Advanced Performance Techniques in ASP.NET 2.0

William Zhang, Ph.D.
Senior Consultant
Microsoft Consulting Services

Agenda

- SQL Server cache dependency (SqlCacheDependency)
- Custom cache dependency (CacheDependency)
- Post-cache substitution
- Asynchronous page with parallel-processed tasks
- Data paging via stored procedure
- Returning multiple result sets from DB
- Script callback (out of band call)

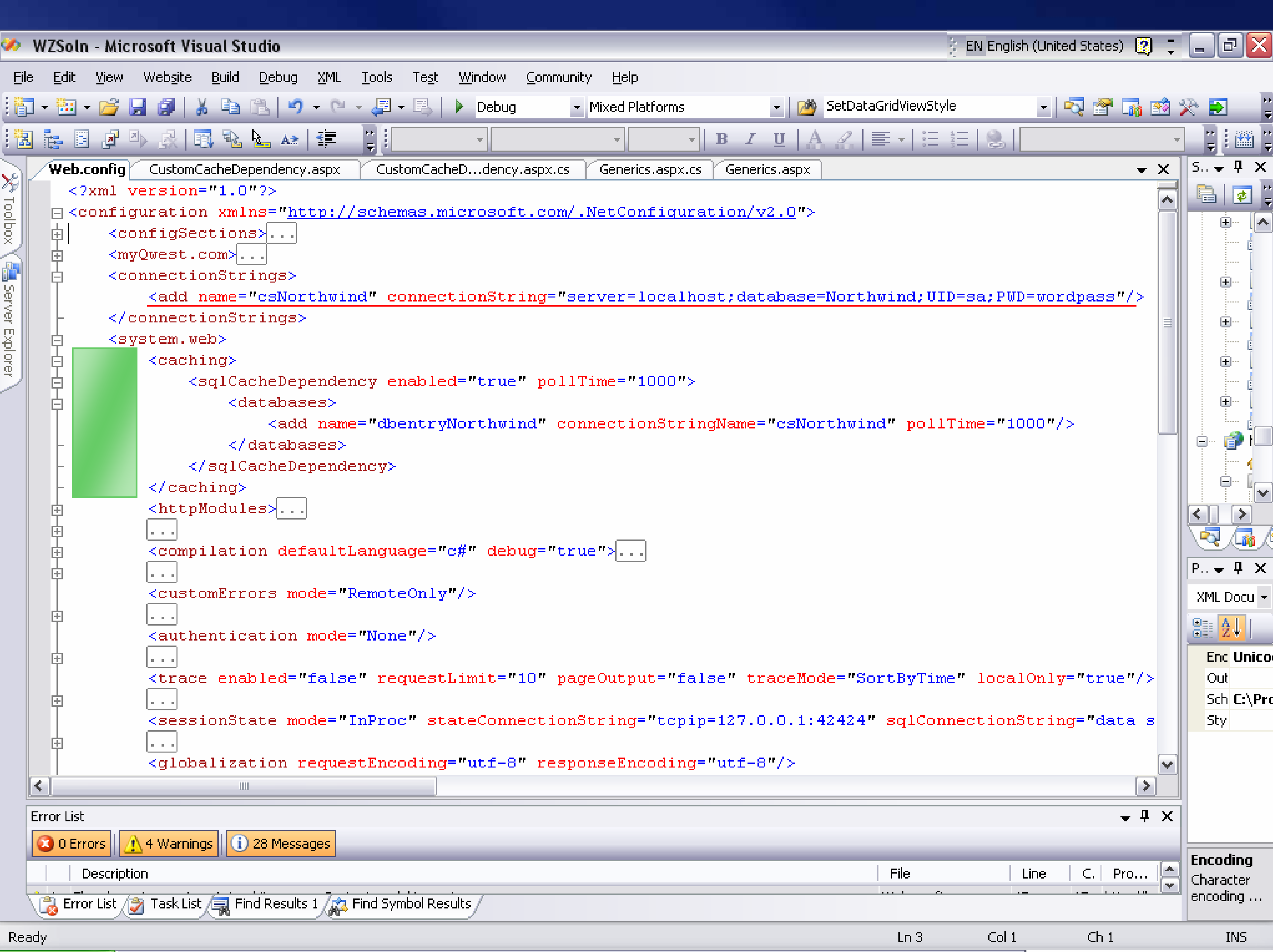
SqlCacheDependency

System.Web.Caching

- SQL 7 & 2000 Support
  - Table change dependencies on SQL 7 & 2000
  - Requires **<cache>** configuration settings
  - One-time setup of SQL Server database
  - Polling model
- SQL Server "Yukon"
  - Result Set dependencies for SQL Yukon
  - Supported through ADO.NET **SqlCommand**
  - No setup required
  - Notification model

SQL Server 7 & 2000

- Table level notifications only
  - Notification when data in table changes
  - Row-level notification <u>is not</u> supported
- Requires one time setup of SQL 7 / 2000
  - Triggers on tables that participate
  - Stored procedures called to check
- Of Note:
  - Entries in cache table < # of tables in DB
  - Entries in cache = # items in cache table



```
<?xml version="1.0"?>
<configuration xmlns="http://schemas.microsoft.com/.NetConfiguration/v2.0">
    <configSections>...
    <myQwest.com>...
    <connectionStrings>
        <add name="csNorthwind" connectionString="server=localhost;database=Northwind;UID=sa;PWD=wordpass"/>
    </connectionStrings>
    <system.web>
        <caching>
            <sqlCacheDependency enabled="true" pollTime="1000">
                <databases>
                    <add name="dbentryNorthwind" connectionStringName="csNorthwind" pollTime="1000"/>
                </databases>
            </sqlCacheDependency>
        </caching>
        <httpModules>...
        ...
        <compilation defaultLanguage="c#" debug="true">...
        ...
        <customErrors mode="RemoteOnly"/>
        ...
        <authentication mode="None"/>
        ...
        <trace enabled="false" requestLimit="10" pageOutput="false" traceMode="SortByTime" localOnly="true"/>
        ...
        <sessionState mode="InProc" stateConnectionString="tcpip=127.0.0.1:42424" sqlConnectionString="data s
        ...
        <globalization requestEncoding="utf-8" responseEncoding="utf-8"/>
```

aspnet_regsqlcache.exe

- Enable database

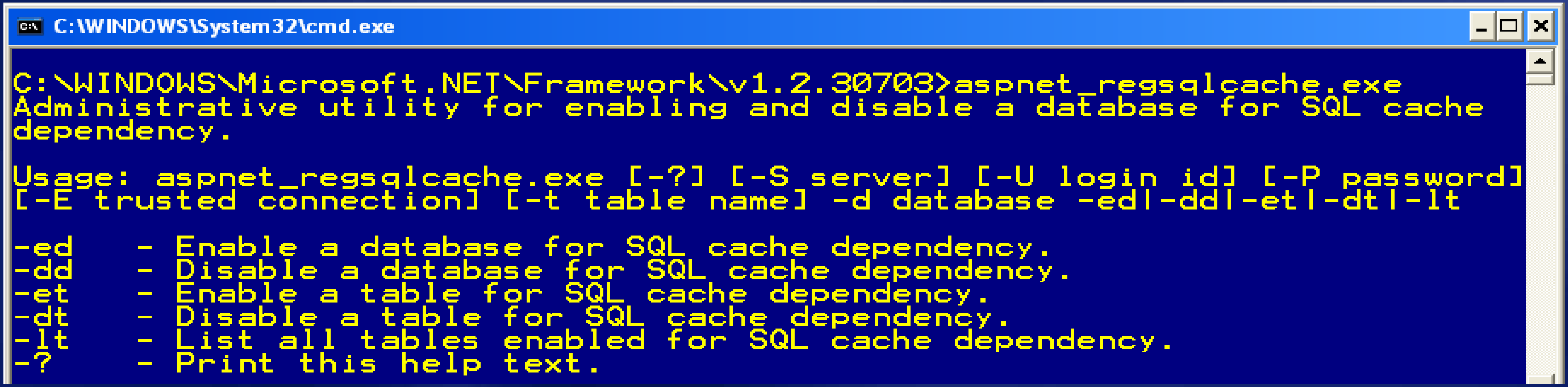
```
aspnet_regsqlcache.exe -S . -E -d Northwind –ed
```

- Enable table

```
aspnet_regsqlcache.exe -S . -E -t Products -d Northwind –et
```

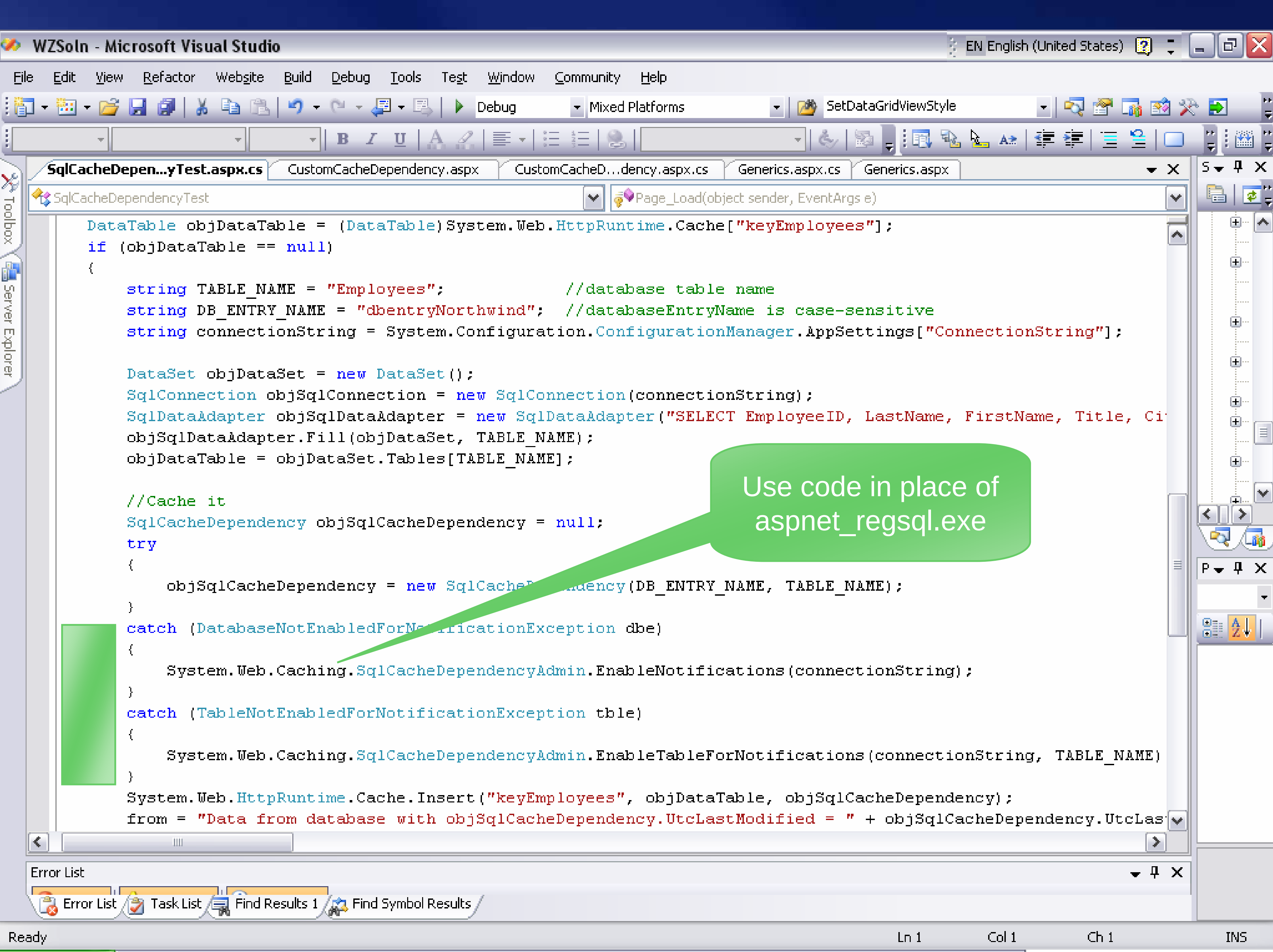
- List enabled tables

```
aspnet_regsqlcache.exe -S . -E -d Northwind -lt
```



