Legacy Controls NOTICE

This article references our legacy Web Forms Web Viewing controls (WebImageViewer, WebAnnotationViewer, WebThumbnailViewer). It is preserved for archival purposes, but support strongly recommends using our modern HTML5 web controls: WebDocumentViewer, WebDocumentThumbnailer instead)

INFO: WebDocumentViewer Whitepaper - Getting Started With Web Viewing

Main Article Content

The WebImageViewer control provides the ability for an ASP.NET Page object to receive an event when a client side script requests a remote invocation. When JavaScript performs a RemoteInvoke, an http POST is performed to send parameters back to the server side. To get similar capabilities without the complexity of events, see the section about remotely invoking Page() methods.

Remote Invoke Event Arguments

A handler for a Remote Invoke Event receives an object of type RemoteInvokeEventArgs. This object contains three properties: Page, Parameters, and ReturnValue.

Property	Description
Page	Object of type System.Web.UI.Page that contains the WebImageViewer that received the event
Parameters	Object of type System.Collection.Specialized.NameValueCollection which contains all parameters provided by the POST.
ReturnValue	ArrayList which is used by event handlers to pass information back. Typically the return value is a one element array list containing a string that represents the return value of the method which has been remotely invoked

Parameters

In addition to other keys provided to the WebImageViewer, there is a key with the name atala_rm.This key is associated with the name of the method requested to be invoked. To retrieve the method name from the Parameters property, do the following:

```
string methodName = eventArgs.Parameters.Get("atala_rm");
```

For each parameter passed in there is a key with a name that follows this pattern:

```
tala_ra<type><parameter number>
```

<type> is s, b, or n, depending on whether or not this parameter is a string, a bool, or a number, respectively. The table that follows summarizes this relationship.

<type>	**parameter**
s	string
b	bool
n	number

<parameter number> is an integer starting from 0 that corresponds to the position of the parameter in the array passed into the JavaScript RemoteInvoke().

Parameters can be retrieved with code like this:

C#

```
int i = 0; ArrayList params = new ArrayList(); ArrayList types = new ArrayList(); while
(true) { string val; val = eventArgs.Parameters.Get("atala_ras" + i); if (val != null) {
types.Add(typeof(string)); params.Add(val); i++; continue; } val =
eventArgs.Parameters.Get("atala_ran" + i); if (val != null) { types.Add(typeof(double));
params.Add(Convert.ChangeType(val, typeof(double))); i++; continue; } val =
eventArgs.Parameters.Get("atala_rab" + i); if (val != null) { types.Add(typeof(bool));
params.Add(Convert.ChangeType(val, typeof(bool))); i++; continue; } break; }
```

Writing an Event Handler

To write a RemoteInvoke event handler, first create the method which will receive the event. This method must take an object and a RemoteInvokeEventArgs and have no return type. Such an event handler might look like the example shown below.

Example

RemoteInvoke Event Handler

C#

```
privatevoid HandleRemoteInvoke(object sender, RemoteInvokeEventArgs args) { // your event
handling code goes here }
```

To install the event handler, tell the WebImageViewer to add your event handler into its chain as shown in the example below.

C#

```
ebImageViewer1.RemoteInvoke += new RemoteInvokeHandler(this.HandleRemoteInvoke);
```

See Also

HOWTO: Remotely Invoke ASP.NET Page Methods (Legacy Web Controls)

Original Article:

Q10361 - HOWTO: Work with Remote Events

Atalasoft Knowledge Base
https://www.atalasoft.com/kb2/KB/50126/HOWTO-Work-with-Remote-Events-Legacy...