Chapter 7

Microsoft Windows Server 2008: Server Core

Solutions in this chapter:

- **Server Core Features**
- **Server Core Components**
- **Server Core Best Practices**
- **Server Core Administration**

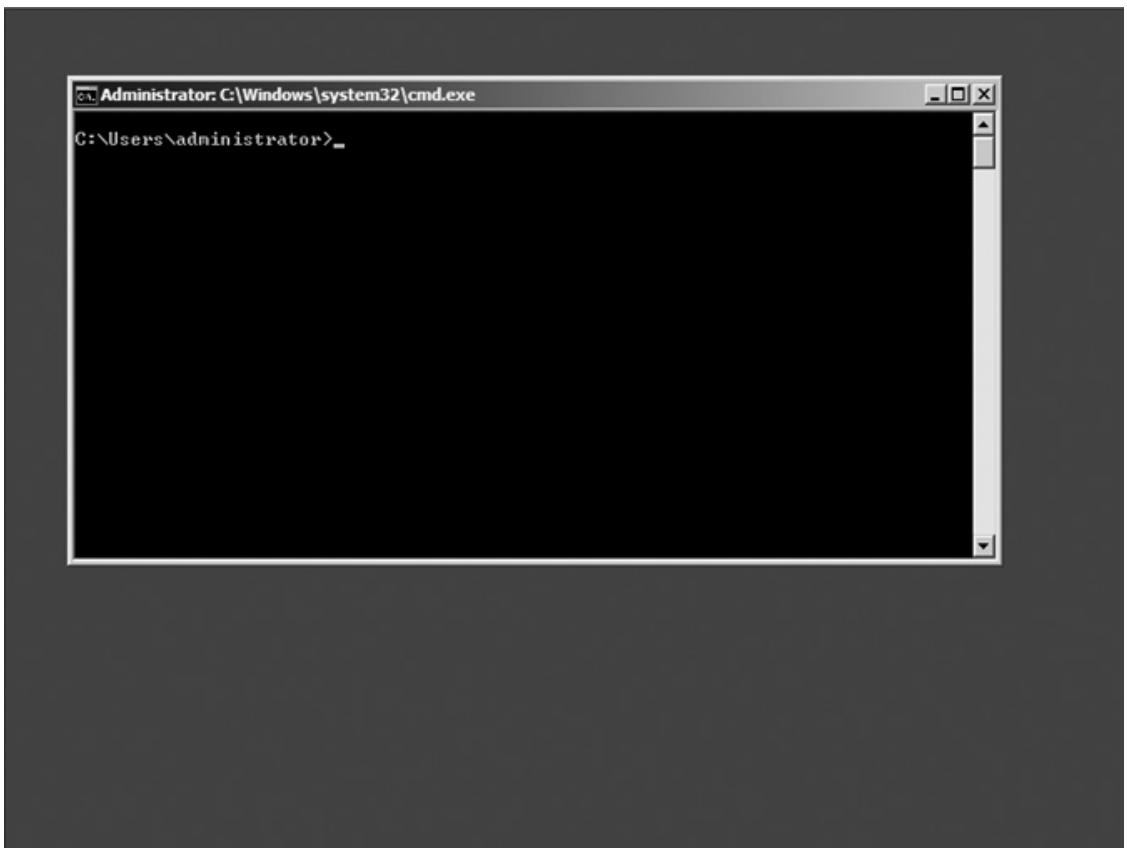
- ☑ **Summary**
- ☑ **Solutions Fast Track**
- ☑ **Frequently Asked Questions**

Introduction

What is Server Core, you ask? It's the "just the facts, ma'am" version of Windows 2008. Microsoft defines Server Core as "a minimal server installation option for Windows Server 2008 that contains a subset of executable files, DLLs and services, and nine server roles." Server Core provides only the binaries needed to support the roles and the base operating system. By default, fewer processes are generally running.

Server Core is so drastically different from what we have come to know with Windows Server NT, Windows Server 2000, or even Windows Server 2003 over the past decade-plus that it looks more like MS-DOS than anything else (see Figure 7.1). Within Server Core, you won't find Windows Explorer, Internet Explorer, a Start menu, or even a clock!

Figure 7.1 The Server Core Console



Becoming familiar with Server Core will take some time. In fact, most administrators will likely need a cheat sheet for a while. To help with it all, you can find some very useful tools on Microsoft TechNet at http://technet2.microsoft.com/windowsserver2008/en/library/47a23a74-e13c-46de-8d30-ad0afb1eaffc1033.mspx?mfr=true. This link provides command and syntax lists that can be used on Server Core. The good news is, for those of you who want the security and features of Server Core with the ease-of-use of a graphical user interface (GUI), you have the ability to manage a Server Core installation using remote administration tools.

Server Core Features

For years, Microsoft engineers have been told that Windows would never stand up to Linux in terms of security simply because it was too darn "heavy" (too much) code, loaded too many modules (services, startup applications, and so on), and was generally too GUI-heavy. With Windows Server 2008, Microsoft engineers can stand tall, thanks to the introduction of Server Core.

The concept behind the design of Server Core is to truly provide a minimal server installation. The belief is that rather than installing all the application, components, services, and features by default, it is up to the implementer to determine what will be turned on or off.

The installation of Windows 2008 Server Core is fairly simple. During the installation process, you have the option of performing a standard installation or a Server Core installation. Once you have selected the hard drive configuration, license key activation, and end-user license agreement (EULA), you simply let the automatic installation continue to take place. When installation is done and the system has rebooted, you will be prompted with the traditional Windows challenge/response screen, and the Server Core console will appear.

When you install Windows Server 2008 without the extra overhead, it limits the number of roles and features that can be used by your server. So why should you install a Server Core in your organization? For the following benefits:

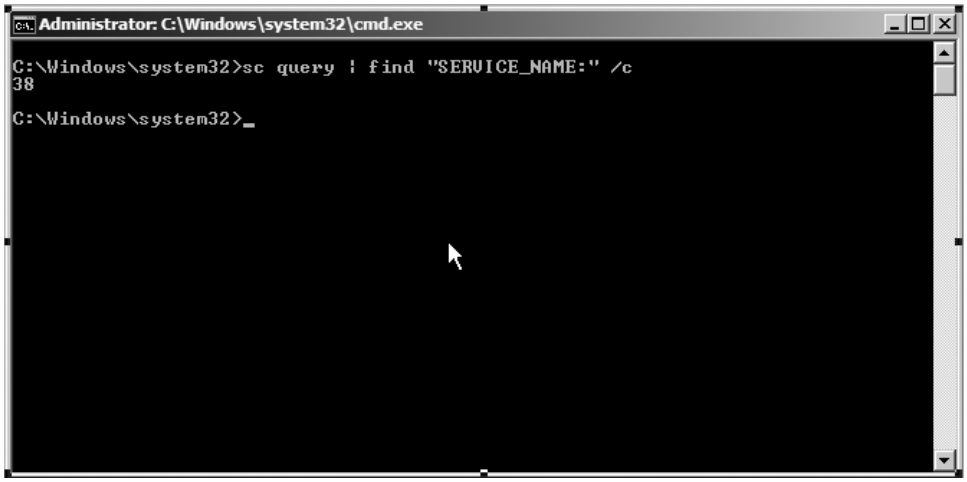
- Minimal attack vector opportunities
- Requires less software maintenance
- Uses less disk space for installation

Server Core Has Minimal Attack Vector Opportunities

Server Core is a bare installation of Windows Server 2008. A machine provisioned with Server Core has fewer binaries installed, which as a result have a reduced attack interface. With less binary available on the system, the change of vulnerable DLLs is decreased. This will force an attacker to expend more effort in finding a security flaw in one of the Windows DLLs.

If we look at the number of services installed by default on a Server Core machine and those on a regular Windows 2008 installation, we see a big difference. I did a comparison on two of my test machines. I executed the command *sc query | find "SERVICE_NAME:" /c* (see Figure 7.2) on the Server Core machine, as well as on my normal Windows 2008 machine. This command counts the number of services installed on a machine. The Server Core had 38 services installed, while the normal Windows 2008 installation had 49 services installed. I did this quick check directly after the initial installation. This means no additional roles were installed. Fewer services installed and running means greater security. By the way, this check was performed between Windows Server 2008 Enterprise (Server Core Installation) and Windows Server 2008 Enterprise (Full Installation).

Figure 7.2 Counting the Number of Services on a Server Core Machine



Microsoft has also removed the most insecure programs like Internet Explorer, Windows Media Player, Windows Messenger, Outlook Express, and so on. Much deeper underlying dependencies are also removed—for example, .NET Framework.

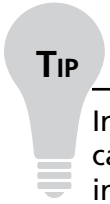
Because .NET Framework is missing, there is no PowerShell, Servermanagercmd.exe, or ASP.Net either. But Microsoft is working on a slimmed-down version of the .NET Framework for a future release. Because fewer applications and services are installed, we can say that the attack surface is far smaller than a regular Windows Server product.

Server Core Requires Less Software Maintenance

We all know the term *Patch Tuesday*, and we all know that some admins don't like rebooting their most important servers to take care of the companies' core business on those occasions. Well, for those admins, there is hope. Microsoft believes that because of the slim version of Server Core, the number of patches required for this Windows version will be reduced by 60 percent, compared with a regular Windows Server 2008 machine. This will dramatically decrease the patch management cycle.

The Server Core machine only does what it has to do. Why should you install a combined DHCP/DNS server on a full Windows installation (which has a lot of additional services and components installed that aren't be used) just to fulfill these two roles? You can better install it on Server Core because this only provides the key network infrastructure roles without all the superfluous DLLs and services.

Last year, I attended a live demo session from Marcus Murray at Tech-Ed Orlando. The session title was *Why I Can Hack Your Network in a Day*. According to Marcus, 95 percent of the software running within a company isn't properly patched, and most of the security flaws are caused by this un-patched software. If we look at Server Core, not much software will be installed besides the antivirus and backup software. And maybe you want to keep it this way. Server Core isn't designed to serve as an application platform.



TIP

In the later section "Server Core – Administration," you will learn how you can enable Windows Update for Server Core. If you want to see a list of installed patches, type the following command:

wmic qfe list

If you want to manually install a patch, use the following command:

Wusa.exe <patchname>.msu /quiet

For more information, type **wusa.exe /?** at the command prompt.

Server Core Uses Less Disk Space for Installation

A Server Core installation requires about 1 gigabyte (GB) of disk space, and for paging, another 512 (MB) is needed. In total, approximately 2GB is required for operations. If we take a look at the official installation requirements, Microsoft minimum installation needs 10GB of disk space; however; 40GB or greater is recommended. Servers with more than 16GB of internal memory require more disk space for paging and dump files.

The advantage of the reduced disk footprint expresses itself in quicker unattended installs and faster booting. Disk costs aren't that expensive anymore. Nevertheless the reduced disk footprint can be a big pro for large datacenters. Imagine having a big datacenter with hundreds of Web-farm front-end servers installed on Server Core, and you are responsible of provisioning them. Imaging these servers with a small cloned image or an unattended installation would be a piece of cake.

Server Core Components

Microsoft has built a new type of operating system (OS) with fewer capabilities, which means less code, so the change of an exploit should be minimal. But what are the consequences of stripping an OS so drastically? One thing's for sure. Server Core can't easily be used as an application server. The strength of Server Core is to fulfill the key functions of a Windows infrastructure. Think about DHCP, DNS, Active Directory Domain Services, and so on. We don't have balloon notifications, but who will miss them. Wait a minute, though… isn't a password expiration a balloon notification? Ok, Microsoft missed that one. Also vendors of antivirus, backup, or other agents have some work to do. Agents installed on Server Core cannot have shell or GUI dependencies and may not require managed code. Many of you may be wondering which components are actually there and which components are missing. The following paragraphs will provide you with the answers to these two questions.

What Is There?

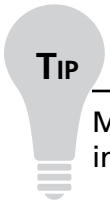
First of all, we have the command prompt. Many DOS commands we know from the past still work for Server Core. So do *fc*, *label*, *defrag*, *ftp*, *diskcomp*, and so on still work? Yes. A good start is the A–Z command line reference. See the link: http://technet2.microsoft.com/windowsserver/en/library/552ed70a-208d-48c4-8da8-2e27b530eac71033.mspx?mfr=true. There is a little GUI support, but most of

the installing and configuring have to be done using the command line. This means that deploying and managing Server Core installations requires a little more knowledge than a normal installation does. This has an advantage in that system administrators have to think about their design and not just click around till it works. However, not all configurations can be done with the standard DOS commands, and this is where *scregedit.wsf* comes in.

scregedit.wsf is a script that helps you edit the Registry easily without opening it. With *scregedit.wsf*, you can enable remote desktop for administration, enable automatic updates, configure the pagefile, enable error reporting, and so on. For more info on how to use this command, see the later section "Server Core – Administration."

