

# U N D E R S T A N D I N G H A R D D I S K S

## Basic Concepts

*This document outlines some basic concepts of hard disk partitioning. Some sections include tips for using PowerQuest products.*

- What Is a Hard Disk?
- What Is Disk Formatting?
- File Systems
- Understanding Partitions
- Partitioned and formatted hard disk
- Managing Partitions
- Freeing Disk Space Before Enlarging a FAT Partition
- Hiding and Unhiding Partitions
- Understanding Drive Letters
- Understanding the BIOS 1,024 Cylinder Limit
- Understanding the 2 GB Boot Code Boundary
- Changing the BIOS LBA Mode Setting
- Restoring System Files

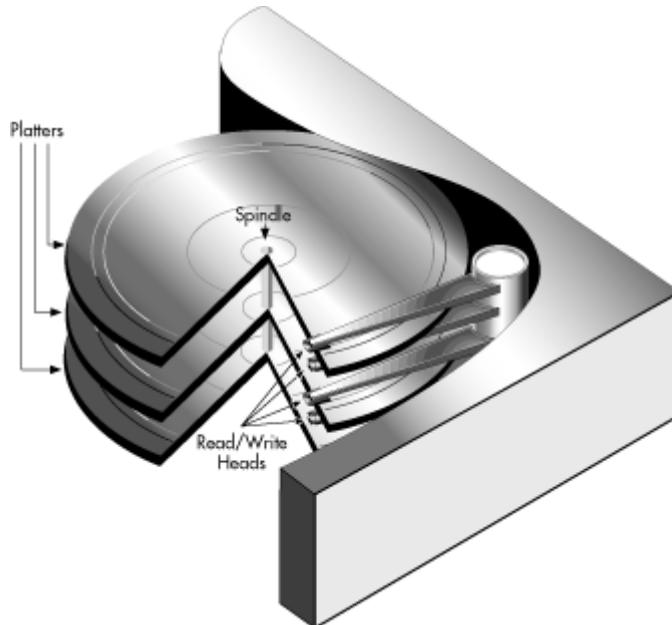
## What Is a Hard Disk?

A **hard disk** or drive is the part of your computer responsible for long-term storage of information. Unlike volatile memory (often referred to as RAM) which loses its stored information once its power supply is shut off, a hard disk stores information permanently,

allowing you to save programs, files, and other data. Hard disks also have much greater storage capacities than RAM; in fact, current hard disks may contain over 19 GB of storage space.

### ***Basic components of a Hard Disk***

A hard disk is comprised of four basic parts: platters, a spindle, read/write heads, and integrated electronics.



**Figure 1:** Basic components of a hard disk

- **Platters** are rigid disks made of metal or plastic. Both sides of each platter are covered with a thin layer of iron oxide or other magnetizable material.
- The platters are mounted on a central axle or **spindle**, which rotates all the platters at the same speed.
- **Read/write heads** are mounted on arms that extend over both top and bottom surfaces of each disk. There is at least one read/write head for each side of each platter. The arms jointly move back and forth between the platters' centers and outside edges; this movement, along with the platters' rotation, allow the read/write heads to access all areas of the platters.

- The **integrated electronics** translate commands from the computer and move the read/write heads to specific areas of the platters, thus reading and/or writing the needed data.

### ***How Is Data Stored and Retrieved?***

Computers record data on hard disks as a series of binary bits. Each bit is stored as a magnetic charge (positive or negative) on the oxide coating of a disk platter.

When a computer saves data, it sends the data to the hard disk as a series of bits. As the disk receives the bits, it uses the read/write heads to magnetically record or “write” the bits on the platters. Data bits are not necessarily stored in succession; for example, the data in one file may be written to several different areas on different platters.

When the computer requests data stored on the disk, the platters rotate and the read/write heads move back and forth to the specified data areas. The read/write heads read the data by determining the magnetic field of each bit, positive or negative, and then relay that information back to the computer.

The read/write heads can access any area of the platters at any time, allowing data to be accessed randomly (rather than sequentially, as with a magnetic tape). Because hard disks are capable of random access, they can typically access any data within a few thousandths of a second.

### **What Is Disk Formatting?**

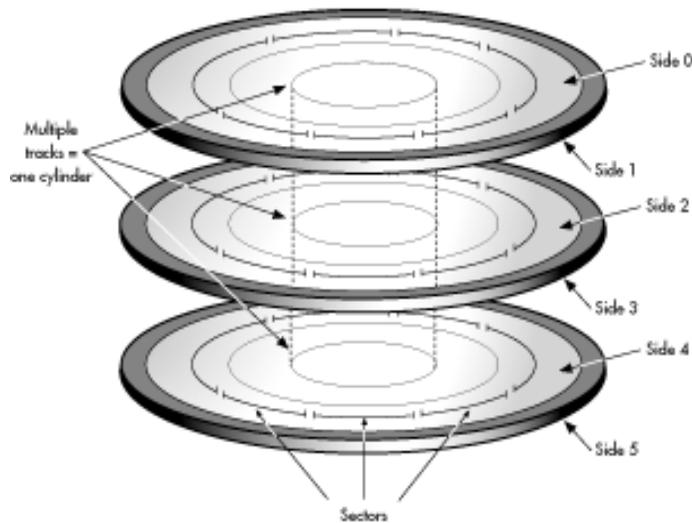
Computers must be able to access needed information on command; however, even the smallest hard disk can store millions and millions of bits. How does the computer know where to look for the information it needs? To solve this problem, hard disks are organized into discrete, identifiable divisions, thus allowing the computer to easily find any particular sequence of bits.

The most basic form of disk organization is called formatting. Formatting prepares the hard disk so that files can be written to the platters and then quickly retrieved when needed. Hard disks must be formatted in two ways: physically and logically.

#### ***Physical Formatting***

A hard disk must be physically formatted before it can be logically formatted. A hard disk’s physical formatting (also called low-level formatting) is usually performed by the manufacturer.

Physical formatting divides the hard disk's platters into their basic physical elements: tracks, sectors, and cylinders. See Figure 2. These elements define the way in which data is physically recorded on and read from the disk.



