

# INTRODUCTION TO PERL

The Practical Extraction and Report Language (Perl) was developed by Larry Wall and was first released in 1987. Initially a scripting language designed to manipulate text files, Perl has evolved into a high-level programming language.

One of the major benefits of using Perl is that it is easy to learn. Perl shares many features with the C programming language but is simpler to understand and use. No prior programming experience is required to learn Perl.

<p><b>Scripts</b></p> <p>A program you write in Perl is called a script. Perl scripts are text files that can be created using any text editor, such as Pico. The Perl interpreter compiles and executes the Perl scripts you create. You can manually execute a script by invoking the Perl interpreter at the command prompt or you can define the location of the Perl interpreter at the beginning of a script to create a self-executing script.</p>	<p><b>Versions</b></p> <p>The Perl programming language is constantly evolving and a new version of Perl is released every few years. Each new version of Perl is compatible with previous versions. This means that scripts you write using the current version of Perl will be compatible with future versions. Perl version 5.005 is used in the examples throughout this book.</p>
<p><b>Availability</b></p> <p>Perl is available free of charge and there are no costs or restrictions related to distributing the scripts you create. If you are using a UNIX system, Perl may already be installed. You can obtain the latest version of Perl for UNIX systems at the <a href="http://www.cpan.org">www.cpan.org</a> Web site. If you are using a Windows system, you can obtain the latest Windows-compatible version of Perl, called ActivePerl, at the <a href="http://www.activestate.com">www.activestate.com</a> Web site. MacPerl, the version of Perl for Macintosh systems, is available at the <a href="http://www.macperl.com">www.macperl.com</a> Web site. Perl installation documentation is provided at the appropriate Web site.</p>	<p><b>Online Resources</b></p> <p>There are many online resources available that you can use to find support and information for Perl. The Comprehensive Perl Archive Network (CPAN) Web site at <a href="http://www.cpan.org">www.cpan.org</a> offers documentation, scripts and source code. The Perl Mongers Web site at <a href="http://www.perl.org">www.perl.org</a> provides the latest information about Perl and technical support. There are also several newsgroups you can consult to read and post articles about Perl. The <a href="mailto:comp.lang.perl.misc">comp.lang.perl.misc</a> and <a href="http://comp.infosystems.www.authoring.cgi">comp.infosystems.www.authoring.cgi</a> newsgroups are two examples.</p>

## PERL FEATURES

<p><b>Modules</b></p> <p>A module is a file that stores Perl code, which can be accessed by Perl scripts to perform a task. Using modules extends the functionality of Perl and makes creating scripts easier by allowing you to re-use sections of code. Perl version 5.005 includes many standard modules and there are numerous modules available free of charge on the Internet. Most of the modules available on the Internet can be found at the <a href="http://www.cpan.org">www.cpan.org</a> Web site. For even greater programming flexibility, Perl also allows you to create and use your own custom modules.</p>	<p><b>Web Programming</b></p> <p>Perl is also a very popular Web programming language and can be used to create a script that will generate a Web page or process the information entered into a form on a Web page.</p> <p>The Perl CGI.pm module makes it easy for you to program using the Common Gateway Interface (CGI), which is the standard used to create interactive Web sites. You can use the CGI.pm module to create Web sites containing features such as forms, tables and cookies.</p> <p>Most Web servers support Perl and can run Perl scripts with few configuration changes. Because Perl scripts are processed on the Web server, users cannot access the code used to create Perl generated Web pages. This makes it safer to work with sensitive data, such as login names and passwords.</p>
<p><b>Database Access</b></p> <p>A very useful Perl feature is the ability to connect to a database. Perl scripts can be used to make information stored in a database available to the users who visit your Web site. Using databases to store information and using Perl scripts to access the information is an efficient method of displaying up-to-date information in a Web site. Perl can also allow users to manipulate the data in a database, such as adding or deleting records.</p>	<p><b>Troubleshooting</b></p> <p>Perl provides many features to help make the lengthy and tedious process of troubleshooting scripts easier. You can have Perl display warning messages about possible problems in your code and display more detailed and comprehensive error messages. You can also use strict error checking in a script, which instructs Perl to perform a complete check of the script for errors and enforce certain programming rules. The Perl debugger is also available and provides a controlled environment where you can execute and interact with a script. Executing a script in the debugger allows you to analyze and troubleshoot errors in the script.</p>

# USING PICO

Pico is a text editor available on most UNIX operating systems. Pico has features that make editing text easy, such as a spell checker and a search feature.

You must use the keyboard rather than the mouse to execute commands in Pico. Depending on your operating system, you may be able to use the mouse to perform some tasks, such as cutting and pasting text.

To start working in Pico, you specify the name of a file. If you do not specify a path for the file, Pico will look for the file in the current directory. If the file does not exist, Pico will create a new blank file.

If the file does exist, Pico will display the file for editing.

When you exit a file, Pico asks if you want to save your changes to the file. You can save the file with the current file name to overwrite an existing file or specify a new file name to create a new file.

If you have difficulty creating or accessing files, you may not have the appropriate permissions to create files and save files to a directory. Contact your system administrator or consult the documentation included with Pico or your operating system for more information.

## Extra

There are many commands that allow you to perform tasks in Pico. The commands you can use depend on your operating system and which version of Pico you are using.

COMMAND:	RESULT:
Ctrl+c	Display the line number and character position of the cursor.
Ctrl+a	Move the cursor to the beginning of the current line.
Ctrl+e	Move the cursor to the end of the current line.
Ctrl+v	Scroll down one page.
Ctrl+y	Scroll up one page.
Ctrl+w	Search for an instance of text.
Ctrl+l	Refresh the screen.
Ctrl+d	Delete the current character.
Ctrl+ ^	Start selecting text at the position of the cursor.
Ctrl+k	Cut the current line or block of selected text.
Ctrl+u	Paste the text you cut at the position of the cursor.
Ctrl+j	Justify the current paragraph.
Ctrl+t	Check the spelling of text.
Ctrl+r	Insert text from another file at the position of the cursor.
Ctrl+o	Save the file.
Ctrl+g	View Pico's help information.
Ctrl+x	Exit the file.

