Restauration logicielle de fichiers effacés sur un disque dur NTFS

Aspects techniques

Nicolas Paglieri
www.ni69.info
14 juin 2009

Introduction

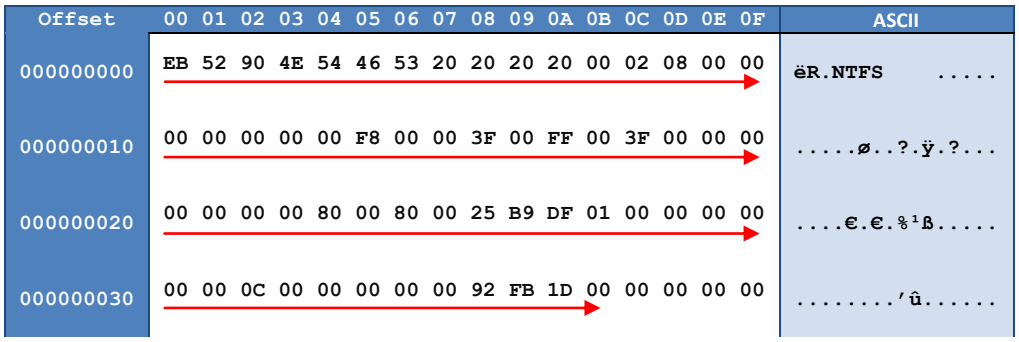
La perte ou la suppression accidentelle de données numériques est un problème courant dans notre société de plus en plus régie par l'informatique. Cependant cela peut s'avérer catastrophique suivant l'importance de l'information disparue. Dans un environnement basé sur le système de fichiers NTFS, quand un fichier est supprimé du disque, même après le vidage de la corbeille sous Windows, il n'a en réalité pas totalement disparu. Il existe en effet un moyen de restaurer l'information auparavant contenue dans le fichier, sous certaines conditions qui seront explicitées dans cette étude.

Nous axerons ce dossier sur un exemple concret de recherche et de restauration d'un fichier effacé sur un disque dur. Une étude exhaustive du système de fichiers NTFS est ici exclue. Nous nous limiterons à l'analyse des seules notions, caractéristiques et seuls aspects techniques essentiels à l'atteinte de cet objectif.

Ce document sert de documentation à un programme réalisé en Delphi et disponible sur mon site internet.

Pré-requis techniques

Bien que les données stockées sur le disque le soient sous forme binaire, nous en adopterons une **représentation hexadécimale**, bien plus lisible. Chaque octet (1 octet = 8 bits) sera ainsi symbolisé par 2 caractères hexadécimaux. Il en sera également proposé une représentation dans l'encodage ASCII. Un aperçu des données sera présenté chaque fois que cela sera jugé nécessaire à la compréhension des concepts évoqués. Les **aperçus de données** se lisent ligne par ligne de gauche à droite et de haut en bas. La colonne Offset à gauche représente l'adresse du premier octet de chaque ligne. Le complément d'adresse (octets suivants) est situé au dessus de chaque colonne. Une représentation ASCII de chaque ligne est affichée à droite.



Offset	00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F	ASCII
000000000	EB 52 90 4E 54 46 53 20 20 20 20 00 02 08 00 00	ëR.NTFS .....
000000010	00 00 00 00 00 F8 00 00 3F 00 FF 00 3F 00 00 00	.....ø..?.ÿ.?...
000000020	00 00 00 00 80 00 80 00 25 B9 DF 01 00 00 00 00	....€.€.%¹ß.....
000000030	00 00 0C 00 00 00 00 00 92 FB 1D 00 00 00 00 00	........'û.....

Attention aux confusions : **octet** (français) = **Byte** (anglais) et **bit** (français) = **bit** (anglais)

Nous prendrons comme convention **1Ko** = **1KB** = **1024 octets**. Les différents **types de données** employés sont :

Nom	Mémoire occupée
`Byte`	1 octet
`Word`	2 octets
`DWord`	4 octets
`Int64`	8 octets

Les **nombres hexadécimaux**, pour les différencier des nombres décimaux, seront précédés du caractère **$** lorsqu'il en sera fait mention en dehors des aperçus de données, eux toujours représentés en hexadécimal.

Indications sur la **typographie** : Les objets informatiques, tels les types de données, les noms spécifiques des champs et paramètres, les noms et attributs de fichiers, les nombres hexadécimaux seront toujours présentés selon la police d'écriture **`Courier New`** . Tout le reste, tels les textes explicatifs, remarques, nombres décimaux, sera typographié avec la police d'écriture courante Calibri.

Remarque importante sur l'encodage des données **little-endian** : L'octet de poids le plus faible est stocké en premier. Ainsi par exemple le nombre hexadécimal **`$F512A83E`** (`DWord`) sera représenté en mémoire par : `3E A8 12 F5`

Toutes les dates sont stockées en mémoire sous la forme d'un `Int64`, dénombrant le nombre de centaines de nanosecondes écoulées depuis le 1er janvier 1601 selon la convention.

Généralités sur le système de fichiers NTFS

NTFS (NT File System) est un système de fichiers, c'est-à-dire une structure de données régissant l'organisation interne des données sur un disque logique, autrement appelé **volume logique**.

Chaque disque est divisé en **secteurs** (dont la taille est généralement de 512 octets). Un **cluster** réunit un certain nombre de secteurs, variable suivant le matériel.

