27
CGI (Common Gateway Interface) and Perl 5

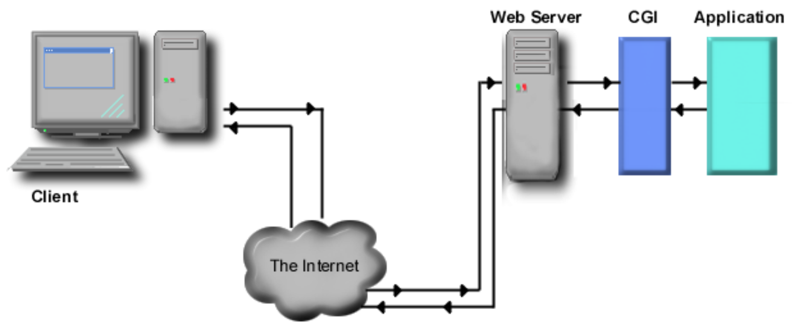
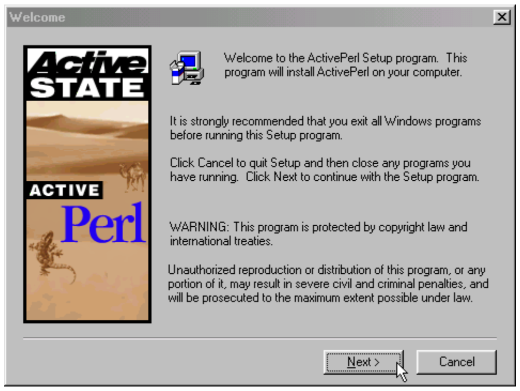


Fig. 27.1 Data path of a typical CGI-based application.



Welcome to the ActivePerl Setup program. This program will install ActivePerl on your computer.

It is strongly recommended that you exit all Windows programs before running this Setup program.

Click Cancel to quit Setup and then close any programs you have running. Click Next to continue with the Setup program.

WARNING: This program is protected by copyright law and international treaties.

Unauthorized reproduction or distribution of this program, or any portion of it, may result in severe civil and criminal penalties, and will be prosecuted to the maximum extent possible under law.

Fig. 27.2 ActivePerl installation **Welcome** dialog.

Command-line switch	Description
`-e 'command'`	Interpret one line of Perl code.
`-S`	Search for the specified script using the **`PATH`** environment variable.
`-U`	Allow unsafe operations to be executed.
-v	Print the version of Perl.
`-w`	Allow warnings to be displayed on compilation of the script.
`-h`	Display all options for **`perl.exe`**.

Fig. 27.3 Some of the common command-line switches used with **`perl.exe`**.

```perl
# Fig. 27.4: first.pl
# A first program in Perl.

print "Welcome to Perl!\n";
```

```
Welcome to Perl!
```

Fig. 27.4 A first program in Perl and its output.

```
# Fig. 27.6: variable.pl
# Program to illustrate the use of scalar variables.

# using a variable in the context of a string
print "Using a variable before initializing: $var\n";

# using a variable in a numeric context
$test = $num + 5;
print "Adding uninitialized variable num to 5 yields: $test.\n";

$a = 5;
print "The value of variable a is: $a\n";

$a = $a + 5;
print "Variable a after adding 5 is $a.\n";

$b = "A string value";
$a = $a + $b;

print "Adding a string to an integer yields: $a\n";

$number = 7;
$b = $b + $number;

print "Adding an integer to a string yields: $b\n";
```

```
Using a variable before initializing:
Adding uninitialized variable num to 5 yields: 5.
The value of variable a is: 5
Variable a after adding 5 is 10.
Adding a string to an integer yields: 10
Adding an integer to a string yields: 7
```

Fig. 27.5 Using scalar variables .

```
# Fig. 27.7: arrays.pl
# Program to demonstrate arrays in Perl

@array = ("Bill", "Bobby", "Sue", "Michelle");

print "The array contains:\n\n";
print "@array \n\n";
print "Third element: $array[2]\n\n";

@array2 = (A..Z);

print "The range operator is used to store all\n";
print "letters from capital A to Z:\n\n";
print "@array2 \n";
```

```
The array contains:

Bill Bobby Sue Michelle

Third element: Sue

The range operator is used to store all
letters from capital A to Z:

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
```

Fig. 27.6 Using arrays .

Package	Functionality
`libwww-perl`	Increases network programming functionality.
`Win32-API`	Enables Perl programs to make Windows system calls.
`Win32-ODBC`	Enables ODBC connectivity within Perl programs.
`Win32-Registry`	Allows Perl programs to read and write to the *Windows Registry* (i.e., a database containing hardware and software information about your computer system).
`XML-Parser`	Allows Perl programs to parse XML documents (see Chapter 28).

Fig. 27.7 Some packages available from ActiveState's Perl package repository.

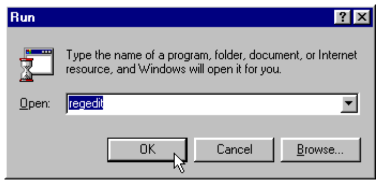

Run
Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.
Open: regedit
OK
Cancel
Browse...

Fig. 27.8 Launching the **Registry Editor** from the **Run** dialog.

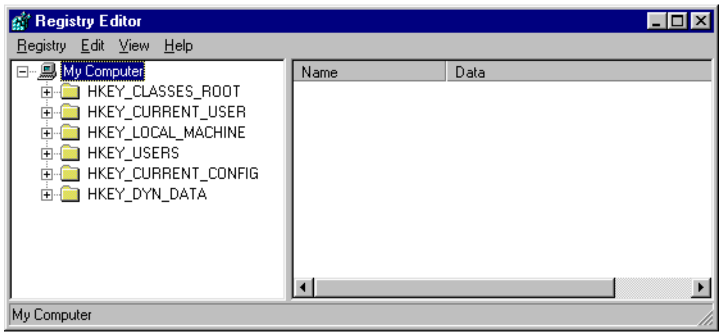


Fig. 27.9 **Registry Editor** application.

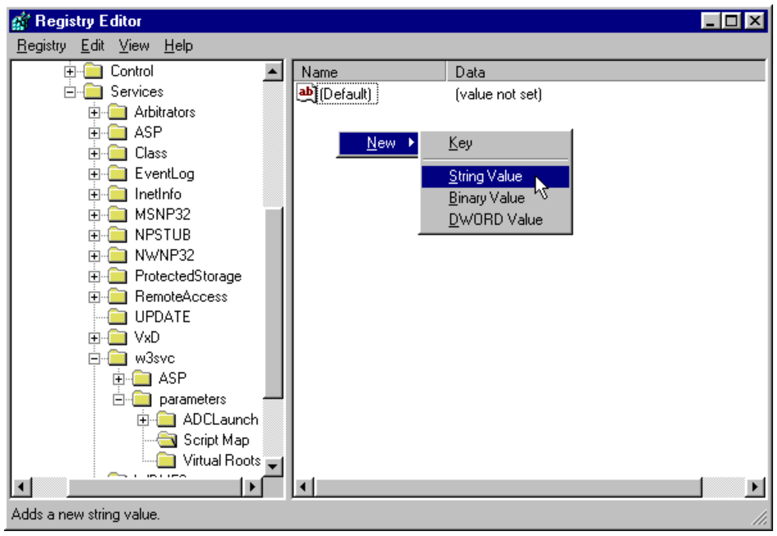
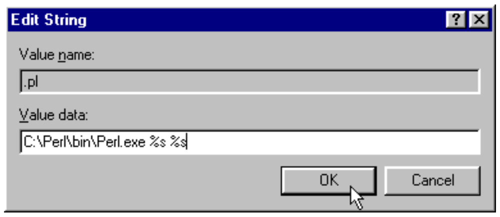


Fig. 27.10 Adding a new association to the **Script Map** key.


Edit String

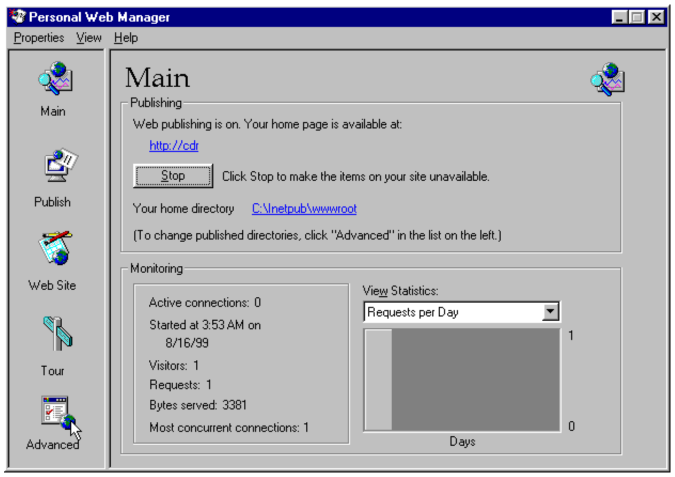
Value name:

.pl

Value data:

C:\Perl\bin\Perl.exe %s %s

Fig. 27.11 Modifying the path information for the `.pl` file extension.