### USING PICO

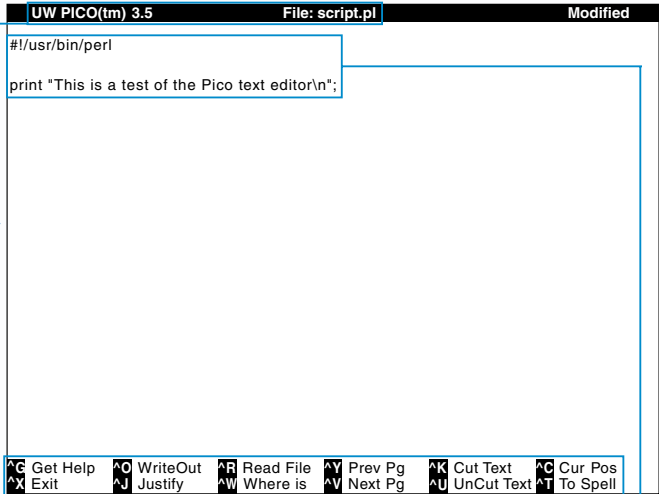


#### CREATE OR OPEN A FILE

1 Type **pico** followed by the name of a file you want to create or open. Then press Enter.

Pico displays the file on your screen.

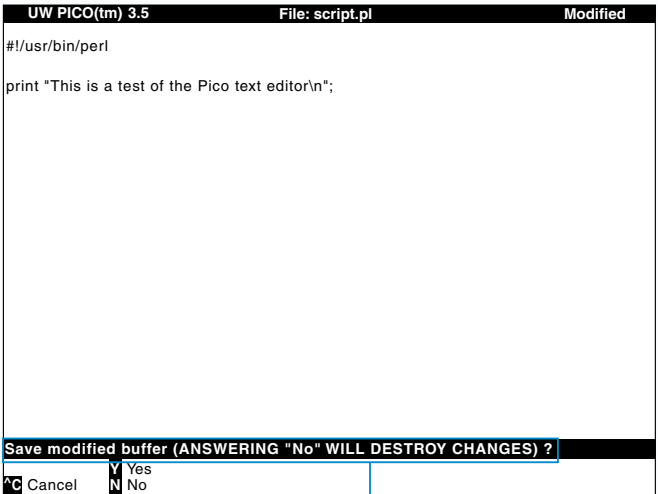
*Note: If you created a new file, a blank file appears.*



This area displays the version of Pico you are using and the name of the current file.

This area displays common commands you can use to work with Pico.

2 Enter the information you want to store in the file.

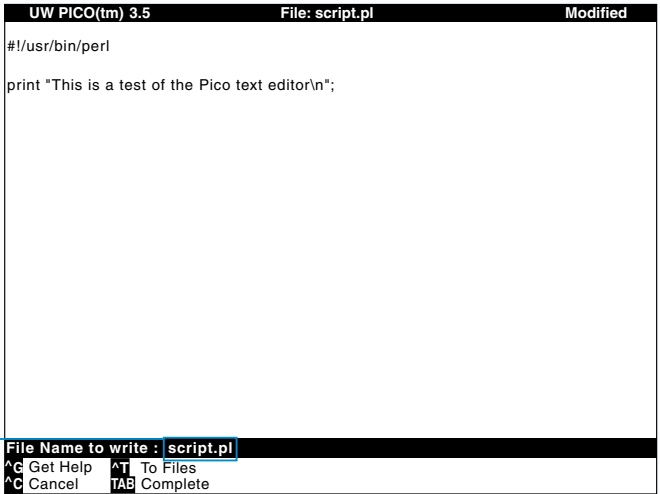


#### EXIT AND SAVE A FILE

3 To exit the file you are working with, type **Ctrl+x**.

A message appears, asking if you want to save the file.

4 Type **y** to save your changes.



The name of the file appears in this area.

5 To confirm the file name and save the changes, press Enter.

*Note: You can also type a new name and then press Enter to save the file with a different name.*

Pico saves the file.

# CREATE A PERL SCRIPT

A program you write in Perl is called a script. Perl scripts are text files that can be created using any text editor, such as Pico.

A Perl script consists of a series of Perl statements. A Perl statement is code, usually consisting of a function and alphanumeric data, which performs a specific action.

One of the most common functions used in Perl is the `print` function. The `print` function is usually used to display information on a screen. For example, the statement `print "Hello\n";`

will display the word "Hello." The newline (`\n`) character is often used with the `print` function to start a new line of text.

Most Perl statements end with a semi-colon (`;`). When a Perl statement spans several lines, a semi-colon is not required at the end of each line, only at the end of the statement.

Perl will ignore any spaces you place at the beginning of a statement. This allows you to indent statements in a script to help make the code easier to read.

## Extra

You can use comments to explain important or difficult sections of your code. This is helpful when you or other people need to modify or troubleshoot your script. Comments are ignored by Perl and will not appear when you execute your script. Any line of code that begins with the hash symbol (`#`) is treated as a comment in Perl code. If the comment you add requires more than one line, you must place a hash symbol at the beginning of each line.

