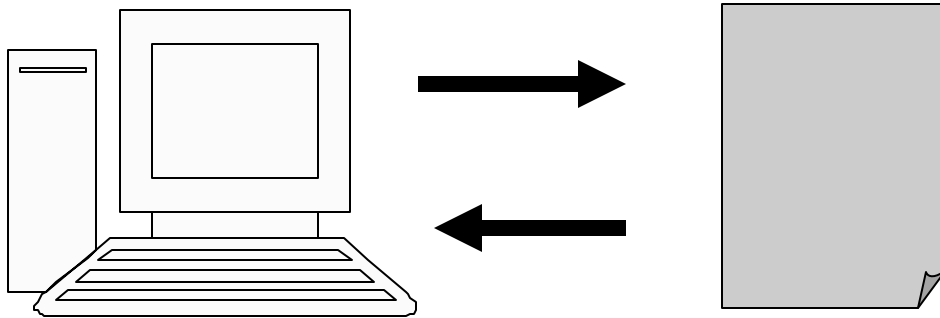


# Introduction to ASP.NET AJAX

Paul Litwin ([paul@deeptesting.com](mailto:paul@deeptesting.com))  
Deep Training and  
Fred Hutchinson Cancer Research Center

# Introduction

AJAX enables you to pass information between a Web browser and Web server without posting to the Web server.



# Ingredients for AJAX BUZZ

## ◆ Nifty Name

AJAX = Asynchronous JavaScript and XML  
(Jesse James Garrett)

## ◆ Cross Browser Support

Internet Explorer, Firefox and Opera

## ◆ Inspiring Sample Application

Google Suggest, Outlook Web Access, etc.

# Ajax Frameworks

- ◆ Dojo – [www.DojoToolkit.org](http://www.DojoToolkit.org)
- ◆ Qooxdoo – [www.Qooxdoo.org](http://www.Qooxdoo.org)
- ◆ YUI – [developer.Yahoo.com/yui](http://developer.Yahoo.com/yui)
- ◆ GWT – Google Web Toolkit
- ◆ Script# -- C# response to GWT
- ◆ Prototype
- ◆ Dean Edward's Base
- ◆ Sarissa
- ◆ Hijax
- ◆ Scriptaculous
- ◆ Lollygag
- ◆ ASP.NET AJAX Extensions

# The Technology

## ◆ Internet Explorer

Uses Microsoft.XmlHttp ActiveX Object  
(IE 7 uses intrinsic object)

## ◆ Firefox, Opera, Safari

Use XMLHttpRequest Intrinsic Browser  
Object

# ASP.NET Ajax Support

- ◆ ASP.NET AJAX Extensions
  - downloadable for VS 2005
  - built into VS 2008
- ◆ ASP.NET AJAX Control Toolkit
  - codeplex project  
<http://www.codeplex.com/AtlasControlToolkit>
- ◆ ASP.NET Futures
  - unsupported preview

# ASP.NET AJAX Extensions

- ◆ ScriptManager
- ◆ ScriptManagerProxy
- ◆ UpdatePanel
- ◆ Timer
- ◆ UpdateProgress

# ScriptManager

- ◆ Injects two JavaScript scripts into page
  - MicrosoftAjax.js
    - base Ajax library
  - MicrosoftAjaxWebForms.js
    - supports partial page rendering (UpdatePanel)
  
- ◆ ScriptManagerProxy
  - used when ScriptManager in Master page

# UpdatePanel Control

- ◆ The big kahuna of ASP.NET Ajax
- ◆ Controls placed inside of this control support async calls (partial rendering)
- ◆ Hijacks post-backs and reroutes them as async calls

# UpdatePanel Control

- ◆ Basic demo of the UpdatePanel control
- ◆ Demos
  - Hello.aspx
  - CascadingDropDownNoAjax.aspx
  - CasasdingDropDownAjax.aspx

