Partial-Page Rendering and AJAX

Introduction

Partial-page rendering removes the need for the whole page to be refreshed as the result of a postback. Instead, only individual regions of the page that have changed are updated. As a result, users do not see the whole page reload with every postback, which makes user interaction with the Web page more seamless. Microsoft ASP.NET AJAX enables you to add partial-page rendering to new or existing ASP.NET Web pages without writing client script.

Scenarios

ASP.NET AJAX enables you to extend existing ASP.NET 2.0 applications and to develop new ones that incorporate AJAX (Asynchronous JavaScript and XML) functionality. Use ASP.NET AJAX when you want to do the following:

- Improve the user experience with Web pages that are richer, that are more responsive to user actions, and that behave like traditional client applications.
- Reduce full-page refreshes and avoid page flicker.
- Enable cross-browser compatibility without writing client script.
- Perform AJAX-style client/server communication without writing client script.
- Use the controls and components from the ASP.NET AJAX Control Toolkit.
- Develop custom Microsoft ASP.NET AJAX controls.

Partial-Page Rendering Features

Partial-page rendering relies on server controls in ASP.NET AJAX and on client functions in the Microsoft AJAX Library. You do not have to use the Microsoft AJAX Library to enable partial-page rendering, because this functionality is provided automatically when you use the ASP.NET AJAX server controls. However, you can use the APIs exposed in the client library for additional AJAX functionality.

Background

Typical Web pages built with ASP.NET Web server controls perform postbacks initiated by a user action on the page, such as clicking a button. In the response, the server renders a new page. Frequently this re-renders controls and text that did not change between postbacks.

With ASP.NET AJAX partial-page rendering, you can refresh individual regions of the page asynchronously and make the page more responsive to the user. You can implement partial-page rendering using ASP.NET Web server controls, and optionally write client script that uses the APIs in the Microsoft AJAX Library.

Server Controls for Partial-Page Updates

To add AJAX functionality to ASP.NET Web pages, you identify individual sections of the page that you want to update. You then put the content of these sections into UpdatePanel controls. The contents of an UpdatePanel control can be HTML or other ASP.NET controls. You can add an UpdatePanel control to the page as you would any other control. For example, in Visual Studio you can drag it from the toolbox to the Web page, or you can add it by using declarative markup in the page. The following example shows the markup for an UpdatePanel control.

C# Example

```
<asp:UpdatePanel ID="UpdatePanel1" runat="server">

<ContentTemplate>

<!-- Place updatable markup and controls here. -->

</ContentTemplate>

</asp:UpdatePanel>
```

Using Client Script for Partial-Page Updates

The ECMAScript (JavaScript) PageRequestManager class in the Microsoft AJAX Library supports partial-page updates. It runs in the browser to manage the response to asynchronous postbacks and to update content in individual regions. You do not have to do anything to enable this functionality. It occurs automatically when you add one or more UpdatePanel controls and a ScriptManager control to the page.

You can also use JavaScript and the PageRequestManager class to customize partial-page updates on a page. For example, you can write script to give precedence to a specific asynchronous postback if more than one is underway. You can also enable users to cancel postbacks that are in progress.

The following example shows client script that provides an event handler that is called when the page finishes loading.

C# Example

```
<script type="text/javascript" language="javascript">

var prm = Sys.WebForms.PageRequestManager.getInstance();

prm.add_pageLoaded(PageLoadedEventHandler);

function PageLoadedEventHandler() {

 // custom script

}

</script>
```

Enabling Partial-Page Rendering Support

You enable or disable partial-page rendering for a page by setting the EnablePartialRendering property of the ScriptManager control. You can also specify whether partial-page rendering is supported for a page by setting the SupportsPartialRendering property of the ScriptManager control. If you do not set the SupportsPartialRendering property and if the EnablePartialRendering property is true (which is the default), the capabilities of the browser are used to determine whether partial-page rendering is supported.