CHAPTER 12

Implementing Hard Drives

In this chapter, you will learn how to

- Explain the partitions available in Windows
- Discuss hard drive formatting options
- Partition and format hard drives
- Maintain and troubleshoot hard drives

From the standpoint of your PC, a new hard drive successfully installed is nothing more than a huge pile of sectors. CMOS sees the drive; it shows up in your autodetect screen and BIOS knows how to talk to the drive, but as far as an operating system is concerned, that drive is unreadable. Your operating system must organize that big pile of sectors so you can create two things: folders and files. This chapter covers that process.

Historical/Conceptual

After you've successfully installed a hard drive, you must perform two more steps to translate a drive's geometry and circuits into something the system can use: partitioning and formatting. *Partitioning* is the process of electronically subdividing the physical hard drive into groups of cylinders called *partitions* (or *volumes*). A hard drive must have at least one partition, and you can create multiple partitions on a single hard drive if you wish. In Windows, each of these partitions typically is assigned a drive letter such as C: or D:. After partitioning, you must *format* the drive. This step installs a *file system* onto the drive that organizes each partition in such a way that the operating system can store files and folders on the drive. Several types of file systems are used in the Windows world. This chapter will go through them after covering partitioning.

Partitioning and formatting a drive is one of the few areas remaining on the software side of PC assembly that requires you to perform a series of fairly complex manual steps. The CompTIA A+ certification exams test your knowledge of *what* these processes do to make the drive work, as well as the steps needed to partition and format hard drives in Windows 2000, XP, and Vista.

This chapter continues the exploration of hard drive installation by explaining partitioning and formatting and then going through the process of partitioning and formatting hard drives. The chapter wraps with a discussion on hard drive maintenance and troubleshooting issues.

Hard Drive Partitions

Partitions provide tremendous flexibility in hard drive organization. With partitions, you can organize a drive to suit your personal taste. For example, I partitioned my 1.5 TB hard drive into a 250-GB partition where I store Windows Vista and all my programs, a second 250-GB partition for Windows 7, and a 1-TB partition where I store all my personal data. This is a matter of personal choice; in my case, backups are simpler because the data is stored in one partition, and I can back up that partition without including the applications.

You can partition a hard drive to store more than one *operating system* (*OS*). Store one OS in one partition and create a second partition for another OS. Granted, most people use only one OS, but if you want the option to boot to Windows or Linux, partitions are the key.

Essentials

Windows 2000/XP and Windows Vista/7 support two different partitioning methods: the older but more universal master boot record (MBR) partitioning scheme and the newer (but proprietary to Microsoft) dynamic storage partitioning scheme. Microsoft calls a hard drive that uses the MBR partitioning scheme a *basic disk* and a drive using the dynamic storage partitioning scheme a *dynamic disk*. A single Windows system with two hard drives may have one of the drives partitioned as a basic disk and the other as a dynamic disk, and the system will run perfectly well. The bottom line? You get to learn about two totally different types of partitioning. Yay! Given that basic disks are much older, we'll start there.

Basic Disks

Basic disk partitioning creates two very small data structures on a drive, the *master boot record* (*MBR*) and a *partition table*, and stores them on the first sector of the hard drive—called the *boot sector*. The MBR is nothing more than a tiny bit of code that takes control of the boot process from the system BIOS. When the computer boots to a hard drive, the BIOS automatically looks for MBR code on the boot sector. The MBR has only one job: to look in the partition table for a partition with a valid operating system (Figure 12-1).

NOTE Only one MBR and one partition table exist per basic disk.

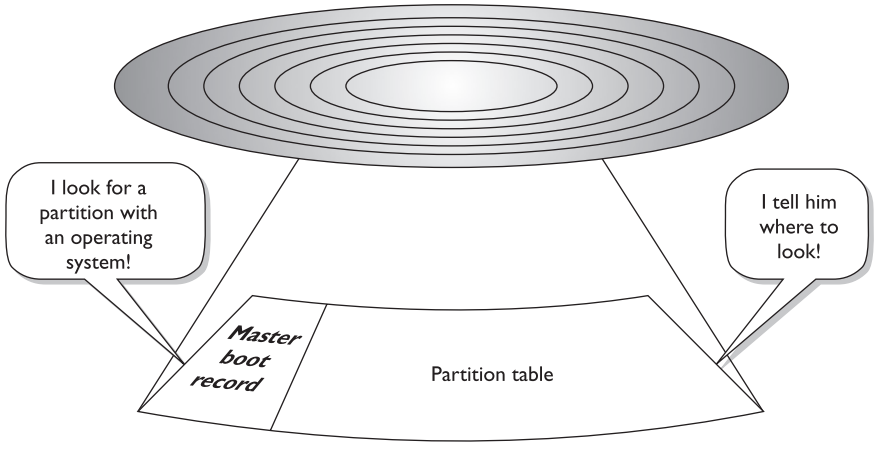


Figure 12-1 Functions of the MBR and partition table

All basic disk partition tables support up to four partitions. The partition table supports two types of partitions: primary partitions and extended partitions. *Primary partitions* are designed to support bootable operating systems. *Extended partitions* are not bootable. A single basic disk may have up to three primary partitions and one extended partition. If you do not have an extended partition, you may have up to four primary partitions.

Each partition must have some unique identifier so users can recognize it as an individual partition. Microsoft operating systems (DOS and Windows) traditionally assign primary partitions a drive letter from C: to Z:. Extended partitions do not get drive letters.

After you create an extended partition, you must create *logical drives* within that extended partition. A logical drive traditionally gets a drive letter from D: to Z:. (The drive letter C: is always reserved for the first primary partition in a Windows PC.)

Windows 2000/XP and Windows Vista/7 partitions are not limited to drive letters. With the exception of the partition that stores the boot files for Windows (which will always be C:), any other primary partitions or logical drives may get either a drive letter or a folder on a primary partition. You'll see how all of this works later in this chapter.

If a primary partition is a bootable partition, why does a basic drive's partition table support up to four primary partitions? Remember when I said that partitioning allows multiple operating systems? This is how it works. You can install up to four different operating systems, each OS installed on its own primary partition, and boot to your choice each time you fire up the computer.

Every primary partition on a single drive has a special setting called *active* stored in the partition table. This setting is either on or off on each primary partition, determining which is the *active partition*. At boot, the MBR uses the active setting in the partition table to determine which primary partition to choose to try to load an OS. Only one partition at a time can be the active partition, because you can run only one OS at a time (see Figure 12-2). This restriction refers to a single drive, by the way. You can have active partitions on more than one physical drive; the settings in CMOS will dictate which drive is the current bootable or system drive.

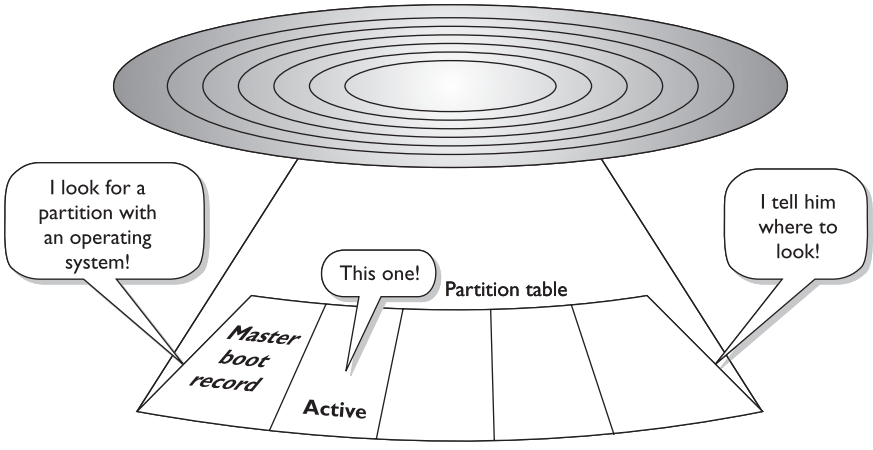
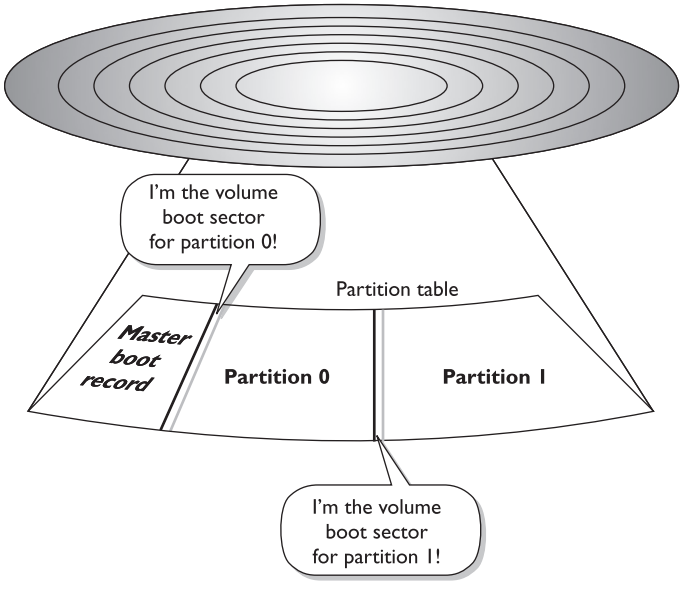


Figure 12-2 The MBR checks the partition table to find the active partition.

The boot sector at the beginning of the hard drive isn't the only special sector on a hard drive. The first sector of the first cylinder of each partition also has a special sector called the *volume boot sector*. Although the "main" boot sector defines the partitions, the volume boot sector stores information important to its partition, such as the location of the OS boot files. Figure 12-3 shows a hard drive with two partitions. The first partition's volume boot sector contains information about the size of the partition and the code pointing to the boot files on this partition. The second volume boot sector contains information about the size of the partition.

Figure 12-3 Volume boot sector



NOTE Every partition on a hard drive has a volume boot sector.

Primary Partitions

If you want to boot an operating system from a hard drive, that hard drive must have a primary partition. The MBR checks the partition table for the active primary partition (see Figure 12-4). In Windows, the primary partition is C:, and that cannot be changed.

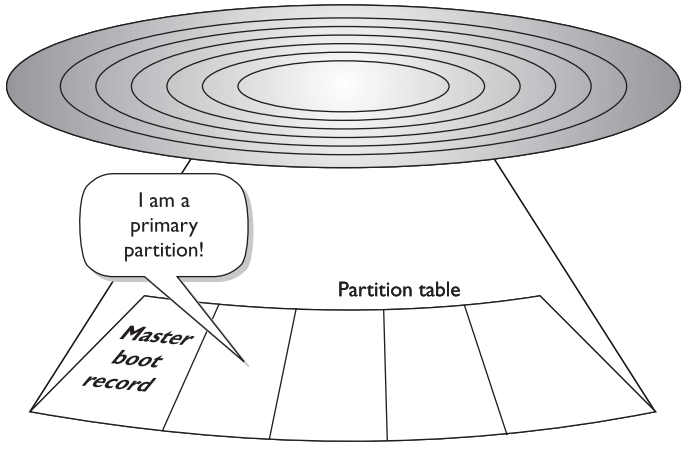


Figure 12-4 The MBR checks the partition table to find a primary partition.

NOTE Don't confuse primary partition with primary controller. The latter, as you'll recall from Chapter 11, "Hard Drive Technologies," refers to the first PATA drive controller on a motherboard.

Even though hard drives support up to four primary partitions, you almost never see four partitions in the Windows world. Windows support up to four primary partitions on one drive, but how many people (other than nerdy CompTIA A+ people like you and me) really want to boot up more than one OS? We use a number of terms for this function, but *dual-boot* and *multiboot* are the most common. The system in my house, for example, uses four primary partitions, each holding one OS: Ubuntu Linux, Windows 2000, Windows XP, and Windows Vista. In other words, I chopped my drive up into four chunks and installed a different OS in each.

To do multiboot, most people use a free, Linux-based boot manager called GRUB (Grand Unified Boot Manager), although some people prefer a third-party tool such as System Commander 9 by VCOM to set up the partitions. Windows 2000 and up come with similar tools that can do this, but they can be messy to use, and GRUB helps simplify the process. When the computer boots, GRUB yanks control from the MBR and asks which OS you wish to boot (see Figure 12-5). You select an OS and it appears.

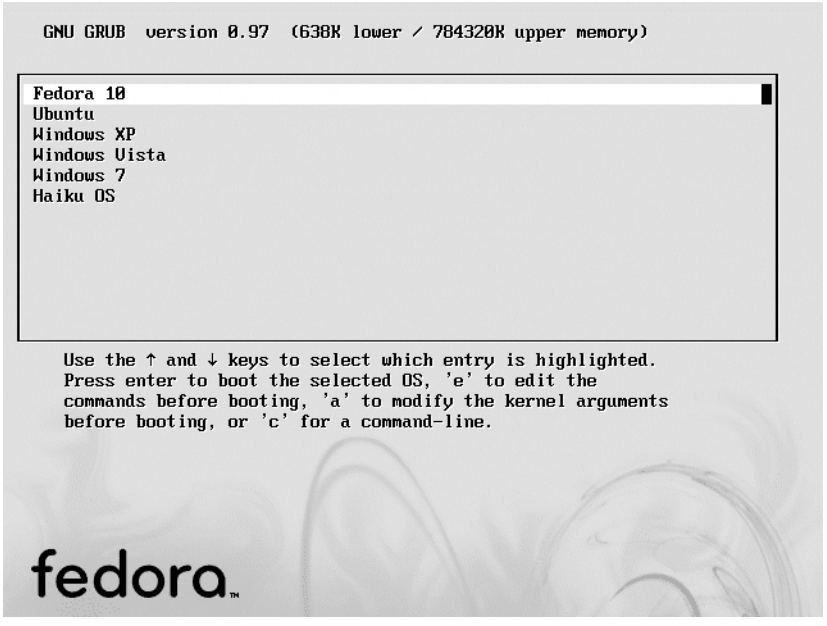


Figure 12-5 GRUB's OS selection menu

Again, few systems use more than one primary partition. You may work on PCs for years and never see a system with more than one primary partition. The CompTIA A+ certification exams certainly don't expect you to demonstrate how to create a system with multiple primary partitions, but they assume that you know you *can* add more than one primary partition to a hard drive if you so desire. The rest of this book assumes that you want only one primary partition.

NOTE If you get the error "No Fixed Disks Present," you can bet you forgot to tell the CMOS to look for the drive. Reboot, access CMOS, and try setting up the drive again.

Active Partition

When you create a primary partition and decide to place an OS on that partition, you must set that partition as active. You must do this even if you use only a single primary partition. Luckily, this step is automated in the Windows installation process. Consider this: When would you want to go though the steps to define a partition as active? That would be when you install an OS on that partition. So when you install Windows on a new system, the installation program automatically sets up your first primary partition as the active partition. It never actually says this in the installation, it just does it for you.

So if you raise your right hand and promise to use only Microsoft Windows and make only single primary partitions on your hard drives, odds are good you'll never have to mess with manually adjusting your active partitions. Of course, because you're crazy enough to want to get into PCs, that means within a year of reading this text you're going to want to install other operating systems such as Linux on your PC (and that's okay—all techs want to try this at some point). The moment you do, you'll enter the world of boot manager programs of which the just-described System Commander is only one of many, many choices. You also might use tools to change the active partition manually—exactly when and how this is done varies tremendously for each situation and is way outside the scope of the CompTIA A+ exams, but make sure you know why you might need to set a partition as active.

When my System Commander boot screen comes up, it essentially asks me, "What primary partition do you want me to make active?"

Extended Partition

Understanding the purpose of extended partitions requires a brief look at the historical PC. The first versions of the old DOS operating system to support hard drives only supported primary partitions up to 32 MB. As hard drives went past 32 MB, Microsoft needed a way to support them. Instead of rewriting DOS to handle larger drives, Microsoft developers created the idea of the extended partition. That way, if you had a hard drive larger than 32 MB, you could make a 32-MB primary partition and the rest of the drive an extended partition. Over the years, DOS and then Windows were rewritten to support large hard drives, but the extended partition is still fully supported.

The beauty of an extended partition is in the way it handles drive letters. When you create a primary partition, it gets a drive letter and that's it. But when you create an extended partition, it does not automatically get a drive letter. Instead, you go through a second step where you divide the extended partition into one or more logical drives. An extended partition may have as many logical drives as you wish. By default, Windows gives each logical drive in an extended partition a drive letter, and most Windows users use drive letters. However, if you'd like, you may even mount the drive letter as a folder on any lettered drive. You can set the size of each logical drive to any size you want. You'll learn how to mount drives later in this chapter—for now, just get the idea that a partition may be mounted with a drive letter or as a folder.

EXAM TIP Primary partitions and logical drives on basic disks are also called *basic volumes*.

Extended partitions are completely optional; you do not have to create an extended partition on a hard drive. So, if you can't boot to an extended partition and your hard drive doesn't need an extended partition, why would you want to create one? First of all, the majority of systems do not use extended partitions. Most systems use only one hard drive, and that single drive is partitioned as one big primary partition—nothing wrong with that! Some users like having an extended partition with one or more logical drives, and they use the extended partitions as a way to separate data. For example, I might store all of my movie files on my G: logical drive.

Instead of assigning drive letters, you can mount logical drives as folders on an existing drive. It's easy to make a logical drive and call it C:\STORAGE. If the C:\STORAGE folder fills up, you could add an extra hard drive, make the entire extra drive an extended partition with one logical drive, unmount the old C:\STORAGE drive, and then mount the new huge logical drive as C:\STORAGE. It's as though you made your C: drive bigger without replacing it.

Dynamic Disks

With the introduction of Windows 2000, Microsoft defined an entirely new type of partitioning called *dynamic storage partitioning*, better known as *dynamic disks*. Dynamic disks drop the word *partition* and instead use the term *volume*. There is no dynamic disk equivalent to primary versus extended partitions. A volume is still technically a partition, but it can do things a regular partition cannot do, such as spanning. A *spanned volume* goes across more than one drive. Windows allows you to span up to 32 drives under a single volume. Dynamic disks also support RAID 0 in Windows 2000 Professional, Windows XP Professional, and Windows Vista Business and Ultimate. Windows 2000, 2003, and 2008 Server editions support RAID 0, 1, and 5.

NOTE Windows XP Home and Windows Media Center do not support dynamic disks, nor do any Vista editions besides Business and Ultimate.

Dynamic disks use an MBR and a partition table, but these older structures are there only for backward compatibility. All of the information about a dynamic disk is stored in a hidden partition that takes up the last 1 MB of the hard drive. Every partition in a partition table holds a 2-byte value that describes the partition. For example, an extended partition gets the number 05. Windows adds a new number, 42, to the first partition on a dynamic disk. When Windows 2000 or XP reads the partition table for a dynamic disk, it sees the number 42 and immediately jumps to the 1-MB hidden partition, ignoring the old-style partition table. By supporting an MBR and partition table, Windows also prevents other disk partitioning programs from messing with a dynamic disk. If you use a third-party partitioning program, it simply sees the entire hard drive as either an unformatted primary partition or a non-readable partition.

NOTE A key thing to understand about dynamic drives is that the technology is *proprietary*. Microsoft has no intention of telling anyone exactly how dynamic disks work. Only fairly recent Microsoft operating systems (Windows 2000 and up) can read a drive configured as a dynamic disk.

You can use five volume types with dynamic disks: simple, spanned, striped, mirrored, and RAID 5. Most folks stick with simple volumes.

Simple volumes work much like primary partitions. If you have a hard drive and you want to make half of it C: and the other half D:, you create two volumes on a dynamic disk.

That's it: no choosing between primary and extended partitions. Remember that you were limited to four primary partitions when using basic disks. To make more than four volumes with a basic disk, you first had to create an extended partition and then make logical drives within the extended partition. Dynamic disks simplify the process by treating all partitions as volumes, so you can make as many as you need.

Spanned volumes use unallocated space on multiple drives to create a single volume. Spanned volumes are a bit risky: if any of the spanned drives fails, the entire volume is permanently lost.

Striped volumes are RAID 0 volumes. You may take any two unallocated spaces on two separate hard drives and stripe them. But again, if either drive fails, you lose all of your data.

Mirrored volumes are RAID 1 volumes. You may take any two unallocated spaces on two separate hard drives and mirror them. If one of the two mirrored drives fails, the other keeps running.

RAID 5 volumes, as the name implies, are for RAID 5 arrays. A RAID 5 volume requires three or more dynamic disks with equal-sized unallocated spaces.

Other Partitions

The partition types supported by Windows are not the only partition types you may encounter; other types exist. One of the most common is called the *hidden partition*. A hidden partition is really just a primary partition that is hidden from your operating system. Only special BIOS tools may access a hidden partition. Hidden partitions are used by some PC makers to hide a backup copy of an installed OS that you can use to restore your system if you accidentally trash it—by, for example, learning about partitions and using a partitioning program incorrectly.

A *swap partition* is another special type of partition, but swap partitions are only found on Linux and BSD systems. A swap partition's only job is to act like RAM when your system needs more RAM than you have installed. Windows has a similar function called a *page file* that uses a special file instead of a partition. Most OS experts believe a swap partition is a little bit faster than a page file. You'll learn all about page files and swap partitions in Chapter 17, "Maintaining and Troubleshooting Windows."

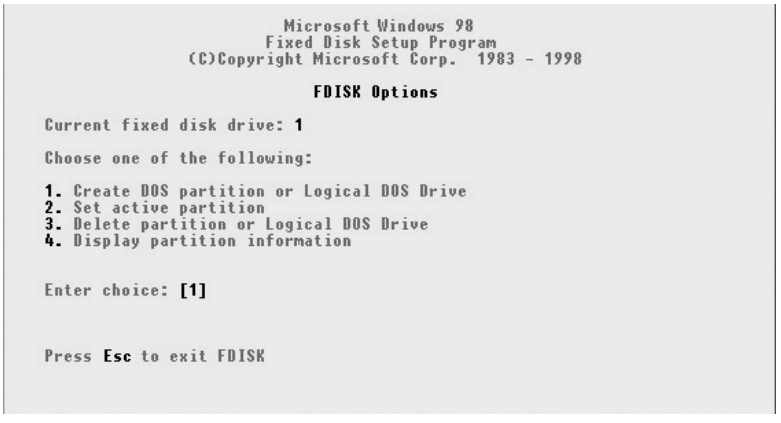
NOTE Early versions of Windows (3.x and 9x/Me) called the page file a *swap file*. Most techs use the terms interchangeably today.

When to Partition

Partitioning is not a common task. The two most common situations likely to require partitioning are when you're installing an OS on a new system, and when you are adding a second drive to an existing system. When you install a new OS, the installation CD at some point asks you how you would like to partition the drive. When you're adding a new hard drive to an existing system, every OS has a built-in tool to help you partition it.

Each version of Windows offers a different tool for partitioning hard drives. For more than 20 years, through the days of DOS and early Windows (up to Windows Me), you used a command-line program called *FDISK* to partition drives. Figure 12-6 shows the FDISK program. Windows 2000, Windows XP, and Windows Vista use a graphical partitioning program called *Disk Management* (Figure 12-7).

