How to recognize different types of sentinel timestamps from quite a long way away

devblogs.microsoft.com/oldnewthing/20051028-29

October 28, 2005

Raymond Chen

Some time ago, I discussed several timestamp formats you might run into. Today we'll take a logical step from that information and develop a list of special values you might encounter. Note that if you apply time zone adjustments, the actual timestamp may shift by up to a day.

Date	Interpretation
January 1, 0001	The value 0 as a CLR System.DateTime.
January 1, 1601	The value 0 as a Win32 FILETIME.
December 29/30, 1899	The value -1 or 0 as an OLE automation date.
December 13, 1901	The value 0x80000000 as a time_t.
December 31, 1969 January 1, 1970	The value -1 or 0 as a time_t.
January 1, 1980	The beginning of the DOS date/time era. (Unlikely to be encountered since 0 is not a valid DOS date/time value.)
January 19, 2038	The value 0x7FFFFFFF as a time_t.
February 7, 2106	The value 0xFFFFFFFF as a time_t.
September 14, 30828	The value 0x7FFFFFFF`FFFFFFFF as a FILETIME.

All of these special values have one thing in common: If you see them, it's probably a bug. Typically they will arise when somebody fails to do proper error checking and ends up treating an error code as if it were a valid return value. (The special values 0, -1, and 0xFFFFFFFF are often used as error codes.)

Raymond Chen

Follow