superuser

What is the difference between size and size on disk?

Asked 10 years ago Active 1 year, 6 months ago Viewed 90k times

85

24

Looking at the properties for a Windows file I get two attributes, "Size" and "Size on disk", and "Size on disk" is always larger.

What do these two metrics mean?

windows filesystems

edited Sep 27 '14 at 1:54
Journeyman Geek ♦
119k 47 228 384

asked Nov 6 '09 at 17:56
Gavin Miller
1,686 3 15 23

1 I am not 100% sure on this, but I believe that compressed files also have an effect on these measurements. – AdminAlive Nov 6 '09 at 18:15

3 "Size on disk" is not always larger. Small files are stored directly on the MFT and will have size on disk = 0. Compressed files often also have smaller size on disk. Same with sparse files – phuclv Jul 31 '17 at 16:20

4 Answers

78

Size is the actual size of the file in bytes.

Size on disk is the actual amount of space being taken up on the disk. They differ because the disk is divided into tracks and sectors, and can allocate blocks of discrete size.

Editing

> We know that a disk is made up of Tracks and Sectors. In Windows that means the OS allocates space for files in "clusters" or "allocation units".
>
> The size of a cluster can vary, but typical ranges are from 512 bytes to 32K or more. For example, on my C:\ drive, the allocation unit is 4096 bytes. This means that Windows will allocate 4096 bytes for any file or portion of a file that is from 1 to 4096 bytes in length.
>
> If I have a file that is 17KB (kilo bytes), then the Size on disk would be 20.48 KB (or 20480 bytes). The calculation would be 4096 (1 allocation unit) x 5 = 20480 bytes. It takes 5 allocation units to hold a 17KB file.
>
> Another example would be if I have a file that is 2000 bytes in size. The file size on disk would be 4096 bytes. The reason is, because even though the entire file can fit inside one allocation unit, it still takes up 4096 of space (one allocation unit) on disk (only one file can use an allocation unit and cannot be shared with other files).
>
> So the size on disk is the space of all those sectors in which the file is saved. That means,usually, the size on disk is always greater than the actual size.
>
> So the actual size of a file(s) or folder(s) should always be taken from the **Size** value when viewing the properties window.

Source: [What's The Difference Between Size And Size On Disk In Windows Folder Properties](#).

edited Jul 31 '17 at 15:38
simlev
3,124 3 6 28

answered Nov 6 '09 at 17:58
Am1rr3zA
4,645 9 41 53

1 So should I look at "size" or "size on disk" when I wish to compare the percentage of how much a folder takes compared to the total that the current partition has? – android developer Oct 24 '14 at 23:17

1 @androiddeveloper size on disk it is – Am1rr3zA Oct 24 '14 at 23:36

OK, thank you. Wonder why they didn't explain it there, or put a little better description. – android developer Oct 25 '14 at 8:46

5 The answer by Synetech below adds important (and potentially confusing) points about Compression and Hard links, both of which can lead to a Size on Disk that is *smaller* than the Size. – Owen Blacker Nov 21 '14 at 18:04

1 @baroquedub You *can* have *massive* difference between the two (like the x1000 factor in your example). This difference can happen especially if there are lots of small files (basically because files are written as "blocks" on the disk, so at least the whole size of one block will be taken. The actual size of the block depends on the file-system, so the size on disk taken may be different on different disks. – Pacopaco Jan 16 at 15:03

24

Imagine you have two 2 x 10 gallon gas cans in your car. Each gas can is an allocation unit. You need to get 12 Gallons of gas, so you need to use both cans. Basically using 20 Gallons of allocated space - but only filling 12 gallons.

Here is the default size for Windows XP

```
Drive size
(logical volume)             Cluster size          Sectors
----------------------------------------------------------
512 MB or less               512 bytes             1
513 MB - 1,024 MB (1 GB)     1,024 bytes (1 KB)    2
1,025 MB - 2,048 MB (2 GB)   2,048 bytes (2 KB)    4
2,049 MB and larger          4,096 bytes (4 KB)    8
```

If you think of the Cluster size as each of your gas cans: Holding 4KB of "gas" each. But your file is 2KB then the fills size is 2K, but size on disk is 4KB

edited Jul 31 '13 at 21:54
nixda
22.8k 11 83 140

answered Nov 6 '09 at 18:03
Holmes
534 2 5

6 Allow me to add to your answer. The allocation unit (bucket) size is chosen based on the size of the disk. If you're using a bucket to empty a bathtub, you would choose a smallish bucket. If you're emptying a swimming pool, you'll use a bigger bucket. – Les Nov 6 '09 at 20:47

13

Cluster Slack Space

You cannot access each individual byte on a storage medium separately. To do so would be terribly inefficient because the system needs some way of keeping track of which ones are used and which are free (ie, a list), so doing so for each byte separately would create too much overheard (for *each* individual byte, ie 1-to-1, the list would be as big as the medium itself!)