C:\WINDOWS\System32\cmd.exe

```
C:\WINDOWS\Microsoft.NET\Framework\v1.2.30703>aspnet_regsqlcache.exe
Administrative utility for enabling and disable a database for SQL cache
dependency.

Usage: aspnet_regsqlcache.exe [-?] [-S server] [-U login id] [-P password]
[-E trusted connection] [-t table name] -d database -ed|-dd|-et|-dt|-lt

-ed    - Enable a database for SQL cache dependency.
-dd    - Disable a database for SQL cache dependency.
-et    - Enable a table for SQL cache dependency.
-dt    - Disable a table for SQL cache dependency.
-lt    - List all tables enabled for SQL cache dependency.
-?     - Print this help text.
```

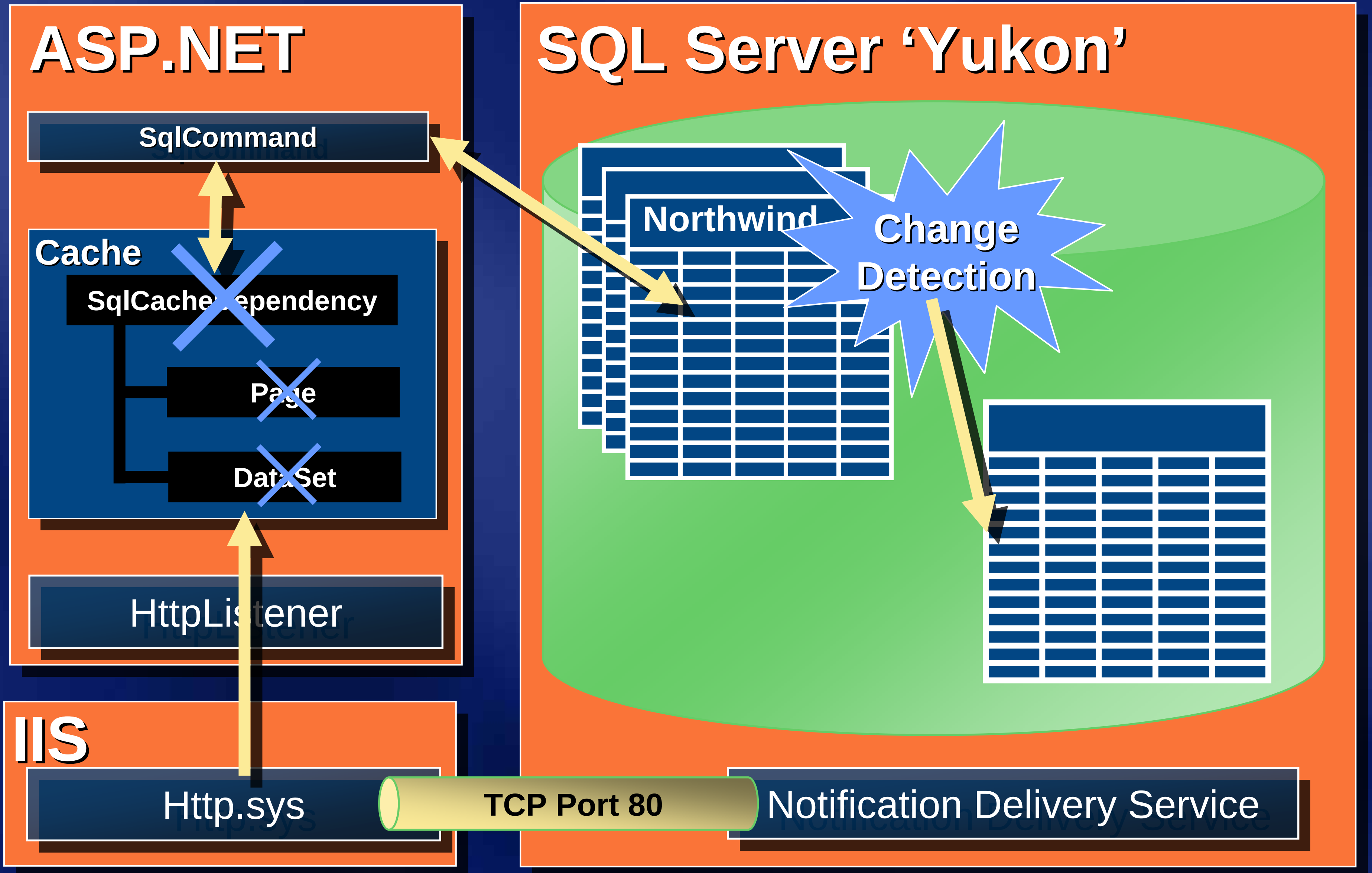


WZSoln - Microsoft Visual Studio
File Edit View Refactor Website Build Debug Tools Test Window Community Help
Debug
Mixed Platforms
SetDataGridViewStyle
SqlCacheDepen...yTest.aspx.cs
CustomCacheDependency.aspx
CustomCacheD...dency.aspx.cs
Generics.aspx.cs
Generics.aspx
SqlCacheDependencyTest
Page_Load(object sender, EventArgs e)
Toolbox
Server Explorer
Error List
Error List
Task List
Find Results 1
Find Symbol Results
Ready
Ln 1
Col 1
Ch 1
INS

```
DataTable objDataTable = (DataTable)System.Web.HttpRuntime.Cache["keyEmployees"];
if (objDataTable == null)
{
    string TABLE_NAME = "Employees";          //database table name
    string DB_ENTRY_NAME = "dbentryNorthwind";  //databaseEntryName is case-sensitive
    string connectionString = System.Configuration.ConfigurationManager.AppSettings["ConnectionString"];

    DataSet objDataSet = new DataSet();
    SqlConnection objSqlConnection = new SqlConnection(connectionString);
    SqlDataAdapter objSqlDataAdapter = new SqlDataAdapter("SELECT EmployeeID, LastName, FirstName, Title, Ci
    objSqlDataAdapter.Fill(objDataSet, TABLE_NAME);
    objDataTable = objDataSet.Tables[TABLE_NAME];

    //Cache it
    SqlCacheDependency objSqlCacheDependency = null;
    try
    {
        objSqlCacheDependency = new SqlCacheDependency(DB_ENTRY_NAME, TABLE_NAME);
    }
    catch (DatabaseNotEnabledForNotificationException dbe)
    {
        System.Web.Caching.SqlCacheDependencyAdmin.EnableNotifications(connectionString);
    }
    catch (TableNotEnabledForNotificationException tble)
    {
        System.Web.Caching.SqlCacheDependencyAdmin.EnableTableForNotifications(connectionString, TABLE_NAME)
    }
    System.Web.HttpRuntime.Cache.Insert("keyEmployees", objDataTable, objSqlCacheDependency);
    from = "Data from database with objSqlCacheDependency.UtcLastModified = " + objSqlCacheDependency.UtcLas
```

Use code in place of aspnet_regsql.exe

How it works: SQL 'Yukon'



Example: Yukon Notifications

```
Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs)

    Response.Write("Page generated: " + DateTime.Now.ToString("r"))

    Dim connection As New SqlConnection(ConfigurationSettings.Connect
    Dim command As New SqlCommand("SELECT * FROM Products", connectio

    Dim sqlDependency As New SqlCacheDependency(command)

    connection.Open()

    GridView1.DataSource = command.ExecuteReader()
    GridView1.DataBind()

    connection.Close()
```

demo

- SQL Server Cache Dependency (SQL Server 2000)
- Source: SqlCacheDependencyTest.aspx for SQL Server 2000