TYPE THIS:

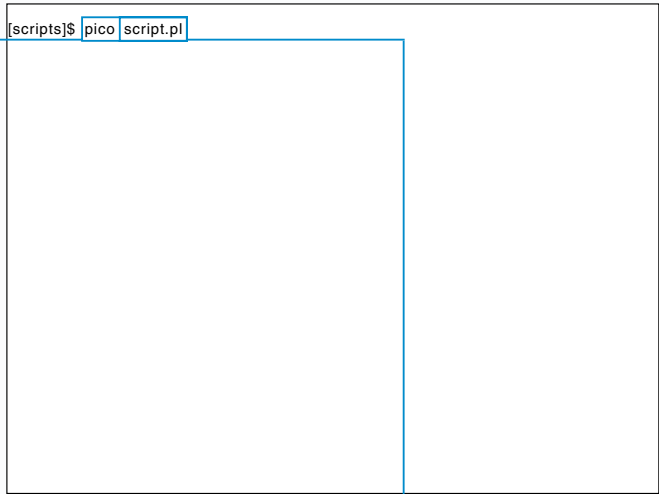
```
print "This code is executed.";      #The statement on this line is
#print "This statement is ignored.";#displayed.
```

RESULT:

```
This code is executed.
```

If you use a text editor that gives you a choice of formats when saving your script, you should save your script in a text-only format, such as ASCII. This ensures that Perl will be able to read and interpret your script. Most Perl scripts are saved with the `.pl` extension. This makes Perl scripts recognizable when you are searching for a script in a directory.

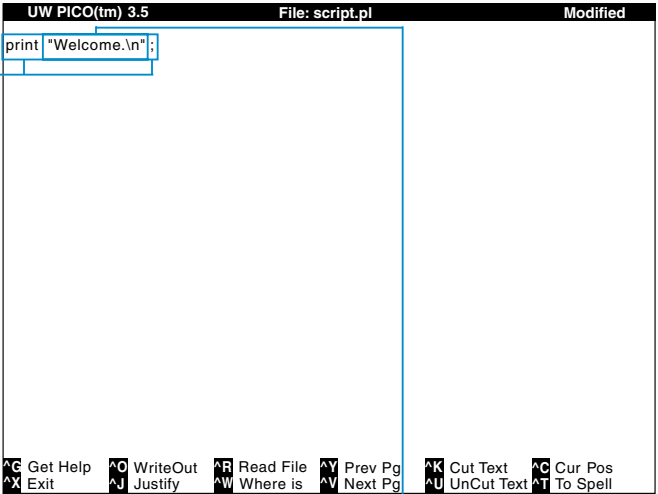
### CREATE A PERL SCRIPT



1 To start the Pico text editor, type `pico`.

2 Type the name of the script you want to create and then press Enter.

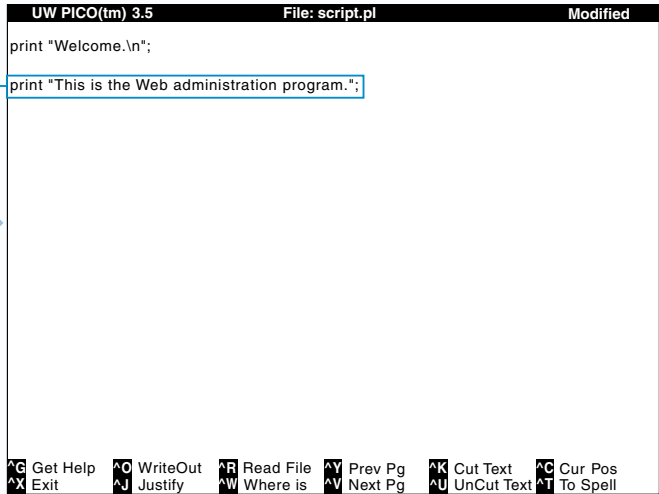
■ Pico displays a blank file where you can enter the code for a script.



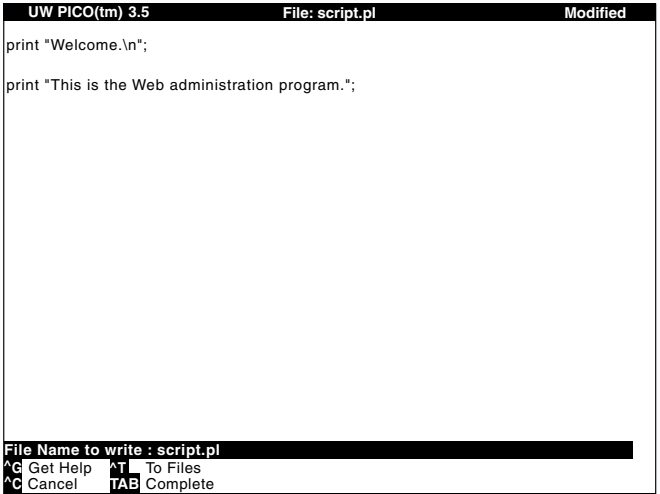
3 To display information on the screen when the script is executed, type `print ;`.

4 Position the cursor over the semi-colon and type the code you want to display.

■ Text must be enclosed in quotation marks.



5 Repeat steps 3 and 4 until you have entered all the code you want to include in the script.



6 To exit and save the script, perform steps 3 to 5 on page 5.

■ You can now execute the script.

# EXECUTE A PERL SCRIPT

After you create a Perl script, you must instruct the Perl program to interpret and then execute the script. The Perl program is called an interpreter. The Perl interpreter reads a script line by line and then compiles the script into a format a computer understands. Once the script is compiled, the script is executed.

To start the Perl interpreter, you use the `perl` command followed by the name of the script you want to execute. Your operating system will use the path environment variable to determine where to find the Perl interpreter. If the path to the Perl interpreter is not stored in the path environment

variable, you may have to include the full path to the Perl interpreter with the `perl` command.

You enter the `perl` command at the command prompt. On a graphic-based UNIX system, the command prompt may be available in a terminal window. On a Windows-based system, you access the command prompt through the DOS prompt or the command prompt window.

When a script is executing, you usually cannot interact with the program unless interaction, such as accepting typed information, is built into the program.

