

Win32 user interface work is inherently single-threaded

 devblogs.microsoft.com/oldnewthing/20071018-00

October 18, 2007



Raymond Chen

At the end of the day, there's a window handle in there. As we saw a few years ago, windows have thread affinity. You can push work to a background thread all you want, but once you send a message to the window (or do something that results in a message being sent to the window), you've given control to the UI thread. Since the shell is all about user interfaces, the shell naturally expects that all threads that use it are single-threaded apartments.

Historically, however, you sort of got away with violating this rule because older versions of the shell used their own custom fake version of COM rather than using the official one in `OLE32.DLL`. As a result, programs that broke the rule on apartment threading often managed to get away with it because the shell assumed everybody was doing the right thing. (The Internet Explorer team faced a similar problem with browser plug-ins.) If you tried to use an object on the wrong thread, nobody would step in and try to stop you. Mind you, you might corrupt memory or crash, but that was your own fault. Nobody said this was going to be easy. As the shell gradually switched to using real COM instead of fake COM, and as new features were added to the shell which relied more and more heavily on callers following the rules for apartment threading, programs that had skirted the rules started running into problems. If you got away with it on earlier versions of Windows and the problem was severe, there was a good chance the shell would have to do some re-architecting to allow your dodgy code to keep working.

A member of the COM team explained that COM assumes fundamentally that multi-threaded apartments are UI-free. No UI means no need to pump messages. If you initialize COM in MTA mode on a thread, you'd better not do any UI or your thread will stop responding to messages whenever COM needs to talk to another thread.

[Raymond Chen](#)

Follow

