Information Management Resource Kit

Module on Management of Electronic Documents

UNIT 6. NETWORKING DOCUMENTS AND DATABASES

LESSON 3. DYNAMIC WEBSITES: ACTIVE SERVER PAGES

NOTE

Please note that this PDF version does not have the interactive features offered through the IMARK courseware such as exercises with feedback, pop-ups, animations etc.

We recommend that you take the lesson using the interactive courseware environment, and use the PDF version for printing the lesson and to use as a reference after you have completed the course.

Objectives

At the end of this lesson, you will be able to:

- understand what **Active Server Pages (ASP)** are; and
- be aware of the main **advantages and disadvantages** of ASP.

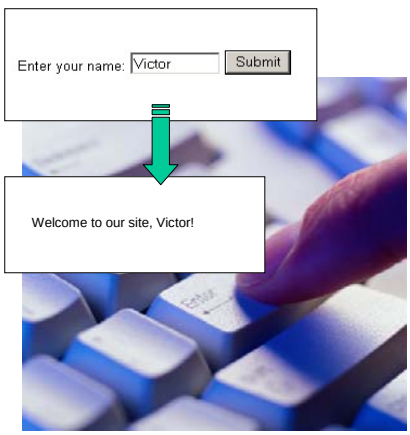
Introduction

Users will have to interact with our website: this is not difficult using ASP!

New forms of scripting allow the limitations of static web pages to be overcome and dynamic web pages to be built easily, without the need for in-depth knowledge of programming.

ASP (Active Server Pages) technology is a powerful tool which offers rapid application development and flexibility.

What is ASP?



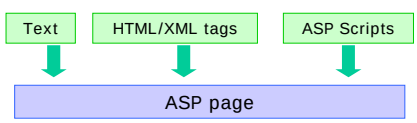
Microsoft's Active Server Pages (ASP) is a **Microsoft proprietary solution** for dynamic website development. Using ASP it is possible to create web pages whose code is processed by the server before it is sent to the client.

These pages can provide information responding to a visitor's different requests. They are **easy to build** and **completely integrated with HTML files**.

How does ASP work?

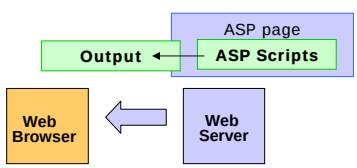
Let's take a closer look at the "inner structure" of ASP pages.

These are specific web pages containing pieces of code written in different scripting languages which will be processed by the server.



An ASP page can contain a mixture of text, HTML/XML tags and scripts.

So an ASP page is similar to an HTML (or XML) web page with calls to ASP functions embedded in it.



Scripts in an ASP page are executed on the server and the resulting page is sent to the client.

Creating an ASP page

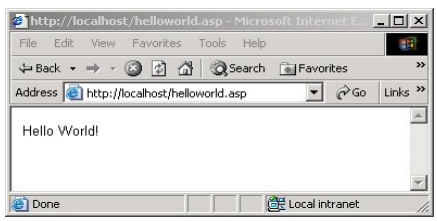
```
<html>

<body>

<% response.write("Hello World!")%>

</body>

</html>
```



This is a simple ASP file. The text in blue is the ASP code, written using Microsoft's **VBScript** scripting language.

As in HTML, there are **delimiters** which define the starting and ending points of a code sequence. These delimiters are expressed by the following symbols:

<% and %>

The **Response** object manages the **output** of the code: in this example, it employs the "write" method to write the text "Hello World!" on the screen.

Creating an ASP page

Scripts in ASP are, by default, written in the VBScript language.

However, scripts can be written in other languages.

ASP ships with native support for two scripting languages (others can be purchased separately):

- **VBScript**, and
- **JavaScript**, a non-Microsoft scripting language.

```
<%@ language="javascript"%>
<html>
<body>
<%Response.Write("Hello World!")%>
</body>
</html>
```

Here you can see the previous **example written in JavaScript**.

The first line declares the language being used.

The resulting HTML page is exactly the same as that created using VBScript.

Creating an ASP page

```
<html>
<head><title>Welcome!</title></
head>
<body>
<center>
<h1>Welcome to our
website!</h1>
You are visitor number
<b><!--#include
file="counter.asp"--></b>
</body>
</html>
```

This ASP page uses the **#include** construct from the Server Side Include (SSI) library, which is a widely used resource in the ASP environment.

