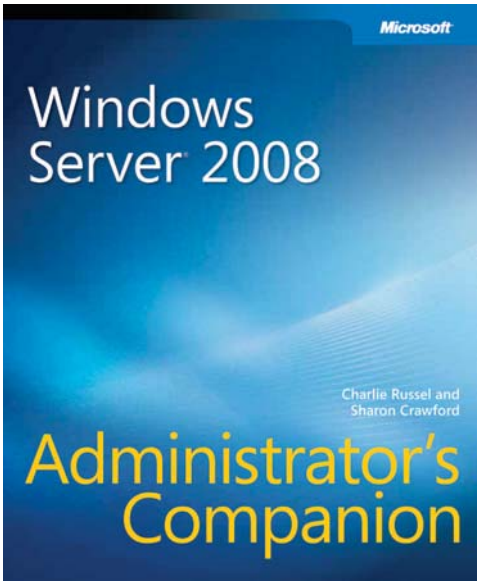

Windows® Server 2008 Administrator's Companion

Charlie Russel and Sharon Crawford

To learn more about this book, visit Microsoft Learning at
http://www.microsoft.com/MSPress/books/11840.aspx

978-0-7356-2505-1

Table of Contents

21 Working with Failover Clusters

- Defining Clusters
- Cluster Scenarios
  - Internet or Intranet Functionality
  - Terminal Services
  - Mission-Critical Availability
- Requirements and Planning
  - Identifying Goals and Risks
  - Identifying a Solution
- Network Load Balancing Clusters
  - Choosing an NLB Cluster Model
  - Creating an NLB Cluster
  - Planning Capacity
  - Providing Fault Tolerance
  - Optimizing
- Server Clusters
  - Concepts
  - Types of Resources
  - Defining Failover and Failback
  - Planning Capacity
  - Creating a Server Cluster
- Shared Folders

Part IV Secure the Network

22 Planning Security

- Security Basics
  - Authentication
  - Access Control
  - Auditing
- Smart Cards
- Public Key Infrastructures (PKI)
  - Public keys & private keys
  - Certificates
- Security Enabled Protocols
  - Secure Internet Mail Extensions
  - Kerberos 5
  - Secure Socket Layer
  - Internet Protocol Security
- Virtual Private Networks
- Remote Access VPNs
- Router-to-Router VPNs
- Security Modules

Chapter 9

Installing and Configuring Server Core

The usual progression for an operating system (or an application, for that matter) is to grow and add features, sometimes well beyond what any of us want or need. Windows Server 2008 reverses that trend with a completely new installation option—Server Core. When you install Windows Server 2008, regardless of which edition you're installing, you have the option of choosing a full installation, with everything, or just the Server Core portion.

Server Core is just the essentials, with little or no graphical interface. The logon provider has the same graphical look, but then, when you've logged in, all you see is a single command-shell window, as shown in Figure 9-1.

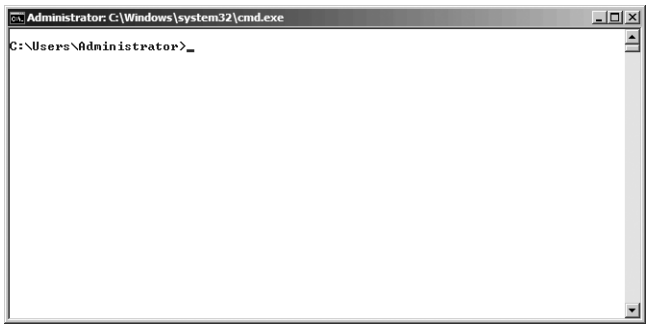

Figure 9-1 The Windows Server 2008 Core desktop.

Note For improved readability in screen shots used here and in the rest of the book, I've changed the default color scheme for Command Prompt windows to dark blue text on a white background.

Benefits of a Server Core Installation

All Windows Server 2008 editions support Server Core, with the exception of Compute Cluster Edition. And installing Server Core doesn't give you a break on the cost of the license—it's exactly the same license and media as the full Windows Server 2008 installation. At install time, you simply choose which edition you are installing. So, if you don't save any money, and you don't have special media, and you have reduced functionality, why in the world would you choose Server Core over the full product? It's simple, really: security and resources. Let's take a look at those two in a bit more depth before we go on to the details of how to actually install and configure Server Core.

Security

In the old days, whenever you installed Windows Server, it automatically installed just about everything that was available, and turned on all the services that you were likely to need. The goal was to make installation as simple as possible, and this seemed like a good idea at the time. Sadly, the world is not a friendly place for computers any more, and that approach is no longer safe or wise. The more services that exist, and the more services that are enabled, the more attack vectors the bad guys have to work with. To improve security, limiting the available attack surfaces is just good common sense.

In Server Core, Microsoft has completely removed all managed code, and the entire .NET Framework. This leaves a whole lot fewer places for possible attack. This does, obviously, impose some severe limits on what you can and can't do with a Server Core installation. And it also means that there isn't any PowerShell possible, which in our opinion is easily the biggest limitation of Server Core—but one that we hope will be resolved in a later version of Windows Server.

The default installation of Server Core has only 38 services running. A typical full Windows Server 2008 installation, with one or two roles enabled, is likely to have 60 or even 70 or more services running. Not only does the reduced number of services limit the potential attack surface that must be protected, but it also limits the number of updates that are likely to be required over the life of the server, making it easier to maintain.

Resources

The second major benefit to running Server Core is the reduced resources required for the base operating system. While the official requirements for installing Windows Server 2008 are the same for Core as for a full installation, the effective numbers are significantly less, in our experience—with the exception of the disk space required (only 2–3 GB of HD space for a running Core installation). Plus with the limited subset of tasks that you can perform, we think Server Core is ideal for running those infrastructure tasks that everyone runs, and that doesn't require much interaction over time. Tasks such as DHCP, DNS, and increasingly Virtualization. Now if it just had PowerShell.

Installing Server Core

Installing Windows Server 2008 Server Core is ultimately the same as installing the full graphical version of Windows Server 2008. The installation engine is the same, and the only difference occurs during the install, when you have to choose which version of Windows Server 2008 to install, as shown in Figure 9-2.

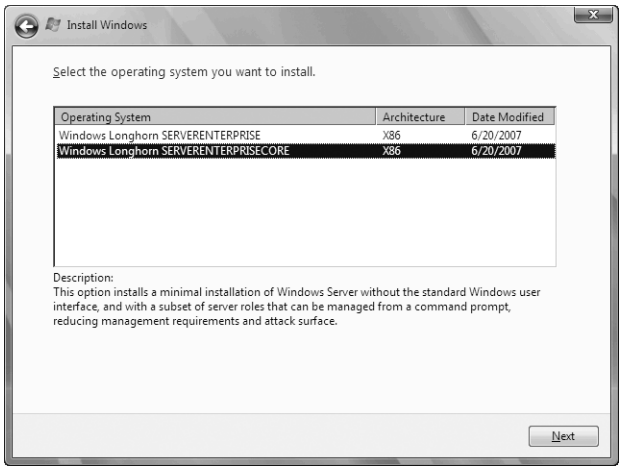

Figure 9-2 During initial installation, you make an irrevocable choice between Server and Server Core.

Once installation completes, you're presented with the initial logon screen. Log on as Administrator, with no password, and you'll be immediately prompted to change the password and then logged on to the desktop, as shown earlier in Figure 9-1. All initial configuration takes place from the command line, though once you've configured the basics, you'll be able to use familiar management consoles remotely.

You can use an unattend.xml file to automate the initial install and configuration of your Server Core installation. For details on the settings and syntax of unattend.xml, see *http://go.microsoft.com/fwlink/?LinkId=81030*.

Configuration

All configuration tasks for Server Core can be done at the command line, and all the initial tasks have to be done at either the command line or as part of the installation process by using an unattend.xml script. Once you're performed these initial configuration tasks, you can then use regular Windows management consoles to manage the additional settings. Unfortunately, there isn't a single command shell for the tasks, but a collection of old favorites, each with a different behavior and syntax.

Initial Configuration

The initial steps you'll need to perform on a Server Core installation will depend somewhat on your intended use of the installation, but we think that the following ones are the most obvious:

- Set a fixed IP address.
- Change the server name to match your internal standards.
- Join the server to a domain.
- Change the default resolution of the console.
- Enable remote management through Windows Firewall.

- Enable remote desktop.
- Activate the server.

We'll walk through these steps for you, and leave you with a couple of basic scripts that you can modify to automate these tasks for your environment. Table 9-1 contains the settings we'll be using during this install scenario.

Table 9-1 Settings for Initial Server Core Configuration (Example)

Setting	Value
IP Address	192.168.51.4
Gateway	192.168.51.1
DNS Server	192.168.51.2
Server Name	Hp350-core-04
Domain To Join	example.local
Default Desktop Resolution	1024x768
Remote Management	Enable for Domain Profile
Windows Activation	Activate

Set IP Address

To set the IP address for the server, you need to use the netsh command-line tool. Follow these steps to configure TCP/IP:

1. From the command window, use netsh to get the "name" (index number) of the network card.

```
netsh interface ipv4 show interfaces
```

2. The result will be something like the following:

```
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
```

3. The *Idx* value for your real network card (2, in this case) will be used as the name value in future commands for netsh.

4. Now, using the *Idx* value from step 2, run the following netsh command:

   ```
   netsh interface ipv4 set address name="<Idx>" source=static address=<IP Address>
   mask=<netmask> gateway=<IP Address of default gateway>
   ```

5. Next, specify the DNS server for the adapter, using netsh again:

   ```
   netsh interface ipv4 add dnsserver name="<Idx>" address=<IP Address of DNS Server>
   index=1
   ```

6. For secondary DNS servers, repeat the command in step 5, increasing the index value by one each time.

Renaming the Server and Joining to a Domain

The next step in initial configuration is assigning the name of the server and joining it to a domain. During initial installation of Windows Server 2008, an automatically generated name is assigned to the server and it is placed in the WORKGROUP workgroup. You'll want to change this to align the computer name with your corporate naming policy and join the server to the correct domain and Organizational Unit. Our naming policy here has three parts: the model of server, the functional role, and a number reflecting its IP address. Thus the Server Core computer we're building in this chapter is named hp350-core-04: it's a Hewlett Packard ML 350 G5 server, it is running Server Core, and the final octet of its IP address is four. Your server naming convention will undoubtedly be different, but the important thing is to be consistent. Our domain for this book is example.local.

To change the name of the server and join it to the example.local domain, follow these steps:

1. From the command prompt, use the netdom command to change the name of the server:

   ```
   netdom renamecomputer %COMPUTERNAME% /newname:<newname>
   ```

2. After you change the name, you must reboot the server.

   ```
   shutdown /t 0 /r
   ```

3. After the server restarts, log on to the Administrator account.

4. Use the netdom command again to join the domain.

   ```
   netdom join %COMPUTERNAME% /DOMAIN:<domainname> /userd:<domain admin account>
   /password:*
   ```

5. You'll be prompted for the password for the domain administrative account you used. Enter the password. When the domain join has succeeded, you'll again need to reboot the server.

   ```
   shutdown /t 0 /r
   ```

6. After the server restarts, log back on to a domain administrator's account. (You'll need to click Change User because the server will default to the local administrator account.)

Scripting Initial Configuration

If you set up more than one or two Server Core computers, you'll quickly get tired of doing all this interactively from the command prompt. We know we did. You have the choice of either using an unattend.xml file to set options during the install or

using simple scripts to automate the process. Both work, and both have their adherents, but we tend to use scripts after the fact. You can modify the following three scripts (which you'll also find on the companion CD) for your environment to automate the initial TCP/IP, server name, and domain join steps.

