

How do I make my accelerators apply only when used in the main window and not when the user is using a modeless dialog?

 devblogs.microsoft.com/oldnewthing/20200807-00

August 7, 2020



Raymond Chen

A customer had a main window and defined some keyboard accelerators for it. One of the accelerators calls up a modeless dialog box. The problem is that even when focus was in the modeless dialog box, the keyboard accelerators for the main window were still active. How can you disable the keyboard accelerators for the main window when focus is in the modeless dialog box?

The standard message loop goes like this:

```
while (GetMessage(&msg, nullptr, 0, 0)) {
    if (!TranslateAccelerator(hwndMain, hacc, &msg)) {
        TranslateMessage(&msg);
        DispatchMessage(&msg);
    }
}
```

For each message, we ask the `TranslateAccelerator` function to process it. The `TranslateAccelerator` function works by looking to see if the message matches any of the accelerators in the provided accelerator table `hacc`. If so, then it posts the corresponding command to the window `hwndMain`.

As written, this message loop treats all messages as eligible for conversion into an accelerator for the main window. Maybe that's what you want.

In the customer's case, they have a modeless dialog box, and in the absence of any special handling, it will still be the case that all messages (even ones destined for the modeless dialog box) are eligible for translation into accelerators for the main window.

```

while (GetMessage(&msg, nullptr, 0, 0)) {
    if (!TranslateAccelerator(hwndMain, hacc, &msg) &&
        !IsDialogMessage(hdlgModeless, &msg)) {
        TranslateMessage(&msg);
        DispatchMessage(&msg);
    }
}

```

Observe that the call to `TranslateAccelerator` is made on every message. This means that messages destined for the modeless dialog box will nevertheless be inspected for keyboard accelerators.

We can write an alternate version of `TranslateAccelerator` that translates accelerators only if the window is destined for the specified window.

```

BOOL TranslateAcceleratorForWindow(
    HWND hwnd, HACCP hacc, LPMSG pmsg)
{
    if (hwnd == pmsg->hwnd ||
        IsChild(hwnd, pmsg->hwnd)) {
        return TranslateAccelerator(hwnd, hacc, pmsg);
    } else {
        return FALSE;
    }
}

```

This does a preliminary check to see if the message is destined for the window or one of its children. Only in that case does it attempt to translate an accelerator. In this way, we limit accelerator translation to the case where the user is working with the specified window (or any of its children).

We can switch to using this helper function in our main message loop:

```

while (GetMessage(&msg, nullptr, 0, 0)) {
    if (!TranslateAcceleratorForWindow(
        hwndMain, hacc, &msg) &&
        !IsDialogMessage(hdlgModeless, &msg)) {
        TranslateMessage(&msg);
        DispatchMessage(&msg);
    }
}

```

We can go a step further and have one set of accelerators for the main window and another for the dialog box.

```
while (GetMessage(&msg, nullptr, 0, 0)) {  
    if (!TranslateAcceleratorForWindow(  
        hwndMain, hacc, &msg) &&  
        !TranslateAcceleratorForWindow(  
            hdlgModeless, haccDlg, &msg) &&  
        !IsDialogMessage(hdlgModeless, &msg)) {  
        TranslateMessage(&msg);  
        DispatchMessage(&msg);  
    }  
}
```

It turns out you knew all this already. We discussed it quite some time ago, but looking at it from the dialog box's point of view.

[Raymond Chen](#)

Follow

