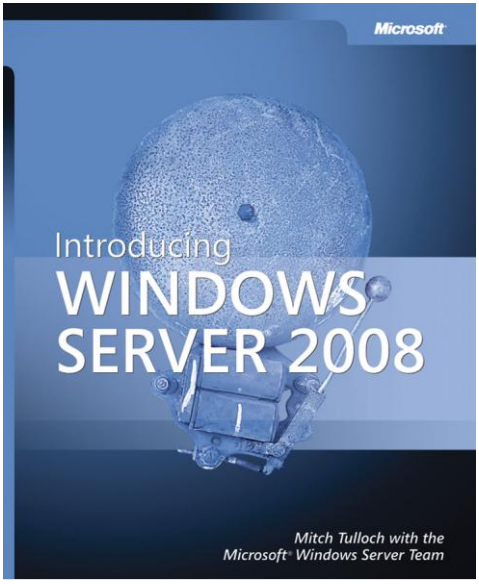


Introducing Windows Server 2008

Mitch Tulluch with the Microsoft Windows Server Team

To learn more about this book, visit Microsoft Learning at
http://www.microsoft.com/MSPress/books/11163.aspx

9780735624214
Publication Date: May 2007

Microsoft Press

Additional Resources for IT Professionals

Published and Forthcoming Titles from Microsoft Press

Windows Server

Microsoft® Windows Server® 2003 *Resource Kit*
Microsoft MVPs and Partners with Microsoft Windows Server Team
978-0-7356-2232-6

Microsoft Windows Server 2003 *Administrator's Companion* Second Edition
Charlie Russel, Sharon Crawford, and Jason Gerend
978-0-7356-2047-6

Microsoft Windows Server 2003 *Inside Out*
William R. Stanek
978-0-7356-2048-3

Microsoft Windows Server 2003 *Administrator's Pocket Consultant* Second Edition
William R. Stanek
978-0-7356-2245-6

Windows Client

Windows Vista™ *Resource Kit*
Tulloch, Northrup, Honeycutt, Russel, and Wilson with the Microsoft Windows Vista Team
978-0-7356-2283-8

Windows Vista *Administrator's Pocket Consultant*
William R. Stanek
978-0-7356-2296-8

Microsoft Windows® XP Professional *Resource Kit* Third Edition
The Microsoft Windows Team with Charlie Russel and Sharon Crawford
978-0-7356-2167-1

Microsoft Windows XP Professional *Administrator's Pocket Consultant* Second Edition
William R. Stanek
978-0-7356-2140-4

Microsoft Windows Command-Line *Administrator's Pocket Consultant*
William R. Stanek
978-0-7356-2038-4

SQL Server 2005

Microsoft SQL Server™ 2005 *Administrator's Pocket Consultant*
William R. Stanek
978-0-7356-2107-7

Microsoft SQL Server 2005 *Administrator's Companion*
Whalen, Garcia, et al.
978-0-7356-2198-5

Inside Microsoft SQL Server 2005: The Storage Engine
Kalen Delaney
978-0-7356-2105-3

Inside Microsoft SQL Server 2005: T-SQL Programming
Itzik Ben-Gan, Dejan Sarka, and Roger Wolter
978-0-7356-2197-8

Exchange Server 2007

Microsoft Exchange Server 2007 *Administrator's Companion*
Walter Glenn and Scott Lowe
978-0-7356-2350-7

Microsoft Exchange Server 2007 *Administrator's Pocket Consultant*
William R. Stanek
978-0-7356-2348-4

Scripting

Microsoft Windows PowerShell™ *Step by Step*
Ed Wilson
978-0-7356-2395-8

Microsoft VBScript *Step by Step*
Ed Wilson
978-0-7356-2297-5

Microsoft Windows Scripting with WMI: Self-Paced Learning Guide
Ed Wilson
978-0-7356-2231-9

Advanced VBScript for Microsoft Windows Administrators
Don Jones and Jeffery Hicks
978-0-7356-2244-9

RELATED TITLES

Microsoft Office SharePoint® Server 2007 *Administrator's Companion*
Bill English with the Microsoft SharePoint Community Experts
978-0-7356-2282-1

Microsoft Windows Security *Resource Kit* Second Edition
Ben Smith and Brian Komar with the Microsoft Security Team
978-0-7356-2174-9

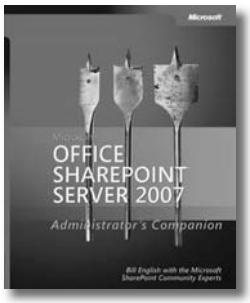
Microsoft Windows Small Business Server 2003 R2 *Administrator's Companion*
Charlie Russel and Sharon Crawford
978-0-7356-2280-7

Microsoft Internet Security and Acceleration (ISA) Server 2004 *Administrator's Pocket Consultant*
Bud Ratliff and Jason Ballard with the Microsoft ISA Server Team
978-0-7356-2188-6

Resources for IT Professionals

Administrator's Pocket Consultant

- Practical, portable guide for fast answers when you need them
- Focus on core operations and support tasks
- Organized for quick, precise reference—to get the job done



Administrator's Companion

- Comprehensive, one-volume guide to deployment and system administration
- Real-world insights, procedures, troubleshooting tactics, and workarounds
- Fully searchable eBook on CD

Resource Kit

- In-depth technical information and tools from those who know the technology best
- Definitive reference for deployment and operations
- Essential toolkit of resources, including eBook, on CD

Self-Paced Training Kit

- Two products in one: official exam prep guide + practice tests
- Features lessons, exercises, and case scenarios
- Comprehensive self-tests; trial software; eBook on CD

Available in 2008 from Microsoft Press

Windows Server

Windows Server® 2008 *Resource Kit*
978-0-7356-2361-3

Windows Server 2008 Active Directory® *Resource Kit*
978-0-7356-2515-0

Windows Server 2008 Virtualization *Resource Kit*
978-0-7356-2517-4

Windows Server 2008 Security *Resource Kit*
978-0-7356-2504-4

Windows® Administration *Resource Kit:* Productivity Solutions For IT Professionals
978-0-7356-2431-3

Windows Server 2008 Networking Guide
978-0-7356-2422-1

Windows Server 2008 TCP/IP Protocols and Services
978-0-7356-2447-4

Windows Server 2008 *Inside Out*
978-0-7356-2438-2

Windows Server 2008 Terminal Services
978-0-7356-2516-7

Windows Server 2008 *Administrator's Companion*
978-0-7356-2505-1

Windows Server 2008 *Administrator's Pocket Consultant*
978-0-7356-2437-5

Windows Group Policy Guide, Second Edition
978-0-7356-2514-3

Understanding IPv6, Second Edition
978-0-7356-2446-7

Internet Information Services

Internet Information Services (IIS) 7.0 *Administrator's Pocket Consultant*
978-0-7356-2364-4

Internet Information Services (IIS) 7.0 *Resource Kit*
978-0-7356-2441-2

Scripting

Windows PowerShell™ Scripting Guide
978-0-7356-2279-1

Windows PowerShell & Command-line *Administrator's Pocket Consultant*
978-0-7356-2262-3

Certification

***MCITP Self-Paced Training Kit* (Exams 70-640, 70-642, 70-643, 70-646): Windows Server Administrator Core Requirements**
978-0-7356-2508-2

***MCTS Self-Paced Training Kit* (Exam 70-640): Configuring Windows Server 2008 Active Directory**
978-0-7356-2513-6

***MCTS Self-Paced Training Kit* (Exam 70-642): Configuring Windows Server 2008 Network Infrastructure**
978-0-7356-2512-9

***MCTS Self-Paced Training Kit* (Exam 70-643): Configuring Windows Server 2008 Applications Platform**
978-0-7356-2511-2

***MCITP Self-Paced Training Kit* (Exam 70-646): Windows Server 2008 Administrator**
978-0-7356-2510-5

***MCITP Self-Paced Training Kit* (Exam 70-647): Windows Server 2008 Enterprise Administrator**
978-0-7356-2509-9

See our full line of learning resources at: **microsoft.com/mspress** *and* **microsoft.com/learning**

Microsoft®

PUBLISHED BY
Microsoft Press
A Division of Microsoft Corporation
One Microsoft Way
Redmond, Washington 98052-6399

Library of Congress Control Number: 2007924650

Printed and bound in the United States of America.

1 2 3 4 5 6 7 8 9 QWT 2 1 0 9 8 7

Distributed in Canada by H.B. Fenn and Company Ltd.

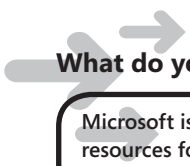
A CIP catalogue record for this book is available from the British Library.

Microsoft Press books are available through booksellers and distributors worldwide. For further information about international editions, contact your local Microsoft Corporation office or contact Microsoft Press International directly at fax (425) 936-7329. Visit our Web site at www.microsoft.com/mspress. Send comments to tkinput@microsoft.com.

Acquisitions Editor: Martin DelRe
Developmental Editor: Karen Szall
Project Editor: Denise Bankaitis

Body Part No. X13-72717

Table of Contents



What do you think of this book? We want to hear from you!

Microsoft is interested in hearing your feedback so we can continually improve our books and learning resources for you. To participate in a brief online survey, please visit:

www.microsoft.com/learning/booksurvey/

Chapter 3

Windows Server Virtualization

In this chapter:

Now that we've examined some possible usage scenarios for Microsoft Windows Server 2008, it's time to start digging deep into the features of the platform. But there are a *lot* of new features and enhancements in Windows Server 2008—why begin with virtualization?

Customer-facing answer? Need.

Technical answer for us IT pros? Architecture.

Why Enterprises Love Virtualization

Virtualization has been around in computing since the mainframe days of the late '60s. Those of us who are old enough to remember punch cards (carrying boxes of them around was a great way of getting exercise) might remember the IBM 360 mainframe system and the CP/CMS time-sharing operating system, which simulated the effect of each user having a full, standalone IBM mainframe at their fingertips. Each user's "virtual machine" was fully independent of those belonging to other users, so if you ran an application that crashed "your" machine, other users weren't affected.

PCs changed this paradigm in the '80s, and eventually gave users' *physical* machines that today are far more powerful than the mainframes of the '60s and '70s. But as desktop PCs began to proliferate, so did servers in the back rooms of most businesses. Soon you'd have two domain controllers, a mail server running Microsoft Exchange, a couple of file servers, a database server, a Web server for your intranet, and so on. Larger companies might have

dozens or even hundreds of servers, some running multiple roles such as AD, DNS, DHCP, or more.

Managing all these separate boxes can be a headache, and restoring them from backup after a disaster can involve costly downtime for your business. But even worse from a business standpoint is that many of them are underutilized. How does virtualization for x86/x64 platforms solve these issues?

Server Consolidation

In a production environment, having a server that averages only 5 percent CPU utilization doesn't make sense. A typical example would be a DHCP server in an enterprise environment that leases addresses to several thousand clients. One solution to such underutilization is to consolidate several roles on one box. For example, instead of just using the box as a DHCP server, you could also use it as a DNS server, file server, and print server. The problem is that as more roles are installed on a box, the uncertainty in their peak usage requirements increases, making it difficult to ensure that the machine doesn't become a bottleneck. In addition, the attack surface of the machine increases because more ports have to be open so that it can listen for client requests for all these services. Patching also becomes more complicated when updates for one of the running service need to be applied—if the update causes a secondary issue, several essential network services could go down instead of one.

Using virtualization, however, you can consolidate multiple server roles as separate virtual machines running on a single physical machine. This approach lets you reduce "server sprawl" and maximize the utilization of your current hardware, and each role can run in its own isolated virtual environment for greater security and easier management. And by consolidating multiple (possibly dozens of) virtual machines onto enterprise-class server hardware that has fault-tolerant RAID hardware and hot-swappable components, you can reduce downtime and make the most efficient use of your hardware. The process of migrating server roles from separate physical boxes onto virtual machines is known as *server consolidation*, and this is probably the number one driver behind the growing popularity of virtualization in enterprise environments. After all, budgets are limited nowadays!

Business Continuity

Being able to ensure business continuity in the event of a disaster is another big driver toward virtualization. Restoring a critical server role from tape backup when one of your boxes starts emitting smoke can be a long and painful process, especially when your CEO is standing over you wringing his hands waiting for you to finish. Having hot-spare servers waiting in the closet is, of course, a great solution, but it costs money, both in terms of the extra hardware and the licensing costs.

That's another reason why virtualization is so compelling. Because *guest* operating systems, which run inside virtual machines (VMs), are generally independent of the hardware on which the *host* operating system runs, you can easily restore a backed-up virtual server to a system that has different hardware than the original system that died. And using virtual machines, you can reduce both scheduled and unscheduled downtime by simplifying the restore process to ensure the availability of essential services for your network.

Testing and Development

IT pros like us are always in learn mode because of the steady flow (or flood) of new technologies arriving on our doorstep. I remember when I had to set up a test network to evaluate Exchange 5.5. I had eight boxes sitting on a bench just so I could try out the various features of the new messaging platform. These included an Exchange 5.0 server, an Exchange 4.0 server, and an MS Mail 3.0 server so that I could test migration from these platforms. Plus I had several different clients running on different boxes. The heat alone from these systems could have kept me warm during a Winnipeg winter.

Testing new platforms is a lot easier today because of virtualization. I can run a half dozen virtual machines easily on a single low-end server, and I can even set up a routed network without having to learn IOS by enabling IP routing on a virtual Microsoft Windows XP machine with two virtual NICs. Architects can benefit from virtualization by being able to create virtual test networks on a single server that mimic closely the complexity of large enterprise environments. Developers benefit too by being able to test their applications in isolated environments, where they can roll back their virtual machines when needed instead of having to install everything from scratch. The whole IT life cycle becomes easier to manage because virtualization reduces the time it takes to move new software from a development environment to test and then production.

Application Compatibility

Another popular use of virtualization today is to ensure application compatibility. Suppose you upgrade the version of Windows you have running on your desktop and find that a critical LOB application won't run properly on the new version. You can try several ways to resolve this problem. You can run the program in application compatibility mode, using the Application Compatibility Toolkit to shim the application so that it works on the new platform. Or you can contact the vendor for an updated version of the application. Another alternative, however, is virtualization: install Microsoft Virtual PC 2007 on each desktop computer where the user needs to use the problem application, install the old version of Windows as a guest OS, and then run the application from there.

Virtualization in the Datacenter

Virtualization also has a special place in the datacenter, as it lets you decouple workloads from hardware to make the best use of your resources. You can rapidly provision workloads as they

are needed so that your solutions can both scale up and scale out easily. Virtualization also simplifies automating complex solutions, though current virtualization products are limited in this regard. But that's where Windows Server 2008 comes in.

Virtualization Today

Virtualization today on Windows platforms basically takes one of two forms: Type 2 or Hybrid. A typical example of Type 2 virtualization is the Java virtual machine, while another example is the common language runtime (CLR) of the .NET Framework. In both examples, you start with the host operating system—that is, the operating system installed directly onto the physical hardware. On top of the host OS runs a Virtual Machine Monitor (VMM), whose role is to create and manage virtual machines, dole out resources to these machines, and keep these machines isolated from each other. In other words, the VMM is the virtualization layer in this scenario. Then on top of the VMM you have the guests that are running, which in this case are Java or .NET applications. Figure 3-1 shows this arrangement, and because the guests have to access the hardware by going through both the VMM and the host OS, performance is generally not at its best in this scenario.

Figure 3-1 Architecture of Type 2 VMM

More familiar probably to most IT pros is the Hybrid form of virtualization shown in Figure 3-2. Here both the host OS and the VMM essentially run directly on the hardware (though with different levels of access to different hardware components), whereas the guest OSs run on top of the virtualization layer. Well, that's not exactly what's happening here. A more accurate depiction of things is that the VMM in this configuration still must go through the host OS to access hardware. However, the host OS and VMM are both running in kernel mode and so they are essentially playing tug o' war with the CPU. The host gets CPU cycles when it needs them in the host context and then passes cycles back to the VMM and the VMM services then provide cycles to the guest OSs. And so it goes, back and forth. The reason why the Hybrid model is faster is that the VMM is running in kernel mode as opposed to the Type 2 model where the VMM generally runs in User mode.

Anyway, the Hybrid VMM approach is used today in two popular virtualization solutions from Microsoft, namely Microsoft Virtual PC 2007 and Microsoft Virtual Server 2005 R2.

The performance of Hybrid VMM is better than that of Type 2 VMM, but it's still not as good as having separate physical machines.

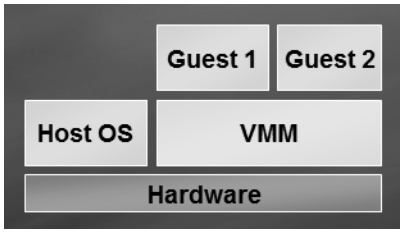


Figure 3-2 Architecture of Hybrid VMM

Note Another way of distinguishing between Type 2 and Hybrid VMMs is that Type 2 VMMs are *process virtual machines* because they isolate processes (services or applications) as separate guests on the physical system, while Hybrid VMMs are *system virtual machines* because they isolate entire operating systems, such as Windows or Linux, as separate guests.

A third type of virtualization technology available today is Type 1 VMM, or hypervisor technology. A *hypervisor* is a layer of software that sits just above the hardware and beneath one or more operating systems. Its primary purpose is to provide isolated execution environments, called *partitions*, within which virtual machines containing guest OSs can run. Each partition is provided with its own set of hardware resources—such as memory, CPU cycles, and devices—and the hypervisor is responsible for controlling and arbitrating access to the underlying hardware.

Figure 3-3 shows a simple form of Type 1 VMM in which the VMM (the hypervisor) is running directly on the bare metal (the underlying hardware) and several guest OSs are running on top of the VMM.

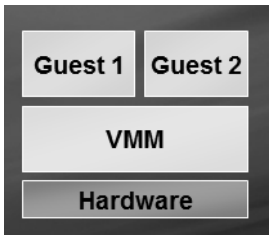


Figure 3-3 Architecture of Type 1 VMM

Going forward, hypervisor-based virtualization has the greatest performance potential, and in a moment we'll see how this will be implemented in Windows Server 2008. But first let's compare two variations of Type 1 VMM: monolithic and microkernelized.

Monolithic Hypervisor

In the monolithic model, the hypervisor has its own drivers for accessing the hardware beneath it. (See Figure 3-4.) Guest OSs run in VMs on top of the hypervisor, and when a guest needs to access hardware it does so through the hypervisor and its driver model. Typically, one of these guest OSs is the administrator or console OS within which you run the tools that provision, manage, and monitor all guest OSs running on the system.

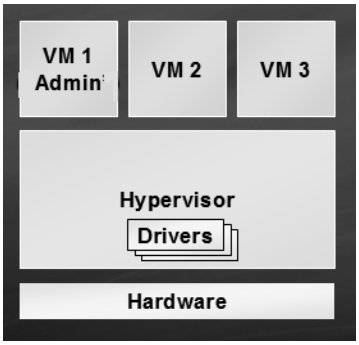


Figure 3-4 Monolithic hypervisor

The monolithic hypervisor model provides excellent performance, but it can have weaknesses in the areas of security and stability. This is because this model inherently has a greater attack surface and much greater potential for security concerns due to the fact that drivers (and even sometimes third-party code) runs in this very sensitive area. For example, if malware were downloaded onto the system, it could install a keystroke logger masquerading as a device driver in the hypervisor. If this happened, every guest OS running on the system would be compromised, which obviously isn't good. Even worse, once you've been "hyperjacked" there's no way the operating systems running above can tell because the hypervisor is invisible to the OSs above and can be lied to by the hypervisor!

The other problem is stability—if a driver were updated in the hypervisor and the new driver had a bug in it, the whole system would be affected, including all its virtual machines. Driver stability is thus a critical issue for this model, and introducing any third-party code has the potential to cause problems. And given the evolving nature of server hardware, the frequent need for new and updated drivers increases the chances of something bad happening. You can think of the monolithic model as a "fat hypervisor" model because of all the drivers the hypervisor needs to support.

Microkernelized Hypervisor

Now contrast the monolithic approach just mentioned with the microkernelized model. (See Figure 3-5.) Here you have a truly "thin" hypervisor that has no drivers running within it. Yes, that's right—the hypervisor has *no drivers at all*. Instead, drivers are run in each partition

so that each guest OS running within a virtual machine can access the hardware through the hypervisor. This arrangement makes each virtual machine a completely separate partition for greater security and reliability.

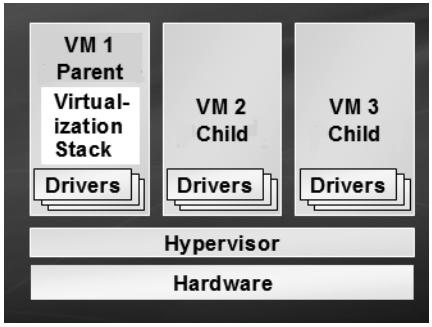


Figure 3-5 Microkernelized hypervisor

In the microkernelized model, which is used in Windows Server virtualization in Windows Server 2008, one VM is the parent partition while the others are child partitions. A partition is the basic unit of isolation supported by the hypervisor. A partition is made up of a physical address space together with one or more virtual processors, and you can assign specific hardware resources—such as CPU cycles, memory and devices—to the partition. The *parent partition* is the partition that creates and manages the *child partitions*, and it contains a virtualization stack that is used to control these child partitions. The parent partition is generally also the *root partition* because it is the partition that is created first and owns all resources not owned by the hypervisor. And being the default owner of all hardware resources means the root partition (that is, the parent) is also in charge of power management, plug and play, managing hardware failure events, and even loading and booting the hypervisor.

Within the parent partition is the virtualization stack, a collection of software components that work in conjunction with and sit on top of the hypervisor and that work together to support the virtual machines running on the system. The virtualization stack talks with the hypervisor and performs any virtualization functions not directory supplied by the hypervisor. Most of these functions are centered around the creation and management of child partitions and the resources (CPU, memory, and devices) they need.

The virtualization stack also exposes a management interface, which in Windows Server 2008 is a WMI provider whose APIs will be made publicly known. This means that not only will the tools for managing virtual machines running on Windows Server 2008 use these APIs, but third-party system management vendors will also be able to code new tools for managing, configuring, and monitoring VMs running on Windows Server 2008.

The advantage of the microkernelized approach used by Windows Server virtualization over the monolithic approach is that the drivers needed between the parent partition and the physical server don't require any changes to the driver model. In other words, existing drivers just work. Microsoft chose this route because requiring new drivers would have been a

showstopper. And as for the guest OSs, Microsoft will provide the necessary facilities so that these OSs just work either through emulation or through new synthetic devices.

On the other hand, one could argue that the microkernelized approach does suffer a slight performance hit compared with the monolithic model. However, security is paramount nowadays, so sacrificing a percentage point or two of performance for a reduced attack surface and greater stability is a no-brainer in most enterprises.

Tip What's the difference between a virtual machine and a partition? Think of a virtual machine as comprising a partition together with its state.

Understanding Virtualization in Windows Server 2008

Before I get you too excited, however, you need to know that what I'm going to describe now is not yet present in Windows Server 2008 Beta 3, the platform that this book covers. It's coming soon, however. Within 180 days of the release of Windows Server 2008, you should be able to download and install the bits for Windows Server virtualization that will make possible everything that I've talked about in the previous section and am going to describe now. In fact, if you're in a hotel after a long day at TechEd and you're reading this book for relaxation (that is, you're a typical geek), you can probably already download tools for your current prerelease build of Windows Server 2008 that might let you test some of these Windows Server virtualization technologies by creating and managing virtual machines on your latest Windows Server 2008 build.

I said *might* let you test these new technologies. Why? First, Windows Server virtualization is an x64 Editions technology only and can't be installed on x86 builds of Windows Server 2008. Second, it requires hardware processors with hardware-assisted virtualization support, which currently includes AMD-V and Intel VT processors only. These extensions are needed because the hypervisor runs out of context (effectively in ring 1), which means that the code and data for the hypervisor are not mapped into the address space of the guest. As a result, the hypervisor has to rely on the processor to support various intercepts, which are provided by these extensions. And finally, for security reasons it requires processor support for hardware-enabled Data Execution Prevention (DEP), which Intel describes as XD (eXecute Disable) and AMD describes as NX (No eXecute). So if you have suitable hardware and lots of memory, you should be able to start testing Windows Server virtualization as it becomes available in prerelease form for Windows Server 2008.

Let's dig deeper into the architecture of Windows Server virtualization running on Windows Server 2008. Remember, what we're looking at won't be available until after Windows Server 2008 RTMs—today in Beta 3, there is no hypervisor in Windows Server 2008, and the operating system basically runs on top of the metal the same way Windows Server 2003 does. So we're temporarily time-shifting into the future here, and assuming that when

we try and add the Windows Virtualization role to our current Windows Server 2008 build that it actually does something!

Figure 3-6 shows the big picture of what the architecture of Windows Server 2008 looks like with the virtualization bits installed.

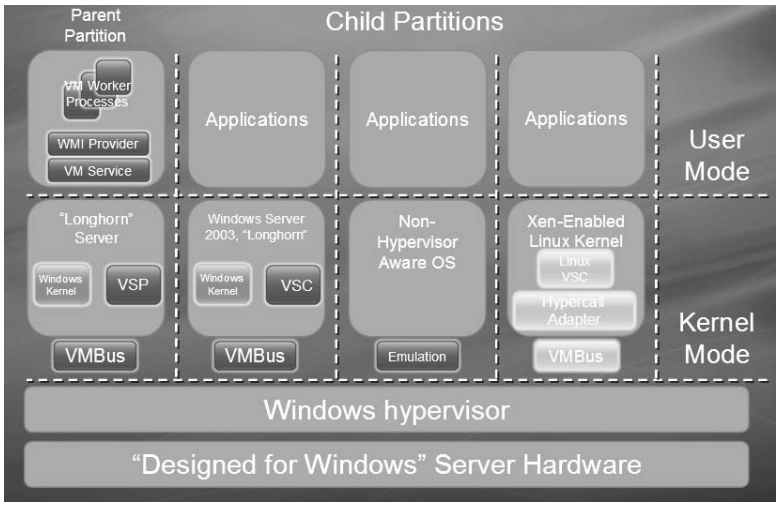


Figure 3-6 Detailed architecture of Windows Server virtualization

Partition 1: Parent

Let's unpack this diagram one piece at a time. First, note that we've got one parent partition (at the left) together with three child partitions, all running on top of the Windows hypervisor. In the parent partition, running in kernel mode, there must be a guest OS, which must be Windows Server 2008 but can be either a full installation of Windows Server 2008 or a Windows server core installation. Being able to run a Windows server core installation in the parent partition is significant because it means we can minimize the footprint and attack surface of our system when we use it as a platform for hosting virtual machines.

Running within the guest OS is the Virtualization Service Provider (VSP), a "server" component that runs within the parent partition (or any other partition that owns hardware). The VSP talks to the device drivers and acts as a kind of multiplexer, offering hardware services to whoever requests them (for example, in response to I/O requests). The VSP can pass on such requests either directly to a physical device through a driver running in kernel or user mode, or to a native service such as the file system to handle.

The VSP plays a key role in how device virtualization works. Previous Microsoft virtualization solutions such as Virtual PC and Virtual Server use emulation to enable guest OSs to access hardware. Virtual PC, for example, emulates a 1997-era motherboard, video card, network

card, and storage for its guest OSs. This is done for compatibility reasons to allow the greatest possible number of different guest OSs to run within VMs on Virtual PC. (Something like over 1,000 different operating systems and versions can run as guests on Virtual PC.) Device emulation is great for compatibility purposes, but generally speaking it's lousy for performance. VSPs avoid the emulation problem, however, as we'll see in a moment.

In the user-mode portion of the parent partition are the Virtual Machine Service (VM Service), which provides facilities to manage virtual machines and their worker processes; a Virtual Machine Worker Process, which is a process within the virtualization stack that represents and services a specific virtual machine running on the system (there is one VM Worker Process for each VM running on the system); and a WMI Provider that provides a set of interfaces for managing virtualization on the system. As mentioned previously, these WMI Providers will be publicly documented on MSDN, so you'll be able to automate virtualization tasks using scripts if you know how. Together, these various components make up the user-mode portion of the virtualization stack.

Finally, at the bottom of the kernel portion of the parent partition is the VMBus, which represents a system for sending requests and data between virtual machines running on the system.

Partition 2: Child with Enlightened Guest

The second partition from the left in Figure 3-6 shows an "enlightened" guest OS running within a child partition. An *enlightened guest* is an operating system that is aware that it is running on top of the hypervisor. As a result, the guest uses an optimized virtual machine interface. A guest that is fully enlightened has no need of an emulator; one that is partially enlightened might need emulation for some types of hardware devices. Windows Server 2008 is an example of a fully enlightened guest and is shown in partition 2 in the figure. (Windows Vista is another possible example of a fully enlightened guest.) The Windows Server 2003 guest OS shown in this partition, however, is only a partially enlightened, or "driver-enlightened," guest OS.)

By contrast, a *legacy guest* is an operating system that was written to run on a specific type of physical machine and therefore has no knowledge or understanding that it is running within a virtualized environment. To run within a VM hosted by Windows Server virtualization, a legacy guest requires substantial infrastructure, including a system BIOS and a wide variety of emulated devices. This infrastructure is not provided by the hypervisor but by an external monitor that we'll discuss shortly.

Running in kernel mode within the enlightened guest OS is the Virtualization Service Client (VSC), a "client" component that runs within a child partition and consumes services. The key thing here is that there is one VSP/VSC pair for each device type. For example, say a

user-mode application running in partition 2 (the child partition second from the left) wants to write something to a hard drive, which is server hardware. The process works like this:

1. The application calls the appropriate file system driver running in kernel mode in the child partition.
2. The file system driver notifies the VSC that it needs access to hardware.
3. The VSC passes the request over the VMBus to the corresponding VSP in partition 1 (the parent partition) using shared memory and hypervisor IPC messages. (You can think of the VMBus as a protocol with a supporting library for transferring data between different partitions through a ring buffer. If that's too confusing, think of it as a pipe. Also, while the diagram makes it look as though traffic goes through all the child partitions, this is not really the case—the VMBus is actually a point-to-point inter-partition bus.)
4. The VSP then writes to the hard drive through the storage stack and the appropriate port driver.

Microsoft plans on providing VSP/VSC pairs for storage, networking, video, and input devices for Windows Server virtualization. Third-party IHVs will likely provide additional VSP/VSC pairs to support additional hardware.

Speaking of writing things to disk, let's pause a moment before we go on and explain how pass-through disk access works in Windows Server virtualization. Pass-through disk access represents an entire physical disk as a virtual disk within the guest. The data and commands are thus "passed through" to the physical disk via the partition's native storage stack without any intervening processing by the virtual storage stack. This process contrasts with a virtual disk, where the virtual storage stack relies on its parser component to make the underlying storage (which could be a .vhd or an .iso image) look like a physical disk to the guest. Pass-through disk access is totally independent of the underlying physical connection involved. For example, the disk might be direct-attached storage (IDE disk, USB flash disk, FireWire disk) or it might be on a storage area network (SAN).

Now let's resume our discussion concerning the architecture of Windows Server virtualization and describe the third and fourth partitions shown in Figure 3-6 above.

Partition 3: Child with Legacy Guest

In the third partition from the left is a legacy guest OS such as MS-DOS. Yes, there are still a few places (such as banks) that run DOS for certain purposes. Hopefully, they've thrown out all their 286 PCs though. The thing to understand here is that basically this child partition works like Virtual Server. In other words, it uses emulation to provide DOS with a simulated hardware environment that it can understand. As a result, there is no VSC component here running in kernel mode.

Partition 4: Child with Guest Running Linux

Finally, in the fourth partition on the right is Linux running as a guest OS in a child partition. Microsoft recognizes the importance of interoperability in today's enterprises. More specifically, Microsoft knows that their customers want to be able to run *any* OS on top of the hypervisor that Windows Server virtualization provides, and therefore it can't relegate Linux (or any other OS) to second-class status by forcing it to have to run on emulated hardware. That's why Microsoft has decided to partner with XenSource to build VSCs for Linux, which will enable Linux to run as an enlightened guest within a child partition on Windows Server 2008. I knew those FOSS guys would finally see the light one day...

Features of Windows Server Virtualization

Now that we understand something about how virtualization works (or will work) on Windows Server 2008, let's look at what it can actually do. Here's a quick summary:

- Creates and manage child partitions for both 32-bit (x86) and 64-bit (x64) operating systems.
- Creates VMs that can use SMP to access 2, 4, or even 8 cores.
- Creates VMs that use up to 1 TB of physical memory. Windows Server virtualization can do this because it's built on 64-bit from the ground up. That means 64-bit HV, 64-bit virtualization stack, and so on.
- Supports direct pass-through disk access for VMs to provide enhanced read/write performance. Storage is often a bottleneck for physical machines, and with virtual disks it can be even more of a bottleneck. Windows Server virtualization overcomes this issue.
- Supports hot-add access to any form of storage. This means you can create virtual storage workloads and manage them dynamically.
- Supports dynamic addition of virtual NICs and can take advantage of underlying virtual LAN (VLAN) security.
- Includes tools for migrating Virtual Server workloads to Windows Server virtualization. This means your current investment in Virtual Server won't go down the drain.
- Supports Windows Server 2008 Core as the parent OS for increased security. I said this earlier, but it bears repeating here because it's important.
- Supports NAT and network quarantine for VMs, role-based security, Group Policy, utilization counters, non-Microsoft guests, virtual machine snapshots using Volume Shadow Copy Service (VSS), resource control using Windows System Resource Manager (WSRM), clustering, and a whole bunch of other things.

To put this all in perspective, take a look at Table 3-1, which provides a comparison between Virtual Server 2005 R2 and Windows Server virtualization.

Table 3-1 Comparison of Virtual Server 2005 R2 and Windows Server Virtualization Features

Feature	Virtual Server 2005 R2	Windows Server Virtualization
32-bit VMs	Yes	Yes
64-bit VMs	No	Yes
SMP VMs	No	Up to 8 core virtual machines
Hot-add memory	No	Yes
Hot-add processors	No	Yes
Hot-add storage	No	Yes
Hot-add networking	No	Yes
Max memory per VM	3.6 GM	> 32 GB
Cluster support	Yes	Yes
Scripting support	Using COM	Using WMI
Max number of VMs	64	No limit—depends only on hardware
Management tool	Web UI	MMC snap-in
Live migration support	No	Yes
Works with System Center Virtual Machine Manager	Yes	Yes

Note Virtual Server 2005 R2 Service Pack 1 will support Intel VT and AMD-V technologies, as well as VSS.

