Migrating from Classic ASP to ASP.NET

Smith Suksmith

Office of Computer Clustering Promotion Director &
Microsoft Most Valuable Professional

Classic ASP vs ASP.NET

- ASP.NET offers a number of advantages over the classic ASP script technology, including:

  1) Better development structure by separating the UI presentation from business logic
  2) Code is fully compiled instead of interpreted as in classic ASP; and
  3) Compile feature in conjunction with its caching support means significantly better performance for sites written in ASP.NET over equivalent sites written in classic ASP.

Benefits of Upgrading to ASP.NET

.NET Platform and the ASP.NET development framework deliver a number of features you can benefit from:

New functionality
- XML Web Services
- Web Forms

Fully Integrated Debugging Support
- Tracing

Cleaner Coding Model based on Code Behind concept

Improved Performance and Scalability
- Compiled language support
- Improved caching

Simpler Configuration and Faster Deployment

Intelligent Web Controls

Improved Session State Management

Transfer Session Variables from Classic ASP to ASP.NET

By Peter A. Bromberg, Ph.D.
http://www.eggheadcafe.com/articles/20021207.asp
C# MVP

Share Session State Between Classic ASP and ASP.NET

- Many existing ASP applications are mission critical and complex.
- One approach to address these issues is to run the ASP and ASP.NET side by side, and convert one section of the application at a time to ASP.NET

How to interoperate Session state between classic ASP and ASP.NET.

- Session State transfer between them was the sticky issue
  - MVP's and MS gurus, are all saying "it can't be done...".

```
<TITLE>ASPPage1.asp</TITLE>
<%
' This is the page where we just set some Classic ASP Session Variables
' ASPPage2.asp is where the work is done.
Session("username")="joeblow"
session("email")="joe@blow.com"
Session("userid")=2
Session("DestPage")="Finalpage.aspx"
Server.Transfer("ASPPage2.asp")
%>
```

How to interoperate Session state between classic ASP and ASP.NET

ASPPage1.asp

Session Object

ISSUE

Session state cannot be shared across a mixed ASP/ASP.NET environment using the intrinsic **Session** object

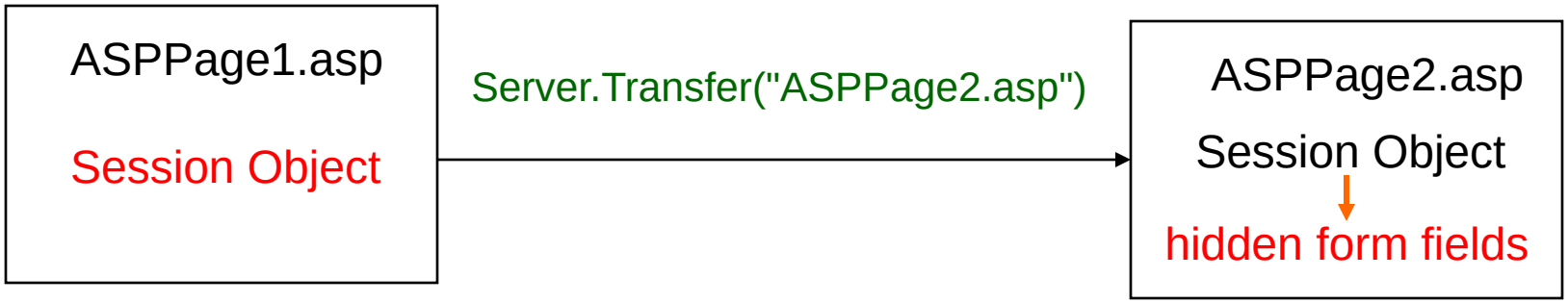
Recommendation

Do not use the intrinsic Session object in your ASP appliction

Alternative methods

- Using cookies
- Using hidden form fields
- Encoding session information in URL strings
- Manually storing and retrieving session information from the database through direct ADO (ASP) and ADO.NET (ASP.NET)
- Using a custom session object that stores state in a database

