

# A Forensic Comparison of NTFS and FAT32 File Systems

---

Summer 2012

**Kelsey Laine Rusbarsky**

**#901-60-8173**

**FSC 630 Forensic Science Internship**

**MU Topic Advisor: Dr. Fenger**

**Internship Agency Supervisor (SSA Lou Ann Stovall, FBI KC Division, Director HARCFL, 816-584-6614 (office), [louann.stovall@ic.fbi.gov](mailto:louann.stovall@ic.fbi.gov))**

**Internship Agency (HARCFL, 4150 N. Mulberry Drive, Suite 250,  
Kansas City, MO 64116-1696, (816)584-4348 (fax))**

**Inclusive Dates: June 4<sup>th</sup>, 2012- August 10<sup>th</sup>, 2012**

**August 10<sup>th</sup>, 2012**

## **ABSTRACT**

The file system on any storage device is essential to the overall organization, storage mechanisms, and data control of the device. Knowing how these file systems work and the layout of key structures, storage mechanisms, associated metadata, and file system characteristics is essential to being able to forensically investigate a computer or other device. The New Technology File System (NTFS) and File Allocation Table (FAT32) are two key file systems that will be compared and contrasted, since both are still actively used and encountered often. Both systems offer forensic evidence that is significant and mandatory in an investigation.

## INTRODUCTION

The file system on any digital storage device is essential to the overall organization, storage mechanisms, and data control of the device. File systems allow computers and other similar digital devices to situate their data in different hierarchal structures through files and directories. Different file systems conduct these processes differently, and most often the file system can be utilized on multiple computers platforms. Even though a file system is usually not unique to a specific computer, a specific file system will have optimal functionality for certain computers and operating systems. Other types of storage devices that utilize file systems include; flash memory such as thumb drives, optical disks such as CD's and DVD's, floppy disks, and hard disk drives. A file system can be thought of as an index in a book, where the book can be broken down into sections and chapters. Without this breakdown of sections and chapters in a book, it would be nearly impossible to find the information that is stored. The same principle lies in the importance of file systems on a computer or storage device.<sup>1</sup>

To expand on the book analogy, just as books can divide into sections and chapters, so can the file system be organized into data categories. There are five main existing categories which are file system, content, metadata, file name, and application. Generally, the five categories are able to be applied to a majority of the file systems, though this model must be applied loosely to the FAT file system. The file system category can tell you where data structures are and how big the data structures are. This is the general information of the file system. The content category has the data that describes the actual content of the file and generally contains the majority of the file data. The content category is divided into virtual containers, which are usually the clusters or blocks of a hard drive. The metadata category describes and holds the, in layman's terms, "data

about data”. In other words, the metadata is the data that describes the file data. The location, size, time and date stamps, and access control is all recorded in the metadata category. The file name category is responsible for giving a name to each file. The file name acts as an address for the file. Rather than the user having to remember the address for the file, the file name takes the place of the numbered code, just as a social security number numerically represents a person’s name. Finally, the last category is the application category. The application category is not necessary for the organization or reading and writing of the files, but it is solely responsible for the special features in a file. An example of a special feature would be user quota statistics. Often the application category is not even utilized; this is the case for the FAT file system.<sup>1</sup>

All of the components of these file systems have the potential to provide forensic evidence in an investigation. Some of the characteristics are helpful to an investigation and some can hinder the investigation due to their properties or method of operation. Digital evidence submitted into court will need all of the metadata possible to support or deny a claim. For instance, metadata can identify whether an action was human or computer and determine whether something was a mistake, misunderstanding, or on purpose. Metadata can be used to investigate fraud, abuse, and system failures. It can also help establish elements such as causation, timing, extent of knowledge or *mens rea*, which means guilty mind. Metadata can reveal information about the creation, authorship, history, and intent of documents and files.<sup>14</sup>

The focus of this research is to differentiate and compare two file systems: NTFS (New Technology File System) and FAT (File Allocation Table), in seven areas. The seven areas are key structures, storage mechanisms, file names, directories, file date and time, file deletion,

encryption. The forensic implications of those areas will be discussed after each section. FTK Imager, a forensic extraction tool, will be utilized to give a visual of these differences between the file systems. By understanding the differences between these two file systems, it will be much easier to navigate and its use a forensic tool will be elevated. NTFS is a relatively newer file system, beginning with Windows NT and 2000, and has brought in many new features, including better metadata support and advanced data structures.<sup>2</sup> Some added features to NTFS are larger file size, large volume size, last accessed times for files, data access and organization efficiency.<sup>23</sup> FAT systems were originally used in DOS and Windows versions prior to windows XP. The “32” in FAT refers to the 32-bit numbers that represent the cluster values, which means that the table entry can have a maximum value of  $2^{32}$  values. Even though the FAT operating system is not utilized in many newer hard drives, it is still often used as a default file system in removable media and storage devices, as well as computers with multiple operating systems. FAT is good for these types of media because it is a very ubiquitous and versatile file system. FAT can also be easily joined with random operating systems, which is why the file system is simplistic when compared to NTFS.

## **MATERIALS AND METHODS**

AccessData Forensic Toolkit (FTK) Imager, Version 3.1.0.1514, © 2011 AccessData Group,

LLC

Toshiba Satellite Intel Celeron M Laptop

2 PNY 4GB Thumb Drives

To examine the two file systems on FTK Imager, two 4GB thumb drives were formatted, one for

NTFS and one for FAT32. Each were then imaged in FTK and compared to note the differences. The literature search was done through the Internet and related sources, such as operating system technical websites, technical journals, and a peer reviewed journal.

## **RESULTS and DISCUSSION**

### **Key Structures**

The data structures in a file system are important, because it organizes and sorts all of the files and their data in a certain way to create an efficient system. There are a number of different types of data structures and each structure is typically utilized for a specific file system. NTFS structure starts in the first sector with a 512 byte record known as the boot record. This record has boot codes, disk signatures (this maintains and identifies the partitions), and a table of primary partitions. It becomes important in file system forensics to be able to identify a correct partition and types of partitions.<sup>7</sup> One major difference between the NTFS and FAT data structures is that NTFS utilizes a Journaling File System.<sup>2</sup> A journaling file system keeps track of changes in the system by use of a journal. This allows for a quicker reboot if there is a system crash or power failure and protects files from becoming corrupted.<sup>21</sup> The NTFS log (\$LogFile) records any changes in the volume due to metadata (the data about data). However, the component that is the center of the NTFS file system is the Master File Table or MFT. MFT keeps data records of itself, so NTFS reserves the first 16 records for MFT data files. Any file names that start with a \$ are MFT stored metadata files. The remaining record is used for file and folder records. (See Figure 2 for an example of a MFT entry.) \$Mft contains a base file record for every file and folder in the NTFS volume. \$MftMirr is a duplicate image of the first

four records of \$Mft, and is a failsafe.<sup>3</sup> (See Figure 1 for metadata file descriptions.<sup>3</sup>) The key thing to remember with NTFS is that every entry is a file. Thus, with the MFT it will contain an entry from every file and directory. An MFT entry is set up so that the first 42 bytes house the data, which acts as a header, and the remainder of the entry is made up of attributes. An attribute contains the actual file data. Only resident data that is 900 bytes or smaller are stored in an attribute. A good place for data to be hidden here is at the end of an entry where there is unused space.<sup>8</sup> This unused space is called slack, which will be described later in detail. Each entry is usually 1,024 bytes long and defined in the boot sector. The attributes are organized into a B-tree structure, which allows NTFS to group or index files in large folders, minimizing the number of disk access. Disk access is a driver that helps enhance the system's BIOS. The B-tree structure has records which points to external clusters, which may contain more data files. This is above and beyond the FAT file system, which has to scan all file names in a large folder in order to create a file listing. This is also referred to as a binary tree.<sup>1,3</sup> (See Figure 4 to view a schematic on the NTFS Architecture.)

The FAT file system is very simple in comparison to the NTFS.<sup>6</sup> The FAT file system has two main data structures; a file allocation table (or FAT) and directory entries. Every file and directory is allocated a data structure, specifically a directory entry. These directory entries are stored in clusters and if more than one cluster is used, then the FAT data structure is used to locate the individual entries. The FAT table will identify the next cluster in a file. (See Figure 5 to view a schematic of the FAT Architecture and Figure 3 to view an example of a FAT32 layout.)

