

Who would ever write a multi-threaded GUI program?

 devblogs.microsoft.com/oldnewthing/20140617-00

June 17, 2014



Raymond Chen

During the development of Windows 95, the user interface team discovered that a component provided by another team didn't work well under multi-threaded conditions. It was documented that the `Initialize` function had to be the first call made by a thread into the component. The user interface team discovered that if one thread called `Initialize`, and then used the component, then everything worked great. But if a second thread called `Initialize`, the component crashed whenever the second thread tried to use it. The user interface team reported this bug back to the team that provided the component, and some time later, an updated version of the component was delivered. Technically, the bug was fixed. When the second thread called `Initialize`, the function now failed with `ERROR_NOT_SUPPORTED`. The user interface team went back to the team that provided the component. "It's nice that your component detects that it is being used by a multi-threaded client and fails the second thread's attempt to initialize it. But given that design, how can a multi-threaded client use your component?" The other team's reply was, "It doesn't matter. Nobody writes multi-threaded GUI programs." The user interface team had to politely reply, "Um, *we are*. The next version of Windows will be built on a multi-threaded shell." The other team said, "Oh, um, we weren't really expecting that. Hang on, we'll get back to you."

The idea that somebody might write a multi-threaded program that used their component caught them by surprise, and they had to come up with a new design of their component that supported multiple threads in a clean way. It was a lot of work, but they came through, and Windows 95 could continue with its multi-threaded shell.

[Raymond Chen](#)

Follow