Main

Publishing

Web publishing is on. Your home page is available at:

http://cdr

Click Stop to make the items on your site unavailable.

Your home directory C:\Inetpub\wwwroot

(To change published directories, click "Advanced" in the list on the left.)

Monitoring

Active connections: 0

Started at 3:53 AM on 8/16/99

Visitors: 1

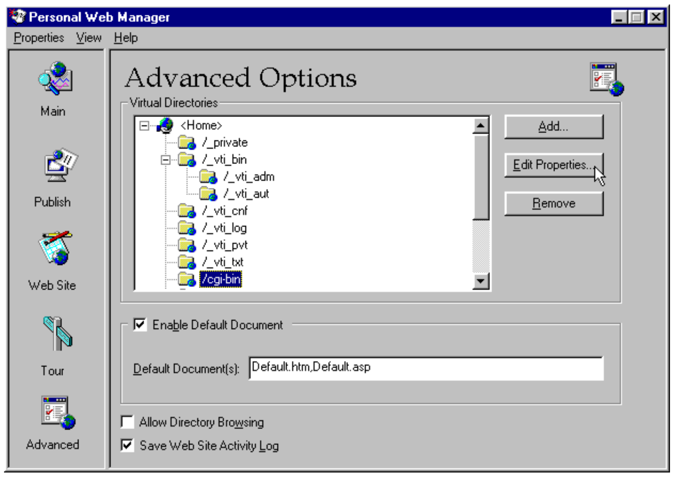
Requests: 1

Bytes served: 3381

Most concurrent connections: 1

View Statistics: Requests per Day

Fig. 27.12 **`Advanced`** icon in PWS.



Advanced Options

Virtual Directories

```
<Home>
  /_private
  /_vti_bin
    /_vti_adm
    /_vti_aut
  /_vti_cnf
  /_vti_log
  /_vti_pvt
  /_vti_txt
  /cgi-bin
```

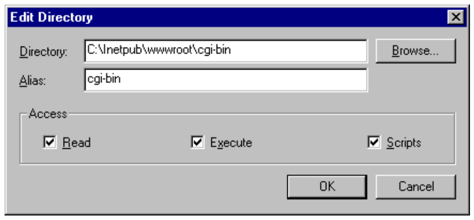
☑ Enable Default Document

Default Document(s): Default.htm,Default.asp

☐ Allow Directory Browsing

☑ Save Web Site Activity Log

Fig. 27.13 Configuring the **`cgi-bin`** directory in PWS.



Directory: C:\Inetpub\wwwroot\cgi-bin
Alias: cgi-bin
Access
☑ Read ☑ Execute ☑ Scripts

Fig. 27.14 Setting permissions for the **`cgi-bin`** directory.

```perl
# Fig. 27.16: equals.pl
# Program to demonstrate the eq operator

my $stringa = "Test";
my $stringb = "Testing";

if ($stringa eq "Test")
{
   print "$stringa matches Test.\n";
}
else
{
   print "$stringa does not match Test.\n";
}

if ($stringb eq "Test")
{
   print "$stringb matches Test.\n";
}
else
{
   print "$stringb does not match Test.\n";
}
```

```
Test matches Test.
Testing does not match Test.
```

Fig. 27.15 Using the **eq** operator .

```
# Fig 27.17: expression1.pl
# searches using the matching operator and regular expressions

$search = "Testing pattern matches";

if ( $search =~ /Test/ )
{
    print "Test was found.\n";
}

if ( $search =~ /^Test/ )
{
    print "Test was found at the beginning of the line.\n";
}

if ( $search =~ /Test$/ )
{
    print "Test was found at the end of the line.\n";
}

if ( $search =~ / \b ( \w+ es ) \b /x )
{
    print "Word ending in es: $1 \n";
}
```

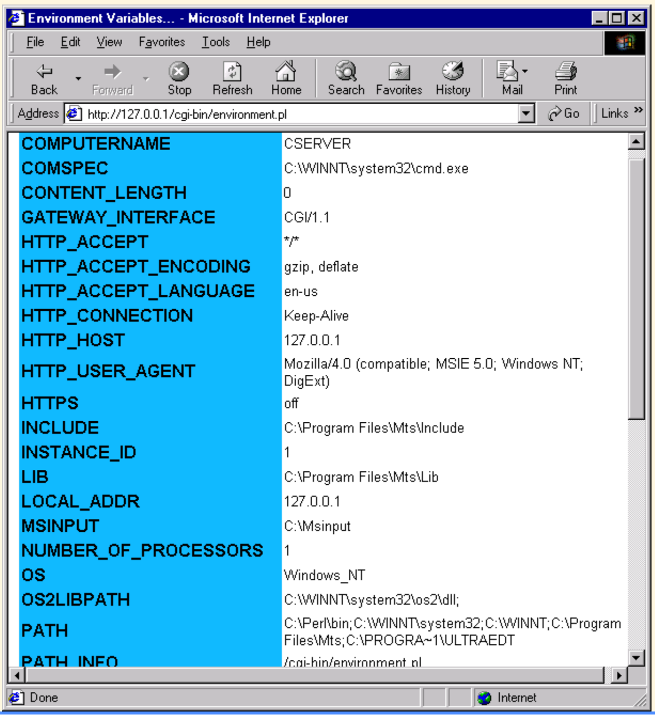
```
Test was found.
Test was found at the beginning of the line.
Word ending in es: matches
```

Fig. 27.16 Using the matching operator **=~** .

Modifying Character	Purpose
`/g`	Search everywhere for the expression (global search).
`/i`	Ignores the case of the search string.
`/m`	The string is evaluated as if it had multiple lines (i.e., contains multiple newline characters) of text.
`/s`	Ignore the newline character and treat it as whitespace. The text is seen as a single line.
`/x`	All whitespace characters are ignored when searching the string.

Fig. 27.17 Some of Perl's modifying characters.

```
# Fig. 27.19: environment.pl
# Program to display CGI environment variables
use CGI qw/:standard/;

print header;
print "<HTML>";
print "   <HEAD>";
print "      <TITLE>Environment Variables...</TITLE>";
print "   </HEAD>";
print "   <BODY TEXT = BLACK BGCOLOR = WHITE>";
print "   <BASEFONT FACE = \"ARIAL,SANS-SERIF\" SIZE = 2>";
print "   <TABLE BORDER = 0 CELLPADDING = 2 CELLSPACING = 0";
print "    WIDTH = 100%>";

foreach $key (sort keys %ENV)
{
   print "<TR>";
   print "<TD BGCOLOR = #11BBFF><STRONG>$key</STRONG></TD>";
   print "<TD><FONT COLOR = BLACK SIZE = 2>$ENV{$key}";
   print "</Font></TD>";
   print "</TR>";
}

print "      </TABLE>";
print "   </BODY>";
print "</HTML>";
```


Environment Variables... - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print
Address http://127.0.0.1/cgi-bin/environment.pl
Go Links
Done
Internet

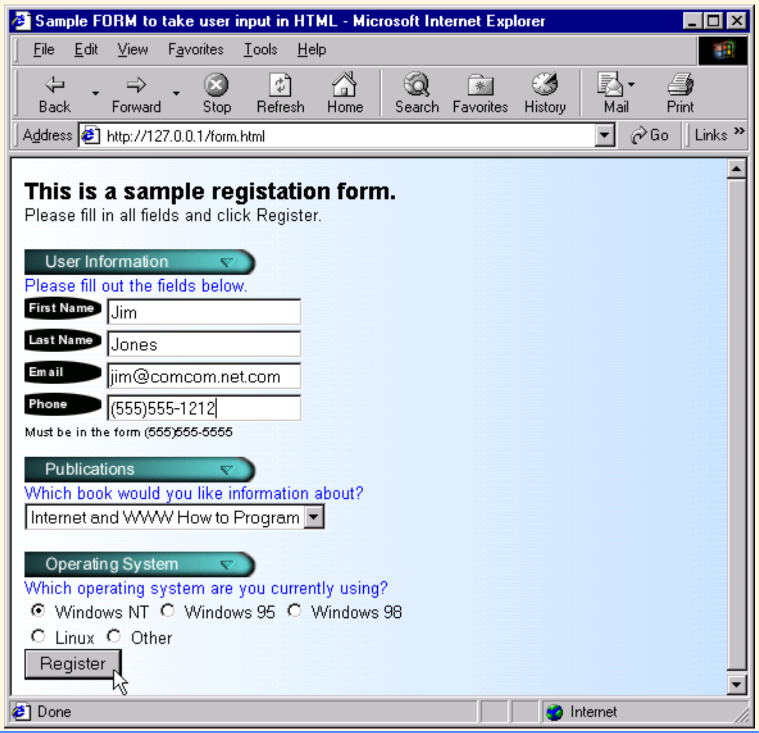
COMPUTERNAME	CSERVER
COMSPEC	C:\WINNT\system32\cmd.exe
CONTENT_LENGTH	0
GATEWAY_INTERFACE	CGI/1.1
HTTP_ACCEPT	*/*
HTTP_ACCEPT_ENCODING	gzip, deflate
HTTP_ACCEPT_LANGUAGE	en-us
HTTP_CONNECTION	Keep-Alive
HTTP_HOST	127.0.0.1
HTTP_USER_AGENT	Mozilla/4.0 (compatible; MSIE 5.0; Windows NT; DigExt)
HTTPS	off
INCLUDE	C:\Program Files\Mts\Include
INSTANCE_ID	1
LIB	C:\Program Files\Mts\Lib
LOCAL_ADDR	127.0.0.1
MSINPUT	C:\Msinput
NUMBER_OF_PROCESSORS	1
OS	Windows_NT
OS2LIBPATH	C:\WINNT\system32\os2\dll;
PATH	C:\Perl\bin;C:\WINNT\system32;C:\WINNT;C:\Program Files\Mts;C:\PROGRA~1\ULTRAEDT
PATH_INFO	/cgi-bin/environment.pl