Figure 12-6 FDISK



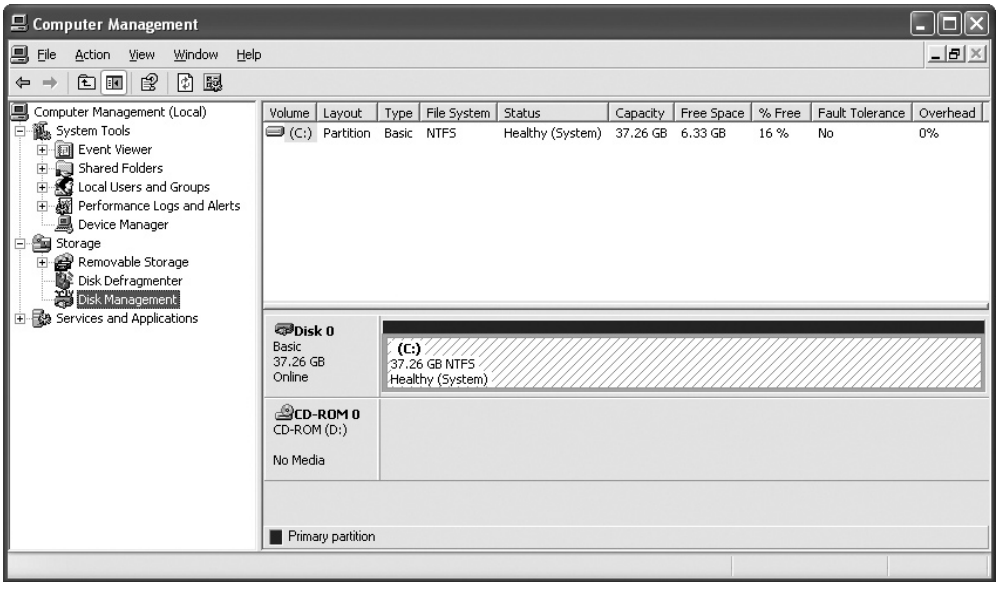


Figure 12-7 Windows XP Disk Management tool in Computer Management

Linux uses a number of different tools for partitioning. The oldest is called FDISK—yup, the exact same name as the DOS/Windows version. However, that's where the similarities end, as Linux FDISK has a totally different command set. Even though every copy of Linux comes with the Linux FDISK, it's rarely used because so many better partitioning tools are available. One of the newer Linux partitioning tools is called GParted. GParted is graphical like Disk Management and is fairly easy to use (Figure 12-8). GParted is also a powerful partition management tool—so powerful that it also works with Windows partitions.

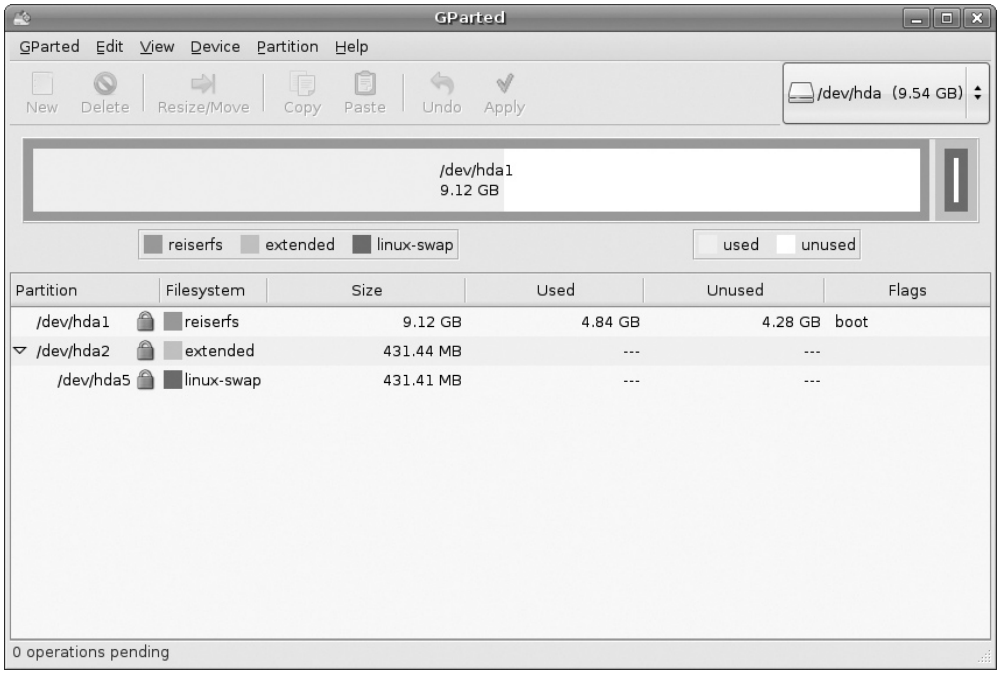


Figure 12-8 GParted in action

Traditionally, once you make a partition, you cannot change its size or type other than by erasing it. You might, however, want to take a hard drive partitioned as a single primary partition and change it to half primary and half extended. Before Windows 2000, there was no way to do this nondestructively. As a result, a few third-party tools, led by Symantec's now famous Norton PartitionMagic, gave techs the tools to resize partitions without losing the data they held. Windows 2000 and XP can nondestructively resize a partition to be larger but not smaller.

In Vista, you can nondestructively resize partitions by shrinking or expanding existing partitions with available free space. Although undoubtedly handy, this is sometimes hampered by the presence of unmovable system files, such as the MBR. You can

sometimes circumvent this problem by disabling such things as Hibernation mode and System Restore, but that doesn't always work, and third-party tools remain necessary in many cases.

NOTE This chapter explains how to partition a hard drive *before* it explains formatting because that is the order in which you as a PC tech will actually perform those tasks. You'll learn all of the specifics of the various file systems—such as FAT32 and NTFS—when I explain formatting in the next section, but until then, just accept that there are several systems for organizing the files on a hard drive, and that part of setting up a hard drive involves choosing among them.

Hard Drive Formatting

Once you've partitioned a hard drive, you must perform one more step before your OS can use that drive: formatting. *Formatting* does two things: it creates a file system—like a library's card catalog—and makes the root directory in that file system. You must format every partition and volume you create so it can hold data that you can easily retrieve. The various versions of Windows you're likely to encounter today can use several different file systems, so we'll look at those in detail next. The *root directory* provides the foundation upon which the OS builds files and folders.

File Systems in Windows

Every version of Windows comes with a built-in formatting utility with which to create one or more file systems on a partition or volume. The versions of Windows in current use support three separate Microsoft file systems: FAT16, FAT32, and NTFS.

The simplest hard drive file system, called FAT or FAT16, provides a good introduction to how file systems work. More complex file systems fix many of the problems inherent in FAT and add extra features as well.

FAT

The base storage area for hard drives is a sector; each sector stores up to 512 bytes of data. If an OS stores a file smaller than 512 bytes in a sector, the rest of the sector goes to waste. We accept this waste because most files are far larger than 512 bytes. So what happens when an OS stores a file larger than 512 bytes? The OS needs a method to fill one sector, find another that's unused, and fill it, continuing to fill sectors until the file is completely stored. Once the OS stores a file, it must remember which sectors hold the file, so it can be retrieved later.

MS-DOS version 2.1 first supported hard drives using a special data structure to keep track of stored data on the hard drive, and Microsoft called this structure the *file allocation table* (*FAT*). Think of the FAT as nothing more than a card catalog that keeps track of which sectors store the various parts of a file. The official jargon term for a FAT is *data structure*, but it is more like a two-column spreadsheet.

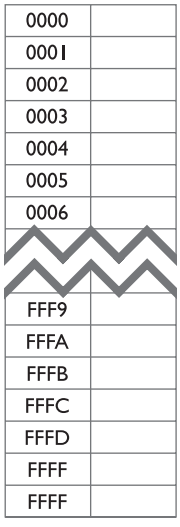


Figure 12-9
16-bit FAT

The left column (see Figure 12-9) gives each sector a number from 0000 to FFFF (in hex, of course). This means there are 65,536 (64 K) sectors.

Notice that each value in the left column contains 16 bits. (Four hex characters make 16 bits, remember?) We call this type of FAT a *16-bit FAT* or *FAT16*. Not just hard drives have FATs. Some USB thumb drives also use FAT16. Floppy disks use FATs, but their FATs are only 12 bits because they store much less data.

The right column of the FAT contains information on the status of sectors. All hard drives, even brand-new drives fresh from the factory, contain faulty sectors that cannot store data because of imperfections in the construction of the drives. The OS must locate these bad sectors, mark them as unusable, and then prevent any files from being written to them. This mapping of bad sectors is one of the functions of *high-level formatting*. After the format program creates the FAT, it proceeds through the entire partition, writing and attempting to read from each sector sequentially. If it finds a bad sector, it places a special status code (FFF7) in the sector's FAT location, indicating that the sector is unavailable for use. Formatting also marks the good sectors as 0000.

NOTE There is such a thing as "low-level formatting," but that's generally done at the factory and doesn't concern techs. This is especially true if you're working with modern hard drives (post-2001).

Using the FAT to track sectors, however, creates a problem. The 16-bit FAT addresses a maximum of 64 K (2^{16}) locations. Therefore, the size of a hard drive partition should be limited to 64 K × 512 bytes per sector, or 32 MB. When Microsoft first unveiled FAT16, this 32-MB limit presented no problem because most hard drives were only 5 to 10 MB. As hard drives grew in size, you could use FDISK to break them up into multiple partitions. You could divide a 40-MB hard drive into two partitions, for example, making each partition smaller than 32 MB. But as hard drives started to become much larger, Microsoft realized that the 32-MB limit for drives was unacceptable. We needed an improvement to the 16-bit FAT, a new and improved FAT16 that would support larger drives while still maintaining backward compatibility with the old style 16-bit FAT. This need led to the development of a dramatic improvement in FAT16, called *clustering*, that enabled you to format partitions larger than 32 MB (see Figure 12-10). This new FAT16 appeared way back in the DOS-4 days.

Clustering simply refers to combining a set of contiguous sectors and treating them as a single unit in the FAT. These units are called *file allocation units* or *clusters*. Each row of the FAT addressed a cluster instead of a sector. Unlike sectors, the size of a cluster is not fixed. Clusters improved FAT16, but it still only supported a maximum of 64 K storage units, so the formatting program set the number of sectors in each cluster according

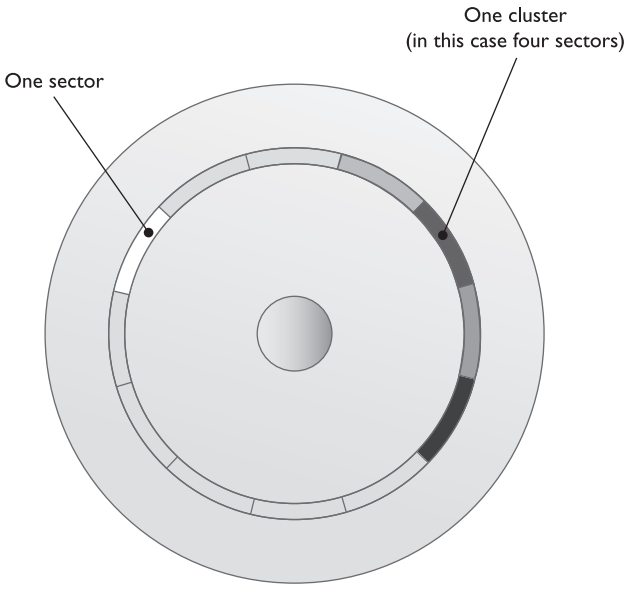


Figure 12-10 Cluster versus sector

to the size of the partition. The larger the partition, the more sectors per cluster. This method kept clustering completely compatible with the 64-K locations in the old 16-bit FAT. The new FAT16 could support partitions up to 2 GB. (The old 16-bit FAT is so old it doesn't really even have a name—if someone says "FAT16," they mean the newer FAT16 that supports clustering.) Table 12-1 shows the number of sectors per cluster for FAT16.

Table 12-1 FAT16 Cluster Sizes

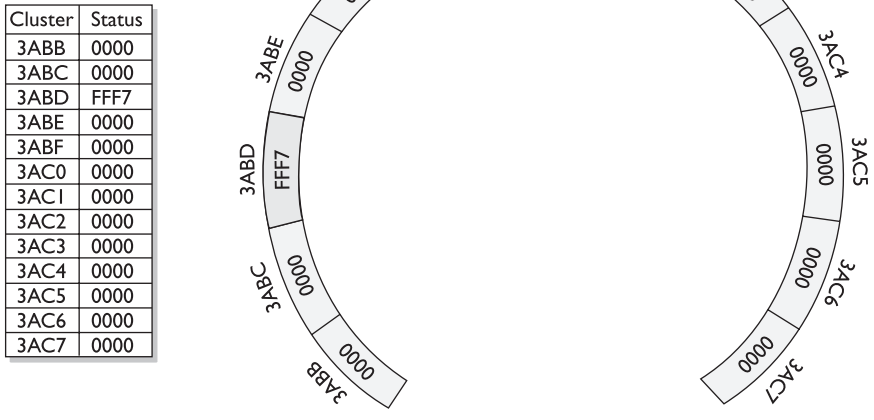
If FDISK makes a partition this big:	You'll get this many sectors/cluster
16 to 127.9 MB	4
128 to 255.9 MB	8
256 to 511.9 MB	16
512 to 1023.9 MB	32
1024 to 2048 MB	64

FAT16 in Action

Assume you have a copy of Windows using FAT16. When an application such as Microsoft Word tells the OS to save a file, Windows starts at the beginning of the FAT, looking for the first space marked "open for use" (0000), and begins to write to that cluster. If the entire file fits within that one cluster, Windows places the code FFFF (last cluster) into the cluster's status area in the FAT. That's called the *end-of-file marker*. Windows then goes to the folder storing the file and adds the filename and the cluster's number

to the folder list. If the file requires more than one cluster, Windows searches for the next open cluster and places the number of the next cluster in the status area, filling and adding clusters until the entire file is saved. The last cluster then receives the end-of-file marker (FFFF).

Let's run through an example of this process, and start by selecting an arbitrary part of the FAT: from 3ABB to 3AC7. Assume you want to save a file called MOM.TXT. Before saving the file, the FAT looks like Figure 12-11.



Cluster	Status
3ABB	0000
3ABC	0000
3ABD	FFF7
3ABE	0000
3ABF	0000
3AC0	0000
3AC1	0000
3AC2	0000
3AC3	0000
3AC4	0000
3AC5	0000
3AC6	0000
3AC7	0000

Figure 12-11 The initial FAT

Windows finds the first open cluster, 3ABB, and fills it. But not all of the MOM.TXT file fits into that cluster. Needing more space, the OS goes through the FAT to find the next open cluster. It finds cluster 3ABC. Before filling 3ABC, the value *3ABC* is placed in 3ABB's status (see Figure 12-12).

Even after filling two clusters, more of the MOM.TXT file remains, so Windows must find one more cluster. The 3ABD has been marked FFF7 (bad cluster or *bad-sector marker*), so Windows skips over 3ABD, finding 3ABE (see Figure 12-13).

Before filling 3ABE, Windows enters the value *3ABE* in 3ABC's status. Windows does not completely fill 3ABE, signifying that the entire MOM.TXT file has been stored. Windows enters the value *FFFF* in 3ABE's status, indicating the end of file (see Figure 12-14).

After saving all of the clusters, Windows locates the file's folder (yes, folders also are stored on clusters, but they get a different set of clusters, somewhere else on the disk) and records the filename, size, date/time, and starting cluster, like this:

MOM.TXT 19234 05-19-07 2:04p 3ABB

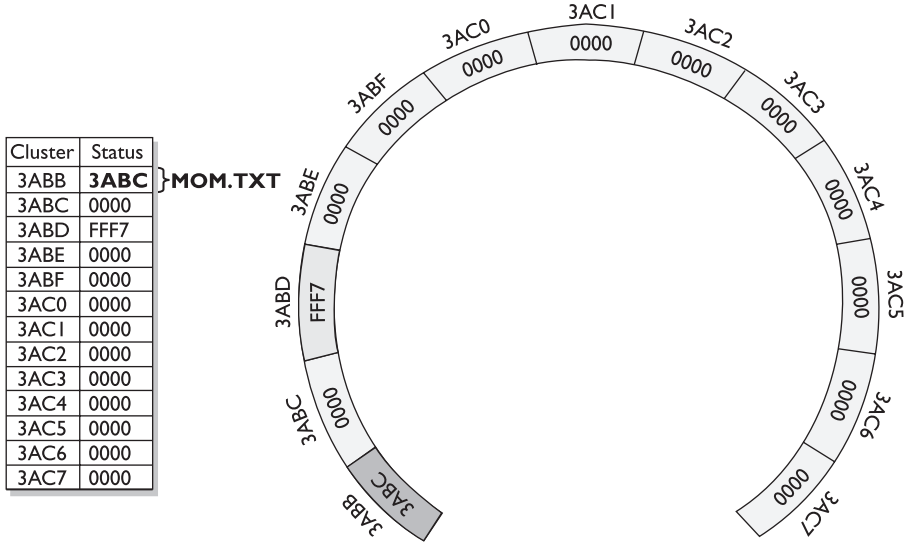


Figure 12-12 The first cluster used

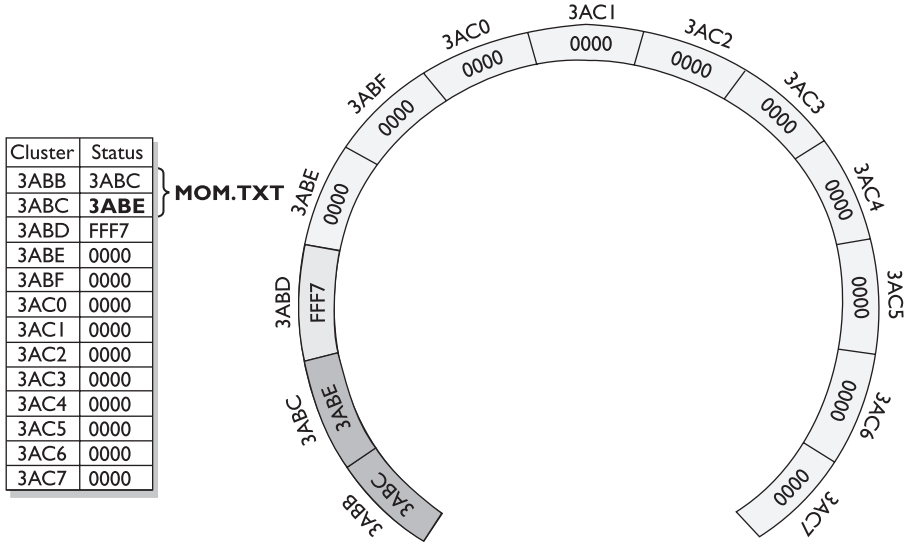


Figure 12-13 The second cluster used

If a program requests that file, the process is reversed. Windows locates the folder containing the file to determine the starting cluster and then pulls a piece of the file from each cluster until it sees the end-of-file cluster. Windows then hands the reassembled file to the requesting application.

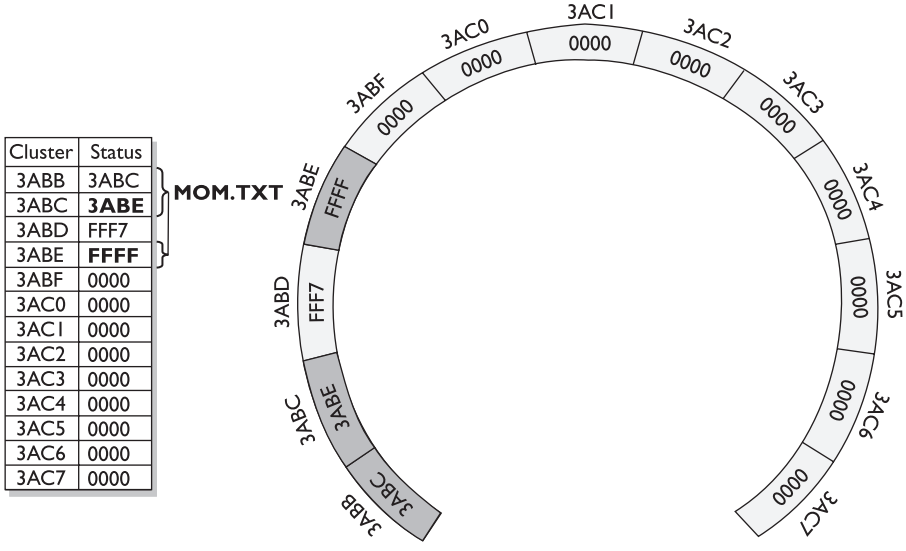


Figure 12-14 End of file reached

Clearly, without the FAT, Windows cannot locate files. FAT16 automatically makes two copies of the FAT. One FAT backs up the other to provide special utilities a way to recover a FAT that gets corrupted—a painfully common occurrence.

Even when FAT works perfectly, over time the files begin to separate in a process called *fragmentation*.

Fragmentation

Continuing with the example, let's use Microsoft Word to save two more files: a letter to the IRS (IRSROB.DOC) and a letter to IBM (IBMHELP.DOC). IRSROB.DOC takes the next three clusters—3ABF, 3AC0, and 3AC1—and IBMHELP.DOC takes two clusters—3AC2 and 3AC3 (see Figure 12-15).

Now suppose you erase MOM.TXT. Windows does not delete the cluster entries for MOM.TXT when it erases a file. Windows only alters the information in the folder, simply changing the first letter of MOM.TXT to the Greek letter Σ (sigma). This causes the file to "disappear" as far as the OS knows. It won't show up, for example, in Windows Explorer, even though the data still resides on the hard drive for the moment (see Figure 12-16).

Note that under normal circumstances, Windows does not actually delete files when you press the DELETE key. Instead, Windows moves the files to a special hidden directory that you can access via the Recycle Bin. The files themselves are not actually deleted until you empty the Recycle Bin. (You can skip the Recycle Bin entirely if you wish, by highlighting a file and then holding down the SHIFT key when you press DELETE).

Because all of the data for MOM.TXT is intact, you could use some program to change the Σ back into another letter and thus get the document back. A number of

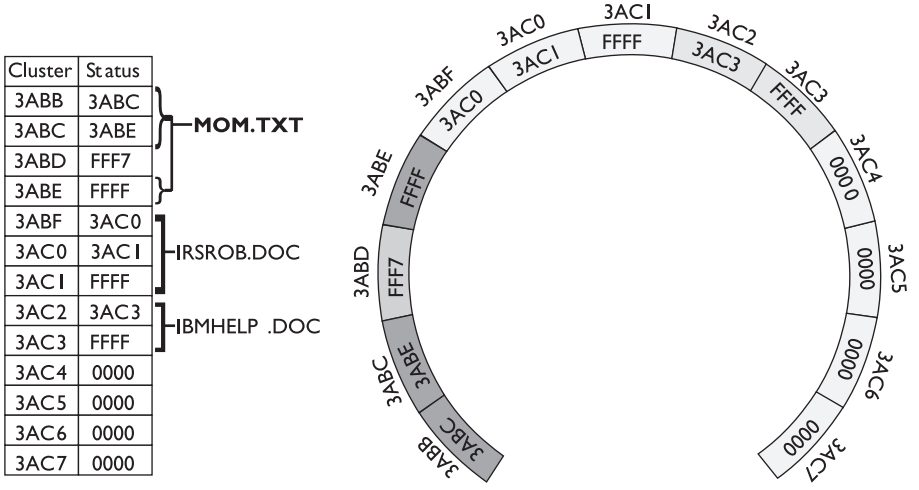


Figure 12-15 Three files saved

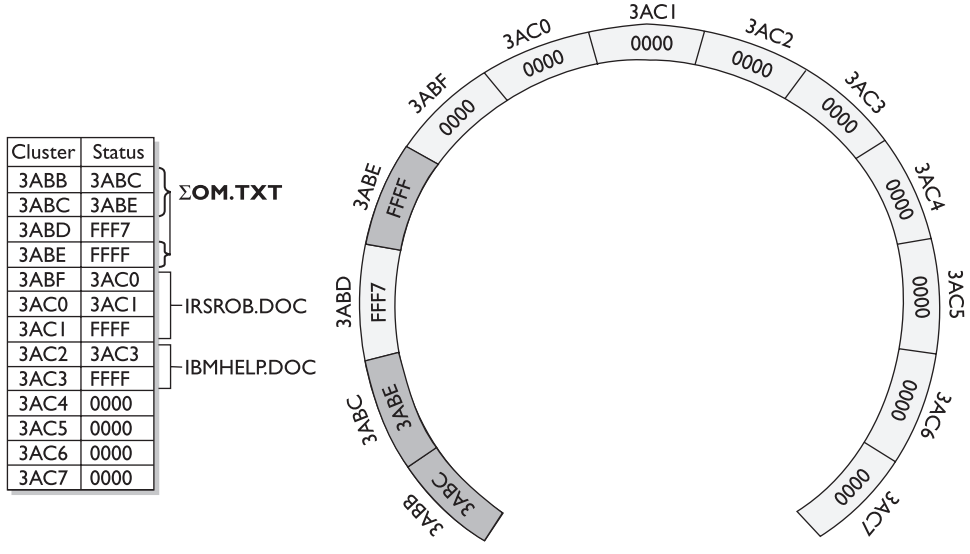


Figure 12-16 MOM.TXT erased

third-party undelete tools are available. Figure 12-17 shows one such program at work. Just remember that if you want to use an undelete tool, you must use it quickly. The space allocated to your deleted file may soon be overwritten by a new file.

