

Apache CGI And Perl On Windows XP (Part II)

| | | |
|-----------|------------------------|----|
| 2008-2-15 | Generate this document | HL |
| | | |

Background

This is the 2nd part of document derived from the narrative from the reference [1]. This part covers value passing between the browser and CGI program.

Value Passed Via Query String

2.1 Values Passed Via Query String

A simple value can be passed in both directions, e.g., from your browser to CGI or from CGI to browser as illustrated in Table 1 and Figure 1.

| Parameter Passing Direction | Method | Note |
|-----------------------------|------------------------------------|------|
| 1. Browser to CGI | Use HTML's <form> and <input> tags | |
| 2. CGI to Browser | Printf(.) as HTML to browser | |

Table 1. Value passing between CGI and the browser.

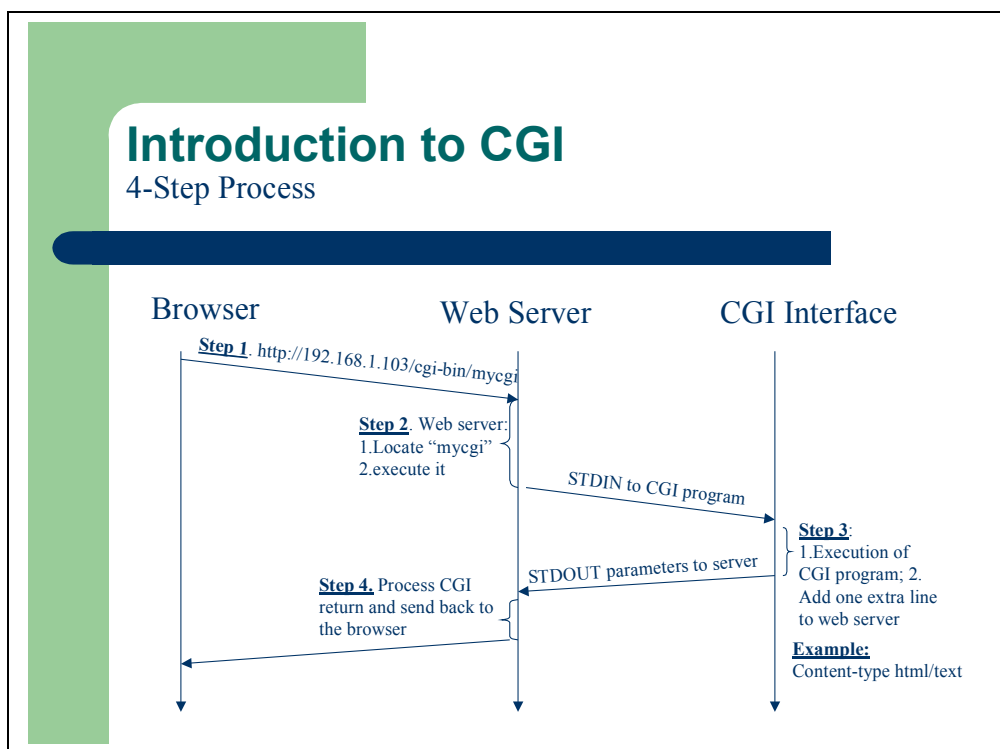


Figure 1. Value back and forth between browser and CGI.

Example 1. Value from browser to CGI:

First the html code is given below in Figure 2 and its execution in Figure 3.

```

<html>

<head>
<title> Boa Web Server Testing </title>
</head>

<body>

<h1>Integer Addition Testing</h1>

<form action="/cgi-bin/testcgi" method="get">

<table>

<tr><td>Enter first number</td>

<td><input type=text name=number1></td>

</tr>

<tr><td>Enter second number</td>

<td><input type=text name=number2></td>

</tr>

```

Figure 2. Html passes value to CGI.

CGI/Browser Interface Example

```

<html>
<head> <title> Boa Web Server Testing </title>
<body>
<h1>Integer Addition Testing</h1>
<form action="/cgi-bin/testcgi" method="get">
<table>
<tr><td>Enter first number</td>
<td><input type=text name=number1></td>
</tr>
<tr><td>Enter second number</td>
<td><input type=text name=number2></td>
</tr>
<tr><td colspan=2>
<input type = submit value = "Calculate Sum" >
</td>
</tr>
<input type=hidden name=cmd value="addition">
</table>
</form>
</body>
</html>

```

Figure 3. Execution of the html code.