Instead, the medium is broken up into chunks, blocks, units, groups, whatever you want to call them (the technical term is *clusters*), each of which contains a—consistent—number of bytes (you can usually specify the size of the clusters since different uses call for different sizes to reduce waste).

When a file is saved to disk, the size of the file is divided by he cluster size and rounded **up** if needed. This means that unless the filesize is exactly divisible by the cluster size, some of the cluster end up being unused and thus wasted.

When you view the properties for a file, you see the true size of the file as well as the size it takes up on disk which includes any “slack”, that is, the “cluster tips” that are unused. This is usually not much *per-file* and the *size on disk* will usually be almost equal to the actual size, but when you add up the wasted space from all the thousands of files on a drive, they can add up. Therefore, when you view the size of a large folder, especially one with many tiny files that are smaller than a cluster, the size on disk (ie, the amount of disk space marked as used) can end up being significantly larger than the actual size (ie, the amount space the files actual require).

In a case like above, what you can try is to reduce the cluster size so that each file wastes less space. Generally, a drive with mostly lost of little files should use the smallest cluster size possible (to reduce waste) and a drive with mostly large files should use the largest cluster size possible (this way the bookkeeping structures end up being smaller).

Even at a lower level, if each cluster is only a single sector, unless a file is an exact multiple of the size of the sectors on the drive (usually 512 bytes traditionally, now often 4,096 with Advanced Format disks), then there will still be unused space between the end of the file and the end of the sector.

Compression

Another scenario where you might see a difference between the actual file size and *size on disk* is with compression. When a drive is compressed (e.g., using DriveSpace, NTFS compression, etc.) then there will be a difference between the size of the actual file (which needs to be know), and the actual size that the file occupies (i.e., uses or “takes up”) on the disk.

Shortcuts and Hardlinks

Yet another scenario that could result in a difference is with hardlinks. With file-systems that support hardlinks, when a duplicate file is created, instead of making a whole new file that takes up space for itself, the file-system creates a shortcut to the file so that both (or all three, etc.) copies point to the same physical file on disk. Therefore, when there are two files pointing to the same data, they each have the same size, but take up only slightly more than the space to store a single copy.

edited Jul 31 '13 at 21:11

answered Oct 16 '11 at 21:40

Synetech
62.7k 30 194 327

Actually with 1B allocation units, the list wouldn't necessarily take up the entire medium. Just an eighth of the size. This is because you only need a single bit to say whether a block is used or free. – flarn2006 Dec 28 '16 at 7:06

Overhead also includes data that indicates several allocation units belong to the same file. If you say each byte has another bit indicating if the data overflows into the next byte, that solves that but is too naive for modern disk size/performance because if the next byte isn't free EVERY byte of the hard drive may need to be moved. Realistically, you'd need more overhead in order to specify offset of next allocation unit or assign a file id of sorts to each

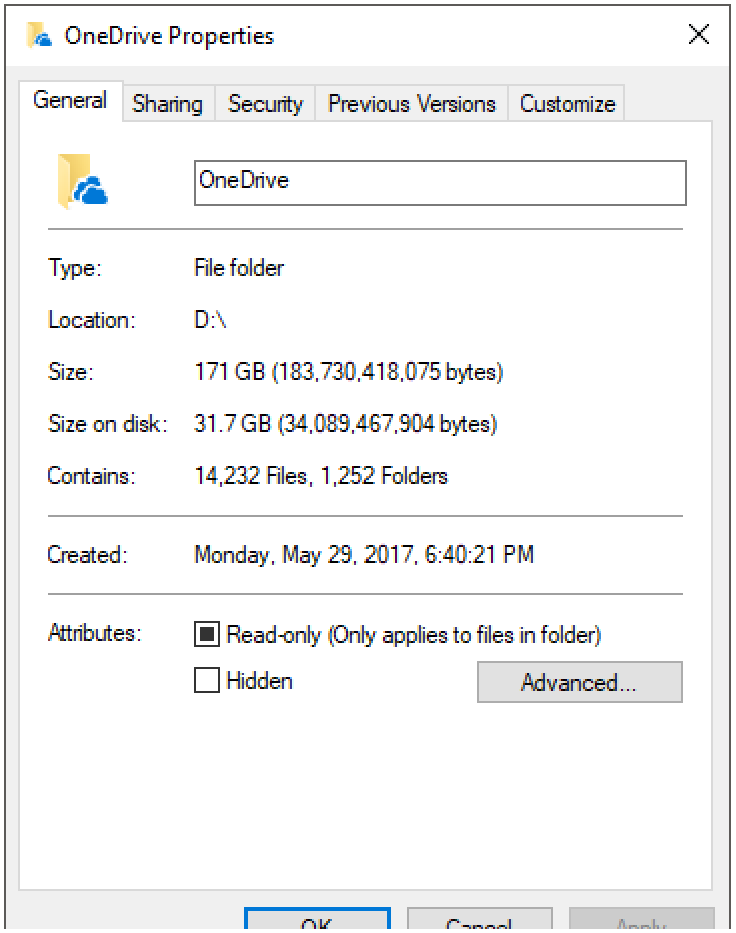
@flarn2006, theoretically maybe, but there's no such thing as 1-byte allocation units; the minimum is a single sector which is 512 (or 4,096 with Advanced Format disks). – Synetech Nov 1 at 21:04

