Realtime
publishers
"Leading the Conversation"

The Definitive Guide™ To

Building a Windows Server 2008 Infrastructure

triCerat

Greg Shields

Copyright Statement

Chapter 5: Server Core

At times throughout the career of every IT administrator comes the thought, “There’s got to be a different way to do this.” Administering servers in the data center, out in the field, and within branch offices across the network involves different sets of skills and experience. The processes that work well for daily administration on a Windows server physically residing down the hall are quite different for those that sit down the street or on the other side of the world.

In addition to handling the common tasks associated with the daily care and feeding of Windows servers, administrators are required to possess another suite of skills to keep those servers secure. Windows has long-held a stigma—most often directed from the UNIX world—that it is an easily hackable operating system (OS). This stigma in many ways originates from Windows’ design goals. Windows Server has always tried to be “everything to everyone.” This monolithic structure to the Windows OS makes it highly flexible. The same Windows OS that runs a Web server can also be used as a file server or an email server. Applications are relatively guaranteed to function across all instances of the same OS version. But at the same time, this flexibility introduces numerous additional touch points for a would-be attacker, each of which is a vector for potential compromise.

With Windows Server 2008, Microsoft takes a hard look at that critical thought. In developing Windows Server 2008, Microsoft has realized that in a few special circumstances there is another way to do Windows administration. There are situations where administrators want to sacrifice flexibility in place of greater security, a lower profile, and easier remote administration. Server hardware that would otherwise be seeing its end of life can be repurposed for special needs. Enlightened administrators who are willing to change their processes just a little for these special circumstances can stand to gain through a streamlined OS that sports minimal interfaces and limited configuration options. That “new and different way” is Windows Server Core.

What Exactly Is Server Core?

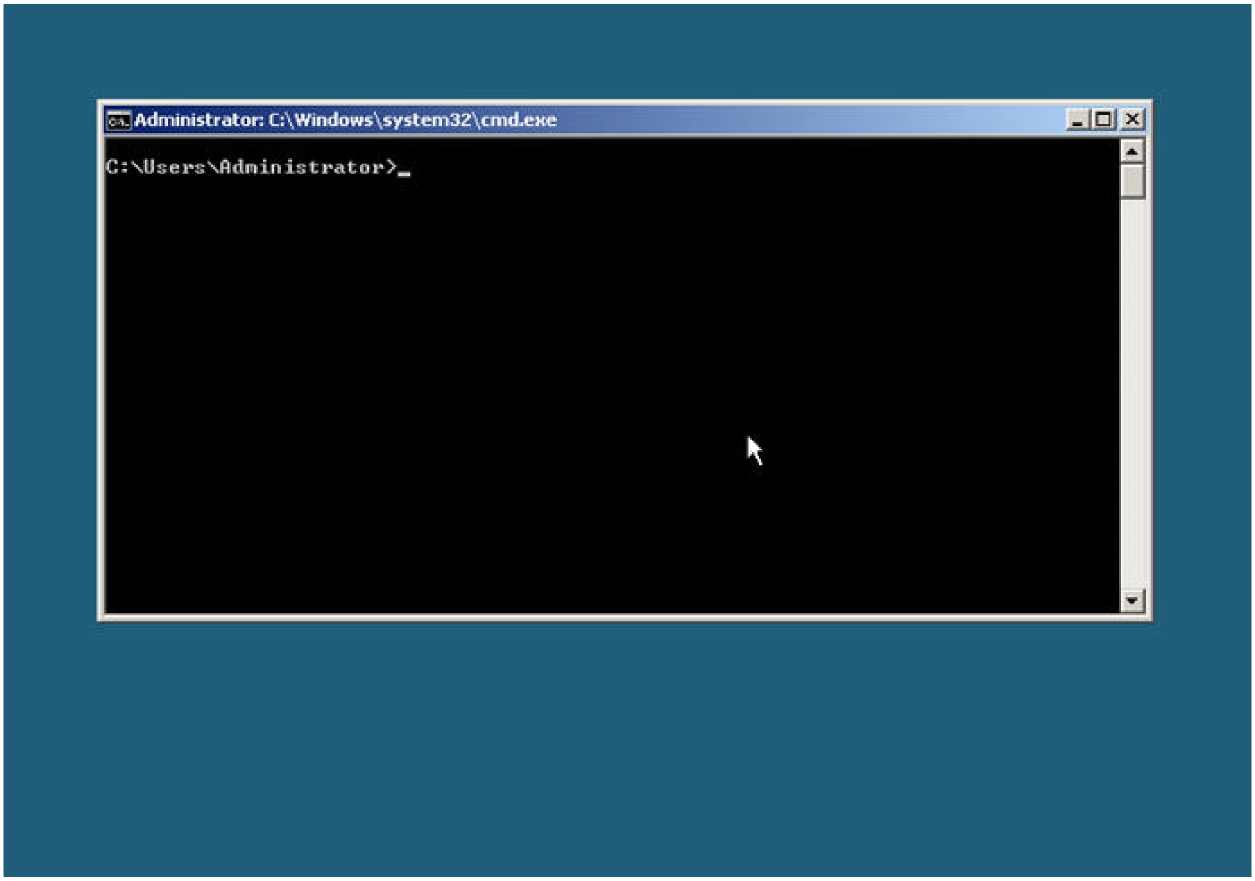
To talk first about what Server Core is, it is in many ways easier to start with a discussion of what it is not. Understanding what Server Core isn't designed to do for your network environment goes far in understanding what it can do. Once you understand its focused potential, you'll find that there are likely places within your network environment where it can be well-positioned. In short, Server Core is not:

- *A completely different OS or OS version.* That being said, Server Core is limited in the types of workloads that it can process. Most of the typical applications that you would install to a "full" Server 2008 instance won't work in Server Core.
- *A different SKU or Windows Server edition.* Microsoft does not consider Server Core to be a separate edition of Server 2008. Thus, buying a copy of Server 2008 nets you the privilege of installing it either as a "full" instance or a Server Core instance. Server 2008 in Standard, Enterprise, or Datacenter editions can be installed with the Server Core option, and each installation enjoys the benefits associated with its related edition.
- *Linux.* Considering how it looks at first blush, Server Core could have been titled Windows Server 2008: DOS Edition. The skills necessary to install, manage, and use Server Core are not all that different than those needed to manage a Windows installation from the command line. Server Core is fundamentally command-line driven, with the command prompt as its primary interface at the console. Traditional Windows command-line tools that do not have a graphical interface are your tools for configuring and managing your Server Core instances at the console. Because of this, if you are familiar with the command-line variants of common GUI tasks, you won't find yourself learning a new OS in order to use Server Core.
- *A solution for every IT need.* Server Core by default can support only a limited number of Roles and Role Services otherwise enjoyed by a "full" Server 2008 instance. Thus, you won't be able to run some workloads on Server Core. Period. In this chapter, we'll talk about the Roles available for Server Core and when and where you're likely to use them.

In considering these bullets explaining what Server Core is not, we can now further define what a Server Core instance is intended to be. First and foremost, Server Core is designed to be a slimmed-down version of the "full" Server 2008 OS. If you remember back to Chapter 1, we discussed how one of the major tenets of Server 2008 was involved with the componentization of the Windows OS. This breaking apart of the traditional processes associated with running the OS has enabled Microsoft to nail up hard dividing lines between functionalities and gain a true understanding of their interrelations.

