

File Allocation Table (FAT)

 networkencyclopedia.com/file-allocation-table-fat

Editor

What is File Allocation Table?

File Allocation Table (FAT) is the name of a computer file system architecture and a family of industry standard file systems utilizing it. The FAT file system is a simple file system originally designed for small disks and simple folder structures.

The FAT file system is named for its method of organization, the file allocation table, which resides at the beginning of the volume. To protect the volume, **two copies** of the table are kept, in case one becomes damaged. In addition, the file allocation tables and the root folder must be stored in a **fixed location** so that the files needed to start the system can be correctly located.

The major versions of the FAT format are named after the number of table element bits: 12 (**FAT12**), 16 (**FAT16**), and 32 (**FAT32**); each of these variants is still in use. The FAT standard has also been expanded in other ways while generally preserving backward compatibility with existing software.

The Windows FAT file system driver is implemented in the

`\Windows\System32\Drivers\Fastfat.sys` file.

How it works

The FAT file system is based on the FAT, a structure that maps the locations of the clusters in which files and folders are stored on the disk. The FAT records the location of each cluster that makes up a given file and the sequence in which it is stored. This is necessary because files are usually not stored in a contiguous location on a hard disk because of the presence of disk fragmentation caused by the creation and deletion of files on the disk.

For each file on a FAT volume, the FAT contains the entry point for the allocation unit in which the first segment of the file is stored, followed by a series of links called the allocation chain. The allocation chain indicates where succeeding segments of the file are located and is then terminated by an end-of-file (EOF) marker.

Two copies of the FAT are kept in fixed locations on the disk to provide redundancy. A disk formatted with the FAT file system is said to be a FAT volume. The sizes of the individual clusters in which file information is stored on a FAT volume depend on the

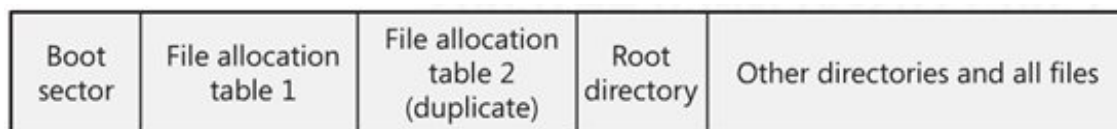
size of the partition or logical drive formatted using FAT, as shown in the following table.

For compatibility reasons, these cluster sizes are the same whether the FAT volume is on an MS-DOS or Windows platform. In the table, you'll see that on small FAT partitions (under 15 MB in size) a special 12-bit FAT file system is used instead of the usual 16-bit FAT. FAT Information for Different Volume Sizes

Drive Size	FAT Type	Sectors/Cluster	Cluster Size
0 MB–15 MB	12-bit	8	4 K
16 MB–127 MB	16-bit	4	2 K
128 MB–255 MB	16-bit	8	4 K
256 MB–511 MB	16-bit	16	8 K
512 MB–1023 MB	16-bit	32	16 K
1024 MB–2047 MB	16-bit	64	32 K
2048 MB–4095 MB	16-bit	128	64 K

Structure of a FAT Volume

The figure below illustrates how the FAT file system organizes a volume.



The file allocation table contains the following types of information about each cluster on the volume

- unused (0x0000);
- cluster in use by a file;
- bad cluster (0xFFF7);
- last cluster in a file (0xFF8-0xFFF for FAT12; 0xFFF8-0xFFFF for FAT16; 0xFFFFF8-0xFFFFFFFF for FAT32).

There is no organization to the FAT folder structure and files are given the **first available location** on the volume.

The entries in the file allocation table define the **file-allocation chains** (shown in **Figure 2**) for files and directories: the links in the chain are indexes to the next cluster of a file's data. A file's directory entry stores the **starting cluster** of the file, the last entry of the file's allocation chain is the reserved value of **0xFFFF** for FAT16

and **0xFFF** for FAT12; the FAT entries for **unused clusters** have a value of **0**. As you can see in **Figure 2**, FILE1 is assigned clusters 2, 3, and 4; FILE2 is fragmented and uses clusters 5, 6, and 8, while FILE3 uses only cluster 7. Reading a file from a FAT volume can involve reading large portions of a file allocation table to traverse the file's allocation chains.

