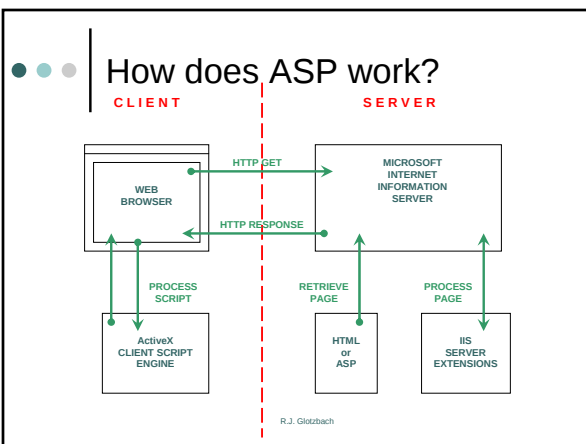


Intro to ASP 3.0

- ## Resources
- o [VBScript Language Reference](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vbscripttoc.asp)
 - <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vbscripttoc.asp>
 - o [HTML & DHTML Reference](http://msdn.microsoft.com/library/default.asp?url=/workshop/author/dhtml/reference/dhtml_reference_entry.asp)
 - http://msdn.microsoft.com/library/default.asp?url=/workshop/author/dhtml/reference/dhtml_reference_entry.asp
 - o [Microsoft Scripting Technologies](http://msdn.microsoft.com/library/default.asp?url=/nhp/default.asp?contentid=28001169)
 - <http://msdn.microsoft.com/library/default.asp?url=/nhp/default.asp?contentid=28001169>
 - o [IE Objects and Events](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dninvbs/html/chapterthreeunderstandingautomationbrowserarchitecture.asp)
 - <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dninvbs/html/chapterthreeunderstandingautomationbrowserarchitecture.asp>
- R.J. Glotzbach

- ## What is ASP?
- o Microsoft's server side technology for creating dynamic content
 - Capabilities:
 - Browser independent
 - Language independent [Vbscript, Javascript, PerlScript]
 - Access databases, install application logic, accept user input via forms, send email, etc.
 - Replacement for CGI programming
- R.J. Glotzbach

- ## Server-Side Includes in ASP
- o Syntax
 - `<!-- #include file="filename.htm" -->`
 - `<!-- #include file="filename.asp" -->`
 - `<!-- #include file="filename.js" -->`
 - Etc.
 - o Purpose
 - Minimize updates to the site
 - Portable
 - Java terminology... "Write once, run anywhere"



- ## General Format
- o Line 1 must declare language to use at Server


```
<%@ Language="VbScript" %>
```
 - o Code segments are encapsulated using


```
<% %> i.e.:
<%
Response.Write("Hello World")
%>
```
- R.J. Glotzbach

General Format

- Plain HTML can not appear inside `<% %>` alone, but can be alternated


```
<% Response.Write ("<h1> Hello </h1>") %>
<h1> <% Response.Write("Hello") %> </h1>
```
- Expression Format `<%= %>`
 Same as `Response.Write`
 Ex: `<%= Date() %> > eq. <% Response.Write(Date()) %>`
 Ex: `<h1> Today is:<%= date() %> </h1>`

R.J. Glotzbach

Example: Hello World

- Display Server Time
- Print Hello World increasing size
 - Expression Format
 - expression format
 - [helloworld_expression.asp](http://www.tech.purdue.edu/cgi/courses/cg356/Private/Examples/Lecture03Examples/HelloWorld/helloworld_expression.asp)
 - http://www.tech.purdue.edu/cgi/courses/cg356/Private/Examples/Lecture03Examples/HelloWorld/helloworld_expression.asp
 - Encapsulated HTML
 - encapsulated format
 - [helloworld_encapsulated.asp](http://www.tech.purdue.edu/cgi/courses/cg356/Private/Examples/Lecture03Examples/HelloWorld/helloworld_encapsulated.asp)
 - http://www.tech.purdue.edu/cgi/courses/cg356/Private/Examples/Lecture03Examples/HelloWorld/helloworld_encapsulated.asp
 - How about using Jscript
 - [helloworld_j.asp](http://www.tech.purdue.edu/cgi/courses/cg356/Private/Examples/Lecture03Examples/HelloWorld/helloworld_j.asp)
 - http://www.tech.purdue.edu/cgi/courses/cg356/Private/Examples/Lecture03Examples/HelloWorld/helloworld_j.asp

