FILE SYSTEM FORENSICS-ntfs8gb.dd

Master Boot Record

The Master Boot Record is created when the disk is partitioned.

It contains a

- small amount of executable code called the master boot code, and
- the partition table for the disk.
- a 2-byte structure called a signature word or end of sector marker, which is always set to 55 AA.

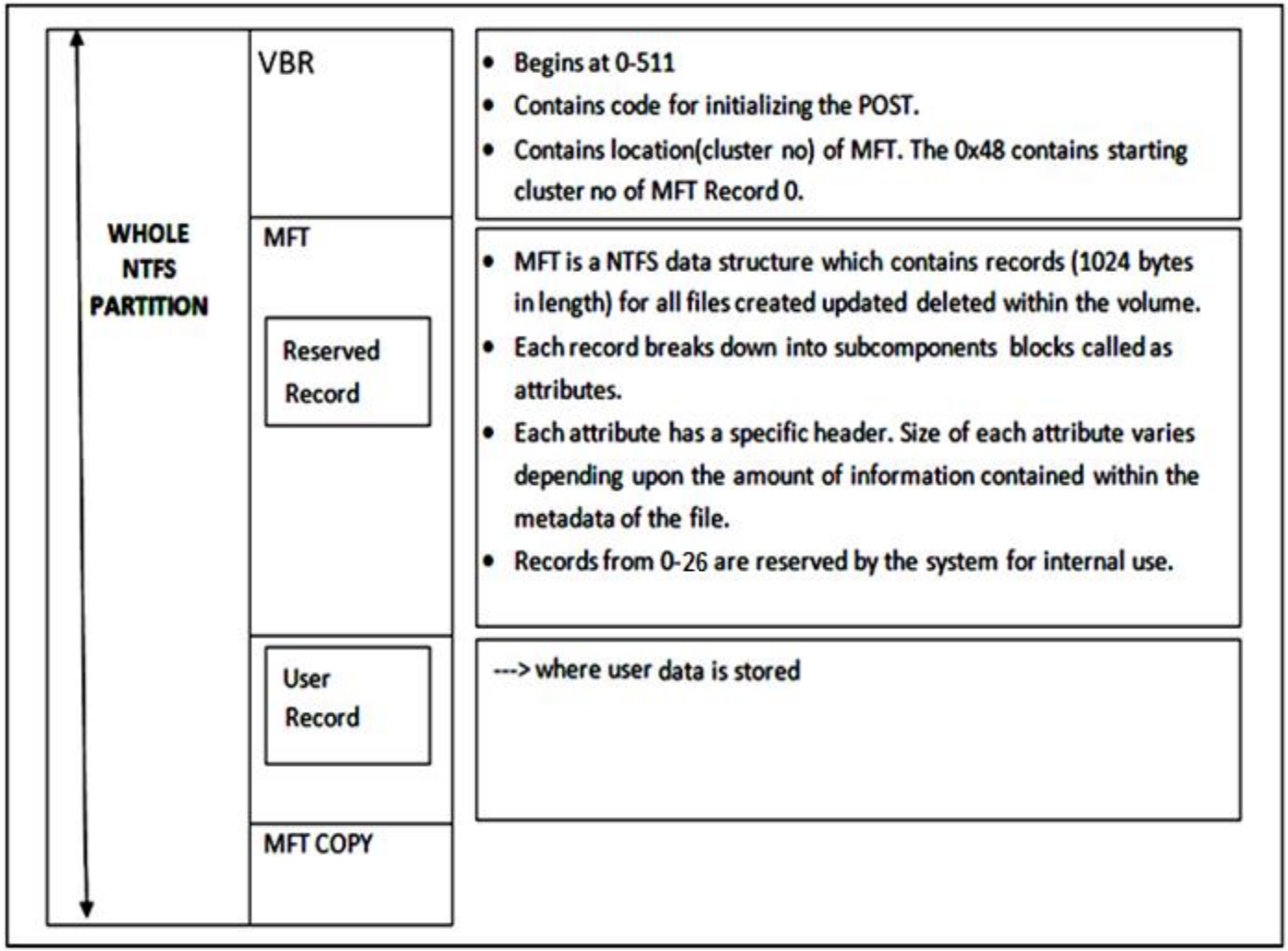
The Master Boot Record (MBR), is located at **sector 0 of cylinder 0, head 0,** of the first physical sector of a hard disk and is not part of any partition.

The master boot code performs the following activities:

- Scans the partition table for the active partition
- Finds the starting sector of the active partition
- Loads a copy of the Volume Boot Record from the active partition into memory
- Transfers control to the executable code in the volume boot record.

New Technology File System (NTFS)

NTFS is made up of several components including: **a partition boot sector** ; the **master file table** that stores a record of all files and directories in the filesystem; A series of **metadata files** that help structure meta data more efficiently.NTFS supports **multiple data streams**, allowing more than one data sequence to be associated with a single file. NTFS uses **file locks** to manage access to the data streams.



Name	MFT Record No.
$MFT	0
$MFTMirr	1
$LogFile	2
$Volume	3
$AttrDef	4
$	5
$Bitmap	6
$Boot	7
$BadClus	8
$Secure	9
$Upcase	10
$Extend	11
	12-15
$Quota	24
$ObjId	25
$Reparse	26

MBR -- ntfs.dd

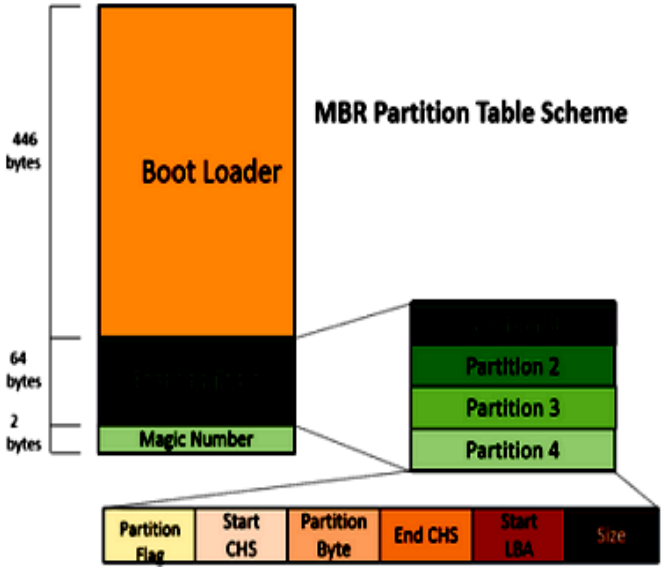
```
Offset(h)  00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F  Decoded text

000000000  41 4B 45 4F FC 31 C0 FA 8E D0 BC 00 7C FB 89 E6  AKEOü1ÀúŽÐ¼.|û‰æ
000000010  89 E7 1E 06 8E D8 BB 13 04 8B 07 48 89 07 C1 E0  ‰ç..ŽØ»..‹.H‰.Áà
000000020  06 2D C0 07 8E C0 B9 00 02 F3 A4 50 68 30 7C CB  .-À.ŽÀ¹..ó¤Ph0|Ë
000000030  8E D8 66 31 DB 8E C3 41 BA 81 00 E8 89 00 72 6D  ŽØf1ÛŽÃAº..è‰.rm
000000040  BB BE 7D B9 04 00 26 80 3F 00 7C 09 75 05 83 C3  »¾}¹..&€?.|.u.ƒÃ
000000050  10 E2 F3 EB 58 BE 94 7D E8 DA 00 E8 CA 00 BA 5A  .âóëX¾"}èÚ.èÊ.ºZ
000000060  7D BE 6E 7D E8 A0 00 B4 01 CD 16 75 3D B4 02 CD  }¾n}è .´.Í.u=´.Í
000000070  16 24 04 75 38 80 3E 93 7D 00 7F 0B BE B4 7D E8  .$.u8€>"}...¾´}è
000000080  B3 00 C6 06 93 7D 12 80 3E 92 7D 00 75 D9 E8 89  ³.Æ."}.€>'}.uÙè‰
000000090  00 C6 06 BE 7D 81 68 80 00 BA 72 7D BE 7E 7D E8  .Æ.¾}.h€.ºr}¾~}è
0000000A0  65 00 5A 07 1F EA 00 7C 00 00 E8 6D 00 E8 78 00  e.Z..ê.|..èm.èx.
0000000B0  BB BE 7D 8B 17 52 B2 80 8B 4F 02 66 8B 5F 08 E8  »¾}‹.R²€‹O.f‹_.è
0000000C0  05 00 73 D5 07 1F CB 60 B4 41 BB AA 55 CD 13 72  ..sÕ..Ë`´A»ªUÍ.r
0000000D0  2C 81 FB 55 AA 75 26 F7 C1 01 00 74 20 61 1E 66  ,.ûUªu&÷Á..t a.f
0000000E0  31 C0 8E D8 66 50 66 53 50 68 00 7C 40 50 6A 10  1ÀŽØfPfSPh.|@Pj.
0000000F0  89 E6 B4 42 CD 13 9F 83 C4 10 9E 1F C3 61 BB 00  ‰æ´BÍ.Ÿƒ Ä.ž.Ãa».
000000100  7C B8 01 02 CD 13 C3 FA 8B 1C 26 66 8B 07 66 89  |¸..Í.Ãú‹.&f‹.f‰
000000110  04 26 89 17 26 8C 4F 02 FB C3 FA BB 20 00 66 A1  .&‰.&ŒO.ûÃú» .f¡
000000120  6E 7D 26 66 89 07 FB C3 B4 01 CD 16 74 06 B4 00  n}&f‰.ûÃ´.Í.t.´.
000000130  CD 16 E2 F4 C3 AC 3C 00 74 09 B4 0E BB 07 00 CD  Í.âôÃ¬<.t.´.»..Í
000000140  10 EB F2 C3 50 2E A0 BE 7D 80 FA 80 75 04 88 C2  .ëòÃP. ¾}€ú€u.ˆÂ
000000150  EB 06 38 C2 75 02 B2 80 58 C3 FA 2E 80 3E 92 7D  ë.8Âu.²€XÃú.€>'}
000000160  00 74 0A 2E FE 0E 93 7D 2E FE 0E 92 7D EA 20 00  .t..þ."}.þ.'}ê .
000000170  00 00 9C 2E FE 06 91 7D 75 03 E8 C7 FF 9A 4C 00  ..œ.þ.'}u.èÇÿšL.
000000180  00 00 9C 2E FE 0E 91 7D 79 03 E8 B7 FF 9D CA 02  ..œ.þ.'}y.è·ÿ.Ê.
000000190  00 FF 49 12 0D 0A 50 72 65 73 73 20 61 6E 79 20  .ÿI...Press any
0000001A0  6B 65 79 20 74 6F 20 62 6F 6F 74 20 66 72 6F 6D  key to boot from
0000001B0  20 55 53 42 2E 00 00 00 A3 00 7E 01 00 00 80 20   USB....£.~...€
0000001C0  21 00 07 FE FF AE 00 08 00 00 00 44 E7 00 00 00  !..þÿ®.....Dç...
0000001D0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000001E0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000001F0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 55 AA  ..............Uª
000000200  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000210  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000220  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

0x 0800 * 0x200 = 0x100000

Structure of a master boot record

Address			description		Size in bytes
Hex	Oct	Dec			
0000	0000	0	Code area		440 (max.446)
01B8	0670	440	Disk signature (optional)		4
01BC	0674	444	Usually nulls; 0x0000		2
01BE	0676	446	Table of primary partitions (Four16-byte entries, IBM partition table scheme)		64
01FE	0776	510	55h	MBR signature	2
01FE	0777	511	AAh		
			MBR, total size:446+64+2=		512



The Volume Boot Record:

- A Volume Boot Record (VBR) (also known as a Master Boot Sector, a partition boot record or a partition boot sector) is a type of boot sector introduced by the IBM Personal Computer.

- The VBR occupies the first partition sector i.e. VBR is located at logical sector zero in the active partition and the operating system loader (NTLDR up to and including Windows XP, winload.exe and the Windows Boot Manager in Vista onwards) occupy subsequent sectors.

- VBR is found on a partitioned data storage device, such as a hard disk, a floppy disk, and contains machine code for bootstrapping programs stored in other parts of the device.

- On non-partitioned storage devices, it is the first sector of the device.

- On partitioned devices, it is the first sector of an individual partition on the device, with the first sector of the entire device being a Master Boot Record (MBR) containing the partition table.

VOLUME BOOT RECORD / MASTER BOOT SECTOR

Offset (h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
00007E00	EB	52	90	4E	54	46	53	20	20	20	20	00	02	08	00	00	ëR.NTFS .....
00007E10	00	00	00	00	00	F8	00	00	3F	00	FF	00	3F	00	00	00	.....ø..?.ÿ.?...
00007E20	00	00	00	00	80	00	80	00	D8	A6	3F	01	00	00	00	00	....€.€.Ø¦?.....
00007E30	00	00	0C	00	00	00	00	00	6D	FA	13	00	00	00	00	00	.....mú.....
00007E40	F6	00	00	00	01	00	00	00	0D	68	14	F4	7F	14	F4	52	ö.....h.ô..ôR
00007E50	00	00	00	00	FA	33	C0	8E	D0	BC	00	7C	FB	B8	C0	07	....ú3ÀŽÐ¼.|û¸À.
00007E60	8E	D8	E8	16	00	B8	00	0D	8E	C0	33	DB	C6	06	0E	00	ŽØè..¸..ŽÀ3ÛÆ...
00007E70	10	E8	53	00	68	00	0D	68	6A	02	CB	8A	16	24	00	B4	.èS.h..hj.ËŠ.$.´
00007E80	08	CD	13	73	05	B9	FF	FF	8A	F1	66	0F	B6	C6	40	66	.Í.s.¹ÿÿŠñf.¶Æ@f
00007E90	0F	B6	D1	80	E2	3F	F7	E2	86	CD	C0	ED	06	41	66	0F	.¶Ñ€â?÷â†ÍÀí.Af.
00007EA0	B7	C9	66	F7	E1	66	A3	20	00	C3	B4	41	BB	AA	55	8A	·Éf÷áf£ .Ã´A»ªUŠ
00007EB0	16	24	00	CD	13	72	0F	81	FB	55	AA	75	09	F6	C1	01	.$.Í.r..ûUªu.öÁ.
00007EC0	74	04	FE	06	14	00	C3	66	60	1E	06	66	A1	10	00	66	t.þ...Ãf`..f¡..f
00007ED0	03	06	1C	00	66	3B	06	20	00	0F	82	3A	00	1E	66	6A	....f;. ..‚:..fj
00007EE0	00	66	50	06	53	66	68	10	00	01	00	80	3E	14	00	00	.fP.Sfh....€>...
00007EF0	0F	85	0C	00	E8	B3	FF	80	3E	14	00	00	0F	84	61	00	….…è³ÿ€>....„a.
00007F00	B4	42	8A	16	24	00	16	1F	8B	F4	CD	13	66	58	5B	07	´BŠ.$...‹ôÍ.fX[.
00007F10	66	58	66	58	1F	EB	2D	66	33	D2	66	0F	B7	0E	18	00	fXfX.ë-f3Òf.·...
00007F20	66	F7	F1	FE	C2	8A	CA	66	8B	D0	66	C1	EA	10	F7	36	f÷ñþÂŠÊf‹ÐfÁê.÷6
00007F30	1A	00	86	D6	8A	16	24	00	8A	E8	C0	E4	06	0A	CC	B8	..†ÖŠ.$.ŠèÀä..Ì¸
00007F40	01	02	CD	13	0F	82	19	00	8C	C0	05	20	00	8E	C0	66	..Í..‚..ŒÀ. .ŽÀf
00007F50	FF	06	10	00	FF	0E	0E	00	0F	85	6F	FF	07	1F	66	61	ÿ...ÿ....…oÿ..fa
00007F60	C3	A0	F8	01	E8	09	00	A0	FB	01	E8	03	00	FB	EB	FE	Ã ø.è.. û.è..ûëþ
00007F70	B4	01	8B	F0	AC	3C	00	74	09	B4	0E	BB	07	00	CD	10	´.‹ð¬<.t.´.»..Í.
00007F80	EB	F2	C3	0D	0A	41	20	64	69	73	6B	20	72	65	61	64	ëòÃ..A disk read
00007F90	20	65	72	72	6F	72	20	6F	63	63	75	72	72	65	64	00	error occurred.
00007FA0	0D	0A	4E	54	4C	44	52	20	69	73	20	6D	69	73	73	69	..NTLDR is missi
00007FB0	6E	67	00	0D	0A	4E	54	4C	44	52	20	69	73	20	63	6F	ng...NTLDR is co
00007FC0	6D	70	72	65	73	73	65	64	00	0D	0A	50	72	65	73	73	mpressed...Press
00007FD0	20	43	74	72	6C	2B	41	6C	74	2B	44	65	6C	20	74	6F	Ctrl+Alt+Del to
00007FE0	20	72	65	73	74	61	72	74	0D	0A	00	00	00	00	00	00	restart.....
00007FF0	00	00	00	00	00	00	00	00	83	A0	B3	C9	00	00	55	AA	.....ƒ ³É..Uª

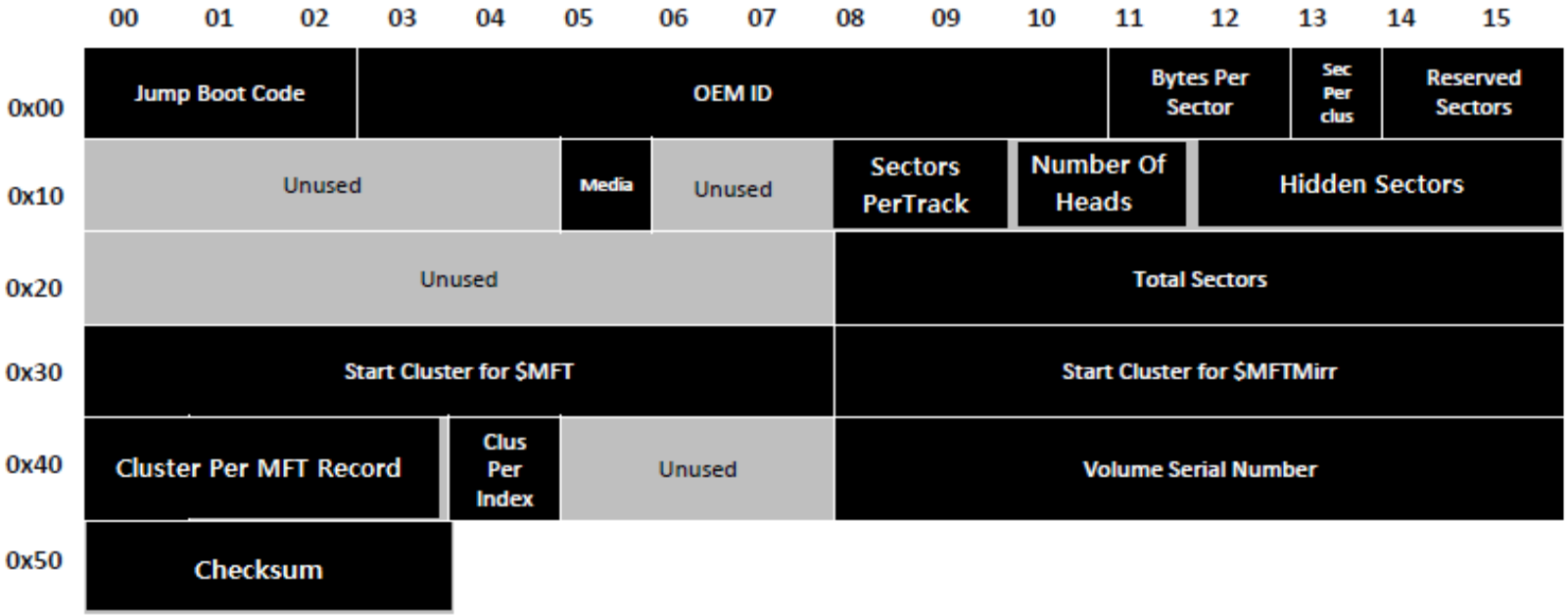
- Jump Instruction
- OEM ID
- BIOS Parameter Block
- Boot Code
- Error Message
- Message Offset
- Signature

Volume Boot Record – ntfs8gb.dd

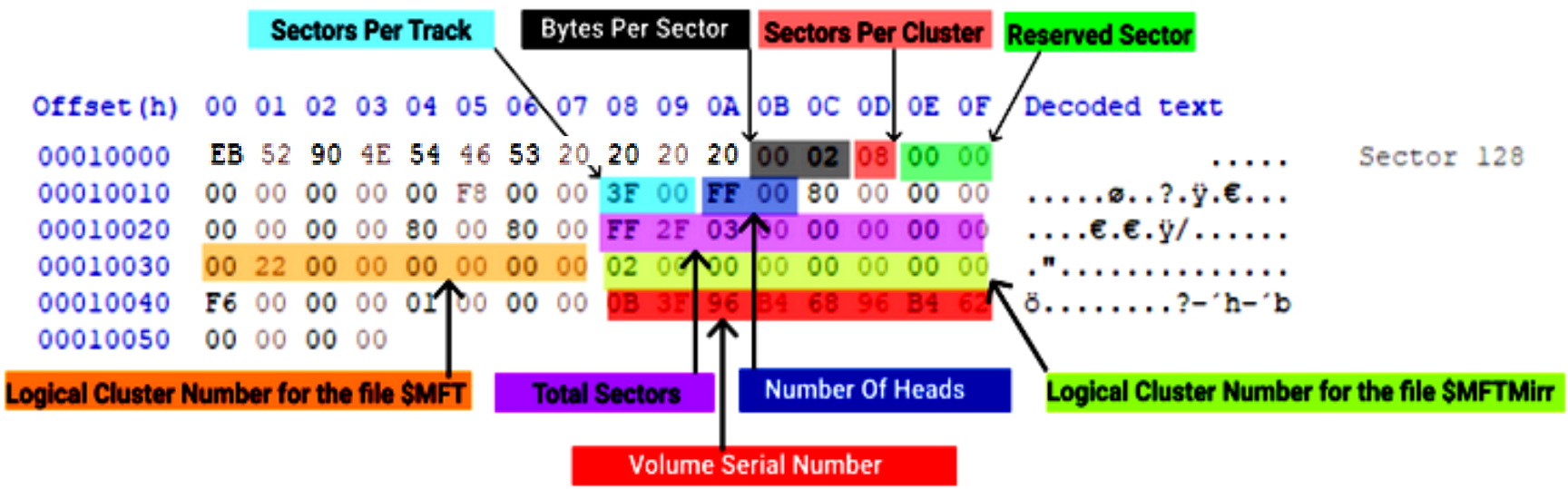
```
Offset(h)  00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F  Decoded text

0000FFFF0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000100000  EB 52 90 4E 54 46 53 20 20 20 20 00 02 08 00 00  ëR.NTFS    .....
000100010  00 00 00 00 00 F8 00 00 3F 00 FF 00 00 08 00 00  .....ø..?.ÿ.....
000100020  00 00 00 00 80 00 00 00 FF 43 E7 00 00 00 00 00  ....€...ÿCç.....
000100030  00 00 0C 00 00 00 00 00 02 00 00 00 00 00 00 00  ................
000100040  F6 00 00 00 01 00 00 00 FD 5C 5D AE 94 5D AE 26  ö.......ý\]®"]®&
000100050  00 00 00 00 FA 33 C0 8E D0 BC 00 7C FB 68 C0 07  ....ú3ÀŽÐ¼.|ûhÀ.
000100060  1F 1E 68 66 00 CB 88 16 0E 00 66 81 3E 03 00 4E  ..hf.Ë^...f.>..N
000100070  54 46 53 75 15 B4 41 BB AA 55 CD 13 72 0C 81 FB  TFSu.´A»ªUÍ.r..û
000100080  55 AA 75 06 F7 C1 01 00 75 03 E9 DD 00 1E 83 EC  Uªu.÷Á..u.éÝ..ƒì
000100090  18 68 1A 00 B4 48 8A 16 0E 00 8B F4 16 1F CD 13  .h..´HŠ...‹ô..Í.
0001000A0  9F 83 C4 18 9E 58 1F 72 E1 3B 06 0B 00 75 DB A3  ŸƒÄ.žX.rá;...uÛ£
0001000B0  0F 00 C1 2E 0F 00 04 1E 5A 33 DB B9 00 20 2B C8  ..Á.....Z3Û¹. +È
0001000C0  66 FF 06 11 00 03 16 0F 00 8E C2 FF 06 16 00 E8  fÿ.......ŽÂÿ...è
0001000D0  4B 00 2B C8 77 EF B8 00 BB CD 1A 66 23 C0 75 2D  K.+Èwï¸.»Í.f#Àu-
0001000E0  66 81 FB 54 43 50 41 75 24 81 F9 02 01 72 1E 16  f.ûTCPAu$.ù..r..
0001000F0  68 07 BB 16 68 70 0E 16 68 09 00 66 53 66 53 66  h.».hp..h..fSfSf
000100100  55 16 16 16 68 B8 01 66 61 0E 07 CD 1A 33 C0 BF  U...h¸.fa..Í.3À¿
000100110  28 10 B9 D8 0F FC F3 AA E9 5F 01 90 90 66 60 1E  (.¹Ø.üóªé_...f`.
000100120  06 66 A1 11 00 66 03 06 1C 00 1E 66 68 00 00 00  .f¡..f.....fh...
000100130  00 66 50 06 53 68 01 00 68 10 00 B4 42 8A 16 0E  .fP.Sh..h..´BŠ..
000100140  00 16 1F 8B F4 CD 13 66 59 5B 5A 66 59 66 59 1F  ...‹ôÍ.fY[ZfYfY.
000100150  0F 82 16 00 66 FF 06 11 00 03 16 0F 00 8E C2 FF  .‚..fÿ.......ŽÂÿ
000100160  0E 16 00 75 BC 07 1F 66 61 C3 A0 F8 01 E8 09 00  ...u¼..faÃ ø.è..
000100170  A0 FB 01 E8 03 00 F4 EB FD B4 01 8B F0 AC 3C 00   û.è..ôëý´.‹ð¬<.
000100180  74 09 B4 0E BB 07 00 CD 10 EB F2 C3 0D 0A 41 20  t.´.»..Í.ëòÃ..A
000100190  64 69 73 6B 20 72 65 61 64 20 65 72 72 6F 72 20  disk read error
0001001A0  6F 63 63 75 72 72 65 64 00 0D 0A 42 4F 4F 54 4D  occurred...BOOTM
0001001B0  47 52 20 69 73 20 6D 69 73 73 69 6E 67 00 0D 0A  GR is missing...
0001001C0  42 4F 4F 54 4D 47 52 20 69 73 20 63 6F 6D 70 72  BOOTMGR is compr
0001001D0  65 73 73 65 64 00 0D 0A 50 72 65 73 73 20 43 74  essed...Press Ct
0001001E0  72 6C 2B 41 6C 74 2B 44 65 6C 20 74 6F 20 72 65  rl+Alt+Del to re
0001001F0  73 74 61 72 74 0D 0A 00 8C A9 BE D6 00 00 55 AA  start...Œ©¾Ö..Uª
000100200  07 00 42 00 4F 00 4F 00 54 00 4D 00 47 00 52 00  ..B.O.O.T.M.G.R.
000100210  04 00 24 00 49 00 33 00 30 00 00 D4 00 00 00 24  ..$.I.3.0..Ô...$
000100220  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000100230  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000100240  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

- Jump Instruction
- OEM ID
- BIOS Parameter Block
- Boot Code
- Error Message
- Message Offset
- Signature



Bios Parameter Block break up



Sectors Per Track | Bytes Per Sector | Sectors Per Cluster | Reserved Sector

```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F  Decoded text
00010000                                   00 02 08 00 00              .....
00010010  00 00 00 00 00 F8 00 00 3F 00 FF 00 80 00 00 00  .....ø..?.ÿ.€...
00010020  00 00 00 00 80 00 80 00 FF 2F 03 00 00 00 00 00  ....€.€.ÿ/......
00010030  00 22 00 00 00 00 00 00 02 00 00 00 00 00 00 00  ."..............
00010040  F6 00 00 00 01 00 00 00 0B 3F 96 B4 68 96 B4 62  ö........?-´h-´b
00010050  00 00 00 00
```

Logical Cluster Number for the file $MFT | Total Sectors | Number Of Heads | Logical Cluster Number for the file $MFTMirr

Volume Serial Number

Finding location for MFT – ntfs8gb.dd

```
000100000  EB 52 90 4E 54 46 53 20 20 20 20 00 02 08 00 00  ëR.NTFS    .....
000100010  00 00 00 00 00 F8 00 00 3F 00 FF 00 00 08 00 00  .....ø..?.ÿ.....
000100020  00 00 00 00 80 00 00 00 FF 43 E7 00 00 00 00 00  ....€...ÿCç.....
000100030  00 00 0C 00 00 00 00 00 02 00 00 00 00 00 00 00  ................
000100040  F6 00 00 00 01 00 00 00 FD 5C 5D AE 94 5D AE 26  ö.......ý\]®"]®&
000100050  00 00 00 00 FA 33 C0 8E D0 BC 00 7C FB 68 C0 07  ....ú3ÀŽÐ¼.|ûhÀ.
000100060  1F 1E 68 66 00 CB 88 16 0E 00 66 81 3E 03 00 4E  ..hf.Ë^...f.>..N
000100070  54 46 53 75 15 B4 41 BB AA 55 CD 13 72 0C 81 FB  TFSu.´A»ªUÍ.r..û
000100080  55 AA 75 06 F7 C1 01 00 75 03 E9 DD 00 1E 83 EC  Uªu.÷Á..u.éÝ..ƒì
```

08 * 512=4096
=0x1000

0x0C0000 * 0x1000(cluster size) + 0x100000

=0xC0100000 location for the begining of MFT

(starts with FILE0 SIGNATURE)

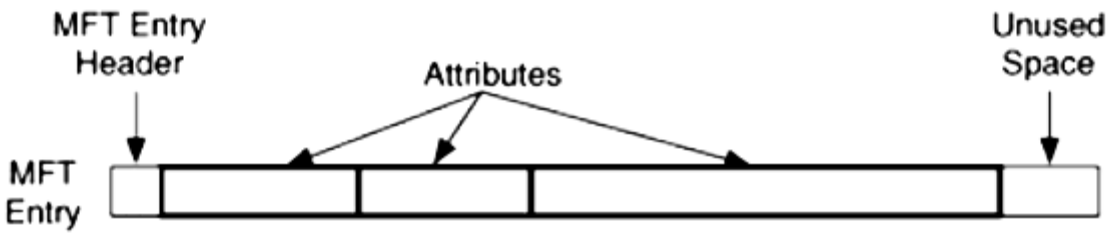
The Master File Table (MFT)

- The Master File Table (MFT) is the primary source of metadata in NTFS.
- It contains or indirectly references everything about a file:
  - its timestamps, size in bytes, attributes (such as permissions), parent directory, and contents.
  - A sizeable area of the NTFS volume is reserved for the MFT to avoid it becoming fragmented as it grows in size.
  - This area, by default, is about 12.5% of the volume size and is known as the “MFT Reserved Area”.
  - As data is added, the MFT can expand to take up 50% of the disk.

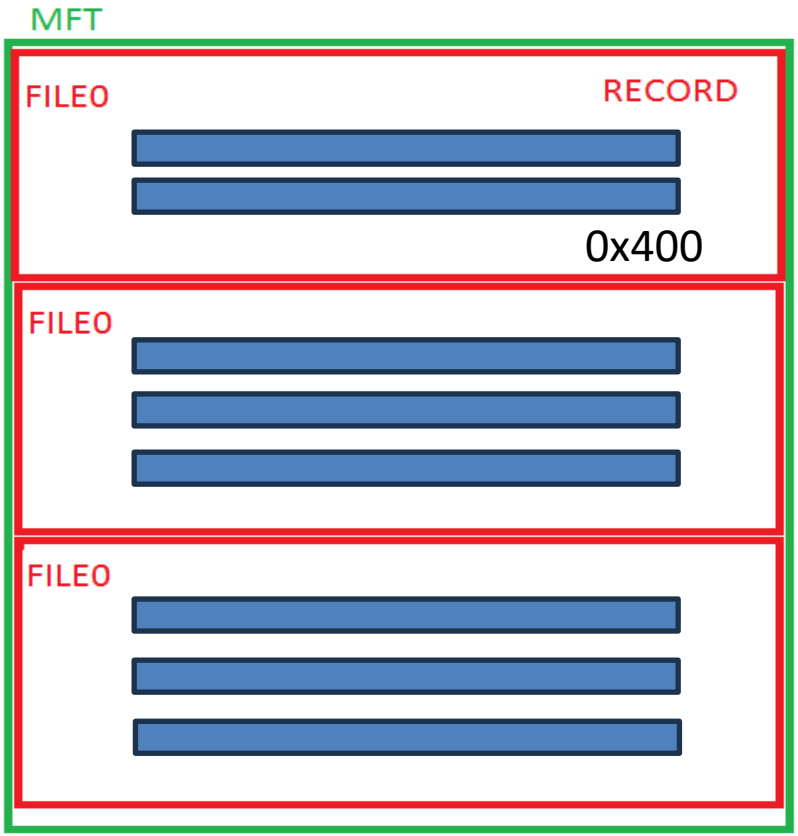
- On a standard hard drive with 512-byte sectors, the MFT is structured as a series of 1,024-byte records, also known as “entries,” one for each file and directory on a volume but only the first 42 bytes (MFT header) have a defined purpose.

- The remaining 982 bytes store attributes, which are small data structures that have a very specific purpose.

- However, on advanced format (AF) drives with 4KB sectors, each MFT record will be 4,096 bytes instead.



Basic layout of MFT entry



MFT CONTAINS RECORDS. EACH RECORD IS OF 1024 BYTES IN LENGTH. WHICH IS FURTHER DIVIDED INTO SUB-SECTIONS CALLED AS ATTRIBUTES

If the MFT Entry for an NTFS volume starts with **FILE0**, this means that the NTFS volume has probably been formatted with Windows XP, or newer.

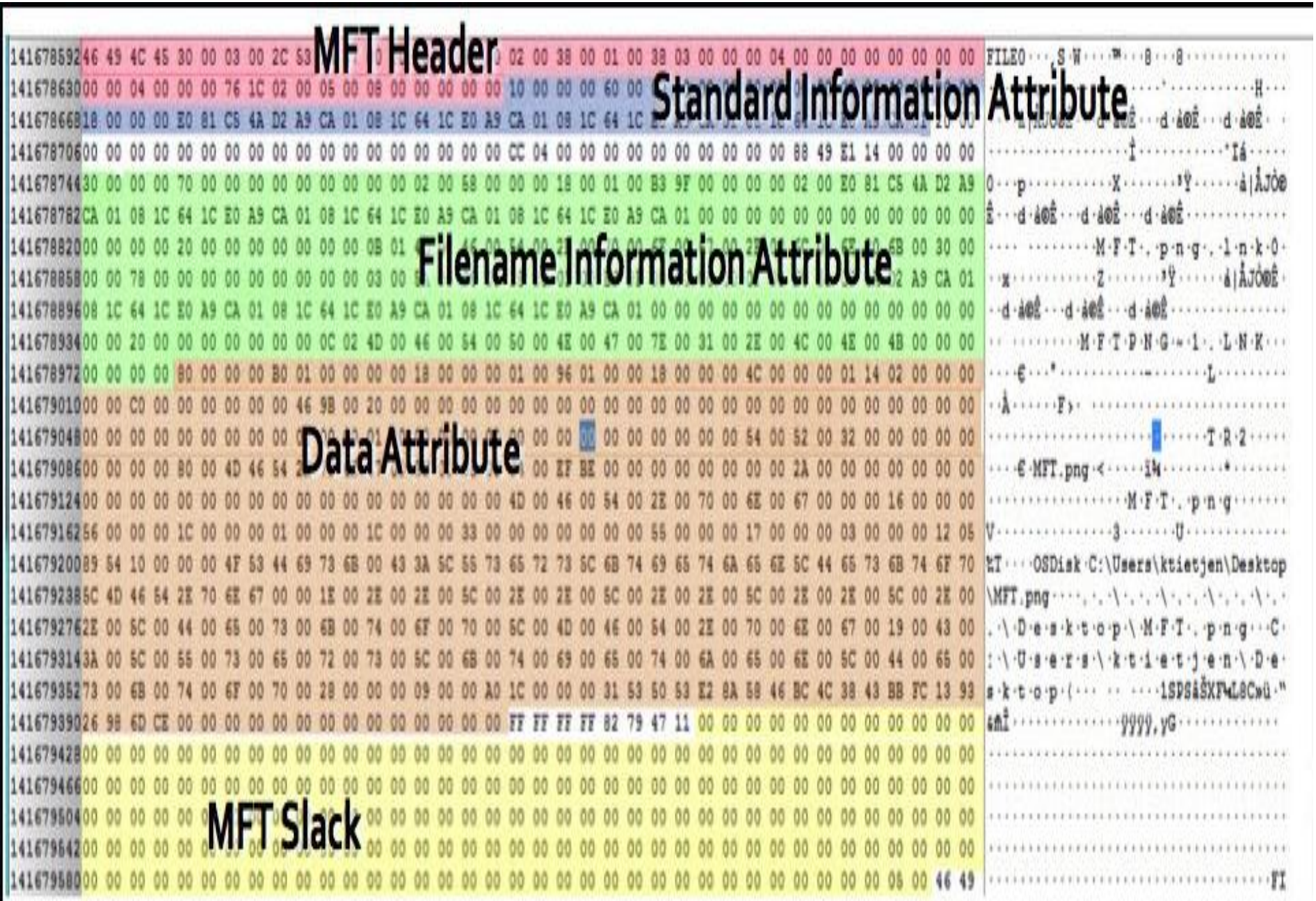
If the MFT Entries start with **FILE***, it means that the volume was probably formatted with Windows 2K or older.

```
0C0100000  46 49 4C 45 30 00 03 00 F4 22 00 01 00 00 00 00  FILE0...ô"......
0C0100010  01 00 01 00 38 00 01 00 A0 01 00 00 00 04 00 00  ....8... .......
0C0100020  00 00 00 00 00 00 00 00 06 00 00 00 00 00 00 00  ................
0C0100030  02 00 00 00 00 00 00 00 10 00 00 00 60 00 00 00  ............`...
0C0100040  00 00 18 00 00 00 00 00 48 00 00 00 18 00 00 00  ........H.......
0C0100050  42 D7 36 56 1F 35 D8 01 42 D7 36 56 1F 35 D8 01  B×6V.5Ø.B×6V.5Ø.
0C0100060  42 D7 36 56 1F 35 D8 01 42 D7 36 56 1F 35 D8 01  B×6V.5Ø.B×6V.5Ø.
0C0100070  06 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0C0100080  00 00 00 00 00 01 00 00 00 00 00 00 00 00 00 00  ................
0C0100090  00 00 00 00 00 00 00 00 30 00 00 00 68 00 00 00  ........0...h...
0C01000A0  00 00 18 00 00 00 03 00 4A 00 00 00 18 00 01 00  ........J.......
0C01000B0  05 00 00 00 00 00 05 00 42 D7 36 56 1F 35 D8 01  ........B×6V.5Ø.
0C01000C0  42 D7 36 56 1F 35 D8 01 42 D7 36 56 1F 35 D8 01  B×6V.5Ø.B×6V.5Ø.
0C01000D0  42 D7 36 56 1F 35 D8 01 00 40 00 00 00 00 00 00  B×6V.5Ø..@......
0C01000E0  00 40 00 00 00 00 00 00 06 00 00 00 00 00 00 00  .@..............
0C01000F0  04 03 24 00 4D 00 46 00 54 00 00 00 00 00 00 00  ..$.M.F.T.......
0C0100100  80 00 00 00 48 00 00 00 01 00 40 00 00 00 01 00  €...H.....@.....
```

System File Records in MFT

Name	MFT Record No.	Purpose
$MFT	0	Contains one record for each file and folder on the volume.
$MFTMirr	1	Duplicate of the first four records of the MFT - in the case of a sector failure.
$LogFile	2	Contains a list of transaction steps for NTFS recoverability.
$Volume	3	Volume information.
$AttrDef	4	Contains a table of attribute names, numbers and descriptions.
$	5	Root folder.
$Bitmap	6	Representation of the volume, showing which clusters are in use (e.g. a map).
$Boot	7	Includes the Boot Partition Block which is used to mount the volume and bootstrap loader code.
$BadClus	8	Contains bad clusters for the volume.
$Secure	9	Security descriptors for all files in the volume.
$Upcase	10	Used for lower to uppercase character conversion.
$Extend	11	Used for optional extensions such as object identifiers and quotas.
	12-15	Reserved for future use.
$Quota	24	User assigned quota limits on the volume space.
$ObjId	25	Contains file object IDs.
$Reparse	26	Information about files/folders reparse points.

MFT USER RECORD SUBSECTIONS

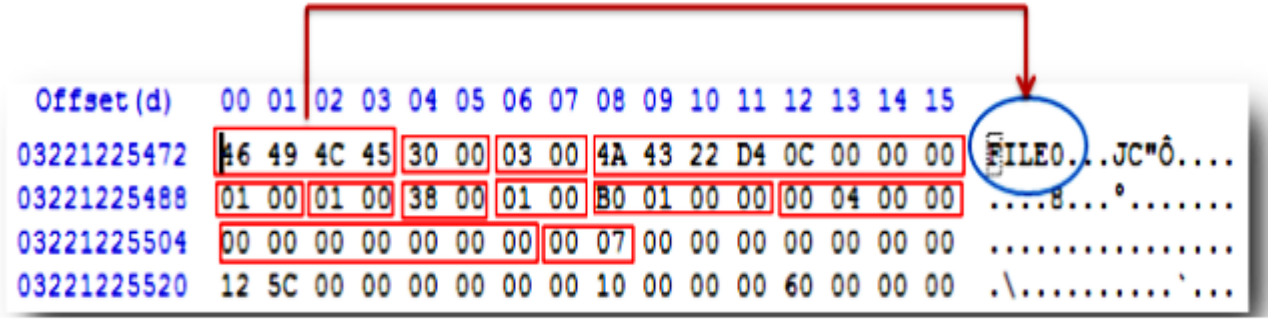


Value	Meaning
$STANDARD_INFORMATION 0x10	File attributes (such as read-only and archive), time stamps (such as file creation and last modified), and the hard link count.
$ATTRIBUTE_LIST 0x20	A list of attributes that make up the file and the file reference of the MFT file record in which each attribute is located.
$FILE_NAME 0x30	The name of the file, in Unicode characters.
$OBJECT_ID 0x40	An 16-byte object identifier assigned by the link-tracking service.
$VOLUME_NAME 0x60	The volume label. Present in the $Volume file.
$VOLUME_INFORMATION 0x70	The volume information. Present in the $Volume file.
$DATA 0x80	The contents of the file.
$INDEX_ROOT 0x90	Used to implement filename allocation for large directories.
$INDEX_ALLOCATION 0xA0	Used to implement filename allocation for large directories.
$BITMAP 0xB0	A bitmap index for a large directory.
$REPARSE_POINT 0xC0	The reparse point data.

Attribute ID	Purpose
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 OS/2 HPFS.
0xE0	$EA For use with OS/2 HPFS.
0x100	$Logged_Utility_Stream This field is used by EFS in Windows 2000, XP, and Vista.

MFT Header

The MFT record starts with a **header** with a size of 42 bytes.

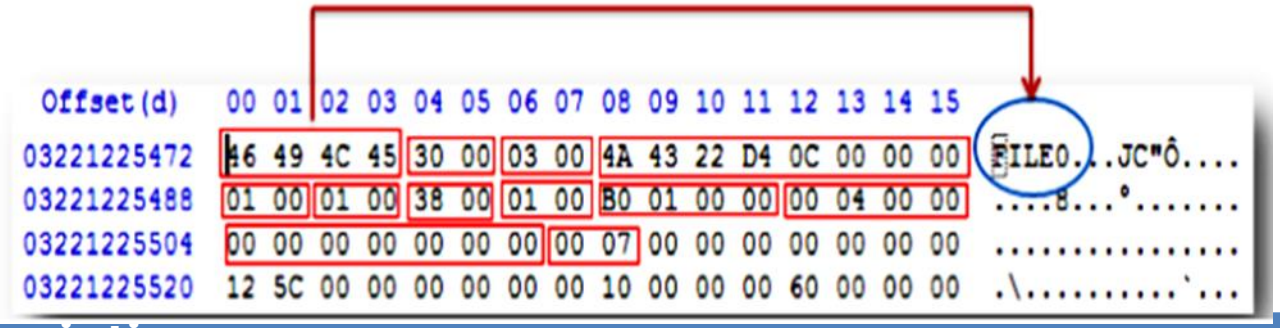


The first 4 bytes (characters) for all MFT records are **FILE**.

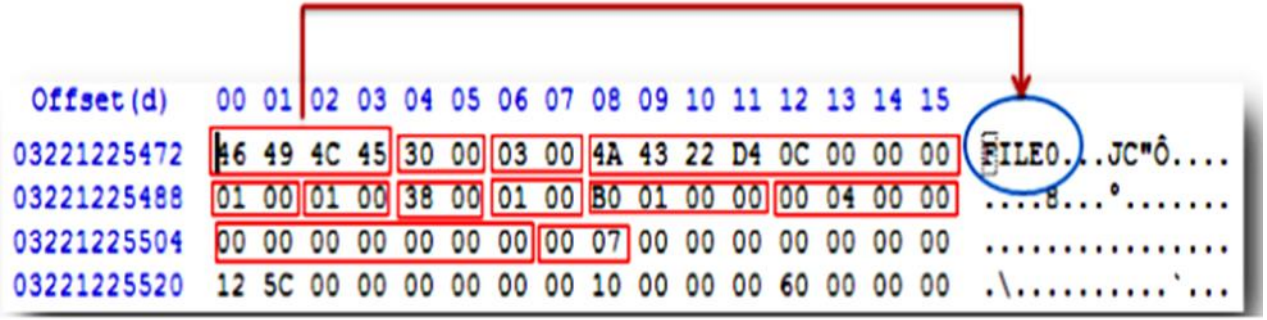
The header information contains additional data specifying **where the first attribute ID starts**, which is typically at offset 0x20 from the beginning of the record.

Each attribute ID has a length value in hexadecimal defining where it ends and where the next attribute starts.

The length value is located 4 bytes from the attribute ID.



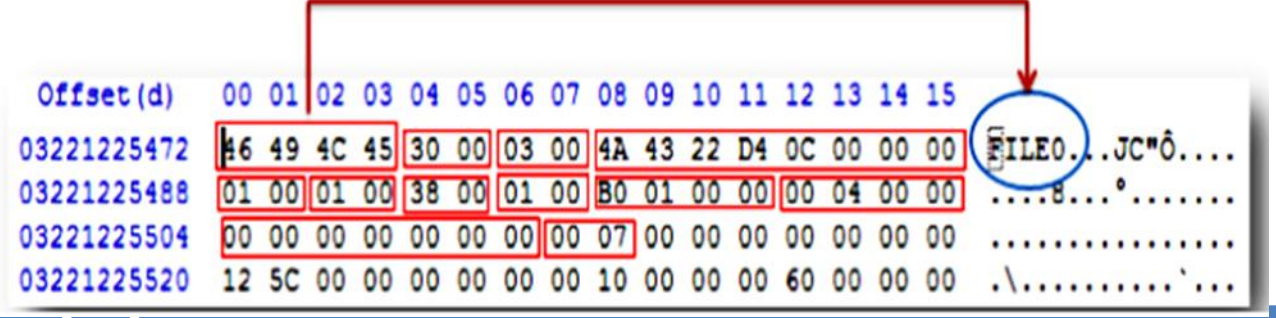
Byte range	Description
0 - 3	Signature (“FILE”). **Size**: 4 bytes
4 - 5	Offset to fixup array - 0x30 00. **Size**: 2 bytes. The fixup array is used to validate sectors within the MFT record. This output is in little-endian ordering, so we need to reverse the order of the numbers. So it becomes 00 30, which is 48 in decimal. This shows that the fixup array is located 48 bytes (0x0030) into the MFT entry.
6 - 7	Number of entries in fixup array - 0x03 00. **Size**: 2 bytes. This output is in little-endian ordering, so we need to reverse the order of the numbers. So it becomes 00 03, which is 3 in decimal. This means that the array has three values in it.
8 - 15	$LogFile sequence number (LSN) - 0x 4A 43 22 D4 0C 00 00 00. **Size**: 8 bytes. Holds the sequence number of the logfile entry that tracks every change to the file. The log records when metadata updates are made to the file system so that a corrupt file system can be more quickly fixed. This output is in little-endian ordering, so we need to reverse the order of the numbers. So it becomes 00 00 00 0C D4 22 43 4A, which is 55098622794 in decimal.



Byte range	Description
16 - 17	Sequence value - 0x 01 00. **Size**: 2 bytes The sequence value is incremented when the entry is either allocated or unallocated, determined by the OS.
18 - 19	Link count - 0x 01 00. **Size**: 2 bytes The link count shows how many directories have entries for this MFT entry. If hard links were created for the file, this number is incremented by one for each link. Microsoft defines hard links as: “NTFS-based links to a file on an NTFS volume. By creating hard links, you can have a single file in multiple folders without duplicating the file. You can also create multiple hard links for a file in a folder if you use different file names for the hard links. Because all of the hard links reference the same file, applications can open any of the hard links and modify the file.” In little endian becomes 00 01 which is 1 in decimal. This brings us to a conclusion that only one directory has entry for this MFT record/entry.

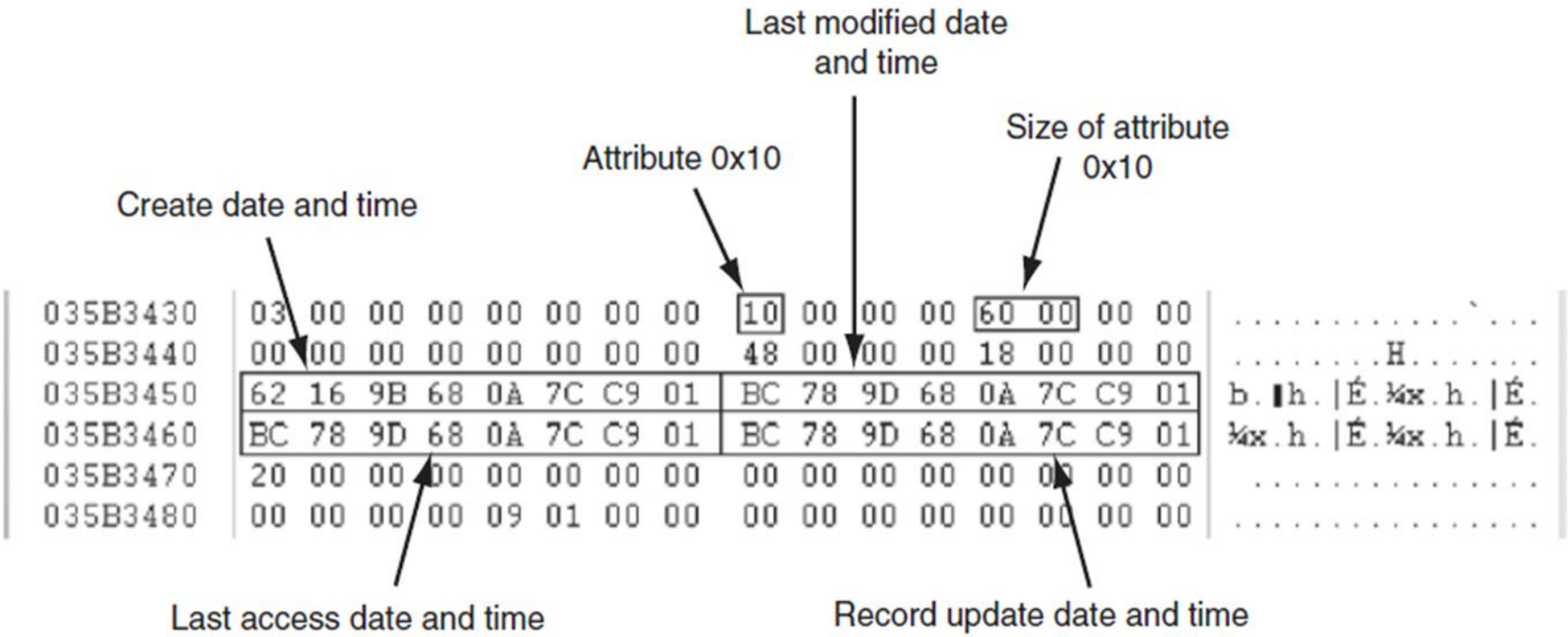
```
Offset(d)     00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15
03221225472   46 49 4C 45 30 00 03 00 4A 43 22 D4 0C 00 00 00   FILE0...JC"Ô....
03221225488   01 00 01 00 38 00 01 00 B0 01 00 00 00 04 00 00   ....8...°.......
03221225504   00 00 00 00 00 00 00 00 07 00 00 00 00 00 00 00   ................
03221225520   12 5C 00 00 00 00 00 00 10 00 00 00 60 00 00 00   .\..........`...
```

Byte range	Description
20 - 21	Offset to first attribute - 0x38 00. **Size**: 2 bytes This is the first attribute for the file. All other attributes follow the first one, and we find them by advancing ahead using the size field in the attribute header. The end of file marker 0xffffffff exists after the last attribute. If a file needs more than one MFT entry, the additional ones will have the file reference of the base entry in their MFT entry. This output is in little-endian ordering, so we need to reverse the order of the numbers. So it becomes 00 38, which is 56 in decimal. This indicates that the first Attribute starts at byte offset 56.
22 -23	Flag (in-use and directory): 0x0000: Deleted file; 0x0001: Allocated file; 0x0002: Deleted directory; 0x0003: Allocated directory. **Size**: 2 bytes In this case we note that the value is 0x01 00 and that it is a FILE record in use
24 - 27	Used size of MFT entry - 0xB0 01 00 00. **Size**: 4 bytes Indicates the "real" length of the file record. If this MFT record is the base entry for the file then this field is zero: if the record is an extension then this field holds the base record reference address. Here it is referred to here as the "logical size". This "logical" size is the actual number of bytes of data stored in the record. Reversing this value, it becomes 00 00 01 B0; which equates to 432 in decimal. Therefore it can be concluded that the entry size is 432 bytes.



Byte range	Description
28 - 31	Allocated size of MFT entry - 00 04 00 00. **Size**: 4 bytes Indicates the allocated storage size of the file record. This is referred to as the "physical" size and this size has already been preset to 1024 bytes by the BPB. In this case translation from the little endian format gives 0x00 00 04 00, which does indeed equate to 1024 bytes in decimal.
32 - 39	File reference to base record. **Size**: 8 bytes It is used when the record to be stored exceeds the allocated space of one or more MFT records.
40 - 41	Next attribute ID. **Size**: 2 bytes
42 - 43	Alignment to 4-byte boundary
44 - 47	MFT file record number (only in NTFS 3.1 and later)
42 - 1023	Attribute and Fixup value

Attribute 0x10: Standard Information Following the MFT header for a data file is the Standard Information attribute, 0x10, which has the following fields



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

- For very small files, about 512 bytes or less, all file metadata and data are stored in the MFT record. These types of records are called resident files because all their information is stored in the MFT record.

- Files larger than 512 bytes are stored outside the MFT. The file or folder's MFT record provides cluster addresses where the file is stored on the drive's partition.

- These cluster addresses are referred to as data runs. This type of MFT record is called nonresident because the file's data is stored outside the MFT.

- Each MFT record starts with a header identifying it as a **resident or nonresident attribute.**

Example

Drive C: | $MFT after | $MFT before

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
00321400	46	49	4C	45	30	00	03	00	44	6C	18	4C	04	00	00	00	FILE0 Dl L
00321410	39	02	01	00	38	00	01	00	C8	01	00	00	00	04	00	00	9 8 È
00321420	00	00	00	00	00	00	00	00	07	00	00	00	85	0C	00	00	▌
00321430	0B	00	00	00	00	00	00	00	10	00	00	00	60	00	00	00	`
00321440	00	00	00	00	00	00	00	00	48	00	00	00	18	00	00	00	H
00321450	DE	3F	D5	5B	BF	A1	CB	01	DC	F0	7B	BC	6D	A1	CB	01	Þ?Õ[¿¡Ë Üð{¼m¡Ë
00321460	06	7C	01	55	81	A6	CB	01	AA	01	B1	4E	81	A6	CB	01	| U▌¦Ë ª ±N▌¦Ë
00321470	20	20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00321480	00	00	00	00	5D	01	00	00	00	00	00	00	00	00	00	00	]
00321490	08	FC	52	58	00	00	00	00	30	00	00	00	78	00	00	00	üRX 0 x
003214A0	00	00	00	00	00	00	06	00	5A	00	00	00	18	00	01	00	Z
003214B0	E7	25	00	00	00	00	01	00	DE	3F	D5	5B	BF	A1	CB	01	ç% Þ?Õ[¿¡Ë
003214C0	DC	F0	7B	BC	6D	A1	CB	01	DC	F0	7B	BC	6D	A1	CB	01	Üð{¼m¡Ë Üð{¼m¡Ë
003214D0	78	EE	4E	41	81	A6	CB	01	00	40	00	00	00	00	00	00	xîNA▌¦Ë @
003214E0	9C	34	00	00	00	00	00	00	20	20	00	00	00	00	00	00	▌4
003214F0	0C	03	61	00	61	00	72	00	64	00	76	00	61	00	72	00	a a r d v a r
00321500	6B	00	2E	00	74	00	78	00	74	00	00	00	00	00	00	00	k . t x t
00321510	80	00	00	00	48	00	00	00	01	00	00	00	00	00	04	00	▌ H
00321520	00	00	00	00	00	00	00	00	03	00	00	00	00	00	00	00	
00321530	40	00	00	00	00	00	00	00	00	40	00	00	00	00	00	00	@ @
00321540	9C	34	00	00	00	00	00	00	9C	34	00	00	00	00	00	00	▌4 ▌4
00321550	41	04	B4	7D	B9	00	00	00	80	00	00	00	68	00	00	00	A ´}¹ ▌ h
00321560	01	0F	40	00	00	00	05	00	00	00	00	00	00	00	00	00	@
00321570	00	00	00	00	00	00	00	00	60	00	00	00	00	00	00	00	`
00321580	00	10	00	00	00	00	00	00	2E	00	00	00	00	00	00	00	.
00321590	2E	00	00	00	00	00	00	00	5A	00	6F	00	6E	00	65	00	. Z o n e
003215A0	2E	00	49	00	64	00	65	00	6E	00	74	00	69	00	66	00	. I d e n t i f
003215B0	69	00	65	00	72	00	00	00	41	01	B8	7D	B9	00	00	00	i e r A ¸}¹
003215C0	FF	FF	FF	FF	82	79	47	11	9C	34	00	00	00	00	00	00	ÿÿÿÿ‚yG ▌4

the chain of attributes from the start of the MFT header.

- the offset to the first attribute in the record is 0x38. At 0x38 is 0x10, $Standard_Information
- at offset 0x04 in the attribute is the length of this attribute, 0x60, which leads to
- attribute 0x30, $File_Name, length 0x78, which leads to
- attribute 0x80, $Data, length 0x48, which leads to
- attribute 0x80, $Data, length 0x68, which leads to
- attribute 0xFFFFFFFF, the end of attributes attribute

Non-Resident Files

When the information for a file is too large to fit in its MFT file record, some of the file attributes are non-resident.

Non-resident attributes are allocated one or more clusters of disk space and **stored as an alternate data stream in the volume.**

NTFS creates the $Attribute_List attribute to describe the location of both resident and non-resident attribute records.

Non Resident is denoted by **01 in the data attribute**

– ntfs8gb.dd

ntfs8gb.dd

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
0C0109000	46	49	4C	45	30	00	03	00	F8	B0	00	01	00	00	00	00	FILE0...ø°.....
0C0109010	02	00	02	00	38	00	00	00	18	02	00	00	00	04	00	00	....8.....
0C0109020	00	00	00	00	00	00	00	00	04	00	00	00	24	00	00	00	.....$...
0C0109030	07	00	00	00	00	00	00	00	10	00	00	00	60	00	00	00	.....`...
0C0109040	00	00	00	00	00	00	00	00	48	00	00	00	18	00	00	00	.....H.....
0C0109050	1D	0F	E6	67	20	35	D8	01	41	31	48	65	C9	B9	D7	01	..æg 5Ø.A1HeÉ¹×.
0C0109060	A6	22	E6	67	20	35	D8	01	1D	0F	E6	67	20	35	D8	01	¦"æg 5Ø...æg 5Ø.
0C0109070	20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
0C0109080	00	00	00	00	05	01	00	00	00	00	00	00	00	00	00	00	.....
0C0109090	00	00	00	00	00	00	00	00	30	00	00	00	78	00	00	00	.....0...x...
0C01090A0	00	00	00	00	00	00	03	00	5A	00	00	00	18	00	01	00	.....Z.....
0C01090B0	05	00	00	00	00	00	05	00	1D	0F	E6	67	20	35	D8	01	.....æg 5Ø.
0C01090C0	1D	0F	E6	67	20	35	D8	01	1D	0F	E6	67	20	35	D8	01	..æg 5Ø...æg 5Ø.
0C01090D0	1D	0F	E6	67	20	35	D8	01	00	40	00	00	00	00	00	00	..æg 5Ø..@.....
0C01090E0	00	00	00	00	00	00	00	00	20	00	00	00	00	00	00	00	....
0C01090F0	0C	02	42	00	54	00	5F	00	4C	00	45	00	43	00	7E	00	..B.T._.L.E.C.~.
0C0109100	31	00	2E	00	58	00	4C	00	53	00	5F	00	20	00	30	00	1...X.L.S._. .0.
0C0109110	30	00	00	00	B8	00	00	00	00	00	00	00	00	00	02	00	0...¸.....
0C0109120	A0	00	00	00	18	00	01	00	05	00	00	00	00	00	05	00	.....
0C0109130	1D	0F	E6	67	20	35	D8	01	1D	0F	E6	67	20	35	D8	01	..æg 5Ø...æg 5Ø.
0C0109140	1D	0F	E6	67	20	35	D8	01	1D	0F	E6	67	20	35	D8	01	..æg 5Ø...æg 5Ø.
0C0109150	00	40	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.@.....
0C0109160	20	00	00	00	00	00	00	00	2F	01	42	00	54	00	5F	00	...../.B.T._.
0C0109170	4C	00	45	00	43	00	54	00	55	00	52	00	45	00	20	00	L.E.C.T.U.R.E. .
0C0109180	32	00	5F	00	20	00	30	00	35	00	5F	00	31	00	30	00	2._. .0.5._.1.0.
0C0109190	5F	00	32	00	30	00	32	00	31	00	20	00	41	00	74	00	_.2.0.2.1. .A.t.
0C01091A0	74	00	65	00	6E	00	64	00	61	00	6E	00	63	00	65	00	t.e.n.d.a.n.c.e.
0C01091B0	20	00	43	00	53	00	45	00	44	00	32	00	33	00	2E	00	.C.S.E.D.2.3...
0C01091C0	78	00	6C	00	73	00	78	00	80	00	00	00	48	00	00	00	x.l.s.x.€...H...
0C01091D0	01	00	00	00	00	00	01	00	00	00	00	00	00	00	00	00	.....
0C01091E0	03	00	00	00	00	00	00	00	40	00	00	00	00	00	00	00	.....@.....
0C01091F0	00	40	00	00	00	00	00	00	08	31	00	00	00	00	07	00	.@.....1.....
0C0109200	08	31	00	00	00	00	00	00	11	04	24	00	00	00	00	00	.1.....$.....
0C0109210	FF	FF	FF	FF	82	79	47	11	00	00	00	00	00	00	00	00	ÿÿÿÿ,yG.....
0C0109220	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....

Deleted files

- When a file is deleted in NTFS, it is marked as deleted within the MFT entry for that file. The clusters that were allocated to the fille are now marked as free, within the $BitMap

- This is shown at offset 22 for 2 bytes; i.e. bytes 22 and 23 of the MFT for that entry.

- For an active file the 22nd and 23rd offsets read “01 00”

- For a deleted file the 22nd and 23rd offsets read “00 00”.

- Active File

ntfs8gb.dd

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
0C0108BF0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	09	00	.....
0C0108C00	46	49	4C	45	30	00	03	00	A8	58	00	01	00	00	00	00	FILE0...¨X.....
0C0108C10	01	00	02	00	38	00	01	00	D8	01	00	00	00	04	00	00	....8..Ø.....
0C0108C20	00	00	00	00	00	00	00	00	05	00	00	00	23	00	00	00	.....#...
0C0108C30	09	00	00	00	00	00	00	00	10	00	00	00	60	00	00	00	.....`...
0C0108C40	00	00	00	00	00	00	00	00	48	00	00	00	18	00	00	00	.....H.....
0C0108C50	E6	4F	E3	67	20	35	D8	01	66	2D	46	AE	CF	8B	CF	01	æOãg 5Ø.f-F®Ï‹Ï.
0C0108C60	92	DD	40	B2	27	35	D8	01	89	A8	EE	D5	21	35	D8	01	'Ý@²'5Ø.‰¨îÕ!5Ø.
0C0108C70	20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
0C0108C80	00	00	00	00	05	01	00	00	00	00	00	00	00	00	00	00	.....
0C0108C90	80	1E	00	00	00	00	00	00	30	00	00	00	78	00	00	00	€.....0...x...
0C0108CA0	00	00	00	00	00	00	04	00	5A	00	00	00	18	00	01	00	.....Z.....
0C0108CB0	05	00	00	00	00	00	05	00	E6	4F	E3	67	20	35	D8	01	.....æOãg 5Ø.
0C0108CC0	66	2D	46	AE	CF	8B	CF	01	F8	76	E3	67	20	35	D8	01	f-F®Ï‹Ï.øvãg 5Ø.
0C0108CD0	89	A8	EE	D5	21	35	D8	01	00	60	02	00	00	00	00	00	‰¨îÕ!5Ø..`.....
0C0108CE0	77	50	02	00	00	00	00	00	20	00	00	00	00	00	00	00	wP..... .....
0C0108CF0	0C	02	42	00	4F	00	59	00	5F	00	53	00	4F	00	7E	00	..B.O.Y._.S.O.~.
0C0108D00	31	00	2E	00	4D	00	50	00	33	00	33	00	00	00	01	00	1...M.P.3.3.....
0C0108D10	30	00	00	00	78	00	00	00	00	00	00	00	00	00	03	00	0...x.....
0C0108D20	5C	00	00	00	18	00	01	00	05	00	00	00	00	00	05	00	\.....
0C0108D30	E6	4F	E3	67	20	35	D8	01	66	2D	46	AE	CF	8B	CF	01	æOãg 5Ø.f-F®Ï‹Ï.
0C0108D40	F8	76	E3	67	20	35	D8	01	89	A8	EE	D5	21	35	D8	01	øvãg 5Ø.‰¨îÕ!5Ø.
0C0108D50	00	60	02	00	00	00	00	00	77	50	02	00	00	00	00	00	.`.....wP.....
0C0108D60	20	00	00	00	00	00	00	00	0D	01	62	00	6F	00	79	00	.....b.o.y.
0C0108D70	5F	00	73	00	6F	00	75	00	6E	00	64	00	2E	00	6D	00	_.s.o.u.n.d...m.
0C0108D80	70	00	33	00	00	00	01	00	80	00	00	00	48	00	00	00	p.3.....€...H...
0C0108D90	01	00	00	00	00	00	01	00	00	00	00	00	00	00	00	00	.....
0C0108DA0	25	00	00	00	00	00	00	00	40	00	00	00	00	00	00	00	%.....@.....
0C0108DB0	00	60	02	00	00	00	00	00	77	50	02	00	00	00	00	00	.`.....wP.....
0C0108DC0	77	50	02	00	00	00	00	00	31	26	13	EE	0B	00	00	00	wP.....1&.î....
0C0108DD0	FF	FF	FF	FF	82	79	47	11	00	00	00	00	00	00	00	00	ÿÿÿÿ,yG.....
0C0108DE0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
0C0108DF0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	09	00	.....
0C0108E00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
0C0108E10	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....

- Deleted File

ntfs8gb.dd

```
Offset(h)  00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F  Decoded text

0C0109000  46 49 4C 45 30 00 03 00 F8 B0 00 01 00 00 00 00  FILE0...ø°......
0C0109010  02 00 02 00 38 00 00 00 18 02 00 00 00 04 00 00  ....8...........
0C0109020  00 00 00 00 00 00 00 00 04 00 00 00 24 00 00 00  ............$...
0C0109030  07 00 00 00 00 00 00 00 10 00 00 00 60 00 00 00  ............`...
0C0109040  00 00 00 00 00 00 00 00 48 00 00 00 18 00 00 00  ........H.......
0C0109050  1D 0F E6 67 20 35 D8 01 41 31 48 65 C9 B9 D7 01  ..æg 5Ø.A1HeÉ¹×.
0C0109060  A6 22 E6 67 20 35 D8 01 1D 0F E6 67 20 35 D8 01  ¦"æg 5Ø...æg 5Ø.
0C0109070  20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ...............
0C0109080  00 00 00 00 05 01 00 00 00 00 00 00 00 00 00 00  ................
0C0109090  00 00 00 00 00 00 00 00 30 00 00 00 78 00 00 00  ........0...x...
0C01090A0  00 00 00 00 00 00 03 00 5A 00 00 00 18 00 01 00  ........Z.......
0C01090B0  05 00 00 00 00 00 05 00 1D 0F E6 67 20 35 D8 01  ..........æg 5Ø.
0C01090C0  1D 0F E6 67 20 35 D8 01 1D 0F E6 67 20 35 D8 01  ..æg 5Ø...æg 5Ø.
0C01090D0  1D 0F E6 67 20 35 D8 01 00 40 00 00 00 00 00 00  ..æg 5Ø..@......
0C01090E0  00 00 00 00 00 00 00 00 20 00 00 00 00 00 00 00  ........ .......
0C01090F0  0C 02 42 00 54 00 5F 00 4C 00 45 00 43 00 7E 00  ..B.T._.L.E.C.~.
0C0109100  31 00 2E 00 58 00 4C 00 53 00 5F 00 20 00 30 00  1...X.L.S._. .0.
0C0109110  30 00 00 00 B8 00 00 00 00 00 00 00 00 00 02 00  0...¸...........
0C0109120  A0 00 00 00 18 00 01 00 05 00 00 00 00 00 05 00   ...............
0C0109130  1D 0F E6 67 20 35 D8 01 1D 0F E6 67 20 35 D8 01  ..æg 5Ø...æg 5Ø.
0C0109140  1D 0F E6 67 20 35 D8 01 1D 0F E6 67 20 35 D8 01  ..æg 5Ø...æg 5Ø.
0C0109150  00 40 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .@..............
0C0109160  20 00 00 00 00 00 00 00 2F 01 42 00 54 00 5F 00   ...../.B.T._.
0C0109170  4C 00 45 00 43 00 54 00 55 00 52 00 45 00 20 00  L.E.C.T.U.R.E. .
0C0109180  32 00 5F 00 20 00 30 00 35 00 5F 00 31 00 30 00  2._. .0.5._.1.0.
0C0109190  5F 00 32 00 30 00 32 00 31 00 20 00 41 00 74 00  _.2.0.2.1. .A.t.
0C01091A0  74 00 65 00 6E 00 64 00 61 00 6E 00 63 00 65 00  t.e.n.d.a.n.c.e.
0C01091B0  20 00 43 00 53 00 45 00 44 00 32 00 33 00 2E 00   .C.S.E.D.2.3...
0C01091C0  78 00 6C 00 73 00 78 00 80 00 00 00 48 00 00 00  x.l.s.x.€...H...
0C01091D0  01 00 00 00 00 00 01 00 00 00 00 00 00 00 00 00  ................
0C01091E0  03 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00  ........@.......
0C01091F0  00 40 00 00 00 00 00 00 08 31 00 00 00 00 07 00  .@.......1......
0C0109200  08 31 00 00 00 00 00 00 11 04 24 00 00 00 00 00  .1........$.....
0C0109210  FF FF FF FF 82 79 47 11 00 00 00 00 00 00 00 00  ÿÿÿÿ,yG.........
0C0109220  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```