Custom Cache Dependencies

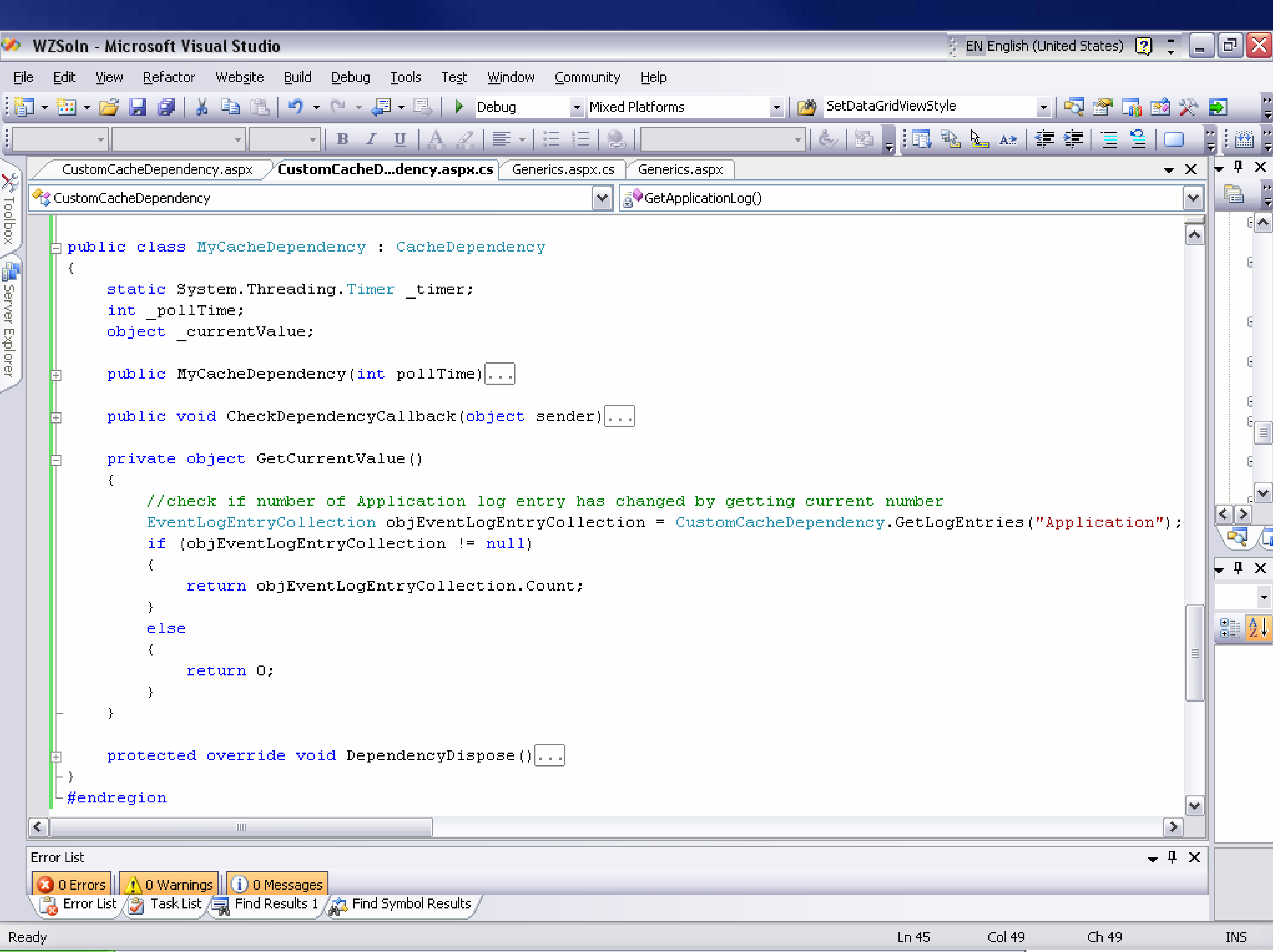
CacheDependency Changes

System.Web.Caching

- No breaking changes to **CacheDependency**
  - Backwards compatible with v1.X code
- ASP.NET 2.0 **CacheDependency** class:
  - New virtual properties/methods
  - Public default constructor
  - Class can be derived, i.e. unsealed

Custom Cache Dependencies

- Anyone can create a dependency
  - WebServiceDependency
  - OracleCacheDependency
- This is just what we did for
  - `SqlCacheDependency`
  - `AggregateDependency`



```
public class MyCacheDependency : CacheDependency
{
    static System.Threading.Timer _timer;
    int _pollTime;
    object _currentValue;

    public MyCacheDependency(int pollTime)...

    public void CheckDependencyCallback(object sender)...

    private object GetCurrentValue()
    {
        //check if number of Application log entry has changed by getting current number
        EventLogEntryCollection objEventLogEntryCollection = CustomCacheDependency.GetLogEntries("Application");
        if (objEventLogEntryCollection != null)
        {
            return objEventLogEntryCollection.Count;
        }
        else
        {
            return 0;
        }
    }

    protected override void DependencyDispose()...
}
#endregion
```

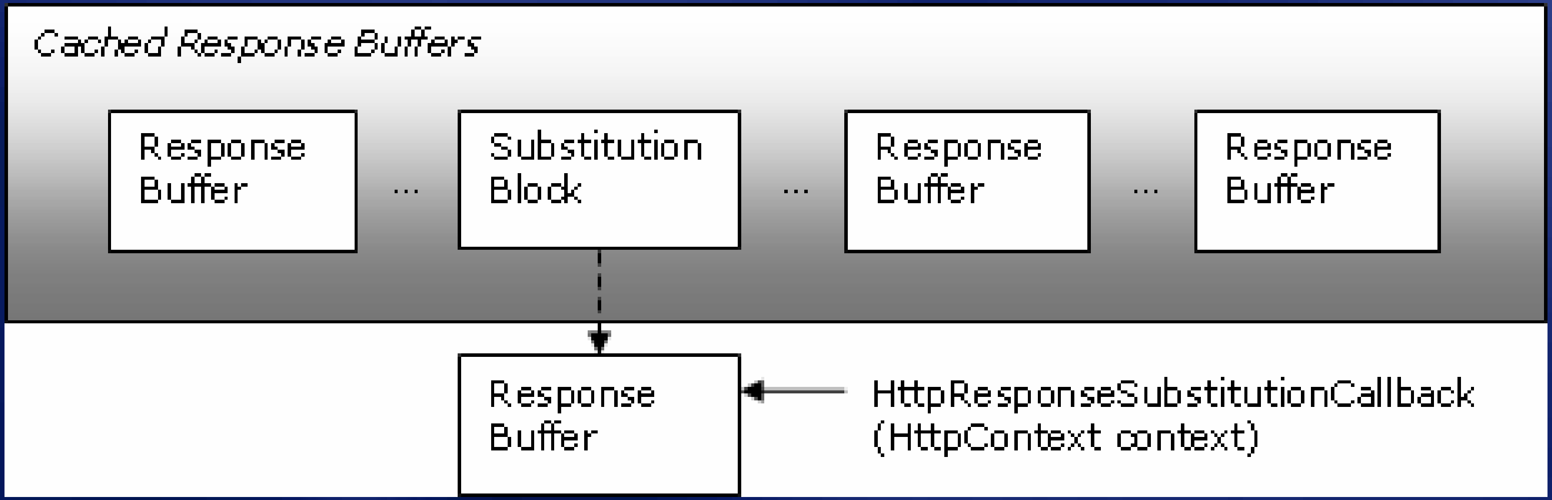
demo

- Custom Cache Dependency (Event Log change invalidates cache)
- Source: CustomCacheDependency.aspx

Post-Cache Substitution

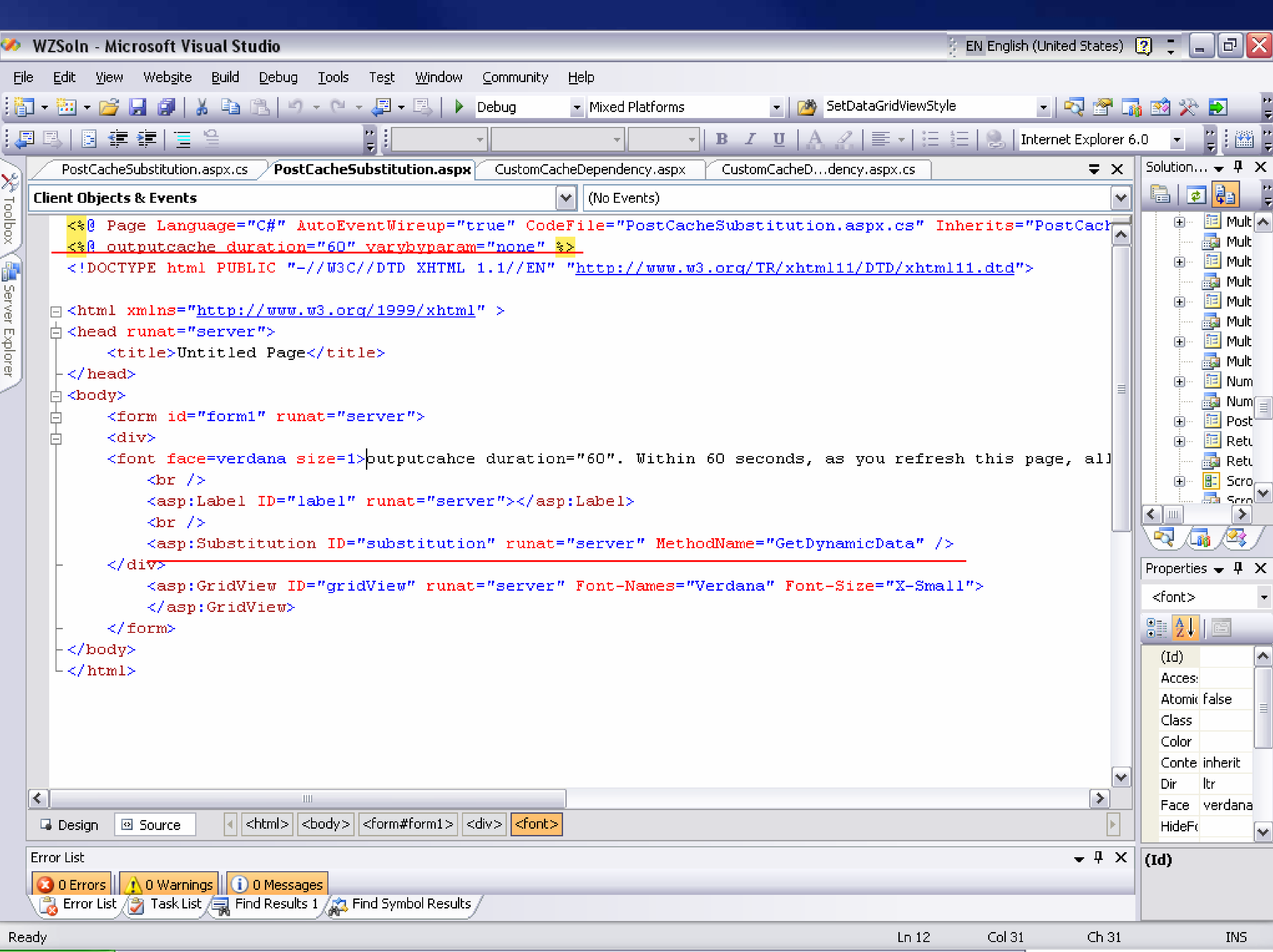
ASP.NET 2.0

- Post-Cache Substitution
  - Output cache entire page
  - Identify regions that are dynamic
- Uses a PlaceHolder buffer



