Chapter 1

What's New in Windows Server 2008 and 2008 R2

As explained in the introduction, Microsoft had to delay Server 2008's appearance by years because of the change in the network security environment...a change for the worse. But once delivered, Server 2008 offered some very nice upgrades, and its "surprise" little sibling, Server 2008 R2, brought even more. Now, explaining all of those features would take much more than a chapter (which is, of course, why we wrote a book!), but let's use these first few pages to give you the lay of the land. Now, we realize that some reading this book are just getting started with Microsoft networking, and so for them, *everything* is new, but many others of you reading this already know tons about Windows networking, and would just like a summary of what's new in Server—this chapter summarizes that and where to find it in the book.

By now, I've sat through about a zillion Microsoft presentations on Windows Server 2008 and then R2, and they all start the same way, so apparently I'm required by law (or at least by custom) to present the following as the first heading when doing a 2008/R2 overview.

Server 2008 and R2 Goals

Hmmm, I wonder what Microsoft's goals were in creating 2008 and R2. C'mon, this is an easy one. Microsoft basically has two goals in creating new versions of Server: to sell more Server licenses and to keep Windows Server a moving target so that Sun, Apple, Linux distros, and the other various Unix variants can't catch up. Fortunately, however, in order to accomplish those goals, Microsoft has to offer us some new tools, and really, Microsoft needs to offer us some new tools that will solve the most annoying problems that the version of Server we *currently* use can't solve.

Server 2008 and R2's new features basically fall into six categories:

- Active Directory
- New setup technologies
- Changes to the underlying operating system
- Networking changes
- File and print services
- Web-based services

And if you're wondering why I haven't included a bullet point with a name like "Changes to security," that's because better security was a major design goal in 2008 and, to a lesser extent,

2008 R2; therefore, you'll see that new and better security is "baked into" a large portion of these technologies, in addition to AD, the underlying OS, networking, file and print, and web-based features. The following sections offer a brief overview of what's new in this book and where to read more about those features.

AD Changes

As you may know, Active Directory (AD) is in many ways the keystone piece of Windows networking, in other words, the central database of user and machine authentication data. Server 2008 and 2008 R2's ADs include several useful new capabilities. Collectively, the new features simplify AD security and disaster recovery, offer new admin tools, and let us run our ADs more flexibly.

Read-Only Domain Controllers

As you may know (but if you don't, then don't worry—we'll cover this stuff later in the book), domain controllers (DCs) are the set of distributed servers that hold the information needed for users to authenticate to services, and as such they are pretty valuable things. If a bad guy can get close enough to steal a DC, then he could—with time, tools, and luck—retrieve usernames and passwords, enabling him to then attack your network with ease.

Now in a perfect world, we'd keep our DCs locked up, safe, and secure behind strong walls. But it ain't a perfect world, and sometimes we need to locate a DC in a branch office where there is no place to physically secure that DC—a fact that's kept many a Windows security professional up late at night since AD's inception in 2000. Windows Server 2008 offers a solution in the form of a new kind of DC called a *read-only domain controller* (RODC).

RODCs can act as DCs in that they can help users get authenticated to the file, print, and web servers that those users need to access. They are, however, different from traditional "read-write" DCs in that you can control exactly how much information they contain. (In a pre-2008 AD domain, every DC knows exactly the same amount as every other DC in that domain.) For instance, suppose you run an organization with 150 employees that has a branch office that houses only 10 of those employees. A standard read-write DC placed locally in that branch office would contain the usernames and passwords of all 150 employees. An RODC, in contrast, could be configured to hold only the passwords for the 10 employees who work in that office. Thus, if that RODC is stolen, then the most that the bad guys can extract from it are the 10 usernames and passwords, leaving the other 140 safe.

Read more about RODCs in Chapter 22.

New Windows Backup

There are many things that I like about Windows servers from NT Server 3.1 onward, but there's one thing that I have *never* liked about Windows servers: disaster recovery. Prior to Server 2008, the built-in backup software in Windows could back up all the files on a system, but if you'd lost the server altogether, then putting those files back together on a different piece of hardware so that you could bring the dead server back to life wasn't very easy. When it comes to functioning servers and backups, it's often the truth that the sum is *significantly* more than the sum of its parts. So, if you're looking at the smoking or waterlogged remains of a Server 2003 system after a fire or a flood, then having a complete set of backup tapes for that server is often cold comfort. For a long time, Windows just plain didn't have a very good disaster recovery solution.

Versions of Windows from Vista onward address this with a backup tool called Windows Server Backup, which can do a "bare-metal" backup, meaning that if you lose a server because of a hardware failure, then you can get a completely different piece of server hardware and restore the Windows Server Backup to that new hardware, and your new server hardware will behave just as your old server did. And the restore probably won't take more than an hour or so. Part of what "will behave just as your old server did" includes is DC functionality. Even if you have only one DC in your organization—something I strongly recommend against!—Windows Server Backup can turn what would have been a disaster in Server 2003 to just an annoying waste of time. Unfortunately, Windows Server Backup is a somewhat mixed blessing, because it cannot back up to tapes—it needs a network share or local hard drive to back up.

Read more about Windows Server Backup in Chapter 18.

Fine-Grained Password Policies

Active Directory does a lot of things besides just keep a list of user account names and passwords, but if we had to choose the most important of its tasks, I think it'd be reasonable to say that protecting and maintaining passwords would be that task.

