Subject: Re: Object boundaries
Posted by Rick Towler on Mon, 02 Aug 2004 21:38:57 GMT
View Forum Message <> Reply to Message

Michael Wallace wrote:

>>> While this works, is there any way to remove the call to xobjview?
>>> When I remove that call and try the code, I get back 0.0 for all of
>>> the tick text ranges. Is this because IDL doesn't know where it is
>>> before it's drawn?
>>
>>
>>
>> Probably. The text objects really don't know how big they are
>> until their dimensions are computed. I presume this occurs just
>> before they are displayed the first time.
>
>
> It seems to me that it should be a fairly common problem then... you
> want to do some smart positioning of text objects. You have text object
> 1 which is placed somewhere. It needs to get drawn, and then text
> object 2 can be placed smartly (since text object one now has correct
> dimension numbers). Now, to see text object 2, you have to draw again.
> So, in order to get everything positioned just right, it appears that
> you have to go through multiple draw operations.
>
> Maybe there's something preventing this from being done, but why
> couldn't the text object at least calculate ahead of time where it falls
> on the view plane? Of course, I'd want that calculation to be lazy so
> that it doesn't get called every time you apply a transformation or
> twiddle some parameter, but there could be a method which would
> calculate those numbers given the current setup of the view. In essence
> this would be a Draw command, except nothing would really get drawn and
> it'd only apply to the text object (and any related object such as an
> axis that it's a property of). Then whenever I need to know those
> numbers, I could just call that method on the object and then get the
> [XYZ]RANGE properties which will now be filled in correctly.
>
> That's enough IDL theory for now. Time to get back to making pretty plots.

You're almost there... Don't be afraid to experiment.

Destination devices have a Draw method, but so does every atom. RSI doesn't provide us with the calling sequence but that is easy enough to figure out:

IDL> odest=obj_new('idlgrwindow')

```
IDL> oview=obj_new('idlgrview')
IDL> oaxis=obj_new('idlgraxis')
IDL> oaxis->getproperty, ticktext=ott
IDL> ott->getproperty, xrange=xr, yrange=yr, zrange=zr
IDL> print, xr,yr,zr
      0.00000000      0.00000000
      0.00000000      0.00000000
      0.00000000      0.00000000

;  The reason you are reading this...
IDL> oaxis->draw, odest, oview

IDL> ott->getproperty, xrange=xr, yrange=yr, zrange=zr
IDL> print, xr,yr,zr
    -0.033318131       1.0333181
    -0.076304187    -0.018814731
      0.00000000      0.00000000
```

So you can do exactly what you want to do. And don't worry, calling an atom's draw method will not actually cause it to be drawn in the destination device.

-Rick