**Figure 2:** Physical format of a typical hard disk

- **Tracks** are concentric circular paths written on each side of a platter, like those on a record or compact disc. The tracks are identified by number, starting with track zero at the outer edge.
- Tracks are divided into smaller areas or **sectors**, which are used to store a fixed amount of data. Sectors are usually formatted to contain 512 bytes of data (there are 8 bits in a byte).
- A **cylinder** is comprised of a set of tracks that lie at the same distance from the spindle on all sides of all the platters. For example, track three on every side of every platter is located at the same distance from the spindle. If you imagine these tracks vertically connected, the set forms the shape of a cylinder.

Computer hardware and software frequently work using cylinders. When data is written to a disk in cylinders, it can be fully accessed without having to move the read/write heads. Because head movement is slow compared to disk rotation and switching between heads, cylinders greatly reduce data access time.

After a hard disk is physically formatted, the magnetic properties of the platter coating may gradually deteriorate. Consequently, it becomes more and more difficult for the read/write heads to read data from or write data to the affected platter sectors. The sectors that can no longer be used to hold data are called **bad sectors**. Fortunately, the quality of

modern disks is such that bad sectors are rare. Furthermore, most modern computers can determine when a sector is bad; if this happens, the computer simply marks the sector as bad (so it will never be used) and then uses an alternate sector.

## **Logical Formatting**

After a hard disk has been physically formatted, it must also be logically formatted. Logical formatting places a file system on the disk, allowing an operating system (such as DOS, OS/2, Windows, or Linux) to use the available disk space to store and retrieve files. Different OSs (operating systems) use different file systems, so the type of logical formatting you apply depends on the OS you plan to install.

For more in-depth information on file systems, see “File Systems” on page 5.

Formatting your entire hard disk with one file system necessarily limits the number and types of OSs you can install on the disk. Luckily, there is a solution to this problem. Before a disk is logically formatted, it can be divided into partitions. Each partition can then be formatted with a different file system, allowing you to install multiple OSs. Dividing your hard disk into partitions also allows you to use disk space more efficiently.

For more information on partitions, see “Understanding Partitions” on page 9.

## **File Systems**

All file systems consist of structures necessary for storing and managing data. These structures typically include an operating system boot record, directories, and files. A file system also performs three main functions: 1) tracking allocated and free space, 2) maintaining directories and file names, and 3) tracking where each file is physically stored on the disk.

Different file systems are used by different operating systems. Some OSs can recognize only one file system, while other OSs can recognize several. Some of the most common file systems are the following:

- FAT (File Allocation Table)
- FAT32 (File Allocation Table 32)
- NTFS (New Technology File System)
- HPFS (High Performance File System)
- NetWare File System
- Linux Ext2 and Linux Swap

## **FAT**

The FAT file system is used by DOS, Windows 3.x, and Windows 95 (in most installations). The FAT file system is also accessible by Windows 98/Me/NT/2000, and by OS/2.

The FAT file system is characterized by the use of a file allocation table (FAT) and clusters. The FAT is the heart of the file system; for safety, the FAT is duplicated to protect its data from accidental deletion or corruption. Clusters are the FAT system's smallest unit of data storage; one cluster consists of a fixed number of disk sectors. The FAT records which clusters are used, which are unused, and where files are located within the clusters.

The FAT file system supports disk or partition sizes up to 2 GB, but only allows a maximum of 65,525 clusters. Therefore, whatever the size of the hard disk or partition, the number of sectors in one cluster must be large enough so that all available space can be included within 65,525 clusters. The larger the available space, the larger the cluster size must be.

**TIP:** In general, large clusters tend to waste more space than small clusters. For more information on managing cluster size, see "Making Efficient Use of Disk Space" (page 14).

The FAT file system also uses a root directory. This directory has a maximum allowable number of entries and must be located at a specific place on the disk or partition. OSs that use the FAT file system represent the root directory with the backward slash character (\) and initially display this directory at boot-up.

The root directory stores information about each sub-directory and file in the form of individual directory entries. For example, a file's directory entry holds information such as the file name, the size of the file, a date and time stamp that indicates when the file was last changed, the starting cluster number (which cluster holds the first portion of the file), and the file's attributes (for example, hidden or system).

**TIP:** You can use PartitionMagic's Resize Root option to change the size (number of root entries) available in a FAT root directory, within predefined limits. For details, see the *PartitionMagic User Guide*, or refer to online help.

**TIP:** All PartitionMagic, Drive Image, BootMagic, and VolumeManager features can be used on a FAT disk or partition.

## **FAT32**

FAT32 is a file system that can be used by Windows 95 OEM Service Release 2 (version 4.00.950B), Windows 98, Windows Me, and Windows 2000. However, DOS, Windows 3.x, Windows NT 3.51/4.0, earlier versions of Windows 95, and OS/2 do not recognize FAT32 and cannot boot from or use files on a FAT32 disk or partition.

FAT32 is an enhancement of the FAT file system and is based on 32-bit file allocation table entries, rather than the 16-bit entries used by the FAT system. As a result, FAT32 supports much larger disk or partition sizes (up to 2 terabytes).

The FAT32 file system uses smaller clusters than the FAT file system, has duplicate boot records, and features a root directory that can be any size and can be located anywhere on the disk or partition.

On a FAT32 disk or partition, you can use all PartitionMagic, Drive Image, BootMagic, and VolumeManager features except Resize Root, which is unnecessary for FAT32.

## **NTFS**

The NTFS (New Technology File System) is accessible only by Windows NT/2000. NTFS is not recommended for use on disks less than 400 MB because it uses a great deal of space for system structures.

The central system structure of the NTFS file system is the MFT (Master File Table). NTFS keeps multiple copies of the critical portion of the master file table to protect against corruption and data loss.

Like FAT and FAT32, NTFS uses clusters to store data files; however, the size of the clusters is not dependent on the size of the disk or partition. A cluster size as small as 512 bytes can be specified, regardless of whether a partition is 500 MB or 5 GB. Using small clusters not only reduces the amount of wasted disk space, but also reduces file fragmentation, a condition where files are broken up over many noncontiguous clusters, resulting in slower file access. Because of its ability to use small clusters, NTFS provides good performance on large drives.

