
Subject: Temporary variables still checked out
Posted by [Phil Aldis](#) on Wed, 26 May 1999 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

I seem to be having problems with this program. It creates a blocking widget, which has loads of exclusive and nonexclusive buttons on it, in columns. It is identical to the widget that pops up, when you do 'new plot' in IDL's insight. Every time I run it, if I hit the okay button, I get the message from IDL 'Temporary variables still checked out', or at least it's virtually like that, after the program exits. I've reduced the code down, and I'm pretty sure that the problem is in the code below. In fact the exact lines, seem to be after I've found out that the button was 'ok', and before I destroy the widget. When I commented these lines out I no longer got the 'Temporary variables still checked out'.

I don't know what's going on, but is it something to do with the structures and passing by value etc.

Cheers,
Phil

P.S. If you reply could you post it please as I need to see it at work.

```
;-----  
-----  
    getOptions  
;-----  
-----
```

PRO getOptions_Events, event

;Find out the button value

CASE buttonValue OF

'Cancel' : Widget_Control, event.top, /Destroy

'OK' : BEGIN

FOR i=0, N_Elements(info.optionsID)-1 DO BEGIN
 ;Loop through the columns and get the value
 Widget_Control, info.optionsID[i], Get_Value=thisOption
 IF info.exclusive[i] EQ 1 THEN BEGIN
 ;This is exclusive

```

        ((*info.ptr)).optIndex[i,*] = 0
        ((*info.ptr)).optIndex[i,thisOption] = 1
        ((*info.ptr)).optString[i,*] = ""
        ((*info.ptr)).optString[i,thisOption] =
info.options[thisOption]
    ENDIF ELSE BEGIN
        ;This is non exclusive
        ((*info.ptr)).optIndex[i,*] = thisOption
        ((*info.ptr)).optString[i,*] = ""
        ((*info.ptr)).optString[i,thisOption] =
info.options[thisOption]
    ENDELSE

```

```

ENDFOR

```

```

        ((*info.ptr)).cancel = 0
        ; Destroy the widget program

```

```

Widget_Control, event.top, /Destroy
END

```

```

ENDCASE

```

```

END

```

```

;-----
-----

```

```

FUNCTION getOptions, options ; A string array of options

```

```

;!!!!!!! This is the code for generating the widgets. It generates
a load of exclusive !!!!!!!!
;!!!!!!! buttons and non exclusive buttons in columns, and then put
labels by the side. !!!!!!!!
;!!!!!!! It's just like if you do in insight , new plot.
!!!!!!!
;!!!!!!! It also puts an okay and a cancel button
!!!!!!!

```

reformedDefaults is simply an array which is of size(the Number of Columns, the number of options)

```

optString = strArr( (Size(reformedDefaults, /DIMENSIONS))[0], (Size

```

```
(reformedDefaults, /DIMENSIONS))[1])
```

```
ptr = Ptr_New({optIndex:reformedDefaults, $ ;The indexes,  
with chosen options set to 1, not chosen set to 0  
  optString:optString, $ ;The strings of  
the chosen options, with non-chosen set to "  
  cancel:1}) ;Did they cancel
```

```
  ; Create info structure to hold information needed in event handler.
```

```
info = {cancel:cancel, $ ;The cancel widget button  
  ok:ok, $ ;The wid of the widget  
button 'ok'  
  optionsID:optionsID, $ ;The options button  
  options:options, $ ;the options array  
  exclusive:exclusive, $ ;Which ones were exclusive  
    ptr:ptr } ; Pointer to file information storage  
location.
```

```
  ; Store the info structure in the top-level base
```

```
Widget_Control, tlb, Set_UValue=info
```

```
  ; Register the program, set up event loop. Make this program a  
  ; blocking widget. This will allow the program to also be called  
  ; from IDL command line without a PARENT parameter. The program  
  ; blocks here until the entire program is destroyed.
```

```
XManager, 'getOptions', tlb, Event_Handler='getOptions_Events'
```

```
  ; OK, widget is destroyed. Go get the file information in the pointer  
  ; location, free up pointer memory, and return the file information.
```

```
retInfo = *ptr  
Ptr_Free, ptr
```

```
RETURN, retInfo  
END
```

```
--== Sent via Deja.com http://www.deja.com/ ==--  
---Share what you know. Learn what you don't.---
```

Subject: Re: Temporary variables still checked out
Posted by [steinhh](#) on Thu, 27 May 1999 07:00:00 GMT

Peter Mason wrote:

[..]

>> ((*(info.ptr)).optIndex)[i,*] = 0

> => Instead of this, try (*(info.ptr).optIndex[i,*]=0; etc.

>

> Things DO get a bit unsettling when (structures of) pointers to

> structures are involved, but a bit of command line action will

> normally clarify IDL's workings with brackets soon enough. (Basically,

> you must restrict your bracketing to just the pointer component(s).)

Which reminds me that I should have registered my wish for
a pointer-to-structure-dereference operator (like C) with
RSI.... What link was that..?

Wouldn't it be great to be able to say just

info.ptr-->optIndex[i,*]

instead of all that gibberish..... And as for the "name" of
the operator, there shouldn't be any real problem in using
"-->" even though it's also used for objects, but if they want
to avoid the extra checking, they could go for "-->" like
above instead.... or something else.

What **really** bugs me is why on earth they didn't provide
an operator like this in the first place when they introduced
pointers..... But by all means, don't get me wrong, pointers
and objects are the best things to happen to IDL since....
well, ... since ... hmm, since I discovered the Emacs
editing/shell mode for IDL!

Regards,

Stein Vidar