Fig. 27.18 Displaying CGI environment variables.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<!-- Fig. 27.20: form.html -->

<HTML>
<HEAD>
<TITLE>Sample FORM to take user input in HTML</TITLE>
</HEAD>

<BODY BACKGROUND = "images/back.gif">
<BASEFONT FACE = "ARIAL,SANS-SERIF" SIZE = 2>

   <FONT SIZE = +2>
      <STRONG>This is a sample registation form.</STRONG>
   </FONT><BR>
   Please fill in all fields and click Register.

   <FORM METHOD = "POST" ACTION = "/cgi-bin/form.pl">
      <IMG SRC = "images/user.gif"><BR>
      <FONT COLOR = BLUE>
         Please fill out the fields below.<BR>
      </FONT>

      <IMG SRC = "images/fname.gif">
      <INPUT TYPE = "TEXT" NAME = "FNAME"><BR>
      <IMG SRC = "images/lname.gif">
      <INPUT TYPE = "TEXT" NAME = "LNAME"><BR>
      <IMG SRC = "images/email.gif">
      <INPUT TYPE = "TEXT" NAME = "EMAIL"><BR>
      <IMG SRC = "images/phone.gif">
      <INPUT TYPE = "TEXT" NAME = "PHONE"><BR>

      <FONT SIZE=-2>
         Must be in the form (555)555-5555<BR><BR>
      </FONT>

      <IMG SRC = "images/downloads.gif"><BR>
      <FONT COLOR = BLUE>
         Which book would you like information about?<BR>
      </FONT>

      <SELECT NAME = "BOOK">
         <OPTION>Internet and WWW How to Program
         <OPTION>C++ How to Program 2e
         <OPTION>Java How to Program 3e
         <OPTION>Visual Basic How to Program 1e
      </SELECT>
      <BR><BR>

      <IMG SRC = "images/os.gif"><BR>
      <FONT COLOR = BLUE>
         Which operating system are you
         currently using?<BR>
```

Fig. 27.19 User entering a valid phone number (part 1 of 2).

```
         </FONT>

         <INPUT TYPE = "RADIO" NAME = "OS" VALUE = "Windows NT"
         CHECKED>
         Windows NT
         <INPUT TYPE = "RADIO" NAME = "OS" VALUE = "Windows 95">
         Windows 95
         <INPUT TYPE = "RADIO" NAME = "OS" VALUE = "Windows 98">
         Windows 98<BR>
         <INPUT TYPE = "RADIO" NAME = "OS" VALUE = "Linux">
         Linux
         <INPUT TYPE = "RADIO" NAME = "OS" VALUE = "Other">
         Other<BR>
         <INPUT TYPE = "SUBMIT" VALUE = "Register">
      </FORM>
   </BODY>
</HTML>
```


Sample FORM to take user input in HTML - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print
Address http://127.0.0.1/form.html Go Links

This is a sample registation form.
Please fill in all fields and click Register.

User Information
Please fill out the fields below.

First Name: Jim
Last Name: Jones
Email: jim@comcom.net.com
Phone: (555)555-1212
Must be in the form (555)555-5555

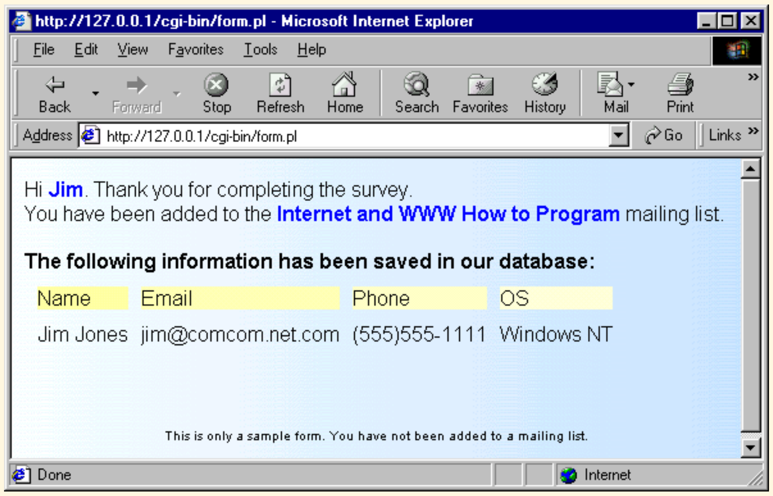
Publications
Which book would you like information about?
Internet and WWW How to Program

Operating System
Which operating system are you currently using?
(●) Windows NT ( ) Windows 95 ( ) Windows 98
( ) Linux ( ) Other

Fig. 27.19 User entering a valid phone number (part 2 of 2).

```
# Fig. 27.21: form.pl
# Program to read information sent to the server
# from the FORM in the form.html document.

use CGI qw/:standard/;

$os = param(OS);
$firstname  = param(FNAME);
$lastname  = param(LNAME);
$email  = param(EMAIL);
$phone  = param(PHONE);
$book  = param(BOOK);

print header;
print "<BODY BACKGROUND = \"/images/back.gif\">";
print "<BASEFONT FACE = \"ARIAL,SANS-SERIF\" SIZE = 3>";

