Subject: Coyote Graphics PS/PDF output size/orientation
Posted by Paul Levine on Tue, 13 Aug 2013 23:19:18 GMT
View Forum Message <> Reply to Message

Greetings,

I've been getting into the Coyote's Guide to Traditional IDL graphics, and am starting to make better use of the Coyote Graphics system. Good stuff!

I am creating some plots with multiple cg commands (cgAxis, cgOplot, etc.) along with al_legend. The graphics window itself displays in a landscape orientation, using the default 640 x 512 window size, which is what I want. If I create a PNG output, using either the Coyote Graphics window "File" drop-down menu, or using cgControl, Output='filename.png', then I end up with a 640 x 512 pixel PNG, which is what I expect. All good so far.

But when I output to PostScript or PDF, using either the Coyote Graphics window "File" drop-down menu, or using cgControl, Output='filename.ps/pdf', I end up with an 8.5 x 11 inch file with my plot rotated 90 degrees counterclockwise. The aspect ratio of my plot is correct, so if I rotate the page, everything looks good, except that I have excess whitespace on either side. Also, I'd like to programmatically create lots of PDF files as output, so having to rotate each of them will be a bit cumbersome (I'm sure I can automate that outside of IDL, but would prefer to learn how to do it right the first time).

Do I need to be wrapping my graphics commands with ps_start and ps_end so that I can explicitly specify the /landscape keyword to ps_start?

I am using Mac OS X 10.8.4 with GPL GhostScript 9.07 and the newest version of Coyote Graphics (downloaded today)

Thanks,
Paul

Subject: Re: Coyote Graphics PS/PDF output size/orientation
Posted by Phillip Bitzer on Tue, 13 Aug 2013 23:59:01 GMT
View Forum Message <> Reply to Message

There's likely a better way, but this should work:

```
fileName = 'test.eps'
cgPLOT, DIST(10), /WIN
cgCONTROL, output =fileName, /PS_ENCAP
```

Subject: Re: Coyote Graphics PS/PDF output size/orientation
Posted by David Fanning on Wed, 14 Aug 2013 00:07:34 GMT
View Forum Message <> Reply to Message

Paul Levine writes:

> But when I output to PostScript or PDF, using either the Coyote
> Graphics window "File" drop-down menu, or using cgControl,
> Output='filename.ps/pdf', I end up with an 8.5 x 11 inch file with my
> plot rotated 90 degrees counterclockwise. The aspect ratio of my plot
> is correct, so if I rotate the page, everything looks good, except that
> I have excess whitespace on either side. Also, I'd like to
> programmatically create lots of PDF files as output, so having to
> rotate each of them will be a bit cumbersome (I'm sure I can automate
> that outside of IDL, but would prefer to learn how to do it right the
> first time).
>
> Do I need to be wrapping my graphics commands with ps_start and ps_end
> so that I can explicitly specify the /landscape keyword to ps_start?

The output from a cgWindow takes on the aspect ratio of the window, since it assumes you want output that resembles as closely as possible what you are looking at in the window. So, the simple way to get a portrait aspect ratio in your PostScript and PDF output is to create a window with a portrait aspect ratio.

You can either grab the window and muscle it into the correct size, or you can make it the right size when you create it. Something like this:

```
cgWindow, WXSize=600, WYSize=800
cgPlot, cgDemodata(1), /AddCmd
```

Or, something like this:

```
cgPlot, cgDemoData(1), /Window, WXSize=600, WYSize=800
```

Note that the latter command only works if there are no cgWindows currently on the display, since it uses whatever window is "open" or available, as normal IDL commands do.

I don't have a way currently to resize a window programmatically, but I will in about five minutes. :-)

> I am using Mac OS X 10.8.4 with GPL GhostScript 9.07 and the newest

> version of Coyote Graphics (downloaded today)

Make sure you got today's *afternoon* version. The version this morning
was causing me some trouble after I made a change in cgPS2Raster. I had
to revert back to the original file until I get the problem figured out.

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.")

Subject: Re: Coyote Graphics PS/PDF output size/orientation
Posted by David Fanning on Wed, 14 Aug 2013 00:22:57 GMT
View Forum Message <> Reply to Message

David Fanning writes:

> I don't have a way currently to resize a window programmatically, but I
> will in about five minutes. :-)

You can download a new cgControl here:

http://www.idlcoyote.com/programs/cgcontrol.pro

To resize a cgWindow programmatically, you can do this:

cgControl, Resize=[600,800] ; Portrait mode window

Or, even this:

cgControl, Resize=600 ; Square aspect window.

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.")

