

# Dynamic Web Documents

COMP476  
Networked Computer Systems

## Server Execution

- Dynamic web pages involve the execution of a program on the web server.
- The program on the server usually generates an HTML document which is sent to the client to be displayed in the web browser.
- Dynamic web pages are often associated with web forms that provide input data to the programs executed on the server.

## Methods for creating Dynamic Documents

- Common Gateway Interface (CGI)
- Active Server Pages (ASP)
- Java Servlets
- *and others*

## Common Gateway Interface

- CGI is a standard defining how a program will interact with a server to generate dynamic documents.
- Can use any programming language.
- Can generate any type of output, although HTML is most common.

## Passing Data to the Server

- Forms are web pages that send the contents of fields to an active document
- Each field has a name defined by the author.
- When you press the “submit” button, an HTTP command is sent to the server requesting execution of a program.
- The form data is sent with the HTTP command.

## Different Methods

- GET
  - Data is passed to the server application as an environmental variable.
- POST
  - The application can read the form data as input from standard input.

## Sending data with GET

- Parameters are sent to the server at the end of the URL separated by a question mark.

`/file?fieldname=value&field2=value2`

## Sending Characters to the Server

- Fields are separated by “&”
- Blanks or spaces are translated to “+”
- Special characters are sent in their hexadecimal form preceded by a “%”

## Passing Data to the CGI Program using GET

- Parameters are passed in environment variables.
- The environment variable `QUERY_STRING` contains the input from the browser.

- C programs can start
 

```
int main(int argc, char* argv[],
        char* env[])
```

## CGI Output

- The output “printed” to stdout by a CGI program is sent to the client browser.
- The CGI program normally prints HTML.
- The first line must identify the data type:
 

```
Content-type: text/html
```
- The CGI program can direct the sever to send an existing file to the browser
 

```
Location: /new/file.txt
```

## Example CGI GET form

```
<HTML> <HEAD>
<TITLE>GET Example</TITLE>
</HEAD> <BODY>
<H1>Example of using GET</H1> <HR>
<P> This form sends data to the server using the GET method.</P>
<FORM METHOD="GET" ACTION="cgi-bin/gettest.exe">
<P>What is the name of this course?</P>
<BLOCKQUOTE> <P>
<INPUT TYPE=TEXT NAME="course" SIZE=35>
<BR> </P> </BLOCKQUOTE> <P>
Are you a computer science major?</P>
<BLOCKQUOTE> <P>
<INPUT TYPE=RADIO NAME="csmajor" VALUE="Yes" CHECKED>
Yes <INPUT TYPE=RADIO NAME="csmajor" VALUE="No">
No<BR> </P>
</BLOCKQUOTE>
<INPUT TYPE=SUBMIT VALUE="Submit Form">
<INPUT TYPE=RESET VALUE="Reset Form">
</FORM> <HR> </BODY> </HTML>
```

## Example GET CGI program

```
#include <iostream>
#include <string>
using namespace std;

int main(int argc, char* argv[]) {
    char *qstring;
    /* GET the data from the web form */
    qstring = getenv("QUERY_STRING");

    cout << "Content-type: text/html\n\n";
    cout << "<html> <head>";
    cout << "<title>Example CGI GET output</title>";
    cout << "</head><body>";
    cout << "<br>The input from the form was:<br>";
    cout << qstring;
    cout << "</body></html>";
    return 0;
}
```

- The previous C++ program has been compiled and stored on the class web server

<http://williams.comp.ncat.edu/Networks/cgiExample.htm>

## Active Server Pages

- Active Server Pages is a Microsoft developed technology for sending dynamic Web content to the client.
- An ASP file (extension `.asp`) is a server-side text file processed in response to a client browser request.

## ASP Execution

- When a server receives an HTTP request from a client, it directs it to be processed by the appropriate Active Server Page.
- VBScript is the de facto language for ASP scripting.
- ASP often involves interacting with a database.
- The server then returns its result to the client.
- This is normally in the form of a HTML document – but other formats (e.g., images) are possible.

## ASP Files

- An ASP file is a combination of HTML and scripting code.
- When a client requests an ASP file, the ASP file is parsed by an ActiveX component named `asp.dll`
- Scripting code is executed as it is encountered.
- The `@LANGUAGE` statement specifies the scripting engine (VBScript by default).

## Invoking ASP

- Normally .asp files reside on the IIS server in folder C:\inetpub\wwwroot or some subfolder thereof.
- So, for example, to execute **clock.asp** on the class web server, you type the following in the browser  
**http://williams.comp.ncat.edu/clock.asp**

## ASP Example

The following clock.asp file displays in the browser:

Simple ASP Example

9:23:00 AM

The time changes each minute.

```
<% @LANGUAGE=VBScript %>
<% Option Explicit %>
<% ' clock.asp %>
<HTML>
<HEAD>
<TITLE>A Simple ASP Example</TITLE>
<META HTTP-EQUIV="REFRESH"
      CONTENT="60; URL=CLOCK.ASP">
</HEAD>
<BODY>
<P>Simple ASP Example</P>
<TABLE>
<TR>
<TD> <% =Time() %> </TD>
</TR>
</TABLE>
</BODY>
</HTML>
```

## Explanation of ASP Example

- The line  
`<% @LANGUAGE=VBScript %>`  
specifies VBScript as the scripting language.
- All VBScript code is enclosed in the scripting delimiters,  
`<% ... %>`.
- The enclosed code is executed on the server.
- The META tag sets the refresh interval for the page.
- The CONTENT attribute specifies the number of seconds until the file with the specified URL is requested.

## Explanation of ASP Example (cont.)

- The content of the only cell in the table is  
`<% =Time() %>`
- This calls VBScript function `Time` to get the current server time.
- This statement is an abbreviation for  
`<% Call Response.Write( Time() ) %>`
- The `Response` object provides functionality for the client to write information on the screen of the client's browser.

## Emailers

One thing that can be done with the form data is to email it to someone. The steps that are required are straightforward:

1. Provide the form for inputting the information you require. This can be done with standard HTML.
2. Create a cgi-bin program that will read in the data and then clean up any extraneous information and retrieve any needed variables.
3. The cgi-bin program then emails the information out to the correct person and sends back an HTML confirmation.