## Extra

When you are working on a Windows-based operating system, use backslashes (\) rather than forward slashes (/) to separate the directories in a path to a Perl interpreter. If you use forward slashes, the system may not recognize the path. In Windows-based operating systems, the path you specify may also need to include the drive letter for the device that stores the Perl interpreter.

Type:  
C:\Perl\bin;  
  
Not:  
/Perl/bin;

You can check the configuration information for Perl by entering the `perl -V | more` command at the command prompt. This is useful if you need to determine the settings of the Perl interpreter when creating a complex script that requires special settings. The configuration information includes information such as the type of operating system Perl is running on and paths to the locations the Perl interpreter will search for various files. If `--More--` appears at the bottom of your screen, you can press Enter to display more information.

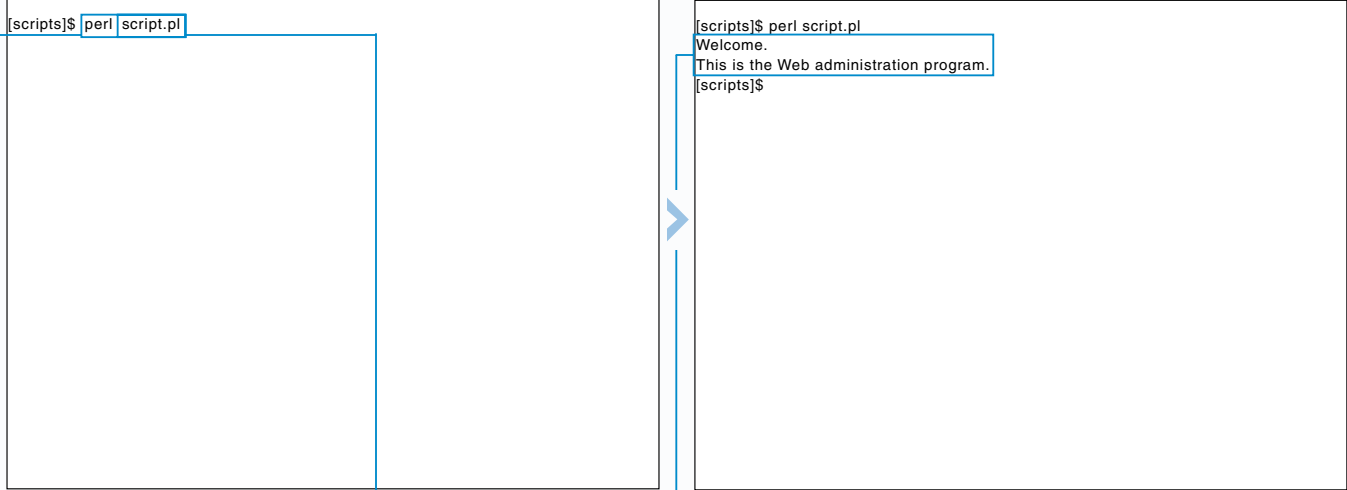
TYPE THIS:

[script]\$ perl -V | more

RESULT:

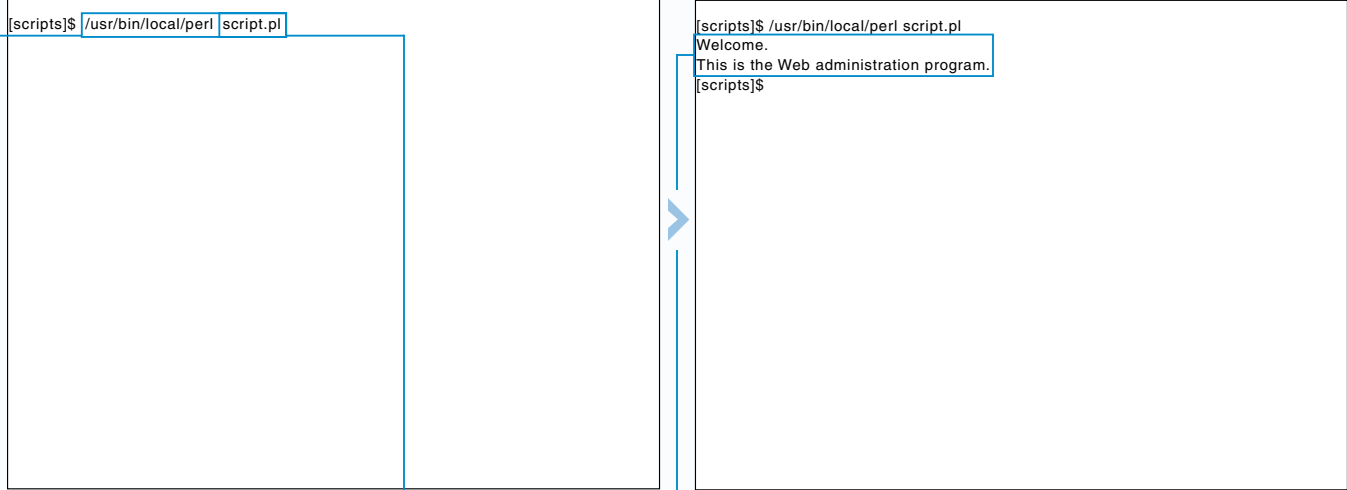
Summary of my perl5 (5.0 patchlevel 5 subversion 3) configuration:  
Platform:  
osname=linux, osvers=2.2.5-22smp, archname=i386-linux  
uname='linux porky.devel.redhat.com 2.2.5-22smp #1 smp wed jun 2 09:11:51 edt 1999 i686 unknown '  
hint=recommended, useposix=true, d\_sigaction=define  
usethreads=undef useperlio=undef d\_sfio=undef  
--More--

## EXECUTE A PERL SCRIPT



- 1 To start the Perl interpreter, type `perl`.
- 2 Type the name of the script you want to execute and then press Enter.
- The script executes and Perl generates the results of the script.

