BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Working with Windows and CLI Systems

Cyber crime, forensics and Incident Handling

(Amit Dua)

Courtesy "Guide to Computer Forensics and Investigations Sixth Edition"

Objectives

- Explain the purpose and structure of file systems
- Describe Microsoft file structures
- Explain the structure of NTFS disks
- List some options for decrypting drives encrypted with whole disk encryption
- Explain how the Windows Registry works
- Describe Microsoft startup tasks
- Explain the purpose of a virtual machine

Understanding File Systems

- File system
  - Gives OS a road map to data on a disk
- Type of file system an OS uses determines how data is stored on the disk
- When you need to access a suspect's computer to acquire or inspect data
  - You should be familiar with both the computer's OS and file systems

Understanding the Boot Sequence (1 of 3)

- Complementary Metal Oxide Semiconductor (CMOS)
- Computer stores system configuration and date and time information in the CMOS
- When power to the system is off
- Basic Input/Output System (BIOS) or Extensible Firmware Interface (EFI)
- Contains programs that perform input and output at the hardware level

Understanding the Boot Sequence (2 of 3)

- **Bootstrap process**
- Contained in ROM, tells the computer how to proceed
- Displays the key or keys you press to open the CMOS setup screen
- CMOS should be modified to boot from a forensic floppy disk or CD

Understanding the Boot Sequence (3 of 3)

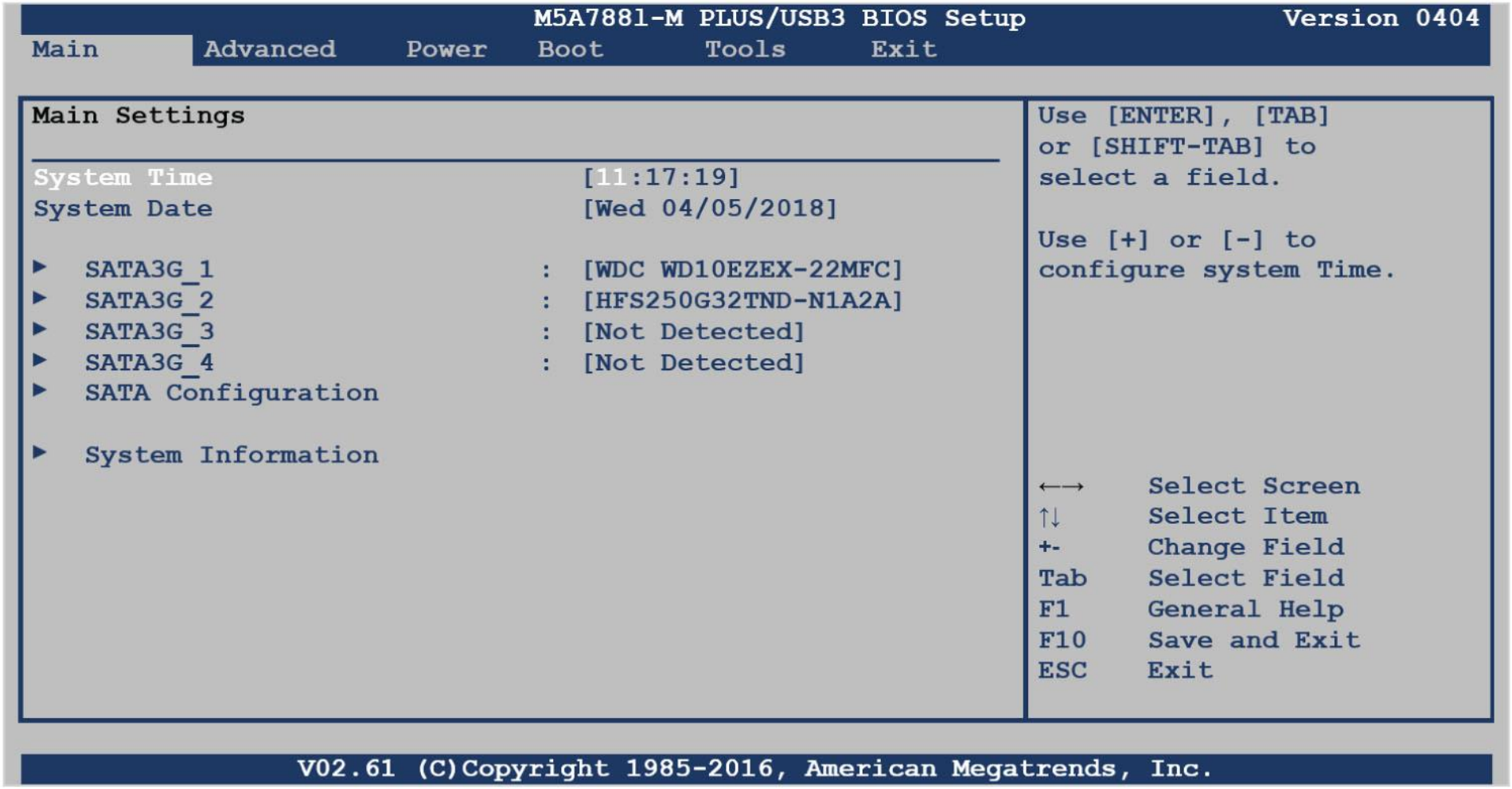


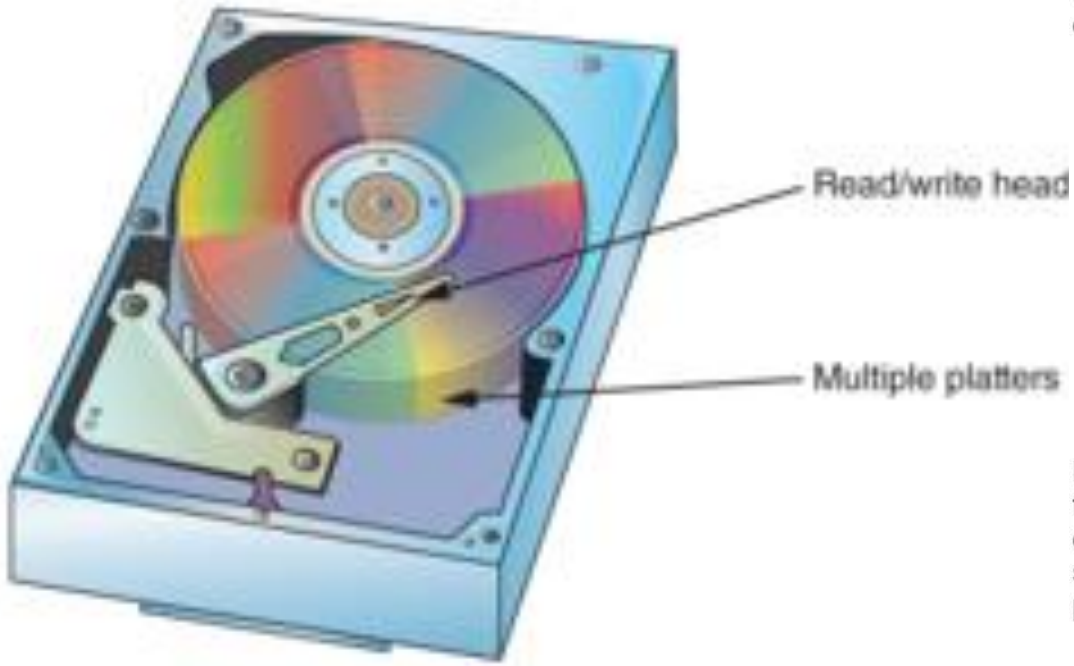
Figure 5-1 A typical CMOS setup screen

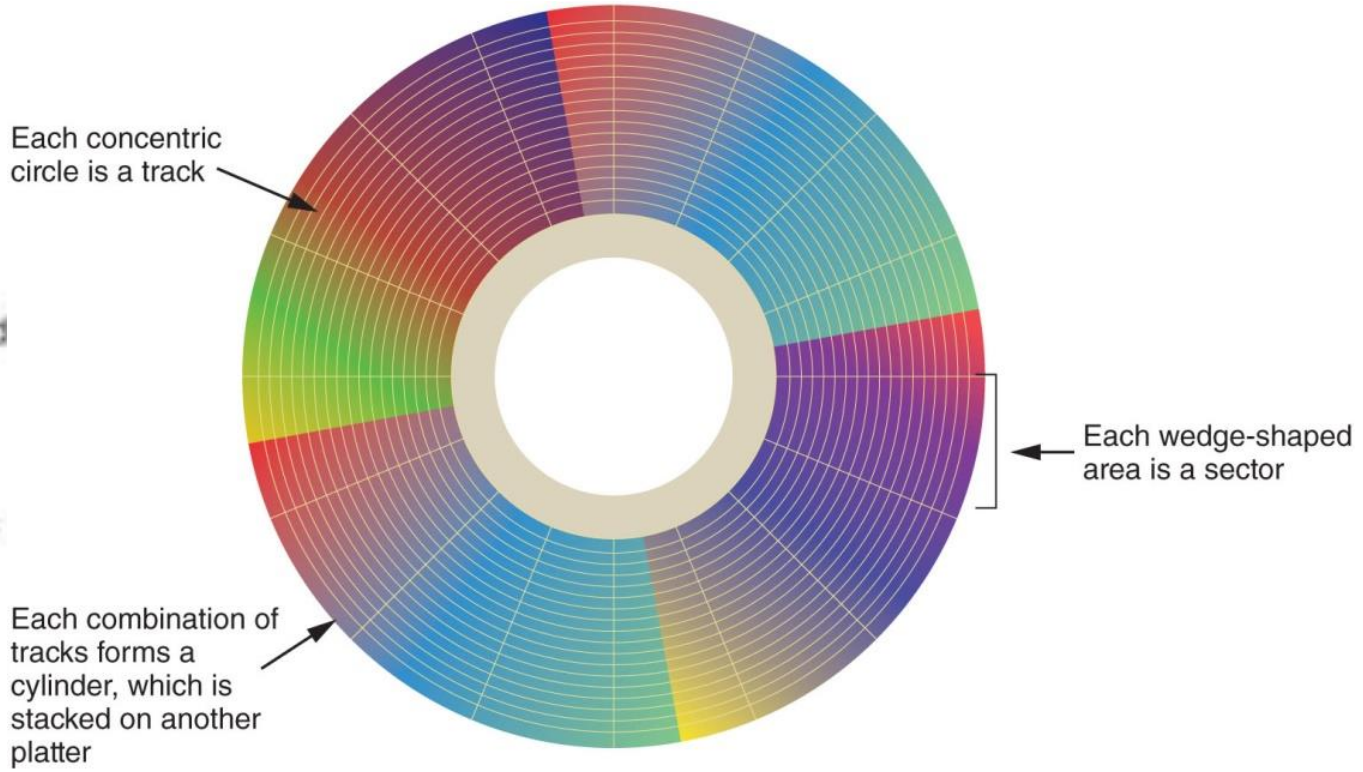
Source: American Megatrends, Inc., *https://ami.com/en/*