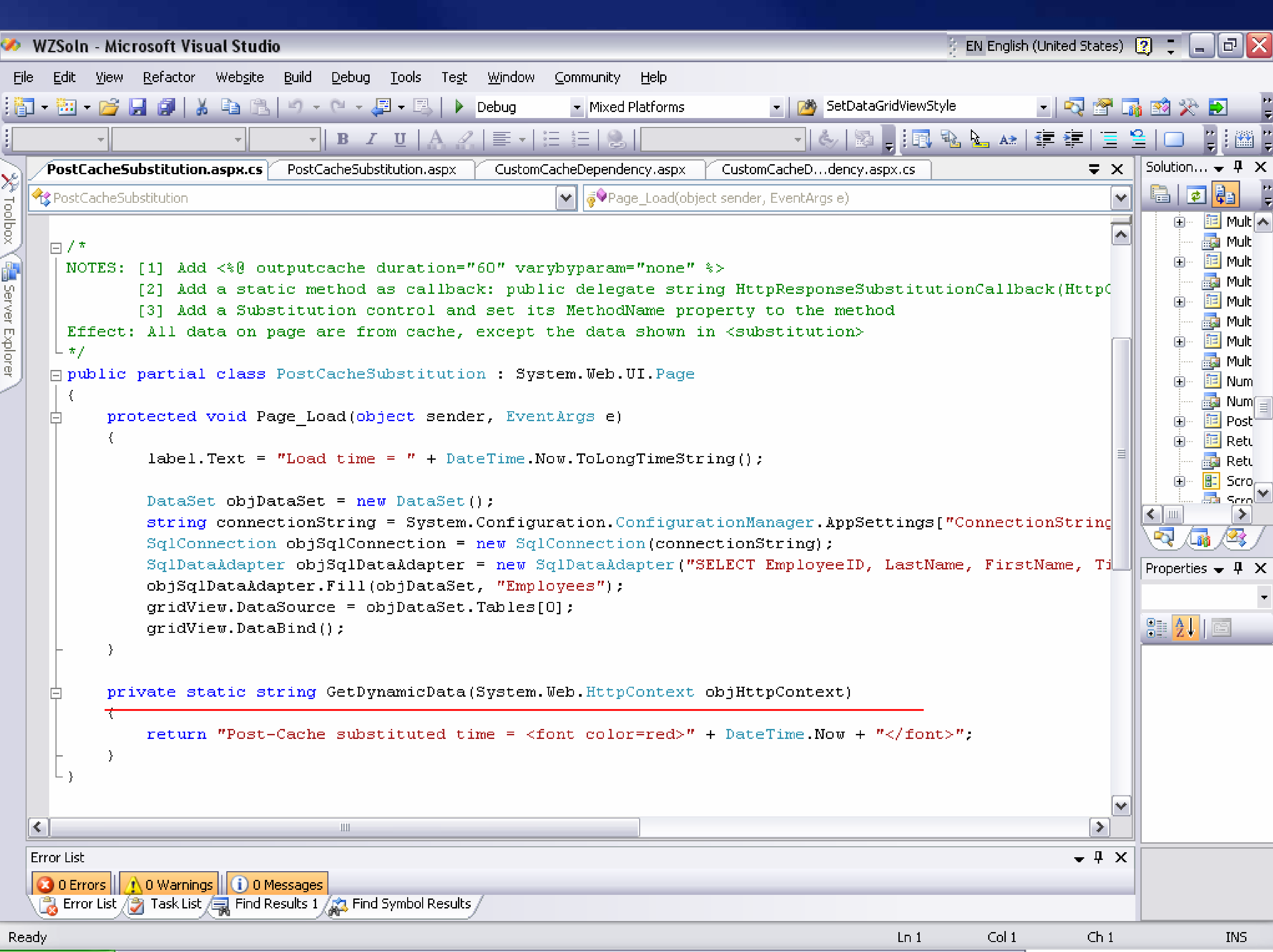
Post-Cache Substitution

- New **`Response.WriteSubstitution()`**
  - Wires-up substitution event on page
  - Adds a substitution buffer to the response
  - Substitution event returns string value to add
- New **`<asp:substitution />`** control
  - Drag-drop where content should go
  - Set the **`MethodName`** property
  - **`<asp:AdRotator>`** built-in support



```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="PostCacheSubstitution.aspx.cs" Inherits="PostCach
<%@ outputcache duration="60" varybyparam="none" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
    <div>
    <font face=verdana size=1>outputcahce duration="60". Within 60 seconds, as you refresh this page, all
        <br />
        <asp:Label ID="label" runat="server"></asp:Label>
        <br />
        <asp:Substitution ID="substitution" runat="server" MethodName="GetDynamicData" />
    </div>
        <asp:GridView ID="gridView" runat="server" Font-Names="Verdana" Font-Size="X-Small">
        </asp:GridView>
    </form>
</body>
</html>
```

WZSoln - Microsoft Visual Studio
File Edit View Website Build Debug Tools Test Window Community Help
Debug
Mixed Platforms
SetDataGridViewStyle
Internet Explorer 6.0
PostCacheSubstitution.aspx.cs
PostCacheSubstitution.aspx
CustomCacheDependency.aspx
CustomCacheD...dency.aspx.cs
Client Objects & Events
(No Events)
Design
Source
<html> <body> <form#form1> <div> <font>
Error List
0 Errors
0 Warnings
0 Messages
Error List Task List Find Results 1 Find Symbol Results
Ready
Ln 12 Col 31 Ch 31 INS
Solution...
Properties
<font>
(Id)
Acces
Atomic false
Class
Color
Conte inherit
Dir ltr
Face verdana
HideF
(Id)
Toolbox
Server Explorer
EN English (United States)


WZSoln - Microsoft Visual Studio
File Edit View Refactor Website Build Debug Tools Test Window Community Help
Debug
Mixed Platforms
SetDataGridViewStyle
PostCacheSubstitution.aspx.cs
PostCacheSubstitution.aspx
CustomCacheDependency.aspx
CustomCacheD...dency.aspx.cs
PostCacheSubstitution
Page_Load(object sender, EventArgs e)
Solution...
Properties
Error List
0 Errors
0 Warnings
0 Messages
Error List
Task List
Find Results 1
Find Symbol Results
Ready
Ln 1
Col 1
Ch 1
INS

```
/*
NOTES: [1] Add <%@ outputcache duration="60" varybyparam="none" %>
       [2] Add a static method as callback: public delegate string HttpResponseSubstitutionCallback(HttpC
       [3] Add a Substitution control and set its MethodName property to the method
Effect: All data on page are from cache, except the data shown in <substitution>
*/
public partial class PostCacheSubstitution : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        label.Text = "Load time = " + DateTime.Now.ToLongTimeString();

        DataSet objDataSet = new DataSet();
        string connectionString = System.Configuration.ConfigurationManager.AppSettings["ConnectionString
        SqlConnection objSqlConnection = new SqlConnection(connectionString);
        SqlDataAdapter objSqlDataAdapter = new SqlDataAdapter("SELECT EmployeeID, LastName, FirstName, Ti
        objSqlDataAdapter.Fill(objDataSet, "Employees");
        gridView.DataSource = objDataSet.Tables[0];
        gridView.DataBind();
    }

    private static string GetDynamicData(System.Web.HttpContext objHttpContext)
    {
        return "Post-Cache substituted time = <font color=red>" + DateTime.Now + "</font>";
    }
}
```

