China-pub.com
下载

China-pub.com
下载

第7章　动态内容Web服务

本章讲述将活跃内容展示到Web上的两种主要技术 —— Apache ModPerl和PHP。

由于信息系统需要全球的可视性，所以Web成为了一种显而易见的选择。Web是如此的普遍存在，以至于它简直无可替代。超文本标记语言（Hypertext Markup Language，HTML）在功能上却比较弱 —— 因为它是静态的。它是一种可以用来共享文档的方法，但不是一种交互式的方式。随着混合装置需求的增加，开发者都在集中精力解决Web的静态特性问题。Java、ActiveX和XML等技术就是最近一些扩展Web能力的尝试。

Perl是一种文本处理语言，它可以用来开发扩充数据库和ModPerl —— 包含在Apache Web服务器中一种特别的Perl版本。PHP语言由C语言和Awk语言派生而成，它对提高SQL数据库的使用效率经常有很好的效果。

本章所讲述的内容主要适应于运行Red Hat Linux 6.0的基于Intel的系统。其中许多信息对其他的平台也是适应的，而且，此处讨论的所有软件（Sybase、PostgreSQL和PHP 3.0）都有移植到各种UNIX的实现版本。

7.1　PHP

PHP（Professional Home Page）是PHP/FI的后继软件，它是一种Web工具，并且具有一定的流行性。PHP允许一种带有SQL扩充的类似于C的语言直接嵌入到HTML文档中，并将它们转换成活跃的内容。该软件近来进行了完整的重新编写以提高性能。PHP不需要浏览器进行任何扩充，它依赖于CGI接口。

PHP支持大量的数据库服务器，包括Sybase和PostgreSQL。所支持的其他软件包括Adabas、mSQL、MySQL、Oracle和ODBC。PHP现在还支持LDAP目录服务和IMAP邮件协议。关于PHP的更多信息包含在开发者站点（http://www.php.net）的文档中。

7.2　安装PHP

在PHP的Web站点和本书所附光盘的sybase/php-3.0.12.tar.gz中都有PHP。用户必须使用该文件，或者该文件的后续版本（如果存在的话）。

有两种方法可以配置PHP。它可以设置成一种CGI二进制的方式，或者在支持的Web服务器（Apache、Netscape Server和Microsoft IIS）运行时以模块方式装载。

以Apache模块的方式装载PHP有很多好处。通过CGI的途径，浏览器每次访问服务器上的PHP时，系统装载整个的PHP解析器并执行，然后终止。此外，如果以模块方式装载，则PHP解析器成为了Apache运行环境的一个集成部分。因为PHP的内存映像不是在每次访问时都被装载和卸出，所以模块化方式的装载在性能上也就更好。在放置和访问PHP源文件时，模块化配置能带来更大的灵活性。当以模块方式使用时，在URL中不需要引用cgi-bin目录；PHP应用程序文件可以放置在Web服务器文档根目录的任何位置中。

但是，Red Hat发布的Apache Web服务器有许多安全变更。每进行一次变更，PHP使能的

China-pub.com
下载

Web服务器根据源文件重新生成。这种重新生成是乏味的。如果系统升级成新版本的 Red Hat Linux，则可能必须重新生成Web服务器。

在将PHP编译成CGI二进制时，PHP生成的方式是与Web服务器安装完全分离的。Web服务器无需担心扰乱PHP的工作。这种配置适宜于不是很忙的Web站点。

但是，以CGI二进制方式运行PHP会带来一个严重的安全问题。Apache允许使用文件（缺省方式下使用的名称为.htaccess）限制对Web页面的访问，其根据是特定的IP地址或提供有效的用户名。PHP的CGI版本不认识.htaccess文件。如果安装成CGI二进制的方式，它可以在服务器文档根目录或用户目录（在用户主帐户的 public_html目录中）中读取所有的 Web页面，而不管在.htaccess中指定的访问控制。幸运的是，PHP的Apache模块版本并没有这样的问题。

最后，那些对Web服务器目录树的安全不很在意的用户，以及那些不想使用 PHP使能Web页面的用户应该安装解析程序的 CGI版本。其他用户则应该多花费一些时间来将 PHP安装成Apache的模块。

