Server-side scripting

- Introduction
- Web-based application structure
- Server-side scripting
- JavaServer Pages
- Summary

Web-based application structure

- Need to separate
  - Presentation (HTML)
  - Business logic used to dynamically update page
- E.g., portfolio account page
Portfolio account page

- Enter ID + password into HTML form
- ==> See status of account
  - dynamically generated
- Can use several technologies to create dynamically

Server-side technologies

- CGI
- Server plugin
- Applet
- Servlet
- Server-side scripting
CGI

- App handles form request
- Lay out HTML form (designer)
- Write CGI script (programmer)
- If layout needs change, most work on CGI script, not on HTML

Server plugin

- W/API of specific server
- create library of code
- added to server to handle specific requests
- works only with servers for which plugin was created
  - Compare: CGI -- based on published standard
  - Changes require work on program rather than HTML
Applet

- Applet that contains form created
- Connect to server process or directly to db
  - Return results
- Most layout burden on developer
- Significant security considerations
  - E.g., interaction via firewall

Servlet

- Instead of CGI or server plugin
- Use servlet API to dynamically create Web page
- Changing HTML ==> programming process, not page design
Server-side scripting

- Why need server-side scripting? (as compared to previous technologies?)
- What?
  - Program code embedded in HTML doc
  - Parsed by Web server
  - ==> More flexibility in layout of HTML doc’s during Web app dev

Server-side scripting -- types

- Active Server Pages (ASP) -- Microsoft
- Server-side JavaScript (previously LiveWire) -- Netscape
- JavaServer pages -- Sun
Why need server-side scripting?

- In CGI (or other)
  - Create script on server
  - Waits for request from HTML form
  - Upon request CGI script
  - Process form
  - Access db
  - Output HTML to client

Why need server-side scripting? Cont.

- ==> Script generates HTML based on dynamic content
- ==> Developer of script (programmer) responsible for visual aspects of output page
  - HTML “hard coded” in CGI script
- ==> Graphics layout task assigned to programmer

Not good utilization of resources
In server-side scripting

• Business logic ref’d to from presentation
  • Instead of presentation embedded in app
  • Graphic designers designs HTML page
  • Uses server-side scripting tools to specify values needed to be displayed in specific locations in page

Server-side scripting, cont.

• Tools refer to developer-created business rules
  • Do actual processing of form
  • Access data source
  • Log transactions
  • Return values
Server-side scripting, cont.

- Graphics experts use special tags to put placeholders for values in Web pages
- When page is requested, ...

When page is requested

- Server executes business logic created by developer
- Inserts values dynamically in HTML template create by layout people
- ==> Can change layout aspects (font, color, etc.) without need to have developer recompile / modify code
- ==> Developer focuses on business process
- Graphics person focuses on presentation
Typical server-side scripting model

1. Request scripted page
2. Compile page and execute script
3. Output HTML and results of script as HTML to new HTML file or directly send result to client

- Client
- Web server
- HTML
- Server-side embedded script
- More HTML
- HTML

Common properties of solutions

- Access to resources
- Compilation
- Programming language
- Special HTML pages
Access to resources

- Mechanisms to access resource on server
  - Files
  - Databases
  - JavaBeans (later)

Compilation

- Special HTML pages compiled s.t. more efficient when called several times by user
- Great improvement over CGI script in PERL
  - Interpret every time called
Programming language

- Written in one specific language
  - Java, Visual Basic, JavaScript, ...
- Servers know how to process only one kind of script
  - Based on file extension

Special HTML pages

- HTML page where scripts embedded
- Usually different file extension
  - .js, .asp,
- ==> Server knows files require special processing when requested by client
Active Server Pages (ASP)

- MS Active Platform strategy
- Additional component added to MS Web server
  - Process scripts inside files with extension .asp
  - Files contain supported scripting languages
    - Primarily VBScript...

