
Subject: Re: Sharing Variables within a Widget
Posted by [peter](#) on Tue, 18 Mar 1997 08:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

john Slavich (jcslavic@ouray.cudenver.edu) wrote:

: To anyone who can help me,

: I'm in the process of building my first widget program and now I'm at
: an impass. This is how my widget is presently structured:

: -> Function 1
: -> Function 2
: -> PRO, Event
: -> PRO (main)

: Function 1 and 2 have no widgets, they are called upon to make
: calculations on data set. Function 1 requires additional information
: calculated in Function 2 (i.e. the data Func. 1 requires is not part of
: the Return from Function 2). Since Functions can only return 1 "thing,"
: I would like to store information, calculated in Function 1 and 2, into
: the top event, so that I may call upon the information in other
: functions or procedures which are not widgets. For example, in the PRO
: event, I can call up information stored in a structure from the PRO
: main, by typing:
: Widget_control, event.top, Get_UValue=info
: All the info I need can be accessed in the structured called "info,"
: but when I do the same thing in Function 1 and 2, it doesn't work.
: I hope this is making sense, I'm still the learning process.

Presumably, function1 and function2 are called from the event handler.
So, retrieve the uvalue in the event handler (as you are doing) then
pass that info to f1 and f2. After f1 and f2 are done with the
structure, you'll return to the event handler, which can then put the
modified structure back into the uvalue of the widget.

This makes an appropriate clean separation between routines that build
and manage the widget, and routines that process data.

A more general comment: functions can return only one thing, but
procedures (and functions) can change as many of their arguments as you
want, since IDL used call-by-reference argument passing (most of the
time!). So it is possible to "return" as many things as you want.

Hope this helps,

Peter