本章假设使用更加安全的方式（也更加费劲），使用PHP使能的Web服务器的驻留程序。使用PHP作为Apache模块所需的步骤在下面讲述。清单 7-1中列出的脚本必须以根用户方式运行。假设PHP的二进制发布（php-3.0.12.tar.gz）和Apache的源RPM位于当前的目录中，建立过程的一个比较好的开始之处是根用户的主目录（ /root，或者在BASH中称为~root）。这是用来建立Apache的一个临时空间，在新的驻留程序安装之后可以删除该空间。 PHP和Apache源文件都包含在本书所附光盘的 sybase目录中。我们并不使用 RPM的建立方法，因为它不是十分的方便。而二进制 RPM版本的mod_php缺乏Sybase的支持。在RPM之外建立的源文件使得建立的区域可以局部化。

在PHP配置中，有一些重要的数据库选择选项。在清单 7-1中，脚本包含了对 Sybase和PostgreSQL的支持。要删除对PostgreSQL的支持，只需要删除--with-pgsql指令。要删除对Sybase的支持，则需要删除--with-sybase-ct指令。如果Sybase没有安装在/home/sybase目录中。（就像本章前面所建议的那样），则在选项中必须指定安装目录（例如，--with-sybase-ct=/opt/sybase）。

清单7-1 建立带有PHP的Apache

```
#!/bin/sh
/etc/rc.d/init.d/httpd stop
mkdir apache
cd apache
rpm2cpio ../apache-1.3.6-7.src.rpm | cpio -i
tar xvzf apache_1.3.6.tar.gz
cd apache_1.3.6
for x in ../*.patch
do
     patch -p1 < $x
done
CFLAGS="-O2" LDFLAGS="-s" ./configure --prefix=/usr \
               --with-layout=RedHat \
               --enable-module=all \
               --disable-rule=WANTHSREGEX \
               --disable-module=auth_dbm \
               --with-perl=/usr/bin/perl
cd ../..
tar xvzf php-3.0.12.tar.gz
```

China-pub.com 下载

```
cd php-3.0.12
# Remove the "--with-pgsql" line below to disable PostgreSQL.
# Remove the "--with-sybase-ct" line below to disable Sybase support.
LDFLAGS="-s" CFLAGS="-O3 -I/usr/include/pgsql" ./configure --disable-debug
\
                --with-system-regex \
                --with-pgsql \
                --with-sybase-ct \
                --with-apache=../apache/apache_1.3.6
make
make install
cd ../apache/apache_1.3.6
CFLAGS="-O2" LDFLAGS="-s" ./configure --prefix=/usr \
                --with-layout=RedHat \
                --enable-module=all \
                --disable-rule=WANTHSREGEX \
                --disable-module=auth_dbm \
                --with-perl=/usr/bin/perl \
                --activate-module=src/modules/php3/libphp3.a
make
cd src
cp -f httpd /usr/sbin
echo 'AddType application/x-httpd-php3 .php3' >> /etc/httpd/conf/srm.conf
echo 'AddType application/x-httpd-php3 .phtml' >> /etc/httpd/conf/srm.conf
cd /etc/httpd/conf
sed 's/^LoadModule/#LoadModule/
s/^ClearModule/#ClearModule/
s/^AddModule/#AddModule/' httpd.conf > httpd.conf.avecphp
mv httpd.conf httpd.conf.sansphp
mv httpd.conf.avecphp httpd.conf
cd
/etc/rc.d/init.d/httpd start
```

上述代码的运行需要一段比较长的时间（从 10min到60min，这要依赖于系统的速度）。该脚本的一个拷贝存储在本书所附光盘的 sybase/PREP-APACHE.sh文件中。

如果需要PHP的CGI配置，则需要以根用户方式运行清单 7-2所示的命令。

清单7-2 建立单独（CGI）的PHP

```
tar xvzf php-3.0.12.tar.gz
cd php-3.0.12
# Remove the "--with-pgsql" line below
# to disable PostgreSQL.
# Remove the "--with-sybase-ct" line below
# to disable Sybase support.
LDFLAGS="-s" CFLAGS="-O3 -I/usr/include/pgsql" ./configure --disable-debug \
                --with-pgsql \
                --with-sybase-ct \
                --with-system-regex
make
cp php /home/httpd/cgi-bin
```

7.3 在Web中使用SELECT

本节假设读者对 HTML有了一定的了解。如果不是这样的话，读者可能需要看一下关于

China-pub.com
下载

HTML的NCSA初学者指南 —— 其站点的地址为 http://www.ncsa.uiuc.edu/General/Internet/WWW/ HTMLPrimer.html.。

如果读者安装了PHP使能的Web服务器，并且输入了前一章所讲述的数据库命令，那么就可以使用SQL建立Web页面了。

PHP与其早期软件PHP/FI都使用类似于C语言的编程结构，并将这种语言嵌入到HTML中，PHP是作为一个文档预处理器来使用的（除了功能更加强大之外，它与 C编译器的预处理器十分相似）。

