

Subject: multiple selections from a list
Posted by [knipp](#) on Tue, 01 Feb 1994 17:55:32 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi

some days ago I've asked the German distributor of IDL if there exists a widget allowing multiple selections (like `WIDGET_LIST`, but with the possibility to select more than one entry). They could not help me.

So I took the time to write a simple program just doing that. It's appended to this message.

Hope it's useful, Karl

// _ \\ // Karlheinz Knipp phone: +49 511 - 762 4922
 /// / /// University of Hannover fax: +49 511 - 762 2483
 // _ // Institute for Photogrammetry
 /// // Nienburger Str.1
 / // / / / FRG 30167 Hannover 1 email: knipp@ipi.uni-hannover.de

~~~~~ CUT HERE ~~~~~

```
;+
; pro wmx_msel, text, select, $
;         title=title, xpos=xpos, ypos=ypos, $
;         rtcode=rtcode
;
; WaveMenu/Widgets: set flags / multiple selection from list
; -----
; START OF DESCRIPTION
;
; subroutine      (c) IPI, Uni Hannover 02'94, modified 02'94
;
; METHOD:          using IDL-Widgets                      (registered)
;                 toggle-buttons
;                 press button to select item
;
;                 routine should run under IDL 3.0 or higher
;                 on SparcStations      (SunOs)
;                 Pc's                  (MsWindows)
```

```

; INPUT PARAMETER: text      string-array with text to select from
;                      select  long-array with selection-index
;
; OUTPUT PARAMETER: select   as defined in routine
;                      elements selected return ONE, the rest ZERO
;
; INPUT KEYWORDS : title     title                      ["set flags"]
;                  x(y)pos   window position (!upper left)  [automatic]
;
; OUTPUT KEYWORDS : rtcode   Return-code : 0 for o.k., -1 for ERROR
;
; EXAMPLE:
;
; wmx_msel, 'DEMO: ' + sindgen(32), select & print, select
;
; END   OF DESCRIPTION
; -----
;-

; =====
=====
; event handling routine of WMX_MSEL

pro wmx_msel_event, event

; -----
; common

common wmx_sf, flags, $ ; tmp. select
      men_uv, $ ; menu-button-user-values
      n_text ; number of buttons

; -----
; WIDGET CONTROL

WIDGET_CONTROL, event.id, GET_UVALUE = value
if n_elements(value) eq 0 then value = '-1'

; -----
; which event

case value of

```

```

; -----
; nothing

'-1': value = value

; -----
; Done

'Done': WIDGET_CONTROL, event.top, /DESTROY

; -----
; buttons, set flag acc. to button

else: begin
  pos = where(men_uv eq value)
  if pos(0) ne -1 $
    then flags(pos(0)) = 1 - flags(pos(0))
  end
; -----
; end case (which event

endcase

; -----
; end (! NO RETURN)

end

; =====
=====
; int. MAIN: WMX_MSEL

pro wmx_msel, text, select, $
  title=title, xpos=xpos, ypos=ypos, $
  rtcode=rtcode

rtcode = -1 ; Return - code
message, ' working ...', /info

; -----
; test input

```

```

if n_params() ne 2 then begin
  print,"The number of parameters was wrong:",n_params(), string(7b)
  doc_library,"wmx_msel"
  return
endif

```

```

if n_elements(text) eq 0 then begin
  print,string(7b)
  message, 'TEXT UNDEFINED : ', /info
  return
endif

```

```

; -----
; test screen, keywords

```

```

device, get_screen_size = wsize

```

```

if n_elements(title) ne 1 then title = 'Set Flags'
title = strtrim(string(title), 2)

```

```

pos_ctrl= (n_elements(xpos) eq 1) or (n_elements(ypos) eq 1)

```

```

if n_elements(xpos) eq 0 then xpos = 256
if n_elements(ypos) eq 0 then ypos = 256
xpos = (xpos > 0) < (wsize(0)*9/10)
ypos = (ypos > 0) < (wsize(1)*9/10)

```

```

; -----
; save old window - index

```

```

old_win = !d.window

```

```

; -----
; common

```

```

common wmx_sf, flags, $ ; tmp. select
  men_uv, $ ; menu-button-user-values
  n_text ; number of buttons

```

```

; -----
; definitions, keywords II

```

```

n_text = n_elements(text) ; number of poss. selections

```

```

men_b = lonarr(n_text) ; button-bases
flags = lonarr(n_text) & flags(*) = 0 ; tmp. select/flags

men_uv = strtrim(sindgen(n_text), 2) ; buttons
men_v = text
pos = where(strlen(men_v) gt 32) ; cut to strings not longer
if pos(0) ne -1 $ ; than 32 characters
    then men_v(pos) = strmid(men_v(pos), 0, 60) + ' ...'

; -----
; main base

message, ' bases ... ',/info

if pos_ctrl $
    then mbase = WIDGET_BASE( /COLUMN, $
        TITLE=title, $
        XOFFSET=xpos, $
        YOFFSET=ypos, $
        /FRAME) $
    else mbase = WIDGET_BASE( /COLUMN, $
        TITLE=title, $
        /FRAME)

; -----
; sub-bases

upp_base = WIDGET_BASE (mbase, /ROW ) ;, FRAME=2 )

low_base = WIDGET_BASE (mbase, /COLUMN, FRAME=2, $
    Y_SCROLL_SIZE=wsize(1)*3/4) ;, $
    ;/EXCLUSIVE )

; -----
; done-button

waste = WIDGET_BUTTON (upp_base, VALUE='Done', UVALUE='Done', $
    FONT='helvetica-oblique-14')

; -----
; title

wtitle = '          ' + title + '          '

```

```

waste = WIDGET_LABEL (upp_base, VALUE = wtitle, $
    FONT='helvetica-bold-16')

; -----
; toggle-buttons

for l=0l, n_text-1l do begin
    waste = WIDGET_BASE(low_base, /EXCLUSIVE)
    men_b(l) = WIDGET_BUTTON (waste, VALUE=men_v(l), UVALUE=men_uv(l), $
        FONT='helvetica-oblique-12')
endfor

; -----
; create

message, ' realize ... ', /info

WIDGET_CONTROL, mbase, /REALIZE

; -----
; register

message, ' register ... o.k.', /info

xmanager, 'wmx_msel', mbase, /MODAL

; -----
; finishing, return & end

message, ' done ', /info
if old_win ne -1 then wset, old_win
select = flags
rcode = 0

return
end

```

---