Realtime publishers

Tips and Tricks Guide™ To

Windows Administration

Don Jones and Dan Sullivan

Copyright Statement

Tip, Trick, Technique 1: Setting up a Server Core Domain Controller

Windows Server 2008's Server Core installation is a great option for domain controllers: The operating system (OS) has a smaller footprint and has so far required significantly fewer patches than the full Windows installation, making it possible to have less downtime and maintenance for your critical domain controllers. In this tip, we'll install a Server Core domain controller from scratch.

Starting the Installation

The installation begins, ironically, with the lightweight GUI installer that's familiar to all editions of Win2008 and to Windows Vista. Select one of the Server Core options.

> **Note**
>
> Note that this is a one-time decision: You can't later "upgrade" to the full Windows installation nor can you "downgrade" a full install to Server Core.

That's about the only decision you have during installation. When it's finished, you'll be looking at a logon screen and might be wondering what to do. Select the "Other User," and log in as Administrator. Use a blank password; you'll be immediately prompted to create a new password.

After changing the password, you'll be logged in and staring at your new, trimmed-down desktop. That's right—not much to see! This is Server Core, and it has only a few graphical elements available to it. To get it up and running, you'll need to run a few commands. Many of these will be commands you're familiar with already; others are new and are unique to Server Core.

Basic Configuration

Since we're building a domain controller, you'll probably want to start by assigning a static IP address. Do so using the **Netsh** command, as shown, to get a list of network interfaces. Use the number in the "Idx" column to refer to the interface in later commands.

```
Netsh interface ipv4 show address
```

With your network adapter identified, assign a static IP address, subnet mask, and default gateway using the **Netsh** command. The **Name=** parameter is where your chosen adapter's ID number goes.

```
Netsh interface ipv4 set address name = 2 source=static address=10.0.1.57
mask=255.255.255.0 gateway=10.0.1.1
```

Use the same technique to assign a DNS server. To assign more than one, increment the **index=** parameter—you can see here that I've attempted to add **index=1** twice, and received an error message. **Ipconfig /all** will confirm that you've added the correct server address.

```
Netsh interface ipv4 add dnsserver name=2 address=10.0.1.1 index=1
```

Activating Windows

Server Core still requires activation, which is a two-step process that uses the **Slmgr** command. First, install a product key. Then activate Windows. Note that Server Core is compatible with enterprise key servers if your organization uses one of those. Run **Slmgr** without any parameters to get a pop-up dialog box of other things it can do; note that the dialog often appears *behind* the command-line window and there's no Task Bar to clue you in. If the command's output doesn't show up quickly, try moving the Cmd.exe window out of the way. *Don't* close it—if you do, press Ctrl+Alt+Delete to get to Task manager, and use the New Task menu option to run a new instance of Cmd.exe.

```
Slmgr –ipk your-product-key-here
```

After installing the key, activate it. This can take some time—wait for the dialog box indicating success or failure, and don't forget that it might appear *behind* the Cmd.exe window.

```
Slmgr -ato
```

Customize the Server

You'll probably want to customize the computer name at this point. Use the **hostname** command to find the current computer name, and then the **Netdom** command to change it to a new one.

```
Netdom renamecomputer old-name /newname:new-name
```

A reboot will be required afterwards, so use the **Shutdown /r** command to reboot.

```
Shutdown /r
```

Installing Roles

I generally like to install the DNS Server role myself so that I can customize it. After installing, you'll need to use the DNS administration console *on another computer* (such as your workstation) to connect to the Server Core computer and configure DNS. Server Core doesn't run any graphical admin tools. You could also use the **Dnscmd** command to configure DNS, if you're comfortable with it. To install the role, use the **Ocsetup** command; I prefer to get this going by using the **Start /w** command, which suspends the command prompt until **Ocsetup** finishes. If you don't do so, the command prompt immediately returns while the installation completes in the background, and you won't know when it's done.

```
Start /w ocsetup DNS-Server-Core-Role
```

Next, you'll need to create an unattended installation file for **Dcpromo** because its graphical wizard isn't available in Server Core. http://www.petri.co.il/creating-unattend-installation-file-dcpromo-windows-server-2008.htm is an excellent reference for Win2008 unattended Dcpromo files—note that the Win2008 syntax is a bit different and newer from the Win2003 one. Server Core *does* have Notepad, so you can use it to create your unattended file if needed. Server Core's Notepad uses an older set of file dialog boxes; pay close attention to these Win95-vintage dialog boxes because they work differently from the newer ones you're used to.

The unattend file tells Dcpromo if you're creating a new domain, a new domain controller in an existing domain, a whole new forest, or whatever. Read through the options carefully! You can also use Dcpromo on an existing full Windows installation (although not on an existing domain controller) to create an unattend file; just run through the Dcpromo wizard and, before you commit to installing AD, save your configuration in a file. That file can then be carried to Server Core (on a USB key, for example) and used with Dcpromo there.

```
[unattended]
unattendmode=fullunattended

[DCINSTALL]
UserName=Administrator
Password=P@ssw0rd
UserDomain=company.pro
DatabasePath=%systemroot%\ntds
LogPath=%systemroot%\ntds
SYSVOLPath=%systemroot%\SYSVOL
SafeModeAdminPassword=P@ssw0rd
CriticalReplicationOnly=no
InstallDNS=yes
DomainNetBIOSName=COMPANY
NewDomain=Forest
NewDomainDNSName=company.pro
RebookOnSuccess=Yes
SiteName=Default-First-Site-Name
ReplicaOrNewDomain=domain
ForestLevel=3
DomainLevel=3
```

With your unattended file ready, run **Dcpromo /unattend:*filename*** to start the AD installation process. You'll see plenty of output telling you what's happening.

```
Dcpromo /unattend:filename
```

Of course, a reboot is in order afterwards, and Dcpromo will handle that automatically. Once the server restarts, you can use Active Directory Users & Computers—again, from another computer—to begin managing your domain.

