Subject: Coyote Graphics 2.0 Update
Posted by David Fanning on Tue, 10 May 2011 20:37:06 GMT
View Forum Message <> Reply to Message

Folks,

Oh, my goodness. This summer is shaping up to be incredibly busy for me. It's not clear how much time I will have to either write books or programs. So, before it gets out of control I thought I would take a quick stab at writing a Coyote Graphics 2.0 plot command. This is the version that looks and works identically to the IDL 8 function graphics plot command.

Oh, my gosh. These things are cool. Really, really cool. And blazingly fast! Honestly, I have never seen anything like it. I'm going to see if I can get a resizeable graphics window working with this so I can let you play with it. Wow!

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.")

Subject: Re: Coyote Graphics 2.0 Update
Posted by David Fanning on Thu, 12 May 2011 14:46:57 GMT
View Forum Message <> Reply to Message

Craig Markwardt writes:

> Sound's like Fermat's enigmatic comment in the margin of a book, which
> needs to be only slightly modified in this context...
>
> "I have discovered a truly marvelous version of this PLOT command,
> which this post is too small to contain."
>
> Exercise left to the reader.

No, no. It's coming momentarily. I've had a number of "adventures" in object programming since my last

post. Some of my own doing, but some not. But, the
code is done. The instructions are in progress. :-)

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.")

Subject: Re: Coyote Graphics 2.0 Update
Posted by David Fanning on Thu, 12 May 2011 16:00:43 GMT
View Forum Message <> Reply to Message

David Fanning writes:

> No, no. It's coming momentarily. I've had a number
> of "adventures" in object programming since my last
> post. Some of my own doing, but some not. But, the
> code is done. The instructions are in progress. :-)

OK, this is *extremely* premature, but I wanted to
give you a taste of what Coyote Graphics 2.0 will
look like. These programs are lightly documented and
even more lightly tested. I've written only the
Plot function and only enough of the Window function
to make things work. I think you can see by looking
at the code, however, that you don't have to be some
kind of programming genius to write these kinds of
programs. I'm pretty sure most of the IDL programmers
on this newsgroup could finish this job!

These programs can be found in the "experimental"
folder of the Coyote Library zip file:

http://www.idlcoyote.com/programs/zip_files/coyoteprograms.z ip

There are four programs required to get this basic
functionality.

cgsgraphic__define.pro -- This program defines and manages
common graphics keywords.
cgsquery.pro -- This is a utility program, written quickly

so I can find the "current" graphics window.
cgsplot.pro -- This is the "Plot" program. The keywords
available to it are identical to the keywords available
to the cgPlot command. That is, the normal PLOT keywords
plus some "added" functionality (ASPECT, LAYOUT, color
names allowed, etc.).
cgswindow.pro -- This is the resizeable graphics window that
the plot command is displayed in, if desired. It is a
slightly modified form of the cgWindow program.

Basically, to use this Plot object, you do something like this.

```
data = cgDemoData(1)
p = cgsPlot(data)
```

The plot is now displayed in a resizeable graphics window.

If you want to add a plot to the current graphics window, you use the /CURRENT keyword. (Not much use for this until I get other commands built, but its here anyway.) For example:

```
w = cgsWindow()
p = cgsPlot(data, /Current)
```

The variable p is a plot object, which you can interact with. How you interact with it depends on which version of IDL you are using.

Suppose, for example, you wanted to make the data red.

IDL 8.x:

```
p.color = 'red'
```

Other versions of IDL:

```
p -> Set, Color='red'
```

Other properties (i.e. keywords) can be set in a similar fashion. For example, to add a title to the plot.

```
p.title = 'CGS Plot'
```

Or,

```
p -> Set, TITLE='CGS Plot'
```

I have added quick and dirty methods to allow you to create

PostScript and raster file output. These will be improved
in the final release, but I wanted to demonstrate that it
it easy enough to do. Like other CGS programs, the raster
files will be created from a PostScript intermediary file,
if the program can find the ImageMagick convert command.

These methods work like the IDL 8 methods work. You supply
the name of an output file. To create a PostScript file,

```
p.ps, 'test.ps'
```

Or,

```
p -> PS, 'test.ps'
```

To create, say, a PNG file (the type of file will depend on
the file extension of the output filename):

```
p.save, 'test.png'
```

Or,

```
p -> Save, 'test.png'
```

You can also create PostScript and raster files from the
pull-down menus in the graphics window.

Have fun!

I have to turn my attention to other matters for awhile, but
this will give you a bit of a preview anyway.

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.")