<!--#include file="counter.asp"-->

In our example, the server will automatically substitute this code with the contents of the "counter.asp" file which resides in the same folder as the ASP page.

Thanks to the "#include" construct, there is **no need to copy the same code to several pages**. Changes to the "counter.asp" file will automatically be applied to all pages.

Creating an ASP page

An important object used in the ASP code is the **Request** object.

The Request object manages the code **input**; it allows rapid access to the information provided by, or related to, a user.

For example, if the user Ellen Smith types her first name and last name in a form, the following request might be sent:

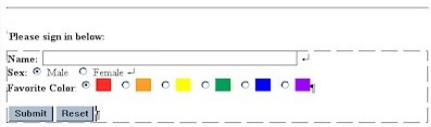
firstname=Ellen&lastname=Smith

Then we can use the information from the form in this script:

Hi, **<%=Request.Form("firstname")%>**!

(where <%= is shorthand for <% response.write)

Creating an ASP page



In your opinion, which of the following objects is used to process data submitted in this form?

- ○ Response
- ○ Request

Please select the answer of your choice

Creating an ASP page

To sum up, in order to create an ASP page you have to:

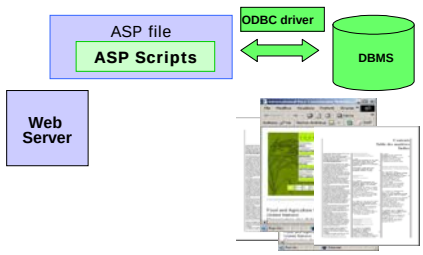
1) open a text editor (Notebook or Word), or an HTML editor;
2) write a page starting with <html><body>and ending with </body></html>;
3) use <%_%> commands and scripts; then
4) save the file as .asp.

Then, you can upload the ASP file on your website.

Our example was a very simple one and it just showed the basic ASP syntax. For further information, you may visit several sites, e.g.:

www.learnasp.com and www.aspalliance.com.

Building a dynamic Website using ASP



ODBC (Open Database Connectivity) is a standard SQL API (Application Programming Interface); it is a software layer between your program (e.g. CGI, Java, etc.) and a relational database.

If you use ODBC, you must have the ODBC driver and a "translator" that communicates in SQL installed on your computer to make the data stored in the database available .

ASP can also be used to **connect to databases using ODBC**.

This allows the developer to create dynamic web pages that are generated using content held in a database.

Through the use of ODBC, ASP can use data from a wide range of databases, for example: SQL Server, Oracle, MySQL, and Access.

Let's look at the stages of building a simple document management website, using a Microsoft Internet Information Server – IIS - as web server and Microsoft SQL Server as database.

Building a dynamic website using ASP

First, we need to **create a database**, e.g. called **documentstore**, with a table document that has all the document details.

In this example, we use SQL to create a table document for **five documents**. The "docid" value (1,2,3,4,5) is the primary key identifying each document.

Then, we set up ODBC in order to use it in our program.

Table document (fragment)

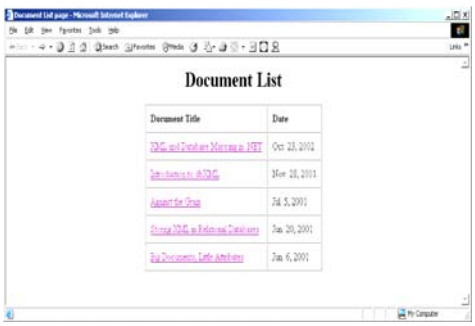
```
INSERT INTO document VALUES (1, 'XML and Database
Mapping in .NET', 'Niel Bornstein', 'Oct. 23, 2002',
'EN', 'PDF', 'D1 - XML and Database Mapping.pdf',
'Continuing his look at .NET`s XML processing from a
Java point of view, Niel Bornstein discovers .NET`s
facilities for binding XML to databases.');


INSERT INTO document VALUES (2, 'Introduction to
dbXML', 'Kimbro Staken', 'Nov. 28, 2001', 'EN', 'PDF',
'D2 - Introduction to dbXML.pdf', 'Following on from
his introduction to native XML databases, Kimbro
Staken introduces the dbXML open source native XML
database');
```

Setting up ODBC

View the entire **Table document**

Building a dynamic website using ASP



View Animation

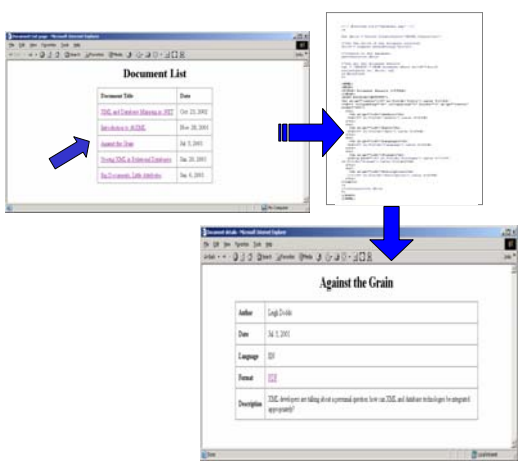
The first page will be similar to the one on the left.