@RetiredAssistant, the overhead of the MFT/FAT is not indicated in the size/on-disk field by Windows. – Synetech Nov 1 at 21:05

4

Another thing that may significantly reduce the Size on Disk value are situations where a file is not actually stored on disk but is still accessible through various means.

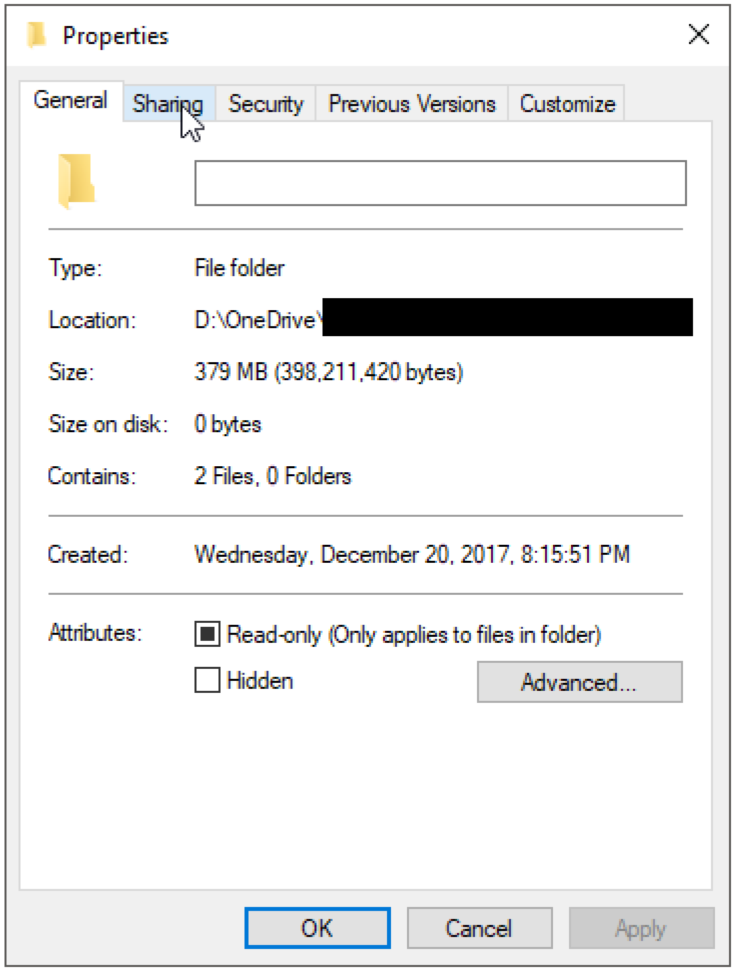
For example, the Offline Files feature of OneDrive enables a user to store a file in such a way that it is accessible via an internet connection. The file still exists on disk and has a certain size, but because it is not on disk until it is downloaded, it takes up no space.



OneDrive

Type:	File folder
Location:	D:\
Size:	171 GB (183,730,418,075 bytes)
Size on disk:	31.7 GB (34,089,467,904 bytes)
Contains:	14,232 Files, 1,252 Folders
Created:	Monday, May 29, 2017, 6:40:21 PM
Attributes:	■ Read-only (Only applies to files in folder)
	☐ Hidden

Example on a folder inside...



Type: File folder

Location: D:\OneDrive\ [redacted]

Size: 379 MB (398,211,420 bytes)

Size on disk: 0 bytes

Contains: 2 Files, 0 Folders

Created: Wednesday, December 20, 2017, 8:15:51 PM

Attributes: [■] Read-only (Only applies to files in folder)

[ ] Hidden

answered May 19 '18 at 4:42

Justin Krejcha

1,863 1 11 25