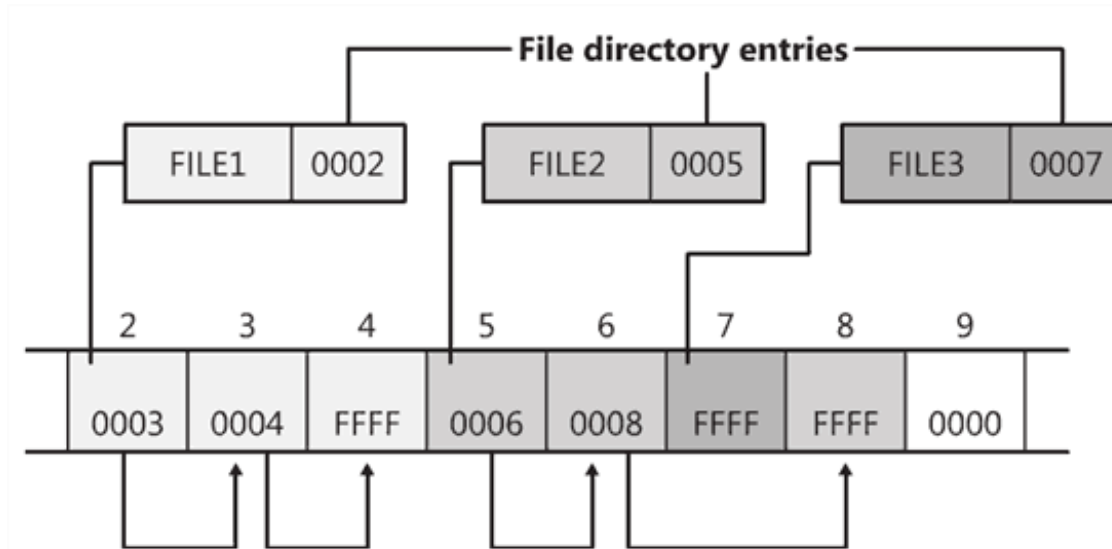


Figure 2: sample of a FAT file-allocation chain.

The root directory of FAT12 and FAT16 volumes is preassigned enough space at the start of a volume to store 256 directory entries, which places an upper limit on the number of files and directories that can be stored in the root directory (there's no preassigned space or size limit on FAT32 root directories).

A FAT directory entry is 32 bytes and stores a file's name, size, starting cluster and time stamp (last-accessed, created and so on) information. If a file has a name that is Unicode or that doesn't follow the MS-DOS 8.3 naming convention, additional directory entries are allocated to store the long file name: the supplementary entries precede the file's main entry. Figure 3 shows a sample directory entry for a file named "The quick brown fox.": the system has created a THEQUI~1 FOX 8.3 representation of the name (that is, you don't see a "." in the directory entry because it is assumed to come after the eighth character) and used two more directory entries to store the Unicode long file name; each row in the figure is made up of 16 bytes.

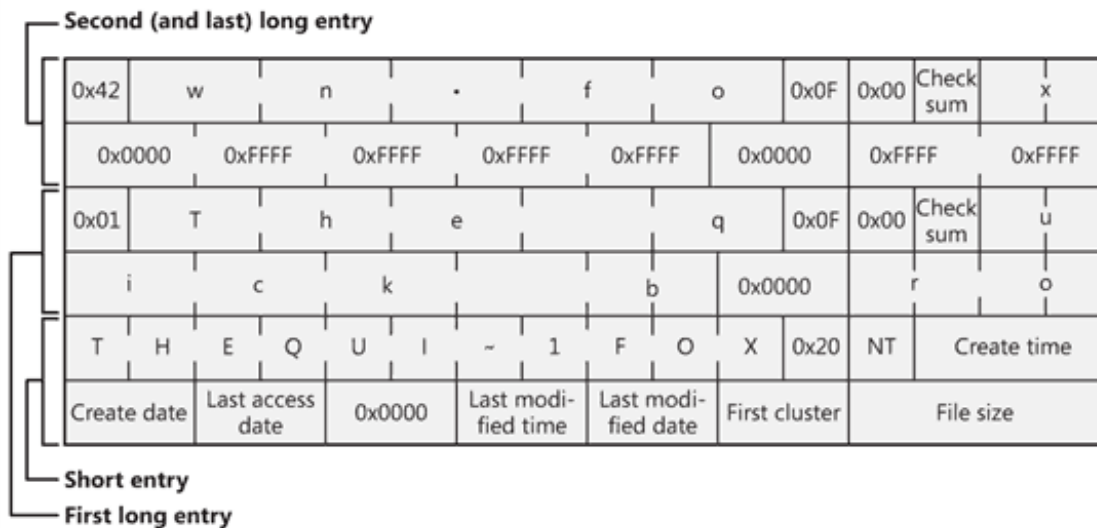


Figure 3: a FAT directory entry.

