Exploring Microsoft File Structures

- Need to understand Microsoft file structures
  - Clusters
  - File Allocation Table (FAT)
  - New Technology File System (NTFS)
- The method an OS uses to store files determines where data can be **hidden**

Exploring Microsoft File Structures

- In Microsoft file structures
  - Sectors are grouped to form **clusters**
  - Clusters are typically 512, 1024, 2048, 4096, or more bytes each
  - Combining sectors **minimizes the overhead** of writing or reading files to a disk
  - Clusters are numbered sequentially starting at 2
  - First sector of all disks contains a system area, the boot record, and a file structure database

Exploring Microsoft File Structures

- OS assigns these cluster numbers, called **logical addresses**
- Sector numbers are called **physical addresses**
- Clusters and their addresses are specific to a logical disk drive, which is a disk partition

Disk Partitions

- Hard disks are partitioned, or divided, into two or more sections
- A **partition** is a logical drive
  - Large disks have to be partitioned
  - FAT16 does not recognize disks larger than 2 GB
- Hidden partitions or voids
  - To hide data on a hard disk
  - Large unused gaps between partitions on a disk **- Partition gap**
  - can be created between the primary partition and the first logical partition
  - Another technique **-** hide incriminating digital evidence at the end of a disk

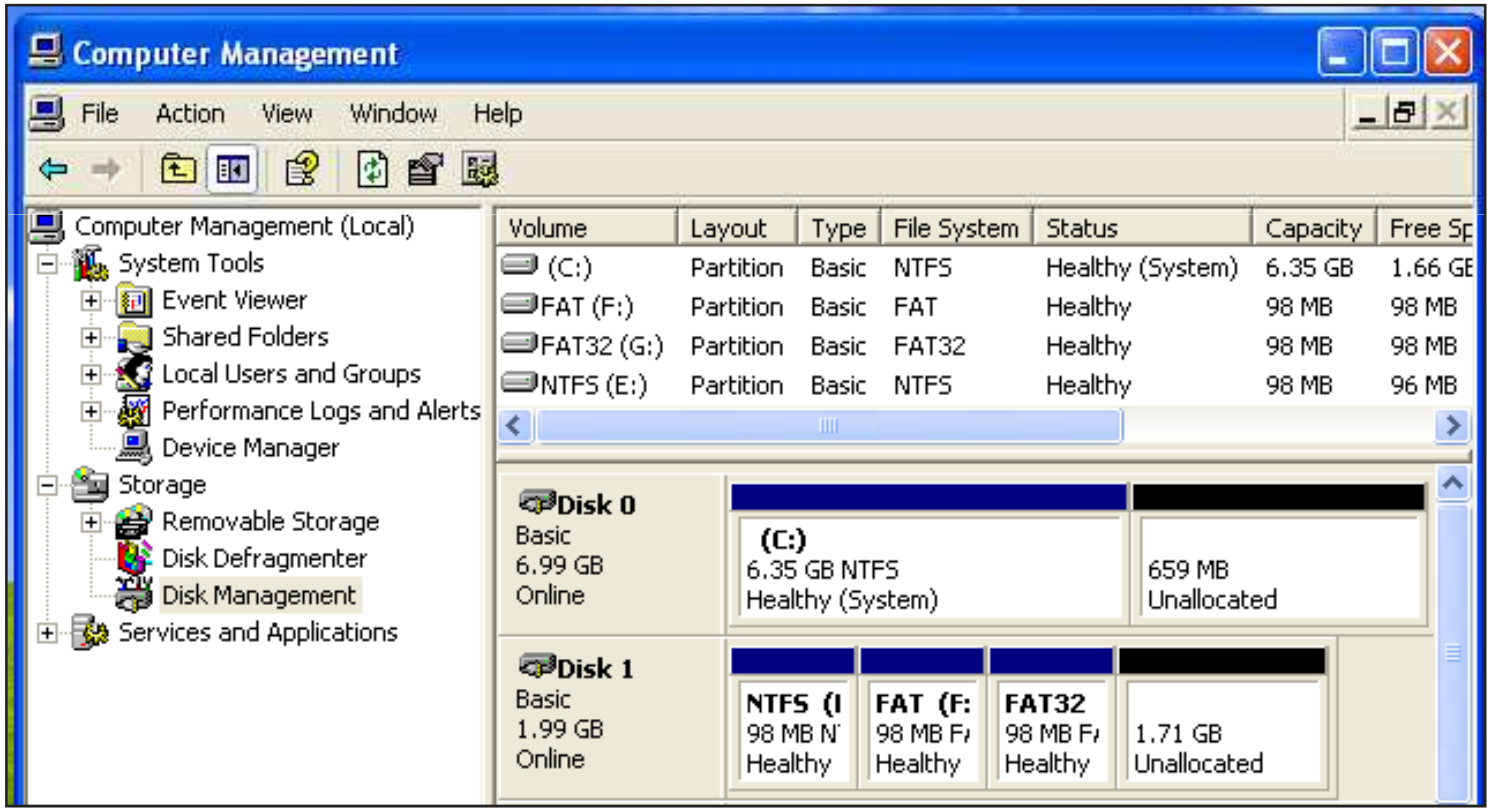
Disk Partitions

- Disk-editing tools
  - access these hidden areas
  - identify the OS on an unknown disk
  - view file headers and other critical parts of a file
    - Both tasks involve analyzing the key hexadecimal codes the OS uses to identify and maintain the file system
  - alter information in partition table - to hide a partition
  - Norton Disk- Edit, WinHex, or Hex Workshop

