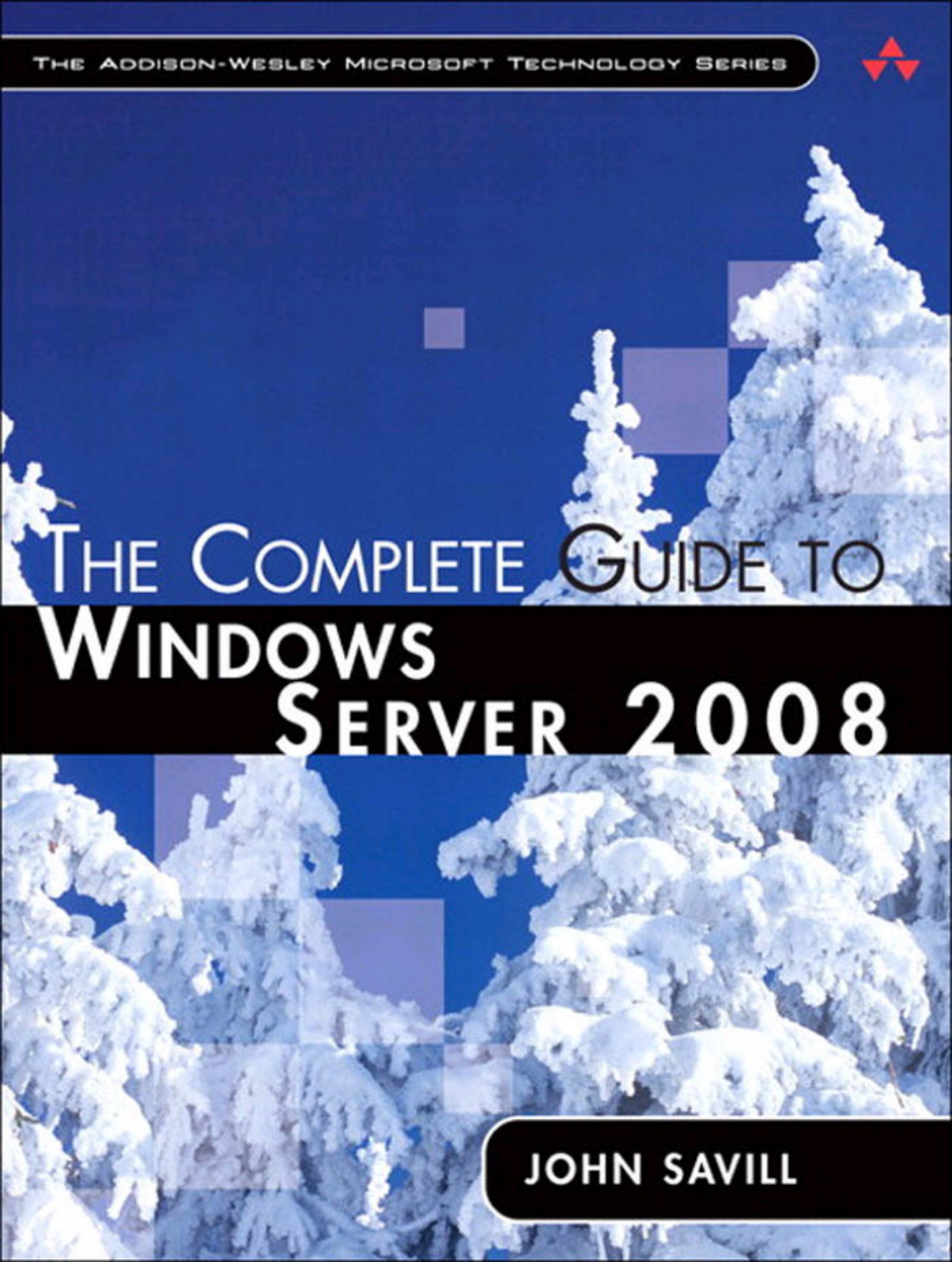
THE ADDISON-WESLEY MICROSOFT TECHNOLOGY SERIES



THE COMPLETE GUIDE TO WINDOWS SERVER 2008

JOHN SAVILL

Library of Congress Cataloging-in-Publication Data:

Savill, John, 1975-
The complete guide to Windows server 2008 / John Savill.
p. cm.
ISBN 0-321-50272-8 (pbk. : alk. paper) 1. Microsoft Windows server. 2. Operating systems (Computers)
I. Title.

QA76.76.O63S35654 2008
005.4'476—dc22

2008025996

ISBN-13: 978-0-321-50272-8
ISBN-10: 0-321-50272-8
Text printed in the United States on recycled paper at Edwards Brothers in Ann Arbor, Michigan.
First printing September 2008

PREFACE

Everyone knows the saying, "Be careful what you wish for." It had long been my goal to write a complete guide to Windows Server, but I never felt I had sufficient time to do justice to the subject. In the middle of 2006, I convinced myself that I could organize my time to allow the undertaking of writing a book on the largest Microsoft server release ever—from scratch. I started writing the book a few months later and finished the final copy editing in June 2008, basically two years from start to finish. Fortunately, Microsoft delayed the release of Windows Server 2008 enough that this book will hit bookshelves while Windows Server 2008 is still new to the market.

With this book, I tried to create a resource that explains the major features of Windows Server 2008, when to use them, how to design the best implementation, and how to manage the deployed environment.

Windows Server 2008 has so many features that I had to leave some out. Those features not discussed are ones I felt would not be interesting to most readers; however, I point out what is not covered and suggest some resources. Windows 2008 is trying to put books out of business; however, although the online help is great, it is task focused. Therefore, I encourage you to follow the online help tool. I concentrate on items that require more design, decision, or are just cool.

Windows Server 2008 is very customer-focused and focuses on a key number of areas such as virtualization, the Web, and security. Usability is also a major area for Windows 2008. A customer does not point to a server and say "that's my windows server"; a customer says "that's my domain controller" or "that's my file server." Windows Server 2008 is designed around how the server is used. Only the basic functions are installed; additional components are installed as roles, and features are added to the server and their management tools accessed through a single server manager interface.

Design of Microsoft-based systems will change in the future. I predict that the process we perform today to design the best practice implementation for our environment will be automated entirely within ten years—

and I'll need a new day job. Think of the process today: We look at the environment and how to use it and then create a design following experience and best practices. We have a number of tools today to help with this: Best Practice Analyzers that check that an installation follows guidelines; System Center Capacity Planner that allows a designer to input information about locations, users, servers, and bandwidth and then creates a server design that services needs; and Microsoft Solution Accelerators that help create solutions with Microsoft technologies. The next step is bringing these together. System Center Configuration Manager and System Center Operations Manager can ascertain the information needed about an environment. This information can then be automatically fed into Capacity Planner-type solutions to produce a best practice design and periodically verify that the design still meets requirements. With the move to virtualization, the design tools will partner with deployment technologies to automatically build new virtual machines for services, as needed, without administrator intervention. Microsoft already has a direction to this type of environment with the Dynamic Systems Initiative. Our involvement will likely be telling these tools about new initiatives and services needed to know what infrastructure to put in place. New versions of software such as Exchange can be downloaded and applied automatically, assuming organizations still have local servers and software. It's entirely possible everything will be a service offered by a "cloud" on the Internet which companies subscribe to.

So with all of that, why is there snow on the cover? Snow makes anything look calm and beautiful. I hope the cover is calming. If ever you start panicking about content in this book, just stop and look at the cover. Like they said in the book *The Hitchhiker's Guide to the Galaxy*, "Don't panic."

Audience for This Book

I've written this book with the IT administrator and architect in mind. Although a background from Windows and networking in general is advantageous, I introduce the basics of each subject, explain how the technologies work, and then build on that transferred understanding until we get to advanced concepts and best practices.

This is not a Microsoft Certified IT Professional study guide, although I did take the exams for both the MS ITP Server Administrator and Enterprise Administrator without studying. I used what I knew from writing this book and easily passed all the exams with high marks. So if you

understand and can apply the information in this book, I would expect you to do well on the Microsoft exams.

This Book's Organization

It would be great if you could sit and read this book from start to finish. Although you may not be able to learn all the features, you may remember items that are possible in day-to-day work and then re-read details of specific features. In the same manner that a chef expects you to eat all courses of a meal instead of picking at each one, I expect this book to be "digested" more like a buffet. You might want to consume the parts relevant to you. I urge you, however, to read a chapter at a time, and not just part of a chapter because each one builds on a subject. In addition, I typically start each chapter with details for you to thoroughly understand the concepts so that we can cover other concepts more quickly.

I want to teach you to drive, not to understand the internal parts of the engine. I'm not big on giving detail on components that don't do you any good from a design or management perspective, but I do give internal details when it aids in learning a technology.

Structure of This Book

This book is made up of 24 chapters:

- **Chapter 1, "Windows 101: Its Origins, Present, and the Services It Provides,"** introduces the major new features of Windows Server 2008. It highlights the key differentiators between the editions of Windows Server 2008 from Web edition through Datacenter.
- **Chapter 2, "Windows Server 2008 Fundamentals: Navigating and Getting Started,"** walks you through the key interface and management components of Windows Vista and Windows Server 2008. The log-on experience for Windows in both workgroup and domain environments is detailed along with the changes to how the built-in Administrator account is handled in Vista and 2008. The chapter discusses User Access Control and how it impacts how to use Windows. Also, key Windows elements, including the Start menu, task bar, and the system tray, are examined along with the available customizations.

Most of your time with Windows Server 2008 is spent in Task Manager, Explorer, and the Microsoft Management Console, so Chapter 2 looks at the major elements of these powerful tools and finishes off with a quick look at the Control Panel.

- **Chapter 3, "Installing and Upgrading Windows Server 2008,"** walks you through the basic system requirements of Windows Server 2008 in terms of memory, processor, and disk space. Windows Server 2008 has a number of activation options, and this chapter looks at both Multiple Activation Keys and Key Management Service.

  The next section walks through performing an upgrade from Windows Server 2003 SP1 to Windows Server 2008, and the various options and limitations associated with an in-place upgrade. The chapter ends with automating local installations using XML answer files.
- **Chapter 4, "Securing Your Windows Server 2008 Deployment,"** discusses security. It looks at authentication and authorization methods, along with the importance of the physical environment that houses your servers. It also discusses BitLocker and how to use it most efficiently.

  This chapter also looks at the built-in certification service in Windows Server 2008, Active Directory Certificate Services (ADCS), and how it is used in (and out) of an organization.

  Finally, Chapter 4 discusses the Security Configuration Wizard and the Security Configuration and Analysis tool that can increase the security of an environment. Increasing network security is handled via the Windows Firewall and IPsec, which this chapter details, along with more information on the User Access Control.
- **Chapter 5, "File System and Print Management Features,"** looks at the facilities that the Windows Server 2008 platform provides for the critical storing of an organization's data. After discussing the new capabilities of NTFS, this chapter looks at creating and managing volumes for data storage. The file permission and ownership capabilities are explained and the concept of shares are introduced and walked through. Then, more advanced subjects are covered, including using quotas to control how much data users can store, file screening technologies to control how the storage is used, and reporting capabilities.

  The second section of Chapter 5 deals with print management, which has taken some big steps in Windows Server 2008. For the

deployment of printers to users, Group Policy can now be used to assign printers to users based on their physical location so that as a user moves, he can be assigned printers that are physically close to him. The chapter closes with a detailed look at printer configuration options.

- **Chapter 6, "TCP/IP,"** starts from the ground up with Internet Protocol (IP). Network Address Translation (NAT) is explored as a means for sharing public IP addresses between multiple computers on a private network. Then, this chapter looks at Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) as methods to provide levels of reliability and extra service to IP communication.