Let's say you just emptied your Recycle Bin. You now save one more file, TAXREC .XLS, a big spreadsheet that will take six clusters, into the same folder that once held MOM.TXT. As Windows writes the file to the drive, it overwrites the space that MOM .TXT used, but it needs three more clusters. The next three available clusters are 3AC4, 3AC5, and 3AC6 (see Figure 12-18).

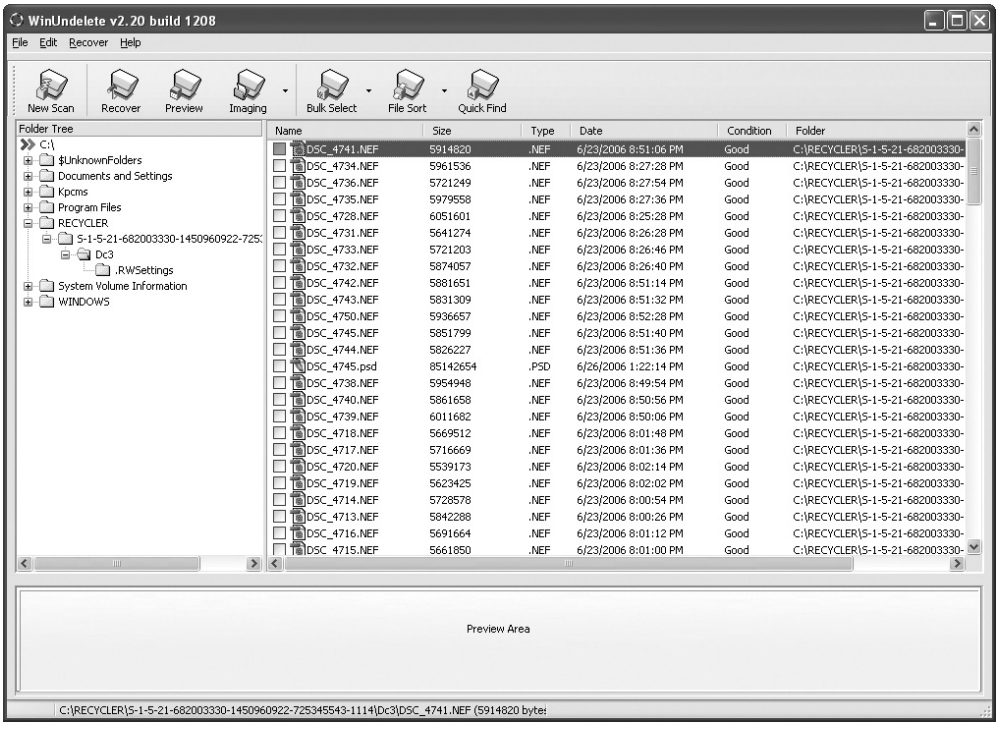


Figure 12-17 WinUndelete in action

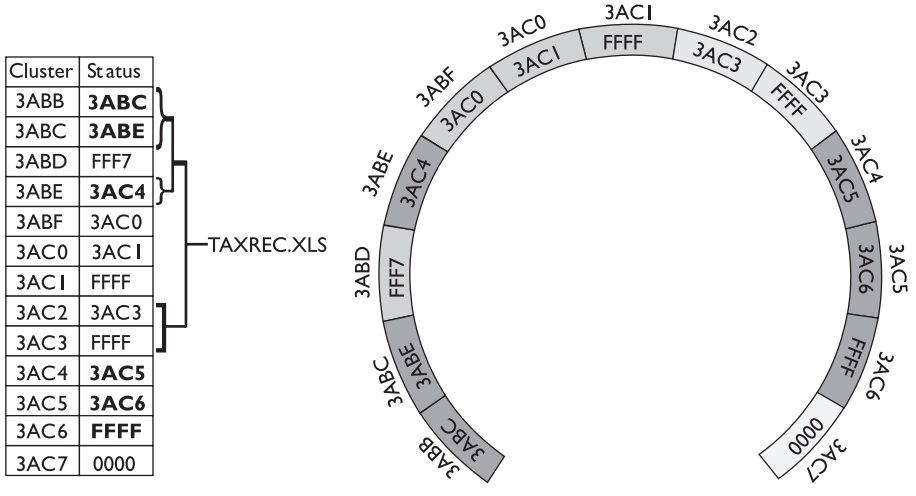


Figure 12-18 TAXREC.XLS fragmented

Notice that TAXREC.XLS is in two pieces, thus *fragmented*. Fragmentation takes place all of the time on FAT16 systems. Although the system easily negotiates a tiny fragmented file split into only two parts, excess fragmentation slows down the system during hard drive reads and writes. This example is fragmented into two pieces; in the real world, a file might fragment into hundreds of pieces, forcing the read/write heads to travel all over the hard drive to retrieve a single file. You can dramatically improve the speed at which the hard drive reads and writes files by eliminating this fragmentation.

Every version of Windows (with the exception of NT) comes with a program called Disk Defragmenter, which can rearrange the files into neat contiguous chunks (see Figure 12-19). Defragmentation is crucial for ensuring the top performance of a hard drive. The "Maintaining and Troubleshooting Hard Drives" section of this chapter gives the details on working with the various Disk Defragmenters in Windows.

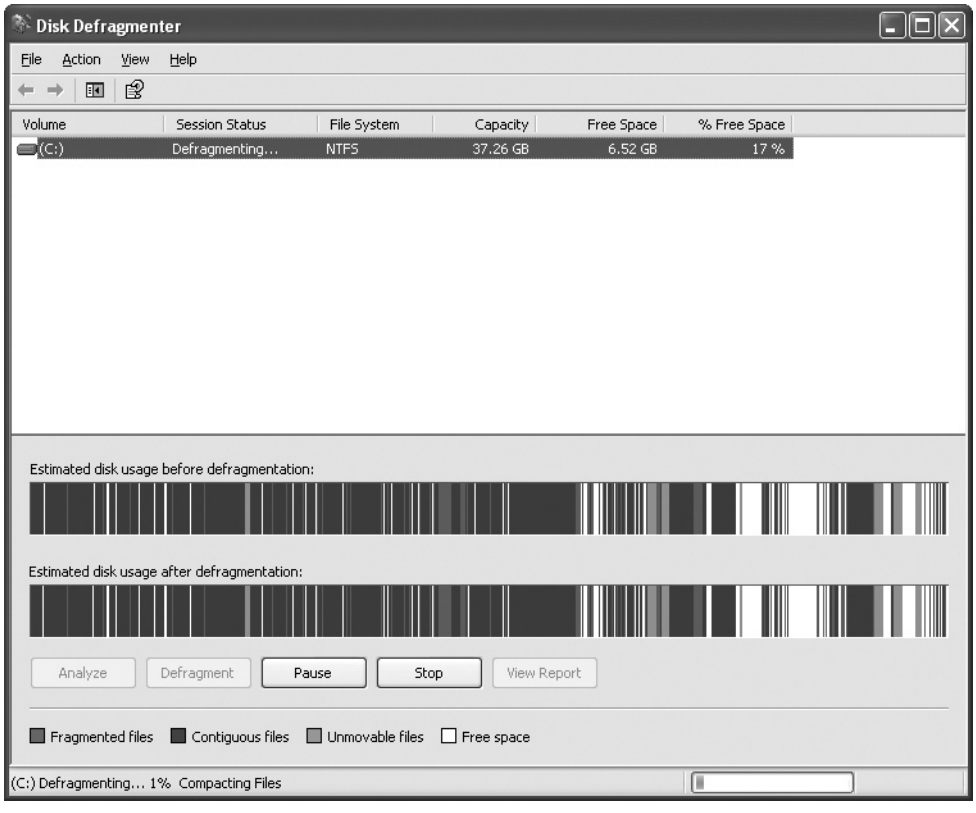


Figure 12-19 Windows Disk Defragmenter

FAT32

When Microsoft introduced Windows 95 OSR2 (OEM Service Release 2), it also unveiled a totally new file format called *FAT32* that brought a couple of dramatic improvements.

First, FAT32 supports partitions up to 2 terabytes (more than 2 trillion bytes). Second, as its name implies, FAT32 uses 32 bits to describe each cluster, which means clusters can drop to more reasonable sizes. FAT32's use of so many FAT entries gives it the power to use small clusters, making the old "keep your partitions small" rule obsolete. A 2-GB volume using FAT16 would use 32-KB clusters, while the same 2-GB volume using FAT32 would use 4-KB clusters. You get far more efficient use of disk space with FAT32, without the need to make multiple small partitions. FAT32 partitions still need defragmentation, however, just as often as FAT16 partitions.

Table 12-2 shows cluster sizes for FAT32 partitions.

Table 12-2 FAT32 Cluster Sizes

Drive Size	Cluster Size
512 MB or 1023 MB	4 KB
1024 MB to 2 GB	4 KB
2 GB to 8 GB	4 KB
8 GB to 16 GB	8 KB
16 GB to 32 GB	16 KB
>32 GB	32 KB

Practical Application

NTFS

The Windows format of choice these days is the *New Technology File System* (*NTFS*). NTFS came out a long time ago with the first version of Windows NT, thus the name. Over the years, NTFS has undergone a number of improvements. The version used in Windows 2000 is NTFS 3.0; the version used in Windows XP and Vista is called NTFS 3.1, although you'll see it referred to as NTFS 5.0/5.1 (Windows 2000 was unofficially Windows NT version 5). NTFS uses clusters and file allocation tables but in a much more complex and powerful way compared to FAT or FAT32. NTFS offers six major improvements and refinements: redundancy, security, compression, encryption, disk quotas, and cluster sizing.

NOTE If you have a geeky interest in what version of NTFS you are running, open up a prompt and type this command: **fsutil fsinfo ntfsinfo c**:

NTFS Structure

NTFS utilizes an enhanced file allocation table called the *master file table* (*MFT*). An NTFS partition keeps a backup copy of the most critical parts of the MFT in the middle of the disk, reducing the chance that a serious drive error can wipe out both the MFT

and the MFT copy. Whenever you defragment an NTFS partition, you'll see a small, immovable chunk in the middle of the drive; that's the backup MFT (Figure 12-20). It's a little difficult to see in black and white, but the color image shows bright green in both the key at the bottom of the screen and in the estimated disk usage bars.

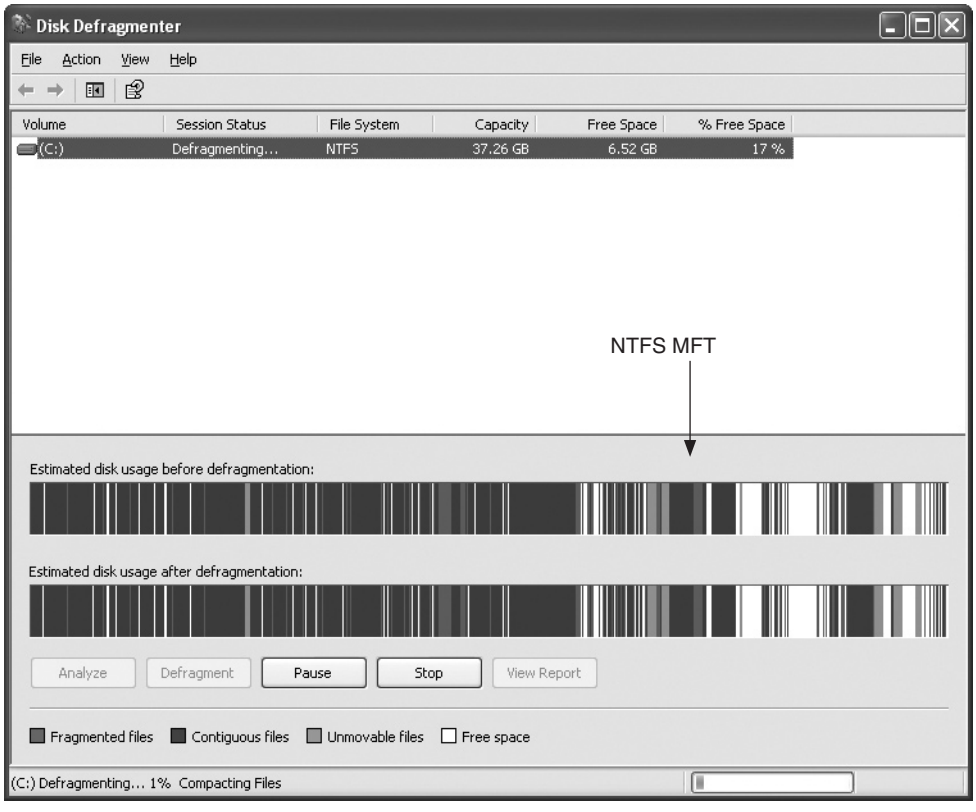


Figure 12-20 An NTFS MFT appears in a defragmenter program as an immovable file.

EXAM TIP Both of the CompTIA A+ exams test you on NTFS, such as when to use it, what advantages it has over FAT32, and how to lock down information. You'll also be quizzed on the tools, such as Disk Management, in both exams. Don't skip anything in this chapter!

Security

NTFS views individual files and folders as objects and provides security for those objects through a feature called the *access control list* (*ACL*). Future chapters go into this in much more detail, but a quick example here should make the basic concept clear.

Suppose Bill the IT Guy sets up a Windows XP PC as a workstation for three users: John, Wilma, and Felipe. John logs into the PC with his user name and password (johns

and f3f2f1f0, respectively, in case you're curious) and begins to work on his project. The project folder is stored on the C: drive as C:\Projects\JohnSuperSecret. When John saves his work and gets ready to leave, he alters the permissions on his folder to deny access to anyone but him. When curious Wilma logs into the PC after John leaves, she cannot access the C:\Programs\JohnSuperSecret folder contents at all, although she can see the entry in Explorer. Without the ACL provided by NTFS, John would have no security over his files or folders at all.

NOTE Microsoft has never released the exact workings of NTFS to the public.

Compression

NTFS enables you to compress individual files and folders to save space on a hard drive. Compression makes access time to the data slower because the OS has to uncompress files every time you use them, but in a space-limited environment, sometimes that's what you have to do.

Encryption

One of the big draws with NTFS is file encryption, the black art of making files unreadable to anybody who doesn't have the right key. You can encrypt a single file, a folder, or a folder full of files. Microsoft calls the encryption utility in NTFS the *encrypting file system* (*EFS*), but it's simply an aspect of NTFS, not a standalone file system. To encrypt a file or folder, right-click it in My Computer or Computer and select Properties to open the Properties dialog box (Figure 12-21). Click the Advanced button to open the

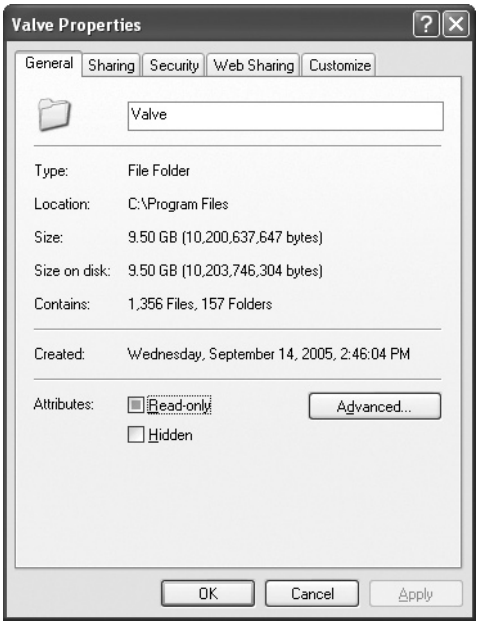


Figure 12-21
Folder Properties

Advanced Attributes dialog box. As you can see in Figure 12-22, encryption (and compression) is simply a selectable checkbox. Click the box next to *Encrypt contents to secure data* and then click the OK button—instantly your file is safe from prying eyes!

Figure 12-22 Options for compression and encryption

EXAM TIP Windows XP Home and Media Center editions do not support EFS.

Encryption does not hide files; it simply makes them unreadable to other users. Figure 12-23 shows a couple of image files encrypted by another user. Note that in addition to the pale green color of the filenames (that you can't tell are pale green in this black-and-white image), the files seem readily accessible. Windows XP can't provide a

Figure 12-23 Encrypted files

thumbnail, however, even though it can read the type of image file (JPEG) easily. Further, double-clicking the files opens the Windows Picture and Fax Viewer, but you still can't see the image (Figure 12-24). Better still, you can try to access the files across your network and the encryption does precisely what it's supposed to do: blocks unwanted access to sensitive data.

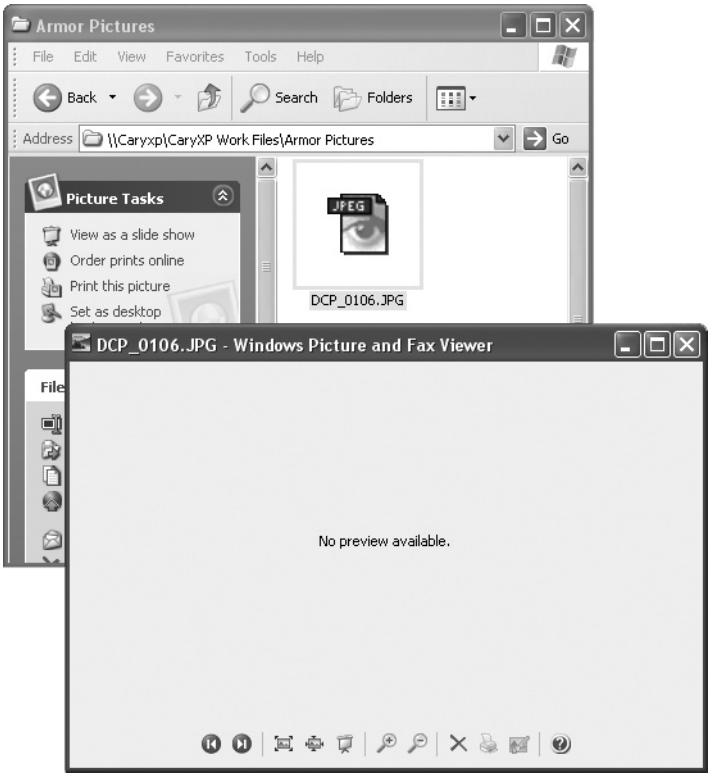


Figure 12-24 Windows Picture and Fax Viewer blocked by file encryption

Remember that encryption is separate from the NTFS file security provided by the ACL—to access encrypted files, you need both permission to access the files based on the ACL and the keys used to encrypt the files. We discuss key management in much more detail in Chapter 16, "Securing Windows Resources."

NOTE Encryption protects against other users, but only if you log out. It might seem obvious, but I've had lots of users get confused by encryption, thinking that the PC *knows* who's clicking the keyboard. All protections and security are based on user accounts. If someone logs into your computer with a different account, the encrypted files will be unreadable. We'll get to user accounts, permissions, and such in later chapters in detail.

Disk Quotas

NTFS supports *disk quotas*, enabling administrators to set limits on drive space usage for users. To set quotas, you must log in as an Administrator, right-click the hard drive name, and select Properties. In the Drive Properties dialog box, select the Quota tab and make changes. Figure 12-25 shows configured quotas for a hard drive. Although rarely used on single-user systems, setting disk quotas on multi-user systems prevents any individual user from monopolizing your hard disk space.

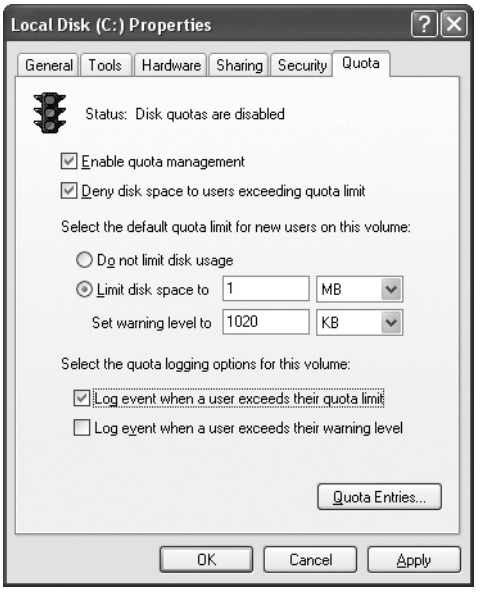


Figure 12-25 Hard drive quotas in Windows XP

Cluster Sizes

Unlike FAT16 or FAT32, you can adjust the cluster sizes in NTFS, although you'll probably rarely do so. Table 12-4 shows the default cluster sizes for NTFS.

Table 12-3 NTFS Cluster Sizes

Drive Size	Cluster Size	Number of Sectors
512 MB or less	512 bytes	1
513 MB to 1024 MB (1 GB)	1024 bytes (1 KB)	2
1,025 MB to 2048 MB (2 GB)	2048 bytes (2 KB)	4
2,049 MB and larger	4096 bytes (4 KB)	8

By default, NTFS supports partitions up to ~16 terabytes on a dynamic disk, (though only up to 2 TB on a basic disk). By tweaking the cluster sizes, you can get NTFS to support partitions up to 16 exabytes, or 18,446,744,073,709,551,616 bytes! That might support any and all upcoming hard drive capacities for the next 100 years or so.

EXAM TIP NTFS supports partitions up to 16 TB by default.

With so many file systems, how do you know which one to use? In the case of internal hard drives, you should use the most feature-rich system your OS supports. If you have Windows 2000 or greater, use NTFS. External hard drives still often use FAT32 because NTFS features such as the ACL and encryption can make access difficult when you move the drive between systems, but with that exception, NTFS is your best choice on a Windows-based system.

The Partitioning and Formatting Process

Now that you understand the concepts of formatting and partitioning, let's go through the process of setting up an installed hard drive by using different partitioning and formatting tools. If you have access to a system, try following along with these descriptions. Remember, don't make any changes to a drive you want to keep, because both partitioning and formatting are destructive processes.