Of course, we also have the kernel and the Hardware Abstraction Layer (HAL). But the number of device drivers is limited. Fortunately, you have the opportunity to install additional drivers after the initial installation.

The core subsystem takes care of the security (logon scenarios), file systems, RPC, winlogon, networking, and so on. Some infrastructure features are also crucial to let Windows Core work properly. These features include (as mentioned before) the command shell, but also: domain join, event log, performance counters, http support, and the WMI infrastructure.



TIP

Much of the device drivers in Server Core aren't installed. To get a list of all installed drivers, type:

Sc query type= driver

If you want to determine the version of a particular file—for example, the hal.DLL—type the following command:

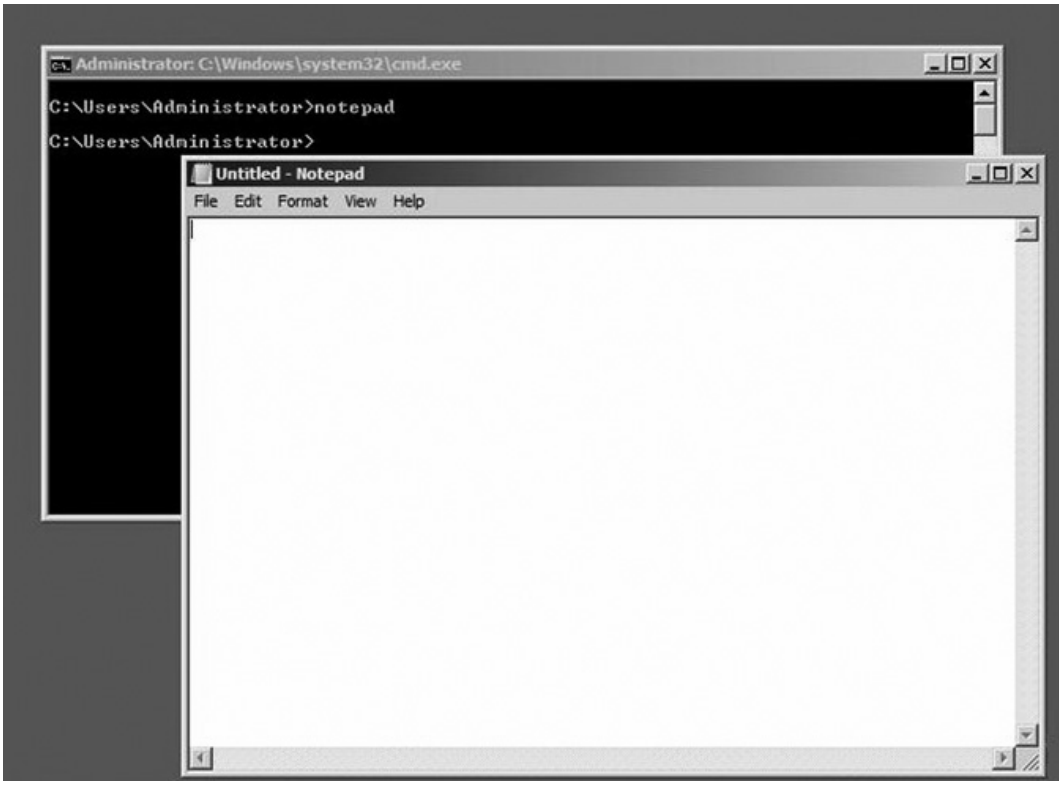
wmic datafile where name="c:\\windows\\system32\\hal.dll" get version

If you want to install a driver that is not available on the Server Core machine, you must copy the driver to a location on the Server Core machine. If you're finished copying, you can execute the following command:

Pnputil –i –a <path>\<driver>.inf

The Server Core architecture is shell-less, but not completely GUI-less. Two Control Panel applets are available. The *regional and language options* applet and the *date and time* applet. Many Technology Adoption Program (TAP) customers have been complaining about the lack of a text editor in the early developing days of Server Core. The only way to open log files, scripts, and so on was remotely. That's why, since the release of the Beta 2 version of Server Core, Notepad is added (see Figure 7.3).

Figure 7.3 Notepad on a Server Core Machine



Another GUI-tool is regedit. Because not all modifications on the system can be made by the command prompt, it's sometimes necessary to edit the Registry. In other words, if you want to make an advanced modification, *regedit* rules.

Besides *regedit*, you can also use the key combination **Ctrl + Shift + Esc**. After pressing these keys, the task manager will appear, letting you see which applications, processes, and services are running. Using task manager, you can also take a quick look at the current load of the server and/or see what the network throughputs are.

Last but not least is the logon screen. These GUI applications are installed by default on Server Core. Firstly, you can add some extra tools or applications like *Mark Russinovich's Sysinternals* for troubleshooting purposes. If you want to install an application on Server Core, read the installation instructions carefully to be sure the application is supported. Many applications you may want to install won't detail possible complications as long they don't depend on DLLs (which aren't available on Server Core).

Server Core also includes the Windows Management Instrumentation (WMI). This is a standardized infrastructure environment that makes it easy to manage servers running Windows operating systems. The command-line version of WMI is the Windows Management Instrumentation Command (WMIC) line. WMIC isn't new, but it can come in handy when administering Server Core. WMIC is more intuitive than WMI because it uses aliases. Later on, we'll see how WMIC can be used to change the computer name or change the pagefile of a Windows Core operating system.

Other good tools are available to remotely administer Server Core. As in the past, we can connect to a server Core Machine with remote desktop. If we have installed the Remote Server Administration Tools (RSAT) on Windows Vista SP1, we can administer Server Core from there. It's obvious we can do our management tasks from a GUI-based Windows 2008 server with Microsoft Management Console (MMC). If we want to connect remotely using remote command capabilities, we can use WS-Management—better known as WINRM/WINRS (Windows Remote Management/Windows Remote Shell). These tools provide more security than others because they operate as a Web services-based mechanism that runs on port 80 or 443. If you must traverse firewalls, WINRM/WINRS lets you open just a single port and even run it encrypted because it supports Secure Sockets Layer (SSL). WINRM/WINRS is not new. It was first introduced with the R2 version of Windows Server 2003. It's also possible to configure this tool in a manner in which remote command capabilities are only possible from a particular subnet. In this way, we can configure WINRM/WINRS to allow only administrators' management workstations to connect to the Server Core machines.

WARNING

If you want to configure WINRM/WINRS to support SSL communication, you must require a valid certificate with server authentication capability and a matching CN. Self-signed certificates don't work.

Which Roles Can Be Installed?

Administrators think of servers in terms of roles. That's our fileserver, that's the DNS server, and so on. A server always fulfills a particular role. For this reason, Microsoft has changed its approach for installing software. A server role provides the key functionality of a particular server. Add/Remove Programs doesn't exist

anymore and has been replaced by Server Manager. If you want to add a role or feature, the Server Manager is the place to do it. But Server Manager doesn't work in Server Core because it uses managed code. So should we use the command-line version *servermanagercmd.exe*? No. Servermanagercmd.exe uses .NET Framework, which isn't modular enough to break it down and let it fit within Server Core. So we use a replacement command called *ocsetup*. With the command *ocsetup*, you can install roles and features on Server Core. One Server Core role you can't install with the *ocsetup* is Active Directory Domain Services (AD DS). In the later section, "Server Core – Administration," you will learn how to use the *ocsetup* and how to install AD DS on Server Core.

The following roles can be installed on Server Core:

- Active Directory Domain Services
- Active Directory Lightweight Directory Services (AD LDS)
- DHCP Server
- DNS Server
- File Services
- Print Server
- Web Server (IIS) (ASP.net is not available for Server Core)
- Hyper-V (This role is not available by default)
- Streaming Media Services (This role is not available by default)

A feature is a functionality that is designed to support the main server roles. The features that Server Core supports are outlined next.

The following features can be installed on Server Core:

- Failover Clustering
- Network Load Balancing
- Subsystem for Unix-based applications
- Backup
- MultipathIo
- Removable Storage Management
- BitLocker Drive Encryption

NOTE

BitLocker Drive Encryption is an integral new security feature in Windows Server 2008 that protects servers at locations such as branch offices, as well as mobile computers (for all those roaming users out there). BitLocker provides offline data and operating system protection by ensuring that data stored on the computer is not revealed if the machine is tampered with when the installed operating system is offline.

To see which roles and features are installed on Server Core, use the command *oclist.exe*. The result of executing the *oclist* command on a Server Core machine is shown next. The following output is modified. The subcomponents from the roles and features are not displayed because it generates six pages of output.

