，这两种请求中发送的信息是相同的。如下代码行显示的是请求的变量及其值：

```
$this->Service="AWSECommerceService";
$this->Operation="ItemSearch";
$this->AWSAccessKeyId=DEVTAG;
$this->AssociateTag=ASSOCIATEID;
$this->BrowseNode=$browseNode;
$this->ResponseGroup="Large";
$this->SearchIndex=$mode;
$this->Sort="salesrank";
$this->TotalPages=$page;
```

以上代码中的某些变量值是在应用的其他部分设置，例如，$browseNode、$mode和$page的变量值。其他值是常量，例如，DEVTAG和ASSOCIATEID.。而另外一部分的变量$this->Service和$this->Operation以及$this->Sort都是静态变量。

不同的请求类型对变量的要求是不同。以上示例用来浏览根据销售排名的特定节点。查询特定物品和关键字的变量是不同的。在

AmazonResultSet.php文件的browseNodeSearch()函数、ASINSearch()函数以及keywordSearch()函数中，你可以找到这些变量的列表。所有请求类型所需变量的详细信息可以在AWS开发人员指南找到。

接下来，我们分别介绍browseNodeSearch()函数中的REST和SOAP查询请求的创建。ASINSearch()函数和keywordSearch()函数的请求创建格式在概念上都是相似的。

33.3.4　使用REST发送和接收请求

在browseNodeSearch()函数（或ASINSearch()，keywordSearch()）设置了所需的类成员变量后，通过HTTP使用REST/XML还要做的就是格式化数据并发送给URL，如下所示：

```
$this->url="http://ecs.amazonaws.com/onca/xml?".
"Service=".$this->Service.
"&Operation=".$this->Operation.
"&AssociateTag=".$this->AssociateTag.
"&AWSAccessKeyId=".$this->AWSAccessKeyId.
"&BrowseNode=".$this->BrowseNode.
"&ResponseGroup=".$this->ResponseGroup.
"&SearchIndex=".$this->SearchIndex.
"&Sort=".$this->Sort.
"&TotalPages=".$this->TotalPages;
```

在上面的例子中，URL是http://ecs.amazonaws.com/onca/xml。对这个URL，你可以在URL字符串后附加变量名称及其值来组成一个GET查询字符串。关于相关信息以及其他可能变量的完整文档可以在AWS开发人员指南找到。在设置所有变量后，你可以调用：

```
$this->parseXML();
```

来完成实际的操作。parseXML()方法如程序清单33-10所示。

程序清单33-10　parseXML()方法——分析由查询返回的XML

```
//Parse the XML into Product object(s)
function parseXML(){
//suppress errors because this will fail sometimes
$xml=@simplexml_load_file($this->url);
if(!$xml){
```

```
//try a second time in case just server busy
$xml=@simplexml_load_file($this->url);
if(!$xml){
return false;
}
}
$this->totalResults=(integer)$xml->TotalResults;
foreach($xml->Items->Item as$productXML){
$this->products[]=new Product($productXML);
}
```

---

simplexml_load_file()函数完成了大部分的操作。它从一个文件读入XML内容，在这个例子中，是从一个URL读入的。它为数据和XML文档中的结构提供了一个面向对象接口。对数据来说，这是一个非常有用的接口，但是由于我们希望一个接口函数集能够处理来自REST或SOAP方法的数据，所以可以为Product类示例中的相同数据创建自己的面向对象接口。请注意，在REST版本中，可以将XML中的属性转换成PHP变量类型。在PHP中，不必使用cast操作符，但是在这个例子中，如果不给出cast操作符，将看到每一个数据的对象表示，这是没有用的。

Product类包含了大多数的访问器函数，这些函数可以用来访问保存在私有成员中的数据，因此打印整个文件并没有很大意义。Product类的结构和构造函数还是值得介绍的。程序清单33-11包含了Product类的部分定义。

程序清单33-11　Product类封装了一个Amazon产品的所有信息

```
class Product{
private$ASIN;
private$productName;
private$releaseDate;
private$manufacturer;
private$imageUrlMedium;
private$imageUrlLarge;
```

```
private$listPrice;
private$ourPrice;
private$salesRank;
private$availability;
private$avgCustomerRating;
private$authors=array();
private$reviews=array();
private$similarProducts=array();
private$soap;//array returned by SOAP calls
function__construct($xml){
if(METHOD=='SOAP'){
$this->ASIN=$xml['ASIN'];
$this->productName=$xml['ItemAttributes']['Title'];
if(is_array($xml['ItemAttributes']['Author'])!=""){
foreach($xml['ItemAttributes']['Author']as$author){
$this->authors[]=$author;
}
}else{
$this->authors[]=$xml['ItemAttributes']['Author'];
}
$this->releaseDate=$xml['ItemAttributes']['PublicationDate'];
$this->manufacturer=$xml['ItemAttributes']['Manufacturer'];
$this->imageUrlMedium=$xml['MediumImage']['URL'];
$this->imageUrlLarge=$xml['LargeImage']['URL'];
$this->listPrice=$xml['ItemAttributes']['ListPrice']['FormattedPrice'];
$this->listPrice=str_replace('$','',$this->listPrice);
$this->listPrice=str_replace(',','',$this->listPrice);
$this->listPrice=floatval($this->listPrice);
$this->ourPrice=$xml['OfferSummary']['LowestNewPrice']['FormattedPrice'];
$this->ourPrice=str_replace('$','',$this->ourPrice);
$this->ourPrice=str_replace(',','',$this->ourPrice);
$this->ourPrice=floatval($this->ourPrice);
$this->salesRank=$xml['SalesRank'];
$this->availability=
$xml['Offers']['Offer']['OfferListing']['Availability'];
$this->avgCustomerRating=$xml['CustomerReviews']['AverageRating'];
```

```
$reviewCount=0;
if(is_array($xml['CustomerReviews']['Review'])){
foreach($xml['CustomerReviews']['Review']as$review){
$this->reviews[$reviewCount]['Rating']=$review['Rating'];
$this->reviews[$reviewCount]['Summary']=$review['Summary'];
$this->reviews[$reviewCount]['Content']=$review['Content'];
$reviewCount++;
}
}
$similarProductCount=0;
if(is_array($xml['SimilarProducts']['SimilarProduct'])){
foreach($xml['SimilarProducts']['SimilarProduct']as$similar){
$this->similarProducts[$similarProductCount]['Title']=
$similar['Title'];
$this->similarProducts[$similarProductCount]['ASIN']=
$review['ASIN'];
$similarProductCount++;
}
}
}else{
//using REST
$this->ASIN=(string)$xml->ASIN;
$this->productName=(string)$xml->ItemAttributes->Title;
if($xml->ItemAttributes->Author){
foreach($xml->ItemAttributes->Author as$author){
$this->authors[]=(string)$author;
}
}
$this->releaseDate=(string)$xml->ItemAttributes->PublicationDate;
$this->manufacturer=(string)$xml->ItemAttributes->Manufacturer;
$this->imageUrlMedium=(string)$xml->MediumImage->URL;
$this->imageUrlLarge=(string)$xml->LargeImage->URL;
$this->listPrice=(string)$xml->ItemAttributes->ListPrice->FormattedPrice;
$this->listPrice=str_replace('$','',$this->listPrice);
$this->listPrice=str_replace(',','',$this->listPrice);
$this->listPrice=floatval($this->listPrice);
```

```
$this->ourPrice=(string)$xml->OfferSummary->LowestNewPrice->
FormattedPrice;
$this->ourPrice=str_replace('$','',$this->ourPrice);
$this->ourPrice=str_replace(',','',$this->ourPrice);
$this->ourPrice=floatval($this->ourPrice);
$this->salesRank=(string)$xml->SalesRank;
$this->availability=(string)$xml->Offers->Offer->OfferListing->
Availability;
$this->avgCustomerRating=(float)$xml->CustomerReviews->AverageRating;
$reviewCount=0;
if($xml->CustomerReviews->Review){
foreach($xml->CustomerReviews->Review as$review){
$this->reviews[$reviewCount]['Rating']=(float)$review->Rating;
$this->reviews[$reviewCount]['Summary']=(string)$review->Summary;
$this->reviews[$reviewCount]['Content']=(string)$review->Content;
$reviewCount++;
}
}
$similarProductCount=0;
if($xml->SimilarProducts->SimilarProduct){
foreach($xml->SimilarProducts->SimilarProduct as$similar){
$this->similarProducts[$similarProductCount]['Title']=
(string)$similar->Title;
$this->similarProducts[$similarProductCount]['ASIN']=
(string)$similar->ASIN;
$similarProductCount++;
}
}
}
}
//most methods in this class are similar
//and just return the private variable
function similarProductCount(){
return count($this->similarProducts);
}
function similarProduct($i){
```

```
return$this->similarProducts[$i];
}
function customerReviewCount(){
return count($this->reviews);
}
function customerReviewRating($i){
return$this->reviews[$i]['Rating'];
}
function customerReviewSummary($i){
return$this->reviews[$i]['Summary'];
}
function customerReviewComment($i){
return$this->reviews[$i]['Content'];
}
function valid(){
if(isset($this->productName)&&($this->ourPrice>0.001)&&
isset($this->ASIN)){
return true;
}else{
return false;
}
}
function ASIN(){
return padASIN($this->ASIN);
}
function imageURLMedium(){
return$this->imageUrlMedium;
}
function imageURLLarge(){
return$this->imageUrlLarge;
}
function productName(){
return$this->productName;
}
function ourPrice(){
return number_format($this->ourPrice,2,'.','');
```

```
}
function listPrice(){
return number_format($this->listPrice,2,'.','');
}
function authors(){
if(isset($this->authors)){
return$this->authors;
}else{
return false;
}
}
function releaseDate(){
if(isset($this->releaseDate)){
return$this->releaseDate;
}else{
return false;
}
}
function avgCustomerRating(){
if(isset($this->avgCustomerRating)){
return$this->avgCustomerRating;
}else{
return false;
}
}
function manufacturer(){
if(isset($this->manufacturer)){
return$this->manufacturer;
}else{
return false;
}
}
function salesRank(){
if(isset($this->salesRank)){
return$this->salesRank;
}else{
```

```
return false;
}
}
function availability(){
if(isset($this->availability)){
return$this->availability;
}else{
return false;
}
}
}
```

这个类的构造函数需要两个不同形式的输入数据并且能够创建一个应用程序接口。请注意，虽然某些处理代码可以更普通，但是根据方法的不同，某些重要的属性，例如评价，具有不同的名称。

了解获得数据的全部步骤后，我们回到getARS()函数，接着就是showBrowseNode()。下一步是：

```
showSummary($ars->products(),$page,
$ars->totalResults(),$mode,
$browseNode);
```

showSummary()函数只显示了AmazonResultSet的信息，如程序清单33-1所示的。因此，这里我们没有再次包含该函数。

33.3.5　使用SOAP发送和接收请求

下面，我们来了解SOAP版本的browseNodeSearch()函数。如下所示，我们将再次给出其代码：

```
$soapclient=new nusoap_client(
'http://ecs.amazonaws.com/AWSECommerceService/AWSECommerceService.wsdl',
'wsdl');
$soap_proxy=$soapclient->getProxy();
$request=array('Service'=>$this->Service,'Operation'=>$this->Operation,
'BrowseNode'=>$this->BrowseNode,'ResponseGroup'=>$this->ResponseGroup,
'SearchIndex'=>$this->SearchIndex,'Sort'=>$this->Sort,
'TotalPages'=>$this->TotalPages);
$parameters=array('AWSAccessKeyId'=>DEVTAG,'AssociateTag'=>ASSOCIATEID,
'Request'=>array($request));
//perform actual soap query
$result=$soap_proxy->ItemSearch($parameters);
if(isSOAPError($result)){
return false;
}
$this->totalResults=$result['TotalResults'];
foreach($result['Items']['Item']as$product){
$this->products[]=new Product($product);
}
unset($soapclient);
```

在以上代码中，没有其他特殊的函数——SOAP客户端为我们完成了所有“工作”。

我们以创建一个SOAP客户端实例为开始：

```
$soapclient=new nusoap_client(
'http://ecs.amazonaws.com/AWSECommerceService/AWSECommerceService.wsdl',
'wsdl');
```

这里，我们向客户端提供了两个参数。第一个就是服务的WSDL描述，而第二个参数可以告诉SOAP客户端这是一个WSDL URL。或者，我们也可以只提供一个参数：服务的终点，这是SOAP服务器的直接URL。

我们选择提供一个参数是有充分原因的，如下代码所示：

```
$soap_proxy=$soapclient->getProxy();
```

这一行代码将根据WSDL文档中的信息创建了一个类，这个类就是SOAP的代理，它拥有对应于Web服务中的方法。这可以使我们的工作变得简单一些。我们可以像与一个本地的PHP类进行交互一样与Web服务进行交互。

接下来，我们设置一个需要传递给browsenode查询的参数数组：

```
$request=array('Service'=>$this->Service,'Operation'=>$this->Operation,
'BrowseNode'=>$this->BrowseNode,'ResponseGroup'=>$this->ResponseGroup,
'SearchIndex'=>$this->SearchIndex,'Sort'=>$this->Sort,
'TotalPages'=>$this->TotalPages);
```

发送给该请求还需要其他两个元素：AWSAccessKeyID和and AssociateTag。这些元素以及$Request对象中的元素数组将保存在$parameters数组参数中。

```
$parameters=array('AWSAccessKeyId'=>DEVTAG,'AssociateTag'=>ASSOCIATEID,
'Request'=>array($request));
```

使用proxy类，你可以只调用Web服务的方法，将该参数数组传递给Web服务的方法：

```
$result=$soap_proxy->ItemSearch($parameters);
```

保存在$result中的数据是一个数组，我们可以将其直接保存为AmazonResultSet类的products数组中的一个Product对象。

33.3.6　缓存请求返回的数据

现在，我们回到getARS()函数，介绍一些关于缓存的内容。回忆一下，这个函数如下所示：

```
//Get an AmazonResultSet either from cache or a live query
//If a live query add it to the cache
function getARS($type,$parameters){
$cache=cached($type,$parameters);
if($cache){
//if found in cache
return$cache;
}else{
$ars=new AmazonResultSet;
if($type=='asin'){
$ars->ASINSearch(padASIN($parameters['asin']),$parameters['mode']);
}
if($type=='browse'){
$ars->browseNodeSearch($parameters['browsenode'],$parameters['page'],
$parameters['mode']);
}
if($type=='search'){
$ars->keywordSearch($parameters['search'],$parameters['page'],
$parameters['mode']);
}
cache($type,$parameters,$ars);
}
return$ars;
```

所有应用程序的SOAP缓存和XML缓存都是通过这个函数实现的。我们还有另一种缓存方法。首先，我们调用cached()方法判断所需的AmazonResultSet对象是否已经被缓存。如果已经被缓存，将直接返回数据，而不是生成一个新的请求：

```
$cache=cached($type,$parameters);
if($cache)//if found in cache{
return$cache;
}
```

如果没有被缓存，就必须从Amazon获得数据，然后再将其添加到缓存中：

```
cache($type,$parameters,$ars);
```

下面，让我们来看看这两个函数：cached()和cache()，如程序清单33-12所示。这些函数实现了Amazon所要求的缓存技术。

程序清单33-12　cached()和cache()函数——cachefunctions.php的缓存函数

```
//check if Amazon data is in the cache
//if it is,return it
//if not,return false
function cached($type,$parameters){
if($type=='browse'){
$filename=CACHE.'/browse.'.$parameters['browsenode'].'.'
.$parameters['page'].'.'.$parameters['mode'].'.dat';
}
if($type=='search'){
$filename=CACHE.'/search.'.$parameters['search'].'.'
.$parameters['page'].'.'.$parameters['mode'].'.dat';
}
if($type=='asin'){
$filename=CACHE.'/asin.'.$parameters['asin'].'.'
.$parameters['mode'].'.dat';
}
//is cached data missing or>1 hour old?
if(!file_exists($filename)||
```

```
((mktime()-filemtime($filename))>60*60)){
return false;
}
$data=file_get_contents($filename);
return unserialize($data);
}
//add Amazon data to the cache
function cache($type,$parameters,$data){
if($type=='browse'){
$filename=CACHE.'/browse.'.$parameters['browsenode'].'.'
.$parameters['page'].'.'.$parameters['mode'].'.dat';
}
if($type=='search'){
$filename=CACHE.'/search.'.$parameters['search'].'.'
.$parameters['page'].'.'.$parameters['mode'].'.dat';
}
if($type=='asin'){
$filename=CACHE.'/asin.'.$parameters['asin'].'.'
.$parameters['mode'].'.dat';
}
$data=serialize($data);
$fp=fopen($filename,'wb');
if(!$fp||(fwrite($fp,$data)==-1)){
echo('<p>Error,could not store cache file');
}
fclose($fp);
```

仔细查看以上代码，可以发现，缓存文件保存在一个由查询类型以及查询参数组成的文件名称的文件中。cache()函数通过对这些数据进行序列化操作，从而保存了这些数据。cached()函数也将在一个小时后覆盖这些数据，这是Amazon协议所规定的。

serialize()函数将所保存的程序数据转换成一个可以单独保存的字符串。在这个例子中，我们为AmazonResultSet对象创建了一个表示给对象的、可以用于保存的格式。调用unserialize()函数可以完成相反的

操作，它可以将保存的数据转换成内存中的数据结构。请注意，像这样的反序列化一个对象意味着必须拥有该类的文件定义，这样在该类被载入后，该类就可以被理解并使用。

在我们的应用程序中，从缓存获取结果只需要很短的时间，而执行一个实时查询则要十秒钟的时间。

33.3.7　创建购物车

到这里，我们了解了Amazon的所有查询功能，可以用它们来做些什么呢？很明显，我们可以使用这些功能来创建一个购物车。由于我们已经在第28章中详细介绍了它，因此在这里不再详细介绍。

