Data Recoverability

In a data loss scenario the most important question is: Are the files still recoverable? This answer depends on what action needs to be taken, whether to pursue the data recovery or to develop strategies of coping with the data loss.

The situation is often very difficult to judge. Sometimes it is not fully clear what caused the data loss in the first place. Some technician might have already tried to solve the problem. Also, the effect of common remedies, such as Microsoft's "Checkdisk", on the recoverability is quite unknown.

This article is supposed to sort out what is possible and what is not. We will try to explain, for a given scenario, what can be expected. For this purpose we must restrict the idea of "recoverability" to "commercially available and affordable data recovery". While it might be possible that the magnetization that once constituted the data is still present on the media, often times the technology to recover the data for an economically reasonable price is not available.

The Type of Data Influences the Success Rate

We must also consider the kind of data that needs to be recovered. Just assume you are able to recover a hypothetical 90% of all lost files. If these files were pictures you can consider this rate a success, you got 9 out of 10 pictures back. If your files were tables of a database and 10% are missing, the entire database will probably be worthless because the data depends on each other. The more the data is depending on each other, the greater the devastation will be if even a small percentage of data is missing. We will also will look at what "90% recovered" really means. Another interesting aspect will be the "time dimension": a data recovery is usually worth less with each day, sometimes each hour, that passed by.

Physical and Logical Data Recovery

We need to distinguish between two different procedures:

1. Extracting the raw data from the affected media (physical data recovery)
2. Reconstructing the files (logical data recovery)

You can have pure logical data losses. For example, file deletion, drive formatting or virus attacks only require logical reconstruction (2). On the other hand, a mechanically failed drive that is successfully repaired (1) will not need logical reconstruction.

In reality many physical problems will need subsequent logical reconstruction because not all data has been retrieved.

Dead Hard Drive

A drive can be considered "dead", if it is not accessible by any software means, e.g. the BIOS, Windows' Disk Management or disk utilities such as Runtime Software's GetDataBack. A dead drive often shows additional symptoms. It does not spin or it "clicks" or it makes other kinds of strange noises.

Theses drives might have a damaged electronic board, damaged read heads, a damaged motor or damaged magnetic media. Data recovery companies with clean room facilities can often resurrect the drive by exchanging the damaged parts. They will then image the drive and perform a logical file reconstruction.

This approach is sometimes successful and then well worth the cost of several hundred or even thousands of dollars; however, sometimes it is not successful.

Physical recovery is not always possible

First, success depends on the extent of the damage. It is not possible, even in theory, to recover data from a platter that was heated up to "Curie temperature" (which is 770°C for iron). This temperature completely demagnetizes the platters. It seems doubtful whether anybody will recover data from a drive that fell on a hard floor. If the platters are unbalanced due to bending or impact they will vibrate while spinning. If the vertical amplitude of this vibration is larger than the distance the read head flies at (50µm), the drive will sustain a permanent head crash making reading the magnetic information impossible and further destructing the surface. Horizontal vibration will make it impossible for the head to stay on the track, which is thinner than 1µm.

While we know that tire shops apply weights to the wheel in order to balance the tire, a comparable technology for unbalanced platters is unknown.

The only technology possibly capable of overcoming this problem is Magnetic Force Microscope (MFM) photography, since this technique does not require the platter to spin. However, MFM requires scanning the whole surface of the platter. The MFM moves from region to region, each region yielding a picture. This alone will take several months. Then all these pictures must be stitched together. A 20GB hard drive consists of 160,000,000,000 bits, probably 300,000,000,000 bits including overhead. Each bit is represented by a magnetic flux change. A picture displaying this flux change will probably use 100 bytes, thus inflating each bit by factor 1000. You will have to analyze the amount of 40 Tera byte of data. It is unknown if this technology is in use. It certainly is not "commercially available and affordable" because a data recovery would cost hundreds of thousands of dollars.

