

# What is the documentation for SetParent trying to tell me about synchronizing the UI state?

 [devblogs.microsoft.com/oldnewthing/20171122-00](http://devblogs.microsoft.com/oldnewthing/20171122-00)

November 22, 2017



Raymond Chen

The documentation for the SetParent function says

When you change the parent of a window, you should synchronize the UISTATE of both windows. For more information, see WM\_CHANGEUISTATE and WM\_UPDATEUISTATE.

What does that mean?

Recall how the WM\_UPDATEUISTATE and WM\_CHANGEUISTATE messages work. When keyboard indicators need to be shown or hidden, the control that initiates the change sends itself a `WM_UPDATEUISTATE` message. This message propagates up the window tree. When it reaches the root of the tree, the message converts to a `WM_CHANGEUISTATE` message and propagates down the tree.

There is an optimization here: If the desired keyboard state is already present, then message propagation stops, because there's no point telling people to change something to what it already is.

However, this optimization assumes that every window in the tree has the same state to begin with.

Let's see what happens if you violate this assumption.

---

A  
hideFocus=0  
hideAccel=0

---

---

---

---

B	C
hideFocus=0	hideFocus=1
hideAccel=0	hideAccel=1

Observe that keyboard indicators are shown in A and B, but they are hidden in C. This is not a valid window configuration, but let's run with it and see where it takes us.

Suppose focus is on C and you tap the **Alt** key to show keyboard accelerators.

Window C sends itself a `WM_UPDATEUISTATE` to request that keyboard accelerators be turned on. This message propagates up the tree to window A. Window A then says, "You woke me up from my nap for this? Why are you asking to show accelerators? Accelerators are already visible! Stop wasting my time." And the message processing stops.

---

A
hideFocus=0
hideAccel=0

---

---

B	C
hideFocus=0	hideFocus=1
hideAccel=0	hideAccel=1

Result: C still doesn't get any keyboard accelerators. C is a sad window. C makes an appointment with a therapist.

The remark in the documentation is trying to tell you that if you are moving windows between parents, and the new parent has a different UI state from its adopted child, then you have two problems:

1. The UI state is inconsistent within the top-level window, so your window looks weird.
2. The UI state will not be properly updated, so your window acts weird.

Okay, so you get it. It's important to get the two UI states in sync. But how do you do that? We'll start looking into that next time.

**Bonus chatter:** That section of the documentation was written by someone who understands the window manager so deeply that they assume everybody else understands it to the same degree, expecting that "synchronize the UI state" is something the reader would

comprehend immediately. (And as we'll see next time, somebody who has never actually tried to synchronize the UI states of two windows, because if they had tried, they would have realized that accomplishing the task is harder than an offhand remark would suggest.)

Raymond Chen

**Follow**

