

Module 2

Understanding File Systems

- To investigate digital evidence effectively, you must understand how the most commonly used OSs work and how they store files.
- A file system gives an OS a road map to data on a disk.
- The 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 related to your investigation, you should be familiar with both the computer's OS and file system so that you can access and modify system settings when necessary.

Understanding the Boot Sequence

- To ensure that you don't contaminate or alter data on a suspect's system, you must know how to access and modify Complementary Metal Oxide Semiconductor (CMOS), BIOS, Extensible Firmware Interface (EFI), and Unified Extensible Firmware Interface (UEFI) settings.

- A computer stores system configuration and date and time information in the CMOS when power to the system is off.
- The system BIOS or EFI contains programs that perform input and output at the hardware level.
- BIOS is designed for x86 computers and typically used on disk drives with Master Boot Records (MBRs).
- EFI is designed for x64 computers and uses GUID Partition Table (GPT)– formatted disks

- When a subject's computer starts, you must make sure it boots to a forensically configured CD, DVD, or USB drive because booting to the hard disk overwrites and changes evidentiary data.
- To do this, you access the CMOS setup by monitoring the computer during the bootstrap process to identify the correct key or keys to use.

- The bootstrap process , which is contained in ROM, tells the computer how to proceed.
- As the computer starts, the screen usually displays the key or keys, such as the Delete key, you press to open the CMOS setup screen.

- The key you press to access CMOS depends on the computer's BIOS. Many BIOS manufacturers use the Delete key to access CMOS; other manufacturers use Ctrl+Alt+Insert, Ctrl+A, Ctrl+S, or Ctrl+F1, F2, or F10.
- A safe method for verifying the BIOS is removing all hard drives from the computer, which enables you to start the computer to verify its BIOS date and time without accessing the disk drive.

Main

Advanced

Power

Boot

Tools

Exit

Main Settings

System Time [11:17:19]
System Date [Wed 04/05/2018]

► SATA3G_1 : [WDC WD10EZEX-22MFC]
► SATA3G_2 : [HFS250G32TND-N1A2A]
► SATA3G_3 : [Not Detected]
► SATA3G_4 : [Not Detected]
► SATA Configuration

► System Information

Use [ENTER], [TAB]
or [SHIFT-TAB] to
select a field.

Use [+] or [-] to
configure system Time.

→ Select Screen
↓ Select Item
+ Change Field
Tab Select Field
F1 General Help
F10 Save and Exit
ESC Exit