Second, success also depends on the drive type. Many data recovery companies can "do" certain drives but cannot do others. Modern drives are conditioned after their assembly to work perfectly with the parts built in, heads, platters etc. It is often not possible to use parts of another drive, even if both drives share the same model number.

There are no "magic" machines that are capable of recovering the data from any kind of drive. If the raw data can be retrieved, a subsequent logical reconstruction of the files must be performed.

Drives With "Bad Sectors"

These drives are still recognized by the BIOS or software such as GetDataBack, but they have read errors in one or more spots. After obtaining a drive image you will need to reconstruct the files from this image with GetDataBack.

Always make an image

You can create an image with disk utilities, such as Runtime's DiskExplorer or GetDataBack. It should be noted that you should not try to make the image yourself, if the drive makes unusual noises. The process of creating the image can further damage the media. Instead, you might give it to a data recovery service company.

The question of course is what data recovery service companies will do other than trying to create an image. You should ask them. Finally, it is your own decision if you want to try it yourself. Before you start you should be well prepared. You should have the imaging

software installed on a working computer and you should know how to use this software. You should have sufficient hard drive space to hold the image of the bad drive. You should be focused on this task and observe its progress. You should not do anything else on this computer at the same time, such as playing games or surfing the Internet. It is difficult to predict how long it will actually take to create the image. This depends primarily on the number of bad sectors on the drive and can take from 30 minutes up to several days.

Attach external drives directly to your motherboard's IDE port

If you have an external drive, for example a USB drive, you should remove it from its case and attach it to the computer as a second (slave) hard drive. Also, you should use the IDE ports on the motherboard, not the IDE ports on an additional PCI card. The build-in ports have a better error handling, which is important for your job. Once you have obtained an image you can run GetDataBack and recover the files.

Logical Reconstruction With GetDataBack for FAT

GetDataBack scans your drive or an image of your drive and tries to put together the original state of all files in the file system. GetDataBack is able to do this, even if certain file system structures are missing, such as partition table or boot records.

Let's examine in detail how GetDataBack for FAT recovers a file:
A file in a FAT file system is completely described by

1. Its *directory entry*
2. Its entry in the *File Allocation Table* (FAT)
3. The *allocated* clusters on the drive containing the content of the file

The directory entry is picked up during the initial scan of the drive when GetDataBack examines each single sector. It contains the file's name, size, date, time and the *first cluster* of its data.

The first cluster directly points to the initial cluster that is allocated for the file. It also points to a FAT entry that describes where the remaining parts of the file can be found. As it turns out IMG_2379.JPG uses the clusters 4-529.
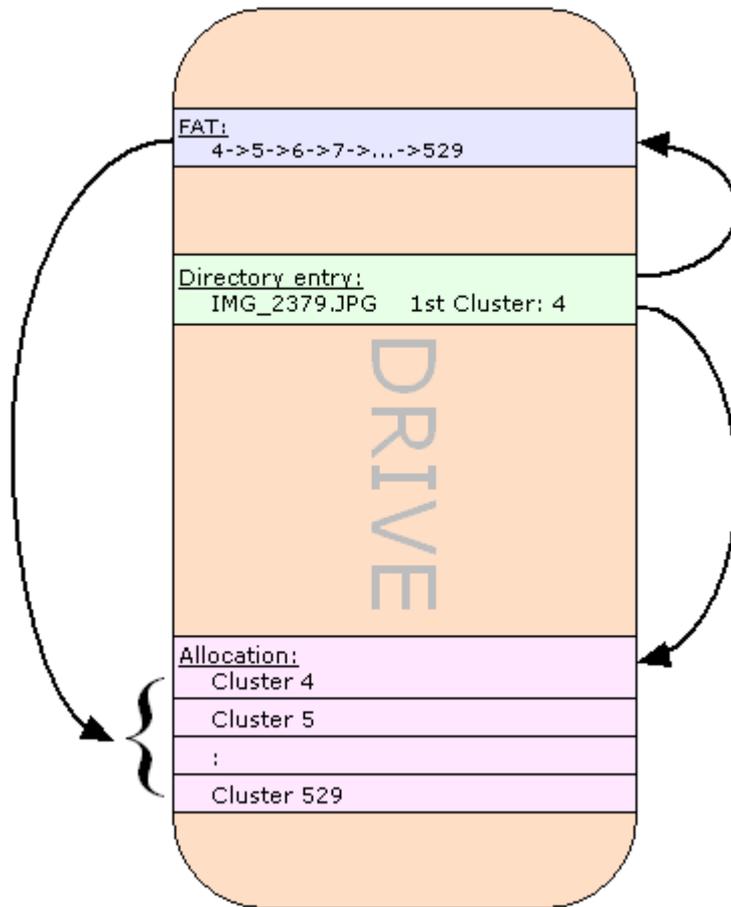
Information about a file in a FAT file system is spread among three different locations. The directory entry contains its name and where on the drive the file begins. The FAT knows where the file continues. Finally, the allocated clusters contain the file's content.