The first script sets the IP address, sets the DNS server, and changes the server name.

```
echo off

REM filename: initsetup1.cmd

REM

REM initial setup for a Server 2008 Server Core installation.

REM command file 1 of 3

REM

REM Created: 4 September, 2007

REM ModHist: 5/9/07 - switched to variables (cpr)

REM

REM Copyright 2007 Charlie Russel and Sharon Crawford. All rights reserved.

REM   You may freely use this script in your own environment, modifying it

REM   to meet your needs. But you may not re-publish it without permission.



REM first, set a fixed IP address. You'll need to know the index number

REM of the interface you're setting, but in a default Server Core install,

REM with only a single NIC, the index should be 2. To find the index,

REM you can run:

REM      netsh interface ipv4 show interfaces

REM
```

```
SETLOCAL

REM Change the values below to match your needs

SET IPADD=192.168.51.4

SET IPMASK=255.255.255.0

SET IPGW=192.168.51.1

SET DNS1=192.168.51.2

SET NEWNAME=hp350-core-04



netsh interface ipv4 set address name="2" source=static address=%IPADD%
mask=%IPMASK% gateway=%IPGW%



REM Next, set DNS to point to DNS server for example.local.

REM   192.168.51.2 in this case

netsh interface ipv4 add dnsserver name="2" address=%DNS1% index=1



REM Now, we need to change the computer name. After we're done, the server

REM must be restarted, and we can continue with the next batch of commands.

REM we use the /force command here to avoid prompts

netdom renamecomputer %COMPUTERNAME% /newname:%NEWNAME% /force



@echo If everything looks OK, the it's time to reboot

pause

REM now, shutdown and reboot. No need to wait.

shutdown /t 0 /r
```

The second script we use is to actually join the server to the domain.

```
@echo off

REM Filename: initsetup2.cmd

REM

REM initial setup for a Server 2008 Server Core installation.

REM command file 2 of 3

REM

REM Created: 4 September, 2007

REM ModHist:

REM

REM Copyright 2007 Charlie Russel and Sharon Crawford. All rights reserved.

REM   You may freely use this script in your own environment, modifying it

REM   to meet your needs. But you may not re-publish it without permission.



SETLOCAL

SET DOMAIN=example.local

SET DOMADMIN=Administrator



REM Join the domain using the netdom join command. Prompts for password

REM of domain administrator account set above



netdom join %COMPUTERNAME% /DOMAIN:%DOMAIN% /userd:%DOMADMIN% /passwordd:*
```

```
REM now, shutdown and reboot. No need to wait, and that's all we can do

REM at this time



shutdown /t 0 /r
```

Finally, use the third script to enable remote management and activate the server.

```
echo off

REM initsetup3.cmd

REM

REM initial setup for a Server 2008 Server Core installation.

REM command file 3 of 3

REM

REM Created: 4 September, 2007

REM ModHist:

REM

REM Copyright 2007 Charlie Russel and Sharon Crawford. All rights reserved.

REM   You may freely use this script in your own environment, modifying it

REM   to meet your needs. But you may not re-publish it without permission.



REM Use netsh to enable remote management through the firewall for the

REM domain profile. This is the minimum to allow using remote MMCs to work

REM from other computers in the domain.



netsh advfirewall set domainprofile settings remotemanagement enable
```

```
REM allow remote administration group
netsh advfirewall firewall set rule group="Remote Administration" new enable=yes



REM Allow remote desktop

REM (also works with group=”Remote Desktop” instead of name=)

netsh advfirewall firewall set rule name="Remote Desktop (TCP-In)" new enable=yes



REM Enable Remote Desktop for Administration, and allow

REM downlevel clients to connect

cscript %windir%\system32\scregedit.wsf /AR 0

cscript %windir%\system32\scregedit.wsf /CS 0



REM Now, run the activation script
REM No output means it worked

Slmgr.vbs -ato
```

Setting Desktop Display Resolution

To set the display resolution for the Server Core desktop, you need to manually edit the registry. We'd give you a script to do it, but it is dependent on correctly identifying the specific GUID for your display adapter. Not something we want to automate. So, to change the resolution on your Server Core desktop, follow these steps:

1. Open regedit.
2. Navigate to HKLM\System\CurrentControlSet\Control\Video.
3. One or more GUIDs is listed under Video. Select the one that corresponds to your video card. Hint: they each have a device description under the 0000 key that can sometimes help.
4. Under the GUID select the 0000 key, and add a DWORD DefaultSettings.XResolution. Edit the value to the X axis resolution you want. For a width of 1024 pixels, use 400 hexadecimal, as shown in Figure 9-3.

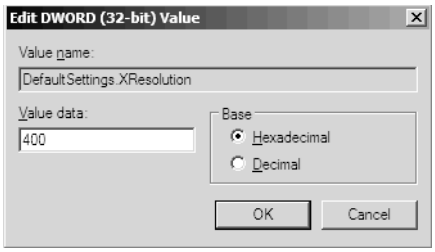

Figure 9-3 Editing the display resolution value for the X axis.

5. Add a DWORD DefaultSettings.YResolution. For height of 768 pixels, use 300 hexadecimal.

> **Note** In some cases, these keys will already exist. If they do, you can simply change their value as necessary.

6. Exit the registry editor and log off using the following:

```
shutdown /l
```

7. Once you log back on, the new display settings will take effect.

Enabling Remote Management

To allow access to the familiar graphical administration tools, you need to enable them to work through Windows Firewall. This requires another set of netsh commands. Use the following steps to enable remote administration and Remote Desktop:

1. From the command prompt, use the netsh command to enable remote management:

```
netsh advfirewall set domainprofile settings remotemanagement enable
```

2. Now, enable the Remote Administration group of firewall rules.

```
netsh advfirewall firewall set rule group="Remote Administration" new enable=yes
```

3. Finally, life is easier when you can connect using remote desktop, so let's enable that, too:

```
netsh advfirewall firewall set rule name="Remote Desktop (TCP-In)" new enable=yes
```

You should now be able to do additional management using familiar graphical tools from another server but connecting to the Server Core computer.

Activating the Server

The final step in basic configuration of the Server Core computer is to activate it. This requires using a Visual Basic script, which is provided. Use the following command:

```
Slmgr.vbs -ato
```

> **Note** All the basic initial setup commands for Server Core are included in the three scripts described in the UnderTheHood sidebar, and are also available on the CD that comes with the book.

Installing Roles

Windows Server 2008 Core doesn't support all the possible roles and features of the full graphical Windows Server, but it does support the most important infrastructure roles. We think one of the most compelling scenarios for Server Core is as a remote site server to enable basic functionality at a remote site where nobody is available on site to administer it. By combining the DHCP Server, DNS Server , File Services, and Print Services roles with a read-only Active Directory Domain Services role, you have a "branch office in a box" solution—just add a remote access device such as a VPN router and you're in business.

The File Services role is added by default as part of the base Server Core installation, but you can add additional role services to support additional functionality.

The command used to install a role in Server Core is Ocsetup.exe. The exact same command is used to uninstall a role, but with the */uninstall* command-line parameter. The full syntax for Ocsetup is:

```
Ocsetup </?|/h|/help>

Ocsetup <component> [/uninstall][/passive][/unattendfile:<file>]
[/quiet][/log:<file>][/norestart][/x:<parameters>]
```

The important thing to remember about Ocsetup is that it is quite unforgiving. It is case sensitive, and even a slight mistake in the case of the component name will cause the command to fail.

A script to install the roles for this solution, except the domain controller role, would look like this:

```
@REM filename: SetupBranch.cmd

@REM

@REM Setup file to install roles for a branch office server

@REM

@REM Created: 5 September, 2007

@REM ModHist:

@REM

@REM Copyright 2007 Charlie Russel and Sharon Crawford. All rights reserved

@REM   You may freely use this script in your own environment,

@REM   modifying it to meet your needs.

@REM   But you may not re-publish it without permission.
```

```
@REM Using "start /w" with ocsetup forces ocsetup to wait until it completes before

@REM going on to the next task.



@REM Install DNS and DHCP

@echo Installing DNS and DHCP roles...

start /w ocsetup DNS-Server-Core-Role

start /w ocsetup DHCPServerCore



@REM Now, install File Role Services

@echo Now installing File Role Services...

start /w ocsetup FRS-Infrastructure

start /w ocsetup DFSN-Server

start /w ocsetup DFSR-Infrastructure-ServerEdition



@REM Uncomment these two lines to add NFS support

@REM start /w ocsetup ServerForNFS-Base

@REM start /w ocsetup ClientForNFS-Base



@REM Install Print Server Role

@echo Installing Print Server Role

start /w ocsetup Printing-ServerCore-Role
```

```
@REM Uncomment next for LPD support

@REM start /w ocsetup Printing-LPDPrintService
```

Note You can't include the DCPromo command in the script above because installing the Print Server role requires a reboot, and locks out DCPromo.

You cannot use DCPromo interactively to create a domain controller – you must create an unattend.txt file to use with it. The basic minimum unattend.txt file is:

```
[DCInstall]

InstallDNS = Yes

ConfirmGC = yes

CriticalReplicationOnly = No

RebootOnCompletion = No

ReplicationSourceDC = hp350-dc-02.example.local

ParentDomainDNSName = example.local

ReplicaOrNewDomain = ReadOnlyReplica

ReplicaDomainDNSName = example.local

SiteName=Default-First-Site-Name

SafeModeAdminPassword = <passwd>

UserDomain = example

UserName = Administrator

Password = <passwd>
```

Important The passwords fields must be correct, and will be automatically stripped from the file for security reasons. For Server Core, you must specify a *ReplicationSourceDC* value. You should set *ReplicaOrNewDomain* to the value shown here and *ReadOnlyReplica* to create a read-only domain controller.

To install the read-only Domain Controller role, follow these steps:

1. Use Notepad or your favorite ASCII text editor (we use GVim, which works quite well in Server Core) to create an unattend.txt file with the necessary settings for the domain you will be joining. The specific filename of the unattend file is not important because you specify it on the command line.
2. Change to the directory that contains the unattend file. If the server has any pending restarts, you *must* complete them before promoting the server to domain controller.
3. Run DCPromo with the following syntax:

   ```
   Dcpromo /unattend:<unattendfilename>
   ```

4. If there are no errors in the unattend file, DCPromo will proceed and promote the server to be a read-only domain controller, as shown in Figure 9-4.

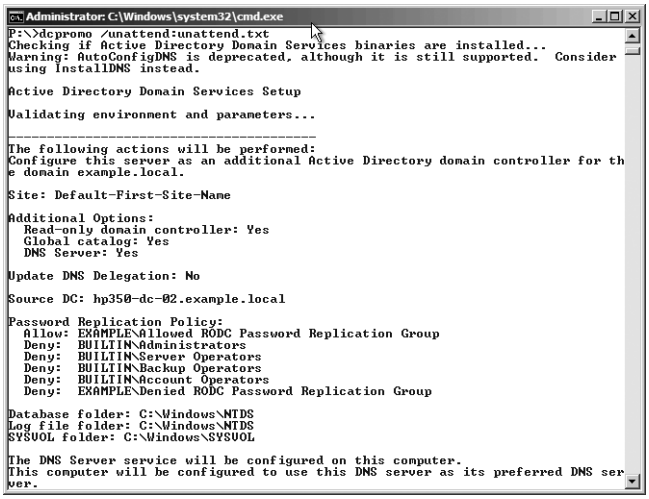

Figure 9-4 Use DCPromo to create a read-only domain controller with an unattend file.

Listing Roles

The Oclist.exe command provides a complete list of the available Server Core roles, role services, and features, as well as their current state. Use Oclist to get the exact, case-sensitive list of the features and roles you want to install.

Managing a Server Core Computer

Managing a Server Core computer is a different experience for most system administrators. None of the graphical tools you're used to using is available *on the server*. But once you've configured the Server Core computer for remote management, as described under "Initial Configuration" earlier in the chapter, you can create management consoles that point to the Server Core computer, which allow you to do all your tasks from a graphical console.

> **More Info** For details on how to create custom MMCs, see Chapter 14, "Managing Daily Operations."

There are four basic ways to manage a Server Core installation. They are:

- Locally using a command prompt.
- Remotely, using Remote Desktop. The shell in Remote Desktop will have only the same functionality (a command prompt) as being logged on locally.
- Remotely using Windows Remote Shell.
- Remotely using an MMC snap-in from a server running Windows Vista or Windows Server 2008.

Some tasks are a bit tricky in Server Core—we're used to usually doing them exclusively from the GUI. An obvious task is changing the password on your account. For that, use the **net user <username> *** command. Some of the tasks that can be a problem. Table 9-2 shows some solutions.

