Setting Up a Windows Web Server

Session Number bpei2

Barbara Peisch
Peisch Custom Software
3138 Roosevelt St. Suite O
Carlsbad, CA 92008
Voice: 760-729-9607
Email: Barbara@peisch.com

Overview

Setting up your first Windows web server may be a challenge due to the many layers of software that need to work together. In this paper, we'll discuss configuration of IIS for HTML and FTP. (This paper will NOT cover SMTP, POP3 or any aspect of Exchange Server.) Included will be examples of how to configure a domain name, and how to setup host headers to accommodate multiple domain names under the same IP address. We'll also cover some basic security precautions.

Pick Your Operating System

Setting up a new system starts with installing the operating system (OS), so it makes sense to start a discussion of setting up a web server with the OS. You want to use an OS that's meant to be a server. Windows Server 2003 or Windows 2000 Server are good choices, and those are the two operating systems I focus on in this paper. You could even go with Windows NT Server if necessary, although that's a pretty old OS in today's world, and is no longer supported by Microsoft.

Although you can develop web applications under Windows 2000 Pro or Windows XP Pro, I wouldn't recommend using them as a real web server. These operating systems weren't meant to handle much of a load and will slow down if your site gets much traffic. Either of these operating systems could be used on a staging server but it would be better to use a server OS if you have one. (A staging server is used to test your application and its installation issues before deploying to the "real" server.) Another problem with these non-server operating systems is that you must use the "Default web site" and "Default FTP site" that are created when you install IIS. You cannot create additional web sites or FTP sites.

I do **not** recommend using Windows 98 or Windows 95 as a server under any circumstances, and do not cover those at all in this paper. This is because they can only use Personal Web Server (PWS) for acting as a web server, and PWS is very much lacking in capability and configurability.

The newer the OS, the longer Microsoft will be supporting it, so for that reason, going with the latest server OS is a good idea.

Unless otherwise specified, I use Windows Server 2003 for my discussions in this paper. By default Windows Server 2003 is installed with its settings very restrictive. This is just the opposite of the default configuration settings with other operating systems, and I point out the differences where appropriate.

Plan Your Directory Structure

Before we get started on any of the server specific items, you should decide where you want to keep your web files on your server. I don't like to put anything under the default directory of \Inetpub\wwwroot because that's the first place that any hacker looks for your files. I like to leave that directory intact, just to throw off any hackers, but there's nothing of importance there. Instead, I create a completely different directory and have subdirectories for each web site I'm hosting under that. For this paper, I've named that top directory \MyWebFiles. (Figure 1)

\Inetpub\wwwroot

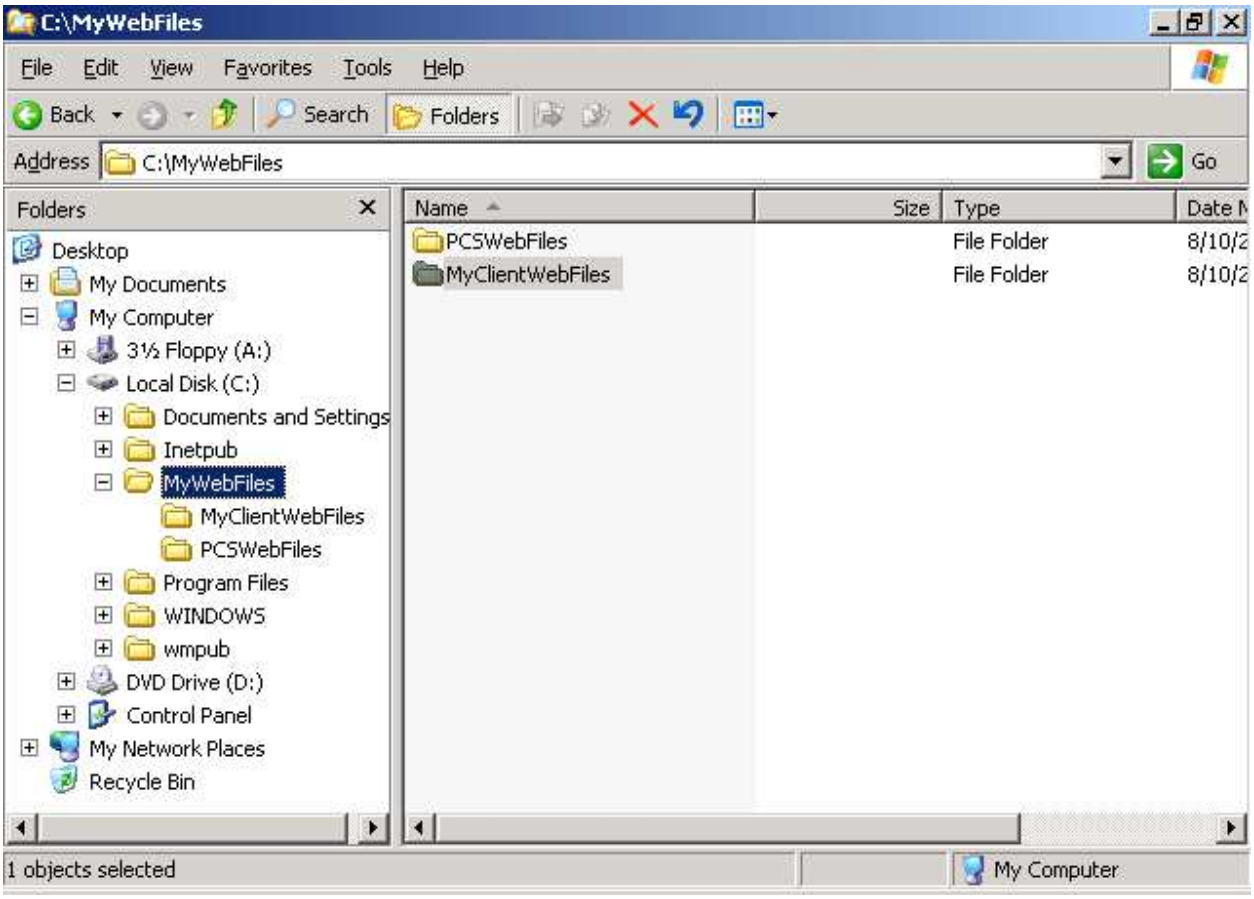


Figure 1 – My directory structure for web files

Windows Explorer Security Settings

While we're in Windows Explorer we should also discuss some security settings that need to be made. If you go to the root node of your hard drive (we'll assume it's C:) and right click, and then select Properties, you'll notice there's a Security tab. Click on that tab, and you'll see the users who have access to your C: drive. (Figure 2)

right click on C: and choose Properties then Security

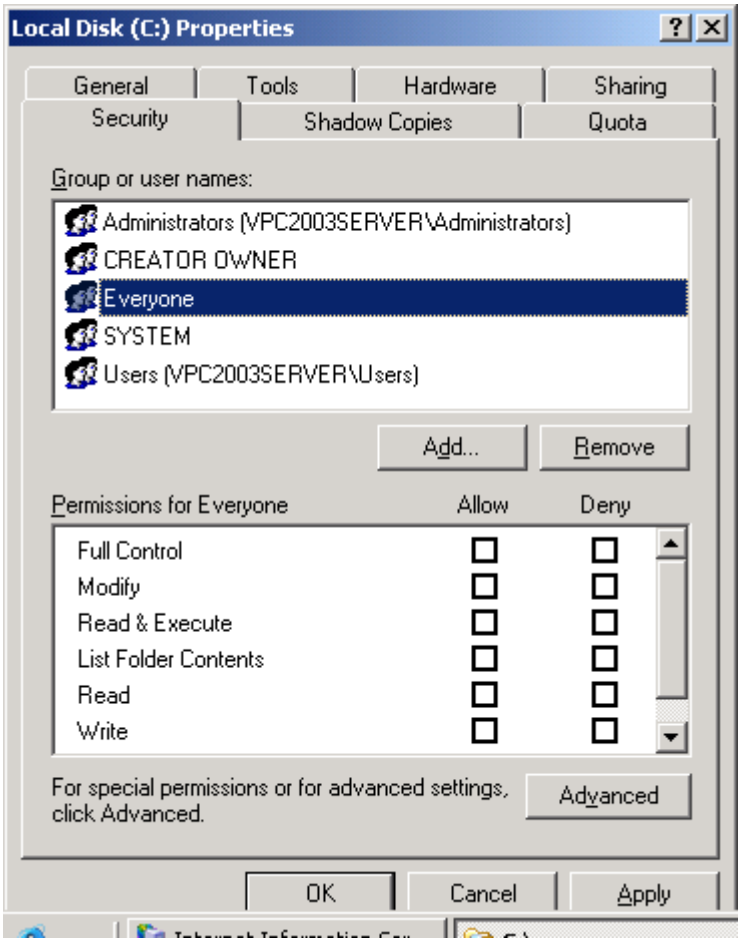


Figure 2 – Users with access to the C: drive

Notice that the user, Everyone has no access. (There are no checkboxes checked in the bottom portion of the window.) This is the default setting in Windows Server 2003, but is not the case in all previous operating systems. I prefer to just click on the Remove button, to remove the Everyone user from the root node of the disk completely. Allowing the Everyone user to have any access to the root node is a serious security risk. I prefer not to grant the Everyone user access to anything anywhere on the disk.

When you install IIS, users called IUSR_machine and IWAM_machine are created. (where the name of your machine is substituted for "machine") IUSR_machine is the Internet guest account. This user must have access to the files in your web directory. I prefer to give this user read-only access by default, specifying execute rights only to directories where it's needed. To setup these rights, go to the directory you created for web files, right-click, select properties, and go to the Security page. Click the Add button to add a user to the access list, and fill in the IUSR_ name, as shown in figure 3, using your machine name instead of "VPC2003server". Then Click OK.

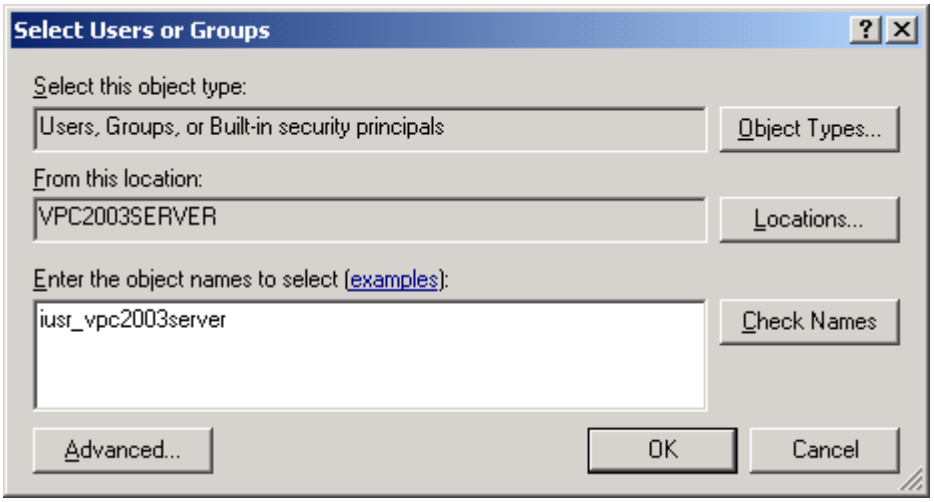


Figure 3 – Adding access rights for the IUSR account

The user is now shown in the list of users with access to your directory. You must check the appropriate boxes to specify the kind of access this user is to have. (Figure 4)

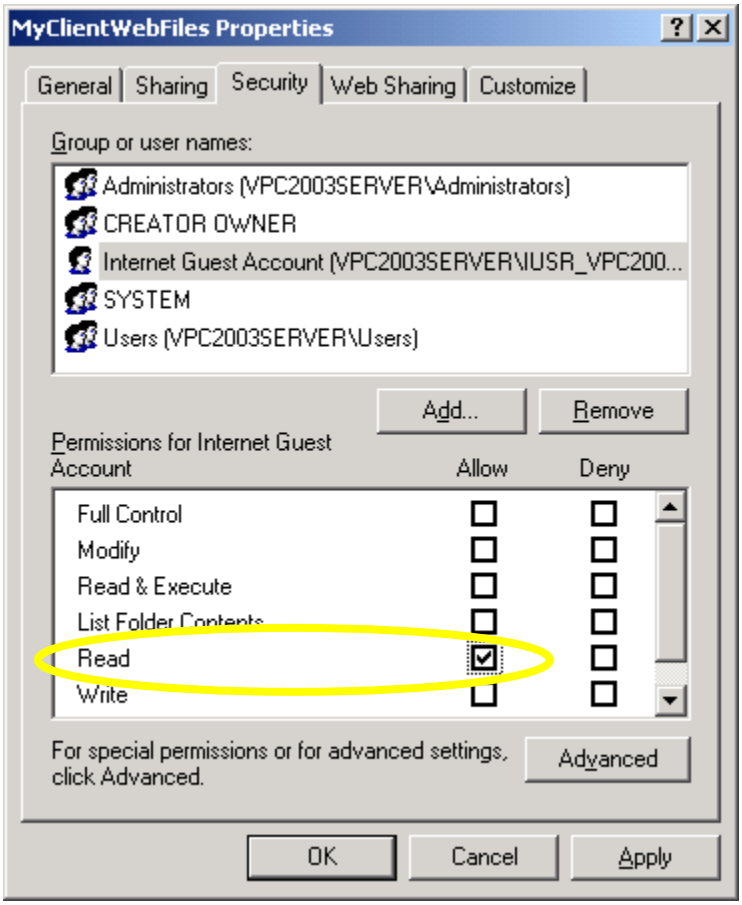


Figure 4 – Specifying access rights for a user

As I mentioned previously, I allow the IUSR account read-only access by default, so I make sure that's the only box that's checked. This is all that's needed for a user to view HTML files and image files. If the user must be able to run ASP, ASPX or other executable or script files, I will allow Read & Execute rights to the folder. I prefer to keep these kinds of files in a subdirectory under the regular web files. I then grant the IUSR account read and execute privileges to that subdirectory.

Caution: Do not try to change the name of your IUSR account or its privileges in the Windows Computer Management utility. These properties cannot be changed without adversely affecting the way your web apps work. If you must rename your machine, you must uninstall and reinstall IIS to create an account with the correct name.

IIS

Internet Information Services (IIS) is a key component on any Windows web server. According to the Windows help file, IIS is defined as "Software services that support Web site creation, configuration, and management, along with other Internet functions. Internet Information Services include Network News Transfer Protocol (NNTP), File Transfer Protocol (FTP), and Simple Mail Transfer Protocol (SMTP)." They left HyperText Transfer Protocol (HTTP) out of that description, but IIS includes HTTP too. This paper will only be covering HTTP and FTP.

Note that the version of IIS you install is tied to the OS on the machine. IIS 5 is what runs under Windows 2000 Server and IIS 6 is what runs under Windows Server 2003. You can determine the version by viewing the properties of the inetinfo.exe file, which resides in \Windows\System32\Inetsrv.

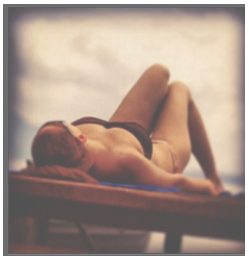
Installing IIS

IIS isn't installed automatically when you install the OS, but it's easy to add. From the "Add or Remove Programs" option on the Control Panel, click on the "Add/Remove Windows Components" button on the left side. (Figure 5)

win 7 ultimate functions this way

Control panel\programs and features\Turn win features on or offf

Config was taken through internet as updates.Restart and here we are:



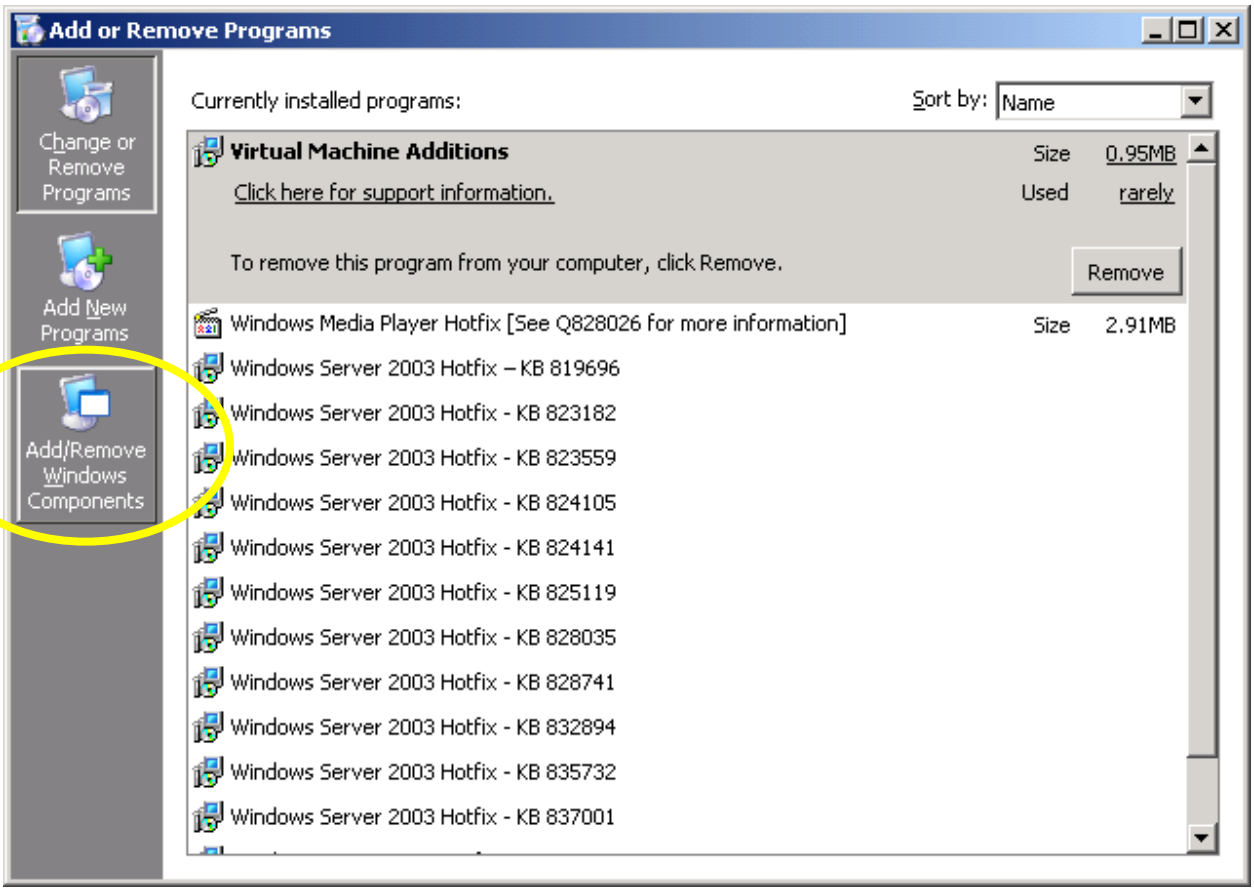


Figure 5 – Click the "Add/Remove Windows Components button on the "Add or Remove Programs" screen

A list of all Windows components is displayed, showing check marks by components already installed and an unchecked box next to components that are not installed. (Figure 6) From the list that's displayed, click on "Application Server" to highlight that selection, and then click on the Details button. Note that on some operating systems, "IIS" and "Management and Monitoring Tools" are listed on the main screen after you click "Add/Remove Windows Components". The diagrams shown are where these options appear in Windows Server 2003.

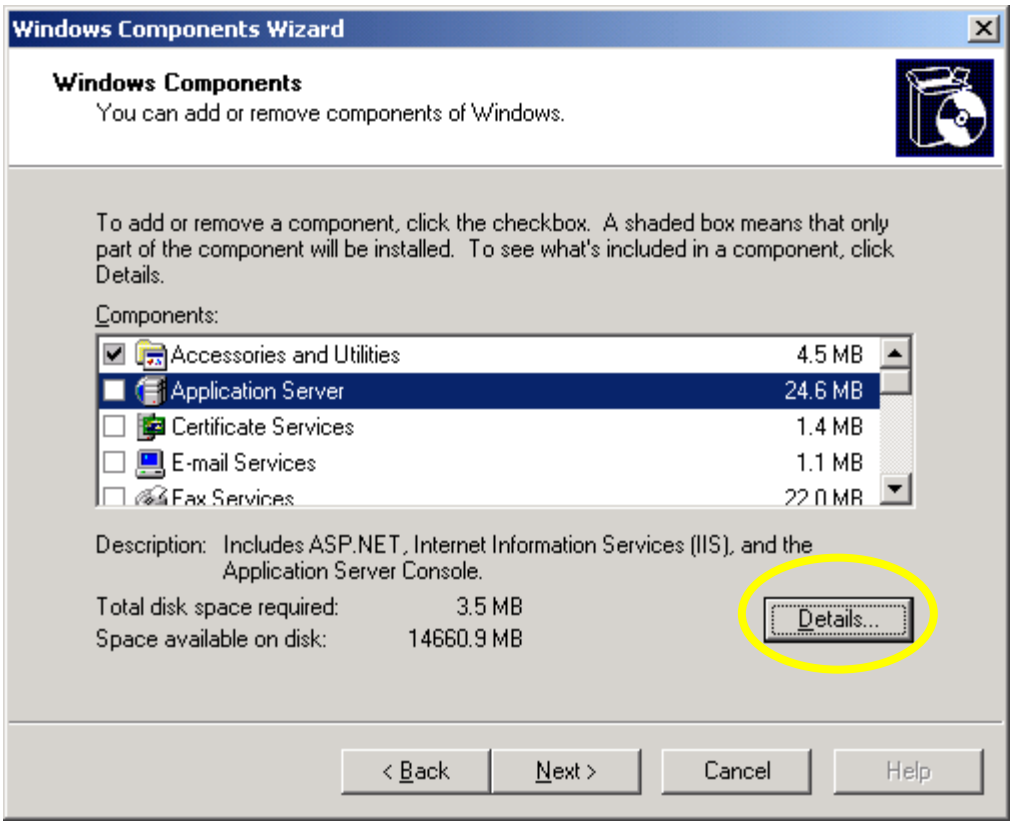


Figure 6 – On the Windows Components screen, highlight "Application Server" and click on the Details button.

