

Why is the limit of window handles per process 10,000?

 devblogs.microsoft.com/oldnewthing/20070718-00

July 18, 2007



Raymond Chen

If your program runs haywire, you will find that it manages to create about 10,000 window manager objects and then the system won't let it have any more. Why stop at 10,000? The first answer is "If you have to ask, you're probably doing something wrong." Programs shouldn't be creating anywhere near ten thousands window manager objects in the first place.

Furthermore, as we saw last time, the maximum number of window manager objects that can be created is around 32,700. Giving a program 10,000 is already a third of the total amount available. That's already pretty darned generous, if you ask me. Preventing a program from running away and consuming all of the window manager objects is an attempt to contain the damage caused by a runaway program. Even if a program goes haywire, there's still around 20,000 objects available for the other programs to use. One of them might even be Task Manager, which the user fired up in order to kill the runaway program.

[Raymond Chen](#)

Follow