Managing Virtual Machines in Windows Server 2008

At the time of this writing, the MMC snap-in for managing virtual machines that is provided with Windows Server virtualization is still evolving, but I wanted to give you a quick preview here. Figure 3-7 shows the Windows Virtualization Management console for a near-Beta 3 build of Windows Server 2008. The console tree on the left displays the name of the server, while the Details pane in the middle shows a number of virtual machines, most of them in an Off state and two in a Saved state. The Actions pane on the right lets you manage virtualization settings, import virtual machines, connect to a virtual machine, and perform other tasks.

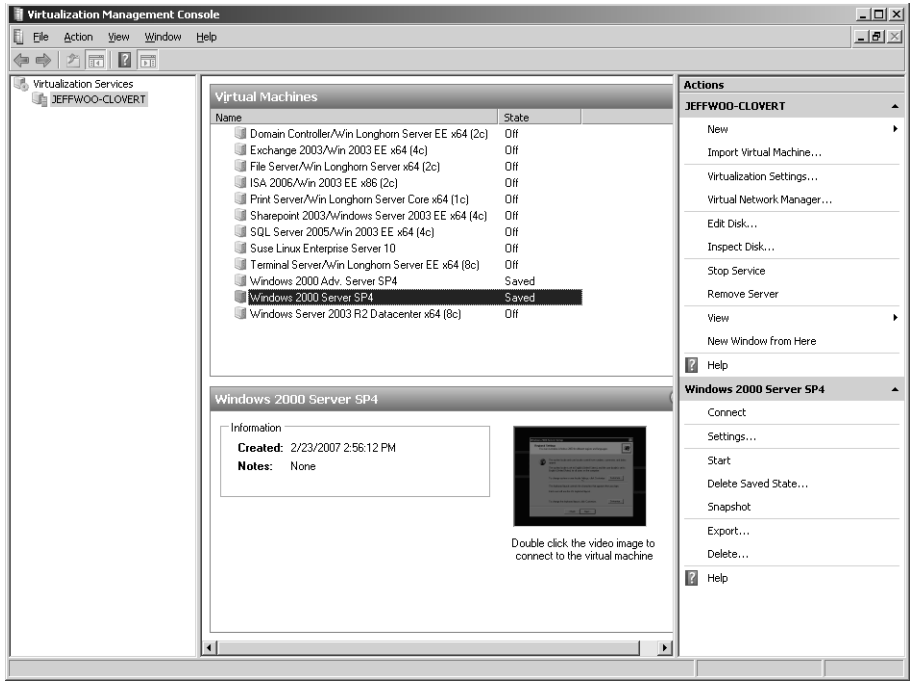


Figure 3-7 Windows Virtualization Management console

So that's a very brief preview of what's in store for virtualization in Windows Server 2008 in terms of managing virtual machines. Fortunately we also have some experts on the product team at Microsoft who provide us with some more information concerning this feature and especially the planning issues surrounding implementing Windows Server virtualization in your environment.

First, here's one of our experts talking about using Windows Server virtualization in conjunction with the Windows server core installation option of Windows Server 2008:

From the Experts: Windows Server Virtualization and a Windows Server Core Installation

The Windows server core installation option of Windows Server 2008 and Windows Server virtualization are two new features of Windows Server 2008 that go hand in hand. The Windows server core installation option is a new minimal GUI shell-less installation option for Window Server 2008 Standard, Enterprise and Datacenter Editions that reduces the management and maintenance required by an administrator. The Windows server core installation option provides key advantages over a full installation of Windows Server 2008 and is the perfect complement to Windows Server virtualization. Here are a couple of reasons why.

- **Reduced attack surface** A Windows server core installation provides a greatly reduced attack surface because it is tailored to provide only what a role requires. By

providing a minimal parent partition, this reduces the need to patch the parent partition. In the past with one workload running per server, if you needed to reboot the server for a patch, it wasn't ideal, but generally one workload was affected. With Windows Server virtualization, you're not just running a single workload. You could be running dozens (even hundreds) of workloads in their own virtual machine. If the virtualization server requires a reboot for a patch (and you don't have a high availability solution in place), the result could be significant downtime.

- **Reduced resource consumption** With the parent partition requiring only a fraction of the memory resources for a Windows server core installation as opposed to a full installation of Windows Server 2008, you can use that memory to run more virtual machines.

In short, it is *highly recommended* that you use Windows Server virtualization in conjunction with a Windows server core installation.

–Jeffrey Woolsey
Lead Program Manager, Windows Virtualization

Next, let's hear another of our experts on the virtualization team at Microsoft share about how to identify what should be virtualized in your environment and what maybe shouldn't:

From the Experts: Virtualization Sizing

It is very important to understand how to roll out virtualization in your organization and what makes the most sense for your environment and business conditions. So often, some enthusiastic users and organizations start either attempting to virtualize everything or start with their most complex middleware environments. There are no right or wrong first candidates for virtualization but you need to ensure that you have fully thought about the impact of using virtualization in your environment and for the workloads in question.

As you think about what to virtualize and how to go about picking the right workloads, the order of deployment, and what hardware capabilities you need, find a model or a set of models that help you conceptualize the end solution. The System Center family of products provides you a set of tools that help simplify some of these issues, and other solutions from vendors like HP provide you tools to help size the deployment environment once you have figured out the candidates and the rollout process.

The next few paragraphs help identify some of the best practices in sizing your virtualization environment. Think of the following as a set of steps that will help you identify what workloads to virtualize and what the deployment schedule should look like.

1. **Assessment** As with any project, the first step is to fully know about where you are today and what capabilities you already have in your environment. The last thing you want to do is to sit and re-create the wheel and invest in things you already have in your environment. As you think about assessment, think about assessing all the components you have in your infrastructure, the types of workloads, and interdependencies of the various workloads. Also evaluate all the management assets you already have in your infrastructure and identify the functions that these are performing, such as monitoring, deployment, data protection, security, and so on. These are the easier items to assess, but the more critical one to assess will be the overall process discipline that exists in your organization and how you deal with change in today's world. While this is a hard factor to quantify, this is critical in evaluating what capacity you have to deploy virtualization. To help you make this assessment from a holistic perspective, there are tools available such as Microsoft's Infrastructure Optimization Model or Gartner's IT Maturity Model that you can choose to use. There is one thing a customer once told me that I will never forget–"If someone tells you they have a solution for your problems when you have not identified or told them what your problems are, most likely they are giving you something you already have in a different package–that is, if you are lucky."
2. **Solution Target** Once you have identified and assessed your current environment, find out where you can use virtualization today. All server virtualization solutions today provide these usage scenarios:
   - Production Server Consolidation, which encompasses all forms of consolidation of systems in existing or new environments.
   - Test and Development Environments, which addresses the use of virtualization for optimizing the test and dev cycles and not only enables you to leverage the cost saving from hardware needs but also enables easy creation and modification of the environments.
   - Business Continuance, where your primary motivator is to leverage the fact that virtualization transforms your IT infrastructure to files (in Microsoft's case a VHD file) to enable new and interesting continuance and disaster recovery solutions.

- Dynamic Datacenter, which is a new set of capabilities unleashed by virtualization to now enable you to not only create and manage your environment more efficiently, but provide a new level of capability to be able to dynamically modify the characteristics of the environments for workloads based on usage. The dynamic resource manipulation enables you to take the consolidation benefits and translate it to now making your IT a more agile environment.
- Branch Office, which while not being a core solution, is one usage scenario where virtualization helps change how IT systems are deployed, monitored, and managed and helps extend the capabilities of the branch environment to bring in legacy and new application environments under one common infrastructure umbrella.

As you are trying to decide which solution area or areas to target for your virtualization solution, do keep in mind the level of complexity of the solutions and the need for increasing levels of management tools and process discipline. Test and dev environments are the easiest to virtualize and usually can manage to take some downtime in case of hiccups–hence this is a natural start for everyone. Server Consolidation is another area that you can start using virtualization in today. The initial cost savings here are in the hardware consolidation benefits–but the true value of consolidation is seen only when you have figured out how to use a unified management infrastructure. Business continuance and branch scenarios need you to have a management infrastructure in place to help orchestrate these solutions and again to see the true value – you will need to have a certain level of processes outlined. Dynamic datacenter is a complex solution for most customers to fully deploy and this usually applies to a certain subset of the org's infrastructure–select the workloads that need this type of solution more carefully as adding the SLAs to maintain such a solution should mean that the workload is really critical to the organization.

3. **Consolidation Candidates** Most users today are deploying virtualization to help consolidate workloads and bring in legacy systems into a unified management umbrella. In this light, it becomes important to identify which workloads are the most logical ones to consolidate today and what makes sense in the future. There are some workloads that sound attractive for virtualization, but might not be ideal at any stretch because of certain I/O characteristics or purely because they are so big and critical that they easily scale up to or beyond the capabilities of the hardware being thrown at them. Operations Manager or Virtual Machine Manager has a report that is generated called the virtualization candidates report that helps scan your entire IT org and tell you exactly what workloads are ideal for virtualization based on a number of thresholds such as CPU utilization, I/O intensity, network usage, size of the workload, and so on. Based on this report and knowing the

interdependencies identified during the assessment phase, you can make intelligent decisions on what workloads to virtualization and when.

4. **Infrastructure Planning** This is where the rubber meets the road so to speak. Once you have identified the candidates to virtualize, you need a place to host the virtualized workloads. Tools from companies such as HP (HP Virtualization Sizing Guide) help you identify the type of servers you will need in your environment to host the virtualization solution that you have identified in the previous step. There is one fundamental rule to consider as you are selecting the infrastructure for virtualization–the two biggest limiting factors for virtualization are memory and I/O throughput–so always ensure that you select a x64 platform for your hardware to ensure a large memory access, and always try to get the best disk subsystem either into the system for DAS or good SAN devices.
5. **Placement** This is not so much an area that is going to affect the sizing of your environment, but has the potential to impact your sizing decisions in the long run. Here we are referring to the act of taking one of the virtualization candidates and actually deploying it to one of the selected virtualization host systems. The knowledge of interdependencies of the various workloads affects some of how this placement occurs but from a high level, this is more about optimizing the placement for a few selected variables. Virtual Machine Manager has an intelligent placement tool that helps you optimize either to a load balancing algorithm or to a maximizing utilization algorithm. You can alternatively also tweak individual parameters to help optimize your environment based on your business weights of the different parameters.

As you size your virtualization environment, also keep in mind the overall manageability factor and how you can scale your management apps to help cover the new environment. Now that you have seen how to size your virtualization environments, keep two things in mind–virtualization is a great technology that can help in multiple levels and scenarios but is still not the panacea for all problems so do take the time to identify your true problems and also remember that you need to look at deploying and managing virtualized environments over a long period of time and hence the need to think about virtualization as a 3-year solution at least.

Virtualization is primarily a consolidation technology that abstracts resources and aids aggregation of workloads, so think carefully about how this affects your environment and what steps you need to have in place to avoid disasters and plan for them early.

–Rajiv Arunkundram
Senior Product Manager, Server Virtualization

Finally, an important planning item for any software deployment is licensing. Here's one of our experts explaining the current licensing plan for Windows virtualization:

From the Experts: Virtualization Licensing

One of the most talked about and often most confused areas for virtualization is licensing. Some of this is primarily caused due to the lack of one industry standard way of dealing with licensing and the other cause is that virtualization is a disruptive technology in how companies operate and hence not clear to customers on what the various policies mean in this new world.

Microsoft's licensing goals are to provide customers and partners cost-effective, flexible, and simplified licensing for our products that will be applicable across all server virtualization products, regardless of vendor. To this effect, several changes were put in place in late 2005 to help accelerate virtualization deployments across vendors:

- Windows server licensing was changed from installation-based licensing to instance-based licensing for server products.
- Microsoft changed licensing to allow customers to run up to 1 physical and 4 virtual instances with a single license of Windows Server 2003 Enterprise Edition on the licensed device; and 1 physical and unlimited virtual instances with Windows Server 2003 Datacenter Edition on the licensed device.
- With the release of SQL Server 2005 SP2, Microsoft announced expanded virtualization use rights to allow unlimited virtual instances on servers that are fully licensed for SQL Server 2005 Enterprise Edition.

With all these changes, you can now easily acquire and license Windows Server and other technologies in a much more efficient process. Virtualization also adds another level of complexity for licensing with the ability to easily move the images or instances around between machines. This is where licensing from the old era makes it tricky. The simple way to remember and ensure that you are fully licensed is to look at the host systems as the primary license holders with the instances being the deployment front. So if you want to move a workload to a system that has Windows Server Enterprise Edition running and already has 4 instances running, you will need an additional license; if it is lower than 4, you will not need an additional license to make the move happen.

Do note that the licensing policies for these apply across virtualization products in the same manner across all server virtualization platforms.

–Rajiv Arunkundram
Senior Product Manager, Server Virtualization

System Center Virtual Machine Manager 2007

The Virtualization Management Console snap-in that is included with Windows Server virtualization is limited in several ways, and it's mainly intended for managing virtual machines on a few servers at a time. Large enterprises want infrastructure solutions, however, and not just point tools. System Center Virtual Machine Manager fills this gap and will enable you to centralize management of a large enterprise's entire virtual machine infrastructure, rapidly provision new virtual machines as needed, and efficiently manage physical server utilization. Plus it's fully integrated with the Microsoft System Center family of products, so you can leverage your existing skill sets as you migrate your network infrastructure to Windows Server 2008.

System Center Virtual Machine Manager runs as a standalone server application, and it can be used to manage a virtualized datacenter that contains hundreds or even thousands of virtual machines in an Active Directory environment. System Center Virtual Machine Manager will be able to manage virtual machines running on both Microsoft Virtual Server 2005 R2 and Windows 2008 Server with Windows Server virtualization installed. You can even deploy System Center Virtual Machine Manager in a fiber-channel SAN environment for performing tasks such as the following:

- Deploying VMs from your SAN library to a host
- Transferring VMs from a host to your library
- Migrating VMs from one host to another host

The administrator console for System Center Virtual Machine Manager is built upon Windows PowerShell, and you can use it to add and manage host machines, create and manage virtual machines, monitor tasks, and even migrate physical machines to virtual ones (something called P2V).

System Center Virtual Machine Manager also includes a self-service Web portal that enables users to independently create and manage their own virtual machines. The way this works is that the administrator predetermines who can create virtual machines, which hosts these machines can run on, and which actions users can perform on their virtual machines.

At the time of this writing, System Center Virtual Machine Manager is in Beta 1 and supports managing only virtual machines hosted on Virtual Server 2005 R2.

SoftGrid Application Virtualization

Finally, another upcoming virtualization technology you should know about is SoftGrid Application Virtualization, which Microsoft took ownership of when it acquired Softricity in July 2006. SoftGrid provides a different kind of virtualization than we've been discussing here—instead of virtualizing an entire operating system, it virtualizes only an application. This functionality makes SoftGrid a more fine-grained virtualization technology than Windows

Server virtualization. Also, it's designed not for the server end but for deploying applications to desktops easily and updating them as necessary.

Essentially, what SoftGrid can do using its streaming delivery mechanism is to transform any Windows program into a dynamic service that then follows users wherever they might go. These services can then be integrated into Microsoft's management infrastructure so that they can be configured and managed using standard policy-based methods. At this point, SoftGrid isn't directly associated with Windows 2008 Server or Windows Server virtualization, but it's a new Microsoft technology you should be aware of as the virtualization landscape continues to evolve.

Conclusion

It would have been nice to have looked in greater depth at how Windows Server virtualization in Windows Server 2008 works. Unfortunately, at the time of this writing the bits aren't there yet. Still, you have to admit that this is one of the hottest features of Windows Server 2008, both from the perspective of the day-to-day needs of IT professionals and as a prime selling point for Windows Server 2008. I've tried to give you a taste of how this new technology will work and a glimpse of what it looks like, but I hope you're not satisfied with that—I'm not. I can't wait till all this comes together, and the plain truth of the matter is that in only a few years virtualization will be inexpensive and ubiquitous. So get ready for it now.

Bring back the mainframe!!

Additional Reading

If you want to find out more about the underlying processor enhancements from Intel and AMD that will support and be required by Windows Server virtualization, check out the following sources:

- See *http://www.intel.com/technology/virtualization/index.htm* for information concerning Intel VT technology
- See *http://www.amd.com/us-en/Processors/ProductInformation/0,,30_118_8826_14287,00.html* for information about AMD-V technology

For information on how Microsoft and XenSource are collaborating to support running Linux on Windows Server 2008, read the following article on Microsoft PressPass: *http://www.microsoft.com/presspass/press/2006/jul06/07-17MSXenSourcePR.mspx*.

The starting point for finding out more about current (and future) Microsoft virtualization products is *http://www.microsoft.com/windowsserversystem/virtualserver/default.mspx* on Microsoft.com.

For more information about System Center Virtual Machine Manager and how you can join the beta program for this product, see *http://www.microsoft.com/windowsserversystem/virtualization/default.mspx* on the Microsoft Web site. From there, you can jump to pages describing Virtual Server 2005 R2, Virtual PC 2007, System Center Virtual Machine Manager, and most likely Windows Server virtualization on Windows Server 2008 in the near future as well.

If you're interested in finding out more about SoftGrid Application Virtualization, see *http://www.softricity.com/index.asp*, although the Softricity Web site will probably be folded soon into Microsoft.com.

Finally, be sure to turn to Chapter 14, "Additional Resources," if you want to find more resources about Windows Server virtualization in Windows Server 2008. In that chapter, you'll find links to webcasts, whitepapers, blogs, newsgroups, and other sources of information on this feature and other Microsoft virtualization technologies.

Chapter 4

Managing Windows Server 2008

In this chapter:

I was kidding, of course, when I said we should bring back the mainframe. After all, remember how much fun it was managing those machines? Sitting at a green screen all day long, dropping armfuls of punch cards into the hopper...what fun! At least running an IBM System/360 could be more fun than operating a PDP-11. When I was a university student years ago (decades actually), I worked one summer for the physics department, where there was a PDP-11 in the sub-sub-basement where the Cyclotron was located. I remember sitting there alone one night around 3 a.m. while an experiment was running, watching the lights blink on the PDP and flipping a switch from time to time to read a paper tape. And that was my introduction to the tools used for managing state-of-the-art computers in those days—specifically, lights, switches, and paper tape.

Computers have come a long way since then. Besides being a lot more powerful, they're also a lot easier to manage. So before we examine other new and exciting features of Microsoft Windows Server 2008, let's look at the new and enhanced tools you can use to manage the platform. These tools range from user interface (UI) tools for configuring and managing servers to a new command-line tool for installing roles and features, tools for remote administration, Windows Management Instrumentation (WMI) enhancements for improved scripted management, Group Policy enhancements, and more.

Performing Initial Configuration Tasks

The first thing you'll notice when you install Windows Server 2008 is the Initial Configuration Tasks screen (shown in Figure 4-1).

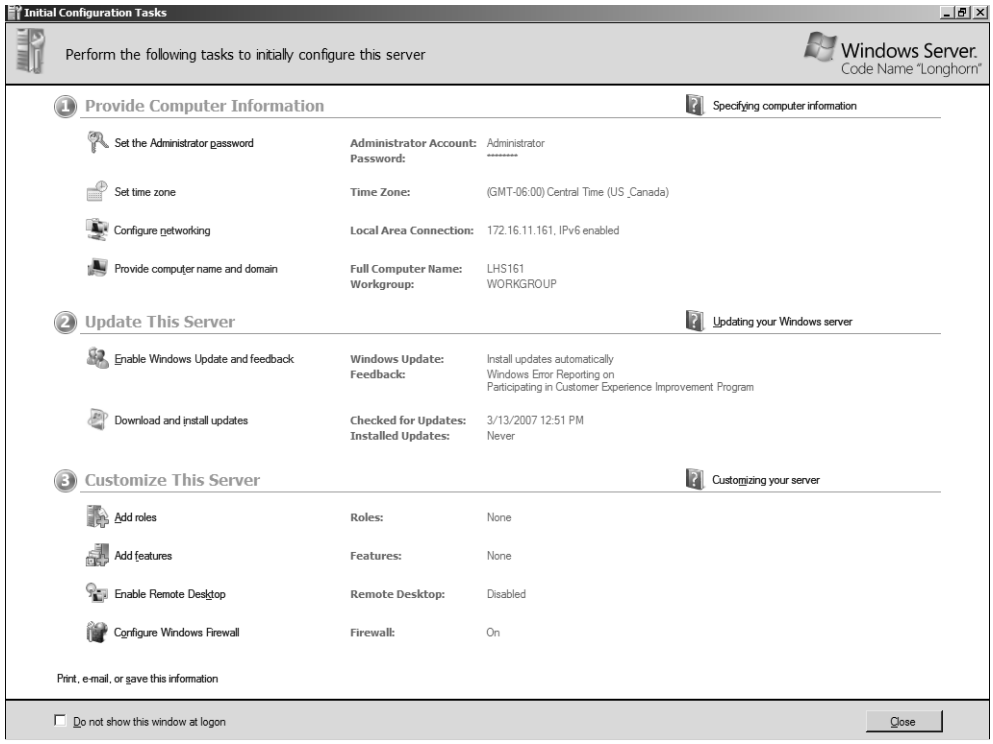


Figure 4-1 The Initial Configuration Tasks screen

Remember for a moment how you perform your initial configuration of a machine running Windows Server 2003 Service Pack 1 or later, where you do this in three stages:

1. During Setup, when you specify your administrator password, network settings, domain membership, and so on
2. Immediately after Setup, when a screen appears asking if you want to download the latest updates from Windows Update and turn on Automatic Updates before the server can receive inbound traffic
3. After you've allowed inbound traffic to your server, when you can use Manage Your Server to install roles on your server to make it a print server, file server, domain controller, and so on

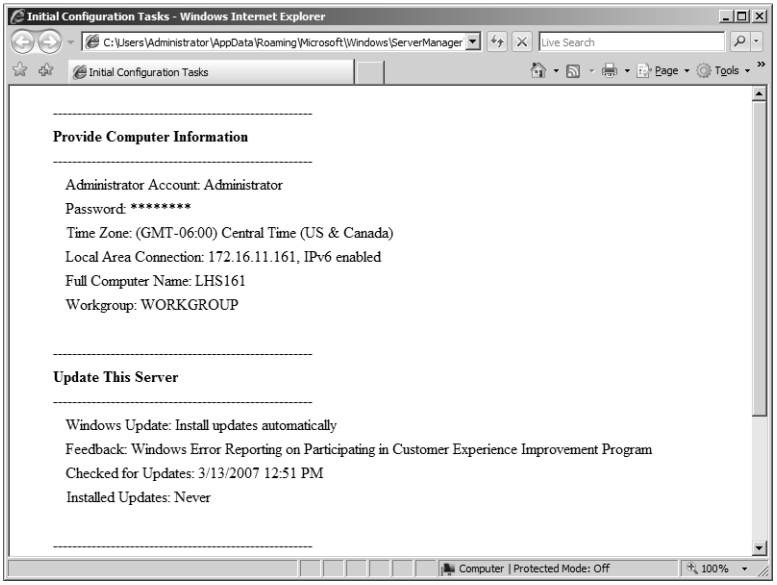
Windows Server 2008, however, consolidates these various server configuration tasks by consolidating during- and post-Setup tasks together and presenting them to you in a single screen called Initial Configuration Tasks (ICT). Using the ICT you can

- Specify key information, including the administrator password, time zone, network settings, and server name. You can also join your server to a domain. For example, clicking the Provide Computer Name And Domain link opens System Properties with the Computer Named tab selected.

- Search Windows Update for available software updates, and enable one or more of the following: Automatic Updates, Windows Error Reporting (WER), and participation in the Customer Experience Improvement Program.
- Configure Windows Firewall on your machine, and enable Remote Desktop so that the server can be remotely managed using Terminal Services.
- Add roles and features to your server—for example, to make it a DNS server or domain controller.

In addition to providing a user interface where you can perform these tasks, ICT also displays status information for each task. For example, if a task has already been performed, the link for the task changes color from blue to purple just like an ordinary hyperlink. And if WER has been turned on, the message "Windows Error Reporting on" is displayed next to the corresponding task item.

Once you've performed the initial configuration of your server, you can click the Print, E-mail Or Save This Information link at the bottom. This opens Internet Explorer and displays a results page showing the settings you've configured.



This results page can be found at %systemdrive%\users\<username>\AppData\Roaming\Microsoft\Windows\ServerManager\InitialConfigurationTasks.html, and it can be saved or e-mailed for reporting purposes.

A few more notes concerning Initial Configuration Tasks:

- Performing some tasks requires that you log off or reboot your machine. For example, by default when you install Windows Server 2008, the built-in Administrator account is enabled and has no password. If you use ICT to change the name of this account or specify a password, you must log off and then on again for this change to take effect.
- If Windows Server 2008 detects that it is deployed on a restricted network (that is, quarantined by NAP) when you first log on, the Update This Server section of the ICT displays a new link named Restore Network Access. Clicking this link allows you to review current network access restrictions and restore full network access for your server, and until you do this your server is in quarantine and has only limited network access. The reason that the other two items in this section (Enable Windows Update And Feedback and Download And Install Updates) are not displayed in this situation is that machines in quarantine cannot access Windows Update directly and must receive their updates from a remediation server. For more information about this, see Chapter 10, "Network Access Protection."
- OEMs can customize the ICT screen so that it displays an additional section at the bottom that can include an OEM logo, a description, and task links that can launch EXEs, DLLs, and scripts provided by the OEM. Note that OEM task links cannot display status information, however.
- The ICT is not displayed if you upgrade to Windows Server 2008 from a previous version of Windows Server.
- The ICT is also not displayed if the following Group Policy setting is configured:

  Computer Configuration\Administrative Templates\System\Server Manager\Do Not Open Initial Configuration Tasks Windows At Logon

Using Server Manager

OK, you've installed your server, performed the initial configuration tasks, and maybe installed a role or two—such as file server and DHCP server—on your machine as well. Now what? Once you close ICT, another new tool automatically opens—namely, Server Manager (shown in Figure 4-2). I like to think of Server Manager as "Computer Management on steroids," as it can do everything compmgmt.msc can do plus a whole lot more. (Look at the console tree on the left in this figure and you'll see why I said this.)

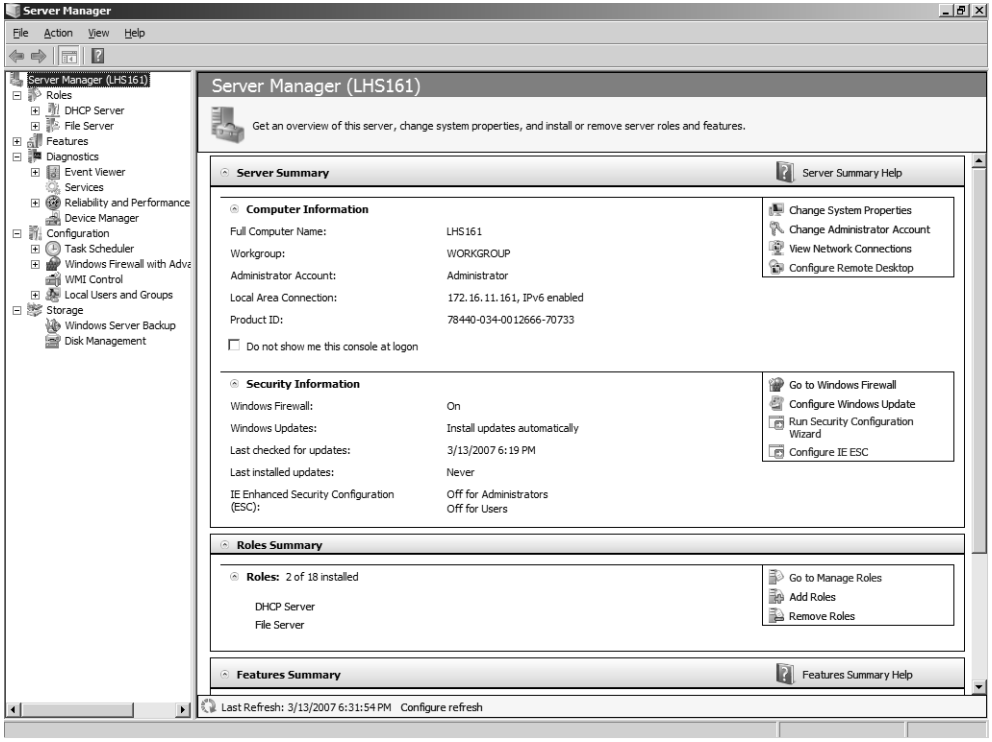


Figure 4-2 Main page of Server Manager

The goal of Server Manager is to provide a straightforward way of installing roles and features on your server so that it can function within your business networking environment. As a tool, Server Manager is primarily targeted toward the IT generalist who works at medium-sized organizations. IT specialists who work at large enterprises might want to use additional tools to configure their newly installed servers, however—for example, by performing some initial configuration tasks during unattended setup by using Windows Deployment Services (WDS) together with unattend.xml answer files. See Chapter 13, "Deploying Windows Server 2008," for more information on using WDS to deploy Windows Server 2008.

Server Manager also enables you to modify any of the settings you specified previously using the Initial Configuration Tasks screen. For example, in Figure 4-2 you can see that you can enable Remote Desktop by clicking the Configure Remote Desktop link found on the right side of the Server Summary tile. In fact, Server Manager lets you configure additional advanced settings that are not exposed in the ICT screen, such as enabling or disabling the Internet Explorer Enhanced Security Configuration (IE ESC) or running the Security Configuration Wizard (SCW) on your machine.

Managing Server Roles

Let's dig a bit deeper into Server Manager. Near the bottom of Figure 4-2, you can see that we've already installed two roles on our server using the ICT screen. We'll learn more about the various roles, role services, and features you can install on Windows Server 2008 later in Chapter 5, "Managing Server Roles." For now, let's see what we can do with these two roles that have already been installed.

Clicking the Go To Manage Roles link changes the focus from the root node (Server Manager) to the Roles node beneath it. (See Figure 4-3.) This page displays a list of roles installed on the server and the status of each of these roles, including any role services that were installed together with them. (Role services will be explained later in Chapter 5.)

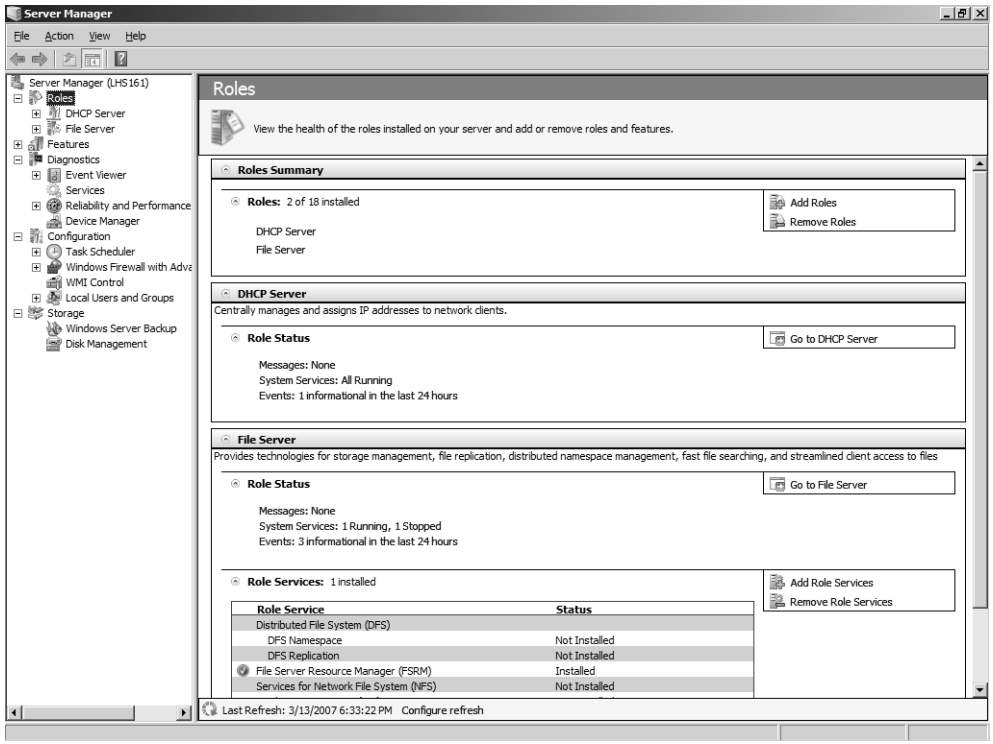
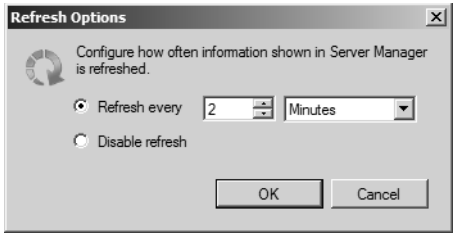


Figure 4-3 Roles page of Server Manager

The status of this page is updated in real time at periodic intervals, and if you look carefully at these figures you'll see a link at the bottom of each page that says "Configure refresh." If you click this link, you can specify how often Server Manager refreshes the currently displayed page. By default, the refresh interval is two minutes.



Selecting the node for the File Server role in the console tree (or clicking the Go To File Server link on the Roles page) displays more information about how this role is configured on the machine (as shown in Figure 4-4). Using this page, you can manage the following aspects of your file server:

- View events relevant to this role (by double-clicking on an event to display its details).
- View system services for this role, and stop, start, pause, or resume these services.
- View role services installed for this role, and add or remove role services.
- Get help on how to perform role-related tasks.