The next screen that's displayed is the one where you can specify that IIS is to be installed. (Figure 7) You'll want to install the "Application Server Console" too, which most people refer to as the "Microsoft Management Console" or "MMC". MMC is the interface you'll use for configuring IIS. In Windows Server 2003, checking "Application Server Console" will automatically check the options to install COM+ and IIS as well.

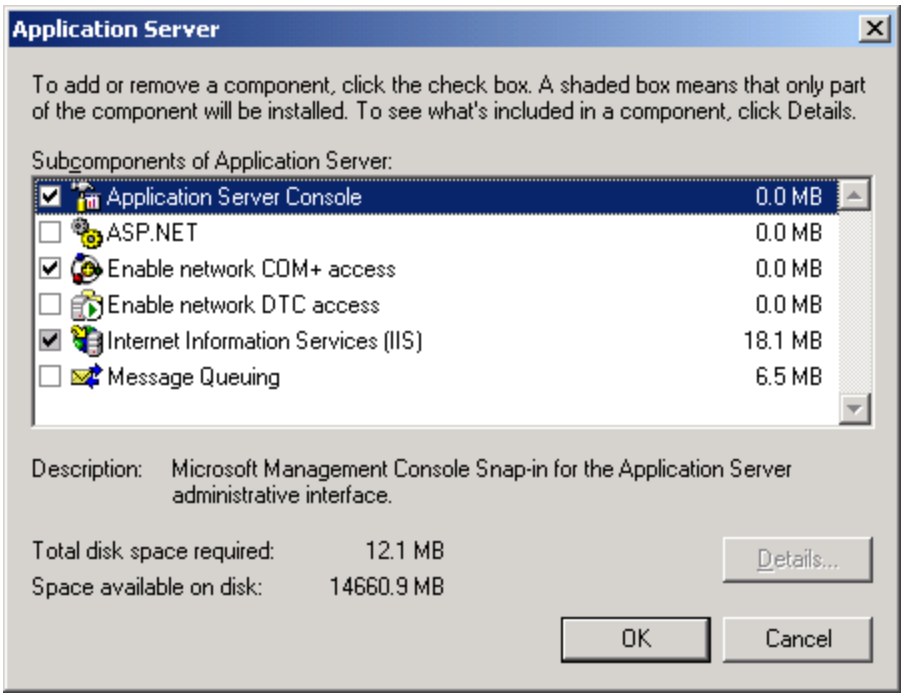


Figure 7 – Selection of IIS and related components for installation

Although IIS is checked, it still won't install all the components we need. Highlight IIS and click on the Details button. This displays an additional screen. (Figure 8)

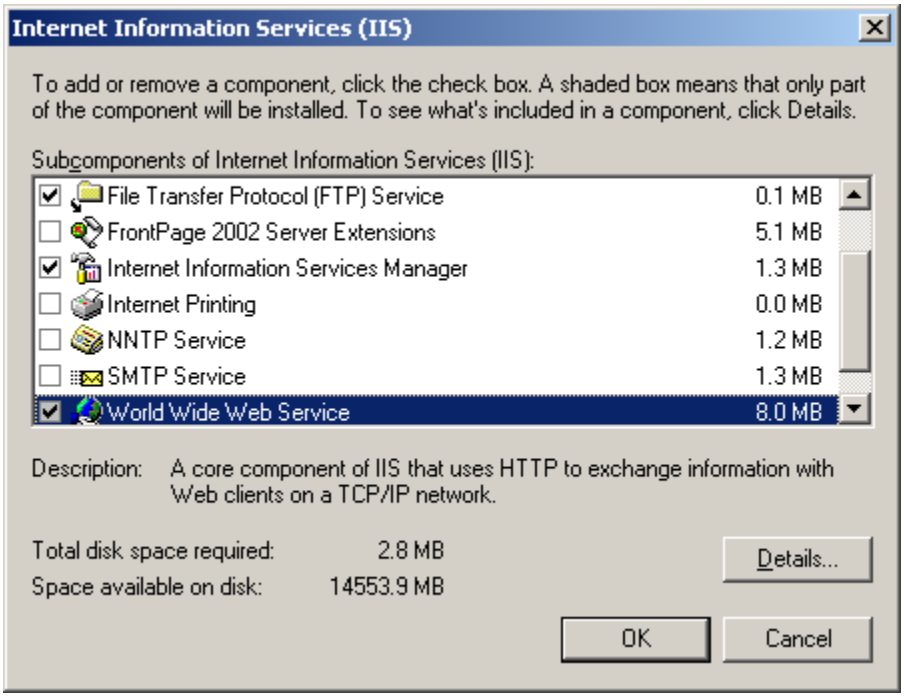


Figure 8 – Additional screen for IIS installation

"File Transfer Protocol (FTP) Server" is not checked by default. Nor is "World Wide Web Services", but we need those too. Make sure you check these options, and then click OK.

Although we are not dealing with ASP in this paper, if you need to install ASP, you'll need to highlight "World Wide Web Service" and click the Details button. From there, you can include ASP with your installation.

Click OK and then the Next button. You may need to have your Windows installation disk handy to complete the installation.

Configuring IIS

There's a lot to configuring IIS, and I can't possibly cover all of it in a single paper (even just for HTTP and FTP). I do cover the most common situations.

There are a few different ways to open MMC for configuring IIS. One way is to go to the Control Panel, select Administrative Tools and then select Internet Information Services Manager. You could also go to Start | Run and type MMC into the command box. But then you have to select "File | Open" and find the IIS.MSC file, which is in the \Windows\System32\Inetsrv directory. In Windows Server 2003, you should have Internet Information Services Manager on your Programs menu. It's easiest to open the configuration screen from there, or even to right-click on that option and drag it to your desktop to create a shortcut. Regardless of which technique you use to open MMC, the screen shown in figure 9 is what is used.

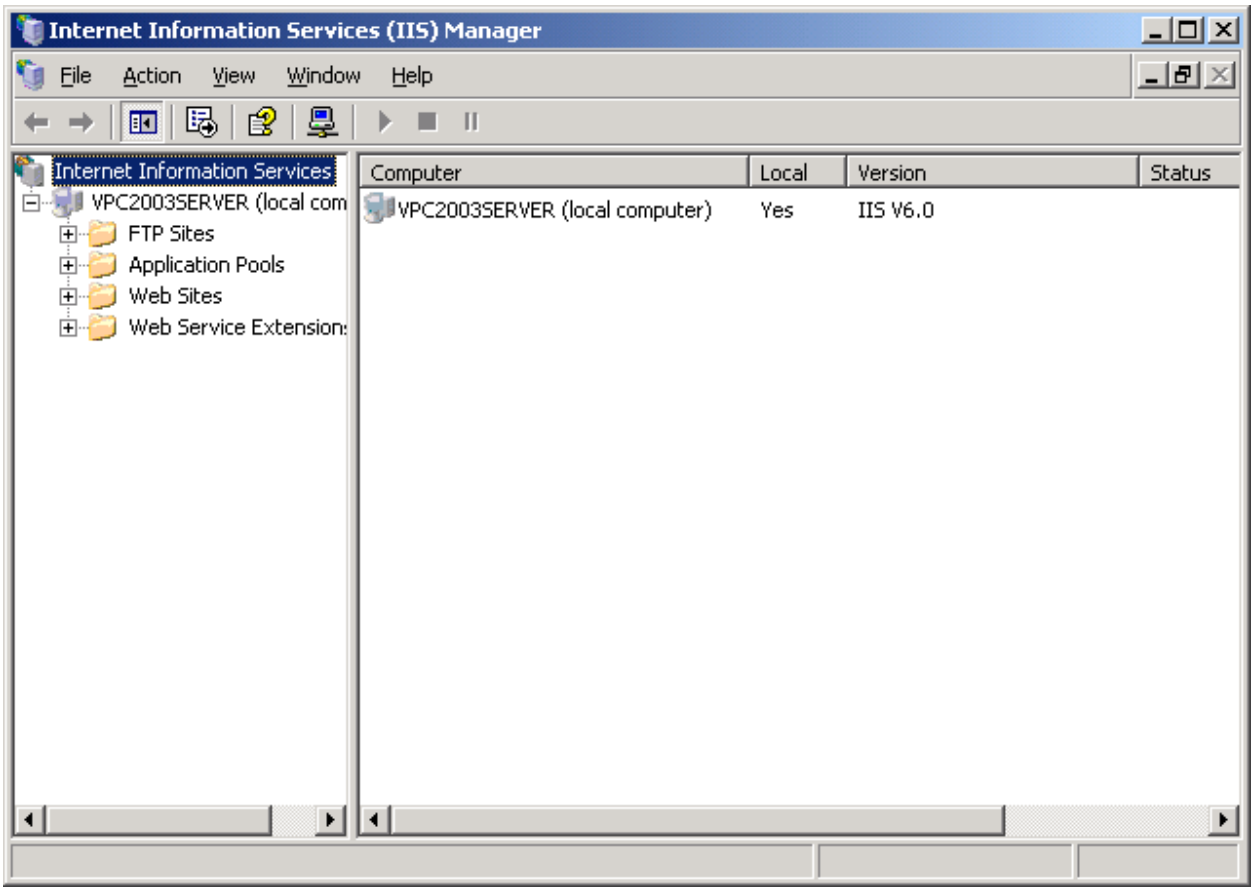


Figure 9 – Main configuration screen for IIS

Creating a Web Site

There is a default web site listed under the "Web Sites" branch of the treeview. If you aren't using a true server OS, then that is the web site you must use, as only a server OS will allow you to create any additional web sites.

On a real server, I don't use the default web site, again, just to throw off hackers. Create a new web site by right-clicking on the Web Sites folder, and choosing "New | Web site" from the menu. A wizard starts and guides you through the setup. Choose a name for the new web site. This name doesn't have to be the same as the directory you created for it, but the same name or something similar helps you keep things straight.

For now, just use the defaults for the IP Address and Port Settings screen (figure 10). Then choose the directory you created for the site when it asks for the path. (Figure 13) When you get to the Access Permissions page, be sure to un-check the "Run scripts" checkbox if users only need read access to your web directory. (Figure 14)

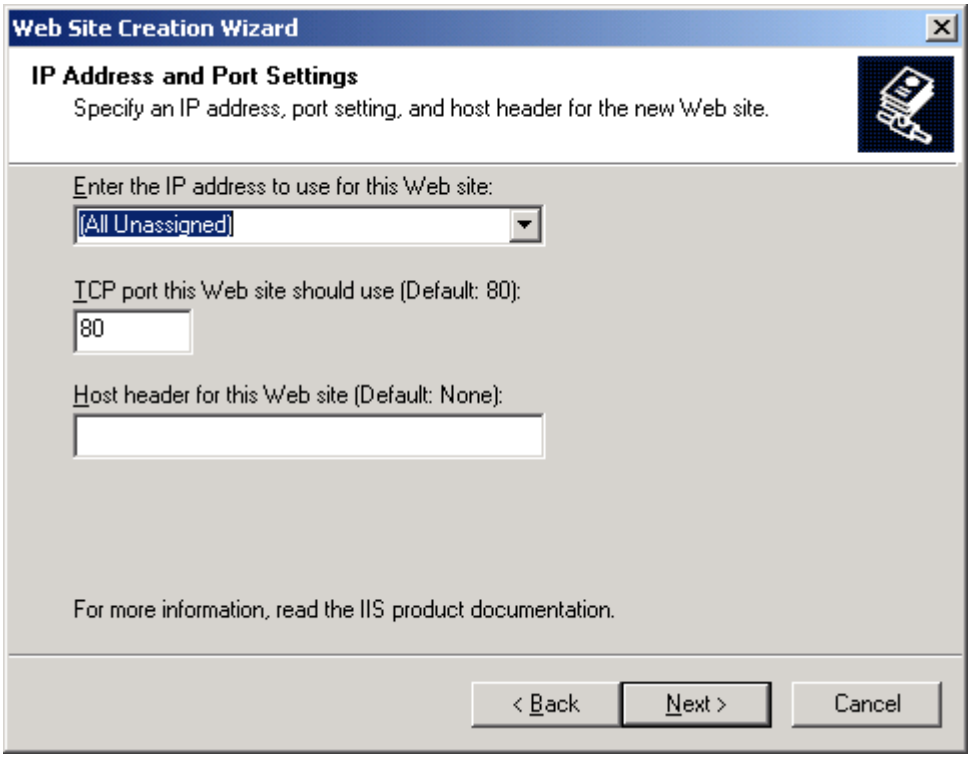


Figure 10 – IP Address and Port Settings for new site

Note that I've left the IP address as "All Unassigned". An IP address will only appear for selection in this list if you've already defined it. (See the "Configure Your Network Connection section later in this paper.) Using "All Unassigned" is fine if you only have one IP address or one web site. To quote the Windows help on this topic, "If you do not assign a specific IP address, this site responds to all IP addresses assigned to this computer and not assigned to other sites, which makes this the default Web site."

Next, specify the directory where the web site is located on your server, and the user access rights. (See figures 11 and 12) Click Next and then let the wizard finish, and your new web site has been created. The new web site will appear under the "Web Sites" node of the treeview on the IIS configuration screen.

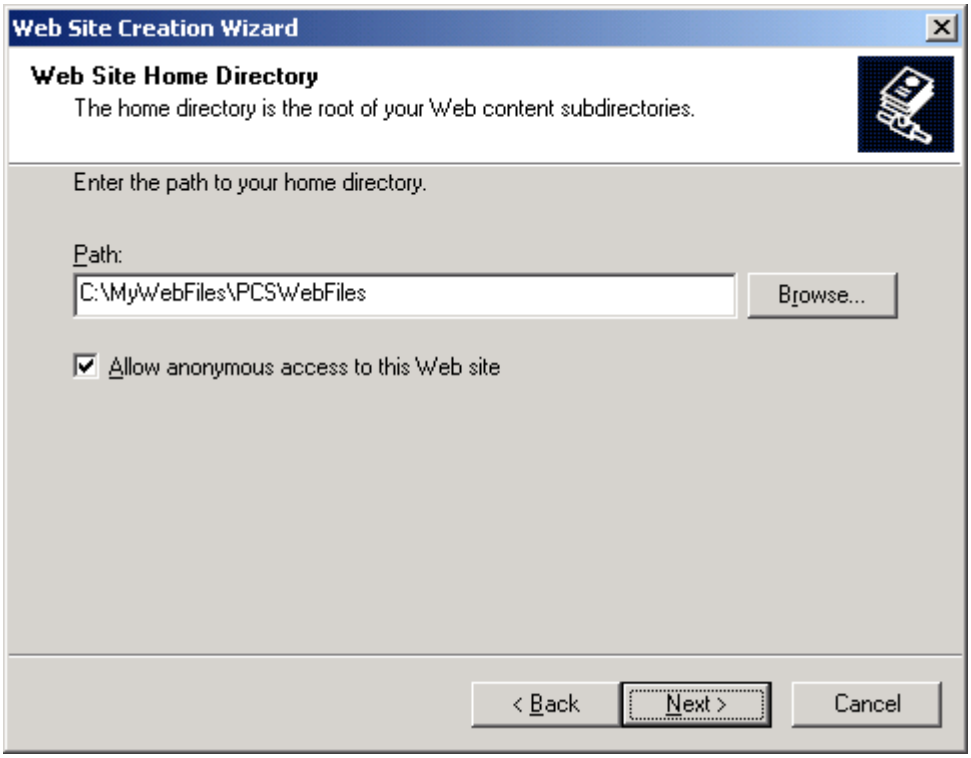


Figure 11 – Setting the path for a new site

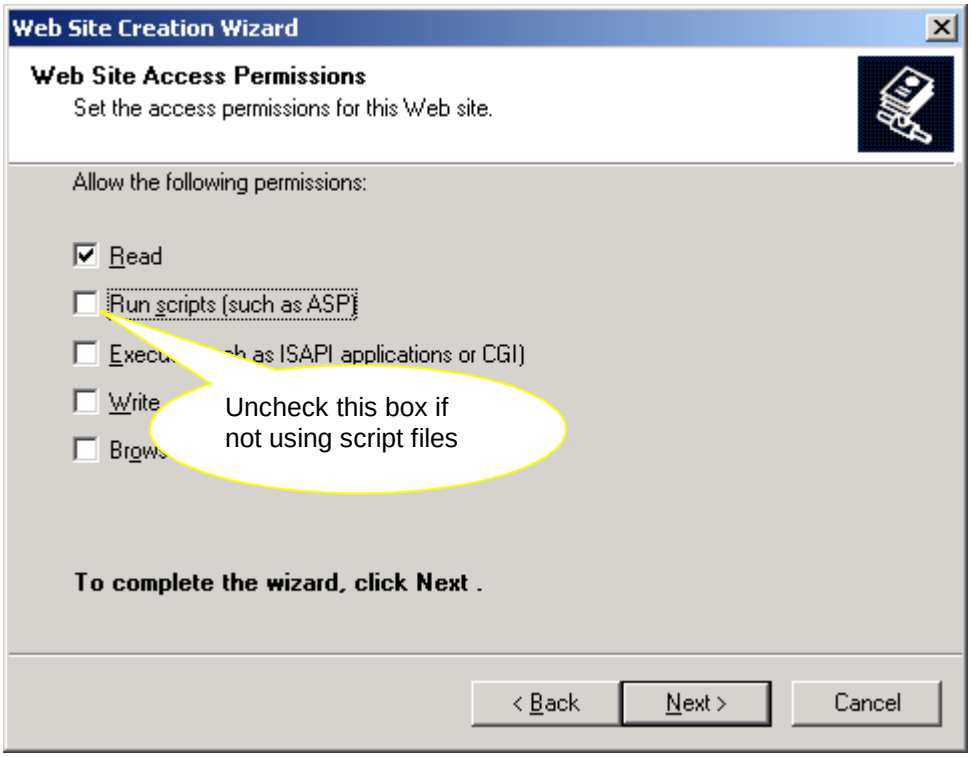


Figure 12 – Setting access rights for the new site

Configuring the new web site

To get to the configuration options for the newly added web site, right-click on the site, and choose “Properties”. The main configuration screen is displayed. (Figure 13)

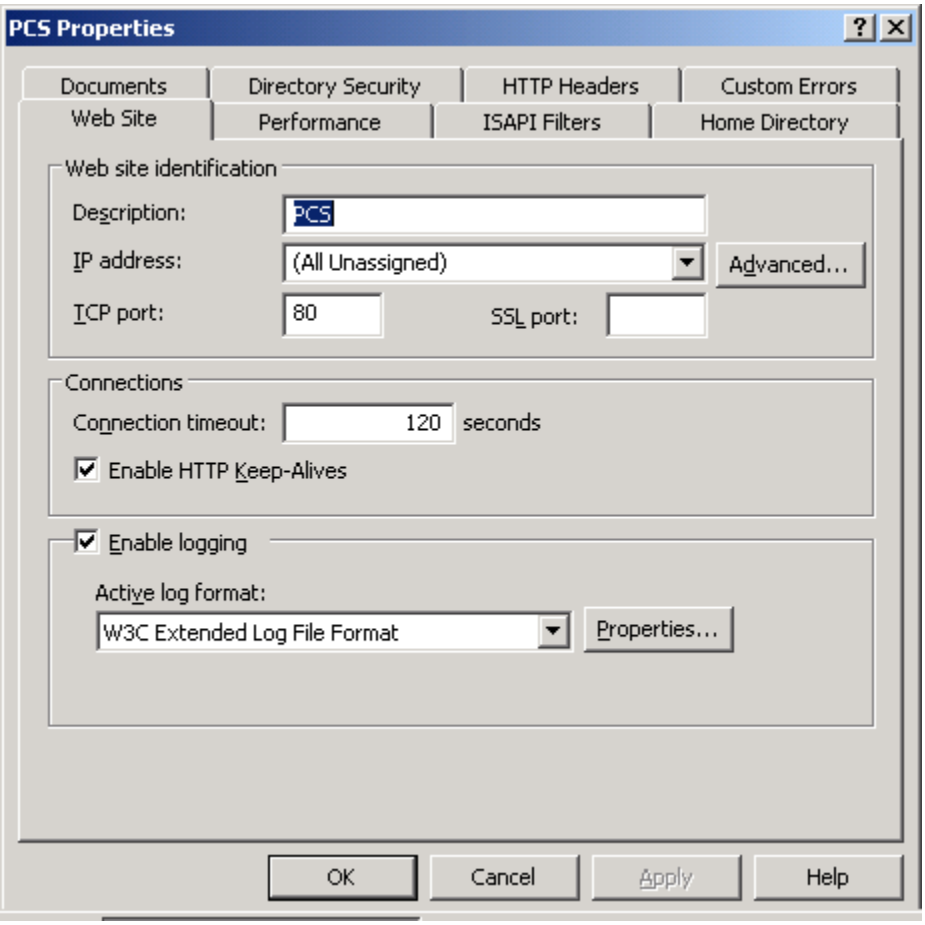


Figure 13 – Web site main configuration screen

I won’t discuss all the options here in detail, but will discuss some key elements. First, let’s focus on security.

