
Subject: Re: Radio Buttons OnButtonPress Bug
Posted by [Benjamin Hornberger](#) on Sun, 28 Nov 2004 22:57:40 GMT
[View Forum Message](#) <> [Reply to Message](#)

Mike,

you get one event for deselecting the previously selected button, and a second event for selecting the new one. You should put something like

IF event.select EQ 0 THEN return

at the beginning of your event handler. Have a look at "Events Returned by Button Widgets" in the online help for widget_button.

Benjamin

Mike wrote:

> Has anyone else encountered this?
>
> With a set of 2 or more radio buttons, each assigned the same
> procedure for OnButtonPress, when you select one of them the procedure
> is run twice.
>
> Note... when you select one of them for the first time (when none of
> them are selected and you click one for the first time) it only runs
> the procedure once, but then whenever a new button is selected, the
> procedure is run twice.
>
> This seems to be a bug in IDL's code execution.
>
> Try it out for yourself, though, and let me know what you think.
> Here's an example program and eventcb file:
>
> radiobttntest.pro:
> -----
>
> ;
> ; IDL Widget Interface Procedures. This Code is automatically
> ; generated and should not be modified.
>
> ;
> ; Generated on: 11/28/2004 15:56.44
> ;
> pro WID_BASE_0_event, Event
>
> wTarget = (widget_info(Event.id,/NAME) eq 'TREE' ? \$
> widget_info(Event.id, /tree_root) : event.id)
>

```

>
> wWidget = Event.top
>
> case wTarget of
>
>   Widget_Info(wWidget, FIND_BY_UNAME='wRadioBtn1'): begin
>     if( Tag_Names(Event, /STRUCTURE_NAME) eq 'WIDGET_BUTTON' )then $
>       RadioPress, Event
>     end
>   Widget_Info(wWidget, FIND_BY_UNAME='wRadioBtn2'): begin
>     if( Tag_Names(Event, /STRUCTURE_NAME) eq 'WIDGET_BUTTON' )then $
>       RadioPress, Event
>     end
>   Widget_Info(wWidget, FIND_BY_UNAME='wRadioBtn3'): begin
>     if( Tag_Names(Event, /STRUCTURE_NAME) eq 'WIDGET_BUTTON' )then $
>       RadioPress, Event
>     end
>   else:
> endcase
>
> end
>
> pro WID_BASE_0, GROUP_LEADER=wGroup, _EXTRA=_VWBExtra_
>
>   Resolve_Routine, 'radiobttntest_eventcb',/COMPILE_FULL_FILE ; Load
> event callback routines
>
>   WID_BASE_0 = Widget_Base( GROUP_LEADER=wGroup, UNAME='WID_BASE_0' $
>     ,XOFFSET=5 ,YOFFSET=5 ,SCR_XSIZE=300 ,SCR_YSIZE=200 $
>     ,TITLE='IDL' ,SPACE=3 ,XPAD=3 ,YPAD=3)
>
>
>   WID_BASE_1 = Widget_Base(WID_BASE_0, UNAME='WID_BASE_1' ,XOFFSET=30
> $
>     ,YOFFSET=55 ,TITLE='IDL' ,ROW=1 ,/EXCLUSIVE)
>
>
>   wRadioBtn1 = Widget_Button(WID_BASE_1, UNAME='wRadioBtn1' $
>     ,/ALIGN_LEFT ,VALUE='Button1')
>
>
>   wRadioBtn2 = Widget_Button(WID_BASE_1, UNAME='wRadioBtn2' $
>     ,/ALIGN_LEFT ,VALUE='Button2')
>
>
>   wRadioBtn3 = Widget_Button(WID_BASE_1, UNAME='wRadioBtn3' $
>     ,/ALIGN_LEFT ,VALUE='Button3')
>

```

```
> Widget_Control, /REALIZE, WID_BASE_0
>
> XManager, 'WID_BASE_0', WID_BASE_0, /NO_BLOCK
>
> end
> ;
> ; Empty stub procedure used for autoloading.
> ;
> pro radiobttntest, GROUP_LEADER=wGroup, _EXTRA=_VWBExtra_
> WID_BASE_0, GROUP_LEADER=wGroup, _EXTRA=_VWBExtra_
> end
>
> -----
>
>
> radiobttntest_eventcb.pro:
> -----
>
> pro RadioPress, Event
>
>     print, "hello"
>
> end
> ;
> ; Empty stub procedure used for autoloading.
> ;
> pro radiobttntest_eventcb
> end
>
> -----
>
>
> -Mike
```
