Subject: droplist widget for long lists with Unix/X
Posted by Hans-Martin v.Stockha on Sun, 08 Dec 1996 08:00:00 GMT
View Forum Message <> Reply to Message

Hi folks,

I do some programming with IDL 4.0.1 (Solaris 2.5). Recently i found out that
the height of the screen is some kind of natural limitation to the number of
selectable elements in droplists, i.e. you can have arbitrarily long droplists,
but you can only select from those elements that fit on the screen.
In the MacOS version of IDL the droplists becomes scrollable in this case...

From the hotline i got a small piece of code which i turned into fully functional
compound widget that can be used as a replacement for WIDGET_DROPLIST under unix/X
if the list is too long.
The widget consists of a BASE, an optional LABEL, and a BUTTON which shows the
current selection. If you press the button a new window pops up which holds a
WIDGET_LIST with a scrollbar and a CANCEL button.

Since i only have access to two different platforms running IDL (MacOS and unix/X),
i do not know how the others handle droplists larger than the screen height...
BTW, for the MacOS CW_DROPLIST is just a wrapper for WIDGET_DROPLIST.

Maybe others came across the large list problem too and find the code useful,

Hans-Martin von Stockhausen
(stockh@pet.mpin-koeln.mpg.de)

PS.: Use the code on your own risk.

