Walkthrough: Creating a Basic Web Page with IronPython

Introduction

This walkthrough provides you with an introduction to IronPython for ASP.NET. It guides you through creating a simple page in Microsoft Visual Studio, adding controls, and adding event handlers in IronPython.

Tasks illustrated in this walkthrough include:

- Adding controls to the default page.
- Adding IronPython event handlers in a separate code file.
- Adding a second page with event-handling code in the page.

Prerequisites

In order to complete this walkthrough, you will need:

- Microsoft Visual Studio 2005 or Microsoft Visual Web Developer.
- IronPython for ASP.NET Community Technology Preview, including Visual Studio Integration or Visual Web Developer Integration, as appropriate

This walkthrough assumes that you have a general understanding of working in Visual Web Developer. For an introduction, see Walkthrough: Creating a Basic Page in Visual Web Developer.

Create a Web Site

In this part of the walkthrough, you will create a Web site with IronPython as the default language.

To create a Web site with a default Web Form

1. Open Visual Studio or Visual Web Developer.
2. On the **File** menu, click **New Web Site**.

   The **New Web Site** dialog box appears.
3. In the **Language** list, click **IronPython** to make IronPython the default language for the Web site.

   > **Note** You can use statically compiled languages in the same Web application by creating pages and components in different programming languages.
4. Under **Visual Studio installed templates**, click **ASP.NET Web Site**.
5. In the **Location** box, select the **File System** box, and then enter the name of the folder where you want to keep the pages of your Web site.

   For example, type the folder name **C:\BasicWebSite**.
6. Click **OK**. Visual Studio creates the folder and a new page named Default.aspx.

Adding Controls to the Default Page

In this part of the walkthrough, you will add server controls to the page.

To add controls to the page

1. Click the **Design** tab to switch to Design view.
2. In the **Toolbox**, from the **Standard** group, drag three controls onto the page: a **TextBox** control, a **Button** control, and a **Label** control.
3. Put the insertion point above the **TextBox** control, and then type **Enter your name:**.

Programming the Button Control

For this walkthrough, you will write code that reads the name that the user enters into the text box and then displays the name in the **Label** control.

To add a button event handler

1. Right-click the page and click **View Code** to show the separate code file, Default.aspx.py.

   The file contains IronPython **import** statements for commonly used .NET Framework namespaces, and a stub event handler for the **Load** event of the page. In IronPython, **pass** is a placeholder that does nothing.

2. Replace the **pass** statement with the following IronPython code to set the label text when the page is initialized:

```
def Page_Load():
    if not IsPostBack:
        Label1.Text = "<Your name here>"
```

3. Add the following code to create an event handler for the button's **Click** event:

```
def Button1_Click(sender, args):
    Label1.Text = Textbox1.Text
```

   In this release, IronPython event handlers must be coded and bound manually. You cannot create them by double-clicking a control in Design mode or by selecting an event in the Properties window.

   Because IronPython does not have typed parameters and variables, you do not need to know the type of the event argument object.

   > **Note** In this release, IntelliSense support is limited in IronPython because variables are dynamically typed. You can press CTRL+SPACE to get a list of IronPython code elements that are currently in scope.

4. Switch to Default.aspx and go to Source view, and then bind the event handler by adding an **OnClick** attribute to the **Button** control markup, as shown in the following example:

```
<form id="form1" runat="server">
<div>
  <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
  <asp:Button ID="Button1" runat="server" Text="Button"
    OnClick="Button1_Click"/><br /><br />
  <asp:Label ID="Label1" runat="server" Text="Label"></asp:Label>
  <br />
</div>
</form>
```

5. Press CTRL+F5 to run the page in the browser using the ASP.NET Development Server.

6. Enter a name into the text box and click the button.

   The name you entered is displayed in the **Label** control. If the name does not appear, check the spelling of the event handler in the **OnClick** attribute.

7. In the browser, optionally view the source of the page you are running.

8. Close the browser.

Add a Page Without a Separate Code File

In this part of the walkthrough, you will add a page that contains its own code.

To add a Web page with its own code

1. In Solution Explorer, right-click the Web site (for example, **C:\BasicWebSite**), and then click **Add New Item**.
2. Under **Visual Studio installed templates**, click **Web Form**.
3. In the **Name** box, type **SecondWebPage**.
4. In the **Language** list, select **IronPython**.
5. Clear the **Place code in separate file** check box to create a single-file page with the code and markup in the same page.
6. Click **Add**.

   Visual Web Developer creates the new page and opens it in Source view.
7. Click the **Design** tab to switch to Design view, and add a **TextBox** control, a **Button** control, and a **Label** control, as you did for the default page.

Programming the Button Control

For this walkthrough, you will add IronPython code in a script block.

To add a default button event handler

1. Click the **Source** tab to switch to Source view.
2. Add the following code to initialize the label and to create an event handler for the button's Click event.

   ```
   <script runat="server">
   def Page_Load():
       if not IsPostBack:
           Label1.Text = "<Your name here>"


   def Button1_Click(sender, args):
       Label1.Text = Textbox1.Text
   </script>
   ```
3. In the Button control, bind the event handler by adding the OnClick attribute , as in the earlier part of the walkthrough. The following example shows the markup.

   ```
   <asp:Button ID="Button1" runat="server" Text="Button"
       OnClick="Button1_Click"/><br />
   ```
4. Press CTRL+F5 to run the page in the browser using the ASP.NET Development Server.

Next Steps

This walkthrough has illustrated the basic use of IronPython to program Web pages. Related walkthroughs include:

- Add data access to IronPython Web pages. See the companion walkthrough "Databinding with IronPython for ASP.NET".
- Learn about debugging IronPython Web pages. See the companion walkthrough "Debugging with IronPython for ASP.NET".
- Create a user control using IronPython. See the companion walkthrough "Creating a User Control with IronPython".
- Add IronPython classes and use them in your Web Pages. See the companion walkthrough "Using Shared Code with IronPython for ASP.NET".

See Also

Walkthrough: Creating a Basic Page in Visual Web Developer