FAT32 Structure Information - MBR, FAT32 Boot Sector Introduction

Summary:

This article introduces detail information about FAT32 file system structure, FAT32 MBR or FAT32 Boot sector etc. If you lost data on FAT32 drive or partition, just free download the best data recovery software here to restore data from FAT32 drive/partition right now.

What is Master Boot Record?

The Master Boot Record is the same for pretty much all Operating Systems. It is located on the first Sector of the Hard Drive, at Cylinder 0, Head 0, Sector 1. It is the first piece of code that your computer runs after it has checked all of your hardware (POST) and turned control of loading software over the hard drive. It also contains the partition table, which defines the different sections of your hard drive. Basically if anything happens to this little 512 byte section, your hard drive is brain dead.

Offset	Description	Size
000h	Executable Code (Boots Computer)	446 Bytes
1BEh	1st Partition Entry (See NextTable)	16 Bytes
1CEh	2nd Partition Entry	16 Bytes
1DEh	3rd Partition Entry	16 Bytes

1EEh	4th Partition Entry	16 Bytes
1FEh	Boot Record Signature (55hAAh)	2 Bytes

- **Partition Entry (Part of MBR)**

Offset	Description	Size
00h	Current State of Partition(00h=Inactive, 80h=Active)	1 Byte
01h	Beginning of Partition - Head	1 Byte
02h	Beginning of Partition - Cylinder/Sector (See Below)	1 Word
04h	Type of Partition (See List Below)	1 Byte
05h	End of Partition - Head	1 Byte
06h	End of Partition - Cylinder/Sector	1 Word
08h	Number of Sectors Betweenthe MBR and the First Sector in the Partition	1 Double Word
0Ch	Number of Sectors in thePartition	1 Double Word

- **Cylinder/Sector Encoding**
  I guess back in the days of 10MB hard drives and 8086's, code was at a premium. So they did everything they could to preserve space. Unfortunately now we have to live with it, but luckily they created new ways of translating the system so the 1024 Cylinder

Limit (2^10) isn't too big of a problem, for newer computers, at least. Older ones usually need some sort of Disk Overlay program to make them see the whole hard drive.

Anyway, to get the Sector out of this, you need to apply an AND mask ($3F) to it. To get the Cylinder, you take the high byte and OR it with the low byte that has been AND masked with ($C0) and then Shifted Left Two. It's not very easy to explain, so I'll just show you how I did it with two routines I made (In Pascal) for Encoding and Decoding the Cylinder/Sector. Hopefully even if you don't know Pascal you'll be able to read it.

```
Function CylSecEncode(Cylinder, Sector: Word): Word;
Begin
   CylSecEncode: = (Lo(Cylinder) shl 8) or (Hi(Cylinder) shl 6) or Sector;
End;
Procedure CylSecDecode(Var Cylinder, Sector: Word; CylSec : Word);
Begin
   Cylinder: = Hi(CylSec) or ((Lo(CylSec) and $C0) shl 2);
   Sector: = (CylSec and $3F);
End;
```

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Cylinder Bits 7to 0								Cylinder Bits 9+8		Sector Bits 5 to0					

Partition Type Listing

There are more than just these shown, but I've only included that ones relevant to MS Operating Systems.

Value	Description
00h	Unknown or Nothing

01h	12-bit FAT
04h	16-bit FAT (Partition Smallerthan 32MB)
05h	Extended MS-DOS Partition
06h	16-bit FAT (Partition Largerthan 32MB)
0Bh	32-bit FAT (Partition Up to2048GB)
0Ch	Same as 0BH, but uses LBA$_1$ 13h Extensions
0Eh	Same as 06H, but uses LBA$_1$ 13h Extensions
0Fh	Same as 05H, but uses LBA$_1$ 13h Extensions

Reading Multiple Partitions

Although having multiple partitions in FAT32 isn't as likely as in FAT16, it still works the same way. The first partition is the Primary Partition, and everything else is stored in the Extended Partition. It's a little tricky when it comes to reading those extra partitions though (not a lot, just a little). The first record in the partition table shows where the Primary partition is (how big it is, where it starts, and where it ends). The second entry in the partition table shows where the Entire Extended Partition is (which may include more than just one partition). To read any more partitions, you go to the where it says the Extended Partition starts, and read the first sector. It acts just like the MBR. It'll have blank where the code is supposed to be, and in the partition table it will have for it's first entry the next Partition in the Drive, and if there are anymore, there will be another Extended partition, just like before. However, all references to Sector Numbers are made using the that new MBR point as the reference, making it a virtual drive. Just incase this doesn't make much sense (and by the way I explain things I can understand if it doesn't), let me show you how a drive with three partitions is setup.

MBR of Whole Drive

Entry #1 - Points to Partition #1
Entry #2 - Points to the Entire Extended Partition
You would read the first sector of that Extended Partition, and see another MBR Structure.

MBR of Extended Partition

Entry #1 - Points to Partition #2
Entry #2 - Points to Rest of Extended Partition after Partition #2
Now, all references to Sector Numbers (most specifically the entry at Offset 08h) in those Entries wouldn't be referenced from the start of the drive, but from the start of the Extended Partition. However, the CHS (Cylinder, Head, Sector) numbers would still be right.

Once again, you would read the first sector of that Extended Partition, and see the next MBR.

MBR of Rest of Extended Partition

Entry #1 - Points to Partition #3
No Entry #2, since this was the Last Partition
If there were another partition, the pattern would continue just like before, until the last one was reached.