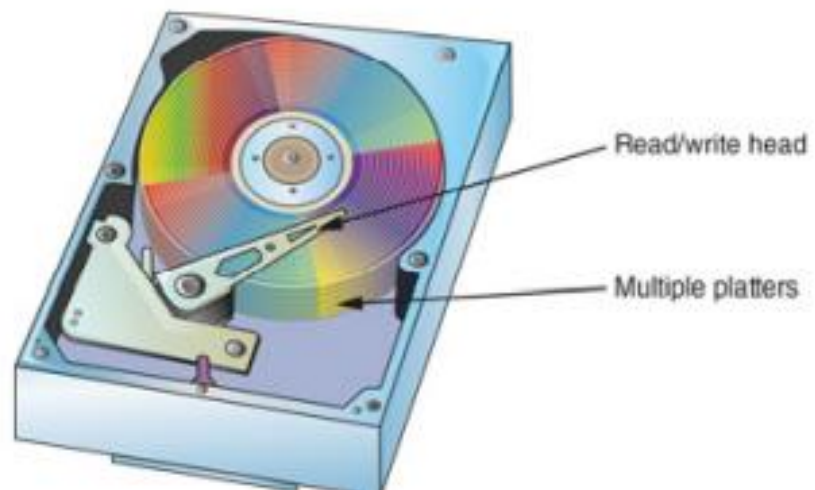
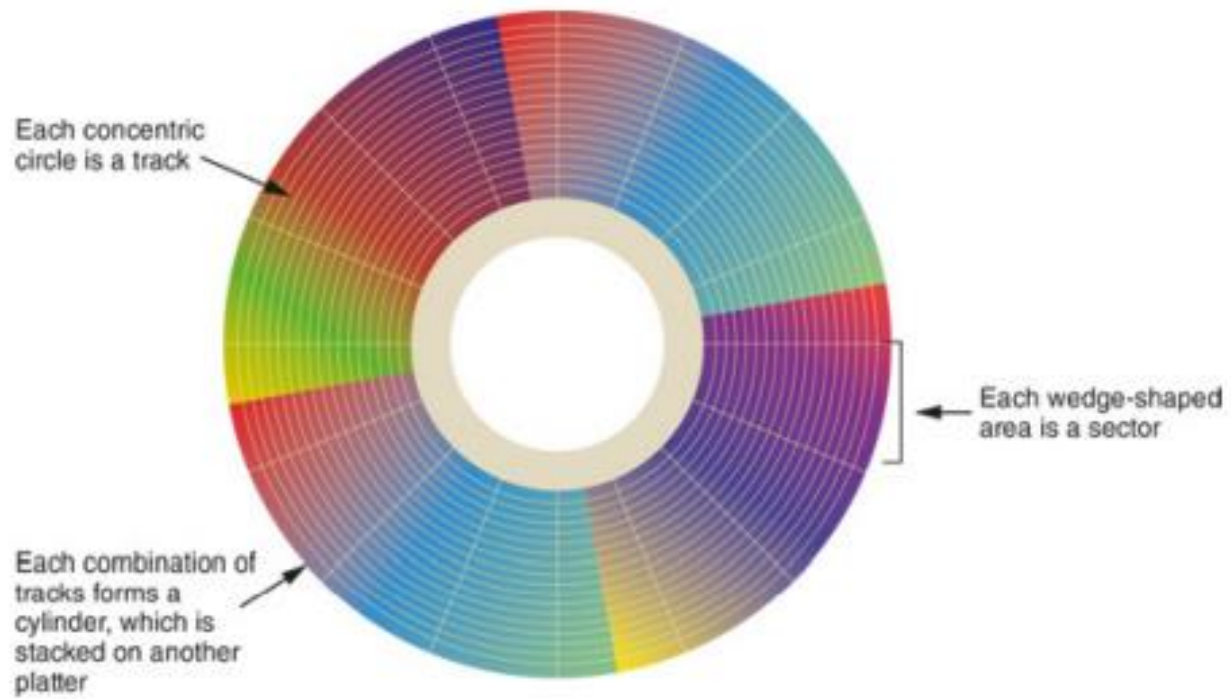
- Figure shows a typical CMOS setup screen, where you check a computer's boot sequence.
- If necessary, you can change the boot sequence so that the OS accesses the CD/DVD drive, for example, before any other boot device.
- Each BIOS vendor's screen is different, but you can refer to the vendor's documentation or Web site for instructions on changing the boot sequence.

Understanding Disk Drives

- You should be familiar with disk drives and how data is organized on a disk so that you can find data effectively.
- Disk drives are made up of one or more platters coated with magnetic material, and data is stored on platters in a particular way.

The disk drive components:

- Geometry— Geometry refers to a disk's logical structure of platters, tracks, and sectors.
- Head—The head is the device that reads and writes data to a drive. There are two heads per platter that read and write the top and bottom sides.
- Tracks— Tracks are concentric circles on a disk platter where data is located.
- Cylinders—A cylinder is a column of tracks on two or more disk platters. Typically, each platter has two surfaces: top and bottom.
- Sectors—A sector is a section on a track, usually made up of 512 bytes



- The manufacturer engineers a disk to have a certain number of sectors per track, and a typical disk drive stores 512 bytes per sector.
- Other disk properties, such as zone bit recording (ZBR) , track density , areal density , and head and cylinder skew , are handled at the drive's hardware or firmware level.
- ZBR is how most manufacturers deal with a platter's inner tracks having a smaller circumference (and, therefore, less space to store data) than its outer tracks.

- Track density is the space between each track.
- As with old vinyl records, the smaller the space between each track, the more tracks you can place on the platter.
- Areal density is the number of bits in one square inch of a disk platter. This number includes the unused space between tracks.
- Head and cylinder skew are used to improve disk performance.