Tip, Trick, Technique 2: Read-Only Domain Controllers

Read-Only Domain Controllers (RODCs) are a new feature in Windows Server 2008 designed specifically for branch offices where the domain controller might not be as physically secure as you would like. A risk with less-secure computers is that the computer or its system hard drive might be stolen, giving an attacker the opportunity to break the encryption on the Active Directory database and then run a dictionary attack against stored passwords, potentially compromising every password in your domain. This isn't far-fetched; while breaking the database encryption would be time-consuming, a dictionary attack that used a pre-generated "rainbow table" (which are readily available) can begin cracking passwords in just minutes. The idea with an RODC is that it doesn't store *any* passwords, so stealing it (or the hard drive) really limits the amount of useful information an attacker can get hold of.

Password Caching

A downside to an RODC is that they don't store passwords—meaning the *primary* function of a domain controller, authentication, can't be performed. Actually, RODCs *can* perform authentication. What they do is contact a writable domain controller, which has passwords

stored, to handle the authentication; the RODC can then cache the password information locally. This allows authentication to occur when a writable domain controller isn't available—provided the user that the password information was retrieved for was cached in the advance. If the RODC is stolen, any cached passwords represent potential security vulnerabilities, but only *those* passwords need to be changed, not the entire domain. Simply force a password change on everyone in that office, and you're fine. You can specify, in advance, which accounts an RODC will cache. Any other accounts will only authenticate if a writable domain controller is available at the time.

You *can* pre-populate the password cache: When adding cache-allowed accounts to the RODC's Password Replication Policy, click Prepopulate Passwords to make this happen. This ensures that all cacheable passwords are cached immediately, without waiting for each of those users to log on.

Caveats

The presence of an RODC doesn't negate the need for a writable domain controller. Any changes made to the domain, including user password changes, need to contact a writable domain controller; Windows clients handle this automatically, but you do need to ensure that branch office connectivity is sufficient to handle these contacts. A branch office that happens to have an active domain administrator might not offer acceptable performance because the administrator would essentially be working over the WAN to administer the domain. Joining a computer to the domain also requires contacting a writable domain controller, and Group Policy administration requires a writable domain controller.

One concern with RODCs is that *certain* information, in addition to passwords, *is* stored locally, including account lockout status. When an RODC locks an account, that lockout is *forwarded* to a writable domain controller but not "replicated" in the AD sense of the term. If the lockout occurs while the WAN link is down, however, no writable domain controller will receive the lockout notice. The AD management tools will *not* show the lockout, but the account *will* be locked on *out the RODC*—although even the RODC's management tools will not show the lockout because it isn't officially in the domain database, yet. ADSIEdit does show the lockout on the RODC, in the lockoutTime attribute (which isn't the attribute the AD management tools look at to see whether an account is locked). Normal account-unlocking methods won't work because they rely on a writable domain controller and the RODC isn't one. The main way to unlock the account is to restore WAN connectivity, allowing the user to authenticate normally. Unfortunately, restoring the WAN link will also *immediately unlock the account* because the writable domain controllers in your domain will overwrite the RODC's lockout status almost immediately. Thus, if the account was locked for a good reason—such as an attempted attack—the account will now be free for another try, and you might not even know that it had been locked on the RODC at all, if no user complained about it.

Filtered Attributes

Some third-party applications that store data in AD may store sensitive information that you don't want replicated to RODCs. In these cases, you can configure a set of attributes in the schema that will not replicate to an RODC—this is called the *RODC filtered attribute set.*

Even if an attacker modifies an RODC and attempts to request replication of these attributes, the domain will deny the request. However, be aware that domain controllers running older versions of Windows *will* honor a request for these attributes because those older domain controllers don't recognize the filtered attribute set. The filtered attribute set is configured on the domain's Schema Master, which must be running Windows Server 2008 in order for the attribute set to be properly stored.

Read-Only DNS

RODCs can also host the Windows DNS Server service, and the RODC will be able to replicate all application directory partitions that DNS uses. Clients can query the DNS server as they would any other for name resolution. However, the DNS service will be read-only and will be unable to accept updates of any kind.

Typically, clients use the DNS server in their site as their "preferred" DNS server, and send updates—including updates for A, AAAA, SRV, and other record types. An RODC has no means of accepting these updates, however, and when queried for an SOA record, the RODC will return the name of a writable domain controller running the DNS service rather than that of the RODC. This is how a secondary DNS server handles updates for zones that are not AD-integrated zones, and it's a well-established DNS standard operation.

The RODC does have a bit of smarts: When it refers a client to a writable DNS server, it waits for a bit and then tries to query any records related to that client from the DNS server. That gives the client a chance to contact a writable DNS server, submit updates, and lets the RODC quickly pull those updates down so that its local, read-only DNS database is up to date. This works only if at least one of your DNS servers is on a Windows Server 2008 computer, and if that computer has registered an NS record for itself in the DNS database.

Bonus: Administrative Separation

RODCs allow you to delegate *local* administrative authority—such as the ability to run backup and restore operations—without delegating any domain authority. This allows branch office personnel to perform basic administrative tasks on the RODC computer without having any broader permission within AD itself.

Application Compatibility

Generally speaking, RODCs are compatible with any AD-enabled application. However, write-intensive applications don't do well when they're co-located with only RODCs because write requests have to be referred to a writable domain controller, which might under some circumstances (such as interrupted WAN connectivity) be unavailable. The write referral is potentially the most difficult operation; while applications that use standard directory programming interfaces should have no problem, not every application is built using these standard interfaces. Only testing will determine whether all your applications will be RODC-compatible, and if they're not, the developer will need to make corrections. Applications built using Microsoft's Active Directory Services Interface (ADSI) will automatically handle write referrals; developers often prefer the higher-performance LDAP, however, which carries referrals but does not automatically "chase" them as ADSI does.

