
Subject: Re: setting widget " EVENT_PRO='myClass::myCallback' " ??? Is it possible?

Posted by [Lajos Foldy](#) on Mon, 15 Apr 2013 18:00:43 GMT

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

Hi,

IDL calls the event handler with one positional parameter, event.

PRO jkcm_window_level_widget_wwEventHandler, event

one positional parameter: event

PRO jkcm_window_level_widget::wwEventHandler, event

two positional parameters: this (implicit) and event

regards,
Lajos

On Monday, April 15, 2013 7:39:55 PM UTC+2, Justin wrote:

> Hi all,

>

>

>

> I have been unable to set EVENT_PRO to a class method. My current work around is to create a dummy 'global' standalone function at the top of the *__DEFINE.pro file that pulls the object data and then calls the method on the object.

>

>

>

> I was curious if it is even possible to set a widget event handler to a class method. What is the rationale behind not letting an event handler be a class method? Perhaps it doesn't make sense to do it...

>

>

>

>

>

> For example,

>

> below is a window/level class that ends up modifying the window/level on a display object that I create. I've been using it for medical imaging.

>

>

>

> -----

```

>
> PRO jkcm_window_level_widget_wwEventHandler, event
>
> widget_control, event.top, get_uvalue=obj
>
> obj->wwEventHandler, event
>
> END
>
> .....
> PRO jkcm_window_level_widget_wlEventHandler, event
>
> widget_control, event.top, get_uvalue=obj
>
> obj->wlEventHandler, event
>
> END
>
> .....
> PRO jkcm_window_level_widget::CREATE_WIDGETS, PARENT_WIDGET =
inp_PARENT_WIDGET
>
>
>
>
>
> self.parent_widget = (N_ELEMENTS(inp_PARENT_WIDGET) NE 0) ? inp_PARENT_WIDGET
: WIDGET_BASE(COLUMN=2, YSIZE=512, TLB_FRAME_ATTR=2)
>
>
>
> ;I would like to be able to say:
>
> ; EVENT_PRO='jkcm_window_level_widget::wwEventHandler'
>
> self.wwSlider = WIDGET_SLIDER(self.parent_widget, UNAME='WW_widget', YSIZE=512,$
>
> TITLE='WW', /VERTICAL, EVENT_PRO='jkcm_window_level_widget_wwEventHandler', $
>
> MINIMUM=0, MAXIMUM=4095, /DRAG)
>
>
>
> self.wlSlider = WIDGET_SLIDER(self.parent_widget, UNAME='WL_widget', YSIZE=512,$
>
> TITLE='WL', /VERTICAL, EVENT_PRO='jkcm_window_level_widget_wlEventHandler', $

```

```

>
> MINIMUM=-1024, MAXIMUM=4095, /DRAG)
>
>
>
> END ;jkcm_window_level_widget::CREATE_WIDGETS
>
> .....
> .....
> ...
>
>
>
> PRO jkcm_window_level_widget::wwEventHandler, event
>
>
>
> val="
>
> widget_control, self.wwSlider, GET_VALUE=val
>
> self.svo->SET_WINDOW_LEVEL, WIDTH=val
>
>
>
> END ;jkcm_window_level_widget::wwEventHandler
>
> .....
> .....
>
>
> PRO jkcm_window_level_widget::wlEventHandler, event
>
>
>
> val="
>
> widget_control, self.wlSlider, GET_VALUE=val
>
> self.svo->SET_WINDOW_LEVEL, LEVEL=val
>
>
>
> END ;jkcm_window_level_widget::wlEventHandler
>
> .....
> .....
>
> FUNCTION jkcm_window_level_widget::INIT, SVO=inp_SVO

```

```

>
>
>
> self.svo = (N_ELEMENTS(inp_svo) NE 0) ? inp_svo : !NULL
>
>
>
> self->CREATE_WIDGETS
>
> RETURN, 1
>
> END
>
> .....
> ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
>
> PRO jkcm_window_level_widget::CLEANUP
>
>
>
> widget_control, self.parent_widget, /destroy
>
>
>
> END
>
> .....
> ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
>
> PRO jkcm_window_level_widget__DEFINE
>
>
>
>
>
> struct = { jkcm_window_level_widget , $
>
> parent_widget:0L,$ ;'top' level widget
>
> wwSlider:0L, $; widget slider for window width
>
> wlSlider:0L, $; widget slider for window level
>
>
>
> svo:OBJ_NEW(), $;reference to single_view_obj_graphics object (dont delete)
>
>
>
>
> ww: double(1), $ ;window width

```

```
>  
>   wl: double(1) $ ;window length  
>  
> }  
>  
> END ;END jkcm_window_level_widget__DEFINE
```