Solid-State Storage Devices

- Flash memory storage devices used in USB drives, laptops, tablets, and cell phones can be a challenge for digital forensics examiners because if deleted data isn't recovered immediately, it might be lost forever.

- When data is deleted on a hard drive, only the references to it are removed, which leaves the original data in unallocated disk space. With forensics recovery tools, recovering data from magnetic media is fairly easy; you just copy the unallocated space.

- In USB drives and other solid state drive systems ,memory cells shift data at the physical level to other cells that have had fewer reads and writes continuously.
- The purpose of shifting (or rotating) data from one memory cell to another is to make sure all memory cells on the flash drive wear evenly.
- Memory cells are designed to perform only 10,000 to 100,000 reads/writes, depending on the manufacturer's design.
- When they reach their defined limits, they can no longer retain data. When you attempt to connect to the device, you get an access failure message. This process is controlled on the flash device's firmware

- In addition, when data is rotated to another memory cell, the old memory cell addresses are listed in a firmware file called a “garbage collector.”
- At some point, the flash drive’s firmware erases data in unallocated cells by overwriting the value of 1 in all cells listed in the garbage collector file.

- 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.
- You can test this feature with a USB drive easily by copying data to it, deleting it, and then making a forensic acquisition with any acquisition tool, such as OSForensics or X-Ways Forensics, immediately after the data is deleted.

- The first acquisition produces recoverable artifacts. If you let the USB drive sit and write no additional data to it, wear-leveling automatically overwrites the unallocated space.
- All solid-state drives(SSD) have an internal power source for memory cells (both allocated and unallocated) so that they can preserve data. I

- If you make another acquisition of the USB drive a day or more later, it reveals that the previously recoverable deleted data no longer exists.
- For mobile device forensics, this feature is extremely important, especially if a suspect deleted relevant messages, for example, just before the device was seized and taken into evidence

Exploring Microsoft File Structures

- Because most PCs use Microsoft software products, you should understand Microsoft file systems so that you know how Windows and DOS computers store files.
- Clusters
- File Allocation Table (FAT)
- NT File System (NTFS).
- The method an OS uses to store files determines where data can be hidden.
- When you examine a computer for forensic evidence, you need to explore these hiding places to determine whether they contain files or parts of files that might be evidence of a crime or policy violation.

clusters

- In Microsoft file structures, sectors are grouped to form clusters , which are 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.
- The OS groups one or more sectors into a cluster.
- The number of sectors in a cluster varies according to the disk size. For example, a double-sided floppy disk has one sector per cluster; a hard disk has four or more sectors per cluster.

File Allocation Table (FAT)

- It is used by the operating system (OS) to manage files on hard drives and other computer systems.
- It is often also found on in flash memory, digital cameras and portable devices.
- It is used to store file information and extend the life of a hard drive.

- Most hard drives require a process known as seeking; this is the actual physical searching and positioning of the read/write head of the drive.
- The FAT file system was designed to reduce the amount of seeking and thus minimize the wear and tear on the hard disc.

A FAT file system has four different sections, each as a structure in the FAT partition. The four sections are:

- Boot Sector:
 - This is also known as the reserved sector
 - It is located on the first part of the disc.
 - It contains:
 - the OS's necessary boot loader code to start a PC system
 - the partition table known as the master boot record (MRB) that describes how the drive is organized
 - BIOS parameter block (BPB) which describes the physical outline of the data storage volume.

- FAT Region:
 - This region generally encompasses two copies of the File Allocation Table which is for redundancy checking and specifies how the clusters are assigned.
- Data Region:
 - This is where the directory data and existing files are stored. It uses up the majority of the partition.

- Root Directory Region:

- This region is a directory table that contains the information about the directories and files.
- It is used with FAT16 and FAT12 but not with other FAT file systems.
- It has a fixed maximum size that is configured when created.

- Clusters are numbered sequentially, starting at 0 in NTFS and 2 in FAT. The first sector of all disks contains a system area, the boot record, and a file structure database. The OS assigns these cluster numbers, referred to as logical addresses . They point to relative cluster positions; for example, cluster address 100 is 98 clusters from cluster address 2.

- Sector numbers, however, are referred to as physical addresses because they reside at the hardware or firmware level and go from address 0 (the first sector on the disk) to the last sector on the disk. Clusters and their addresses are specific to a logical disk drive, which is a disk partition.

