
Subject: Idl pointers/widget events

Posted by [dhamilton](#) on Thu, 13 Aug 1998 07:00:00 GMT

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

I am having a pointer problem. I use a pointer to maintain information from a widget event handler (as per David Fanning). However, despite the fact that the "info" structure which contains the pointer is passed successfully into the container with "set_uvalue", the pointer itself becomes invalid. Is there some problem with IDL for VMS or am I just doing something wrong?

```
ptr = ptr_new({thing:thing, junk:junk})
info = {stuff:stuff, ptr:ptr}
widget_control, tlb, set_uvalue=info, /no_copy
xmanager, "config", tlb, /no_block
```

...

(now, in my event_handler "config_event":)

```
widget_control, event.top, get_uvalue=info, /no_copy
```

(at this point, "info" is a completely valid structure, and I can examine "info.stuff" to my heart's content. However, an attempt to reference "(*info.ptr)" results in crashing my program and an invalid pointer message.)

I have also noticed some inconsistency with this problem between regular IDL and IDL/DE, which seems impossible. However, sometimes my code works in the development environment, but not in the regular command line. ?

Thanks for your time and help
Damian Hamilton, SLAC

Subject: Re: Idl pointers/widget events

Posted by [rmlongfield](#) on Mon, 17 Aug 1998 07:00:00 GMT

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

In article <6qvmce\$2i\$1@nnrp1.dejanews.com>,
mirko_vukovic@notes.mrc.sony.com wrote:

```
>
> Since you are copying only the pointer address, there is no need to
> use the /no_copy. That was the main point of using pointers. They
> eliminate the use of /no_copy in widget events, and thus, any changes
> you make to info, you do not need to restore it to the widgets uvalue
> at the end of the event routine.
>
> mirko
```

Aha, mirko, thanks for finally clarifying what 'globality' of pointers means.

I've been using pointers to structures to hold all sorts of information. At first I thought global pointers meant that I didn't have to use the `get_Uvalue` statements at all, (i.e. like common statements). But no, it was just with respect to this `/no_copy` command. So, there was some improvement, except that now there are these crazy formats to access the data. I almost always put the parentheses in the wrong place. Accessing an array element, as you mentioned, was even more fun and I figured it out only by trial and error. One word of advice to anyone with a memory as bad as mine: write mirko's format down somewhere and pin it on the wall. Regarding widgets, I can say that I am able to write 'functioning' widget programs, although they are very messy and very unfriendly, due in part to this clumsy pointer referencing. COMMON statements are indeed cleaner and user friendly, (but the dark path is always the easiest...) Regarding objects, I am a bit reluctant to make the jump, although I am reading all the discussion in this group just to get used to the words and how these words are used. (This is the worst part of learning a new concept). Having worked with widgets, I can see where objects might be useful.

I really appreciate the discussion in this usergroup as well as answers to specific questions.

Rose (Still trying to think of a nice quote)

-----== Posted via Deja News, The Leader in Internet Discussion ==-----
http://www.dejanews.com/rg_mkgrp.xp Create Your Own Free Member Forum
