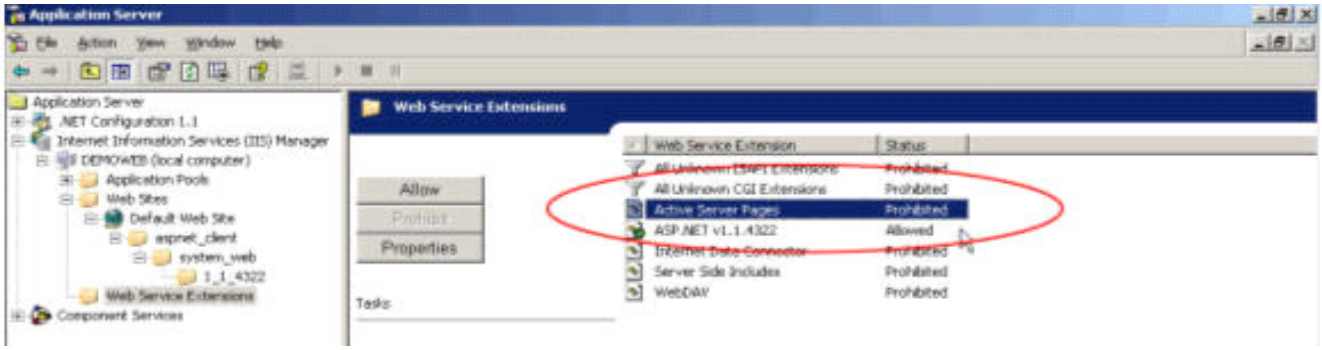
WINDOWS 2003/ IIS6 Additional Steps Required:

(1) Windows 2003 need to allow ASP pages (see copy of article below)



(2) Using Windows 2003 on the IIS server you will come across an error on TS and Exp Authorization unless you do the following.

In the IIS Administration tool, you need to enable the MIME type for the .inc file extension with the MIME type of application/octet-stream.
You do this from the HTTP Headers tab in the properties of the StarWeb virtual directory.

(3) Enable Parent Paths Is Disabled by Default in IIS 6.0

You receive this error message if the application calls an ASP method that requires the **Parent Paths** option to be enabled. The exact error message depends on the method that is being called.

To resolve this problem without changing the application:

1. Click **Start**, click **Administrative Tools**, and then click **Internet Information Services (IIS) Manager**.
2. Double-click your computer name in the left pane, and then double-click **Web Sites**.
3. Locate the Web site and directory that houses the ASP application.
4. Right-click the application site or directory, and then click **Properties**.
5. Select **Home Directory**, and then click **Configuration**.
6. Click **Options**, and then click to select the **Enable Parent Paths** check box.

7. Click **OK** two times.

IIS 6 Articles : Installing IIS 6.0 on Windows Server 2003

Posted by **jctsup1** on 7/19/2004 22:01:38 (6564 reads)
http://www.iis-resources.com/modules/news/article.php?storyid=262

Title: Installing IIS 6.0 on Windows Server 2003
Level: Basic
Author: Jeff Cochran (**jeff@zina.com**)
Date: July 19, 2004

Installing Internet Information Server (IIS) in Windows Server 2003 can be confusing for those new to either IIS or Server 2003, and is even confusing for experienced admins of earlier IIS versions since some items enabled by default need to be specifically enabled for use. So we're going to start with a brand new Server 2003 installation and step through a common IIS install.

