Workstation
Server
Enterprise

CHAPTER

3

Configuring Disks

About Chapter 3

This chapter explores the basics of configuring a computer's hard disk(s) in a Windows NT environment.

It begins with a discussion of the various file systems that Windows NT supports, namely, FAT and NTFS. The strengths, limitations, and special features of these two file systems are explained. Converting from FAT to NTFS is also covered.

Chapter 3 then presents an in-depth explanation of working with partitions. Disk Administrator, a Windows NT tool used to create and format partitions, is introduced. Then various disk partitioning schemes are explored and compared, including: disk mirroring, stripe sets, stripe sets with parity, and volume sets.

Finally, Chapter 3 provides valuable information on how to recover from a single or multiple disk failure. Updating the Emergency Repair Disk is also covered.

This chapter includes two hands-on labs. In the first lab you use Disk Administrator to manage partitions on your computer's hard disk. In the second you get an opportunity to update your own Emergency Repair Disk.

No matter which of the three Windows NT 4.0 Microsoft Certified Professional exams you're preparing for, you'll want to read this chapter. The information in this chapter covers several objectives listed in the Planning, Installation and Configuration, and Troubleshooting sections in these exams' objectives.

File Systems

Before you attempt to configure disks in a Windows NT environment, you should have a clear understanding of the different file systems that NT supports. Windows NT 4.0 supports three file systems: the *file allocation table* (FAT) *file system,* the *Windows NT file system* (NTFS), and the *Compact Disc Filing System* (CDFS). Windows NT 4.0 does *not* support the *high performance file system* (HPFS), although earlier versions of NT did. (If you are upgrading from an earlier version of Windows NT that used HPFS, you must convert to NTFS before performing the upgrade.) Table 3-1 shows which file systems are supported by various operating systems.

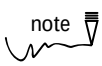
TABLE 3-1 FILE SYSTEM SUPPORT BY OPERATING SYSTEM

OPERATING SYSTEM	*FILE SYSTEMS SUPPORTED*
Windows NT 4.0	FAT, NTFS, CDFS
Windows NT 3.51 (and earlier versions)	FAT, NTFS, CDFS, HPFS
Windows 95	FAT, CDFS
Windows 3.*x* and 3.1*x*	FAT, CDFS
OS/2	FAT, CDFS, HPFS
MS-DOS	FAT, CDFS

FAT

The *file allocation table* (FAT) *file system* used on Windows NT is a modified version of the FAT file system used by MS-DOS. FAT is the only hard disk file system supported by Windows 3.*x*,Windows 3.1*x*, Windows 95, and MS-DOS. So, if you want to configure a Windows NT computer to dual boot between Windows NT and Windows 3.1*x*, Windows 95, or MS-DOS, your computer's first partition must use the FAT file system.

Don't confuse the FAT file system with the FAT32 file system. Windows NT does not support the FAT32 file system (an enhanced rendition of FAT) that is supported on the original equipment manufacturer (OEM) version of Windows 95 that includes Service Pack 2.



Service Pack 2 for Windows 95 is only available on the OEM version of Windows 95, not on the retail version.

Now it's time to familiarize yourself with the characteristics and features of the FAT file system, including security, naming conventions, speed of access to files, and partition size.

Security

The FAT file system does *not* support file and folder security in Windows NT. Because file and folder security is not supported on a FAT partition, any user who is logged on locally to a computer has full control of all of the files and folders located in the FAT partition(s) on that computer. This applies only to local access.

However, you can use share permissions to control users' access to shared folders over the network. Share permissions affect only the access of files and folders over the network, not when someone is logged on locally. So, if you need local file and folder security, you should use an NTFS partition instead of a FAT partition.

Naming conventions

The FAT file system, as used by Windows NT, supports the use of long filenames. This file system permits the full path to a file (including the filename) to be up to 255 characters long.

Filenames can contain any character *except* \ /:*?"<>| and should begin with an alphanumeric character. Filenames can contain spaces and multiple periods, and the characters after the last period are considered the filename extension.

The FAT file system preserves upper- and lowercase in filenames, but filenames are not case-sensitive. Because of this, I can request the file `ALAN.DOC` by typing **Alan.doc**, **ALAN.DOC**, or **alan.doc**, and Windows NT always retrieves `ALAN.DOC`.

Speed of access to files

Access speed to files on a FAT partition is dependent on many factors, including file type, file size, partition size, number of files in a folder, and fragmentation.

Windows NT accesses files on FAT partitions smaller than 500MB faster than it accesses files on other similar-sized file system partitions discussed here. Additionally, NT accesses certain types of files on FAT partitions more efficiently than on partitions formatted with other file systems.

On very large partitions, however, or when there is a large number of files in a folder, Windows NT accesses files on NTFS partitions much faster than it accesses files on a FAT partition of similar size.

Windows NT usually accesses files on a highly fragmented FAT partition slower than it accesses files on an NTFS partition of similar size.

Partition size

The maximum size of a FAT partition is 4GB. The maximum size of a file in a FAT partition is 4GB.

The FAT file system does *not* support file compression.

NTFS

The *Windows NT file system* (NTFS) is the most powerful file system supported by Windows NT. Only Windows NT (both Windows NT Workstation and Windows NT Server) supports NTFS—no other operating systems currently support this file system.

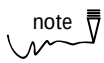
When it comes to security, naming conventions, speed of access to files, and partition size, NTFS has its own unique characteristics. Additionally, NTFS has some features not supported by the FAT file system.

Security

NTFS provides file and folder security for both local and remote users on a network. NTFS is the only file system discussed here that permits the assigning of permissions to individual files and folders.

So how does NTFS security actually work? NTFS security controls access to files on an NTFS partition by utilizing the user's *security identifier* (SID) to determine which files that user can access. (Each file and folder on an NTFS partition has an *access control list* [ACL] associated with it. The ACL is a list that contains user and group SIDs, with the associated privileges of each user and group.)

NTFS and share security are covered in depth in Chapter 12.



In addition to the security provided by NTFS, remember that because Windows NT requires a user to log on before accessing files, NT's security is greater than operating systems that don't require the user to log on.

Naming conventions

Like the FAT file system, NTFS supports the use of long filenames. Names of files and folders (including extensions) can be up to 255 characters long.

You can use most characters in NTFS file and folder names. However, the characters ?"/\ < >* | : can't be used.

NTFS preserves upper- and lowercase in filenames. Filenames are not case-sensitive (except when used by a POSIX application). For example, a Win32 application does not distinguish between `Money.DOC`, `MONEY.DOC`, and `money.doc`—it treats all three names as though they were the same file.

The POSIX subsystem, however, is case-sensitive with respect to filenames, because it does not translate a request for a file into all uppercase letters as the Win32 and other subsystems do. A POSIX application treats the filenames in the previous paragraph as though they were three separate files: `Money.DOC`, `MONEY.DOC`, and `money.doc`. You must use a POSIX application if you want to access these three different files — if you attempt to access `Money.DOC` with a Win32 application (no matter how you type the filename) you will normally always retrieve the `MONEY.DOC` file because the Win32 Subsystem translates file requests into all uppercase letters.

Speed of access to files

NTFS usually provides faster access to files stored on a large partition that contains many files than the FAT file system. NTFS is able to access files in this situation faster than the FAT file system because NTFS uses an enhanced binary tree to locate files. A binary tree search is a faster mechanism for searching through a large number of filenames than the sequential read mechanism used on FAT partitions.

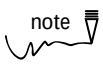
Partition size

The maximum theoretical size of an NTFS partition is 16 exabytes (an *exabyte* is one billion billion bytes, or a giga-gigabyte). However, when you actually implement NTFS on current standard industry hardware, there is a functional limitation of 2 terabytes.

Additional features not supported by the FAT file system

NTFS has several other unique attributes and features that are not found in, nor supported by, the FAT file system.

- NTFS supports a compression attribute for each file. You can choose which files to compress and which ones to leave uncompressed. The compression algorithm NTFS uses is similar to the one used by Drivespace in MS-DOS. Using compression provides an approximately 40 to 50 percent increase in hard disk space.



Compression can cause some performance degradation on partitions with substantial write activity. Additionally, accessing uncompressed files is faster than accessing compressed files.

- NTFS is a highly reliable, recoverable file system. It is not necessary to periodically run `Chkdsk.exe` on an NTFS partition.
- Using NTFS greatly reduces fragmentation on partitions. However, files can still become fragmented when their size is increased. Windows NT does not include an NTFS defragmentation utility, but there are several third-party utilities available.
- NTFS maintains a recycle bin for each user.

A couple of final facts on NTFS: You can't use NTFS to format floppy disks, and NTFS does not permit you to change media in a removable media drive (such as a Zip drive) without rebooting. (The FAT file system does support changing media without rebooting.)

CDFS

The *Compact Disc Filing System* (CDFS) supports access to compact discs. It is only used on CD-ROM devices that read and/or write compact discs.

Converting from FAT to NTFS

In Windows NT you can format a new partition with either FAT or NTFS. But what do you do when you want to change the file system on an existing partition?

You can change an existing FAT partition, and retain the data on it, into an NTFS partition by using `Convert.exe`. This is a fairly simple procedure.

However, it is a one-way process—there is no way to convert an NTFS partition into a FAT partition without first backing up, reformatting the disk, and restoring the data.

To convert a FAT partition into an NTFS partition, use the following syntax:

`CONVERT` *`drive`*`: /FS:NTFS [/V]`

The following is an explanation of syntax:

- *`Drive`*: This specifies the letter of the drive to convert to NTFS.

- `/FS:NTFS`: This indicates that the file system should be converted to NTFS. This is an outdated switch, because NTFS is the only file system that you can use `Convert.exe` to switch to in Windows NT 4.0; but its use, in terms of command syntax, is still required.
- `/V`: This specifies that `Convert.exe` will run in verbose mode.

For example, to convert drive D: from FAT to NTFS, use the following command line: **CONVERT D: /FS:NTFS**

To successfully use the `Convert.exe` command, `Convert.exe` must be the *only* application that accesses the drive you want to change during the conversion process. If Windows NT Explorer accesses the drive you are trying to convert, if you are trying to convert the boot partition, or if your active command prompt has the drive you are trying to convert as its current drive, Windows NT will display an error message stating that `Convert.exe` can't gain exclusive access to the drive.

