Subject: Re: window graphics resizing after setting to ps device
Posted by davidf on Wed, 24 Nov 1999 08:00:00 GMT
View Forum Message <> Reply to Message

Tony L. (alNOSPAM@bnl.gov) writes:

> I have a widget routine I've written that outputs an image to a graphics
> window where the width and height of the window are adjusted based on
> the size of the data array. I have the device set to 'WIN' with the
> set_plot command (NT box) and open the window with a line like this:
>
> window,0,title=filename,xsize=factx,ysize=facty
>
> All this works fine, I can plot a number of elements in this dataset
> with no problems. However, I then have a widget button that outputs this
> to a postscript file. I calculate the image size in a similar way to how
> I've done it for the graphics window, open the device using
> set_plot,'ps' and set the device using a line such as:
> device,/port, file= filename ,/color,bits_per_pixel=8
> and output using something like:
> tv,bytscl(reim,min=nowmin,max=nowmax,top=!d.table_size-1),$
> !x.window(0),!y.window(0),$
> xsize=x_pos-!x.window(0),$
> ysize=y_pos-!y.window(0),/norm
> and then I close the postscript device. All this works fine but when I
> then go to plot another data element in the graphics window, and this
> ONLY happens after outputting a file to postscript, the plot displays in
> the window resized to a smaller horizontal dimension (the window size
> remains the same but the x width of the plot is smaller).

> Anybody have any quick ideas what I might be doing wrong?

Uh, well, I want to say this in the gentlest possible
way, but pretty much everything. :-)

I think your *immediate* problem is that the aspect
ratio of your display window is not the same as the
aspect ratio of your PostScript "window". So that
whatever you are doing in the PostScript device--
and it can't be just doing a TV command, because
that doesn't set the system variables you are using
here--is changing the !X system variable.

But the fact that this works at all can only
be attributed to clean living and a strict
adherence to the straight and narrow, because
from a programming perspective it's the equivalent
of fast cars and wild women. Fun, but probably

not a good idea.

May I offer a couple of suggestions?

First, don't use a regular IDL graphics window in a widget program. Use a draw widget. If you want a separate window, then put the draw widget in it's own top-level base. But you have absolutely NO control over a graphics window and you have LOTS of control over a draw widget. For example, a draw widget is simple to resize. You can't resize a regular graphics window at all; you have to recreate it if you want it a new size.

Second, you may want to read a couple of articles on my web page about how to Produce Perfect PostScript Output.

```
http://www.dfanning.com/documents/tips.html#PostScript
```

You are doing one thing right: you are using Normalized coordinates. That is the key, for sure. You can get the aspect ratios of your windows correct by using the PSWindow program you find described in this article:

```
http://www.dfanning.com/tips/ps_aspect.html
```

For example, to get your PostScript "window" set up with the same aspect ratio as your current display window, all you have to do it this:

```
setup = PSWindow()
Set_Plot, 'PS'
Device, _Extra=setup
```

Third, it is not a good idea in general to position images using system variables (I.e., !X.Window) that are set by most commands *except* TV commands. It is absolutely fortuitous that your TV command works at all. And I think the chances of it working incorrectly are VERY high. (In fact, it wouldn't surprise me if this conversation is just one manifestation of this problem.)

If the image you are outputting to the PostScript file is just a screen dump of the graphics window (as seems likely to me), then all you really have to do if your

aspect ratios are correct is issue a simple TV command.
The image will fill up the window as you expect.

Fourth, if the image your are outputting to the PostScript file
is a screen dump, then I would encourage you NOT to do
that. Re-issue whatever commands you used to put the
graphics in the display window in the first place.
You do know what they are, because you are running
the widget program that created them!

Fifth, see if you can find someone who will let you
borrow a copy of my book. Creating these kinds of
simple widget programs that can send their output
to PostScript, JPEG, and GIF files is *exactly*
what it is all about. :-)

Good programming!

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438 E-Mail: davidf@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155