如果读者正在使用PostgreSQL，那么将清单7-3中的HTML拷贝到自己系统的/home/httpd/html/dbprint.php3文件中。

清单7-3 PHP中简单的PostgreSQL选择

```
<HTML>
<HEAD>
<TITLE>View Database Records</TITLE>
</HEAD>
<BODY>

<DIV ALIGN="center">

<P>View Database Records</P>

<TABLE BORDER>

<TR>
<TH>item</TH>
<TH>vendorname</TH>
<TH>quantity</TH>
</TR>

<?PHP

        $conn = pg_Connect("", "", "", "", "shopping");

        if (!$conn) {
                echo "</table>An error occurred.\n";
                exit;
        }

    $result = pg_Exec($conn,
       "SELECT list.item, vendors.vendorname, list.quantity
               FROM list, vendors
               WHERE list.vendorcode = vendors.vendorcode
                       ORDER BY list.item;");

        if (!$result) {
                echo "</table>An error occurred.\n";
                exit;
        }

        $num = pg_NumRows($result);
        $i = 0;
```

China-pub.com 下载

```
while ($i < $num) {
        echo "<TR><TD>";
                echo pg_Result($result, $i, "item");
                echo "</TD><TD>";
                echo pg_Result($result, $i, "vendorname");
                echo "</TD><TD>";
                echo pg_Result($result, $i, "quantity");
                echo "</TD></TR>";
                $i++;
        }

        pg_FreeResult($result);
        pg_Close($conn);
?>

</TABLE>

</BODY>
</HTML>
```

上述的PHP代码检索出vendorname，并将其按照字母顺序打印在浏览器中。

注意，上述SELECT语句的ORDER BY子句，如果“ vendors ”表变得很大的时候，vendors.vendorname上的索引就很有用。

SQL使能的文档可以通过URL http://127.0.0.1/dbform.php3浏览到（对于PHP的CGI实现，其URL为http://127.0.0.1/cgi-bin/php/dbform.php3）。正式域名可用来替代本地主机，以便通过浏览器在不同的主机上浏览页面。

View Database Records

item	vendorname	quantity
African Violet	ACME Plant Store	1
Ice Cream	Super Grocer	1
Napkins	General Department Store	50
Root Beer	Super Grocer	3
Spark Plugs	General Auto Parts	4

图7-1 dbprint.php3的输出

假设所有的工作都很正常，并且用户根据前一章所述内容建立了数据库，加入了记录，则浏览器上的显示将如图7-1所示。

PHP还有一条指令，它与许多通用的Web服务器所支持的Server Side Include（SSI）很相似。如果在PHP块中遇到了include（"/some/path/to/a/file.html"），则指定的文件将由PHP插入并解析。甚至还有一个phpIncludePath变量，可以设置该变量，以便Web开发者分配路径名。传统的SSI指令在PHP的CGI版本中不能处理。在SSI中可以包含其他主机的URL位置，如include（"http://www.redhat.com"）。

PHP的CGI版本可以将用户主目录中的文件以下面的URL形式进行装载：

http://127.0.0.1/cgi-bin/php/~luser/dbprint.php3

上述HTML示例中的算法和语法都比较简单。它们几乎是直接从PHP/FI文档继承过来的。注意“ <?PHP和?> ”标志用来包含非HTML语言。它们标示PHP语言语句的开始和结束。任何时候，用户都可以将PHP语句插入到HTML文件的分界符中。

这些PHP语句进行下面的工作：

pg_Connect ( )：建立杂货数据库与本地服务器的连接。 PostgreSQL数据库服务器可能放置在一个分离的主机上。如果是这样一种情况，则目标主机应在此处输入。

