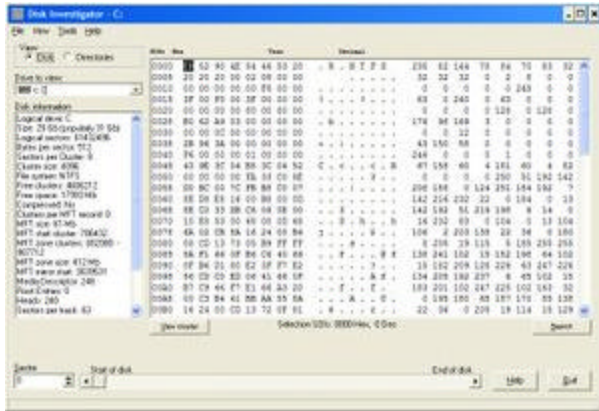


## 8 bits

Techblog over IT, informatiebeveiliging en open source forensics

14-6-07

### NTFS op bitniveau



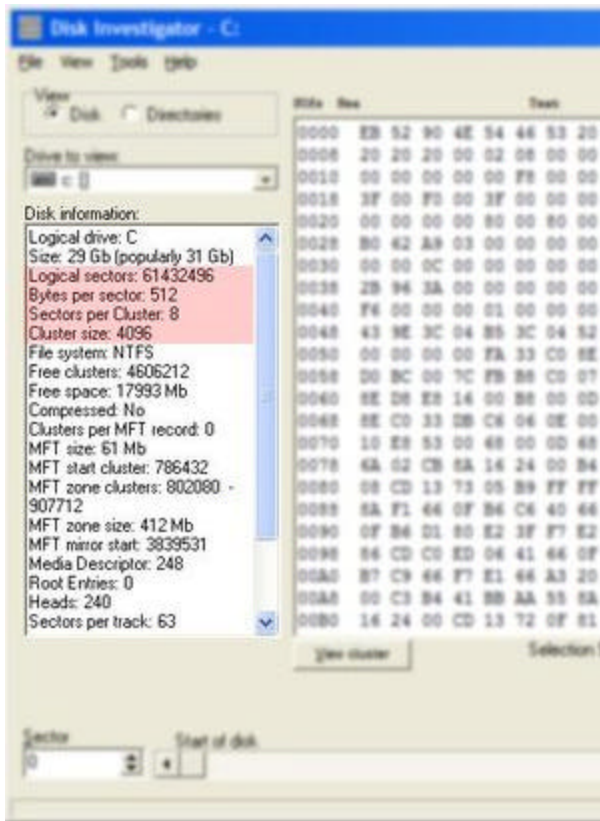
Windows computers (zowel XP als Vista) maken gebruik van NTFS (New Technology File System). Om een beetje meer inzicht in het NTFS bestandssysteem te krijgen kun je gebruik maken van het gratis programma [Disk Investigator](#) of [FTK Imager](#).

Het hoofddoel van elk bestandssysteem (dus ook van NTFS) is het kunnen onderbrengen en terugvinden van bestanden. Als je NTFS van dichtbij (op bitniveau) bekijkt dan kom je een aantal specifieke eigenaardigheden tegen.

#### Sectoren en clusters

NTFS zet bestanden weg in sectoren. Het aantal sectoren dat op een harddisk aanwezig is hangt af van de geometrie van de disk en wordt daarom door de fabrikant bepaald. Elke sector heeft een vaste lengte van 512 bytes. De sectoren worden ondergebracht in clusters om het systeem meer transparant te maken, zodat er minder processing overhead plaats vindt.

Hoeveel sectoren en clusters een harde schijf precies heeft, nadat het is geformatteerd volgens NTFS, kun je via Disk Investigator terugvinden in de linkerbalk.



