ASP.NET AJAX 소개

• ASP.NET AJAX 소개

ASP.NET AJAX Extension

ASP.NET AJAX Control Toolkit

- 1 -



• AJAX(에이잭스) 란?

AJAX = Asynchronous JavaScript and XML Term coined by Jesse James Garrett in Feb. 2005 "Ajax: A New Approach to Web Applications"

- 웹 상에서의 비 동기 호출 기법
- 내부적으로 XMLHTTP 이용
 - Microsoft.XMLHTTP (IE 6.0 이하)
 - Window.XMLHttpRequest (IE 7 이상 및 다른 브라우저)
- 기존 비 동기 호출 기술의 재 정리
- 기대 효과
 - 네트워크의 사용량 감소
 - 서버의 CPU 사용량 감소
 - 응용 프로그램 자원의 효과적인 사용

XMLHTTP

- Introduced in 1999 with Internet Explorer 5
 - ActiveX object supporting callbacks to Web server
 - Created for benefit of Outlook Web Access (OWA)
- · Later adopted by Firefox, Safari, and others
 - Implemented as native object rather than ActiveX
 - Currently being standardized by W3C
 - http://www.w3.org/TR/XMLHttpRequest/
- Supported by approx. 99% of browsers used today
 - Approx. 85%-95% have JavaScript enabled
- Chief enabling technology behind AJAX

기존 웹 통신 방식의 문제점

- 반복적인 Roundtrip
- 서버 중심적 설계
- 클라이언트와 서버가 함께 통합되어있지 않다



기존 웹 응용 프로그램 모델 (동기적 패턴, Synchronous)



• 요청을 처리하기 위한 Postback은 이제 필요치 않다



Ajax 웹 응용 프로그램 모델 (비 동기적 패턴, Asynchronous)

내부 처리 프로세스 비교



전형적인 웹 응용 프로그램 모델



AJAX 웹 응용 프로그램 모델

- 6 -

ASP.NET AJAX

ASP.NET AJAX provides:

- Partial-page updates
- Data integration by using Web services
- Support for the most commonly used browsers
- UI elements that are familiar to users
- Integration with ASP.NET security features
- Increased efficiency because a large amount of the processing can be carried out in the browser
- A framework that simplifies ASP.NET AJAX operations

Architecture of ASP.NET AJAX



Microsoft AJAX Library

Components Non-visual Components, Behaviors, Controls

Browser Compatibility

Support for Microsoft Internet Explorer, Mozilla Firefox, Apple Safari

Networking

Asynchronous Requests, XML & JSON Serialization. Web & Application Services

Core Services

JavaScript Base Class Extensions, Type System, Events, Serialization



Server

ASP.NET AJAX Extensions

Script Support Localization, Globalization, Debugging, Tracing

Web Services Proxy Generation, Page Methods, XML & JSON Serialization

Application Services Authentication & Profile

Server Controls ScriptManager, UpdatePanel, UpdateProgress, Timer

Using the ASP.NET AJAX Extensions

- What Are the ASP.NET AJAX Extensions?
 - Contain the core server-side components
 - Extend ASP.NET server-side control framework
 - Integrate with Visual Studio 2008
 - Add controls to the Visual Studio Toolbox

ASP.NET AJAX Server Controls

- ScriptManager
 - Manages script resources for client components
- UpdatePanel
 - Enables a refresh of part of a Web page
- UpdateProgress
 - Provides the status of a partial-page update
- Timer
 - Performs postbacks at defined intervals



ScriptManager

- Starting point for ASP.NET AJAX pages
- What does ScriptManager do?
 - Downloads JavaScript files to client
 - Enables partial-page rendering using UpdatePanel
 - Provides access to Web services via client-side proxies
 - Manages callback timeouts and provides error handling options and infrastructure
 - Provides registration methods for scripts
 - Enables ASP.NET AJAX localization support
- Every page requires <u>one</u> ScriptManager instance!