Table 9-2 Common Task Workarounds in Server Core

Task	Solution/Workaround
Enable automatic updates	`Cscript %windir%\system32\scregedit.wsf /AU [value]` Where values are: 1 – disable automatic updates 4 – enable automatic updates /v – view current setting
Enable Remote Desktop for Administrators	`Cscript %windir%\system32\scregedit.wsf /AR [value]` Where values are: 0 – enable Remote Desktop 1 – disable Remote Desktop /v – view current setting
Enable Terminal Server clients from Windows versions prior to Windows Vista	`Cscript %windir%\system32\scregedit.wsf /CS [value]` Where values are: 0 – enable prior versions 1 – disable prior versions /v – view current setting
Allow IPSec Monitor remote management	`Cscript %windir%\system32\scregedit.wsf /IM [value]` Where values are: 0 – disable remote management 1 – enable remote management /v – view current setting

Configure DNS SRV record weight and priority	`Cscript %windir%\system32\scregedit.wsf /DP [value]`
	Where DNS SRV priority values are:
	0-65535. (Recommended value = 200)
	/v – view current setting
	`Cscript %windir%\system32\scregedit.wsf /DW [value]`
	Where DNS SRV weight values are:
	0-65535. (Recommended value = 50)
	/v – view current setting

Update User passwords	Net user <username> [/domain] *
Installing .msi files	Use the /q or /qb switches from the command line with the full .msi filename. /q is quiet; /qb is quiet but with a basic user interface
Changing the time zone, date, or time	timedate.cpl
Change internationalization settings	intl.cpl
Using Disk Management console	From the command line of the Server Core installation: Net start VDS Then run Disk Management remotely.
Get Windows version information	Winver is not available. Use systeminfo.exe instead.
Get Help (regular Windows Help and Support files are not viewable in Server Core)	`Cscript %windir%\system32\scregedit.wsf /cli`

Using Windows Remote Shell

You can use Windows Remote Shell to remotely execute commands on a Server Core computer. But before you can run Windows Remote Shell, you need to first enable it on the target Server Core computer. To enable Windows Remote Shell, use the following command:

```
WinRM quickconfig
```

To run a command remotely, use the WinRS command from another computer using the following command:

```
Winrs –r:<ServerName> <command string to execute>
```

Using Terminal Server RemoteApp

One neat trick that we like is to use the new TS RemoteApp functionality of Windows Server 2008 to publish a Command Prompt window for the Server Core computer directly

onto our desktop. This is simpler and more direct, and saves screen real estate, which is always a benefit. To create an RDP package that you can put on your desktop, follow these steps:

1. On a Windows Server 2008 server that has the Terminal Services role enabled, open the TS RemoteApp Manager, as shown in Figure 9-5.

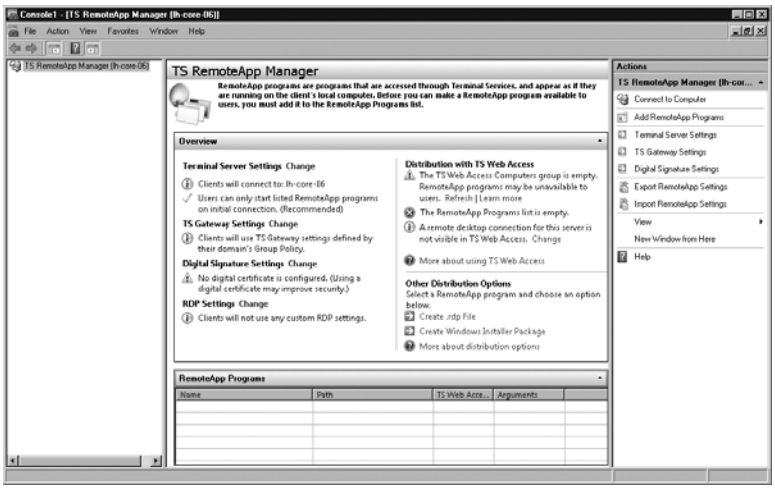

Figure 9-5 Use the TS RemoteApp Manager to create a remote cmd.exe window.

2. Connect to the Server Core computer you want to build an RDP package for.
3. Click Add RemoteApp Programs in the actions pane to open the RemoteApp Wizard.
4. Click Next to open the Choose Programs To Add To The RemoteApp Programs List page, shown in Figure 9-6.

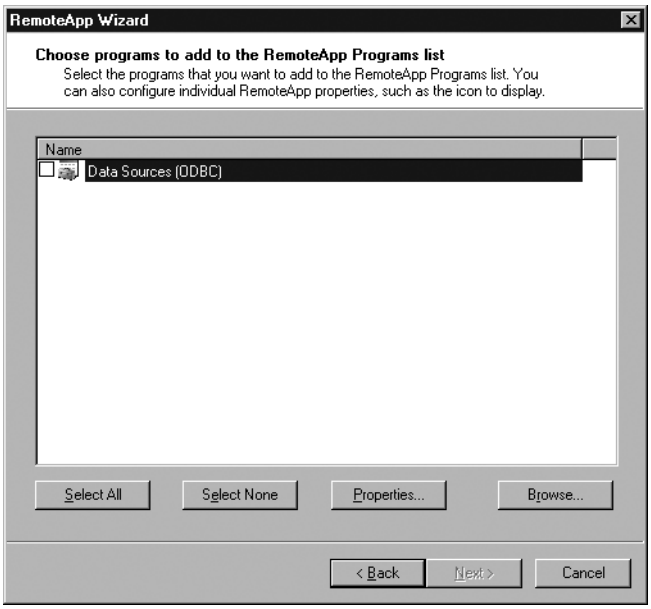

Figure 9-6 The Choose Programs To Add To The RemoteApp Programs List page of the RemoteApp Wizard.

5. Click Browse, and navigate to \<ServerName>\c$\windows\system32\cmd.exe. Click Open.
6. Click Next and then click Finish to add the remote program and return to the TS RemoteApp Manager.
7. Select cmd.exe in the RemoteApp programs pane and click Create .rdp File in the actions pane.
8. Click Next, and specify any additional package settings for the RDP package. Note the location where the package will be saved.
9. Click Next and then click Finish to create the RDP package.
10. Copy the package to the computer where you will use it.

Now you can open a Command Prompt window directly onto the Server Core computer simply by double-clicking the RDP package you created and saved.

Summary

In this chapter we've covered some basic steps for setting up and configuring the new Server Core installation option of Windows Server 2008. We think this is an exciting and powerful new way to get the power of Windows Server while maintaining very high levels of security and ease of management. And yes, we know that sounds a bit like marketing hype, but we actually think that Server Core is an important step forward.

In the next chapter, we'll cover managing and configuring your printers using the Printer Management console.

Chapter 19
Implementing Disk Management

Servers are used for many functions and have many reasons for existence, but the single most pervasive function of most servers is storage. And you can't store anything if you don't have something to store it on. For servers, that something is primarily hard disks. Rather than cover all topics related to storage in a single chapter, we've split it up a bit. Both for reasons of length (our editors have this irrational fear of 100+ pages chapters) and also to group topics together rationally.

In this chapter, we'll start by defining some terms that we'll use throughout our discussions of storage. Once we've got that basic ground covered, we'll move on to the physical aspects of storage—the disk subsystem and how you manage and administer it. This includes disks, partitions, and volumes, along with logical drives. And we'll cover special features of the NTFS file system, including encryption and quotas. Throughout this chapter we'll cover both the graphical way to do things and the command-line way.

In Chapter 20 "Managing Storage," we'll shift gears and talk about storage from a logical perspective, with full coverage of the Storage Resource Manager, and we'll also cover Storage Area Networks (SANs)—a way to centralize and abstract storage for a group of servers.

The hard disk management functions of Windows Server 2008 build on earlier versions of Windows Server to make hard disk management flexible and easy for administrators while hiding the complexities from end users. One important—and long overdue—new feature is the ability to grow or shrink partitions dynamically without losing data.

Understanding Disk Terminology

Before going into the details of managing disks and storage, let's review some definitions:

Physical drive
- The actual hard disk itself, including the case, electronics, platters, and all that stuff. This is not terribly important to the disk administrator.

Partition
- A portion of the hard disk. In many cases, this is the entire hard disk space, but it needn't be.

Allocation unit
- The smallest unit of managed disk space on a hard disk or logical volume. It's also called a *cluster*.

Primary partition
- A portion of the hard disk that's been marked as a potentially bootable logical drive by an operating system. MS-DOS can support only a single primary partition, but Windows Server 2008 can support multiple ones. There can be only four primary partitions on any hard disk.

Extended partition

- A nonbootable portion of the hard disk that can be subdivided into logical drives. There can be only a single extended partition per hard disk, but it can be divided into multiple logical drives.

Extended volume

- Similar to, and sometimes synonymous with, a spanned volume. This is any dynamic volume that has been extended to make it larger than its original size. When an extended volume uses portions of more than one physical disk, it is more properly referred to as a spanned volume.

Logical drive

- A section or partition of a hard disk that acts as a single unit. An extended partition can be divided, for example, into multiple logical drives.

Logical volume

- Another name for a logical drive.

Basic disk

- A traditional disk drive that is divided into one or more partitions, with a logical drive in the primary partition, if present, and one or more logical drives in any extended partitions. Basic disks do not support the more advanced functions of Disk Management, but they can be converted to dynamic disks in many cases.

Dynamic disk

- A managed hard disk that can be used to create various volumes.

Volume

- A unit of disk space composed of one or more sections of one or more disks. Prior versions of Windows Server used volume only when referring to dynamic disks, but Windows Server 2008 uses it to mean partitions as well.

Simple volume

- Used interchangeably with partition in Windows Server 2008, earlier versions of Windows used simple volume only when referring to a dynamic disk. A portion of a single disk, a simple volume can be assigned either a single drive letter or no drive letter and can be attached (mounted) on zero or more mount points.

RAID (redundant array of independent [formerly "inexpensive"] disks)

- The use of multiple hard disks in an array to provide for larger volume size, fault tolerance, and increased performance. RAID comes in different levels, such as RAID-0, RAID-1, RAID-5, and so forth. Higher numbers don't necessarily indicate greater performance or fault tolerance, just different methods of doing the job.

Spanned volume

- A collection of portions of hard disks combined into a single addressable unit. A spanned volume is formatted like a single drive and can have a drive letter assigned to it, but it will span multiple physical drives. A spanned volume—occasionally referred to as an extended volume—provides no fault tolerance and increases your exposure to failure, but does permit you to make more efficient use of the available hard disk space.

Striped volume

- Like a spanned volume, a striped volume combines multiple hard disk portions into a single entity. A striped volume uses special formatting to write to each of the portions equally in a stripe to increase performance. A striped volume provides no fault tolerance and actually increases your exposure to failure, but it is faster than either a spanned volume or a single drive. A stripe set is often referred to as RAID-0, although this is a misnomer because plain striping includes no redundancy.

Mirror volume

- A pair of dynamic volumes that contain identical data and appear to the world as a single entity. Disk mirroring can use two drives on the same hard disk controller or use separate controllers, in which case it is sometimes referred to as *duplexing*. In case of failure on the part of either drive, the other hard disk can be split off so that it continues to provide complete access to the data stored on the drive, providing a high degree of fault tolerance. This technique is called RAID-1.

RAID-5 volume

- Like a striped volume, a RAID-5 volume combines portions of multiple hard disks into a single entity with data written across all portions equally. However, it also writes parity information for each stripe onto a different portion, providing the ability to recover in the case of a single drive failure. A RAID-5 volume provides excellent throughput for read operations, but it is substantially slower than all other available options for write operations.

SLED (single large expensive disk)

- Now rarely used, this strategy is the opposite of the RAID strategy. Rather than using several inexpensive hard disks and providing fault tolerance through redundancy, you buy the best hard disk you can and bet your entire network on it. If this doesn't sound like a good idea to you, you're right. It's not.

JBOD

- Just a bunch of disks. The hardware equivalent of a spanned volume, this has all the failings of any spanning scheme. The failure of any one disk will result in catastrophic data failure.

More Info Additional RAID levels are supported by many hardware manufacturers of RAID controllers. These include RAID 0+1, RAID 10, RAID 6, and RAID 50. For more details on various RAID levels, see the manufacturer of your RAID controller or *http://en.wikipedia.org/wiki/RAID#Standard_RAID_levels*.

Disk Technologies for the Server

The first time we wrote a chapter about disk management, basically three possible technologies were available: Modified Field Modification (MFM), Pulse Frequency Modulation (PFM), and Small Computer System (or Serial) Interface (SCSI). Unless you were a total geek (and had oodles of money), your systems used either MFM or PFM, and RAID wasn't even an option. Over time, SCSI became the only real choice for the vast majority of servers and even became mainstream on high-end workstations. Servers at the high end might use fiber, but SCSI had the vast majority of the server disk market. SCSI has changed over the years to support faster speeds,

more disks, and greater ease of configuration and use, but is finally reaching its limits as a parallel interface.

Integrated Device Electronics (IDE), later called Advanced Technology Attachment (ATA), became the standard on the personal computer. However, IDE never made a serious inroad into the server market because, while fast for single tasks, it lacked the inherent multitasking support and bus mastering that a server disk interface technology required, and no real hardware RAID solutions supported it.