Zo kan een schijf zo zijn opgebouwd dat er 8 sectoren per cluster aanwezig zijn. Aangezien elke sector 512 bytes lang is, is de rekensom hierbij:

$$8 \times 512 \text{ bytes} = 4096 \text{ bytes per cluster}$$

in bovenstaand voorbeeld zie je dit ook terug in de waarde achter Cluster size.

### Master Boot Record (MBR)

De eerste fysieke sector van een harddisk, ook wel sector 0 genoemd, is de plek waar de Master Boot Record te vinden is. De BIOS van de computer zoekt hier naar programmacodes waarmee de computer kan booten. Uiteraard is de MBR (die uit 1 sector bestaat) 512 bytes lang. De MBR is opgebouwd uit programmacode, foutmeldingen en een partitietabel (master partition table). Er is maar één MBR aanwezig, in tegenstelling tot Volume Boot Records (VBR), waarvan er meer aanwezig kunnen zijn.

Sinds MS-DOS 2.00 in het verre verleden bevatten de eerste drie bytes van sector 0 de zogenoemde Jump Instruction. Deze eerste drie bytes zijn: EB 52 90. De volgende 8 bytes vormen de "OEM ID" of systeemnaam ("NTFS" en vier lege ruimtes), gevolgd door de BPB (BIOS Parameter Block).

De laatste 125 bytes van de 1e sector (Boot Record) bevatten foutmeldingen, Message Offset bytes en de Word-sized signature ID (ookwel Magic number genoemd): AA 55.

Elke foutmelding begint met de hexadecimale bytes OD en OA, die respectievelijk een Carriage Return en Line Feed voorstellen, en eindigt met een 00 byte.

De zes fysieke sectoren die hierna volgen bevatten de code die communiceren met het NTLDR bestand, waardoor een OS partitie met Windows kan worden opgestart. Deze code wordt de Bootstrap code genoemd.

Als je kijkt met Disk Investigator naar sector 0 van je eigen computer dan zie je hoe het een en ander er precies uit ziet.

Windows 2000 / XP / Vista maken bovendien een "backup" van elke NTFS volume boot record, welke wordt opgeslagen in de laatste sector van de betreffende partitie. Vanwege dit principe is het werkelijke aantal sectoren altijd 1 sector meer dan het overzicht van het "totaal aantal sectoren" dat in de Boot record wordt opgegeven.

[meer informatie >>](#)

### \$Boot sectoren

De MBR (1 sector) en de "Bootstrap Code" bestaan in totaal uit zeven sectoren. Uit de code blijkt dat alle 16 sectoren van de NTFS Boot Record "area" in het geheugen worden geladen. Binnen NTFS zijn de eerste 16 sectoren bekend als \$Boot (clusters 0 en 1). De tweede sector begint altijd met de hexadecimale bytes:

```
05 00 4E 00 54 00 4C 00 44 00 52 00 04 00 24 00
```

```
N T L D R $
```

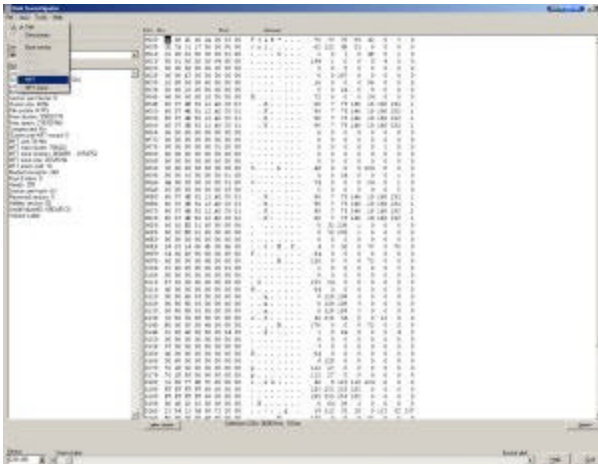
hetgeen unicode is voor de 5 karakters: NTLDR.

De derde tot en met de zesde sector bevatten niet veel schokkends. De zevende sector eindigt met 300 "00" bytes.