## EXECUTE A PERL SCRIPT-SPECIFY A PATH



- 1 To start the Perl interpreter when the path is not stored in the path environmental variable, type the full path to the interpreter.
- 2 Type the name of the script you want to execute and then press Enter.
- The script executes and Perl generates the results of the script.

# CREATE A SELF-EXECUTING PERL SCRIPT

Perl allows you to create a self-executing script. On UNIX systems, a self-executing script will start the Perl interpreter and execute the script when you enter the name of the script at the command prompt. Most Web servers require scripts to be self-executing.

To create a self-executing Perl script, you include the location of the Perl interpreter on the first line of the script. On most UNIX systems, the location of the Perl interpreter is /usr/bin/perl. You prefix the location with the #! symbols.

You will also need to set the execute permission at the command prompt using the `chmod +x` command. This allows all users to execute the script.

You should make sure the correct file permissions are set so Perl can execute the script. For example, if the script is stored in a directory that only you have permission to access, you should set the permissions to allow Perl to access the directory as well. For information about UNIX file permissions, see the top of page 156.

## Extra

On Windows-based operating systems, `#!/usr/bin/perl` is interpreted as a comment, so you cannot use this method to create a self-executing script. If you are using ActivePerl on a Windows-based operating system, the `.pl` extension is automatically associated with the Perl interpreter. When ActivePerl makes this association, any files you create with the `.pl` extension will display an icon you can double-click to execute the file.

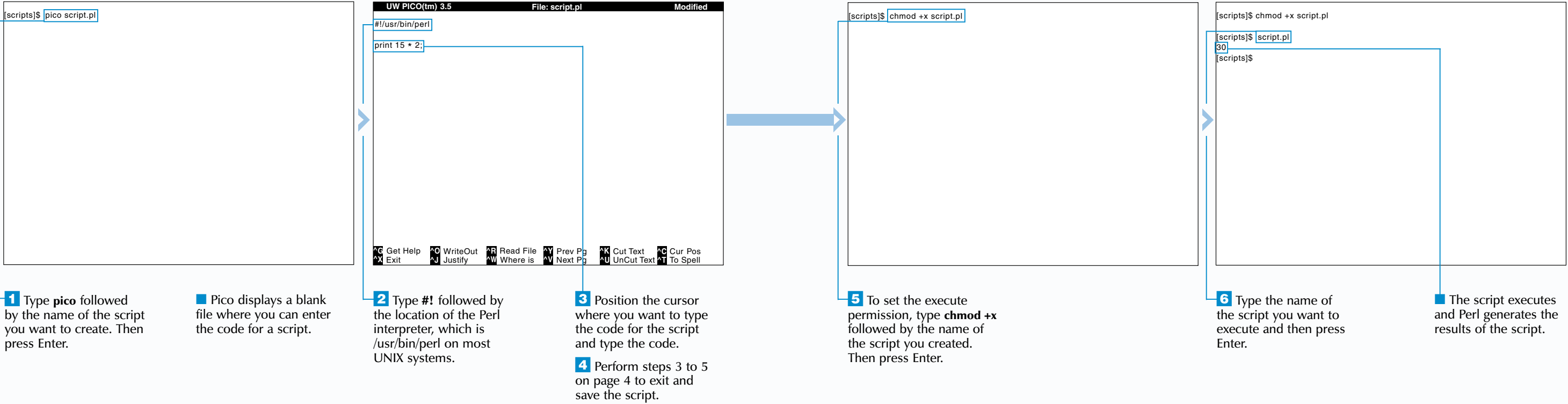
If the `.pl` extension is not associated with the Perl interpreter, refer to the documentation included with your operating system for information about associating the `.pl` extension with the Perl interpreter.

If you need to determine the location of the Perl interpreter on your UNIX operating system, enter the `whereis perl` command at the command prompt. If this command is successful, Perl will display the location. If the command fails, you can enter the `find / -name perl -print` command at the command prompt to determine the location.

### Example:

```
[scripts]$ whereis perl
perl: /usr/bin/perl5.00503 /usr/bin/perl /usr/man/man1/perl.1
```

## CREATE A SELF-EXECUTING PERL SCRIPT





# USING COMMAND LINE OPTIONS

There are several command line options you can use when starting the Perl interpreter. Command line options, also called switches or flags, can be used to specify the way you want to work with a script, perform a specific operation or obtain information about the Perl interpreter.

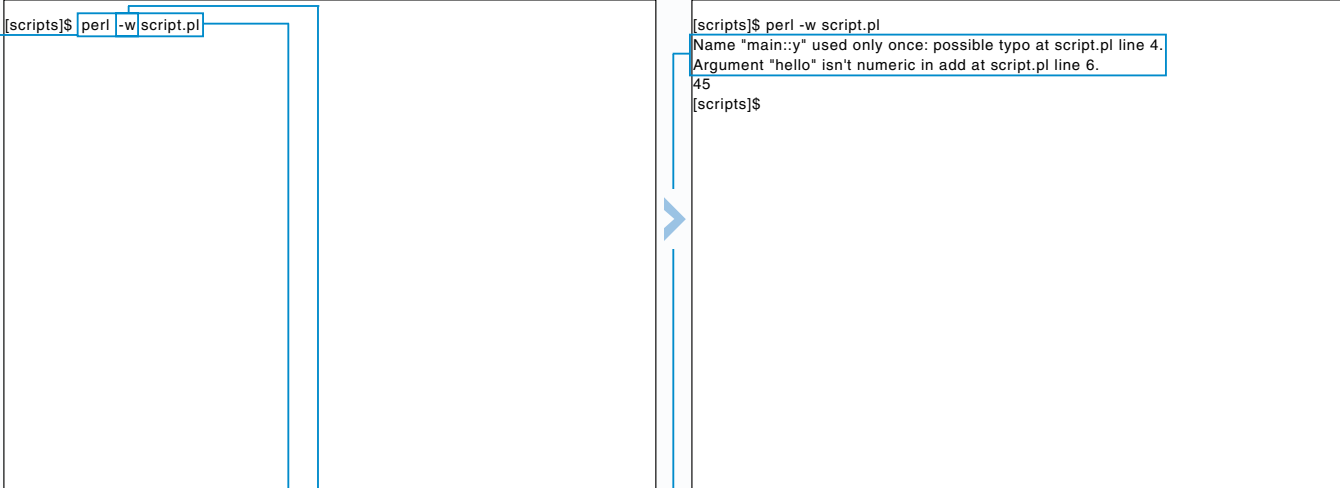
After entering the `perl` command at the command prompt, you can enter the command line option you want to use. A command line option consists of a dash (-) and a letter that represents an option. Some command line options take arguments, which are usually entered directly following the option.