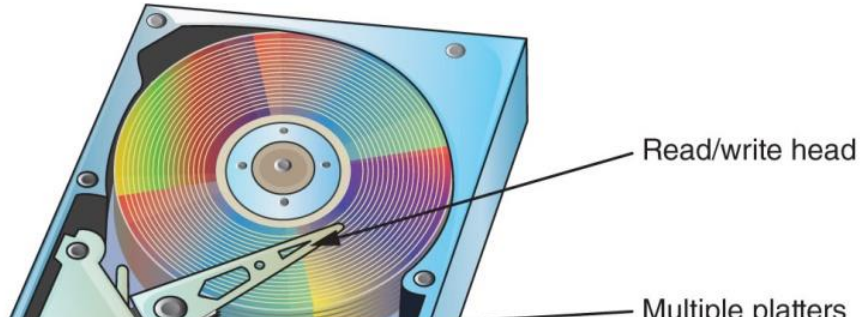
Understanding Disk Drives (1 of 4)

- Disk drives are made up of one or more platters coated with magnetic material
- Disk drive components
  - Geometry
  - Head
  - Tracks
  - Cylinders
  - Sectors

Understanding Disk Drives (2 of 4)







Understanding Disk Drives

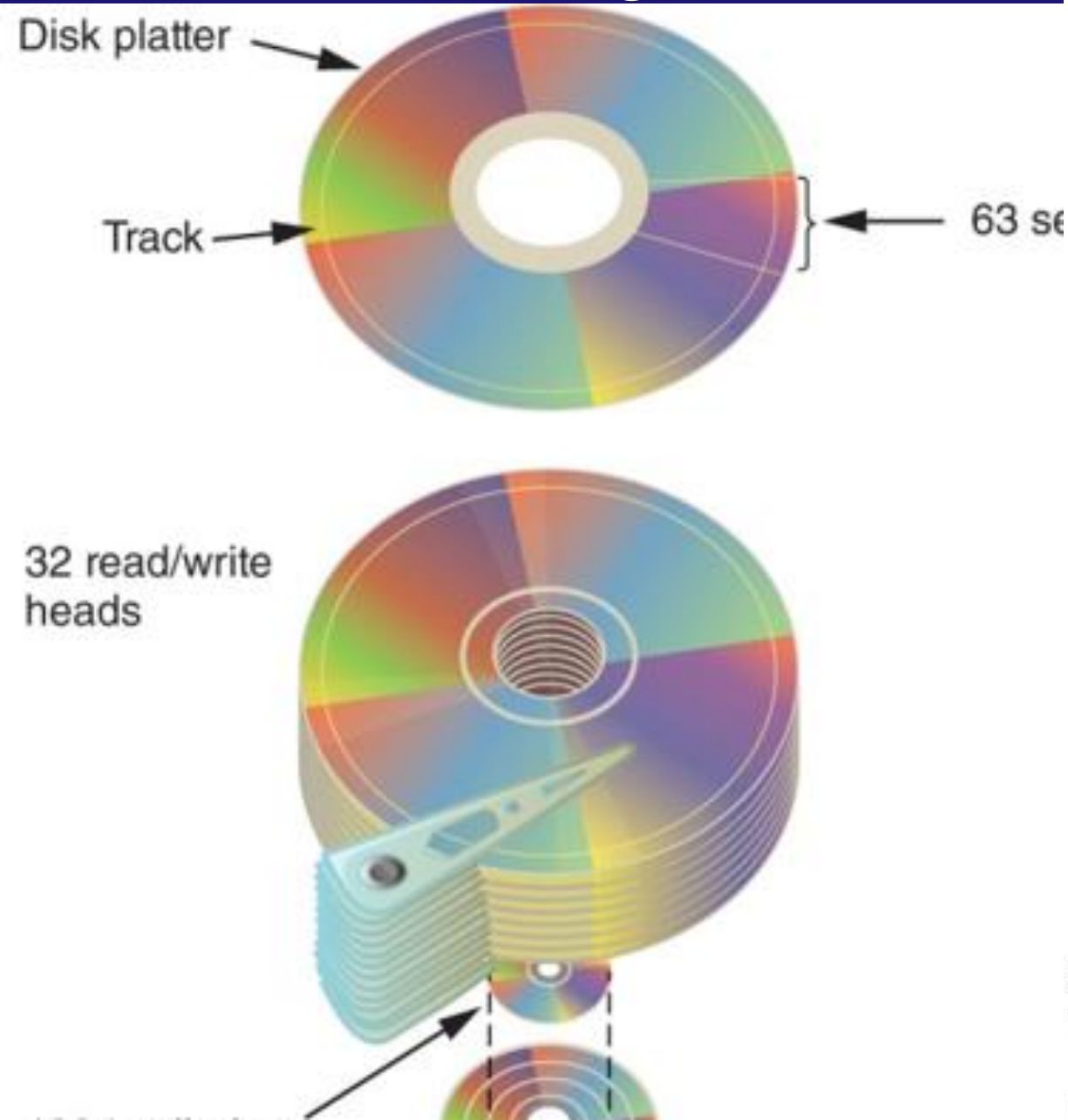


Figure 5-3 CHS calculation

Understanding Disk Drives (4 of 4)

- Properties handled at the drive's hardware or firmware level
  - **Zone bit recording (ZBR)**
  - **Track density**
  - **Areal density**
  - **Head and cylinder skew**

Solid-State Storage Devices

- All flash memory devices have a feature called **wear-leveling**
  - An internal firmware feature used in solid-state drives that ensures even wear of read/writes for all memory cells
- When dealing with **solid-state devices**, making a full forensic copy as soon as possible is crucial
  - In case you need to recover data from unallocated disk space

Exploring Microsoft File Structures (1 of 2)

- In Microsoft file structures, sectors are grouped to form **clusters**
  - Storage allocation units of one or more sectors
- Clusters range from 512 bytes up to 32,000 bytes each
- Combining sectors minimizes the overhead of writing or reading files to a disk

Exploring Microsoft File Structures (2 of 2)

- Clusters are numbered sequentially starting at 0 in NTFS and 2 in FAT
  - First sector of all disks contains a system area, the boot record, and a file structure database
- 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 (1 of 3)

- A **partition** is a logical drive
- Windows OSs can have three primary partitions followed by an extended partition that can contain one or more logical drives
- Hidden partitions or voids
  - Large unused gaps between partitions on a disk
- **Partition gap**
  - Unused space between partitions

Disk Partitions (2 of 3)

- The partition table is in the **Master Boot Record (MBR)**
  - Located at sector 0 of the disk drive
- In a hexadecimal editor, such as **WinHex**, you can find the first partition at offset 0x1BE
  - The file system's hexadecimal code is offset 3 bytes from 0x1BE for the first partition

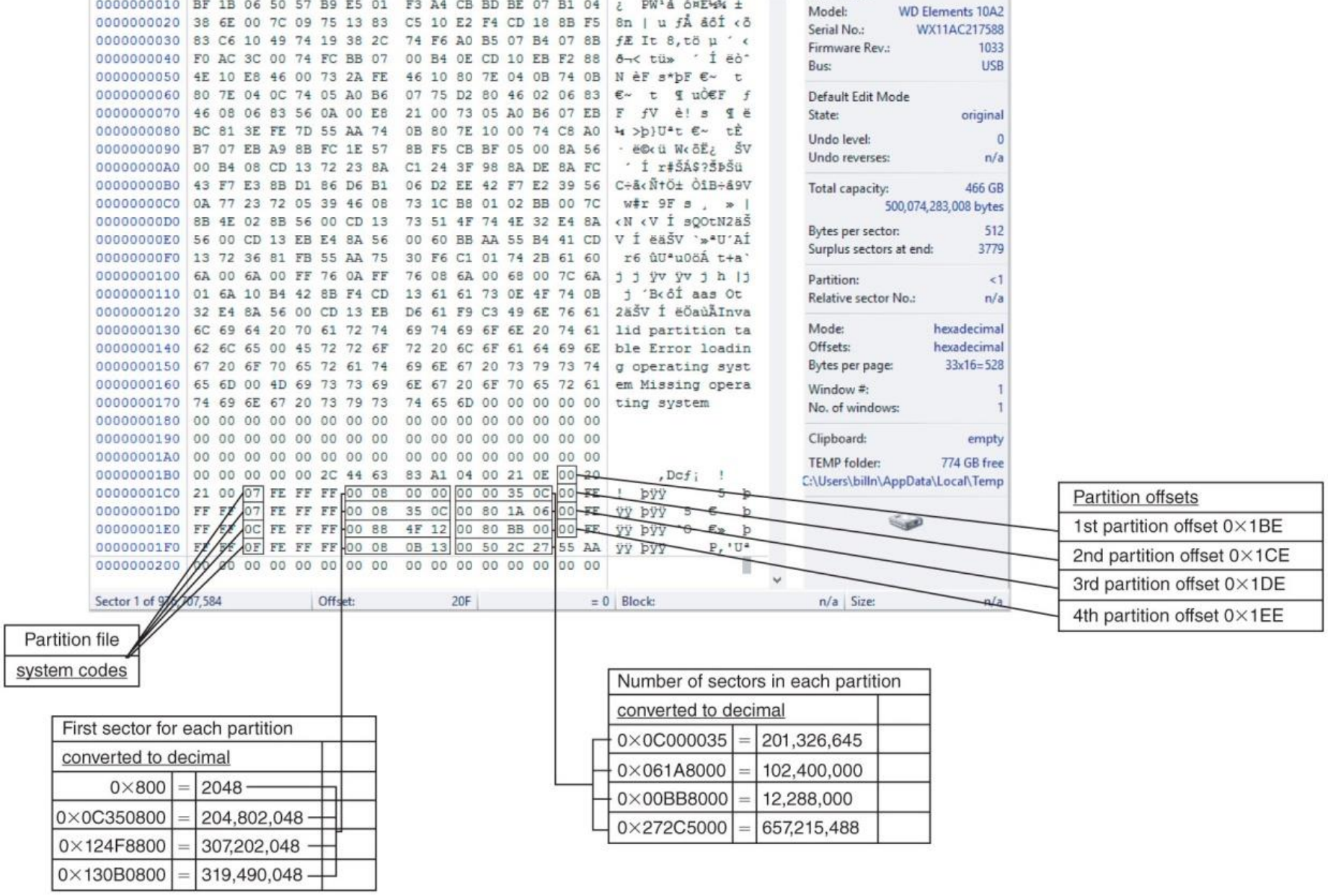


Figure 5-4 The partition table in a hexadecimal editor

Source: X-Ways AG, *www.x-ways.net*

Examining FAT Disks (1 of 7)