pg_Exec ( )：执行SQL查询，并将结果存储在名为$result的变量中。

China-pub.com
下载

pg_NumRows ( )：返回前一个SELECT语句从表中检索出的行数。

while loop：检索并打印（使用 echo）出结果，使用 pg_Result ( )函数调用一行一行地显示。

pg_FreeResult ( )：废弃前一个pg_Exec的内容，这在运行多个连续的大查询时十分重要。

pg_Close ( )：关闭连接并释放缓冲内存。

如果使用的是 PHP的二进制版本，则该二进制软件可以用作一个脚本解释器，并可以使用Shell进行调用。考虑前面脚本的一种变化形式，如清单 7-4所示。

清单7-4 在Shell上运行PHP脚本

```
#!/usr/local/bin/php -f
<?PHP

        $conn = pg_Connect("", "", "", "", "shopping");

        if (!$conn) {
                echo "An error occurred.\n";
                exit;
        }

    $result = pg_Exec($conn,
       "SELECT list.item, vendors.vendorname, list.quantity
              FROM list, vendors
              WHERE list.vendorcode = vendors.vendorcode
                       ORDER BY list.item;");

        if (!$result) {
                echo "An error occurred.\n";
                exit;
        }

        $num = pg_NumRows($result);
        $i = 0;

        while ($i < $num) {
                echo pg_Result($result, $i, "item");
                echo "\t";
                echo pg_Result($result, $i, "vendorname");
                echo "\t";
                echo pg_Result($result, $i, "quantity");
                echo "\n";
                $i++;
        }

        pg_FreeResult($result);
        pg_Close($conn);
?>
```

如果PHP的CGI版本放置在/usr/local/bin中，则这些脚本可以在 Shell中直接执行。基于这样的原因，管理员经常既将 PHP建立成Apache模块的方式，还将其建立成 CGI的方式。特别要注意，/usr/local/bin目录不能被Web服务器访问，所以没有 CGI的安全问题。如果用户正在安装用于Shell脚本的PHP CGI版本，则将一个拷贝放置到 /home/httpd/cgi-bin中，或其他的允许

China-pub.com 下载

Web服务器访问的位置。清单7-5示出对一个Sybase数据库的调用是如何执行的。

清单7-5　PHP中简单的Sybase选择

```
<HTML>
<HEAD>
<TITLE>View Database Records</TITLE>
</HEAD>
<BODY>

<DIV ALIGN="center">

<P>View Database Records</P>

<TABLE BORDER>

<TR>
<TH>item</TH>
<TH>vendorname</TH>
<TH>quantity</TH>
</TR>

<?PHP
        error_reporting(1);

  $conn = sybase_connect("SYBASE", "nobody", "nobodypass");

        sybase_select_db("shopping",$conn);

        if (!$conn) {
                echo "</table>An error occurred.\n";
                exit;
        }

    $rc = sybase_query(
       "SELECT list.item, vendors.vendorname, list.quantity
               FROM list, vendors
               WHERE list.vendorcode = vendors.vendorcode
                        ORDER BY list.item",$conn);
        if (!$rc) {
                echo "</table>An error occurred.\n";
                exit;
        }

        while ($result = sybase_fetch_array($rc)) {

// The alternate syntaxes of the following two lines
// are both legal.

                echo "<TR><TD>$result[item]";
                echo "</TD><TD>" . $result["vendorname"];

                echo "</TD><TD>$result[quantity]";
                echo "</TD></TR>";
        }
```

China-pub.com
下载


```
        sybase_free_result($rc);
        sybase_close($conn);

?>

</TABLE>

</BODY>
</HTML>
```

上述Sybase示例的语法依赖于一个PHP函数fetch_array( )。fetch_array( )函数返回联合数组(associative arrays)，它使得$result [ ]数组元素以列名标识，而不是以数字偏移量标识。这种方法的唯一缺点是，如果两个表有同样名称的列，则第二个这样的列就不能用联合数组标识。然而，Sybase有一种方法可以重命名一个列：

```
SELECT "newname" = list.item, vendors.vendorname,
list.quantity
     FROM list, vendors
     WHERE list.vendorcode = vendors.vendorcode
          ORDER BY list.item
```

使用这样的Sybase SQL语法，“item”列在PHP中就可以用$results["newname"]标识了。

当活跃的数据库发生变更时（如使用上述的 sybase_select_db ("shopping", $conn)），Sybase还能在PHP中产生警告消息。error_report(1)可以抑制这些消息。可以设置Sybase登录缺省使用一个特定的数据库，方法是调用前一章讲述的 sp_modifylogin存储过程，从而可以忽略sybase_select_db ( )和error_reporting (1)调用。

7.4 在Web中使用INSERT

本节假设读者对 HTML表单有一定的了解。如果不是这样的话，读者可能需要看一下NCSA关于表单方面的资料，它们位于 http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/Docs/fill-out-forms/overview.html。

将记录插入到数据库中的HTML包含两个部分。首先，必须建立一个HTML表单，以便信息可以很容易地通过浏览器息输入。其次，信息必须被返回，并由 PHP处理，以便将其增加到表中。

