Guide to Computer Forensics and Investigations Sixth Edition

Chapter 5

Working with Windows and CLI Systems

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

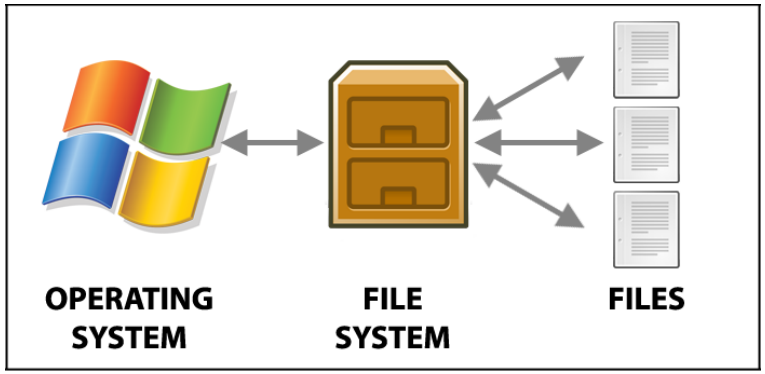
© Cengage Learning 2018

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



https://www.cleverfiles.com

© Cengage Learning 2018

Understanding the Boot Sequence

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

- **Bootstrap process**
  - Contained in ROM, tells the computer how to proceed
  - Displays the key or keys you press to open the CMOS setup screen.
    - *Besides date and time, BIOS settings are stored in CMOS*

- CMOS should be modified to boot from a forensic floppy disk or CD
  - *To prevent evidence from hard disk being overwritten*

© Cengage Learning 2018

Understanding the Boot Sequence (Cont)

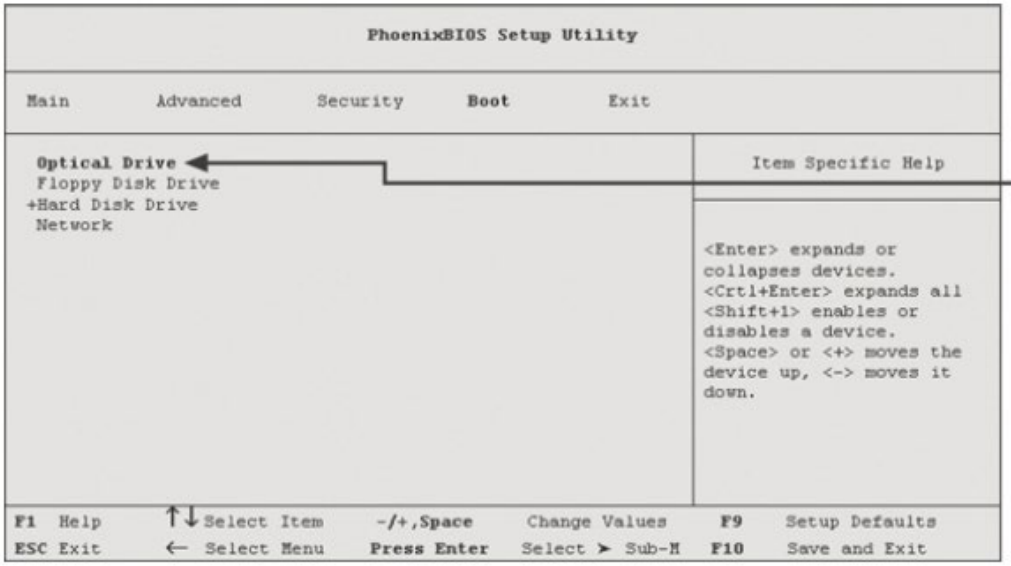


Figure 5-1 A typical CMOS setup screen
Courtesy of Phoenix Technologies, Ltd.

BIOS can configure:-

- *System Time/Date*
- *Boot Sequence*
- *Plug and Play*
- *Mouse/Keyboard*
- *Drive Configuration*
- *Security*
- *Power Management*
- *etc…*

© Cengage Learning 2018

Understanding Disk Drives

- Disk drives are made up of one or more platters coated with magnetic material
- Disk drive components
  - Geometry (*how data is structured*)
  - Head
  - Tracks
  - Cylinders
  - Sectors
    - Typically one sector has 512 Bytes

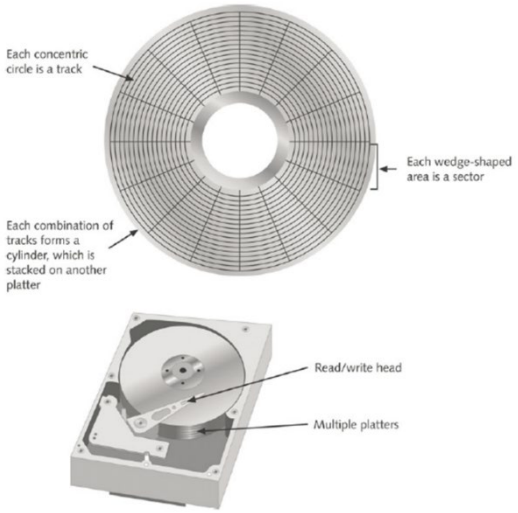
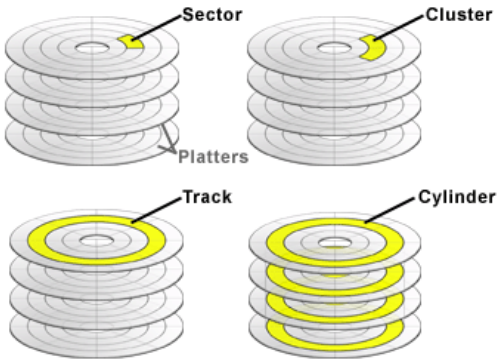


Figure 5-2 Components of a disk drive
© Cengage Learning®



https://stackoverflow.com/questions/32642016/chs-to-lba-mapping-disk-storage

© Cengage Learning 2018

Understanding Disk Drives (Cont)