  Chapter 6 rounds off with a look at troubleshooting IP communication through various utilities. It also looks at tracing network traffic, which is invaluable for resolving issues and understanding more complex protocols.
- **Chapter 7, "Advanced Networking Services,"** looks at two main capabilities that make the Internet Protocol more usable and manageable in an environment: Dynamic Host Configuration Protocol (DHCP) and Domain Name System (DNS). The chapter ends with a brief look at WINS and how its capabilities are hopefully no longer required.
- **Chapter 8, "Remote Access and Securing and Optimizing the Network,"** looks at extending the visibility of our enterprises resources to external users in a controlled manner via a virtual private network (VPN). It also looks at the different types of VPN that are available and the pros and cons of each. NAT is explained and its impact on VPNs explored.

  Finally, Chapter 8 looks at one of the major features in Windows Server 2008: Network Access Protection (NAP). It walks through the various types of NAP available, how to use NAP, and how best to configure it. It looks at implementation options for NAP to ensure the most secure environment while minimizing potential impact to the organizations users, thus, avoiding business impact.
- **Chapter 9, "Terminal Services,"** kicks off with an overview of Terminal Services (TS) before walking through the basic steps to enable Remote Desktop and then use Remote Desktop. New security features related to Remote Desktop are examined. Licensing is key with TS, and licensing options are documented and advice given

on which of the licensing modes work in different types of organizations.

The next section looks at installing the full TS role in Windows Server 2008 and its role services, which include TS Gateway for access over SSL and Remote Applications to enable seamless application execution on a terminal server without having a full desktop on the remote server visible. Tied in with Remote Applications, the chapter looks at TS Web, which gives a Web-based portal to launch remote applications.

As TS becomes more important in an organization, it will be necessary to ensure that users can get sessions and good responses, so that multiple terminal servers are pooled together into a farm. Chapter 9 looks at the technologies to facilitate terminal server farms.

- **Chapter 10, "Active Directory Domain Services Introduction,"** looks at the history of domains in Windows and the basic building blocks of Active Directory Domain Services (ADDS). It looks at trust relationships and how they are a core part of Active Directory (AD) hierarchical structure. The chapter then expands on the structure of ADDS by looking at features such as Organization Units, Global Catalog servers, and the special Flexible Single Master of Operations (FSMO) roles.

  Replication is key to ADDS, and this chapter looks at the site components that document to ADDS the physical structure of the environment, the subnets for each location, and the links between each location. Chapter 10 ends with a look at the various domain and forest modes that enable additional features.

- More advanced AD concepts are explored in **Chapter 11, "Designing and Installing Active Directory."** This chapter begins by adding a replica domain controller to an existing domain to give the domain high availability and support for more users and distributed environments.

  For Windows Server Core installations and automated AD deployments, an unattended approach is required. The unattended answer format is explored along with an easy way to create the answer file that is new in Windows Server 2008.

  Management functions related to the FSMO domain controllers are explored, including normal movement of FSMO actions and exception FMO movement options. The last setting the chapter looks at is Global Catalog creation.

The next section deals with creating a new domain, but more importantly, the reasons of when and why a new domain is created. Steps related to verifying a new domain controller are described. The chapter then looks at demoting a domain controller to a normal member server.

One of the major new features in Windows Server 2008 is the Read-Only Domain Controller (RODC); the chapter looks in detail at the capabilities of RODC, its usage considerations, and the restrictions. Chapter 11 closes with a detailed look at the various types of trust relationships and how to create them.

- **Chapter 12, "Managing Active Directory and Advanced Concepts,"** looks at managing AD, backing up and restoring the AD, and other more advanced features. It looks at AD management tools, both graphical and command line-based.

  This chapter also looks at how backing up the AD has changed in Windows Server 2008, using new AD snapshots and restoring deleted objects.

  Chapter 12 closes with a look at changing the replication technology from FRS to DFS-R when you are running a pure Windows Server 2008 domain controller environment.
- **Chapter 13, "Active Directory Federated Services, Lightweight Directory Services, and Rights Management,"** deals with the other role services that complement ADDS, namely Active Directory Lightweight Directory Services (AD LDS), Active Directory Rights Management Services (AD RMS), and Active Directory Federated Services (AD FS).
- **Chapter 14, "Server Core,"** starts with an overview of server core followed by how to perform a Windows Server 2008 installation for a server core instance. When the installation is complete, the hard part is configuring and managing because you don't have the same local graphic tools available that are normally present in a full Windows Server 2008 instance.

  The various command line utilities are explored to perform configuration in addition to walking through configurations that can be done with limited graphical tools such as the Registry editor. Along with this configuration, the chapter explores how to keep a server core patched and what applications can be installed on a server core installation. Finally this chapter looks at managing a server core installation.

- **Chapter 15, "Distributed File System,"** discusses one of the greatest challenges in a distributed environment: managing data and making the data available to all users in a timely fashion. The Distributed File System (DFS) consists of two components: Distributed File System Namespace (DFSN) and Distributed File System Replication (DFSR).
  Chapter 15 closes with a look at best practices to design a DFSR topology and how to troubleshoot and monitor the overall health of replication.
- **Chapter 16, "Deploying Windows,"** starts with a brief history of deployment and then introduces the technologies required to deploy modern operating systems (OSs).
  Installing and configuring WDS is covered, along with the considerations of running WDS and DHCP together and separately. After WDS installation is explained, the process of importing images is introduced and the process discussed. Automated installations are key in large environments, and the process of creating an unattended answer file using the Windows System Image Builder is explained.
  Chapter 16 also looks at creating custom images from reference installations and then maintaining the images by installing fixes, additional drivers, and even language packs. Finally, multicast deployments are explored.
- **Chapter 17, "Managing and Maintaining Windows Server 2008,"** looks at the major tasks and utilities that relate to managing and maintaining Windows Server 2008. The majority of the chapter is spent exploring Server Manager: how to manage the roles and features of Windows Server 2008 using Server Manager and, more than just management actions, how Server Manager gives consolidated insight into each role and is a go-to point to troubleshoot.
  Chapter 17 then looks at Windows Server Backup, the major changes in Windows Server 2008, and details on the Volume Shadow Copy Service (VSS).
  Patch Updates are critical to keeping your environment healthy and secure. The chapter looks at the options for patching systems, their advantages and disadvantages, and finally, the Registry.
- **Chapter 18, "Highly Available Windows Server 2008,"** looks at the two high availability features of Windows Server 2008: Network Load Balancing (NLB) and Failover Cluster. Validating hardware for Windows Server 2008 clustering is shown, as well as the process

to create and manage a Failover Cluster. Chapter 18 finishes with the migration options from a Windows 2003-based failover cluster.

- **Chapter 19, "Virtualization and Resource Management,"** focuses on two main virtualization technologies: machine virtualization and the new hypervisor-based virtualization solution in Windows 2008, Hyper-V, including how to install Hyper-V, and best practices of configuring and managing. We then complete the section with a look at high availability solutions for Hyper-V through failover-clustering.

  Chapter 19 closes with Windows Server Resource Management. It is not a virtualization technology but allows multiple applications/services to be run on a single OS instance while allocating a specific amount of memory and processor to processor. This allocation of resources allows performance guarantees to be made when consolidating multiple OSs running an application, down to a single OS instance running multiple applications.
- **Chapter 20, "Troubleshooting Windows Server 2008 and Vista Environments,"** starts with the basic building blocks of the OS in terms of processes, threads, jobs, and handles—these are key items that are manipulated when troubleshooting. The chapter looks at the boot options for Windows and then delves into the Windows Recovery Environment (RE) that fixes problem systems from outside of Windows.

  The Reliability and Performance Monitoring interface gives access to performance attributes of an OS instance in addition to a historical view of issues on the system for a general "health" view.

  The Event Viewer is covered extensively because it is the main portal to see what is going on in the Windows installation. When there are problems, an event log is typically written to see the system events, pertinent event logs, and how to receive specific event logs from other systems in our environment.

  Chapter 20 closes with a look at System Center, which has solutions that help monitor an environment and preemptively troubleshoot and resolve issues before users are impacted. It's better to fix something before it's a problem.
- **Chapter 21, "Group Policy,"** starts with an overview of Group Policy, its architecture, and basic usage, before going into detail about the Group Policy Management Console (GPMC), the tool of choice for group policy management. Using the GPMC, advanced

concepts are covered, such as using no override, block inheritance, and filtering capabilities. Resultant Set of Policy features are explored that help ascertain how policy is applied for a user/computer and how policy is applied in different circumstances, for example, if the user was moved to another Organizational Unit.

Chapter 21 then looks at features that are new to Windows Server 2008, including the new Starter GPO functionality and Group Policy Preferences capability to set initial configurations for a computer that the user can override.

- **Chapter 22, "The Command Prompt and PowerShell,"** kicks off with a look at the old style command prompt (cmd.exe) environment with information on customization and how to access and set environment variables, before moving onto more advanced concepts such as chaining commands and redirecting output.

  The Windows Scripting Host is explored as a way to create more complicated sequences of logic with some VBScript examples.

  PowerShell is explored with focus on its structure and capabilities for forming complex action sequences. PowerShell can interact with the environment including system processes, the Registry, and file systems. Scripting with PowerShell is explored and some scripts are showcased to further explain capabilities and error handling features.
- **Chapter 23, "Connecting Windows Server 2008 to Other Environments,"** discusses integration with UNIX and NetWare, an important capability in mixed environments. Windows Server has capabilities to integrate and migrate with both UNIX and NetWare environments
- **Chapter 24, "Internet Information Services,"** looks at the Internet Information Services role in Windows Server 2008. The chapter starts with the new architecture that is a radical change from in previous versions, giving administrators and developers greater power to customize IIS processing.

  The configuration of IIS is explained, as well as the various levels of configuration made possible by the new configuration architecture of IIS 7. The process of IIS role service installation is shown along with the steps required to create and access new web sites.

  The chapter looks at new capabilities in IIS 7 including URL authorization that allows specific users to access a site and new management delegation capabilities. IIS is one of the roles supported by

Windows Server Core, and the restrictions associated with this IIS support are communicated. Chapter 24 concludes with the Windows Web Server 2008 SKU.

Code and Command Entry

Some code statements presented in this book are too long to appear on a single line. In these cases, a code continuation character (➥) indicates that the following line is a continuation of the current statement. Scripts can be found at www.savilltech.com/completeguidetowindows2008.

CHAPTER 14

SERVER CORE

This chapter looks at a new feature in Windows Server 2008, an installation option known as a Server Core installation (as opposed to a full installation). Windows Server Core is not a separate product or even a separate license; it simply installs a bare metal server installation with the components needed to run a small set of core network roles, such as domain controller service and file service features, without everything else…which includes the exclusion of the familiar Windows Explorer GUI. You start with a detailed overview of Windows Server Core, which details the roles that it can fulfill, and then you look at ways to install Server Core. Finally, you learn how to configure and use your GUI-less Server Core environment.

Overview of Windows Server Core

In nearly all environments today, servers are designated for a single purpose. Often when you go to a client's site, the conversation is "these are the domain controllers, here are the file servers" and so on. Microsoft recognizes this specialization of servers. This recognition can be seen in the role-based nature of Windows Server 2008. However, even though your domain controllers, for example, need only a limited number of services to function (and maybe domain name system [DNS]), the server has a plethora of unneeded components. These components bloat the server, requiring the server to have more resources to function than are needed for its main function. Most importantly, the more components the system has installed, the more possible vulnerabilities it has. The more components there are, the greater the attack surface and the more patches required, resulting in more management overhead.