To access, for example, the third document, the user will have to:

- select the **document title**, then
- select the format, "**PDF**", to bring up the document itself.

Now let's look at how this result can be achieved.

Building a dynamic website using ASP



When the user selects the title "Against the Grain" on the document list page, a **details** page **generates the screen** with the details of the relevant document.

We will only need one details page.

Let's look at the ASP pages.

Building a dynamic website using ASP

The first ASP page we will create will make the database calls.

So we create a file called **database.asp**.

Database.asp (fragment)

```
<!-- #include File="adovbs.inc" -->
 <%
 //Constants
DSN = "documentstoreDSN"
DATABASE = "documentstore"
USERNAME = "username"
PASSWORD = "password"

//Get the database connection
sub getConnection(ByRef conn)
conn.Open "DATABASE=" & DATABASE & ";DSN=" &
DSN & ";UID=" & USERNAME & ";Password=" &
PASSWORD
end sub
```

View the entire **database.asp** file

Building a dynamic website using ASP

The file **database.asp** can establish the connection to my DBMS (which is Microsoft SQL Server), execute SQL statements and disconnect.

In this fragment, you can see the part of the script where a connection to the database is specified.

More information about database connection

Database.asp (fragment)

```
 //Get the database connection
sub getConnection(ByRef conn)
conn.Open "DATABASE=" & DATABASE & ";DSN=" &
DSN & ";UID=" & USERNAME & ";Password=" &
PASSWORD
```

This line opens the connection to the database

Building a dynamic website using ASP

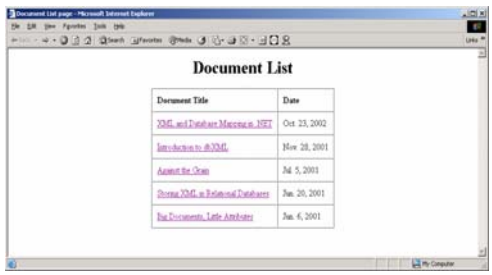
Now we are ready to create the first page that will contain the document list.

We call this page: **DynamicDocumentList.asp**.

We **include the database.asp page**, using the construct:

```
<!-- #include FILE="database.asp" -->
```

Thus the **DynamicDocumentList.asp** can dynamically get the document list from the database.



View the **DynamicDocumentList.asp** file

Building a dynamic website using ASP

DynamicDocumentList.asp (fragment)

```
 <tr>
    <td><a href="details.asp?docid=<%=
rs.Fields("docid").value %>"><%=
rs.Fields("title").value %></a></td>
    <td><%= rs.Fields("date").value
%></td>
  </tr>
```

There is a hypertext link to details.asp including the **docid** as a parameter.

This means that we only need one details page.

The docid will tell the **details.asp** page which document was selected.

Building a dynamic website using ASP

The **details.asp** page will then **get all the details** for each document from the database.

The Request.QueryString retrieves data appended to the query string in the URL and relating to the selected document.

Adding a new document is now done by simply inserting a new row into the document table in the database; no change to the ASP pages is needed.

Details.asp (fragment)

```
<!-- #include file="database.asp" -->
<%

Set dbcon =
Server.Createobject("ADODB.Connection")

//Get the docid of the document selected
docid = request.QueryString("docid")
```

View the entire **details.asp** file

Advantages and Disadvantages of ASP

ASP is a proprietary Microsoft technology, so most of the advantages of using it come only when you are already using a Microsoft platform (Windows operating system and Microsoft Internet Information Server – IIS - as the web server).