New Technology File System (NTFS)

- The New Technology File System (NTFS) is the standard file structure for the Windows NT operating system. It is used for retrieving and storing files on the hard disk.
- The NTFS introduced a number of enhancements, including innovative data structures that increased performance, improved metadata, and added expansions like security access control (ACL), reliability, disk space utilization, and file system journaling.

- The NTFS replaced the Windows 95 file allocation table (FAT), which were used in MS-DOS and earlier operating system versions.
- NTFS is also used with Windows 2000, Windows XP, and Windows Server 2003.

Disk Partitions

- Many hard disks are partitioned, or divided, into two or more sections.
- 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.

Disk Partitions

In order to use a hard drive, or a portion of a hard drive, in Windows you need to **first partition it** and **then format it**

Some hard drives can be divided into 1 or more partitions (called volumes)

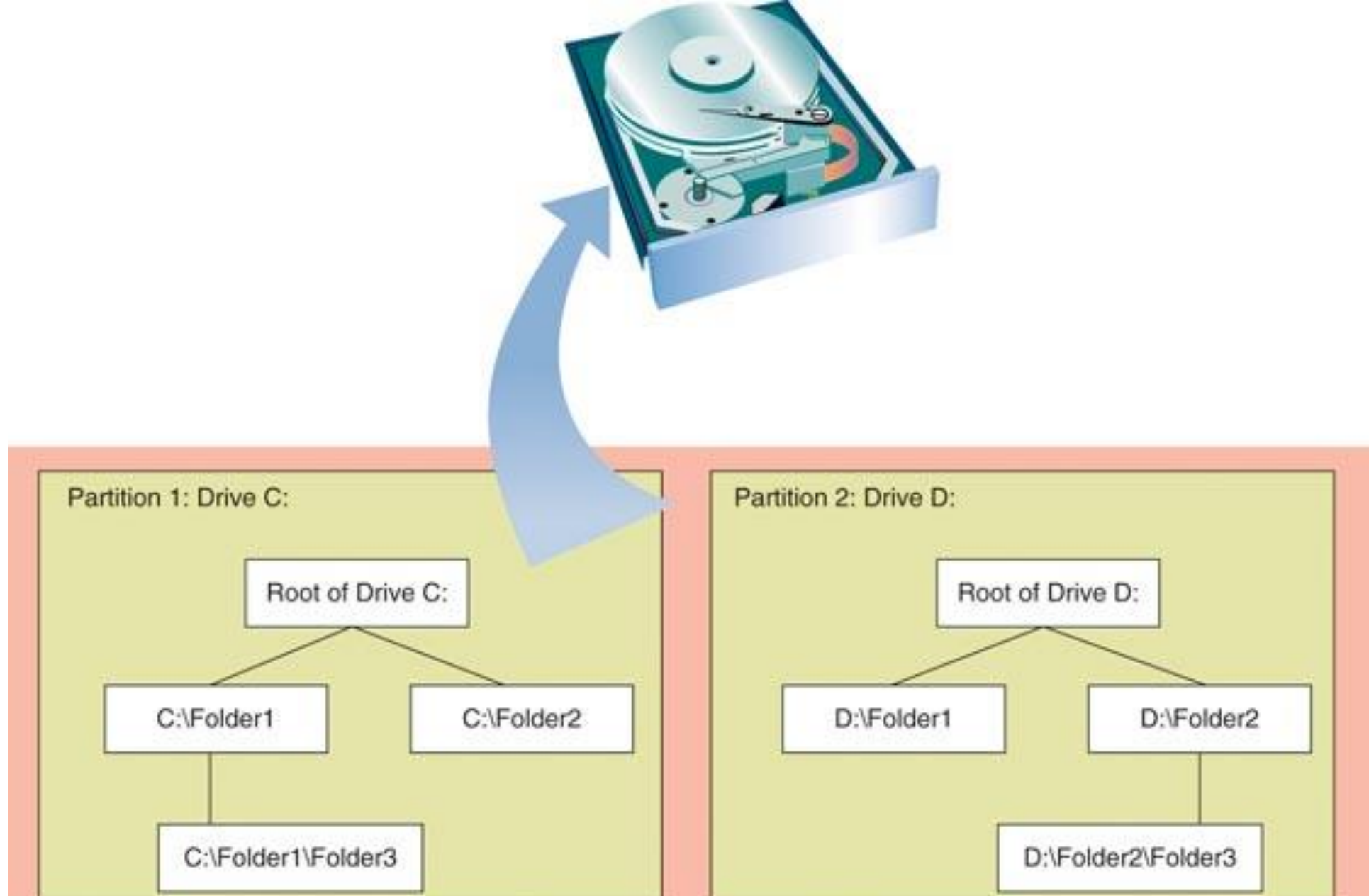
A hard drive is physical, partitions are logical

