File System

FAT

Structure of a FAT Volume

MBR and VBR (PBR)

- Only one MBR exists on a physical disk
  - First physical sector of the disk
- Each partition/volume has a VBR

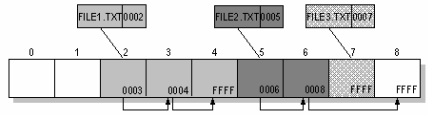
FAT Partition Boot Sector

Byte Offset (in hex)	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

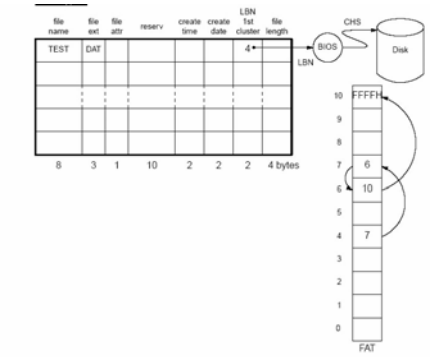
FAT 12, 16, 32

- What the number mean?

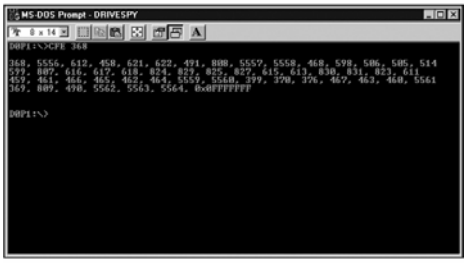
File Allocation Table



Example



DriveSpy



DriveSpy CFE command

Cluster

Sectors and Bytes Per Cluster

Drive size	Number of sectors	FAT16	FAT32
256-511 MB	16	8 KB	4 KB
512 MB-1 GB	32	16 KB	4 KB
1-2 GB	64	32 KB	4 KB
2-8 GB	8	N/A	4 KB
8-16 GB	16	N/A	8 KB
16-32 GB	32	N/A	16 KB
More than 32 GB	64	N/A	32 KB

FAT Root Folder

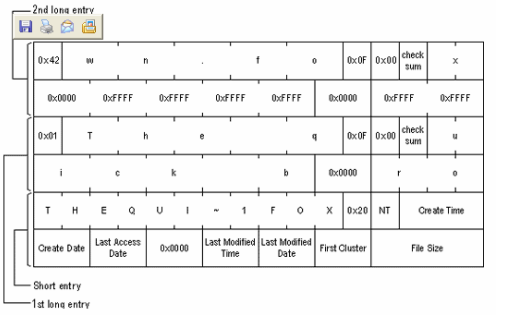
FAT Folder Structure

- 32-byte folder entries for each file and subfolder
  - Name (8+3)
  - Attribute
  - Create time
  - Create date
  - Last access date
  - Last modified time
  - Starting cluster number in the file allocation table
  - File size

Supports for Long Filenames

- Begins with VFAT
- Secondary folder entries
  - 13 characters
  - Unicode
  - Volume, read-only, system, and hidden file attributes bit set

Example



FAT32 Features

- Supports drives over 2GB
- Use smaller clusters than on large FAT16 drives
- Four bytes per cluster in file allocation table
  - Only 28 bits are used

FAT32 Features (cont'd)

- More reserved sectors
- Boot sector modification
  - Needs more than one sector
  - Count of free clusters
  - The cluster number of the most recently allocated cluster
- Root directory
- Sectors per FAT

FAT32 Features (cont'd)

- FAT Mirroring
  - Enable
  - Disable

FAT32 Partition Types

Value	Description
PART_UNKNOWN (00h)	Unknown
PART_DOS2_FAT (01h)	12-bit FAT
PART_DOS3_FAT (04h)	16-bit FAT. Partitions smaller than 32MB.
PART_EXTENDED (05h)	Extended MS-DOS Partition
PART_DOS4_FAT (06h)	16-bit FAT. Partitions larger than or equal to 32MB.
PART_DOS32 (0Bh)	32-bit FAT. Partitions up to 2047GB.
PART_DOS32X (0Ch)	Same as PART_DOS32 (0Bh), but uses Logical Block Address Int 13h extensions.
PART_DOSX13 (0Eh)	Same as PART_DOS4_FAT (06h), but uses Logical Block Address Int 13h extensions.
PART_DOSX13X (0Fh)	Same as PART_EXTENDED (05h), but uses Logical Block Address Int 13h extensions.

What happens when files are deleted?

Slack Space

- What is file slack?
  - The area of the disk cluster between the end of the file and the end of the cluster
- Two types of slack space
  - RAM Slack
  - Drive Slack

RAM Slack

- The space between the end of the file and the end of that sector.