The typical server has the full .NET Framework, Internet Explorer, Media Player, and Outlook Express, all of which will likely never be used but still have to be managed.

With Windows Server Core, the "extra" parts of Windows Server 2008 have been removed, leaving a much thinner core operating system than with a normal Windows Server 2008 full installation. Because it has far fewer components, you benefit from having a reduced attack surface and less to manage and maintain. Server Core has only the critical components of the operating system necessary to support the various roles and features made available on a Windows Server Core installation. Many of the non-value-add legacy and client components are missing from Server Core.

This much smaller footprint, and optimized installation based around specific roles such as a domain controller or file server, means the following:

- As already discussed, Server Core presents less attack surface because it involves fewer components with less possible vulnerabilities.
- Because you have fewer components installed, fewer patches apply to a Server Core installation than to a normal full installation. You often hear of an urgent patch related to an Internet Explorer vulnerability. If Internet Explorer is not installed, you don't need to apply that patch. Microsoft believes there will be a large reduction in the number of patches needed for a Server Core install compared to a full installation. It's not possible to know how many patches will be released for Windows Server 2008 or what components the patches will be applicable to. But if a core version had been available for Windows 2000, it would have required 60 percent fewer patches than a full installation, and if available for Windows 2003 there would have been a 40 percent reduction in patches. The servicing stack in Windows Server 2008 downloads and applies only fixes that apply to components installed on the system. No actions or special Windows Update site is required that is Server-Core-specific.
- Administrators can focus more on their technology area without having to be so worried about general Windows knowledge because all the extra parts are no longer installed.
- With fewer components running, the installation uses fewer system resources and becomes more reliable because the fewer different components executing, the less chance of problems occurring.
- Less disk space. A typical core installation uses 1GB of disk space for the install and additional disk space for its actual operation. In

terms of other resources, there is not a great deal of difference, although obviously with fewer components, fewer resources are used overall. But remember: A Windows 2008 install alone requires 512MB of RAM.

The Server Core is available as an installation option for the Standard, Enterprise, and Datacenter editions of Windows Server 2008 and is available on both the x86 and x64 architectures.

Because the Server Core is a minimal installation of Windows, not all the full Windows Server components can run. For example, because the .NET Framework is not present in Server Core, which in turn means no Common Language Runtime (CLR), no managed code can run. That means no PowerShell. A Server Core installation has many "nots":

- There is no Explorer-based shell, so the Start button, taskbar notification area (system tray), and taskbar are eliminated. There are no fancy wall papers, screen savers (a default screen saver shows the Windows Server 2008 logo), and no Aero Glass. Explorer itself is not available, which means no My Computer. Because you have no system tray, you get no balloon notifications, which also means no password prompts because they are balloon notifications.
- No Explorer means no Internet Explorer, no Search, no Run, and no Help, but you do get Notepad.
- No .NET Framework. This is because the .NET Framework is monolithic, meaning all or nothing. And .NET has a lot of multimedia-related code and other components that do not fit the Server Core model. However, a "core" version of the .NET Framework is expected for the Windows Server 2008 R2 timeframe. This means no managed code, which requires .NET.
- No Microsoft Management Console (MMC), which means no snap-ins either. That is an issue because nearly everything is managed with the MMC.
- Only two Control Panel applets.

So let's get it clear. With Server Core, there is no graphical interface, no management tools, no Explorer, no Control Panel applets? Before you get freaked, this is a great feature. The advantages of the reduced overhead are worth a little hardship. You do have a shell, but it's the command

prompt. However, if you think about it, nearly every MMC snap-in you have today can connect to a remote computer, which helps you manage your GUI-less Server Core installation.

What do you get? Much more than in the early builds of Longhorn when the only roles available were Active Directory Domain Servers (a domain controller), DNS, DHCP and File Servers. You are a lot further than that now. As you've seen, with Windows Server you have roles, which are important components of Windows Server 2008, and features, which are less important than their older, driving Role brothers. Table 14-1 provides a list of the roles and features available in Windows Server Core. Note there are no relationships between the roles and features; they are in a table only to save space.

Table 14-1 Windows Server Core Roles and Features

Server Core Roles	Server Core Features
Active Directory Domain Services (ADDS)	BitLocker Drive Encryption (and remote admin tools)
Active Directory Lightweight Directory Services (formally known as ADAM)	Failover Clustering
DHCP Server	Multipath I/O
DNS Server	NAP Client
File Services	QoS (Qwave)
Internet Information Services (IIS)	Removable Storage Management
Print Services	Simple Network Management Protocol (SNMP) Services
Streaming Media Services	Subsystem for UNIX-based applications
Windows Server Virtualization (Hyper-V)	Telnet Clients
	Windows Process Activation Service
	Windows Server Backup
	WINS Server

Don't forget that Server Core is not a separate operating system. It just takes advantage of the highly componentized nature of Windows Server and deploys only the most critical components. Core still has the same kernel as a normal installation in addition to other core components such as

the Hardware Abstraction Layer (HAL), memory manager, security subsystem, Winlogon, file systems, networking subsystem, Windows File Protection, Distributed Component Object Model (DCOM), and remote procedure call (RPC), and device drivers for NIC, disk, and basic video. Many of the other drivers have been removed from Core, such as audio drivers and modem drivers. However, you can add them manually. Imagine a print server, however; print drivers are also not included with Server Core because Windows Server 2008 has nearly 1GB of printer drivers. Instead of including drivers in Server Core for a role that might not be used, the print drivers are not included. When you enable the Print Server role, the spooler starts, and drivers need to be manually added using the Print Management Console remotely from a Windows Vista/Windows 2008 machine.

Also included are features such as the event log, which is critical to nearly all components of Windows, performance counters, WS-Management for remote management, and Windows Management Instrumentation (WMI).

Think of Server Core as a subset of the full Windows installation. If a core kernel patch is released, the same patch for Windows Server is applicable to a Server Core installation. How do you use this? What do you get to manage this Server Core environment? Let's look at installing Server Core, and then you can see the usable environment.

Installation

Server Core installation does not warrant its own section because it's the same as a normal installation of Windows Server 2008. The install media is placed into the server or the server boots over the network, and a product key is entered that identifies the particular edition of Windows Server 2008. In this case, it needs to be Standard, Enterprise, or Datacenter. The only difference is during the actual installation, after entering the product key, you select the type of Windows Server 2008 installation, full or core, as shown in Figure 14-1.

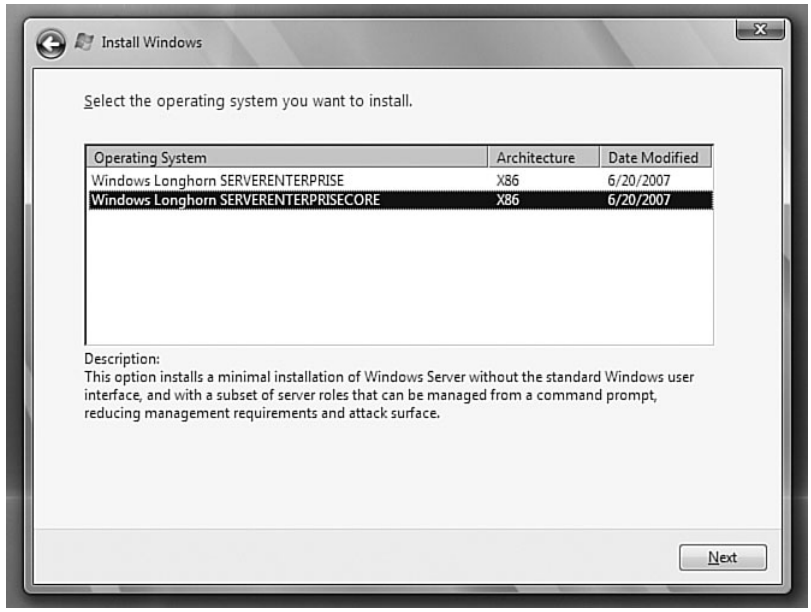


FIGURE 14-1 The description is your first clue that your command-line skills are about to get much better.

When the installation is complete, you get the familiar and comfortable Press Ctrl+Alt+Del to Log On dialog with the pretty Windows Server logo at the bottom. If you press the secure attention sequence, you are prompted to enter logon credentials, so for now all you can do is log on with the administrator account with a blank password.

So far, this is looking great, as Figure 14-2 shows. After clicking the logon button, you are prompted to change the password as normal, and you set a new administrator account password. The normal process of applying local policy and preparing the desktop takes place, and then your Server Core desktop loads, as shown in Figure 14-3.

Note that you cannot upgrade from Windows Server 2003 to Server Core; only fresh installations of Server Core are supported. You also cannot upgrade from Server Core to the full Windows Server 2008 product, nor can you downgrade from Windows Server 2008 to Server Core. If you need to switch between versions, perform a clean installation.

FIGURE 14-2 So far this Server Core environment looks familiar.

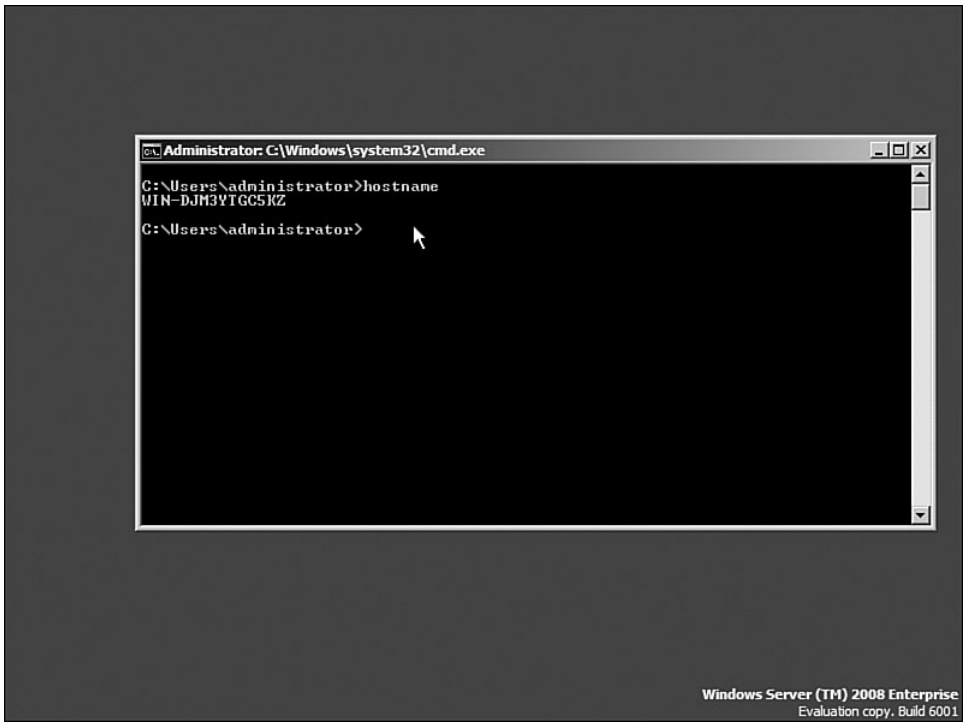


FIGURE 14-3 In keeping with trends from server-based computer to client/server back to server-based, you are now back to a command prompt server environment.

Server Core Configuration

Now that Server Core is installed, first you need to configure it. Without the normal graphical elements, you don't have the nice Initial Configuration Tasks (ICT) interface that you would normally use to configure Windows Server 2008 server, so you have two options:

- Manually configure the server using command-line tools.
- Automate the configuration using answer files during the actual installation.

