
Subject: Re: Help with pick menu widget code
Posted by [davidf](#) on Tue, 15 Sep 1998 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Doug Larson (djl@REMOVECAPSloki.srl.caltech.edu) writes:

> I am attempting to make a small program to pop up some menu
> buttons based on a user set resource. I am attaching the
> code to this post in the hopes that someone can tell me
> how to make it exit properly!
>
> My intent was to understand how to handle different
> combinations of widgets in one window. Even after repeated
> readings from "The Book of David" I still can't get this to go.

Whew! For a moment there I thought you were reading my book, but then I had a look at the code. Since I **ALWAYS** put the main program unit as the last one in the file, I can tell it's not my book you are reading. :-)

You are not going to believe this Doug, but the only thing this program is missing is that you forgot to get the pointer in your button event handler. :-)

I added this line to the top of the PickFinish_Event function:

```
Widget_Control, event.top, Get_UValue=pickPtr
```

Now it exits almost perfectly. There is one small problem, however. And that is that you **did** remember to put the pointer back at the end of the routine with this line:

```
WIDGET_CONTROL, event.top, SET_UVALUE= pickPtr, /NO_COPY
```

This piece of code fails now, however, because the action of the DONE button is to destroy the top-level base. Of course, you can't stick something into the user value of a widget that no longer exists.

I might fix this problem like this:

```
IF Widget_Info(event.top, /Valid_ID) THEN $  
  WIDGET_CONTROL, event.top, SET_UVALUE= pickPtr, /NO_COPY
```

But, to tell you the truth, I wouldn't be fooling around with NO_COPY keywords if what I'm passing around is a pointer. There really is no point to it. (Pun intended.) A pointer is a two

byte integer. I wouldn't be worrying about getting it with a NO_COPY and I wouldn't be worrying about putting it back. If you have the pointer reference, you have the thing itself. There is nothing whatsoever to copy.

Cheers,

David

David Fanning, Ph.D.
Fanning Software Consulting
E-Mail: davidf@dfanning.com
Phone: 970-221-0438, Toll-Free Book Orders: 1-888-461-0155
Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

Subject: Re: Help with pick menu widget code
Posted by [davidf](#) on Wed, 16 Sep 1998 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Doug Larson (djl@REMOVECAPSloki.srl.caltech.edu) writes:

> Well, I did what David told me, and of course it works :)

I've been working on a hoary satellite data problem this morning (Talk about working for cheap, this is the LAST time I bid a job on a project basis. Who writes this satellite packing documentation anyway? They are not going to lose their jobs, that's for sure. They are the only people on God's green earth who can understand that overly thick pile of nonsense!)...where was I?

Oh, yes. I've been working on a hoary satellite data problem this morning and I decided a bike ride in the crisp, Colorado air was exactly what the doctor ordered. I hadn't gone a block before I realized I had probably just screwed Doug up for good with my last answer. :-(

I can't really make out *what* he is trying to do with this program, but it's possible he wants to pass the pickedPtr back to the calling program. In that case, he definitely does NOT want to destroy the pointer in the CLEANUP routine. In fact, if that is what he wants to do, the program should be working now, as far as I can see.

Cheers,

David

David Fanning, Ph.D.
Fanning Software Consulting
E-Mail: davidf@dfanning.com
Phone: 970-221-0438, Toll-Free Book Orders: 1-888-461-0155
Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

Subject: Re: Help with pick menu widget code
Posted by [davidf](#) on Wed, 16 Sep 1998 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Doug Larson (djl@REMOVECAPSloki.srl.caltech.edu) writes:

> Well, I did what David told me, and of course it works :)

Well, you're lucky. I figured your chances were better than average at most when I fired the answer off at 11:00 last night. :-)

> The second part with the "Widget_Info" still confuses me as to why it
> works.

Here is the code:

```
IF (Widget_Info(event.top, /Valid_ID) THEN $  
    WIDGET_CONTROL, event.top, SET_UVALUE= pickPtr, /NO_COPY
```

You are probably confused because I did the usual advanced IDL thing of combining two commands in one. Widget_Info is a function that returns (in this case) either a zero or one based on whether the event.top widget is still a valid widget or not. I'm using the return value as the test in the IF statement. If the widget ID is valid, I put something in it's user value. If the widget ID is not valid, I just skip this statement.

