
Subject: Problems with pull-down menus in Motif
Posted by [thompson](#) on Mon, 29 Mar 1993 22:38:56 GMT
[View Forum Message](#) <> [Reply to Message](#)

I ran into a mysterious problem with a widget routine that worked fine in OpenLook on a Sun workstation, but didn't work in Motif on a VAX/VMS machine. Basically, I have a button defined as a pull-down menu which I can't make pull down the menu. The simplified routine below demonstrates the problem. This works fine on the Sun, but not on the VAX.

I can't figure it out because I've got another button widget with basically the same code that works fine on both platforms. However, I've tracked the problem down to the fact that the base widget is defined as a column widget. If the pull-down menu button is defined within a row widget base, then everything's fine.

Am I missing something, or is this a real bug?

Bill Thompson

```
*****
;
; *****
; ***** ***** ***** ***** ***** *****
; *****
; *****
;
PRO TEST_EVENT, EVENT
;
; Event handler for the TEST widget routine.
;
COMMON COLORS, R_ORIG, G_ORIG, B_ORIG, R_CURR, G_CURR, B_CURR
COMMON TEST, LEFT, RIGHT, VEL, R0, G0, B0, R1, G1, B1
;
; Get the widget event.
;
WIDGET_CONTROL, EVENT.ID, GET_UVALUE = EVENTVAL
;
; Process the event. If the top slider was moved, then change the top value,
; and stretch the color table accordingly. If the top and bottom sliders are
; locked together, then also move the bottom slider.
;
CASE EVENTVAL OF
;
"BLUE": BEGIN
VEL.VALUE = 0
GOTO, SET_COLOR
END
"LIGHTBLUE": BEGIN
VEL.VALUE = 1
GOTO, SET_COLOR
END
```

```

"GREEN": BEGIN
  VEL.VALUE = 2
  GOTO, SET_COLOR
END
"TURQUOISE": BEGIN
  VEL.VALUE = 3
SET_COLOR:
  FOR I = 0,3 DO WIDGET_CONTROL, VEL.COLOR(I), $
    SENSITIVE = I NE VEL.VALUE
  GOTO, LOAD_VELOCITY
END
;
;
; Load the velocity color table.
;
;
"VELOCITY": BEGIN
LOAD_VELOCITY:
  LEFT.VEL = 1
  GOTO, SET_VELOCITY
END
;
;
; Define the current table as being either a velocity or intensity color
; table.
;
;
"SETVEL": BEGIN
  LEFT.VEL = 1 - LEFT.VEL
SET_VELOCITY:
  WIDGET_CONTROL, VEL.OPTIONS, SENSITIVE=LEFT.VEL
  GOTO, DO_STRETCH
END
;
;
; Load one of the standard color tables.
;
;
ELSE: BEGIN
DO_STRETCH:
  END
ENDCASE
;
;
END
.*****
;
. ****  ****  ****  ****  ****  ****  ****  ****  ****  ****
;
.*****
;
PRO TEST, SILENT=SILENT, GROUP=GROUP, NOLOAD=NOLOAD
;
;
COMMON COLORS, R_ORIG, G_ORIG, B_ORIG, R_CURR, G_CURR, B_CURR
COMMON TEST, LEFT, RIGHT, VEL, R0, G0, B0, R1, G1, B1
;
;
; If a copy of TEST is already running, then exit.
;
;

```

```

IF (XREGISTERED("test") NE 0) THEN GOTO, EXIT_POINT
;
; Initialize the common block.
;
LEFT = {TEST, VEL: 0, $
TOP: 0L, $
BOT: 0L, $
GAMMA: 0L, $
G_LBL: 0L, $
LOCK: 0, $
CHOP: 0, $
GAMMATYPE: 0, $
LOCK_BUTTON: [0L,0L], $
CHOP_BUTTON: [0L,0L], $
GAMMA_BUTTON: [0L,0L], $
SETV_BUTTON: [0L,0L], $
SILENT: 0, $
GROUP: 0}
VEL = {TEST_VEL, OPTIONS: 0L, VALUE: 0, COLOR: [0L, 0L, 0L, 0L]}
;
; Save the value of the GROUP keyword in the common block.
;
IF N_ELEMENTS(GROUP) NE 0 THEN LEFT.GROUP = GROUP
;
; Define some defaults.
;
W_HEIGHT = 50 ;Height of ramp
CUR_WIN = !D.WINDOW
;
; Define the widget base.
;
BASE = WIDGET_BASE(TITLE="Test", /COLUMN)
;
; Define the individual widget components. Start with the graphics window for
; displaying the color table, the "DONE" and "HELP" buttons.
;
SHOW = WIDGET_DRAW(BASE, YSIZE=W_HEIGHT, XSIZE=256, /FRAME, RETAIN = 2)
;
; Define the button for loading the velocity table.
;
JUNK = WIDGET_BUTTON(BASE, VALUE='Load Velocity Table', $
UVALUE="VELOCITY")
;
; Define the velocity options.
;
VEL.OPTIONS = WIDGET_BUTTON(BASE, VALUE=' Velocity Options ', /MENU)
VEL.COLOR(0) = WIDGET_BUTTON(VEL.OPTIONS, VALUE='Blue', UVALUE="BLUE")
VEL.COLOR(1) = WIDGET_BUTTON(VEL.OPTIONS, VALUE='Light Blue', $

```

```

    UVALUE="LIGHTBLUE")
VEL.COLOR(2) = WIDGET_BUTTON(VEL.OPTIONS, VALUE='Green', $
    UVALUE="GREEN")
VEL.COLOR(3) = WIDGET_BUTTON(VEL.OPTIONS, VALUE='Turquoise', $
    UVALUE="TURQUOISE")
;
;
; Desensitize the velocity options, and the blue color button.
;
WIDGET_CONTROL, VEL.OPTIONS, SENSITIVE = 0
WIDGET_CONTROL, VEL.COLOR(0), SENSITIVE = 0
;
; Realize the widget.
;
WIDGET_CONTROL, BASE, /REALIZE
;
; Show the current color table in the graphics widget.
;
WIDGET_CONTROL, SHOW, GET_VALUE=SHOW_WIN
WSET, SHOW_WIN
TVSCL, INDGEN(256) # REPLICATE(1, W_HEIGHT)
IF (CUR_WIN NE -1) THEN WSET, CUR_WIN
;
; Start the widget manager.
;
XMANAGER, "test", BASE, GROUP_LEADER = GROUP
;
EXIT_POINT:
END

```
