Show

RUNNING APPLICATIONS IN 64-BIT MODE

Iron Speed Designer Version 6.1 generates code that will work in either 32-bit or 64-bit mode without any change. When you run your generated application on a 64-bit machine, there are three factors that determine whether your application is running under 32-bit or 64-bit mode:

- Whether Internet Information Server (IIS) is running in 64-bit mode.
- Whether the Application Pool is set to use 64-bit mode (only applies to IIS 7.0)
- Whether all of your application DLL's were built using "Any CPU" mode

Running Microsoft IIS in 64-bit mode

You can easily identify whether Microsoft IIS is running in 32-bit mode or 64-bit mode by looking at the Processes in the Task Manager. The w3wp.exe will show *32 next to the image name if the World Wide Web (w3) worker process (wp) is running in 32-bit mode.

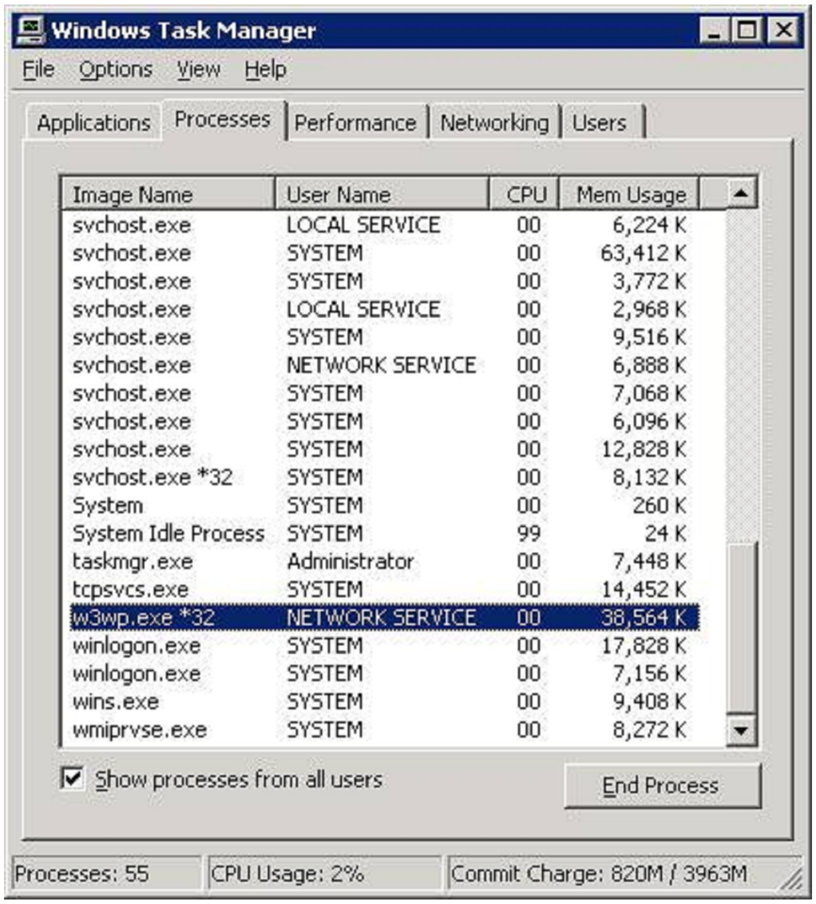

Windows Task Manager
File Options View Help
Applications Processes Performance Networking Users
Show processes from all users
End Process

Image Name	User Name	CPU	Mem Usage
svchost.exe	LOCAL SERVICE	00	6,224 K
svchost.exe	SYSTEM	00	63,412 K
svchost.exe	SYSTEM	00	3,772 K
svchost.exe	LOCAL SERVICE	00	2,968 K
svchost.exe	SYSTEM	00	9,516 K
svchost.exe	NETWORK SERVICE	00	6,888 K
svchost.exe	SYSTEM	00	7,068 K
svchost.exe	SYSTEM	00	6,096 K
svchost.exe	SYSTEM	00	12,828 K
svchost.exe *32	SYSTEM	00	8,132 K
System	SYSTEM	00	260 K
System Idle Process	SYSTEM	99	24 K
taskmgr.exe	Administrator	00	7,448 K
tcpsvcs.exe	SYSTEM	00	14,452 K
w3wp.exe *32	NETWORK SERVICE	00	38,564 K
winlogon.exe	SYSTEM	00	17,828 K
winlogon.exe	SYSTEM	00	7,156 K
wins.exe	SYSTEM	00	9,408 K
wmiprvse.exe	SYSTEM	00	8,272 K

Processes: 55 CPU Usage: 2% Commit Charge: 820M / 3963M

If no *32 is shown next to the image name, then the worker process is running in 64-bit mode.

