Cat32 File System

Table of Contents

File System Structure

The overall structure of the file system is described by the following table.

Name	Offset	Size	Description
Boot Sector	0	Sector Size	Boot strap code.
File System Definition	Sector Size	Sector Size	Defines the characteristics of the file system.
Cluster Allocation Table	Sector Size * 2	?	Defines the usage of clusters.
Data Area	After the CAT	Rest of FS	Holds the file and directory data.

Boot Sector

The boot sector is the very first sector of the file system. It is exactly one sector in size. The boot sector is only present when the file system is located at the very first address of the media. If the file system is not located at the very first address of the media, the boot sector portion of the file system will not exist and the file system definition will be the first sector of the file system.

File System Definition

The file system definition contains the parameters of the file system. It contains necessary information about the file system such as sector count, cluster size, etc. If the boot sector exists in the file system than the file system definition will occupy the second sector of the file system; otherwise, the file system definition will occupy the very first sector of the file system.

Cluster Allocation Table

The cluster allocation table describes how the clusters in the file system's data area are assigned and used. The cluster allocation table starts at the sector directly after the file system definition and is as long as needed to account for all of the clusters in the data area.

For every cluster in the data area there exists a corresponding four byte entry in the cluster allocation table (the cluster allocation table is described in more detail below). Because of this, it is possible that the last sector in of the cluster allocation table will not be completely full of four byte entries. In this case, the excess space in that sector is considered reserved and should not be used for any purpose.

Data Area

The data area is where all of the file content and directory data is stored. The data area begins directly after the cluster allocation table. The size of the data area is dependent on the number of cluster allocation table entries there are. The actual size of the data area is the size of a cluster multiplied by the number of cluster allocation table entries.

File System Definition

The file system definition contains the parameters of the file system along with other information pertaining to the file system. All addresses within the file system definition that are not presented in the following table are considered reserved and undefined.

Name	Offset	Size	Description
Sector Size in Bytes	0 – 3	4	Size of a single sector on the disk.
Cluster Size in Bytes	4 – 7	4	A power of two greater than or equal to Sector Size In Bytes.
Cluster Size in Sectors	8 – 11	4	The number of sectors that make up a single cluster.
Sector Count	12 – 15	4	Total number of sectors being used by the file system.
Reserved	16 – 19	4	
Cluster Count	20 – 23	4	The number of clusters used in the file system and the number of entries in the CAT.
Reserved	24 – 27	4	
CAT Size in Exact Bytes	28 – 31	4	The exact size of the cluster allocation table in bytes.
Reserved	32 – 35	4	
CAT Size in Bytes	36 – 39	4	The size of the cluster allocation table in bytes rounded up.
Reserved	40 – 43	4	
CAT Size in Sectors	44 – 47	4	(CAT Size In Bytes) / (Sector Size In Bytes) rounded up.
Reserved	48 – 51	4	
Data Area Offset in Bytes	52 – 55	4	Offset in bytes to the first cluster.
Reserved	56 – 59	4	
Data Area Offset in Sectors	60 – 63	4	Offset in sectors to the first cluster.
Reserved	64 – 67	4	
Data Area Size in Bytes	68 – 71	4	The size of the data area in bytes.
Reserved	72 – 75	4	
Data Area Size in Sectors	76 – 79	4	The size of the data area in sectors.
Reserved	80 – 83	4	

All offsets are inclusive.

Sector Size in Bytes

The sector size in bytes defines the size, in bytes, of each sector on the disk. Generally the sector size will always match the native sector size of the media the file system is on. Common native sector sizes are 512 bytes or 2048 bytes. The minimum sector size supported by the file system is 512 bytes. If the native sector size is smaller than 512 bytes, it is still possible to use the file system on that media. In this case several native sectors will have to be combined to make up one of the accepted sector sizes by the file system.

Cluster Size in Bytes

The cluster size in bytes defines the number of bytes that makes up each cluster. The cluster size has to be greater than or equal to the sector size. The cluster size must also be a multiple of sector size.

Cluster Size in Sectors

This is the same as the cluster size in bytes but measured in sectors.

Sector Count

This is the total number of sectors being used in the file system. This should be the count of sectors starting from the boot sector (or file system definition if no boot sector exists) to the very end of the data area.

