Guide to Computer Forensics and Investigations Fourth Edition

Chapter 6

Working with Windows and DOS Systems

Objectives

- Explain the purpose and structure of file systems
- Describe Microsoft file structures
- Explain the structure of New Technology File System (NTFS) disks
- List some options for decrypting drives encrypted with whole disk encryption

Objectives (continued)

- Explain how the Windows Registry works
- Describe Microsoft startup tasks
- Describe MS-DOS 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
- A file system is usually directly related to an OS
- When you need to access a suspect's computer to acquire or inspect data
  - You should be familiar with the computer's platform

Understanding the Boot Sequence

- 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)
  - Contains programs that perform input and output at the hardware level

Understanding the Boot Sequence (continued)

- **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 (continued)

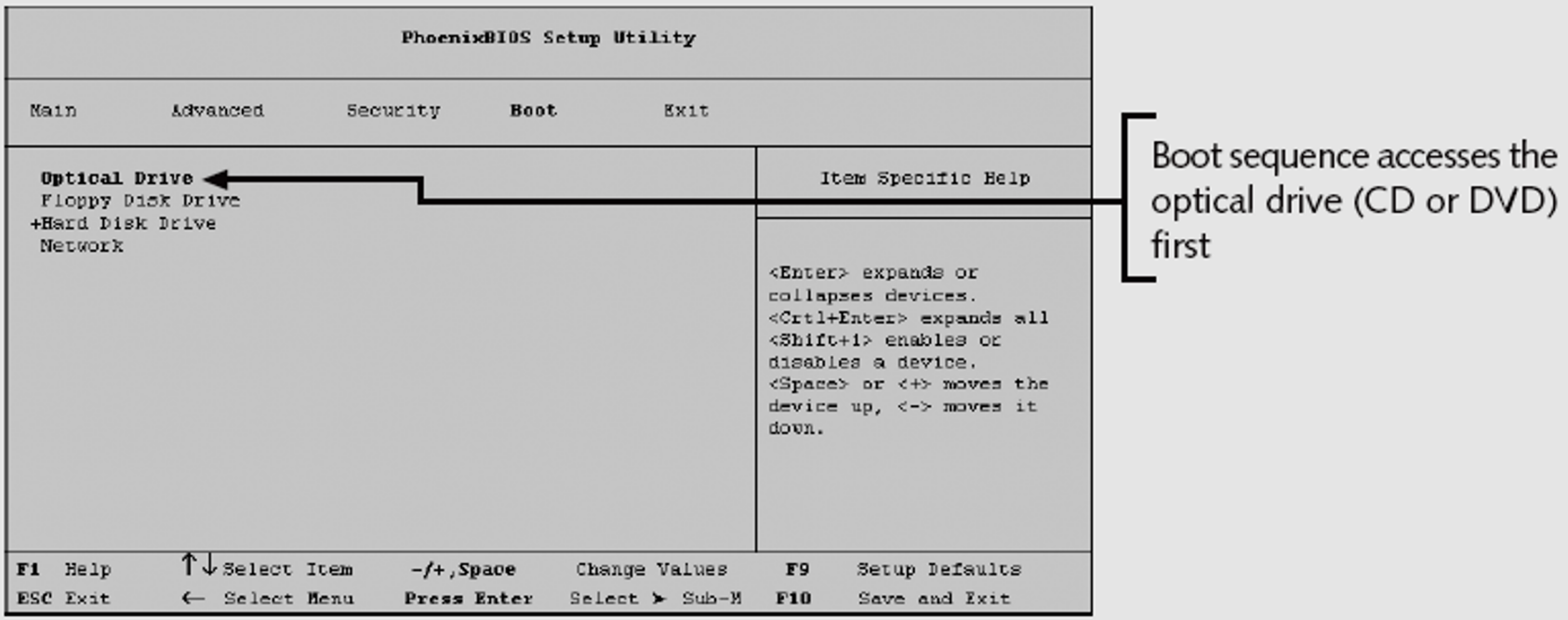


Figure 6-1 A typical CMOS setup screen

Understanding Disk Drives

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

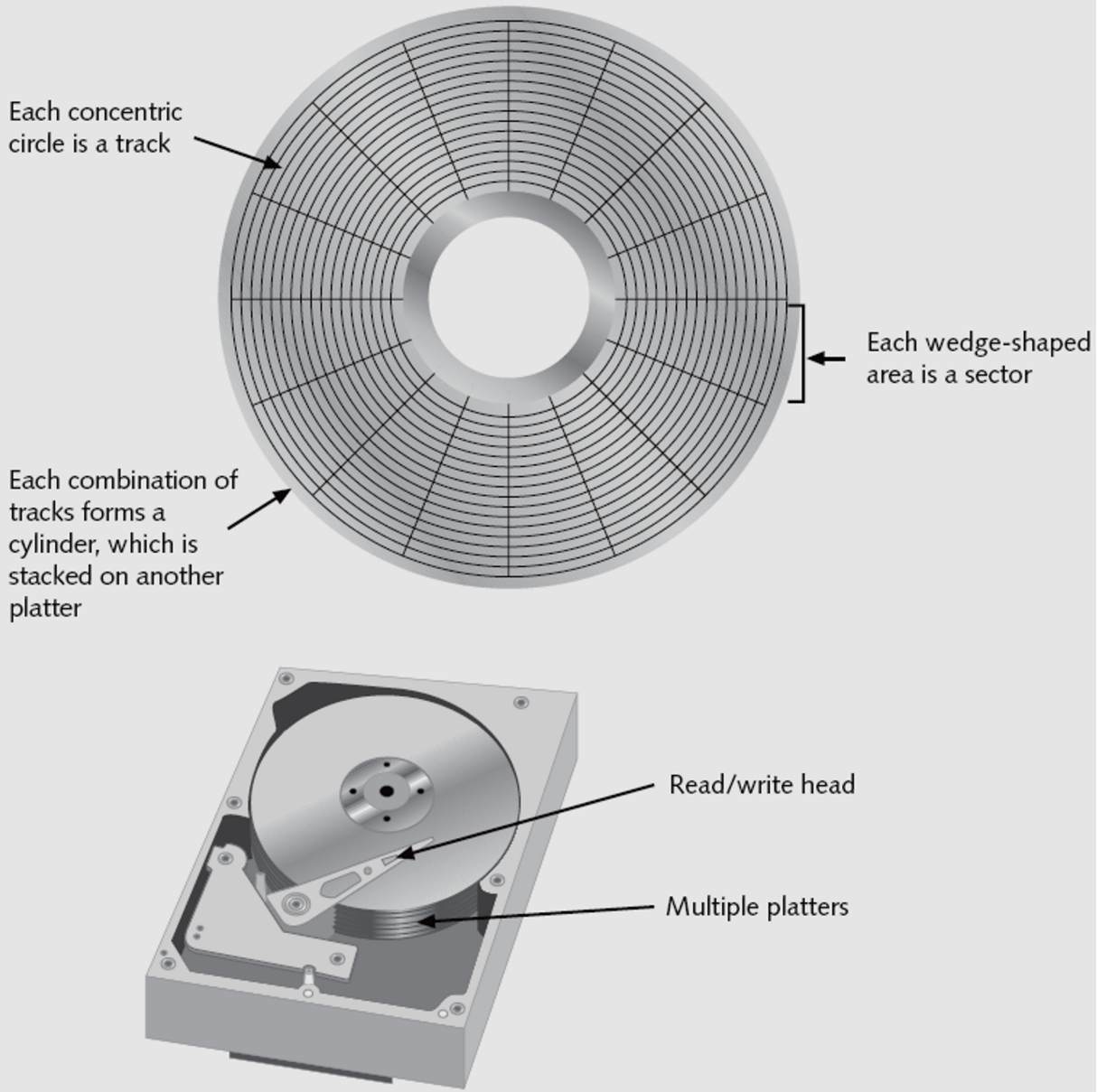


Figure 6-2 Components of the drive structure

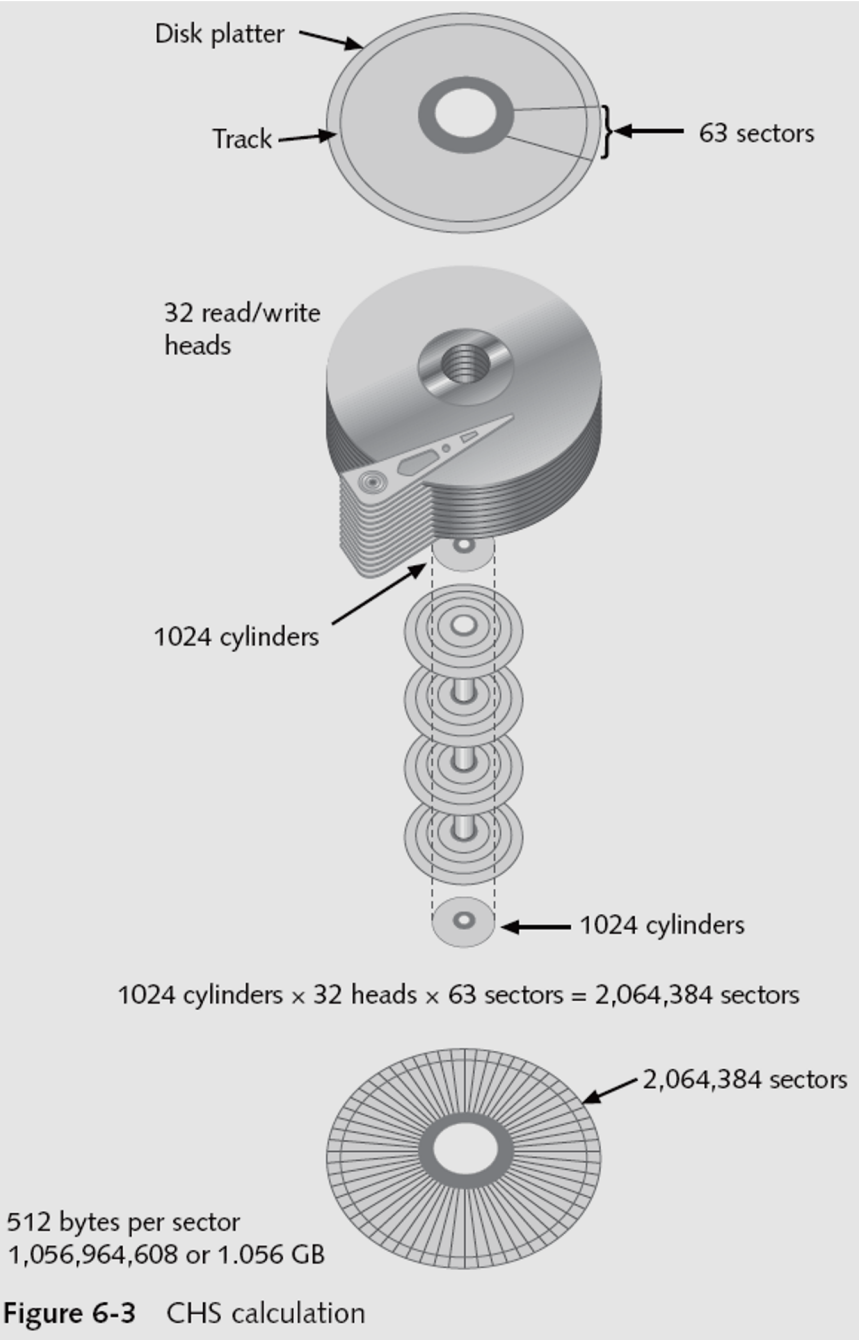


Figure 6-3 CHS calculation

Understanding Disk Drives (continued)

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

Exploring Microsoft File Structures

- In Microsoft file structures, sectors are grouped to form **clusters**
  - Storage allocation units of one or more sectors
- Clusters are typically 512, 1024, 2048, 4096, or more bytes each
- Combining sectors minimizes the overhead of writing or reading files to a disk

Exploring Microsoft File Structures (continued)

- Clusters are numbered sequentially starting at 2
  - 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

- A **partition** is a logical drive
- FAT16 does not recognize disks larger than 2 MB
  - Large disks have to be partitioned
- Hidden partitions or voids
  - Large unused gaps between partitions on a disk
- **Partition gap**
  - Unused space between partitions

Disk Partitions (continued)

- Disk editor utility can alter information in partition table
  - To hide a partition
- Can examine a partition's physical level with a disk editor:
  - Norton DiskEdit, WinHex, or Hex Workshop
- Analyze the key hexadecimal codes the OS uses to identify and maintain the file system

Table 6-1 Hexadecimal codes in the partition table

Hexadecimal code	File system
01	DOS 12-bit FAT
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
08	AIX bootable partition
09	AIX data partition
0B	DOS 32-bit FAT
0C	DOS 32-bit FAT for interrupt 13 support
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, 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