> With the destruction of the top-level base preceding the SET_UVALUE, how
> does PickFinish ever know what was done to the values in the pickPtr? I
> guess I still have not grasped this whole event driven style of code yet.

To tell you the truth, PickFinish doesn't even care. :-)
Each time you go through *any* of your event handlers and do something to the data pointed to by pickPtr, you are changing the thing pointed to, not the pointer itself.

And the thing pointed to exists someplace else. (It exists on

the HEAP, which is where all pointers and objects live.) What you have in this program is a *reference* to that space. When you destroy that reference, you do not destroy the things that the reference points to. This is both good news and bad news, although I would say it is mostly bad news for your program at the moment.

I say this because it looks like to me that when you destroy that top-level base you are destroying the only reference you have to that stuff stored in the pointer. This results in a condition we call "leaking memory". In other words, each time you run this program some of your memory gets allocated and then leaks away as you destroy the only way you have to get to it. Over time, you will find that you have no more memory to do anything useful. (And Rose talks about cryptic error messages. You should see this one!).

To clean this pointer up, I recommend you use a CLEANUP routine. The purpose of the cleanup routine is to destroy pointer and object references, delete pixmaps, and generally do the sorts of things that need to be done when a program exits. This routine can be assigned to the top-level base at the time you call XMANAGER with the CLEANUP keyword:

```
XMANAGER, 'picked', pickTLB, GROUP_LEADER = pickTLB, $
CLEANUP='PICKMENU_CLEANUP'
```

The routine (and please place *all* your event handler and other routines IN FRONT OF your PickMenu routine. That way your program will work correctly when it is called and you won't have to individually compile it before you use it.) is written like this:

```
PRO PICKMENU_CLEANUP, pickTLB
Widget_Control, pickTLB, Get_UValue=pickPtr
Ptr_Free, pickPtr
END
```

When the TLB dies, this routine is called. It's sort of like calling a priest to issue last rites. The TLB is "mostly dead" to use one of my favorite phrases from the movie "The Prince's Bride". All the TLB can do is whisper the name of the thing in its user value. :-) You use this name to your own advantage, in this case freeing both the pointer and the data the pointer points to, thereby avoiding memory leakage.

> Would it be better to have the destruction "percolate up" to
> a handler for the pickTLB? If so, how do I do this?

Better, I think, to have a separate event handler for the Done button. I would assign it when I created the button, like this:

```
done = WIDGET_BUTTON(finishbase, VALUE = 'DONE', $  
    Event_Pro='Out_of_Here')
```

It would be written like this (provided I had written the Cleanup routine as I recommended above):

```
PRO Out_Of_Here, event  
Widget_Control, event.top, /Destroy  
END
```

Cheers,

David

P.S. I would *really* give that event handler the name "PICKMENU_OUT_OF_HERE" if I was trying to write code that works all of the time. :-)

David Fanning, Ph.D.
Fanning Software Consulting
E-Mail: davidf@dfanning.com
Phone: 970-221-0438, Toll-Free Book Orders: 1-888-461-0155
Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

Subject: Re: Help with pick menu widget code
Posted by [Doug Larson](#) on Wed, 16 Sep 1998 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Well, I did what David told me, and of course it works :)

The added "event.top" line I understand, sloppy of me, bad Doug!

The second part with the "Widget_Info" still confuses me as to why it works.

With the destruction of the top-level base preceding the SET_UVALUE, how does PickFinish ever know what was done to the values in the pickPtr? I guess I

still have not grasped this whole event driven style of code yet.

Would it be better to have the destruction "percolate up" to a handler for the pickTLB? If so, how do I do this?