---------- CUT HERE ----------

```
; $Id: cw_droplist.pro,v 1.0 1996/11/29 16:09:51 $
;+
; NAME:
; CW_DROPLIST
;
```

```
; PURPOSE:
; This widget emulates a droplist with a button and a pop-up window.
; It was developed for Motif where droplists must fit on the screen
; in order to be able to access all elements...
; On the MAC this widget is only a wrapper for the original
WIDGET_DROPLIST.
;
; CATEGORY:
; Compound widgets.
;
; CALLING SEQUENCE:
; Result = CW_DROPLIST(Parent)
;
; INPUTS:
; Parent: The widget ID of the widget to be the droplist's parent.
;
; KEYWORD PARAMETERS:
; TITLE: An optional string containing the text to be used as the label
;  for the droplist.
;
; VALUE: The list to be displayed.
;
; XSIZE: An explicit horizontal size in pixels of the button showing the
;  current selection.
;
; FONT: A string containing the name of the X Windows font to use
;  for the TITLE, the BUTTON, and the LIST.
;
; All other keyword parameters will be given to the base of the
; compound widget via the _EXTRA mechanism.
;
; OUTPUTS:
; This function returns the widget ID of the button which hides the
list.
;
; COMMON BLOCKS:
; None.
;
; PROCEDURE:
; Create the widgets, set up the appropriate event handlers, and return
; the widget ID of the button that hides the list.
;
; To modify the state of the widget (selection or list) use:
;
; CW_DROPLIST_SET, id, list=list, set_list_select=set_list_select
;
; To query the state of the widget (selection or list) use:
;
```

```
; CW_DROPLIST_INFO, id, get_list_select=get_list_select,
get_list_number=get_list_number
;
; EXAMPLE:
; The code below creates a main base with a simple droplist in it:
;
;  base = WIDGET_BASE()
;  dl = CW_DROPLIST(base, VALUE=['a','b','c'], /FRAME)
;  WIDGET_CONTROL, base, /REALIZE
;
; MODIFICATION HISTORY:
;  Written by: H.-M. von Stockhausen   November 1996
;-


PRO CW_DL_SELECT_event, event
;
; select event: store position in list, set new button value
;
  WIDGET_CONTROL, event.top, GET_UVALUE=info, /NO_COPY
  info.Select = event.index ; current selection in list
  item = info.Empty  &  STRPUT, item, info.List(info.Select), 0
  WIDGET_CONTROL, info.Button, SET_VALUE=' '+item

  WIDGET_CONTROL, WIDGET_INFO(info.Button, /PARENT), /NO_COPY, $
                SEND_EVENT={Id:LONG(info.Button), Top:LONG(info.Top),
Handler:LONG(info.Button), Index:event.index}

  WIDGET_CONTROL, info.Button, SET_UVALUE=info, /NO_COPY

  WIDGET_CONTROL, event.top, /DESTROY
END


PRO CW_DL_CANCEL_event, event
;
; cancel event -> close list window
;
  WIDGET_CONTROL, event.top, GET_UVALUE=info, /NO_COPY
  WIDGET_CONTROL, info.Button, SET_UVALUE=info, /NO_COPY
  WIDGET_CONTROL, event.top, /DESTROY
END


FUNCTION GetScrOffset, id
;
; get the screen coordinates of the upper left corner of a widget
;
```

```
  offs = [0,0]
  WHILE (id NE 0) DO BEGIN
    geom = WIDGET_INFO(id, /GEOMETRY) ; get offset from parent
    offs = offs + [geom.xoffset, geom.yoffset]
    id = WIDGET_INFO(id, /PARENT)
  END

  RETURN, offs
END


PRO CW_DL_DROP_event, event
;
; create list in an extra window to select from
;
  WIDGET_CONTROL, event.id, GET_UVALUE=info, /NO_COPY
  info.Top = event.Top
  count = info.Count

  base = WIDGET_BASE(/COLUMN, TITLE=' ', /TLB_FRAME_ATTR)
  drop_list = WIDGET_LIST(base, VALUE=info.List, YSIZE=(count < 30),
EVENT_PRO='CW_DL_SELECT_event', FONT=info.Font)
  cancel = WIDGET_BUTTON(base, VALUE='cancel', /ALIGN_CENTER,
EVENT_PRO='CW_DL_CANCEL_event')

  ; position list window
  DEVICE, GET_SCREEN_SIZE=screen_size
  WIDGET_CONTROL, info.Top, TLB_GET_OFFSET=FramePos ; position of frame
hosting the group
  geom_base = WIDGET_INFO(base, /GEOMETRY)          ; size / position of
list top base
  geom_but = WIDGET_INFO(event.id, /GEOMETRY)       ; size / position of
DROPLIST-button

  offs = GetScrOffset(event.id)
  xoff = (offs(0) + geom_but.xsize) > 0
  base_xsize = geom_base.xsize + 2 * geom_base.margin
  IF ((xoff + base_xsize) GE screen_size(0)) THEN xoff = (offs(0) -
base_xsize) > 0
  base_ysize = geom_base.ysize + 2 * geom_base.margin + 30 ; extra for
window frame
  yoff = ((offs(1) + (geom_but.ysize / 2) - (base_ysize / 2)) > 0)
  ydiff = (yoff + base_ysize - screen_size(1) > 0)

  WIDGET_CONTROL, base, TLB_SET_XOFFSET=xoff
  WIDGET_CONTROL, base, TLB_SET_YOFFSET=yoff - ydiff

  WIDGET_CONTROL, base, /REALIZE
```

```
  WIDGET_CONTROL, drop_list, SET_LIST_SELECT=info.Select
  WIDGET_CONTROL, base, SET_UVALUE=info, /NO_COPY

  XMANAGER, 'drop_list', base, /MODAL
end


FUNCTION CW_DROPLIST_INFO, id, get_list_select=get_list_select,
get_list_number=get_list_number
;
; create list in an extra window to select from
;
  IF (!version.os EQ 'MacOS') THEN BEGIN
    IF KEYWORD_SET(get_list_select) THEN RETURN, WIDGET_INFO(id,