ScriptManager Schema

```
<asp:ScriptManager ID="ScriptManager1" Runat="server"</pre>
 EnablePartialRendering="true|false"
 EnablePageMethods="true|false"
 AsyncPostBackTimeout="seconds"
 AsyncPostBackErrorMessage="message"
 AllowCustomErrorsRedirect="true|false"
  OnAsyncPostBackError="handler"
 EnableScriptGlobalization="true|false"
  EnableScriptLocalization="true|false"
  ScriptMode="Auto|Inherit|Debug|Release"
  ScriptPath="path">
  <Scripts>
    <!-- Declare script references here -->
  </Scripts>
  <Services>
    <!-- Declare Web service references here -->
  </Services>
</asp:ScriptManager>
```

UpdatePanel

- Partial-page rendering in a box
 - Clean round trips to server and flicker-free updates
 - Requires no knowledge of JavaScript or AJAX
- Leverages client-side PageRequestManager class
 - EnablePartialRendering="true" in ScriptManager
- Supports explicitly defined triggers
 - By default, postbacks from all controls in an UpdatePanel are converted into async callbacks
 - Triggers expand (or shrink) postback->callback scope
- Works in virtually all scenarios

UpdatePanel Schema

Triggers Example

```
<asp:UpdatePanel ID="UpdatePanel1" Runat="server"
UpdateMode="Conditional">
<Triggers>
<asp:AsyncPostBackTrigger ControlID="Button1" />
<asp:AsyncPostBackTrigger ControlID="TreeView1"
EventName="TreeNodeExpanded" />
<asp:AsyncPostBackTrigger ControlID="TreeView1"
EventName="TreeNodeCollapsed" />
<asp:PostBackTrigger ControlID="Button2" />
</Triggers>
<ContentTemplate>
...
</ContentTemplate>
```

Demo1: Implementing the UpdatePanel Control

- 1. Create a new ASP.NET Web site.
- 2. Open the default.aspx page in Design view.
- 3. Add a ScriptManager and UpdatePanel to the page.
- 4. In the UpdatePanel, add a TextBox, Button, Label control to the form.
- 5. Set the Text property of the button to the value Inside Panel and create a Click event I
 - Label1.Text = "Hello," + TextBox1.Text;
- 6. On the form, outside the UpdatePanel, add a second Button control.
- 7. Set the **Text** property of the second button to the value **Outside Panel**.
- 8. Save the default.aspx page, and then view the page in the browser.
- 9. Click each button in turn and verify that the button inside the **UpdatePanel** does not cause a page postback but the button outside of the pan el does cause a postback.
- 10.Close the browser window.
- 11.In the default.aspx page, delete the Inside Panel button.
- 12.On the form, inside the UpdatePanel, add a Label control.
- 13.Double-click Outside Panel to create a Click event handler.
- 14.Add code to the Click event handler to set the Text of the Label to the string Current Time: followed by the current date and time.
 - Label2.Text = "Current Time: " + DateTime.Now.ToString();
- 15.Switch to Design view of the default.aspx page.
- 16.View the properties of the UpdatePanel control. On the Triggers property click the ellipsis.
- 17.In the UpdatePanelTrigger Collection Editor dialog box, click Add.
- 18.Set the ControlID property of the new trigger to Button2, and then click OK.
- 19.Save the Web site, and then view the default.aspx page in the browser.
- 20.Click **Outside Panel** several times and verify that the content in the label changes but a page refresh does not occur.

	Default.aspx.cs Default.aspx 시작 페이지								
F	ScrintManager - ScrintManager1 asp:updatepanel#UpdatePanel1 Inside Panel Label Label								
	Outside Panel								

- 16 -

Triggers

- AsyncPostBackTrigger
 - Converts postbacks into asynchronous callbacks
 - Typically used to trigger updates when controls outside an UpdatePanel post back and fire events
 - If ChildrenAsTriggers="false", can be used to specify which controls inside UpdatePanel should call back rather than post back
- PostBackTrigger
 - Allows controls inside an UpdatePanel to post back
 - Typically used to allow certain controls to post back when ChildrenAsTriggers="true"

Periodic Updates

- Combine UpdatePanel with Timer control to implement pages that perform periodic updates
- Use Timer control Tick events as triggers

```
<asp:Timer ID="Timer1" Runat="server" Interval="5000"
OnTick="OnTimerTick" />
...
<asp:UpdatePanel UpdateMode="Conditional" ...>
<Triggers>
<asp:AsyncPostBackTrigger ControlID="Timer1" />
</Triggers>
...
</asp:UpdatePanel>
```

UpdateProgress

- Companion to UpdatePanel controls
- Displays custom template-driven UI for:
 - Indicating that an async update is in progress
 - Canceling an async update that is in progress
- Automatically displayed when update begins or "DisplayAfter" interval elapses

```
<asp:UpdateProgress ID="UpdateProgress1" Runat="server"
DisplayAfter="milliseconds"
DynamicLayout="true|false"
AssociatedUpdatePanelID="UpdatePanelID">
<ProgressTemplate>
<!-- Declare UpdateProgress UI here -->
</ProgressTemplate>
</asp:UpdateProgress>
```