Taille du volume	Taille de cluster	Nombre de secteurs
moins de **512 Mo**	512 octets	1
entre **512 Mo** et **1 Go**	1 Ko	2
entre **1 Go** et **2 Go**	2 Ko	4
plus de **2 Go**	4 Ko	8

Deux moyens permettent de repérer un cluster sur le disque :

- **LCN (Logical Cluster Number)** : Numéro du cluster absolu par rapport au début du disque (LCN 0)
- **VCN (Virtual Cluster Number)** : Numéro du cluster relatif par rapport au cluster précédemment lu

On peut schématiser de manière simplifiée un disque logique NTFS :

Secteur de Boot

Le premier secteur de tout disque logique correctement formaté (que le format de fichiers soit NTFS ou non) est une zone dite **Secteur de Boot**, contenant des informations générales sur le disque. C'est une séquence d'amorçage de taille fixe, située au tout début du disque. Nous détaillerons uniquement les informations étant nécessaires pour la suite. Certaines notions relatives à la MFT seront expliquées dans la partie suivante.

Structure du secteur de Boot :

Nom	Type	Repérage
_jmpcode	array[1..3] of Byte	
cOEMID	array[1..8] of Byte	
wBytesPerSector	Word	
bSectorsPerCluster	Byte	
wSectorsReservedAtBegin	Word	
Mbz1	Byte	
Mbz2	Word	
Reserved1	Word	
bMediaDescriptor	Byte	
Mbz3	Word	
wSectorsPerTrack	Word	
wSides	Word	
dwSpecialHiddenSectors	DWord	
Reserved2	DWord	
Reserved3	DWord	
TotalSectors	Int64	
MftStartLcn	Int64	
Mft2StartLcn	Int64	
ClustersPerFileRecord	DWord	
ClustersPerIndexBlock	DWord	
VolumeSerialNumber	Int64	
_loadercode	array[1..430] of Byte	
wSignature	Word	

Aperçu du secteur de boot d'un disque NTFS :

Offset	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	ASCII
000000000	EB	52	90	4E	54	46	53	20	20	20	20	00	02	08	00	00	ëR.NTFS .....
000000010	00	00	00	00	00	F8	00	00	3F	00	FF	00	3F	00	00	00	.....ø..?.ÿ.?...
000000020	00	00	00	00	80	00	80	00	25	B9	DF	01	00	00	00	00	....€.€.%¹ß.....
000000030	00	00	0C	00	00	00	00	00	92	FB	1D	00	00	00	00	00	........'û......
000000040	F6	00	00	00	01	00	00	00	37	6E	52	48	A3	52	48	6A	ö.......7nRH£RHj
000000050	00	00	00	00	FA	33	C0	8E	D0	BC	00	7C	FB	B8	C0	07	....ú3ÀŽÐ¼.|û¸À.
000000060	8E	D8	E8	16	00	B8	00	0D	8E	C0	33	DB	C6	06	0E	00	ŽØè..¸..ŽÀ3ÛÆ...
000000070	10	E8	53	00	68	00	0D	68	6A	02	CB	8A	16	24	00	B4	.èS.h..hj.ËŠ.$.´
000000080	08	CD	13	73	05	B9	FF	FF	8A	F1	66	0F	B6	C6	40	66	.Í.s.¹ÿÿŠñf.¶Æ@f
000000090	0F	B6	D1	80	E2	3F	F7	E2	86	CD	C0	ED	06	41	66	0F	.¶Ñ€â?÷â†ÍÀí.Af.
0000000A0	B7	C9	66	F7	E1	66	A3	20	00	C3	B4	41	BB	AA	55	8A	·Éf÷áf£ .Ã´A»ªUŠ
0000000B0	16	24	00	CD	13	72	0F	81	FB	55	AA	75	09	F6	C1	01	.$.Í.r..ûUªu.öÁ.
0000000C0	74	04	FE	06	14	00	C3	66	60	1E	06	66	A1	10	00	66	t.þ...Ãf`..f¡..f
0000000D0	03	06	1C	00	66	3B	06	20	00	0F	82	3A	00	1E	66	6A	....f;. ..,:..fj
0000000E0	00	66	50	06	53	66	68	10	00	01	00	80	3E	14	00	00	.fP.Sfh....€>...
0000000F0	0F	85	0C	00	E8	B3	FF	80	3E	14	00	00	0F	84	61	00	....è³ÿ€>....„a.
000000100	B4	42	8A	16	24	00	16	1F	8B	F4	CD	13	66	58	5B	07	´BŠ.$...‹ôÍ.fX[.
000000110	66	58	66	58	1F	EB	2D	66	33	D2	66	0F	B7	0E	18	00	fXfX.ë-f3Òf.·...
000000120	66	F7	F1	FE	C2	8A	CA	66	8B	D0	66	C1	EA	10	F7	36	f÷ñþÂŠÊf‹ÐfÁê.÷6
000000130	1A	00	86	D6	8A	16	24	00	8A	E8	C0	E4	06	0A	CC	B8	..†ÖŠ.$.ŠèÀä..Ì¸
000000140	01	02	CD	13	0F	82	19	00	8C	C0	05	20	00	8E	C0	66	..Í..‚..ŒÀ. .ŽÀf
000000150	FF	06	10	00	FF	0E	0E	00	0F	85	6F	FF	07	1F	66	61	ÿ...ÿ.....…oÿ..fa
000000160	C3	A0	F8	01	E8	09	00	A0	FB	01	E8	03	00	FB	EB	FE	Ã ø.è.. û.è..ûëþ
000000170	B4	01	8B	F0	AC	3C	00	74	09	B4	0E	BB	07	00	CD	10	´.‹ð¬<.t.´.».Í.
000000180	EB	F2	C3	0D	0A	45	72	72	2E	20	6C	65	63	74	75	72	ëòÃ..Err. lectur
000000190	65	20	64	69	73	71	75	65	00	0D	0A	4E	54	4C	44	52	e disque...NTLDR
0000001A0	20	6D	61	6E	71	75	65	00	0D	0A	4E	54	4C	44	52	20	manque...NTLDR
0000001B0	65	73	74	20	63	6F	6D	70	72	65	73	73	82	00	0D	0A	est compress,...
0000001C0	45	6E	74	72	65	7A	20	43	74	72	6C	2B	41	6C	74	2B	Entrez Ctrl+Alt+
0000001D0	53	75	70	70	72	20	70	6F	75	72	20	72	65	64	82	6D	Suppr pour red,m
0000001E0	61	72	72	65	72	0D	0A	00	0D	0A	00	00	00	00	00	00	arrer...........
0000001F0	00	00	00	00	00	00	00	00	83	99	A8	BE	00	00	55	AA	........ƒ™¨¾..Uª

