Subject: Re: explicit redraw does nothing until expose(?) event - IDLitComponent propertysheets
Posted by David Fanning on Wed, 21 Apr 2004 19:27:52 GMT
View Forum Message <> Reply to Message

Sean Dettrick writes:

> I've got an object window with a bunch of plots on it. I popup a
> WIDGET_PROPERTYSHEET to change the properties of my IDLgrLegend
> (registered with IDLitComponent), and it works nicely, but the actual
> graphic is never redrawn unless I move my legend around with the
> mouse, or drag my popup window over the top of the draw window.
>
> So I add a Re-Draw button to my pop-up property sheet, in the hope
> that I can force a draw manually (I make sure it has access to the
> right IDLgrWindow and IDLgrView objects). Clicking on my Re-Draw
> button multiple times does nothing, until I either click on the legend
> or drag the popup over the draw window. Then all my queued Re-Draw
> events execute one after the other.
>
> What's happening? Is there a way to get my redraw event processed
> immediately? Why doesn't the thing process the event until I expose
> or interact with the draw widget?

What happens if you remove the FLOATING keyword from
the pop-up? Sounds like a semi-modal widget problem to
me. I wonder if the FLOATING keyword has something to
do with that?

Cheers,

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

Subject: Re: explicit redraw does nothing until expose(?) event - IDLitComponent propertysheets
Posted by Chris[2] on Wed, 21 Apr 2004 19:29:43 GMT
View Forum Message <> Reply to Message

How are you sending the event to your button handler? Are you simply calling
the method directly? Or are you trying to use widget_control with
send_event? Using send_event would be a bad idea.

Why not simply add a win->draw at the end of your property sheet event

handler? You know that if a property changes, you want to redraw.

-Chris

"Sean Dettrick" <sdettrick@hotmail.com> wrote in message
news:c233c3ea.0404211042.19412b3d@posting.google.com...
> Hi,
>
> I've got an object window with a bunch of plots on it. I popup a
> WIDGET_PROPERTYSHEET to change the properties of my IDLgrLegend
> (registered with IDLitComponent), and it works nicely, but the actual
> graphic is never redrawn unless I move my legend around with the
> mouse, or drag my popup window over the top of the draw window.
>
> So I add a Re-Draw button to my pop-up property sheet, in the hope
> that I can force a draw manually (I make sure it has access to the
> right IDLgrWindow and IDLgrView objects). Clicking on my Re-Draw
> button multiple times does nothing, until I either click on the legend
> or drag the popup over the draw window. Then all my queued Re-Draw
> events execute one after the other.
>
> What's happening? Is there a way to get my redraw event processed
> immediately? Why doesn't the thing process the event until I expose
> or interact with the draw widget?
>
> confused,
> Sean
> In more detail:
> The TLB and Draw widgets are defined as:
>
> tlb = Widget_Base(TLB_Size_Events=1)
> drawID = Widget_Draw(tlb, $
> Button_Events=1, $
> Expose_Events=1, $
> Retain=0, $
> Graphics_Level=2, $
> Event_Pro='Select_Button_Events')
>
> The plot legend object has SELECT_TARGET=1, so after it is clicked on,
> the event handler calls a method of the selected target:
>
> target -> propertysheet, group_leader=event.top, $
> window=info.mywindow, $
> view=info.myview
>
> This class method pops up a widget with the ReDraw button and the
> WIDGET_PROPERTYSHEET. It looks a bit like this:
>

```
>  pro myLegend::PropertySheet, group_leader=group_leader, $
>                          window=window,$
>                          view=view
>  self.window=window
>  self.view=view
>  wpopup = widget_base( /FLOATING, $
>                      GROUP_LEADER=GROUP_LEADER $
>                    )
>  button = WIDGET_BUTTON( wpopup, $
>                       value='Re-draw window', $
>                       event_pro='handle_all_events', $
>                       UVALUE = {object:self, $
>                              method:'ReDraw'} $
>                      )
>  result = WIDGET_PROPERTYSHEET( wpopup, $
>                       VALUE = self , $
>                       event_pro = 'handle_all_events', $
>                       UVALUE = {object:self, $
>                              method:'ChangeProperty'} $
>                      )
>  widget_control, wpopup, /realize
>  end
>
>  (Thanks to David Fanning and Stein Vidar for this way of getting a
>  class method to be a proxy event handler.)
>  The handle_all_events.pro routine passes the WIDGET_BUTTON event to
>  the myLegend::ReDraw method, which is as simple as this:
>
>  pro myLegend::ReDraw, event
>     if event.select eq 1 then begin
>        print,'REDRAW on request:'
>        self.window -> Draw, self.view
>     endif
>  end
>
>  Self.window and self.view are the original draw window and view.
>  If I click on the button 5 times, nothing happens until I click on the
>  legend or move the pop-up over the draw window.  Then all 5 draw
>  events execute directly!
>
>  Any suggestions?
```

