
Subject: changing IDLgrAxis and IDLgrSurface ranges
Posted by [T Bowers](#) on Tue, 09 Nov 1999 08:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi again,
I'm trying to interactively (by way of menus) allow the user to adjust the x,y,z axes so they can just view what they want of an IDLgrSurface. So I have something like:

```
pro handleMenuEvents, event
;Event handler.
```

```
Widget_Control, event.top, Get_UValue=info, /No_Copy
Widget_Control, event.id, Get_UValue=menuItem
case menuItem of
  "AXISRANGE": begin
    ;get current axis ranges
    info.xAxis0->GetProperty, CRange=xRange
    info.yAxis0->GetProperty, CRange=yRange
    info.zAxis0->GetProperty, CRange=zRange
```

```
    ;lets just worry about X axis, say xRange=[0,100], and say a user
    through
    ; an input dialog set a variable newXRange to [10,50] (leaving the
    others
    ; as unchanged), i.e did the equivalent of the following commands:
    newXRange=[10,50] & newYRange=yRange & newZRange=zRange
```

```
    ;Now, if I just set the new ranges to the axis objects, x displays
    correctly
    ; but they don't re-position so xmin intersects y at ymin
    info.xAxis->SetProperty, Range=newXRange
    info.yAxis->SetProperty, Range=newYRange
    info.zAxis->SetProperty, Range=newZRange
```

```
    ;but I can't do the same to the surface since XRANGE, etc. are not
    ; accessible with the setProperty method:
    ;info.thisSurface->setProperty, XRange=newXRange, YRange=newYRange,
    ZRange=newZRange
```

```
    ;How do I get the surface to show only a part of itself without having
    to
    ; get the surface's data, chop it to a new array, create a new
    surface, and
    ; display that?
```

```
    end;/"AXISRANGE"
  endcase
```

```

end;//pro handleMenuEvents

pro myTest,...

;snip blah, blah, blah...

;Create surface
dataColoringByValue = Reform(ByteScl(data, /NaN), s[0]*s[1])
thisSurface = OBJ_NEW('IDLgrSurface', Style=style, Shading=1, $
  Vert_Colors=dataColoringByValue, Palette=thisPalette,
  UVALUE=dataColoringByValue)
thisSurface->setproperty, DATAZ=data, DATAX=x, DATAY=y

;Create axes
xAxis = Obj_New("IDLgrAxis", 0, Color=[255,255,255],
Exact=Keyword_Set(exact))
yAxis = Obj_New("IDLgrAxis", 1, Color=[255,255,255],
Exact=Keyword_Set(exact))
zAxis = Obj_New("IDLgrAxis", 2, Color=[255,255,255],
Exact=Keyword_Set(exact))
;Normalize a la D. Fanning
thisSurface->GetProperty, XRange=xrange, YRange=yrange, ZRange=zrange
xs = Normalize(xrange, Position=[-0.5,0.5])
ys = Normalize(yrange, Position=[-0.5,0.5])
zs = Normalize(zrange, Position=[-0.5,0.5])
thisSurface->SetProperty, XCoord_Conv=xs, YCoord_Conv=ys, ZCoord_Conv=zs
xAxis->SetProperty, Location=[9999.0, -0.5, -0.5], XCoord_Conv=xs
yAxis->SetProperty, Location=[-0.5, 9999.0, -0.5], YCoord_Conv=ys
zAxis->SetProperty, Location=[-0.5, 0.5, 9999.0], ZCoord_Conv=zs

;create menu
tlb = Widget_Base(Column=1, TLB_Size_Events=1, MBar=menubase)
;//Axes ranges
tAxesRange = Widget_Button(menubase, Value='Edit Axes Ranges...', $
  Event_Pro="handleMenuEvents", UValue='AXISRANGE')

;snip realization and all that other stuff...

end;//pro myTest

```

My question is in the handleMenuEvents code, how do I tell the surface to draw only a limited range of itself. Unfortunately there are no keywords like viewXRange, viewYRange, viewZRange to the setProperty method so I can tell it that even though the data ranges in x from 0 to 100, just realize the surface from 10 to 50. A very ugly solution is to pass the dataX, dataY, and dataZ arrays or a reference to them (can't use getProperty to retrieve the data), chop them up to the new ranges and stick these in new arrays, create a new surface, etc. Hideous.

Thanks for any advice,
TB
