

Introduction to ASP.NET

Paul Litwin
Lead Programmer, Collaborative Data Services
plitwin@fhcrc.org

FHCRC Collaborative Data Services

1

CDS Seminars & Training Classes

- ◆ CDS Brownbag seminars
 - June 14, Upsizing Access to SQL Server, Paul Litwin
 - July 12, Introduction to ASP.NET, Paul Litwin
 - Aug 9, Life Made Easier w/ Regular Expressions, John Syre
 - Sep 13, Hacked: Understanding and Preventing Malicious Input Attacks, Paul Litwin
- ◆ Training classes
 - July 20-Aug 2, 8:30-12:30, ASP.NET with C# Class, Paul Litwin
 - SQL Server for Access Developers, sometime in Sept/Oct timeframe
- ◆ Send me email (plitwin) to get on announcement list for seminars/classes

FHCRC Collaborative Data Services

2

Me.About

- ◆ **Developer**
 - Focus: ASP.NET, ASP, VB, C#, SQL Server, Access, ...
 - Microsoft ASP.NET MVP, MCSD
 - Based in Seattle, WA
 - After 11 years on my own, I recently took a position as Lead Programmer at Fred Hutchinson Cancer Research Center in Collaborative Data Services (CDS)
- ◆ **Co-Founder and Owner**
 - Deep Training
 - www.deeptraining.com
- ◆ **Conference Chair/Speaker/User Group Leader**
 - Chair, Microsoft ASP.NET Connections
 - Speaker at Tech*Ed, VSLive, DevDays
 - Member INETA Speakers Bureau
- ◆ **Author**
 - Author/co-author of a dozen books, including...
 - ASP.NET for Developers
 - Five editions of Access Developer Handbooks
 - Access Cookbook

FHCRC Collaborative Data Services

3

Agenda

- ◆ Web Programming
- ◆ ASP and ASP.NET
- ◆ Web Forms
- ◆ Data Access
- ◆ Web Services
- ◆ Caching, Configuration & Deployment

FHCRC Collaborative Data Services

4

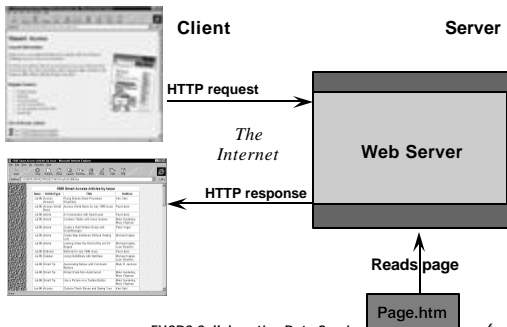
Brief History of Web Programming

- ◆ In the beginning there was plain old (static) HTML
- ◆ Client-side JavaScript improved the user experience but since the code ran on the client can only go so far
- ◆ Server-side Web programming (ASP, Cold Fusion, JSP, PHP, ASP.NET, etc.) runs scripts or compiled programs on the server to create truly dynamic pages

FHCRC Collaborative Data Services

5

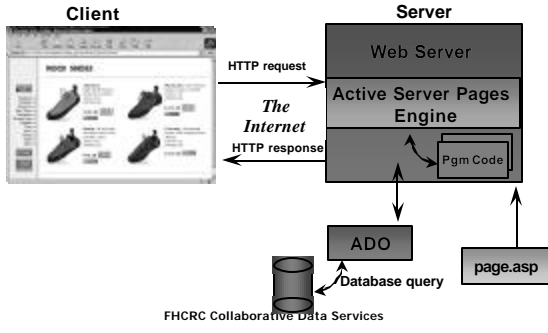
Static HTML



FHCRC Collaborative Data Services

6

Server-Side Programming



Agenda

- ◆ Web Programming
- ◆ ASP and ASP.NET
- ◆ Web Forms
- ◆ Data Access
- ◆ Web Services
- ◆ Caching, Configuration & Deployment

ASP (Classic)

- ◆ Most popular web app development platform
 - Millions of developers have used ASP
 - Basis for most web platforms today (JSP, PHP)
- ◆ Simple and approachable
 - No compilers required – “just hit save”
 - Easy scripting languages + simple object model

ASP (Classic): The Not So Good

- ◆ Too much code required
- ◆ Spaghetti page code
- ◆ Limited language support
- ◆ Weak tool support
- ◆ Scripting engines limit performance