- **File Allocation Table (FAT)**
  - File structure database that Microsoft originally designed for floppy 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**
- Three current FAT versions
  - **FAT16, FAT32, and exFAT** (used for mobile personal storage devices)
- Cluster sizes vary according to the hard disk size and file system

Examining FAT Disks (2 of 7)

Table 5-2 Sectors and bytes per cluster

Drive size	Sectors per cluster	FAT16
8–32 MB	1	512 bytes
32–64 MB	2	1 KB
64–128 MB	4	2 KB
128–256 MB	8	4 KB
256–512 MB	16	8 KB
512–1024 MB	32	16 KB
1024–2048 MB	64	32 KB
2048–4096 MB	128	64 KB

Examining FAT Disks (3 of 7)

- 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's content and the end of the cluster
- Drive slack includes:
  - **RAM slack** and **file slack**
- An unintentional side effect of FAT16 allowing large clusters was that it reduced fragmentation
  - As cluster size increased

Examining FAT Disks (4 of 7)

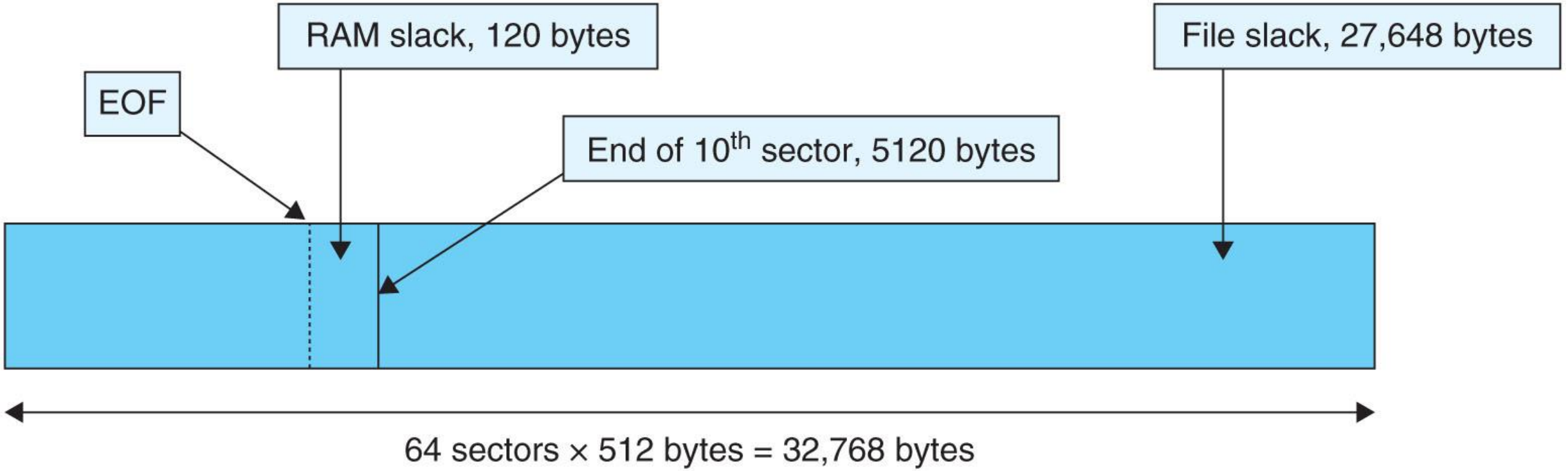


Figure 5-8 File slack space

Examining FAT Disks (5 of 7)

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

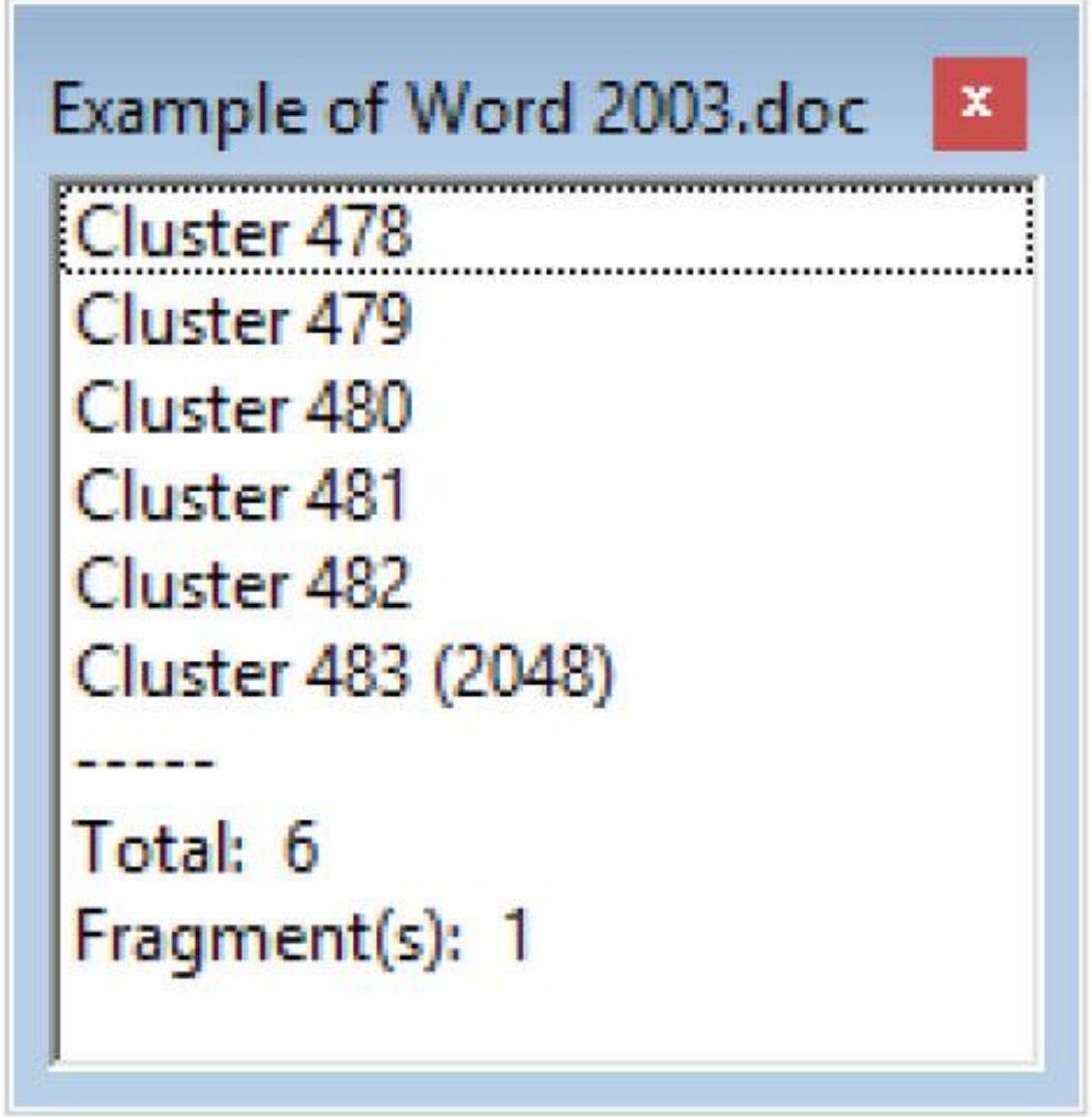


Figure 5-9 Chained sectors associated with

Examining FAT Disks (7 of 7)

- 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 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 (1 of 3)

- **NT File System (NTFS)**
  - Introduced with Windows NT
  - Primary file system for Windows 10
- 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**
  - It **records a transaction** before the system carries it out

Examining NTFS Disks (2 of 3)

- 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 (3 of 3)

Table 5-3 Cluster sizes in an NTFS disk

Drive size	Sectors per cluster	Cluster size
7–512 MB	8	4 KB
512 MB–1 GB	8	4 KB
1–2 GB	8	4 KB
2 GB–2 TB	8	4 KB
2–16 TB	8	4 KB
16–32 TB	16	8 KB
32–64 TB	32	16 KB
64–128 TB	64	32 KB
128–256 TB	128	64 KB

NTFS System Files (1 of 3)

- MFT 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 (2 of 3)

Table 5-4 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.

Table 5-4 Metadata records in the MFT (*continued*)

Filename	System file	Record position	Description
$AttrDef	Attribute definitions	4	A table listing attribute names, numbers, and definitions.
$	Root filename index	5	This is the root folder on the NTFS volume.
$Bitmap	Boot sector	6	A map of the NTFS partition shows 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 (1 of 7)

- 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**
- File or folder information is typically stored in one of two ways in an MFT record:
  - Resident and nonresident

MFT and File Attributes (2 of 7)

- Files larger than 512 bytes are stored outside the MFT
  - MFT record provides **cluster addresses** where the file is stored on the drive's partition
    - Referred to as **data runs**
- Each MFT record starts with a **header** identifying it as a resident or nonresident attribute

7)