if ( $phone =~ / \( \d{3} \) \d{3} - \d{4} /x )
{
   print "Hi <FONT COLOR = BLUE><B>$firstname</B></FONT>";
   print ". Thank you for completing the survey.<BR>";
   print "You have been added to the ";
   print "<FONT COLOR = BLUE><STRONG>$book </STRONG></FONT>";
   print "mailing list.<BR><BR>";
   print "<STRONG>The following information has been saved ";
   print "in our database:</STRONG><BR>";
   print "<TABLE BORDER = 0 CELLPADDING = 0";
   print "       CELLSPACING = 10>";
   print "<TR><TD BGCOLOR = #FFFFAA>Name </TD>";
   print "    <TD BGCOLOR = #FFFFBB>Email</TD>";
   print "    <TD BGCOLOR = #FFFFCC>Phone</TD>";
   print "    <TD BGCOLOR = #FFFFDD>OS</TD></TR>";
   print "<TR><TD>$firstname $lastname</TD><TD>$email</TD>";
   print "<TD>$phone</TD><TD>$os</TD></TR>";
   print "</TABLE>";
   print "<BR><BR><BR>";
   print "<CENTER><FONT SIZE = -3>";
   print "This is only a sample form. ";
   print "You have not been added to a mailing list.";
   print "</FONT></CENTER>";
}
else
{
   print "<FONT COLOR = RED SIZE = +2>";
   print "INVALID PHONE NUMBER</FONT><BR>";
   print " A valid phone number must be in the form";
   print "<STRONG>(555)555-5555</STRONG>";
   print "<FONT COLOR = BLUE> Click the Back button, ";
   print "enter a valid phone number and resubmit.<BR><BR>";
   print "Thank You.";
}
```

Fig. 27.20 Script to process user data from `form.html` (part 1 of 2).

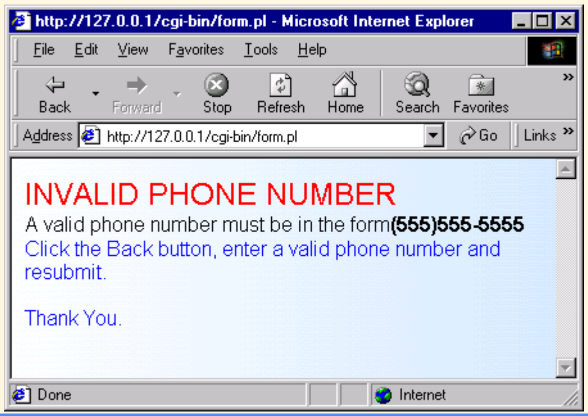

http://127.0.0.1/cgi-bin/form.pl - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print
Address http://127.0.0.1/cgi-bin/form.pl
Go Links
Done
Internet

Hi **Jim**. Thank you for completing the survey.
You have been added to the **Internet and WWW How to Program** mailing list.

The following information has been saved in our database:

Name	Email	Phone	OS
Jim Jones	jim@comcom.net.com	(555)555-1111	Windows NT

This is only a sample form. You have not been added to a mailing list.


http://127.0.0.1/cgi-bin/form.pl - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites
Address http://127.0.0.1/cgi-bin/form.pl
Go Links
Done
Internet

INVALID PHONE NUMBER
A valid phone number must be in the form**(555)555-5555**
Click the Back button, enter a valid phone number and resubmit.

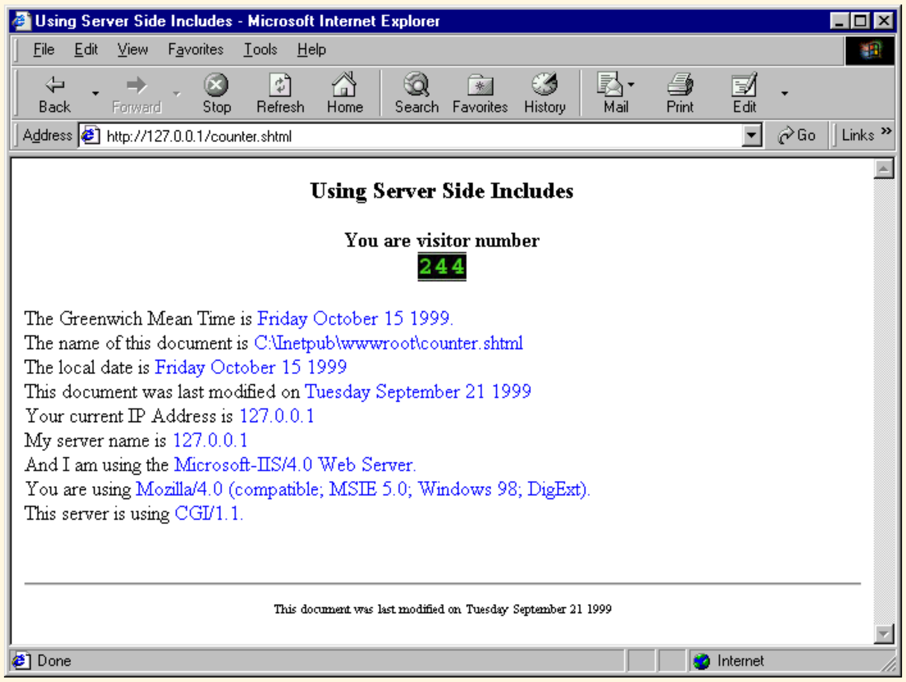
Thank You.

Fig. 27.20 Script to process user data from **`form.html`** (part 2 of 2).

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<!-- Fig. 27.22 counter.shtml -->

<HTML>
   <HEAD>
      <TITLE>Using Server Side Includes</TITLE>
   </HEAD>

<BODY>
   <CENTER>
      <H3> Using Server Side Includes</H3>
   </CENTER>

   <!-- #EXEC CGI="/cgi-bin/counter.pl" --><BR>
   The Greenwich Mean Date is
   <FONT COLOR = BLUE>

   <!-- #ECHO VAR="DATE_GMT" -->.
   </FONT><BR>
   The name of this document is
   <FONT COLOR = BLUE>

   <!-- #ECHO VAR="DOCUMENT_NAME" -->
   </FONT><BR>
   The local date is
   <FONT COLOR = BLUE>

   <!-- #ECHO VAR="DATE_LOCAL" -->
   </FONT><BR>
   This document was last modified on
   <FONT COLOR = BLUE>

   <!-- #ECHO VAR="LAST_MODIFIED" -->
   </FONT><BR>
   Your current IP Address is
   <FONT COLOR = BLUE>

   <!-- #ECHO VAR="REMOTE_ADDR" -->
   </FONT><BR>
   My server name is
   <FONT COLOR = BLUE>

   <!-- #ECHO VAR="SERVER_NAME" -->
   </FONT><BR>
   And I am using the
   <FONT COLOR = BLUE>

   <!-- #ECHO VAR="SERVER_SOFTWARE" -->
   Web Server.</FONT><BR>
   You are using
   <FONT COLOR = BLUE>

```

Fig. 27.21 Incorporating a Web-page hit counter and displaying environment variables (part 1 of 2).

```
      <!-- #ECHO VAR="HTTP_USER_AGENT" -->.
      </FONT><BR>
      This server is using <FONT COLOR = BLUE>

      <!-- #ECHO VAR="GATEWAY_INTERFACE" -->.
      </FONT><BR>
      <BR><BR>
      <CENTER>
      <HR>
      <FONT SIZE = -5>This document was last modified on

      <!-- #ECHO VAR="LAST_MODIFIED" --></FONT>

      </CENTER>
   </BODY>
</HTML>
```


Using Server Side Includes - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print Edit
Address http://127.0.0.1/counter.shtml
Go Links
Done
Internet

Using Server Side Includes

You are visitor number
244

The Greenwich Mean Time is Friday October 15 1999.
The name of this document is C:\Inetpub\wwwroot\counter.shtml
The local date is Friday October 15 1999
This document was last modified on Tuesday September 21 1999
Your current IP Address is 127.0.0.1
My server name is 127.0.0.1
And I am using the Microsoft-IIS/4.0 Web Server.
You are using Mozilla/4.0 (compatible; MSIE 5.0; Windows 98; DigExt).
This server is using CGI/1.1.

This document was last modified on Tuesday September 21 1999

Fig. 27.21 Incorporating a Web-page hit counter and displaying environment variables (part 2 of 2).

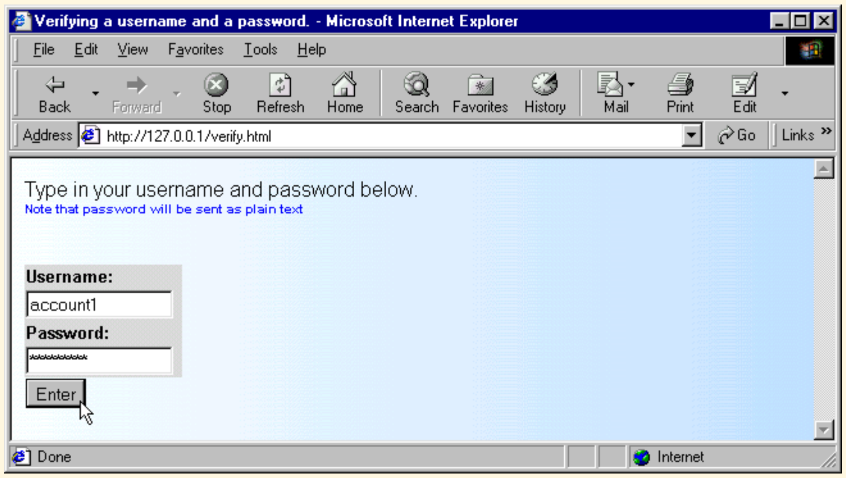
```
# Counter.pl
# Program to track the number of times a web page
# has been accessed.

open(COUNTREAD, "counter.dat");
   my $data = <COUNTREAD>;
   $data++;
close(COUNTREAD);

open(COUNTWRITE, ">counter.dat");
   print COUNTWRITE $data;
close(COUNTWRITE);

print "<CENTER>";
print "<STRONG>You are visitor number</STRONG><BR>";

for ($count = 0; $count < length($data);$count++)
{
   $number = substr( $data, $count, 1 );
   print "<IMG SRC=\"images/counter/$number.jpg\">";
}

print "</CENTER>";
```

Fig. 27.22 Perl script for counting Web page hits.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<!-- Fig. 27.24: verify.html -->

<HTML>
<HEAD>
<TITLE>Verifying a username and a password.</TITLE>
</HEAD>

<BODY BACKGROUND = "images/back.gif">
   <P>
      <FONT FACE = Arial>
      Type in your username and password below.
      </FONT><BR>
      <FONT COLOR = #0000FF FACE = Arial SIZE = 1>
      <STRONG>
      Note that password will be sent as plain text
      </STRONG>
      </FONT>
   </P>

