Eventually the window manager simply says that enough is enough

devblogs.microsoft.com/oldnewthing/20120607-00

June 7, 2012

Raymond Chen

Many window manager operations are recursive, and eventually the window manager will simply say that enough is enough when the recursion goes too deep. We've seen this when you nest windows more than 50 levels deep or nest menus more than 25 levels deep, for example. (Note also that these limits may change in the future, so don't rely on being able to walk right up to the edge. Those values came from 32-bit Windows XP; I don't know if the limits have been dropped even further in subsequent versions of Windows, and I'm not sufficiently motivated to find out.)

A customer had some code which installed a message hook, and they found that the message hook was not called consistently. They tracked it down to another component in their application, and that component also installed a message hook. The contact actually came from the developer who maintained the other component: Did I write my message hook incorrectly? Am I accidentally messing up other message hooks in the system?

The developer included their hook-management code, and it didn't look obviously wrong. All code paths eventually called `CallNextHookEx`, so there shouldn't be any hook-loss.

The customer was kind enough to include a copy of their program with instructions on how to trigger the problem, and stepping through the hook code quickly revealed the source of the problem. Maybe you can see it too. Here's a stack trace when the end of the hook chain is reached:

```
ChildEBP RetAddr
0011cdc4 16846e6e contoso!ContosoWindowEventHook+0x11b
0011cdf8 768231eb user32!DispatchHookA+0x104
0011ce38 76824260 user32!CallHookWithSEH+0x21
0011ce6c 773e642e user32!__fnHkINLPMSG+0x71
0011ceb0 768447aa ntdll!KiUserCallbackDispatcher+0x2e
0011ceb4 76844787 user32!NtUserCallNextHookEx+0xc
0011ced8 6fbdb1e0 user32!CallNextHookEx+0x71
0011cf14 16846e6e contoso!ContosoWindowEventHook+0x11b
0011cf48 768231eb user32!DispatchHookA+0x104
0011cf88 76824260 user32!CallHookWithSEH+0x21
0011cfbc 773e642e user32!__fnHkINLPMSG+0x71
0011d000 768447aa ntdll!KiUserCallbackDispatcher+0x2e
0011d004 76844787 user32!NtUserCallNextHookEx+0xc
0011d028 6fbdb1e0 user32!CallNextHookEx+0x71
0011d064 16846e6e contoso!ContosoWindowEventHook+0x11b
0011d098 768231eb user32!DispatchHookA+0x104
0011d0d8 76824260 user32!CallHookWithSEH+0x21
0011d10c 773e642e user32!__fnHkINLPMSG+0x71
0011d150 768447aa ntdll!KiUserCallbackDispatcher+0x2e
0011d154 76844787 user32!NtUserCallNextHookEx+0xc
0011d178 6fbdb1e0 user32!CallNextHookEx+0x71
0011d1b4 16846e6e contoso!ContosoWindowEventHook+0x11b
0011d1e8 768231eb user32!DispatchHookA+0x104
0011d228 76824260 user32!CallHookWithSEH+0x21
0011d25c 773e642e user32!__fnHkINLPMSG+0x71
0011d2a0 768447aa ntdll!KiUserCallbackDispatcher+0x2e
0011d2a4 76844787 user32!NtUserCallNextHookEx+0xc
0011d2c8 6fbdb1e0 user32!CallNextHookEx+0x71
0011d304 16846e6e contoso!ContosoWindowEventHook+0x11b
0011d338 768231eb user32!DispatchHookA+0x104
0011d378 76824260 user32!CallHookWithSEH+0x21
0011d3ac 773e642e user32!__fnHkINLPMSG+0x71
0011d3f0 768447aa ntdll!KiUserCallbackDispatcher+0x2e
0011d3f4 76844787 user32!NtUserCallNextHookEx+0xc
0011d418 6fbdb1e0 user32!CallNextHookEx+0x71
0011d454 16846e6e contoso!ContosoWindowEventHook+0x11b
