

## Sector layout

By convention, there are exactly four primary partition table entries in the MBR partition table scheme, although some operating systems and system tools extended this to five (Advanced Active Partitions (AAP) with PTS-DOS 6.60<sup>[10]</sup> and DR-DOS 7.07), eight (AST and NEC MS-DOS 3.x<sup>[11][12]</sup> as well as Storage Dimensions SpeedStor), or even sixteen entries (with Ontrack Disk Manager).

Structure of a classical generic MBR

Address		Description	Size (bytes)
Hex	Dec		
+000 <sub>hex</sub>	+0	Bootstrap code area	446
+1BE <sub>hex</sub>	+446	Partition entry №1	16
+1CE <sub>hex</sub>	+462	Partition entry №2	
+1DE <sub>hex</sub>	+478	Partition entry №3	
+1EE <sub>hex</sub>	+494	Partition entry №4	
+1FE <sub>hex</sub>	+510	55 <sub>hex</sub>	2
+1FF <sub>hex</sub>	+511	AA <sub>hex</sub>	
Total size: 446 + 4×16 + 2			512

Structure of a modern standard MBR

Address		Description	Size (bytes)
Hex	Dec		
+000 <sub>hex</sub>	+0	Bootstrap code area (part 1)	218
+0DA <sub>hex</sub>	+218	0000 <sub>hex</sub>	2
+0DC <sub>hex</sub>	+220	Original physical drive (80 <sub>hex</sub> –FF <sub>hex</sub> )	1
+0DD <sub>hex</sub>	+221	Seconds (0–59)	
+0DE <sub>hex</sub>	+222	Minutes (0–59)	
+0DF <sub>hex</sub>	+223	Hours (0–23)	
+0E0 <sub>hex</sub>	+224	Bootstrap code area (part 2, code entry at +000 <sub>hex</sub> )	216 (or 222)
+1B8 <sub>hex</sub>	+440	32-bit disk signature	4
+1BC <sub>hex</sub>	+444	0000 <sub>hex</sub> (5A5A <sub>hex</sub> if copy-protected)	
+1BE <sub>hex</sub>	+446	Partition entry №1	16
+1CE <sub>hex</sub>	+462	Partition entry №2	
+1DE <sub>hex</sub>	+478	Partition entry №3	
+1EE <sub>hex</sub>	+494	Partition entry №4	
+1FE <sub>hex</sub>	+510	55 <sub>hex</sub>	2
+1FF <sub>hex</sub>	+511	AA <sub>hex</sub>	
Total size: 218 + 6 + 216 + 6 + 4×16 + 2			512

## Partition table entries

An artifact of hard disk technology from the era of the PC XT, the partition table subdivides a storage medium using units of cylinders, heads, and sectors (CHS addressing). These values no longer correspond to their namesakes in modern disk drives, as well as being irrelevant in other devices such as solid-state drives, which do not physically have cylinders or heads.

In the CHS scheme, sector indices have (almost) always begun with sector 1 rather than sector 0 by convention, and due to an error in all versions of MS-DOS/PC DOS up to including 7.10, the number of heads is generally limited to 255<sup>[9]</sup> instead of 256. When a CHS address is too large to fit into these fields, the *tuple* (1023, 254, 63) is typically used today, although on older systems, and with older disk tools, the cylinder value often wrapped around modulo the CHS barrier near 8 GB, causing ambiguity and risks of data corruption. (If the situation involves a "protective" MBR on a disk with a GPT, Intel's *Extensible Firmware Interface* specification requires that the tuple (1023, 255, 63) be used.) The 10-bit cylinder value is recorded within two bytes in order to facilitate making calls to the original/legacy INT 13h BIOS disk access routines, where 16 bits were divided into sector and cylinder parts, and not on byte boundaries.<sup>[14]</sup>