Recently, the introduction of Serial ATA (SATA) technology has made serious inroads into the lower end of the server marketplace. With SATA RAID controllers built into many motherboards, and stand-alone SATA RAID boards that support 8 or more SATA drives and have substantial battery-backed RAM cache onboard, many low- to mid-range servers are finding that SATA RAID solutions provide a cost-effective alternative to SCSI. While most SATA RAID controllers lack the ability to hot-swap a failed drive, and don't have the performance potential of SCSI or Serially Attached SCSI (SAS), they are still quite attractive alternatives where cost is a primary factor. SATA also makes sense as secondary or "near-line" storage for a server.

The new kid on the block, however, is SAS. This is the most interesting addition to the server storage equation in quite a while. Using the same thin cables and connectors as SATA, with none of the configuration nuisance of traditional SCSI, SAS is definitely the way to go. When combined with new 2.5-inch drives, the ability to put a really large amount of very fast storage in a small space has taken a significant step forward. SAS drives interoperate with SATA drives to combine the two technologies on the same controller. SAS disk controllers can control SATA drives as well, though the reverse is not true.

With the main bottleneck for servers continuing to be I/O in general, and especially disk I/O, there will *continue* to be pressure to find new and faster methods to access disk-based storage. SAS, combined with 2.5-inch drives, enables fast and flexible storage arrays in remarkably smaller spaces. Because 64-bit servers are the only real option, and because of the enormous datasets supported on 64-bit Windows Server 2008, the need for fast and easily expandable disk storage keeps increasing. Windows Virtualization Technology and the move to greater virtualization in the data center also drive the need for faster disk and I/O subsystems.

Overview of Disk Management

While solid state and hybrid disks are starting to find their way into laptops and even some desktops, conventional hard disk storage continues to be the long-term storage method of choice for modern computers, from the mainframe to the desktop. In Windows Server 2008, you must first initialize this conventional hard disk storage and organize it into volumes, drives, and partitions before you can use it.

RAID

RAID (redundant array of independent disks) is a term used to describe a technique that has gone from an esoteric high-end solution to a normal procedure on most

servers. Fifteen years ago, RAID was mostly unheard of, although the original paper defining RAID was written in 1988. In the past, most server systems relied on expensive, higher-quality hard disks—backed up frequently. Backups are still crucial, but now you can use one form or another of RAID to provide substantial protection from hard disk failure. Moreover, this protection costs much less than those big server drives did.

You can implement RAID at a software or hardware level. When implemented at the hardware level, the hardware vendor provides an interface to administer the arrays and the drivers to support the various operating systems it might need to work with. Processing for the RAID array is handled by a separate processor build into the RAID controller, offloading the work from the computer's CPU. Additionally, many hardware RAID controllers include a substantial dedicated RAM cache, often with a battery backup. The combination of a separate, dedicated processor and a separate, dedicated cache provides a substantial performance advantage over software RAID. Additionally, most server-class hardware RAID controllers offer additional RAID levels when compared to software RAID, providing redundancy advantages such as multiple disk failure protection. Hardware RAID is generally substantially more expensive than the software RAID built into Windows Server 2008, though many manufacturers today include basic hardware RAID capabilities on the motherboard.

Windows Server 2008 includes an excellent and flexible implementation of RAID levels 0, 1, and 5 in *software*. It doesn't cover all the possibilities by any means, but it is certainly sufficient for some purposes. However, most serious servers should be using hardware RAID.

The primary GUI for managing disks in Windows Server 2008 is the Disk Management console, Diskmgmt.msc, shown in Figure 19-1, which can be run stand-alone or as part of Server Manager. The primary command-line tool for managing disks is DiskPart.exe.

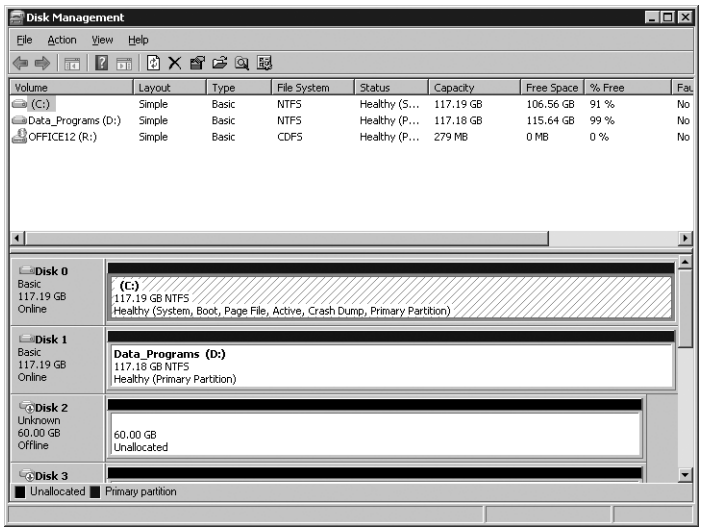

Figure 19-1 The Disk Management console.

To open Disk Management, you can start it stand-alone by running Diskmgmt.msc from a command line, or by typing it into the Run dialog box on the Start menu. Disk Management is also part of the Server Manager console, in the Storage section, as shown in Figure 19-2.

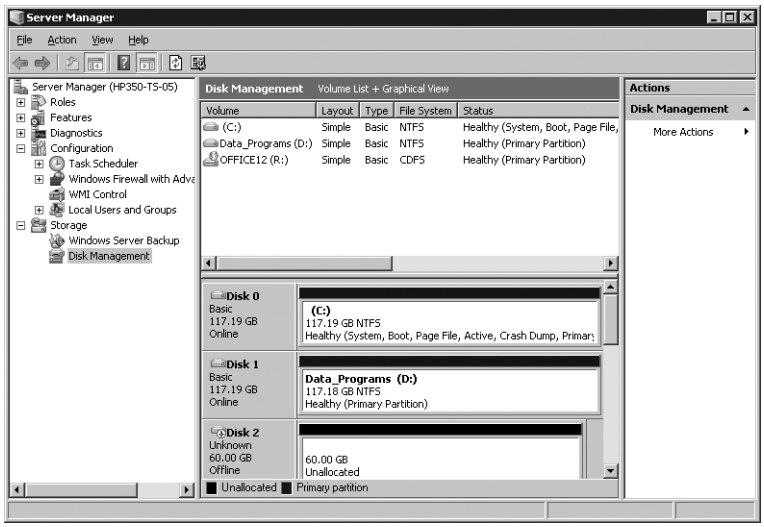

Figure 19-2 The Server Manager console.

Hardware RAID

Although Disk Management provides an adequate software RAID solution, hardware RAID is widely available, from either the original server vendor or from third parties, and it provides substantial advantages over software RAID. Hardware RAID solutions range from a simple, motherboard-integrated RAID controller to fully integrated, stand-alone subsystems. Features and cost vary, but all claim to provide superior performance and reliability over a simple software RAID solution such as that included in Windows Server 2008. In general, they do, with the notable exception of some basic motherboard-integrated solutions offered on consumer-level motherboards for SATA drives. Even if circumstances force you to use what is an essentially desktop system, avoid using the built-in RAID on the motherboard, except as a simple SATA controller. Acceptable, uncached, stand-alone RAID controllers are reasonably priced and will provide far better performance and reliability. If your budget is so limited that even that is too much, use Windows Server 2008's built-in software RAID.

Some advantages that a good hardware RAID controller offers can include the following:

- Hot-swap and hot-spare drives, allowing for virtually instantaneous replacement of failed drives
- Integrated disk caching for improved disk performance
- A separate, dedicated system that handles all processing, for improved overall performance

- Increased flexibility and additional RAID levels, such as RAID-1+0 or RAID 0+1, combinations of striping (RAID-0) and mirroring (RAID-1) that provide for fast read and write disk access with full redundancy

Not all stand-alone hardware RAID systems provide all these features, but all have the potential to *improve* the overall reliability and performance of your hard disk subsystem. They belong on any server that isn't completely fungible.

Remote Management

The Disk Management console in Windows Server 2008 lets you manage not only the local hard disks but also drives on other computers running any version of Windows 2000, Windows XP, Windows Server 2003, Windows Vista, or Windows Server 2008, allowing an administrator to manage disk tasks and space allocations from a workstation without having to sit at the computer that is being administered. This capability is a boon for remote site management and also simplifies management of Windows Server 2008 Core.

More Info For details on how to create custom management consoles that connect to remote computers, see Chapter 14 "Managing Daily Operations."

Dynamic Disks

Dynamic disks were introduced in Windows 2000 Server. By converting a disk to a dynamic disk, you give Disk Management the ability to manage it in new ways, *without requiring a reboot* in most cases. You can extend a disk volume, span a volume across multiple physical disks, stripe the volume for improved performance, mirror it, or add it to a RAID-5 array—all from the Disk Management console and all without a reboot, after the disk is converted to a dynamic disk. When combined with the new remote management functionality, dynamic disks give the system administrator powerful tools for managing the type and configuration of hard disk storage across the enterprise.

Dynamic versus Basic Disks

We used to be big fans of dynamic disks. They provided increased flexibility and functionality in a way that was pretty transparent. And they were a huge step forward when they were introduced in Windows 2000. At the time, RAID controllers were both more expensive and less functional, and many servers didn't have hardware RAID on them. That's simply not the case anymore.

If using dynamic disks increases your options, isn't that a good thing? Well, yes. But. And it's a big but. A dynamic disk complicates the disaster recovery process, and we dislike anything that creates potential issues in a disaster recovery scenario. We definitely don't think dynamic disks are appropriate for a system disk. And we just have a hard time seeing where the upside is given the functionality that your RAID controller or SAN array management application provides.

If you do find a need that can't be solved any other way, then by all means use dynamic disks. There's no apparent performance cost, and you use the same tools to manage both dynamic disks in Windows Server 2008 and basic disks. But avoid converting your system disk to *dynamic*. And make sure your disaster recovery procedures are updated appropriately.

Command Line

Windows Server 2008 includes a full command-line interface for disks. The primary command-line tool is DiskPart.exe. This command-line utility is scriptable or it can be used interactively. Additional functionality is available using Fsutil.exe and Mountvol.exe. As we go through the steps to manage disks in this chapter we'll provide the equivalent command lines and a few basic scripts that you can use as the starting point for building your own command-line tools.

The one task that doesn't appear to have a command-line solution is initializing a new disk. As far as we've been able to tell, you need to use Disk Management to initialize new disks before they can be used.

Adding a New Disk

Adding a new disk to a Windows Server 2008 server is straightforward. First, obviously, you need to physically install and connect the drive. If you have a hot-swappable backplane and array, you don't even have to shut the system down to accomplish this task. If you're using conventional drives, however, you need to shut down and power off the system.

After you install the drive and power up the system again, Windows Server 2008 automatically recognizes the new hardware and makes it available. If the disk is a basic disk that is already partitioned and formatted, you can use it without initializing, but it will initially appear "offline" in Disk Management. If it's a brand-new disk that has never been partitioned or formatted, you need to initialize it first. And if it's a dynamic disk or disks, but from another computer, you need to import it before it's available. If the disk has never been used before, you're prompted by the Initialize And Convert Disk Wizard.

Note If you're adding a drive to your server that uses a different technology than existing drives, or simply a different controller, it might require a new driver before the system recognizes the disk.

Setting a Disk Online

To set an offline disk to online, follow these steps:

1. Open Disk Management.
2. Right-click the disk you want to bring online, and select Online from the Action menu, as shown in Figure 9-3.

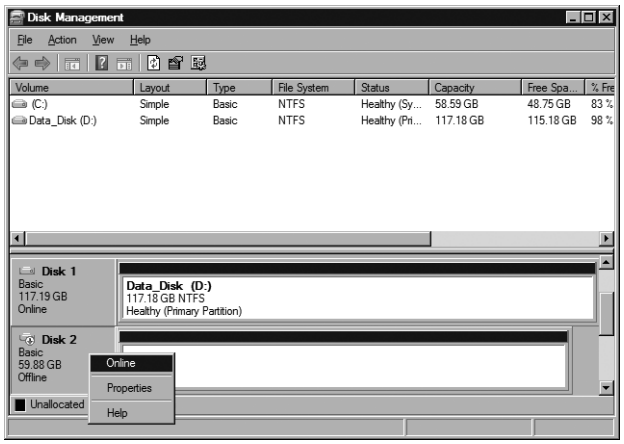

Figure 19-3 Bringing a disk online using Disk Management.