   <FORM ACTION = "/cgi-bin/password.pl" METHOD = "post">
   <BR>

   <TABLE BORDER = "0" CELLSPACING = "0" STYLE = "HEIGHT: 90px;
          WIDTH: 123px" CELLPADING = "0">
      <TR>
         <TD BGCOLOR = #DDDDDD COLSPAN = 3>
         <FONT FACE = Arial SIZE = 2>
         <STRONG>Username:</STRONG>
         </FONT>
         </TD>
      </TR>
      <TR>
         <TD BGCOLOR = #DDDDDD COLSPAN = 3>
         <INPUT SIZE = "40" NAME = "USERNAME"
                STYLE = "HEIGHT: 22px; WIDTH: 115px">
         </TD>
      </TR>
      <TR>
         <TD BGCOLOR = #DDDDDD COLSPAN = 3>
         <FONT FACE = Arial SIZE = 2>
         <STRONG>Password:</STRONG>
         </FONT></TD>
      </TR>
      <TR>
         <TD BGCOLOR = #DDDDDD COLSPAN = 3>
         <INPUT SIZE = "40" NAME = "PASSWORD"
                STYLE = "HEIGHT: 22px; WIDTH: 115px"
                TYPE = PASSWORD>
         <BR></TD>
      </TR>
      <TR>
```

Fig. 27.23 Entering a username and a password (part 1 of 3).

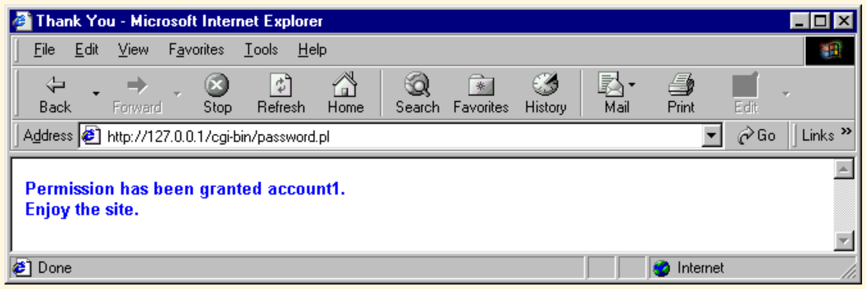
```
               <TD COLSPAN = 3>
               <INPUT TYPE = "submit" VALUE = "Enter"
                      STYLE = "HEIGHT: 23px; WIDTH: 47px">
               </TD>
            </TR>
         </TABLE>
      </FORM>
   </BODY>
</HTML>
```


Verifying a username and a password. - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print Edit
Address http://127.0.0.1/verify.html
Go Links
Done
Internet

Type in your username and password below.
Note that password will be sent as plain text

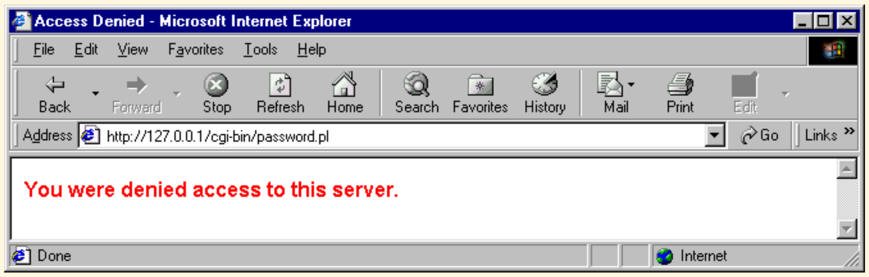
Username:
account1
Password:
****
Enter

Fig. 27.23 Entering a username and a password (part 2 of 3).


Thank You - Microsoft Internet Explorer

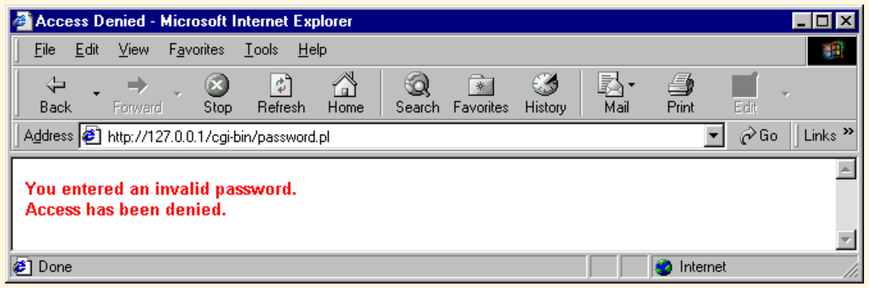
http://127.0.0.1/cgi-bin/password.pl

Permission has been granted account1.
Enjoy the site.


Access Denied - Microsoft Internet Explorer

http://127.0.0.1/cgi-bin/password.pl

You were denied access to this server.


Access Denied - Microsoft Internet Explorer

http://127.0.0.1/cgi-bin/password.pl

You entered an invalid password.
Access has been denied.

Fig. 27.23 Entering a username and a password (part 3 of 3).

```
# Fig. 27.25: password.pl
# Program to search a database for usernames and passwords.
use CGI qw/:standard/;

my $username = param(USERNAME);
my $password = param(PASSWORD);

open(FILE, "data.txt") ||
   die "The database could not be opened";

   while(<FILE>)
   {
      @data = split(/\n/);

      foreach $entry (@data)
      {
         ($name, $pass) = split(/,/, $entry);

         if($name eq "$username")
         {
            $userverified = 1;
            if ($pass eq "$password")
            {
               $passwordverified = 1;
            }
         }
       }
   }

   close(FILE);

   if ($userverified && $passwordverified)
   {
      &accessgranted;
   }
   elsif ($userverified && !$passwordverified)
   {
      &wrongpassword;
   }
   else
   {
      &accessdenied;
   }

sub accessgranted
{
   print header;
   print "<TITLE>Thank You</TITLE>";
   print "<FONT FACE = Arial SIZE = 2 COLOR = BLUE>";
   print "<STRONG>Permission has been granted $username.";
   print "<BR> Enjoy the site.</STRONG></FONT>";
}

```

Fig. 27.24 Contents of `password.pl` Perl script (part 1 of 2).

```
sub wrongpassword
{
   print header;
   print "<TITLE>Access Denied</TITLE>";
   print "<FONT FACE=Arial SIZE=2 COLOR=Red><STRONG>";
   print "You entered an invalid password.<br> ";
   print "Access has been denied.</STRONG></FONT>";
   exit;

}

sub accessdenied
{
   print header;
   print "<TITLE>Access Denied</TITLE>";
   print "<FONT FACE=Arial SIZE=3 COLOR=Red><STRONG>";
   print "You were denied access to this server.";
   print "</STRONG></FONT>";
   exit;
}
```

Fig. 27.24 Contents of `password.pl` Perl script (part 2 of 2).

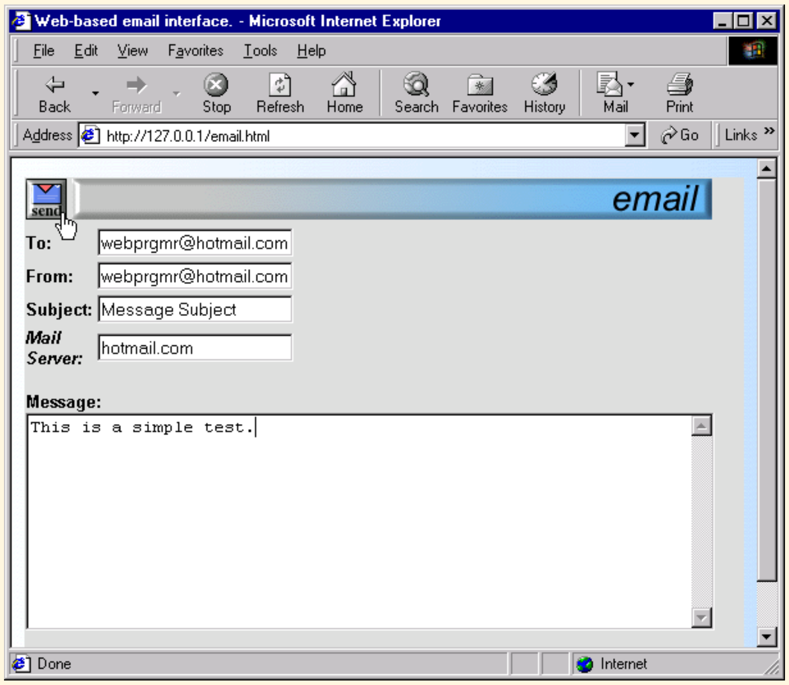
```
account1,password1
account2,password2
account3,password3
account4,password4
account5,password5
account6,password6
account7,password7
account8,password8
account9,password9
account10,password10
```

Fig. 27.25 Database **`data.txt`** containing user names and passwords.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<!-- Fig. 27.27: email.html -->
<HTML>
<HEAD>
   <TITLE>Web-based email interface.</TITLE>
</HEAD>

<BODY BACKGROUND = "images/back.gif">
   <FORM ACTION = "cgi-bin/mail.pl" METHOD = "POST">
   <TABLE BORDER = "0" CELLSPACING = "0" CELLPADING = "0">

      <TR>
         <TD BGCOLOR = #DDDDDD COLSPAN = 3>
         <INPUT SRC = "images/send.gif" TYPE = "IMAGE">
         <IMG SRC = "images/bar.gif">
         </TD>
      </TR>

      <TR>
         <TD BGCOLOR = #DDDDDD WIDTH = "10%"><STRONG>
         <FONT FACE = "Arial" SIZE = "2">To:&nbsp;</FONT>
         </STRONG>
         </TD>
         <TD BGCOLOR = #DDDDDD><INPUT NAME = "TO">
         </TD>
      </TR>

      <TR>
         <TD BGCOLOR = #DDDDDD>
         <P><FONT FACE = Arial SIZE = 2>
         <STRONG>From:</STRONG>
         </FONT></P>
         </TD>
         <TD BGCOLOR = #DDDDDD><INPUT NAME = "FROM">
         </TD>
      </TR>

      <TR>
         <TD BGCOLOR = #DDDDDD>
         <P><FONT FACE = Arial SIZE = 2>
         <STRONG>Subject:</STRONG>
         </FONT></P>
         </TD>
         <TD BGCOLOR = #DDDDDD><INPUT NAME = "SUBJECT">
         </TD>
      </TR>

      <TR>
         <TD BGCOLOR = #DDDDDD>
         <P><FONT FACE = "Arial" SIZE = 2><STRONG><EM>Mail
         Server:</EM></STRONG></FONT></P>
```

Fig. 27.26 HTML to display Web-based email **FORM** (part 1 of 2).