\$Boot wordt gevolgd door de SMFT (Master File Table).

### Master File Table

De Master File Table (\$MFT) is via Disk Investigator direct te benaderen door in de bovenbalk te klikken op View > MFT



Disk Investigator zorgt er voor dat de logische sector, waar de MFT begint, in beeld verschijnt. Deze sector kan een nogal hoog nummer hebben, bijvoorbeeld 6291456

De MFT is een van de meest belangrijke onderdelen van het NTFS systeem. In feite is het een relationele database waarin allerlei attributen zijn opgenomen met betrekking tot de bestanden die aanwezig zijn in de volume. De MFT fungeert als startpunt en als 'centrale manager' van de NTFS volume, als ware het een soort 'inhoudsopgave'.

In een FAT volume houden 32 byte entries in directories de naam van een object bij, of het een bestand of een folder is, waar het bestand begint en hoe groot het bestand is, alsmede de bijbehorende datum en tijd. In NTFS houdt de Master File Table (\$MFT) alle gegevens van de bestanden bij. \$MFT weet waar alle bestanden en mappen staan in een volume, hoe de namen zijn etc. Elk object dat ergens in het volume aanwezig is heeft een vermelding in de \$MFT.

De informatie in de \$MFT (zone) kan met Disk Investigator worden bekeken door de bij de \$MFT behorende sectoren door te bladeren via het veldje aan de linkeronderzijde.

Indien een bestand of map wordt aangemaakt op het systeem dan wordt een record aangemaakt in de MFT. Elke record is min of meer gelijk aan de clusteromvang van het systeem, met een minimum van 1024 bytes en een maximum van 4096 bytes (zelfs als een clustergrootte van 512 bytes wordt aangehouden is de omvang van elke MFT record minstens 1024 bytes groot en als de clustergrootte meer is dan 4096 bytes is de maximum omvang van de MFT record 4096 bytes). Elke MFT record neemt dus minimaal twee sectoren in beslag (elke sector is over het algemeen 512 bytes groot).

Het NTFS systeem gebruikt de MFT records om informatie op te slaan over het bestand of de map. Elke record begint met een header waarin de details staan met betrekking tot het bestand en of het bestand gealloceerd danwel gedeleted is. De informatie bevat bovendien allerlei attributen. Omdat de omvang van elk MFT record gelimiteerd is zijn er verschillende manieren om de bestandsattributen op te slaan: als resident attributen die zijn opgeslagen in de MFT record, of als non-resident attributen welke worden opgeslagen in aanvullende MFT records of in ruimte die buiten de MFT (elders binnen het volume) ligt.

Onder NTFS is er geen speciaal onderscheid tussen data in een bestand en de attributen die het bestand beschrijven. De data zelf is gewoon de inhoud van de "data attribute". De inhoud van kleine bestanden (die kleiner zijn dan de omvang van de MFT record) kan resident in de MFT worden opgeslagen. Dergelijke kleine bestanden vereisen geen aanvullende opslagruimte binnen de volume, hetgeen de performance van het systeem ten goede komt. De inhoud van kleine bestanden en diens attributen (in omvang kleiner dan 1024 bytes) kan dus compleet worden aangetroffen in de MFT.

Grotere bestanden vragen een iets gecompliceerder aanpak. Van grote bestanden wordt de verwijzing opgeslagen in de MFT, terwijl de attributen en de inhoud van de bestanden elders binnen het volume (non-resident) wordt opgeslagen. Als een klein bestand, dat eerst in de MFT paste, groter wordt dan 1024 bytes dan wordt het bestand met diens attributen geplaatst buiten de MFT. Als het bestand later weer kleiner wordt, dan keert het echter niet meer terug in de MFT.