Finally, the NTFS file system supports hot fixing, a process through which bad sectors are automatically detected and marked so that they will not be used.

**TIP:** All PartitionMagic, Drive Image, and VolumeManager features can be used with NTFS, except the following features that are specific to the FAT and FAT32 file systems:

- Resize Root

- Change Cluster Size
- All conversions
- Bad Sector Retest

**TIP:** VolumeManager can only access partitions under Windows 2000 that are found on basic disks. Currently, dynamic Windows 2000 disks cannot be manipulated using VolumeManager. For more information about basic and dynamic disks, see the Windows 2000 documentation.

## **HPFS**

The HPFS (High Performance File System) is the preferred file system for OS/2 and is also supported by older versions of Windows NT.

Unlike the FAT file systems, HPFS sorts its directory based on file names. HPFS also uses a more efficient structure to organize the directory. As a result, file access is often faster and space used more efficiently than with the FAT file system.

HPFS allocates file data in sectors instead of clusters. To keep track of which sectors have or have not been used, HPFS organizes a disk or partition into 8 MB bands, with 2 KB allocation bitmaps between the bands. This banding improves performance because the read/write heads don't have to return to track zero each time the OS needs to access information about available space or a needed file's location.

All PartitionMagic, Drive Image, BootMagic, and VolumeManager features can be used with HPFS, except those features that are specific to the FAT and FAT32 file systems.

## **NetWare File System**

The Novell NetWare operating system uses the NetWare File System, which was developed specifically for use by NetWare servers.

PartitionMagic identifies NetWare 3.x or 4.x partitions, allowing you to view these partitions' information. All other PartitionMagic features, however, are unavailable for use with the NetWare File System. VolumeManager and Server Image are specifically designed to support Novell NetWare partitions.

**TIP:** If you need to move, resize, copy, or otherwise manage NetWare partitions, you can use PowerQuest's VolumeManager to perform these tasks.

## **Linux Ext2 and Linux Swap**

The Linux Ext2 and Linux Swap file systems were developed for the Linux OS (a freeware version of UNIX). The Linux Ext2 file system supports a maximum disk or partition size of 4 terabytes.

**TIP:** All PartitionMagic, Drive Image, BootMagic, and VolumeManager features can be used with Linux Ext2 and Linux Swap with the exception of the following:

- Linux Swap does not have volume IDs so the Label operation will not function.
- Hide/Unhide will not function on either Linux Ext2 or Linux Swap.

## **Understanding Partitions**

After a disk has been physically formatted, it can be divided into separate physical sections or partitions. Each partition functions as an individual unit, and can be logically formatted with any desired file system. Once a disk partition has been logically formatted, it is referred to as a volume.

As part of the formatting operation, you are asked to give the partition a name, called the “volume label.” This name helps you easily identify the volume.

### **Why Use Multiple Partitions?**

Many hard disks are formatted as one large partition. This setup, however, doesn’t always provide the best possible use of your disk space or resources. The alternative is to separate your hard disk into partitions. Using multiple partitions, you can:

- Install more than one OS on your hard disk;
- Make the most efficient use of your available disk space;
- Make your files as secure as possible;
- Physically separate data so that it is easy to find files and back up data.

The following sections discuss partitions in greater detail, helping you create and use partitions to get the most out of your hard disk.

### **Partition Types**

There are three kinds of partitions: primary, extended, and logical. Primary and extended partitions are the main disk divisions; one hard disk may contain up to four primary partitions, or three primary partitions and one extended partition. The extended partition can then be further divided into any number of logical partitions.

## ***Primary Partitions***

A primary partition may contain an operating system along with any number of data files (for example, program files or user files). Before an OS is installed, the primary partition must be logically formatted with a file system compatible to the OS.

If you have multiple primary partitions on your hard disk, only one primary partition may be visible and active at a time. The active partition is the partition from which an OS is booted at computer startup. Primary partitions other than the active partition are hidden, preventing their data from being accessed. Thus, the data in a primary partition can be accessed (for all practical purposes) only by the OS installed on that partition.

If you plan to install more than one operating system on your hard disk, you probably need to create multiple primary partitions; most operating systems can be booted only from a primary partition.

## ***Extended Partitions***

The extended partition was invented as a way of getting around the arbitrary four-partition limit. An extended partition is essentially a container in which you can further physically divide your disk space by creating an unlimited number of logical partitions.

An extended partition does not directly hold data. You must create logical partitions within the extended partition in order to store data. Once created, logical partitions must be logically formatted, but each can use a different file system.

## ***Logical Partitions***

Logical partitions can exist only within an extended partition and are meant to contain only data files and OSs that can be booted from a logical partition (OS/2, Linux, and Windows NT).

The illustration below shows a hard disk that contains four main partitions: three primary partitions and one extended partition. The extended partition has been further divided into two logical partitions.

Each primary partition has been formatted to use a different file system (FAT, NTFS, and HPFS). The two logical partitions have both been formatted to use the FAT file system.

Although Figure 3 shows all partitions on a single side of one platter, in actual use the partitions would probably be spread across the sides of several platters.