购物车函数的代码如程序清单33-13所示。

程序清单33-13　cartfunctions.php——实现购物车

```
<?php
require_once('AmazonResultSet.php');
//Using the function showSummary()in the file bookdisplay.php display
//the current contents of the shopping cart
function showCart($cart,$mode){
//build an array to pass
$products=array();
foreach($cart as$ASIN=>$product){
$ars=getARS('asin',array('asin'=>$ASIN,'mode'=>$mode));
if($ars){
$products[]=$ars->getProduct(0);
}
}
//build the form to link to an Amazon.com shopping cart
echo"<form method=\"POST\"
action=\"http://www.amazon.com/gp/aws/cart/add.html\">";
foreach($cart as$ASIN=>$product){
$quantity=$cart[$ASIN]['quantity'];
echo"<input type=\"hidden\"name=\"ASIN.".$ASIN."\"
value=\"".$ASIN."\">";
echo"<input type=\"hidden\"name=\"Quantity.".$ASIN."\"
value=\"".$quantity."\">";
}
echo"<input type=\"hidden\"name=\"SubscriptionId\"
value=\"".DEVTAG."\">
```

```
<input type=\"hidden\"name=\"AssociateTag\"
value=\"".ASSOCIATEID."\">
<input type=\"image\"src=\"images/checkout.gif\"
name=\"submit.add-to-cart\"value=\"Buy
From Amazon.com\">
When you have finished shopping press checkout to add all
the items in your Tahuayo cart to your Amazon cart and
complete your purchase.
</form>
<br/><a href=\"index.php?action=emptycart\"><img
src=\"images/emptycart.gif\"alt=\"Empty Cart\"
border=\"0\"></a>
If you have finished with this cart,you can empty it
of all items.
</form>
<br/>
<h1>Cart Contents</h1>";
showSummary($products,1,count($products),$mode,0,true);
}
//show the small overview cart that is always on the screen
//only shows the last three items added
function showSmallCart(){
global$_SESSION;
echo"<table border=\"1\"cellpadding=\"1\"cellspacing=\"0\">
<tr><td class=\"cartheading\">Your Cart$".
number_format(cartPrice(),2)."</td></tr>
<tr><td class=\"cart\">".cartContents()."</td></tr>";
//form to link to an Amazon.com shopping cart
echo"<form method=\"POST\"
action=\"http://www.amazon.com/gp/aws/cart/add.html\">
<tr><td class=\"cartheading\"><a
href=\"index.php?action=showcart\"><img
src=\"images/details.gif\"border=\"0\"></a>";
foreach($_SESSION['cart']as$ASIN=>$product){
$quantity=$_SESSION['cart'][$ASIN]['quantity'];
echo"<input type=\"hidden\"name=\"ASIN.".$ASIN."\"
```

```
value=\"".$ASIN."\">";
echo"<input type=\"hidden\"name=\"Quantity.".$ASIN."\"
value=\"".$quantity."\">";
}
echo"<input type=\"hidden\"name=\"SubscriptionId\"
value=\"".DEVTAG."\">
<input type=\"hidden\"name=\"AssociateTag\"
value=\"".ASSOCIATEID."\">
<input type=\"image\"src=\"images/checkout.gif\"
name=\"submit.add-to-cart\"value=\"Buy From
Amazon.com\">
</td></tr>
</form>
</table>";
}
//show last three items added to cart
function cartContents(){
global$_SESSION;
$display=array_slice($_SESSION['cart'],-3,3);
//we want them in reverse chronological order
$display=array_reverse($display,true);
$result='';
$counter=0;
//abbreviate the names if they are long
foreach($display as$product){
if(strlen($product['name'])<=40){
$result.=$product['name']."<br/>";
}else{
$result.=substr($product['name'],0,37)."...<br/>";
}
$counter++;
}
//add blank lines if the cart is nearly empty to keep the
//display the same
for(;$counter<3;$counter++){
$result.="<br/>";
```

```
}
return$result;
}
//calculate total price of items in cart
function cartPrice(){
global$_SESSION;
$total=0.0;
foreach($_SESSION['cart']as$product){
$price=str_replace('$','',$product['price']);
$total+=$price*$product['quantity'];
}
return$total;
}
//add a single item to cart
//there is currently no facility to add more than one at a time
function addToCart(&$cart,$ASIN,$mode){
if(isset($cart[$ASIN])){
$cart[$ASIN]['quantity']+=1;
}else{
//check that the ASIN is valid and look up the price
$ars=new AmazonResultSet;
$product=$ars->ASINSearch($ASIN,$mode);
if($product->valid()){
$cart[$ASIN]=array('price'=>$product->ourPrice(),
'name'=>$product->productName(),'quantity'=>1);
}
}
}
//delete all of a particular item from cart
function deleteFromCart(&$cart,$ASIN){
unset($cart[$ASIN]);
}
```

在这里，我们实现购物车的方法与前面稍微有些不同，例如addToCart()函数。当我们向购物车添加一个物品时，可以检查它是否

具有一个有效的ASIN并且查找当前的价格（至少是缓存的价格）。

真正有趣的地方是这个问题：当客户付账时，如何让数据到达Amazon？

33.3.8 到Amazon付账

请仔细阅读程序清单33-13所示的showCart()函数。如下是相关的一部分：

```
//build the form to link to an Amazon.com shopping cart
echo"<form method=\"POST\"
action=\"http://www.amazon.com/gp/aws/cart/add.html\">";
foreach($cart as$ASIN=>$product){
$quantity=$cart[$ASIN]['quantity'];
echo"<input type=\"hidden\"name=\"ASIN.".$ASIN."\"
value=\"".$ASIN."\">";
echo"<input type=\"hidden\"name=\"Quantity.".$ASIN."\"
value=\"".$quantity."\">";
}
echo"<input type=\"hidden\"name=\"SubscriptionId\"
value=\"".DEVTAG."\">
<input type=\"hidden\"name=\"AssociateTag\"
value=\"".ASSOCIATEID."\">
<input type=\"image\"src=\"images/checkout.gif\"
name=\"submit.add-to-cart\"value=\"Buy
From Amazon.com\">
When you have finished shopping press checkout to add all
the items in your Tahuayo cart to your Amazon cart and
complete your purchase.
</form>";
```

付账按钮是一个表单按钮，它可以将购物车连接到客户在Amazon站点上的购物车。我们通过POST变量发送ASIN、数量和我们的会员ID。通过点击本章开始处的图33-5所示的每一个链接，可以看到最终结果。

这个接口的一个困难在于这是一个单向的交互。我们将物品添加到Amazon的购物车，但不能从Amazon购物车中删除物品。这就意味

着人们无法在我们的站点和Amazon站点之间来回浏览，并且删除购物车中重复的物品。

33.4 安装项目代码

如果希望安装本章的项目代码，必须进行一些必要的操作。将代码保存在服务器的合适位置后，我们还必须完成以下操作：

■创建一个缓存目录。

■设置这个缓存目录的读写权限，这样脚本就可以对其进行写操作。

■编辑constants.php代码，使其提供缓存的位置。

■注册一个Amazon开发人员令牌。

■编辑constants.php代码，使其包括开发人员令牌，此外，还可以选择是否包括会员ID。

■确认已经安装了NuSOAP。我们将其安装在Tahuayo目录下，但是也可以将其安装在不同位置，并对其进行修改。

■检查PHP5是否是在带有XML支持的条件下进行编译的。

33.5 扩展这个项目

你可以很容易对这个项目进行如下方面的扩展：扩展Tahuayo站点所提供的搜索类型。更多信息，请参阅Amazon的Web服务资源中心提供的关于创新性示例应用的链接以及“文章与教程”和“社区代码”部分。

购物车是使用Amazon Web服务的最基本功能，但它并不是唯一的功能。

33.6 进一步学习

关于XML和Web服务，互联网上有大量的图书和在线资源。通常，W3C是一个非常不错的起点，可以在该站点找到关于XML工作组页面，其URL如下所示：http://www.w3.org/XML/Core/。

以及Web服务事件页：http://www.w3.org/2002/ws/。

这里仅仅是一个开始。

第34章　使用Ajax构建Web 2.0应用

互联网是以一系列包含文本和图像、音频以及视频文件链接的静态页面组成。如今，尽管大多数页面都加入了通过服务器端脚本生成的文本和多媒体信息，但在很大程度上互联网还是处于这样的状况。这些通过服务器端脚本生成的文本和多媒体信息也就是本书所介绍的应用所创建的。但是，Web 2.0的出现可以引领开发人员探索与Web服务器和数据库进行用户交互的新方法，而正是这些Web服务器和数据库保存了我们所需要的信息。使用Ajax（异步JavaScript和XML）编程进行交互是如今日益流行的方法，它在减少读取静态元素时间的同时改进了交互性。

注意　要更好理解Web 2.0的概念，请参阅Tim O'Reilly的文章：http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html。

本章将介绍Ajax编程的基础以及可以集成到Web应用的Ajax示例元素。本章并不会全面介绍Ajax编程，只是为在未来工作中使用该技术提供一个基础。本章将包括如下：

■整合脚本和置标语言创建Ajax应用。

■一个Ajax应用的基本组成部分，包括发送请求以及解析来自服务器端的响应。

■如何修改前面章节介绍应用的元素来创建支持Ajax的页面。

■代码库以及如何需求帮助信息。

34.1　Ajax是什么

Ajax本身不是一个编程语言，也不是一个单一的技术。通常，Ajax编程是将处理XML格式数据传输的客户端JAVASCRIPT编程与某种服务器端编程语言（例如，PHP）的结合。此外，XHTML和CSS也被用来展示支持Ajax的元素。

典型的，Ajax编程的结果是为一个交互式应用提供更加清晰和快捷的用户接口————例如，连接到FACEBOOK，FLICKER以及其他处于Web 2.0前沿的社交网站的接口。这些应用支持用户不需要重新刷新或载入整个页面来执行许多任务，这也正是Ajax的用武之地。客户端的编程将调用许多服务器端的编程，但是只有用户浏览器所显示的特定区域才是会被从新刷新的。这种动作结果模拟了在单个应用才会产生的动作结果，但却是在Web环境下发生的。

一个常见的例子是，处理工作单的脱机应用和查看网站上信息丰富的表格。在脱机应用中，用户可以修改一个单元格并将公式应用到其他单元格，或者点击一个列进行排序，所有这些操作不需要离开最初的界面。在一个静态的Web环境中，点击一个链接对一列进行排序可能需要给服务器发送一个新请求，服务器返回一个新结果给浏览器，浏览器重新刷新该页面并展示给用户。在支持Ajax的Web环境中，表格可以根据用户的请求进行排序，不需要重新载入整个页面。

接下来的内容将介绍使用Ajax时需要涉及的不同技术。这些信息也不是全面的，文中将给出一些相关的资源。

34.1.1 HTTP请求和响应

超文本传输协议，或者HTTP，是一个Internet标准，它定义了Web服务器和Web浏览器之间相互交流的方式。当用户在Web浏览器的地址栏输入一个URL请求Web页面，或者点击一个链接，提交一个表单，或者执行任何能将用户带到新页面的时候，浏览器将发送一个HTTP请求。

这个请求将发送给Web服务器，服务器将返回许多可能响应的一种。要获得来自Web服务器的可理解响应，必须正确地创建请求。当使用Ajax的时候，了解请求和相应的格式是非常关键的，因为开发人员必须在Ajax应用中编写正确的请求以及期望特定的返回结果。

当发送HTTP请求时，客户端将用以下格式发送信息。

■开始行，包含了发送方法，资源路径以及使用的HTTP版本，如下示例：

```
GET http://server/phpmysql4e/chapter34/test.html HTTP/1.1
```

其他常见的方法包括POST和HEAD。

■可选的报头行，以“参数:值”格式出现，如下示例：

```
User-agent:Mozilla/5.0(Windows;U;Windows NT 6.0;en-US;rv:1.9.0.1)
Gecko/2008070208 Firefox/3.0.1
```

以及/或者：

```
Accept:text/plain,text/html
```

关于HTTP报头，请参阅：http://www.w3.org/Protocols/rfc2616/。

■空白行

■可选的消息正文（Message Body）

发送HTTP请求后，客户端将接收到一个HTTP响应。

HTTP响应的格式如下所示：

■开始行，或者状态行，包含了所使用的HTTP版本以及响应码，如下示例：

```
HTTP/1.1 200 OK
```

状态码的第一个数字（这里是200中的2）给出响应的类型。以1为开始的状态码表示响应为信息性的，2表示请求成功，3表示请求被重定向，4表示客户端错误，例如404表示客户端请求缺少相关内容项，5表示服务器端错误，例如，500表示非正常的脚本。

关于HTTP状态码的完整列表，请参阅：http://www.w3.org/Protocols/rfc2616/。

■可选的报头行，以“参数:值”格式出现，如下示例：

```
Server:Apache/2.2.9
Last-Modified:Fri,1 Aug 2008 15:34:59 GMT
```

34.1.2 DHTML和XHTML

动态HTML，或者DHTML，是一个用来描述将静态HTML，级联样式单（CSS）以及JavaScript结合，并且在载入一个静态Web页面所有元素后，通过文档对象模型修改页面外观的术语。表面上看，这个功能非常类似支持Ajax的站点页面，事实上，在一定程度的确如此。它们的区别就在于服务器和客户端之间的异步连接，也就是，Ajax中的字母"A"（异步）。

尽管DHTML驱动的站点可以在导航的下拉列表显示动态变化或者根据用户选择改变表单元素，但是这些元素的所有数据都是已经从服务器获得的。例如，如果要设计一个用户将鼠标滑过一个链接或按钮时显示文本第一部分并且滑过另一个链接或按钮时显示文本第二部分的DHTML网站，第一部分和第二部分的文本事实上已经载入到用户的浏览器中。开发人员可以通过JavaScript编程根据用户鼠标的动作设置CSS的可见性（Visibility）属性来显示或隐藏文本。在支持Ajax的站点中，根据远程调用服务器端的脚本执行结果，填充为第一部分和第二部分预留的区域，而页面的其他区域保持不变。

可扩展的超文本置标语言，或者XHTML，通过客户端设备（Web浏览器，手机或者其他手持设备）来显示标记的内容，并且支持集成CSS来提供显示的额外控制。这些功能与HTML和DHTML是类似的。XHTML和HTML的不同之处在于XHTML遵循XML的语法，以及除了标准的Web浏览工具外，XHTML还可以被XML工具来解析。

XHMTL的元素和属性全部是小写字母（例如，＜head＞＜/head＞，而不是＜HEAD＞＜/HEAD＞；href而不是HREF）。此外，所有属性值必须封闭在单引号或双引号内，所有元素必须显式封闭——通过标记对里的结束标记或者在单一元素内，例如，＜img/＞或＜br/＞。

关于XHTML的详细信息，请参阅：http://www.w3.org/TR/xhtml1/。

34.1.3 级联样式单

级联样式单（CSS）用来进一步定义静态、动态以及支持Ajax页面的显示。使用CSS允许开发人员修改一个文档内部（该样式单）的标记、类以及ID的定义，从而使该修改立即在所有连接到该样式单的页面体现。通过选择器（Selector），声明（Declaration）以及值（Value），这些定义，或者规则遵循如下特定格式：

■选择器是HTML标记的名称，例如body或h1（标题级别1）。

■声明是样式单本身的属性，例如background或font-size。

■值与声明相对应，例如white或12pt。

如下所示的是一个样式单示例，它定义了文档正文背景颜色为白色，文档内所有文本为常规粗细，12磅大小，Verdana或sans-serif字体：

```
body{
background:white[or#fff or#ffffff];
font-family:Verdana,sans-serif;
font-size:12pt;
font-weight:normal;
}
```

当要展示该页面的某个元素，如果该元素的样式在该样式单定义，这些值就会产生效果。例如，当遇到h1，客户端将显示h1文本，然而，h1的样式已经被定义——可能使用大于12pt的字体大小并且使用Bold属性。

除了定义选择器，也可以在样式单中定义自定义的类和ID。使用类（可以在一个页面的多个元素使用）或ID（只能在一个页面中使用一次），还可以进一步定义站点内所显示元素的外观以及功能。这种重定义对于支持Ajax的站点尤为重要，因为文档的预定义区域可以显示通过远程脚本动作获取的新信息。

类的定义类似于选择器的定义——名称定义后需要花括号，定义内容需要分号间隔。如下所示的是ajaxarea类的定义：

```
.ajaxarea{
width:400px;
height:400px;
background:#fff;
border:1px solid#000;
}
```

在该例中，当将该类应用到div容器中，将产生一个400像素（宽）×400像素（高）的矩形，其背景颜色为白色，并具有细的黑边框。其使用如下所示：

```
<div class="ajaxarea">some text</div>
```

使用样式单的最常见方法是创建一个包含了所有样式定义的单独文件，将该文件链接到HTML文档，如下所示：

```
<head>
<link rel="stylesheet"href="the_style_sheet.css"type="text/css">
</head>
```

关于CSS的详细信息，请参阅：http://www.w3.org/TR/CSS2/。

34.1.4 客户端编程

客户端编程通常在整个页面从Web服务器获取之后，在客户端Web浏览器中发生。所有的程序函数都包含在从Web服务器获取的数据中，等待客户端的调用。在客户端，常见的动作包括显示或隐藏部分的文本或图像，修改文本或图像的颜色、大小或位置，执行计算或者在用户将表单发送给Web服务器处理之前验证用户的表单输入。

最常见的客户端脚本编程语言是JavaScript——Ajax中的字母"J"。VBScript也是一种客户端脚本编程语言，但它是微软特有的，因此它并不是针对所有操作系统和Web浏览器的好选择。

34.1.5 服务器端编程

服务器端编程包括所有存在于Web服务器并且在给客户端发送响应之前需要解释或编译的脚本。通常，服务器端编程包括服务器端与数据库的连接；因此，与数据库之间的请求和响应也成为脚本本身一部分。

这些脚本可以用任何服务器端语言编写，例如，Perl、JSP、ASP或PHP——显然，本章将使用PHP作为服务器端编程的语言。通常，由于服务器端的响应是在某些标准HTML的标记区域显示数据，因此需要考虑终端用户的环境。

34.1.6 XML和XSLT

在本书的第33章介绍了XML，其中涵盖了XML的格式、结构和使用。在Ajax应用的上下文中，XML就是Ajax的字母"x"，它主要用于交换数据；XSLT用来操作数据。数据本身将通过所创建的Ajax应用发送或获取。

关于XML的详细信息，请参阅：http://www.w3.org/XML/。关于XSLT的详细信息，请参阅：http://www.w3.org/TR/xslt20/。

34.2 Ajax基础

到这里，已经了解了一个Ajax应用的可能组成部分，本节将使用这些组成部分创建一个可用的示例程序。请记住，使用Ajax的一个主要原因：创建一个能够响应用户动作的交互式网站，而这些响应并不需要刷新整个页面。

要实现此功能，Ajax应用需要包括一个能够处理发生在所请求的Web页面与负责生成该页面的Web服务器之间额外层。这个额外层通常就是指Ajax框架（也就是Ajax引擎）。该框架用来处理终端用户和Web服务器之间的请求，它可以不需要额外的动作就交流请求和响应，例如重新刷新页面，或者中断用户正在执行的动作（例如，滚动、点击或阅读一段文本）。

接下来的内容将介绍使用Ajax应用的不同组成部分创建一个流畅的用户体验。

34.2.1 XMLHTTPRequest对象

本章前面的内容已经介绍了HTTP请求和响应以及如何在一个Ajax应用中使用客户端编程。当连接Web服务器以及发送一个不需要重新载入整个原始页面的请求时，XMLHTTPReuqest对象是非常重要的，它也是JavaScript的特有对象。

注意　由于安全的原因，该对象只能调用同一域的URL；不能直接调用远程服务器。

XMLHTTPRequest对象也被看作是Ajax应用的“通道（guts）”，因为它是客户端请求和服务器端响应的“通道”。虽然接下来将介绍创建和使用XMLHTTPRequest对象的基础知识，请参阅http://www.w3.org/TR/XMLHttpRequest/获取详细信息。

XMLHTTPRequest对象具有一些属性，如表34-1所示。

表34-1 XMLHTTPRequest对象的属性

属 性	描 述
Onreadystatechange	指定当readyState属性变化时应该调用的函数
readyState	请求的状态。整数0表示未初始化，1表示正在载入，2表示已载入，3表示交互，4表示已完成
responseText	包含以字符串形式返回的数据
responseXml	包含以XML格式文档对象返回的数据
status	服务器返回的HTTP状态码，例如200
statusText	服务器返回的HTTP状态字符串，例如OK

XMLHTTPRequest对象具有一些方法，如表34-2所示。

表34-2 XMLHTTPRequest对象的方法

方 法	描 述
abort()	停止一个请求
getAllResponseHeaders()	以字符串的形式返回响应的所有报头
getResponseHeader(header)	以字符串的形式返回响应的报头“header”的值
open('method', 'URL', 'a')	指定HTTP请求的方法（参数值包括POST、GET和HEAD），目标URL以及该请求是否为异步（参数值为“true”或“false”）
send(content)	使用POST方法发送该请求，参数为可选
setRequestHeader('x', 'y')	设置一个参数对（x为参数名，y为参数值），并且作为报头参数发送请求