Offset (bytes)	Field length	Description																	
+0 <sub>hex</sub>	1 byte	Status or physical drive (bit 7 set is for active or bootable, old MBRs only accept 80 <sub>hex</sub> , 00 <sub>hex</sub> means inactive, and 01 <sub>hex</sub> –7F <sub>hex</sub> stand for invalid) <sup>[c]</sup>																	
+1 <sub>hex</sub>	3 bytes	CHS address of first absolute sector in partition. <sup>[d]</sup> The format is described by three bytes, see the next three rows.																	
+1 <sub>hex</sub>	1 byte	<table border="1"> <tr> <td colspan="8">h<sub>7-0</sub></td> <td rowspan="2">head<sup>[e]</sup></td> </tr> <tr> <td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td> </tr> </table>	h <sub>7-0</sub>								head <sup>[e]</sup>	x	x	x	x	x	x	x	x
h <sub>7-0</sub>								head <sup>[e]</sup>											
x	x	x	x	x	x	x	x												
+2 <sub>hex</sub>	1 byte	<table border="1"> <tr> <td colspan="3">C<sub>9-8</sub></td> <td colspan="5">S<sub>5-0</sub></td> <td rowspan="2">sector in bits 5–0; bits 7–6 are high bits of cylinder<sup>[e]</sup></td> </tr> <tr> <td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td> </tr> </table>	C <sub>9-8</sub>			S <sub>5-0</sub>					sector in bits 5–0; bits 7–6 are high bits of cylinder <sup>[e]</sup>	x	x	x	x	x	x	x	x
C <sub>9-8</sub>			S <sub>5-0</sub>					sector in bits 5–0; bits 7–6 are high bits of cylinder <sup>[e]</sup>											
x	x	x	x	x	x	x	x												
+3 <sub>hex</sub>	1 byte	<table border="1"> <tr> <td colspan="8">C<sub>7-0</sub></td> <td rowspan="2">bits 7–0 of cylinder<sup>[e]</sup></td> </tr> <tr> <td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td> </tr> </table>	C <sub>7-0</sub>								bits 7–0 of cylinder <sup>[e]</sup>	x	x	x	x	x	x	x	x
C <sub>7-0</sub>								bits 7–0 of cylinder <sup>[e]</sup>											
x	x	x	x	x	x	x	x												
+4 <sub>hex</sub>	1 byte	Partition type <sup>[15]</sup>																	
+5 <sub>hex</sub>	3 bytes	CHS address of last absolute sector in partition. <sup>[d]</sup> The format is described by 3 bytes, see the next 3 rows.																	
+5 <sub>hex</sub>	1 byte	<table border="1"> <tr> <td colspan="8">h<sub>7-0</sub></td> <td rowspan="2">head<sup>[e]</sup></td> </tr> <tr> <td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td> </tr> </table>	h <sub>7-0</sub>								head <sup>[e]</sup>	x	x	x	x	x	x	x	x
h <sub>7-0</sub>								head <sup>[e]</sup>											
x	x	x	x	x	x	x	x												
+6 <sub>hex</sub>	1 byte	<table border="1"> <tr> <td colspan="3">C<sub>9-8</sub></td> <td colspan="5">S<sub>5-0</sub></td> <td rowspan="2">sector in bits 5–0; bits 7–6 are high bits of cylinder<sup>[e]</sup></td> </tr> <tr> <td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td> </tr> </table>	C <sub>9-8</sub>			S <sub>5-0</sub>					sector in bits 5–0; bits 7–6 are high bits of cylinder <sup>[e]</sup>	x	x	x	x	x	x	x	x
C <sub>9-8</sub>			S <sub>5-0</sub>					sector in bits 5–0; bits 7–6 are high bits of cylinder <sup>[e]</sup>											
x	x	x	x	x	x	x	x												
+7 <sub>hex</sub>	1 byte	<table border="1"> <tr> <td colspan="8">C<sub>7-0</sub></td> <td rowspan="2">bits 7–0 of cylinder</td> </tr> <tr> <td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td> </tr> </table>	C <sub>7-0</sub>								bits 7–0 of cylinder	x	x	x	x	x	x	x	x
C <sub>7-0</sub>								bits 7–0 of cylinder											
x	x	x	x	x	x	x	x												
+8 <sub>hex</sub>	4 bytes	LBA of first absolute sector in the partition <sup>[f]</sup>																	
+C <sub>hex</sub>	4 bytes	Number of sectors in partition <sup>[f]</sup>																	