The Partition Boot Sector

The Partition Boot Sector contains information that the file system uses to access the volume. On x86-based computers, the Master Boot Record use the Partition Boot Sector on the system partition to load the operating system kernel files. Table 1 describes the fields in the Partition Boot Sector for a volume formatted with the FAT file system.

Byte Offset	Field Length	Sample Value	Meaning
00	3 bytes	EB 3C 90	Jump instruction
03	8 bytes	MSDOS5.0	OEM Name in text
0B	25 bytes		BIOS Parameter Block
24	26 bytes		Extended BIOS Parameter Block
3E	448 bytes		Bootstrap code
1FE	2 bytes	0x55AA	End of sector marker

Table 1: fields in the Partition Boot Sector.

FAT Root Folder

The root folder contains an entry for each file and folder in the root. The only difference between the root folder and other folders is that the root folder is on a specified location on the disk and has a fixed size (512 entries for a hard disk, number of entries on a floppy disk depends on the size of the disk).

Long File Names

One of the user experience goals for the designers of Windows 95 was the ability to use **long filenames** (LFNs, up to 255 UTF-16 code points long), in addition to classic 8.3 filenames (Short File Name, SFNs). For backward compatibility, LFNs were implemented as **an optional extension on top of the existing** FAT file system structures using a workaround in the way directory entries are laid out.

This transparent method to store long file names in the existing FAT file systems without altering their data structures is usually known as **VFAT** (for “Virtual FAT”), after the Windows 95 virtual device driver. In **Windows NT**, support for VFAT long filenames started **from version 3.5**. Non VFAT-enabled operating systems can still access the files under their short file name alias without restrictions; however, the associated long file names may get lost, when files with long file names are copied under non VFAT-aware operating systems.

Maximum Sizes on FAT Volumes

Before you format a volume, you can evaluate the types of files to be stored on the volume so that you can determine whether to use the default cluster size. If you format a volume but do not specify a cluster size, default values are used. If you want to change the cluster size after the volume is formatted, you must reformat the volume.

FAT16 and FAT32 have the following size limitations:

- FAT volumes smaller than 16 MB are formatted as FAT12.
- FAT16 volumes larger than 2 GB are not accessible from computers running MS-DOS, Windows 95, Windows 98, Windows Millennium Edition (Me), and many other operating systems. This limitation occurs because these operating systems do not support cluster sizes larger than 32 KB, which results in the 2 GB limit.
- In theory, FAT32 volumes can be about 8 terabytes; however, the maximum FAT32 volume size that Windows Server 2003 can format is 32 GB. Therefore, you must use NTFS to format volumes larger than 32 GB. However, Windows Server 2003 can read and write to larger FAT32 volumes formatted by other operating systems.

The largest possible file for a FAT32 volume is 4 GB minus 1 byte. FAT32 contains 4 bytes per cluster in the file allocation table; FAT16 contains 2 bytes per cluster; and FAT12 contains 1.5 bytes per cluster. A FAT32 volume must have at least 65,527 clusters

FAT16 supports a maximum of 65,524 clusters per volume. The following table lists FAT16 size limits.

FAT16 Size Limits

Description	Limit
Maximum file size	Tested: 4 GB minus 1 byte (232 bytes minus 1 byte)
Maximum volume size	Tested: 4 GB
Files per volume	Approximately 65,536 (216 files)
Maximum number of files and folders within the root folder	512 (Long file names can reduce the number of available files and folders in the root folder.)

A FAT32 volume must have a minimum of 65,527 clusters. Windows Server 2003 can format FAT32 volumes up to 32 GB, but it can mount larger FAT32 volumes created by other operating systems. The following table lists FAT32 size limits.

FAT32 Size Limits

Description	Limit
Maximum file size	Tested: 4 GB minus 1 byte (232 bytes minus 1 byte)
Maximum volume size	Tested: 32 GB (implementation)
Files per volume	4,177,920
Maximum number of files and subfolders within a single folder	65,534 (The use of long file names can significantly reduce the number of available files and subfolders within a folder.)