Table 5-5 Attributes in the MFT

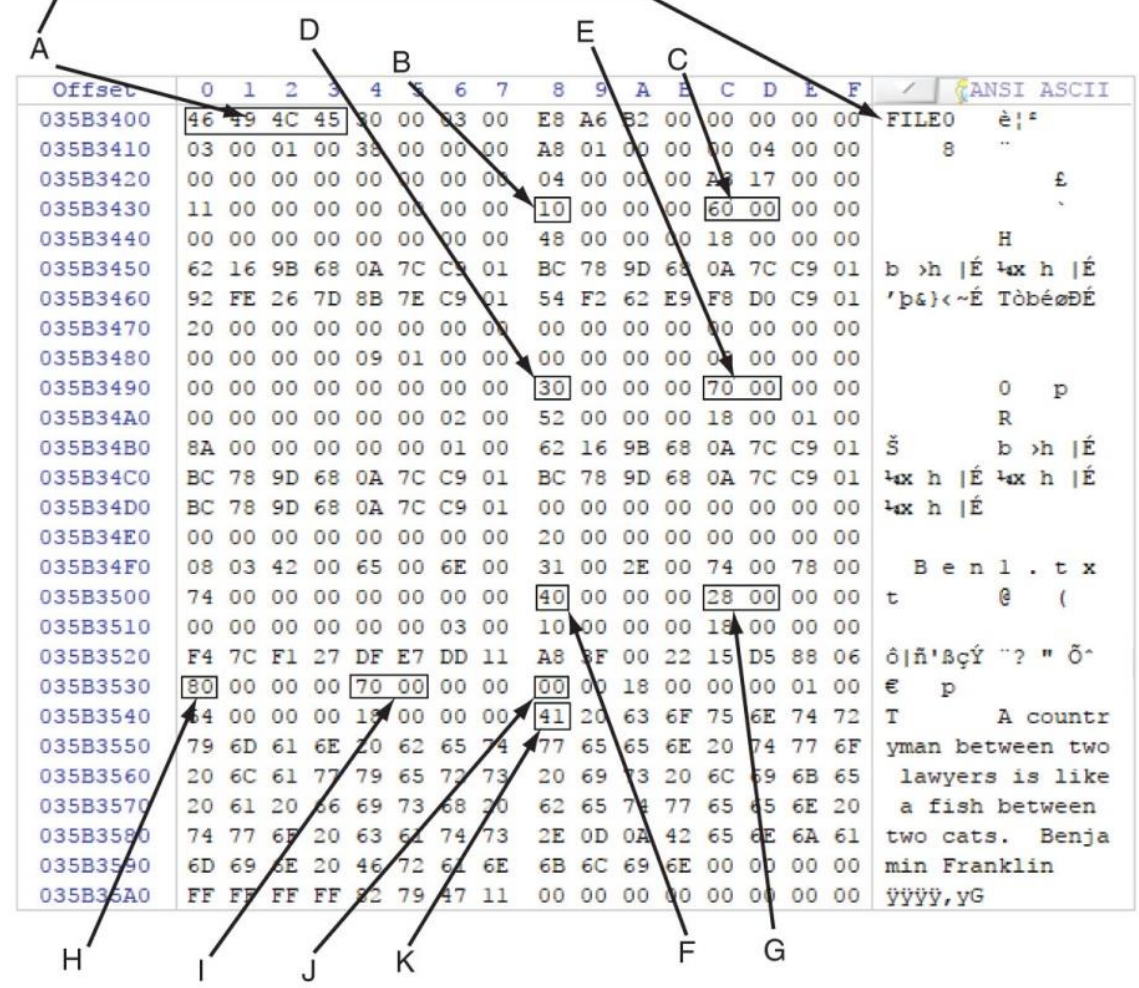
Attribute ID	Purpose
0x10	$Standard Information This field contains data on file creation, alterations, MFT changes, read dates and times, and DOS file permissions.
0x20	$Attribute_List Attributes that don't fit in the MFT (nonresident attributes) are listed here along with their locations.
0x30	$File_Name 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. In older Windows OSs, long filenames have two ID 0x30s in the MFT record: one for the short name and one for the long name. In Windows 10, there's only one 0x30 that combines the short and long filenames.
0x40	$Object_ID ($Volume_Version in Windows NT) 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 for resident files or data runs for nonresident files.
0x90	$Index_Root Implemented for use of folders and indexes.
0xA0	$Index_Allocation Implemented for use of folders and indexes.

MFT and File Attributes (4 of 7)

Table 5-5 Attributes in the MFT (*continued*)

Attribute ID	Purpose
0xB0	$Bitmap A bitmap indicating cluster status, such as which clusters are in use and which are available.
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 OS/2 HPFS.
0xE0	For use with OS/2 HPFS.
0x100	$Logged_Utility_Stream This field is used by Encrypting File System (EFS) in Windows 2000 and later.

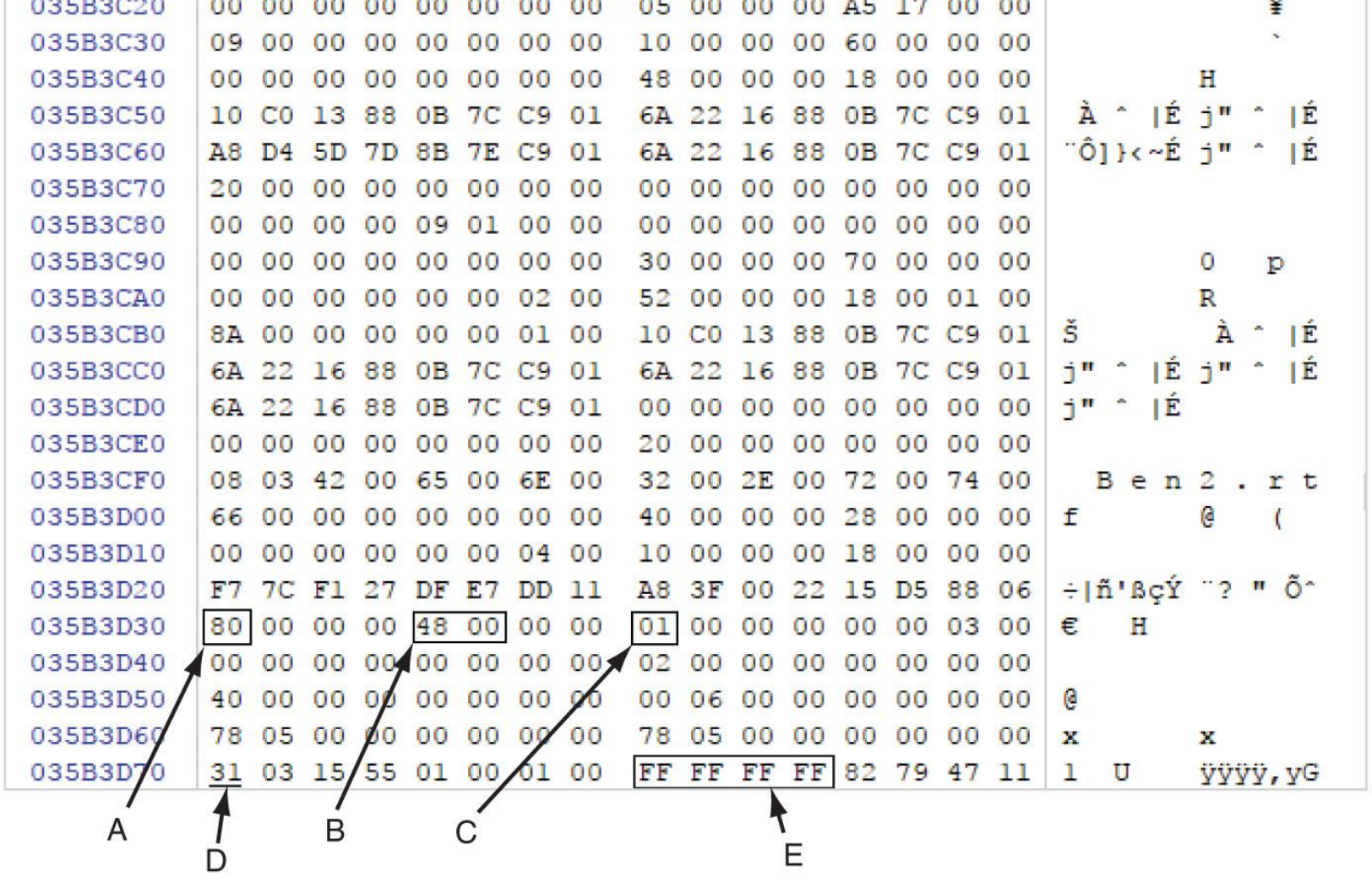
MFT and File Attributes (5 of 7



Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	ANSI ASCII
035B3400	46	49	4C	45	30	00	03	00	E8	A6	B2	00	00	00	00	00	FILE0 è¦²
035B3410	03	00	01	00	38	00	00	00	A8	01	00	00	00	04	00	00	8 ¨
035B3420	00	00	00	00	00	00	00	00	04	00	00	00	A3	17	00	00	£
035B3430	11	00	00	00	00	00	00	00	10	00	00	00	60	00	00	00	`
035B3440	00	00	00	00	00	00	00	00	48	00	00	00	18	00	00	00	H
035B3450	62	16	9B	68	0A	7C	C9	01	BC	78	9D	68	0A	7C	C9	01	b ›h |É ¼x h |É
035B3460	92	FE	26	7D	8B	7E	C9	01	54	F2	62	E9	F8	D0	C9	01	'þ&}‹~É TòbéøÐÉ
035B3470	20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
035B3480	00	00	00	00	09	01	00	00	00	00	00	00	00	00	00	00	
035B3490	00	00	00	00	00	00	00	00	30	00	00	00	70	00	00	00	0 p
035B34A0	00	00	00	00	00	00	02	00	52	00	00	00	18	00	01	00	R
035B34B0	8A	00	00	00	00	00	01	00	62	16	9B	68	0A	7C	C9	01	Š b ›h |É
035B34C0	BC	78	9D	68	0A	7C	C9	01	BC	78	9D	68	0A	7C	C9	01	¼x h |É ¼x h |É
035B34D0	BC	78	9D	68	0A	7C	C9	01	00	00	00	00	00	00	00	00	¼x h |É
035B34E0	00	00	00	00	00	00	00	00	20	00	00	00	00	00	00	00	
035B34F0	08	03	42	00	65	00	6E	00	31	00	2E	00	74	00	78	00	B e n 1 . t x
035B3500	74	00	00	00	00	00	00	00	40	00	00	00	28	00	00	00	t @ (
035B3510	00	00	00	00	00	00	03	00	10	00	00	00	18	00	00	00	
035B3520	F4	7C	F1	27	DF	E7	DD	11	A8	8F	00	22	15	D5	88	06	ô|ñ'ßçÝ ¨? " Õˆ
035B3530	80	00	00	00	70	00	00	00	00	00	18	00	00	00	01	00	€ p
035B3540	54	00	00	00	18	00	00	00	41	20	63	6F	75	6E	74	72	T A countr
035B3550	79	6D	61	6E	20	62	65	74	77	65	65	6E	20	74	77	6F	yman between two
035B3560	20	6C	61	77	79	65	72	73	20	69	73	20	6C	69	6B	65	lawyers is like
035B3570	20	61	20	66	69	73	68	20	62	65	74	77	65	65	6E	20	a fish between
035B3580	74	77	6F	20	63	61	74	73	2E	0D	0A	42	65	6E	6A	61	two cats. Benja
035B3590	6D	69	6E	20	46	72	61	6E	6B	6C	69	6E	00	00	00	00	min Franklin
035B35A0	FF	FF	FF	FF	82	79	47	11	00	00	00	00	00	00	00	00	ÿÿÿÿ,yG

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

Figure 5-10 Resident file in an 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 5-12 Nonresident file in an MFT record