Directory Security

Click the “Directory Security” tab and then click the Edit button. (Figure 14)

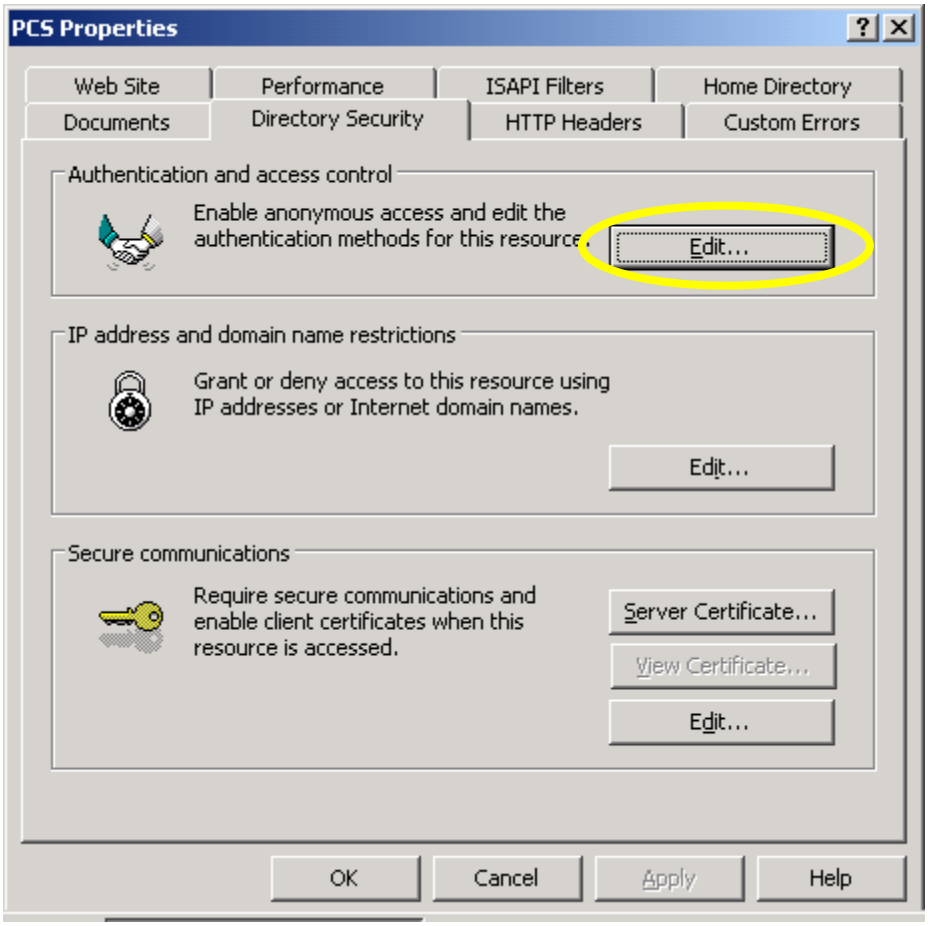


Figure 14 – Web site directory security

The Windows Authentication screen is shown. (Figure 15) In most cases, you want to leave "Enable anonymous access" checked. By un-checking this box, whenever someone tries to access the site, a Windows login screen pops-up, asking for a user name and password. You would do this on any subdirectory where you want the files available to the outside world, but only to authorized users.

If you do choose not to allow anonymous access to a directory, you have several choices for the authentication method. The method you choose is dependent on your specific circumstances. Here are the options:

- *Integrated Windows authentication* uses cryptography to send the user name and password from the browser to the server
- *Digest Authentication* uses a hash value and only works with Active Directory
- *Basic Authentication* sends the user name and password across the internet in plain text. If you must use this method and are concerned about security, then you must use a secure site for your login (one that uses a certificate)
- *.NET Passport Authentication* uses a web service for authentication

For more information on each of these authentication methods, visit the Microsoft web site on this topic at http://support.microsoft.com/default.aspx?scid=kb;en-us;324276

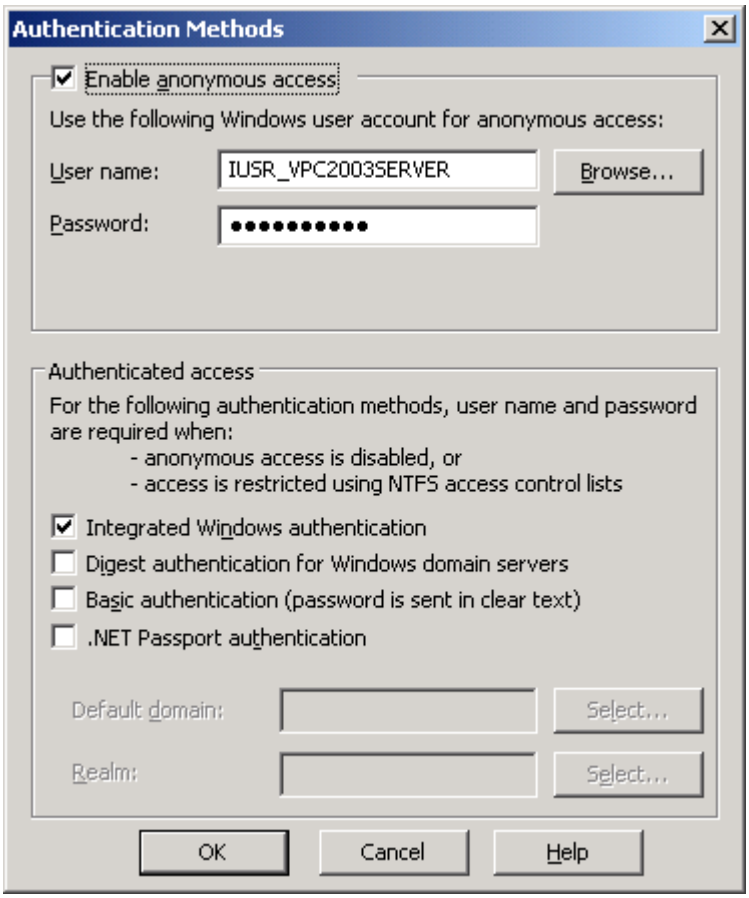


Figure 15 – Setting web site authentication

Documents

The Documents page is used to define the file returned to the browser when the user doesn't specify a file name in the URL. For example, if you navigate to http://www.peisch.com you're not specifying a file name, but the page that's displayed is Index.html. This is because Index.html is the first file found that's in the list of default documents. (Figure 16) Notice that Index.html is not the first file in the list. The reason this is the file displayed is because there is no Default.htm, Default.asp or Index.htm file in the directory. If instead you navigate to http://www.peisch.com/downloads.html then you're specifically saying you want to see the downloads.html file instead of the default document.

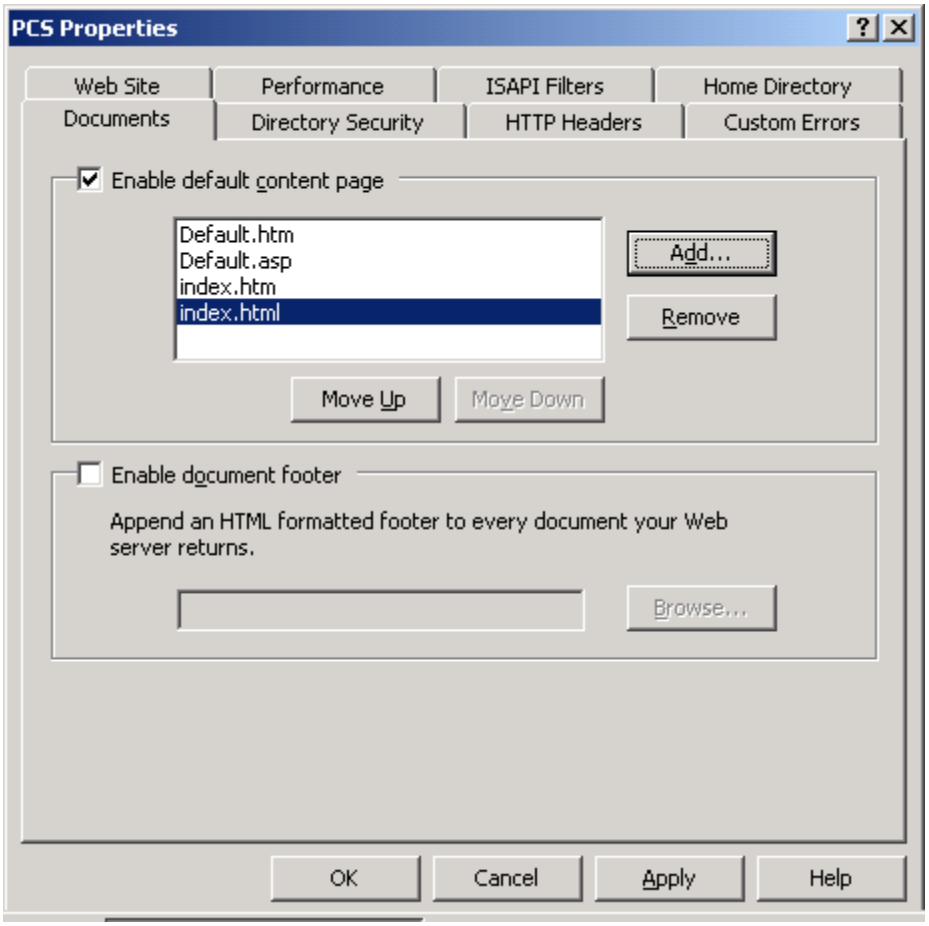


Figure 16 – Default documents

Another good security measure is to use a default document name that doesn't start with "Default" or "Index". There's at least one script I know of that replaces the Default.htm, Default.asp, Index.htm and Index.asp files in all your virtual directories with its own version. If you don't list those as default documents, and use something else for your default document, even if those files are present, they can never affect your web site.

Application Pool/Application Protection

An Application Pool is new to Windows Server 2003. Previous operating systems used "Application Protection" instead. Application protection was a way to isolate web sites so that if one crashed it didn't affect other sites running on the same server. In Windows 2000 Server the choices are High, Medium and Low. (See figure 17)

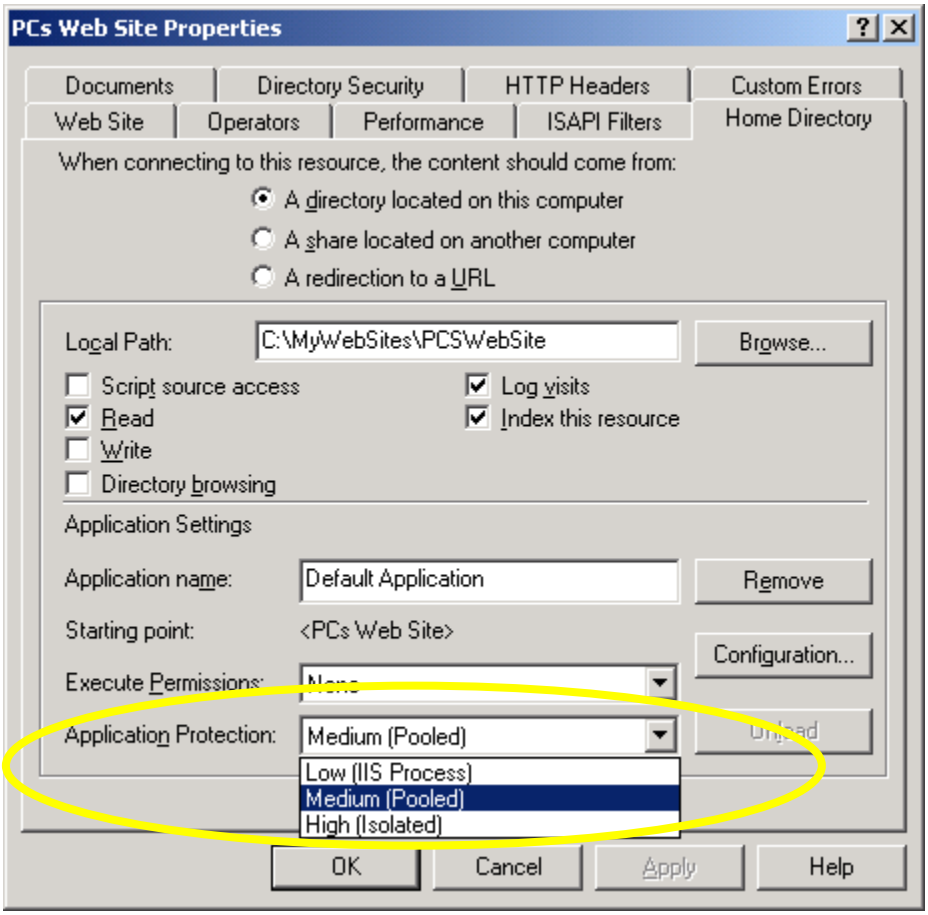


Figure 17 – Application Protection options in Windows 2000 Server

Although it may sound attractive to set the level to High for everything, the trade-off is a loss of performance. The "protection" didn't always work as well as expected either. See http://www.microsoft.com/windows2000/en/server/iis/default.asp?url=/windows2000/en/server/iis/htm/core/iiwarndc.htm for an explanation of each of these settings.

Windows Server 2003 has introduced "Application Pools" to replace Application Protection. See http://www.developer.com/net/asp/article.php/2245511 and http://www.microsoft.com/windowsserver2003/techinfo/training/iis.mspx for information about Application Pools. "Application Pools" is one of the nodes listed in the treeview. (See figure 18)

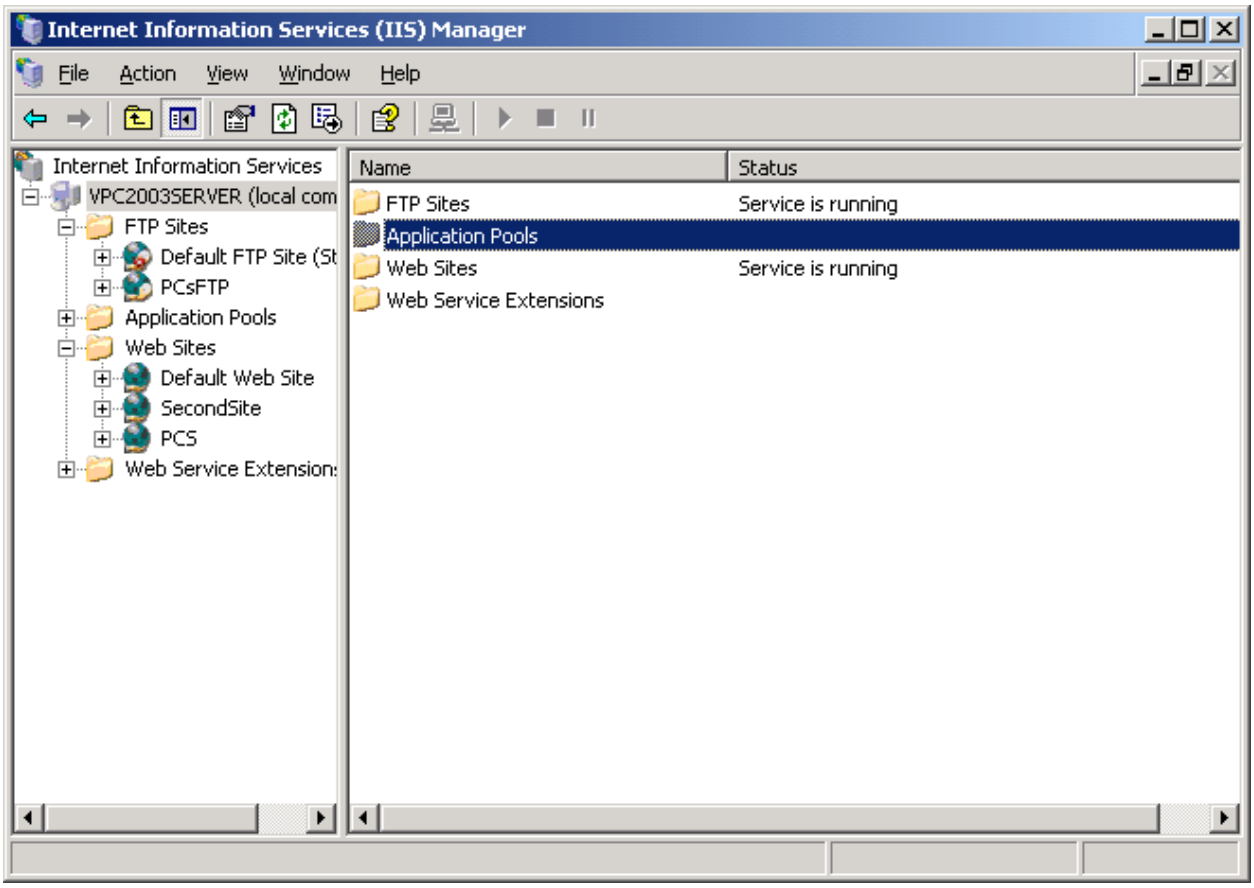


Figure 18 – Application pools

To create a new pool, right-click on “Application Pools” in the right side window and select “New”, then “Application Pool”. The screen in figure 19 is shown.

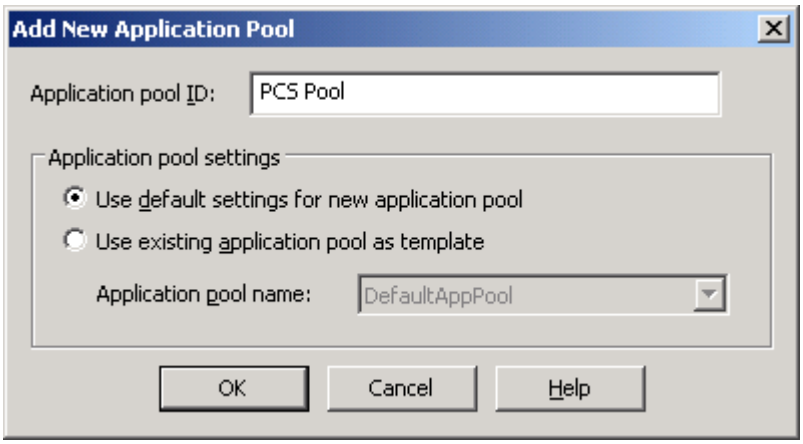


Figure 19 – Creating a new application pool

Click OK. The new application pool now appears under the "Application Pools" node of the treeview, and is available from the Application Pool dropdown in the properties for the web site. (See figure 20)

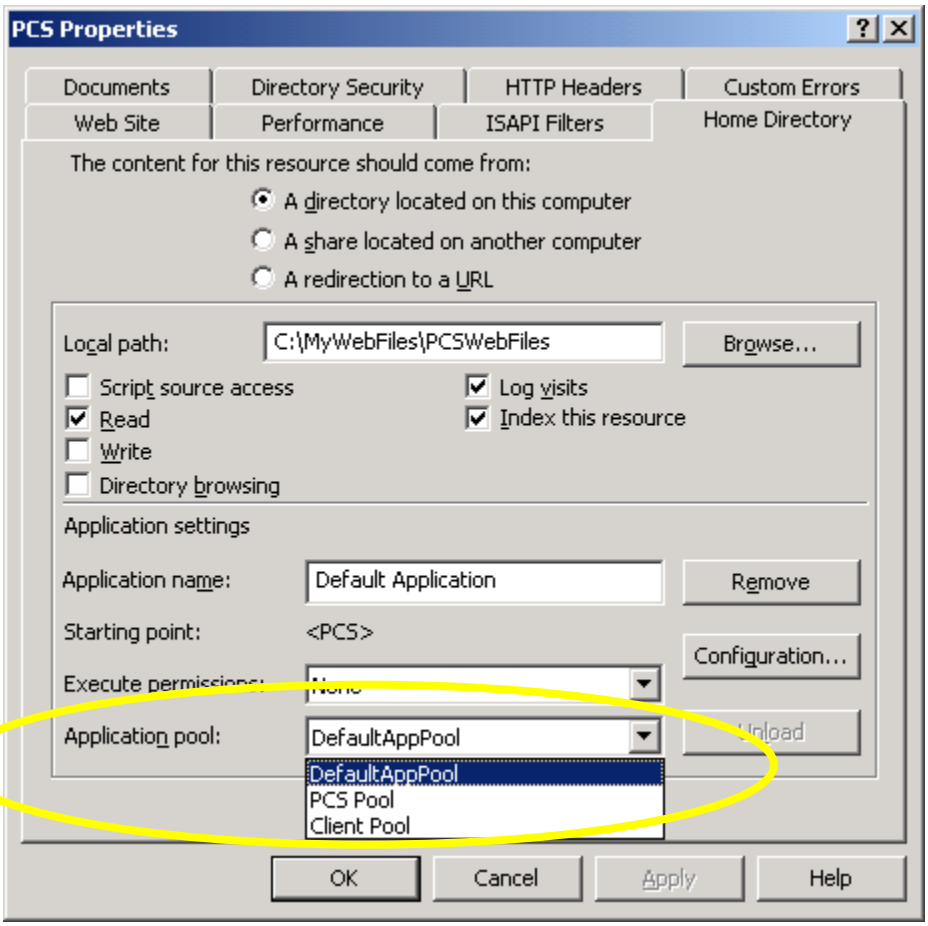


Figure 20 – Choosing an application pool for a site

The decision about how to define and choose your application pools is beyond the scope of this paper. The links I've given you for information on application pools should help give you a better understanding as to how application pools work and how to decide how to set them up.