R.J. Glotzbach

More General Concepts

- IF statements do not need to be contiguous
 - If counter=7 Then


```
%>
<a href="http://www.purdue.edu">Purdue</a>
Place a page of text or more html
elements in here... as much as needed
<%
End If
```
- Client never sees the code on the server
 - Mix and Match of languages
 - Security

R.J. Glotzbach

ASP Objects

- Response Object
 - Send information to the client
- Request Object
 - Request information from the client
- Server Object
 - Manages server objects (ADO, CDO, etc.)
- Application Object
 - Allows users to share information
- Session Object
 - Allows data to persist beyond the current page
- ObjectContext Object
 - Manipulates Transactions

R.J. Glotzbach

Response Object

- Controls
 - Data sent to the user in the HTTP header and body
 - When the data is sent
- Useful Methods
 - `Response.Write` – writes body content
 - `Response.Cookies` – sets cookie on the client
 - `Response.Buffer (Response.End)` – buffer and send
 - `Response.Redirect` – Sends the user to another URL (header)

R.J. Glotzbach

Request Object

- Controls
 - Data sent to the SERVER in the HTTP header and body
 - When the data is sent
- Useful Methods
 - `Request.QueryString` – retrieves content sent with Get
 - `Request.Cookies` – retrieves cookie on the client
 - `Request.Form` – retrieves content sent with Post
 - `Request.ServerVariables` – retrieves server info.

R.J. Glotzbach

Response Object

Logging Into a system

- Hard coded server side security
 - Login.asp (form)
 - Login_Action.asp (form_handler)
- Examine Get vs. Put
- Add cookies to recognize user
 - Session Object

R.J. Glotzbach

Design Form

- Use HTML to design the form
- [login1.htm](#)
 - <http://www.tech.purdue.edu/cg/courses/cgt356/Private/Examples/Lecture03Examples/Login1/login1.htm>
 - Construct as “normal”
 - Get method is the default <watch out>
 - Set the Action to the form handler
 - Name of each field is used to access on “the other side”

R.J. Glotzbach

Design Login_Action Page

- login_action.asp
 - **Request Object** retrieves FORM information using the assigned name
 - Get is encoded into Query String
 - Retrieved using Request.QueryString(“fieldname”)
 - Post is sent in the header
 - Retrieved using Request.Form(“fieldname”)

R.J. Glotzbach

Get vs. Post

- | | |
|---|---|
| <ul style="list-style-type: none">• Get<ul style="list-style-type: none">• Encoded into URL<ul style="list-style-type: none">• Visible to users• Not good for passwords• Limited by size• Retrieval<ul style="list-style-type: none">• Request.QueryString(“Name”)• Request(“Name”) | <ul style="list-style-type: none">• Post<ul style="list-style-type: none">• Encoded into page header<ul style="list-style-type: none">• Not visible to users• Not limited by size• Retrieval:<ul style="list-style-type: none">• Request.Form(“Name”)• Request(“Name”) |
|---|---|

R.J. Glotzbach

Design Form (again)

- Use HTML to design the form
- [login2.htm](#)
 - <http://www.tech.purdue.edu/cg/courses/cgt356/Private/Examples/Lecture03Examples/Login2/login2.htm>
 - Method now equals Post
 - Keep Action set to the form handler
 - Name of each field used on “the other side”

R.J. Glotzbach

Redirecting the user

- Alter login_action.asp
 - Verify user has the proper login information
 - Use Response.Redirect
 - [login3.htm](#)
 - <http://www.tech.purdue.edu/cg/courses/cgt356/Private/Examples/Lecture03Examples/Login3/login3.htm>

R.J. Glotzbach

Cookies?

