ASP.NET 2.0
- Y\@œœç Á¸^¸Ñ

Mike Ormond
Developer & Platform Group, Microsoft

mike.ormond@microsoft.com
http://blogs.msdn.com/mikeormond

Agenda

ASP.NET 2.0 Overview

- ASP.NET 2.0 Goals
- ASP.NET 2.0 Developer Stack
- Visual Studio 2005 Web Development

Y @œœq ÁP^¸ Ñ

- User Interface
- Personalisation
- Data Access
- Security
- Code Behind 2.0
- Client Callbacks

ASP.NET 2.0 Themes

Developer Productivity

Administration and Management

Themes

Totally Extensible Platform

Performance and Scalability

ASP.NET 2.0 Developer Stack
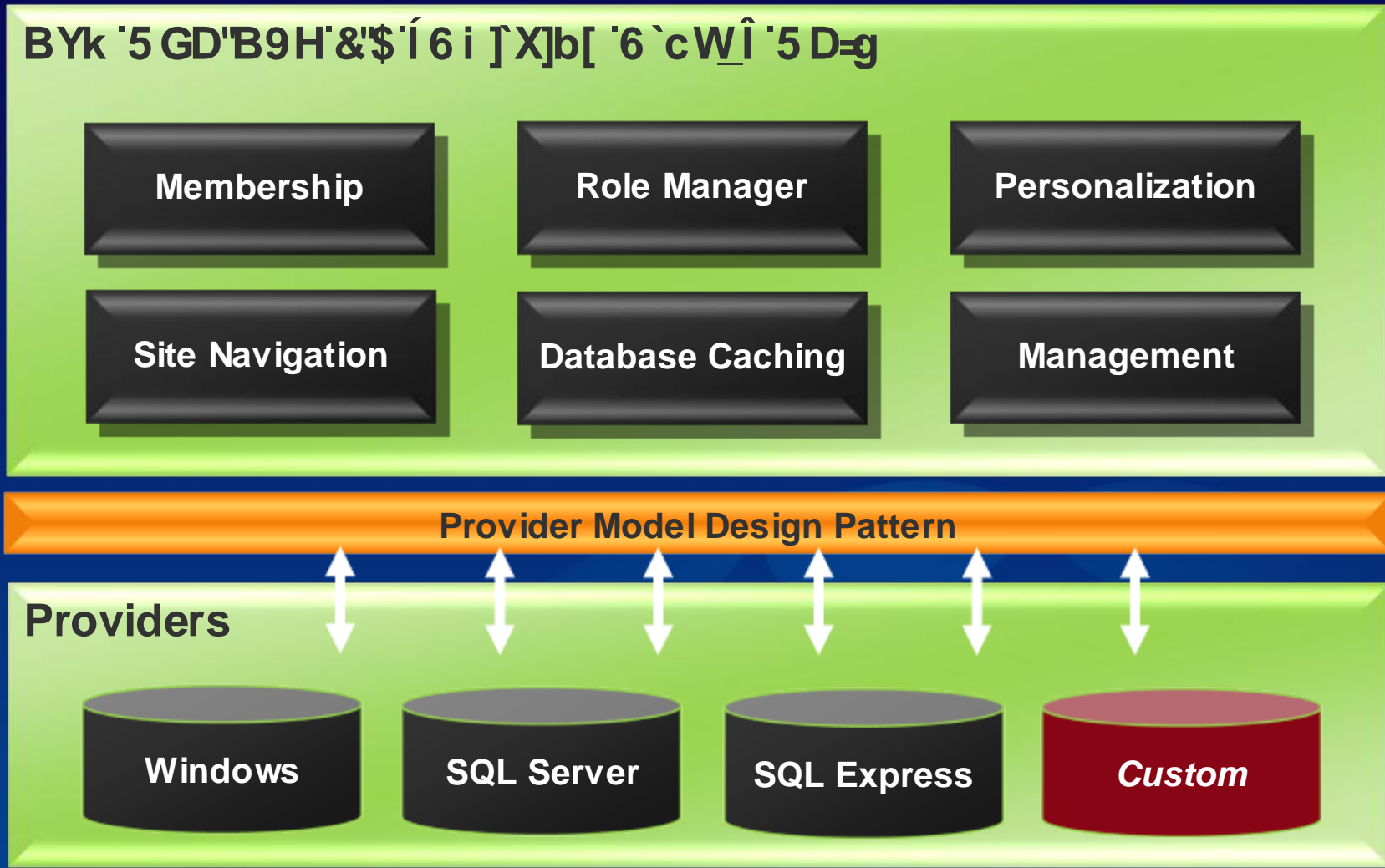
BYk ˙5 GD'B9H˙&'$˙Í 6 i]˙X]b[˙6 `cW̲Î ˙5 D=g

