An introduction to ASP

Aslan Neishaboori

1

The history of ASP

- ASP -> Active Server Pages
- Microsoft's server-side technology for dynamically-generated web pages that is marketed as a companion to **Internet Information Server (IIS).**

2

ASP from beginning until now

- ASP has gone through four major iterations
  - ASP 1.0 (distributed with IIS 3.0)
  - ASP 2.0 (distributed with IIS 4.0)
  - ASP 3.0 (distributed with IIS 5.0)

  } Classic
  - ASP.NET (part of the Microsoft .NET platform).
  - ASP.NET (1.0 , 1.1 , 2.0…)

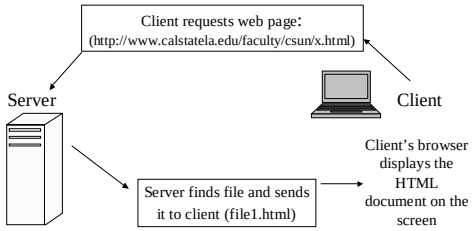
3

Web pages

- Static web pages
- Client Side scripting
- Server side scripting / programming.

4

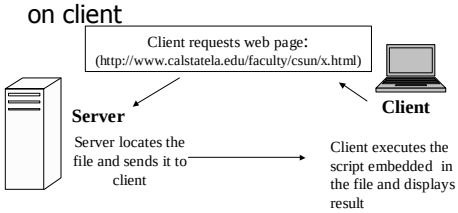
Static Web Pages



5

Client-Side Scripting

In client-side scripting, processing is done on client

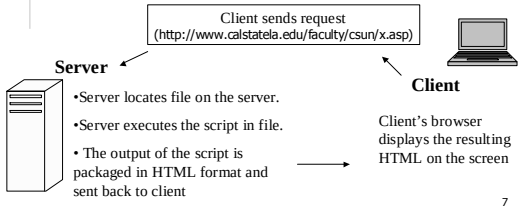


6

Server-side scripting

ASP is an example of server-side scripting

In server side-scripting, processing is done at server rather than at client



7

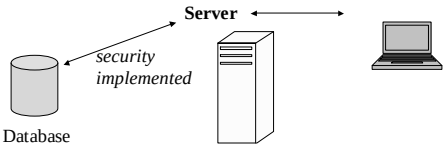
Advantages of Server-Side Scripting

- Control over server side components are available (access, SQL 2000, etc.); so, no client-side plug-ins required
- Compatibility issues minimized (browser does not have to support client-side scripting)
- Bandwidth enhanced (send only the data the client needs)

8

One important Advantage of ASP

- Security enhanced
  - Users never see code
  - server is intermediary between client and data source



9

Some general points about ASP

- Web Pages can be generated by
  - mixing server-side scripting code (including database access)
  - with HTML
  - and client-side script.
- ASP is not a language it is a "server technology"

10

Some general points about ASP

- ASP technology is built directly into Microsoft Web servers, and therefore is supported on all Microsoft Web servers
- ASP runs as a service of the Web server and is optimized for multiple threads and multiple users.

11

Many scripting languages can be used to write ASP

- VBScript (a subset of Microsoft Visual Basic programming language)
- Jscript (Microsoft's implementation of JavaScript)
- PerlScript—similar to Perl
- Python another scripting language used in web development

12

What you need to run ASP

- You must be running a web server
  - Free web server products are available from Microsoft
  - Internet Information Server (IIS) requires NT Server or Win2k, Windows XP professional and above…
  - Personal Web Server (PWS) (stripped-down version of IIS) runs on Windows 95 or better

13

ASP Without IIS or PWS

- Several 3rd-party companies have created software to allow ASP to run on various non-Microsoft servers and platforms
- Examples:
  - Halcyon Software's Instant ASP
  - Chili!soft's Chili!ASP
- These product extend ASP functionality to Non-MS Web Servers such as:
  - Apache, Sun Web Server, Netscape Enterprise Server
- These products run on Non-MS platforms such as:
  - Linux, Sun Solaris, Apple Mac OS, IBM AIX

14

Creating ASP Pages

- File name must have .ASP extension
- Scripts are embedded within the HTML (but executed at Server)
- Start page off with @language directive:

```
<%@ Language=VBScript %>
```

- enclose script commands in "<%" and "%>"