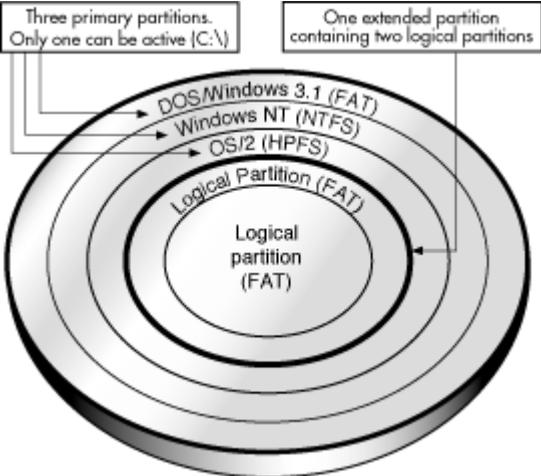


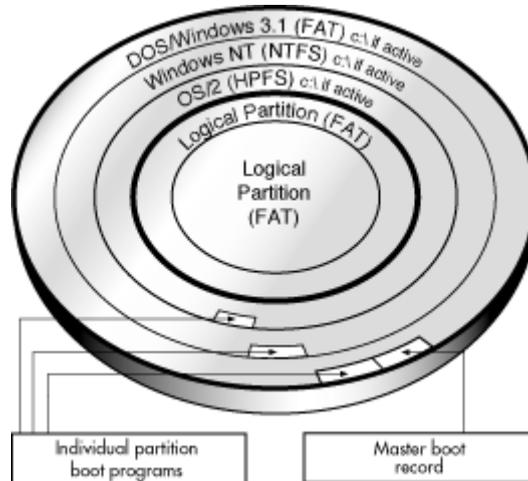
Figure 3: Partitioned and formatted hard disk

## Understanding How a Computer Boots

The way a computer boots from a hard disk depends on the way the hard disk is partitioned and on the operating system being booted.

## The Basic Boot Process

When you turn on the power to your computer, the CPU (central processing unit) takes control. The CPU immediately executes the instructions built into the computer's ROM BIOS, a program that contains the startup procedures. The last part of the BIOS instructions contains the boot routine. This routine is programmed to read the master boot record from the first sector of the first physical hard disk.



**Figure 4:** Hard disk with master boot record and three partition (OS-specific) boot records

The MBR (master boot record) contains a master boot program and a partition table that describes all of the hard disk's partitions. The BIOS boot routine executes the master boot program, which then continues the boot process. The master boot program looks at the partition table to see which primary partition is active. If there is only one primary partition, that partition's OS is loaded and booted into operation.

If the hard disk has more than one primary partition, each bootable partition (that is, containing an OS) has its own boot record stored in its first sector. This boot record holds a boot program designed specifically to start that partition's installed OS. This OS-specific boot record is usually written to the partition when the partition is logically formatted, but can also be added later with an OS-specific utility (for example, the DOS SYS utility).

After identifying the active partition, the master boot program starts that partition's boot program. In turn, the boot program loads the necessary OS files and starts the OS.

## ***Operating System–Specific Boot Information***

Most operating systems, including DOS, and Windows 3.x/95/98/Me/NT/2000 rely on the active primary partition when they boot from a hard disk. However, different operating systems rely on the active primary partition in different ways.

- DOS, Windows 3.x, and Windows 95/98 must boot from an active primary partition on the first hard disk drive.
- Windows NT/2000 can boot from a logical partition, but the Windows NT/2000 boot program must be in the active primary partition on the first hard disk.
- OS/2 can be booted from a logical partition; however, the extended partition containing the logical partition must be contained within the first 2 GB of the hard disk. Additionally, the Boot Manager utility provided with OS/2 must be present on the hard disk in order to install OS/2.

## **Managing Partitions**

The following sections introduce you to concepts and activities that help you use disk partitioning to your best advantage.

### ***Setting an Active Primary (Boot) Partition***

When you create multiple primary partitions to hold different operating systems, you must tell the computer which primary partition to boot from. The primary partition from which the computer boots is called the active partition. If there is not an active primary partition on the first physical hard disk, your computer will not be able to boot from your hard disk.

**WARNING!** Before you make a primary partition active, make sure that it is a bootable partition. Bootable partitions are logically formatted and have the necessary OS files installed. Partitions without an OS cannot be booted.

**TIP:** The **Set Active** option in PartitionMagic, VolumeManager and Drive Image lets you easily choose which primary partition you want to be the active partition. For more information, see the product User Guide, or refer to online help.

### ***Making Good Use of Logical Partitions***

There are three good reasons for creating an extended partition and dividing it into logical partitions:

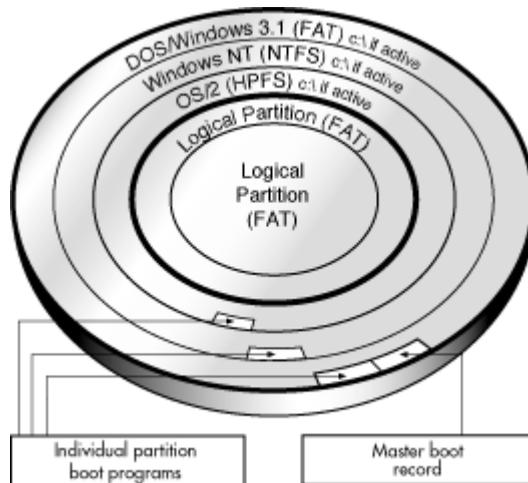
- You can access logical partition files from multiple OSs.

- Logical partitions help you make efficient use of disk space.
- Logical partitions physically separate groups of files for easier organization or increased security.

### **Accessing the Same Files From Multiple OSs**

Multiple logical partitions do not need to be hidden like primary partitions. You can have many logical partitions visible at the same time. Therefore, you can access data stored in a logical partition from multiple OSs installed in different primary or logical partitions, provided that the logical partition uses a file system that the OSs recognize.

For example, refer to the partitioned hard disk shown below. Because DOS/Windows, Windows NT, and OS/2 all recognize FAT partitions, any of those three primary partitions could be active and still be able to recognize and use the files saved within either of the logical partitions.



**Figure 5:** Both logical FAT partitions can be accessed by any of the three primary partition OSs.

### **Making Efficient Use of Disk Space**

If you have a large hard disk and want to use the FAT file system on all or most of the disk, you can prevent wasted space by using several small FAT partitions.

All data on a FAT partition are stored in allocation units called clusters. Each cluster is made up of a fixed number of disk sectors.

The FAT file system supports disk or partition sizes up to 2 GB, but only allows a maximum of 65,525 clusters. Therefore, whatever the size of the hard disk or partition, the number of sectors in one cluster must be large enough so that all available space can be included within 65,525 clusters. The larger the available space, the larger the cluster size must be.

However, using a large cluster size wastes disk space. Even if a data file (or the last portion of a data file) is much smaller than the cluster size, the computer must still use a complete cluster to store the data. The rest of the cluster space goes unused.