If you divide a physical disk into multiple partitions, each partition has its own root directory

Why would you want multiple partitions?

You can use multiple partitions to organize data; the OS is on the C drive and data is stored on a second partition

You can set up a dual or multi-boot system



A hard drive can be divided into 1 or more partitions that are identified by a letter such as drive C: or drive D:

- Someone who wants to hide data on a hard disk can create hidden partitions or voids—large unused gaps between partitions on a disk drive.
- For example, partitions containing unused space can be created between the primary partitions or logical partitions.
- This unused space between partitions is called the partition gap .
- It's possible to create a partition, add data to it, and then remove references to the partition so that it can be hidden in Windows.
- If data is hidden in this partition gap, a disk editor utility could be used to access it.

- With disk editing tools, however, you can access these hidden or empty areas of the disk.

- One way to examine a partition's physical level is to use a disk editor, such as WinHex or Hex Workshop.
- These tools enable you to view file headers and other critical parts of a file. Both tasks involve analyzing the key hexadecimal codes the OS uses to identify and maintain the file system.

- 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 starting at offset 0x1BE. The second partition starts at 0x1CE, the third partition starts at 0x1DE, and the fourth partition starts at 0x1EE

Examining FAT Disks

- File Allocation Table (FAT) is the file structure database that Microsoft designed for floppy disks. It's used to organize files on a disk so that the OS can find the files it needs.
- The 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 (archive, hidden, system, and read-only).

- There are three current versions of FAT—FAT16, FAT32, and exFAT (used for mobile personal storage devices).
- Three older FAT formats, which are FATX, Virtual FAT (VFAT), and FAT12.

Evolution of FAT versions:

- FAT12—This version is used specifically for floppy disks, so it has a limited amount of storage space. It was originally designed for MS-DOS 1.0, the first Microsoft OS, used for floppy disk drives and drives up to 16 MB.
- FAT16—To handle larger disks, Microsoft developed FAT16, which is still used on older Microsoft OSs, such as MS-DOS 3.0 through 6.22, Windows 95 (first release), and Windows NT 3.5 and 4.0. FAT16 supports disk partitions with a maximum storage capacity of 4 GB.

- FAT32—When disk technology improved and disks larger than 2 GB were developed, Microsoft released FAT32, which can access larger drives.
- exFAT—Developed for mobile personal storage devices, such as flash memory devices, secure digital eXtended capacity (SDCX), and memory sticks. The exFAT file system can store very large files, such as digital images, video, and audio files.

Table lists the number of sectors and bytes assigned to a cluster on FAT16 disk according to hard disk size.

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

- Microsoft OSs allocate disk space for files by clusters.
- This practice results in drive slack , composed of the 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 (found mainly in older Microsoft OSs) and file slack .

Deleting FAT Files

- When a file is deleted in Windows Explorer or with the MS-DOS delete command, the OS inserts a HEX E5 (0xE5) in the filename's first letter position in the associated directory entry. This value tells the OS that the file is no longer available and a new file can be written to the same cluster location.

- The area of the disk where the deleted file resides becomes unallocated disk space (also called “free disk space”). The unallocated disk space is now available to receive new data from newly created files or other files needing more space as they grow. Most forensics tools can recover data still residing in this area.

Examining NTFS Disks

- NTFS offers substantial improvements over FAT file systems. It provides more information about a file, including security features, file ownership, and other file attributes. With NTFS, you also have more control over files and folders (directories) than with FAT file systems.