addressing,<sup>[17][18]</sup> a transition was made to using LBA, or **logical block addressing**. Both the partition length and partition start address are sector values stored in the partition table entries as 32-bit quantities. The sector size used to be considered fixed at 512 (2<sup>9</sup>) bytes, and a broad range of important components including **chipsets**, **boot sectors**, **operating systems**, **database engines**, **partitioning tools**, **backup** and **file system** utilities and other software had this value hard-coded. Since the end of 2009, disk drives employing 4096-byte sectors (**4Kn** or **Advanced Format**) have been available, although the size of the sector for some of these drives was still reported as 512 bytes to the host system through conversion in the hard-drive firmware and referred to as 512 emulation drives (**512e**).

Since block addresses and sizes are stored in the partition table of an MBR using 32 bits, the maximal size, as well as the highest start address, of a partition using drives that have 512-byte sectors (actual or emulated) cannot exceed 2 TiB–512 bytes (2,199,023,255,040 bytes or 4,294,967,295 (2<sup>32</sup>–1) sectors × 512 (2<sup>9</sup>) bytes per sector).<sup>[2]</sup> Alleviating this capacity limitation was one of the prime motivations for the development of the GPT.

Since partitioning information is stored in the MBR partition table using a beginning block address and a length, it may in theory be possible to define partitions in such a way that the allocated space for a disk with 512-byte sectors gives a total size approaching 4 TiB, if all but one partition are located below the 2 TiB limit and the last one is assigned as starting at or close to block 2<sup>32</sup>–1 and specify the size as up to 2<sup>32</sup>–1, thereby defining a partition that

requires 33 rather than 32 bits for the sector address to be accessed. However, in practice, only certain LBA-48-enabled operating systems, including GNU/Linux, FreeBSD and Windows 7<sup>[19]</sup> that use 64-bit sector addresses internally actually support this. Due to code space constraints and the nature of the MBR partition table to only support 32 bits, boot sectors, even if enabled to support LBA-48 rather than LBA-28, often use 32-bit calculations, unless they are specifically designed to support the full address range of LBA-48 or are intended to run on 64-bit platforms only. Any boot code or operating system using 32-bit sector addresses internally would cause addresses to wrap around accessing this partition and thereby result in serious data corruption over all partitions.

For disks that present a sector size other than 512 bytes, such as [USB external drives](#), there are limitations as well. A sector size of 4096 results in an eight-fold increase in the size of a partition that can be defined using MBR, allowing partitions up to 16 TiB ( $2^{32} \times 4096$  bytes) in size.<sup>[20]</sup> Versions of Windows more recent than Windows XP support the larger sector sizes, as well as Mac OS X, and [Linux](#) has supported larger sector sizes since 2.6.31<sup>[21]</sup> or 2.6.32,<sup>[22]</sup> but issues with boot loaders, partitioning tools and computer BIOS implementations present certain limitations,<sup>[23]</sup> since they are often hard-wired to reserve only 512 bytes for sector buffers, causing memory to become overwritten for larger sector sizes. This may cause unpredictable behaviour as well, and therefore should be avoided when compatibility and standard conformity is an issue.

Where a data storage device has been partitioned with the GPT scheme, the master boot record will still contain a partition table, but its only purpose is to indicate the existence of the GPT and to prevent utility programs that understand only the MBR partition table scheme from creating any partitions in what they would otherwise see as free space on the disk, thereby accidentally erasing the GPT.