
Subject: Re: Keeping Button Pressed In?

Posted by [Dick Jackson](#) on Mon, 24 Oct 2005 04:39:53 GMT

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

Hi Benjamin:

"Benjamin Hornberger" <benjamin.hornberger@stonybrook.edu> wrote in message
news:43592bf7\$1_1@marge.ic.sunysb.edu...

>

> Interesting. I find it inconsistent though that bitmap buttons behave like
> that, while buttons with a text value (label) turn into radio buttons in
> an exclusive base. What if I want a "depressed" button with a text value?
>

> Benjamin

I've found this to work pretty well, a function that takes the text value as
a string and converts it to the RGB byte array for your Widget_Button. I
realize it's not perfect about aligning one button with no descenders (e.g.,
'GJPY') and another with descenders (e.g., 'gjpy'), but I don't have time to
make that very doable fix right now.

Hope this is of help to someone:

=====

FUNCTION BitmapForButtonText, str

;; Return an RGB byte array (w, h, 3) suitable for use as the Value of a
;; Widget_Button to display the text given in parameter 'str'.

;; Example:

```
;; wTLB0=Widget_Base(/Row,/Exclusive)
;; wBtn1=Widget_Button(wTLB0,Value=BitmapForButtonText('1'))
;; wBtn2=Widget_Button(wTLB0,Value=BitmapForButtonText('2'))
;; Widget_Control,wTLB0,/Realize
```

;; Dick Jackson / D-Jackson Software Consulting / dick@d-jackson.com

```
wTLB = Widget_Base()
wBtn = Widget_Button(wTLB)
font = Widget_Info(wBtn, /FontName)
sysColors = Widget_Info(wBtn, /System_Colors)
xs = 200 ; Maximum width of button text
ys = 40
x0 = 3
y0 = 6
border = 3
Window, XSize=xs, YSize=ys, /Pixmap, /Free
```

```

Erase, Color=Total(sysColors.Face_3D * [1, 256, 65536L])
blankRGB = TVRD(True=3)
Device, Set_Font=font
XYOutS, x0, y0, str, /Device, Font=0, $
    Color=Total(sysColors.Button_Text * [1, 256, 65536L])
textRGB = TVRD(True=3)
WDelete, !D.Window
text2D = Total(textRGB NE blankRGB, 3)
whereX = Where(Total(text2D, 2) NE 0, nWhereX)
whereY = Where(Total(text2D, 1) NE 0, nWhereY)

IF nWhereX * nWhereY EQ 0 THEN result = blankRGB[0, 0, *] $
ELSE result = textRGB[(whereX[0]-border): $
    (whereX[nWhereX-1]+border) < (xs-1), $
    (whereY[0]-border): $
    (whereY[nWhereY-1]+border), *]

IF !Order EQ 1 THEN result = Reverse(result, 2)

Return, result

END

=====

--
Cheers,
--
-Dick

```

Dick Jackson / dick@d-jackson.com
 D-Jackson Software Consulting / http://www.d-jackson.com
 Calgary, Alberta, Canada / +1-403-242-7398 / Fax: 241-7392