Drive Slack

- Additional sectors maybe needed to round out the block size in the last cluster
- Drive slack is padded with what was stored on the storage device before

Example

- Hello++++++++++++++++++++|----------(EOC)

NTFS

Default Cluster Sizes

Volume size	FAT16 cluster size	FAT32 cluster size	NTFS cluster size
7 MB–16 MB	2 KB	Not supported	512 bytes
17 MB–32 MB	512 bytes	Not supported	512 bytes
33 MB–64 MB	1 KB	512 bytes	512 bytes
65 MB–128 MB	2 KB	1 KB	512 bytes
129 MB–256 MB	4 KB	2 KB	512 bytes
257 MB–512 MB	8 KB	4 KB	512 bytes
513 MB–1,024 MB	16 KB	4 KB	1 KB
1,025 MB–2 GB	32 KB	4 KB	2 KB
2 GB–4 GB	64 KB	4 KB	4 KB
4 GB–8 GB	Not supported	4 KB	4 KB
8 GB–16 GB	Not supported	8 KB	4 KB
16 GB–32 GB	Not supported	16 KB	4 KB
32 GB–2 TB	Not supported	Not supported	4 KB

NTFS

NTFS

- **Partition Boot Sector**
  - The first data set of an NTFS disk
  - 7 sectors long
  - can be expanded up to 16 sectors.

Partition Boot sector

Byte Offset	Field Length	Field Name
0x00	3 bytes	Jump Instruction
0x03	LONGLONG	OEM ID
0x0B	25 bytes	BPB
0x24	48 bytes	Extended BPB
0x54	426 bytes	Bootstrap Code
0x01FE	WORD	End of Sector Marker

Master File Table

- **Master File Table –** Used by NTFS to track files. It contains information about the access rights, date and time stamps, system attributes, and parts of the file. It is also the first file on the NTFS volume

NTFS

- Everything on the volume is a file
- Everything in a file is an attribute
  - Filename attribute
  - Security attribute
  - Data attribute

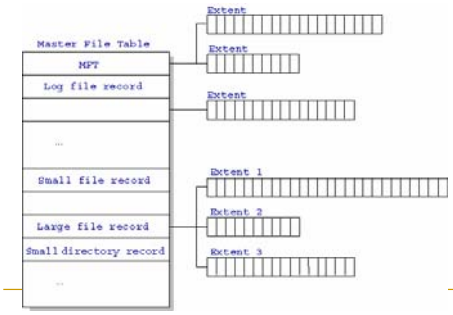
NTFS

- Master File Table
  - All components are files
    - Master File Table
    - Data files
    - Directories
    - Free list (bit map)
    - Boot images
    - Recovery logs

Master File Table

- All files stored on disk are described by the MFT
- All files are logically stored in the MFT
- A table with one row per file

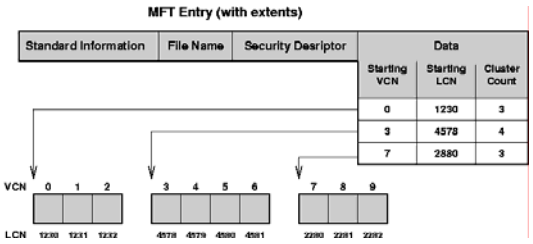
Master File Table



MFT Record for a Small File or Directory

Standart information	File or directory name	Security descriptor	Data or index	

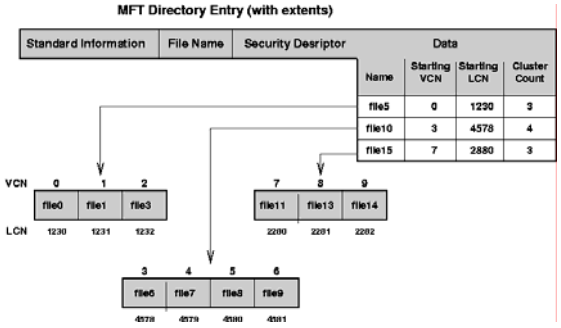
MFT Entry with Extent



MFT Directory Entry

MFT Directory Entry (Everything Fits)