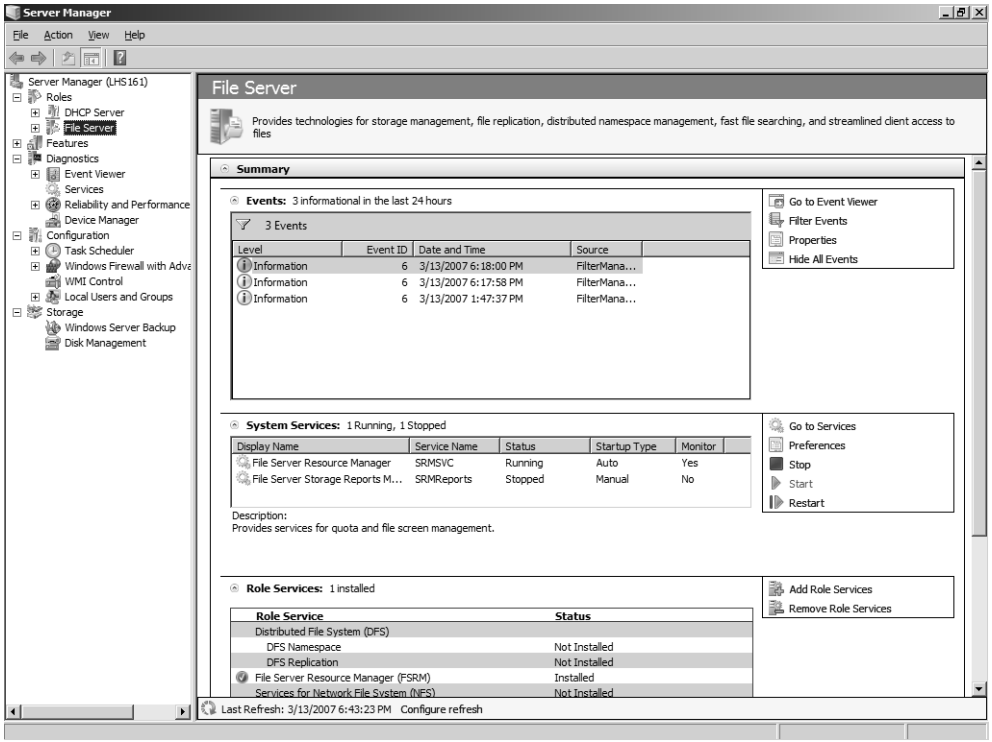


Figure 4-4 Main page for File Server role

Note the check mark in the green circle beside File Server Resource Manager (FSRM) under Role Services. This means that FSRM, an optional component or "role service" for the File

Server role, has been installed on this server. You probably remember FSRM from Windows Server 2003 R2—it's a terrific tool for managing file servers and can be used to configure volume and folder quotas, file screens, and reporting. But in Windows Server 2003 R2, you had to launch FSRM as a separate administrative tool—not so in Windows Server 2008. What's cool about Server Manager is that it is implemented as a managed, user-mode MMC 3.0 snap-in that can host other MMC snap-ins and dynamically show or hide them inline based on whether a particular role or feature has been installed on the server.

What this means here is that we can expand our File Server node, and underneath it you'll find two other snap-ins—namely, File Server Resource Manager (which we chose to install as an additional role service when we installed the File Server role on our machine) and Shared Folders (which is installed by default whenever you add the file server role to a machine.) And underneath the FSRM node, you'll find the same subnodes you should already be familiar with in FSRM on Windows Server 2003 R2. (See Figure 4-5.) And anything you can do with FSRM in R2, you do pretty much the same way in Windows Server 2008. For example, to configure an SMTP server for sending notification e-mails when quotas are exceeded, right-click on the File Server Resource Manager node and select Properties. (In addition to hosting the FSRM snap-in within Server Manager, adding the FSRM role service also adds the FSRM console to Administrative Tools.)

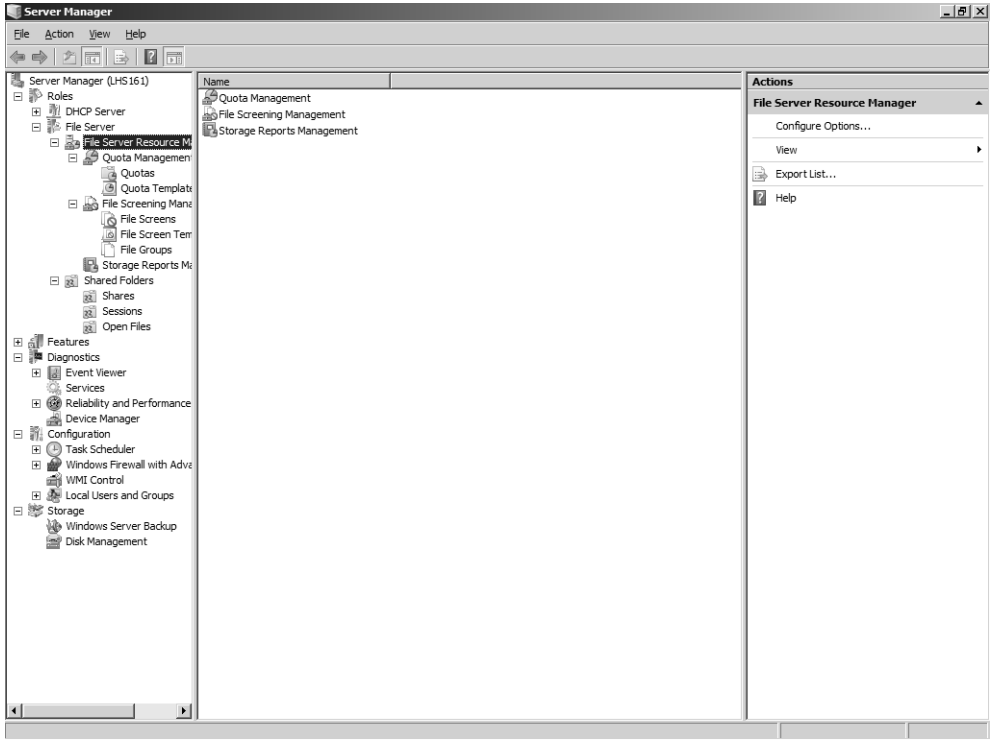


Figure 4-5 File Server role showing hosted snap-ins for File Server Resource Manager and Shared Folders

Here are a few more important things to know about Server Manager. First, Server Manager is designed to be a single, all-in-one tool for managing your server. In that light, it replaces both Manage Your Server (for adding roles) and the Add/Remove Windows Components portion of Add Or Remove Programs found on previous versions of Windows Server. In fact, if you go to Control Panel and open Programs And Features (which replaced Add Or Remove Programs in Windows Vista), you'll see a link called Turn Windows Features On And Off. If you click that link, Server Manager opens and you can use the Roles or Features node to add or remove roles, role services, and features. (See Chapter 5 for how this is done.)

Also, when Server Manager is used to install a role such as File Server on your server, it makes sure that this role is *secure by default*. (That is, the only components that are installed and ports that are opened are those that are absolutely necessary for that role to function.) In Windows Server 2003 Service Pack 1 or later, you needed to run the Security Configuration Wizard (SCW) to ensure a server role was installed securely. Windows Server 2008 still includes the SCW, but the tool is intended for use by IT specialists working in large enterprises. For medium-sized organizations, however, IT generalists can use Server Manager to install roles securely, and it's much easier to do than using SCW. In addition, while Server Manager can be used for installing new roles using *smart defaults*, SCW is mainly designed as a post-deployment tool for creating security policies that can then be applied to multiple servers to harden them by reducing their attack surface. (You can also compare policies created by SCW against the current state of a server for auditing reasons to ensure compliance with your corporate security policy.) Finally, while Server Manager can only be used to add the default Windows roles (or out-of-band roles made available later, as mentioned in the extensibility discussion a bit later), SCW can also be used for securing nondefault roles such as Exchange Server and SQL Server. But the main takeaway for this chapter concerning Server Manager vs. SCW is that when you run Server Manager to install a new role on your server, you don't need to run SCW afterward to lock down the role, as Server Manager ensures the role is already secure by default.

Server Manager relies upon something called Component Based Servicing (CBS) to discover what roles and services are installed on a machine and to install additional roles or services or remove them. For those of you who might be interested in how this works, there's a sidebar in the next section that discusses it in more detail. Server Manager is also designed to be extensible. This means when new features become available (such as Windows Server Virtualization, which we talked about in Chapter 3, "Windows Server Virtualization"), you'll be able to use Server Manager to download these roles from Microsoft and install them on your server.

Server Manager is designed to manage one server only (the local server) and cannot be used to manage multiple servers at once. If you need a tool to manage multiple servers simultaneously, use Microsoft System Center. You can find out more about System Center products and their capabilities at *http://www.microsoft.com/systemcenter/*, and it will be well worth your time to do so. In addition, the status information displayed by Server Manager is limited to

event information and whether role services are running. So if you need more detailed information concerning the status of your servers, again be sure to check out System Center, the next generation of the SMS and MOM platforms.

Unlike using Computer Management, you can't use Server Manager to remotely connect to another server and manage it. For example, if you right-click on the root node in Server Manager, the context menu that is displayed does not display a Connect To A Different Computer option. However, this is not really a significant limitation of the tool because most admins will simply enable Remote Desktop on their servers and use Terminal Services to remotely manage them. For example, you can create a Remote Desktop Connection on a Windows Vista computer, use it to connect to the console session on a Windows Server 2008 machine, and then run Server Manager within the remote console session. And speaking of Computer Management, guess what happens if you click Start, right-click on Computer, and select Manage? In previous versions of Windows, doing this opened Computer Management—what tool do you think opens if you do this in Windows Server 2008?

Finally, a few more quick points you can make note of:

- Server Manager cannot be used to manage servers running previous versions of the Windows Server operating system.
- Server Manager cannot be installed on Windows Vista or previous versions of Microsoft Windows.
- Server Manager is not available on a Windows server core installation of Windows Server 2008 because the supporting components (.NET Framework 2.0 and MMC 3.0) are not available on that platform.
- You can configure the refresh interval for Server Manager and also whether the tool is automatically opened at logon by configuring the following Group Policy settings:

  Computer Configuration\Administrative Templates\System\Server Manager\Do Not Open Server Manager Automatically At Logon

  Computer Configuration\Administrative Templates\System\Server Manager\ Configure The Refresh Interval For Server Manager

From the Experts: The Security Configuration Wizard in Windows Server 2008

The Security Configuration Wizard (SCW) reduces the attack surface of Windows Servers by asking the user a series of questions designed to identify the functional requirements of a server. Functionality not required by the roles the server is performing is then disabled. In addition to being a fundamental security best practice, SCW reduces the number of systems that need to be immediately patched when a vulnerability is exposed. Specifically, SCW:

- Disables unneeded services.
- Creates required firewall rules.
- Removes unneeded firewall rules.
- Allows further address or security restrictions for firewall rules.
- Reduces protocol exposure to server message block (SMB), LanMan, and Lightweight Directory Access Protocol (LDAP).

SCW guides you through the process of creating, editing, applying, or rolling back a security policy based on the selected roles of the server. The security policies that are created with SCW are XML files that, when applied, configure services, Windows Firewall rules, specific registry values, and audit policy. Those security policies can be applied to an individual machine or can be transformed into a group policy object and then linked to an Organizational Unit in Active Directory.

With Windows Server 2008 some important improvements have been made to SCW:

- On Windows Server 2003, SCW was an optional component that had to be manually installed by administrators. SCW is now a default component of Windows Server 2008 which means Administrators won't have to perform extra steps to install or deploy the tool to leverage it.
- Windows Server 2008 will introduce a lot of new and exciting functionality in Windows Firewall. To support that functionality, SCW has been improved to store, process, and apply firewall rules with the same degree of precision that the Windows Firewall does. This was an important requirement since on Windows Server 2008 the Windows Firewall will be on by default.
- The SCW leverages a large XML database that consists of every service, firewall rule and administration option from every feature or component available on Windows Server 2008. This database has been totally reviewed and updated for Windows Server 2008. Existing roles have been updated, new roles have been added to the database, and all firewall rules have been updated to support the new Windows Firewall.

- SCW now validates all XML files in its database files using a set of XSD files that contains the SCW XML schema. This will help administrators or developers extend the SCW database by creating new SCW roles base on their own requirements or applications. Those XSD files are available under the SCW directory.
- All SCW reports have been updated to reflect the changes made to the SCW schema regarding support for the new Window Firewall. Those reports include the Configuration Database report, the Security Policy report and the Analysis report that will compare the current configuration of Windows Server 2008 against an SCW security policy.

SCW provides an end to end solution to reduce the attack surface of Windows Server 2008 machines by providing a possible configuration of default components, roles, features, and any third-party applications that provide an SCW role.

SCW is not responsible for installing or removing any roles, features, or third-party applications from Windows Server 2008. Instead, Administrators should use Server Manager if they need to install roles and features, or use the setup provided with any third party application. The installation of roles and features via Server Manager is made based on security best practices.

While SCW complements well Server Manager, its main value is in the configuration of the core operating system and third-party applications that provide an SCW role. SCW should be used every time the configuration of a default component on Windows Server 2008 needs to be modified or when a third-party application is added or removed. In some specific scenarios, like for remote administration, running SCW after using Server Manager might provide some added value to some specific roles or features. Using SCW after modifying a role or feature through Server Manager is not a requirement, however.

–*Nils Dussart*
Program Manager for the Security Configuration Wizard (SCW), Windows Core Operating System Division

ServerManagerCmd.exe

In addition to the Server Manager user interface, there is also a command-line version of Server Manager called ServerManagerCmd.exe that was first introduced in the IDS_2 build of Windows Server 2008 (that is, the February CTP build). This command-line tool, which is found in the %windir%\system32 folder, can be used to perform the following tasks:

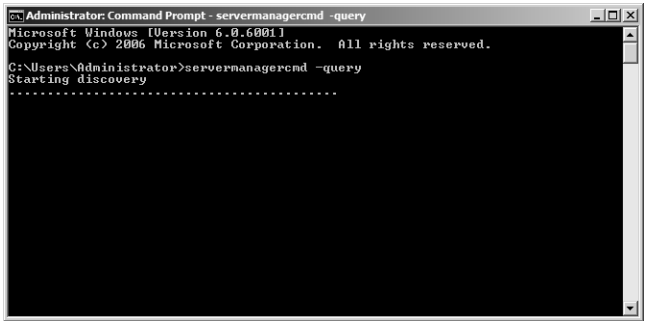
- Display a list of roles and features already installed on a machine.
- Display a list of role services and features that would be installed *if* you chose to install a given role.
- Add a role or feature to your server using the default settings of that role or feature.

- Add several roles/features at once by providing an XML answer file listing the roles/features to be installed.
- Remote roles or features from your server.

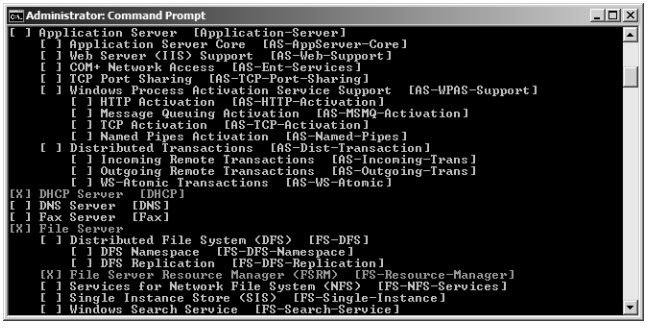
What ServerManagerCmd.exe *can't* do includes the following:

- Install a role or feature, and change its default settings.
- Reconfigure a role or feature already installed on the machine.
- Connect to a remote machine, and manage roles/features on that machine.
- Manage roles/features on machines running a Windows server core installation of Windows Server 2008.
- Manage non-OOB roles/features—such as Exchange Server or SQL Server.

Let's take a look at the **servermanagercmd –query** command, which displays the list of roles and features currently available on the computer, along with their command-line names (values that should be used to install or remove the role or feature from the command line). When you run this command, something called *discovery* runs to determine the different roles and features already installed.



After discovery completes (which may take a short period of time), the command generates output displaying installed roles/features in green and marked with "X".



You can also type **servermanagercmd –query results.xml** to send the output of this command to an XML file. This is handy if you want to save and programmatically parse the output of this command.

Let's now learn more about ServerManagerCmd.exe from one of our experts at Microsoft:

From the Experts: Automating Common Deployment Tasks with ServerManagerCmd.exe

Rolling out a new internal application or service within an organization frequently means setting up roles and features on multiple servers. Some of these servers might need to be set up with exactly the same configuration, and others might reside in remote locations that are not readily accessible by full-time IT staff. For these reasons, you might want to write scripts to automate the deployment process from the command line.

One of the tools that can facilitate server deployment from the command line is ServerManagerCmd.exe. This tool is the command-line counterpart to the graphical Server Manager console, which is used to install and configure server roles and features. The graphical and command-line versions of Server Manager are built on the same synchronization platform that determines what roles and features are installed and applies user-specified configurations to the server.

ServerManagerCmd.exe provides a set of command-line switches that enable you to automate many common deployment tasks as follows:

View the List of Installable Roles and Features

You can use the –query command to see a list of roles and features available for installation and find out what's currently installed. You can also use –query to look up the command-line names of roles and features. These are listed in square brackets [] after the display name.

Install and Uninstall Roles and Features

You can use the –install and –remove commands to install and uninstall roles and features. One issue to be aware of is that ServerManagerCmd.exe enables you only to install and uninstall. Apart from a few notable exceptions for required settings, you cannot specify configuration settings as you can with the graphical Server Manager console. You need to use other role-specific tools, such as MMC snap-ins and command-line utilities, to specify configuration settings after installing roles and features using ServerManagerCmd.exe.

Run in "What-If" Mode

After you create a script to set up the server with ServerManagerCmd.exe, you might want to check that the script will perform as expected. Or you might want to see what will happen if you type a specific command with ServerManagerCmd.exe. For these scenarios, you can supply the –whatif switch. This switch tells you exactly what would be

installed and removed by a command or answer file, based on the current server configuration, without performing the actual operations.

Specify Input Parameters via an Answer File

ServerManagerCmd.exe can operate in an interactive mode, or it can be automated using an answer file. The answer file is specified using the –inputPath <answer.xml> switch, where <answer.xml> is the name of an XML file with the list of input parameters. The schema for creating answer files can be found in the ServerManagerCmd.exe documentation.

Redirect Output to a Results File

It is usually a good practice to keep a history of configuration changes to your servers in case you need to troubleshoot a problem, migrate the settings of an existing server to a new server, or recover from a disaster or failure. To assist with record keeping, you can use the resultPath <results.xml> switch to save the results of an installation or removal to a file, where <results.xml> is the name of the file where you want the output to be saved.

–Dan Harman
Program Manager, Windows Server, Windows Enterprise Management Division

You'll learn more about using ServerManagerCmd.exe for adding roles and features in Chapter 5, but for now let's move on and look at more tools for managing Windows Server 2008.

Remote Server Administration Tools

What if you want to manage our file server running Windows Server 2008 remotely from another machine? We already saw one way you could do this—enable Remote Desktop on the file server, and use Terminal Services to run our management tools remotely on the server. Once we have a Remote Desktop Connection session with the remote server, we can run tools such as Server Manager or File Server Resource Manager as if we were sitting at the remote machine's console.

In Windows Server 2003, you can also manage remote servers this way. But you can also manage them another way by installing the Windows Server 2003 Administration Tools Pack (Adminpak.msi) on a different Windows Server 2003 machine, or even on an admin workstation running Windows XP Service Pack 2. And once the Tools Pack is installed, you can open any of these tools, connect to your remote server, and manage roles and features on the server (provided the roles and features are installed).

Is there an Adminpak for Windows Server 2008? Well, there's an equivalent called the Remote Server Administration Tools (RSAT), which you can use to install selected management tools on your server even when the binaries for the roles/features those tools will manage are not

installed on your server. In fact, the RSAT does Adminpak one better because Adminpak installs all the administrative tools, whereas the RSAT lets you install only those tools you need. (Actually, you can just install one tool from Adminpak if you want to, though it takes a bit of work to do this—see article 314978 in the Microsoft Knowledge Base for details.)

What features or roles can you manage using the RSAT? As of Beta 3, you can install management tools for the following roles and features using the RSAT:

- Roles
  - Active Directory Domain Services
  - Active Directory Certificate Services
  - Active Directory Lightweight Directory Services
  - Active Directory Rights Management Services
  - DNS Server
  - Fax Server
  - File Server
  - Network Policy and Access Services
  - Print Services
  - Terminal Services
  - Web Server (IIS)
  - Windows Deployment Services
- Features:
  - BitLocker Drive Encryption
  - BITS Server Extensions
  - Failover Clustering
  - Network Load Balancing
  - Simple SAN Management
  - SMTP Server
  - Windows System Resource Management (WSRM)
  - WINS Server

How do you install individual management tools using the RSAT? With Windows Server 2008, it's easy—just start the Add Feature Wizard, and select the RSAT management tools you want to install, such as the Terminal Services Gateway management tool. (See Figure 4-6. Note that installing some RSAT management tools might require that you also install additional features. For example, if you choose to install the Web Server (IIS) management tool from the

RSAT, you must also install the Configuration APIs component of the Windows Process Activation Service [WPAS] feature.)

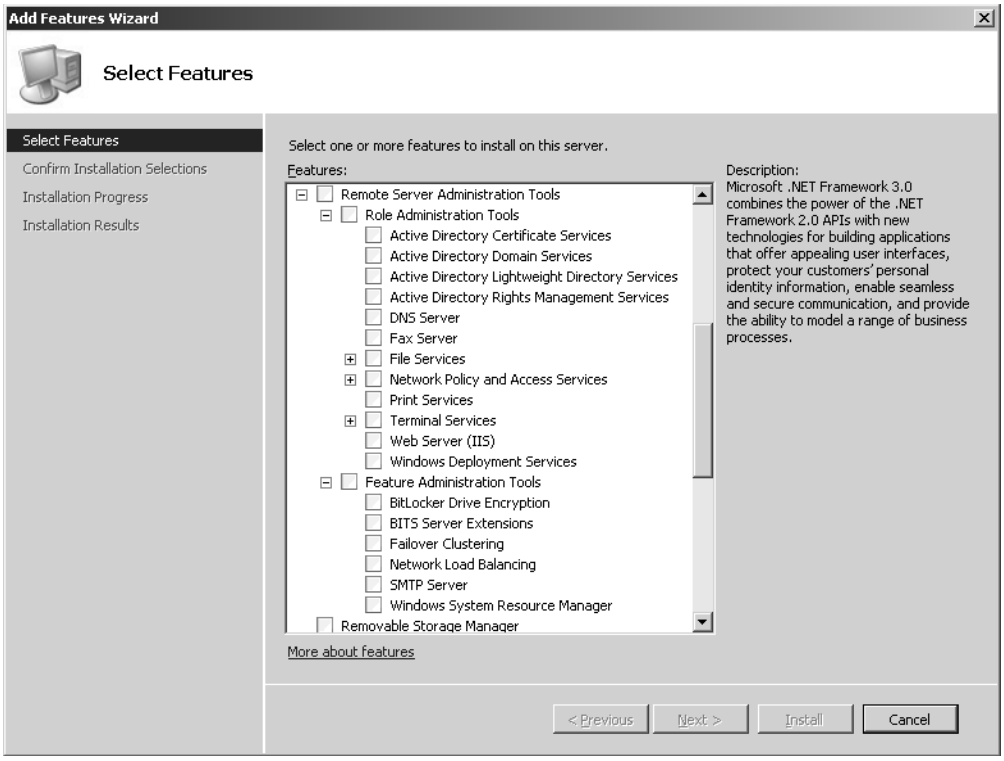


Figure 4-6 Installing a management tool using the RSAT feature

The actual steps for installing features on Windows Server 2008 are explained in Chapter 5. For now, just note that when you install an RSAT subfeature such as TS Gateway, what this does is add a new shortcut under Administrative Tools called TS Gateway. Then if you click Start, then Administrative Tools, then TS Gateway, the TS Gateway Manager console opens. In the console, you can right-click on the root node, select Connect To TS Gateway Server, and manage a remote Windows Server 2008 terminal server with the TS Gateway role service installed on it without having to enable Remote Desktop on the terminal server.

Finally, the Windows Server 2003 Adminpak can be installed on a Windows XP SP2 workstation, which lets you administer your servers from a workstation. Can the RSAT be installed on a Windows Vista machine so that you can manage your Windows Server 2008 machines from there?

As of Beta 3, the answer is "not yet." Plans for how RSAT will be made available for Windows Vista are uncertain at this moment, but it's likely we can expect something that can do this around or shortly after Windows Vista Service Pack 1. We'll just have to wait and see.

Other Management Tools

There are other ways you can manage Windows Server 2008 besides the tools we've discussed so far. Let's examine these now. Specifically, we're going to look at the following items:

- Group Policy
- Windows Management Instrumentation (WMI)
- Windows PowerShell
- Microsoft System Center

Group Policy

Group Policy in Windows Vista and Windows Server 2008 has been enhanced in several ways, including:

- Several new areas of policy management, including configuring Power Management settings, blocking installation of devices, assigning printers based on location, and more.
- A new format for Administrative Templates files called *ADMX* that is XML-based and replaces the proprietary-syntax ADM files used in previous versions of Windows.
- Network Location Awareness to enable Group Policy to better respond to changing network conditions and remove the need for relying on ICMP for policy processing.
- The ability to use local group policy objects, the capability of reducing SYSVOL bloat by placing ADMX files in a central store, and several other new features and enhancements.

A good source of information about Group Policy in Windows Vista (and therefore also in Windows Server 2008, because the platforms were designed to fit together) is Chapter 13, "Managing the Desktop Environment," in the *Windows Vista Resource Kit* from Microsoft Press. Meanwhile, while your assistant is running out to buy a couple of copies of that title (I was lead author for that title and my retirement plans are closely tied to the royalties I earn from sales, so please go buy a dozen or so copies), let's kick back and listen to one of our experts at Microsoft telling us more about post-Vista enhancements to Group Policy found in Windows Server 2008:

From the Experts: What's New in Group Policy in Windows Server 2008

The following is a description of some of the Group Policy enhancements found in Windows Server 2008.

Server Manager Integration

The first noticeable change in Windows Server 2008 is how the Group Policy tools are presented. In past operating systems, other than Windows Vista, an admin would have to go to the Microsoft Web site to download the Group Policy Management Console (GPMC) and install it on every administrative workstation where Group Policy management is performed. In Windows Server 2008, the installation bits are delivered with the operating system. No more downloads, no more wondering where the installation media is—it is just there.

A difference in this new environment is how optional Windows components are installed. Windows Server 2008 introduces a new management console for servers called Server Manager. This is the tool that is used to install server roles, as well as optional Windows components. If you choose to go the old-school route and add Windows components from the Add/Remove Control Panel, it will launch Server Manager.

Not only do you use Server Manager to install the optional components, but the GPMC console itself is hosted within the Server Manager console. This means all of your administrative tools are kept in one place and are easily discoverable. Of course, you will still be able to find the tools in the common locations, such as Administrative Tools.

Search/Filters, Comments, and Starter GPOs

These features really enhance the administrative experience around managing and authoring policy. They are, technically, multiple features, but they work well when described as a "feature set," as they all address the same business problem—difficulty in authoring policy. As you are probably aware, in the Windows Vista/Windows Server 2008 wave of operating systems there are hundreds of new settings to be managed. This means the total number of settings approaches 3000. That is a lot of manageable settings. Even though this provides a ton of value to the IT Professional, it increases the complexity when it comes to actually locating the setting or policy item that you are trying to manage. Microsoft has provided a "settings" spreadsheet that contains all the Group Policy settings in one relatively easy-to-use document, but it really doesn't solve the problem. Microsoft has received feedback from many IT pros that there needs to be a method within the Group Policy tool itself to make finding the right settings easier.

Now with Search and Filters, when you are authoring a policy right in the editor you have a great mechanism to locate the setting you are looking for. You will see a new Filter button in the toolbar, and if you right-click on the Administrative Templates node in the editor you will see a menu item called Filter Options. Filter Options allows you to set the

criteria that you are looking to search on. For example, you can narrow your view to only *configured* items, specific key words, or the system requirements (for example, Internet Explorer 6.0 settings). Filter Options provides a very intuitive interface and has great flexibility to help in locating the settings that you are looking for. Once you set Filter Options and turn on the Filter (global setting), the editor displays only settings that you are targeting. The Group Policy team is really excited to bring these features to you because we know it will reduce some of the administrative burden of what is otherwise a fantastic management technology.

You can also filter for settings that have Comments. This is also a new feature introduced in Windows Server 2008. You can now place a comment on any setting that you want. This means when admins are authoring policy, they can document their intentions at author time and other administrators can use that Comment as a search criteria. This feature is incredible at helping Group Policy administrators communicate to themselves, or other administrators, why specific settings are being managed and what the impact of those settings is.

The last piece of this feature set is called Starter GPOs. Starter GPOs are a starting point for administration. When a GPO is created, you can still create a blank GPO, or you can choose to create your GPO from one of the pre-existing Starter GPOs. Starter GPOs are a collection of preconfigured Administrative Template settings, complete with comments. You will see a node in the Group Policy Management Console (GPMC) called Starter GPOs. Simply right-click on this node and choose New. You will have a Starter GPO that is available to edit. There is delegation available on the Starter GPO container to ensure that only specific administrators can modify it..

This feature set—Search/Filters, Comments, and Starter GPOs—comes together to greatly enhance the authoring and management experience around Group Policy. It provides ease of authoring and discovering settings, inline documentation of Group Policy settings, and baseline configurations for starting the process.

ADMX/ADML

ADMX/AMDL files were introduced in Windows Vista to replace the legacy data format of the ADM files that we have become used to. ADMX files are XML files that contain the same type of information that we have become familiar with to build the administrative experience around Administrative Template settings. Using XML makes the whole process more efficient and standardized. ADML files are language-specific files that are critical in a multilanguage enterprise. In the past, all localization was done right within each ADM file. This caused some confusing version control issues when multiple administrators were managing settings in a GPO from workstations using different languages. With ADMX/ADML, all administrators work off of the same GPOs and simply call the appropriate ADML file to populate the editor.

Another value associated with ADML/ADMX files is that GPOs no longer contain the ADM files themselves. Prior to Windows Vista/Windows Server 2008, each GPO created

would contain all the ADM files. This was about 4 MB by default. This was a contributing factor in SYSVOL bloat.

Take a look at *http://www.microsoft.com/GroupPolicy* to read more on ADMX/ADML. You can also find the ADMX migration utility to help in moving to this new environment at *http://technet2.microsoft.com/windowsserver/en/technologies/featured/gp/default.mspx*. Just a note that ADM and ADMX can coexist; read up on it on one of the sites just referenced.

Central Store

Related to ADMX files is the Central Store. As was previously stated, ADM files used to be stored in the GPO itself. That is no longer the case. Now the GPO contains only the data that the client needs for processing Group Policy. In Windows Vista/Windows Server 2008, the default behavior for editing is that the editor pulls the ADMX files from the local workstation. This is great for smaller environments with few administrators managing Group Policy, but in larger, more complex environments or environments that need a bit more control, a Central Store has been introduced. The Central Store provides a single instance in SYSVOL that holds all of the ADMX/ADML files that are required. Once the Central Store is set up, all administrators load the appropriate files from the Central Store instead of the local machine. Check out one of the Group Policy MVP's Central Store Creation Utility at *http://www.gpoguy.com/cssu.htm*. You can also find more information on the Central Store at *http://www.microsoft.com/grouppolicy*.

Summary

Windows Server 2008 and Windows Vista have introduced a lot of new functionality for Group Policy. Administrators will find that these new features for management, along with the around 700 new settings to manage, will increase the ease of use of Group Policy and expand the number of areas that can be managed with policy.

–Kevin Sullivan
Lead Program Manager for Group Policy, Windows Enterprise Management Division

Pretty cool enhancements, eh? Sorry, that's the Canadian coming out of me, or through me, or channeling through me—whatever.

Windows Management Instrumentation

WMI is a core Windows management technology that administrators can use to write scripts to perform administrative tasks on both local and remote computers. There are no specific enhancements to WMI in Windows Server 2008 beyond those included in Windows Vista,

but it's important to know about the Windows Vista enhancements since these apply to Windows Server 2008 also. Here are a few of the more significant changes to WMI in Windows Vista and Windows Server 2008:

- **Improved tracing and logging** The WMI service now uses Event Tracing for Windows (ETW) instead of the legacy WMI log files used on previous Windows platforms, and this makes WMI events available through Event Viewer or by using the Wevtutil.exe command-line tool.
- **Enhanced WMI namespace security** The NamespaceSecuritySDDL qualifier can now be used to secure any namespace by setting WMI namespace security in the Managed Object Format (MOF) file
- **WMI namespace security auditing** WMI now uses the namespaces system access control lists (SACL) to audit namespace activity and report events to the Security event log.
- **Get and Set security descriptor methods for securable objects** new scriptable methods to get and set security descriptors have been added to Win32_Printer, Win32_Service, StdRegProv, Win32_DCOMApplicationSetting, and __SystemSecurity.
- **Manipulate security descriptors using scripts** The Win32_SecurityDescriptorHelper class now has methods that allow scripts to convert binary security descriptors on securable objects into Win32_SecurityDescriptor objects or Security Descriptor Definition Language (SDDL) strings.
- **User Account Control** User Account Control (UAC) affects what WMI data is returned, how WMI is remotely accessed, and how scripts must be run.

What all this basically means is that WMI is more secure and more consistent in how it works in Windows Server 2008, which is good news for administrators who like to write WMI scripts to manage various aspects of their Windows-based networks.

Still, from personal experience, I know that writing WMI scripts isn't always easy, especially if you're trying to get them to run properly against remote machines. Windows Vista and Windows Server 2008 complicate things in this regard because of their numerous security improvements, including User Account Control (UAC). So it's instructive if we sit back and listen now to one of our experts at Microsoft, who will address this very issue in detail (this sidebar is worth its weight in gold):

From the Experts: WMI Remote Connection

Talking about management obviously implies the need to connect remotely to the Windows systems you want to manage. Speaking about remote connection immediately implies security. Management and security are not always easy to combine. It is not rare to see situations where management represents a breach of security, or the other way around; it is not rare either to see security settings preventing the proper management of

a system. In this respect, WMI is not different from any other technologies; it provides remote management capabilities involving some security considerations.

Windows Vista and Windows Server 2008 come with a series of new security features. The most important one is called User Account Control (UAC). It is very likely that every administrator in the world will be challenged by the presence of UAC, especially if you use the Local Accounts part of the Administrator group to perform remote access. This is because any token account used in this context is automatically filtered and finally acts as a normal user in the remote system. Therefore, it is wise to consider the various security aspects to properly and securely manage your remote systems.

Before looking at the UAC aspects, let step back and look at the requirements to call WMI remotely. This applies to any Windows platform since Windows 2000. We will examine the Windows Vista and Windows Server 2008 aspects next.

To connect remotely, four conditions must be met:

1. **Firewall** Introduced with Windows XP, the Windows Firewall must be properly set up to enable connectivity for the WMI RPC traffic. Usually, you get an "RPC connection failure" if the Windows Firewall is enabled and RPC is disallowed. If you get an "access denied" message, the firewall is not the root cause of the issue. Keep in mind that the firewall is the key component to go through before anything else happens. Before Windows Vista and Windows Server 2008, RPC traffic must be enabled to allow the WMI traffic to go through. With Windows Vista and Windows Server 2008, a dedicated set of Firewall WMI rules is available to enable only WMI traffic. (This can be done with the FW.MSC MMC snap-in, Group Policies, Scripting, or NETSH.EXE.) Note that if you use WMIDiag (available on Microsoft Download Center), it will tell you which NETSH.EXE command to use to configure your firewall properly.
2. **DCOM** Once the firewall gate is passed, it is time to consider the DCOM security. The user issuing the remote call must have the right to "Launch and Activate" (which can be viewed and changed with DCOMCNFG.EXE) for both the My Computer and Windows Management and Instrumentation objects. By default, only users who are part of the Administrators group of the remote machine have the right to remotely "Launch and Activate" these DCOM objects.
3. **WMI namespace** Once the DCOM security is verified, WMI namespace security comes next. In this case, the user connecting to a remote WMI namespace must have at the minimum the Enable Remote and Enable Account rights granted for the given namespace. By default, only users who are part of the Administrators group of the remote machine have the Enable Remote right granted. (This can be updated with WMIMGMT.MSC.)