0011d488 768231eb user32!DispatchHookA+0x104
0011d4c8 76824260 user32!CallHookWithSEH+0x21
0011d4fc 773e642e user32!__fnHkINLPMSG+0x71
0011d540 768447aa ntdll!KiUserCallbackDispatcher+0x2e
0011d544 76844787 user32!NtUserCallNextHookEx+0xc
0011d568 6fbdb1e0 user32!CallNextHookEx+0x71
0011d5a4 16846e6e contoso!ContosoWindowEventHook+0x11b
0011d5d8 768231eb user32!DispatchHookA+0x104
0011d618 76824260 user32!CallHookWithSEH+0x21
0011d64c 773e642e user32!__fnHkINLPMSG+0x71
0011d690 768447aa ntdll!KiUserCallbackDispatcher+0x2e
0011d694 76844787 user32!NtUserCallNextHookEx+0xc
0011d6b8 6fbdb1e0 user32!CallNextHookEx+0x71
0011d6f4 16846e6e contoso!ContosoWindowEventHook+0x11b
0011d728 768231eb user32!DispatchHookA+0x104
```

```
0011d768 76824260 user32!CallHookWithSEH+0x21
0011d79c 773e642e user32!__fnHkINLPMSG+0x71
0011d7e0 768447aa ntdll!KiUserCallbackDispatcher+0x2e
0011d7e4 76844787 user32!NtUserCallNextHookEx+0xc
0011d808 6fbdb1e0 user32!CallNextHookEx+0x71
0011d844 16846e6e contoso!ContosoWindowEventHook+0x11b
0011d878 768231eb user32!DispatchHookA+0x104
0011d8b8 76824260 user32!CallHookWithSEH+0x21
0011d8ec 773e642e user32!__fnHkINLPMSG+0x71
0011d930 768447aa ntdll!KiUserCallbackDispatcher+0x2e
0011d934 76844787 user32!NtUserCallNextHookEx+0xc
0011d958 6fbdb1e0 user32!CallNextHookEx+0x71
0011d994 16846e6e contoso!ContosoWindowEventHook+0x11b
0011d9c8 768231eb user32!DispatchHookA+0x104
0011da08 76824260 user32!CallHookWithSEH+0x21
0011da3c 773e642e user32!__fnHkINLPMSG+0x71
0011da80 768447aa ntdll!KiUserCallbackDispatcher+0x2e
0011da84 76844787 user32!NtUserCallNextHookEx+0xc
0011daa8 6fbdb1e0 user32!CallNextHookEx+0x71
0011dae4 16846e6e contoso!ContosoWindowEventHook+0x11b
0011db18 768231eb user32!DispatchHookA+0x104
0011db58 76824260 user32!CallHookWithSEH+0x21
0011db8c 773e642e user32!__fnHkINLPMSG+0x71
0011dbd0 768447aa ntdll!KiUserCallbackDispatcher+0x2e
0011dbd4 76844787 user32!NtUserCallNextHookEx+0xc
0011dbf8 6fbdb1e0 user32!CallNextHookEx+0x71
0011dc34 16846e6e contoso!ContosoWindowEventHook+0x11b
0011dc68 768231eb user32!DispatchHookA+0x104
0011dca8 76824260 user32!CallHookWithSEH+0x21
0011dcdc 773e642e user32!__fnHkINLPMSG+0x71
0011dd20 768447aa ntdll!KiUserCallbackDispatcher+0x2e
0011dd24 76844787 user32!NtUserCallNextHookEx+0xc
0011dd48 6fbdb1e0 user32!CallNextHookEx+0x71
0011dd84 16846e6e contoso!ContosoWindowEventHook+0x11b
0011ddb8 768231eb user32!DispatchHookA+0x104
0011ddf8 76824260 user32!CallHookWithSEH+0x21
0011de2c 773e642e user32!__fnHkINLPMSG+0x71
0011de70 768447aa ntdll!KiUserCallbackDispatcher+0x2e
0011de74 76844787 user32!NtUserCallNextHookEx+0xc
0011de98 6fbdb1e0 user32!CallNextHookEx+0x71
0011ded4 16846e6e contoso!ContosoWindowEventHook+0x11b
0011df08 768231eb user32!DispatchHookA+0x104
0011df48 76824260 user32!CallHookWithSEH+0x21
0011df7c 773e642e user32!__fnHkINLPMSG+0x71
0011dfc0 768447aa ntdll!KiUserCallbackDispatcher+0x2e
0011dfc4 76844787 user32!NtUserCallNextHookEx+0xc
0011dfe8 6fbdb1e0 user32!CallNextHookEx+0x71
0011e024 16846e6e contoso!ContosoWindowEventHook+0x11b
0011e058 768231eb user32!DispatchHookA+0x104
0011e098 76824260 user32!CallHookWithSEH+0x21
0011e0cc 773e642e user32!__fnHkINLPMSG+0x71
0011e110 768447aa ntdll!KiUserCallbackDispatcher+0x2e
```

```
0011e114 76844787 user32!NtUserCallNextHookEx+0xc
0011e138 6fbdb1e0 user32!CallNextHookEx+0x71