GetDataBack uses this information to reconstruct the files.

Because information about a file is stored at three different spots it will obviously cause problems if any of these are missing or incomplete.

FAT Recovery Matrix:

Alloc	Dir	FAT	
X	X	X	File will be recovered perfectly
X	X	-	File will probably be recovered. Problem with fragmented files.*
X	-	X	File has no name. It is possibly recoverable as a "lost file".**
X	-	-	
-	X	X	File is not recoverable, although its file name can still be seen.***

-	X	-	
-	-	X	File is not recoverable. No trace of its existence is left.***
-	-	-	

X This information is available
- This information in not available

* Fragmentation

A very common situation, caused by file deletion, format or partition deletion is a missing FAT entry. As long as the file size is smaller than the cluster size (e.g. 32 KB, depending on the drive size) you will get a perfectly recovered file because you actually do not need the FAT entry.

When the file is larger, it is usually allocated in consecutive clusters. This is why the most promising strategy for a data recovery software is to assume consecutive clusters when it rebuilds the file without FAT entry. This works for most of the files but runs into problems for files that increase over time. These files will necessarily be fragmented if they cannot be allocated consecutively because other files meanwhile use these clusters. Sadly, many important files fall into this category: Email files, databases, large documents and directories.

GetDataBack employs several techniques in order to recover even fragmented files correctly. These techniques include taking the allocation of other files into consideration. GetDataBack also is capable of reassembling fragmented directories. But make no mistake; these efforts are doomed to fail for large and heavily fragmented files.

As annoying as it is, although their content is still somewhere on the drive, these files are unrecoverable.

There is no automated data recovery software available that can solve fragmentation satisfactory. If you want to recombine a file consisting of 10 clusters on a 20 GB drive you must analyze, given a cluster size of 32 KB, all possible combinations of one known cluster with 9 other clusters out of possible 625000. There are 6250009 possible combinations, a number with 52 digits.

The only possible and more intelligent approach is a "manual" data recovery for a particular file. With Runtime's DiskExplorer you would begin at the known cluster and search downward looking for data you know belongs to the missing part of the file. Finally, you put all your findings together into a new file. The limitations of this approach are obvious. This can only be done for a couple of files with a known content.

Even data recovery service companies will most likely not produce better results. While they might have a couple of tools, e.g. for extracting readable text, they do not have your knowledge about the content of the file.

** Lost Files

If the directory entry was lost but the file content is still on the drive you might be able to recover the file if you knew where it is located. This problem is different from the fragmentation problem. You do not know the name, size and start of the file. This happens if the original directory entry was re-used by the operating system while the file content was left unchanged. If you formatted a drive and put 500 MB of a new Windows OS on the drive, the

first 500 MB of the drive, including a lot of directory information, will be destroyed, although the files themselves might still sit in locations beyond the overwritten 500 MB.