As we discussed there, most of the fruits of Microsoft's componentization efforts are all behind the scenes and something you'll never see. But others—such as Roles, Role Services, and Features—are very obvious new additions. Once Microsoft took the time to formalize the interconnections between individual components of the Windows OS, it then became a much easier task to figure out which pieces could be pulled out to create this minimal new OS installation option. By trimming out large sections of the Windows OS—the GUI, numerous services, management endpoints, code execution runtime, and so on—Microsoft was able to create Server Core as an OS solution that runs on slower and older hardware while sporting fewer places for a would-be attacker to penetrate.

What's particularly exciting about Microsoft's implementation of Server Core in Server 2008 is that most of the traditional GUI-based remote management interfaces still remain available. Though your console interaction with Server Core will be highly command-line driven, as seen by the sparse desktop in Figure 5.1, most of your administration will be through remote interfaces such as the Remote Server Administration Tools (RSAT).



```
C:\Users\Administrator>_
```

Figure 5.1: Server Core's desktop at the console displays little more than a command prompt window.

> ✎ RSAT are the replacement for the Administration Tools (adminpak.msi) previously found on the Windows Server 2003 media and typically installed to manage workstations like Windows Vista. This toolset includes many of the tools originally found in adminpak.msi as well as a number of new tools and can be downloaded from the Microsoft web site at http://support.microsoft.com/kb/941314.

Positioning Server Core in Your Environment

This understanding of what Server Core does and doesn’t attempt to be should assist you with understanding what it truly is. At the same time, it should help you gain an understanding of where it can be best positioned in your environment. Server Core provides three specific categories of benefit to the IT environment:

- *Reduced hardware needs.* At minimum, Server Core can be installed to a server with a 1GHz x86 processor or a 1.4GHz x64 processor, 512MB of RAM, and 10GB of disk space. Although these are the same minimum requirements for the full version of Windows Server 2008, Server Core’s minimal footprint brings down its effective needs closer to the minimums than seen with “full” instances. This is particularly useful when environments want to retain aging server hardware that would otherwise be nearing the end of its operational life cycle. It is also handy when Server Core is used to host virtualization environments such as Hyper-V. With Server Core as the primary partition in a Hyper-V installation, the reduced resource requirements free processing power for residing virtual machines.
- *Reduced attack surface.* Microsoft patching touches all components of the Windows OS. Thus, when fewer components of the OS are present on a particular server, this reduces the number of patches required to keep the instance up-to-date. With fewer interruptive patches required for a Server Core installation and fewer touch points for a potential attacker, Server Core’s reliability and security are greatly enhanced.
- *Reduced management requirements.* With the reduction in capability to host server functionalities comes an equal reduction in the level of management required for a Server Core instance. Thus, managing Server Core will likely involve less time than with a “full” instance. Going further with this concept is the nature of command-line management itself. Because individual commands used to manage a Server Core instance can be easily wrapped into batch files or scripts, you’ll quickly develop a pool of configuration scripts useful for rapidly completing most major tasks.

☞ If Server Core’s initial learning curve frustrates you, consider this last point thoroughly. Once you’ve completed educating yourself on the new commands necessary to get a Server Core instance operational, keep those commands close at hand. You’ll be able to consolidate them into a small number of batch files and/or scripts for later configuration of additional systems. We’ll talk about many of the critical ones you’ll need to get started later in this chapter.

Considering these categories of benefit along with its architecture, there are a few areas within your IT environment where positioning Server Core can best fulfill your needs:

- *Branch office environments.* In many cases, branch office IT environments don't include the same level of security and control as those at a business' primary location. When these small environments determine the need for server resources to be hosted on-site at the branch office, those server resources often find themselves stored in quasi-secured locations such as closets and under the desks of semi-trusted individuals. This reduction in overall security in many ways makes these branch office servers some of the highest-risk assets in an IT infrastructure.

  The graphical nature of the "full" Windows OS can also make challenging the administration of these servers across latent WAN connections. By leveraging Server Core in these environments, it is possible to use command-line management and the added intrinsic security of Server Core to better handle the needs of branch offices. Also, the resource needs of branch offices are typically lower, so Server Core's reduced hardware footprint enables the use of lower-performing or otherwise end-of-life hardware.

- *Infrastructure servers.* Infrastructure servers such as DNS, DHCP, domain controllers, and file servers typically have an overall lower usage profile than other servers. These servers often fulfill a small function or even a single function that does not require substantial hardware resources. In these cases, Server Core's reduced hardware footprint and streamlined OS makes it an excellent fit for infrastructure servers. Server Core's streamlined OS reduces the chance for application or configuration conflict, while its hardware needs allow otherwise end-of-life machines to remain in service for low-use needs.

- *Virtualization.* In many ways the opposite of the previous two examples, Server Core works well in the high-use environments typically seen on virtualization hosts. For virtualization platforms such as Hyper-V, a primary partition is needed to handle the functionality of the host machine. That primary partition is not a virtual machine, but its presence and functionality is necessary for the host to support residing virtual machines. Server Core requires substantially fewer resources than a "full" instance of Server 2008. This reduced footprint enables more resources to be targeted for use by residing virtual machines, which increases the capacity for virtual machine workload as well as the number of simultaneously hosted virtual machines on the host.