Partitions

- Partition Types
  - NTFS: 07
  - FAT: 06
  - FAT32: 0B

- **Windows** + R
- diskmgmt.msc

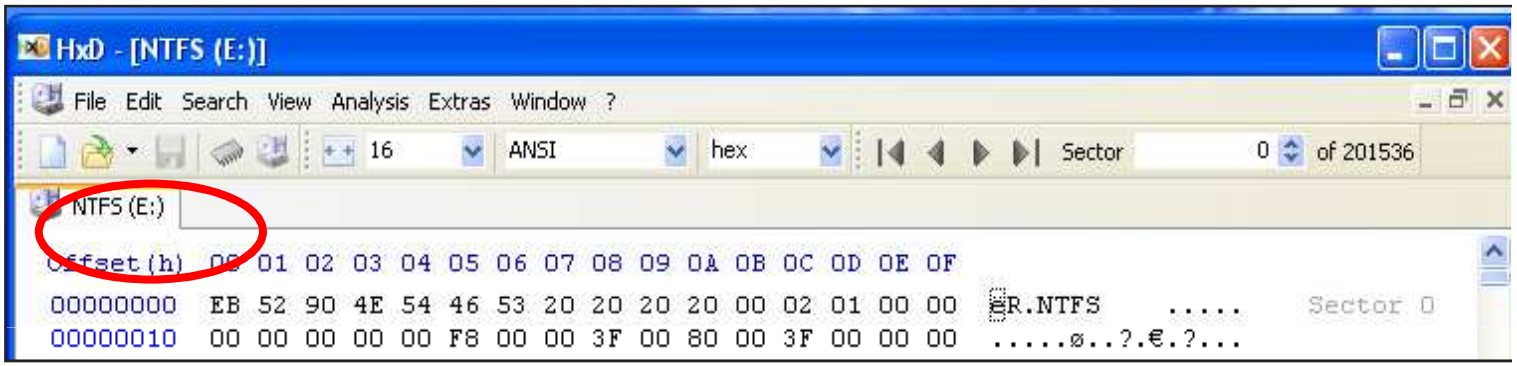


Master Boot Record Structure

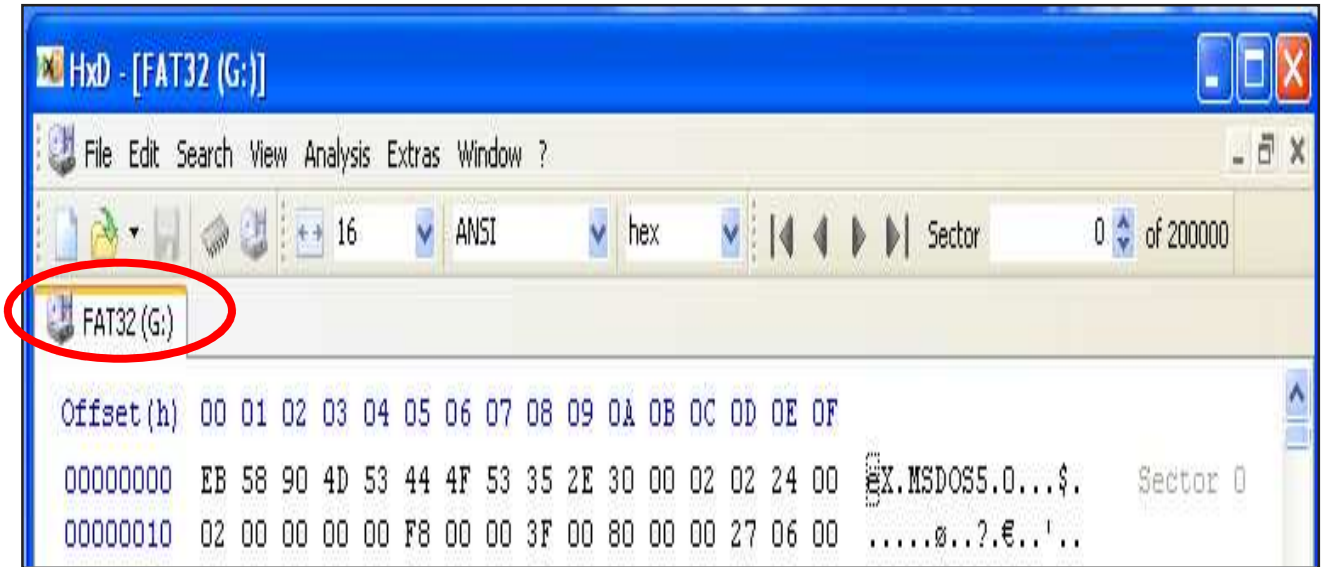
- Boot disk contains a file called the Master Boot Record (MBR)
- Stores information about partitions on a disk and their locations, size, and other important items
- Several software products can modify the MBR, such as Partition Magic’s Boot Magic
- These boot partition utilities can interfere with some computer forensics acquisition tools

Partition Mark at Start of Volume

- NTFS



- FAT32



Examining FAT Disks

- **File Allocation Table (FAT)**
  - File structure **database**
  - Microsoft originally designed for **floppy disks**
  - originally used 12 or 16 bits for each cluster
  - It is used by the operating system (OS) to manage files on hard drives
  - It is often also found on in flash memory, digital cameras and portable devices
  - Used before Windows NT and 2000

