

File Shredders....

File shredders “delete” a file – but they do that in a way that is different from how the Windows operating system (and all regular Windows applications) delete files. To understand the difference, we'll also look at

... and, just what is a file?

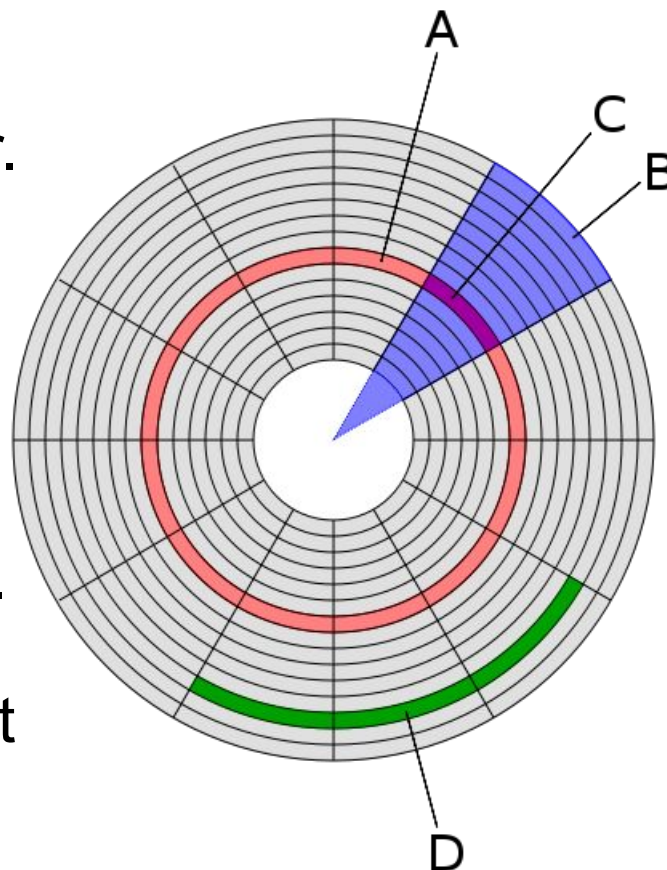
Disk Structure - Terms

(A) Track. The same track across multiple data surfaces is called a Cylinder.

(B) Geometrical sector

(C) Track sector: or just “sector” or “block”. The minimum I/O unit. The standard size was 512 bytes, now is 4KB.

(D) Cluster: the minimum # of sectors that can be allocated (in this example, the cluster factor = 3).



A drive can have several data surfaces. Thus to specify a single “track sector” on a disk, you need to specify Cylinder, Head & Sector. This “CHS” addressing is usually mapped to a single number.



Table of File Pointers

At the beginning of every partition, there is a directory table (File Allocation Table, Master File Table, ...) - a special type of data area that defines a directory (also known as a folder).

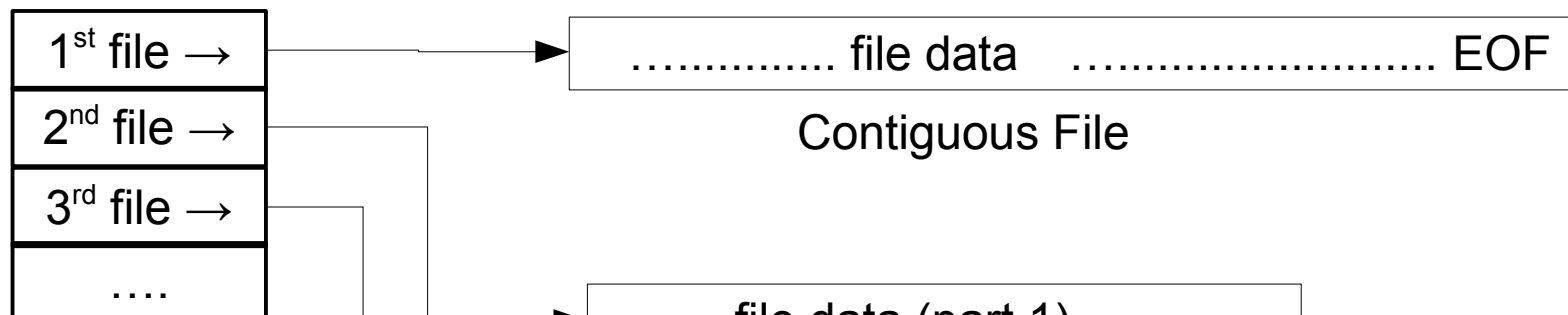
Each entry within it is a fixed-length record which stores the name & extension, attributes (size, archive, directory, hidden, read-only, system, date and time of creation,...), and finally a pointer to (address of) the first sector of the file data.