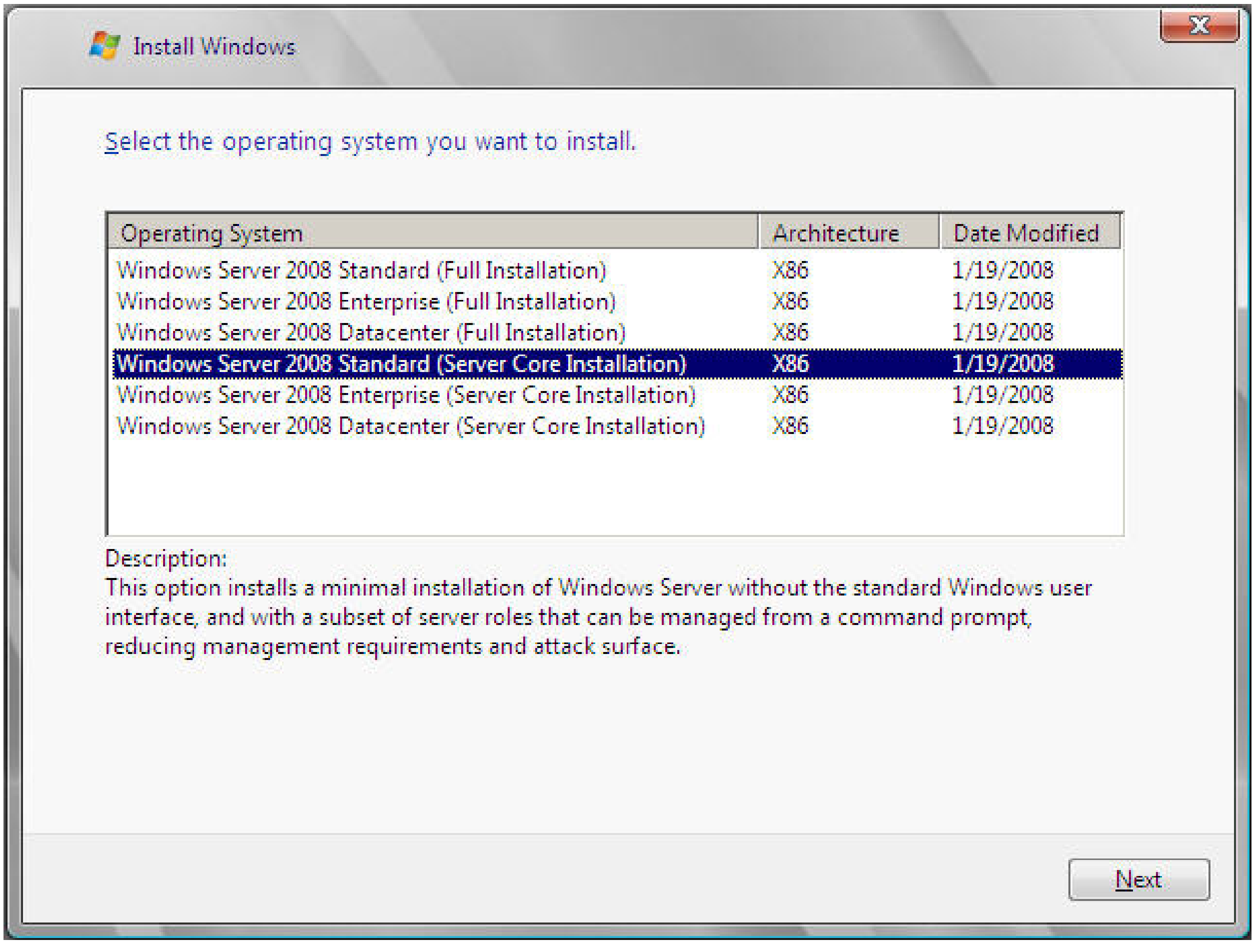
Installing Server Core

The installation of Server Core involves the same processes as installing a “full” instance of Server 2008. Server Core can only be installed through the Server 2008 media as it is an installation option rather than an installable Role or Role Service. Considering this, there are some important notes considering the installation of Server Core:

- It is not possible to upgrade from a previous version of Windows Server to a Server Core installation. A clean installation is the only supported option.
- It is not possible to upgrade from a full installation of Server 2008 to Server Core.
- It is not possible to upgrade from Server Core to a “full” installation of Server 2008.

Considering these gotchas, the first step in installing Server Core is to simply insert the Server 2008 media into the drive of your computer and start the machine. Then, use the following steps to complete the installation:

1. When the Install Windows screen appears immediately after startup, click Next.
2. At the resulting screen, click Install Now.
3. A screen will appear that looks similar to Figure 5.2. You’ll see that Server Core installation options appear for each edition of the OS. Select this version of the correct edition for which you have purchased, and click Next.
4. Select the *I accept the license terms* check box, and click Next.
5. Click Custom (advanced) to perform a clean installation. This should be the only option available.
6. In the resulting screen, you will be provided options for the drive to install Windows. Select the appropriate drive, and click Next. If additional tasks are required, such as the loading of disk drivers or the formatting of existing drives, those can be done through the interface at this point.



Select the operating system you want to install.

Operating System	Architecture	Date Modified
Windows Server 2008 Standard (Full Installation)	X86	1/19/2008
Windows Server 2008 Enterprise (Full Installation)	X86	1/19/2008
Windows Server 2008 Datacenter (Full Installation)	X86	1/19/2008
Windows Server 2008 Standard (Server Core Installation)	X86	1/19/2008
Windows Server 2008 Enterprise (Server Core Installation)	X86	1/19/2008
Windows Server 2008 Datacenter (Server Core Installation)	X86	1/19/2008

Description:
This option installs a minimal installation of Windows Server without the standard Windows user interface, and with a subset of server roles that can be managed from a command prompt, reducing management requirements and attack surface.

Figure 5.2: The Server 2008 installation is the only location where Server Core can be selected for installation.

After these very simple steps, the Server Core instance will begin its installation. Note the speed at which the installation completes. A Server Core installation uses substantially fewer files, which decreases the total time necessary to complete its installation.

Configuring Server Core

As with the "full" installation of Server 2008, Serve Core's install routine does not ask most of the questions you're used to seeing in previous OS versions. Thus, those configurations must be done after the installation. In the "full" version, the *Initial Configuration Tasks* wizard handles many of these needs. But with Server Core, there is no GUI, so there is no Initial Configuration Tasks. Thus, to get your newly installed Server Core instance on the network, added to the domain, and generally ready for management through remote utilities, there are several actions that need to be completed from the command line.

If you are unfamiliar with the command-line adjuncts for the GUI configurations you're used to seeing in the "full" version, this process can be somewhat daunting. But know that the process to set up an initial configuration requires only a short setup before the machine is ready for remote administration. In this section, I'll provide you the short list to get you started. Once we get our new machine's network connection correctly configured, we'll then discuss customization elements as well as how to begin the process of installing Roles, Role Services, and Features.

Initial Configuration

The first step after the installation process completes and the machine reboots is the initial login. After the installation, a Server Core instance can be logged into using the *administrator* username and a blank password. Upon logging in, the interface will require an initial password change. Complete this process to bring forward the Server Core user interface.

The resulting desktop looks like the screen we've already seen in Figure 5.1. Once there, use the following set of tasks to complete an initial configuration:

1. First, reset the password for the administrator account if necessary. If the initial administrator password you just set requires changing either now or at a later point, you can do this using the command

   ```
   net user administrator {password}
   ```

2. As with a "full" version installation, the initial computer name will be a randomized string of characters. That computer name can be changed to something more appropriate to your organization's naming standards using the command

   ```
   netdom renamecomputer %computername% /newname:{newComputerName}
   ```

3. Restart the computer to accept the name change. To immediately restart the computer, forcing closed all running processes and applications, use the command

   ```
   shutdown /r /f /t 0.
   ```

4. Once the reboot is complete, identify the network interfaces available on the computer. Do so with the command

   ```
   netsh interface ipv4 show interfaces
   ```

For a server with a single interface, the result from this command may resemble the result that Listing 5.1 shows.

```
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
```

Listing 5.1: The result from running the* netsh interface ipv4 show interfaces *command.

5. You will use the interface name information from step 4 to change the IP address, subnet mask, and default gateway of this server. Do so all in one step using the command

   ```
   netsh interface ipv4 set address {interfaceName} static
   {ipAddress} {subnetMask} {defaultGateway}
   ```

   Replace *{interfaceName}* with the name received from step 4. In the example shown in Listing 5.1, that name is “Local Area Connection” (with the quotes).

6. This server will require name resolution in order to connect to a domain. Connect it to a DNS server using the same interface name as with the previous step and using the command

   ```
   netsh interface ipv4 add dnsserver name={interfaceName}
   address={dnsServerIpAddress} index=1
   ```

7. It might be necessary to install and activate the license key for this server. That license key can be installed via the command line using the *slmgr* tool. This tool is used in Vista and Server 2008 for managing all components of licensing from the command line. Install and activate a license key using the twin commands

   ```
   slmgr -ipk {licenseKey}
   ```

   and

   ```
   slmgr -ato
   ```

8. Next, you’ll want to join this computer to the domain. Once the Server Core instance is joined to the domain, Group Policies that relate to the instance will begin applying, which will assist with some of the further configuration tasks. You can join this computer to the domain using the command

   ```
   netdom join {computerName} /domain:{domainName}
   /userD:{domainUsername} /passwordD:{domainPassword}
   ```

9. Reboot the computer to complete joining the computer to the domain.

Though at first blush, this can seem like a complicated process that would normally be a few mouse clicks though a GUI, remember that command-line actions are automatically easily scriptable. It is trivial to group all these commands together into a few batch files that complete all the necessary pieces for you all at once. Consider two batch files:

Batch File 1

```
netdom renamecomputer %computername% /newname:w2008c
shutdown /r /f /t 0
```

Batch File 2

```
netsh interface ipv4 set address “Local Area Connection” static 192.168.0.20 255.255.255.0
            192.168.0.1
netsh interface ipv4 add dnsserver name=dnsserver1 address=192.168.0.10 index=1
slmgr -ipk XXXXX-XXXXX-XXXXX-XXXXX-XXXXX
slmgr -ato
netdom join w2008c /domain:realtime-windowsserver.com /userD:administrator
            /passwordD:P@ssw0rd!
shutdown /r /f /t 0
```

Or, if you’re not a fan of batch files but like using scripted installations using Windows System Image Manager (WSIM), it is also possible to create an autounattended.xml file that installs Server Core with many of these configurations already pre-generated. By using WSIM like we discussed in Chapter 1 to create unattended installation files, this entire process can be rolled into the core installation itself.