The second option is the way to go for any sizable deployment. One of the big pushes of the latest operating system has been zero-touch deployments, so you can automate the install and configuration of all the main components. However, this does take up-front effort and planning but is definitely an option. Other areas of the book talk about unattended installations, so for now concentrate on the manual configuration of the server. However, if you go the unattended route, Server Core uses the same unattended syntax as Windows Vista and a normal server. Use the Windows System Image Manager from the Windows Automated Installation Kit (WAIK) to help create the unattended eXtensible Markup Language (XML) answer file. There are some advantages to using the unattended XML, however, because some items are quite hard to configure in Server Core. For example, configuring screen resolution is quite complex without the Display Control Panel applet! The display options are part of the Microsoft-Windows-Shell-Setup component, and a sample code extract for an unattend.xml is shown here:

```
<settings pass="oobeSystem">
    <component name="Microsoft-Windows-Shell-Setup"
processorArchitecture="x86" publicKeyToken="31bf3856ad364e35"
language="neutral" versionScope="nonSxS"
xmlns:wcm="http://schemas.microsoft.com/WMIConfig/2002/State"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <Display>
            <HorizontalResolution>1280</HorizontalResolution>
            <VerticalResolution>1024</VerticalResolution>
            <ColorDepth>16</ColorDepth>
        </Display>
    </component>
</settings>"
```

If you examine the content of the install.wim file for Windows Server 2008, you see that a CORE version exists for each operating system. If you are using Windows Deployment Services (WDS) or any other XML installation, select the CORE post-fixed version, as shown in Figure 14-4.

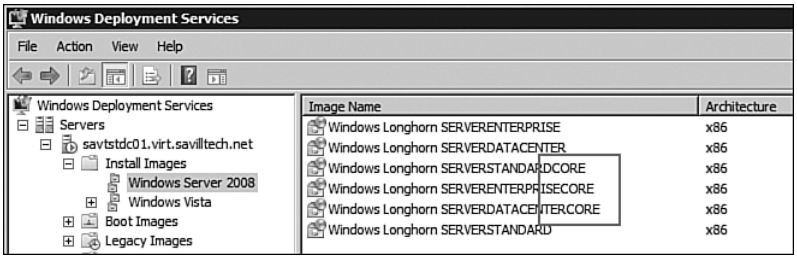


FIGURE 14-4 The core versions of the main Server 2008 editions.

Let's look at the main things you normally do when you configure a new server:

- Set the administrator password.
- Set the machine name.
- Set static TCP/IP v4 details.
- Set the time zone.
- Join a domain.
- Set keyboard and international settings.
- Set the default scripting engine.
- Activate the server.
- Install patches.
- Configure the firewall.
- Configure the server pagefile.
- Enable Remote Desktop.
- Configure hardware.
- Add roles and features.

You would normally do all this via GUI interfaces. For example, you would use Network and Sharing Center to configure IP settings, Windows Update for patches, and so on, but none of these interfaces are available. You can still set all of these things using the command line and some Server Core–specific commands. However, most of these are standard commands and can be used on normal installations for configuration and for scripted communication.

Setting the Administrator Password

The Winlogon and security subsystem in Core is the same as in a standard installation of Windows Server 2008, so to change the password of the logged-on account, just press Ctrl+Alt+Delete as you would normally do. Select the Change a Password link from the menu, and the normal change password dialog displays.

Passwords can also be changed via the `net user` command as on any other Windows installation by passing the username and the new password or passing the wildcard (`*`) character to be prompted for the new password, as shown in Figure 14-5. To change a domain account password, add the `/domain` switch.

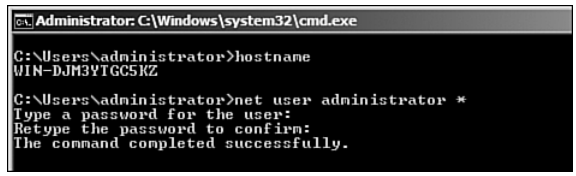


FIGURE 14-5 The `net user` command is an easy way to manage local account passwords.

Setting the Server Name

In the first screen, you viewed the server name using the `hostname` command. However, to change the server name, use the `netdom` command with the `renamecomputer` switch. To avoid having to type in the long default computer name, use the `%computername%` environment variable and then pass the new server name with the `/NewName` switch:

```
C:\Windows\System32>netdom renamecomputer %computername% /New
➥Name:savtstcore01
This operation will rename the computer WIN-DJM3YTGC5KZ
to savtstcore01.

Certain services, such as the Certificate Authority, rely on a
fixed machine name. If any services of this type are running on
WIN-DJM3YTGC5KZ, 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.
```

This change does not take immediate effect; a reboot is required by selecting the Restart option from the Ctrl+Alt+Del screen shutdown options as shown in Figure 14-6 or by using the `shutdown /r /t 0` command. When the reboot is complete, the server has taken the new name, which you can verify by rerunning the `hostname` command.

FIGURE 14-6 Although you don't have a Start menu, access shutdown options via the Ctrl+Alt+Del menu.

Setting Static TCP/IP v4 Information

By default, the new installation has been dynamically allocated an IP address. However, in most cases a server needs static IPv4 information, which can be seen with the `ipconfig /all` command. It will show `DHCP Enabled` set to `Yes`.

Because you can't use the normal Network interface to set the IP properties, instead use the `netsh` command. However, before you can set the

IP properties, check which interface you are configuring. By default your server has two network interfaces: the "real physical" interface and a second IntraSite Automatic Tunnel Addressing Protocol (ISATAP) tunneling interface, which sends IPv6 packets over an IPv4 network by encapsulating the IPv6 packet in the IPv4 header. You need to configure the physical connection and not the ISATAP one, so list your interfaces to identify the index of the physical adapter.

```
C:\Users\administrator>netsh interface ipv4 show interfaces

Idx  Met   MTU   State        Name
---  ---  -----  -----------  -------------------
  2   10   1500  connected    Local Area Connection
  1   50 4294967295  connected    Loopback Pseudo-Interface 1
```

When the adapter is identified, which in this case is index 2, the IP details can be set. They most likely consist of an IP address, a subnet mask, a gateway, and one, possibly two, DNS servers.

To set the IP address, subnet mask, and gateway, run the following and change the information for your environment:

```
C:\Users\administrator>netsh interface ipv4 set address
➥name="2" source=static address=192.168.1.232
➥mask=255.255.255.0 gateway=192.168.1.1
```

You can now add the DNS servers. The primary DNS server gets an index of 1, the secondary DNS server gets an index of 2.

```
C:\Users\administrator>netsh interface ipv4 add dnsserver
➥name="2" address=192.168.1.230 index=1


C:\Users\administrator>netsh interface ipv4 add dnsserver
➥name="2" address=192.168.1.10 index=2
```

If you need to configure primary and secondary Windows Internet Name Service (WINS) servers, use the same syntax as for adding DNS servers but use `winsserver` instead of `dnsserver`. The first index would be the primary WINS server and the second index the secondary WINS server.

If you now examine the IP information with `ipconfig/all`, the configured settings are displayed, as shown in the following example:

```
C:\Users\administrator>ipconfig /all

Windows IP Configuration

   Host Name . . . . . . . . . . . . : savtstcore01
   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-0E-0D-F9
   DHCP Enabled. . . . . . . . . . . : No
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . :
fe80::c49a:b729:8c8b:471e%2(Preferred)
   IPv4 Address. . . . . . . . . . . : 192.168.1.232(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.1.1
   DNS Servers . . . . . . . . . . . : 192.168.1.230
                                       192.168.1.10
   NetBIOS over Tcpip. . . . . . . . : Enabled
```

If you need to remove a DNS server, or more likely a WINS server, after you finally get it killed off, use the `del` keyword instead of `add`. For example:

```
Netsh interface ipv4 del winsserver name="2"
address=192.168.1.10
```

Setting the Time Zone

The date and time are easy to set using the `date` and `time` command lines, but using a command-line method to set the time zone is trickier. There are Registry areas for the time zone. However it's not necessary to use the Registry. Remember that Control Panel is unavailable in Server Core except for two applets. The Date and Time Control Panel applet is one of them; start it via the following command:

```
control timedate.cpl
```

After loading the applet, perform the normal date/time and time zone configurations, as shown in Figure 14-7. Note that in a domain environment, the time synchronizes; however, you might need to set the time zone.

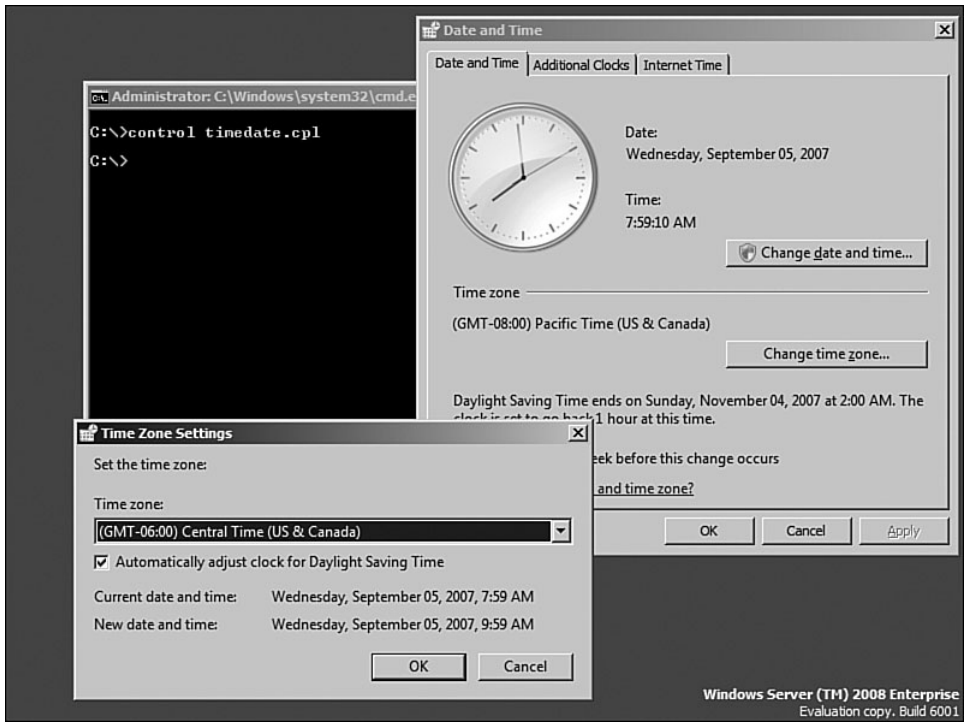


FIGURE 14-7 At last, a graphical way to configure something.

Joining a Domain

It is most likely your servers are part of a domain, and unless the server was preprovisioned during deployment or used an answer file, you need to configure your server to join a domain. After the IP configuration is configured with the correct DNS servers, the computer name is set, and the time configuration is correct, you can join the domain.

To join a domain, use the same command that you used to rename the computer: `netdom`. Full help can be seen by running `netdom join /?`, which gives information on specifying a specific organizational unit (OU) to place the computer into. However, at the most basic level, pass the domain you want to join, the account to use to perform the join, and its password:

```
C:\>netdom join %computername% /domain:virt.savilltech.net
➥/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.
```

Replace the domain name with your domain, and then restart the server. After the reboot, you can log on as a domain user, which confirms the domain join operation worked successfully. You can also verify your connectivity to the domain using the `netdom /verify` command as in the following example:

```
C:\Users\administrator.VIRT>netdom verify %computername%
➥/domain:virt.savilltech.net
The secure channel from SAVTSTCORE01 to the domain VIRT.
SAVILLTECH.NET has been verified.  The connection
is with the machine \\SAVTSTDC01.VIRT.SAVILLTECH.NET.

The command completed successfully.
```

Configuring International Settings

The second Control Panel applet available in Server Core is the Regional and Language Options applet. It enables the configuration of the keyboard layouts, languages, and location. To launch the applet, run the following command and configure as a normal installation:

```
Control intl.cpl
```

Setting the Default Scripting Engine

With Server Core, you do a lot via various scripts executed by the Windows Scripting Host, which has a GUI and a command-line engine. By default the GUI engine is the preferred tool, which goes against the idea of managing Server Core from the command line and requires you to remember to put `cscript` at the start of your scripts to process the script using the command-line interpreter.