在使用XMLHTTPRequest对象之前，必须创建该对象的实例。如下语句所示：

```
var request=new XMLHTTPRequest();
```

尽管以上代码可以在非IE的浏览器中工作，但是理想情况下，该代码应该可以支持所有用户。因此，如下所示的代码可以在所有浏览器下创建XMLHTTPRequest对象的实例：

```
function getXMLHTTPRequest(){
var req=false;
try{
/*for Firefox*/
req=new XMLHttpRequest();
}catch(err){
try{
/*for some versions of IE*/
req=new ActiveXObject("Msxml2.XMLHTTP");
}catch(err){
try{
/*for some other versions of IE*/
req=new ActiveXObject("Microsoft.XMLHTTP");
}catch(err){
req=false;
}
}
}
return req;
}
```

如果将以上代码保存在一个文件ajax_functions.js中并且将其保存在Web服务器上，这样就创建了一个Ajax函数库。

当需要在Ajax应用中创建一个XMLHTTPRequest对象的实例，可以在代码中引入（include）包含所需函数的文件。

```
<script src="ajax_functions.js"type="text/javascript"><script>
```

接下来调用这个新对象以及继续编写应用的代码，如下所示：

```
<script type="text/javascript">
var myReq=getXMLHTTPRequest();
</script>
```

接下来的内容将介绍在Ajax函数文件中添加新的函数。

34.2.2 与服务器通信

在前面内容介绍的例子中，它的主要功能是创建一个新的XMLHTTPRequest对象；还没有执行任何使用该对象进行真正通信的操作。在接下来的例子中，将创建一个能够向服务器发送请求的JavaScript函数，具体地说是向servertime.php脚本发送请求。

```
function getServerTime(){
var thePage='servertime.php';
myRand=parseInt(Math.random()*999999999999999);
var theURL=thePage+"?rand="+myRand;
myReq.open("GET",theURL,true);
myReq.onreadystatechange=theHTTPResponse;
myReq.send(null);
}
```

函数的第一行创建一个名为thePage的变量，变量值为servertime.php。这是位于服务器端的脚本文件名称。

第二行代码作用不重要，因为它只是创建一个随机数。可能会有问题“一个随机数与服务器时间有什么关系？”答案是它对脚本本身并没有任何直接的影响。创建随机数并将其附加在URL中（如第三行代码所示）的原因是避免浏览器（或者代理服务器）缓存该请求带来的任何问题。如果URL只是http://yourserver/yourscript.php，服务器返回的结果可能会被缓存。然而，如果URL是http://yourserver/yourscript.php?rand=randval，浏览器或代理服务器就没法缓存，因为每次的URL都是不同的，而函数的实际功能是没有变化的。

该函数的最后三行代码使用了XMLHTTPRequest对象实例的三个方法（open，onreadystatechange和send），该XMLHTTPRequest对象是通过前面示例的getXMLHTTPRequest()方法获得。

在调用open()方法的代码行中，该方法的参数是请求类型（"GET"），URL（"theURL"），以及"true"表示该请求为异步的。

在调用onreadystatechange()方法的代码行中，当对象状态发生变化时，该函数调用一个新的函数——theHTTPResponse()。

在调用send()方法的代码行中，该函数向服务器端脚本发送了空的内容（NULL）。

到这里，创建一个名为servertime.php的文件，并且输入如程序清单34-1所示代码：

程序清单34-1　servertime.php代码

```
<?php
header('Content-Type:text/xml');
echo"<?xml version=\"1.0\"?>
<clock>
<timestring>It is".date('H:i:s')."on".date('M d,Y').".</timestring>
</clock>";
?>
```

该脚本将通过PHP的date()函数的调用获得服务器的当前时间，并且以XML编码的字符串返回。需要注意的是，date()函数被调用了两次；一次是date('H:i:s')，它将以24小时格式返回服务器当前时间的小时，分钟和秒。而以date('M d,Y')格式调用将返回脚本被调用时的月份、日期和年份。

返回结果的XML字符串如下示例所示，其中方括号中的字符将被真实值所代替：

```
<?xml version="1.0"?>
<clock>
<timestring>
It is[time]on[date].
```

```
</timestring>
</clock>
```

接下来的内容将介绍如何创建剩下的函数——theHTTPResponse()，并且通过服务器上的脚本对响应进行一些处理。

34.2.3 处理服务器响应

前面介绍的getServerTime()函数可以调用theHTTPResponse()函数并且对返回的字符串数据进行一些操作。如下例将解释响应并向终端用户显示一个字符串：

```
function theHTTPResponse(){
if(myReq.readyState==4){
if(myReq.status==200){
var timeString=
myReq.responseXML.getElementsByTagName("timestring")[0];
document.getElementById('showtime').innerHTML=
timeString.childNodes[0].nodeValue;
}
}else{
document.getElementById('showtime').innerHTML=
'<img src="ajax-loader.gif"/>';
}
}
```

函数中的if...else语句将检查对象的状态；如果对象状态不是4（已完成），函数将显示一段动画（(）。然而，如果myReq的readState是4，将进一步检查来自服务器的状态是不是200（OK）。

如果状态码是200，将创建一个新的变量：timeString。该变量将被赋值为从服务器端脚本返回的XML字符串中的timestring元素值，而该元素值是通过调用myReq对象的responseXML属性的getElementByTagname()方法获得，如下代码所示：

```
var timeString=myReq.responseXML.getElementsByTagName("timestring")[0];
```

下一步就是在HTML文件的CSS预定义区域显示该值。在这个例子中，时间值将显示在名为showtime的文档元素中：

```
document.getElementById('showtime').innerHTML=
timeString.childNodes[0].nodeValue;
```

到这里，就完成了ajax_functions.js脚本，如程序清单34-2所示。

程序清单34-2　ajax_functions.js代码

```
function getXMLHTTPRequest(){
var req=false;
try{
/*for Firefox*/
req=new XMLHttpRequest();
}catch(err){
try{
/*for some versions of IE*/
req=new ActiveXObject("Msxml2.XMLHTTP");
}catch(err){
try{
/*for some other versions of IE*/
req=new ActiveXObject("Microsoft.XMLHTTP");
}catch(err){
req=false;
}
}
}
return req;
}
function getServerTime(){
var thePage='servertime.php';
myRand=parseInt(Math.random()*999999999999999);
var theURL=thePage+"?rand="+myRand;
```

```
myReq.open("GET",theURL,true);
myReq.onreadystatechange=theHTTPResponse;
myReq.send(null);
}
function theHTTPResponse(){
if(myReq.readyState==4){
if(myReq.status==200){
var timeString=
myReq.responseXML.getElementsByTagName("timestring")[0];
document.getElementById('showtime').innerHTML=
timeString.childNodes[0].nodeValue;
}
}else{
document.getElementById('showtime').innerHTML=
'<img src="ajax-loader.gif"/>';
}
}
```

接下来的内容将完成该HTML并且结合所有部分创建一个Ajax应用。

34.2.4　整合应用

本章前面已经介绍，Ajax是一个整合的技术。在上一节，已经介绍了如何使用JavaScript和PHP（客户端和服务器端编程）发送HTTP请求并获取响应。没有介绍的部分是关于页面的显示：使用XHTML和CSS产生用户所见到的结果。

程序清单34-3给出了ajaxServerTime.html的代码，该文件包含了样式单以及能够调用远程PHP脚本并获取服务器响应的JavaScript脚本。

程序清单34-3　ajaxServerTime.html的代码

```
<!DOCTYPE html PUBLIC"-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"xml:lang="en"dir="ltr"lang="en">
<head>
<style>
body{
background:#fff;
font-family:Verdana,sans-serif;
font-size:12pt;
font-weight:normal;
}
.displaybox{
width:300px;
height:50px;
background-color:#ffffff;
border:2px solid#000000;
line-height:2.5em;
margin-top:25px;
font-size:12pt;
font-weight:bold;
}
</style>
<script src="ajax_functions.js"type="text/javascript"></script>
```

```
<script type="text/javascript">
var myReq=getXMLHTTPRequest();
</script>
</head>
<body>
<div align="center">
<h1>Ajax Demonstration</h1>
<p align="center">Place your mouse over the box below
to get the current server time.<br/>
The page will not refresh;only the contents of the box
will change.</p>
<div id="showtime"class="displaybox"
onmouseover="javascript:getServerTime();"></div>
</div>
</body>
</html>
```

以上代码以XHTML声明为开始，后面跟着 < html > 和 < head > 标记。在文档的head区域，插入了以 < style > < /style > 封闭的样式单定义。这里只定义了两项：boday标记内所有内容的格式以及使用displaybox类的元素格式。Displaybox类定义为300像素宽，50像素高的白色背景矩形，并且带有黑色边框。此外，该矩形内所有内容将是12pt的粗体字。

定义样式单内容后，在head元素中给出指向JavaScript函数库的链接。

```
<script src="ajax_functions.js"type="text/javascript"></script>
```

如下所示代码创建了一个新XMLHTTPRequest对象并赋值给myReq变量：

```
<script type="text/javascript">
var myReq=getXMLHTTPRequest();
```

```
</script>
```

这样就结束了head元素的定义，开始定义body元素。在body元素内，只是给出了XHTML内容。在一个对齐属性为居中的div元素中，可以找到该页面的标题文本（"Ajax Demonstration"）以及指导用户将鼠标移至文本框来获得服务器当前时间的信息。

在div元素中的showtime属性是真正定义了脚本应该执行的动作，具体的是，onmouseover事件处理器。

```
<div id="showtime"class="displaybox"
onmouseover="javascript:getServerTime();"></div>
```

onmouseover事件处理器的使用意味着，当用户鼠标进入名为showtime的div元素，调用getServerTime()函数。调用该函数将初始化发送给服务器的请求，服务器响应以及出现在div元素中的结果文本。

注意　JavaScript函数可以通过多种方法调用，例如，通过表单按钮的onclick事件。

图34-1、图34-2以及图34-3所示的是脚本实际执行时这些事件的调用顺序。ajaxServerTime.html页面的内容不会重新载入，只有showtime这个div元素中的内容将重新载入。

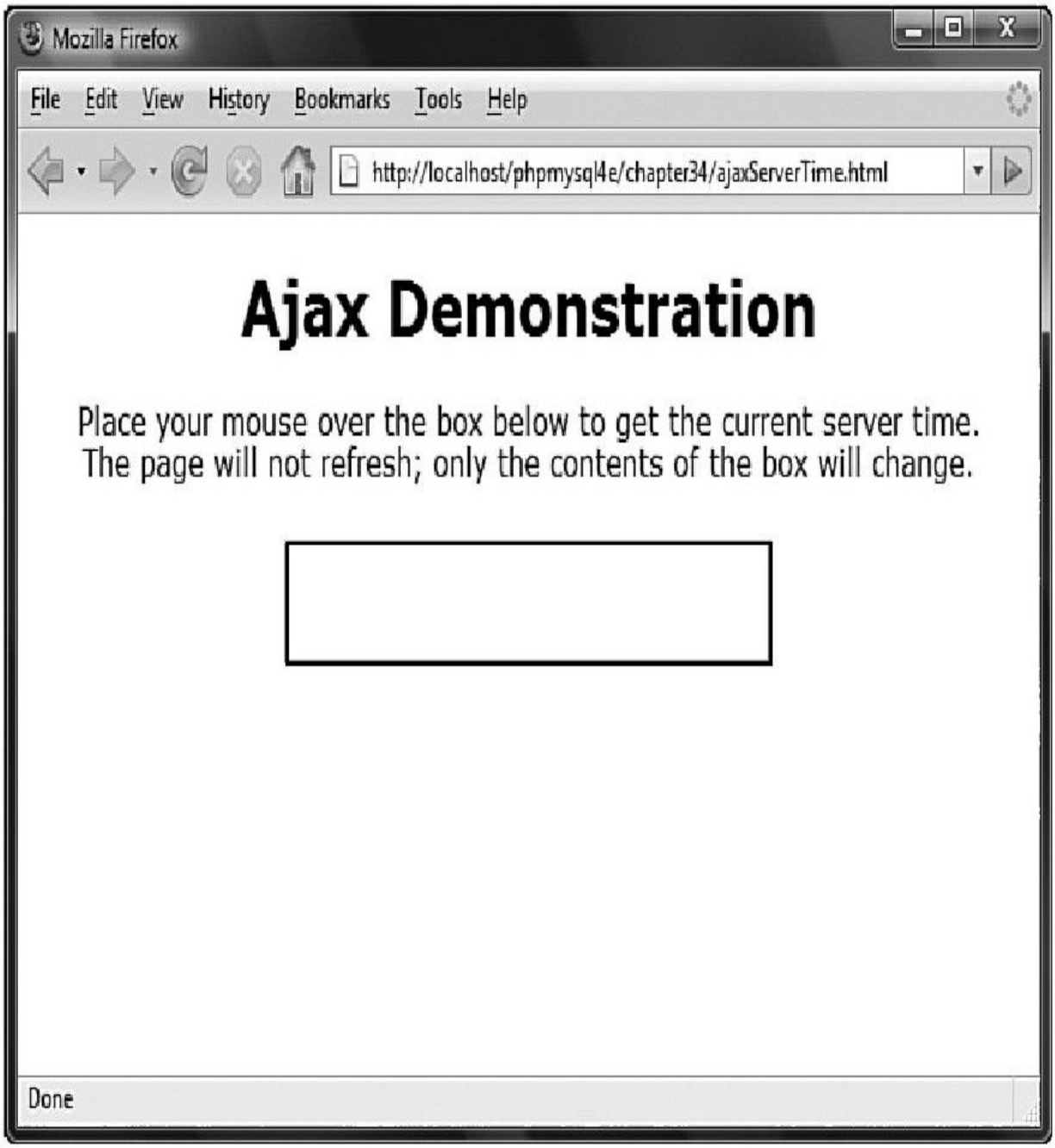


图　34-1　最初载入ajaxServerTime.html页面所显示的指示信息以及空白文本框

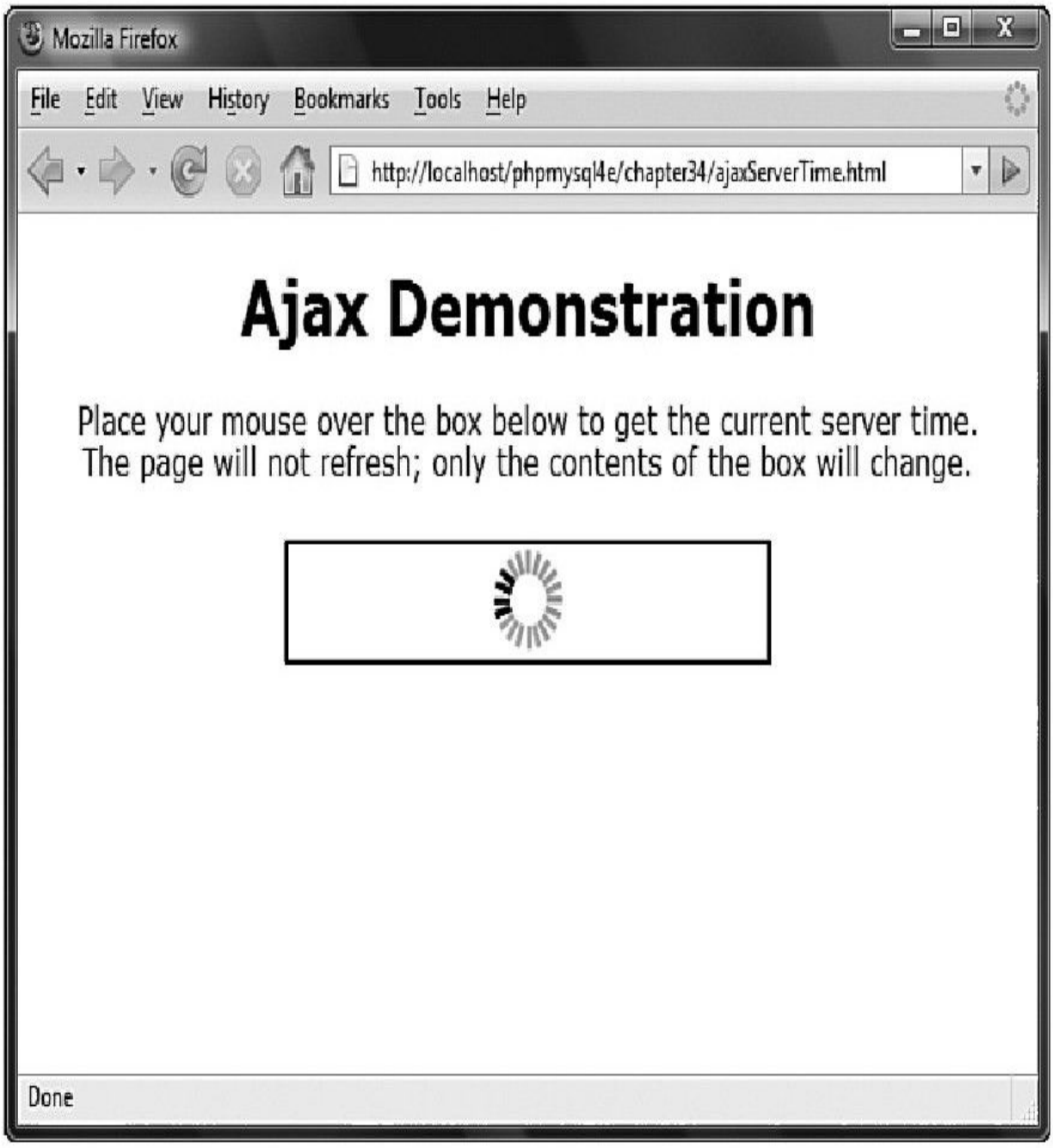


图　34-2　用户鼠标移过该区域将开始请求；图中所示图标表示对象正在载入



图　34-3　服务器端的结果显示在名为showtime的div元素中；再次将鼠标移过该区域将产生脚本的再次调用

34.3　在以前的项目添加Ajax元素

本书第五篇所介绍的任何项目都不是支持Ajax的。这些项目都包含了一系列的表单提交以及页面的重新载入。尽管页面包含动态元素，但是没有一个提供了Web 2.0应用带来的流畅的用户体验。

当然，在这些项目中添加Ajax元素可能会将重点从使用PHP和MySQL创建Web应用转移到Ajax。换句话说，必须先学会走，才能开始学跑。但是，既然已经知道如何跑（或者能蹦一点）可以开始考虑修改这些应用的元素来引入Ajax，或者开始考虑在未来将要创建的新应用引入Ajax元素。

Ajax开发人员的思考过程大概是这样的：什么是独有的用户操作，什么样的页面动作将调用这些操作？例如，是否期望用户通过点击一个按钮元素提交一个表单，然后再继续下一步操作，或者是否可以通过改变一个表单元素（文本输入框，单选框或复选框）的焦点来调用一个发送给服务器的异步请求？在确定了需要包括的动作类型后，可以编写这些调用PHP脚本的JavaScript函数，而PHP脚本将处理这些发送给服务器的请求或来自服务器的结果。

接下来的内容将介绍如何在本书前面章节创建的脚本中添加Ajax元素。

在PHPBookmark应用中添加Ajax元素

在第27章中，创建了一个名为PHPbookmark应用。该应用要求用户在保存书签并查看其他用户所保存的书签推荐之前进行用户注册和登录。

由于这个应用已经创建并且由一些联系紧密的PHP脚本和函数库组成，要做的第一步就是考虑如何在已有结构中添加额外的文件——是否某些文件是样式单、JavaScript函数或者处理服务器端动作的PHP脚本。回答是非常简单的：为样式单创建一个单独文件，为所有的JavaScript函数创建一个单独文件。然后，在第27章的已有PHP脚本添加一些代码来引入这两个外部文件，如果必要，也可以直接加入