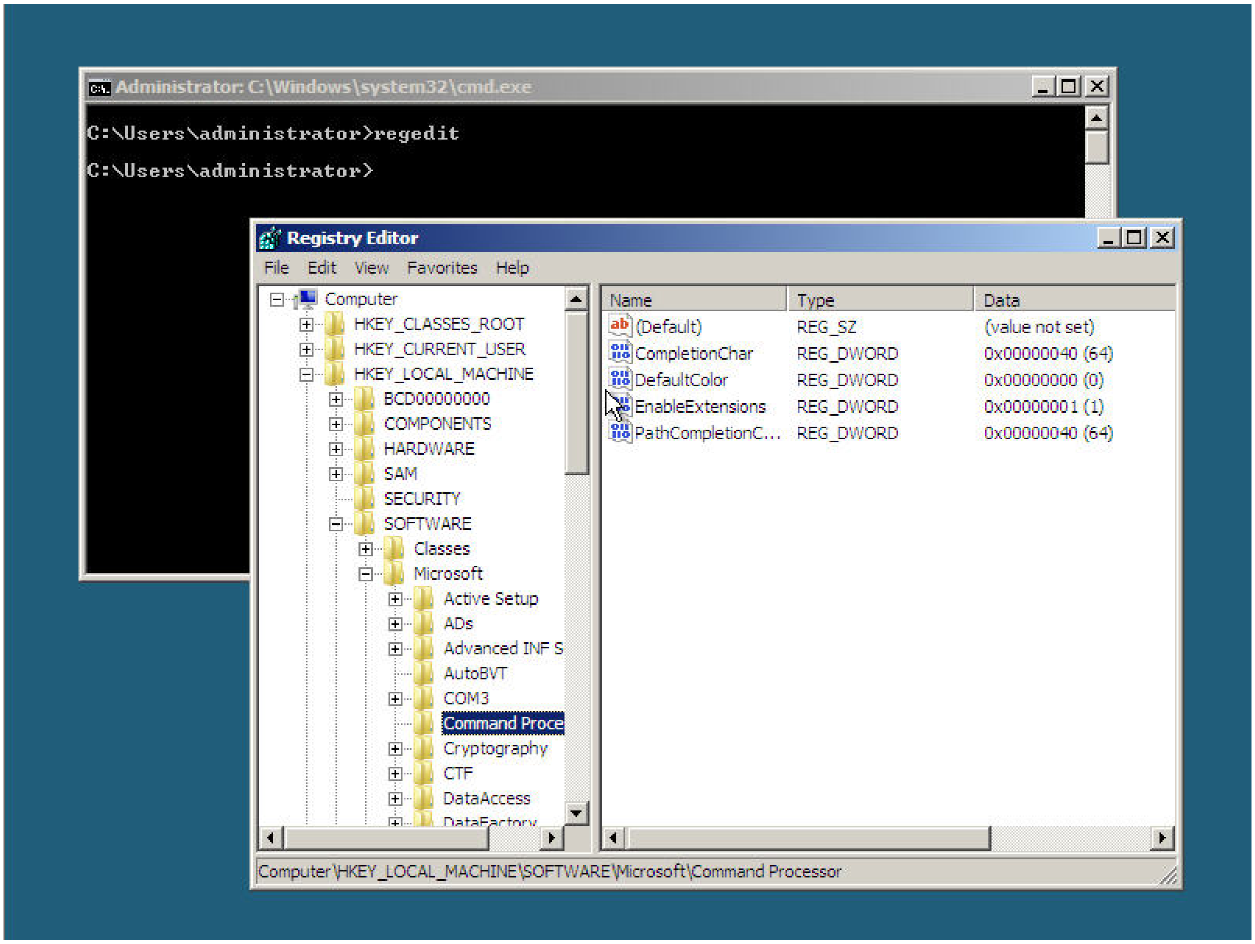
Customizing Server Core

Even upon the conclusion of the previously listed tasks, there may still be several customizations you want to add to your Server Core console to make comfortable your later management activities. Once these tasks are complete, your Server Core instance will still look a lot like what we first saw back in Figure 5.1. Also, unless Group Policies are created that specifically set these next few values, they remain set with their default configurations. In this section, we’ll talk about some of the additional commands you might want to run to further configure your Server Core instance.

In many cases, the process to customize the interface involves directly manipulating the registry. One of the few Server Core GUI tools available at the console is Registry Editor, launched with the

```
regedit
```

command (see Figure 5.3). But in keeping with the command-centric nature of Server Core, the following registry manipulations will use the command-line Registry Editor tool *reg*.



```
C:\Users\administrator>regedit

C:\Users\administrator>
```

Name	Type	Data
(Default)	REG_SZ	(value not set)
CompletionChar	REG_DWORD	0x00000040 (64)
DefaultColor	REG_DWORD	0x00000000 (0)
EnableExtensions	REG_DWORD	0x00000001 (1)
PathCompletionC...	REG_DWORD	0x00000040 (64)

Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Command Processor

Figure 5.3: The same Registry Editor seen with other versions of Windows is available in Server Core, launched using the regedit ***command.***

This tool is available in virtually every edition of Windows and is useful for adding and removing keys and values as well as manipulating existing keys and values. Learning the effective use of the *reg* tool is useful for scripting many of these operations, and experience with it dovetails well into the same processes you can use to manage “full” Server 2008 instances:

- *Removing unused IPv6 interfaces.* Most IT environments haven’t yet made the jump to IPv6. But with Server 2008, Microsoft has made IPv6 an important part of the networking stack. In fact, if IPv6 functionality is not removed from your Server Core (or Server 2008) instance, you might get IPv6-related responses occasionally when running common networking commands. To prevent this, it is possible to remove IPv6 functionality from your Server Core instance by hacking the registry. Do so with the command

  ```
  reg add HKLM\SYSTEM\CurrentControlSet\Services\tcpip6\Parameters
  /v DisabledComponents /t REG_DWORD /d 0xff
  ```

- *Enable and Configure the Windows Firewall.* The Windows Firewall is significantly improved in Server 2008, with a large effort towards making the firewall easily manageable through Group Policy. It can, however, also be activated and configured individually through the command line or through the *Windows Firewall with Advanced Security* console in Administrative Tools. The command-line tool *netsh* is typically used to manage Windows Firewall connections, though the syntax to use this command can be somewhat challenging and lacks much of the easy-to-use configuration interfaces available in the GUI console.

  If you do not plan to use Group Policy to manage your Windows Firewall but instead want to manage it for the local machine only, it is possible to enable remote management of the firewall through another computer. Do so with the command

  ```
  netsh advfirewall set currentprofile settings remotemanagement
  enable
  ```

  Once entered, create a new MMC on a “full” Server 2008 or Vista machine using the *mmc* command. Click File | Add/Remove Snap-In and add the *Windows Firewall with Advanced Security* snap-in to the console. In the resulting screen, you’ll be asked to provide the computer whose firewall you want to manage. Enter the name of your Server Core computer into the resulting screen. You can now remotely manage the firewall of your Server Core computer through the GUI.

