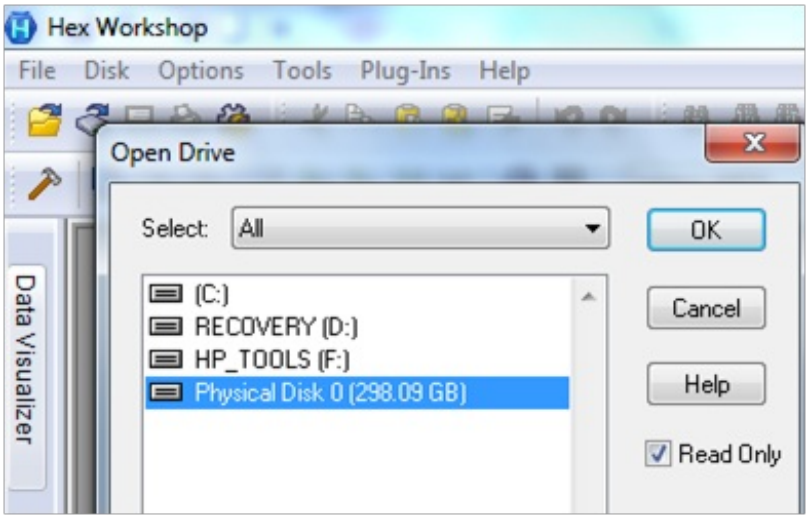
Analysing the Master Boot Record (MBR) with a hex editor (Hex Workshop)

blog.hakzone.info/posts-and-articles/bios/analysing-the-master-boot-record-mbr-with-a-hex-editor-hex-workshop

It is irrelevant which hex editor you use as long as it enables you to access the hard disk to analyse the selected sectors. I will brief the analysis process using Hex Workshop because:

- I think it is ideal to the objective of this article and few other which I am writing.
- It has a very rich set of hexadecimal development tools and you can edit, cut, copy, paste, insert, fill and delete binary data.
- More information (including download links) can be gathered from its official website.

To avoid damaging the system or unintentionally changing data, select the *Read Only* option. Nevertheless, make sure you open the Physical Disk drive rather than a partition to gain access to the whole disk to read areas such as the MBR.



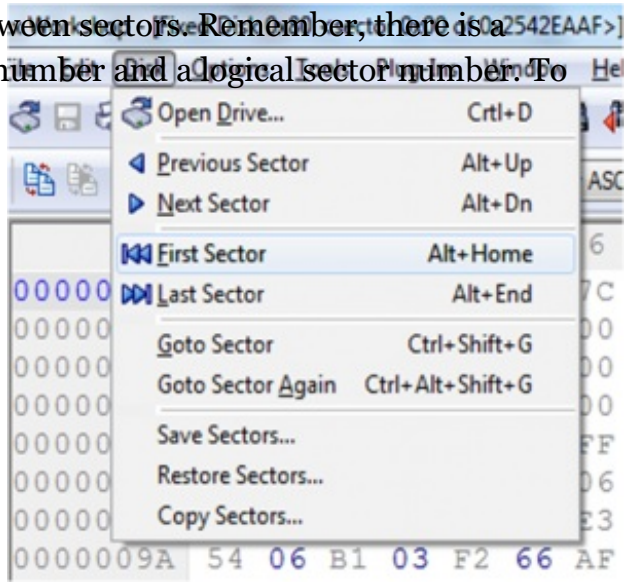
Hex Workshop: open drive in 'Read Only'.

The MBR is usually invetigated because it contains information about the existing partitions in the system. After BIOS decides that no external bootable device (e.g. floppy, CD etc) exist, the control is passed to the MBR.

The MBR location starts with the very first sector of a physical disk. To be more precise, at the physical/absolute sector 0 (0x00).

Absolute vs Relative Sectors

With Hex Workshop you can easily move between sectors. Remember, there is a difference between physical/absolute sector number and a logical sector number. To locate the MBR at the begining of your hard disk, you need to go to the actual first sector of the disk, the absolute sector 0.



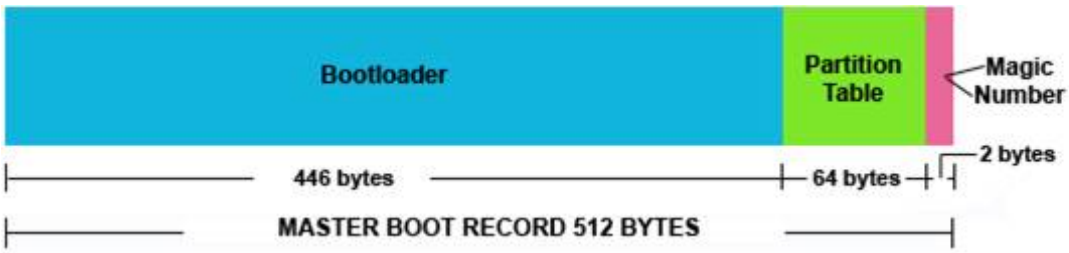
Hex Workshop: locate the first sector.