FHCRC Collaborative Data Services

10

ASP.NET

- ◆ Rich Page Development Model
 - Declarative server-side UI control model
 - Clean code/content organization
 - Validation, data manipulation, state management
 - Choice of compiled language (VB, C#, Delphi, etc.)
- ◆ Built-in XML Web Service Support
 - Enable rich application-to-application communication
- ◆ Rich Security Model
 - Forms and Windows Authentication
- ◆ Great Deployment, Reliability & Performance
 - No Registration Required: "XCopy" deployment
 - Built-in crash, memory leak and deadlock recovery
 - Caching of pages significantly improves performance

FHCRC Collaborative Data Services

11

Agenda

- ◆ Web Programming
- ◆ ASP and ASP.NET
- ◆ Web Forms
- ◆ Data Access
- ◆ Web Services
- ◆ Caching, Configuration & Deployment

FHCRC Collaborative Data Services

12

Before ASP.NET Web Forms

- ◆ In the olden days of server-side programming (ASP classic, Cold Fusion, JSP, PHP, etc.)...
 - Web apps were made up of lots of pages...
 - When you wanted to have some action occur you posted a form to another page where server-side ASP code ran
 - No event-model. To react to something required another page

FHCRC Collaborative Data Services

13

Enter ASP.NET Web Form

- ◆ ASP.NET Web Form:
smart *self-posting* Web page
 - Reduces the number of pages in the app
 - Brings event-driven programming model popularized by Visual Basic to web programming
 - Brings structured, object-oriented programming to the Web

FHCRC Collaborative Data Services

14

ASP.NET Web Form

- ◆ How does it work?
 - Add runat="server" attribute to form
 - Add runat="server" controls to form
 - Write runat="server" event-handlers that react to events with server-side code
 - Refer to controls by their server-side ID attribute
 - No need to work with confusing client-side DOM-like hierarchy

VS
.NET
makes
easy

FHCRC Collaborative Data Services

15

Self-Posting ASP.NET Web Form

```
Sub Page_Load()  
If Not Page.IsPostBack Then  
BindDataGrid()  
End If  
Sub Edit_Command()  
Sub Update_Command()  
Dim drFound As DataRow  
drFound =  
ds.Tables("Employees").  
e.Item.Cells(2).Text  
...  
End Sub
```

Client



Server

FHCRC Collaborative Data Services

16

ASP.NET Server Controls

- ◆ Denoted by tag with `runat="server"`
 - `<input type="text" runat="server">`
 - `<asp:textbox runat="server">`
- ◆ Handle both rendering and interaction
 - Controls render HTML to the client
 - ASP.NET automatically maintains state (properties) of all server-side controls between postings. (This feature is called ViewState)
 - ASP.NET fires events to trigger user code

FHCRC Collaborative Data Services

17

ASP.NET Server Controls

- ◆ Web server controls include
 - TextBox
 - Label
 - DropDownList
 - ListBox
 - RadioButtonList
 - DataList
 - DataGrid
 - Validation controls
 - ...lots more...

FHCRC Collaborative Data Services

18

Demo

- ◆ Building an ASP.NET Web Form (page) with Visual Studio .NET

FHCRC Collaborative Data Services

19

Agenda

- ◆ Web Programming
- ◆ ASP and ASP.NET
- ◆ Web Forms
- ◆ Data Access
- ◆ Web Services
- ◆ Caching, Configuration & Deployment

FHCRC Collaborative Data Services

20

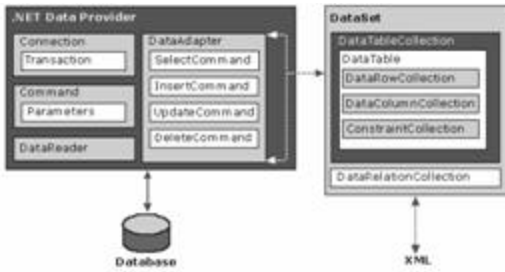
.NET Data Access

- ◆ Data access in .NET is accomplished via a number of classes referred to collectively as ADO.NET
- ◆ Two basic sets of classes
 - Classes used to connect to a data source
 - Classes used to manipulate disconnected data in memory

FHCRC Collaborative Data Services

21

ADO.NET Object Model



FHCRC Collaborative Data Services

22

Connecting to Data Sources

- ◆ xConnection – use to establish connection to data source
- ◆ xCommand class – use to execute ad-hoc SQL or stored proc
- ◆ xDataReader class – use to hold data returned by xCommand
- ◆ xDataAdapter class – use to ferry data to/from DataSet class
where x is Sql, OleDb, Oracle, or Odbc

FHCRC Collaborative Data Services

23

Demo

- ◆ Updating data with SqlConnection and SqlCommand classes

FHCRC Collaborative Data Services

24

Manipulating Disconnected Data in Memory

- ◆ The disconnected classes for manipulating data
 - DataSet
 - DataTable
 - DataRow
 - DataColumn
 - DataView
 - DataRelation
- ◆ These objects have no knowledge of where the data came from