The **main advantages** of ASP are:

Click on each feature to read the explanation

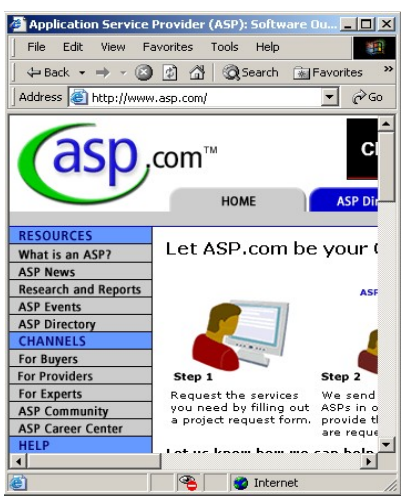
- Windows-based
- Tightly Integrated Development Tools
- Tight Integration with other Microsoft technologies
- Wide Developer Support

ASP can be used on the full range of Microsoft **server** operating systems.
It can also be used on Windows 2000 Professional and Windows XP Professional. In the case of Windows 95 you can use Microsoft's Personal web server.
ASP can also be used on some UNIX/Linux based systems and with web servers other than IIS, using third party ASP engines (e.g.: www.chilisoft.net).

Advantages and Disadvantages of ASP

Tightly Integrated Development Tools	Microsoft's Visual Studio is a highly sophisticated development tool that provides a high level of support for developing and testing ASP applications.
Tight Integration with other Microsoft technologies	ASP can be integrated with other Microsoft technologies such as COM, COM+, MSMQ, and web services (using the .NET platform).
Wide Developer Support	Microsoft tools including ASP are widely used. Therefore, there is a wide bank of developers from which resources can be drawn.

Advantages and Disadvantages of ASP



The main disadvantages of ASP stem from the fact that it is proprietary Microsoft technology.

Proprietary

ASP is a Microsoft-owned technology and is subject to the usual improvement/upgrade cycle. Support from alternative vendors and non-Microsoft platforms is very limited.

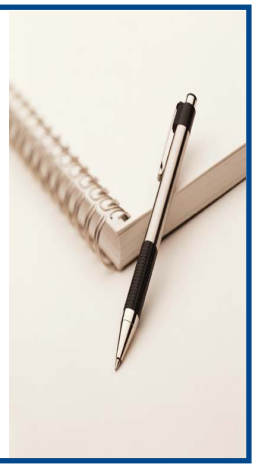
Cost

While ASP is a free component of Microsoft's server operating systems, development tools such as Visual Studio have significant costs associated with them.

Alternative technologies have freely available open source development and deployment environments available.

Summary

- **Microsoft's Active Server Pages** (**ASP**) is a Microsoft proprietary solution for dynamic website development.
- ASP pages are **easy to build** and completely **integrated with HTML** files; they can contain a mixture of text, HTML/XML tags and scripts.
- ASP ships with native support for two scripting languages: **VBScript** and **JavaScript**.
- ASP can also be used to **connect to databases using ODBC**.
- Microsoft tools including ASP are **widely used**, but most of the advantages of using it come only when you are already **using a Microsoft platform**.
- The main disadvantages of ASP stem from the fact that it is **proprietary** Microsoft technology.



Exercises

The following four exercises will allow you to test your understanding of the concepts covered in the lesson.

Good luck!

Exercise 1

Which of the following structures can represent an ASP file?

Please select the answer of your choice

Exercise 2

Can you identify the components of the following simple ASP code?

```
<% response.write("Welcome!")%>
```

response	method
write	output
Welcome!	object

Please pair the items in each columns.

Exercise 3

What is the function of the “include” construct?

- ○ It retrieves information provided by or related to a user.
- ○ It manages the output of the ASP code.
- ○ It avoids the need for duplication of code on several pages.

Please select the answer of your choice

Exercise 4

Which of the following are advantages of ASP technology?

- ☐ Easy to use
- ☐ Open source
- ☐ Portable
- ☐ Integrated with HTML

Please select the answer of your choice

If you want to know more...

- ASP 101 - A database interfacing primer.
- ASP Wire - Active server pages news and information source.
- ASP Developer Net web-based ASP tutorials.
- Fuzzysoftware - The definitive active server resource.
- 4 Guys from Rolla - Offers ASP articles, FAQs, message boards, tips, reviews & more.
- ASP Kicker - Articles designed to help the beginning ASP programmer.
- The Humble ASP FAQ - Frequently asked ASP questions.
- ASP Tutorial http://www.w3schools.com/asp/default.asp
- General ASP resources : http://www.aspin.com
- Microsoft ASP resources : http://msdn.microsoft.com/asp
- Learnasp.com – ASP resources and tutorials