- Properties handled at the drive's hardware or firmware level include:-
  - Zone bit recording (ZBR)
    - *Grouping tracks by zones ensures that all tracks hold the same amount of data.*
  - Track density
    - *Space between each track. The smaller the space, the more track on platter*
  - Areal density
    - *Number of bits in one square inch of a disk platter*
  - Head and cylinder skew
    - *Improve disk performance by minimize the movement of read/write head*

© Cengage Learning 2018

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
    - *In general, memory cells can perform 10k – 100k reads/writes*

- 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
    - wear-leveling feature automatically overwrites the unallocated space

© Cengage Learning 2018

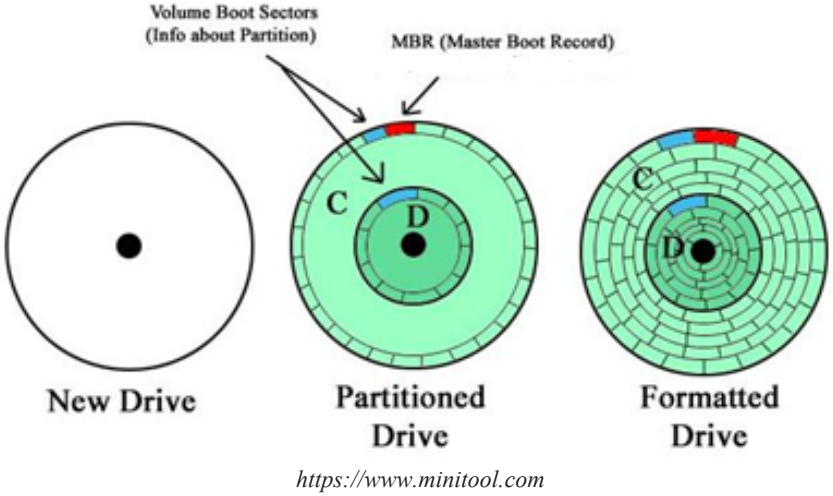
Exploring Microsoft File Structures

- In Microsoft file structures, sectors are grouped to form **clusters**
  - Storage allocation units of one or more sectors
- Clusters can have one sector or more
- Combining sectors minimizes the overhead of writing or reading files to a disk
- 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**
  - *Address point to relative position*
- Sector numbers are called **physical addresses**
  - *From address 0 to last sector on disk*
- Clusters and their addresses are specific to a logical disk drive, which is a disk partition

© Cengage Learning 2018

Disk Partitions

- A **partition** is a logical drive. *i.e "C", "D", "E" and etc*
- Windows OSs can have three primary partitions followed by an extended partition that can contain one or more logical drives



https://www.minitool.com

- **Partition gap**
  - Unused space between partitions
  - ***Can use to hide data!***

© Cengage Learning 2018

Disk Partitions (Cont)

Table 5-1 Hexadecimal codes in the partition table

Hexadecimal code	File system
01	DOS 12-bit FAT (floppy disks)
04	DOS 16-bit FAT for partitions smaller than 32 MB
05	Extended partition
06	DOS 16-bit FAT for partitions larger than 32 MB
07	NTFS and exFAT
08	AIX bootable partition
09	AIX data partition
0B	DOS 32-bit FAT
0C	DOS 32-bit FAT for interrupt 13 support
0F	Extended Partition with Logical Block Address (LBA)
17	Hidden NTFS partition (XP and earlier)
1B	Hidden FAT32 partition
1E	Hidden VFAT partition
3C	Partition Magic recovery partition
66–69	Novell partitions
81	Linux
82	Linux swap partition (can also be associated with Solaris partitions)
83	Linux native file systems (Ext2, Ext3, Ext4, Reiser, Xiafs)
86	FAT16 volume/stripe set (Windows NT)
87	High Performance File System (HPFS) fault-tolerant mirrored partition or NTFS volume/stripe set
A5	FreeBSD and BSD/386
A6	OpenBSD
A9	NetBSD
C7	Typical of a corrupted NTFS volume/stripe set
EB	BeOS

© 2015 Cengage Learning®

*A **partition table** is a **table** maintained on disk by the operating system describing the **partitions** on that disk.*

***key hexadecimal codes** is used by OS to identify and maintain the file system.*

© Cengage Learning 2018

Disk Partitions (Cont)

- The **partition table** is in the **Master Boot Record (MBR)**
  - Located at **sector 0** of the disk drive, preceding the first partition.
- **MBR** stores information about partitions on a disk and their locations, size, and other important items

- In a hexadecimal editor, such as WinHex, you can find the first partition at **offset 0x1BE**
  - **File system's hexadecimal code** is offset 3 bytes from 0x1BE for the first partition
  - **Sector address** of where this partition starts on the drive is offset 8 bytes from 0x1BE.
  - **The number of sectors** assigned to the partition are offset 12 bytes from position 0x1BE.

© Cengage Learning 2018

Disk Partitions (Cont)