If you can't gain exclusive access to a drive, run `Convert.exe` as shown in the example. The file system conversion will occur when you restart your computer.

Working with Partitions

Before you can format a hard disk with a file system such as FAT or NTFS, you must mark the disk to identify which parts of it will contain a file system (or systems). This is called *partitioning*. A hard disk can be separated into a maximum of four partitions, or one partition can occupy all of the space on a disk.

You may recall that back in Chapter 2, when you installed Windows NT 4.0, you were presented with the option to create a partition on your computer's hard disk. Once Windows NT is installed, the primary tool for creating, formatting, and managing various types of partitions is Disk Administrator. The latter part of this chapter covers how to use Disk Administrator to perform specific tasks on partitions. But first, it's important that you understand the basic types of partitions.

Partition Types

Windows NT supports two types of partitions: primary and extended. Both types of partitions can coexist on the same hard disk. A disk can have more than one primary partition, but it can have only one extended partition.

Primary partitions

A *primary partition* can occupy all of the space on a disk, or any portion of it. A hard disk can have up to four partitions, and all four can be primary partitions. A primary partition can be formatted as a single logical drive (but not as multiple logical drives).

Any primary partition on the first hard disk in the computer can be designated as the active partition. The active partition is significant because when the computer boots, it attempts to load the operating system from the active primary partition on the first hard disk in the computer.

The Windows NT 4.0 system partition must be located on the active primary partition on the first hard disk in a computer.

Remember the system partition and the boot partition? If your memory needs refreshing, take a peek at Chapter 2, in the "Hard Disk Partition Information" section of the Preinstallation Checklist.

Extended partitions

There can be only one *extended partition* on a disk. An extended partition can't be marked active, and it can't be used for the system partition of a computer. The Windows NT boot partition, however, can be located on an extended partition.

An extended partition can be formatted as one or more logical drives, where each partition is assigned a different drive letter. Logical drives can be formatted with either FAT or NTFS. You can have one logical drive formatted with FAT, and another logical drive in the same extended partition formatted with NTFS.

Extended partitions are convenient for breaking up a physical disk into more than four logical drives.

Using Disk Administrator

Windows NT includes a useful tool to manage disks after NT has been installed: Disk Administrator. Disk Administrator can help you create, format, and otherwise manage various types of partitions.

caution **It's good practice to use Disk Administrator only during times when no one else is accessing the server. Some of Disk Administrator's functions take a significant amount of time to complete, and some require the server to be rebooted. This means that service to clients during these times can be seriously slowed or interrupted. Plan to perform disk management tasks during nonbusiness hours whenever possible, just as you would other administration tasks that require the server to be down.**

There are two versions of Disk Administrator: the version that ships with Windows NT Workstation, and the version that ships with Windows NT Server. The only difference between the two is the Windows NT Workstation version does not have the Fault Tolerance menu found on the version that ships with Windows NT Server. (Actually, the tool is identical on both operating systems, but Disk Administrator always checks the Registry to determine which NT operating system is being used, and then displays the appropriate menu.)

Figure 3-1 shows a screen shot of the main dialog box of the Windows NT Workstation version of Disk Administrator, and Figure 3-2 shows a screen shot of the main dialog box of the Windows NT Server version of Disk Administrator. Note the Fault Tolerance menu on the NT Server version of the dialog box, which is missing from the dialog box shown for the NT Workstation version.

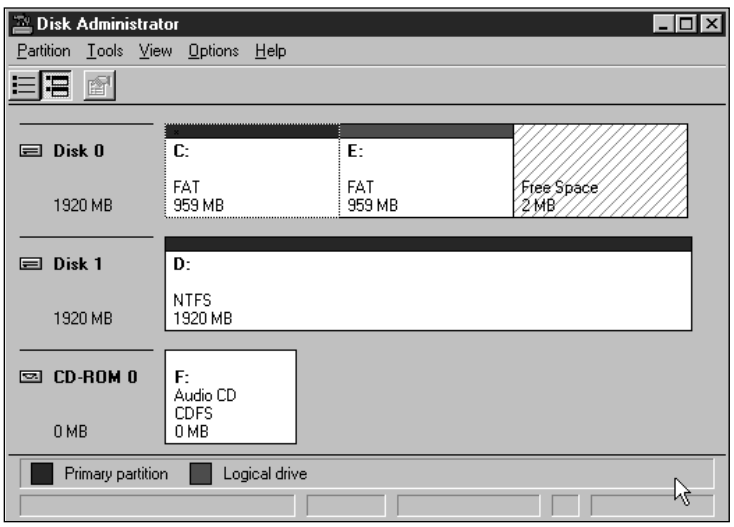


FIGURE 3-1 Windows NT Workstation Disk Administrator

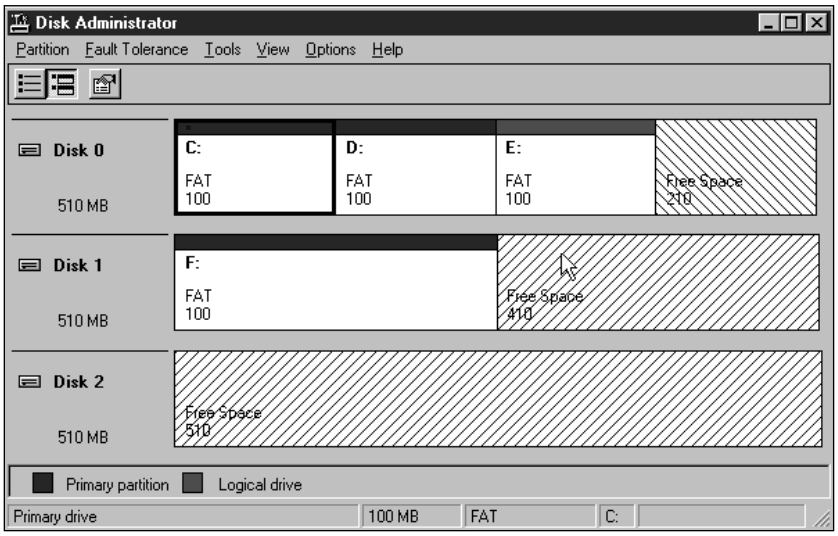


FIGURE 3-2 Windows NT Server Disk Administrator

The next several sections explain how you can use Disk Administrator to create partitions, format partitions, and establish various disk partitioning schemes, including disk mirroring, stripe sets, stripe sets with parity, and volume sets.

Creating partitions

You can use Disk Administrator to create partitions on your computer's hard disk.

I've always wished that some text or course would have given me more precise information and detailed instructions on working with Disk Administrator. It's for this reason that I've presented the instructions to perform several disk and partition management tasks in the rest of this chapter in a step-by-step format.

But this doesn't mean I want you to sit down at your computer and create some partitions right now. Even though the descriptions of how to perform these tasks are presented in a step-by-step format, they're meant to be informational—not actual lab exercises. (You can sink your teeth into Lab 3.3: "Managing Partitions" at the end of this chapter soon enough.) What I really have in mind is for you to read through these steps and refer to them when you have a need to do these various disk management procedures.

Here's a look at the basic steps to create and format a primary partition, and the steps to create an extended partition.

Creating and formatting a primary partition In order to create a primary partition on a hard disk, you must have an area of free space on the hard disk.

The steps in the various how-to sections in the rest of this chapter are meant to be informational, not a lab exercise. (Lab 3.3: "Managing Partitions" is found at the end of this chapter.) If you decide to create a partition on your computer, don't delete or reformat any partition that contains data you don't want to lose. And remember, back up all important data and programs *before* you make any changes to your computer's disk configuration.

The steps I've listed to create a primary partition and an extended partition should work as shown about 90 percent of the time. Don't be disconcerted, though, if an extra dialog box or two is displayed. Any number of hardware or system configuration differences can cause a minor deviation from the steps listed. If a different or extra dialog box is displayed, just provide the appropriate response and continue.

TO CREATE A PARTITION, FOLLOW THESE STEPS:

1. Start Disk Administrator. (Select Start ≥ Programs ≥ Administrative Tools (Common) ≥ Disk Administrator.)
2. If this is the first time you have run Disk Administrator since installing NT, a dialog box appears. If this box appears, click OK to update the system configuration.
3. Click a diagonally striped area of the Disk Administrator dialog box marked Free Space.
4. Select Partition ≥ Create.
5. Depending on the partitions that already exist on your computer's hard disk, a Confirm dialog box may appear stating that "This operation will result in a disk whose partition scheme may not be compatible with MS-DOS." Click Yes if you want to continue.
6. The Create Primary Partition dialog box appears. In the "Create partition of size" text box, enter the size, in MB, that you want the new partition to be. Click OK. The Disk Administrator main dialog box reappears, and the new partition is displayed in a boxed area with the next available drive letter. (If you have several network drives connected, the drive letter

might be H:, even though the next local drive letter would be D:. Don't worry, you can fix this later.) The lettering in the box representing the new partition is gray rather than black like all the other partitions shown.

7. To complete the partition creation process, select Partition ≥ Commit Changes Now.
8. Click Yes in the Confirm dialog box to save the changes to your disk.
9. Click OK in the Disk Administrator dialog box. This causes the Disk Administrator main dialog box to reappear, and your newly created partition appears in full color with black lettering.

After you've created a partition, you'll usually want to format it.

TO FORMAT THE PARTITION YOU JUST CREATED, FOLLOW THESE STEPS:

1. Start Disk Administrator (if the Disk Administrator main dialog box is not already displayed).
2. Click in the boxed area representing the new partition, and then select Tools ≥ Format.
3. The Format dialog box (displayed in Figure 3-3) appears. Configure the characteristics you want this partition to have, including the file system type and volume label. (See the detailed discussion following these steps on the various options that are configured in this dialog box.)
4. When you are finished, click the Start button. A warning message appears, reminding you that formatting will erase all data on your disk. Only click OK if you want to continue.
5. Click OK in the Formatting dialog box that appears after the formatting is complete. Click Close when the Format dialog box reappears.
6. The Disk Administrator main dialog box reappears. A file system type (for example FAT, NTFS, and the like) now appears in the boxed area representing your newly created and formatted partition, directly beneath the drive letter of the new partition.

When you format a partition, you have the option to specify several of its characteristics. Figure 3-3 shows a screen shot of the Format dialog box within Disk Administrator.