0011e174 16846e6e contoso!ContosoWindowEventHook+0x11b
0011e1a8 768231eb user32!DispatchHookA+0x104
0011e1e8 76824260 user32!CallHookWithSEH+0x21
0011e21c 773e642e user32!__fnHkINLPMSG+0x71
0011e260 768447aa ntdll!KiUserCallbackDispatcher+0x2e
0011e264 76844787 user32!NtUserCallNextHookEx+0xc
0011e288 6fbdb1e0 user32!CallNextHookEx+0x71
0011e2c4 16846e6e contoso!ContosoWindowEventHook+0x11b
0011e2f8 768231eb user32!DispatchHookA+0x104
0011e338 76824260 user32!CallHookWithSEH+0x21
0011e36c 773e642e user32!__fnHkINLPMSG+0x71
0011e3b0 768447aa ntdll!KiUserCallbackDispatcher+0x2e
0011e3b4 76844787 user32!NtUserCallNextHookEx+0xc
0011e3d8 6fbdb1e0 user32!CallNextHookEx+0x71
0011e414 16846e6e contoso!ContosoWindowEventHook+0x11b
0011e448 768231eb user32!DispatchHookA+0x104
0011e488 76824260 user32!CallHookWithSEH+0x21
0011e4bc 773e642e user32!__fnHkINLPMSG+0x71
0011e500 768447aa ntdll!KiUserCallbackDispatcher+0x2e
0011e504 76844787 user32!NtUserCallNextHookEx+0xc
0011e528 6fbdb1e0 user32!CallNextHookEx+0x71
0011e564 16846e6e contoso!ContosoWindowEventHook+0x11b
0011e598 768231eb user32!DispatchHookA+0x104
0011e5d8 76824260 user32!CallHookWithSEH+0x21
0011e60c 773e642e user32!__fnHkINLPMSG+0x71
0011e650 768447aa ntdll!KiUserCallbackDispatcher+0x2e
0011e654 76844787 user32!NtUserCallNextHookEx+0xc
0011e678 6fbdb1e0 user32!CallNextHookEx+0x71
0011e6b4 16846e6e contoso!ContosoWindowEventHook+0x11b
0011e6e8 768231eb user32!DispatchHookA+0x104
0011e728 76824260 user32!CallHookWithSEH+0x21
0011e75c 773e642e user32!__fnHkINLPMSG+0x71
0011e7a0 768447aa ntdll!KiUserCallbackDispatcher+0x2e
0011e7a4 76844787 user32!NtUserCallNextHookEx+0xc
0011e7c8 6fbdb1e0 user32!CallNextHookEx+0x71
0011e804 16846e6e contoso!ContosoWindowEventHook+0x11b
0011e838 768231eb user32!DispatchHookA+0x104
0011e878 76824260 user32!CallHookWithSEH+0x21
0011e8ac 773e642e user32!__fnHkINLPMSG+0x71
0011e8f0 768447aa ntdll!KiUserCallbackDispatcher+0x2e
0011e8f4 76844787 user32!NtUserCallNextHookEx+0xc
0011e918 6fbdb1e0 user32!CallNextHookEx+0x71
0011e954 16846e6e contoso!ContosoWindowEventHook+0x11b
0011e988 768231eb user32!DispatchHookA+0x104
0011e9c8 76824260 user32!CallHookWithSEH+0x21
0011e9fc 773e642e user32!__fnHkINLPMSG+0x71
0011ea40 768447aa ntdll!KiUserCallbackDispatcher+0x2e
0011ea44 76844787 user32!NtUserCallNextHookEx+0xc
0011ea68 6fbdb1e0 user32!CallNextHookEx+0x71
0011eaa4 16846e6e contoso!ContosoWindowEventHook+0x11b
```

```
0011ead8 768231eb user32!DispatchHookA+0x104
0011eb18 76824260 user32!CallHookWithSEH+0x21
0011eb4c 773e642e user32!__fnHkINLPMSG+0x71
0011eb90 768447aa ntdll!KiUserCallbackDispatcher+0x2e
0011eb94 76844787 user32!NtUserCallNextHookEx+0xc
0011ebb8 6fbdb1e0 user32!CallNextHookEx+0x71
0011ebf4 16846e6e contoso!ContosoWindowEventHook+0x11b
0011ec28 768231eb user32!DispatchHookA+0x104
0011ec68 76824260 user32!CallHookWithSEH+0x21
0011ec9c 773e642e user32!__fnHkINLPMSG+0x71
0011ece0 768447aa ntdll!KiUserCallbackDispatcher+0x2e
0011ece4 76844787 user32!NtUserCallNextHookEx+0xc
0011ed08 6fbdb1e0 user32!CallNextHookEx+0x71
0011ed44 16846e6e contoso!ContosoWindowEventHook+0x11b
0011ed78 768231eb user32!DispatchHookA+0x104
0011edb8 76824260 user32!CallHookWithSEH+0x21
0011edec 773e642e user32!__fnHkINLPMSG+0x71
0011ee30 768447aa ntdll!KiUserCallbackDispatcher+0x2e
0011ee34 76844787 user32!NtUserCallNextHookEx+0xc