/DROPLIST_SELECT)
    IF KEYWORD_SET(get_list_number) THEN RETURN, WIDGET_INFO(id,
/DROPLIST_NUMBER)
    RETURN, 0
  ENDIF

  WIDGET_CONTROL, id, GET_UVALUE=info

  IF KEYWORD_SET(get_list_select) THEN RETURN, info.Select
  IF KEYWORD_SET(get_list_number) THEN RETURN, info.Count

  RETURN, 0
end


PRO CW_DROPLIST_SET, id, list=list, set_list_select=set_list_select
;
; create list in an extra window to select from
;

  IF (!version.os EQ 'MacOS') THEN BEGIN
    IF (N_ELEMENTS(list) NE 0) THEN WIDGET_CONTROL, id, SET_VALUE=list
    IF (N_ELEMENTS(set_list_select) NE 0) THEN WIDGET_CONTROL, id,
SET_DROPLIST_SELECT=set_list_select
    RETURN
  ENDIF

  WIDGET_CONTROL, id, GET_UVALUE=info, /NO_COPY

  IF (N_ELEMENTS(list) NE 0) THEN BEGIN
    IF (list(0) EQ '') THEN BEGIN
      count = 0
      width = 10
    END ELSE BEGIN
```

```
      count = N_ELEMENTS(list)
      width = MAX(STRLEN(list))
    END
    IF (count NE info.Count) THEN $   ; rebuild complete structure
       info = {Button:Info.Button, Top:info.Top, Width:width,
Empty:STRING(REPLICATE(32B, width)), $
               Select:Info.Select, Count:count, Font:info.Font,
List:list} $
    ELSE $
      info.List = list               ; just replace list
  ENDIF
  IF N_ELEMENTS(set_list_select) THEN info.Select = set_list_select
  info.Select = ((info.Select < (info.Count - 1)) > 0)

  item = info.Empty  &  STRPUT, item, info.List(info.Select), 0
  WIDGET_CONTROL, info.Button, SET_VALUE=' '+item
  WIDGET_CONTROL, info.Button, SENSITIVE=(info.Count GT 0)

  WIDGET_CONTROL, id, SET_UVALUE=info, /NO_COPY
end


FUNCTION CW_DROPLIST, parent, VALUE=list, TITLE=title, XSIZE=xsize,
FONT=font, _EXTRA=extra
;

  IF NOT KEYWORD_SET(FONT) THEN font = ''
  IF NOT KEYWORD_SET(xsize) THEN xsize = 0
  IF (N_ELEMENTS(title) EQ 0) THEN title = ''

  count = N_ELEMENTS(list)
  IF (count NE 0) THEN BEGIN
    width = MAX(STRLEN(list)) ; add always one blank left to the item
  END ELSE BEGIN
    width = 10 ; arbitrary width of empty string
    list = ''
  END

  IF (!version.os EQ 'MacOS') THEN $
    RETURN, WIDGET_DROPLIST(parent, VALUE=list, TITLE=title,
XSIZE=xsize, FONT=font, _EXTRA=extra)

  ; create scrollabel droplist for NON-MacOS
  base = WIDGET_BASE(parent, /ROW, /BASE_ALIGN_CENTER, _EXTRA=extra)

  info = {Button:0L, Top:0L, Width:width, Empty:STRING(REPLICATE(32B,
width)), $
          Select:0, Count:count, Font:font, List:list}
```

```
  IF (N_ELEMENTS(title) NE 0) THEN text = WIDGET_LABEL(base,
VALUE=STRING(title), FONT=font)
  button = WIDGET_BUTTON(base, VALUE=info.Empty, XSIZE=xsize,
EVENT_PRO='CW_DL_DROP_event', $
                           /ALIGN_LEFT, /DYNAMIC_RESIZE, FONT=font)

  info.Button = button
  item = info.Empty  &  STRPUT, item, list(info.Select), 0
  WIDGET_CONTROL, button, SET_VALUE=' '+item
  WIDGET_CONTROL, button, SENSITIVE=(info.Count GT 0)
  WIDGET_CONTROL, button, SET_UVALUE=info, /NO_COPY

  RETURN, button
END


 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;; ;

; ========== a larger example =============

;PRO MAIN13_Event, Event
;
;common a, cw_dl

;  WIDGET_CONTROL,Event.Id,GET_UVALUE=Ev

;  CASE Ev OF

;  'list' : print, 'Item: ', event.index

;  'but2' : CW_DROPLIST_SET, cw_dl,
list=['111','22','333333333','444','5555','666666','77777']
;  'but3' : CW_DROPLIST_SET, cw_dl, set_list_select=4
;  'but4' : CW_DROPLIST_SET, cw_dl, list=''
;  'but5' : PRINT,'item: ', CW_DROPLIST_INFO(cw_dl, /get_list_select)

;  'BUTTON2': BEGIN
;      Print, 'Event for Ende'
;      widget_control, event.top, /destroy
;      END
;  else:

;  ENDCASE
;END


;PRO CW_DL_TEST, GROUP=Group, title=title, frame=frame, font=font
```

```
;common a, cw_dl

;  IF N_ELEMENTS(Group) EQ 0 THEN GROUP=0

;  MAIN13 = WIDGET_BASE(GROUP_LEADER=Group, ROW=1, MAP=1,
TITLE='DROP-WIDGET Test')

;  ; This is my large string array, which represent my data.
;
list=["bbb","ccc","ddd","eee","fff","ggg","hhh", "iii","jjj","kkk","lll",
$
;
"aaa","bbb","ccc","ddd","eee","fff","ggg","hhh ","iii","jjj","kkk","lll",
$
;
"aaa","bbb","ccc","ddd","eee","fff","ggg","hhh ","iii","jjj","kkk","lll",
$
;
"aaa","bbb","ccc","ddd","eee","fff","ggg","hhh ","iii","jjj","kkk","lll"]

;  cw_dl = CW_DROPLIST(MAIN13, VALUE=list, TITLE=title, FRAME=frame,
/row, uvalue='list', xsize=100, font='5x8')

;  button = WIDGET_BUTTON( MAIN13, UVALUE='but2', VALUE=' new ')
;  button = WIDGET_BUTTON( MAIN13, UVALUE='but3', VALUE=' select 4 ')
;  button = WIDGET_BUTTON( MAIN13, UVALUE='but4', VALUE=' empty ')
;  button = WIDGET_BUTTON( MAIN13, UVALUE='but5', VALUE=' query ')
;  BUTTON2 = WIDGET_BUTTON( MAIN13, UVALUE='BUTTON2', VALUE='Ende')

;  WIDGET_CONTROL, MAIN13, /REALIZE

;  XMANAGER, 'MAIN13', MAIN13
;END
```