GetDataBack is capable of recovering many file kinds whose directory entries were lost. We call them "lost files". When GetDataBack examines each sector while scanning the drive it compares each sector to a list of user defined file signatures. For example, a Word document begins always with the bytes d0-cf-11-e0. If GetDataBack does not find a matching directory entry for this signature it will create a "lost file".

This way GetDataBack recovers lost DOC, JPG, BMP, ZIP and other files. The user can expand this list in the file GDB.INI or GDBNT.INI.

*** File's Allocation Was Overwritten

If the file's allocation - as in the four bottom cases in the recovery matrix - had been destroyed or overwritten by other data, there is no possibility at all to recover this file. Once overwritten, it is unfeasible to retrieve the information that was originally stored there. Theoretically, you might be able to read the "rest magnetization" with an advanced technology such as MFM (Magnetic Force Microscope), but it is unknown if anybody can actually do this. Certainly, if this technology exists it is not "commercially available and affordable".

It is important to understand that no data recovery software and no data recovery service company will be able to recover this file although you still might be able to see its file name in GetDataBack.

Logical Reconstruction With GetDataBack for NTFS

As we will see, NTFS is the better file system when it comes to data recovery. Usually, there is NO problem with fragmentation.

How GetDataBack for NTFS recovers a file:

1. Its *MFT* (Master File Table) entry
2. The *allocated* clusters on the drive containing the content of the file

The MFT entry is picked up during the initial scan of the drive when GetDataBack examines each single sector. It contains the file's name, size, date, time and the clusters occupied by its data. Other than the directory entry in a FAT file system it contains the complete list of the used clusters, called "run list".
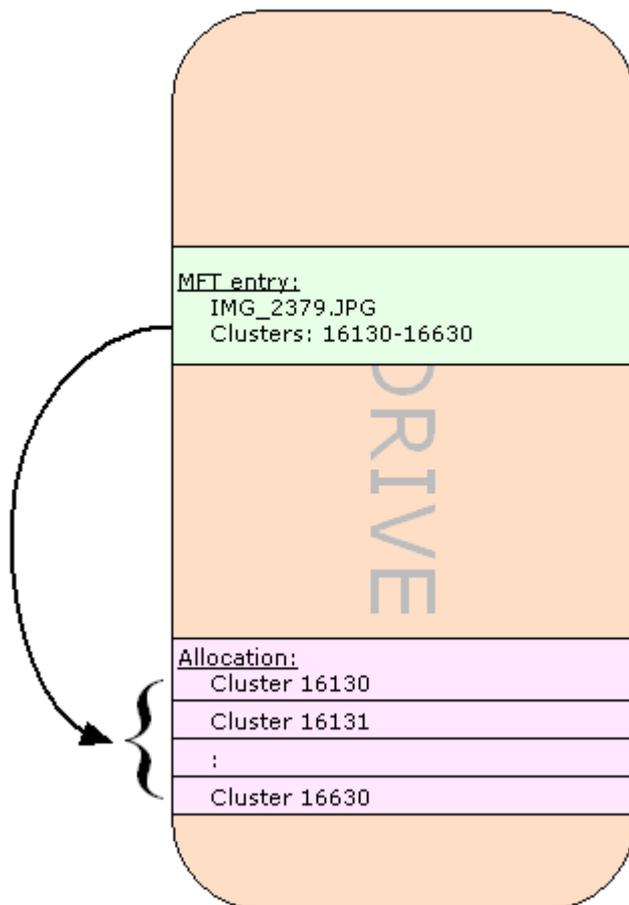
Sector	Name	Type	Attributes	Size	Date	1st cluster	NT Attributes
x00055D00	**IMG_2379.JPG**	FILE	a____	2051780	4/21/2004 1:50:08 PM	x003F02	10 30 80
351488	No: x6B8[xAAAD], Parent directory: x6B7[x2], Run: 22:F5 01 02 3F						
x00055D02	**IMG_2380.JPG**	FILE	a____	1652342	4/21/2004 1:50:24 PM	x0040F7	10 30 80
351490	No: x6B9[x2], Parent directory: x6B7[x2], Run: 22:94 01 F7 40						
x00055D04	**IMG_2381.JPG**	FILE	a____	1982323	4/21/2004 1:50:34 PM	x00428B	10 30 80
351492	No: x6BA[x2], Parent directory: x6B7[x2], Run: 22:E4 01 8B 42						
x00055		FILE	a____	1917452	4/21/2004 1:50:44 PM	x00446F	10 30 80
351494	y: x6B7[x2], Run: 22:D						
x00055		FILE	a			4	10 30 80
351496	ectory: x6B7[x2], Run:						
x00055		FILE	a			9	10 30 80
351498	No: x6BD[x2], Parent directory: x6B7[x2], Run: 22:0						
x00055D0C	**IMG_2385.JPG**	FILE	a			3	10 30 80
351500	No: x6BE[x2], Parent directory: x6B7[x2], Run: 22:B						
x00055D0E	**IMG_2386.JPG**	FILE	a			6	10 30 80
351502	No: x6BF[x2], Parent directory: x6B7[x2], Run: 22:B						
x00055D10	**MVI_2387.AVI**	FILE	a____	4111804	4/21/2004 1:52:08 PM	x004D1E	10 30 80
351504	No: x6C0[xAAB7], Parent directory: x6B7[x2], Run: 22:EC 03 1E 4D						
x00055D12	**MVI_2387.THM**	FILE	a____	4953	4/21/2004 1:52:08 PM	x00510A	10 30 80
351506	No: x6C1[x2], Parent directory: x6B7[x2], Run: 21:02 0A 51						

