

Master Boot Record

The MBR, the most important data structure on the disk, is created when the disk is partitioned. The MBR contains a small amount of executable code called the master boot code, the disk signature, and the partition table for the disk. At the end of the MBR is a 2-byte structure called a signature word or end of sector marker, which is always set to 0x55AA. A signature word also marks the end of an extended boot record (EBR) and the boot sector.

The disk signature, a unique number at offset 0x01B8, identifies the disk to the operating system. Windows 2000 uses the disk signature as an index to store and retrieve information about the disk in the registry subkey:

HKEY_LOCAL_MACHINE\SYSTEM\MountedDevices

Master Boot Code

The master boot code performs the following activities:

1. Scans the partition table for the active partition.
2. Finds the starting sector of the active partition.
3. Loads a copy of the boot sector from the active partition into memory.
4. Transfers control to the executable code in the boot sector.

If the master boot code cannot complete these functions, the system displays one of the following error messages:

- Invalid partition table
- Error loading operating system
- Missing operating system



Note

There is no MBR on a floppy disk. The first sector on a floppy disk is the boot sector. Although every hard disk contains an MBR, the master boot code is used only if the disk contains the active, primary partition.

For more information about troubleshooting MBR problems, see "Damaged MBRs and Boot Sectors" later in this chapter.

Partition Table

The partition table, a 64-byte data structure used to identify the type and location of partitions on a hard disk, conforms to a standard layout independent of the operating system. Each partition table entry is 16 bytes long, with a maximum of four entries. Each entry starts at a predetermined offset from the beginning of the sector, as follows:

- Partition 1\ 0x01BE(446)
- Partition 2\ 0x01CE(462)
- Partition 3\ 0x01DE(478)
- Partition 4\ 0x01EE(494)



Note

Only basic disk makes use of the partition table in Windows 2000. Dynamic disk uses the Disk Management database located at the end of the disk for disk configuration information. The partition table is not updated when volumes are deleted or extended after the dynamic disk upgrade, or when new dynamic volumes are created.

The following example shows a partial printout of an MBR revealing the partition table from a computer with three partitions. When there are fewer than four partitions on a disk, the remaining partition table fields are set to the value 0.

```
000001B0: 80 01 ..
000001C0: 01 00 07 FE BF 09 3F 00 - 00 00 4B F5 7F 00 00 00 .....?...K. ...
000001D0: 81 0A 07 FE FF FF 8A F5 - 7F 00 3D 26 9C 00 00 00 ..... .=&....
000001E0: C1 FF 05 FE FF FF C7 1B - 1C 01 D6 96 92 00 00 00 .....
000001F0: 00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 .....
```

Table 1.2 describes the fields in each entry in the partition table. The sample values correspond to the first partition table entry shown in the preceding example. The Byte Offset values correspond to the addresses of the first partition table entry. There are three additional entries whose values can be calculated by adding 10h to the byte offset value specific for each additional partition table entry (for example, add 20h for partition table entry 3 and 30h for partition table entry 4).

Table 1.2 Partition Table Fields

Byte Offset	Field Length	Sample Value	Field Name and Definition
0x01BE	BYTE	0x80	Boot Indicator . Indicates

			whether the volume is the active partition. Legal values include: 00. Do not use for booting. 80. Active partition.
0x01BF	BYTE	0x01	Starting Head .
0x01C0	6 bits	0x01 *	Starting Sector . Only bits 0-5 are used. The upper two bits, 6 and 7, are used by the Starting Cylinder field.
0x01C1	10 bits	0x00 *	Starting Cylinder . Uses 1 byte in addition to the upper 2 bits from the Starting Sector field to make up the cylinder value. The Starting Cylinder is a 10-bit number, with a maximum value of 1023.
0x01C2	BYTE	0x07	System ID . Defines the volume type. See Table 1.3 for sample values.
0x01C3	BYTE	0xFE	Ending Head .
0x01C4	6 bits	0xBF *	Ending Sector . Only bits 0-5 are used. The upper two bits, 6 and 7, are used by the Ending Cylinder field.
0x01C5	10 bits	0x09 *	Ending Cylinder . Uses 1 byte in addition to the upper 2 bits from the Ending Sector field to make up the cylinder value. The Ending Cylinder is a 10-bit number, with a maximum value of 1023.
0x01C6	DWORD	0x3F000000	Relative Sectors . The offset from the beginning of the disk to the beginning of the volume, counting by sectors.
0x01CA	DWORD	0x4BF57F00	Total Sectors . The total number of sectors in the volume.