Als meer en meer bestanden en mappen worden toegevoegd aan het systeem dan moet NTFS meer records toevoegen aan de MFT. Hier houdt NTFS rekening mee door tijdens de installatie ongeveer 12,5% ruimte te reserveren direct achter de MFT. Dit wordt de MFT zone genoemd. Deze MFT zone kun je met Disk Investigator inzien door te klikken op View > MFT zone.

Gewone bestanden en mappen zullen deze gereserveerde ruimte niet eerder in gebruik nemen totdat de rest van het volume vol is. Als de MFT zone vol is zal NTFS meer ruimte zoeken op het volume om de MFT in onder te brengen. Hierdoor kan de MFT behoorlijk uitgroeien. Helaas kan de MFT niet worden gedefragmenteerd, dus na veelvuldig gebruik en in het geval van veel bestanden en mappen kan het systeem behoorlijk langzaam worden.

De eerste 16 records in de MFT worden gereserveerd voor de Metadata bestanden (systeembestanden) van het volume, zoals \$MFTMirr (een backup van de eerste vier \$MFT records) en \$LogFile (het NTFS journaal dat vele megabytes groot kan zijn).

#### \$Bitmap en clusters

Samen met \$MFT is \$Bitmap een van de meest kenmerkende onderdelen van NTFS. In het FAT bestandssysteem houdt de File Allocation Table bij welke clusters in gebruik zijn en welke beschikbaar zijn voor allocatie (om te worden gevuld met data). In het NTFS systeem is dat anders. Hier houdt \$Bitmap bij

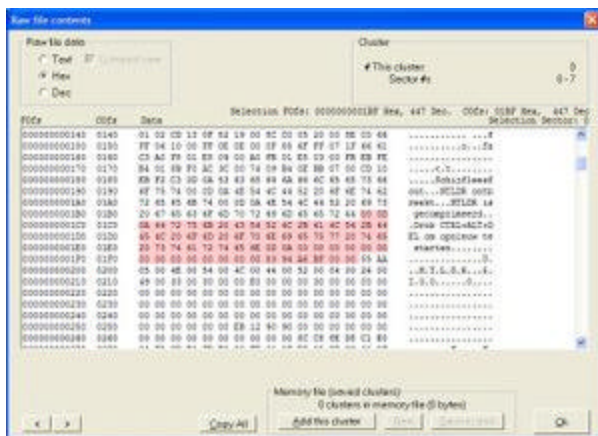
welke clusters bij het volume horen alsmede welke gevuld zijn danwel beschikbaar zijn voor allocatie. Elke byte in het \$Bitmap bestand houdt de administratie van 8 clusters bij. De 8 individuele bits in een byte refereren naar aparte clusters. Voorbeeld: een byte met een hexadecimale waarde 3F wordt binair uitgedrukt als 0011-1111. Van de acht clusters die door deze byte worden vertegenwoordigd zijn er 6 gealloceerd (niet beschikbaar) en 2 zijn wel beschikbaar. De waarde 0 geeft aan dat het cluster beschikbaar is, de waarde 1 geeft aan dat de cluster gealloceerd is.

### Master partition table

De master partition table kan maximaal vier partities beschrijven. Elke partitiebeschrijving wordt een partition record genoemd, elke record is 16 bytes lang. Een partition record bestaat uit:

1. Actief (1 byte, 80 = ja, 00 = nee)
2. Start sector in CHS formaat (3 bytes)
3. Partitie type (1 byte)
4. Eind sector in CHS formaat (3 bytes)
5. Relatieve sector offset (4 bytes)
6. Totaal aantal sectoren in de partitie (4 bytes)

Je kunt de master partition table vinden vanaf sector offset 446. De master partition table is 64 bytes lang en eindigt direct voor de hexadecimale code 55 AA (dat tevens het einde van de MBR is). In Disk Investigator ziet dat er zo uit:



In deze 64 bytes lange hexadecimale reeks kan NTFS terugvinden hoeveel partities er zijn, in welke clusters deze partities beginnen en uit hoeveel clusters de partities bestaan.