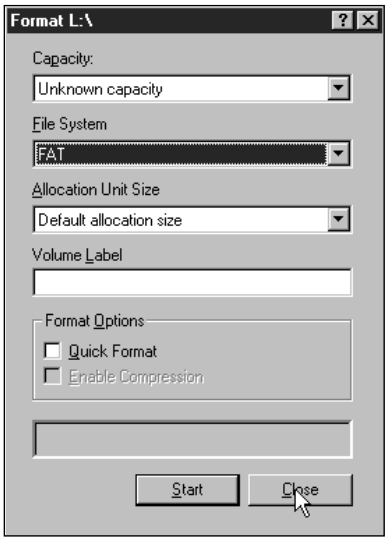


FIGURE 3-3 Formatting a partition

As Figure 3-3 shows, there are several options in the Format dialog box to be configured:

- **Capacity:** This option displays "Unknown capacity" when you format a new partition. (When you format an existing partition, it displays the size of the partition.) This is an informational item only—no configuration is allowed.
- **File System:** The file system choices available are FAT or NTFS.
- **Allocation Unit Size:** Allocation Unit Size refers to the sector size Disk Administrator uses when it formats a partition. If you chose NTFS as the file system type, you can select the sector size to be used during formatting. Sector sizes in this menu vary from 512B to 4,096B.

note **If you want to use a sector size larger than 4,096B, you must format the partition from the command line using the `Format.exe` command. The `Format.exe` command supports sector sizes up to 64KB**

for NTFS partitions. You can type format /? at the command prompt for a complete list of the switches that can be used with the `Format.exe` command. File compression is *not* supported when using sector sizes larger than 4,096B.

If you choose FAT as the file system type, you can't set the Allocation Unit Size. You must accept the default Allocation Unit Size, which varies depending on the size of the partition being formatted.

For additional information on Allocation Unit Size, as well as detailed file system information, see Chapter 17 of the *Microsoft Windows NT Workstation Resource Kit for version 4.0*, published by Microsoft Press.

- **Volume Label:** This option permits you to give your partition a name. Type in the text you want to use for the volume label. Configuring this item is optional—an entry is not required in this box.
- **Format Options:**
  - Selecting the *Quick Format* option instructs NT to write only the necessary data to the disk to support a volume, and not to check for bad sectors during formatting.
  - The *Enable Compression* option is only available if you choose NTFS as the file system. (If you choose FAT as the file system, this check box is grayed out.) Selecting this option causes all files and folders placed in this partition to be compressed by default. You can also set this attribute later by using Windows NT Explorer.

Creating an extended partition This section discusses how you can use Disk Administrator to create an extended partition.

TO USE DISK ADMINISTRATOR (ON EITHER WINDOWS NT WORKSTATION OR WINDOWS NT SERVER) TO CREATE AN EXTENDED PARTITION, FOLLOW THESE STEPS:

1. Start Disk Administrator.
2. Click a diagonally striped, boxed area of the Disk Administrator dialog box marked Free Space.
3. Select Partition ➢ Create Extended.

4. The Create Extended Partition dialog box appears. In the "Create partition of size" text box, enter the size, in MB, that you want the extended partition to be. Click OK.
5. Disk Administrator's main dialog box reappears. Notice that the area of free space is still marked Free Space, but the diagonal lines now run from top left to bottom right, which indicate an extended partition. (Free space that is not an extended partition is indicated by diagonal lines that run from top right to bottom left.) Select Partition ➢ Commit Changes Now.
6. In the Confirm dialog box, click Yes to save the changes to disk. Then click OK to acknowledge that the disks were updated successfully. The Disk Administrator main dialog box reappears.
7. To create a logical drive(s) for this extended partition, select Partition ➢ Create.
8. When the Create Logical Drive dialog box appears, enter the size you want the drive to be, in MB, in the "Create logical drive of size" text box. Click OK.
9. The Disk Administrator main dialog box reappears. Notice that a logical drive box is displayed at the bottom of this dialog box. Select Partition ➢ Commit Changes Now.
10. Click Yes in the Confirm dialog box to save the changes to disk. A dialog box appears indicating that the disks were updated successfully, and reminding you to update your Emergency Repair Disk. Click OK. The Disk Administrator main dialog box reappears. The procedure is complete.

Disk Mirroring

Disk mirroring is a fault tolerance method that enables operations to continue when *one* hard disk fails. (Later in this chapter you'll learn how to recover from a single disk failure.)

The term disk mirroring is somewhat of a misnomer. It should be called partition mirroring. Any partition can be mirrored, including the system partition and the boot partition. In disk mirroring, Disk Administrator makes an exact replica of the partition being mirrored on a *separate* hard disk. You can't make a replica of a partition on the same physical disk—the mirror image must be produced on a different hard disk.

Disk mirroring is fairly expensive, as fault tolerance methods go, because twice the normal amount of disk space is required. However, you get a high level

of fault tolerance for your money. Disk mirroring is used in situations where the integrity of data is more important than minimizing costs. For example, a financial institution might decide that disk mirroring is cost-effective for their company because the extra safety provided by disk mirroring outweighs the cost of additional disk space.

Disk mirroring does not provide fault tolerance in the event of multiple disk failure, and it does not guarantee continued operations if a server goes down. Disk mirroring is also known as RAID level 1. (RAID stands for *Redundant Array of Inexpensive Disks*.)

Disk mirroring is supported by Windows NT Server, but *not* by Windows NT Workstation.

TO CREATE A MIRROR SET, FOLLOW THESE STEPS:

1. Start Disk Administrator, and then select (click) the partition you want to mirror. If you don't have any partitions on your hard disk, or you want to create a new partition to mirror, follow the instructions in the "Creating and formatting a primary partition" section earlier in this chapter to create and format a partition. Then do the steps listed here.
2. Press and hold Ctrl while you click an area of free space on a different hard disk that is at least as large as the partition you want to mirror.
3. Select Fault Tolerance ➪ Establish Mirror. Figure 3-4 displays the Fault Tolerance menu of the Disk Administrator dialog box. Note that Establish Mirror has been selected and highlighted.
4. Disk activity occurs and may continue for quite a while, depending on the size of the partition to be mirrored and the amount of data on that partition. Finally, as Figure 3-5 shows, Disk Administrator displays its main dialog box. Notice that there are now two drives of the same size with the same drive letter, located on different hard disks. The mirrored partitions are highlighted in the color that corresponds to the mirror set box at the bottom of the dialog box. A mirror set has been established.

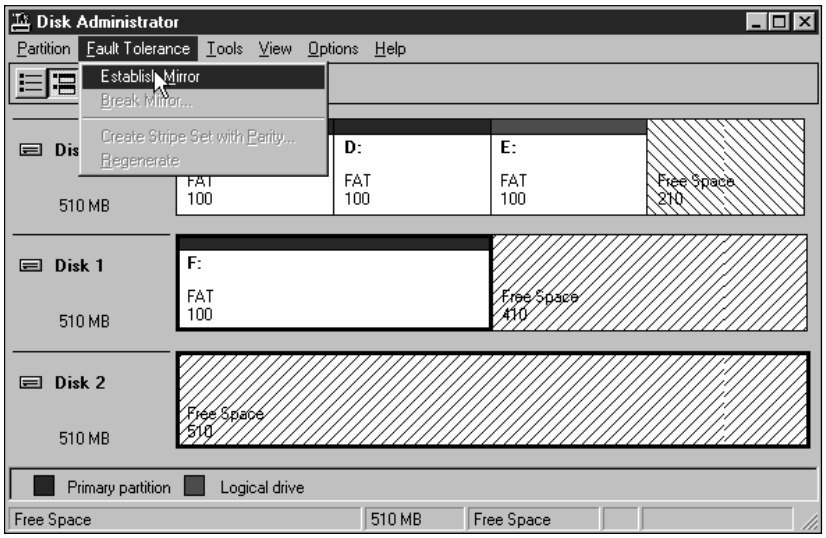


FIGURE 3-4 **Establishing a mirror set**

FIGURE 3-5 **Mirror set established**

Now that a mirror set is established, a fault tolerance boot disk should be created in case the original disk in the mirror set (the disk that was mirrored) fails.

Creating a fault tolerance boot disk

A *fault tolerance boot disk* is a floppy disk that enables you to boot your computer in the event that one of the disks in your computer's mirror set fails. The fault tolerance boot disk should be created *before* the disk failure occurs.

If the original disk in a mirror set fails, and if that disk contains the original boot partition, you will not be able to reboot your computer because the `Boot.ini` file will be pointing at the failed hard disk and partition.

When this happens, you need to use a fault tolerance boot disk that points at the mirrored disk and partition that are still functional to boot your computer.

TO CREATE A FAULT TOLERANCE BOOT DISK, FOLLOW THESE STEPS:

1. Format a 3.5-inch floppy disk by using the Format menu from Windows NT Explorer. This *must* be done in Windows NT, not in Windows 95.
2. Copy the `ntldr`, `Ntdetect.com`, and `Boot.ini` files to the floppy disk. Also copy `Bootsect.dos` to the floppy disk if your computer is configured to dual boot, and copy `Ntbootdd.sys` to the floppy disk if this file exists in the root of your computer's system partition. (An `Ntbootdd.sys` file will exist in your system partition only if you have entries in your `Boot.ini` file that begin with scsi.)
3. Edit the `Boot.ini` file on the floppy disk (not the `Boot.ini` file on your hard drive) to point at the mirrored disk and partition that still function, instead of at the disk and partition that failed. (The next section of this chapter discusses how to edit the `Boot.ini` file.)

Editing the Boot.ini file

The `Boot.ini` file is a read-only file in the root of the active partition on the first hard disk in the computer.

Before you can edit the `Boot.ini` file, you must remove its read-only attribute.

Remember, you are editing the `Boot.ini` file that has been copied to a floppy disk, not the `Boot.ini` file on your hard drive.

You can use Windows NT Explorer to deselect this attribute (which is found in the General tab of the Boot.ini Properties dialog box). After you remove the read-only attribute, you can use any text editor, such as Notepad, to edit the `Boot.ini` file.

However, before you go on to edit this file, you might want to take a closer look at it to understand its structure and syntax. I've reproduced below a sample `Boot.ini` file.