Subject: Re: explicit redraw does nothing until expose(?) event - IDLitComponent propertysheets
Posted by sdettrick on Fri, 23 Apr 2004 01:55:30 GMT
View Forum Message <> Reply to Message

> How are you sending the event to your button handler? Are you simply calling
> the method directly? Or are you trying to use widget_control with
> send_event? Using send_event would be a bad idea.

I don't even know what send_event is. Instead I am sending the button event using the EVENT_PRO keyword to specify a proxy event handler called handle_all_events:

```
button = WIDGET_BUTTON( wpopup, $
                  value='Re-draw window', $
                  event_pro='handle_all_events', $
                  UVALUE = {object:self, $
                          method:'ReDraw'} $
                 )
```

The handle_all_events routine inspects the UVALUE of the widget causing the event, and uses CALL_METHOD to call my ReDraw method on the object (David Fanning credits Stein Vidar for this idea)

```
  PRO Handle_All_Events, event
  Widget_Control, event.id, Get_UValue=info
  Call_Method, info.method, info.object, event
  END
```

I can't see why that should change anything. But could CALL_METHOD somehow slow things down?

>
> Why not simply add a win->draw at the end of your property sheet event
> handler? You know that if a property changes, you want to redraw.
>

Well I tried that too. It doesn't change the behaviour. I think that for some reason the events are just not making it to the event handler until I interact with the draw window. Could they be trying to go somewhere else, e.g. into a non-existent iTool?

To answer David's question:

> What happens if you remove the FLOATING keyword from
> the pop-up? Sounds like a semi-modal widget problem to
> me. I wonder if the FLOATING keyword has something to
> do with that?

Good question. I pull out the FLOATING keyword and the behaviour doesn't change, only the popup appears on a different part of the screen. My REDRAW button event doesn't get processed any sooner.

It's a little frustrating but I haven't a clue on how to resolve it.
Any other ideas?

BTW I tried it on both Linux and Windows, and it shows the same
behaviour.

Thanks,
Sean

Subject: Re: explicit redraw does nothing until expose(?) event - IDLitComponent propertysheets
Posted by David Fanning on Fri, 23 Apr 2004 03:30:40 GMT
View Forum Message <> Reply to Message

Sean Dettrick writes:

> The handle_all_events routine inspects the UVALUE of the widget
> causing the event, and uses CALL_METHOD to call my ReDraw method on
> the object (David Fanning credits Stein Vidar for this idea)

Really!? I guess I thought I had "invented" it myself.
Maybe I re-discovered it independently after Stein Vidar
showed it to me and I forgot all about it. :-(

> I can't see why that should change anything. But could CALL_METHOD
> somehow slow things down?

Well, probably. But what you are seeing isn't a slow
down, it is a screeching halt.

>> Why not simply add a win->draw at the end of your property sheet event
>> handler? You know that if a property changes, you want to redraw.
>>
>
> Well I tried that too. It doesn't change the behaviour.

Wait a minute. You called the method *directly* from
the event handler module you *know* you are in, and
it didn't change anything!? Now, that is truly strange.

It looks to me as if something is blocking events.
I can't imagine it is the PropertySheet, since I've
managed to interact with PropertySheets before and
I get updates directly and immediately. And there
would be no reason for a property sheet to block,
I don't think.