Cluster Count

This is the total number of clusters that make up the data area. This also corresponds to the number of cluster allocation entries.

CAT Size in Exact Bytes

This is the exact size of the cluster allocation table in bytes. If the last sector of the cluster allocation table has reserved space at the end of it (i.e. there was not enough entries to fill up the entire last sector) than that blank space is not counted in this measure.

CAT Size in Bytes

This is the size of the cluster allocation table in bytes. This measure includes any reserved space at the end of the cluster allocation table.

CAT Size in Sectors

This is the same as the CAT size in bytes but measured in sectors.

Data Area Offset in Bytes

This is the offset, in bytes, to the beginning of the data area.

Data Area Offset in Sectors

This is the offset, in sectors, to the beginning of the data area.

Data Area Size in Bytes

This is the total size of the data area measured in bytes.

Data Area Size in Sectors

This is the total size of the data area measured in sectors.

Cluster Allocation Table

The cluster allocation table is what defines how the data portion of the file system is being used. The cluster allocation contains 32-bit unsigned integer values. Each value is a number that either contains a special value or points to another CAT entry. The possible values and their meanings are described in the table below.

Value	Meaning
0000`0000	Reserved (Root Directory)
0000`0001 – FFFF`FFEF	Next Cluster in Chain
FFFF`FFF0 – FFFF`FFFB	Reserved
FFFF`FFFC	Reserved Cluster (Unused Cluster)
FFFF`FFFD	Bad Cluster
FFFF`FFFE	End of Cluster Chain
FFFF`FFFF	Free Cluster

All values are in hex and use inclusive ranges.

Upon initialization of the file system all CAT entries must be set to FFFF`FFFF except for the first CAT entry which must be set to FFFF`FFFFE. The root directory always starts in the first cluster and therefore the first cluster is considered to be reserved and should never be used for any purpose other than for the root directory contents.

CAT entries can create a chain of clusters that represent the data of a file or a list of clusters that hold directory entries for a directory. For example, let's take a file whose data occupies three clusters. The starting cluster is cluster 13, the next cluster is cluster 16, and the last cluster is cluster 17. This file also has three CAT entries that correspond to the clusters the file occupies. The starting CAT entry is entry number 13, the next entry is entry number 16, and the last entry is entry number 17. CAT entry 13 (the starting entry) will contain the value 16 which means the next cluster in the file is cluster 16. We then look at CAT entry 16 and we see that entry contains the value 17. So we move to CAT entry 17 which contains the value 0xFFFF`FFFE which means that cluster 17 is the last cluster to the file. This same principal goes for directories as well. However, filename clusters can never be chained.

File System Data Area

The file system data area is the area in the file system where the file, directory data and filename data is stored. The file system data area is split up into clusters.

Directory Entries

A directory entry is a structure in the file system that defines either a subdirectory or a file. The basic structure is described in the table below.

Name	Offset	Size	Description
Attributes	0	1	The attributes of the directory or file.
Reserved	1 – 7	7	
First Cluster	8 – 11	4	The first cluster of this entry's data.
Reserved	12 – 15	4	
File Size	16 – 19	4	Exact size of the file in bytes.
Reserved	20 – 23	4	
Filename Cluster	24 – 27	4	Cluster which contains the filename.
Reserved	28 – 31	4	
Creation Time	32 – 39	8	Time of the creation of the file or directory.
Modified Time	40 – 47	8	Time of when this file or directory was last modified.
Accessed Time	48 – 55	8	Time of when this file or directory was last accessed or used.
Reserved	56 – 63	8	

All offsets are inclusive.

Attributes

The attributes field holds bit values that specify attributes and properties for the directory or file.

Bit 0 – Entry Exists
Set if the directory entry exists, otherwise it is cleared.

Bit 1 – Is File
Set if the directory entry points to a file, otherwise it is cleared.

Bit 2 – Hidden
Set if this file or directory should be considered hidden to the user by default, otherwise it is cleared. Note that this is in no way enforced.

Bit 3 – Read Only
Set if this file or directory should be considered read only, otherwise it is cleared. Note that this is in no way enforced.

Bit 4 – System
Set if this file or directory should be considered a system file or system directory. Note that this is in no way enforced.