Bootable Disks

Imagine you've built a brand new PC. The hard drive has no OS so you need to boot up something to set up that hard drive. Any software that can boot up a system is by definition an operating system. You need a floppy disk, optical disk, or USB thumb drive with a bootable OS installed. Any removable media that has a bootable OS is generically called a *boot device* or *boot disk*. Your system boots off of the boot device, which then loads some kind of OS that enables you to partition, format, and install an OS on your new hard drive. Boot devices come from many sources. All Windows OS installation discs are boot devices, as are Linux installation discs. You can make your own bootable devices, and most techs do, because a boot device often has a number of handy tools included to do certain jobs.

In Chapter 13, "Removable Media," I go through the steps to make a number of different boot devices for different jobs. If you want to follow along with some of the steps in this chapter, you may want to jump ahead to the next chapter to make a boot device or two and then return here.

Partitioning and Formatting with the Windows XP Installation CD

When you boot up a Windows XP installation CD and the installation program detects a hard drive that is not yet partitioned, it prompts you through a sequence of steps to partition (and format) the hard drive. Chapter 14, "Installing and Configuring Windows," covers the entire installation process, but we'll jump ahead and dive into the partitioning part of the installation here to see how this is done, working through two examples by using one and then two partitions. Even though this example uses the Windows XP

installation CD, don't worry, because this part of the Windows 2000 installation is almost identical and the next section discusses Vista in detail.

Single Partition

The most common partitioning scenario involves turning a new blank drive into a single bootable C: drive. To accomplish this goal, you need to make the entire drive a primary partition and then make it active. Let's go through the process of partitioning and formatting a single, brand-new, 200-GB hard drive.

The Windows installation begins by booting from a Windows installation CD-ROM like the one shown in Figure 12-26. The installation program starts automatically from the CD. The installation first loads some needed files but eventually prompts you with the screen shown in Figure 12-27. This is your clue that partitioning is about to start.

Figure 12-26 Windows installation CD

Press the ENTER key to start a new Windows installation and accept the license agreement to see the main partitioning screen (Figure 12-28). The bar that says Unpartitioned Space is the drive.

The Windows installer is pretty smart. If you press ENTER at this point, it partitions the hard drive as a single primary partition, makes it active, and installs Windows for you—but what fun is that? Instead, press C to create a partition. The installer then asks you how large a partition to make (Figure 12-29). You may make the partition any size you want by typing in a number, from a minimum of 8 MB up to the size of the

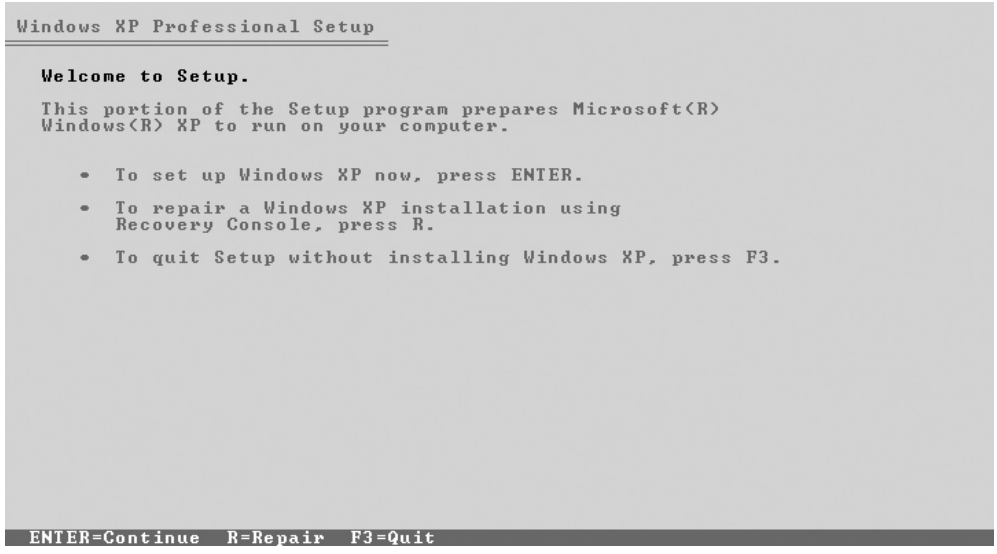


Figure 12-27 Welcome to Setup

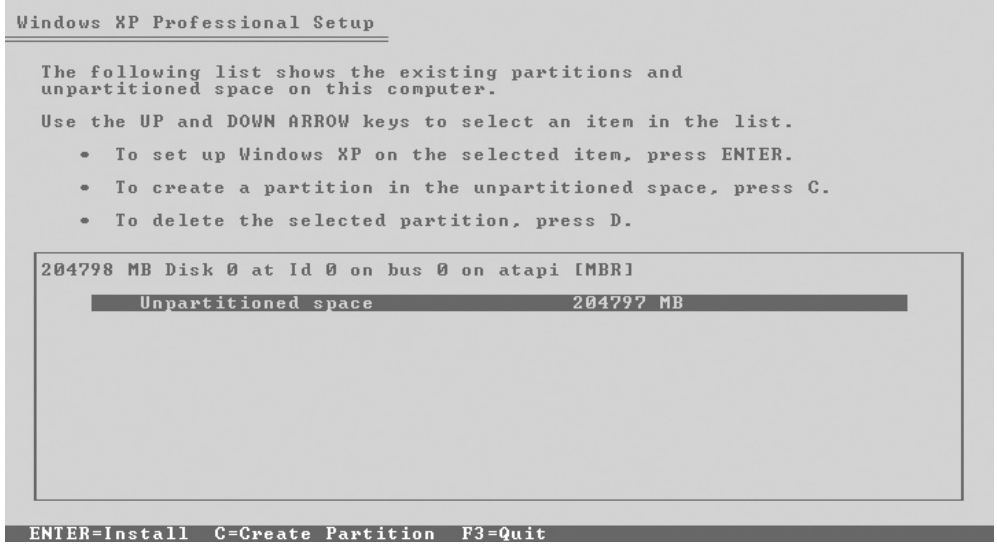


Figure 12-28 Partitioning screen

entire drive (in this case, 204789 MB). Let's just make the entire drive a single C: drive by pressing ENTER.

Ta-da! You just partitioned the drive! Now Windows asks you how you want to format that drive (Figure 12-30). So you might be asking, where's the basic versus dynamic?

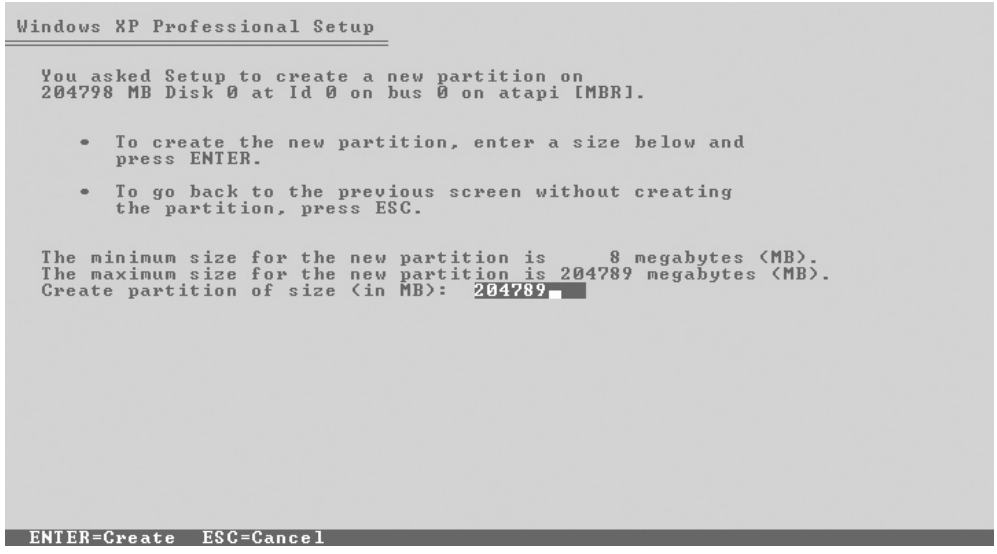


Figure 12-29 Setting partition size

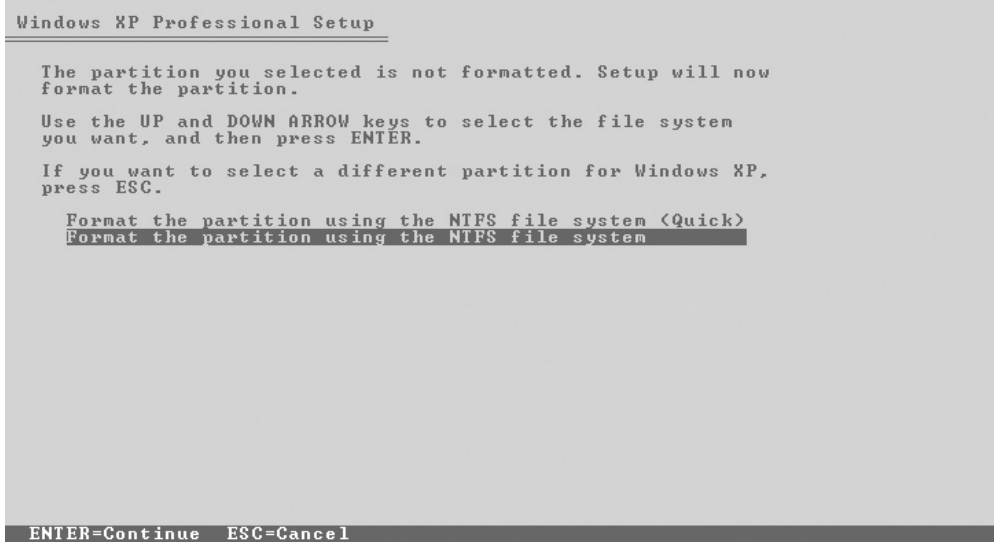


Figure 12-30 Format screen

Where do you tell Windows to make the partition primary instead of extended? Where do you set it as active?

The Windows installer makes a number of assumptions for you, such as always making the first partition primary and setting it as active. The installer also makes all hard drives basic disks. You'll have to convert it to dynamic later (if you even want to convert it at all).

Select NTFS for the format. Either option—quick or full—will do the job here. (Quick format is quicker, as the name would suggest, but the full option is more thorough and thus safer.) After Windows formats the drive, the installation continues, copying the new Windows installation to the C: drive.

Two Partitions

Well, that was fun! So much fun that I'd like to do another new Windows installation, with a bit more complex partitioning. This time, you again have the 200-GB hard drive, but you want to split the drive into three drive letters of roughly 66 GB each. That means you need to make a single 66-GB primary partition, then a 133-GB extended partition, and then split that extended partition into two logical drives of 66 GB each.

Back at the Windows installation main partitioning screen, first press C to make a new partition, but this time change the 204789 to 66666, which will give you a partition of about 66 GB. When you press ENTER, the partitioning screen should look like Figure 12-31. Even though the installation program doesn't tell you, the partition is primary.

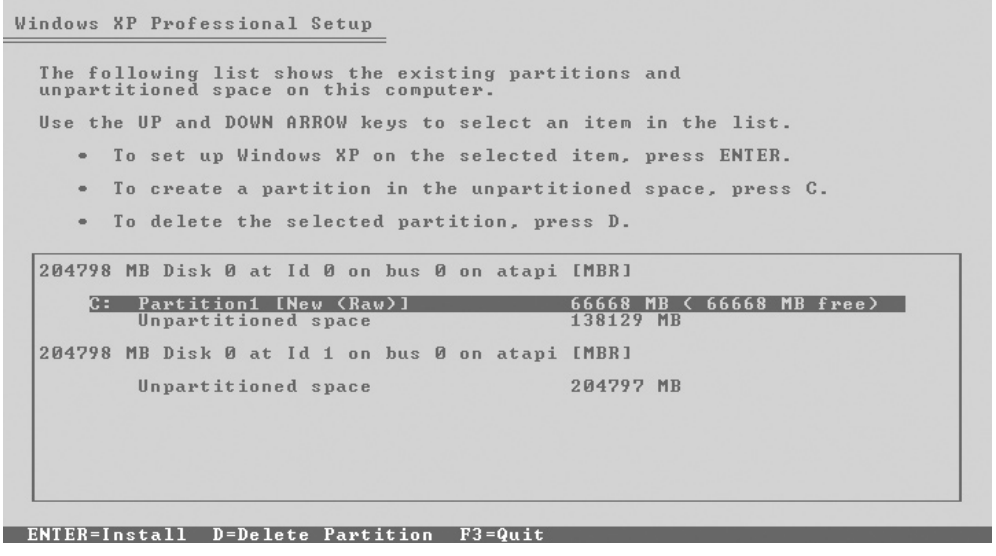


Figure 12-31 You've created a 66-GB partition.

NOTE Windows almost always adjusts the number you type in for a partition size. In this case, it changed 66666 to 66668, a number that makes more sense when translated to binary. Don't worry about it!

Notice that two-thirds of the drive is still unpartitioned space. Move the selection down to this option and press C to create the next partition. Once again, type **66666** in the partition size screen and press ENTER, and you'll see something similar to Figure 12-32.

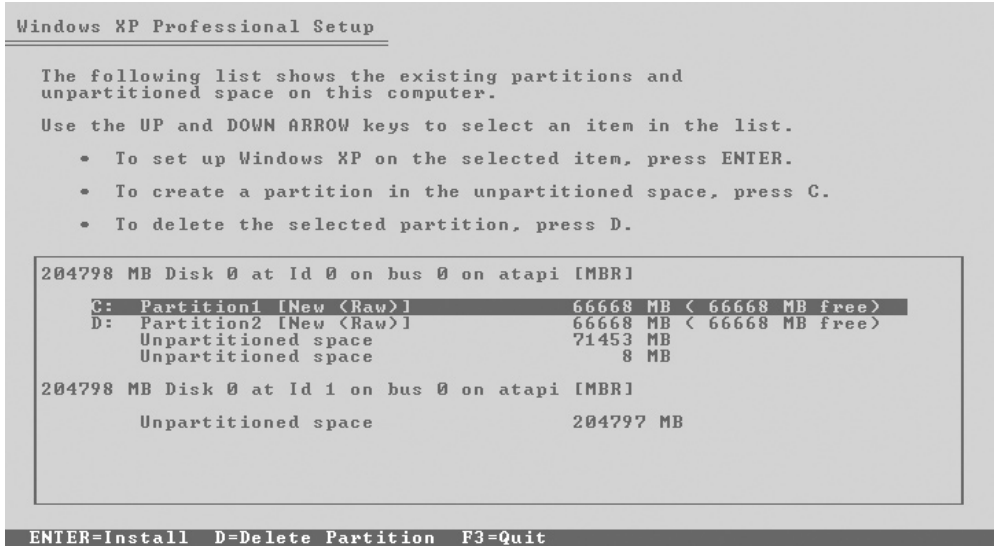


Figure 12-32 Second partition created

Create your last partition exactly as you made the other two to see your almost-completely partitioned drive (Figure 12-33). (Note that the example is not realistic in one respect: you would never leave any unpartitioned space on a drive in a typical PC.)

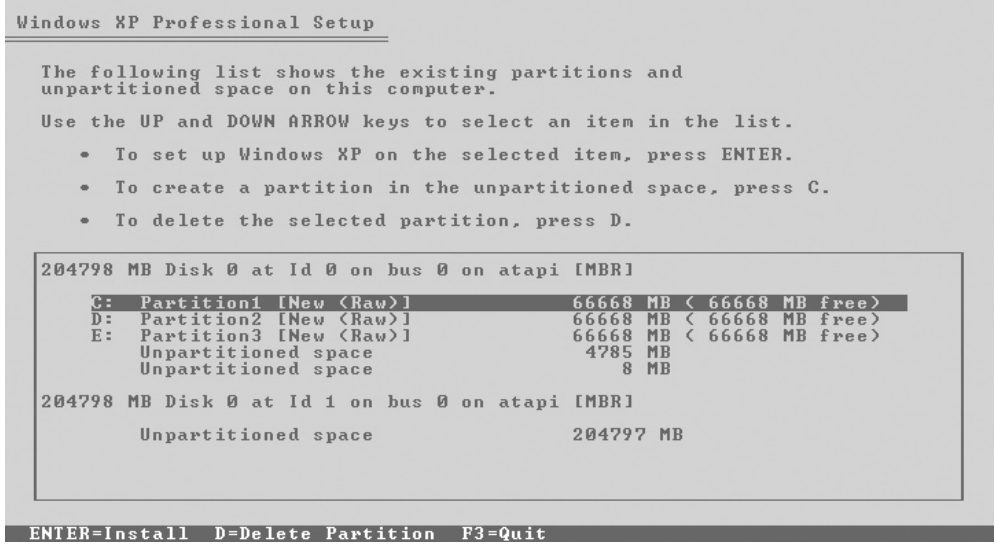


Figure 12-33 Fully partitioned drive

Even though the Windows installation shows that you've made three partitions, you've really made only two: the primary partition, which is C:, and then two logical drives (D: and E:) in an extended partition. Once again, the next step, formatting, is saved for a later section in this chapter.

You've just created three drive letters. Keep in mind that the only drive you must partition during installation is the drive on which you install Windows.

The installation program can delete partitions just as easily as it makes them. If you use a hard drive that already has partitions, for example, you just select the partition you wish to delete and press the letter D. This brings up a dialog box where Windows gives you one last change to change your mind (Figure 12-34). Press L to kill the partition.

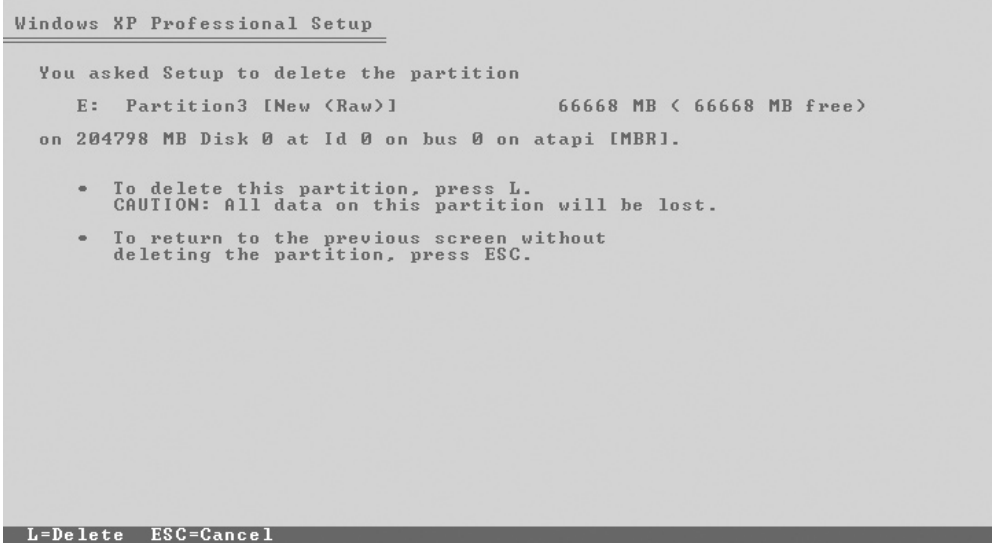


Figure 12-34 Option to delete partition

Partitioning and Formatting with the Windows Vista Installation DVD

Among the many changes in Microsoft's newest operating system is a completely revamped installation process, complete with a fancy looking and, more importantly, easy-to-use graphical user interface. Again, the entire installation process will be covered in Chapter 14, "Installing and Configuring Windows," but because you've already looked at partitioning in Windows XP, you should at least be familiar with what's changed in Vista.

Single Partition

One thing that definitely hasn't changed with Vista is that the most common installation is on a single active partition, so let's start there. Again, you're going to partition and format a single 200-GB drive.

The Vista installation GUI has a few more steps than XP before you get to the actual formatting page, so let's get through those as quickly as possible to get to the fun stuff. When you boot from the installation DVD, you'll be greeted with a screen asking you for language and regional information (Figure 12-35). Unless you're having this book read to you by a translator, I expect you'll want to keep the language set to English, but set the other entries as needed.

Figure 12-35 The Windows Vista language preferences screen

The next page has a large Install Now button, so click that and move on. After that, the installer asks for a product key (Figure 12-36). Don't bother entering one yet—just leave the field blank and click Next to move on to the next page.

The next page asks which version of Vista you want to install (Figure 12-37). Every Vista installation DVD contains all editions of the operating system—your product key ultimately determines which edition you can install, so you wouldn't get this page if you entered a product key when first prompted. Select Windows Vista ULTIMATE (in this example) and move on to the next page.

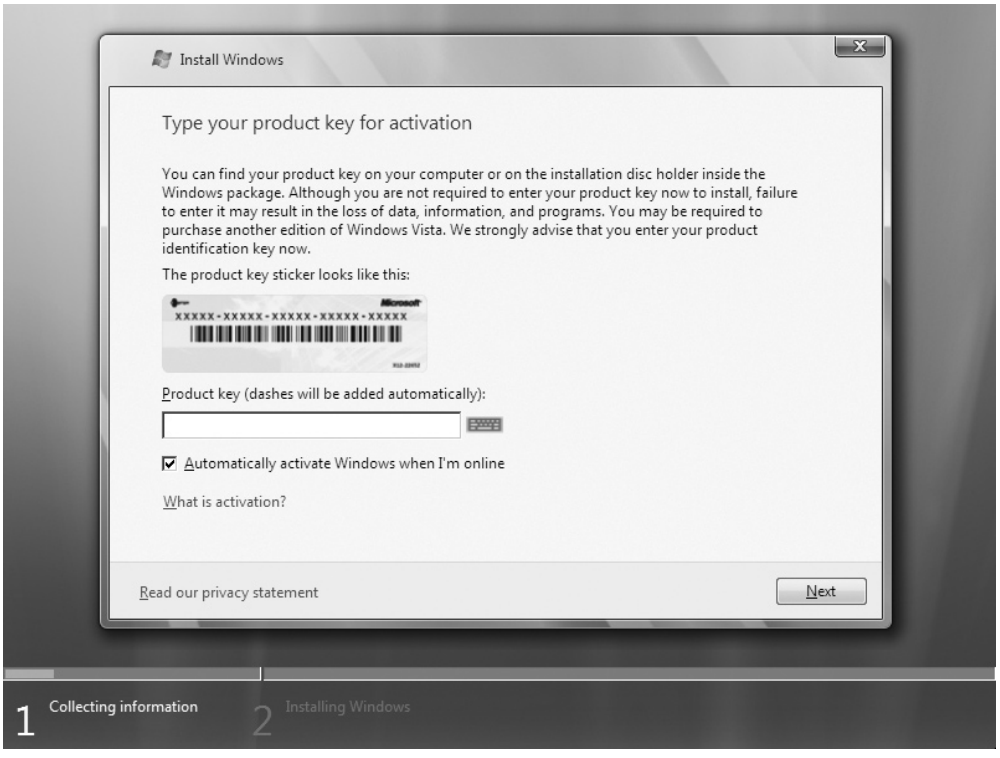


Figure 12-36 The product key page

The next page is just a license agreement that you'll need to, ahem, agree with before moving on. Getting impatient to do some formatting? Don't worry—I know this process is a tad longer than on XP, but you're almost there. Click the Custom install button on the next page and you'll be greeted with the partitioning page (Figure 12-38). Whew!

