
Subject: Re: Persistent Object Graphics Not Persistent in Resizeable Windows
Posted by [Phil Williams](#) on Fri, 06 Jun 1997 07:00:00 GMT

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

David,

There is a suggestion in one of the manuals that retain=2 should not be used for draw widgets that are objects (graphics_level=2). Instead use /expose_events keyword and then have something like this

```
if event.id eq info.drawID then begin
  if event.type eq 4 then info.thisWindow->draw, info.thisView
endif
```

in the event handler to draw the window when it's exposed.

I added this to your sample code and the axes appear fine when the widget is resized.

Here's the edited code:

```
;-----
Pro XSurface_Cleanup, tlb

  ; Come here when program dies. Free all created objects.

Widget_Control, tlb, Get_UValue=info
Obj_Destroy, info.thisContainer
END
;-----

PRO Draw_Event, event
; Event handler to handle draw events.

Widget_Control, event.top, Get_UValue=info, /No_Copy

; The window was exposed so redraw it!
if event.type eq 4 then info.thisWindow->draw, info.thisView

Widget_Control, event.top, Set_UValue=info, /No_Copy

end

;-----

PRO XSurface_Resize, event
```



```
thisView = OBJ_NEW('IDLgrView', Color=[80,80,80],  
Viewplane_Rect=[-1,-1,2,2])
```

; Create a model and add it to the view.

```
thisModel = OBJ_NEW('IDLgrModel')  
thisView->Add, thisModel
```

; Create a wire mesh surface. Make it yellow.

```
thisSurface = OBJ_NEW('IDLgrSurface', data, Color=[255,255,0])
```

; Create axes for the surface. Color them green.

```
xAxis = Obj_New("IDLgrAxis", 0, color=[0,255,0], Ticklen=0.1)  
yAxis = Obj_New("IDLgrAxis", 1, color=[0,255,0], Ticklen=0.1)  
zAxis = Obj_New("IDLgrAxis", 2, color=[0,255,0], Ticklen=0.1)
```

; Add the surface and axes to the model.

```
thisModel->Add, thisSurface  
thisModel->Add, xAxis  
thisModel->Add, yAxis  
thisModel->Add, zAxis
```

; Get the data ranges for the surface.

```
thisSurface->GetProperty,XRange=xrange,YRange=yrange,ZRange=zrange
```

; Set the range and location of the axes. In this case,
; we are scaling the data into -0.5 to 0.5, so that even
; when the surface is rotated, it stays inside the -1 to 1
; viewing rectangle. Note that not all values in the Location
; keyword are used. (I've put really large values into the
; positions that are not being used to demonstrate this.) For
; example, with the X axis only the Y and Z locations are used.

```
xAxis->SetProperty, Range=xrange, Location=[1000.0, -0.5, -0.5]  
yAxis->SetProperty, Range=yrange, Location=[-0.5, 1000.0, -0.5]  
zAxis->SetProperty, Range=zrange, Location=[-0.5, 0.5, 1000]
```

; Scale the surface and axes into the range -0.5 to 0.5.

```
xs = [-0.5, 1/(xrange[1]-xrange[0])]  
ys = [-0.5, 1/(yrange[1]-yrange[0])]  
zs = [-0.5, 1/(zrange[1]-zrange[0])]  
thisSurface->SetProperty,XCoord_Conv=xs, YCoord_Conv=ys, ZCoord_Conv=zs
```

```
xAxis->SetProperty, XCoord_Conv=xs
yAxis->SetProperty, YCoord_Conv=ys
zAxis->SetProperty, ZCoord_Conv=zs
```

```
; Rotate model to the standard view.
```

```
thisModel->Rotate,[1,0,0], -90 ; To get the Z-axis vertical.
thisModel->Rotate,[0,1,0], 30 ; Rotate it slightly to the right.
thisModel->Rotate,[1,0,0], 30 ; Rotate it down slightly.
```

```
;Create the widgets to view the surface.
```

```
tlb = Widget_Base(Title='Surface Example', Column=1, TLB_Size_Events=1)
drawID = Widget_Draw(tlb, XSize=300, YSize=300, Graphics_Level=2, $
/expose_events, $
event_pro = 'draw_event')
Widget_Control, tlb, /Realize
```

```
; Get the window destination object.
```

```
Widget_Control, drawID, Get_Value=thisWindow
```

```
; Draw the view in the window.
```

```
thisWindow->Draw, thisView
```

```
; Create a container object to hold all the other
; objects. This will make it easy to free all the
; objects when we are finished with the program.
```

```
thisContainer = Obj_New('IDLgrContainer')
```

```
; Add the view to the container.
```

```
thisContainer->Add, thisView
```

```
; Create an INFO structure to hold needed program information.
```

```
info = { thisContainer:thisContainer, $ ; The object container.
drawID:drawID, $ ; The draw widget ID
thisWindow:thisWindow, $ ; The destination window
object.
thisView:thisView } ; The view object.
```

```
; Store the info structure in the UValue of the TLB.
```

```
Widget_Control, tlb, Set_UValue=info, /No_Copy
```

```
; Call XManager. Set a cleanup routine so the objects  
; can be freed upon exit from this program.
```

```
XManager, 'xsurface', tlb, Cleanup='XSurface_Cleanup', /No_Block, $  
  Event_Handler='XSurface_Resize'  
END
```

```
--
```

```
/*****  
Phil Williams, Ph.D.  
Research Instructor  
Children's Hospital Medical Center  "One man gathers what  
Imaging Research Center           another man spills..."  
3333 Burnet Ave.                  -The Grateful Dead  
Cincinnati, OH 45229  
email: williams@irc.chmcc.org  
URL: http://scuttle.chmcc.org/~williams/  
*****/
```