Membership	**Role Manager**	**Personalization**
Site Navigation	**Database Caching**	**Management**

ASP.NET 2.0 Developer Stack

BYk ˙5 GD'B9H˙&'$˙Í 6 i]`X]b[ ˙6 `cW̲Î ˙5 D=g

Membership	Role Manager	Personalization
Site Navigation	Database Caching	Management

Provider Model Design Pattern

Providers

Windows	SQL Server	SQL Express	*Custom*

The Provider Model

Controls

Login | LoginStatus | LoginView | Other Login Controls

Membership API

Membership | MembershipUser

Membership Providers

SqlMembershipProvider | Other Membership Providers

Membership Data

SQL Express | SQL Server | Other Data Stores

ASP.NET 2.0 Developer Stack

BYk ˙5 GD'ʼB9H˙&'ʼ$ ˙Í 6 i]ˋX]b[˙6 `c W̲Î ˙5 D=g

Membership	**Role Manager**	**Personalization**
Site Navigation	**Database Caching**	**Management**

ASP.NET 2.0 Developer Stack

BYk ˙5 GD'B9 H'&'$˙Í DUͺ Y˙: f Ua Yk c f_Î ˙: YUͱi f Yg

Master Pages	**Themes/Skins**	**Adaptive UI**

BYk ˙5 GD'B9 H'&'$˙Í 6 i]̖ X]b[˙6 `c W̲Î ˙5 D=g

Membership	**Role Manager**	**Personalization**
Site Navigation	**Database Caching**	**Management**