Standard Information	File Name	Security Desriptor	Index file5	file10	file15



NTFS File Attributes

- Resident attributes
  - File name
  - Time stamp
- Nonresident attributes

NTFS File Attributes

Attribute type	Purpose
Standard information	Time stamp data and link (inode) count information is listed here.
Attribute list	Attributes that do not fit within the MFT are listed here. This lists the location of the nonresident attributes.
Filename	The long and short name for the file is contained here. Up to 255 Unicode bytes are available for long file names. For POSIX requirements, additional names or hard links can also be listed here.
Security descriptor	Ownership and who has access rights to the file or folder is listed here.
Data	File data is stored here. Multiple data attributes are allowed for each file. When more space is needed for additional data an inode is assigned linking to a new MFT attribute record.
Object ID	The volume-unique file identifier is listed here. Not all files will need this unique identifier.
Logged tool stream	This field is used by the encrypted file system service that was implemented in Windows 2000 and XP.
Reparse point	This is used for volume mount points and for installable file system (IFS) filter drivers. For the IFS it marks specific files that are used by drivers.
Index root	Implemented for use of folders and indexes.
Index allocation	Implemented for use of folders and indexes.
Bitmap	Implemented for use of folders and indexes.
Volume information	Used by the $Volume system file. The volume version number is listed here.

NTFS File Attributes

Attribute type	Purpose
Volume name	Used by the $Volume system file. The volume version label is listed here.

NTFS System Files

Filename	System file	Record position	Description
$Mft	MFT	0	Base file record for each folder on the NTFS volume. Other record positions within the MFT will be allocated if more space is needed.
$MftMirr	MFT 2	1	The first four records of the MFT are saved in this position. If a single sector fails in the first MFT, the records can be restored allowing for recovery of the MFT.
$LogFile	Log file	2	Previous transactions are stored here to allow for recovery after a system failure has occurred in the NTFS volume.
$Volume	Volume	3	Information specific to the volume such as label and version is stored here.
$AttrDef	Attribute definitions	4	A table listing the attribute names, numbers, and definitions.

NTFS Multiple Data Stream

- Ways in which data can be appended to a file intentionally or not.
- Stream name identifies a new data attribute on the file

NTFS Compressed Files

What's new in NTFS5

- Encryption
- Disk Quotas
- Reparse Points
- Volume Mount Points
- Sparse Files
- Distributed Link Tracing

EFS

Encrypted File System (EFS) –
Symmetric key encryption first used in Windows 2000 on NTFS formatted disks.

Keep files safe from intruders

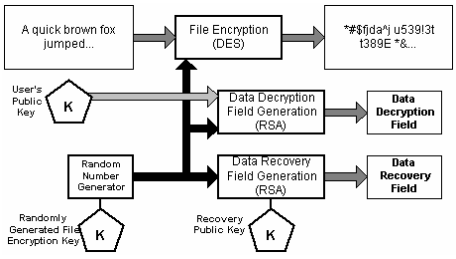
EFS

- Benefits over 3rd party encrypting application
  - Transparent to user and any applications
  - Strong key security
  - All encrypting/decrypting are performed in kernel mode
  - Data recovery mechanism

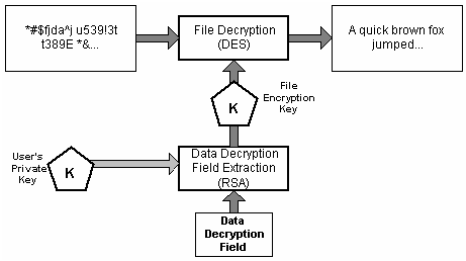
EFS Internals

- Symmetric key encryption in combination with public key technology
- File Encryption Key (FEK)
- Data Decryption Field (DDF)
  - FEK encrypted with public key
- Data Recovery Field (DRF)
  - FEK encrypted with public key of the recover agent

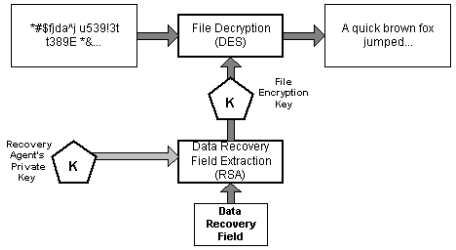
EFS Encryption



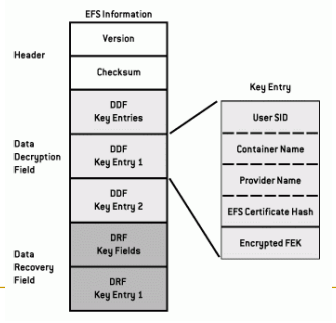
EFS Decryption



EFS Recovery



EFS-Encrypted Files and Folders



Issues with EFS

- Temporary file is “deleted”, but not “erased”
- File names in encrypted folder are not protected
- Key security

Data Integrity and Recoverability with NTFS

- Transaction based

NTFS Sparse Files