Subject: Re: PNGs without X?
Posted by JD Smith on Fri, 14 Feb 2003 19:41:13 GMT
View Forum Message <> Reply to Message

On Fri, 14 Feb 2003 12:21:33 -0700, Dr. Sven Geier wrote:

> Boy, I hope top-posting is alright...
>
>
> A while ago I posted the thing appended below -- and got a few
> responses. I dug through the verious methods and none of them really
> work. I had noted before that postscript isn't really an option and the
> same reasons really apply to plotting into the Z-buffer: neat trick
> that, and I may well use that in the future; but the output I get is
> nowhere near the output that I get when plotting to an X window. The
> fonts are entirely different, the colors are differnt, pretty much
> everything is different.
>
> The color part, I suppose, is an outcropping of another problem, that I
> never managed to find a solution to: I find it very hard (to say the
> least) to use a certain given color in IDL. It seems like the simplest
> thing in the world to have a graph with four lines and I want one of
> them to be red and one green and one yellow and one blue. However there
> doesn't seem to be a mechanism in IDL(?) that accomplishes such a thing.
> Before I knew of 24-bit displays, I had a self-made colortable (with
> tvlct()) in which I knew index 1 is red and index 5 is blue and such,
> but the first time I tried this on a true-color visual, I learned
> something.
>
> The problem is that I need to be certain what the colors are before the
> plot, since there are some comments added to the data (via xyouts) that
> refer to "red line = such-n-such" and such. For the same reason I must
> be able to rely on a certain plot geometry (which I am consistently not
> getting when plotting in the Z-buffer) because the positions of things
> are carefully computed. The entire thing has been evolved over years to
> work just right and if you suddenly run it with a larger character size
> (like the Z-buffer seems to do) it'll all blow up.
>
> As an example, here's some random data and label:
>
> data = tan(findgen(5))
> plot,data,/nodata
> oplot,data,color='ff'x
> oplot,data,psym=5
> xyouts, 1,1.53,' <- triangle'
>
> On my 24-bit display this draws a few red lines, white triangles and
> labels one of the triangles with "triangle". If I do this on a

> "window,xsize=350,ysize=350" and then I do the same thing into the
> Z-buffer following David's website (great site, by the way. Or terrible
> site, as you may look at it: Many idl programmers lose much productive
> time browsing through it... :) I get something that is "kinda like" the
> original, except that the plotting area has a different size, the fonts
> are different and the color is gone - pretty much as different as a plot
> can be while still showing the same thing.
>
> !P.font = -1 or 0 give the same result in the Z-buffer, !p.font=1 gives
> a slightly prettyier (but less legible) one, but all three are different
> from their counterparts that are plotted into the X-buffer and not one
> of them is as clear and legible as the p.font=0 in X-windows.
>
> If I plot into the PS device and turn the result into an image like
> this:
>
> gs -q -sDEVICE=ppm -r100 -sOutputFile=- idl.ps | pnmflip -r90 | pnmscale
> \ -xsize=350 -ysize=350 | pnmtojpeg > test.jpg
>

```
device,decomposed=0  ; David's tip of the Holocene Epoch
tvlct,255b,0b,0b,1
set_plot,'PS'
 device,/color,bits_per_pixel=8,filename='test.eps',/encapsul ated
data = tan(findgen(5))
plot,data,/nodata
oplot,data,color=1
oplot,data,psym=5
xyouts, 1,1.53,' <- triangle'
device,/close
```

Then use ghostscript.

JD