ASP.NET 2.0 Developer Stack

BYk ˙5 GD"B9 H ˙&'$ ˙Í 7 cblfc`˙6 i W_YhgÎ ˙f5 ddfcl]a UhY m˙) $Ł

Security   **Data**   **Navigation**   **Web Parts**

BYk ˙5 GD"B9 H ˙&'$ ˙Í DŲ Y : fU˙ Yk cf_Î ˙: YUli fYg

Master Pages   **Themes/Skins**   **Adaptive UI**

BYk ˙5 GD"B9 H ˙&'$ ˙Í 5 i]˙ X7b[˙6 `cW_Î ˙5 D=g

Membership   **Role Manager**   **Personalization**

Site Navigation   **Database Caching**   **Management**
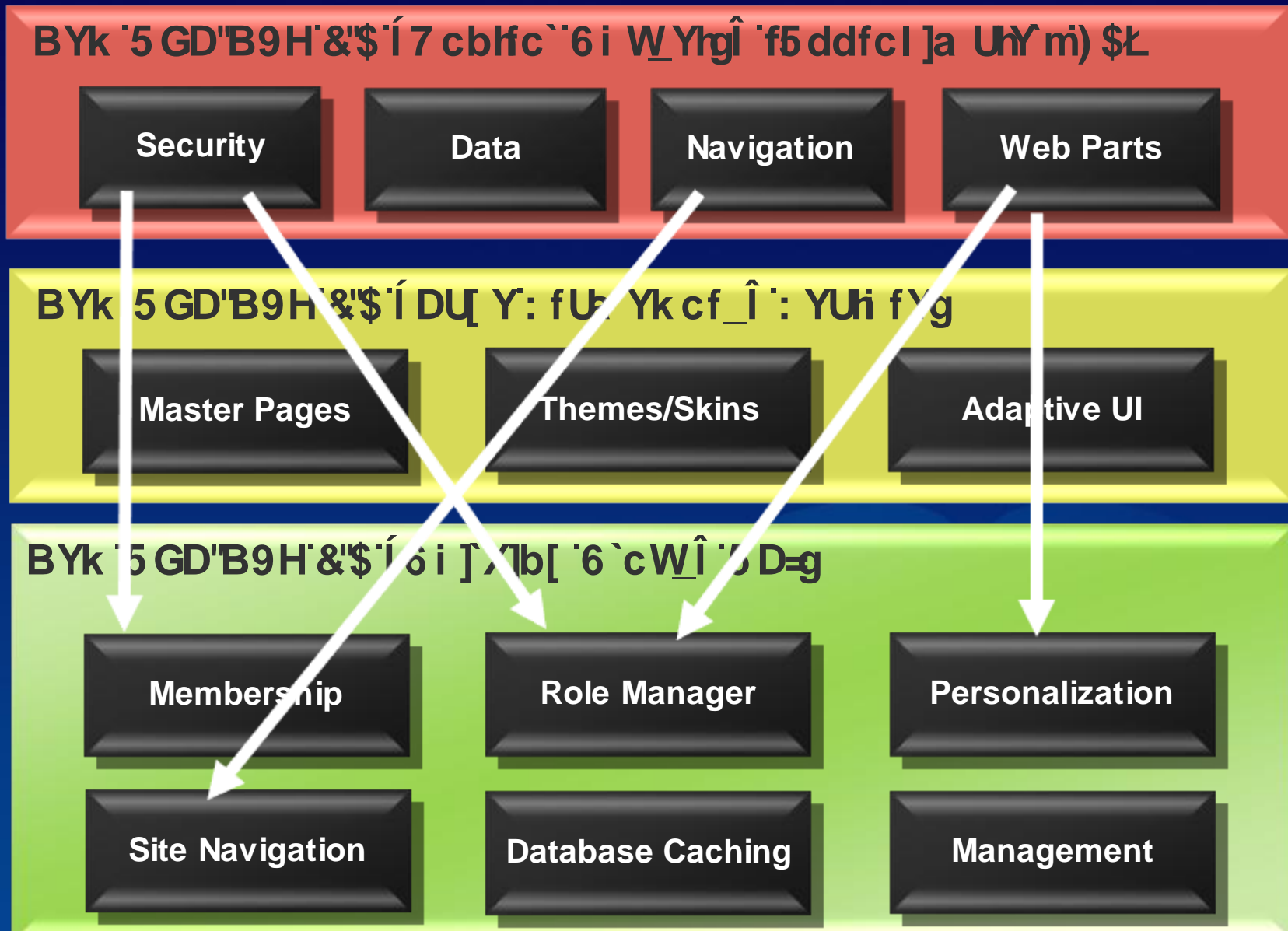
Web Development in Visual Studio
Major Improvements for Visual Studio 2005

FrontPage Server Extensions No Longer Required

˝ Now File System, IIS, FTP *and* FPSE Web Sites

Directory Based Project System Model

˝ Edit any web anywhere (no project file required)
˝ Dramatic performance improvements with large web sites
˝ No longer required to build entire site into a single DLL
˝ Can now re-build just individual pages to test and run

HTML Source Preservation & Formatting Options

˝ Tool will never re-format or mangle your code
˝ Precisely control formatting of your HTML

Web Development in Visual Studio

Major Improvements for Visual Studio 2005

Intellisense Everywhere

˝ Code behind, Inline code, HTML, CSS, ASP.NET page directives, Web.config, XML Files, etc

HTML Tag Navigator and Tag Outlining

˝ Easily view, navigate + explore complex pages

XHTML Support with Pluggable Validation Engine

˝ Real-time validation intellisense as you type HTML
˝ Validation driven intellisense for client-side script

Rich Master Page Design-Time Support

˝ Full WYSIWYG Editing Support

Web Development in Visual Studio

Major Improvements for Visual Studio 2005

Enhanced ASP.NET Code-Behind Page Editing
- ˝ Cleaner and more OO code editing support
- ˝ Removes brittleness of current tool wire-up

ASP.NET Single File Page Editing Support
- ˝ Full intellisense and debugging support

Built-in ASP.NET Web Server (no IIS required)
- ˝ Only allows local browser requests
- ˝ Automatic shutdown when VS IDE is closed

Publish Web
- ˝ Easily synchronize/publish webs to remote servers

User Interface Design

YV@œœqÁ¸^¸Ñ

Master pages	˝ "Visual inheritance" for Web pages ˝ Applied declaratively or programmatically
Themes and skins	˝ Theme controls, pages, and entire sites ˝ Applied declaratively or programmatically
New controls (around 50 in all)	˝ Menus, TreeViews, Wizards, and more