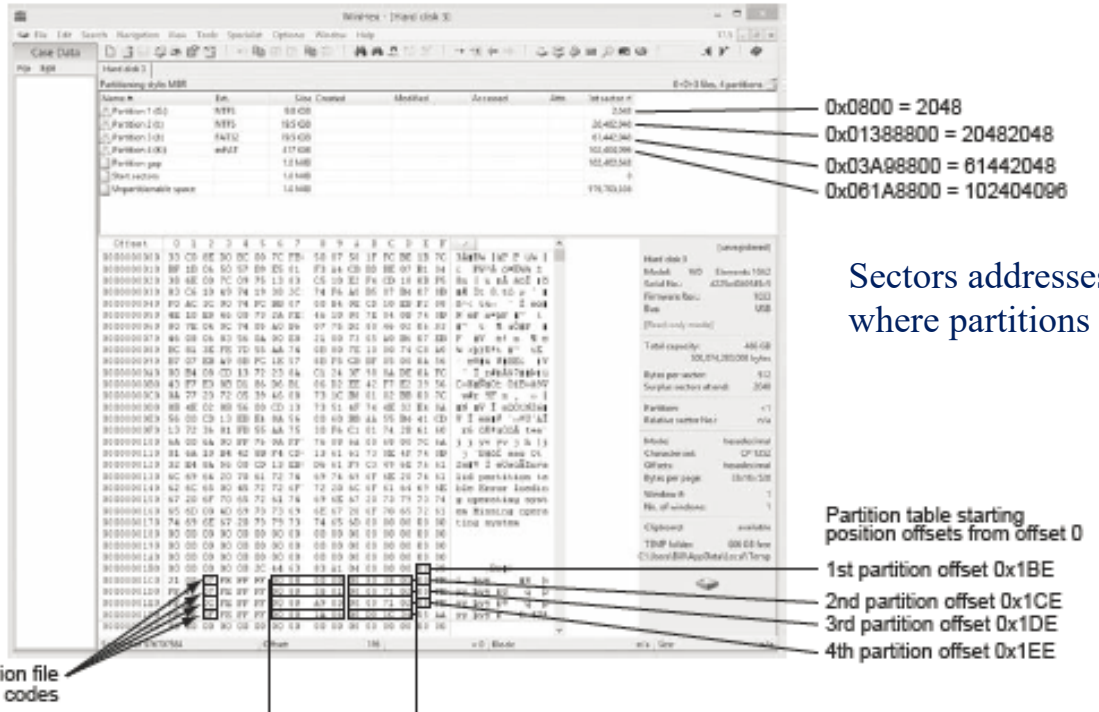


Figure 5-4 The partition table in a hexidecimal editor

© Cengage Learning 2018

Examining FAT Disks

- **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
- Four current FAT versions
  - FAT12, FAT16, FAT32, and exFAT (used by Xbox game systems)

© Cengage Learning 2018

Examining FAT Disks (Cont)

- Cluster sizes vary according to disk drive size and file system

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 (1024)
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

© Cengage Learning®

Note : sector size = 512 bytes

© Cengage Learning 2018

Examining FAT Disks (Cont)

- 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** (*portion of the last sector used in the last assigned cluster*) and
  - **File slack** (*unused space allocated for a file*)

- An unintentional side effect of FAT16 having large clusters was that it reduced fragmentation
  - As cluster size increased

© Cengage Learning 2018

Examining FAT Disks (Cont)

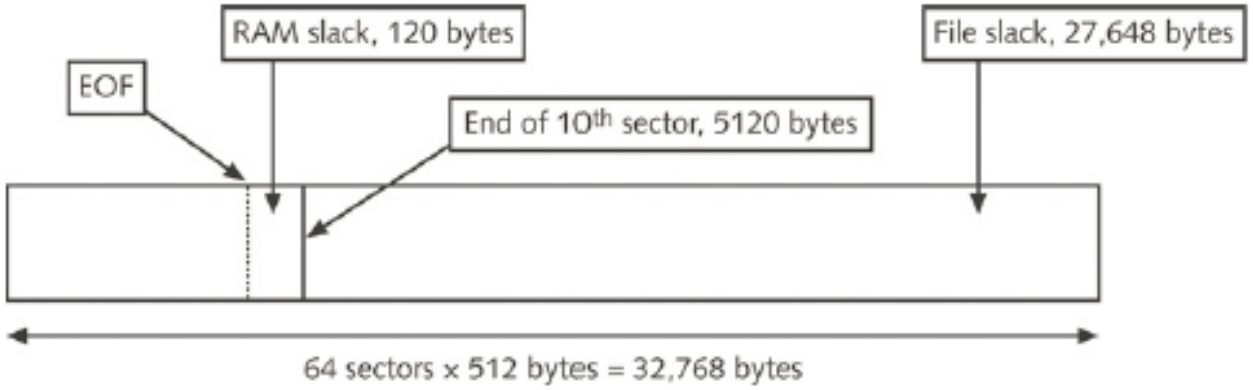


Figure 5-8 File slack space
© 2015 Cengage Learning®

- *Document size is 5000 bytes of data.*
- *Assuming hard disk size is 1.6G, the OS allocates about 32,000 bytes, or 64 sectors (512 bytes per sector), for this file.*
- *The unused space, 27,000 bytes, is the file slack.*
- *RAM slack is the portion of the last sector (10th sector) used in the last assigned cluster, and the remaining sectors are referred to as "file slack".*
- *See diagram above*

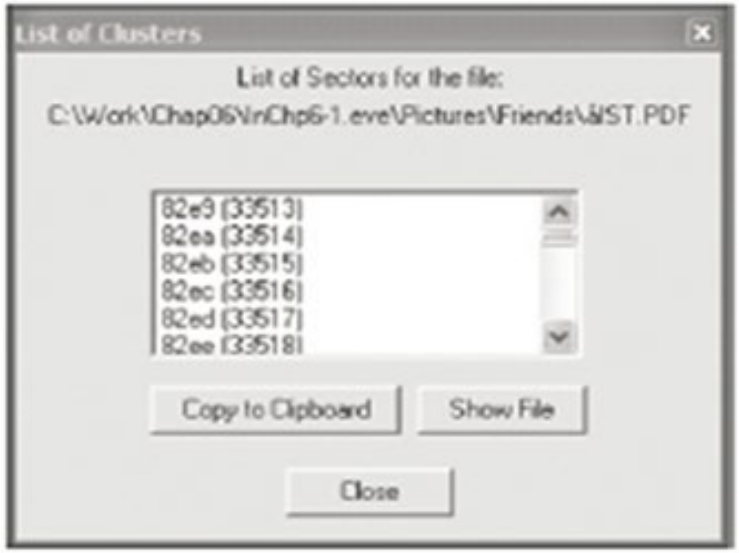
Note : The data used to fill the 120-byte void (RAM Slack) is pulled from RAM. Any information in RAM at that point, such as logon IDs or passwords, is placed in RAM slack!!