Example:

```
<%Response.Cookies("MyFavMovie")
        =“Godfather"%>
```

15

An ASP example

```
<HTML>
<HEAD>
<TITLE>The Polite Web Server</TITLE>
</HEAD>
<BODY BGCOLOR="wheat">
<H1>Welcome</H1>
<FONT SIZE="3">
<B>This is the Polite Web Server, at <% = Time %> on <% = Date %>
    </B></FONT><BR>
<BR>
<% If Hour(Now) < 9 Then  %>
  Do you know what time it is? I was still in bed!
<% Else
     Randomize
     intChoice = Int(Rnd * 4)
     Select Case intChoice
          Case 0 %> So, where do you want me to go today?
       <% Case 1 %> Well, look who's back visiting us again!
       <% Case 2 %> Hi there, and welcome to our site.
       <% Case 3 %> It's raining here - would you like to play virtual
   checkers?
    <% End Select
  End If %> <BR>
<HR>
</BODY>
</HTML>
```

16

ASP Objects

- ASP includes five standard objects for global use:
  - **Response:** send info to user
  - **Request:** get info from user
  - **Server:** Controls the IIS server (i.e. creates objects and supplies access to methods and properties on the web server)
  - **Session:** stores session information for individual users as they navigate a web site
  - **Application:** stores and shares information for use during an active application

17

The Request object

- The Request object is used to get information from the user that is passed along in an HTTP request
  - Form—to get data from an HTML form
  - Cookies—to get the value of application-defined cookie
  - ServerVariables—to get HTTP information such as the server name

18

Example

- Request.ServerVariables(Remote_Host)
- This example returns the Internet name or IP address of the visiting computer

19

Example

- Assume user has downloaded a form-based page from a site. The user selects an item in an Option drop-down list and hits the submit button.

```
<FORM ACTION
  =http://www.calstatela.edu/faculty/csun
  /grades.asp
METHOD=POST>
```

20

Example (Continued)

- The action calls the ASP page containing the following code:

```
If Request.Form("Answer") = "I agree"
    Response.Write "You may proceed"
Else
    Response.Write "You cannot proceed"
Endif
```

21

The Response Object

- The Response object is used to send information to the user. The Response object supports a number of properties and methods.

22

Some *Properties* supported by Response Object

- ContentType—to set the type of content (i.e: text/HTML, Excel, etc.)
- Expires—sets the expiration (when the data in the user's cache for this Web page is considered invalid) based on minutes (i.e.: expires in 10 minutes).
- ExpiresAbsolute—allows you to set the expiration date to an absolute date and time.

23

Example

```
….
<H1>Welcome to the New Products Seminar</H1>
    <%
      Dim strCity, strDate
      strDate = Request.Form("Date")
      strCity = Request.Form("City")
      Response.Write "Held in "
      Response.Write strCity
      Response.Write " on "
      Response.Write strDate
    %>
…
```

24

The following methods are supported by the Response object:

- **AddHeader**—Adds an HTML header with a specified value
- **AppendToLog**—Appends a string to the end of the Web server log file
- **BinaryWrite**—writes binary data (i.e, Excel spreadsheet data)
- **Clear**—clears any buffered HTML output.
- **End**—stops processing of the script.
- **Flush**--sends all of the information in the buffer.
- **Redirect**—to redirect the user to a different URL
- **Write**—to write into the HTML stream. This can be done by using the construct

```
Response.write("hello")
```

or the shortcut command

```
<%="hello"%>
```

25

Session Object

- Session object uses cookies to maintain state information
- To set session variables

```
Session("chrCity") =
Request("chrCity")
```

26

Working with Database Objects

- Makes use of the **A**ctiveX **D**ata **O**bjects Component (**ADO**)
- Two important ADO objects: Connection and Recordset
  - Connection object is used to make the connection to the database
  - Recordset performs two tasks:
    - instructs database as to what information you're interested in
    - Stores the requested information returned by the database

27

Database Objects in Subscription

```
' Create an ADO database connection
    set dbSubs =
    server.createobject("adodb.connection")
' Open the connection using our ODBC file
DSN
    dbSubs.open("filedsn=SubForm")
' Execute the SQL statement
    dbSubs.execute(sql)
```

28

Database Objects in Subscription

```
` Create an ADO database connection
    set dbSubs = server.createobject("adodb.connection")
    set rsSubs = server.CreateObject("adodb.recordset")
` Open the connection using our ODBC file DSN
    dbSubs.open("filedsn=SubForm")
` Execute an SQL statement
    dbSubs.execute sql
` Execute the statement and retrieve the record set
    set rsSubs = dbSubs.Execute(sql)
```

29

Resources:

- Example codes are from Beginning ASP 3.0 provided at www.wrox.com

30