A BYTE is 8 bits, a WORD is 16 bits, a DWORD is 32 bits, and a LONGLONG is

64 bits. Sample values marked with an asterisk (*) do not accurately represent the value of the fields, because the fields are either 6 bits or 10 bits and the data is recorded in bytes.

Numbers larger than one byte are stored in little endian format, or reverse-byte ordering. Little endian format is a method of storing a number so that the least significant byte appears first in the hexadecimal number notation. For example, the sample value for the **Relative Sectors** field in the previous table, 0x3F000000, is a little endian representation of 0x0000003F. The decimal equivalent of this little endian number is 63.

Boot Indicator Field

The first element of the partition table, the **Boot Indicator** field, indicates whether or not the volume is the active partition. Only one primary partition on the disk can have this field set.

It is possible to have different operating systems and different file systems on different volumes. By using disk configuration tools such as the Windows 2000-based Disk Management or the MS-DOS-based Fdisk to designate a primary partition as active, the **Boot Indicator** field for that partition is set in the partition table.

System ID Field

Another element of the partition table is the **System ID** field. It defines which file system, such as FAT16, FAT32, or NTFS, was used to format the volume and the FT characteristics of the volume. The **System ID** field also identifies an extended partition, if one is defined.

Windows 2000 uses the **System ID** field to determine which file system device drivers to load during startup. Table 1.3 identifies the values for the **System ID** field.

Table 1.3 System ID Values

Partition Type	ID Value
0x01	FAT12 primary partition or logical drive (fewer than 32,680 sectors in the volume)
0x04	FAT16 partition or logical drive (32,680–65,535 sectors or 16 MB–33 MB)

0x05	Extended partition
0x06	BIGDOS FAT16 partition or logical drive (33 MB–4 GB)
0x07	Installable File System (NTFS partition or logical drive)
0x0B	FAT32 partition or logical drive
0x0C	FAT32 partition or logical drive using BIOS INT 13h extensions
0x0E	BIGDOS FAT16 partition or logical drive using BIOS INT 13h extensions
0x0F	Extended partition using BIOS INT 13h extensions
0x12	EISA partition
0x42	Dynamic disk volume
0x86	Legacy FT FAT16 disk *
0x87	Legacy FT NTFS disk *
0x8B	Legacy FT volume formatted with FAT32 *
0x8C	Legacy FT volume using BIOS INT 13h extensions formatted with FAT32 *

Partition types denoted with an asterisk (*) indicate that they are also used to designate non-FT configurations such as striped and spanned volumes.

When a mirrored or RAID-5 volume is created in Windows NT 4.0 or earlier, the high bit of the **System ID** field byte is set for each primary partition or logical drive that is a member of the volume. For example, a FAT16 primary partition or logical drive that is a member of a mirrored or RAID-5 volume, has a **System ID** value of 0x86. A FAT32 primary partition or logical drive has a **System ID** value of 0x8B, and an NTFS primary partition or logical drive has a **System ID** value of 0x87. Volumes that have the high bit set can only be directly accessed by Windows 2000 and Windows NT. Disk shares on FT disks, however, are also available to computers running MS-DOS, Windows 95, and Windows 98.



Note

MS-DOS can only access volumes that have a **System ID** value of 0x01, 0x04, 0x05, or 0x06. However, you can delete volumes that have the other values listed in Table 1.3 using MS-DOS tools such as Fdisk. If you use a low-level disk editor, such as DiskProbe, you can read and write to any sector, including ones that are in NTFS volumes.

Starting and Ending Cylinder, Head, and Sector Fields

The **Starting** and **Ending Cylinder** , **Head** , and **Sector** fields (collectively known as the **CHS** fields) are additional elements of the partition table. These fields are essential for starting the computer. The master boot code uses these fields to find and load the boot sector of the active partition. The **Starting CHS** fields for non-active partitions point to the boot sectors of the remaining primary partitions and the EBR of the first logical drive in the extended partition as shown in Figure 1.1.

Knowing the starting sector of an extended partition is very important for low-level disk troubleshooting. If your disk fails, you need to work with the partition starting point (among other factors) to retrieve stored data.



Note

To have a written record of the starting and ending sectors of the partitions on your hard disk, as well as other useful disk configuration data, use the DiskMap tool. For more information about DiskMap, see the documentation provided on the *Windows 2000 Resource Kit* companion CD.

Figure 1.1 shows the MBR, partition table, and boot sectors on a disk with four partitions. The definitions of the fields in the partition table and the extended partition tables are the same.

