

Active Server Pages from Microsoft

Nancy McCracken

Northeast Parallel Architectures Center at
Syracuse University

111 College Place, Syracuse, NY 13244

<http://www.npac.syr.edu/>

Abstract

- Active Server Pages allow special tags and script code in some language, typically VBScript, embedded in HTML files. These tags and code are processed by the web server to obtain a dynamically-produced HTML page to the browser.
 - another architecture in the web-based distributed application arsenal.
 - produce dynamic web pages on the server side (as does CGI), but separate application logic from the appearance of the page.
 - the tags allow previously compiled code, in the form of ActiveX components, to be used
 - allows fast development and testing.

Development of ASP

- Microsoft's Active Server Pages (ASP) was released with IIS2.0 in approximately 1997.
- ASP only runs on Microsoft IIS (Internet Information Server) and Personal Web Servers, and O'Reilly's WebSite Pro.
- There has grown up a full suite of software including ASP editors, such as the ones in Microsoft Visual Studio and from Macromedia. There is also a Microsoft script debugger.
- There are lots of web materials and examples, including performance tips.
- ASP is free for WindowsNT and other Microsoft platforms.
- ASP has recently become available for Unix platforms from Chilisoft and Halcyon.

ASP elements

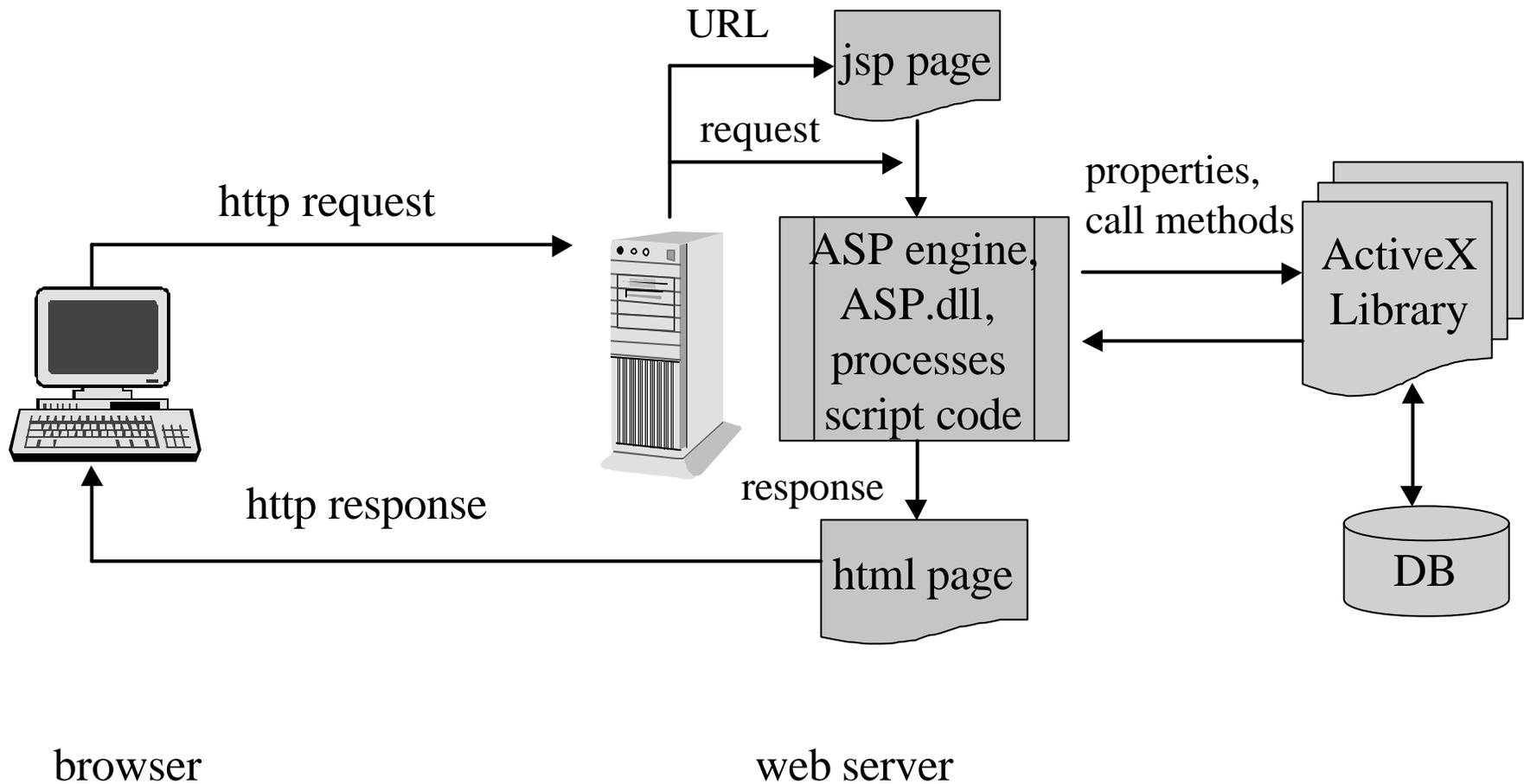
- An ASP page looks like a standard HTML page with additional elements processed by the ASP engine. Typically, these elements create text that is inserted into the resulting document.
- ASP elements
 - Scripts: anything within a script tag `<SCRIPT RUNAT=server>` and `</SCRIPT>`, or more commonly between the `<%` and `%>` markers is interpreted as a script.
 - The script language is usually VBScript, but may also be any language supported by the ActiveX scripting host, which are JScript and PerlScript.
 - Expressions: anything between `<%=` and `%>` markers is evaluated by the ASP engine as an expression in the server environment.