Most Microsoft applications work fine against an RODC, although the following ones require special steps if actually installed on an RODC (see http://technet.microsoft.com/en-us/library/cc732790.aspx for details):

- Office Live Communications Server
- Office Outlook
- SharePoint Services
- SQL Server 2005
- DHCP Server

Probably the big challenge is Exchange Server, which does not use RODCs. Outlook clients, however, can use an RODC for read-only Global Catalog address book lookups.

Generally, "special steps" means creating appropriate service accounts on a writable domain controller and then ensuring they replicate to the RODC before beginning the software installation.

Ultimate Security

The best security is achieved when RODCs are combined with two other Windows Server 2008 features: BitLocker and a hardware Trusted Platform Module (TPM). The latter technologies provide volume-wide encryption for the system drive, providing yet another layer an attacker must work through in order to access data. The TPM helps by checking the hardware configuration against what's stored in its secure memory to ensure that nothing has been tampered with before allowing the host to boot—helping to prevent unauthorized hardware modifications that might be used to subvert or compromise the OS. Combined, these three features don't make it impossible to hack a domain controller, but they make it pretty impractical and ultimately unrewarding.

Edge Cases

Aside from security and logon performance at branch offices, RODCs offer benefits in a couple of odd scenarios. One is a line-of-business application, which will only work if physically installed on a domain controller—a poor practice, to be sure, but one which some administrators face. An RODC will work with many of these applications (subject to the caveats mentioned earlier), providing a sort of special-purpose domain controller just for that application. RODCs also provide better security in some extranet scenarios, where you need to expose authentication capabilities but don't necessarily want passwords to be compromised.

RODCs: Pros and Cons

- Better security for domain controllers that might not be physically secure
- Better logon performance for branch offices with limited WAN connectivity
- Potential user support and security issues around account lockouts
- Potential application compatibility concerns

Tip, Trick, Technique 3: No More CHKDSK

In the past, a corrupted file or segment of disk storage could typically only be repaired by taking the entire server offline and running on offline CHKDSK. No more: Under Win2008, a new service detects corrupted files automatically and spawns a thread that attempts to fix them. The affected files remain offline, meaning applications—including the Server service that provides file sharing—can't access the file but everything else on disk remains accessible and the server itself remains online. Access to the file is restored automatically if Windows is able to repair the corruption; if not, that area of disk is marked off-limits so that no other processes try to write files there.

You don't even need to do anything to take advantage of this feature, but you do need to be aware that it's happening. Client applications may display misleading "access denied" messages, for example, when a file is under repair. It's not a permissions issue but rather the fact that Windows has taken the file "out of service" while attempting to fix it. Your first troubleshooting step, therefore, should be to see whether you can access the file as a full-privilege administrator to eliminate permissions as a possible cause of the error (keeping in mind that with User Account Control enabled on your own workstation, you won't appear to be a real administrator unless you explicitly launch Explorer or another application "as Administrator.")

Tip, Trick, Technique 4: Internet Information Services 7

IIS 7 is pretty much a total re-write of IIS. It's such a drastic change, in fact, that Win2008 continues to ship with the old IIS6 management tools so that you can manage existing IIS6 installations! Many of the common IIS management tasks have changed completely, all the way down to how you install and set up FTP services.

All-New Console

As before, IIS maintains a top-level, server-wide set of configuration options, and Web sites can inherit these. You can also configure per-site settings on each individual Web site. What's new is how you do so: The IIS Management console has been vastly extended, so making everything accessible from a single Properties dialog box was no longer practical. Instead, the server and each site present a page of configuration icons, and double-clicking one opens a page for that specific item.

