.NET Framework 4 - ASP.NET

**Walkthrough: Creating a Synchronous HTTP Handler**

This walkthrough illustrates how to create an HTTP handler that performs synchronous processing of requests. The example handler processes requests for resources in an ASP.NET application whose URL ends with .sample.

When users request a resource whose URL ends in .sample, the Web server forwards the request to ASP.NET. ASP.NET then calls the HTTP handler, which returns a response. The response is created dynamically by the handler. There is no need for a file that has the file name extension .sample to exist.

For more information about how the ASP.NET runtime interacts with IIS 6.0, see [ASP.NET Application Life Cycle Overview for IIS 5.0 and 6.0](http://msdn.microsoft.com/en-us/library/ms178473.aspx). For more information about ASP.NET integration with IIS 7.0, see [ASP.NET Application Life Cycle Overview for IIS 7.0](http://msdn.microsoft.com/en-us/library/bb470252.aspx).

Tasks illustrated in this walkthrough include the following:

* How to create the code for an HTTP handler class. The class must implement the [ProcessRequest](http://msdn.microsoft.com/en-us/library/system.web.ihttphandler.processrequest.aspx) method and the [IsReusable](http://msdn.microsoft.com/en-us/library/system.web.ihttphandler.isreusable.aspx) property.
* How to register the handler in the Web.config file and map the .sample file name extension to it.
* How to map the .sample file name extension to ASP.NET In Internet Information Services (IIS).

|  |
| --- |
| **Description: NoteNote** |
| The ASP.NET Development Server will serve the request for the resource after the configuration file is changed to include a reference to the new handler. For more information about the ASP.NET Development Server, see [Web Servers in Visual Studio for ASP.NET Web Projects](http://msdn.microsoft.com/en-us/library/58wxa9w5.aspx). To enable IIS to serve the request, see the procedures later in this walkthrough. |

Description: http://i.msdn.microsoft.com/Global/Images/clear.gifPrerequisites

In order to complete this walkthrough, you will need:

* Visual Studio or Visual Web Developer.
* An ASP.NET Web site that you can run by using IIS.
* IIS 6.0 or IIS 7.0.

Description: http://i.msdn.microsoft.com/Global/Images/clear.gifCreating a Web Site that Runs Under IIS

For this walkthrough, you must run a Web site by using IIS.

**To create a Web site that runs under IIS**

1. Open Visual Studio or Visual Web Developer.
2. In the **File** menu, click **New Web Site**.
3. In the **Location** list, select **HTTP** and then enter "http://localhost/HttpHandler" in the text box.
4. Click **OK**.

Description: http://i.msdn.microsoft.com/Global/Images/clear.gifCreating a Synchronous HTTP Handler Class

**To create the custom HelloWorldHandler HTTP handler class**

1. In **Solution Explorer**, right-click the project, click **Add ASP.NET Folder**, and then click **App\_Code**.
2. In the App\_Code folder, create a class named HelloWorldHandler and add the following code to the class file.

Visual Basic

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl37_ctl00_ctl00_code');" \o "Copy Code)

Imports System.Web

Public Class HelloWorldHandler

Implements IHttpHandler

Public Sub ProcessRequest(ByVal context As \_

System.Web.HttpContext) Implements \_

System.Web.IHttpHandler.ProcessRequest

Dim request As HttpRequest = context.Request

Dim response As HttpResponse = context.Response

' This handler is called whenever a file ending

' in .sample is requested. A file with that extension

' does not need to exist.

response.Write("<html>")

response.Write("<body>")

response.Write("<h1>Hello from a synchronous custom HTTP handler.</h1>")

response.Write("</body>")

response.Write("</html>")

End Sub

Public ReadOnly Property IsReusable() As Boolean \_

Implements System.Web.IHttpHandler.IsReusable

Get

Return False

End Get

End Property

End Class

C#

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl37_ctl00_ctl01_code');" \o "Copy Code)

using System.Web;

public class HelloWorldHandler : IHttpHandler

{

public HelloWorldHandler()

{

}

public void ProcessRequest(HttpContext context)

{

HttpRequest Request = context.Request;

HttpResponse Response = context.Response;

// This handler is called whenever a file ending

// in .sample is requested. A file with that extension

// does not need to exist.

Response.Write("<html>");

Response.Write("<body>");

Response.Write("<h1>Hello from a synchronous custom HTTP handler.</h1>");

Response.Write("</body>");

Response.Write("</html>");

}

public bool IsReusable

{

// To enable pooling, return true here.

// This keeps the handler in memory.

get { return false; }

}

}

The code implements the [ProcessRequest](http://msdn.microsoft.com/en-us/library/system.web.ihttphandler.processrequest.aspx) method and writes a string to the [Response](http://msdn.microsoft.com/en-us/library/system.web.httpcontext.response.aspx) property of the current [HttpContext](http://msdn.microsoft.com/en-us/library/system.web.httpcontext.aspx) object.