Relative sector numbers apply when you open a logical drive or partition. In that case, sector 0, 1, 2 etc might actually be sectors 1024, 1025, 1026 etc on the actual disk.

Structure of a generic MBR

The following diagram, illustrates the main areas of a MBR, these are:

1. The Bootstrap Code Area/Bootloader
2. Partition Table
3. Boot Record Signature/Magic Number



MBR: generic structure. Taken from [4]

The following table includes further detail about their location within the sector. This information is critical in order to correctly analyse the MBR.

Structure of a generic MBR			
Offsets within sector		Length (in *bytes*)	Description
Dec	*Hex*		
000 - 445	000 - 1BD	446	Bootstrap Code Area
446 - 509	1BE - 1FD	64	Partition Table
510 - 511	1FE - 1FF	2	Boot Record Signature

MBR structure in further detail

The Bootstrap Code Area

Also called the Master Boot Code or the bootloader area. Bootstrapping is a simple process activating a more complicated system. The code is responsible for 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

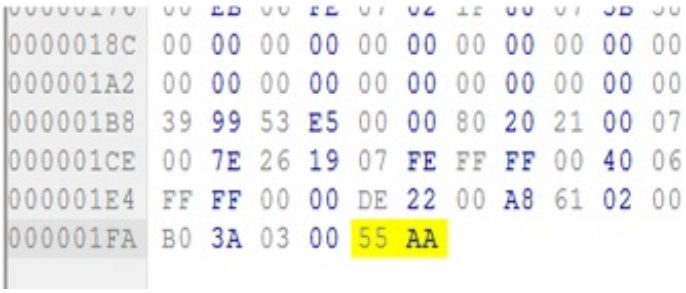
Boot Record Signature

Also refered to as the Magic Number. Is a 2 bytes of code acting as a signature for the MBR. Located at offsets 1FEh and 1FFh and it's values are: 55 AA in hex

To confirm the boot record signature in our system, read 2 bytes starting from offset 1FEh using the hex editor as in the following diagram

Partition Table

To investigate the master partition table, read between offset 1BEh and 1FDh taking the following structure of the generic partition table into consideration.

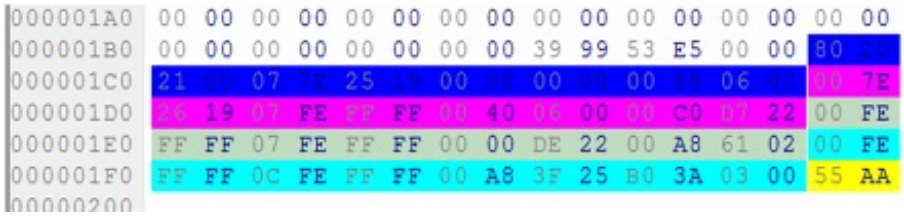


MBR Signature

The *generic* 64-byte *Primary* Partition Table			
Offsets within MBR sector		Length (in *bytes*)	Contents
Dec	*Hex*		
446 – 461	1BE - 1CD	16	Table Entry for Primary Partition # 1
462 – 477	1CE - 1DD	16	Table Entry for Primary Partition # 2
478 – 493	1DE - 1ED	16	Table Entry for Primary Partition # 3
494 - 509	1EE - 1FD	16	Table Entry for Primary Partition # 4

MBR Partition Table

The standard partition table is limited to 4 partitions only. However, the last partition can be used as an extended partition table to include/support further partitioning. To investigate the partitions further, we can highlight them with a background colour for a better view. For instance, I highlighted the bytes between offsets 1BEh and 1CDh in blue, and applied a similar approach for the other three partitions.



MBR – Partition Table Entries

The first entry for partition #1 will be analysed in this article, you can apply the same proceedure to analyse the other 3 entries (partitions). At this point, the structure of the 16-byte partition table entry is need:

Structure of a *16-byte* Partition Table Entry		
Relative Offsets (*within entry*)	Length (*bytes*)	Contents
0	1	Boot Indicator (80h = *active*)
1 - 3	3	Starting CHS values
4	1	*Partition-type* Descriptor
5 - 7	3	Ending CHS values
8 - 11	4	Starting Sector
12 - 15	4	Partition Size (in sectors)

MBR – Partition Table Entry

Based on the structure above, I am now reformatting the hexadecimal values for each of the four entries I have found in the MBR as follows:

- 80 20 21 00 07 7E 25 19 00 08 00 00 00 38 06 00
- 00 7E 26 19 07 FE FF FF 00 40 06 00 00 C0 D7 22
- 00 FE FF FF 07 FE FF FF 00 00 DE 22 00 A8 61 02
- 00 FE FF FF 0C FE FF FF 00 A8 3F 25 B0 3A 03 00

This would help me to distinguish between the different parts. I could now start analysing the first entry step by step

80 20 21 00 07 7E 25 19 00 08 00 00 00 38 06 00

The value 80h indicated an Active Partition which is where the boot flag is set. An active partition indicates to a MS-DOS/MS Windows-type boot loader which partition to boot. In Windows, this is labelled as a SYSTEM partition.