- NTFS was Microsoft's move toward a journaling file system. The system keeps track of transactions such as file deleting or saving. This journaling feature is helpful because it records a transaction before the system carries it out. That way, in a power failure or other interruption, the system can complete the transaction or go back to the last good setting.

- In NTFS, everything written to the disk is considered a file.
- On an NTFS disk, the first data set is the Partition Boot Sector , which starts at sector [0] of the disk and can expand to 16 sectors.
- Immediately after the Partition Boot Sector is the Master File Table (MFT) .
- The MFT is the first file on the disk.

- An MFT file is created at the same time a disk partition is formatted as an NTFS volume and usually consumes about 12.5% of the disk when it's created. As data is added, the MFT can expand to take up 50% of the disk.

- An important advantage of NTFS over FAT is that it results in much less file slack space.
- Clusters are smaller for smaller disk drives. This feature saves more space on all disks using NTF

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

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 - files less than 512 bytes
 - Nonresident > 512 bytes
- 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
 - This cluster address are referred to as **data runs**

Table 5-5 Attributes in the MFT

Attribute ID	Purpose
0x10	<p>\$Standard Information</p> <p>This field contains data on file creation, alterations, MFT changes, read dates and times, and DOS file permissions.</p>
0x20	<p>\$Attribute_List</p> <p>Attributes that don't fit in the MFT (nonresident attributes) are listed here along with their locations.</p>
0x30	<p>\$File_Name</p> <p>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.</p>
0x40	<p>\$Object_ID (\$Volume_Version in Windows NT)</p> <p>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.</p>

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.

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

Attribute ID	Purpose
0xB0	<p>\$Bitmap</p> <p>A bitmap indicating cluster status, such as which clusters are in use and which are available.</p>
0xC0	<p>\$Reparse_Point</p> <p>This field is used for volume mount points and Installable File System (IFS) filter drivers.</p> <p>For the IFS, it marks specific files used by drivers.</p>
0xD0	<p>\$EA_Information</p> <p>For use with OS/2 HPFS.</p>
0xE0	For use with OS/2 HPFS.
0x100	<p>\$Logged_Utility_Stream</p> <p>This field is used by Encrypting File System (EFS) in Windows 2000 and later.</p>

- 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)**

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

Deleting NTFS Files

- Using File explorer, 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
 - Doesn't rename and move to recycle bin, but Eliminates the file from the MFT listing in the same way FAT does

Tasks during delete

1. Windows changes the filename and moves the file to a subdirectory with a unique identity in the Recycle Bin.
2. Windows stores information about the original path and filename in the **Info2file** , which is the control file for the Recycle Bin. It contains ASCII data, Unicode data, and the date and time of deletion for each file or folder.

Deleting NTFS Files

- Following steps apply when a user empties the Recycle Bin
 1. The associated clusters are designated as free—that is, marked as available for new data.
 2. The **\$Bitmap** file attribute in the MFT is updated to reflect the file's deletion, showing that this space is available.
 3. The file's record in the MFT is marked as being available.
 4. **VCN/LCN** cluster locations linked to deleted nonresident files are then removed from the original MFT record.
 5. A run list is maintained in the MFT of all cluster locations on the disk for nonresident files. When the list of links is deleted, any reference to the links is lost.

NTFS is more efficient than FAT at reclaiming deleted space. Deleted files are overwritten more quickly



The Resilient File System (ReFS) is Microsoft's newest file system, designed to maximize data availability, scale efficiently to large data sets across diverse workloads, and provide data integrity with resiliency to corruption

Resilient File System

- **Resilient File System (ReFS)** - designed to address very large data storage needs
 - Such as the cloud
-