1 st file
2 nd file
3 rd file
....

Directory & Data Areas

Directory Table
(Root-directory)

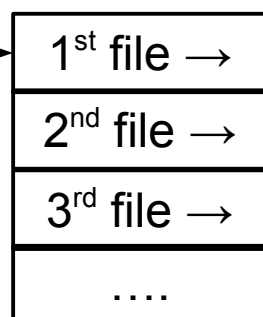
End-Of-File mark



..... file data (part 1).....→

..... file data (part n) EOF

Fragmented File

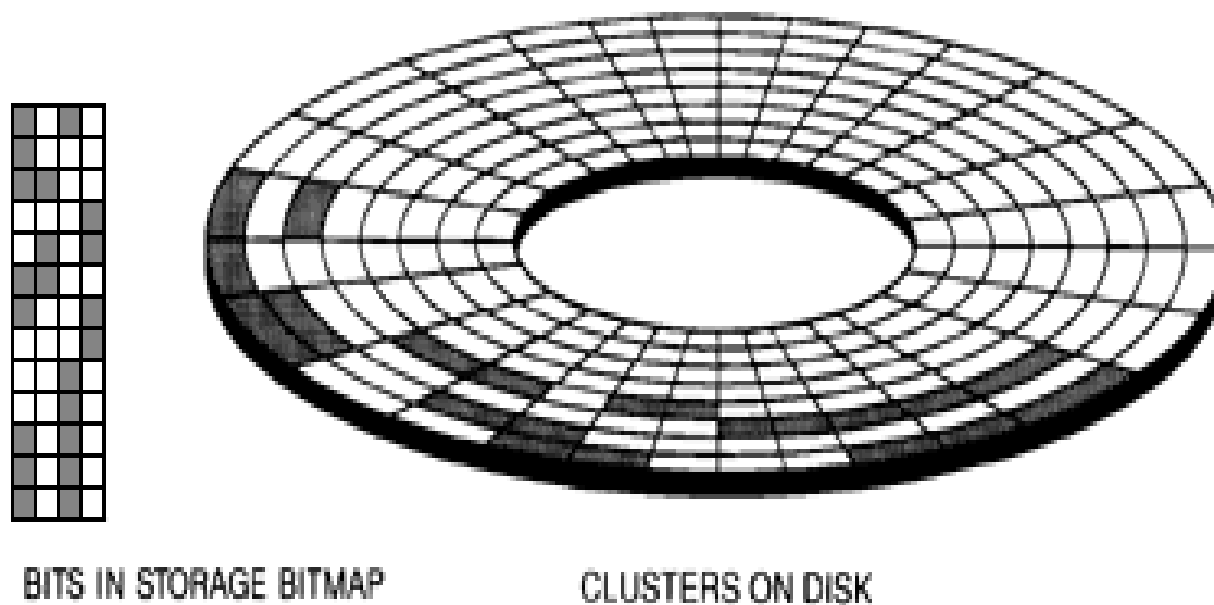


Sub-directory

Where a file's table entry resides determines which directory the file appears in – not anything about where the actual data is on the disk.

Bitmap

An array of bit entries: each bit indicates whether its corresponding cluster is used (allocated) or free (available for allocation).



Every File Has Three Components

- So, files have at least three components / data areas (although stored & configured quite differently in different file systems).
 - File Table
 - Data area
 - BitMap
- If you look at each entry in the File Table, and follow each link to the EOF mark, you can identify every block that's used. So, the BitMap data are redundant, but are needed for speed.
- The BitMap can also show areas of the disk that were not properly deleted, but have lost their File Table entries. These areas are called “Lost Clusters”

Lost Clusters

The CHKDSK utility could be used to recalculate the BitMap from the File Table, and compare that against the stored BitMap. e.g.

> chkdsk C: /f

If the two BitMaps don't agree, it will report (in our example above):

3 lost clusters found in 2 chains.

Convert lost chains to files (Y/N)?