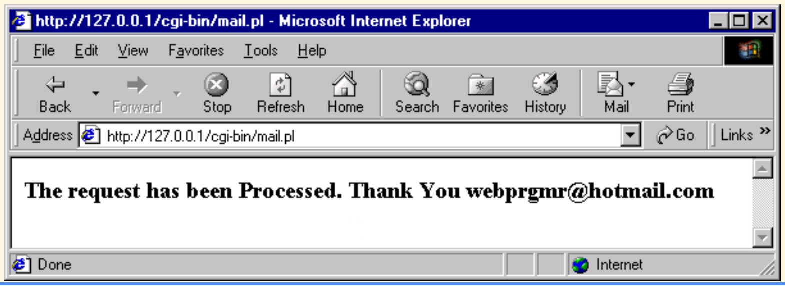
```
            </TD>
            <TD BGCOLOR = #DDDDDD><INPUT NAME = "MAILSERVER">
            </TD>
         </TR>

         <TR>
            <TD BGCOLOR = #DDDDDD COLSPAN = 3>
            <P>
            <STRONG><FONT FACE = "Arial" SIZE = "2"><BR>Message:
            </FONT>
            </STRONG><BR>
            <TEXTAREA COLS = 50 NAME = "MESSAGE" ROWS = 6
            STYLE = "HEIGHT: 170px; WIDTH: 538px"></TEXTAREA>
            </P><P> </P>
            </TD>
         </TR>

      </TABLE>
      </FORM>
   </HTML>
```


Web-based email interface. - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print
Address http://127.0.0.1/email.html
Go Links
send
email
Done
Internet

To: webprgmr@hotmail.com
From: webprgmr@hotmail.com
Subject: Message Subject
Mail Server: hotmail.com
Message:
This is a simple test.

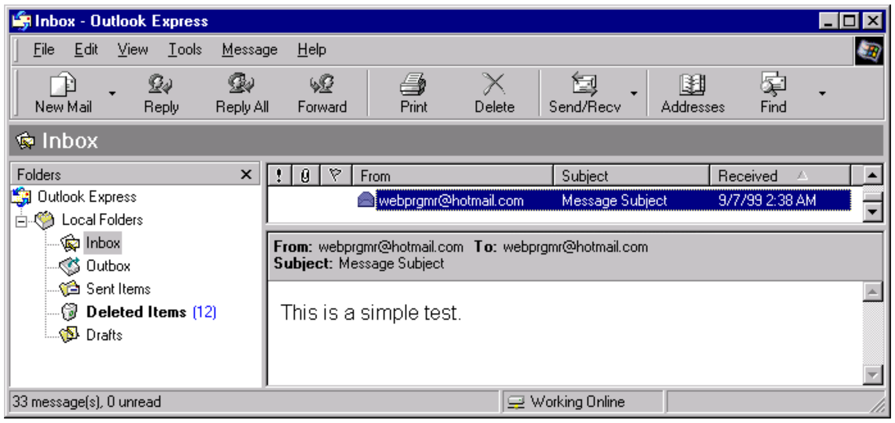
Fig. 27.26 HTML to display Web-based email **FORM** (part 2 of 2).

```
# Fig. 27.28: mail.pl
# Program to send email from a Web-based form.

use Net::SMTP;
use CGI qw/:standard/;

my $to = param("TO");
my $from  = param("FROM");
my $subject  = param("SUBJECT");
my $message  = param("MESSAGE");
my $mailserver = param("MAILSERVER");

print header;
print "<H3>The request has been Processed. ";
print "Thank You $from</H3>";

$smtp = Net::SMTP->new($mailserver);

$smtp->mail($ENV{USER});
$smtp->to("$to");
$smtp->data();
$smtp->datasend("To: $to \n");
$smtp->datasend("From: $from \n");
$smtp->datasend("Subject: $subject \n");
$smtp->datasend("\n");
$smtp->datasend("$message \n");
$smtp->dataend();

$smtp->quit;
```


http://127.0.0.1/cgi-bin/mail.pl - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print
Address http://127.0.0.1/cgi-bin/mail.pl
Go Links
Done
Internet

The request has been Processed. Thank You webprgmr@hotmail.com

Fig. 27.27 Results of **`email.html`** after user clicks **send** in Fig. 27.27 .



webprgmr@hotmail.com Message Subject 9/7/99 2:38 AM

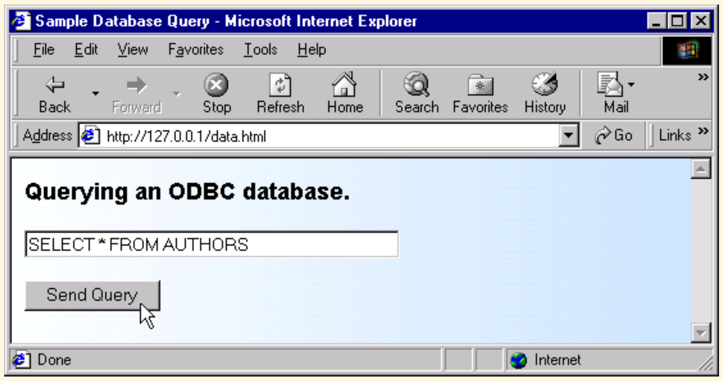
From: webprgmr@hotmail.com **To:** webprgmr@hotmail.com
Subject: Message Subject

This is a simple test.

33 message(s), 0 unread Working Online

Fig. 27.28 Inbox of Microsoft **Outlook Express** showing a new message.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<!-- Fig. 27.30: data.html -->

<HTML>
<HEAD>
<TITLE>Sample Database Query</TITLE>
</HEAD>

<BODY BACKGROUND = "images/back.gif">
<BASEFONT FACE = "ARIAL,SANS-SERIF" SIZE = 2>

   <FONT SIZE = +2>
      <STRONG>Querying an ODBC database.</STRONG>
   </FONT><BR>

   <FORM METHOD = "POST" ACTION = "cgi-bin/data.pl">
      <INPUT TYPE = "TEXT" NAME = "QUERY" SIZE = 40
             VALUE = "SELECT * FROM AUTHORS"><BR><BR>
      <INPUT TYPE = "SUBMIT" VALUE = "Send Query">
   </FORM>
</BODY>
</HTML>
```


Sample Database Query - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail
Address http://127.0.0.1/data.html
Go Links
Send Query
Done
Internet

Querying an ODBC database.

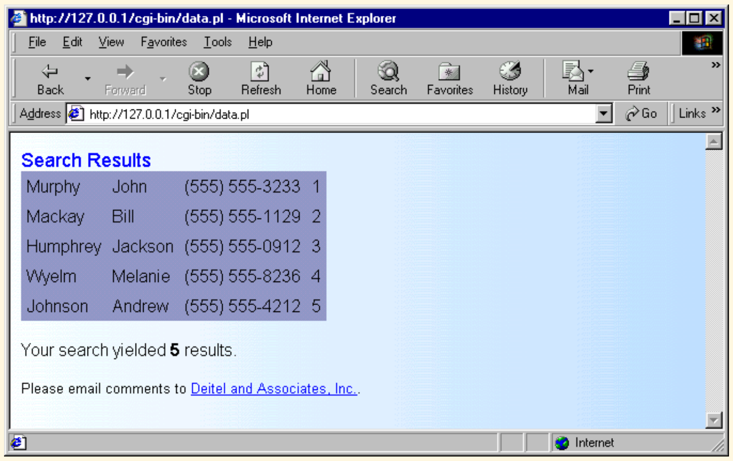
SELECT * FROM AUTHORS

Fig. 27.29 Source code and output of the **`data.html`** document .