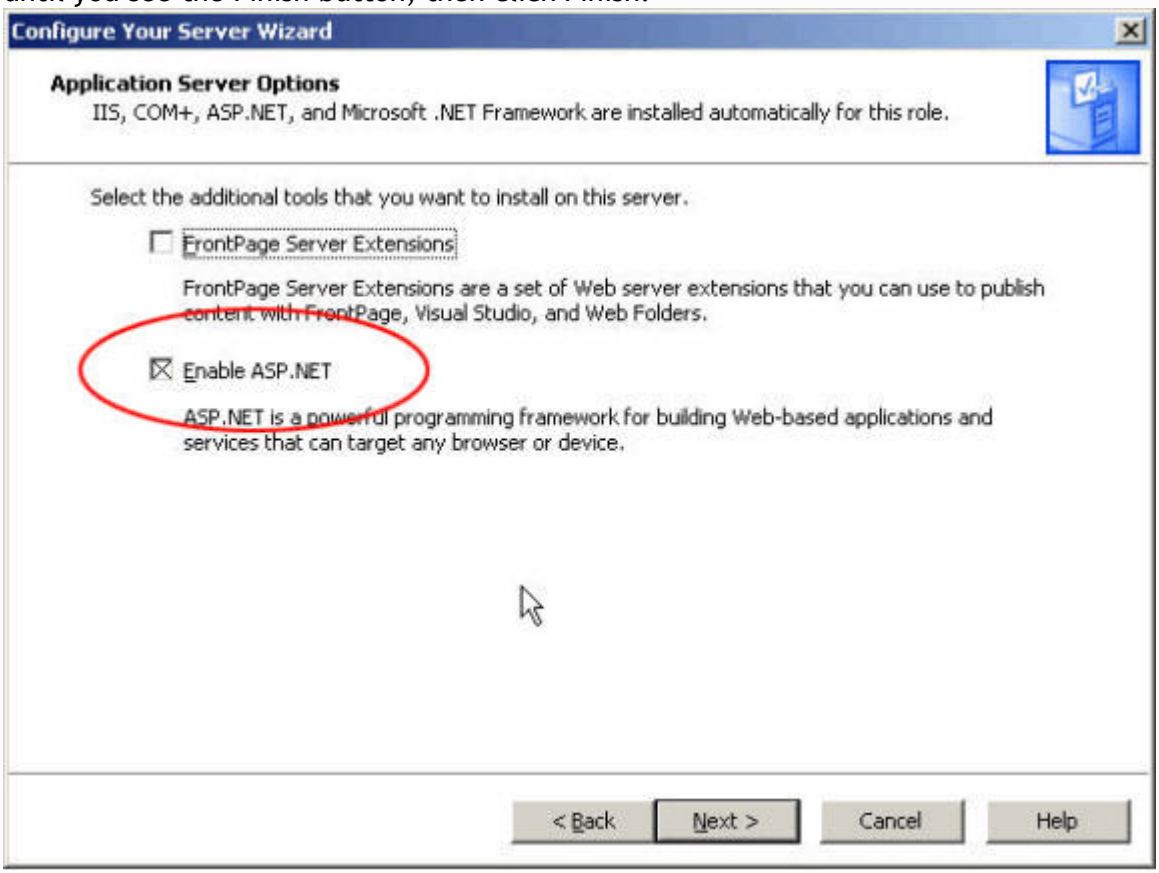
The system this is being installed on is nothing special, an 863 MHz Pentium 4 with 512 MB of RAM, two 20 GB hard drives and a basic CD-ROM. Not very sophisticated, and probably about what most organizations have already retired as a workstation, this is a surplus system that could be had for a few hundred dollars used, probably even less. But it suits our purpose here.

Windows Server 2003 Standard has been installed on this system, along with all updates and critical security fixes. It sits behind a firewall so we aren't using the Server 2003 firewall, but we have installed Anti-Virus software (F-Prot – **http://www.f-prot.com/**) and a screen capture program (SnagIt 7 - **http://www.techsmith.com/**). F-Prot has current virus signature files, and one advantage is that F-Prot doesn't disable scripting on the system the way Norton's and some other products do.

Adding the Application Server Role to Windows Server 2003

When you start your server, you should see the Manage Your Server wizard. If you don't display this by default, you can go to Administrative Tools and click on Manage Your Server. When the wizard opens, click on Add or Remove a Role.

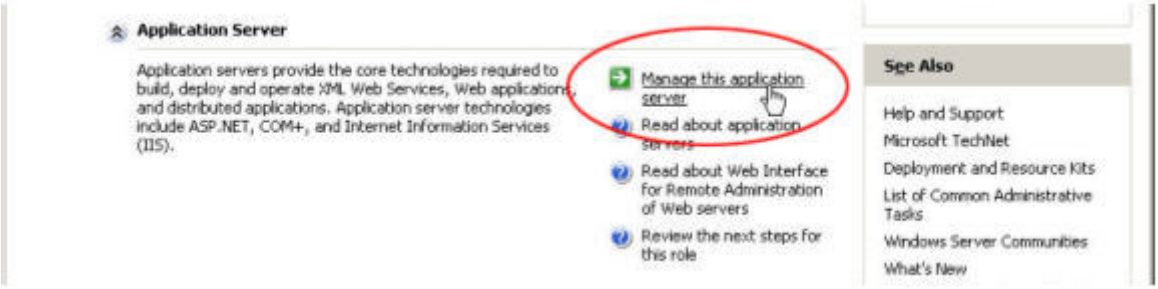
Click Next and let the server detect your settings, then choose Custom Configuration and click Next, then choose the Application Server role and click Next This should bring you to the Application Server Options screen of the wizard. We're going to choose to Enable ASP.NET, though you may choose not to if you won't have any .NET applications on the web server. In addition, we will not choose to enable FrontPage Extensions at this time, since we will configure this later. If you want to publish .NET applications from VisualStudio, you may want to enable FrontPage Extensions to start with. At any rate, once we've selected what we want, we click Next to set up the role. The process may ask you for the Server 2003 CD and will take several minutes to complete. Just let it run until you see the Finish button, then click Finish.