That's why it's so odd that Windows authentication systems, both pre- and post-AD, lack some really obvious things. For example, using an English word as a password in a modern network is a painfully stupid thing to do, because about the only thing keeping the bad guys from guessing your password is the fact that Windows supports (theoretically, at least) around 300,000,000,000,000,000,000,000,000,000,000,000,000 different passwords, and even with today's computers, that's a *lot* of possibilities to brute-force. In contrast, there are only about 400,000 words in the English language, and brute-forcing that can be done in about 20 seconds by the average PC—simplicity itself, which is why anyone using an English word as a password might just as well choose a blank or the word *password*. So, why is there no feature in Windows that lets you scan user accounts for passwords that match English words? Unfortunately, Windows Server 2008 doesn't fix *that* problem, but it does address another long-term password annoyance.

In ADs based on Windows 2000 Server or Windows Server 2003, there's no way to tell Windows, "Let the nonadministrative users change their passwords every six months, and let them use eight-character passwords, but make the administrators change their passwords every 60 days, and require the passwords to be at least 12 characters." With Server 2008's ADs, in contrast, you can create as many different password policies as you like and attach them to groups and/or particular users. Called *fine-grained* password policies, they just may be the single coolest new thing in Windows Server 2008's AD.

Read more about fine-grained password policies in Chapter 6.

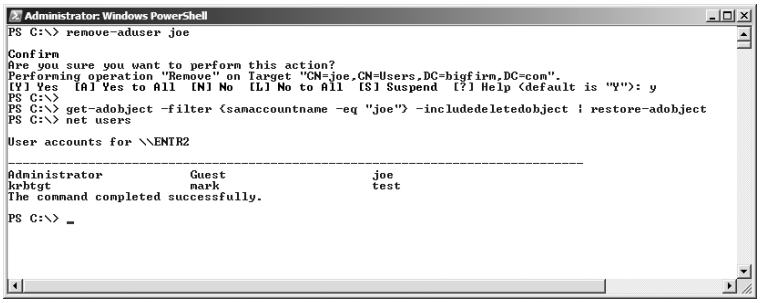
AD Snapshots and the AD Recycle Bin

Hey, we're all human. Once in a while, we drag the wrong icon or click OK when we meant to click "Oh, God, no, don't delete that user account!" It's in these times that we've got an unpleasant task ahead of us: rebuild or restore a user account or, if it's been a really bad day, rebuild or restore an entire organizational unit *and* the hapless users deleted with the OU. (OUs are essentially "sub-ADs" in that you can partition a piece of your organization and give someone god-like control of that piece of your organization while keeping him powerless to mess with the rest of the organization.) Windows Server 2008 and R2 each offer technologies intended to simplify undeleting AD objects.

Server 2008 introduced a notion that was a partial solution to the AD undelete problem, sort of an 8-foot ladder "solution" for scaling a 13-foot wall "problem." The idea is that you can, in a twinkling, take a complete backup of your Active Directory—a backup that is fast and lightweight but that you can't restore. I know, that sounds about as useful as a set of rectangular cement tires, but it can, with a bit of work, allow you to do something interesting: run a program named `dsamain` on a domain controller that lets you look at that backup as if it were a separate running Active Directory, almost like being able to fire up Active Directory Users and Computers (the program that most folks use to create and manage user accounts, often abbreviated ADUC), and tell it to go back in time three weeks, and voila! You see AD as it existed three weeks ago, not as it is now. You could then run various sorts of applications to retrieve information from that bygone AD to simplify rebuilding an accidentally deleted object. It's not the best answer, agreed, but it's potentially useful in being able to relatively quickly document what's changed in AD over time.

The better answer to the "How do I undelete a user account?" question comes with Server 2008 R2 and a thing called the AD Recycle Bin. Although its name makes you imagine some nice graphical user interface (GUI) tool like the Recycle Bin that has sat on Windows desktops for the past 15 years, R2's undeletion tool is actually a set of command-line tools that honestly ain't pretty, but they get the job done. You can see it in action in Figure 1.1.

FIGURE 1.1
Sample AD Recycle Bin run



Learn more about AD snapshots and the AD Recycle Bin in Chapter 18.

PowerShell and AD Administrative Center

Ever since the advent of Windows, Microsoft has shipped operating systems whose administrative tools have, in the main, been graphically based tools; in fact, many Windows administrators can go weeks at a time without having to open a command line. That's good in that it means learning Windows administration is easier for new administrators than it would be for novices trying to learn Unix/Linux administration, because that latter group of operating systems is more heavily dependent on command-line administrative tools than GUI-based administrative tools.

What being command-line-centric does for the Unix/Linux world, however, is to make automating administrative tasks easier in Unix/Linux than it would be to automate many Windows administrative tasks. (You can put a command-line instruction into a batch file, which can then automate whatever task you're trying to accomplish. You can't, in contrast, put mouse clicks in a batch file.) So, Microsoft is trying to give Windows the "automate ability" that it lacks and that Unix and Linux have with a new command shell called PowerShell. It's designed to let you take boring, repetitive tasks and automate them easily (once you get over the initial learning curve, which is a mite steep), and so Microsoft intends to make PowerShell as important an administrative platform as is the host of GUI tools that exist today.