表单的实现（尤其是 vendor字段）在设计时十分困难。 SELECT表单标志是最明显的HTML表单元素，可以用来展示一个预定义的 vendor列表。这样的列表可以硬编码到HTML中吗？如果另一个厂商增加到了“ vendors ”表中， HTML将不能自动更新，用户就不能根据新的厂商输入数据。厂商SELECT标志的表单元素可以在表单每次被访问时由 PHP产生。这种方法的缺点是增加了服务器的负载。

另一种方法是，每次“ vendors ”表被更新时，都产生用于表单的 HTML。这可用很多种方法加以实现，实际使用的一种方法是在 UNIX Shell提示符，或CGI环境下调用PHP的CGI版本。

第二种方法则是，数据表项表单的页面每次被访问时， PHP都使用一个SQL SELECT语句从“ vendors ”表中获取数据，以建立HTML的SELECT列表。

要实现这样的示例，将清单 7-6中的HTML拷贝到用户系统中的 /home/httpd/html/

China-pub.com 下载

dbform.php3文件中。

清单7-6 使用PostgreSQL准备表单

```
<HTML>
<HEAD>
<TITLE>Insert Database Record</TITLE>
</HEAD>
<BODY>

<DIV ALIGN="center">

<H1>Add Database Record</H1>

<FORM METHOD="post" ACTION="dbinsert.php3">

<TABLE BORDER="0">

<TR>
<TD>New Item:</TD>
<TD><INPUT NAME="item"></TD>
</TR>

<TR>
<TD>Vendor:</TD>
<TD><SELECT NAME="vendor" SIZE="1">

<?PHP

        $conn = pg_Connect("", "", "", "", "grocery");

        if (!$conn) {
                echo "An error occurred.\n";
                exit;
        }

        $rc = pg_Exec($conn,
                "SELECT vendorcode, vendorname
                        FROM vendors
                        ORDER BY vendorname;");

        if (!$rc) {
                echo "An error occurred.\n";
                exit;
        }

    $num = pg_Numrows($rc);
    $i = 0;
while ($i < $num) {
     echo "\n<OPTION VALUE=\"" .
          pg_Result($rc, $i, "vendorcode") .
          "\">" .
          pg_Result($rc, $i, "vendorname");
     $i++;
}
```

China-pub.com
下载


```
        pg_FreeResult($result);
        pg_Close($conn);
?>

</SELECT></TD>
</TR>

<TR>
<TD>Quantity:</TD>
<TD><INPUT NAME="quantity"></TD>
</TR>


</TABLE>

<INPUT TYPE="submit">

</FORM>

</BODY>
</HTML>
```

注意上述SELECT语句中的ORDER BY子句，读者可以断定，当“ vendors ”表变得很大时， vendors.vendorname上的索引将很有用。上面的PHP代码以字母顺序检索出厂商名称，并将其以OPTION前缀的方式显示到浏览器上。

SQL使能的文档可以通过URL http://127.0.0.1/dbform.php3浏览到（PHP CGI实现的URL可通过http://127.0.0.1/cgi-bin/php/dbform.php3浏览到）。

假设运行正常的话，一个如图7-2所示的HTML表单将展示出来。

清单7-7展示了Sybase版本的dbform.php3的样子。

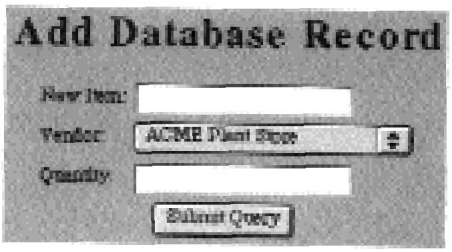


图7-2 dbform.php3的输出

清单7-7 在Sybase中准备表单

```
<HTML>
<HEAD>
<TITLE>Insert Database Record</TITLE>
</HEAD>
<BODY>

<DIV ALIGN="center">

<H1>Add Database Record</H1>

<FORM METHOD="post" ACTION="dbinsert.php3">

<TABLE BORDER>

<TR>
<TD>New Item:</TD>
<TD><INPUT NAME="item">
</TR>
```

China-pub.com 下载

```
<TR>
<TD>Vendor:</TD>
<TD><SELECT NAME="vendor" SIZE="1">

<?PHP
   error_reporting(1);

   $conn = sybase_connect("SYBASE", "nobody", "nobodypass");

   sybase_select_db("shopping",$conn);
   if (!$conn) {
      echo "An error occurred.\n";
      exit;
   }

   $rc = sybase_query(
      "SELECT vendorcode, vendorname
         FROM vendors
         ORDER BY vendorname", $conn);

   if (!$rc) {
      echo "An error occurred.\n";
      exit;
   }

   while ($result = sybase_fetch_array($rc)) {
      echo "<OPTION VALUE=\"$result[vendorcode]\">$result[vendorname]";
   }

   sybase_free_result($rc);
   sybase_close($conn);
?>

</SELECT></TD>
</TR>

<TR>
<TD>Quantity:</TD>
<TD><INPUT NAME="quantity"></TD>
</TR>


</TABLE>

<INPUT TYPE="submit">

</FORM>

</BODY>
</HTML>
```