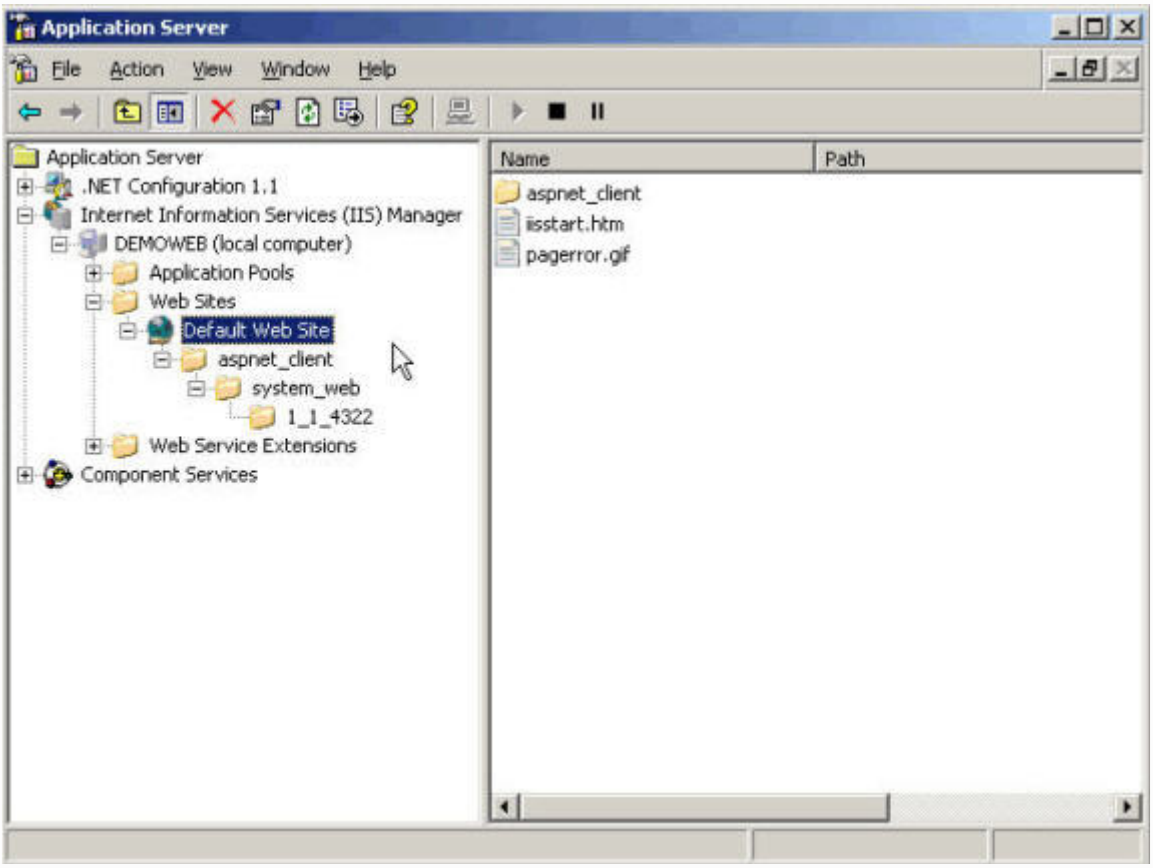
Checking Your IIS Installation

Once you've installed the Application Server role to your server, you'll naturally want to check and see if it works. The Manage Your Server wizard should now show the

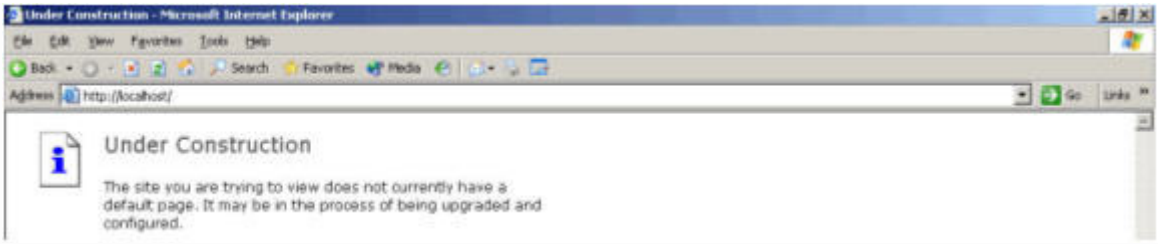
Application Server role installed, so click on Manage This Application Server.