# UpdatePanel Properties

- ◆ ChildrenAsTriggers
  - bool which determines whether child controls will create async post-back
    - default=true
- ◆ UpdateMode
  - always (default)
    - will update when any UpdatePanel on page does async post-back
  - conditional
    - only updated by child controls, triggers or Update method

# Triggers

- ◆ Sometimes you wish to trigger a async call from outside of an UpdatePanel control
- ◆ Makes async calls more efficient
- ◆ Two types of triggers
  - Aysnc postback
    - performs async post-back – this is the one you want to use 99% of time
  - Postback
    - don't use unless you want a sync post-back

# Triggers Demo

- ◆ Demo of triggers
  - Triggers.aspx

# Timer Control

- ◆ Performs `window.setInterval()`
- ◆ Can be used inside or outside of `UpdatePanel` control

# Timer Control Demo

- ◆ Timer.aspx

# UpdateProgress Control

- ◆ Displays progress during long-running async call
- ◆ Properties
  - DisplayAfter
    - time in milliseconds to wait before displaying progress indicator
      - by default = 500 (half second)
  - DynamicLayout
    - display:none or display:hidden
  - AssociatedUpdatePanelId

# UpdateProgress Control

- ◆ Free site to create animated gifs for progress meters:
  - <http://www.ajaxload.info/>
- ◆ Examples of UpdateProgress Control
  - ShowUpdateProgress.aspx
  - TriggersProgress.aspx

# Server-Side Page Execution Life Cycle

- ◆ These events occur for both sync and async post-backs
  - PreInit
  - Init
  - Load
  - PreRender
- ◆ `ScriptManager.IsInAsyncPostBack`
  - Use to detect async post-back

# Client-Side Page Execution Life Cycle

- ◆ Events
  - initializeRequest
  - beginRequest
  - pageLoading
  - pageLoaded
  - endRequest
- ◆ Canceling and aborting requests
  - use initializeRequest event handler

# Multiple Async Postbacks

- ◆ UpdatePanel doesn't support multiple async postbacks
- ◆ By default, last postback will abort previous postbacks

# Client-Side Page Execution Life Cycle Demos

- ◆ UpdatePanelCancel.aspx
  - demonstrates how to cancel last postback
- ◆ UpdatePanelAbort.aspx
  - demonstrates how to cancel a pending postback

# Performance Considerations

- ◆ ViewState
- ◆ Whole-page execution life-cycle executes unless you use `ScriptManager.IsInAsyncPostBack` property
- ◆ Consider using pure client-side Ajax

# Handy Ajax Debugging Tools

- ◆ Fiddler (<http://fiddler2.com>)
  - works with IE, Firefox, etc.
  - Tip: Add dot (.) after localhost so that fiddler sees the page
    - E.g., <http://localhost.:8345/mypage.aspx>
- ◆ Firebug (<https://addons.mozilla.org/en-US/firefox/addon/1843>)
  - Firefox only

# Performance Demos

- ◆ ViewStateEnabled.aspx and ViewStateDisabled.aspx
  - Use fiddler and SQL Profiler to see data being passed around
  - See table on next slide

# Measuring Bandwidth Usage with Fiddler: An Example

Operation	No UpdatePanels; ViewState Enabled		Four UpdatePanel controls with ViewState Enabled		Four UpdatePanel controls with ViewState Disabled	
	Request Bytes	Response Bytes	Request Bytes	Response Bytes	Request Bytes	Response Bytes
Selecting a record	7,922	17,654	7,718	16,552	2,132	11,813
Saving unchanged parent record	7,785	17,994	7,686	8,972	2,618	3,889
Saving unchanged child record	7,693	16,870	7,719	9,549	2,481	4,786

# ASP.NET Ajax Toolkit

- ◆ 34 Controls
- ◆ Updated every couple of months or so
- ◆ Source-code available
- ◆ A community project
  - hosted on CodePlex
  - Not an officially-supported Microsoft product (though Microsoft is heavily involved)

