
Subject: Looking for CW_BSELECTOR like widget
Posted by [doetzl](#) on Mon, 14 Mar 1994 16:45:11 GMT

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

CW_BSELECTOR is a compound widget that appears as a pull-down menu whose label shows the widget's current value. When the button is pressed, the menu appears and the newly selected value becomes the new title of the pull-down menu.

I'd like the same basic behavior, except that when the menu appears I'd like it to have a scroll bar. My pull-down menus could potentially have several (>30) values; when there are this many the menu is so long that the items at the bottom of the menu go off of the screen.

Or alternatively I guess I could have a button that pops up a list widget, that in turn changes the value of the button to the value selected in the list, but this is awkward. I could use just a list, but a single button makes the overall appearance of the application less cluttered. Any suggestions?

--Joe

--

--Joe

|| Joe Doetzl | Climate Modeling Section ||
|| doetzl@pendragon.cgd.ucar.edu | National Center for Atmospheric Research ||

Subject: Re: Looking for CW_BSELECTOR like widget
Posted by [stl](#) on Tue, 15 Mar 1994 07:16:26 GMT

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

In article <2m24an\$enh@ncar.ucar.edu> doetzl@pendragon.cgd.ucar.edu (Joe Doetzl) writes:

> I'd like the same basic behavior, except that when the menu appears I'd
> like it to have a scroll bar. My pull-down menus could potentially
> have several (>30) values; when there are this many the menu is so long
> that the items at the bottom of the menu go off of the screen.
> Or alternatively I guess I could have a button that pops up a list widget,
> that in turn changes the value of the button to the value selected in the
> list, but this is awkward. I could use just a list, but a single button
> makes the overall appearance of the application less cluttered. Any
> suggestions?

>
hi,

This is really a problem. First, I would suggest looking in an interface style guide (Motif OSF style guide if you are using motif) because this will tell you or point you in the right direction of what kind of widget you should use. However this type of widget may not exist in IDL (I am guessing a dropdown-listbox would work very well).

It seems that a Menu is not the right thing to use here, just because of the size, and the fact that a menu should be pretty static and change only if the view of the application changes. How about a scrollable list box? (can you build a scrollable base widget with buttons on it? I don't think so, but I have seen this in OS/2) Also, using this many buttons is kind of poor, because again you want to limit the amount of choices (not to mention the space problem) You could be creative and link the focus of a entry field with the displaying of a little box window, and once the selection is made, have it fill the entry field (but again this isn't very standard, but might serve your needs well)

I guess these are all just ideas you can throw around, hope they help.

-stephen

PS: does anyone know if you can get directly at Motif from IDL. A little while ago, I seem to remember someone saying that they opened a dialog box or something.. I would love to see/hear more about this. Any any information would be helpful.

--

Stephen C Strebel / SKI TO DIE
stl@maz.sma.ch / and
Swiss Meteorological Institute, Zuerich / LIVE TO TELL ABOUT IT
01 256 93 85 / (and pray for snow)

Subject: Re: Looking for CW_BSELECTOR like widget
Posted by [zawodny](#) on Tue, 15 Mar 1994 13:03:07 GMT
[View Forum Message](#) <> [Reply to Message](#)

In article <2m24an\$enh@ncar.ucar.edu> (Joe Doetzi) writes:
> I'd like the same basic behavior, except that when the menu appears I'd
> like it to have a scroll bar. My pull-down menus could potentially
> have several (>30) values; when there are this many the menu is so long
> that the items at the bottom of the menu go off of the screen.
> Or alternatively I guess I could have a button that pops up a list widget,
> that in turn changes the value of the button to the value selected in the
> list, but this is awkward. I could use just a list, but a single button
> makes the overall appearance of the application less cluttered. Any
> suggestions?
>

Yes I have run into this every now and again as well. It seems to be more of a problem on my laptop (uses proportionally larger fonts relative to the screen size). Anyway, I'll append to this message a widget that appears initailly as a standard widget_button and when activated, creates a pop-up

widget comprised of an ARRAY of widget_buttons. Depending upon the length of the button value strings, you can easily fit well over 100 buttons in one of these widgets. The code is pretty fresh and has been tested for basic functionality. It is, however, in a pre-release form and does not have all of the standard error traps and such that I normally include in my programs. Since time is usually of the essence, I'll post it anyway.

Don't forget to crop off the .signature file at the end of this posting.

```
function WBUT_ARRAY_EVENT,event
;+
; Event handler for WBUT_ARRAY_EVENT widget. Creates a pop-up button array
; and returns the selected button value via an event structure.
; If you are familiar with widgets then you can figure out what is going on
; here.
;
; Written 03/10/94 by J M Zawodny, NASA LaRC
;-

; Get the array of button values
widget_control,event.id,get_uvalue=inf,/no_copy

; Give it a base
root = widget_base(title=inf.title,group=event.top)
wid = widget_base(root,column=inf.col,row=inf.row)

; For every button
for k=0,n_elements(inf.values)-1 do begin
; Format the value and make a button
if(inf.format eq "") then val = inf.values(k) $
else val = string(inf.values(k),form=inf.format)
but = widget_button(wid,value=val)
endfor

; Realize the Pop-up widget
widget_control,root,/realize
wevent = widget_event(root)

; Get the result, store it in the structure, and then destroy the widget
widget_control,wevent.id,get_value=answer
inf.result = answer
widget_control,root,/destroy

; Return the state info to the widget
widget_control,event.id,set_uvalue=inf,/no_copy

; Make an event structure to return
return,event
```

```

end
; =====

function WBUT_ARRAY(base,value=value,uvalue=uvalue
;+
; Routine to generate an array of buttons widget
; I have not finished documenting this, but since it may be of use to you
; I'll post it a bit earlier than I normally would.
;
; Basically you configure the widget by setting the uvalue of this widget
;
; id = WBUT_ARRAY(base,value=value,uvalue=uvalue)
;
; BASE is the id of the parent
; VALUE is a string used to label the button that activates the
; pop-up button array widget
;
; UVALUE is a structure of the form:
; {values:[0], row:0, col:0, title:'Pop-Up', format:'(i1)',result:''}
;
; Where the field:
; VALUES is an array of values to assign to the buttons
; ROW is non-zero when it specifies the number of rows of buttons
; COL is non-zero when it specifies the number of columns of buttons
; TITLE is the title for the header bar in the pop-up widget
; FORMAT is a format string used to format the button values
; RESULT is for internal use only and will be obliterated under normal
; conditions
;
; Don't set both ROW and COL to non-zero values. I have no idea what will
; happen and I have not implemented the code to trap this condition yet.
;
; The event handler uses widget_event and so if you activate the pop-up button
; array widget, you will have to select a button before you can continue on.
;
; Written 03/10/94 by J M Zawodny, NASA LaRC
;-
; Set defaults
if(n_elements(value) eq 0) then value = 'None'
; This should never use this default
if(n_elements(uvalue) eq 0) then uvalue = {values:[0], row:0, col:0 $
, title:'Pop-Up', format:'(i1)',result:''}

; Make the button
wid = widget_button(base,value=value,uvalue=uvalue $
,event_func='wbut_array_event')

; Return the pointer to the base

```

return,wid

end

; =====

--

Joseph M. Zawodny (KO4LW)

NASA Langley Research Center

Internet: zawodny@arbd0.larc.nasa.gov

MS-475, Hampton VA, 23681-0001

Packet: ko4lw@n4hog.va.usa