```perl
# Fig. 27.31: data.pl
# Program to query a database and send
# results to the client.

use Win32::ODBC;
use CGI qw/:standard/;

my $querystring = param(QUERY);
$DSN = "Products";

print header;

if (!($Data = new Win32::ODBC($DSN)))
{
   print "Error connecting to $DSN\n";
   print "Error: " . Win32::ODBC::Error() . "\n";
   exit;
}

if ($Data->Sql($querystring))
{
   print "SQL failed.\n";
   print "Error: " . $Data->Error() . "\n";
   $Data->Close();
   exit;
}

print "<BODY BACKGROUND = \"/images/back.gif\">";
print "<BASEFONT FACE = \"ARIAL,SANS-SERIF\" SIZE = 3>";
print "<FONT COLOR = BLUE SIZE = 4> Search Results </FONT>";

$counter = 0;

print "<TABLE BORDER = 0 CELLPADDING = 5 CELLSPACING = 0>";

while($Data->FetchRow())
{

   %Data = $Data->DataHash();


   print "<TR>";

   foreach $key( keys( %Data ) )
   {
      print "<TD BGCOLOR = #9999CC>$Data{$key}</TD>";
   }
   print "</TR>";
   $counter++;
}
print "</TABLE>";
print "<BR>Your search yielded <B>$counter</B> results.";
```

Fig. 27.30 Data returned by the database query (part 1 of 2).

```
print "<BR><BR>";
print "<FONT SIZE = 2>";
print "Please email comments to ";
print "<A href = \"mailto:deitel\@deitel.com\">Deitel ";
print "and Associates, Inc.</A>.";
print end_html;

$Data->Close();
```


http://127.0.0.1/cgi-bin/data.pl - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print
Address http://127.0.0.1/cgi-bin/data.pl
Go Links
Internet

Search Results

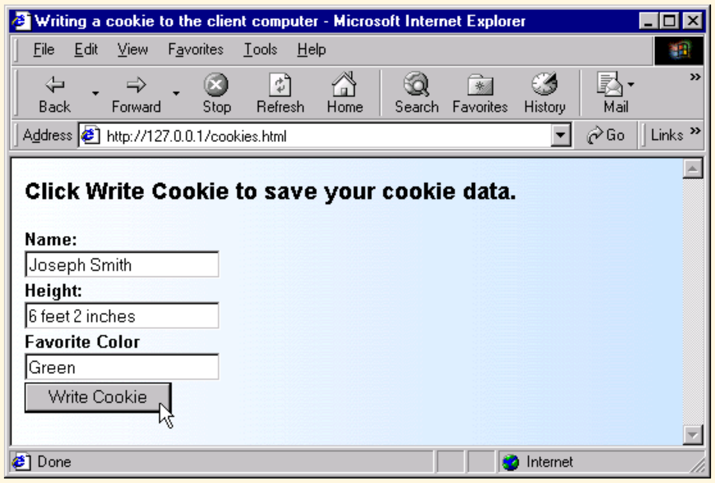
Murphy	John	(555) 555-3233	1
Mackay	Bill	(555) 555-1129	2
Humphrey	Jackson	(555) 555-0912	3
Wyelm	Melanie	(555) 555-8236	4
Johnson	Andrew	(555) 555-4212	5

Your search yielded **5** results.

Please email comments to Deitel and Associates, Inc..

Fig. 27.30 Data returned by the database query (part 2 of 2).

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<!-- Fig. 27.32: cookies.html -->

<HTML>
   <HEAD>
      <TITLE>Writing a cookie to the client computer</TITLE>
   </HEAD>

<BODY BACKGROUND = "images/back.gif">
<BASEFONT FACE = "ARIAL,SANS-SERIF" SIZE = 2>

   <FONT SIZE = +2>
      <B>Click Write Cookie to save your cookie data.</B>
   </FONT><BR>

   <FORM METHOD = "POST" ACTION = "cgi-bin/cookies.pl">
      <STRONG>Name:</STRONG><BR>
      <INPUT TYPE = "TEXT" NAME = "NAME"><BR>
      <STRONG>Height:</STRONG><BR>
      <INPUT TYPE = "TEXT" NAME = "HEIGHT"><BR>
      <STRONG>Favorite Color</STRONG><BR>
      <INPUT TYPE = "TEXT" NAME = "COLOR"><BR>
      <INPUT TYPE = "SUBMIT" VALUE = "Write Cookie">
   </FORM>
</BODY>
</HTML>
```


Writing a cookie to the client computer - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail
Address http://127.0.0.1/cookies.html
Go Links
Done
Internet

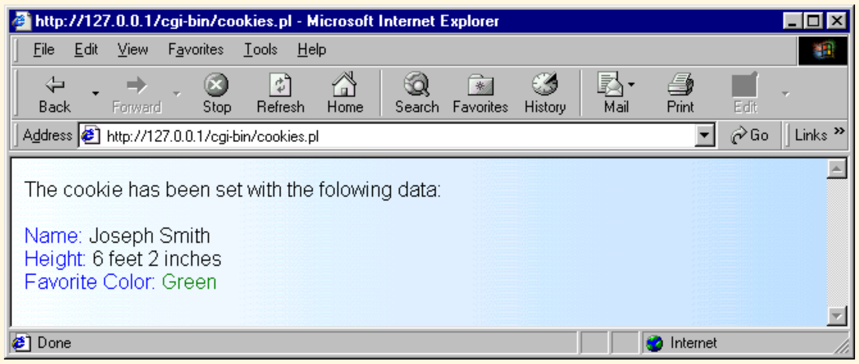
Click Write Cookie to save your cookie data.

Name:
Joseph Smith
Height:
6 feet 2 inches
Favorite Color
Green
Write Cookie

Fig. 27.31 Source for **cookies.html** Web page .

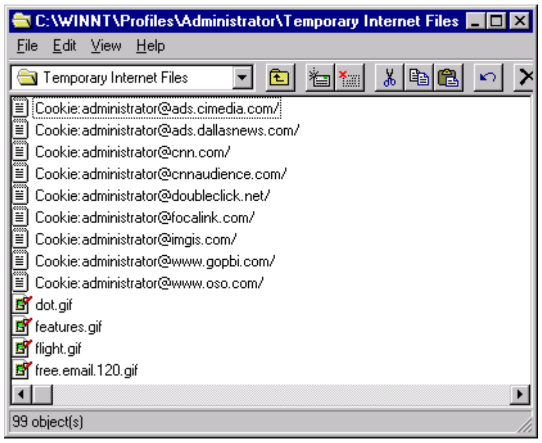
```perl
# Fig. 27.33: cookies.pl
# Program to write a cookie to a client's machine

use CGI qw/:standard/;

my $name = param(NAME);
my $height = param(HEIGHT);
my $color = param(COLOR);

$expires = "Monday, 20-Dec-99 16:00:00 GMT";
$path = "";
$server_domain = "127.0.0.1";

print "Set-Cookie: ";
print "Name", "=", $name, "; expires=", $expires,
      "; path=", $path, "; domain=", $server_domain, "\n";

print "Set-Cookie: ";
print "Height", "=", $height, "; expires=", $expires,
      "; path=", $path, "; domain=", $server_domain, "\n";

print "Set-Cookie: ";
print "Color", "=", $color, "; expires=", $expires,
      "; path=", $path, "; domain=", $server_domain, "\n";

print header;
print "<BODY BACKGROUND = \"/images/back.gif\">";
print "<BASEFONT FACE = \"ARIAL,SANS-SERIF\" SIZE = 3>";
print "The cookie has been set with the folowing data:";
print "<BR><BR>";
print "<FONT COLOR=BLUE>Name:</FONT> $name <BR>";
print "<FONT COLOR = BLUE>Height:</FONT> $height<BR>";
print "<FONT COLOR = BLUE>Favorite Color:</FONT> ";
print "<FONT COLOR = $color> $color<BR>";
```

Fig. 27.32 Writing a cookie to the client (part 1 of 2).


http://127.0.0.1/cgi-bin/cookies.pl - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print Edit
Address http://127.0.0.1/cgi-bin/cookies.pl Go Links
Done
Internet

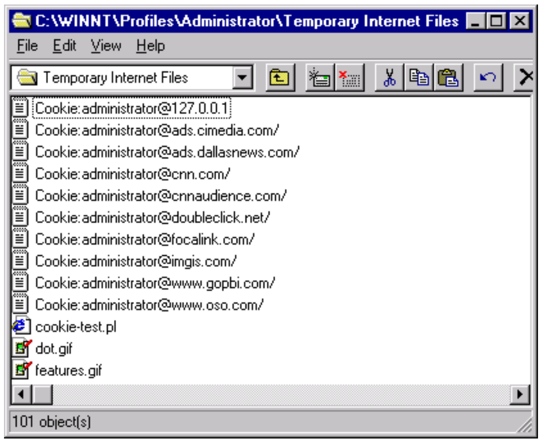
```
The cookie has been set with the folowing data:

Name: Joseph Smith
Height: 6 feet 2 inches
Favorite Color: Green
```

Fig. 27.32 Writing a cookie to the client (part 2 of 2).



```
Cookie:administrator@ads.cimedia.com/
Cookie:administrator@ads.dallasnews.com/
Cookie:administrator@cnn.com/
Cookie:administrator@cnnaudience.com/
Cookie:administrator@doubleclick.net/
Cookie:administrator@focalink.com/
Cookie:administrator@imgis.com/
Cookie:administrator@www.gopbi.com/
Cookie:administrator@www.oso.com/
dot.gif
features.gif
flight.gif
free.email.120.gif
99 object(s)
```

Fig. 27.33 **Temporary Internet Files** directory before a cookie is written.