The command-line equivalent is shown in Figure 19-4.

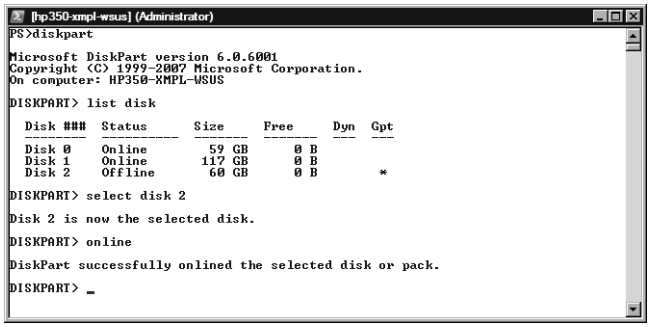

Figure 19-4 Bringing a disk online using the command line.

Initializing a New Disk

When you install a brand-new disk that has never been formatted or used by Windows, you need to initialize it. It might initially be shown as offline. If so, you need to first set the disk online, and then initialize it. If the new disk is online, the Initialize Disk dialog box will automatically display when you start Disk Management, as shown in Figure 19-5.

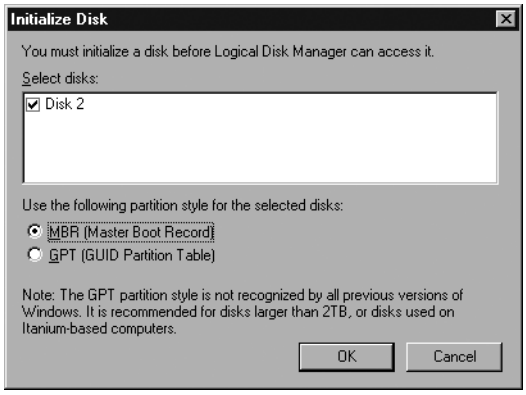

Figure 19-5 The Initialize Disk dialog box.

When you initialize the disk, you can choose whether to use Master Boot Record (MBR) or GUID Partition Table (GPT) as the partition style. For any disk larger than 2TB, GPT is recommended. We're still using MBR for all our disks, except for the one huge SAN volume we have, but we're leaning toward changing that for all new disks.

Partitions and Volumes

In Windows Server 2008 the distinction between volumes and partitions is somewhat murky. When using Disk Management, a regular partition on a basic disk is called a simple volume, even though technically a simple volume requires that the disk be a dynamic disk.

As long as you use only simple volumes or partitions, you can easily convert between a basic disk (and partition) and a dynamic disk (and a volume). Once you use a feature that is only supported on dynamic disks, however, changing back to a basic disk will mean data loss. Any operation that would require conversion to a dynamic disk will give you fair warning, as shown in Figure 19-6.

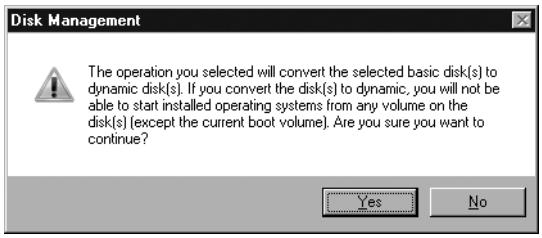

Figure 19-6 Disk Management will warn you before any operation that would cause a conversion to dynamic disks.

When using Disk Management, the conversion to dynamic disks as required happens automatically. When using DiskPart, however, you need to explicitly specify each step of the process.

Creating a Volume or Partition

You can create a new volume or partition on any disk that has empty space. If the disk is dynamic, a volume is created. If the disk is a basic disk, a primary partition is created. If the empty space is part of an extended partition, a new logical drive will be created. All of them called a simple volume, but each one a different structure.

> **Note** You can no longer create an extended partition in Disk Manager. If you need to create an extended partition, you need to use DiskPart.exe. But there's really no longer any need for extended partitions.

To create a new volume or partition, follow these steps:

1. In Disk Management, right-click the unallocated disk and select the type of volume to create, as shown in Figure 19-7. Click Next.

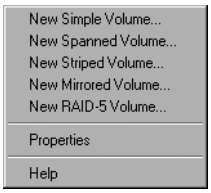

Figure 19-7 Creating a volume.

Depending on the number of available unallocated volumes, you see one or more options for the type of volume, including the following:

- New Simple Volume
- New Spanned Volume
- New Striped Volume
- New Mirrored Volume
- New RAID-5 Volume.

2. Select the type you want to create. The New Volume Wizard for that specific type of volume will open. Figure 19-8 shows the New RAID-5 Volume Wizard.

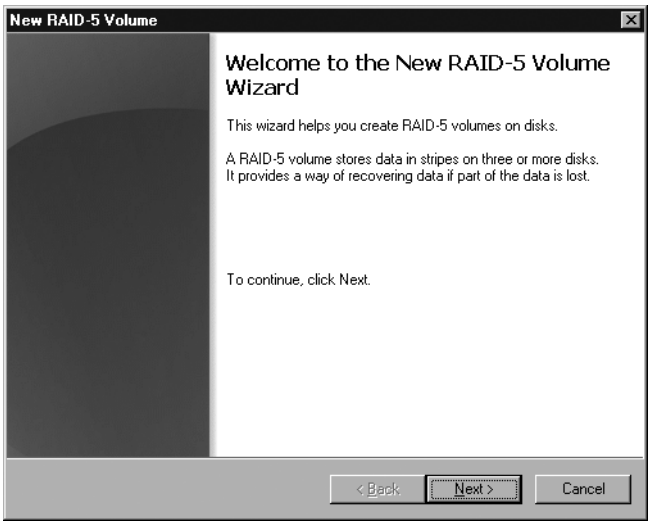

Figure 19-8 The New RAID-5 Volume Wizard.

3. Select the disks to use for the new volume. The choices available and the selections you need to make depend on the type of volume you're creating and the number of available unallocated disks. Figure 19-1 shows a RAID-5 volume being created.

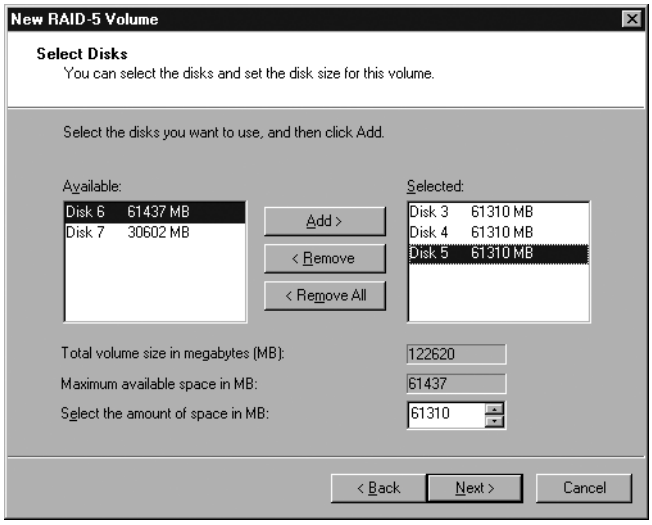

Figure 19-9 Select the disks that will be part of this volume.

4. On the same page, adjust the size of the new volume. By default, the new volume will use the maximum available space from each of the selected disks. For spanned volumes, this will be the sum of the free space on the selected disks; for other types of volumes, it will be the number of disks multiplied by the available space on the smallest of the selected disks. Click Next.

5. Select either a drive letter or a mount point for the new volume as shown in Figure 19-10, or opt not to assign a drive letter or path at this time. With Windows Server 2008, you can "mount" a volume on an empty subdirectory, minimizing the number of drive letters and reducing the complexity of the storage that is displayed to the user. If you want to take advantage of this feature, click Browse to locate the directory where you will mount the new volume. Click Next. (See the RealWorld sidebar "Mounted Volumes" for more about this subject.)

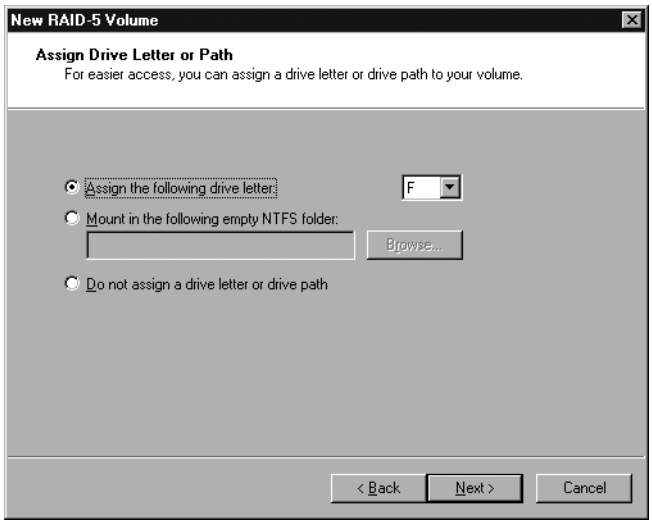

Figure 19-10 Select a drive letter or mount point for the new volume.

6. Select the formatting options you want (shown in Figure 19-11). Even when mounting the volume rather than creating a new drive, you can choose your format type without regard to the underlying format of the mount point. Click Next.

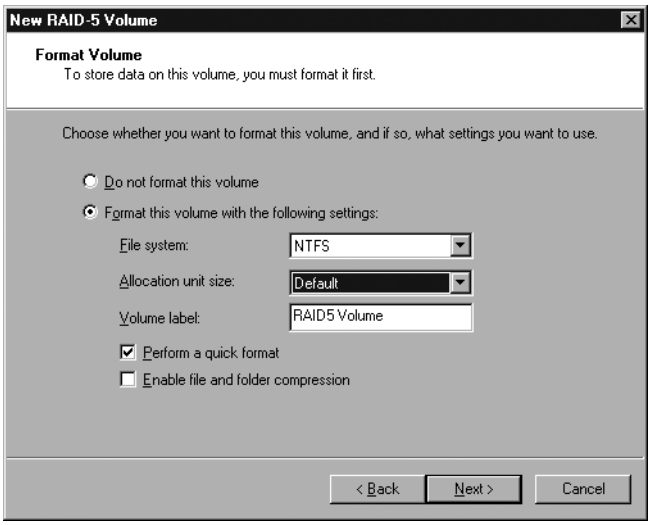

Figure 19-11 Set the formatting options for the new volume.

7. On the confirmation page, if all the options are correct, click Finish to create and format the volume. If the type you've selected requires that the disks be converted to dynamic disks, you'll see a confirmation message from Disk Management, as shown in Figure 19-12.

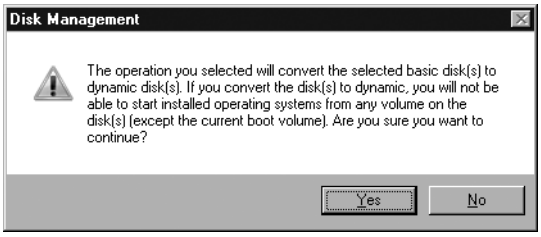

Figure 19-12 Before converting disks to dynamic, you must confirm the change.

8. Once the volume is created, it's displayed in Disk Management, as shown in Figure 19-13.

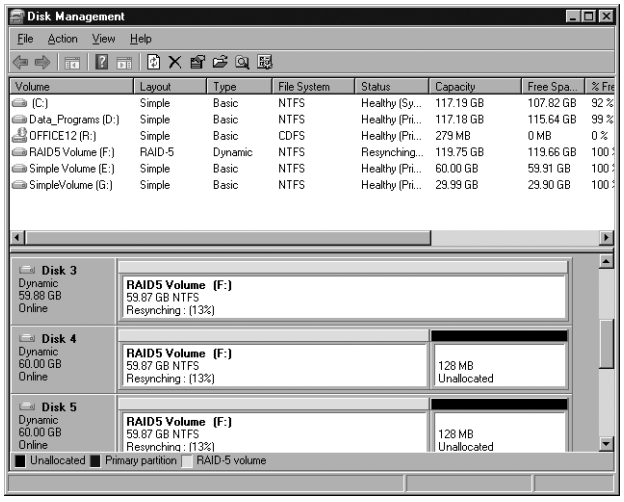

Figure 19-13 The new RAID-5 volume being created.

