ASP Tutorial

Application Handling
Part I: 3/15/02

Agenda **Managing User Sessions and Applications**

Section I **Groundwork for Web applications**
Topics: Asp objects, IIS, global.asa

Section II **Application Objects**
Topics: Methods & Events, Contents, Static Objects

Section III **Sessions Objects**
Topics: Methods, Events, Properties, Contents, Static Objects, Session Problems

Section IV **Cookies**
Topics: Example, Usage

Appendix A **Example global.asa file**

Section I Groundwork for Web Applications

Objects in ASP 3.0

7 built-in objects

Covered already	Covered today	Others
Request	Application	ASPError
Response	Session	ObjectContext
		Server

Internet Information Server (IIS)

IIS or Internet Server Manager (ISM)

IIS 5	my machine
Option pack 4 – IIS 4	ltl6
ISM 3	ltl22, ltl10

Defining a Web application is IIS – all ASP, JSP, XML, HTML… pages that reside under a virtual directory (subfolders included) & set application properties

What's a virtual directory?
What's an alias?
What's a virtual server?
Note: multiple virtual servers are only available with the Server version of II5

Uniquely identified by one of (IP, TCP port number, Host header name)
http://ltl6, http://ltl6:7200, http://ltl6:25803

What's a Web Site
What's the difference?
What are the benefits of a virtual server?

global.asa

(see Appendix A)
Must go in the root folder of your web application
Only one global.asa file per application.
Application_OnStart
Application_OnEnd
Session_OnStart
Session_OnEnd
Declaring Objects:
<OBJECT RUNAT=Server SCOPE=Scope ID=Application PROGID="prodgid"></OBJECT> or
<OBJECT RUNAT=Server SCOPE=Scope ID=Application CLASSID="ClassID"></OBJECT>

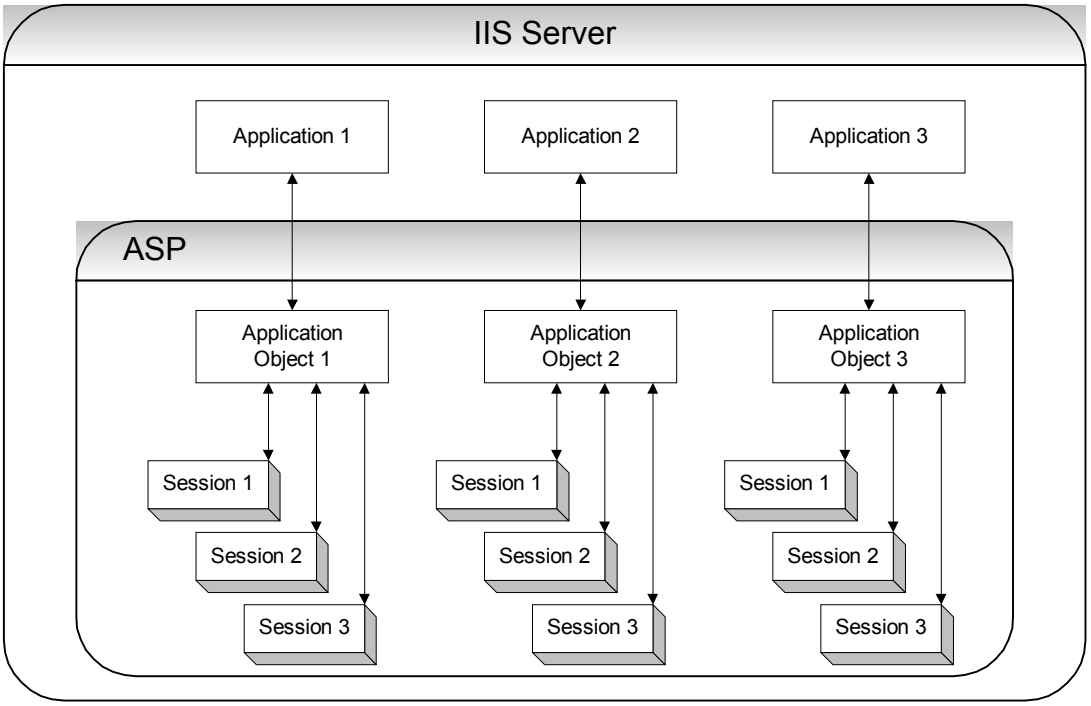
Section II Applications Objects

Methods:	**Events:**
Application.Contents.Remove Application.Contents.RemoveAll Application.Lock Application.Unlock	Application_OnEnd Application_OnStart