© Cengage Learning 2018

Examining FAT Disks (Cont)

- 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 *due to deleted files or expansion files*
- 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

Using Prodiscover



Chained sectors associated with clusters as a result of increasing file size

© Cengage Learning 2018

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

© Cengage Learning 2018

Examining NTFS Disks

- **NT File System (NTFS)**
  - Introduced with Windows NT
  - Primary file system for Windows 8 or later
- 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. i.e deleting a file

http://ipkonfig.com

© Cengage Learning 2018

Examining NTFS Disks (Cont)

- 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

© Cengage Learning 2018

Examining NTFS Disks (Cont)

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

© 2015 Cengage Learning®

Note : sector size = 512 bytes

© Cengage Learning 2018

NTFS System Files

- Master File Table (MFT) contains information about all files on the disk
  - Including the system files OS uses
- In the MFT, the first 15 records are reserved for system files
- Records in the MFT are called **metadata**

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.
$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	Cluster bitmap	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.

© 2015 Cengage Learning®

© Cengage Learning 2018

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

© Cengage Learning 2018

MFT and File Attributes (Cont)

- Each MFT record starts with a header identifying it as a **resident** or **nonresident** attribute
- Files larger than 512 bytes (*nonresident*) 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**
- For very small files, about 512 bytes or less (*resident*), all file metadata and data are stored in the MFT record.

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. Long filenames have two attribute ID 0x30s in the MFT record: one for the short name and one for the long name.
0x40	$Object_ID ($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.

*Basic information of a file in MFT starts at **0x10***

© Cengage Learning 2018

MFT and File Attributes (Cont)

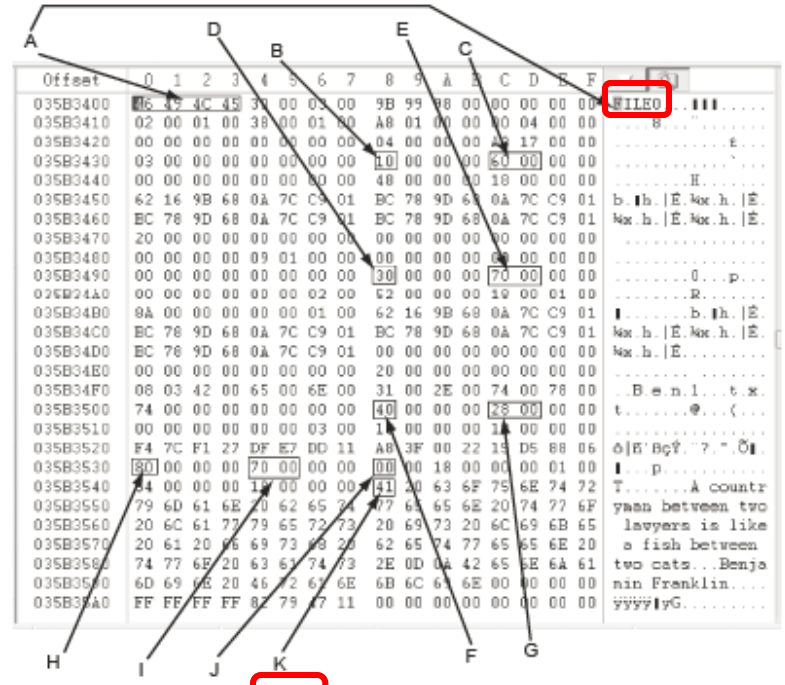
Table 5-5 Attributes in the MFT (*continued*)

Attribute ID	Purpose
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.
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

© 2015 Cengage Learning®

© Cengage Learning 2018

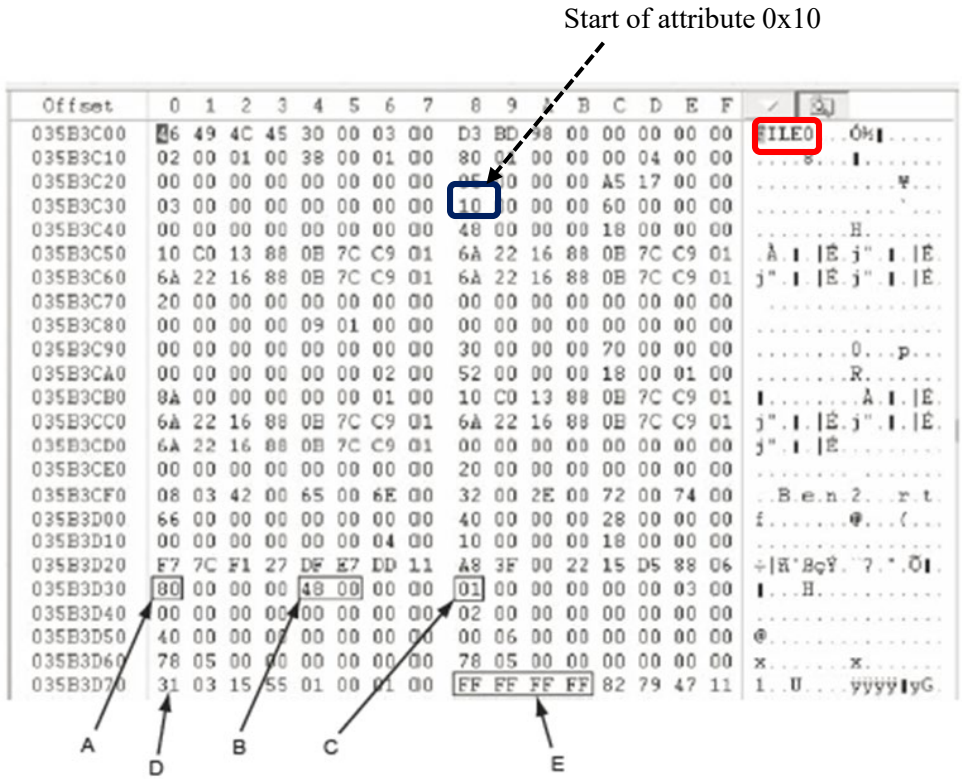
MFT and File Attributes (Cont)