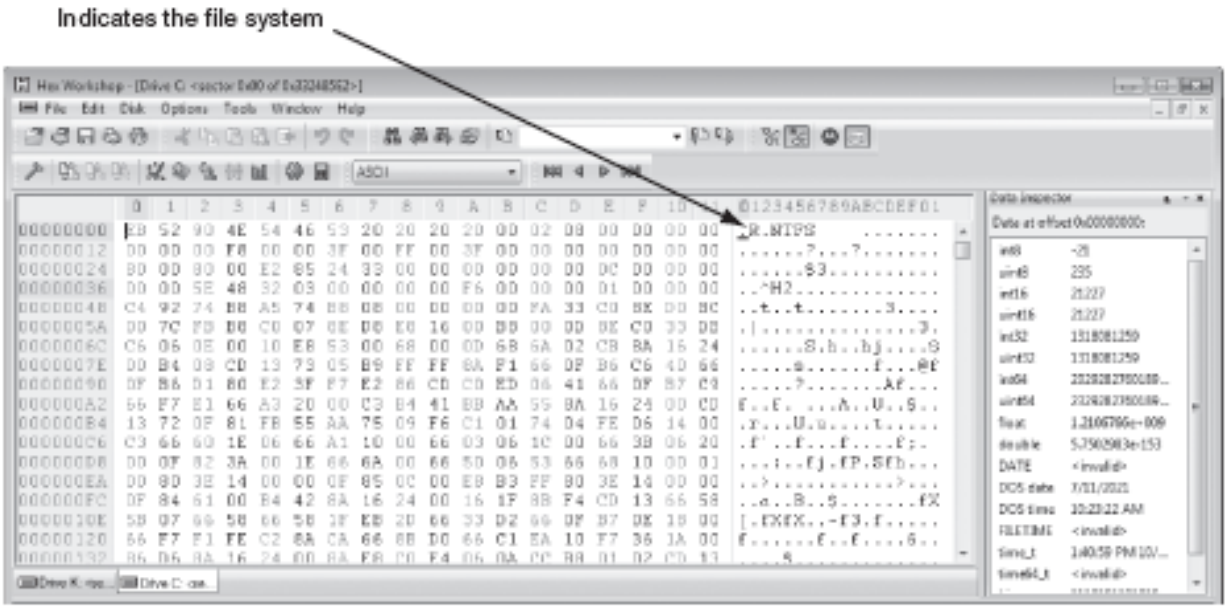


Figure 6-4 Hex Workshop identifying the file system

Disk Partitions (continued)

- Hex Workshop allows you to identify file headers
  - To identify file types with or without an extension