To change the Windows Scripting Host to use the command-line interpreter by default, use the following command:

```
C:\Windows>cscript //H:CScript //NOLOGO //s
Command line options are saved.
The default script host is now set to "cscript.exe".
```

If you've enabled `cscript` as the default engine, you don't need to type it every time.

Activating the Server

Server Core includes the Slmgr.vbs script, which when passed with the `-ato` switch, performs an automated activation of the operating system. Slmgr.vbs is not a Server Core feature; it is present in Windows Vista and full Windows Server 2008 deployments and is the main license manager for the Vista/2008 products.

Because Server Core has no taskbar or system tray, you do not receive any prompts to activate the server, so remember to do so shortly after the installation of Server Core.

Before you activate, check your status to see how far into your initial 30-day grace period you are by using the `-xpr` switch as shown here:

```
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 10/4/2007 2:48:10 PM
```

There is also more information available via the `-dli` switch or the `-dlv` switch to get detailed info.

```
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: 2T9PJ
License Status: Initial grace period
Time remaining: 42000 minute(s) (29 day(s))
```

If you have a normal license key or Multiple Activation Key (MAK) that activates with Microsoft, you can go ahead and just activate. However, if you have a local Key Management Service (KMS), tell the activation to use it via the –skms <KMS server> switch. If you need to clear the configured KMS server, use the –ckms switch. If you are using an enterprise license key, use the –ipk <key> switch.

To activate, use the –ato switch as previously mentioned. Rerun the display of license information to see the status is now licensed with no time remaining.

```
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 code name "Longhorn",
ServerEnterpriseCore edition
 (f00d81ce-df2c-47cb-a359-36d652296e56) ...
Product activated successfully.


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: 2T9PJ
License Status: Licensed
```

Installing Patches and Configuring Auto-Update

You can use various methods to patch Server Core. You can push patches with Group Policy or System Center Configuration Manager or any other deployment-type product. You can use Windows Update, which is disabled by default. (You can confirm that with the /au /v switches with scregedit.wsf.) To enable Windows Update to perform the normal 3 a.m. checks, run the following commands. The scregedit.wsf script is Server Core–specific and was written to help perform the functions that are

otherwise difficult to do from the command line. The script is installed automatically on all Server Core installations.

```
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.

C:\Windows\System32>net stop wuauserv
The Windows Update service is stopping.
The Windows Update service was stopped successfully.


C:\Windows\System32>net start wuauserv
The Windows Update service is starting.
The Windows Update service was started successfully.
```

You can force an update pass to run using the following command:

```
C:\Windows\System32>wuauclt /detectnow
```

You can't configure options to download patches and prompt for installation. You can either enable automatic download and application of patches or have automatic update turned off: There is no in-between configuration. You can always check the state of patch installations via the `wmic qfe list` command.

You can manually install patches using the `wusa` command, as in the following example:

```
wusa <patch name>.msu /quiet
```

Remember the patches all have applicability rules, so they won't install if the patch does not apply. If you want to check whether a patch applies, run the command without the `/quiet` switch. If you are prompted to install, it means the patch applies; if you are not prompted, it means the patch does not apply to Server Core and has been ignored. You learn more detail about patching in Chapter 17, "Managing and Maintaining Windows Server 2008."

Configuring the Pagefile

By default, the pagefile is set as managed by the system. This behavior can be modified by disabling the automatic pagefile management and manually configuring a specific pagefile size. For example, the following disables the automatic pagefile management and sets the pagefile to 1GB minimum, 2GB maximum. In general, the default Windows settings for the pagefile should not be changed—do so only if given specific guidance by an expert or vendor of an application being installed. Notice the code in the following listing is using the Windows Management Instrumentation Command-Line (WMIC) environment, which opens up a lot of functionality. Some of the other commands you performed could have been done with the WMIC. After running the commands in this listing, you must restart the server for the changes to take effect.

```
C:\Windows\System32>wmic computersystem set
➥AutomaticManagedPagefile=false
Updating property(s) of '\\SAVTSTCORE01\ROOT\CIMV2:Win32_
ComputerSystem.Name="SA
VTSTCORE01"'
Property(s) update successful.

C:\Windows\System32>wmic pagefileset where name="C:\\
➥pagefile.sys" set InitialSize=1000,MaximumSize=2000
Updating property(s) of '\\SAVTSTCORE01\ROOT\CIMV2:Win32_
PageFileSetting.Name="C
:\\pagefile.sys"'
Property(s) update successful.
```


Configuring the Firewall

On a new Server Core installation, the firewall is enabled by default and blocking almost everything. You can turn off the firewall by using the following command, which opens up the ports and allows Remote Desktop, SNMP, and so forth. You can enable the firewall again by changing `disable` to `enable`.

```
Netsh firewall set opmode disable
```

You can configure the firewall elements using the `netsh` command and its various components. For example, to enable the Remote Desktop, use the following command:

```
C:\Windows\System32>netsh firewall set service
➥type=remotedesktop mode=enable
```

There is an easier way, however. The Windows Firewall MMC snap-in can connect to a remote machine, so let's try that approach as opposed to working out the hundreds of possible `netsh` commands. If you are configuring many servers, however, it would be worth creating a script with the `netsh` commands, or configuring the firewall using Group Policy. If you want to use Group Policy, the firewall is available as part of Computer Configuration, Windows Settings, Security Settings, Windows Firewall with Advanced Security. Right-click Inbound Rules (see Figure 14-8) and select a new rule, and you can use the predefined Remote Administration and Remote Desktop rules. It might not be practical to place the Server Core machines in their own OU for the application of the Group Policy, so you can use a WMI filter to check the `OperatingSystemSKU` of the server for the values 12, 13, and 14, which correspond to the Datacenter, Standard, and Enterprise Server Core installations, respectively. A sample WMI filter follows:

```
select * from Win32_OperatingSystem where OperatingSystemSKU=12
or OperatingSystemSKU=13 or OperatingSystemSKU=14
```

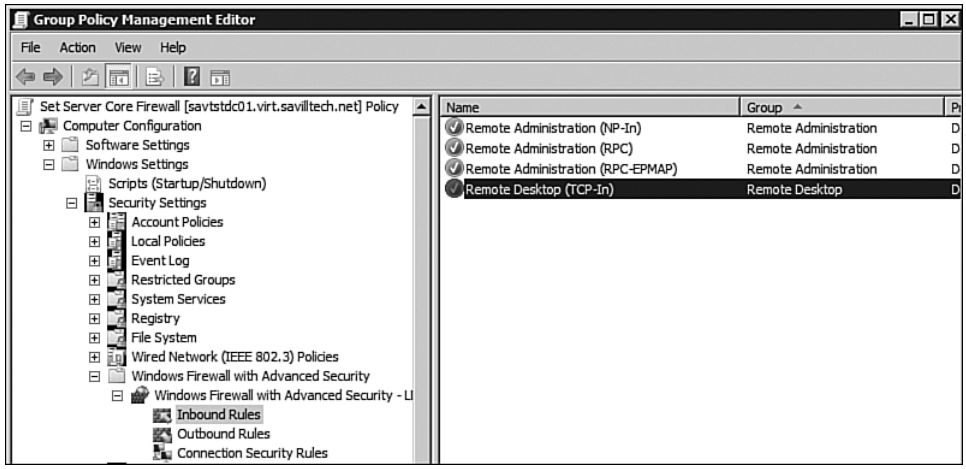


FIGURE 14-8 Using Group Policy to configure the firewall is a good option for larger deployments.

Before you try this, you get an error when you launch the remote firewall snap-in because the firewall you are trying to configure blocks remote management by default. So, you need one more `netsh` command to enable the remote management capability:

```
C:\Windows\System32>netsh firewall set service
➥type=remoteadmin mode=enable
```

Now let's manage remotely:

1. Open a new MMC instance (Start, Run, MMC).
2. From the File menu, select Add/Remove Snap-In.
3. Select Windows Firewall with Advanced Security, and click the Add button (see Figure 14-9).

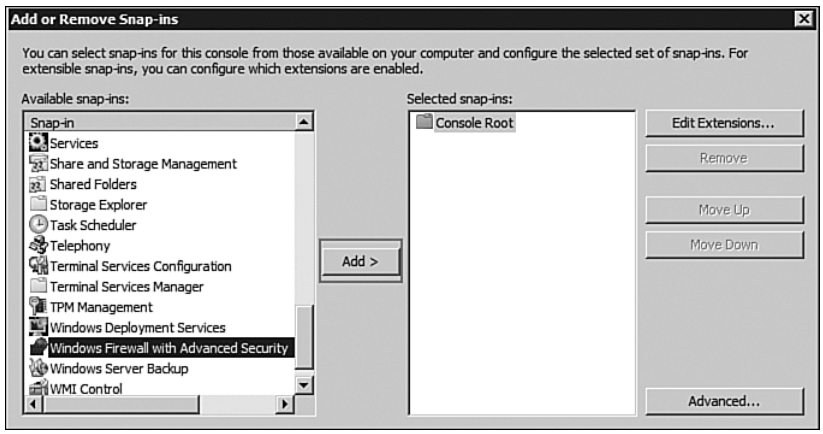


FIGURE 14-9 Select Windows Firewall with Advanced Security.

4. You are prompted to indicate whether the computer is the local computer or another computer. Check Another Computer (see Figure 14-10), specify the name of your Server Core computer, and click Finish.
5. Click OK to close the Add or Remove Snap-Ins dialog box.

Now configure the firewall remotely and enable exceptions as required.

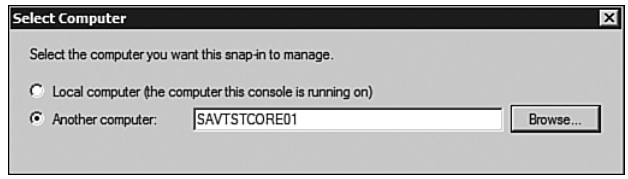


FIGURE 14-10 Check Another Computer.

Enabling Remote Desktop

Server Core contains the Remote Desktop component, which can be a useful way to manage a Server Core environment. But due to its mainly command prompt–based interface nature, there are less resource-greedy ways of managing a Server Core install.

To check the current state of Remote Desktop, use the scregedit.wsf script with the `/ar /v` switches, as shown in the following listing. In this case, by default, the Remote Desktop is disabled because the Deny Terminal Server Connections setting is set to true. You must be in the Windows\System32 folder to run the script:

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

System\CurrentControlSet\Control\Terminal Server
fDenyTSConnections
View Registry setting.
1
```

To enable Remote Desktop, use the `/ar 0` switch:

```
C:\Windows\System32>cscript scregedit.wsf /ar 0
Microsoft (R) Windows Script Host Version 5.7
Copyright (C) Microsoft Corporation. All rights reserved.

Registry has been updated.

C:\Windows\System32>cscript scregedit.wsf /ar /v
Microsoft (R) Windows Script Host Version 5.7
```

```
Copyright (C) Microsoft Corporation. All rights reserved.