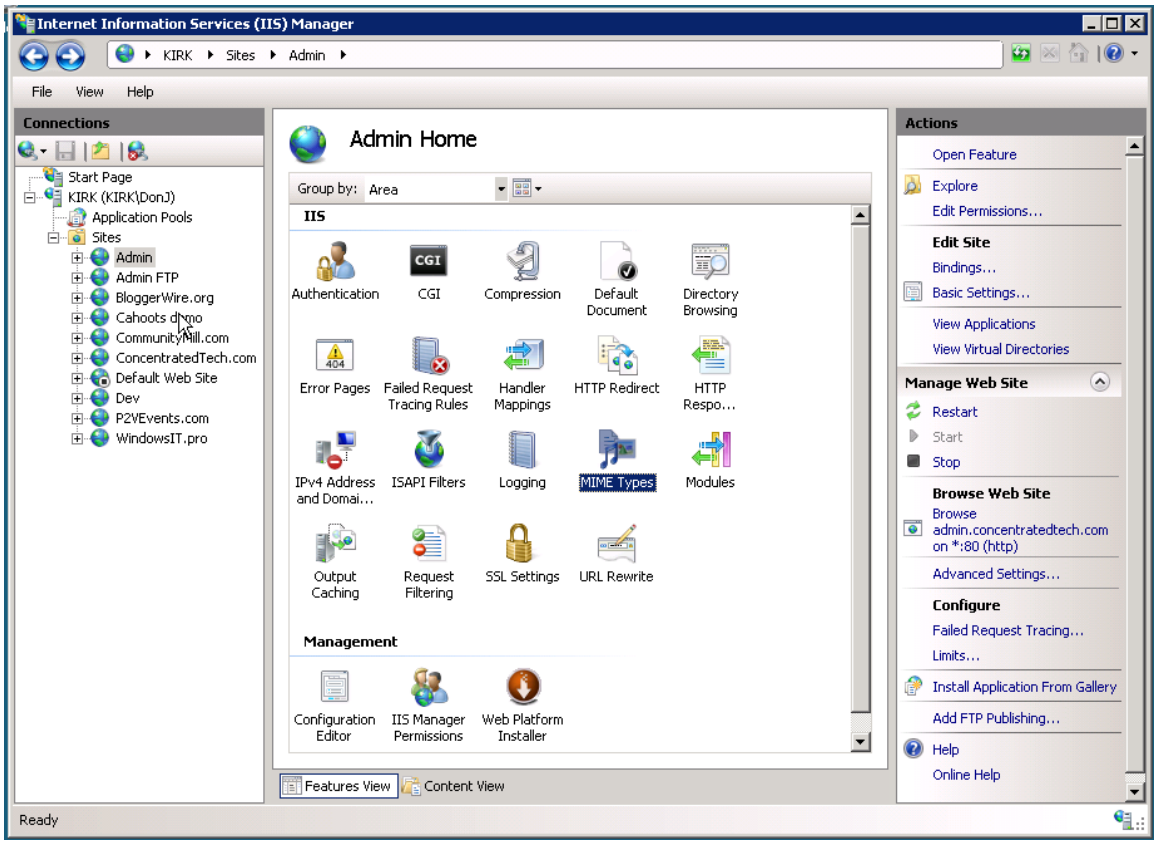

Figure 1: IIS 7 Manager.

In most cases, the layout of these item-specific pages is new, too, because most of them are also extensible. Authentication, for example, is no longer a set of four radio buttons but rather a list of all installed authentication choices, and the ability to enable or disable each.

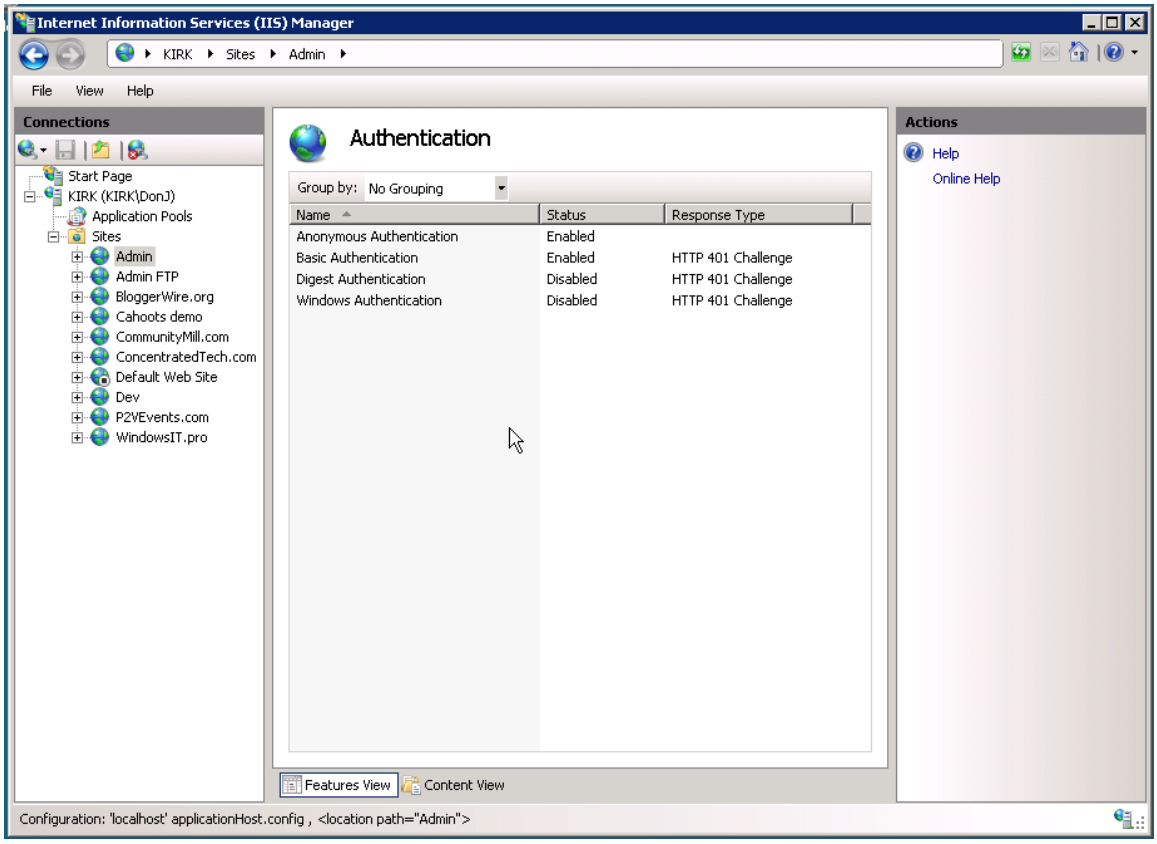

Figure 2: Authentication configuration.

In some cases, it can be a bit tricky to find the setting you're after: Editing site bindings, for example (which determines the host names, IP addresses, and port numbers a site will respond to), is accessed from the right-hand sidebar, as are functions for stopping and restarting sites.

Application Pools

IIS continues to host sites within Application Pools, which are used to configure the number of threads servicing one or more sites, the user identity the sites operate under, and so forth. Unlike IIS 6, though, IIS 7 will—by default—create a new App Pool for each new Web site you create. It's an easy-to-change setting when you create a new site, but it's also easy to miss, and there are disadvantages to having one Application Pool per site.