Source: X-Ways AG, *www.x-ways.net*

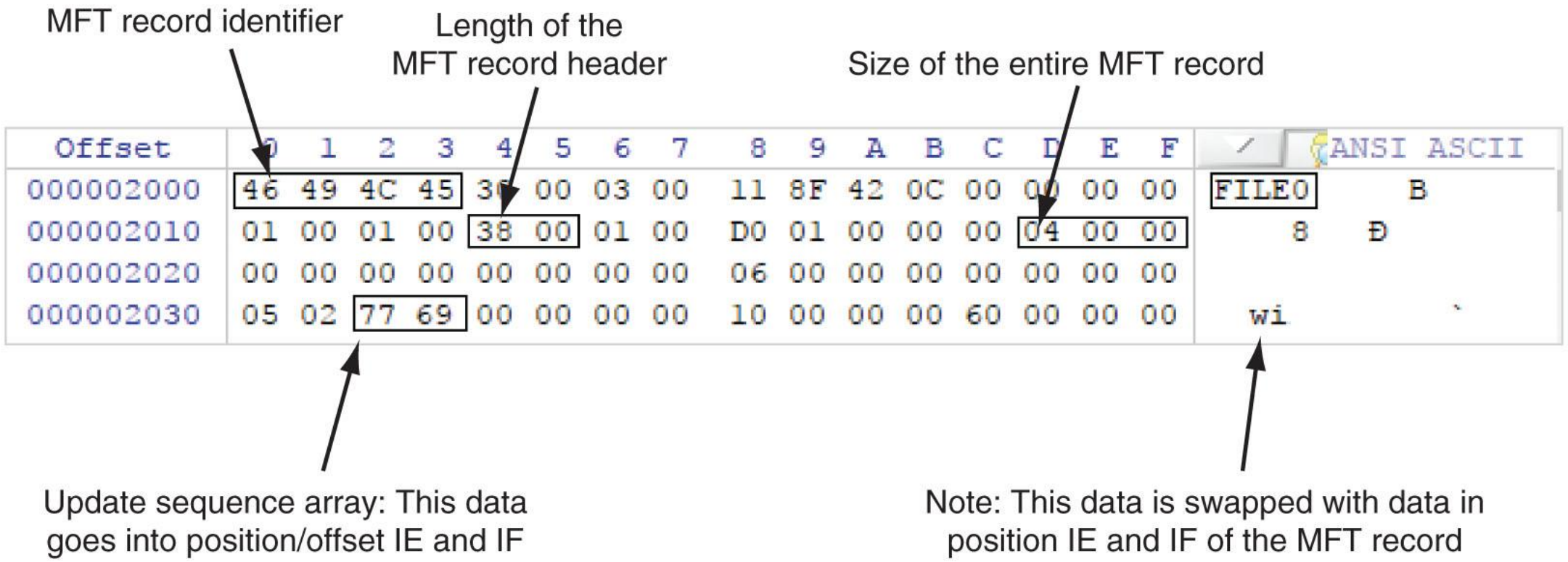
MFT and File Attributes (7 of 7)

- 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**
- When data is first written to nonresident files, an LCN address is assigned to the file
  - This **LCN becomes the file's virtual cluster number (VCN)**

MFT Structures for File Data (1 of 7)

- For the header of all MFT records, the record fields of interest are as follows:
  - *At offset 0x00 - the MFT record identifier FILE*
  - *At offset 0x1C to 0x1F - size of the MFT record*
  - *At offset 0x14 - length of the header (indicates where the next attribute starts)*
  - *At offset 0x32 and 0x33 - the update sequence array, which stores the last 2 bytes of the first sector of the MFT record*

MFT Structures for File Data (2 of 7)



Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	ANSI ASCII
000002000	46	49	4C	45	30	00	03	00	11	8F	42	0C	00	00	00	00	FILE0 B
000002010	01	00	01	00	38	00	01	00	D0	01	00	00	00	04	00	00	8 Ð
000002020	00	00	00	00	00	00	00	00	06	00	00	00	00	00	00	00	
000002030	05	02	77	69	00	00	00	00	10	00	00	00	60	00	00	00	wi `

Figure 5-13 An MFT header

Source: X-Ways AG, *www.x-ways.net*

MFT Structures for File Data (3 of 7)

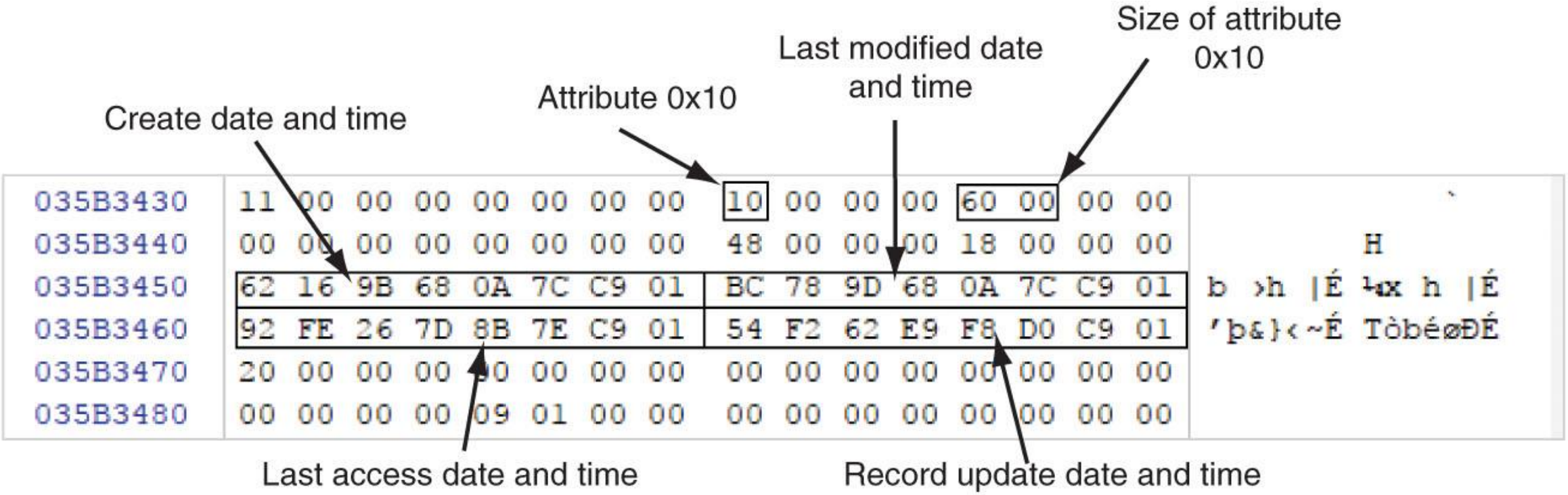
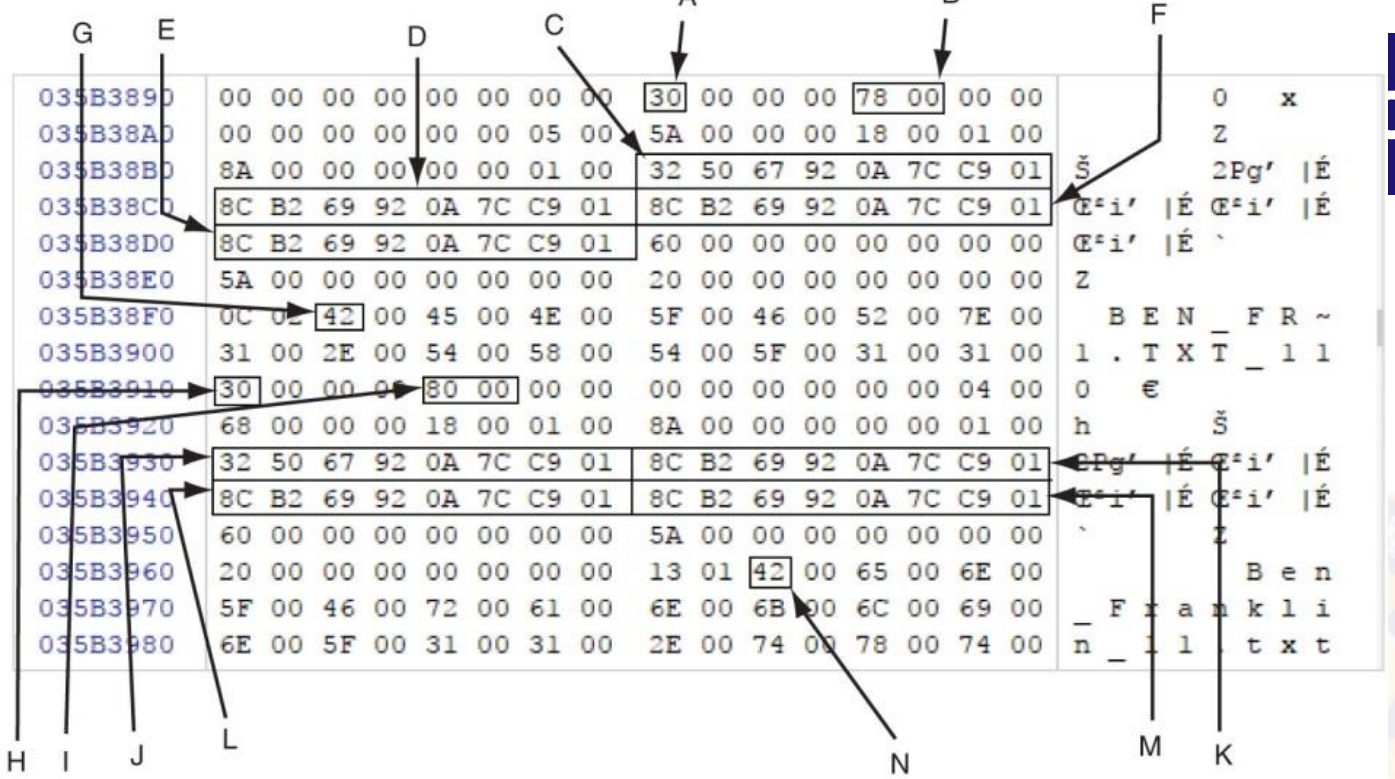


Figure 5-14 Attribute 0x10: Standard Information

Source: X-Ways AG, *www.x-ways.net*

# of 7)



035B3890	00	00	00	00	00	00	00	00	30	00	00	00	78	00	00	00	0 x
035B38A0	00	00	00	00	00	00	05	00	5A	00	00	00	18	00	01	00	Z
035B38B0	8A	00	00	00	00	00	01	00	32	50	67	92	0A	7C	C9	01	Š 2Pg' |É
035B38C0	8C	B2	69	92	0A	7C	C9	01	8C	B2	69	92	0A	7C	C9	01	Œ²i' |É Œ²i' |É
035B38D0	8C	B2	69	92	0A	7C	C9	01	60	00	00	00	00	00	00	00	Œ²i' |É `
035B38E0	5A	00	00	00	00	00	00	00	20	00	00	00	00	00	00	00	Z
035B38F0	0C	02	42	00	45	00	4E	00	5F	00	46	00	52	00	7E	00	B E N _ F R ~
035B3900	31	00	2E	00	54	00	58	00	54	00	5F	00	31	00	31	00	1 . T X T _ 1 1
035B3910	30	00	00	00	80	00	00	00	00	00	00	00	00	00	04	00	0 €
035B3920	68	00	00	00	18	00	01	00	8A	00	00	00	00	00	01	00	h Š
035B3930	32	50	67	92	0A	7C	C9	01	8C	B2	69	92	0A	7C	C9	01	2Pg' |É Œ²i' |É
035B3940	8C	B2	69	92	0A	7C	C9	01	8C	B2	69	92	0A	7C	C9	01	Œ²i' |É Œ²i' |É
035B3950	60	00	00	00	00	00	00	00	5A	00	00	00	00	00	00	00	` Z
035B3960	20	00	00	00	00	00	00	00	13	01	42	00	65	00	6E	00	B e n
035B3970	5F	00	46	00	72	00	61	00	6E	00	6B	00	6C	00	69	00	_ F r a n k l i
035B3980	6E	00	5F	00	31	00	31	00	2E	00	74	00	78	00	74	00	n _ 1 1 . t x t