现在，“前端”已经完成了，必须再实现“后端”。将清单7-8所示的HTML拷贝到/home/httpd/html/dbinsent.php3文件中。

在填入“list”表中item、vendorcode和quantity字段的完整信息之后，数据就可以通过

China-pub.com
下载

pg_Exec ( )调用插入到数据库中。

清单7-8 PostgreSQL的数据库插入

```
<HTML>
<HEAD>
<TITLE>Confirm Database Insert</TITLE>
</HEAD>
<BODY>

<DIV ALIGN="center">

<?PHP

        $conn = pg_Connect("", "", "", "", "shopping");

        if (!$conn) {
                echo "An error occurred.\n";
                exit;
        }

        pg_Exec($conn,
             "INSERT INTO list (item, vendorcode, quantity)
                  VALUES ('$item', $vendor, $quantity);");

        pg_Close($conn);
?>

Database Updated

</BODY>
</HTML>
```

注意，在dbform.php3中，HTML SELECT标志中的vendor变量名在dbinsert.php3的PHP中是可用的。同样，item和quantity表单元素用在了pg_Exec ( )中。PHP将变量信息带到表单动作中，而无需开发者的干涉。

注意SQL INSERT语句的格式，它列出的字段名称分别与 VALUES子句相对应。这类SQL INSERT语句在开发中经常使用，它使得当表格结构发生改变时，无需进行重新编码。在INSERT中没有引用的任何字段，如果表格式允许的话，其值为 NULL。

上面的PHP执行SQL插入事务，并向浏览器发送一个确认消息。如果这样的 .php3文件要用来展示一个数据表项界面，则将 dbform.php3或include ( )拷贝到dbinsert.php3中，这样表单可以持续地显示。

dbinsert.php的Sybase版本展示在清单7-9中。

清单7-9 在Sybase中进行数据插入

```
<HTML>
<HEAD>
<TITLE>Confirm Database Insert</TITLE>
</HEAD>
<BODY>

<DIV ALIGN="center">
```

China-pub.com 下载

```
<?PHP
        error_reporting(1);

  $conn = sybase_connect("SYBASE", "nobody", "nobodypass");

        sybase_select_db("shopping",$conn);

        if (!$conn) {
                echo "An error occurred.\n";
                exit;
        }

      sybase_query(
         "INSERT INTO list (item, vendorcode, quantity)
              VALUES ('$item', $vendor, $quantity)", $conn);

         sybase_close($conn);
?>

Database Updated

</BODY>
</HTML>
```

PHP的另一个示例是认证机制，一个类似于 .htaccess的文件将用来演示如何在 Sybase数据库中进行认证。

将清单7-10中的代码拷贝到/home/httpd/html/auth.php3文件中。

清单7-10　使用Sybase数据库进行认证

```
<?php

   error_reporting(1);

   if(!isset($PHP_AUTH_USER))
   {
      Header("WWW-Authenticate: Basic realm=\"PHP Auth\"");
          Header("HTTP/1.0 401 Unauthorized");
          echo "You must log in to access these pages.\n";
          exit;
   }
   else
   {
    $link=sybase_connect("SYBASE", "nobody", "nobodypass");

          sybase_select_db("security",$link);

          $rc = sybase_query("SELECT password
                      FROM auth
                      WHERE name='$PHP_AUTH_USER'", $link);

          /* should be 1 row */
          if(sybase_num_rows($rc) != 1)
          {
      Header("WWW-Authenticate: Basic realm=\"PHP Auth\"");
```

China-pub.com
下载


```
        Header("HTTP/1.0 401 Unauthorized");
        echo "Attempt to log in as $PHP_AUTH_USER failed.\n";
        exit;
            }

            $row = sybase_fetch_array($rc);

            if($PHP_AUTH_PW != $row[password])
            {
        Header("WWW-Authenticate: Basic realm=\"PHP Auth\"");
        Header("HTTP/1.0 401 Unauthorized");
        echo "Attempt to log in as $PHP_AUTH_USER failed.\n";
                exit;
            }

            sybase_close($link);

    // $PHP_AUTH_USER and $PHP_AUTH_PW are global and authenticated.
        }
    ?>
```