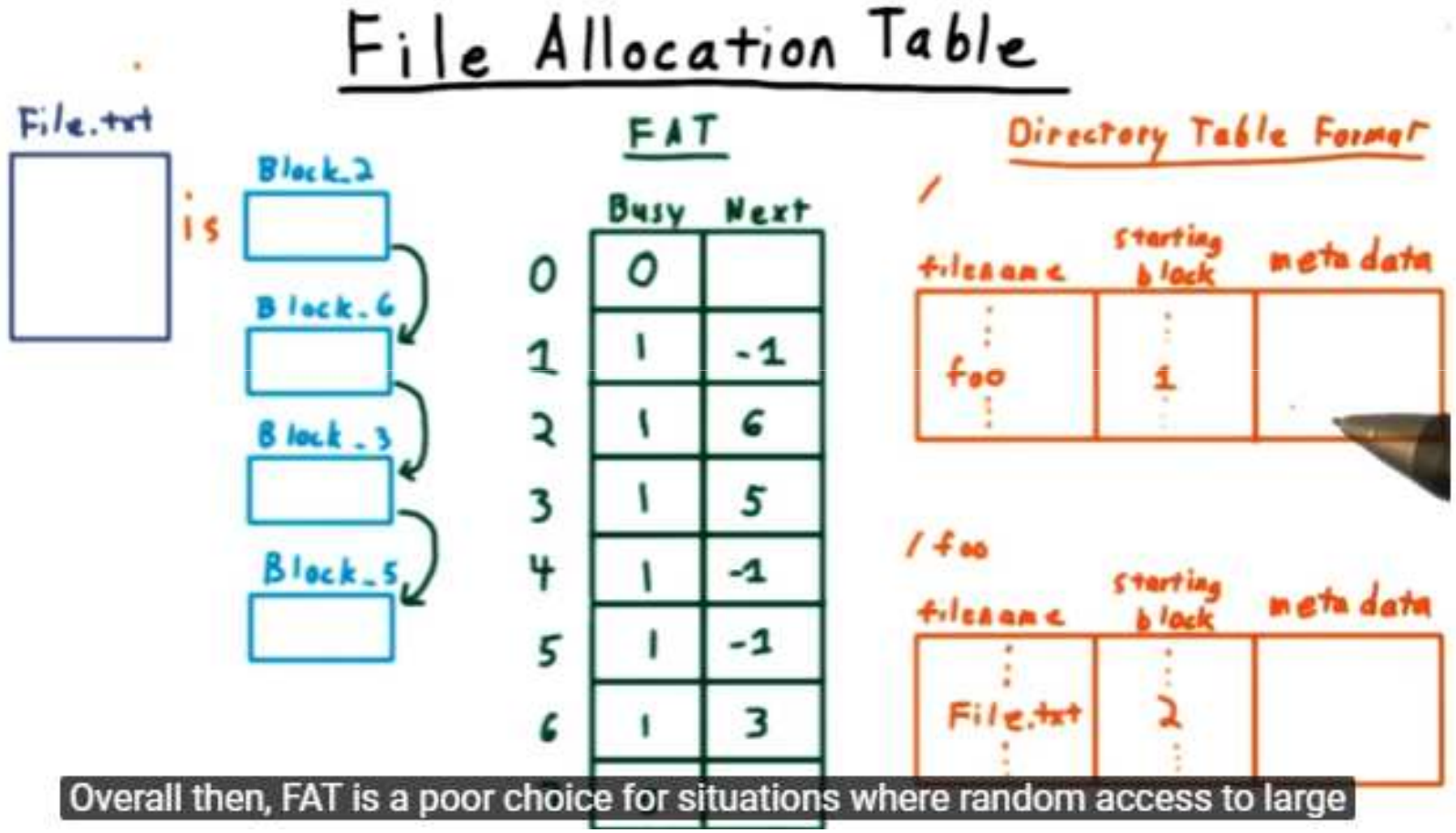
Examining FAT Disks

- FAT database is typically written to a disk's outermost track and contains:
  - Filenames, directory names, date and time stamps, the starting cluster number, and file attributes
- FAT versions
  - FAT12, FAT16, FAT32, FATX (for Xbox), and VFAT

FAT Versions

- FAT12—for floppy disks, max size 16 MB
- FAT16—allows hard disk sizes up to 2 GB
- FAT32— allows hard disk sizes up to 2 TB
- FATX—For Xbox media
  - The date stamps start at the year 2000, unlike the other FAT formats that start at 1980
- VFAT (Virtual File Allocation Table)
  - Allows long file names on Windows (MS-DOS had 8.3 limitation)

Examining FAT Disks



Examining FAT Disks

- Cluster sizes vary according to the hard disk size and file system
- This table is for FAT-16

Table 6-2 Sectors and bytes per cluster

Drive size	Number of sectors per cluster	FAT16
0–32 MB	1	512 bytes
33–64 MB	2	1 KB
65–128 MB	4	2 KB
129–255 MB	8	4 KB
256–511 MB	16	8 KB
512–1023 MB	32	16 KB
1024–2047 MB	64	32 KB
2048–4095 MB	128	68 KB

Examining FAT Disks

- Microsoft OSs allocate disk space for files by clusters
  - Results in **drive slack**
    - Unused space in a cluster between the end of an active file and the end of the cluster
- Drive slack includes:
  - **RAM slack** and **file slack**
- An unintentional side effect of FAT16 having large clusters was that it reduced fragmentation
  - As cluster size increased