### Volume Boot Record

Nu de MBR gevonden is, is het niet zo moeilijk te zien waar het relatieve startpunt is van de Volume Boot Record (VBR). Als je kijkt met behulp van Disk Investigator, dan zie je in sector 3 de term NTFS staan (4E 54 46 53 in hexadecimale codes). Deze term wordt de OEM vendor name genoemd, die is verschenen nadat de harddisk is geformatteerd.

De grootte van een partitie (het aantal sectoren waaruit de partitie bestaat) kan NTFS terugvinden in de VBR vanaf sector offset 40. De waarde is 8 bytes lang (bijvoorbeeld 40 6E 3B 00 00 00 00). Als je deze 8 bytes zou decoderen via Little Endian dan zou je de grootte (het aantal sectoren van de partitie) in beeld zien. Helaas heb ik nog geen gratis Little Endian decoders aangetroffen op het internet.

De laatste sector van een volume bevat een backup van de VBR. Deze laatste sector wordt verder niet door

NTFS meegeteld, puur en alleen omdat deze sector bedoeld is als backup voor de VBR en nergens anders voor gebruikt mag worden.

Er zijn verschillende manieren om een harddisk te partitioneren. Zoals al eerder genoemd kan de master partition table maximaal vier beschrijvingen bevatten. Over het algemeen worden er twee gebruikt. De eerste beschrijving betreft over het algemeen een Primary partitie (ook wel boot volume genoemd, bijvoorbeeld C:). De tweede beschrijving kan ook een Primary partitie zijn, maar het kan ook een Extended partitie zijn waarin één of meerdere volumes zijn opgenomen (bijvoorbeeld D: en E:).

NTFS partities en hun omvang

Je kunt onder NTFS gebruik maken van primaire of logische partities. Logische partities passen binnen een extended partitie. Oorspronkelijk is NTFS bedoeld voor gebruik in bedrijfs- en kantooromgevingen en daarom staat dit bestandssysteem het toe om zeer grote partities te maken. De maximum partitie onder FAT16 en Windows is 2 GB (32 kilobyte clusters) of 4 GB (64 kilobyte clusters, hetgeen oude Windows versies niet ondersteunen).

Bedrijven en kantoren hebben een andere wens wat partities betreft, en velen maken zelfs gebruik van RAID om nog grotere volumes aan te maken. Daarom zijn de partities die onder FAT16 mogelijk waren totaal niet meer voldoende. Daarom is zo'n tien jaar geleden NTFS ontwikkeld.

Onder NTFS is een maximum formaat voor een partitie 2 tot de 64e macht ( $2 \times 2 \times 2$  etc. (64 keer)), wat overeenkomt met 18.446.744.073.709.551.616 bytes (oftewel 16 exabyte).

NTFS directories (mappen)

Net als vrijwel elk ander bestandssysteem houdt NTFS voor de rangschikking van bestanden en mappen een hiërarchisch model (boomstructuur) aan. Aan de basis ligt de root directory (een van de Metadata bestanden van NTFS). Binnen deze root directory worden verwijzingen naar alle andere bestanden opgeslagen. Elke map kan op zijn beurt weer een combinatie van bestanden en/of andere mappen bevatten. Anders dan in FAT zijn bestanden onder NTFS verzamelingen van attributen. De bestanden kunnen hier, naast de inhoudelijke data, hun eigen beschrijving bevatten. Een NTFS map houdt alleen informatie over de map bij en niet over de bestanden die in de map staan.