User Interface Design
Site Navigation

Navigation
- ˝ Can be tedious to implement
- ˝ Especially if it relies on client-side script

New controls
- ˝ Simplify site navigation
- ˝ TreeView, Menu, SiteMapPath - Navigation UI
- ˝ SiteMapDataSource - XML site maps

Public API
- ˝ Provides foundation for controls
- ˝ Provider-based for flexibility

Demo – User Interface Design

Build our Master Page → Implement some Navigation

Implement some Navigation ↓ Have a look at Themes
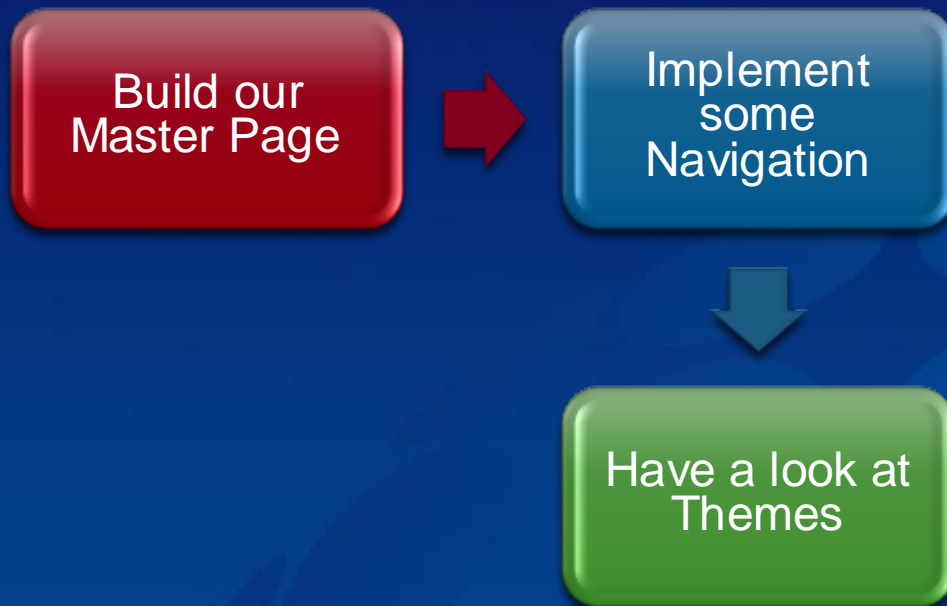
User Interface Design
Validation Groups

Validation control grouping

˝ New ValidationGroup property

Post back when validators in target group satisfied

˝ Implemented by:
˝ Validation controls
˝ Button, LinkButton, and ImageButton

Addresses limitation in ASP.NET 1.x

User Interface Design
Personalisation - Profiles

Store per-user data persistently

- Strongly typed access (unlike session state)
- On-demand lookup (unlike session state)
- Long-lived (unlike session state)
- Supports authenticated and anonymous users

Strong Typing

Accessed through dynamically compiled HttpProfileBase derivatives (HttpProfile)

Provider-based

Flexible data storage

User Interface Design
Web Parts

|	
Framework for building portal-style apps	˝ Patterned after SharePoint Portal Server ˝ System.Web.UI.WebControls.WebParts
Rich UIs	˝ Minimal code ˝ Edit page layout using drag-and-drop ˝ Edit appearance and behavior and more
Seamless personalization	
%Ô[} } ^ & c } • +	˝ Between web parts ˝ Provider ˝ Consumer