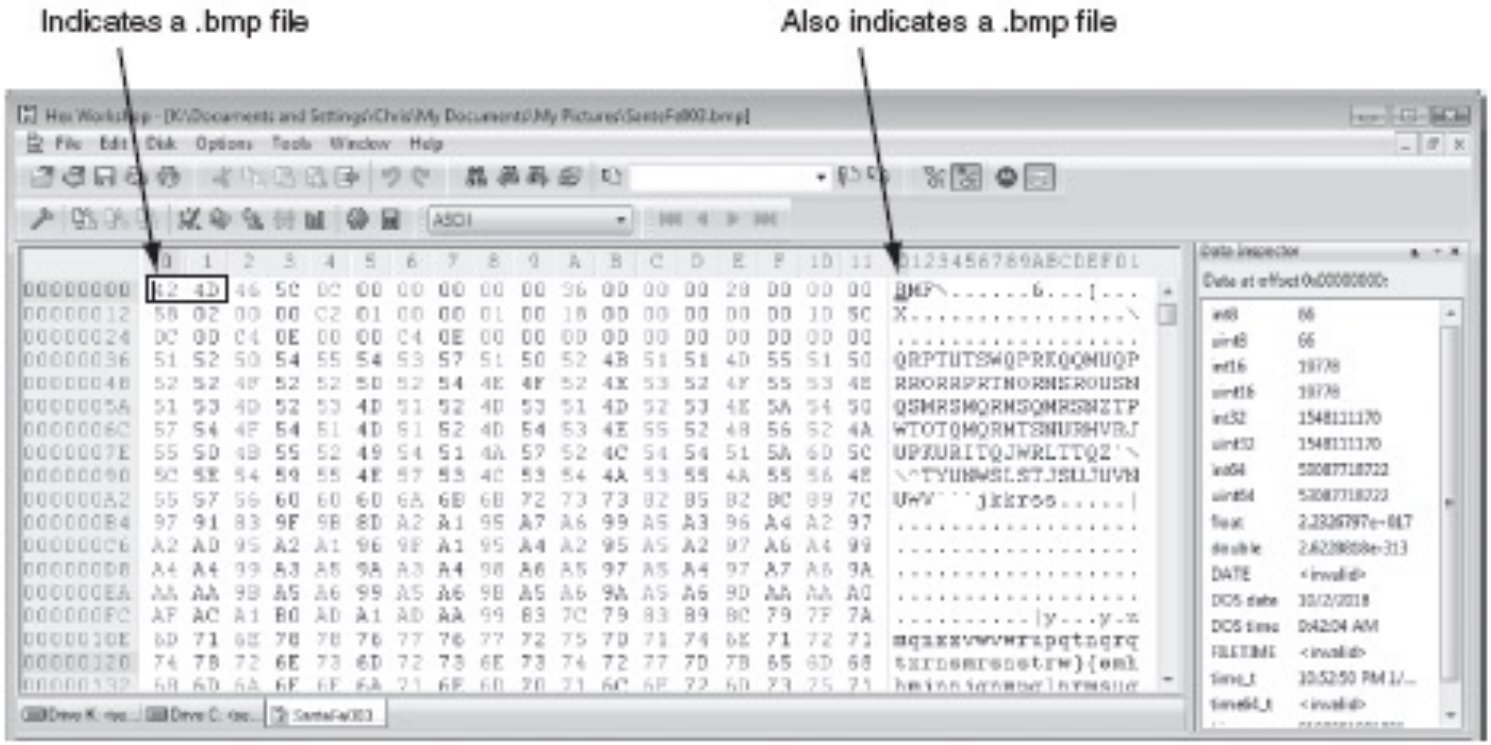


Figure 6-5 Hex Workshop indicating a .bmp file

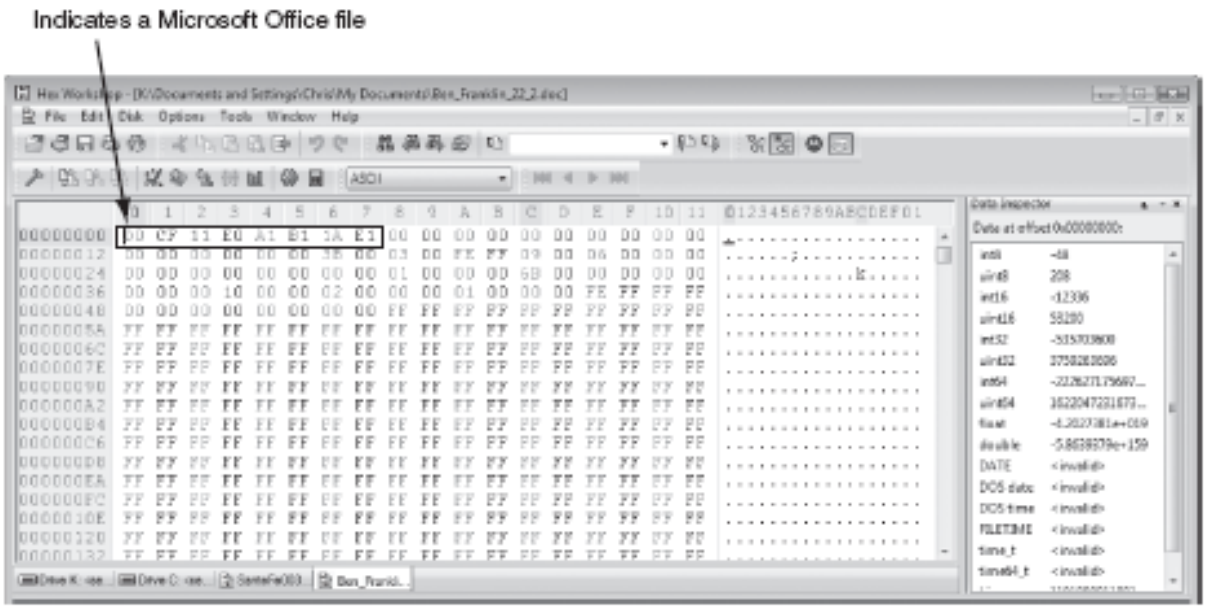


Figure 6-6 Hex Workshop indicating a Microsoft Office file

Master Boot Record

- On Windows and DOS computer systems
  - Boot disk contains a file called the **Master Boot Record (MBR)**
- 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 PartitionMagic's Boot Magic

Examining FAT Disks

- **File Allocation Table (FAT)**
  - File structure database that Microsoft originally designed for floppy disks
  - Used before Windows NT and 2000
- 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, and VFAT

Examining FAT Disks (continued)

- Cluster sizes vary according to the hard disk size and file system

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 (continued)

- 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 (continued)

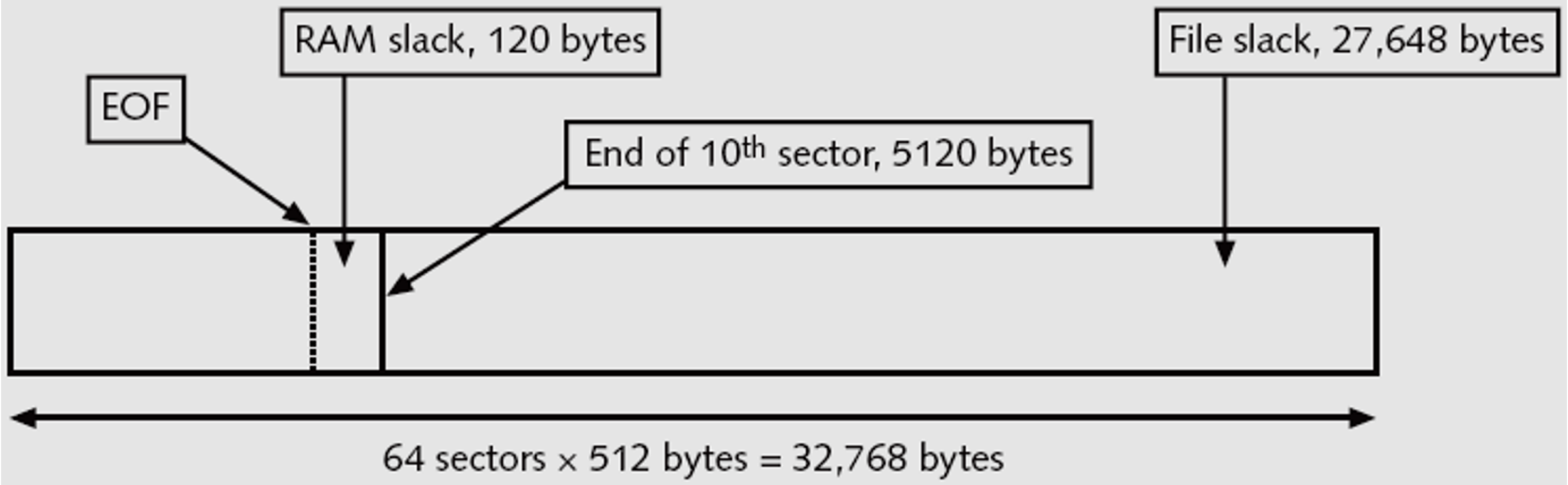


Figure 6-7 File slack space

Examining FAT Disks (continued)

- 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 (continued)

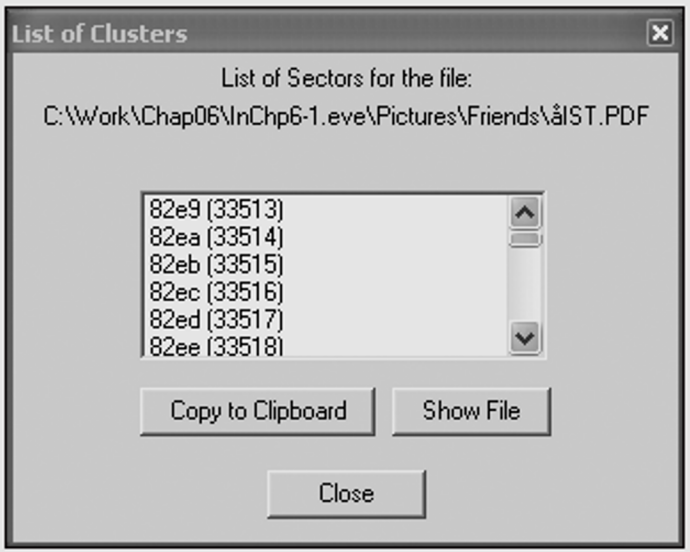


Figure 6-8 Chained sectors associated with clusters as a result of increasing file size

Examining FAT Disks (continued)

- 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
  - Primary file system for Windows Vista
- 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

Examining NTFS Disks (continued)

- 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 (continued)

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

Table 6-3 Cluster sizes in an NTFS disk (continued)

Drive size	Sectors per cluster	Cluster size
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 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 (continued)

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 (continued)

