
Subject: Re: Display and Navigate Image in IDL 8.2
Posted by [lecacheux.alain](#) on Fri, 07 Sep 2012 13:58:37 GMT
[View Forum Message](#) <> [Reply to Message](#)

Le vendredi 7 septembre 2012 15:15:35 UTC+2, David Fanning a écrit :

> Klemen writes:

>

>

>

>> David, I have jsut upgraded to IDL 8.2 so I can test it...

>

>

>

> Oh, sorry. This is the third time I've paid my money and

>

> upgraded to something unusable to me, too. I'm a really

>

> slow learner. :-(

>

>

>

>> The code is, I think, ok (I haven't so much experience with Image). The only problem is your definition of you georeference. I think that your ranges are defined for a usual Mercator projection with origin in 0N, 0E. But you have moved your origin to 105.1W, 40.6N. This point should have in map coordinates values 0, 0. This means that below defined range is false:

>

>>

>

>> XRange: [-11711131.0, -11688226.0] (meters)

>

>> YRange: [4914254.0, 4937159.5] (meters)

>

>>

>

>> You could redefine your range as:

>

>> n = 600

>

>> res = 38.1757

>

>> x = findgen(n)* res - n*res*0.5

>

>> y = reverse(x)

>

>> xrange = [min(x), max(x)]

>

>> yrange = [min(y), max(y)]

>

>>
>
>> Well, then I get at least something, I am not sure if this is exactly what you are looking for, but
right mouse click gives me at least the proper longitude when I scroll over the image.
>
>
>
> Maybe I'm seeing something different from you, but
>
> when I put this code into my program, I see a
>
> map projection filling the window, and a small spec
>
> in the middle of my window which presumably represents
>
> my image.
>
>
>
> Probably not what I am looking for. :-)
>
>
>
> Cheers,
>
>
>
> David
>
>
>
>
>
> --
>
> David Fanning, Ph.D.
>
> Fanning Software Consulting, Inc.
>
> Coyote's Guide to IDL Programming: <http://www.idlcoyote.com/>
>
> Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Hi David,

Here is my solution:

1st step: define the size of the image in decimal degrees, by using your "magic" resolution

number. I guess that it might be retrieved by another way.

```
centerlat = 40.6000d
centerlon = -105.1000d
res = 38.1757
map = Map_Proj_Init('mercator', /gctp, ELLIPSOID='wgs 84')
cm = map_proj_forward(centerlon, centerlat, MAP=map)
xrange = cm[0] + [-300,300]*res
yrange = cm[1] + [-300,300]*res
xr = map_proj_inverse(xrange, yrange, MAP=map)
print,xr
; -105.20288    40.521535
; -104.99712    40.678372
```

As you can see, numbers are slightly different from yours (in latitude).

2nd step: draw the image and add the grid map

```
obj = IMAGE(googleimage, $
  GRID_UNITS=2, ;units of degrees
  MARGIN=[0.05,0.15,0.05,0.05], $ ;to manage space for labels
  IMAGE_LOCATION=xr[0:1,0],$
  IMAGE_DIMENSIONS=[xr[0,1]-xr[0,0],xr[1,1]-xr[1,0]],$
  MAP_PROJECTION='Mercator')
obj.mapprojection.mapgrid.BOX_AXES = 1
obj.mapprojection.mapgrid.BOX_THICK = 10
obj.mapprojection.mapgrid.LINESTYLE = 1
obj.mapprojection.mapgrid.GRID_LONGITUDE = 0.04
obj.mapprojection.mapgrid.GRID_LATITUDE = 0.03
obj.mapprojection.mapgrid.LABEL_POSITION = 0
```

The trick (that I have found in spite of the cryptic Exelis documentation, but better solutions are likely to exist) was to consider that any NG building consists in a tree of graphics objects. Then, when plotting your image, the use of the MAP_PROJECTION keyword implies that such a corresponding MAP object is added to the IMAGE one, then that a MAPGRID object should be also available. The latter can be retrieved through the MAPPROJECTION property. Please consider how much a simple underline can change to your life !

Now I did not succeed with the BOX_AXES keyword. I leave that for your exercising ...

Alain.