Setting up Web Service Extensions

Figure 21 shows "Web Service Extensions" as the highlighted item in the treeview. A web service extension is a shortcut that connects a filename extension in a URL with an executable. For example, you may want any URL that includes a filename with an extension of .ASP to run the executable asp.dll. This is the standard program that runs ASP files.

It's important to note that extensions you configure via this interface are global for all web sites on the server. You may want to define extensions that are specific to your app. We'll discuss that shortly.

It’s not necessary to define or allow an extension for HTM, HTML or image file extensions, as these are handled by the browser rather than the server.

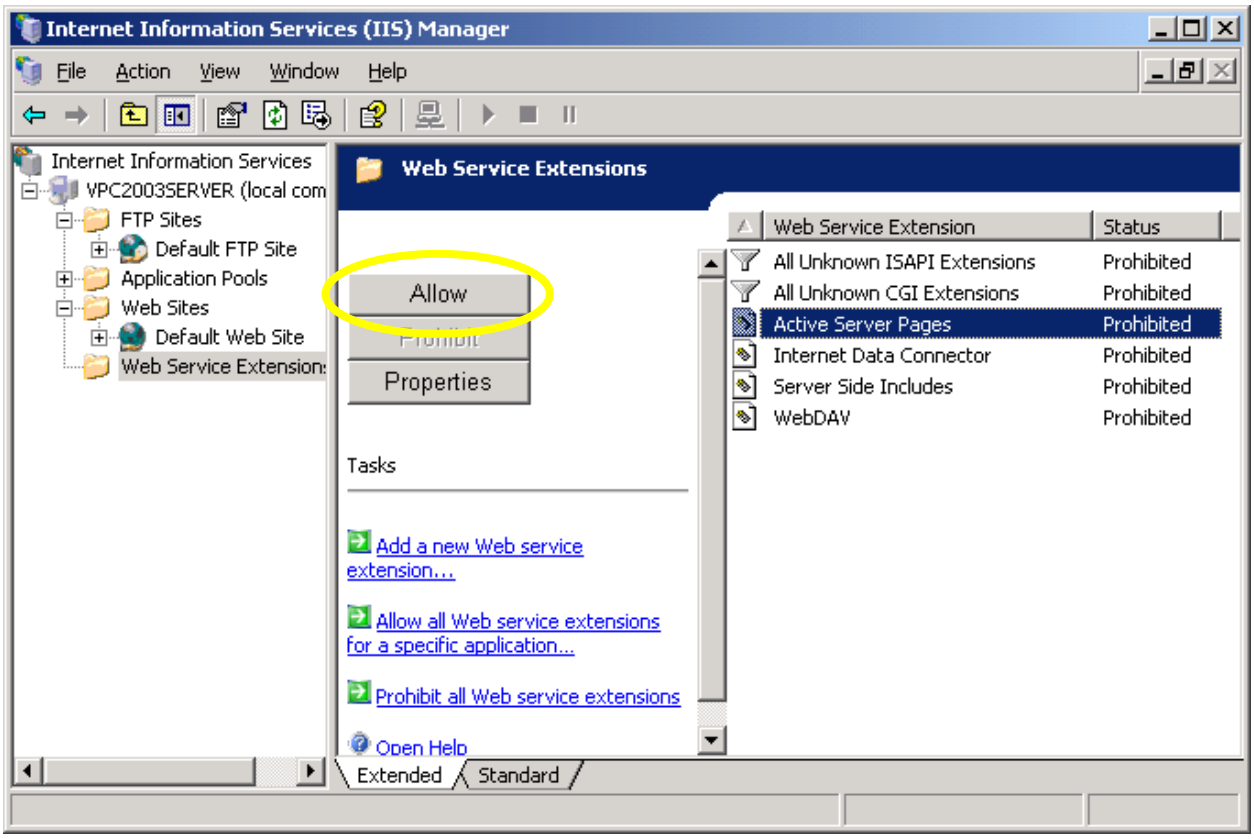


Figure 21 – Allowing standard Web Service Extensions

If you have an application that uses a specific extension, say .PCS, then you need to add that as an allowed extension. At the time you add a web service extension, you also specify the program that is to be run when the user asks for a file with that extension. This can be any program, but the problem is, you can only specify one program that is used globally and you may not want that extension to be used by all the web sites on this server. Although the link that says “Allow all Web Service extensions for a specific application” might sound like what you want, what Microsoft considers a “specific application” is one of the standard applications that are already listed on the right side of the screen. Unfortunately, your only choice if you don’t want your web service extensions to apply to all web sites on the entire machine is to click “Allow” for “All Unknown ISAPI Extensions”.

This interface for allowing or prohibiting extensions globally is unique to Windows Server 2003. Only under Windows Server 2003 are all extensions disabled by default. Note that prohibiting extensions doesn’t remove them from the list of mappings (which I’ll get to shortly), but will prevent them from working. About the only thing I consider this new level of configuration good for is if you need to globally shut off web service extensions for some reason.

Configuration of web service extensions for a specific web site uses the same interface for all operating systems. Right click on the web site name in the treeview and select “Properties” from

the popup menu. From the screen that comes up, click the “Home Directory” tab. The screen shown in figure 22 is displayed.

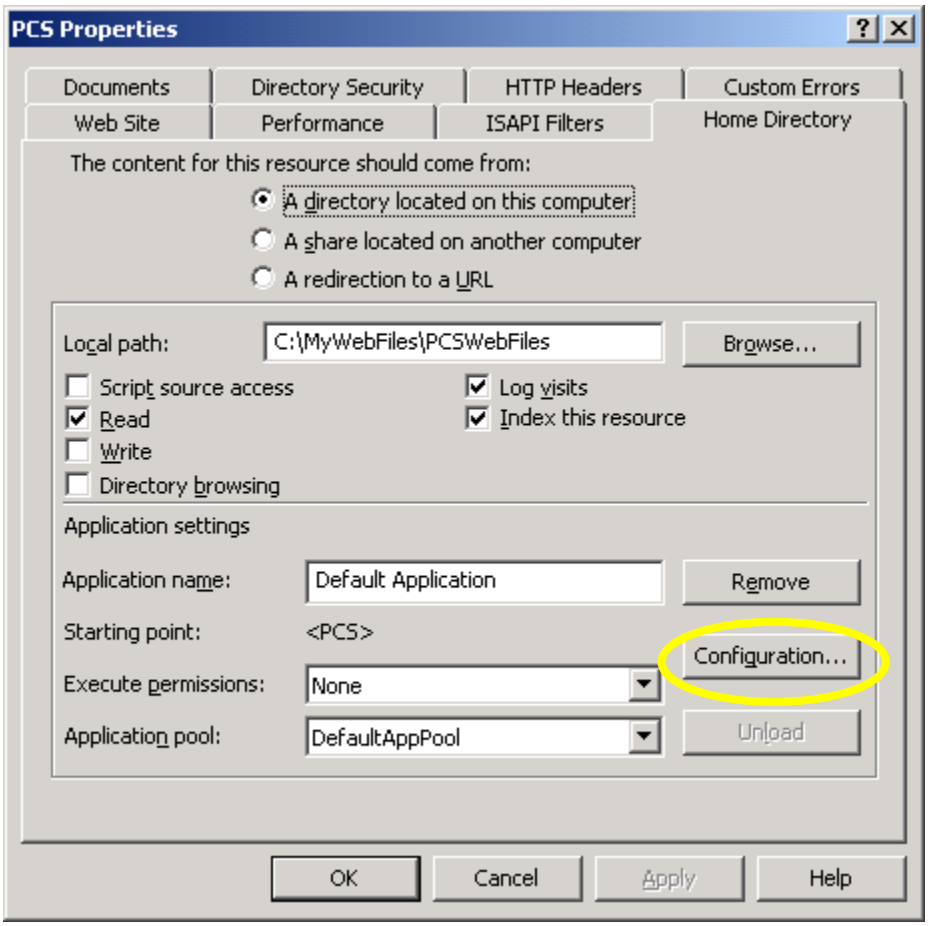


Figure 22 – The Home Directory page

Click on the Configuration button. In some cases, this button may be disabled, and the button above it says “Create”. If you see that situation, click the Create button, and the Configuration button will be enabled. After clicking the Configuration button, the screen shown in figure 23 is displayed.

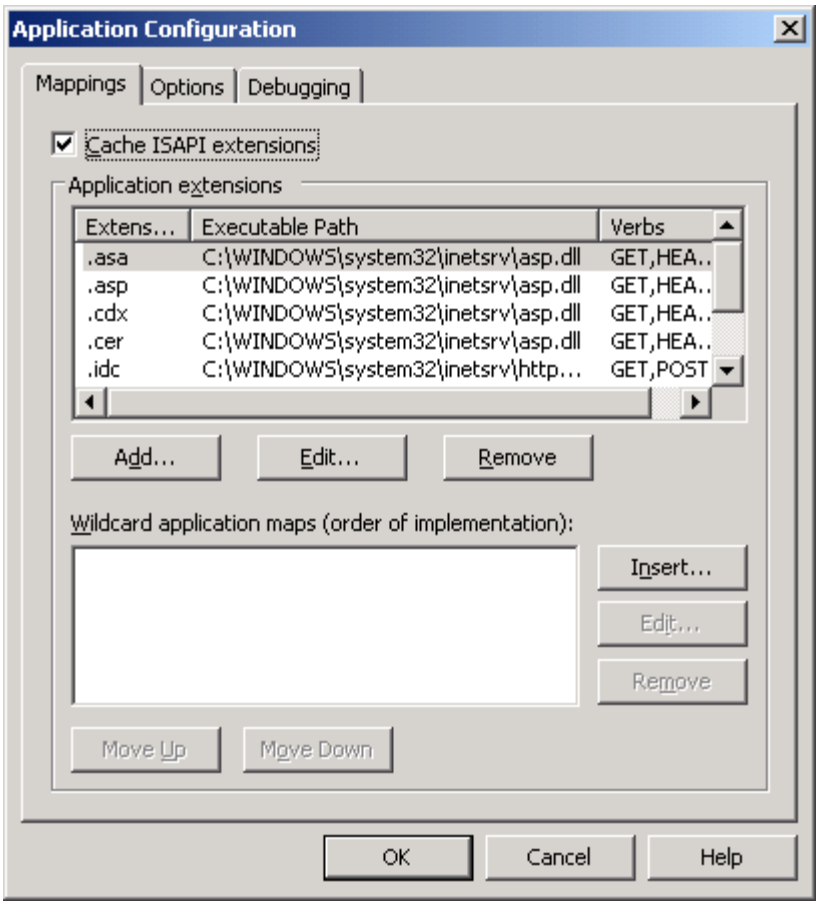


Figure 23 – List of Web Service Extensions for a site

Notice there's a long list of extensions shown. One of the first things I do after creating a web site is to remove all the unwanted extensions to be sure that they are not active. In our example, the only extensions we want are ASP, which is already in the list, and PCS, which is not.

To remove unwanted extensions, click on the extension in the list and then click the Remove button. A confirmation dialog is shown.

To add a new extension, click the Add button. The screen in figure 24 is shown.

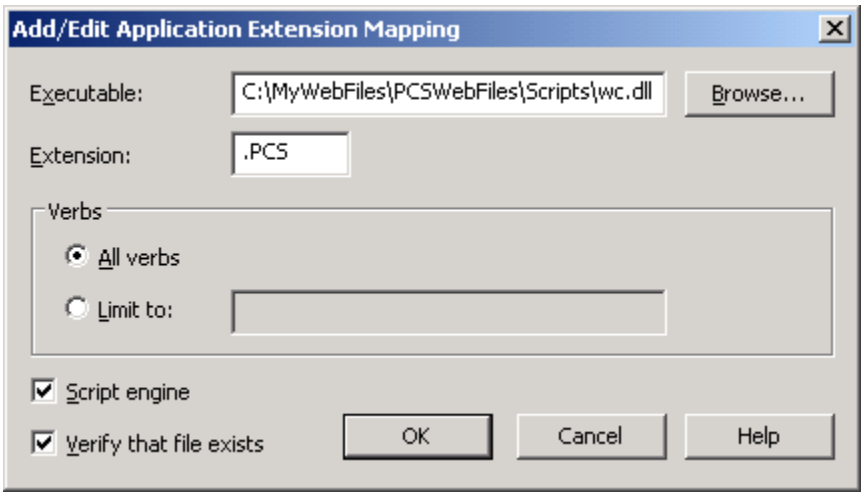


Figure 24 – Adding an extension

Choose the program you want to be invoked when a URL has the specified extension, and enter the new extension. Note that "Verify that file exists" is checked by default. In many situations you want to uncheck this. Having this box checked means that the file named in the URL must exist in your virtual directory. If the file doesn't exist, the browser will receive an error. If the file named in the URL is only a method name in the program you're calling, you want to make sure this box is not checked.

Another security feature is to select "Limit to" under verbs. If you know the extension should only be calling your program from a form's Post operation (i.e. clicking the "Submit" button on a form), then you should put POST in the list and nothing else. If the extension may use a query string to call your program (i.e. a hyperlink), then you'll want to add GET to the list of verbs. The verbs in the list should be separated by commas.

Leave the Script Engine box checked. This determines if the extension can run in a directory that doesn't allow scripts or executables. If you've only allowed read access for the main directory of your virtual, then this box needs to be checked. If you've allowed scripts or executables to be run in this directory, then checking this box is irrelevant.

Configure Your Network Connection

You also need to set the correct properties for TCP/IP on your network connection. From the Control Panel, choose "Network connections" and select your network card. Click the Properties button on the screen that comes up. The next screen shows you a list of all the network protocols installed on your computer. Make sure TCP/IP is checked. (Figure 25)

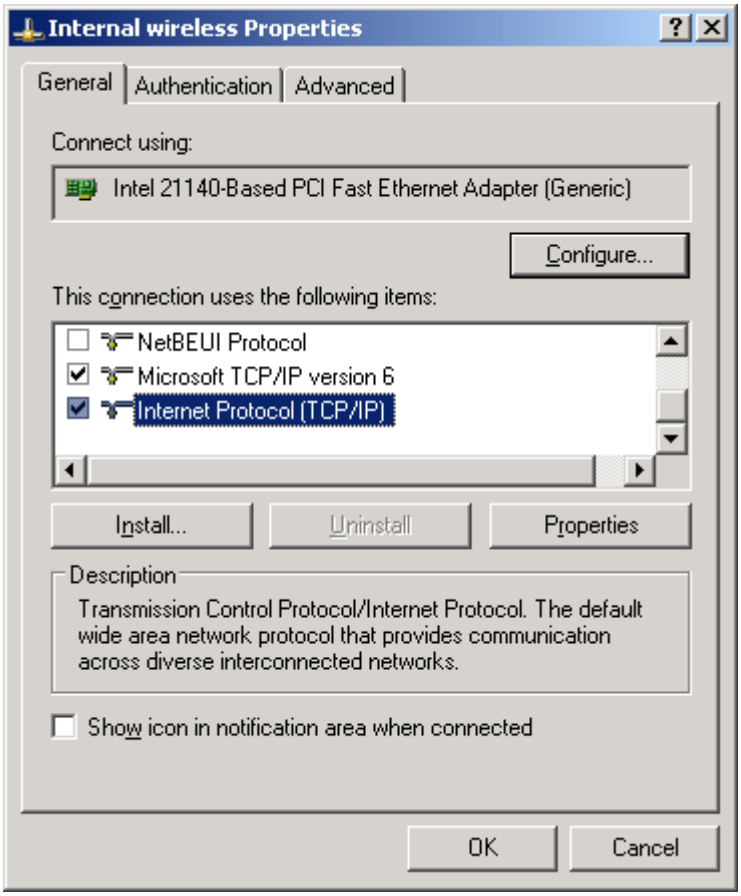


Figure 25 – Protocol selection for a network card

Click on the Properties button.

For simplicity's sake, I'm going to assume you have a fixed IP address for the server. This is preferred, but if you're just running a staging server, you may be able to get away with using a Dynamic DNS service. Some of these are listed in the Resources section at the end of this paper.

You configure your TCP/IP settings on the screen shown in figure 26.

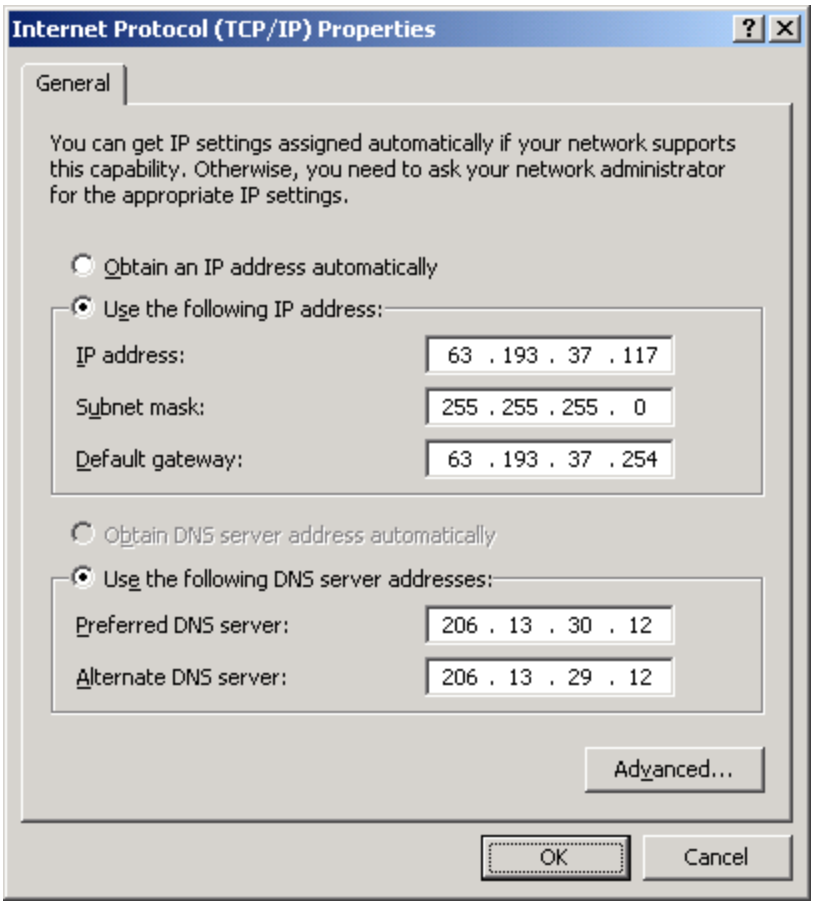


Figure 26 – TCP/IP configuration

Choose the "Use the following IP address" option and the "Use the following DNS server addresses" option. What goes into these fields must be given to you by your Internet Service Provider. The IP address will be your fixed IP address. Your ISP will tell you what to use for the other settings. Click OK and then close the window for your network connection.

We haven't configured a domain name for the server yet, but this server is now accessible from the outside world by using the IP address. For example, based on the settings I've discussed, you could navigate to http://63.193.37.117/ from any browser and view the web site.

Domain Names

Getting to your web site by using an IP address is fine—if all you're doing is getting there by links from somewhere else. If you want to tell people how to get to your site, you want to be able to give them a domain name. The first thing you need to do to setup a domain name for the server is select a name that doesn't already belong to someone else. To check if your potential domain name is already in use, go to http://whois.com/.