Two new files would be created on the root directory of the disk with the following filenames:

FILE0000.CHK

FILE0001.CHK

Media-specific File Structures

Some features of the file system / device driver are specific to the type of disk / media. For example:

Hard Drives: have a “sector skew”

Flash Drives: have a unique algorithm for selecting where to write a file

Writeable CDs: have an extensible file table

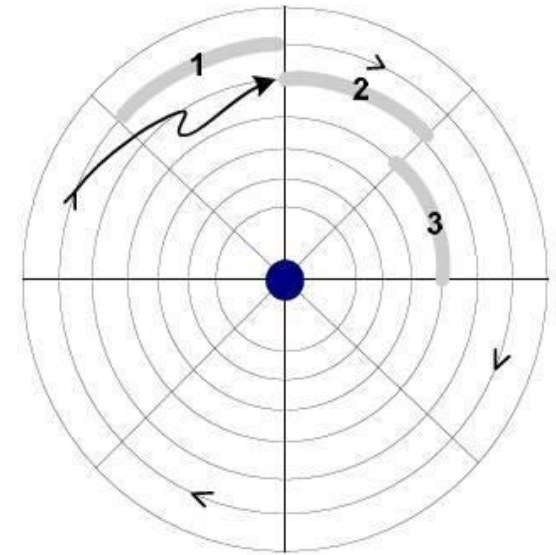
Hard Drives: Sector Skew / Offset

Given: you have a large file (one that will require several tracks to hold all the data). You're operating in an area of the disk where there are no other used blocks to have to worry about. Just how are the data stored?

Let's say the file starts in sector 1, and data are written over the entire track. Where does writing continue?

The head will move to the adjacent track, but that takes some time, during which the disk is still spinning. So, writing resumes at sector 2. If a third track is needed, it would start at sector 3.

Each track is skewed from the previous track. But, these skips do not constitute a fragment. This is a contiguous file, and has only one pointer and one fragment.



Flash Drives: File Distribution

Each cell of commonly used flash memory is good for around 10,000 writes. This may be “forever” for an area storing e.g. photos, but not for an area used as a “Temporary Directory”.

Some file systems are designed to distribute usage over an entire memory device without concentrating usage on any part (e.g. for a directory); this even distribution prolongs the life of simple flash memory devices.

Other notes on Flash Drives:

- Flash drives can be defragmented, but this brings little advantage as there is no mechanical head that moves from fragment to fragment.

Writable CD: Multisession

ISO 9660 is by design a read-only, pre-mastered file system. This means that all the data has to be written in one go to the medium. As with HD & Flash, partition tables are contiguous (cannot be fragmented).

With multiple-session writing, data can be written to disc, then later more data can be added as long as there is unused space left on the disc. With the Multisession extension to ISO 9660, if a CD mastering program wants to add a single file to a CD-R, it has to append a session containing at least an updated copy of the entire partition table, plus the new file. The duplicated directory entries can still reference the data files in the previous session(s).

In a similar way, file data can be updated or even removed. Removal is, however, only virtual: the removed content does not appear any more in the directory shown to the user, but it is still physically present on the disc. It can therefore be recovered, and it takes up space (such that the CD will become full even though appearing to still have unused space).

Deleting a File

- Now we can finally ask: “What happens when a file is deleted”. This question may be more involved than you think.
- As we've discussed, there are 3 parts to a file, and there are also many types of “deletion”.
 - Move (within a partition)
 - Move (between partitions)
 - Delete
 - “Permanently” Delete
 - Shred
- So, the task is to fill in the following table

Deletion: What Really Happens

	File Table	BitMap	Data
Move (intra-partition)	Entry relocated	No Change	No Change
Move (inter-partition)	Entry made in a different table Old entry cleared	New bits set Old bits cleared	New data written Old data unchanged
Delete (to the Recycle Bin)	Entry relocated	No Change	No Change
Delete (Permanently)	Entry Cleared	Bits Cleared	No Change!!
Shred			

*These are the only options from Microsoft –
and none of them delete the data!*

Deletion: What Really Happens

	File Table	BitMap	Data
Move (intra-partition)	Entry relocated	No Change	No Change
Move (inter-partition)	Entry made in a different table Old entry cleared	New bits set Old bits cleared	New data written Old data unchanged
Delete (to the Recycle Bin)	Entry relocated	No Change	No Change
Delete (Permanently)	Entry Cleared	Bits Cleared	No Change!!
Shred	Entry Cleared	Bits Cleared	Data Cleared!!

