

Can we continue to call `FindNextFile()` with the same handle that returned an error last time?

devblogs.microsoft.com/oldnewthing/20140612-00

June 12, 2014



Raymond Chen

A customer wanted to know whether it was okay to call `FindNextFile` with the same handle that returned an error last time. In other words, consider the following sequence of events:

1. `h = FindFirstFile(...)`; succeeds
2. `FindNextFile(h, ...)`; fails
3. `FindNextFile(h, ...)`; ??? profit ???

The customer elaborated:

Suppose that the directory contains four files, A, B, C, and D. We expect the following:

- `FindFirstFile` returns A
- `FindNextFile` returns B
- `FindNextFile` fails (C is selected but an error occurred)
- `FindNextFile` returns D ← is this expected?

After `FindNextFile` returns an error, can we continue to search with the same handle? Or should we close the handle and get a new one from `FindFirstFile`? If it depends on the type of error that occurred, the customer would like to know more details about when the search can be continued and when it cannot.

We asked the customer what problem they're encountering that is causing them to ask this strange question. The customer replied, "Sometimes we get the error `ERROR_FILE_CORRUPT` or `ERROR_INVALID_FUNCTION`, but we don't know what end-user configurations are causing those error codes. We would like to know whether we can continue to use `FindNextFile` in these two cases." The assumption "C is selected by an error occurred" is already wrong. The error might not have selected C. The error may have failed before C was selected. (For example, maybe the network cable was unplugged, so the server doesn't even know that we tried to select C.) Or the error may result in C and D both being skipped. Since an error occurred, any of these things may have happened. There is no value in trying to continue to

use a find handle that is in an error state because you cannot reason about it. Maybe it got stuck in a permanent error state (the user removed the USB drive). Maybe it is a transient error state (somebody finds the network cable and plugs it back in). It's like asking, "If something is broken, can I expect it to continue working?" Even closing the handle and restarting the enumeration may not succeed. For example, as long as the drive or network cable remains unplugged, your enumeration will fail. And it might be a repeatable error state due to drive corruption which causes enumerations always to fail at the same place. (My guess is that `ERROR_FILE_CORRUPT` is the case of drive corruption, and `ERROR_INVALID_FUNCTION` is some sort of device driver error state, perhaps because the device was unplugged.)

You should just accept that you cannot enumerate the contents and proceed accordingly.

Raymond Chen

Follow

