Why is the System.DateCreated property off by a few seconds?

devblogs.microsoft.com/oldnewthing/20150408-00

April 8, 2015

Raymond Chen

If you ask for the `System.DateCreated` property of a shell item, the timestamp that comes back is up to two seconds different from the file's actual timestamp. (Similarly for `System.DateModified` and `System.DateAccessed`.) Why is that?

This is an artifact of a decision taken in 1993.

In general, shell namespace providers cache information in the ID list at the time the ID list is created so that querying basic properties from an item can be done without accessing the underlying medium.

In 1993, saving 4KB of memory had a measurable impact on system performance. Therefore, bytes were scrimped and saved, and one place where *four whole bytes* were squeezed out was in the encoding of file timestamps in ID lists. Instead of using the 8-byte `FILETIME` structure, the shell used the 4-byte DOS date-time format. Since the shell created thousands of ID lists, a four-byte savings multiplied over thousands of items comes out to several kilobytes of data.

But one of the limitations of the DOS date-time format is that it records time in two-second increments, so any timestamp recorded in DOS date-time format can be up to two seconds away from its actual value. (The value is always truncated rather than rounded in order to avoid problems with timestamps from the future.) Since Windows 95 used FAT as its native file system, and FAT uses the DOS date-time format, this rounding never created any problems in practice, since all the file timestamps were already pre-truncated to 2-second intervals.

Of course, Windows NT uses NTFS as the native file system, and NTFS records file times to 100-nanosecond precision. (Though the accuracy is significantly less.) But too late. The ID list format has already been decided, and since ID lists can be saved to a file and transported to another computer (*e.g.* in the form of a shortcut file), the binary format cannot be tampered with. Hooray for compatibility.

Bonus chatter: In theory, the ID list format could be extended in a backward-compatible way, so that every ID list contained two timestamps, a compatible version (2-second precision) and a new version (100-nanosecond precision). So far, there has not been significant demand for more accurate timestamps inside of ID lists.

Raymond Chen

Follow