(Sector:Offset)=x00055D00:x000 (351488:0)   Selection=x00055D00:x000-x00055D00:x000

MFT entry for IMG_2379.JPG

The "run" describes the clusters that have been allocated for this file: x1F5 clusters beginning at cluster x3F02

The run directly points to the clusters that are allocated for the file. It turns out IMG_2379.JPG uses the x1F5 (501) clusters beginning at cluster x3F02 (16130).

MFT entry:
   IMG_2379.JPG
   Clusters: 16130-16630

DRIVE

Allocation:
   Cluster 16130
   Cluster 16131
   :
   Cluster 16630

Information about a file in an NTFS file system is spread among two different locations. The MFT entry contains both, the file's name and where on the drive it is allocated.

GetDataBack uses this information to reconstruct the files. Note that in NTFS, other than in FAT, we do not have a fragmentation problem. As soon as there is an MFT entry we exactly know where the file is allocated. This will yield better data recovery results for fragmented files.

Because information about a file is stored in two different spots it will obviously cause problems if any of these are missing or incomplete.

NTFS Recovery Matrix:

Alloc	MFT	
X	X	File will be recovered perfectly
X	-	File has no name. It is possibly recoverable as a "lost file".*
-	X	File is not recoverable, although its file name can still be seen.**
-	-	File is not recoverable. No trace of its existence is left.**

X This information is available
- This information is not available

* Lost Files

If the MFT entry was lost but the file content is still on the drive you might be able to recover the file if you knew where it is located.

You do not know the name, size and allocation of the file. This happens if the original MFT entry was re-used by the operating system while the file content was left unchanged. For example, if you formatted a drive and put 500 MB of a new Windows OS on the drive, the first 500 MB of the drive, including many MFT entries, will be destroyed although the files themselves might still sit on locations beyond the overwritten 500 MB.

GetDataBack is capable of recovering many file kinds whose MFT entries were lost. We call them "lost files". When GetDataBack examines each sector while scanning the drive it compares each sector to a list of user defined file signatures. For example, a Word document begins always with the bytes d0-cf-11-e0. If GetDataBack does not find a matching directory entry for this signature it will create a "lost file".

This way GetDataBack recovers lost DOC, JPG, BMP, ZIP and other files. The user can expand this list in the file GDB.INI or GDBNT.INI.