4. **Manageable entity** Last but not least, once WMI has accepted the remote request, it is actually executed against the manageable entity (which could be a Windows Service or a Terminal Server configuration setting, for instance). This last step must also succeed for the WMI operation to succeed. WMI does not add any privilege that the user does not have when issuing the WMI request. (By default, WMI impersonates the calls, which means it issues the call within the security context of the remote user.) So, depending on the WMI operation requested and the rights granted to the remote user, the call might succeed or fail at the level of the manageable entity. For instance, if you try to stop a Windows service remotely, the Service Control Manager requires the user to be an Administrator by default. If you are not, the WMI request performing this operation will fail.

This describes the behavior of WMI since Windows 2000. In the light of Windows Vista and Windows Server 2008, things can be slightly different because UAC is enabled by default on both platforms and everything depends on whether you use a local account or a domain account. If you use a local user of the remote machine who is a member of the Local Administrators group, the Administrators membership of the user is always filtered. In this context, DCOM, WMI, and the manageable entity are applying the security restrictions with respect to the filtered token presented. Therefore, with respect to the UAC behavior, the token is a user token, not an administrative token! As a consequence, the Local User is actually acting as a plain user on that remote machine even if it is part of the Local Administrators group. By default, a user does not have the rights to pass the security gates defined earlier (in step 2, 3, and 4).

Now that the scene is set, how do you manage a remote Windows Vista machine or 2008 server while respecting the Firewall, UAC, DCOM, WMI, and manageable entity security enforcements?

This challenge must be looked at in two different ways:

1. **The remote machine is part of a domain.** If the remote machine is part of a domain, it is highly recommended that you use a Domain User part of the Local Administrators group of the remote machine (and *not* a Local User part of the Local Administrators group). By doing so, you will be a plain Administrator because UAC does *not* filter users out of the Local Administrators group when the user is a Domain User. UAC only filters Local Users out of the Local Administrators group.
2. **Your machine is a workgroup machine.** If your machine is in a workgroup environment, you are forced to use a Local User part of the Local Administrators group to connect remotely. Obviously, because of the UAC behavior, that user is filtered and acts as a plain user. The first approach if you are in a large enterprise infrastructure is to consider the possibility of making this machine part of a domain to use a

Domain User. If this is not possible because you must keep the machine as part of a workgroup, from this point you have two choices:

- You decide to keep UAC active. In this case, you must adjust the security settings of DCOM and WMI to ensure that the Local User has the explicit rights to get remote access. Don't forget that a best practice is to use a dedicated Local Group and make this Local User a member of that group. In this context, the WMI requests against the manageable entity might work or not depending on the manageable entity security requirements (discussed in step 3). If the manageable entity does not allow a plain user to perform the task requested, you might be forced to change the security at the manageable entity level to explicitly grant permissions to your Local User or Group as well. Note that this is not always possible because it heavily depends on the manageable entity security requirements and security management capabilities of the manageable entity. For the Windows Services example, this can be done with the SC.EXE command via an SDDL string, the *Win32_Service* WMI class (with the *Get/SetSecurityDescriptor* methods implemented in Windows Vista and Windows Server 2008), or Group Policies (GPEDIT.MSC). By updating the security at these three levels, you will be able to gracefully pass the DCOM and WMI security gates and stop a Windows Service as a plain user. Note that this customization represents clearly the steps for a granular delegation of the management. Only the service you changed the security for can be stopped by that dedicated user (or group). In this case, you actually define a very granular security model for a specific task. (You can watch the "Running Scripts Securely While Handling Passwords and Security Contexts Properly" webcast at *http://go.microsoft.com/fwlink/?LinkId=39643* to understand this scenario better.) Now it is possible that some manageable entities only require the user to be an Admin (typical for most devices) because there is no possibility to update the security descriptor. In such a case, for a workgroup scenario, only the second option (discussed next) becomes possible. Last but not least, keep in mind that these steps are also applicable in a domain environment to delegate some management capabilities to a group of domain users.
- You decide to disable the UAC filtering for remote access. This must be the last-resort solution. It is not an option you should consider right away if you want to maintain your workgroup system with a high level of security. So consider it only after investigating the possibility of making your system part of a domain or after reviewing the security wherever needed. If making your system part of a domain is not possible, you can consider this option. In this case, you must set the registry key in the reference shown below to ZERO on

the remote system. Note that you must be an administrator to change that registry key. So you need to do this locally once, before any remote access is made. Note that this configuration setting disables the filtering on Local Accounts only; it does not disable UAC as a whole.

```
[HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System]"Local
AccountTokenFilterPolicy"=dword:00000001
```

Once set, the registry key is created and set to ONE, and the Local User remotely accessing the machine will be an administrator (if the user is a member of the Local Administrators group).Therefore, by default, the user will pass the security gates defined in steps 2, 3, and 4. Note that it is required to reboot the machine to get this change activated.

–Alain Lissoir
Senior Program Manager, Windows Enterprise Management Division (WEMD)
Check out Alain's Web site at http://www.lissware.net.

Windows PowerShell

Another powerful tool for automating administrative tasks in Windows Server 2008 is Windows PowerShell, a command-line shell and scripting language. PowerShell includes more than 130 command-line tools (called *cmdlets*), has consistent syntax and naming conventions, and uses simplified navigation for managing data such as the registry and certificate store. PowerShell also includes an intuitive scripting language specifically designed for IT administration. As of Beta 3, PowerShell is included as an optional feature you can install on Windows Server 2008.

PowerShell can be used to efficiently perform Windows Server 2008 administration tasks, including managing services, processes, and storage. PowerShell can also be used to manage aspects of server roles, such as Internet Information Services (IIS) 7.0, Terminal Services, and Active Directory Domain Services. Some of the things you can do with PowerShell on Windows Server 2008 include

- Managing command-line services, processes, the registry, and WMI data using the get-service, get-process, and get-wmiobject cmdlets.
- Automating Terminal Services configuration, and comparing configurations across a Terminal Server farm.
- Deploying and configuring Internet Information Services 7.0 across a Web farm.
- Creating objects in Active Directory, and listing information about the current domain.

For example, let's look at the third item in this list—managing IIS 7.0 using PowerShell. But rather than have me explain this, why don't we listen to one of our experts at Microsoft concerning this?

From the Experts: PowerShell Rocks!

Of all the new Microsoft technology coming down the pipe, PowerShell has got to be one of the most exciting (after IIS 7.0 of course). You might wonder why I am so excited about the new scripting shell for Windows. Even if PowerShell is better than Command Prompt on steroids, what does this have to do with my main passion, Web servers and Web applications? Check out the Channel9 video I did with Jeffrey Snover, architect of PowerShell, to get an idea of how cool PowerShell really is (see *http://channel9.msdn.com/Showpost.aspx?postid=256994*). In the video, we show off a demo we put together for Bob Muglia's keynote article in TechEd IT Forum this week, which appears to have gone very, very well. Well done, Jeffrey.

A long, long, long time ago, when I was in school and even after that, before I came to Microsoft and joined the IIS team, I used Linux and spent my days in BASH and ZSH getting work done. Until now, we sadly never really had the productive power of an interactive shell on Windows. So as a previously heavy user of shells, I have to tell you what I really like about this new shell interface on its own, and then I'll explain the many ways PowerShell can make work simpler for IIS administrators.

OK, first off, in PowerShell you input commands on objects, not text, and PowerShell returns objects and not text. So you can easily pipe commands together in one line. This allows me to input in just one line complicated commands like this one:

```
PS C:\Windows\System32> Get-ChildItem -Path G:\ -Recurse -Include
  *.mp3 | Where-Object -FilterScript {($_.LastWriteTime -gt
  "2006-10-01") -and ($_.Name -match "pearl jam")}| Copy
  -Destination C:\User\bills\Desktop\New_PJ_MP3s
```

which recursively looks through my entire external hard drive (G:), collects all the "Pearl Jam" mp3s that were recently added, and copies them into a folder on my desktop. Never was I given a text output listing all the mp3s, and I didn't have to use the Copy command over and over. I just piped all the items to Copy once.

Another thing I like so much about PowerShell is how consistent PowerShell commands are. In the preceding example, I used only one Get-ChildItem command, but rest assured if I wanted to get anything else, the command for that would start with Get. Similarly, if we want to stop a process or an application or anything, we always use the Stop command, not kill, not terminate, not halt, just stop.

Finally, I love that PowerShell is extensible. I love this because it means my team can produce a whole set of IIS PowerShell cmdlets to help you manage IIS 6.0, IIS 7.0, and future versions of IIS. You will also be able to submit your IIS PowerShell scriptlets to this community area (coming very soon).

Now that I've listed my favorite things about this new shell, I'd like to give you a few ways that PowerShell can and will make IIS administration simpler than ever before. The top 5...

1. IIS 7.0 has a new WMI Provider for quickly starting, stopping, creating, removing, and configuring sites and applications. Now use PowerShell to give a list of applications sorted by a particular configuration setting. Then pipe apps with the particular setting into the tasks you were performing before with the WMI Provider. My colleague Sergei Antonov wrote and just published a fantastic article, titled "Writing PowerShell Command-lets for IIS 7.0," that describes how to write PowerShell cmdlets using our WMI provider.
2. 2. Because IIS 7.0 has a distributed file-based configuration store, you can store your application's IIS configurations in a *web.config* file in the application's directory next to its code and content. Use PowerShell to rapidly XCopy deploy the application to an entire Web farm in one step.
3. IIS 7.0's new Web.Administration API allows admins to write short programs in .NET to programmatically tackle frequent IIS 7.0 management tasks. Then, because PowerShell completely supports the .NET Framework, use it to pipe IIS objects in and out of these handy bits of code.
4. With IIS 7.0, you can use the new Runtime Status and Control API to monitor the performance of your Web applications. Use PowerShell to monitor performance information at a regular interval of every five minutes, and then have this valuable runtime information displayed to the console or sent to a log file whenever CPU is above 80%.
5. Take advantage of IIS 7.0's extensibility by writing your own custom request processing module with its own configuration and IIS Manager plug-in. Then write a PowerShell cmdlet to serve as a management interface to expose your custom IIS configuration to the command line and to power your IIS Manager plug-in.

For more information on managing IIS 7.0 using PowerShell, see "An Introduction to Windows PowerShell and IIS 7.0," found at *http://www.iis.net/default.aspx?tabid=2&subtabid=25&i=1212*.

–Bill Staples
Product Unit Manager, IIS

Like WMI discussed earlier, Windows PowerShell is a work in progress and is still evolving. For example, Windows PowerShell version 1.0 doesn't yet have any cmdlets for managing Active Directory, but by using the .NET Framework 2.0 together with PowerShell, you can manage Active Directory even so.

Chapter 14, "Additional Resources," has lots of pointers to where you can find more information about using PowerShell to manage Windows Server 2008. But before you flip ahead to look there, listen to what another expert at Microsoft has to say concerning the *raison d'être* behind PowerShell:

From the Experts: The Soul of Automation

"Civilization advances by extending the number of important operations which we can perform without thinking about them."

Alfred North Whitehead, "Introduction to Mathematics" (1911)
English mathematician & philosopher (1861 - 1947)

I really understood Whitehead's point during the great windstorm of 2006 when we lost power in my area for six days. During this time, we were without the benefits of most of the things I took for granted. I was struck by how much time it took to do things that previously I performed without thinking about them. Washing the dishes in the sink by hand took a lot more time than using the dishwasher. There were dozens of things like this. I didn't mind terribly, but I found myself resenting that I didn't have time to do as much reading as I usually do.

Whitehead's point is *not* that civilization advances by us becoming non-thinking idiots. Rather, by increasing the number of things that we don't have to think about, we free up time to think about *new* things and solve *new* problems, and then transform those things into things that we no longer have to think about. And so on and so on. Because I spent time doing dishes means that I didn't have time to read, which meant that I didn't get more educated, which would have made it easier to move the ball forward.

This is the essence of PowerShell and the soul of automation. In our world, there is no end of interesting and hard problems to think about, and the degree that our tools continue to make us think about the low-level junk is the degree to which we reduce the time that we have to think about the interesting problems. The ball gets moved forward as we adopt better and better tools that do what we want them to do without us having to tell them, and by our getting in the habit of using automation for repeating operations and sharing that automation with others.

Huge advances come from the accumulation of small deltas. In *David Copperfield*, Charles Dickens wrote, "Annual income twenty pounds, annual expenditure nineteen pounds six, result happiness. Annual income twenty pounds, annual expenditure twenty ought and size, result misery." Einstein said it this way, "The most powerful force in the universe is compound interest." So the next time you find yourself thinking about

how to do something that you've done before, you should take it as an opportunity to invest a little bit and automate the activity so that you don't have to think about it again. Give the function a good long name so that you can remember it, find it, and recognize it when you see it; then give it an alias so that you can minimize your typing (for example, Get-FileVersionInfo and gfvi).

Last but not least, SHARE. Put your script out on a blog or newsgroup or Web site so that others can benefit from your thinking. Newton might have figured out gravity, but if he didn't share his thoughts with others, he would not have moved the ball forward. OK, so your script is not in the same league as "F=ma," but share it anyway because "huge advances come from the accumulation of small deltas."

Enjoy!

–Jeffrey Snover
Partner Architect, Windows Management

Microsoft System Center

Finally, the Microsoft System Center family of enterprise management solutions will be supporting management of Windows Server 2008, though at the time of this writing, the date for such support has not been made known to me. System Center is a collection of products that evolved from the earlier Microsoft Systems Management Server (SMS) and Microsoft Operations Manager (MOM) platforms. The plan for the System Center family currently includes the following products:

- System Center Operations Manager (the next generation of MOM)
- System Center Configuration Manager (the next generation of SMS)
- System Center Data Protection Manager
- System Center Essentials
- System Center Virtual Machine Manager
- System Center Capacity Planner

Keep your eye on these products as Microsoft announces its support for Windows Server 2008. You can find out more about System Center at *http://www.microsoft.com/systemcenter*.

Conclusion

Windows Server 2008 can be managed using a number of in-box and out-of-band tools. If you only need to manage a single server, use Initial Configuration Tasks and Server Manager. If you need to do this remotely, enable Remote Desktop on your server. If you need to manage multiple servers roles on different machines, install the Remote Server Administration Tools (RSAT) and use each tool to manage multiple instances of a particular role. And if you need to automate the administration of Windows Server 2008 machines, use ServerManagerCmd.exe, WMI, Windows PowerShell, or some combination of the three.

Additional Resources

TechNet has a level 300 webcast called "Installing, Configuring, and Managing Server Roles in Windows Server 2008" that you can download from *http://msevents.microsoft.com/cui/WebCastEventDetails.aspx?EventID=1032294712&EventCategory=5&culture=en-US&CountryCode=US* (registration required).

If you have access to the Windows Server 2008 beta on Microsoft Connect (*https://connect.microsoft.com/*), you can download the following items:

- Microsoft Windows Server 2008 Server Manager Lab Companion
- Microsoft Windows Server 2008 Initial Configuration Tasks Step-By-Step Guide
- Live Meeting on Server Manager

If you don't have access to beta builds of Windows Server 2008, you can still test drive Server Manager online using the Microsoft Windows Server 2008 Server Manager Virtual Lab, available at *http://msevents.microsoft.com/CUI/WebCastEventDetails.aspx?EventID=1032314461&EventCategory=3&culture=en-IN&CountryCode=IN*.

A good starting point for exploring the potential of using Windows PowerShell to manage Windows Server 2008 is *http://www.microsoft.com/windowsserver/2008/powershell.mspx*.

Information about Group Policy enhancements in Windows Vista and Windows Server 2008 can be found at *http://technet2.microsoft.com/WindowsVista/en/library/a8366c42-6373-48cd-9d11-2510580e48171033.mspx?mfr=true*.

More information about WMI enhancements in Windows Vista and Windows Server 2008 can be found on MSDN at *http://msdn2.microsoft.com/en-gb/library/aa394053.aspx*.

And if you want to find out more about Microsoft System Center, see *http://www.microsoft.com/systemcenter/*.

Finally, be sure to turn to Chapter 14 for more information on the topics in this chapter and also for webcasts, whitepapers, blogs, newsgroups, and other sources of information about all aspects of Windows Server 2008.

Chapter 6

Windows Server Core

In this chapter:

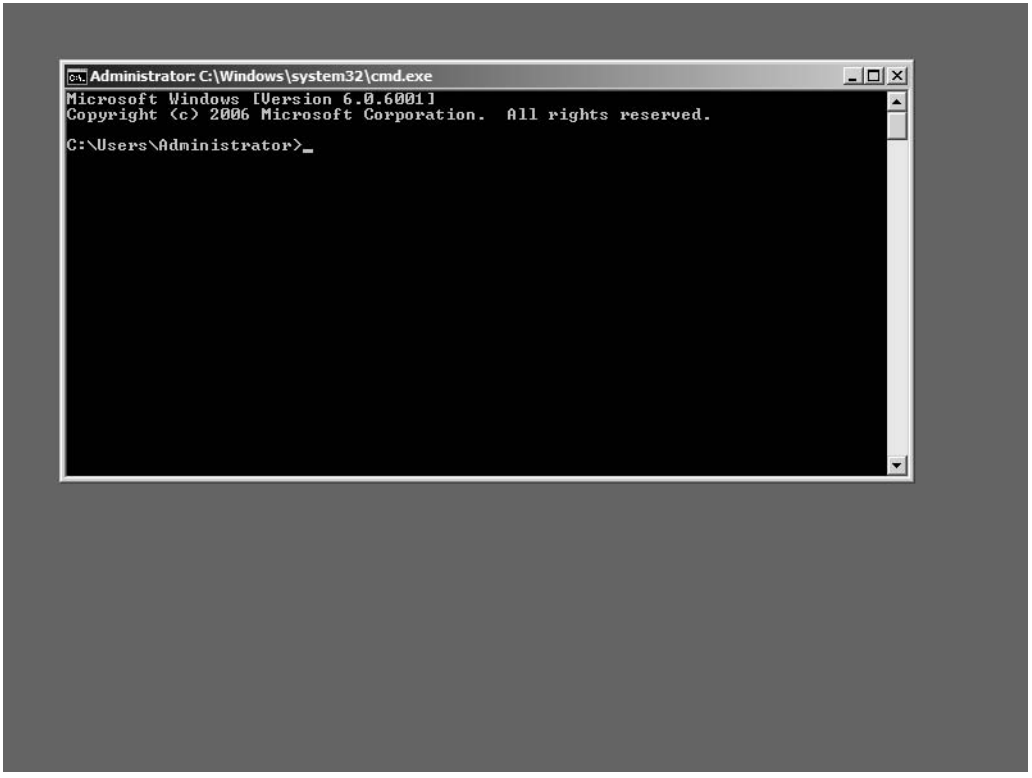
When you try to install Microsoft Windows Server 2008 manually from media on a system, you're presented with two installation options to choose from:

- A full installation of the Microsoft Windows Server 2008 operating system
- A Windows server core installation of the Windows Server 2008 operating system

Selecting the first option means you get the type of Windows server you're used to, with its full slate of GUI tools, support for the .NET Framework, and support for a wide range of possible roles and features you can install on your machine. But what if you select the second option? What's a Windows server core installation of Windows Server 2008? And how does this differ from a full installation of the product? Well, that's what this chapter is all about—read on!

What Is a Windows Server Core Installation?

The best way of learning about the Windows server core installation option is to simply install it and log on. Here's what you see when you first log on to a Windows server core server.



That's it? Where's the task bar and Start menu? There is no task bar or Start menu. How do you start Windows Explorer then? You can't—the tool is not available in a Windows server core installation. Where's the Initial Configuration Tasks screen? It's not there. How can I open Server Manager to add roles and features? Sorry, Server Manager is unavailable on a Windows server core installation. Well, what can I do with this thing then? Am I stuck with only a command prompt to work with?

You can do a lot with a Windows server core installation, as we'll see in a moment. And no, you're not just stuck with a command prompt. But if you were, would it be bad? Ever hear a Unix admin complain about "being stuck" with having to use the command line to administer a server? Isn't command-line administration of servers a *good* thing because it means you can automate complex management tasks using batch files and scripts and there is no graphical UI taking resources away from server tasks?

And that's one of the things that a Windows server core installation is all about—scripted administration of Windows servers in enterprise (and especially datacenter) environments. But why remove the desktop and all the GUI management tools? Doesn't that cripple the server? Not at all—in fact, just the opposite!

Understanding Windows Server Core

Windows server core is a "minimal" installation option for Windows Server 2008. What this means is that when you choose this option during setup (or when using unattended setup), Windows Server 2008 installs a minimum set of components on your machine that will allow you to run certain (but not all) server roles. In other words, selecting the Windows server core installation option installs only a subset of the binaries that are installed when you choose the full installation option for Windows Server 2008.

Here are some of the Windows Server 2008 components that are *not* installed when you specify the Windows server core installation option during setup:

- No desktop shell (which means no glass, wallpaper, or screen savers either)
- No Windows Explorer or My Computer (we already said no desktop shell, right?)
- No .NET Framework or CLR (which means no support for managed code, which also means no PowerShell support)
- No MMC console or snap-ins (so no Administrative tools on the Start menu—whoops! I forgot, no Start menu!)
- No Control Panel applets (with a few small exceptions)
- No Internet Explorer or Windows Mail or WordPad or Paint or Search window (no Windows Explorer!) or GUI Help and Support or even a Run box.

Wow, that sounds like a lot of stuff that's missing in a Windows server core installation of Windows Server 2008! Actually though, it's not—compare the preceding list to the following list of components that *are* available on a Windows server core server.

First, you've still got the kernel. You always need the kernel.

Then you've got hardware support components such as the Hardware Abstraction Layer (HAL) and device drivers. But it's only a limited set of device drivers that supports disks, network cards, basic video support, and some other stuff. A lot of in-box drivers have been removed from the Windows server core installation option, however—though there is a way to install out-of-box drivers if you need to, as we'll see later in this chapter.

Next, you've still got all the core subsystems that are needed by Windows Server 2008 in order to function. That means you've got the security subsystem and Winlogon, the networking subsystem, the file system, RPC and DCOM, SNMP support, and so on. Without these subsystems, your server simply wouldn't be able to do anything at all, so they're a necessity for a Windows server core installation.

Then you've got various components you need to configure different aspects of your server. For example, you have components that let you create user accounts and change passwords, enable DHCP or assign a static IP address, rename your server or join a domain, configure Windows Firewall, enable Automatic Updates, choose a keyboard layout, set the time and date, enable Remote Desktop, and so on. Many of these configuration tasks can be performed

using various command-line tools included in a Windows server core installation (more about tools in a moment), but a few of them use scripts or expose minimal UI.

There are some additional infrastructure components present as well on a Windows server core installation. For instance, you still have the event logs plus a command-line tool for viewing, configuring, and forwarding them using Windows eventing. You've got performance counters and a command-line tool for collecting performance information about your server. You have the Licensing service, so you can activate and use your server as a fully licensed machine. You've got IPSec support, so your server can securely communicate on the network. You've got NAP client support, so your server can participate in a NAP deployment. And you've got support for Group Policy of course.

Then there are various tools and infrastructure items to enable you to manage your Windows server core server. As we saw in our screen shot earlier, you've got the command prompt cmd.exe, so you can log on locally to your server and run various commands from a command-prompt window. In fact, as we saw, a command-prompt window is already open for you when you first log on to a Windows server core server. What happens, though, if you accidentally close this window? Fortunately, a Windows server core installation still includes Task Manager, so if you close your command window you can start another by doing the following:

1. Press CTRL+SHIFT+ESC, to open Task Manager.
2. On the Applications tab, click New Task.
3. Type **cmd** and click OK.

In addition to the command prompt, of course, there are dozens (probably over a hundred, and more when different roles and features are installed) of different command-line tools available on Windows Server 2008 for both full and server core installation options. What I'm talking about is Arp, Assoc, At, Attrib, BCDEdit Cacls, Certutil, Chdir, chkdsk, Cls, Copy, CScript, Defrag, Dir, and so on. A lot of the commands listed in the "Windows Command-Line Reference A–Z," found on Microsoft TechNet, are available on a Windows server core server—not all, mind you, but a lot of them.

You can also enable Remote Desktop on a Windows server core installation, and this lets you connect to it from another machine using Remote Desktop Connection (RDC) and start a Terminal Services session running on it. Once you've established your session, you can use the command prompt to run various commands on your server, and you can even use the new Remote Programs feature of RDC 6.0 to run a remote command prompt on a Windows server core server from an administrative workstation running Windows Vista. (We'll learn more about that soon.)

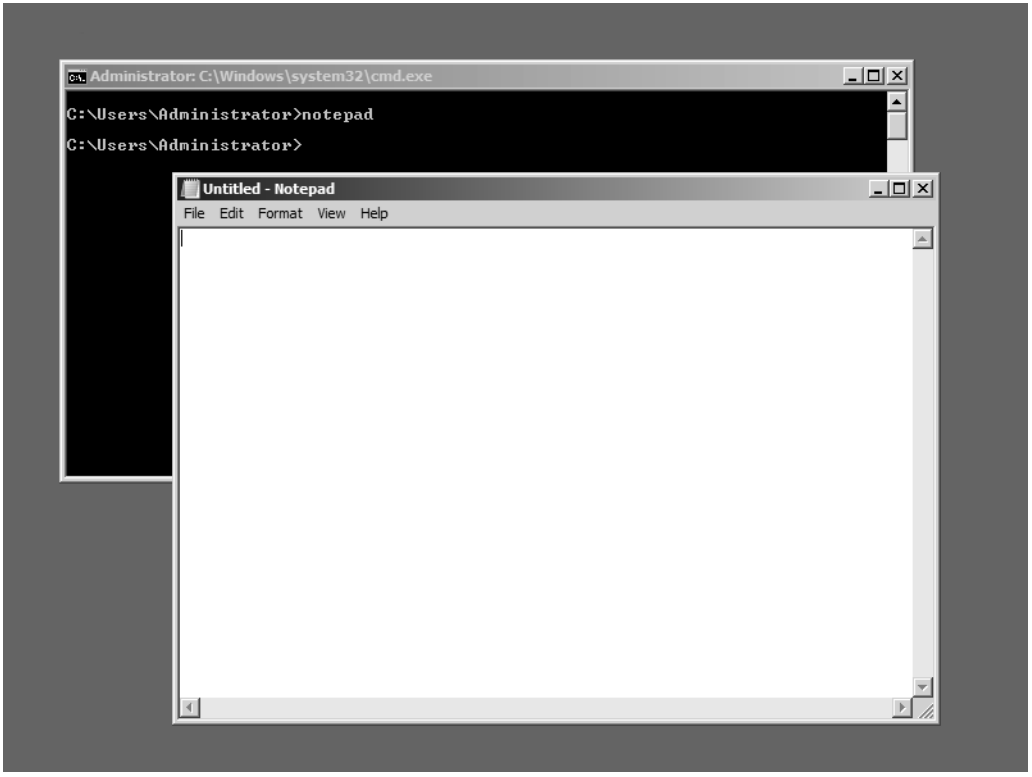
There's also a WMI infrastructure on your Windows server core server that includes many of the usual WMI providers. This means you can manage your Windows server core server either by running WMI scripts on the local machine from the command prompt or by scheduling their operation using schtasks.exe. (There's no Task Schedule UI available, however.) Or you can manage your server remotely by running remote WMI scripts against it from another machine. And having WMI on a Windows server core server means that remote UI tools

such as MMC snap-ins running on other systems (typically, either a full installation of Windows Server 2008 or an administrator workstation running Windows Vista with Remote Server Administration Tools installed) can connect to and remotely administer your Windows server core server. Plus there's also a WS-Management infrastructure on a Windows server core installation. WS-Management is a new remote-management infrastructure included in Windows Vista and Windows Server 2008, and involves Windows Remote Management (WinRM) on the machine being managed and the Windows Remote Shell (WinRM) for remote command execution from the machine doing the managing. We'll talk about remote management of Windows server core servers later in this chapter.

Then there are various server roles and optional features you can install on a Windows server core server so that the machine can actually do something useful on your network, like be a DHCP server or a domain controller or print server. We'll look later at exactly which roles and features are available for installing on a Windows server core server and which roles/features you can't install.

Then there are a few necessary GUI tools that actually *are* present on a Windows server core server. For example, we already saw that the command prompt (cmd.exe) is available, and so is Task Manager. Another useful tool on a Windows server core server is Regedit.exe, which can be launched either from the command line or from Task Manager. Then there's Notepad.

Notepad?



Yes, Notepad. The reason for including Notepad on a Windows server core installation option of Windows Server 2008 is simple: Microsoft listens to its customers. I'm not kidding! (Plus I'm serious about Microsoft listening to customers.) During the early stages of developing and testing Windows Server 2008, one of the most common requests from participants in the Microsoft Technology Adoption Program (TAP) for Windows Server 2008 was this: We need a tool on Windows server core servers that we can use to view logs, edit scripts, and perform other essential administrative tasks. Give us Notepad! We want Notepad!

Who ever expected that the lowly and oft-maligned Notepad would be so important to administrators who work in enterprise environments?

Anyway, before we move on and talk a bit about the rationale behind why Microsoft decided to offer the Windows server core installation option in Windows Server 2008, let's hear from one of our experts about how the Windows server core product team managed to make this thing work. After all, Windows components have a lot of dependencies with one another and especially with the desktop shell and Internet Explorer, so it will be interesting to hear how they took so many components out of this installation option for the product without causing it to break. Plus we'll also learn a bit about how we can try to get applications that we need to have running on a Windows server core server running properly. And finally, we'll learn something about getting Notepad to run properly on a Windows server core server:

From the Experts: Shimming Applications in Windows Server Core

The primary goal of the Windows server core installation option is to minimize the disk and servicing footprint. Thus, a number of Windows components—such as Media Player and Internet Explorer—are not installed as part of a Windows server core installation. This means that because of their dependencies on parts of Internet Explorer, the common dialog boxes are not functional in a Windows server core installation. Thus, the file open and save dialog boxes in Notepad, for example, will not work.

A Windows server core installation leverages the application compatibility shim infrastructure in Windows to develop a clever solution to this problem. A *shim* is a thin layer of code that sits between an application and a Windows API. The shimming infrastructure redirects the API call made by the application to the shim code, which can then make some changes to the parameters, call the original API, or do something else entirely.

A Windows server core installation installs two shims. The first one is called *RegEditImportExportLoadHive* and is a specialized shim that allows *RegEdit* to import and export registry files. The second shim is called *NoExplorerForGetFileName*. It's a general shim for file open and save dialog boxes and is currently used by Notepad. This second shim changes some parameters to the API call that displays the file open or save dialog so that the old-style dialog box from pre-Windows 95 is displayed, instead of the new Explorer-style dialog box.

The shimming engine allows the end user to apply existing shims to other applications. The tool used to do this is the Application Compatibility Toolkit. Copy the sysmain.sdb database located at %SYSTEMROOT%\AppPatch (or %SYSTEMROOT%\AppPatch\AppPatch64 on x64 machines) on the Windows server core machine to a Windows Server 2008 machine. Use the Application Compatibility Toolkit to edit the database. Copy the new database back to the Windows server core machine, and install it using sdbinst.exe, located at %SYSTEMROOT%\System32.

–Rahul Prasad
Software Development Engineer, Windows Core Operating System Division

The Rationale for Windows Server Core

The need for something like the Windows server core installation option of Windows Server 2008 is pretty obvious. Windows Server today is frequently deployed to support a single role in an enterprise or to handle a fixed workload. For example, organizations often deploy the DHCP Server role on a dedicated Windows Server 2003 machine to provide dynamic addressing support for client computers on their network. Now think about that for a moment—you've just installed Windows Server 2003 with all its various services and components on a solid piece of hardware, just to use the machine as a DHCP server and nothing more. Or maybe as a file server as part of a DFS file system infrastructure you're setting up for users. Or as a print server to manage a number of printers on your network. The point is, you've got Windows Server 2003 with all its features doing only one thing. Why do you need all those extra binaries on your machine then? And think about when you need to patch your system—you've got to apply all new software updates to the machine, even though the functionality that many of those updates fix will never actually be used on that particular system. Why should you have to patch IIS on your server if the server is not going to be used for hosting Web sites? And might not having IIS binaries on your server make it more vulnerable even though the IIS component is not actually being used on it or is even installed? The more stuff you've got on a box, the more difficult it is to secure (or to be sure that it's secure) and the more complex it is to maintain.

Enter the Windows server core installation option of Windows Server 2008. Now, instead of installing all of Windows Server 2008 on your box while using only a portion of it, you can install a minimal subset of Windows Server 2008 binaries and you need to maintain only those particular binaries. The value proposition for enterprises of the Windows server core installation option is plain to see:

- Fewer binaries mean a reduced attack surface and, hence, a greater degree of protection for your network.
- Less functionality and a role-based paradigm also mean fewer services running on your machine and, therefore, again less attack surface.

- Fewer binaries also mean a reduced servicing surface, which means fewer patches, making your server easier to service and orienting your patch management cycle according to roles instead of boxes. Estimates indicate that using the Windows server core installation option can reduce the number of patches you need to apply to your server by as much as 50 percent compared with full installations of Windows Server 2008.
- Fewer roles and features also mean easier management of your servers and enable different members of your IT staff to specialize better according to the server roles they need to support.
- Finally, fewer binaries also mean less disk space needed for the core operating system components, which is a plus for datacenter environments in particular.

The Windows server core installation option of Windows Server 2008 is all of these and more, and it's included in the Standard, Enterprise, and Datacenter editions of Windows Server 2008. Windows server core is not a separate product or SKU—it's an installation option you can select during manual or unattended install. And it's available on both the x86 and x64 platforms of Windows Server 2008. (It's not available on IA64 and on the Web edition SKU of Windows Server 2008.) The bottom line? The Windows server core installation option of Windows Server 2008 is more secure and more reliable, and it requires less management overhead than using a full installation of Windows Server 2008 for an equivalent purpose in your enterprise.

A Windows server core server provides you with minimal server operating system functionality and a low attack surface for targeted roles. To give you a better idea of the functionality that is (and isn't) available in the Windows server core installation option, Table 6-1 shows included and excluded roles and Table 6-2 shows included and excluded optional features.