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:IIS-WebServerRole
Not Installed:Microsoft-Windows-RemovableStorageManagementCore
Not Installed:MultipathIo
Not Installed:NetworkLoadBalancingHeadlessServer
```

Not Installed:Printing-ServerCore-Role
Not Installed:QWAVE
Not Installed:ServerForNFS-Base
Not Installed:SNMP-SC
Not Installed:SUACore
Not Installed:TelnetClient
Not Installed:WAS-WindowsActivationService
Not Installed:WindowsServerBackup
Not Installed:WINS-SC

If you want to install a role, you can type ***start /w ocsetup DNS-Server-Core-Role*** to install DNS. If you want to install DHCP, use the command ***start /w ocsetup DHCPServerCore***. Remember, the commands are case-sensitive. Make sure to supply the correct role or feature name after the command *start /w ocsetup* because the role names are not consistently used. Some roles are written with a hyphen, while others aren't. If you want to de-install a role, first stop the role service with ***NET STOP***, and then type ***start /w ocsetup serverrolename /uninstall*** to uninstall the filled-in role.

What Is Missing?

In my opinion, the biggest disadvantages of Server Core are that PowerShell and ASP.net on IIS don't work on Server Core because they both depend on .NET Framework. Ok, that's said. If it makes you happy, PHP is supported. But let's go further. We all know Vista and that WOW feeling—and one thing's for sure, that WOW feeling is definitely missing. We don't have glassy windows and flip 3D; we don't even have a Desktop, Start Menu, Control Panel, Explorer, Windows Media Player, and the rest. But the big question is, do we need all this on a server? If we take the roles and features that can be installed on Server Core and compare this with the roles and features that can be installed on a regular Windows Server 2008 machine, we get the list shown in Table 7.1.

Table 7.1 Available and Unavailable Roles and Features on Server Core

Roles	Availablility
Active Directory Certificate Services	No
Active Directory Domain Services	Yes
Active Directory Federation Services	No
Active Directory Lightweight Directory Services (AD LDS)	Yes
Active Directory Rights Management Services (AD RMS)	No
Application Server	No
DHCP Server	Yes
DNS Server	Yes
Fax Server	No
File Services	Yes
Print Server	Yes
Streaming Media Server	Yes
Hyper-V	Yes
Network Policy and Access Services	No
Terminal Services	No
Universal Description, Discovery, and Integration Services	No
Web Server (IIS)	Yes
Windows Deployment Services	No
Windows Sharepoint Services	No
Features	**Availability**
BitLocker	Yes
BitLocker Remote Administration Tool	Yes
BITS Server Extensions	No
Connection Manager Administration Kit	No
Desktop Experience	No
Failover Cluster	Yes
Group Policy Management	No
Internet Printing Client	No

Continued

Table 7.1 Continued. Available and Unavailable Roles and Features on Server Core

Roles	Availablility
Internet Storage Name Server	No
LPR Port Monitor	No
Message Queuing	No
Microsoft .NET Framework 3.0 Features	No
MultipathIo	Yes
NLB	Yes
Peer Name Resolution Protocol	No
Quality Windows Audio Video Experience	Yes
Remote Assistance	No
Remote Differential Compression	No
Remote Server Administration Tools	No
Removable Storage Management	Yes
RPC over HTTP Proxy	No
Services for NFS	No
Simple TCP/IP Services	No
SMTP Server	No
SNMP Services	
Storage Manager for Storage Area Networks	No
Subsystem for Unix-based applications	Yes
Telnet Client	
Telnet Server	No
Trivial File Transfer Protocol Client	No
WAS (WindowsActivationService)	Yes
Windows Internal Database	No
Windows Internet Name Service (WINS)	
Windows PowerShell	No
Windows Process Activation Service	No
Windows Server Backup Features	
Windows System Resource Manager	No
Wireless LAN Service	No

Server Core Best Practices

If you work as a field engineer and must install Server Core at various customer locations, wouldn't it be nice to have some kind of manual that summarizes some of the best practices? Some documentation exists in books and on the Internet, but the neater tricks are hard to find, or can't be found at all. Working as a consultant, I collected some of these tricks and bunched them together. Some made me think, "Hey, why didn't I think of that?" Other tricks (I think) are pretty cool, like "*enabling remote cmd.exe with terminal services*." In the paragraphs that follow, you'll find some practical tips that will come in handy when implementing Server Core.

Installing Software

Just to be sure… you do have backup clients and antivirus engines running on your servers, don't you? Thankfully, it's possible to install antivirus software like Microsoft's ForeFront and backup agents such as Symantec Backup Exec 12 on Windows 2008 Server Core. But how do you arrange this if you don't have *Add/Remove Programs* or even a GUI? Well, you still have msiexec.exe and the normal executable files. If you want to install an application with *msiexec*, just type ***msiexec /i* productname *.msi/***. See Table 7.2 for some of the most oft-used command-line switches for *msiexec*. If you want to see the full list, use the link: http://support.microsoft.com/kb/227091. You may get the feeling that without a GUI nothing can happen with your Server Core installation. With fewer DLLs, the attack surface may be reduced, but it's still advisable to install antivirus and backup agents on the machine. Maybe it's better to say that Server Core is shell-less and a little bit GUI-less. If you want, you can still install lots of software, as long as the software doesn't need DLLs (which are aren't available on Server Core). It's even possible to install a browser like Mozilla Firefox on Server Core. But it's strongly recommended you only install supported software on Server Core.

Table 7.2 *msiexec* Command-Line Parameters

Switch	Parameters	Description
/i	Package|ProductCode	Installs or configures a product
/f	[p|o|e|d|c|a|u|m|s|v] Package|ProductCode	Repairs a product p - Reinstalls a product only if a file is missing o - Reinstalls a product if a file is missing or if an older version of a file is installed e - Reinstalls a product if a file is missing or an equal or older version of a file is installed d - Reinstalls a product if a file is missing or a different version of a file is installed c - Reinstalls a product if a file is missing or the stored checksum value does not match the calculated value a - Forces all files to be reinstalled u - Rewrites all required user-specific Registry entries m - Rewrites all required computer-specific Registry entries s - Overwrites all existing shortcuts v - Runs from the source file and re-caches the local package
/a	Package	Administrative installation option; installs a product on the network
/x	Package|ProductCode	Uninstalls a product

Changing Background Settings and More

Imagine you are a system administrator and working in a server park with approximately 200 Core Servers. Ten of them are very important because these are installed with IIS and take care of the companies' core business. You surely don't want to mess up these servers. So you are looking for a manner to distinguish these servers from the others.

Well let's use the old fashioned way. We can change the background color to (for instance) red. Type ***regedit*** in the console, browse to the key ***HKEY_CURRENT_USER\Control Panel\Colors\Background***, and change the value to **255 0 0**. Don't forget to log off and log on again so your Registry changes are applied. The default background is now changed to red.

If you want to disable the screensaver, again type ***regedit*** at the command prompt and go to **HKEY_CURRENT_USER\Control Panel\Desktop\ScreenSaveActive**. Then, change the value from **1** to **0**. But maybe you want to do the opposite and add a screensaver with a warning text that says *Don't touch my Web server!* The Web servers are still your companies' core business, right? The screensaver we're taking about is called Marquee and the screensaver file is not available on Server Core by default, so we have to copy it. Locate the file *ssmarque.scr* (c:\windows\system32\) on an XP machine and copy it to the same location on a Server Core machine. On the Core Machine, open the Registry with ***regedit.exe*** and browse to **HKEY_CURRENT_USER\Control Panel\Desktop**. Change the value ***SCRNSAVE.EXE*** to ***C:\WINDOWS\system32\ssmarque.scr*** and you're almost done. If you want to change the default screensaver timeout of ten minutes, change the value ***ScreenSaveTimeOut*** from *600* seconds to a value better suited to your needs. The last thing we must do is change the default text from the screensaver. To arrange this, type the command ***c:\windows\system32\ssmarque.scr*** in the command prompt and change the text in the box. (See Figure 7.4.)

Figure 7.4 Changing the Screensaver in Server Core

Tools & Traps…

Changing Your Display Resolution

You can increase or decrease your display resolution by editing the Registry. First, make a Registry export using the following key: *HKLM\SYSTEM\CurrentControlSet\Contro\Video*. Now you can safely edit the Registry because you have a backup. Browse to the key *HKLM\SYSTEM\CurrentControlSet\Contro\Video*. Beneath this key are a couple of GUIDS. You have to "play" with it a little to find out which of the GUIDS belongs to your video card. Under the key *video\GUID\0000* should be two *Dword* values: *defaultsettings.xresolution* and *defaultsettings.yresolution*. After editing these settings so they correspond with your display resolution, don't forget to reboot. Remember, modifying the Registry can be dangerous.

If you want to set your display settings during an unattended install, use the following tags:

```
<Display>
  <HorizontalResolution>1024</HorizontalResolution>
  <VerticalResolution>768</VerticalResolution>
  <ColorDepth>16</ColorDepth>
</Display>
```

The following link displays a complete unattended sample file: http://technet2.microsoft.com/windowsserver2008/en/library/47a23a74-e13c-46de-8d30-ad0afb1eaffc1033.mspx?mfr=true.

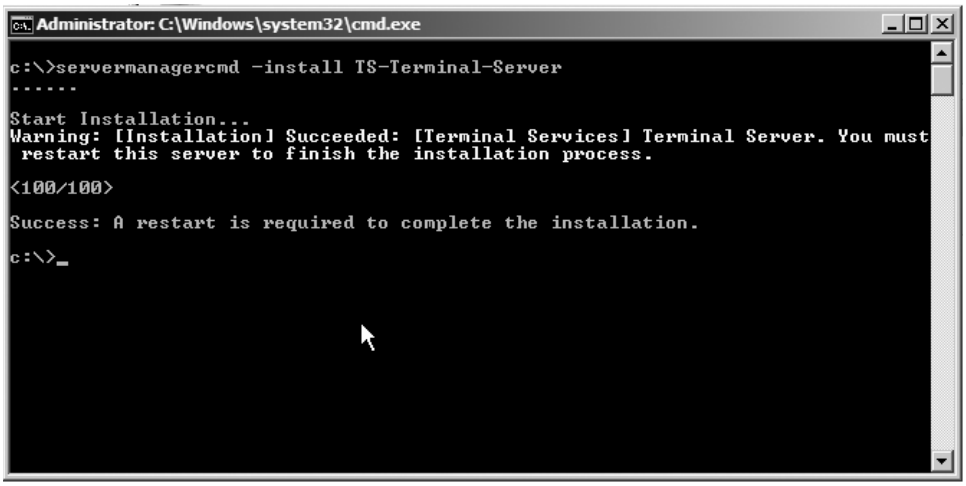
Enabling *remote cmd.exe* with Terminal Services

Imagine you are still working on that company that has approximately 200 Core Servers, and you are looking for a way to remotely administer them. You are in possession of one GUI-based Server 2008 machine. The following steps should be performed to get remote cmd.exe working as a Terminal Services Remote Program. This "cool" function is similar to administering Server Core with *mstsc.exe /v server name*. The only difference is that you don't use the full-sized remote desktop functionality anymore, only a "published" remote application. The protocol used is still RDP.

1. Enable Remote Desktop on the Server Core computer by typing the command prompt ***cscript c:\windows\system32\scregedit.wsf /ar 0***.

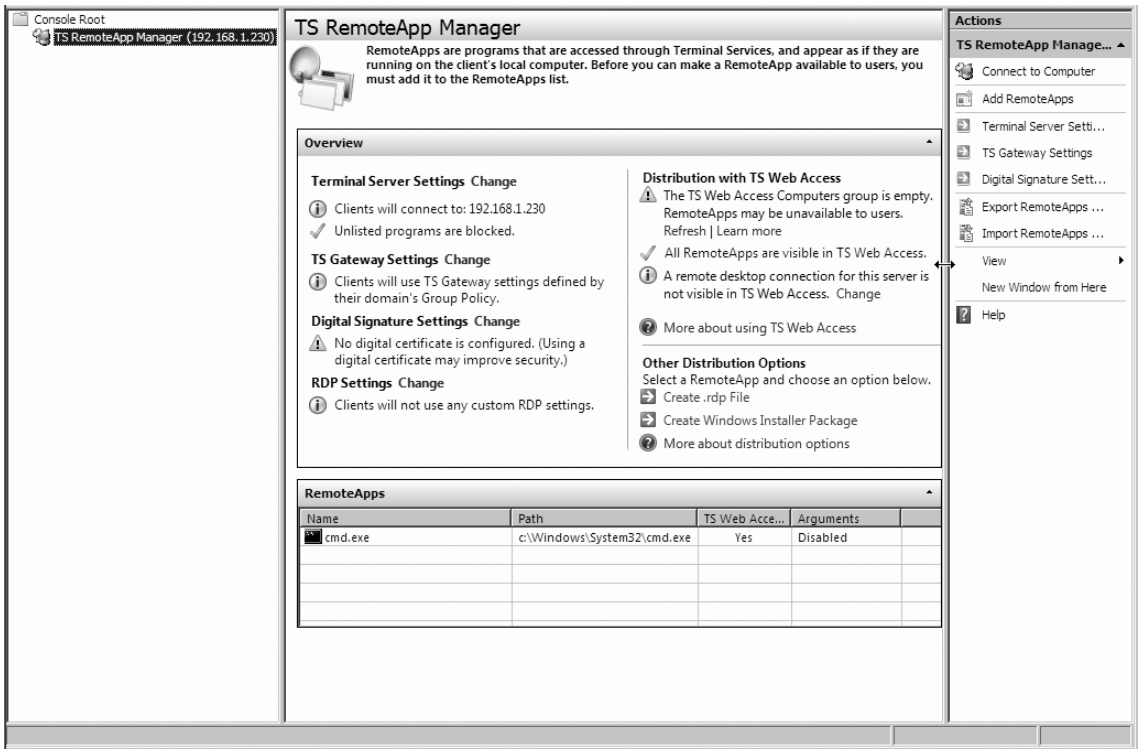
2. Install the role *Terminal Server* on the GUI Server 2008 machine with Server Manager or by typing ***servermanagercmd.exe –install TS-Terminal-Server*** at the command prompt. Don't forget to reboot after installation. (See Figure 7.5.)

Figure 7.5 Installing Terminal Services on a Full Windows 2008 Installation



3. After the reboot, open the MMC **TS RemoteAPP Manager** you just installed. Remember, you must open TS RemoteAPP Manager with MMC because Server Manager doesn't let you make connections with other servers.
4. Instead of a local computer, select the IP address or hostname of the Core Server. (See Figure 7.6.)
5. Click **Add RemoteApps** in the upper right corner, and then click **Next**.
6. Click **Browse** and type ***servercorename*\c$\system32\cmd.exe**, and then click **Open | Next | Finish**.
7. cmd.exe will be added to the list of remote programs.
8. In the **RemoteApps** pane, you should see the application you just created. Right-click the application and select **create .rdp File**.
9. Save the RDP file to the location of your choice.
10. If you open the RDP file, a remote command session will start to the Server Core machine.

Figure 7.6 Remotely Connected to a Server Core Machine with TS RemoteApp Manager

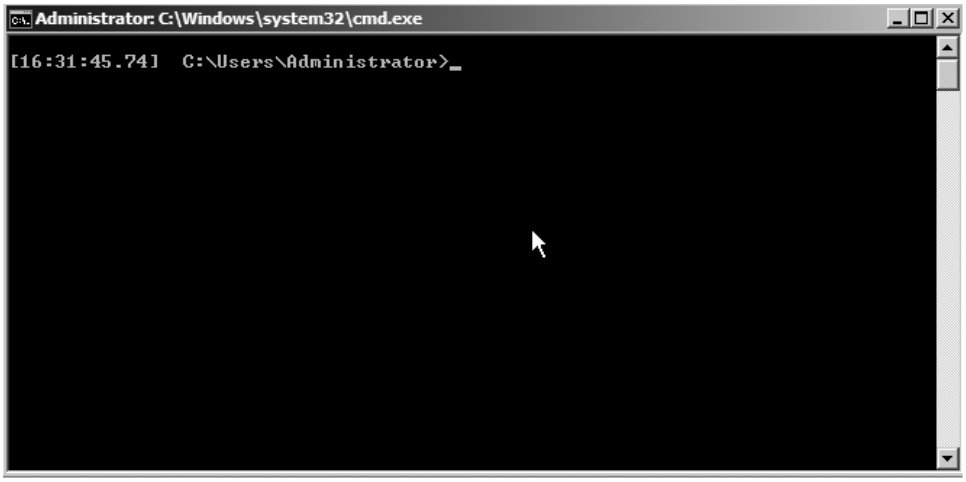


If you walked through the preceding steps, you just administered Server Core with *remoteAPPS*. Why should you open the full screen RDP version with *mstsc* if you only need the command prompt window? This is an excellent way to administer your Server Core machines. After opening the RDP file, you'll see a normal DOS box like any other DOS box on your local machine. Keep in mind, however, that every command you're executing will be executed on the remote machine, not the local. Especially when connecting to multiple remote Server Core machines, you probably don't remember anymore which DOS box is for which server. To help with this, the next paragraph offers a solution to this dilemma.

Changing the Command Prompt

Maybe you didn't notice it, but there is no time indication on a Server Core machine. After you execute the command ***prompt [$t]$ssp$g***, your command prompt will look like that shown in Figure 7.7, letting you know exactly when it's lunchtime.

Figure 7.7 Changing the Look of the Command Prompt



If we analyze the command, we see that *[$t]* is the variable that shows the time, *$s* is a space, *$p* shows the current drive and path, and *$g* shows the greater than sign >. If you perform remote administration on more than one machine, it's a best practice to change the default prompt in such a way as to distinguish the command prompts from each other. The variable *%computername%* can help you with this. After executing the command ***prompt [$t]$s[%computername%]sp$g***, your prompt will look like the following:

```
[16:46:42.65] [CORE] C:\Users\Administrator>
```

where *Core* is the computer name of the Server Core machine. Unfortunately, when you log off, your settings are lost. If you wish to save your changes permanently, use the Registry. Type ***Regedit*** at the command prompt. Locate the key **HKLM\System\CurrentControlSet\Control\Session Manager\Environment**. Create an expandable string value with the name **prompt**. The value of the string can be the same as the prompt commands we used earlier—for example, *[$t]$s[%computername%]sp$g*. Type **prompt /?** to see what methods are available to change the command prompt.

Prompt can be made up of normal characters and the following special codes:

$A & (Ampersand)
$B | (Pipe)
$C ((Left parenthesis)
$D Current date
$E Escape code (ASCII code 27)

$F) (Right parenthesis)
$G > (Greater-than sign)
$H Backspace (erases previous character)
$L < (less-than sign)
$N Current drive
$P Current drive and path
$Q = (Equal sign)
$S (Space)
$T Current time
$V Windows version number
$_ Carriage return and linefeed
$$ $ (Dollar sign)

If Command Extensions are enabled, the PROMPT command supports the following additional formatting characters:

$+ zero or more plus sign (+) characters depending upon the depth of the PUSHD directory stack, one character for each level pushed.

$M Displays the remote name associated with the current drive letter or the empty string if current drive is not a network drive.

Administrating Server Core with RDP

Every system administrator knows the command *mstsc /v servername /console*. With this command, we start a Remote Desktop session to another machine. If we want to connect to a Server Core machine, we must first enable Remote Desktop on this particular server. With the GUI versions of Windows, we right-clicked **Computer** to open Properties, afterwards selected the tab **Remote**, and then marked the checkbox **Enable Remote Desktop on this Computer**. In Server Core, we can't do this anymore. But don't worry… we can execute the command *cscript c:\windows\system32\scregedit.wsf /ar 0* to enable Remote Desktop. This command will also create an exception rule in Windows Firewall. With the command *cscript c:\windows\system32\scregedit.wsf /ar 1*, we can disable it again. If we want to see the current settings, we use the command *cscript c:\windows\system32\scregedit.wsf /ar /v*. If you have problems connecting to Server Core with Windows XP, execute the command *cscript c:\windows\system32\scregedit.wsf /cs 0*. This disables some enhanced security settings that are implemented by Server 2008 and Vista.

Tools & Traps…

Administering Terminal Server Sessions

They aren't often used by administrators, but certain command-line syntaxes can be employed to administer terminal server sessions. For reference, see: http://technet2.microsoft.com/windowsserver/en/library/1db49727-f587-424d-8d98-bb51630d13a01033.mspx?mfr=true. The following command queries all the processes on a server with IP address 192.168.1.230.

query process * /server:192.168.1.230

USERNAME	SESSIONNAME	ID	PID	IMAGE
(unknown)	services	0	4	system
system	services	0	308	smss.exe
system	services	0	376	csrss.exe
system	services	0	420	wininit.exe
system	services	0	504	services.exe
system	services	0	516	lsass.exe
system	services	0	524	lsm.exe
system	services	0	688	svchost.exe
network service	services	0	740	svchost.exe
local service	services	0	780	svchost.exe
system	services	0	872	svchost.exe
network service	services	0	888	slsvc.exe
local service	services	0	952	svchost.exe
network service	services	0	1048	svchost.exe
local service	services	0	1160	svchost.exe
system	services	0	1308	taskeng.exe
system	services	0	1376	frameworkser...
system	services	0	1500	mcshield.exe
system	services	0	1528	vstskmgr.exe
system	services	0	1556	naprdmgr.exe
network service	services	0	1580	svchost.exe
local service	services	0	1612	svchost.exe
network service	services	0	1652	svchost.exe
system	services	0	1976	svchost.exe
network service	services	0	1664	msdtc.exe
system	console	1	2788	csrss.exe

Continued

system	console	1	2828	winlogon.exe
(unknown)	console	1	1324	taskeng.exe
(unknown)	console	1	1432	taskmgr.exe
(unknown)	console	1	3472	cmd.exe
(unknown)	console	1	2540	rundll32.exe
(unknown)	console	1	2648	notepad.exe

The following command kills the process with Process ID Number (PID) 2648. In this case, it's Notepad.

tskill /server:192.168.1.230 2648

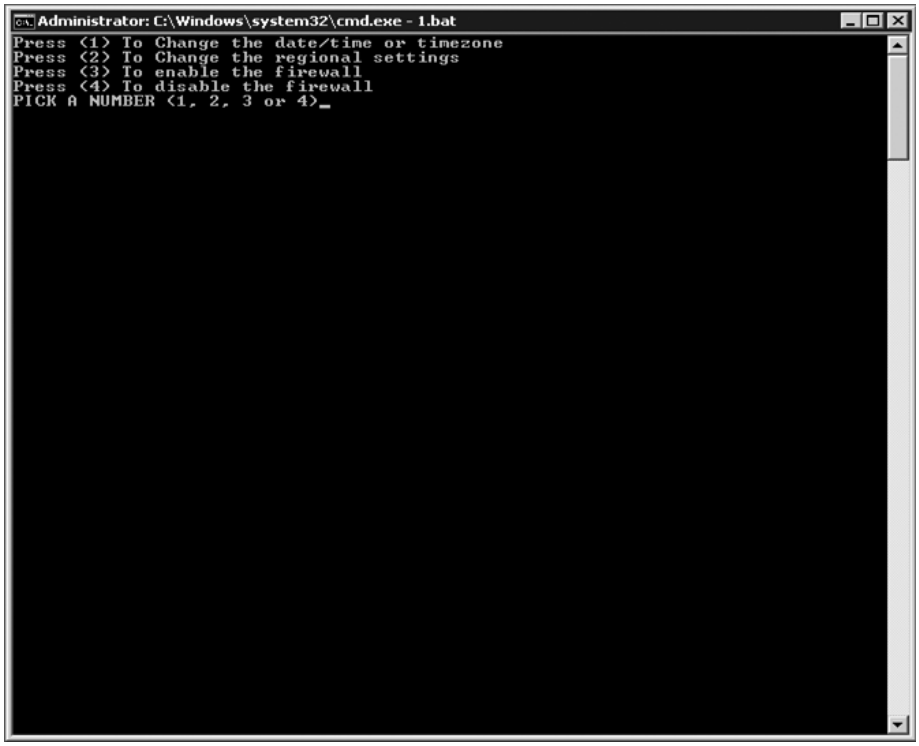
Creating Batch Menus

Maybe you remember the good old *choice.exe* from the Windows NT and Windows 2000 resource kits. Choice.exe allows users to select one item from a list of choices and returns the index of the selected choice. The resource kit tool is so often used that it became a built-in command in Windows 2003. Unfortunately, the Server Core version of Windows 2008 doesn't have a choice.exe or a replacement command. Instead of typing long commands—for example, to disable or enable the Windows firewall—you can use choice.exe to create a batch file menu that represents the shortcuts of the long commands. You can download choice.exe from the following location.ftp://ftp.microsoft.com/Services/TechNet/samples/PS/Win98/Reskit/SCRPTING/CHOICE.EXE. Look at the batch file that follows. This simple batch file gives you an idea how choice.exe can be used. Download choice.exe from the location just mentioned and place it somewhere in the Path—for instance, %systemroot%\system32. Copy and paste the following text in a text file and save it with the extension **.bat**. If you execute the batch file, your screen should show what's displayed in Figure 7.8.