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

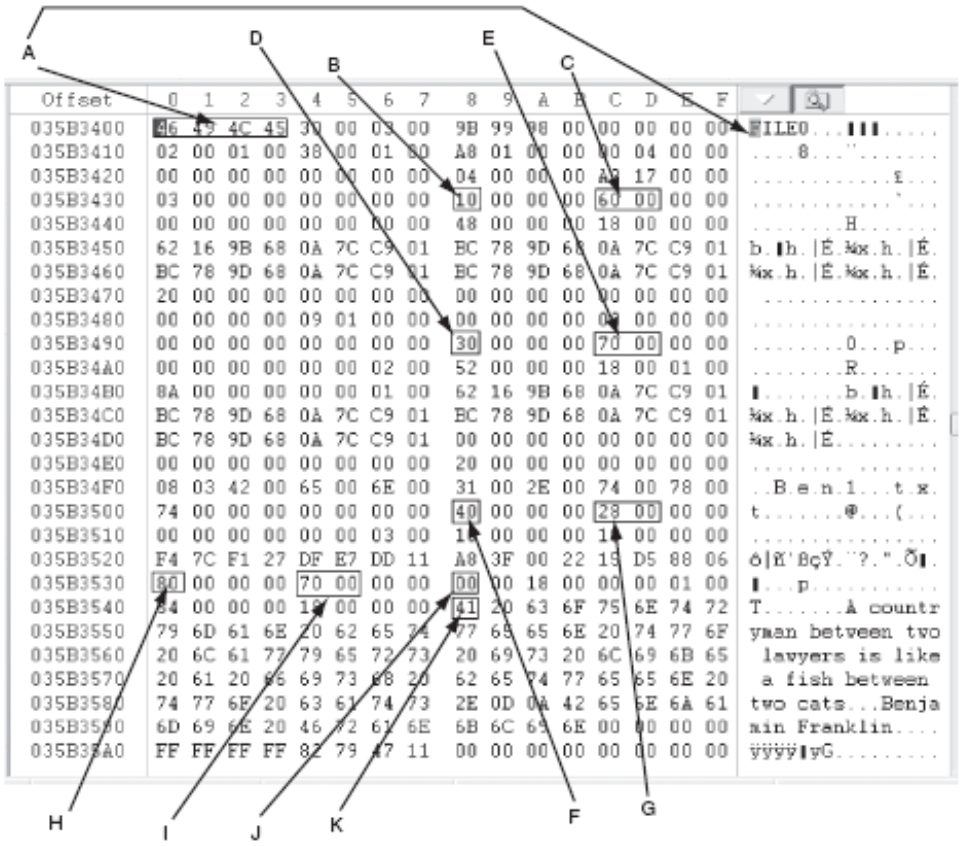
- In the NTFS MFT
  - All files and folders are stored in separate records of 1024 bytes each
- Each record contains file or folder information
  - This information is divided into record fields containing metadata
- A record field is referred to as an **attribute ID**
- File or folder information is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

MFT and File Attributes (continued)

- 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

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

Figure 6-9 Resident file in an MFT record

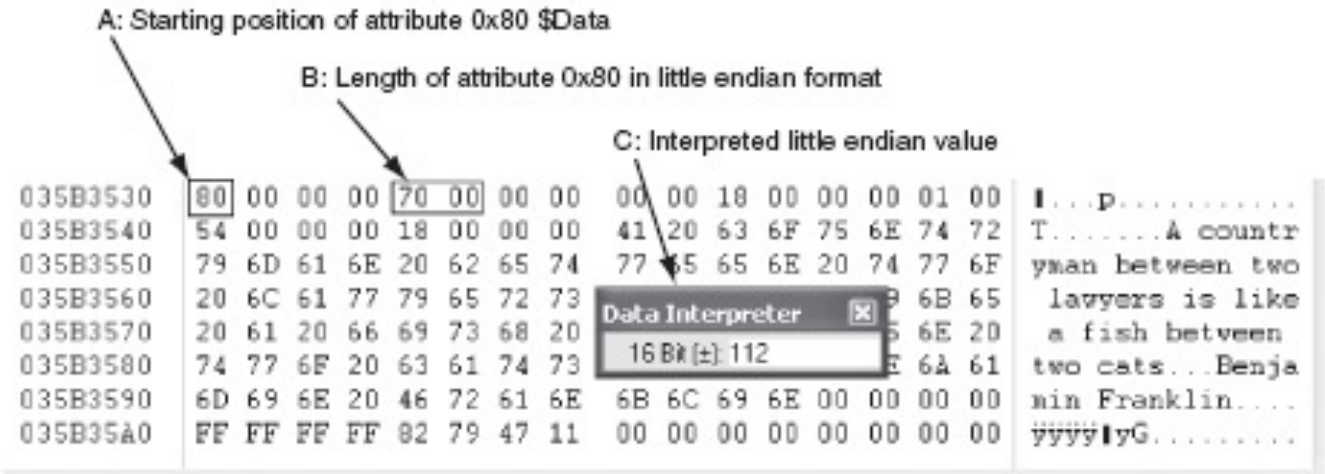


Figure 6-10 File data for a resident file

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
035B3C00	46	49	4C	45	30	00	03	00	D3	BD	98	00	00	00	00	00
035B3C10	02	00	01	00	38	00	01	00	80	01	00	00	00	04	00	00
035B3C20	00	00	00	00	00	00	00	00	05	00	00	00	A5	17	00	00
035B3C30	03	00	00	00	00	00	00	00	10	00	00	00	60	00	00	00
035B3C40	00	00	00	00	00	00	00	00	48	00	00	00	18	00	00	00
035B3C50	10	C0	13	88	0B	7C	C9	01	6A	22	16	88	0B	7C	C9	01
035B3C60	6A	22	16	88	0B	7C	C9	01	6A	22	16	88	0B	7C	C9	01
035B3C70	20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
035B3C80	00	00	00	00	09	01	00	00	00	00	00	00	00	00	00	00
035B3C90	00	00	00	00	00	00	00	00	30	00	00	00	70	00	00	00
035B3CA0	00	00	00	00	00	00	02	00	52	00	00	00	18	00	01	00
035B3CB0	8A	00	00	00	00	00	01	00	10	C0	13	88	0B	7C	C9	01
035B3CC0	6A	22	16	88	0B	7C	C9	01	6A	22	16	88	0B	7C	C9	01
035B3CD0	6A	22	16	88	0B	7C	C9	01	00	00	00	00	00	00	00	00
035B3CE0	00	00	00	00	00	00	00	00	20	00	00	00	00	00	00	00
035B3CF0	08	03	42	00	65	00	6E	00	32	00	2E	00	72	00	74	00
035B3D00	66	00	00	00	00	00	00	00	40	00	00	00	28	00	00	00
035B3D10	00	00	00	00	00	00	04	00	10	00	00	00	18	00	00	00
035B3D20	F7	7C	F1	27	DF	E7	DD	11	A8	3F	00	22	15	D5	88	06
035B3D30	80	00	00	00	48	00	00	00	01	00	00	00	00	00	03	00
035B3D40	00	00	00	00	00	00	00	00	02	00	00	00	00	00	00	00
035B3D50	40	00	00	00	00	00	00	00	00	06	00	00	00	00	00	00
035B3D60	78	05	00	00	00	00	00	00	78	05	00	00	00	00	00	00
035B3D70	31	03	15	55	01	00	01	00	FF	FF	FF	FF	82	79	47	11

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 (continued)

- 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**
  - Ways data can be appended to existing files
  - Can obscure valuable evidentiary data, intentionally or by coincidence
- In NTFS, a 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 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 Vista Business Edition or higher, XP Professional, or 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

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
  - Copy
  - Efsrecvr (used to decrypt EFS files)

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 in the same way FAT does

Understanding Whole Disk Encryption

- 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

Understanding Whole Disk Encryption (continued)

- 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
  - A **Trusted Platform Module (TPM)** microchip to generate encryption keys and authenticate logins

Understanding Whole Disk Encryption (continued)