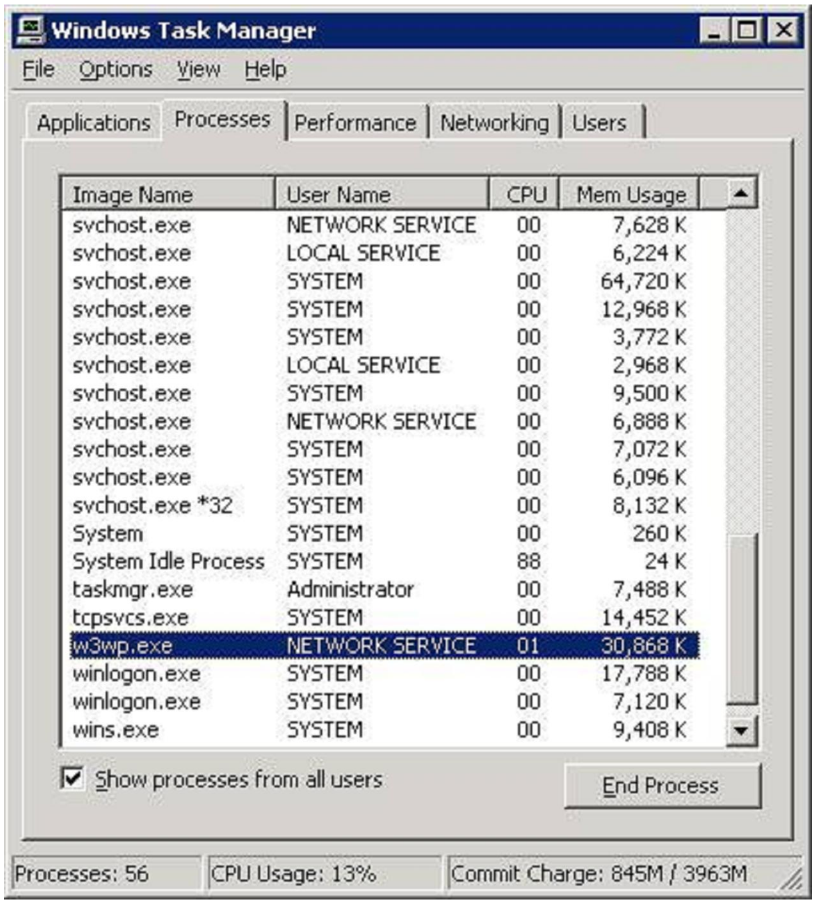

Windows Task Manager
File Options View Help
Applications Processes Performance Networking Users
Show processes from all users
End Process

Image Name	User Name	CPU	Mem Usage
svchost.exe	NETWORK SERVICE	00	7,628 K
svchost.exe	LOCAL SERVICE	00	6,224 K
svchost.exe	SYSTEM	00	64,720 K
svchost.exe	SYSTEM	00	12,968 K
svchost.exe	SYSTEM	00	3,772 K
svchost.exe	LOCAL SERVICE	00	2,968 K
svchost.exe	SYSTEM	00	9,500 K
svchost.exe	NETWORK SERVICE	00	6,888 K
svchost.exe	SYSTEM	00	7,072 K
svchost.exe	SYSTEM	00	6,096 K
svchost.exe *32	SYSTEM	00	8,132 K
System	SYSTEM	00	260 K
System Idle Process	SYSTEM	88	24 K
taskmgr.exe	Administrator	00	7,488 K
tcpsvcs.exe	SYSTEM	00	14,452 K
w3wp.exe	NETWORK SERVICE	01	30,868 K
winlogon.exe	SYSTEM	00	17,788 K
winlogon.exe	SYSTEM	00	7,120 K
wins.exe	SYSTEM	00	9,408 K

Processes: 56 CPU Usage: 13% Commit Charge: 845M / 3963M

To run the 64-bit version of Microsoft IIS:

Step 1: Click Start, click Run, type cmd, and then click OK.