The following table shows the minimum cluster size and typical wasted space for various partition sizes.

<b>Partition Sizes</b>	<b>Minimum Required Cluster Size</b>	<b>% Wasted Space (approx.)</b>
16-127 MB	2 KB	2%
128-255 MB	4 KB	4%
256-511 MB	8 KB	10%
512-1023 MB	16 KB	25%
1,024-2,047 MB	32 KB	40%
2,048-4,096 MB	64 KB	50%

64K clusters are only available in Windows NT and Windows 2000. Other operating systems cannot use 64K clusters, no matter how large the partition.

You can prevent wasted disk space by using smaller partitions, because smaller partitions use smaller cluster sizes. For example, a 1,024,047 MB partition has a cluster size of 32 KB. If you saved a 2 KB file to this partition, an entire 32 KB cluster would be used to

save the file, wasting 30K of space. However, if you divide your storage space into 120 MB partitions, these partitions would only use 2K clusters. When you save the same 2K file, the file would fit neatly into a 2K cluster with no wasted space.

**TIP:** If you have large FAT partitions, either primary or logical, you can probably recover wasted storage space by resizing them with the PartitionMagic and VolumeManager **Resize Cluster** option. Any recovered space can then be divided into other small partitions. For more information, see the product user guide, or refer to online help.

### ***Simplifying File Access and Enhancing File Security***

If you have a large hard disk, placing all your files and subdirectories under one root directory quickly results in a large and complex directory structure. The larger and more complex the root directory, the harder it is for you to keep track of your files. Organization becomes difficult, forcing you to sort through numerous directories and subdirectories just to find the files you want.

Smart use of logical partitions can help you avoid this problem. Simply separate your files into groups, storing each group in an individual logical partition. When you need a particular group of files, you can easily switch to the corresponding logical partition. The complexity of your directory structure is minimized, allowing you to access desired files much more quickly.

You can also use additional partitions to enhance security for sensitive files. For example, if you wish to limit access to a particular group of files, you can store those files on a logical partition and then hide that partition from access.

**TIP:** For more information on hiding partitions, see the product User Guide, or refer to online help.

If you use multiple OSs, you could also format a logical data partition with the file system of the OS that provides the best security features. The OS could then be used to limit access to the data partition.

Logical partitions can also be used to store additional copies of critical files. For example, if you put copies of your critical files on a FAT logical partition, this partition could be accessed by any of your OSs that recognize FAT. If one OS crashed or became corrupted, you could boot another OS and still have access to the critical files.

## Freeing Disk Space Before Enlarging a FAT Partition

**TIP:** If you want to make a FAT partition larger using Drive Image, PartitionMagic, or VolumeManager's **Resize** option, you must first understand that the resulting partition may require a larger cluster size. In general, larger cluster sizes cause a greater percentage of wasted space in the partition.

When resizing a partition, space must be allocated for any increase in the required cluster size. Therefore, there must be unused space available inside the partition. If the partition is almost full, there may not be enough space to allow you to resize the partition. If this is the case, you can create unused space by either deleting files from the partition or by moving files to a different partition.

The following table approximates the amount of unused space necessary to resize a partition within the listed size ranges. The actual amount of unused space needed can vary, depending on the number and sizes of the files in the partition.

<b>Partition Size (within this range)</b>	<b>Required Cluster Size</b>	<b>% Wasted Space</b>	<b>Unused Space Needed for Resize</b>
128-255 MB	4 KB	4%	5.1 MB
256-511 MB	8 KB	10%	25.6 MB
512-1023 MB	16 KB	25%	128.0 MB
1,024-2,047 MB	32 KB	40%	409.6 MB
2,048-4,096 MB	64 KB	50%	1024.0 MB

## Hiding and Unhiding Partitions

PartitionMagic, Drive Image, BootMagic, and VolumeManager allow you to hide a partition from OS detection. When you hide a partition, that partition is not assigned a drive letter during OS boot up. Therefore, the partition is invisible to the OS and all connected applications. Subsequent partitions that are still visible to the OS are assigned a new drive letter.

Hiding a partition is useful when you need to protect sensitive data from other users or you want to prevent others from inadvertently deleting critical files. You can hide any primary or logical FAT, FAT32, NTFS, or HPFS partition.

Of course, PartitionMagic, Drive Image, BootMagic, and VolumeManager also allows you to unhide any partitions that you have hidden. Making a partition visible allows the booted OS to detect the partition, and therefore assign it a drive letter. Once a partition is unhidden, subsequent partitions are again assigned a new drive letter.

**IMPORTANT!** Partitions can only be detected by OSs that recognize the partition's file system. Unhiding a partition with an unrecognized file type will not make the partition visible to the OS.

**WARNING!** You must be careful when unhiding primary partitions. In general, you should not make two primary partitions visible at the same time, as this can cause data loss in some OSs.

For specific instructions on hiding and unhiding partitions, see the product user guide, or refer to online help.

## Understanding Drive Letters

When you boot an OS, that OS assigns drive letters (e.g., C:, D:, etc.) to the primary and logical partitions on each hard disk. These drive letters are used by you, your system, and all your applications to reference files on the partition.

Your OS may change the drive letter assignments if you add or remove a second hard disk. Drive letter assignments may also be altered if you add, remove, or copy a disk partition; reformat a partition with a different file system; or boot a different OS. These kinds of drive letter changes can sometimes invalidate parts of your system configuration. For example, applications that are programmed to look for startup files on a specific drive may no longer launch.

To avoid configuration changes and/or fix configuration problems, you need to understand the following:

- How the OS assigns drive letters;
- Problems caused by drive letter changes;
- What you can do when partitioning to avoid drive letter changes;
- How to fix configuration problems caused by unavoidable changes.