--

Douglas J. Larson e-mail: djl@loki.srl.caltech.edu
Space Radiation Lab
California Institute of Technology
Mail Code: 220-47
Pasadena, CA 91125

Subject: Re: Help with pick menu widget code
Posted by [davidf](#) on Wed, 16 Sep 1998 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Rose Longfield writes:

- > I wasn't able to solve Doug's problem. I found the Uvalue error but could
- > not understand why the Widget ID was not valid, when, every time I printed it
- > it had a number that matched the event.top.
- >
- > That's because I was looking in the wrong place! As David Fanning astutely
- > pointed out, the error was in the SET command at the bottom of the program,
- > FAR away from where I thought it was.

When you have taught as many IDL courses as I have both of these common errors just sort of jump out at you. :-) But if you get an error message that a pointer (or info) structure is not defined, there are only about three things that could be wrong: (1) You didn't copy it into the program module (the problem here, but the least likely error in my experience), (2) You forgot to put the pointer (or info structure) into the UValue of the TLB (the most likely possibility), or (3) you took it out with a NO_COPY, but failed to put it back before you exited the module. (The last problem should occur only with info structures. You shouldn't be copying pointers, as I pointed out yesterday.)

- > The message from XMANAGER states simply: % XMANAGER: Caught unexpected error
- > from client application. Message follows... % WIDGET_CONTROL: Invalid widget
- > identifier: 61.

I agree that this is a cryptic message, although one I see so often when people are writing their first widget programs that I know the solution almost without thinking about it now. I do remember struggling for an hour or so the first time it happened to me, however.

- > There are widget controls all over the code, is there any way for XMANAGER to

> be a little more helpful about identifying, WHICH widget_control was causing
> the error?

Let me give you a tip for solving errors like this a little more easily. The XManager routine has a CATCH statement in it. The way error handling in IDL works is that if there is a CATCH statement anywhere along the "program trace" (I don't know a good way to say this, but if one program calls another and that program calls another, you are now three programs deep in the "trace"), then the error gets handled by that CATCH statement.

What you really want is for the error in your widget program to propagate all the way up to the main IDL level, where IDL will attach a line number to the error. Then, at least, you will know what line of code caused the error, if not what the error means exactly.

To make the error propagate all the way up, you have to turn any CATCH error handling off in the module that is causing the error, and you have to turn it off for XManager. You do this by typing this command on the IDL command line *before* you run your program:

```
IDL> XManager, Catch=0
```

Now when you run the program, you get this error message:

```
% WIDGET_CONTROL: Invalid widget identifier: 3.  
% Execution halted at: PICKFINISH_EVENT 323 C:\RS\IDL51\david\junk.pro
```

I can go to line 323 in my code and try to figure out why the only widget identified there (event.top) is somehow invalid. It may only take me a half-hour to realize that I just killed the sucker. :-)

Cheers,

David

P.S. Rose, are you coming to my October course in Denmark?

David Fanning, Ph.D.
Fanning Software Consulting
E-Mail: davidf@dfanning.com
Phone: 970-221-0438, Toll-Free Book Orders: 1-888-461-0155
Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

Subject: Re: Help with pick menu widget code
Posted by [rmlongfield](#) on Wed, 16 Sep 1998 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi All,

In article <MPG.1068db96e92bab119896b2@news.frii.com>,
davidf@dfanning.com (David Fanning) wrote
> Doug Larson (djl@REMOVECAPSloki.srl.caltech.edu) writes:

I wasn't able to solve Doug's problem. I found the Uvalue error but could not understand why the Widget ID was not valid, when, every time I printed it it had a number that matched the event.top.

That's because I was looking in the wrong place! As David Fanning astutely pointed out, the error was in the SET command at the bottom of the program, FAR away from where I thought it was.

The message from XMANAGER states simply: % XMANAGER: Caught unexpected error from client application. Message follows... % WIDGET_CONTROL: Invalid widget identifier: 61.

There are widget controls all over the code, is there any way for XMANAGER to be a little more helpful about identifying, WHICH widget_control was causing the error?

Rose

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

Subject: Re: Help with pick menu widget code
Posted by [davidf](#) on Thu, 17 Sep 1998 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Rose Longfield (rmlongfield@my-dejanews.com) writes:

> Does this mean that I have to know before-hand that my module is going to
> cause an error? (Isn't this rather pessimistic) Or, I guess it is a good
> de-bugging technique, to be applied AFTER the program crashes. In this case I
> will try it out.

