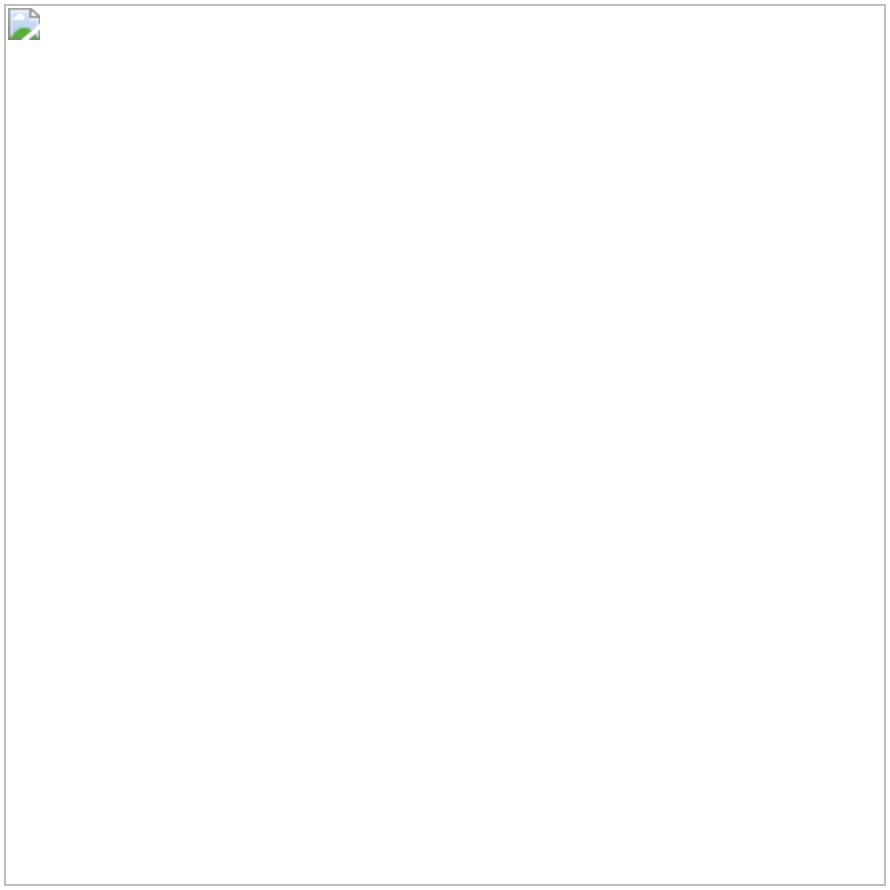
If my window hosts multiple windowless ActiveX controls, which one do I call IOleInPlaceActiveObject::TranslateAccelerator and IOleInPlaceObjectWindowless::OnWindowMessage on?

devblogs.microsoft.com/oldnewthing/20120528-00

May 28, 2012

Raymond Chen

Commenter Farproc asks how one goes about hosting two windowless ActiveX controls in a single window. In particular, "none of the documentation explains how to choose which control to send `IOleInPlaceActiveObject::TranslateAccelerator` and `IOleInPlaceObjectWindowless::OnWindowMessage` on?" Actually, the documentation does say. The documentation for `IOleInPlaceActiveObject::TranslateAccelerator` says, "Active in-place objects must always be given the first chance at translating accelerator keystrokes." So you pass the message to the active in-place object. Your window may host multiple windowless ActiveX controls, but at most one of them is the active object at a time. And most of the time, none of them will be active. For example, in Word, most of the time the insertion point is in the text part of the document. Only occasionally do you activate an in-place object by, say, double-clicking on an embedded Excel spreadsheet, at which point Excel adds its menu items to your menu bar and basically takes over your application window for a while. Here's an example of Windows 95's Wordpad hosting Paint as an in-place active object.



Source: 2.1.6 OLE/COM example: using compound documents

If you have an in-place active object, then it's the one that gets the `IOleInPlaceActive-Object::TranslateAccelerator` If, as is usually the case, you don't have an in-place active object, then nobody's `IOleInPlaceActiveObject::TranslateAccelerator` gets called because they aren't the in-place active object. (It's right there in the interface name.) For `IOleInPlaceObjectWindowless::OnWindowMessage`, the documentation is even more explicit. It contains pretty much a checklist of what you need to do.

> For the following messages, the container should first dispatch the message to the windowless object that has captured the mouse, if any. Otherwise, the container should dispatch the message to the windowless object under the mouse cursor. If there is no such object, the container is free to process the message itself:
>
> - WM_MOUSEMOVE
> - WM_SETCURSOR
> - WM_XBUTTONDOWN
> - WM_XBUTTONUP
> - WM_XBUTTONDBLCLK
>
> The container should dispatch the message to the windowless object with the keyboard focus for the following messages:
>
> - WM_CANCELMODE
> - WM_CHAR
> - WM_DEADCHAR
> - WM_HELP
> - WM_IMExxx
> - WM_KEYDOWN
> - WM_KEYUP
> - WM_SYSDEADCHAR
> - WM_SYSKEYDOWN
> - WM_SYSKEYUP
>
> For all other messages, the container should process the message on its own.

There it is, plain as day. Farproc's last question was "how to track or set 'focus' if there is at least one windowless control." Um, in a variable? I was kind of confused by this question because it's part of the deal that when you use windowless controls, you don't have the window manager to take care of keeping track of which sub-object has focus. That now becomes your job.

The user clicked on an object. I guess that's the focus object now. Oh wait, now the user hit the left arrow. I guess the object to the left of that object has focus now. It's just like any other control with windowless sub-components, like list boxes. You have to keep track yourself of the currently-selected item and other properties which the window manager normally does for you. If you don't have any windows, then there is nothing for the window manager to manage. From the window manager's point of view, focus is on your container. You then have to manage focus within your window yourself by keeping track of which of your sub-objects is the focus object.

Raymond Chen

Follow