JavaScript函数本身的调用。任何新创建的PHP脚本将区别于已有的应用文件，单独存在。

在确定如何管理新创建的文件后，可以开始考虑哪些用户函数需要支持Ajax。尽管该应用的用户注册和登录是支持Ajax的主要部分，但是这里，主要将围绕添加和编辑用户保存的书签部分转换成支持Ajax的部分进行介绍。

此外，还需要对已有的应用文件进行修改。将第27章的应用文件复制到一个新目录是必要的，这样可以方便本章的使用。本章所做的任何修改都将上传到这个目录，而不是PHPBookmark应用的工作目录中。

注意　如果要将应用的用户注册部分支持Ajax，可以通过也能够调用PHP脚本的JavaScript函数来验证用户的电子邮件以及用户名称是否已经存在于系统中。如果已经存在，该函数将显示一个错误信息，因此在这些错误被修改之前，不会继续注册的提交过程。

1.创建额外的文件

正如前面提到的，我们将在已有的应用结构中添加新的文件。尽管在接下来的内容会逐步将新加文件添加到应用中，但记住这一点也是有必要的。

假设至少有两个新文件：一个样式单和一个JavaScript函数库。现在先创建两个名为new_ss.css和new_ajax.js的新文件。由于新样式单还没有定义任何新样式，因此该文件为空。而new_ajax.js文件包含了本章前面所创建的getXMLHTTPRequest()函数，该函数可以在所有浏览器中创建一个XMLHTTPRequest对象实例。虽然还要在该文件添加更多的代码，但是现在就可以将该脚本上传到Web服务器。

接下来需要在PHPBookmark应用的某一个已有的显示函数中添加这两个文件的链接。这样就可以像JavaScript函数库中的函数一样，确保样式单中的样式定义是可以使用的。如果回忆第27章内容，控制head元素输出的函数名为do_html_header()，它位于output_fns.php文件中。该函数的新版本如程序清单34-4所示。

程序清单34-4　修改后的do_htmp_header()函数包含了指向新样式单和JavaScript函数库的链接

```
function do_html_header($title){
//print an HTML header
?>
<html>
<head>
<title><?php echo$title;?></title>
<style>
body{font-family:Arial,Helvetica,sans-serif;font-size:13px;}
li,td{font-family:Arial,Helvetica,sans-serif;font-size:13px;}
hr{color:#3333cc;}
a{color:#000000;}
</style>
<link rel="stylesheet"type="text/css"href="new_ss.css"/>
<script src="new_ajax.js"type="text/javascript"></script>
</head>
<body>
<img src="bookmark.gif"alt="PHPbookmark logo"border="0"
align="left"valign="bottom"height="55"width="57"/>
<h1>PHPbookmark</h1>
<hr/>
<?php
if($title){
do_html_heading($title);
}
}
```

在上载了新的样式单文件，JavaScript函数库，output_fns.php文件以及在PHPBookmark系统中打开任何页面，这些新文件都应该被正确引入，没有任何错误。接下来，将在这两个文件分别添加新的样式单和脚本，并创建一些Ajax功能。

2.以Ajax方式添加书签

目前，添加书签的操作出现在用户输入书签URL并点击表单提交按钮。点击表单提交按钮的动作将调用另一个PHP脚本，该脚本将添加书签，返回用户的书签列表，并且显示新添加的书签。换句话说，页面将被重新载入。而Ajax方式是给出一个添加书签的表单，通过后台调用一个JavaScript函数来调用PHP脚本，从而将该书签添加在数据库中并且将响应返回给用户，它不需要重新载入页面的表单提交按钮————所有这些操作都不需要离开已经被载入的页面。这个新函数将请求output_fns.php脚本中的display_add_bm_form()函数。

程序清单34-5所示的是修改后的函数。这里，删除了表单动作，添加了一个id为文本输入框的元素，以及修改了按钮元素的属性。此外，还添加了对getXMLHTTReuquest()函数的调用。

程序清单34-5　修改后的display_add_bm_form()函数

```
function display_add_bm_form(){
//display the form for people to enter a new bookmark in
?>
<script type="text/javascript">
var myReq=getXMLHTTPRequest();
</script>
<form>
<table width="250"cellpadding="2"cellspacing="0"bgcolor="#cccccc">
<tr><td>New BM:</td>
<td><input type="text"name="new_url"name="new_url"value="http://"
size="30"maxlength="255"/></td></tr>
<tr><td colspan="2"align="center">
<input type="button"value="Add bookmark"
onClick="javascript:addNewBookmark();"/></td></tr>
</table>
</form>
<?php
}
```

请仔细查看按钮元素，如下所示：

```
<input type="button"value="Add bookmark"
onClick="javascript:addNewBookmark();"/>
```

当该按钮被点击，onClick事件处理器将调用addNewBookmark()函数。该函数将向服务器发送一个请求，具体的，它将向尝试在数据库里插入一条记录的PHP脚本发送请求。该函数代码如程序清单34-6所示。

程序清单34-6　addNewBookmark()函数

```
function addNewBookmark(){
var url="add_bms.php";
var params="new_url="+encodeURI(document.getElementById('new_url').value);
myReq.open("POST",url,true);
myReq.setRequestHeader("Content-type","application/x-www-form-urlencoded");
myReq.setRequestHeader("Content-length",params.length);
myReq.setRequestHeader("Connection","close");
myReq.onreadystatechange=addBMResponse;
myReq.send(params);
}
```

以上函数类似于本章前面使用过的getServerTime()函数。其逻辑非常类似：创建变量，向PHP脚本发送数据以及调用函数处理服务器端的响应。

如下所示代码根据表单域的名称和用户输入值创建了一个名称/值对：

```
var params="new_url="+encodeURI(document.getElementById('new_url').value);
```

函数的最后一行代码将params参数值发送给后台的PHP脚本，如下所示：

```
myReq.send(params);
```

在发送之前，还将发送3个请求报头，这样服务器端就知道如何处理POST请求中的数据，如下所示：

```
myReq.setRequestHeader("Content-type","application/x-www-form-urlencoded");
myReq.setRequestHeader("Content-length",params.length);
myReq.setRequestHeader("Connection","close");
```

接下来就是创建JavaScript函数来处理服务器响应；这里定义为addBMResponse。

```
myReq.onreadystatechange=addBMResponse;
```

同样的，以上代码类似于本章前面介绍过的theHTTPResponse函数。程序清单34-7所示的是该函数代码。

程序清单34-7　addBMResponse()函数代码

```
function addBMResponse(){
if(myReq.readyState==4){
if(myReq.status==200){
result=myReq.responseText;
document.getElementById('displayresult').innerHTML=result;
}else{
alert('There was a problem with the request.');
}
}
}
```

该函数将首先检查对象的状态；如果请求已经完成，将检查来自服务器端的响应码是否是200（OK）。如果响应码不是200，函数将给出一个警告：“There was a problem with the request（该请求存在问题）”。任何其他的响应都将来自add_bms.php脚本的执行，并且将现在

id值为displayresult的div元素中。这里，displayresult定义在new_ss.css样式单中，如下所示（白色背景）：

```
#displayresult{
background:#fff;
}
```

在PHP函数display_add_bm_form()的form标记结束前，添加了如下所示的代码，该代码将显示用户将看到的服务器端结果：

```
<div id="displayresult"></div>
```

接下来，将对已有代码add_bms.php进行修改。

3.对已有代码进行修改

如果希望通过不修改add_bms.php脚本来添加书签，检查用户权限和添加书签的操作就足够了。然而，这样操作所返回的结果可能会不好理解，如图34-4所示。图中的标题和logo以及脚注链接都出现了两次，同时该应用的显示也出现一些问题。

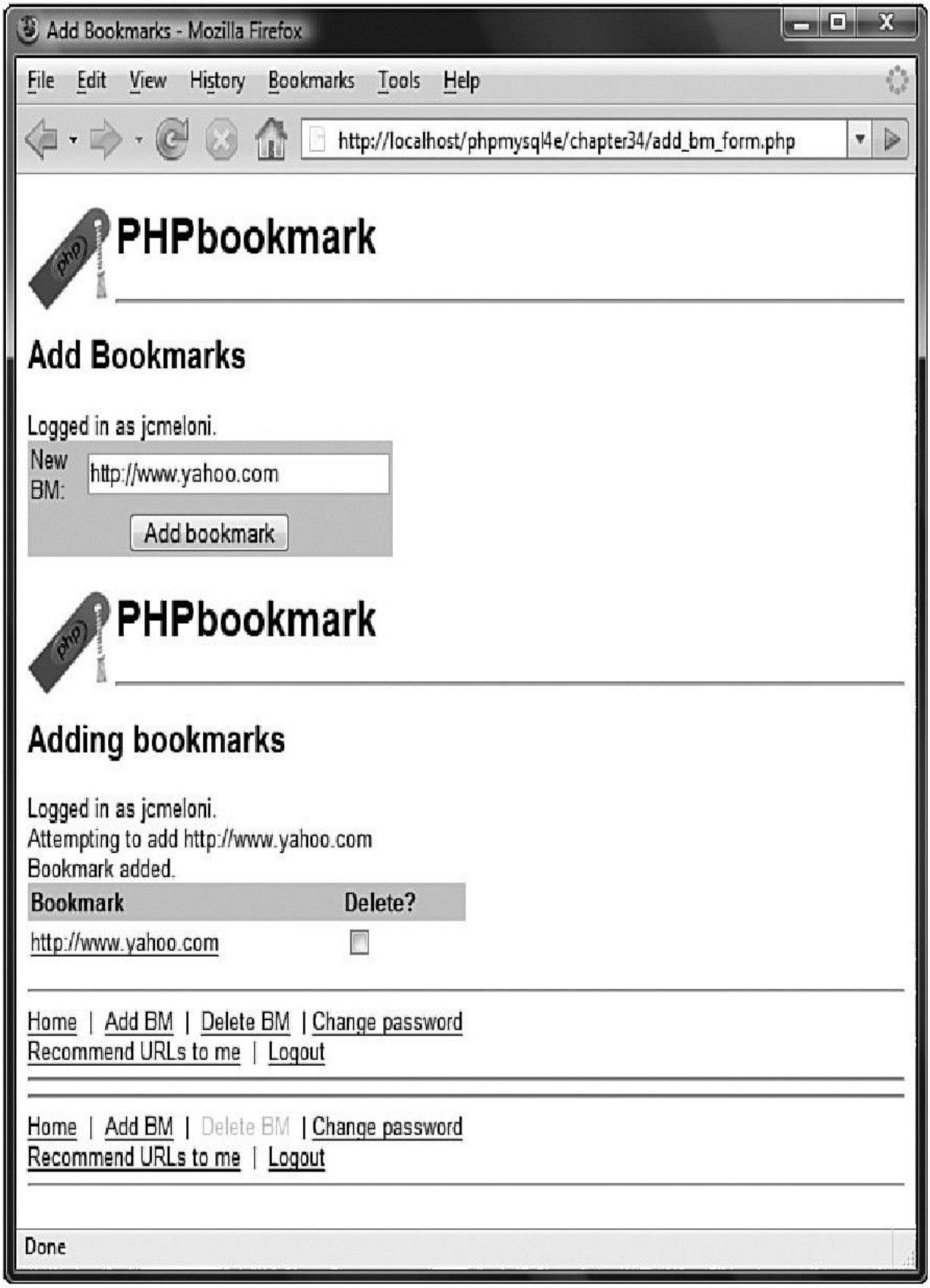

Add Bookmarks - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/phpmysql4e/chapter34/add_bm_form.php
PHPbookmark
Add Bookmarks
Logged in as jcmeloni.
New BM:
http://www.yahoo.com
Add bookmark
PHPbookmark
Adding bookmarks
Logged in as jcmeloni.
Attempting to add http://www.yahoo.com
Bookmark added.
Bookmark
Delete?
http://www.yahoo.com
Home | Add BM | Delete BM | Change password
Recommend URLs to me | Logout
Home | Add BM | Delete BM | Change password
Recommend URLs to me | Logout
Done

图　34-4　在修改add_bms.php脚本前添加书签

在PHPBookmark应用的非Ajax版本中，该表单只是一个页面，提交后的返回结果是另一个页面，而且所有页面元素都将实时的重新载入。然而，在支持Ajax的环境中，添加新书签，获得服务器结果以及是否继续添加新书签都不需要重新载入任何页面元素。这个新功能需要对add_bms.php代码进行一定的修改。最初的代码如程序清单34-8所示。

程序清单34-8　最初的add_bms.php代码

```
<?php
require_once('bookmark_fns.php');
session_start();
//create short variable name
$new_url=$_POST['new_url'];
do_html_header('Adding bookmarks');
try{
check_valid_user();
if(!filled_out($_POST)){
throw new Exception('Form not completely filled out.');
}
//check URL format
if(strstr($new_url,'http://')===false){
$new_url='http://'.$new_url;
}
//check URL is valid
if(!(@fopen($new_url,'r'))){
throw new Exception('Not a valid URL.');
}
//try to add bm
add_bm($new_url);
echo'Bookmark added.';
//get the bookmarks this user has saved
if($url_array=get_user_urls($_SESSION['valid_user'])){
display_user_urls($url_array);
```

```
}
}
catch(Exception$e){
echo$e->getMessage();
}
display_user_menu();
do_html_footer();
?>
```

该代码的第一行引入了所需的bookmark_fns.php文件。如果查看bookmark_fns.php文件代码，该文件还将调用其他文件：

```
<?php
//We can include this file in all our files
//this way,every file will contain all our functions and exceptions
require_once('data_valid_fns.php');
require_once('db_fns.php');
require_once('user_auth_fns.php');
require_once('output_fns.php');
require_once('url_fns.php');
?>
```

在支持Ajax的环境中，添加书签的操作可能不需要（也可能需要）这些所有引入文件，但就像该文件头的注释提到——每一个文件都包含了所有注释和异常。在这种情况下，由于应用架构由一系列的动态页面转移到支持Ajax的功能环境，在确认不需要某些代码功能之前，最好还是保留这些额外的元素。请保留add_bms.php的第一行代码。

第二行代码将开始或者继续一个用户会话，同样应该保留；即使是在支持Ajax的环境中，这个动作对安全的完整性是有意义的。同样的，第三行代码也可以保留。它给出了在请求发送中POST参数值，如下所示：

```
$new_url=$_POST['new_url'];
```

最后，可以删除如下所示的代码行：

```
do_html_header('Adding bookmarks');
```

由于已经正在浏览包含HTML报头信息的页面（add_bm_form.php），就不需要重复载入该页面了——不需要转移到其他页面。这种重复将产生两套页面标题和Logo图像，正如图34-4所示。类似的原因，也可以删除add_bm_form.php最后两行的代码，如下所示：

```
display_user_menu();
do_html_footer();
```

如果删除这些元素，上载文件到服务器以及添加其他书签，其结果都会是期望的，当然还是需要修改一些脚本。图34-5所示的就是对脚本修改后的应用运行的结果。

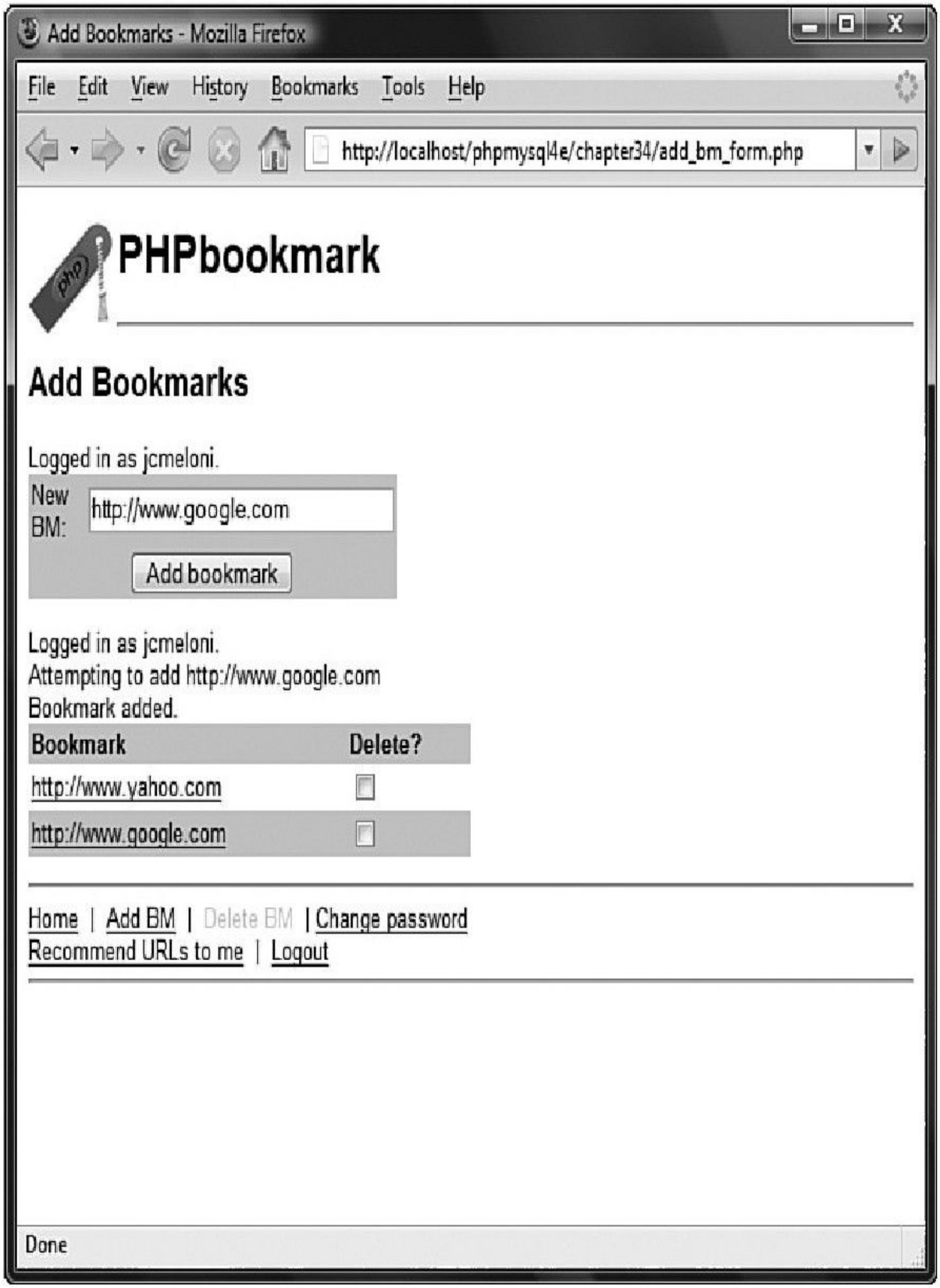

Add Bookmarks - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/phpmysql4e/chapter34/add_bm_form.php
PHPbookmark
Add Bookmarks
Logged in as jcmeloni.
New BM:
http://www.google.com
Add bookmark
Logged in as jcmeloni.
Attempting to add http://www.google.com
Bookmark added.
Bookmark
Delete?
http://www.yahoo.com
http://www.google.com
Home | Add BM | Delete BM | Change password
Recommend URLs to me | Logout
Done

图　34-5　修改add_bms.php脚本后添加书签的运行结果

在结果页面上，用户登录状态信息仍然是重复的，但这个问题已经没有以前明显了。接下来要删除这些重复的信息并且修改一些与异常处理相关的功能，这些功能在支持Ajax的环境中才有真正的意义。