Step 2: Type the following command to disable the 32-bit mode:

cscript %SYSTEMDRIVE%\inetpub\adminscripts\adsutil.vbs SET W3SVC/AppPools/Enable32bitAppOnWin64 0

Step 3: Type the following commands to install the version of ASP.NET 2.0 and to install the script maps at the IIS root and under:

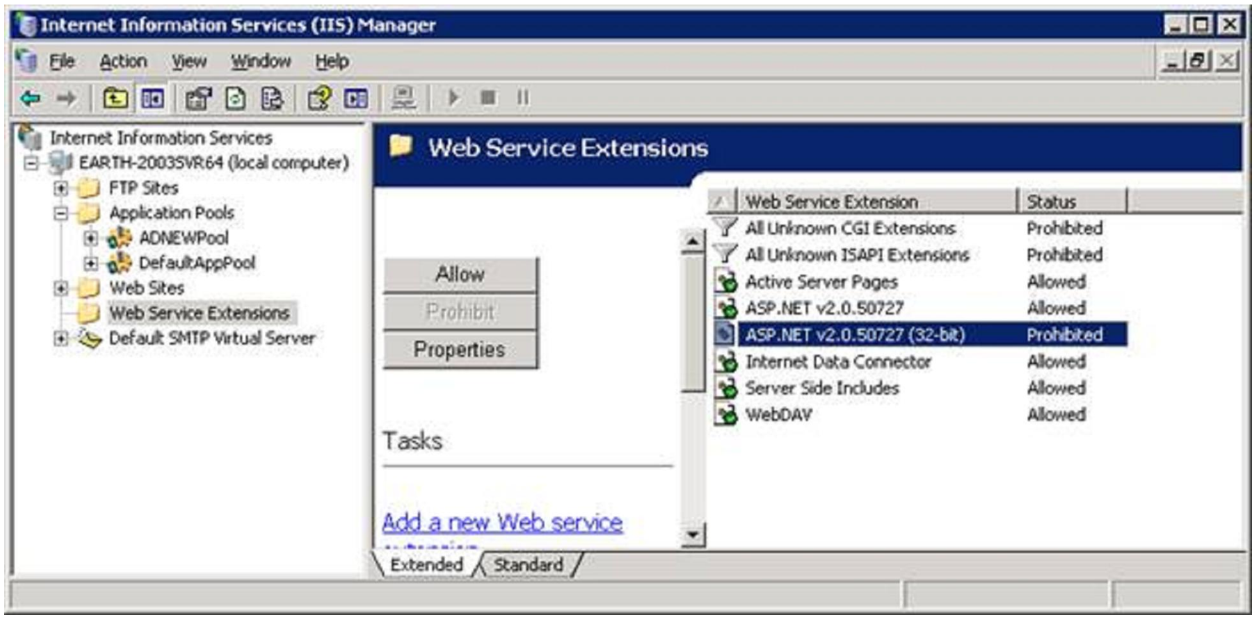
%SYSTEMROOT%\Microsoft.NET\Framework64\v2.0.50727\aspnet_regiis.exe -u

%SYSTEMROOT%\Microsoft.NET\Framework64\v2.0.50727\aspnet_regiis.exe -i

%SYSTEMROOT%\Microsoft.NET\Framework64\v2.0.50727\aspnet_regiis.exe -c

Step 4: Make sure that the status of ASP.NET version 2.0. 50727 is set to Allowed in the Web service extension list in Internet Information Services Manager.

The build version of ASP.NET 2.0 may differ depending on the currently released build version. These steps are for build version 2.0. 50727.



Web Service Extension	Status
All Unknown CGI Extensions	Prohibited
All Unknown ISAPI Extensions	Prohibited
Active Server Pages	Allowed
ASP.NET v2.0.50727	Allowed
ASP.NET v2.0.50727 (32-bit)	Prohibited
Internet Data Connector	Allowed
Server Side Includes	Allowed
WebDAV	Allowed

To restore the 32-bit version of Microsoft IIS:

Step 1: Click Start, click Run, type cmd, and then click OK.

Step 2: Type the following command to enable the 32-bit mode:

```
cscript %SYSTEMDRIVE%\inetpub\adminscripts\adsutil.vbs SET
W3SVC/AppPools/Enable32bitAppOnWin64 1
```

Step 3: Type the following commands to install the version of ASP.NET 2.0 and to install the script maps at the IIS root and under:

```
%SYSTEMROOT%\Microsoft.NET\Framework\v2.0.50727\aspnet_regiis.exe -u
```

```
%SYSTEMROOT%\Microsoft.NET\Framework\v2.0.50727\aspnet_regiis.exe -i
```

```
%SYSTEMROOT%\Microsoft.NET\Framework\v2.0.50727\aspnet_regiis.exe -c
```

Step 4: Make sure that the status of ASP.NET version 2.0. 50727 (32-bit) is set to Allowed in the Web service extension list in Internet Information Services Manager.