- 20 -



Canceling an Update

```
<asp:UpdateProgress DisplayAfter="500" ...>
  <ProgressTemplate>
    <b>Working...</b>
    <asp:Button ID="CancelButton" Runat="server" Text="Cancel"</pre>
      OnClientClick="cancelUpdate(); return false" />
  </ProgressTemplate>
</asp:UpdateProgress>
<script type="text/javascript">
function cancelUpdate()
ł
   var obj = Sys.WebForms.PageRequestManager.getInstance();
    if (obj.get_isInAsyncPostBack())
        obj.abortPostBack();
</script>
```

- 22 -

Demo2 : Scott's ToDo List

http://www.asp.net/learn/3.5-videos/video-362.aspx

ØL	Intitle	d Page	- Windows Internet Explor				
G	0	🗸 🙋 ht	tp://localhost:2270/ToDoList-CS/	iault,aspx 💌 🗲 🗙 Live Search 🖉	· •	d	ho Tasks: Fil01
į I	ŀ일(Ē)	편집(<u>E</u>)	· 보기(⊻) 즐겨찾기(<u>A</u>) 도구(도움말(<u>H</u>)			Q 01
🖈	\$	🏉 Untitle	ed Page	📄 🔹 🔊 🔹 🖶 🖬 🔂 🖓 मागत्रा (२) 🗸 🍈 ६२ (०) 🗸	**	10	E VI
	Sco	ott's	s ToDo List			• (s	Name Complete
2	Status:	Active	×			—	
i	ΓoDo	List Item	S				
		TaskId	Name	Complete		P	볼 속성
	편십	2	Build ToDo List Demo			F	
	편집	3	Build C# version of ToDo List			ł	
	편집	4	More Work				기보값 또는
							데이터 형식
	😳 Ad	ld a New	v Task			E	∃ 테이블 디자이
	Name	е					DTS 개비답
	Comp	olete				Ç	비 ID 사양 고 계산 열 사양 ID 사양
	삽밉	취소					
완료				Succel intranet 🔍 100% •	•		

dbo.Tasks: 테이블₩DATABASE.MDF) StyleSheet.css 〒 ×								
	열 이름	데이터 형식	Null 허용					
₽ ₿	TaskId	int						
	Name	varchar(100)	 Image: A start of the start of					
	Complete	bit	 Image: A set of the set of the					
9	별 속성							
	8∎ 2↓ =							
	Null 허용	아니요		<u>^</u>				
	기본값 또는 바인딩							
	데이터 형식	int						
1	크 테이블 디자이너							
	DTS 74418	아니오		_				
1	표 ID 사양	예						
	그 계산 열 사양			<u> </u>				
	ID 사양							

Using the ASP.NET AJAX Control Toolkit

- Shared source project between the ASP.NET AJAX community and Microsoft
- Built upon the ASP.NET 2.0 AJAX Extensions
- Contains a collection of web-client components
- Provides an infrastructure to write reusable and extensible ASP.NET AJAX extenders and controls

Accessing the ASP.NET AJAX Control Toolkit

1 Download the latest release of the Control Toolkit

2 Unpack the Control Toolkit into a folder

3 Open or create a Web application

Add the Control Toolkit items to the Toolbox

5 Install the AJAX Control Toolkit templates

http://go.microsoft.com/fwlink/?LinkId=111447 http://go.microsoft.com/fwlink/?LinkId=111448

Demo: Implementing an ASP.NET AJAX Extender Control

- 1. Create a new ASP.NET Web site.
- 2. In the Toolbox, create a tab named AJAX Control Toolkit.
- 3. Add the items from the file E:\Democode\AjaxControlToolkit.dll to the AJAX Control Toolkit tab.
- 4. Show the new Toolbox items.
- 5. Open the default.aspx Web form in Design view.
- 6. On the form, add a Label and Button control to the form.
- 7. Create a Click event handler for the button.
- 8. In the event handler, add code to set the **Text** property of the label to the string **Button click occurred**.
- 9. View the default.aspx page in the browser, and then click the button on the page. Verify that the label text changes, and then close the browser window.
- 10.In the Toolbox, on the AJAX Extensions tab, drag a ScriptManager and drop it at the top of the form.
- 11.In the Toolbox, on the AJAX Control Toolkit tab, drag a ConfirmButtonExtender and drop it onto Button1 on the form.
- 12.In the Properties window, set the **ConfirmText** property of the **ConfirmButtonExtender** control to **Continue?**.
- 13.View the default.aspx page in the browser.
- 14.Click the button and verify that the confirmation dialog box appears.