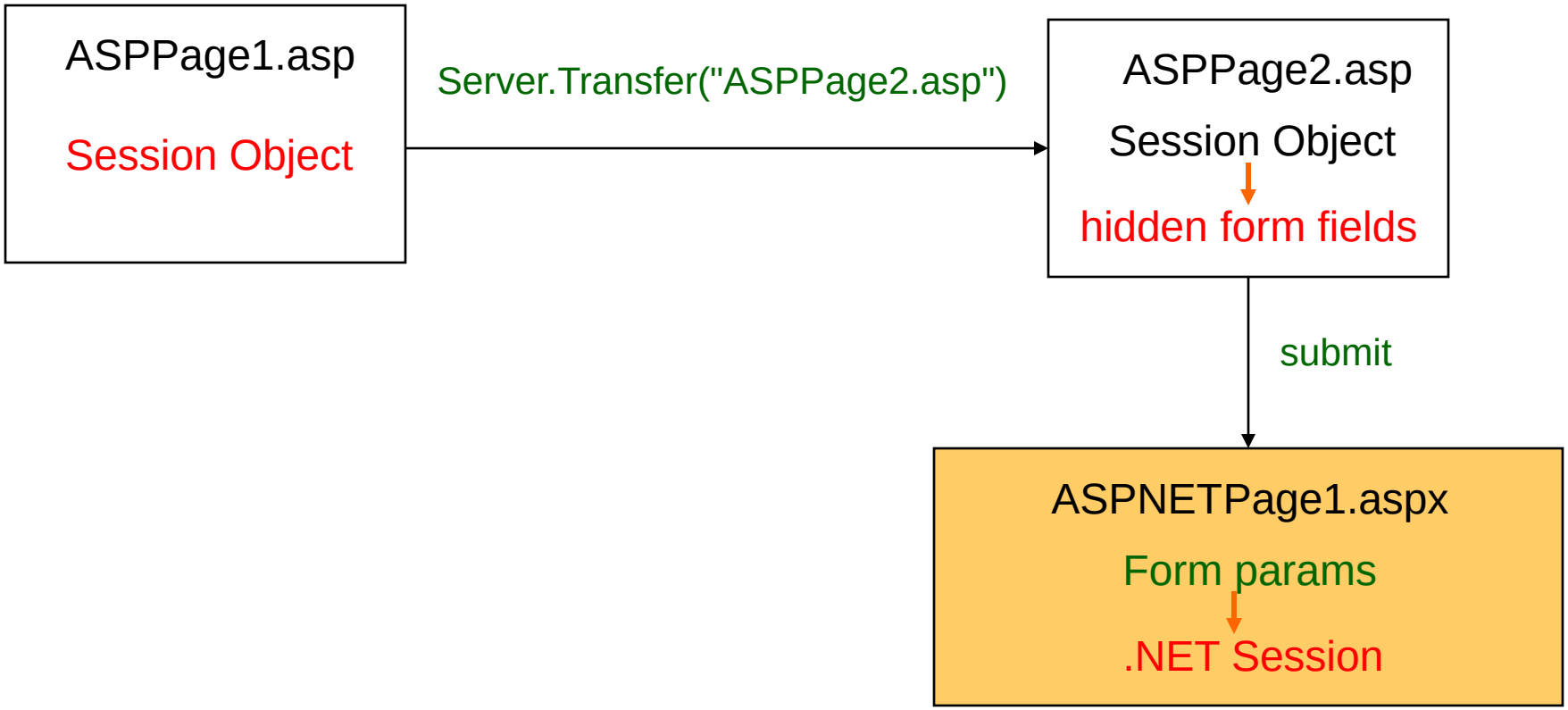
How to interoperate Session state between classic ASP and ASP.NET



How to interoperate Session state between classic ASP and ASP.NET.

```
<TITLE>ASPPage2.asp</TITLE>
<%
' We graf all the session variable names/values and stick them in a form
' and then we submit the form to our receiving ASP.NET page
(ASPNETPage1.aspx)...
Response.Write("<form name=t id=t action=ASPNETPage1.aspx method=post >")
For each Item in Session.Contents
Response.Write("<input type=hidden name=" & Item)
Response.Write( " value=" & Session.Contents(item) & " >")
next
Response.Write("</FORM>")
Response.Write("<script>t.submit();</script>")
%>

================================================================================
<TITLE>ASPNETPage1.aspx</TITLE>
<%@ Page language="c#" %>
<script runat=server>
// We iterate through the Form collection and assign the names and values
// to ASP.NET session variables! We have another Session Variable, "DestPage"
// that tells us where to go after taking care of our business...
private void Page_Load(object sender, System.EventArgs e)
{
for(int i=0;i<Request.Form.Count;i++)
{
Session[Request.Form.GetKey(i)]=Request.Form[i].ToString();
}
Server.Transfer(Session["DestPage"].ToString(),true);
}
</script>
```