Table 6-1 Included/Excluded Roles in the Windows Server Core Installation Option of Windows Server 2008

Roles available	Roles unavailable
Active Directory	Active Directory Certificate Services
Active Directory LDS	Active Directory Federation Services
DHCP Server	Active Directory RMS
DNS Server	Application Server
File Services (includes DFSR and NFS)	Fax Server
Print Services	Network Policy and Access Services
Streaming Media Services	Terminal Services
Windows Server Virtualization	UDDI Services
	Web Server (IIS)
	Windows Deployment Services
	Windows SharePoint Services

Table 6-2 Included/Excluded Features in the Windows Server Core Installation Option of Windows Server 2008

Features available	Features unavailable
BitLocker Drive Encryption	.NET Framework 3.0
Failover Clustering	BITS Server Extensions
Multipath I/O	Connection Manager Administration Kit
Removable Storage Management	Desktop Experience
SNMP Services	Internet Printing Client
Subsystem for UNIX-based Applications	Internet Storage Naming Server
Telnet Client	LPR Port Monitor
Windows Server Backup	Message Queuing
WINS Server	Network Load Balancing
	Peer Name Resolution Protocol
	Remote Assistance
	Remote Server Administration Tools
	RPC over HTTP Proxy
	Simple TCP/IP Services
	SMTP Server
	Storage Manager for SANs
	Telnet Server
	TFTP Client
	Windows Internal Database
	Windows Process Activation Service
	Windows System Resource Manager (WSRM)
	Wireless Networking

Performing Initial Configuration of a Windows Server Core Server

In Chapter 5, "Managing Server Roles," we saw how to perform the initial configuration of a Windows Server 2008 server using the Initial Configuration Tasks screen. Of course, many of these initial configuration tasks can also be performed using an unattend.xml answer file during an unattended installation.

The Windows server core installation option of Windows Server 2008 can also have its initial configuration done in two ways: from the command line after a manual install, or by doing an unattended installation. In this chapter, we're going to look only at the first method (using the command line after a manual install). For more information on unattended installation of Windows Server 2008, see Chapter 13, "Deploying Windows Server 2008."

Performing Initial Configuration from the Command Line

Some of the initial configuration tasks you will want to perform on a Windows server core server include the following:

- Set a password for the Administrator account.
- Set the date, time, and time zone.
- Configure networking, which might mean assigning a static IP address, subnet mask, and default gateway (unless DHCP is being used) and pointing the DNS settings to a domain controller.
- Changing the server's name and joining the domain.

Other initial configuration tasks can include activating your server, enabling Automatic Updates, downloading and installing any available software updates, enabling Windows Error Reporting and the Customer Experience Improvement Program, and so on.

Let's see how to perform some of these tasks.

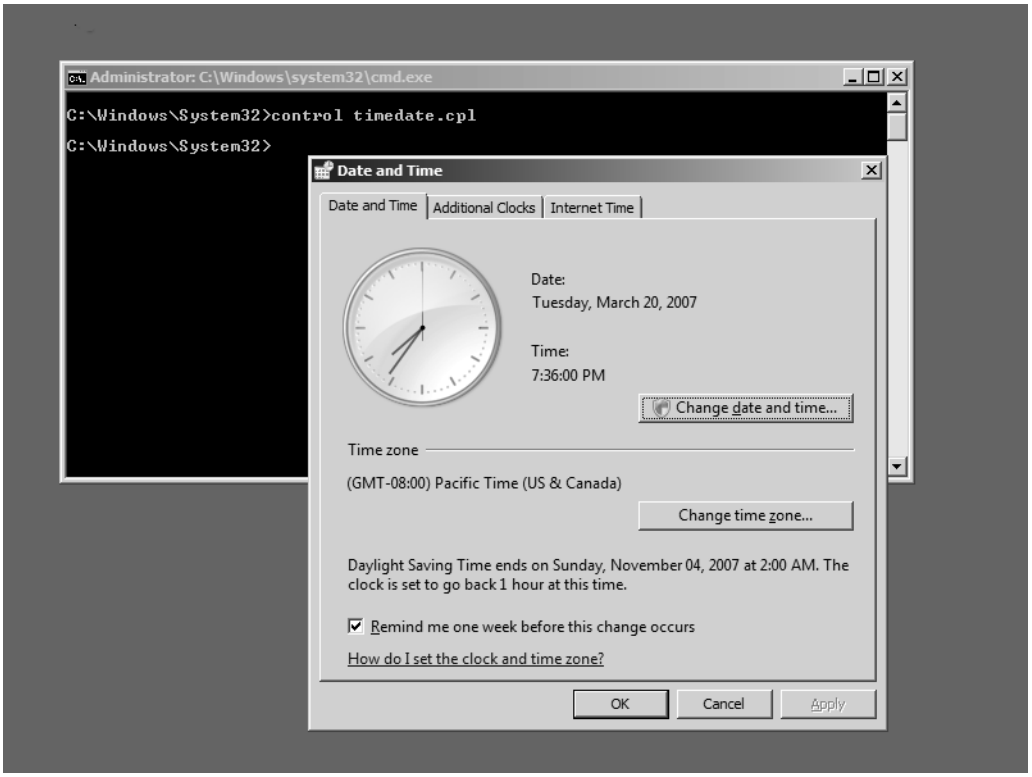
Changing the Administrator Password

There are two ways you can change the Administrator password on a Windows server core server:

- Press CTRL+ALT+DEL, click Change Password, and enter your old and new password.
- Type **net user administrator *** at the command prompt, and enter your new password twice.

Setting Date, Time, and Time Zone

To set the time zone for your server, type **control timedate.cpl** at the command prompt. This opens the same Date And Time applet that can be opened from Control Panel in the full installation of Windows Server 2008:



The reason for using a Control Panel applet to do these tasks is simply that it's easier for admins to do it this way than to try and do it from the command line. And because it's a task that is likely to be performed only occasionally (even just once), and because there are no dependencies between the Date And Time applet and other system components that have been removed from the Windows server core installation option, the product team decided to leave this in as one of the few GUI tools still available in the Windows server core installation option of Windows Server 2008. Of course, you can also specify these settings in an unattend.xml answer file if you're performing an unattended installation of your server. And by the way, control.exe by itself doesn't work on a Windows server core installation. Only the two included .cpls work.

Before we go further, let's briefly hear from one of our experts on the Windows Server 2008 product team at Microsoft concerning configuring the Windows server core installation option of Windows Server 2008:

From the Experts: Shell-less vs. GUI-less

If you have been working with a Windows server core installation, you might have noticed that there is some GUI support in a Windows server core installation of Windows Server 2008. To be completely accurate, the GUI of a Windows server core server is shell-less, not entirely GUI-less. There are several low-level GUI DLLs that are included because of current dependencies, such as gdi32.dll and shlwapi.dll. In a future release we hope to be able to remove the dependencies and also remove these files. However, including them does provide some advantages for making a Windows server core server easier to manage using the current tools.

In Beta 1, we didn't include any text editor. Although you could remotely connect to a Windows server core server to view logs, edit scripts, and so on, we heard lots of feedback that there should be an on-the-box text editor. Therefore, we added Notepad. However, because of the reduced environment the Windows server core installation option provides, not all of Notepad is functional—for example, help doesn't work.

In addition, the Windows server core installation option also includes two control panels, which you can access using the following commands:

- Control timedate.cpl
- Control intl.cpl

Timedate.cpl lets you set the time zone for your server, while intl.cpl lets you change your keyboard for different layouts.

–Andrew Mason
Program Manager, Windows Server

Configuring Networking

Now let's configure networking for our server. First let's run **ipconfig /all** and see the server's current networking settings:

```
C:\Windows\System32>ipconfig /all
Windows IP Configuration

   Host Name . . . . . . . . . . . . : LH-3TBCQ4I1ONRA
   Primary Dns Suffix  . . . . . . . :
   Node Type . . . . . . . . . . . . : Hybrid
   IP Routing Enabled. . . . . . . . : No
   WINS Proxy Enabled. . . . . . . . : No

Ethernet adapter Local Area Connection:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Intel 21140-Based PCI Fast Ethernet Adapter
(Emulated)
   Physical Address. . . . . . . . . : 00-03-FF-27-88-8C
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::c25:d049:5b0c:1585%2(Preferred)
   Autoconfiguration IPv4 Address. . : 169.254.21.133(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.0.0
   Default Gateway . . . . . . . . . :
   DHCPv6 IAID . . . . . . . . . . . : 67109887
   DNS Servers . . . . . . . . . . . : fec0:0:0:ffff::1%1
                                       fec0:0:0:ffff::2%1
                                       fec0:0:0:ffff::3%1
   NetBIOS over Tcpip. . . . . . . . : Enabled

Tunnel adapter Local Area Connection*:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : isatap.{B4B31F3D-B6C8-4303-BA3C-5A54B05F2FDD}
   Physical Address. . . . . . . . . : 00-00-00-00-00-00-00-E0
   DHCP Enabled. . . . . . . . . . . : No
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::5efe:169.254.21.133%3(Preferred)
   Default Gateway . . . . . . . . . :
   DNS Servers . . . . . . . . . . . : fec0:0:0:ffff::1%1
                                       fec0:0:0:ffff::2%1
                                       fec0:0:0:ffff::3%1
   NetBIOS over Tcpip. . . . . . . . : Disabled
```

Note that **ipconfig /all** displays two network interfaces on the machine: a physical interface (NIC) and an ISATAP tunneling interface. Before we can use netsh.exe to modify network

settings, we need to know which interface we need to configure. To determine this, we'll use the **netsh interface ipv4 show interfaces** command as follows:

```
C:\Windows\System32>netsh interface ipv4 show interfaces

Idx  Met   MTU   State        Name
---  ---  -----  -----------  -------------------
  2   20   1500  connected    Local Area Connection
  1   50 4294967295  connected    Loopback Pseudo-Interface 1
```

From this, we can see that our physical interface Local Area Connection has index number 2 (first column). Let's use this information to set the TCP/IP configuration for this interface. Here's what we want the settings to be:

- IP address: 172.16.11.162
- Subnet mask: 255.255.255.0
- Default gateway: 172.16.11.1
- Primary DNS server: 172.16.11.161
- Secondary DNS server: none

To do this, we can use two netsh.exe commands as follows:

```
C:\Windows\System32>netsh interface ipv4 set address name="2" source=static
address=172.16.11.162 mask=255.255.255.0 gateway=172.16.11.1

C:\Windows\System32>netsh interface ipv4 add dnsserver name="2" address=172.16.11.161
index=1
```

Now let's run **ipconfig /all** again and check the result:

```
C:\Windows\System32>ipconfig /all
Windows IP Configuration

   Host Name . . . . . . . . . . . . : LH-3TBCQ4I1ONRA
   Primary Dns Suffix  . . . . . . . :
   Node Type . . . . . . . . . . . . : Hybrid
   IP Routing Enabled. . . . . . . . : No
   WINS Proxy Enabled. . . . . . . . : No

Ethernet adapter Local Area Connection:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Intel 21140-Based PCI Fast Ethernet Adapter
(Emulated)
   Physical Address. . . . . . . . . : 00-03-FF-27-88-8C
   DHCP Enabled. . . . . . . . . . . : No
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::c25:d049:5b0c:1585%2(Preferred)
```

```
   IPv4 Address. . . . . . . . . . . : 172.16.11.162(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 172.16.11.1
   DNS Servers . . . . . . . . . . . : 172.16.11.161
   NetBIOS over Tcpip. . . . . . . . : Enabled

Tunnel adapter Local Area Connection*:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : isatap.{B4B31F3D-B6C8-4303-BA3C-5A54B05F2FDD}
   Physical Address. . . . . . . . . : 00-00-00-00-00-00-00-E0
   DHCP Enabled. . . . . . . . . . . : No
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::5efe:172.16.11.162%3(Preferred)
   Default Gateway . . . . . . . . . :
   DNS Servers . . . . . . . . . . . : 172.16.11.161
   NetBIOS over Tcpip. . . . . . . . : Disabled
```

So far, so good. Let's move on.

Changing the Server's Name

Next let's change the name of our server. When you install a Windows server core server manually from media, the server is assigned a randomly generated name. We want to change that, and we can use netdom.exe to do this. First let's see what the current name is, and then let's change it to DNSSRV because we're planning on using this particular machine as a DNS server on our network:

```
C:\Windows\System32>hostname
LH-3TBCQ4I1ONRA

C:\Windows\System32>netdom renamecomputer %computername% /NewName:DNSSRV
This operation will rename the computer LH-3TBCQ4I1ONRA
to DNSSRV.

Certain services, such as the Certificate Authority, rely on a fixed machine
name. If any services of this type are running on LH-3TBCQ4I1ONRA,
then a computer name change would have an adverse impact.

Do you want to proceed (Y or N)?
y
The computer needs to be restarted in order to complete the operation.

The command completed successfully.
```

We can restart the server using the **shutdown /r /t 0** command. Once the machine is restarted, typing **hostname** shows that the server's name has been successfully changed:

```
C:\Windows\System32>hostname
DNSSRV
```

Joining a Domain

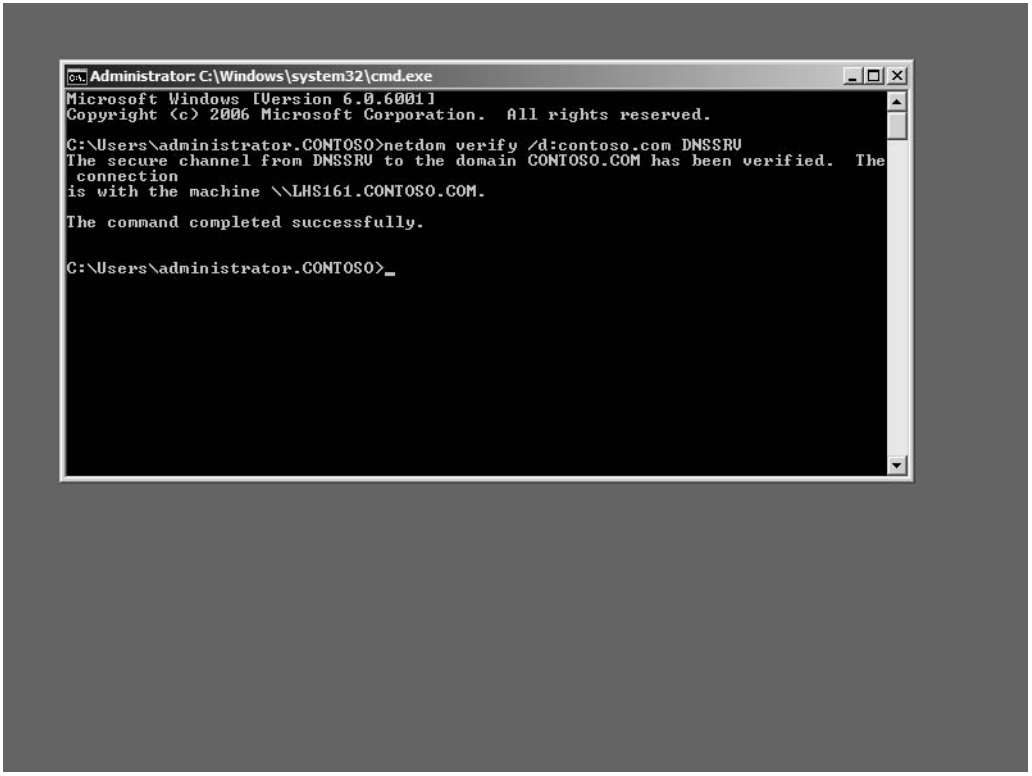
Now let's join our server to our domain. We'll use netdom.exe again to do this, and we're going to join our server to a domain named contoso.com. Here's how we do this:

```
C:\Windows\System32>netdom join DNSSRV /domain:CONTOSO /userd:Administrator /
passwordd:*
Type the password associated with the domain user:

The computer needs to be restarted in order to complete the operation.

The command completed successfully.
```

Again, we'll use **shutdown /r /t 0** to restart the machine. Once it's restarted, we'll log on as a domain admin this time and use netdom.exe again to verify that our server has established a secure channel to the domain controller.



Activating the Server

To activate our server, we can use a built-in script named slmgr.vbs found in the %windir%\System32 directory. (This script is also in Windows Vista and in full installations of Windows Server 2008, and it can be run remotely from those platforms to activate a Windows server core installation.) Typing **cscript slmgr.vbs /?** shows the available syntax for this command:

```
C:\Windows\System32>cscript slmgr.vbs /?
Windows Software Licensing Management Tool
Usage: slmgr.vbs [MachineName [User Password]] [<Option>]
          MachineName: Name of remote machine (default is local machine)
          User:        Account with required privilege on remote machine
          Password:    password for the previous account

Global Options:
-ipk <Product Key>
    Install product key (replaces existing key)
-upk
    Uninstall product key
-ato
    Activate Windows
-dli [Activation ID | All]
    Display license information (default: current license)
-dlv [Activation ID | All]
    Display detailed license information (default: current license)
-xpr
    Expiration date for current license state

Advanced Options:
-cpky
    Clear product key from the registry (prevents disclosure attacks)
-ilc <License file>
    Install license
-rilc
    Re-install system license files
-rearm
    Reset the licensing status of the machine
-dti
    Display Installation ID for offline activation
-atp <Confirmation ID>
    Activate product with user-provided Confirmation ID
```

Let's first use the **–xpr** option to display the expiration date for the current license state:

```
C:\Windows\system32>cscript slmgr.vbs -xpr
Microsoft (R) Windows Script Host Version 5.7
Copyright (C) Microsoft Corporation. All rights reserved.

Initial grace period ends 3/31/2007 1:13:00 AM
```

Now let's use **-dli** to display more info concerning the server's current license state:

```
C:\Windows\system32>cscript slmgr.vbs -dli
Microsoft (R) Windows Script Host Version 5.7
Copyright (C) Microsoft Corporation. All rights reserved.

Name: Windows(TM) Server 2008, ServerEnterpriseCore edition
Description: Windows Operating System - Windows Server 2008, RETAIL channel
Partial Product Key: XHKDR
License Status: Initial grace period
Time remaining: 14533 minute(s) (10 day(s))
```

Now let's activate the server using the **-ato** option:

```
C:\Windows\system32>cscript slmgr.vbs -ato
Microsoft (R) Windows Script Host Version 5.7
Copyright (C) Microsoft Corporation. All rights reserved.

Activating Windows(TM) Server 2008, ServerEnterpriseCore edition
 (f00d81ce-df2c-47cb-a359-36d652296e56) ...
Product activated successfully.
```

Finally, let's try the **-xpr** and **-dli** options again and see the result:

```
C:\Windows\system32>cscript slmgr.vbs -xpr
Microsoft (R) Windows Script Host Version 5.7
Copyright (C) Microsoft Corporation. All rights reserved.

The machine is permanently activated.


C:\Windows\system32>cscript slmgr.vbs -dli
Microsoft (R) Windows Script Host Version 5.7
Copyright (C) Microsoft Corporation. All rights reserved.

Name: Windows(TM) Server code name "Longhorn", ServerEnterpriseCore edition
Description: Windows Operating System - Server code name "Longhorn", RETAIL channel
Partial Product Key: XHKDR
License Status: Licensed
```

Enabling Automatic Updates

To enable Automatic Updates on our server, we'll use another built-in script named scregedit.wsf. This script is unique to the Windows server core installation option of Windows Server 2008, and it's one of the few binaries on a Windows server core server that is

not found on a full installation of Windows Server 2008. To view the syntax of this script, type cscript scregedit.wsf /? at the command prompt:

```
C:\Windows\System32>cscript scregedit.wsf /?
Microsoft (R) Windows Script Host Version 5.7
Copyright (C) Microsoft Corporation. All rights reserved.

Automatic Updates - Manage Automatic Windows Updates
These settings can be used to configure how Automatic Updates are applied to the
Windows system. It includes the ability to disable automatic updates and to set the
installation schedule.

/AU [/v][value]

    /v    View the current Automatic Update settings
    value    value you want to set to.

    Options:
    4 - Enable Automatic Updates
    1 - Disable Automatic Updates


Windows Error Reporting Settings
Windows can send descriptions of problems on this server to Microsoft. If you choose
to automatically send generic information about a problem, Microsoft will use the
information to start working on a solution.

This setting might be overridden by the following Group Policy:
    Key : Software\Policies\Microsoft\Windows\Windows Error Reporting\Consent,
          Value : DefaultConsent

/ER [/v][value]
    /v    View the current Windows Error Reporting settings
    value    value you want to set to.

Opt-in Settings:
   2 - Automatically send summary reports (Recommended)
   3 - Automatically send detailed reports
   1 - Disable Windows Error Reporting

For more information on what data information is collected, go to
http://go.microsoft.com/fwlink/?linkid=50163


Terminal Service - Allow Remote Administration Connections
This allows administrators to connect remotely for administration purposes.

/AR [/v][value]

    /v    View the Remote Terminal Service Connection setting
    value    (0 = enabled, 1 = disabled)


Terminal Service - Allow connections from previous versions of Windows
```

```
This setting configures CredSSP based user authentication for Terminal Service
connections

/CS  [/v][value]

    /v    View the Terminal Service CredSSP setting
    value    (0 = allow previous versions, 1 = require CredSSP)


IP Security (IPSEC) Monitor - allow remote management
This setting configures the server to allow the IP Security (IPSEC) Monitor to be able
to remotely manage IPSEC.

/IM [/v][value]

    /v    View the IPSEC Monitor setting
    value    (0 = do not allow, 1 = allow remote management)


DNS SRV priority - changes the priority for DNS SRV records
This setting configures the priority for DNS SRV records and is only useful on Domain
Controllers.
For more information on this setting, search TechNet for LdapSrvPriority

/DP [/v][value]

    /v    View the DNS SRV priority setting
    value    (value from 0 through 65535. The recommended value is 200.)


DNS SRV weight - changes the weight for DNS SRV records
This setting configures the weight for DNS SRV records and is useful only on Domain
Controllers.
For more information on this setting, search TechNet for LdapSrvWeight

/DW [/v][value]

    /v    View the DNS SRV weight setting
    value    (value from 0 through 65535. The recommended value is 50.)


Command Line Reference
This setting displays a list of common tasks and how to perform them from the command
line.

/CLI
```

First let's see what the current setting for Automatic Updates is on the machine:

```
C:\Windows\system32>cscript scregedit.wsf /au /v
Microsoft (R) Windows Script Host Version 5.7
Copyright (C) Microsoft Corporation. All rights reserved.

SOFTWARE\Microsoft\Windows\CurrentVersion\WindowsUpdate\Auto Update AUOptions
Value not set.
```

Looks like Automatic Updates is not yet configured, so let's enable it:

```
C:\Windows\system32>cscript scregedit.wsf /au 4
Microsoft (R) Windows Script Host Version 5.7
Copyright (C) Microsoft Corporation. All rights reserved.

Registry has been updated.
```

Now let's verify by using our previous command:

```
C:\Windows\system32>cscript scregedit.wsf /au /v
Microsoft (R) Windows Script Host Version 5.7
Copyright (C) Microsoft Corporation. All rights reserved.

SOFTWARE\Microsoft\Windows\CurrentVersion\WindowsUpdate\Auto Update AUOptions
View registry setting.
4
```

Note that on a Windows server core server you can configure Automatic Updates only to download and install updates automatically. You can't configure it to download updates and prompt you to install them later.

There are other initial configuration tasks we could do, but let's move on. Actually, let's hear first from one of our experts concerning a configuration task that's *not* easy to do from the command line:

From the Experts: Configuring Display Resolution

Although there is no tool on a Windows server core server to allow you to change your display resolution, you can configure this by using an unattend file. However, it is possible to change the display resolution so that you can run at a higher resolution than what you might have ended up with at the end of setup. Doing this requires editing the registry; however, if you pick a resolution your video card or monitor cannot display, you might have to reinstall—although you should still be able to boot and remotely modify the settings in the registry.

To do this, you need to open regedit.exe and navigate to the following location:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Video

Under this will be a list of GUIDs, and you need to determine which one corresponds to your video card/driver. You might have to experiment to determine the right one. Under the GUID, you can set

\0000\DefaultSettings.XResolution

\0000\DefaultSettings.YResolution

to the resolution you would like to use. If these don't exist, you can create them. You must log off and log back on again for the change to take effect. Be careful doing this because if you specify an unsupported display resolution, you might need to reinstall your machine or remotely connect to the registry from another computer to change it, and remotely reboot.

–Andrew Mason
Program Manager, Windows Server

Managing a Windows Server Core Server

Once we've performed initial configuration of our Windows server core server, we can then add roles and optional features so that it can provide needed functionality to our network. In this section, we're going to examine how to perform such common tasks, and we'll also look at different ways of managing a Windows server core server, including using the following:

- Local administration from the command prompt
- Remote administration using Terminal Services
- Remote administration using Remote Server Administration Tools
- Remote administration using Group Policy
- Remote administration using WinRM/WinRS

Local Management from the Command Line

When we log on to the console of a Windows server core server, a command prompt appears. From this command prompt, we can do a lot of things:

- Run common tools such as netsh.exe and netdom.exe to perform various tasks, as we saw previously.
- Use special tools such as oclist.exe and ocsetup.exe to install roles and optional features on our server to give it more functionality.

- Run in-box scripts such as slmgr.vbs and scregedit.wsf, as we saw earlier, to perform certain kinds of tasks.
- Create our own scripts using Notepad, and run them using Cscript.exe and the supported WMI providers.
- Use the WMI command line (WMIC) to do almost anything from the command line that you can do by writing WMI scripts.

As we mentioned before, however, one thing you can't do is run PowerShell commands to administer your server. The reason for this omission is that PowerShell is managed code that requires the .NET Framework in order to work, and the .NET Framework is not included in the Windows server core installation option. Why? Because the .NET Framework has dependencies across the whole spectrum of different Windows components, and leaving it in would have increased the size of the Windows server core installation option until it was very nearly the size of a full installation of Windows Server 2008. For future versions of the Windows server core installation, however, a slimmed-down .NET Framework might be available that can provide PowerShell cmdlet functionality without the need of increasing the footprint significantly. But we'll have to see, as that's something that would happen after RTM. Note that you can however use PowerShell remotely to manage a Windows server core installation if the script strictly uses only WMI commands and not cmdlets.

Let's look how to perform two important tasks from the command line: adding server roles and adding optional features.

Installing Roles

Let's start by seeing what roles are currently installed on our server and what roles are available to install. We'll use the oclist.exe command to do this:

```
C:\Windows\System32\>oclist
Use the listed update names with Ocsetup.exe to install/uninstall a server role or
optional feature.

Adding or removing the Active Directory role with OCSetup.exe is not supported. It can
leave your server in an unstable state. Always use DCPromo to install or uninstall
Active Directory.

================================================================================
Microsoft-Windows-ServerCore-Package
================================================================================
Not Installed:BitLocker
Not Installed:BitLocker-RemoteAdminTool
Not Installed:ClientForNFS-Base
Not Installed:DFSN-Server
Not Installed:DFSR-Infrastructure-ServerEdition
Not Installed:DHCPServerCore
Not Installed:DirectoryServices-ADAM-ServerCore
Not Installed:DirectoryServices-DomainController-ServerFoundation
```

```
Not Installed:DNS-Server-Core-Role
Not Installed:FailoverCluster-Core
Not Installed:FRS-Infrastructure
Not Installed:MediaServer
Not Installed:Microsoft-Windows-MultipathIo
Not Installed:Microsoft-Windows-RemovableStorageManagementCore
Not Installed:NetworkLoadBalancingHeadlessServer
Not Installed:Printing-ServerCore-Role
    |
    |--- Not Installed:Printing-LPDPrintService
    |
Not Installed:ServerForNFS-Base
Not Installed:SIS
Not Installed:SNMP-SC
Not Installed:SUACore
Not Installed:TelnetClient
Not Installed:WindowsServerBackup
Not Installed:WINS-SC
```

Note that the oclist.exe command displays information about both roles and features installed and not installed on the machine. We can see from the command output that the DNS Server role is not presently installed on the machine. We can also verify this by typing **net start** in the command line:

```
C:\Windows\System32>net start
These Windows services are started:

   Application Experience
   Background Intelligent Transfer Service
   Base Filtering Engine
   COM+ Event System
   Computer Browser
   Cryptographic Services
   DCOM Server Process Launcher
   DHCP Client
   Diagnostic Policy Service
   Diagnostic System Host
   Distributed Transaction Coordinator
   DNS Client
   Group Policy Client
   IKE and AuthIP IPsec Keying Modules...
```

In fact, the only DNS binaries presently installed are those for the DNS client:

```
C:\Windows\System32>dir dns*.*
Volume in drive C has no label.
 Volume Serial Number is FC68-BDF4

 Directory of C:\Windows\system32

02/09/2007  10:00 PM           163,840 dnsapi.dll
02/09/2007  09:59 PM            24,064 dnscacheugc.exe
02/09/2007  10:00 PM            84,480 dnsrslvr.dll
               3 File(s)        272,384 bytes
               0 Dir(s)  27,578,523,648 bytes free
```

Now let's install the DNS Server role using the ocsetup.exe command as follows:

```
C:\Windows\System32>start /w ocsetup DNS-Server-Core-Role
```

After a short while, the command prompt appears again. The reason we used the /**w** switch with **start** is because that way control is not returned to the command prompt until the **ocsetup** command finishes its work. (By the way, note that **ocsetup** is case sensitive.) Now if we type **oclist**, we should see that the DNS Server role has been added to our server:

```
C:\Windows\System32\>oclist
...
Not Installed:DirectoryServices-ADAM-ServerCore
Not Installed:DirectoryServices-DomainController-ServerFoundation
    Installed:DNS-Server-Core-Role
Not Installed:FailoverCluster-Core
Not Installed:FRS-Infrastructure
...
```

We can also see that three additional binaries for DNS are now present on the server:

```
C:\Windows\System32>dir dns*.*
Volume in drive C has no label.
 Volume Serial Number is FC68-BDF4

 Directory of C:\Windows\system32

03/20/2007  11:59 PM    <DIR>          dns
02/09/2007  11:42 AM           484,864 dns.exe
02/09/2007  10:00 PM           163,840 dnsapi.dll
02/09/2007  09:59 PM            24,064 dnscacheugc.exe
02/09/2007  11:42 AM           162,816 dnscmd.exe
02/09/2007  11:42 AM            13,312 dnsperf.dll
02/09/2007  10:00 PM            84,480 dnsrslvr.dll
               6 File(s)        933,376 bytes
               1 Dir(s)  27,576,926,208 bytes free
```

And if we type **net stop dns**, we can now stop the DNS Server service without getting an error because the service is now present on the machine. Now that our machine is a DNS Server, we can use the dnscmd.exe command to further configure this role if we want from the command line.

Installing other server roles is similar to what we just did and uses the ocsetup.exe command, with the exception being that the process installs the Active Directory role. This is because Dcpromo.exe in Windows Server 2008 now installs the Active Directory binaries during promotion and uninstalls the binaries during demotion, so you should *not* use ocsetup.exe to add or remove the Active Directory role as then the promotion/demotion will not take place and your server may not function correctly.

Anyway, to add or remove the Active Directory role, you therefore have to use the dcpromo.exe tool, but you also have to run it in unattended mode because the GUI form of this tool (the Active Directory Installation Wizard) can't run on a Windows server core server because of the lack of a desktop shell to run it in. The syntax for running dcpromo.exe in unattended mode is **dcpromp /unattend:unattend.txt**, and a sample unattend.txt file you could use (or further customize) for doing this is as follows:

```
[DCInstall]
ReplicaOrNewDomain = Domain
NewDomain=Forest
NewDomainDNSName = contoso.com
AutoConfigDNS=Yes
DNSDelegation=Yes
DNSDelegationUserName=dnsuser
DNSDelegationPassword=p@ssword!
RebootOnSuccess = NoAndNoPromptEither
SafeModeAdminPassword = p@ssword!
```

For more information on using dcpromo in unattended mode, type **dcpromo /?:unattend** at the command prompt.

Installing Optional Features

Installing optional features is very similar to installing roles. Type **oclist** to display a list of installed and uninstalled features and to determine the internal name of each feature. For example, the Failover Cluster feature is named FailoverCluster-Core, and we need to use this internal form of the name when we run ocsetup to install this feature. You can also remove features by adding an **/uninstall** switch to your **ocsetup** command. You can remote roles that way too, but be sure to stop the role's services before you remove the role.

Other Common Management Tasks

There are lots of other common management tasks you might need to perform on a Windows server core server. The following is just a sampling of some of these tasks.

First, you can add new hardware to your server. Windows server core servers include support for Plug and Play. So if your new device is PnP and there's an in-box driver available for your device, you can just plug the device in and the server will recognize it and automatically install a driver for it. But we did mention earlier that the Windows server core server installation option of Windows Server 2008 does not include that many in-box drivers. So what do you do if your device is not supported by an in-box driver because of its date of manufacture? In that case, follow this procedure:

1. Copy the driver files from the driver media for the device to a temporary directory on your server.
2. Change your current directory to this temporary directory, and type **pnputil –i –a <driver>.inf** at the command prompt.
3. Reboot your server if prompted to do so.

Note that if you want to find what drivers are currently installed on your server, you can type **sc query type= driver** at a command prompt.

What if you want to install some application on your server? First of all, beware—any application that has a GUI might not function properly when you install it. Obviously, that means we can't install Microsoft Exchange Server, Microsoft SQL Server, or other Windows Server System products on a Windows server core server, because these products all have GUI management tools (and more importantly, a Windows server core server is missing a lot of components needed by these products such as the .NET Framework for running managed code).

What kinds of applications might you want to install on a Windows server core server? The usual stuff—antivirus agents, network backup agents, system management agents, and so on. Most agents like this are GUI-less and should install fine and work properly on a Windows server core server. And the Windows Installer service is yet another feature that's still present on a Windows server core server—and if you need to install an agent manually, you should try and do so in quiet mode using msiexec.exe with the **/qb** switch to display the basic UI only. For example, you can do this by typing **msiexec /qb <package>** at the command prompt.

If you need to configure Windows Firewall, the NAP client, or your server's IPSec configuration, you can use netsh.exe to do this. I won't go into all the details here, as you can just check TechNet for the proper netsh.exe syntax to use for each task.

What about patch management? We already described how to enable Automatic Updates on the server, and if you have Windows Server Update Service (WSUS) deployed, you can manage patches for your server using that as well. For Windows server core servers that you want

to manually perform patch management on, however, you can use the wusa.exe command to install and remove patches from the command prompt. To do this, first download the patch from Windows Update and expand to get the .msu file. Then copy the .msu file to your server, and type **wsua <patch>.msu /quiet** at the command prompt to install the patch. You can also remove installed patches from your server by typing **pkgmgr /up /m:<package>.cab /quiet** at the command prompt.

Let's hear more about patch management on a Windows server core installation of Windows Server 2008 from one of our experts:

From the Experts: Servicing Windows Server Core

When using Windows server core, the new minimal installation option for Windows Server 2008, a common topic of discussion is servicing. First a little background and then some methods to make dealing with patches easier.