```
@ECHO OFF
REM - Script written by Remco Wisselink
:BEGIN
CLS
```

```
ECHO Press (1) To Change the date/time or timezone
ECHO Press (2) To Change the regional settings
ECHO Press (3) To enable the firewall
ECHO Press (4) To disable the firewall
CHOICE /N /C:1234 PICK A NUMBER (1, 2, 3 or 4)%1
IF ERRORLEVEL ==4 GOTO Four
IF ERRORLEVEL ==3 GOTO THREE
IF ERRORLEVEL ==2 GOTO TWO
IF ERRORLEVEL ==1 GOTO ONE
GOTO END
:Four
ECHO You pressed (4)
netsh firewall set opmode mode=disable
GOTO END
:THREE
ECHO You Pressed (3)
netsh firewall set opmode mode=enable
GOTO END
:TWO
ECHO You Pressed (2)
control intl.cpl
GOTO END
:ONE
ECHO You Pressed (1)
control timedate.cpl.
:END
```

Figure 7.8 What's Displayed in a Batch File



Combining Server Core, Read-Only Domain Controller, and BitLocker

Branch offices are often badly secured. If a branch office's domain controller gets stolen, it's wise to reset all your passwords. Not only must the user account passwords be reset but also the passwords from administrative and service accounts. Why? Because the passwords are locally cached on the domain controller. You don't have to be a rocket scientist to crack all the domain passwords with password cracking tools like lovecrack. Windows 2008 has a new infrastructure solution called Read-Only Domain Controller (RODC). The advantage of an RODC is that you can define which passwords should be cached on the server. For this reason, the RODC is a perfect solution for Branch Offices. If you implement this solution, it's a good thing to only replicate and cache the passwords from normal user accounts that have low level privileges. And it's obvious that you replicate and cache only passwords from accounts that actually reside at the branch office. In the event of a stolen RODC, you only have to reset the accounts whose passwords are cached on the domain controller.

Another security measure we can take is to encrypt the disk with Windows Server 2008's BitLocker. BitLocker is a security feature that protects the operating systems disks by doing a full drive encryption. If you have to design a solution for branch offices, think about the perfect combination: Server Core, RODC, and BitLocker.

Server Core Administration

If we want to describe the ways to administer Server Core, we can divide this into three sections. The first section is called *Installing Server Core*. After the installation part, we do some initial configuration, like setting IP addresses and joining the domain. This section is called *Configuring Server Core.* Personally, I think this part is the most fun because of all the command-line stuff. After the configuration part, we can reach the Server Core machine with remote tools like MMC, Remote Administration Tools (RSAT), or WS-Management (Web-Services Management). We will cover these tools in the final section of this chapter, "Administrating Server Core."

Installing Server Core

If you want to install Server Core, you must begin from scratch. You can't upgrade from legacy Windows versions to Server Core, you can't upgrade from a full Windows 2008 installation to Windows Server Core, and you can't upgrade from Windows Server Core to a full Windows 2008 installation either.

The following versions of Server Core are available:

- Standard Version
- Enterprise Version
- Datacenter Version

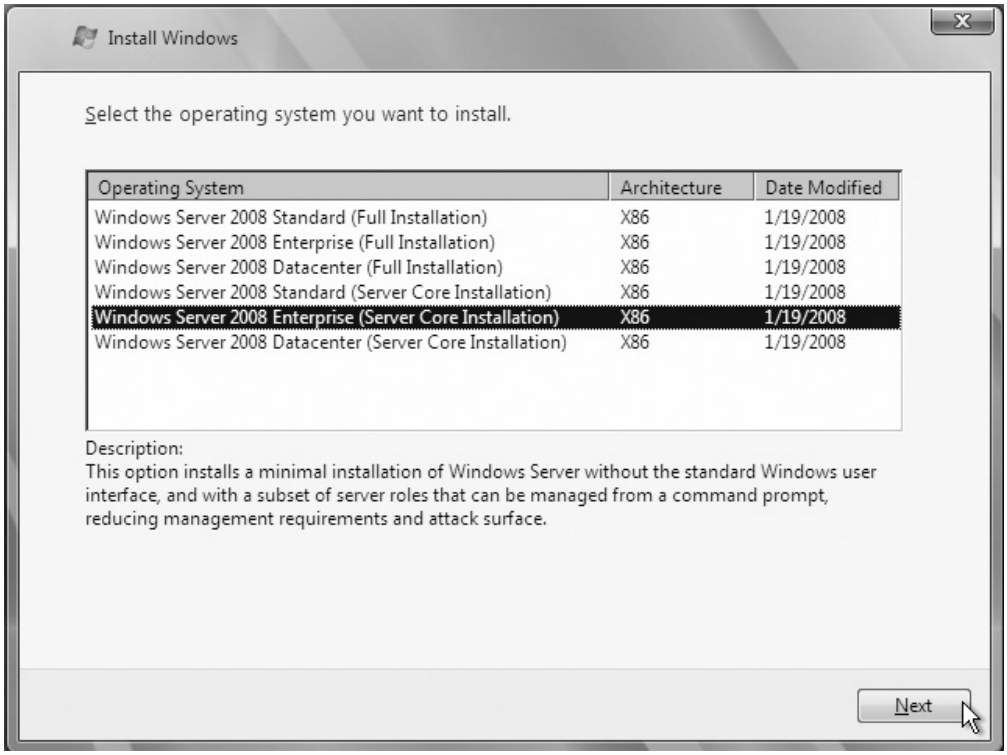
Windows Server Core is available for 32- and 64-bit architecture. The installation of Server Core is not difficult. At the beginning of the installation process, you can choose to install the full server version or the stripped version, which is Server Core. Another way to install Server Core is unattended. An unattended installation is a way of automating the Server Core installation process. By installing Server Core unattended, you can quickly deploy multiple servers and be confident they are identical.

Steps for a Normal Installation

Perform the following steps to install Server Core with a graphical user interface (GUI). (See Figure 7.9.)

1. Insert the Windows Server 2008 installation media into your DVD drive.
2. When the auto-run dialog box appears, click **Install Now**.
3. Follow the instructions on the screen to complete Setup.
4. After Setup completes, press **Ctrl+Alt+Del**, click **Other User**, type **Administrator** with a blank password, and then press **Enter**. You will be prompted to set a password for the Administrator account.

Figure 7.9 Installation Options for Server Core



Steps for an Unattended Installation

If you want to install Server Core without user intervention, perform the following steps.

1. Use a text editor to create an .xml file titled **unattend.xml**.
2. Copy the unattend.xml file to a local drive, USB stick, floppy, or shared network resource.
3. Boot the server to Windows Preinstallation Environment (Windows PE).
4. Insert the Windows Server 2008 DVD into your DVD drive.

5. At a command prompt, change to the drive that contains the Windows Server 2008 DVD.
6. Type the following **setup /unattend:<path>\unattend.xml**.
7. Complete the setup.

Configuring Server Core

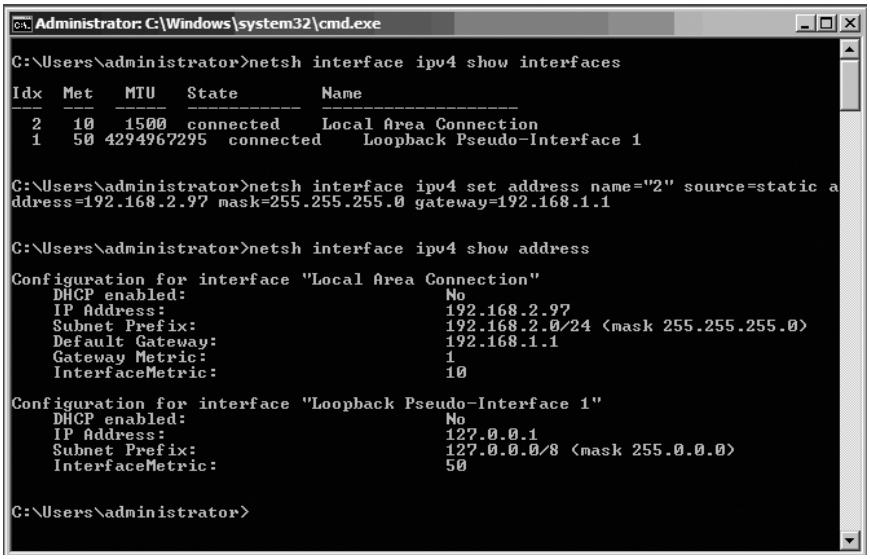
In the following paragraphs, we will learn how we do the initial configuration of a Server Core machine. Because there is no GUI, practically all configuring must be done through the command line. (I hope the Unix and Linux guys like this part.) You can use this chapter as a reference while you're configuring your Server Core machine. When you're finished with the initial configuration, most of the managing stuff can be done remotely with GUI tools.

Configuring the IPV4 IP-Stack

When the installation is done and the system has rebooted, you will be prompted with the traditional Windows command prompt. Since don't have a graphical user interface, how you do configure, for example, IP addresses? Well, the answer isn't that difficult. We can use the command *netsh*. This command isn't new, and was used in previous versions of Windows, like Windows 2000, Windows 2003, and Windows XP. netsh is not very familiar to GUI-admins either. Nevertheless, it is a powerful command-line utility and we must use it if we want to configure the network configuration on a Server Core machine. The following steps show how to configure the IPv4-stack command line.

1. Identify the network adapter. To do this, in the console window type ***netsh interface ipv4 show interfaces*** and record the number shown under the Idx column.
2. Set the IP address, subnet mask, and default gateway for the server. To do so, type ***netsh interface ipv4 set address name=<ID> source=static address=<StaticIP> mask=<SubnetMask> gateway=<DefaultGateway>***. *<ID>* represents the identification of the networking interface for which you want to change the address. The *<ID>* is identified in step 1, *<StaticIP>* represents the IP address we will assign, *<SubnetMask>* represents the subnet mask, and *<Default Gateway>* represents the IP address of the server's default gateway. See Figure 7.10 for our sample configuration.

Figure 7.10 Configuring IP Addressing on Server Core



3. Assign the IP address of the DNS server. From the console, type ***netsh interface ipv4 add dnsserver name=<ID> address=<DNSIP> index=1***. *<ID>* represents the number from step 1, and *<DNSIP>* represents the IP address of the DNS server; *index=1* positions the DNS server as primary DNS server, while index=2 positions the DNS server as an alternate DNS server.
4. Assign the IP address of a WINS server. From the console, type ***netsh interface ipv4 add winsserver name=<ID> address=<WINSIP> index=1***. *<ID>* represents the number from step 1, and *<WINSIP>* represents the IP-address of the WINS server; *index=1* positions the WINS server as primary WINS server, while *index=2* positions the WINS server as an alternate WINS server.
5. Unfortunately, you can't change the DNS suffix using the *netsh* command, so you must use regedit or group policy. If your server will become a member server, then in most circumstances you won't have to configure the DNS suffix because it will automatically get the suffix from the domain to which it belongs. If you want to change the DNS suffix, use the command ***Regedit***, localize **HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters**, and change the values for *Domain (REG_SZ)* and SearchList *(REG_SZ)*. If you are not sure how to do this, try it on a normal GUI-based Windows machine and export/import the Registry value.

6. If you want to change the default value for *NetBIOS over TCP/IP* from Enable to Disable, you must use the command *regedit*. It is not possible to configure *NetBIOS over TCP/IP* settings with the command *netsh* or with group policy. After starting ***regedit*** from the command prompt, localize the key **HKLM\SYSTEM\CurrentControlSet\Services\NetBT\Parameters\Interfaces**. Each network interface card is represented with a subkey *Tcpip_{NETWORKCARDGUID}* below its location. Each subkey has a DWORD value *NetbiosOptions* with a default value of 2. If you change this value to 0, then *NetBIOS over TCP/IP* is disabled.

Configuring Windows Firewall

The Windows Firewall is turned on by default on a Windows Server 2008 machine. You can turn it off with the command *netsh firewall set opmode mode=disable*. However, this should only be done in a test environment, not in a production environment. If you want to enable the Windows Firewall, use the same syntax but substitute *mode=disable* with *mode=enable*. If you install a particular role on a Server Core machine, then the required ports to fulfill the role service will be opened. To enable Remote Administration in Windows Firewall, use the command *netsh advfirewall firewall set rule group="Remote Administration" new enable=yes*. This will enable remote management for any MMC snap-in. In some situations, it may be more appropriate to limit the number of MMCs that can connect. This is where Rule Groups come in. Windows Firewall has some default Rule Groups that correspond to MMCs. If you enable a particular Rule Group, then the corresponding firewall rule will be added to the firewall configuration. Table 7.3 shows the Rule Groups defined within Server 2008.

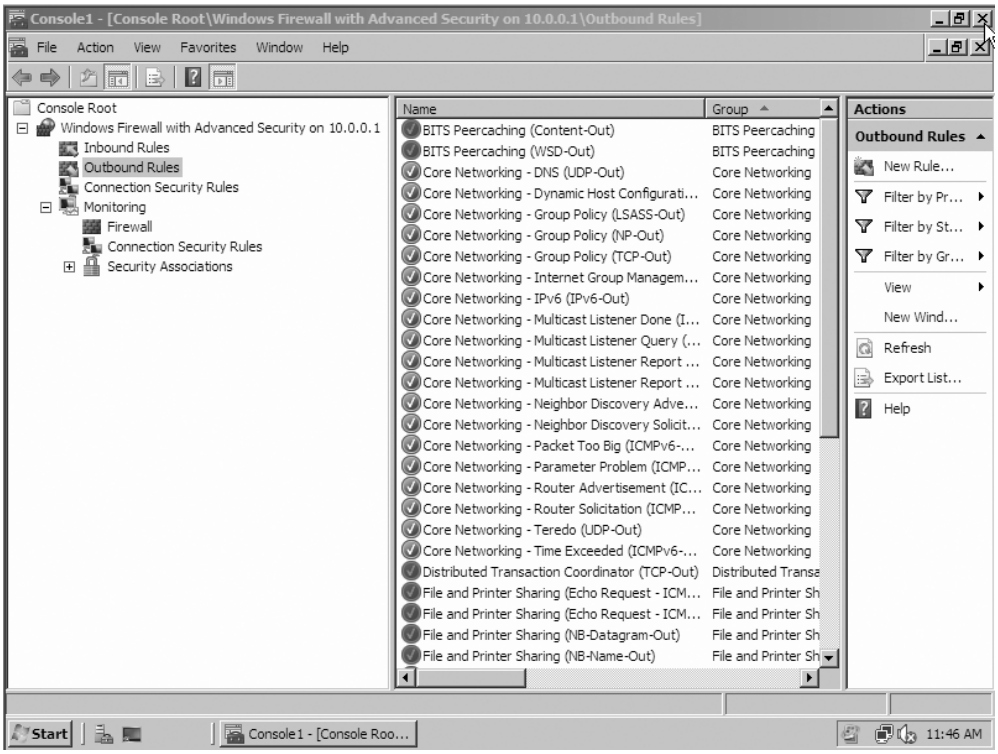
Table 7.3 MMC Snap-ins and the Corresponding Firewall Rule Groups

MMC Snap-in	Rule Group
Event Viewer	Remote Event Log Management
Services	Remote Service Management
Shared Folders	File and Printer Sharing
Task Scheduler	Remote Scheduled Tasks Management
Reliability and Performance	"Performance Logs and Alerts" and "File and Printer Sharing"
Disk Management	Remote Volume Management
Windows Firewall with Advanced Security	Windows Firewall Remote Management

If you want to allow only specific MMC snap-ins to connect, type ***netsh advfirewall firewall set rule group="<rulegroup>" new enable=yes*** at the command prompt. Replace *<rulegroup>* with one of the values mentioned in Table 7.3. If you, for example, want to allow other computers or servers to connect to a Server Core machine with *eventviewer execute*, type in the following command at the command prompt: ***netsh advfirewall firewall set rule group=" Remote Event Log Management" new enable=yes***.

Configuring Windows Firewall through the command line can prove quite complex in some situations. It's much easier to use the Windows Firewall snap-in from a computer running Windows Vista or Windows Server 2008, and then remotely manage the firewall on a server running a Server Core installation. To accomplish this, first execute the command ***netsh advfirewall set currentprofile settings remotemanagement enable***. After executing this command, you're allowed to connect to the Server Core machine with the Windows Firewall MMC. In Figure 7.11, you can see a regular Windows 2008 server connected to a Server Core machine with IP address 10.0.0.1.

Figure 7.11 Configuring Windows Firewall on Server Core from a Regular Windows 2008 Server



Changing the Hostname

After the initial setup, Server Core has a default random configured hostname. To determine the hostname, use the command *hostname* or *ipconfig /all*. The syntax to change the hostname is *netdom renamecomputer <ComputerName> /NewName:<New ComputerName>*. If you changed the hostname, you have to reboot the machine. You can reboot the machine with the command *shutdown /r /t 0*. After the reboot, you can check to see if the hostname is changed by again typing the command *hostname*. If you want to change the hostname with WMI, use the command *wmic.exe ComputerSystem Where Name="%ComputerName%" Rename Name="NewComputerName"*.

Joining a Domain

You need administrative credentials to join a computer to a domain. To accomplish this, use the following command: *netdom join <ComputerName> /domain:<DomainName> /userd:<UserName> /password:******. <ComputerName>* represents the hostname of the machine that is running the Server Core installation, *<DomainName>* is the name of the domain to join, and *<UserName>* is a user account that has permission to join the domain.

Activating the Server

To activate Windows Core Server, you must use a built-in script. The script is named *slmgr.vbs* and can be found in *%windir%\system32*. This script is also used in Windows Vista. If you type the command ***cscript c:\windows\system32\slmgr.vbs –ato***, the product will be activated. With the command ***cscript c:\windows\system32\slmgr.vbs –xpr***, you can check the expiration date for the current license, and with the command ***cscript c:\windows\system32\slmgr.vbs –dlv***, you get the detailed license information. The following are all the command-line options for slmgr.vbs.

