

RE: CreateFile() and FILE_FLAG_WRITE_THROUGH

Source:

<http://www.tech-archive.net/Archive/Development/microsoft.public.win32.programmer.kernel/2008-09/msg00136.htm>

- *From:* Hugo_gleaves@xxxxxxxxxxx <hugh<underbar>>
 - *Date:* Fri, 19 Sep 2008 02:13:02 -0700
-

Just to clarify, if you have the file opened with these flags and you do a write, when that write returns you can be 100% confident that the data is on the physical media.

"Nick Tucker" wrote:

Hi everyone

If I create a file using CreateFile(), specifying the FILE_FLAG_WRITE_THROUGH flag, then write to that file, the file still shows as zero bytes long in the directory and in Windows Explorer, and if I then turn the power off, that file is still zero bytes long on restart, so the data I wrote is unavailable. I have tried this test on a variety of operating systems including Vista, XP and Windows Server 2003/2008 using a variety of hardware platforms with both SATA and SCSI disk systems, and they all behave the same way.

Surely the whole point of specifying that flag is that I want the data to be written to disk, together with it's corresponding metadata, so if a catastrophic failure occurs, the data is available to me on restart.

My researches of newsgroup articles on this subject suggest that if I want this, then I also have to use FileFlushBuffers(), but if that is so then surely I don't need to specify the write through flag since all file and folder data will be committed to disk when I call that function.

So, what is the point of specifying the write-through flag in this scenario?

I think the documentation of CreateFile() could do with a little clarification!

Thanks in advance for responses.

Nick

RE: CreateFile() and FILE_FLAG_WRITE_THROUGH