Subject: Re: Coyote Graphics PS/PDF output size/orientation
Posted by Paul Levine on Wed, 14 Aug 2013 00:33:37 GMT
View Forum Message <> Reply to Message

On 2013-08-14 00:07:34 +0000, David Fanning said:

> Paul Levine writes:
>
>> But when I output to PostScript or PDF, using either the Coyote
>> Graphics window "File" drop-down menu, or using cgControl,
>> Output='filename.ps/pdf', I end up with an 8.5 x 11 inch file with my
>> plot rotated 90 degrees counterclockwise. The aspect ratio of my plot
>> is correct, so if I rotate the page, everything looks good, except that
>> I have excess whitespace on either side. Also, I'd like to
>> programmatically create lots of PDF files as output, so having to
>> rotate each of them will be a bit cumbersome (I'm sure I can automate
>> that outside of IDL, but would prefer to learn how to do it right the
>> first time).
>>
>> Do I need to be wrapping my graphics commands with ps_start and ps_end
>> so that I can explicitly specify the /landscape keyword to ps_start?
>
>
> The output from a cgWindow takes on the aspect ratio of the window,
> since it assumes you want output that resembles as closely as possible
> what you are looking at in the window. So, the simple way to get a
> portrait aspect ratio in your PostScript and PDF output is to create a
> window with a portrait aspect ratio.

That much I understood from reading your explanation at http://www.idlcoyote.com/cg_tips/cglandscape.php but in this case (and I apologize if my original post was a bit confusing) I do NOT want a portrait. I want the same landscape image that I have in the display window (the same one that I get when I write to PNG), but I'm ending up with a portrain-oriented PDF with a sideways graphic. The display window and PNG output are the same, but the PDF output is rotated 90 degrees in terms of both the graphic and the aspect ratio.

>
> You can either grab the window and muscle it into the correct size, or
> you can make it the right size when you create it. Something like this:
>
>   cgWindow, WXSize=600, WYSize=800
>   cgPlot, cgDemodata(1), /AddCmd
>
> Or, something like this:
>
>   cgPlot, cgDemoData(1), /Window, WXSize=600, WYSize=800

>
> Note that the latter command only works if there are no cgWindows
> currently on the display, since it uses whatever window is "open" or
> available, as normal IDL commands do.

I tried adding the WXSize and WYSize (though I used 800 for x and 600
for y because I am trying to get landscape output), but I still get the
same result, i.e. PNG oriented correctly and PDF rotated 90 degrees.

But this brings up a somewhat-related question about PNG output. When
I use WXSize=800 and WYSize=600 my PNG file ends up being 752 x 556
pixels, and the bottom of my x axis title is chopped off. Should I not
be getting 800 x 600 pixels out? And is my truncated title indicative
of some mistake on my part? The PDF file shows the entire x axis
title, but of course the file itself is 8.5 x 11 inches.

> Make sure you got today's *afternoon* version. The version this morning
> was causing me some trouble after I made a change in cgPS2Raster. I had
> to revert back to the original file until I get the problem figured out.

Indeed, using the latest version.

Subject: Re: Coyote Graphics PS/PDF output size/orientation
Posted by David Fanning on Wed, 14 Aug 2013 00:38:34 GMT
View Forum Message <> Reply to Message

Paul Levine writes:

> I tried adding the WXSize and WYSize (though I used 800 for x and 600
> for y because I am trying to get landscape output), but I still get the
> same result, i.e. PNG oriented correctly and PDF rotated 90 degrees.
>
> But this brings up a somewhat-related question about PNG output. When
> I use WXSize=800 and WYSize=600 my PNG file ends up being 752 x 556
> pixels, and the bottom of my x axis title is chopped off. Should I not
> be getting 800 x 600 pixels out? And is my truncated title indicative
> of some mistake on my part? The PDF file shows the entire x axis
> title, but of course the file itself is 8.5 x 11 inches.

Well, I tested this pretty thoroughly before I posted, but I'm almost
certainly not using the code you are. Can you send me some example code
so we are comparing apples with apples?

I'm never surprised to find the Mac acting strangely, but this seems
reasonable straightforward to me.

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.")