- 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

Examining Microsoft BitLocker

- Available only with Vista Enterprise and Ultimate editions
- Hardware and software requirements
  - A computer capable of running Windows Vista
  - 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:
  - PGP Whole Disk Encryption
  - Voltage SecureDisk
  - Utimaco SafeGuard Easy
  - Jetico BestCrypt Volume Encryption
  - SoftWinter Sentry 2020 for Windows XP
- Some available open-source encryption tools:
  - TrueCrypt
  - CrossCrypt
  - FreeOTFE

Understanding the Windows Registry

- **Registry**
  - A database that stores hardware and software configuration information, network connections, user preferences, and setup information
- For investigative purposes, the Registry can contain valuable evidence
- To view the Registry, you can use:
  - Regedit (Registry Editor) program for Windows 9x systems
  - Regedt32 for Windows 2000 and XP

Exploring the Organization of the Windows Registry

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

Exploring the Organization of the Windows Registry (continued)

Table 6-6 Registry file locations and purposes

Filename and location	Purpose of file
Windows 9x/Me	
Windows\System.dat	User-protected storage area; contains installed program settings, usernames and passwords associated with installed programs, and system settings
Windows\User.dat Windows\profile\user-account	Contains the most recently used (MRU) files list and desktop configuration settings; every user account created on the system has its own user data file
Windows NT, 2000, XP, and Vista	
Documents and Settings\user-account\Ntuser.dat	User-protected storage area; contains the MRU files list and desktop configuration settings
Winnt\system32\config\Default	Contains the computer's system settings
Winnt\system32\config\SAM	Contains user account management and security settings
Winnt\system32\config\Security	Contains the computer's security settings
Winnt\system32\config\Software	Contains installed programs settings and associated usernames and passwords
Winnt\system32\config\System	Contains additional computer system settings

Exploring the Organization of the Windows Registry (continued)

Table 6-7 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

Examining the Windows Registry

- Use ProDiscover Basic to extract System.dat and User.dat from an image file

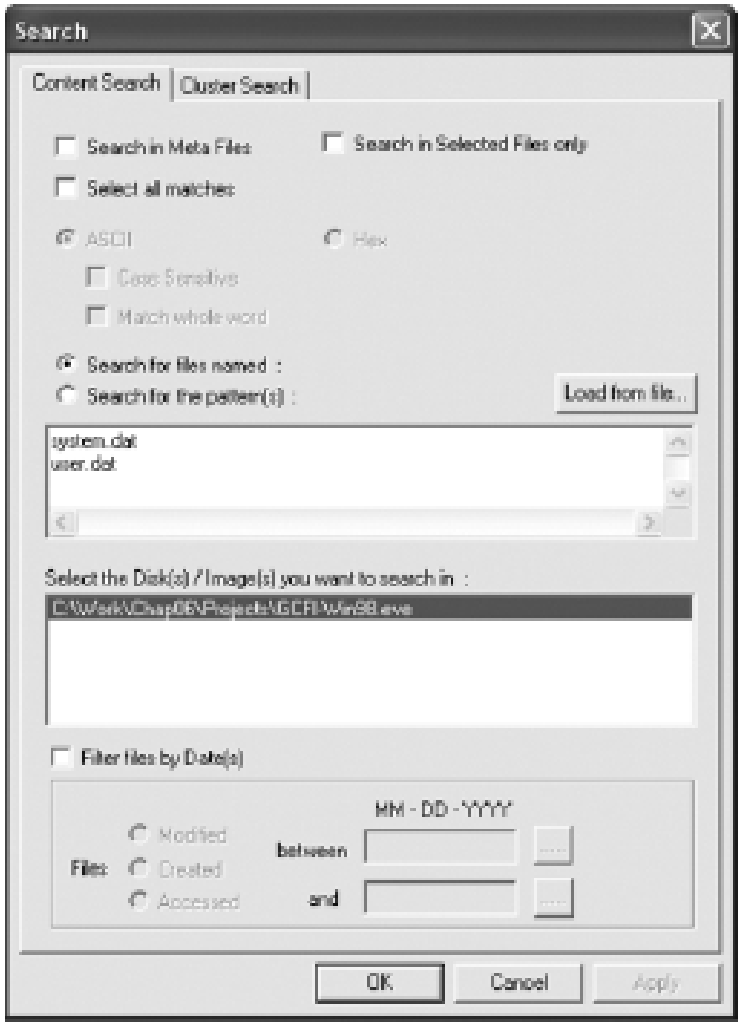


Figure 6-26 Searching for Registry files

Examining the Windows Registry (continued)

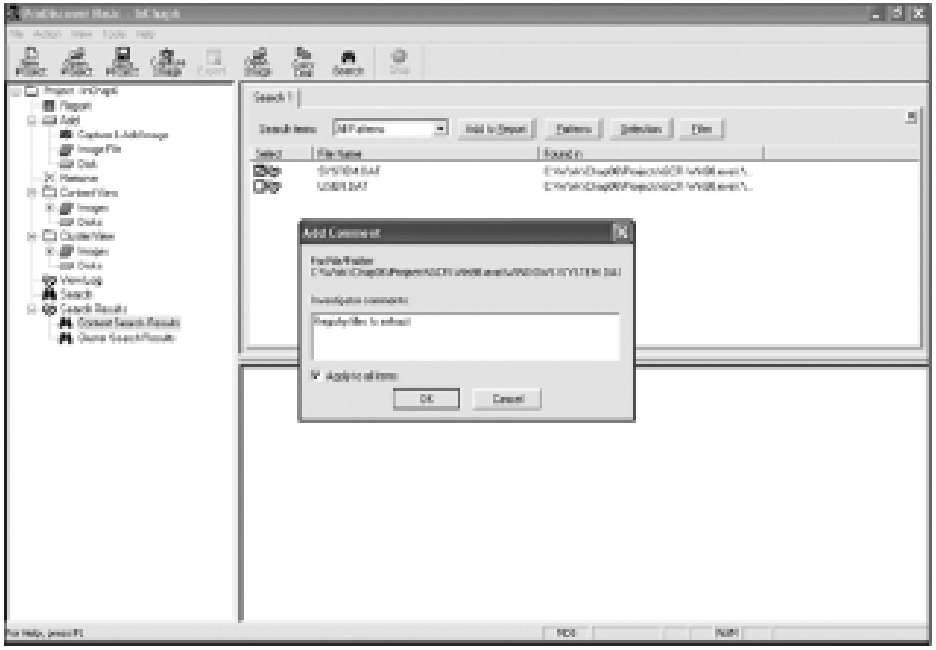


Figure 6-27 Selecting files in the search results

Examining the Windows Registry (continued)

- Use AccessData Registry Viewer to see what information you can find in these files

Examining the Windows Registry (continued)