How to interoperate Session state between classic ASP and ASP.NET

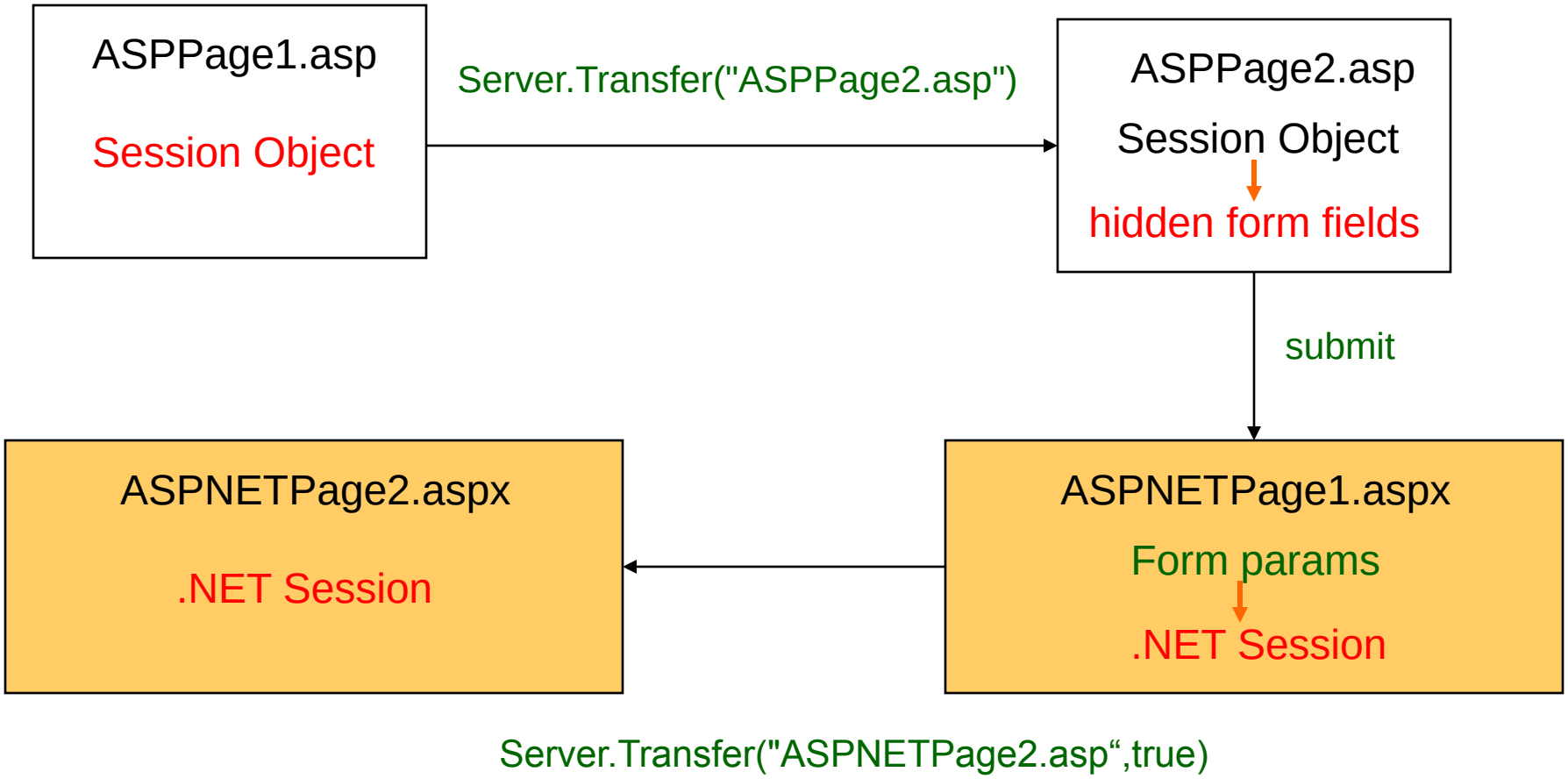


How to interoperate Session state between classic ASP and ASP.NET.

```
<TITLE>FinalPage.aspx</TITLE>
<%@ Page language="c#" %>
<script runat=server>
// This page is just a "proof of concept page"...

private void Page_Load(object sender, System.EventArgs e)
{
Response.Write("Shared Session Variable Names/Values between Classic ASP and
ASP.NET:<BR>");
for (int i = 0; i < Session.Contents.Count; i++)
{
Response.Write("Assigned to \"" +Session.Keys[i].ToString()+"\"");
Response.Write(" Value: "+ Session[i].ToString() +"<BR>");
}
}
</script>
```

1) Contruct a dynamic form in a new Classic ASP page consisting of their names and values as hidden form fields.
2) Submit it to our receiving ASP.NET page where we simply iterate the Form NameValueCollection
3) Sticking the names and values into ASP.NET variables! You want to use Server.
4) Transfer because its much more efficient than making another browser trip with Response.Redirect.