要删除用户登录的重复信息，可以删除add_bms.php脚本中的如下代码行：

```
check_valid_user();
```

有效用户的检查已经在add_bms_form.php页面载入时执行；当进入这个已经调用了Ajax函数的页面时，就不用再担心用户是否有效。

接下来删除try语句块和异常处理。删除它们的原因是希望脚本结束运行时能够将已经保存的URL列表显示给用户。这就意味着某些修改和调整可能会引入一些新的错误信息文本。程序清单34-9所示的是修改后的add_bms.php代码。

程序清单34-9　修改后的add_bms.php代码

```
<?php
require_once('bookmark_fns.php');
session_start();
//create short variable name
$new_url=$_POST['new_url'];
//check that form has been completed
if(!filled_out($_POST)){
//has not
echo"<p class=\"warn\">Form not completely filled out.</p>";
}else{
//has;check and fix URL format if necessary
if(strstr($new_url,'http://')===false){
$new_url='http://'.$new_url;
}
```

```
//continue on to check URL is valid
if(!(@fopen($new_url,'r'))){
echo"<p class=\"warn\">Not a valid URL.</p>";
}else{
//it is valid,so continue to add it
add_bm($new_url);
echo"<p>Bookmark added.</p>";
}
}
//regardless of the status of the current request
//get the bookmarks this user has already saved
if($url_array=get_user_urls($_SESSION['valid_user'])){
display_user_urls($url_array);
}
?>
```

以上代码仍然延续了可能事件的逻辑顺序，但是能够显示适当的错误信息，不会显示任何重复的页面元素。

以上代码首先将检查表单本身是否已填写。如果没有完全填写，将在书签添加表单和用户已有书签列表之间显示错误信息。如图34-6所示的就是这个响应。

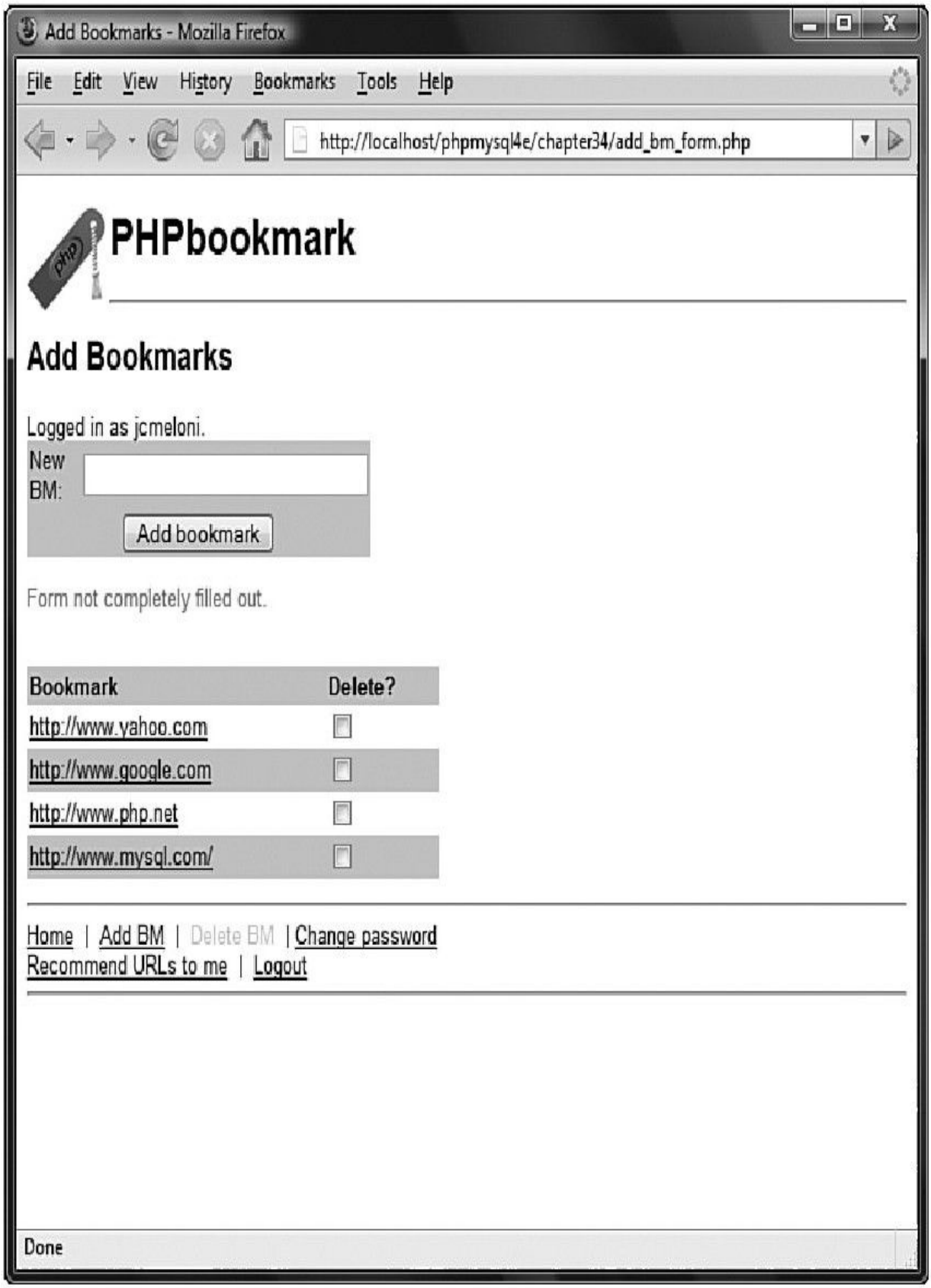

Add Bookmarks - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/phpmysql4e/chapter34/add_bm_form.php
PHPbookmark
Add Bookmarks
Logged in as jcmeloni.
New BM:
Add bookmark
Form not completely filled out.
Bookmark
Delete?
http://www.yahoo.com
http://www.google.com
http://www.php.net
http://www.mysql.com/
Home | Add BM | Delete BM | Change password
Recommend URLs to me | Logout
Done

图　34-6　尝试添加一个空值

其次，以上代码将检查URL格式是否正确；如果不正确，该URL字符串将被转换成正确的URL格式并继续执行。接下来，将打开一个Socket测试该URL是否有效。如果测试失败，将在书签添加表单和用户已有书签列表之间显示错误信息。然而，如果URL是有效的，它将被添加到用户已有书签列表。如图34-7所示的是添加一个无效URL时，应用给出的响应信息。

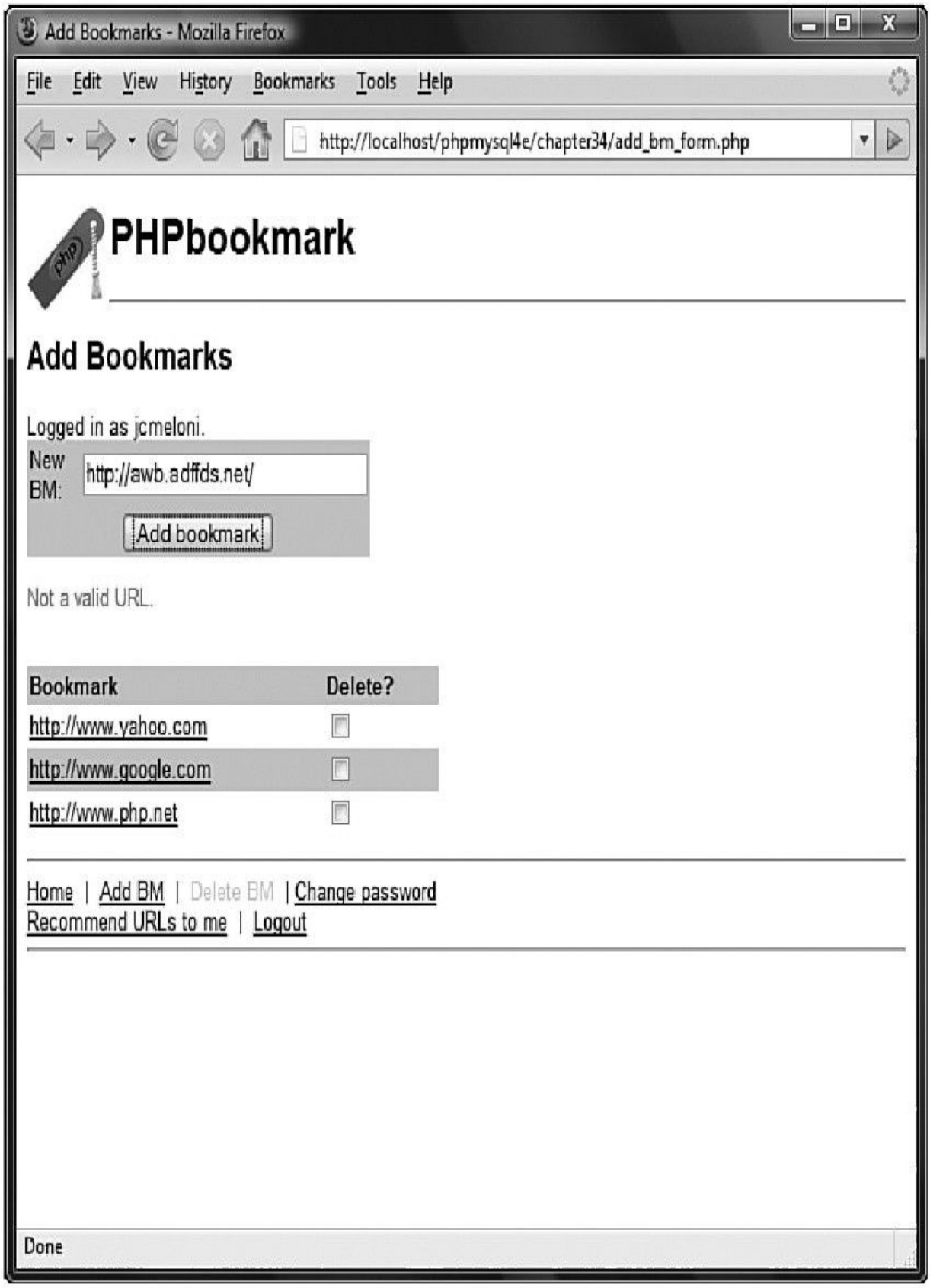

Add Bookmarks - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/phpmysql4e/chapter34/add_bm_form.php
PHPbookmark
Add Bookmarks
Logged in as jcmeloni.
New BM:
http://awb.adffds.net/
Add bookmark
Not a valid URL.
Bookmark
Delete?
http://www.yahoo.com
http://www.google.com
http://www.php.net
Home | Add BM | Delete BM | Change password
Recommend URLs to me | Logout
Done

图　34-7　尝试添加一个无效URL

最后，无论添加URL至用户已有书签列表是否存在错误，都将显示用户的已有书签列表。如图34-8所示的是该结果。

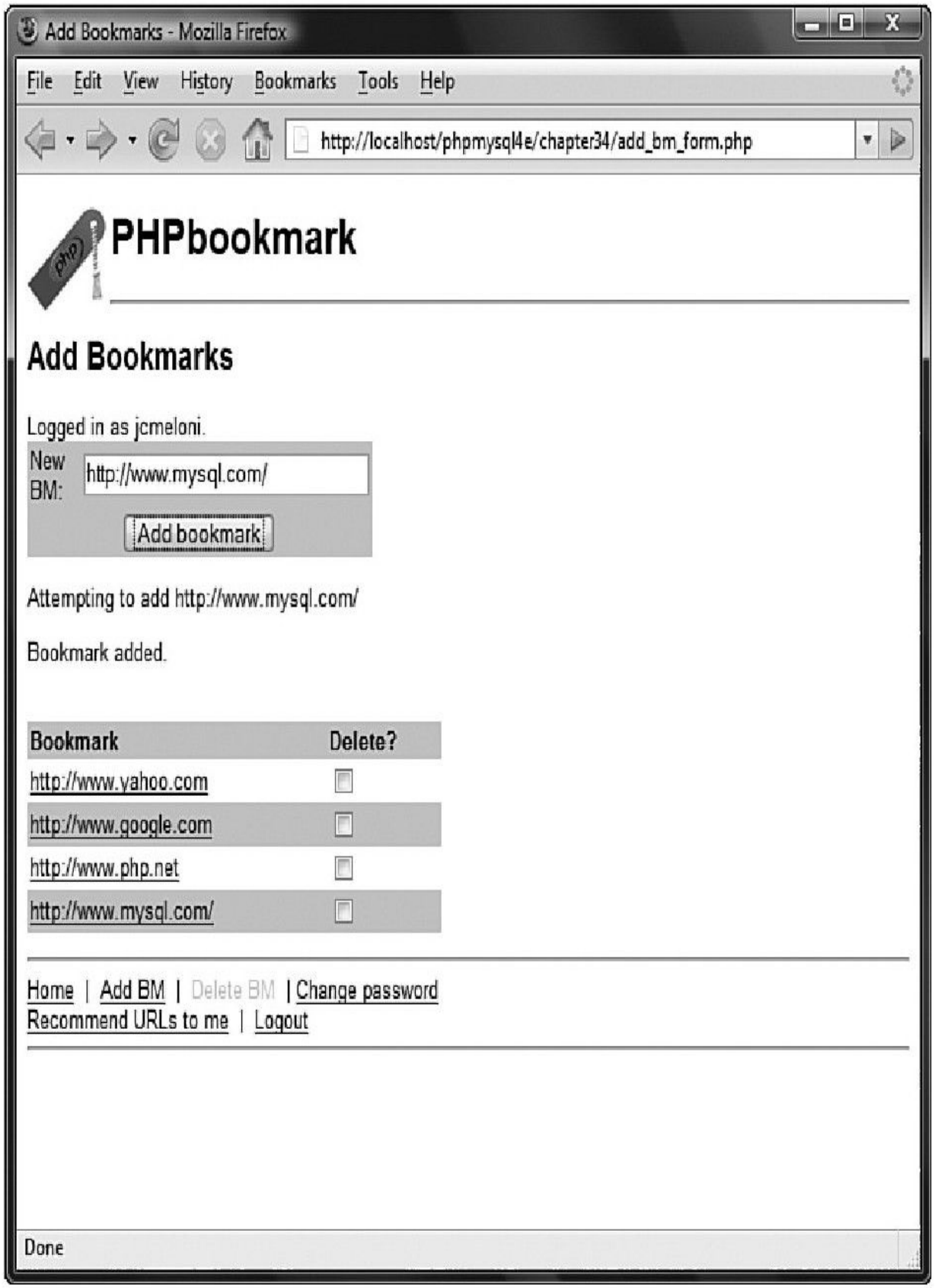

Add Bookmarks - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/phpmysql4e/chapter34/add_bm_form.php
PHPbookmark
Add Bookmarks
Logged in as jcmeloni.
New BM:
http://www.mysql.com/
Add bookmark
Attempting to add http://www.mysql.com/
Bookmark added.
Bookmark
Delete?
http://www.yahoo.com
http://www.google.com
http://www.php.net
http://www.mysql.com/
Home | Add BM | Delete BM | Change password
Recommend URLs to me | Logout
Done

图　34-8　成功——添加了一个有效的URL

尽管添加书签这个核心功能已经成功地支持Ajax，还需要对一些元素进行修改。例如，在新系统中，ulr_fns.php脚本的add_bm()函数包含的一些异常处理可能会导致错误信息。程序清单34-10所示的是已有的add_bm()函数。

程序清单34-10　ulr_fns.php脚本中已有的add_bm()函数代码

```
function add_bm($new_url){
//Add new bookmark to the database
echo"Attempting to add".htmlspecialchars($new_url)."<br/>";
$valid_user=$_SESSION['valid_user'];
$conn=db_connect();
//check not a repeat bookmark
$result=$conn->query("select*from bookmark
where username='$valid_user'
and bm_URL='".$new_url."'");
if($result&&($result->num_rows>0)){
throw new Exception('Bookmark already exists.');
}
//insert the new bookmark
if(!$conn->query("insert into bookmark values
('".$valid_user."','".$new_url."')")){
throw new Exception('Bookmark could not be inserted.');
}
return true;
}
```

在这里，我们希望能够将异常转换成错误信息，并且继续代码的执行（显示）。这可以通过修改两个if语句块实现：

```
if($result&&($result->num_rows>0)){
echo"<p class=\"warn\">Bookmark already exists.</p>";
}else{
```

```
//attempt to add
if(!$conn->query("insert into bookmark values
('".$valid_user."','".$new_url."')")){
echo"<p class=\"warn\">Bookmark could not be inserted.</p>";
}else{
echo"<p>Bookmark added.</p>";
}
}
```

以上代码仍然延续了可能事件的逻辑顺序，但是能够显示适当的错误信息。在检查要添加的URL是否已经存在于用户已有书签列表后，或者在书签添加表单和用户已有书签列表之间给出一个错误信息，或者脚本尝试添加这个URL。

如果不能添加这个书签，在书签添加表单和用户已有书签列表之间还将显示一个错误信息。但是，如果成功添加该URL后，将显示“Bookmark Added（已添加书签）”信息。从add_bm.php脚本删除的echo语句被添加到add_bm()函数中，否则用户还将看到一个错误信息，例如，在“Bookmark could not be inserted（不能插入书签）”消息之后还将显示“Bookmark Added（已添加书签）”的成功消息，而事实的输出确实是不正确的。

如图34-9所示的就是这些修改的结果。

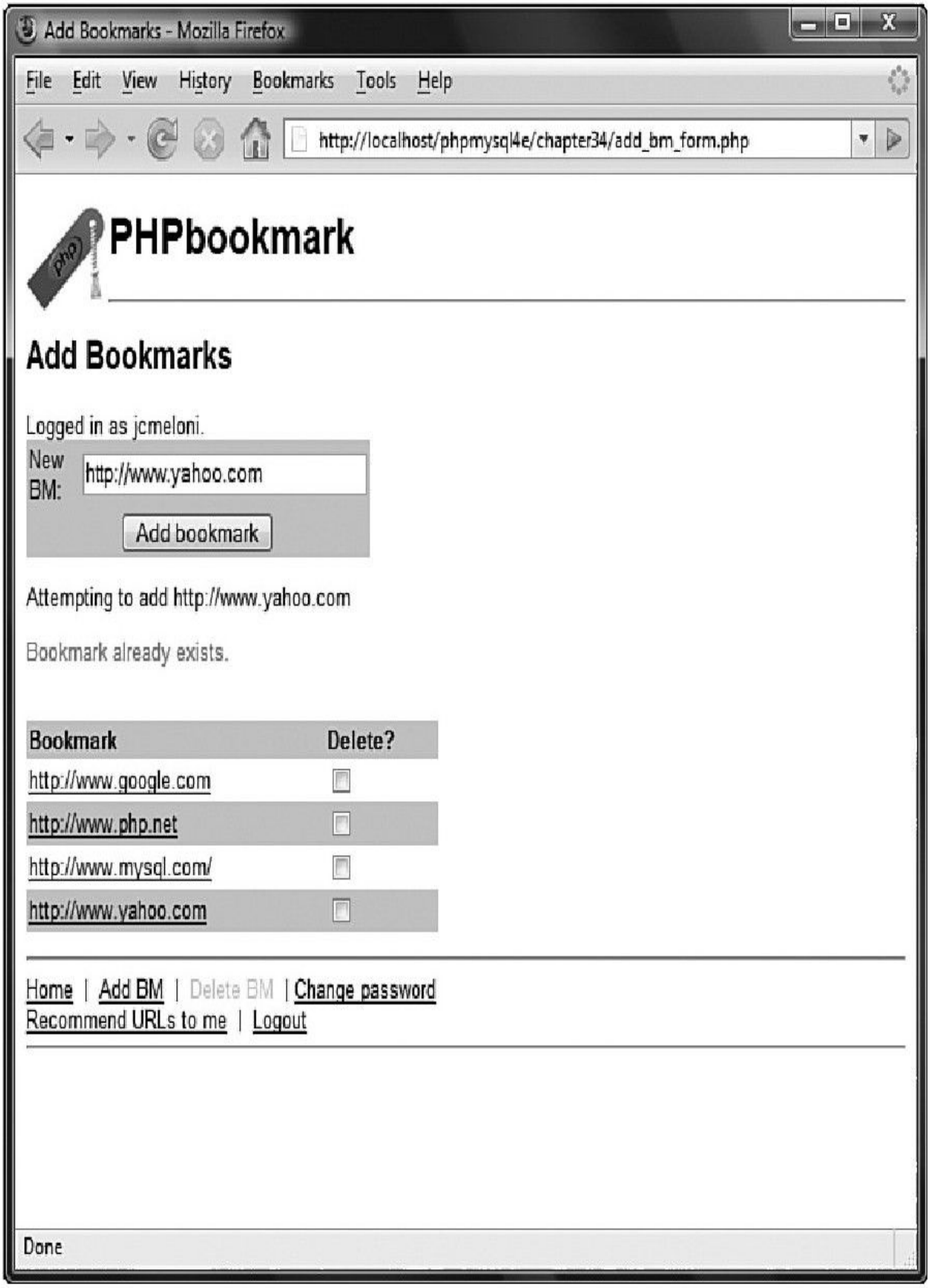

Add Bookmarks - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/phpmysql4e/chapter34/add_bm_form.php
PHPbookmark
Add Bookmarks
Logged in as jcmeloni.
New BM:
http://www.yahoo.com
Add bookmark
Attempting to add http://www.yahoo.com
Bookmark already exists.
Bookmark
Delete?
http://www.google.com
http://www.php.net
http://www.mysql.com/
http://www.yahoo.com
Home | Add BM | Delete BM | Change password
Recommend URLs to me | Logout
Done