A: All MFT records start with FILE0
B: Start of attribute 0x10
C: Length of attribute 0x10 (value 60) **60 00 → 00 60**
D: Start of attribute 0x30
E: Length of attribute 0x30 (value 70) **70 00 → 00 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
Courtesy of X-Ways AG, *www.x-ways.net*

Resident file attributes



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

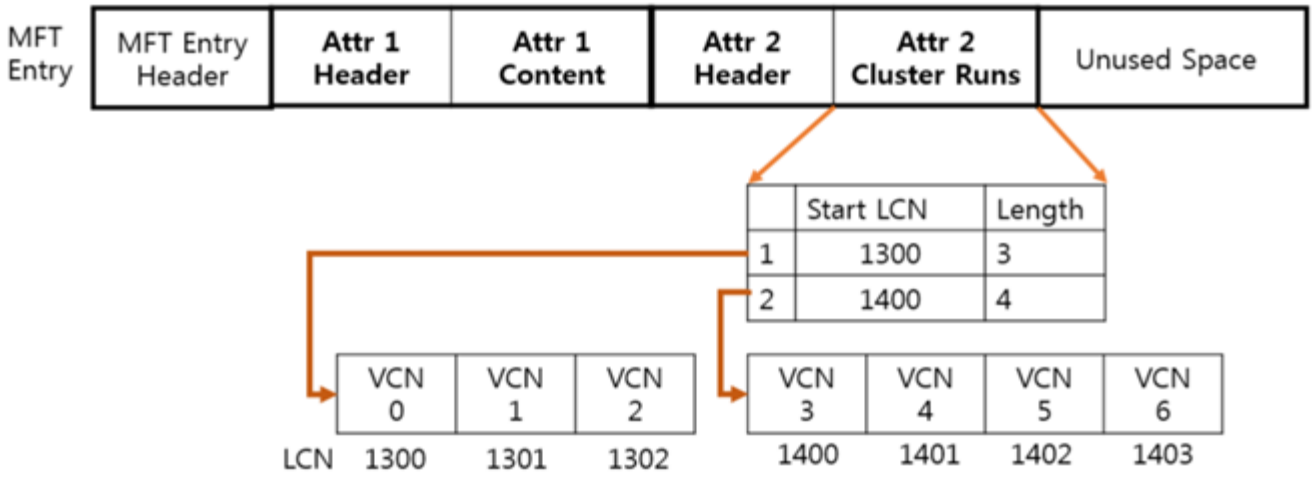
Figure 5-12 Nonresident file in an MFT record
Courtesy of X-Ways AG, *www.x-ways.net*

Non-resident file attributes

© Cengage Learning 2018

MFT and File Attributes (Cont)

- 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)**



https://m.blog.naver.com

© Cengage Learning 2018

MFT Structures for File Data

- 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 0x14* - length of the header (indicates where the next attribute starts) *38 00 → 00 38 = 56 bytes!!*
  - *At offset 0x1C to 0x1F* - size of the MFT record
  - *At offset 0x32 and 0x33* - the update sequence array, which stores the last 2 bytes of the first sector of the MFT record
    - *The update sequence array is used as a checksum for record integrity validation*

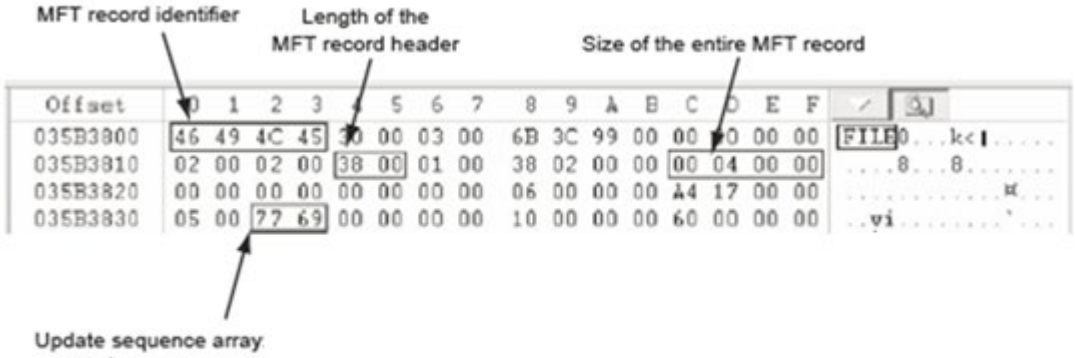


Figure 5-13 An MFT header
Courtesy of X-Ways AG, *www.x-ways.net*

© Cengage Learning 2018

MFT Structures for File Data (cont)

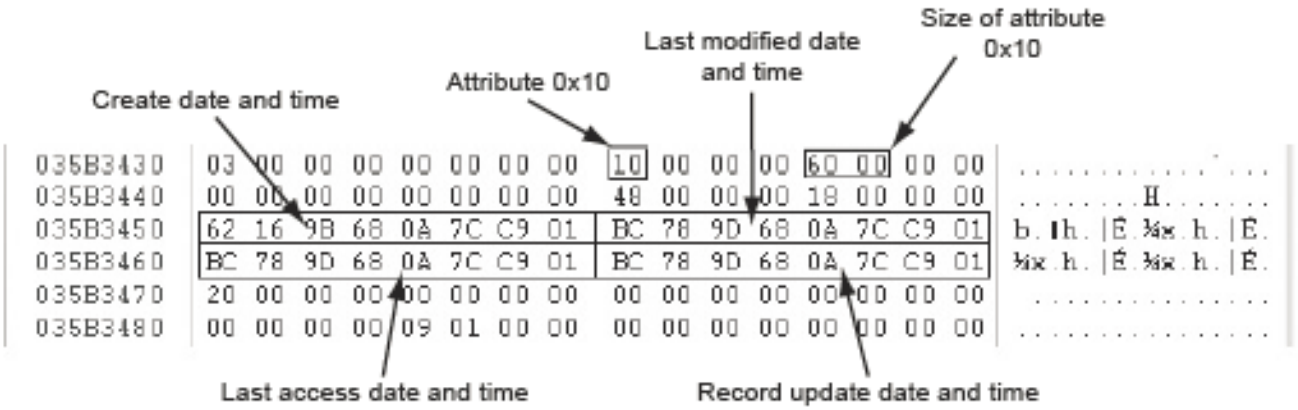


Figure 5-14 Attribute 0x10: Standard Information
Courtesy of X-Ways AG, *www.x-ways.net*