Examining FAT Disks

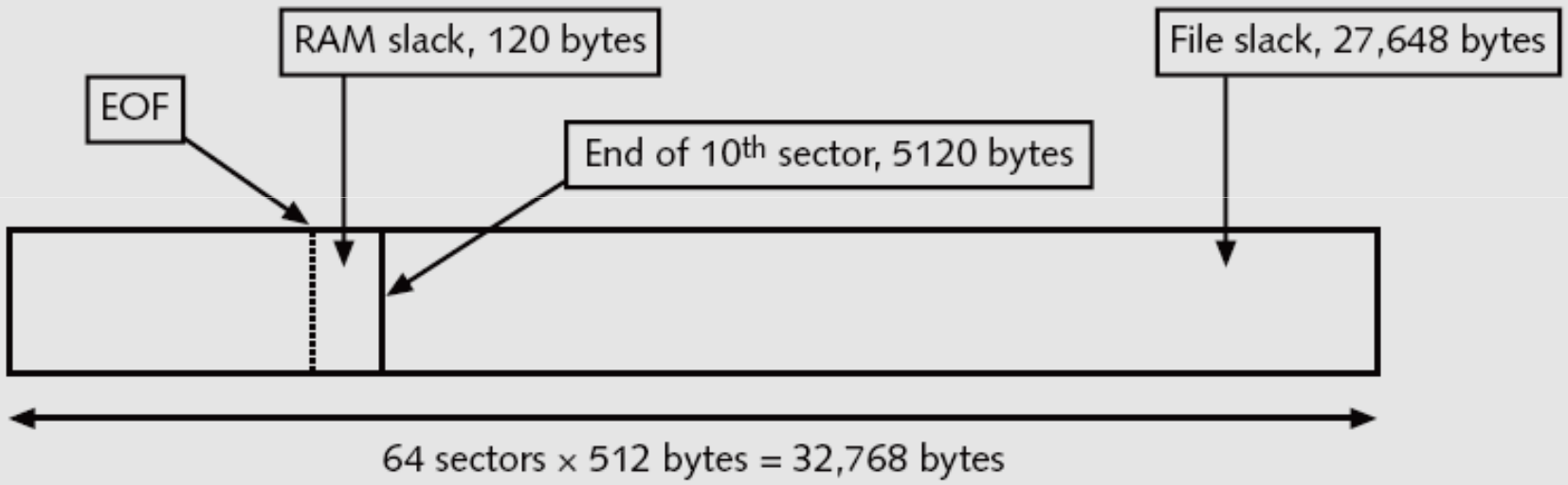


Figure 6-7 File slack space

Examining FAT Disks

- When you run out of room for an allocated cluster
  - OS allocates another cluster for your file, which creates more slack space on the disk
- As files grow and require more disk space, assigned clusters are chained together
  - The chain can be broken or fragmented

Examining FAT Disks

- When the OS stores data in a FAT file system, it assigns a starting cluster position to a file
  - Data for the file is written to the first sector of the first assigned cluster
- When this first assigned cluster is filled and runs out of room
  - FAT assigns the next available cluster to the file
- If the next available cluster isn’t contiguous to the current cluster
  - File becomes fragmented

Deleting FAT Files

- In Microsoft OSs, when a file is deleted
  - Directory entry is marked as a deleted file
    - With the HEX E5 (σ) character replacing the first letter of the filename
    - FAT chain for that file is set to 0
- Data in the file remains on the disk drive
- Area of the disk where the deleted file resides becomes **unallocated disk space**
  - Available to receive new data from newly created files or other files needing more space

Examining NTFS Disks

- **New Technology File System (NTFS)**
  - Introduced with Windows NT
  - Recommended file system for Windows
- Improvements over FAT file systems
  - NTFS provides **more information** about a file
  - NTFS gives more **control over files** and folders
- NTFS was Microsoft’s move toward a journaling file system
  - system keeps track of transactions such as file deleting or saving

Examining NTFS Disks

- In NTFS, everything written to the disk is considered a file
- On an NTFS disk
  - First data set is the **Partition Boot Sector**
  - Next is **Master File Table (MFT)**
- NTFS results in much less file slack space
- Clusters are smaller for smaller disk drives
- NTFS also uses **Unicode**
  - An international data format

Examining NTFS Disks

Table 6-3 Cluster sizes in an NTFS disk

Drive size	Sectors per cluster	Cluster size
0–512 MB	1	512 bytes
512 MB–1 GB	2	1024 bytes
1–2 GB	4	2048 bytes
2–4 GB	8	4096 bytes
4–8 GB	16	8192 bytes
8–16 GB	32	16,384 bytes
16–32 GB	64	32,768 bytes
More than 32 GB	128	65,536 bytes

NTFS File System

- MFT
  - First file in NTFS
  - contains information about all files on the disk
  - Including the system files the OS uses
- In the MFT, the first 15 records are reserved for system files
- Records in the MFT are called **metadata**

NTFS File System

Table 6-4 Metadata records in the MFT

Filename	System file	Record position	Description
$Mft	MFT	0	Base file record for each folder on the NTFS volume; other record positions in the MFT are 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 recovery of the MFT.
$LogFile	Log file	2	Previous transactions are stored here to allow recovery after a system failure 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 attribute names, numbers, and definitions.
$	Root file-name index	5	This is the root folder on the NTFS volume.

NTFS File System

Table 6-4 Metadata records in the MFT (continued)

Filename	System file	Record position	Description
$Bitmap	Boot sector	6	A map of the NTFS volume showing which clusters are in use and which are available.
$Boot	Boot sector	7	Used to mount the NTFS volume during the bootstrap process; additional code is listed here if it's the boot drive for the system.
$BadClus	Bad cluster file	8	For clusters that have unrecoverable errors, an entry of the cluster location is made in this file.
$Secure	Security file	9	Unique security descriptors for the volume are listed in this file. It's where the access control list (ACL) is maintained for all files and folders on the NTFS volume.
$Upcase	Upcase table	10	Converts all lowercase characters to uppercase Unicode characters for the NTFS volume.
$Extend	NTFS extension file	11	Optional extensions are listed here, such as quotas, object identifiers, and reparse point data.
		12–15	Reserved for future use.