A: Attribute 0x30 short filename
B: Size of attribute 0x30 short filename
C: Short create date and time
D: Short last modified date and time
E: Short last access date and time
F: Short record update date and time
G: Starting position of short filename
H: Attribute 0x30 long filename
I: Size of attribute 0x30 long filename
J: Long create date and time
K: Long last modified date and time
L: Long last access date and time
M: Long record update date and time
N: Starting position of long filename

Figure 5-15 Attribute 0x30: short and long filenames

Source: X-Ways AG, *www.x-ways.net*

MFT Structures for File Data (5 of 7)

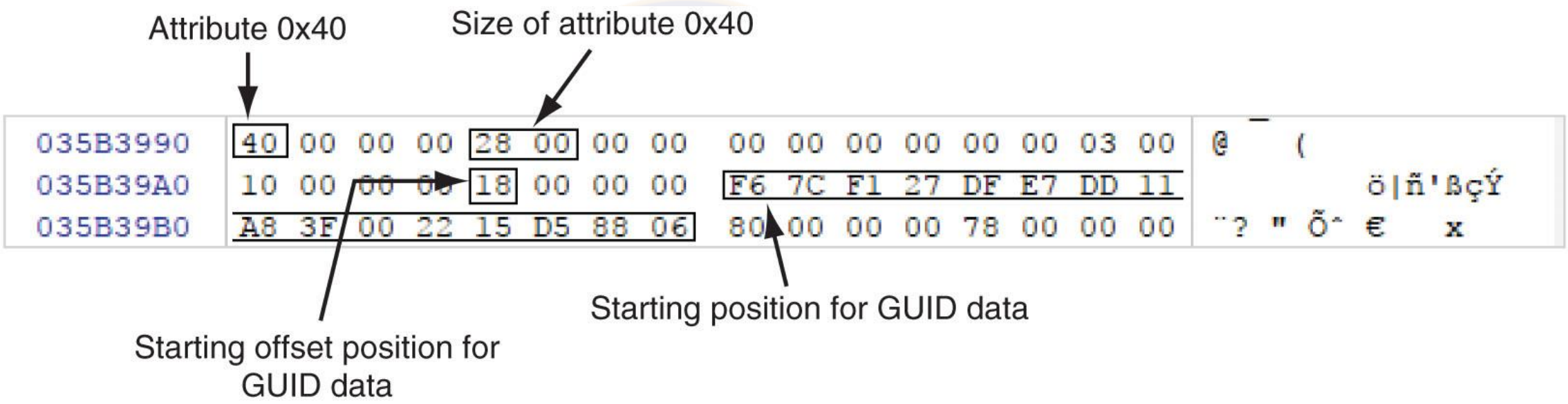


Figure 5-16 Attribute 0x40: Object_ID

Source: X-Ways AG, *www.x-ways.net*

MFT Structures for File Data (6 of 7)

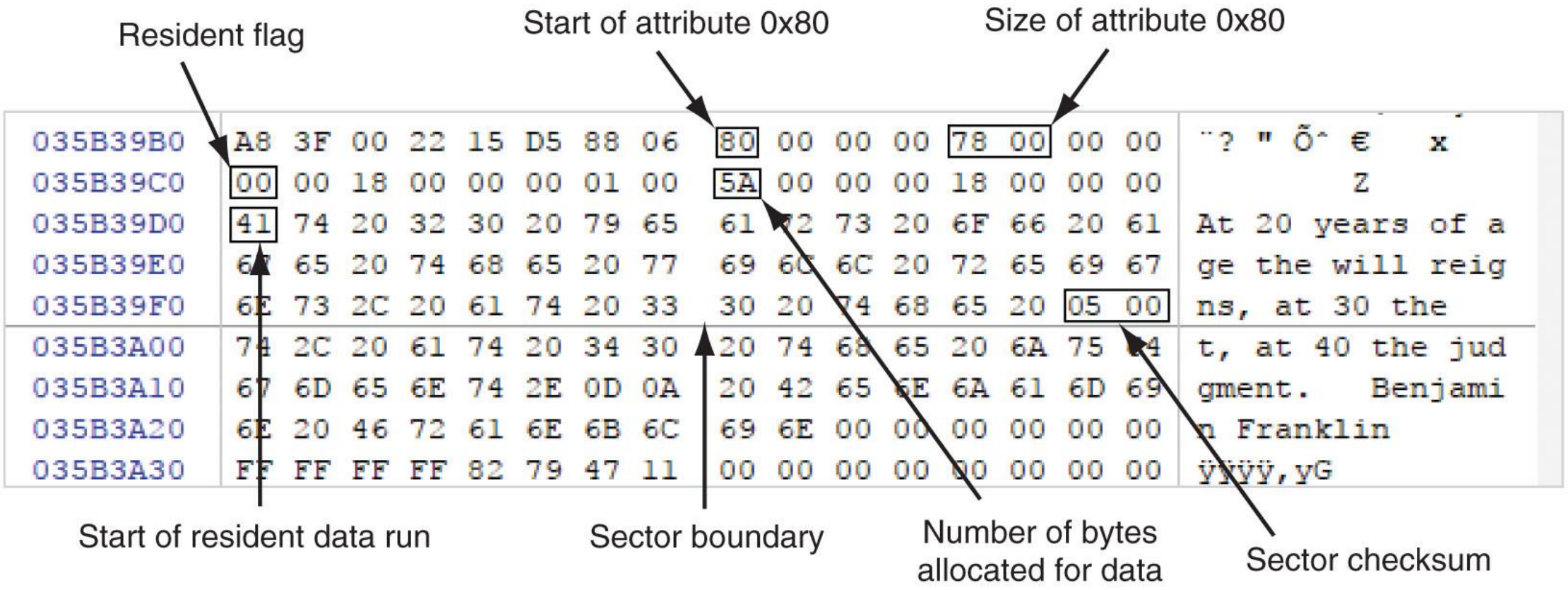
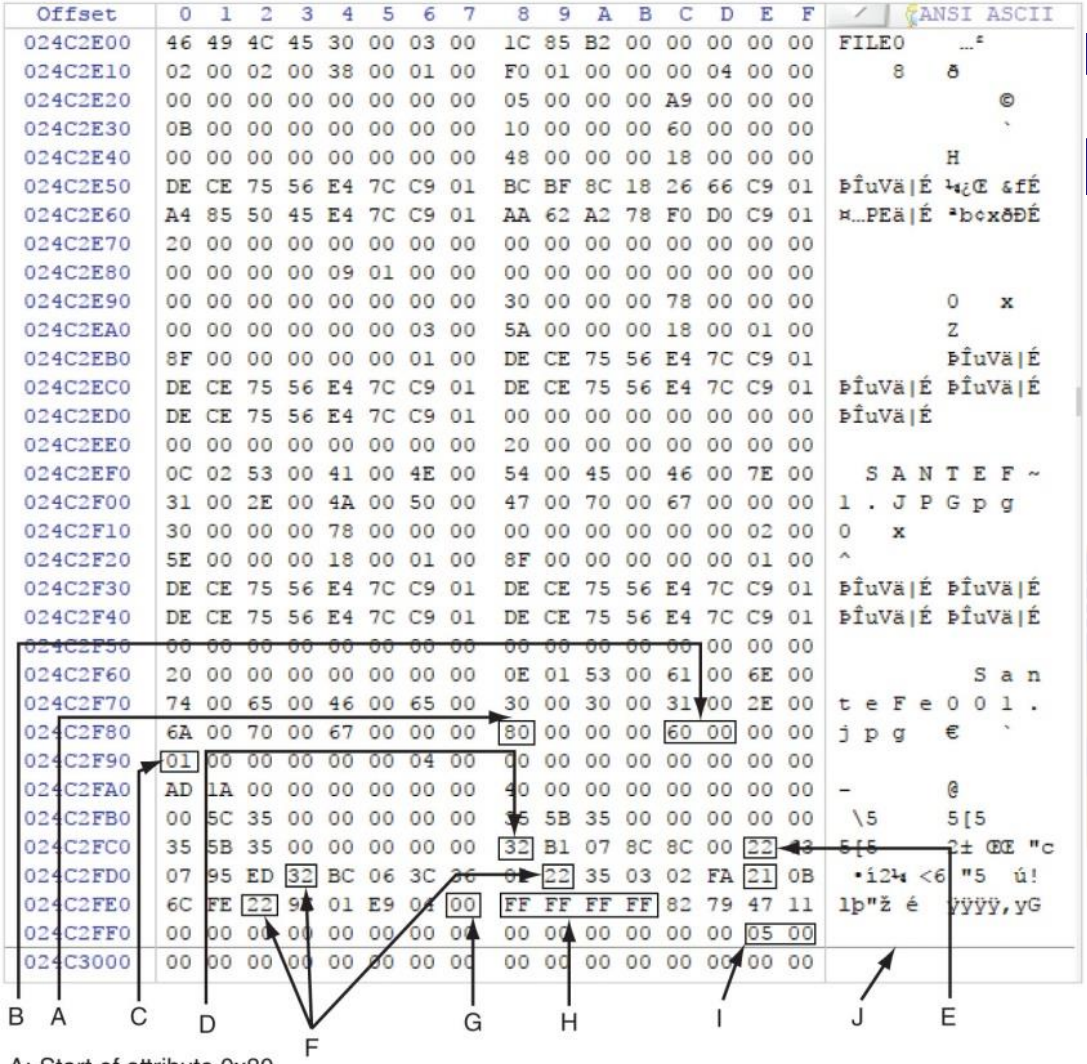


Figure 5-17 Attribute 0x80: Data for a resident file

Source: X-Ways AG, *www.x-ways.net*

Data (7 of 7)



Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	ANSI ASCII
024C2E00	46	49	4C	45	30	00	03	00	1C	85	B2	00	00	00	00	00	FILE0 …²
024C2E10	02	00	02	00	38	00	01	00	F0	01	00	00	00	04	00	00	8 ð
024C2E20	00	00	00	00	00	00	00	00	05	00	00	00	A9	00	00	00	©
024C2E30	0B	00	00	00	00	00	00	00	10	00	00	00	60	00	00	00	`
024C2E40	00	00	00	00	00	00	00	00	48	00	00	00	18	00	00	00	H
024C2E50	DE	CE	75	56	E4	7C	C9	01	BC	BF	8C	18	26	66	C9	01	ÞÎuVä|É ¼¿Œ &fÉ
024C2E60	A4	85	50	45	E4	7C	C9	01	AA	62	A2	78	F0	D0	C9	01	¤…PEä|É ªb¢xðÐÉ
024C2E70	20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
024C2E80	00	00	00	00	09	01	00	00	00	00	00	00	00	00	00	00	
024C2E90	00	00	00	00	00	00	00	00	30	00	00	00	78	00	00	00	0 x
024C2EA0	00	00	00	00	00	00	03	00	5A	00	00	00	18	00	01	00	Z
024C2EB0	8F	00	00	00	00	00	01	00	DE	CE	75	56	E4	7C	C9	01	ÞÎuVä|É
024C2EC0	DE	CE	75	56	E4	7C	C9	01	DE	CE	75	56	E4	7C	C9	01	ÞÎuVä|É ÞÎuVä|É
024C2ED0	DE	CE	75	56	E4	7C	C9	01	00	00	00	00	00	00	00	00	ÞÎuVä|É
024C2EE0	00	00	00	00	00	00	00	00	20	00	00	00	00	00	00	00	
024C2EF0	0C	02	53	00	41	00	4E	00	54	00	45	00	46	00	7E	00	S A N T E F ~
024C2F00	31	00	2E	00	4A	00	50	00	47	00	70	00	67	00	00	00	1 . J P G p g
024C2F10	30	00	00	00	78	00	00	00	00	00	00	00	00	00	02	00	0 x
024C2F20	5E	00	00	00	18	00	01	00	8F	00	00	00	00	00	01	00	^
024C2F30	DE	CE	75	56	E4	7C	C9	01	DE	CE	75	56	E4	7C	C9	01	ÞÎuVä|É ÞÎuVä|É
024C2F40	DE	CE	75	56	E4	7C	C9	01	DE	CE	75	56	E4	7C	C9	01	ÞÎuVä|É ÞÎuVä|É
024C2F50	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
024C2F60	20	00	00	00	00	00	00	00	0E	01	53	00	61	00	6E	00	S a n
024C2F70	74	00	65	00	46	00	65	00	30	00	30	00	31	00	2E	00	t e F e 0 0 1 .
024C2F80	6A	00	70	00	67	00	00	00	80	00	00	00	60	00	00	00	j p g € `
024C2F90	01	00	00	00	00	00	04	00	00	00	00	00	00	00	00	00	
024C2FA0	AD	1A	00	00	00	00	00	00	40	00	00	00	00	00	00	00	- @
024C2FB0	00	5C	35	00	00	00	00	00	35	5B	35	00	00	00	00	00	\5 5[5
024C2FC0	35	5B	35	00	00	00	00	00	32	B1	07	8C	8C	00	22	43	5[5 2± ŒŒ "C
024C2FD0	07	95	ED	32	BC	06	3C	36	00	22	35	03	02	FA	21	0B	•í2¼ <6 "5 ú!
024C2FE0	6C	FE	22	9A	01	E9	04	00	FF	FF	FF	FF	82	79	47	11	lþ"š é ÿÿÿÿ‚yG
024C2FF0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	05	00	
024C3000	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	

B A C D F G H I J E

A: Start of attribute 0x80
B: Size of attribute 0x80
C: Nonresident flag
D: First data run
E: Second data run
F: Additional data runs
G: End of data run
H: End of MFT record
I: Sector checksum
J: Sector boundary

Figure 5-18 Attribute 0x80: Data for a nonresident file

Source: X-Ways AG, *www.x-ways.net*

NTFS Alternate Data Streams (1 of 2)

- **Alternate data streams**
  - Ways data can be appended to existing files
  - Can obscure valuable evidentiary data, intentionally or by coincidence
- In NTFS, an alternate data stream becomes an additional file attribute
  - **Allows the file to be associated with different applications**
- You can only tell whether a file has a data stream attached by examining that file's MFT entry

NTFS Alternate Data Streams (2 of 2)

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	ANSI ASCII
024C0E00	46	49	4C	45	30	00	03	00	5C	AC	B2	00	00	00	00	00	FILE0 \¬²
024C0E10	04	00	01	00	38	00	00	00	B8	01	00	00	00	04	00	00	8 ¸
024C0E20	00	00	00	00	00	00	00	00	07	00	00	00	A1	00	00	00	¡
024C0E30	09	00	00	00	00	00	00	00	10	00	00	00	60	00	00	00	`
024C0E40	00	00	00	00	00	00	00	00	48	00	00	00	18	00	00	00	H
024C0E50	B2	4C	07	BB	D3	7C	C9	01	14	4A	A3	3D	C8	7D	C9	01	²L »Ó|É J£=È}É
024C0E60	14	4A	A3	3D	C8	7D	C9	01	60	69	A5	EC	F8	D0	C9	01	J£=È}É `i¥ìøÐÉ
024C0E70	20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
024C0E80	00	00	00	00	09	01	00	00	00	00	00	00	00	00	00	00	
024C0E90	00	00	00	00	00	00	00	00	30	00	00	00	70	00	00	00	0 p
024C0EA0	00	00	00	00	00	00	02	00	54	00	00	00	18	00	01	00	T
024C0EB0	8A	00	00	00	00	00	01	00	B2	4C	07	BB	D3	7C	C9	01	Š ²L »Ó|É
024C0EC0	B2	4C	07	BB	D3	7C	C9	01	B2	4C	07	BB	D3	7C	C9	01	²L »Ó|É ²L »Ó|É
024C0ED0	B2	4C	07	BB	D3	7C	C9	01	00	00	00	00	00	00	00	00	²L »Ó|É
024C0EE0	00	00	00	00	00	00	00	00	20	00	00	00	00	00	00	00	
024C0EF0	09	03	42	00	46	00	31	00	5F	00	34	00	2E	00	74	00	B F 1 _ 4 . t
024C0F00	78	00	74	00	00	00	00	00	80	00	00	00	50	00	00	00	x t € P
024C0F10	00	00	18	00	00	00	05	00	38	00	00	00	18	00	00	00	8
024C0F20	57	65	6C	6C	20	64	6F	6E	65	20	69	73	20	62	65	74	Well done is bet
024C0F30	74	65	72	20	74	68	61	6E	20	77	65	6C	6C	20	73	61	ter than well sa
024C0F40	69	64	2E	0D	0A	20	20	42	65	6E	6A	61	6D	69	6E	20	id. Benjamin
024C0F50	46	72	61	6E	6B	6C	69	6E	80	00	00	00	58	00	00	00	Franklin€ X
024C0F60	01	06	40	00	00	00	06	00	00	00	00	00	00	00	00	00	@
024C0F70	00	00	00	00	00	00	00	00	50	00	00	00	00	00	00	00	P
024C0F80	00	02	00	00	00	00	00	00	3B	00	00	00	00	00	00	00	;
024C0F90	3B	00	00	00	00	00	00	00	68	00	69	00	64	00	64	00	; h i d d
024C0FA0	65	00	6E	00	00	00	00	00	31	01	31	A2	00	00	21	E4	e n 1 1¢ !ä
024C0FB0	FF	FF	FF	FF	62	79	47	11	00	00	00	00	00	00	00	00	ÿÿÿÿ,yG

Second attribute 0x80

Start of data run for second attribute 0x80 (location of hidden alternate data stream)

Size of second attribute 0x80

Figure 5-24 A text alternate data stream

Source: X-Ways AG, *www.x-ways.net*

NTFS Compressed Files

- NTFS provides compression similar to FAT DriveSpace 3 (a Windows 98 compression utility)
- With 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 and later
  - 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

EFS Recovery Key Agent

- **Recovery Key Agent** implements the recovery certificate
  - Which is in the Windows administrator account
- Windows administrators can recover a key in two ways: through Windows or from a command prompt
- Commands:
  - cipher
  - copy

Deleting NTFS Files

- When a file is deleted in Windows NT and later
  - **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 in the same way FAT does

Resilient File System

- **Resilient File System (ReFS)** - designed to address **very large data** storage needs
  - Such as the cloud
- Features incorporated into ReFS's design:
  - Maximized data **availability**
  - Improved data **integrity**
  - Designed for **scalability**
- ReFS uses disk structures similar to the MFT in NTFS

Understanding Whole Disk Encryption (1 of 3)

- In recent years, there has been more concern about loss of
  - **Personal identity information (PII)** and trade secrets caused by computer theft
- Of particular concern is the theft of laptop computers and handheld devices
- To help prevent loss of information, software vendors now provide **whole disk encryption**

Understanding Whole Disk Encryption (2 of 3)

- Current whole disk encryption tools offer the following features:
  - **Preboot** authentication
  - Full or partial disk encryption with secure hibernation
  - Advanced encryption algorithms
  - Key management function

Understanding Whole Disk Encryption (3 of 3)

- Whole disk encryption tools encrypt each sector of a drive separately
- Many of these tools encrypt the drive's boot sector
  - To prevent any efforts to bypass the secured drive's partition
- To examine an encrypted drive, decrypt it first
  - Run a vendor-specific program to decrypt the drive
  - Many vendors use a bootable CD or USB drive that prompts for a one-time passphrase

Examining Microsoft BitLocker

- Available Vista Enterprise/Ultimate, Windows 7, 8, and 10 Professional/Enterprise, and Server 2008 and later
- Hardware and software requirements
- A computer capable of running Windows Vista or later
- The TPM microchip, version 1.2 or newer
- A computer BIOS compliant with Trusted Computing Group (TCG)
- Two NTFS partitions
- The BIOS configured so that the hard drive boots first before checking other bootable peripherals

Examining Third-Party Disk Encryption Tools

- Some available third-party WDE utilities:
  - Endpoint Encryption
  - Voltage SecureFile
  - Jetico BestCrypt Volume Encryption

Understanding the Windows Registry

- **Registry**
  - A database that stores hardware and *software configuration information*, **network connections, user preferences**, and setup information
- To view the Registry, you can use:
  - Regedit (Registry Editor) program for Windows 9x systems
  - Regedt32 for Windows 2000, XP, and Vista
  - Both utilities can be used for Windows 7 and 8

Exploring the Organization of the Windows Registry (1 of 5)

- Registry terminology:
  - Registry
  - Registry Editor
  - HKEY
  - Key
  - Subkey
  - Branch
  - Value
  - Default value
  - Hives

