
Subject: Re: "Ghostfont" - reproduced

Posted by [promashkin](#) on Tue, 22 Aug 2000 07:00:00 GMT

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

Craig,

I am convinced that the problem only shows up after a use of .Reset_session, which appears first in 5.3. I observed no problem in 5.2. Regarding N_XR, it really was undefined; it turns out the 5.3 is tolerant to me passing undefined variables through keywords, while 5.2 is not. I am attaching the code that does run on 5.2 (I just killed the N_XR calls). I tried it on 5.3 and it still produces the "ghostfont".

Cheers,

Pavel

P.S. I avoid attachments because my overly smart mailer (Eudora) wraps text attachments, too. But I'll try - maybe, Netscape does not.

Craig Markwardt wrote:

```
>
> Pavel Romashkin <promashkin@cmdl.noaa.gov> writes:
>> 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:
>
> Pavel, I get the following error even before I run anything else:
>
> % IDLGRAxis::SETPROPERTY: Variable is undefined: N_XR.
>
> I am running on IDL v5.2, if that helps any.
>
> Also, much as I hate attachments, can you post these files as
> attachments? The word wrapping did indeed suck.
>
> Craigg
; Written by P.A.Romashkin, 08-2000
; Class: Utility, Object Graphics
; Purpose: Create object graphics window for 2D plots that allows addition (overplotting),
; removal of plots and zooming in and out.
```

function junk

```
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  
Y_axis -> setProperty, range=yrange  
X_axis_top -> setProperty, range=xrange  
Y_axis_top -> setProperty, range=yrange
```

```

; 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

```

File Attachments

1) [junk.pro](#), downloaded 89 times