Le champ `cOEMID` permet de dire qu'il s'agit bien là d'un disque NTFS.
Le champ `wBytesPerSector` donne le nombre d'octets par secteur. On retrouve la valeur `$200` = 512, attendue.
Le champ `bSectorsPerCluster` donne le nombre de secteurs par cluster. Il y en a ici 8.
Les champs `MftStartLcn` et `ClustersPerFileRecord` seront expliqués dans la partie suivante.

MFT (Master File Table)

La MFT (Master File Table) est une spécificité du système de fichiers NTFS. Il s'agit concrètement d'un **tableau d'enregistrements** de taille fixe, dont chaque enregistrement représente un unique objet du disque. Les 16 premiers enregistrements sont réservés à des fichiers de métadonnées spécifiques au système de fichiers : `$MFT` `$MFTMIRR` `$LOGFILE` `$VOLUME` `$ATTRDEF` `$BITMAP` `$BOOT` `$BADCLUS` `$SECURE` `$UPCASE` `$EXTEND`

Le champ `MftStartLcn` vu au paragraphe précédent donne accès au LCN de la MFT. Pour obtenir son adresse physique, il suffit de multiplier cette valeur par `wBytesPerSector` puis par `bSectorsPerCluster`. On obtient ainsi : `$C0000` × `$200` × `$8` = `$C0000000`

Le champ `ClustersPerFileRecord` est plus problématique : c'est une valeur hexadécimale signée. En effet quand la taille d'un cluster est supérieure à la taille d'un enregistrement (et c'est très souvent le cas), on ne peut pas exploiter directement sa valeur. Nous poserons une nouvelle variable `BytesPerFileRecord` qui contiendra le nombre d'octets par enregistrement, plus simple à utiliser.

Si `ClustersPerFileRecord < $80, BytesPerFileRecord = ClustersPerFileRecord × BytesPerCluster`
Sinon, si `ClustersPerFileRecord ≥ $80, BytesPerFileRecord = 1 lsh ($100 - ClustersPerFileRecord)`
On a donc dans notre exemple `BytesPerFileRecord = 1 lsh ($100 – $F6) = $400 =` 1024 octets

Première approche du problème

Le processus de restauration sera donc le suivant : Il s'agira de parcourir tous les enregistrements de la MFT à la recherche de l'entrée correspondant au fichier supprimé que nous voulons restaurer, localiser grâce à l'enregistrement les données le constituant sur le disque, et enfin les écrire dans un nouveau fichier.

Notion d'Enregistrement

Tous les enregistrements de la MFT (y compris les 16 premiers malgré leur rôle particulier) sont composés d'une structure identique : un entête, suivi d'attributs consécutifs, et d'un marqueur de fin (**$FFFFFFF**). Voici la structure des **entêtes d'enregistrements** :

Nom	Type	Repérage
Identifier	array[1..4] of Byte	
UsaOffset	Word	
UsaCount	Word	
LSN	Int64	
SequenceNumber	Word	
ReferenceCount	Word	
AttributesOffset	Word	
Flags	Word	
BytesInUse	DWord	
BytesAllocated	DWord	
BaseFileRecord	Int64	
NextAttributeID	Word	
Padding	Word	
RecordNumber	DWord	
UsaNumber	Word	
UsaArray	array of Word	

Offset	00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F	ASCII
000000000	46 49 4C 45 30 00 03 00 89 2E 51 25 00 00 00 00	FILE0...‰.Q%....
000000010	01 00 01 00 38 00 01 00 D8 01 00 00 00 04 00 00	....8...Ø.......
000000020	00 00 00 00 00 00 00 00 06 00 00 00 1E 00 00 00	....
000000030	76 02 72 00 00 00 00 00	v.r.....

Le champ **Identifier** contient toujours FILE quand l'enregistrement décrit un fichier ou un répertoire (si un problème a été détecté par chkdsk dans l'enregistrement, la valeur devient **BAAD**. Certains enregistrements spécifiques au système d'exploitation se voient également attribuer d'autres valeurs comme **INDX**). Le champ **AttributesOffset** renseigne sur l'adresse du 1er attribut par rapport au début de l'enregistrement. Le champ **Flags** caractérise l'état de disponibilité et le genre du fichier décrit par l'enregistrement :

Valeur de Flags	Signification
$00	Fichier supprimé
$01	Fichier disponible
$02	Répertoire supprimé
$03	Répertoire disponible

Notion d'Attribut

Chaque attribut est lui-même constitué d'un entête et d'une séquence de données. Voici la structure d'entête des attributs :

Nom	Type	Repérage
AttributeType	DWord	
Length	DWord	
NonResident	Byte	
NameLength	Byte	
NameOffset	Word	
Flags	Word	
AttributeNumber	Word	

Offset	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	ASCII
000000000	10	00	00	00	60	00	00	00	00	00	00	00	00	00	00	00	....`...........