Bits 5 – 7
Reserved.

First Cluster
This field points to the first cluster of the directory entry's data. If the directory entry points to a file then this field points to the first cluster of the file's data. If the directory entry points to a sub directory then this field points to the first cluster that holds the sub directory's entries.

File Size
The file size field gives the exact size of the file in bytes. This field is useful when a file's real size is not a multiple of the cluster size. For example, if the cluster size is 512 bytes and the exact size of the file is only 200 bytes, then the file size is required so you know to load only the first 200 bytes of the first cluster.

Filename Cluster
This field points to the cluster that contains the filename. The filename its self has a maximum length of 255 characters which has to be terminated by a null character. The maximum size of the filename is 256 bytes counting the null terminating character. The filename has to begin at the very first byte of the cluster and must be contiguous. The cluster size has to be greater than or equal to 256 bytes as this insures that all filenames will always fit into a single cluster. If the cluster size is greater than 256 bytes, the extra space at the end of the cluster cannot be reused for any purpose. It must remain unused and undefined. Filename clusters can never be chained.

Creation Time
The creation time field represents the original creation time of the file or directory. If a file or directory is copied, this field should not be changed.

Modified Time

The modified time field represents the date and time of when the file or directory was last modified. If a file is changed in any way, resaved, or copied then this field should reflect the time of the operation. This field should not be modified if the file or directory is moved. Upon setting the creation time field, this field should be set to the same value as the creation time field.

Accessed Time

The accessed time field represents the date and time of when the file or directory was last accessed. Upon setting the creation time field or the modified time field, the accessed time field should be set to the value of the field that was set.

Storage of Directory Entries

The root directory entry is implicit and does not actually exist as a physical directory entry. The implicit root directory entry stores its directory entries (the files and directories that are inside the root directory) starting in the first cluster (cluster 0). The root directory is allows to expand as needed by creating a chain of clusters.

When a directory (such as the root directory) contains files and sub directories, the First Cluster field of the directory entry will point to the cluster which holds the first files and directories. A cluster may contain one or more directory entries. If the cluster size is 512 bytes than that means 8 directory entries can fit inside one single cluster (a directory entry is 64 bytes big). When a cluster can no longer contain any more directory entries that cluster will become a chain and the CAT entry for that cluster will contain the value of the next cluster in the chain.

When a file or directory is deleted, it is possible that its directory entry exists in the middle of the cluster. In this case the directory entry is cleared to zero. Note that it is not actually required for the entire entry to be cleared; only the Exists bit in the directory entry's attribute field has to be set to zero. Not clearing the entire entry could allow for possible file recovery. The Exists bit is used to determine if a directory entry exists in a specific spot in a cluster. When creating a file in a directory, every directory entry for that directory should be checked. The file should be created using the first directory entry whose Exists bit is set to zero; however, this is not a requirement.

File Times

File times (creation, modified and accessed) are stored using eight byte integers. The integer is unsigned and stores the amount of time since 00:00 0001/1/1 in 100 nanosecond increments called ticks. The two most significant bits of the unsigned integer are unused. The last valid date is 23:59:59 9999/12/31. The exact maximum amount of valid ticks that can be stored is 3,153,599,999,999,999,999. Leap year is not accounted for.

Filename Rules

Filenames can be as long as 255 characters long and as small as a single character. The filenames . and .. are not allowed. Every filename must end with a null terminated character (ASCII and UTF-8 code 0). With this null terminated character appended to the end of the filename, the file name can effectively be up to 256 characters. Any spare data after the null terminated character is considered unused and should not be used for any specific purpose. This spare data is considered undefined. All filenames are case sensitive. Furthermore, filenames can be a maximum of 512 bytes including the null terminating character. Filenames are encoded as UTF-8 characters. This means that if the filename is encoded in 16-bit characters, the full 256 character limit can be used in the filename. However, if 32-bit characters are used, only 128 characters of the limit can be used. The only characters that cannot be used in the filename are the null character (ASCII and UTF-8 code 0) and the forward slash character (ASCII and UTF-8 code 47). The same rules apply for directory names.

Endianness

The endianness of the file system does not matter as long as the entire file system is consistent. It is assumed that the OS using the file system knows, or can figure out, what endian order the file system is using.