Normally, you wouldn't be configuring a domain on your server directly. Unless you are an ISP and own your own IP address, you wouldn't be running a domain name server, and therefore, wouldn't configure a domain name for an IP address. What most people who run web servers do is register a domain name that points to a domain name server (DNS server) of the ISP. The DNS server at your ISP will then direct traffic for your domain name to your IP Address. If you're interested in understanding more about how DNS servers work, I recommend you go to http://computer.howstuffworks.com/dns.htm.

Domain Name Registration Services

If you want to have a domain name that people can use to reach your web site, then you need to register that domain name with one of the registration services. The site http://www.internic.com maintains public information regarding domain name registration services. I recommend that you use a registration service from their Accredited Registrar List, which can be found at http://www.internic.net/regist.html.

Custom HTTP Headers

Your site may function without the use of a Custom HTTP Header, but this can cause problems because what is sent back to client in the HTTP header about your site's location is the IP address, not the domain name. This can be a security concern (see http://www.securityspace.com/s_survey/data/man.200408/firewalled_cloc.html) and may not be accurate if your server is behind a firewall and uses an internal IP address. It's preferable to send back the domain name you intend to use for your site instead. At a minimum, we need to define what is returned for the location of the site.

While you're in the IIS configuration screen and viewing the Properties for your web site, click on the HTTP Headers tab, and the screen in figure 27 will be shown. Click the Add button.

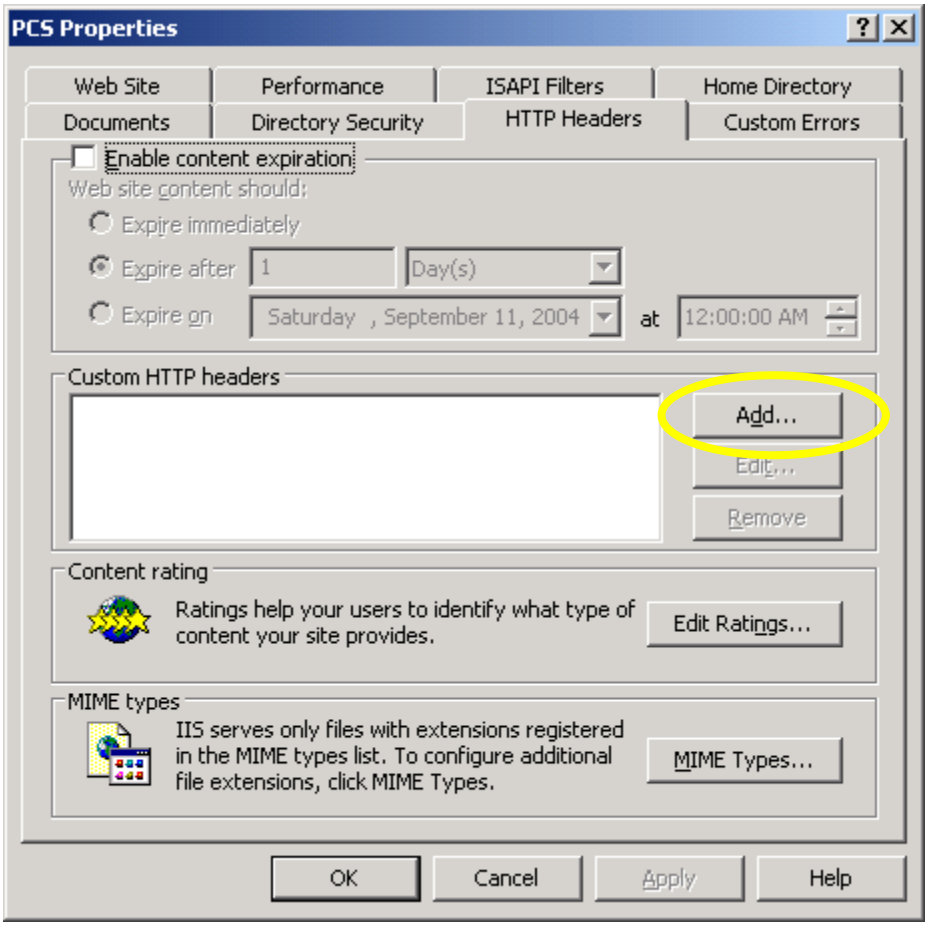


Figure 27 – Adding HTTP header info

Fill in the information as shown in figure 28. Always use “Content-Location” for the Custom header name and fill in the full URL for your site for the Custom header value.

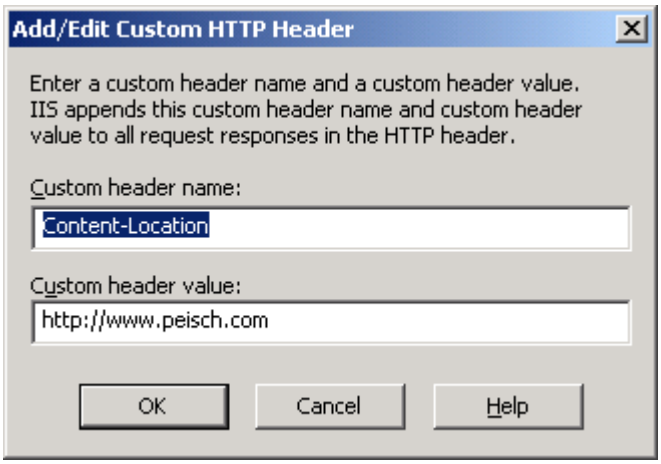


Figure 28 – Filling in the info for the HTTP header

Click OK and you're done.

There are a lot of other things you can define using Custom HTTP Headers. You can use them to instruct your site to expire after a certain period, thereby forcing the browser to refresh the page rather than displaying what's in its cache. You can also add content rating and P3P privacy information. See http://www.microsoft.com/windows2000/techinfo/reskit/en-us/default.asp?url=/windows2000/techinfo/reskit/en-us/iisbook/c08_setting_up_custom_http_headers.asp and http://webmaster.info.aol.com/p3psetup.html for information on how to do this.

Note that the HTTP header is not something you can view by right-clicking on a site in your browser and choosing "View source". Viewing the HTTP header requires low-level viewing of the HTTP stream. The Resources section near the end of this paper offers some choices for ways of viewing HTTP headers. Figure 29 shows an example of what you might find in an HTTP header.

```
HTTP/1.1 200 OK
Server: Microsoft-IIS/5.0
X-Powered-By: ASP.NET
Content-Location: http://www.peisch.com/index.html
Date: Fri, 03 Sep 2004 22:03:59 GMT
Content-Type: text/html
Accept-Ranges: bytes
Last-Modified: Wed, 21 Jul 2004 01:28:42 GMT
ETag: "d28af7ec26ec41:b80"
Content-Length: 7459
```

Figure 29 – Contents of an HTTP header

Using Host Headers for Multiple Sites

It is possible to use different domain names with a single IP address, and have them go to completely different sites on your server. Defining Host Headers for each site on the IP address will allow us to do this. First, I'd like to demonstrate the problem you run into if you try to run multiple sites without host headers. Add a new site using the same technique as you did when you added the first site. You now see the second site in the list, but it says "(Stopped)" after it. (Figure 30)

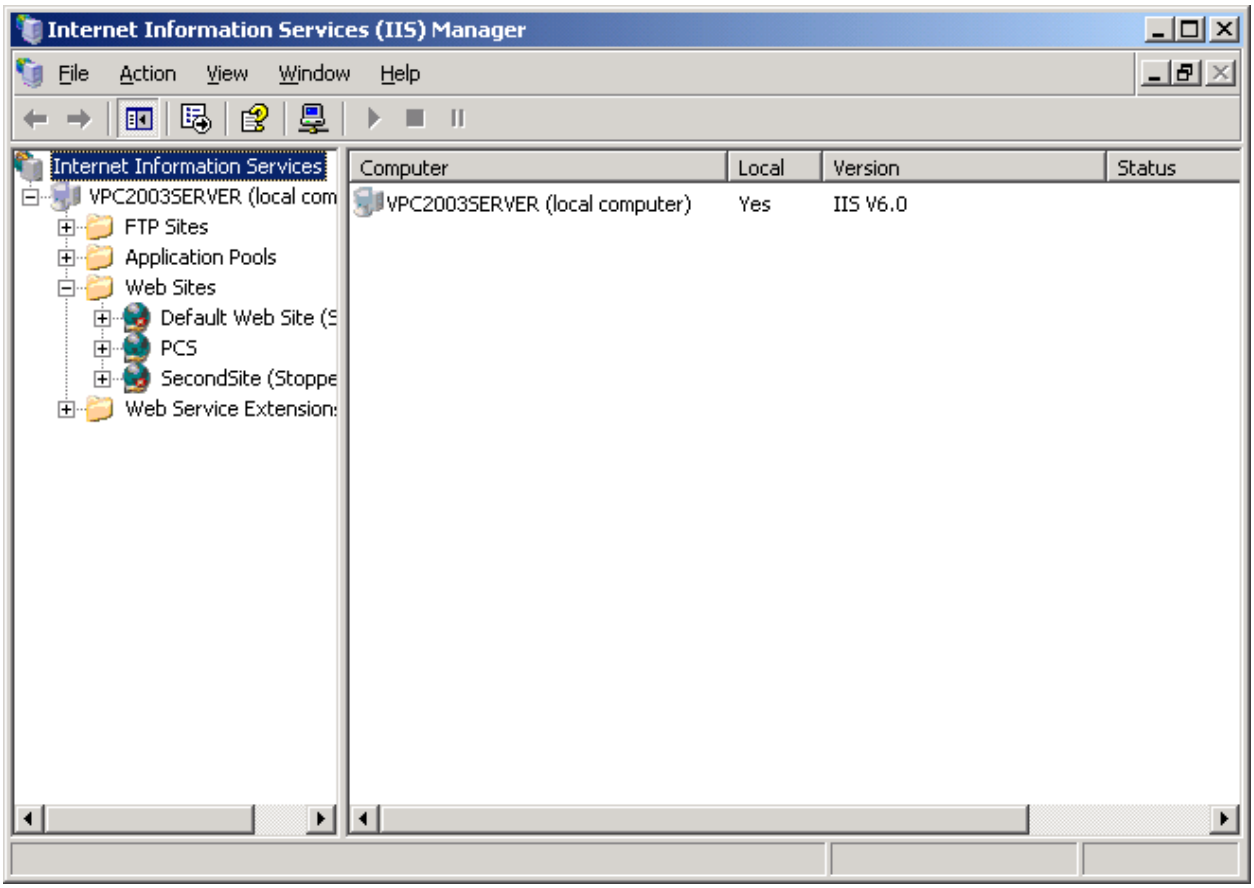


Figure 30 – The list shows the newly created web site

Normally, you can start or stop a web site by right-clicking on it and selecting "Start" or "Stop" from the popup menu. If you try it in this case, though, you get the following message from IIS:

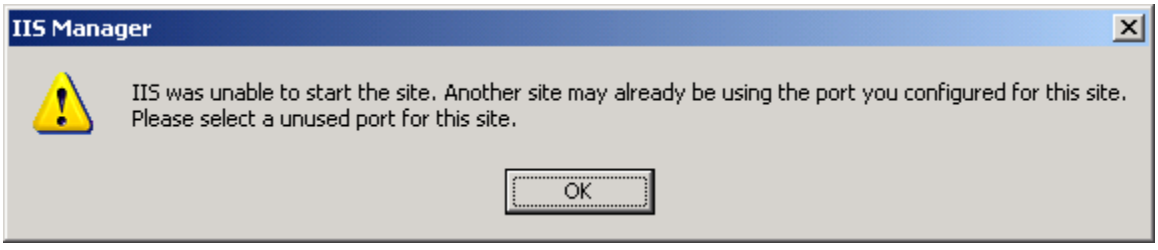


Figure 31 – Error message when trying to "Start" a second site

All we have to do to fix this situation is to add a Host Header for each of the sites. For the first site, right-click, select Properties, and click on the Web Site tab. Click on the Advanced button to the right of the IP address field. (Figure 32)

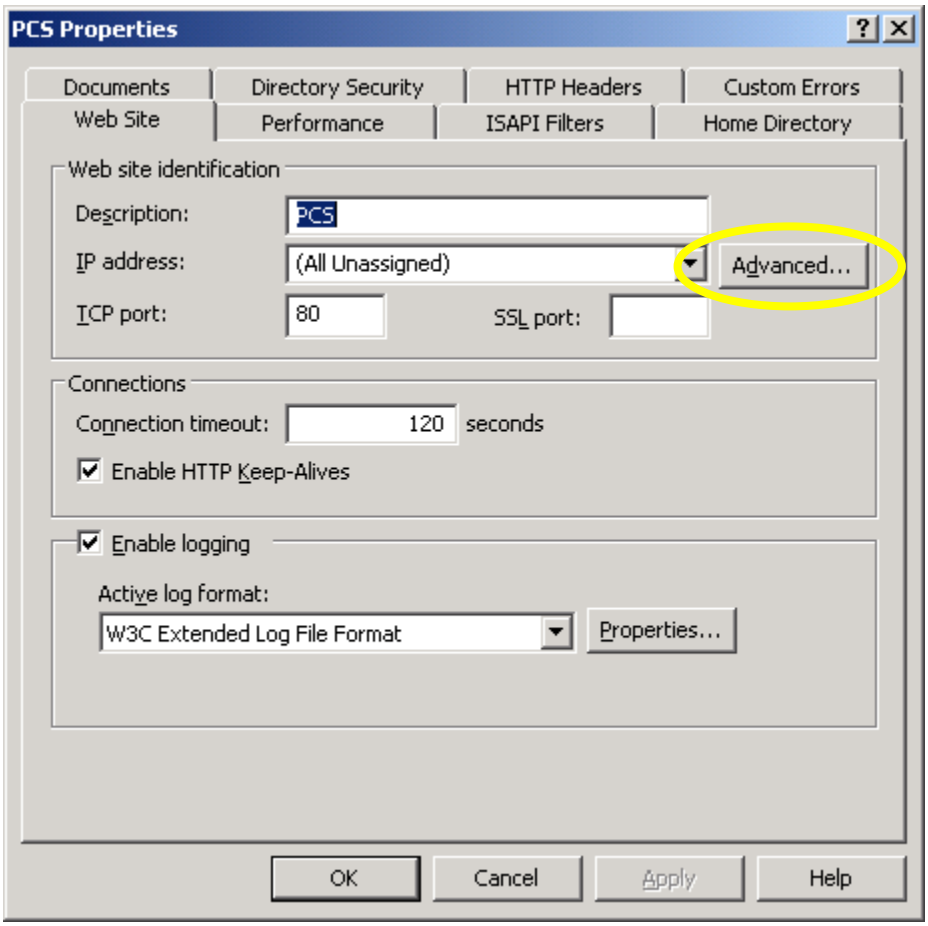


Figure 32 – Advanced button for setting up host headers

You'll see the screen shown in figure 33.

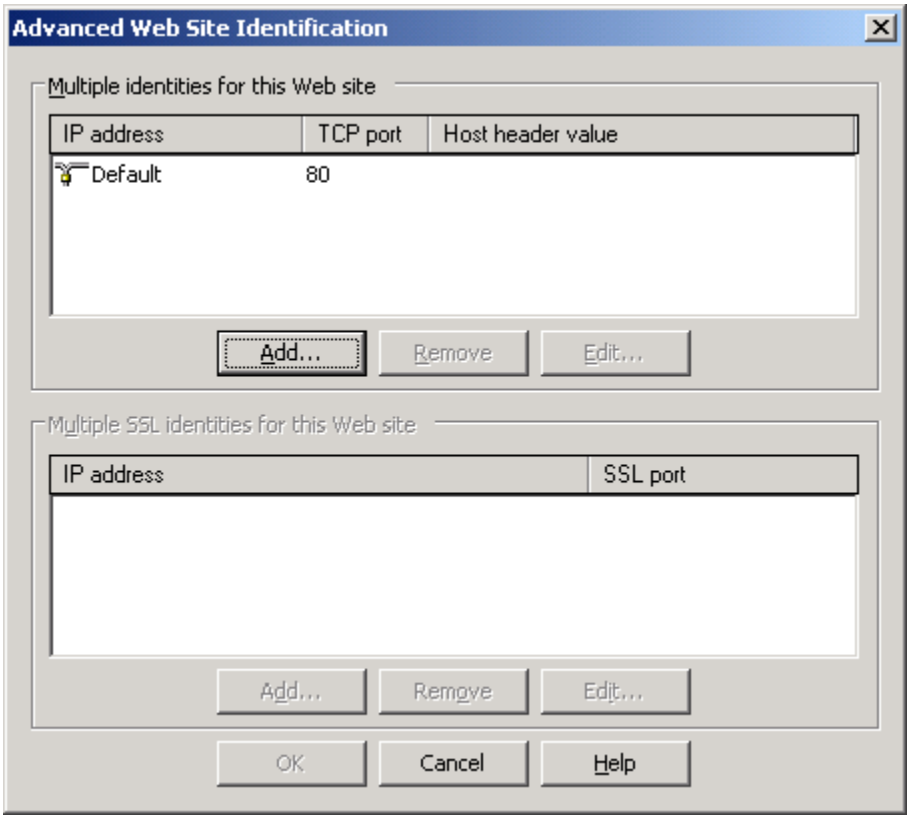


Figure 33 – Defining host headers

Highlight the entry in the top window, click Edit, and enter a domain name for the host header value. (Figure 34)

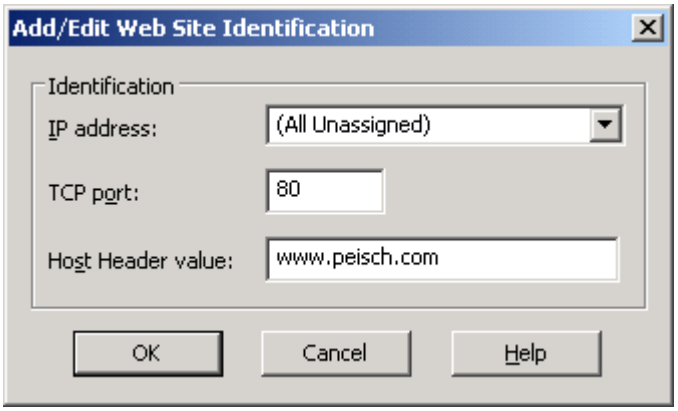


Figure 34 – Fill in the host header name

Click OK and continue to close the windows until the treeview is displayed again. Then enter the host header name for the second site. Once you do, you can right-click on the second site, select "Start", and IIS won't complain. Provided you've set whatever else needs to be set for this second site, both your sites are now up and running and ready to receive requests!

FTP

If you want to be able to upload or download larger files to or from your server, you want to setup FTP (File Transfer Protocol) too. I've generally found that transferring smaller files over HTTP is faster than transferring them over FTP. But larger files can cause HTTP to timeout and cancel your transfer. For these situations, it's best to use FTP instead. We installed FTP at the same time we installed IIS, so that part is done. Now we must configure it.

As with IIS, we must first decide the directory structure where we want files for uploading or downloading. Once again, I don't use the default of \Inetpub\ftproot. Instead, I'll create my own directories for FTP files. For this example, I call my main FTP directory MyFTPSites, as shown in figure 35. If you want to have both a public FTP area and a private one, I recommend creating separate directories for each area.