图　34-9　尝试添加一个已存在的书签

4.对PHPBookmark应用的额外修改

将书签添加功能修改为支持Ajax的用户界面只是要对这个应用进行的许多修改之一。接下来的选择可能就是书签删除功能。书签删除的操作如下所示：

■从页面脚注删除“Delete BM（删除书签）”链接。

■当用户选中某个书签左边的“Delete?（是否删除）”复选框时，调用一个JavaScript函数。

■修改delete_bm.php脚本，这样新的JavaScript函数就可以调用这个脚本完成删除操作并向用户返回信息。

■对功能进行必要的修改，确保操作能够合理完成，并且在新的用户界面显示正确的信息。

有了修改的结构，就可以根据本章所介绍的内容实现这些修改。接下来的内容将提供一些资源链接，这些链接包含了关于创建支持Ajax站点的更多更详细信息。

请记住，Ajax是一系列技术的组合，用于创建更流畅的用户体验；在了解了Ajax这些组成部分及其功能后，这也让重新思考一个应用的架构成为必要。

34.4 进一步学习

本章关于创建支持Ajax的应用所涉及的内容只是非常肤浅的。有一本新书《Sams Teach Yourself Ajax,JavaScript and PHP All-in-One》详尽的讨论了本章所涉及的所有内容，它可以作为学习了本章内容后进一步提高的工具。当然，也有许多网站专门或者部分地介绍构建Ajax应用的技术，以及一些可以加速开发进程的第三方代码库，这样就能避免重复工作。

34.4.1 进一步了解文档对象模型（DOM）

尽管本书介绍了使用PHP进行服务器端编程以及使用MySQL作为支撑动态应用的关系数据库，它并没有涵盖任何关于客户端的内容，例如，XHTML、CSS、JavaScript以及文档对象模型（Document Object Modal，DOM）。如果还不熟悉DOM，可能需要丰富这方面的内容，因为要创建一个功能全面的Ajax应用，这方面的内容是必不可少的。

许多（如果不是所有）Ajax应用都会使用JavaScript来操作DOM。无论是使用显示元素，浏览器历史记录或者窗口位置，透彻地理解DOM对象及其属性对于创建一个流畅的用户界面来说是至关重要的，这也是Ajax应用的目标。

如下站点包含了大量学习DOM的信息：

■ W3C关于文档对象模型的技术报告：http://www.w3.org/DOM/DOMTR

■DOM脚本编程组织的主页：http://domscripting.webstandards.org/

■ Mozilla项目关于DOM的开发人员文档：http://developer.mozilla.org/en/docs/DOM（以及关于JavaScript文档的链接：http://developer.mozilla.org/en/docs/JavaScript）

34.4.2 Ajax应用可用的JavaScript函数库

Ajax应用大概出现在2005年，是Jesse James Garret在他的文章提出了这个名词，因为在向其客户描述这个技术时，“他需要一个更短的名词来表述‘异步的JavaScript+CSS+DOM+XMLHtttpRequest’”。从那以后，有很多第三方开发人员提供了许多可供普通开发人员在创建Ajax应用时使用的JavaScript函数库。

注意　请参阅Garret的文章“Ajax:A New Approach to Web Applications（Ajax：Web应用的新方法）”。

尽管花些时间浏览任何Ajax开发站点可以很快找到更多信息，这里还是给出一些流行的函数库。选择其中一个都将减少开发时间，正如前面提到的，不需要开发人员进行一些重复工作。

■Prototype JavaScript框架可以简化DOM对象的操作。在创建复杂的Ajax应用时，可以使用XMLHTTPRequest对象。更多信息，请参阅：http://www.prototypejs.org/。

■Dojo是一个开放源代码的工具包，它包含了许多基本的JavaScript函数，一个创建框架以及有效的打包并向终端用户分发代码的机制。更多信息，请参阅：http://dojotoolkit.org/。

■MochiKit是一个轻量级的代码库，它包含了处理DOM以及格式化客户端输出的函数。MochiKit的口号有点“不友好”但是很真实：“MochiKit使得JavaScript不那么糟糕”。MochiKit提供的函数和解决方案，可供开发人员使用的文档以及使用MochiKit创建的示例项目都值得一看。更多信息，请参阅：http://mochikit.com/。

34.4.3 Ajax开发人员网站

最后，学习Ajax开发的最佳途径就是实践。找到一些代码，确定如何集成到已有应用，以及向已经使用该技术的开发人员学习。如下所示的网站将有助于更好的开始Ajax应用的开发：

■Ajaxian是一个开发人员门户网站，它为新的或者有经验的开发人员提供了新闻、文章、教程以及示例代码。更多信息，请参阅：http://ajaxian.com/。

■Ajax Matters包含一些深度介绍Ajax开发的文章。更多信息，请参阅：http://www.ajaxmatters.com/。

■Ajax Lines是另一个开发人员门户网站，它提供了关于Ajax所有内容的新闻和文章。更多信息，请参阅：http://www.ajaxlines.com/。

附录

附录A　安装PHP及MySQL

Apache、PHP和MySQL都可以用于多种操作系统和Web服务器的组合。在本附录中，我们将介绍如何在各种服务器平台上安装Apache、PHP和MySQL。我们所介绍的内容包括UNIX和Windows Vista的大多数共有选项。

在本附录中，我们将介绍以下主要内容：

■将PHP作为一个CGI解释器或一个模块来运行

■在UNIX下安装Apache、SSL、PHP和MySQL

■在Windows下安装Apache、PHP和MySQL

■使用phpinfo()函数测试它是否正常工作

■安装PEAR

■考虑其他配置

提示　本附录没有给出如何在Microsoft Internet Information Server或其他Web服务器中添加PHP模块的介绍。我们建议尽可能使用Apache Web服务器。关于在Microsoft IIS和个人Web服务器（PWS）上安装PHP，请参阅PHP手册（http://www.php.net/manual/en/install.windows.iis.php）。

本附录的目的是提供Web服务器的安装指导，同时可以使Web服务器成为多个网站的宿主服务器。某些站点，就像在示例中涉及的，要求使用Secure Sockets Layer（SSL，加密套接字协议）来实现电子商务解决方案。而且，大多数站点的驱动方法都是通过使用脚本程序连接到数据库（DB）服务器，从而读取并处理数据。

许多PHP用户根本不需要在一台机器上安装PHP，这就是为什么我们将这部分的内容作为附录而不是第1章来介绍。通过使用一个快速的Internet连接来访问一个可靠并且已经安装了PHP的服务器的最简单方法就是在成百上千个主机服务或遍布全球的主机服务零售商那里注册一个账户。

根据安装PHP的用途不同，我们可能会做出不同的决定。如果拥有一台能够长期连接到互联网的机器，并且将该机器作为一台实时服务器，那其性能就非常重要。如果只是构建一个开发服务器并且只是编写和测试代码，拥有与实时服务器类似的配置，同样也是需要考虑的重要因素。如果希望在同一台机器上运行ASP和PHP，可能需要不同的配置。

提示　PHP解释器可以作为一个模块或一个单独的CGI二进制代码来运行。通常情况下，使用模块版本是出于性能原因考虑的。但是，使用CGI版本可能出于多种原因：有时是在无法使用模块版本的服务器中，或者由于它允许Apache用户在不同的用户ID下运行支持PHP的不同页面，我们可能也会选择使用CGI版本。

在本附录中，我们主要介绍以模块方式运行PHP。

A.1　在UNIX环境下安装Apache、PHP和MySQL

根据我们的需求以及对UNIX系统的熟悉程度的不同，可能会选择一个二进制代码安装或直接通过编译源代码进行安装。这两种方法各有利弊。

对于一个UNIX熟练人员来说，安装二进制代码会非常快，而对初学者来说，花费的时间可能要相对长一点，但是这都将导致系统可能存在多个版本，并且其中一个版本是按照其他人所选的选项进行配置的。

源代码安装需要花费几个小时的时间来下载代码、安装和配置。对于最初的几次，我们不得不这样做，但是，这样的安装可以提供完全的控制。例如，我们可以选择安装的内容，使用的版本以及设置的配置指令。

A.1.1　二进制代码安装

大多数Linux操作系统都会包括一个预先配置好的Apache Web服务器，同时也内置有PHP。而到底这些操作系统都提供了哪些内容是由所选择的操作系统和版本确定的。

二进制代码安装的一个缺点是无法获得最新版本的程序。根据最近几次bug修复版本的重要性，使用一个早期版本可能不会有太大的问题。最大的问题是我们无法选择将哪些选项编译到程序中。

最灵活和最可靠的方法是对所需的程序进行完全的源代码重新编译。与安装RPM相比，这样做所花费的时间可能要长一些，因此可以选择使用RPM或者其他可供使用的RPM包。即使在正式站点无法获得带有所需配置的二进制文件，也可以通过一些搜索引擎找到一些非正式的版本。

A.1.2　源代码安装

现在，让我们在UNIX环境下安装Apache、PHP和MySQL。首先，必须确定到底需要安装哪一个模块。由于本书中介绍的一些例子在Web事务中都使用了安全服务器，因此我们将安装一个支持SSL（加密套接字）的服务器。

作为本书的用途，我们的PHP配置将或多或少地使用默认设置，但是也还是会包括启用PHP中gd2库的设置。

gd2只是PHP众多可供使用的库当中的一个。我们将包括它的安装步骤，这样可以知道在PHP中需要哪些额外的库来支持这些特性。大多数UNIX程序的编译都会使用相似的步骤。

通常，在安装了新库后，必须重新编译PHP，因此，如果事先知道所需要的内容，可以在机器上安装所有必需的，然后再开始编译PHP模块。

在这里，我们的安装将在SuSE Linux服务器上进行，但是这些安装操作是非常规范的，适用于其他UNIX服务器。

首先，我们必须搜集安装所需的文件。我们需要如下所示的程序：

■Apache（http://httpd.apache.org/）：Web服务器。

■OpenSSL（http://www.openssl.org/）：实现了加密套接字的开放源代码工具包。

■Mod_SSL（http://www.modssl.org/）：提供了OpenSSL的Apache模块接口。

■MySQL（http://www.mysql.com/）：关系型数据库。

■PHP（http://www.php.net/）：服务器端的脚本语言。

■ftp://ftp.uu.net/graphics/jpeg/：PDFlib和gd所需的JPEG库。

■http://www.libpng.org/pub/png/libpng.html：gb所需的PNG库。

■http://www.zlib.net/：PNG库所需的zlib库。

■http://www.libtiff.org/：PDFlib所需的TIFF库。

■ftp://ftp.cac.washington.edu/imap/：IMAP C客户端，用于IMAP。

如果希望使用mail()函数，必须安装一个MTA（邮件传输代理）。在这里，我们不介绍它。

我们假设已经拥有这台服务器的root用户访问权限，而且已经在系统中安装了如下所示的工具：

■gzip或gunzip

■gcc和GNU make文件

当准备开始安装过程时，应该首先将所有tar文件下载到一个临时目录。请确认将它们保存在一个有足够空间的位置。在我们的例子中，使用了/usr/src作为临时目录。应该以root用户下载它们，这样可以避免文件权限问题。

1.安装MySQL

在本节中，我们将介绍如何完成MySQL的二进制代码安装。这个安装过程把不同的文件自动放置在不同的目录中。我们为其他部分所选择的目录包括：

■/usr/local/apache

■/usr/local/ssl

在安装这些软件前，通过修改路径前缀，可以将这些应用程序安装在不同的目录下。

下面，就让我们开始吧。使用su命令，成为root用户。

```
$su root
```

输入root用户的密码。进入到保存源文件的目录，例如：

```
#cd/usr/src
```

目前，MySQL建议人们下载MySQL的二进制代码，而不是通过编译它来进行安装。具体使用哪个版本将由所期望的功能确定。虽然MySQL的预发行版本通常也是非常稳定的，但是建议如果选择在产品站点上不使用它们。如果正在了解或调试机器，可以选择使用这些版本之一。

我们必须下载如下所示的RPM包：

```
MySQL-server-VERSION.i386.rpm
MySQL-Max-VERSION.i386.rpm
MySQL-client-VERSION.i386.rpm
```

（VERSION是版本号的占位符。对于所选择的版本，请确认选择了匹配的RPM包）。如果想在同一台机器上运行MySQL客户端和服务器，并且将MySQL编译成能够支持其他程序，例如PHP，需要以上所有RPM包。

使用如下命令安装以上所有RPM包：

```
rpm-i MySQL-server-VERSION.i386.rpm
rpm-i MySQL-Max-VERSION.i386.rpm
rpm-I MySQL-client-VERSION.i386.rpm
```

现在，MySQL服务器应该已经安装并运行起来。

接下来，可以为root用户设置密码。请确认，使用所选择的新密码替换了如下所示命令行中的"new-password"；否则，"new-

password"将成为root用户密码：

```
mysqladmin-u root password'new-password'
```

安装MySQL时，它将自动创建两个数据库。一个是mysql表，该表控制了服务器投入使用后的用户、主机和DB权限。另一个为测试数据库。可以通过如下所示的命令检查数据库：

```
#mysql-u root-p
Enter password:
mysql>show databases;
+--------------------+
|Database|
+--------------------+
|mysql|
|test|
+--------------------+
2 rows in set(0.00 sec)
```

输入quit或\q来退出MySQL客户端。

MySQL的默认配置允许任何用户在不提供用户名称或密码的前提下访问系统。显然，这并不是我们所希望的。

最后，完成MySQL配置所需的操作是删除匿名用户。打开一个命令提示符，输入如下所示的命令就可以完成该操作：

```
#mysql-u root-p
mysql>use mysql
mysql>delete from user where User='';
mysql>quit
```

还需要执行如下命令：

```
mysqladmin-u root-p reload
```

该命令可以使这些修改生效。

还应该启用MySQL服务器的二进制登录，因为在计划使用复制功能时将需要该特性。要启用二进制登录，首先需要停止服务器的运行：

```
mysqladmin-u root-p shutdown
```

创建一个/etc/my.cnf文件并且使其作为MySQL的选项文件。这时候，只需要一个选项，但是在这里可以设置几个选项。请查阅MySQL手册，了解完整的选项列表。

打开该文件并输入：

```
[mysqld]
log-bin
```

保存该文件并退出。运行mysqld_safe命令重新启动该服务器。

2.安装PHP

仍然需要以root身份进行安装，如果不是，请重新进入root身份。

在安装PHP之前，必须对Apache进行预配置，这样它就可以知道所有软件包都在哪里（当介绍如何设置Apache服务器时，我们还要详细介绍这一点）。回到保存了源代码的目录：

```
#cd/usr/src
#gunzip-c httpd-2.2.9.tar.gz|tar xvf-
#cd httpd-2.2.9
#./configure--prefix=/usr/local/apache2
```

好了，现在可以开始设置PHP了。对源文件进行解压缩操作，进入其目录：

```
#cd/usr/src
#gunzip-c php-5.2.6.tar.gz|tar xvf-
#cd php-5.2.6
```

这里需要再次提到的是，PHP的configure命令具有许多选项。使用./configure--help|less可以确定所希望添加的内容。在这个例子中，我们希望添加对MySQL、Apache、PDFLib和gd的支持。

请注意，如下所示的内容只是一个命令。我们可以将所有选项放在一行，或者就像我们现在这样，使用延续字符，反斜杠（\）允许我们在多行中输入一条命令，这样可以提高命令的可读性。

```
#./configure--prefix=/your/path/to/php
--with-mysqli=/your/path/to/mysql_config\
--with-apxs2=/usr/local/apache2/bin/apxs\
--with-jpeg-dir=/path/to/jpeglib\
--with-tiff-dir=/path/to.tiffdir\
--with-zlib-dir=/path/to/zlib\
--with-imap=/path/to/imapcclient\
--with-gd
```

接下来，执行make命令并安装二进制代码：

```
#make
#make install
```

将一个INI文件复制到lib目录：

```
#cp php.ini-dist/usr/local/lib/php.ini
```

或者

```
#cp php.ini-recommended/usr/local/lib/php.ini
```

在以上命令中，两个版本的php.ini文件具有不同的选项集。第一个 php.ini-dist 是为开发用户的机器而准备的。例如，它将display_errors，选项设置为On。这个设置将使得开发过程变得更加简单，但是对于一台产品机器，这个设置却是不合适的。当我们推荐使用本书中给出的默认设置时，使用了这个版本的php.ini文件。第二个版本，php.ini-recommended是适用于产品机器的。

可以编辑这个php.ini文件来设置PHP的选项。可以选择任何想要的选项，但是除了一部分以外，其他的对我们来说都是没有意义的。如果希望通过脚本发送电子邮件，需要设置sendmail_path选项。

现在，可以设置OpenSSL。在创建临时证书和CSR文件时，我们将使用这个库。--prefix选项表示主要的安装目录。

```
#gunzip-c openssl-0.9.8h.tar.gz|tar xvf-
#cd openssl-0.9.8h
#./config--prefix=/usr/local/ssl
```

使用如下命令执行make，测试并安装OpenSSL：

```
#make
#make test
#make install
```

接下来，我们需要为其编译而对Apache进行配置。配置选项"——enable-so"将启用动态共享对象（DSO），而配置选项"——enable-ssl"将启用mod_ssl模块。这里强烈建议ISP或安装维护人员使用DSO最大化软件的灵活性。请注意，Apache并不是在所有平台都支持DSO。

```
#cd../httpd-2.2.9
#SSL_BASE=../openssl-0.9.8h\
./configure\
--prefix=/usr/local/apache2\
--enable-so
--enable-ssl
```

