

Re: unable to run asp.net on 2000 server

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.aspnet/2004-02/3434.html>

From: jqpdev (jqpdev_at_counterattack.com)

Date: 02/15/04

Date: Sun, 15 Feb 2004 17:15:24 -0500

Give this a shot... This is a copy of an article straight from MSDN. Its about the tool that I had to run on my box to get VS.NET\ASP.NET\IIS to play nice. I hope you can read this because is did select, copy and paste. If you are unable to read it do a search for "aspnet_regiis":

By default, when the .NET Framework is installed on a computer with an existing installation, all ASP.NET applications are automatically updated to use this 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. Although later versions of the .NET Framework are designed to be backward compatible, you might want to configure an ASP.NET application to use an earlier version. The following sections describe the process to configure an ASP.NET application to target a specific version of the .NET Framework.

Viewing the Script Map for an ASP.NET Application

When managing a computer with multiple versions of the .NET Framework installed, it is often useful to view the script map for an ASP.NET application to determine which version is used by the application. You can view the script map for an ASP.NET application using the Internet Information Services management console.

To view the script map for an ASP.NET Application

- 1.. Open the IIS management console, expand the local computer by clicking the plus sign, and navigate to the folder that contains the ASP.NET application.
- 2.. Right-click the folder and click Properties. The application's Properties dialog box appears.
- 3.. On the Directory tab, click the Configuration button. The Application Configuration dialog box appears.
- 4.. 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 version used by the application. By default, the ASP.NET ISAPI is installed in the following location:

systemroot\Microsoft.NET\Framework\versionNumber

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

Updating Script Maps Using Aspnet_regiis.exe

To make it easier to reconfigure the script map for an ASP.NET application, each installation of the .NET Framework comes with an associated version of the ASP.NET IIS Registration tool (Aspnet_regiis.exe). Administrators can use this tool to remap an ASP.NET application to the ASP.NET ISAPI version associated with the tool.

Note Because Aspnet_regiis.exe is linked to a specific version of the .NET Framework, administrators must use the appropriate version of Aspnet_regiis.exe to reconfigure the script map for an ASP.NET application. Aspnet_regiis.exe only reconfigures the script map of an ASP.NET application to the ASP.NET ISAPI version associated with 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 command window. (Click Start, click Run, type cmd, and then click OK.)
- 2.. Navigate to the directory of the Aspnet_regiis.exe version 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:
systemroot\Microsoft.NET\Framework\versionNumber

- 3.. Use the `-s` or `-sn` option of Aspnet_regiis.exe along with the path to the application to set up the script maps. The following shows a sample command line that updates the script maps for an application called SampleApp1.

Aspnet_regiis.exe -s W3SVC/1/ROOT/SampleApp1Application Pooling in IIS 6.0
IIS versions earlier than 6.0 use 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 introduces a new isolation feature called application pooling. Application pooling allows 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.

When multiple versions of the .NET Framework are installed on a computer that uses IIS 6.0, you might encounter 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. Please 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. Different versions of the .NET Framework and runtime cannot coexist 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.

To use application pooling, first make sure IIS 5.0 isolation mode is disabled. Next, navigate to the desired application in the Internet Information Services management console and assign it an application pool. If you want to isolate your application from the existing application pools, create a new pool designation.

Note 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 and expand the local computer by clicking the plus sign.
- 2.. Right-click the Web Sites folder and click Properties.
- 3.. On the Service tab, clear the Run Web 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, expand the local computer by clicking the plus sign, and navigate to the folder that contains the ASP.NET application.
- 2.. Right-click the application and then click Properties. The application's properties dialog box appears.
- 3.. 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 and expand the local computer by clicking the plus sign.
- 2.. Right-click the Application Pools folder, point to New, and then click Application Pool. The Add New Application Pool dialog box appears.
- 3.. Enter the new pool designation in the Application pool text box, and then click OK.