Standard Information attribute –

- *Create date and time*
- *Last modified date and time*
- *Last access date and time*
- *Record update date and time*

© Cengage Learning 2018

NTFS Alternate Data Streams

- **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

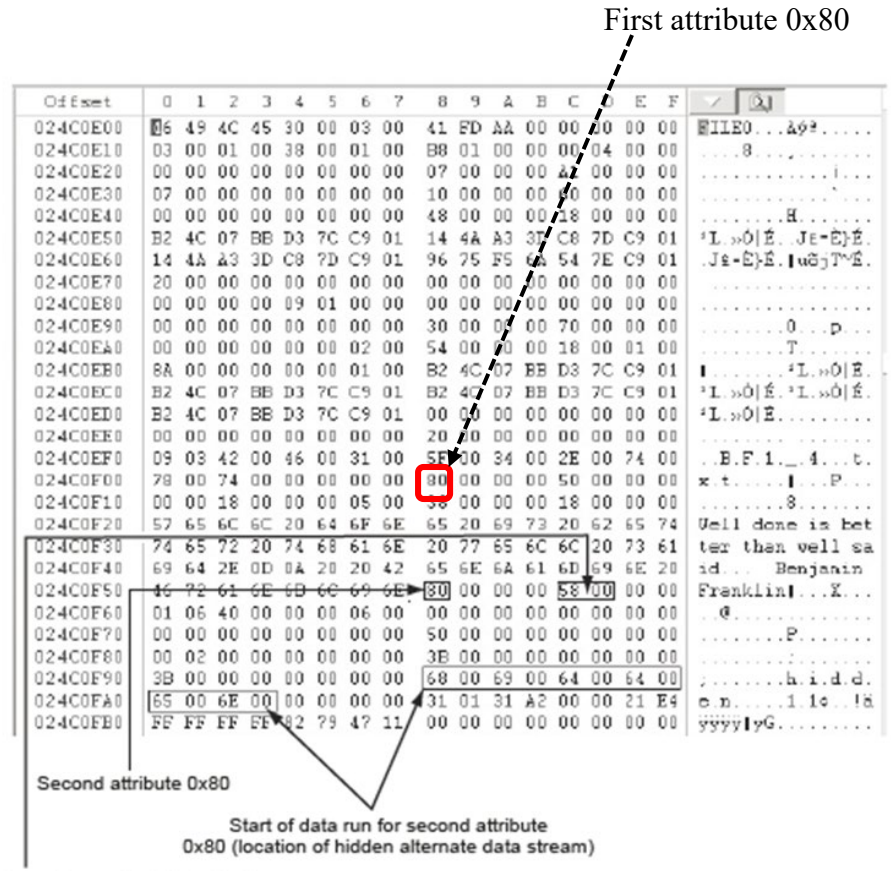


Figure 5-24 A text alternate data stream
Courtesy of X-Ways AG, *www.x-ways.net*

© Cengage Learning 2018

NTFS Compressed Files

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

http://www.penguincoders.net

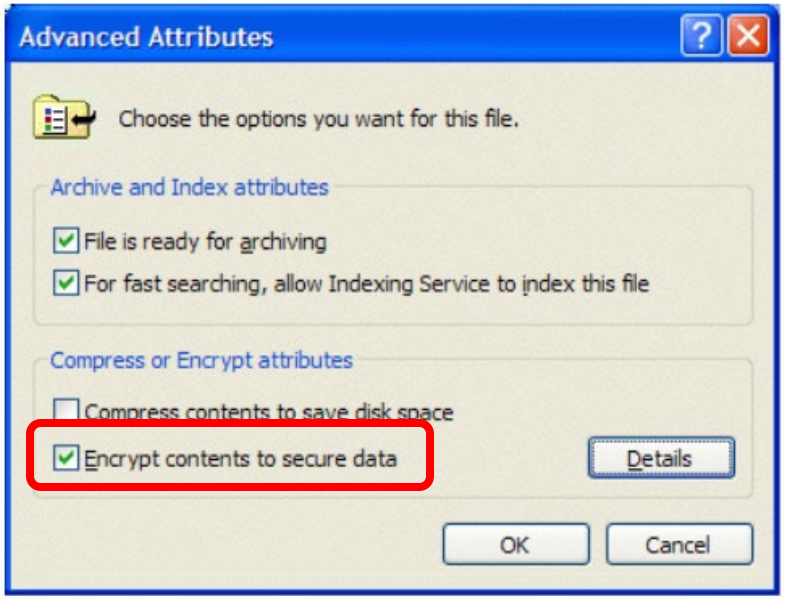
© Cengage Learning 2018

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
    - *Recovery certificate is required if there is a problem*

- Users can apply **EFS** to files stored on their local workstations or a remote server



http://ntfs.com

© Cengage Learning 2018

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 an MS-DOS command prompt
- MS-DOS commands
  - Cipher (*for NTSF only*)
  - copy
  - efsrecvr (used to decrypt EFS files. *For NTSF only*)
  - *Use <command> /? to find out more*

© Cengage Learning 2018

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 to delete file too
  - *This method does not rename and move deleted file to recycle bin but* eliminates the file from the MFT listing in the same way FAT does

© Cengage Learning 2018

Understanding Whole Disk Encryption (WDE)

- 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 other handheld devices

- To help prevent loss of information, software vendors now provide **whole disk encryption**