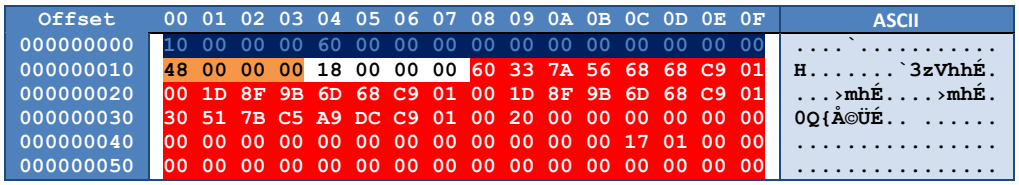
Il existe plusieurs types d'attributs, repérés chacun par une valeur différente de `AttributeType` :

AttributeType	Nom
$10	StandardInformation
$20	AttributeList
$30	FileName
$40	ObjectId
$50	SecurityDescriptor
$60	VolumeName
$70	VolumeInformation
$80	Data
$90	IndexRoot
$A0	IndexAllocation
$B0	Bitmap
$C0	ReparsePoint
$D0	EAInformation
$E0	EA
$F0	PropertySet
$100	LoggedUtilityStream

Le champ `Length` donne la longueur de l'attribut en octets, en incluant l'entête de l'attribut dans le décompte. Un attribut peut être dit **résident** (les données sont alors contenues directement dans l'attribut), ou bien **non-résident** (les données sont trop longues pour être contenues directement dans l'attribut qui a une longueur limitée par la taille de l'enregistrement, et sont ainsi placées ailleurs sur le disque ; on dispose alors uniquement de leur emplacement sur le disque sous forme d'une liste chaînée nommée `DataRuns` qui sera détaillée plus loin). Le champ `NonResident` renseigne sur ce caractère (`$00` = résident ; `$01` = non-résident).

Structure d'attribut résident

Nom	Type	Repérage
[Entête Atttibut]	array[1..16] of Byte	■
ValueLength	DWord	■
ValueOffset	Word	
Flags	Byte	
Padding	Byte	
Value	array of Byte	■

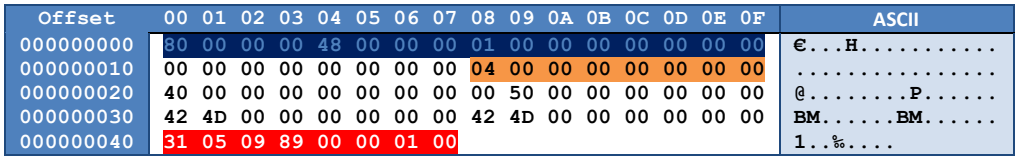


Offset	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	ASCII
000000000	10	00	00	00	60	00	00	00	00	00	00	00	00	00	00	00	....`..........
000000010	48	00	00	00	18	00	00	00	60	33	7A	56	68	68	C9	01	H.......`3zVhhÉ.
000000020	00	1D	8F	9B	6D	68	C9	01	00	1D	8F	9B	6D	68	C9	01	...›mhÉ....›mhÉ.
000000030	30	51	7B	C5	A9	DC	C9	01	00	20	00	00	00	00	00	00	0Q{Å©ÜÉ.. ......
000000040	00	00	00	00	00	00	00	00	00	00	00	00	17	01	00	00	....
000000050	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....

Le champ **ValueLength** donne la longueur du champ **Value**, ce dernier contenant lui les données en elles-mêmes, dont le contenu varie selon le type d'attribut considéré.

Structure d'attribut non résident

Nom	Type	Repérage
[Entête Atttibut]	array[1..16] of Byte	■
LowVCN	Int64	
HighVCN	Int64	■
DataRunsOffset	Word	
CompressionUnit	Byte	
Padding	array[1..5] of Byte	
AllocatedSize	Int64	
DataSize	Int64	
InitializedSize	Int64	
CompressedSize	Int64	
Name	array of Byte	
DataRuns	array of Byte	■



Offset	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	ASCII
000000000	80	00	00	00	48	00	00	00	01	00	00	00	00	00	00	00	€...H...........
000000010	00	00	00	00	00	00	00	00	04	00	00	00	00	00	00	00	....
000000020	40	00	00	00	00	00	00	00	00	50	00	00	00	00	00	00	@........P......
000000030	42	4D	00	00	00	00	00	00	42	4D	00	00	00	00	00	00	BM......BM......
000000040	31	05	09	89	00	00	01	00									1..‰....

CompressedSize n'est présent que si les données sont compressées (ce n'est pas le cas dans l'exemple).
Name n'est présent que si l'attribut est nommé (ce n'est pas le cas dans l'exemple).
HighVCN ne nous sera utile que lorsque nous aurons besoin de connaître la taille de la MFT (détails plus loin).

Nous analyserons par la suite uniquement les attributs **StandardInformation** (résident), **FileName** (résident), et **Data** (résident ou non résident) à partir d'un exemple d'enregistrement de fichier.

Exemples d'Attributs

Attribut StandardInformation

Cet attribut est toujours résident quel que soit l'enregistrement. Structure du champ **Value** correspondant :

Nom	Type	Repérage
CreationTime	Int64	
ChangeTime	Int64	
LastWriteTime	Int64	
LastAccessTime	Int64	
FileAttributes	DWord	
Alignment	array[1..3] of DWord	
QuotaID	DWord	
SecurityID	DWord	
QuotaCharge	Int64	
USN	Int64	

Offset	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	ASCII
000000000	60	33	7A	56	68	68	C9	01	00	1D	8F	9B	6D	68	C9	01	`3zVhhÉ....›mhÉ.
000000010	00	1D	8F	9B	6D	68	C9	01	30	51	7B	C5	A9	DC	C9	01	...›mhÉ.0Q{Å©ÜÉ.
000000020	00	20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	. .....
000000030	00	00	00	00	17	01	00	00	00	00	00	00	00	00	00	00	.....
000000040	00	00	00	00	00	00	00	00									.....

Le Champ **CreationTime** renseigne sur la date de création du fichier, **ChangeTime** sur celle de dernière modification, et **LastAccessTime** sur celle du dernier accès.

Attribut FileName

Cet attribut est toujours résident quel que soit l'enregistrement. Il peut être présent plusieurs fois dans un même enregistrement (le nom du fichier peut être stocké plusieurs fois dans des espaces de nommage différents – voir plus loin). Voici la structure du champ **Value** correspondant :

Nom	Type	Repérage
DirectoryFileReferenceNumber	Int64	
CreationTime	Int64	
ChangeTime	Int64	
LastWriteTime	Int64	
LastAccessTime	Int64	
AllocatedSize	Int64	
DataSize	Int64	
FileAttributes	DWord	
AlignmentOrReserved	DWord	
NameLength	Byte	
NameType	Byte	
Name	array of Word	

Offset	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	ASCII
000000000	C2	16	00	00	00	00	0D	00	00	31	8C	2C	96	DC	C9	01	Â.....1Œ,–ÜÉ.
000000010	00	31	8C	2C	96	DC	C9	01	00	31	8C	2C	96	DC	C9	01	.1Œ,–ÜÉ..1Œ,–ÜÉ.
000000020	00	31	8C	2C	96	DC	C9	01	00	00	00	00	00	00	00	00	.1Œ,–ÜÉ.....
000000030	00	00	00	00	00	00	00	00	20	20	00	00	00	00	00	00	....
000000040	0B	03	55	00	6E	00	65	00	72	00	61	00	73	00	65	00	..U.n.e.r.a.s.e.
000000050	2E	00	65	00	78	00	65	00									..e.x.e.

Les quatre premiers octets du champ **DirectoryFileReferenceNumber** correspondent au numéro de l'enregistrement du répertoire parent dans la MFT (ici, ce numéro est ainsi **$16C2** = 5826). Les champs **CreationTime**, **ChangeTime**, **LastWriteTime**, **LastAccessTime**, **AllocatedSize** et **DataSize** ne fournissent pas des informations correctes (se référer aux champs de même nom dans les attributs **StandardInformation** et **Data**). **NameLength** représente la longueur du nom de fichiers. **NameType** représente l'espace de nommage.

NameType	Espace de Nommage
$00	POSIX
$01	WIN32
$02	DOS
$03	WIN32 & DOS

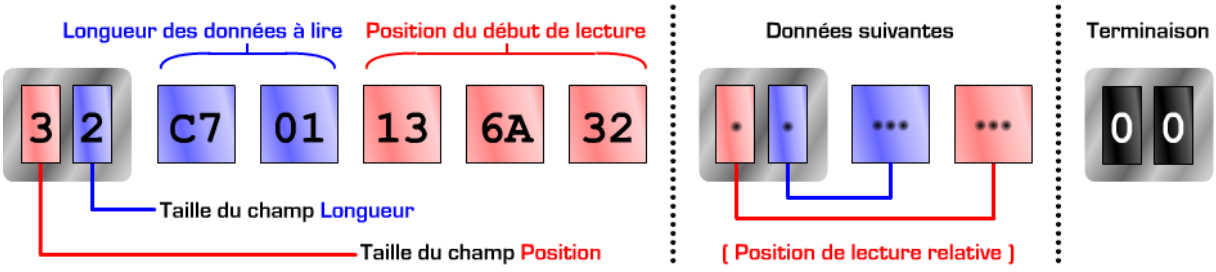
Le champ **Name** contient pour finir le nom de fichier au format Unicode.

Attribut Data

Cet attribut est selon les cas soit résident, soit non-résident non-nommé. Tout dépend de la quantité de données constituant le fichier à stocker en mémoire. Dans le cas où l'attribut est résident, le contenu du champ **Value** est exactement le contenu du fichier. Dans le cas contraire, le champ **Value** est occupé par une liste chaînée (**DataRuns**) décrivant les emplacements du disque où trouver les données.

Notion de DataRuns

Les DataRuns sont des listes chaînées décrivant l'emplacement de données sur le disque. Les attributs non résidents sont stockés dans des Runs (intervalles de clusters). Chaque Run est représenté par l'adresse de son premier cluster, ainsi que sa longueur totale. Le premier octet de chaque Run renseigne sur la taille des deux champs qui le suivent : Longueur et Position. Le champ Position (ici **$326A13**) donne l'adresse du début des données (relativement à l'adresse du Run précédent, ici relativement au LCN 0 car il s'agit du premier Run), et le champ Longueur (ici **$1C7**) indique la quantité de clusters à lire à partir de cet emplacement. Les DataRuns peuvent ainsi décrire l'emplacement de plusieurs blocs de données, même disjoints, sur le disque. L'octet nul (**$00**) indique la fin de la chaîne. Les blocs de données ainsi lus et mis bout à bout forment le contenu du fichier.



Ce système permet de représenter à la fois des fichiers normaux, mais aussi des fichiers compressés, fragmentés, ou creux.

Notion de Fixup

Le système de fichiers NTFS est doté d'un procédé de détection d'erreurs dans les clusters du disque, permettant une protection de l'intégrité des données. L'entête de chaque enregistrement contient un élément de type **`Word`** nommé **`UsaNumber (Update Sequence Number)`**, et une chaîne d'éléments **`Word`** nommée **`UsaArray (Update Sequence Array)`**. La longueur totale de **`UsaNumber`** et **`UsaArray`** (nombre de **`Word`**) est donnée par **`UsaCount`**. Voir la partie Notion d'Enregistrement pour plus de détails sur les positions des champs.

Lors de l'écriture de données dans un enregistrement, les deux derniers octets de chaque secteur composant l'enregistrement sont copiés les uns à la suite des autres dans la chaîne **`UsaArray`**, puis sont remplacés par la valeur de **`UsaNumber`**. A chaque nouvelle écriture, **`UsaNumber`** est incrémenté de un.

Lors de la lecture des enregistrements directement à partir du disque, il est donc nécessaire de tenir compte de cette particularité afin de lire des données correctes. Il faut donc, après avoir lu l'intégralité de l'enregistrement, vérifier que les deux derniers octets de chaque secteur sont bien identiques à la valeur de **`UsaNumber`**, puis replacer les éléments de **`UsaArray`** à leur place initiale (dernier Word de chaque secteur). L'opération s'appelle un Fixup, et doit être réalisée avant chaque analyse de données d'un enregistrement.

Voici un aperçu des données initialement présentes sur le disque, puis des données une fois le Fixup réalisé. Sont représentés l'entête de l'enregistrement ainsi que les deux derniers octets de chaque secteur. On considère ici que l'enregistrement, de 1024 octets, comporte deux secteurs de 512 octets (cas le plus courant). Le champ **`UsaNumber`** est surligné en ■. Le champ **`UsaArray`** est surligné en ■.

Avant FixUp (données protégées) :

Offset	00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F	ASCII
000000000	46 49 4C 45 30 00 03 00 89 2E 51 25 00 00 00 00	FILE0...‰.Q%....
000000010	01 00 01 00 38 00 01 00 D8 01 00 00 00 04 00 00	....8...Ø.......
000000020	00 00 00 00 00 00 00 00 06 00 00 00 1E 00 00 00	....
000000030	76 02 72 00 00 00 00 00	v.r.....
	⋮	
0000001F0	[Fin du Secteur n°1 :] 76 02	v.r.
	⋮	
0000003F0	[Fin du Secteur n°2 :] 76 02	v.r.

Après FixUp (données correctes) :

Offset	00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F	ASCII
000000000	46 49 4C 45 30 00 03 00 89 2E 51 25 00 00 00 00	FILE0...‰.Q%....
000000010	01 00 01 00 38 00 01 00 D8 01 00 00 00 04 00 00	....8...Ø.......
000000020	00 00 00 00 00 00 00 00 06 00 00 00 1E 00 00 00	....
000000030	76 02 72 00 00 00 00 00	v.r.....
0000001F0	[Fin du Secteur n°1 :] 72 00	v.r.
0000003F0	[Fin du Secteur n°2 :] 00 00	v.r.

Aperçu d'un enregistrement de fichier résident

Enregistrement de 1024 octets sur deux secteurs de 512 octets. L'entête et les différents attributs sont colorés.

Offset	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	ASCII
0C09B0800	46	49	4C	45	30	00	03	00	90	D1	6A	25	00	00	00	00	FILE0....Ñj%....
0C09B0810	0C	00	02	00	38	00	01	00	18	02	00	00	00	04	00	00	....8...........
0C09B0820	00	00	00	00	00	00	00	00	07	00	00	00	C2	26	00	00	............Â&..
0C09B0830	02	00	20	66	00	00	00	00	10	00	00	00	60	00	00	00	.. f........`...
0C09B0840	00	00	00	00	00	00	00	00	48	00	00	00	18	00	00	00	........H.......
0C09B0850	40	E7	4B	BA	FB	EC	C9	01	20	71	EC	26	FC	EC	C9	01	@çK°ûìÉ. qì&üìÉ.
0C09B0860	80	5E	68	43	FC	EC	C9	01	C0	A2	D9	3F	FC	EC	C9	01	€^hCüìÉ.À¢Ù?üìÉ.
0C09B0870	20	20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0C09B0880	00	00	00	00	65	01	00	00	00	00	00	00	00	00	00	00	....e...........
0C09B0890	00	00	00	00	00	00	00	00	30	00	00	00	78	00	00	00	........0...x...
0C09B08A0	00	00	00	00	00	00	05	00	5A	00	00	00	18	00	01	00	........Z.......
0C09B08B0	50	23	00	00	00	00	01	00	C0	A2	D9	3F	FC	EC	C9	01	P#......À¢Ù?üìÉ.
0C09B08C0	20	71	EC	26	FC	EC	C9	01	20	71	EC	26	FC	EC	C9	01	qì&üìÉ. qì&üìÉ.
0C09B08D0	C0	A2	D9	3F	FC	EC	C9	01	28	00	00	00	00	00	00	00	À¢Ù?üìÉ.(.......
0C09B08E0	25	00	00	00	00	00	00	00	20	20	00	00	00	00	00	00	%....... .....
0C09B08F0	0C	02	45	00	58	00	45	00	4D	00	50	00	4C	00	7E	00	..E.X.E.M.P.L.~.
0C09B0900	31	00	2E	00	54	00	58	00	54	00	69	00	63	00	68	00	1...T.X.T.i.c.h.
0C09B0910	30	00	00	00	98	00	00	00	00	00	00	00	00	00	04	00	0...˜...........
0C09B0920	7A	00	00	00	18	00	01	00	50	23	00	00	00	00	01	00	z.......P#......
0C09B0930	C0	A2	D9	3F	FC	EC	C9	01	20	71	EC	26	FC	EC	C9	01	À¢Ù?üìÉ. qì&üìÉ.
0C09B0940	20	71	EC	26	FC	EC	C9	01	C0	A2	D9	3F	FC	EC	C9	01	qì&üìÉ.À¢Ù?üìÉ.
0C09B0950	28	00	00	00	00	00	00	00	25	00	00	00	00	00	00	00	(.......%.......
0C09B0960	20	20	00	00	00	00	00	00	1C	01	45	00	78	00	65	00	........E.x.e.
0C09B0970	6D	00	70	00	6C	00	65	00	20	00	64	00	65	00	20	00	m.p.l.e. .d.e. .
0C09B0980	46	00	69	00	63	00	68	00	69	00	65	00	72	00	20	00	F.i.c.h.i.e.r. .
0C09B0990	43	00	6F	00	75	00	72	00	74	00	2E	00	74	00	78	00	C.o.u.r.t...t.x.
0C09B09A0	74	00	69	63	69	20	6C	65	40	00	00	00	28	00	00	00	t.ici le@...(...
0C09B09B0	00	00	00	00	00	00	06	00	10	00	00	00	18	00	00	00	....
0C09B09C0	C1	BB	74	11	C4	4A	DE	11	B9	45	08	00	27	A3	72	D6	Á»t.ÄJÞ.¹E..'£rÖ
0C09B09D0	80	00	00	00	40	00	00	00	00	00	18	00	00	00	01	00	€...@...........
0C09B09E0	25	00	00	00	18	00	00	00	56	6F	69	63	69	20	6C	65	%.......Voici le
0C09B09F0	20	63	6F	6E	74	65	6E	75	20	64	65	20	63	65	02	00	contenu de ce..
0C09B0A00	69	63	68	69	65	72	20	74	65	78	74	65	2E	00	00	00	ichier texte....
0C09B0A10	FF	FF	FF	FF	82	79	47	11	00	00	00	00	00	00	00	00	ÿÿÿÿ‚yG........
0C09B0A20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0C09B0A30	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0C09B0A40	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0C09B0A50	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0C09B0A60	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0C09B0A70	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0C09B0A80	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0C09B0A90	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0C09B0AA0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0C09B0AB0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0C09B0AC0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0C09B0AD0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0C09B0AE0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0C09B0AF0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0C09B0B00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0C09B0B10	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0C09B0B20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0C09B0B30	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0C09B0B40	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0C09B0B50	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0C09B0B60	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0C09B0B70	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0C09B0B80	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0C09B0B90	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0C09B0BA0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0C09B0BB0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0C09B0BC0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0C09B0BD0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0C09B0BE0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0C09B0BF0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	02	00	....

Aperçu d'un enregistrement de fichier non résident

Enregistrement de 1024 octets sur deux secteurs de 512 octets. L'entête et les différents attributs sont colorés.

Offset	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	ASCII
0C0C5E000	46	49	4C	45	30	00	03	00	25	EA	74	25	00	00	00	00	FILE0...%êt%....
0C0C5E010	04	00	02	00	38	00	01	00	F0	01	00	00	00	04	00	00	....8...ð.......
0C0C5E020	00	00	00	00	00	00	00	00	05	00	00	00	78	31	00	00	............x1..
0C0C5E030	02	00	00	00	00	00	00	00	10	00	00	00	60	00	00	00	............`...
0C0C5E040	00	00	00	00	00	00	00	00	48	00	00	00	18	00	00	00	........H.......
0C0C5E050	60	9F	4D	D5	FF	EC	C9	01	00	7D	4C	A8	FF	EC	C9	01	`ŸMÕÿìÉ..}L¨ÿìÉ.
0C0C5E060	00	7D	4C	A8	FF	EC	C9	01	90	26	4F	D5	FF	EC	C9	01	.}L¨ÿìÉ..&OÕÿìÉ.
0C0C5E070	20	20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
0C0C5E080	00	00	00	00	65	01	00	00	00	00	00	00	00	00	00	00	....e...........
0C0C5E090	00	00	00	00	00	00	00	00	30	00	00	00	78	00	00	00	........0...x...
0C0C5E0A0	00	00	00	00	00	00	03	00	5A	00	00	00	18	00	01	00	........Z.......
0C0C5E0B0	98	24	00	00	00	00	06	00	60	9F	4D	D5	FF	EC	C9	01	˜$......`ŸMÕÿìÉ.
0C0C5E0C0	60	9F	4D	D5	FF	EC	C9	01	60	9F	4D	D5	FF	EC	C9	01	`ŸMÕÿìÉ.`ŸMÕÿìÉ.
0C0C5E0D0	60	9F	4D	D5	FF	EC	C9	01	00	00	00	00	00	00	00	00	`ŸMÕÿìÉ........
0C0C5E0E0	00	00	00	00	00	00	00	00	20	20	00	00	00	00	00	00	....
0C0C5E0F0	0C	02	45	00	58	00	45	00	4D	00	50	00	4C	00	7E	00	..E.X.E.M.P.L.~.
0C0C5E100	31	00	2E	00	54	00	58	00	54	00	69	00	63	00	68	00	1...T.X.T.i.c.h.
0C0C5E110	30	00	00	00	90	00	00	00	00	00	00	00	00	00	02	00	0...............
0C0C5E120	78	00	00	00	18	00	01	00	98	24	00	00	00	00	06	00	x.......˜$......
0C0C5E130	60	9F	4D	D5	FF	EC	C9	01	60	9F	4D	D5	FF	EC	C9	01	`ŸMÕÿìÉ.`ŸMÕÿìÉ.
0C0C5E140	60	9F	4D	D5	FF	EC	C9	01	60	9F	4D	D5	FF	EC	C9	01	`ŸMÕÿìÉ.`ŸMÕÿìÉ.
0C0C5E150	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0C0C5E160	20	20	00	00	00	00	00	00	1B	01	45	00	78	00	65	00	........E.x.e.
0C0C5E170	6D	00	70	00	6C	00	65	00	20	00	64	00	65	00	20	00	m.p.l.e. .d.e. .
0C0C5E180	46	00	69	00	63	00	68	00	69	00	65	00	72	00	20	00	F.i.c.h.i.e.r. .
0C0C5E190	4C	00	6F	00	6E	00	67	00	2E	00	74	00	78	00	74	00	L.o.n.g...t.x.t.
0C0C5E1A0	80	00	00	00	48	00	00	00	01	00	00	00	00	00	04	00	€...H...........
0C0C5E1B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0C0C5E1C0	40	00	00	00	00	00	00	00	00	10	00	00	00	00	00	00	@...............
0C0C5E1D0	93	05	00	00	00	00	00	00	93	05	00	00	00	00	00	00	".......".......
0C0C5E1E0	31	01	2B	88	00	00	01	00	FF	FF	FF	FF	82	79	47	11	1.+ˆ....ÿÿÿÿ,yG.
0C0C5E1F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	02	00	....
0C0C5E200	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0C0C5E210	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0C0C5E220	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0C0C5E230	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0C0C5E240	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0C0C5E250	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0C0C5E260	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0C0C5E270	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0C0C5E280	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0C0C5E290	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0C0C5E2A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0C0C5E2B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0C0C5E2C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0C0C5E2D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0C0C5E2E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0C0C5E2F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0C0C5E300	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0C0C5E310	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0C0C5E320	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0C0C5E330	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0C0C5E340	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0C0C5E350	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0C0C5E360	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0C0C5E370	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0C0C5E380	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0C0C5E390	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0C0C5E3A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0C0C5E3B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0C0C5E3C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0C0C5E3D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0C0C5E3E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0C0C5E3F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	02	00	....

Récupération effective des données

Tous les concepts généraux étant désormais énoncés, il ne reste plus qu'à présenter la procédure de récupération d'un fichier effacé, qui n'est qu'une juxtaposition de l'utilisation des principes vus auparavant.

Il faut tout d'abord localiser la MFT à l'aide du champ correspondant à son adresse dans le secteur de Boot.

Le premier enregistrement de la MFT (`$MFT`) a une structure commune à tous les autres enregistrements. Nous devons localiser son attribut `Data` qui est non résident, plus précisément la valeur du champ `HighVCN` qui renseigne sur la taille totale (en clusters) de la MFT. Il suffit de le multiplier par `BytesPerCluster` et de diviser cette taille par la taille d'un enregistrement `BytesPerFileRecord` pour obtenir le nombre d'enregistrements.

Le parcours des enregistrements de fait ensuite de manière linéaire. Tous les enregistrements doivent être scannés les uns après les autres à la recherche d'une valeur `$00` dans le champ `flag` de l'entête d'enregistrement. Si cette valeur est trouvée, on lit les attributs contenus dans l'enregistrement de proche en proche, en récupérant au passage les informations du fichier (nom, taille, dates, données).

Pour restaurer un fichier trouvé, il suffit d'enregistrer les données lues à partir des indications de l'attribut `Data` dans un autre fichier créé pour l'occasion. La récupération est ainsi terminée.

Limites de la méthode & Recommandations

Il est très important de prendre conscience que ce processus ne permet aucunement d'être certain de pouvoir effectuer la restauration, et ne permet pas non plus de garantir l'intégrité des données récupérées.

Les enregistrements de fichiers effacés étant marqués comme libres par le système de fichiers, ils sont susceptibles à tout moment d'être recouverts par les enregistrements d'autres fichiers nouvellement créés. À savoir que le remplissage des enregistrements libres se fait linéairement depuis le début de la MFT ; donc si un enregistrement [A] se libère au tout début du disque, il sera recouvert plus vite qu'un enregistrement [B] se libérant à la fin de la MFT. En effet, il y a statistiquement plus d'enregistrements libres entre le début de la MFT et l'enregistrement [B] qu'entre le début de la MFT et l'enregistrement [A].

De manière générale, **n'écrivez jamais quoi que ce soit sur le disque contenant des données importantes que vous venez juste de supprimer et que vous comptez récupérer**. Même l'installation d'un logiciel de récupération de données pourrait corrompre les données effacées du disque. Si les données vous sont très précieuses, déconnectez le disque dur de votre ordinateur et branchez-le sur un autre ordinateur où a été installé un logiciel de récupération ; ou bien utilisez un logiciel ne nécessitant pas d'installation et pouvant s'exécuter à partir d'un disque amovible, d'une clé USB, d'un CD, d'une disquette ou d'un emplacement réseau.

Ne sauvegardez pas les données récupérées par un logiciel de restauration directement sur le disque contenant encore des données effacées que vous désirez récupérer. L'écriture de fichiers sur le disque peut interférer avec le processus de restauration et recouvrir des enregistrements de la MFT ou des zones du disque contenant des données. Il est préférable de sauvegarder les données récupérées sur un autre disque dur, une clé USB, une disquette ou bien un emplacement réseau.

Les opérations de défragmentation des disques diminuent considérablement les possibilités de restauration de fichiers effacés, de par les mouvements de données sur le disque et les possibles recouvrements en résultant.