MFT and File Attributes

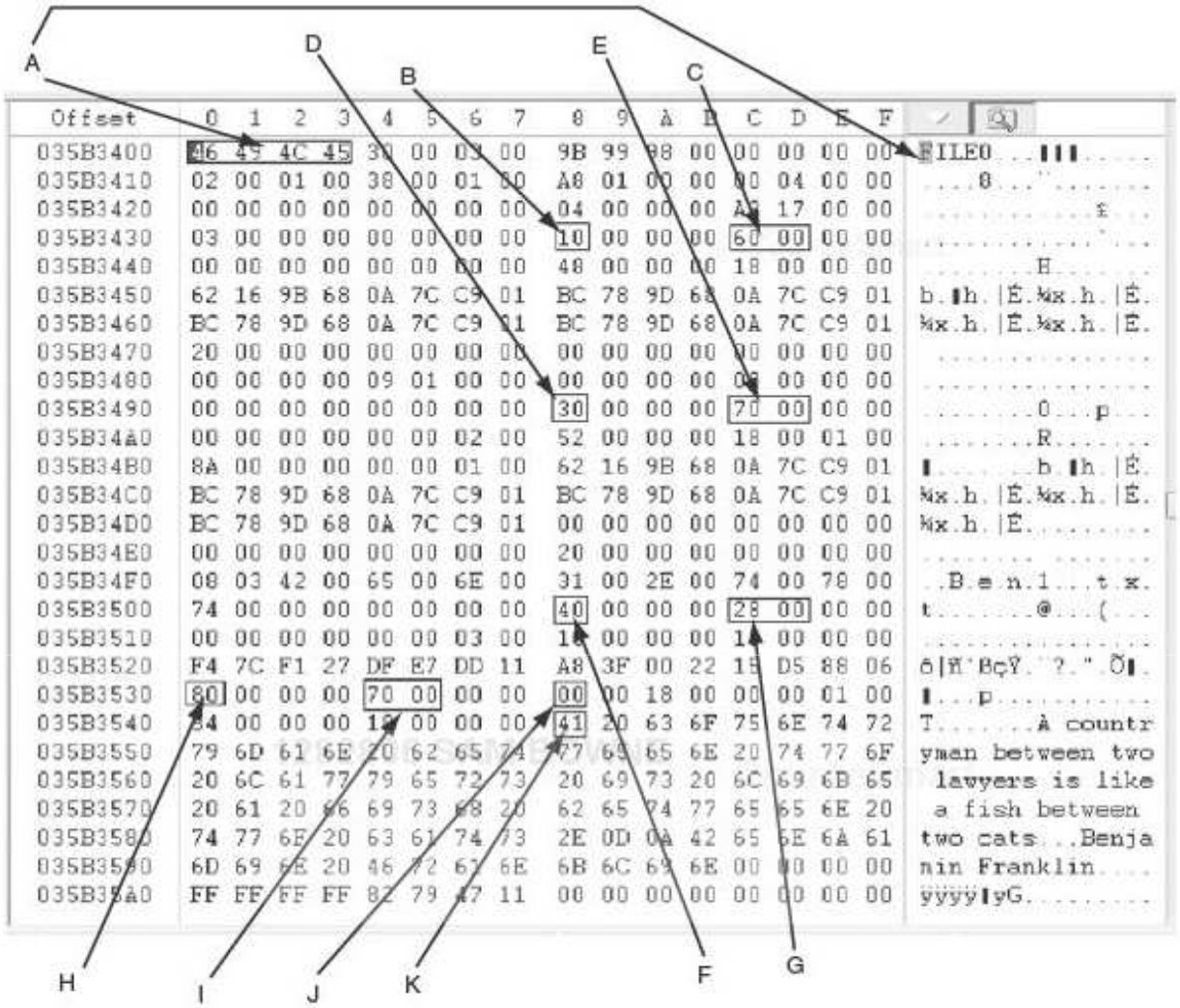
- In the NTFS MFT
  - All files and folders are stored in **separate records** of 1024 bytes each
  - Each record contains file or folder information
    - This information is divided into record fields containing metadata
  - A record field is referred to as an **attribute ID**

MFT and File Attributes

- File or folder information is typically stored in one of two ways in an MFT record:
  - Resident
    - very small files, about 512 bytes or less, all file metadata and data are stored in the MFT record
  - Nonresident
    - Files larger than 512 bytes are stored outside the MFT
    - File record provides cluster addresses where the file is stored on the drive's partition
    - These cluster addresses are referred to as **data runs**
- Each MFT record starts with a header identifying it as a resident or nonresident attribute

