
Subject: Re: Display and Navigate Image in IDL 8.2
Posted by [DavidF\[1\]](#) on Mon, 10 Sep 2012 18:29:48 GMT
[View Forum Message](#) <> [Reply to Message](#)

I wrote last Friday:

> There is, apparently, no way to get box axes on the plot,
> but to get even this far in two days time is a major
> achievement! I'm going for a beer.

I learned today, with some help from the folks at ExcelisVIS,
how to add a box axis to the image.

The secret is to NOT use the LIMIT and BOX_AXES
keywords when you create the map projection, since
this results in a blank window. (I don't know why
I didn't think of this!)

Rather, you should set the LIMIT and BOX_AXES
properties on your map projection sometime after
you display the image and sometime before you
throw the goddamn computer out the window.

In other words, don't do this:

```
obj = Image(googleImage, MAP_PROJECTION='mercator', $  
  ELLIPSOID='WGS 84', GRID_UNITS=1, $  
  X RANGE=xrange, Y RANGE=yrange, $  
  DIMENSIONS=[700,700], POSITION=[0.05, 0.05, 0.85, 0.85], $  
  IMAGE_DIMENSIONS=[xdim,ydim], IMAGE_LOCATION=[xloc,yloc], $  
  LIMIT=limit, /BOX_AXES)
```

Do this instead:

```
obj = Image(googleImage, MAP_PROJECTION='mercator', $  
  ELLIPSOID='WGS 84', GRID_UNITS=1, $  
  X RANGE=xrange, Y RANGE=yrange, $  
  DIMENSIONS=[700,700], POSITION=[0.05, 0.05, 0.85, 0.85], $  
  IMAGE_DIMENSIONS=[xdim,ydim], IMAGE_LOCATION=[xloc,yloc])  
obj.mapprojection.mapgrid.BOX_AXES=1  
obj.mapprojection.LIMIT = limit
```

It's true, the box axes are not labeled correctly, but at least
they are there. :-)

But, now I see the real reason for the bad documentation. The
technical writer, understanding that reading the documentation
is not going to help at all (indeed, it seems designed to lead

you astray!) has decided to do as poor a job of writing as he can get away with, causing you to more easily give up on the documentation altogether. In this way, he leads you inevitably toward the true path of learning: unfocused and random experimentation. It is really pure genius, when you think about it!

Cheers,

David
