**Forms and Introduction to CGI**

Because forms are usually so bound up with scripting languages, some discussion of scripts and scripting languages, and how scripts relate to forms, is necessary. This discussion will entail some jumping around between regular HTML and more esoteric CGI scripting information.

As discussed in Book 7, a form is created using the FORM tag. It goes in the BODY of a regular HTML page, just like most other HTML tags.

Example:

<form>

<!-- one or more form elements, with their accompanying HTML context -->

<!-- a submit button -->

</form>

Now we need to talk about CGI scripts.

CGI stands for Common Gateway Interface. All web servers have CGI capabilities. The CGI is the means for web pages to communicate with the server directly, with power to read and rewrite information on the server hard drive, whether in a database or in a more ordinary directory or file.

CGI scripts can be written in any programming language, including C, C++, Java, Perl and PHP. Perl and PHP are the two most common programming languages used for writing CGI scripts as regards web pages and forms.

Because CGI scripts have the power to rewrite or erase a web server's hard drive, they are considered to be a potential security hazard or security breach by web server administrators (with good reason!). Unless you have special privileges or direct physical access to an actual web server computer, you will probably not be allowed to write CGI scripts yourself.

Web servers have something called the "cgi-bin", which is simply a directory on the server which contains all of the CGI scripts for that server. These CGI scripts may be accessed if you know the URL for the script and its "cgi-bin" directory. Most ISPs (Internet Server Providers) have a collection of CGI scripts available to their customers, and have published FAQ sheets on their website which give information on how to connect your HTML pages and HTML forms to these pre-made CGI scripts.

Have you ever seen a "counter" on a web page which tells you how many people have visited that page? That counter is created using a commonly-available CGI script. There is also a very common CGI script which takes the contents of an HTML form, converts that information into text, inserts the text into the body of an email message, and sends that email to whomever has been specified. These scripts, and many more, are usually readily available in the cgi-bin of any ISP.

If you register your own domain name in combination with an ISP, your ISP server administrator may be able to give you a separate cgi-bin for your domain. If you arrange for a separate cgi-bin for your domain, you will no longer have access to the main cgi-bin for the ISP (with its accompanying scripts), but you WILL be able to write your own CGI scripts, if that is important to you. Again, if you work for a company with its own web server and you have access privileges to the actual web server computer, you will probably be able to write CGI scripts and drop them directly into the cgi-bin on that computer.

CGI scripts usually come with one of two dot-extensions on the file name, ".cgi" (for a generic CGI script) and ".pl" (for a Perl script specifically).  
  
"The Common Gateway Interface (CGI) is a standard for interfacing external applications with information servers, such as HTTP or Web servers. A plain HTML document that the Web daemon retrieves is static, which means it exists in a constant state: a text file that doesn't change. A CGI program, on the other hand, is executed in real-time, so that it can output dynamic information.  
  
For example, let's say that you wanted to "hook up" your Unix database to the World Wide Web, to allow people from all over the world to query it. Basically, you need to create a CGI program that the Web daemon will execute to transmit information to the database engine, and receive the results back again and display them to the client. This is an example of a gateway, and this is where CGI got its origins.  
  
The database example is a simple idea, but most of the time rather difficult to implement. There really is no limit as to what you can hook up to the Web. The only thing you need to remember is that whatever your CGI program does, it should not take too long to process. Otherwise, the user will just be staring at their browser waiting for something to happen."

Now, back to the FORM tag.

The FORM tag has a few attributes which relate to connecting it to a script. You will need to understand what they are, what they do, and what type of scripts they relate to.

Two attributes for the FORM tag are used when connecting a form to a CGI script: METHOD and ACTION.

Tag: **FORM**  
Attribute: **METHOD**  
Value: **get** (default) or **post**  
Description: this attribute defines the manner in which the form information is conveyed to the script.

Example (abbreviated):

<form method="post">

<!-- one or more form elements, with their accompanying HTML context -->

<!-- a submit button -->

</form>

Tag: **FORM**  
Attribute: **ACTION**  
Value: a URL (to the CGI script in the cgi-bin in question)  
Description: the ACTION attribute of the FORM tag connects that form to a CGI script; when the submit button for the form is pressed, the web browser seeks out the CGI script at the URL indicated by the value of ACTION and causes the web server to execute that CGI script.

**Example (abbreviated):**

<form method="post" action="myScript.pl">

<!-- one or more form elements, with their accompanying HTML context -->

<!-- a submit button -->

</form>

Again, both the METHOD and the ACTION attributes are required when connecting a form to a CGI script. Notice that it is unnecessary to reference cgi-bin.