The NTFS organizational structure helps make locating data in the file system easy and straightforward. Useful information can be extracted from the MFT metadata files, including where to locate certain attributes and possibly hidden spaces such as slack space. But in order to understand where this data can be hidden, one must first understand the structure. The FAT32 file system is significantly less complicated. It relies mainly on its FAT table to locate data. It can be possibly more complicated to extract data from the FAT table, but the final result will typically yield the same data as NTFS.

### **Storage Mechanisms**

A disk drive can be described by the previous book model. The book represents a disk partition and as books can be a certain volume, so can a disk partition. A Disk Partition is created when a hard disk drive is divided up into logical storage units, or partitions. A Disk Partition can be viewed in two different places on the computer, one in disk management and the other in the command window. (See figures 8 and 9 to view disk partitions.) Within the partition are rings around the disk or tracks. A track is a physical division of data on a disk. These would be a section in the book. Inside the track lies clusters, which house sectors. These are chapters and paragraphs. The NTFS and FAT file systems both record data onto clusters which vary in size based on the volume size. NTFS utilizes smaller cluster sizes. With this, the smaller the cluster size the more efficient a disk can store information. This is due to unused space that is left on the end of the cluster when a small file is stored there. This is defined as slack, the term mentioned earlier. Slack space is the unused space at the end of a cluster that cannot be used by other data files. As noted in the key structures results, NTFS utilizes a Master File Table,

whereas FAT uses the directory entries and the file allocation table. When NTFS assigns data to a cluster, it starts at sector zero. \$Bitmap, an MFT metadata file, finds the first available cluster and assigns the file to that cluster. The \$BadClus MFT metadata file holds a listing that identifies any bad clusters, so that these will be avoided in the search for empty space.<sup>3,8</sup> (See figures 6 and 7 to see an example of the disk drive layout.)

The FAT file system does not start recording data at sector zero, but after a set number of reserved sectors and FAT areas.<sup>6</sup> This reserved sector starts at sector zero. The FAT area contains one or more FAT structures, in which there are always two copies stored in the file system. The data area contains the file content that will be stored in the cluster, and starts after the FAT area. The FAT structure has an entry for every cluster in the file system. If the table has an entry value of 0, then the cluster is not allocated to a file. If it contains the value, 0x0fff fff7, then the cluster has been determined to be damaged and is not used. FAT is harder to locate files than NTFS, because cluster addresses do not start at the beginning of the file system and must be found through the use of sector addresses. Since the data size does not always match the cluster size, there can be extra sectors at the end of a data area that are not part of the cluster or slack. It is these areas that can be used to hide data or store data that is intended to be hid. This area will not have a cluster address.<sup>1</sup>

The forensic significance of the storage mechanisms closely resembles that of the key structures. Since NTFS has reduced slack space due to its size control of the clusters, there is less potential for hidden data here. Whereas with the FAT file system, the cluster size space is typically larger and thus has more potential to hide the data in the slack space. The difficulty is that this is



unallocated space, meaning that the slack space has no addressing. Since FAT does not index the files, it will tend to fracture large files, which can cause some issues. However, FAT32 does typically have a mirrored copy of its file allocation table, which helps with data recovery.

## **File Names and Directories**

A file name or directory would be like the page number in a book. These page numbers link back to an index for easy access to the data of interest. NTFS stores and records file names in the MFT file record. When data is stored in the MFT file record it is known as a resident attribute. File names are always stored as a resident attribute. [If any records are contained in an external (non-resident) cluster, then the B-tree structure points to those data entries.] There are three attributes that are important in forensics and are used in the NTFS file system and contain much of the metadata that a MFT entry records. They are the \$STANDARD\_INFORMATION attribute, the \$FILE\_NAME attribute, and the \$DATA attribute. The \$STANDARD\_INFORMATION attribute contains all the core metadata for a file or directory. The \$FILE\_NAME attribute contains the file reference for the parent directory. The \$DATA attribute is used to store any sort of data and contains no specific values. NTFS also has the capability to support multiple data streams.<sup>10</sup> A data stream is defined as a sequence of bytes, where an application can write data a specific spots along the stream and every file has an associated unnamed stream assigned to it. NTFS, however, allows that file to have alternate data streams that can be aligned with the unnamed stream. This allows related data to be managed as a single unit.<sup>3</sup> (See Figures 10 and 11 to view FTK Imager snapshots of metadata files shown.)

FAT file systems are a little different. FAT will save the data under both an 8.3 file name and a long file name. An 8.3 filename is a compressed version of the long filename. MS-DOS uses the 8.3 file name to find and access a file. The 8.3 file name is saved in one or more secondary folder entries for the data file. Each folder entry holds 13 characters of the long file name. FAT does not allow for multiple data streams per file. Each file is only assigned a single data stream.<sup>6</sup> If you try to move or copy an alternate data stream to a FAT volume, then an error message will be displayed.<sup>3</sup>

The obvious significant component is the attributes in NTFS. These attributes house the locations and sizes of all these data records. Knowing how to utilize these attributes in a forensic investigation is priceless. Not only do they store this data, but even if the file is deleted, they can still be used to find and recover this data. The data streams are also useful for finding related data to a record of interest. The \$LogFile as described in figure 1, records transactions and entries in case of system failure. The information stored here can be valuable in a forensic investigation. FAT has a disadvantage here, because it simply tracks the files by the 8.3 filename. If able to view a FAT table, it may be hard to piece together a large file, or view the list in a coherent state.

### **File Date and Times**

A file date/time stamp is like a bookmark or ear-marked page, which identifies where the book was last opened. A file time represents the milliseconds elapsed since a certain time. In this case, the time is 12:00 am, January 1 1601, which is referred to as UTC or Coordinated

Universal Time. One thing to be careful about is that some file systems will log time according to local time. This can cause problems, because when a forensic examiner is examining a computer, he needs to know how the computer records time. Especially if there is a time difference from where the computer was confiscated to where it is analyzed. If a computer stores time with UTC, then the time differences do not matter. If they do not, then the examiner must be careful about recording date/time stamps. NTFS is not affected by these time differences, since, NTFS stores the file times in UTC. FAT stores file times based on the computer's local time.<sup>4</sup>

NTFS has four main time and date stamp attributes which are creation time, modified time, MFT entry modified time, and accessed time, or MACE.<sup>22</sup> The creation time is the time when the file was created. The modified time is the time that the content of the \$DATA and/or \$INDEX attributes were last modified. The MFT entry modified time is the time when the metadata of the file was last modified. The accessed time is the time that refers to when the content of the file was last accessed. The three main attributes discussed above are very important when considering date and time stamps. The \$STANDARD\_INFORMATION attribute is where the primary set of date and time stamps are located. The four date/time stamps are also recorded in the \$FILE\_NAME attribute, but usually correspond to when the file was created, renamed, or moved. The \$DATA attribute has no defined values.<sup>4,5</sup>

In a FAT file system, there are three date/time stamps that a directory entry utilizes; last accessed, last written, and created. The time values in the FAT file system is non-essential and could be false under some circumstances, because there are few requirements in the FAT

specification of date/time stamps. When FAT creates a directory entry for a new file it also creates a date/time stamp. This time value stays the same even if a copy of the original entry is moved to a different location. If a file is renamed or moved the original date/time stamp remains the same. The only exception is if the move is done from the windows 2000/XP system to a new or different volume, it will generate a new creation time. The written date/time stamps are created when new content is added to the file. The original written values stay with the data files, even when the file is moved or copied. The time is only updated when content is written or it is an automatic or manual save. If a file is moved, both creation and written values stay the same, but if you copy a file, the written time will stay the same and the creation time will be new. The accessed date/time stamp is accurate to the day, and also the time that is most frequently updated. So, if the file is opened or properties viewed a new access time will be created. Moving or copying the file will also update the time.<sup>1</sup> (See figures 12-15 to view FTK Imager snapshots of metadata shown.)

Some of the most significant forensic evidence is found in this category. If properly recorded and maintained, date/time stamps can reveal a lot about a specific file or folder. Both systems record these stamps, minus the entry modified time for FAT. Again the NTFS attributes house a lot of this metadata, plus additional data that FAT does not. Such metadata is permissions, \$Mft entry location, and that additional date/time stamp. Although FAT has these date/time stamps, like discussed above, they are not always true. One must take a careful approach in determining the validity of these stamps to assure accuracy.