VBScript

- Scripting extension to HTML
- Based on Visual Basic
- Capable of calling ActiveX components
- MS-centric solution
- Implemented in IIS (Internet Information Server)
ASP, cont.

- Not intended to replace client-side scripting
- Designed to provide “glue” between HTML, client-side scripting (e.g., JavaScript) & business logic on server

ASP application is composed of

- HTML
- Scripting code
- ActiveX components
- Example: hello.asp ...
- Code defines function WriteDate
  - Executed in body of HTML doc
hello.asp

```vbs
<%@ LANGUAGE="VBSCRIPT" %>
<SCRIPT RUNAT=SERVER LANGUAGE=VBSCRIPT>
Sub WriteDate()
    Response.Write("<B>The current date is " & Date & "</B><BR>);
End Sub
</SCRIPT>

• And the HTML page …

<HTML>
<HEAD>
<TITLE>ASP Example</TITLE>
</HEAD>
<BODY>
The next line is an ASP script embedded in the HTML: <BR><BR>
<% Call WriteDate() %>
Response.Write("That was the date.");
</BODY>
</HTML>

• Notes …
ASP notes

- Function is executed in body of HTML doc between special tags `<%` and `%>

- When client requests hello.asp,
  - Server
    - Treats HTML in file as template
    - Interprets script
    - Inserts results in place
    - Sends all to client

ASP notes, cont.

- `.asp` ==> server automatically interprets as server-side app
- `==>` All code designated by scripting tags replaced w/ results
- `<%@` tag designates VBScript script
- `<SCRIPT>` tag designates that script expected to be executed on server
Results on client

```html
<HTML>
<HEAD>
<TITLE>ASP Example</TITLE>
</HEAD>
<BODY>
The next line is an ASP script embedded in the HTML: <BR><BR>
<B>The current date is 2/14/2001</B><BR>
That was the date.

</BODY>
</HTML>
```

ASP advantages, disadvantages

- Integrated w/ MS technologies
- Supports API for adding other scripting language
  - E.g., JScript, PERL
- Only on MS servers
- Doesn’t work w/ standard Java libraries
Server-side JavaScript

• JavaScript originally developed to make client-side HTML scripting possible
• ==> Web page can do some processing'
  • E.g., field validation on client
  • ==> Save extra trip on network

JavaScript notes

• Object model / prog paradigm NOT similar to Java
• Syntax is similar
• JScript -- MS implementation
• ECMAScript -- standard for all derivations of JavaScript support
  • ECMA: European Computer Manufacturers Association
    www.ecma.ch/news/E262-II.htm
Server-Side JavaScript

- Originally: LiveWire
- Process forms, access db, create dynamic HTML on Web server
- Similar to ASP
- Call prog libs that use Netscape API
- Same role as VBScript w/ ASP
- HelloWorld ...

HelloWorld

```
<HTML>
<HEAD>
<TITLE>Server-side JavaScript Example</TITLE>
</HEAD>
<BODY>
<SERVER>
function writeDate()
{
    var theDate = new Date();
    var theDateString = theDate.toLocaleString();
    write("<B>The current date is " & theDateString & "<BR>");
}
</SERVER>
The next line is in Server-side JavaScript embedded in the HTML:
<br><br>
<SERVER>
writeDate();
write("That was the date.");
</SERVER>
</BODY>
</HTML>
```
Notes

• Doc compiled into byte codes
• Interpreted by Server-side JavaScript component in Netscape server
• Tag, syntax diff than ASP / VBScript
• Otherwise same

JavaServer Pages

• Closely duplicate functionality of ASP, Server-side JavaScript
• Java as primary language
• ==> More flexibility / power
• Less learning (for Java programmers)
• Better integration with other Java components
JavaServer Pages

- Using JHTML -- HTML file format for JSP
- JHTML and JSP

Using JHTML

- Page compilation
  - 1. Create HTML file w/ Java code embedded between <java> HTML tags
  - 2. Save w/ .jhtml extension
  - 3. W/ Java Web Server, when client requests file, server sends file to PageCompileServlet
    - Part of com.sun.server.webserver package
**JHTML, cont.**

- Servlet uses .jhtml file as template
- Generates servlet that outputs HTML in original file
- Executes Java code between `<java>` tags
- 4. From then on, every time client requests this .jhtml file the new servlet is sent the request and responds
- Servlet recompiled automatically on next request after .jhtml file changes

**JHTML notes**

- To try examples on Java Web Server
- Set up server
- Put .jhtml files in public_html directory under JWS directory
  - Access from browser
Example .jhtml file