最后，你可以调用make文件生成Apache二进制及其证书并安装它们：

```
#make
```

如果顺利完成了以上所有操作，将看到类似于如下消息：

```
+-----------------------------------------------------------------------+
|Before you install the package you now should prepare the SSL|
|certificate system by running the'make certificate'command.|
|For different situations the following variants are provided:|
||
|%make certificate TYPE=dummy(dummy self-signed Snake Oil cert)|
|%make certificate TYPE=test(test cert signed by Snake Oil CA)|
|%make certificate TYPE=custom(custom cert signed by own CA)|
|%make certificate TYPE=existing(existing cert)|
|CRT=/path/to/your.crt[KEY=/path/to/your.key]|
||
|Use TYPE=dummy when you're a vendor package maintainer,|
|the TYPE=test when you're an admin but want to do tests only,|
|the TYPE=custom when you're an admin willing to run a real server|
|and TYPE=existing when you're an admin who upgrades a server.|
|(The default is TYPE=test)|
||
|Additionally add ALGO=RSA(default)or ALGO=DSA to select|
|the signature algorithm used for the generated certificate.|
||
```

```
|Use'make certificate VIEW=1'to display the generated data.|
||
|Thanks for using Apache&mod_ssl.Ralf S.Engelschall|
|rse@engelschall.com|
|www.engelschall.com|
+-----------------------------------------------------------------+
```

现在，可以创建一个自定义的证书。这个选项将提示给出证书位置、公司名称和其他信息。对于联系信息，可以使用真实数据。对于操作过程中的其他问题，可以使用默认的回答。

```
#make certificate TYPE=custom
```

现在，可以安装Apache：

```
#make install
```

如果所有操作都正确的话，将看到类似于如下消息：

```
+-------------------------------------------------------+
|You now have successfully built and installed the|
|Apache 2.2 HTTP server.To verify that Apache actually|
|works correctly you now should first check the|
|(initially created or preserved)configuration files|
||
|/usr/local/apache2/conf/httpd.conf
||
|and then you should be able to immediately fire up|
|Apache the first time by running:|
||
|/usr/local/apache2/bin/apachectl start
||
||
```

```
|Thanks for using Apache.The Apache Group|
|http://www.apache.org/|
+-----------------------------------------------------+
```

现在，可以查看Apache和PHP是否已经工作。但是，我们必须编辑httpd.conf文件，将PHP类型添加到配置文件中。

A.1.3 httpd.conf文件：摘录

打开httpd.conf文件，添加或者取消如下语句行的注释。如果是按照前面所介绍的步骤执行的，httpd.conf文件将被保存在/usr/local/apache2/conf目录中。在该文件中，适用于PHP的addtype选项已经被注释了，必须取消这一行的注释，如下所示：

```
AddType application/x-httpd-php.php
AddType application/x-httpd-php-source.phps
```

现在，我们可以启动Apache服务器看它是否工作正常。首先，我们以不支持SSL的方式启动服务器看它是否能启动。我们将检查PHP支持，然后停止服务器，并再次以启用SSL支持的方式启动它看是否一切工作正常。

使用配置测试configtest命令将检查整个配置是否设置正确。

```
#cd/usr/local/apache2/bin
#./apachectl configtest
Syntax OK
#./apachectl start
./apachectl start:httpd started
```

如果一切工作正常，当通过一个Web浏览器连接到服务器时，将会看到类似于图A-1所示的输出。

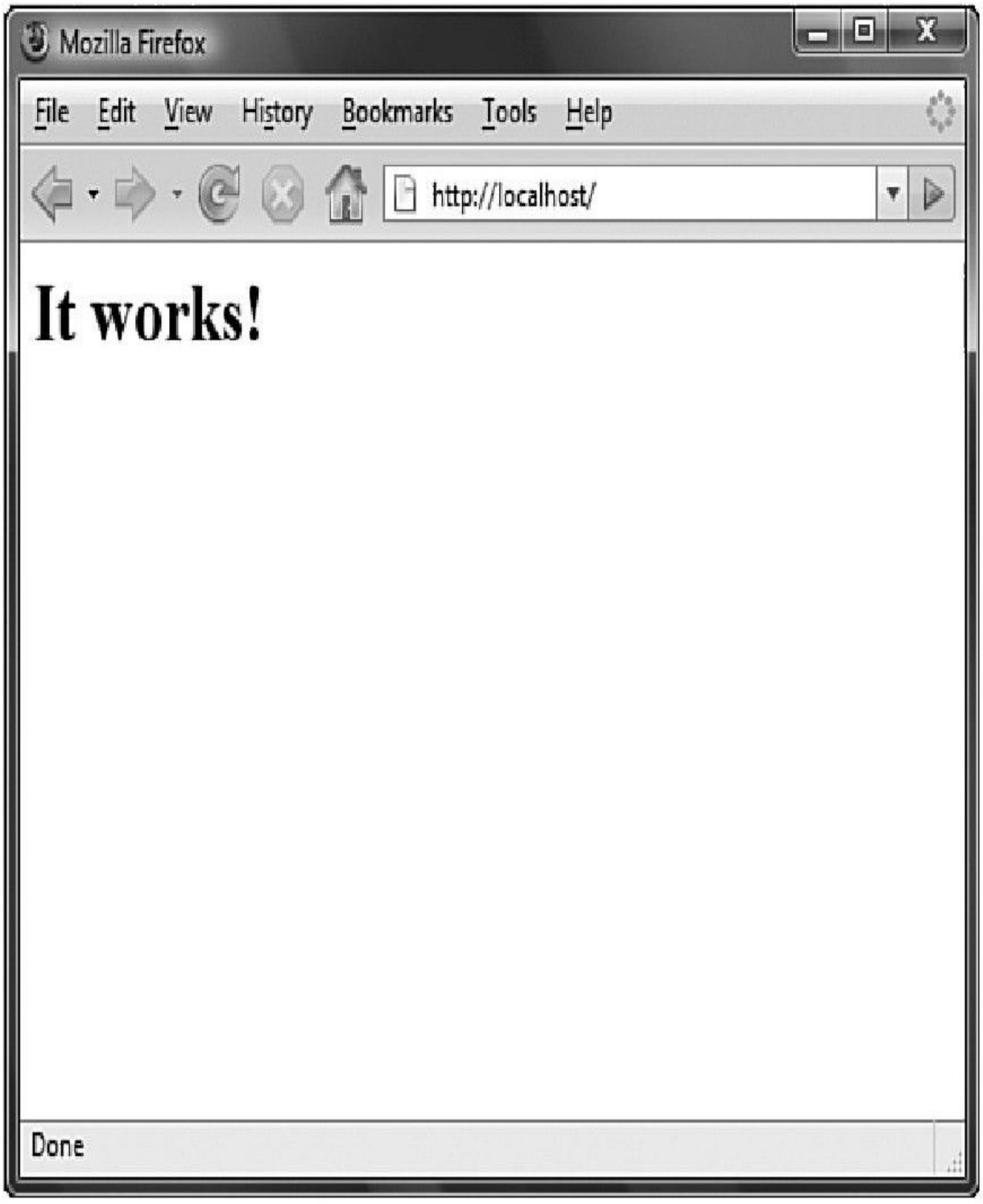


图　A-1　由Apache提供的默认测试页

提示　可以用域名或计算机的实际IP地址来连接服务器。检查这两种情况，以保证一切工作正常。

A.1.4 PHP支持是否工作正常

现在，我们来检查PHP支持。创建一个包含下列代码，名为test.php的文件。该文件需要放置在文档的根路径下，其默认设置为/usr/local/apache/htdocs。请注意，这是由我们在前面所选择的目录前缀确定的。但是，也可以在httpd.conf中修改它。

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

其输出结果如图A-2所示。

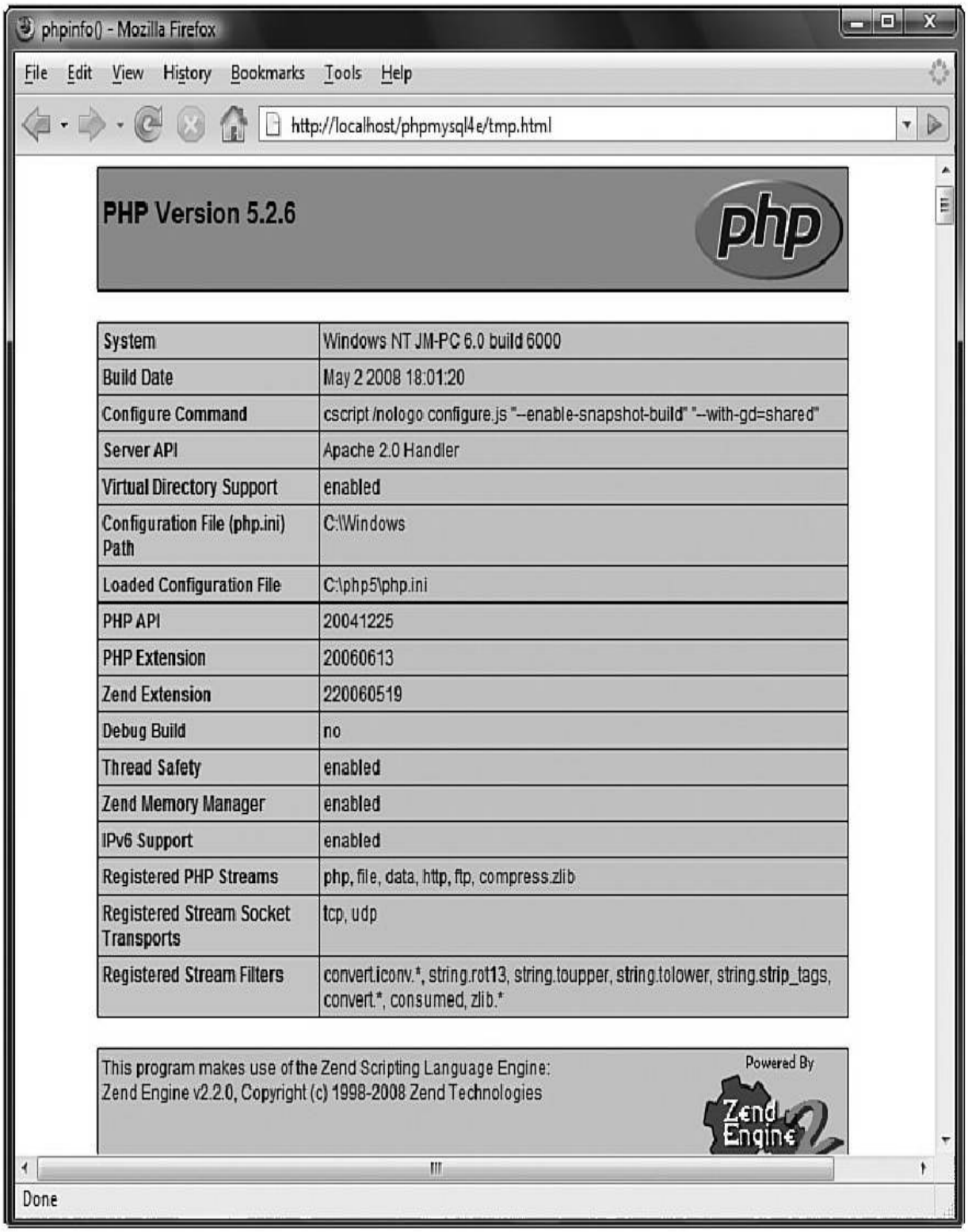


图　A-2　函数phpinfo()能够提供有用的配置信息

A.1.5　SSL是否工作正常

在Apache 2.2中，要启用SSL，你所要做的就是在httpd.conf文件中取消关于它的注释，原注释语句如下所示：

```
#Include conf/extra/httpd-ssl.conf
```

修改为:

```
Include conf/extra/httpd-ssl.conf
```

在httpd-ssl.conf文件中,你还可以对SSL进行更多配置修改；查看Apache的文档：http://httpd.apache.org/docs/2.2/mod/mod_ssl.html获得相关信息。

配置改变，可停止或启动服务器：

```
#/usr/local/apache2/bin/apachectl stop
#/usr/local/apache2/bin/apachectl start
```

要检查SSL是否正常工作，可以通过一个Web浏览器并且选用https协议连接到服务器，使用如下所示的URL：

```
https://yourserver.yourdomain.com
```

也可以按如下方式输入服务器的IP地址，如下所示：

```
https://xxx.xxx.xxx.xxx
```

或

```
http://xxx.xxx.xxx.xxx:443
```

如果SSL正常工作，该服务器将发送证书到该浏览器从而建立一个安全的连接。这样，将使得浏览器提示接受亲笔签名的证书。如果它是一个来自证书验证机构的证书，那么浏览器已经信任了该证书，因此也就不会再提示。在我们的例子中，我们创建并签名了自己的证书。我们并不想立刻购买一个，因为希望首先保证所有东西都能正常运行。

如果我们正在使用IE或FireFox，会在状态条上看到一个挂锁标志。这就告诉我们浏览器与服务器之间已经建立了一个安全连接。FireFox使用的图标如图A-3所示。通常，该图标位于浏览器的左下角或右下角。



图　A-3　Web浏览器显示了一个表示当前浏览的页面是来自SSL连接的图标

要作为共享对象使用你所安装的PHP，你还需要完成一些步骤。

首先，将已经编译的模块文件复制到PHP扩展目录中。这个目录可能是：

```
/usr/local/lib/php/extensions
```

在php.ini文件中，添加如下行：

```
extension=extension_name.so
```

在对php.ini文件修改后，你必须重起Apache服务器。

A.2　在Windows下安装Apache、PHP和MySQL

在Windows环境下，这些软件的安装过程稍有不同，因为PHP要么以CGI（php.exe）脚本方式进行安装，要么作为一个SAPI（php5isapi2_2.dll）模块进行安装。但是，Apache和MySQL安装方法与在UNIX下的安装方法类似。在Windows环境下安装之前，请确认已经在该机器上安装了操作系统最新的服务补丁。

提示　PHP 5.3已经放弃了对任何早于Windows 2000版本的支持；如今，它只支持Windows2000、Windows Server 2003、Windows Server 2008以及Windows Vista（和未来版本）。

如果网络连接速度较慢，可以选择使用CD版本，但是CD版本相对来说不一定是最新的版本。

A.2.1　在Windows下安装MySQL

我们给出的如下安装指南是针对Windows Vista操作系统的。

首先开始设置MySQL。可以从如下所示站点下载所需的*.msi文件：http://www.mysql.com。

双击该文件开始安装操作。

安装过程的前几个向导风格界面包含了安装和MySQL许可的常规信息。请查看这些屏幕并且点击"Continue"按钮继续安装过程。你将遇到的第一个重要选择是安装类型——“Typical（常规），“Compact（简洁）以及Custom（自定义）”。常规类型的安装就足够了，因此保持默认选择并点击"Next"按钮继续安装。

当完成安装后，继续执行MySQL配置向导，根据自己的需求，创建一个自定义my.ini文件。要开始MySQL配置向导，选中“Configure MySQL Server Now（现在配置MySQL服务器）”复选框，点击"Finish"按钮。

在MySQL配置向导过程中，选择合适的配置选项；如果需要关于这些配置选项的详细介绍，请参阅MySQL手册：http://dev.mysql.com/doc/refman/5.0/en/index.html。当完成配置后（包括root用户的密码）该向导将启动MySQL服务。

安装服务器后，你可以停止、启动或者通过Windows系统的服务工具（在控制面板可以找到）设置为自动启动。要启动服务，点击“开始”，选择“控制面板”，双击“管理工具”，再双击“服务”。

图A-4所示的是服务工具。如果你要设置任何MySQL选项，你必须停止该服务，并指定其启动参数。重新启动MySQL服务。MySQL服务可以通过服务工具或使用命令行"NET STOP MySQL"或"mysqladmin shutdown"停止。MySQL提供了许多命令行工具。如果没有在Windows的PATH环境变量中正确设置MySQL运行文件的路径，运行这些命令行工具将会很困难。

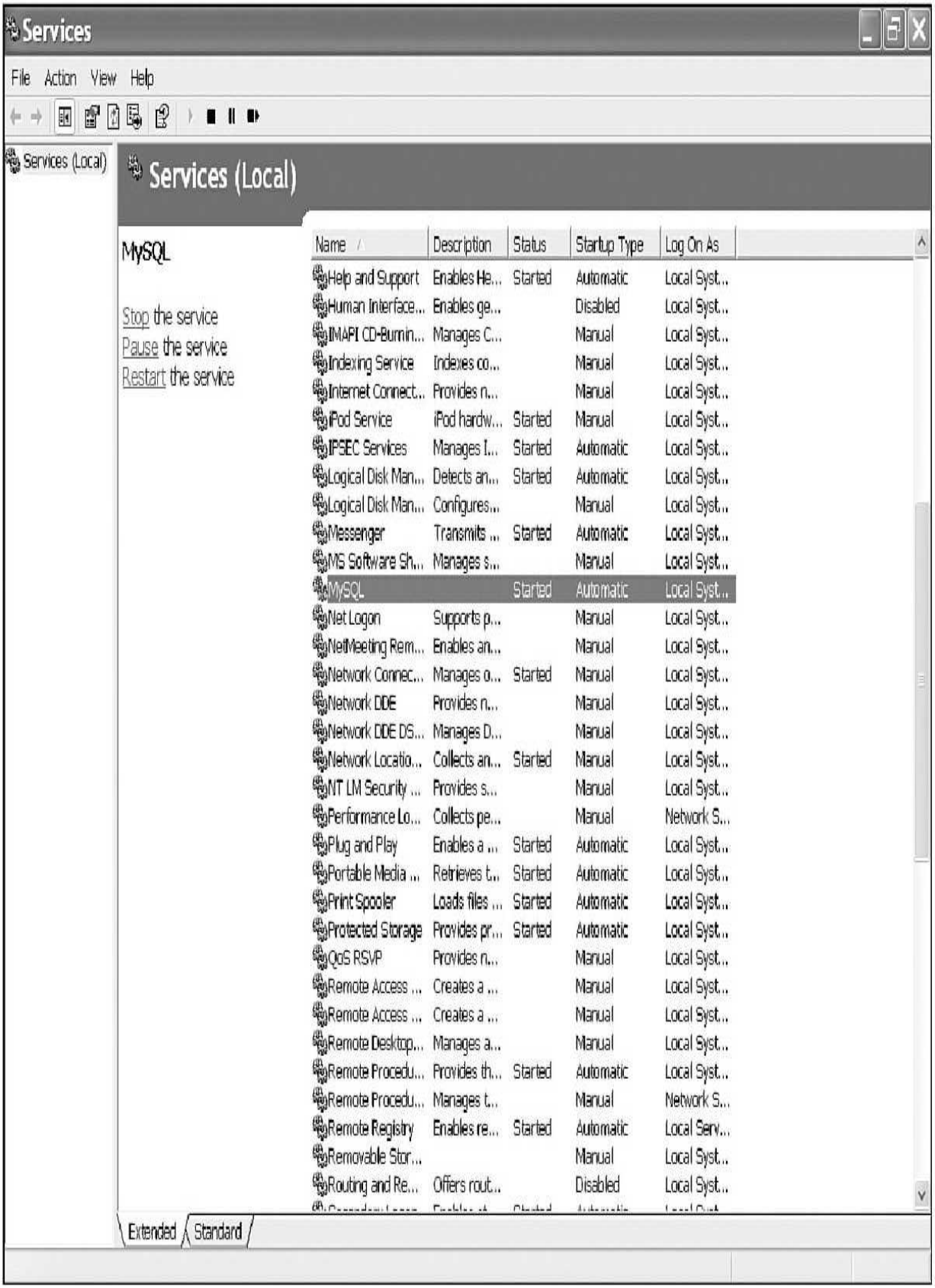

Services
File Action View Help
Services (Local)
Services (Local)
MySQL
Stop the service
Pause the service
Restart the service
Name
Description
Status
Startup Type
Log On As
Help and Support Enables He... Started Automatic Local Syst...
Human Interface... Enables ge... Disabled Local Syst...
IMAPI CD-Burnin... Manages C... Manual Local Syst...
Indexing Service Indexes co... Manual Local Syst...
Internet Connect... Provides n... Manual Local Syst...
iPod Service iPod hardw... Started Manual Local Syst...
IPSEC Services Manages I... Started Automatic Local Syst...
Logical Disk Man... Detects an... Started Automatic Local Syst...
Logical Disk Man... Configures... Manual Local Syst...
Messenger Transmits ... Started Automatic Local Syst...
MS Software Sh... Manages s... Manual Local Syst...
MySQL Started Automatic Local Syst...
Net Logon Supports p... Manual Local Syst...
NetMeeting Rem... Enables an... Manual Local Syst...
Network Connec... Manages o... Started Manual Local Syst...
Network DDE Provides n... Manual Local Syst...
Network DDE DS... Manages D... Manual Local Syst...
Network Locatio... Collects an... Started Manual Local Syst...
NT LM Security ... Provides s... Manual Local Syst...
Performance Lo... Collects pe... Manual Network S...
Plug and Play Enables a ... Started Automatic Local Syst...
Portable Media ... Retrieves t... Started Automatic Local Syst...
Print Spooler Loads files ... Started Automatic Local Syst...
Protected Storage Provides pr... Started Automatic Local Syst...
QoS RSVP Provides n... Manual Local Syst...
Remote Access ... Creates a ... Manual Local Syst...
Remote Access ... Creates a ... Manual Local Syst...
Remote Desktop... Manages a... Manual Local Syst...
Remote Procedu... Provides th... Started Automatic Local Syst...
Remote Procedu... Manages t... Manual Network S...
Remote Registry Enables re... Started Automatic Local Serv...
Removable Stor... Manual Local Syst...
Routing and Re... Offers rout... Disabled Local Syst...
Extended
Standard