** File's Allocation Was Overwritten

If the file's allocation - as in the two cases below - had been destroyed or overwritten by other data, there is no possibility at all to recover this file. Once overwritten, it is unfeasible to retrieve the information that was originally stored there. Theoretically, you might be able to read the "rest magnetization" with an advanced technology such as MFM (Magnetic Force Microscope), but it is unknown if anybody can actually do this. Certainly, if this technology exists it is not "commercially available and affordable".

It is important to understand that no data recovery software and no data recovery service company will be able to recover this file although you still might be able to see its file name in GetDataBack.

Data Recovery From an Image After a Physical Problem (bad sectors)

When you run GetDataBack on an image obtained from a physically damaged drive, you will usually get good recovery results, assuming this image contains only "some" unrecoverable sectors.

Several factors contribute to this optimistic outlook:

A drive with bad sectors is usually not altered too much by user attempts to "fix" the problem.
If it is a FAT drive, the file allocation table and its copy are still there to be used by GetDataBack.
Most file system structures are available.

Of course, success depends on your ability to obtain this image. Files that were allocated in the damaged portions will be damaged after the recovery.

Data Recovery After Deleting/Recreating a Partition

When you delete a partition, only the partition table and the boot record are affected. Important structures, such as MFT and FAT are usually undamaged.

Even recreating the partition - as long as you do not format the volume - should not alter important data structures.

With GetDataBack you should be able to perform an almost perfect data recovery.

Data Recovery After Formatting

In FAT, formatting a volume clears both file allocation tables and deletes the root directory. All data is still there, but you have lost:

All entries in the root directory. Files can only be recovered as "lost files". Sub directories of the first level will have only numbers instead of their original name. Sub directories of deeper levels show their original name.
The file allocation tables. This will cause the "fragmentation problem" as discussed in the chapter "Logical reconstruction with GetDataBack for FAT".

Within the limitations above, you will get a "fair" data recovery. Most files should be uncorrupted. You will need to look for your files in the numbered directories. Fragmented files, such as Outlook email files or databases, will be corrupted and probably unusable.

In NTFS, formatting a volume creates a new MFT. However, this affects only the first 25 or so entries. It usually does not touch the MFT entries of previous user files.

That means you can expect a "good" data recovery. Almost all files should be correctly retrieved.

Your results will be even better when you formatted a drive that was previously FAT-formatted with NTFS or vice versa. In this case the original FAT or MFT will probably not be damaged because these structures are located at different areas on the drive.

Data Recovery After Installing a New Windows Operating System

Here's where the trouble really begins. Installing a new OS easily overwrites 2 GB.

All files that were once located in these 2GB will be irrevocably lost. Also, directories entries (FAT) and MFT entries (NTFS) located there will be lost, leaving only files without reference ("lost file"), that had been allocated in undamaged areas beyond the 2 GB.

In FAT this will also destroy the FATs, thus causing the *fragmentation problem*.

As explained before, a technology capable of recovering data from "rest magnetization" is not commercially available. All you possibly can recover will come from the not overwritten area.

Damage Assessment

Let's suppose you originally have a 20 GB FAT-formatted hard drive with 10 GB used for 50000 files in 2000 directories.

You put a new OS of 2 GB on that drive.

- You lose 10% of all data on the drive (2 of 20 GB).
- You lose 20% of your data on the drive (2 of 10 GB).
- You lose 20% of your files that were allocated in the overwritten area.
- Because most of the directory entries are located in the first 2 GB you lose an additional 30% of your files. (You might be able to recover some of these files as "lost files", without file names.)
- You lose an additional 10% of all files due to fragmentation and overlapping between the two areas.

You will be able to recover just about 40% of your files undamaged. 10% will be partly recoverable and possibly half of the "lost files", 15% will be recovered without file name.

If your files on the drive were "depending" on other files, e.g. tables for databases, this number drops even further:

- If your files depend on each other pair-wise, e.g. you have one Word document for "Contracts" and one for "Appendixes", only 0.4*0.4*100 = 16% of all pairs (projects) are left.
- If you have projects of 5 files each on the drive only $(0.4)^5*100 = 1\%$ of these projects are recovered without damage.