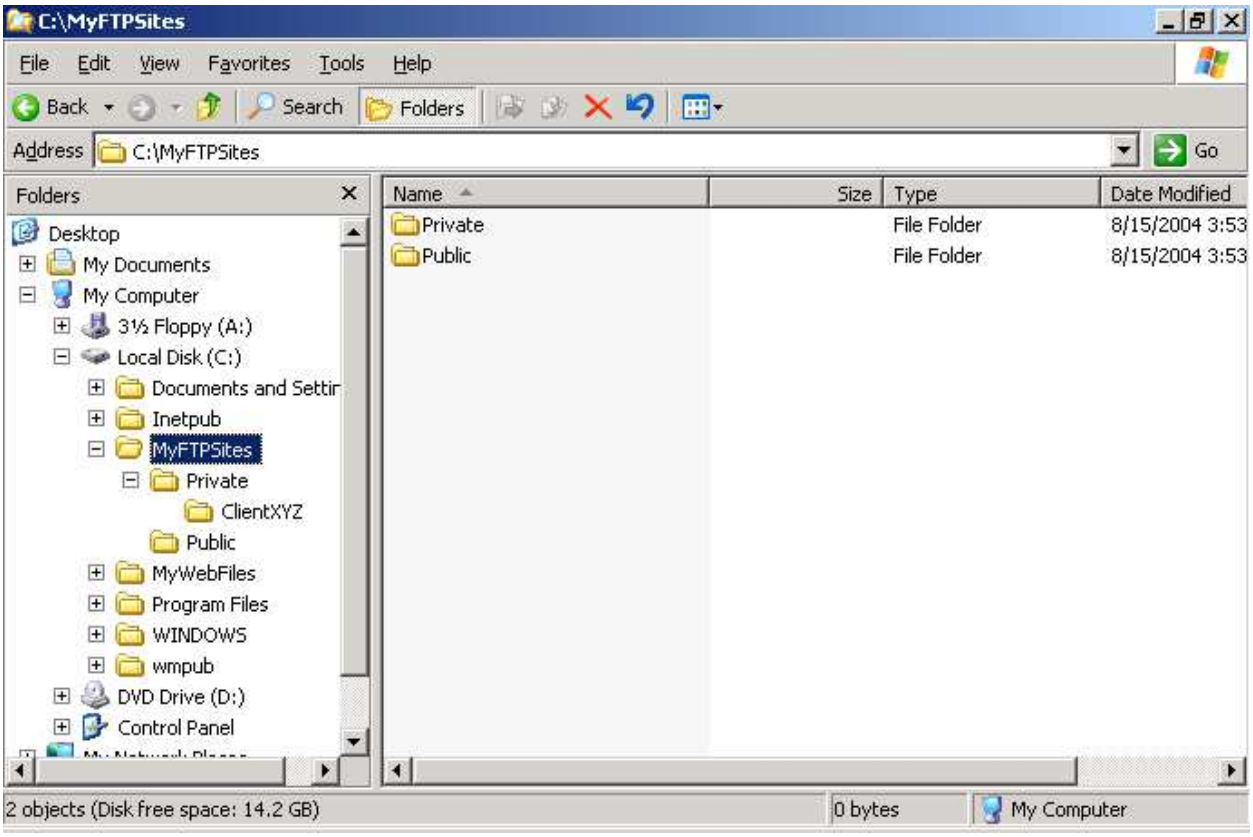


Figure 35 – Setting up the directory structure for FTP

The Private directory has subdirectories for each of my clients who need to send me files. I create a special user for each client, and allow that user read/write/execute privileges. In Windows Explorer you need to give read access to each user who needs access to any of the FTP areas, including the IUSR_ account. For a specific directory and a specific client, I grant Modify access for that directory and user. This allows the client to upload, download or delete files from their FTP area.

You can define users and the group they belong to through Administrative Tools | Computer Management. From the treeview, expand the System Tools node, then the Local Users and Groups node, and right-click on "Users" to get a menu that allows you to add a new user. (See figure 36)

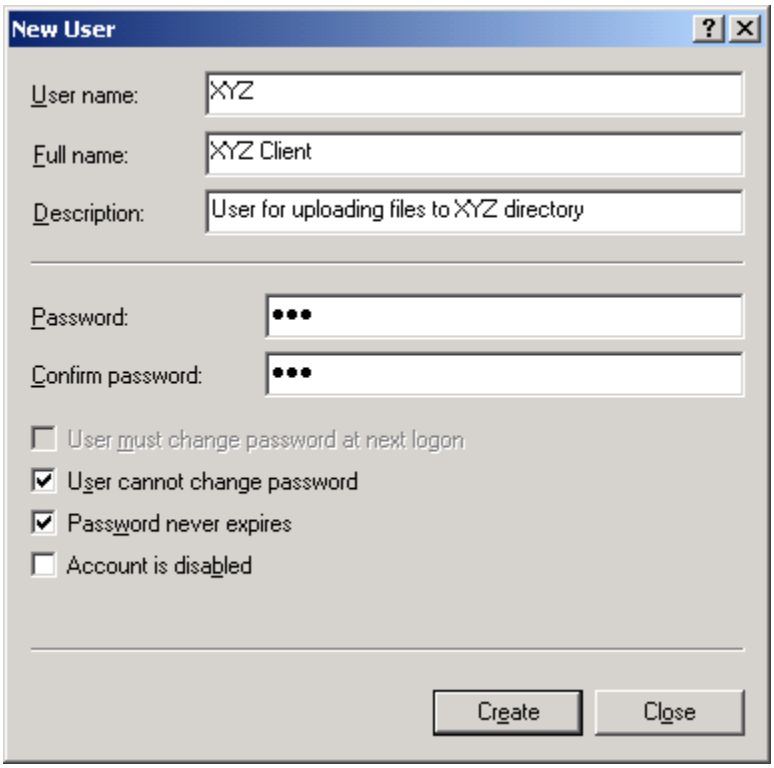


Figure 36 – Setting up a new user

In Windows Explorer, you need to grant the user access to the directory. (See figure 37)

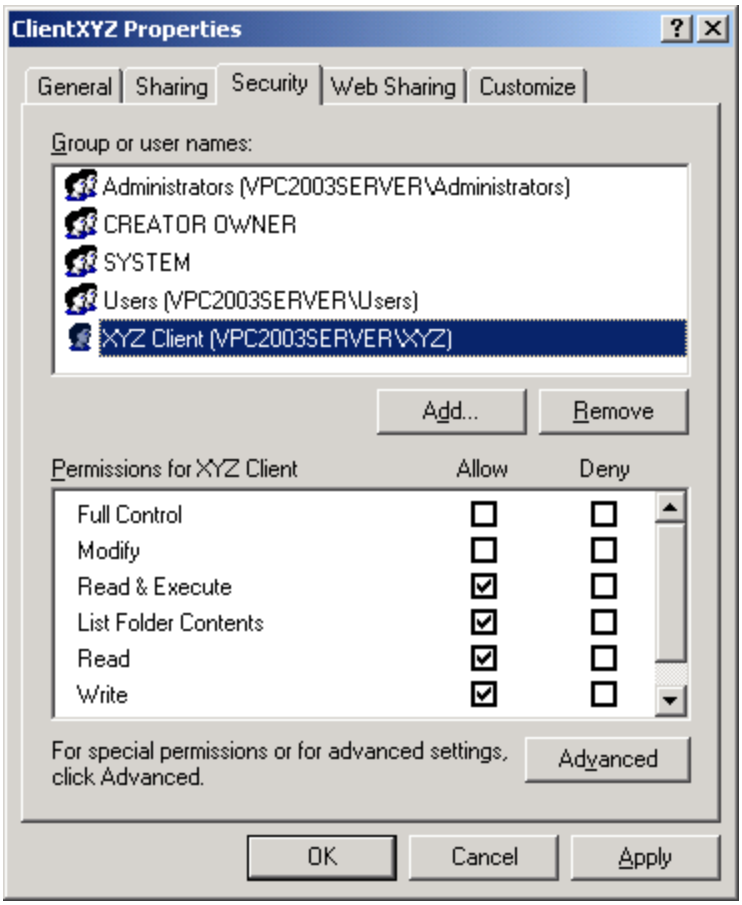


Figure 37 – Setting Windows permissions for client access to FTP

For the Public directory, allow the IUSR account read-only access. If you allow write access to a public area, you're just begging for some hacker to use your disk for storage of huge files. The kinds of files I put in the Public folder are any files I want to make available for general download from my web site.

Configuration of FTP is done through the same interface we used to configure IIS.

Use the same technique to create a new FTP site as we did to create a new web site. A similar wizard will come up. One important step to know about is setting the user isolation option, shown in figure 38. This option is new to Windows Server 2003, and it's something you can't change later without deleting the FTP site and recreating it.

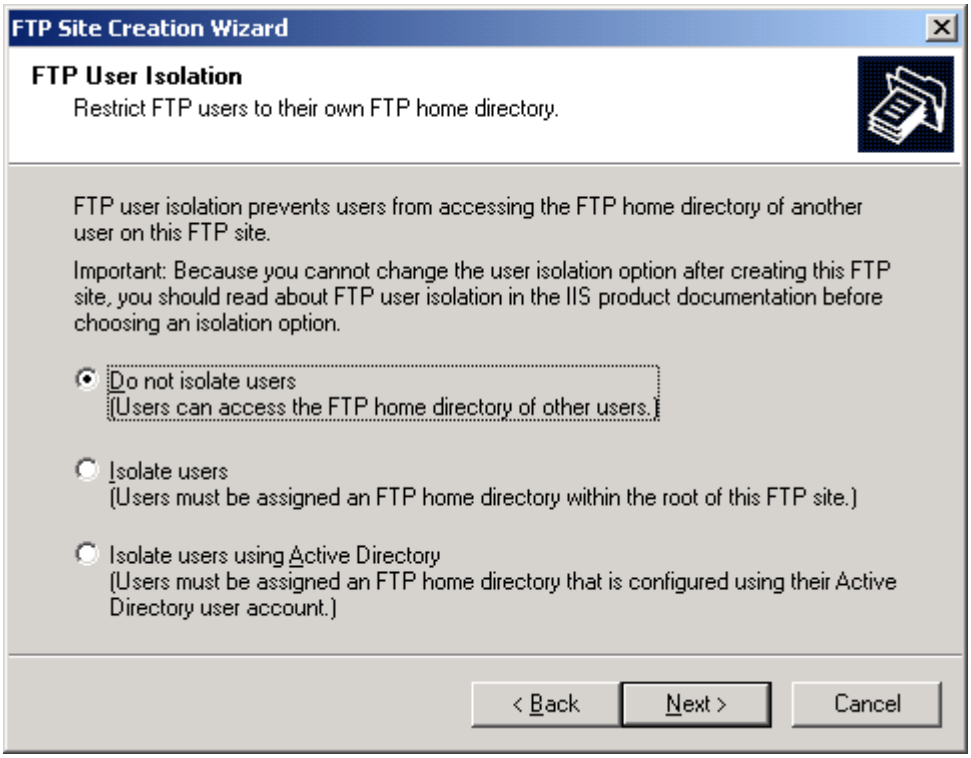


Figure 38 – Determining user isolation for a new web site

The default is option 1. If you only have private sites where the user must always log on, option 2 is probably the better choice. (I haven't figured out how to get this option to work, even when the FTP site only allows users who login.) With option 2, theoretically, one client won't be able to see the other client's FTP directories on your server. (They wouldn't be able to see anything in those other directories, due to the restrictions you've setup in Windows Explorer, but why even let them know they're there?) If you want a single web site to serve both your public area and private areas, then option 2 won't work because it will ask for a login for any access.

The next screen is where you define the location of the root of the FTP site. See figure 39.

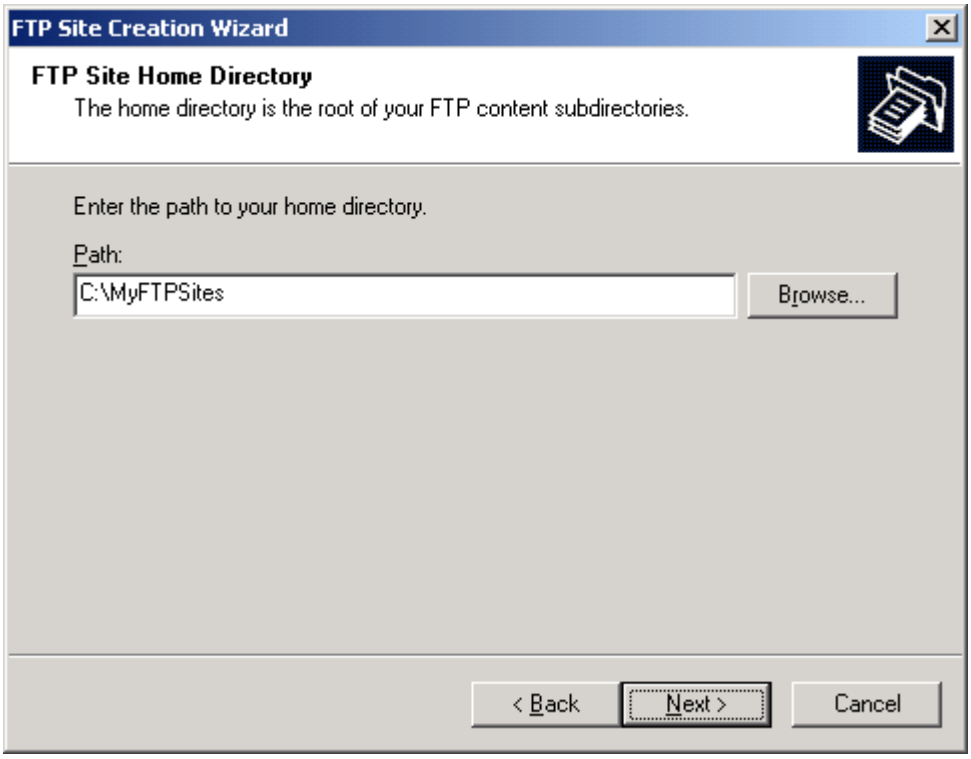


Figure 39 – Specifying the location of your FTP site.

Although we specified the kind of access (read or write) in Windows Explorer, we need to do it in IIS too. (Figure 40) Since this is access for the entire FTP site, specify Read only access. You won't need to specify the access for individual directories because Windows will take care of that, based on the user.

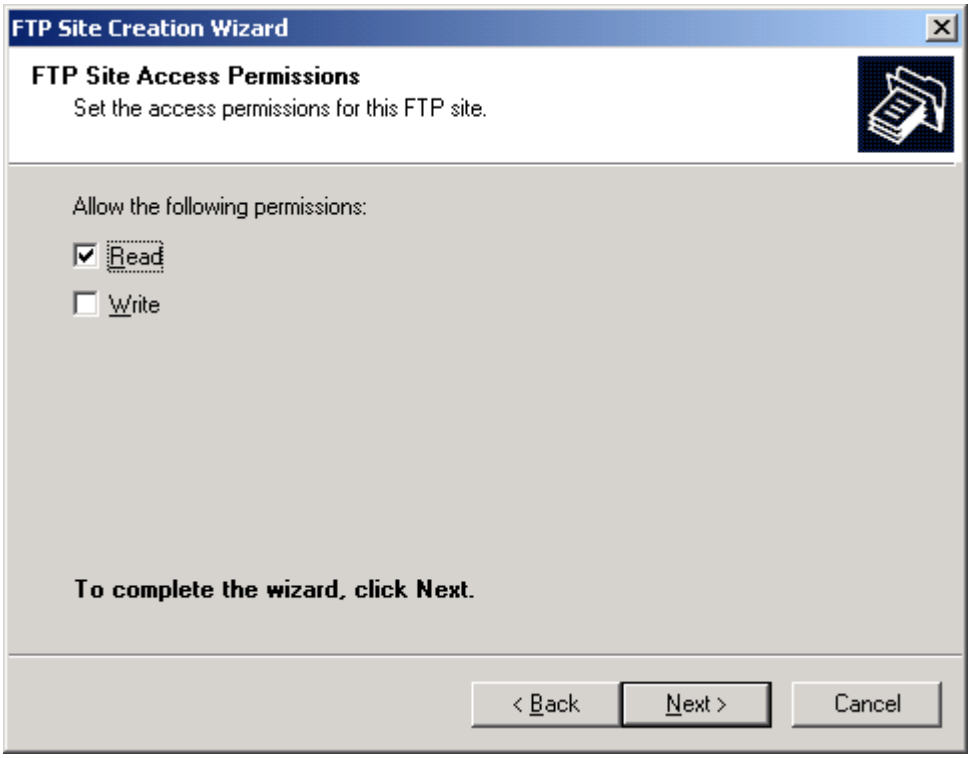


Figure 40 – FTP Access permissions

Next, open the IIS configuration screen. You now see the newly create FTP site listed. (See figure 41)

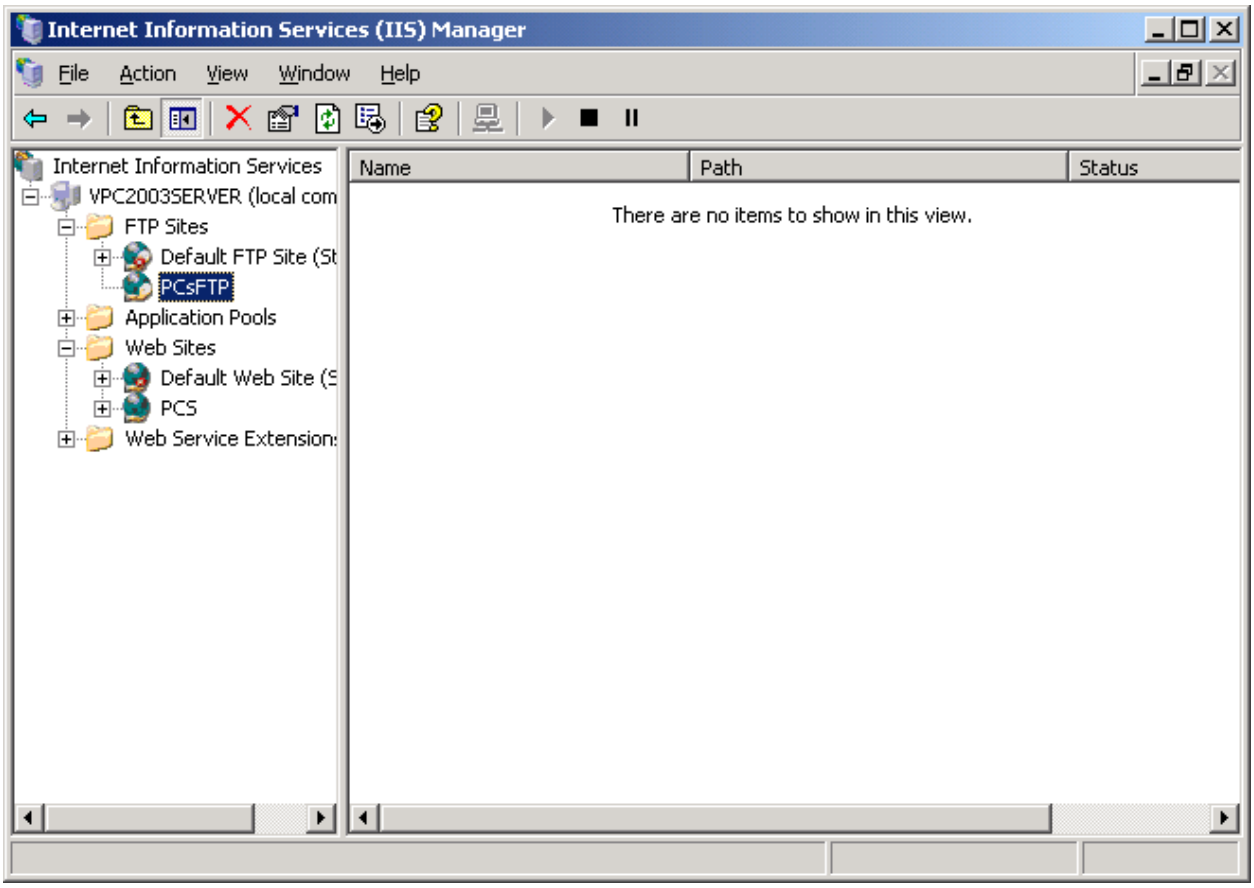


Figure 41 – Newly added FTP site

Should you wish to do something like remove anonymous access from your FTP site, you can do so at any time by right-clicking on the FTP site, and selecting Properties from the popup menu. From there, you select the Security Accounts tab, as shown in figure 42.

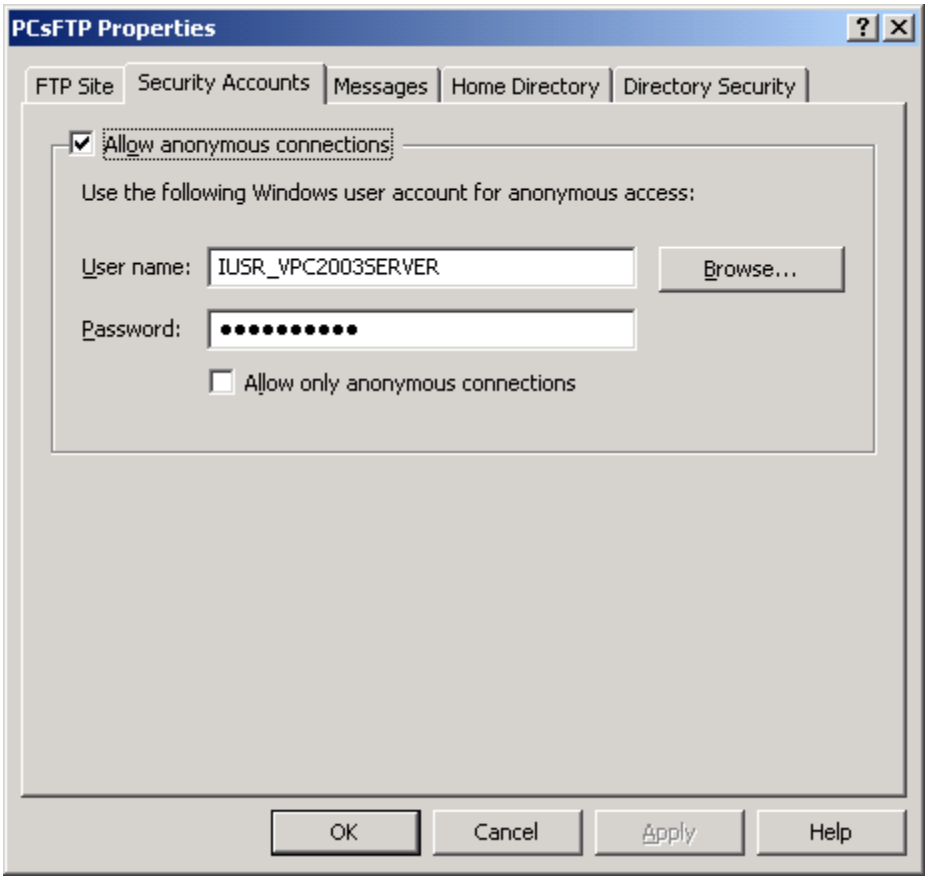


Figure 42 – Security Accounts tab in IIS

From here, you can uncheck the box that allows anonymous connections to force all users to log in to your FTP site. If you do, you'll get a somewhat ominous sounding warning from Windows. (See figure 43)

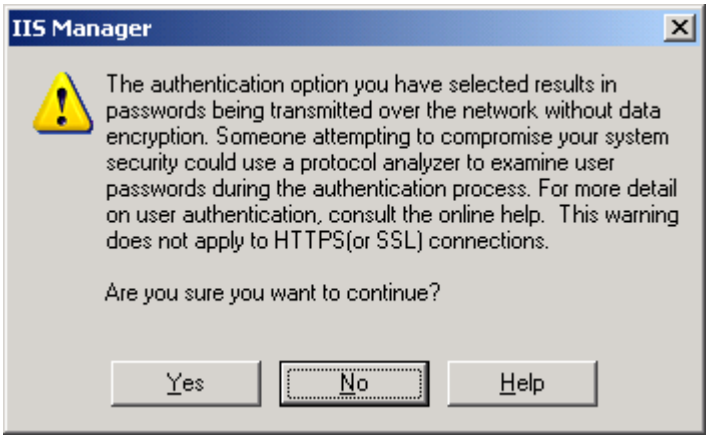


Figure 43 – Warning when removing anonymous FTP access

Accessing the FTP Sites

There are many third party FTP tools that may be preferable for accessing FTP sites. Your browser works too. For this paper, the only browser I'll focus on is Internet Explorer.