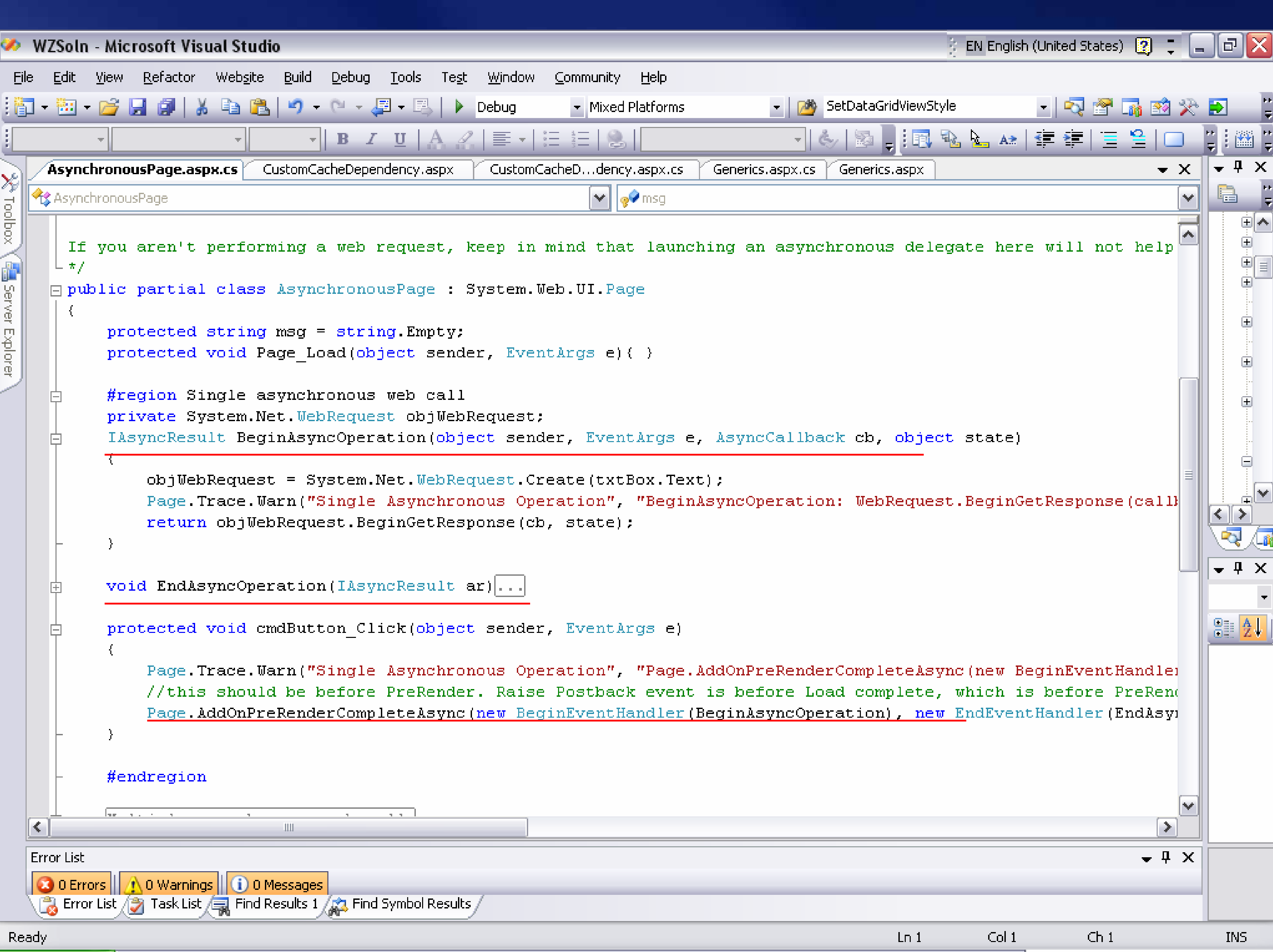
demo

- Post-Cache Substitution
- Source: PostCacheSubstitution.aspx

Asynchronous ASPX Page and Parallel Tasks

Asynchronous ASPX Page

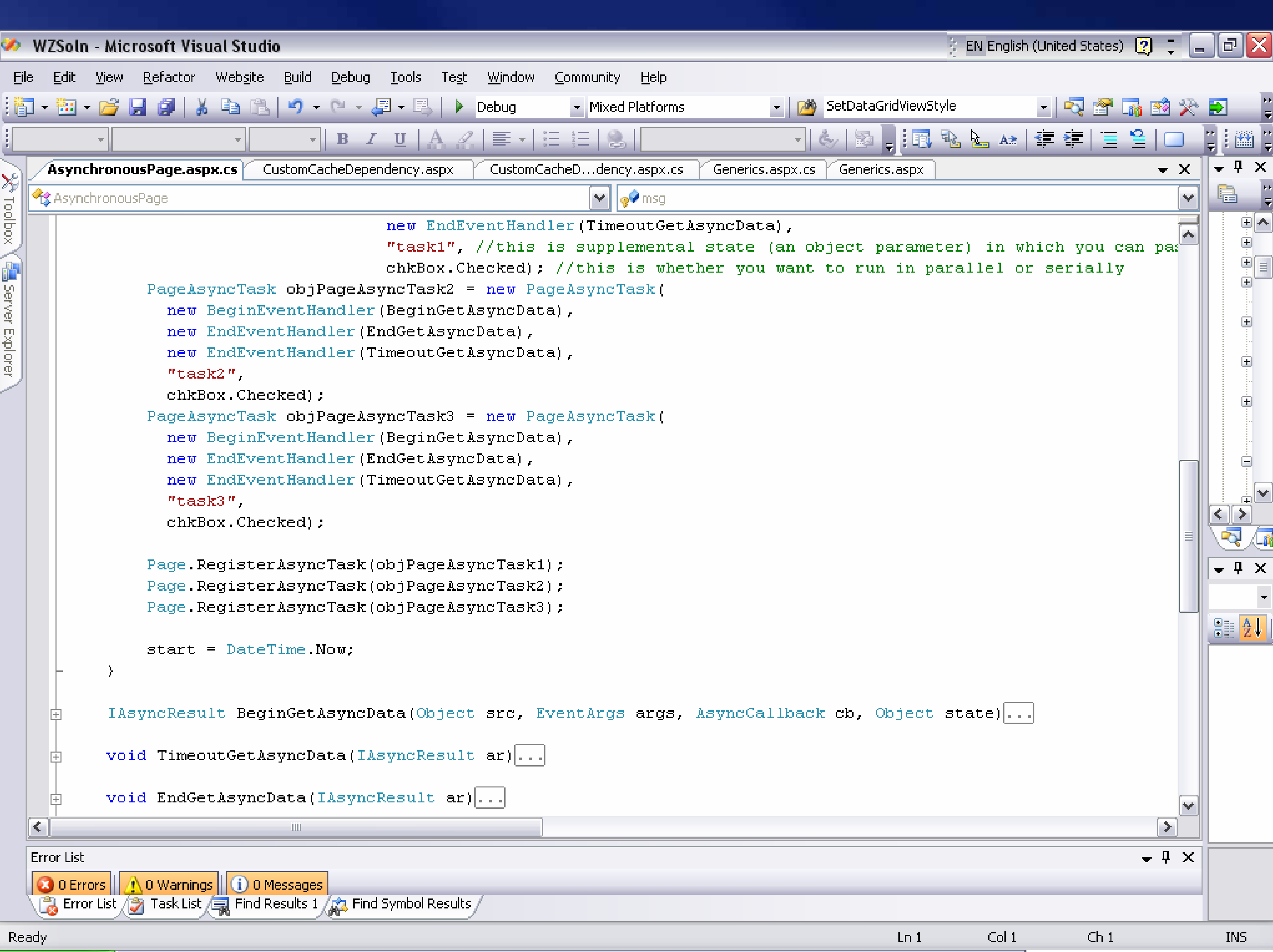
- By default, page processing in ASP.NET is synchronous
- Assigned thread does nothing else until the request completes
- ASP.NET has a limited number of threads at its disposal to process requests
- Requests are rejected with 503 "Server Unavailable" errors when queue is filled up to its capacity (100)
- Asynchronous ASPX page is for this


WZSoln - Microsoft Visual Studio
File Edit View Refactor Website Build Debug Tools Test Window Community Help
Debug
Mixed Platforms
SetDataGridViewStyle
AsynchronousPage.aspx.cs
CustomCacheDependency.aspx
CustomCacheD…dency.aspx.cs
Generics.aspx.cs
Generics.aspx
AsynchronousPage
msg
Toolbox
Server Explorer

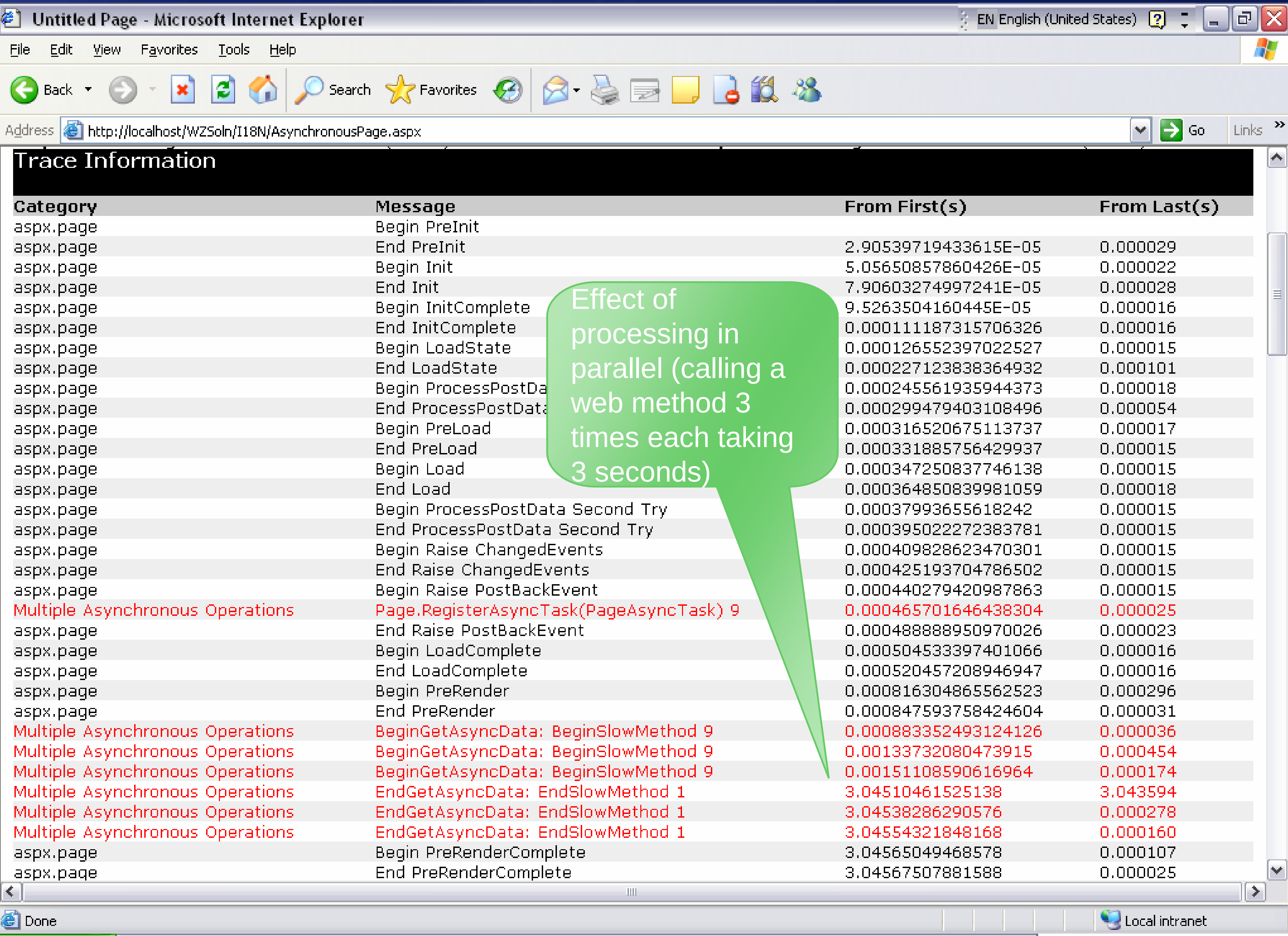
```
If you aren't performing a web request, keep in mind that launching an asynchronous delegate here will not help
*/
public partial class AsynchronousPage : System.Web.UI.Page
{
    protected string msg = string.Empty;
    protected void Page_Load(object sender, EventArgs e){ }

    #region Single asynchronous web call
    private System.Net.WebRequest objWebRequest;
    IAsyncResult BeginAsyncOperation(object sender, EventArgs e, AsyncCallback cb, object state)
    {
        objWebRequest = System.Net.WebRequest.Create(txtBox.Text);
        Page.Trace.Warn("Single Asynchronous Operation", "BeginAsyncOperation: WebRequest.BeginGetResponse(callb
        return objWebRequest.BeginGetResponse(cb, state);
    }

    void EndAsyncOperation(IAsyncResult ar)...

    protected void cmdButton_Click(object sender, EventArgs e)
    {
        Page.Trace.Warn("Single Asynchronous Operation", "Page.AddOnPreRenderCompleteAsync(new BeginEventHandler
        //this should be before PreRender. Raise Postback event is before Load complete, which is before PreRend
        Page.AddOnPreRenderCompleteAsync(new BeginEventHandler(BeginAsyncOperation), new EndEventHandler(EndAsyn
    }

    #endregion
```

Error List
0 Errors | 0 Warnings | 0 Messages
Error List | Task List | Find Results 1 | Find Symbol Results

Ready   Ln 1   Col 1   Ch 1   INS



```
                                        new EndEventHandler(TimeoutGetAsyncData),
                                        "task1", //this is supplemental state (an object parameter) in which you can pa
                                        chkBox.Checked); //this is whether you want to run in parallel or serially
        PageAsyncTask objPageAsyncTask2 = new PageAsyncTask(
          new BeginEventHandler(BeginGetAsyncData),
          new EndEventHandler(EndGetAsyncData),
          new EndEventHandler(TimeoutGetAsyncData),
          "task2",
          chkBox.Checked);
        PageAsyncTask objPageAsyncTask3 = new PageAsyncTask(
          new BeginEventHandler(BeginGetAsyncData),
          new EndEventHandler(EndGetAsyncData),
          new EndEventHandler(TimeoutGetAsyncData),
          "task3",
          chkBox.Checked);

        Page.RegisterAsyncTask(objPageAsyncTask1);
        Page.RegisterAsyncTask(objPageAsyncTask2);
        Page.RegisterAsyncTask(objPageAsyncTask3);

        start = DateTime.Now;
    }

    IAsyncResult BeginGetAsyncData(Object src, EventArgs args, AsyncCallback cb, Object state)...

    void TimeoutGetAsyncData(IAsyncResult ar)...

    void EndGetAsyncData(IAsyncResult ar)...
```