LISTING 3-1 Sample `Boot.ini` file

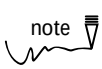
```
[boot loader]
timeout=30
default=multi(0)disk(0)rdisk(1)partition(1)\WINNT
[operating systems]
multi(0)disk(0)rdisk(1)partition(1)\WINNT="Windows NT
     Workstation Version 4.00"
multi(0)disk(0)rdisk(1)partition(1)\WINNT="Windows NT
     Workstation Version 4.00 [VGA mode]" /basevideo /sos
C:\="Microsoft Windows"
```

Note that there are two sections to the `Boot.ini` file: `[boot loader]` and `[operating systems]`.

The first section, `[boot loader]`, contains two entries. The first entry, `timeout`, determines how long, in seconds, the boot loader screen (or boot menu) is displayed when the computer boots. The default timeout is thirty seconds. The second entry, `default`, specifies which operating system loads if no selection is made within the timeout period.

The second section of the `Boot.ini` file, `[operating systems]`, first lists entries consisting of ARC *(Advanced RISC Computing)* pathnames to various operating systems. Only Windows NT uses ARC pathnames in the `Boot.ini` file to indicate which partition, physical disk, and folder contains the files used to start the operating system. Next, the drive letter and path to any other operating systems are listed. The operating system named at the end of each operating systems entry, after the = sign (whether it is an ARC pathname entry or not), is displayed in the boot loader screen.

There are two types of ARC pathname entries: multi and scsi. The terms *multi* and *SCSI* refer to the type of hard disk that is listed in the ARC pathname.



The term *scsi* is normally presented in lowercase letters when it is used to indicate a type of file entry. It is normally presented in uppercase letters when it is used to refer to a disk, adapter, or controller.

All hard disks that can be detected by the computer's BIOS, or by the BIOS on a SCSI adapter, are referred to as *multi*. All hard disks connected to SCSI adapters that do not have their BIOS enabled are referred to as *SCSI*. SCSI disks require a device driver to be loaded before the operating system can access the disk. The Windows NT installation program copies the device driver for a SCSI adapter to the root of the system partition, and renames the file as `Ntbootdd.sys`.

The syntax of operating systems entries that begin with multi is as follows:

`multi(`**`W`**`)disk(`**`X`**`)rdisk(`**`Y`**`)partition(`**`Z`**`)\path`

- **W** is the ordinal number of the adapter. It should always be zero.
- **X** is not used for multi. It is always zero.
- **Y** is the ordinal for the hard disk on the controller. It is always 0 or 1 for disks connected to the primary controller, including SCSI adapters that have their BIOS enabled. It is 0, 1, 2, or 3 on dual channel EIDE controllers.
- **Z** is the partition number. The range of Z is usually 1–4.

The syntax of operating system entries that begin with scsi is as follows:

`scsi(`**`W`**`)disk(`**`X`**`)rdisk(`**`Y`**`)partition(`**`Z`**`)\path`

- **W** is the ordinal number of the adapter.
- **X** is the SCSI ID of the disk.
- **Y** is the logical unit number (LUN) of the disk. It is usually zero.
- **Z** is the partition number. The range of Z is usually 1–4.

Operating systems entries that begin with scsi are typically used in three types of situations:

- When the hard disk containing the system partition is on a SCSI adapter that does *not* have its BIOS enabled
- When the hard disk containing the system partition is on a SCSI adapter *and* has an SCSI ID greater than one
- When the hard disk containing the system partition is on a SCSI adapter *and* there is an IDE or EIDE controller in the system

Now that you understand the structure of the `Boot.ini` file, and the types of entries and syntax used in this file, you're ready to edit it.

Suppose that Figure 3-6 represents a computer that has a newly created mirror set (Disk 1 and Disk 2). The boot partition for this computer is located on the second disk (Disk 1), and uses the drive letter F:. The system partition of this computer (which contains the `ntldr`, `Ntdetect.com`, `Boot.ini`, and `Bootsect.dos` files) is located on the first disk (Disk 0), and uses the drive letter C:. The computer in this example uses a dual-channel EIDE controller.

You want to create a fault tolerance boot disk so you can reboot the computer in the event that Disk 1 fails. You follow the steps outlined in the "Creating a fault tolerance boot disk" section up to the point where you edit the `Boot.ini` file on the floppy disk. The `Boot.ini` file looks like this:

LISTING 3-2 `Boot.ini` **file**

```
[boot loader]
timeout=30
default=multi(0)disk(0)rdisk(1)partition(1)\WINNT
[operating systems]
multi(0)disk(0)rdisk(1)partition(1)\WINNT="Windows NT
    Workstation Version 4.00"
multi(0)disk(0)rdisk(1)partition(1)\WINNT="Windows NT
    Workstation Version 4.00 [VGA mode]" /basevideo /sos
C:\="Microsoft Windows"
```

FIGURE 3-6 **New mirror set**

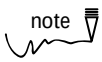
In this situation, you must edit the `Boot.ini` file so it points to Disk 2. (The `Boot.ini` file must point to Disk 2 because if Disk 1 fails, Disk 2 will be the only disk in the mirror set that still works.) You remove the `Boot.ini` file's read-only attribute and use a text editor, such as Notepad, to edit the file. The edited version of the `Boot.ini` file on your newly created fault tolerance boot disk is presented here.

LISTING 3-3 `Boot.ini` **file edited for fault tolerance boot disk**

```
[boot loader]
timeout=30
default=multi(0)disk(0)rdisk(2)partition(1)\WINNT
[operating systems]
multi(0)disk(0)rdisk(2)partition(1)\WINNT="Windows NT
    Workstation Version 4.00"
multi(0)disk(0)rdisk(2)partition(1)\WINNT="Windows NT
    Workstation Version 4.00 [VGA mode]" /basevideo /sos
C:\="Microsoft Windows"
```

Notice that the ARC pathnames in the operating systems section now point to Disk 2 instead of Disk 1.

Finally, there are several switches you can add at the end of an operating systems entry in the `Boot.ini` file. Table 3-2 lists and describes these switches.



These switches are not typically used when creating a fault tolerance boot disk. However, because they are used frequently during troubleshooting, and because this is the only section of this book that addresses the `Boot.ini` file in detail, I've covered the switches here for completeness.

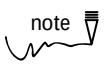
TABLE 3-2 BOOT.INI FILE SWITCHES

SWITCH	*DESCRIPTION*
`/BASEVIDEO`	This switch causes the computer to use the standard VGA driver when it starts, and is useful in troubleshooting video driver problems.
`/BAUDRATE=nnnn`	This switch specifies the baud rate used during debugging, and includes all of the functionality of the `/DEBUG` switch.
`/CRASHDEBUG`	This switch forces the debugger to load in an inactive state until an error occurs.

SWITCH	DESCRIPTION
/DEBUG	This switch causes the debugger to be loaded. It can be activated by another computer connected to this computer by a modem or null-modem cable.
/DEBUGPORT=comx	This switch specifies which COM port the debugger uses, and includes all of the functionality of the /DEBUG switch.
/MAXMEM:n	This switch specifies the maximum amount of memory that Windows NT can use. It is useful for troubleshooting memory problems.
/NODEBUG	This switch specifies that the debugger will not run, and that no debugging information will be generated.
/NOSERIALMICE=COMx|COMx,y,z...|	This specifies that the indicated serial port will not be tested for the presence of a mouse. Use this switch if you have an *uninterruptible power supply* (UPS) or some other device connected to a serial port.
/SOS	This switch provides a verbose listing of each device driver as it is loaded during the boot sequence. It is useful for troubleshooting device drivers.

Stripe Sets

In a *stripe set*, which is made up of two to thirty-two hard disks, data is stored, a block at a time, evenly and sequentially among all of the disks in the set. Stripe sets are sometimes referred to as disk striping. *Disk striping* alludes to the process wherein a file is written, or striped, one block at a time; first to one disk, then to the next disk, and then to the next disk, and so on, until all of the data has been evenly distributed among all of the disks.



Neither the boot nor the system partition can be on a stripe set.

A stripe set is accessed by using a single drive letter, as if all of its disks were combined into a single drive. A stripe set is created from identical amounts of free space on each of the disks that belong to the set.

Stripe sets provide faster disk access than volume sets or large individual hard disks because the stripe sets store a single file across multiple disks. The various pieces of the file can be read nearly simultaneously from the multiple disks, thus increasing performance. Access speed is the primary advantage and common reason for using a stripe set. The tradeoff or downside to using a stripe set is that

the potential disk failure rate is increased because there are more possible points of failure when a file is accessed across several disks.

Stripe sets have no additional cost associated with them because they use the same amount of disk space in which that data would normally be stored. However, stripe sets do not provide any fault tolerance. If one partition or disk in a stripe set fails, all data on the stripe set is lost.

A stripe set (or disk striping) is also known as RAID level 0.

Windows NT Server and Windows NT Workstation both support stripe sets.

TO CREATE A STRIPE SET, FOLLOW THESE STEPS:

1. Start Disk Administrator.
2. Click an area of free space on one of your computer's hard disks.
3. Press and hold Ctrl while you click one or more additional areas of free space on other hard disks. You can't select more than one area of free space per physical disk.
4. Select Partition ➢ Create Stripe Set.
5. The Create Stripe Set dialog box is displayed, as shown in Figure 3-7. Note that it shows the minimum and maximum amount of total space that can be used for the stripe set. The maximum total size is the amount of free space from the smallest area of free space you selected, multiplied by the number of disks you selected. (Remember, in stripe sets, the size of the free space areas on the disks used in the set are identical.) You can either accept this number or type in a smaller value in the "Create stripe set of total size" text box. Click OK to create the stripe set.
6. The Disk Administrator main dialog box reappears. Select Partition ➢ Commit Changes Now. Click Yes to save the changes to disk. Then click OK to acknowledge that the disks were updated successfully.
7. The Disk Administrator main dialog box reappears. Figure 3-8 shows the stripe set at this point. Note that the stripe set is created and uses the drive letter H: on Disks 1, 2, and 3, but it still needs to be formatted.
8. Follow the directions in the "Creating and formatting a primary partition" section earlier in this chapter to format the stripe set. The stripe set is now complete.