I use it as a debugging technique after I get over the shock and dismay that my carefully crafted program didn't run perfectly. :-)

> I haven't had much luck with error handling in IDL. It usually causes me more
> trouble because I don't notice that an error has occurred.

This can be one of the big disadvantages of the CATCH mechanism for catching errors. I once spent about three hours thinking IDL had gone completely bonkers only to finally realize I had added a silent CATCH to one of my modules. Needless to say, I don't add any more silent CATCHes to *anything*. I write them like this:

```
CATCH, error
IF error NE 0 THEN BEGIN
  Catch, /Cancel
  ok = Dialog_Message(!Error_State.Msg)
  Print, !Error_State.Msg
  Widget_Control, event.top, Set_UValue=info, /No_Copy
  RETURN
ENDIF
```

The Cancelling of the Catch keeps me from the infinite loop that results when I incorporate errors in my error handling code. Sigh... The Dialog_Message makes sure I know an error occurred, because it is a modal dialog I *must* respond to. The Print statement is there for me to refer to when I hit the OK on the dialog before carefully reading the message and I am trying to debug my code.

This CATCH will keep a widget program alive indefinitely, but it also usually gives me enough notice to be able to fix a problem. I suppose more bells and whistles and flashing lights wouldn't hurt. My problem is that I see this damn dialog so often that I have become habituated to it and almost never *really* see it. :-)

Of course, occasionally, I want to disable this CATCH, perhaps to find one of those "client application" errors. Rather than add semi-colons to each of these lines (there are occasionally more as I sometimes do *real* error handling too), I modify the code like this:

```
;CATCH, error
error = 0
IF error NE 0 THEN BEGIN
  Catch, /Cancel
  ok = Dialog_Message(!Error_State.Msg)
  Print, !Error_State.Msg
  Widget_Control, event.top, Set_UValue=info, /No_Copy
  RETURN
ENDIF
```

Cheers,

David

David Fanning, Ph.D.
Fanning Software Consulting
E-Mail: davidf@dfanning.com
Phone: 970-221-0438, Toll-Free Book Orders: 1-888-461-0155
Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

Subject: Re: Help with pick menu widget code
Posted by [rmlongfield](#) on Thu, 17 Sep 1998 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

HI All,

In article <MPG.10697a29e82919e09896b3@news.frii.com>, davidf@dfanning.com (David Fanning) wrote:

A lot of interesting stuff about errors but this one confuses me:

>
> To make the error propagate all the way up, you have to turn
> any CATCH error handling off in the module that is causing
> the error, and you have to turn it off for XManager. You
> do this by typing this command on the IDL command line
> *before* you run your program:
>
> IDL> XManager, Catch=0
>
> Now when you run the program, you get this error message:
>
> % WIDGET_CONTROL: Invalid widget identifier: 3.
> % Execution halted at: PICKFINISH_EVENT 323 C:\RSI\IDL51\david\junk.pro
>
> I can go to line 323 in my code and try to figure out why the only
> widget identified there (event.top) is somehow invalid. It may
> only take me a half-hour to realize that I just killed the sucker. :-)

Does this mean that I have to know before-hand that my module is going to cause an error? (Isn't this rather pessimistic) Or, I guess it is a good de-bugging technique, to be applied AFTER the program crashes. In this case I will try it out.

I haven't had much luck with error handling in IDL. It usually causes me more trouble because I don't notice that an error has occurred. I tried handling the error by skipping over some code and printing, 'you have an error'. This statement frequently gets printed without me noticing until somewhere else the program notices that I skipped over something and THERE is where the error occurs. These days I just let the program crash and then try to fix

it. ("ambulance down in the valley " thinking).

I recently discovered SOUND among Ray Stern's programs. (If he could do it with a SUN, I can do it with a SGI.) This way I can hear that something is wrong. (A few well chosen Darth Vader sounds would be suitable, I think.)

Good luck with the widgets, Doug!

Rose

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