0011ee58 6fbdb1e0 user32!CallNextHookEx+0x71
0011ee94 16846e6e contoso!ContosoWindowEventHook+0x11b
0011eec8 768231eb user32!DispatchHookA+0x104
0011ef08 76824260 user32!CallHookWithSEH+0x21
0011ef3c 773e642e user32!__fnHkINLPMSG+0x71
0011ef80 768447aa ntdll!KiUserCallbackDispatcher+0x2e
0011ef84 76844787 user32!NtUserCallNextHookEx+0xc
0011efa8 6fbdb1e0 user32!CallNextHookEx+0x71
0011efe4 16846e6e contoso!ContosoWindowEventHook+0x11b
0011f018 768231eb user32!DispatchHookA+0x104
0011f058 76824260 user32!CallHookWithSEH+0x21
0011f08c 773e642e user32!__fnHkINLPMSG+0x71
0011f0d0 768447aa ntdll!KiUserCallbackDispatcher+0x2e
0011f0d4 76844787 user32!NtUserCallNextHookEx+0xc
0011f0f8 6fbdb1e0 user32!CallNextHookEx+0x71
0011f134 16846e6e contoso!ContosoWindowEventHook+0x11b
0011f168 768231eb user32!DispatchHookA+0x104
0011f1a8 76824260 user32!CallHookWithSEH+0x21
0011f1dc 773e642e user32!__fnHkINLPMSG+0x71
0011f220 768447aa ntdll!KiUserCallbackDispatcher+0x2e
0011f224 76844787 user32!NtUserCallNextHookEx+0xc
0011f248 6fbdb1e0 user32!CallNextHookEx+0x71
0011f284 16846e6e contoso!ContosoWindowEventHook+0x11b
0011f2b8 768231eb user32!DispatchHookA+0x104
0011f2f8 76824260 user32!CallHookWithSEH+0x21
0011f32c 773e642e user32!__fnHkINLPMSG+0x71
0011f370 768447aa ntdll!KiUserCallbackDispatcher+0x2e
0011f374 76844787 user32!NtUserCallNextHookEx+0xc
0011f398 6fbdb1e0 user32!CallNextHookEx+0x71
0011f3d4 16846e6e contoso!ContosoWindowEventHook+0x11b
0011f408 768231eb user32!DispatchHookA+0x104
0011f448 76824260 user32!CallHookWithSEH+0x21
0011f47c 773e642e user32!__fnHkINLPMSG+0x71
```

```
0011f4c0 768447aa ntdll!KiUserCallbackDispatcher+0x2e
0011f4c4 76844787 user32!NtUserCallNextHookEx+0xc
0011f4e8 6fbdb1e0 user32!CallNextHookEx+0x71
0011f524 16846e6e contoso!ContosoWindowEventHook+0x11b
0011f558 768231eb user32!DispatchHookA+0x104
0011f598 76824260 user32!CallHookWithSEH+0x21
0011f5cc 773e642e user32!__fnHkINLPMSG+0x71
0011f610 768447aa ntdll!KiUserCallbackDispatcher+0x2e
0011f614 76844787 user32!NtUserCallNextHookEx+0xc
0011f638 6fbdb1e0 user32!CallNextHookEx+0x71
0011f674 16846e6e contoso!ContosoWindowEventHook+0x11b
0011f6a8 768231eb user32!DispatchHookA+0x104
0011f6e8 76824260 user32!CallHookWithSEH+0x21
0011f71c 773e642e user32!__fnHkINLPMSG+0x71
0011f760 768447aa ntdll!KiUserCallbackDispatcher+0x2e
0011f764 76844787 user32!NtUserCallNextHookEx+0xc
0011f788 6fbdb1e0 user32!CallNextHookEx+0x71
0011f7c4 16846e6e contoso!ContosoWindowEventHook+0x11b
0011f7f8 768231eb user32!DispatchHookA+0x104
0011f838 76824260 user32!CallHookWithSEH+0x21
0011f86c 773e642e user32!__fnHkINLPMSG+0x71
0011f8b0 768447aa ntdll!KiUserCallbackDispatcher+0x2e
0011f8b4 76844787 user32!NtUserCallNextHookEx+0xc
0011f8d8 6fbdb1e0 user32!CallNextHookEx+0x71
0011f914 16846e6e contoso!ContosoWindowEventHook+0x11b
0011f948 768231eb user32!DispatchHookA+0x104
0011f988 76824260 user32!CallHookWithSEH+0x21
0011f9bc 773e642e user32!__fnHkINLPMSG+0x71
0011f9d0 00030000 ntdll!KiUserCallbackDispatcher+0x2e
0011fa28 6fbdb1e0 0x30000
0011fa64 16846e6e contoso!ContosoWindowEventHook+0x11b
0011fa98 768231eb user32!DispatchHookA+0x104
0011fad8 76824260 user32!CallHookWithSEH+0x21
0011fb0c 773e642e user32!__fnHkINLPMSG+0x71
0011fb20 00030000 ntdll!KiUserCallbackDispatcher+0x2e
0011fb7c 768292a9 0x30000
0011fba8 6b2ce010 user32!PeekMessageW+0xfb
```