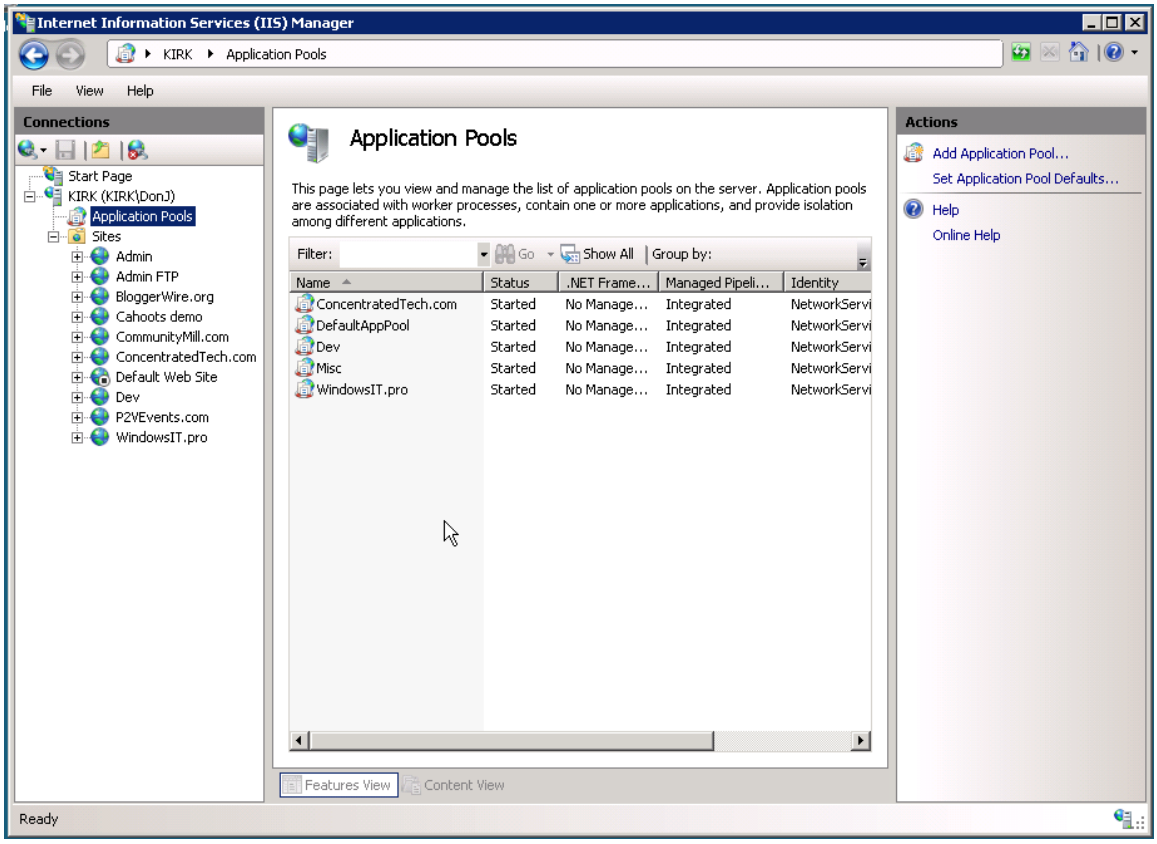

Figure 3: Configuring Application Pools.

Each Application Pool consists of at least one thread of execution. Infrequently-used sites can easily share a single thread, while busier sites may benefit from multiple threads for parallel servicing of multiple incoming requests. Each thread, however, brings a small amount of overhead, so having one thread apiece for several less-busy sites may actually hamper server performance. The moral? Don't accept the defaults until you've' decided whether that's suitable for your specific situation.

Web Platform Installer

IIS 7 is probably the most extensible version of IIS ever, and Microsoft—as well as third parties—is making numerous extensions available. To make installing all of these easier, Microsoft has created the Web Platform Installer, which is available for free at www.iis.net. This installer queries available extensions and offers to install them for you—up to and including non-Microsoft platforms such as PHP, which enjoys better support than ever under IIS 7.