File Shredder v2.0

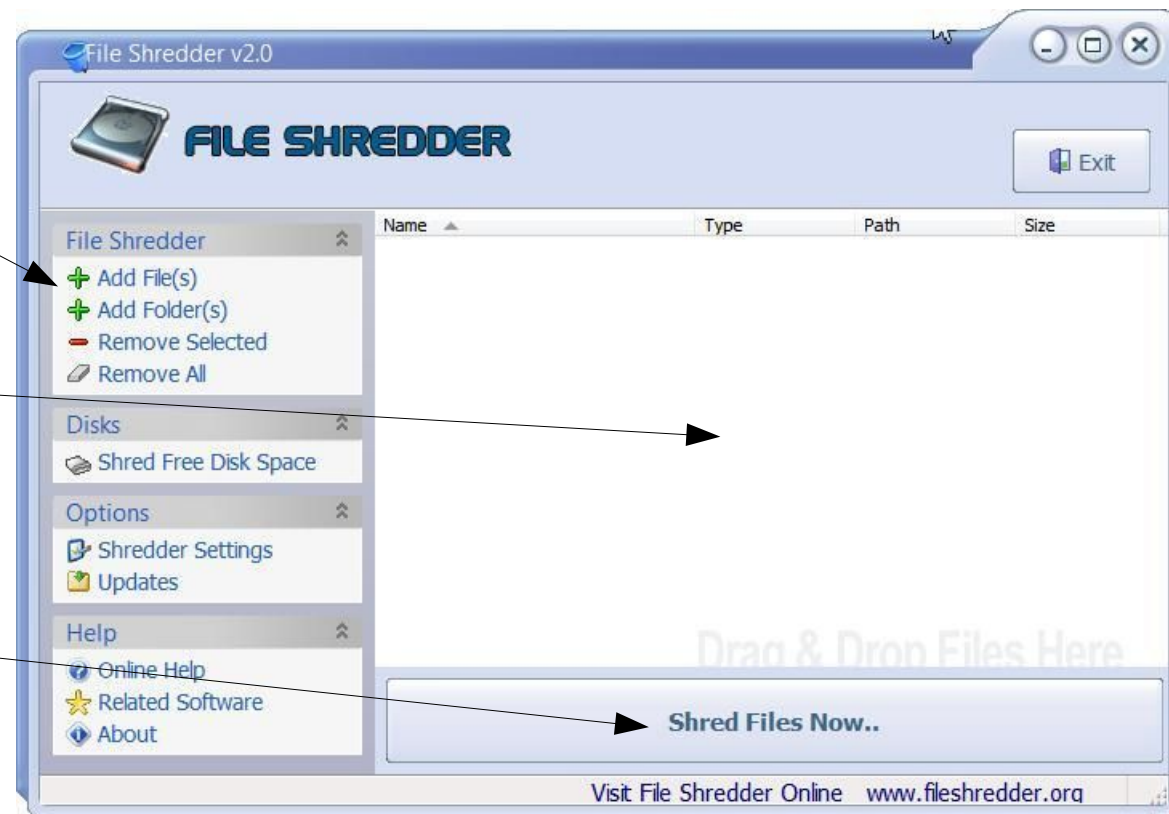
<http://www.fileshredder.org> or the club's Freeware Page

The Windows Explorer extension lets you shred the selected file, or bring up the File Shredder app.

It's easy to use:

1) Select files and/or folders
or just “drag & drop”