Many command line options are used to work with a script. You must specify the name of the script you want the command line option to affect, such as `perl -d debug.pl`.

To enter more than one command line option, use a single dash (-) followed by each option you want to use, such as `perl -nTU script.pl`.

If your script is self-executing, the command line option is placed in the script after the code that specifies the location of the Perl interpreter, such as `#!/usr/bin/perl -w`.

## USING COMMAND LINE OPTIONS



- 1 At the command prompt, type **perl** and then press the Spacebar.
- 2 Type the command line option you want to use and, if appropriate, an argument for the option. Then press the Spacebar.
- To use the command line option to work with a script, type the name of the script you want to affect and then press Enter.

## COMMAND LINE OPTIONS

There are many common command line options you can use when starting the Perl interpreter. Optional arguments are shown in brackets [].

Command Line Option:	Description:
-0[octal]	Specifies the input record separator as an octal number. If no argument is specified, the null character is used.
-a	Enables autosplit mode when used with -n or -p. This mode splits lines into words and stores the words in the @F array.
-c	Checks the syntax of a script without executing the script.
-d	Executes the script in the Perl debugger.
-d:modulename	Executes the script using the specified debugging module.
-e 'command'	Executes one line of code at the command line instead of a script. Several -e command line options can be used.
-F/pattern/	Specifies the split pattern for autosplit mode when used with -a.
-h	Prints a list of the command line options.
-i[extension]	Enables simultaneous reading and writing of a file. If an extension is provided, a backup copy of the file is created.
-Idirectory	Specifies a directory to search for modules.
-l[octal]	Automatically removes the output record separator. Sets the value of the output record separator when an octal number is provided.
-[mM][-]module..	Includes the specified module before executing the script.
-n	Uses a while (<>) { } loop around your script.
-p	Uses a while (<>) { } loop around your script and prints lines.
-P	Runs the script through the C preprocessor before compilation.
-s	Allows switches placed after the script name to be interpreted.
-S	Looks for the script using the PATH environment variable.
-T	Turns on taint checks.
-u	Dumps the core after parsing the script.
-U	Allows unsafe operations.
-v	Prints the version number and patch level of Perl.
-V[:variable]	Prints Perl interpreter configuration information. Prints a variable's value when a variable name is provided.
-w	Turns on warnings for compilation of the script. This is a recommended command line option.
-x[directory]	Extracts a script from a message, starting at a line that begins with #! and contains perl. If a directory is provided, Perl switches to the directory before executing the script.

# UPLOAD A PERL SCRIPT TO A WEB SERVER

A Perl script you create must be stored on a Web server to be available on the Web. You can use a File Transfer Protocol (FTP) program to transfer a script stored on your computer to a Web server. WS\_FTP Pro for Windows is a popular FTP program. In the example below, we use WS\_FTP Pro version 6.5. You can obtain the latest version of WS\_FTP Pro at the [www.ipswitch.com](http://www.ipswitch.com) Web site.


Before you can transfer a script to a Web server, you must set up a connection to the server. You only need to set up a connection to a Web server once. After you set up a connection, you can connect to the server at any time.

To set up a connection to a Web server, you must know the address of the server, your user ID and your password. If you do not know this information, contact the system administrator.


Many FTP programs allow you to save your password, which saves you from having to retype your password each time you transfer a script to the Web server. When you save your password, anyone who uses your computer will be able to connect to the Web server, so you should not save your password if other people will have access to your computer.

## Extra

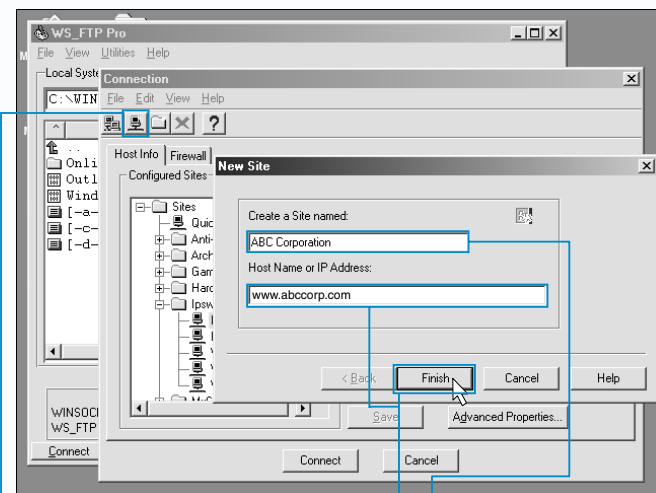
If you are running the UNIX operating system on your computer, a standard FTP program should already be installed on the computer. To start the FTP program, type **ftp** at the command prompt. If the program is not installed, contact your system administrator.

