# When does GetTickCount consider the system to have started?

devblogs.microsoft.com/oldnewthing/20141113-00

November 13, 2014

Raymond Chen

The `GetTickCount` and `GetTickCount64` functions return "the number of milliseconds that have elapsed since the system was started." (The 32-bit version wraps around after around 50 days.) But when exactly is the system considered to have started? Is it when power is applied to the computer? When the BIOS completes POST? When the user picks the operating system from the boot menu? When the kernel switches to protected mode? It isn't defined exactly when the timer starts. Because that's not its purpose. The purpose of `GetTickCount` is to let you measure intervals of time. It provides a common clock source so that multiple components can coordinate their actions. It also allows you to retrieve the tick count at one point, then retrieve the tick count at another point, subtract them, and conclude how much time has elapsed between those two points. The absolute value of the tick count is not meaningful. The only way to extract meaning from it is to subtract it from another tick count to get the delta. In fact, on debugging builds of Windows, the kernel artificially sets the `GetTickCount` counter to "one hour before 32-bit timer tick rollover"; it effectively backdates the boot time by around 50 days. This is done to help identify bugs related to timer tick rollover. If your goal is to measure operating system boot time from the application of power to the computer, then `GetTickCount` is not going to be useful. After all, Windows isn't even running at the moment you apply power to the computer. The BIOS does its work without any operating system all, so Windows has no idea how long the BIOS took to POST. The text in MSDN could be a bit more explicit and say "elapsed since Windows started", or it could be pointlessly nitpicky and say "elapsed since the Windows HAL initialized the programmable interval timer." Better would be if it simply described how the timer is intended to be used. " `GetTickCount` returns a value which increases at a rate of 1000 per second." Perhaps with some clarifying text: "By convention, the zero point of the `GetTickCount` counter is the approximate time the system booted. Note, however, that this convention is violated on occasion (such as on a checked build of Windows), so applications should not ascribe any meaning to the zero point of the tick counter."

If you want to know how much time elapsed since the application of power, you need to use a stopwatch.

Raymond Chen

**Follow**