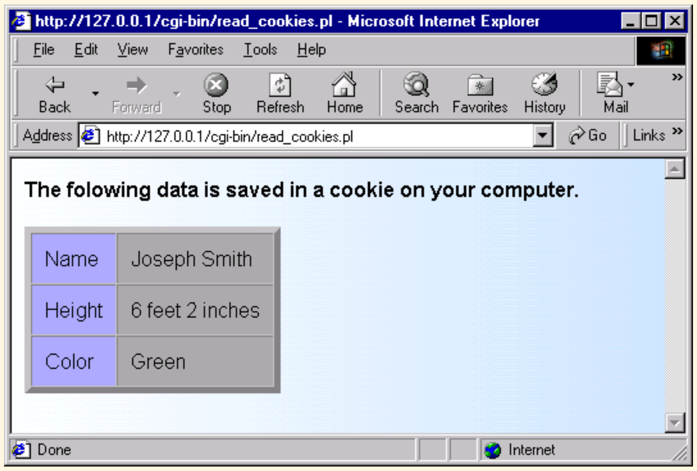


```
Cookie:administrator@127.0.0.1
Cookie:administrator@ads.cimedia.com/
Cookie:administrator@ads.dallasnews.com/
Cookie:administrator@cnn.com/
Cookie:administrator@cnnaudience.com/
Cookie:administrator@doubleclick.net/
Cookie:administrator@focalink.com/
Cookie:administrator@imgis.com/
Cookie:administrator@www.gopbi.com/
Cookie:administrator@www.oso.com/
cookie-test.pl
dot.gif
features.gif
101 object(s)
```

Fig. 27.34 **Temporary Internet Files** directory after a cookie is written.

```
# Fig. 27.36: read_cookies.pl
# Program to read cookies from the client's computer

use CGI qw/:standard/;

print header;
print "<BODY BACKGROUND = \"/images/back.gif\">";
print "<BASEFONT FACE = \"ARIAL,SANS-SERIF\" SIZE = 3>";
print "<STRONG>The following data is saved in a cookie on your ";
print "computer.</STRONG><BR><BR>";

my %cookie = &readCookies;

print ("<TABLE ",
       "BORDER = \"5\" ",
       "CELLSPACING = \"0\" ",
       "CELLPADDING = \"10\">");

foreach $cookie_name (keys %cookie)
{
   print "<TR>";
   print "   <TD BGCOLOR=#AAAAFF>$cookie_name</TD>";
   print "   <TD BGCOLOR=#AAAAAA>$cookie{$cookie_name}</TD>";
   print "</TR>";
}
print "</TABLE>";

sub readCookies
{
   my @cookie_values = split (/; /,$ENV{'HTTP_COOKIE'});

   foreach (@cookie_values)
   {
      my ($cookie_name, $cookie_value) = split ( /=/, $_ );
      $cookies{$cookie_name} = $cookie_value;
   }

   return %cookies;
}
```

Fig. 27.35 Output displaying the cookie's content (part 1 of 2).

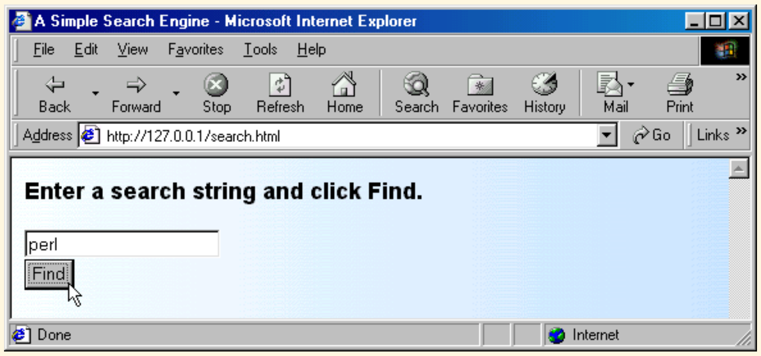

http://127.0.0.1/cgi-bin/read_cookies.pl - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail
Address http://127.0.0.1/cgi-bin/read_cookies.pl
Go Links
Done
Internet

The folowing data is saved in a cookie on your computer.

Name	Joseph Smith
Height	6 feet 2 inches
Color	Green

Fig. 27.35 Output displaying the cookie's content (part 2 of 2).

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<!-- Fig. 27.37: search.html -->

<HTML>
   <HEAD>
      <TITLE>A Simple Search Engine</TITLE>
   </HEAD>

<BODY BACKGROUND = "images/back.gif">
<BASEFONT FACE = "ARIAL,SANS-SERIF" SIZE = 2>

   <FONT SIZE = +2>
      <STRONG>Enter a search string and click Find.</STRONG>
   </FONT><BR>

   <FORM METHOD = "POST" ACTION = "cgi-bin/search.pl">
      <INPUT TYPE = "TEXT" NAME = "SEARCH"><BR>
      <INPUT TYPE = "SUBMIT" VALUE = "Find">
   </FORM>
</BODY>
</HTML>
```


A Simple Search Engine - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print
Address http://127.0.0.1/search.html
Go Links
Done
Internet

Enter a search string and click Find.

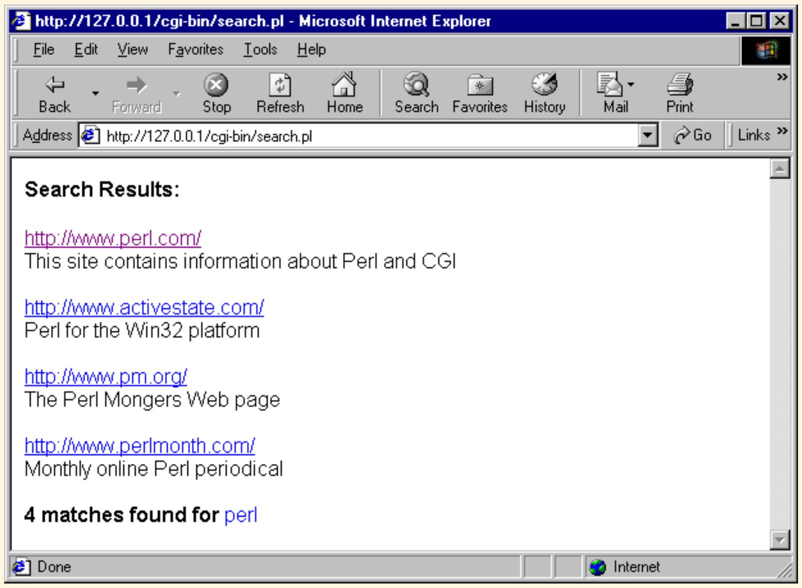
perl

Find

Fig. 27.36 Web site used to enter text to search for with the search engine.

```perl
# Fig. 27.38: search.pl
# Program to search for Web pages

use CGI qw/:standard/;

my $search = param(SEARCH);
my $counter = 0;

print header;
print "<BASEFONT FACE = \"ARIAL,SANS-SERIF\" SIZE = 3>";

open(FILE, "urls.txt") ||
   die "The URL database could not be opened";

while(<FILE>)
{
   my @data = split(/\n/);

   foreach $entry (@data)
   {
      my ($data, $url) = split(/;/, $entry);

      if ($data =~ /$search/i)
      {
         if ($counter == 0)
         {
            print "<STRONG>Search Results:<BR><BR></STRONG>";
         }

         print "<A HREF=\"http://$url/\">";
         print "http://$url/";
         print "</A>";
         print "<BR>$data<BR><BR>";
         $counter++;
      }
   }
}
close FILE;

if ($counter == 0)
{
   print "<B>Sorry, no results were found matching </B>";
   print "<FONT COLOR = BLUE>$search</FONT>. ";
}
else
{
   print "<STRONG>$counter matches found for </STRONG>";
   print "<FONT COLOR = BLUE>$search</FONT>";
}
```

Fig. 27.37 Perl program to implement a simple search engine (part 1 of 2).


http://127.0.0.1/cgi-bin/search.pl - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print
Address http://127.0.0.1/cgi-bin/search.pl Go Links
Done
Internet

Search Results:

http://www.perl.com/
This site contains information about Perl and CGI

http://www.activestate.com/
Perl for the Win32 platform

http://www.pm.org/
The Perl Mongers Web page

http://www.perlmonth.com/
Monthly online Perl periodical

4 matches found for perl

Fig. 27.37 Perl program to implement a simple search engine (part 2 of 2).

```
This site contains information about Perl and CGI;www.perl.com
The Deitel and Deitel Web Site;www.deitel.com
Purchase books on this web site;www.amazon.com
Perl for the Win32 platform;www.activestate.com
The Perl Mongers Web page;www.pm.org
Monthly online Perl periodical;www.perlmonth.com
```

Fig. 27.38 Database (`urls.txt`) containing URLs and brief descriptions of the Web sites.