## **File Deletion**

The best way to associate file deletion with a book would be to mark a page as unused or place an indicator on the corner, then roll the page towards the binding. The page is still there, but you can't read the page. The page would not be torn out until another page replaces it. When considering file deletion in NTFS, the master file table is the key structure. A MFT entry, as discussed above, is created for every file or folder. This MFT entry holds the metadata for that file, including location, time/date stamps, etc. The location is accurate down to the start cluster and how long the data file is. When a file is deleted in NTFS, a special indicator file is unmarked. This special indicator shows up as "used" when marked. This allows the file system to allocate another file to this location. However, the data is still recoverable in the file system until it is overwritten by new data. Recoverability in NTFS is generally better than in the FAT file system.<sup>12</sup> There are a number of issues that can cause the file to not be able to be deleted. One problem is that some files use an Access Control List (ACL), which means the user does not have the permission to delete the file. Some other issues that may exist is that the file is in use, there is file system corruption preventing access, or the file name has a reserved or invalid name.<sup>13</sup>

The FAT file system deletes files in a similar fashion to NTFS deletion system. When a FAT file is deleted the first character of the directory entry is replaced by a HEX E5h special character entry, which tells the operating system that the file can be ignored. Clusters that are assigned to the file as data entries are marked as "available" in the File Allocation Table. If new data is wrote over this area the old files can no longer be recovered or undeleted. Prior to rewriting data over the deleted information, undelete software can be used to recover old data. This data can

also be recovered manually. Another component that really assists in recovering intact data is that the file's data must be in consecutive clusters.<sup>12</sup> This is not a requirement, but greatly helps.

Both systems have the added bonus that the files can be recovered as long as the clusters have not been reused with new data entries. The file system permissions that NTFS has could be an advantage to forensic recovery, because it may block some deletion. However, in both cases, care must be taken so as not to overlook these files. The addresses of these files no longer "exist", so data may be found in the unallocated space or slack spaces.

## **Encryption**

Encryption for a book would be like a locked diary, with only one person with a special key able to unlock it. NTFS was designed with access control and security as a priority. NTFS utilizes an improved security system over FAT, which insures authorized access. However, this security only works properly if Windows is opened correctly. It is possible for a person to access NTFS by using a low-level disk utility, allowing a person to bypass all security measures.<sup>9</sup> The system put into place to accomplish the encryption is the Encryption File System or EFS. This encrypts the files and folders and allows the user access to these encrypted files when they log in. There are two encryption mechanisms used by the EFS; the "public" and "private" keys.<sup>11</sup> Each user has their own public and private key. The public key can be made known to others, and the private key is only for select individuals. When a file is encrypted, the EFS uses the public key. When the file needs to be decrypted, a private key must be used. NTFS utilizes the BitLocker Drive encryption system. BitLocker is a logical volume encryption system that allows for full

drive encryption capabilities. There are three authentication mechanisms that BitLocker uses, but are not essential to this discussion.<sup>20</sup>

The FAT file system was not designed for encryption and has no internal security measures. The only way to encrypt a FAT file is through external or third-party encryption programs. Since FAT is mostly used in storage devices like Thumb drives, it really has no need for a security system like NTFS, because it was designed for versatility. With a robust encryption system in place, FAT would not be able to go between systems as easily.

Obviously, gaining data from an unencrypted system is an advantage with the FAT32 file system. However, the NTFS encryption can help the evidence there to be preserved and not tampered with. The issue is having the appropriate permissions to access the file. With forensic tools though, this is not often an issue.

## **CONCLUSIONS**

When considering the FAT32 file system, it has many good qualities in areas other than the strong areas of NTFS. These qualities are such things as versatility and compatibility. FAT32 has very little security, and if one has access to the drive, can access any files or folders there. FAT32 is much more susceptible to disk errors and do not recover as readily as NTFS. FAT32 does not support file compression, which helps greatly with organization. Since NTFS allows smaller cluster sizes than FAT32, it wastes less disk space, and has less potential for hidden files. However, again FAT32 has its uses. It is compatible with any Windows Operating System, Apple's HFS and file system, and many nix file systems (ext 2/3/4)<sup>22</sup> and can be converted to

NTFS without reformatting. If NTFS were to be converted to FAT32 for some reason, the NTFS would have to be reformatted.<sup>17</sup>

NTFS was designed to be a robust file system. With its added features, such as, data streams, hierarchical storage, file compression and encryption, plus a very high performance level, NTFS has proved to be a very capable system.<sup>7</sup> However, if an older Windows system, earlier than Windows NT (2003), is used, NTFS may not be compatible with it. Also, older software programs may not be able to function with NTFS. Permissions are allowed in NTFS to control file and folder access, but this puts the chance for errors in the system way up.<sup>16</sup>

To visualize the future and most recent developments of the respective file systems, two file systems will be discussed; Extended File Allocation Table (exFAT) and the Resilient File System (ReFS).

ExFAT was introduced in November 2006, but is the most recent of the FAT file Systems. exFAT was designed as an upgrade and successor to the FAT file system family. As mobile technology grows and expands, file systems competent to support these devices are needed. ExFAT was designed with this in mind and has the capability of supporting large files for media storage. It also has seamless interoperability between computers and these devices, allowing easy copying and moving of files and folders. External media greater than 32GB can also be formatted with exFAT. ExFAT has some improvements over the FAT32 file system, while keeping the simplicity of the FAT family. Some of these improvements are support for large files and storage devices, support for performance improvement, support for future innovation,



and greater compatibility with flash media. In addition to this, exFAT adds in a cluster bitmap, per-file contiguous bit, better on-disk layout, and support for UTC time stamps. Metadata structures that are template-based were also added to support custom extensions.<sup>19</sup>

The Resilient File System was created for the new Windows 8 operating system coming out in October 2012. ReFS was built upon the foundation of NTFS, utilizing much of its features. Some features include BitLocker Encryption, access-control lists, USN (update sequence number) Journal or change journal (records changes in the volume), and file IDs (an archive of content description).<sup>18</sup> The change in the new file system comes in the on-disk store engine that implements on-disk structures like the MFT file table. Some of the features that are not supported in ReFS from NTFS is named streams, object IDs, short filenames, file compression, file level encryption, user data transactions, sparse files, hard-links, extended attributes, and disk quotas.

## **FUTURE RESEARCH**

Future research into different file systems will always be a necessity, especially in the forensics arena. With constantly changing technology, operating systems and file systems will be constantly updating to support the technology. As a forensic examiner it is important to keep up with and understand these new file systems. The next step is a look into ReFS, and how it compares to past file system forensic analysis. Also, since ReFS will be introduced with the new Windows 8 server, an analysis on data extracted from the OS could be looked at, with respect to the control and organization of the resilient file system.

## ACKNOWLEDGEMENTS

SSA Lou Ann Stovall, SA Cindy Smith, Sgt. Jeff Owen, RCFL Examiners, Josh Brunty, & Dr. Terry Fenger