- Explain Cookies:
 - Small text chunks left on the user's machine
- Restrictions
 - Size at most 4k
 - Only the domain that set it can read it
 - Unless explicitly set, die when user closes browser
 - Explicitly deleted by setting expiration date to "some time in the past"

R.J. Glotzbach

Session Object

- Session = One user's continuous visit to the site
 - By default, session dies at 20 minutes of inactivity
- Events
 - On Start Session, On End Session
 - Can only be trapped at the Global.ASA file
 - Only be executed for "Web applications"
- Variables
 - "Global Variables" across the user's entire visit
 - Relies on the use of cookies (behind the scenes).

R.J. Glotzbach

Modification to Login.ASP

- If user has visited the site, their name should be filled into the login form
- If user has been away for more than 20 minutes, they have to re-log in.
- If user has not logged in, they are redirected to the login page.
 - [login4.asp](#)

R.J. Glotzbach

Session Object

- Use the Session object to store data that needs to be recalled on later pages
- Use Dimension (Dim) variables on data that is ONLY used on that one page that has the variable

R.J. Glotzbach

Declaring

- Session("UserName") = Request.Form("login")
 - This assigns the value from the form input field to the Session object named UserName
 - Later, on the same page, or on another page, to recall what was entered in the input field, use:
 - Session("UserName")
 - Example:
 - Response.Write(Session("UserName"))
 - Will write the value to the browser window
 - "UserName" can be anything you want... it is a variable name

R.J. Glotzbach

Timeout

- By default, Session Timeout is 20 minutes
- To set Session Timeout:
 - Session.Timeout = 15
 - Sets timeout to 15 minutes
 - Session.Timeout = 5 (admin pages)
 - Sets timeout to 5 minutes
 - Session.Timeout = -1
 - Session should expire immediately

R.J. Glotzbach



Timeout (cont.)

- When does timeout reset?
 - Anytime there is action that requires the server.
 - Click a link, timeout is reset
 - Scroll a page? No
 - Click the back button? ...Maybe, depends
 - Basically, any type of navigation that requests another page from the server. That tells the server to reset the timeout period for the current user.



SessionID

- SessionID is a property of the Session object
- Session.SessionID cannot be changed
 - The web server assigns a SessionID to each unique user
 - Only works when cookies are enabled
 - Abandon usually does not erase SessionID
 - See it: Response.Write(Session.SessionID)

R.J. Glotzbach



Catching Session Timeout

- Cannot use SessionID
 - Even if session times out, or if it is abandoned, SessionID remains the same
- Must catch a variable, typically their login name
 - Session("Login") or Session("UserName")
 - Depending on what you chose to call it...

R.J. Glotzbach



Catching Session Timeout (cont.)

- Login Pass-Thru page
 - ...
'if login & password are correct
Session("Login") = Request.Form("Login")
Response.Redirect("welcome.asp")
...
 - Session("Login") now equals something

R.J. Glotzbach



Catching Session Timeout (cont.)

- Top of welcome.asp
 - ...
If Session("Login") = "" Then
 Response.Redirect("error.asp")
End If
...
 - Catches Session("Login") in case the session has expired or been abandoned.

R.J. Glotzbach



Error.asp

- Redirecting a user to error.asp
 - Whenever you feel a user is trying to hack into your system
 - If a user types in the URL of an admin page instead of navigating to it
 - If a user's session times out
 - If a user logs out, then tries to go back to their pages
 - Etc.

R.J. Glotzbach

Error.asp (cont.)

- Typical contents:

- ...

```
'Clear Session variables
```

```
Session("Login") = ""
```

```
Session("foo") = ""
```

```
'End Session
```

```
Session.Abandon()
```

```
'Then your typical error statements to the user
```

R.J. Glotzbach

Logout.asp

- The contents of error.asp would also be very close to the contents of logout.asp

- If a user logs out

- Want to erase all Session variables
- Want to abandon the Session

R.J. Glotzbach

Contents

- Want to see all of the contents of the Session object?

- Session.Contents

-

```
For Each Item In Session.Contents  
    Response.Write(Item)
```

```
...
```

```
Next
```

R.J. Glotzbach