All of the good intentions in the world, however, are of no use if some of the Windows product groups choose not to build PowerShell cmdlets (PowerShell commands, pronounced "command-lets") to control their part of Windows. For example, 2008 R2 includes no PowerShell cmdlets to administer essential Windows networking tools like DHCP (the software that hands out Internet addresses to computers in your network) and DNS (the software that keeps track of which computer at what network address has what name). If you're rusty on DNS and DHCP, then you might consider picking up a copy of *Mastering Windows Server 2008 Networking Foundations*, our beginner's book on Windows networking. With Windows Server 2008 R2, however, the AD team has released more than 70 AD-related PowerShell cmdlets, one of which is `Restore-ADobject`, the heart of the AD Recycle Bin you just read about. You can see some more PowerShell in action in Figure 1.2.

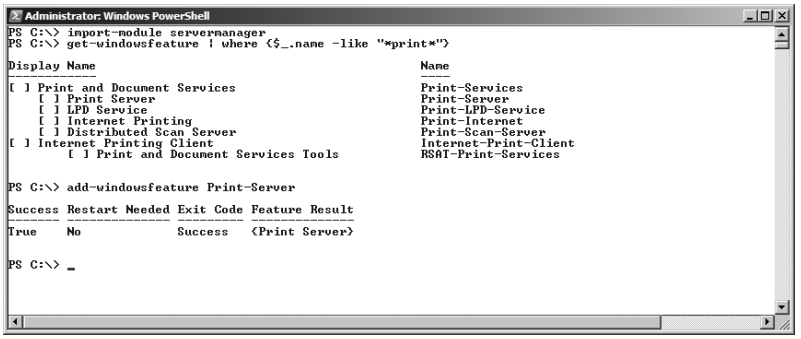


FIGURE 1.2
Using PowerShell to install a server role

The version of PowerShell in Windows Server 2008 R2, version 2.0, even lets you create GUI tools, which is sort of interesting, and Windows Server 2008 R2 ships with an example PowerShell GUI AD administration tool called the Active Directory Administrative Center (ADAC), giving you two user management tools: ADUC and ADAC. You'll find it on the Start ➤ All Programs ➤ Administrative Tools menu on any Server 2008 R2 domain controller or any R2 or Windows 7 system where you've installed the Remote Server Administrative Tools (RSAT). We haven't covered it in any detail in the book because it is essentially no more than a differently organized subset of ADUC, but if you feel like a bit of variety, fire it up some time and see how you like it.

You can read about the various AD cmdlets throughout the book.

DCPromo Improvements

AD administrators have used the Active Directory installation wizard—or, as it's better known, `DCPromo`—to convert Windows servers to DCs (called *promotion*) or to convert DCs to simple member servers (called *demotion*) since Windows 2000. Server 2008 brought a number of changes to `DCPromo` in that you can now create the RODCs discussed earlier, *and* `DCPromo` will now actually write scripts for you, enabling you to automate `DCPromo` itself. Cool, eh? (*I* asked the Microsoft folks for that—well, me and about a million other people.)

You'll see `DCPromo`'s new features in Chapters 6, 22, and 23.

OS Changes Under the Hood

Strictly speaking, a book on the Windows Server 2008 R2 operating system would say nothing about Active Directory or DNS or DHCP or file and print services because technically those are

all nothing more than applications on the same level as Word or Excel...just stuff that sits atop the platform that is the core operating system. So, let's get purist for a moment and ask, "What's really new in Server 2008 and 2008 R2?" Server 2003's operating system platform was fairly solid, as anyone who's running a 2003 system right now can attest, but 2008 and R2 saw dozens of small changes, most of which work silently in the background and can be safely ignored; however, there are a few big OS changes that admins should know about.

R2 Is 64-Bit Only

It's been coming for quite some time, but starting with Server 2008 R2, it's official: you must have 64-bit hardware to run Server. This isn't a big surprise, particularly when Exchange Server 2007 (Microsoft's email server product) shipped in a 64-bit-only manner. But given that anything fast enough to run 2008 can run 2008 R2, it may frustrate a few admins who didn't know that R2 was coming so fast and so decided to save a couple of bucks and buy 32-bit hardware for their 2008 servers, only to see R2's new features and wish that they could just upgrade their 2008 boxes to R2 boxes.

As I write this, it's actually becoming a bit difficult to even *find* 32-bit server hardware, and computer memory (RAM) is becoming cheap enough that even a small outfit can afford servers with 16GB (that is, a bit more than 16 billion bytes) of RAM. (4GB RAM is the limit that most 32-bit *software* can access, and Windows imposes an additional limitation in that half of that 4GB is set aside for the operating system and applications—and remember that "applications" includes AD, SQL Server, Exchange Server, and the like—so even if you did buy a 16GB server and put most 32-bit versions of Server on it, the software would use just 2GB of RAM for the apps and 2GB for the OS, and the remaining 12GB would do nothing but heat the room.)

Why care about more RAM? More RAM means that there's enough room to hold entire databases in the system's RAM, which means much faster performance. What's that you say? You don't really *do* databases? Well, for example, AD is a database, Exchange Server is a database, and SharePoint is a database. In fact, "64-bitness" has been a major factor in deciding to go to Exchange Server 2007; my clients with large message stores tell me that Exchange Server just flies on a system with a ton of RAM.