© Cengage Learning 2018

Understanding Whole Disk Encryption (WDE) - Feature

- Current whole disk encryption tools offer the following features:
  - Preboot authentication
    - *single sign-on password, fingerprint scan*
  - Full or partial disk encryption with secure hibernation
    - *Password protected screen saver*
  - Advanced encryption algorithms
    - i.e AES or IDEA
  - Key management function
    - *Passphrase to reset password*

© Cengage Learning 2018

Understanding Whole Disk Encryption (Cont)

- **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**

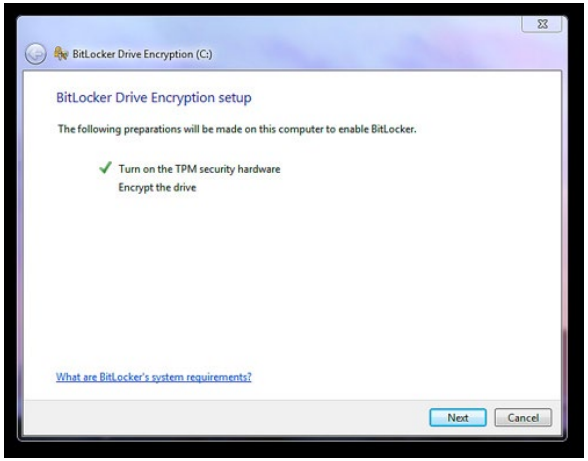
- Note : *Without the necessary credentials to unlock an encrypted drive, it is not possible to view any logical level information.*

© Cengage Learning 2018

Examining Microsoft BitLocker (example)

- Available Vista Enterprise/Ultimate, Windows 7 and 8 Professional/Enterprise, and Server 2008, 2012 and 2016.

- Hardware and software requirements
  - A computer capable of running Windows Vista or later
  - The **Trusted Platform Module**(TPM) microchip, version 1.2 or newer
    - *stores RSA encryption keys specific to the host system for hardware authentication.*
  - 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

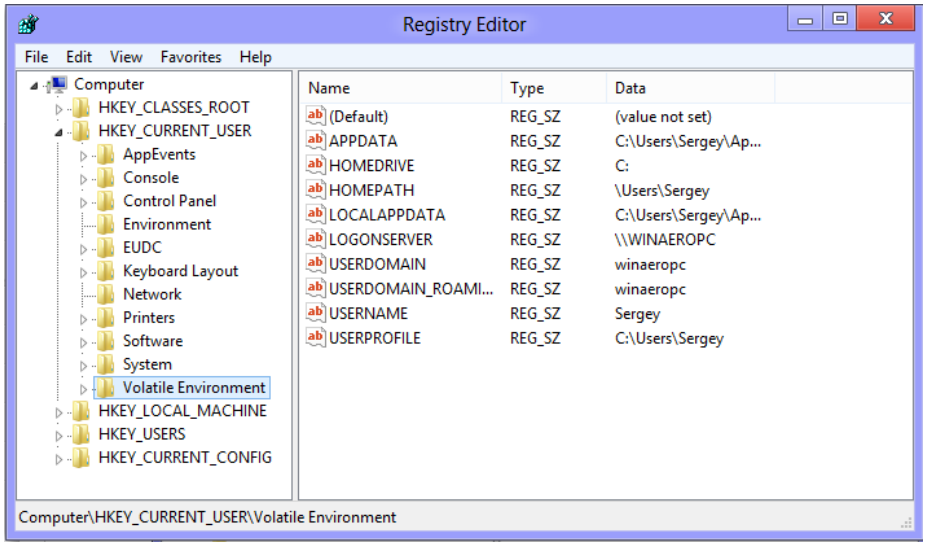


© Cengage Learning 2018

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 later.



https://winaero.com

© Cengage Learning 2018

Exploring the Organization of the Windows Registry

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

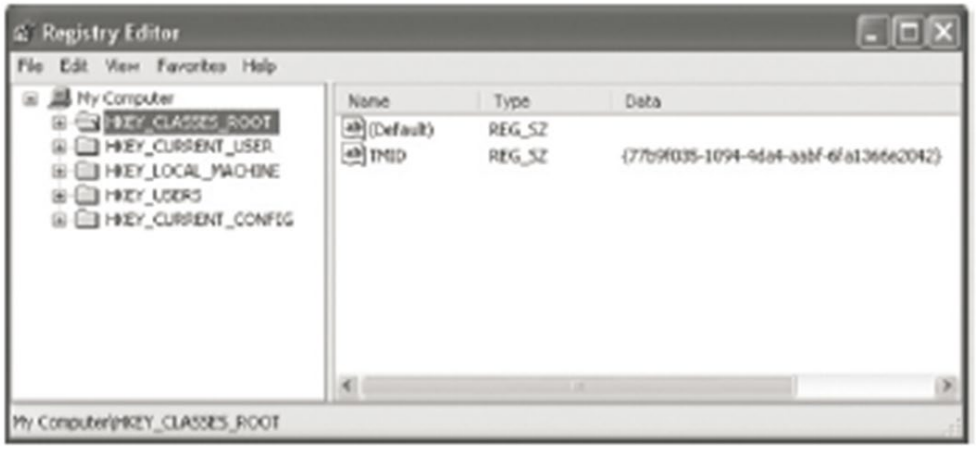


Figure 5-26 Viewing HKEYs in Registry Editor
Courtesy of Microsoft Corporation

Check out : Guide to Computer Forensics and Investigations-Course Technology 5th edition for details

© Cengage Learning 2018

Exploring the Organization of the Windows Registry

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

© 2015 Cengage Learning®

© Cengage Learning 2018

Exploring the Organization of the Windows Registry (Cont)

Table 5-7 Registry HKEYs and their functions

HKEY	Function
HKEY_CLASS_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
HKEY_CURRENT_CONFIG	A symbolic link to HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Hardware Profile*xxxx* (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