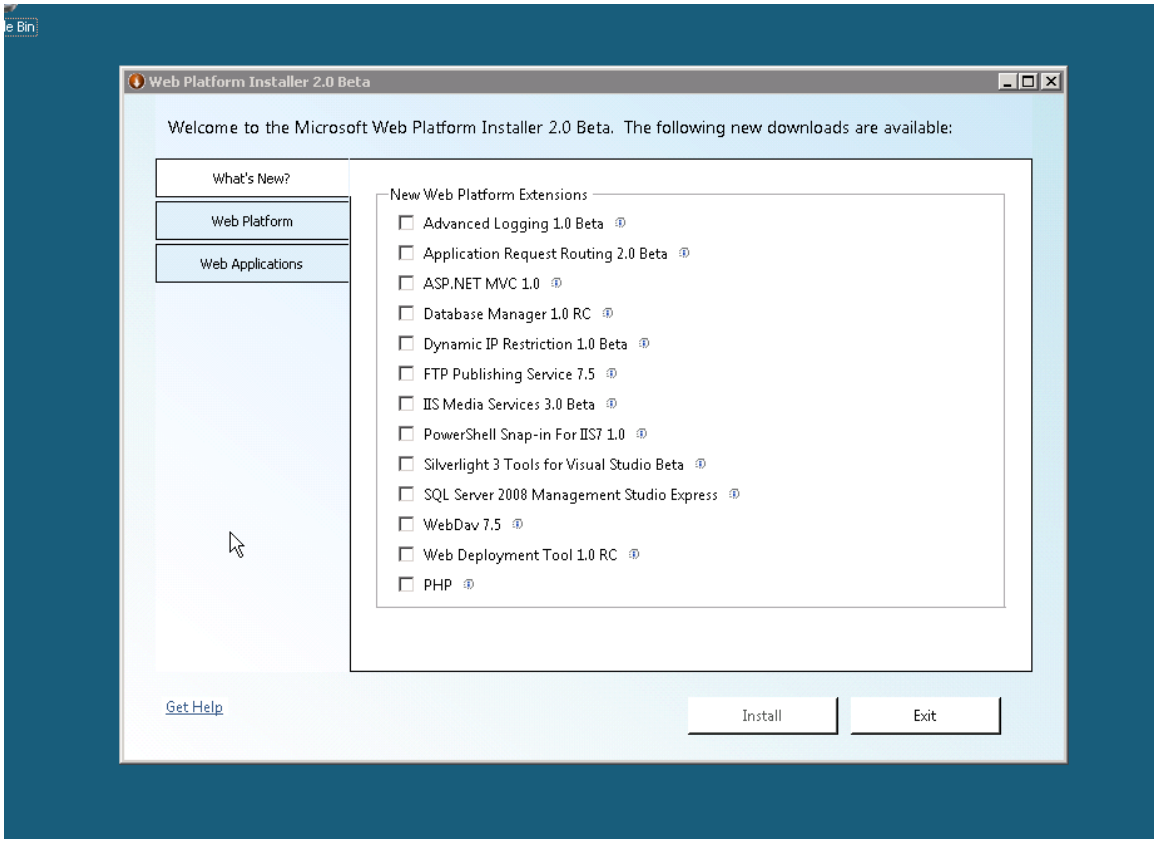

Figure 4: Web Platform Installer.

Once set up, the Installer is available from the management page of any Web site. It'll remind you a bit of the easy-to-use Web-based management consoles that many hosting companies provide: You can even use it to install selected pre-packaged Web applications such as DasBlog, Drupal, Subtext, WordPress, and more.

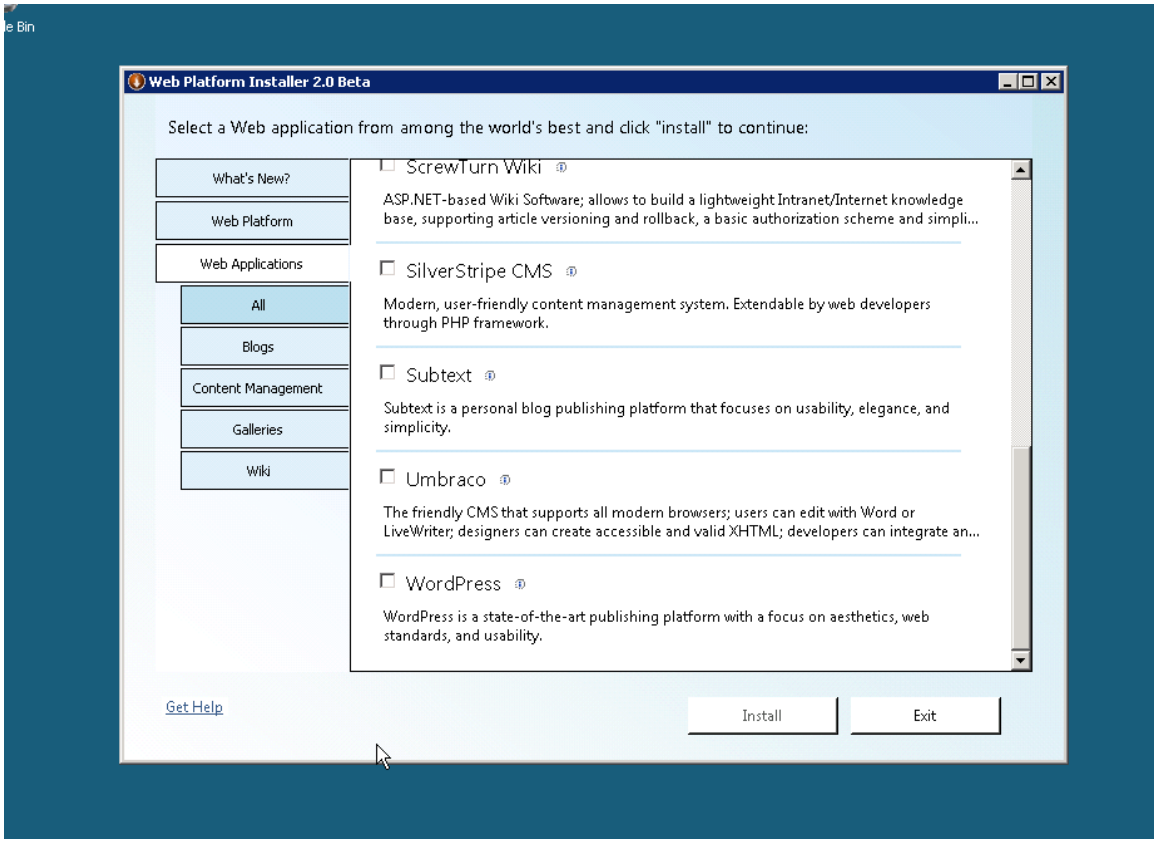

Figure 5: Installing Web applications.

The Web Platform Installer is probably the easiest way to extend IIS we've ever had.

FTP

Although Win2008 includes the old FTP Publishing Service, you don't want it. In fact, if it's already installed, un-install it using Server Manager (go to the Web Server role, and click "Remove Role Services"), and use the new FTP service available through the Web Platform Installer.