Getting the Most from 64 Bits

64 bits is pretty neat, but merely having a 64-bit OS may not be enough to offer the sort of better performance that I've promised here. For example, a Server 2008 R2 system running a 32-bit version of SQL Server 2008 (Microsoft's database server product) on a server containing 16GB of RAM could use most of that 16GB to store its AD but only 2GB to hold SQL Server databases because the SQL Server database engine only sees 32-bit-sized memory. So, remember, to get the most out of a 64-bit system, you need 64-bit apps as well.

There's not much more to know about Server's "64-bitness," because either an application is built to support 64 bits or it isn't, so there's really no other coverage in the book.

Server Core

Years ago, I wrote a book called *Linux for Windows Administrators* (Sybex, 2002) wherein I explained Linux in Windows terms—the idea was to create a sort of "fast path" to Linux for those already knowledgeable about Windows. As I did the research for that book, I couldn't

help but notice a great strength of Linux over Windows: its ability to turn the GUI on or off at a whim without affecting the basic OS's server functionality.

On Unix/Linux systems, the GUI is nothing more than just another application, like Word or Solitaire on Windows. As you've already read, Unix/Linux systems boast a wide variety of command-line administration tools, and so it was quite easy to set up DHCP, DNS, or file servers without the need for a GUI, and that seemed pretty cool.

What's cool about it? Well, shutting off the GUI frees up RAM and CPU power, enabling a Unix/Linux server to be that—just a server. And when you think about it, how often are you really sitting at your server and administering it via its GUI? I'm sure the answer to that varies from system to system, but at least in my case, I don't think I'm actually logged onto our domain controllers either in person or via Remote Desktop Services (the new name for Terminal Services) more than about 1 percent of the time. If I could just turn the Windows GUI off when I was done doing some account management, think of how much faster our logins would be!

With Server 2008, Microsoft answered my prayers, kind of. You can choose to install a version of Server called Server Core that has, as you read in the introduction, no Start menu and a very, very limited GUI. There's no way to turn the full GUI back on temporarily in Server Core, but Server Core *is* a pretty powerful first step in that direction, and you'll learn how to set it up and get it going in Chapter 3. You'll also see how to administer a server from the command line pretty much throughout the rest of the book.

Hyper-V

Server virtualization—breaking one physical server up into a bunch of *virtual machines*—is one of the most significant changes in server management in the past 10 years. I wrote "server management" in lowercase because it's used not just in Windows Server but in various flavors of Linux, Unix, Sun Solaris, and so on. Being able to buy one big, powerful, reliable piece of hardware and fool it into believing that it's actually 10 or 20 smaller separate pieces of computer hardware and then installing separate server OSes on those bits of "virtual server hardware" has greatly simplified server management for operations big and small. Furthermore, it has solved a server management problem that has bedeviled server room planners for years: underutilized hardware. The tool that fools the computer into thinking that it is actually many separate computers is generically called a *virtual machine manager* (VMM).

You see, ever since the start of server computing, most organizations have preferred to put each server function—email, AD domain controller, file server, web server, database server—on its own separate physical server. Thus, if you needed a domain controller, a web server, and an email server for your domain, you would commonly buy three separate server computers, put a copy of Windows Server on each one, and make one a DC, one a web server (by enabling Internet Information Services, R2's built-in web server software, on the server), and one an Exchange server. You wouldn't do that because you *had* to for any technological reason but instead for a management reason: the web folks are probably different people from the Active Directory folks, who in turn are probably different people than the email folks. Rebuilding a three-in-one server, then, would require getting a lot of people together, and that seemed like a bad idea. The downside of this was that each of those servers would probably run at fairly low load levels: it wouldn't be surprising to learn that the DC ran about 5 percent of the CPU's maximum capacity, the web server a bit more, and the email server a bit more than that. Running a bunch of pieces of physical server hardware below their capacity meant wasting electricity, and that's just not green thinking, y'know? In contrast, buying one big physical server and using a

VMM to chop it up into (for example) three virtual servers would probably lead to a physical server that's working near capacity, saving electricity and cooling needs.

In the past 10 years, then, VMMs have become important bits of operating system software, and since 2004, Microsoft has been trying to become a recognized leader in the VMM field with products such as Virtual PC and Virtual Server, neither of which have garnered much respect. But with Windows Server 2008, Microsoft shipped an all-new and quite powerful VMM called Hyper-V Server that's slowly finding its way into data centers everywhere. You can read about Hyper-V in Chapter 29.

Networking Changes

Servers are no good without the ability to talk to one another, but—of course—the downside of being able to communicate with other systems means that *infected* systems can try to spread their malware joy. ("Want to secure your server? Easy...disconnect the Ethernet cable!") Server 2008 and R2 offer some networking changes to make Windows networking a bit faster and a bit more secure.

TCP

Windows Server 2008 brought two changes to Transmission Control Protocol (TCP) and Internet Protocol (IP), the Internet's central pair of protocols. Windows' IP stack now includes IP version 6 (IPv6), the backbone of a slowly growing and newer Internet that will soon complement and eventually supplant the IP version 4 (IPv4)–based Internet that we've used for decades. We chose not to include much IPv6 coverage in this book because of space considerations and the fact that, at the moment, none of my clients unfortunately has the slightest interest in IPv6.

