

What happens if I call KillTimer with a NULL hwnd?

 devblogs.microsoft.com/oldnewthing/20120531-00

May 31, 2012



Raymond Chen

A customer discovered a bug in their code and wanted some information on how serious it was, so they could assess how urgently they need to issue a fix.

We have code that calls `SetTimer` with a valid window handle, but then we destroy the window before we get around to calling `KillTimer`. When we finally do call `KillTimer`, we do so with a `NULL` window handle. The `KillTimer` calls are probably harmless, but are we leaking the timers?

The customer's real concern was actually in the part of the problem they thought was a point of little concern. The window manager cleans up orphaned timers when the associated window is destroyed, so there is no timer leak in this case. Of course, it's still good practice to clean up those timers. (Note however that [a similar situation does lead to leaked timers](#).) The real danger is in the `KillTimer` call. By passing a null window handle, you are killing a thread timer. Maybe you're lucky and there is no thread timer whose ID is the value you passed as the second parameter, but someday your luck will run out and you will accidentally kill somebody else's timer.

The customer was pleased with this explanation.

That's exactly the information we were looking for. Thanks.

Raymond Chen

Follow