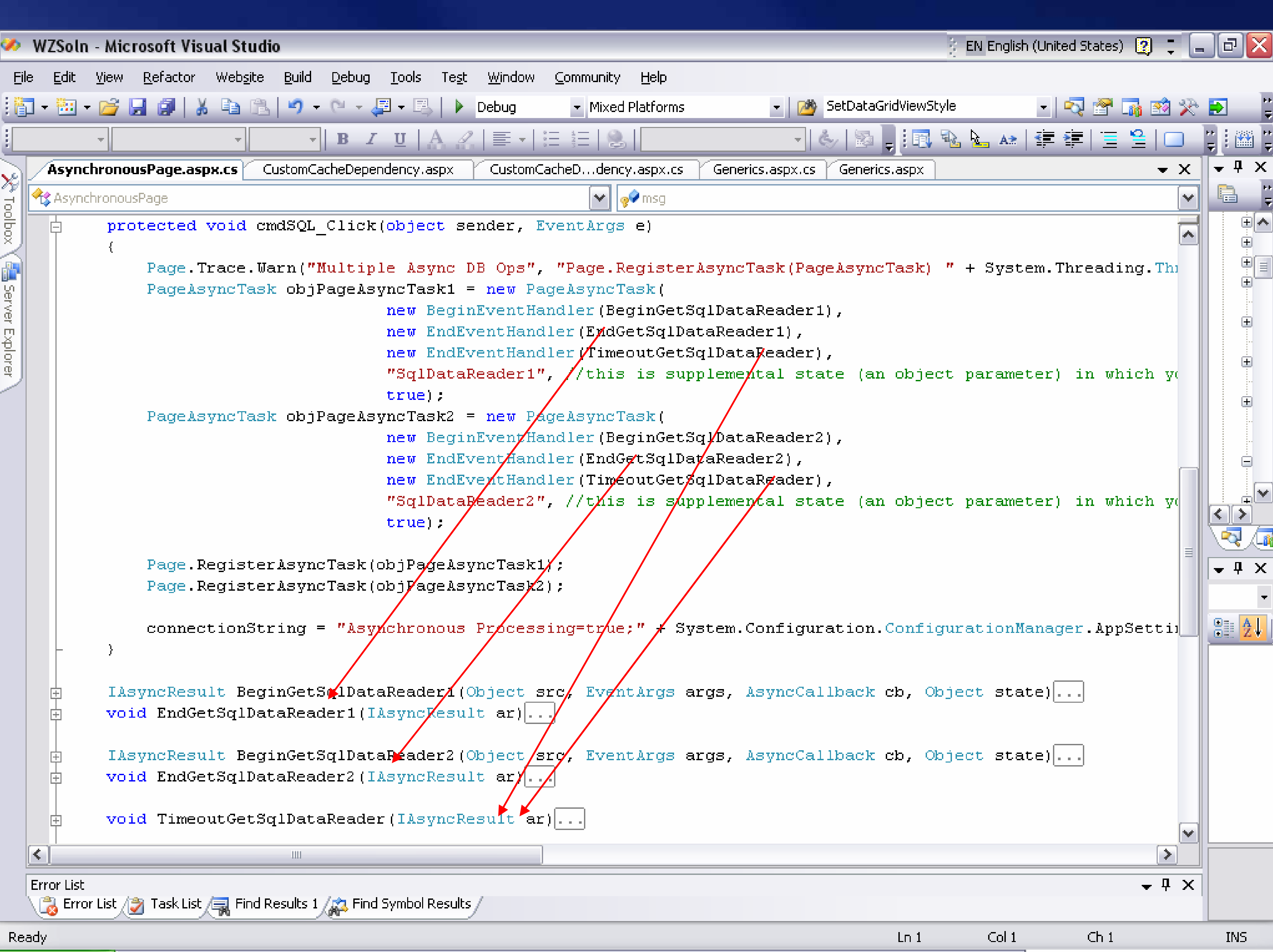

Untitled Page - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address
Go
Links
Done
Local intranet

http://localhost/WZSoln/I18N/AsynchronousPage.aspx

Trace Information

Category	Message	From First(s)	From Last(s)
aspx.page	Begin PreInit		
aspx.page	End PreInit	2.90539719433615E-05	0.000029
aspx.page	Begin Init	5.05650857860426E-05	0.000022
aspx.page	End Init	7.90603274997241E-05	0.000028
aspx.page	Begin InitComplete	9.5263504160445E-05	0.000016
aspx.page	End InitComplete	0.000111187315706326	0.000016
aspx.page	Begin LoadState	0.000126552397022527	0.000015
aspx.page	End LoadState	0.000227123838364932	0.000101
aspx.page	Begin ProcessPostDa	0.000245561935944373	0.000018
aspx.page	End ProcessPostData	0.000299479403108496	0.000054
aspx.page	Begin PreLoad	0.000316520675113737	0.000017
aspx.page	End PreLoad	0.000331885756429937	0.000015
aspx.page	Begin Load	0.000347250837746138	0.000015
aspx.page	End Load	0.000364850839981059	0.000018
aspx.page	Begin ProcessPostData Second Try	0.00037993655618242	0.000015
aspx.page	End ProcessPostData Second Try	0.000395022272383781	0.000015
aspx.page	Begin Raise ChangedEvents	0.000409828623470301	0.000015
aspx.page	End Raise ChangedEvents	0.000425193704786502	0.000015
aspx.page	Begin Raise PostBackEvent	0.000440279420987863	0.000015
Multiple Asynchronous Operations	Page.RegisterAsyncTask(PageAsyncTask) 9	0.000465701646438304	0.000025
aspx.page	End Raise PostBackEvent	0.000488888950970026	0.000023
aspx.page	Begin LoadComplete	0.000504533397401066	0.000016
aspx.page	End LoadComplete	0.000520457208946947	0.000016
aspx.page	Begin PreRender	0.000816304865562523	0.000296
aspx.page	End PreRender	0.000847593758424604	0.000031
Multiple Asynchronous Operations	BeginGetAsyncData: BeginSlowMethod 9	0.000883352493124126	0.000036
Multiple Asynchronous Operations	BeginGetAsyncData: BeginSlowMethod 9	0.00133732080473915	0.000454
Multiple Asynchronous Operations	BeginGetAsyncData: BeginSlowMethod 9	0.00151108590616964	0.000174
Multiple Asynchronous Operations	EndGetAsyncData: EndSlowMethod 1	3.04510461525138	3.043594
Multiple Asynchronous Operations	EndGetAsyncData: EndSlowMethod 1	3.04538286290576	0.000278
Multiple Asynchronous Operations	EndGetAsyncData: EndSlowMethod 1	3.04554321848168	0.000160
aspx.page	Begin PreRenderComplete	3.04565049468578	0.000107
aspx.page	End PreRenderComplete	3.04567507881588	0.000025

Effect of processing in parallel (calling a web method 3 times each taking 3 seconds)



```
    protected void cmdSQL_Click(object sender, EventArgs e)
    {
        Page.Trace.Warn("Multiple Async DB Ops", "Page.RegisterAsyncTask(PageAsyncTask) " + System.Threading.Thr
        PageAsyncTask objPageAsyncTask1 = new PageAsyncTask(
                                new BeginEventHandler(BeginGetSqlDataReader1),
                                new EndEventHandler(EndGetSqlDataReader1),
                                new EndEventHandler(TimeoutGetSqlDataReader),
                                "SqlDataReader1", //this is supplemental state (an object parameter) in which yo
                                true);
        PageAsyncTask objPageAsyncTask2 = new PageAsyncTask(
                                new BeginEventHandler(BeginGetSqlDataReader2),
                                new EndEventHandler(EndGetSqlDataReader2),
                                new EndEventHandler(TimeoutGetSqlDataReader),
                                "SqlDataReader2", //this is supplemental state (an object parameter) in which yo
                                true);

        Page.RegisterAsyncTask(objPageAsyncTask1);
        Page.RegisterAsyncTask(objPageAsyncTask2);

        connectionString = "Asynchronous Processing=true;" + System.Configuration.ConfigurationManager.AppSetti
    }

    IAsyncResult BeginGetSqlDataReader1(Object src, EventArgs args, AsyncCallback cb, Object state)...
    void EndGetSqlDataReader1(IAsyncResult ar)...

    IAsyncResult BeginGetSqlDataReader2(Object src, EventArgs args, AsyncCallback cb, Object state)...
    void EndGetSqlDataReader2(IAsyncResult ar)...

    void TimeoutGetSqlDataReader(IAsyncResult ar)...
```

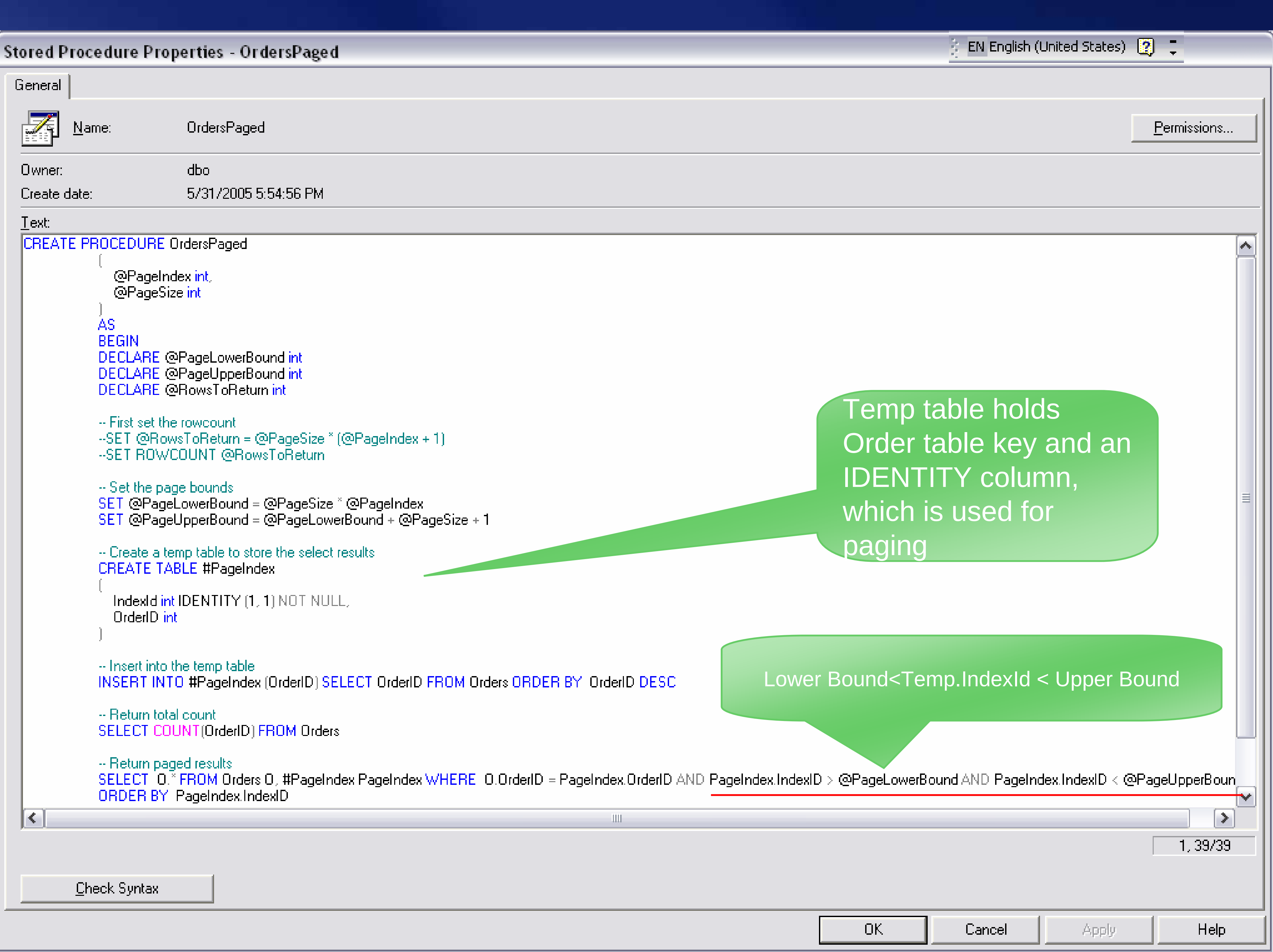
demo

- Asynchronous ASPX Page with parallel processing
- Source: AsynchronousPage.aspx

Data Paging via Stored Procedure

Data Paging via SP

- DataGrid (ver 1.1) and GridView(ver 2.0) both do data paging
- However, the price is large ViewState.
- Your data layer will need to return all of the data and then the DataGrid will filter all the displayed records based on the current page.
- Use SP to return proper page of data, only, not all data.


