Superstition: Why is GetFileAttributes the way old-timers test file existence?

devblogs.microsoft.com/oldnewthing/20071023-00

October 23, 2007

Raymond Chen

If you ask an old-timer how to test for file existence, they'll say, "Use `GetFileAttributes`." This is still probably the quickest way to test for file existence, since it requires only a single call. Other methods such as `FindFirstFile` or `CreateFile` require a separate `FindClose` or `CloseHandle` call, which triggers another network round-trip, which adds to the cost. But back in the old days, the preference for `GetFileAttributes` wasn't just a performance tweak. If you tried to open the file to see if it existed, you could get the wrong answer! Some network providers had a feature called a "data path". You can add directories to the data path, and if any attempt to open a file failed, the network provider would try again in all the directories in the data path. For example, suppose your data path was `\\server1\backups\dir1;\\server1\backups\dir2` and you tried to open the file `File.txt`. If the file `File.txt` could not be found in the current directory, the network provider would try again, looking for `\\server1\backups\dir1\File.txt`, and if it couldn't find that file, it would try again with `\\server1\backups\dir2\File.txt`. All this extra path searching happened behind Windows's back. Windows had no idea it was happening. All it knew is that it issued an open call, and the open succeeded. As a result, if you used the "open a file to see if it exists" algorithm, you would get the wrong result for any file that existed on the data path but not in the current directory. These network providers didn't search the data path in response to `GetFileAttributes`, however, so a call to `GetFileAttributes` would fail if the file didn't exist in the current directory, regardless of whether it existed on the data path.

I don't know if any network providers still implement this data path feature, but Windows code in general sticks to `GetFileAttributes`, because it's better to be safe than sorry.

Raymond Chen

Follow