ASP.NET AJAX Control Toolkit

- ASP.NET Extension과는 별도로 다운로드
 - 무료로 제공되는 AJAX 지원 컨트롤 라이브러리
 - <u>http://www.asp.net/</u>에서 다운로드 가능
- 다달이 꾸준히 업데이트 중
 - ASP.NET Extension 의 업데이트 주기와는 무관하게 업데이트
- 공동 제작 소스 모델을 사용하여 개발
 - Microsoft 개발자 및 비-Microsoft 개발자 참여
 - 모든 소스는 자유롭게 사용 가능
- 이미 상당히 멋진 컨트롤들이 포함되어 있음
 - 목표는 50-100개의 훌륭한 컨트롤들을 제공하는 것

현재 제공되는 컨트롤들

DEMO : http://www.asp.net/ajax/ajaxcontroltoolkit/samples/

Accordion, AlwaysVisibleControl, Animation, CascadingDropDown, CollapsiblePanel, ConfirmButton, DragPanel, DropShadow, FilteredTextBox DynamicPopulate, HoverMenu, ModalPopup, NoBot, NumericUpDown, PagingBulletedList, PasswordStrength, PopupControl, Rating, ReorderList, ResizableControl, RoundedCorners, Slider, TextBoxWatermark ToggleButton, UpdatePanelAnimation



<u>색상과 폰트 크기는 아무 의미 없음</u>



- 다중 Pane(AccordionPane)으로 구성되며, 사용자 클릭에 대해 한번에 하나의 Pane을 슬라 이딩, 페이드 효과로 보여주는 웹 컨트롤
- 포스트백 사이에 기존 선택 상태를 유지



AlwaysVisibleControlExtender

- 페이지가 스크롤 되거나, 크기 조정되어도 지정한 위치에 컨트롤이 항상 고정적으로 나타나도록 하는 컨트롤 확장
- 모든 ASP.NET 컨트롤에 적용 가능



AnimationExtender

• 기존 페이지에서 강력한 애니메이션 프레임워크를 선언적인 방식으로 쉽게 사용할 수 있도록 하는 extender



CascadingDropDown

- DropDownList에 부착하여, 데이터 자동 채움을 구현하게 하는 Extender
- 각 DropDownList의 선택이 변경될 때마다, 지정된 웹 서비스를 호출하여 가져온 데이터로 다음 DropDownList 채움





- 접힘 가능한 영역을 페이지에 추가하는 유연한 Extender
- ASP.NET Panel 컨트롤을 대상으로 하며, Accordion 컨트롤과 유사하게 동작
- 포스트백 사이에 기존 접힘 상태를 유지



DragPanel

- 컨트롤에게 드래그 기능을 적용해주는 Extender
- ASP.NET Panel 컨트롤을 대상으로 함



ModalPopup

- 특정 컨텐츠를 modal 방식으로 출력 가능케 하는 Extender
- 모달 영역 외에는 사용자 상호작용을 불가능하게 함



PagingBulletedList

• ASP.NET BulletedList 컨트롤에 부착하여, 클라이언트 측에서의 정렬된 페이징을 제공





• 대부분 팝업 윈도우는 UpdatePanel 내에 위치시켜 AJAX식으로 서버와 통신하도록 구성



Slider

- 사용자가 한정된 범위 내에서 숫자 값을 선택할 수 있도록 TextBox 서버 컨트롤을 슬라이더로 변 환하는 Extender
- 슬라이더는 수평이나 수직적으로 표현 가능



ASP.NET AJAX 기반의 3rd Party 컴포넌트

- Componentart사의 Web.UI
- ASP.NET AJAX 기반의 최초 상용 컨트롤 라이브러리
- ASP.NET AJAX Extensions와 완전하게 통합
- http://www.componentart.com/
- 차세대 웹 UI의 모습을 살펴볼 수 있다



참고자료

- <u>http://www.asp.net/ajax/</u>
- <u>http://www.asp.net/learn/</u>
- <u>http://weblogs.asp.net/scottgu</u>
- <u>http://www.nikhilk.net/</u>
- <u>http://www.dotnetslackers.com/</u>
- <u>http://www.taeyo.pe.kr</u>
- <u>http://www.hoons.kr</u>