Example ASP page

- This page prints the current time, and either “Good Morning” or “Good Day”, depending on the time. Note the use of the time variable and hour function from VBScript.

```
<html><head>
<TITLE>hi.asp</TITLE>
</head>
<body bgcolor="#FFFFFF">
Today is <%=now%> and all is well<br>
<%if hour(now())>13 THEN%>
Good Morning
<%ELSE%>
Good Day!
<%END IF%>
</body></html>
```

Architecture



ASP Built-in Objects

- The Server object represents the web server. Its most widely used method is CreateObject method, which instantiates any COM object in the library. This includes ActiveX components, both those developed by the user and the ActiveX Data Objects (ADO).
 - ADO is fairly lightweight and allows easy access to any ODBC or OLEDB compliant data source including Microsoft Access (Jet), Microsoft SQL Server, and Oracle databases. In addition, ADO has built-in connection pooling.
- The Session object maintains a set of variables on the server, and provides a way to share data among different pages used by the same client. For example, this script reads sets a value to a session variable named uid:

```
<% Session("uid")= "somevalue" %>
```

- session data is also kept in the global.asa file
- Note that the session object maintains its state by cookies, unless you use the Cookie Munger filter on NT.

The Remaining Objects

- Two more objects, Request and Response, have methods and variables to represent the request information from the browser and generate the response back to the browser.
 - If Firstname is the name of a field on a form from the browser:
`request.querystring("Firstname")`
 - Obtain information from the request environment variables:
`request.servervariables("HTTP_HOST")`
 - Use the method `response.write` to output the html document back to the browser, or just insert expression values into the html document.
 - Note that upper and lower case don't matter, so we could also write `Request.QueryString` and `Response.Write`.
- The Application object is similar to the Session object, but can share data among pages used by all the client browsers.
- TheObjectContext object is only useful when IIS interacts with MTS and you write transactional ASP scripts

An ADO Example

- Here is a part of a simple ASP script that creates an ADO Recordset on a database called Pubs:

```
<% Dim rs
    Set rs = Server.CreateObject _
        ("ADODB.Recordset")
    rs.Open "Authors", "DSN=pubs"
%>
```

- After the query, you can use the recordset to write HTML output

```
<% Response.Write rs("au_fname") & " " _
    & rs("au_lname") %>
```

or

```
<%= rs("au_fname") & " " & rs("au_lname") %>
```

More Server Components

- In addition to using ADO for database access, developers can write a component in ActiveX or any language used to write COM server objects to write backend services. These include C++, Visual Basic, and Delphi.
- If ASP is running with the NT Options pack, then there is a component CDONTS that uses Microsoft SMTP server to send email from an ASP Script.
- There are also many third party component software packages to send email, read news, etc. from ASP scripts. One package particularly mentioned is ASPMail.

ASP Resources

- Microsoft: msdn.microsoft.com/workshop/server/asp
- Sites with lots of tutorial materials and examples:
 - www.asp-zone.com
 - www.learnasp.com (tricky dynamic selection menus!), so use www.learnasp.com/learn/index.asp
 - www.tcp-ip.com
 - www.devasp.com
- PerlScript resources:
www.activestate.com/ActivePerl/docs/perlscript.html
- Server products, including ASPMail:
www.serverobjects.com/products.htm
- Other ASP servers: www.chilisoft.com and