To get to an FTP site, just use FTP in front of the URL instead of HTTP. (Figure 44)

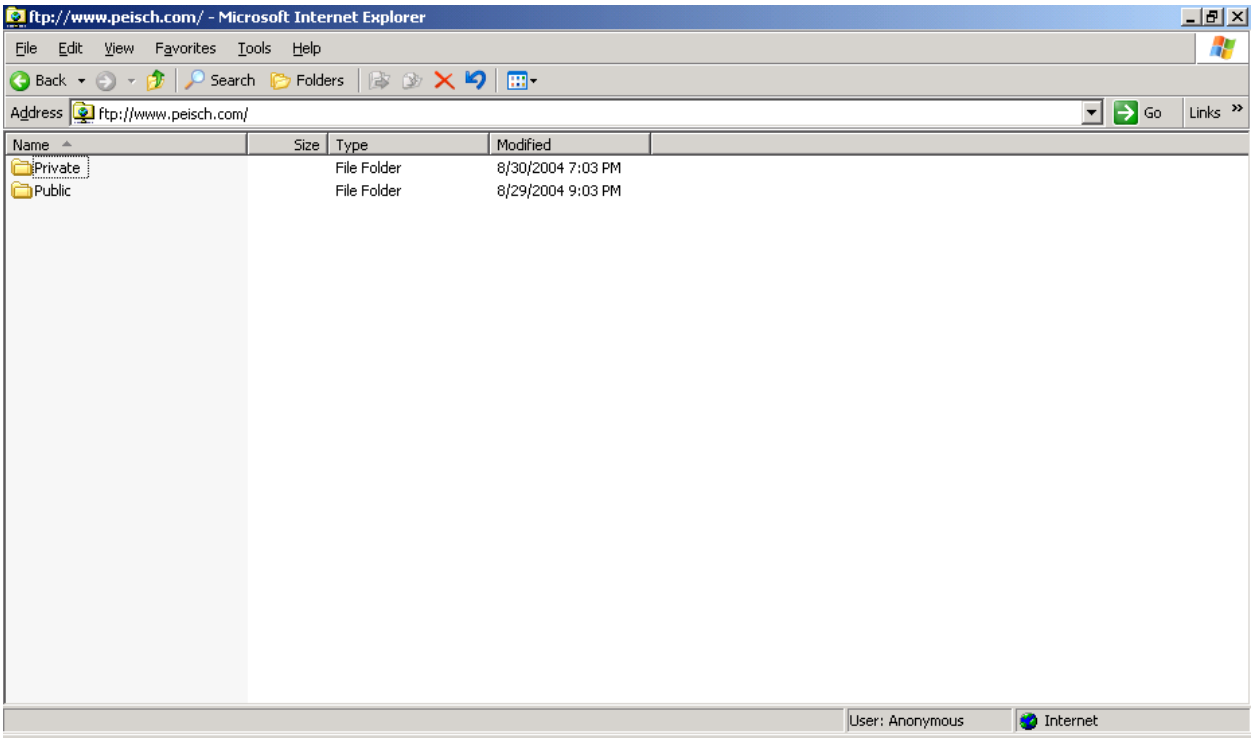


Figure 44 – Accessing a public FTP area

The user can click on the Public subdirectory and see the files there, but if they try to click on the Private directory, they'll get an error. (Figure 45)

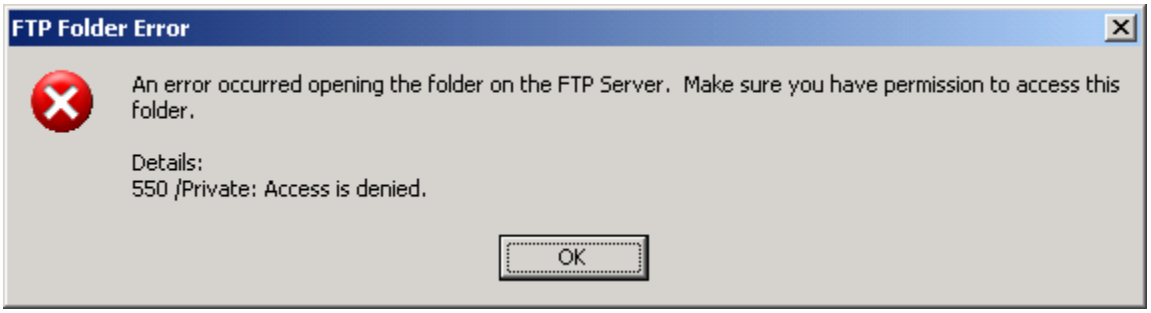


Figure 45 – Anonymous user trying to access a private area

For your users to login to access the private areas, Internet Explorer offers a "Login as" option on the File menu. This brings up a screen where they can enter their username and password. (Figure 46)

Figure 46 – Login screen for FTP

Once the user logs in, they can see the subdirectories under the Private area, but may only view the files in their own directory.

I'm not sure if Netscape or FireFox offer this login feature. I couldn't find anything about FTP at all in the help file for either browser. If you're at all serious about using FTP from a client, I recommend using a dedicated FTP client like WSFTP (http://www.ipswitch.com/), 3-DFTP (http://3dftp.com/) or CuteFTP (http://www.globalscape.com/store/cuteftpdeal.asp?CMP=AFC-CJ30FTP&AID=10299028&PID=1400314).

Running Multiple FTP Sites Using a Single IP Address

We have the same limitation running multiple FTP sites with a single IP address as we did running multiple web sites. I showed you how to use Host Headers to run multiple web sites under a single IP address, but FTP sites don't offer Host Headers.

We can use different ports for different sites to achieve this for FTP sites. We could also do this for web sites, but it's much less likely that you want to require your users to change ports to get to your web site than it is to require this for an FTP site.

By default, FTP is initiated over port 21, and frankly, I don't know how to change this if you're just using a browser to access an FTP site. But it is common to use a dedicated FTP client instead of a browser for accessing FTP sites, and these can be easily configured to initiate FTP over a port other than 21.

So, what we can do is create one FTP site which is our public site, and have it use port 21 (the default). For this site we will allow read-only access, and will not require a login. (See figures 47 and 48)

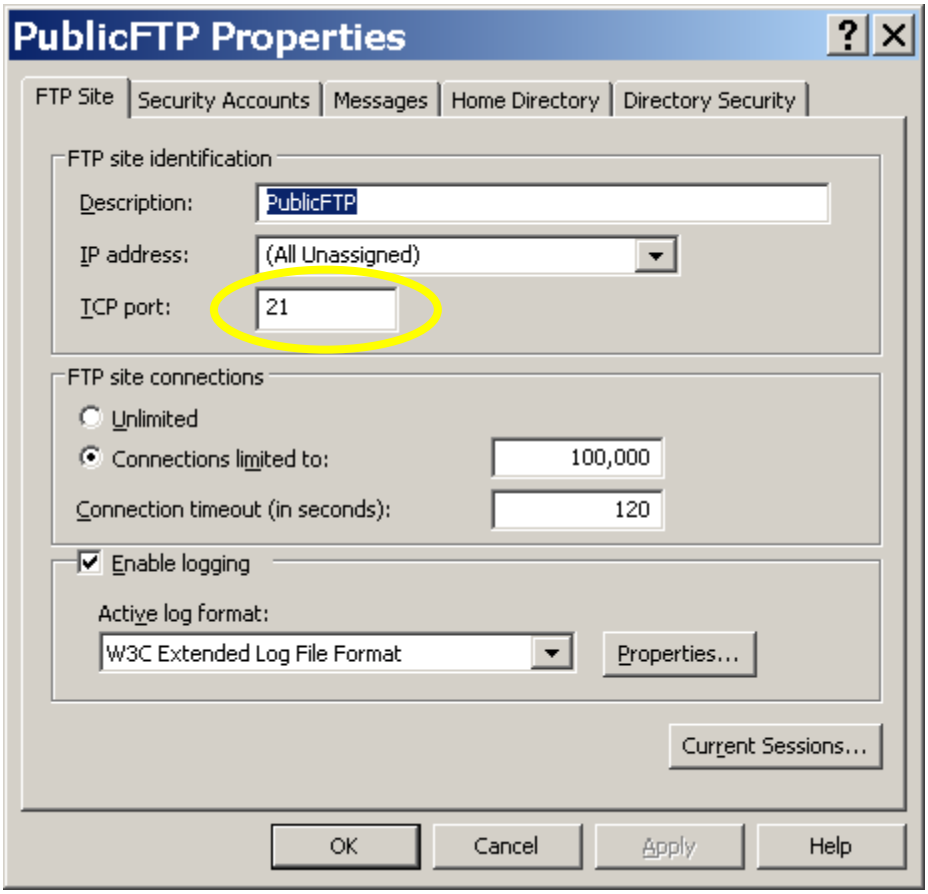


Figure 47 – Main configuration screen for public FTP site

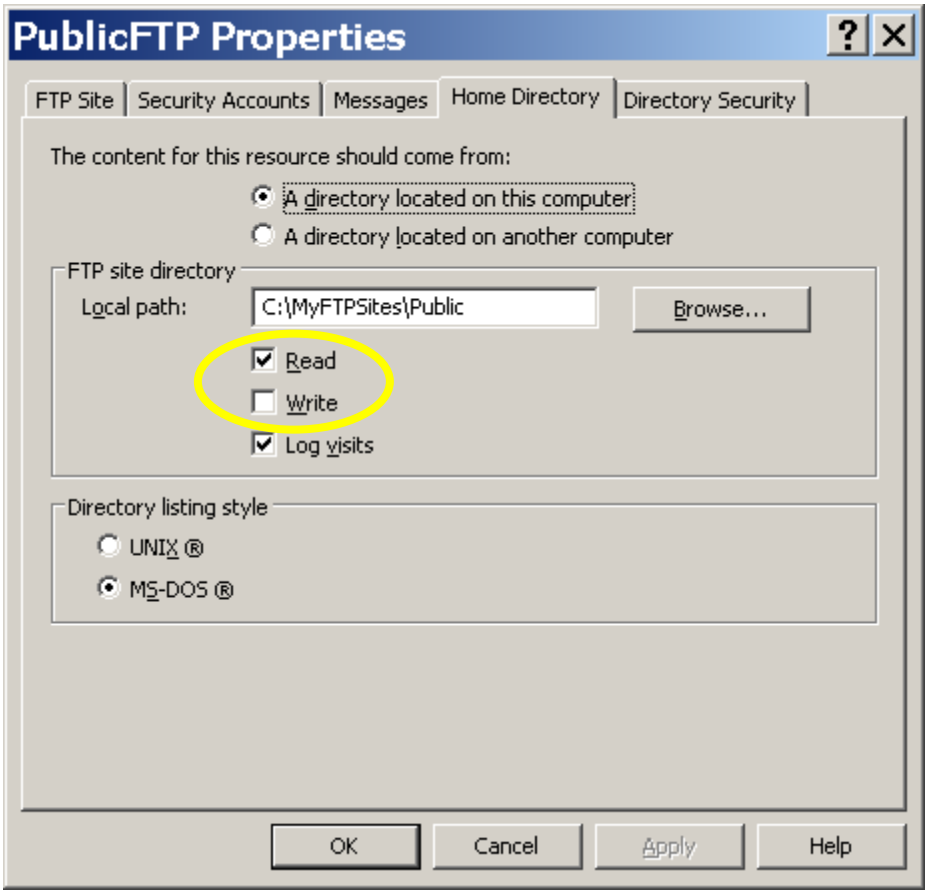


Figure 48 – Read-only access allowed for public FTP site

For the private sites, I'm choosing port 4083 for FTP. This means I have to tell my clients to use this port in their configuration. In figure 49, I show how this is done in WS_FTP by going to the "Advanced" page in the properties for the FTP connection.

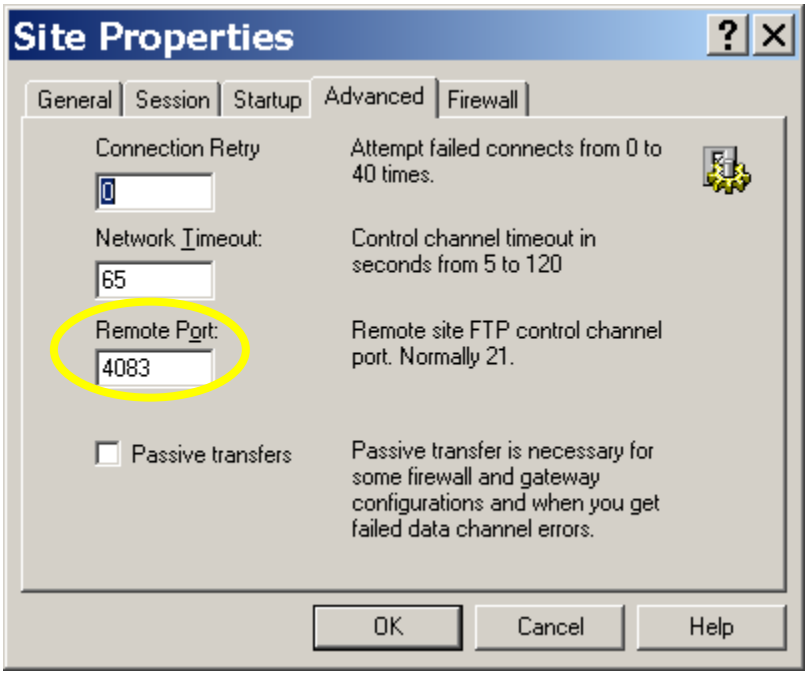


Figure 49 – Configuring the port in WS_FTP

Figure 50 shows how I configure my FTP site to use port 4083 instead of port 21.

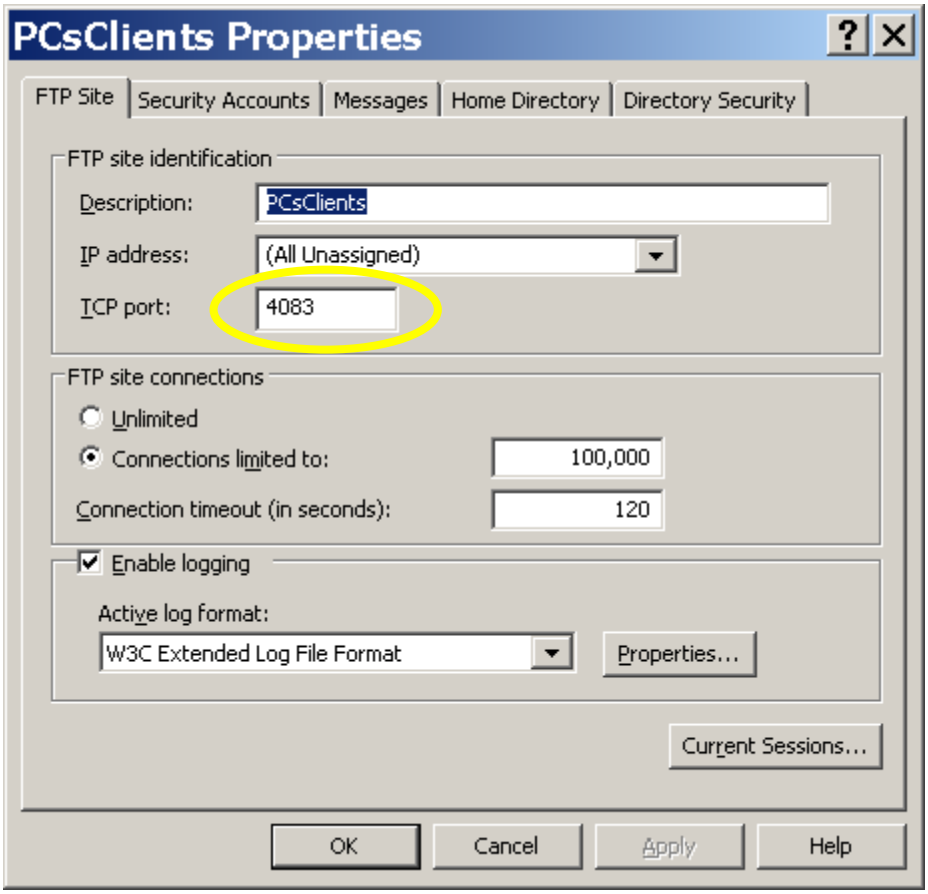


Figure 50 – Using port 4083 for the FTP site

Figure 51 shows how I've unchecked the box that allows anonymous users to get to the site.

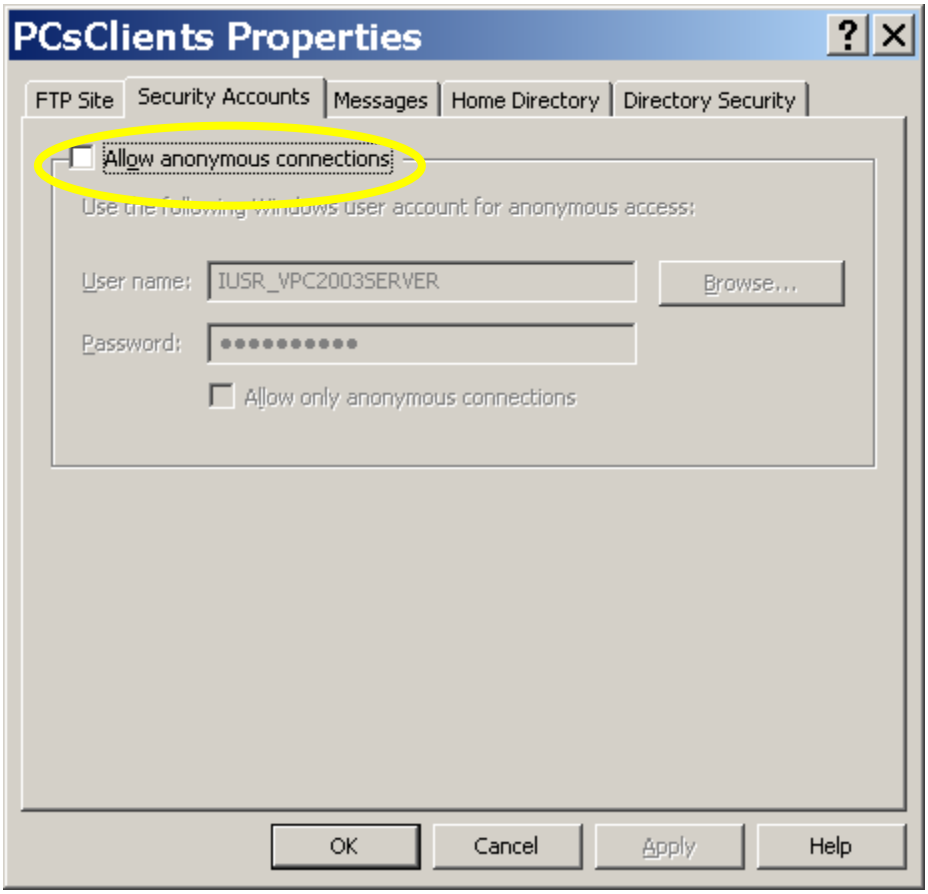


Figure 51 – Removing access for anonymous users

Figure 52 shows how I allow users write access to the FTP site.