> **For More on IPv6**
>
> IPv6 *is* covered in my Server 2008 audio course in some detail, if you need some background on this technology. That's also in part why I didn't cover Microsoft's VPN alternative DirectAccess, because DirectAccess requires that your network tunnel an IPv6 network over the IPv4 Internet—well, that and because the only clients that could take advantage of it were the top-priced Windows 7 Enterprise and Ultimate, because you need a certificate infrastructure, and because of a bunch of other stuff that's a bit exotic as I write this. In time, I'm sure that'll change, but not for a few years and certainly not before I get a chance to put another book together!

The TCP change, in contrast, cries out for coverage. Strange as it sounds, the Internet authorities decided way back in 1992 to enlarge the maximum size of TCP data blocks allowable on the Internet, but Windows essentially hasn't really supported those big blocks completely until Vista and Server 2008. Ordinarily, this would be a no-news bit of information, because you needn't do anything at all to get the benefit of larger TCP blocks and their attendant higher data transfer rates—it happens automatically. But many networks contain at least a small amount of creaky old network hardware that just plain can't handle big blocks, with the seemingly paradoxical result that Vista, Windows 7, 2008, and 2008 R2's big blocks actually *slow things down*, sometimes significantly. Don't misunderstand, this is not a bug in Windows—it's just a Windows improvement that can trip

over some long-unnoticed bugs in your hardware or your ISP's hardware. See how to smoke out and work around this problem in Chapter 4.

Network Access Protection (NAP)

Gone are the days when our company computers spent 99 percent of their lives inside the apparent safety of the organization's network firewalls, because we buy a lot more laptops than desktops nowadays, and it's getting pretty hard to find a cell phone that doesn't do email, which implies that the cell phone is a computer, has networking capabilities, and has an IP address.

As a result, you just never know what's going to happen when your users bring their Internet-enabled mobile devices back into your organization's network, and today's network administrator would be well justified to wonder each morning as the employees show up, "What fresh malware is this?" Many firewall and networking vendors, such as Cisco, have built so-called quarantine systems that refuse to give a system an IP address until it's been at least minimally patted down to assure that it's not infected, and Microsoft has a tool like that in Server 2008/R2 called Network Access Protection (NAP). It's still in its infancy (*very* early infancy!), so we didn't do extensive coverage of it, but you can get its basics in Chapter 20.

Secure Socket Tunneling Protocol (SSTP) VPN

Building some sort of quarantine system into our networks may exist only in the future for many of our networks, but virtual private networks (VPNs) are most assuredly *not* them—being able to punch through our firewalls from outside the office to get to company data and email is a "must-have" for almost every organization, and so you'll find VPNs on most networks today. Most folks choose to purchase a VPN appliance—there are many on the market and one for every budget—but Windows's Routing and Remote Access has always offered *some* kind of VPN. Unfortunately for Microsoft, however, encryption is the heart of any sort of virtual private networking system, and for a long time Microsoft insisted on creating its own home-brewed encryption system, and such systems are usually cracked, as was the case with Windows 2000 Server and Windows Server 2003's Point-to-Point Tunneling Protocol (PPTP) encryption technology. Windows Server 2008, however, offers a VPN technology built atop not a home-brewed encryption method but instead the well-known and well-trusted technologies used in SSL. Microsoft calls its SSL-based VPN a Secure Socket Tunneling Protocol (SSTP) connection. You'll get the scoop on implementing it (and other Microsoft VPN options) in Chapter 20.

New Setup Technologies

Rolling out more than four or five new servers soon gets a bit monotonous; we tire of clicking OK or Next as Setup runs on a soon-to-be-running system, and we long for a bit of automation. Microsoft has always included *some* sort of unattended installation abilities via various kinds of text setup files, but such automation tools have never appealed to many, because using them was considered to be only a hairsbreadth easier than just returning to clicking OK or Next.

The Setup programs for Windows Vista and newer, however, are far more flexible and far easier to automate than any you've ever seen from Microsoft; they incorporate a setup engine named Panther, and I can't recommend enough that you get to know Panther's new tools. Covering everything that Panther can do would constitute another book, so I'll point you to two places to find out more. First, I have some good information on working with Server 2008 and R2's Setup

programs in Chapter 2. Second, documenting what Windows Setup can do has become something of a sideline job, and I have some fairly extensive articles on `www.minasi.com`, where you can find technical newsletters #59–62, 65, 71, and 72. (They're free.)

New Management Tools

Any good networking operating system should offer ways to simplify the job of keeping one server or one thousand servers up and running with the smallest amount of effort possible on the part of the humans doing the server administration. No one operating system has *the* answer for server administration, but Windows Server has gotten a bit better in 2008 and R2 with some useful new tools.

Server Manager

Prior to Windows Server 2008, the main "general management tool" was the Computer Management snap-in, `compmgmt.msc`. In 2008 and R2, however, right-clicking Computer and choosing Manage brings up a new GUI tool named Server Manager. It's the way to add capabilities to a server, such as when you want the server to offer DHCP or DNS services. You can see it displaying your server's installed roles in Figure 1.3.