> I think that
> for some reason the events are just not making it to the event handler
> until I interact with the draw window. Could they be trying to go
> somewhere else, e.g. into a non-existent iTool?

In RSI supplied code!? No, I don't think so. Now, if
you thought there was an error that was being handled
silently...well, let's just say I wouldn't dismiss it
out of hand.

> It's a little frustrating but I haven't a clue on how to resolve it.
> Any other ideas?

Not really, but I have risked $10 in favor of a programmer error
rather than a software bug. :-)

There is always a danger in programming object-widgets that
events are not occurring in exactly the way you *think*
they are occurring. In fact, programming widget-objects
is a good way to find yourself embracing a mystical
religious tradition. You might want to step through
your program to figure out where you are coming
from and going to when you pop up that dialog. I have
occasionally found myself at the end of a procedure
I thought I had exited eons ago. Something like this
might be occurring. (Fortunately, we built a "verbose"
switch into the 2nd version of the Catalyst Library,
so it can show us graphically which program modules
we enter and leave. This has saved us from just this
kind of frustration several times.)

Do you happen to be in Boulder, Sean? I'm not sure
why I think that, but I'm going to be up that way
tomorrow. I'd be curious to see what this looked like
in real life.

Cheers,

David

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

Subject: Re: explicit redraw does nothing until expose(?) event - IDLitComponent propertysheets
Posted by sdettrick on Fri, 23 Apr 2004 17:19:04 GMT
View Forum Message <> Reply to Message

David Fanning <david@dfanning.com> wrote in message
news:<MPG.1af23bcb89afdf6f98972a@news.frii.com>...
> Sean Dettrick writes:
>
>> The handle_all_events routine inspects the UVALUE of the widget
>> causing the event, and uses CALL_METHOD to call my ReDraw method on
>> the object (David Fanning credits Stein Vidar for this idea)
>
> Really!? I guess I thought I had "invented" it myself.
> Maybe I re-discovered it independently after Stein Vidar
> showed it to me and I forgot all about it. :-(

Well perhaps I was reading too much detail into your crediting?
To quote you from Date: Tue, 9 May 2000 14:24:51 -0600:
; All my events go to one giant event handler. (I learned this
; from Stein Vidar. :-) And I write the event handler like this:
; pro handle_all_events, event

>
>> It's a little frustrating but I haven't a clue on how to resolve it.
>> Any other ideas?
>
> Not really, but I have risked $10 in favor of a programmer error
> rather than a software bug. :-)

I'll bet $10 on that too. We just have to find someone who'll bet the
other way.

> Do you happen to be in Boulder, Sean? I'm not sure
> why I think that, but I'm going to be up that way
> tomorrow. I'd be curious to see what this looked like
> in real life.

Sorry to say I'm not... I'm in Orange County, south of LA.

About your catalyst library, what is it? (The only catalyst library I
could find on google was described as "Tools for Encouraging the
Growth of Indigenous Hymnody" in ethnomusicology, which doesn't sound
right.) Is it a set of Object wrappers for IDL widget functions or
some such? And is it publicly available?

Cheers,
Sean

Subject: Re: explicit redraw does nothing until expose(?) event - IDLitComponent propertysheets
Posted by JD Smith on Fri, 23 Apr 2004 18:21:07 GMT
View Forum Message <> Reply to Message

On Thu, 22 Apr 2004 21:30:40 -0600, David Fanning wrote:

> Sean Dettrick writes:
>
>> The handle_all_events routine inspects the UVALUE of the widget
>> causing the event, and uses CALL_METHOD to call my ReDraw method on
>> the object (David Fanning credits Stein Vidar for this idea)
>
> Really!? I guess I thought I had "invented" it myself.
> Maybe I re-discovered it independently after Stein Vidar
> showed it to me and I forgot all about it. :-(

Wait one minute... I thought *I* invented it. Call the patent attorneys. Actually, it's a fairly obvious extension of the object event callback. Another level of abstraction that I find useful is to give individual widgets a UVALUE that is just a method name, or "something else", which I call an "action". So then the event handler can do one of several things:

1. Re-write the action based on some condition.
2. Handle the action directly itself.
3. Let the action fall through to a method call.