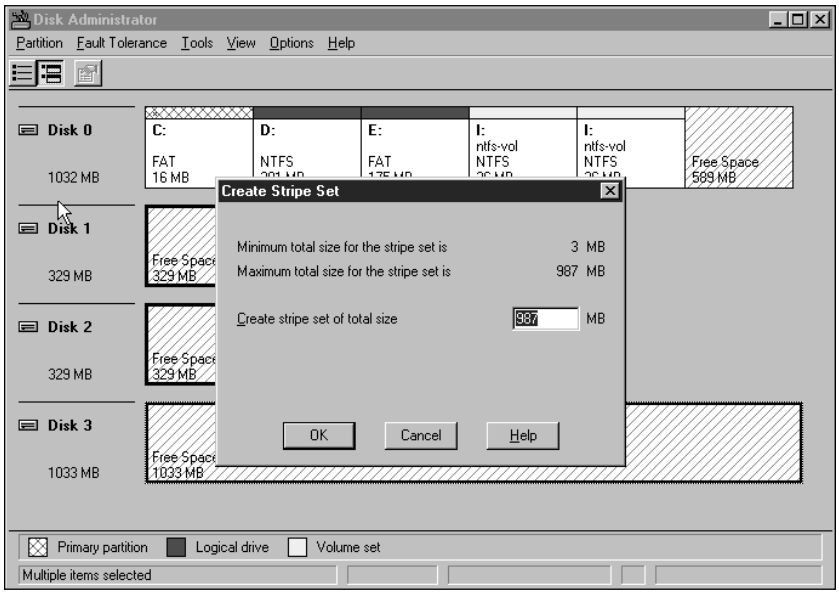


FIGURE 3-7 **Creating a stripe set**

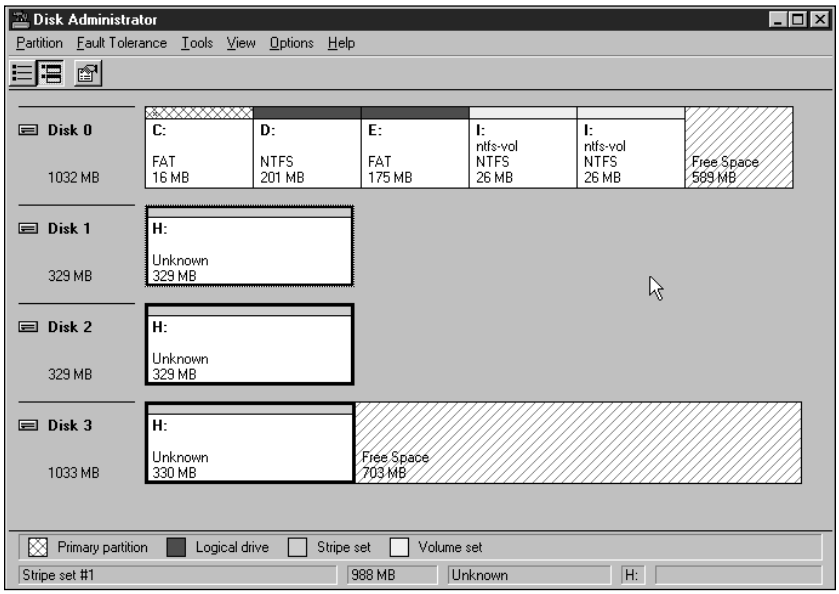
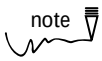


FIGURE 3-8 **Unformatted stripe set**

Stripe Sets with Parity

A *stripe set with parity* is similar to a stripe set, but it provides a degree of fault tolerance which the stripe set cannot. In a stripe set with parity, data is not only distributed a block at a time, evenly and sequentially among all the disks in the set, but parity information is also written across all of the disks in the set.



Neither the boot nor the system partition can be on a stripe set with parity.

A stripe set with parity is made up of three to thirty-two hard disks. Like stripe sets, stripe sets with parity are created from identical amounts of free space on each disk that belongs to the set.

Striping with parity, in my experience, is the most common method of fault tolerance. It is less costly than disk mirroring (because data is not replicated on another disk), is faster than disk mirroring, and provides a modest level of data safety.

A stripe set with parity provides the same read performance as a stripe set, but its write performance is a little slower.

If a single disk in a stripe set with parity fails, the parity information contained in the other disks is used to regenerate the data from the failed disk. (Recovering from a single disk failure in a stripe set with parity is discussed later in this chapter.) You cannot recover your data in a stripe set with parity if more than one disk fails.

Stripe sets with parity (sometimes called *striping with parity*) are also known as RAID level 5.

Stripe sets with parity are supported by Windows NT Server, but not by Windows NT Workstation.

TO CREATE A STRIPE SET WITH PARITY, FOLLOW THESE STEPS:

1. Start Disk Administrator.
2. Click an area of free space on one of your computer's hard disks.
3. Press and hold Ctrl while you click two or more additional areas of free space on other hard disks. You can't select more than one area of free space per physical disk.

4. Select Fault Tolerance ➢ Create Stripe Set with Parity.
5. The Create Stripe Set with Parity dialog box is displayed showing the minimum and maximum amount of space that can be used for the stripe set. The maximum amount is the amount of free space from the smallest area of free space you selected, multiplied by the number of disks you selected. You can either accept this number or type in a smaller value in the “Create stripe set of total size” text box. Click OK to create the stripe set with parity.
6. The Disk Administrator main dialog box reappears. Select Partition ➢ Commit Changes Now. Click Yes to save the changes to disk. Then click OK to acknowledge that the disks were updated successfully. At this point, the stripe set with parity is created, but still needs to be formatted.
7. To format the stripe set with parity, follow the directions presented earlier in this chapter in the “Creating and formatting a primary partition” section.
8. At the conclusion of the formatting process, Disk Administrator performs additional functions on the newly created stripe set with parity. The Disk Administrator main dialog box reappears. Click any disk that is part of the stripe set with parity to view the status of this process. As Figure 3-9 shows, Disk Administrator indicates that it is INITIALIZING the stripe set with parity. Notice that the stripe set with parity uses the drive letter F: and consists of Disks 1, 2, and 3.

   The initializing process takes quite a while. (It may take several minutes to an hour or more, depending on the size of the stripe set with parity.)

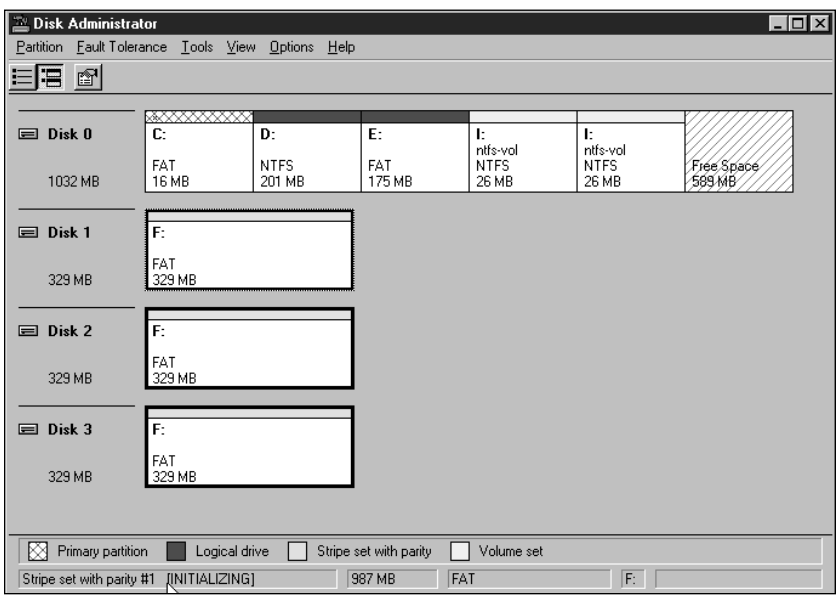


FIGURE 3-9 Stripe set with parity (INITIALIZING)

9. When the initializing process is finally complete, the Disk Administrator main dialog box changes. Figure 3-10 shows the completed stripe set with parity. Notice at the bottom of the dialog box the stripe set with parity is called HEALTHY.

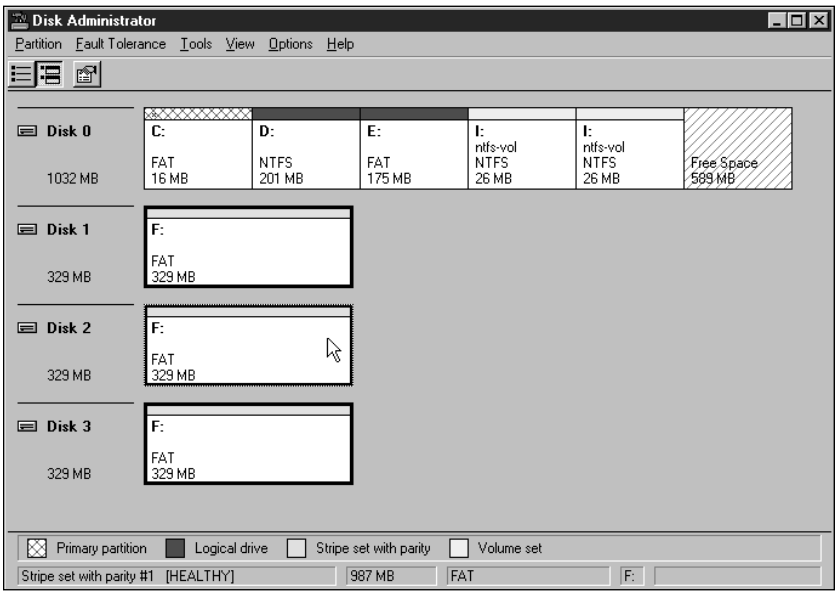


FIGURE 3-10 **Stripe set with parity (HEALTHY) established**

Volume Sets

A *volume set* is a combination of free space areas over two to thirty-two hard disks that is treated as a single drive. (The free space areas do not need to be of identical size.)

note **Neither the boot nor the system partition can be on a volume set.**

The primary purpose and use of a volume set is to access disk space on more than one hard disk by using a single drive letter. A volume set is sometimes used when a drive becomes full and you want to enlarge its capacity.

A volume set is similar to a stripe set. However, a file in a volume set is usually fully contained on a single hard disk, instead of being striped over multiple hard disks.

