How to recover NTFS

(Freeware Guide)

Step by step with examples

Contents

1. Overview

The most common causes of NTFS partition issues are:

- Physical damage of critical sectors on a HDD (known as unreadable or 'bad sectors')
- Loss of information due to an electrical failure or power surge
- Accidental deletion of the logical drive/partition
- Accidental formatting of the logical disk/partition
- Accidental deletion or damage of system files
- Damage of the MBR, Partition Table, Volume Boot Sectors by a software virus or malware
- Improper use or execution failures of backup/recovery software tools

When the volume is damaged it usually displays one of the following symptoms:

- Original partition/drive is no longer visible to the Operating System (deleted, damaged, or overwritten)
- Partition/Volume is visible but important files/folders are not visible (drive re-formatted or damaged)

In both cases partition recovery software must analyze the surface of the physical drive for residual logical data and organization clues in order to reconstruct the partition/drive parameters (such as the first sector number, cluster size, file system type, etc.). After a user obtains an access to this virtual drive, he is able to re-create partition (recover partition information) or just to copy lost data to another drive (with use of a file recovery program).

Examples of low level partition damage and recovery procedures

We assume that you have some knowledge of a HDD and the File System's organization to be able to understand the recovery terminology and examples. If not, please visit www.ntfs.com to get knowledge of hard disk organization: http://www.ntfs.com/hard-disk-basics.htm and NTFS basics: http://www.ntfs.com/ntfs_basics.htm

2. NTFS Partition Recovery Concepts

System Boot Process

In most cases, the first indication of a problem with hard drive data is a refusal of the machine to boot properly. For the computer to be able to find startup partition and to start booting, the following conditions must apply:

- Master Boot Record (MBR) or GIUD Partition Table (GPT) exists and is safe
- Partition Table exists and contains at least one Active partition
- Active partition contains all necessary and not damaged system files for the OS launch

If the above is in place, executable code in the MBR selects an active partition and passes control there, so it can start loading the standard files (COMMAND.COM, NTLDR, BOOTMGR ...) depending on the OS and the file system type on that partition. If these files are missing or corrupted it will be impossible for the OS to boot - you understand the situation if you have ever seen the famous "NTLDR is missing ..." error message.

Volume Visibility

A more serious situation exists if your computer will start and cannot see a drive partition*. For the partition to be visible to the Operating System the following conditions must apply:

- - Partition/Drive can be found via Partition Table
- - Partition/Drive/Volume boot sector is safe
- - Volume system areas (MFT, Root) are safe and accessible

If the above conditions are true, the Operating System can read the partition or physical drive parameters and display the drive in the list of the available drives.

If the file system is damaged (Master File Table (MFT) records on NTFS) the drive's content might not be displayed and we might see errors like "MFT is corrupted", or "Drive is invalid" ... If this is the case it is less likely that you will be able to restore your data in full. Do not despair, as there may be some tricks or tips to display some of the residual entries that are still safe, allowing you to recover your data to another location.

Partition Recovery Includes

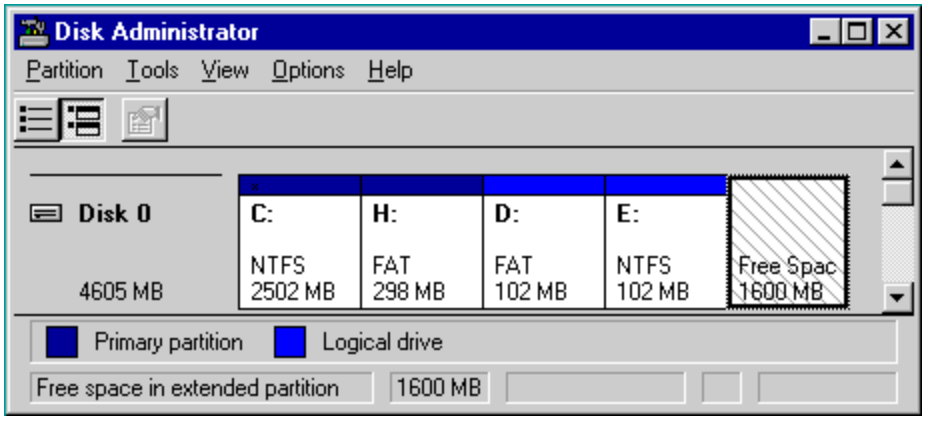
- Physical partition recovery. The goal is to identify the problem and write information to the proper place on the hard drive (to MBR and Boot Sectors) so that the partition becomes visible to the Operating System again. This can be done using manual Disk Editors along with proper guidelines or using partition recovery software, designed specifically for this purpose.
- Virtual partition recovery. The goal is to determine the critical parameters of the deleted/damaged/overwritten partition and render it open to scanning in order to display its content to copy important data to the safe place. This approach can be applied in some cases when physical partition recovery is not possible (for example, partition boot sector is dead and physically unreadable) and is commonly used by file recovery software. This process is almost impossible to implement it manually.

Other Hard Drive Partition Recovery Topics

Let's consider the topics, related to the recovery of partitions in common, not specific to the particular file system. We have the following cases:

- Master Boot Record (MBR) is damaged
- Partition is deleted or Partition Table is damaged
- Partition Boot Sector is damaged
- Missing or Corrupted System Files

As an example we'll use the following disk layout:



Master Boot Record (MBR) is damaged

The Master Boot Record (MBR) will be created when you create the first partition on the hard disk. It is very important data structure on the disk. The Master Boot Record contains the Partition Table for the disk and a small amount of executable code for the boot start. The location is always the first sector on the disk.

The first 446 (0x1BE) bytes are MBR itself, the next 64 bytes are the Partition Table, the last two bytes in the sector are a signature word for the sector and are always 0x55AA.

For our disk layout we have MBR:

```
Physical Sector: Cyl 0, Side 0, Sector 1
000000000   33 C0 8E D0 BC 00 7C FB  50 07 50 1F FC BE 1B 7C   3AZ??.|uP.P.u?.|
000000010   BF 1B 06 50 57 B9 E5 01  F3 A4 CB BE BE 07 B1 04   ?..PW?a.o¤E??.±.
000000020   38 2C 7C 09 75 15 83 C6  10 E2 F5 CD 18 8B 14 8B   8,|.u.??.aoI.‹.‹
000000030   EE 83 C6 10 49 74 16 38  2C 74 F6 BE 10 07 4E AC   i??.It.8,to?..N¬
000000040   3C 00 74 FA BB 07 00 B4  0E CD 10 EB F2 89 46 25   <.tu».·.?.I.eo‰F%
000000050   96 8A 46 04 B4 06 3C 0E  74 11 B4 0B 3C 0C 74 05   –SF.?.<.t.?.<.t.
000000060   3A C4 75 2B 40 C6 46 25  06 75 24 BB AA 55 50 B4   :Au+@?F%.u$»?UP?
000000070   41 CD 13 58 72 16 81 FB  55 AA 75 10 F6 C1 01 74   AI.Xr.?uU?u.oA.t
000000080   0B 8A E0 88 56 24 C7 06  A1 06 EB 1E 88 66 04 BF   .Sa?V$C.?.e.?f.?
000000090   0A 00 B8 01 02 8B DC 33  C9 83 FF 05 7F 03 8B 4E   ..?..‹U3E?y.•.‹N
0000000A0   25 03 4E 02 CD 13 72 29  BE 46 07 81 3E FE 7D 55   %.N.I.r)?F.?>?}U
0000000B0   AA 74 5A 83 EF 05 7F DA  85 F6 75 83 BE 27 07 EB   ?tZ?i.•U…ou??'.e
0000000C0   8A 98 91 52 99 03 46 08  13 56 0A E8 12 00 5A EB   S?‘R™.F..V.e..Ze
0000000D0   D5 4F 74 E4 33 C0 CD 13  EB B8 00 00 00 00 00 00   OOta3AI.e?......
0000000E0   56 33 F6 56 56 52 50 06  53 51 BE 10 00 56 8B F4   V3oVVRP.SQ?..V‹o
0000000F0   50 52 B8 00 42 8A 56 24  CD 13 5A 58 8D 64 10 72   PR?.BSV$I.ZX?d.r
000000100   0A 40 75 01 42 80 C7 02  E2 F7 F8 5E C3 EB 74 49   .@u.B€C.a?o^AetI
000000110   6E 76 61 6C 69 64 20 70  61 72 74 69 74 69 6F 6E   nvalid partition
```

```
000000120   20 74 61 62 6C 65 00 45  72 72 6F 72 20 6C 6F 61    table.Error loa
000000130   64 69 6E 67 20 6F 70 65  72 61 74 69 6E 67 20 73   ding operating s
000000140   79 73 74 65 6D 00 4D 69  73 73 69 6E 67 20 6F 70   ystem.Missing op
000000150   65 72 61 74 69 6E 67 20  73 79 73 74 65 6D 00 00   erating system..
000000160   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ................
000000170   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ................
000000180   00 00 00 8B FC 1E 57 8B  F5 CB 00 00 00 00 00 00   ...‹u.W‹oE......
000000190   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ................
0000001A0   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ................
0000001B0   00 00 00 00 00 00 00 00  A6 34 1F BA 00 00 80 01   ........¦4.?..€.
0000001C0   01 00 07 FE 7F 3E 3F 00  00 00 40 32 4E 00 00 00   ...?•>?...@2N...
0000001D0   41 3F 06 FE 7F 64 7F 32  4E 00 A6 50 09 00 00 00   A?.?•d•2N.¦P....
0000001E0   41 65 0F FE BF 4A 25 83  57 00 66 61 38 00 00 00   Ae.??J%?W.fa8...
0000001F0   00 00 00 00 00 00 00 00  00 00 00 00 00 00 55 AA   ..............U?
```

What will happen if the first sector has been damaged (by virus, for example)?

Lets overwrite the first 16 bytes with zeros.

```
000000000   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ................
000000010   BF 1B 06 50 57 B9 E5 01  F3 A4 CB BE BE 07 B1 04   ?..PW?a.o¤E??.±.
```

When we try to boot after hardware testing procedures, we see just blank screen without any messages. It means the piece of code at the beginning of the MBR could not be executed properly. That's why even error messages could not be displayed. However, if we boot from the other media (load Active@ Boot Disk from USB, for example), we can see the partition, files on it and we are able to perform standard operations like file copy, program execution... It happens because in our example only part of the MBR has been damaged which does not allow the system to boot properly. However, the partition table is safe and we can access our drives when we boot from the operating system installed on the other drive.

What will happen if sector signature (last word 0x55AA) has been removed or damaged?

Let's write zeros to the location of sector signature.

```
Physical Sector: Cyl 0, Side 0, Sector 1
0000001E0   41 65 0F FE BF 4A 25 83  57 00 66 61 38 00 00 00   Ae.??J%?W.fa8...
0000001F0   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ................
```

When we try to boot now, we see an error message like "Operating System not found".

Thus the first thing if computer does not boot is to run Disk Viewer and check the first physical sector on HDD, whether it looks like valid MBR or not:

- check, maybe it's filled up with zeros or any other single character
- check whether error messages (like you can see above "Invalid partition table"...) are present or not
- check whether disk signature (0x55AA) is present or not

The simplest way to repair or re-create MBR is to run Microsoft's standard utility called FDISK with a parameter /MBR, like

```
A:\> FDISK.EXE  /MBR
```

FDISK is a standard utility included in MS-DOS, Windows 95, 98, ME.

If you have Windows NT / 2000 / XP / Vista / 7, you can boot from startup floppy disks USB or CD-ROM, choose repair option during setup, and run Recovery Console. When you are logged on, you can run FIXMBR command to fix MBR, for example:

```
C:\> FIXMBR \Device\HardDisk0
```

You can use Edit Partition Table function of Active@ Partition Manager to fix manually the MBR signature and the partition table.

Also you can use third party MBR recovery software or if you've created MBR backup, restore it from there (Active@ Partition Recovery has such capabilities).

What will happen if the first sector is bad or unreadable?

Most likely we'll get the same black screen, which we got when trying to boot. When you try to read it using Disk Viewer/Editor you should get an error message saying that sector is unreadable. In this case recovery software is unable to help you to bring HDD back to the working condition, i.e. physical partition recovery is not possible. The only thing that can be done is to scan and search for partitions (i.e. perform virtual partition recovery), and in case if something is found - display them and give the user an opportunity to save important data to another location. Third party software, like Active@ File Recovery (www.file-recovery.com), will help you here.

Partition is deleted or Partition Table is damaged

The information about primary partitions and extended partition is contained in the Partition Table, a 64-byte data structure, located in the same sector as the Master Boot Record (cylinder 0, head 0, sector 1). The Partition Table conforms to a standard layout, which is independent of the operating system. The last two bytes in the sector are a signature word for the sector and are always 0x55AA.

For our disk layout we have Partition Table:

```
Physical Sector: Cyl 0, Side 0, Sector 1
0000001B0                                                 80 01   ..............Ђ.
0000001C0   01 00 07 FE 7F 3E 3F  00 00 00 40 32 4E 00 00 00   ...?•>?...@2N...
0000001D0   41 3F 06 FE 7F 64 7F 32  4E 00 A6 50 09 00 00 00   A?.?•d•2N.¦P....
0000001E0   41 65 0F FE BF 4A 25 83  57 00 66 61 38 00 00 00   Ae.??J%?W.fa8...
0000001F0   00 00 00 00 00 00 00 00  00 00 00 00 00 00 55 AA   ..............U?
```

We can see three existing and one empty entries:

- Partition 1, offset 0x01BE (446)
- Partition 2, offset 0x01CE (462)
- Partition 3, offset 0x01DE (478)
- Partition 4 - empty, offset 0x01EE (494)

Each Partition Table entry is 16 bytes long, making a maximum of four entries available. Each partition entry has fields for Boot Indicator (BYTE), Starting Head (BYTE), Starting Sector (6 bits), Starting Cylinder (10 bits), System ID (BYTE), Ending Head (BYTE), Ending Sector (6 bits), Ending Cylinder (10 bits), Relative Sector (DWORD), Total Sectors (DWORD).

Thus the MBR loader can assume the location and size of partitions. MBR loader looks for the "active" partition, i.e. partition that has Boot Indicator equals 0x80 (the first one in our case) and passes control to the partition boot sector for further loading.

Let’s consider the situations which cause computer to hang up while booting or data loss.

1. What will happen if no partition has been set to the Active state (Boot Indicator=0x80)?

Let’s remove Boot Indicator from the first partition:

```
0000001B0                                               00 01  ..............
0000001C0 01 00 07 FE 7F 3E 3F  00 00 00 40 32 4E 00 00 00  ...?•>?...@2N...
```

When we try to boot now, we see an error message like "Operating System not found". It means that the loader cannot determine which partition is system and active to pass control to. Use Mark Partition as Active function of Active@ Partition Manager to specify the main partition to start booting from, and fixing this issue.

2. What will happen if partition has been set to the Active state (Boot Indicator=0x80) but there are no system files on that partition?
(it could happen if we had used for example FDISK and selected not the proper active partition).

Loader will try to boot from there, fails, try to boot again from other devices like floppy, and if fails to boot again, we'll see an error message like "Non-System Disk or Disk Error". You need either to specify another partition Active, or copy system files here.

3. What will happen if partition entry has been deleted?
If it has been deleted, next two partitions will move one line up in the partition table.

```
Physical Sector: Cyl 0, Side 0, Sector 1
0000001B0                                                 80 00   ..............Ђ.
0000001C0   41 3F 06 FE 7F 64 7F 32  4E 00 A6 50 09 00 00 00   A?.?•d•2N.¦P....
0000001D0   41 65 0F FE BF 4A 25 83  57 00 66 61 38 00 00 00   Ae.??J%?W.fa8...
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   ..............U?
```

If we try to boot now, the previous second (FAT) partition becomes the first and the loader will try to boot from it. And if it's not a system partition, we'll get the same error messages. Active@ Partition Recovery software will help you to undelete accidentally deleted partition.

4. What will happen if partition entry has been damaged?
Let's write zeros to the location of the first partition entry.

```
Physical Sector: Cyl 0, Side 0, Sector 1
0000001B0                                                 80 00   ..............Ђ.
0000001C0   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ................
0000001D0   41 3F 06 FE 7F 64 7F 32  4E 00 A6 50 09 00 00 00   A?.?•d•2N.¦P....
0000001E0   41 65 0F FE BF 4A 25 83  57 00 66 61 38 00 00 00   Ae.??J%?W.fa8...
0000001F0   00 00 00 00 00 00 00 00  00 00 00 00 00 00 55 AA   ..............U?
```

If we try to boot now, the MBR loader will try to read and interpret zeros (or other garbage) as partition parameters and we'll get an error message like "Missing Operating System".

Thus, the second step in partition recovery is to run Disk Viewer and to make sure that the proper partition exists in the partition table and has been set as active.

How can recovery software help you in the above-mentioned scenarios?

1. Discover and suggest you to choose the partition to be active (even FDISK does so).
2. Discover and suggest you to choose the partition to be active.
3. Perform a free disk space scan to look for partition boot sector or remaining of the deleted partition information in order to try to reconstruct Partition Table entry for the deleted partition.
4. Perform all disk space scan to look for partition boot sector or remaining of the damaged partition information in order to try to reconstruct Partition Table entry for the damaged partition entry.

Why partition boot sector is so important?

Because, if recovery software finds it, all necessary parameters to reconstruct partition entry in the Partition Table are there (see Partition Boot Sector topic for details).

What would happen if partition entry had been deleted then recreated with other parameters and re-formatted?

In this case, instead of the original partition entry we would have a new one and everything would work fine except that later on we could recall that we had some important data on the original partition. If you've created MBR, Partition Table, Volume Sectors backup (for example, Active@ Partition Recovery can do it) before, you can virtually restore it back and look for your data (in case if it has not been overwritten with new data yet). Some advanced recovery tools also have an ability to scan disk surface and try to reconstruct the previously deleted partition information from the pieces of left information (i.e. perform virtual partition recovery). However it is not guaranteed that you can recover something.

Partition Boot Sector is damaged

The Partition Boot Sector contains information, which the file system uses to access the volume. On personal computers, the Master Boot Record uses the Partition Boot Sector on the system partition to load the operating system kernel files. Partition Boot Sector is the first sector of the Partition.

For our first NTFS partition we have boot sector:

```
Physical Sector: Cyl 0, Side 1, Sector 1
000000000   EB 5B 90 4E 54 46 53 20  20 20 20 00 02 01 00 00   e[?NTFS    .....
000000010   00 00 00 00 00 F8 00 00  3F 00 FF 00 3F 00 00 00   .....o..?.y.?...
000000020   00 00 00 00 80 00 80 00  3F 32 4E 00 00 00 00 00   ....Ђ.Ђ.?2N.....
000000030   5B 43 01 00 00 00 00 00  1F 19 27 00 00 00 00 00   [C........'.....
000000040   02 00 00 00 08 00 00 00  10 EC 46 C4 00 47 C4 0C   .........iFA.GA.
000000050   00 00 00 00 00 00 00 00  00 00 00 00 00 FA 33 C0   ..............u3A
000000060   8E D0 BC 00 7C FB B8 C0  07 8E D8 C7 06 54 00 00   Z??.|u?A.ZOC.T..
000000070   00 C7 06 56 00 00 00 C7  06 5B 00 10 00 B8 00 0D   .C.V...C.[...?..
000000080   8E C0 2B DB E8 07 00 68  00 0D 68 66 02 CB 50 53   ZA+Ue..h..hf.EPS
000000090   51 52 06 66 A1 54 00 66  03 06 1C 00 66 33 D2 66   QR.f?T.f....f3Of
0000000A0   0F B7 0E 18 00 66 F7 F1  FE C2 88 16 5A 00 66 8B   .·...f?n?A?.Z.f‹
0000000B0   D0 66 C1 EA 10 F7 36 1A  00 88 16 25 00 A3 58 00   ?fAe.?6..?.%.?X.
0000000C0   A1 18 00 2A 06 5A 00 40  3B 06 5B 00 76 03 A1 5B   ?..*.Z.@;.[.v.?[
0000000D0   00 50 B4 02 8B 16 58 00  B1 06 D2 E6 0A 36 5A 00   .P?.‹.X.±.O?.6Z.
0000000E0   8B CA 86 E9 8A 36 25 00  B2 80 CD 13 58 72 2A 01   ‹E†eS6%.?ЂI.Xr*.
0000000F0   06 54 00 83 16 56 00 00  29 06 5B 00 76 0B C1 E0   .T.?.V..).[.v.Aa
000000100   05 8C C2 03 D0 8E C2 EB  8A 07 5A 59 5B 58 C3 BE   .?A.?ZAeS.ZY[XA?
000000110   59 01 EB 08 BE E3 01 EB  03 BE 39 01 E8 09 00 BE   Y.e.?a.e.?9.e..?
000000120   AD 01 E8 03 00 FB EB FE  AC 3C 00 74 09 B4 0E BB    .e..ue?¬<.t.?.»
000000130   07 00 CD 10 EB F2 C3 1D  00 41 20 64 69 73 6B 20   ..I.eoA..A disk
000000140   72 65 61 64 20 65 72 72  6F 72 20 6F 63 63 75 72   read error occur
000000150   72 65 64 2E 0D 0A 00 29  00 41 20 6B 65 72 6E 65   red....).A kerne
000000160   6C 20 66 69 6C 65 20 69  73 20 6D 69 73 73 69 6E   l file is missin
000000170   67 20 66 72 6F 6D 20 74  68 65 20 64 69 73 6B 2E   g from the disk.
000000180   0D 0A 00 25 00 41 20 6B  65 72 6E 65 6C 20 66 69   ...%.A kernel fi
000000190   6C 65 20 69 73 20 74 6F  6F 20 64 69 73 63 6F 6E   le is too discon
0000001A0   74 69 67 75 6F 75 73 2E  0D 0A 00 33 00 49 6E 73   tiguous....3.Ins
0000001B0   65 72 74 20 61 20 73 79  73 74 65 6D 20 64 69 73   ert a system dis
0000001C0   6B 65 74 74 65 20 61 6E  64 20 72 65 73 74 61 72   kette and restar
0000001D0   74 0D 0A 74 68 65 20 73  79 73 74 65 6D 2E 0D 0A   t..the system...
0000001E0   00 17 00 5C 4E 54 4C 44  52 20 69 73 20 63 6F 6D    ...\NTLDR is com
0000001F0   70 72 65 73 73 65 64 2E  0D 0A 00 00 00 00 55 AA   pressed.......U?

Offset      0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
```

The printout is formatted in three sections:

- Bytes 0x00– 0x0A are the jump instruction and the OEM ID (shown in bold print).

- Bytes 0x0B–0x53 are the BIOS Parameter Block (BPB) and the extended BPB. This block contains such essential parameters as Bytes Per Sector (WORD, offset 0x0B), Sectors Per Cluster (BYTE, offset 0x0D), Media Descriptor (BYTE, offset 0x15), Sectors Per Track (WORD, offset 0x18), Number of Heads (WORD, offset 0x1A), Hidden Sectors (DWORD, offset 0x1C), Total Sectors (LONGLONG, offset 0x28), etc...
- The remaining code is the bootstrap code (that is necessary for the proper system boot) and the end of sector marker (shown in bold print).

This sector is so important on NTFS, for example, duplicate of the boot sector is located on the disk.

Boot Sector for FAT looks different, however its BPB contains parameters similar to the above mentioned. There is no extra copy of this sector stored anywhere, so recovery on FAT is as half as less successful than on NTFS.

What will happen if Partition Boot Sector is damaged or bad/unreadable?

Let's fill up with zeros several lines of Partition Boot Sector:

```
000000000   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ................
000000010   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ................
000000020   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ................
000000030   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ................
000000040   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ................
000000050   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ................
000000060   8E D0 BC 00 7C FB B8 C0  07 8E D8 C7 06 54 00 00   Z??.|u?A.ZOC.T..
```

If we try to boot, we'll see "Non System Disk" or "Disk Error..". After we fail to load from it and from floppy, partition becomes unbootable.

Because a normally functioning system relies on the boot sector to access a volume, it is highly recommended that you run disk-scanning tools such as CHKDSK regularly, as well as back up all of your data files to protect against data loss in case you lose access to the volume.

Tools like Active@ Partition Recovery and Active@ UNERASER allow you to create backup of MBR, Partition Table and Volume Boot Sectors so that if for some reason it fails to boot, you can always restore your partition information and have an access to files/folders on that partition.

What to do if this sector is damaged?

- If we do have backup of the whole disk or MBR/Boot Sectors we can try to restore it from there.
- If we do not have backup, in case of NTFS we could try to locate a duplicate of Partition Boot Sector and get information from there.
- If duplicate boot sector is not found, only virtual partition recovery might be possible if we can determine critical partition parameters such as Sectors per Cluster, etc..

How can we fix NTFS boot sector using standard Windows NT/2000/XP/Vista/7 tools?

On NTFS copy of boot sector is stored at the middle or at the end of the Volume.

You can boot from startup floppy disks or CD-ROM, choose repair option during setup, and run Recovery Console. When you are logged on, you can run FIXBOOT command to try to fix boot sector, for example:

```
A:\> FIXBOOT C:
```

How can recovery software help you in this situation?

- It can backup MBR, Partition Table and Boot Sectors and restore them in case of damage
- It can try to find out duplicate boot sector on the drive and re-create the original one or perform virtual data recovery based on found partition parameters
- Some advanced techniques allow assuming drive parameters even if duplicate boot sector is not found (i.e. perform virtual partition recovery) and give the user virtual access to the data on the drive to be able to copy them to the safer location.

Missing or Corrupted System Files

For Operating System to boot properly, system files required to be safe.

- Windows Vista, Windows 2008 Server, Windows 7 - BOOTMGR and Boot folder located at the root folder of the bootable volume. Boot folder should contain BCD file containing bootable configuration.
- Windows NT / 2000 / XP / Windows 2003 Server NTLDR, ntdetect.com, boot.ini, located at the root folder of the bootable volume, Registry files (i.e., SAM, SECURITY, SYSTEM and SOFTWARE), etc.
- Windows 95 / 98 / ME - msdos.sys, config.sys, autoexec.bat, system.ini,at the root folder, system.dat, user.dat, etc.

If these files have been deleted, corrupted, damaged by virus, Windows will be unable to boot. You'll see an error message "NTLDR is missing" or "BOOTMGR is missing".

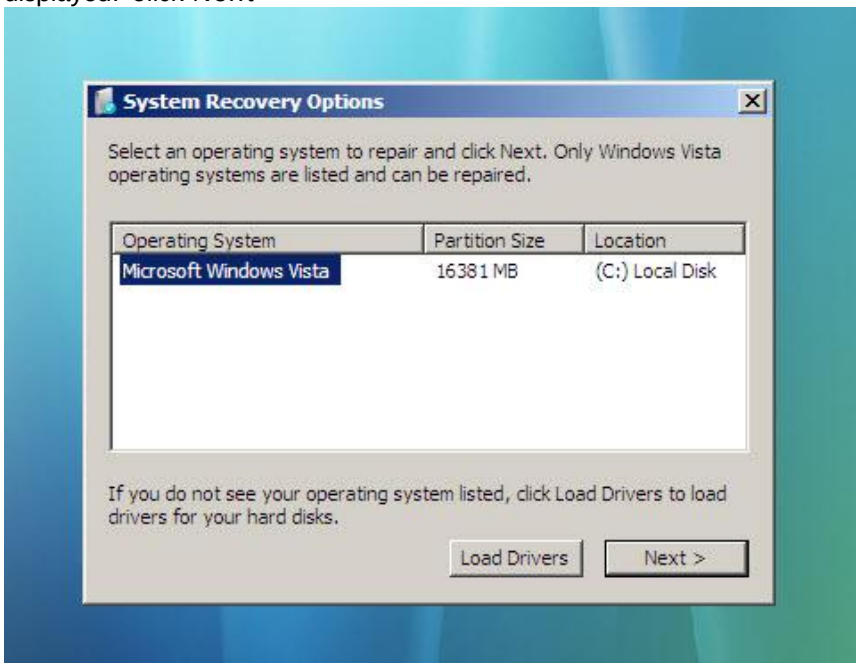
Once it is determined that the operating system won't start, the next step in the recovery process is to check the existence and safety of these vital system files.

Recovery in Windows Vista / Server 2008 / Windows 7

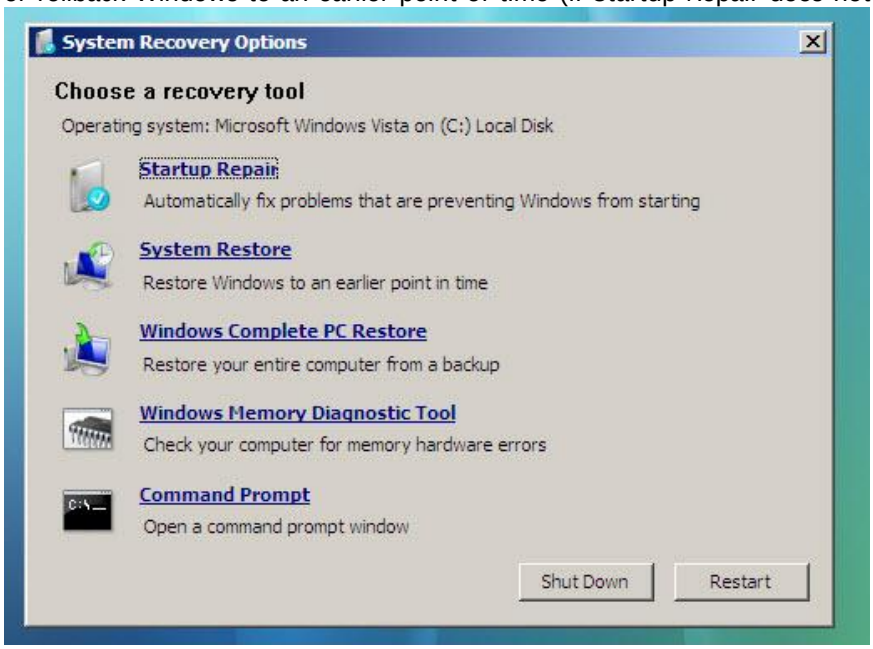
To do this in Windows Vista / 2008 / Windows 7:

- Boot the system from the installation DVD-ROM (or use a system recovery partition)
- Select a language and other settings (if needed) and click Next
- Do NOT click Install Now button. Click Repair your computer

- Select a drive where Windows was installed. You can locate drivers if proper drive is not displayed. Click Next



- Click Startup Repair. At this step you can restore full Windows system from a backup, or rollback Windows to an earlier point of time (if Startup Repair does not help)



- Wait until Windows repairs itself and reboot a machine

Recovery in Windows NT / 2000 / XP / 2003

To do this in Windows NT-based systems use the Emergency Repair Process, Recovery Console or third party recovery software.

Emergency Repair Process

To proceed with Emergency Repair Process, you need an Emergency Repair Disk (ERD). It is recommended that you create an ERD immediately after you install and customize Windows. To create one now, use the "Backup" utility by clicking Start > Programs > Accessories > System Tools. Follow the on-screen directions.

You can use the ERD to repair a damaged boot sector, damaged MBR, repair or replace missing or damaged NT Loader (NTLDR) and ntdetect.com files.

If you do not have an ERD, the emergency repair process can attempt to locate your Windows installation and start repairing your system, but it may not be able to do so.

To run the process, boot from the Windows installation floppy disks or CD, and choose the Repair option when the system suggests you to proceed with installation or repairing. Press R to run the Emergency Repair Process and choose Fast or Manual Repair option. Fast Repair is recommended for most users, Manual Repair is to be used by Administrators and advanced users only.

If the emergency repair process is successful, your computer will automatically restart and you should have a working system.

Recovery Console

Recovery Console is a command line utility similar to MS-DOS command line. With it you can list and display folder content, copy, delete, replace files, format drives and perform many other administrative tasks.

To run Recovery Console, boot from Windows installation disks or CD and choose the Repair option. When the system suggests that you proceed with installation or repairing and then press C to run Recovery Console. You will see a logon screen with Administrator's password. After you are logged on - you can display the drive's contents, check the existence and safety of critical files and, if it appropriate, copy files back to the root location if they have been accidentally deleted.

Recovery Software

Third party recovery software in most cases does not allow you to deal with system files due to the risk of further damage to the system. However you can use a recovery utility to check for the existence and safety of these files. See Recommended Software section below.

Recovery in Windows 95 / 98 / ME

To do this in Windows 95 / 98 / ME - boot the system in Command Prompt Mode, or from a bootable floppy, USB or CD-ROM. Check for system files with the DOS command "dir". If they are missing - copy them back from installation media.

3. NTFS File Recovery Concepts

File recovery process can be briefly described as drive or folder scanning to find deleted entries in Master File Table (MFT) then for the particular deleted entry, defining clusters chain to be recovered and then copying contents of these clusters to the newly created file.

Different file systems maintain their own specific logical data structures, however basically each file system:

- Has a list or catalog of file entries, so we can iterate through this list and entries, marked as deleted
- Keeps for each entry a list of data clusters, so we can try to find out set of clusters composing the file

After finding out the proper file entry and assembling set of clusters, composing the file, read and copy these clusters to another location.

Step by Step with examples:

1. Scan Disk for deleted entries
2. Defining clusters chain for the deleted entry
3. Clusters chain recovery

However, not every deleted file can be recovered, there are some assumptions, for sure:

- First, we assume that the file entry still exists (not overwritten with other data). The less the files have been created on the drive where the deleted file was resided, the more chances that space for the deleted file entry has not been used for other entries.
- Second, we assume that the file entry is more or less safe to point to the proper place where file clusters are located. In some cases (it has been noticed in Windows XP, on large FAT32 volumes) operating system damages file entries right after deletion so that the first data cluster becomes invalid and further entry restoration is not possible.
- Third, we assume that the file data clusters are safe (not overwritten with other data). The less the write operations have been performed on the drive where deleted file was resided, the more chances that the space occupied by data clusters of the deleted file has not been used for other data storage.

As general advices after data loss:

1. DO NOT WRITE ANYTHING ONTO THE DRIVE CONTAINING YOUR IMPORTANT DATA THAT YOU HAVE JUST DELETED ACCIDENTALLY! Even file recovery software installation could spoil your sensitive data. If the data is really important to you and you do not have another logical drive to install software to, take the whole hard drive out of the computer and plug it into another computer where data recovery software has been already installed or use recovery

software that does not require installation, for example recovery software which is capable to run from bootable floppy.

2. DO NOT TRY TO SAVE ONTO THE SAME DRIVE DATA THAT YOU FOUND AND TRYING TO RECOVER! When saving recovered data onto the same drive where sensitive data is located, you can intrude in process of recovering by overwriting FAT/MFT records for this and other deleted entries. It's better to save data onto another logical, removable, network or floppy drive.

Disk Scan for deleted entries

Disk Scan is a process of low-level enumeration of all entries in Master File Table (MFT) on NTFS, NTFS5. The goal is to find and display deleted entries.

In spite of different file/folder entry structure for the different file systems, all of them contain basic file attributes like name, size, creation and modification date/time, file attributes, existing/deleted status, etc...

Given that a drive contains root file table and any file table (MFT, root folder of the drive, regular folder, or even deleted folder) has location, size and predefined structure, we can scan it from the beginning to the end checking each entry, if it's deleted or not and then display information for all found deleted entries.

Deleted entries are marked differently depending on the file system. On NTFS deleted entry has a special attribute in file header that points whether the file has been deleted or not.

You will need a freeware tool like Active@ Disk Editor to do such a research manually, or a tool like Active@ File Recovery to detect deleted entries automatically.

Example of scanning folder on NTFS5:

For our drive we have input parameters:

- Total Sectors 610406
- Cluster size 512 bytes
- One Sector per Cluster
- MFT starts from offset 0x4000, non-fragmented
- MFT record size 1024 bytes
- MFT Size 1968 records

Thus we can iterate through all 1968 MFT records, starting from the absolute offset 0x4000 on the volume looking for the deleted entries. We are interested in MFT entry 57 having offset 0x4000 + 57 * 1024 = 74752 = 0x12400 because it contains our recently deleted file "My Presentation.ppt"

Below MFT record number 57 is displayed:

```
Offset      0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F

00012400   46 49 4C 45 2A 00 03 00  9C 74 21 03 00 00 00 00   FILE*...?t!.....
00012410   47 00 02 00 30 00 00 00  D8 01 00 00 00 04 00 00   G...0...O.......
00012420   00 00 00 00 00 00 00 00  05 00 03 00 00 00 00 00   ................
00012430   10 00 00 00 60 00 00 00  00 00 00 00 00 00 00 00   ....`...........
00012440   48 00 00 00 18 00 00 00  20 53 DD A3 18 F1 C1 01   H....... SY?.nA.
00012450   00 30 2B D8 48 E9 C0 01  C0 BF 20 A0 18 F1 C1 01   .0+OHeA.A?  .nA.
00012460   20 53 DD A3 18 F1 C1 01  20 00 00 00 00 00 00 00    SY?.nA. .......
00012470   00 00 00 00 00 00 00 00  00 00 00 00 02 01 00 00   ................
00012480   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ................
00012490   30 00 00 00 78 00 00 00  00 00 00 00 00 00 03 00   0...x...........
000124A0   5A 00 00 00 18 00 01 00  05 00 00 00 00 00 05 00   Z...............
000124B0   20 53 DD A3 18 F1 C1 01  20 53 DD A3 18 F1 C1 01    SY?.nA. SY?.nA.
000124C0   20 53 DD A3 18 F1 C1 01  20 53 DD A3 18 F1 C1 01    SY?.nA. SY?.nA.
000124D0   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ................
000124E0   20 00 00 00 00 00 00 00  0C 02 4D 00 59 00 50 00    .........M.Y.P.
000124F0   52 00 45 00 53 00 7E 00  31 00 2E 00 50 00 50 00   R.E.S.~.1...P.P.
00012500   54 00 69 00 6F 00 6E 00  30 00 00 00 80 00 00 00   T.i.o.n.0...€...
00012510   00 00 00 00 00 00 02 00  68 00 00 00 18 00 01 00   ........h.......
00012520   05 00 00 00 00 00 05 00  20 53 DD A3 18 F1 C1 01   ........ SY?.nA.
00012530   20 53 DD A3 18 F1 C1 01  20 53 DD A3 18 F1 C1 01    SY?.nA. SY?.nA.
00012540   20 53 DD A3 18 F1 C1 01  00 00 00 00 00 00 00 00    SY?.nA.........
00012550   00 00 00 00 00 00 00 00  20 00 00 00 00 00 00 00   ........ .......
00012560   13 01 4D 00 79 00 20 00  50 00 72 00 65 00 73 00   ..M.y. .P.r.e.s.
00012570   65 00 6E 00 74 00 61 00  74 00 69 00 6F 00 6E 00   e.n.t.a.t.i.o.n.
00012580   2E 00 70 00 70 00 74 00  80 00 00 00 48 00 00 00   ..p.p.t.€...H...
00012590   01 00 00 00 00 00 04 00  00 00 00 00 00 00 00 00   ................
000125A0   6D 00 00 00 00 00 00 00  40 00 00 00 00 00 00 00   m.......@.......
000125B0   00 DC 00 00 00 00 00 00  00 DC 00 00 00 00 00 00   .U.......U......
000125C0   00 DC 00 00 00 00 00 00  31 6E EB C4 04 00 00 00   .U......1neA....
000125D0   FF FF FF FF 82 79 47 11  00 00 00 00 00 00 00 00   yyyy,yG.........
000125E0   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ................
000125F0   00 00 00 00 00 00 00 00  00 00 00 00 00 00 03 00   ................
................
00012600   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ................
```

MFT Record has pre-defined structure. It has a set of attributes defining any file of folder parameters.

MFT Record begins with standard File Record Header (first bold section, offset 0x00):

- "FILE" identifier (4 bytes)
- Offset to update sequence (2 bytes)
- Size of update sequence (2 bytes)
- $LogFile Sequence Number (LSN) (8 bytes)
- Sequence Number (2 bytes)
- Reference Count (2 bytes)
- Offset to Update Sequence Array (2 bytes)
- Flags (2 bytes)
- Real size of the FILE record (4 bytes)
- Allocated size of the FILE record (4 bytes)
- File reference to the base FILE record (8 bytes)
- Next Attribute Id (2 bytes)

The most important information for us in this block is a file state: deleted or in-use. If Flags(in red color) field has bit 1 set, it means that file is in-use. In our example it is zero, i.e. file is deleted.

Starting from 0x48, we have Standard Information Attribute (second bold section):

- File Creation Time (8 bytes)
- File Last Modification Time (8 bytes)
- File Last Modification Time for File Record (8 bytes)
- File Access Time for File Record (8 bytes)
- DOS File Permissions (4 bytes) 0x20 in our case Archive Attribute

Following standard attribute header, we have File Name Attribute belonging to DOS name space, short file names, (third bold section, offset 0xA8) and again following standard attribute header, we have File Name Attribute belonging to Win32 name space, long file names, (third bold section, offset 0x120):

- File Reference to the Parent Directory (8 bytes)
- File Modification Times (32 bytes)
- Allocated Size of the File (8 bytes)
- Real Size of the File (8 bytes)
- Flags (8 bytes)
- Length of File Name (1 byte)
- File Name Space (1 byte)
- File Name (Length of File Name * 2 bytes)

In our case from this section we can extract file name, "My Presentation.ppt", File Creation and Modification times, and Parent Directory Record number.

Starting from offset 0x188, there is a non-resident Data attribute (green section).

- Attribute Type (4 bytes) (e.g. 0x80)
- Length including header (4 bytes)
- Non-resident flag (1 byte)
- Name length (1 byte)
- Offset to the Name (2 bytes)
- Flags (2 bytes)
- Attribute Id (2 bytes)
- Starting VCN (8 bytes)
- Last VCN (8 bytes)
- Offset to the Data Runs (2 bytes)
- Compression Unit Size (2 bytes)
- Padding (4 bytes)
- Allocated size of the attribute (8 bytes)
- Real size of the attribute (8 bytes)
- Initialized data size of the stream (8 bytes)
- Data Runs ...

In this section we are interested in Compression Unit size (zero in our case means non-compressed), Allocated and Real size of attribute that is equal to our file size (0xDC00 = 56320 bytes), and Data Runs (see the next chapter).

Defining clusters chain for the deleted entry

To define clusters chain we need to scan drive, going through one by one all file (NTFS) clusters belonging (presumably) to the file until we reach the file size equals to the total size of the selected clusters. If the file is fragmented, clusters chain will be composed of several extents in case of NTFS.

Location of these clusters can vary depending on file system. On NTFS each file has _DATA_ attribute that describes "data runs". Disassembling data runs to "extents" for each extent we have start cluster offset and number of clusters in extent, so enumerating extents, we can compose file's cluster chain.

You can try to define clusters chain manually, using low-level disk editors, like freeware Active@ Disk Editor, however it's much simpler to use data recovery tools, like Active@ File Recovery.

Example of defining clusters chain on NTFS

When recovering on NTFS part of DATA attribute called Data Runs give us location about file clusters. In most cases DATA attribute is stored inside MFT record, so if we found MFT record for the deleted file, most likely we'll be able to determine cluster's chain.

In example below DATA attribute is marked with a green color. Data Runs inside, marked as Bold.

```
Offset      0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F

00012580   2E 00 70 00 70 00 74 00  80 00 00 00 48 00 00 00   ..p.p.t.Ђ...H...
00012590   01 00 00 00 00 00 04 00  00 00 00 00 00 00 00 00   ................
000125A0   6D 00 00 00 00 00 00 00  40 00 00 00 00 00 00 00   m.......@.......
000125B0   00 DC 00 00 00 00 00 00  00 DC 00 00 00 00 00 00   .U.......U......
000125C0   00 DC 00 00 00 00 00 00  31 6E EB C4 04 00 00 00   .U......1neA....
000125D0   FF FF FF FF 82 79 47 11  00 00 00 00 00 00 00 00   yyyy,yG.........
```

Data Runs need to be decrypted. First byte (0x31) shows how many bytes are allocated for the length of the run (0x1 in our case) and for the first cluster offset (0x3 in our case). Next, we take one byte (0x6E) that points to the length of the run. Next, we pick up 3 bytes pointing to the start cluster offset (0xEBC404). Changing bytes order we get first cluster of the file 312555 (equals 0x04C4EB). Starting from this cluster we need to pick up 110 clusters (equals 0x6E). Next byte (0x00) tells us that no more data runs exist. Our file is not fragmented, so we have the only one data run.

Lets check, isn't there enough information about the file data?
Cluster size is 512 bytes.
We have 110 clusters, 110*512 = 56320 bytes
Our file size was defined as 56320 bytes, so we have enough information now to recover the file clusters.

Clusters chain recovery for the deleted entry

After clusters chain is defined, automatically or manually, the only task left is to read and save contents of the defined clusters to another place verifying their contents.

We have a chain of clusters; we can calculate each cluster offset from the beginning of the drive, using standard formulas. After that we copy amount of data equals to the cluster size, starting from the calculated offset into the newly created file. For the last one we copy not all cluster, but reminder from the file size minus number of copied clusters multiplied by cluster size.

Formulas for calculating cluster offset could vary depending on file system.

To calculate, for example, offset of the cluster for FAT we need to know:

- Boot sector size
- Number of FAT supported copies
- Size of one copy of FAT
- Size of main root folder
- Number of sectors per cluster
- Number of bytes per sector

On the NTFS, we have linear space so we can calculate cluster offset simply as cluster number multiplied by cluster size.

Example of recovery clusters chain on NTFS

In our example we just need to pick up 110 clusters starting from the cluster 312555.
Cluster size is 512 byte, so the offset of the first cluster would be 512 * 312555 = 160028160 = 0x0989D600

```
Offset      0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F

0989D600   D0 CF 11 E0 A1 B1 1A E1  00 00 00 00 00 00 00 00   РП.аЎ±.б........
0989D610   00 00 00 00 00 00 00 00  3E 00 03 00 FE FF 09 00   ........>...юя..
0989D620   06 00 00 00 00 00 00 00  00 00 00 00 01 00 00 00   ................
0989D630   69 00 00 00 00 00 00 00  00 10 00 00 6B 00 00 00   i...........k...
0989D640   01 00 00 00 FE FF FF FF  00 00 00 00 6A 00 00 00   ....юяяя....j...
0989D650   FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF   яяяяяяяяяяяяяяя
```

Here is our data. What's left to do is just reading from this point 110 clusters (56320 bytes) and then copy them to another location. Data recovery is complete now.

4. Recommended Software

Active@ Disk Editor (http://www.ntfs.com/) – freeware software for viewing, inspecting and editing content of raw disk sectors on USB and HDD disks, Floppy and CD/DVD/Blu-ray media.

Active@ Partition Manager (http://www.ntfs.com/) – freeware software that helps you create, delete, format, change properties and name partitions on your computer.

Active@ Partition Recovery (http://www.partition-recovery.com) –software tool for scanning disks and detecting deleted or severely damaged volumes, and for recovering deleted or damaged NTFS partitions.

Active@ File Recovery (http://www.file-recovery.com/) – software utility for scanning disks and detecting deleted or damaged volumes and files, and for recovering deleted or otherwise lost files on NTFS.

5. Recommended Reading

Recovering NTFS boot sector on NTFS partitions (Q153973)
http://support.microsoft.com/default.aspx?scid=kb;EN-US;q153973

Description of the Windows XP Recovery Console for advanced users (Q314058)
http://support.microsoft.com/kb/314058/EN-US/

How to Recover From a Corrupt NTFS Boot Sector (Q121517)
http://support.microsoft.com/default.aspx?scid=kb;en-us;Q121517

Windows XP Repair Overview
http://www.microsoft.com/resources/documentation/windows/xp/all/proddocs/en-us/options_to_use_when_a_system_will_not_start.mspx?mfr=true

Description of the Windows XP Recovery Console (Q314058)
http://support.microsoft.com/kb/314058/EN-US/

Disk organization, file systems and recovery concepts
http://www.ntfs.com/

6. Glossary of Terms

compressed cluster

When you set a file or folder property to compress data, the file or folder uses less disk space. While the size of the file is smaller, it must use a whole cluster in order to exist on the hard drive. As a result, compressed clusters contain "file slack space". This space may contain residual confidential data from the file that previously occupied this space. KillDisk can wipe out the residual data without touching the existing data.

cluster

A logical group of disk sectors, managed by the operating system, for storing files. Each cluster is assigned a unique number when it is used. The operating system keeps track of clusters in the hard disk's root records or MFT records. (See lost cluster)

free cluster

A cluster that is not occupied by a file. This space may contain residual confidential data from the file that previously occupied this space. KillDisk can wipe out the residual data.

file slack space

The smallest file (and even an empty folder) takes up an entire cluster. A 10-byte file will take up 2,048 bytes if that is the cluster size. File slack space is the unused portion of a cluster. This space may contain residual confidential data from the file that previously occupied this space. KillDisk can wipe out the residual data without touching the existing data.

deleted boot records

All disks start with a boot sector. In a damaged disk, if the location of the boot records is known, the partition table can be reconstructed. The boot record contains a file system identifier.

ISO

An International Organization for Standardization ISO-9660 file system is a standard CD-ROM file system that allows you to read the same CD-ROM whether you're on a PC, Mac, or other major computer platform. Disk images of ISO-9660 file systems (ISO images) are a common way to electronically transfer the contents of CD-ROMs. They often have the filename extension .ISO (though not necessarily), and are commonly referred to as "ISOs".

lost cluster

A cluster that has an assigned number in the file allocation table, even though it is not assigned to any file. You can free up disk space by reassigning lost clusters. In DOS and Windows, you can find lost clusters with the ScanDisk utility.

MFT records

Master File Table. A file that contains the records of every other file and directory in an NTFS-formatted hard disk drive. The operating system needs this information to access the files.

root records

File Allocation Table. A file that contains the records of every other file and directory in a FAT-formatted hard disk drive. The operating system needs this information to access the files. There are FAT32, FAT16 and FAT versions.

sector

The smallest unit that can be accessed on a disk. Tracks are concentric circles around the disk and the sectors are segments within each circle.

unallocated space

Space on a hard disk where no partition exists. A partition may have been deleted or damaged or a partition may not have been created.

Windows system records

The Windows registry keeps track of almost everything that happens in windows. This enhances performance of the computer when doing repetitive tasks. Over time, these records can take up a lot of space.