Table 6-5 Attributes in the MFT

Attribute ID	Purpose
0x10	$Standard Information This field contains data on file creation, alterations, MFT changes, read dates and times, and DOS file permissions.
0x20	$Attribute_List Attributes that don't fit in the MFT (nonresident attributes) are listed here along with their locations.
0x30	$File_Name The long and short names for a file are contained here. Up to 255 Unicode bytes are available for long filenames. For POSIX requirements, additional names or hard links can also be listed. Files with short filenames have only one attribute ID 0x30. Long filenames have two attribute ID 0x30s in the MFT record: one for the short name and one for the long name.
0x40	$Object_ID (for Windows NT, it's named $Volume_Version) Ownership and who has access rights to the file or folder are listed here. Every MFT record is assigned a unique GUID. Depending on your NTFS setup, some file records might not contain this attribute ID.
0x50	$Security_Descriptor Contains the access control list (ACL) for the file.
0x60	$Volume_Name The volume-unique file identifier is listed here. Not all files need this unique identifier.
0x70	$Volume_Information This field indicates the version and state of the volume.
0x80	$Data File data or data runs to nonresident files.
0x90	$Index_Root Implemented for use of folders and indexes.
0xA0	$Index_Allocation Implemented for use of folders and indexes.
0xB0	$Bitmap Implemented for use of folders and indexes.
0xC0	$Reparse_Point This field is used for volume mount points and Installable File System (IFS) filter drivers. For the IFS, it marks specific files used by drivers.
0xD0	$EA_Information For use with OS2 HPFS file systems.
0xE0	$EA For use with OS2 HPFS file systems.
0x100	$Logged_Utility_Stream This field is used by Encrypting File System in Windows 2000 and XP.



A: All MFT records start with FILE0
B: Start of attribute 0x10
C: Length of attribute 0x10 (value 60)
D: Start of attribute 0x30
E: Length of attribute 0x30 (value 70)
F: Start of attribute 0x40
G: Length of attribute 0x40 (value 28)
H: Start of attribute 0x80
I: Length of attribute 0x80 (value 70)
J: Attribute 0x80 resident flag
K: Starting position of resident data