Note that, depending on the kind of data, losing 10% of the raw data can cause the loss of 99% of your projects.

If you are dealing with a previously NTFS-formatted drive your prospects are brighter:

- You will lose less files due to fragmentation and overlapping, only 5% instead of 10%.

- You also will lose less MFT entries than you would lose directory entries in FAT, 10% instead of 30%. NTFS tends to spread the MFT across the drive.

You would recover 65% of your files undamaged.

You would recover 42% of your 2-file *projects*, almost three times more than with FAT.

You would recover 12% of your 5-file *projects*, twelve times more than with FAT.

Data Recovery After Imaging or "Ghosting" a Drive

The consequences of imaging over a drive, for example with Norton's Ghost, are similar to the ones you face after putting a new OS on it. If the image was quite large, chances that you will recover a lot of files are pretty slim.

Data Recovery After Deleting Files

Although seemingly easy, this is tougher than recovering files from a bad sector drive or after an Fdisk or format.

File deletion is the least understood topic.

Ironically, what makes data recovery of deleted files so hard is the fact that the user can still work with his drive. His attempts to recover the just deleted files often ruin his chances.

Let's have a look at how the operating system deletes a file.

File Deletion In FAT

A single file gets deleted by:

1. Marking the directory entry with E5
2. Freeing the associated FAT entry

Whole directories are deleted by:

1. Marking the directory's directory entry with E5. The files' directory entries inside are usually left unchanged.
2. Freeing the FAT entries for both, the directory and the files inside.

After deletion there is a possible fragmentation problem because the allocation information, which is stored in the FAT, is irrevocably lost.

File Deletion in NTFS

A file gets deleted by flagging its MFT entry as unused. The MFT still contains the files allocation.

The Recycle Bin

What we described above applies to the permanent deletion of files. If you do not delete them permanently they are moved to the "Recycle Bin" and can be recovered from there.

While moving the files to the Recycle Bin they get renamed (for whatever reason) to numbers while keeping their extension. For example, "My vacation.DOC" will get a new name like "D24.DOC" in the Recycle Bin folder. This does not matter as long as these files are still in the Recycle Bin. The OS will provide you with the correct name when you choose to undelete these files.

If you "empty" the Recycle Bin, the deletion processes described above are carried out for these renamed files. If you later wish to recover "My vacation.DOC" you will actually have to look for an unknown file name with the extension "DOC".

Chances for successfully recovering deleted files

The locations of the deleted files are not protected by the file system anymore. Those locations might be recycled the next time the OS creates a new file. That's why it is such a problem, if the user continues to work with the affected hard drive.

Files are created all the time. Processes write log files. Therefore, even booting and running Windows from the affected drive can overwrite the critical areas. Browsing a single Website downloads multiple pictures to the local hard drive.

To protect those deleted files, the user must stop working with the drive immediately and connect it to another computer as a second (slave) drive.

We've tested how long a deleted file was recoverable before the OS recycled the deleted file's directory entry, MFT or allocation. It happened almost instantly. This leads us to be very pessimistic about the prospects of recovering "a couple" of deleted files.

Whereas, if you deleted - let's say 1 GB consisting of 1000 files - and do not continue to work with this drive, chances to recover most of these files are pretty good. If you work with FAT, you will possibly face the fragmentation problem.

Recovering Data In Time

The time dimension often gets underestimated when it comes to data recovery. Losing data for a week can be as bad as losing the data forever.

This emphasizes the importance of a data recovery software like Runtime's GetDataBack. Appointing a data recovery service company will always have a turnaround time of several days. Doing it yourself just requires a little preparation and after a couple of hours the data recovery is completed.

While you have to seek professional help for physically damaged drives, most data losses are logical file system corruptions and the recovery services won't use better tools than you can use.