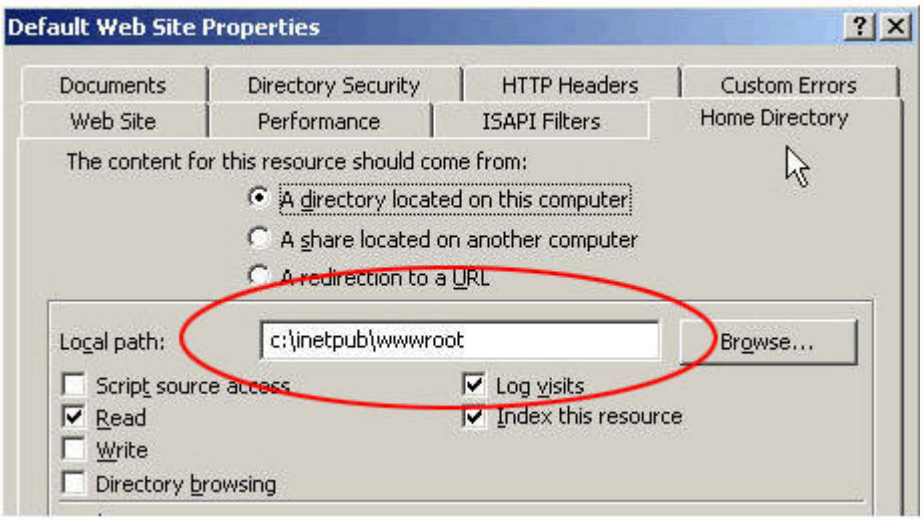
This brings up the Application Server Management Console (MMC). Expand the Internet Information Services (IIS) Manager, then expand your server (local computer), and then the Web Sites folder. You should see the Default Web Site listed, and it shouldn't say (Stopped). If it does, you need to troubleshoot using the Event Viewer.



For now, ignore the files and folders listed in your default web site, we just want to test to make sure IIS is running and serving a web page. On the server itself, launch Internet Explorer (IE) and browse to **http://localhost/.** "Localhost" is a networking indicator for the system you are physically sitting at, and should work in a default configuration of Windows Server 2003. You should see the "Under Construction" web page if your server is running correctly.



Okay, at this point you have a working IIS installation. But let's add our own files to the web site and also test the site from a system other than the server. Go back to the Application Server Management Console and right-click on the default web site. Choose properties to bring up the web site properties dialog. Then click on the Home Directory tab. Make a note of the Local Path for the home directory. That's where the files for this web site go.



So let's create a generic HTML file to test the web server with. A traditional programming test file is often referred to as a "Hello World!" file, and has only one purpose – to print the words "Hello World!" on the screen. This will be a HTML version of the same file.

We don't need some fancy graphical web site programming interface and environment with special features to create our HTML file. HTML is plain old text, so any text editor will do. Even Notepad, which is right there on your server already. So open Notepad and type in the following text:

```
<html>
<head>
<title>Hello World!</title>
</head>
<body>
<p>Hello World!</p>
</body>
</html>
```

This is the HTML code to display “Hello World!” in the browser window. You can get fancy of course, but when testing anything, the KISS principle (Keep It Simple Stupid) is always best. Since there’s nothing in this simple HTML file that will affect the display on any browser or from any web server, if the page doesn’t display then you know it’s not the page coding itself.

When you have this in Notepad, click on File then Save As. Notepad has a tricky tendency to add a .txt extension to files it saves, which won’t work in our web site, so when you save the file, make sure you choose the All Files type in the Save As Type option box.

Save the file as “hello.htm” in the same location as the Local Path in the web site Home Directory properties. Keeping your filenames as all lower case will help if you ever transfer files to other systems. Windows Server 2003 is case insensitive and treats “hello.htm”, “Hello.HTM” and “HeLLO.htM” as all the same file, but other systems may not.

Once you’ve saved the file, go back to Internet Explorer and browse to **http://localhost/hello.htm.** This time we are specifying the exact file we want displayed from the web site, so we should see the file we just saved.

That’s it. Boring and simple, but we now have installed IIS, and tested it with a file we created. That’s the basics of web site development. Okay, the very basic basics. So let’s

try some real programming, and do an Active Server Pages (ASP) file.

Open a new document in Notepad, and type in:

<%
Response.Write “Hello World!”
%>

Then save it in the same folder as before, as “hello.asp”. The ASP extension lets the server know this is an ASP file and needs to be processed on the server.