# ASP.NET Ajax Toolkit

- ◆ Most of toolkit controls are extender controls
- ◆ An extender is a control which extends the functionality of another control
- ◆ VS 2008 integrates extender controls into IDE designer

# Toolkit Controls (2007-11-19 release)

- ◆ Accordion
- ◆ AlwaysVisibleControl
- ◆ Animation
- ◆ AutoComplete
- ◆ Calendar
- ◆ CascadingDropDown
- ◆ CollapsiblePanel
- ◆ ConfirmButton
- ◆ DragPanel
- ◆ DropDown
- ◆ DropShadow
- ◆ DynamicPopulate
- ◆ FilteredTextBox
- ◆ HoverMenu
- ◆ ListSearch
- ◆ MaskedEdit
- ◆ ModalPopup
- ◆ MutuallyExclusiveCheckBox
- ◆ NoBot
- ◆ NumericUpDown
- ◆ PagingBulletedList
- ◆ PasswordStrength
- ◆ PopupControl
- ◆ Rating
- ◆ ReorderList
- ◆ ResizableControl
- ◆ RoundedCorners
- ◆ Slider
- ◆ SlideShow
- ◆ Tabs
- ◆ TextBoxWatermark
- ◆ ToggleButton
- ◆ UpdatePanelAnimation
- ◆ ValidatorCallout

# Toolkit

- ◆ You can download VS 2005 and VS 2008 versions with or without source code
- ◆ To add toolkit to toolbox
  1. On toolbar, Add Tab
  2. Choose Items...
  3. Browse... to  
SampleWebSite\bin\AjaxControlToolkit.dll
- To get easy access to docs
  - Point share to SampleWebSite folder

# Client-Side Ajax

- ◆ SOAP vs. JSON
- ◆ SOAP = Simple Object Access Protocol
  - XML is not simple and not concise
- ◆ JSON = JavaScript Object Notation
  - lightweight data-interchange format
  - lean and mean

# Client-Side Ajax

- ◆ Calling services from the client using JSON
- ◆ Web Methods
  - Web services
  - Page methods
- ◆ Application Services
  - Authentication service
  - Role service (VS 2008 only)
  - Profile service

# Registering a Service Reference for a Web Service Call

```
<asp:ScriptManager ID="ScriptManager1"
runat="server">
  <Services>
    <asp:ServiceReference
      Path="~/Services/QuotationService.asmx" />
  </Services>
</asp:ScriptManager>
```

# Calling Web Service from Client

```
[ScriptService]
public class QuotationService : System.Web.Services.WebService
{...
```

```
<script type="text/javascript">
function pageLoad()
{
    QuotationService.GetQuote(getQuoteSuccess, getQuoteFail);
}
function getQuoteSuccess(result)
{
    $get("spanQuote").innerHTML = result;
}
function getQuoteFail(error)
{
    alert(error.get_message());
}
</script>
```

# Registering a Service Reference for a Page Method Call

```
<asp:ScriptManager ID="ScriptManager1"  
  EnablePageMethods="true"  
  runat="server" />
```

# Calling Page Method from Client

```
[System.Web.Services.WebMethod]  
public static string GetQuote()  
{...
```

```
<script type="text/javascript">  
function pageLoad()  
{  
    PageMethods.GetQuote(getQuoteSuccess, getQuoteFail);  
}  
function getQuoteSuccess(result)  
{  
    $get("spanQuote").innerHTML = result;  
}  
function getQuoteFail(error)  
{  
    alert(error.get_message());  
}  
</script>
```

# Calling Web Service from Client Example

- ◆ ShowQuoteService.aspx
  - Illustrates use of both Page Method call and Web Service call (to QuoteService.asmx) from client

# Thank You!

- ◆ Paul Litwin
  - [paul@deeptesting.com](mailto:paul@deeptesting.com)
  - [www.deeptesting.com](http://www.deeptesting.com)
  
- ◆ I will post the lecture demos after workshop at
  - [www.deeptesting.com/litwin](http://www.deeptesting.com/litwin)