How to interoperate Session state between classic ASP and ASP.NET

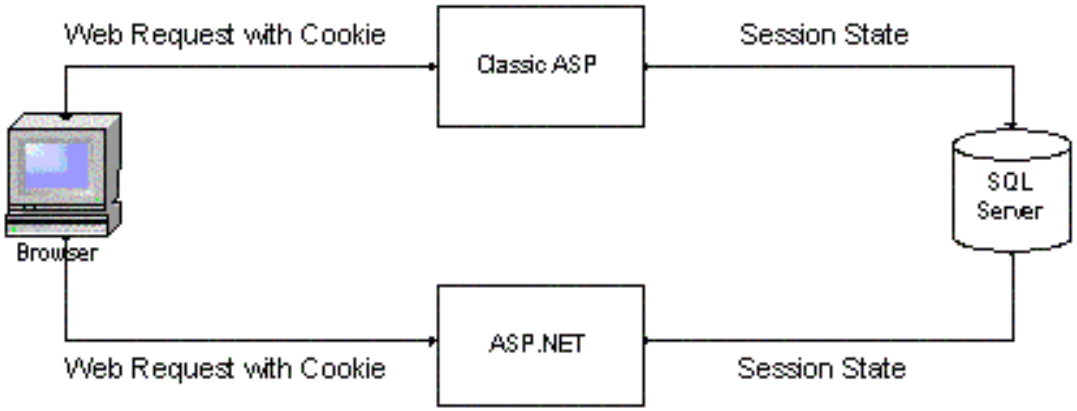


Share Session State Between Classic ASP and ASP.NET

Billy Yuen
Microsoft Corporation

Conceptual Overview

- Cookies are the most common way for Web applications to identify the user session, and can be used to identify session state for both classic ASP and ASP.NET.
- Session state information is stored in memory in ASP script and can't be shared with other applications.



ASP.NET implement custom session class

ASP.NET implementation

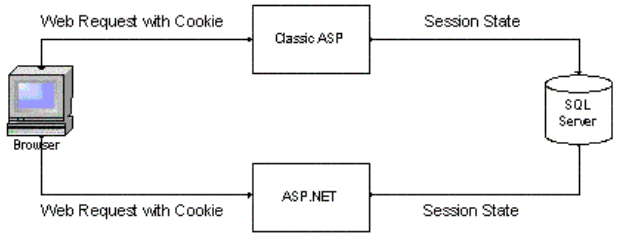
```
public class SessionPage : System.Web.UI.Page
{
   ...
   public new mySession Session = null;
   ...
}
```

```
   [Serializable]
public class mySession
   {
      private HybridDictionary dic = new HybridDictionary();

      public mySession()
      {
      }

      public string this [string name]
      {
         get
         {
            return (string)dic[name.ToLower()];
         }
         set
         {
            dic[name.ToLower()] = value;
         }
      }
   }
```

เก็บ **String**
เพื่อ interoperability กับ **Classic ASP** ได้

ASP.NET implement

```
override protected void OnInit(EventArgs e)
{
   InitializeComponent();
   base.OnInit(e);
}
private void InitializeComponent()
{
   cookie = this.Request.Cookies[sessionPersistence.SessionID];

   if (cookie == null)
   {
      Session = new mySession();
      CreateNewSessionCookie();
      IsNewSession = true;
   }
   else
      Session = sessionPersistence.LoadSession(
Server.UrlDecode(cookie.Value).ToLower().Trim(),
dsn,
SessionExpiration
);
```




```
public  mySession LoadSession(string key, string dsn,
                              int SessionExpiration)
{
   SqlConnection conn = new SqlConnection(dsn);
   SqlCommand LoadCmd = new SqlCommand();
   LoadCmd.CommandText = command;
   LoadCmd.Connection = conn;
   SqlDataReader reader = null;
   mySession Session = null;

   try
   {
      LoadCmd.Parameters.Add("@ID", new Guid(key));
      conn.Open();
      reader = LoadCmd.ExecuteReader();
      if (reader.Read())
      {
         DateTime LastAccessed =
```

ASP Implementation

- The native ASP session can only store session data in memory.
- In order to store the session data to Database, a custom Microsoft® Visual Basic® 6.0 **COM object** is written to manage the session state instead of using the native session object.
  - COM object will be instantiated in the beginning of each Web request and reload the session data from Database Server.
  - When the ASP script is finished, this object will be terminated and the session state will be persisted back to Database.

Sharing Session State Using Database

Issue

- ASP.NET Session object can be stored in Database automatically
- ASP cannot access this session data without extra work and custom code
- Cannot directly instantiate ASP.NET Session Object through COM interoperability and use it to retrieve session state

Recommendation

- Replace Session object with custom implementation for storing session state
- Custom implementation should follow the dictionary pattern mySession(“key”) = “value”

Thank you.

MSDN and Google ☺

http://forums.asp.net/default.aspx?GroupID=12