How do the FILE_SHARE_* bits interact with the desired access bits?

devblogs.microsoft.com/oldnewthing/20040511-00

May 11, 2004

Raymond Chen

It's really not that complicated. If you permit, say, `FILE_SHARE_READ`, then you're saying, "I'm okay with other people reading this file while I have it open." And if you leave off the flag, then you're saying, "I do not want other people reading this file while I have it open." Now all that's left to do is work out what that means. So suppose you omit the flag, indicating that you don't want to let others read the file. Then when you attempt to open it yourself, the open will fail if anybody else has the file open for reading. And if the open succeeds, then the system will prevent anybody else from opening the file for reading until you close your handle. That's all. Of course, if the file is already open, then a corresponding check is made between your desired access and the file sharing mode of the people who already opened it. For example, if somebody already has the file open and denies read sharing, then if you try to open for read, you will get a sharing violation. These restrictions are cumulative, of course. If one person opens a file without `FILE_SHARE_READ` and another person opens a file without `FILE_SHARE_WRITE`, then attempts to open the file for read or for write will fail. (The read fails because the first person didn't permit read, and the write fails because the second person didn't permit write.) Repeat the above logic for "delete" and "write" permission, and that's it in a nutshell. There is a big nasty table in MSDN that walks through all the combinations, but I personally think it confuses the matter rather than clarifying. Even more confusingly, the table uses "X" to mean that the combination is **permitted**. They should've used a bullet ("•") or a check mark ("ü"), since an "X" has the connotation of "not allowed". Let's look at one row of the table and see how the information in it is "obvious": Say, the row that reads "GENERIC_READ / GENERIC_WRITE / FILE_SHARE_READ". You are asking for read and write, and you permit read (and implicitly deny write). The requested access (read/write) requires that all previous openers have granted both read and write. There are three columns that correspond to this, namely the ones that say "FS_R FS_W". The requested sharing mode (read only) requires that all previous openers have requested read-only access. In other words, there can't be any G_W entries. That rules out two of the columns, leaving just "G_R FS_R FS_W", and indeed only one column is checked in the table.

Notice that the file share bits you pass don't have to match up with your file access bits. The file share bits indicate what you want to allow other people to do. The access bits indicate what you want to do yourself.

Raymond Chen

Follow