Alles (zowel bestanden als mappen) wordt binnen NTFS beschouwd als een bestand. Elke map heeft een record in de Master File Table (zie boven). Het MFT record dat bij een map hoort bevat:

Header (H)

Standard Information Attribute (SI)

File Name Attribute (FN)

Index Root Attribute

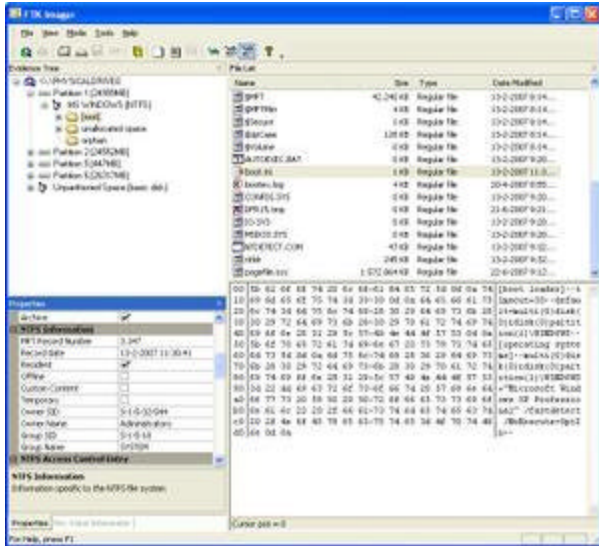
Index Allocation Attribute

Security Descriptor (SD) Attribute (waaronder de Access Control List)

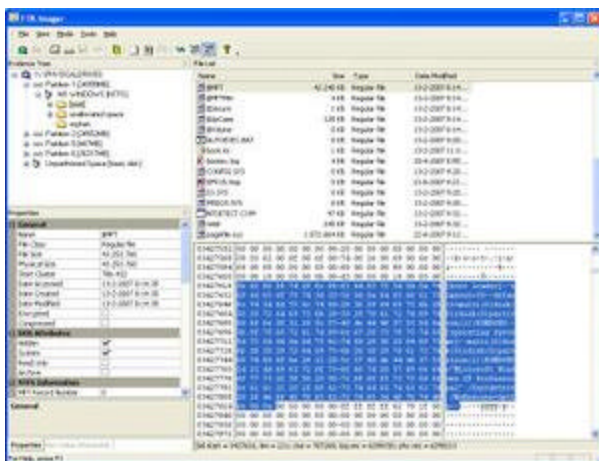
Kleine mappen worden geheel in hun MFT record opgeslagen, net als kleine bestanden. Informatie met betrekking tot grotere bestanden is onderverdeeld over meerdere records, waarnaar wordt verwezen in de root directory van de MFT.

Bestanden die bijvoorbeeld geheel in de MFT worden ondergebracht zijn internet cookies, kleine afbeeldingen en kleine tekstbestanden. Een voorbeeld hiervan is het bestand boot.ini

boot.ini kan een formaat hebben van 200 kb of minder.



In bovenstaand voorbeeld heeft boot.ini een logische en fysieke omvang van 211 bytes. Dit past dus compleet in de 1024 bytes die de MFT voor het bestand ter beschikking heeft. Er zal echter nooit file slack overblijven. Het MFT record nummer voor boot.ini is hier 3347. Tevens geeft FTK hier aan dat het bestand resident is. Als we de MFT bekijken, en dan record 3347 opzoeken, dan zien we boot.ini in zijn geheel staan:



Tip: je kunt zoeken naar een gedeelte in de MFT via de rechtermuisknop > find > boot.ini

Om aan de snelheid tegemoet te komen hebben NTFS mappen een speciale managementstructuur, genaamd B-tree. Het concept hiervan is overgenomen van ontwerpen van relationele databases. Dankzij B-tree zijn de mappen onder NTFS zelf-sorterend.

NTFS bestanden en data opslag

Net als de meeste andere bestandssystemen wordt de meest fundamentele unit waarin data wordt opgeslagen bestand genoemd. Een bestand is een collectie data en kan van alles bevatten, zoals programmacode, tekst, afbeelding, audio etc. Het bestandssysteem maakt geen onderscheid tussen bestanden. Een bestand wordt door een specifieke applicatie weer verder geïnterpreteerd.