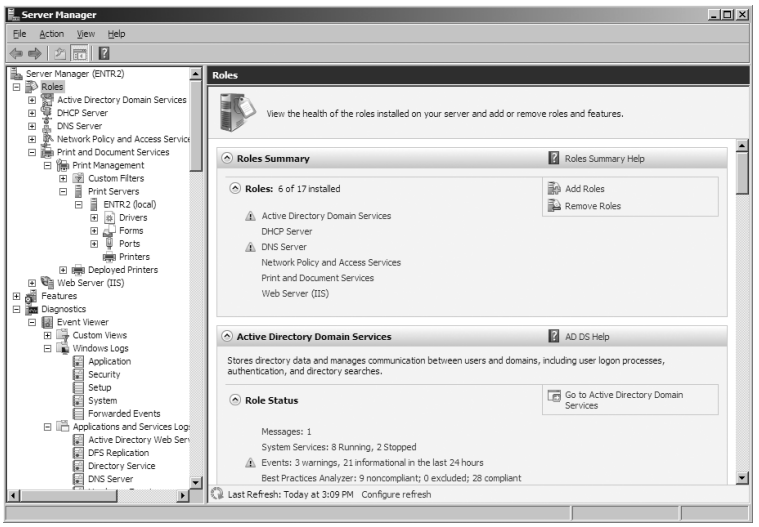


FIGURE 1.3
Server Manager in action

Server Manager's control of 2008's various capabilities—divided by 2008 into *roles* and *features*—extends as well to the command line, with tools named `servermanagercmd.exe` and `ocsetup.exe`.

With Server 2008 R2, Microsoft extended Server Manager's power by adding the ability to use Server Manager to control remote servers, or at least to control *some* things on remote servers, most notably adding and removing roles and features. It also scrapped both the `servermanagercmd` and `ocsetup` tools (or, rather, deprecated them), replacing them with the new `dism.exe` tool and a group of PowerShell cmdlets.

Where to read more on these tools? Well, Server Manager's use is so powerful and wide-ranging that it would have been ludicrous to put it all in one chapter, so you'll find it in at least part of a good number of chapters.

The New Remote Tools: WinRM and WinRS

It's the case all too often that new operating systems include some really important and useful features that go largely unnoticed. Windows Vista, Windows 7, Server 2008, and Server R2 contain one of those neat-but-largely-unknown features in a new network protocol called WinRM, short for "Windows Remote Management." To understand why WinRM is a great feature, let's consider what WinRM is intended to replace: a protocol known as the Remote Procedure Call (RPC).

Even if you've never heard of RPC, chances are that you've been using it for years. RPC's job is to allow one program to talk to another program, even if those programs are running on different computers. For example, if you've ever started up Outlook to read your email on an Exchange Server instance, then you've used RPC: it's how Outlook can tap Exchange on the shoulder and say, "Can I have my email, please?" Or if you've ever used an MMC snap-in like the DNS, DHCP, or Computer Management snap-in to remotely control those functions on a remote computer from your desktop, you've used RPC.

RPC is a protocol that has provided much service over the years, but it has one big problem: it's hard to secure. Microsoft invented RPC back in the days when there was no Internet, and the vast majority of LANs extended no farther than the distance from the first floor to the top floor in an office building, so security wasn't all that big a concern. Years later, when security became a big concern, Microsoft tried to retrofit security onto RPC with some optional changes wrought first by XP SP2, but by that point the horse was out of the barn, and requiring RPC security would just end up breaking hundreds or perhaps thousands of RPC-dependent applications.

Clearly, the time for a change in how Windows programs talk to each other had come, so Microsoft decided to adopt a protocol that did the same sort of thing that RPC did, with a few changes. First, it's not proprietary but is standards-based and platform-independent—there are similar implementations popping up on Linux and, I'm told, the Mac OS. Second, it's a modified form of HTTPS. Third, and not surprisingly, its communications are encrypted, and fourth, it requires authentication to use.

Components of Windows 2008 and R2 that use WinRM include event log collection; the ability to use the new Server Manager snap-in on remote servers; and my personal favorite, a secure remote command shell called Windows Remote Shell, or `winrs`. If you need a secure, low-bandwidth remote-control tool, look to `winrs`. (You can even retrofit it to XP and 2003 boxes with a hotfix referred to in Microsoft Knowledge Base article 936059.) Read more about WinRM in Chapter 14.

Remote Desktop Services: Terminal Services with a New Name and New Features

Ever since the latter days of NT 4, Windows has supported the notion of "terminal services," whereby a single powerful server creates, maintains, and presents a user's desktop across a network connection. This has the advantage of keeping the user's data, operating system, and user settings all on a server that's housed in a central location, making it simple to protect and back up all of that and allowing the user to view and interact with her desktop via a simple program running on just about any computer. Thus, she can start up Windows' terminal services client program, called the Remote Desktop Connection application, from anywhere and needn't worry

if the computer she's sitting on crashes, because her session state and data are far away on that terminal services server.

As time has gone on, Terminal Services has grown to support the familiar Remote Desktop feature of XP and later versions of Windows. The XP/2003 version of Terminal Services was pretty good, but Terminal Services fans will be quite pleased with the new stuff that Server 2008 and R2 bring. The smallest and least change is a name change, because Windows 7 and Server 2008 R2 refer no longer to Terminal Services but instead to Remote Desktop Services (RDS). The other changes? Well, there's actually a whole bunch of them, but perhaps the most interesting is the ability to deliver not an entire desktop but optionally just one or two applications, enabling network administrators to deploy a single application simply via RDS with a new feature called *remote applications*.

Get all the details on Remote Desktop Services and what's new in them in Chapter 14 (which discusses remote administration) and Chapter 25 (which covers server-based RDS).