## REFERENCES

1. Carrier, Brian. File System Forensic Analysis. Chapters 8-13. Pearson Education. 2005.
2. NTFS. Copyright 1998-2012. <http://www.ntfs.com> [accessed June 9<sup>th</sup>, 2012]
3. Windows Server. File System Technologies, NTFS Technical Reference. [http://technet.microsoft.com/en-us/library/cc778296\(v=ws.10\)](http://technet.microsoft.com/en-us/library/cc778296(v=ws.10)) [accessed June 14<sup>th</sup>, 2012]
4. Windows. File Times. [http://msdn.microsoft.com/en-us/library/windows/desktop/ms724290\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/ms724290(v=vs.85).aspx) [accessed July 3<sup>rd</sup>, 2012]
5. Where is Your Data?. Dates: NTFS Created, Modified, Accessed, Written. 2009. <http://whereismydata.wordpress.com/2009/02/14/dates-ntfs-created-modified-accessed-written/>. [accessed July 3<sup>rd</sup>, 2012]
6. Windows Server. File System Technologies, FAT Technical Reference. [http://technet.microsoft.com/en-us/library/cc758586\(v=ws.10\)](http://technet.microsoft.com/en-us/library/cc758586(v=ws.10)). [accessed June 14<sup>th</sup>, 2012]
7. Medeiros, Jason. NTFS Forensics: A Programmers View of Raw Filesystem Data Extraction. Grayscale Research. 2008. <http://grayscale-research.org/new/pdfs/NTFS%20forensics.pdf>. [accessed June 7<sup>th</sup>, 2012]
8. Kozierok, Charles M. The PC Guide. NTFS Architecture and Structures. Copyright 1997-2004. <http://www.PCGuide.com/ref/hdd/file/ntfs/arch.htm>. [accessed July 10<sup>th</sup>, 2012]
9. Kozierok, Charles M. The PC Guide. Other NTFS Features and Advantages, Encryption. Copyright 1997-2004. <http://www.PCGuide.com/ref/hdd/file/ntfs/other.htm>. [accessed July 12<sup>th</sup>, 2012]
10. Kozierok, Charles M. The PC Guide. NTFS Directories and Files. Copyright 1997-2004. <http://www.PCGuide.com/ref/hdd/file/ntfs/files.htm>. [accessed July 12<sup>th</sup>, 2012]
11. AccessData. Forensic Toolkit: Sales and Promotional Summary. AccessData Corp. [http://accessdata.com/media/en\\_us/print/techdocs/Forensic%20Toolkit.pdf](http://accessdata.com/media/en_us/print/techdocs/Forensic%20Toolkit.pdf) [accessed July 12<sup>th</sup>, 2012]
12. DIY DataRecovery. Undelete: deleted file recovery. Created 2006. [http://www.diydatarecovery.nl/kb\\_undelete\\_article.htm](http://www.diydatarecovery.nl/kb_undelete_article.htm) [accessed July 16<sup>th</sup>, 2012]
13. Microsoft Support. You cannot delete a file or folder on an NTFS file system volume. <http://support.microsoft.com/kb/320081> [accessed July 16<sup>th</sup>, 2012]
14. Ruhnka, John; Bagby, John. The CPA Journal, Forensic Uses of Metadata. June 2008. <http://www.nysscpa.org/cpajournal/2008/608/essentials/p68.htm> [accessed July 19<sup>th</sup>, 2012]
15. Forensic Data Recovery. Forensic Data Recovery vs Data Recovery. [http://www.cnwrecovery.com/html/forensic\\_dr.html](http://www.cnwrecovery.com/html/forensic_dr.html) [accessed July 19<sup>th</sup>, 2012]
16. Yousef, Mohammad. Tech Junkeez. File Systems Exposed (Part 2). August 2004. [http://www.techjunkeez.com/archive/general/file\\_systems\\_exposed\\_2.htm](http://www.techjunkeez.com/archive/general/file_systems_exposed_2.htm) [accessed July 24<sup>th</sup>, 2012]
17. Foley, Jim. The Elder Geek. FAT32 or NTFS: Making the Choice. Copyright 2002-2011.

- [http://www.theelderageek.com/ntfs\\_or\\_fat32\\_file\\_system.htm](http://www.theelderageek.com/ntfs_or_fat32_file_system.htm) [accessed July 24<sup>th</sup>, 2012]
18. MSDN Blogs. Building Windows 8: An Inside Look from the Windows Engineering Team. Building the next generation file system for Windows: ReFS. Pub. January 16<sup>th</sup>, 2012. <http://blogs.msdn.com/b/b8/archive/2012/01/16/building-the-next-generation-file-system-for-windows-refs.aspx> [accessed July 24<sup>th</sup>, 2012]
  19. Microsoft Support. Description of the exFAT file system driver update package. <http://support.microsoft.com/kb/955704> [accessed July 25<sup>th</sup>, 2012]
  20. Microsoft Windows. BitLocker Drive Encryption. Copyright 2012. <http://windows.microsoft.com/en-us/windows-vista/Bitlocker-Drive-Encryption-Overview> [accessed July 25<sup>th</sup>, 2012]
  21. Corbet, Jonathan. Barriers and Journaling Filesystems. Copyright 2008. <http://lwn.net/Articles/283161/> [accessed June 9<sup>th</sup>, 2012]
  22. Brunty, Josh. NTFS Filesystem PowerPoint. Fall 2012.
  23. Fenger, Terry, Ph.D. NTFS (New Technology File System) Foundations and Fundamentals. Fall 2011.

## FIGURES

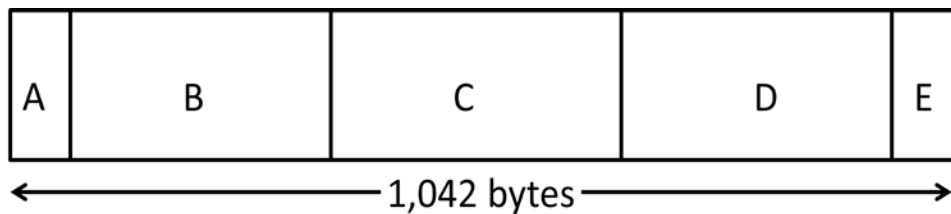
**Figure 1: Table of Metadata File Types**

System File	File Name	MFT Record	Purpose of the File
Master file table	\$Mft	0	Contains one base file record for each file and folder on an NTFS volume. If the allocation information for a file or folder is too large to fit within a single record, other file records are allocated as well.
Master file table mirror	\$MftMirr	1	Guarantees access to the MFT in case of a single-sector failure. It is a duplicate image of the first four records of the MFT.
Log file	\$LogFile	2	Contains information used by NTFS for faster recoverability. The log file is used by Windows Server 2003 to restore metadata consistency to NTFS after a system failure. The size of the log file depends on the size of the volume, but you can increase the size of the log file by using the Chkdsk command.
Volume	\$Volume	3	Contains information about the volume, such as the volume label and the volume version.
Attribute definitions	\$AttrDef	4	Lists attribute names, numbers, and descriptions.
Root file name index	.	5	The root folder.
Cluster bitmap	\$Bitmap	6	Represents the volume by showing free and unused clusters.
Boot sector	\$Boot	7	Includes the BPB used to mount the volume and additional

			bootstrap loader code used if the volume is bootable.
Bad cluster file	\$BadClus	8	Contains bad clusters for a volume.
Security file	\$Secure	9	Contains unique security descriptors for all files within a volume.
Uppcase table	\$Uppcase	10	Converts lowercase characters to matching Unicode uppercase characters.
NTFS extension file	\$Extend	11	Used for various optional extensions such as quotas, reparse point data, and object identifiers.
		12–15	Reserved for future use.

Windows Server. File System Technologies, NTFS Technical Reference.  
[http://technet.microsoft.com/en-us/library/cc778296\(v=ws.10\)](http://technet.microsoft.com/en-us/library/cc778296(v=ws.10)) [accessed 7/24/2012]

**Figure 2: MFT Entry**



- ‘A’ represents a 42 byte header that contains entry data
- ‘B’, ‘C’, and ‘D’ represent attributes
- An attribute contains the actual resident file data
- ‘E’ = “Slack Space”

Figure made in Microsoft word 8/6/2012

**Figure 3: FAT Structure**

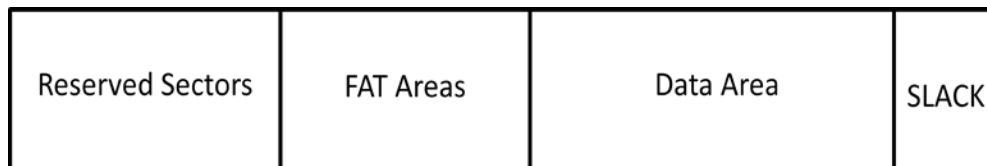
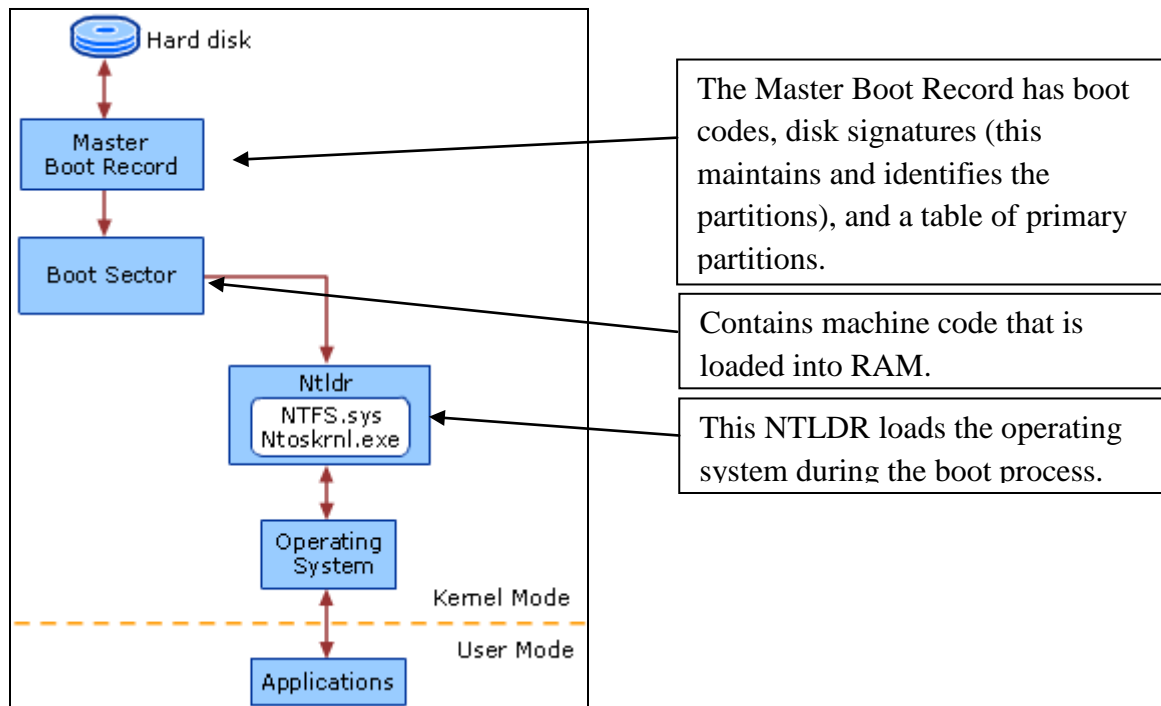


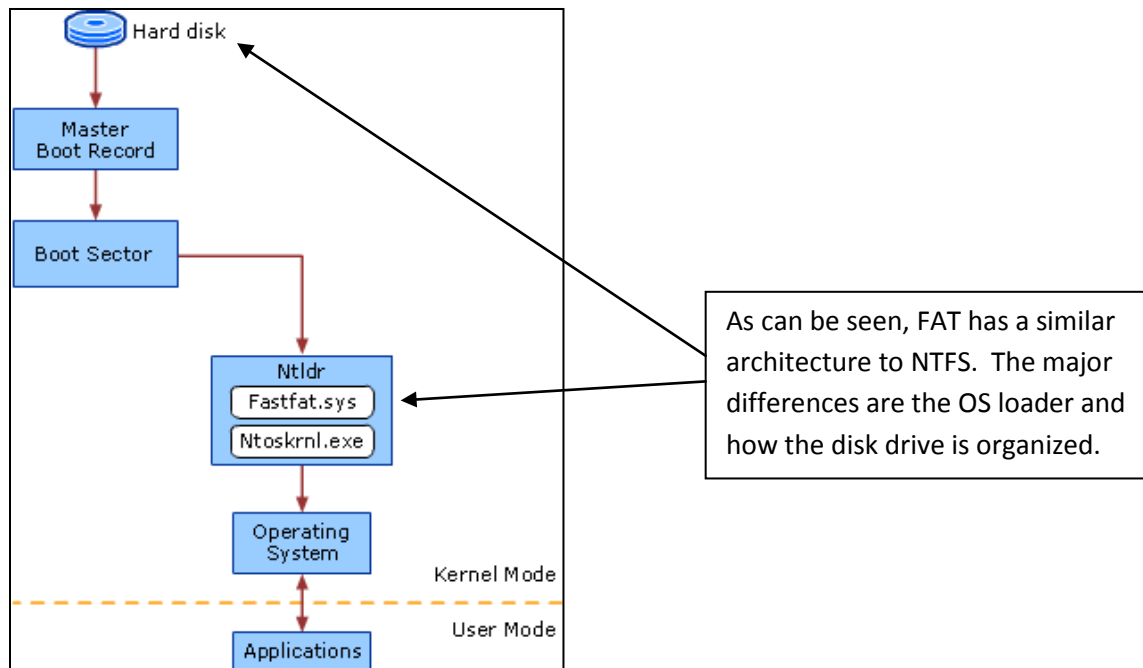
Figure made in Microsoft word 8/6/2012

**Figure 4: NTFS Architecture**



Windows Server. File System Technologies, NTFS Technical Reference.  
[http://technet.microsoft.com/en-us/library/cc778296\(v=ws.10\)](http://technet.microsoft.com/en-us/library/cc778296(v=ws.10)) [accessed 6/21/2012]

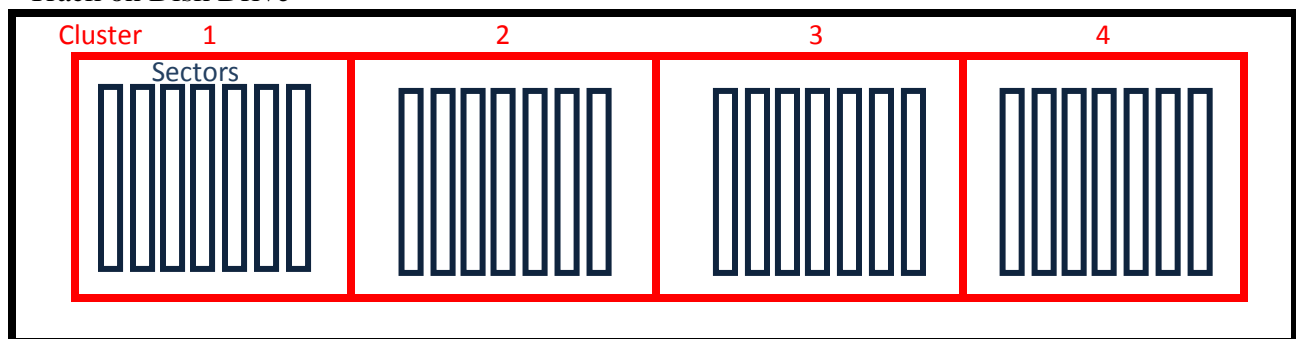
**Figure 5: FAT Architecture**



Windows Server. File System Technologies, FAT Technical Reference.  
[http://technet.microsoft.com/en-us/library/cc758586\(v=ws.10\)](http://technet.microsoft.com/en-us/library/cc758586(v=ws.10)). [Accessed 6/21/2012]

**Figure 6: Example of a Disk Drive**

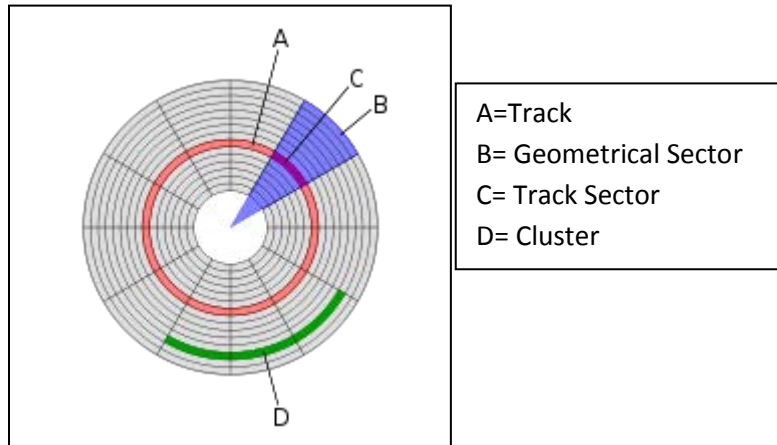
Track on Disk Drive



- Track- A circular track on a disk drive that is a physical division of data.
- Cluster (Size changes based on file system)
- Sector- subdivision of a track.

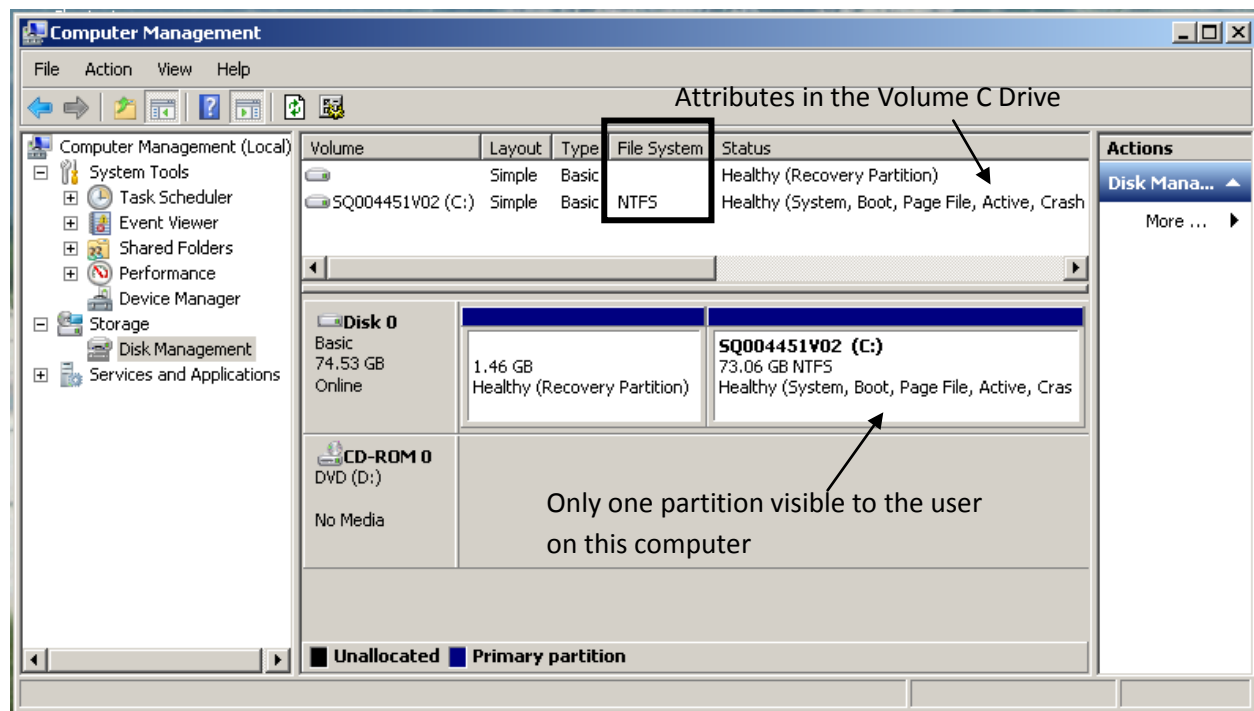
Figure made in Microsoft word 7/24/2012

**Figure 7: Hard Disk Model**

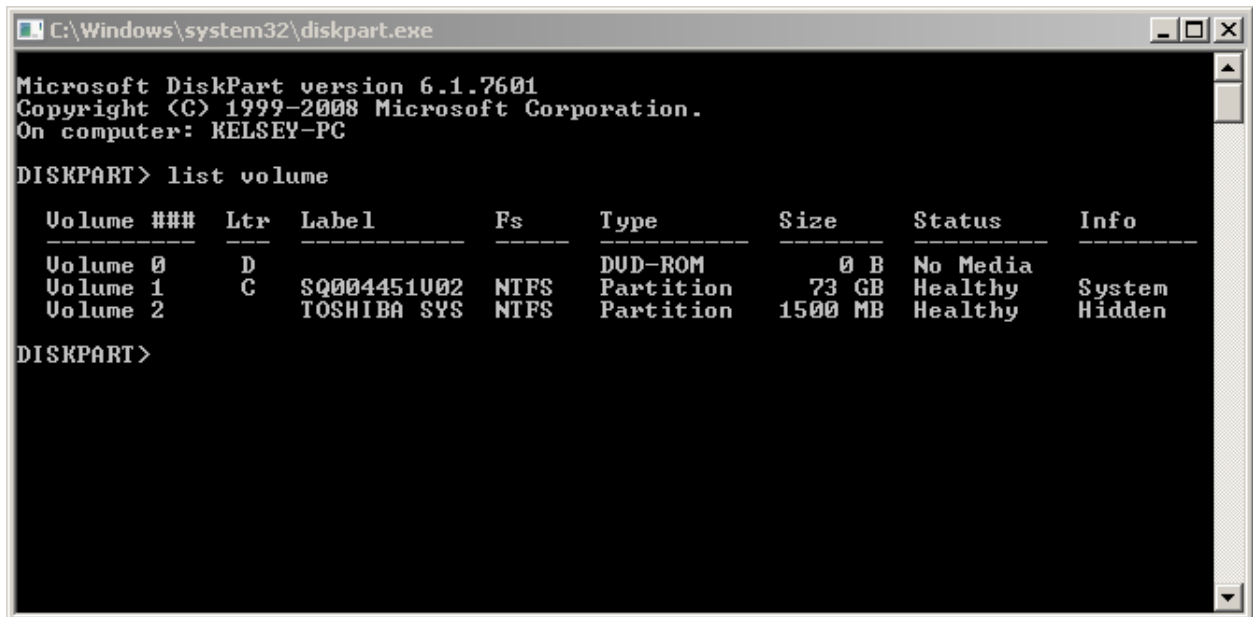


Tracks (Disk Drive). Updated July 2012. [http://en.wikipedia.org/wiki/Track\\_\(disk\\_drive\)](http://en.wikipedia.org/wiki/Track_(disk_drive)) [accessed 7/18/2012]

**Figure 8: Disk Partition for a Windows 7 OS with NTFS File System**



Screen Shot off of Toshiba Laptop 7/25/2012

**Figure 9: Disk Partition description**

```
C:\Windows\system32\diskpart.exe
Microsoft DiskPart version 6.1.7601
Copyright (C) 1999-2008 Microsoft Corporation.
On computer: KELSEY-PC

DISKPART> list volume

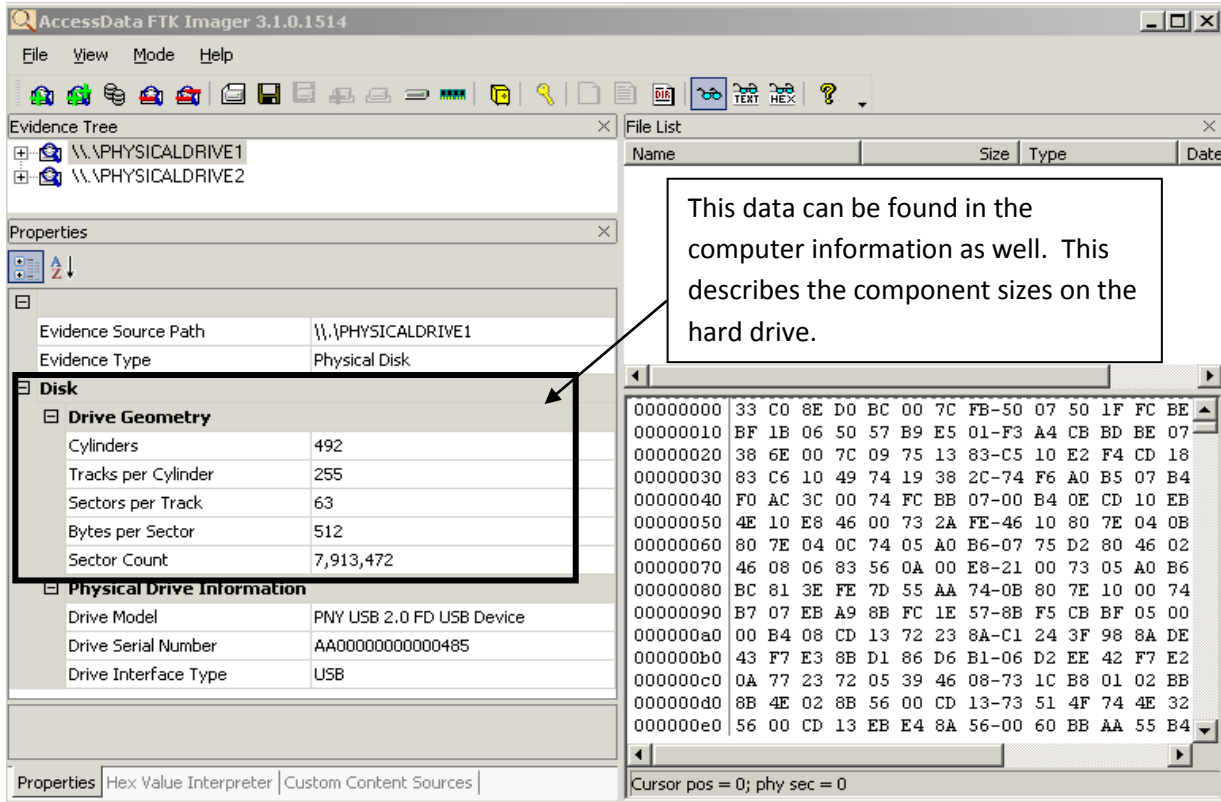
  Volume ###  Ltr  Label              Fs          Type          Size      Status       Info
  -----  -  -  -  -  -  -  -  -  -
  Volume 0                D                DUD-ROM      0 B         No Media
  Volume 1                C  SQ004451V02      NTFS        Partition    73 GB      Healthy     System
  Volume 2                TOSHIBA SYS      NTFS        Partition    1500 MB    Healthy     Hidden

DISKPART>
```

Can remove mount point or drive letter here. Volume 2 is most likely the Recovery partition shown in the Disk Manager. Screen Shot off of Toshiba Laptop 7/25/2012

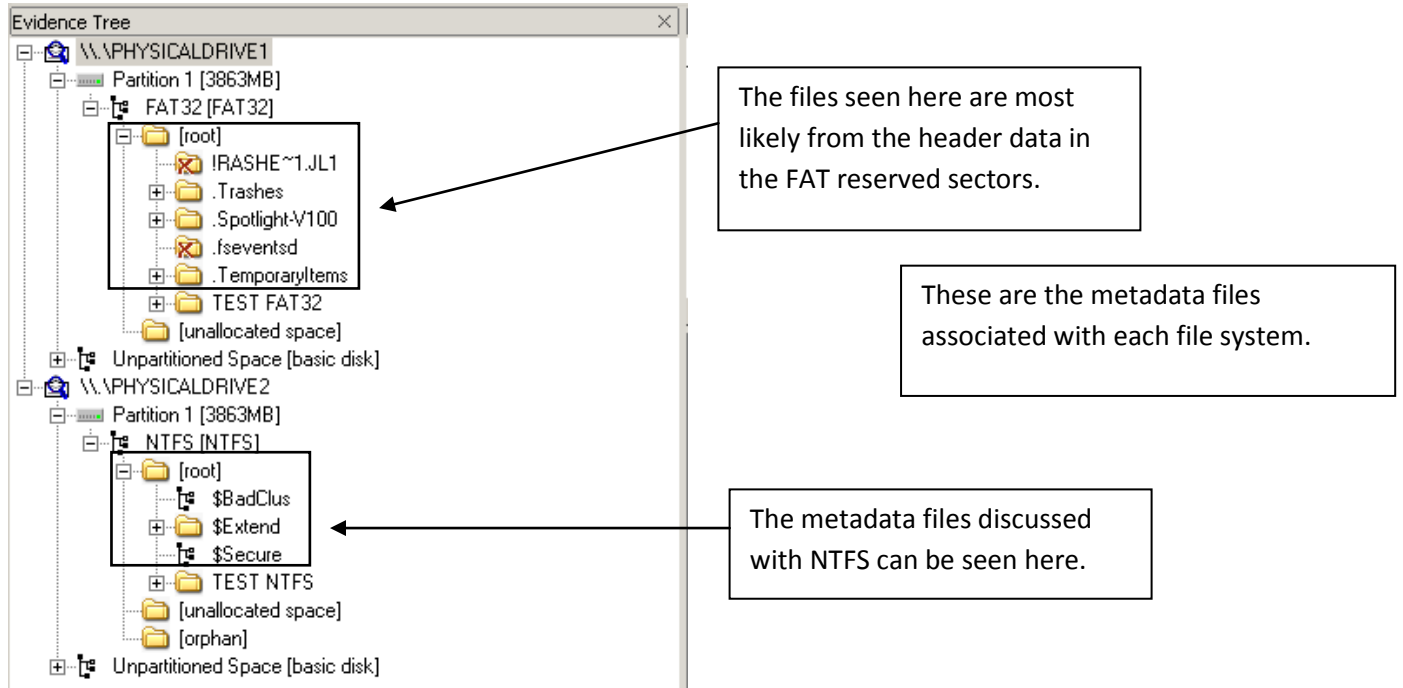


Figure 10: Thumb drive disk description in FTK Imager



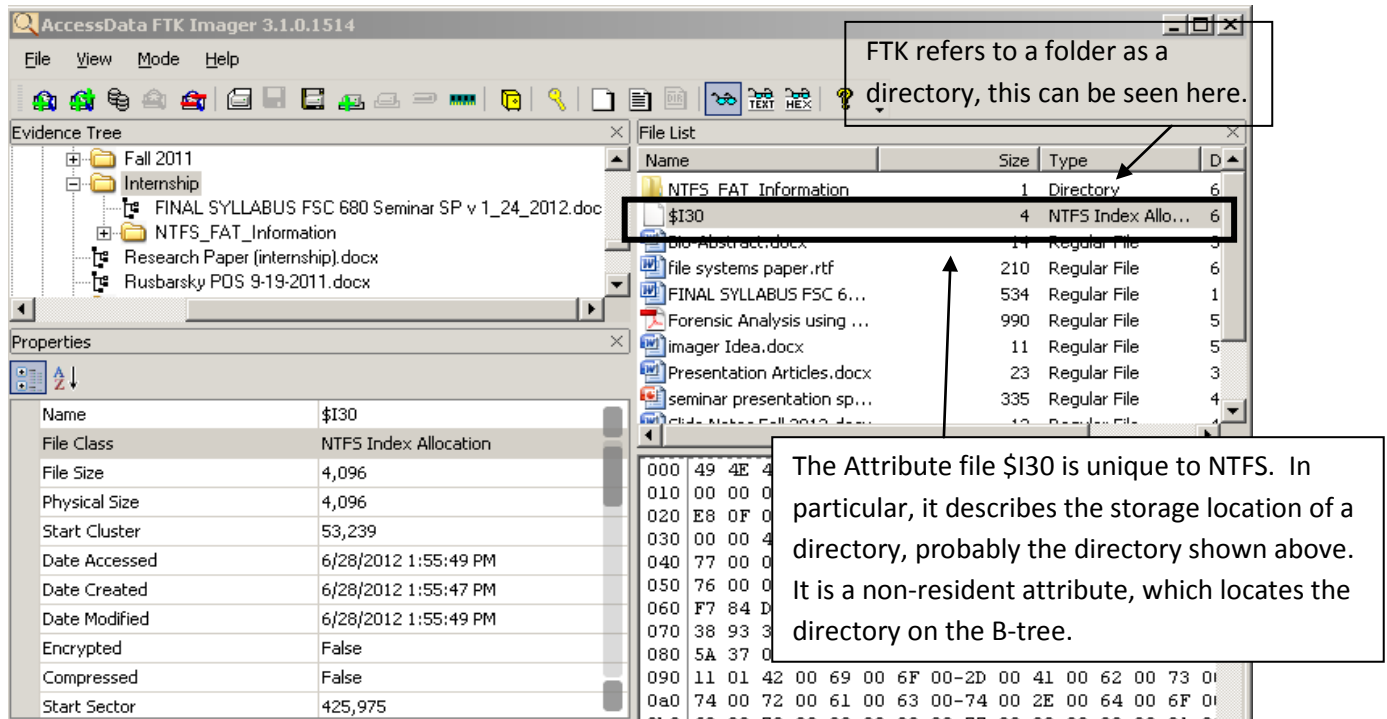
Screen Shot of FTK Imager screen off of Toshiba Laptop 8/6/2012

**Figure 11: Root Directory File types displayed in FTK Imager**



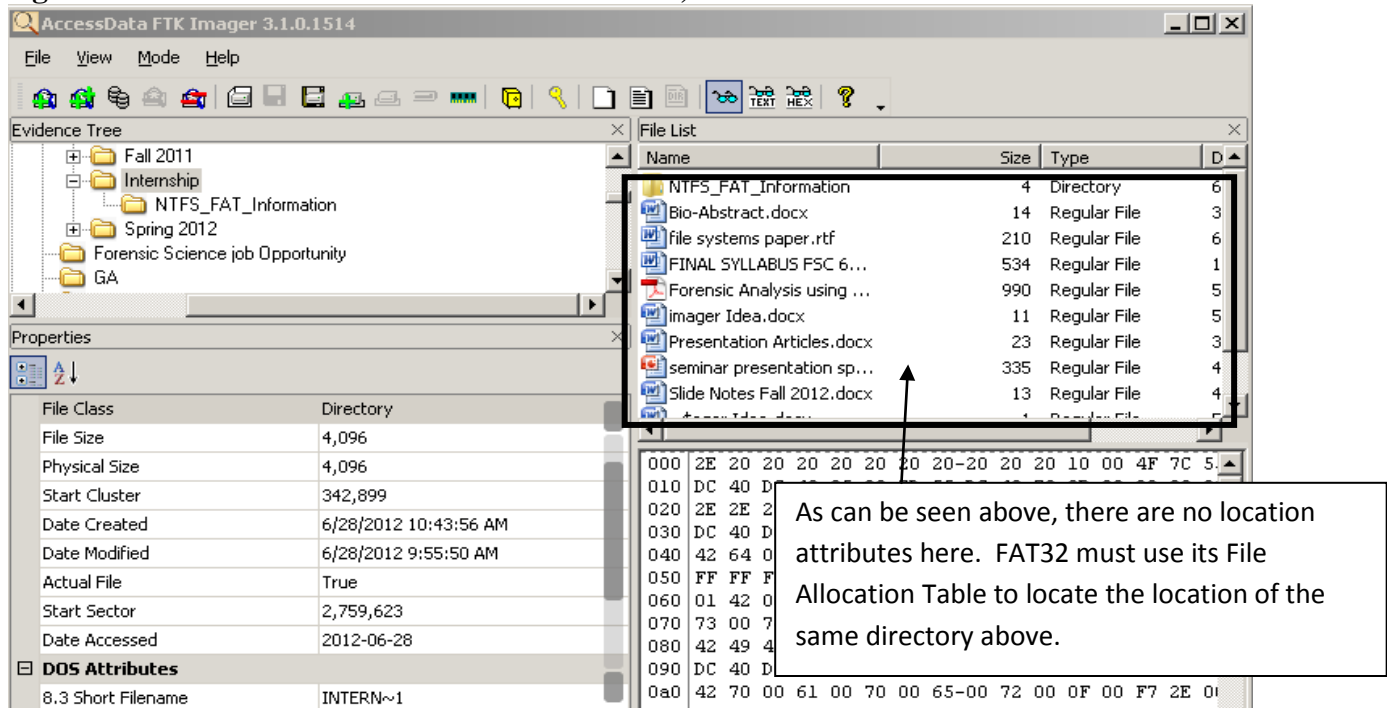
Screen Shot of FTK Imager screen off of Toshiba Laptop 8/6/2012

**Figure 12: Metadata File associated with NTFS in FTK Imager**



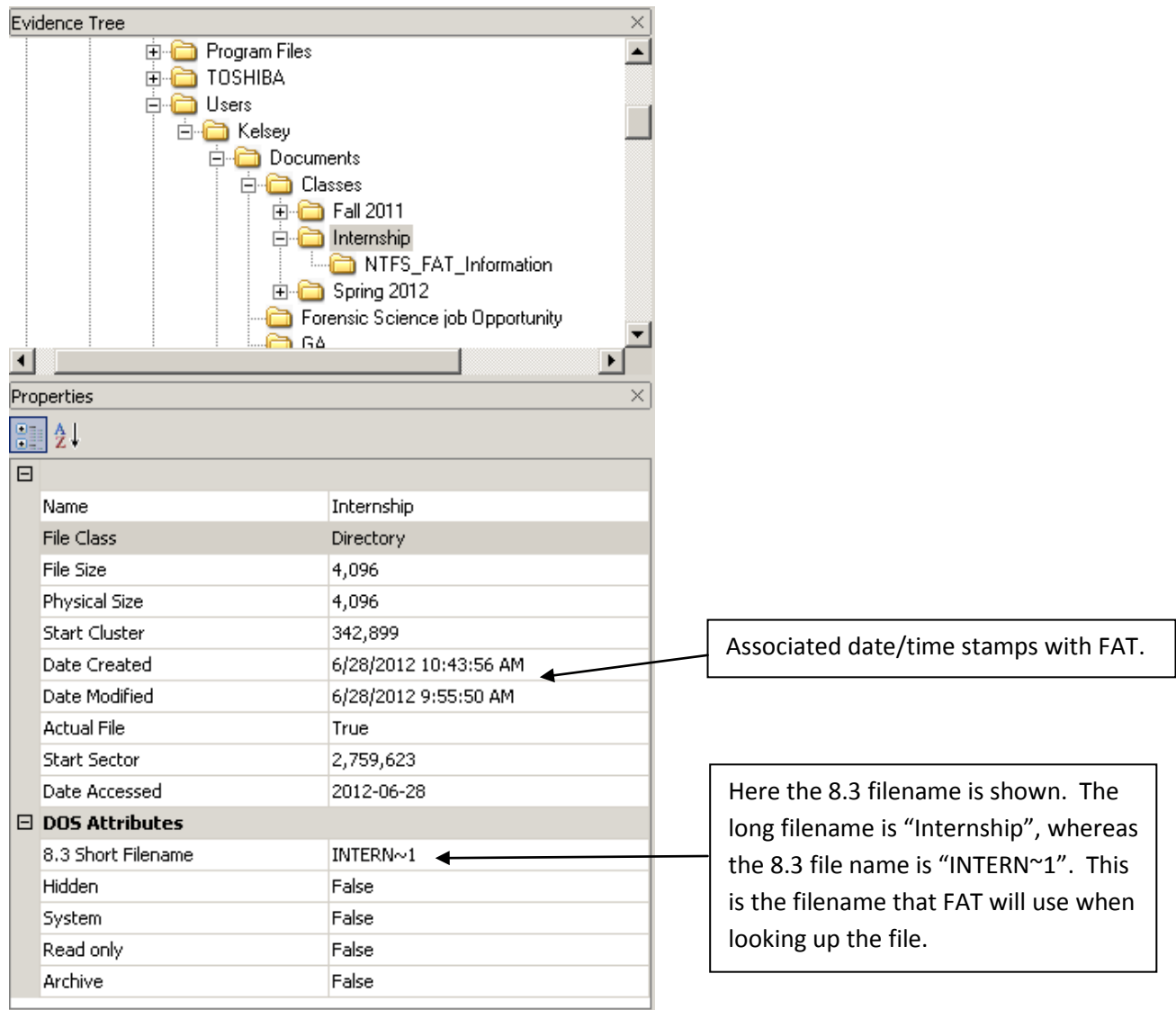
Screen Shot of FTK Imager screen off of Toshiba Laptop 8/6/2012

**Figure 13: No metadata files associated with FAT; all are stored in reserved sectors**



Screen Shot of FTK Imager screen off of Toshiba Laptop 8/6/2012

Figure 14: Data associated with the FAT File System



Screen Shot of FTK Imager screen off of Toshiba Laptop 8/6/2012

**Figure 15: Data associated with the NTFS File System**

Name	Internship
File Class	Directory
File Size	56
Physical Size	56
Date Accessed	6/28/2012 1:55:49 PM
Date Created	6/28/2012 1:55:47 PM
Date Modified	6/28/2012 1:55:49 PM
Encrypted	False
Compressed	False
Actual File	True
Alternate Data Stream Count	1
<b>DOS Attributes</b>	
8.3 Short Filename	INTERN~1
Hidden	False
System	False
Read only	False
Archive	False
<b>NTFS Information</b>	
MFT Record Number	118 (120832)
Record date	6/28/2012 1:55:49 PM
Resident	True
Offline	False
Sparse	False
Temporary	False
Owner SID	S-1-5-21-1854686018-192265998-178;
Group SID	S-1-5-21-1854686018-192265998-178;
<b>NTFS Access Control Entry</b>	
ACE Type	Allow Access
Inheritable	True
SID	S-1-1-0
Name	Everyone
Access Mask	001f01ff
Traverse Folder	True
List Folder	True
Create Files	True
Create Folders	True
Delete Subfolders and Files	True
Delete	True
Read Permissions	True
Change Permissions	True
Take Ownership	True

As can be seen, a lot more metadata is extracted from NTFS.

An additional date/time stamp.

The 8.3 filename

The MFT record number, which identifies where the \$MFT metadata file is located.

All of the file permissions, encryption information, and security.

Screen Shot of FTK Imager screen off of Toshiba Laptop 8/6/2012