Your hard drive is the bar that says Disk 0 Unallocated Space, which is currently the only thing there. If you just click Next, Windows automatically partitions and formats the drive for you, but I still fail to see any fun in that, so let's once again manually create a partition on the drive. Click the *Drive options* (*advanced*) button to see the advanced drive features. To create a new partition, click the New button. You could simply click Apply to make a 200-GB partition, but, to demonstrate one of Vista's handy new features, type **100000** and then click Apply (Figure 12-39).

Once you have created your 100-GB partition, click the Format button. Notice that the installer never asks you what file system to use. Vista can read FAT drives, but it will not install itself to one by default. There are, of course, some people on the Internet who have figured out how to install Vista to a FAT32 drive, but why anyone would want to lose all of NTFS's functionality is beyond me.

Figure 12-37 Choose your edition

So now you have set up a 100-GB partition, but what if you want to make it a 200-GB partition? In XP, you would have to delete the partition and start over, but not so in Vista. You can simply click the Extend button and then apply the rest of the unallocated space to your currently formatted partition. The extend function allows you to easily tack unpartitioned space onto an already partitioned drive.

Multiple Partitions

You can format a drive to contain two partitions just as easily as formatting a single drive. Just as in the XP example, you'll be creating three 66-GB partitions. Unlike in Windows XP, this process actually leaves you with two primary partitions, not a primary partition and an extended partition with two logical drives. Vista will not create extended partitions if a user has fewer than four partitions on a drive, so if you're making three partitions, you're actually creating three primary partitions. If you made a fourth, it would manifest itself as a logical drive on an extended partition.

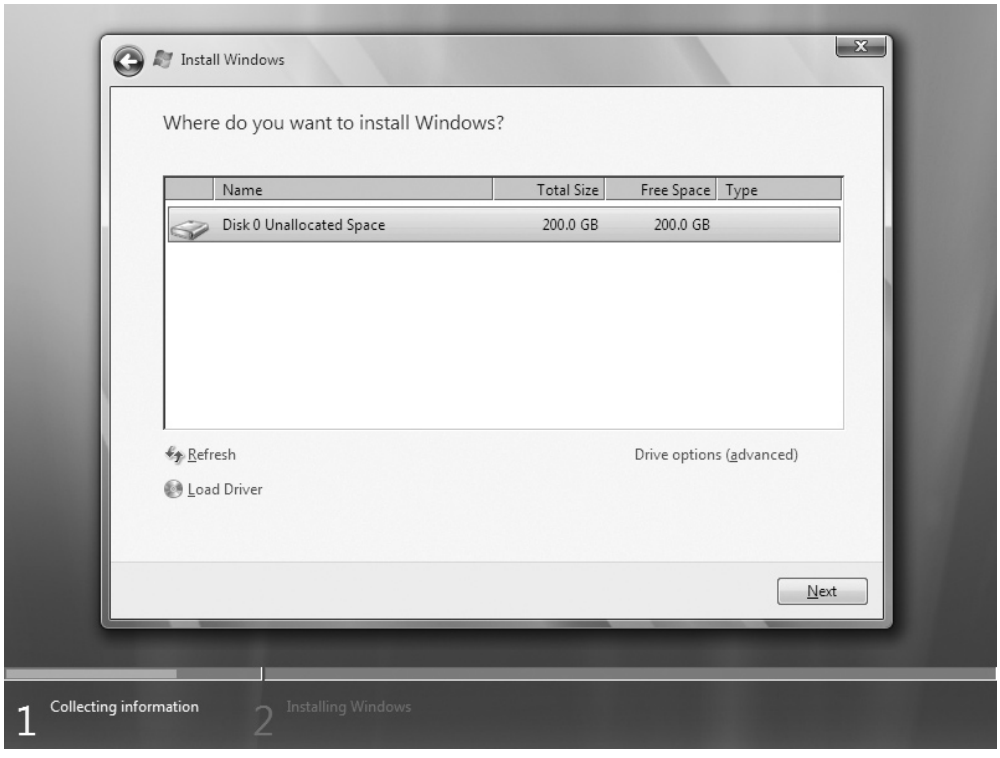


Figure 12-38 The Vista partitioning page

However, you're not going to create four partitions for this exercise, so you don't need to worry about that. You'll start out, again, with a 200-GB drive, but this time, after clicking the New button, type **66666** into the Size box and click Apply. That will give you a 66-GB (more or less) primary partition (Figure 12-40).

Do the same thing to create the next two drives and you're finished. That was pretty easy, wasn't it?

Partitions and Drive Letters

So you have a hard drive, maybe several hard drives, all partitioned up, and you've installed Windows on one of them, but where do those drive letters come from? Older systems assigned drive letters based on some fairly complicated rules having to do with master and slave drives, but things are much simpler on modern systems.

The primary active partition will always be C: and you can't change that, but the rest of the drives are assigned the next available letter, with hard drives taking priority over optical drives. If you have two hard drives and an optical drive in your computer, the hard drives will be C: and D:, and the optical drive will be E:. If, however, you later install another hard drive in the computer, it will become your F: drive. Newly installed drives do not take drive letters from previously installed drives.

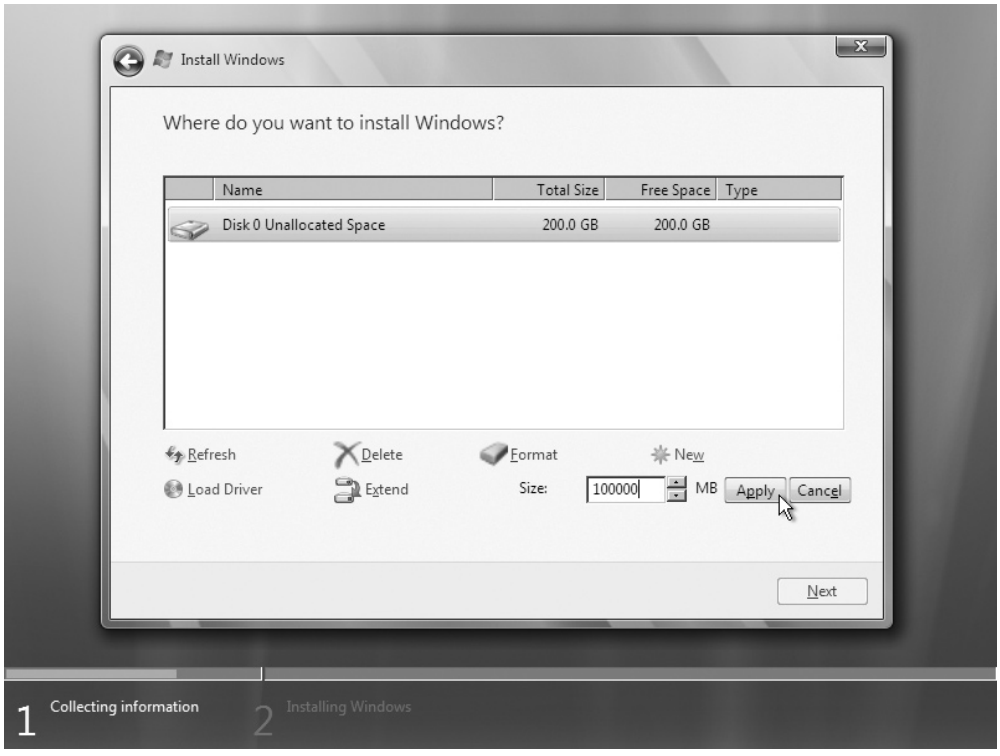


Figure 12-39 Setting partition size

You can change the lettering on every drive but your system partition, which you'll find out how to do in the next section.

Disk Management

The real tool for partitioning and formatting is the Disk Management utility. You can use Disk Management to do everything you want to do to a hard drive in one handy tool. You can access Disk Management by going to the Control Panel and opening the Computer Management applet. If you're cool, you can click Start | Run, type in **diskmgmt.msc**, and press ENTER. Windows 2000/XP and Windows Vista/7 come with Disk Management (Figure 12-41).

Disk Management works only within Windows, so you can't use Disk Management from a boot device. If you install Windows from an installation disc, in other words, you must use the special partitioning/formatting software built into the installation program you just saw in action.

One of the most interesting parts of Disk Management is disk initialization. Every hard drive in a Windows system has special information placed onto the drive. This initialization information includes identifiers that say "this drive belongs in this system" and other information that defines what this hard drive does in the system. If the hard drive is part of a RAID array, its RAID information is stored in the initialization. If it's part of a spanned volume, this is also stored there. All new drives must be initialized before you

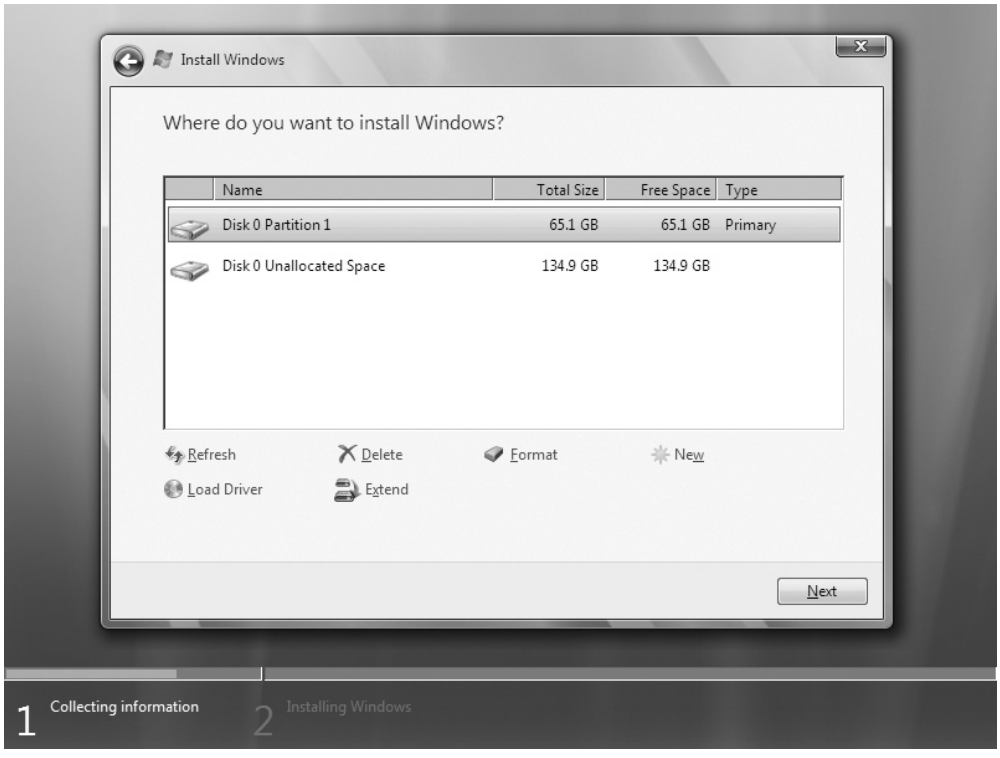


Figure 12-40 The first 66-GB partition

can use them. When you install an extra hard drive into a Windows system and start Disk Management, it notices the new drive and starts the Hard Drive Initialization Wizard. If you don't let the wizard run, the drive will be listed as unknown (Figure 12-42).

To initialize a disk, right-click the disk icon and select Initialize. Once a disk is initialized, you can see the status of the drive—a handy tool for troubleshooting.

Disk Management enables you to view the status of every drive in your system. Hopefully, you'll mostly see the drive listed as Healthy, meaning that nothing is happening to it and things are going along swimmingly. You're also already familiar with the Unallocated and Active status, but here are a few more to be familiar with for the test:

- **Foreign drive** You see this when you move a dynamic disk into another computer.
- **Formatting** As you might have guessed, you see this when you're formatting a drive.
- **Failed** Pray you never see this status, because it means that the disk is damaged or corrupt and you've probably lost some data.
- **Online** This is what you see if a disk is healthy and communicating properly with the computer.
- **Offline** The disk is either corrupted or having communication problems.

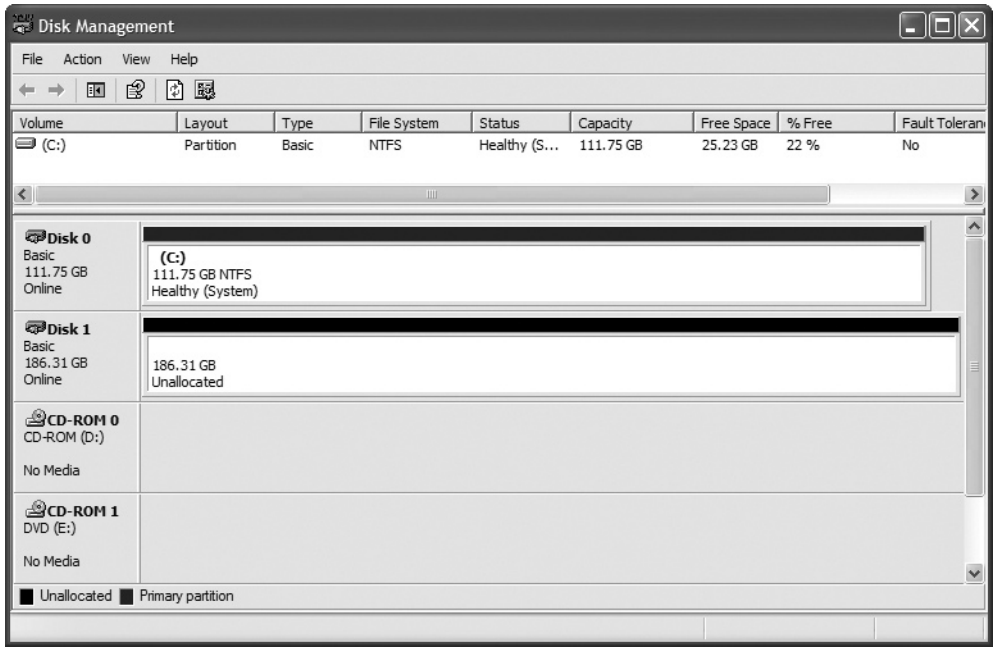


Figure 12-41 Disk Management

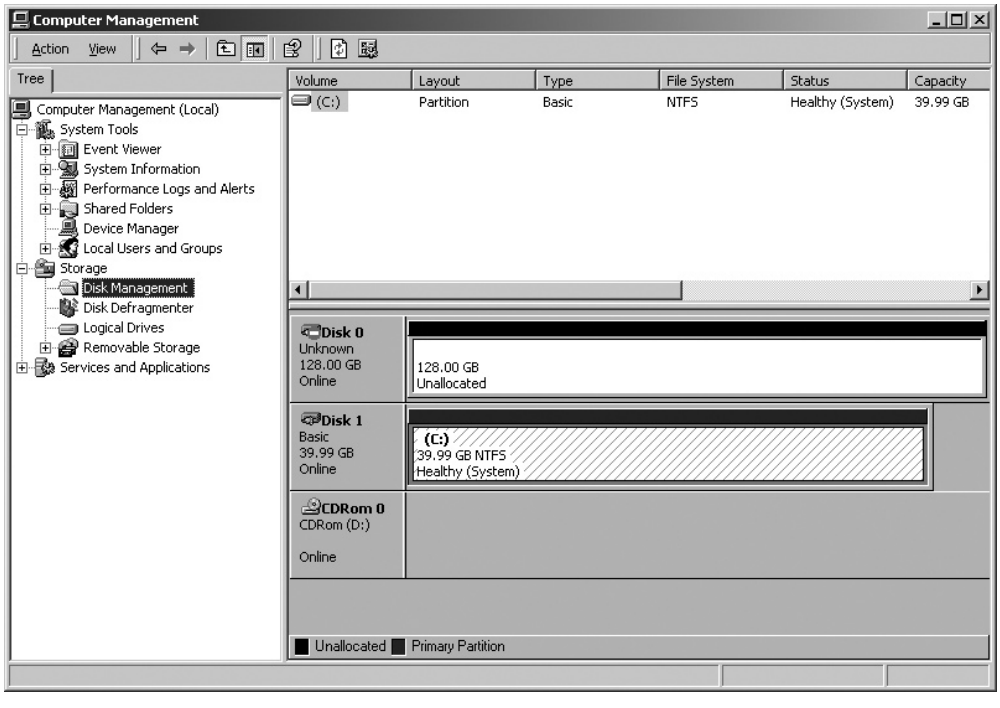


Figure 12-42 Unknown drive in Disk Management

A newly installed drive is always set as a basic disk. There's nothing wrong with using basic disks, other than that you miss out on some handy features. To create partitions, right-click the unallocated part of the drive and select New Partition. Disk Management runs the New Partition Wizard, with which you can select a primary or extended partition (Figure 12-43). Afterward, you see a screen where you specify the size partition you prefer (Figure 12-44).

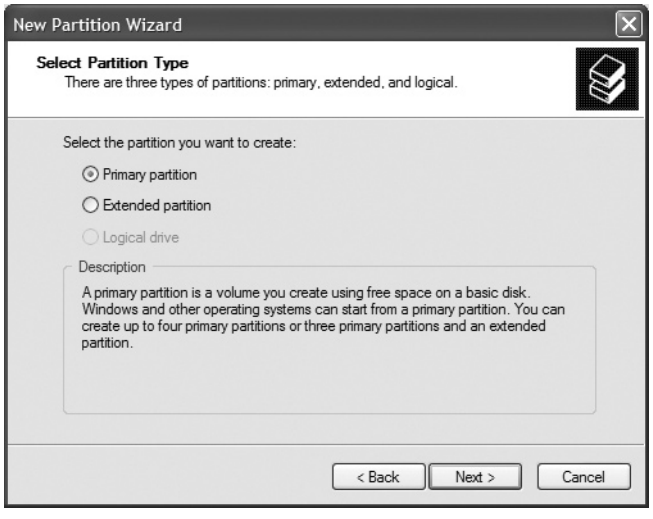


Figure 12-43 The New Partition Wizard

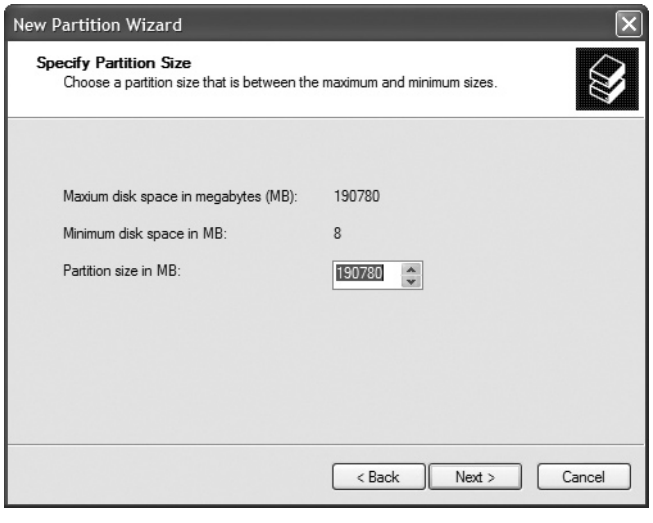


Figure 12-44 Specifying the partition size

If you choose to make a primary partition, the wizard asks if you want to assign a drive letter to the partition, mount it as a folder to an existing partition, or do neither (Figure 12-45). (If you choose to make an extended partition, you just get a confirmation screen and you are returned to Disk Management.) In almost all cases, you'll want to give primary partitions a drive letter.

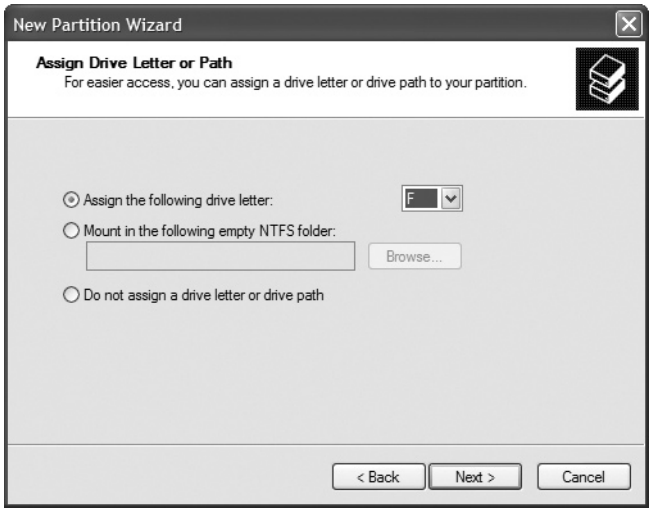


Figure 12-45 Assigning a drive letter to a primary partition

The last screen of the New Partition Wizard asks for the type of format you want to use for this partition (Figure 12-46). If your partition is 4 GB or less, you may format it as FAT, FAT32, or NTFS. If your partition is greater than 4 GB but less than 32 GB, you can make the drive FAT32 or NTFS. Windows requires NTFS on any partition greater than 32 GB. Although FAT32 supports partitions up to 2 TB, Microsoft wants you to use NTFS on larger partitions and creates this limit. In today's world of big hard drives, there's no good reason to use anything other than NTFS.

NOTE Windows 2000/XP and Windows Vista/7 read and write to FAT32 partitions larger than 32 GB; they just don't allow Disk Management to make them. If you ever stumble across a drive from a system that ran the old Windows 9x/Me that has a FAT32 partition larger than 32 GB, it will work just fine in a modern Windows system.

You have a few more tasks to complete at this screen. You can add a volume label if you want. You can also choose the size of your clusters (Allocation unit size). There's no reason to change the default cluster size, so leave that alone—but you can sure speed up the format if you select the Perform a quick format checkbox. This will format your drive without checking every cluster. It's fast and a bit risky, but new hard drives almost always come from the factory in perfect shape—so you must decide whether to use it or not.

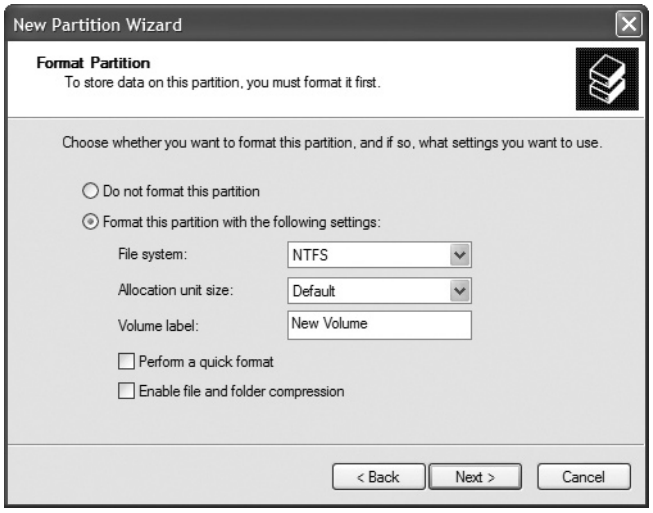


Figure 12-46 Choosing a file system type

Last, if you chose NTFS, you may enable file and folder compression. If you select this option, you'll be able to right-click any file or folder on this partition and compress it. To compress a file or folder, choose the one you want to compress, right-click, and select Properties. Then click the Advanced button to turn compression on or off (Figure 12-47). Compression is handy for opening up space on a hard drive that's filling up, but it also slows down disk access, so use it only when you need it.