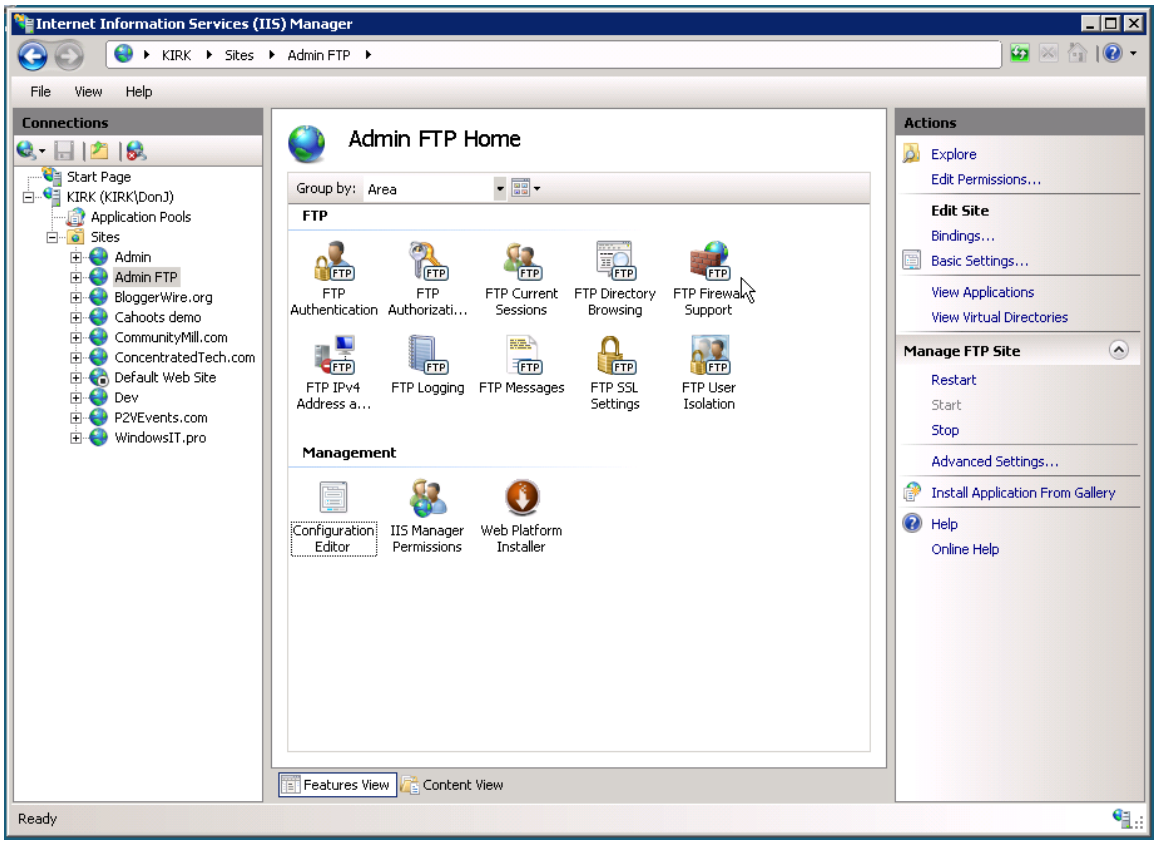

Figure 6: The new FTP service.

This new service, which *cannot* be installed if the old IIS6-compatible FTP Publishing Service is installed, offers secure FTP, FTP firewall support, better FTP logging, and much more. It's a more scalable and more efficient FTP service that can be managed from within the IIS7 Manager console (the old service requires the use of the old IIS6 console).

URL Rewriting

One of the most annoying aspects of using IIS, as opposed to something like Apache, is the availability of URL rewriting. Numerous popular Web applications make use of this feature to provide search engine-friendly URLs as well as other capabilities. Apache makes it easy by using an industry-standard rewriting syntax in a simple text file, named .htaccess. Dropping an .htaccess file into a Web site's root folder, or any subfolder, enables rewriting for that site or folder. Under IIS, third-party commercial tools were required to provide this capability—until IIS7. The Web Platform Installer can be used to get a free URL rewriting module, which appears as a configuration option in IIS Manager.