Onder NTFS zijn bestanden opgeslagen als een collectie attributen. Hieronder valt ook de data in het bestand (data attribute). De manier waarop de data in het bestand is opgeslagen is afhankelijk van het formaat van het bestand. De structuur is over het algemeen als volgt:

Header (H)

Standard Information Attribute (SI)

File Name Attribute (FN)

Data (Data) Attribute

Security Descriptor (SD) Attribute (waaronder de Access Control List)

Hiernaast zijn ook andere, aanvullende attributen mogelijk. Als een bestand klein genoeg is zodat alle attributen in de MFT record passen, dan wordt het gehele bestand in de MFT opgeslagen (zie boven).

NTFS bestandsformaat

Een van de belangrijkste eisen van bedrijfsapplicaties (met name databases) onder Windows is dat bestanden voldoende groot kunnen worden. Onder FAT is de maximum grootte voor bestanden 2 GB, soms 4 GB.

Zoals al eerder uitgelegd probeert NTFS eerst om het bestand onder te brengen in de MFT record. Als het bestand te groot is wordt het buiten de MFT geplaatst. Als er maar ruimte genoeg is op de disk dan is er feitelijk geen maximum grootte voor een bestand. Een enkel bestand kan als het ware de gehele volume voor zijn rekening nemen (behalve dan de ruimte voor de MFT en andere belangrijke bestanden).

NTFS heeft ook file-based compression waarmee grote bestanden veel minder ruimte in kunnen nemen. Ook biedt NTFS ondersteuning voor sparse bestanden, die speciaal geschikt zijn voor specifieke applicaties.

Bestandsnamen

De eerste versies van MS-Dos waren niet flexibel wat de lengte van bestandsnamen betrof. Hierbij werd het 8.3 principe toegepast (8 tekens voor de bestandsnaam en 3 tekens voor de bestandsextensie). Onder NTFS kan een bestandsnaam 255 karakters bevatten. Elk karakter is in principe mogelijk. Alle bestandsnamen worden onder NTFS in unicode opgeslagen, terwijl het vroeger gebruik was om bestandsnamen als ASCII op te slaan.

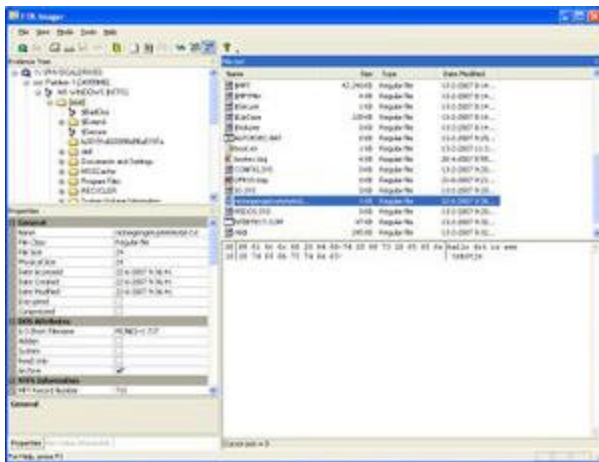
Het bestandssysteem maakt toch nog automatisch een korte bestandsnaam aan (8.3), zodat ook oudere software nog kan worden gebruikt onder NTFS.

Bestandsnamen worden in de MFT opgeslagen in de File Name Attribute van het betreffende bestand. Zolang de naam niet langer is dan 8.3 dan is alleen een enkele Name Attribute in de MFT aanwezig. Als een bestandsnaam langer is dan wordt een tweede File Name Attribute in de MFT aangemaakt, de eerste blijft echter ook intact. Beide File Name Attributes worden in unicode weergegeven.

