Subject: Re: using plot() function in a batch file
Posted by David Fanning on Tue, 11 Jan 2011 19:19:18 GMT
View Forum Message <> Reply to Message

Marc Buie writes:

> I've been investing some time in working with the new plotting
> functions in IDL v8.0. Things are still a little rough around
> the edges but there are definite advantages, particularly the
> device independence.
>
> There are occasions when I generate graphics in a batch file or cron
> job. Under these circumstances, the IDL process does not have access
> to a display (X or otherwise). With direct graphics I used the 'Z
> buffer' as the device.

I'll just mention again that the new Coyote Graphics routines are all device and color model independent, and appear to work identically in the Z-buffer, on display, or in a PostScript file. :-)

They have just been updated about five minutes ago with a new routine, FSC_DefCharSize(), which creates default character sizes that old people like you and I can actually read, Marc. This also makes them look a great deal more like the function graphics routines.

Cheers,

David

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

Subject: Re: using plot() function in a batch file
Posted by David Fanning on Tue, 11 Jan 2011 19:22:15 GMT
View Forum Message <> Reply to Message

David Fanning writes:

> I'll just mention again that the new Coyote Graphics
> routines are all device and color model independent,
> and appear to work identically in the Z-buffer, on

> display, or in a PostScript file. :-)

I guess I should also mention that these routines work
in *any* version of IDL, as far as I know.

You can learn more about them here:

http://www.dfanning.com/graphics_tips/newoldcmds.html

Cheers,

David

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

Subject: Re: using plot() function in a batch file
Posted by Michael Galloy on Tue, 11 Jan 2011 19:31:46 GMT
View Forum Message <> Reply to Message

On 1/11/11 11:45 AM, Marc Buie wrote:
> Folks -
>
> I've been investing some time in working with the new plotting functions in IDL v8.0. Things are still a little rough around the edges but there are definite advantages, particularly the device independence.
>
> There are occasions when I generate graphics in a batch file or cron job. Under these circumstances, the IDL process does not have access to a display (X or otherwise). With direct graphics I used the 'Z buffer' as the device.
>
> Has anyone found a way to do this with the new tools? I've been all over the documentation on this and can't find any relevant discussion. I did try the following:
>
> p=plot(x,y,hide=1)
> p.save,'test.png'
>
> but this crashes with the following message:
>
> % WIDGET_CONTROL: Unable to connect to X Windows display: :0.0
> % PLOT: WIDGET_CONTROL: Unable to establish X Connection.
>
> This doesn't sound very encouraging but I'm still holding out hope that there's something I've forgotten about and just haven't been able to locate in the documentation.

>
> Cheers,
> Marc

Use the /BUFFER keyword to send the output to a buffer. It can then be
saved with the save method:

p = plot(x, y, /buffer)
p->save, 'test.png'

Mike
--
www.michaelgalloy.com
Research Mathematician
Tech-X Corporation

Subject: Re: using plot() function in a batch file
Posted by David Fanning on Tue, 11 Jan 2011 20:03:11 GMT
View Forum Message <> Reply to Message

Michael Galloy writes:

> Use the /BUFFER keyword to send the output to a buffer. It can then be
> saved with the save method:
>
> p = plot(x, y, /buffer)
> p->save, 'test.png'

Wow! This just gave me an idea for my Coyote Graphics
routines. I realized I could add an OUTPUT keyword to
my routines, which in about 10 lines of code will give
me the ability to create PostScript, PNG, JPEG, TIFF, etc.
output files from my routines without drawing anything
on the display.

I just tried it with FSC_Plot and got a terrific PNG
file, with beautiful PostScript fonts (not ugly Hershey
fonts) by typing this command. OUTPUT=2 selects PNG files.

```
IDL> fsc_plot, loaddata(1), color='red', axiscolor='blue', output=2
```

Man, you have GOT to like that! :-)

Cheers,

David

P.S. I have *really* got to do something else this afternoon,
but check back tomorrow. This is outstanding! Goodness!
Maybe I will sell a couple of copies of this new book. ;-)

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

Subject: Re: using plot() function in a batch file
Posted by Marc Buie on Tue, 11 Jan 2011 20:27:08 GMT
View Forum Message <> Reply to Message

Mike -

Thanks so much. I _knew_ that I'd seen something on this somewhere but just couldn't find it. Now that you point it out I can easily find it in the docs. This is the simple answer I was hoping for, though I don't want to take anything away from David's latest plotting project. That's pretty decent stuff, too.

--Marc

Subject: Re: using plot() function in a batch file
Posted by David Fanning on Tue, 11 Jan 2011 20:51:53 GMT
View Forum Message <> Reply to Message

David Fanning writes:

> Wow! This just gave me an idea for my Coyote Graphics
> routines. I realized I could add an OUTPUT keyword to
> my routines, which in about 10 lines of code will give
> me the ability to create PostScript, PNG, JPEG, TIFF, etc.
> output files from my routines without drawing anything
> on the display.
>
> I just tried it with FSC_Plot and got a terrific PNG
> file, with beautiful PostScript fonts (not ugly Hershey
> fonts) by typing this command. OUTPUT=2 selects PNG files.
>
> IDL> fsc_plot, loaddata(1), color='red', axiscolor='blue', output=2
>
> Man, you have GOT to like that! :-)

Alright, now that I've had a cold shower and a chance

to cool off (I could have run outside and run around
in the snow on this frigid, sunny day in Colorado), I
am coming to my senses. :-)

What I accomplished in FSC_PLOT in 10 lines above,
I can really accomplish in three lines at the IDL
command line:

```
IDL> PS_Start
IDL> FSC_Plot, loaddata(1), color='red', axiscolor='blue'
IDL> PS_End, /PNG
```

(The extra lines were for the CASE statement, selecting
the different kinds of output plots.)

The problem with putting that *into* FSC_PLOT is that
PS_Start and PS_End have keywords that can control some
of the properties of the output (eg., FILENAME). If I
put the commands into FSC_Plot, I have to have some way
of getting these keywords into FSC_Plot, too. So that's
another 10 or so lines, plus documentation. It was right
about in here that I began to wonder if this was such a
bright idea after all. :-)

I think I'll probably leave things they way they are for
the moment. After the book is out, I plan to turn these
routines into objects anyway (accessed as functions,
the way IDL 8 function graphics are). I guess it probably
makes more sense to leave the extra bells and whistles
alone until then.

Cheers,

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