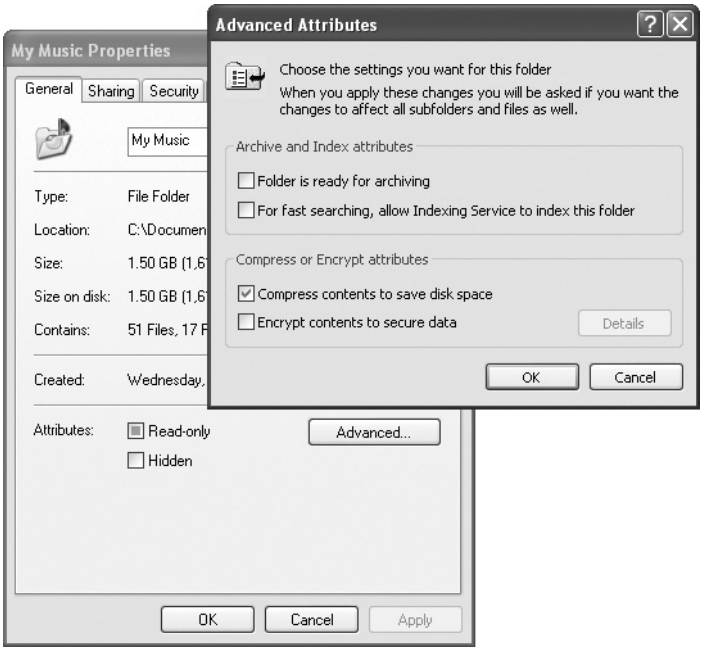


Figure 12-47 Turning on compression

After the drive finishes formatting, you'll go back to Disk Management and see a changed hard drive landscape. If you made a primary partition, you will see your new drive letter. If you made an extended partition, things will look a bit different. Figure 12-48 shows the extended partition as free space because it has no logical drive yet. As you can easily guess from Figure 12-49, to create a logical drive, simply right-click in that extended partition and choose New Logical Drive. Disk Management fires up the New Partition Wizard again, this time with the option to create a logical drive.

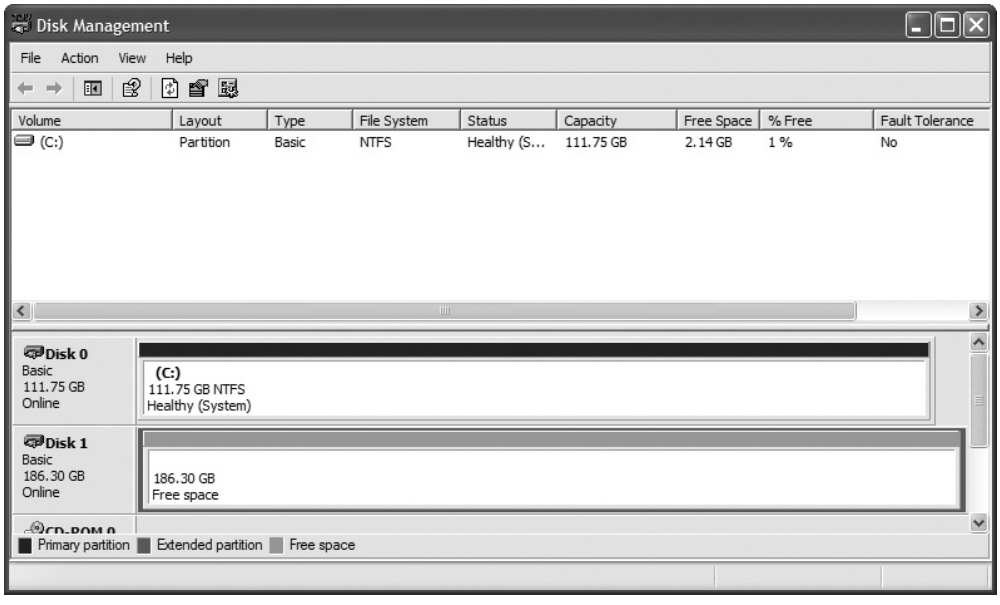


Figure 12-48 Extended partition with no logical drives

NOTE In Windows Vista, you cannot create extended partitions unless you already have three primary partitions on a drive and are creating a fourth. Microsoft has tried to simplify drive implementation as much as possible.

When you create a logical drive, the New Partition Wizard automatically gives you the same options to format the partition by using one of the three file systems you saw earlier with primary partitions (Figure 12-50). You get another confirmation screen, and then the Disk Management console shows you the newly created drive.

One interesting aspect of Windows is the tiny (approximately 8 MB) mysterious unallocated partition that shows up on the C: drive. The Windows installation program does this when you first install Windows on a new system, to reserve a space Windows needs for converting the C: drive to a dynamic disk. It doesn't hurt anything and it's tiny, so just leave it alone. If you want to make a volume and format it, feel free to do so.

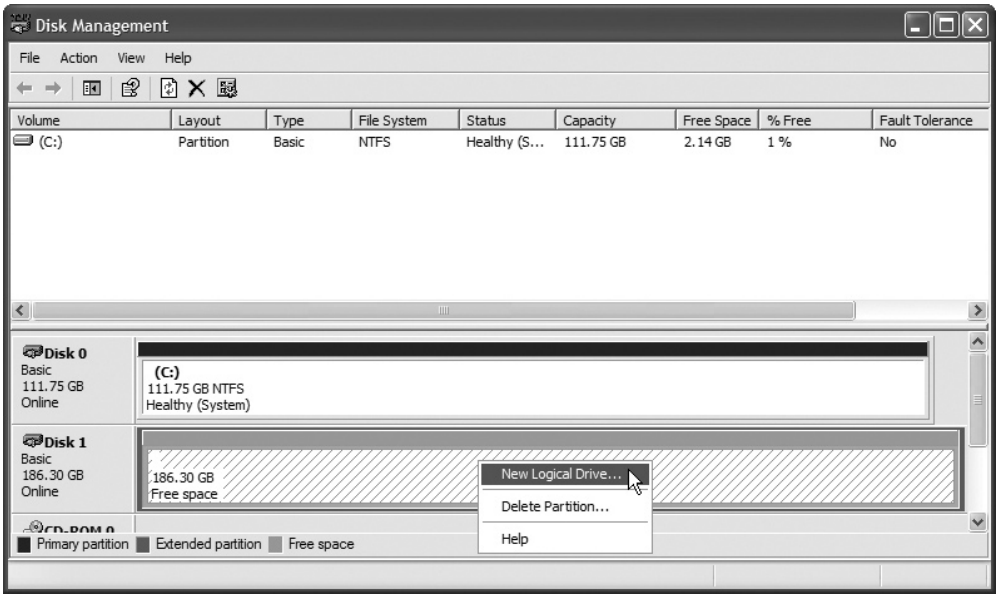
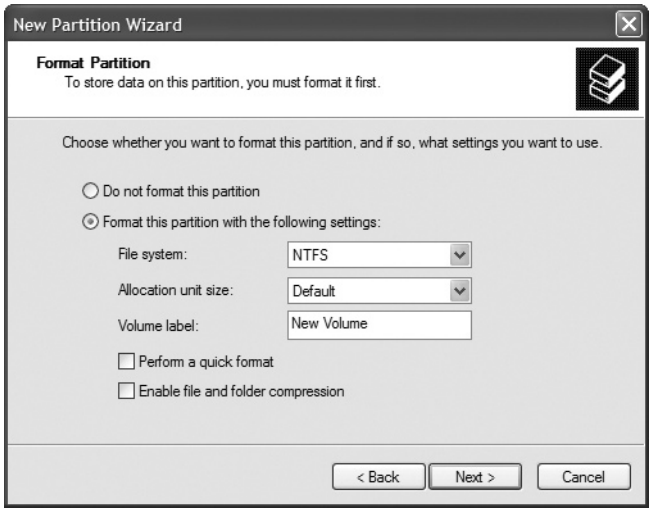


Figure 12-49 Selecting to create a logical drive in the extended (free space) partition

Figure 12-50 The New Partition Wizard offering formatting options



Dynamic Disks

You create dynamic disks from basic disks in Disk Management. Once you convert a drive from a basic to a dynamic disk, primary and extended partitions no longer exist; dynamic disks are divided into volumes instead of partitions.

EXAM TIP When you move a dynamic disk from one computer to another, it shows up in Disk Management as a foreign drive. You can import a foreign drive into the new system by right-clicking the disk icon and selecting Import Foreign Disks.

To convert a basic disk to dynamic, just right-click the drive icon and select Convert to Dynamic Disk (Figure 12-51). The process is very quick and safe, although the reverse is not true. The conversion from dynamic disk to basic disk first requires you to delete all partitions off of the hard drive.

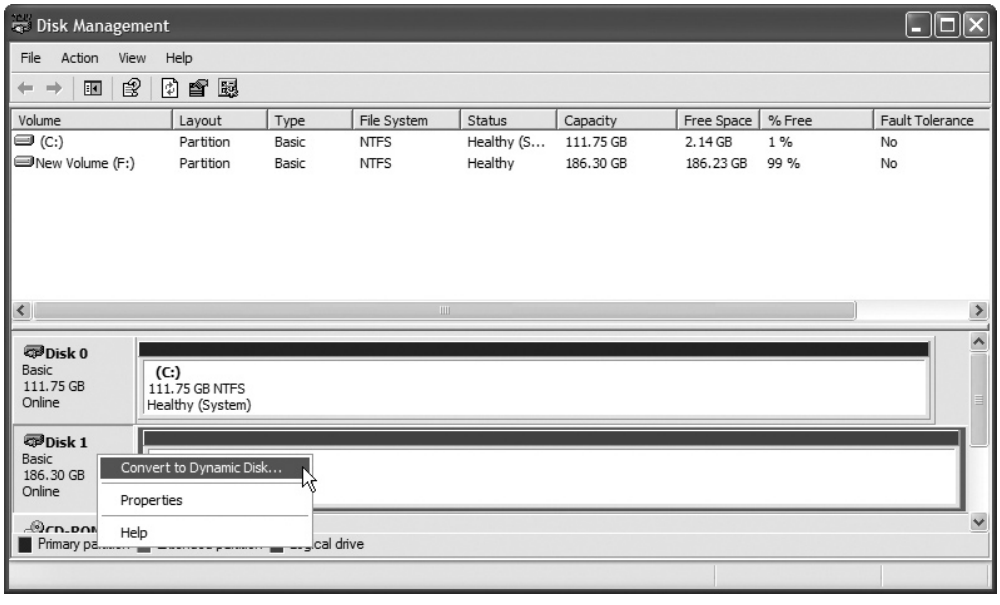


Figure 12-51 Converting to a dynamic disk

Once you've converted, no partitions exist, only volumes. You can make five types of volumes on a dynamic disk: simple, spanned, striped, mirrored, and RAID 5, although you'll commonly see only the first three in a Windows 2000/XP Professional or Windows Vista Business environment. You'll next learn how to implement the three most common volume types. The final step involves assigning a drive letter or mounting the volume as a folder.

EXAM TIP The home editions of Windows XP and Windows Vista do not support dynamic disks.

Simple Volumes

A simple volume acts just like a primary partition. If you have only one dynamic disk in a system, it can have only a simple volume. It's important to note here that a simple volume may act like a traditional primary partition, but it is very different. If you install a hard drive partitioned as a simple volume dynamic disk into any version of Windows prior to Windows 2000, you would see no usable partition.

In Disk Management, right-click any unallocated space on the dynamic disk and choose New Volume (Figure 12-52) to run the New Volume Wizard. You'll see a series of screens that prompt you on size and file system, and then you're finished. Figure 12-53 shows Disk Management with three simple volumes.

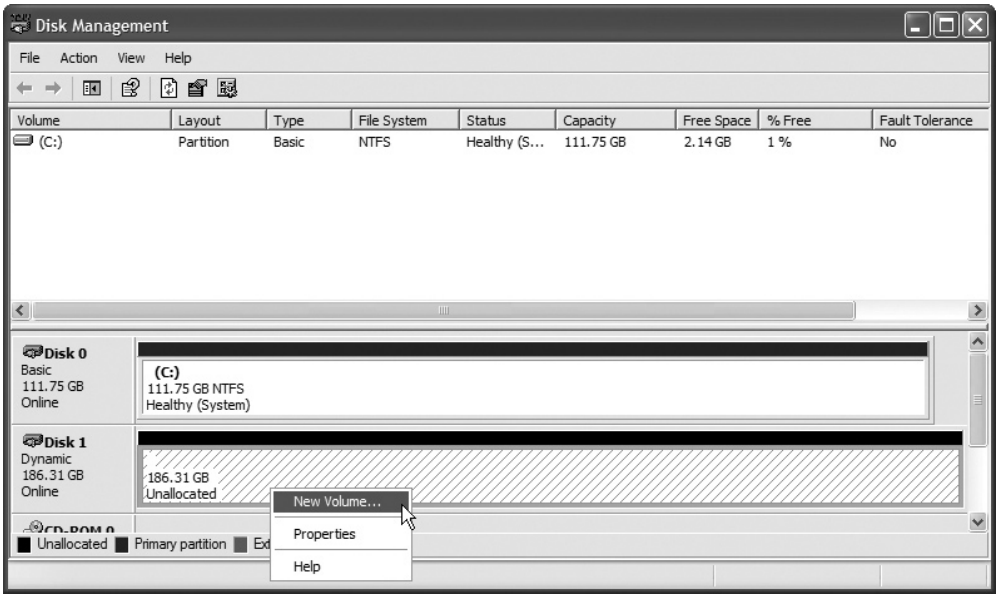


Figure 12-52 Selecting to open the New Volume Wizard

Spanning Volumes

You can extend the size of a simple volume to any unallocated space on a dynamic disk. You can also extend the volume to grab extra space on completely different dynamic disks, creating a spanned volume. To extend or span, simply right-click the volume you want to make bigger, and choose Extend Volume from the options (Figure 12-54). This opens the Extend Volume Wizard, which prompts you for the location of free space on a dynamic disk and the increased volume size you want to assign (Figure 12-55). If you have multiple drives, you can span the volume just as easily to one of those drives.

The capability to extend and span volumes makes dynamic disks worth their weight in gold. If you start running out of space on a volume, you can simply add another

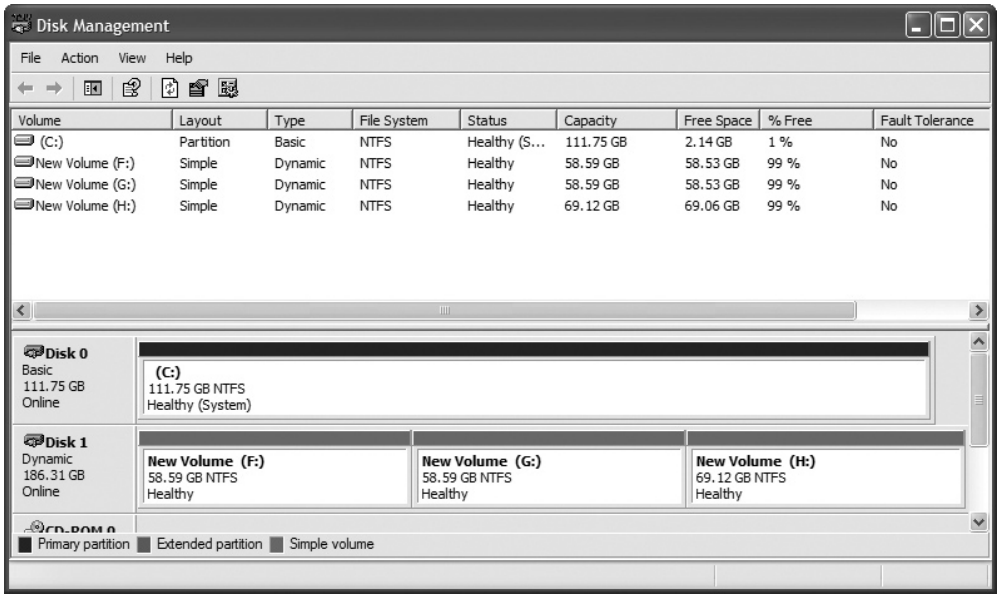


Figure 12-53 Simple volumes

physical hard drive to the system and span the volume to the new drive. This keeps your drive letters consistent and unchanging so your programs don't get confused, yet enables you to expand drive space when needed.

CAUTION Once you convert a drive to dynamic, you cannot revert it to a basic disk without losing all of the data on that drive. Be prepared to back up all data before you convert.

You can extend or span any simple volume on a dynamic disk, not just the "one on the end" in the Disk Management console. You simply select the volume to expand and the total volume increase you want. Figure 12-56 shows a simple 4-GB volume named Extended that has been enlarged an extra 7.91 GB in a portion of the hard drive, skipping the 2-GB section of unallocated space contiguous to it. This created an 11.91-GB volume. Windows has no problem skipping areas on a drive.

NOTE You can extend and shrink hard drives in Windows Vista/7 without using dynamic disks. You can shrink any primary partition with available free space (though you can't always shrink the partition by the whole amount of free space, based on the location of unmovable sectors such as the MBR), and you can expand partitions with unpartitioned space on the drive. Using dynamic disks in Vista/7 still has benefits, however, because you cannot expand a partition by using space on another drive, and the unpartitioned space has to be contiguous with the partition you're expanding.

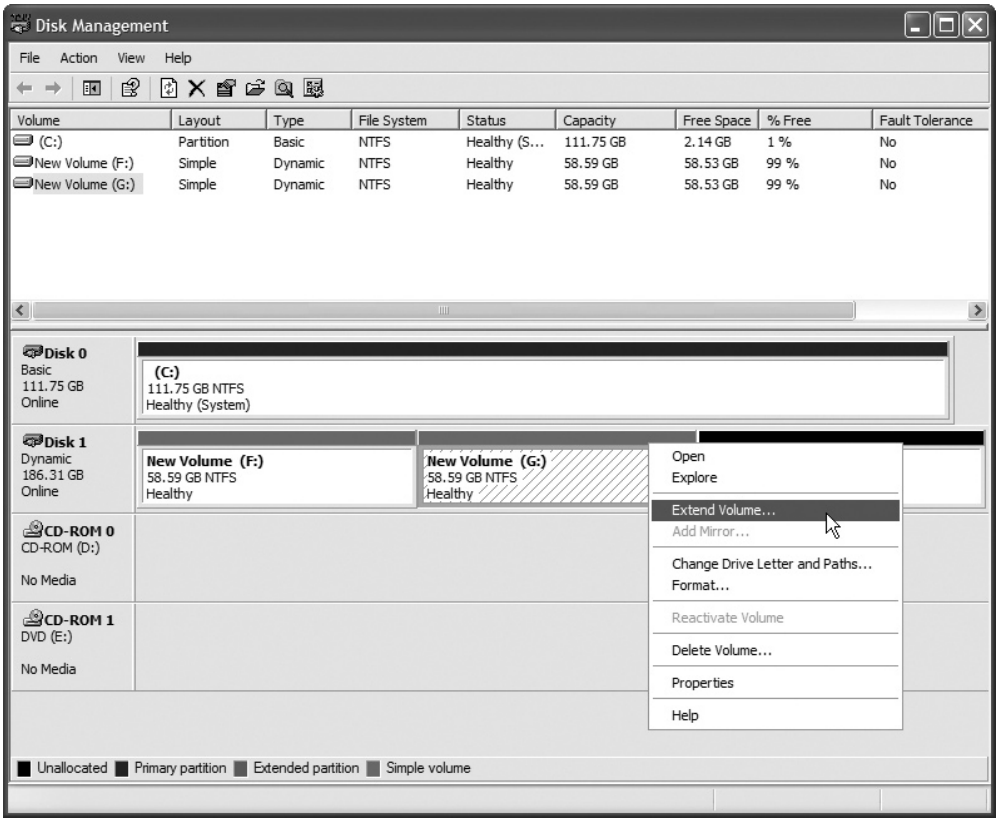


Figure 12-54 Selecting the Extend Volume option

Striped Volumes

If you have two or more dynamic disks in a PC, Disk Management enables you to combine them into a *striped* volume. A striped volume spreads out blocks of each file across multiple disks. Using two or more drives in a group called a *stripe set*, striping writes data first to a certain number of clusters on one drive, then on the next, and so on. It speeds up data throughput because the system has to wait a much shorter time for a drive to read or write data. The drawback of striping is that if any single drive in the stripe set fails, all data in the stripe set is lost.

To create a striped volume, right-click unused space on a drive and choose New Volume and then Striped. The wizard asks for the other drives you want to add to the stripe, and you need to select two unallocated spaces on other dynamic disks. Select the other unallocated spaces and go through the remaining screens on sizing and formatting until

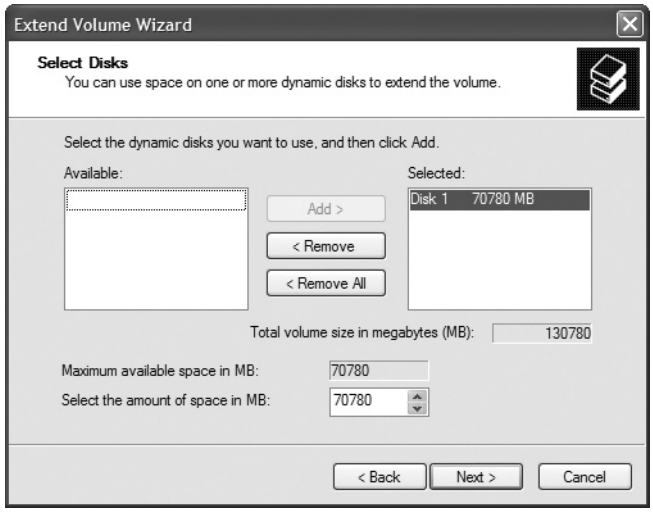


Figure 12-55 The Extend Volume Wizard

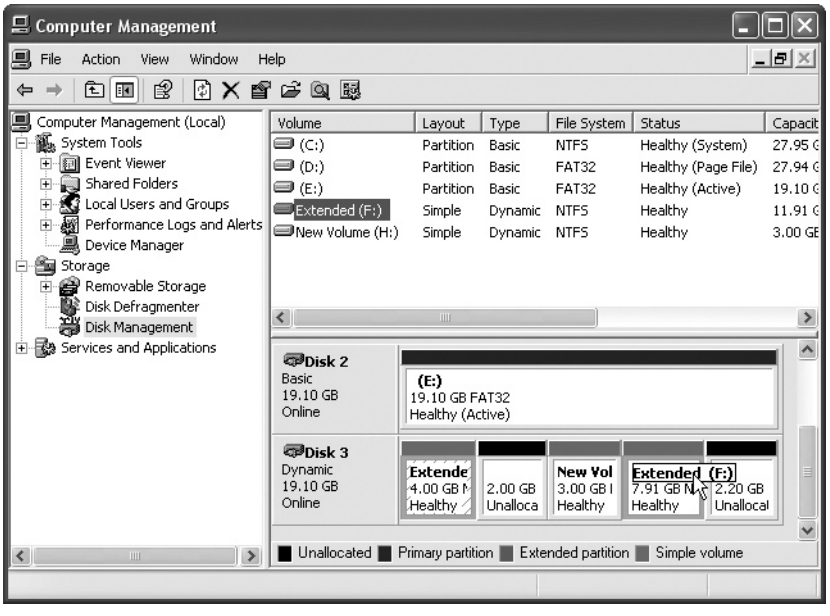


Figure 12-56 Extended volume

you've created a new striped volume (Figure 12-57). The two stripes in Figure 12-57 appear to have different sizes, but if you look closely you'll see they are both 4 GB. All stripes must be the same size on each drive.

