Now that computers have more than 4MB of memory, can we get seconds on the taskbar?

devblogs.microsoft.com/oldnewthing/20220411-00

April 11, 2022

Raymond Chen

The clock in the Windows taskbar does not display seconds. Originally, this was due to the performance impact on a 4MB system of having to keep in memory the code responsible for calculating the time and drawing it. But computers nowadays have lots more than 4MB of memory, so why not bring back the seconds?

Although it's true that computers nowadays have a lot more than 4MB of memory, bringing back seconds is still not a great idea for performance.

On multi-users systems, like Terminal Server servers, it's not one taskbar clock that would update once a second. Rather, each user that signs in has their own taskbar clock, that would need to update every second. So once a second, a hundred stacks would get paged in so that a hundred taskbar clocks can repaint. This is generally not a great thing, since it basically means that the system is spending all of its CPU updating clocks.

This is the same reason why, on Terminal Server systems, caret blinking is typically disabled. Blinking a caret at 500ms across a hundred users turns into a lot of wasted CPU. Even updating a hundred clocks once a minute is too much for many systems, and most Terminal Server administrators just disable the taskbar clock entirely.

Okay, but what about systems that aren't Terminal Server servers? Why can't my little single-user system show seconds on the clock?

The answer is still performance.

Any periodic activity with a rate faster than one minute incurs the scrutiny of the Windows performance team, because periodic activity prevents the CPU from entering a low-power state. Updating the seconds in the taskbar clock is not essential to the user interface, unlike telling the user where their typing is going to go, or making sure a video plays smoothly. And the recommendation is that inessential periodic timers have a minimum period of one minute, and they should enable timer coalescing to minimize system wake-ups.

Raymond Chen

Follow