Volume sets do not involve additional cost, because no additional hard disks are required over the number normally needed to store data.

caution **Volume sets, like stripe sets, do not perform any fault tolerance function. If one disk in a volume set fails, all data on the volume may be lost, because Windows NT can't access data unless all of the disks that make up the volume set are functional.**

Volume sets are said to be *created* when areas of free space only (not existing volumes) are combined into a volume set. Volume sets are said to be *extended* when an existing NTFS partition is enlarged.

in the real world **It's been my experience that extending a volume set is far more common than creating one. It's one way to approach a situation when a drive is filled to capacity and you still need additional space in that volume.**

Windows NT Server and Windows NT Workstation both support volume sets. The following sections explain how to create and extend volume sets.

TO CREATE A VOLUME SET, FOLLOW THESE STEPS:

1. Start Disk Administrator.
2. Click an area of free space on one of your computer's hard disks.
3. Press and hold Ctrl while you click one or more additional areas of free space on other disks.
4. Select Partition ⩾ Create Volume Set.
5. The Create Volume Set dialog box is displayed showing the minimum and maximum amount of space that can be used for the volume set. Enter the size you want the volume set to be in the "Create volume set of total size" text box. Click OK to create the volume set.
6. The Disk Administrator main dialog box reappears. Select Partition ⩾ Commit Changes Now. Click Yes to save the changes to disk. Then click OK to acknowledge that the disks were updated successfully.
7. The Disk Administrator main dialog box reappears. Figure 3-11 shows the status of the volume set at this point. Note that the volume set, which uses the drive letter F: over Disks 1, 2, and 3 is created, but still needs to be formatted.

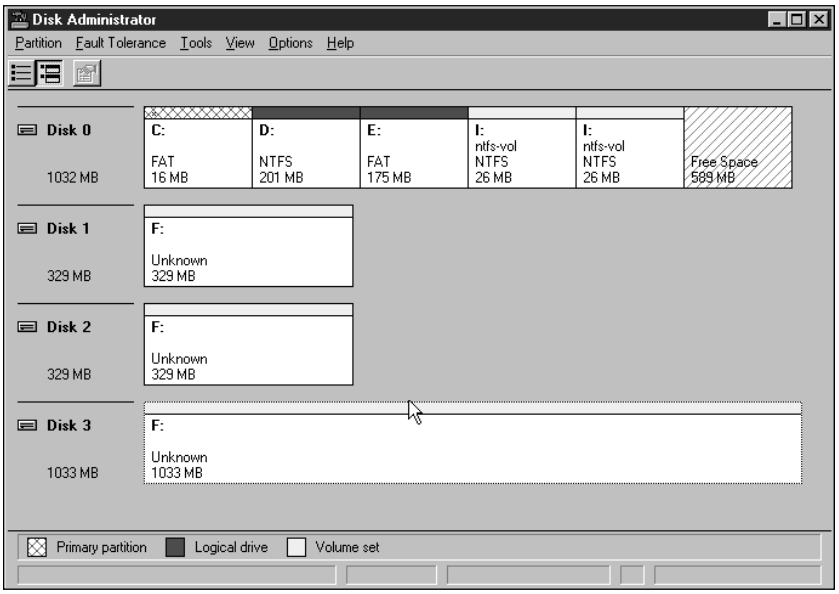


FIGURE 3-11 **Unformatted volume set**

8. To format the volume set, follow the directions from the "Creating and formatting a primary partition" section earlier in this chapter. Figure 3-12 displays the completed, formatted volume set.

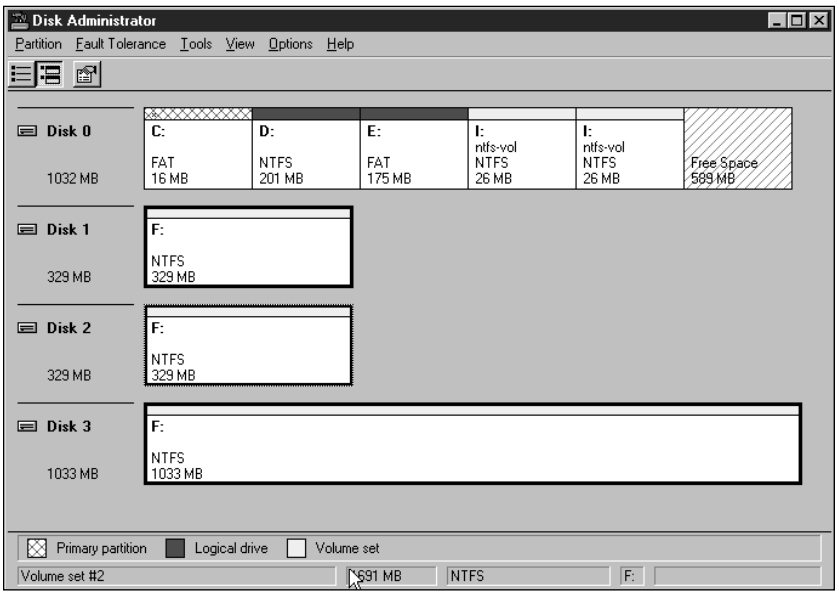


FIGURE 3-12 **Completed, formatted volume set**

TO EXTEND A VOLUME SET, FOLLOW THESE STEPS:

1. Start Disk Administrator.
2. Click an NTFS partition on one of your computer's hard disks. (Only NTFS volumes and volume sets can be extended.)
3. Press and hold Ctrl while you click one or more additional areas of free space on other disks.
4. Select Partition ➢ Extend Volume Set.
5. The Extend Volume Set dialog box is displayed showing the minimum and maximum amount of space that can be used for the total volume set. Enter the size you would like the total volume set to be in the "Create volume set of total size" text box. Click OK to extend the volume set.
6. The Disk Administrator main dialog box reappears. The volume set is now extended. You don't have to format extensions to volume sets.
7. Exit Disk Administrator and reboot your computer to make the extended volume set effective.

Comparison of Disk Partitioning Schemes

At this point, you've had a chance to examine four different disk partitioning schemes: disk mirroring, stripe sets, stripe sets with parity, and volume sets.

Table 3-3 compares the fault tolerance, cost, and access speed provided by these four disk partitioning schemes. Note that the most expensive scheme, disk mirroring, also provides the highest level of fault tolerance of the methods listed. As with most things in life, you get what you pay for.

TABLE 3-3 COMPARISON OF DISK PARTITIONING SCHEMES

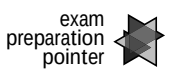
DISK PARTITIONING SCHEME	*FAULT TOLERANCE*	*COST*	*ACCESS SPEED*
Disk mirroring	High	High	Normal (Slowest)
Stripe sets with parity	Medium	Medium	Medium-Fast
Stripe sets	Low (None)	Low	Fastest
Volume sets	Low (None)	Low	Normal (Slowest)

Now that you're familiar with the various disk partitioning schemes, it's appropriate to discuss how to recover from disk failure when each of these schemes is used.

Recovering from Disk Failure

So what do you do when it all comes crashing down—when the remote possibility of disk failure that you planned for, but never thought would actually happen, is a painful reality?

This section provides information on how to recover from a single or multiple hard disk failure. Specifically, you'll learn how to recover from disk failure in situations where disk mirroring, stripe sets, stripe sets with parity, and volume sets are used.



Both the Server and the Enterprise exams have an objective on recovering from a fault tolerance failure. Because you are unlikely to get a lot of practice at this in real life, I recommend that you study this section carefully, and revisit it just before you take the exams.

In some cases your disk configuration may enable you to continue operations (but without any fault tolerance) until you can replace the failed hard disk and restore your fault tolerance configuration.

In cases of stripe set, volume set, or multiple hard disk failure, you must repair the hardware and restore your data from tape to continue operations. If you don't have a tape backup in these situations, Windows NT will ***not*** be able to recover your data.

Tape backup is critically important. For more information on data backup and restoration, see Chapter 15.

When you have a disk failure (or a multiple disk failure in the case of disk mirroring or a stripe set with parity), and you don't have a tape backup, you might consider contacting a third-party data recovery service if the data is extremely important or valuable to you. The data recovery service may be able to retrieve some of your data from the failed disk. Be forewarned, however, that this process is expensive and takes time to complete.

The next several sections explain the detailed steps you can take to recover from disk failure in situations where disk mirroring, stripe sets, stripe sets with parity, and volume sets are involved.

Disk Mirroring

Sometimes a disk that is part of a mirror set fails. Here's what you can do if that happens.

The steps in this and other how-to sections in this book are meant to be informational, not a lab exercise. Don't delete or reformat any partition that contains data that you don't want to lose. And remember, back up all important data and programs before you make any changes to your computer's configuration.

TO RESTORE YOUR FAULT TOLERANCE DISK CONFIGURATION AFTER ONE DISK IN A MIRROR SET FAILS, FOLLOW THESE STEPS:

1. Boot the computer (by using the fault tolerance boot disk if needed) and then start Disk Administrator.
2. Click the disk in the mirror set that is still operational.
3. Select Fault Tolerance ➢ Break Mirror. Click Yes in the Confirm dialog box to end mirroring and to create two independent partitions.
4. The Disk Administrator main dialog box reappears. Select Partition ➢ Commit Changes Now.
5. In the Confirm dialog box, click Yes to save the changes.
6. Disk Administrator indicates that the disks were updated successfully. Click OK.
7. The Disk Administrator main dialog box reappears. Exit Disk Administrator, shut down the computer, and replace the failed hard disk.
8. Restart the computer (by using the fault tolerance boot disk if needed) and then start Disk Administrator.
9. Click the partition that you broke the mirror on in Step 3. (This is the good [non-failed] partition that has your data on it.)
10. Press and hold Ctrl while you click an area of free space on the new hard disk that is at least as large as the partition you want to mirror.
11. From the Fault Tolerance menu, select Establish Mirror.

12. The Disk Administrator main dialog box reappears. Disk activity occurs and may last for some time, depending on the size of the partition to be mirrored and the amount of data on that partition. Your mirror set is now reestablished.

Stripe Sets

Recovering from a failed disk or disks in a stripe set is fairly straightforward.

If you don't have a tape backup of the files on the stripe set, Windows NT can't recover your data. If you have a tape backup, first delete the existing stripe set, and then create a new stripe set. Follow the steps below to recover from a stripe set disk failure.