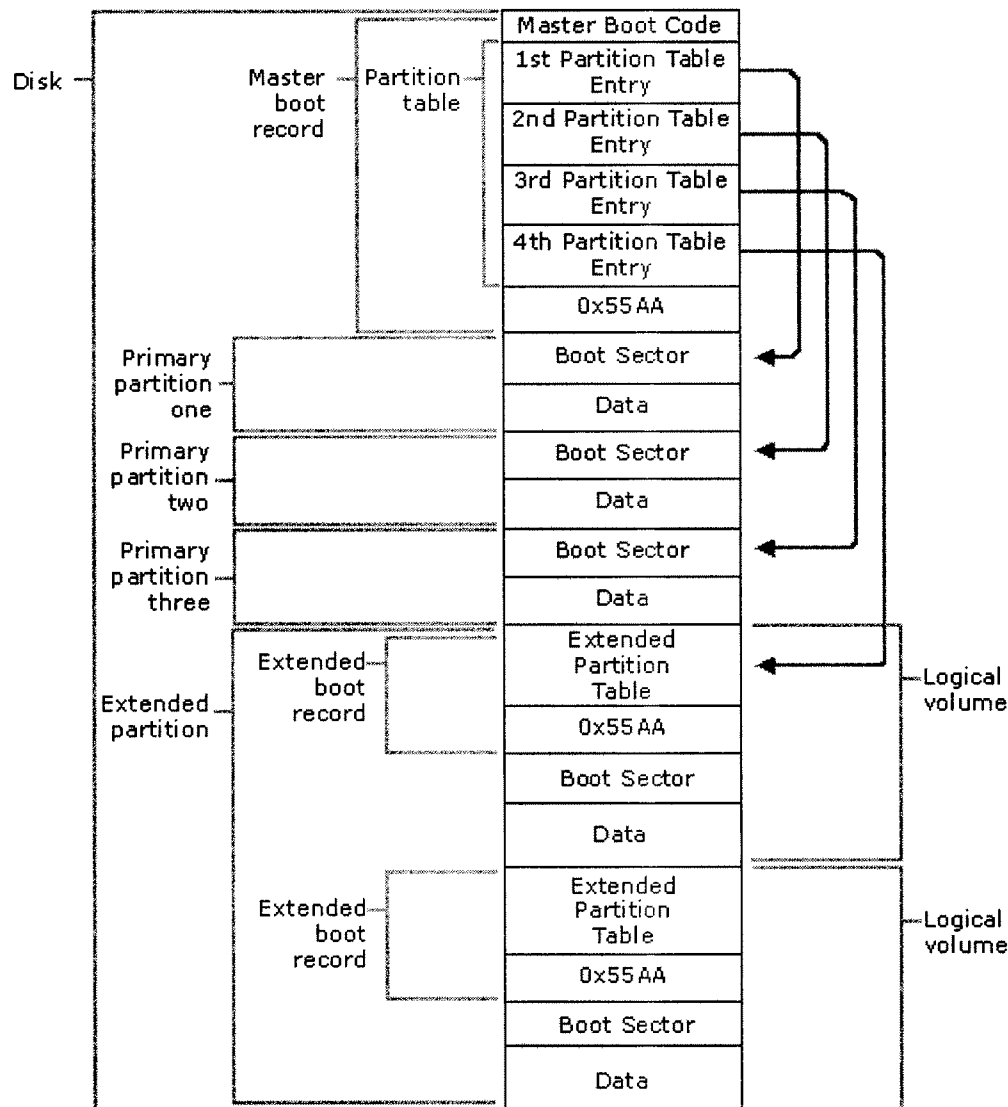


Figure 1.1 Detail of a Basic Disk with Four Partitions

The **Ending Cylinder** field in the partition table is 10 bits long, which limits the number of cylinders that can be described in the partition table to a range of 0–1,023. The **Starting Head** and **Ending Head** fields are each one byte long, which limits the field range to 0–255. The **Starting Sector** and **Ending Sector** fields are each six bits long, which limits the range of these fields to 0–63. However, the enumeration of sectors starts at 1 (not 0, as for other fields), so the maximum number of sectors per track is 63.

Because all hard disks are low-level formatted with a standard 512-byte sector, the maximum disk capacity described by the partition table is calculated as follows:

Maximum capacity = sector size x cylinders (10 bits) x heads (8 bits) x sectors per track (6 bits)

Using the maximum possible values yields:

$$512 \times 1024 \times 256 \times 63 \text{ (or } 512 \times 2^{24}) = 8,455,716,864 \text{ bytes or 7.8 GB}$$

The calculation results in a maximum capacity of slightly less than 8 gigabytes (GB). Before BIOS INT 13h extensions drive geometry translation (also known as logical block addressing, or LBA) were introduced, the active, primary partition could not exceed 7.8 GB, regardless of the file system used.



