
Subject: Re: IDL Runtime

Posted by [R.Bauer](#) on Thu, 06 Dec 2001 17:49:29 GMT

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

Dear Andre, dear David

I am sorry that's I have not spent time last three weeks to read more detailed this issue

Attached is a small example from our IDL widget lessons which shows a very nice functionality about starting an event while an event is already running.

You have not to wait until UP is finished, you can directly press DOWN or QUIT.

The trick for this is to call always as event_pro set by the widget definition "wid1_event".

In the cases of 'wid1_UP' and 'wid1_DOWN' always a second own event manager

"wid1_own_event" is started which did nothing else as asking for events from the buttons.

If an event by a button is returned the event procedure "wid1_event" is runnind independent from the XMANAGER.

This means the "wid1_event" runs many times started like child-processes and only

if you have used pointers you know the variables in each process and they are the same.

If you press "Quit" all of this child-processes of "wid1_event" are stopped at once.

I have many of my widgets programmed in this way, but it is hard to reprogramm an existing one. Sometimes it is enough to give only a few buttons these functionality as an addition.

If I should explain something in more detail please give me a note.

David if you like you can add this widget to your tips page.

regards

Reimar

>
> So again, the design involves having options initially (menu). Some options
> aren't relevant initially such as "Save" and "Print", so if the user clicks on
> these I issue messages for the user telling them there's no point clicking
> these ones yet. The most likely thing they will want to do initially is
> "Analysis". "Analysis" is the button with the event handler I described above:
> programs that just simply get executed one after the other. During execution of
> the event-handler programs, the menu buttons can't be clicked, mainly due to
> the fact that the region drawing/moving uses the CURSOR command - so the
> computer is waiting for mouse clicks inside the graphics window. (You might
> prefer draw widgets w/o CURSOR etc etc, but there's really no problem I can see
> with doing it the way I have. It essentially creates MODAL functionality which
> ensures the user gets to the end. Anyway I think this is beside the point). But
> after this part is done and the results have been displayed, that's the end of
> "Analysis". Now they might want to print the results... or do another
> analysis...etc.
>
> Also, exactly what DOES happen when instead, the user doesn't get fed up,
> finishes the whole event handling routine, and we arrive at the "END" command
> at the conclusion of the event-handler? Where is program execution? What is
> happening? I'm thinking that we're ready for another menu-button press??
>
> Would appreciate any guidance on a structure,
> Thanks,
> Andre Kyme

--
Reimar Bauer

Institut fuer Stratosphaerische Chemie (ICG-1)
Forschungszentrum Juelich
email: R.Bauer@fz-juelich.de
<http://www.fz-juelich.de/icg/icg1/>

=====
a IDL library at Forschungszentrum Juelich
http://www.fz-juelich.de/icg/icg1/idl_icglib/idl_lib_intro.html

<http://www.fz-juelich.de/zb/text/publikation/juel3786.html>
=====

read something about linux / windows
<http://www.suse.de/de/news/hotnews/MS.html>
PRO wid1_own_event,event

```

IF WIDGET_INFO(event.top,/active) THEN $
  a=WIDGET_EVENT(WIDGET_INFO(event.top,find_by_undef='wid1_DOW
N'),/nowait,bad_id=bad)
  IF WIDGET_INFO(event.top,/active) THEN $
    a=WIDGET_EVENT(WIDGET_INFO(event.top,find_by_undef='wid1_UP'
),/nowait,bad_id=bad)
  IF WIDGET_INFO(event.top,/active) THEN $
    a=WIDGET_EVENT(WIDGET_INFO(event.top,find_by_undef='wid1_QUI
T'),/nowait,bad_id=bad)

END

```

```

PRO wid1_event, Event
  WIDGET_CONTROL,event.top,get_uvalue=values
  CASE event.id OF
    WIDGET_INFO(event.top,find_by_undef='wid1_QUIT'): BEGIN
      (*values).is_up =0
      (*values).is_down =0
      WIDGET_CONTROL, event.top,/destroy
    END
    WIDGET_INFO(event.top,find_by_undef='wid1_UP'): BEGIN
      (*values).is_up =1
      (*values).is_down =0
      IF event.select EQ 1 THEN BEGIN
        WIDGET_CONTROL,WIDGET_INFO(event.top,find_by_undef='wid1_DRA
W'),get_uvalue=WINDOW
        WSET,WINDOW
        WHILE (*values).counter LE (*values).max AND $
          (*values).is_up EQ 1 DO BEGIN
            ERASE
            XYOUTS,0.5,0.5,/NORM,'UP',alignment=0.5, $
              charsize=(*values).counter/100.
            (*values).counter=(*values).counter+1
            wid1_own_event,event
          ENDWHILE
        ENDIF
      END
    END
  END

```

```

WIDGET_INFO(event.top,find_by_undef='wid1_DOWN'): BEGIN
  (*values).is_up =0
  (*values).is_down =1
  IF event.select EQ 1 THEN BEGIN
    WIDGET_CONTROL,WIDGET_INFO(event.top,find_by_undef='wid1_DRA
W'),get_uvalue=WINDOW
    WSET,WINDOW
    WHILE (*values).counter GT (*values).min AND $

```

```

(*values).is_down EQ 1 DO BEGIN
  ERASE
  XYOUTS,0.5,0.5,/NORM,'DOWN',alignment=0.5, $
    charsize=(*values).counter/100.
  (*values).counter=(*values).counter-1
  wid1_own_event,event
ENDWHILE
ENDIF
END
ELSE:
ENDCASE
END

PRO wid1, GROUP_LEADER=wGroup, _EXTRA=_VWBExtra_
  ID_BASE_0=WIDGET_BASE( GROUP_LEADER=wGroup,$
    SCR_XSIZE=300 ,SCR_YSIZE=200,TITLE='IDL',$
    SPACE=3 ,XPAD=3,YPAD=3)

  ID_DRAW_0=WIDGET_DRAW(id_base_0,$
    XOFFSET=100 ,YOFFSET=7, SCR_XSIZE=185 ,$
    SCR_YSIZE=160,uname='wid1_DRAW')

  ID_UP=WIDGET_BUTTON(id_base_0, $
    XOFFSET=44 ,YOFFSET=60,XSIZE=50,/ALIGN_CENTER ,$
    VALUE='UP',event_pro='wid1_event',uname='wid1_UP')

  ID_DOWN=WIDGET_BUTTON(id_base_0, $
    XOFFSET=44, YOFFSET=83, XSIZE=50, $
    /ALIGN_CENTER, VALUE='DOWN',$
    event_pro='wid1_event',uname='wid1_DOWN')

  ID_QUIT=WIDGET_BUTTON(id_base_0, $
    /ALIGN_CENTER ,VALUE='QUIT', $
    event_pro='wid1_event',uname='wid1_QUIT')

  WIDGET_CONTROL, /REALIZE, id_BASE_0

  WIDGET_CONTROL,id_draw_0,get_value=window_no
  WIDGET_CONTROL,id_draw_0,set_uvalue=window_no
  WSET>window_no
  ERASE

  values=PTR_NEW(CREATE_STRUCT('counter',0L,'min',1L,'max',100 00L,'is_up',0,$
    'is_down',0))

```

```
WIDGET_CONTROL, /REALIZE, id_BASE_0, set_uvalue=values
```

```
XMANAGER, 'wid1', id_BASE_0
```

```
PTR_FREE, values
```

```
END
```

File Attachments

1) [wid1.pro](#), downloaded 217 times