TO REPLACE THE FAILED DISK AND DELETE THE EXISTING STRIPE SET (INCLUDING ALL REMAINING DATA ON THE DISKS IN THE SET), FOLLOW THESE STEPS:

1. Replace the failed hard disk or disks. Reboot the computer.
2. Start Disk Administrator.
3. Click any of the good (non-failed) partitions in the stripe set. (Clicking any good partition in the stripe set highlights all remaining partitions in the set.) Select Partition ➢ Delete. Click Yes in the Confirm dialog box to delete the partition.
4. The Disk Administrator main dialog box reappears. Select Partition ➢ Commit Changes Now.
5. In the Confirm dialog box, click Yes to save the changes.
6. Disk Administrator indicates that the disks were updated successfully. Click OK. The Disk Administrator main dialog box reappears.

THEN, TO CREATE A NEW STRIPE SET (THAT WILL INCLUDE THE NEW DISK) FOLLOW THESE STEPS:

1. To create a new stripe set that will include the new disk, follow the steps presented earlier in this chapter for creating a stripe set.
2. Restore all data from tape.

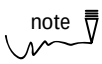
Stripe Sets with Parity

If you have more than one failed hard disk in a stripe set with parity, follow the steps for recovering from a disk failure in a stripe set (in the previous section). If a single hard disk in your stripe set with parity fails, the following steps explain how to recover.

TO RECOVER FROM A *SINGLE* HARD DISK FAILURE IN A STRIPE SET WITH PARITY, FOLLOW THESE STEPS:

1. Replace the failed disk and reboot the computer.
2. Start Disk Administrator.
3. Click any portion of the existing stripe set with parity. Press and hold Ctrl while you click the free space area on the new hard disk.
4. Select Fault Tolerance ➢ Regenerate.
5. The Disk Administrator main dialog box reappears. Click any disk that is part of the stripe set with parity. A significant amount of disk activity occurs as the computer regenerates the data on the new hard disk.

   During the regeneration process, various messages are displayed at the bottom of the Disk Administrator dialog box, including: RECOVERABLE, REGENERATING, and HEALTHY. The process is complete when HEALTHY is displayed.



Sometimes it is necessary to reboot the computer during Step 5. For example, if Disk Administrator displays RECOVERABLE for a long period of time (ten minutes or more), you might try rebooting the computer to start the regeneration process.

Volume Sets

Recovering from a failed disk or disks in a volume set is fairly straightforward.

If you don't have a backup of the files on the volume set, Windows NT can't recover your data. If you have a tape backup, follow these steps to recover from the disk failure.

FIRST, TO DELETE THE EXISTING VOLUME SET (AND ALL REMAINING DATA ON THE DISKS IN THE SET), FOLLOW THESE STEPS:

1. Replace the failed hard disk or disks. Reboot the computer.
2. Start Disk Administrator.
3. Click any of the good (non-failed) partitions in the volume set. (Clicking any partition in the volume set highlights all remaining partitions in the volume set.) Select Partition ➢ Delete. Click Yes in the Confirm dialog box to delete the partition.
4. The Disk Administrator main dialog box reappears. Select Partition ➢ Commit Changes Now.
5. In the Confirm dialog box, click Yes to save the changes.
6. Disk Administrator indicates that the disks were updated successfully. Click OK. The Disk Administrator main dialog box reappears.

NEXT, CREATE A NEW VOLUME SET (THAT WILL INCLUDE THE NEW HARD DISK) BY FOLLOWING THESE STEPS:

1. To create a new volume set that will include the new hard disk, follow the steps presented earlier in this chapter for creating a volume set.
2. Restore all data from tape.

Updating the Emergency Repair Disk

The *Emergency Repair Disk* is a floppy disk used to restore the Windows NT Registry to the configuration that existed when the Emergency Repair Disk was created (or updated).

If you have made any changes to your computer's system configuration since the Emergency Repair Disk was created or last updated, those changes will be lost during the emergency repair process. For this reason, you should update your Emergency Repair Disk *every time* you make a change to your computer's system configuration, including any changes to your disk configuration. Windows NT will prompt you to update your Emergency Repair Disk every time you make a change

to your disk configuration in Disk Administrator. (You'll have a chance to update your Emergency Repair Disk in Lab 3.4 at the end of this chapter.)

concept link

This chapter only discusses updating the Emergency Repair Disk. Chapter 27 explains how to use the Emergency Repair Disk to restore the Registry.

The Emergency Repair Disk is initially created during the installation of Windows NT. To update it at any time after installation (or to create an Emergency Repair disk after installation if one was not created at that time), you must use the `Rdisk.exe` utility.

You won't find the `Rdisk.exe` utility in the Start menu. You will either need to run it from the command line, run it by selecting the Run option from the Start menu, or you can create a shortcut for it on your desktop.

When you run `Rdisk.exe`, a Repair Disk Utility dialog box is displayed. Figure 3-13 shows the first Repair Disk Utility dialog box.

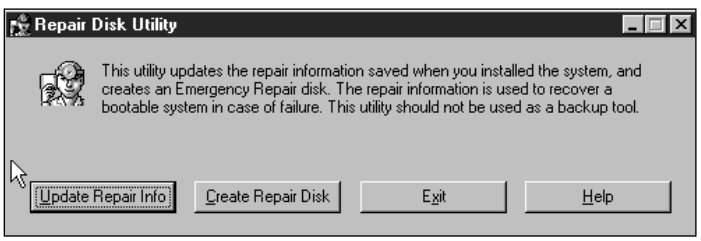


FIGURE 3-13 The first Repair Disk Utility dialog box

If you click the Update Repair Info command button in the Repair Disk Utility dialog box (and click Yes in the next dialog box displayed to confirm), `Rdisk.exe` will save all of your Registry (with the exception of the Security hive and the Security Accounts Manager [SAM] database) to the `<winntroot>\repair` directory. `<winntroot>` is the directory that was specified for the installation of Windows NT, usually `C:\winnt`.

concept link

For additional information on the Windows NT Registry, see Chapter 27, as well as the *Microsoft Windows NT Workstation Resource Kit for version 4.0*, Part V, "Windows NT Registry."

After `Rdisk.exe` saves your Registry configuration information, it will ask if you want to create an Emergency Repair Disk. Figure 3-14 displays this Repair Disk Utility dialog box. Click Yes to create the Emergency Repair Disk.

FIGURE 3-14 Creating (Updating) the Emergency Repair Disk

note **Don't be thrown because `Rdisk.exe` asks if you want to create an Emergency Repair Disk when what you're really trying to do is to update it. `Rdisk.exe` uses the same process and terminology for both.**

At this point, Repair Disk prompts you to insert a labeled 3.5-inch floppy disk into drive A:. If you have the Emergency Repair Disk you created during the installation of Windows NT, use this disk to create an updated Emergency Repair Disk. If you didn't create an Emergency Repair Disk during installation, insert any blank floppy disk to create an Emergency Repair Disk now.

A second way you can update your Emergency Repair Disk is to use the one command line switch that can be used with `Rdisk.exe`, the `/S` switch. The *s* stands for security. Using the `/S` switch causes the entire Registry, including the Security hive and the SAM database, to be backed up. When you run `Rdisk.exe` with the `/S` switch, `Rdisk.exe` automatically begins saving your Registry information to the repair directory, and then prompts you to create the Emergency Repair Disk.

tip **It's a good idea to use the `Rdisk.exe` utility with the `/S` switch. This will provide you with a backup of the SAM database in addition to the one on your regularly scheduled tape backup.**

tip **Another good idea is to write down the Administrator's password when you create a new or updated Emergency Repair Disk. It's very important to know what the Administrator's password is after you have restored the Registry from the Emergency Repair Disk. Personally, I go so far as to write the Administrator's password on the Emergency Repair Disk label.**

To use the `Rdisk.exe` utility with the `/S` switch, select the Run option from the Start menu, and then type **rdisk /s** in the text box within the Run dialog box.

Key Point Summary

This chapter introduced the three file systems that Windows NT supports, as well as partitions and how to work with them, how to recover from disk failure, and how to update the Emergency Repair Disk.

- Windows NT supports the *file allocation table* (FAT) *file system*, the *Windows NT file system* (NTFS), and the *Compact Disc Filing System* (CDFS).
  - The *FAT file system* does not support file and folder security in Windows NT. The FAT file system supports the use of long filenames, and normally permits faster access to files on partitions smaller than 500MB than other file systems. The maximum size of a FAT partition is 4GB. The FAT file system does not support file compression.
  - *NTFS* is the most powerful file system supported by Windows NT. NT is the only operating system that supports NTFS. NTFS provides file and folder security. NTFS, like FAT, supports the use of long filenames. NTFS usually provides faster access to files stored on a large partition that contains a large number of files than the FAT file system. The maximum functional size of an NTFS partition is two terabytes. NTFS supports file compression, and is a highly reliable, recoverable file system. You can use the `Convert.exe` command to convert an existing FAT partition into an NTFS partition. This is a one-way conversion. It is much more difficult to convert an NTFS partition into a FAT partition.
- Marking a disk to identify which parts contain a file system is called *partitioning*. A hard disk can be separated into a maximum of four partitions. Windows NT supports two types of partitions: *primary* and *extended*. The Windows NT 4.0 *system partition* must be located on the active primary partition on the first hard disk in a computer. The *boot partition* can be located on either a primary or extended partition. A hard

disk can have more than one primary partition on it, but can have only one extended partition on it.

- *Disk Administrator* is the Windows NT tool that is used to create, format, and otherwise manage partitions. This chapter includes detailed steps explaining how to use Disk Administrator for creating partitions, formatting partitions, establishing disk mirroring, creating a fault tolerance boot disk (including editing the `Boot.ini` file), and creating stripe sets, stripe sets with parity, and volume sets. Table 3-3 compares the levels of fault tolerance, cost, and access speed provided by four common disk partitioning schemes: *disk mirroring*, *stripe sets with parity*, *stripe sets*, and *volume sets*. Of the four schemes, disk mirroring is the most expensive and provides the highest level of fault tolerance. Stripe sets and volume sets are inexpensive, but neither provide any fault tolerance.
- Specific steps on recovering from single and multiple disk failure in instances of disk mirroring, stripe sets, stripe sets with parity, and volume sets are presented. Tape backup is critically important to this process.
- The `Rdisk.exe` utility is used to update the Emergency Repair Disk. The *Emergency Repair Disk* is used to restore the Windows NT Registry to the configuration that existed when the Emergency Repair Disk was created (or updated). The Emergency Repair Disk should be updated *every time* you make a change to your computer's system or disk configuration. `Rdisk.exe` can be run with the `/S` switch to backup the Security hive and the SAM database.