With Windows Server 2008, each patch that is released contains a set of applicability rules. When a patch is sent to a server, either by Windows Update or another automated servicing tool, the servicing infrastructure examines the patch to determine if it applies to the system based on the applicability rules. If not, it is ignored and nothing is changed on the server.

If you have already downloaded a set of patches and want to determine if they apply to a Windows server core installation, you can do the following:

1. Run **wusa <patch_name>**.
2. If the dialog box that appears asks if you want to apply the patch, click No. This means that the patch applies, and you should move on to the next step. Otherwise, the dialog box will state that the patch doesn't apply and you can ignore the patch.
3. Run **wusa <patch_name> /quiet** to apply the patch.

After applying patches, you can run either the wmic qfe command or systeminfo.exe to see what patches are installed.

–Andrew Mason
Program Manager, Windows Server

What else can you do in terms of managing your Windows server core installation of Windows Server 2008? Lots! For example, if you need to manage your disks and file system on your server, you can use commands such as diskpart, defrag, fsutil, vssadmin, and so on. And if you need to manage permissions and ownership of files, you can use icacls.

You can also manage your event logs from the command line using the wevtutil.exe command, which is new in Windows Vista and Windows Server 2008. This powerful command can be used to query your event logs for specific events and to export,

archive, clear, and configure your event logs as well. For example, to query your System log for the most recent occurrence of a shutdown event having source USER32 and event ID 1074, you can do this:

```
C:\Windows\system32>wevtutil qe System /c:1 /rd:true /f:text /
q:*[System[(EventID=1074)]]
Event[0]:
  Log Name: System
  Source: USER32
  Date: 2007-03-20T22:26:36.000
  Event ID: 1074
  Task: N/A
  Level: Information
  Opcode: N/A
  Keyword: Classic
  User: S-1-5-21-3620207985-2970159875-1752314906-500
  User Name: DNSSRV\Administrator
  Computer: DNSSRV
  Description:
The process C:\Windows\system32\shutdown.exe (DNSSRV) has initiated the restart of
computer DNSSRV on behalf of user DNSSRV\Administrator for the following reason: No
title for this reason could be found
 Reason Code: 0x840000ff
 Shutdown Type: restart
 Comment:
```

To create and manage data collectors for performance monitoring, you can use the logman.exe command. You can also use the relog.exe command to convert a performance log file into a different format or change its sampling rate. And you can use the tracerpt.exe command to create a remote from a log file or a real-time stream of performance-monitoring data.

To manage services, you can use the sc command, which is a very powerful command that provides even more functionality than the Services.msc snap-in.

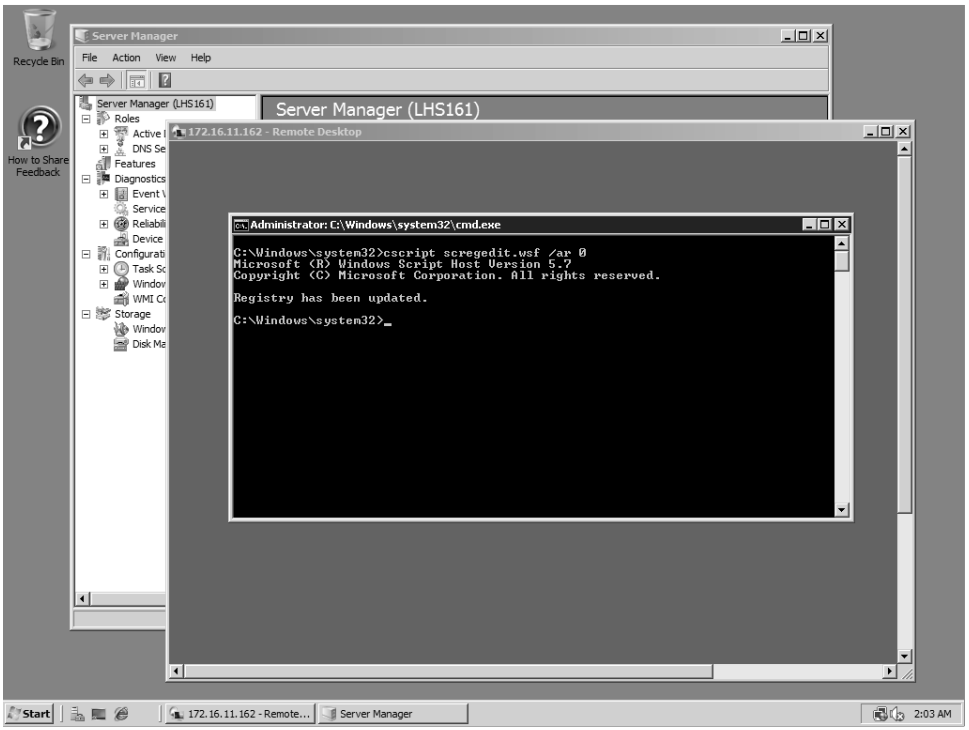
What else can you do? Lots. Let's move on now to remote management.

Remote Management Using Terminal Services

You can also manage Windows server core servers from another computer using Terminal Services. To do this, you first have to enable Remote Desktop on your server, and because we can't right-click on Computer and select Properties to do this, we'll have to find another way. Here's how—use the scregedit.wsf script we looked at previously. The syntax for performing this task is **cscript scregedit.wsf /ar 0** to enable Remote Desktop and **cscript scregedit.wsf /ar 1** to disable it again. To view your current Remote Desktop settings, type **cscript scregedit.wsf /ar /v** at a command prompt. Note that in order to allow pre-Windows Vista

versions of the TS client to connect to a Windows server core installation, you need to disable the enhanced security by running the **cscript scregedit.wsf /cs 0** command.

Once you've enabled Remote Desktop like this, you can connect to your Windows server core server from another machine using Remote Desktop Connection (mstsc.exe) and manage it as if you were logged on interactively at your server's console. In this figure I'm logged on to a full installation of Windows Server 2008 and have a Terminal Services session open to my remote Windows server core server to manage it.



There's more! Later in Chapter 8, "Terminal Services Enhancements," we'll describe a new feature of Terminal Services in Windows Server 2008 that lets you remote individual application windows instead of entire desktops. Let's hear now from one of our experts concerning how this new Terminal Services functionality can be used to make managing Windows server core servers easier.

From the Experts: Enabling Remote Command Line Access on Server Core

There are several ways to administer a Windows server core installation, ranging from using the local console to remote administration from a full Windows Server 2008 server using MMC. A really cool mechanism is to manage the Windows server core installation using Terminal Services RemoteApp to make the command line console available. This allows command-line administration without having to be physically present at the box, and without having a full-blown terminal server session. (After all, a Windows server core installation does not need the full desktop; it just needs the console, and Terminal Services RemoteApp is perfect for this.) A full Windows Server 2008 machine is necessary, along with the Windows server core installation that is to be administered.

On the Windows Server 2008 machine, add the Terminal Server Role using the Server Manager administrative tool. Only the Terminal Server role itself is needed, not the TS Licensing role, TS Session Broker role, TS Gateway role, or TS Web Access role. After the TS role is installed, start MMC and add the TS RemoteApp Manager snap-in, providing the name of the Windows server core machine to the snap-in. Once the snap-in is installed, connect to the Windows server core machine and click Add Remote Apps. Navigate to the %SYSTEMROOT%\System32 folder using the administrative share, select cmd.exe, and complete the wizard. Select the cmd.exe entry in the RemoteApp pane, click Create .rdp File, and follow the wizard to save the RDP file. Ensure that TS is enabled on the Windows server core machine. (Use the scregedit.wsf script.) You can now copy the RDP file to any client machine and connect to the Windows server core installation through it. The console will be integrated into the task bar of the client, like a local application. For more information on Terminal Services and TS RemoteApp, please see Chapter, "Terminal Services Enhancements."

–Rahul Prasad
Software Development Engineer, Windows Core Operating System Division

And here's another expert from the product team at Microsoft sharing some additional tips on managing Windows server core servers using Terminal Services:

From the Experts: Tips for Using Terminal Services with Windows Server Core

When you're using Terminal Services in a Windows server core server without the GUI shell, some common tasks require you to do things a little differently.

Logging off of a Terminal Services Session

On a Windows server core server, there is no Start button and therefore no GUI option to log off. Clicking the X in the corner of the Terminal Services window disconnects your

session, but the session will still be using resources on the server. To log off, you need to use the Terminal Services logoff command. While in your Terminal Services session, you simply run logoff. If you disconnect your session, you can either reconnect and use logoff, use the logoff command remotely, or use the Terminal Services MMC to log off the session.

Restarting the Command Prompt

When logged on locally, if you accidentally close the command prompt you can either log off and log on, or press CTRL+ALT+DEL, start Task Manager (or just press CTRL+SHIFT+ESC), click file, and run cmd.exe to restart it. You can also configure the Terminal Services client to have the Windows keys pass to the remote session when not maximized so that you can use CTRL+SHIFT+ESC to start task manager and run cmd.exe.

Working with Terminal Services Sessions

If you ever need to manage Terminal Services sessions from the command line, the query command is the tool to use. Running query sessions (which can also be used remotely) will tell you what Terminal Services sessions are active on the box, as well as who is logged in to them. This is handy if you need to restart the box and want to know if any other administrators are logged on. Query has some other useful options, and there are a variety of other Terminal Services command-line tools.

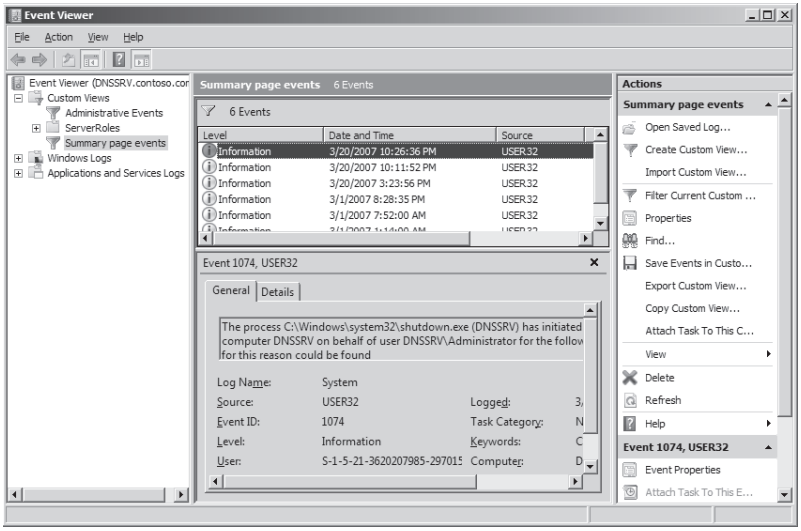
–Andrew Mason
Program Manager, Windows Server

Remote Management Using the Remote Server Administration Tools

Although you can manage file systems, event logs, performance logs, device drivers, and other aspects from the command line, there's no law that says you have to. For example, the syntax for wvetutil.exe is quite complex to learn and understand, especially if you want to use this tool to query event logs for specific types of events. It would be nice if you could just use Event Viewer to display, query, and filter your event logs on a Windows server core server. You can! But you have to do it remotely from another computer running either Windows Vista or Windows Server 2008 and with the appropriate Remote Server Administration Tools (RSAT) installed on it.

We talked about RSAT earlier in Chapter 4, "Managing Windows Server 2008," and it's basically the Windows Server 2008 equivalent of the Adminpak.msi server tools on previous versions of Windows Server. So if you want to use MMC snap-in tools to administer a Windows server core server from a Windows Vista computer or a machine running a full installation of Windows Server 2008, you might or might not need to install the RSAT on this machine because both Windows Vista and full installations of Windows Server 2008 already include many MMC snap-in tools that can be accessed from the Start menu using Administrative

Tools. Event Viewer is one such built-in tool, and here it is running on a full installation of Windows Server 2008, showing the previously mentioned shutdown event in the System event log on our remote Windows server core server.



Remote Administration Using Group Policy

Another way of remotely administering Windows server core servers is by using Group Policy. For example, although the **netsh advfirewall** context commands can be used to configure Windows Firewall, doing it this way can be tedious. It's much easier to use the following policy setting:

Computer Configuration\Windows Settings\Security Settings\Windows Firewall With Advanced Security

By creating a GPO that targets your Windows server core servers, either by placing these servers in an OU and linking the GPO to that OU or by using a WMI filter to target the GPO only at Windows server core servers, you can remotely configure Windows Firewall on these machines using Group Policy. For example, you can use the *OperatingSystemSKU* property of the *Win32_OperatingSystem* WMI class to determine whether a given system is running a Windows server core installation of Windows Server 2008 by checking for the following return values:

- 12 – Datacenter Server Core Edition
- 13 – Standard Server Core Edition
- 14 – Enterprise Server Core Edition

You can use this property in creating a WMI filter that causes a GPO to target only Windows server core servers.

Remote Management Using WinRM/WinRS

Finally, you can also manage Windows server core servers remotely using the Windows Remote Shell (WinRS) included in Windows Vista and the full installation of Windows Server 2008. WinRS uses Windows Remote Management (WinRM), which is Microsoft's implementation of the WS-Management protocol developed by the Desktop Management Task Force (DMTF). WinRM was first included in Windows Server 2003 R2 and has been enhanced in Windows Vista and Windows Server 2008.

To use the Windows Remote Shell to manage a Windows server core server, log on to the Windows server core server you want to remotely manage and type **WinRM quickconfig** at the command prompt to create a WinRM listener on the machine:

```
C:\Windows\System32>WinRM quickconfig
WinRM is not set up to allow remote access to this machine for management.
The following changes must be made:

Create a WinRM listener on HTTP://* to accept WS-Man requests to any IP on this
machine.

Make these changes [y/n]? y

WinRM has been updated for remote management.

Created a WinRM listener on HTTP://* to accept WS-Man requests to any IP on this
machine.
```

Now on a different machine running either Windows Vista or the full installation of Windows Server 2008, type **winrs –r:<server_name> <command>**, where <server_name> is your Windows server core server and <command> is the command you want to execute on your remote server. Here's an example of the Windows Remote Shell at work:

```
C:\Users\Administrator>winrs -r:DNSSRV "cscript C:\Windows\System32\slmgr.vbs -dli"
Microsoft (R) Windows Script Host Version 5.7
Copyright (C) Microsoft Corporation. All rights reserved.

Name: Windows(TM) Server Windows Server 2008, ServerEnterpriseCore edition
Description: Windows Operating System - Windows Server 2008, RETAIL channel
Partial Product Key: XHKDR
License Status: Licensed
```

You can also run WinRM quickconfig during unattended installation by configuring the appropriate answer file setting for this service.

Windows Server Core Installation Tips and Tricks

Finally, let's conclude this chapter with a list of 101 things (well, not really 101) you might want to know about or do with a Windows server core installation of Windows Server 2008. Some of these are tips or tricks for configuring or managing a Windows server core server; others are just things you might want to make note of. They're all either interesting, useful, or both. Here goes....

First, if you want quick examples of a whole lot of administrative tasks you can perform from the command line, just type **cscript scregedit.wsf /cli** at the command prompt:

```
C:\Windows\System32\>cscript scregedit.wsf /cli
Microsoft (R) Windows Script Host Version 5.7
Copyright (C) Microsoft Corporation. All rights reserved.

To activate:
     Cscript slmgr.vbs -ato

To use KMS volume licensing for activation:
     Configure KMS volume licensing:
          cscript slmgr.vbs -ipk [volume license key]
     Activate KMS licensing

          cscript slmgr.vbs -ato
     Set KMS DNS SRV record
          cscript slmgr.vbs -skma [KMS FQDN]
Determine the computer name, any of the following:
     Set c

     Ipconfig /all
     Systeminfo

Rename the Server Core computer:
     Domain joined:
          Netdom renamecomputer %computername% /NewName:new-name
  /UserD:domain-username /PasswordD:*

     Not domain joined:
          Netdom renamecomputer %computername% /NewName:new-name

Changing workgroups:
     Wmic computersystem where name="%computername%" call
  joindomainorworkgroup name="[new workgroup name]"

Install a role or optional feature:
     Start /w Ocsetup [packagename]

     Note: For Active Directory, run Dcpromo with an answer file.
View role and optional feature package names and current installation state:
     oclist
Start task manager hot-key:
     ctrl-shift-esc
```

```
Logoff of a Terminal Services session:
     Logoff

To set the pagefile size:
     Disable system pagefile management:
          wmic computersystem where name="%computername%" set
  AutomaticManagedPagefile=False

     Configure the pagefile:
          wmic pagefileset where name="C:\\pagefile.sys" set
  InitialSize=500,MaximumSize=1000
Configure the timezone, date, or time:

     control timedate.cpl
Configure regional and language options:

     control intl.cpl
Manually install a management tool or agent:
     Msiexec.exe /i [msipackage]

List installed msi applications:
     Wmic

     product
Uninstall msi applications:

     Wmic product get name /value
     Wmic product where name="[name]" call uninstall
To list installed drivers:
     Sc query type= driver
Install a driver that is not included:
     Copy the driver files to Server Core
     Pnputil –i –a [path]\[driver].inf
Determine a file's version:
     wmic datafile where name="d:\\windows\\system32\\ntdll.dll" get version
List of installed patches:
     wmic qfe list
Install a patch:
     Wusa.exe [patchame].msu /quiet
Configure a proxy:
     Netsh winhttp proxy set [proxy_name]:[port]
Add, delete, query a Registry value:
     reg.exe add /?
     reg.exe delete /?
     reg.exe query /?
```

Now here are a bunch of random insights into and tips for running a Windows server core installation of Windows Server 2008:

The SMS 2005 and MOM 2005 agents should run fine on Windows server core servers, but for best systems management functionality you probably want to use the upcoming Microsoft System Center family of products instead.

You can deploy the Windows server core installation option using Windows Deployment Services (WDS) just like the full installation option of Windows Server 2008. It's the same product—just a different setup option to choose.

To install the Windows server core installation option on a system, the system needs a minimum of 512 MB RAM. That's not because Windows server core servers need that much RAM, however—in fact, they need just over 100 MB of RAM to run with no roles installed. But the setup program for installing Windows Server 2008 requires 512 MB or more of memory or setup will fail. You *can* install the Windows server core installation option on a box with 512 MB RAM and then after installation pull some of the RAM, but at the time of this writing, this procedure is not supported.

The Windows server core installation option uses much less disk space than a full installation of Windows Server 2008. We're talking roughly 1 MB vs. 5 MB here, and that shows you how much stuff has been pulled out of Windows server core to slim it down.

When patching Windows server core servers, you actually don't need to presort patches into those that apply to the Windows server core installation option and those that don't apply. Instead, you can just go ahead and patch, and only updates that apply to Windows server core servers will actually be applied.

You can manage Windows server core servers remotely using the RSAT, but you can't install the RSAT on Windows server core to manage the server locally.

The Windows server core installation option does support Read Only Domain Controllers (RO DC). This support makes Windows server core servers ideal for branch office scenarios, especially with BitLocker installed as well.

You won't get any User Account Control (UAC) prompts if you log on to a Windows server core server as a nonadministrator and try to perform an administrative task. Why not? UAC needs the desktop shell to function.

One way of seeing how slimmed-down Windows server core is is to compare the number of installed and running services on the two platforms. Table 6-3 shows a rough comparison, assuming no roles have been installed.

Table 6-3 Comparison of default number of services for server core installation vs. full installation

Feature compared	Server core	Server
Number of services installed by default	~40	~75
Number of services running by default	~30	~50

If you're trying to run the Windows Remote Shell from another machine and use it to manage a Windows server core server and it doesn't work, you might not have the right credentials on the Windows server core server to manage it. If this is the case, first try connecting to the

Windows server core server from your machine using the **net use \\<server_name>\ipc$ /u:<domain>\<user_name>** command using a user account that has local admin privileges on the Windows server core server. Then try running your WinRS commands again. Note that this tip also applies to using MMC admin tools to remotely manage a Windows server core installation since the MMC doesn't let you specify different credentials for connecting remotely.

If you're trying to use Computer Management on another machine to manage the disk subsystem on your Windows server core server using Disk Management and you can't, type **net start vds** at the command prompt on your Windows server core server to start the Virtual Disk Service on the server. Then you should be able to manage your server's disks remotely using Disk Management.

If you've enabled Automatic Updates on your Windows server core server and you want to check for new software updates immediately, type **wuauclt /detectnow** at the command prompt.

And yes, the Windows server core installation option does support clustering. A clustered file server running on Windows server core servers would be cool.

Our last tip will be provided by one of our experts:

From the Experts: What Time Is It?

Here is a flash back to the old MS-DOS days. Because Windows server core does not have the system tray, there is no clock. If you are used to having the time available on the screen, you can add it to your prompt in the command prompt window.

Entering the following:

```
prompt [$t]$s$p$g
```

will display:

```
[14:27:06.28] C:\users\default>
```

–Andrew Mason
Program Manager, Windows Server

Conclusion

We're used to Microsoft piling features into products, not stripping features out of them. The Windows server core installation option of Windows Server 2008 is a new direction Microsoft is pursuing in its core product line, but it's a direction being driven by customer demand. When I said that Microsoft listened to their customers, I was serious. And Windows server core is a good example of this.

Additional Resources

You'll find a brief description of the Windows server core installation of Windows Server 2008 at *http://www.microsoft.com/windowsserver/Windows Server 2008/evaluation/overview.mspx*. By the time you read this chapter, this page will probably be expanded or the URL will redirect you to somewhere that has a lot more content on the subject.

If you have access to the Windows Server 2008 beta program on Microsoft Connect (*http://connect.microsoft.com*), you can get some great documentation from there, including these:

- Microsoft Windows Server Code Name 2008 Server Core Step-By-Step Guide
- Live Meeting on Server Core
- Live Chat on Server Core

There's also a TechNet Forum where you can ask questions and help others trying out the Windows server core installation option of Windows Server 2008. See *http://forums.microsoft.com/TechNet/ShowForum.aspx?ForumID=582&SiteID=17* for this forum. (Windows Live registration is required.)

There's a Windows server core blog on TechNet that is definitely something you won't want to miss. See *http://blogs.technet.com/server_core/*.

Finally, be sure to turn to Chapter 14, "Additional Resources," for more sources of information concerning the Windows server core installation option, and also for links to webcasts, whitepapers, blogs, newsgroups, and other sources of information about all aspects of Windows Server 2008.

Chapter 7

Active Directory Enhancements

In this chapter:

Active Directory and its related services form the foundation for enterprise networks running Microsoft Windows, and the new features and enhancements to Active Directory and its related services in Windows Server 2008 are numerous. This chapter takes a look at these enhancements and at the direction in which Active Directory and its related services are heading as an integrated identity and access platform for enterprises—that is, as a platform for provisioning and managing network identity.

Understanding Identity and Access in Windows Server 2008

Before we jump in and examine the various enhancements to Active Directory and its related services in Windows Server 2008, however, let's first step back a bit and get the big picture of how Active Directory and its related services have been evolving since they were first introduced in Windows 2000 Server and what these services are becoming in Windows Server 2008 and beyond. It's important to understand this big picture, as otherwise the many improvements to Active Directory and related services in Windows Server 2008 might seem like a miscellaneous grab-bag of changes without much in common. But they have a lot in common as we'll shortly see.

Understanding Identity and Access

So why is identity and access (IDA) important to enterprises? Think for a moment about what goes on when a user on your network needs access to confidential business information stored on a server. Tony is in the Marketing department, and he needs access to a product

specification so that he can work on a marketing presentation for a customer. The document containing the specification is stored on a server on the company's network, and Tony tries to open the document so that he can cut and paste information contained in it into his presentation. To safeguard such specifications, you'd like your IDA infrastructure to do the following:

1. Determine who the user is who wants to use the document.
2. Grant the user the appropriate level of access to the document.
3. Protect confidential information contained in the document.
4. Maintain a record of interaction concerning the user's accessing of the document.

For example, you might want to restrict access to product specifications to full-time employees (FTEs) only and provide read-only access to users in the Marketing department so that they can view but not modify specifications. You might also want to prevent Marketing department users from copying and pasting text from specifications into other documents. And you might want an audit trail showing the day and time that the user accessed the specification.

The challenge of implementing an IDA solution that can do all of this becomes even greater once you start extending the boundaries of your enterprise with "anywhere access" devices, Web services, and collaboration tools like e-mail and instant messaging. It becomes even more complicated once you have to start applying the IDA process not just to FTEs but also to contractors, temps, customers, and external partners. The challenge is to build an IDA solution that can handle all these different scenarios, and Microsoft has steadily been working toward this goal since Active Directory was first released with Windows 2000 Server. Let's briefly summarize the evolution of Microsoft's IDA solution, beginning with Windows 2000 Server and working up to the current platform for Windows Server 2003 R2 and then to Windows Server 2008 and beyond.

Identity and Access in Windows 2000 Server

Active Directory directory service is a Windows-based directory service that was first introduced in Windows 2000 Server. Active Directory directory service stores information about various kinds of objects on a network—such as users, groups, computers, printers, and shared folders—and it makes this information available to users who need to access these resources and administrators who need to manage them. Active Directory provides network users with controlled access to permitted resources anywhere on the network using a single logon process. Active Directory directory service also provides administrators with an intuitive, hierarchical view of the network and its resources, and it provides a single point of administration for all network objects.

Windows 2000 Server also included a separate component, called Certificate Services, that can be used to set up a certificate authority (CA) for issuing digital certificates as part of a Public Key Infrastructure (PKI). These certificates can be used to provide authentication for users and computers on your network to secure e-mail, provide Web-based authentication,

and support smart-card authentication. Certificate Services also provides customizable services for issuing and managing certificates for your enterprise. What's important to understand here is that in Windows 2000 Server, Active Directory directory service and Certificate Services are two separate components that are not integrated together. In other words, the two services are managed separately and have policy implemented differently.

In addition to these two built-in IDA services, Microsoft also released an out-of-band service for Windows 2000 Server called Microsoft Metadirectory Services (MMS). In its final version, MMS 2.2 was an enterprise metadirectory that enterprises could use to integrate all their various directories together into a single consolidated central repository. MMS 2.2 consisted of one or more metadirectory servers, management agents, and the connected directories, and it provided users with access to this consolidated information via Lightweight Directory Access Protocol (LDAP). The goal of MMS 2.2 was to provide enterprises with a provisioning solution that could be used to effectively provide consistent identity management across many different databases and directories. For example, if you had both an Active Directory directory service infrastructure and a Lotus Notes infrastructure and you wanted Active Directory directory service users to be able to look up e-mail addresses from the Lotus Notes directory, MMS 2.2 could make this possible. MMS 2.2 could also simplify the deployment of Active Directory directory service for enterprises that already had information about employees or customers stored in other directories by enabling real-time synchronization of information from these directories into Active Directory directory service. Finally, MMS 2.2 could also be used to simplify the migration and consolidation of multiple directories into Active Directory directory service.

Identity and Access in Windows Server 2003

Although these Windows 2000 Server offerings did meet the needs of some enterprises, they were still provided as separate services and MMS was even a totally separate product. Customers wanted something more integrated, and they also wanted additional IDA features, such as document rights protection and role-based authorization. In addition to making improvements to how Active Directory directory service and Certificate Services work and how they are managed, Microsoft added a new feature called Authorization Manager to Windows 2003 Server that provided role-based authorization for users of line-of-business applications. Although Active Directory directory service by itself provides object-based access control using ACLs, the role-based access control (RBAC) provided by Authorization Manager enables permissions to be managed in terms of the different job roles users might have. Authorization Manager works by providing a set of COM-based runtime interfaces that enables an application to manage and verify a client's requests to perform operations using the application. Authorization Manager also includes an MMC snap-in that application administrators can use to manage different user roles and permissions.

Another IDA service that Microsoft released for Windows Server 2003 is Windows Rights Management Service (RMS), an information-protection technology that works with RMS-enabled applications to help businesses safeguard valuable digital information from

unauthorized use whether online or offline and whether inside the firewall or outside the firewall. Windows RMS was also designed to help organizations comply with a growing number of regulatory requirements that mandated information protection, including the U.S. Sarbanes-Oxley Act, the Gramm-Leach-Bliley Act, the Health Insurance Portability and Accountability Act (HIPAA), and others. To use Windows RMS, enterprises can create centralized custom usage policy templates, such as "Confidential – Read Only," that can work with any RMS-enabled client and can be directly applied to sensitive business information such as financial reports, product specifications, or e-mail messages. Implementing Windows RMS requires an Active Directory directory service infrastructure, a PKI, and Internet Information Services—all of which are included in Windows Server 2003. In addition, RMS-enabled client applications such as Microsoft Office 2003 and Internet Explorer are needed, plus Microsoft SQL Server to provide the underlying database for the service.

While these additional IDA services and add-ons for Active Directory directory service were being released, Microsoft also released a follow-up to MMS 2.2 called Microsoft Identity Integration Server (MIIS) 2003, which provides a centralized service that stores and integrates identity information for organizations with multiple directories. It also provides a unified view of all known identity information about users, applications, and resources on a network. MIIS 2003 is designed for life-cycle management of identity and access to simplify the provisioning of new user accounts, strong credentials, access policies, rights management policies, and so on. MIIS 2003 is available in two versions. First, there's Microsoft Identity Integration Server 2003 SP1, Enterprise Edition, which includes support for identity integration/directory synchronization, account provisioning/deprovisioning, and password synchronization and management. And second, there's Identity Integration Feature Pack 1a for Microsoft Windows Server Active Directory, a free download that provides the same functionality as Microsoft Identity Integration Server 2003 SP1, Enterprise Edition (identity integration/directory synchronization, account provisioning/deprovisioning, and password synchronization) but only between Active Directory directory service, Active Directory Application Mode (ADAM), and Microsoft Exchange Server 2000 and later. Enterprises that need to interface with repositories other than Active Directory, ADAM, or Exchange Server, however, must use MIIS 2003, Enterprise Edition, rather than the free Feature Pack version.

Identity and Access in Windows Server 2003 R2

With the R2 release of Windows Server 2003, Microsoft added two more IDA services to the slate of various services already available on Windows Server 2003 either as in-box services, downloadable add-ons, or separate server products built upon Active Directory directory services. These two new IDA services are Active Directory Application Mode and Active Directory Federation Services.

Active Directory Application Mode (ADAM) is essentially a standalone version of Active Directory directory service that is designed specifically for use with directory-enabled

applications. ADAM does not require or depend upon Active Directory forests or domains, so you can use it in a workgroup scenario on standalone servers if desired—you don't have to install it on a domain controller. In addition, ADAM stores and replicates only application-related information and does not store or replicate information about network resources, such as users, groups, or computers. And because ADAM is not an operating system service, you can even run multiple instances of ADAM on a single computer, with each instance of ADAM supporting a different directory-enabled application and having its own directory store, assigned LDAP and SSL ports, and application event log. ADAM is provided as an optional component of Windows Server 2003 R2, but there's also a downloadable version that can be installed on either Windows Server 2003 or Windows XP.

Active Directory Federation Services (ADFS) is another optional component of Windows Server 2003 R2 that provides Web single sign-on (SSO) functionality to authenticate a user to multiple Web applications over the life of a single online session. ADFS works by securely sharing digital identity and entitlement rights across security and enterprise boundaries, and it supports the WS-Federation Passive Requestor Profile (WS-F PRP) Web Services protocol. ADFS is tightly integrated with Active Directory, and it can work with both Active Directory directory services and ADAM. Using ADFS, an enterprise can extend its existing Active Directory infrastructure to the Internet to provide access to resources that are offered by trusted partners across the Internet. These trusted partners can be either external third parties or additional departments or subsidiaries within the enterprise.

Identity and Access in Windows Server 2008

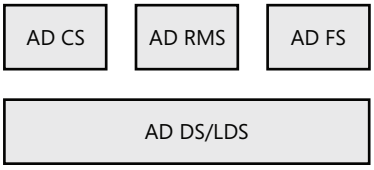
Looking back over this evolution of Active Directory–based IDA services since Windows 2000 Server, we have the following IDA solution for the current platform Windows Server 2003 R2:

- Active Directory directory services and Certificate Services—two core services that can be deployed separately or together.
- Authorization Manager, ADAM, and ADFS—separate optional components that require Active Directory directory services. (Authorization Manager also requires Certificate Services.)
- MIIS 2003, which is available both as a separate product or as a free Feature Pack (depending on whether or not you need to synchronize with non-Microsoft directory services).
- Windows Rights Management Service (RMS), which is available as an optional download from the Microsoft Download Center.

Microsoft's vision with Windows Server 2008 (and beyond) is to consolidate all these various IDA capabilities into a single, integrated IDA solution built upon Active Directory. This consolidation picture as of Beta 3 of Windows Server 2008 is as follows.

As shown in the following diagram, there are four key integrated IDA components present in Windows Server 2008:

- Active Directory Domain Services (AD DS) and Active Directory Lightweight Directory Services (AD LDS), which provide the foundational directory services for domain-based and standalone network environments.
- Active Directory Certificate Services (AD CS), which provides strong credentials using PKI digital certificates.
- Active Directory Rights Management Services (AD RMS), which protects information contained in documents, e-mails, and so on.
- Active Directory Federation Services (AD FS), which eliminates the need for creating and maintaining multiple separate identities.



Note the following rebranding of IDA services in Windows Server 2008:

- Active Directory directory services is now known as Active Directory Domain Services (AD DS).
- Active Directory Application Mode is now called Active Directory Lightweight Directory Services (AD LDS).
- Certificate Services is now called Active Directory Certificate Services (AD CS).
- Windows Rights Management Services is now named Active Directory Rights Management Services (AD RMS).
- Finally, Active Directory Federation Services (ADFS) is still called Active Directory Federation Services (AD FS) but now includes an extra space in the abbreviation.

And for identity life-cycle management, Microsoft also plans on releasing a follow-up to MIIS 2003 called Identity Lifecycle Manager (ILM) 2007 in mid-2007. Initially, ILM 2007 will run on Windows Server 2003, Enterprise Edition. ILM 2007 builds on the metadirectory and user-provisioning capabilities in MIIS 2003 by adding new capabilities for managing strong credentials such as smart cards and by providing an integrated approach that pulls together metadirectory, digital certificate and password management, and user provisioning across Microsoft Windows platforms and other enterprise systems. Microsoft is also working on the next version of ILM, which is codenamed Identity Lifecycle Manager "2." This version is planned for release around the same time as Windows Server 2008, but it will install separately. Before we go any further, let's hear from one of our experts at Microsoft concerning plans for ILM "2" as an identity-management solution for Windows Server 2008:

From the Experts: Identity Lifecycle Manager "2"

Identity Lifecycle Manager "2" is the codename for Microsoft's identity management solution for Windows Server 2008. The principles behind Identity Lifecycle Manager "2" are that identity is everywhere and it can be managed how you want it to be.