Demo – User Interface Design

Validation Groups → Profiles

Profiles ↓ Web Parts

msdn
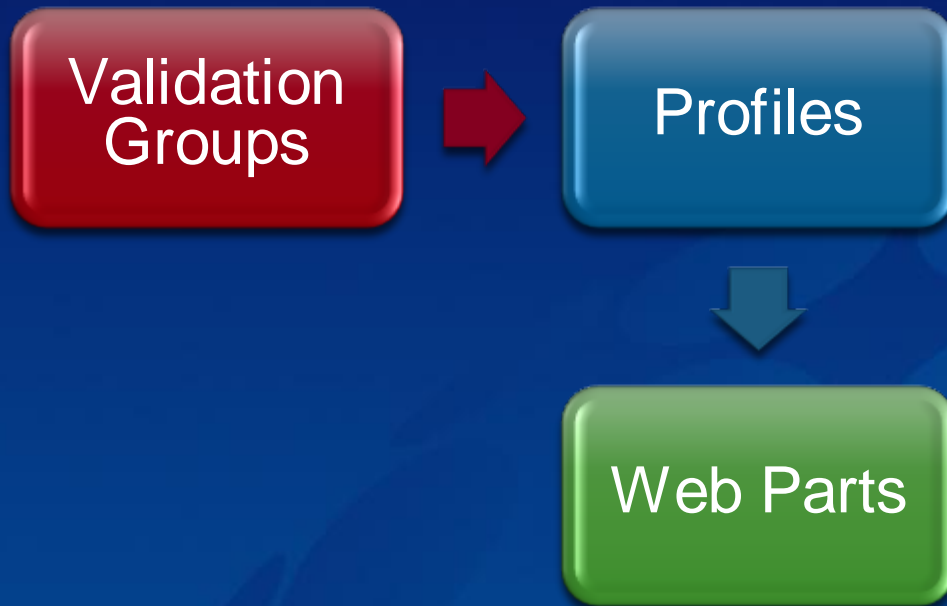
User Interface Design
Localisation

Auto-culture handling
- ˝ Declarative mapping of Accept-Language headers to relevant thread properties

Simplified resource handling
- ˝ Declarative mapping of control properties to ¦^•[˘¦&^•Ã•ą̃*ÅĀ^ÅÅ̃ÑÅ̂¢]¦^••ą̃}•
- ˝ Strongly typed programmatic resource loading

<asp:localize runat="server">

```
<asp:Image ID="Image1" ImageUrl='<%$ Resources:AppResources, FlagImageUrl %>'
 Runat="server"></asp:Image>
<asp:Button ID="Button1" meta:resourcekey="Button1Resource1" runat="server"
 Text="Button" />
```

Data Access

Y©œœqÁP^¸Ñ

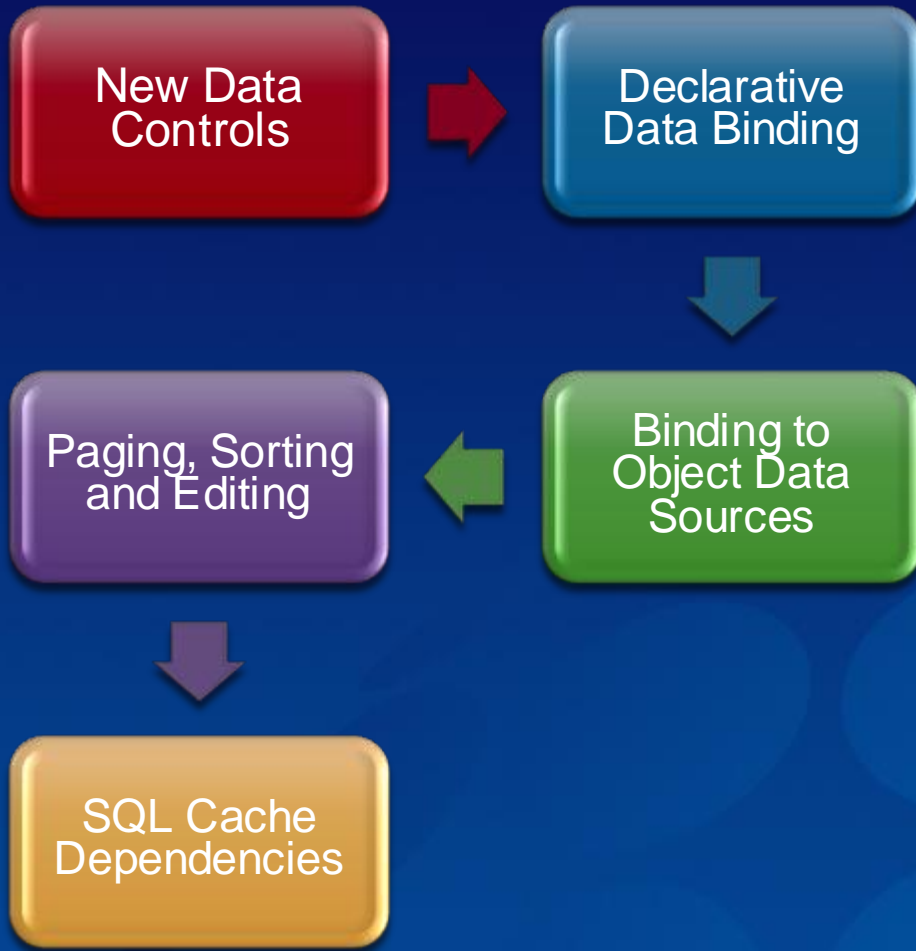|	
Data controls	˝ GridView - Like DataGrid, only better ˝ DetailsView - Companion to GridView
Data source controls	˝ Declarative 2-way data binding
SQL cache dependencies	˝ Key cached items to database entities