- *Manage other configurations and connections.* The settings of six additional configurations for your Server Core instance are consolidated into the *scregedit.wsf* tool. This script is designed to assist with the configuration of a series of somewhat unrelated configurations. The configurations that can be managed through scregedit.wsf are outlined in Table 5.1 for better clarity. To use the scregedit.wsf tool, first navigate to the C:\Windows\system32 folder and use the command format

  ```
  cscript scregedit.wsf {configuration} {value}
  ```

For each, to view the currently set value, use the command

```
cscript scregedit.wsf {configuration} /v.
```

Scregedit Configuration	Explanation
cscript scregedit.wsf /AU 4	Automatic Updates: Enable Automatic Updates and automatically download and install updates.
cscript scregedit.wsf /AU 1	Automatic Updates: Disable Automatic Updates
cscript scregedit.wsf /AR 0	Terminal Services: Allow Remote Administration Connections
cscript scregedit.wsf /AR 1	Terminal Services: Disable Remote Administration Connections
cscript scregedit.wsf /CS 1	Terminal Services: Require Network Level Authentication for Terminal Services Connections. This setting requires the Remote Desktop Client to be at version v6.0 or greater.
cscript scregedit.wsf /CS 0	Terminal Services: Do not require Network Level Authentication for Terminal Services Connections. This setting allows Remote Desktop Clients of any version to connect to the server.
cscript scregedit.wsf /IM 1	IPSec Monitor: Allow remote management of IPSec using the IPSec Monitor.
cscript scregedit.wsf /IM 0	IPSec Monitor: Do not allow remote management of IPSec using the IPSec Monitor.
cscript scregedit.wsf /DP {value}	DNS SRV: Changes the priority for Domain Controller DNS SRV records for this server to {value}. This number can be a value from 0 to 65535. The recommended value is 200.
cscript scregedit.wsf /DW {value}	DNS SRV: Changes the weighting for Domain Controller DNS SRV records for this server to {value}. This number can be a value from 0 to 65535. The recommended value is 50.
cscript scregedit.wsf /CLI	Command Line Reference: This command provides a list of common commands used at the command line to manage a Server Core instance.

Table 5.1: The scregedit.wsf command can be used to set a number of Server Core configurations.

- *Viewing and manipulating running processes.* Though its instantiation is not obvious, it is possible to launch the standard Task Manager to view processes, running applications, and the performance of the server. Do so using the Control-Alt-Delete sequence to bring forward the Windows security screen. Then click Start Task Manager.
- *Creating additional command prompt windows.* If you like to multi-task with more than just one command prompt window, it is possible to create additional ones through the Task Manager. First, bring forward the Task Manager as explained in the previous bullet. Then, click File | New Task (Run…). In the resulting window, enter

  ```
  cmd
  ```

  and click OK. Any number of additional command prompt windows can be created using this method.

✐ When you bring forward the *New Task (Run…)* window, you'll see the Browse button. But, if you click the button, you'll notice that nothing happens. Here and in a number of places within the Server Core OS, there are elements and buttons that don't work. In this case, the components that run Windows Explorer are not present in a Server Core instance. So, if you find a button that doesn't work, don't fret. It is possible that its functionality has been removed.

- *Modifying Time, Date, Regional, and Language Options.* Though most GUI controls have been removed, two Control Panels actually still function within Server Core. These are the Date and Time control panel as well as the Regional and Language Options control panel. To configure these settings, use the commands

  ```
  control intl.cpl
  ```

  and

  ```
  control timedate.cpl
  ```

- *Changing the Screen Resolution.* Though Server Core is predominantly focused on command-line use, it still resides within a graphical quasi-desktop. Hence, the blue-green screen upon which the command prompt window appears. To change the resolution of the screen at the console, you can do so at command line by modifying the registry. First, you must determine the hexadecimal values for width (the "x" axis) and height (the "y" axis) of your desired resolution. This can be done using the Windows calculator. As an example, the 800 × 600 resolution converts to the hexadecimal values of 320 × 258. Each Server Core instance will also have a number of GUID values in the registry location HKLM\System\CurrentControlSet\Control\Video. Trial and error will assist you with locating the proper value to change for your console display. Then, plug those values into the two commands

  ```
  reg add HKLM\SYSTEM\CurrentControlSet\Control\Video\{GUID}\0000
  /v DefaultSettings.XResolution /t REG_DWORD /d {xAxisValueInHex}
  ```

  and

  ```
  reg add HKLM\SYSTEM\CurrentControlSet\Control\Video\{GUID}\0000
  /v DefaultSettings.YResolution /t REG_DWORD /d {yAxisValueInHex}
  ```

- *Changing the number of colors.* With the correct GUID information for your console display, it is also possible to change the number of colors displayed on the desktop. The number of colors is determined by a value known as bit depth. For bit depth, 8 bits is equal to 16 colors, 15 bits is equal to 256 colors, 16 bits is equal to High Color, and 24 bits is equal to True Color. Use this information as well as your GUID in the command

  ```
  reg add HKLM\SYSTEM\CurrentControlSet\Control\Video\{GUID}\0000
  /v DefaultSettings.BitsPerPel /t REG_DWORD /d {new size in hex}
  ```

Installing Roles, Role Services, and Features

After you have the interface configured and customized to your liking, the next step is to install the necessary Roles, Role Services, and Features that enable the desired functionality of your server. As we've already discussed, Server Core does not include all the functionality associated with every component normally found in the "full" edition of Server 2008. Server Core is limited to only a small subset of the complete set of Roles, Role Services, and Features. That subset effectively includes the following Roles and their associated Role Services.

> ✎ Note that for each Role, not all Role Services are necessarily available. As an example, for the Web Server (IIS) Role, only a small subset of the more than 40 possible components is available.

The following list highlights the Roles available in Server Core as well as their titles as labeled in the interface:

- Active Directory Domain Services (DirectoryServices-DomainController-ServerFoundation)
- Active Directory Lightweight Domain Services (DirectoryServices-ADAM-ServerCore)
- DHCP Server (DHCPServerCore)
- DNS Server (DNS-Server-Core-Role)
- File Services (FRS-Infrastructure)
- Web Server (IIS) (IIS-WebServerRole)
- Print Services (Printing-ServerCore-Role)
- Streaming Media Services (MediaServer)
- Hyper-V (Microsoft-Hyper-V)

> 💣 Be aware of the restrictions not only on Roles but also Role Services when considering what workload you intend your Server Core instance to host.

Server Core also has the ability to host a small subset of features as well. These features typically relate to server-based functions and for obvious reasons do not include most of the typical client-side management utilities.

- Bitlocker (BitLocker)
- Client for NFS (ClientForNFS-Base)
- Failover Clustering (FailoverCluster-Core)
- Web Server Management Tools (IIS-WebServerManagementTools)
- Removable Storage Management (Microsoft-Windows-RemovableStorageManagement Core)
- MultiPath I/O (MultipathIo)
- Network Load Balancing (NetworkLoadBalancingHeadlessServer)
- Quality Windows Audio Video Experience (QWAVE)
- Server For NFS (ServerForNFS-Base)
- SNMP (SNMP-SC)
- Subsystem for UNIX Applications (SUACore)
- Telnet Client (TelnetClient)
- Windows Server Backup (WindowsServerBackup)
- WINS (WINS-SC)

> Be aware that for some components, an additional download and installation may be required, as in the case of the Hyper-V role.