Important

When using the standard 512-byte sector, the maximum cluster size that you can use for FAT16 volumes while running Windows 2000 is 64 kilobytes (KB). Therefore, the maximum size for a FAT16 volume is 4 GB.

If you use a multiple-boot configuration with Windows 95, Windows 98, or MS-DOS, FAT16 volumes must be limited to 2 GB to be accessed from these operating systems. In addition, a Macintosh computer that accesses volumes on a computer running Windows 2000 cannot access a FAT16 volume that is larger than 2 GB. If you try to use a FAT16 volume larger than 2 GB when running MS-DOS, Windows 95, or Windows 98, or try to access such a volume from a Macintosh computer, you might get a message that zero bytes are available.

The maximum FAT16 volume size that you can use on a computer depends on the disk geometry and the maximum values that fit in the partition table entry fields. Table 1.4 shows the typical FAT16 volume size when LBA is enabled or disabled. The number of cylinders in both cases is 1,024 (0–1,023). When a primary partition or logical drive extends beyond the 1,023rd cylinder, all fields described in this section contain the maximum values.

Table 1.4 FAT16 Volume Size When LBA Is Enabled or Disabled

Translation Mode	Number of Heads	Sectors per Track	Maximum Size for System or Boot Partition
Disabled	64	32	1 GB
Enabled	255	63	4 GB



Warning

Do not change the LBA setting on any hard disk containing data. You can adversely affect the process in which the system translates the disk attributes for storing data and corrupt all the files and partitions on the physical disk. Refer to your computer owner's manual before modifying this BIOS setting.

To accommodate sizes larger than 7.8 GB, Windows 2000 ignores the values in the **Starting** and **Ending Sector** fields of the partition table in favor of the **Relative Sectors** and **Total Sectors** fields.

Relative Sectors and Total Sectors Fields

The **Relative Sectors** field represents the offset from the beginning of the disk to the beginning of the volume, counting by sectors, for the volume described by the partition table entry. The **Total Sectors** field represents the total number of sectors in the volume.

Using the **Relative Sectors** and **Total Sectors** fields (resulting in a 32-bit number) provides eight more bits than the CHS scheme to represent the total number of sectors. This allows partitions containing up to 2³² sectors to be defined. With a standard sector size of 512 bytes, the 32 bits used to represent the **Relative Sectors** and **Total Sectors** fields translates into a maximum partition size of 2 terabytes (or 2,199,023,255,552 bytes).

This addressing scheme is only used in Windows 2000 with NTFS and FAT32.



Note

In addition, the Format tool of Windows 2000 limits the maximum size of FAT32 volumes it can create to 32 GB. However, Windows 2000 can directly access larger FAT32 volumes created by Windows 95 OSR2 or Windows 98.

Windows 2000 uses the fields in the partition table entries to access all partitions. A partition that is formatted while Windows 2000 is running puts data into the **Starting** and **Ending CHS** fields to have compatibility with MS-DOS, Windows 95, and Windows 98, and to maintain compatibility with the BIOS INT 13h for startup.

Extended Boot Record

An EBR, which consists of an extended partition table and the signature word for the sector, exists for each logical drive in the extended partition. It contains the only information on the first side of the first cylinder of each logical drive in the extended partition. The boot sector in a logical drive is usually located at either Relative Sector 32 or 63. However, if there is no extended partition on a disk, there are no EBRs and no logical drives.



Note

This information applies only to disks configured with basic disk.

The first entry in an extended partition table for the first logical drive points to its own boot sector. The second entry points to the EBR of the next logical drive. If no further logical drives exist, the second entry is not used and is recorded as a series of zeroes. If there are additional logical drives, the first entry of the extended partition table for the second logical drive points to its own boot sector. The second entry of the extended partition table for the second logical drive points to the EBR of the next logical drive. The third and fourth entries of an extended partition table are never used.

As shown in Figure 1.2, the EBRs of the logical drives in the extended partition are a linked list. The figure shows three logical drives on an extended partition, illustrating the difference in extended partition tables between preceding logical drives and the last logical drive.