Resident File Data in the MFT

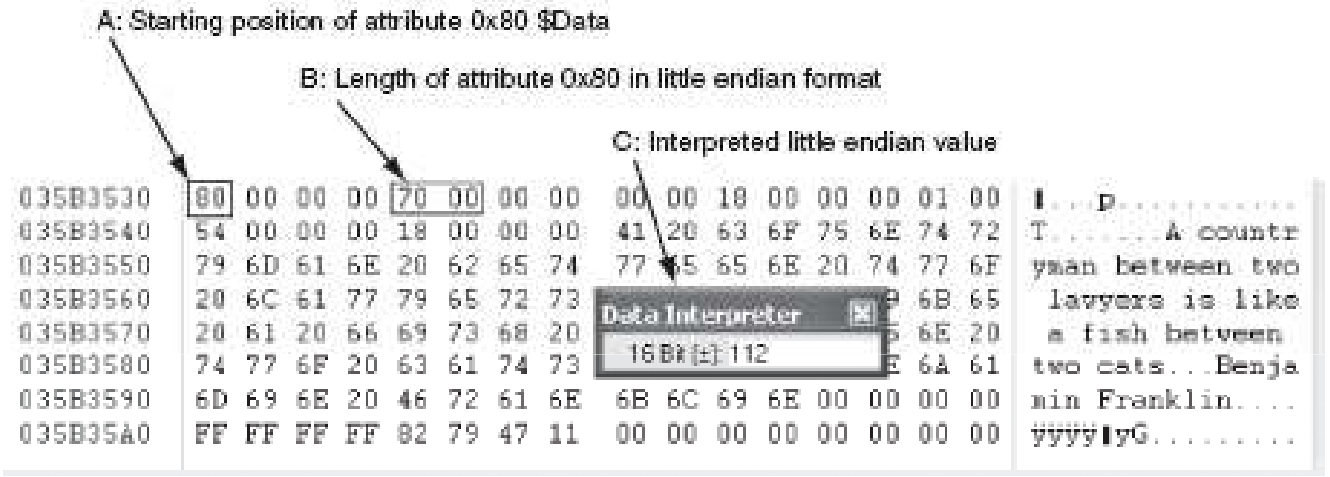
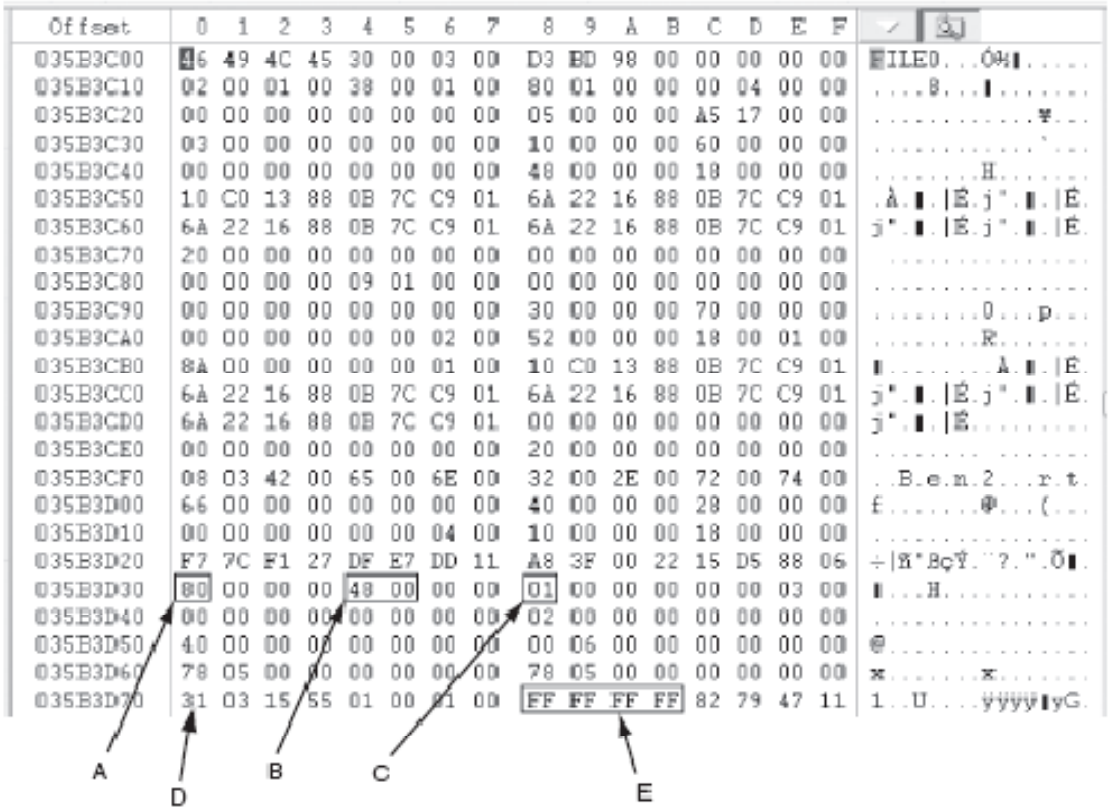


Figure 6-10 File data for a resident file

- This figure is a repeat of a portion of the previous one

Nonresident File's MFT Record



A: Start of nonresident attribute 0x80
B: Length of nonresident attribute 0x80
C: Attribute 0x80 nonresident flag
D: Starting point of data run
E: End-of-record marker (FF FF FF FF) for the MFT record

Figure 6-11 Nonresident file in an MFT record

MFT and File Attributes

- When a disk is created as an NTFS file structure
  - OS assigns logical clusters to the entire disk partition
- These assigned clusters are called **logical cluster numbers (LCNs)**
  - Become the addresses that allow the MFT to link to nonresident files on the disk's partition

NTFS Data Streams

- **Data streams**
- In NTFS, a data stream becomes an additional file attribute
  - Allows the file to be associated with different applications

NTFS Compressed Files

- NTFS provides compression similar to FAT DriveSpace 3
- Under NTFS, files, folders, or entire volumes can be compressed
- Most computer forensics tools can uncompress and analyze compressed Windows data

NTFS Encrypting File System (EFS)

- **Encrypting File System (EFS)**
  - Introduced with Windows 2000
  - Implements a **public key** and **private key** method of encrypting files, folders, or disk volumes
- When EFS is used in Windows 2000
  - A **recovery certificate** is generated and sent to the local Windows administrator account
- Users can apply EFS to files stored on their local workstations or a remote server

Deleting NTFS Files

- When a file is deleted in Windows XP, 2000, or NT
  - The OS renames it and moves it to the Recycle Bin
- Can use the Del (delete) MS-DOS command
  - Eliminates the file from the MFT listing