ASP.NET

**How to: Configure ASP.NET Applications for an ASP.NET Version**

By default, when the .NET Framework is installed on a computer where the runtime is already installed, all ASP.NET applications are automatically updated to use that version of the .NET Framework. The only exceptions are applications that are bound to an incompatible version of the runtime or to a later version of the runtime. (Compatibility is typically determined by the .NET Framework version number. Major version differences are usually incompatible, whereas minor updates and build number version differences are compatible.) Although later versions of the .NET Framework are designed to be backward compatible, you might want to configure an ASP.NET application to use a specific version of the runtime.

When managing a computer with multiple versions of the .NET Framework installed, it is often useful to view an ASP.NET application's script map (sometimes referred to as an application map) to determine which version the application is using. You can view the script map for an ASP.NET application using the Internet Information Services (IIS) management console.

**To view the script map for an ASP.NET application**

1. Open the IIS management console.
2. Expand the local computer node, expand **Web Sites**, and then expand **Default Web Server**.
3. Right-click the folder for the application, and then click **Properties**.
4. On the **Directory** tab, click **Configuration**.

The **Application Configuration** dialog box appears.

1. On the **Mappings** tab, select an ASP.NET application extension, such as .asmx or .aspx.

The **Executable Path** column of the dialog box lists the path to the ASP.NET ISAPI extension used by the application. By default, the ASP.NET ISAPI extension is installed in the following location:

*%system root%*\Microsoft.NET\Framework\*versionNumber*

The version number shown in the path indicates the version number of the ASP.NET ISAPI extension used by the application. The version of the ASP.NET ISAPI extension in turn determines which version of the runtime is used by the application.

**Configuring Applications to Use Script Maps**

To make it easier to reconfigure the script map for an ASP.NET application, each installation of the .NET Framework has its own version of the ASP.NET IIS Registration tool (Aspnet\_regiis.exe). A user with administrator privileges on the Web server computer can use this tool to remap an ASP.NET application to the version of the ASP.NET ISAPI extension that matches the tool.

|  |
| --- |
| **Description: NoteNote** |
| Because the Aspnet\_regiis.exe tool is associated with a specific version of the .NET Framework, you must use the appropriate version of Aspnet\_regiis.exe to reconfigure the script map for an ASP.NET application. The Aspnet\_regiis.exe tool reconfigures the script map of an ASP.NET application to the version of the ASP.NET ISAPI extension that matches the tool. |

The tool can also be used to display the status of all installed versions of ASP.NET, register the associated version of ASP.NET, create client-script directories, and perform other configuration operations.

**To use Aspnet\_regiis.exe to update a script map for an ASP.NET application**

1. Open a Windows command window.
2. Navigate to the directory containing the version of the Aspnet\_regiis.exe tool that you want to use. Remember that each version of the .NET Framework comes with its own version. The file is usually located in the following directory:

*%system root%*\Microsoft.NET\Framework\*versionNumber*

1. Run the Aspnet\_regiis.exe tool with the **-s** or **-sn** option and the path to the application.

The following example shows a command that updates the script maps for an application called SampleApp1:

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl01_code');" \o "Copy Code)

Aspnet\_regiis.exe -s W3SVC/1/ROOT/SampleApp1

**Configuring Application Pools**

IIS version 5.0 uses the ASP.NET process model (Aspnet\_wp.exe). Under the ASP.NET process model, each unique application version automatically runs in a separate process at run time. All applications that target the same version of the runtime share the same process (or processes in Web garden mode). However, IIS 6.0 uses the IIS 6.0 process model (w3wp.exe) and uses an isolation feature called application pooling. Application pooling enables applications to run together in one or more processes, as long as they share the same pool designation. Applications that are assigned different application pools never run in the same process.

|  |
| --- |
| **Description: Caution noteCaution** |
| ASP.NET version 1.0 is supported only in the IIS 5.0 isolation mode of IIS 6.0. |

When multiple versions of the .NET Framework are installed on a computer that uses IIS 6.0, you might see the following error message in the Application Event log:

**It is not possible to run different versions of ASP.NET in the same IIS process. Use the IIS Administration Tool to reconfigure your server to run the application in a separate process.**

This error occurs when more than one version of ASP.NET is configured to run in the same process, because different versions of the .NET Framework and runtime cannot run side-by-side in the same process. Therefore, an ASP.NET application that uses a particular version of the runtime must not share a process with an application that uses a different version. This error commonly occurs when two or more applications are mapped to different versions of ASP.NET, but share the same application pool.

|  |
| --- |
| **Description: NoteNote** |
| The process isolation for each application is configured manually. |

**To disable IIS 5.0 isolation mode in IIS 6.0**

1. Open the IIS management console.
2. Right-click the **Web Sites** folder and then click **Properties**.
3. On the **Service** tab, clear the **Run WWW service in IIS 5.0 isolation mode** check box.

**To assign a pool designation to an ASP.NET application in IIS 6.0**

1. Open the IIS management console.
2. Expand the local computer node, expand **Web Sites**, and then expand **Default Web Server**.
3. Right-click the folder for the application, and then click **Properties**.
4. On the **Directory** tab, select the desired pool designation from the **Application Pool** list.

**To create a pool designation in IIS 6.0**

1. Open the IIS management console.
2. Expand the local computer node.
3. Right-click the **Application Pools** folder, click **New**, and then click **Application Pool**.

The **Add New Application Pool** dialog box appears.

1. Enter the new pool designation in the **Application pool** text box, and then click **OK**.