New Group Policies and Tools

Group Policy lets you set up a set of central rules about what users and computers can do and that configure parts of the computer's software, enabling you to simply make one change to those central rules and see that change quickly propagate to dozens, hundreds, or thousands of machines in an enterprise. Group Policy has been around since Windows 2000, but Windows Server 2008 and Vista introduced some big changes. (Windows 7 and R2 really produced very little in Group Policy changes, short of being able to create and manage Group Policy objects via PowerShell.)

What got better? Plenty. Managing Group Policy objects (GPOs) got easier with the now built-in Group Policy Management console. Want to control some registry entry, but Microsoft hasn't created a Group Policy setting for it? A new class of policies called *Group Policy preferences* lets you essentially build your own new policies from scratch, and not just registry entries—you can deploy printers, shortcuts, and the like. SYSVOL filling up from tons and tons of repetitious copies of administrative templates? No problem—just create a Central Store, and Group Policy trims more SYSVOL fat than a truckload of acai berries could.

In addition to better Group Policy management tools, Windows Vista and newer let you control a wider variety of things via Group Policy. For example, Vista and newer let you control power management settings via Group Policy, so for example you could create a domain-wide policy requiring all screens to dim after, say, five minutes of inactivity. Microsoft claims that being able to lock down power settings centrally saved the company $6 million in power bills, although I've never quite figured out how Microsoft "proved" that. Chapter 8 covers domain-based group policies and policy management tools in detail, and all throughout the book you'll see examples of the new Group Policy settings.

New Event Viewer

It seems hard to imagine that the Event Viewer application would be a featured "star of the show," but it is, believe it or not. The new Event Viewer is a complete from-the-ground-up rewrite that resolves some old problems—no more must you limit the size of your event logs or face blue screens from a lack of nonpaged pool memory—and adds some welcome new features. You can now tell a bunch of Windows computers, "Please centralize your event log entries by storing them on such-and-such computer." For the command-line junkies, there's even a new CLI version of the Event Viewer named `wevtutil.exe`. You'll learn more about the Event Viewer in Chapter 17.

File and Print Sharing

Back before we ran web or email services on our Windows servers, we only used Server to share just two things: big hard drives and expensive printers. File and print are the oldest services offered by Microsoft networks...but apparently they're not too old to learn a few new tricks.

SMB 2.0

Windows' file server service bears the official name of SMB, which stands unhelpfully for "Server Message Block." (Blame IBM, not Microsoft, because an IBM guy first designed it.) SMB has changed little over its roughly 25 years of life, with its biggest changes being support of somewhat bigger block sizes so as to be able to make use of networks faster than 100Mbps (appeared in 2000) and adding digital signatures so as to foil man-in-the-middle attacks (appeared in 2001). Windows Vista, Server 2008, and newer versions, however, sport a somewhat-reworked version of SMB that handles slow networks better, handles encryption more intelligently, and cranks up throughput on file transfers between Vista, Server 2008, Windows 7, and R2 systems. Chapter 4 covers this in more detail.

More Reliable SYSVOL Replication

File shares are all pretty much the same...except for SYSVOL. SYSVOL is a built-in file share created, maintained, and replicated on every domain controller in your Active Directory. On that share, AD stores Group Policy information and login scripts (among other things), so if SYSVOL fails, then AD can't get you started in the morning.

Now, in general, SYSVOL works fairly well, so long as you're careful to leave at least one gigabyte free on whatever drive SYSVOL resides upon and you don't have an excess of bad karma, but if, for some reason, SYSVOL starts to get out of whack, then your AD is done for. I'm not exaggerating; in 10 years of looking at ADs, I haven't seen one damaged by problems in Active Directory replication of user accounts, passwords, and the like. I *have*, however, seen a few ADs brought low by SYSVOL problems.

The major root cause of SYSVOL problems is the SYSVOL replication engine, a service called the File Replication Service (FRS). Fixing FRS is a bit of a pain because the more one gets to know FRS, the more that one is forced to conclude that FRS was designed and coded on a weekend. A weekend with tequila.

Windows Server 2008–based ADs can, however, rip out that wobbly FRS and replace it with a faster, less-bandwidth-intensive, more self-healing service called DFSR. Getting to DFSR on SYSVOL is a laudable goal, but consider this: do you *really* want to rip out *one* SYSVOL replication engine and replace it with another? On a production Active Directory?

The answer is, "You surely do, and as quickly as possible!" so long as you know how, and Chapter 12 shows you the way.

Print Management Console and Printer Driver Isolation

Windows Server 2003 R2 didn't offer us much in the way of all-new stuff, but it did have one desirable feature: a new tool called the Print Management console (PMC). The PMC offered one-stop-shopping for examining and controlling all of your print queues and would generate Group Policy objects allowing you to deploy printers via group policies—the answer to a perennial request. So if you skipped 2003 R2 and now run 2008 or 2008 R2, then give the PMC a look.

You probably know that in order to control a piece of hardware, such as a mouse, a display board, a network card, or a printer, Windows needs a program called a *device driver*, more commonly shortened to *driver*. Drivers have been a source of great pain over the years for Windows administrators because they're just about the only piece of "privileged" code running on your Windows system that didn't get the level of beta testing that the rest of the OS did, which is why when you get a blue screen, the chances are good that the culprit was a third-party device driver.