## ***How the OS Assigns Drive Letters***

### **DOS, Windows 3.x, Windows 95/98/Me, and OS/2**

These OSs assign drive letters in a fixed sequence which cannot be changed. This sequence is as follows:

- The OS begins by assigning a drive letter to the first primary partition that it recognizes on the first system hard disk. The OS then assigns drive letters to the first primary partition recognized on each successive hard disk. For example, imagine you have three hard disks in your system. When you boot your OS, it assigns drive letter C: to the active primary partition on the first hard disk. Drive letter D: is assigned to the first primary partition that the OS recognizes on the second hard disk, and drive letter E: is likewise assigned to the first primary partition on the third disk.

If you have multiple visible primary partitions on a single hard disk, the OS assigns the drive letter to the active partition. If none of the partitions are active, the drive letter is assigned to the first visible primary partition recognized by the OS.

**WARNING!** Making multiple primary partitions visible on the same drive can cause data loss in DOS, Windows 3.x, Windows 95/98/Me and OS/2.

- Next, all logical partitions recognized by the OS are assigned drive letters, starting with the logical partitions on the first hard disk and proceeding in order. For example, suppose you have two hard disks in your system, each with one primary and two logical partitions. The OS first assigns C: and D: to the two primary partitions, then assigns drive letters E: and F: to the first and second logical partitions on the first hard disk. Drive letters G: and H: are assigned to the two logical partitions on the second disk.
- The OS then assigns drive letters to any remaining visible primary partitions, starting with those on the first hard disk. The OS proceeds to any visible primary partitions on the second disk, then the third disk, and so on.
- Finally, CD-ROM drives and other types of removable media are assigned a drive letter.

Because the OS always follows this sequence to assign drive letters, adding or removing a second hard disk can cause changes to your drive letter assignments. Likewise, drive letters can change if you add, remove, or copy a disk partition; reformat a partition with a different file system; or boot a different OS.

## Windows NT/2000

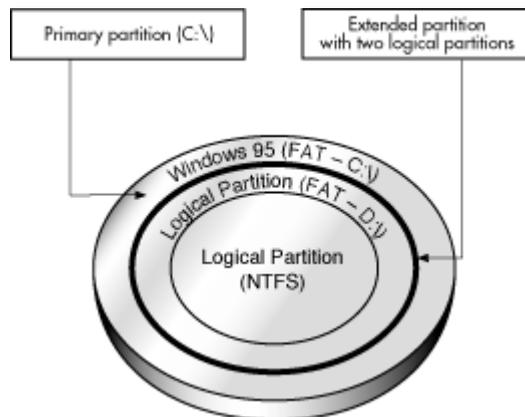
When Windows NT/2000 is first installed, it assigns drive letters in the same manner as described above. Once assigned, however, these drive letters do not change, regardless of changes to the hard disks or partitions in your system. The drive letters are “sticky,” so to speak, and remain permanently assigned to the same partitions.

If you want to reassign or remove a drive letter in Windows NT/2000, you can either use PartitionMagic, VolumeManager for NT, or the Windows NT/2000 Disk Administrator utility.

## Drive Letter Scenarios

To illustrate how drive letters are assigned, consider the following examples.

A computer has one hard disk, on which Windows 95 is installed. The partitioned disk is shown below (Figure 6).

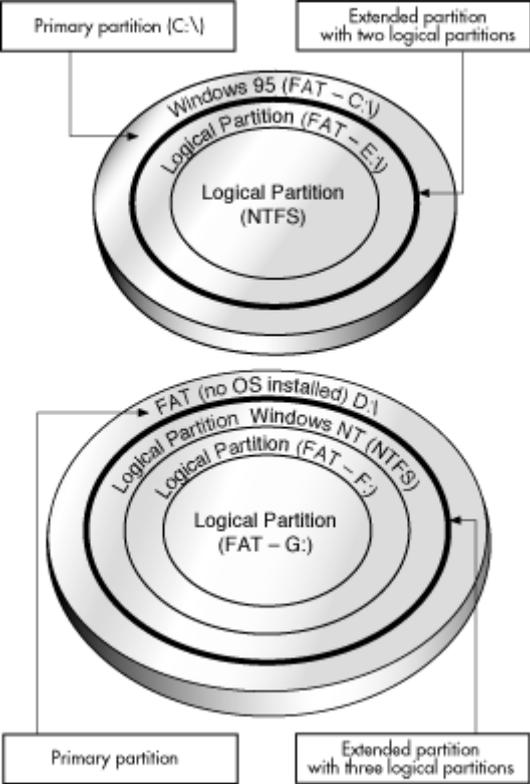


**Figure 6:** Computer with a single hard disk containing one primary and two logical partitions

The disk is partitioned into a primary partition and an extended partition that contains two logical partitions. The primary partition is formatted with the FAT file system and has Windows 95 installed. The first logical partition is formatted with the FAT file system, which Windows 95 recognizes, but the second logical partition is formatted with NTFS, a file system that Windows 95 does not recognize.

On this disk, Windows 95 would assign the drive letter C: to the primary partition and the drive letter D: to the first logical partition. It would not assign a drive letter to the second logical partition because it does not recognize the file system on that drive.

Now consider a computer identical to the first example except that a second hard disk has been installed.



**Figure 7:** Computer with two hard disks, booted from Windows 95

The first hard disk is partitioned in exactly the same way as in the first example. It has the same file systems on the partitions, and Windows 95 is installed on the single primary partition.

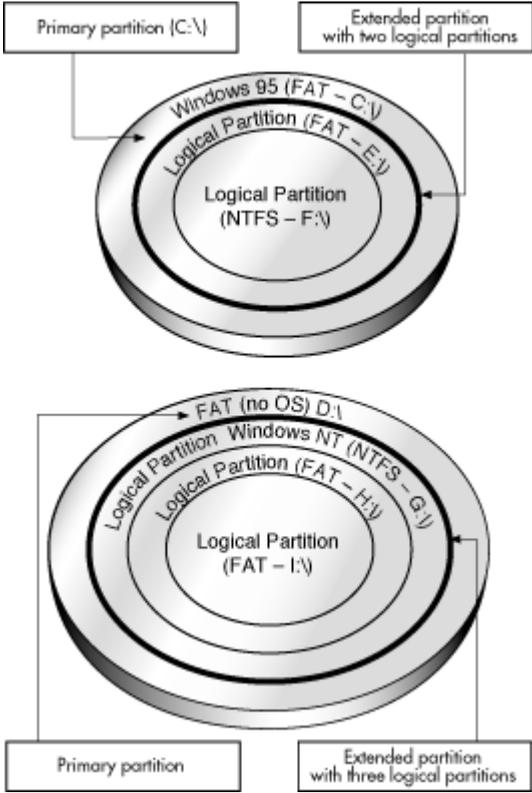
The second disk also has one primary partition and an extended partition; however, its extended partition contains three logical partitions. There is no OS installed on the primary FAT partition of the second hard disk. The first logical partition is an NTFS partition with Windows NT installed. The two remaining logical partitions are FAT partitions.