Although the “full” version of Server 2008 includes a native command-line tool for automatically installing and uninstalling these components in *servermanagercmd.exe*, that tool does not function on the Server Core OS. Because of Server Core’s inability to run managed code, it requires another tool to manage package installation and uninstallation. Two commands are used in Server Core for this activity. The first, *oclist.exe* is used in much the same way as servermanagercmd.exe is used with its -*query* switch. Running oclist.exe on a Server Core instance shows the components that can be installed as well as those that are already installed. Listing 5.2 shows an example snippet of the results from this command. To use the command, simply run

```
oclist
```

from the command prompt.

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

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

=======================================================================
Microsoft-Windows-ServerCore-Package
Not Installed:BitLocker
Not Installed:BitLocker-RemoteAdminTool
Not Installed:ClientForNFS-Base
Not Installed:DFSN-Server
Not Installed:DFSR-Infrastructure-ServerEdition
Not Installed:DHCPServerCore
Not Installed:DirectoryServices-ADAM-ServerCore
Not Installed:DirectoryServices-DomainController-ServerFoundation
Not Installed:DNS-Server-Core-Role
Not Installed:FRS-Infrastructure
Not Installed:IIS-WebServerRole
  |
  |--- Not Installed:IIS-FTPPublishingService
  |    |
  |    |--- Not Installed:IIS-FTPServer
  |    |
  |
  |
  |--- Not Installed:IIS-WebServer
  |    |
  |    |--- Not Installed:IIS-ApplicationDevelopment
  |    |    |
  |    |    |--- Not Installed:IIS-ASP
```

Listing 5.2: The result from running the **oclist** ***command.***

You'll see in Listing 5.2 that components are tiered to show dependencies. Components that rely on other components are tiered for easy reading within the command prompt interface. Installed and not installed components are marked in the result, showing you which components are already functioning on the server.

Installed components operate much the same here as in the "full" version of Server 2008, in that once installed, there are usually more configurations required to bring them to the needed level of functionality.

What is good about how Microsoft has broken apart the components of Server Core is that the remote management interfaces for making use of installed components can still be used through other interfaces. So, once the component is installed on the console, it is possible to then manage that component through your typical suite of management tools.

Installing any of these components is done using the ocsetup.exe command. This command includes the necessary functionality for setting up logging of the installation, pointing to unattended installation files, and supplying parameters to an underlying installer. One quirk about using this command is that it does not provide a response back to the console showing the success or failure of the component installation. It also by default returns control immediately back to the console as it completes an installation in the background. Thus, some additional commands are necessary to ensure that control is not returned until the installation is complete.

For example, to install DNS Server onto your Server Core instance and to delay returning control back to the console until after the installation completes, use the command

```
start /w ocsetup.exe DNS-Server-Core-Role
```

> Beware! Although the Server Manager and its servermanagercmd.exe command-line adjunct will locate and automatically install the proper dependent components, ocsetup will not. If a component you want has dependent components, you'll need to install them manually.

Though *ocsetup.exe* doesn't directly provide information about the success or failure of the installation, it is possible to get this information through either of two locations. First, by re-running the oclist command, you can get an immediate heads-up response regarding the success of the installation. Notice in Listing 5.3 how after the DNS Server and File Services components have been installed, the value Not Installed has changed to Installed for these two components.

```
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
  Installed:DNS-Server-Core-Role
  Installed:FRS-Infrastructure
Not Installed:IIS-WebServerRole
```

Listing 5.3: The result from running the* oclist *command after a component installation.

In the case in which additional information is needed to confirm the installation or determine why an installation didn't complete successfully, the Component-Based Servicing log includes detailed information about the progress of the installation. This debug-level log, located at *%WINDIR%\logs\cbs\cbs.log* includes substantial information about the progress of the installation and its success. Listing 5.4 shows a snippet from this log associated with the installation of the FRS-Infrastructure installation. For space, not all lines from the log are copied; only those that relate to the initiation and completion of the installation.

```
2008-04-10 11:21:54, Info CBS  Exec: Installing Package: Microsoft-
Windows-FileReplication-Package~31bf3856ad364e35~x86~~6.0.6001.18000,
Update: FRS-Infrastructure, InstallDeployment: x86_microsoft-windows-
f..lication-
deployment_31bf3856ad364e35_6.0.6001.18000_none_ff9be9468cd78e7e

[snip]