Stored Procedure Properties - OrdersPaged
EN English (United States)
General
Permissions...
Check Syntax
OK
Cancel
Apply
Help
1, 39/39

Name: OrdersPaged

Owner: dbo

Create date: 5/31/2005 5:54:56 PM

Text:

```
CREATE PROCEDURE OrdersPaged
        (
          @PageIndex int,
          @PageSize int
        )
        AS
        BEGIN
        DECLARE @PageLowerBound int
        DECLARE @PageUpperBound int
        DECLARE @RowsToReturn int

        -- First set the rowcount
        --SET @RowsToReturn = @PageSize * (@PageIndex + 1)
        --SET ROWCOUNT @RowsToReturn

        -- Set the page bounds
        SET @PageLowerBound = @PageSize * @PageIndex
        SET @PageUpperBound = @PageLowerBound + @PageSize + 1

        -- Create a temp table to store the select results
        CREATE TABLE #PageIndex
        (
          IndexId int IDENTITY (1, 1) NOT NULL,
          OrderID int
        )

        -- Insert into the temp table
        INSERT INTO #PageIndex (OrderID) SELECT OrderID FROM Orders ORDER BY  OrderID DESC

        -- Return total count
        SELECT COUNT(OrderID) FROM Orders

        -- Return paged results
        SELECT  O.* FROM Orders O, #PageIndex PageIndex WHERE  O.OrderID = PageIndex.OrderID AND PageIndex.IndexID > @PageLowerBound AND PageIndex.IndexID < @PageUpperBoun
        ORDER BY  PageIndex.IndexID
```

Temp table holds Order table key and an IDENTITY column, which is used for paging

Lower Bound<Temp.IndexId < Upper Bound

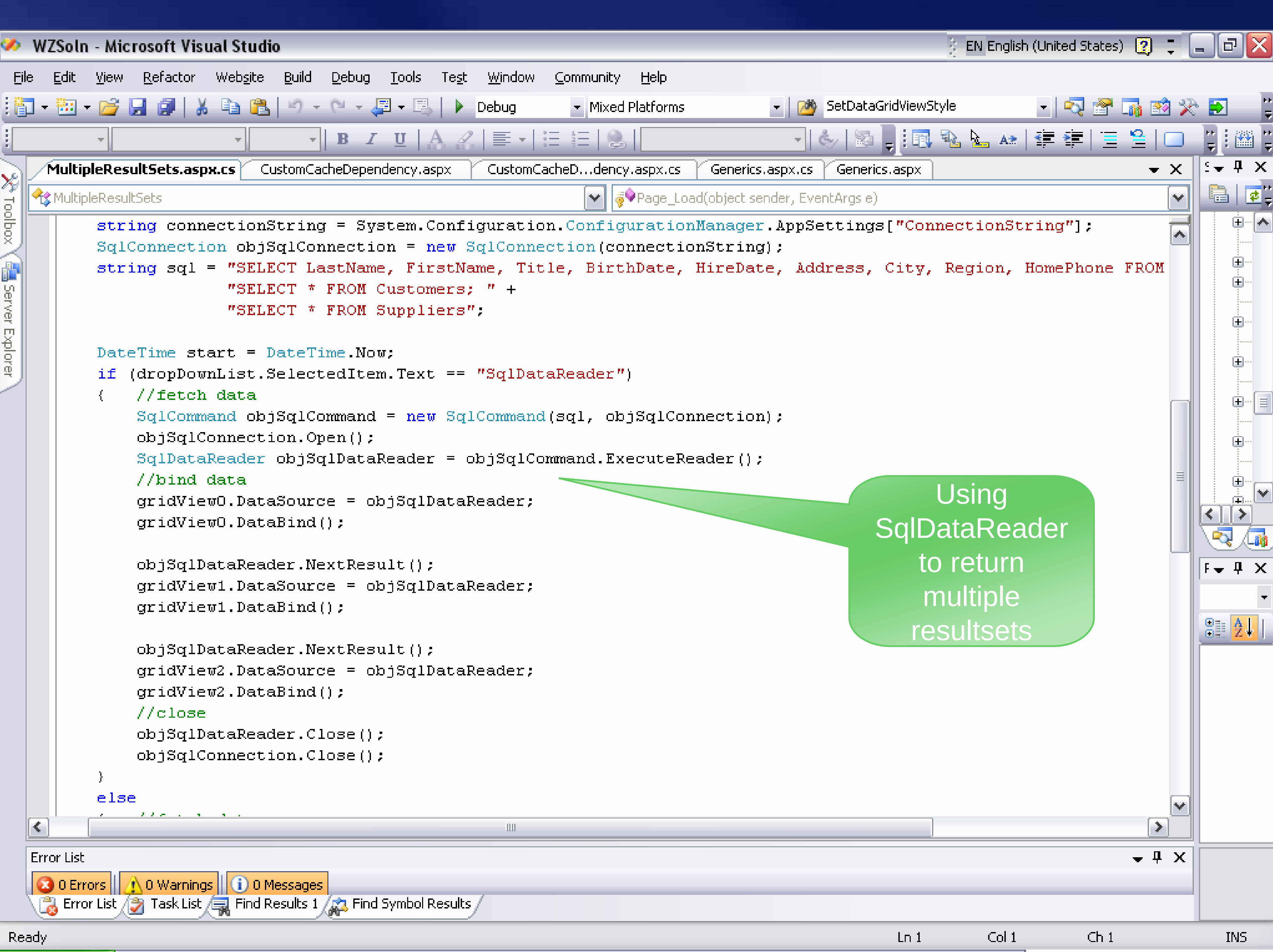
demo

- Data paging using stored procedure
- Source: DataPaging.aspx and DataPagingClient.aspx

Returning Multiple Resultsets

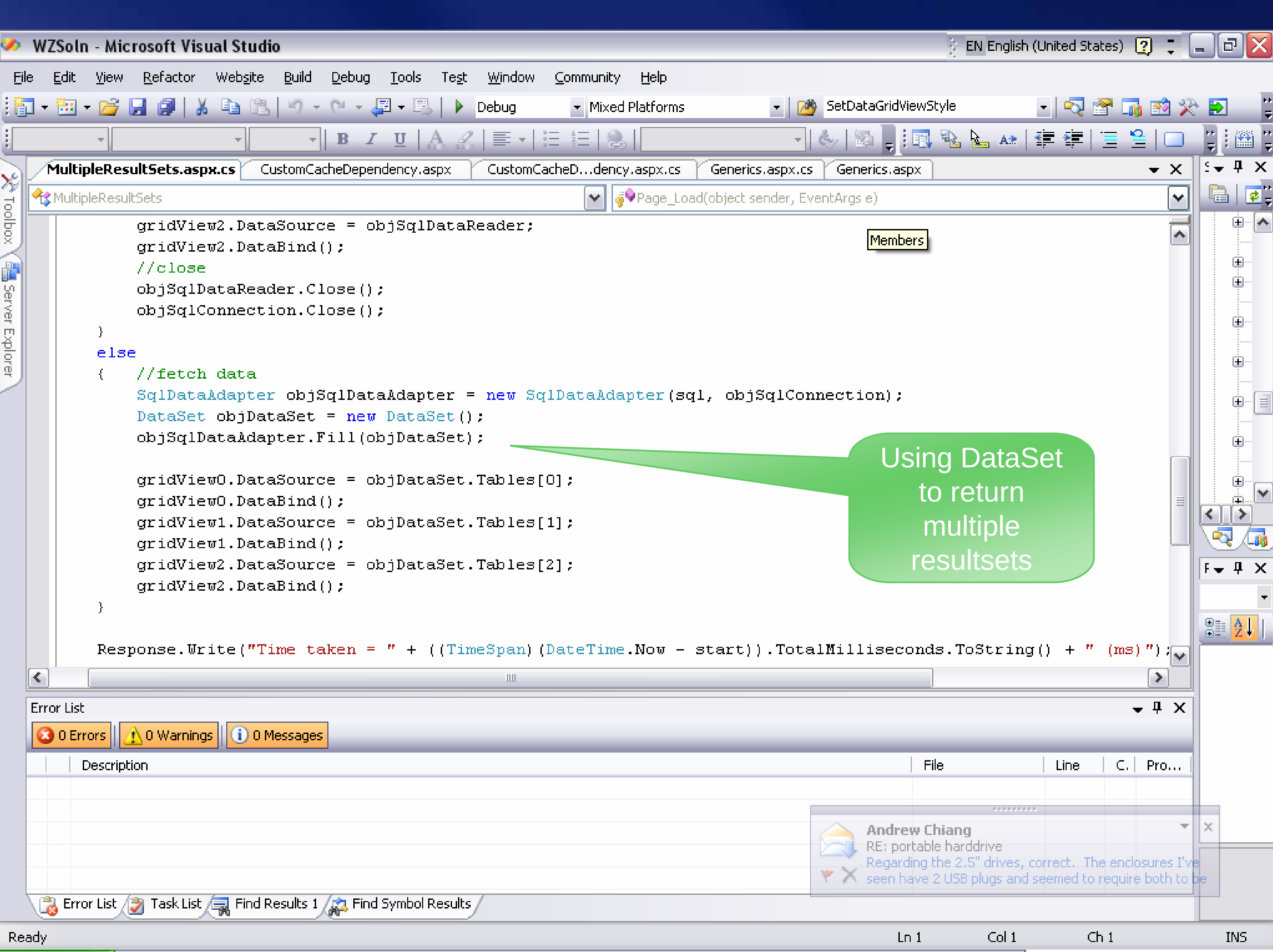
Returning Multiple Resultsets

- Improve scalability by reducing cross process/network requests
- Both DataSet and SqlDataReader allow you to return multiple resultsets


WZSoln - Microsoft Visual Studio
File Edit View Refactor Website Build Debug Tools Test Window Community Help
Debug
Mixed Platforms
SetDataGridViewStyle
MultipleResultSets.aspx.cs
CustomCacheDependency.aspx
CustomCacheD...dency.aspx.cs
Generics.aspx.cs
Generics.aspx
MultipleResultSets
Page_Load(object sender, EventArgs e)
Toolbox
Server Explorer

```
    string connectionString = System.Configuration.ConfigurationManager.AppSettings["ConnectionString"];
    SqlConnection objSqlConnection = new SqlConnection(connectionString);
    string sql = "SELECT LastName, FirstName, Title, BirthDate, HireDate, Address, City, Region, HomePhone FROM
                 "SELECT * FROM Customers; " +
                 "SELECT * FROM Suppliers";

    DateTime start = DateTime.Now;
    if (dropDownList.SelectedItem.Text == "SqlDataReader")
    {   //fetch data
        SqlCommand objSqlCommand = new SqlCommand(sql, objSqlConnection);
        objSqlConnection.Open();
        SqlDataReader objSqlDataReader = objSqlCommand.ExecuteReader();
        //bind data
        gridView0.DataSource = objSqlDataReader;
        gridView0.DataBind();

        objSqlDataReader.NextResult();
        gridView1.DataSource = objSqlDataReader;
        gridView1.DataBind();

        objSqlDataReader.NextResult();
        gridView2.DataSource = objSqlDataReader;
        gridView2.DataBind();
        //close
        objSqlDataReader.Close();
        objSqlConnection.Close();
    }
    else
```