HTTP by nature is a client driven protocol. A stateless environment

All users share one application object. (Application-Scope Variable)

Application("Name") = "VISCApps" → Application("Name")



Good use is with storing DB connection information

Application level variables are actually elements of the Application object

The Application object has 2 Collections:

1. **Contents**

all variables added by scripts commands
Application.Contents (“myAppVariable”) or Application (“myAppVariable”)

```
For each Key in Application.Contents
    …some script code…
Next
```

2. **Static Objects**

All objects added with the <object> tag in the global.asa in the application scope

Application Object Methods

Methods explained:
Remove – removes a variable

```
Application.Contents.Remove(“myAppVariable”)
```

RemoveAll – removes all the variables for the Application Object

```
Application.Contents.RemoveAll
```

Lock – only the one currently accessing the variable can make changes
Unlock – frees a previously lock variable

Section III Sessions Objects

Methods:	**Events:**
Abandon Remove RemoveAll	Session_OnStartt Session_OnEnd

How to deal with dynamic pages so we can make more than one request?

Session Objects - Allows the sever/developer to track a user from page to page in an application.

Example: finance.yahoo.com

The user session starts when the user first hits any of the pages in the application

Allows us to treat a user's interaction with the web site with in a specified period of time, as a set of saved variables, rather than just a disconnected series of page request

Classic example: Shopping basket
Authentication tag

The Session object has 2 Collections:

1. **Contents**

Session.Contects("key") or Session("key")
Session("FirstName") = "Mary" → Session("FirstName")

```
For Each item in Session.Contents
        If IsObject(Session.Contents(item)) Then
                Response.Write("Item " & item & " is an object, can't display")
        Else
                If IsArray(Session.Contents(item)) Then
                        Response.Write "Array: " &  Session.Contents(item)
                        For each objArray in Session.Contents(item)
                                Response.Write Session.Contents(item)(objArray) &  "<BR>"
                        Next
                Else
                        Response.Write("Item " & " : " & Session.Contents(item) & "<BR>")
        End If
    Next
```

2. **StaticObjects**

All objects added with the <object> tag in the global.asa in the session scope

Session Object Properties

SessionID – session ID number for each user
    Session.SessionID
Timeout – sets the Session timeout property directly (in minutes)
    Session.Timeout = 30

Session Object Methods

Abandon – Destroys the user session
    Session.Abandon
Remove
RemoveAll

Problem with ending Sessions: Fires when the timeout fires not when the user leaves the site or closes the browser. How long should the server wait while storing all the users information and using up server resources?

Section IV Cookies

Example: www.msnbc.com

Cookies – acts as an identifier so the sever knows which set of session information on the server is associated with which request. (John Peterson, www.DevGuru.com)

Text files written by the client's browser containing info send by the web server, on the clients computer.

Using Cookies in ASP

Creating cookies: Response.Cookies("cookiename") = value
HTML output Set-Cookie:YOURCOOKIENAME=somevalue

Retrieving cookies: Request.Cookies("cookiename") ("keyname").attribute

Display cookie contents: Response.Write Request.Cookies("cookiename")

Set key value: Response.Cookies("cookiename")("keyname") = value

Set Expiration Date: Response.Cookies("cookiename").Expires = "July 4, 2002"
Response.Cookies("cookiename").Expires = Date + 1

Deleting cookies: Response.Cookies("cookiename").Expires = Date – 1

See Cookie examples: login.asp & Checklogin.asp
login2.asp & Checklogin2.asp

Appendix A

----------Global.asa----------

```
<Script Language="VBScript" RUNAT=Server>

Sub Application_OnEnd()
End Sub

Sub Application_OnStart()
    Application("NumSession") = 0
    Application("NumVisited") = 0
    Session.Timeout = 10
End Sub

Sub Session_OnEnd()
    Application("NumSession") = Application("NumSession") - 1
End Sub

Sub Session_OnStart()
    Application("NumSession") = Application("NumSession") + 1
    Application("NumVisited") = Application("NumVisited") + 1
End Sub

</Script>
```

----------home.asp----------

```
<HTML>
<BODY>

Response.Write "You are " & Application("NumSession") & " of " & Application("NumVisited") & "
users."

</BODY>
</HTML>
```

----------output----------

You are 2 of 6 users.