When the computer is booted from Windows 95, Windows 95 first assigns the drive letter C: to the active primary partition on the first hard disk. Next, Windows 95 assigns the letter D: to the first recognized primary partition on the second hard disk (i.e., the FAT

primary partition). Windows 95 then assigns drive letters to each logical partition that it recognizes. Therefore, it assigns drive letter E: to the first logical FAT partition on the first disk, but skips the second logical partition because Windows 95 does not recognize the NTFS file system. On the second disk, it skips the first logical NTFS partition, assigns drive letter F: to the second logical partition (a FAT partition), and assigns drive letter G: to the third logical partition (also a FAT partition).

In this second example, it is important to notice that the drive letter assigned to the first logical partition on the first hard disk changed, even though the computer was booted with the same OS as in the first example, and the partitioning of the first disk did not change. The drive letter change is the result of a second drive being installed in the computer; Windows 95 must assign a drive letter (D:) to the first recognized primary partition on that second drive *before* assigning one to the first logical partition on the first drive.

In a final example, consider how drive letters are assigned on the same computer, with exactly the same hard disks and partitions as used in the previous example; however, this time the computer (Figure 8) is booted with a newly-installed version of Windows NT (installed in the first logical partition on the second disk).



**Figure 8:** Computer with two hard disks, booted from Windows NT

First, because Windows NT recognizes the FAT file system, it assigns the drive letter C: to the primary partition on the first hard disk. The letter D: is assigned to the primary partition on the second hard disk, which is also FAT. Windows NT then assigns drive letters to each logical partition in order, because all these partitions contain file systems Windows NT recognizes (either FAT or NTFS). On the first disk, Windows NT assigns E: to the first logical partition and F: to the second logical partition. Windows NT then assigns G: to the first logical partition, H: to the second logical partition, and I: to the last logical partition.

It is important to notice in this example that the drive letters assigned to the recognized logical partitions changed from those assigned in example two, even though the number of hard disks and partitions are exactly the same. The only difference is that the computer was booted with a new installation of Windows NT, which recognized the file system on all partitions and thus assigned them all letters.

**TIP:** After Windows NT is installed, its assigned drive letters do not change, regardless of changes to the hard disks or partitions in your system. The “sticky” drive letters remain permanently assigned to the same partitions.

Drive letter changes can occur for other reasons than those illustrated in the above examples. Adding, deleting, hiding/unhiding, or reformatting a partition can also cause changes in drive lettering.

### ***Problems Caused by Drive Letter Changes***

Changes in your computer’s drive letters can disable your application configurations. For example, suppose you install several programs to a logical partition with the drive letter D:. You decide to create icons for these programs, so you can start them from your Windows 95 OS. Each time you double-click an icon, Windows looks on drive D: to find and launch the corresponding program. If the drive letter for the logical partition changes, however, your icons will no longer point to the correct partition. When you double-click an icon, Windows 95 continues to look on drive D: for the program files, even though D: now identifies a different partition.

Changing the drive letter of a partition also affects all system configurations that are based on the original drive letter of the partition. For instance, commands based on a drive letter that are entered in your AUTOEXEC.BAT, CONFIG.SYS, WIN.INI, SYSTEM.INI, or other system files might be invalidated by drive letter changes. Likewise, Windows 95/98 registry entries that include drive letter references may become inoperative if those letters change.

### ***Partitioning to Avoid Drive Letter Changes***

Using the following partitioning strategies can help you avoid unwanted drive letter changes when using DOS, Windows 3.x/95/98/Me, and OS/2.

## **Preventing Changes Caused by Adding Primary Partitions**

To avoid drive letter assignment changes caused by adding a primary partition, add primary partitions only to hard disks that already have at least one primary partition. The additional primary partitions can be hidden, thus keeping only one primary partition visible on each drive. While not always possible, this strategy prevents changes in partition letter assignments.

## **Preventing Changes Caused by Adding Logical Partitions**

Whenever possible, add a new logical partition as the last logical partition on the last hard disk. By doing so, you ensure that the drive letter assignments for all existing partitions remain the same. If the partition must be added to a hard disk other than the last, try adding it as the last logical partition on the target disk. The drive letter assignments for the logical partitions on that disk remain unchanged, as do the drive letters for any previous hard disks; however, all the logical partitions on subsequent disk drives are assigned new drive letters.

**TIP:** If there is unallocated space between existing partitions on a hard disk, move all the partitions to the left until all the unallocated space is shifted to the right end of the disk. You can then use this space to create a new logical partition at the end of the disk.

**TIP:** For more information on moving partitions, see the PartitionMagic or VolumeManager user guide, or refer to online help.

## **Preventing Changes Caused by Booting a Different OS**

You can prevent many drive letter changes caused by booting different operating systems. If you have partitions formatted with file systems recognized by only one or two of your OSs, simply place them after any partitions with file systems recognized by all your OSs.

For example, suppose you use both DOS and Windows NT. Some of your partitions are FAT partitions, while others are NTFS partitions. Because both DOS and Windows NT recognize FAT partitions, place these partitions first on your disk drives. The NTFS partitions can then be positioned at the ends of the drives. Now whenever you boot up, the drive letters assigned to your FAT partitions remain the same, regardless of whether you boot with DOS or Windows NT.

FAT partitions are recognized by the largest number of OSs. Therefore, we recommend placing all FAT partitions before any FAT32, NTFS, or HPFS partitions.