You could use the following script to perform the same RAID-5 volume creation using DiskPart.exe:

```
REM Filename: RAID5Vol.txt
REM
REM This is a DiskPart.exe Script. Run from the command line
REM or from another script, using the syntax:
REM
REM    diskpart /s RAID5Vol.txt > logfile.log
REM
REM to run this script and dump the results out to a log file.
REM
REM This script creates a RAID5 Volume combining disks 3,4 and 5,
REM and then formats it and assigns the next available drive letter to it.

REM First, list out our disks. Not required for scripting, but useful
REM to show the overall environment if we need to troubleshoot problems
list disk

REM Create the volume (No SIZE parameter, so the maximum size for the
REM selected disks will be used.)
create volume RAID disk=3,4,5

REM Format the new volume.
Format fs=NTFS label="RAID 5 Volume" quick

REM Assign without parameters will choose the next available HD letter.
Assign
```

Mounted Volumes

Windows Server 2008 borrows a concept from the UNIX world by adding the ability to mount a volume or partition on a subfolder of an existing drive letter. A mounted volume can also have a drive letter associated with it—although it does not need to—and it can be mounted at more than one point, giving multiple entry points into the same storage.

A volume must be mounted on an empty subfolder of an existing NTFS volume or drive. FAT and FAT32 drives do not support mounted volumes. You can, however, mount a FAT or FAT32 volume at any mount point. (But really, it's time to let go of FAT as a file system for hard disks!) You can mount only a single volume at a given mount point, but you can then mount further volumes on top of an existing mounted volume, with the same rules and restrictions as any other mount. The properties of a drive do not show all the available disk space for that drive, because they do not reflect any volumes mounted on the drive.

You can use mounted volumes to provide a mix of redundant and nonredundant storage in a logical structure that meets the business needs of the enterprise while hiding the complexities of the physical structure from the users. Unfortunately, mounted volumes are not handled correctly by Network File System (NFS) shares and should be avoided in environments where Server for NFS is used.

Creating Extended Partitions and Logical Drives

If you have extended partitions on your disks for some reason, you can create logical drives on the partition using DiskPart.exe. However, you no longer have a graphical way to create an extended partition or a logical drive, nor any real need to do so. With Windows Server 2008 providing full support for GPT disks, the old limit of a maximum of four partitions on a disk is gone—GPT disks in Windows Server 2008 support 128 partitions. If you have any existing MBR disks that include an extended partition, either because you moved a disk from another computer to your Windows Server 2008 computer or because you upgraded to Windows Server 2008 from an earlier version, we suggest you remove the existing extended partition and convert the disk to GPT.

Converting a Disk to a Dynamic Disk

Unlike earlier versions of Windows Server, with Windows Server 2008 you generally have no need to directly convert a disk to a dynamic disk. Operations that require conversion to a dynamic disk will perform the conversion as part of the operation. And deleting a volume that required dynamic disks causes the disks to convert back to basic disks in most cases. There are a few cases where the automatic conversion doesn't happen if you're using DiskPart.exe to manipulate the disk, but all the operations you perform in Disk Management do automatic conversions. For those few situations in DiskPart where explicit conversion is necessary, use the following commands:

```
DISKPART> select disk <n>
DISKPART> convert BASIC
```

Where <*n*> is the disk number you want to convert, and where BASIC can be replaced by DYNAMIC depending on which conversion you need to do.

Conversions can only occur when there are no structures on the disk that are not supported in the target disk type.

Converting a Disk to a GPT Disk

One of the important new features of Windows Server 2008 disk management is full support for GPT disks. GPT disk support was initially only available in 64-bit Itanium versions of Windows Server, but with the release of Windows Server 2003 Service Pack 1 and the initial version of x64 Windows Server 2003, GPT support was added for all versions of Windows Server 2003. In Windows Server 2008, this support is fully integrated.

You can convert a disk between MBR and GPT as long as the disk is completely empty. Unfortunately, once you've created any partitions or volumes on the disk, you can no longer convert between the two types.

To convert a disk to GPT, follow these steps:

1. In Disk Management, delete any existing volumes or partitions.

> **Note** Deleting a volume or partition will delete any data on the volume or partition. It will not destroy the data, however, so that it might be possible to recover the data.

2. Right-click the empty disk and select Convert To GPT Disk, as shown in Figure 19-14.

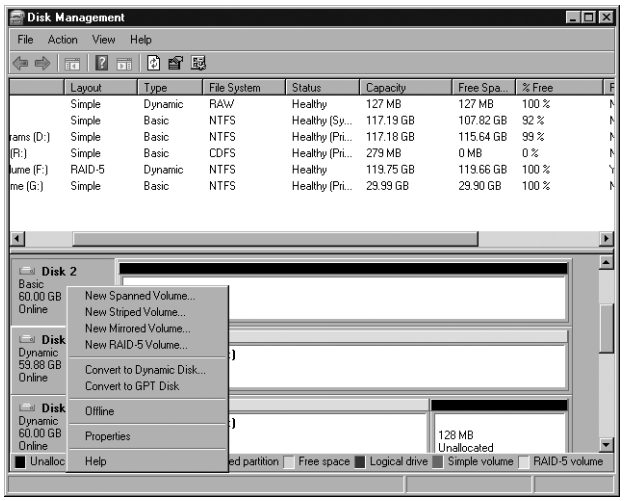

Figure 19-14 Converting from an MBR disk to a GPT disk.

3. To do the same operation from DiskPart, type the following command:

```
DISKPART> select disk <n>
DISKPART> convert GPT
```

Where <*n*> is the disk to be converted. That's all there is to it.

Changing the Size of a Volume

Windows Server 2008 allows you to change the size of an existing volume without losing data. You can extend the volume, either by using additional free space on the existing disk, or by spanning onto another disk that has free space. This capability is essentially unchanged from earlier versions of Windows Server. New to Windows Server 2008, however, is the ability to shrink a volume without having to use a third-party product or lose data.

When you extend or shrink a volume, only a simple volume or a spanned volume can be modified: You cannot extend or shrink striped, mirrored and RAID-5 volumes without deleting the volume and recreating it.

> **Important** Once you extend a volume across multiple disks, you normally cannot shrink it back down onto a single disk without deleting the volume entirely and recreating it. This means you *will* lose data, so consider carefully before you decide to extend a volume across multiple disks.

Extending a Volume

You can add space to a volume without having to back up, reboot, and restore your files if the volume is a simple volume or a spanned volume. To extend a volume, follow these steps:

1. In Disk Management, right-click the volume you want to extend. Choose Extend Volume from the menu to open the Extend Volume Wizard. Click Next.
2. Highlight one or more disks from the list of disks that are available and have unallocated space, as shown in Figure 19-15. Click Add to add the selected disk or disks, and indicate the amount of space you want to add. Click Next.

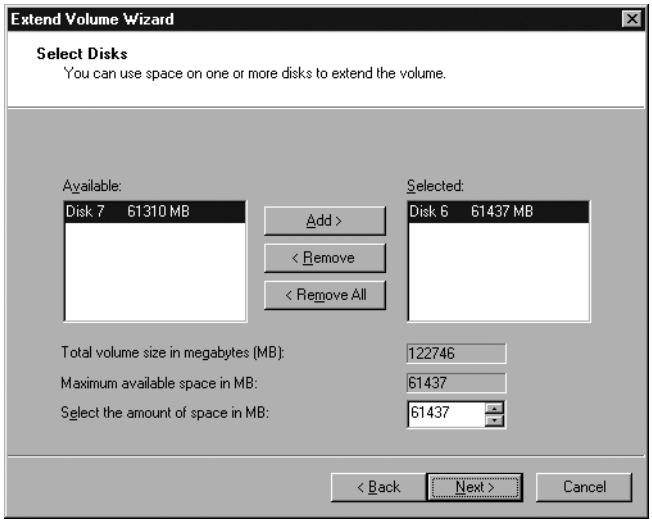

Figure 19-15 Selecting the disks to use to extend the volume.

3. The Extend Volume Wizard displays a final confirmation page before extending the volume. Click Finish to extend the volume, or click Cancel if you change your mind. If you need to convert any of the disks to dynamic before extending, you'll get another confirmation prompt.
4. To perform the same steps from the DiskPart command line, use the commands shown in Figure 19-16.

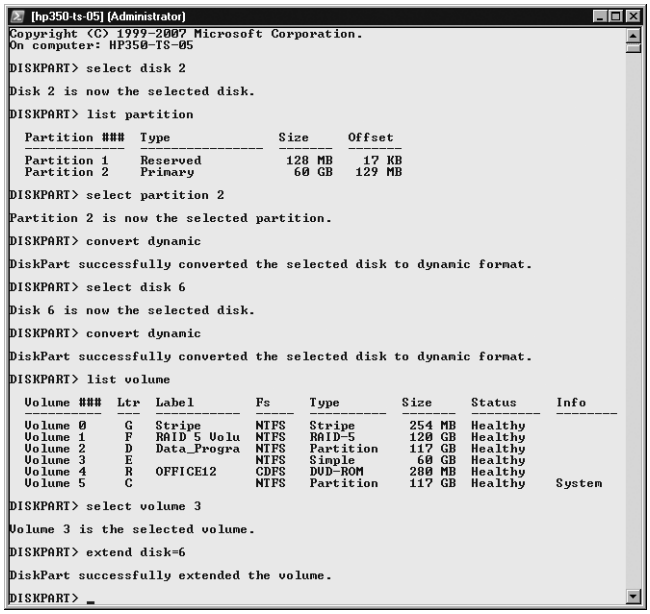

Figure 19-16 Extending a disk using the DiskPart command-line tool.

As you can see from the figure, using the command line to extend a volume is quite a few more steps than using Disk Management. Given that we hardly ever extend a volume (see the RealWorld sidebar), it's probably just as well to use Disk Management for this particular task. We're firm believers in using the command line whenever possible, but sometimes it just doesn't make sense.

Note A spanned (extended) volume is actually less reliable than a simple disk. Unlike a mirror or RAID-5 volume, which both have built-in redundancy, a spanned or striped volume will be broken and all data lost if any disk in the volume fails.

Extending—Administrator's Friend or Foe?

Most administrators have wished at some point that they could simply increase the users' home directory space on the fly. Without having to bring the system offline for several hours while the entire volume is backed up and reformatted to add the additional hard disks, the backup is restored, and the share points are re-created. Fun? Hardly. Risky? Certainly. And definitely a job that means coming in on the weekend or staying late at night—in other words, something to be avoided if at all possible.

All this makes Windows Server 2008's ability to create additional space on a volume without the need to back up the volume, reformat the disks, and re-create the volume a seductive feature. However, if you're using conventional hard disks without hardware RAID, you might want to think twice before jumping in. Only spanned or striped volumes allow you to add additional storage on the fly, and, because neither is redundant, using them exposes your users to the risks of a failed drive. Yes, you have a backup, but even under the best of circumstances, you'll lose some data if you need to restore a backup. Further, using spanned volumes actually

increases your risk of a hard-disk failure. If any disk used as part of the spanned volume fails, the entire volume is toast and will need to be restored from backup.

Why, then, would anyone use spanning? Because they have hardware RAID to provide the redundancy. This combination offers the best of both worlds—redundancy provided by the hardware RAID controller and flexibility to expand volumes as needed, using Disk Management. Yet another compelling argument for hardware RAID, in case you needed any more.

Shrinking a Volume

While most of the time we're concerned with increasing the size of a volume on the server, there can be occasions when it might be convenient to shrink a volume. For example, if you are using a single large RAID array for multiple volumes, and one of the volumes has empty space while another volume on the same array is running out of space, it would be handy to be able to shrink the volume that has extra space and then extend the one that is running out of room. In the past, the only way you could do this was to back up the volume you wanted to shrink, delete it, extend the volume that needed growing, recreate the volume you deleted, and restore the backup. Possible, certainly. But both risky and highly disruptive to your users. The other alternative was to use a third-party product, such as Acronis Disk Director Server (*http://www.acronis.com/enterprise/products/diskdirector/*).

Now, in Windows Server 2008, you can use Disk Management to shrink a volume without having to delete it and recreate it. While not quite as flexible as products like Acronis Disk Director, this new capability is all that most system administrators will need. To shrink a volume, follow these steps:

1. In Disk Management, right-click the volume you want to shrink. Choose Shrink Volume from the menu to open the Shrink dialog box shown in Figure 19-17.

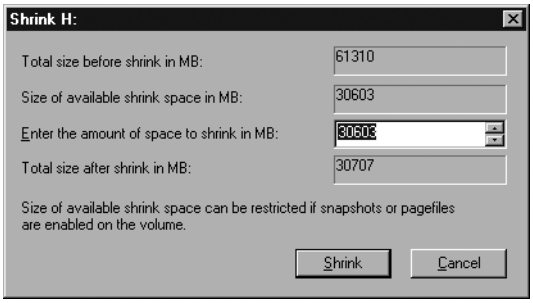

Figure 19-17 Shrinking a volume.