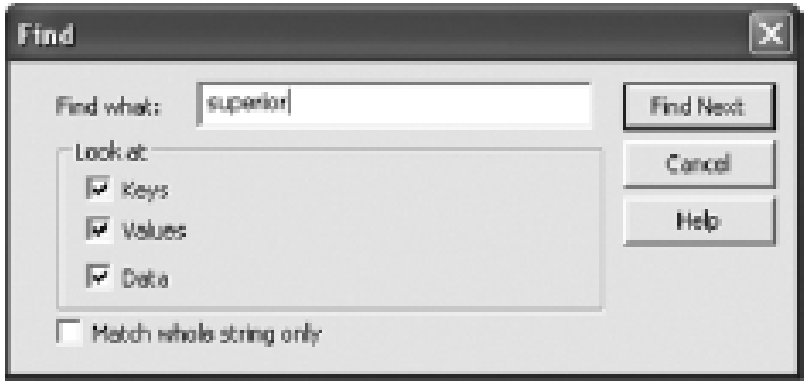


Figure 6-28 Entering a search term in Registry Viewer

Examining the Windows Registry (continued)

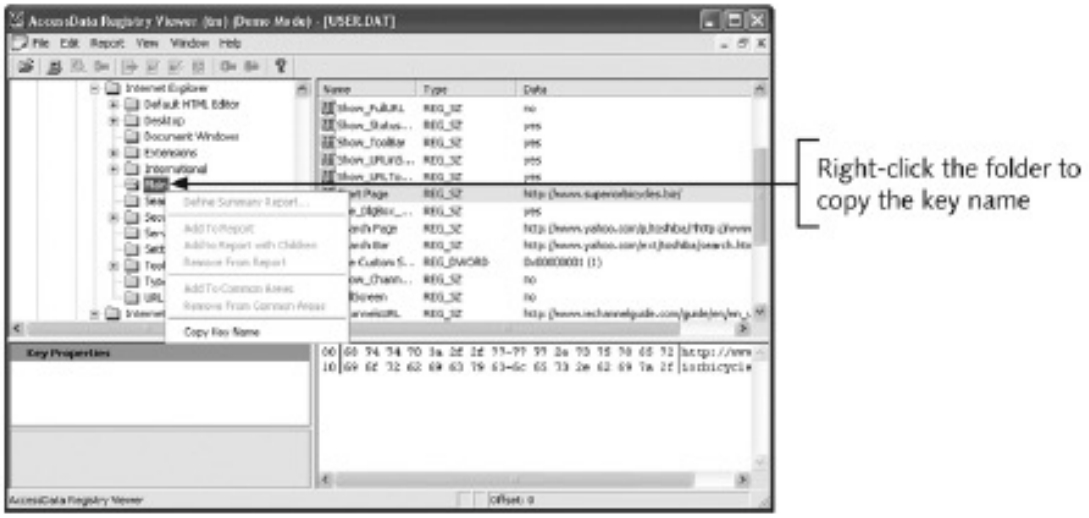


Figure 6-29 Copying a key name in Registry Viewer

Examining the Windows Registry (continued)

Figure 6-30 The search results showing paths for keys of interest

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

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

Startup in Windows NT and Later (continued)

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

Startup in Windows NT and Later (continued)

- Windows XP System Files

Table 6-8 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 Win32subsystem
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

Startup in Windows NT and Later (continued)

- 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

Startup in Windows 9x/Me

- System files in Windows 9x/Me containing valuable information can be altered easily during startup
- Windows 9x and Windows Me have similar boot processes
  - With Windows Me you can't boot to a true MS-DOS mode
- Windows 9x OSs have two modes:
  - **DOS protected-mode interface (DPMI)**
  - **Protected-mode GUI**

Startup in Windows 9x/Me (continued)

- The system files used by Windows 9x have their origin in MS-DOS 6.22
  - **Io.sys** communicates between a computer's BIOS, the hardware, and the OS kernel
    - If F8 is pressed during startup, Io.sys loads the Windows Startup menu
  - **Msdos.sys** is a hidden text file containing startup options for Windows 9x
  - **Command.com** provides a command prompt when booting to MS-DOS mode (DPMI)

Understanding MS-DOS Startup Tasks

- Two files are used to configure MS-DOS at startup:
  - **Config.sys**
    - A text file containing commands that typically run only at system startup to enhance the computer's DOS configuration
  - **Autoexec.bat**
    - A batch file containing customized settings for MS-DOS that runs automatically
- Io.sys is the first file loaded after the ROM bootstrap loader finds the disk drive

Understanding MS-DOS Startup Tasks (continued)

- Msdos.sys is the second program to load into RAM immediately after Io.sys
  - It looks for the Config.sys file to configure device drivers and other settings
- Msdos.sys then loads Command.com
- As the loading of Command.com nears completion, Msdos.sys looks for and loads Autoexec.bat

Other Disk Operating Systems

- Control Program for Microprocessors (CP/M)
  - First nonspecific microcomputer OS
  - Created by Digital Research in 1970
  - 8-inch floppy drives; no support for hard drives
- Digital Research Disk Operating System (DR-DOS)
  - Developed in 1988 to compete with MS-DOS
  - Used FAT12 and FAT16 and had a richer command environment

Other Disk Operating Systems (continued)

- Personal Computer Disk Operating System (PC-DOS)
  - Created by Microsoft under contract for IBM
  - PC-DOS works much like MS-DOS

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

Figure 6-32 A virtual machine running on the host computer's desktop

Understanding Virtual Machines (continued)

- In computer forensics
  - Virtual machines make it possible to restore a suspect drive on your virtual machine
    - 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

Creating a Virtual Machine

- Two popular applications for creating virtual machines
  - VMware and Microsoft Virtual PC
- Using Virtual PC
  - You must download and install Virtual PC first

Creating a Virtual Machine (continued)