Table 4 Illustrate how browser passes value to CGI.

| | | |
|---|--|------|
| <code><form action="/cgi-bin/testcgi" method="get"></code> | Invoke CGI program | Note |
| <code><input type=text name=number1></code> | Pass value "number1" as text to CGI | |
| <code><input type = submit value = "Calculate Sum" ></code> | Submit calculate to CGI | |
| <code><input type=hidden name=cmd value="addition"></code> | Pass value "cmd" as "addition" without showing on browser screen | |

Table 4. Browser passes values to CGI

Now let's look how CGI program in C uses a function `getenv("QUERY_STRING")` to capture the value by parsing the Query_String as in Figure 4 and Table 5.

```

unsigned char *vstr;

...

vstr = (unsigned char *) getenv ( "REQUEST_METHOD");
if(vstr == NULL) return 0;
    if(strcmp((const char *)vstr,"POST") == 0)
    {
        ....
        vstr = (unsigned char *) getenv("CONTENT_LENGTH");
        ....
        vstr = (unsigned char *) getenv("QUERY_STRING");
        ....
    }

```

Figure 4. CGI C program uses a function `getenv("QUERY_STRING")` to capture the value.

Example 2: Now let's take a look how the above example passes a value from CGI to back to browser as in Figure 5.

```

main ()
{
    int x, y;
    unsigned char *str1,*str2, *cmd;
    cmd = getval((unsigned char *) "cmd");
    printf("Content-type: text/html\n\n");
    printf("<html> <head> <title> Web Serve </title>\n\n");
    if (strcmp((const char *)cmd, "addition") == 0)
    {
        printf("<body>\n\n");
        printf("    <h1>Integer Addition Testing Result\n\n");
        str1 = getval((unsigned char *) "number1");
        str2 = getval((unsigned char *) "number2");
        if (str1 == NULL || str2 == NULL) {
            printf("<p>Input data error\n");
        } else {
            x = atoi ((const char *)str1);
            y = atoi ((const char *)str2);
            printf("<p>The sum of %d and %d is %d\n",x,y,x+y);
        }
    } else {

```

```
        printf("<p>Sorry, the request is invalid.\n");  
    }  
    printf("</body></html>\n");  
    exit (0);  
}
```

Figure 5. CGI uses printf(..) passes value back to the browser.

Example 3: Value from CGI to browser based on C programming.

passed by bring up the form in your browser.

2.2 Multiple Values Passed Via Query String

CGI Programming

We will take

3.1 Windows

See separate report on this topic.

3.2. Installing

" should be checked.

3.3 Configure Apache and Enable CGI

First go to "My Documents" and make a new folder "My Website". This is where you're going to store your web pages and CGI programs.

3.4 Writing Perl Programs

. The next part, `/usr/bin/perl`, is the location (or *path*) of the Perl interpreter.

Figure 4. Perl is running.

Note: The difference of perl for Windows is listed in Table 2 below.

`bin/perl`.

Debugging via Log Files

Debugging by examining webserver log files. The server logs are located at /usr/local/etc/
The

Table 3. Environment variables.

The following perl program prints environment variables, see Figure 5 and 6.

Figure 5. Perl to print environment variables.

Figure 6. Print environment variables.

3.6 Sending Data from Web Form to CGI

There are two ways to send data from a web form to a CGI program: GET and POST as in Table 3.

Table 3. Parameter passing.

Now a simple program to get variables from CGI to the web server or the browser is listed in Figure 7. Note you will have to make sure that you have “printenv.pl” in the /cgi-bin directory. The execution result is given in Figure 8.

Figure 7. HTML code for parameter passing from CGI.

Figure 8. The HTML page to get parameters from Perl via CGI.

Reference

1. <http://www.cgi101.com/learn/connect/winxp.html>
2. <http://www.cgi101.com/book/ch1/text.html>

(End)