
Subject: Re: problems with passing structures around in gui widgets

Posted by [David Fanning](#) on Wed, 04 Aug 2004 20:11:04 GMT

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

RS writes:

> I'm writing a GUI to streamline some data reductions and have run into
> two problems passing information from the main program to the event
> handler and back:
>
> 1) I can't reassign arrays to arrays of different size.
> I have created a 1x1 array (as close to empty as IDL could get-- [0])
> in the main (base widget) that I want to assign 1024x1024 or 1024x160
> values during event handler calls. However, IDL won't write a larger
> array over a smaller one in this case, even though it will at the
> command line... ex:
> a=indgen(1,1,1)
> a=indgen(2,2,2)
>
> 2) I can only access (& change) the data from the main once and then
> IDL spews back this message:
> % Expression must be a structure in this context: INFO.
> % Execution halted at: INTERFACE_EVENT 16
> % WIDGET_PROCESS_EVENTS
> % \$MAIN\$
> Which is great, except that info IS a structure that I'm using in the
> main to contain everything I want to be able to alter during the
> events and setting it to be the uvalue of the top-level base. So, I
> can load the data, but not perform any operations on it, because that
> would require calling a second event. Peachy.
>
> Any ideas as to workarounds?

Oh, dear. You need a book. :-)

When something in a structure is changing (either size or data type)
we make that field a pointer:

```
info = { name:'Coyote', data:Ptr_New(/Allocate_Heap)}
```

Then, you can put whatever you like into it:

```
*info.data = FltArr(10)  
*info.data = BytArr(50, 100)
```

Be sure to free you pointer in your CLEANUP procedure.
What!? You don't have a clean up procedure? Better read
the documentation on the XMANAGER command. :-)

```
PRO Myprogram_Cleanup, tlb
Widget_Control, tlb, Get_UValue=info
Ptr_Free, info.data
END
```

The most common reason for your error message is that you took the info structure out of your TLB with a NO_COPY and forgot to put it back before you left that event handler. This means you can process one event, but no more.

The standard rule is to check the info structure out at the top of the event handler, and check it back in before you exit the event handler:

```
PRO MyProgram_Events, event
Widget_Control, event.top, Get_UValue=info, /No_Copy

info.data = blahblahblah

Widget_Control, event.top, Set_UValue=info, /No_Copy
END
```

Cheers,

David

--

David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: <http://www.dfanning.com/>