Demo – Data Access

New Data Controls

Declarative Data Binding

Binding to Object Data Sources

Paging, Sorting and Editing

SQL Cache Dependencies

Security

Y̌ＶＶＯ＠œœœÁ̂＾¸Ñ

Membership service

″ Service for managing users and credentials

″ Provider-based for flexible data storage

```
if (Membership.ValidateUser (UserName.Text, Password.Text))
    RedirectFromLoginPage (UserName.Text, RememberMe.Checked);
```

Login controls

″ Controls for logging in users, recovering

```
<html>
  <body>
    <form runat="server">
    <asp:Login RunAt="server" />
    </form>
  </body>
</html>
```

Log In

User Name:

Password:

☐ Remember me next time.

Log In

Forgot your password?

Security

Y˜V@œœœÁ·P^¸Ñ

Role Management service	˝ Combine forms authentication and role-based authorization without writing code!
Security Trimming	˝ Restrict navigation based on role
Connection Strings	˝ Store in web.config ˝ Easily encrypt section of web.config ˝ aspnet_regiis -pe "connectionStrings" -app "/MyApp"

Demo – Security

Membership & Logins → Encrypting Config Sections

Configuration and Admin
YV@œœçÁP^¸Ñ

Administrative tools	˝ ASP.NET MMC snap-in ˝ Web Site Administration Tool
Configuration API	˝ Read/write access to configuration settings ˝ Simplified custom configuration sections
Instrumentation	˝ Perf counters, health monitoring, and more

Behind the Scenes
Code Behind 2.0

Hello.aspx

```
<%@ Page CodeFile="Hello.aspx.cs" Inherits="MyPage_aspx" %>

<html>
  <body>
    <form runat="server">
      <asp:TextBox ID="Input" RunAt="server" />
      <asp:Button Text="Test" OnClick="OnTest" RunAt="server" />
      <asp:Label ID="Output" RunAt="server" />
    </form>
  </body>
</html>
```

```
using System;

partial class MyPage_aspx : Page
{
    protected void OnTest (Object sender, EventArgs e)
    { Output.Text = "Hello, " + Input.Text; }
}
```

Hello.aspx.cs

Behind the Scenes
Client Callbacks (AJAX)

"Lightweight" postbacks

- Client-side callback manager transmits asynchronous XML-HTTP requests to server
- Server receives, processes the request, but does <u>not</u> re-render the page
- Callback manager receives response and notifies the client via callback