You can create a new directory to store and organize your Web server connections. In the Configured Sites area of the Connection dialog box, click the directory you want to store the new subdirectory. If you want to create a main directory, click the Sites directory. Then click  to create the directory. In the New Folder dialog box, type the name of the new directory and then click Finish.

FTP programs allow you to set up multiple connections. This is useful if you want to use one connection to transfer your scripts to a Web server and another connection to download files from a different Web server. Perform steps 1 to 9 below for each connection you want to set up.


You can delete a connection you no longer need. Click the connection in the Configured Sites area of the Connection dialog box and then click  to delete the connection. To confirm the deletion, click Yes in the dialog box that appears.

## SET UP A CONNECTION



**1** Start the FTP program you will use to transfer your script to a Web server.

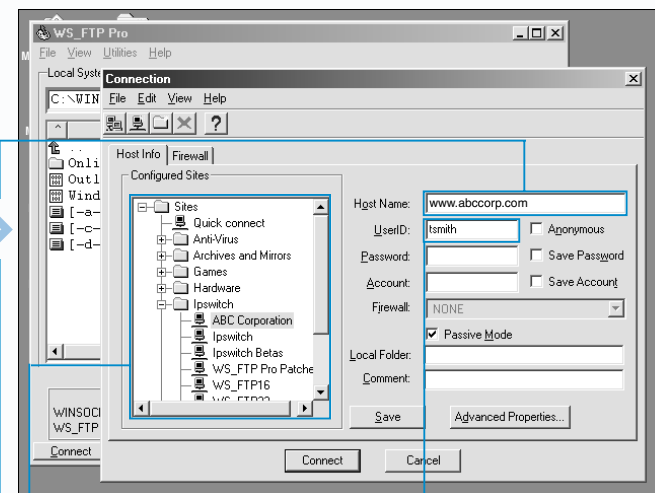
■ The Connection dialog box appears.

**2** Click  to display the New Site dialog box and set up a new connection to the Web server.

**3** Type a name for the connection.

**4** Click this area and type the address of the Web server you want to transfer your script to.

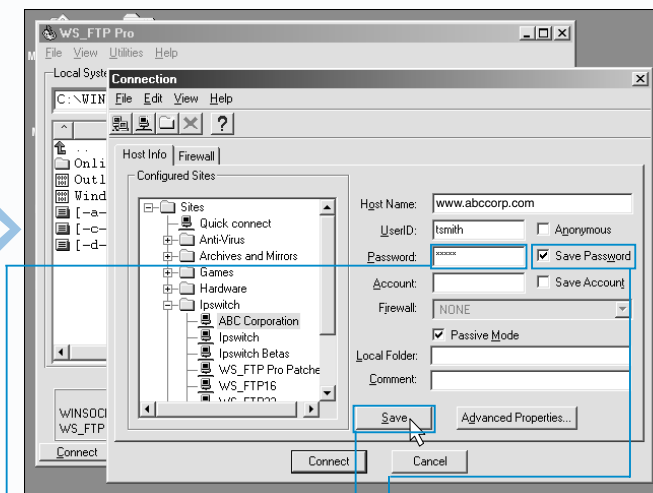
**5** Click Finish to continue.



■ The name of the connection appears in this area.

■ The address of the Web server you specified appears in this area.

**6** Click this area and type your user ID.

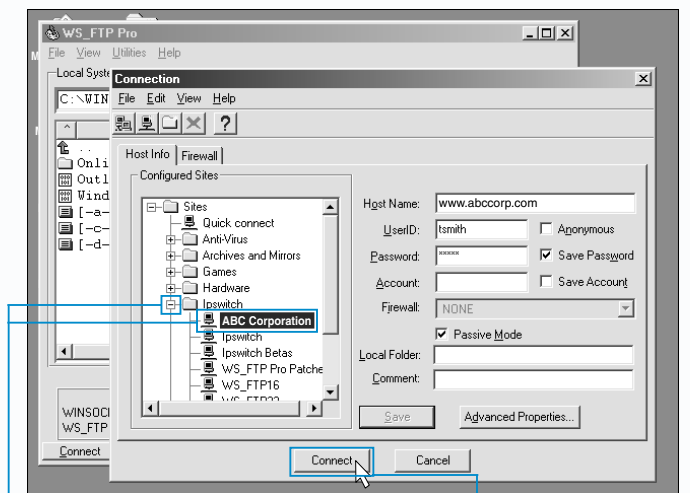


**7** Click this area and type your password. A symbol (x) appears for each character you type to prevent others from seeing your password.

**8** To save your password so you will not need to retype the password again later, click this option (☐ changes to ☒.

**9** Click Save to have the program store the information you entered for the connection.

## CONNECT TO THE WEB SERVER



**1** Click the connection for the Web server you want to transfer your script to.

■ If the connection you set up is not displayed, click the plus sign (+) beside the Ipswitch directory (+ changes to -).

**2** Click Connect to connect to the Web server.

# UPLOAD A PERL SCRIPT TO A WEB SERVER (CONTINUED)

Once you have connected to a Web server, you can transfer information to the server.

You can transfer a single script, multiple scripts or an entire directory to the Web server at once. Before you transfer files and directories to a Web server, you must locate the directory on the server you want to transfer your scripts to. The directory used to store Perl scripts on a Web server is often named "cgi-bin." If you do not know the name of the directory, contact the system administrator.

If your connection to the Web server is idle for an extended period of time, the server may automatically disconnect you. This helps ensure that the Web server's resources are available for other people who need to access the server.

If you later make changes to the scripts stored on your computer, you must transfer the updated scripts to the Web server. The updated scripts will replace the old scripts on the server. When you transfer updated scripts, a message may appear indicating that the updated scripts will replace the old scripts.