上述的示例依赖于名为 security的数据库，该数据库中包含一个名为“ auth ”的表（尽管可以用另一个数据库和 /或表名代替，如果它们拥有相同的格式的话）。“ auth ”表中必须包含明文的name和password表项。

另一个PHP文档可以通过include ( )指令来使用认证子列程。作为一个示例，清单 7-11展示了这样的PHP文档。

清单7-11 调用认证子列程

```
<?php

include("auth.php3");

?>

<HTML>
<HEAD>
<TITLE>Authenticated Document</TITLE>
</HEAD>
<BODY>

<?php

echo "some sensitive data";

?>

</BODY>
</HTML>
```

试图装载该文档时，系统将出现一个口令提示。

清单7-12中的示例展示了PHP使用数据库领域之外应用程序的情况。假设 HTML有下面的格式：

China-pub.com 下载

清单7-12 使用HTML进行文件上载

```
<html>
<body>
<form enctype="multipart/form-data" action=".upload.php3"
method=post>
<input type="hidden" name="MAX_FILE_SIZE" value="10000000">
Send this file:<input name="userfile" type="file">
<input type="submit" value="Send File">
</form>
</body>
</html>
```

清单7-13中的PHP把一个文件从Web浏览器传过来，并存到服务器上。

清单7-13 使用PHP进行文件上载

```
<html>
<body>

<?php

if (is_file("/home/httpd/html/upload/$userfile_name"))
              echo "The file $userfile_name already exists;
              pick another name.";
else
{
     $f1 = fopen("$userfile","r");

     if(!$f1)
     {
          echo "File not received properly.";
          exit();
     }

     $f2 =

       fopen("/home/httpd/html/upload/$userfile_name","w");

     if(!$f2)
     {
          echo "File could not be created.";
          exit();
     }

     while($buffer = fread($f1, 1024))
          fwrite($f2, $buffer);

     fclose($f1);
     fclose($f2);

   echo "The file $userfile_name was successfully stored.";
}

?>
```

China-pub.com
下载


```
</body>
</html>
```

上面的脚本将文件置于/home/httpd/html/upload目录中。该目录必须对nobody开放写权限（达到这个目的一种方法是将该目录的所属组改成 nobody，并为该组分配写权限）。文件上载的最大长度不由上述的HTML控制，而由/usr/local/lib/php3.ini中的upload_max_filesize PHP配置指令控制。缺省的文件大小为2MB。

7.5 Apache mod_perl

mod_perl是一个特殊版本的perl解释程序，它包含在Apache Web服务器中。mod_perl使得在Web环境中执行Perl脚本会带来惊人的速度改善。

此处RPM版本的mod_perl不能与本章前面讲述的Apache-PHP配置文档一同工作；它必须以stock Apache RPM发布方式安装。同时运行PHP和mod_perl是不明智的，这使得Web服务器的二进制映像会变得十分巨大。如果两个软件包的功能都需要的话，最好是配置两个分离的Web服务器，它们运行在不同的端口上，每一个服务器分别安装适当的模块。

Red Hat Linux中包含了一个mod_perl RPM映像，但它缺省时不安装。本书所附光盘中包含了它的文件，文件名为RedHat/RPMS/mod_perl-1.19-2.i386。输入清单7-14所示的命令以进行安装。

清单7-14 为Apache安装mod_perl

```
/etc/rc.d/init.d/httpd stop

rpm -Uvh mod_perl-1.19-2.i386.rpm

echo 'LoadModule perl_module        modules/libperl.so
AddModule mod_perl.c

<Files ~ "\.pl$">
SetHandler perl-script
PerlHandler Apache::Registry
Options ExecCGI
</Files>

PerlSendHeader On' >> /etc/httpd/conf/httpd.conf

/etc/rc.d/init.d/httpd start
```

此时，文档根目录中的perl脚本将由于mod_perl而获益。要测试mod_perl，创建清单7-15所示的perl脚本。

清单7-15 测试mod_perl

```
echo '#!/usr/bin/perl

print "Content-type: text/html\n\n";

print $ENV{"GATEWAY_INTERFACE"};
print "<br>";
print $ENV{"MOD_PERL"};' > /home/httpd/html/ptest.pl
```

China-pub.com 下载

在Web浏览器上通过URL http://127.0.0.1/ptest.pl即可装载该脚本。其输出为：

```
CGI-Perl/1.1
mod_perl/1.19
```