图　A-4　“服务”工具允许配置运行在机器上的服务

许多在Windows命令提示符下能够使用的常见命令，例如dir和cd，都内置在cmd.exe命令中。其他的命令，例如format和ipconfig，都有自己的可执行文件。输入"C:\winnt\system32\format"来执行格式化一个硬盘的命令是非常不方便的。同样，输入"C:\mysql\bin\mysql\"来运行MySQL服务程序也是不方便的。

包含Windows基本命令可执行文件（例如format.exe）的目录自动保存在PATH环境变量中，这样只要输入"format"就可以执行这个程序。要使mysql命令行工具具有同样的方便性，我们也应该将其添加到PATH环境变量中。

点击“开始”并选择“设置”、“控制面版”。双击“系统”，选择“高级”选项卡。如果点击“环境变量”按钮，将看到一个对话框，在这个对话框中可以查看当前系统所有环境变量。双击"PATH"将允许对其进行修改。

在当前路径之后添加一个“;”号，可以将新路径与前面的路径相隔离，添加"C:\mysql\bin"。当点击“确认”按钮时，所添加的路径项将保存在机器的注册表中。当下一次重新启动机器时，就可以直接输入"mysql"而不用输入"C:\mysql\bin\mysql"。

A.2.2　在Windows下安装Apache

Apache 2.2能够在大多数Windows平台运行。比较Apache 1.3和2.0版本，它提供了大量性能改进和稳定性的提高。你可以通过源代码编译Apache，但是由于并不是很多用户具有编译器，本节主要介绍MSI安装程序版本。

访问http://httpd.apache.org站点并下载Apache 2.2 Windows安装程序。这里，下载apache_2.2.9-win32-x86-openssl-0.9.8hr2.msi文件。它包含了Apache的最新版本（2.2）以及OpenSSL 0.9.8版本。MSI文件是Windows安装程序使用的打包格式。

如果你并不是真正发现一个Bug或者对Apache开发工作感兴趣，你就不需要编译Apache的源代码。这个文件包含了可以直接安装和使用的Apache服务器。

双击要下载的文件开始下载过程。安装过程对你来说非常简单。如图A-5所示，安装程序与大多数Windows安装程序类似。



图　A-5　Apache安装程序操作简单

安装程序将提示输入如下所示的信息：

■网络名称、服务器名称和管理员的电子邮件地址。如果要安装一个可供实际使用的服务器，应该知道这些问题的答案。如果只是为了个人使用而安装这个服务器，这些答案并不是非常重要的。

■是否希望将Apache安装成一个服务。通常对于MySQL来说，这样的安装是非常简单的。

■安装类型。我们建议使用“完全安装”选项，但是如果不愿意安装其中某些组件，可以根据文档所述选择“自定义安装”。

■安装Apache的目录。（默认为C:\Program Files\Apache Software Foundation\Apache2.2）选择了这些选项后，Apache服务器将被安装并启动。

在Apache服务器启动后，它将侦听80端口（除非在配置文件中修改了端口号、侦听或绑定地址等指令）。要连接服务器并访问其默认页面，可以启动一个浏览器并且输入如下所示的URL：http://localhost/。

这个URL将显示一个类似于图A-1所示的欢迎页面。如果没有出现该页面或者看到了一个错误页面，可以查看logs目录下的error.log文件。如果主机没有连接到Internet，还可以使用如下所示的URL：http://127.0.0.1/。

这是一个表示本地主机的IP地址。

如果修改了侦听的端口，必须在URL字符串后添加":port_number"。

请注意，Apache无法与其他TCP/IP应用程序共享同一个端口。

可以从“开始”菜单来启动和停止Apache服务：Apache将其自身添加在“程序”->“Apache HTTP服务器”。在“控制Apache服务器”标题下，将发现可以启动、停止和重新启动服务器。

在安装Apache之后，可能需要编辑位于conf目录下的配置文件。我们将在安装PHP的时候讨论如何编辑配置文件httpd.conf。

A.2.3　在Windows下安装PHP

要在Windows下安装PHP，首先应该从如下所示的地址下载PHP文件：http://www.php.net。

对于Windows安装，必须下载两个文件。一个是包含PHP的zip文件（文件名有时类似于php-5.2.6-Win32.zip），而另一个是类库文件（pecl-5.2.6-Win32.zip或类似名称）。

将Zip文件解压到所选择的目录。常见的位置为C:\PHP，我们将在如下的介绍中使用这个目录。

将PECL文件解压到扩展目录，可以安装PECL库。使用C:\PHP作为基目录，扩展目录可以是C:\PHP\ext\。

现在，执行以下操作：

1）在主目录中，将看到一个名为php.exe和一个名为php5ts.dll的文件。这些是将PHP以CGI模块方式运行所必需的文件。如果希望以SAPI模块方式运行PHP，可以使用针对Web服务器的DLL文件。在这里，该文件就是php5apache.dll。

SAPI模块的运行速度更快，而且更容易实现更高的安全性；CGI版本支持从命令行方式运行PHP，这一切都是由我们来决定的。

2）设置php.ini配置文件。PHP本身自带有两个已经准备好的文件：php.ini-dist和php.ini-recommended。学习PHP或在开发服务器上使用PHP时，我们建议使用前者。

但是如果在一个产品服务器，建议使用后者。复制该文件，并将其重命名为php.ini。

3）编辑php.ini文件。在该文件中，有许多可供设置的指令选项，目前，大多数选项都可以忽略。需要修改的设置包括：

■将extension_dir指令修改为指向保存了扩展DLL目录的位置。在常规安装中，这个目录通常是C:\PHP\ext。因此，php.ini文件将包含：

```
extension_dir=c:/php/ext
```

■将doc_root指令修改为指向Web服务器的根路径。如果使用的是Apache，这个目录通常都是：

```
doc_root="c:/Program Files/Apache Software Foundation/Apache2.2/htdocs"
```

■也可以选择运行一些其他扩展。在这一步，我们建议只运行那些PHP工作所必需的扩展。可以根据需要添加扩展。要添加一个扩展，可以查看“Windows扩展”项下的列表。

我们将发现许多行代码，例如：

```
;extension=php_pdf.dll
```

要启用这些扩展，只需删除语句前面的分号（添加分号，可以注释掉这行语句）。请注意，如果希望添加更多的扩展，应该在修改了php.ini文件后重新启动Web服务器，这样可以使这些修改生效。

在本书中，将使用php_pdf.dll、php_gd2.dll、php_imap.dll和php_mysqli.dll文件。必须取消这些扩展前的注释符号。我们可能还会发现，php_mysqli.dll文件项不存在，如果这样，可以使用如下所示命令添加：

```
extension=php_mysqli.dll
```

保存并关闭这个文件。

4）如果使用NTFS，请确认运行Web服务器的用户身份具有读php.ini文件的权限。

1.将PHP添加到Apache配置中

我们可能会需要编辑Apache的其中一个配置文件。在我们喜欢的编辑器中打开httpd.conf文件。通常，这个文件保存在"C:\Program Files\ApacheSoftware Foundation\Apache2.2\conf\"目录中。打开该文件并找到如下语句行：

```
LoadModule php5_module c:/php/php5apache2_2.dll
PHPIniDir"c:/php/"
AddType application/x-httpd-php.php
```

如果这些语句行不存在，可以在文件的最后手动添加这些语句，保存并关闭该文件，重新启动Apache服务器。

2.测试你的设置

接下来就是启动你的Web服务器，测试并确保你的PHP能够正常工作。创建一个test.php文件并且添加如下代码行：

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

确认该文件保存在文档根目录（通常是C:\Program File\Apache Software Foundation\Apache2.2\htdocs）；在浏览器中访问该文件，如下所示：

```
http://localhost/test.php，
```

或者：

```
http://your-ip-number-here/test.php。
```

如果你看到如图A-2所示的输出，你就知道你的PHP已经正常工作。

A.3 PEAR安装

PHP5应该附带了PHP扩展及应用库（PEAR）安装程序包。如果使用的是Windows，可以在命令行下使用如下所示的命令：

```
c:\php\go-pear
```

go-pear脚本将询问一些非常简单的问题，主要是关于安装程序包和标准的PEAR类的安装位置，然后将开始下载并安装PEAR（在Linux下，第一个步骤并不是必需的，但是安装的其他部分是相同的）。

在这一步中，应该有一个PEAR安装程序的安装版本和基本的PEAR库。可以输入如下所示的命令进行包的安装：

```
pear install package
```

其中，必须使用希望安装的包名来替换"package"字符串。

要知道所有可供安装的包名称，输入如下命令：

```
pear list-all
```

要查看当前系统已经安装的PEAR包，输入如下命令：

```
pear list
```

要安装在第30章中所介绍的MIME邮件包，输入如下命令：

```
pear install Mail_Mime
```

在第11章中所介绍的DB也应该被安装，输入如下命令：

```
pear install MDB2
```

如果要查看任何已经已经安装包的最新版本，使用如下命令：

```
pear upgrade pkgname
```

如果由于某些原因，以上过程无法实现PEAR的正确安装，我们建议访问http://pear.php.net/packages.php站点并且直接下载PEAR包。

从这个站点可以浏览不同的可供使用的包。例如，在本书中，我们所使用的Mail_Mime。

浏览该包的页面，点击“下载最新版本”，可以下载最新版本。需要对所下载的文件解压缩，将其保存在include_path路径中。你应该创建或者已经拥有C:\pho\pear目录。如果你要手动下载这些包，我们建议将这些包存在PEAR目录树中。PEAR具有标准结构，因此我们建议将软件包放置在标准位置；这也是安装程序保存这些包的位置。例如，Mail_Mime包属于Mail部分，因此，在这个例子中，我们将其保存在C:\php\pear\Mail目录中。

A.4　设置其他配置

你可以在其他Web服务器中设置PHP和MySQL，例如Omni、HTTPD和Netscape的企业服务器。在本附录中没有介绍这些服务器，但是可以在如下MySQL和PHP的Web站点中找到如何在这些服务器中设置它们的信息：

```
http://www.mysql.com
```

和

```
http://www.php.net
```

附录B　Web资源

本附录列出了一些在Web上可供使用的资源，可以通过这些资源找到一些教程、文章、新闻和PHP的示例代码。这些资源仅仅是Web上可供使用资源的一小部分。显然，可用的资源远远不只是本附录所给出的这些，而且，随着Web开发者越来越多地使用和熟悉PHP和MySQL，新的资源将会不断涌现。

有些资源可能是用其他语言编写的，比如，德语、法语或其他不是我们母语的语言。因此建议使用翻译器，比如http://www.systransoft.com，这样就可以使用母语浏览这些Web资源。

B.1　PHP资源

PHP.Net（http://www.php.net）PHP的源站点。可以在这里下载PHP的所有源代码和手册的副本，浏览邮件列表存档以及获得PHP的最新消息。

Zend.Com（http://www.zend.com）实现PHP的Zend引擎源代码。一个包括了论坛、文章、教程以及可供使用的示例类和代码的数据库的门户站点。

PEAR（http://pear.php.net）PHP扩展及应用库。这是PHP的正式扩展站点。

PECL（http://pecl.php.net）PEAR的姊妹站点。PEAR提供了用PHP编写的类；PECL（发音为"pickle"）提供了用C编写扩展。PECL类有时候更难安装，但是它可以实现大量功能并且通常比基于PHP的代码功能更强大。

PHPCommunity（http://www.phpcommunity.org/）一个基于社区的新站点。

php|architect（http://www.phparch.com/）一个PHP杂志。这个站点提供了免费的文档，我们可以订阅并接收PDF格式或印刷品格式的杂

志。

PHP Magazine（http://www.phpmag.net/）另一个PHP杂志，也可以通过电子方式或印刷品方式获得。

PHPWizard.net（http://www.phpwizard.net）一个能够提供许多有趣的PHP应用程序的站点，例如，phpChat和phpIRC。

PHPMyAdmin.Net（http://www.phpmyadmin.net/）MySQL的基于PHP的Web站点。

PHPBuilder.com（http://www.phpbuilder.com）PHP教程的门户站点。在这里，可以找到任何希望得到的指南。该站点还有一个论坛及一个消息板供人们提问。

DevShed.com（http://www.devshed.com）该站点是一个门户站点，它提供了关于PHP、MySQL、Perl和其他开发语言的优秀教程。

PX-PHP Code Exchange（http://px.sklar.com）该站点是学习PHP的、优秀的入门站点。在这里，可以找到很多示例代码及函数。

The PHP Resource（http://www.php-resource.de）一个提供了非常不错的教程、文章和脚本的站点。唯一的问题是该站点使用德语。我们建议使用翻译服务站点来浏览它。也可以使用这种方法来阅读示例源代码。

WeberDev.com（http://www.weberDev.com）也就是我们以前所知道的Berber's PHP示例页面，该站点从一无所有迅速发展成一个拥有众多教程和示例代码的站点。该站点主要面向PHP和MySQL用户，同时也涉及一些关于安全及通用数据库的问题。

HotScripts.com（http://www.hotscripts.com）一个脚本程序的分类精选站点。该站点涵盖了多种脚本语言，例如：PHP、ASP.NET以及Perl，此外，还包含大量优秀的PHP脚本集。

该站点更新频率很高，如果要寻找某些脚本，这个站点是必看的。

PHP Base Library（http://phplib.sourceforge.net）面向开发大规模PHP项目开发人员的站点。除模板制作和数据库抽象外，还提供了一个包含许多会话管理工具的软件库。

PHP Center（http://www.php-center.de）又一个德语门户站点，它提供了教程、脚本、技巧、广告等内容。

PHP Homepage（http://www.php-homepage.de）又一个德语站点，它提供许多PHP脚本、文章、新闻以及其他内容等，此外还包括一个快速参考部分。

PHPIndex.com（http://www.phpindex.com）一个非常不错的法语门户站点，它包含了大量关于PHP的内容。该站点包括新闻、常见问题解答、文章、工作列表等。

WebMonkey.com（http://www.webmonkey.com）一个提供了诸多Web资源的门户站点，其中包括教程、示例代码等。此外，该站点还涉及设计、编程、后台和多媒体等内容。

The PHP Club（http://www.phpclub.net）PHP俱乐部为PHP初学者提供了许多资源，该站点为初学者提供了新闻、图书简介、示例代码、论坛、常见问题回答以及许多教程。

PHP Classes Repository（http://phpclasses.org）该站致力于用PHP编写的、可供免费使用的类的发布。如果我们正在开发代码或者项目由类组成，必看该站点。它具有良好搜索功能，因此可轻易地找到想要的内容。

The PHP Resource Index（http://php.resourceindex.com）它是一个文档、脚本和类的门户站点。该站点最酷的地方是组织结构非常清晰，可以为我们节省时间。

PHP Developer（http://www.phpdeveloper.org）另一个提供PHP新闻、文章和教程的站点。

Evil Walrus（http://www.evilwalrus.com）一个看起来很酷的PHP脚本门户站点。

SourceForge （http://sourceforge.net）丰富的开放源代码资源。Source Forge不仅可以找到有用的代码，而且还为开放源代码开发者提供CVS访问、邮件列表和机器。

Codewalkers（http://codewalkers.com/）该站点包含了文章、图书评论、教程以及饶有趣味的PHP竞赛。该站点每两周组织一次代码竞赛。

PHP开发人员网络统一论坛（http://forums.devnetwork.net/index.php）该站点讨论所有与PHP相关的内容。

PHP kitchen（http://www.phpkitchen.com/）关于PHP的文章、新闻和宣传。

Postnuke（http://www.postnuke.com/）一个经常使用的PHP内容管理系统。

PHP应用程序工具（http://www.php-tools.de/）一个有用的PHP类集合。

Codango（http://www.codango.com/php/）这是一个关于PHP Web应用、函数库、脚本、主机服务、教程以及其他信息的站点。

B.2 MySQL和SQL的特定资源

MySQL站点（http://www.mysql.com）MySQL的正式站点。它提供了一些非常优秀的文档、支持和信息。如果正在使用MySQL，尤其是对于需要那些开发环节和邮件列表存档的开发人员来说，该站点是必读的。

The SQL Course（http://sqlcourse.com）通过简单易懂的说明来介绍SQL的教程。它允许通过一个在线SQL解释器来练习所学到的内容。该解释器的高级版本可以在http://www.sql-course2.com找到。

SearchDatabase.com （http://searchdatabase.techtarget.com）一个提供了许多关于数据库有用信息的门户站点。它提供了非常不错的教程、技巧、白皮书、FAQ、图书评论等内容。该站点是必读的。

B.3 Apache资源

Apache Software（http://www.apache.org）如果需要下载Apache Web服务器的源代码或二进制程序，可以从这里开始。该站点还提供在线文档。

Apache Week（http://www.apacheweek.com）在线周刊，它为任何运行Apache服务器或Apache服务的人提供所有必需的信息。

Apache Today（http://www.apachetoday.com）一个关于Apache的每日新闻、源代码及信息更新的站点。用户必须订阅才能发帖子。

B.4 Web开发

Philip和Alex的Web发布指南（http://philip.greenspun.com/panda/）一本关于用于Web软件工程的、幽默和谦逊的指南。关于该主题的书只有几本，其中一本是由他们和Samoyed合著的。

光盘内容

光盘下载地址：http://pan.baidu.com/s/1ntmHkUl