

---

Subject: Re: Problem with non-modal children of modal dialogs

Posted by [Liam Gumley](#) on Mon, 14 Jun 1999 07:00:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Phillip & Suzanne David wrote:

> I remember this issue being discussed shortly after IDL 5.0 came out,  
> but not noticing that this affected me at the time. However, in some of  
> my programs, I display moderately complicated (modal) dialogs, and want  
> to provide a help capability for these. The problem I have using IDL is  
> that if I display the help in a different window, IDL requires that  
> window to be modal as well. Thus, when the user actually desires to use  
> the information in the help window, s/he must first close the window  
> (remembering the help tip) and act on the help information. There  
> appears to be no way to make the window stay up while allowing the user  
> to switch back to the modal dialog. My solution so far has been to  
> stick with the IDL 4.0 convention to XManager that allows non-modal  
> children of modal dialogs (although there appears to be no way to close  
> them except by closing the modal dialog). Does anyone know of a better  
> way to handle this?

I find it more convenient to manage widget sensitivity manually, using the SENSITIVE keyword to WIDGET\_CONTROL. For example, your program information structure could contain a master list (array) of widget names, and corresponding widget ids. Whenever you create a new widget, you pass the names of the widgets which should be \*insensitive\* to a routine that scans through the master list of widget names and ids. Whenever it finds a match on a widget name, it checks to see if the widget is active. If it is active, it desensitizes the widget using WIDGET\_CONTROL, ID, SENSITIVE=0. When the widget dies, you then call the same routine with the SENSITIVE keyword set, and again it scans through the list of widget names. Whenever it finds a match on a widget name, it checks to see if the widget is active. If it is active, it sensitizes the widget using WIDGET\_CONTROL, ID, SENSITIVE=1.

This way you don't have to worry about MODAL keywords, and you can set the sensitivity of your widgets exactly to your taste.

Cheers,  
Liam.

--

Liam E. Gumley  
Space Science and Engineering Center, UW-Madison  
<http://cimss.ssec.wisc.edu/~gumley>

---

Subject: Re: Problem with non-modal children of modal dialogs

Posted by [steinhh](#) on Mon, 14 Jun 1999 07:00:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Phillip wrote:

- > My solution so far has been to
- > stick with the IDL 4.0 convention to XManager that allows non-modal
- > children of modal dialogs (although there appears to be no way to close
- > them except by closing the modal dialog). Does anyone know of a better
- > way to handle this?

Yes. Make RSI change the way modal widgets work :-)

This may in fact be quite difficult, I'm not really sure to what extent the current *modus operandi* is determined by the architecture of the different window managers/window systems.

Note that I'm using the term "modal widgets", not "modal dialogs". Using the buzzword "dialog" was probably what tricked RSI into using the current solution. A dialog is essentially a braindead thing consisting of a question and a yes/no/maybe answer, and as long as an *\*application\** needs to get an answer to that question before going on with its work, it's ok if the rest of that *\*application\** is frozen.

But IDL has the potential to function almost like an "operating system" in some settings - with most of the work being done going on within IDL, running several different (not necessarily related) widget applications. Some applications will at times spawn *\*complex\** sub-applications that need to be modal - i.e. the calling application needs to be frozen until the sub-application finishes its task.

But how would *\*you\** react if e.g. your word processor used some braindead "dialog" to ask you whether or not to save your file, and that suddenly blocked all other functions on your computer..? Just won't do, will it?

I'm not sure I understand exactly what you mean by "there appears to be no way to close them except by closing the modal dialog", though...?

Regards,

Stein Vidar

---