```
Microsoft (R) Windows Script Host Version 5.7
Copyright (C) Microsoft Corporation. All rights reserved.

Unrecognized option: /?
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)
-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

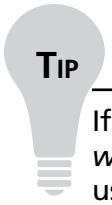
Advanced Options:
-cpky
Clear product key from the registry (prevents disclosure attacks)
-ilc <License file>
Install license
-rilc
Re-install system license files
-rearm
Reset the licensing status of the machine
-upk
Uninstall product key

Volume Licensing: Key Management Service (KMS) Client Options:
-skms <Name[:Port] | :Port>
Set the name and/or the port for the KMS computer this machine will use
-ckms
Clear name of KMS computer used (sets the port to the default)

Enabling Automatic Updates

Of course, it's important to patch the Server Core machine. Microsoft claims that the number of patches is decreased by 60 percent, compared to a full Windows Server 2008 version. Nevertheless, it is advisable to enable automatic updates. If you want to configure automatic updates, use the following commands. To enable automatic updates, type **cscript C:\Windows\System32\Scregedit.wsf /au /4**. To disable automatic updates, type **cscript C:\Windows\System32\Scregedit.wsf /au /1**. To view your current settings, type **cscript C:\Windows\System32\Scregedit.**

wsf /au /v. If you want to configure WSUS, use Group Policy or edit the Registry with the command **regedit**.



TIP

If you easily forget commands, try to at least remember this one *Cscript c:\windows\system32\scregedit.wsf /cli*. It displays some of the most frequently used commands on a Server Core machine.

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 with 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 hotkey:
Ctrl + Shift + Esc

Log off 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 win http proxy set [proxy_name]:[port]

Add, delete, or query a Registry value:
reg.exe add /?
reg.exe delete /?
reg.exe query /?

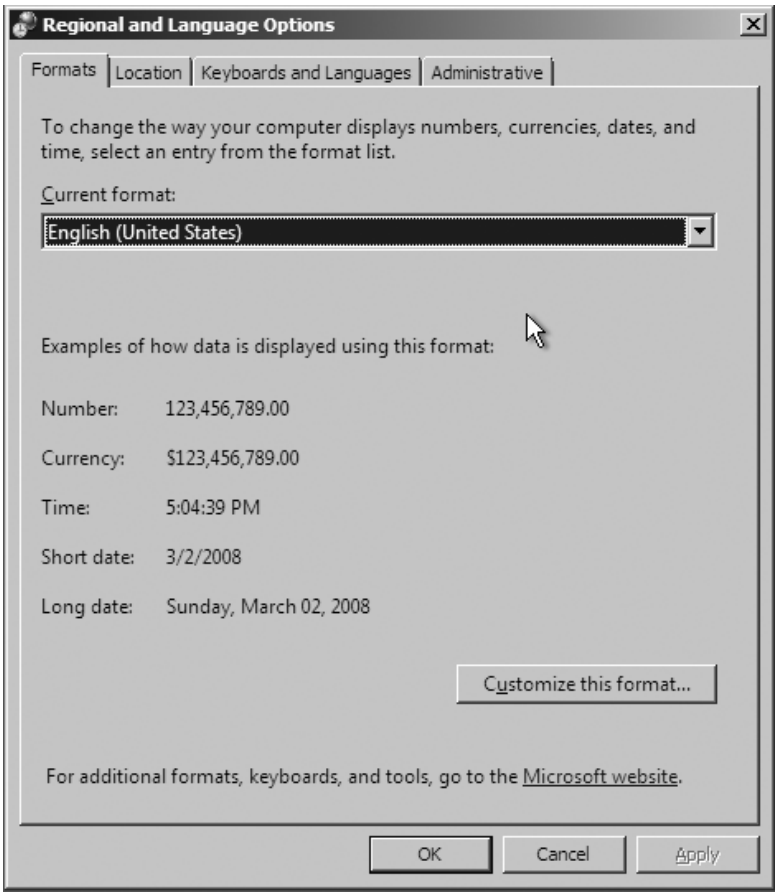
Swapping Mouse Buttons

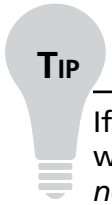
Without the windows you probably don't do a lot of mouse clicking in Server Core. But in the event that you're the only left-handed system administrator in a large server park, you may want to switch the left and right mouse button. After executing the command ***RunDll32.exe USER32.DLL,SwapMouseButton*** at the command prompt, the primary and secondary mouse buttons are switched.

Changing the Regional Settings

You can change the regional settings by specifying the settings in an answer file during an unattended setup, or you can set it manually. If you run the command **control intl.cpl**, you will notice that Server Core is not completely GUI-less (see Figure 7.12). After typing the previous command, the Control Panel applet *regional and language options* will appear. Because of some dependencies on a few low-level GUI DLLs, it is not yet possible to use a complete command-line version of this applet. Of course, it's also possible to edit the Registry with *regedit*, but why should you use it if a GUI is available?

Figure 7.12 Changing the Regional and Language Options





TIP

If you want to change the default numlock behavior, you can enable numlock with the command *reg add HKCU\Control Panel\Keyboard /v InitialKeyboardIndicators /t REG_SZ /d 2*, and disable it with the command *reg add HKCU\Control Panel\Keyboard /v InitialKeyboardIndicators /t REG_SZ /d 0*.

Changing the Date/Time or Timezone

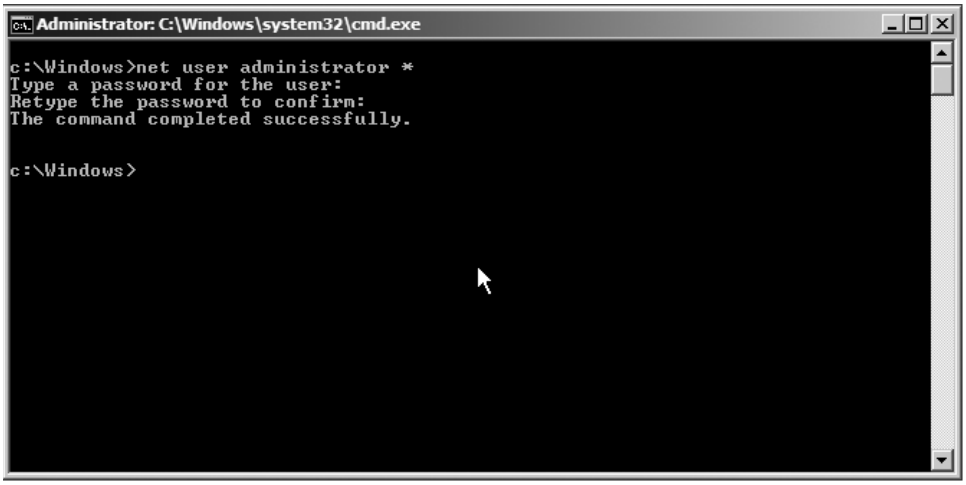
To change the date or time on a Server Core machine, you can respectively use the "old" DOS commands *date* or *time*. If you prefer to use a GUI, you can type **control**

timedate.cpl at the command prompt. After typing this command, the corresponding Control Panel applet will appear. The Control Panel applet *date and time properties* uses the same low-level DLLs as the *regional and language options* applet. For this reason, this applet is still functioning. Microsoft is working on removing these dependencies so these applets probably won't be working in a future release of Server 2008. In the meantime, system administrators can easily manage their servers by using the *intl.cpl* en *timedate.cpl* applets.

Changing the Administrator Password

One of the reasons to install Server Core is because of the reduced attack surface. Fewer binaries are installed, meaning that less software updates are required for the Core Server. Thus, you can conclude that Server Core is more secure. If a security flaw is discovered in a file and the file isn't installed on the system, there won't be a security issue. So why does the installation routine of Server Core allow a blank password? I don't know, but maybe it's better to change this. You can change the password in two ways. The first method is to press **Ctrl+Alt+Del** after the setup completes, then type **Administrator** for the user with a blank password, and press **Enter**. You'll get a prompt to set the password for the Administrator account. But shouldn't we change the password command line since we are administering a Server Core machine? The syntax for this command is easy. *Net user administrator* *. After executing this command, you will get a prompt to change your password. (See Figure 7.13.)

Figure 7.13 Changing the Regional and Language Options



Adding Users to the Local Administrator Group

In some situations, you might want to add users to the local administrators group on a server. The following command adds a user to the local administrator group.

net localgroup Administrators /add <domain>\<username>

If you want to add the user rwisselink sitting in the domain wisselink.local, the command would be:

net localgroup Administators /add wisselink\rwisselink

If you want to delete the user, use the command shown next:

net localgroup Administrators /delete <domain\username>

net localgroup Administators /delete wisselink\rwisselink

Setting the Pagefile

After adding memory to your Server Core machine, it's likely you will want to change the pagefile. Even though you don't have a GUI, it's still possible to change the pagefile. By default, the pagefile is configured by the Windows System. If you want to see the current settings, type ***wmic.exe pagefile list /format:list***. If you want a manually configured pagefile, you must first disable the system-managed pagefile with the command ***wmic.exe computersystem where name="%computername%" set AutomaticManagedPagefile=False***. After executing the previous command, you can manually set the pagefile. It's recommended that you give the initial size and the maximum size the same value. In this example, we set the minimum and maximum pagefile size to 1024MB. Use the command ***wmic.exe pagefileset where name="C:\\pagefile.sys" set initialsize=1024,maximumsize=1024***. After you change the pagefile setting, you must reboot your computer. Use the command ***shutdown /r /t 0***. If you prefer the interactive mode, use ***shutdown /i***. Of course, you'll want to make sure the settings were successful. To check this, use the command ***wmic.exe pagefileset list /format:list***.

Installing Server Core Roles

We can't use Server Manager because it has .NET Framework dependencies, and we can't use the command *servermanagercmd.exe* either for the same reason. If we want to install roles or features on Server Core, we need the command *ocsetup.exe*. *ocsetup* is often used to perform scripted installations of Windows components, and substitutes the Sysocmgr.exe tool that we know from previous Windows versions. *ocsetup* has one

disadvantage when compared with Server Manager. Server Manager carefully checks dependencies when installing roles or features, while *ocsetup* doesn't. This means you have to install or de-install all the roles and features in the correct order. *ocsetup* installs the following file types.