2. Select the amount of space to shrink the volume by, and click Shrink.

3. From the command line, the syntax of the DiskPart command is:

```
SHRINK [DESIRED=<N>] [MINIMUM=<N>] [NOWAIT] [NOERR]
SHRINK QUERYMAX [NOERR]
```

where SHRINK by itself will shrink the selected volume the maximum amount possible.

Note Shrinking a volume is one place where DiskPart is well behaved. If you select a partition on a basic disk and attempt to shrink it, DiskPart doesn't require you to first convert the disk to dynamic before you can shrink the volume.

Adding a Mirror to a Volume

When your data is mission critical and you want to make sure that no matter what happens to one of your hard disks the data is protected and always available, consider mirroring the data onto a second drive. Windows Server 2008 can mirror a dynamic disk onto a second dynamic disk so that the failure of either disk does not result in loss of data. To mirror a volume, you can either select a mirrored volume when you create the volume (as described in the "Creating a Volume" section earlier in this chapter) or add a mirror to an existing volume. To add a mirror to an existing volume, follow these steps:

1. In the Disk Management console, right-click the volume you want to mirror. If a potential mirror is available, the shortcut menu lists the Add Mirror command, as shown in Figure 19-18.

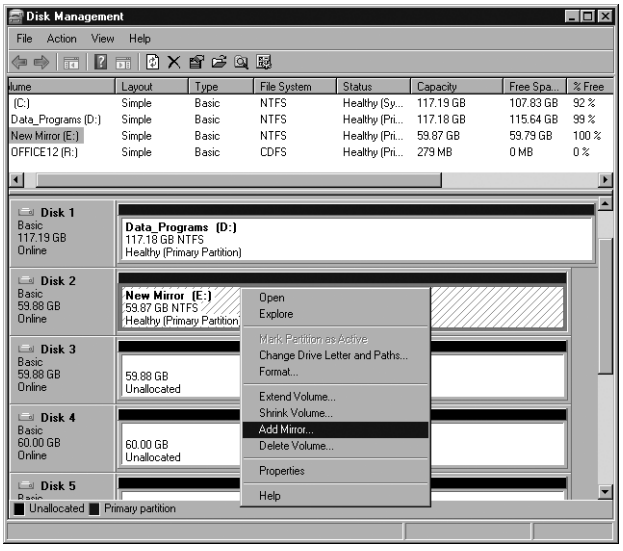

Figure 19-18 The action menu for Disk 2 includes the Add Mirror command.

2. Choose Add Mirror to display the Add Mirror dialog box (shown in Figure 19-19), where you can select the disk to be used for the mirror.

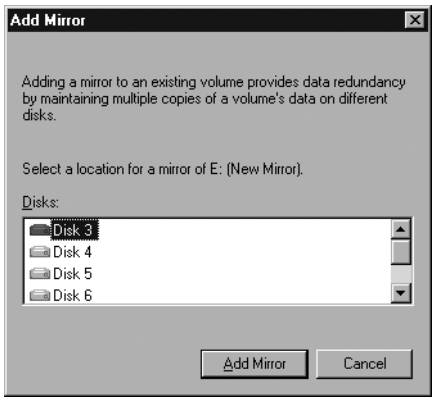

Figure 19-19 The Add Mirror dialog box.

3. Highlight the disk that will be the mirror and click Add Mirror. You'll be prompted that this action will convert the disks to dynamic. Click Yes. The mirror is created immediately and starts duplicating the data from the original disk to the second half of the mirror, as shown in Figure 19-20. This process is called regeneration or resynching. (The process of regeneration is also used to distribute data across the disks when a RAID-5 volume is created.)

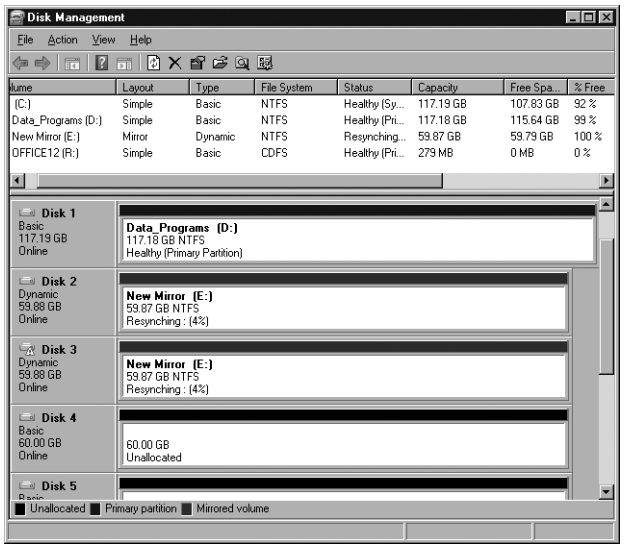

Figure 19-20 A newly created mirrored disk in the process of regeneration.

4. Mirroring can also be done from the DiskPart command line. First select the disk and then use the ADD command, which has the following syntax:

```
ADD DISK=<N> [ALIGN=<N>] [WAIT] [NOERR]
```

where DISK is the disk that will be added to make the mirror, and ALIGN is used to align with a specific hardware RAID Logical Unit Number (LUN) alignment boundary.

Best Practices Regeneration is both CPU-intensive and disk-intensive. When possible, create mirrors during slack times or during normally scheduled downtime. Balance this goal, however, with the equally important goal of providing redundancy and failure protection as expeditiously as possible.

Best Practices To improve your overall data security and reliability, mirror your volumes onto disks that use separate controllers whenever possible. This process is known as *duplexing*, and it eliminates the disk controller as a single point of failure for the mirror while actually speeding up both reading and writing to the mirror, because the controller and bus are no longer potential bottlenecks.

Drive Failure in a Mirrored Volume

If one of the disks in a mirrored volume fails, you continue to have full access to all your data without loss. If a disk in the mirror set fails, the failed disk is marked missing and offline, and the mirror is unavailable, as shown in Figure 19-21. An alert is sent to the alert log.

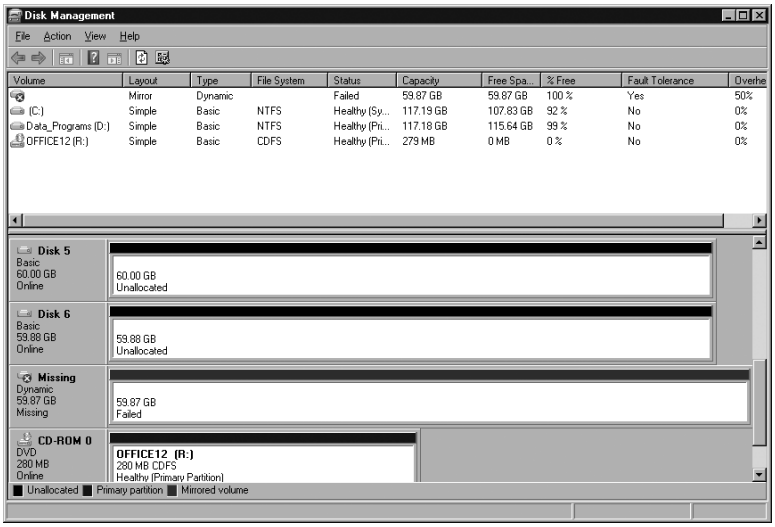

Figure 19-21 Failed disk in mirror shown as missing and offline.

Once the mirror is unavailable, you need to remove, or "break," the mirror, bringing the good disk back online and available. Once the problem disk has been replaced, you can rebuild the mirror by following the steps in the section "Adding a Mirror to a Volume" earlier in the chapter.

To remove the mirror, follow these steps:

1. In Disk Management, right-click either disk and select Remove Mirror from the action menu, as shown in Figure 19-22.

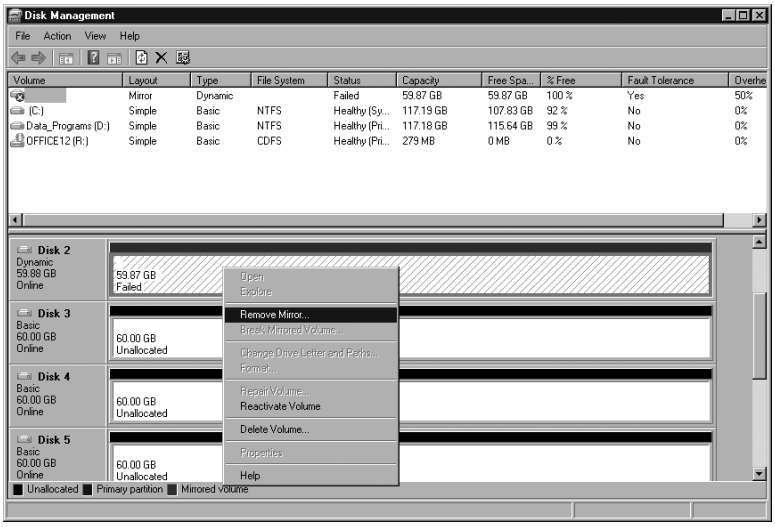

Figure 19-22 Breaking the mirror of a failed mirror pair.

2. In the Remove Mirror dialog box, select the failed disk and click Remove Mirror.

After you replace the failed disk or correct the problem and reactivate the failed disk, The mirror automatically starts regenerating if you didn't have to remove the mirror. If you can solve the problem without powering down the system, you can regenerate the mirror on the fly. To reactivate the failed disk, follow these steps:

1. Right-click the icon for the failed disk on the left side of the Disk Management console.
2. Choose Reactivate Disk. Windows Server 2008 warns you about running chkdsk on any affected volumes, brings the disk back online, and starts regenerating the failed mirror.

Removing a Mirror

We all know that every system administrator is always fully aware of the ongoing requirements of her servers, and never runs out of disk space without plenty of warning. Oh, wait, this is a RealWorld sidebar. OK, reality check, then. If you have the luxury of huge budgets and large, flexible, highly redundant Storage Area Networks, you probably haven't been caught short on disk space. But if you're running a more ordinary network where budgets interfere and resources are constrained, we strongly suspect you've certainly had times when you were scrambling to clean up disks to make sure you didn't run out of room for a critical process. Certainly we have. If you have a mirrored volume, you can get yourself out of trouble pretty quickly. But at a significant risk in the long run.

Just remove the mirror from the mirrored volume. When you remove a mirror, the data on one of the disks is untouched, but the other disk becomes unallocated space. You can then use the unallocated space to extend the volume that is short.

Of course, you will have lost all redundancy and protection for the data, so you need to take steps to restore the mirror as soon as possible. Plus the volume you've extended is now more susceptible to failure, since it has an extra disk included in it. Until you can buy more disks, you'll want modify your backup schedule for the affected disks. And don't put off buying the new disks—you're at serious risk until you get your system back to where it should be.

Setting Disk Quotas

Windows Server 2008 supports two mutually exclusive methods for setting quotas on the amount of file system resources a user can use—disk quotas or directory quotas. Disk quotas were introduced in Windows 2000, and are applied to specific users and limit the amount of disk space that user can use on a particular volume. Directory quotas are applied to all users and limit the amount of disk space that users can use in a particular folder and its subfolders. Directory quotas were introduced in Windows Server 2003 R2 with the new File Server Resource Manager, and they are covered in detail in Chapter 20.

Enabling Quotas on a Disk

By default, disk quotas are disabled in Windows Server 2008. You can enable disk quotas on any volume that has been assigned a drive letter. To enable quotas on a volume, follow these steps:

1. In Windows Explorer, right-click a drive letter and open the properties of that drive.
2. Click the Quota tab, shown in Figure 19-23, and then click Show Quota Settings.

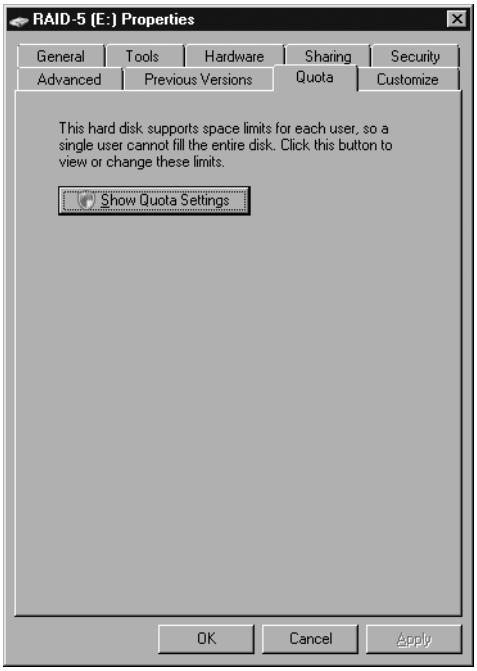

Figure 19-23 The Quotas tab of a drive's properties.