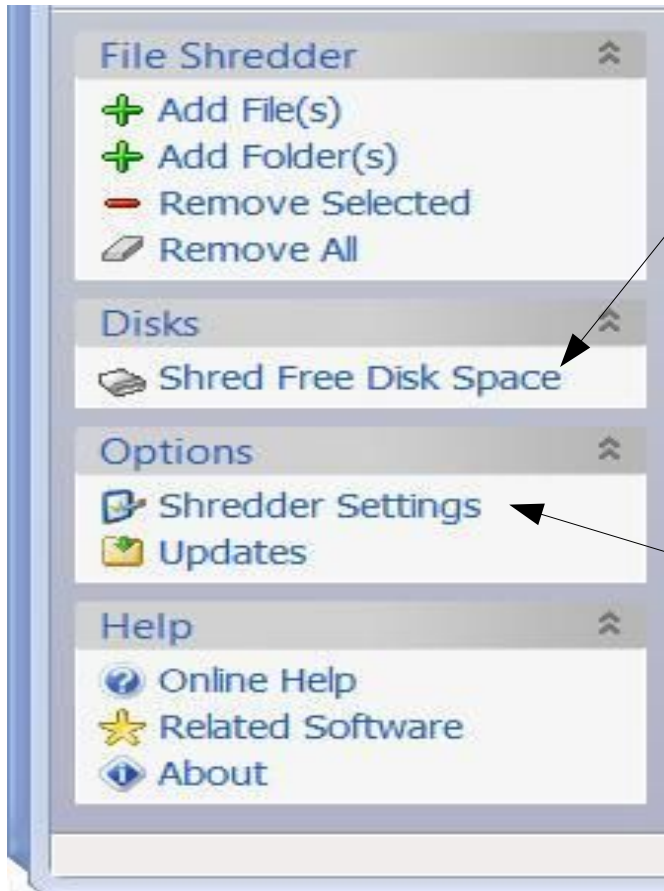
2) Shred



It's tiny about 2 MB

File Shredder – Options & GUI

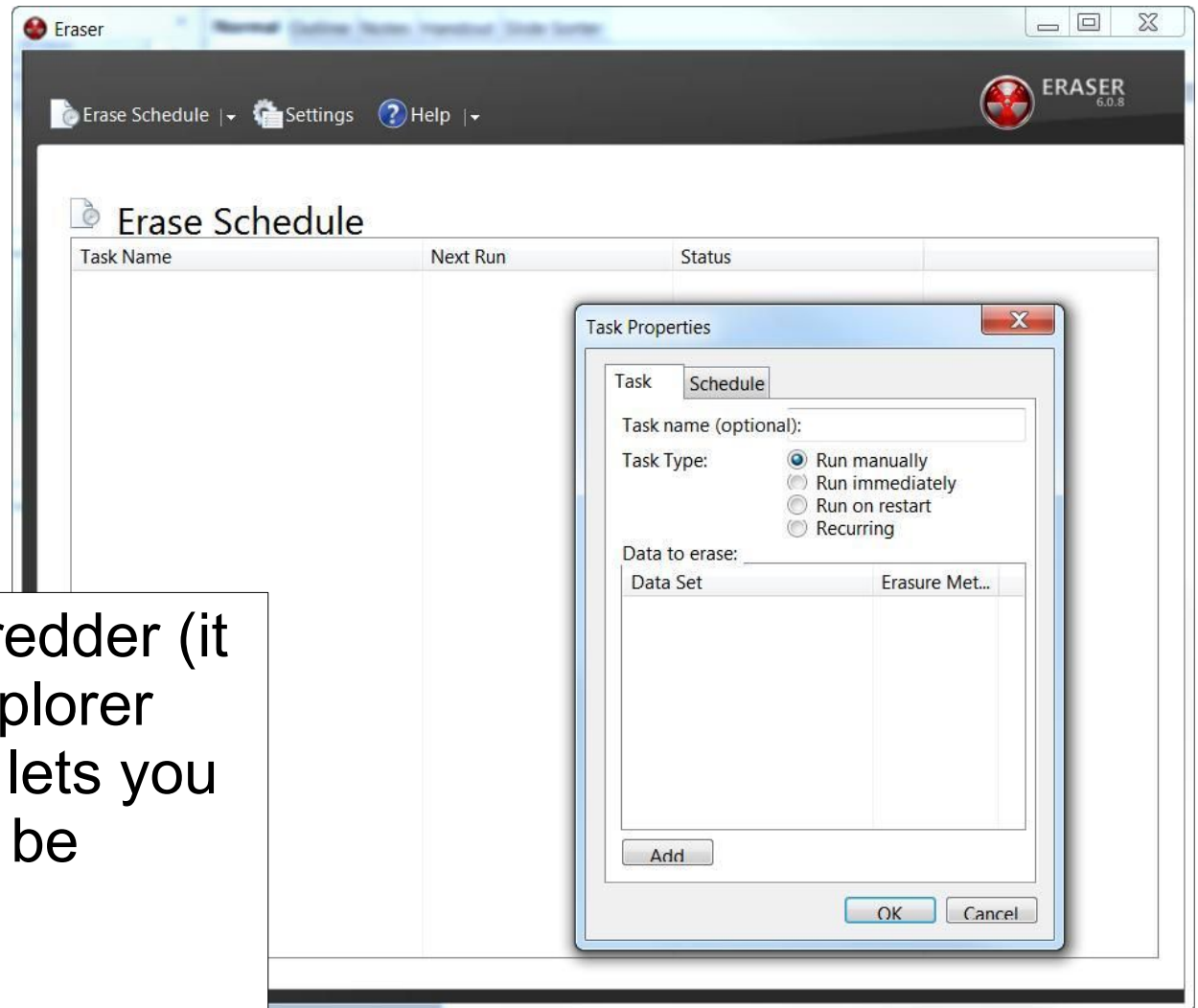
Shred all “unused space”



Eraser [\(<http://eraser.heidi.ie>\)](http://eraser.heidi.ie)

Very similar to File Shredder (it too has a Windows Explorer extension), but Eraser lets you define “tasks” that can be scheduled.

It's large (~50 MB).



Eraser – even more overwrite options