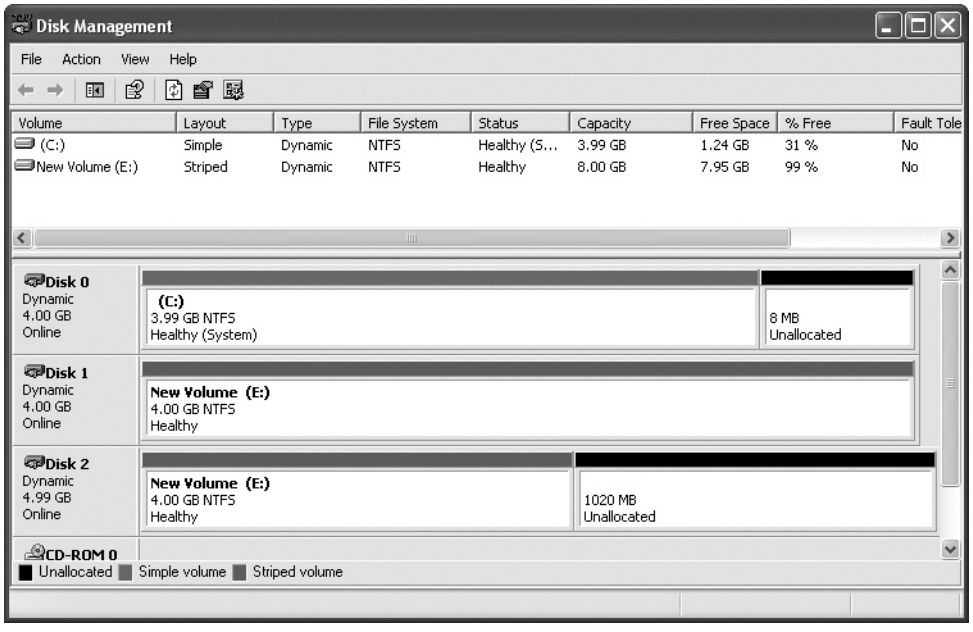


Figure 12-57 Two striped drives

Mount Points

The one drive that can't take full advantage of being dynamic is the drive containing the operating system, your primary master C: drive. You can make it dynamic, but you still can't do all of the cool dynamic things such as extending and spanning. So what good is being able to allocate more space to a volume if you can't use it when you start to fill up your C: drive? If you can't add to that drive, your only option is to replace it with a new, bigger drive, right?

Not at all! Earlier we discussed the idea of mounting a drive as a folder instead of a drive letter, and here's where you get to do it. A *volume mount point* (or simply *mount point*) is a place in the directory structure of an existing volume that you can point to a volume or partition. The mounted volume then functions just like a folder, but all files stored in that segment of the directory structure will go to the mounted volume. After partitioning and formatting the drive, you don't give it a drive letter; instead, you *mount* the volume to a folder on the C: drive and make it nothing more than just another folder (Figure 12-58). You can load programs to that folder, just as you would to your Program Files folder. You can use it to store data files or backed-up system files. In *function*, therefore, the new hard drive simply extends the capacity of the C: drive, so neither you nor your client need ever trouble yourselves with dealing with multiple drive letters.

To create a mount point, right-click an unallocated section of a dynamic disk and choose New Volume. This opens the New Volume Wizard. In the second screen, you

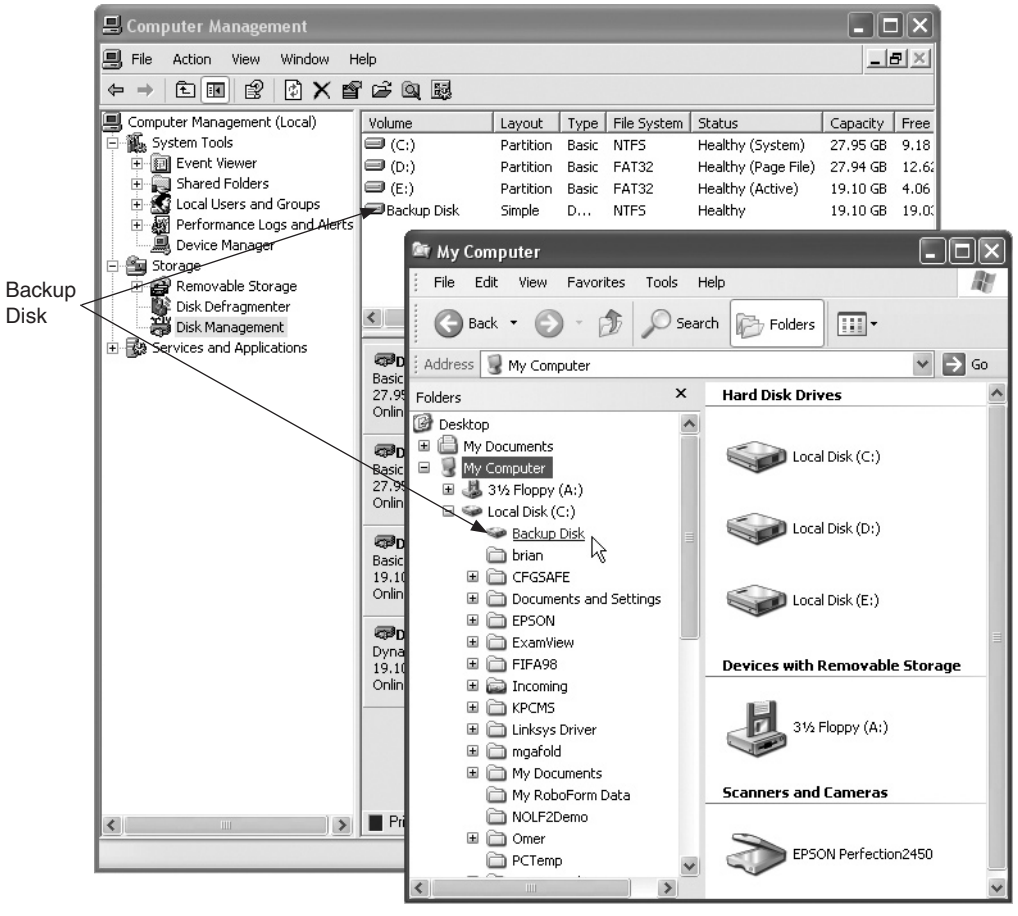


Figure 12-58 A drive volume mounted as a folder of drive C:

can select a mount point rather than a drive letter (Figure 12-59). Browse to a blank folder on an NTFS-formatted drive or create a new folder and you're in business.

With mount points, Microsoft dramatically changed the way you can work with hard drives. You're no longer stuck in the rut of adding drive letters that mess up Windows' mapping of the optical drive. You don't have to confuse clients with multiple drive letters when they just want a little more space. You can resurrect smaller hard drives, making them a functional part of today's computer. With the Disk Management console in Windows 2000/XP and Windows Vista/7, Microsoft got it right.

Formatting a Partition

You can format any Windows partition/volume in My Computer/Computer. Just right-click the drive name and choose Format (Figure 12-62). You'll see a dialog box that asks for the type of file system you want to use, the cluster size, a place to put a volume

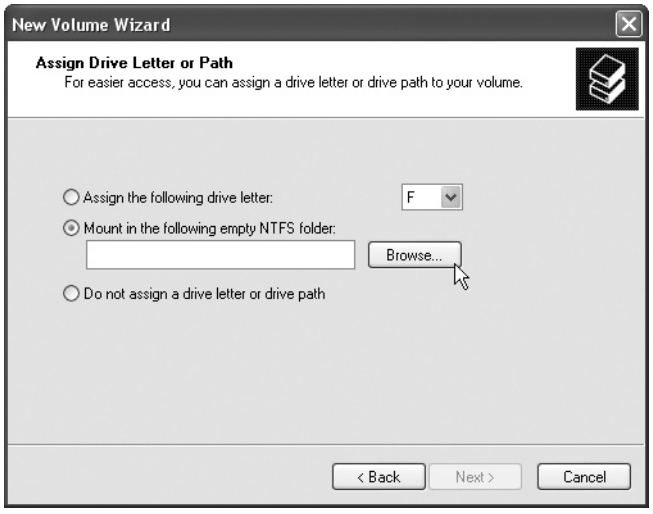


Figure 12-59 Choosing to create a mounted volume

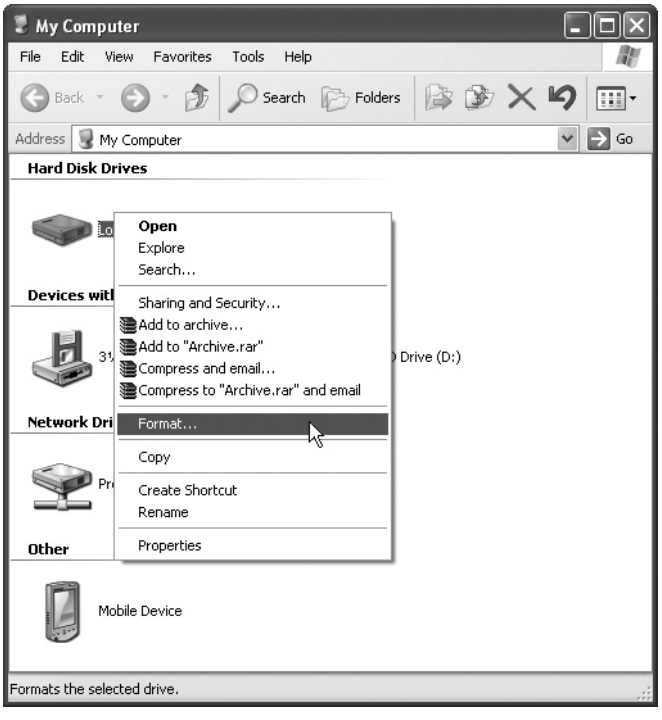


Figure 12-60 Choosing Format in My Computer

label, and two other options. The Quick Format option tells Windows not to test the clusters and is a handy option when you're in a hurry—and feeling lucky. The Enable Compression option tells Windows to give users the capability to compress folders or files. It works well but slows down your hard drive.

Disk Management is today's preferred formatting tool for Windows 2000, XP, and Vista. When you create a new volume on a dynamic disk or a new partition on a basic disk, the New Volume Wizard also asks you what type of format you want to use. Always use NTFS unless you're that rare and strange person who wants to dual-boot Windows XP or Windows Vista with some ancient version of Windows.

All OS installation discs partition and format as part of the OS installation. Windows simply prompts you to partition and then format the drive. Read the screens and you'll do great.

Maintaining and Troubleshooting Hard Drives

Hard drives are complex mechanical and electrical devices. With platters spinning at thousands of rotations per minute, they also generate heat and vibration. All of these factors make hard drives susceptible to failure. In this section, you will learn some basic maintenance tasks that will keep your hard drives healthy, and for those inevitable instances when a hard drive fails, you will also learn what you can do to repair them.

Maintenance

Hard drive maintenance can be broken down into two distinct functions: checking the disk occasionally for failed clusters, and keeping data organized on the drive so it can be accessed quickly.

Error-Checking

Individual clusters on hard drives sometimes go bad. There's nothing you can do to prevent this from happening, so it's important that you check occasionally for bad clusters on drives. The tools used to perform this checking are generically called error-checking utilities, although the terms for two older Microsoft tools—ScanDisk and *CHKDSK* (pronounced "Checkdisk")—are often used. Microsoft calls the tool *Error-checking* in Windows XP/Vista/7. Whatever the name of the utility, each does the same job: when the tool finds bad clusters, it puts the electronic equivalent of orange cones around them so the system won't try to place data in those bad clusters.

EXAM TIP CompTIA A+ uses the term CHKDSK rather than Error-checking.

Most error-checking tools do far more than just check for bad clusters. They go through all of the drive's filenames, looking for invalid names and attempting to fix them. They look for clusters that have no filenames associated with them (we call these *lost chains*) and erase them. From time to time, the underlying links between parent and child folders are lost, so a good error-checking tool checks every parent and child folder. With a folder such as C:\TEST\DATA, for example, they make sure that the folder DATA is properly associated with its parent folder, C:\TEST, and that C:\TEST is properly associated with its child folder, C:\TEST\DATA.

To access Error-checking on a Windows 2000/XP or Windows Vista/7 system, open My Computer/Computer, right-click the drive you want to check, and choose Properties

to open the drive's Properties dialog box. Select the Tools tab and click the Check Now button (Figure 12-61) to display the Check Disk dialog box, which has two options (Figure 12-62). Check the box next to *Automatically fix file system errors*, but save the option to *Scan for and attempt recovery of bad sectors* for times when you actually suspect a problem, because it takes a while on bigger hard drives.

Figure 12-61 The Tools tab in the Properties dialog box in Windows XP

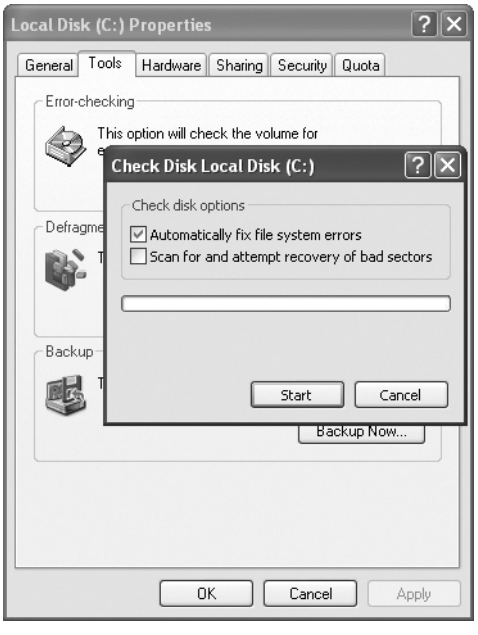


Figure 12-62 Options

Now that you know how to run Error-checking, your next question should be, "How often do I run it?" A reasonable maintenance plan would include running it about once a week. Error-checking is fast (unless you use the *Scan for and attempt recovery* option), and it's a great tool for keeping your system in top shape.

Defragmentation

Fragmentation of clusters can increase your drive access times dramatically. It's a good idea to *defragment*—or *defrag*—your drives as part of monthly maintenance. You access the defrag tool that runs with Windows 2000, XP, Vista, and 7, called Disk Defragmenter, the same way you access Error-checking—right-click a drive in My Computer/Computer and choose Properties—except you click the Defragment Now button on the Tools tab to open the Defragmenter (Figure 12-63).

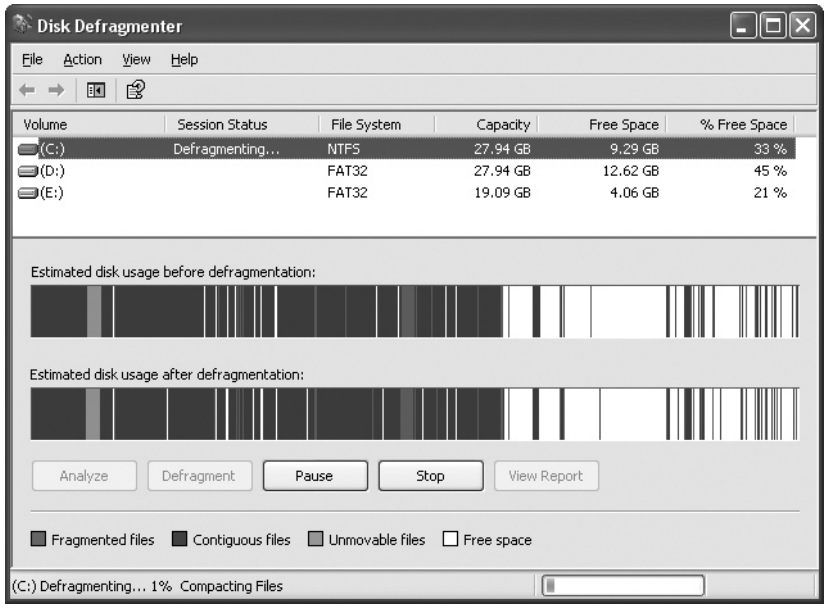


Figure 12-63 Disk Defragmenter in Windows XP

Defragmentation used to be interesting to watch—once. Now, though, just schedule it to run late at night. You should defragment your drives about once a month, although you could run it every week, and if you run it every night, it takes only a few minutes. The longer you go between defrags, the longer it takes. If you don't run Disk Defragmenter, your system will run slower. If you don't run Error-checking, you may lose data.

Disk Cleanup

Did you know that the average hard drive is full of trash? Not the junk you intentionally put in your hard drive such as the 23,000 e-mail messages that you refuse to delete

from your e-mail program. This kind of trash is all of the files that you never see that Windows keeps for you. Here are a few examples:

- **Files in the Recycle Bin** When you delete a file, it isn't really deleted. It's placed in the Recycle Bin in case you decide you need the file later. I just checked my Recycle Bin and found 3 GB worth of files (Figure 12-64). That's a lot of trash!

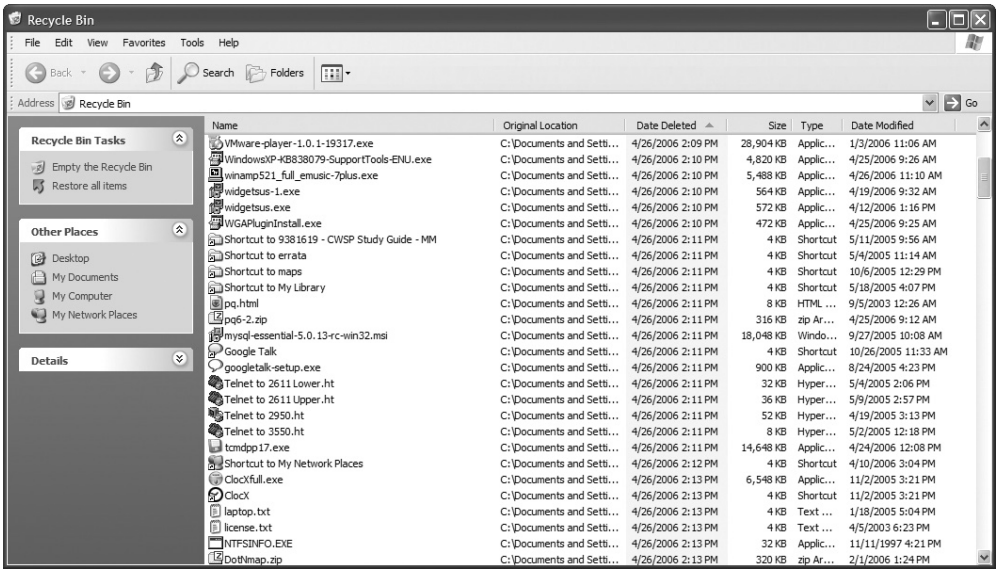


Figure 12-64 Mike's Recycle Bin

- **Temporary Internet files** When you go to a Web site, Windows keeps copies of the graphics and other items so the page will load more quickly the next time you access it. You can see these files by opening the Internet Options applet on the Control Panel. Figure 12-65 shows my temporary Internet files.
- **Downloaded program files** Your system always keeps a copy of any Java or ActiveX applets it downloads. You can see these in the Internet Options applet. You'll generally find only a few tiny files here.
- **Temporary files** Many applications create temporary files that are supposed to be deleted when the application is closed. For one reason or another, these temporary files sometimes aren't deleted. The location of these files varies with the version of Windows, but they always reside in a folder called TEMP.

Every hard drive eventually becomes filled with lots of unnecessary trash. All versions of Windows tend to act erratically when the drives run out of unused space. Fortunately, all versions of Windows have a powerful tool called *Disk Cleanup* (Figure 12-66).

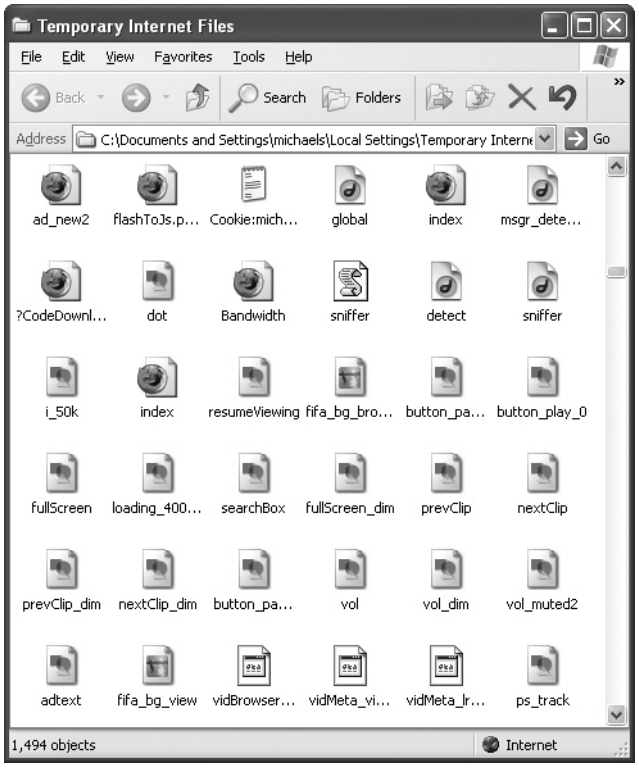


Figure 12-65 Lots of temporary Internet files

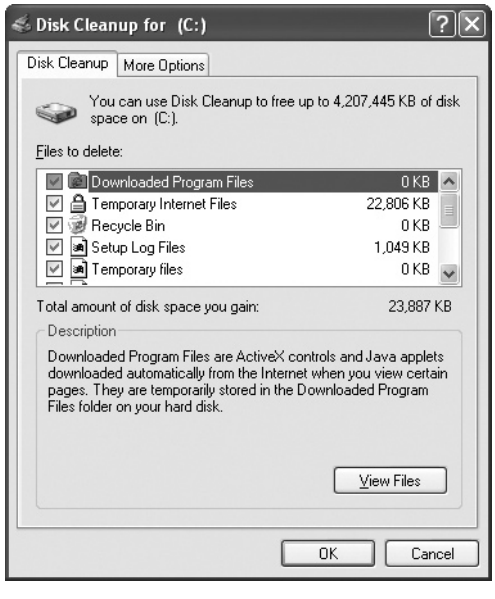


Figure 12-66 Disk Cleanup

You can access Disk Cleanup in all versions of Windows by choosing Start | All Programs | Accessories | System Tools | Disk Cleanup.

Disk Cleanup gets rid of the four types of files just described (and a few others). Run Disk Cleanup once a month or so to keep plenty of space available on your hard drive.

Troubleshooting Hard Drive Implementation

There's no scarier computer problem than an error that points to trouble with a hard drive. This section looks at some of the more common problems that occur with hard drives and how to fix them. These issues fall into three broad categories: installation, data corruption, and dying hard drives.

Installation Errors

Installing a drive and getting to the point where it can hold data requires four distinct steps: connectivity, CMOS, partitioning, and formatting. If you make a mistake at any point on any of these steps, the drive won't work. The beauty of this is that if you make an error, you can walk back through each step and check for problems. The troubleshooting section in Chapter 11 covered physical connections and CMOS, so this section concentrates on the latter two issues.

Partitioning Partitioning errors generally fall into two groups: failing to partition at all and making the wrong size or type of partition. You'll recognize the former type of error the first time you open My Computer/Computer after installing a drive. If you forgot to partition it, the drive won't even show up in My Computer, only in Disk Management. If you made the partition too small, that'll become painfully obvious when you start filling it up with files.

The fix for partitioning errors is simply to open Disk Management and do the partitioning correctly. If you've added files to the wrongly sized drive, don't forget to back them up before you repartition.

Formatting Failing to format a drive makes the drive unable to hold data. Accessing the drive in Windows results in a drive "is not accessible" error, and from a C:\ prompt, you'll get the famous "Invalid media" type error. Format the drive unless you're certain that the drive has a format already. Corrupted files can create the invalid media error. Check one of the sections on corrupted data later in this chapter for the fix.