FHCRC Collaborative Data Services

25

Demo

- ◆ DataGrid bound to DataSet

FHCRC Collaborative Data Services

26

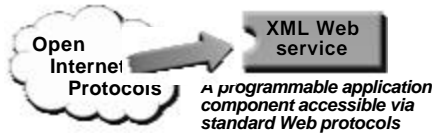
Agenda

- ◆ Web Programming
- ◆ ASP and ASP.NET
- ◆ Web Forms
- ◆ Data Access
- ◆ Web Services
- ◆ Caching, Configuration & Deployment

FHCRC Collaborative Data Services

27

Web Service



- ◆ What is it?
 - A programmable application component accessible via standard Web protocols
 - Built on top of SOAP (Simple Object Access Protocol)
 - Examples: stock quote, weather, and work flow, team collaboration
 - Makes building distributed applications easy
 - Interoperability in a heterogeneous world

FHCRC Collaborative Data Services

28

Creating a Web Service with VS .NET

- ◆ Create a Project of type ASP.NET Web Service
- ◆ Basic skeleton of Web Service is created for you
- ◆ Add Web Methods and go!

FHCRC Collaborative Data Services

29

Web Service Help Page

- ◆ If you browse to an .asmx page, ASP.NET automatically generates a help page that..
 - describes all the Web Service's operations (methods)
 - lets you browse the WSDL (Service Description) of the service
 - lets you test the Web Service using a sample form

FHCRC Collaborative Data Services

30

Demo

- ◆ Building a Web Service with Visual Studio .NET

FHCRC Collaborative Data Services

31

Web Service Clients

- ◆ Regardless of the platform used, to create a Web Service client you must program to the Web Service's API or contract; this requires
 - Understanding services' Web Services Description Language (WSDL)
 - Ability to communicate using Simple Object Access Protocol (SOAP)
- ◆ Fortunately, Visual Studio .NET understands both WSDL and SOAP which means that you don't have to

FHCRC Collaborative Data Services

32

Creating a Web Service Client w/ VS .NET

- ◆ Create new ASP.NET Web Application (or Windows Form app or another Web Service app...)
- ◆ Right-click on References in Solution Explorer and select Add Web Reference
 - Enter http address of service in dialog
 - After navigating to service (.asmx), click on enabled Add Reference button
- ◆ Proxy is automatically generated for you
 - Proxy talks to Web Service using SOAP
 - You code against proxy (w/ IntelliSense support) as if Web Service were local class

FHCRC Collaborative Data Services

33

Demo

- ◆ Building a Web Service client with Visual Studio .NET

FHCRC Collaborative Data Services

34

Agenda

- ◆ Web Programming
- ◆ ASP and ASP.NET
- ◆ Web Forms
- ◆ Data Access
- ◆ Web Services
- ◆ Caching, Configuration & Deployment

FHCRC Collaborative Data Services

35

Caching

- ◆ ASP.NET supports
 - Page caching – ASP.NET stores html image of page in memory for desired lifetime; can be used with form and querystring parameters
 - Fragment caching – cache portions of pages using ASP.NET user controls
 - Object caching – store objects (e.g., DataSet) in memory for desired lifetime

FHCRC Collaborative Data Services

36

ASP.NET Configuration Model

- ◆ Stored in plain text (Web.Config)
- ◆ No more registry hacking
- ◆ Uses XML for easy extendibility
- ◆ Cascading .Config files supported
 - Highest level is Machine.Config
 - Application Web.Config
 - Folder Web.Config
 - ...
- ◆ Cached in memory to reduce bottlenecks

FHCRC Collaborative Data Services

37

Deployment

- ◆ X-Copy deployment – copy and go
- ◆ No need to stop the Web server when updating components

FHCRC Collaborative Data Services

38

Demo

- ◆ Improving performance with page caching

FHCRC Collaborative Data Services

39

ASP.NET Future

- ◆ ASP.NET 2.0, part of Visual Studio 2005 is in beta now
- ◆ Very exciting improvements, including
 - Smarter controls & automated data-binding
 - Master pages & themes
 - Built-in membership, personalization & role manager
 - Database invalidation of cache
 - Significant reduction in code you have to write!

FHCRC Collaborative Data Services

40

Thank You

- ◆ Paul Litwin, Collaborative Data Services
 - plitwin@fhcrc.org
 - x3426
- ◆ Slides and samples
 - download from www.deeptraining.com/fhcrc

FHCRC Collaborative Data Services

41
