Is it legal to have a cross-process parent/child or owner/owned window relationship?

devblogs.microsoft.com/oldnewthing/20130412-00

April 12, 2013

Raymond Chen

A customer liaison asked whether it was legal to use `SetParent` to create a parent/child relationship between windows which belong to different processes. “If I remember correctly, the documentation for `SetParent` used to contain a stern warning that it is not supported, but that remark does not appear to be present any more. I have a customer who is reparenting windows between processes, and their application is experiencing intermittent instability.”

Is it technically legal to have a parent/child or owner/owned relationship between windows from different processes?

Yes, it is technically legal.

It is also technically legal to juggle chainsaws.

Creating a cross-thread parent/child or owner/owned window relationship implicitly attaches the input queues of the threads which those windows belong to, and this attachment is transitive: If one of those queues is attached to a third queue, then all three queues are attached to each other. More generally, queues of all windows related by a chain of parent/child or owner/owned or shared-thread relationships are attached to each other.

Exercise: What are the equivalence classes generated by taking the transitive closure of parent/child windows, and what would be a natural choice of class representative? What about the equivalence classes generated by the transitive closure of parent/child and owner/owned windows?

This gets even more complicated when the parent/child or owner/owned relationship crosses processes, because cross-process coordination is even harder than cross-thread coordination. Sharing variables within a process is much easier than sharing variables across processes. On top of that, some window messages are blocked between processes.

So yes, it is technically legal, but if you create a cross-process parent/child or owner/owned relationship, the consequences can be very difficult to manage. And they become near-impossible to manage if one or both of the windows involved is unaware that it is participating in a cross-process window tree. (I often see this question in the context of somebody who wants to grab a window belonging to another process and forcibly graft it into their own process. That other process was totally unprepared for its window being manipulated in this way, and things may stop working. Indeed, things will *definitely* stop working if you change that other window from a top-level window to a child window.)

The existing text was probably removed when somebody pointed out that the action is technically legal (though not recommended for beginners), and instead of trying to come up with new text that describes the situation, merely removed the text that was incorrect. The problem with coming up with new text that describes the situation is that it only leads to more questions from people who want to do it in spite of the warnings. (It’s one of those “if you don’t already know what the consequences are, then you are not smart enough to do it correctly” things. You must first become the master of the rules before you can start breaking them.)

Raymond Chen

Follow