

Specifications

FAT32 File System

Usage: Taking over where FAT16 was used

Introduction

This is the 32-bit version of the FAT file system. The 32-bit part describes the way units are allocated on the drive. The FAT32 file system uses a 32-bit number to identify each allocation unit (called cluster), and this gives it a total of 4.294.967.296 clusters. The size of each cluster is defined in the boot sector of the volume (volume = partition).

Boot Sector Structure

The first sector on the volume is the boot sector. It is exactly 512 bytes long and have the following structure.

Structure of the FAT32 Boot sector

Part	Offset	Size	Description
Code	0000h	3 bytes	JMP 0x80h
OS Name	0003h	8 bytes	Oem ID - Name of the formatting OS
	000Bh	2 bytes	Bytes per Sector on the physical medium - Normally 512 bytes
	000Dh	1 bytes	Sectors per Cluster - 1, 2, 4, 8, 16, 32, 64 or 128 sectors
	000Eh	2 bytes	Reserved sectors in front of the FAT(s) incl. the Boot sector
	0010h	1 bytes	Number of FAT copies - Normally 2
	0011h	4 bytes	Not used in FAT32
	0015h	1 bytes	Media Descriptor - The same as in FAT16, but FAT32 is only allowed on harddrives, so the value is F8h
	0016h	2 bytes	Not used in FAT32
BIOS Parameter Block	0018h	2 bytes	Sectors per Track - The disc geometry used when formatting the partition.
	001Ah	2 bytes	Heads - The disc geometry used when formatting the partition.
	001Ch	4 bytes	The number of sectors on the disk from the start of the partition to the beginning of the first FAT.
	0020h	4 bytes	Number of sectors in the partition
	0024h	4 bytes	Sectors per FAT
	0028h	2 bytes	FAT handling flags
	002Ah	2 bytes	FAT32 Drive Version (High byte = Major version, Low byte = Minor version)
	002Ch	4 bytes	Cluster number for the start of the Root Directory Table
	0030h	2 bytes	Sector number from the start of the partition, for the File System Information Sector
	0032h	2 bytes	Sector number from the start of the partition, for the Backup Boot Sector
	0034h	12 bytes	Reserved
	0040h	1 bytes	Logical Drive Number - Normally 00h for floppies and 80h for hard drives.
Ext. BIOS Parameter Block	0041h	1 bytes	Current Head
	0042h	1 bytes	Signature
	0043h	4 bytes	ID - Random generated serial number
	0047h	11 bytes	Volume Label - The same as stored in a special file in the root directory.
	0052h	8 bytes	System ID - This is the string 'FAT32 '
Code	005Ah	420 bytes	Free - Used for executable code - May shrink in the future.
Sig.	01FEh	2	Executable sector signature (AA55h when read into a register)

FAT Handling Flags

This word contain flags which tell how the FAT(s) are supposed to be handled. The M flags enables (when set) or disables (when cleared) the FAT mirroring. When FAT mirroring is enabled, all the FAT copies are updated. When FAT mirroring is disabled, only the FAT copy pointed to by the Active FAT field, is updated.

FAT Handling Flags

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Unused						M	Unused					Active FAT			0000h

Current Head

This value is original used to store the track on which the boot sector is located. But Windows NT uses this byte to store two flags. The lowest bit would indicates that a check disk should be run at boot time, and the second lowest flag indicates that a surface scan should be run.

Signature

This value must be either 28h or 29h in order for Windows NT to recognize the partition. What exactly either of them means I don't know.

Code

This code in the boot sector will be different depending on the system which is intended to be loaded of this FAT16 volume. This means that the code will be different, if DOS, Windows 95, Windows 98, Windows NT or another FAT16 booting operating system has been installed.

File System Information Sector

The FAT(s)

The address of the first FAT is calculated by taking the address of the first sector in the partition + the number of hidden sectors, as defined in the boot sector. The FAT area are filled with 32-bit values which by default tell which cluster is next in chain. There are also some reserved values, such as:

Valid FAT Values

Value	Description
00000000h	Free cluster
000000001h - FFFFFFFF5h	Number of the next cluster
FFFFFFF6h - FFFFFFFF7h	One or more bad sectors in cluster
FFFFFFFh	End-of-file

The FAT area start with a value of FFFFFFFF8h in the first cluster and FFFFFFFFh in the second cluster.

Directory Entries

Each of these entries are 32 bytes long. The root directory can contain the number of entries as defined in the Boot Parameter Block in the boot sector. When creating a sub directory, one cluster will be assigned to it. This sub directory can then be filled with entries, until the cluster is full. If filled beyond the limit then another cluster will be allocated. The number of directory entries which can fit in each cluster, depends on the size of the cluster itself. Hence more space equals more 32 byte entries. This is the format of the directory entries:

Structure of Directory Entries

Offset	Size	Description
00h	8 bytes	Filename
08h	3 bytes	Filename extension
0Bh	1 bytes	Flag byte
0Ch	8 bytes	Unused in FAT32 - But should be left as it was previously read
14h	2 bytes	Starting cluster (High word)
16h	2 bytes	Time
18h	2 bytes	Date
1Ah	2 bytes	Starting cluster (Low word)
1Ch	4 bytes	File size in bytes

Flag Byte

The flag byte defines a set of flags which is set for directories, volume name, hidden files, system files, etc. These are the flags:

Flags in the flag byte

7	6	5	4	3	2	1	0	
Unused	A	D	V	S	H	R		0000h

Achieved Flag

The A flag is set by a backup program, so that the user/program knows which files that has been backed up. This flag is not used correctly by many user and perhaps also by many operating systems.

System

This flag shows that the file/directory is important for the system, and shouldn't be manipulated.

Hidden

This flag tell the system and programs that the file should be hidden for the user. But in a lot of programs this can be overwritten by the user.

Read Only

The flag is used to prevent programs from not automatically overwriting or deleting this file/directory.

Directory

This flag is set, when an entry in the directory table is not pointing to the beginning of a file, but to another directory table. A sub-directory. The sub-directory is placed in the cluster, where the Starting Cluster field points to. The format of this sub-directory table is identical to the root directory table.

Volume Name

When this flag is set, the directory entry is not pointing to a file, but to nothing. The only information used from this entry is the filename (8 bytes) plus the filename extension (3 bytes). These bytes form an 11 bytes long volume label (without any .) There may be only **one** valid entry on the entire disk with this flag set. And preferably this entry should be among the first 3 entries in the root directory table, if not, then MS-DOS can have trouble displaying the right volume label. This volume name should be the same as the one in the boot sector. The latter one is infact rarely used.

Conclusion

The FAT family of file systems are relative simple file systems. The complexity can be enhanced by adding support for long filenames, using the [VFAT Long File Names](#). Also have a look at the [16 bit version](#) of the FAT file system.