Voorbeeld:

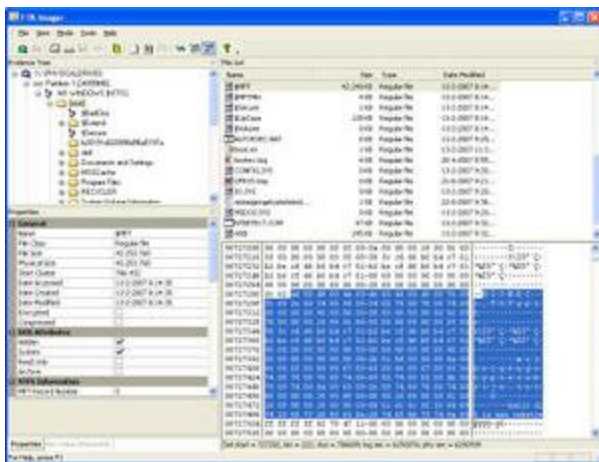
Met behulp van Notepad maak ik een klein tekstberichtje aan en noem dat mijneigengetyptetekstje.txt

Ik sla het tekstberichtje op onder c:\ en bekijk vervolgens met behulp van FTK Imager het bestand:



FTK geeft al aan dat het bestand 24 bytes groot is (klein genoeg dus om compleet in de MFT te passen) en dat de korte bestandsnaam MIJNEI~1.TXT is. De lange bestandsnaam (zoals ik het bestand genoemd heb) wordt ook vermeld.

Ik zoek vervolgens in de \$MFT of ik het complete bestandje terug kan vinden:



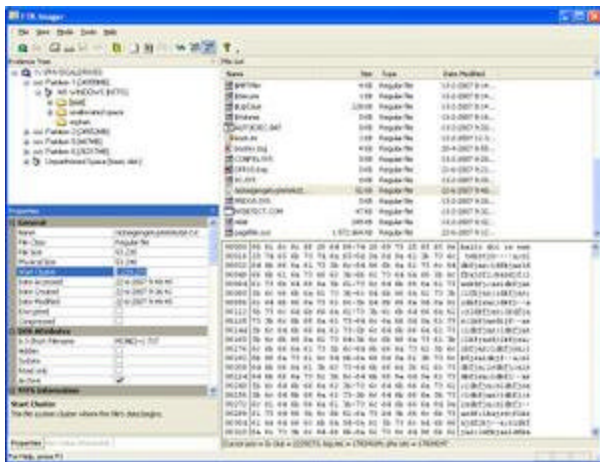
Uiteraard is dat het geval. Achtereenvolgens zie je staan:

korte bestandsnaam

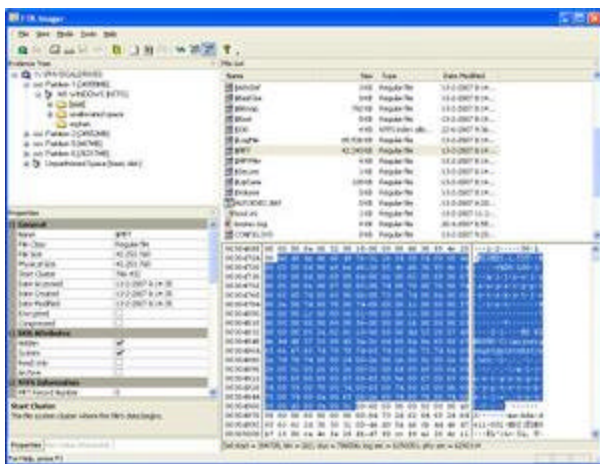
lange bestandsnaam

inhoud van het tekstbestandje

Als ik het tekstbestand nu groter maak dan 1024 bytes door er allerlei tekens bij te zetten, dan zie ik via FTK Imager het volgende als ik het bestand zelf bekijk:



Nu is er een vermelding bijgekomen dat het bestand begint in cluster 2229273 (het is dus niet meer resident aanwezig in de MFT). Als ik de MFT bekijk dan zie ik het volgende staan:



Zowel de korte als de lange bestandsnamen zijn nog aanwezig, echter de inhoud van het tekstbestandje is uit de MFT verdwenen.