Another value to expect is 00 which is an indication of a non-active partition.

80 20 21 00 07 7E 25 19 00 08 00 00 00 38 06 00

These bytes represent the partition's starting sector in CHS (Cylinder-Head-Sector) values. They read 0, 21, 20 (hex) because they were stored on the disk in little-endian.

80 20 21 00 07 7E 25 19 00 08 00 00 00 38 06 00

This byte represent the partition's file system. 07 is an indication for NTFS.

Information about MBR partition types can be found online:
– http://www.win.tue.nl/~aeb/partitions/partition_types-1.html
– http://en.wikipedia.org/wiki/Partition_type

Some MBR partition types such as 05h and 0Fh will indicate an extended partition. MBR bytes will only tell if an extended partition exist, and its size; Further detail must be extracted from each partition records directly. E.g. the extended partition table in the Extended Boot Records (EBRs).

With more EBRs linked to further EBR tables from its previous link, obtaining the complete layout of any hard disk requires an investigation of the data in the Extended partition tables of each EBR as well as the Master Partition Table!

80 20 21 00 07 7E 25 19 00 08 00 00 00 38 06 00

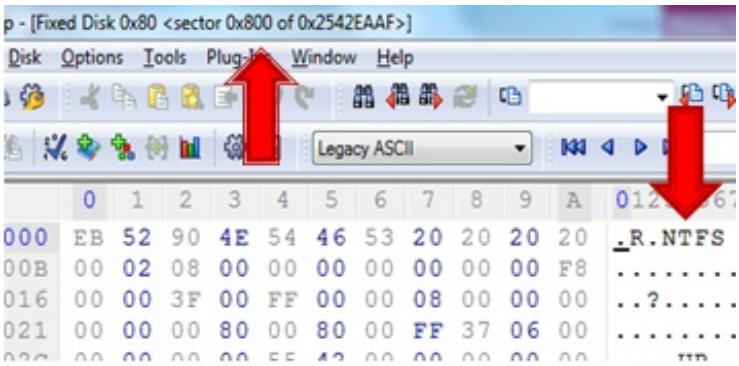
These bytes represent the partition's ending sector in CHS (Cylinder-Head-Sector) values. They read 19, 25, 7E (hex) because they were stored on the disk in little-endian

80 20 21 00 07 7E 25 19 00 08 00 00 00 38 06 00

Starting sector: 00 08 00 00 becomes 00 00 08 00 in hex because it was stored on disk in little-endian, which is 2048 in Decimal.

Using Hex Workshop this can be confirmed, go to sector 800h

80 20 21 00 07 7E 25 19 00 08 00 00 00 38 06 00



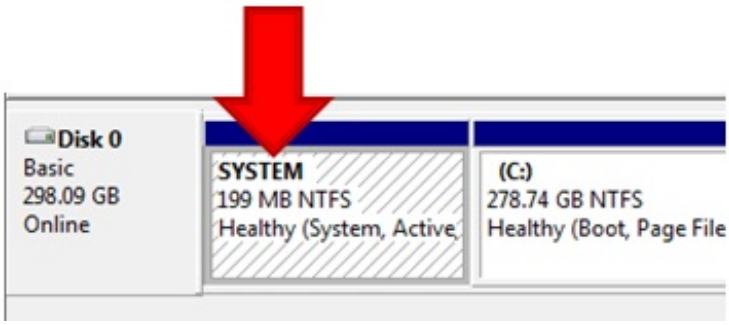
MBR- Sector 800h

The size of the partition: 00 38 06 00 becomes 00 06 38 00 = 407552 sectors (by converting to Decimal) = 208666624 bytes = 199 MiB

For demonstration purposes, the information learned about partition #1 will be compared with that from the *Windows Disk Management*. Information learned so far include:

- Active Partition. In windows this can indicate a system partition.
- NTFS
- start at sector 2048
- Size: 407552 sectors = 199 MiB

Since the partition we covered turned out to be a System partition, I may conclude the article by discussing the differences between system partitions and boot partitions.



Partition Information using Windows Disk Management

As stated by Microsoft tech notes, the system partition hosts the hardware-related files that tell a computer where to look to start Windows while a boot partition directly hosts the Windows operating system files, which are located in the Windows file folder. This is very useful when you have a multiboot computer.