- Microsoft System Installer (MSI) files (Windows Installer service, msiexec.exe)
- Component-Based Servicing (CBS) components (Package Manager)
- MSI or CBS packages that have an associated custom installer .exe file

The full command-line options for ocsetup are ocsetup.exe [/?] [/h] [/help] component [/log:file] [/norestart] [/passive] [/quiet] [/unattendfile:file] [/uninstall] [/x: parameter]. For the full explanation of the flags see Table 7.4.

Table 7.4 Command-Line Options for *ocsetup.exe*

Parameter	Description
/?, /h, /help	Displays help for all options when run with or without options.
Component	The name of the component to be installed or uninstalled. The component name is case-sensitive.
/log:file	Specifies a non-default log file location.
/norestart	The computer is not rebooted even if required after component installation.
/passive	Unattended mode. Progress only.
/quiet	Quiet mode. No user interaction.
/unattendfile:file	The file contains overrides or additions to default configuration settings. Implies passive mode.
/uninstall	Uninstalls the component. Installation is default.
/x: parameter	Additional configuration parameters to be applied when installing a component that requires a custom installer. *ocsetup* will pass these parameters to the custom installer.

To install a role, type the command ***start /w ocsetup <serverrole-name>***. The following examples show the installation commands for all the roles that can be installed on Server Core.

- Active Directory Lightweight Directory Services (AD LDS), *start /w ocsetup DirectoryServices-ADAM-ServerCore.*
- DHCP Server, *start /w ocsetup DHCPServerCore.*
- DNS Server, *start /w ocsetup DNS-Server-Core-Role.*
- File Services:
  - For File Replication service, type *start /w ocsetup FRS-Infrastructure.*
  - For Distributed File System service, type *start /w ocsetup DFSN-Server.*
  - For Distributed File System Replication, type *start /w ocsetup DFSR-Infrastructure-ServerEdition*.
  - For Network File System, type *start /w ocsetup ServerForNFS-Base and start /w ocsetup ClientForNFS-Base*.
- Print Services, *start /w ocsetup Printing-ServerCore-Role.*
- Streaming Media Server, *start /w ocsetup MediaServer*. First, you have to copy the appropriate Microsoft Update Standalone package (MSU) to your Server Core installation, and then run the corresponding .MSU files.
- Web Server, *start /w pkgmgr /iu:IIS-WebServerRole;WAS-WindowsActivationService; WAS-ProcessModel.*
- Hyper-V (only available for X64), type *start /w ocsetup Microsoft-Hyper-V.*
- Active Directory Domain Services (AD DS), *dcpromo /unattend:<unattendfile.xml>*. (See the section titled "Installing Active Directory Domain Services on Server Core.")

Notes from the Underground…

Full IIS Installation

If you want to install all available options from Internet Information Services, copy the following command and paste it into the command prompt in Server Core.

start /w pkgmgr /iu:IIS-WebServerRole;IIS-WebServer;IIS-CommonHttpFeatures;IIS-StaticContent;IIS-DefaultDocument;IIS-DirectoryBrowsing;IIS-HttpErrors;IIS-HttpRedirect;IIS-ApplicationDevelopment;IIS-ASP;IIS-CGI;IIS-ISAPIExtensions;IIS-ISAPIFilter;IIS-ServerSideIncludes;IIS-HealthAndDiagnostics;IIS-HttpLogging;IIS-LoggingLibra-ries;IIS-RequestMonitor;IIS-HttpTracing;IIS-CustomLogging;IIS-ODBCLogging;IIS-Security;IIS-BasicAuthentication;IIS-WindowsAuthentication;IIS-DigestAuthentication;IIS-ClientCertificateMappingAuthentication;IIS-IISCertificateMappingAuthentication;IIS-URLAuthorization;IIS-RequestFiltering;IIS-IPSecurity;IIS-Performance;IIS-HttpCompressionStatic;IIS-HttpCompressionDynamic;IIS-WebServerManagementTools;IIS-ManagementScriptingTools;IIS-IIS6ManagementCompatibility;IIS-Metabase;IIS-WMICompatibility;IIS-LegacyScripts;IIS-FTPPublishingService;IIS-FTPServer;WAS-WindowsActivationService;WAS-ProcessModel

If you want to install a feature, type ***oclist*** to find the required feature name. If you execute this command, you will see a list with all the roles and features that can be installed on Server Core, and whether they are installed or not. The command can't make a distinction between a role or feature. The reason for this is that *oclist* uses a lower-level API, which has no hardcode logic in it. Microsoft wants to keep the command flexible so it's possible to easily add roles and features at a later time. (To see what roles and features are available for Server Core, see Table 7.1 earlier in this chapter.) Type ***start /w ocsetup <feature-name>*** to install a feature. Remember that the role and features names are case-sensitive. The following examples show the installation commands for all the features that can be installed on Server Core

- BitLocker drive encryption, *start /w ocsetup* BitLocker.
- BitLocker remote administration tool, *start /w ocsetup BitLocker-Remote AdminTool*.

- Cluster service, *start /w ocsetup FailoverCluster-Core*.
- Removable Storage Management, *start /w ocsetup Microsoft-Windows-RemovableStorageManagementCore*
- MultipathIo, *start /w ocsetup MultipathIo*.
- Network Load Balancing (NLB), *start /w ocsetup NetworkLoadBalancing HeadlessServer*.
- Quality Windows Audio/Video Experience, *start /w ocsetup QWAVE*.
- SNMP, *start /w ocsetup SNMP-SC*.
- Subsystem for Unix-based applications, *start /w ocsetup :SUACore*.
- Telnet client, *start /w ocsetup TelnetClient*.
- Windows Activation Service, *start /w ocsetup WAS-Windows ActivationService*.
- Windows Backup, *start /w ocsetup WindowsServerBackup*.
- Wins, *start /w ocsetup WINS-SC*.

Administrating Server Core

After installing and configuring the Server Core machine, it's time to administer it. This can be done remotely with WINRM/WINRS or you can use the MMCs that become available after you install the Remote Server Administration Tools or RSAT. It's up to the administrator which tools he or she prefers. To be honest, if I have to choose between the command *dnscmd.exe* or the DNS MMC snap-in, I prefer the snap-in. It's quicker and there's less room for error. The following sections detail the administration tools available for Server Core.

Remote Server Administration Tools (RSAT)

Server Manager is the single all-in-one tool that you generally use to administer a Server 2008 machine. You can only open the tool if you connect via RDP or sit behind the console. Server Manager is not available on a Server Core installation because it needs .NET Framework 2.0 and MMC 3.0, and these two components are not installed on Server Core. Because the option *connect to a different computer* is not available within Server Manager, it isn't possible to connect with this tool to a Server Core machine or even a regular Server 2008 installation.

One of the options available to you to administer Server Core is to use Remote Server Administration Tools (RSAT). You can compare RSAT with the legacy "Adminpack.msi," which was used to administer Windows Server 2003. The RSAT tools are actually a collection of MMC tools. RSAT is a feature component within Windows Server 2008. If you install a role or feature in the GUI-version of Server 2008, the corresponding "RSAT tool" or MMC snap-in is automatically installed. To install all management tools for the Roles available in Server 2008, use the command *ServerManagerCmd.exe -install RSAT-Role-Tools*. To install all feature management tools, use the command *ServerManagerCmd.exe -install RSAT-Feature-Tools*. So, even if a role or feature is not installed on a server, it's possible to install management tools to connect to other servers—for example, Server Core.

NOTE

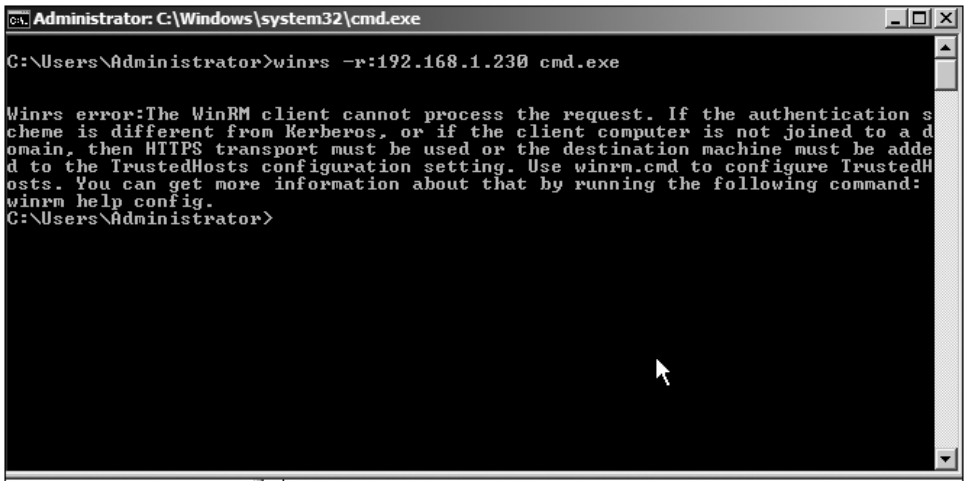
RSAT is in public beta testing at the moment of writing and can only be installed on Windows Vista SP1. With RSAT tools installed, you'll be able to manage Windows Server 2008 servers, including Server Core. With many RSAT tools, it's also possible to manage servers running Windows Server 2003. RSAT will be released as an Out of Band component shortly after Vista SP1 is released.

WINRM/WINRS

WINRM/WINRS has many similarities with psexec. To explain what WINRM/WINRS is, we must first take a look at what psexec actually does. Many of you are familiar with the Sysinternals command-line tools, better know as pstools (from Mark Russinovich). For the download location, click the following link: http://technet.microsoft.com/en-us/sysinternals/bb896649.aspx. One utility of the pstools suite is psexec, which is easy to use. The command *psexec \\192.168.1.230 cmd.exe* starts a remote command session to a server with IP address 192.168.1.230. The big advantage is that you don't have to install any client software. The disadvantage is that psexec doesn't traverse well through firewalls. Here's where WINRM/WINRS comes in. WINRM/WINRS has capabilities similar to those of psexec. WINRM stands for Windows Remote Management, while WINRS stands for Windows Remote Shell. We first need to create a listener with the command *winrm quickconfig*. After executing this

command, a listener is created on port 80. With the following command, we can connect from a remote host to this listener: *winrs –r:<servername> <command>*. In Figure 7.14, you can see we are trying to connect to the server with IP address 192.168.1.230, this time using Winrs. If you get the same error, you're probably working in a workgroup and the default Kerberos authentication mechanism won't work.

Figure 7.14 A WINRM Error



To loosen the security, run the following commands:
Run on the server and client side:

- winrm set winrm/config/service/auth @{Basic="true"}
- winrm set winrm/config/client @{TrustedHosts="<local>"}.

Run on the client side:

- winrm set winrm/config/client @{TrustedHosts="RemoteHost"} and winrm set winrm/config/client @{TrustedHosts="Servername"}.

Be careful, the syntax is case-sensitive. For more configuration details, see the command *winrm help config*.

Managing Server Core with Group Policy

If you have a large amount of Server Core machines, the absolute easiest way to administer Server Core is by using group policy. Just put all the Server Core machines in the right OU and you're done. This will likely prove to be a little more convenient than having to edit the settings on each system individually.

The number of policy settings that can be done has increased from approximately 1,700 in Windows 2003 to around 2,400 in Windows 2008. If you want to do some troubleshooting or other group policy–related tasks, the tools *gpupdate, gpresult*, and *secedit* are still available.