This is a really basic ASP file. All it does is print the words “Hello World!” to the screen. Which is all we want it to do, again, using the KISS principle. So let’s go back to Internet Explorer and test it the same way we did our hello.htm file. Browse to **http://localhost/hello.asp** and you should see…

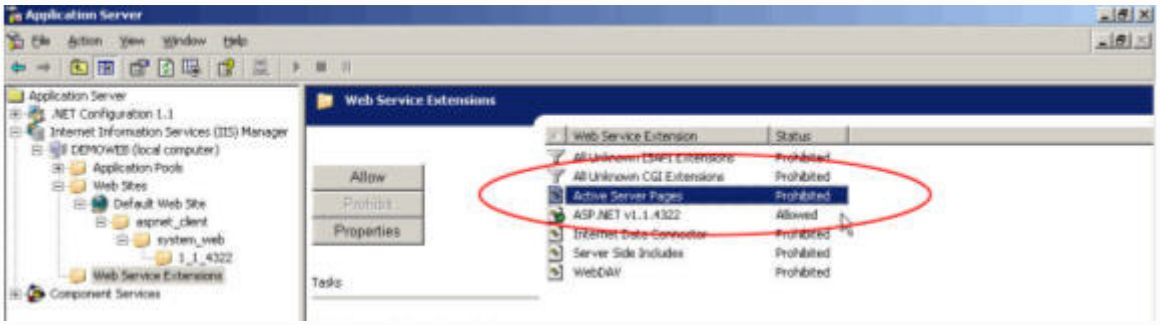
…Bummer.

Actually, that’s normal, and one of the frustrating things for admins familiar with previous version of IIS. Because of the security enhancements in Windows Server 2003 and IIS 6, ASP pages are not enabled by default. Yes, we did install the server for ASP.NET, but ASP and ASP.NET aren’t the same thing. Yeah, Microsoft named them sort of confusingly, but think of them as two entirely separate technologies and your mental state will improve.

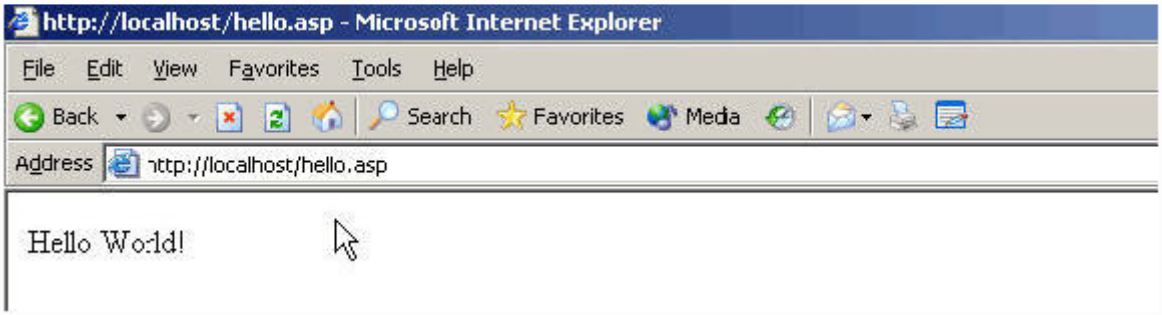
In IIS 6, technologies like ASP, ASP.NET and so on are called Web Service Extensions. The same is true of Server Side Includes, PERL and other CGI scripting, PHP and a host of

other add-ons to web servers. The security model being “don’t activate what you don’t need” we only want to allow ASP pages or other technologies if we’ll use them. Now that we’re trying to use ASP, let’s activate ASP.

In the Application Server Management Console, click on the Web Service Extensions folder underneath the server name. You should see that Active Server Pages are Prohibited, this is the default configuration of IIS.



To Allow ASP pages, simply highlight Active Server Pages and click the Allow button. ASP is now active on this web server, and if you go back to Internet Explorer, then hold [CTRL] and click the Refresh icon to reload the page from the server, you should see the friendly “Hello World!” text on the screen.



As a last test, let’s see if we can get to our new ASP web page from another system instead of the server itself. We’re going to do this by using the IP Address of the web server, since by default IIS answers web requests on All Unassigned IP Addresses on the server. Since we haven’t assigned any specifically to this web site, any IP Address on the server that can be reached by another system will work. So, from another system, browse to **http://{IP_Address}/hello.asp.** You should see the same page as above, if you don’t, it’s time to start diagnosing network connectivity. But at least you know IIS is working, to serve both HTML and ASP pages you create.

serious attempts to increase the author's net worth. (Hey, you got more than 1,500 words for free already)