System\CurrentControlSet\Control\Terminal Server
  fDenyTSConnections
View Registry setting.
0
```

Additionally, by default, only connections from the newest Remote Desktop Protocol (RDP) clients that support the Credential Security Service Provider (CredSSP) are accepted, which allows the user's current credentials to be automatically passed to the target server. However, you can change this behavior using the `/CS 0` switch with scregedit.wsf.

Configuring Hardware

Some things, such as screen resolution, are difficult to configure from Server Core. One of the few GUI tools provided is the Registry Editor, which means you can perform configurations; it's just a bit ugly. Normally, you are advised to use the Registry Editor only as a last resort, but for some things in Server Core it's your only option. Using the Registry Editor, navigate to the HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Video\<GUID of graphics card>\0000 key. Modify the DefaultSettings.XResolution (see Figure 14-11) and DefaultSettings.YResolution values to the desired values. Just make sure they are right.

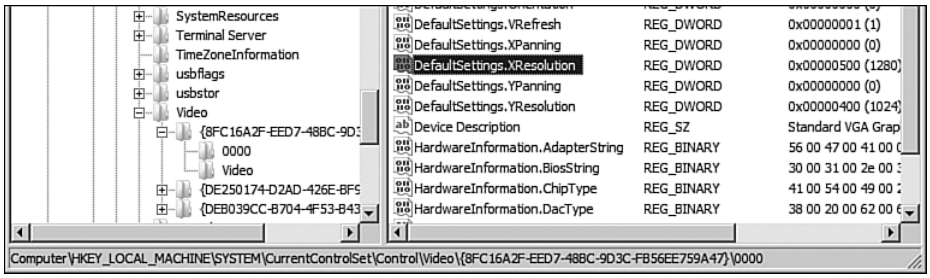


FIGURE 14-11 Setting the screen resolution for the system.

If you want to modify screen saver values, for example, do it in the Registry as well. By default, the screen saver kicks in after 10 minutes and uses the logon screen saver (logon.scr), requiring a password when the

screen saver is deactivated. To modify this, use the Registry Editor again and move to the HKEY_CURRENT_USER\Control Panel\Desktop key. The following values can be changed:

- **ScreenSaveActive**. 1 means screen saver is turned on, 0 disables.
- **ScreenSaverIsSecure**. 1 means password is required, 0 no password needed.
- **ScreenSaveTimeOut**. Time in seconds of inactivity before screensaver starts.
- **SCRNSAVE.EXE**. The name of the screen saver. Logon.scr or scrnsave.scr for the blank screen saver.

You can also specify a background wallpaper by creating a string value named WallPaper under the same key with the full name and path of the image to use as the background.

In terms of adding hardware, if you need to install drivers, you are not prompted to install a driver for new hardware as in a normal Windows Server installation. Instead you need to manually install the driver and then, depending on the hardware, reboot the server for the new driver to be used with the hardware. Copy the driver files to a location on the server and then run the following command to load the driver:

```
Pnputil –i –a <driver>.inf
```

You can list all drivers on the system via the `sc query type= driver` command (note the space between `type=` and `driver`). When you have the service name of the driver, uninstall with the `sc delete <service_name>` command.

Adding Roles and Features

So far everything you have done configures the server. So far it does not do anything; it's not running any roles or features that are the cornerstone of Windows Server 2008.

You don't have access to the normal Server Manager interface to add roles and features, and all the features, except ADDS, are added via the

`Ocsetup` command. `Ocsetup` is a case-sensitive command and is part of all Windows Server 2008 installations. Active Directory installation is installed via the `dcpromo` command, which installs the binaries and configures things via an unattended answer file. You can't use DCPROMO GUI. You have to use an unattended answer file or command-line switches. See the Active Directory chapters for examples of unattended Active Directory installations.

To uninstall roles and features, use the same command but add `/uninstall` at the end. The exception again is ADDS, which once again uses DCPROMO.

Tables 14-2 and 14-3 list the names of the components and what they correspond to in features and roles. However, you can run `oclist` for a complete list; `oclist` is a Server Core–specific command. New roles and features will be added to Server Core in the future. For example, WDS support is expected in the Windows Server 2008 R2 timeframe.

Table 14-2 Server Roles and `Ocsetup` Names

Server Role	`Ocsetup` Name
Active Directory Lightweight Directory Services (ADAM)	DirectoryServices-ADAM-ServerCore
DHCP	DHCPServerCore
DNS	DNS-Server-Core-Role
Distributed File System Service	DFSN-Server
Distributed File System Replication (DFSR)	DFSR-Infrastructure-ServerEdition
File Services	File-Server-Core-Role
File Replication Service (FRS)	FRS-Infrastructure
IIS (no ASP.NET)	IIS-WebServerRole (plus subcomponents visible via oxlist)
Network File System (NFS)	ServerForNFS-Base
Media Server	MediaServer
Hyper-V	Microsoft-Hyper-V

Table 14-3 Server Features and `Ocsetup` Names

Server Feature	`Ocsetup` Name
Backup	WindowsServerBackup
BitLocker Drive Encryption	BitLocker
BitLocker Remote Admin Tool	BitLocker-RemoteAdminTool
Failover Cluster	FailoverCluster-Core
Multipath IO	Microsoft-Windows-MultipathIO
NFS Client	ClientForNFS-Base
Network Load Balancing	NetworkLoadBalancingHeadlessServer
Quality of Service	QWAVE
Removable Storage Management	Microsoft-Windows-RemovableStorageManagementCore
SNMP	SNMP-SC
Subsystem for UNIX-bases applications	SUACore
Telnet Client	TelnetClient
Windows Activation Service (WAS)	WAS-WindowsActivationService
WINS	WINS-SC

By default, if you execute `Ocsetup` with a package to install, the command prompt returns instantly while the installation happens in the background, and you will not know when the install has completed. To work around this, run the `Ocsetup` command after a `start /w` to tell the command to execute and to wait for the execution to complete.

Let's install the DNS Server role, as shown in Figure 14-12. During the install, the TrustedInstaller process is activated and responsible for the actual installation.

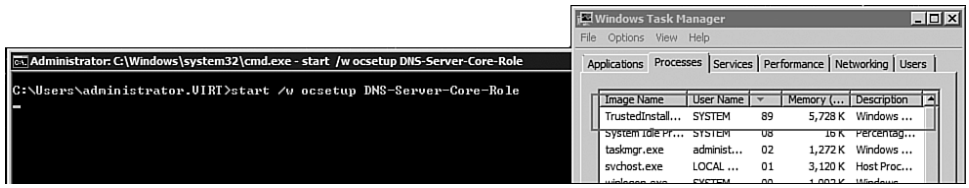


FIGURE 14-12 Installing a role is a one-step process.

After you install the role, it is marked as installed in the Optional Component listing, as shown in the following:

```
C:\Users\administrator.VIRT>oclist
Use the listed update names with Ocsetup.exe to install/unin-
stall 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-ServerFounda-
tion
    Installed:DNS-Server-Core-Role
Not Installed:FailoverCluster-Core
Not Installed:FRS-Infrastructure
```

In the DNS case, the service could be managed locally via DNSCMD, which is a standard part of the DNS role to facilitate command-line management, or more likely you can run the DNS MMC snap-in on a Vista/2008 box and remotely connect and manage the DNS service on the core installation. For example, in Figure 14-13, the root of the DNS navigation node is right-clicked and the Server Core installation is added, which you can now manage with the GUI remotely.

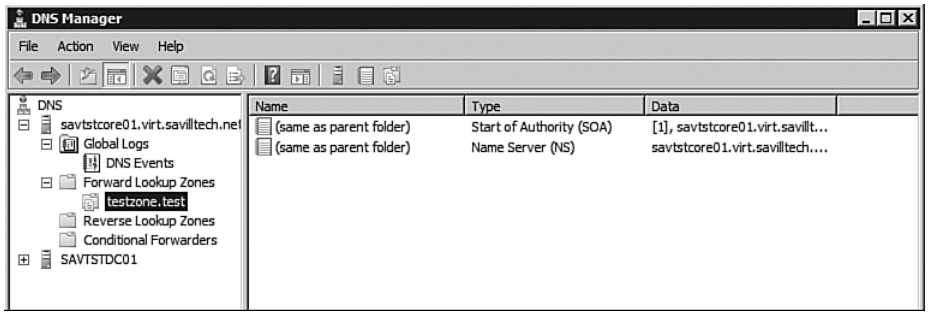


FIGURE 14-13 In reality, you remotely control most of the server core areas of functionality.

As with all the remote GUI tools, if you receive an Access Denied error, solve it by performing a `net use` to the machine before remotely connecting. The command establishes an authenticated session:

```
C:\Users\john>net use * \\savtstcore01.virt.savilltech.net\c$
➥/user:virt\administrator *
Type the password for \\savtstcore01.virt.savilltech.net\c$:
*****
Drive Z: is now connected to
\\savtstcore01.virt.savilltech.net\c$.

The command completed successfully.
```

A better way is to use `cmdkey`, which allows credentials to be set for various target systems:

```
C:\Users\john>cmdkey /add:savtstcore01.virt.savilltech.net
➥/user:virt\administrator /pass:********

CMDKEY: Credential added successfully.
```

Installing Applications

For the Windows Server 2008 release, Server Core is designed to run in-the-box functions, that is, the supported server roles and features and not additional applications.

None of the major products are supported on Server Core; for example, Exchange, SharePoint, SQL, and so on. For additional applications, there is some planning for the future when managed code support is added to Server Core. However, there are limits to what can be added to Server Core; otherwise, it becomes a normal Windows installation.

Agents should be installable and supportable under Server Core, for example, backup agents, Microsoft Operations Manager (MOM), Systems Management Server (SMS) agents, and so on, which are managed via a remote administrative console function. You can install antivirus agents on Server Core installations and manage them remotely. For example, ForeFront runs on Server Core. Virtual machine additions can be installed and they run fine; in fact, they are recommended. The general rule of thumb is that agents have no shell or GUI dependencies and do not require managed code; if all these are true, the agent runs under Server Core.

To install additional software, execute the setup executables or manually install the MSI files using this command:

```
Msiexec /i <application>.msi
```

To check the installed applications, use the `wmic` command and the production function as shown in the following:

```
C:\Windows\System32>wmic
wmic:root\cli>product
AssignmentType  Caption                    Description
1               Virtual Machine Additions  Virtual Machine
Additions
```

This output is long, so you need to scroll to see everything.

To uninstall an application, use the `wmic` command by checking the name of the application and then calling uninstall for it, for example:

```
C:\Windows\System32>wmic product get name /value
Name=Virtual Machine Additions

C:\Windows\System32>wmic product where name="Virtual Machine
➥Additions" call uninstall
```

In the short term, the only installations you do will likely be agents and antivirus, but who knows what the future will bring?

Performing Common Actions Using Server Core

One quick way to get information about your environment is with the `systeminfo` command, as shown executing in the following listing:

```
C:\Windows\System32>systeminfo.exe

Host Name:                 SAVTSTCORE01
OS Name:                   Microsoftr Windows Serverr 2008
Enterprise
OS Version:                6.0.6001 Service Pack 1, v.222 Build
6001
OS Manufacturer:           Microsoft Corporation
OS Configuration:          Member Server
OS Build Type:             Multiprocessor Free
Registered Owner:          Windows User
Registered Organization:
Product ID:                78440-034-0066664-70918
Original Install Date:     9/4/2007, 4:05:28 PM
System Boot Time:          9/9/2007, 6:46:54 PM
System Manufacturer:       Microsoft Corporation
System Model:              Virtual Machine
System Type:               X86-based PC
Processor(s):              1 Processor(s) Installed.
                           [01]: x86 Family 6 Model 15 Stepping
6 GenuineIntel ~
4 Mhz
BIOS Version:              American Megatrends Inc. 080002 ,
2/22/2006
Windows Directory:         C:\Windows
System Directory:          C:\Windows\system32
Boot Device:               \Device\HarddiskVolume1
System Locale:             en-us;English (United States)
Input Locale:              en-us;English (United States)
Time Zone:                 (GMT-06:00) Central Time (US &
Canada)
Total Physical Memory:     1,023 MB
Available Physical Memory: 778 MB
Page File: Max Size:       2,299 MB
Page File: Available:      2,092 MB
Page File: In Use:         207 MB
```