Identity Is Everywhere

Identity Lifecycle Manager "2" provides a plethora of ready-to-deploy self-service identity and access solutions. Users can manage their own information and that of their staff, and navigate through the organizational hierarchy. They can reset their own passwords, provision their own smart cards, and retrieve their certificates. They can create security groups and distribution lists, request access to one another's groups, and manage approval.

Best of all, they can do all of this right from within their Office applications and Windows desktops. So, with Identity Lifecycle Manager "2," if you want to request to join a group, you can do that right within Outlook. And when you are asked to approve an action by another user, the Approve and Reject buttons are right there in the approval request mail. And if you forget your password and need to reset it, you can do so right where you are most likely to find that you have forgotten it: at the Windows log-in prompt. All the facilities of Identity Lifecycle Manager "2" are also available from a central portal, hosted within Windows SharePoint Services.

Identity Is Managed How You Want It to Be

Identity Lifecycle Manager "2" lets you manage identity your way by allowing you to accurately model your business processes and attach them to identity and access events. Modeling your unique business procedures around identity and access management processes is meant to be something that each staff member can do for themselves, without having to depend on programmers to do it for them. Thus, Identity Lifecycle Manager "2" provides a simple graphical user interface for modeling your business procedures—the Identity Lifecycle Manager "2" Process Designer. Moreover, you don't have to deploy any special software onto your user's desktops for them to be able to use the Process Designer. The Process Designer is fully incorporated within the Identity Lifecycle Manager "2" portal, which is a Windows SharePoint Services 3 application. So all that users of the Process Designer need to access the designer is their browser.

The three fundamental types of processes that you can model in Microsoft Identity Lifecycle Manager "2" are authentication processes, approval processes, and action processes. Indeed, within Identity Lifecycle Manager "2," processing proceeds by first executing your authentication processes, then your approval processes, and finally your action processes.

Authentication processes are for confirming a user's identity. The steps in an authentication process challenge the user for credentials. This process can also include several steps to define a multifactor authentication process required for more

sensitive operations. Both the built-in authentication activities and your custom ones can leverage the Windows GINA and Windows Vista Credential Provider technologies to challenge users for their credentials at the Windows log-in prompt. This is a desirable option, because then users are challenged to prove their identity precisely where they expect to be challenged.

A second core type of process in the process model of Microsoft Identity Lifecycle Manager "2" is the approval process. Approval processes are for confirming that a user has permission to perform a requested operation. Typically, an approval process involves sending an e-mail message to the owner of a resource asking them to confirm that a user has permission to perform some requested operation on that resource. Identity Lifecycle Manager "2" allows users to respond to those approval requests right from within Outlook, which is precisely where a user would naturally want to be able to do so. Another type of activity in an approval process is one that requires users to submit a business justification for an operation they want to perform. In Identity Lifecycle Manager "2," approval processes can involve any activities that a user might have to complete before being allowed to proceed with an operation. The enabling power of Identity Lifecycle Manager "2" is that it gives you the freedom to determine how you want to gather approvals for users' actions. Then it surfaces the approvals on the end users' desktops, inside an appropriate application context where they would expect to find them—saving the user from having to go elsewhere to manage permissions.

The third and final core type of process in the process model of Microsoft Identity Lifecycle Manager "2" is the action process. Action processes define what happens as a consequence of an operation. A simple example is just having a notification sent to the owner of a resource to inform the owner of a change. A more interesting and, indeed, more common type of activity to perform as a consequence of an identity management operation is an entitlement activity. Thus, you might define a process that, as a consequence of assigning a user to a particular group, allocates a parking permit in the correct lot and issues the appropriate card key for the user's building. The point is that Identity Lifecycle Manager "2" action processes are truly a blank slate. On that blank slate, you get to define how actions on objects within Identity Lifecycle Manager "2" propagate out to the identity stores and resources of your enterprise.

We've said that the principal idea is that you get to define processes that model the identity management procedures of your enterprise and that you get to attach those processes to identity and access events. Up to this point, we have discussed quite a lot about the processes. Now let us turn to the subject of attaching those processes to events.

Events are the triggers that cause Identity Lifecycle Manager "2" processes to be executed. So, in attaching a process to an event, you are defining the circumstances under which the process will be executed. In the nomenclature of Identity Lifecycle Manager "2," we refer to this as mapping a process to an event. We provide a simple user interface for accomplishing it. You identify the process that you have created using the Process Designer, and then you specify the event to which you want to attach the process.

So what is an *event* in Identity Lifecycle Manager "2?" Well, an event is something that happens to a set of one or more objects. For example, you might update the cost center assigned to a particular team of people, or you might update the office telephone number of a single individual. Both constitute examples of events. Another example is the addition of a person to a team—in that case, there is an event for the person being added, as well as an event for the team that the person is joining.

Because an event is something that happens to a set of one or more objects, when you map a process to an event, you must identify the set of objects to which the event is expected to occur. Identity Lifecycle Manager "2" gives you considerable power to identify the sets of objects. You get to define the rules by which objects are included in sets. Those rules can be as rich and complex or as bare and simple as you want them to be. You can define them so as to include any number of objects in a set, and any variety of types of objects as well. Once you have defined rules to identify a set of objects, you can select the events on those objects that you want to serve as triggers for your processes. There are two types of events in Identity Lifecycle Manager "2" that can trigger your processes: request events and transition events.

Request events are events by which the data of an object or set of objects is retrieved or manipulated. So, included in the category of request events are create, read, update, and delete events. Transition events occur when an object moves in or out of a set of objects. So, in the earlier example of a person joining a team, there is a transition for that person in being included in the group and a transition for the group in having that person join.

All in all, the authentication, approval, and action processes that you compose using approval actions, notification actions, and entitlement actions in the Process Designer can be mapped to any request or transition event on any set of objects that you identify via your rules. We believe that this simple model of designing processes and then mapping those processes to events gives you tremendous power to manage the identity life cycle of your organization. Whatever identity-related occurrences that you can imagine happening in your enterprise can be represented as events within Identity Lifecycle Manager "2," and then you can describe processes to handle those events—processes that confirm the identity of the person initiating the event, that confirm the person's permission to initiate the event, or that define the consequences. Crucially, you get to define

those processes as models representing the business policies and procedures that uniquely govern the identity-related assets of your enterprise.

Microsoft Identity Lifecycle Manager "2" is built on the Windows Communication Foundation, Windows Workflow Foundation, and Windows SharePoint Services 3 technologies, and it exposes a thoroughly standards-based API that implements WS-Transfer, WS-ResourceTransfer, WS-Enumeration, and WS-Trust.

–Donovan Follette
Identity and Access Developer Evangelist, Windows Server Evangelism

After reading all this, you hopefully understand now the big picture of what Microsoft's vision is for identity and access, and how Active Directory in Windows Server 2008 fits into this picture. Now it's time to look at each piece of this picture and learn about the new features and enhancements to Active Directory in Windows Server 2008. We'll begin with core improvements to AD DS/LDS.

Active Directory Domain Services

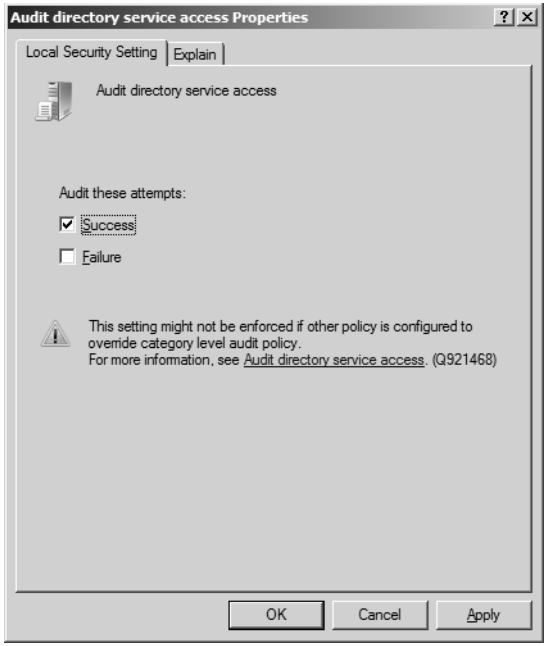
Let's look at four enhancements to Active Directory in Windows Server 2008:

- AD DS auditing enhancements
- Read-only domain controllers
- Restartable AD DS
- Granular password and account lockout policies

There are other improvements as well, including some changes to the user interface for managing Active Directory and also to the Active Directory Installation Wizard. But we'll focus here on the three enhancements just mentioned, as they're big gains for many enterprises.

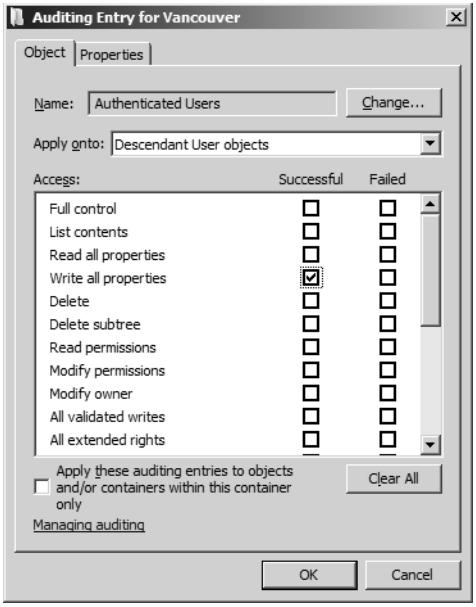
AD DS Auditing Enhancements

The first enhancement we'll look at is AD DS auditing. In the current platform, Windows Server 2003 R2 (and in Windows Server 2008 also), you can enable a global audit policy called Audit Directory Service Access to log events in the Security event log whenever certain operations are performed on objects stored in Active Directory. Enabling logging of objects in Active Directory is a two-step process. First, you open the Default Domain Controller Policy in Group Policy Object Editor and enable the Audit Directory Service Access global audit policy found under Computer Configuration\Windows Settings\Security Settings\Local Policies\Audit Policy.

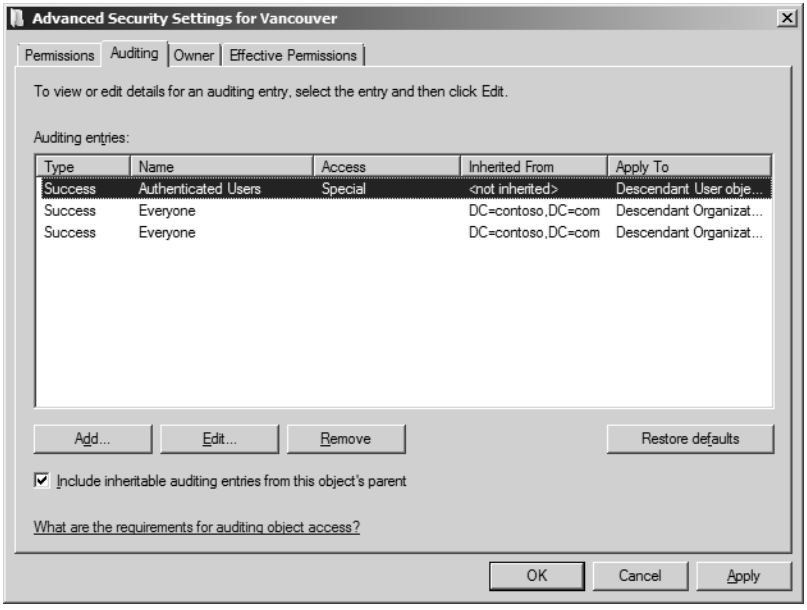


Then you configure the system access control list (SACL) on the object or objects you want to audit. For example, to enable Success auditing for access by Authenticated Users to User objects stored within an organizational unit (OU), you do the following:

1. Open Active Directory Users and Computers, and make sure Advanced Features is selected from the View menu.
2. Right-click on the OU you want to audit, and select Properties.
3. Select the Security tab, and click Advanced to open the Advanced Security Settings for the OU.
4. Select the Audit tab, and click Add to open the Select User, Computer or Group dialog.
5. Type **Authenticated Users**, and click OK. An Auditing Entry dialog opens for the OU.
6. In the Apply Onto list box, select Descendant User Objects.
7. Select the Write All Properties check box in the Select column.

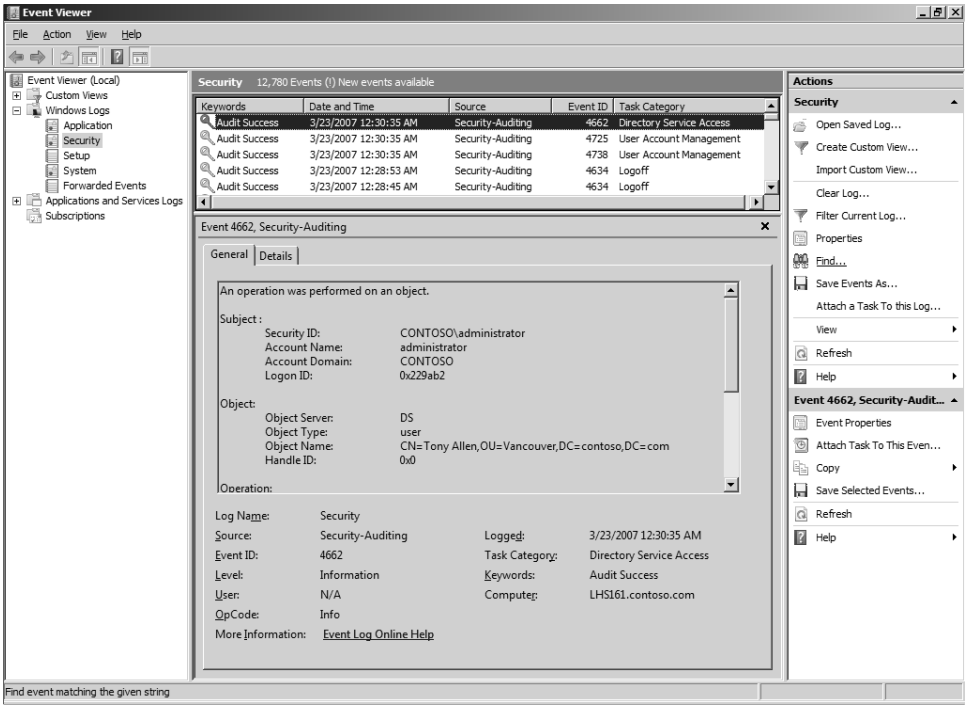


8. Click OK to return to Advanced Security Settings for the OU, which should now show the new SACL you configured.



9. Close all dialog boxes by clicking OK as needed.

Now if you go ahead and change a property of one of the user accounts in your OU—for example, by disabling an account—an event should be logged in the Security log with event ID 4662 and source Directory Service Access to indicate that the object was accessed.



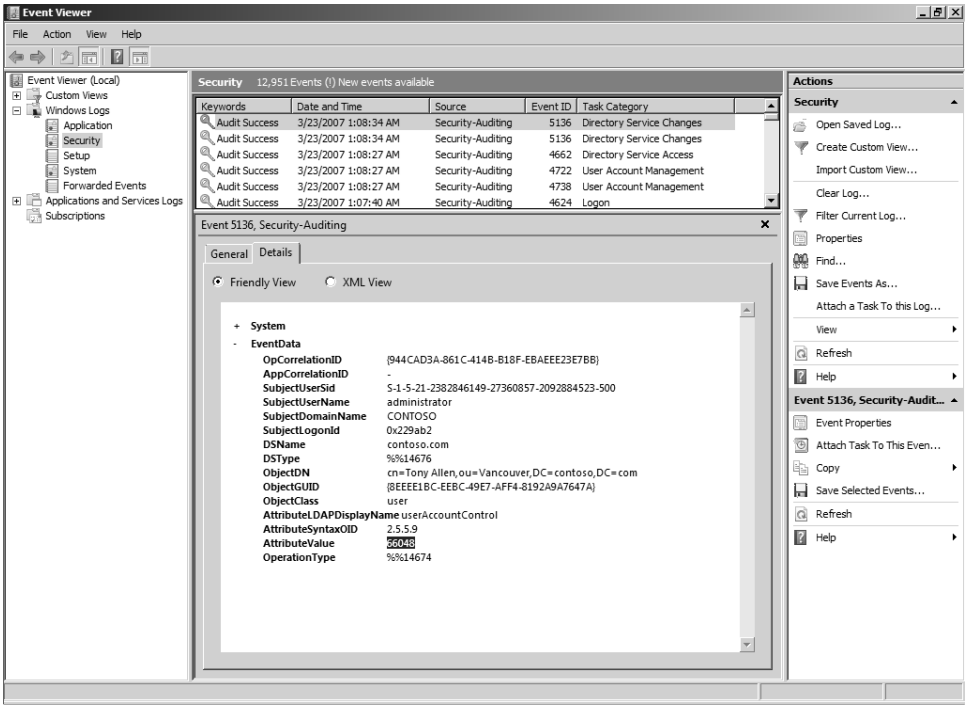
So far, this is the same in Windows Server 2008 as in previous versions of Windows Server. What's new in Windows Server 2008, however, is that while in previous Windows Server platforms there was only one audit policy (Audit Directory Service Access) that controlled whether auditing of directory service events was enabled or disabled, in Windows Server 2008 this policy has been divided into four different subcategories as follows:

- Directory Service Access
- Directory Service Changes
- Directory Service Replication
- Detailed Directory Service Replication

One of these subcategories—Directory Service Changes—has been enhanced to provide the ability to audit the following changes to AD DS objects whose SACLs have been configured to enable the objects to be audited:

- Objects that have had an attribute modified will log the old and new values of this attribute in the Security log.
- Objects that are newly created will have the values of their attributes at the time of creation logged in the Security log.
- Objects that are moved from one container to another within a domain will have their old and new locations logged in the Security log.
- Objects that are undeleted will have the location to which the object has been moved logged in the Security log.

The usefulness of this change should be obvious to administrators concerned about maintaining an audit trail of changes made to Active Directory, and auditing actions like these is an important part of an overall IDA strategy for an organization. For instance, using the Security log and filtering for a particular User object, you can now track in detail all changes to the attributes of that object over the entire lifetime of the object. When you enable Success auditing for the Audit Directory Service Access global audit policy (and this policy has Success auditing enabled for it by default within the Default Domain Controllers Policy), the effect of this is to also enable Success auditing for the first of the four subcategories (Directory Service Access) described earlier, which audits only attempts to access directory objects. If you need to, however, you can selectively enable or disable Success and/or Failure auditing for each of these four auditing subcategories individually by using the Auditpol.exe command-line tool included in Windows Server 2008. For example, if you wanted to enable Success auditing for the second subcategory (Directory Service Changes) so that you can maintain a record of the old and new values of an object's attribute when the value of that attribute is successfully modified, you can do so by typing **auditpol /set /subcategory:"directory service changes" /success:enable** at a command prompt on your domain controller. If we do this in the preceding example and then enable the user account we disabled previously, three new directory service audit events are added to the Security log.



The first (earliest) of these events is 4662, indicating the User object has been accessed, while the second event (5136) records the old value of the attribute modified and the third event (also 5136) records the new value of the attribute. Table 7-1 lists the possible event IDs for Directory Service Changes audit events.

Table 7-1 Event IDs for Directory Service Changes Audit Events

Event ID	Meaning
5136	An attribute of the object has been modified.
5137	The object was created.
5138	The object has been undeleted.
5139	The object has been moved within the domain.

In addition to enabling you to track the history of an object this way, Windows Server 2008 also gives you the option of setting flags in the Active Directory schema to specify which attributes of an object you want to track changes for and which attributes you don't want to track changes for. This can be very useful because tracking changes to objects can lead to a whole lot of audit events and your Security log can fill up awfully fast.

Read-Only Domain Controllers

Another new feature of AD DS in Windows Server 2008 is the Read-Only Domain Controller (RODC), a domain controller that hosts a read-only replica of the AD database. The main rationale for RODCs (apart from nostalgia for the BDCs of good old NT4 days) is to provide a solution for branch offices that have inadequate physical security. For example, a corporate headquarters probably has the resources to adequately protect their domain controllers against theft or other physical dangers—at least, they better have such resources. Small branch offices, however, might not have the facilities, budget, or expertise to ensure a domain controller present there would be physically secure. One solution to this problem might be to not have a domain controller at all at your branch office and just have users there authenticate over a WAN link with a domain controller at headquarters. The problem with this approach is if the WAN link is too slow, unreliable, or saturated with other forms of traffic. The result could be unacceptably slow logons for users or difficulty logging on at all. If your WAN link is unsuitable, the other option is to place a domain controller at your branch office and have users there authenticate locally while the DC itself replicates with DCs at headquarters to ensure its directory database is always up to date. The problem with *this* approach, however, is that domain controllers are the heart and soul of your Windows-based network because they contain all the accounts for all the users and computers on your network. So if the domain controller at your branch office somehow got stolen (perhaps by some clever social engineering like, "Hi, I've come to clean your domain controller, can you show me where it is?"), your whole network should be considered compromised and your only viable solution is to flatten everything and rebuild it all from scratch.

And those are the only two solutions today for branch offices using domain controllers running Windows Server 2003—authenticate over the WAN or risk placing a domain controller at your branch office. RDOC, however, solves this dilemma by providing a *secure* way to have a domain controller at your branch office. The only requirement for using RDOC is that the domain controller that holds the PDC Emulator FSMO role on your network has to be running Windows Server 2008. Once this is the case and you've deployed an RDOC at your branch office, changes made to the directory on your normal (writable) domain controllers replicate to the RDOC, but nothing replicates in the opposite direction. That's because the directory database of a RDOC is read-only, so you can't write anything to it locally—it has to receive all changes to its database via replication from another (writable) domain controller. (RDOCs can't replicate with each other either, so there's no point having more than one RDOC at a given site—plus it could cause inconsistent logon experiences for users if you did do this.) So RDOC replication is completely unidirectional—and this applies to DFS replication traffic as well.

RDOCs also advertise themselves as the Key Distribution Center (KDC) for the branch office where they reside, so they handle all requests for Kerberos tickets from user and computer accounts at the remote site. RDOCs don't store user or computer credentials in their directory database, however; so when a user at the branch office tries to log on, the RDOC contacts a

writable DC at the hub site to request a copy of the user's credentials. How the hub DC responds to the RDOC's request depends on how the Password Replication Policy is configured for that RDOC. If the policy says that the user's credentials can be replicated to the RDOC, the writable DC does this, and the RDOC caches the credentials for future use (until the user's credentials change). The result of all this is that RDOCs generally have few credentials stored on them. So if an RDOC somehow gets stolen (remember the DC cleaning guy), only those credentials are compromised and replacing them is much less work than rebuilding your entire directory from scratch.

Another feature of RDOCs is that a domain administrator can delegate the local administrator role for an RDOC to an ordinary domain user. This can be very useful for smaller branch offices that have no full-time expert IT person on site. So if you need to load a new driver into your DC at a remote site, you can just give instructions to your "admin" by phone on how to do this. The admin is simply an ordinary user who can follow instructions, and delegating RDOC admin rights to him doesn't enable him to perform any domain-wide administrative tasks or log on to a writable DC at headquarters—the damage he can do is limited to wrecking only the RDOC.

Let's hear now from a Microsoft MVP and directory services expert concerning some enhancements that have been made to dcpromo.exe in Windows Server 2008 and how these enhancements relate to deploying RODCs:

From the Experts: New Active Directory Setup Wizard (dcpromo.exe)

When you want to install Active Directory, you have to use the Active Directory Setup Wizard (dcpromo.exe). It provides you with some possibilities and assumes that you have a proper design written down and you know what you want to accomplish. However, we have received many support calls and questions on the Internet because Active Directory and DNS were not set up in a way that reflects best practices. Considering the vast amount of installations of Active Directory, it's very clear that it's far easier to find the Active Directory Installation Wizard on the server operating system than it is to find best practices or good consultancy. Common support issues included having the wrong FSMO-Roles together on the same system, not enough Global Catalog servers, or issues in the DNS-Design that were leading to logons over the WAN lines.

In Windows Server 2008, Microsoft has put a huge effort into changing dcpromo.exe. Now it is reflecting best practices. You get a normal mode if you just want to quickly install Active Directory, and you get an advanced mode if you want to do any special configurations. Dcpromo is leveraging best practices, and it provides a lot of additional tasks. It's checking the FSMO roles for you, and it recommends whether to automatically move the Infrastructure Master if necessary. It allows you to enable the Global Catalog on a new domain controller. It is checking the DNS infrastructure, and it allows you to automatically create forwarders and delegations. Also, dcpromo enables you to choose

your replication partner for the initial replication so that you can make sure to target a specific DC.

In addition, dcpromo supports the new Read Only Domain Controller (RODC) in multiple ways. You are either able to precreate a RODC-Account in your domain and delegate a site admin to join the RODC to the domain, or you are able to fully install the RODC while selecting whether it should also be a Global Catalog server a DNS-server, or both.

Last but not least, dcpromo finally supports unattended installations from the command line without an answer script. Simply run **dcpromo /?:unattend** to figure out what parameters you have to script the installation of your Windows Server 2008 Active Directory Domain Controller.

–Ulf B. Simon-Weidner
MVP for Windows Server–Directory Services author, consultant, speaker, and trainer

Finally, because domain controllers often host the DNS Server role as well (because DNS is the naming system used by AD), the RDOCs need a special read-only form of DNS Server running on them also. To learn more about this feature, however, let's listen to another one of our experts at Microsoft:

From the Experts: Advanced Considerations for DNS on RODCs in Branch Office Sites

When installing a Windows Server 2008 Read Only Domain Controller (RODC) at a branch office site, using the Active Directory Installation Wizard or the DCPromo command-line tool, you are prompted to specify a DNS domain for the Active Directory domain that you are joining the RODC to during promotion. During this process, you are prompted with DNS Server installation options. A DNS Server is required to locate domain controllers and member computers in an Active Directory domain, at both the hub site and the local branch office site. The default option is to install a DNS Server locally on the RODC, which replicates the existing AD-integrated zone for the domain specified and adds the local IP address in the DNS Server list of the domain controller local DNS Client setting.

As a best practice, Microsoft recommends that client computers have Dynamic DNS updates turned on by default and that DHCP Servers be used to configure the DNS Server list. Similarly for branch office sites, clients should be configured to use Dynamic DNS updates, and you should set the Primary DNS Server or use DHCP to set the DNS Server list to direct clients to the DNS Server running on the RODC.

If there is only one DNS Server and RODC running at the branch office site, Microsoft recommends that client computers also point to a DNS Server running on a domain controller at the hub site. This can be done either by configuring clients with an Alternate DNS Server for the hub-site DNS Server or by configuring DHCP Servers to set the DNS Server list to first the local DNS Server and then the remote DNS Server at the hub site. The DNS Server on the RODC should be the first DNS Server in the list to optimize resolution performance for branch office clients.

In larger branch office scenarios, if setting up two or more RODCs at a site, you are provided the default option to install DNS Server locally on all the RODCs. Within the same site, the RODCs do not replicate directly with each other. The RODCs rely mainly on replication with domain controllers at the hub site during scheduled intervals to refresh local data in the directory. Hence, a branch office DNS Server on an RODC receives updated DNS zone data during the normal replication cycle from a hub-site domain controller connected to the local RODC.

In addition to replication from the hub site, DNS Servers on RODCs also attempt to replicate local data after receiving a client update request. The branch office DNS Server redirects the client to a hub-site DNS Server on a domain controller that is writable and can process the update. Shortly thereafter, it attempts to contact a hub-site domain controller to update its local copy of the data with the changed record. Any other branch office DNS Server on RODCs at the site do not attempt to obtain a local copy of the single record update because they did not receive the original client update request. This mechanism has the advantage of allowing an updated client record to be resolved quickly within the branch office, without necessitating frequent and large replication requests for all domain data from the hub site. If network connectivity is lost, or no domain controller at the hub site is able to provide the updated record data to the DNS Server in the branch office, the record will be available locally only after the next scheduled replication from the hub-site domain controllers, and it will be available to all RODCs at the branch office site.

As a consequence of a DNS Server's attempt to replicate individual records between replication cycles, if DNS zone data is stored across multiple RODCs, the local branch office records might accumulate some incongruities. To ensure a high level of consistency for DNS data, the recommendation is to configure all client computers at the branch office site with the same DNS Server list—for example, by using DHCP.

If, however, in the more rare case that timely resolution of local branch office client records is absolutely critical, to avoid any inconsistencies for resolution, you can install DNS Servers on all RODCs at the site, but point clients only to a single DNS Server.

–Moon Majumdar
Program Manager, DNS (Server and Client) and DC Locator, Directory and Service Team

Restartable AD DS

Another new feature of AD DS in Windows Server 2008 is the ability to restart the Active Directory directory services without having to restart your domain controller in Directory Services Restore Mode. In previous versions of Windows Server, when you wanted to do some maintenance task on a domain controller—such as performing offline defragmentation of the directory database or performing an authoritative restore of the Active Directory directory service database—you had to restart your domain controller in Directory Services Restore Mode by pressing F8 during startup and selecting this from the list of startup options. You then logged on to your domain controller by using the local Administrator account specified previously when you ran the Active Directory Installation Wizard (dcpromo.exe) on your machine to promote it from a member server to a domain controller. Once logged on in Directory Services Restore Mode, you could perform maintenance on your domain controller and clients couldn't authenticate with it during your maintenance window.

Having to reboot a domain controller like this to perform maintenance operations resulted in longer downtime for clients who needed to be authenticated by your domain controller. To reduce this downtime window, AD DS has been re-architected in Windows Server 2008. Instead of rebooting your machine and logging on in Directory Services Restore Mode, you simply stop the Domain Controller service by using the Services snap-in (shown in Figure 7-1) or typing **net stop ntds** at a command line, perform your maintenance tasks while still logged on as a domain admin, and when you're finished start this service again using the snap-in or the **net start ntds** command. Stopping and starting the Domain Controller service like this also has no effect on other services such as the DHCP Server service that might be running on your domain controller.

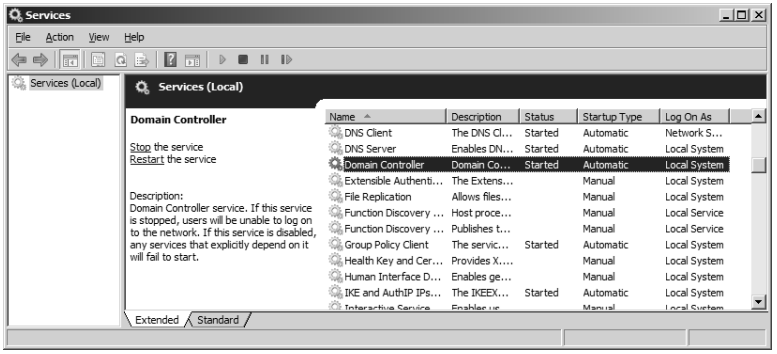


Figure 7-1 You can now stop and start the Domain Controller (NTDS) service without rebooting your domain controller and logging on in Directory Services Restore Mode

While domain controllers running previous versions of Windows Server had two Active Directory directory service modes (normal mode and Directory Services Restore Mode), domain controllers running Windows Server 2008 now have three possible modes or *states* they can be running in:

- **AD DS Started** This is the normal state when the NTDS service is running and clients can be authenticated by the domain controller. This state is similar to how AD directory services worked in Windows 2000 Server and Windows Server 2003.
- **Directory Services Restore Mode** This state is still available on domain controllers running Windows Server 2008 through the F8 startup options, and it's unchanged from how it worked in Windows 2000 Server and Windows Server 2003.
- **AD DS Stopped** This is the new state for domain controllers running Windows Server 2008. A domain controller running in this state shares characteristics of both a domain controller running in Directory Services Restore Mode and a member server that is joined to a domain. For example, as in Directory Services Restore Mode, a domain controller running in the AD DS Stopped state has its directory database (Ntds.dit) offline. And similar to a domain-joined member server, a domain controller running in this state is still domain-joined, and users can log on interactively or over the network by using another domain controller. But it's a good idea not to let your domain controller remain in the AD DS Stopped state for an extended period of time because not only will it be unable to service user logon requests, it also will be unable to replicate with other domain controllers on the network.

Granular Password and Account Lockout Policies

New in Beta 3 of Windows Server 2008 is the ability to have multiple password policies and account lockout policies in a domain. To learn about this particular feature, let's hear from a Microsoft MVP and directory services expert:

From the Experts: Granular Password Policies in Windows Server 2008

If you want to deploy multiple password policies in your forest, the domain has always been the boundary for this. This was confusing for many customers because you are able to change passwords in every Group Policy Object (GPO). However, remember that password settings (and account lockout settings) are configured in the Computer Settings part of the GPO. They apply only to computer objects, and therefore, to local accounts on those computer objects. An exception to this rule is policies that are linked to the domain head (the top node of the domain). GPOs linked here that hold password

settings are the administrative interface for the password and account lockout settings for domain objects. Actually, they are written back to attributes on the domain head object and take effect from there. Domain controllers that receive a password change request compare the settings on the domain head with the password, and they either allow the password change or deny it. So it's important to understand that password and account lockout settings are maintained on the domain head in Active Directory. You also need to keep in mind that Group Policies are only the administrative interface and that password settings configured in any GPO linked to any other OU or site are applied only to the local user accounts of the computer object to which the policy applies.

So, in the past, password and account lockout settings were limited to the domain and we were able to apply only one setting per domain. If we wanted to have different password policies, we were required to deploy multiple domains.

This has been changed in Windows Server 2008. Active Directory is extended, and the password settings validation on the domain controllers have been extended so that we are able to configure multiple password and account lockout settings for each domain now. How are they administered? Not via GPO—as mentioned before, GPO has been only an administrative interface. So the new fine-grained password policies are configured as new objects in the domain and are linked to either groups or users in the domain.

If you want to experiment with this, simply use ADSIEdit.msc. Expand the Password Settings Container underneath the System Container in the domain, right-click, and select New. You are prompted to fill in the following mandatory attributes, which define password and account lockout policies:

- *msDS-PasswordSettingsPrecendence* This attribute is just a virtual number you can make up. (Be sure you leave some space in the numbering for future use.) It defines which password settings take effect if multiple settings apply to the same object (user or group, but settings on the user always take precedence over settings on the group).

  This will usually reflect on the "level" of the settings object. For example, if you have stronger settings, they have a lower value, and if you have higher settings, you're probably assigning a higher precedence to them.

- *msDS-PasswordReversibleEncryptionEnabled* This attribute is Boolean and defines whether you want to store the passwords of the accounts (that is, specify to whom the password settings object applies) in reversible encryption or not. The default and best practice is to set this value to FALSE.

- *msDS-PasswordHistoryLength* This setting defines how many old passwords the user cannot reuse again (to prevent the user from changing the password back and forward to the same one or changing it multiple times until he's able to reuse his old password).

