
Subject: Re: exchanging parameters with a modal dialog
Posted by [David Fanning](#) on Fri, 24 Aug 2001 23:29:38 GMT
[View Forum Message](#) <> [Reply to Message](#)

Rick Baer (rick_baer@agilent.com) writes:

> What is the most elegant way to exchange data with a modal dialog box? I
> would like to change some parameters that exist within the scope of the main
> program. These parameters are currently part of the info structure of the
> main program, so they are like member variables of the "App" class (in MFC
> parlance). I could include the ID of the top-level base of the main program
> in the user data of the modal dialog, and then use widget_control to access
> the uvalue and modify the parameters. This would be like passing the
> "theApp" pointer and dereferencing it. Alternatively, I could just place the
> parameters in a common block. It would be much simpler, but I could only run
> one invocation of the program at a time. Any recommendations?

If this is truly a "modal" dialog, chances are you are going to pass the top-level base identifier to the program anyway as the Group Leader of the modal dialog. So, accessing the TLB's info structure will work fine.

Except... Here is a case where you really wish that TLB user value was a pointer to the info structure rather than the structure itself. Everything would be much easier that way. As a structure, you are going to be tempted to make another copy of it for the dialog's use.

But don't worry, keep passing the thing around with those NO_COPY keywords, and it acts like a pointer anyway. Just be sure you stuff it back in the original location (i.e., the user value of the Group Leader) before you issue the command that destroys the modal dialog. And since "commands" rarely kill widget programs (mice wielding users do), this might very well require some creative use of a Kill_Notify or TLB_Kill_Request_Events keyword.

Another way to do this, and one I would probably use if it were me, is to write the modal dialog as a function. If done properly, you could then call it like this:

```
Widget_Control, event.top, Get_UValue=info, /No_Copy  
modifiedinfo = MyDialog(info, Group_Leader=event.top)
```

Widget_Control, event.top, Set_UValue=modifiedinfo , /No_Copy

It could take the current info structure in, and pass a modified info structure back. You will have to take some care about what happens if the user kills the dialog before anything happens, but these are the usual details.

Cheers,

David

P.S. You are absolutely correct to **always** avoid common blocks in widget programs. They always lead to problems. :-)

--

David W. Fanning, Ph.D.

Fanning Software Consulting

Phone: 970-221-0438, E-mail: david@dfanning.com

Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

Toll-Free IDL Book Orders: 1-888-461-0155