Most of the time, formatting is a slow, boring process. But sometimes the drive makes "bad sounds" and you start seeing errors like the one shown in Figure 12-67 at the bottom of the screen.

An *allocation unit* is FORMAT's term for a cluster. The drive has run across a bad cluster and is trying to fix it. For years, I've told techs that seeing this error a few (610) times doesn't mean anything; every drive comes with a few bad spots. This is no longer true. Modern drives actually hide a significant number of extra sectors that they use to replace bad sectors automatically. If a new drive gets a lot of "Trying to recover lost allocation unit" errors, you can bet that the drive is dying and needs to be replaced. Get the hard drive maker's diagnostic to be sure. Bad clusters are reported by S.M.A.R.T.

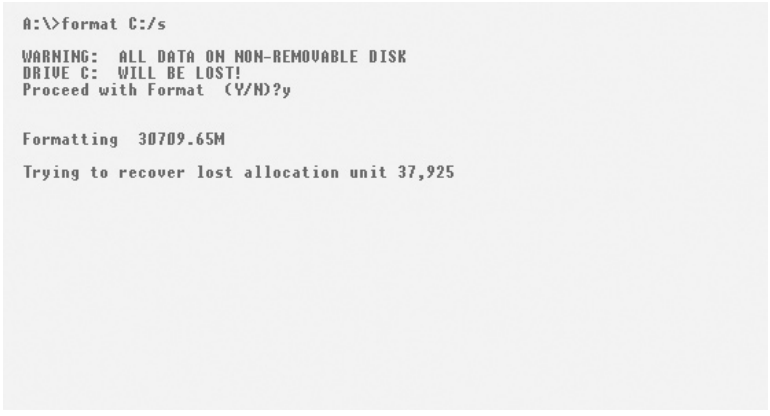


Figure 12-67 The "Trying to recover lost allocation unit" error

Mental Reinstallation Focus on the fact that all of these errors share a common thread—you just installed a drive! Installation errors don't show up on a system that has been running correctly for three weeks; they show up the moment you try to do something with the drive you just installed. If a newly installed drive fails to work, do a "mental reinstallation." Does the drive show up in the CMOS autodetect? No? Then recheck the cables, master/slave settings, and power. If it does show up, did you remember to partition and format the drive? Did it need to be set to active? These are commonsense questions that come to mind as you march through your mental reinstallation. Even if you've installed thousands of drives over the years, you'll be amazed at how often you do things such as forget to plug in power to a drive, forget CMOS, or install a cable backward. Do the mental reinstallation—it really works!

Data Corruption

All hard drives occasionally get corrupted data in individual sectors. Power surges, accidental shutdowns, corrupted installation media, and viruses, along with hundreds of other problems, can cause this corruption. In most cases, this type of error shows up while Windows is running. Figure 12-68 shows a classic example.

Figure 12-68 A corrupted data error

You may also see Windows error messages saying one of the following:

- "The following file is missing or corrupt"
- "The download location information is damaged"
- "Unable to load file"

If core boot files become corrupted, you may see text errors at boot, such as the following:

- "Cannot find COMMAND.COM"
- "Error loading operating system"
- "Invalid BOOT.INI"
- "NTLDR is missing or corrupt"

On older programs, you may see a command prompt open with errors such as this one:

```
Sector not found reading drive C: Abort, Retry, Fail?
```

The first fix for any of these problems is to run the Error-checking utility. Error-checking will go through and mark bad clusters and hopefully move your data to a good cluster.

Windows 2000/XP Extract/Expand If Error-checking fails to move a critically important file—such as a file Windows needs so it can load—on pre-Vista systems you can always resort to the command line and try to extract the file from the Windows cabinet files. Most Windows programs store all files in a compressed format called CAB (which is short for cabinet file). One CAB file contains many files, and most installation discs have lots of CAB files (see Figure 12-69).

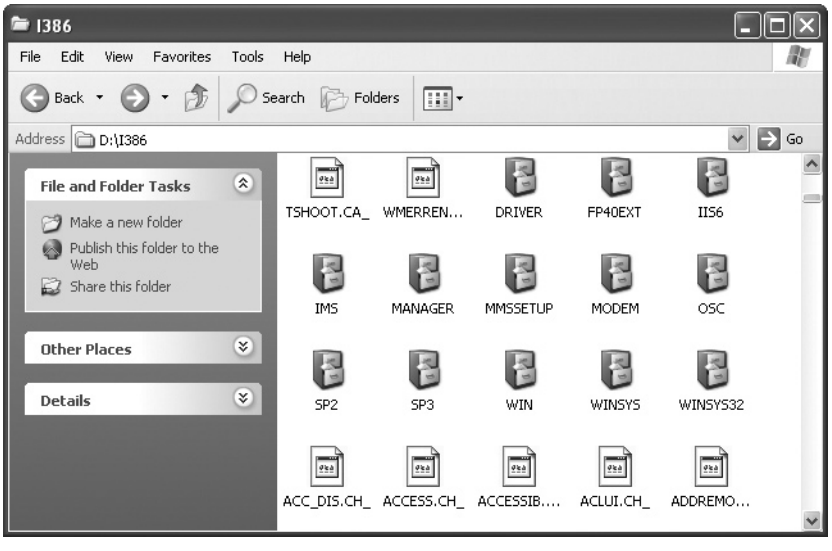


Figure 12-69 CAB files

To replace a single corrupt file this way, you need to know two things: the location of the CAB file that contains the file you need, and how to get the file out so you can copy it back to its original spot. Microsoft supplies the *EXPAND* program to enable you to

get a new copy of the missing file from the CAB files on the installation CD-ROM. Also notice how they are numbered—that's the secret to understanding these programs.

In most cases, all of the CAB files for a program are piled into some folder, as shown in Figure 12-71. Let's say you need a file called OLEPRO32.DLL. (I have no idea what this file does. I only know that Windows can't find it and you need to put it back.) Get to a command prompt within Windows and tell EXPAND to check *all* of the CAB files on your installation CD (drive E: in this example) with this command:

```
EXPAND e:\I386\*.CAB -F:OLEPRO32.DLL
```

EXPAND goes through all of the CAB files and finds the file. If you want to see details on the EXPAND command, use Windows Help or type **EXPAND /?** at a command prompt.

NOTE Chapter 15, "Working with the Command-Line Interface," goes into a lot of detail on using the command line.

Corrupted Data on Bad Sectors If the same errors continue to appear after you run the disk-checking utility, there's a chance that the drive has bad sectors.

Almost all drives today take advantage of built-in *error correction code* (*ECC*) that constantly checks the drive for bad sectors. If the ECC detects a bad sector, it marks the sector as bad in the drive's internal error map. Don't confuse this error map with a FAT. The partitioning program creates the FAT. The drive's internal error map was created at the factory on reserved drive heads and is invisible to the system. If the ECC finds a bad sector, you will get a corrupted data error as the computer attempts to read the bad sector. Disk-checking utilities fix this problem most of the time.

Many times, the ECC thinks a bad sector is good, however, and fails to update the internal error map. In this case, you need a program that goes back into the drive and marks the sectors as bad. That's where the powerful SpinRite utility from Gibson Research comes into play. SpinRite marks sectors as bad or good more accurately than ECC and does not disturb the data, enabling you to run SpinRite without fear of losing anything. And if it finds a bad sector with data in it, SpinRite has powerful algorithms that usually recover the data on all but the most badly damaged sectors (see Figure 12-70).

Without SpinRite, you must use a low-level format program supplied by the hard drive maker, assuming you can get one (not all are willing to distribute these). These programs work like SpinRite in that they aggressively check the hard drive's sectors and update the internal error map. Unfortunately, all of them wipe out all data on the drive. At least you can use the drive, even if it means repartitioning, formatting, and reinstalling everything.

Dying Hard Drive

Physical problems are rare but devastating when they happen. If a hard drive is truly damaged physically, there is nothing that you or any service technician can do to fix it. Fortunately, hard drives are designed to take a phenomenal amount of punishment without failing. Physical problems manifest themselves in two ways: either the drive works properly but makes a lot of noise, or the drive seems to disappear.

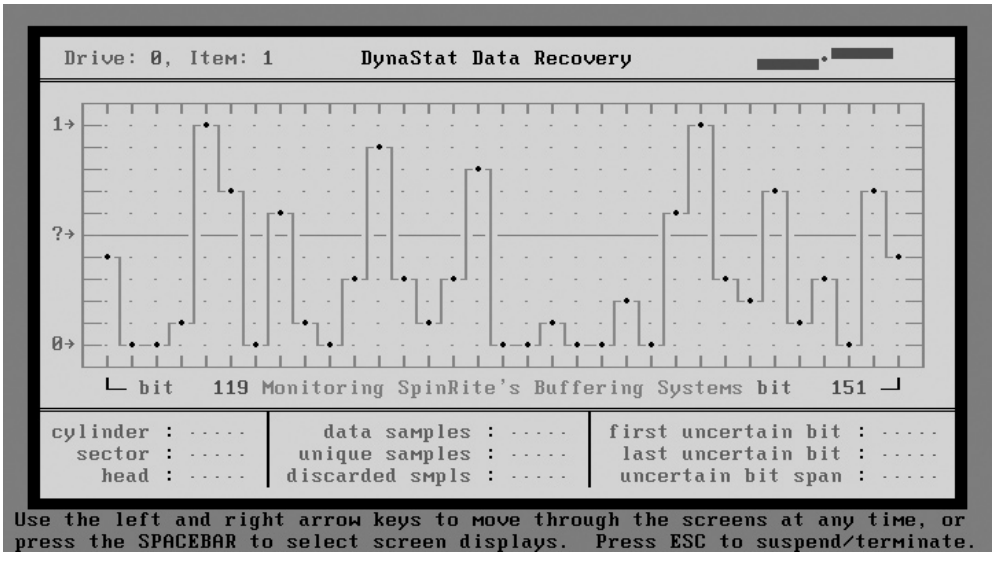


Figure 12-70 SpinRite at work

All hard drives make noise—the hum as the platters spin and the occasional slight scratching noise as the read/write heads access sectors are normal. However, if your drive begins to make any of the following sounds, it is about to die:

- Continuous high-pitched squeal
- Series of clacks, a short pause, and then another series of clacks
- Continuous grinding or rumbling

Back up your critical data and replace the drive. Windows comes with great tools for backing up data.

NOTE Most hard drives have three-year warranties. Before you throw away a dead drive, check the hard drive maker's Web site or call them to see if the drive is still under warranty. Ask for a return material authorization (RMA). You'll be amazed how many times you get a newer, usually larger, hard drive for free. It never hurts to check!

You'll know when a drive simply disappears. If it's the drive that contains your operating system, the system will lock up. When you try to restart the computer, you'll see this error message or something similar to it:

```
No Boot Device Present
```

If it's a second drive, it will simply stop showing up in My Computer/Computer. The first thing to do in this case is to fire up the System Setup program and see if autodetect

sees the drive. If it does, you do not have a physical problem with the drive. If autodetect fails, shut off the system and remove the ribbon cable, but leave the power cable attached. Restart the system and listen to the drive. If the drive spins up, you know it is getting good power. This is usually a clue that the drive is probably good. In that case, you need to look for more mundane problems such as an unplugged power cord or jumpers incorrectly set. If the drive doesn't spin up, try another power connector. If it still doesn't spin up and you've triple-checked the jumpers and ribbon cable, you have a problem with the onboard electronics, and the drive is dead.

NOTE If you ever lose a hard drive that contains absolutely critical information, you can turn to a company that specializes in hard drive data recovery. The job will be expensive—prices usually start around $1000 (U.S.)—but when you have to have the data, such companies are your only hope. Do a Web search for "data recovery" or check the Yellow Pages for companies in this line of business.

Beyond A+

Modern hard drives have many other features that are worth knowing about but that rarely impact beginning techs. A couple of the more interesting ones are spindle speed and third-party hard drive tools. If you have a burning desire to dive into hard drives in all their glory, you need not go any farther than the Storage Review, an excellent site dedicated solely to hard drives. Here's the link: www.storagereview.com.

Third-Party Partition Tools

Disk Management is a good tool, but it's limited for some situations. Some really great third-party tools on the market can give you incredible flexibility and power to structure and restructure your hard drive storage to meet your changing needs. They each have interesting unique features, but in general they enable you to create, change, and delete partitions on a hard drive *without* destroying any of the programs or data stored there. Slick! These programs aren't on the CompTIA A+ exams, but all PC techs use at least one of them, so let's explore three of the most well-known examples: Symantec's Norton PartitionMagic, Avanquest Partition Commander Professional, and the open source Linux tool, GParted.

Probably the most well-known third-party partition tool is PartitionMagic, although it's quite dated at this point. It supports older versions of Windows but has problems with Windows Vista/7. With it, you can create, resize, split, merge, delete, undelete, and convert partitions without destroying your data. Among the additional features it advertises are the capability to browse, copy, or move files and folders between supported partitions; to expand an NTFS partition—even if it's a system partition—without rebooting; to change NTFS cluster sizes; and to add new partitions for multiple OSs by using a simple wizard.

Avanquest offers a variety of related products, one of which is the very useful Partition Commander Professional. Unlike PartitionMagic, it supports all versions of Windows and enables you to play with your partitions without destroying your data. Among its niftier features are the capability to convert a dynamic disk to a basic disk nondestructively (which you can't do with the Microsoft-supplied Windows tools); to defrag the master file table on an NTFS partition; and to move unused space from one partition to another on the same physical drive, automatically resizing the partitions based on the amount of space you tell it to move. Figure 12-71 shows the Partition Commander dialog box for moving unused space between partitions.

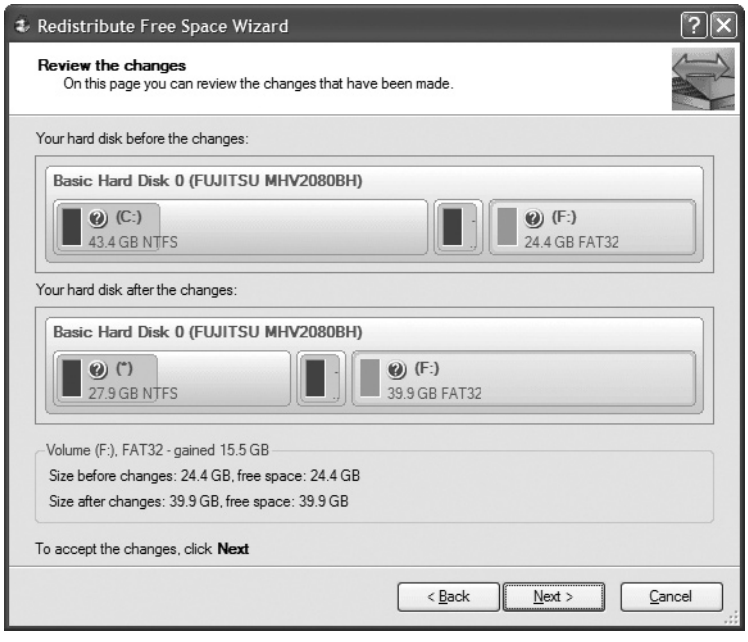


Figure 12-71 Partition Commander

The only problem with PartitionMagic and Partition Commander is that they cost money. There's nothing wrong with spending money on a good product, but if you can find something that does the job for free, why not try it? If you think like I do, check out the Gnome Partition Editor, better known as GParted. You can find it at http://sourceforge.net/.

GParted is an incredibly powerful partition editor and does almost everything the for-pay partition editors do, but it's free. It's still in beta—which means it's constantly changing and it has a few bugs (that are constantly being fixed)—but I use it all of the time and love it. If you look closely at Figure 12-72, you'll notice that it uses strange names for the partitions, such as HDA1 or HDB3. These are Linux conventions and are well documented in GParted's Help screens. Take a little time and you'll love GParted too.

The one downside to GParted is that it is a Linux program—because no Windows version exists, you need Linux to run it. So how do you run Linux on a Windows system

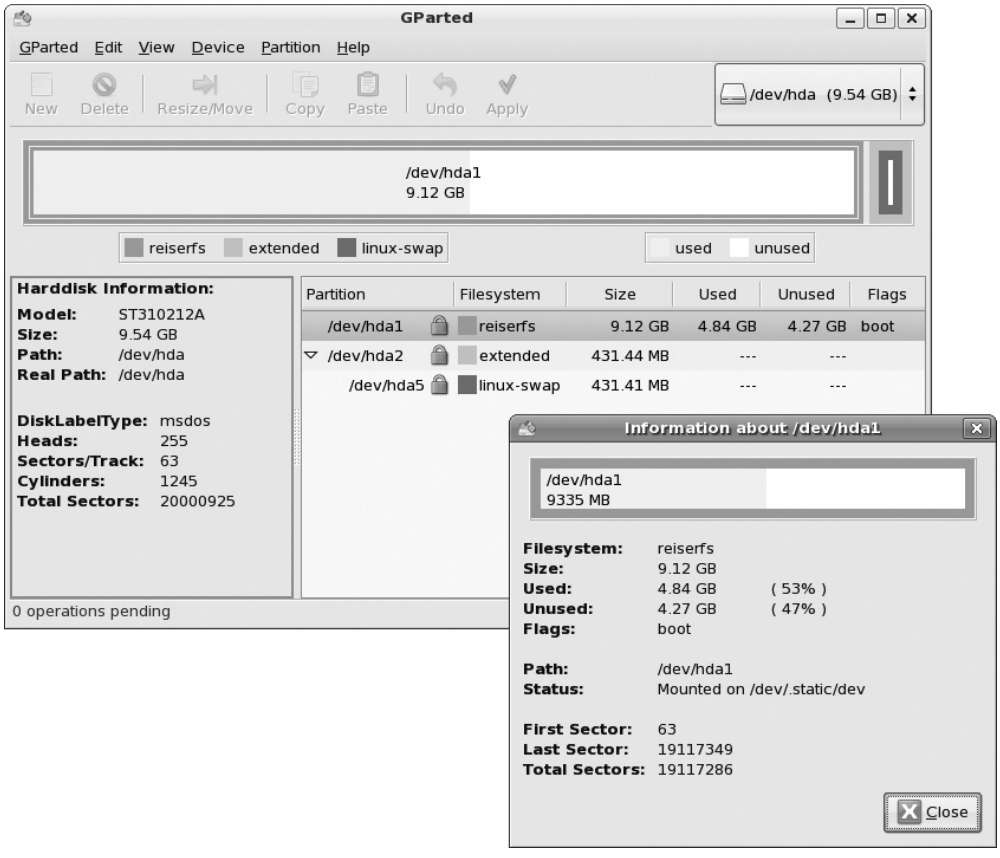


Figure 12-72 GParted in action

without actually installing Linux on your hard drive? The answer is easy—the folks at GParted will give you the tools to burn a live CD that boots Linux so you can run GParted!

A live CD is a complete OS on a CD. Understand this is not an installation CD like your Windows installation disc. The OS is already installed on the CD. You boot from the live CD and the OS loads into RAM, just like the OS on your hard drive loads into RAM at boot. As the live CD boots, it recognizes your hardware and loads the proper drivers into RAM so everything works. You get everything you'd expect from an OS with one big exception: a live CD does not touch your hard drive. Of course you may run programs (such as GParted) that work on your hard drive, which makes live CDs popular with PC techs, because you can toss them into a cranky system and run utilities.

The truly intrepid might want to consider using The Ultimate Boot CD (UBCD), basically a huge pile of useful freeware utilities compiled by frustrated technician

Ben Burrows, who couldn't find a boot disk when he needed one. His Web site is www.ultimatebootcd.com. The UBCD has more than 100 different tools, all placed on a single live CD. It has all of the low-level diagnostic tools for all of the hard drive makers, four or five different partitioning tools, S.M.A.R.T. viewers, hard drive wiping utilities, and hard drive cloning tools (nice for when you want to replace a hard drive with a larger one). Little documentation is provided, however, and many of the tools require experience way beyond the scope of the CompTIA A+ exams. I will tell you that I have a copy and I use it.

Chapter Review Questions

1. Which is the most complete list of file systems Windows 2000/XP and Windows Vista/7 can use?
   - **A.** FAT16, FAT32, NTFS
   - **B.** FAT16, FAT32, FAT64, NTFS
   - **C.** FAT16, FAT32
   - **D.** FAT16, NTFS
2. Which of the following correctly identifies the four possible entries in a file allocation table?
   - **A.** Filename, date, time, size
   - **B.** Number of the starting cluster, number of the ending cluster, number of used clusters, number of available clusters
   - **C.** An end-of-file marker, a bad-sector marker, code indicating the cluster is available, the number of the cluster where the next part of the file is stored
   - **D.** Filename, folder location, starting cluster number, ending cluster number
3. What program does Microsoft include with Windows 2000/XP and Windows Vista/7 to partition and format a drive?
   - **A.** Format
   - **B.** Disk Management console
   - **C.** Disk Administrator console
   - **D.** System Commander
4. What does NTFS use to provide security for individual files and folders?
   - **A.** Dynamic disks
   - **B.** ECC
   - **C.** Access Control List
   - **D.** MFT

5. Adam wants to create a new simple volume in some unallocated space on his hard drive, but when he right-clicks the space in Disk Management he sees only an option to create a new partition. What is the problem?
   - **A.** The drive has developed bad sectors.
   - **B.** The drive is a basic disk and not a dynamic disk.
   - **C.** The drive has less than 32 GB of unallocated space.
   - **D.** The drive is jumpered as a slave.
6. Jaime wishes to check her hard drive for errors. What tool should she use?
   - **A.** FDISK
   - **B.** Format
   - **C.** Disk Management
   - **D.** Error-checking
7. To make your files unreadable by others, what should you use?
   - **A.** Clustering
   - **B.** Compression
   - **C.** Disk quotas
   - **D.** Encryption
8. How can you effectively expand the capacity of an NTFS?
   - **A.** Create an extended partition to extend the capacity.
   - **B.** Install a second drive and mount it to a folder on the original smaller NTFS drive.
   - **C.** Convert the drive to a dynamic disk and create a mirrored set.
   - **D.** Format the drive with the Quick Format option.
9. Which volume configuration uses parity for fault tolerance?
   - **A.** RAID 5
   - **B.** Mirrored set
   - **C.** Spanned volume
   - **D.** Striped volume
10. You will almost certainly destroy your hard drive if you reverse which of the following?
    - **A.** Power cable
    - **B.** Data cable
    - **C.** Jumpers
    - **D.** Pins

Answers

1. **A.** Windows 2000/XP and Windows Vista/7 can use FAT16, FAT32, and NTFS.
2. **C.** The four possible entries in a file allocation table are an end-of-file marker, a bad-sector marker, code indicating the cluster is available, and the number of the cluster where the next part of the file is stored.
3. **B.** Windows 2000/XP and Windows Vista/7 use the Disk Management console to partition and format a drive.
4. **C.** Because NTFS views individual files and folders as objects, it can provide security for those objects through an Access Control List.
5. **B.** The drive is a basic disk and not a dynamic disk. Partitions are created on basic disks, while volumes are created on dynamic disks.
6. **D.** Error-checking is used to check a drive for errors.
7. **D.** To make your files unreadable by others, use encryption.
8. **B.** You can effectively expand the capacity of an NTFS drive by installing a second drive and mounting it to a folder on the original smaller NTFS drive.
9. **A.** RAID 5 uses parity for fault tolerance.
10. **A.** You will almost certainly destroy your hard drive if you reverse the power cable.