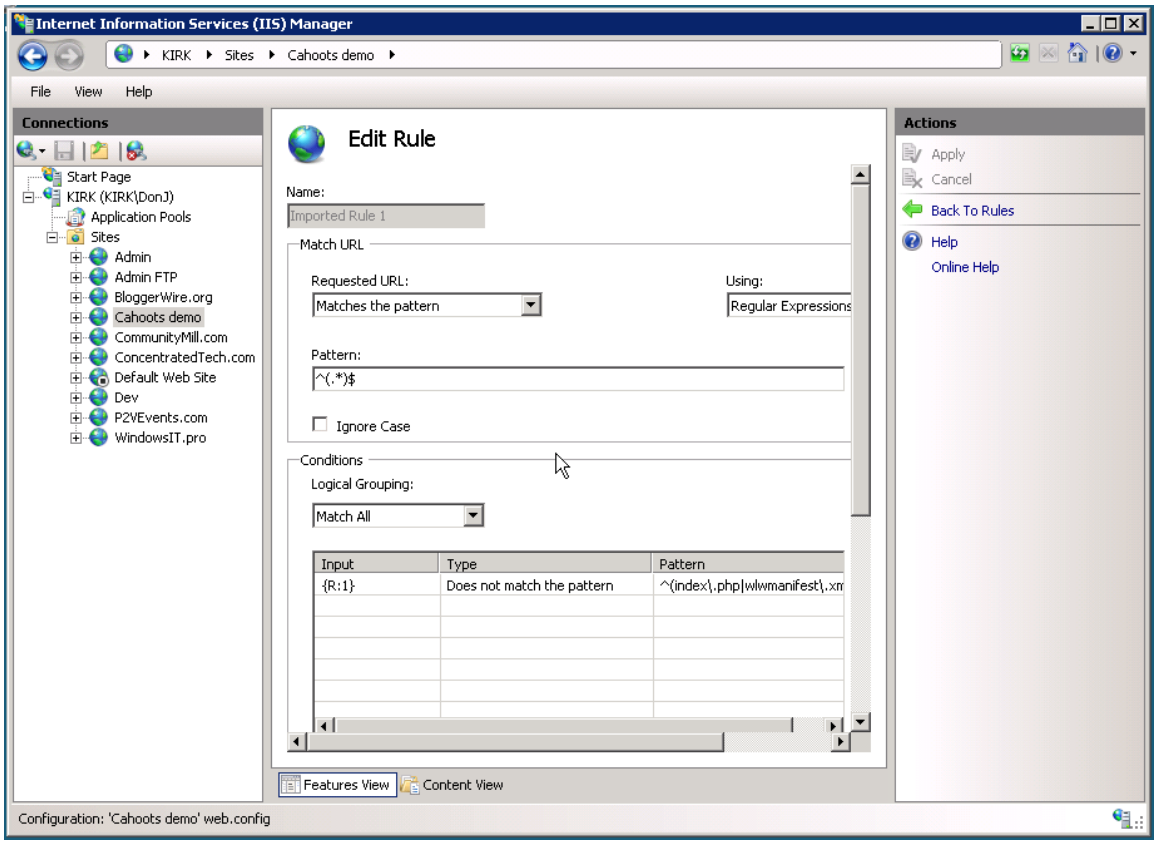

Figure 7: Editing a URL rewrite rule.

Although IIS still (somewhat irritatingly) doesn't use simply .htaccess files, it *can* import those files into its own URL rewriting module. You can create custom rules, and a wizard provides shortcuts for creating common types of rules. For example, one rule (see Figure 8) can be used to remove the "www" from incoming requests, forcing users to "realtimepublishers.com" rather than "www.realtimepublishers.com." This is a common trick for helping search engines see only one version of the site and avoiding the "duplicate content penalty" many engines impose when they think they're seeing the same content on two different Web sites (one starting with www, and the other without).

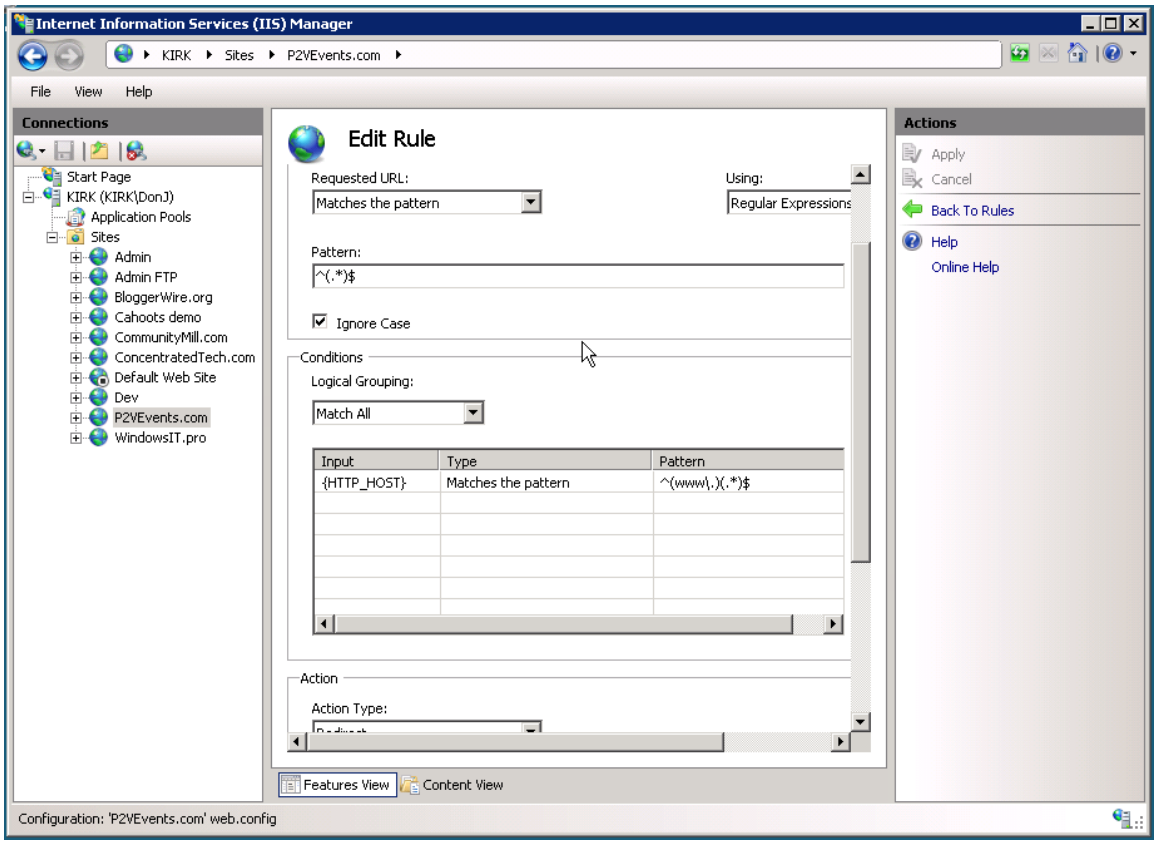

Figure 8: The "No-WWW" rule.

To create this rule, create a new, blank URL rewrite rule. Set it to match the pattern:

```
^(.*)$
```

Which is a regular expression (regex) for any URL coming into the site (the site's bindings will ensure that only requests intended for that site make it this far). Under the rule's conditions, specify a single condition:

- Input: **{HTTP_HOST}**
- Type: **Matches the pattern**
- Pattern: **^(www\.)(.*)$**

And set the action to redirect to:

```
http://your-site-URL-without-www{PATH_INFO}
```

Select the "Append query string" check box and make the redirect a Permanent (301) redirect. This will grab whatever URL the user was trying to reach, if it starts with "www," and redirect to the non-www version of the URL. You can also use this to capture old domain names and permanently redirect them to a new one.

Download Additional eBooks from Realtime Nexus!

Realtime Nexus—The Digital Library provides world-class expert resources that IT professionals depend on to learn about the newest technologies. If you found this eBook to be informative, we encourage you to download more of our industry-leading technology eBooks and video guides at Realtime Nexus. Please visit http://nexus.realtimepublishers.com.