
Subject: Objects and Widgets

Posted by [Mark Guagenti](#) on Tue, 09 May 2000 07:00:00 GMT

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

I am trying to learn how to use objects and widgets together and I have ran across a question for all of you. When I am creating a widget button such as:

```
openID = Widget_Button(fileID, Value='Open...', Event_Pro='XPU_RI_Open')
```

Do I have to make a procedure XPU_RI_Open such as:

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

```
info.obj->open
```

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

How come you can't call the object method directly from Event_Pro? Since I will be using a lot of methods will I always have to make a procedures to call them? Thanks for the help.

-- Mark

Grace and peace to you from God our Father and the Lord Jesus Christ.
1 Cor. 1:3

Subject: Re: objects and widgets

Posted by [btt](#) on Wed, 05 Jul 2006 13:21:50 GMT

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

gnarlo@libero.it wrote:

```
> Dear all, I havent been around for pretty a long time, anyway, hope you  
> all are doing fine.  
> Imagine a program with a gui interface and callback procedures.  
> i am writing the callback procedures, (basically the core of the  
> program) but one thing occurs. I am using objects (self made, i wrote  
> structures and methods) and i create them with obj_new at the '  
> beginning' of my program, that is, i create the objects in the realize  
> callback procedure of the base_0 widget. when I try to use the objects  
> i have created in another unit program, for example a mouse button  
> procedure, the object appears as undefined. I know for sure that the  
> object was created but it appears that it doesnt have global scope. can  
> anyone help? thanx to all in advance.
```

>

Hi,

While objects are persistent heap variables they really don't have the global scope you're assuming they do. You still have to tuck a reference to the object somewhere handy in your program. Your best bet is to store a reference to your objects in the UVALUE of the widget generating the event. I like to store both the object reference and the method that I want to call when the event occurs.

```
myWidgetID = WIDGET_SOMETHING(parent, ..., $
  UVALUE = {OBJECT:myObjectReference, METHOD: "MethodForHandlingEvent"})
```

In the event callback access the UVALUE that contains the reference using (very simple example)

```
PRO MYEVENT, ev
  WIDGET_CONTROL, ev.ID, get_Uvalue = myInfo
  CALL_METHOD, myInfo.method, myInfo.object, ev
END
```

On the ATTVIS user contribution website there are some generic object-associated event handlers written by Jim Pendelton that give you more details. As always, check out David Fanning's very detailed resources on this subject.

Cheers,
Ben

Subject: Re: objects and widgets
Posted by [Michael Galloy](#) on Wed, 05 Jul 2006 14:15:45 GMT
[View Forum Message](#) <> [Reply to Message](#)

gnarlo@libero.it wrote:

> Dear all, I haven't been around for pretty a long time, anyway, hope you
> all are doing fine.
> Imagine a program with a GUI interface and callback procedures.
> I am writing the callback procedures, (basically the core of the
> program) but one thing occurs. I am using objects (self made, I wrote
> structures and methods) and I create them with obj_new at the '
> beginning' of my program, that is, I create the objects in the realize
> callback procedure of the base_0 widget. When I try to use the objects
> I have created in another unit program, for example a mouse button
> procedure, the object appears as undefined. I know for sure that the

> object was created but it appears that it doesn't have global scope. can
> anyone help? thanx to all in advance.
>
> cheers,
> Michele
>

Check out this article for one example of doing this:

<http://michaelgalloy.com/2006/06/14/object-widgets.html>

Mike

--

www.michaelgalloy.com

Subject: Re: objects and widgets

Posted by [larkn10](#) on Thu, 06 Jul 2006 14:36:20 GMT

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

Another thing you can do is something like the following:

```
;GET ALL VALID OBJECTS
  ov = obj_valid()

;COUNT THE INSTANCES OF YOUR CLASS
  className = 'TheNameOfYourClass'
  count = 0
  for i=0,n_elements(ov)-1 do begin
    if obj_isa(ov[i],className) gt 0 then count++
  endfor;i
  print,count

;IF YOU WANT THE OBJECT REFERENCE(S)
  if count gt 0 then begin
    ref = obj_arr(count)
    for i=0,n_elements(ov)-1 do begin
      if obj_isa(ov[i],className) gt 0 then ref = ov[i]
    endfor;i
  endif
```

This would retrieve all subclasses of the class you are interested in, so you might want to do something more to search for the specific class you need.

You probably wouldn't want to do this in a program, but it could be useful at times.

-Larry