3. Select the Enable Quota Management check box to enable quotas for the disk, as shown in Figure 19-24.

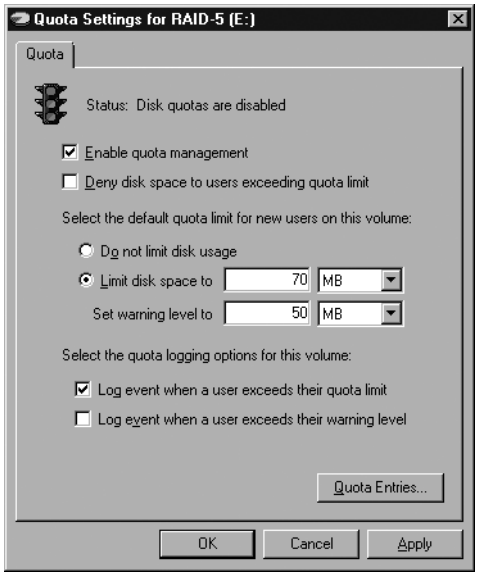

Figure 19-24 The Quota Settings dialog box for a disk.

4. To enable hard quotas that can't be exceeded, select the Deny Disk Space To Users Exceeding Quota Limit check box.
5. Set the limits and warning level, as shown in Figure 19-24. You can also enable logging on this page.
6. Click OK to enable the quotas. You'll be prompted one last time to confirm, as shown in Figure 19-25. Click OK and the quotas will be enabled.

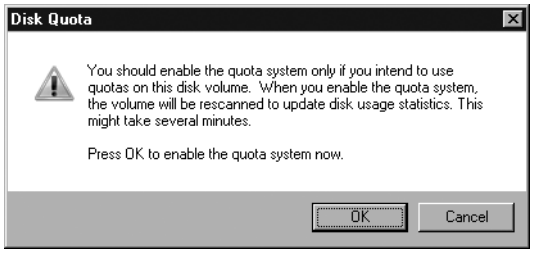

Figure 19-25 The Disk Quota confirmation message.

Setting Per-User Quotas

You can set quota limits on individual users, or you can have limits apply equally to all non-administrative users. Unfortunately, you can't set limits on groups of users. And any users who already own files on the disk will have their quotas initially disabled. New users will have the default quotas for the disk applied as you would expect when they first save a file on the disk.

To set the quotas for individual users, follow these steps:

1. In Windows Explorer, right-click a drive letter and open the properties of that drive.

2. Click the Quota tab, and then click Show Quota Settings to bring up the Quota Settings dialog box for that disk.
3. Click Quota Entries to open the Quota Entries dialog box for the disk, as shown in Figure 19-26.

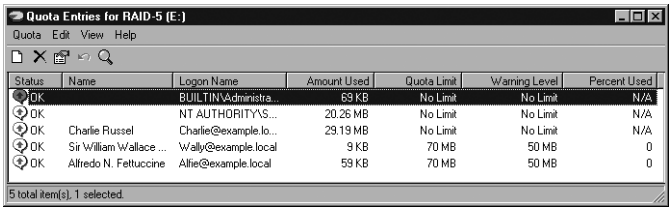

Figure 19-26 The Quota Entries dialog box for a disk.

4. To modify the quota for a user already listed, select the user and then click Properties to open the quota settings for that user, as shown in Figure 19-27. Set the quota for the user and click OK to return to the Quota Entries dialog box.

Figure 19-27 The Quota Settings dialog box for an individual user.

5. To create a quota for a user who doesn't have one yet, and who needs a quota different from the default for the disk, click New Quota Entry.
6. Select the user or users to apply the new quota to, and click OK to bring up the Add New Quota Entry dialog box, as shown in Figure 19-28.

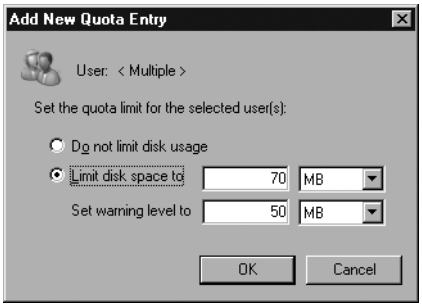

Figure 19-28 The Add New Quota Entry dialog box.

7. Click OK to add the new entry and return to the Quota Entries dialog box. Close the Quota Entries dialog box, click OK in the Quota Settings dialog box, and then click OK in the Properties dialog box for the drive.
8. To manage quotas from the command line, you need to use Fsutil.exe. Even for a determined command-line type, it's pretty lame. Stick to the GUI, and use import and export whenever possible.

Importing and Exporting Quotas

Managing disk quotas is a potentially tedious job if you try to use fine-grained control of individual quotas. The best solution is to use a single, general quota that is correct for almost all users, and then only do limited exceptions to that quota for very specialized cases. If you do have complicated quotas, however, and you need to transfer them to another server or another volume, you can export a set of quotas and then import them to another volume.

To export the quotas on a volume, follow these steps:

1. Open the Quota Settings page for the volume you want to export the quotas from.
2. Click Quota Entries to open the Quota Entries dialog box.
3. Highlight the quotas you want to export.
4. Choose Export from the Quota menu. Type in a name and location for the export file and click Save.

To import a quota file to a volume, follow these steps:

1. Open the Quota Settings page for the volume you want to import the quotas to.
2. Click Quota Entries to open the Quota Entries dialog box.
3. Choose Import from the Quota menu. Type in a name and location for the import file and click Open.
4. If there are conflicting quotas, you'll be prompted to replace the existing quotas, as shown in Figure 19-29.

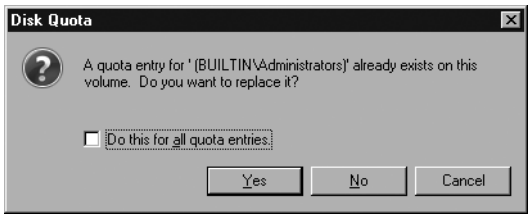

Figure 19-29 Importing quotas can cause an existing quota to be replaced.

5. Choose to replace a quota by clicking Yes or to not keep the existing one by clicking No. You can have the action repeated for any further conflicts by selecting the Do This For All Quota Entries check box.

Just Say No to Disk Quotas

Disk quotas, which were originally introduced in Windows 2000, were a big step forward and gave the Windows system administrator a new and valuable tool to limit the spiraling growth of storage requirements on the server. But like many Microsoft version 1.0 implementations, it wasn't a perfect solution. It's difficult to manage quotas effectively without creating too many exceptions to easily keep track of. You can only apply quotas on a per-drive letter level, and they don't affect mounted volumes at all. And quotas are indiscriminant—they treat document files the same way they treat .MP3 files.

Quotas also arrived too late to the scene. Just about the time disk quotas were introduced, the hard disk industry started a round of massive growth in hard drive size. At the same time, the price of even enterprise-class hard drives came down dramatically.

Finally, with the introduction of the File Server Resource Manager, we now have folder-level quotas and file-type filtering. If you need quotas, we recommend that you use these.

Enabling File Encryption

With the introduction of Windows 2000, Microsoft added the ability to encrypt individual files or entire subdirectories stored on an NTFS volume in a totally transparent way. To their creator, encrypted files look exactly like regular files—no changes to applications are required to use them. However, to anyone except the creator/encryptor, the files are unavailable. Even if someone did manage to gain access to them, they would be gibberish because they're stored in encrypted form.

Encryption is simply an advanced attribute of the file, like compression. However, a file cannot be both compressed and encrypted at the same time—the attributes are mutually exclusive. Encrypted files are available only to the encryptor, but they can be recovered by the domain or machine recovery agent if necessary. You can back up encrypted files by normal backup procedures if the backup program is Windows Server 2008–aware. Files remain encrypted when backed up, and restored files retain their encryption.

Under normal circumstances, no user except the actual creator of an encrypted file has access to the file. Even a change of ownership does not remove the encryption. This prevents sensitive data—such as payroll information, annual reviews, and so on—from being accessed by the wrong users, even ones with administrative rights.

Note Encryption is available only on NTFS. If you copy the file to a floppy disk or to any other file system, the file is no longer encrypted. This means that if you have a USB key drive, for example, that is formatted with FAT, or if you use NFS file systems, copying the file there will remove the encryption.

When you encrypt a folder, all new files created in that folder are encrypted from that point forward. You can also elect to encrypt the current contents when you perform the encryption. However, be warned that if you choose to encrypt the contents of a folder when it already contains files or subfolders, those files and subfolders are encrypted *for the user performing the encryption only*. This means that even files owned by another user are encrypted and available for your use only—the owner of the files will no longer be able to access them.

When new files are created in an encrypted folder, the files are encrypted for use by the creator of the file, not the user who first enabled encryption on the folder. Unencrypted files in an encrypted folder can be used by all users who have security rights to use files in that folder, and the encryption status of the file does not change unless the filename itself is changed. Users can read, modify, and save the file without converting it to an encrypted file, but any change in the name of the file triggers an encryption, and the encryption makes the file available only to the person that triggers the encryption.

Note If you use EFS, it is *essential* that you back up EFS certificates and designate a Recovery Agent to protect against *irreversible* data loss. EFS certificates and recovery agents are covered in Chapter 23, "Implementing Security."

To encrypt a file or folder, follow these steps:

1. In Windows Explorer, right-click the folder or files you want to encrypt, and choose Properties from the shortcut menu.
2. Click Advanced on the General tab to open the Advanced Attributes dialog box shown in Figure 19-30.

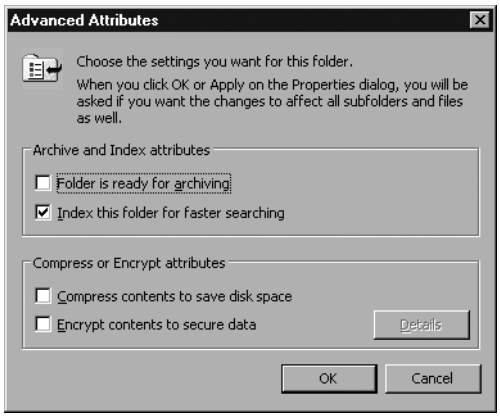

Figure 19-30 The Advanced Attributes dialog box.

3. Select the Encrypt Contents To Secure Data check box and click OK to return to the main Properties window for the folder or file. Click OK or Apply to enable the encryption. If any files or subfolders are already in the folder, you're presented with the dialog box shown in Figure 19-31.

Figure 19-31 Choosing whether to encrypt the files already in a folder or just new files.

4. If you choose Apply Changes To This Folder Only, all the current files and subfolders in the folder remain unencrypted, but any new files and folders are encrypted by the creator as they are created. If you choose Apply Changes To This Folder, Subfolders, And Files, all the files and folders below this folder are encrypted so that only you can use them, regardless of the original creator or owner of the file.
5. Click OK and the encryption occurs.

The Limitations of EFS

The EFS capabilities of Windows Server 2008 provide a useful way to encrypt folders and files to prevent unauthorized access. However, EFS has limitations, and you need to manage it carefully to not create issues.

Once an EFS folder is created, any files created in the folder will always be encrypted *by the creator of the file*. This is not at always what you intend. If you have a publicly available folder that has encryption on it, you need to carefully manage who has access to that folder using NTFS file permissions, share permissions, or other methods of preventing unauthorized access.

Another problem is that anyone who has access to your system drive *can* break EFS encryption. This shouldn't be a big problem on a well-secured server, but it's still a concern. The solution is to enable BitLocker on your server. BitLocker was introduced with Windows Vista as a solution for the mobile laptop, but it has very real possibilities for the enterprise trying to fully secure their environment. For more on BitLocker, see Chapter 23.

Summary

Windows Server 2008 provides the system administrator with a richer set of disk management tools than any previous version of Windows. Disk Management is now smarter, with automatic, seamless conversion between basic and dynamic disks. The full support for GPT disks eliminates the need for extended partitions, and gives Windows Server 2008 the ability to support really *large* disks. And the ability to shrink or extend a volume without taking it offline gives the system administrator much greater flexibility.

In the next chapter, we'll cover the many aspects of storage, including Storage Area Networks, the Storage Resource Manager, and removable and remote storage.