What is the maximum number of timers a program can create?

devblogs.microsoft.com/oldnewthing/20090827-00

August 27, 2009

Raymond Chen

As MSDN correctly notes, "Timers are a limited global resource." How limited are they? (We're talking about timers created by `SetTimer`.) Not as limited as they once were. Originally, there were eight timers total for the entire system. When there are only eight timers, you have to be very careful not to waste them and to destroy them when they aren't needed. Windows 3.0 increased this to 32. Windows 95 increased this to around 2500, although the debug version of Windows got mad at you if a single program created more than 32. Windows NT 3.1 would create timers until you exhausted the desktop heap. The maximum number of timers depends on what other window manager objects had been created. Starting with Windows NT 4, the window manager treats timers as User Objects for bookkeeping purposes, so all your timers, windows, menus, and so on fight for space in the per-process 10,000 object limit and the 32,000-ish object limit for a desktop. As you can see, the precise limit changes over time, so you couldn't count on any particular limit being in effect. Who knows, maybe the limit will change again in a future version of Windows.

These limits are pretty high, but that doesn't mean you can be lazy. Jessica pointed out that the WinForms Timer component is easily leaked "because most people don't know that they have to call Dispose on it." Another thing to check for when you find that you can't create any more timers.

Raymond Chen

Follow