© 2015 Cengage Learning®

© Cengage Learning 2018

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

© Cengage Learning 2018

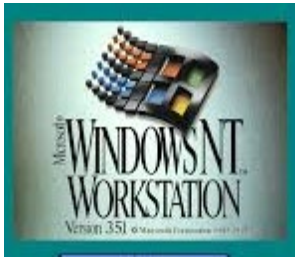
Startup in Windows 7 and Windows 8

- Windows 7/8 is a multiplatform OS
  - Can run on desktops, laptops, tablets, and smartphones
- The boot process uses a boot configuration data (BCD) store
- The BCD contains the boot loader that initiates the system's bootstrap process
  - Press F8 or F12 when the system starts to access the Advanced Boot Options, *ie. Start in Safe Mode*

© Cengage Learning 2018

Startup in Windows NT and Later

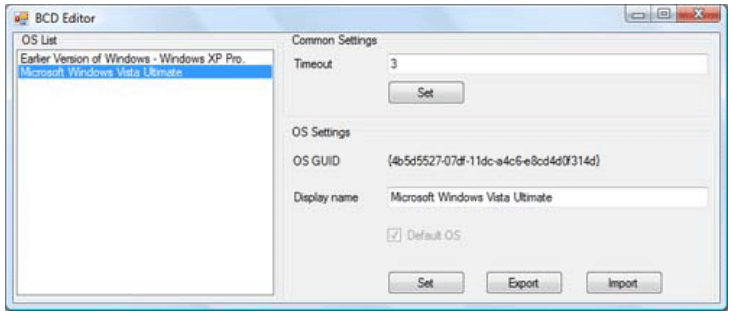
- All NTFS computers perform the following steps when the computer is turned on:
  - Power-on self test (POST)
  - Initial startup
  - Boot loader
  - Hardware detection and configuration
  - Kernel loading
  - User logon



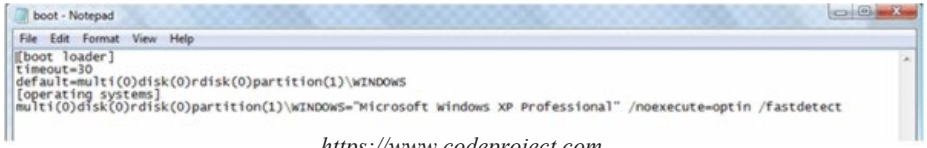
© Cengage Learning 2018

Startup in Windows NT and Later (Cont)

- Startup Files for Windows Vista:
  - The Ntldr (*NT Loader*) program in Windows XP used to load the OS has been replaced with these three boot utilities:
    - Bootmgr.exe
    - Winload.exe
    - Winresume.exe
  - Windows Vista includes the BCD editor for modifying boot options and updating the BCD registry file
  - The BCD store (*Namespace container for BCD Objects*) replaces the Windows XP boot.ini file
    - Old Windows uses *boot.ini* to configure boot sequences



https://www.codeproject.com



https://www.codeproject.com

© Cengage Learning 2018

Startup in Windows NT and Later (Cont)

- Startup Files for Windows XP:
  - NT Loader (NTLDR)
  - Boot.ini
  - Ntoskrnl.exe
  - Bootvid.dll
  - Hal.dll
  - BootSect.dos
  - NTDetect.com
  - NTBootdd.sys
  - Pagefile.sys

Table 5-8 Windows XP system files

Filename	Description
Ntoskrnl.exe	The XP executable and kernel
Ntkrnlpa.exe	The physical address support program for accessing more than 4 GB of physical RAM
Hal.dll	The Hardware Abstraction Layer (described earlier)
Win32k.sys	The kernel-mode portion of the Win32 subsystem
Ntdll.dll	System service dispatch stubs to executable functions and internal support functions
Kernel32.dll	Core Win32 subsystem DLL file
Advapi32.dll	Core Win32 subsystem DLL file
User32.dll	Core Win32 subsystem DLL file
Gdi32.dll	Core Win32 subsystem DLL file

© 2015 Cengage Learning®

Check out : Guide to Computer Forensics and Investigations-Course Technology 5th edition for details

© Cengage Learning 2018

Startup in Windows NT and Later (Cont)

Why need to know the start up process?

- Contamination Concerns with Windows XP
  - When you start a Windows XP NTFS workstation, several files are accessed immediately
    - The last access date and time stamp for the files change to the current date and time
    - Destroys any potential evidence that shows when a Windows XP workstation was last used

© Cengage Learning 2018

Understanding Virtual Machines

- **Virtual machine**
  - Allows you to create a representation of another computer on an existing physical computer
- A virtual machine is just a few files on your hard drive
  - Must allocate space to it
- A virtual machine recognizes components of the physical machine it's loaded on
  - Virtual OS is limited by the physical machine's OS as behavior of virtual OS could be affected by physical machine's OS (i.e *certain operations may be blocked*)

© Cengage Learning 2018

Understanding Virtual Machines (Cont)

- In digital forensics
  - Virtual machines make it possible to restore a suspect drive on your virtual machine
    - *Test can be performed* and run nonstandard software the suspect might have loaded
- From a network forensics standpoint, you need to be aware of some potential issues, such as:
  - A virtual machine used to attack another system or network.
  - *File slack, unallocated space, and so forth don't exist on a virtual machine, so many standard items don't work on virtual drives.*

© Cengage Learning 2018

Creating a Virtual Machine

- Popular applications for creating virtual machines
  - VMware Server, VMware Player and VMware Workstation, Oracle VM VirtualBox, Microsoft Virtual PC, and Hyper-V

- Using VirtualBox
  - An open-source program that can be downloaded at www.virtualbox.org/wiki/Downloads

https://lifehacker.com

© Cengage Learning 2018

Summary

- When booting 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

© Cengage Learning 2018

Summary (Cont)

- When files are deleted in a FAT file system, the Greek letter sigma (0xE5) 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, data streams can obscure information that might have evidentiary value

© Cengage Learning 2018

Summary (Cont)

- 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
- Virtual machines enable you to run other OSs from a Windows computer

© Cengage Learning 2018