What most folks don't know, however, is that the print drivers are strange critters when compared to other device drivers because—and I'm simplifying here—print drivers end up running in the system not as separate pieces of code (as is the case for most types of drivers) but instead attach themselves to the print spooler service. A side effect of this is that when a print driver fails, it takes down the whole spooler service with it.

Windows Server 2008 R2 lets you change that by allowing you to enclose print drivers in their own processes, with the result that if the print driver fails, it doesn't crash the spooler service and all of the other print drivers!

Chapter 13 covers printing, including PMC and printer isolation.

Web-Based Services

Finally, there's the subset of the Internet that's become more important than all the rest of the 'Net put together: the Web and related services. They're important to Windows, and they've seen some big changes in 2008 and 2008 R2.

Web Server (IIS)

Windows' file services may not have changed much over the years, but that's not the case for Windows' *web* server. In the 90s, Windows Server's built-in web server software, Internet Information Services (IIS), changed rapidly to keep pace with the then-breakneck speed of Internet innovation. Then, in the first decade of the 21st century, IIS still had to run hard to keep up, but this time it was keeping up with the security needs facing *any* piece of software directly connected to an Internet that got scarier with the month. 2000's IIS 5.0 showcased a powerful web platform, but its vulnerabilities (which made possible the Code Red and Nimda worms) led 2003's IIS 6.0 to be a much harder-to-crack server.

One key to hardening any server product is to keep the amount of code exposed to the Internet to a bare minimum; if a web server can support (for example) something called FastCGI but your website doesn't *need* FastCGI, then why run FastCGI on an Internet-facing server and risk the possibility that someone discovers a way to use IIS's FastCGI to hack the server? Clearly you wouldn't, so it'd be nice to just strip your web server software of the things that you aren't going to need. (Security folks call this "minimizing the attack surface." Sometimes we think they play too much *Halo*.)

The perfect web server, then, would be composed of dozens of small modules, each of which could be removed or added as needed to allow the web administrator to build a web server that did exactly what she needed it to do...but no more. That was the guiding light for Windows Server 2008's IIS 7.0, a complete overhaul of IIS including some of the latest security technologies, including WinRM. (When you're doing remote administration of an IIS 7 box, you're using that protocol rather than RPC.)

No one has hacked IIS 7 yet to my knowledge, nor have they taken down IIS 7.5, which is the update shipped with Windows Server 2008 R2. Web admins will also like the cleaner, task-oriented

interface of 7.*x*'s IIS Administration tools, which you can see in Figure 1.4. Even if you're not a web-slinger by trade, it's never a bad idea to understand the current Windows web server—so don't skip Chapter 16.

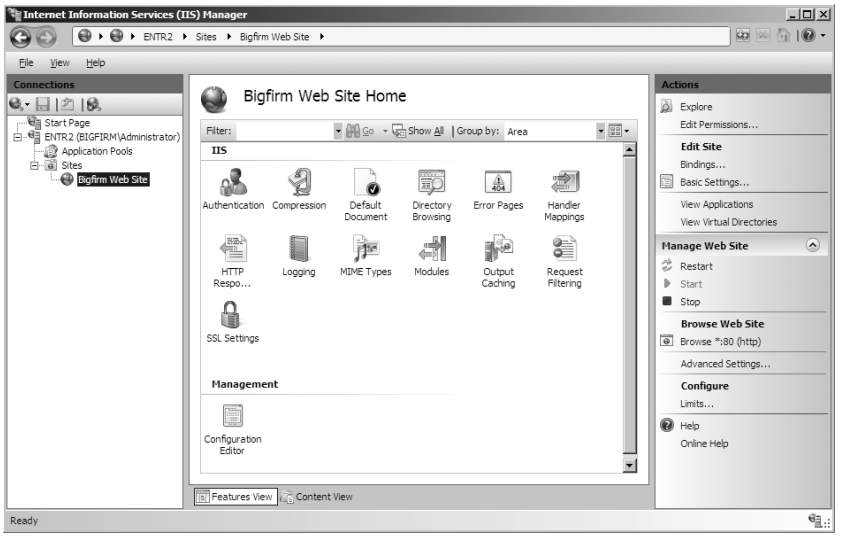


FIGURE 1.4
IIS's new management tool

FTP Server

Microsoft gets some things right and some things wrong. In a few cases, the company gets things terribly wrong, as was the case with the built-in File Transfer Protocol (FTP) server software that shipped with Windows for the past 15 years or so. It was so clunky, was so difficult to configure, offered such minimally useful logs and an inability to configure things that *should* have been childishly easy to configure (such as user home directories) that just about everyone that I know who needed a Windows FTP server ended up shelling out a few bucks for a third-party FTP server. (Many ended up buying WFTPD or WFTPD Pro from my friend Alun Jones, who wrote Chapter 19 of this book.)

With Windows Server 2008 and R2, however, things have changed considerably. As far as I can see, Microsoft tossed out all the FTP server code and rebuilt it from scratch, so if you need a Windows-based FTP server, flip over to the IIS chapter (Chapter 16) to learn about the new FTP server.

Windows Server Update Services (WSUS)

Having to download and deploy Windows patches on the second Tuesday of every month is a pain but necessary. For years, Microsoft has provided web-based services that simplify deploying and tracking patches in your domain. Windows Server 2008 Server is the first version of Server that includes the patch server, called Windows Server Update Services (WSUS). It's in the box now, so we cover it in Chapter 27.

Well, enough preliminaries—let's get on to the meat of the book!