Subject: Re: x/y margin from map_set to plot position and cgimage
Posted by David Fanning on Tue, 27 Aug 2013 01:11:26 GMT

View Forum Message <> Reply to Message

AMS writes:

> Suppose I have set up a part of a window to draw a map, and used xmargin and ymargin to specify the margins around this map. For example:
>
> map_set,0,0,xmargin=[2,2],ymargin=[2,2]
>
> Suppose I then want to draw an image (with cgimage) inside this map. Is there a way to pass the xmargin/ymargin information so the image is drawn inside the map region rather than the whole plot window? I don't see those as keywords to cgimage. If not, is there some way I can pull out the plot position which arose from my map_set call and pass this to cgimage? I can't necessarily assume that !p.multi=0, so I don't think the cgimage position keyword would help me here?

Coyote Graphics routines do not use margins. Margins are based on the strangest unit ever: character size. It is impossible to explain how margins work, and even more impossible to get anything to line up with them. All Coyote Graphics routines are positioned with the POSITION keyword.

> (I am switching over a lot of old code written for IDL 5/6 to use newer features and the Coyote library... so am now becoming familiar with some 'under the hood' stuff which I had not looked at in detail before. I'm now using IDL 7.1.1 and 8.2.2.)
>
> I would not be averse to recoding to use something like the Coyote Graphics position keyword in my map_set call, but as far as I know that can't be done with map_set (and I don't want to pass the position keyword to map_set because often !p.multi will not be 0).

The cgMap_Set program is an exception to the rule. I don't use cgMap_Set much, and it is simply a wrapper for Map_Set. I do almost all my map projection work with the Map_Proj_*** routines, which are old, but still mostly professional. The cgMap object is a wrapper to the Map_Proj_Init command and *does* allow you to position a map projection in a graphics window. You can use the same position for the cgImage command. Usually, it it done the other way around, and we create a map projection for a map projected image. That is done something like this:

```
   map = cgMap('Mercator', ...)
   pos = [0.1, 0.1, 0.9, 0.9]
   cgImage, myImage, Position=pos, OPosition=opos, /Keep_Aspect
   map -> SetProperty, Position=opos
   map -> Draw
   cgMap_Continents, Map=map
```

I wouldn't use !P.Multi to position maps, probably. I think I would use cgLayout to calculate the map positions. cgLayout is much more flexible than !P.Multi, although it does the same job of laying out multiple plots in a gridded display.

> The longer story for what I am trying to do is that I have written code to map regularly or irregularly lat-lon gridded data from e.g. satellites. This basically makes use of polyfill to draw each point. If the grid is not regular, my code allows for a proper expression of the ground pixel size. But if the grid is regular (e.g. level 3 data), I figure I should be able to map it as an image, which could be a lot faster than using polyfill.

Yes, I am very close to having a new cgWarpToMap program that will do this kind of thing very simply and quickly. Maybe I'll have time to finish it this week.

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
Sepore ma de ni thue. ("Perhaps thou speakest truth.")