Under what conditions could a ReadFile or WriteFile fail to transfer all of the bytes, and how do I detect that?

devblogs.microsoft.com/oldnewthing/20250808-00/?p=111447

August 8, 2025

A customer wanted to understand the conditions under which the `ReadFile` and `WriteFile` functions would fail to transfer all of the bytes, and how to detect that this has occurred.

The obvious reason why the `ReadFile` functions would fail to transfer all the bytes is if there aren't that many bytes to read. For a disk file, this typically happens because you are reading past the end of the file. You can also get this for other types of file handles: For a pipe in nonblocking mode, there may not be enough bytes in the pipe. Or you might have a message pipe, and the message is smaller than the size of your buffer. Or you might be accessing a device, and the device doesn't have all the bytes available.

Similarly, the obvious reason why the `WriteFile` function would fail to transfer all the bytes is if there isn't enough room for all the bytes. For a disk file, the disk might be full, or you have reached your disk quota. For a pipe in nonblocking mode, a write may be short if there is not enough buffer space in the pipe to hold all the requested data. In all of these cases, you can detect the short write by checking whether the actual number of bytes written is less than the number of bytes requested.

If the number of bytes actually transferred is nonzero, then the `ReadFile` and `WriteFile` functions will return success, but the actual number of bytes transferred will be less than the number of bytes requested.