The event handlers look in skeleton form like:

```
pro object_event, ev
  widget_control, ev.top, get_uvalue=self
  self->Event,ev
end
```

Notice how the self object is retrieved from the top level base, not the widget itself...

```
pro Event, ev
 widget_control, ev.id, get_uvalue=action

 ;; Example action rewrite
 if ev.clicks eq 2 then action='viewrecord'

 case action of
   'someaction': print,'Got some action'
   'save-as': self->Save,/AS
    ...
   else: call_method,action,self ;all others, just call the named method
```

```
 endcase
end
```

and when you're setting up the widgets:

```
b1=widget_button(base,VALUE='Save Project...',uvalue='save')
...
widget_control, base,SET_UVALUE=self,/REALIZE
 XManager,'MyObject',base,/NO_BLOCK,EVENT_HANDLER='object_eve nt'
```

The nice thing about this system: you can decide whether to handle the event in-place, or farm it out to a method, and multiple different widgets can trivially call the same method. Each widget only needs to know its "action" and doesn't need a copy of the self object (which is just stuck in the TLB).

JD

Subject: Re: explicit redraw does nothing until expose(?) event - IDLitComponent propertysheets
Posted by sdettrick on Fri, 23 Apr 2004 21:44:16 GMT
View Forum Message <> Reply to Message

Problem Solved!

Thanks for your efforts, but I think what we all missed was that I had not registered the pop-up widget with Xmanager. I take it that is what I'm supposed to do. It seems to do the trick. I didn't forget to do it, rather I simply didn't know about it, this being my first popup widget (actually my second - the first had the same problem).

I suppose then that the events were generated, but were never processed until some action was taken with a registered widget (such as an expose event).

There is now an extra line in my PropertySheet object method:

```
>  pro myLegend::PropertySheet, group_leader=group_leader, $
>                               window=window,$
>                               view=view
>  self.window=window
>  self.view=view
>  wpopup = widget_base( /FLOATING, $
>                        GROUP_LEADER=GROUP_LEADER $
>                      )
>  button = WIDGET_BUTTON( wpopup, $
>                          value='Re-draw window', $
```

```
>                     event_pro='handle_all_events', $
>                     UVALUE = {object:self, $
>                             method:'ReDraw'} $
>                    )
>  result = WIDGET_PROPERTYSHEET( wpopup, $
>                     VALUE = self , $
>                     event_pro = 'handle_all_events', $
>                     UVALUE = {object:self, $
>                             method:'ChangeProperty'} $
>                    )
   xmanager, 'PropertySheet', $
           wpopup, $
           event_handler = 'handle_all_events', $
           /NO_BLOCK
>  widget_control, wpopup, /realize
>  end
```

Putting this extra line of code in was a wild idea for me which came from the blue. Please let me know if it's not what I'm supposed to be doing.

All the best,
Sean

Subject: Re: explicit redraw does nothing until expose(?) event - IDLitComponent propertysheets
Posted by David Fanning on Sat, 24 Apr 2004 00:34:47 GMT
View Forum Message <> Reply to Message

Sean Dettrick writes:

> About your catalyst library, what is it? (The only catalyst library I
> could find on google was described as "Tools for Encouraging the
> Growth of Indigenous Hymnody" in ethnomusicology, which doesn't sound
> right.)

Well, all the obvious names were taken. :-(

> Is it a set of Object wrappers for IDL widget functions or
> some such?

It is an object library in for building applications with graphical user interfaces. One of the things it does is make objects of all the IDL widgets (except the table widget, naturally). We think of it as an application framework.

> And is it publicly available?

I don't know. Maybe. No, probably not. It would mean I'd have to write some documentation for it, and that really does not sound like something I want to do. I use it to build client applications. It is not hard to understand (say, the way other well know object libraries are nearly impossible to understand), but it requires some hand-holding up front (in leu of written instructions). People have used it to build some amazing applications in an unbelievably (to me, anyway) short time.

Cheers,

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