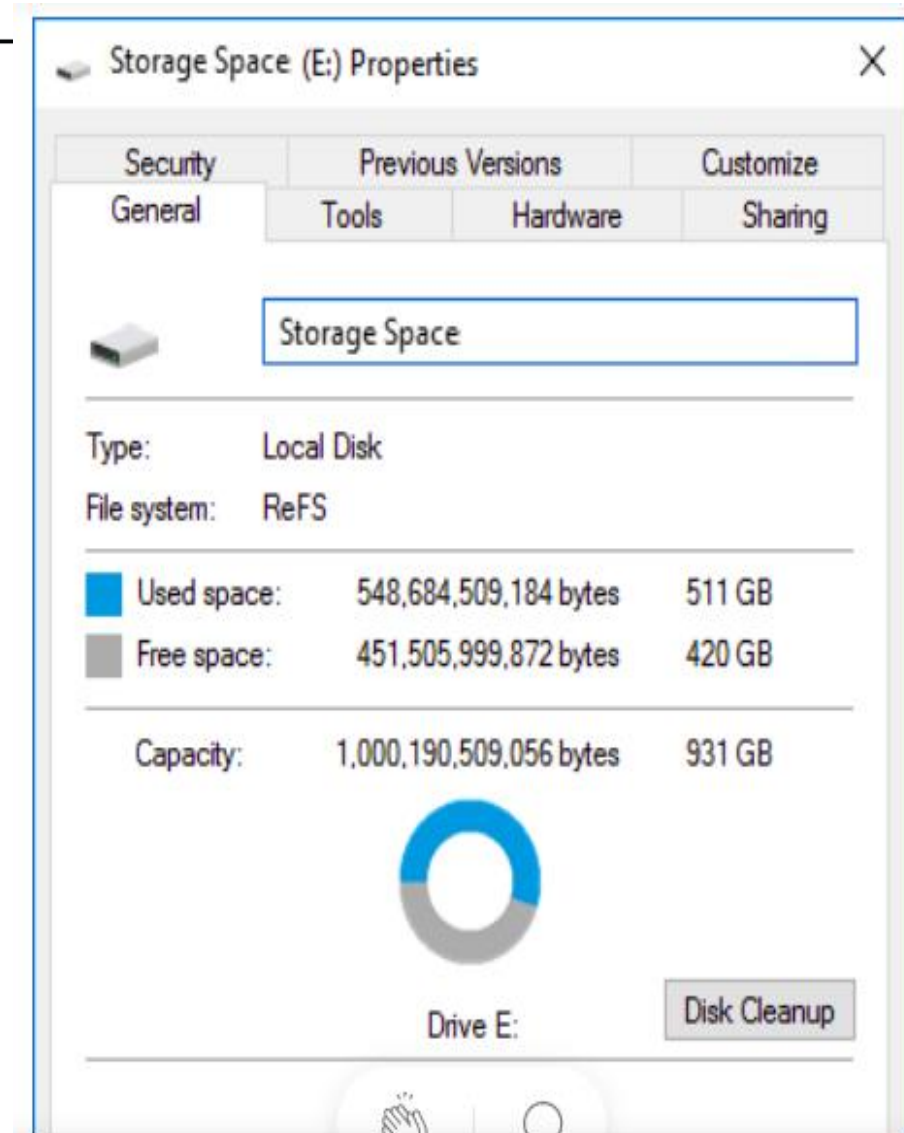
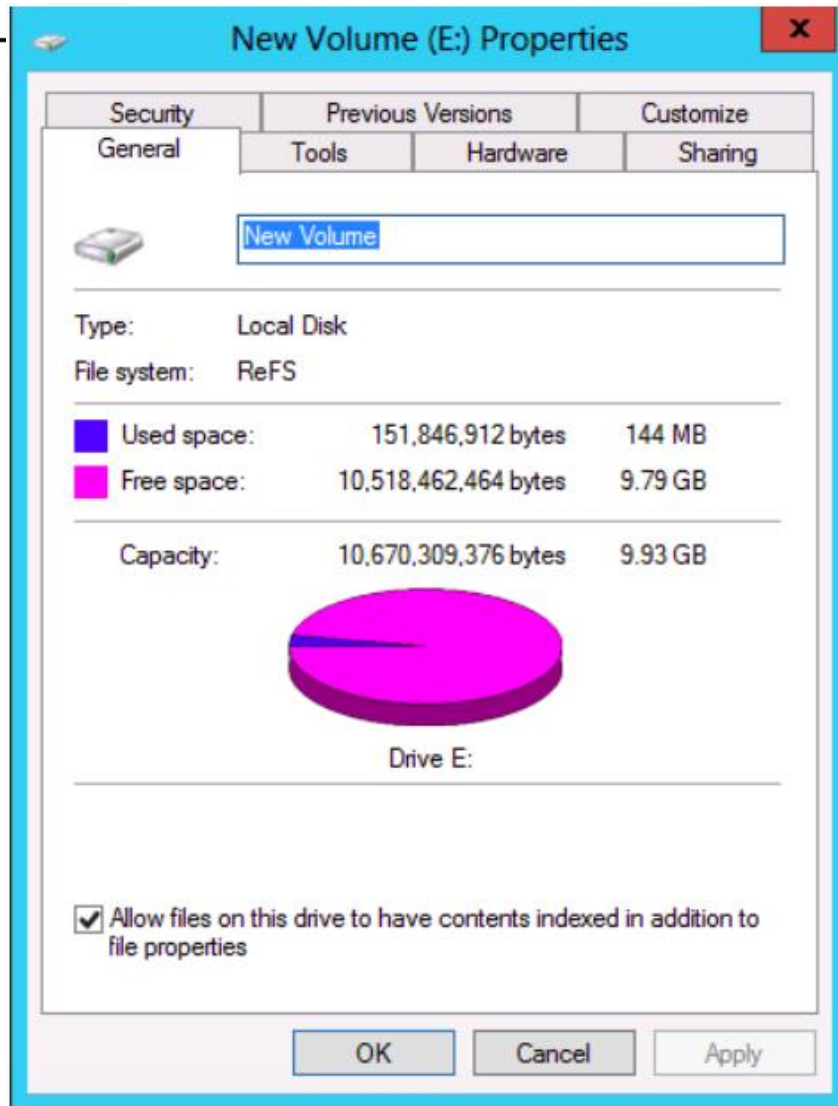
- Features incorporated into ReFS's design:

- Maximized data availability
- Improved data integrity
- Designed for scalability



- ReFS uses disk structures similar to the MFT in NTFS
- Its main intention for storage, so it cannot be used for BOOT drive
- Supported on Window 8/8.1 and above windows server 2012 and above
- its storage engine uses a B+ tree sort method for fast access to large data sets
- It uses a method called “allocate-on-write” that copies updates of data files to new locations, similar to shadow paging, it prevents overwriting the original data files
- The purpose of writing updates to new locations is to ensure that the original data can be recovered easily if a failure occurs in the update write to disk

In NTFS, file names are limited to 255 characters, while ReFS allows up to 32768 characters in a file name




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

Understanding Whole Disk Encryption

- Current whole disk encryption tools offer the following features:
 - Preboot authentication -single sign-on password, fingerprint scan, or token (USB device)
 - Full or partial disk encryption with secure hibernation
 - Advanced encryption algorithms
 - Key management function



 Windows 10
BitLocker

BitLocker Drive Encryption is a data protection feature that integrates with the operating system and addresses the threats of data theft or exposure from lost, stolen, or inappropriately decommissioned computers.

BitLocker provides the most protection when used with a Trusted Platform Module (TPM) version 1.2 or later. The TPM is a hardware component installed in many newer computers by the computer manufacturers. It works with BitLocker to help protect user data and to ensure that a computer has not been tampered with while the system was offline.

In addition to the TPM, BitLocker offers the option to lock the normal startup process until the user supplies a personal identification number (PIN) or inserts a removable device, such as a USB flash drive, that contains a startup key. These additional security measures provide multifactor authentication and assurance that the computer will not start or resume from hibernation until the correct PIN or startup key is presented.



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

BitLocker™ Drive Encryption

- BitLocker™ Drive Encryption gives you improved data protection on your Windows Vista and Windows Server codenamed “Longhorn” systems
 - Notebooks – Often stolen, easily lost in transit
 - Desktops – Often stolen, difficult to safely decommission
 - Servers – High value targets, often kept in insecure locations
 - All three can contain very sensitive IP and customer data
- Designed to provide a transparent user experience that requires little to no interaction on a protected system
- Prevents thieves from using another OS or software hacking tool to break OS file and system protections
 - Prevents offline viewing of user data and OS files
 - Provides enhanced data protection and boot validation through use of a Trusted Platform Module (TPM) v1.2