Applying What You've Learned

Now it's time to regroup, review, and apply what you've learned in this chapter.

The questions in the following Instant Assessment section bring to mind key facts and concepts.

The two exercises in the Review Activities section focus on planning, and give you a chance to apply your knowledge of disk management to real-world situations.

The hands-on lab exercises will really reinforce what you've learned, and give you an opportunity to practice some of the tasks tested by the Microsoft Certified Professional exams.

Instant Assessment

1. What are the three file systems supported by Windows NT 4.0?
2. If you want your computer to dual boot between Windows NT 4.0 and Windows 95, which file system must your computer's first partition use?
3. Which is easier to accomplish: converting a FAT partition to NTFS, or converting an NTFS partition to a FAT partition?
4. When a computer boots, from which partition on which disk does it attempt to *begin* the process of loading the operating system?
5. When booting from the hard disk, on which partition must the Windows NT 4.0 system partition be located?
6. How many primary partitions can you have on one disk? How many extended partitions? How many total partitions?
7. Which Windows NT tool is used to create, format, and otherwise manage partitions?
8. Which of the four disk partitioning schemes involves making an exact replica of a partition onto a separate hard disk?
9. When is a fault tolerance boot disk used?
10. What two sections make up the `Boot.ini` file?
11. Which disk partitioning scheme involves the distribution of data, a block at a time, evenly and sequentially, among all of the disks in the set?
12. Which of the four disk partitioning schemes discussed provide no fault tolerance (that is to say, if one disk fails, all data is lost)?
13. Which of the four disk partitioning schemes is the most commonly used method of fault tolerance? Why?
14. Which of the four disk partitioning schemes is the most expensive and provides the highest amount of fault tolerance?
15. Which utility can you use to update your Emergency Repair Disk?
16. After starting Disk Administrator, what's the first main step in recovering from a single disk failure when the disk is part of a mirror set?

T/F

17. The FAT file system supports file and folder security in Windows NT. _____
18. The FAT file system supports the use of long filenames. _____
19. The FAT file system supports file compression. _____
20. NTFS provides file and folder security in Windows NT. _____
21. NTFS supports the use of long filenames. _____
22. NTFS does not preserve upper- and lowercase in filenames. _____
23. The minimum functional size of an NTFS partition is 2 terabytes. _____
24. NTFS supports a compression attribute for each file. _____
25. NTFS is a highly reliable, recoverable file system. _____
26. Using NTFS increases the likelihood of fragmentation on partitions. _____
27. You can use NTFS to format a floppy disk. _____

For answers to the Instant Assessment questions see Appendix D.

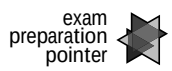
Review Activities

The following activities test your understanding of disk configuration topics in two different planning exercises.

Workstation
Server
Enterprise

File system planning exercise

Planning involves considering the facts and needs of a given situation *before* you make a choice or commitment to a particular path.



Planning is not only critically important in the real world, it's also tested on the Microsoft Certified Professional exams. In fact, it's a large part of the exam objectives. So, try to start thinking about planning situations, not just as you do this exercise, but throughout the rest of your study and preparation for the exams.

For each of the following scenarios, consider the given facts and any expressed needs or goals. Then choose the most appropriate file system for each scenario.

Problem 1 You want to run Windows NT Workstation on a computer. Your computer's hard disk has one 480MB partition and one 20MB partition. Each

folder on the partitions contains a moderate amount of files. You want to optimize access speed to the files. Which file system should you choose?

Problem 2 You want your new computer to dual boot between Windows NT Server and MS-DOS. Your computer's 1GB hard drive has only one partition on it. Which file system should you choose?

Problem 3 Your computer has two hard disks, and each hard disk has a 4GB partition. There are a large number of files in the folders on these partitions. You want to run Windows NT on this computer and optimize access speed to the files. Which file system should you choose?

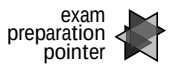
Problem 4 You plan to install and run Windows NT Server on a computer. Your supervisor informs you that file and folder security are required because of highly sensitive data that is stored on this computer. You also need to be able to utilize file compression because of the amount and size of files on this computer. Which file system should you choose?

For answers to the Review Activity see Appendix D.

Disk partitioning and fault tolerance planning exercise

Server
Enterprise

This chapter introduced four disk partitioning schemes: disk mirroring, stripe sets with parity, stripe sets, and volume sets.



Both the Server and Enterprise exams test your ability to plan the disk drive configuration for various requirements, including choosing a fault tolerance method.

I know this seems like a lot of stuff to memorize, but for experienced network professionals, making these decisions is almost second nature. It should also be second nature to you by the time you take the exams!

For each of the following scenarios, consider the given facts and any expressed criteria that must be met. Then choose the most appropriate disk partitioning scheme for each scenario.

(Hint: As you may recall, Table 3-3 compares the levels of fault tolerance, cost, and access speed provided by the four schemes.)

Problem 1 You are in charge of planning a new server for your company. Management tells you that protection of data is the most important requirement for this new server, more important than speed and even more important than cost. Which disk partitioning scheme should you choose?

Problem 2 You are planning a new server for your company's network. The server will contain a large database. Speed of access to files on the database is your primary concern. Fault tolerance is not important, but you do need to minimize costs. Which disk partitioning scheme should you choose?

Problem 3 You are planning to add a new server to your organization's network. You want to achieve a moderate amount of fault tolerance *and* minimize costs. Which disk partitioning scheme should you choose?

For answers to the Review Activity see Appendix D.

Hands-on Lab Exercises

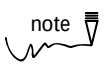
The following hands-on lab exercises provide you with two different opportunities to use the disk management knowledge you've gained in this chapter.

Lab 3.3 *Managing Partitions*

Workstation
Server
Enterprise

The objective of this hands-on lab exercise is for you to gain experience using Disk Administrator to manage partitions.

Follow the steps carefully to successfully partition and format the remaining 10MB of your computer's hard drive with an NTFS partition.



If you didn't leave enough space on your hard disk to perform this, I recommend that you reinstall NT Workstation and NT Server to accommodate this lab. If this is not possible, you should create an NTFS partition somewhere on one of your hard disks so that you can perform the file security labs later in this book.

Don't delete or reformat any partition that contains data that you don't want to lose. As always, make sure you back up all important data and programs *before* you do the lab exercise.

1. Boot your computer to either Windows NT Server or Windows NT Workstation. Log on as Administrator. (Remember the password? It's *password*.)
2. Close the Welcome to NT dialog box if it appears. (Hint: If you never want to see this dialog box again, the second time it appears, you are given a check box to select if you don't want this box to appear each time you run NT.)
3. Select Start ➪ Programs ➪ Administrative Tools (Common) ➪ Disk Administrator.
4. Disk Administrator displays a dialog box indicating that this is the first time Disk Administrator has been run. Click OK to update the system configuration.
5. If this is the first time Disk Administrator has been run, a Confirm dialog box appears, indicating that no signature is found on Disk 0. Click Yes to write a signature on Disk 0.
6. Click the box that indicates a drive with a 10MB partition (to highlight it), and select Partition ➪ Delete. (Caution! If this partition is displayed as Free Space, don't do this step. Instead, skip to Step 11.)
7. Click Yes in the Confirm dialog box to delete the partition.
8. Select Partition ➪ Commit Changes Now.
9. Click Yes in the Confirm dialog box to save the changes to your disk.
10. Click OK to return to the Disk Administrator main dialog box.
11. Click the box that indicates the drive with 10MB of free space (to highlight it), and then select Partition ➪ Create.
12. A Confirm dialog box appears. Click Yes to continue to create the partition.
13. Click OK in the Create Primary Partition dialog box to create a new partition.
14. Select Partition ➪ Commit Changes Now.
15. Click Yes in the Confirm dialog box to save the changes to your disk.
16. Click OK to return to the Disk Administrator main dialog box.
17. Click the box that indicates the drive with a 10MB partition (labeled Unknown) to highlight it.
18. Select Tools ➪ Format.
19. Choose NTFS in the File System drop-down list box, accept all the other defaults, and then click Start.
20. A warning dialog box appears. Click OK to format the drive.
21. A dialog box appears indicating that the format is complete. Click OK.

22. Click Close to return to the Disk Administrator main dialog box.
23. Select Partition ➢ Exit to exit Disk Administrator.

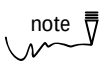
Congratulations! You have now formatted the 10MB partition on your computer with NTFS.

Lab 3.4 *Updating the Emergency Repair Disk*

Server Enterprise

The purpose of this hands-on lab exercise it to provide you with the skills required to update your Emergency Repair Disk, and to give you experience performing this task.

In order to do this lab, you need the Emergency Repair Disks you created in Labs 2.1 and 2.2 during the original installations of your Windows NT operating systems.



The Emergency Repair Disks for Windows NT Workstation and Windows NT Server are different. If you have both operating systems on the same computer, you need to update both disks.

Do this lab twice, once from each operating system.

1. Boot to Windows NT (either Workstation or Server).
2. Select Start ➢ Run.
3. When the Run dialog box appears, type **RDISK** in the drop-down dialog box.
4. Click OK to run `Rdisk.exe`.
5. When the Repair Disk Utility dialog box appears, click Update Repair Info.
6. Click Yes to update your repair information.
7. `Rdisk.exe` saves your current configuration. This takes a couple of minutes. Click Yes to create the Emergency Repair Disk.
8. When prompted, place the Emergency Repair Disk you created in Lab 2.1 (if you are running Windows NT Workstation) or Lab 2.2 (if you are running Windows NT Server) in drive A:. Click OK to create the Emergency Repair Disk. (This process erases your original Emergency Repair Disk and creates a new Emergency Repair Disk with your current system configurations.) It takes a couple of minutes for the Repair Disk Utility to complete this process.
9. Click Exit to exit the Repair Disk Utility. Your Emergency Repair Disk is now updated. Remove it from drive A:. (Remember to do this lab again with your other NT operating system.)