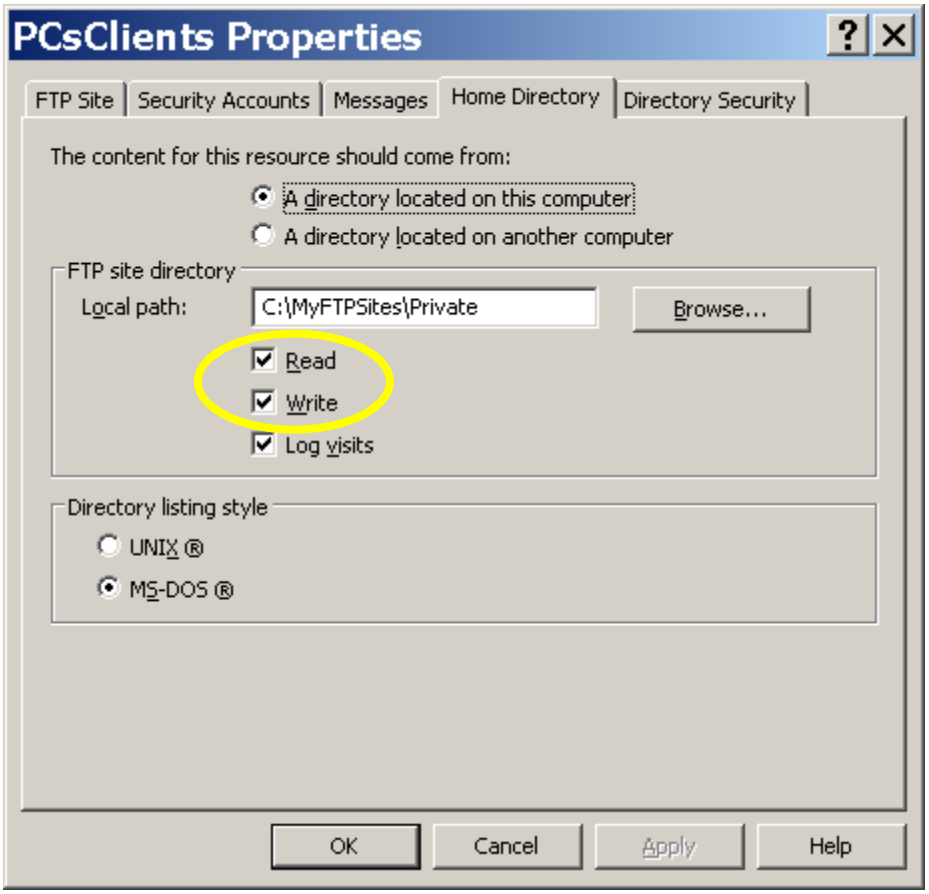


Figure 52 – Allowing user Write access to the FTP site

One advantage of using a port other than 21 for your FTP access is that a hacker trying to get FTP access to areas other than the public site won't know which port you're using, of course if they are scanning for open ports on your server, they may discover it anyway. See the Resources section for information about port scanning.

Using this technique, any client logging in can see the directories for other clients, but can't access those other directories. If you prefer to hide those other directories completely, then you may choose to setup a different site for each client, each using a different port.

Certificates

If any part of your web site needs to be secure, then you'll need to purchase a certificate, and access your site through secure HTTP (HTTPS). By simply having HTTPS in front of your URL instead of HTTP for these pages, your browser will attempt to access the site using a certificate. This attempt will fail unless you've installed a valid certificate on your server. Purchasing and installing a certificate is fairly involved, and I could write another paper on just this topic. If you want to look into running a secure web site, I recommend you first read the section on Certificates in IIS help under Server Administration Guide→Security→Certificates. When it comes to purchasing a certificate, I recommend getting your certificate from either Verisign

http://www.verisign.com/products-services/security-services/ssl/index.html?sl=b46310164670057000 or Thawte http://www.thawte.com/ Either of those vendors will have instructions for how to obtain and install a certificate.

Adding a Router

Although I'm not going to address setting up multiple servers for the same web site in this paper, setting up a router so that multiple computers can access the outside world is a more common need. (And one I have experience with.) If you also want a server in this arrangement, you won't be able to use the configuration settings recommended by the router manufacturer, as those will prevent any traffic from outside from seeing your server.

There are two philosophies for setting up a server and router. One is to put the server between the outside world and the router, in what's called the "De-Militarized Zone" or "DMZ". The rest of the computers would be behind the router, and therefore hidden from the outside world.

The second technique, and the one I follow, is to put the server as well as the rest of the computers behind the router, so the setup is ISP→Modem→Router→Server and other computers.

You'll need to make sure you select a router that supports Port Forwarding. Linksys is a good choice of manufacturer. I've found that their basic Cable/DSL 4-port router works fine in both the wired and wireless models.

The first thing you need to do is change your server's IP address to an internal address because the router will now take on your fixed IP address assigned to you by your ISP. There are a few ranges of IP addresses that are reserved for internal use only. The most commonly used range is 192.168.0.0 through 192.168.255.255. Within your network, you'll probably be using 192.168.1.1 to get to the router, so choose a number other than that within the range to be used on your server.

Your router will come with instructions for the basic setup. Start with this setup initially so that you get the router configured with the settings necessary. Make sure you change the administrator password so that it's not so easy for just anyone to get into your router's configuration. Also follow the steps in this paper on Configuring Your Network Card, changing the IP address to the internal number you decided to use. The default gateway in your network card's configuration will be the internal number of the router, probably 192.168.1.1. Once you finish this setup, test your browser on the server to make sure you can get to an outside web site. Add any additional computers to the router, following the manufacturer's instructions. You can use Dynamic Hardware Control Protocol (DHCP) to assign the IP address to the non-server computers. Since DHCP is handled by the router, we never had to install it on the server. (The only time you would need to have DHCP installed on your server is for an internal network where the server is a file server, and it assigns the IP address dynamically to all the workstations that connect to it.)

Once you've got the router working according to the manufacturer's instructions, you're ready to modify the configuration so that the outside world can see your server.

Your router should have a screen for setting up port forwarding, similar to the screen shown in figure 53.

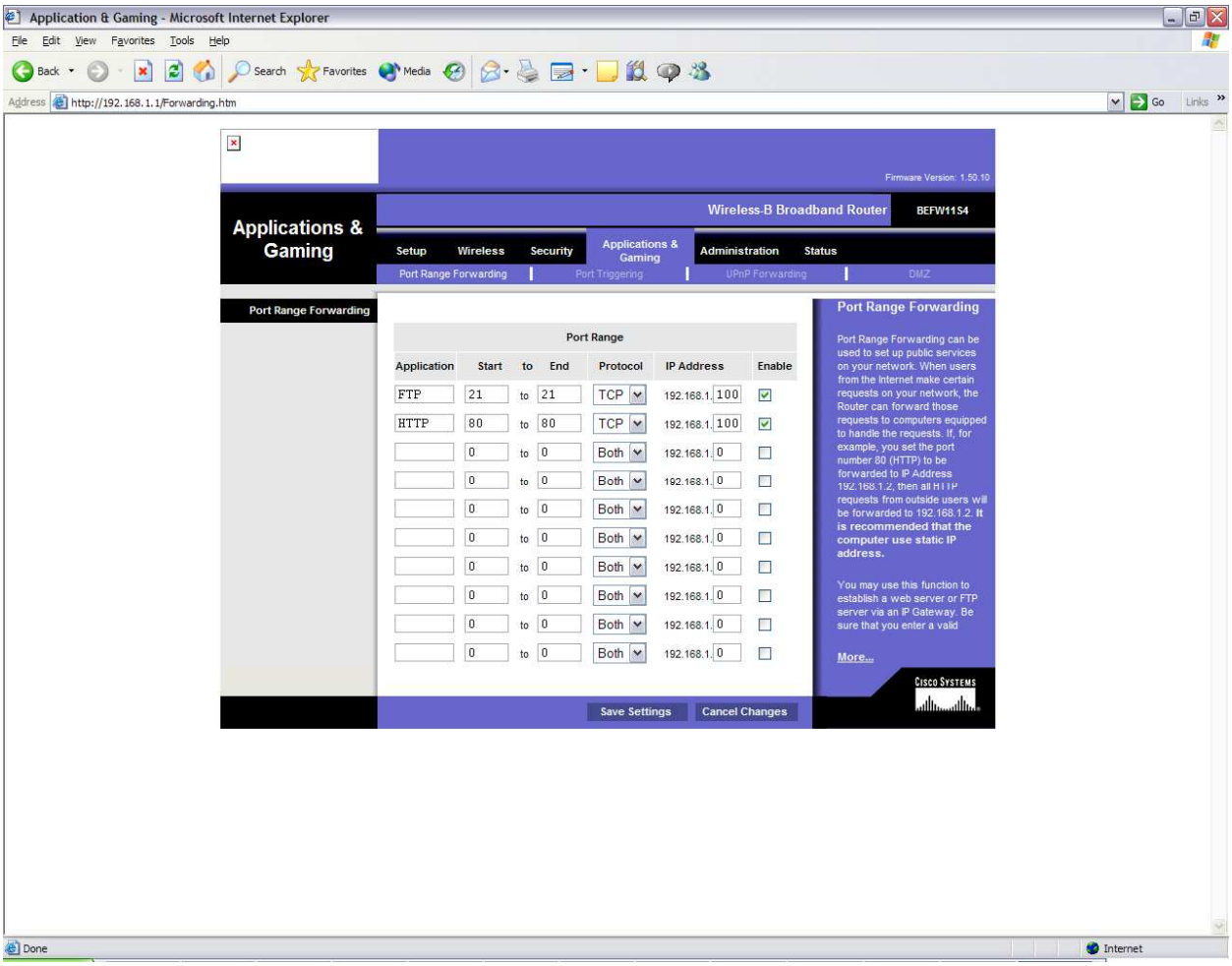


Figure 53 – Configuring Port Forwarding

Notice I've configured port forwarding for ports 21 and 80, which are normally used by FTP and HTTP. If you're using SSL on your server too, you'll want to add HTTPS to the list, which uses port 443. If you're using an FTP site that uses a port other 21, as I suggested in the FTP section, then you'd have to add that port to the list as well. If you use a range of ports for private FTP sites, you can enter that range with a single entry. In the IP Address column, I've specified 192.168.1.100, which is the internal IP address I've assigned to the server. Any requests coming in over the ports specified in this list will be forwarded to the computer at the internal IP address listed, which enables the outside world to get to your server.

One caution to keep in mind when using port forwarding is that by doing so, you negate any hardware firewall protection on the server that the router would normally offer. For this reason, it's extremely important that you run a software firewall on the server.

Other Software

In addition to the software discussed, there are other things you probably want to install on your server.

Firewalls and Other Protection

I've already mentioned using a firewall on your server. You need to make sure the firewall software you purchase is meant to run on a server. Mcafee is well known for their anti-virus and general protection software (http://www.mcafeesecurity.com/us/products/mcafee/host_ips/category.htm). Symantec is also very popular (http://enterprisesecurity.symantec.com/content/productlink.cfm). Windows now comes with its own firewall. You can also put "Server Firewall" into Google and find a whole list of options. Just make sure you do run firewall and anti-virus programs, and keep them up-to-date.

Remote Access

You'll probably also want to have Remote Control software to allow you to access your server from other locations. PCAnywhere, now owned by Symantec is one of the oldest packages that does this, but it can be a bit expensive. Terminal Server from Microsoft is another option. There are many other options that are inexpensive or free. Some of the more popular choices are:

- PC-Duo http://www.vector-networks.com/pc-duo-remote-control/
- VNC – Free. http://www.realvnc.com/
- GoToMyPC https://www.gotomypc.com/

Monitoring Your Server

Every request to your server is stored in a log. You can customize what goes into these logs and where they're stored through the IIS configuration screen. (Figure 54) Both web sites and FTP sites have logs, but are stored separately, so make sure you check the configuration for both types. Also, check the settings for each site, because the configuration for each may be different.

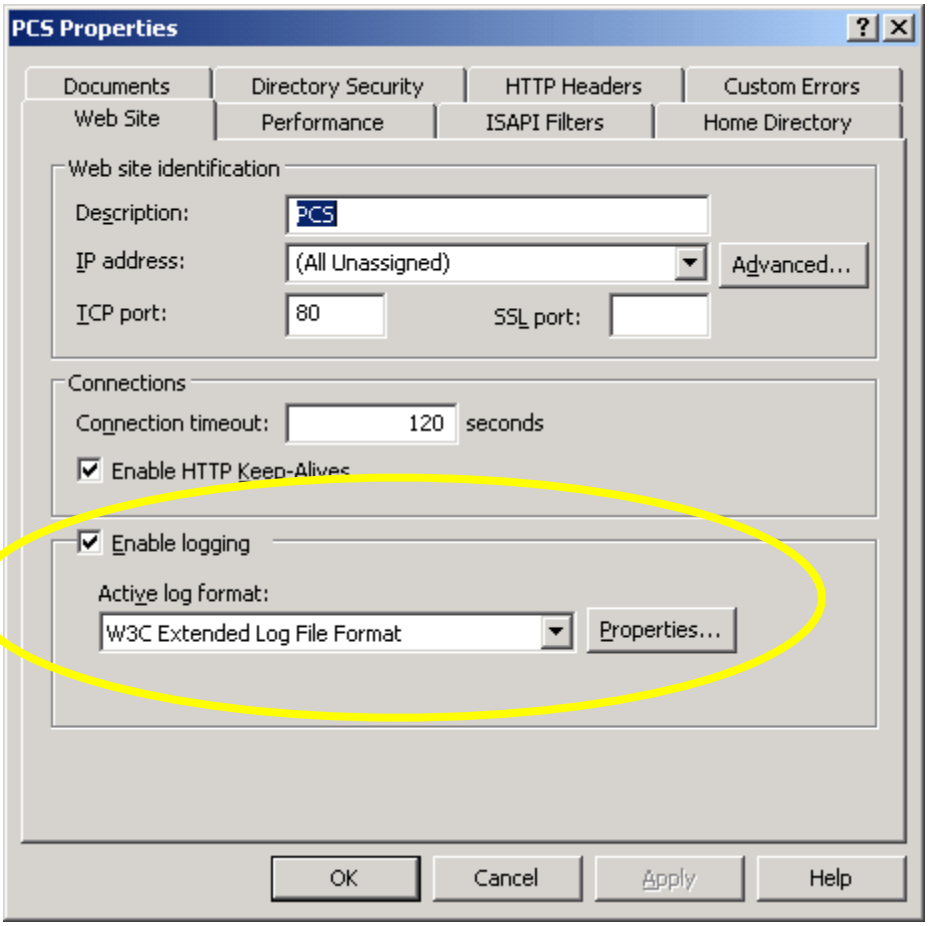


Figure 54 – Configuring logging

The logs are plain ASCII files, but reading them in this format may not be what you're after. There are some tools that do analysis on these files and present the data in more attractive ways. Most of these are free.

Some suggestions:

WebTrends: http://www.netiq.com/webtrends/default.asp

AWStats: http://awstats.sourceforge.net/

The Webalizer: http://www.mrunix.net/webalizer/

NetTracker: http://www.sane.com/ads/google/whoiscoming.html?ntc=4-B&ntk=Log%20analyzer

Resources

Online Definitions for Computer Acronyms and Terms

One of my favorite places for helping me keep straight the alphabet soup that the computer world has become is http://www.webopedia.com. This site is actually good for looking up any computer term, but I find it especially useful for all those acronyms.

Dynamic DNS Services

Although it's best to have a fixed IP address for your server, there are some cases where you can use a DNS Service instead. I've never used a DNS Service and so I cannot attest to how well they work. Here are a few you can check out

http://www.dynip.com

http://www.no-ip.com

http://www.dyndns.org

Info on How DNS Works

http://www.jhsoft.com/help/index.html?df_hostsfile.htm

Windows Server 2003 Info

http://labmice.techtarget.com/windows2003/default.htm

IIS Status Codes

Both your browser and the server's log files will often show you return codes from IIS and you won't know what they mean, unless you look them up here.

http://support.microsoft.com/default.aspx?scid=kb;en-us;318380

HTTP Header Viewers

If you want to know what is in the HTTP header returned by your web site and your site is online, you can use these services to view the HTTP header:

Webmaster Toolkit: http://www.webmaster-toolkit.com/http-header-viewer.shtml

Delorie Software: http://www.delorie.com/web/headers.html

1-Hit.com: http://www.1-hit.com/all-in-one/php/header-check.php

Or you can use a plug-in tool for Internet Explorer:

HTTPWatch: http://www.httpwatch.com/

IEWatch: http://www.iewatch.com/

Port Scanning

Port scanning is a technique used by hackers to find open ports on your server, and a possible way to break in. Although it's necessary to have some ports open on a server to allow access, you don't want to have ports open if you're not using them. Here are some sites with info about this:

Gibson Research Corp: https://www.grc.com/x/ne.dll?bh0bkyd2

Nmap: http://www.insecure.org/nmap/

Atelier Web: http://www.atelierweb.com/pscan/

Appendix A – Log examples

Web Services Log

Below are a couple of examples of entries from one of my web site logs.

```
#Software: Microsoft Internet Information Services 5.0
#Version: 1.0
#Date: 2004-06-25 08:19:16
#Fields: date time c-ip cs-username s-sitename s-computername s-ip s-port cs-method cs-uri-stem cs-uri-query sc-status sc-win32-status cs-version cs-host cs(User-Agent) cs(Cookie) cs(Referer)
2004-06-25 08:19:19 64.160.96.182 - W3SVC3 WIN2KADVSERVER 192.168.1.100 80 GET /scripts/..%5c../winnt/system32/cmd.exe /c+dir 404 3 HTTP/1.0 www - - -
2004-06-25 09:19:38 193.95.47.130 - W3SVC3 WIN2KADVSERVER 192.168.1.100 80 GET /publicfiles/everything.zip - 200 64 HTTP/1.0 63.193.37.117 Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.1) - http://www.peisch.com/downloads.html
```

You can see why it's easier to read these logs with an analyzer. Nevertheless, there are a couple of things I'd like to point out. The first entry is a hacker's attempt to get (and run) a file they shouldn't have access to. The second entry is a legitimate request. Below, I've isolated some important parts the first entry and broken it out into pieces.

```
2004-06-25 08:19:19 64.160.96.182 - W3SVC3 WIN2KADVSERVER
|_________________| |___________|   |____| |_____________|
         |                 |           |          |
GMT date and time    IP address     Type of     OS of
Of log entry         of user        service     Server

192.168.1.100 80 GET /scripts/..%5c../winnt/system32/cmd.exe
|___________| __ |_| |_____________________________________|
      |        |  |                  |
 IP address  Port |            File requested
 of server       Verb
```

```
/c+dir 404 3 HTTP/1.0 www - - -
|____| |_|
   |    |
Param   IIS return code
Passed
To prog
```

The reason I know this entry is a hack attempt is because there is no valid reason anyone would be looking for a file or trying to run the cmd.exe file on my server. Note the return code is 404, which is an error code showing that the file is not found. Obviously, there is a cmd.exe file in my system32 directory or my server wouldn't be running. But I don't let IUSR or the Everyone user have any access to that directory. Although you can't keep these kinds of hack attempts away from your server, you can make sure they aren't successful.

The second entry is an attempt to download a file called Everything.zip, which is a file I've made available for download from my web site. Note that the IIS return code for this request is 200, which means the request was successful.

FTP Log

Below are a couple of entries from my FTP log.

```
#Software: Microsoft Internet Information Services 5.0
#Version: 1.0
#Date: 2004-05-05 17:15:27
#Fields: time c-ip cs-method cs-uri-stem sc-status
17:15:27 192.168.1.1 [9]USER anonymous 331
17:15:27 192.168.1.1 [9]PASS IEUser@ 530
#Software: Microsoft Internet Information Services 5.0
#Version: 1.0
#Date: 2004-05-05 17:20:43
#Fields: time c-ip cs-method cs-uri-stem sc-status
17:29:46 192.168.1.1 [19]USER ClientXYZ 331
17:29:46 192.168.1.1 [19]PASS - 230
17:29:57 192.168.1.1 [19]created PCS.BAT 226
```

Once again, the first entry is a hack attempt. I don't currently allow anonymous access to my FTP site at all. There are two rows for request [9]. The first shows the user who attempted to login (anonymous) and the code that was returned is 331, indicating that the client browser needs to take additional action to fulfill the request. In this case, that would be entering a legitimate password, which was attempted on the second line for request [9]. The password entered was IEUSER@, which is not valid, and a code of 530 was return, which means the user name/password combination was not valid.

The second request ([19]) shows that a user attempted to login with the user name ClientXYZ. The log doesn't reveal the user's password, but does show that the IIS return code was 230, which means the user logged in successfully. The next line shows that a file called PCS.BAT was uploaded, and the return code is 226, which means the file was uploaded successfully and the data connection was closed.