Using SqlDataReader to return multiple resultsets



WZSoln - Microsoft Visual Studio
File Edit View Refactor Website Build Debug Tools Test Window Community Help
Debug
Mixed Platforms
SetDataGridViewStyle
MultipleResultSets.aspx.cs
CustomCacheDependency.aspx
CustomCacheD...dency.aspx.cs
Generics.aspx.cs
Generics.aspx
MultipleResultSets
Page_Load(object sender, EventArgs e)
Members
Error List
0 Errors
0 Warnings
0 Messages
Description
File
Line
Ready
Ln 1
Col 1
Ch 1
INS

```
        gridView2.DataSource = objSqlDataReader;
        gridView2.DataBind();
        //close
        objSqlDataReader.Close();
        objSqlConnection.Close();
    }
    else
    {   //fetch data
        SqlDataAdapter objSqlDataAdapter = new SqlDataAdapter(sql, objSqlConnection);
        DataSet objDataSet = new DataSet();
        objSqlDataAdapter.Fill(objDataSet);

        gridView0.DataSource = objDataSet.Tables[0];
        gridView0.DataBind();
        gridView1.DataSource = objDataSet.Tables[1];
        gridView1.DataBind();
        gridView2.DataSource = objDataSet.Tables[2];
        gridView2.DataBind();
    }

    Response.Write("Time taken = " + ((TimeSpan)(DateTime.Now - start)).TotalMilliseconds.ToString() + " (ms)");
```

Using DataSet to return multiple resultsets

Andrew Chiang
RE: portable harddrive
Regarding the 2.5" drives, correct. The enclosures I've seen have 2 USB plugs and seemed to require both to be

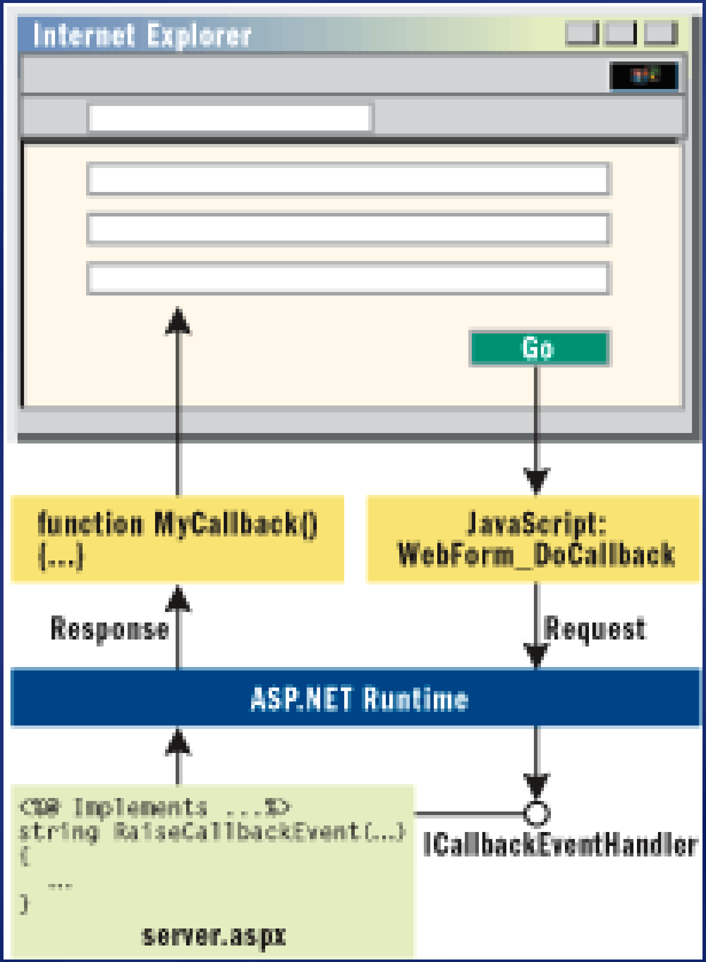
demo

- Returning multiple resultsets
- Source: MultipleResultSets.aspx

Script Callback

Script Callback

- Making server round trip without page postback

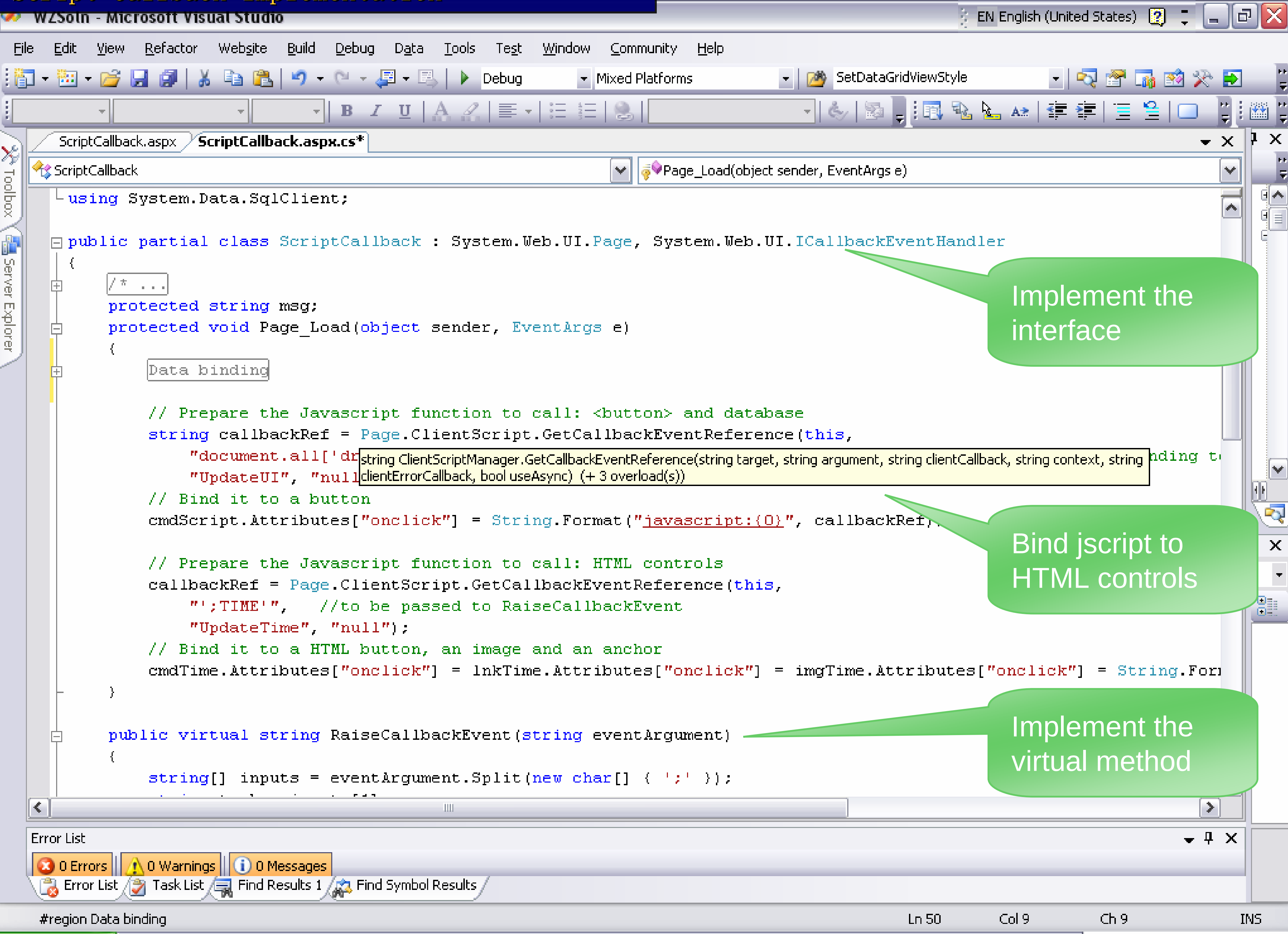


```
<%@ Implements ...%>
string RaiseCallbackEvent(...)
{
    ...
}
```

Script Callback Implementation

- Implement interface System.Web.UI.ICallbackEventHandler
- Implement public virtual string RaiseCallbackEvent(string eventArgument)
- Bind jscript string to HTML controls (not input type) using Page.ClientScript.GetCallbackEventReference() method

Script Callback Implementation



```
using System.Data.SqlClient;

public partial class ScriptCallback : System.Web.UI.Page, System.Web.UI.ICallbackEventHandler
{
    /* ...
    protected string msg;
    protected void Page_Load(object sender, EventArgs e)
    {
        Data binding

        // Prepare the Javascript function to call: <button> and database
        string callbackRef = Page.ClientScript.GetCallbackEventReference(this,
            "document.all['dr                                                                  nding t
            "UpdateUI", "null
        // Bind it to a button
        cmdScript.Attributes["onclick"] = String.Format("javascript:{0}", callbackRef)

        // Prepare the Javascript function to call: HTML controls
        callbackRef = Page.ClientScript.GetCallbackEventReference(this,
            "';TIME'",   //to be passed to RaiseCallbackEvent
            "UpdateTime", "null");
        // Bind it to a HTML button, an image and an anchor
        cmdTime.Attributes["onclick"] = lnkTime.Attributes["onclick"] = imgTime.Attributes["onclick"] = String.For
    }

    public virtual string RaiseCallbackEvent(string eventArgument)
    {
        string[] inputs = eventArgument.Split(new char[] { ';' });
```

```
string ClientScriptManager.GetCallbackEventReference(string target, string argument, string clientCallback, string context, string
clientErrorCallback, bool useAsync) (+ 3 overload(s))
```

Implement the interface

Bind jscript to HTML controls

Implement the virtual method

demo

- Script callback (out of band call)
- Source: ScriptCallback.aspx

Summary

- SQL Server cache dependency (SqlCacheDependency)
- Custom cache dependency (CacheDependency)
- Post-cache substitution
- Asynchronous page with parallel-processed tasks
- Data paging via stored procedure
- Returning multiple result sets from DB
- Server round trip without postback: script callback
- OTHERS (not covered in this talk):
  - Windows Server 2003 features
    - Kernel mode caching in IIS 6.0
    - Gzip compression
    - Use mscorsvr.dll instead of mscorwks.dll
  - In stored procedures
    - Use Set NOCOUNT ON to prevent DONE_IN_PROC messages
    - Do not use sp_prefix in stored proc names to prevent checking into master db
  - Connection pooling

Q&A