

Who sends the initial WM_UPDATEUISTATE message?

devblogs.microsoft.com/oldnewthing/20130517-00

May 17, 2013



Raymond Chen

Last time, we looked at the confusingly-named `WM_UPDATEUISTATE` and `WM_CHANGEUISTATE` messages. But how does the whole indicator thingie get off the ground?

The default state for a window is to show all indicators. But as a special trick, the dialog manager will send a `WM_UPDATEUISTATE` message with `UIS_INITIALIZE` after the dialog has been initialized, which turns off the indicators if the last input event was a mouse event. This is its way of inferring whether the dialog box was triggered by a mouse or keyboard action and setting the initial indicators accordingly. (Note that if the user checked *Underline keyboard shortcuts and access keys*, then the dialog manager leaves the indicators enabled regardless of the last input event.)

That special `WM_UPDATEUISTATE` message is what gives dialog boxes the extra special feature of hiding the keyboard accelerators until you use the keyboard.

But notice that only the dialog manager does this. If you want this behavior in your own non-dialog windows, you will need to send the message yourself.

```
BOOL MyWindow::OnCreate(...)
{
    ... create and initialize any child windows ...

    // initialize indicators
    BOOL fAlwaysUnderline = FALSE;
    SystemParametersInfo(SPI_GETKEYBOARDUCUES, 0,
                        &fAlwaysUnderline, 0);
    if (!fAlwaysUnderline) {
        SendMessage(this->m_hwnd, WM_UPDATEUISTATE,
                    MAKEWPARAM(UIS_INITIALIZE, 0), 0);
    }
}
```

Exercise: Why is it important to create and initialize the child windows before sending the `WM_UPDATEUISTATE` message?

Exercise: Why can't the window manager do this automatically after `WM_CREATE` returns?

Exercise: Explain the behavior this customer observes.

We have a dialog box with three buttons. Sometimes the dialog displays underlines for the hotkeys, and sometimes it doesn't. I know about the feature which hides keyboard accelerators by default, but that doesn't explain why the setting gets ignored sometimes. The first time I show the dialog in my program, I get the underlines, but the second and subsequent times, I do not.

Raymond Chen

Follow