PowerShell

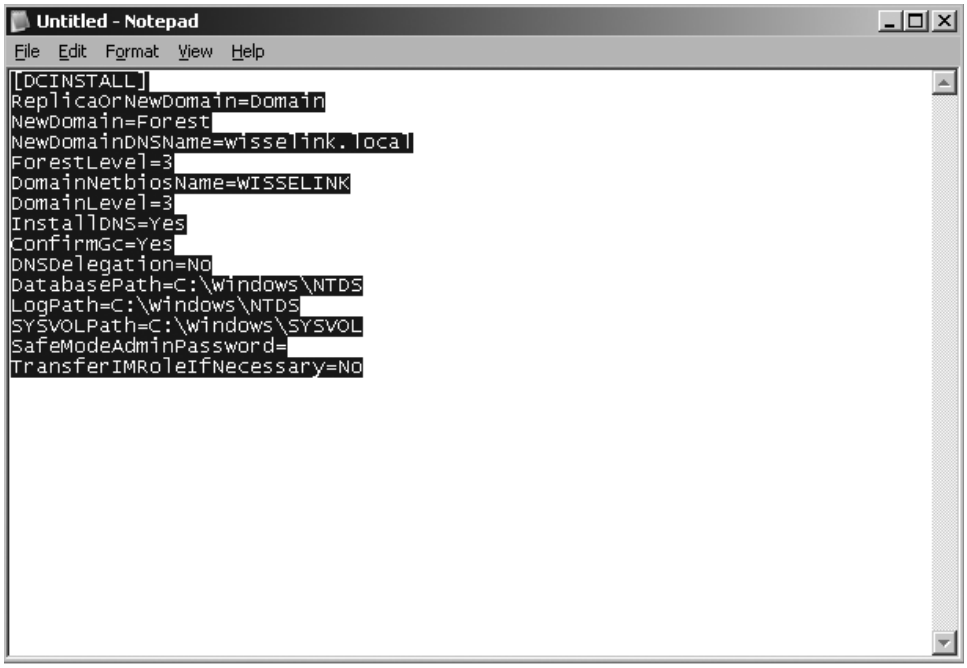
As said before, PowerShell running on Server Core isn't supported. PowerShell requires the .NET Framework, and because the framework has a lot of GUI dependencies, it can't be installed on Server Core. But that doesn't mean you can't use PowerShell from your management workstation to administer a Server Core machine. You are allowed to use the Windows Management Instrumentation (WMI) or Active Directory Services Interface (ADSI) within PowerShell to administer the Server Core box. At the time of this writing, Microsoft was developing a light version of .NET Framework. The expectation is that PowerShell will probably be part of Server Core in a future release.

Installing Active Directory Domain Services on Server Core

So, here is where things get a little tricky. When installing the Directory Services role on a full server installation, we simply open up a **Run** window and type in **Dcpromo**. Then, we follow the prompts for configuration (domain name, file location, level of forest/domain security), and finally restart the system. Installing the role in Server Core isn't that simple, but on the other hand, it's not exactly rocket science either. In order to make this installation happen, we need to configure an *unattended installation file*. An unattended installation file (see Figure 7.15) is nothing more than a text file that answers the questions that would have been answered during the Dcpromo installation. So, let's assume you've created the unattended file and placed it on a floppy disk, CD, or other medium, and then inserted it into the Server Core server. Let's go ahead and install Directory Services:

1. Sign in to the server.
2. In the console, change drives to the removable media. In our example, this is drive E.
3. Once you have changed drives, type **dcpromo.exe /answer:e:\unattended.txt**. Unattended.txt is the name of our unattended file (see Figure 7.15).

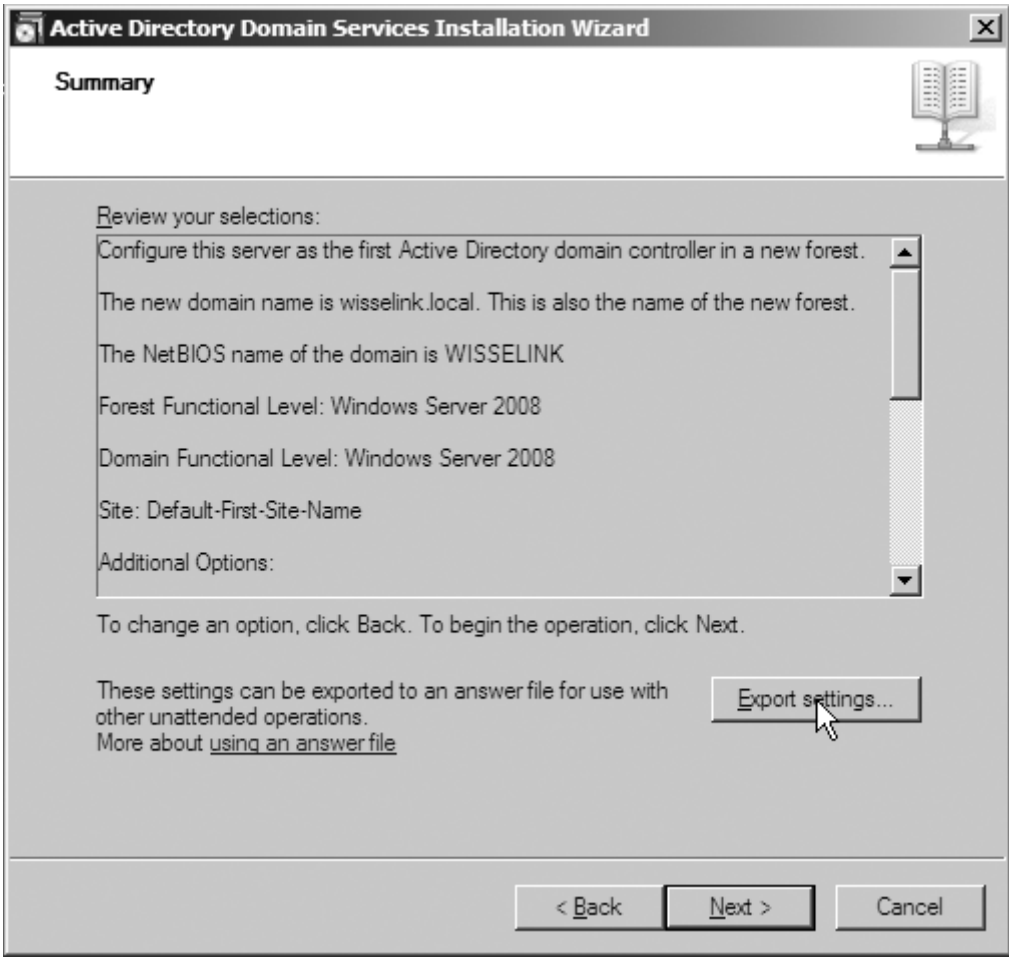
Figure 7.15 A Simple Unattended File



This is the only way to install Active Directory Domain Services on Server Core.

If you don't know how to configure the settings in an unattended file in order to install a domain controller, you can still create an unattended file after typing Dcpromo.exe on a normal Windows 2008 machine. Walk through the screens until you see the one displayed in Figure 7.16. Click the **Export Settings** button and save the file to a suitable location. Afterward, cancel the Dcpromo setup.

Figure 7.16 The Active Directory Domain Services Installation Wizard



Summary

It's obvious why you should choose Server Core for Microsoft Windows 2008. The attack vector is much smaller than a full Windows installation because Server Core contains less code then its big brother. The chance for an attacker to find a security hole is, for this reason, decreased. Because there is less code installed on a Server Core machine, there is also less to manage. This results in a decreased patch managing cycle. The low disk footprint can help you free up some time when you have to plan a big Server Core rollout.

We've learned what roles and features are and which roles and features are available for Server Core. We have seen that the main console from Server Core is the Windows command prompt. We know which components belong to Server Core and have seen that some GUI tools, like Notepad, two Control Panel applets, regedit, and the Windows logon screen, are still available. Most DOS commands still work on Server Core, so reading the command-line reference A–Z is a good start before administrating a Server Core machine.

We provided you with some tips to help you easily recognize which server you're working on, like adding the hostname of the server to the command prompt or changing the background color for the most important servers. If no commands are available to help you with the configuration of Server Core, you can always use regedit. Many of the Registry settings are the same if you compare Server Core with a normal Windows machine. You can change a particular setting on a normal Windows machine, look at how the change affects the Registry, and apply the Registry change on a Windows Core machine. Little software is available for Server Core. In general, the software can be installed with the executable file or with the command msiexec /i productname.msi. Before installing software on a Server Core machine, it's best to first check the official installation instructions. Instead of typing long commands, use the resource kit tool choice.exe to create menu-send batch files. For those administrators out there with less scripting knowledge, it's now easier to administer Server Core because you don't have to remember all those long or difficult commands. You only have to press a number that corresponds to the command.

In the chapter's final section, we learned how to administer Server Core. The section was divided into three separate parts: installing, configuring, and administrating Server Core. After reading the first paragraph, you should have quickly concluded that installing Server Core is rather easy. You can install it manually or unattended. Command-line freaks should have had a lot of fun with the section on configuring

Server Core since all the command-line options are covered there. After the initial configuration is done, you saw that administrators can manage their Server Core machines with the normal MMC tools (or Remote Server Administration Tools RSAT), WS-management, or with group policies.

Solutions Fast Track

Server Core Features

- ☑ Server Core is a stripped-down version of Windows Server 2008. Most of the installing, configuring, and maintenance must be done using the command line.
- ☑ Server Core has fewer services installed and is, for this reason, less vulnerable to exploits and outside attacks.
- ☑ It's expected that the total number of patches will be reduced by 60 percent because the components that require the greater burden of servicing are removed.
- ☑ Server Core uses only 2GB of disk space for operations, and because of this it's a good candidate for large datacenters.

Server Core Components

- ☑ You can divide Server Core into *hardware support components* (disk, net card), a *Coresubsystem* (RPC, Winlogon File Systems, TCP/IP), *infrastructure features* (command shell, performance counters, WINRM/WINRS), and some thin *management tools* to configure IP addresses, join a domain, create users, and so on.
- ☑ With the command *oclist.exe*, you can see what roles and features are installed on the Server Core machine.
- ☑ The following roles are available for Server Core: Active Directory Domain Services (AD DS), Active Directory Lightweight Directory Services (AD LDS), DHCP Server, DNS Server, File Services, Streaming Media Services, Print Server, Web Server (IIS), and Hyper-V.
- ☑ Server Core doesn't support managed code like .NET Framework. For this reason, PowerShell isn't supported on Server Core like the command *servermanagercmd.exe*. *servermanagercmd.exe* is replaced by *ocsetup.exe*. This command lets you install and uninstall roles and features on Server Core.

Server Core Best Practices

- ☑ With a Registry change, you can adjust the default layout from the command prompt. This helps you distinguish the command prompts when you have several remote command boxes open.
- ☑ You can connect to a Server Core machine with Remote Desktop. It's also possible to create an RDP file, which starts (after double-clicking it) a remote command shell session to a Server Core machine.
- ☑ Creating menu-send DOS menus can help you more easily administrate Server Core.

Server Core Administration

- ☑ You can install Server Core unattended or by booting the Windows Server 2008 DVD and selecting Server Core as an installation option.
- ☑ You can remotely manage Server Core machines with the Remote Server Administration Tools (RSAT) installed on Windows Vista with Service Pack 1. RSAT is also a built-in feature component of the full installation version of Windows Server 2008. If you install a role or feature, the corresponding RSAT tool or MMC will be installed on the full version of Windows Server 2008. If you use WINRM/WINRS, you are able to remotely administer Server Core command lines. Just like the previous versions of Windows, it's possible to administer Server Core with group policies.

Frequently Asked Questions

Q: Why should I install Server Core?

A: Because it's more secure due to fewer installed DLLs and services, making it less vulnerable to attacks. Server Core requires less software maintenance than a regular Windows machine and requires only 2GB of disk space.

Q: Which roles are available for Server Core?

A: Active Directory Domain Services, Active Directory Lightweight Directory Services (AD LDS), DHCP Server, DNS Server, File Services, Streaming Media Services, Print Server, Web Server (IIS), and Hyper-V.

Q: How can I see which roles are installed or uninstalled?

A: Execute the command *oclist* at the command prompt.

Q: How do I install a role or feature?

A: Use the command *start /w ocsetup <server role name/feature name>*.

Q: How do I install Active Directory on Server Core if there is no RUN in the Start menu where we can type DCPROMO?

A: First of all, the name Active Directory has been changed to Active Directory Domain Services (AD DS), and you can only install Active Directory Domain Services unattended.

Q: How do I install software on Server Core?

A: If the application has an executable, you can run that. If it is an MSI file, you can use *Msiexec.exe /i <msipackage>*. It's a best practice to install applications in silent mode.

Q: How do you uninstall software on Server Core if there is no Add/Remove Programs?

A: Use *Msiexec.exe /x <msipackage>*. If the application isn't an MSI file but an executable, look for the product documentation. Or type *application.exe /?* for the de-installation options.

Q: Do I have to learn a new scripting language because all the administering and configuring must be done at the command prompt?

A: No. Many commands that were also available in previous versions of Windows can be used on Server Core. A good start is the command-line reference A–Z, which can be found at http://technet2.microsoft.com/windowsserver/en/library/552ed70a-208d-48c4-8da8-2e27b530eac71033.mspx?mfr=true.

Q: Do I have to activate Windows Server Core?

A: Yes, you must activate Windows Server Core within 60 days of installation.