The build version of ASP.NET 2.0 may differ depending on what the currently released build version is. These steps are for build version 2.0. 50727.

Configuring Application Pools (Microsoft IIS 7 only)

On a 64-bit Windows 2003 Server, you cannot run worker processes in both 32-bit mode and as well as 64 bit mode. On machines that run IIS 7 such as Windows Vista or Windows 2008 server, you can configure IIS to run either a 32-bit application or a 64-bit application. More information about enabling both 32-bit and 64-bit is described at:

http://www.codedigest.com/Articles/IIS/21_IIS7_-_Running_32-bit_and_64-bit_ASPNET_versions_at_the_same_time_on_different_worker_processes.aspx

Configuring your application for 64-bit operation

After Microsoft IIS and the Application Pools are configured to run in 64-bit, you will need to ensure that all of the DLL's that are in the Bin folder of an application are in 64-bit mode. If you have the Visual Studio solution file, you can verify the configuration under the Build menu to ensure that the build configuration is set to "Any CPU". All DLL's shipped with Iron Speed Designer Version 6.1 or later are compiled in "Any CPU" mode so they will run in either 32-bit or 64-bit mode.

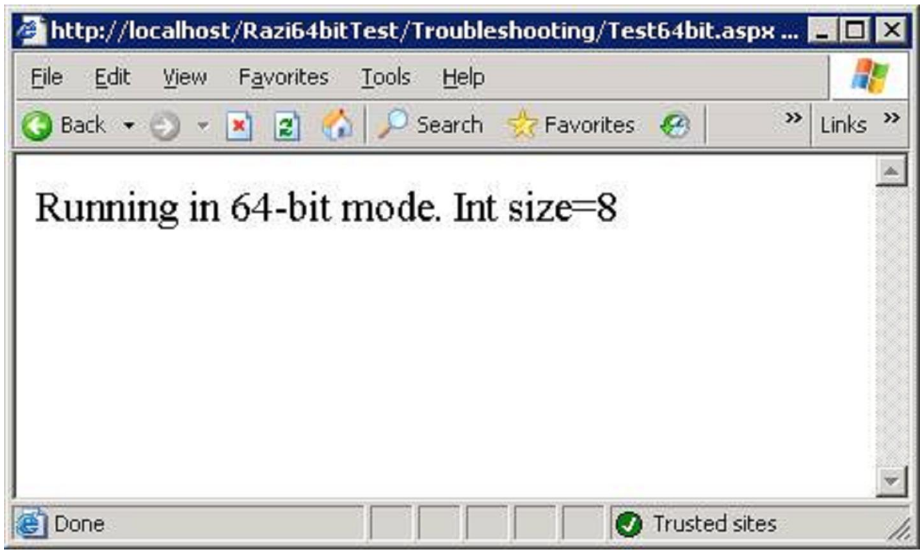
There is absolutely no difference in either the generated application code or in the DLL's between the 32-bit or 64-bit mode.

You can verify whether your application is running in 64-bit mode by creating a sample file called Test64Bit.aspx with the following code:

```
<%@ Page Language="cs" AutoEventWireup="false"  Debug="true" %>
<%
   if (IntPtr.Size == 8) {
    Response.Write("Running in 64-bit mode.  Int size=" + IntPtr.Size.ToString());
   } else {
    Response.Write("Running in 32-bit mode.  Int size=" + IntPtr.Size.ToString());
   }
%>
```

The above code checks to see the size of the Integer pointer. If the size is 8 bits, the application is running in 64-bit mode, and if it is 4-bits, the application is running in 32-bit mode.

Running this page from your application will display a web page such as:


http://localhost/Razi64bitTest/Troubleshooting/Test64bit.aspx ...
File Edit View Favorites Tools Help
Back Search Favorites Links
Done
Trusted sites

```
Running in 64-bit mode. Int size=8
```

See Also

Configuring the Microsoft IIS Web Server

Configuring Microsoft IIS 7 for Microsoft Access Applications

Configuring Microsoft IIS on Microsoft Vista

Running as Administrator in Microsoft Vista

Installing Microsoft IIS on Microsoft Windows XP Professional

Installing Microsoft IIS on Windows Server 2003

Using Non-Standard Ports in Microsoft IIS

Application Pools in Microsoft IIS

Running Applications in 64-bit Mode