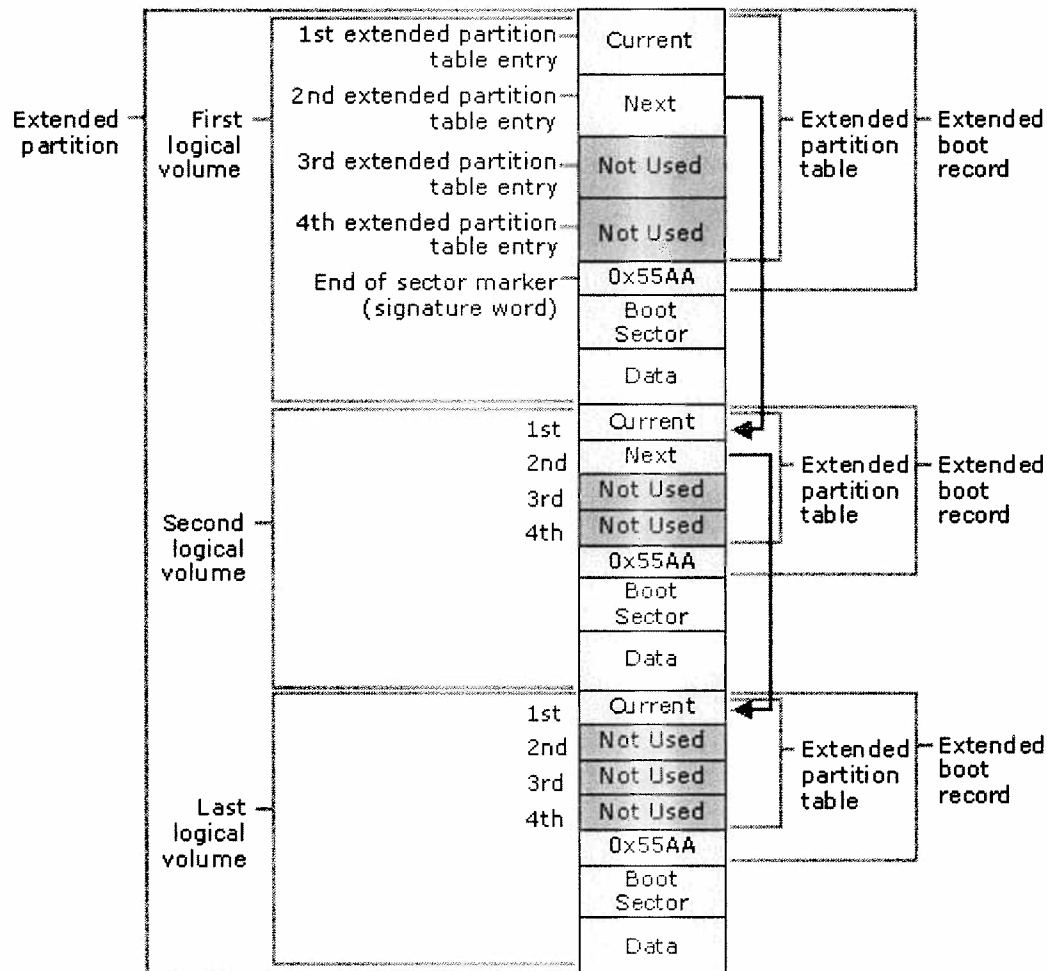


Figure 1.2 Detail of an Extended Partition

With the exception of the last logical drive on the extended partition, the format of the extended partition table, described in Table 1.5, is repeated for each logical drive: the first entry identifies the logical drive's own boot sector and the second entry identifies the next logical drive's EBR.

The extended partition table for the last logical drive has only its own partition entry listed. The second through fourth entries of the last extended partition table are not used.

Table 1.5 Contents of Extended Partition Table Entries

Extended Partition Table Entry	Entry Contents
First	Information about the current logical drive in the extended partition, including the starting address for data.
Second	Information about the next logical drive in the extended partition, including the address of the sector that contains the EBR for the next logical drive. If no further logical drives exist, this field is not used.
Third	Not used
Fourth	Not used

The fields in each entry of the extended partition table are identical to the MBR partition table entries. See Table 1.2 for more information about partition table fields.

The **Relative Sectors** field in an extended partition table entry shows the number of bytes that are offset from the beginning of the extended partition to the first sector in the logical drive. The number in the **Total Sectors** field refers to the number of sectors that make up the logical drive. The value of the **Total Sectors** field equals the number of sectors from the boot sector defined by the extended partition table entry to the end of the logical drive.

Because of the importance of the MBR and EBR sectors, it is recommended that you run disk-scanning tools regularly as well as regularly back up all your data files to protect against losing access to a volume or an entire disk.