

Modality, part 6: Interacting with a program that has gone modal

 devblogs.microsoft.com/oldnewthing/20050228-00

February 28, 2005



Raymond Chen

Earlier we saw the importance of setting the right owner window for modal UI. It is also important, when manipulating a window, to respect its modality. For example, consider the program we ended up with last time, the one which calls the `MessageBox` function to display a modal dialog. If we wanted to get that program to exit and sent a `WM_CLOSE` message to the main window instead of its modal popup, the main window would likely exit and leave the message box stranded, resulting in the same stack trace without support we saw in part 4.

Respect the modality of a window. If it is disabled, don't try to get it to do things; it's disabled because it doesn't want to do anything right now. You can go hunting for its modal pop-up and talk to that pop-up. (Unless, of course, that pop-up is itself disabled, in which case you get to keep on hunting.)

Raymond Chen

Follow