Requires Internet Explorer 5.0 or higher

Great way to improve UI responsiveness

Behind the Scenes
How Client Callbacks Work

Client   **Server**

Client

Callback Manager

Page

1 Client initiates callback by calling function returned by GetCallback-EventReference

2 Callback manager launches async XML-HTTP call to server

3 Page's RaiseCall-backEvent method is called

5 Client is notified that the call completed and handed the result

4 Callback manager is notified that the call completed
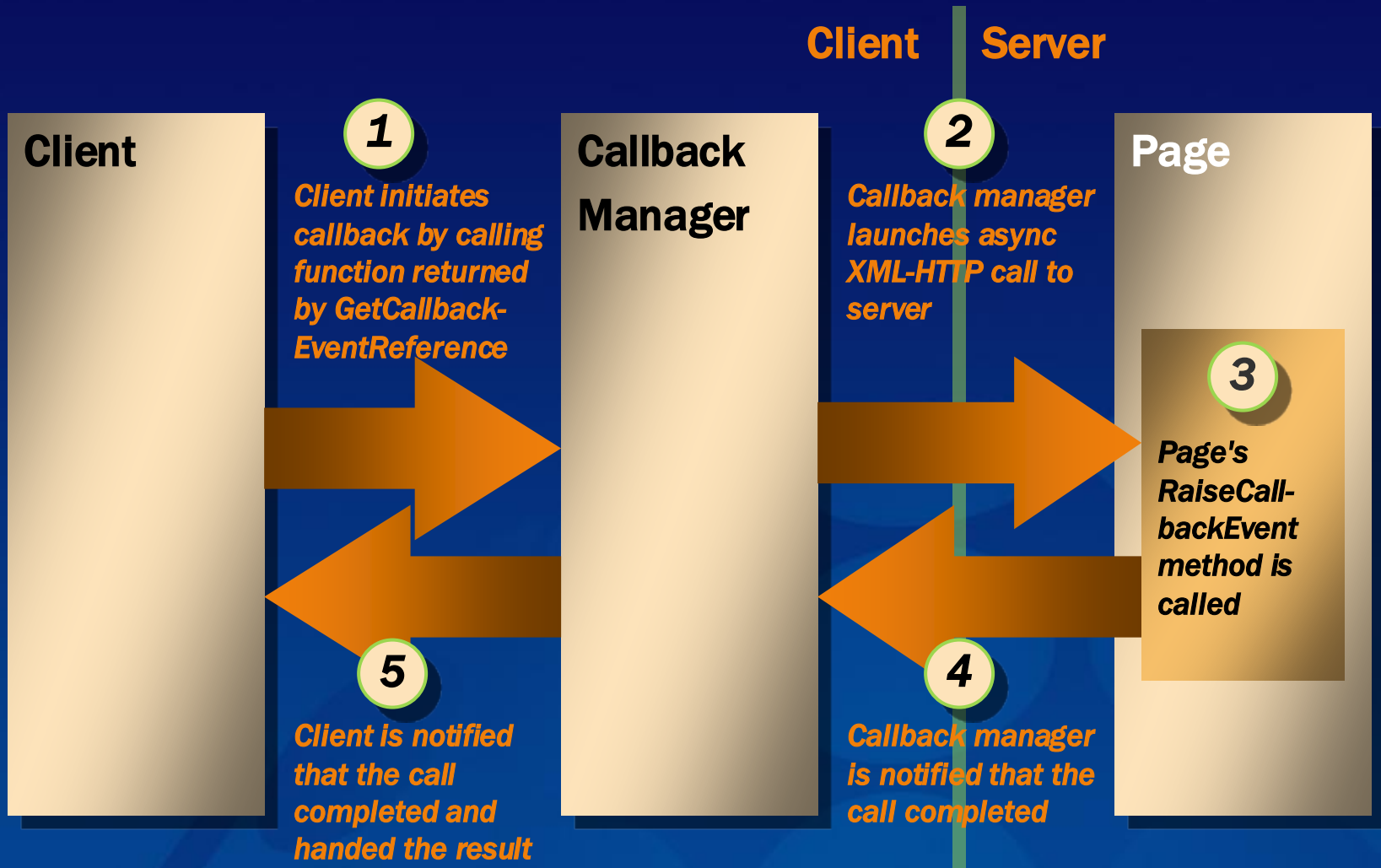
Demo – AJAX

Client
Callbacks

http://msdn.microsoft.com/msdnmag/issues/04/08/CuttingEdge
http://www.schwarz-interactive.de
http://weblogs.asp.net/scottgu/archive/2005/06/28/416185.aspx
http://weblogs.asp.net/bleroy

Further Information

- ASP.NET Developer Center (http://msdn.microsoft.com/asp.net)
- Newsgroups & Forums
  - http://forums.asp.net
  - http://msdn.microsoft.com/newsgroups/default.aspx
- MSDN Magazine Articles (http://msdn.microsoft.com/msdnmag)
  - Filter [}
    - Up Your Site with the Improved View State in ASP.NET 2.0+
- Webcasts (http://msdn.microsoft.com/asp.net/support/multimedia)
  - Learn ASP.NET 2.0 with Jeff Prosise
  - http://www.microsoft.com/events/series/essentialaspnet.mspx
  - http://www.learn2asp.net
- Books
  - ASP.NET v2.0-The Beta Version by Homer, Sussman & Howard (Addison-Wesley)
  - Introducing MS ASP.NET 2.0 by Dino Esposito (MS Press)
  - ASP.NET 2.0 Beta Preview by Bill Evjen (Wrox Press)