2008-04-10 11:21:56, Info CBS  Pkgmgr: Completed installing selectable
updates for: Windows Foundation, hr: 0x0
```

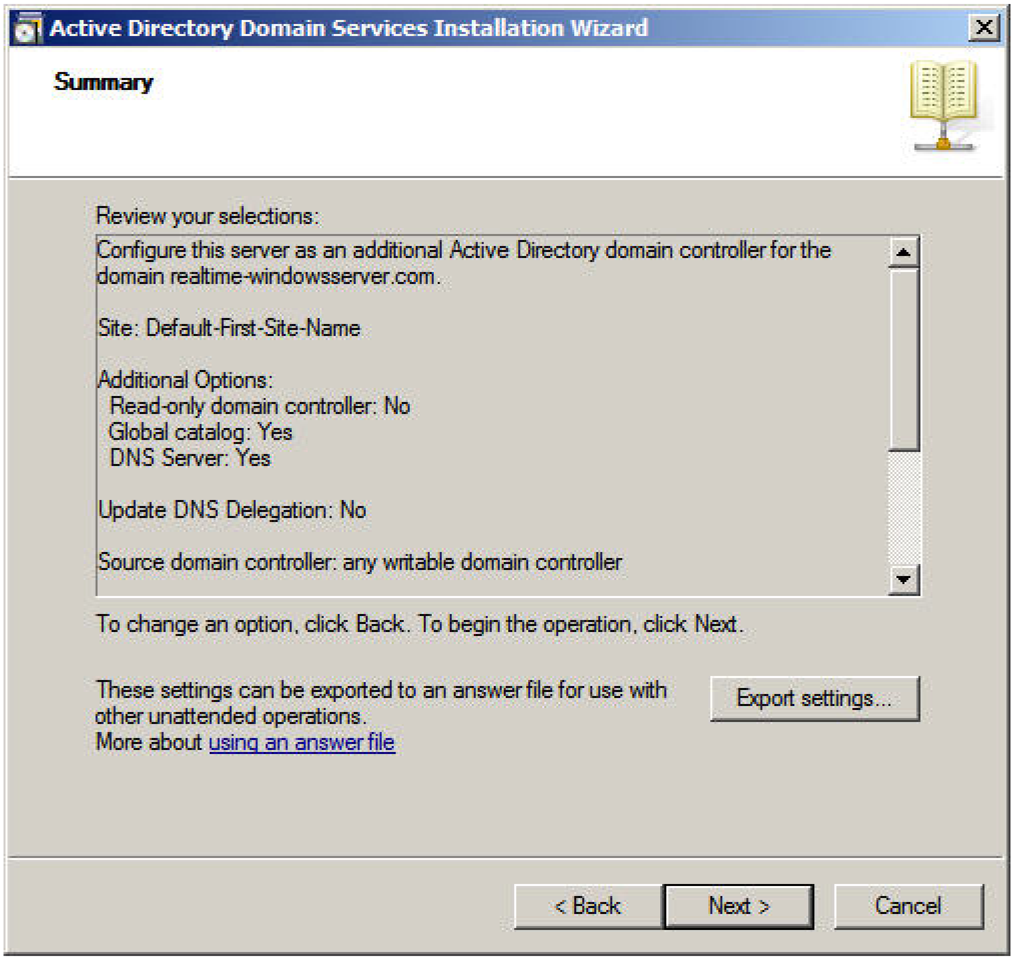
Listing 5.4: The CBS log includes detailed information about the success or failure of a component installation. Two lines from the log are included that show the beginning and end of the FRS-Infrastructure component installation.

☞ It is also possible to use the ocsetup command to install other types of packages such as Microsoft Installer (MSI) files or EXE files; however, these are advanced installations. See the Microsoft document “OCSetup Command-Line Options” for more information at http://technet2.microsoft.com/WindowsVista/en/library/9a9fe5ed-5cfb-47f0-99e3-af4ef1442ee71033.mspx?mfr=true.

Installing Active Directory Domain Services

Unfortunately, with every group, there’s always got to be a one-off. Server Core is no different. Although it appears possible to install the Active Directory Domain Services (AD DS) components using ocsetup, the installation of AD DS through this mechanism is not supported. The reason for this differing installation is likely due to the unattended installation files required for properly setting up AD DS. Unlike other components installed through ocsetup, AD DS requires certain configurations to be made at the point of installation. As a result of this, installing AD DS is completed by using the DCPROMO command.

DCPROMO in Server Core is different, however. As we’ve discussed already, Server Core does not support the types of GUIs required by the standard DCPROMO installation routine. Thus, DCPROMO in Server Core has been limited to function only when used with a text-based unattended installation file. The good news is that the creation of these files has gotten much easier with Server 2008. With previous versions, the only way to create an unattended text file was by manually building the file in a text editor. With Server 2008, this process becomes significantly easier through a single button added to the last screen of the DCPROMO utility in the “full” version of Server 2008. Figure 5.4 shows this button, titled *Export Settings*.



Review your selections:

Configure this server as an additional Active Directory domain controller for the domain realtime-windowsserver.com.

Site: Default-First-Site-Name

Additional Options:
Read-only domain controller: No
Global catalog: Yes
DNS Server: Yes

Update DNS Delegation: No

Source domain controller: any writable domain controller

To change an option, click Back. To begin the operation, click Next.

These settings can be exported to an answer file for use with other unattended operations.
More about using an answer file

Figure 5.4: The Export Settings button provides an easy way to create unattended installation files for a Server Core domain controller installation.

There's a trick to using this new button. To get the unattended installation file that contains the configurations you want for your Server Core instance, first launch DCPROMO on an existing domain member server. Then answer the configuration questions in the interface as if you were answering it for how you plan to install AD DS on your Server Core instance. For example, if your Server Core instance will be a secondary domain controller, select it in the interface. If it will host Global Catalog (GC) functionality, select that as well. At the final screen (which looks similar to Figure 5.4), click *Export settings* and save the resulting text file to a location on the network. Do not hit Next at this screen; instead click Cancel to cancel the DCPROMO process.

What you have at the conclusion of this process is an unattended installation file that can be used by another computer—in this case, your Server Core instance—to configure DCPROMO. Listing 5.5 shows an example of this file in the case where a secondary domain controller is to be created for the realtime-windowsserver.com domain. Copy this file to a location that is accessible by your Server Core instance. Then run DCPROMO on the Server Core instance with the following syntax to begin the AD DS installation

```
dcpromo.exe /unattend:{unattendedTextFile}
```

```
; DCPROMO unattend file (automatically generated by dcpromo)
; Usage:
;  dcpromo.exe /unattend:C:\dcpromo.txt
;
; You may need to fill in password fields prior to using the unattend
file.
; If you leave the values for "Password" and/or "DNSDelegationPassword"
; as "*", then you will be asked for credentials at runtime.
;
[DCInstall]
; Replica DC promotion
ReplicaOrNewDomain=Replica
ReplicaDomainDNSName=realtime-windowsserver.com
SiteName=Default-First-Site-Name
InstallDNS=Yes
ConfirmGc=Yes
CreateDNSDelegation=No
UserDomain=realtime-windowsserver.com
UserName=*
Password=*
DatabasePath="C:\Windows\NTDS"
LogPath="C:\Windows\NTDS"
SYSVOLPath="C:\Windows\SYSVOL"
; Set SafeModeAdminPassword to the correct value prior to using the
unattend file
SafeModeAdminPassword=
; Run-time flags (optional)
; CriticalReplicationOnly=Yes
; RebootOnCompletion=Yes
```

Listing 5.5: A sample unattended installation file to be used to create a secondary domain controller.

Server Core + BitLocker + RODC = A Secure Branch Office

It's worth stopping at this point to talk about how an effective combination of Server Core and the RODC option secured through BitLocker can substantially improve the security of servers at branch office locations. We've already talked in Chapter 3 about the benefits of RODCs to quasi-secured locations. Different than "full" domain controllers, this new category of domain controller prevents the loss of a single domain controller from requiring the repermissioning of virtually the entire AD forest. By restricting which accounts are synchronized down to the individual domain controller, the loss of an RODC can substantially restrict the loss of accounts.

Server Core's reduced attack surface adds to this security by reducing the number of potential interfaces a would-be attacker can use to dig into the server. Additionally, its command-based management means that most management activities can be done over the WAN with little need for passing full desktops across the wire.

But when you take an RODC installed to a Server Core instance and combine it with Microsoft's BitLocker technology, this combination elevates the security of a branch office server to an entirely different beast. As we'll talk about in Chapter 9, when we focus on system security and BitLocker in more detail, BitLocker is an entire-drive encryption tool. This tool, when removed from its native network location where decryption of the drive contents can be done, ultimately turns useful data into unreadable gibberish. The encryption protocols used by BitLocker are powerful enough to prevent even the most determined hacker from discovering the true contents of the drive.

> 🕮 We'll discuss BitLocker in more detail in Chapter 9, but for now, consider this three-fold combination as a highly secure possibility for your quasi-secured branch office locations.

Other Powerful Tools for Managing Server Core

In addition to the command line, Group Policy, and RSAT tools used in managing the configuration of a Server Core instance, there are a few tools that can come in handy in certain situations: Terminal Services, PowerShell, and Windows Remote Shell.Terminal Services provides a powerful mechanism for remoting the desktop of any server. In Server Core, Terminal Services can be used to the very same ends. Utilizing Terminal Server can gain you access to the "desktop" of your Server Core instance for performing the same types of management activities you would normally do at the console. As Table 5.1 illustrated, the command to enable Remote Administration connections to a Server Core instance is