  The domain default is to not allow the last 24 passwords of that user.

- *msDS-PasswordComplexityEnabled* This attribute is also a Boolean and defines whether the password needs to be complex (that is, it has at least three of the following character sets applied: lower letters, capital letters, numbers, symbols, or unicode characters).

  The domain default and best practice is to turn it on (TRUE).

- *msDS-MinimumPasswordLength* This attribute defines the minimum length of a password in characters. The domain default is seven characters long.

- *msDS-MinimumPasswordAge* The msDS-MinimumPasswordAge attribute does just what its name suggests—it defines the minimum age for passwords. The minimum age is necessary to prevent a user from changing her password *x* amount of times on the same day until she exceeds the Password History limit and can change the password back to the same value as before.

  This is a negative number that you can compile or decompile, using the scripts at *http://msdn2.microsoft.com/en-us/library/ms974598.aspx* as a guideline. (The domain default is 1 day, which equals -864000000000.)

- *msDS-MaximumPasswordAge* This attribute is just the opposite of the previous one. It defines when you have to change your password. It is also a negative number just like the previous one. (The domain default is 42 days, which equals -36288000000000.)

- *msDS-LockoutThreshold* Defines how many failed attempts at entering a password a user can have before the user object will be locked. (The domain default is 0, which equals "Don't lock out accounts after invalid passwords.")

- *msDS-LockoutObservationWindow* This attribute determines at which time the bad password counter should be reset. (The domain default is 6 minutes, which equals -18000000000.)

- *msDS-LockoutDuration* This attribute determines how long a password should be locked. (The domain default is 6 minutes, which equals -18000000000.)

After you create your own password settings object (PSO), you have to link it to a user or group. I recommend, for administrative purposes, always linking it to groups instead of to users. (Otherwise, it will get messy and hard to administer.) To link the PSO to a group or user, you simply change its *msDS-PSOAppliesTo* attribute to the distinguished name of the group or user (for example, *cn=Administrators,cn=Users,dc=example,dc=com*). This is a multivalued attribute, so you are able to link the same PSO to multiple groups or users.

For administrative purposes, there are also two attributes that help you determine which password policies are applied to which users or groups. On the group or user, you will find the *msDS-PSOApplied* attribute, which is actually the back link of the *msDS-PSOAppliesTo* attribute and lists all PSOs that are directly linked to this object.

To help you figure out which PSO is the effective one, there's the constructed attribute *msDS-ResultantPSO*, which shows you which PSO is effective for the object in question.

At the beta stage that is current at the writing of this book, this is a new feature that lacks adequate administrative support in the graphical user interface. However, you are able to administer it easily using ADSIEdit.msc. And Joe Richards, a Directory Services MVP who wrote Active Directory command line tools such as ADFind and ADMod, has created a new command-line utility named PSOMgr.exe, which helps you create and link PSOs. You'll find it at *www.joeware.net*.

–Ulf B. Simon-Weidner
MVP for Windows Server–Directory Services author, consultant, speaker, and trainer

Active Directory Lightweight Directory Services

Another feature of Active Directory in Windows Server 2008 is the new built-in Active Directory Lightweight Directory Services (AD LDS) server role. Well, actually it's not new because this is essentially the same Active Directory Application Mode (ADAM) feature that was available as an out-of-band download for Windows Server 2003 and Windows XP. What's new is mainly that this directory service is now available as an in-box role that can be added to your Windows Server 2008 server using the Role Manager tool described in Chapter 4, "Managing Windows Server 2008," instead of it needing to be downloaded from the Microsoft Download Center as in previous versions of Windows.

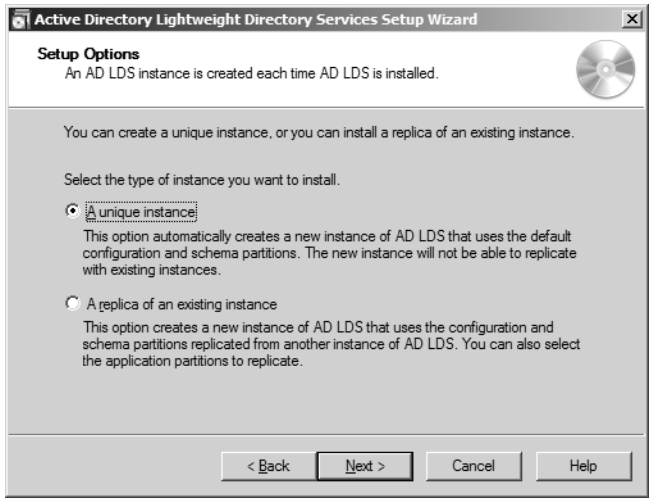
So AD LDS is basically just ADAM, but what's ADAM? ADAM (we'll call it by its new name now, AD LDS) is basically a stripped-down version of AD DS that supports a lot of the features of AD DS (multimaster replication, application directory partitions, LDAP over SSL access, the ADSI API) but doesn't store Windows security principals (such as domain user and computer accounts), domains, global catalogs, or Group Policy. In other words, AD LDS gives you all the benefits of having a directory but none of the features for managing resources on a network. Instead, AD LDS is designed to support applications that need a directory for storing their configuration and data instead of storing these in a database, flat file, or other form of repository. Examples of directory-enabled LOB apps that could use AD LDS include CRM and HR applications or global address book apps. Because such apps often require schema changes in order to work with AD DS, a big advantage of AD LDS is that you can avoid having to make such changes to your AD DS schema, as making mistakes when you modify your AD DS schema can be costly—think flatten and rebuild everything from scratch! And it's particularly useful also if your directory-enabled LOB apps will be made available to customers or partners over an extranet or VPN connection because using AD LDS instead of AD DS in this scenario means you don't have to risk exposing your domain directory to nondomain users and computers.

Once you've added the AD LDS role in Server Manager, to use this feature you create an AD LDS instance. An AD LDS instance is an application directory that is independent of your

domain-based AD DS and can run on either a member server or a domain controller if desired. (There's no conflict when running AD DS and AD LDS on the same machine as long as the two directories use a different LDAP path and different LDAP/SSL ports for accessing them. And you can even run multiple AD LDS instances on a single machine—for example, one instance for each LOB app on the machine—without conflict as long as their paths and ports are unique.)

Let's quickly walk through creating a new AD LDS instance and show how you can manage it:

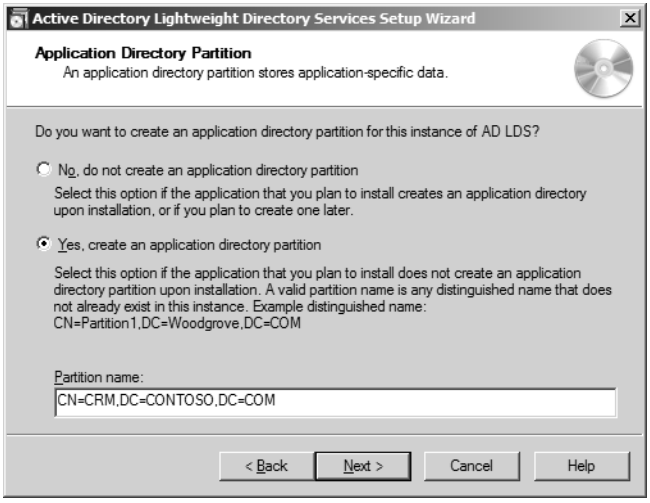
1. After installing the AD LDS role on your server, select the Active Directory Lightweight Directory Services Setup Wizard from Administrative Tools on your Start menu. This launches a wizard for creating a new instance of AD LDS on the machine:



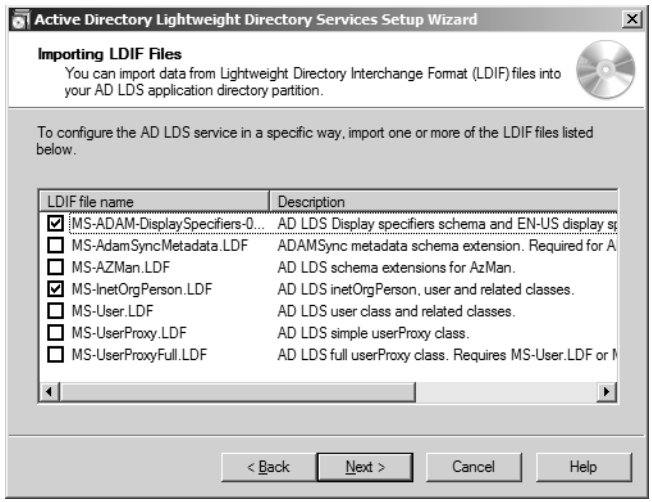
2. Select the A Unique Instance option, and click Next. Then specify a name for the new instance (using only alphanumeric characters and the dash in your name):

3. Click Next, and specify LDAP and SSL ports for accessing your instance:

4. Click Next, and either allow the application to create its own directory partition when you install the application or type a unique distinguished name (DN) for the new application partition you are going to create:



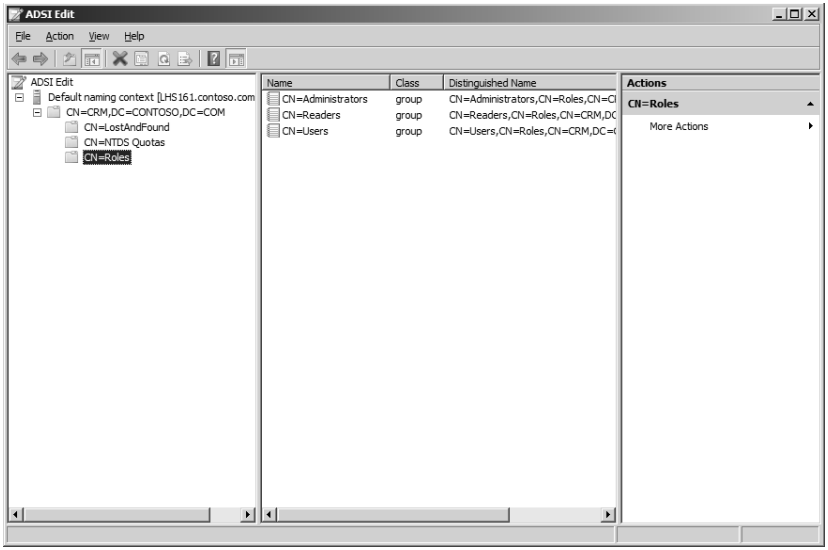
5. Click Next, and in the following wizard pages specify the location where data and recovery files for the partition will be stored, the service account under whose context the AD LDS instance will be running, and the user or group who will have administrative privileges for managing your instance. After completing these steps, you'll be asked to select from a list of default LDIF files you can import to add specific functionality to your instance:



6. Click Next to confirm your selections, and then click Finish to run the wizard and create the instance.

Once you've created your new AD LDS instance, you can manage it using ADSI Edit, an MMC snap-in available from Active Directory Lightweight Directory Services under Administrative Tools. To do this, open ADSI Edit, right-click on the root node, and select Connect To. When the Connection Settings dialog opens, specify the DN for the connection point to your instance (which was CN=CRM,DC=CONTOSO,DC=COM in our example) and click the Advanced button to specify the LDAP port (50000 in our example) for connecting to the instance:

Clicking OK then opens your AD LDS instance in ADSI EDIT. Then you can navigate the directory tree and view and create or modify objects and their attributes in your application directory partition as needed to support the functionality of your directory-enabled LOB app.



Active Directory Certificate Services

Let's move on and briefly describe improvements to Active Directory Certificate Services (AD CS) in Windows Server 2008. We'll focus on the following key improvements:

- Improvements to certificate Web enrollment support
- Support for Network Device Enrollment Service to allow network devices such as routers to enroll for X.509 certificates
- Support for the Online Certificate Status Protocol to easily manage and distribute certificate revocation status info
- The inclusion of PKIView for monitoring the health of Certification Authorities (CAs)

There are other improvements as well for AD CS—such as new Group Policy settings—but we'll pass over these for now because they'll be well documented once Windows Server 2008 RTMs. But we will also hear from the AD CS product group concerning some other enhancements to AC CS in Windows Server 2008.

Certificate Web Enrollment Improvements

Enrollment is the process of issuing and renewing X.509 certificates to users and computers when a PKI has been deployed in your enterprise. Users and computers belonging to an Active Directory domain can take advantage of a mechanism called *autoenrollment*, which

allows them to automatically enroll domain-joined computers when they boot and domain users when they log on. Windows Server 2003 also includes a Certificate Request Wizard to enable domain users to request a new certificate manually when they need to.

Users and computers that are not domain joined or that run a non-Microsoft operating system can use Web enrollment instead. Web enrollment is built on top of Internet Information Services and allows a user to use a Web page to request a new certificate or renew an existing one over an Internet or extranet connection.

What's changed with this feature in Windows Server 2008 is that the old XEnroll.dll ActiveX control for the Web enrollment Web application has now been retired for both security and manageability reasons. In its place, a new COM control named CertEnroll.dll is now used, which is more secure than the old control but whose use can pose some compatibility issues in a mixed environment. For reasons of time, we can't get into these compatibility issues here, but see the "Additional Resources" section at the end of this chapter for more information on this topic.

Network Device Enrollment Service Support

Another enhancement in AD CS in Windows Server 2008 is the inclusion of built-in support for the Network Device Enrollment Service (NDES). Let's listen to one of our experts at Microsoft briefly describe this new feature (and see the "Additional Resources" section at the end of the chapter for links to more information on the subject):

From the Experts: Network Device Enrollment Service

Network Device Enrollment Service is one of the optional components of the Active Directory Certificate Services (AD CS) role. This service implements the Simple Certificate Enrollment Protocol (SCEP). SCEP defines the communication between network devices and a Registration Authority (RA) for certificate enrollment.

SCEP enables network devices that cannot authenticate to enroll for x.509 certificates from a Certification Authority (CA). At the end of the transactions defined in this protocol, the network device will have a private key and associated certificate that are issued by a CA. Applications on the device can use the key and its associated certificate to interact with other entities on the network. The most common usage of this certificate on a network device is to authenticate the device in an IPSec session.

–Oded Shekel
Program Manager, Windows Security

Online Certificate Status Protocol Support

Another new feature of AD CS in Windows Server 2008 is support for the Online Certificate Status Protocol (OCSP). In a traditional PKI, such as one implemented using Certificate

Services in Windows Server 2003, certificate revocation is handled by using certificate revocation lists (CRLs). There has to be a way of revoking certificates that expire or are compromised; otherwise, a PKI system won't be secure. CRLs provide a way of doing this by enabling clients to download a list of revoked certificates from a CA to ensure the certificate they're trying to verify (for example, a certificate belonging to a server the client is trying to connect to) is valid. Unfortunately, once a lot of certificates have been revoked in an enterprise, the CRL can become quite large and have an impact on performance when authenticating over slow WAN links or during peak traffic times, like the beginning of the workday when everyone is trying to log on to the network at the same time.

To improve performance in checking for revoked certificates and increase the scalability of a PKI system, Windows Server 2008 includes an optional Online Certificate Status Protocol role service you can install on a server by adding the Active Directory Certificate Services role using Server Manager. OCSP provides an Online Responder that can receive a request to check for revocation of a certificate without the client having to download the entire CRL. This speeds up certificate revocation checking and reduces the network bandwidth used for this process, which can be especially helpful when such checking is done over slow WAN links. AD CS in Windows Server 2008 even supports *Responder* arrays, in which multiple OCSP Online Responders are linked together to provide fault tolerance, increased scalability, or functionality needed for geographically dispersed PKI deployments.

OCSP support is described in more detail in one of the links in the "Additional Resources" section at the end of this chapter. Meanwhile, let's hear from one of our experts at Microsoft concerning this new feature:

From the Experts: Online Responder

The Online Responder server rule implements the server component of the Online Certificate Status Protocol (OCSP).

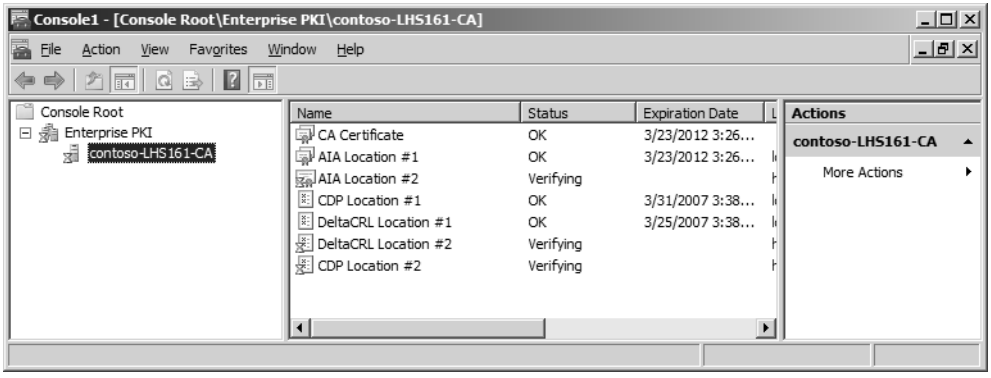
OCSP uses Hypertext Transfer Protocol (HTTP) and allows a relying party to submit a certificate status request to an OCSP responder. This returns a definitive, digitally signed response indicating the certificate status. The Microsoft Online Responder was built with scalability, performance, security, and manageability in mind. It includes the following two components:

- **Online Responder Web Proxy Cache** First and foremost, this component is the service interface for the Online Responder. It is implemented as an Internet Server API (ISAPI) Extension hosted by Microsoft Windows Internet Information Services (IIS).
- **Online Responder Service** This component is a Microsoft Windows NT service (ocspsvc.exe) that is running with NETWORK SERVICE privileges.

–Oded Shekel
Program Manager, Windows Security

Enterprise PKI and CAPI2 Diagnostics

Monitoring the health of CAs in an enterprise PKI deployment is important to prevent problems from arising and to troubleshoot issues when they arise. The Windows Server 2003 Resource Kit included a tool called PKI Health that could be used to display the status of each CA in a chain of CAs; in Windows Server 2008, this tool has been renamed Enterprise PKI (PKIView) and has been re-implemented as an MMC snap-in. Using PKIView, enterprise PKI admins can check the validity or accessibility status of authority information access (AIA) locations and certificate revocation list (CRL) distribution points (CDPs) for multiple CAs within an enterprise that has a Windows Server–based PKI deployed:



PKIView isn't the only way of troubleshooting problems with a Windows Server 2008–based PKI, however. Another useful tool is CAPI2 Diagnostics, which is described in the next sidebar contributed by one of our experts:

From the Experts: Troubleshooting PKI Problems on Windows Vista and Windows Server 2008

Microsoft Windows Vista and Microsoft Windows Server 2008 have a new feature—CAPI2 Diagnostics—that can help you with PKI troubleshooting. This feature enables administrators to troubleshoot PKI problems by collecting detailed information about certificate chain validation, certificate store operations, and signature verification. In case of errors in PKI-enabled applications, detailed information—such as the low-level API results and errors, objects retrieved, and status flags raised at different steps—is available in the logs. This functionality can help reduce the time required to diagnose problems. For troubleshooting purposes, enable CAPI2 logging, reproduce the problem, and use the data in the logs to identify the root cause. To enable logging, follow these steps:

1. Open the Event Viewer, and go to Application And Services Logs\Microsoft\Windows\CAPI2 to get the CAPI2 channel.

2. Right-click Operational, and select Enable Log to enable CAPI2 Diagnostics logging.
3. To save the log to a file, right-click Operational and select the Save Events As option. You can save the log file in the .evtx format (which can be opened through the Event Viewer) or in XML format.
4. If there is data present in the logs before you reproduce the problem, it is recommended that you clear the logs before the repro. This allows only the data relevant to the problem to be collected from the saved logs. To clear the logs, right-click Operational and select the Clear Log option.
5. The default size for the event log is 1 MB. For CAPI2 Diagnostics, the logs tend to grow in size quickly, and it is recommended that you increase the log size to at least 4 MB to capture relevant events. To increase the log size, right-click Operational and select the Properties option. In the log properties, increase the maximum log size.

To learn more about CAPI2 Diagnostics, check out the whitepaper titled "Troubleshooting PKI Problems on Windows Vista" at *http://www.microsoft.com/downloads/details.aspx?FamilyID=FE8EB7EA-68DA-4331-9D38-BDBF9FA2C266&displaylang=en*.

–Yogesh Mehta
Program Manager, Windows Security

Other AD CS Enhancements

Finally, let's briefly hear from one of our experts on the product team at Microsoft concerning two more enhancements to AD CS in Windows Server 2008. Our first sidebar outlines some important changes to V3 certificate templates and the cryptographic algorithms they support in Windows Server 2008 (and in Windows Vista):

From the Experts: V3 Certificate Templates

One important change in Windows Server 2008 and Windows Vista is the support for CNG (Suite-B). With Suite-B algorithms, it is possible to use alternate and customized cryptographic algorithms for encryption and signing certificates.

To support these algorithms, a new certificate template version was added—V3. A V3 certificate template is enhanced in the following ways:

- Support for asymmetric algorithms implemented by a Key Service Provider (KSP) for CNG. By default, Windows implements the following algorithms: DSA, ECDH_P256, ECDH_P384, ECDH_P521, ECDSA_P256, ECDSA_P384, ECDSA_P521, and RSA.

- Support for hash algorithms implemented by a KSP. By default, Windows implements the following algorithms: MD2, MD4, MD5, SHA1, SHA256, SHA384, and SHA512.
- A discrete signature (PKCS#1 V2.1) can be required for certificate requests. Activating this option forces a client that uses the certificate autoenrollment functionality or enrolls a certificate through the Certificates MMC snap-in to generate a certificate request that carries a discrete signature. Selecting this option does not mean that a certificate that is issued from this template also carries a discrete signature. The setting applies to the certificate request only. Also, the setting is not relevant for certificate requests that are created with the certreq.exe command-line tool.
- The Advanced Encryption Standard (AES) algorithm can be specified to encrypt private keys while they are transferred to the CA.
- For machine templates, read permissions on the private key can be added to the Network Service so that services such as IIS have permission to use certificates and keys that are available in the computer's certificate store. In previous versions of Windows, manually adjusting permissions on the computer's certificate store is required.
- The list of asymmetric algorithms is filtered based on the template purpose in the Request Handling tab.

–Oded Shekel
Program Manager, Windows Security

And our second sidebar describes the new restricted enrollment agent functionality in Windows Server 2008's implementation of Enterprise CA:

From the Experts: Restricted Enrollment Agent

Enrollment agents are one or more authorized individuals within an organization. The enrollment agent needs to be issued an Enrollment Agent certificate, which enables the agent to enroll for certificates on behalf of users. Enrollment agents are typically members of the corporate security, IT security, or help desk teams because these individuals have already been trusted with safeguarding valuable resources. In some organizations, such as banks that have many branches, help desk and security workers might not be conveniently located to perform this task. In this case, designating a branch manager or other trusted employee to act as an enrollment agent is required.

The Windows Server 2003 Enterprise CA does not provide any configurable means to control enrollment agents except from enrollment agents' certificates enforcement. The enrollment agent certificate is a certificate containing the Certificate Request Agent application policy extension (OID=1.3.6.1.4.1.311.20.2.1).

The restricted enrollment agent is a new functionality that allows limiting the permissions that enrollment agents have for enrolling on behalf of other users. On a Windows Server 2008 Enterprise CA, an enrollment agent can be permitted for one or many certificate templates. For each certificate template, you can configure which users or security groups the enrollment agent can enroll on behalf of. You cannot constrain an enrollment agent based on a certain Active Directory organizational unit (OU) or container. As mentioned previously, you must use security groups. Note that the restricted Enterprise enrollment agent is not available on a Standard CA.

–Oded Shekel
Program Manager, Windows Security

Active Directory Federation Services

Active Directory Federation Services (AD FS) is another important part of the overall IDA solution provided by Windows Server 2008. AD FS is designed to address a situation that is common in business nowadays—a partner or client that resides on a different network has to access a Web application exposed by your own organization's extranet. In a typical scenario, the client has to enter secondary credentials to this when she tries to access a Web page on your extranet. That's because the client's credentials on her own network might not be compatible or might not even be known by the directory service running on your own network.

AD FS is designed to eliminate the need for entering such secondary credentials by providing a mechanism for supporting single sign-on (SSO) between different directories running on different networks. AD FS does this by providing the ability to create trust relationships between the two directories that can be used to project a client's identity and access rights from her own network to networks belonging to trusted business partners. By deploying one or more federation servers in multiple organizations, federated business-to-business (B2B) partnerships can also be established to facilitate B2B transactions between trusted partners.

To deploy AD FS, at least one of the networks involved must be running either AD DS or AD LDS. AD FS has been around since Windows Server 2003 R2, but it has been enhanced in several ways in Windows Server 2008. For example, AD FS is now easier to install and configure in Windows Server 2008 because it can be added as a server role using Server Manager. AD FS is also easier to administer in Windows Server 2008, and the process of setting up a federated trust between two organizations by exporting and importing policy files is now simpler and more robust. Finally, AD FS now includes improved application support and is more tightly integrated with Microsoft Office SharePoint Services 2007 and also the Active Directory Rights Management Services (AD RMS) component of Windows Server 2008.

Let's learn some more about the improved import/export functionality in AD FS in Windows Server 2008 from some of our product group experts:

From the Experts: Using Import/Export Functionality to More Efficiently Create Federation Trusts

There's no doubt about it. Setting up a federation trust between two organizations can be a daunting task because of the many sequential steps involved in manually setting up both partners for successful AD FS communications. In this scenario, both administrators are equally responsible for entering in values and addresses (that is, URIs, URLs, and claims) within the AD FS snap-in that are unique to their company's federation environment.

Once this initial setup phase has been completed, each administrator must then provide these values to the administrator in the other organization so that a federation trust can be properly established. Even when these values are sent to the intended partner administrator, there is the distinct possibility that an administrator can accidentally type in a value incorrectly and inadvertently cause himself or herself many hours of headaches trying to locate the source of the problem with the new trust.

In Windows Server 2008, improvements have been made that allow partner administrators to export their generic trust policy and partner trust policy into a small xml file format that can easily be forwarded via e-mail to a partner administrator in another organization. The generic trust policy contains the Federation Server Display Name, URI, Federation Server Proxy URL, and any verification certificate information; whereas the partner trust policy file also includes information about each of the claims. With this in mind, the second-half of the federation trust can then be quickly established by importing the partner's trust policy and mapping the claims.

This "export and e-mail" process adds the following benefits for the partner administrator who receives the xml file:

- Expedites the process of establishing a federation trust because the administrator can choose to import the contents of the xml file in the Add Partner Wizard and simply click through the wizard pages to verify that the imported settings are suitable
- Eliminates the additional step of importing the account verification certificate because the import process does this automatically
- Provides for easy claim mapping
- Eliminates the possibility of manual typing errors

You can test-drive this new functionality by walking through the Windows Server 2008 version of the AD FS Step-by-Step Guide.

–Nick Pierson
Technical Writer of CSD (Connected System Division) UA team

–Lu Zhao
Program Manager, Active Directory Federation Service

–Aurash Behbahani
Software Design Engineer, Active Directory Federation Service

Another new feature of AD FS in Windows Server 2008 is the ability to use Group Policy to prevent setting up unauthorized federation servers in your domain. Here's how some of our experts at Microsoft describe this enhancement:

From the Experts: Limiting Federation Service Deployment Using Group Policy

In Windows Server 2003 R2, AD FS did not provide control mechanisms that prevented users from installing or configuring their own federation service. In Windows Server 2008, AD FS administrators can now turn on Group Policy settings that prevent unauthorized federation servers in their domain. This new setting helps to satisfy the needs of an IT department when they want to enforce compliance or legal process requirements.

Once the Group Policy setting has been enabled, the value *DisallowFederationService* is inserted into the registry key on each federation server in that domain. Before an AD DS domain-joined computer running the Windows Server 2008 operating system can install the Federation Service server role, the server first checks to make sure that the Don't Allow Non-authorized Federation Servers In This Domain Group Policy setting is enabled. If this setting is enabled, the installation of the Federation Service will fail. If it is not enabled, which is the default setting, installation of a Federation Service will be allowed and the installed Federation Service will function normally.

The registry key value is checked only when the trust policy file is loaded, so there might be a delay between when the update appears that brings down the policy and when the Federation Service observes the policy. By default, the policy is read when a file change notification is received and also once every hour.

Note that this feature applies only to Windows Server 2008 federation servers and does not affect new or existing installations of a Federation Service in Windows Server 2003 R2.

–Lu Zhao
Program Manager, Active Directory Federation Service

–Nick Pierson
Technical Writer of CSD (Connected System Division) UA team

Finally, AD FS can be integrated with AD CS, but when problems occur with this scenario you need to know how to troubleshoot them. Here are some more of our experts explaining how to do this:

From the Experts: Troubleshooting Certificate Revocation Issues

Certificate issues are among the top five AD FS troubleshooting hot spots for the product support team here at Microsoft. One particular AD FS-related certificate issue centers on a known routine process that checks for the validity of a certificate by comparing it to a CA-issued list of revoked certificates. This process, in the world of PKI, is known as certificate revocation list (CRL) checking.

The revocation verification setting configured for an account partner on a federation server is used by the federation server to determine how revocation verification will be performed for tokens sent by that account partner. The revocation verification setting of the federation server itself, configured on the Trust Policy node of the AD FS snap-in, is used by the federation server and by any AD FS Web agent bound to the federation server to determine how the revocation verification process will be performed for the federation server's own token signing certificate. The verification process will make use of CRLs imported on the local machine or that are available through the CRL Distribution Point.

When troubleshooting certificate issues, it is important to be able to quickly disable revocation checking to help you locate the source of the problem. For example, this can be helpful in deployment scenarios where there are no CRLs available for the token-signing certificates.

To help troubleshoot CRL-checking issues, the AD FS product team has provided a method within the AD FS snap-in in Windows Server 2008 where you can adjust or disable how revocation checking behaves within the scope of a federation service. For example, you can set revocation checking to check for the validity of all the certificates in a certificate chain or only the end certificate in the certificate chain.

–*Nick Pierson*
Technical Writer of CSD (Connected System Division) UA team

–*Lu Zhao*
Program Manager, Active Directory Federation Service

–*Aurash Behbahani*
Software Design Engineer, Active Directory Federation Service

–*Marcelo Mas*
Software Design Engineer in Testing, Active Directory Federation Service

Active Directory Rights Management Services

The last (but certainly not least) IDA component in Windows Server 2008 that we'll look at is Active Directory Rights Management Service (AD RMS). As we mentioned at the beginning of this chapter, AD RMS is the follow-up to Windows RMS. Windows RMS is an optional component for the Windows Server 2003 platform that can be used to protect sensitive information stored in documents, in e-mail messages, and on Web sites from unauthorized viewing, modification, or use. AD RMS is designed to work together with RMS-enabled applications such as the Microsoft Office 2007 System and Internet Explorer 7.0, and it also includes a set of core APIs that developers can use to code their own RMS-enabled apps or add RMS functionality to existing apps.

AD RMS works as a client/server system in which an AD RMS server issues rights account certificates that identify trusted entities such as users and services that are permitted to publish rights-protected content. Once a user has been issued such a certificate, the user can assign usage rights and conditions to any content that needs to be protected. For example, the user could assign a condition to an e-mail message that prevents users who read the message from forwarding it to other users. The way this works is that a publishing license is created for the protected content and this license binds the specified usage rights to the piece of content. When the content is distributed, the usage rights are distributed together with it, and users both inside and outside the organization are constrained by the usage rights defined for the content.

Users who receive rights-protected content also require a rights account certificate to access this content. When the recipient of rights-protected content attempts to view or work with this content, the user's RMS-enabled application sends a request to the AD RMS server to request permission to consume this content. The AD RMS licensing service then issues a unique use license that reads, interprets, and applies the usage rights and conditions specified in the publishing licenses. These usage rights and conditions then persist and are automatically applied wherever the content goes. AD RMS relies upon AD DS to verify that a user attempting to consume rights-protected content has the authorization to do so.

AD RMS has been enhanced in several ways in Windows Server 2008 compared with its implementation in Windows Server 2003. These enhancements include an improved installation experience whereby AD RMS can be added as a role using Server Manager; an MMC snap-in for managing AD RMS servers rather than the Web-based interface used in the previous platform; self-enrollment of the AD RMS cluster without the need of Internet connectivity; integration with AD FS to facilitate leveraging existing federated relationships between partners; and the ability to use different AD RMS roles to more effectively delegate the administration of AD RMS servers, policies and settings, rights policy templates, and log files and reports.

Conclusion

Identity and access is key to how businesses communicate in today's connected world. Active Directory in Windows Server 2008 is a significant advance in the evolution of a single, unified, and integrated IDA solution for businesses running Windows-based networks that need to connect to other businesses that are running either Windows or non-Windows networks. Keeping the big picture for IDA in mind helps us to see how all these various improvements to Active Directory work together to provide a powerful platform that can unleash the power of identity for your enterprise.

I know, the Marketing Police are knocking at my door after that last sentence and they want to get me for that one. But whether it sounds like marketing gobbledygook or not, it's true!

Additional Resources

The starting point for finding information about all things IDA on Microsoft platforms is *http://www.microsoft.com/ida/*. Although this link currently redirects you to *http://www.microsoft.com/windowsserver2003/technologies/idm/default.mspx*, I have a feeling this will change as Windows Server 2008 approaches RTM.

The Windows Server 2008 main site on Microsoft.com also has a general overview called "Identity and Access in Windows Server Longhorn" that you can read at *http://www.microsoft.com/windowsserver/longhorn/ida-mw.mspx*. By the time you read it, there probably will be more details on the site than there are at the time of writing this.

You can also find a developer-side overview of the directory, identity, and access services included in Windows platforms (including Windows Server 2008) on MSDN at *http://msdn2.microsoft.com/en-us/library/aa139675.aspx*.

If you have access to the Windows Server 2008 beta program on Microsoft Connect (*http://connect.microsoft.com*), you can get a lot of detailed information about AD DS, AD CS, AD FS, and so on. First, you'll find the following Step-By-Step guides (and probably others will be there by the time you read this):

- Installing, Configuring, and Troubleshooting OCSP
- Auditing Active Directory Domain Services Changes
- Active Directory Domain Services Backup and Recovery
- Planning, Deploying, and Using a Read-Only Domain Controller
- Restartable Active Directory

- Certificate Settings
- Active Directory Rights Management Services
- Identity Federation with Active Directory Rights Management Services
- Active Directory Domain Services Installation and Removal
- Active Directory Federation Services

Be sure also to turn to Chapter 14, "Additional Resources," for more sources of information concerning the Windows server core installation option, and also for links to webcasts, whitepapers, blogs, newsgroups, and other sources of information about all aspects of Windows Server 2008.