Description: http://i.msdn.microsoft.com/Global/Images/clear.gifRegistering the Custom HTTP Handler in IIS 6.0

After you have created the custom HTTP handler class, you must register it in the application's Web.config file. This enables ASP.NET to find the handler when requests are made for resources whose URL ends with .sample.

There are different procedures for registering the handler, depending on whether you are working with IIS 6.0 or IIS 7.0. This section describes how to register a handler in IIS 6.0. The next section describes how to register a handler in IIS 7.0.

**To register the handler in IIS 6.0**

1. If the Web site does not already have a Web.config file, create one under the root of the site.
2. Add the following highlighted element to the Web.config file.

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl38_ctl00_ctl00_code');" \o "Copy Code)

<configuration>

<system.web>

<httpHandlers>

<add verb="\*" path="\*.sample"

type="HelloWorldHandler"/>

</httpHandlers>

</system.web>

</configuration>

The configuration element registers the custom handler by class name, and it maps the .sample file name extension to that handler.

Register an application extension mapping for the .sample file name extension by using IIS Manager. For more information, see [How to: Configure an HTTP Handler Extension in IIS](http://msdn.microsoft.com/en-us/library/bb515343.aspx).

Description: http://i.msdn.microsoft.com/Global/Images/clear.gifRegistering the Custom HTTP Handler in IIS 7.0

In IIS 7.0, an application can run in either Classic or Integrated mode. In Classic mode, requests are processed much the same way as they are in IIS 6.0. In Integrated mode, IIS 7.0 manages requests by using a pipeline that enables it to share requests, modules, and other features with ASP.NET.

For IIS 7.0, the handler registration requires either registering the handler in the Web.config file or in IIS Manager. Because of the centralized administration in IIS 7.0, changes in an application's Web.config file are reflected in IIS Manager interface for the application and vice versa. In the following procedures, the handlers are registered in the Web.config file.

There are different procedures for registering the handler for IIS 7.0 running in Classic mode and running in Integrated mode. Follow the procedure for the IIS mode that you are using.

**To register the handler in IIS 7.0 running in Classic mode**

1. If the Web site does not already have a Web.config file, create one under the root of the site.
2. Add the following highlighted element to the Web.config file.

|  |
| --- |
| **Description: NoteNote** |
| Substitute the correct path of the aspnet\_isapi.dll file. The .dll file is in the folder where the .NET Framework is installed. By default this is C:\WINDOWS\Microsoft.NET\Framework\*version*. |

1. [Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl39_ctl00_ctl01_code');" \o "Copy Code)
2. <configuration>
3. <system.web>
4. <httpHandlers>
5. <add verb="\*" path="\*.sample"
6. type="HelloWorldHandler"/>
7. </httpHandlers>
8. </system.web>
9. <system.webServer>
10. <handlers>
11. <add verb="\*" path="\*.sample"
12. name="HelloWorldHandler"
13. type="HelloWorldHandler"
14. modules="IsapiModule"/>
15. scriptProcessor="%path%\aspnet\_isapi.dll"
16. </handlers>
17. </system.webServer>
18. </configuration>
19. The configuration element registers the custom handler by class name and maps the .sample file name extension to that handler.

|  |
| --- |
| **Description: NoteNote** |
| Because you are registering a custom file name extension, you register the handler in both the **handlers** section and the **httpHandlers** section. In Classic mode, for backward compatibility, the handler is specified as an ISAPI module by using the **modules** attribute. The path of the ASP.NET ISAPI dll is specified by using the **scriptProcessor** attribute. The **name** attribute is required in the **handlers** section. |

**To register the handler in IIS 7.0 running in Integrated mode**

1. If the Web site does not already have a Web.config file, create one under the root of the site.
2. Add the following highlighted element to the Web.config file.

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl39_ctl00_ctl03_code');" \o "Copy Code)

<configuration>

<system.webServer>

<handlers>

<add verb="\*" path="\*.sample"

name="HelloWorldHandler"

type="HelloWorldHandler"/>

</handlers>

</system.webServer>

</configuration>

The configuration element registers the custom handler by class name and maps the .sample file name extension to that handler.

|  |
| --- |
| **Description: NoteNote** |
| The registration is done in the **handlers** section, but not in the **httpHandlers** section. The **name** attribute is required. |

Description: http://i.msdn.microsoft.com/Global/Images/clear.gifTesting the Custom HTTP Handler

After you have created and registered the custom HTTP handler, you can test it.

**To test your custom HTTP handler**

1. In the browser, request a page from the Web application.
2. In the browser, enter a URL that ends in **.sample**. For example, enter the following URL:

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl40_ctl00_ctl00_code');" \o "Copy Code)

http://localhost/HttpHandler/test.sample

The text defined in the HelloWorldHandler class is displayed.