```
cscript scregedit.wsf /AR 1
```

> 💣 One caution with Terminal Services is to ensure that you enter the command *logout* to end your session rather than closing the command prompt or by simply closing the Remote Desktop Client window. It is possible to leave sessions active that will later need to be cleared out with the *reset session* command.

Windows PowerShell is a powerful scripting language that has a somewhat odd relationship with Server Core. Although PowerShell commands cannot directly run on a Server Core instance, they can be used on a management workstation to manage elements of a Server Core instance. PowerShell works with elements such as the Windows Management Instrumentation (WMI) and other management databases and methods that are exposed for remote access.

Windows Remote Shell is a third powerful tool that brings the command line of a Server Core instance directly to the command prompt of a management workstation. Think of Windows Remote Shell as a variant of the highly popular Sysinternals tool *psexec.exe*. Windows Remote Shell is a component of Windows Remote Management and can be used on a management workstation such as a Vista box or another Server 2008 instance to locally launch the Server Core's command prompt.

To do so, you'll first need to enable Windows Remote Management on your Server Core instance by running

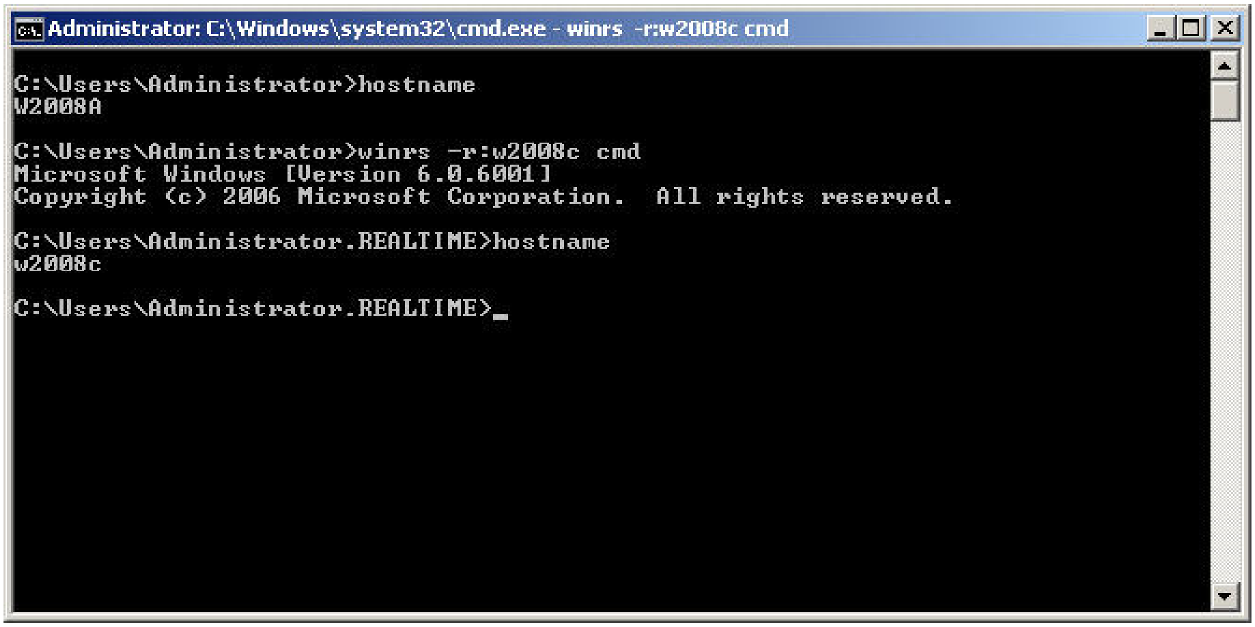
```
winrm quickconfig
```

from the console of the Server Core instance. This will complete the initial configuration of Windows Remote Management to enable it for use. This initial configuration includes starting the Windows Remote Management service, creating a listener, and opening the correct firewall ports to enable its functionality.

Once this is complete, from your management workstation, run the command

```
winrs -r:{computerName} cmd
```

As Figure 5.5 shows, running this command changes the local command prompt on the management workstation to operate as if it were really local on the remote Server Core instance. Notice how the result from the hostname command changes after launching the remote shell. This is the case because the shell being operated against is actually on the Server Core instance named w2008c.


Administrator: C:\Windows\system32\cmd.exe - winrs -r:w2008c cmd

```
C:\Users\Administrator>hostname
W2008A

C:\Users\Administrator>winrs -r:w2008c cmd
Microsoft Windows [Version 6.0.6001]
Copyright (c) 2006 Microsoft Corporation.  All rights reserved.

C:\Users\Administrator.REALTIME>hostname
w2008c

C:\Users\Administrator.REALTIME>_
```

Figure 5.5: Windows Remote Shell lets you remotely launch the command prompt from your local management workstation.

Server Core Command-Line Crib Sheet

Lastly, to give you a summary of the commands that are available in Windows Server core for console management, take a look at Table 5.2, which includes a short crib sheet of available options. This table gives you a single-glimpse cheat sheet for assisting you with the process of configuring and administering your Server Core instance. Each of these can also be viewed within a Server Core instance by navigating to the C:\Windows\system32 folder and entering the command

```
cscript scregedit.wsf /CLI
```

Some are duplicates with what we've already discussed in this chapter.

Command	Explanation
cscript slmgr.vbs -ato	Activate a license
cscript slmgr.vbs -ipk {volume license key}	Configure KMS volume licensing
cscript slmgr.vbs -ato	Activate KMS licensing
cscript slmgr.vbs -skma {KMS FQDN}	Set KMS DNS SRV Record
set c, ipconfig /all, systeminfo.exe, & hostname.exe	Determine the computer name
netdom renamecomputer %computername% /NewName:new-name /UserD:domain-username /PasswordD:*	Rename the computer while already domain-attached
netdom renamecomputer %computername% /NewName:new-name	Rename the computer while not domain-attached
wmic computersystem where name="%computername%" call joindomainorworkgroup name="{new workgroup name}"	Change workgroups
start /w ocsetup [packagename]	Install a Role, Role Service, or Feature
oclist	View available and installed Roles, Role Services, or Features
Control-Shift-ESC	Start Task Manager
logoff	Logoff a Terminal Services session
wmic computersystem where name="%computername%" set AutomaticManagedPagefile=False	Disable system pagefile management
wmic pagefileset where name="C:\\pagefile.sys" set InitialSize=500,MaximumSize=1000	Configure the pagefile
control timedate.cpl	Control the timezone, date, or time.
control intl.cpl	Control regional and language options.
msiexec.exe /i {msiPackage}	Manually install an .MSI application
wmic product	List installed MSI applications
wmic product get name /value wmic product where name="{name}" call uninstall	Uninstall MSI applications
Sc query type= driver	List installed drivers
Pnputil -i -a {path}\{driver}.inf	Install a driver. Driver files must first be copied to the Server Core instance.

Command	Explanation
netsh interface set interface name="Local Area Connection" newname="PrivateNetwork"	Rename a network adapter
netsh interface set interface name="Local Area Connection 2" admin=DISABLED	Disable a network adapter
wmic datafile where name="c:\\windows\\system32\\ntdll.dll" get version	Determine a file's location
wmic qfe list	List installed patches
wusa.exe [patchame].msu /quiet	Install a patch
netsh winhttp set proxy {proxy_name}:{port}	Configure a proxy
reg.exe add /?, reg.exe delete /?, reg.exe query /?	Add, delete, or query a registry value.

Table 5.2: Server Core command-line crib sheet.

A Compelling "New and Different Way" for Windows Server 2008

So we've seen in this chapter that there really is a "new and different way" to manage Windows Server. With Server 2008 and Server Core, we can devolve our long-held GUI-based management in certain circumstances to the elegance and easy automation found in other command-driven OSs. Though Server Core does indeed have a learning curve if you're not already familiar with the command-line adjuncts to common GUI configuration tools, the process of learning them will directly augment your abilities to manage other "full" Server 2008 instances. Server Core is a compelling installation option for Server 2008 for which specific organizations should find excellent positioning for certain uses within their environments.

Though it was glossed over in this chapter, Server Core instances also follow all the rules (that they're capable of) associated with Group Policy. So, if you have Group Policies that relate to a Server Core instance, they will configure themselves per those settings as well. In our next chapter, we'll talk about Windows Group Policy and how you can use them to manage, control, lock down, and ultimately secure your computing environment. We'll talk about how the Group Policy Central Store improves its replication, how Group Policy Preferences bring item-level control to your management quiver, and a host of other topics that I think you'll find useful in managing your Windows Server 2008 infrastructure.

Download Additional eBooks from Realtime Nexus!