

# Why can't you trap TerminateProcess?

---

 [devblogs.microsoft.com/oldnewthing/20040722-00](http://devblogs.microsoft.com/oldnewthing/20040722-00)

July 22, 2004



Raymond Chen

If a user fires up Task Manager and clicks “End Task” on your program, Windows first tries to shut down your program nicely, by sending WM\_CLOSE messages to GUI programs and CTRL\_CLOSE\_EVENT events to console programs. But you don't get a chance to intercept TerminateProcess. Why not? TerminateProcess is the low-level process killing function. It bypasses DLL\_PROCESS\_DETACH and anything else in the process. Once you kill with TerminateProcess, no more user-mode code will run in that process. It's gone. Do not pass go. Do not collect \$200. If you could intercept TerminateProcess, then you would be escalating the arms race between programs and users. Suppose you could intercept it. Well, then if you wanted to make your program unkillable, you would just hang in your TerminateProcess handler! And then people would ask for “a way to kill a process that is refusing to be killed with TerminateProcess,” and we'd be back to where we started.

Tomorrow: About those processes that don't go away even though you've killed them. They're really dead, but they won't go away.

Raymond Chen

**Follow**

