Subject: Re: Huge Maps & a device for faking a large window
Posted by Liam Gumley on Wed, 31 Mar 2004 21:59:26 GMT
View Forum Message <> Reply to Message

The Z-buffer offers a convenient way to define map projections without
needing a X display, e.g.

```
xsize = 43200    ; width of window
ysize = 21600    ; height of window
res = 1.0    ; Map resolution in kilometers
set_plot, 'Z'
device, set_resolution=[xsize, ysize], set_colors=256, z_buffering=0, $
  set_character_size=[10, 12]
scale = res * 4.0e6
map_set, latcen, loncen, scale=(scale * (!d.x_px_cm / 40.0)), /lambert, $
  position=[0, 0, 1, 1], /noerase
```

The scale transformation is to account for direct graphics devices which
don't have the same number of pixels per centimeter as the default X device.

That said, I've also had good luck with Xvfb.

Cheers,
Liam.
Practical IDL Programming
http://www.gumley.com/

"JD Smith" <jdsmith@as.arizona.edu> wrote in message
news:pan.2004.03.31.20.02.23.598950@as.arizona.edu...
>
>
> Any makers of large map projection images here? I'm having a
> conceptual problem creating a very large (~1Gpix) projected image. I
> bin a large data set into small bins tiling the entire range of
> latitude and longitude (43200x21600). I do this in a series of
> "tiles" to avoid working with the entire data set at once. So far so
> good. If I then want to warp this image to a given projection (like
> Aitoff), it seems I must first use MAP_SET to specify the projection
> details, *and* have a window open of the desired output size. The
> problem is, I have no intention of actually displaying the projected
> image (too large!), so all of the memory allocated for creating that
> big window is wasted (which is more than a nuisance when building such
> huge images).
>
> Unfortunately, MAP_PATCH (possibly via an undocumented keyword to
> TRIGRID -- MAP) and MAP_IMAGE (via CONVERT_COORDS) rely on a presently
> set window to dictate the size of the projected image. If
> TRIGRID(MAP=) were documented, perhaps I could do this myself, but it

> seems likely it also internally consults the current window to set the
> size for the coordinate transform. I see two ways out:
>
> a) Does anyone know of a way to access the mapping transformations
> directly (aside from re-coding them yourself), independent of any
> particular window geometry? Why shouldn't I be able to perform an
> arbitrary coordinate transformation using one of the many mapping
> transforms MAP_SET offers? Coupling this to a specific display
> device size is an unnecessary limitation.
>
> b) Barring this, is there a device in which a window can be
> established which does not consume any memory or accept display
> commands, but simply provides a dummy framework from which
> CONVERT_COORDS etc. can take window info?
>
> Thanks,
>
> JD
>