如果perl源代码不正确，则mod_perl比PHP更具有破坏性。阅读由下面的命令产生的文档，读者可以学到mod_perl在编程结构方面的限制：

```
perldoc cgi_to_mod_perl
perldoc mod_perl_traps
```

mod_perl不适合于初级编程人员。这类编程人员从纯 PHP开始学习更好。

7.6 Pg Perl

PostgreSQL与perl的接口包含在Red Hat Linux 6.0的PostgreSQL软件包中（实际上是在postgresqlclients-6.4.2-3.i386.rpm中）。

perl接口相对直接了当，并且在在线手册页中有完整的文档（使用 man Pg能够访问到相应信息）。

清单7-16是一个使用Perl访问购物数据库内容的脚本。

清单7-16 Pg Perl的一个示例

```
#!/usr/bin/perl

use Pg;

$conn = Pg::connectdb("dbname = shopping");

$result = $conn-exec("SELECT list.item, vendors.vendorname,
list.quantity
                  FROM list, vendors
                  WHERE list.vendorcode = vendors.vendorcode
                  ORDER BY list.item;");

$ntuples = $result->ntuples;

for($i=0; $i < $ntuples; $i++)
{
     $item = $result-getvalue($i, 0);
     $vendorname = $result-getvalue($i, 1);
     $quantity = $result-getvalue($i, 2);
     print "$quantity\t$item\t$vendorname\n";
}
```

请注意，运行脚本的用户是试图访问数据库的 PostgreSQL用户。在最近两章的示例中，只有luser和nobody用户有这样的权限，其他人不能运行这些脚本。

7.7 sybperl

用于Sybase数据库连接的最优秀的 perl是sybperl，它由Michael Peppler编写。sybperl-2.10-02.tar可以从http://www.mbay.net/~mpeppler（这是作者的个人主页）处找到。另外，Michael Peppler还编写了Sybase的DBI/DBD Perl接口，但这些接口还处于 alpha版本中，所以在本书中没有包含。

China-pub.com
下载

下面的脚本只建立perl支持的Sybase CT-Lib。对旧版本Sybase DB-Lib的支持，以及基于其上的工具（如BCP例程）并没有建立，因为它们与各种 RPM软件包冲突。对这些冲突的解决办法超过了本书的范围。

清单7-17是一个建立sybperl-2.10_02的脚本。其源代码包可以从本书所附光盘得到，其文件名为sybase/sybperl-2.10_02.tar.gz，也可以从Michael Peppler的Web站点处获取。请注意，如果sybase不是安装在/home/sybase中，服务器名为SYBASE，且系统管理员的口令为 sapass的话，则下面的脚本要进行相应的改变，以反映其实际情况。要建立 sybperl，以根用户方式输入下面的命令（下面版本的编译要花费很长的时间 —— 超过17min，所以不要打断它们）。

清单7-17 编译Sybperl

```
#!/bin/sh
tar xvzf sybperl-2.10_02.tar.gz
cd sybperl-2.10_02
sed 's/^SYBASE=.*$/SYBASE=\/home\/sybase/
s/^DBLIBVS=.*$/DBLIBVS=0/' CONFIG > CONFIG.NEW
mv -f CONFIG.NEW CONFIG
sed 's/PWD=/PWD=sapass/
s/SRV=.*$/SRV=SYBASE/' PWD > PWD.NEW
mv -f PWD.NEW PWD
perl Makefile.PL
make
make test
make install
```

假设perl可以与Sybase的CT-Lib功能一起使用，则清单7-18中的脚本在Shell下运行时将从Sybase中进行一个SQL查询。

清单7-18 调用sybperl的一个示例

```
#!/usr/bin/perl

# Load the Sybase::CTlib module:
use Sybase::CTlib;

# Allocate a new Database 'handle':
$dbh = new Sybase::CTlib 'luser', 'luserpass', 'SYBASE';

# Select the target database:
$dbh->ct_sql("use shopping");

# Send the query to the server:
$dbh->ct_execute("SELECT list.item, vendors.vendorname,
list.quantity
                  FROM list, vendors
                  WHERE list.vendorcode = vendors.vendorcode
                  ORDER BY list.item");

# Retrieve the result sets
while($dbh->ct_results($restype) == CS_SUCCEED) {

   # Skip non-fetchable results:
   next unless $dbh->ct_fetchable($restype);
```

China-pub.com 下载

```
    # Retrieve actual data rows:
    while(($item, $vendorname, $quantity) = $dbh->ct_fetch) {
         print "$quantity\t$item\t$vendorname\n";
     }
}
```

对该脚本进行简单的修改就可以在HTML上下文中运行。