```html
<HTML>
<HEAD>
<TITLE>Sample1 JHTML File<TITLE>
</HEAD>

<BODY>
Hello
<java>
String name;
if ((name = request.getParameter("name")) != null)
out.print(name);
</java>
</BODY>
</HTML>
```

Example .jhtml, cont.

```html
<!-- Note that for simplicity, the form calls the same file with the value for
the name text field filled in. -->

<FORM METHOD=POST ACTION="sample1.jhtml">
Enter your name and press submit
<INPUT TYPE=TEXT NAME=name>
<BR><BR><INPUT TYPE=SUBMIT>
</FORM>
```
Example .jhtml, cont.

```java
<java>
    for (int i = 1; i < 6; i++)
        out.print("<H"+ i +"> Heading " + i);
</java>

</BODY>
</HTML>

What happens?

- First time sample1.jhtml requested
  PageCompilation servlet called by JWS
- Generates & compiles servlet called _sample1.java
- _sample1.class stored by default in directory called pagecompilation
- Server called from then on whenever sample1.jhtml is requested
- If .jhtml file changes, process repeated
Advantage

• Servlet doesn’t need to be compiled every time it is requested

sample1.java

• code omitted
• code between <java> tags inserted into service method of servlet
• servlet reads .jhtml file into variable _fileData
• writes all HTML not between <java> tags
• wherever <java> tags located, executes Java code
JHTML and JSP

- JHTML: 1st generation for JSP
- 2nd gen: .jsp file
- main design diff: JSP designed to provide scripting capabilities on server and browser (potentially)
- JSP designed to support multiple scripting languages
  - Currently only Java
- Basic syntax of JHTML, JSP similar

JHTML and JS, cont.

- JSP goal: integrate into server-scripting
- ==> looks a lot like ASP
  - But code in Java
Example .jsp file

```html
<HTML>
<HEAD>
<TITLE>Sample5 JSP File</TITLE>
</HEAD>

<BODY>
Hello <%= request.getParameter("name") %> 
</BODY>
```

Example .jsp, cont.

```html
<!--
Note that for simplicity, the form calls the same file with the value for the name text field filled in.
-->
<FORM METHOD=POST ACTION="sample5.jsp">
Enter your name and press submit
<INPUT TYPE=TEXT NAME=name>
<br><br><INPUT TYPE=SUBMIT>
</FORM>
```
Example .jsp, cont.

```jsp
<%
for (int i = 1; i < 6; i++)
    out.print("<H" + i + "> Heading " + i);
%

</BODY>
</HTML>
```

And the servlet generated

- Looks very much like the JHTML version
Summary

- Mechanism to create dynamic Web pages
  - HTML + programming logic
- ASP & Server-side JavaScript
  - Provide basic functionality required for server-side scripting

Summary, cont.

- But
  - JSP represents progress by relying on full-fledged prog lang
- Drawback
  - Longer learning time than VBScript, JavaScript
- Advantage
  - More power of Java
Summary, cont.

- JavaServer Pages to create dynamic Web pages
  - Access server-side business logic & data sources
  - Currently supported only as server-side scripting technique
  - => All processing on server
  - Web browsers can be enhanced to move some processing from server to client
  - Milestone in maintainable & extensible Web app's