FAT32 Boot Record Information

This information is located in the first sector of every partition.

Offset	Description	Size

00h	Jump Code + NOP	3 Bytes
03h	OEM Name (Probably MSWIN4.1)	8 Bytes
0Bh	Bytes Per Sector	1 Word
0Dh	Sectors Per Cluster	1 Byte
0Eh	Reserved Sectors	1 Word
10h	Number of Copies of FAT	1 Byte
11h	Maximum Root DirectoryEntries (N/A for FAT32)	1 Word
13h	Number of Sectors inPartition Smaller than 32MB (N/A for FAT32)	1 Word
15h	Media Descriptor (F8h forHard Disks)	1 Byte
16h	Sectors Per FAT in Older FATSystems (N/A for FAT32)	1 Word
18h	Sectors Per Track	1 Word
1Ah	Number of Heads	1 Word

1Ch	Number of Hidden Sectors inPartition	1 Double Word
20h	Number of Sectors inPartition	1 Double Word
24h	Number of Sectors Per FAT	1 Double Word
28h	Flags (Bits 0-4 IndicateActive FAT Copy) (Bit 7 Indicates whether FAT Mirroringis Enabled or Disabled) (If FATMirroring is Disabled, the FAT Information is onlywritten to the copy indicated by bits 0-4)	1 Word
2Ah	Version of FAT32 Drive (HighByte = Major Version, Low Byte = Minor Version)	1 Word
2Ch	Cluster Number of the Startof the Root Directory	1 Double Word
30h	Sector Number of the FileSystem Information Sector (See Structure Below)(Referenced from the Start of the Partition)	1 Word
32h	Sector Number of the BackupBoot Sector (Referenced from the Start of the Partition)	1 Word
34h	Reserved	12 Bytes
40h	Logical Drive Number ofPartition	1 Byte

41h	Unused (Could be High Byteof Previous Entry)	1 Byte
42h	Extended Signature (29h)	1 Byte
43h	Serial Number of Partition	1 Double Word
47h	Volume Name of Partition	11 Bytes
52h	FAT Name (FAT32)	8 Bytes
5Ah	Executable Code	420 Bytes
1FEh	Boot Record Signature (55hAAh)	2 Bytes

File System Information Sector instruction

Usually, this exists a Second Sector of the partition, although since there is a reference in the Boot Sector to it. I'm assuming it can be moved around. I never got a complete picture of this one. Although I do know where the important fields are at.

Offset	Description	Size
00h	First Signature (52h 52h 61h41h)	1 Double Word

04h	Unknown, Currently (Mightjust be Null)	480 Bytes
1E4h	Signature of FSInfo Sector(72h 72h 41h 61h)	1 Double Word
1E8h	Number of Free Clusters (Setto -1 if Unknown)	1 Double Word
1ECh	Cluster Number of Clusterthat was Most Recently Allocated.	1 Double Word
1F0h	Reserved	12 Bytes
1FCh	Unknown or Null	2 Bytes
1FEh	Boot Record Signature (55hAAh)	2 Bytes

FAT32 Drive Layout

Offset	Description
Start of Partition	Boot Sector
Start + # of ReservedSectors	Fat Tables
Start + # of Reserved + (#of Sectors Per FAT * 2)	Data Area (Starts withCluster #2)

Cluster Meaning

A Cluster is a Group of Sectors on the Hard Drive that have information in them. A 4K Cluster has 8 Sectors in it (512*8=4096). Each Cluster is given a spot in the FAT Table. When you look at an Entry in the FAT, the number there tells you whether or not that cluster has data in it, and if so, if it is the end of the data or there is another cluster after it. All Data on a Partition starts with Cluster #2. If the FAT Entry is 0, then there is no data in that cluster. If the FAT Entry is 0FFFFFFFh, then it is the last entry in the chain.

This is one of my biggest holes in my information. I am unable to find anyplace that shows what numbers mean what when it comes to the FAT table. I was able to tell the end of the chain just by looking at an FAT32 Drive, but I don't know what stands for a BAD Cluster or what the maximum valid number for showing data is.

For now, you can calculate the maximum valid cluster in a partition with this formula:

((# of Sectors in Partition) - (# of Sectors per Fat * 2) - (# of Reserved Sectors)) / (# of Sectors per Cluster)

If there is any remainder in the answer to that formula, it just means that there were a few extra clusters at the end of the partition (probably not enough to make another cluster), so you can just get rid of anything after the decimal point.

Directory Table
Another aspect when looking at a File System at Low Level is the Directory Table. The Directory Table is what stores all of the File and Directory Entries. Basically, there is only one difference between the Directory Table of FAT16 and FAT32. The Difference is: the Reserved OS/2 Byte (Offset 20 [14h]) in the Short Filename Structure is replaced with the High Word of the Cluster Number (since it's now 4 bytes instead of 2).

File Allocation Table
Footnotes

1 - LBA = Logical Block Addressing - Uses the Int 13h Extensions built into newer BIOS's to access data above the 8GB barrier, or to access strickly in LBA mode, instead of CHS (Cylinder, Head, Sector)

Extra tip: restore lost data from FAT32 hard drive with EaseUS hard drive recovery software

If you accidentally deleted, removed or formatted data on an FAT32 hard drive, USB or other storage devices, don't be panic. Your best choice is to let professional hard drive data recovery software for help! Here we'd like to recommend you try EaseUS Data Recovery Wizard which allows you to effectively undelete, unformat and restore all lost data within 3 simple steps. Free download it to get all your lost FAT32 data back now: