



DATA RECOVERY E-BOOK V1.5

FOREWORD

License:

Copyright (C) 2006 CHENGDU YIWO Tech Development Co., Ltd (YIWO Tech Ltd, for short). All rights reserved.

Data Recovery E-book is a Free book. You may use Data Recovery E-book for free as well as copy, distribute and transmit Data Recovery E-book. And it is only for your personal, non-commercial use.

The core of information age is the information technology, while the core of the information technology consists in the information process and storage. Along with the rapid development of the information and the popularization of the personal computer, people find information more and more useful and need it ever more than the past. Owing to the massive data, there is a huge challenge to the data storage technology. So how to save so many documents and how to visit document as fast as possible become the key point. We know we need storage devices to save data, while there are so many kinds of storage devices and modes to save data. What's more, when saving the data and information, it is more important to ensure the storage security as well as the accuracy, usability, reliability of data. Often, what is invaluable is that invisible data.

CONTENTS

I .Elementary knowledge of data recovery.....	4
1.Connotation of data.....	4
2.The essence of data recovery	5
3.The scope of data recovery	5
4.The principle of data recovery	6
II .Data loss.....	6
1.Software reason.....	7
2.Hardware reason	7
III.Data Protecting Technologies.....	8
1.SMART Technology	8
2.SPS.....	9
3.DFT	9
4.Floppy disk array technology.....	9
5.SAN.....	10
6.NAS.....	10
7.Backup	10
IV.Elementary knowledge of hard disk.....	10
1.History of hard disk development	10
2.Main technical specification and parameter of hard disk.....	12
3.Physical structure of hard disk	14
4.Logical organization of hard disk.....	15
5.Connection synopsis of hard disk	19
V .Hard disk data organization	19
1.Primary formatting of hard disk.....	19
2.Advanced formatting of hard disk.....	21
3.Data storage region of hard disk	23
VI.Common Cases of Partition Recovery	26
1.MBR Recovery	26
2.Recovery of Partition	29
3.Partition Table Doctor	29
4.The FAT table recovery	37
VII. FAT16 file system disk.....	39
1.File management in root directory of FAT16	39
2. FAT16 sub-directory management	42
3. File deletion in FAT16.....	43
4. Contents deletion in FAT16.....	46
5. Fast high level format in FAT16 partition	46
6.Full advanced format in FAT16 partition	50
7.Searching files in FAT16 partition	51
VIII. Management of FAT32 file system	56
1.Root directory management in FAT32 partition.....	56

2. Management of sub-directory in FAT32	61
3. File deletion in FAT32.....	63
4. Sub-directory deletion in FAT32	66
5.High-level format in FAT32	68
IX, Management of NTFS file system	72
1. The NTFS file system introduction	72
2.NTFS file system terminology	72
3. Data constructions of NTFS file system	73
4.Drivers of NTFS.....	73
5.DBR of NTFS file system	74
6. Metadata of NTFS file system	77
7. Files and folders of NTFS partition	81
8. Analysis of important metafile in NTFS file system - \$MFT file analyzes	82
9. NTFS index record and contents.....	88
X. Dynamic disk introduction.....	91
1. Raid background	91
2. Realization of RAID	92
3. Transform basic disk into dynamic disk.....	93
4. Some terms.....	95
5.Characteristics of Dynamic disk	98
6. Dynamic disk management	99
7. RAID 0 -- Striped Disk Array	100
XI.Case Study.....	105
1. Introduction of Data Recovery Wizard 3.0	105
2. Matters needs attention before recovery	105
3. Deletedrecovery	106
4.FormatRecovery	112
5.Recover encrypt/compressed files in NTFS.....	113
6.RAW RECOVERY	114
7. Recovery when parts of partitions are lost:	115
8. Data recovery in dynamic volume	119
9. Data recovery on inaccessible partition.	119
10. File recovery on RAW partition	125
11. Data recovery when all the partitions are lost	125
12. Data recovery when GHOST Image restore failed.....	129
13. After Partition Magic size revision/ combination/ division of partitions fails, how to recover the lost data?.....	129
14. When using Data Recovery Wizard 3.0 to recover files, there is some strange sound in HD. How to handle it?	129
15. HD cannot be detected in BIOS, how to recover data by Data Recovery Wizard 3.0	129
16. There is not enough space in hard disk to save the recovered files, nor there is removable storage device, how to handle it?	129
17. How to recover data in other storage devices (eg: floppy disk, flash drive, removable disk etc)?.....	130

18. To recover image file, how can I know it can really recover the data before I buy Data Recovery Wizard 3.0 ?	132
19. There are so many files recovered, how can I find the files I want fast?	133
20. I have recovered some files, but I cannot rightly open them.....	135
21. In what occasion I cannot rightly recover data?.....	135
XII.Introduction of data security software.....	136

Elementary

I .Elementary knowledge of data recovery

1.Connotation of data

Connotation of data is comprehensive, it includes not only multi-media files such as data documents, images, voices that stored in file system or data base, but also hardware information,

network addresses and network services, which are used to deposit and manage those information.

2.The essence of data recovery

Data recovery means retrieving lost, deleted, unusable or inaccessible data that lost for various reasons.

Data recovery not only restores lost files but also recovers corrupted data.

On the basis of different lost reason, we can adopt different data recovery methods. There are software and hardware reasons that cause data loss, while we can recover data by software and hardware ways.

Being different from prevention and backup, data recovery is the remedial measure. The best way to insure the security of your data is prevention and backup regularly. To operate and use your data according to the normative steps, you can reduce the danger of data loss to the lowest.

3.The scope of data recovery

There are so many forms and phenomenon on data problem, we can divide the objects or scope of data recovery according to different symptoms.

System problem

The main symptom is that you cannot enter the system or the system is abnormal or computer closes down. There are complex reasons for this, thus we need adopt different processing methods. Reasons for this symptom may be the key file of system is lost or corrupted, there is some bad track on hard disk, the hard disk is damaged, MBR or DBR is lost, or the CMOS setting is incorrect and so on.

Bad track of hard disk

There are logic and physical bad track. Logic bad track is mainly caused by incorrect operation, and it can be restored by software. While physical bad track is caused by physical damage, which is real damage, we can restore it by changing the partition or sector. When there is physical bad track, you'd better backup your data for fear that the data can not be used any more because of the bad track.

Partition problem

If partition can not be identified and accessed, or partition is identified as unformatted, partition recovery tools such as Partition Table Doctor can be used to recover data.

Files loss

If files are lost because of deletion, format or Ghost clone error, files restoring tools such as Data Recovery Wizard can be used to recover data.

Password loss

If files, system password, database or account is lost, some special decryption tools that correspond to certain data form such as Word, Winzip can be used.

Files repair

For some reasons, some files can not be accessed or used, or the contents are full of troubled characters, the contents are changed so as they can not be read. In this condition, some special files restoring tools can be tried to restore the files.

4.The principle of data recovery

Data recovery is a process of finding and recovering data, in which there may be some risk, for no all situations can be anticipated or prearranged. It means maybe there will be some unexpected things happen. So you need reduce the danger in data recovery to the lowest:

Backup all the data in your hard disk

Prevent the equipment from being damaged again

Don't write anything to the device on which you want to recover data

Try to get detailed information on how the data lost and the losing process

Backup the data recovered in time.

II .Data loss

Actually, there are various reasons that cause data loss; software, hardware, factitious, natural, intended, unintended, all may cause data loss or damage on storage devices.

Generally, There are two main reasons for data problem: software and hardware whose corresponding reasons are software reason and hardware reason.

1. Software reason

Virus, format, mis-partition, mis-clone, mis-operation, network deletion, power-cut during operation all may be the software reasons. The symptoms are usually mis-operation, read error, can not find or open file, report no partition, not formatted, password lost and troubled characters.

A: Computer Viruses: some malicious virus programs will destroy data, overwrite, or erase the data contents.

B: Mis-format: fast or completely format partition, thus changing the file system form (NTFS, FAT32) of partition.

C: Mis-Clone: when backing up the hard disk, mis-clone or overlay the original data on hard disk.

For these, we can use software tools to recover it. So called soft recovery means data can be recovered by software, not referring to hardware fixing operation for its fault is not because of hardware failure.

The following are prompts that system can not start up normally:

Invalid Partition Table: Invalid partition table information.

Missing Operating System: "55AA" mark in DOS boot sector lost or DBR corrupted.

Disk Boot Failure: System file read failure.

Bad or missing command interpreter: Can not find command.com file or 'COMMAND.COM' file corrupted.

Invalid system disk: DOS boot record corrupted.

Type the name of the command, Interpreter: DOS partition mark in partition table error or 'COMMAND.COM' file lost, corrupted.

Error Loading Operating System: Main boot startup program read boot sector unsuccessfully.

Not found any active partition in HDD: Active partition mark in partition table changed as inactive partition mark.

2. Hardware reason

Sometimes data loss is because of hardware, such as bad sector in hard disk, power cut, head damage, circuit panel problem, etc.

When your hardware has some problems, you probably will find: the speed of hardware become slow, you cannot operate successfully; you cannot read data, etc, which are most often physical bad track failures.

Correspondingly, data recovery in hardware fix is considered as hard recovery, such as memory medium damage, track damage, hard disk scrape, head damage, electric machinery damage, chip burnout and so on..

The most distinct feature or difference between soft recovery and hard recovery is whether the memory medium itself can be normally accessed by fixing or replacing parts.

III.Data Protecting Technologies

Data security and fault freedom of storage are paid more and more attention. People are attaching more and more importance to developing new technologies to protect data.

1.SMART Technology

SMART, also called Self-Monitoring Analysis and Report Technology, mainly protects HD from

losing data when there is some problems on the HD. SMART drive can reduce the risk of data loss, it alarms to predict and remind thus enhancing the data security.

2.SPS

Shake Protecting System, can prevent the head from shaking thus enhancing the anti-knock characteristics of HD, avoiding damages caused by shake.

3.DFT

DFT, a kind of IBM data protecting technology, can check hard disk via using DFT program to access the DFT micro codes in hard disk. By DFT, users can conveniently check the HD operation.

4.Floppy disk array technology

Originally 'Redundant Arrays of Inexpensive Disks'. A project at the computer science department of the University of California at Berkeley, under the direction of Professor Katz, in conjunction with Professor John Ousterhout and Professor David Patterson.

The project is reaching its culmination with the implementation of a prototype disk array file server with a capacity of 40 GBytes and a sustained bandwidth of 80 MBytes/second. The server is being interfaced to a 1 Gb/s local area network. A new initiative, which is part of the Sequoia 2000 Project, seeks to construct a geographically distributed storage system spanning disk arrays and automated libraries of optical disks and tapes. The project will extend the interleaved storage techniques so successfully applied to disks to tertiary storage devices. A key element of the research will be to develop techniques for managing latency in the I/O and network paths.

The original ('Inexpensive') term referred to the 3.5 and 5.25 inch disks used for the first RAID system but no longer applies.

The following standard RAID specifications exist:

RAID 0 Non-redundant striped array

RAID 1 Mirrored arrays

RAID 2 Parallel array with ECC

RAID 3 Parallel array with parity

RAID 4 Striped array with parity

RAID 5 Striped array with rotating parity

The basic idea of RAID (Redundant Array of Independent Disks) is to combine multiple inexpensive disk drives into an array of disk drives to obtain performance, capacity and reliability that exceeds that of a single large drive. The array of drives appears to the host computer as a single logical drive. The Mean Time Between Failure (MTBF) of the array is equal to the MTBF of an individual drive, divided by the number of drives in the array. Because of this, the MTBF of

a non-redundant array (RAID 0) is too low for mission-critical systems. However, disk arrays can be made fault-tolerant by redundantly storing information in various ways.

5.SAN

SAN, called Storage Area Network or Network behind servers, is specialized, high speed network attaching servers and storage devices. A SAN allows "any to any" connection across the network, using interconnect elements such as routers, gateways, hubs and switches. It eliminates the traditional dedicated connection between a server and storage, and concept that the server effectively "owns and manages" the storage devices. It also eliminates any restriction to amount of data that a server can access, currently limited by the number of storage devices, which can be attached to the individual server. Instead, a SAN introduces the flexibility of networking to enable one server or many heterogeneous servers to share a common storage "utility", which may comprise many storage devices, including disk, tape, and optical storage. And, the storage utility may be located far from the servers which use it.

6.NAS

NAS is Network Attached Storage. It can store the quick-increased information. Backup means to prepare a spare copy of a file, file system, or other resource for use in the event of failure or loss of the original. This essential precaution is neglected by most new computer users until the first time they experience a disk crash or accidentally delete the only copy of the file they have been working on for the last six months. Ideally the backup copies should be kept at a different site or in a fire safe since, though your hardware may be insured against fire, the data on it is almost certainly neither insured nor easily replaced.

7.Backup

Backup in time may reduce the danger and disaster to the lowest, thus data security can be most ensured. In different situations, there are different ways. Both backing up important data of system with hardware and backing up key information with cloning mirror data to different storage device can work well.

IV.Elementary knowledge of hard disk

1.History of hard disk development

The hard disk drive has short and fascinating history. In 24 years it evolved from a monstrosity with fifty two-foot diameter disks holding five MBytes (5,000,000 bytes) of data to today's drives measuring 3 /12 inches wide and an inch high (and smaller) holding 400 GBytes (400,000,000,000 bytes/characters). Here, then, is the short history of this marvelous device.

Before the disk drive there were drums... In 1950 Engineering Research Associates of Minneapolis built the first commercial magnetic drum storage unit for the U.S. Navy, the ERA 110. It could store one million bits of data and retrieve a word in 5 thousandths of a second..

In 1956 IBM invented the first computer disk storage system, the 305 RAMAC (Random Access Method of Accounting and Control). This system could store five MBytes. It had fifty, 24-inch diameter disks!

By 1961 IBM had invented the first disk drive with air bearing heads and in 1963 they introduced the removable disk pack drive.

In 1970 the eight inch floppy disk drive was introduced by IBM. My first floppy drives were made by Shugart who was one of the "dirty dozen" who left IBM to start their own companies. In 1981 two Shugart 8 inch floppy drives with enclosure and power supply cost me about \$350.00. They were for my second computer. My first computer had no drives at all.

In 1973 IBM shipped the model 3340 Winchester sealed hard disk drive, the predecessor of all current hard disk drives. The 3340 had two spindles each with a capacity of 30 MBytes, and the term "30/30 Winchester" was thus coined.

In 1980, Seagate Technology introduced the first hard disk drive for microcomputers, the ST506. It was a full height (twice as high as most current 5 1/4" drives) 5 1/4" drive, with a stepper motor, and held 5 Mbytes. My first hard disk drive was an ST506. I cannot remember exactly how much it cost, but it plus its enclosure, etc. was well over a thousand dollars. It took me three years to fill the drive. Also, in 1980 Phillips introduced the first optical laser drive. In the early 80's, the first 5 1/4" hard disks with voice coil actuators (more on this later) started shipping in volume, but stepper motor drives continued in production into the early 1990's. In 1981, Sony shipped the first 3 1/2" floppy drives.

In 1983 Rodime made the first 3.5 inch rigid disk drive. The first CD-ROM drives were shipped in 1984, and "Grolier's Electronic Encyclopedia," followed in 1985. The 3 1/2" IDE drive started its existence as a drive on a plug-in expansion board, or "hard card." The hard card included the drive on the controller which, in turn, evolved into Integrated Device Electronics (IDE) hard disk drive, where the controller became incorporated into the printed circuit on the bottom of the hard disk drive. Quantum made the first hard card in 1985.

In 1986 the first 3 /12" hard disks with voice coil actuators were introduced by Conner in volume, but half (1.6") and full height 5 1/4" drives persisted for several years. In 1988 Conner introduced the first one inch high 3 1/2" hard disk drives. In the same year PrairieTek shipped the first 2 1/2" hard disks.

In 1997 Seagate introduced the first 7,200 RPM, Ultra ATA hard disk drive for desktop computers and in February of this year they introduced the first 15,000 RPM hard disk drive, the Cheetah

X15. Milestones for IDE DMA, ATA/33, and ATA/66 drives follow:

1994 DMA, Mode 2 at 16.6 MB/s

1997 Ultra ATA/33 at 33.3 MB/s

1999 Ultra ATA/66 at 66.6 MB/s

6/20/00 IBM triples the capacity of the world's smallest hard disk drive. This drive holds one gigabyte on a disk which is the size of an American quarter. The world's first gigabyte-capacity disk drive, the IBM 3380, introduced in 1980, was the size of a refrigerator, weighed 550 pounds (about 250 kg), and had a price tag of \$40,000.

2.Main technical specification and parameter of hard disk

Capacity

We can see the capacity in two aspects: the total capacity and the capacity of one disk. The whole capacity is made up of each disk capacity.

If we increase the disk capacity, we would not only improve the disk capacity, improve the speed of transmission, but also cut the cost down.

Rotate speed.

Rotate speed is the speed disk rotate. It is measured by RPM (Round Per Minute).The rotate speed of IDE hard disk are 5400RPM, 7200RPM etc.

Average Seek Time

The average seek time gives a good measure of the speed of the drive in a multi-user environment where successive read/write request are largely uncorrelated.

Ten ms is common for a hard disk and 200 ms for an eight-speed CD-ROM.

Average Latency

The hard disk platters are spinning around at high speed, and the spin speed is not synchronized to the process that moves the read/write heads to the correct cylinder on a random access on the hard disk. Therefore, at the time that the heads arrive at the correct cylinder, the actual sector that is needed may be anywhere. After the actuator assembly has completed its seek to the correct track, the drive must wait for the correct sector to come around to where the read/write heads are located. This time is called *latency*. Latency is directly related to the spindle speed of the drive and such is influenced solely by the drive's spindle characteristics. This operation page discussing spindle speeds also contains information relevant to latency.

Conceptually, latency is rather simple to understand; it is also easy to calculate. The faster the disk is spinning, the quicker the correct sector will rotate under the heads, and the lower latency will be. Sometimes the sector will be at just the right spot when the seek is completed, and the latency for that access will be close to zero. Sometimes the needed sector will have just passed the head and in this "worst case", a full rotation will be needed before the sector can be read. On average, latency will be half the time it takes for a full rotation of the disk.

Average Access Time

Access time is the metric that represents the composite of all the other specifications reflecting random performance positioning in the hard disk. As such, it is the best figure for assessing overall positioning performance, and you'd expect it to be the specification most used by hard disk manufacturers and enthusiasts alike. Depending on your level of cynicism then, you will either be very surprised or not surprised much at all, to learn that it is rarely even discussed. Ironically, in the world of CD-ROMs and other optical storage it is the figure that is universally used for comparing positioning speed. I am really not sure why this discrepancy exists.

Perhaps the problem is that access time is really a derived figure, comprised of the other positioning performance specifications. The most common definition is:

$$\text{Access Time} = \text{Command Overhead Time} + \text{Seek Time} + \text{Settle Time} + \text{Latency}$$

The speed with which data can be transmitted from one device to another. Data rates are often measured in megabits (million bits) or megabytes (million bytes) per second. These are usually abbreviated as Mbps and MBps, respectively.

Buffer Size (Cache)

A small fast memory holding recently accessed data, designed to speed up subsequent access to the same data. Most often applied to processor-memory access but also used for a local copy of data accessible over a network etc.

When data is read from, or written to, main memory a copy is also saved in the cache, along with the associated main memory address. The cache monitors addresses of subsequent reads to see if the required data is already in the cache. If it is (a cache hit) then it is returned immediately and the main memory read is aborted (or not started). If the data is not cached (a cache miss) then it is fetched from main memory and also saved in the cache.

The cache is built from faster memory chips than main memory so a cache hit takes much less time to complete than a normal memory access. The cache may be located on the same integrated circuit as the CPU, in order to further reduce the access time. In this case it is often known as primary cache since there may be a larger, slower secondary cache outside the CPU chip.

The most important characteristic of a cache is its hit rate - the fraction of all memory accesses which are satisfied from the cache. This in turn depends on the cache design but mostly on its size relative to the main memory. The size is limited by the cost of fast memory chips.

The hit rate also depends on the access pattern of the particular program being run (the sequence of addresses being read and written). Caches rely on two properties of the access patterns of most programs: temporal locality - if something is accessed once, it is likely to be accessed again soon, and spatial locality - if one memory location is accessed then nearby memory locations are also likely to be accessed. In order to exploit spatial locality, caches often operate on several words at a

time, a "cache line" or "cache block". Main memory reads and writes are whole cache lines.

When the processor wants to write to main memory, the data is first written to the cache on the assumption that the processor will probably read it again soon. Various different policies are used. In a write-through cache, data is written to main memory at the same time as it is cached. In a write-back cache it is only written to main memory when it is forced out of the cache.

If all accesses were writes then, with a write-through policy, every write to the cache would necessitate a main memory write, thus slowing the system down to main memory speed. However, statistically, most accesses are reads and most of these will be satisfied from the cache. Write-through is simpler than write-back because an entry that is to be replaced can just be overwritten in the cache as it will already have been copied to main memory whereas write-back requires the cache to initiate a main memory write of the flushed entry followed (for a processor read) by a main memory read. However, write-back is more efficient because an entry may be written many times in the cache without a main memory access.

When the cache is full and it is desired to cache another line of data then a cache entry is selected to be written back to main memory or "flushed". The new line is then put in its place. Which entry is chosen to be flushed is determined by a "replacement algorithm".

Some processors have separate instruction and data caches. Both can be active at the same time, allowing an instruction fetch to overlap with a data read or write. This separation also avoids the possibility of bad cache conflict between say the instructions in a loop and some data in an array which is accessed by that loop.

Noise & Temperature

It comes from motor. So motor is the key to reduce the noise and temperature. If you can keep the temperature of hard disk down, then you can keep your hard disk effective.

3.Physical structure of hard disk

HD consists of platter, control circuit board and interface parts.

A hard disk is a sealed unit containing a number of platters in a stack. Hard disks may be mounted in a horizontal or a vertical position. In this description, the hard drive is mounted horizontally.

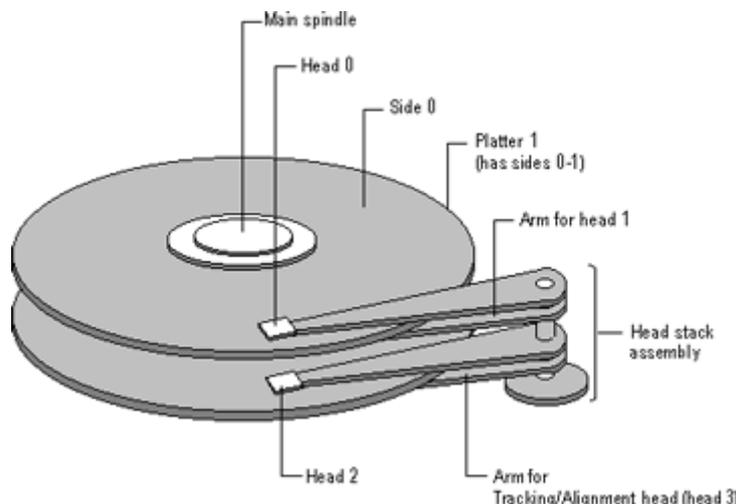
Electromagnetic read/write heads are positioned above and below each platter. As the platters spin, the drive heads move in toward the center surface and out toward the edge. In this way, the drive heads can reach the entire surface of each platter.

Making Tracks

On a hard disk, data is stored in thin, concentric bands. A drive head, while in one position can read or write a circular ring, or band called a track. There can be more than a thousand tracks on a 3.5-inch hard disk. Sections within each track are called sectors. A sector is the smallest physical storage unit on a disk, and is almost always 512 bytes (0.5 kB) in size.

The figure below shows a hard disk with two platters.

Figure 3-1 Parts of a Hard Drive



The structure of older hard drives (i.e. prior to Windows 95) will refer to a cylinder/ head/ sector notation. A cylinder is formed while all drive heads are in the same position on the disk. The tracks, stacked on top of each other form a cylinder. This scheme is slowly being eliminated with modern hard drives. All new disks use a translation factor to make their actual hardware layout appear continuous, as this is the way that operating systems from Windows 95 onward like to work..

To the operating system of a computer, tracks are logical rather than physical in structure, and are established when the disk is low-level formatted. Tracks are numbered, starting at 0 (the outermost edge of the disk), and going up to the highest numbered track, typically 1023, (close to the center). Similarly, there are 1,024 cylinders (numbered from 0 to 1023) on a hard disk.

The stack of platters rotate at a constant speed. The drive head, while positioned close to the center of the disk reads from a surface that is passing by more slowly than the surface at the outer edges of the disk. To compensate for this physical difference, tracks near the outside of the disk are less-densely populated with data than the tracks near the center of the disk. The result of the different data density is that the same amount of data can be read over the same period of time, from any drive head position.

The disk space is filled with data according to a standard plan. One side of one platter contains space reserved for hardware track-positioning information and is not available to the operating system. Thus, a disk assembly containing two platters has three sides available for data. Track-positioning data is written to the disk during assembly at the factory. The system disk controller reads this data to place the drive heads in the correct sector position.

4.Logical organization of hard disk

Sectors and Clusters

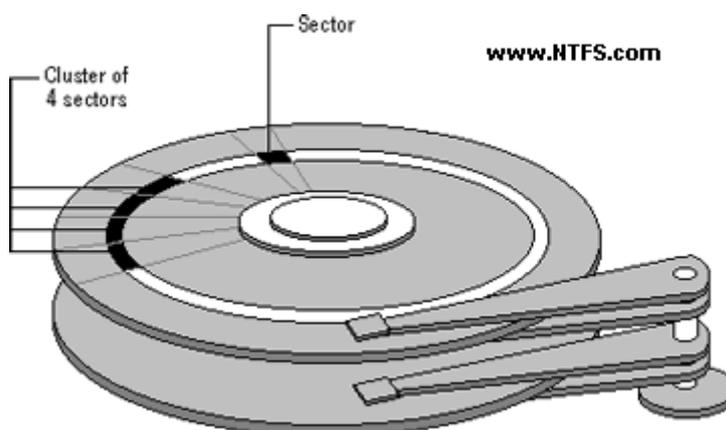
A sector, being the smallest physical storage unit on the disk, is almost always 512 bytes in size because 512 is a power of 2 (2 to the power of 9). The number 2 is used because there are two states in the most basic of computer languages - on and off.

Each disk sector is labelled using the factory track-positioning data. Sector identification data is written to the area immediately before the contents of the sector and identifies the starting address of the sector.

The optimal method of storing a file on a disk is in a contiguous series, i.e. all data in a stream stored end-to-end in a single line. As many files are larger than 512 bytes, it is up to the file system to allocate sectors to store the file's data. For example, if the file size is 800 bytes, two 512 k sectors are allocated for the file. A cluster is typically the same size as a sector. These two sectors with 800 bytes of data are called two clusters.

They are called clusters because the space is reserved for the data contents. This process protects the stored data from being over-written. Later, if data is appended to the file and its size grows to 1600 bytes, another two clusters are allocated, storing the entire file within four clusters.

Figure 3-2 Sectors and Clusters



If contiguous clusters are not available (clusters that are adjacent to each other on the disk), the second two clusters may be written elsewhere on the same disk or within the same cylinder or on a different cylinder - wherever the file system finds two sectors available. A file stored in this non-contiguous manner is considered to be fragmented. Fragmentation can slow down system performance if the file system must direct the drive heads to several different addresses to find all the data in the file you want to read. The extra time for the heads to travel to a number of addresses causes a delay before the entire file is retrieved.

Cluster size can be changed to optimize file storage. A larger cluster size reduces the potential for fragmentation, but increases the likelihood that clusters will have unused space. Using clusters larger than one sector reduces fragmentation, and reduces the amount of disk space needed to store the information about the used and unused areas on the disk.

Most disks used in personal computers today rotate at a constant angular velocity. The tracks near the outside of the disk are less densely populated with data than the tracks near the center of the disk. Thus, a fixed amount of data can be read in a constant period of time, even though the speed of the disk surface is faster on the tracks located further away from the center of the disk..

Modern disks reserve one side of one platter for track positioning information, which is written to the disk at the factory during disk assembly. It is not available to the operating system. The disk controller uses this information to fine tune the head locations when the heads move to another location on the disk. When a side contains the track position information, that side cannot be used for data. Thus, a disk assembly containing two platters has three sides that are available for data.

Hard disk interfaces

Hard disks also come in several flavors such as IDE (actually ATA), SCSI and SATA, as do optical drives. ATA is the most common interface used today. SCSI disks can usually be found on servers.

IDE

Integrated Drive Electronics, more commonly called by its acronym IDE, is an interface for hard drives. IDE is a marketing term; the real standard is called ATA.

EIDE (Enhanced IDE) or ATA-2 was later developed and increased transfer speed, added 32-bit transactions and DMA support.

ATA

ATA stands for Advanced Technology Attachment. The ATA -term is commonly used interchangeably with IDE. The older and more common parallel ATA (P-ATA) is currently being replaced by serial ATA (SATA).

Most PCs have two IDE controllers on the motherboard. One IDE controller can support two devices, so four storage devices is usually the maximum. Parallel ATA interface uses ribbon cables with 40 -pin connectors to connect the hard drives to the motherboard. The cable has usually three connectors. Of these one is connected to the motherboard and the rest two are left for hard drives. If two hard drives are connected to the same controller, one must be defined as master and the other one as slave. This is done with jumpers.

ATA-2 is the real standard for what is widely known as EIDE. ATA-2 introduced higher speed data transfer modes: PIO Modes 3 and 4 plus Multiword DMA Mode 1 and 2. These modes allow the ATA interface to run data transfers up to about 16MB/second.

SATA

Serial ATA, also known as SATA or S-ATA, is a bus used to communicate between the CPU and internal storage devices such as hard drives and optical drives. It is designed to eventually replace the ATA (also known as IDE) bus. Traditional ATA is beginning to be referred to as Parallel ATA, P-ATA, or PATA to avoid confusion.

The main difference between SATA and PATA is in the cabling. SATA does away with the

master/slave relationship of PATA (hence the difference in names), as well as PATA's ungainly ribbon cables. Instead, SATA has much slimmer and easier to manage cables, which will enable better airflow through cases. The connectors are keyed, preventing connectors from being plugged upside down. Truly native SATA drives will have different power connectors also.

A third advantage of SATA is hotplugging.

Currently, SATA has a transfer rate of 150 MB/s, which is only 17 MB/s more than standard PATA. However, with the introduction of SATA II, this is expected to go up to 300 MB/s, with 600 MB/s being released sometime around 2007. The faster bus isn't expected to affect performance in the short term, since hard drive performance is usually bottlenecked by the moving parts of the drive. During the transitional period before true native SATA drives are released, most SATA drives actually have onboard PATA controllers, which connect to SATA by a bridge. This generally causes a 30-50% performance drop. Also, PATA power connectors are still being used.

DMA

DMA (Direct Memory Access) is a function of the memory bus in the computer that lets connected devices like hard disks transfer data to the memory without the intervention of the CPU, thus speeding up the transfer. This is superior to the way PIO works.

There are two distinct types of direct memory access, DMA and bus mastering DMA. The plain DMA relies on the DMA controller on the motherboard to grab the system bus and transfer the data. In bus mastering DMA all this is done by the logic on the interface card itself. Bus mastering allows the hard disk and memory to work without relying on the old DMA controller built into the system, or needing any support from the CPU.

USB

USB (Universal Serial Bus) is a hardware bus using a serial protocol used by many different hardware devices and supported in most computers/mainboards. Originally developed by Compaq, Intel, NEC and Microsoft. It allows many devices to be connected to the bus at the same time, the theoretical maximum is 127 devices. The maximum data transfer bandwidth is about 12Mbit/s (USB2.0 supports 480 Mbit/sec).

Firewire is a less known alternative to USB that (at its time) was better than USB for media related tasks. As of USB2 there have been significant increases, specifically more bandwidth.

SCSI

SCSI - Small Computer System Interface. Pronounced "scuzzy". It's a specification for a hardware interface for connecting devices such as hard disks and scanners to a computer.

Most PCs have an ATA(IDE) bus instead of SCSI for connecting internal hard disks. SCSI is seen more often in servers, as it tends to be faster and more reliable (though more expensive). Another advantage of SCSI controller is that it requires only one IRQ and can handle usually at least 7 devices whereas ATA can handle only 2.

Typically, you put a SCSI card in your computer, and then connect internal hard disks with a ribbon cable to some connector on the card. Also, the card will have an external connector which you might also be using simultaneously.

5.Connection synopsis of hard disk

Fiber Channel

Fibre Channel Hard Disk Drive

The Enterprise Virtual Array supports any combination of five different Fibre Channel Hard Disk Drives (HDD) with multiple capacity points and two different rotational speeds. Three drive capacity points are supported at 36 GB, 72 GB, and 146 GB. Two rotational speeds are supported at 10,000 RPM and 15,000 RPM.

The following individual drive capacity/rotational speed combinations are available:

146GB 10,000 RPM Fibre Channel HDD

72GB 15,000 RPM Fibre Channel HDD

72GB 10,000 RPM Fibre Channel HDD

36GB 15,000 RPM Fibre Channel HDD

36GB 10,000 RPM Fibre Channel HDD

Five different Fibre Channel HDDs for the Enterprise Virtual Array provides tremendous flexibility to the target customer base by allowing mixing and matching of capacity and performance to application needs. Application areas seen as potential markets include OLTP, ERP, and any other applications requiring large amounts of online storage.

IEEE

Also called Firewire. it is a less known alternative to USB that (at its time) was better than USB for media related tasks. As of USB2 there have been significant increases, specifically more bandwidth.

Intermediate

V.Hard disk data organization

1.Primary formatting of hard disk

Before restoring data, hard disk usually needs low-level format, partition, high-level format to be used. The function is establishing certain data logical structure on physical hard disk. Usually hard disk is divided into 5 regions: MBR, DBR, DIR, FAT and DATA (Here we do not introduce FAT

and NTFS file system), which altogether store and manage data.

Low level format

After setting parameter of hard disk in CMOS Setting, why the hard disk is still unusable? That's about Cylinder, Header and Sector. When hard disk is firstly made in the factory, it usually is "blank". Only after partitioning tracks and sectors, we can save data on hard disk (Now, before leaving the factory, many disks have been low-level formatted. So you may need not do the operation, but it is not unnecessary.)

Main functions of low level format

Low level format can also be called physical format, whose functions are to detect the magnetic media, to partition tracks, to partition sectors for each track, and to arrange the order of partitions in track according to the interleave the customer choose. Its main functions are as following:

1. Test the hard disk media
2. Partition tracks for hard disk
3. Arrange sectors for each track according to the specified interleave
4. Set the sector ID to each track and finish setting sectors
5. Test the hard disk surface, mark "bad" to the damaged track and sector
6. Write a certain ASC II to each sector of hard disk

Hard disk is an important storage resource in computer system. Do not low-level format the hard disk unless it is the only thing possible. For hard disk being used, you need back up important data before low level format; even if back up is unnecessary, it may take much time to partition, high-level format, and install system and application programs. Usually, low level format can be used in the following cases:

When you have bought a new hard disk or hard disk adapter, you'd better low level format it again, which is for the better matching of hard disk and hard disk adapter.

"Bad" sectors, which result from long-time operation, often cause "sector not found" error in DOS. This is because of the loss of sector ID. Sector ID is used to distinguish the sectors. It is marked onto disk as the magnetization map, which however, may dribble away for long-time storage or use. Low level format is the only way for computer users to refresh sector ID in disk. This assignment cannot be done by high level format.

Appropriate set of interleave can fasten data transfer. In most conditions, low level format is the only way to change the interleave.

When there are always inexplicable problems, you can take low level format into consideration.

Ways to low-level format

There are many ways to low-level format. In early time, it can be completed in CMOS or by some

special disk tools, or by writing some short programs in Debug. Nowadays, people usually use special tools provided free by hard disk manufacturers.

2. Advanced formatting of hard disk

High-level format

After partitioning the hard disk, some “independent” logical drivers are founded. If now we start system from the floppy drive, enter DOS, then you can see the drive letters of DOS partition, which is on behalf of logical driver, for instance “C:”, “D:” and so on. The system commonly arranges letters according to alphabet. Now let’s try to enter “C:” or “D:” after that we can see the system prompt that “DISK MEDIA ERROR”. Why? These logical disks are empty; to use them, we need create file system. The whole process is high-level format of logic disk. The high-level format certainly aims at the logic disk, neither physical disk nor certain directory. For file system is corresponding to logic disk, we can say that high-level format aims at file system. In this article, logical disk means logical drive.

Format partition

High-level format of DOS logic disk can be completed by “**format**” command. Main functions of high-level format are as following:

Assign logical serial numbers for sectors (serial numbers in partition) from cylinder that assigned by each logical drive

Establish DBR in basic partition, and load 3 system files of DOS if there is “/S” parameter in the command.

Establish file allocation table (FAT) in each logical disk.

Establish File Directory Table (FDT) that is corresponding to root directory and data area.

If you carry out high-level format by “Format” command, please pay attention to following 4 items.

1. To already activated basic DOS partition (generally it is disk C), you need the following command:

```
Format C:/s
```

By this command, you may install DOS system files after high-level format, to make this logical disk to become the boot disk. Certainly, you may also use “SYS” command to send system files after high-level format, that is complete the boot disk and file transmission by the following two commands:

```
Format C:
```

SYS C:

Continuously using these two commands equals to “Format C:/S” command.

2. For other logical disk, we only need to carry out the following commands:

Format [d:]

“d” is the logical disk drive.

3. Before format, on the screen it may appear the following prompt information:

```
WARNING: ALL DATA ON THE DISK  
DRIVE C: WILL BE LOST!  
Proceed with Format (Y/N) ? _
```

This information is warning user: The format will cause all data lose! Then, if user choose “Y”, then the high-level format officially carries on, if user choose “N”, then nothing will happened and exit.

4. For the using disk without adjusting the partition, also may carry on the fast format, the command is:

Format C: /Q

The full command of “Format” in Windows 2000 is as following shows:

```
FORMAT volume [/FS:file-system] [/U:label] [/Q] [/A:size] [/C] [/X]  
FORMAT volume [/U:label] [/Q] [/F:size]  
FORMAT volume [/U:label] [/Q] [/T:tracks /N:sectors]  
FORMAT volume [/U:label] [/Q] [/1] [/4]  
FORMAT volume [/Q] [/1] [/4] [/8]
```

Format hard disk partition in Windows

In explore of Windows, everything is displayed by graphics, and different forms (partition) are expressed by different colors. Click the right key in the corresponding partition, and choose “format”, you may also choose fast format, complete format and so on.

Format hard disk partition by Partition Magic

In Partition Magic, everything is displayed by graphics, and different forms (partition) are expressed by different colors. Click the right key in the corresponding partition, and choose “format”. In the dialogue box, there will be a prompt indicating this operation may destroy your own data, and in the box you may also choose different format.

Format hard disk by various hard disks special-purpose tool in hard disk factory

Low level format tool provided by various hard disks factory can help hard disk breakthrough

hard disk capacity limit, as well as complete low level format, the high-level format and make partitions. After partitions are done, you can choose corresponding options step by step.

Attention: To partitions with data, backup the data before format.

High-level format establish the file system, after format, it may carry on write in and read out operations with file as unit.

3.Data storage region of hard disk

In command to know hard disk better, we must have a simple understanding of hard disk construction. (NTFS uses different file management technology with FAT16 and the FAT32 file system, here we only introduce FAT16 and FAT32) The hard disk data may divide into 5 parts approximately according to its different characteristics and functions: MBR area, DBR area, FAT area, DIR area and DATA area. Among them, MBR is founded by the partition software, while DBR area, FAT area, DIR area and DATA area are founded by high-level format procedure. When file system writes in data, it just rewrites corresponding FAT area, DIR area and DATA area. Also it is the result which these 5 regions affect together. Only by this way, hard disk can be managed methodically. Here are some introductions to the 5 regions.

MBR:

The first physical sector (cylinder 0, head 0, sector 1) of the first hard drive in the system (the first hard drive with the BIOS device number 0x80); each hard drive contains an MBR, but not every BIOS can start the corresponding operation system from every hard drive. When booting from the hard drive, the BIOS or a special Firmware loads the contents of the MBR to a fixed address in the memory and allows it to take control. This code then loads either the operation system from a bootable hard drive partition, or from a complex boot loader, such as LILO.

Short for DOS Boot Record, it is the sector at cylinder 0, column 1, and sector 1 of a hard disk. DBR is the first sector that the operation system visits. It contains a boot program and a BPB (BIOS Parameter Block). The main task of the boot program is to determine whether the first two files in root directory of this partition are the boot files of operation system, when MBR hands over the system mastery to it. Take an example of DOS, i.e. IO.SYS and MSDOS.SYS. DOS of low edition requests that these two files are the first two files, and located at the section start of the root directory, covering the first two directory items (the high edition does not have this requirement.). Moreover, Windows and DOS are families; therefore, Windows follows the same management manner, except for the filenames. If it does exist, then reads IO.SYS in the memory, and hands over mastery to IO.SYS. BPB parameter block records the start sector, ending sector, file storage form, descriptor of hard disk media, size of root directory, number of FAT and size of allocated cell.

File Allocation Table (FAT) is a file system that was developed for MS-DOS and is the primary file system for consumer versions of Microsoft Windows up to and including Windows ME. The FAT file system is considered relatively uncomplicated, and because of that, it is a popular format

for floppy disks; moreover, it is supported by virtually all existing operation systems for personal computers, and because of that, it is often used to share data between several operation systems booting on the same computer (a multi-boot environment). It is also used on solid-state memory cards and other similar devices. It has a serious drawback in that when files are deleted and new files written to the media, the files can become scattered over the entire media making reading and writing a slow process. De-fragmentation is one solution to this, but is often a lengthy process in itself and has to be repeated regularly to keep the FAT file system clean.

FAT is also called 12-bit FAT, the file allocation table (FAT) for a floppy disk. The location of files on a floppy disk are listed in a one-column table in the FAT. Because the width of each entry in a floppy disk column is 12 bits, the FAT is called FAT12. As a file system for floppy disks, it had a number of limitations: no support for hierarchical directories, cluster addresses were “only” 12-bits long (which made the code manipulating the FAT a bit tricky) and the disk size was stored as a 16-bit count of sectors, which limited the size to 32MB.

The FAT file system, as is the case with most file systems, does not utilize individual sectors, and there are several performance reasons for this. By using individual sectors, the process of managing disks becomes overly cumbersome since files are being broken into 512-byte pieces. If you were to take a 20 GB disk volume set up with 512 byte sectors and manage them individually, the disk would have over 40 million individual sectors. Just keeping track of this many pieces of information is both time, as well as resource, consuming. While some operation systems do allocate specific sector storage, they also require some advanced intelligence to do so. Bear in mind how old the FAT file system is, as it was designed many years ago as merely a simple file system, without the capability to managed individual sectors.

In order for FAT to manage files with some form of efficiency is to group sectors into larger blocks referred to as clusters, or allocation units. Cluster size, however, is not a predetermined size, but rather is determined by the size of the disk volume itself, with small volumes (disk sizes) resulting in smaller clusters, and larger volumes (disk sizes) using larger cluster sizes. For the most part, a cluster ranges in size from 4 sectors or 2,048 bytes to 64 sectors or 32,768 bytes. You should be aware that you may, on some occasions, find 128-sector clusters in use at 65,536 bytes per cluster, as well as some floppy disks with smaller clusters that is usual at just 1 sector per cluster. In all cases, the sectors in a cluster are continuous, therefore each cluster is a continuous block of space on the disk.

Cluster sizing, and therefore partition or volume size, as they are directly related, have an important impact on performance and disk utilization. In all cases, cluster size is determined at the time a disk volume is partitioned. Certain third-party partitioning utilities such as Partition Magic by PowerQuest can alter the cluster size of an existing partition within specific parameters. However, this aside, once the partition size is selected, so are the cluster sizes fixed.

FAT 16 means that file allocation table that uses 16 bits for addressing clusters. It is commonly used with DOS and Windows 95 systems. A 16-bit DOS and Windows file system (see FAT) that varies cluster sizes based on hard drive size. Cluster sizes range from 4K (for drives up to 127MB),

to 4K (255MB drives), 8K (511MB drives), 16K (1GB drives). and 32K (for drives up to 2GB). The ultimate capacity of a FAT16 partition is 2GB.

FAT 32 is a disk file allocation system from Microsoft that uses 32-bit values for FAT entries instead of 16-bit values used by the original FAT system, enabling partition sizes up to 2TB (terabytes). FAT32 first appeared in Windows 95B and is also found in Windows 98 and Windows NT 5.0.

In order to overcome the volume size limit of FAT16 while still allowing memory-constrained DOS real-mode code to handle the format, Microsoft decided to implement a newer generation of FAT, known as FAT32, with 32-bit cluster numbers, of which 28 bits are currently used.

In theory, this should support a total of approximately 268,435,438 ($< 2^{28}$) clusters, allowing for drive sizes in the range of 2 terabytes. However, due to limitations in Microsoft's scandisk utility, the FAT is not allowed to grow beyond 4,177,920 ($< 2^{24}$) clusters, placing the volume limit at 124.55 gigabytes, unless "scandisk" is not needed. Windows 2000 and XP placed a limit on the size of FAT32 partitions they can create at 32 GB, Microsoft says this is by design but does not explain why, and those versions of Windows are quite capable of reading and writing larger FAT32 partitions created by other means. FAT32 was introduced with Windows 95 OSR2. The many changes it incorporated made it a major improvement.

The maximum possible file size for a FAT32 volume is 4 GB minus 1 byte ($2^{32}-1$ bytes). For most users, this has become the most nagging limit of FAT32 as of 2005, since video capture and editing applications can easily exceed this limit, as can the system swap file.

32-bit File Allocation Table File System Not the same as VFAT or FAT, which are both 16-bit file systems.

DIR

Means Directory, also called FDT, File Directory Table. DIR is the root sector, following after the second FAT (backup FAT). It records each start cell, files. Operation system can locate files according to the outset of FAT and FAT.

DATA

DATA area is the real place where data is stored. It is after DIR, covering the most space of hard disk.

The location of the 5 areas is as following:



Usually, MBR covers 63 sectors (actually it covers only one); DBR covers 32 sectors (actually it covers the first and the sixth sectors. The first sector works while the sixth is backup of the first); FAT1=FAT2. The length of FAT will change according to the size of partition and the number of sectors. DIR changes the most. In early time system, DIR has fixed length of 32 sectors while each

file directory covers 32 bytes. As a result, there are at most 512 items under root directory. Floppy disk can only contain 112 items, or there would be no file or directory created under root directory. Afterward, the limitation is broken. From then on, there will be no single root directory, which becomes part of DATA. Even, root directory files are not right after FAT. They can be in any position in DATA.

VI.Common Cases of Partition Recovery

1.MBR Recovery

On condition that there is no problem with hardware, the first step is MBR recovery. MBR recovery is simple because it is system data. Though it may be created by different software and the code might be different, the method is the same. Even if multi-system boot, it is not hard. You can backup the data to be recovered after the system boot turn to be normal, and then restore the

multi system boot.

Recover MBR by fdisk

The simplest way to recover MBR is Fdisk, whose command is simple too; you can use “**Fdisk/MBR**”. Please note that, the hard disk to be operated should be connected on mater IDE interface as the master hard disk. As to other connection way, we need appoint the interface location of IDE device in form of “**Fdisk/CMBR**”.

The command syntax of Fdisk command line is “**Fdisk/parameter switch**”. Besides that obtained by “**FDISK/?** ”, there are some hidden parameters information:

/ACTOK

Parameter Function: not to check bad sectors on disk surface

Details: It can speed up partition operation.

/CMBR

Parameter Function: to re-create MBR of appointed disk

Details: Equals to /MBR parameter, except that it can appoint certain disk

/EXT

Parameter Function: to create extend partition.

Details: Creates extend partition on the currency disk , which used to create logical partition.

/FPRMT

Parameter Function: to check the usage of FAT16 and FAT32 in interactive mode.

Details: When /FPRMT parameter is added, there will be no query of that whether supports high- capacity hard disk; while there will be a query that it uses FAT16 or FAT32 when creating a new partition.

/LO

Parameter Function: to rebuild logical partition.

Details: Used to create logical disk, /LOG and /EXT should work together.

/LOGO

Parameter Function: to create logical partition with FAT16

/MBR

Parameter Function: to re-create MBR of master disk

Details: to clear the system booting choice recorded in MBR after uninstalling Windows NT or Windows 2000

/PRI

Parameter Function: to create primary partition and activate it.

Details: e to create primary partition, and the partition will be set active automatically.

/PRIO

Parameter Function: to create primary partition of FAT16 and activate it.

/Q

Parameter Function: not to restart computer when ending Fdisk

Details: unnecessary to restart computer after changing the partition table.

/STATUS

Parameter Function: to display details of current partition

Details: When there is no logical partition in extend partition, the extend partition will not be displayed.

/X

Parameter Function: no LBA attribute

Details: there would be no partition with LBA attribute.

It makes handier to use Fdisk with these parameters. However, to hide the parameter will be more dangerous, which calls for more caution.

Uses Fixmbr to restore MBR

Provided by Microsoft, Fixmbr is a MBR recovery tool, which determines hard disk partition and re-construct MBR through overall search.

Only when using Windows 2000 recovery console that we can use Fixmbr. Windows 2000 recovery console can boot from Windows install CD. Fixmbr only revises MBR; it does not write other sectors, which is safe. You can get help information of Fixmbr as following when using Fixmbr/? .

```
Usage: FIXMBR [DriveNo] [/A] [/D] [/P] [/Z] [/H]
DriveNo Hard disk scope 0-3, default is all drive.
/A      Active DOS partition.
/P      Display partition.
/D      Display MBR.
/Z      Zero MBR.
/H      This message.
```

The parameter “DriveNo” is to write a new MBR (driver). The device name can be obtained from output of the map command. For example, device name:

```
/Device/HardDisk0
```

The following command is to write a new MBR to the appointed device:

```
fixmbr /Device/HardDisk0
```

Attention: If we do not assign DriverNo, the new MBR will be written in booting device, namely the driver that loads host system. If the system detects invalid or the non-standard partition mark, it will prompt that whether continue to execute this command or not. Only if there are some problems with the driver you visit; otherwise, please do not continue.

By default MBR structure will be checked. If it is abnormal, it will prompt that whether recover or not. If choose “Y”, it will search partitions. When it has found the partition, it will also prompt that whether to revise MBR or not. If choose “Y”, recovery will be finished. If the system is down now, please inactivate the anti-virus function in BIOS first and then continue.

By default, it will search all existing hard disk, and finish all mentioned operations above. If the result is not right, you may use “/Z” parameter to clear the result and restart; then it returns to the original condition.

2.Recovery of Partition

The partition recovery is generally the second step of the whole process. Because apart from some tools that directly reads and writes hard disk, most of tool software runs under operation system, working with the system calling. While operation system’s visiting disk is on the basis of MBR and DBR; without MBR and DBR, operation system is unable to visit file system. Therefore, if the partition table is corrupted, we need rebuild partition table, which is usually fulfilled manually; in some special cases it can be done automatically by some working software.

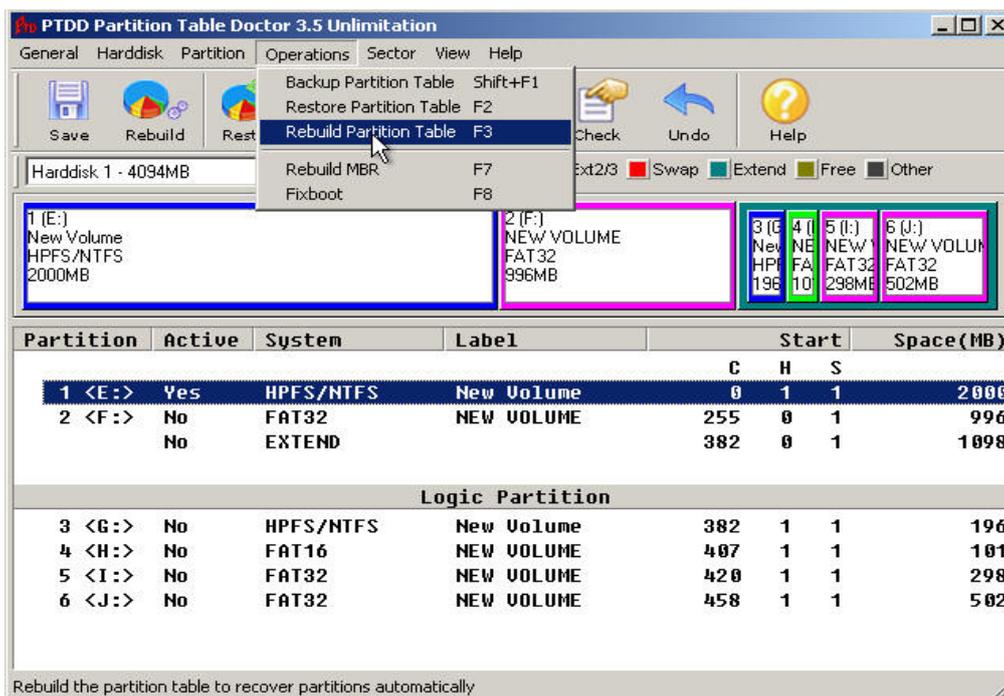
If partition table is corrupted, there are many tools to rebuild it automatically, if only the problem is not too serious. If it is too serious, or the partition table structure is too complex, it may possibly be out of the reach of their ability to rebuild. In this case, we need do it manually. Usually we use some tool software to recover the lost partition table, such as Norton Utilities 8.0, DiskMan, KV3000/Kavfix 和 PartitionMagic etc. Here we introduce **Partition Table Doctor**.

3.Partition Table Doctor

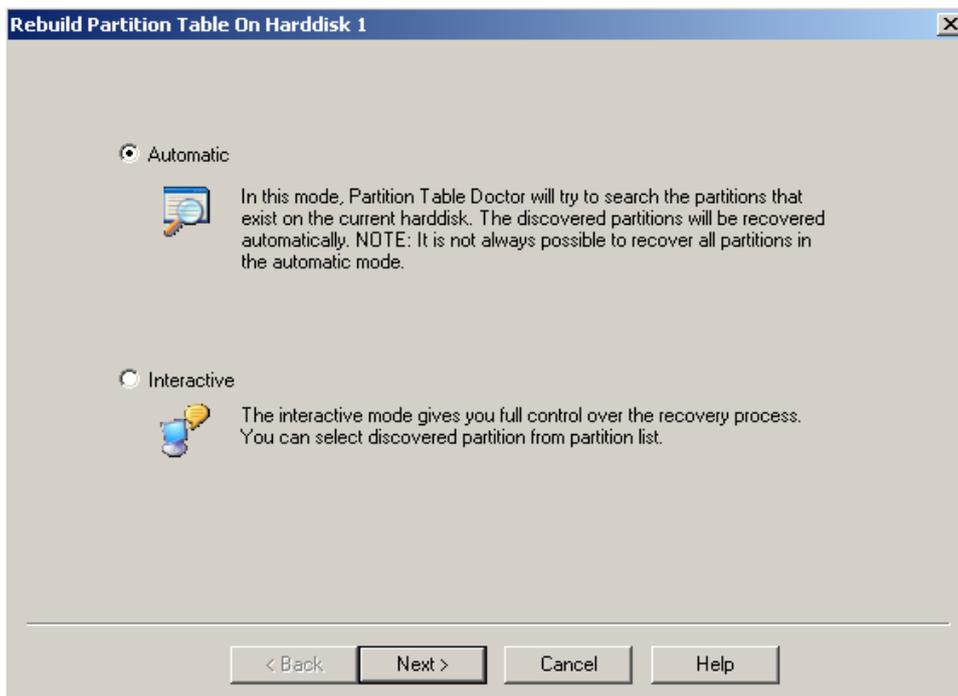
Partition Table Doctor is the only real software for hard disk partitions recovery. When you come up against a drive error (not hardware failure) this versatile tool would automatically check and repair the Master Boot Record, partition table, and the boot sector of the partition with an error, to recover the FAT16/FAT32/NTFS/NTFS5/EXT2/EXT3/SWAP partition on IDE/ATA/SATA/SCSI hard disk drives. It can create an emergency floppy disk or a bootable CD to recover the bad partition even if your operation system fails to boot. Partition Table Doctor manages for MS-DOS, Freedos, Windows 95/98/Me, Windows NT 4.0, Windows 2000, Windows XP and Windows 2003. There are two modes for partition recovery: “auto mode” and “interactive mode”.

Auto mode

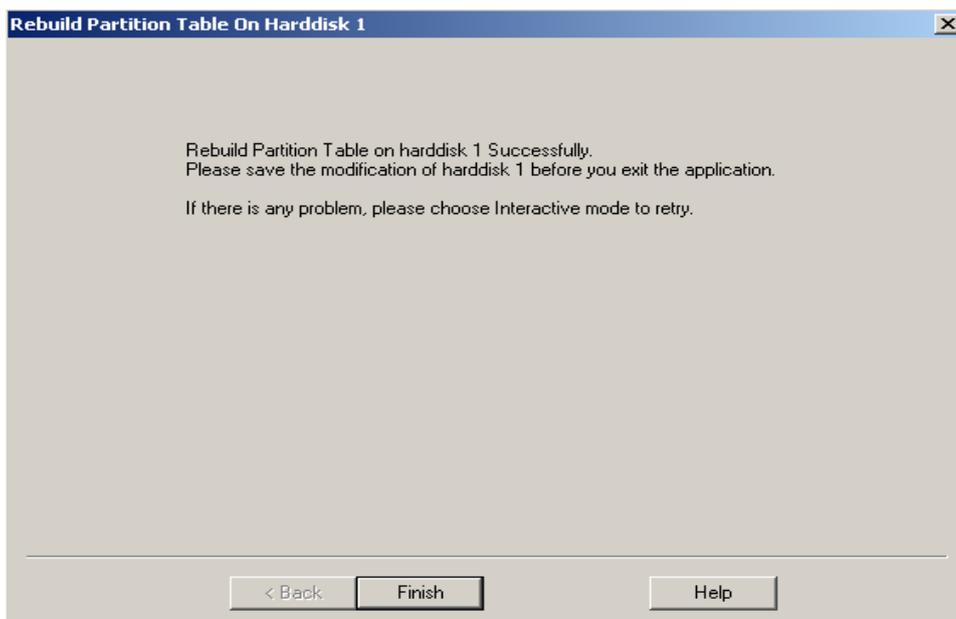
If you can enter operation system, you can install and run Partition Table Doctor and choose “Rebuild Partition Table”



In the procedure of rebuilding partition table, there are two ways to recover the lost partition:



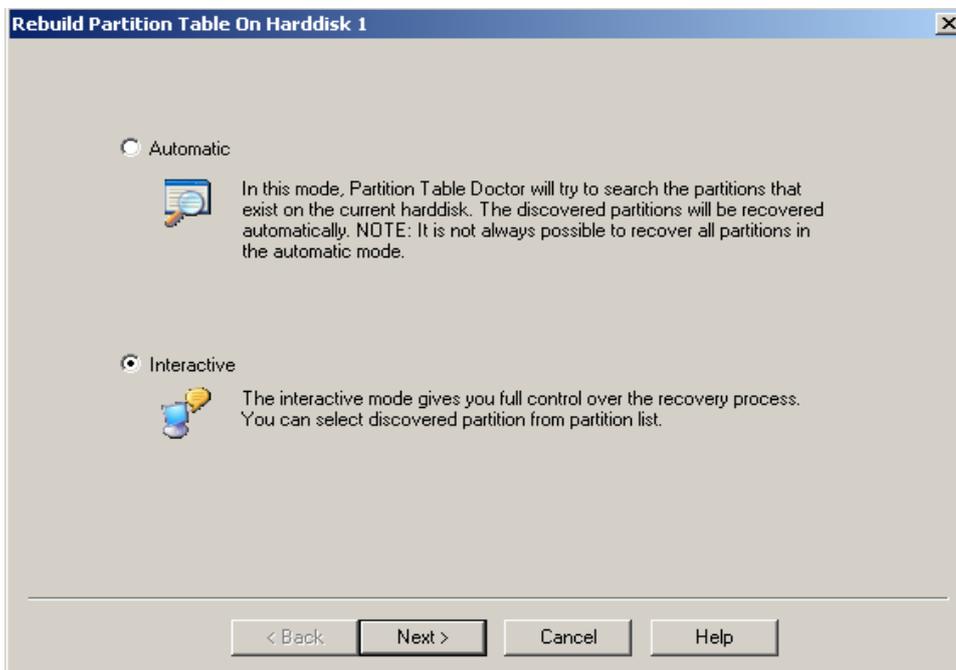
Here we mainly refer to “Automatic”, so we choose “Automatic” (after this, in the whole process the users can not operate by themselves, the software will finish the work automatically.)



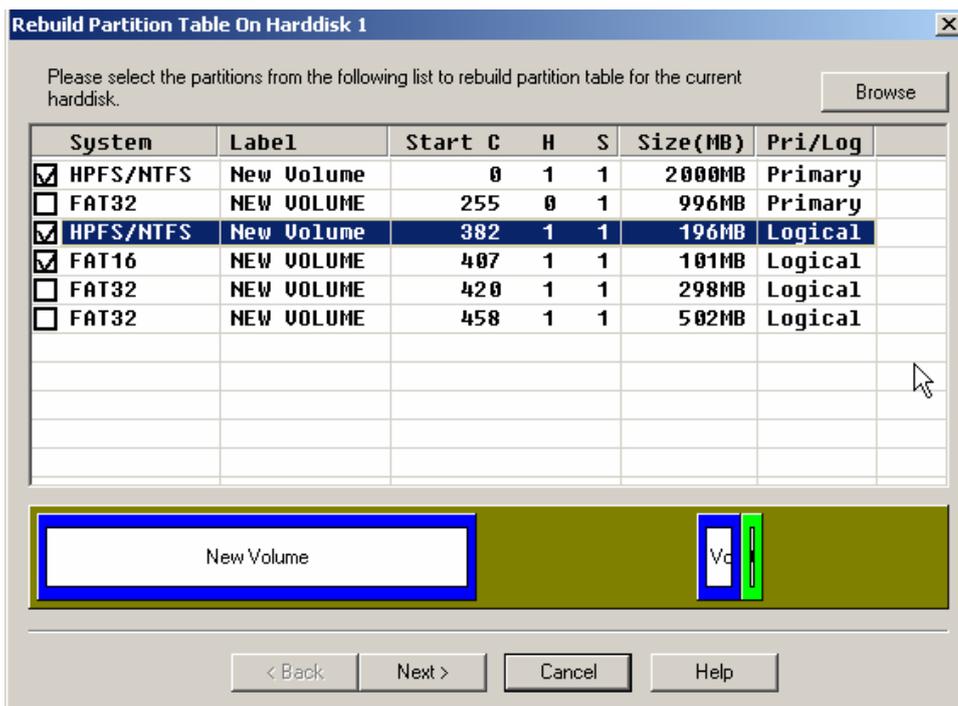
Automatic will automatically rebuild and recover partition according to your former partition information. Usually we suggest this mode if users do not know much about partition. Of course, if you are not satisfied with the result, you can choose “Interactive” mode.

Interactive

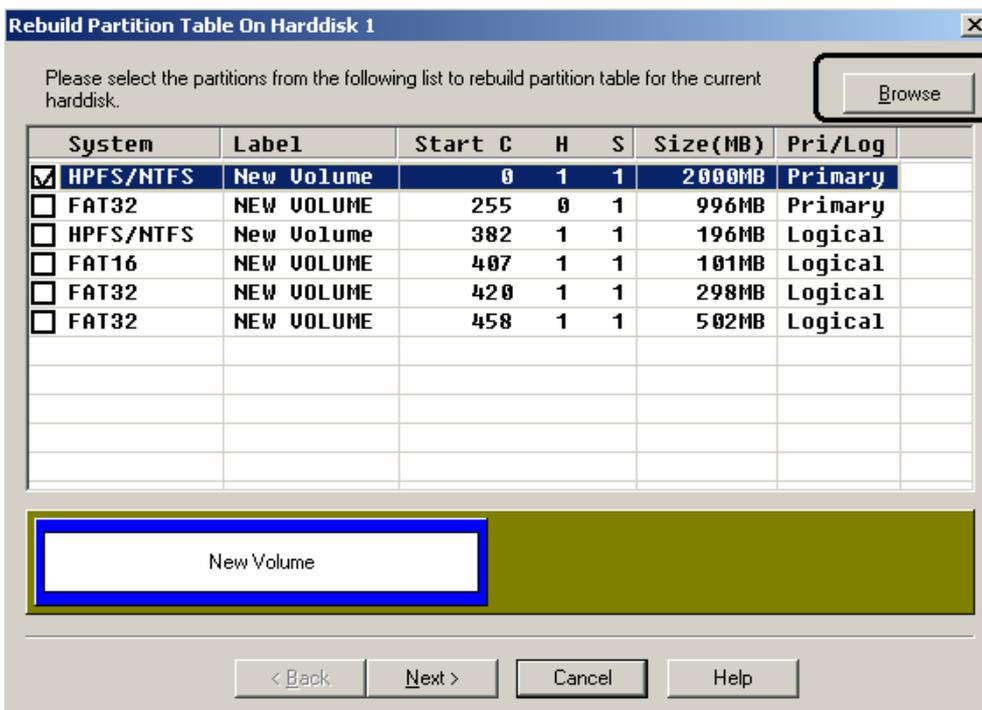
Run Partition Table Doctor and choose “Interactive” in “Rebuild Partition Table”.



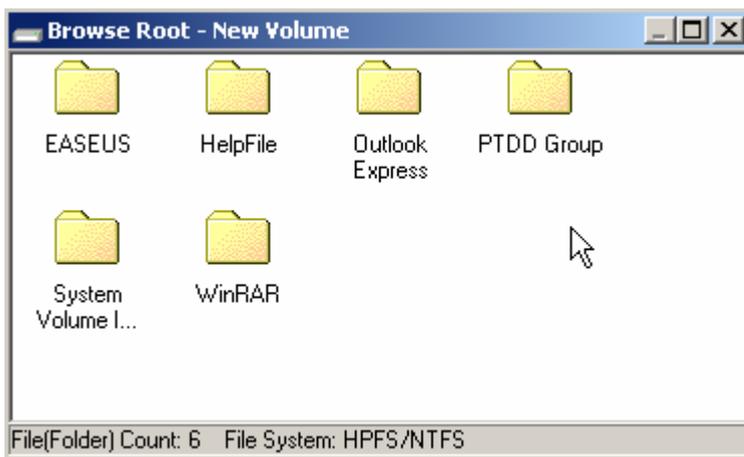
By this mode, Partition Table Doctor will display all the partition found, you can choose the partition you want to recover.



In the partition interface that Interactive mode has recovered, we can see clearly all the partition information that existed as well as information on file system, volume label, CHS, size and logics of the corresponding partition. With the information, user can locate the partition they want to recover. Also, to verify that whether the partition can be recovered normally and the recovered partition is the partition user want, the software provides a “Browse” function, with which we can choose the right partition:

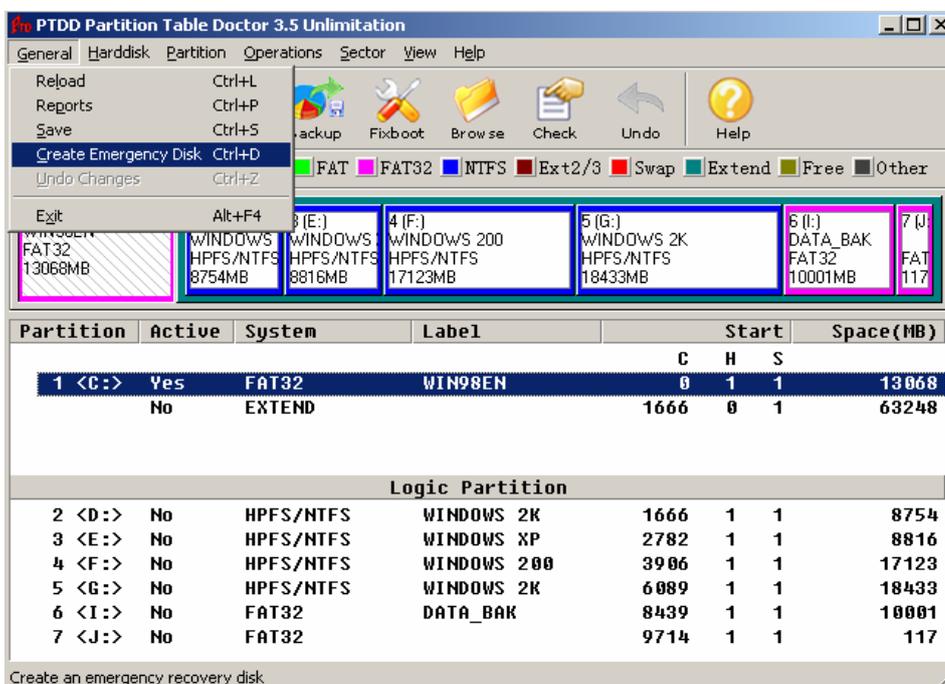


Click “Browse” and we can get file information of partition under root directory after recovery.



And then you can be surer about the accuracy and efficiency, thus know the final result.

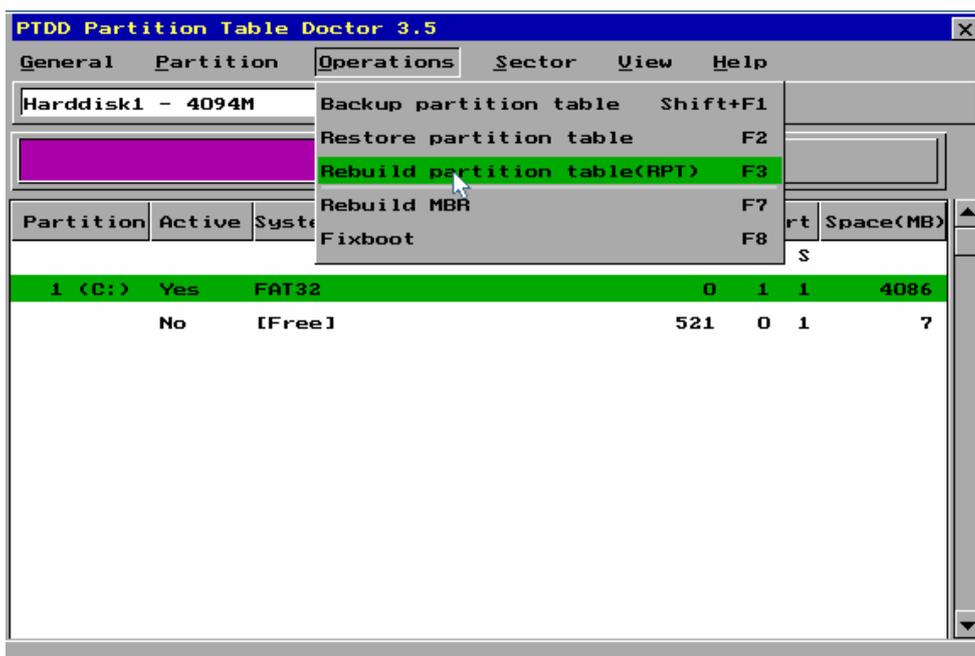
If the partition where your operation system locates is damaged and you cannot enter it, you can install Partition Table Doctor in another operation system,



By “Create Emergency Disk”, we can create an application under DOS to recover your partition:



Afterwards you can set your BIOS and then start from floppy disk, thus recover MBR by operation mode of Partition Table Doctor in DOS:



DBR recovery

MBR is for the whole hard disk, while DBR is for individual partition.

The first sector of each MBR is DBR. Just as MBR, DBR contains some information that the boot operation system need. If DBR is corrupted, you can neither visit the partition nor start up the operation system of the partition.

If boot sector is damaged, the possible symptoms are:

1. Invalid media type reading drive

2. Abort Retry Fail?
 3. File system is displayed as “RAW”
 4. Windows may ask if you want to format the drive
 5. File names contain “weird” characters
 6. “Sector not found” messages
- Etc.

Moreover, for partitions of NTFS, the functions of DBR are not all the same as that of FAT partition. For FAT partition, DBR locates FDT and FAT (correspondingly as well as DATA), but not verifying the correctness and reasonableness of FDT and FAT. For partition of NTFS, we need more units to load the file system, which is more complex than FAT.

What if when the DBR is destroyed? Usually, there are methods as following:

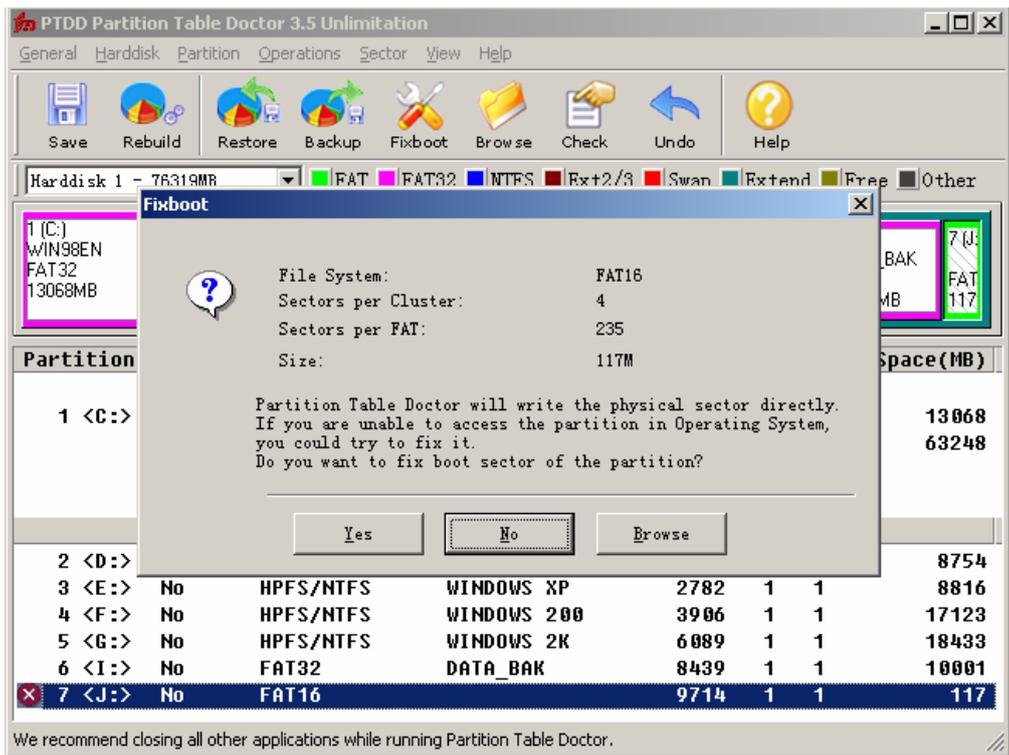
Recover DBR by Format

If there is no important data in this partition, or you have backed up the data, the best way to recover DBR is direct high-level format, fast format or complete format. If there is no limitation of partition form and capacity, there would be no difference between DOS format and Windows format except speed. Format is quite thorough, it can completely rearrange the data storage, even “reset” former file fragmentation.

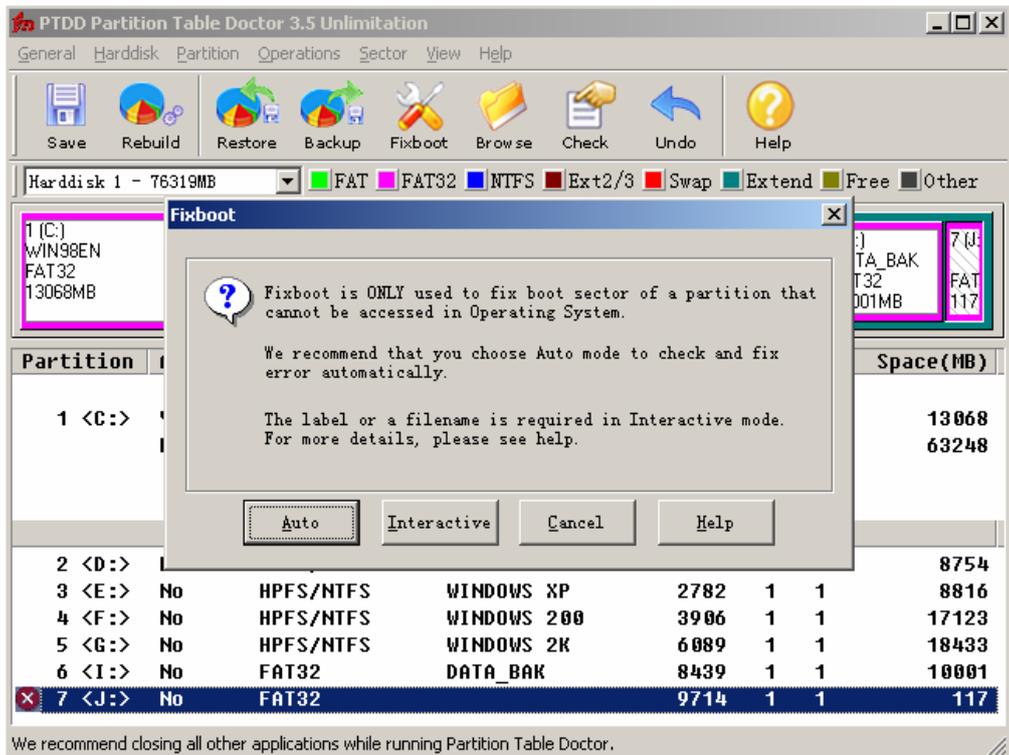
Although this method is simple, it cannot recover data actually especially if you choose some different parameters. If you choose different system reserved sectors, or use clusters of different size, or change the size of FAT table etc, data recovery will be more difficult.

Data recovery by Fixboot of **Partition Table Doctor**

If the boot sector of a Fat16/Fat32/Ntfs partition was corrupted, it will be marked with **X** by Partition Table Doctor. If you cannot access a Fat16/Ntfs partition and the partition was marked with **X**. Right click the partition and choose Fixboot. Partition Table Doctor will automatically check and restore the boot sector of the partition.



If you cannot access the Fat32 partition and the partition was marked with X Right click the partition and choose Fixboot, there will be two choices:



'Auto' mode: Partition Table Doctor will automatically check and restore the boot sector of the partition. We recommend you choose this mode. If 'Auto' mode cannot help you, you can choose 'Interactive' mode. If so, you need input the volume label or a file name (under the root directory).

If you do not know what file name to input, follow the file name that is suggested:

Boot partition:

io.sys msdos.sys ntldr bootlog.txt

Other partitions: _restore recycled

Note:

For Fat16/Fat32 partition, fixboot can effectively restore damaged boot sector of partition.

For NTFS partition, even if boot sector is correct but MFT (Main File Table) is corrupted, symptoms are the same. We recommend you download the demo version of Partition Table Doctor to determine whether boot sector of partition was corrupted.

Mostly, scandisk that originally in operation system will destroy more than they retrieve. Please stop scandisk after logging on.

In addition, you may use WinHex to recover DBR

WinHex is powerful in disk editor. With backup DBR in WinHex to recover the DBR sector is convenient and fast. But for its strong specialization of WinHex we recommend that you choose easy-to-use software tool for integrity and correctness of the data.

4.The FAT table recovery

CIH destroys data backwards from partitions. In this case, system data in the former part may be destroyed and lost. If FAT2 is still intact, we may make FAT2 to cover FAT1. Usually we use DiskEdit and WinHex. Regarding to other forms of destruction such as format and so on, we usually make use of tool software to scan the whole disk, seldom manual recovery; because there are even dozens of trillions sectors a partition has several trillions. Depending on the manual analysis is impossible. For some extremely important data file, we can also recover manually.

Recover FAT by DiskEdit

After recovering DBR of FAT, if part of FAT1 is damaged while FAT2 remains intact (It is the most situation when destroyed by CIH), we may use FAT2 to cover FAT1. The specific method is to find the start sector of FAT2 and then start searching the start sector of DATA (if it is FAT16, search FDT). By this way, we can figure out the length of FAT table. According to length and the start sector of FAT2, we may know the start sector of FAT1. Copy FAT2 to the damaged FAT1, we can finally recover the whole partition.

Recover FAT by WinHex

Principle of recovering FAT by WinHex is the same as that by DiskEdit. After recovering DBR, we can make FAT2 to cover FAT1. After finding FAT2, we begin searching the start sector of DATA (if it is FAT16, search FDT). The division is distinct, because the conclusion part of FAT must be 0 regions, otherwise there is not any free space (even so, in ordinary circumstances, there is still a bit of space in FAT after scanning DATA area. So the end of the last sector must be 0 too.). While at the beginning of DATA region or FDT region it mustn't be 0. No matter there is fixed

FDT, the system always begins from second cluster. If there is FDT, it follows closely FAT2, and its file registration must exist; if there is not, then begins from data area where some data must

exists. Thus we may figure out the length of the FAT table, and then the start sector of FAT1 according to the length and the start sector of FAT2. Copy FAT2 to the damaged FAT1 we can finally recover this partition.

Advanced

VII. FAT16 file system disk

FAT16, the same as FAT32, manages files with the cooperation of DBR, FDT and FAT. The directory entry of FAT16 and FAT32 are 32 bites. For Fat 32 is development of FAT16, the definition of 32 byte has been expanded. In addition, the FAT table of FAT16 denotes a cluster with 2 bytes, while FAT32 4 bytes, which is the difference between FAT16 and FAT32.

1.File management in root directory of FAT16

As we all know, file directory of FAT16 is 32 bytes, now let's take an example. We divide a partition of approximately 118M by system disk manager, and make each cluster of 2KB i.e. 4 sectors a cluster. Then in the root directory we create a text files whose BPB parameter are:

Offset	Title	Value
0	JMP instruction	EB 3C 90
3	OEM	MSDOS5.0
BIOS Parameter Block		
11	Bytes per sector	512
13	Sectors per cluster	4
14	Reserved sectors	1
16	Number of FATs	2
17	Root entries	512
19	Sectors (under 32 MB)	0
21	Media descriptor (hex)	F8
22	Sectors per FAT	235
24	Sectors per track	63
26	Heads	255
28	Hidden sectors	63
32	Sectors (over 32 MB)	240912
36	BIOS drive (hex, HD=8x	80
37	(Unused)	0
38	Ext. boot signature (29f	29
39	Volume serial number	1620538091
39	Volume serial number	EB 72 97 60
43	Volume label	NO NAME
54	File system	FAT16
510	Signature (55 AA)	55 AA

The name of the text file is file1.txt, its size is 12KB, and it covers 6 clusteres. Its contents are:

Notes for Using And Running

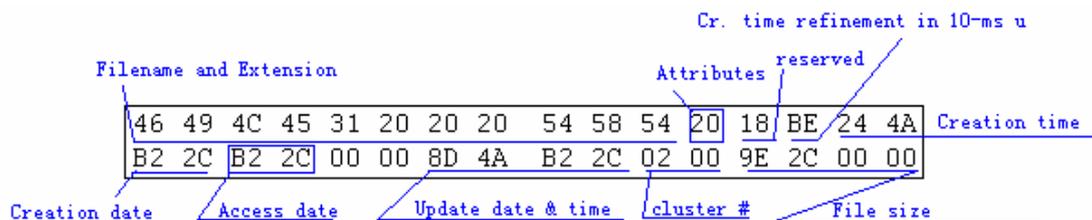
Partition Table Doctor is the only real software for Harddisk partitions recovery when you experience a drive error (other than hardware failure). This versatile tool automatically checks and repairs the Master Boot Record, partition table, and the boot sector of the partition with an error, to recover the FAT16/FAT32/NTFS/NTFS5/EXT2/EXT3/SWAP partition on IDE/ATA/SATA/SCSI Harddisk drives. Partition Table Doctor manages under MSDOS, FreeDOS, and Windows 95/98/Me, Windows NT 4.0, Windows 2000, Windows XP and Windows 2003.

Besides starting Partition Table Doctor from the Windows Start menu, you can also start it from inside Windows Explorer or My Computer. It can create an emergency floppy disk or a bootable CD to recover the bad partition even if your operating system fails to boot. When running, Partition Table Doctor will automatically check the Master Boot Record, partition table, and the boot sector of the partition. If the partition table on the Harddisk drive is damaged, Partition Table Doctor will ask you to Rebuild Partition Table for partition recovery on the Harddisk drive. If the boot sector of one partition was

It is opened by wordpad, the FDT of its partition is:

46 41 54 31 36 20 20 20	20 20 20 08 00 00 00 00	FAT16
00 00 00 00 00 00 97 49	B2 2C 00 00 00 00 00 00	格?.....
E5 B0 65 FA 5E 20 00 87	65 2C 67 0F 00 D2 87 65	灏e鵠 .隰,g..覈e	
63 68 2E 00 74 00 78 00	74 00 00 00 00 00 FF FF	ch..t.x.t.....	
E5 C2 BD A8 CE C4 7E 31	54 58 54 20 00 BE 24 4A	迓建文~1TXT .?J	
B2 2C B2 2C 00 00 25 4A	B2 2C 00 00 00 00 00 00	??..%J?.....	
46 49 4C 45 31 20 20 20	54 58 54 20 18 BE 24 4A	FILE1 TXT .?J	
B2 2C B2 2C 00 00 8D 4A	B2 2C 02 00 9E 2C 00 00	??..啤?..?..	

It is the content of file directory. The first directory entry is volume label whose attribute byte is 08H, which is 00001000B. Now we will analysis the file1.txt:



From this FDT entry, we may know that the size of file1.txt file have 11422 bytes, the start cluster is 02H. So, on one hand we may search file1.txt in the 2nd cluster of DATA; on the other hand we may search its successor cluster in FAT table. Take a look at the first step: make sure of logical sector of the 2nd cluster. From the BPB parameters block chart we may know, there reserved 1

sector, 2 FAT tables. The length of each FAT table have 235 sectors, FDT takes 32 sectors. So the start sector of DATA is: $1+235*2+32=503$, each cluster has 4 sectors, then the start sector of the 2nd cluster is: $503+(2-2)*4=503$. The text files contents are:

4E 6F 74 65 73 20 66 6F 72 20 55 73 69 6E 67 20	Notes for Using
41 6E 64 20 52 75 6E 6E 69 6E 67 0D 0A 0D 0A 50	And Running....P
61 72 74 69 74 69 6F 6E 20 54 61 62 6C 65 20 44	artition Table D
6F 63 74 6F 72 20 69 73 20 74 68 65 20 6F 6E 6C	ector is the onl
79 20 72 65 61 6C 20 73 6F 66 74 77 61 72 65 20	y real software
66 6F 72 20 48 61 72 64 64 69 73 6B 20 70 61 72	for Harddisk par
74 69 74 69 6F 6E 73 20 72 65 63 6F 76 65 72 79	titions recovery
20 77 68 65 6E 20 79 6F 75 20 65 78 70 65 72 69	when you experi
65 6E 63 65 20 61 20 64 72 69 76 65 20 65 72 72	ence a drive err
6F 72 20 28 6F 74 68 65 72 20 74 68 61 6E 20 68	or (other than h
61 72 64 77 61 72 65 20 66 61 69 6C 75 72 65 29	ardware failure)
2E 20 54 68 69 73 20 76 65 72 73 61 74 69 6C 65	. This versatile
20 74 6F 6F 6C 20 61 75 74 6F 6D 61 74 69 63 61	tool automatica
6C 6C 79 20 63 68 65 63 6B 73 20 61 6E 64 20 72	lly checks and r
65 70 61 69 72 73 20 74 68 65 20 4D 61 73 74 65	epairs the Maste
72 20 42 6F 6F 74 20 52 65 63 6F 72 64 2C 20 70	r Boot Record, p
61 72 74 69 74 69 6F 6E 20 74 61 62 6C 65 2C 20	artition table,
61 6E 64 20 74 68 65 20 62 6F 6F 74 20 73 65 63	and the boot sec
74 6F 72 20 6F 66 20 74 68 65 20 70 61 72 74 69	tor of the parti
74 69 6F 6E 20 77 69 74 68 20 61 6E 20 65 72 72	tion with an err
6F 72 2C 20 74 6F 20 72 65 63 6F 76 65 72 20 74	or, to recover t
68 65 20 46 41 54 31 36 2F 46 41 54 33 32 2F 4E	he FAT16/FAT32/N
54 46 53 2F 4E 54 46 53 35 2F 45 58 54 32 2F 45	TFS/NTFS5/EXT2/E
58 54 33 2F 53 57 41 50 20 70 61 72 74 69 74 69	XT3/SWAP partiti
6F 6E 20 6F 6E 20 49 44 45 2F 41 54 41 2F 53 41	on on IDE/ATA/SA

From the chart we can see, when every segment ends, it is marked with“OD 0A”, which means “enter” and “newline” The second step is to find cluster chain. The FAT is:

F8 FF FF FF 03 00 04 00 05 00 06 00 07 00 FF FF
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

In FDT, the start cluster of the file is 2, then the corresponding offset address of the 2nd cluster is: $2*2=4$, that is the two bytes from the offset 4, is the entrance of file file1.txt in FAT. These two bytes are "03 00", therefore the following file is saved in the 0003 cluster. We can seek the

content and the FAT chain of the file in offset 3*2=6.

From the FAT we may see, the place used to record cluster 0 and cluster 1 is "F8 FF FF FF", which is the fixed start mark of FAT table in FAT16 file system. The cluster number starts from 2, corresponding with its offset address. When computing the offset address, we can do nothing but to multiply 2.

Following this cluster chain, we can find "0700", that is the 7th cluster, which locates in 503+ (7-2)*4=523 sector; then we obtain "FF FF" in the FAT, which indicates the end of file. Hence we've found the whole file. For the last cluster, it has not been used up. The place where it ends can be calculated according to the file length. In FDT record, the size of file1.txt file is 11422 bytes, we may figure out 11422 DIV 512=22, therefore, the file actually takes 22 integer sectors and part of the 23rd sector, 503+22=525. The last sector of file is the 525th sector. 11422 MOD 512=158, so the last sector only uses 158 bytes, that is 158D=9E of hexadecimal. Therefore the last sector uses 0~ 9DH bytes.

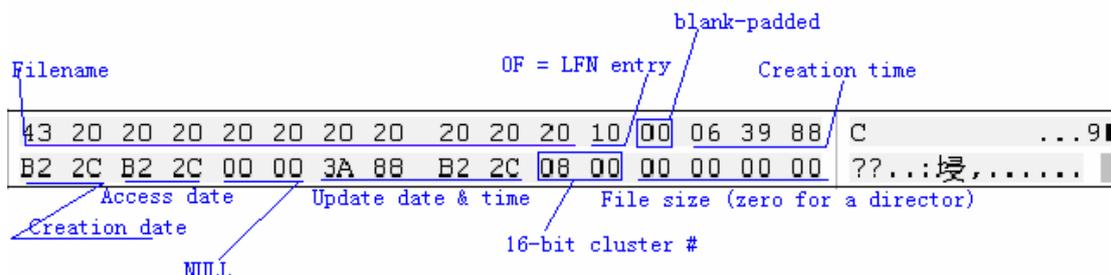
2. FAT16 sub-directory management

In root directory of logical driver, we may directly create files as well as primary directory, that is called the sub-directory under the root directory. Correspondingly, the root directory is called parental directory of the sub-directory. The sub-directory and the parental directory are relative, a parental directory may have many sub-directories, while a sub-directory has only one parental directory. In sub-directory, we can also create more sub-directories of lower level, thus forming the directory tree. For sub-directory, its entrance is still in root directory. The content of root directory FDT in sectors is:

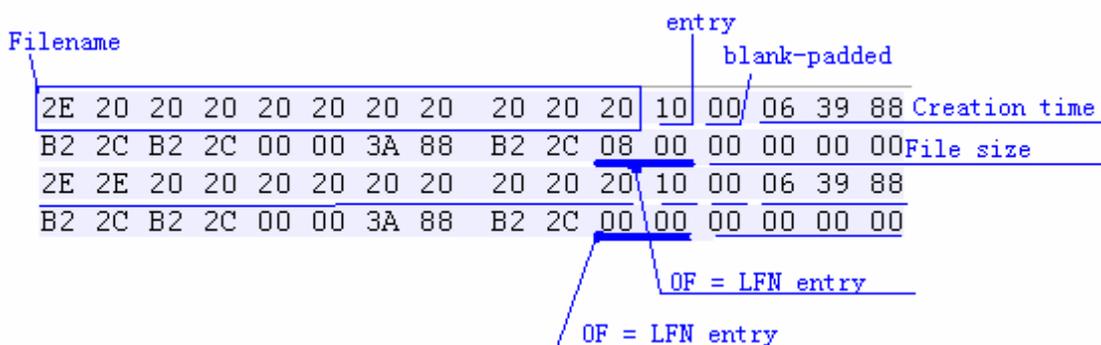
46 41 54 31 36 20 20 20 20 20 20 08 00 00 00 00	FAT16
00 00 00 00 00 00 3D 82 B2 2C 00 00 00 00 00 00=循,.....
31 32 33 20 20 20 20 20 20 20 20 10 00 92 91 82	123 ..捺
B2 2C B2 2C 00 00 92 82 B2 2C 02 00 00 00 00 00	??..攀?.....
46 49 4C 45 31 20 20 20 54 58 54 20 18 A5 D4 82	FILE1 TXT .匕
B2 2C B2 2C 00 00 D2 82 B2 2C 06 00 DD 2E 00 00	??..罍?..?..
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

The first directory registration entry is volume label whose attritugte byte (0BH byte) is 08H. The 2ndis the registration entries of the sub-directory "123". The 3th is the registration entries of the file "file1.txt".

Establish a "C" sub-directory under its root directory. The analysis chart of "C" file directory is:



According to the root directory folder chart, compared to the files in root directory, the expressive way of directory name and filename is the same; and so is that of start cluster except that of attribute and length. For folder, the fixed length is "0". Setting like this is for the convenience of system administration, otherwise, any change to the folder will affect the efficiency of management; moreover the size of directories can not define properly, it might conflict with the file length. As the same, like the root directory structure, the content storage in folder saves sperate file directory entry whose management is similar to that in root directory. But there are still some differences between sub-directory and the root directory, which lie in their own management of their own management: the root directory is determined by DBR, while sub-directory is determined by the directory registration entry under root directory. That is because these sub-directories are created by root directory. For sub-directory, it is necessary to discuss the first two sub directories as following:



The first directory registration is directory "." (2EH), the second is directory ".." (2E2EH); actually "." is on behalf of current directory, while ".." parental directory.

3. File deletion in FAT16

Many people believe that when files are deleted, the system will clear all the content of the deleted files, that is to write "0" to corresponsgding blocks of the disk. The actual situation is different. Think about that, how much time we shall take to delete a large file (for example several hundred MB files)! The system working efficiency will turn to be very low.

In fact, when deleting files, system just makes a deletion mark on this file and marks the the clusters they cover in FAT table as "empty", while in the DATA region clusters still preserved the original file content. Certainly, underoperating operating system, you cannot see the files with

deletion marks without the help of special software or program. For the operating system, the files are "truly" deleted! And when writing data again to the disk, the system probably may cover these original files, that are to write new information to cover the blocks of deleted files.

From Windows 95 operating systems, there is a Recycle Bin in the Windows system. In fact, the Recycle Bin is only some space on the hard disk; the Windows system automatically establishes a folder "RECYCLED" (under root directory of disk's each partition) with hiding attribute to save temporarily deleted files. Only when more deleting or executing "Clear" command, these files then can be completely deleted (as to operating system). As "Recycle Bin" we see on the desktop, it is only a shortcut. Then we will introduce temporary deletion and complete deletion separately.

In FAT16 (Windows 98), we simply delete some files, that is put it to the Recycle Bin, in this situation files are recoverable. So, let's have a look at that for FDT, FAT and the DATA, what is the difference between before and after deletion.

Before deleting long file~.txt, the FDT is as the following chart:

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Access
000241152	46	41	54	31	36	20	20	20	20	20	08	00	00	00	00	00	FAT16
000241168	00	00	00	00	00	00	23	88	B2	2C	00	00	00	00	00	00#
000241184	46	49	4C	45	31	20	20	20	54	58	54	20	18	8B	2B	88	FILE1 TXT
000241200	B2	2C	B2	2C	00	00	D2	82	B2	2C	02	00	DD	2E	00	00	??..
000241216	43	20	20	20	20	20	20	20	20	20	10	00	06	39	88		C
000241232	B2	2C	B2	2C	00	00	3A	88	B2	2C	08	00	00	00	00	00	??..:
000241248	54	55	52	42	4F	43	32	20	20	20	10	08	07	39	88		TURBOC2
000241264	B2	2C	B2	2C	00	00	3A	88	B2	2C	09	00	00	00	00	00	??..:
000241280	43	74	00	00	00	FF	FF	FF	FF	FF	FF	0F	00	D4	FF	FF	Ct...
000241296	FF	00	00	FF	FF	FF	FF										
000241312	02	65	00	6C	00	6F	00	6E	00	67	00	0F	00	D4	20	00	.e.l.o.n.g...?
000241328	66	00	69	00	6C	00	65	00	2E	00	00	00	74	00	78	00	f.i.l.e.....t.x.
000241344	01	6C	00	6F	00	6E	00	67	00	20	00	0F	00	D4	66	00	.l.o.n.g. ...
000241360	69	00	6C	00	65	00	20	00	6E	00	00	00	61	00	6D	00	i.l.e.a.m.
000241376	4C	4F	4E	47	46	49	7E	31	54	58	54	20	00	46	5D	88	LONGFI~1TXT .F]
000241392	B2	2C	B2	2C	00	00	6C	88	B2	2C	0A	00	5C	04	00	00	??..:1
000241408	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000241424	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000241440	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000241456	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000241472	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000241488	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

After deleting long file~.txt, the FDT is as the following chart:

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Access
000241152	46	41	54	31	36	20	20	20	20	20	20	08	00	00	00	00	FAT16
000241168	00	00	00	00	00	00	23	88	B2	2C	00	00	00	00	00	00#壞,.....
000241184	46	49	4C	45	31	20	20	20	54	58	54	20	18	8B	2B	88	FILE1 TXT ?!
000241200	B2	2C	B2	2C	00	00	D2	82	B2	2C	02	00	DD	2E	00	00	??..??..??..??..
000241216	43	20	20	20	20	20	20	20	20	20	20	10	00	06	39	88	C ...9!
000241232	B2	2C	B2	2C	00	00	3A	88	B2	2C	08	00	00	00	00	00	??..:壞,.....!
000241248	54	55	52	42	4F	43	32	20	20	20	20	10	08	07	39	88	TURBOC2 ...9!
000241264	B2	2C	B2	2C	00	00	3A	88	B2	2C	09	00	00	00	00	00	??..:壞,.....9!
000241280	E5	74	00	00	00	FF	FF	FF	FF	FF	FF	0F	00	D4	FF	FF	繞...
000241296	FF	00	00	FF	FF	FF	FF										
000241312	E5	65	00	6C	00	6F	00	6E	00	67	00	0F	00	D4	20	00	錯.l.o.n.g...?..
000241328	66	00	69	00	6C	00	65	00	2E	00	00	00	74	00	78	00	f.i.l.e.....t.x.
000241344	E5	6C	00	6F	00	6E	00	67	00	20	00	0F	00	D4	66	00	錄.o.n.g. ...診.
000241360	69	00	6C	00	65	00	20	00	6E	00	00	00	61	00	6D	00	i.l.e.a.m.
000241376	E5	4F	4E	47	46	49	7E	31	54	58	54	20	00	46	5D	88	鏽NGFI~1TXT .F]!
000241392	B2	2C	B2	2C	00	00	6C	88	B2	2C	0A	00	5C	04	00	00	??..!壞,..\...!
000241408	41	52	00	65	00	63	00	79	00	63	00	0F	00	21	6C	00	AR.e.c.y.c...!l.
000241424	65	00	64	00	00	00	FF	FF	FF	FF	00	00	FF	FF	FF	FF	e.d... ..
000241440	52	45	43	59	43	4C	45	44	20	20	20	16	00	20	78	48	RECYCLED .. xH
000241456	B3	2C	B3	2C	00	00	79	48	B3	2C	11	00	00	00	00	00	??..yH?.....!l.

After deleting the long file~. txt, in the file directory, only the first byte is changed into "E5H", the others are not. And so is the long filename, all the first byte of registration entries that describe long file name are changed into "E5H". This "E5H" means this file was deleted. The operating system would consider it already "has not existed". When the user requests this file, the operating system will tell that this file does not exist.

And the corresponding FAT would not be changed. In fact, after the file is deleted, the space the file covers is not really "released", it also exists in DATA. Let's have a look at the corresponding DATA before and after deletion:

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Access
000273920	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	1111111111111111
000273936	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	1111111111111111
000273952	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	1111111111111111
000273968	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	1111111111111111
000273984	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	1111111111111111
000274000	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	1111111111111111
000274016	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	1111111111111111
000274032	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	1111111111111111
000274048	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	1111111111111111
000274064	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	1111111111111111
000274080	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	1111111111111111
000274096	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	1111111111111111
000274112	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	1111111111111111
000274128	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	1111111111111111
000274144	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	1111111111111111
000274160	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	1111111111111111

In addition, the recycling bin is also changed. Because we do not delete any file or directory in the logical disk, there is no sub-direcotry of Recycle Bin on the partition. When deleteing this file, the system automatically establishes Recycle Bin sub-directory, which FDT content is as following chart:

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Access
000241248	54	55	52	42	4F	43	32	20	20	20	10	08	07	39	88		TURBOC2 ...9
000241264	B2	2C	B2	2C	00	00	3A	88	B2	2C	09	00	00	00	00		??...:煖,.....
000241280	E5	74	00	00	00	FF	FF	FF	FF	FF	FF	0F	00	D4	FF		統...
000241296	FF	00	00	FF	FF	FF											
000241312	E5	65	00	6C	00	6F	00	6E	00	67	00	0F	00	D4	20		錯.l.o.n.g...?.
000241328	66	00	69	00	6C	00	65	00	2E	00	00	00	74	00	78		f.i.l.e.....t.x.
000241344	E5	6C	00	6F	00	6E	00	67	00	20	00	0F	00	D4	66		錶.o.n.g. ...諗.
000241360	69	00	6C	00	65	00	20	00	6E	00	00	00	61	00	6D		i.l.e. .n...a.m.
000241376	E5	4F	4E	47	46	49	7E	31	54	58	54	20	00	46	5D		繼NGFI~1TXT .F]■
000241392	B2	2C	B2	2C	00	00	6C	88	B2	2C	0A	00	5C	04	00		??..l煖,..\...
000241408	41	52	00	65	00	63	00	79	00	63	00	0F	00	21	6C		AR.e.c.y.c...ll.
000241424	65	00	64	00	00	00	FF	FF	FF	FF	00	00	FF	FF	FF		e.d... ..
000241440	52	45	43	59	43	4C	45	44	20	20	20	16	00	20	78		RECYCLED .. xH
000241456	B3	2C	B3	2C	00	00	79	48	B3	2C	11	00	00	00	00		??..yH?.....
000241472	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	

When file registration entry of the deleted file in FDT has been marked with deletion (first letter change to E5H), in Recycle Bin. It also makes an indicator that pointer to the first cluster of the deleted file. Of course, only these information are not enough. When we restore the deleted files to their original place, we can see much more information. Or you cannot make choice when restoring.

The information is recorded in file INFO2, the reader who is interested may further analyze. If it is complete deletion, there will be no file INFO2 in system. Then what about after restoring?

After files in Recycle Bin are restored, the file register entries are correspondingly changed as "E5H" (deletion status), and files in FDT are returned to original status.

We can understand this deletion process like this: when the system is deleting files, it moves the files to the Recycle Bin folder; then it overwrites the original FDT registration to "deletion" as well as establishes a file similar to the log in the recycling bin folder, recording related operation information; By this way, we may know some necessary information when we refer to the Recycle Bin.

4. Contents deletion in FAT16

Simple deletion of sub-directory in FAT16 is only to make a deletion mark to FDT that describes sub-directory. All files and all records of sub-directories are not been changed, just as we "move" the sub-directory to the Recycle Bin.

5. Fast high level format in FAT16 partition

Actually quick advanced format is completely deleting all files and sub-directories, for there will not any more file or directory record in FDT and FAT, thus releasing the whole disk space.

Now let's have a look at the changes of FDT and FAT because of the advanced format. Before the

quick format, the FDT content is as following chart:

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Access	
000241152	46	41	54	31	36	20	20	20	20	20	20	08	00	00	00	00	FAT16	
000241168	00	00	00	00	00	00	23	88	B2	2C	00	00	00	00	00	00#...	
000241184	46	49	4C	45	31	20	20	20	54	58	54	20	18	8B	2B	88	FILE1 TXT ?!	
000241200	B2	2C	B2	2C	00	00	D2	82	B2	2C	02	00	DD	2E	00	00	??..??...9!	
000241216	43	20	20	20	20	20	20	20	20	20	20	10	00	06	39	88	C ...9!	
000241232	B2	2C	B2	2C	00	00	3A	88	B2	2C	08	00	00	00	00	00	??..:...	
000241248	54	55	52	42	4F	43	32	20	20	20	20	10	08	07	39	88	TURBOC2 ...9!	
000241264	B2	2C	B2	2C	00	00	3A	88	B2	2C	09	00	00	00	00	00	??..:...	
000241280	E5	74	00	00	00	FF	FF	FF	FF	FF	FF	0F	00	D4	FF	FF	繞...	
000241296	FF	00	00	FF	FF	FF	FF											
000241312	E5	65	00	6C	00	6F	00	6E	00	67	00	0F	00	D4	20	00	錯.l.o.n.g...?..	
000241328	66	00	69	00	6C	00	65	00	2E	00	00	0F	00	74	00	78	00	f.i.l.e.....t.x.
000241344	E5	6C	00	6F	00	6E	00	67	00	20	00	0F	00	D4	66	00	錄.o.n.g. ...診.	
000241360	69	00	6C	00	65	00	20	00	6E	00	00	00	61	00	6D	00	i.l.e.a.m.	
000241376	E5	4F	4E	47	46	49	7E	31	54	58	54	20	00	46	5D	88	鎚NGFI~1TXT .F]!	
000241392	B2	2C	B2	2C	00	00	6C	88	B2	2C	0A	00	5C	04	00	00	??..!...	
000241408	41	52	00	65	00	63	00	79	00	63	00	0F	00	21	6C	00	AR.e.c.y.c...!l.	
000241424	65	00	64	00	00	00	FF	FF	FF	FF	00	00	FF	FF	FF	FF	e.d... ..	
000241440	52	45	43	59	43	4C	45	44	20	20	20	16	00	20	78	48	RECYCLED .. xH.	
000241456	B3	2C	B3	2C	00	00	79	48	B3	2C	11	00	00	00	00	00	??..yH?.....)M	
000241472	41	52	00	65	00	6C	00	65	00	61	00	0F	00	1B	73	00	AR.e.l.e.a....s.	
000241488	65	00	00	00	FF	FF	FF	FF	FF	FF	00	00	FF	FF	FF	FF	e...	
000241504	52	45	4C	45	41	53	45	20	20	20	20	10	00	30	7C	4D	RELEASE ..0 M	
000241520	B3	2C	B3	2C	00	00	7D	4D	B3	2C	0A	00	00	00	00	00	³,³,...}M³,.....	

After quick advanced format:

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Access
000241152	46	41	54	31	36	20	20	20	20	20	20	08	00	00	00	00	FAT16
000241168	00	00	00	00	00	00	00	D0	4D	B3	2C	00	00	00	00	00壘?.....
000241184	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000241200	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000241216	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000241232	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000241248	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000241264	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000241280	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000241296	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000241312	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000241328	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000241344	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000241360	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000241376	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000241392	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000241408	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000241424	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000241440	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000241456	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000241472	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000241488	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000241504	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000241520	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

After the quick advanced format, the system will clear all directory registration entries including recycle bin except a volume label (there is no data, naturally no deleted file or directory; recycling

bin is unnecessary to exist; only for the first time simple deletion is executed, system will automatically establish this directory).

Correspondingly, before the quick advanced format the FAT content is as the following chart:

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Access
000000512	F8	FF	FF	FF	03	00	04	00	05	00	06	00	07	00	08	00	?
000000528	09	00	0A	00	0B	00	0C	00	0D	00	0E	00	0F	00	10	00
000000544	11	00	FF	FF	13	00	14	00	15	00	16	00	17	00	18	00
000000560	19	00	1A	00	1B	00	1C	00	1D	00	1E	00	1F	00	20	00
000000576	21	00	22	00	23	00	24	00	25	00	26	00	27	00	28	00	!.\"#\$%&'(.)*.+,.-.../0. 1.2.3.4.5.6.7.8. 9.:;.<.=.>?.@. A.B.C.D.E.F.G.H. I.J.K.L.M.N.O.P. Q.R.S.T.U.V.W.X. Y.Z.[.\.].^._.`. a.b.c.d.e.f.g.h. i.j.k.l.m.n.o.p. q.r.s.t.u.v.w.x. y.z.{. .}.~.!.!. ???????? .o.p. ?????????u.v.w.x. ?????????}.~.!.!. ????????? ????????? ????????? B1 00 B2 00 B3 00 FF FF B5 00 B6 00 B7 00 B8 00 ???? ???? B9 00 BA 00 BB 00 BC 00 BD 00 BE 00 BF 00 C0 00 ????????? C1 00 C2 00 C3 00 C4 00 C5 00 C6 00 C7 00 C8 00 ????????? C9 00 CA 00 CB 00 CC 00 CD 00 CE 00 CF 00 D0 00 ????????????? D1 00 D2 00 D3 00 D4 00 D5 00 D6 00 D7 00 D8 00 ?????????

After quick advanced format, the content of FAT is as following:

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Access
000000512	F8	FF	FF	FF	00	00	00	00	00	00	00	00	00	00	00	00	?
000000528	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000544	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000560	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000576	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000592	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000608	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000624	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000640	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000656	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000672	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000688	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000704	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000720	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000736	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000752	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000768	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000784	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000800	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000816	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000832	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000848	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000864	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000880	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000896	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000912	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

We can see FAT is completely cleared. If files are saved incontinuously, it is very difficult to be recovered.

Then, after quick advanced format, will the content of sub-directory exist yet? After quick advanced format, FAT will definitely be zero, but file directory entrys of the sub-directory are reserved. For example, the following chart is the content of sub-directory after quick advanced format:

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Access
000259072	14	00	10	00	0A	00	01	00	69	00	0F	00	03	00	00	00i.....
000259088	03	00	00	00	00	00	46	00	00	40	F1	FF	02	00	46	00F..@?...F..
000259104	0C	04	02	00	63	6B	87	65	00	00	0B	00	00	00	03	24	...ck喫.....\$
000259120	03	31	24	00	61	24	03	00	24	00	43	4A	15	00	4B	48	.1\$.a\$...\$.CJ..KH
000259136	02	00	50	4A	03	00	5F	48	01	04	61	4A	18	00	6D	48	..PJ.._H...aJ..mH
000259152	09	04	6E	48	04	08	73	48	09	04	74	48	04	08	00	00	..nH...sH..tH....
000259168	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000259184	1C	00	41	40	F2	FF	A1	00	1C	00	0C	05	06	00	D8	9E	..A@??.....貫mH
000259200	A4	8B	B5	6B	3D	84	57	5B	53	4F	00	00	00	00	00	00	袖=刳[S0.....
000259216	00	00	00	00	00	00	24	00	4C	40	01	00	02	00	24	00\$.L@....\$.
000259232	0C	04	02	00	E5	65	1F	67	00	00	0E	00	0F	00	0F	84	...錯.g.....!
000259248	64	00	56	44	C4	09	5E	84	64	00	00	00	00	00	00	00	d.VD?^刳.....
000259264	7D	00	00	00	1E	00	00	0C	00	00	01	00	FF	FF	FF	FF	}.....
000259280	00	00	00	00	65	00	00	00	7F	00	00	00	9A	40	00	00	...e...! ...歆..
000259296	00	30	00	00	00	00	00	00	00	80	00	00	00	80	98	40	.0.....! ...! 檣
000259312	00	00	00	00	00	00	00	00	00	00	00	80	00	00	00	80! ...!
000259328	00	04	00	00	FA	04	00	00	03	00	00	00	00	04	00	00?.....
000259344	FA	04	00	00	04	00	00	00	00	04	00	00	FA	04	00	00	?.....?..?..檣
000259360	05	00	00	00	0F	00	00	F0	6C	00	00	00	00	00	06	F0饅.....!
000259376	18	00	00	00	02	04	00	00	02	00	00	00	01	00	00	00
000259392	01	00	00	00	01	00	00	00	02	00	00	00	1F	00	01	F0!
000259408	2C	00	00	00	52	00	07	F0	24	00	00	00	05	05	59	1C	...R..?.....Y..!
000259424	2D	EB	12	A0	55	6E	6D	99	2B	C8	0C	D9	72	93	FF	00	-?燐nm??賠?.....
000259440	78	0F	00	00	01	00	00	00	1E	0C	00	00	00	00	37	0A	x.....7..
000259456	40	00	1E	F1	10	00	00	00	FF	FF	00	00	00	00	FF	00	@.?.?... ..

From the chart we may see, after quick advanced format, contents under sub-directory are not deleted; but there is no entrance record in FDT and FAT, so, for operating system, they do not exist. Here we call out the contents of the cluster making use of some tool software to record the clusters where they were before format; at the same time we may see the cluster of contents is not covered. However, if we do not back up before format, how do we know in which clusters they are stored? And how can we explain the contents of the cluster? So here we call out the past record, but this does not mean we know their condition before the format. Actually, for operating system, quick advanced format itself can not be reversed.

6.Full advanced format in FAT16 partition

After full advanced format, FDT and FAT of root directory definitely reset. Now let's refer to the difference between full format and quick format in processing sub-directory and data region. Firstly, we create a file in this partition named "ABC", whose contents are as followingchart:

Filename	Ext.	Size	Created	Modified	Accessed	Attr.
Deleted Objects						
Abc		2.0 KB	05-19-2002 10:0...	05-19-2002 10:0...	05-19-2002	

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Access
000257536	2E	20	20	20	20	20	20	20	20	20	20	10	00	55	30	51UOQ
000257552	B3	2C	B3	2C	00	00	31	51	B3	2C	02	00	00	00	00	00	^,^,..1Q^,.....
000257568	2E	2E	20	20	20	20	20	20	20	20	20	10	00	55	30	51UOQ
000257584	B3	2C	B3	2C	00	00	31	51	B3	2C	00	00	00	00	00	00	^,^,..1Q^,.....
000257600	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000257616	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000257632	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000257648	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000257664	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000257680	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000257696	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000257712	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000257728	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

After full format, the content of “ABC” is:

Filename	Ext.	Size	Created	Modified	Accessed	Attr.
Deleted Objects						
Abc		2.0 KB	05-19-2002 10:0...	05-19-2002 10:0...	05-19-2002	

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Access
000257536	2E	20	20	20	20	20	20	20	20	20	20	10	00	55	30	51UOQ
000257552	B3	2C	B3	2C	00	00	31	51	B3	2C	02	00	00	00	00	00	^,^,..1Q^,.....
000257568	2E	2E	20	20	20	20	20	20	20	20	20	10	00	55	30	51UOQ
000257584	B3	2C	B3	2C	00	00	31	51	B3	2C	00	00	00	00	00	00	^,^,..1Q^,.....
000257600	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000257616	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000257632	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000257648	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000257664	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000257680	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000257696	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000257712	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000257728	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Obviously, neither of full format and quick format changes the contents of sub-directory.

In addition,full format doesn't change the file contents.

7.Searching files in FAT16 partition

Let's see a boot sector (start sector is DBR) in FAT16 after advanced format. The OEM mark is MSWIN5.0.

79900402176	EB 3C 90 4D 53 44 4F 53	35 2E 30 00 02 04 01 00	?恣SDOS5.0.....
79900402192	02 00 02 00 00 F8 EB 00	3F 00 FF 00 3F 00 00 00 ?. ?..
79900402208	10 AD 03 00 80 00 29 3A	E2 60 10 4E 4F 20 4E 41	.?.!.):鈇.NO NA
79900402224	D 45 20 20 20 20 46 41	54 31 36 20 20 20 33 C9	ME FAT16 3I
79900402240	8E D1 BC F0 7B 8E D9 B8	00 20 8E C0 FC BD 00 7C	幘捺{仄? 幫 .
79900402256	38 4E 24 7D 24 8B C1 99	E8 3C 01 72 1C 83 EB 3A	8N\$}\$嬰樂<.r.泮:
79900402272	66 A1 1C 7C 26 66 3B 07	26 8A 57 FC 75 06 80 CA	f? &f;.&奧黓.!!
79900402288	02 88 56 02 80 C3 10 73	EB 33 C9 8A 46 10 98 F7	.均.!?s?蓓F.椽
79900402304	66 16 03 46 1C 13 56 1E	03 46 0E 13 D1 8B 76 11	f..F..V..F..禱v.
79900402320	60 89 46 FC 89 56 FE B8	20 00 F7 E6 8B 5E 0B 03	`煨鼓V .塵輝..
79900402336	C3 48 F7 F3 01 46 FC 11	4E FE 61 BF 00 00 E8 E6	禽冀.F?N??.桃
79900402352	00 72 39 26 38 2D 74 17	60 B1 0B BE A1 7D F3 A6	.r9&8-t.`?尽}螃
79900402368	61 74 32 4E 74 09 83 C7	20 3B FB 72 E6 EB DC A0	at2Nt.∆ ;鹤棋軒
79900402384	FB 7D B4 7D 8B F0 AC 98	40 74 0C 48 74 13 B4 0E	鹽破嬌瑯@t.Ht.?
79900402400	BB 07 00 CD 10 EB EF A0	FD 7D EB E6 A0 FC 7D EB	??.膈胆}膾}I
79900402416	E1 CD 16 CD 19 26 8B 55	1A 52 B0 01 BB 00 00 E8	鬼.?&媮.R??.
79900402432	3B 00 72 E8 5B 8A 56 24	BE 0B 7C 8B FC C7 46 F0	;.r鑿葵\$? 嫩苻I
79900402448	3D 7D C7 46 F4 29 7D 8C	D9 89 4E F2 89 4E F6 C6	=}苻?}屬塏驂N釜
79900402464	06 96 7D CB EA 03 00 00	20 0F B6 C8 66 8B 46 F8	.枚岁... .度f婢I
79900402480	66 03 46 1C 66 8B D0 66	C1 EA 10 EB 5E 0F B6 C8	f.F.f嫩f陵.臟.度
79900402496	4A 4A 8A 46 0D 32 E4 F7	E2 03 46 FC 13 56 FE EB	JJ查.2敝?F?V
79900402512	4A 52 50 06 53 6A 01 6A	10 91 8B 46 18 96 92 33	JRP.Sj.j.悽F.料3
79900402528	D2 F7 F6 91 F7 F6 42 87	CA F7 76 1A 8A F2 8A E8	吟鯨黯B噶鱗.姦婉
79900402544	C0 CC 02 0A CC B8 01 02	80 7E 02 0E 75 04 B4 42	捞..谈...!~..u.蓓
79900402560	8B F4 8A 56 24 CD 13 61	61 72 0B 40 75 01 42 03	嫻葵\$?aar.@u.B.
79900402576	5E 0B 49 75 06 F8 C3 41	BB 00 00 60 66 6A 00 EB	^.Iu. A?.`fj.I
79900402592	B0 4E 54 4C 44 52 20 20	20 20 20 20 0D 0A 4E 54	瘡TLDR ..NT
79900402608	4C 44 52 20 69 73 20 6D	69 73 73 69 6E 67 FF 0D	LDR is missing .
79900402624	0A 44 69 73 6B 20 65 72	72 6F 72 FF 0D 0A 50 72	.Disk error ..Pr
79900402640	65 73 73 20 61 6E 79 20	6B 65 79 20 74 6F 20 72	ess any key to r
79900402656	65 73 74 61 72 74 0D 0A	00 00 00 00 00 00 00 00	estart.....
79900402672	00 00 00 00 00 00 00 00	00 00 00 AC BF CC 55 AA 菴I

From the chart we can see, the parameters of file system are all record in BPB parameter table of DBR; with the parameter table, system can locate FAT, FDT and DATA. The meaning of the parameters is shown in the following chart:

Another way to express:

Offset	Title	Value
0	JMP instruction	EB 3C 90
3	OEM	MSDOS5.0
BIOS Parameter Block		
11	Bytes per sector	512
13	Sectors per cluster	4
14	Reserved sectors	1
16	Number of FATs	2
17	Root entries	512
19	Sectors (under 32 MB)	0
21	Media descriptor (hex)	F8
22	Sectors per FAT	235
24	Sectors per track	63
26	Heads	255
28	Hidden sectors	63
32	Sectors (over 32 MB)	240912
36	BIOS drive (hex, HD=8x)	80
37	(Unused)	0
38	Ext. boot signature (29h)	29
39	Volume serial number (decimal)	274784826
39	Volume serial number (hex)	3A E2 60 10
43	Volume label	NO NAME
54	File system	FAT16
510	Signature (55 AA)	55 AA

We have known that system can divide all sectors located in the same cylinder into a partition with cylinder as the division surface in partition. System manages partition in 5 regions. There is no special rule between the 5 regions and partition, thus the number of DATA sectors is probably not a integer; in other words, when executing the cluster managment of DATA, the last several sectors do not rightly form a cluster. So the system can not execute cluster management to these secotrs. By WinHex, we call them residual sectors.Now let’s see how system divide and organize the 5 regions:

Firstly, when formating, system will calculate that how many sectors should be covered by FAT, that is how many sectors should be allocated to FAT. As the following chart, we have known the number of sum sectors, reserved sectors and covered secotrs by FDT, thus we can calculate the number of FAT sectors and DATA sectors.

Reserved sectors=M	FAT1=?	FAT2=FAT1	FDT=N	DATA=?	Spare sectors
Total sectors					

Obviously, FAT sector number has a close relationship with DATA sector number. Because the

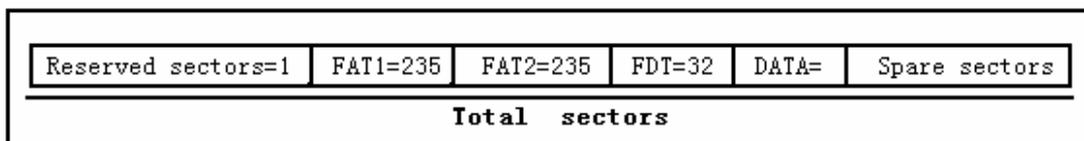
more FAT occupies, the less DATA does, thus there need not so large FAT; while FAT reduces, DATA enlarges, and FAT should be larger; so there is a balance point. According to the balance point, we can get the most reasonable assignment (the most optimal principle), thus getting a formula to calculate the FAT length:

$$\text{FAT sectors} = \frac{\left[\left(\frac{(\text{Total sectors}) - (\text{Reserved sectors}) - (\text{FAT sectors}) - (\text{FDT sectors})}{\text{Sectors per cluster}} * 2 \right) + 4 \right]}{512}$$

FAT sectors= {[(Total sectors – Reserved sectors – FAT sectors*2 – FDT sectors)/512]*2 + 4}/512
 The formula is a quite complex one, here is just an example. Because calculates manually quite complex, here does not calculate the best assignment, only carries on the confirmation. Actually when some systems formatted small partition, because FAT is small, directly uses the sector total and each cluster of sector number to calculate FAT is also a good and convenient way.

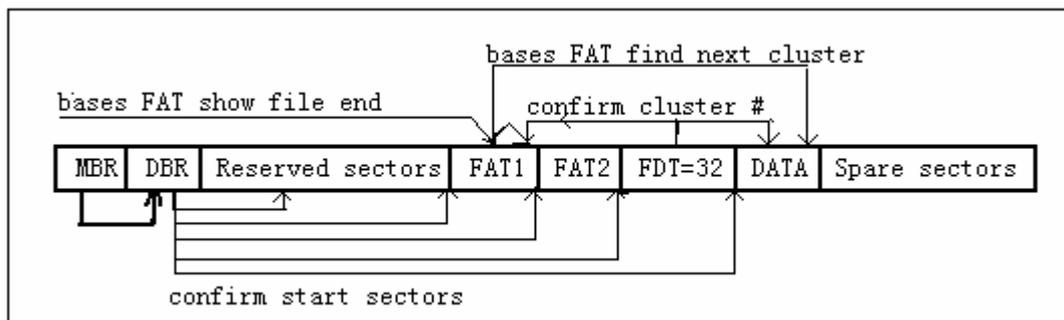
FAT16 partition example:

FAT= {[(240912-1-235*2-32)/4]*2+4}/512=234.7822, take an integer 235. Then there is a problem, if the FAT sector number is too big, the manageable DATA sectors certainly surpasses the actual size of DATA region. Will it write in the data to the outside the DATA region? Of course, system will not have this problem, for it does not use all FAT last sectors but uses part that adapts with DATA. According to the computation, we can obtain the relations among regions as following:



In addition, please note that in practice, the length of FAT often does not meet the optimal principle. If thus, we need just manage according to the actual size, for it does not affect much.

Reserved sectors, FAT and FDT are recorded in DBR. According to these parameters the entrance of each region can be determined. The FDT records parameters of sub-directory or files under the root directory. According to these parameters, it may determine the first cluster of file or directory in DATA region and the start point of cluster chain in FAT. Then, make use of the cluster chain in FAT to track the file or directory in DATA until the file or directory ends. Thus we can know the whole file/directory storage situation in disk. Thereinto, in FAT, there are different situations in file registration, such as backwards, forwards and cross record; this may be have something with the order that the system finds the free cluster when writing files. Here is an example of a file composed by two clusters:



In practice, system manages files and directories in this way. Also we may see that DATA is a very special region, it only has start and end point; the region itself does not have any relation with the file or directory. DATA only records the data, as for relation among the data, it is explained by FDT and FAT. There are only FDT and FAT as the entrance of DATA region, therefore, FDT and FAT are extremely important to the DATA region. If FDT and FAT are destroyed, it is unable to explain the actual meaning of data in DATA. It is system's responsibility to explain them.

VIII. Management of FAT32 file system

1. Root directory management in FAT32 partition

First here is a simple introduction of FDT:

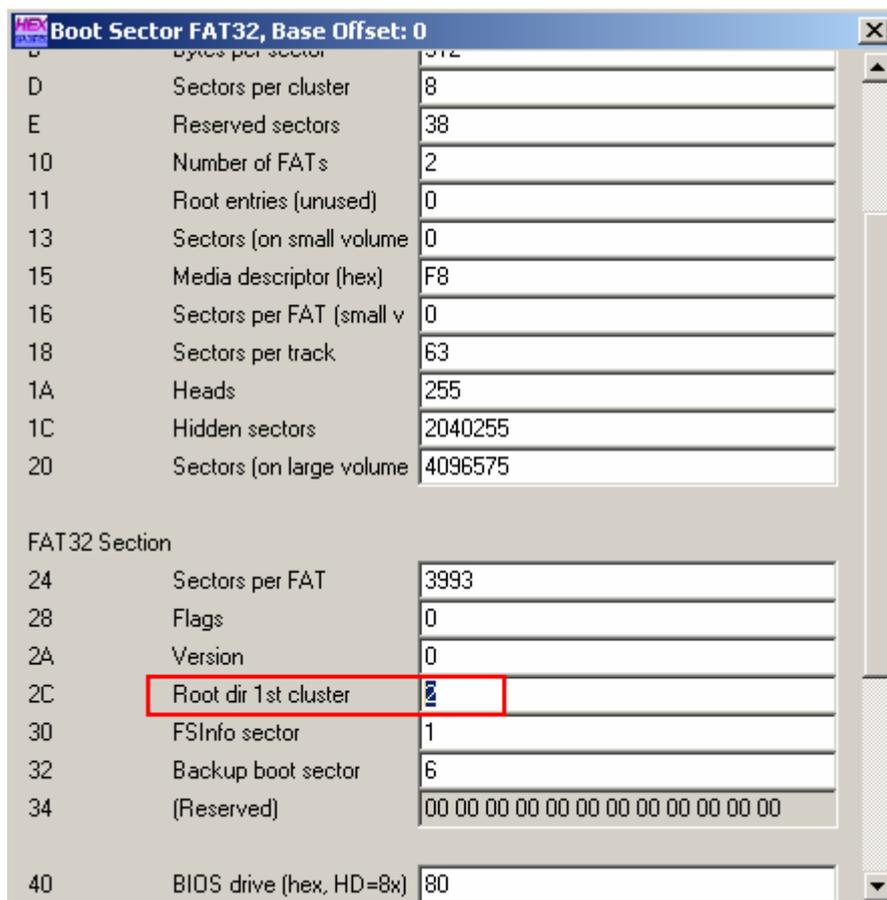
All files/folders in FAT32 have corresponding file entries record in FDT, each file entry records important information of the file/folder (including: file or folder, long filename or short filename, start cluster, the create time, the capacity of file/folder and so on); the file system of operating system searches and localizes corresponding file/folder according to the file information in FDT of each partition. Under FAT32, size of each FDT is 32 bytes.

FAT32 root directory management includes management of files with short and long filename, and management of directories under root directory..

Management of files with short filename

Here is an introduction of management of files with short filename. Take an example of searching a file with short filename (test.txt) under root directory:

After reating or copying file test.txt in FAT32 partition, use utility software to view the BPB parameter of this partition:



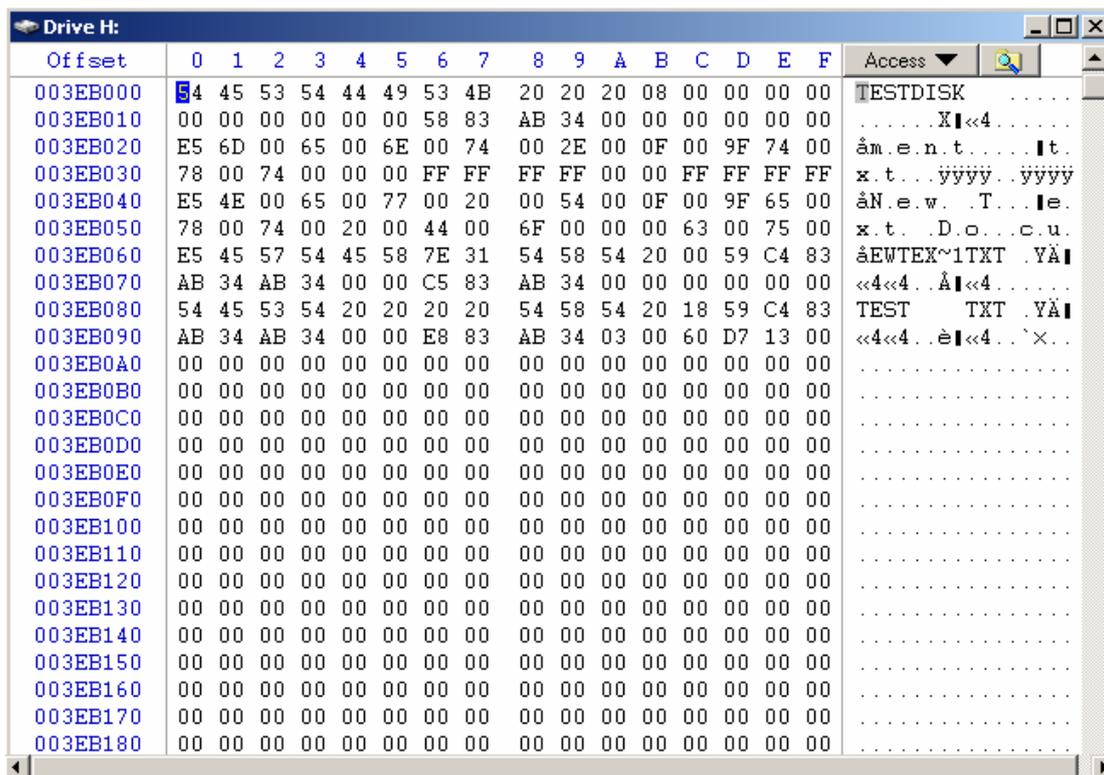
From the chart we may know the start cluster of FDT is 2,

Here is a formula to calculate the FDT start sector:

FDT start sector = reserved sector number + sector number in each FAT * FAT number + (FDT start cluster - 2) * sector number of each cluster.

By the formula we may obtain the start sector of FDT = 8024 sector.

Examining 8024 sector content:



Thereinto the first entry is the partition volume label, search test.txt in the FDT downwards, we can find registration entry of file test.txt in offset 003EB080.

```
003EB080 | 54 45 53 54 20 20 20 20 54 58 54 20 18 59 C4 83 | TEST  TXT .YÄ|
003EB090 | AB 34 AB 34 00 00 E8 83 AB 34 03 00 60 D7 13 00 | <<4. .è|<<4. .`X. .
```

Contents and the meaning of directory entry

Offset	bytes	Meanings
0 ~ 7	8	Filename
8 ~ 10	3	Extended name
11	1	Attribute byte
12 ~ 21	10	Reserved
22 ~ 23	2	Create time
24 ~ 25	2	File date
26 ~ 27	2	First cluster number

28 ~ 31	4	File length
---------	---	-------------

Referring to the above table, from this FDT we may know: length of test.txt is 1270K; the start cluster number is 3rd cluster; from DBR, reserved sector number =38; FAT number = 2, each FAT length =3993 sectors, so the DATA start sector is 38+2*3993=8024; each cluster has 8 sectors. The corresponding sector of the 3rd cluster is 8032rd sector; this sector content is as following chart:

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	Access
003EC000	74	68	69	73	20	61	20	74	65	73	74	20	74	68	69	73	this a test this
003EC010	20	61	20	74	65	73	74	20	74	68	69	73	20	61	20	74	a test this a t
003EC020	65	73	74	20	74	68	69	73	20	61	20	74	65	73	74	20	est this a test
003EC030	74	68	69	73	20	61	20	74	65	73	74	20	74	68	69	73	this a test this
003EC040	20	61	20	74	65	73	74	20	74	68	69	73	20	61	20	74	a test this a t
003EC050	65	73	74	20	74	68	69	73	20	61	20	74	65	73	74	20	est this a test
003EC060	74	68	69	73	20	61	20	74	65	73	74	20	74	68	69	73	this a test this
003EC070	20	61	20	74	65	73	74	20	74	68	69	73	20	61	20	74	a test this a t
003EC080	65	73	74	20	74	68	69	73	20	61	20	74	65	73	74	20	est this a test
003EC090	74	68	69	73	20	61	20	74	65	73	74	20	74	68	69	73	this a test this
003EC0A0	20	61	20	74	65	73	74	20	74	68	69	73	20	61	20	74	a test this a t
003EC0B0	65	73	74	20	74	68	69	73	20	61	20	74	65	73	74	20	est this a test
003EC0C0	74	68	69	73	20	61	20	74	65	73	74	20	74	68	69	73	this a test this
003EC0D0	20	61	20	74	65	73	74	20	74	68	69	73	20	61	20	74	a test this a t
003EC0E0	65	73	74	20	74	68	69	73	20	61	20	74	65	73	74	20	est this a test
003EC0F0	74	68	69	73	20	61	20	74	65	73	74	20	74	68	69	73	this a test this
003EC100	20	61	20	74	65	73	74	20	74	68	69	73	20	61	20	74	a test this a t
003EC110	65	73	74	20	74	68	69	73	20	61	20	74	65	73	74	20	est this a test
003EC120	74	68	69	73	20	61	20	74	65	73	74	20	74	68	69	73	this a test this
003EC130	20	61	20	74	65	73	74	20	74	68	69	73	20	61	20	74	a test this a t
003EC140	65	73	74	20	74	68	69	73	20	61	20	74	65	73	74	20	est this a test
003EC150	74	68	69	73	20	61	20	74	65	73	74	20	74	68	69	73	this a test this
003EC160	20	61	20	74	65	73	74	20	74	68	69	73	20	61	20	74	a test this a t
003EC170	65	73	74	20	74	68	69	73	20	61	20	74	65	73	74	20	est this a test
003EC180	74	68	69	73	20	61	20	74	65	73	74	20	74	68	69	73	this a test this
003EC190	20	61	20	74	65	73	74	20	74	68	69	73	20	61	20	74	a test this a t
003EC1A0	65	73	74	20	74	68	69	73	20	61	20	74	65	73	74	20	est this a test
003EC1B0	74	68	69	73	20	61	20	74	65	73	74	20	74	68	69	73	this a test this
003EC1C0	20	61	20	74	65	73	74	20	74	68	69	73	20	61	20	74	a test this a t

Sector 8032 of 4096575 Offset: 3EC000 = 116 Block: n/a Size: n/a

Now, let's search the sector position of next cluster:

In FAT32, record of each cluster in FAT takes 4 bytes, so the corresponding offset address of the 3rd cluster in FAT is: 4*3 the =12 bytes. Here is the FAT content:

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0000019456	F8	FF	FF	0F	FF	04	00	00	00							
0000019472	05	00	00	00	06	00	00	00	07	00	00	00	08	00	00	00
0000019488	09	00	00	00	0A	00	00	00	0B	00	00	00	0C	00	00	00
0000019504	0D	00	00	00	0E	00	00	00	0F	00	00	00	10	00	00	00
0000019520	11	00	00	00	12	00	00	00	13	00	00	00	14	00	00	00
0000019536	15	00	00	00	16	00	00	00	17	00	00	00	18	00	00	00
0000019552	19	00	00	00	1A	00	00	00	1B	00	00	00	1C	00	00	00
0000019568	1D	00	00	00	1E	00	00	00	1F	00	00	00	20	00	00	00
0000019584	21	00	00	00	22	00	00	00	23	00	00	00	24	00	00	00

The beginning 4 bytes of offset 12 is “04 00 00 00”. Record of each cluster records cluster number of the next cluster; for high order is afterward, the next cluster of test.txt is stored in the 04 cluster in the partition; from offset 04*4=16, the first 4 bytes are “05 00 00 00”, so the following files are saved in 05 cluster; by this method, the last cluster of the file in FAT cluster record is “FF FF FF 0F”, then the FAT chain of test.txt comes to the end:

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Access
0000018528	19	00	00	00	1A	00	00	00	FF	FF	FF	0F	FF	FF	FF	0Fyyy.yyy.
0000018544	FF	FF	FF	0F	FF	FF	FF	0F	00	00	00	00	00	00	00	00	yyy.yyy.....
0000018560	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0000018576	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0000018592	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0000018608	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0000018624	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0000018640	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0000018656	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Management of the long filename file

File registration entry in FDT only reserved 8 bytes for the filename, 3 bytes for extension name (8.3 form for short), so we call it long filename file whose filename scales out.

In FAT32, if it is a short filename, then it can be saved in 32 byte dircotry entry of 8.3 form. When creating a long filename, there are following 3 principles for saving its corresponding short filename:

1. To take the first 6 characters of the long filename, add “~1” to form the short filename, its extension is unchanged;
2. If there has been a file with this filename, the number after “~” will be automatically increased, like“~2”;
3. If there is invalid character of DOS and Windows3.x, use underline “_” to instead.

When some directory entry is used as a long filename directory, its attribute byte is 0FH, saving 13 bytes. A long filename needs more than one directory registration entries. By above storage method, in DOS or Windows3.x, we can only see its corresponding short filename, completely neglecting the long filename. When Windows application requests filename via operating system , Windows will assign different filenames according to attributes of the application - 16 bit application gets filenames of 8.3 form, 32 bit application gets long filenames.

Here’s an example to analyze its management. To establish a “long file test long file1 long file2 long file3 long file4 long file5 long file6 long file7 long file8 long file9 long file10 long file11” file, the sector content in FDT is as following chart:

003EB0F0	AC 34 AC 34 00 00 C5 5C AC 34 00 00 00 00 00 00	-4-4..Å~-4.....
003EB100	4C 6C 00 65 00 31 00 31 00 2E 00 0F 00 D4 74 00	Ll.e.1.1.....Ôt.
003EB110	78 00 74 00 00 00 FF FF FF FF 00 00 FF FF FF FF	x.t...yyyy..yyyy
003EB120	0B 6C 00 65 00 31 00 30 00 20 00 0F 00 D4 20 00	.l.e.1.0.....Ô
003EB130	6C 00 6F 00 6E 00 67 00 20 00 00 00 66 00 69 00	.l.o.n.g.f.i.
003EB140	0A 69 00 6C 00 65 00 39 00 20 00 0F 00 D4 20 00	.i.l.e.9.Ô
003EB150	6C 00 6F 00 6E 00 67 00 20 00 00 00 66 00 69 00	.l.o.n.g.f.i.
003EB160	09 66 00 69 00 6C 00 65 00 38 00 0F 00 D4 20 00	.f.i.l.e.8...Ô
003EB170	20 00 6C 00 6F 00 6E 00 67 00 00 00 20 00 66 00	.l.o.n.g.f.
003EB180	08 20 00 66 00 69 00 6C 00 65 00 0F 00 D4 37 00	..f.i.l.e...Ô7.
003EB190	20 00 20 00 6C 00 6F 00 6E 00 00 00 67 00 20 00	..l.o.n.g.
003EB1A0	07 67 00 20 00 66 00 69 00 6C 00 0F 00 D4 65 00	.g. .f.i.l...Ôe.
003EB1B0	36 00 20 00 20 00 6C 00 6F 00 00 00 6E 00 67 00	6. .l.o.n.g.
003EB1C0	06 6E 00 67 00 20 00 66 00 69 00 0F 00 D4 6C 00	.n.g. .f.i...Ôl.
003EB1D0	65 00 35 00 20 00 20 00 6C 00 00 00 6F 00 6E 00	e.5. .l...o.n.
003EB1E0	05 6F 00 6E 00 67 00 20 00 66 00 0F 00 D4 69 00	.o.n.g. .f...Ôi.
003EB1F0	6C 00 65 00 34 00 20 00 20 00 00 00 6C 00 6F 00	.l.e.4.l.o.
003EB200	04 6C 00 6F 00 6E 00 67 00 20 00 0F 00 D4 66 00	.l.o.n.g.Ôf.
003EB210	69 00 6C 00 65 00 33 00 20 00 00 00 20 00 6C 00	i.l.e.3.l.
003EB220	03 20 00 6C 00 6F 00 6E 00 67 00 0F 00 D4 20 00	..l.o.n.g...Ô
003EB230	66 00 69 00 6C 00 65 00 32 00 00 00 20 00 20 00	f.i.l.e.2....
003EB240	02 74 00 20 00 6C 00 6F 00 6E 00 0F 00 D4 67 00	.t. .l.o.n...Ôg.
003EB250	20 00 66 00 69 00 6C 00 65 00 00 00 31 00 20 00	.f.i.l.e...l.
003EB260	01 6C 00 6F 00 6E 00 67 00 20 00 0F 00 D4 66 00	.l.o.n.g.Ôf.
003EB270	69 00 6C 00 65 00 20 00 74 00 00 00 65 00 73 00	i.l.e. .t...e.s.
003EB280	4C 4F 4E 47 46 49 7E 31 54 58 54 20 00 27 C4 5C	LONGFI~1TXT . 'Å
003EB290	AC 34 AC 34 00 00 F3 5C AC 34 41 01 C0 AE 27 00	-4-4..o~-4A.Å@.

Analyze downwards long filename registration entry the file covers:

The FDT registration entry that last describes long filename is at the very beginning. Its attribute byte is “0FH” (in red frames). This is for distinguishing different file or directory registration entries. This registration entry with an end mark is the tail of long filename registration entry. To search downwards till the end of long filename registration entry, the closely followed first registration entry of the short filename is the corresponding short filename, which is described as “LONGFI~1TXT” (in red frames). From the first long filename before short filename, forwards to the end of long filename registration entry, it is corresponding long file. We can view the file’s location and data in FAT table, DATA region via this file’s short filename registration entry (“LONGFI~1TXT”).

Problems and notes of long filename:

Although Windows95 and its superior system realize the compatibility of long filename and DOS, Windows3.x via above method, there are also several problems:

1. When changing the name of created long filename or deleting the filename, it will be lost, and the long filename directory entry space is also unable to be taken back. Because files track and manage mainly by short filename, the long filename exists adhering the short filename.
2. If run 16 bits application, when changing a filename, the corresponding long filename will be lost.
3. The directory registration entry that long filename uses must be continual, frequently creating and deleting long filename may cause massive disk fragments.

Therefore do not create long filename under root directory unless you have to. You can run disk defragment frequently, and recycle lost directory registration entry.

2. Management of sub-directory in FAT32

Besides establishing the file directly in logical drive root directory, we may also establish the inferior directory, which is called sub-directory of the root directory. So called sub-directory and parental directory are relative: a parental directory may have many sub-directories, while a sub-directory has only one parental directory. Under the sub-directory of root directory, we may create more inferior sub-directories, thus forming a directory tree. For directories under root directory, its entrance still exists in root directory. .

To establish a long filename sub-directory “test folder” under the FAT32 root directory, the registration entry in root directory FDT are as following chart:

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Access
0008196096	54	45	53	54	20	44	49	53	4B	20	20	08	00	00	00	00	TEST DISK
0008196112	00	00	00	00	00	00	1B	53	B2	34	00	00	00	00	00	00	S^4
0008196128	41	74	00	65	00	73	00	74	00	20	00	0F	00	8C	66	00	At.e.s.t.lf.
0008196144	6F	00	6C	00	64	00	65	00	72	00	00	00	00	00	FF	FF	o.l.d.e.r.....yy
0008196160	54	45	53	54	46	4F	7E	31	20	20	20	10	00	96	23	53	TESTFO~1 ...!#S
0008196176	B2	34	B2	34	00	00	24	53	B2	34	03	00	00	00	00	00	?4?4..\$S^4.....
0008196192	42	20	00	49	00	6E	00	66	00	6F	00	0F	00	72	72	00	B..I.n.f.o...rr.
0008196208	6D	00	61	00	74	00	69	00	6F	00	00	00	6E	00	00	00	m.a.t.i.o...n...
0008196224	01	53	00	79	00	73	00	74	00	65	00	0F	00	72	6D	00	.S.y.s.t.e.e...rm.

From the chart we may see, the sub-directory management completely complies with the FAT32 long filename management rule.

From this file’s registration entry, we may know the start cluster of “test folder” is located in cluster “00000003”. Here is an examination of its content:

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Access
0008200192	2E	20	20	20	20	20	20	20	20	20	20	10	00	96	23	53!#S
0008200208	B2	34	B2	34	00	00	24	53	B2	34	03	00	00	00	00	00	?4?4..\$S^4.....
0008200224	2E	2E	20	20	20	20	20	20	20	20	20	10	00	96	23	53!#S
0008200240	B2	34	B2	34	00	00	24	53	B2	34	00	00	00	00	00	00	?4?4..\$S^4.....
0008200256	41	73	00	75	00	62	00	64	00	69	00	0F	00	F5	72	00	As.u.b.d.i...ör.
0008200272	65	00	63	00	74	00	6F	00	72	00	00	00	79	00	00	00	e.c.t.o.r...y...
0008200288	53	55	42	44	49	52	7E	31	20	20	20	10	00	2A	42	53	SUBDIR~1 ...*BS
0008200304	B2	34	B2	34	00	00	43	53	B2	34	0B	00	00	00	00	00	?4?4..CS^4.....
0008200320	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

1. Folder“.”and“..”

Note: These two are system folders, existing under each sub-directory. Thereinto“.” represents current directory, while“..”represents parental directory of the current directory, pointing to the start cluster of its parental directory. Readers who ever used DOS may remember that using DOS command “CD ..”can jump to the upper directory. The system realizes the bidirectional connection

between directories by this way, thus system can switch between directories.

2.Folder “subdirectory”, this is a long filename folder, which can be known from the 0FH mark of file registration entry.

From above introduction, we may know that, managements of sub-directory and the file have similarities as well as differences. Firstly, the expressive way of directory name and filename is same, including the naming rule of its long and short form as well as all dates, time definitions and so on; secondly, the expressive way of start cluster is also same, except that of attribute and length; for sub-directory, its length is fixed 0, which is convenient for system administration. Otherwise, any change under the sub-directory will affect it, causing low efficiency; moreover, the directory length is not convenient to define. it may conflict with the file length.

Let’s check the start cluster of (000B) content of sub-directory “subdirectory”:

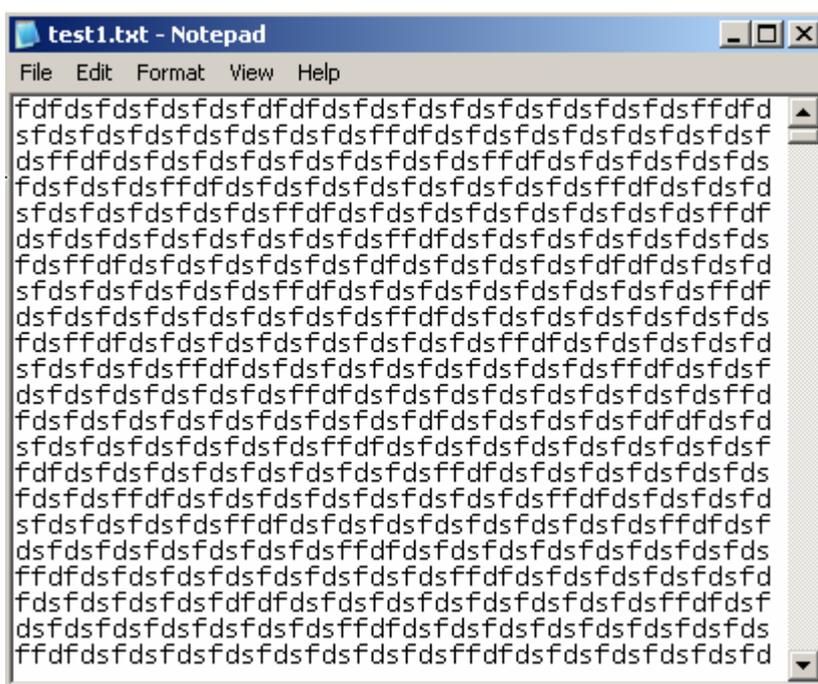
Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Access
0008232960	2E	20	20	20	20	20	20	20	20	20	20	10	00	2A	42	53	. . . *BS
0008232976	B2	34	B2	34	00	00	43	53	B2	34	0B	00	00	00	00	00	²4²4..CS²4.....
0008232992	2E	2E	20	20	20	20	20	20	20	20	20	10	00	2A	42	53	. . . *BS
0008233008	B2	34	B2	34	00	00	43	53	B2	34	03	00	00	00	00	00	²4²4..CS²4.....
0008233024	E5	78	00	74	00	00	00	FF	FF	FF	FF	0F	00	99	FF	FF	âx.t...ÿÿÿÿ..ÿÿÿÿ
0008233040	FF	00	00	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿ..ÿÿÿÿ									
0008233056	E5	74	00	65	00	73	00	74	00	20	00	0F	00	99	66	00	ât.e.s.t.¹f.
0008233072	69	00	6C	00	65	00	20	00	31	00	00	00	2E	00	74	00	i.l.e. .1.....t.
0008233088	E5	45	53	54	46	49	7E	31	54	58	54	20	00	54	5C	53	âESTFI~1TXT .T\S
0008233104	B2	34	B2	34	00	00	7B	53	B2	34	0C	00	E0	97	00	00	²4²4..{S²4..à ..
0008233120	42	78	00	74	00	00	00	FF	FF	FF	FF	0F	00	B9	FF	FF	Bx.t...ÿÿÿÿ..¹ÿÿÿ
0008233136	FF	00	00	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿ..ÿÿÿÿ									
0008233152	01	74	00	65	00	73	00	74	00	20	00	0F	00	B9	66	00	.t.e.s.t.¹f.
0008233168	69	00	6C	00	65	00	20	00	32	00	00	00	2E	00	74	00	i.l.e. .2.....t.
0008233184	54	45	53	54	46	49	7E	32	54	58	54	20	00	54	5C	53	TESTFI~2TXT .T\S
0008233200	B2	34	B2	34	00	00	81	53	B2	34	16	00	F0	4B	00	00	²4²4.. S²4..âK..
0008233216	54	45	53	54	31	20	20	20	54	58	54	20	18	54	5C	53	TEST1 TXT .T\S
0008233232	B2	34	B2	34	00	00	7B	53	B2	34	0C	00	E0	97	00	00	²4²4..{S²4..à ..

Obviously, the start sector content of “subdirectory” is similar to the root directory structure; it stores separate file directory entry, whose management is an extension of the file directory management under the root directory. This sub-directory manages its files and directories in the same way as the root directory does. But there is some difference between sub-directory and the root directory, that is how they manage themselves: the root directory is determined by DBR, while the sub-directory of root directory by directory registration under root directory.

From the chart above we can see the last file registration is file “teset1.txt”, the start cluster is “0000000C”. To examine this cluster:

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Access
0008237056	66	64	66	64	73	66	64	73	66	64	73	66	64	73	66	64	fdfsdfsdfsdfsdf
0008237072	66	64	66	64	73	66	64	73	66	64	73	66	64	73	66	64	fdfsdfsdfsdfsdf
0008237088	73	66	64	73	66	64	73	66	64	73	66	66	64	66	64	73	sdfsdfsdfsffdfs
0008237104	66	64	73	66	64	73	66	64	73	66	64	73	66	64	73	66	fdfsdfsdfsdfsdf
0008237120	64	73	66	64	73	66	66	64	66	64	73	66	64	73	66	64	dsfsdffdfsdfsdf
0008237136	73	66	64	73	66	64	73	66	64	73	66	64	73	66	64	73	sdfsdfsdfsdfsdf
0008237152	66	66	64	66	64	73	66	64	73	66	64	73	66	64	73	66	ffdfsdfsdfsdfsdf
0008237168	64	73	66	64	73	66	64	73	66	64	73	66	66	64	66	64	dsfsdfsdfsffdfs
0008237184	73	66	64	73	66	64	73	66	64	73	66	64	73	66	64	73	sdfsdfsdfsdfsdf
0008237200	66	64	73	66	64	73	66	66	64	66	64	73	66	64	73	66	fdfsdffdfsdfsdf
0008237216	64	73	66	64	73	66	64	73	66	64	73	66	64	73	66	64	dsfsdfsdfsdfsdf
0008237232	73	66	66	64	66	64	73	66	64	73	66	64	73	66	64	73	sffdfsdfsdfsdfs
0008237248	66	64	73	66	64	73	66	64	73	66	64	73	66	66	64	66	fdfsdfsdfsffdfs
0008237264	64	73	66	64	73	66	64	73	66	64	73	66	64	73	66	64	dsfsdfsdfsdfsdf
0008237280	73	66	64	73	66	64	73	66	66	64	66	64	73	66	64	73	sdfsdfsffdfsdfs
0008237296	66	64	73	66	64	73	66	64	73	66	64	73	66	64	73	66	fdfsdfsdfsdfsdf
0008237312	64	73	66	66	64	66	64	73	66	64	73	66	64	73	66	64	dsffdfsdfsdfsdf
0008237328	73	66	64	73	66	64	73	66	64	73	66	64	73	66	66	64	sdfsdfsdfsdfsdf
0008237344	66	64	73	66	64	73	66	64	73	66	64	73	66	64	73	66	fdfsdfsdfsdfsdf
0008237360	64	66	64	73	66	64	73	66	64	73	66	64	73	66	64	66	dfdfsdfsdfsdfsdf
0008237376	64	66	64	73	66	64	73	66	64	73	66	64	73	66	64	73	dfdfsdfsdfsdfs

The corresponding file contents are in the following notepad:



3. File deletion in FAT32

When deleting a file, the system only makes a deletion mark on this file's directory entry, marking clusters it covers in FAT as "empty"; clusters in DATA remains original file's contents. When writing in data again, the original file content might be covered by new information.

There is a Recycle Bin in Windows, its principle of operation is:

The recycling bin is only some space on the hard disk; the Windows system automatically

establishes a folder "RECYCLED" (under root directory of each disk partition) with hiding attribute to save temporarily deleted files. Only when deleting or executing "Clear" command, these files then can be completely deleted (as to operating system). As "the recycling bin" we see on the desktop, it is only a shortcut. Then we will introduce fast deletion and complete deletion separately.

Fast deletion

Fast deletion of files is just to put them into Recycle Bin. In this situation, the data can be recovered.

Comparing the changes of FDT, FAT and DATA between before and after deletion, we can find the rules.

FDT before deleting "test1.txt":

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Access
0008233120	42	78	00	74	00	00	00	FF	FF	FF	FF	0F	00	B9	FF	FF	Ex.t...yyyy..1yy
0008233136	FF	00	00	FF	FF	FF	FF	yyyyyyyyyy..yyy									
0008233152	01	74	00	65	00	73	00	74	00	20	00	0F	00	B9	66	00	.t.e.s.t...1f.
0008233168	69	00	6C	00	65	00	20	00	32	00	00	00	2E	00	74	00	i.l.e..2....t.
0008233184	54	45	53	54	46	49	7E	32	54	58	54	20	00	54	5C	53	TESTFI~2TXT.T\S
0008233200	B2	34	B2	34	00	00	81	53	B2	34	16	00	F0	4B	00	00	?4?4..IS?4..8K..
0008233216	54	45	53	54	31	20	20	20	54	58	54	20	18	54	5C	53	TEST1.TXT.T\S
0008233232	B2	34	B2	34	00	00	7B	53	B2	34	0C	00	E0	97	00	00	?4?4..(S?4..aI..
0008233248	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

FAT before deletion:

Attention: the first clusters number of the file is "000C", i.e. 12. The offset of the cluster record in FAT is "0184800", whose content is "0D 00 00 00"

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Access
0000018432	F8	FF	FF	0F	FF	0F	FF	FF	FF	0F	øyy.yyyyyyy.yyy.						
0000018448	FF	FF	FF	0F	yyy.yyy.yyy.yyy.												
0000018464	FF	FF	FF	0F	yyy.yyy.yyy.yyy.												
0000018480	0D	00	00	00	0E	00	00	00	0F	00	00	00	10	00	00	00
0000018496	11	00	00	00	12	00	00	00	13	00	00	00	14	00	00	00
0000018512	15	00	00	00	FF	FF	FF	0F	17	00	00	00	18	00	00	00	...yyy.....
0000018528	19	00	00	00	1A	00	00	00	FF	FF	FF	0F	FF	FF	FF	0F	...yyy.yyy..
0000018544	FF	FF	FF	0F	FF	FF	FF	0F	00	00	00	00	00	00	00	00	yyy.yyy.....
0000018560	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

FDT after deletion:

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Access
0008233120	42	78	00	74	00	00	00	FF	FF	FF	FF	0F	00	B9	FF	FF	Ex.t...yyyy..1yy
0008233136	FF	00	00	FF	FF	FF	FF	yyyyyyyyyy..yyy									
0008233152	01	74	00	65	00	73	00	74	00	20	00	0F	00	B9	66	00	.t.e.s.t...1f.
0008233168	69	00	6C	00	65	00	20	00	32	00	00	00	2E	00	74	00	i.l.e..2....t.
0008233184	54	45	53	54	46	49	7E	32	54	58	54	20	00	54	5C	53	TESTFI~2TXT.T\S
0008233200	B2	34	B2	34	00	00	81	53	B2	34	16	00	F0	4B	00	00	?4?4..IS?4..8K..
0008233216	E5	45	53	54	31	20	20	20	54	58	54	20	18	54	5C	53	âEST1.TXT.T\S
0008233232	B2	34	B2	34	00	00	7B	53	B2	34	0C	00	E0	97	00	00	?4?4..(S?4..aI..

From two charts we can see, after deleting "test1.txt", the first byte of registration entry is changed

released empty cluster, thus covering the space ever used by the two files.

3. After complete deletion, there is no change in DATA sectors at all. However, data in DATA region is explained by FDT and FAT. If FDT and FAT marked the DATA as “unused”, system will certainly consider it as “empty” space and will write in new information at any time. Thus, the room is still released. This is different with fast deletion.

4. Sub-directory deletion in FAT32

Operating system manages sub-directory in the same way as manages files. So, the deletion ways are same, too.

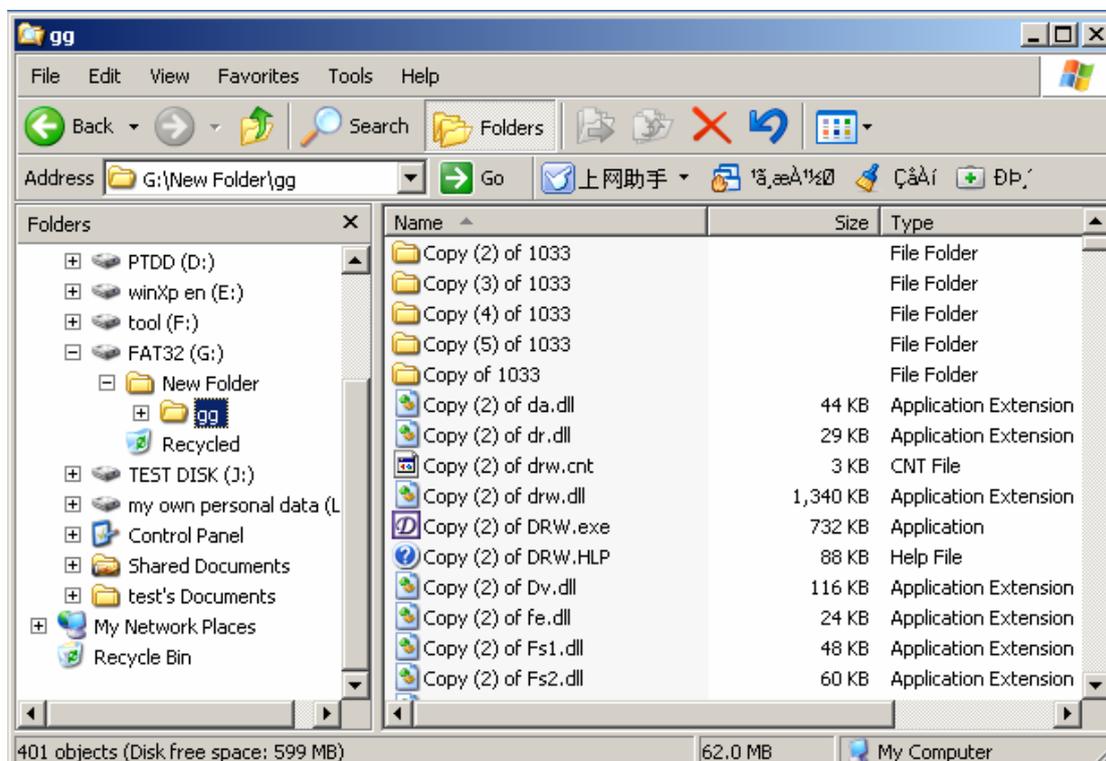
Fast deletion

Fast deletion of sub-directory is the same as that of files. It just marked a deletion mark to the beginning byte in FDT that describes sub-directory; all files under this sub-directory and records of its inferior sub-directory are not changed, that is, just to “remove” this sub-directory into recycling bin.

Complete deletion

Take an example of completely deleting the sub-directory “\New Folder\gg”:

The situation is:



To completely delete it, and compare changes of FDT, FAT and DATA between before and after

defragment tools, connecting the discontinuous regions to enhance system efficiency. While high-level format does not reserve data. The best defragment tool resets system regions in the whole disk, which makes it a blank disk, and all clusters are marked as available.

Fast high level format

To compare some regions, we can see changes of in FDT and FAT by high level format.

FDT before fast high level format:

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Access
001380352	46	41	54	33	32	20	20	20	20	20	20	08	00	00	00	00	FAT32
001380368	00	00	00	00	00	00	7B	55	B3	34	00	00	00	00	00	00{U³4.....
001380384	41	74	00	65	00	73	00	74	00	20	00	0F	00	8C	66	00	At.e.s.t. . . . f.
001380400	6F	00	6C	00	64	00	65	00	72	00	00	00	00	00	FF	FF	o.l.d.e.r. . . . ÿÿ
001380416	54	45	53	54	46	4F	7E	31	20	20	20	10	00	38	84	55	TESTFO~1 . . . 8 U
001380432	B3	34	B3	34	00	00	85	55	B3	34	03	00	00	00	00	00	³4³4. . . U³4.
001380448	42	20	00	49	00	6E	00	66	00	6F	00	0F	00	72	72	00	B . I.n.f.o. . . r.r.
001380464	6D	00	61	00	74	00	69	00	6F	00	00	00	6E	00	00	00	m.a.t.i.o. . . n. . .
001380480	01	53	00	79	00	73	00	74	00	65	00	0F	00	72	6D	00	.S.y.s.t.e. . . r.m.
001380496	20	00	56	00	6F	00	6C	00	75	00	00	00	6D	00	65	00	.V.o.l.u. . . m.e.
001380512	53	59	53	54	45	4D	7E	31	20	20	20	16	00	39	84	55	SYSTEM~1 . . . 9 U
001380528	B3	34	B3	34	00	00	85	55	B3	34	04	00	00	00	00	00	³4³4. . . U³4.
001380544	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
001380560	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

FDT after fast high level format:

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Access
001380352	46	41	54	33	32	20	20	20	20	20	20	08	00	00	00	00	FAT32
001380368	00	00	00	00	00	00	20	56	B3	34	00	00	00	00	00	00 V³4.
001380384	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
001380400	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
001380416	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
001380432	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
001380448	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
001380464	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
001380480	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

We can see that after fast high level format, the system will clear all directory registration entries including Recycle Bin except a volume label (there is no data, naturally no deleted file or directory; Recycle Bin is unnecessary to exist; only for the first time simple deletion is executed, system will automatically establish this directory).

FAT before fast high level format:

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Access
000017408	F8	FF	FF	0F	FF	0F	FF	FF	FF	0F	eÿÿ.ÿÿÿÿÿÿÿ.ÿÿÿ.						
000017424	FF	FF	FF	0F	ÿÿÿ.ÿÿÿ.ÿÿÿ.ÿÿÿ.												
000017440	09	00	00	00	0A	00	00	00	0B	00	00	00	0C	00	00	00
000017456	FF	FF	FF	0F	FF	FF	FF	0F	FF	FF	FF	0F	00	00	00	00	ÿÿÿ.ÿÿÿ.ÿÿÿ.
000017472	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000017488	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

FAT after fast high level format:

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Access
000017408	F8	FF	FF	0F	FF	0F	00	00	00	00	øÿÿ.ÿÿÿÿÿÿÿÿ.....						
000017424	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000017440	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000017456	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000017472	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000017488	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000017504	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000017520	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

From here we can see the FAT table has been cleared. If the files do not be stored continuously, it would be very difficult to recover.

So, how about the contents of sub-directory after fast high level formatst? The FAT must be reset, but the file directory entries still exist. The following is the file registration entry under a sub-directory after format:

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Access
001384448	2E	20	20	20	20	20	20	20	20	20	20	10	00	38	84	558 U
001384464	B3	34	B3	34	00	00	85	55	B3	34	03	00	00	00	00	00	³4³4... U³4.....
001384480	2E	2E	20	20	20	20	20	20	20	20	20	10	00	38	84	558 U
001384496	B3	34	B3	34	00	00	85	55	B3	34	00	00	00	00	00	00	³4³4... U³4.....
001384512	41	73	00	75	00	62	00	64	00	69	00	0F	00	F5	72	00	As.u.b.d.i...ör.
001384528	65	00	63	00	74	00	6F	00	72	00	00	00	79	00	00	00	e.c.t.o.r...y...
001384544	53	55	42	44	49	52	7E	31	20	20	20	10	00	3B	84	55	SUBDIR~1 ... U
001384560	B3	34	B3	34	00	00	85	55	B3	34	07	00	00	00	00	00	³4³4... U³4.....
001384576	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
001384592	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

From the chart we can see, after fast high-level format, the sub-directory content is not deleted, in other words, data in DATA region is not affected. But, there is no entrance for them in FDT and FAT, for operating system, they do not exist. Here why we can see them is that we have used some tool software, which records their clusteres before format. Also we can see this cluster's content has not been covered. But, if users do not backup before format, how do they know which clusteres they are stored in? How to explain these clusteres' contents? Therefore, here we call out their past record. But, it does not mean we know the situation before fast high-level format. Fast high-level format is irreversible for operating system.

Complete high level format

After complete high level format, FDT and FAT of root directory are definitely reset.

Let's see whether the contents of sub-directory are changed:

Before format:

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Access	
001384448	2E	20	20	20	20	20	20	20	20	20	10	00	38	84	55	.	..8 U	
001384464	B3	34	B3	34	00	00	85	55	B3	34	03	00	00	00	00	³4³4.. U³4.....		
001384480	2E	2E	20	20	20	20	20	20	20	20	10	00	38	84	55	.	..8 U	
001384496	B3	34	B3	34	00	00	85	55	B3	34	00	00	00	00	00	³4³4.. U³4.....		
001384512	41	73	00	75	00	62	00	64	00	69	00	0F	00	F5	72	00	As.u.b.d.i...õr.	
001384528	65	00	63	00	74	00	6F	00	72	00	00	00	79	00	00	00	e.c.t.o.r...y...	
001384544	53	55	42	44	49	52	7E	31	20	20	20	10	00	3B	84	55	SUBDIR~1 ..; U	
001384560	B3	34	B3	34	00	00	85	55	B3	34	07	00	00	00	00	00	³4³4.. U³4.....	
001384576	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
001384592	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	

After format:

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Access	
001384448	2E	20	20	20	20	20	20	20	20	20	10	00	38	84	55	.	..8 U	
001384464	B3	34	B3	34	00	00	85	55	B3	34	03	00	00	00	00	00	³4³4.. U³4.....	
001384480	2E	2E	20	20	20	20	20	20	20	20	10	00	38	84	55	.	..8 U	
001384496	B3	34	B3	34	00	00	85	55	B3	34	00	00	00	00	00	00	³4³4.. U³4.....	
001384512	41	73	00	75	00	62	00	64	00	69	00	0F	00	F5	72	00	As.u.b.d.i...õr.	
001384528	65	00	63	00	74	00	6F	00	72	00	00	00	79	00	00	00	e.c.t.o.r...y...	
001384544	53	55	42	44	49	52	7E	31	20	20	20	10	00	3B	84	55	SUBDIR~1 ..; U	
001384560	B3	34	B3	34	00	00	85	55	B3	34	07	00	00	00	00	00	³4³4.. U³4.....	
001384576	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
001384592	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	

Obviously, the same as fast format, complete format does not change the contents of sub-directory. Neither dose complete format changes the contents of file.

Summarization:

Now we know no matter simple deletion, complete deletion, complete format or fast format, neither of them would write “0” to DATA, completely destroying data. When system is dealing with large files or the whole partition, the time it takes is much less than copying files does. Thus, it is impossible to write “0” into thoes regions. Please pay attention, even complete format is not secure (it is also the base of recovery). Especially for secret and important data, we need some software of third party to erase for fear of their being stolen. In this occasion, it is irrecoverable.

IX, Management of NTFS file system

1. The NTFS file system introduction

Before Windows NT 3.1 released, Microsoft invented two file systems: FAT (File Allocation Table) based on MS-DOS and Windows, and High Performance File System (HPFS) used in OS/2 operating system. As Windows NT coming to the market, Microsoft needed a new file system to support the security and reliability of NT. FAT and HPFS obviously were born to have defects in this. After deliberation, the design team decided to create a brand-new file system with good tolerance and security - NTFS (New Technology File System). From Windows NT 3.1 to Windows 2000/XP, NTFS was being developed unceasingly. Compared with FAT and HPFS, besides the enviable performance features and backward compatibility, NTFS also becomes the first file system that provides sound file service for high server and Intel workstation family.

From the very beginning, NTFS is set for enterprise file system. In order to reduce data loss by sudden power cut or system collapsing, the file system should always guarantee the integrity of metadata in file system. In order to protect the sensitive data from illegal visit, there should be a comprehensive security model in file system; In order to protect the user data, the file system should provide inexpensive data redundancy plan based on software instead of expensive one based on hardware.

High-level features of NTFS

1. Multi-data streams
2. Name based on Unicode
3. General index mechanism
4. The dynamic bad cluster reprints maps
5. Supports POSIX
6. File compression
7. File encrypts
8. Disk quota
9. Hard link and soft link
10. Link tracks
11. Log records
12. Fragmentation

2. NTFS file system terminology

LCN: Logical Cluster Number

VCN: Virtual Cluster Number

BPB: BIOS Parameter Block

FSD: File System Driver

SCB: System Control Block

FCB: File Control Block
EFS: Encrypt File System
MFT: Master File Table
MFT Mirror: Master File Table Mirror

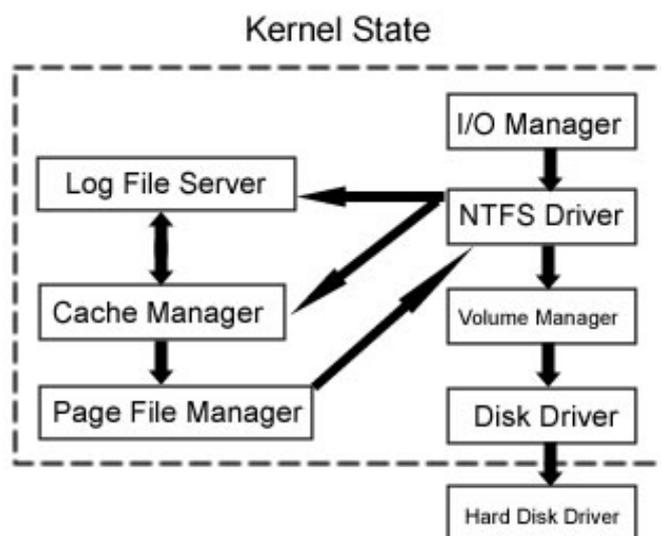
Metadata: It's data stored in volume, supporting file system management. It cannot be visit by application program, just provides service for the system.

3. Data constructions of NTFS file system

4. Drivers of NTFS

Win 32 I/O API is completed by I/O management. I/O management sends requests of I/O to NTFS FSD to be executed.

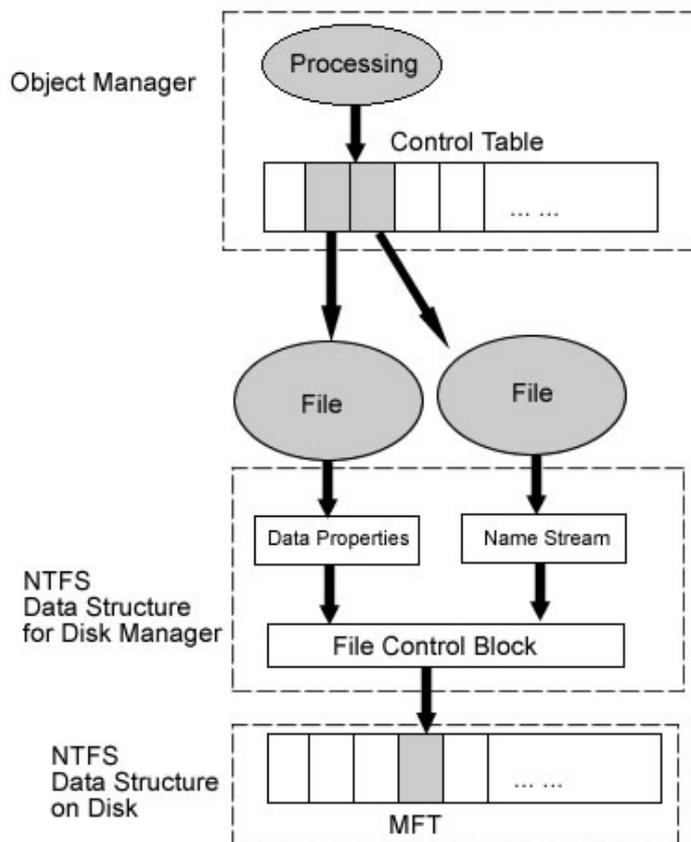
In implementation, I/O management also works together with high speed buffer management, memory management, file log service, volume management and disk driver.



Applications create and store files via FSD of NTFS. This process is includes following steps: Firstly Windows 2000/XP checks authority, only legal users' request can be run. Then I/O management transforms the file handle into the file object indicator. Finally NTFS obtains files in disk through the file object indicator.

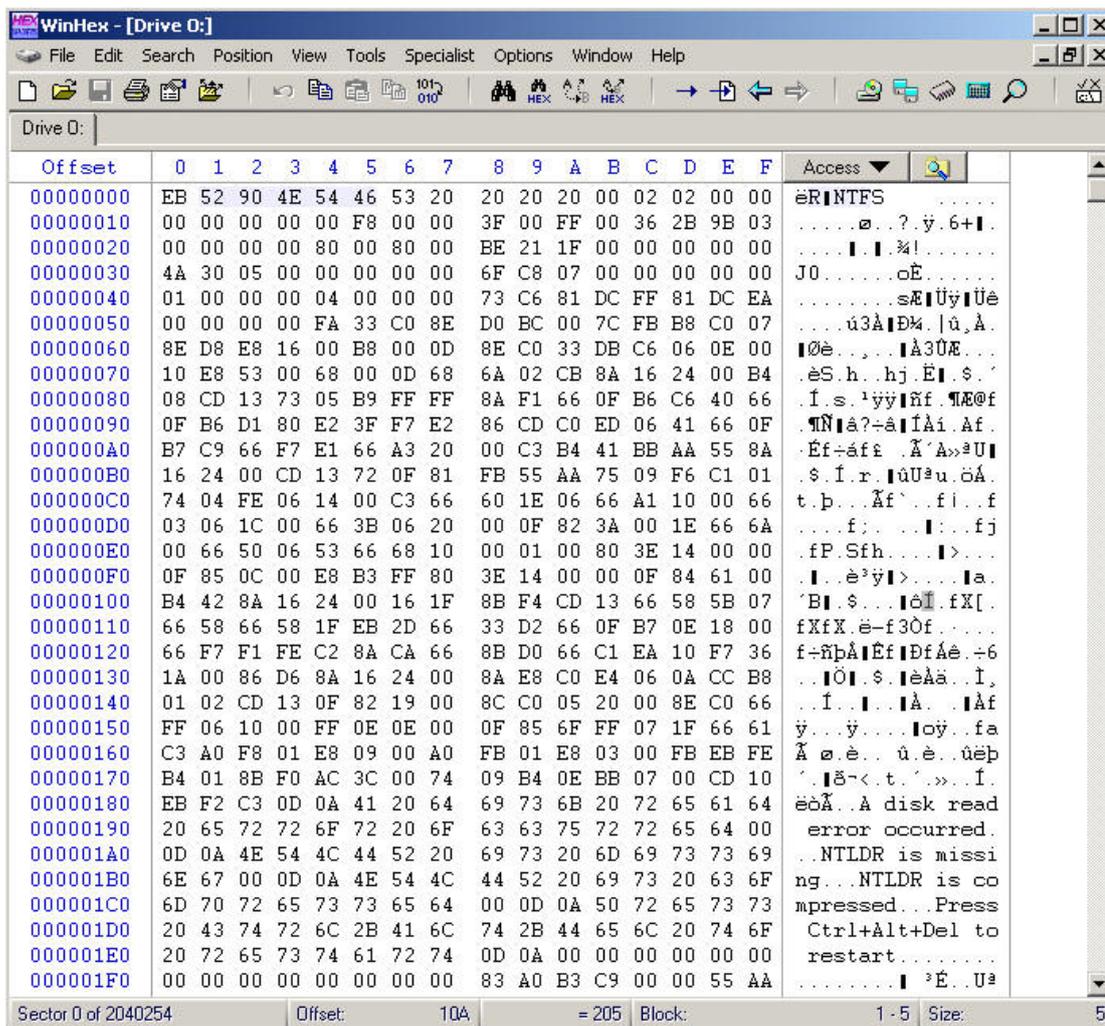
Now let's analyze how NTFS obtains files in disk through the file object indicator. NTFS obtains stream control block (SCB) of file attribute through the file object indicator. Each SCB expresses the single attribute file, and includes information on how it obtains that attribute. All SCBs of a file point to a common data construction File Control Block (FCB). FCB contains an indicator that points to the file record of main file table (MFT). NTFS gets the file access authority through this

indicator.



5.DBR of NTFS file system

The effect of boot-sector of NTFS is the same as that of FAT16 and FAT32: MBR boots to DBR of active partition, then DBR boots operating system; for Windows NT/2000/XP/2003, DBR calls in fold NTLDR, and then NTLDR calls in system kernel.



NTFS 的 DBR
DBR of NTFS

Offset	Title	Value
0	JMP instruction	EB 52 90
3	SystemID	NTFS
B	Bytes per sector	512
D	Sectors per cluster	2
E	Reserved sectors	0
10	(always zero)	00 00 00
13	(unused)	00 00
15	Media descriptor	F8
16	(unused)	00 00
18	Sectors per track	63
1A	Heads	255
1C	Hidden Sectors	60500790
20	(unused)	00 00 00 00
24	(always 80 00 80 00)	80 00 80 00
28	Total sectors	2040254
30	Start C# \$MFT	340042
38	Start C# \$MFTMirr	510063
40	Clusters per FILE record	1
41	(unused)	0
44	Clusters per INDX block	4
45	(unused)	0
48	32-bit serial number (hex)	73 C6 81 DC
48	32-bit SN (hex, reversed)	DC81C673
48	64-bit serial number (hex)	73 C6 81 DC FF 81 DC EA
50	Checksum	0
1FE	Signature (55 AA)	55 AA

BPB parameter
BPB parameter table

Byte offset	Length	Common range	note
0x0b	bit	0x0002	number of bytes in each sector
0x0d	byte	0x08	number of sectors in each cluster
0x0e	bit	0x000	reserved sector
0x10	3byte	0x000000	always 0
0x13	bit	0x0000	free NTFS, 0
0x15	byte	0xf8	medium description
0x16	bit	0x0000	always 0
0x18	bit	0x3f00	number of sectors in each track
0x1a	bit	0xff00	number of heads
0x1c	Double bit	0x3f000000	hidden sector

0x20	Double bit	0x00000000	free NTFS, 0
0x24	Double bit	0x80008000	free NTFS, 0
0x28	8byte	0x4af57f0000000000	total number of sectors
0x30	8byte	0x0400000000000000	logic start cluster number of \$MFT
0x38	8byte	0x54ff070000000000	logic start cluster number of \$MFTMirr
0x40	Double bit	0xf6000000	cluster record number of each MFT
0x44	Double bit	0x01000000	number of index cluster
0x48	8byte	0x14a51b74c91b741c	volume lable
0x50	Double bit	0x00000000	checks the sum

In NTFS volume, there is an extended BPB formed by data fields that follow the BPB. Data in these fields enables NTLDR to find master file table \$MFT in starting process. In the NTFS volume, \$MFT isn't placed in a pre-definition sector, which is different from that in FAT16 volume and FAT32 volume. So if there is some bad sectors in normal position of MFT, we can move the \$MFT to another place. But, if the data is destroyed, the position of \$MFT cannot be found out; then Windows 2000 will consider this volume as unformatted. Therefore, if a NTFS volume prompt unformatted, it is possible that \$MFT is not destroyed. And it may reconstruct BPB according to the meaning of each fields of BPB.

6. Metadata of NTFS file system

In NTFS file system; the files are also assigned according to clusters. A cluster must be the integral multiple of physical sector, moreover always integer powers of 2. NTFS file system does not care for about sector, nor sector size (for example not 512 bytes), but cluster size is assigned automatically according to the volume size by format program when formatting.

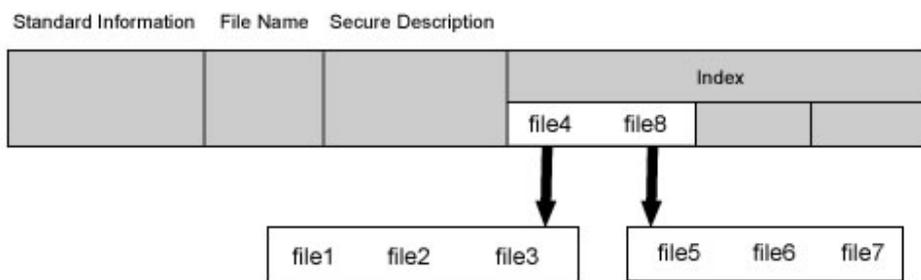
By master file table (MFT) to determine the file storage location in disk. The master file table is a corresponding database composed of series of file records—each file has a record in a volume (large files may have many corresponding records). The first file record is called basic file record with some information of other extended file records. Master file table itself also has its own file record.

Each file in NTFS volume has a unique identifier of 64 bit called file quotation number/File Reference Number. The file quotation number consists of two parts: file number and file order. The file number is 48 bit, corresponding to the position of MFT. The file order increases as repeating usage of file records, which is designed for internal consistency examination of NTFS.

NTFS locates clusters with logic cluster number (LCN) and virtual cluster number (VCN). LCN is the simple number for all clusters from beginning to end. With volume factors multiplying LCN, we can get physical byte offset in volume, thus obtaining physics disk address. VCN is the number for specified files from beginning to end, which is convenient for quoting the data of files. VCN

can be mapped as LCN, but it does not request physical continuity.

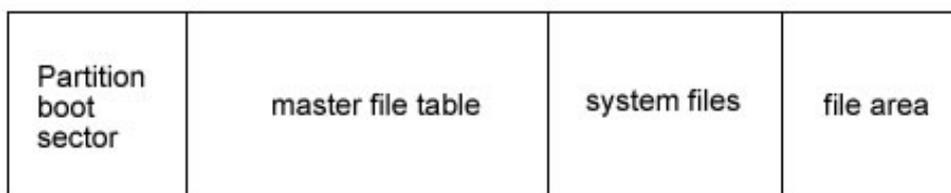
Directory of NTFS is only a simple filename and index of the file quotation number. If the directory attribute list is smaller than the length of a record, then all information of this directory are saved in the main file table records, as to that bigger than record directory, it was managed with B+ tree.



In the basic file recording of master file table, it has an indicator aiming at the index cache which is not usually used, including sub-directory and file exterior cluster, but B+ tree structure is convenient for the file and the sub-directory in large-scale table of contents.

In NTFS, all data saved in the volume are contained in file, including file construction that is used to locate and obtain files, boot program and bitmap files that record volume size and service condition. This manifests the principle of NTFS: all things in disk are called files. Saving everything in files makes the file system very easy to locate and maintain data. The file fixes its storage location in disk with master file table.

The relationship of NTFS' each region:



Size of file records in MFT is generally fixed. No matter the size of a cluster, the records all are 1KB, which equals to inode (i node) in Linux. File records in MFT file record array are physically continual, number from 0. Therefore NTFS can be considered as the pre-definition file system. MFT is only used by system's organization and framework file system. This is called Metadata in NTFS. Thereinto the basic first 16 records are extremely important Metadata files used by operating system. These Metadata file names all start with "\$", the hidden files, cannot be listed in Windows NT/2000/ XP/2003 by dir order as ordinary files. But Microsoft Corporation provides an OEM TOOL called NFI.EXE, which may demonstrates the important Metadata files of NTFS

master file table. The result is as following:

```

C:\WINDOWS\system32\cmd.exe
H:\>nfi 1: !MORE
NTFS File Sector Information Utility.
Copyright (C) Microsoft Corporation 1999. All rights reserved.

File 0
Master File Table (<$Mft>)
  $STANDARD_INFORMATION (resident)
  $FILE_NAME (resident)
  $DATA (nonresident)
    logical sectors 728-735 (0x2d8-0x2df)
    logical sectors 6291464-6291615 (0x6000008-0x600009f)
  $BITMAP (nonresident)
    logical sectors 744-751 (0x2e8-0x2ef)

File 1
Master File Table Mirror (<$MftMirr>)
  $STANDARD_INFORMATION (resident)
  $FILE_NAME (resident)
  $DATA (nonresident)
    logical sectors 736-743 (0x2e0-0x2e7)

File 2
Log File (<$LogFile>)
  $STANDARD_INFORMATION (resident)
    
```

Number	Metadata	Function
0	\$MFT	MFT itself
1	\$MFTMirr	Part image of MFT
2	\$LogFile	Log file
3	\$Volume	volume file
4	\$AttrDef	Attribute definition list
5	\$Root	root directory
6	\$Bitmap	Bitmap file
7	\$Boot	boot file
8	\$BadClus	Bad cluster file
9	\$Secure	Secure file
10	\$UpCase	Capitalized file
11	\$Extended metadata directory	Extended Metadata directory
12	\$Extend\Reparse	Reparse Points file
13	\$Extend\UsnJrnl	Log changing file
14	\$Extend\Quota	Quota management file
15	\$Extend\ObjId	Object ID file
16~23		Reserved
24~		User files and directories

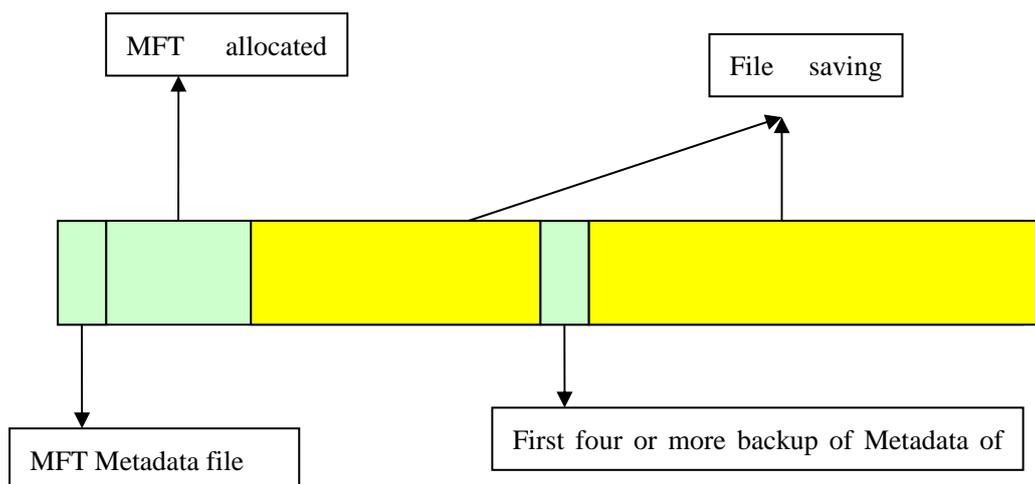
Each MFT recording corresponds with different file. If a file has many attributes or be dispersed into several fragments, it is probable that it needs more file records. By now, the first record that

stores its file record position is called “base file record”.

Records in \$MFT

	Metadata	Function
1 st record	\$MFT	\$MFT itself
2 nd record	\$MFTMirr	For the importance of \$MFT file, to insure reliability of file system structure, system sets a Mirror file (\$MFTMirr) in the beginning of its record, that is 2nd record of MFT.
3 rd record	\$LogFile	Log file. It is designed for the recoverability and security of NTFS. It records all the operations that affect NTFS volume construction.
4 th record	\$Volume	Volume file. It contains volume label, NTFS version and a mark bit that labels whether the disk is spoiled. Hereby NTFS can decide whether it needs Chkdsk to recover.
5 th record	\$AttrDef	Attribute definition table. It saves all file attributes that volume supports, and indicates whether they can be indexed or recovered.
6 th record	\$Root	It saves index of all files and directories in root directory. After visiting the first file, NTFS can retain MFT quote of the file. After that it can visit directly.
7 th record	\$Bitmap	Bitmap file. Service conditions of clusters in NTFS are all saved in it; thereinto each bit represents a cluster in volume, indicating it is free or assigned.
8 th record	\$Boot	Boot file. It is another important file system, storing boot program code of Windows NT/2000/XP/2003. It must be located in specified place of disk to boot system.
9 th record	\$BadClus	Bad cluster file. It records all destroyed cluster numbers, preventing system assigning and using it.
10 th record	\$Secure	Secure file. It saves security descriptor database of the whole volume. NTFS file and directory have their own security descriptors. To save space, NTFS saves the same security descriptors of them in the public file.
11 th record	\$UpCase	Upper case file.
12 th record	\$Extended metadata directory	Extended Metadata directory
13 th record	\$Extend\\$\\$Reparse	Reparse Points file
14 th record	\$Extend\\$\\$UsnJrnl	Log changing file
15 th record	\$Extend\\$\\$Quota	Quota management file
16 th record	\$Extend\\$\\$ObjId	Object ID file
17~24 th record		Reserved for further extend

Space allocation of MFT:



NTFS volume divides disk into two major parts, in which about 12% assigned to MFT to meet unceasing growth of the file quantity. In order to maintain the continuity of Metadata in MFT, MFT enjoys the sole right to this 12% space, the rest 88% space is assigned to save files. The leavings disk space contains all physical leaving space – including that of MFT. In another word, when files use up the storage space, Windows operating system would reduce the MFT space simply, and assign it to the file storage. When there is leavings space, it would be divided again to MFT. Although the system tries its best to maintain the dedication of MFT space, sometimes, it has to sacrifice. Sometimes although the MFT fragments are unendurable, it is unable to prevent its occurrence actually.

The process of NTFS's visiting the volume through MFT questionnaire is as following:

First, when NTFS visits a volume, it must be "loading" this volume: NTFS will check the boot file (file defined by \$Boot Metadata file), and find physical disk address of MFT.

Then, it can obtain mapping information from VCN to LCN in data attribute of file records, and save it in memory. This mapping information locates where MFT runs in disk.

The next, NTFS opens MFT records of several Metadata files, and then opens these files. If it is necessary, NTFS will start to execute file system recovery operation. After opening the leavings Metadata file in NTFS, users can visit this volume.

7. Files and folders of NTFS partition

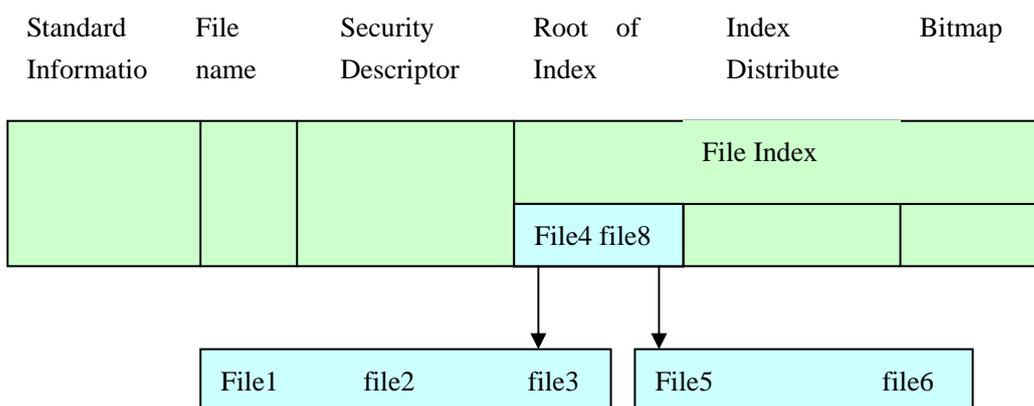
NTFS treats files as a unit of attribute/attribute value. That is the differences between NTFS and other file system. File data is attribute value without names. Other file attributes includes file name, file owner and file time mark, etc.

Standard Information	File or directory name	Security Descriptor	Data of Index	
----------------------	------------------------	---------------------	---------------	--

Each attribute is composed by single stream, namely simple character array. Strictly speaking, NTFS does not operate files, but read and write the attribute stream. NTFS operates in various ways: creation, deletion, read (byte scope) and writes in (byte scope). The read-write operation aims at unnamng attribute of file, as to those with name, it can operate by by named data stream.

Folder of NTFS is only a simple filename and index of file quotation number. If the directory attribute list is smaller than the length of a record, then all information of this folder is saved in MFT record. As to folders bigger than record, it uses B+ tree to manage and uses an indicator to point to an extent cluster. This cluster is used to save attributes of folders that cannot be saved in MFT.

NTFS directory

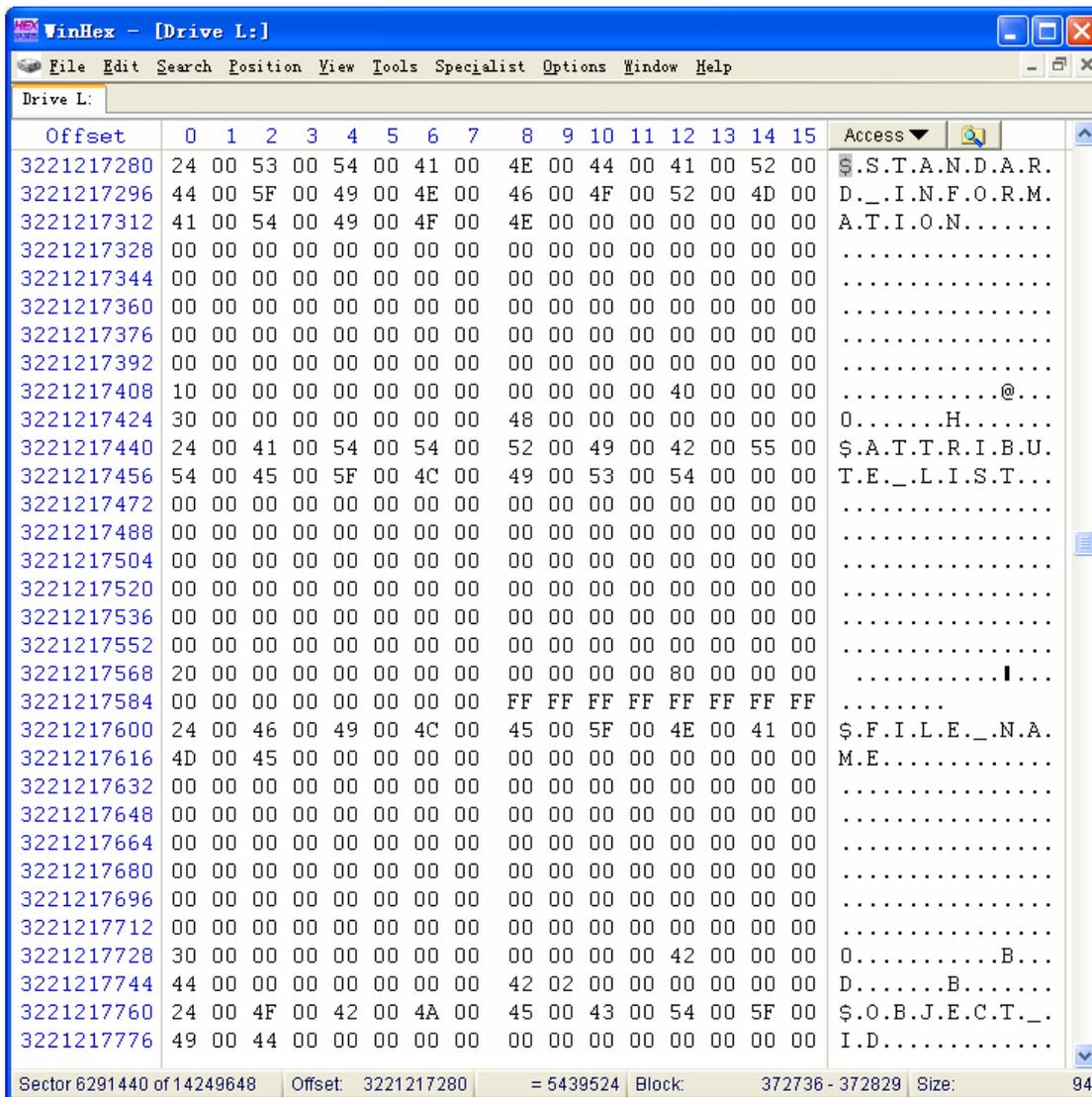


8. Analysis of important metafile in NTFS file system - \$MFT file analyzes

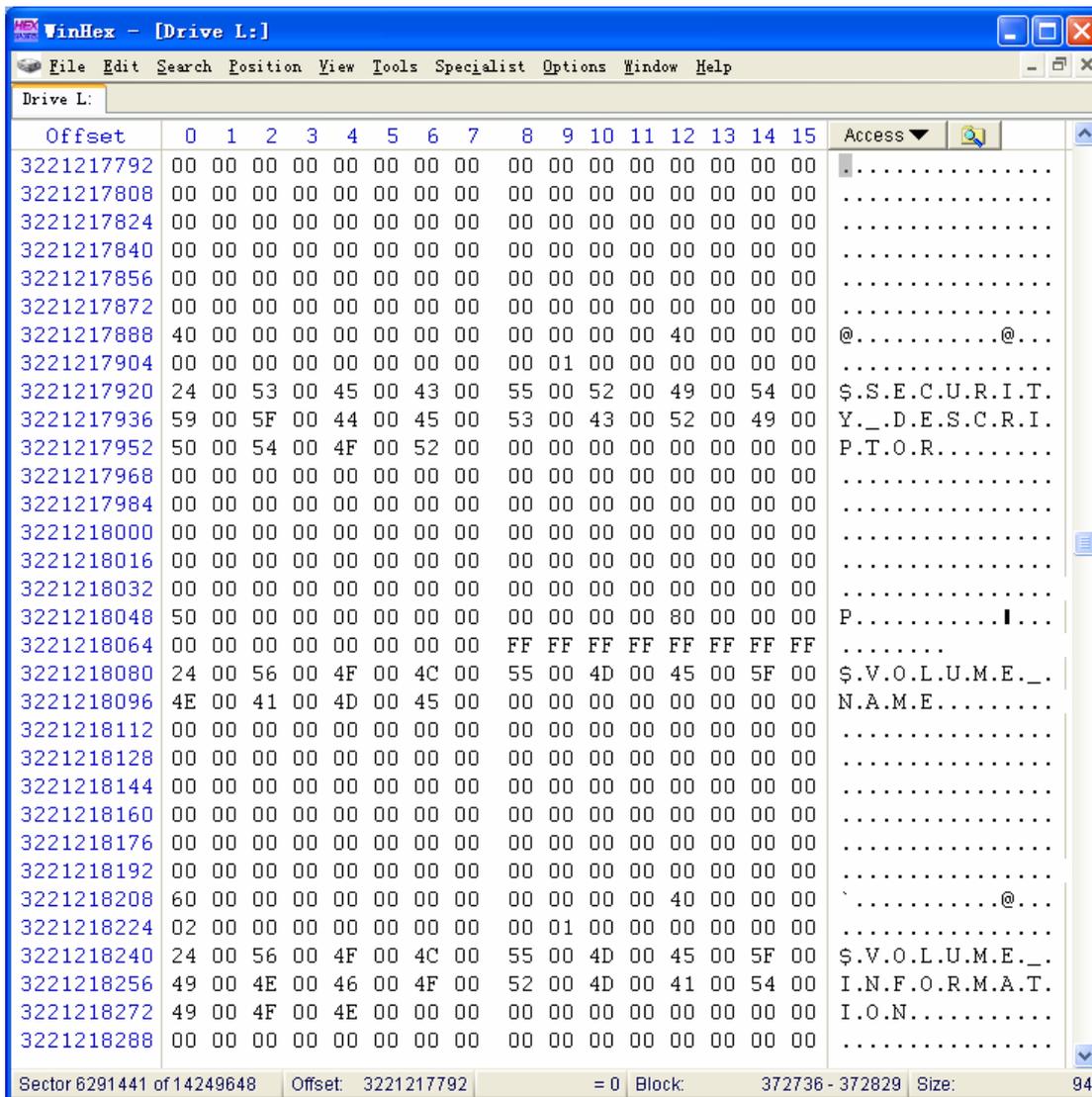
\$MFT file analysis

Metadata \$MFT is the most important file of NTFS, which records all situations of all files and directories, including the volume information, start files, \$MFT file itself etc. that located in the volume. It also records information like filenames, security attribute, file size, storage location, etc, all of which is similar to FAT+FDT function in FAT file system. It stores much more file attributes than FAT+FDT.

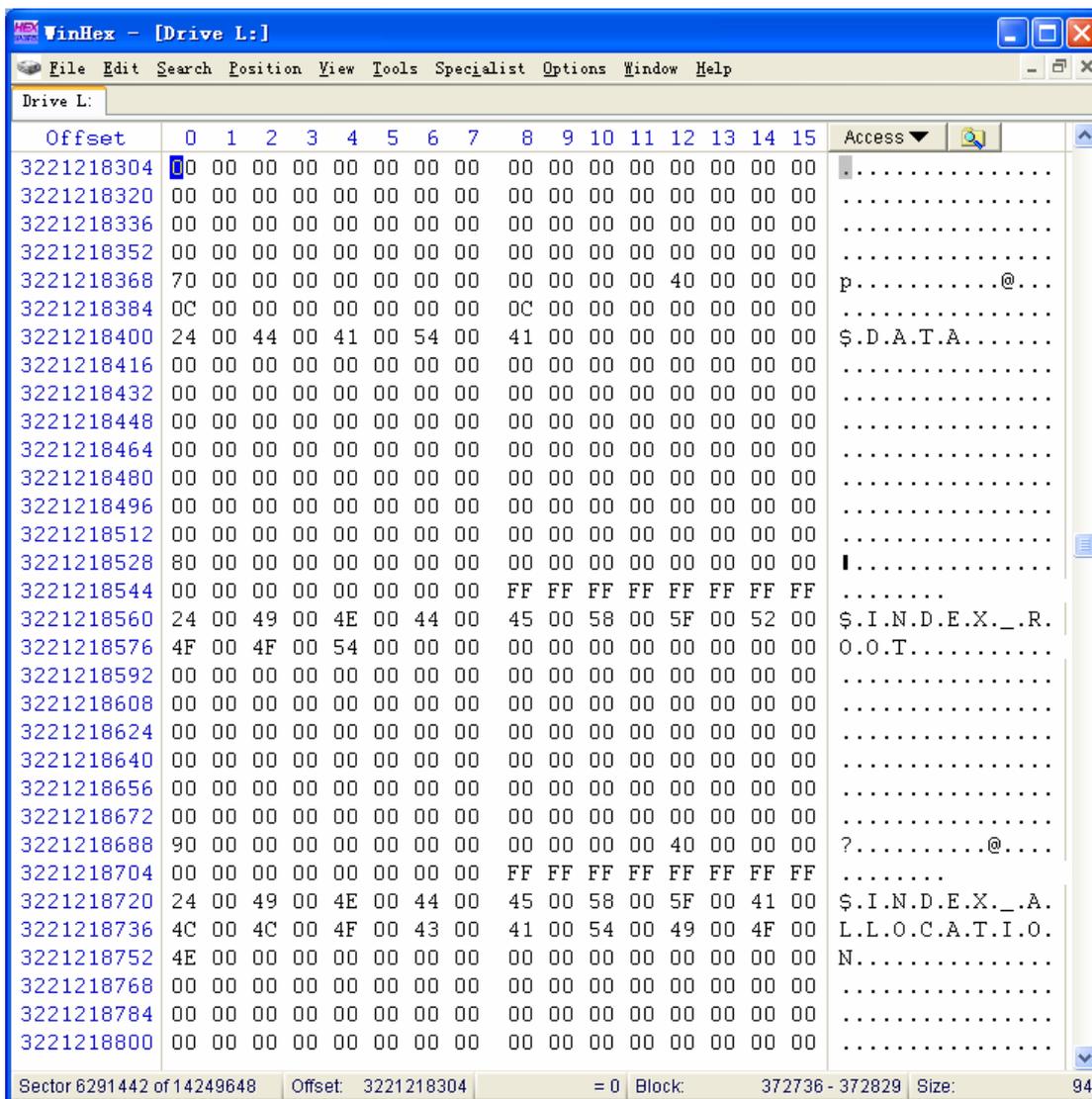
Metadata \$MFT is composed by series of file records, while each file record is composed by the recording head and the attribute part, ended by "FF FF FF FF". The general size is 1KB, or a cluster size (generally this is bigger). Its first sector content is as the following chart:



Content A of \$AttrDef:



Content B of \$AttrDef:



The content C of \$AttrDef:

From above three charts, we may see, attribute of standard information name is marked by 10H, attribute name in attribute list is marked by 20H, attribute of filename is marked by 30H and so on. Saving attribute names in common use in a file can greatly save system expense.

Each attribute is divided into two parts: Standard attribute head and content. Although in attribute list of file records, these two parts of attribute are recorded in inverted order, to better understand it, we will introduce them in this order:

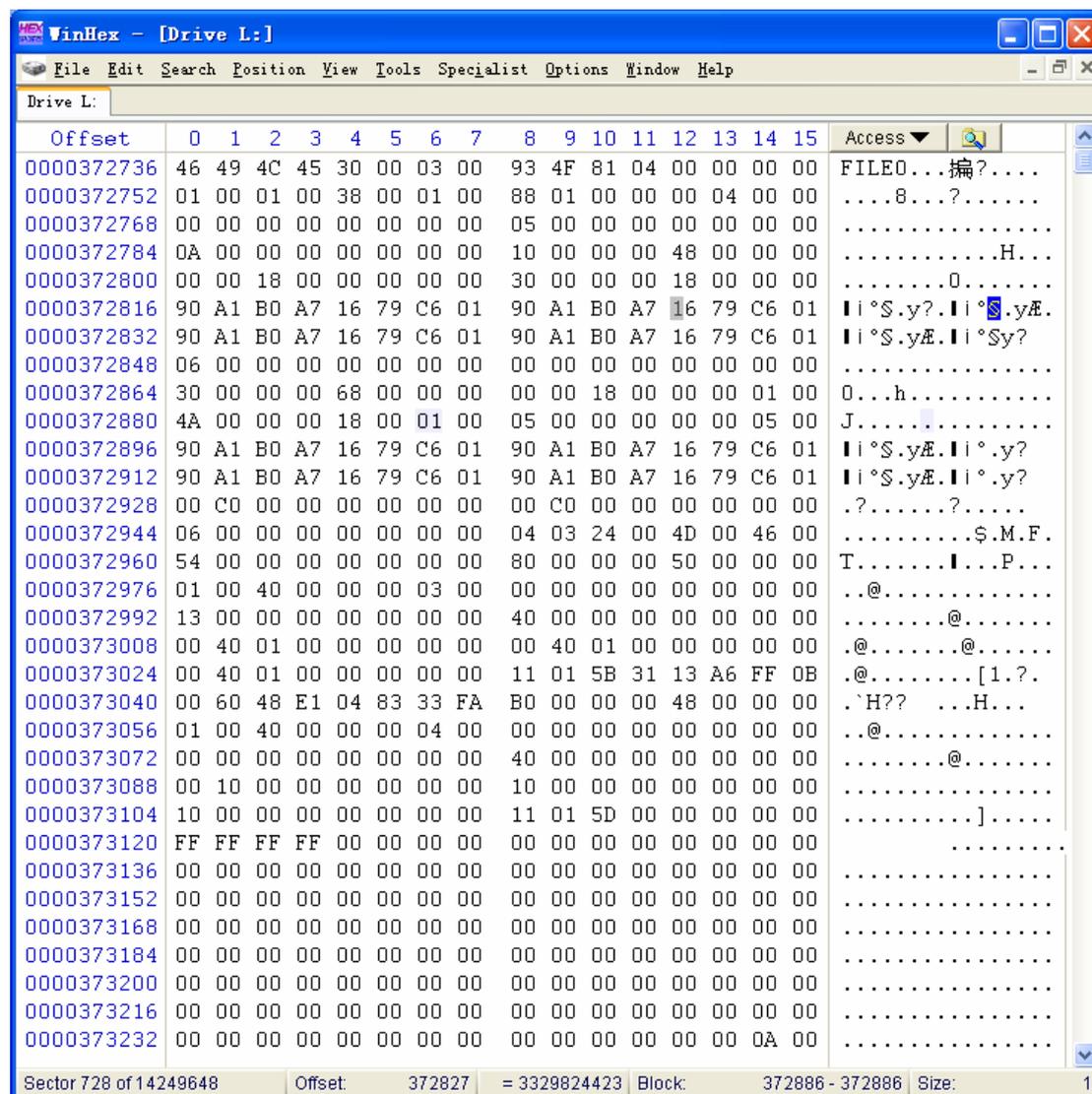
Content part: Its structure always starts with attribute name (the length is N byte). After attribute name, it defines this attribute as resident or not. When data stream of file attribute is stored after its attribute name, it is the resident attribute. By this way, it can provide a better access to files attributes with small and unchangeable flows. If a file attribute is not resident, its stream is saved in one or more extends or runs. The run is a continual region in logical cluster number. In order to visit the runs, NTFS saves a table named run list following the file attribute name.

Head part: its structure is showed as the following chart

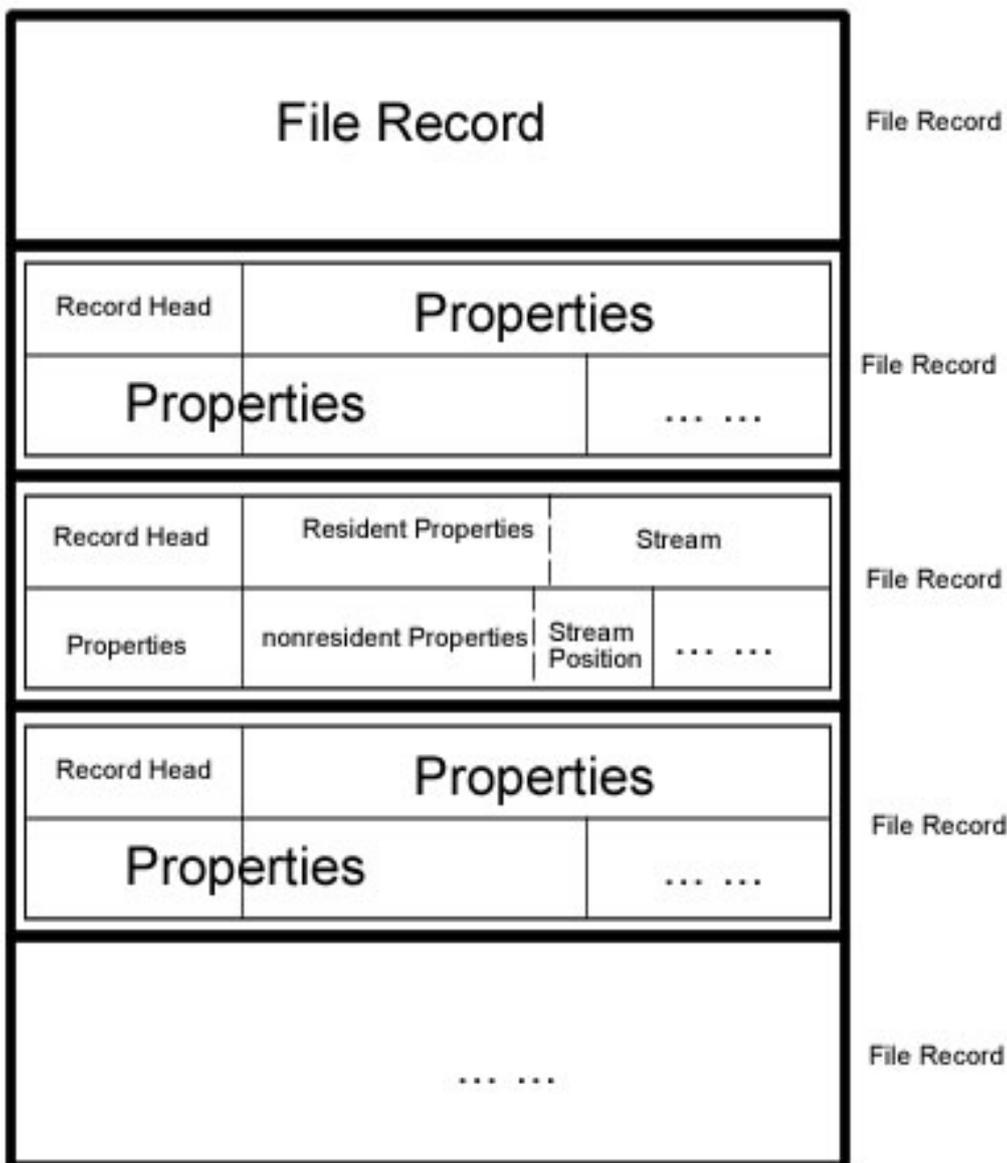
Head structure of attribute

Offset	Size	description
0x00	4	Type
0x04	4	Length
0x08	1	Non-residentflag
0x09	1	N=Name length
0x0a	2	Offset to the content part
0x0c	2	Compressed flag
0x0e	2	Identificator
The following part is propitious to resident porperty		
0x10	4	Length of the stream
0x14	2	Offset to the stream
0x16	2	Indexedflag

File records of \$MFT:



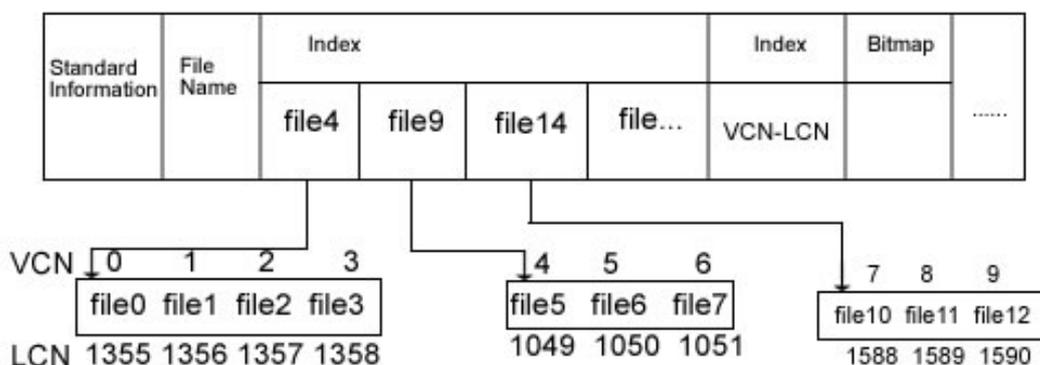
Structure of \$MFT



9. NTFS index record and contents

In NTFS, the file directory is merely a filename index. NTFS organizes the filenames in a special way for fast access. When creating a directory, NTFS must index the filenames in it.

Root directory files index:

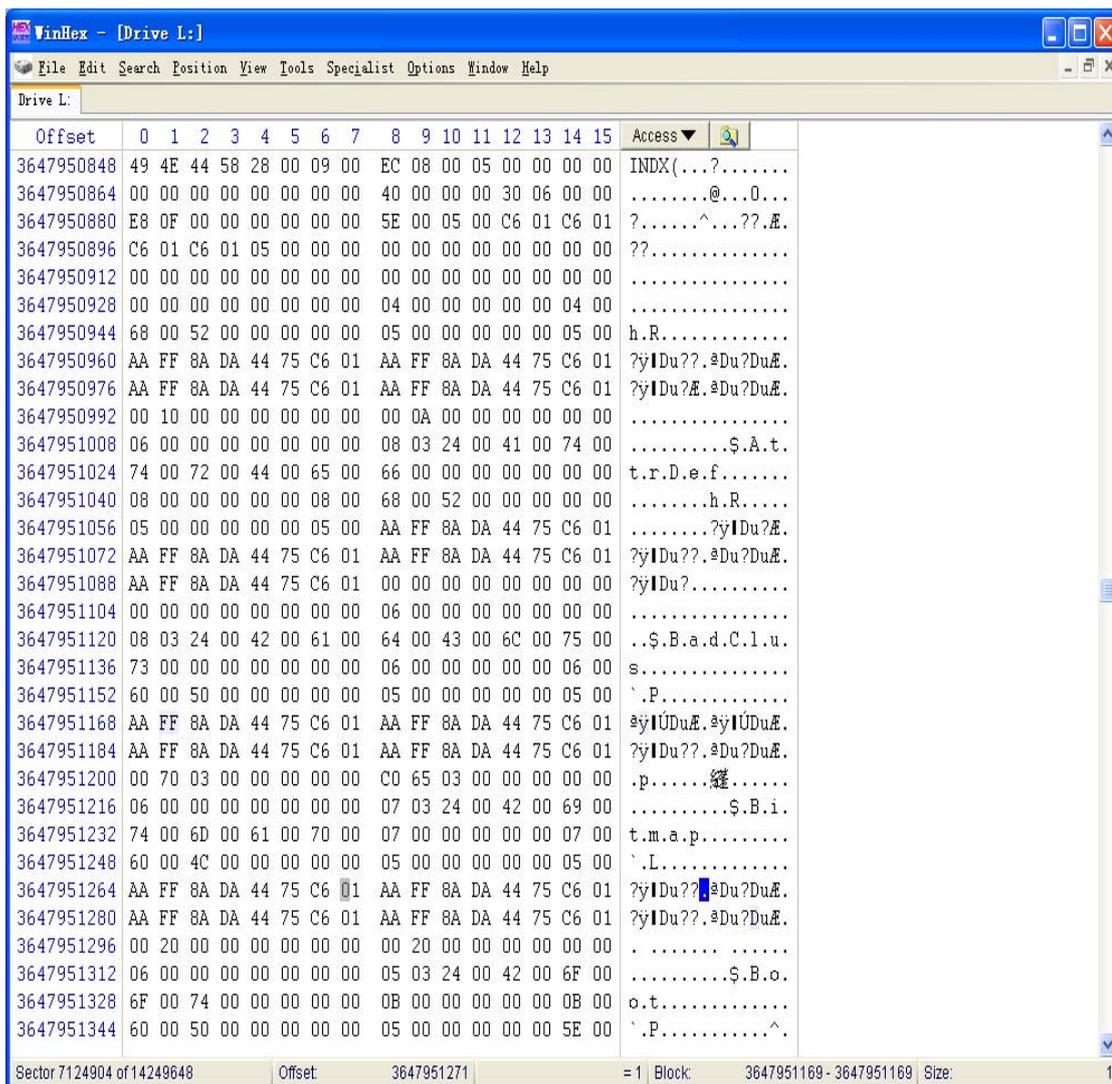


MFT record of a directory sorts filenames and sub-directories in it, and preserves in the index root attribute. However, for a big directory, filename actually is stored in index cache with a fixed organization filename of 4KB. The index cache is realized by the B+ tree data structure. The B+ tree is one of balance trees. For data in disk, the balance tree is an ideal sorting organization form, which reduces the times that searching an entry to the lowest. The root index attribute contains B+ tree's first level (root sub-directory) and point to the index cache of the next level (major sub directories, possibly files).

The above chart only shows filenames and the index cache in the root index attribute. But each entry in index also includes information on file quotation that describes where description file is in MFT, file time and file size and so on. NTFS duplicates the time mark and file size information according to file MFT records. This technique need write renew information into two places, is troublesome. However, it still is a good way to fasten browsing directories. Because it may display each file's time and size on the condition of that the file system does not open any file in directory.

The index assignment attribute contains VCN to LCN mapping in index cache, and bitmap attribute tracks that in index cache which VCNs are in use which are not. The above chart shows that each file entry occupies a VCN, but in fact many file entries are packed in the same cluster. Each index cache of 4KB size may contain 20 to 30 file entries.

In NTFS, directory is also a kind of file using file record to manage. Take an example of root directory, its file record are as the following chart:



From the above chart we may see, the root directory file record is also composed by the standard record head and the attribute, this directory has some attributes such as 0x10/0x30/0x40/0x50/0x90/0xa0/0xb0 and so on, which are respectively standard attribute, filename attribute, object ID attribute, security descriptor attribute, index root attribute, index assignment attribute, bitmap attribute.

X. Dynamic disk introduction

In computing, a **redundant array of independent disks**, also known as **redundant array of inexpensive disks** (commonly abbreviated **RAID**) is a system which uses multiple hard drives to share or replicate data among the drives. Depending on the version chosen, the benefit of RAID is one or more of increased data integrity, fault-tolerance, throughput or capacity compared to single drives. In its original implementations (in which it was an abbreviation for "redundant array of *inexpensive* disks"), its key advantage was the ability to combine multiple low-cost devices using older technology into an array that offered greater capacity, reliability, speed, or a combination of these things, than was affordably available in a single device using the newest technology.

At the very simplest level, RAID combines multiple hard drives into a single logical unit. Thus, instead of seeing several different hard drives, the operating system sees only one. RAID is typically used on server computers, and is usually (but not necessarily) implemented with identically-sized disk drives. With decreases in hard drive prices and wider availability of RAID options built into motherboard chipsets, RAID is also being found and offered as an option in more advanced user computers. This is especially true in computers dedicated to storage-intensive tasks, such as video and audio editing.

The original RAID specification suggested a number of prototype "RAID levels", or combinations of disks. Each had theoretical advantages and disadvantages. Over the years, different implementations of the RAID concept have appeared. Most differ substantially from the original idealized RAID levels, but the numbered names have remained. This can be confusing, since one implementation of RAID 5, for example, can differ substantially from another. RAID 3 and RAID 4 are often confused and even used interchangeably.

The very definition of RAID has been argued over the years. The use of the term *redundant* leads many to split hairs over whether RAID 0 is a "real" RAID type. Similarly, the change from *inexpensive* to *independent* confuses many as to the intended purpose of RAID. There are even some single-disk implementations of the RAID concept. For the purpose of this article, we will say that any system which employs the basic RAID concepts to combine physical disk space for purposes of reliability, capacity, or performance is a RAID system.

1. Raid background

Norman Ken Ouchi at IBM was awarded U.S. Patent 4,092,732 titled "System for recovering data stored in failed memory unit" in 1978 and the claims for this patent describe what would later be termed RAID 5 with full stripe writes. This 1978 patent also mentions that disk mirroring or duplexing (what would later be termed RAID 1) and protection with dedicated parity (what would later be termed RAID 4) were prior art at that time.

In 1988, RAID levels 1 through 5 were formally defined by David A. Patterson, Garth A. Gibson and Randy H. Katz in the paper, "A Case for Redundant Arrays of Inexpensive Disks (RAID)".

This was published in the SIGMOD Conference 1988: pp 109–116. The term "RAID" started with this paper.

It was a particularly ground-breaking work in that the concepts are both novel and "obvious" in retrospect once they had been described. This paper spawned the entire disk array industry.

2. Realization of RAID

RAID can be implemented either in dedicated hardware or custom software running on standard hardware. Additionally, there are hybrid RAIDs that are partly software- and partly hardware-based solutions.

With a software implementation, the operating system manages the disks of the array through the normal drive controller (IDE/ATA, SCSI, Fibre Channel, etc.). With present CPU speeds, software RAID can be faster than hardware RAID, though at the cost of using CPU power which might be best used for other tasks. One major exception is where the hardware implementation of RAID incorporates a battery backed-up write back cache which can speed up an application, such as an OLTP database server. In this case, the hardware RAID implementation flushes the write cache to secure storage to preserve data at a known point if there is a crash. The hardware approach is faster than accessing the disk drive and limited by RAM speeds, the rate at which the cache can be mirror to another controller, the amount of cache and how fast it can flush the cache to disk. For this reason, battery-backed caching disk controllers are often recommended for high transaction rate database servers. In the same situation, the software solution is limited to no more flushes than the number of rotations or seeks per second of the drives. Another disadvantage of a pure software RAID is that, depending on the disk that fails and the boot arrangements in use, the computer may not be able to be rebooted until the array has been rebuilt.

A hardware implementation of RAID requires at a minimum a special-purpose RAID controller. On a desktop system, this may be a PCI expansion card, or might be a capability built in to the motherboard. In larger RAIDs, the controller and disks are usually housed in an external multi-bay enclosure. The disks may be IDE, ATA, SATA, SCSI, Fibre Channel, or any combination thereof. The controller links to the host computer(s) with one or more high-speed SCSI, Fibre Channel or iSCSI connections, either directly, or through a fabric, or is accessed as network attached storage. This controller handles the management of the disks, and performs parity calculations (needed for many RAID levels). This option tends to provide better performance, and makes operating system support easier. Hardware implementations also typically support hot swapping, allowing failed drives to be replaced while the system is running. In rare cases hardware controllers have become faulty, which can result in data loss. Hybrid RAIDs have become very popular with the introduction of inexpensive *hardware RAID controllers*. The hardware is a normal disk controller that has no RAID features, but there is a boot-time application that allows users to set up RAIDs that are controlled via the BIOS. When any modern operating systems are used, they will need specialized RAID drivers that will make the array look like a single block device. Since these controllers actually do all calculations in software, not hardware, they are often called "fakeraid". Unlike software RAID, these "fakeraid" typically cannot span multiple controllers.

Both hardware and software versions may support the use of a *hot spare*, a preinstalled drive which is used to immediately (and almost always automatically) replace a failed drive. This reduces the mean time to repair period during which a second drive failure in the same RAID redundancy group can result in loss of data.

Basic disk storage: Basic disk storage uses normal partition tables that can be supported by MS-DOS, Microsoft Windows 95, Microsoft Windows 98, Microsoft Windows Millennium Edition (Me), Microsoft Windows NT, Microsoft Windows 2000, Windows Server 2003 and Windows XP. A disk initialized for basic storage is called a basic disk. A basic disk contains basic volumes, such as primary partitions, extended partitions, and logical drives.

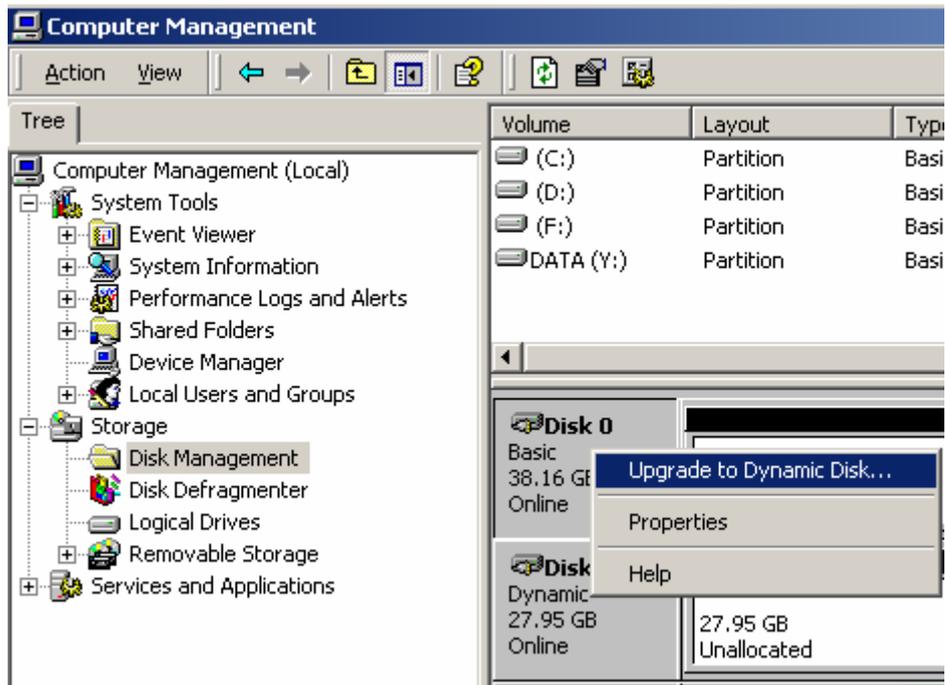
The dynamic disk storage: Dynamic storage is supported in Windows XP Professional, Windows 2000 and Windows Server 2003. A disk initialized for dynamic storage is called a dynamic disk. A dynamic disk contains dynamic volumes, such as simple volumes, spanned volumes, striped volumes, mirror volumes, and RAID-5 volumes.

Basic or dynamic disk may contain FAT16, FAT32 or NTFS partition or random combination of volumes. The disk system may contain random storage combination. But, all volumes on the same disk must use the same kind of storage.

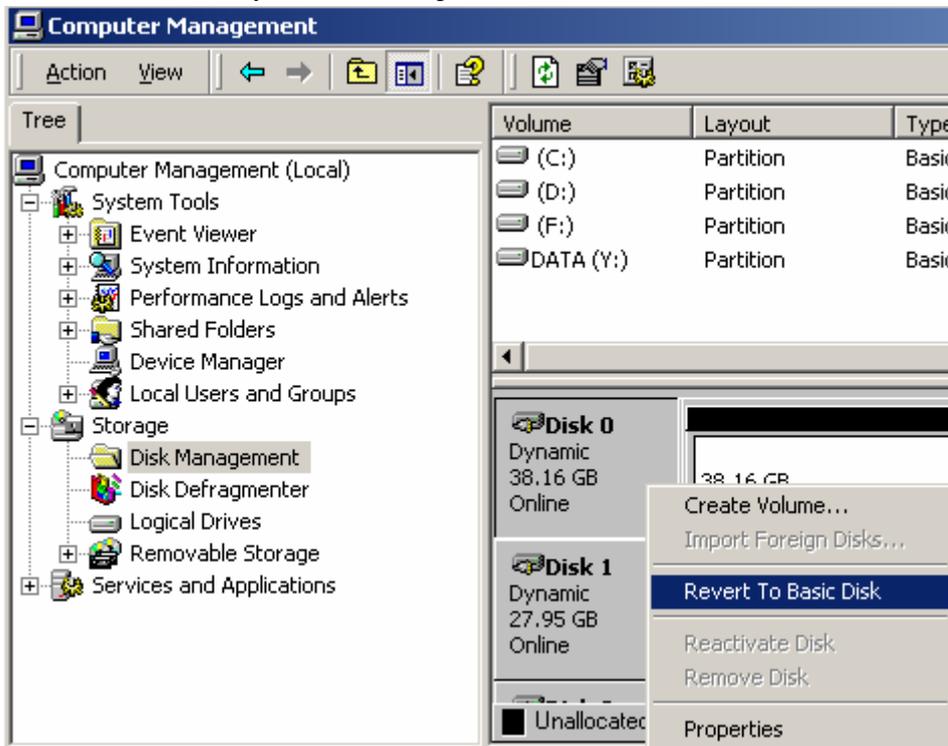
3. Transform basic disk into dynamic disk

For example, we may transform basic disk into dynamic disk by the disk management utility of Windows XP.

1. Log on by administrator or administrator component member
2. Click start, then click control panel, click administrative tool, then double click computer management, click disk management, click by the right key on the basic disk you want to upgrade, then the click upgrade to dynamic disk.
3. When the system prompts upgrading disk, click yes and click OK.



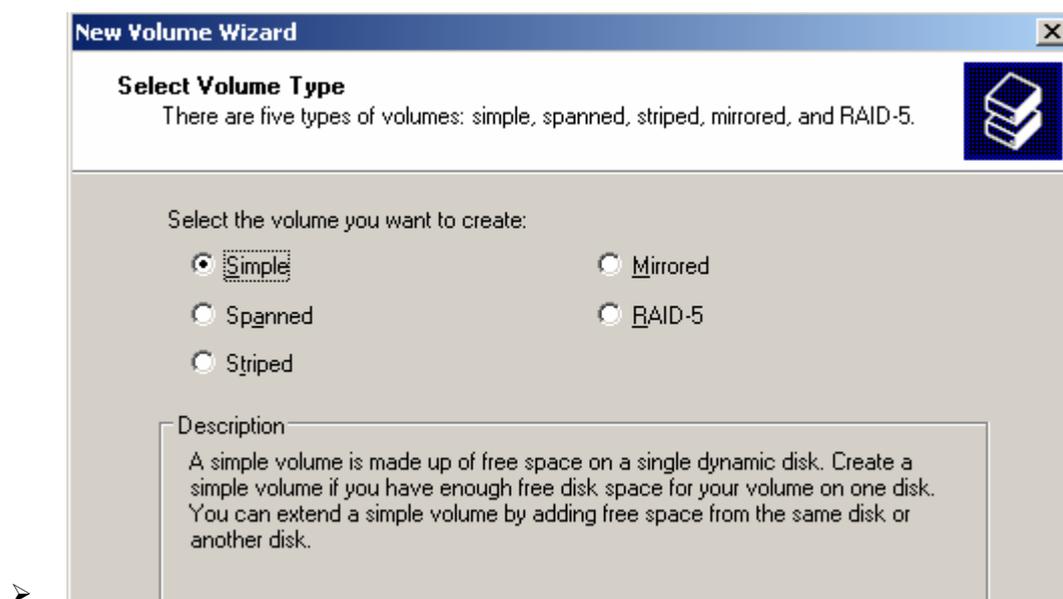
But after basic disk upgrading to dynamic disk, we only can locally access dynamic disk in Windows 2000 and Windows XP Professional or higher. In addition, after upgrading basic disk to dynamic disk, dynamic volume is unable to be changed as partition any more. In this case, we must delete all dynamic volumes on disk first, then transform dynamic disk back to basic disk. If you want to reserve the data, you need back up first, and move data to another volume.



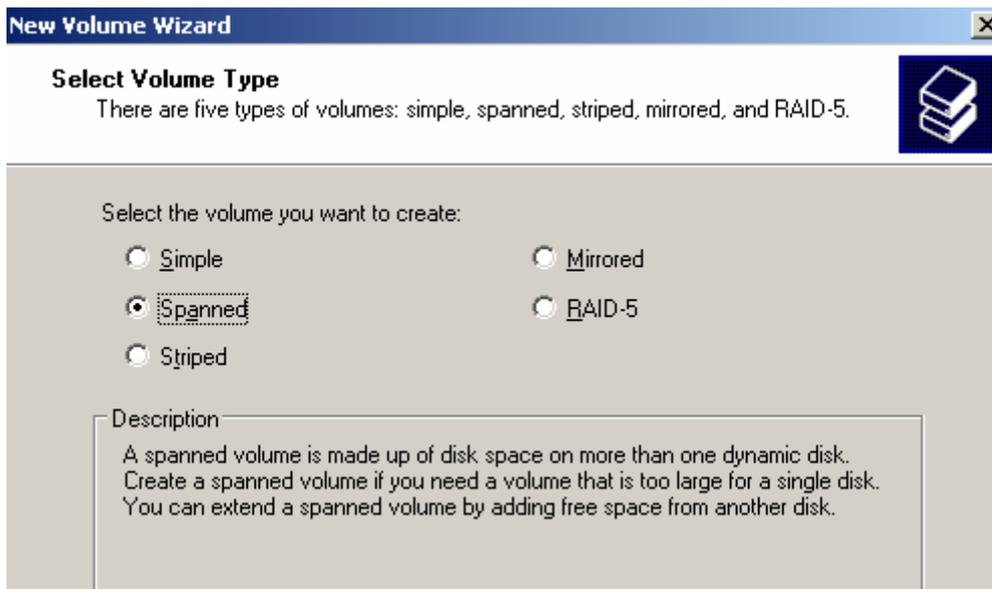
4. Some terms

A volume is a storage unit made from free space on one or more disks. It can be formatted with a file system and assigned a drive letter. Volumes on dynamic disks can have any of the following types: simple, spanned, mirror, striped, or RAID-5.

A simple volume uses free space from a single disk. It can be a single region on a disk or consist of multiple, consecutive regions. A simple volume can be extended within the same disk or onto additional disks. If a simple volume is extended across multiple disks, it becomes a spanned volume.

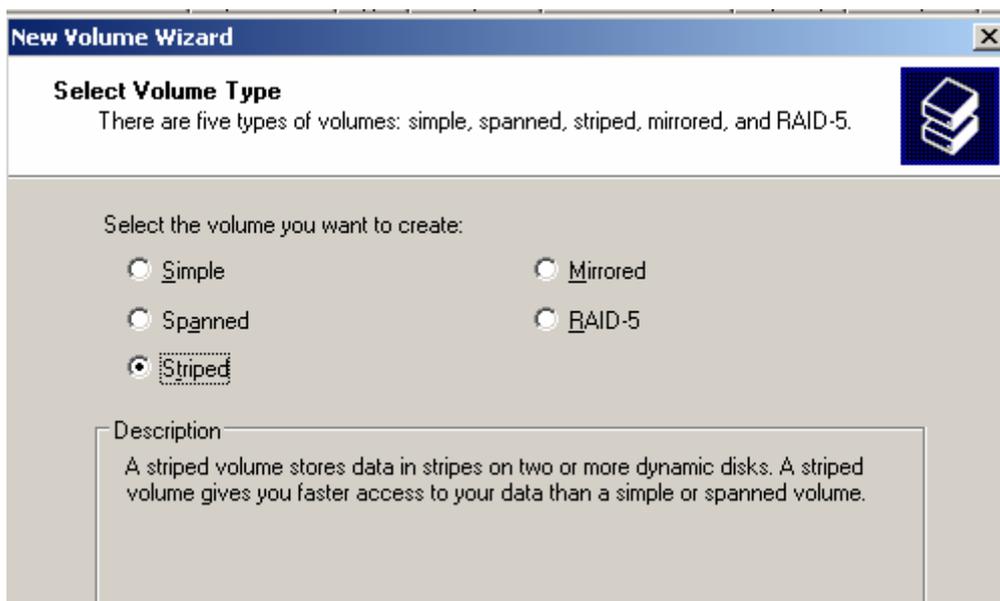


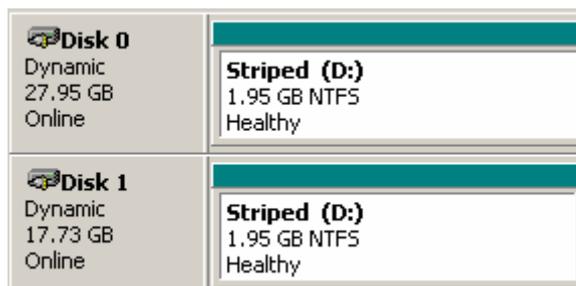
A spanned volume is created from free disk space that is linked together from multiple disks. You can extend a spanned volume onto a maximum of 32 disks. A spanned volume cannot be mirror image and is not fault-tolerant.



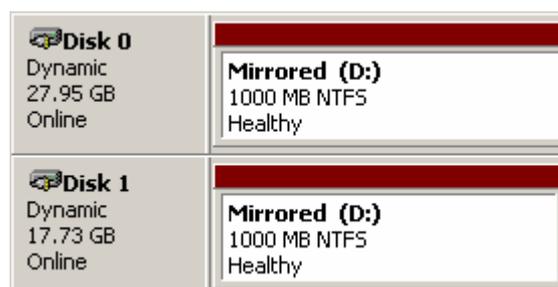
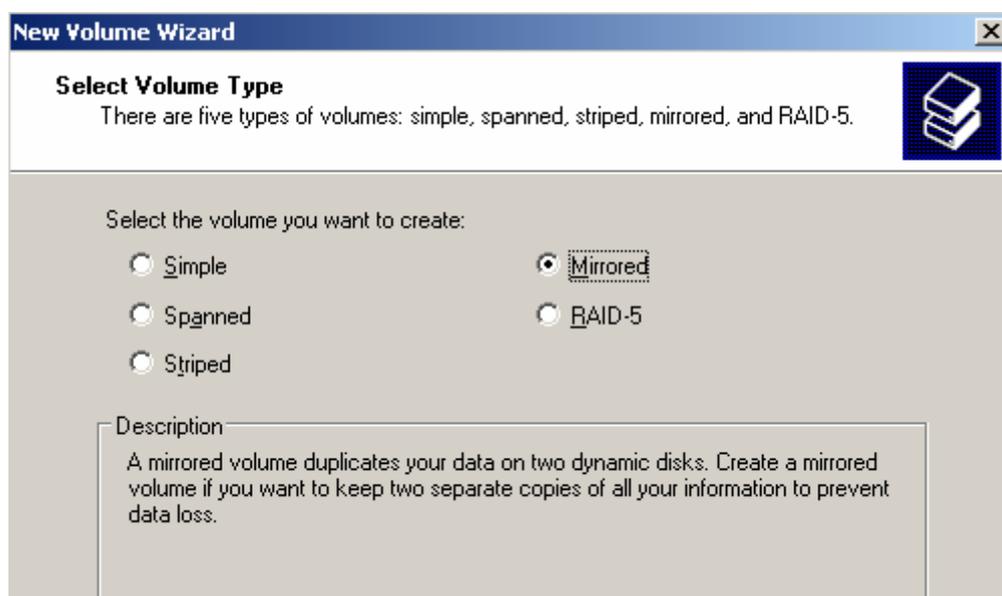
Disk 0 Dynamic 27.95 GB Online	Spanned (D:) 2.93 GB NTFS Healthy
Disk 1 Dynamic 17.73 GB Online	Spanned (D:) 1.95 GB NTFS Healthy

A striped volume is a volume whose data is alternately stored across two or more physical disks. The data on this type of volume is allocated alternately and evenly to each of the physical disks. A striped volume cannot be mirror image or extended and is not fault-tolerant. The **striped volume** is also called RAID-0.

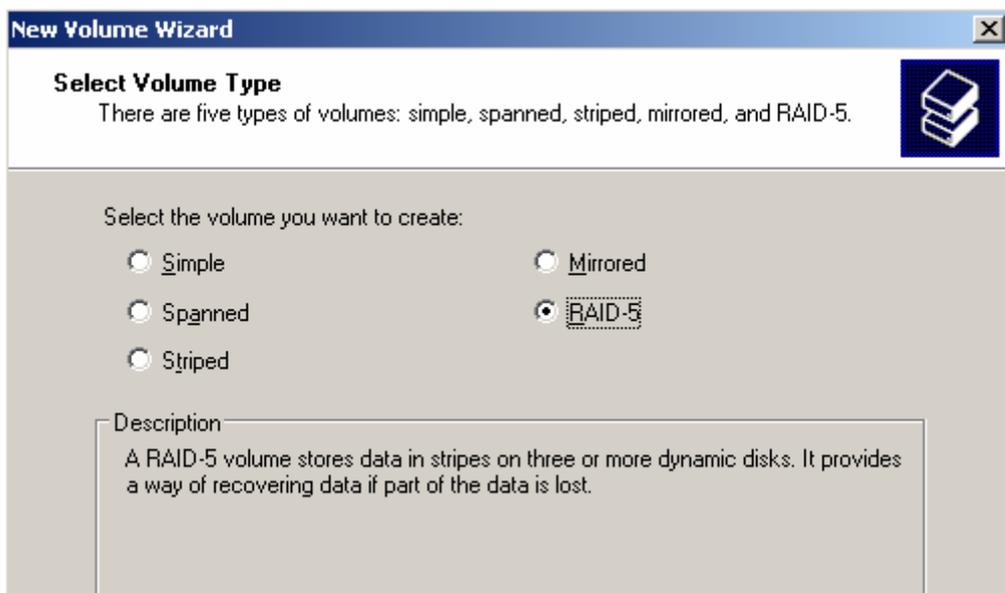




A mirror volume is a fault-tolerant volume whose data is duplicated on two physical disks. All of the data on one volume is copied to another disk to provide data redundancy. If one of the disks fails, the data can still be accessed from the remaining disk. A mirror volume cannot be extended. Mirroring is also known as RAID-1.



A RAID-5 volume is a fault-tolerant volume whose data is striped across an array of three or more disks. Parity (a calculated value that can be used to reconstruct data after a failure) is also striped across the disk array. If a physical disk fails, the portion of the RAID-5 volume that was on that failed disk can be re-created from the remaining data and the parity. A RAID-5 volume cannot be mirror or extended.



Disk 0 Dynamic 27.95 GB Online	Raid-5 (D:) 888 MB NTFS Healthy
Disk 1 Dynamic 17.73 GB Online	Raid-5 (D:) 888 MB NTFS Healthy
Disk 2 Dynamic 38.16 GB Online	Raid-5 (D:) 888 MB NTFS Healthy

5.Characteristics of Dynamic disk

Compared to basic disk, dynamic disk has following characteristic:

It may change disk capacity at random. You can change the capacity of disk without losing data and restarting computer. That of basic disk may lose data.

Disk space limitation. Dynamic disk may be extended to incontinuous disk space in disk. It also may create volume collection that crosses disks and gather several disks into a big volume collection. While basic disk partition must be in a continual space on the same disk which cannot cross disks. The partition maximum capacity is disk capacity.

Volume collection or partition number. Dynamic disk does not have limit about the volume collection number on a disk. But basic disk only can divide into 4 or 3 primary partitions and 1 extended partition mostly in a disk. But extended partition may contain several logical drives.

Disk allocation information. Disk allocation information of dynamic disk is stored in disk, not in registry or other place that is not good for renew. At the same time disk allocate information can be copied to other dynamic disks. Thus, it is convenient for dynamic disk to be transplanted between different machines.

Visiting Speed of disk. Basic read-write speed is determined by hardware, which cannot promote disk efficiency without extra extend. But dynamic disk may creat striped volume to deal with read-write operations to many disks at the same time, thus promoting disk efficiency.

Fault-tolerant function of disk. The basic disk has no fault-tolerance, nor provides data protection measures. If the disk is out of order without backup, data will be lost. While dynamic disk may creat the mirror volume, data would be “mirrored” automatically to mirror volume. thus reducing data loss to the lowest. At the same time, parity check provided by Raid-5 also can protect data from losing.

6. Dynamic disk management

We may manage disk through Disk Management of Computer Management. Disk Management supports not only basic disk, but also dynamic disk. We may use upgrading guide to transform basic disk to dynamic disk. And we can simultaneously use basic disk, dynamic disk and different file systems (FAT16, FAT32, NTFS) on the same computer system. To transform dynamic disk to basic disk, it need delete all volumes in disk.

Microsoft Windows doesn't support dynamic disk on notebook computer, removable storage, USB or disk with FireWire interface. Basic disk and dynamic disk can not be mixed in the same disk. For dynamic disk, Windows supports following operating system:

Operating system	Dynamic Disk
Windows 9X/Me,DOS	☆
Windows NT 4.0	★
Windows 2000	★
Windows XP Home Edition	☆
Windows XP Professional	★
Windows Server 2003	★

Notes: ★ support
 ☆ not support

SCSI Short for small computer system interface, a parallel interface standard used by Apple Macintosh computers, PCs, and many UNIX systems for attaching peripheral devices to computers. Nearly all Apple Macintosh computers, excluding only the earliest Macs and the recent iMac, come with a SCSI port for attaching devices such as disk drives and printers. SCSI interfaces provide for faster data transmission rates (up to 80 megabytes per second) than standard

serial and parallel ports. In addition, you can attach many devices to a single SCSI port, so that SCSI is really an I/O bus rather than simply an interface.

Hardware realization

RAID card It can realize RAID function. It has a RAID control chip and manages data alone. It is fast and has certain fault-tolerance.

SCSI Short for small computer system interface, a parallel interface standard used by Apple Macintosh computers, PCs, and many UNIX systems for attaching peripheral devices to computers. Nearly all Apple Macintosh computers, excluding only the earliest Macs and the recent iMac, come with a SCSI port for attaching devices such as disk drives and printers. SCSI interfaces provide for faster data transmission rates (up to 80 megabytes per second) than standard serial and parallel ports. In addition, you can attach many devices to a single SCSI port, so that SCSI is really an I/O bus rather than simply an interface.

Hard disk In order to make the best of RAID, the speed request for hard disk is generally high. We may choose interfaces like SCSI, SATA and so on to guarantee the visit speed.

Solution scenario based on hardware is better than that on software in technical and service performance. Specifically, in abilities to detect and recover multi-bit errors, RAID protected bootable array, error disk automatic detect, leaving space replace, array reconstruction, common or appointed leaving space and colorful code alarm etc. the former is better than the latter. Moreover, it has abilities to erect multi-RAID, remotely detect and manage for multi-operating system. RAID based on hardware has high security and low CPU occupancy rate.

RAID levels

There are a number of different RAID levels:

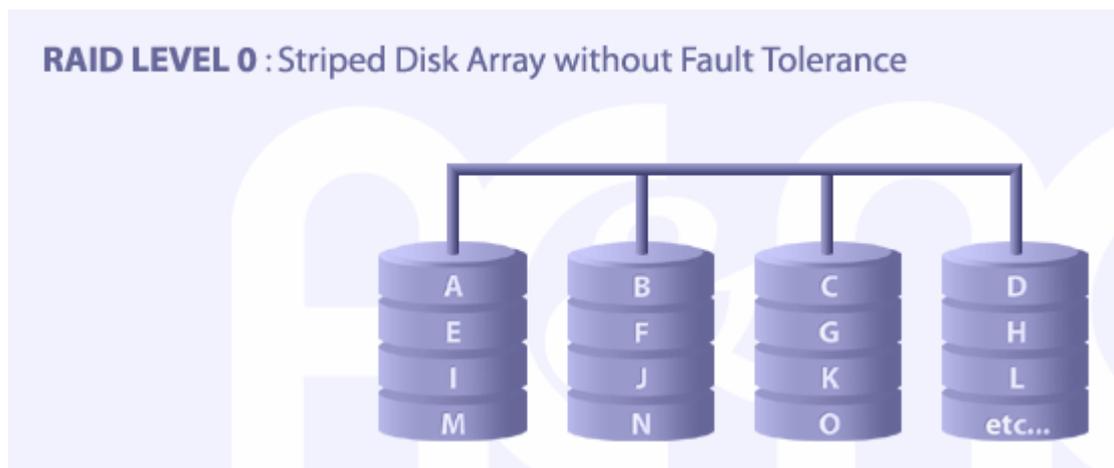
7. RAID 0 -- Striped Disk Array

Striped Disk Array without Fault Tolerance: Provides data striping (spreading out blocks of each file across multiple disk drives) but no redundancy. This improves performance but does not deliver fault tolerance. If one drive fails then all data in the array is lost.

Minimum number of drives: 2

Strengths: Highest performance.

Weaknesses: No data protection; One drive fails, all data is lost.



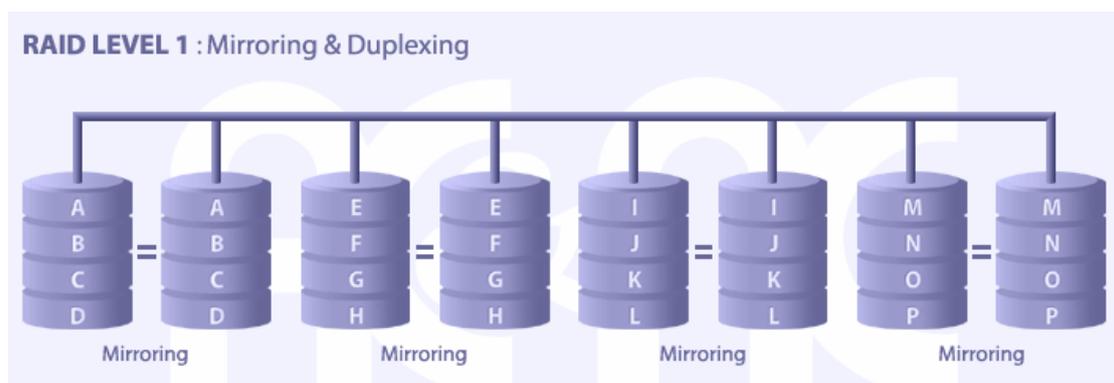
RAID 1 Disk mirroring.

Mirroring and Duplexing: Provides disk mirroring. Level 1 provides twice the read transaction rate of single disks and the same write transaction rate as single disks.

Minimum number of drives: 2

Strengths: Very high performance; Very high data protection; Very minimal penalty on write performance.

Weaknesses: High redundancy cost overhead; because all data is duplicated, twice the storage capacity is required.



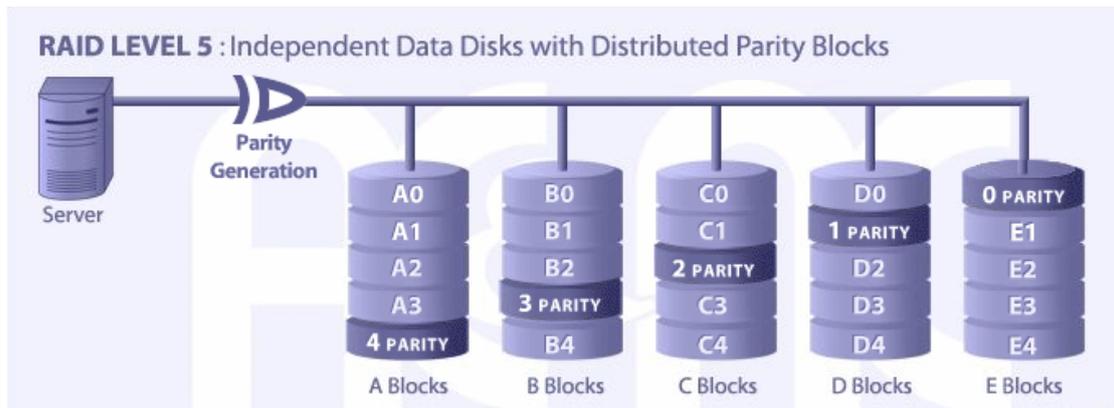
RAID 5 Block-level data striping with distributed parity.

Block Interleaved Distributed Parity: Provides data striping at the byte level and also stripe error correction information. This results in excellent performance and good fault tolerance. Level 5 is one of the most popular implementations of RAID.

Minimum number of drives: 3

Strengths: Best cost/performance for transaction-oriented networks; Very high performance, very high data protection; Supports multiple simultaneous reads and writes; can also be optimized for large, sequential requests.

Weaknesses: Write performance is slower than RAID 0 or RAID 1.



RAID 0+1

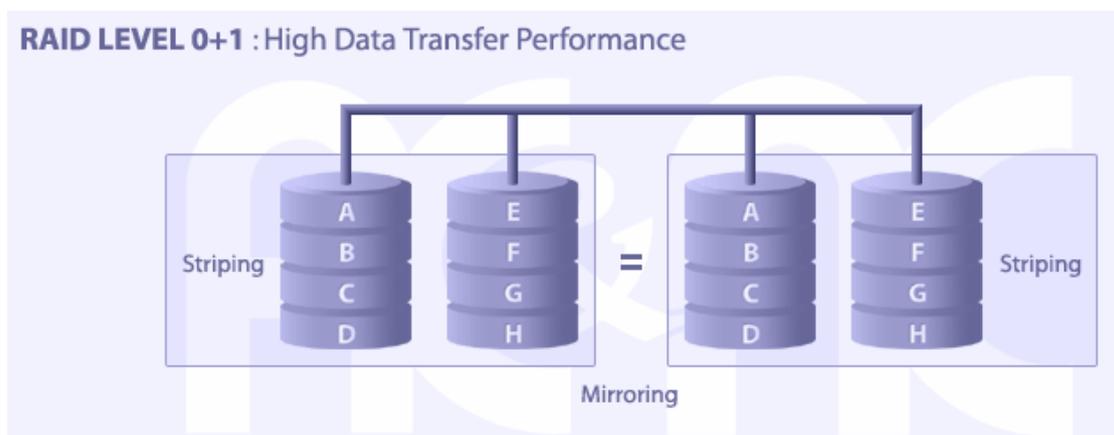
Combination of RAID 0 (data striping) and RAID 1 (mirroring).
Note that RAID 10 is another name for RAID (0+1) or RAID 0/1.

A Mirror of Stripes: Not one of the original RAID levels, two RAID 0 stripes are created, and a RAID 1 mirror is created over them. Used for both replicating and sharing data among disks.

Minimum number of drives: 4

Strengths: Highest performance, highest data protection (can tolerate multiple drive failures).

Weaknesses: High redundancy cost overhead; because all data is duplicated, twice the storage capacity is required; requires minimum of four drives.



From RAID1 to RAID5, whenever disk is damaged, we can extract the damaged and insert a new one (this needs hot-plugging in hardware), data will not be spoiled and the content of damaged disk can be reconstructed quickly. The reconstruction also can be completed by RAID hardware or RAID software. But RAID0 does not provide the error checking function, so someone said that it can not be regarded as RAID. Actually, that is why RAID0 is called RAID0 ---0 represents "none".

Choice of RAID

There are three factors that may influence your choice of RAID level: usability (redundant data), performance and cost. If you do not need usability, RAID-0 can perform best. If usability and performance are more important than cost, RAID-1 or RAID-10(it depends on the number of disks) is a good choice. If price is as important as usability and performance, you can choose RAID-3, RAID-30, RAID-5 or RAID-50(it depends on the transmission type and the number of disk drivers)

Strengths of RAID:

- Low cost, little power consumption, high transmission speed.
- Fault-tolerance function
- Data security

RAID applies for:

- E-Mail server
- Workgroup/file server
- Corporation server
- Storage LAN
- Internet news server
-

RAID failures

Nowadays, RAID is a quite good storage technique. It has advantages in storage capacity, storage security and storage speed. However, once there are some failures in RAID, the loss is usually tremendous.

Situations in which RAID information may lose:

The RAID array card breaks down

Physical failures of disk

Power cut

Extracting disk in wrong order

Reallocation RAID array information

Disk electric line falls off

High temperature of node caused by heat dissipation

Unstable voltage

Some methods for above questions

Not easily attempt rebuild, synchronization etc.

Not initialize

Mirror disk

Analyze mirror file, reconstruct data

Creat RAID information, analyze the construction of data

Some problem can be solved by software, such as RAID Reconstructor, R—Studio Data Recovery Wizard Professional 3.0 etc. These software can help recreate Raid information, analyze disk order, reconstruct array information, finally restoring Raid disk.

Case Study

XI. Case Study

1. Introduction of Data Recovery Wizard 3.0

Data Recovery Wizard is an advanced data recovery software. In Windows, this software can recover data on different storage media and partitions.

General Functions of Data Recovery Wizard 3.0

Deletedrecovery: This module works only with deleted files and allows to “undelete” them (another popular term is “unerase”). Intact file system is important for this module. If you know that there is something wrong with your file system (for example, you did not delete some folder/files but you cannot access them) or if you see something strange with Windows, you should use “AdvancedRecovery” module.

Formattedrecovery: A common data recovery situation is accidentally reformatting a partition. The FormatRecovery tool will allow you to recover files from a partition, which has been accidentally formatted or reinstalled. This type of recovery will ignore the existing file system structures and search for structures associated with the previous file system. If you are not satisfied with the result, you should use “AdvancedRecovery” module

AdvancedRecovery: you can use this function to recover your damaged system, deleted partitions, misoperation of HD and deletion caused by virus.

RawRecovery: The RawRecovery tool allows you to scan severely corrupted partitions for files with a file signature search algorithm. This tool will help you recover files from a partition with damaged directory structures.

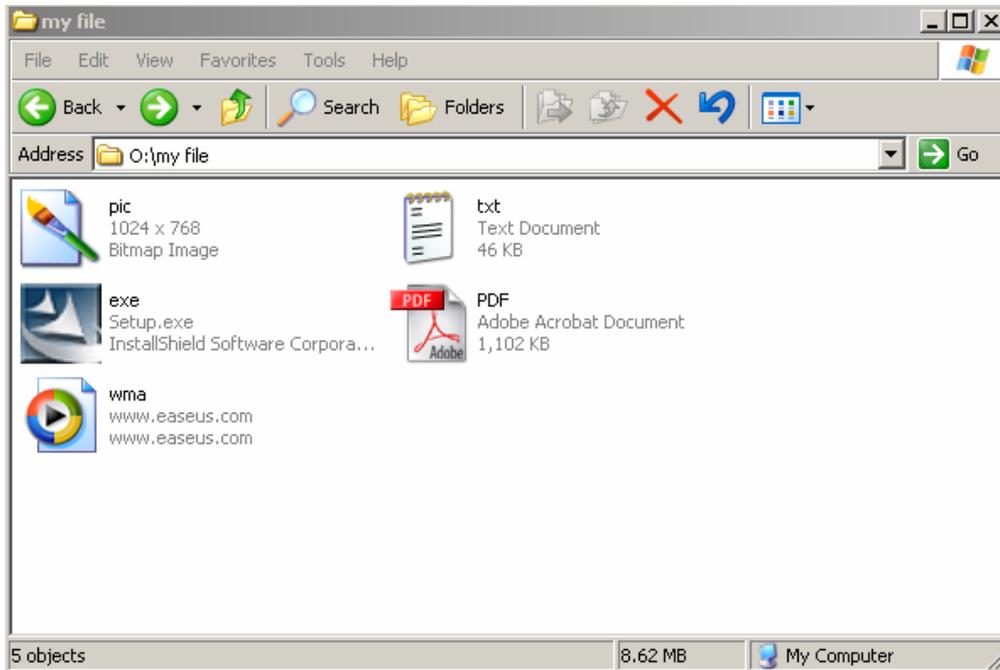
2. Matters needs attention before recovery

- (1) Never operate on partition (such as write and create file) where the data lost.
- (2) Please close any other application program when Data Recovery Wizard 3.0 is running.
- (3) Make sure that there is no physical failure (such as physical bad track) on the disk you are operating. If there is any problem, please stop running Data Recovery Wizard 3.0, and send your disk to maintenance station.
- (4) Do not save the recovered files to the original partition. You need make sure that there is enough free space to save the recovered data; also you can save your files to removable devices or network devices.

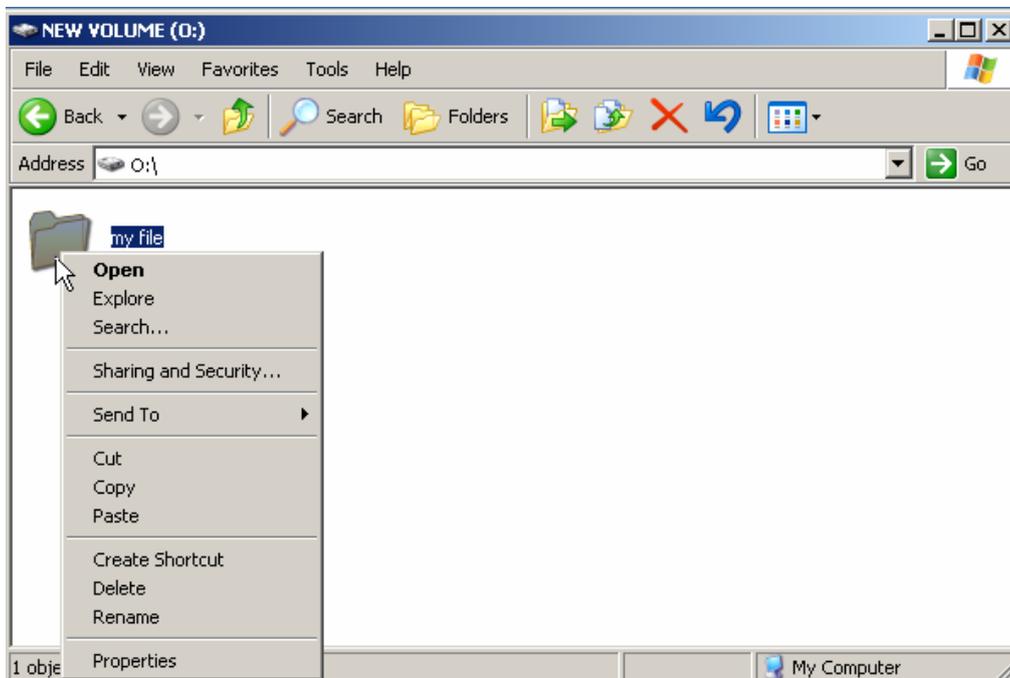
Here are some examples of using Data Recovery Wizard 3.0.

3. Deletedrecovery

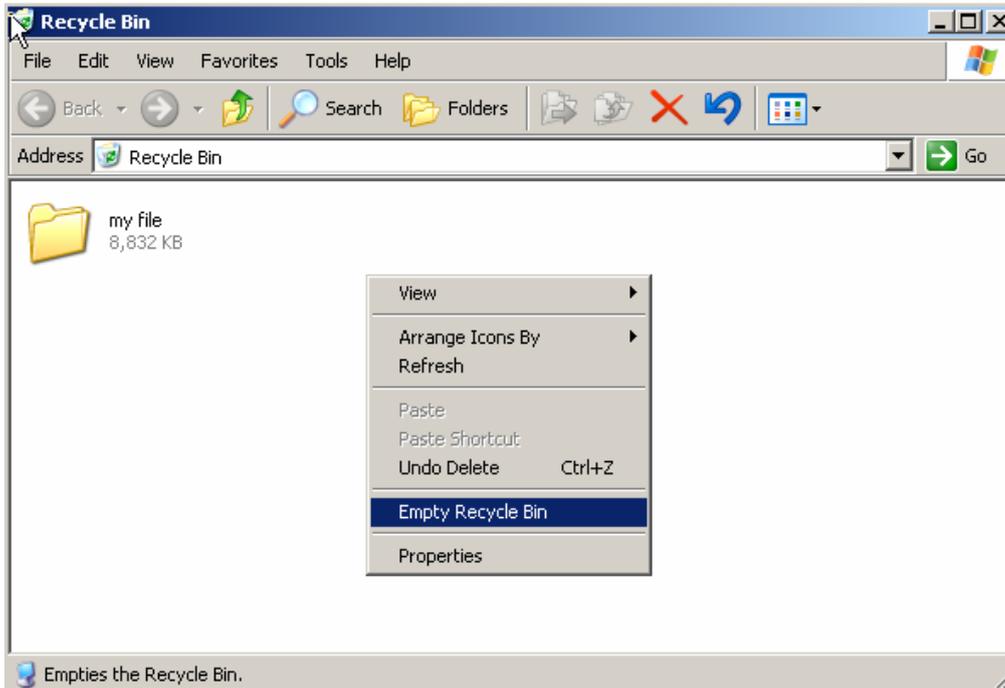
Eg: Here is a folder” my file”, whose contents are as following:



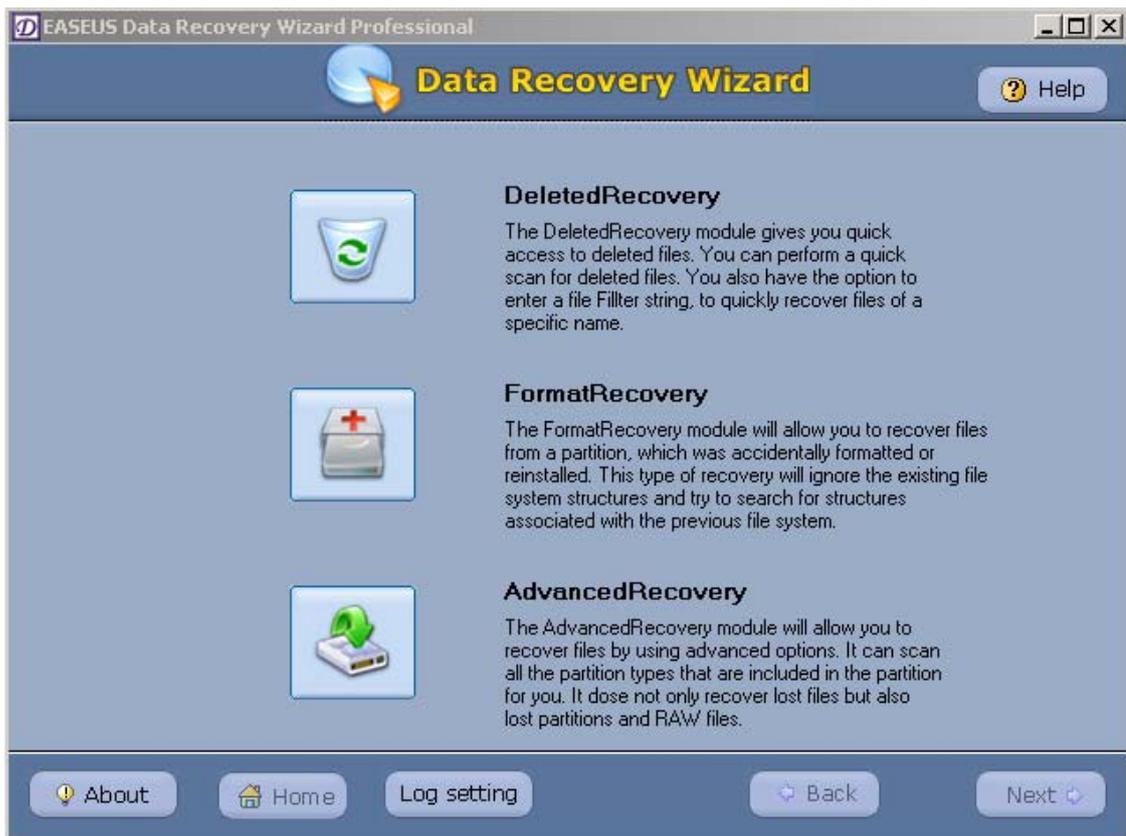
Delete it:



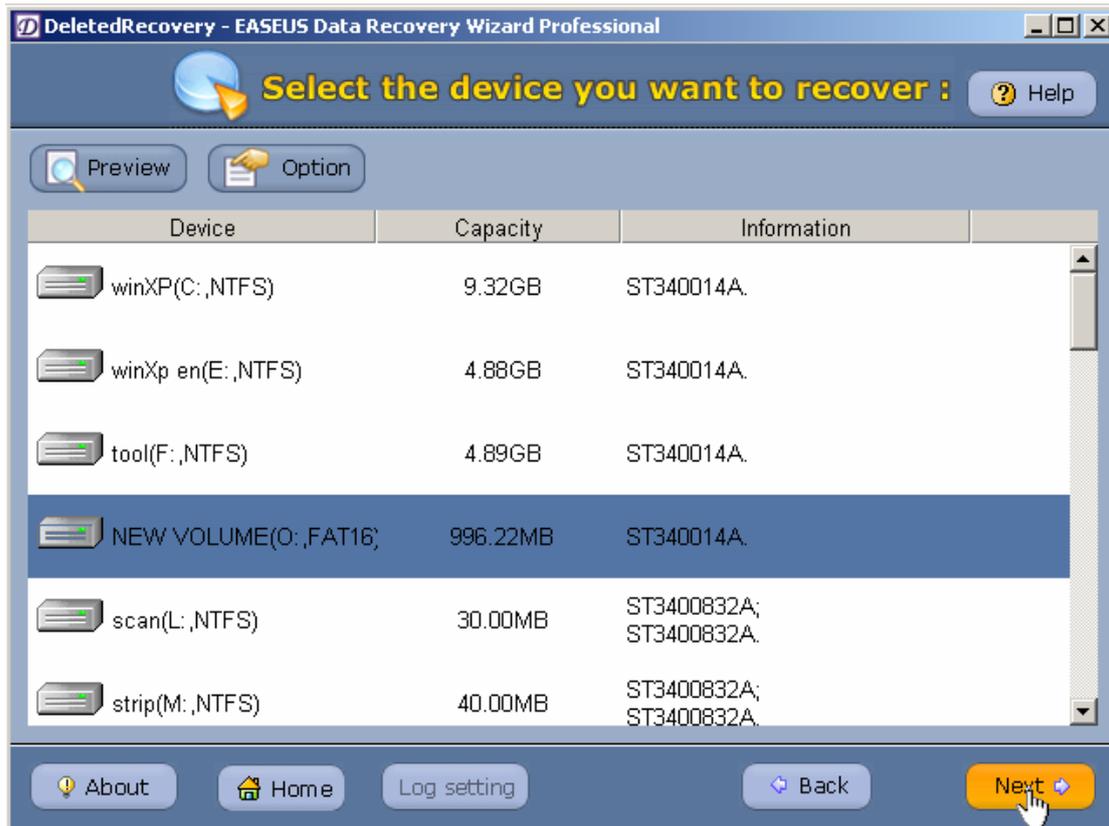
Empty your “recycle bin”



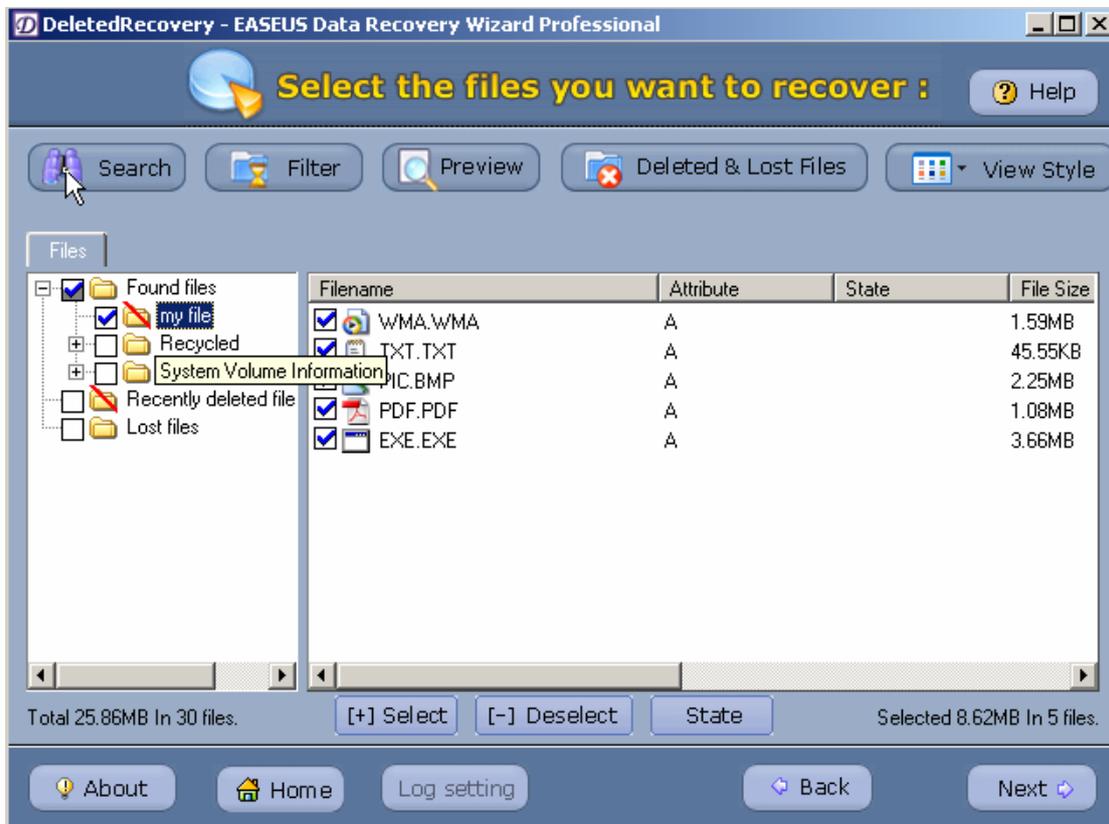
Run Data Recovery Wizard 3.0:



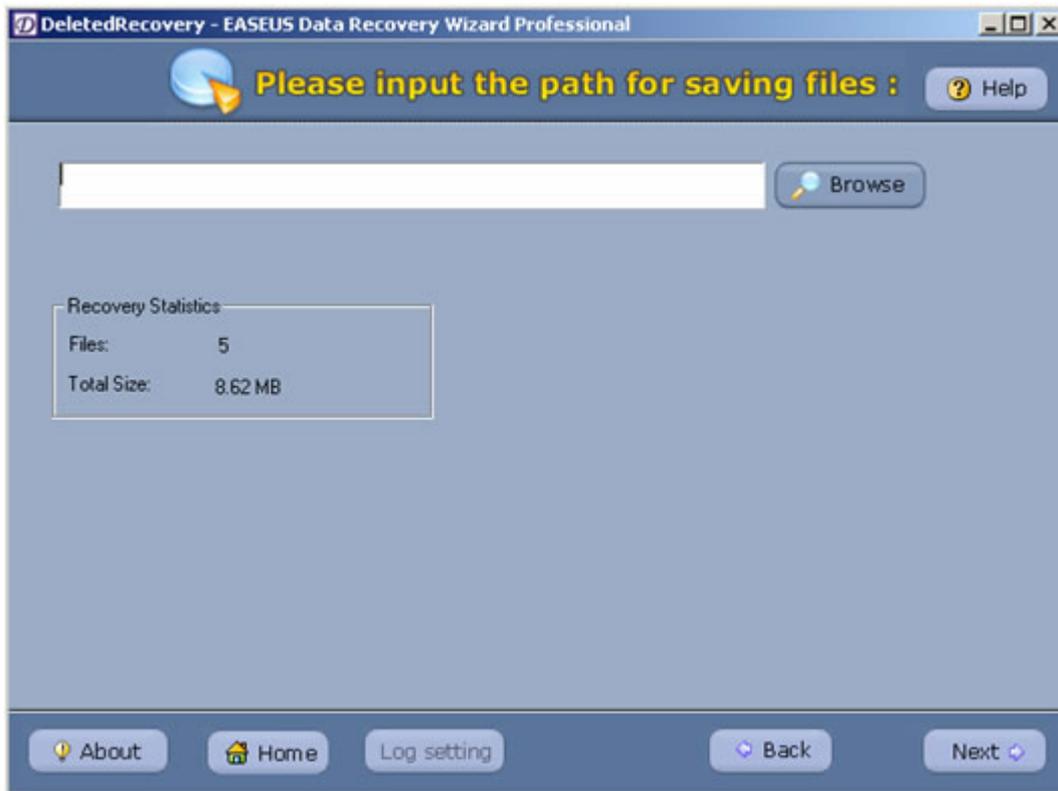
Select “deletedrecovery” then choose the partition that you want to recover.



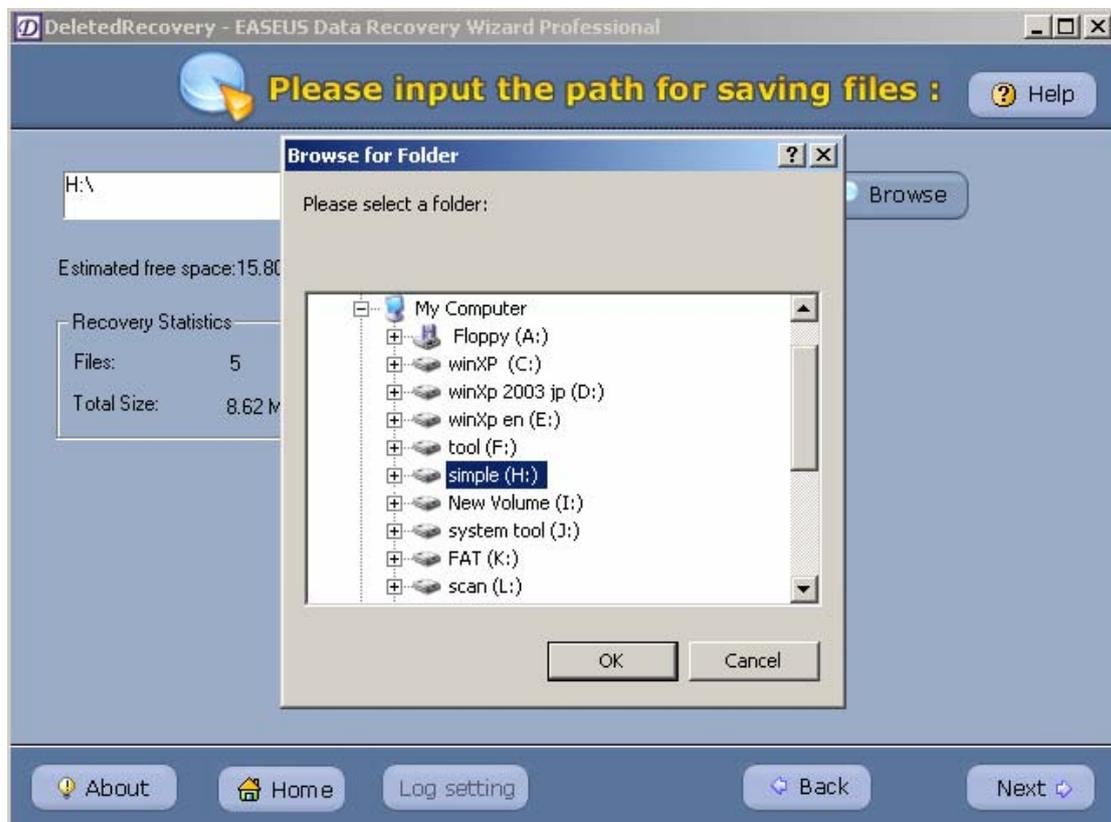
Click “Next” and scan the deleted files, choose the deleted folder” my file”

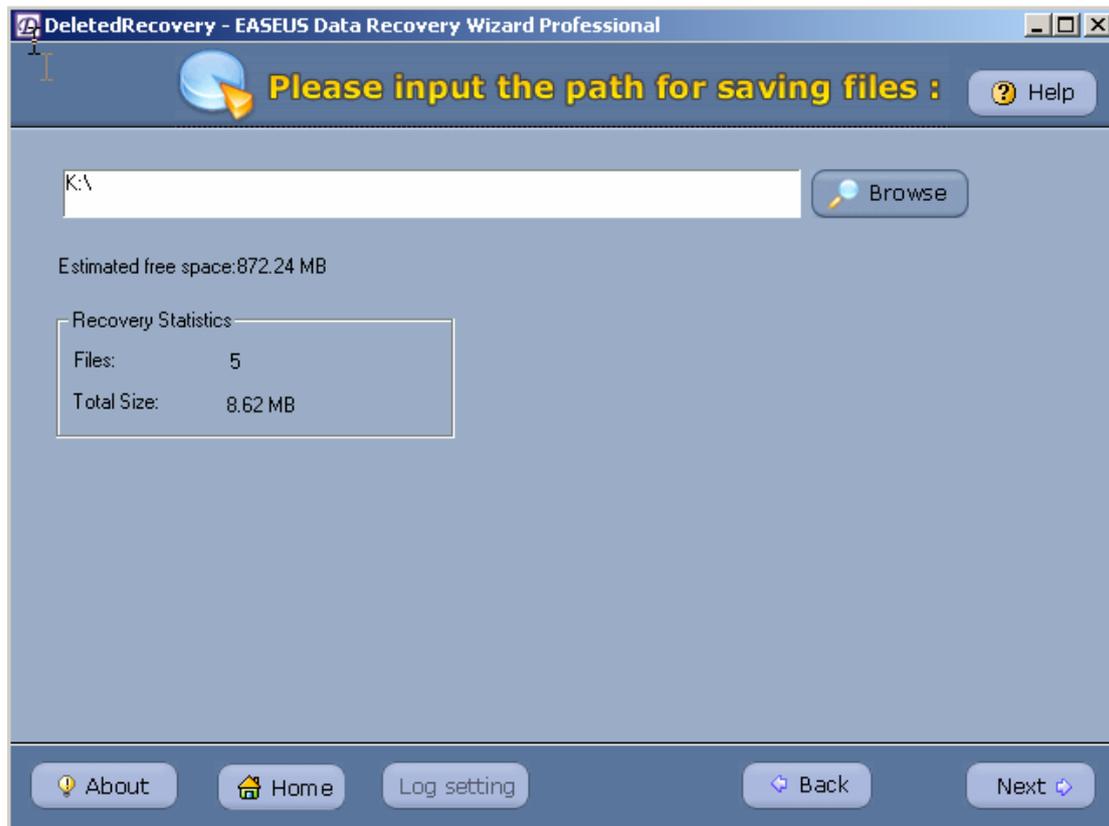


Tick it and click “Next” to enter the interface of choosing path for saving files.



Choose a partition with large space to save the files

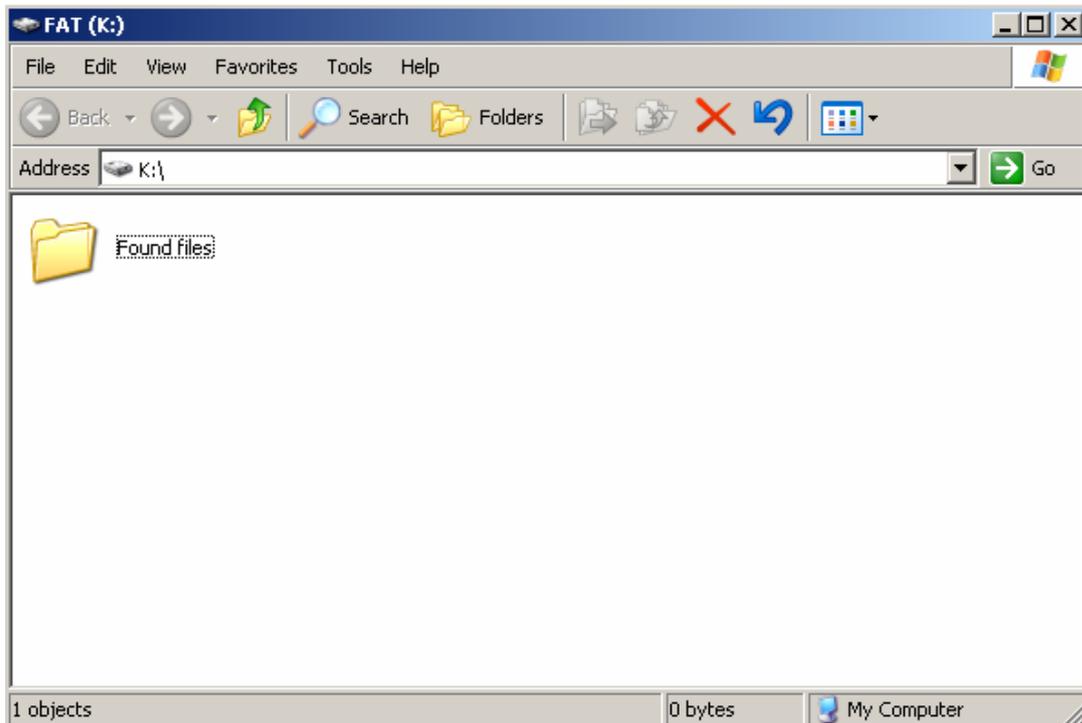




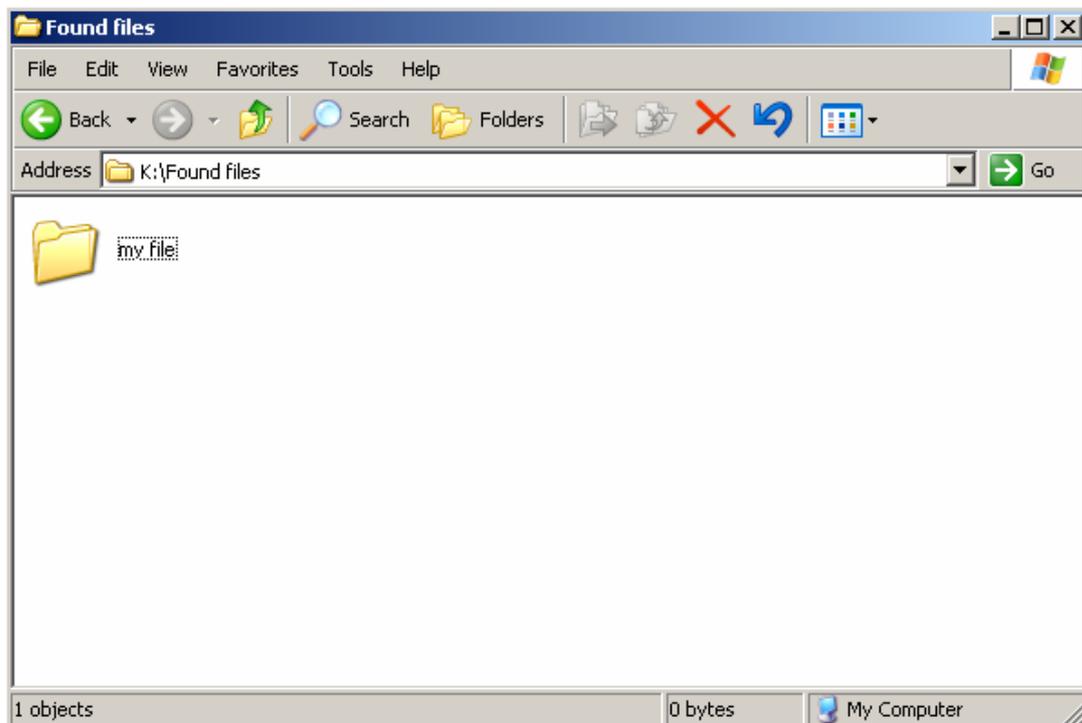
Click "Next", after recovery, there will be a Report interface.

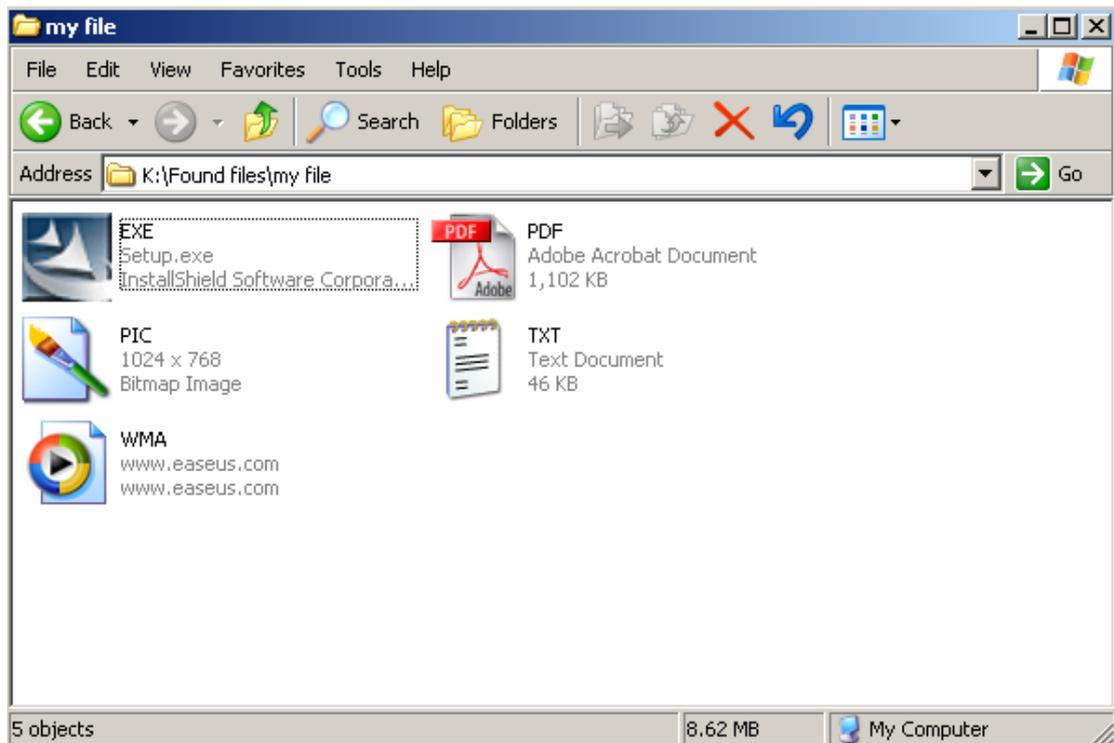


If you want to check the recovered files, please click “file saved to”.



Enter the folder “found files” created by Data Recovery Wizard 3.0 and you can see the recovered folder/file:



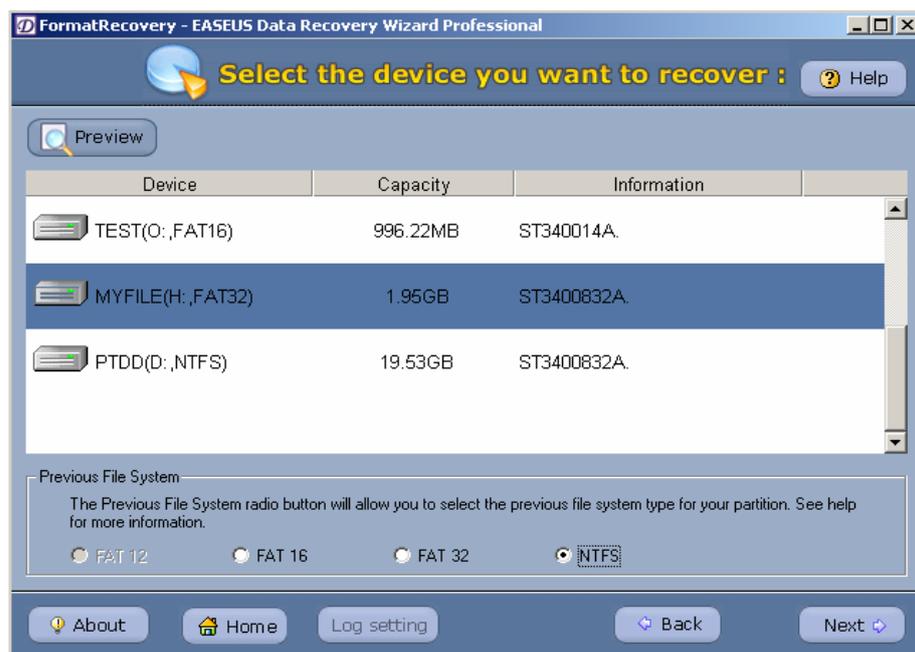


If you are not satisfied with the result, you can use “AdvancedRecovery” module, and repeat steps above.

4.FormatRecovery

Eg: Format a partition from NTFS to FAT32

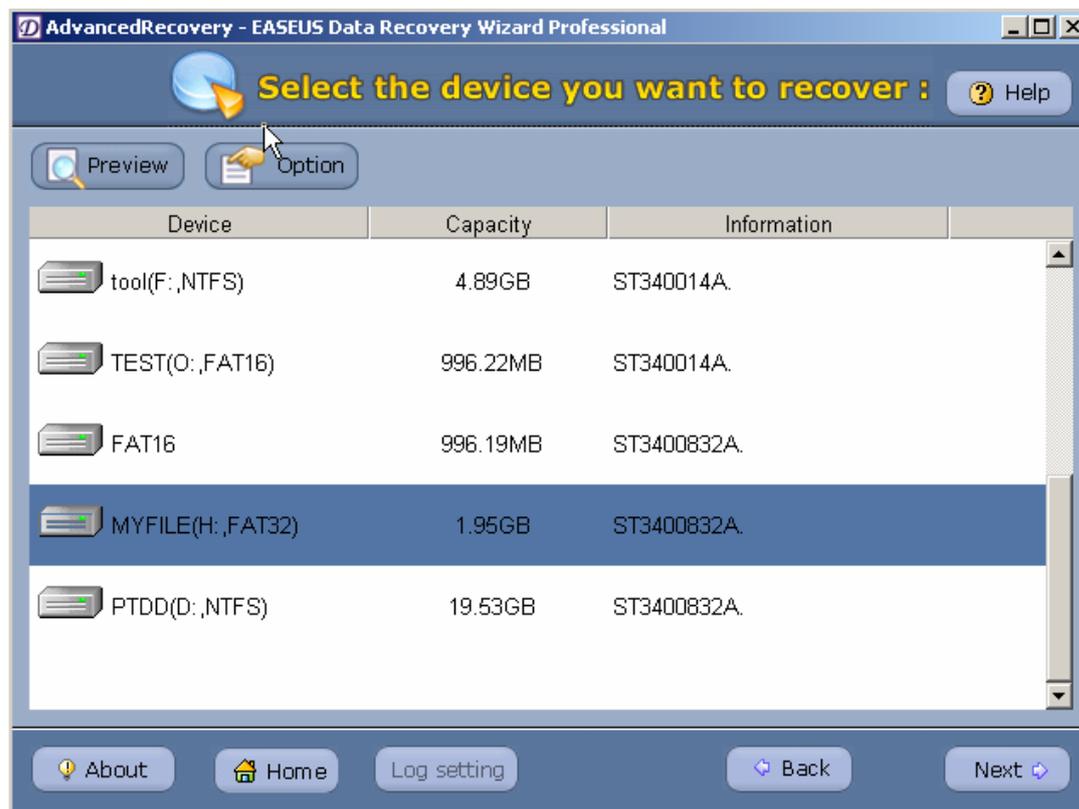
Run and choose “FormatRecovery”, and then choose the formatted partition and tick the previous file system(here is “NTFS”)in “Previous File System”



Click “Next” to search files, the following steps are the same as previous methods.

Attention: if you are not satisfied with the result or you can not remember the previous file system on the partition or the program can not figure out the size of clusters on the partition, you can refer to the following steps:

Choose “AdvancedRecovery” and choose the partition



Click “Next”, the following steps please refer to previous methods.

5.Recover encrypt/compressed files in NTFS

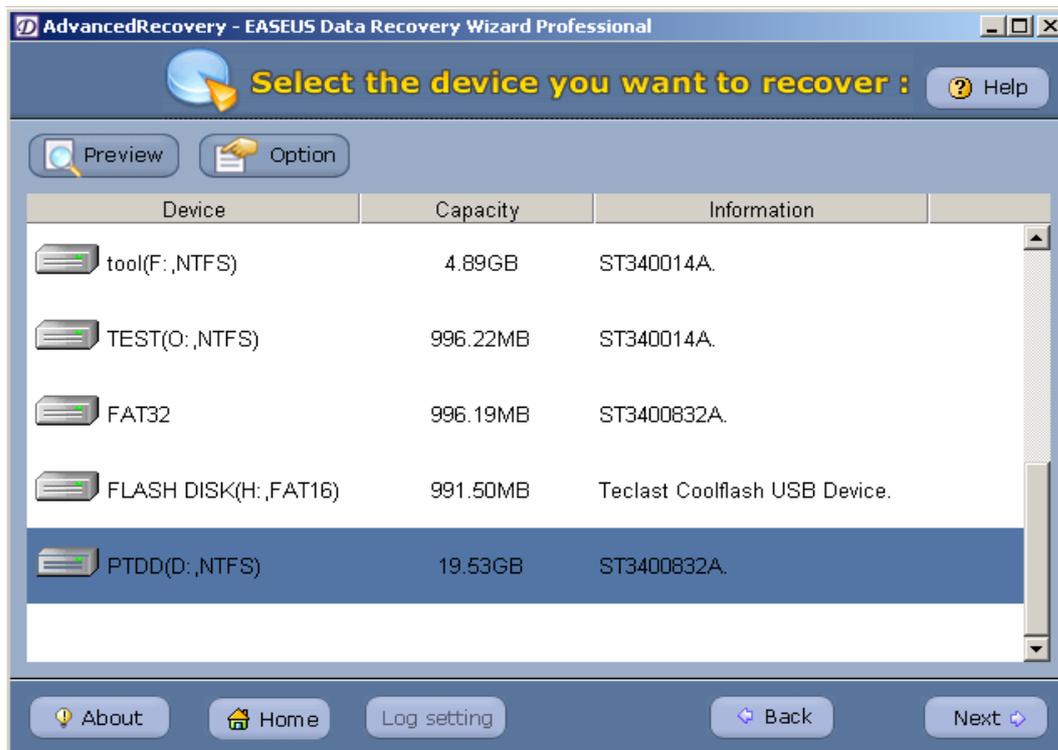
Attention: if you want to recover encrypt/compressed file in NTFS, you need Data Recovery Wizard Professional 3.0, for Data Recovery Wizard 3.0 does not support encrypt/compressed file recovery.

Encrypt/compressed file recovery and deletedrecovery are mostly the same. But more attention should be paid to that to rightly recover encrypt/compressed files , you need use the account that create encrypt/compressed files to log on Windows; moreover, the encrypt files must be recovered and saved to partition of other NTFS type not FAT partition, or the recovered encrypt files can not be opened correctly.

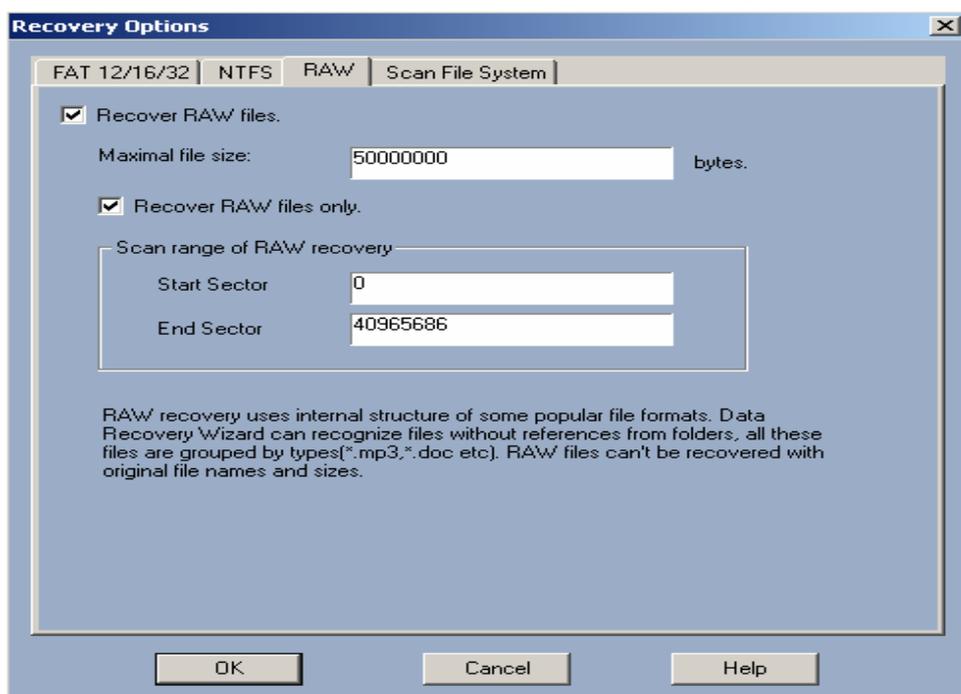
6.RAW RECOVERY

When the partition file system is completely destroyed or “AdvancedRecovery” cannot find data you want, you can try “RAWRECOVERY”.

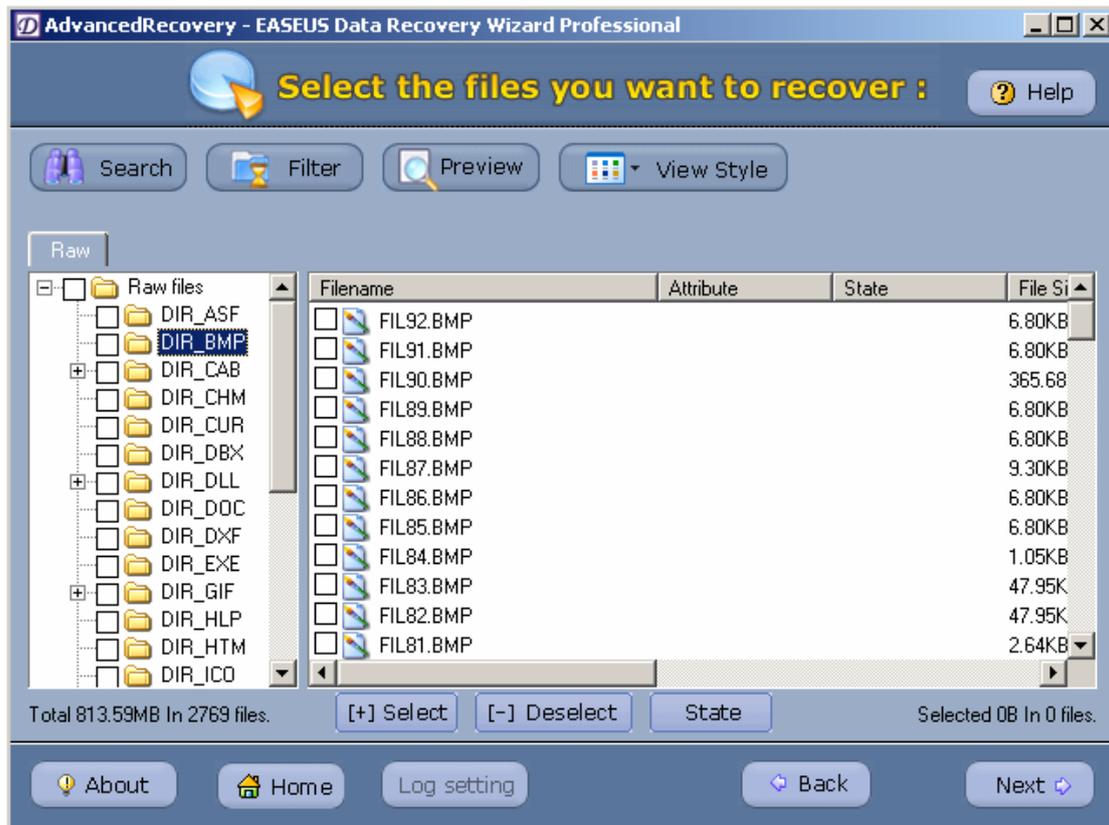
Run the program and choose the partition you want to recover



Click “Option”, choose “RAW” and tick “Recover RAW files only.”:



Click “OK” and “Next” to finish the searching operation:

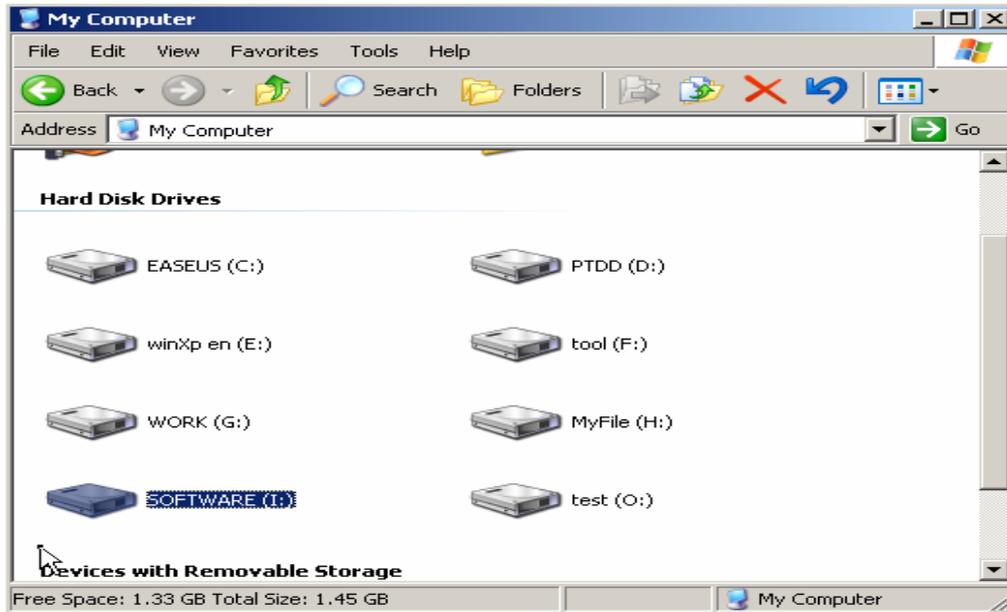


Attention: this recovery module does not scan files via file system, so the files found have no usual filename and path. The program will classify the files according to the file types.

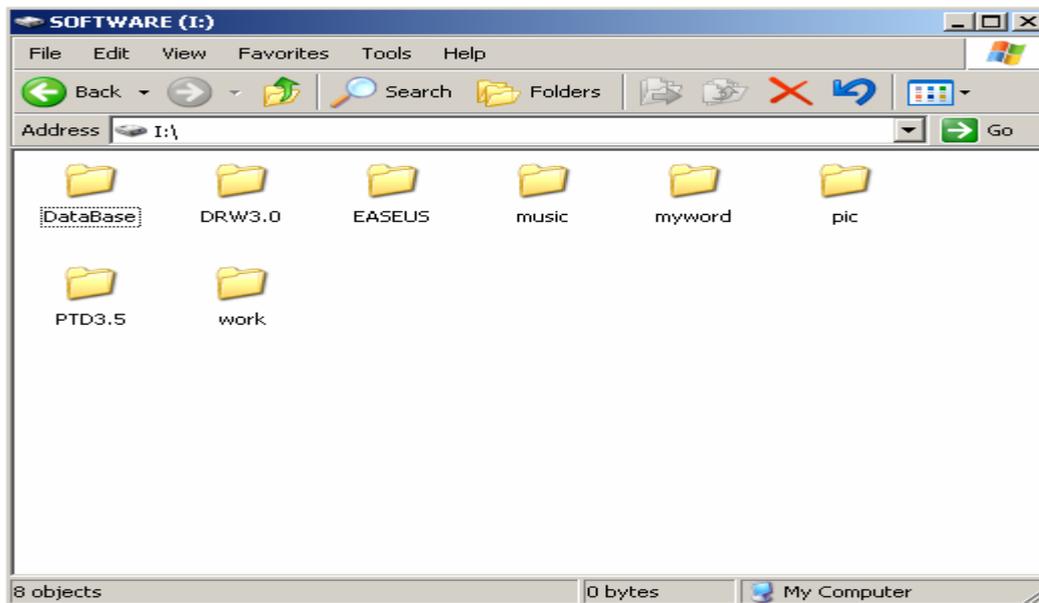
Click “Next”, the following steps are the same as the previous methods.

7. Recovery when parts of partitions are lost:

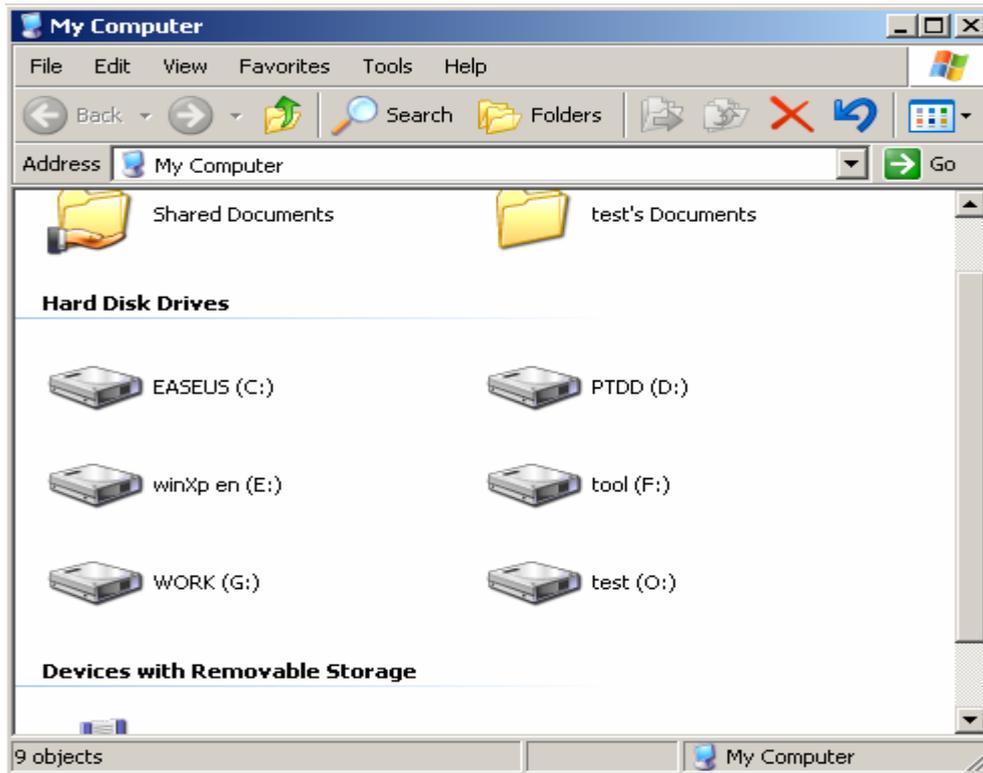
Here are the partitions before lost:



Files in the partition:



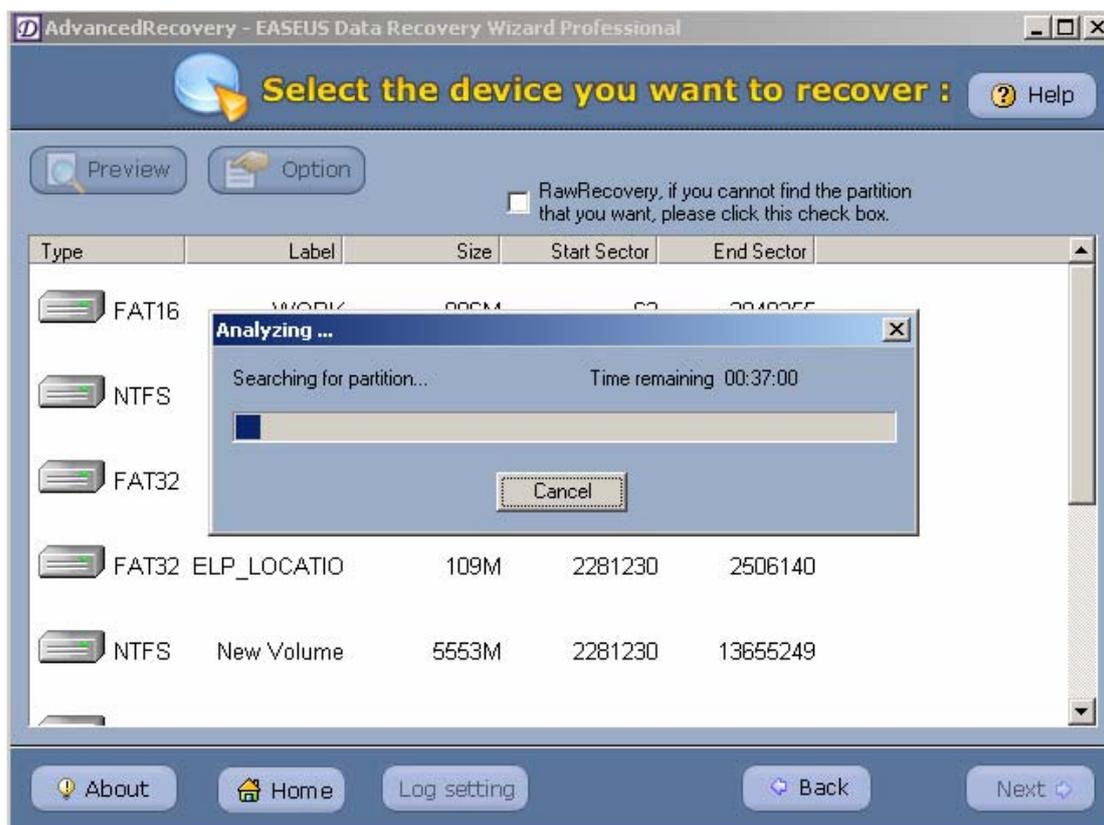
After deletion, we cannot see the partition in Windows explore:



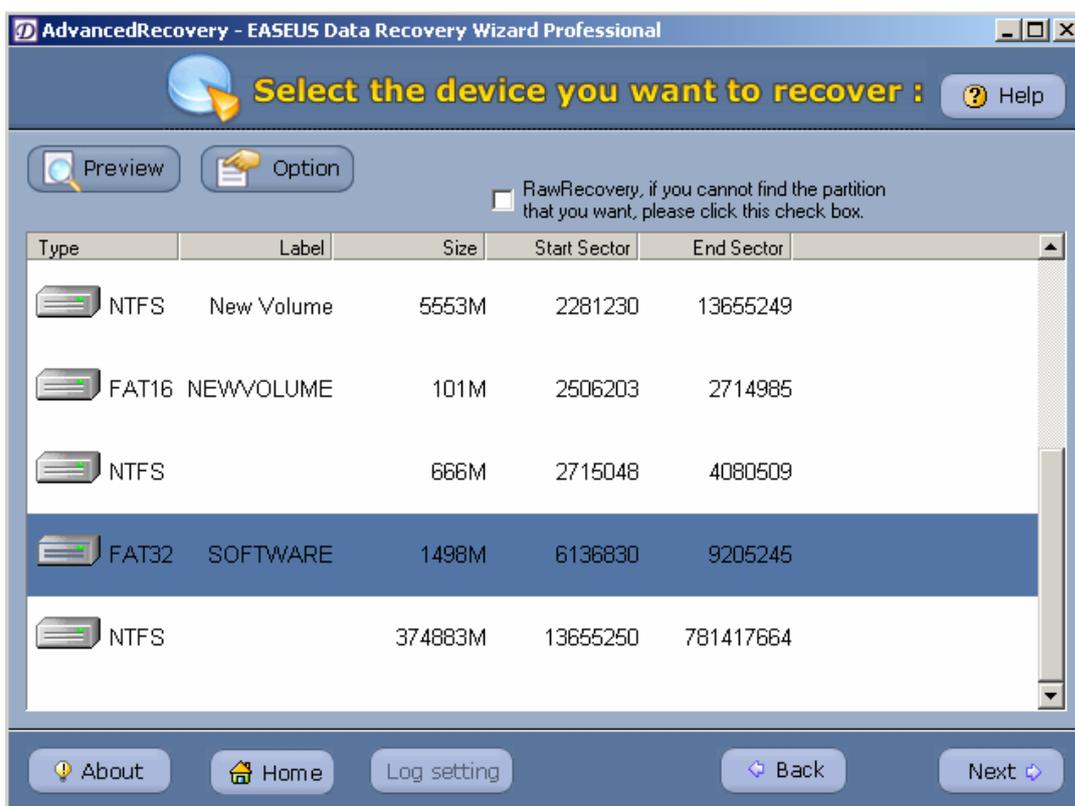
Run Data Recovery Wizard, enter “AdvancedRecovery”, and choose the disk where you lost your partition.



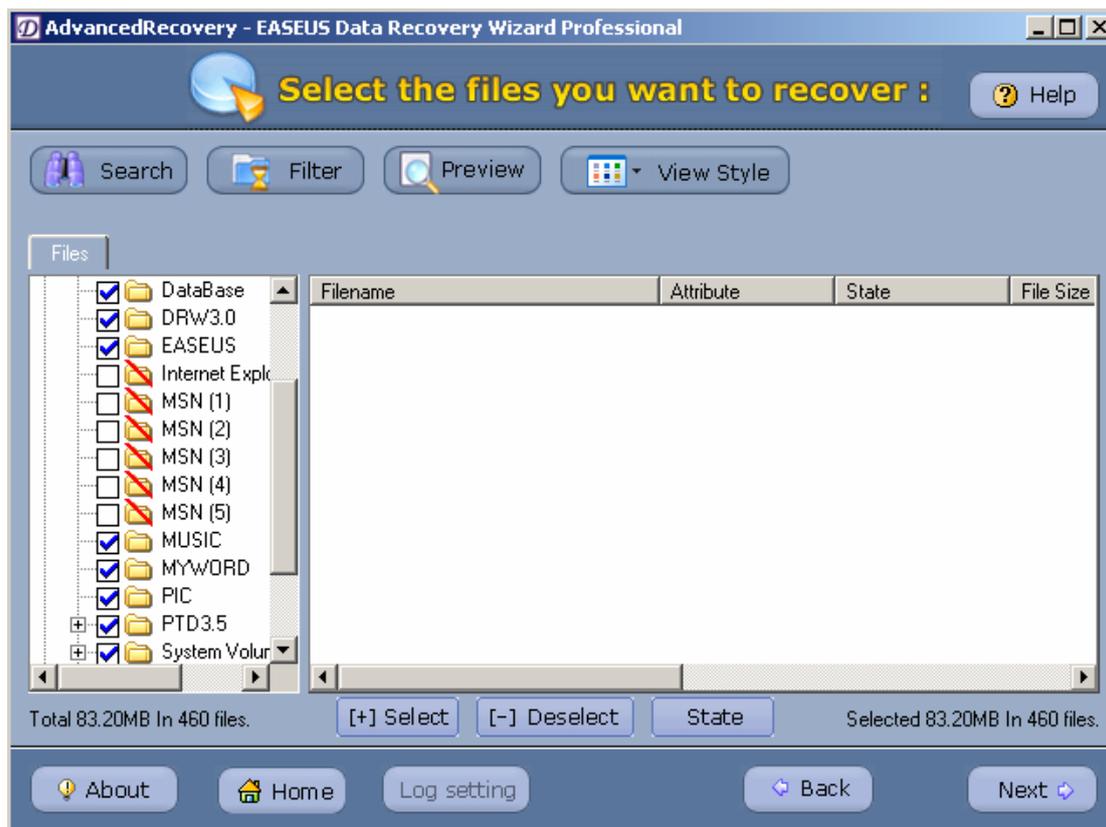
Click “Next”, search the lost partition.



After searching, find the lost partition on the list, choose it:



Click “Next”, the result of search is as following:



The rest steps are the same as the previous methods.

8. Data recovery in dynamic volume

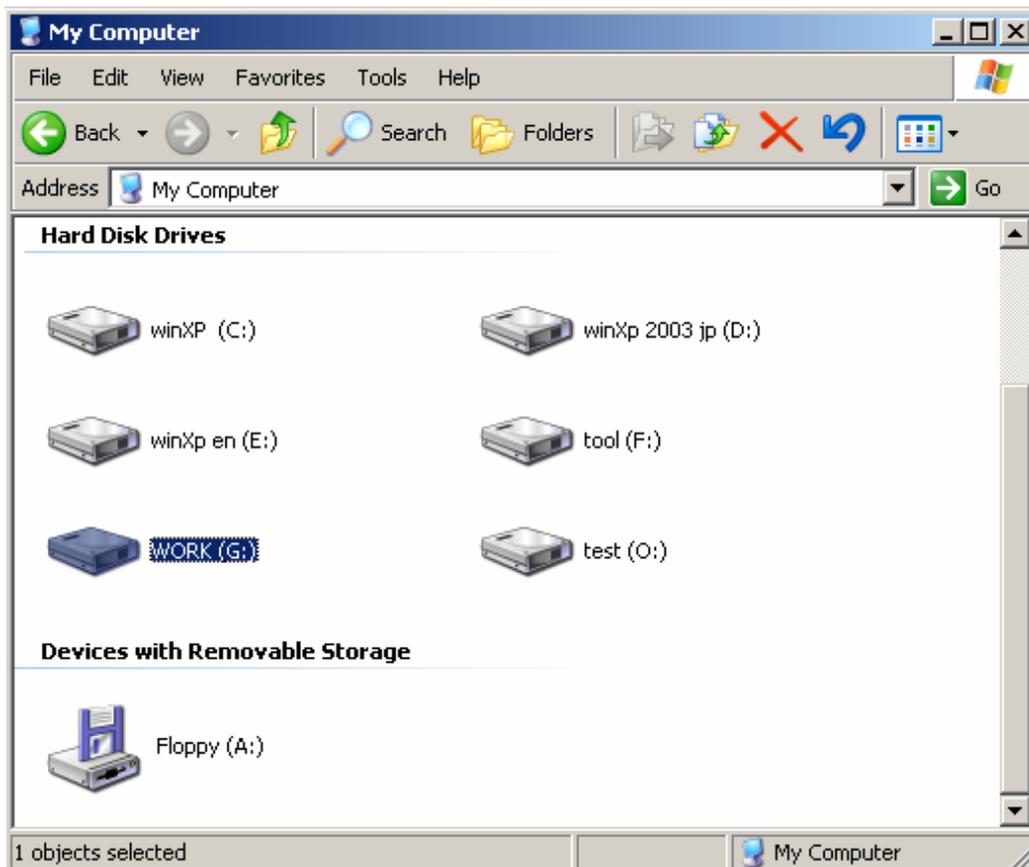
If you want to recover the data lost in a dynamic volume, you need Data Recovery Wizard Professional. Data Recovery Wizard does not support dynamic volume recovery. Data Recovery Wizard professional supports simple volume, spanned volume, striped volume, mirrored volume and RAID5.

The method is the same as that of other types of partitions.

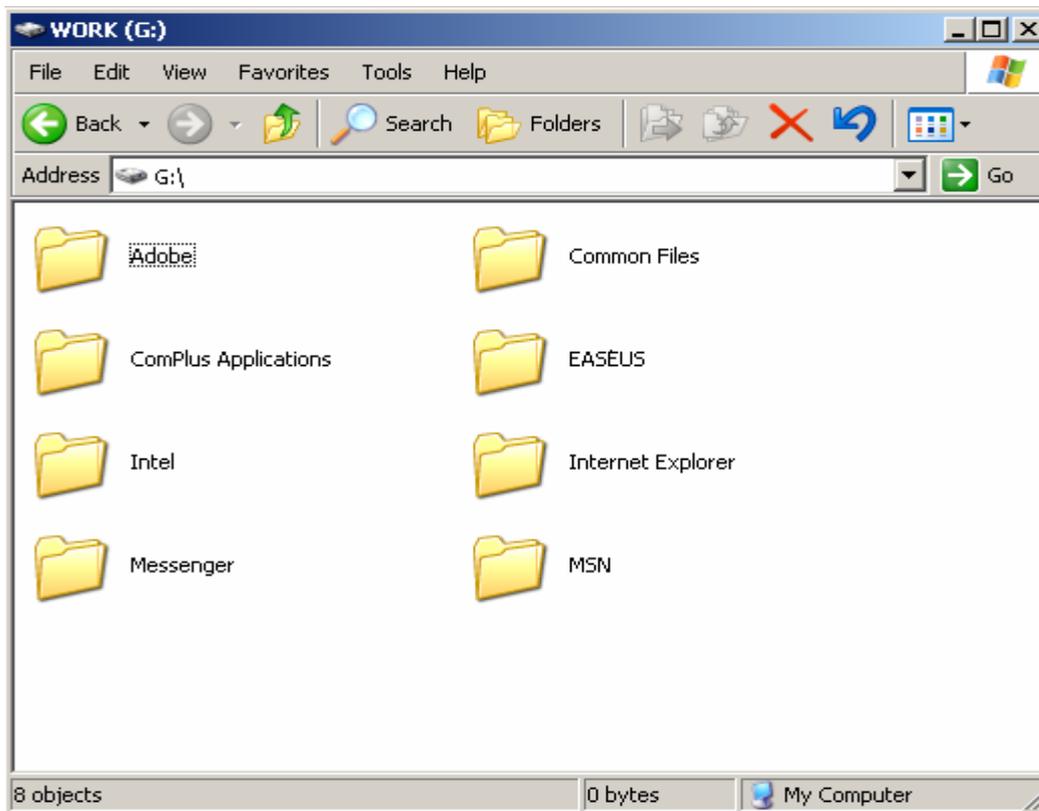
Attention: if you have lost the dynamic volume, Data Recovery Wizard professional can not recover your data except that on simple volume.

9. Data recovery on inaccessible partition.

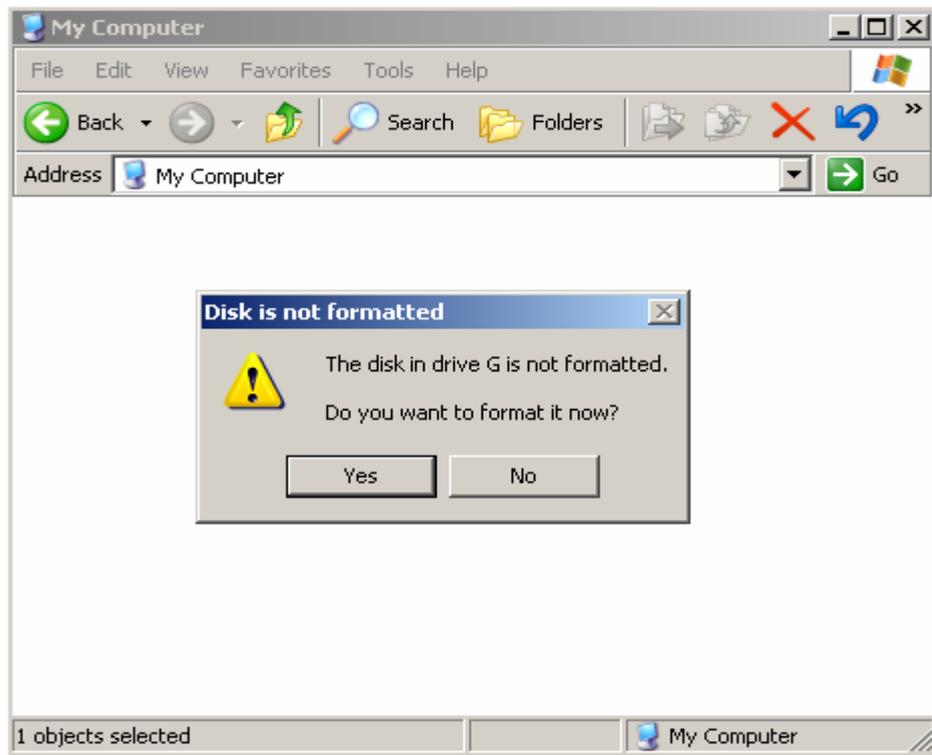
Before the partition was destroyed:



The files in partition are:

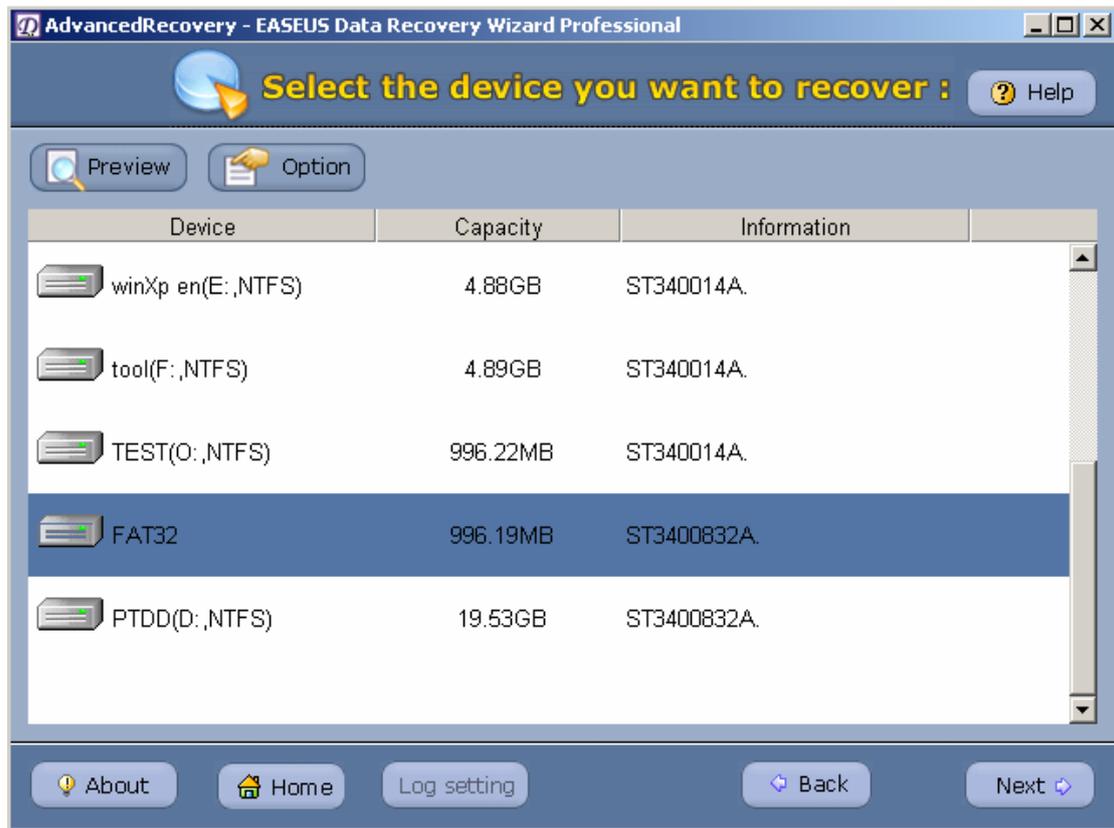


After the partition is destroyed, when enter the partition, it will prompts “The partition is not formatted.”



Run the program, choose AdvancedRecovery, and choose the partition.

Attention: the volume lable might have been destroyed, which may cause that the partition label cannot be shown on the partition list. In this case, you can choose partition according to the type and size of the partition.

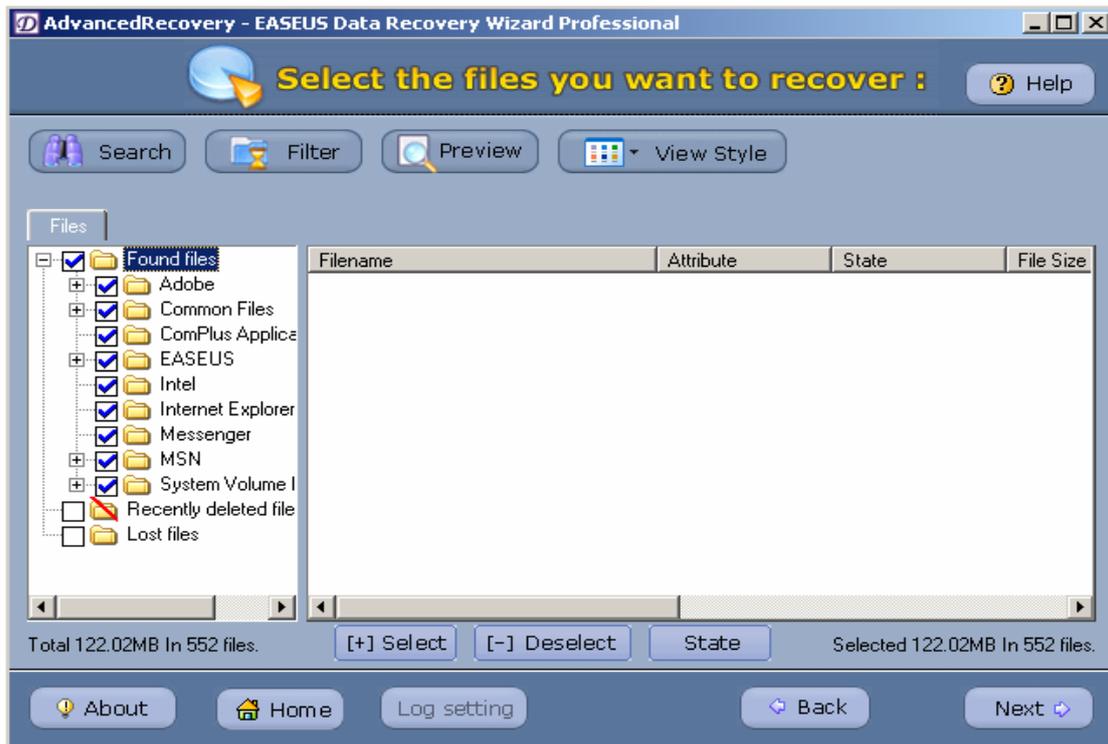


Click “Next”, the following steps are as the previous methods.

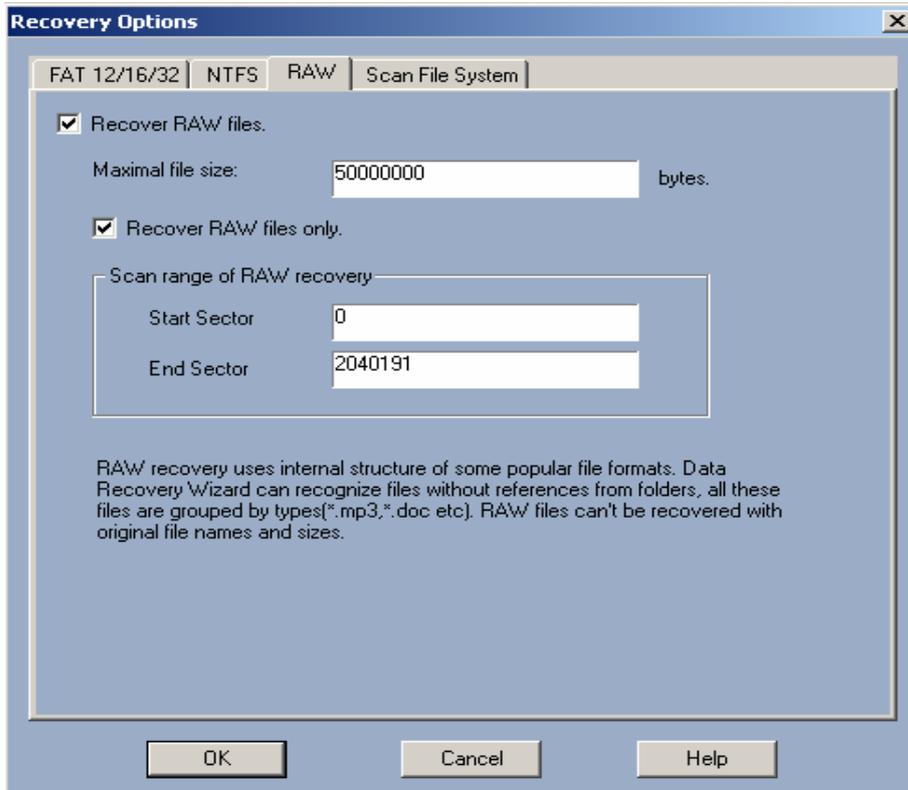
If the program cannot recover the files, there will be a prompt message:



Choose the partition, click “Option”, and choose “Scan File System”:



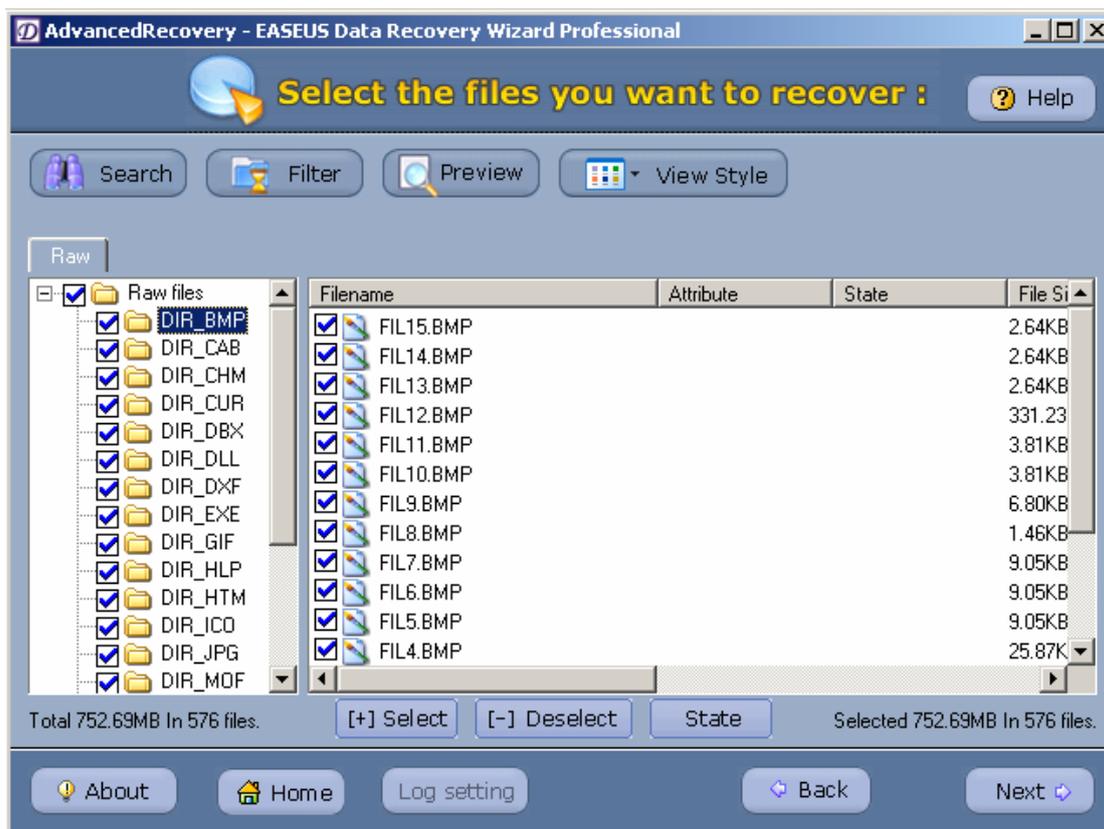
If you are not satisfied with the result, please Click “Option”, choose “RAW”, and tick “Recover Raw files only”



Keep the default, click “OK” and then click “Next”.

Attention: “RAW Function” does not scan files via file system, so the files found have no usual

filename and path. The system will classify the files according to their file types.



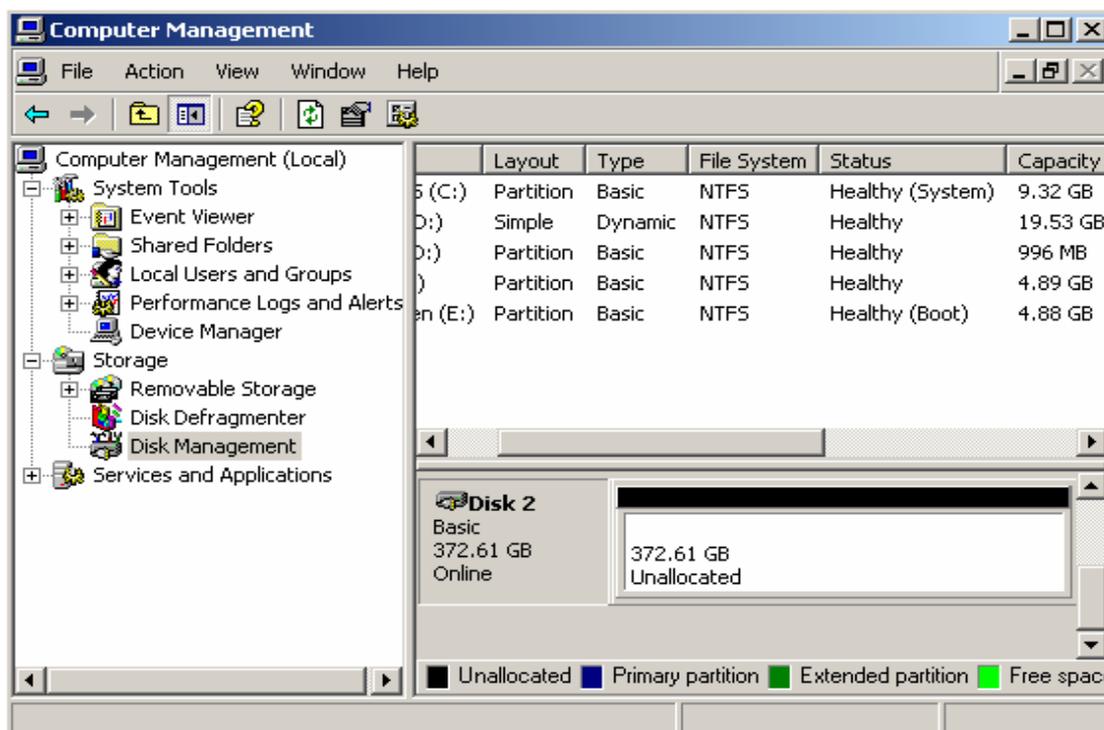
Click “Next”, the rest recovery steps refers to the previous methods.

10. File recovery on RAW partition

In this case, the methods are the same as that of “Data recovery on inaccessible partition.”

11. Data recovery when all the partitions are lost

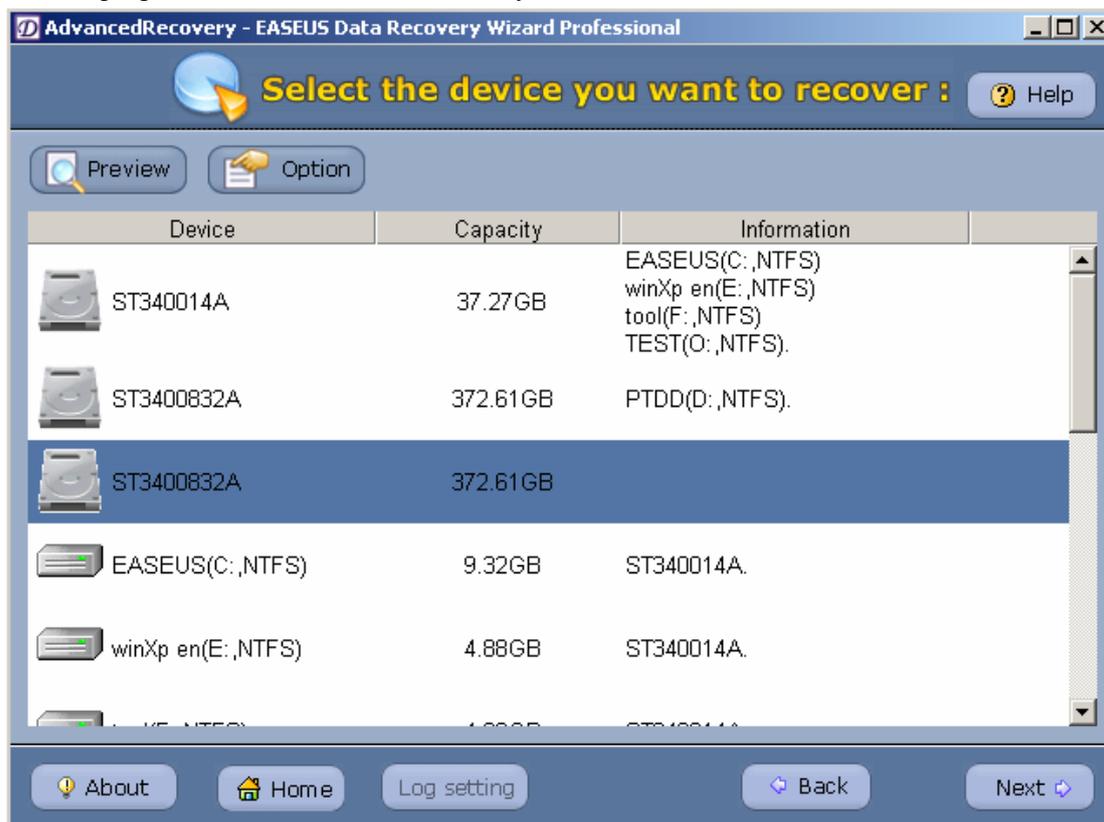
When all the partitions are destroyed, you can not see any partition in disk management:



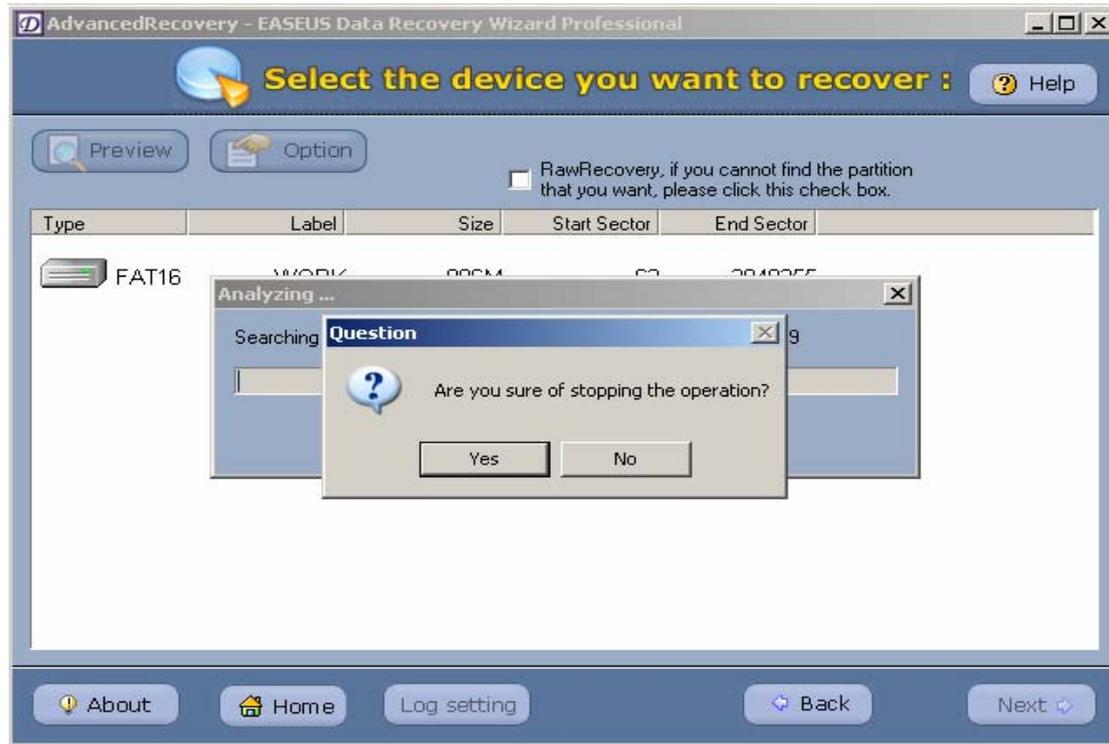
In this case, you can refer to the method of “Data recovery when parts of partitions are lost”.

If you cannot find the partition where you want to recover data when searching the partition, you can follow these steps:

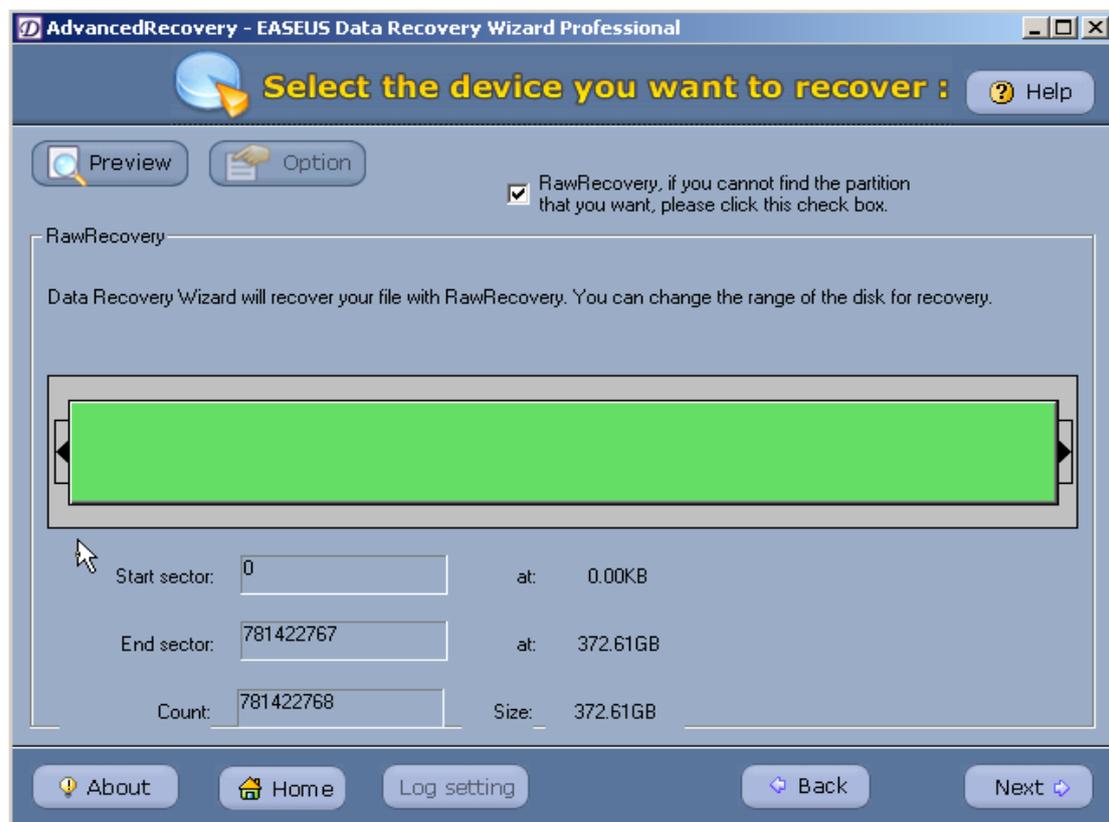
Run the program, choose “AdvancedRecovery”, then choose the HD:



Click "Next", click "Cancel" when searching the partition, then exit.

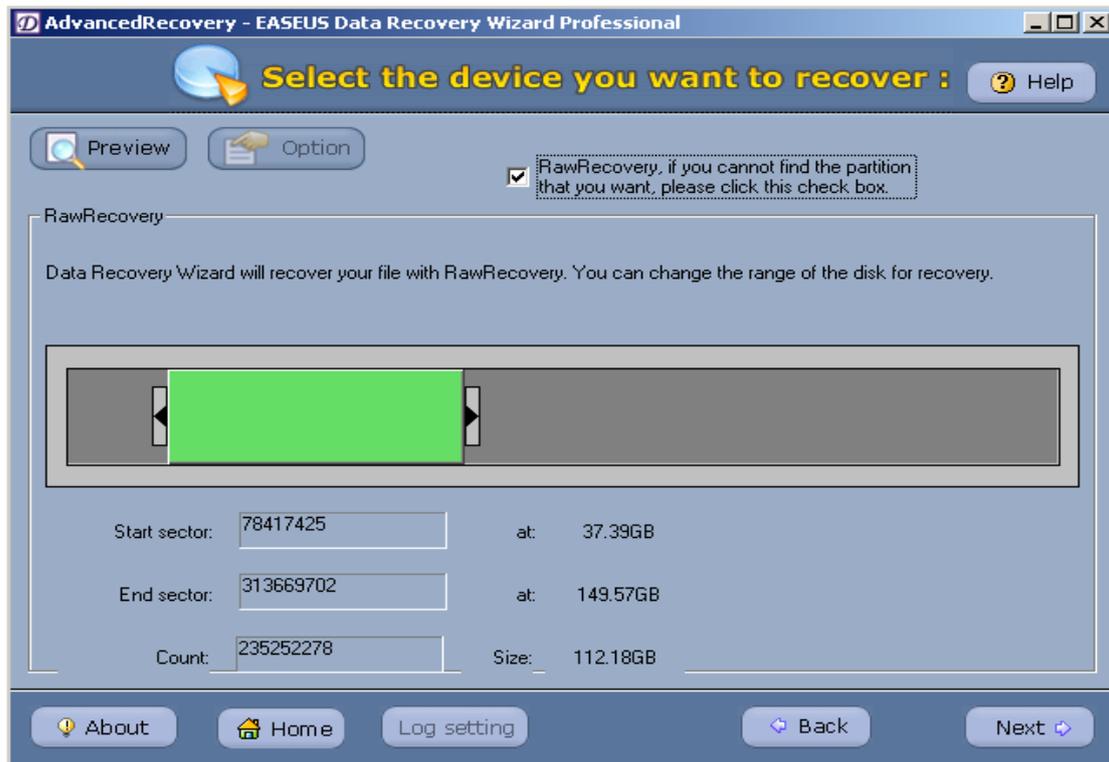


Tick "RawRecovery, if you cannot find the partition that you want, please click this check box".

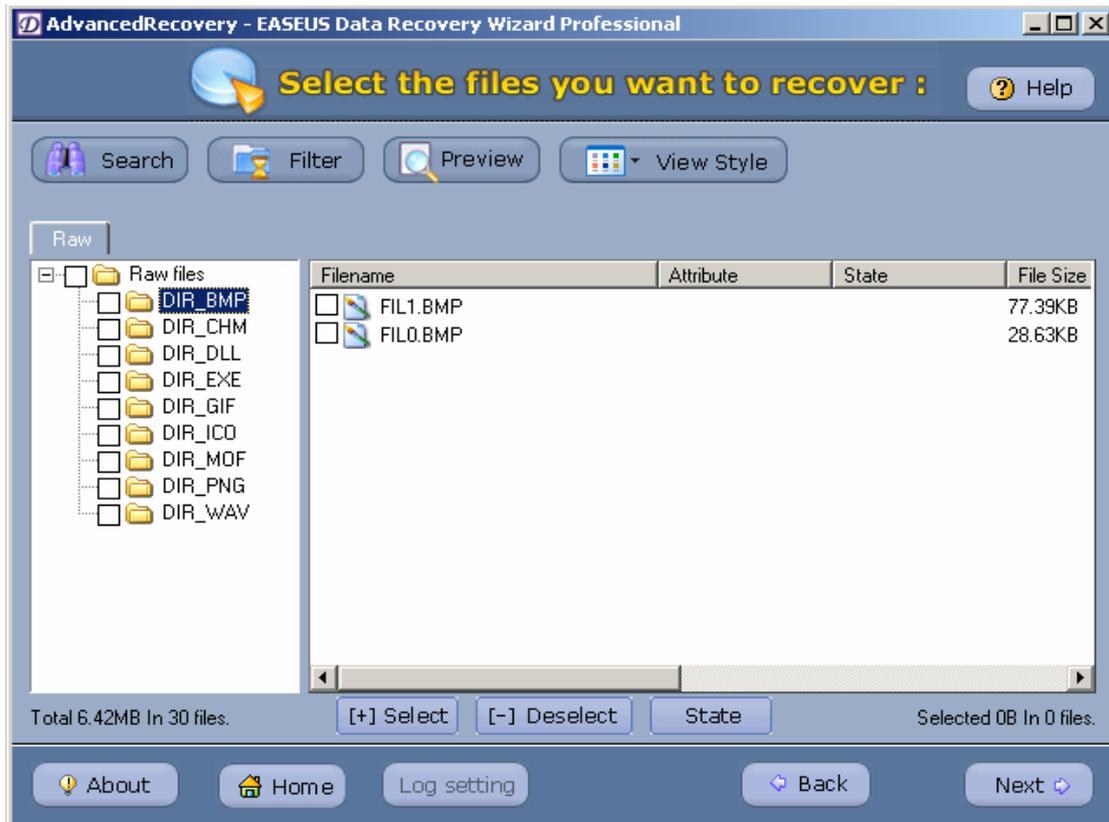


Roll the block to adjust the range of the sector you want to recover (you can set the range

according to the place and size of the partition):



Click "Next" to search files in specified sectors



The rest steps are the same as the previous methods.

12. Data recovery when GHOST Image restore failed.

In this case, there can be different recovery scenarios according to specific damage of the partition and file system:

Usually, after the failure of GHOST Image restore, partition table of the target disk would be in some damaged condition, you can search the partition where you want to recover data by “Searching for Partition” function in “AdvancedRecovery”.

If the partition is found, please refer to “Data recovery when parts of partitions are lost”.

If not, please refer to “data recovery when all the partitions are lost”.

13. After Partition Magic size revision/ combination/ division of partitions fails, how to recover the lost data?

In this occasion please refer to “Data recovery when GHOST Image restore failed”

14. When using Data Recovery Wizard 3.0 to recover files, there is some strange sound in HD. How to handle it?

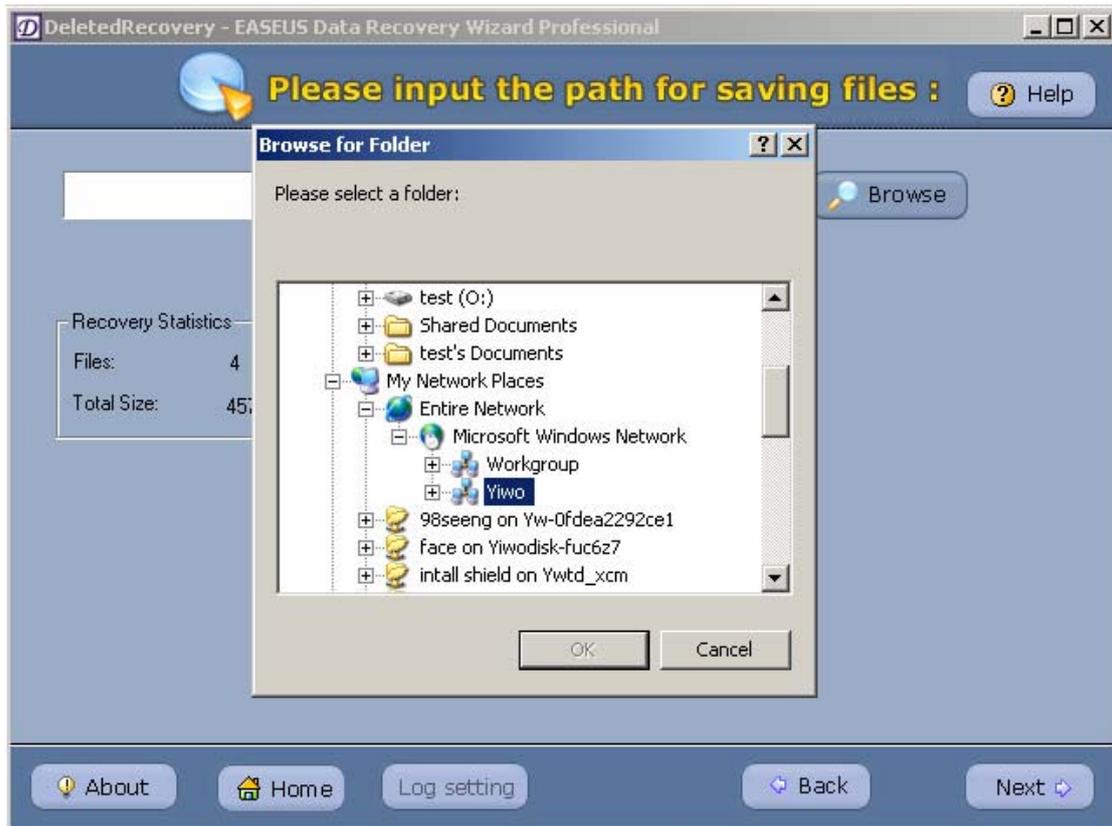
Your HD has some hardware problems. In this occasion, you need stop running Data Recovery Wizard 3.0 at once, and then send your HD to HD maintenance station.

15. HD cannot be detected in BIOS, how to recover data by Data Recovery Wizard 3.0

The precondition of data recovery by Data Recovery Wizard 3.0 is that the storage device has no hardware problem and runs normally; or Data Recovery Wizard 3.0 can not help you.

16. There is not enough space in hard disk to save the recovered files, nor there is removable storage device, how to handle it?

You can save you files to other host computers via network, please refer to steps as following:
Choose another host computer on network:



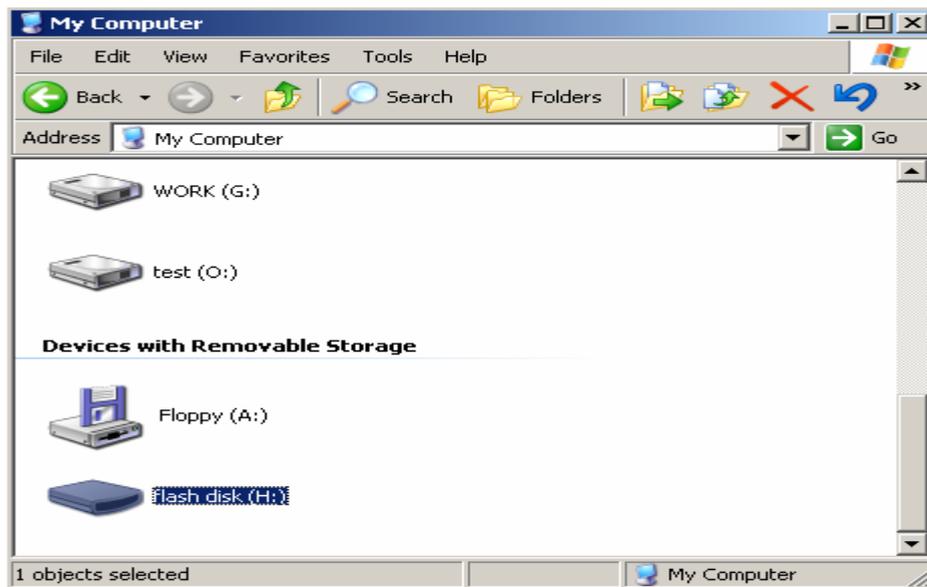
Choose the host computer where you want to save your files, the rest steps please refer to the previous methods.

17. How to recover data in other storage devices (eg: floppy disk, flash drive, removable disk etc)?

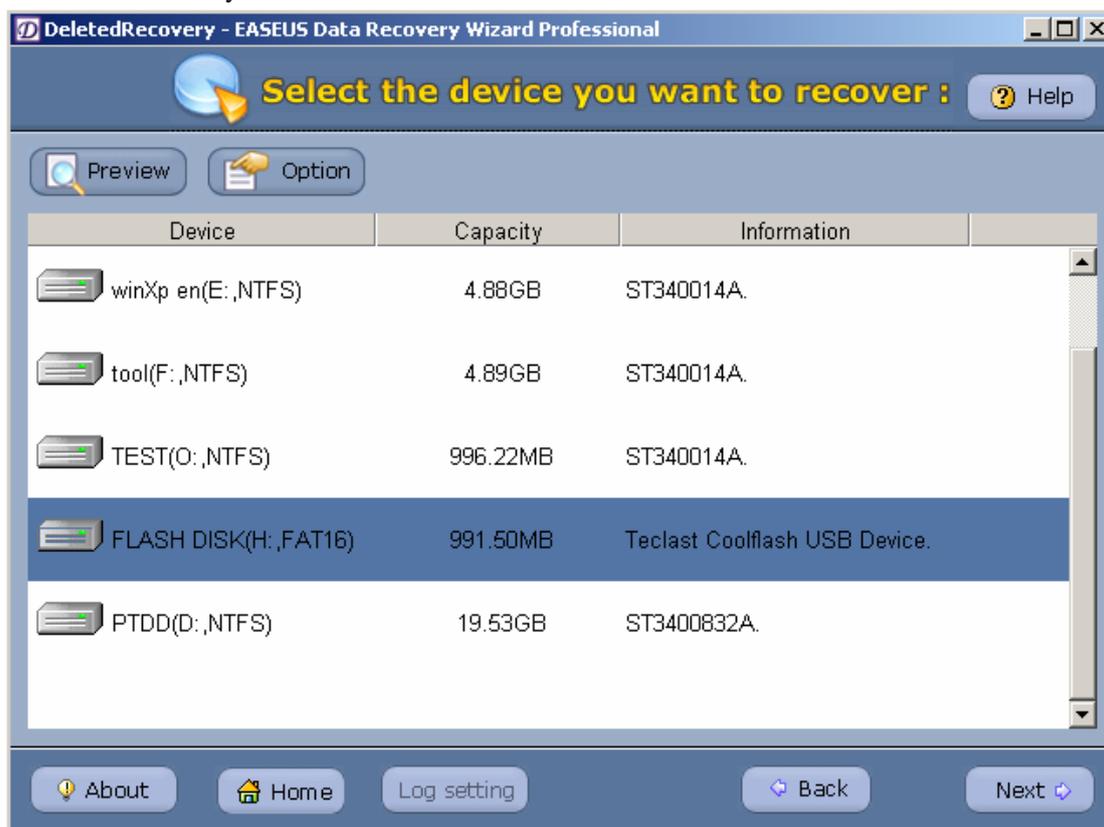
Attention: Date Recovery Wizard 3.0 supports storage devices both with MBR and without.

Eg: To recover data in flash drive

Connect flash drive with host computer:



Choose different ways according to specific damage condition. Here we choose “AdvancedRecovery”:

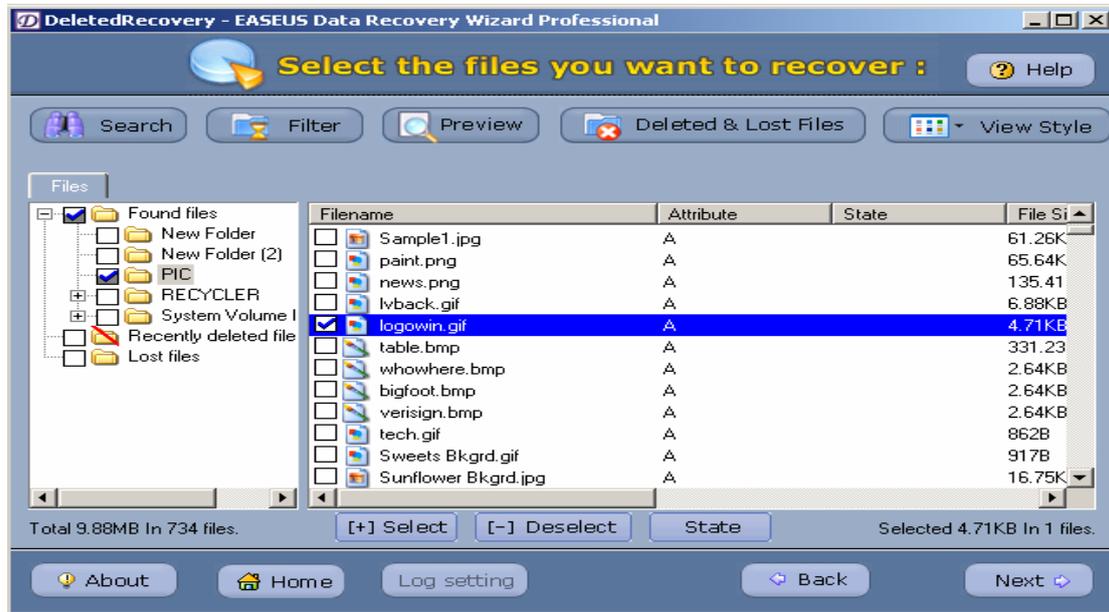


Click “Next”, the rest steps please refer to the previous method.

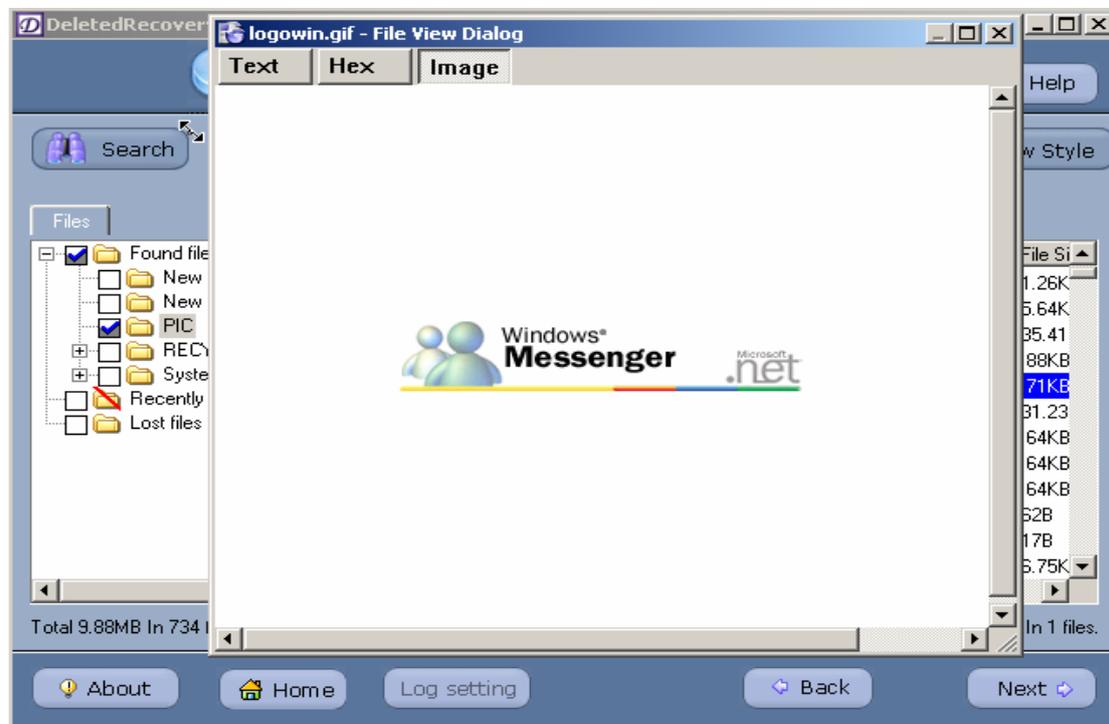
18. To recover image file, how can I know it can really recover the data before I buy Data Recovery Wizard 3.0 ?

You can use “Preview” function to preview;

Run demo version to search the files, then choose an image file, click “Preview”



Preview result:



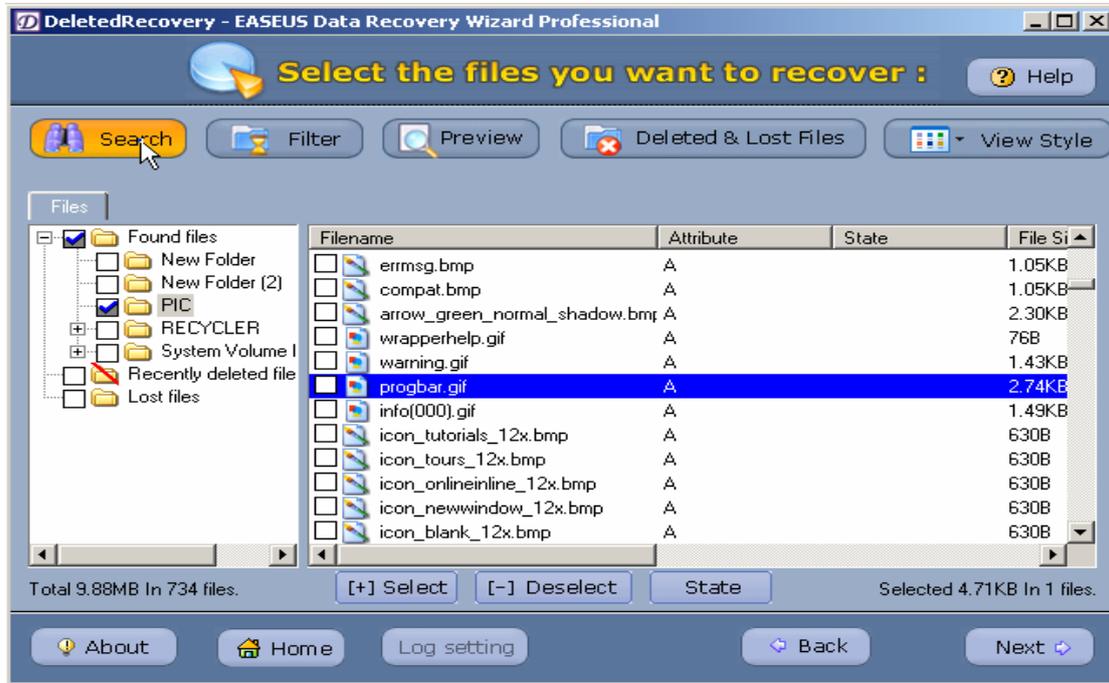
If the programmer can preview the image, Data Recovery Wizard 3.0 can rightly recover it.

19. There are so many files recovered, how can I find the files I want fast?

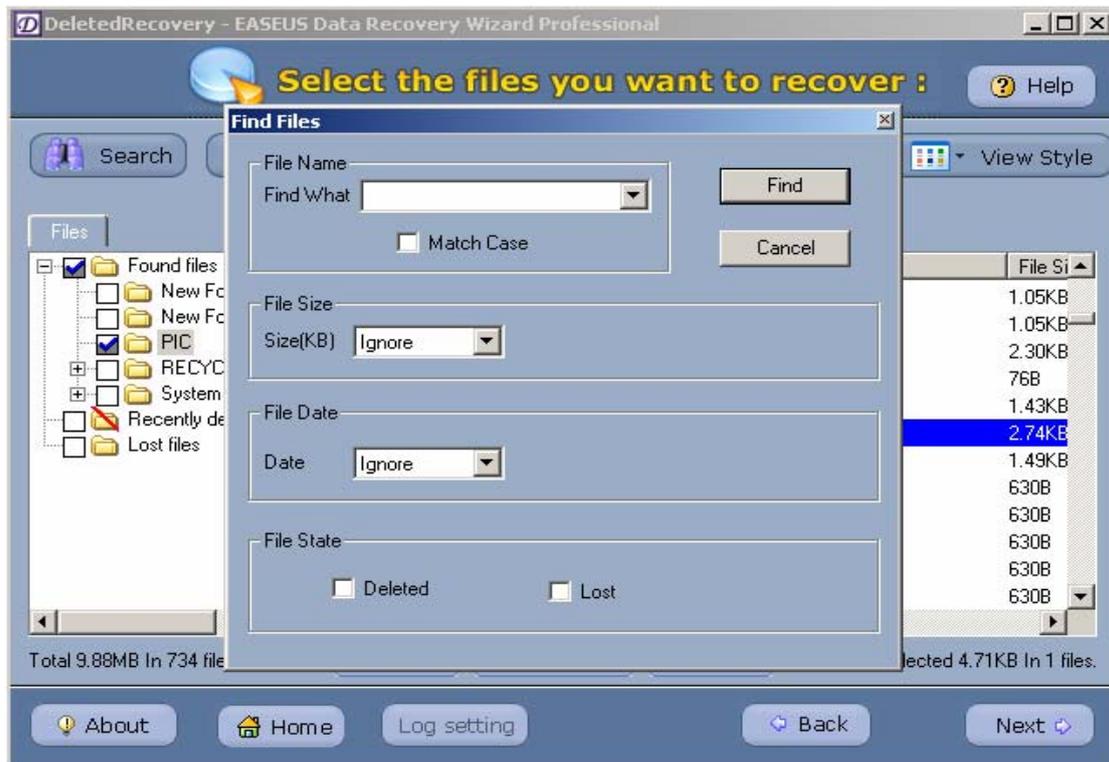
With “Search” and “Filter”, you can find files you want.

Eg for “Search”,

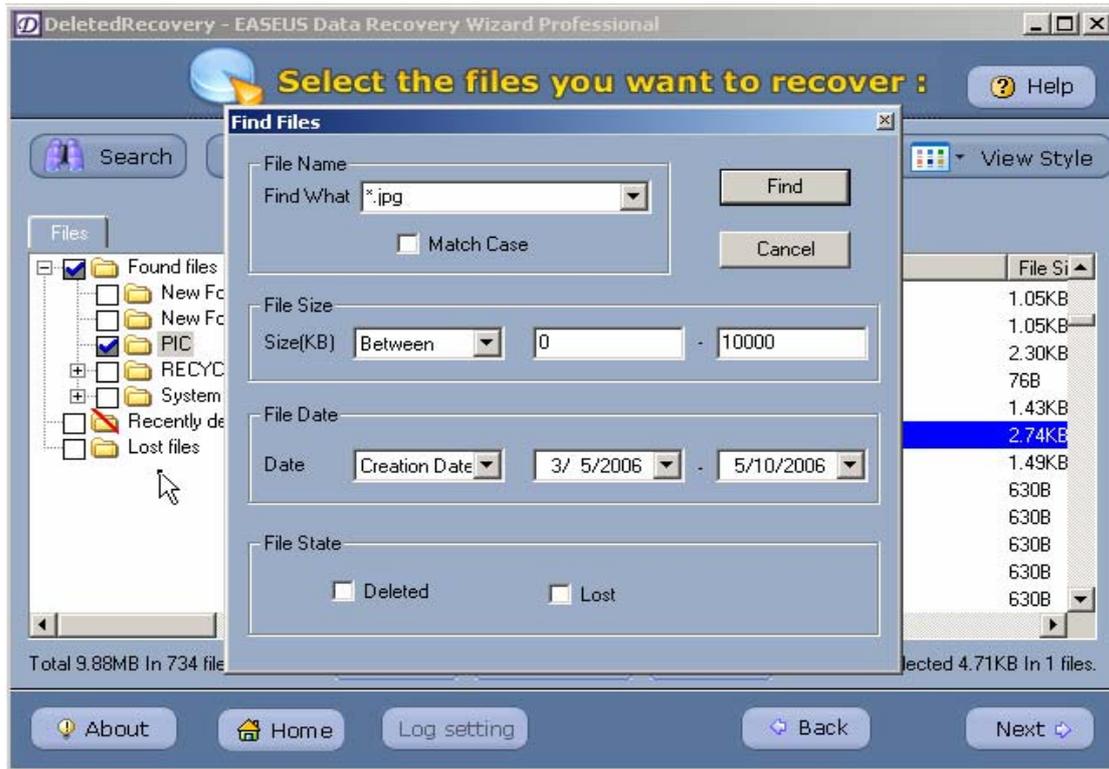
After searching:



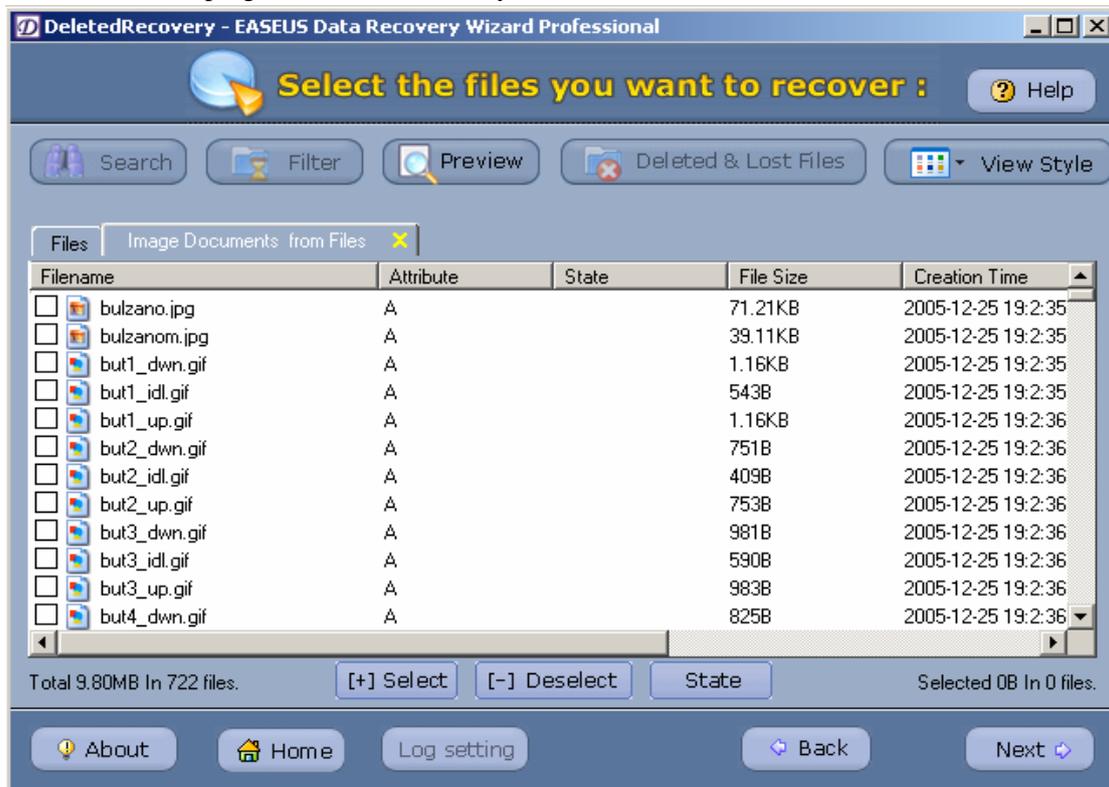
Click “Search”:



Enter the query information of your file:



Click "Find", the program will list the files you want:



Click "Next", the rest steps please refer to the previous methods.

20. I have recovered some files, but I cannot rightly open them.

In some cases, files recovered by using Data Recovery Wizard cannot be opened, which means the data has been badly destroyed.

You can try the following steps:

1. Send the badly destroyed files to our email (repair@easeus.com); we will try our best to recover for you.

2. Try to fix them with some file recovery tools

Attentions: Some documents that are badly damaged are irrecoverable.

21. In what occasion I cannot rightly recover data?

In occasions as following, you cannot rightly recover data:

1. Operations that cause the data are covered, such as: failure of GHOST image restore, virus attack, and mass write operation to the disk where you want to recover data etc.
2. There are some physical problems in storage devices.

The Data Protection

XII.Introduction of data security software

Because the file might be recovered easily, it is quite possible to expose your sensitive information. Data Security Wizard can guarantee your data irrecoverable forever. Security of Data Security Wizard security lies in:

1. It can erase information thoroughly by US DOD arithmetic. It guarantees irrecoverability.
2. It can erase information exactly. It can erase the system information of a file, and leave no trace.
3. Data Security Wizard can erase a whole hard disk.

Download Data Security Wizard from: <<http://www.easeus.com/download.htm>>