

---

Subject: Re: dynamic pull down menus, removing items

Posted by [btt](#) on Tue, 13 Jul 2004 18:21:04 GMT

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

---

David Fanning wrote:

> Ben Tupper writes:

>

>

>> I would like to have a 'Window' pull down menu item on my GUI's menu  
>> bar. In theory, this 'Window' would contain a list of image window  
>> names where the number and names of windows can change while the GUI is  
>> running. This menu is simply a way to show what windows are available  
>> and which is the 'current' window.

>>

>> The current selection is easy to show using the CHECKED\_MENU option to  
>> WIDGET\_BUTTON (see [http://www.dfanning.com/widget\\_tips/checkmarks.html](http://www.dfanning.com/widget_tips/checkmarks.html))  
>> (I must say, the check mark has an unexpected appearance on my Mac.)

>

>

> Do you have a picture of this? JPEG maybe. I'll put it  
> in the article so people don't get completely frightened off. :-)

>

>

>> Now, I can add buttons as more windows are opened...  
>> [http://www.dfanning.com/widget\\_tips/dynamic\\_menus.html](http://www.dfanning.com/widget_tips/dynamic_menus.html)

>>

>> But how the heck to I get rid of an item if its window is closed?

>>

>> I have tried

>>

>> WIDGET\_CONTROL, thisParticularButton, /Destroy

>>

>> but it doesn't update the pull down menu. The only way I can get it to  
>> update as I would like is to kill the entire GUI and re-realize it. Yuck.

>

>

> The \*entire\* GUI!? Wow. I wouldn't have thought that was  
> necessary. I think you are going to have to build the whole  
> pull-down menu again, but I'm surprised about taking down  
> the entire GUI. Do you really mean the \*entire\* GUI?

>

>

>> I had the whole shebang in a list widget, but i was hoping to save real  
>> estate by moving the list to a pull down menu.

>

Hi,

Well, it turns out that each button can be removed. Whew!

>  
> Do you have a test program? I'd be curious. :-)

Yes (see below), and it works like it is supposed to. Doesn't that just figure!

Thanks.

Ben

```
*****START
;-----
; Event
;-----
PRO Dynamic_PullDownEvent, ev

Catch, error
If Error NE 0 then begin
  OK = Error_Message(/trace)
  Catch, /cancel
  If n_elements(info) NE 0 then $
    Widget_Control, ev.top, set_Uvalue = info,/no_copy
  Return
EndIf

Widget_Control, ev.top, get_Uvalue = info,/no_copy

;get the number of items in the list
If ptr_Valid(info.pButtons) then $
  n = n_elements(*info.pButtons) else $
  n = 0

Case ev.ID of

info.AddID: Begin

  If n NE 0 Then begin

    ;in this case, the list is occupied by at least one item

    lastnum = -1L
```

```

For i = 0, n-1 do begin
  widget_Control, (*info.pButtons)[i], get_Uvalue = yourNumber
  lastNum = yourNumber > lastNum
EndFor
button = Widget_Button(info.pullDownID, $
  value = 'Numbah ' + StrTrim(lastNum+2,2),$
  Event_pro = 'Dynamic_PullDownEvent', $
  /Checked_menu, $
  uValue = lastNum + 1)
*info.pbuttons = [*info.pButtons, button]

```

EndIf Else Begin

```

;in this case the list is empty

```

```

button = Widget_Button(info.pullDownID, $
  value = 'Numbah 1',$
  /Checked_menu, $
  Event_pro = 'Dynamic_PullDownEvent', $
  uValue = 0)
info.pButtons = Ptr_New(button)
Widget_Control, button, /set_Button

```

EndElse

End ;add an item

info.RemoveID: Begin

```

;are there items in the list?
If n GT 0 then begin

```

```

if n GT 1 Then begin

```

```

ok = intarr(n)
For i = 0, n-1 Do $
  ok[i] = Widget_info((*info.pButtons)[i], /button_set)

```

```

A = Where(ok EQ 1,cnt, comp = comp, ncomp = n)

```

```

if cnt gt 0 then begin

```

```

  RemoveThisID = (*info.pButtons)[A[0]]

```

```

  *info.pButtons = (*info.pButtons)[comp]
  Widget_Control, (*info.pButtons)[0 < (i-1)], /set_Button

```

```
Widget_Control, removeThisID, /destroy
EndIF
```

```
EndIf Else Begin
```

```
    ;there was only one item left in the list anyway
    ;so purge the list
    Widget_Control, (*info.pButtons)[0], /Destroy
    Ptr_free, info.pButtons
```

```
EndElse ; n GT 1
```

```
EndIf ;n GT 0
```

```
End ;remove an item
```

```
Else:Begin
```

```
    ;set the check mark to the selected list item
    For i = 0, n-1 Do $
        Widget_Control, (*info.pButtons)[i], $
        set_button = ((*info.pButtons)[i] EQ ev.ID)
```

```
    End ; pullDown selection change
```

```
EndCase
```

```
Widget_Control, ev.top, set_Uvalue = info,/no_copy
END; PulldownEvent
```

```
;-----
; Cleanup
;-----
PRO Dynamic_Pulldown_Cleanup, tlb
Widget_Control, tlb, get_Uvalue = info
PTR_FREE, info.pButtons
END; Cleanup
```

```
;-----
; main
;-----
PRO Dynamic_Pulldown
```

```

tlb = widget_base(column = 1, $
/base_align_center, $
mBar = menuID)

pullDownID = WIDGET_BUTTON(menuID, $
value = 'PullDown', $
/menu)

buttons = IonArr(3)
For i = 0, 2 do $
buttons[i] = Widget_Button(pullDownID, $
value = 'Nambah ' + StrTrim(i + 1, 2), $
Event_Pro = 'Dynamic_PullDownEvent', $
/Checked_Menu, $
uValue = i)

pButtons = Ptr_New(buttons)
Widget_Control, pullDownID, set_Uvalue = pButtons

addID = widget_Button(tlb, value = 'Add item', $
event_pro = 'Dynamic_PullDownEvent')

removeID = Widget_Button(tlb, value = 'Remove item', $
event_pro = 'Dynamic_PullDownEvent')

Widget_Control, tlb, $
set_Uvalue = {pullDownID: pullDownID, $
pButtons: pButtons, $
addID: addID, $
removeID: removeID}, $
/No_Copy

CenterTLB,tlb
Widget_Control, buttons[0], /set_button
Widget_Control, tlb, /realize

XManager, 'dynamic_pulldown', tlb, $
/No_Block, Cleanup='Dynamic_Pulldown_Cleanup'
END

*****END

```

---