## ***Fixing Configuration Problems Caused by Drive Letter Changes***

You can fix application configuration problems caused by drive letter changes with PartitionMagic's DriveMapper utility. DriveMapper quickly and automatically replaces all invalidated (reassigned) drive letter references with the newly assigned partition drive letter. For more information on using DriveMapper, see the PartitionMagic user guide, or refer to online help.

Because Windows NT/2000 does not reassign drive letters once they have been assigned, DriveMapper is not normally needed with Windows NT/2000.

After a partition is created or deleted, the OS sometimes fails to assign a drive letter to the CD-ROM drive upon reboot. If this happens, see the Troubleshooting section of the product user guide, or refer to the product's online help.

## **Understanding the BIOS 1,024 Cylinder Limit**

You can safely use PartitionMagic or VolumeManager to partition any drive, regardless of the number of cylinders on the drive. In fact, to prevent you from performing partition operations that might cause problems, the products are careful to observe the BIOS 1,024 cylinder limit on systems where it applies.

The BIOS 1,024 cylinder limit pertains to your system only if:

- You have a hard disk with a capacity of more than 504 MB.
- The hard disk was manufactured before (approximately) 1994.
- The BIOS in your system does not have the INT 13 BIOS extensions built in.

You may also encounter problems if your hard disk is larger than 8 GB.

If you have such a system and you use only DOS, neither the DOS FDISK utility nor any PowerQuest product allows you to see cylinders beyond the 1,024th cylinder or include them in any partition. Space beyond the 1,024th cylinder always remains invisible.

Even if the BIOS 1,024 cylinder limit applies to your system, however, you can use PartitionMagic without difficulty. The only instance where you may encounter problems is if all the following criteria apply:

- you use both DOS and another OS;
- the other OS is able to see and use disk space past the first 1,024 cylinders of the disk;

- You use PartitionMagic or the FDISK utility of the other OS to create a partition extending beyond the 1,024th cylinder;
- You then run the DOS PartitionMagic executable.

When you run the DOS PartitionMagic executable, you may or may not be able to see the newly-created partition that extends beyond the 1,024 cylinder limit. Even if you can see the partition, you cannot use the DOS PartitionMagic executable to perform any operations on that partition. This restriction applies to both primary and extended partitions that contain space beyond the 1,024th cylinder. If an extended partition exceeds the cylinder limit, you cannot perform operations on any one of the contained logical partitions, even if the logical partition itself does not extend past the 1,024th cylinder.

**TIP:** If you use DOS, but your system has the INT 13 BIOS extensions, PartitionMagic lets you see cylinders past the normal 1,024 limit and perform operations on partitions extending beyond this boundary.

## Understanding the 2 GB Boot Code Boundary

An OS's boot code is stored in both the master boot record (MBR) and the partition boot record, enabling the OS to boot properly. In some OSs, however, this boot code is written in such a way that it inadvertently imposes a limit on the location of both the partition boot record and the files needed to boot the OS.

DOS (versions 6.x and earlier) and Windows NT (versions 4.0 and earlier), and Windows 2000 are both affected by this boot code limitation. When booting these two operating systems, the cylinder-head-sector (CHS) address of the beginning boot code sector must be calculated in order to retrieve the sector's information and load and execute the next part of the boot process. The CHS value for the needed sector is calculated as follows:

$$\text{Sector Number} / \text{Sectors Per Track}$$

Because of the way the boot code is written, the product of this calculation must fit in a 16-bit register. The largest value a 16-bit register can contain is 64K. If the number is larger than 64K, the number is truncated, resulting in an incorrect value that skews the remaining calculations. The boot process fails to load and execute the needed sector, thus preventing the OS from booting.

Most current hard disks have 63 sectors per track, creating a 64K boot code boundary at 2 GB.

If your disk is older or uses drive overlay software, this boundary may be lower.

If a partition begins or extends beyond this boundary, the CHS value of the partition's boot code sector cannot be correctly calculated; therefore, the partition and its OS cannot boot.

This same limit applies to the DOS IO.SYS file and the Windows NT file, NTLOADER.EXE. If either of these files are installed or moved beyond the 64K boot code boundary, the corresponding OS fails to boot.

**IMPORTANT!** To boot properly, DOS requires that the first three sectors of IO.SYS are below the 2 GB boot code boundary.

Both the IO.SYS and NTLOADER.EXE files are usually located near the beginning of the partition in which they are installed. When you resize partitions using PartitionMagic, Drive Image, and VolumeManager, this area of the partition may be vacated to make room for a larger FAT or other file system structures. As a result, IO.SYS or NTLOADER.EXE may be moved beyond the 64K boundary, thus preventing the OS from booting.

## Changing the BIOS LBA Mode Setting

**WARNING!** Never change the LBA mode in your system BIOS once data is present on any hard disk. Changing this setting may cause data corruption and loss.

Most modern system BIOS designs support LBA or Logical Block Addressing. The LBA mode setting, whether enabled or disabled on your system, determines how your computer translates logical cylinder-head-sector (CHS) addresses. If you change this setting, the resulting shift in CHS values may corrupt all the files and partitions on your hard disks.

If you must change the LBA mode setting in your system BIOS, first back up all data on your hard disks. Contact the BIOS and/or disk manufacturer's technical support departments to ensure you understand how to proceed safely.

## Restoring System Files

When you use PartitionMagic or VolumeManager to create or modify your computer's partitions, these products make the necessary changes to your system files, allowing your computer to recognize the new or modified partition. Because these files are changed, any

previous system backups you have made are no longer accurate. Restoring system files from a backup created prior to using PartitionMagic or VolumeManager may cause problems.

**TIP:** Whenever you make changes using PartitionMagic, Drive Image, BootMagic, or VolumeManager, be sure to create new backups for your system files. Never use a previous backup to restore the following files: BOOT.INI, BOOT.DOS, BOOT.OS2, BOOTSECT.DOS, BOOTSECT.W95, or BOOTSECT.W98.

**TIP:** PartitionMagic and VolumeManager only modify system files. Therefore, you can use previous backups to restore user files.