

Walkthrough: Using Shared Code with IronPython for ASP.NET

Introduction

In this walkthrough, you will create a simple IronPython class and then use it in an ASP.NET Web 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 Studio. For an introduction, see [Walkthrough: Creating a Basic Page in Visual Web Developer](#).

Creating the Web Site and Page

If you have already created a Web site in Visual Studio (for example, by working with the companion walkthrough "Creating a Basic Web Page with IronPython"), you can use that Web site and go to the next section. Otherwise, create a new Web site and page by following these steps.

To create a file system Web site

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, select **IronPython** as the default for your Web site.
Note You can include 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, click **File System** and then type the name of the folder where you want to keep the pages of your Web site.
For example, type the folder name C:\SampleSite.
6. Click **OK** to create the folder and a default page.

Creating a Shared Class

You can create reusable IronPython classes by keeping them in a folder named App_Script. The App_Script folder is created automatically when you create a new Web site using IronPython as the default language.

Note You should put only classes (and other supported shared types) into the App_Script folder. Do not put pages, Web user controls, or other files that contain non-code elements into the App_Script folder.

To create a shared class in the App_Script folder

1. In Solution Explorer, right-click **App_Script**, and then click **Add New Item**.

Note Be sure to create the new item in the App_Script folder, not in the root folder of the Web site.

2. Under **Visual Studio installed templates**, click **IronPython Module**.
3. In the **Name** box, type SampleModule.
4. Click **Add**.

Visual Studio opens the new module file in the editor. The module has nothing in it but a doc comment.

5. Add a class named single property named **TestString** by creating access methods and assigning them to the property name with the **property** function. The following example shows the code for the class and property.

```
class SampleClass:
    "Sample class with one property"
    _testString = ""
    def SetTestString(value):
        _testString = value
    def GetTestString():
        return _testString
    TestString = property(GetTestString, SetTestString)
```

Using the Shared Class

The next step is to use the shared class in an ASP.NET Web page. You can use the Default.aspx page that was created when you created the Web site.

To use the shared class

1. Open or switch to the Default.aspx page, and then switch to Design view.
Note If you do not have a Default.aspx page, you can use another page or add a new page to the Web site.
2. From the **Standard** folder in the Toolbox, drag a **TextBox** control, **Label** control, and **Button** control onto the page.

Note For this walkthrough, the layout of the page is not important.

3. Right-click the page and click **View Code**, and add import statements for the new module using the following code:

```
import SampleModule
from SampleModule import SampleClass
```

4. Add an event handler for the **Click** event of the button, with the following code:

```
def Button1_Click(sender, args):
    sc = SampleClass1()
    sc.TestString = TextBox1.Text
    Label1.Text = sc.TestString
```

Note 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.

5. Switch to Source view, and then bind the event handler to the **Click** event using the following markup.

```
<asp:Button ID="Button1" runat="server" Text="Button"
    OnClick="Button1_Click" /><br />
```

6. Press CTRL+F5 to run the page.
7. When the page appears in the browser, enter something in the text box, and then click the button.

The property of the sample class is set to the value in the **TextBox**, and then displayed in the **Label** control.

If you use Microsoft Windows Explorer to examine the directory where the Web site is located, you will see your page and the App_Script folder. Note that there is no .dll or other executable code in the App_Script folder or anywhere under the root of the Web site. The module is compiled dynamically.

See Also

[Walkthrough: Creating a Basic Page in Visual Web Developer](#)