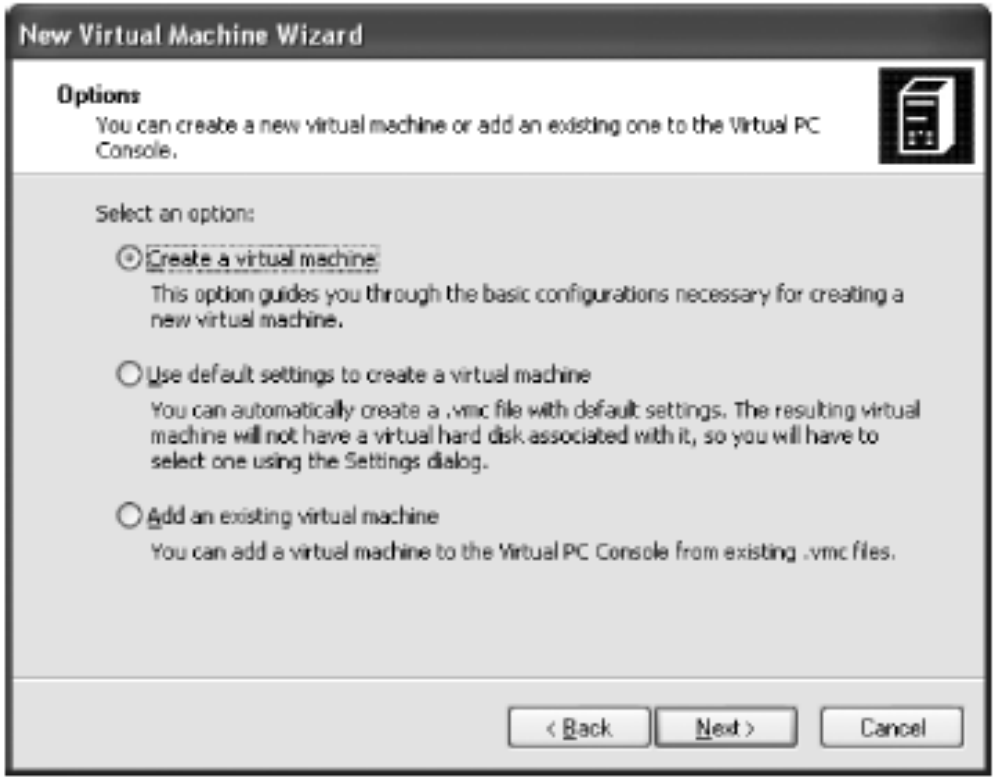


Figure 6-33 Creating a new virtual machine

Creating a Virtual Machine (continued)

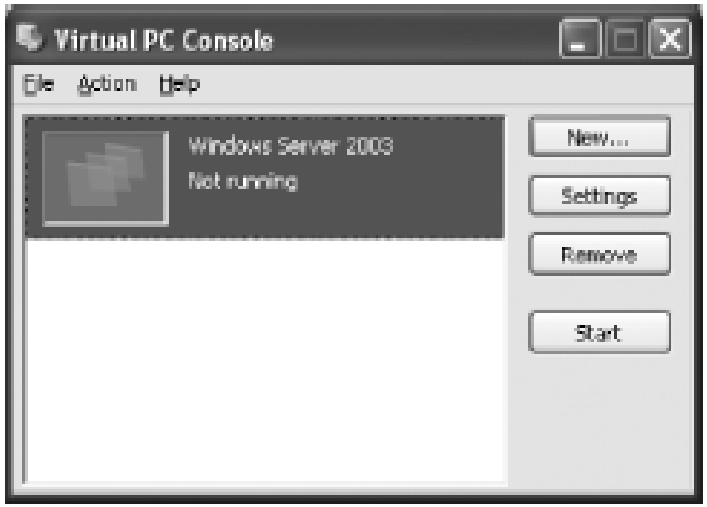


Figure 6-34 The Virtual PC Console with a virtual machine available

Creating a Virtual Machine (continued)

- You need an ISO image of an OS
  - Because no OSs are provided with Virtual PC
- Virtual PC creates two files for each virtual machine:
  - A .vhd file, which is the actual virtual hard disk
  - A .vmc file, which keeps track of configurations you make to that disk
- See what type of physical machine your virtual machine thinks it's running
  - Open the Virtual PC Console, and click Settings

Creating a Virtual Machine (continued)

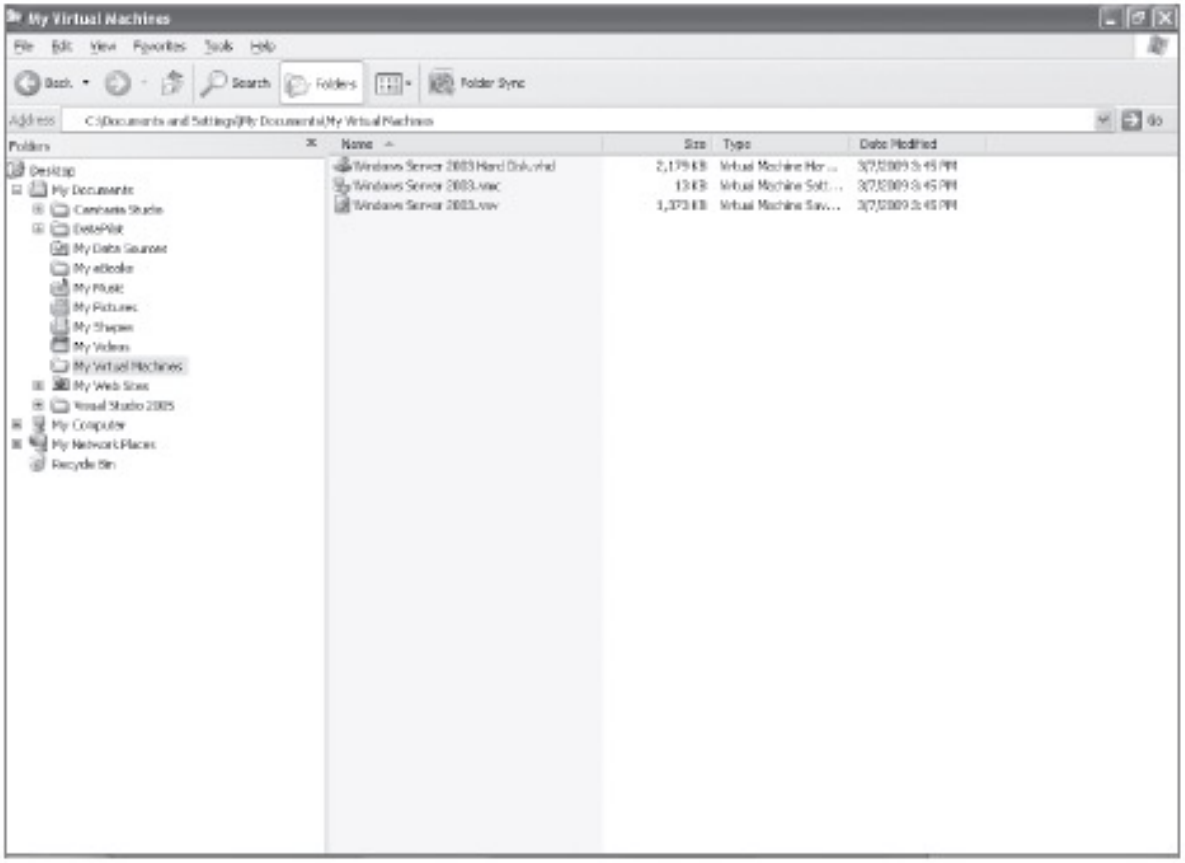


Figure 6-35 Virtual machine configuration files

Creating a Virtual Machine (continued)

Figure 6-36 Properties of a virtual machine

Summary

- When booting a suspect's computer, using boot media, such as forensic boot floppies or CDs, 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 (continued)

- When files are deleted in a FAT file system, the Greek letter sigma (0x05) is inserted in the first character of the filename in the directory
- New Technology File System (NTFS) is more versatile because it uses the Master File Table (MFT) to track file information
- In NTFS, data streams can obscure information that might have evidentiary value

Summary (continued)

- Maintain a library of older operating systems and applications
- 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