

---

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](mailto: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?
-