```
Page File Location(s):     C:\pagefile.sys
Domain:                    virt.savilltech.net
Logon Server:              \\SAVTSTDC01
Hotfix(s):                 N/A
Network Card(s):           N/A
```

One item many users struggle with is no system clock, which they get used to in the System Tray. You can update your prompt to include the time with the following prompt command:

```
C:\Windows\System32>prompt [$T]$S$P$G

[10:50:03.26] C:\Windows\System32>
[10:50:04.50] C:\Windows\System32>
```

What else do you normally use on a system? The Task Manager. Its keyboard shortcut still works in Server Core, so press Ctrl+Shift+Esc to open the Windows Task Manager or access it via the Windows Security Dialog by pressing Ctrl+Alt+Del. There is no Windows Task Manager help, however, because the help is based on HTML, which is not included in the Server Core.

What about rebooting, shutting down, and logging off? You can access the Windows Security dialog and elect to shut down or reboot, or you can use the Windows standard shutdown command. The key switches you use are as follows, but you can find full information by running `shutdown /?`.

- `/s`. Shutdown.
- `/r`. Reboot.
- `/t 0`. Wait 0 seconds to perform the action.
- `/a`. Abort a shutdown. This is usable only if you had a time other than 0, so you can type before the reboot/shutdown occurs.

For example, to reboot the computer immediately, use this command:

```
Shutdown /r /t 0
```

To log out, you can use the logoff command-line utility or the Windows Security dialog.

From a utility perspective, both Notepad and Regedit are included in Server Core, RegEdit because you need it and Notepad because customers demanded it. However, neither has help because the help has

dependencies on HTML. As noted previously, HTML is not included. However, the basic Copy, Paste, Find, and other commands all function. The Open and Save dialogs might look familiar, but not in a good way (see Figure 14-14).

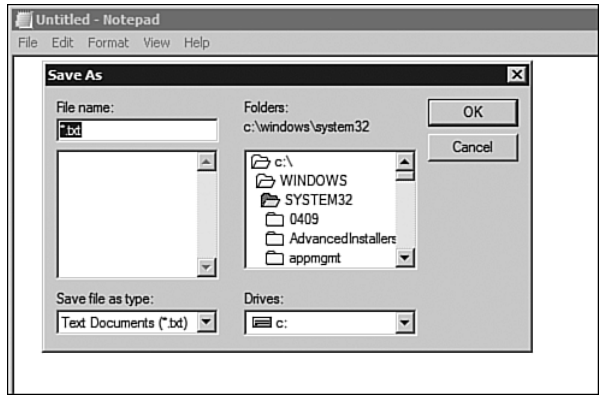


FIGURE 14-14 The days before the new operating system dialogs.

Remotely Managing Server Core

You have seen that you can use the command line for many configuration items, and nearly all Windows components come with command-line tools for management. So, you could manage Server Core locally. However, that is far from ideal, and you turned on the remote admin mode of Terminal Services so that you could remote to the box and access Server Core as if logged on locally. You have also seen how the MMC snap-ins can run on other computers and remotely manage the services on a Server Core installation. For example, the DNS MMC snap-in is probably the most-used remote management method. There are other options which give you the ability to remotely run commands on the Server Core installation—thanks to the inclusion of RPC and DCOM on Server Core, which facilitate the remote administration. Remember to enable the RemoteManage firewall rule.

Three of the MMC snap-ins require additional configuration on the Server Core installation:

- For Device Manager, enable the PnP policy. Even when enabled, Device Manager runs in a read-only mode, which is useful for checking hardware and device driver info. Load the local policy on the Server Core box (or create a Group Policy Object [GPO] that applies to Server Core) and enable the Allow Remote Access to the PnP Interface policy under Computer Configuration, Administrative Templates, System, Device Installation and reboot the Server Core computer, as shown in Figure 14-15.
- The Disk Management MMC snap-in requires two changes. Enable a firewall group on the server core installation and on the machine performing the remote management: `netsh advfirewall firewall set rule group="Remote Volume Management" new enable=yes`
  In addition, run the Virtual Disk Service via this command:
  `net start VDS`
- Enable Remote IPsec Monitor management using the SCRegEdit.wsf script:

```
C:\Windows\System32> cscript scregedit.wsf /IM 1
```

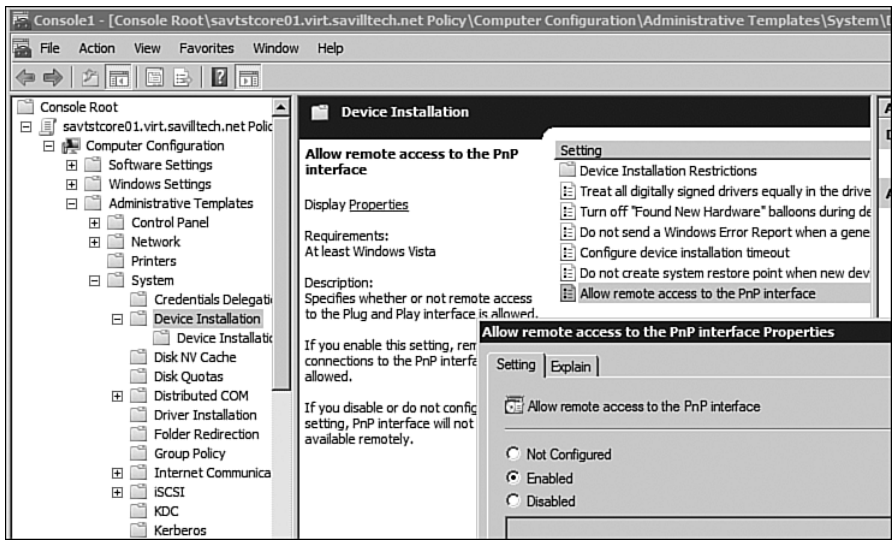


FIGURE 14-15 Enable the Allow Remote Access to the PnP Interface policy.

Another command-line option is the WS-Management and Windows Remote Shell. However, this runs the command, remotely catches the output, and sends it back. The advantage of WS-Management is that it operates over HyperText Transfer Protocol (HTTP) or HyperText Transfer Protocol over Secure Sockets Layer (HTTPS), so there are no additional port requirements for WS-Management to function because the HTTP ports are normally open by default. However, you can change this port if required.

When you enable WS-Management on the server using the quick configuration, the best security method available to the installation is used. For example, on a domain-joined machine, Kerberos is selected. Run the `winrm quickconfig` command as shown in the following listing:

```
C:\Users\administrator.VIRT>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.
Enable the WinRM firewall exception.

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.
WinRM firewall exception enabled.
```

This can be configured via Group Policy through Computer Configuration, Administrative Templates, Windows Components, Windows Remote Management, WinRM Service and enabling Allow Automatic Configuration of Listeners. There are other options in the same policy location regarding the use of Kerberos/Basic authentication.

On the client side, use the `WinRS` command in the following format:

```
WinRS –r:<remote system> command
```

For the remote system, type in the name of the remote computer or enter it in the form of a URL; for example, http://ip address:port or

http://fqdn; http://192.168.1.232:80 or http://savtstcore01.virt.savilltech.com. By default, your existing credentials are used. Credentials are passed using the `/domain`, `/userd` and `/password` arguments. Some sample uses are shown in the following listing:

```
C:\Users\Administrator>winrs -r:savtstcore01 ipconfig

Windows IP Configuration


Ethernet adapter Local Area Connection:

   Connection-specific DNS Suffix  . . :
   Link-local IPv6 Address . . . . . :
fe80::c49a:b729:8c8b:471e%2
   IPv4 Address. . . . . . . . . . . : 192.168.1.232
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.1.1

Tunnel adapter Local Area Connection*:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . . :

Tunnel adapter Local Area Connection* 2:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . . :

C:\Users\Administrator>winrs -r:savtstcore01 cscript
➥c:\windows\system32\scregedit.wsf /AR /v
Microsoft (R) Windows Script Host Version 5.7
Copyright (C) Microsoft Corporation. All rights reserved.

System\CurrentControlSet\Control\Terminal Server
  fDenyTSConnections
View Registry setting.
0
```

You can use the Task Scheduler as in a normal Windows installation, as items such as event logging/forwarding and performance counters, which

you can fully access via the Computer Management MMC running remotely, as shown in Figure 14-16. You can use the reliability interface against Server Core. Note that you can access both the Task Scheduler and Event Viewer through the Computer Management MMC snap-in. You have full access to the local users and groups. (Although you could use the `net user` and `net localgroup` commands to perform user/group management locally.)

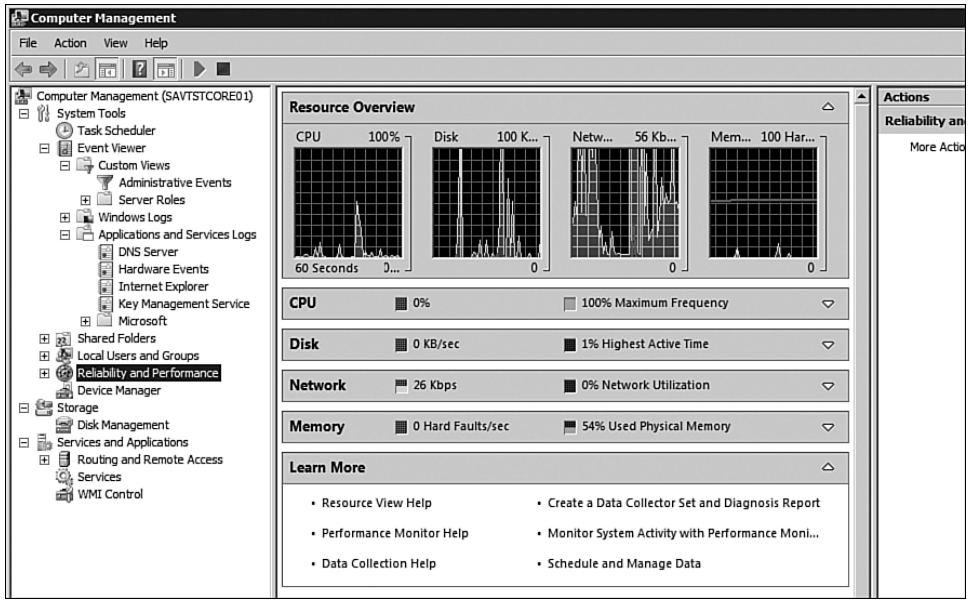


FIGURE 14-16 The remote capabilities of RPC and DCOM give the GUI-less Server Core a great remote GUI experience.

If you want to view the event log locally on a Server Core installation, use the `Wevtutil.exe` command. For example, to view the five most recent event logs in text format from the SYSTEM log, use the following command:

```
C:\Windows\System32>wevtutil qe system /rd:true /c:5 /f:text
Event[0]:
  Log Name: System
  Source: Virtual Disk Service
  Date: 2007-09-09T19:04:09.000
```

```
  Event ID: 4
  Task: N/A
  Level: Information
  Opcode: N/A
  Keyword: Classic
  User: N/A
  User Name: N/A
  Computer: savtstcore01.virt.savilltech.net
  Description:
Service stopped.
```

You can also search for specific event IDs. For example, to search for reboots, which are event ID 1074, use the following command:

```
wevtutil qe system /q:*[System[(EventID=1074)]] /f:text
```

SNMP can be enabled on Server Core to allow management by your management tools if they are SNMP-based. SNMP is enabled by installing the SNMP feature. Both SNMPv1 and SNMPv2c are supported. Normal WMI scripting can be used both locally and, more likely, from a remote station.

Let's look back at the Remote Desktop option. Maybe you want a remote command prompt without a full session. With the new application publishing features of Windows Server 2008, you can publish a command prompt by performing the following actions.

You need to use the Terminal Services (TS) RemoteApp Manager, which is available on a full Windows 2008 installation with the Terminal Server Role installed. So, on a full Windows Server installation, add the Terminal Server role. Notice you need Terminal Server only because you want access to the RemoteApp Manager. Alternatively, you can add the Remote Server Administration Tools feature and select only the Terminal Server Tools option if you don't want to install the Terminal Server role on any server, as shown in Figure 14-17. This latter option avoids installing TS and allows configuration from any platform supporting Remote Server Administration Tools; for example, Windows Vista.

After the TS RemoteApp Manager is running, change the server that the client connects to so that the Server Core machine is selected via the Connect to Computer option in the Action menu of the Actions pane. In the Actions pane, click the Add RemoteApp Programs link, which starts the RemoteApp Wizard (covered in detail in Chapter 9, "Terminal Services"). After clicking Next at the Introduction screen of the wizard, a

list of programs that can be published is displayed. Click the Browse button, navigate to the Windows\System32 folder, and select cmd.exe. Click Next, as shown in Figure 14-18, and then click Finish.

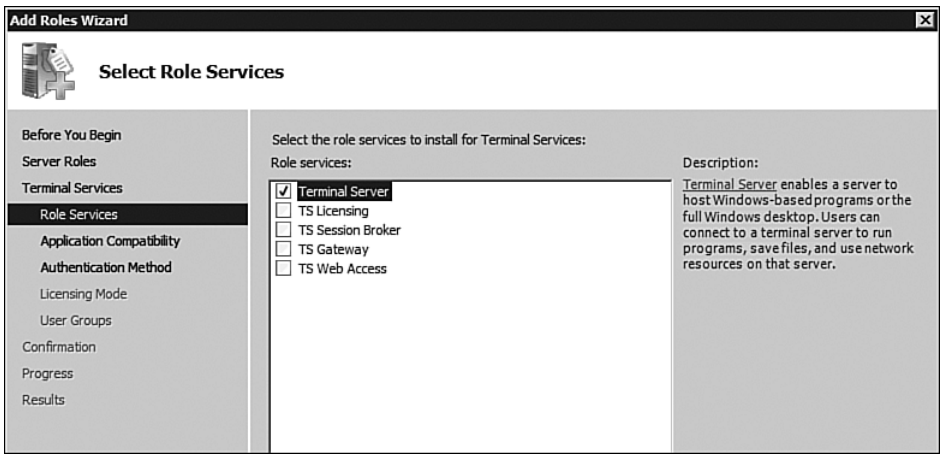


FIGURE 14-17 Adding the Terminal Server role to a server. There's no need for the other components.

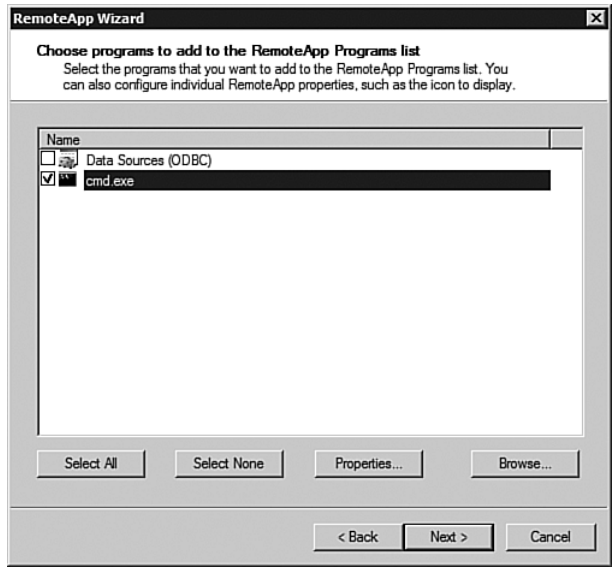


FIGURE 14-18 Publishing the cmd.exe application.

Finally, as shown in Figure 14-19, click the Create .rdp File link. Figure 14-19 shows the options for connecting to a computer and starting the Add RemoteApp Program Wizard.

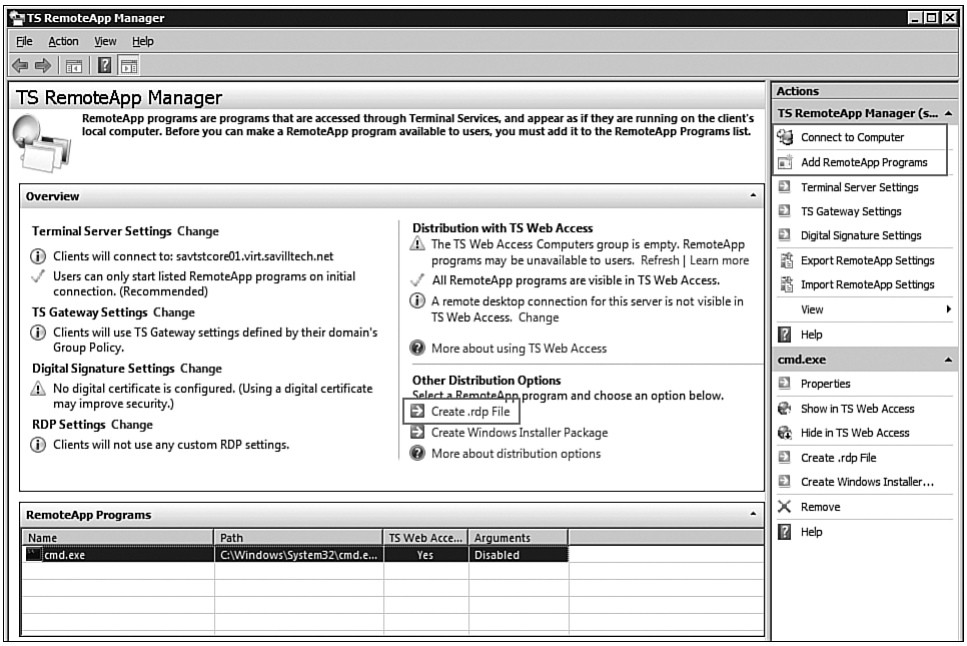


FIGURE 14-19 Creating a published application RDP for the cmd prompt.

You can take the generated RDP file and run it from any Vista/Windows Server 2008 client to open a seamless command window that is running on the Server Core installation. The following listing shows the content of the generated RDP that can be modified with an updated server name:

```
disableclipboardredirection:i:0
redirectposdevices:i:0
redirectprinters:i:1
redirectcomports:i:1
redirectsmartcards:i:1
devicestoredirect:s:*
drivestoredirect:s:*
redirectdrives:i:1
```

```
session bpp:i:32
span monitors:i:1
remoteapplicationmode:i:1
server port:i:3389
allow font smoothing:i:1
promptcredentialonce:i:0
authentication level:i:0
gatewayusagemethod:i:2
gatewayprofileusagemethod:i:0
gatewaycredentialssource:i:0
full address:s:savtstcore01.virt.savilltech.net
alternate shell:s:||cmd
gatewayhostname:s:
remoteapplicationname:s:cmd.exe
remoteapplicationcmdline:s:
screen mode id:i:2
winposstr:s:0,1,424,117,835,356
compression:i:1
smart sizing:i:1
keyboardhook:i:2
audiomode:i:0
redirectclipboard:i:1
displayconnectionbar:i:1
autoreconnection enabled:i:1
prompt for credentials:i:0
negotiate security layer:i:1
remoteapplicationicon:s:
shell working directory:s:
disable wallpaper:i:1
disable full window drag:i:1
allow desktop composition:i:0
disable menu anims:i:1
disable themes:i:0
disable cursor setting:i:0
bitmapcachepersistenable:i:1
```

When using a terminal server connection with the Terminal Server Client (`mstsc`), ensure that disk drives are available from the client machine by selecting the drives as available under the Local Resources tab (you have to click the More button) or as an option, as shown in Figure 14-20. The local client drives are accessible as \\tsclient\<drive> on the remote system. You can then map to these drives to get full access as in the following example:

```
C:\Windows\System32>net use * \\tsclient\d
Drive Y: is now connected to \\tsclient\d.

The command completed successfully.
```

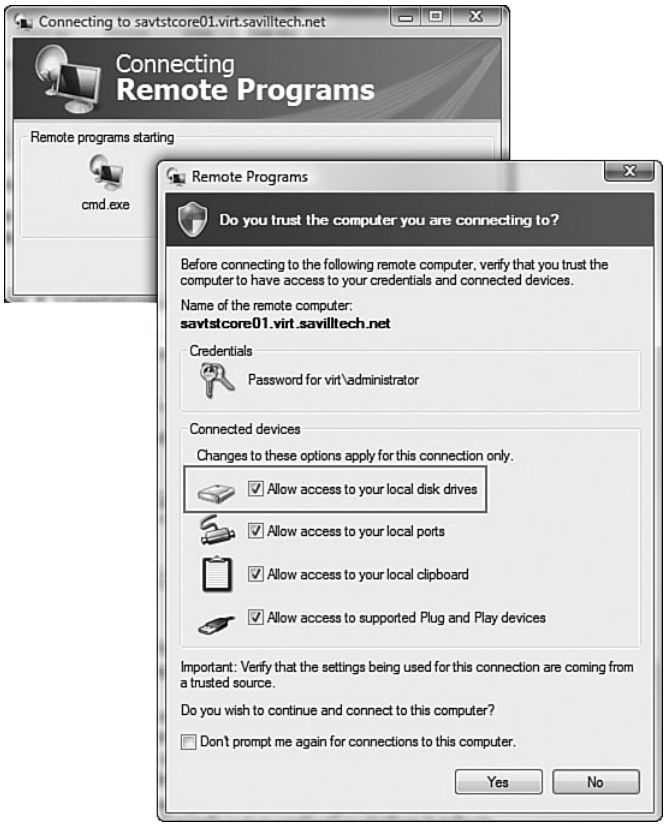


FIGURE 14-20 The Remote Programs client gives easy access to controlling the local resource access.

Summary

Server Core is a great addition to the Windows Server family. If the estimate of Server Core needing only one-third the patches required by a full Windows installation is accurate, Server Core will be much easier to manage. Resource usage is lower, with around two-thirds of the normal number of services both installed and running.

So far the top uses for Server Core are expected to be domain controllers (and Server Core also supports Read Only Domain Controllers [RODC]) making it ideal for branch office locations, IIS Web servers, and file/print servers.

The installation process for Server Core is the same as for a full installation, and you can use the same automated installation methods as for a full installation. To configure and manage a core installation, use a slightly different process than you use for a full installation, even though many of the methods you use for Server Core can be used on a full server installation. The command-line options in Server Core give you the ability to perform nearly all functions, and where they fall short you can remotely manage your server. But remember that Server Core is still a Windows installation. All the normal command-line tools and scripting capabilities are available, so just because this chapter didn't cover it does not mean it's not an option. For example, the `tasklist` command is great for seeing the processes running on the system, and `taskkill` is great for stopping them!

If you want a lower overhead Windows installation, Server Core might be the answer.

INDEX

B

C

D

E

F

G

H

I

M

N

O

P

T

U

V

W

X - Y - Z