Exploring the Organization of the Windows Registry (2 of 5)

Table 5-6 Registry file locations and purposes

Filename and location	Purpose of file
Users*user-account*\Ntuser.dat	User-protected storage area; contains the list of most recently used files and desktop configuration settings
Windows\system32\config\Default.dat	Contains the computer's system settings
Windows\system32\config\SAM.dat	Contains user account management and security settings
Windows\system32\config\Security.dat	Contains the computer's security settings
Windows\system32\config\Software.dat	Contains installed programs' settings and associated usernames and passwords
Windows\system32\config\System.dat	Contains additional computer system settings
Windows\system32\config\systemprofile	Contains additional NTUSER information

Exploring the Organization of the Windows Registry (3 of 5)

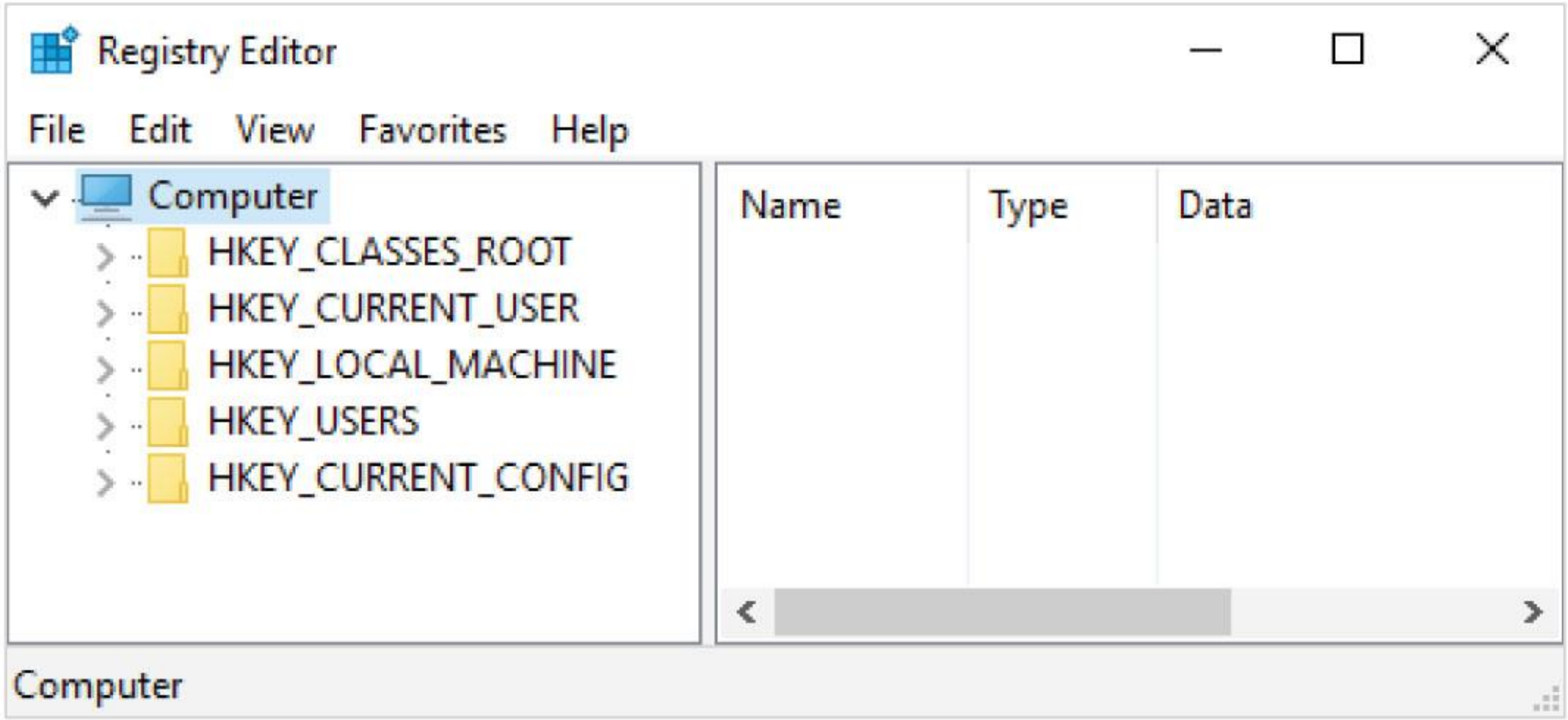


Figure 5-26 Viewing HKEYs in Registry Editor

Exploring the Organization of the Windows Registry (4 of 5)

Table 5-7 Registry HKEYs and their functions

HKEY	Function
HKEY_CLASSES_ROOT	A symbolic link to HKEY_LOCAL_MACHINE\SOFTWARE\Classes; provides file type and file extension information, URL protocol prefixes, and so forth
HKEY_CURRENT_USER	A symbolic link to HKEY_USERS; stores settings for the currently logged-on user
HKEY_LOCAL_MACHINE	Contains information about installed hardware and software
HKEY_USERS	Stores information for the currently logged-on user; only one key in this HKEY is linked to HKEY_CURRENT_USER

(*continues*)

Exploring the Organization of the Windows Registry (5 of 5)

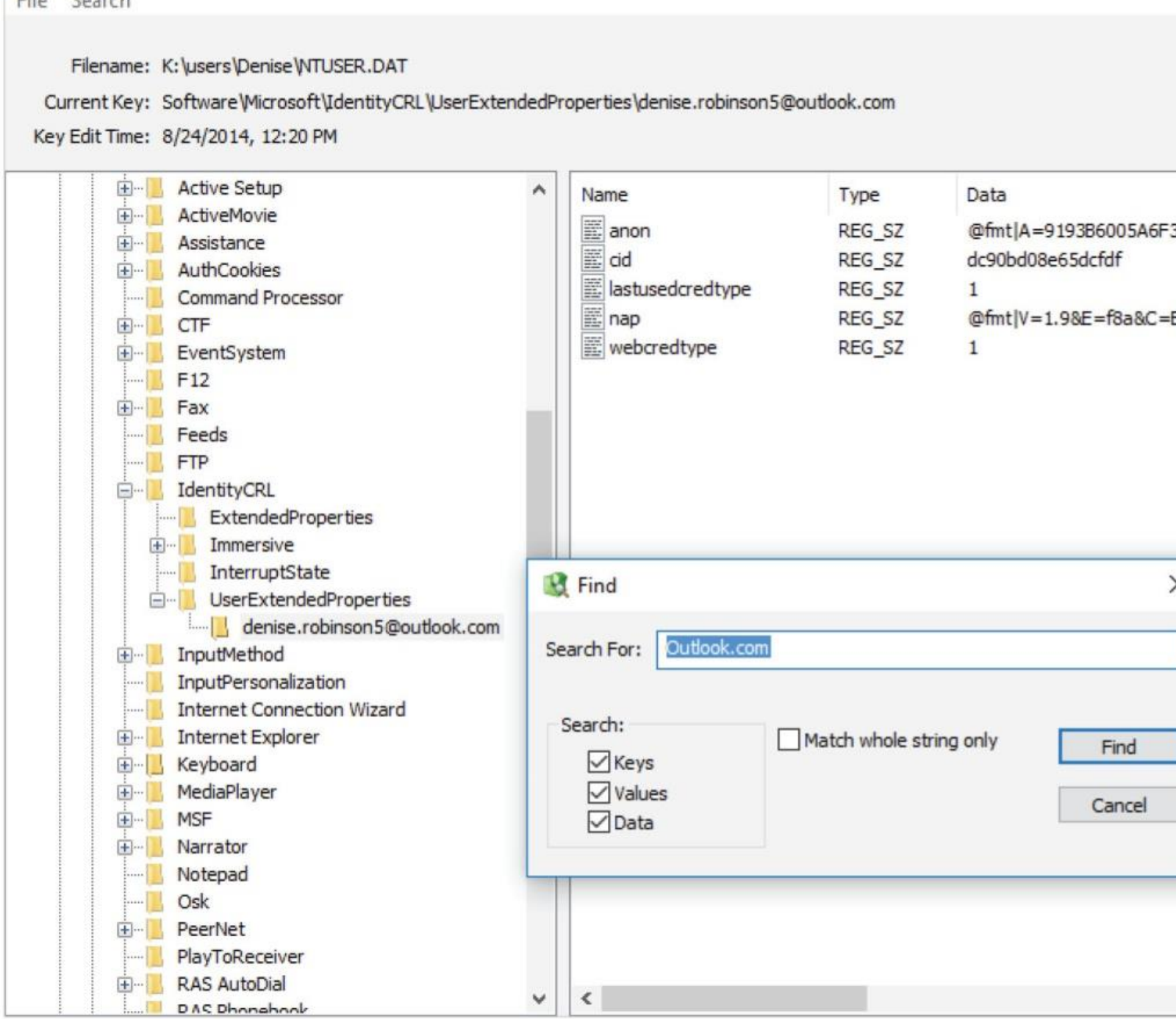
Table 5-7 Registry HKEYs and their functions (*continued*)

HKEY	Function
HKEY_CURRENT_CONFIG	A symbolic link to HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Hardware ProfileVxxxx (with xxxx representing the current hardware profile); contains hardware configuration settings
HKEY_DYN_DATA	Used only in Windows 9x/Me systems; stores hardware configuration settings

Examining the Windows Registry (1 of 2)

- Tools with built-in or add-on Registry viewers:
  - X-Ways Forensics
  - OSForensics
  - Forensic Explorer
  - FTK

Examining the Windo



Understanding Microsoft Startup Tasks

- Learn what files are accessed when Windows starts
- This information helps you determine when a suspect's computer was last accessed
  - Important with computers that might have been used after an incident was reported

Summary (1 of 3)

- When starting a suspect's computer, using boot media, such as forensic boot CDs or USB drives, you must ensure that disk evidence isn't altered
- The Master Boot Record (MBR) stores information about partitions on a disk
- Microsoft used FAT12 and FAT16 on older operating systems
- To find a hard disk's capacity, use the cylinders, heads, and sectors (CHS) calculation

Summary (2 of 3)

- When files are deleted in a FAT file system, the hexadecimal value 0x05 is inserted in the first character of the filename in the directory
- NTFS is more versatile because it uses the Master File Table (MFT) to track file information
- Records in the MFT contain attribute IDs that store metadata about files
- In NTFS, alternate data streams can obscure information that might be of evidentiary value

Summary (3 of 3)

- File slack, RAM slack, and drive slack are areas in which valuable information can reside on a drive
- NTFS can encrypt data with EFS and BitLocker
- NTFS can compress files, folders, or volumes
- Windows Registry keeps a record of attached hardware, user preferences, network connections, and installed software
- Virtualization software enables you to run other OSs on a host computer