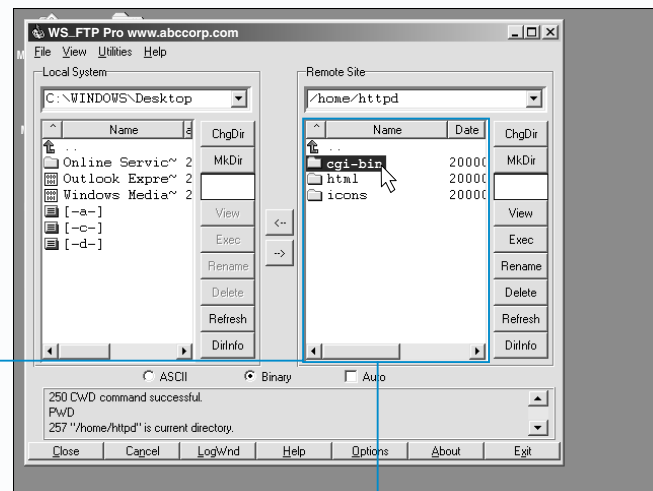
## Extra

If you have difficulty transferring a script to a directory on a Web server, you may not have the appropriate permissions to transfer files to the directory. To change the permissions for the directory, contact the system administrator.

WS\_FTP Pro allows you to set the permissions for a file you transferred to a UNIX Web server. Select the script you want to change in the right pane of the WS\_FTP Pro window and then right-click the script. In the menu that appears, select the chmod (UNIX) option. In the Remote file permissions dialog box, select the permissions you want for the script and then click OK.

You should delete scripts you no longer need from the Web server. This helps save storage space on the Web server. To delete a script from the Web server, select the script in the right pane of the WS\_FTP Pro window and then click the Delete button. To confirm the deletion, click Yes in the dialog box that appears.

## TRANSFER SCRIPTS TO THE WEB SERVER

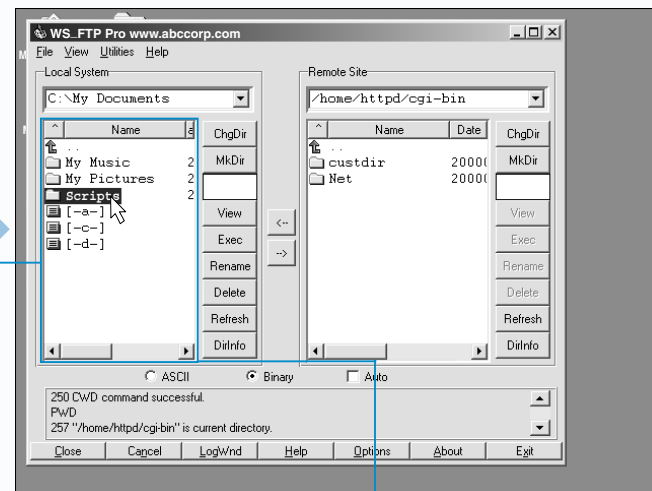


■ The WS\_FTP Pro window appears.

■ This area displays the directories and files stored on the Web server.

1 Locate the directory you want to transfer your scripts to. In many cases, the directory is named "cgi-bin."

2 Double-click the directory to display the contents of the directory.

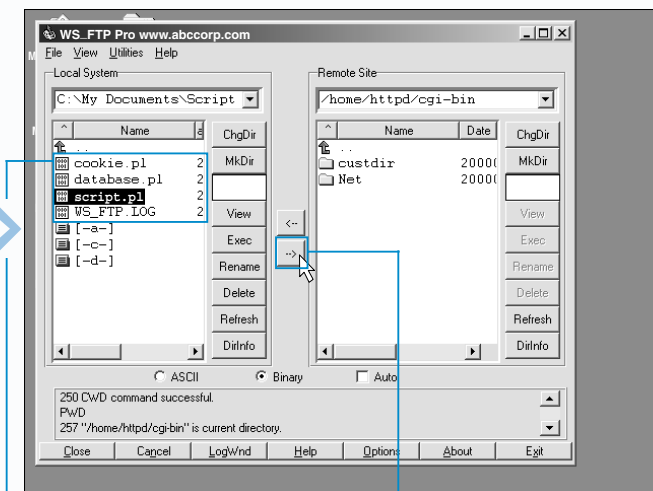


■ This area displays the directories and files stored on your computer.

3 Locate the directory that contains the script you want to transfer to the Web server.

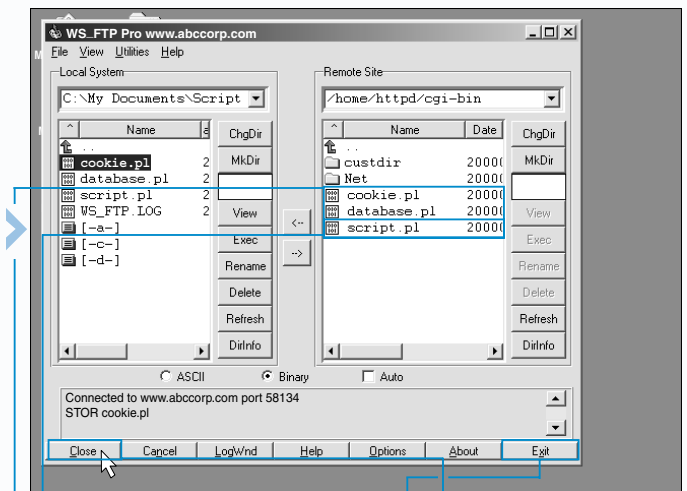
Note: You can double-click to move up one level in the directory structure.

4 Double-click the directory to display the contents of the directory.



5 Click the script or directory you want to transfer to the Web server.

6 Click to transfer the script or directory to the Web server.



■ The script or directory appears on the Web server.

7 Repeat steps 3 to 6 for each script and directory you want to transfer.

8 Click Close to end the connection to the Web server.

9 Click Exit to exit the program.