What does the FOF_NOCOPYSECURITYATTRIBS flag really do (or not do)?

devblogs.microsoft.com/oldnewthing/20101015-00

October 15, 2010

Raymond Chen

In the old days, the shell copy engine didn't pay attention to ACLs. It just let the file system do whatever the default file system behavior was. The result was something like this:

- If you copied a file, it opened the destination, wrote to it, and that was it. Result: The copied file has the security attributes of destination (specifically, picking up the inheritable attributes from the container).
- If you moved a file within the same drive, it moved the file with `MoveFile`, and that was it. Result: The file retained its security attributes.
- If you moved a file between drives, then it was treated as a copy/delete. Result: The moved file has the security attributes of the destination (specifically, picking up the inheritable attributes from the container).

Perfectly logical, right? If a new file is created, then the security attributes are inherited from the container. If an existing file is moved, then its security attributes move with it. And since moving a file across drives was handled as a copy/delete, moving a file across drives behaved like a copy. Users, however, found this behavior confusing. For example, they would take a file from a private folder like their *My Documents* folder, and move it to a public location like *Common Documents*, and... the file would still be private. The `FOF_NOCOPYSECURITY-ATTRIBS` flag was introduced in Windows 2000 to address this confusion. If you pass this flag, then when you move a file, even within a drive, the security attributes of the moved file will match the destination directory. (The way the shell implements this flag, by the way, is to move the file like normal, and then reset the security attributes to match the destination. So even though it sounds like a flag that says "don't do X" would be less work than doing X, it's actually *more* work, because we actually do X+Y and then undo the X part. But it's still far cheaper than copying the file and deleting the original.) Note that omitting the `FOF_NOCOPY-SECURITYATTRIBS` flag does not mean "Always copy security attributes." If you don't pass the flag, then the security attributes follow the default file system behavior, which sometimes transfers the security attributes and sometimes doesn't. In retrospect, the flag might have been better-named something like `FOF_SETSECURITYATTRIBSTOMATCHDESTINATION`. Now the question is how to summarize all this information in the MSDN documentation for the

`FOF_NOCOPYSECURITYATTRIBS` flag? After receiving this explanation of how the flag works, one customer suggested that the text be changed to read "Do not copy the security attributes of the moved file. The destination file receives the security attributes of its new folder. Note that this flag has no effect on copied files, which will always receive the security attributes of the new folder." But this proposed version actually can be misinterpreted. Everything starting with "Note that" is intended to be guidance. It isn't actually part of the specification; rather, it's sort of "thinking out loud", taking the actual specification and calling out some of its consequences. But how many people reading the above proposed text would fail to realize that the first two sentences are normative but the third sentence is interpretive? In particular, the interpretation says that the copied file will "always" receive the security attributes of the new folder. Is that really true? Maybe in the future there will be a new flag like `COPY_FILE_INCLUDE_SECURITY_ATTRIBUTES`, and now the "always" isn't so "always" any more.

Anyway, now that you know what the `FOF_NOCOPYSECURITYATTRIBS` flag does (and doesn't do), maybe you can answer this customer's question:

> Download a file via Internet Explorer and put it on the desktop. The file will be marked as having come from the Internet Zone.
>
> Now copy the file with the `FOF_NOCOPYSECURITYATTRIBS` to some other location.
>
> The resulting file is still marked as Internet Zone. I expected that `FOF_NOCOPYSECURITY-ATTRIBS` would remove the Internet Zone security information. Is this a bug in `SHFileOperation`?

(This article provides enough information for you to explain why the Internet Zone marker is not removed. The answer to the other half of the customer's question—actually removing the marker—lies in this COM method.)

Raymond Chen

Follow