

# When Windows copies a file, does it ever copy bytes that are in the slack space?

[devblogs.microsoft.com/oldnewthing/20180213-00](https://devblogs.microsoft.com/oldnewthing/20180213-00)

February 13, 2018



Raymond Chen

A customer who works with highly sensitive information wanted confirmation that when they copy a file with Explorer, Windows will copy only the data that logically belongs to the file and no data that happens to reside in the slack space.

*Slack space* refers to physical storage allocated to a file but not used to hold any file contents. Slack space typically appears if the last unit of file data storage is not filled with file data. For example, if you have a cluster size of **4KB**, and the file is 10KB in size, then one way of storing the information is to allocate three clusters: The first cluster holds the first 4KB of data; the second cluster holds the next 4KB of data, and the third cluster holds the last 2KB of data. The last 2KB of the third cluster is unused.

The concern is that the last 2KB of the third cluster, which formally does not contain any file data, may nevertheless be copied to the destination file (also as slack data). When copying the last piece of the file, is it possible that Windows reads the entire cluster (even the slack space), and then writes the entire cluster (including the slack space) to the destination? If there is any confidential information in the slack space of a non-confidential file, then could copying the non-confidential file inadvertently copy the confidential information?

Fortunately, the answer is no.

The contents of the slack space are not visible outside the file system driver itself. File copying is handled at a higher level. For example, the `CopyFile` function has access only to user-mode-visible file contents. It cannot see the slack space in the source file, and therefore cannot copy it.

Even in the case of offloaded data transfer (ODX), the code that performs the transfer communicates with the file system, and the file system driver won't let anybody see the data in the slack space. Therefore, the transfer will not transfer any data that resides in the slack space of the source file.

On the other hand, if you are copying data by operating at a level below the file system driver, then the file system driver can't stop you from seeing the slack space. For example, if you use direct volume access and read sectors straight off the hard drive, you will see everything on the volume, including slack space.

Raymond Chen

**Follow**