As you can see, a *third* component in their application installed at least *thirty-five hooks*. After the thirty-fifth hook, the window manager stepped in and said, "That's it, I'm cutting you off."

Now, the limit isn't actually thirty-five. The window manager keeps dispatching hooks until the kernel stack starts running low, and then it gives up. This happens with a lot of recursive algorithms: The window manager plays the game for a while, but when it looks like you're about to bluescreen, it stops short and says, "Okay, I'm not going to do that any more."

The developers now got to take their problem to the developer responsible for the Contoso component, and figure out why it's installing so many hooks. Maybe that component could try to consolidate identical hooks. Or maybe it's a leak. They never did report back (not that I

was expecting them to).

Bonus chatter: Why is hook dispatch done recursively? Shouldn't it be done iteratively?

Remember that windows hooks came from 16-bit Windows, where economy was paramount. And the existing `CallNextHook` pattern was preserved, though it changed to `CallNextHookEx`, where you pass the hook handle directly instead of its address. One advantage of the `CallNextHookEx` model over an iterative model is that explicitly forwarding to the previous hook lets you do work on the back end. I.e., you can forward the call down the chain, and then do something when control returns. This is the sort of thing you probably use a lot when you subclass a window or override a method in a derived class and call the base class from your override.

Raymond Chen

Follow