
Subject: "Ghostfont" - reproduced
Posted by [promashkin](#) on Mon, 21 Aug 2000 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

I have sent the following test method to RSI and Randy, but I thought some curious soul may try it too.

Below is some sample code. It may turn out wrapped, I am sorry, please unwrap yourself if it is.

To reproduce the glitch, please do the following. Once the sample code is compiled in a fresh session of IDL 5.3, execute:

```
;***** start of test*****  
x = junk()  
; Kill widget base using system native method, proceed  
obj_destroy, x  
heap_gc, /ver  
  
;No objects should remain in the heap after X has been cleaned up.  
;Now, do the following.  
  
x = junk()  
  
;DO NOT kill widget window, and call .Reset_session  
;This will destroy the widget and reset everything.  
  
.Reset_session  
  
;Now, execute:  
  
x = junk()  
; Kill widget base using system native method, proceed  
obj_destroy, x  
heap_gc, /ver  
  
;There should be an instance of IDLGRFONT left.  
;*****end of test*****
```

You can call the whole sequence over again without .Reset, and it will keep creating an extra IDLGRFONT.

I have tried it on IDL5.3 for PPC, G4-400 and on HP Unix. Both have this feature. I have nottried it on a PC, as my version of IDL on it is only 5.1 :-(

```
;***** code begins *****  
; Written by P.A.Romashkin, 08-2000  
; Class: Utility, Object Graphics
```

```

function junk
compile_opt IDL2, OBSOLETE

top_ID = widget_base(title='Object window')
draw_ID = widget_draw(top_ID, graphics=2, /button_event, xsize=600,
ysize=350, $
event_pro='junk_event', retain=2)

; Need to realize it all now, because we can only get value of realized
widget_draw in object mode.
widget_control, top_ID, /realize
xmanager, 'PARgrDisplay', top_ID, /No_block

;===== OBJECT INITIALIZATION START
=====
widget_control, draw_ID, get_value=Obj_Win
scene = obj_new('IDLgrScene')
container = obj_new('IDL_Container')
axes_list = obj_new('IDL_Container')

; Make X and Y ranges.
yrange = (xrange = [0.0, 1.0])

; Need to create new Model, View etc. objects.
model = obj_new('IDLgrModel')
view = obj_new('IDLgrView', viewplane_rect=[-0.1, -0.1, 1.12, 1.16],
location=[0.0, 0.0])

; Create and initialize axes.
default_font = obj_new('IDLgrFont', 'Helvetica', size=9.)
x_axis = obj_new('IDLgrAxis', 0, ticklen=0.03, name='X_AXIS',
location=[1000, 0, 0], /exact, _EXTRA=extra)
y_axis = obj_new('IDLgrAxis', 1, ticklen=0.03, name='Y_AXIS',
location=[0, 1000, 0], /exact, _EXTRA=extra)
; Create upper and right axes.
x_axis_top = obj_new('IDLgrAxis', 0, ticklen=0.03, tickdir=1, loc=[1000,
1, 0], /notext, /exact, _EXTRA=extra)
y_axis_top = obj_new('IDLgrAxis', 1, ticklen=0.03, tickdir=1, loc=[1,
1000, 0], /notext, /exact, _EXTRA=extra)
; Only place labels on the left and bottom axes.
x_axis -> getProperty, ticktext=x_tick_labels
y_axis -> getProperty, ticktext=y_tick_labels
x_tick_labels -> setProperty, recompute=2, font=default_font
y_tick_labels -> setProperty, recompute=2, font=default_font
; Create default axis titles.
x_title = obj_new('IDLgrText', 'X title', font=default_font, recompute=2)
y_title = obj_new('IDLgrText', 'Y title', font=default_font, recompute=2)
x_axis -> setProperty, title=x_title, tickformat='(G0.1)'

```

```

y_axis -> setProperty, title=y_title, tickformat='(G0.1)'
; Create plot title.
title = obj_new('IDLgrText', 'Object plot', font=default_font,
recompute=2, location=[0.5, 1.02], align=0.5)
; Set axes ranges and remove extra zeros from tick labels if they are present.
X_axis -> setProperty, range=xrange, xcoord_conv=n_xr
Y_axis -> setProperty, range=yrange, ycoord_conv=n_yr
X_axis_top -> setProperty, range=xrange, xcoord_conv=n_xr
Y_axis_top -> setProperty, range=yrange, ycoord_conv=n_yr
; Add atomic objects to the model.
model -> add, x_axis
model -> add, y_axis
model -> add, x_axis_top
model -> add, y_axis_top
model -> add, title
; Add model to its view and add view to the Scene.
view -> add, model
scene -> add, view
; Need to store all newly created objects in CONTAINER so that they can
be easily found or killed.
container -> add, model ; To locate with ease. Pos 0
container -> add, view ; To locate with ease. Pos 1
container -> add, default_font ; To kill with ease.
container -> add, scene ; To kill with ease.
; Create separate container for Axes.
axes_list -> add, x_axis
axes_list -> add, y_axis
axes_list -> add, x_axis_top
axes_list -> add, y_axis_top
; Create container for titles.
container -> add, title ; To locate or kill with ease. Pos 0
container -> add, x_title ; To locate or kill with ease. Pos 1
container -> add, y_title ; To locate or kill with ease. Pos 2
; Update the plot.
Obj_win -> draw, scene
;===== OBJECT INITIALIZATION END
=====
container -> add, title
container -> add, axes_list

return, container
end
;*****code ends*****

```

Cheers,
Pavel