Subject: Re: Coyote Graphics PS/PDF output size/orientation
Posted by Paul Levine on Wed, 14 Aug 2013 02:11:00 GMT
View Forum Message <> Reply to Message

On 2013-08-14 00:38:34 +0000, David Fanning said:

> Paul Levine writes:
>
>> I tried adding the WXSize and WYSize (though I used 800 for x and 600
>> for y because I am trying to get landscape output), but I still get the
>> same result, i.e. PNG oriented correctly and PDF rotated 90 degrees.
>>
>> But this brings up a somewhat-related question about PNG output. When
>> I use WXSize=800 and WYSize=600 my PNG file ends up being 752 x 556
>> pixels, and the bottom of my x axis title is chopped off. Should I not
>> be getting 800 x 600 pixels out? And is my truncated title indicative
>> of some mistake on my part? The PDF file shows the entire x axis
>> title, but of course the file itself is 8.5 x 11 inches.
>
> Well, I tested this pretty thoroughly before I posted, but I'm almost
> certainly not using the code you are. Can you send me some example code
> so we are comparing apples with apples?
>
> I'm never surprised to find the Mac acting strangely, but this seems
> reasonable straightforward to me.
>
> Cheers,
>
> David

Hi David,

I made some example code that simulates the plots I'm trying to make
(two data sets, two y axes, each with a title, plotted over time,

x-axis labels are month and year, so x axis title needs to get pushed down). The example creates both a png and a pdf for each of the following cases:

1. no resizing
2. resizing to 800 x 800
3. resizing to 800 x 600
4. resizing to 800 x 400
5. resizing to 600 x 800

I tried three different ways of resizing:
1. Calling cgWindow with xwsize and ywsize as the first command, followed by cgPlot with /addcmd
2. Calling cgPlot with xwsize and ywsize as arguments as the first command
3. Putting xwsize and ywsize as arguments in the new and improved cgControl

All three of these resize methods produced identical results, as I imagine they should, so I went with the third method

I suspect you might find all sorts of bad in my example code (even things unrelated to graphic file output); please feel free to point out anything I am doing wrong.

Also, I should mention, I am using:
Mac OS X 10.8.4
IDL 8.2.3
Coyote Graphics downloaded at 5:30 pm PDT on August 13, 2013
GPL Ghostscript 9.06
ImageMagick 6.8.6-6 (installed via MacPorts, and I included SetEnv, 'PATH=/opt/local/bin:$PATH' as you suggested on your webpage)

Example code:

```
time = indgen(100)*36 + 2451545 ; creates a vector of julian days from
2000-2009
date_label = label_date(date_format = ['%M!C%Y'])
data1 = (randomu(seed,100)*60)-30
data2 = (randomu(seed,100)*20)-10
cgPlot, time, data1 , thick=4, title = 'Title for Plot', /window, $
 yrange=[-50, 50], ytickinterval=10, yminor=1, ytitle = 'Data Value', $
 xtickunits = 'months', xtickformat = 'label_date', xticks = 20,
xtickinterval = 12, $
 xrange=[julday(1,1,2000),julday(1,1,2009)], xtitle = '!CDate'
cgAxis, Yaxis=1, yrange=[-10,10], ytickinterval=5, yminor=2, ytitle =
'Another Data Value', /window, /save
cgOplot, time, data2, linestyle=2, thick=1, /window
al_legend, ['Data 1', 'Data 2'], linestyle=[0,2], thick=[4,1], /window
```

```
cgControl, Output='Resize_none.pdf'
cgControl, Output='Resize_none.png'
cgControl, Resize=800, Output='Resize_800x800.pdf'
cgControl, Resize=800, Output='Resize_800x800.png'
cgControl, Resize=[800,600], Output='Resize_800x600.pdf'
cgControl, Resize=[800,600], Output='Resize_800x600.png'
cgControl, Resize=[800,400], Output='Resize_800x400.pdf'
cgControl, Resize=[800,400], Output='Resize_800x400.png'
cgControl, Resize=[600,800], Output='Resize_600x800.pdf'
cgControl, Resize=[600,800], Output='Resize_600x800.png'
```

The resulting PDF and PNG files can be seen at
https://www.dropbox.com/sh/3945ubx8b61xyoo/APfJsg-cl

But to summarize what I got:

All PDF files are 8.5 x 11 inches
Resize_none.pdf, Resize_800x600.pdf, and Resize_800x400.pdf are all rotated 90 degrees (i.e., I would need to rotate the PDF display 90 degrees clockwise to see the graphic correctly), and the graphic is framed at different sizes within the 8.5 x 11 inch page
Resize_800x800.pdf and Resize_600x800 show the graphic in the correct orientation; both are framed at different sizes within the 8.5 x 11 inch page.

All PNG files have the correct orientation (no rotation of the graphic)
Resize_none.png is 717 x 573 pixels
Resize_800x800.png is 573 x 573 pixels, and the right side y axis title is partially cut off
Resize_800x600.png is 742 x 556 pixels, and the x axis title is partially cut off
Resize_800x400.png is 742 x 371 pixels, the x axis title is completely cut off, and the x axis labels are partially cut off
Resize_600x800.png is 556 x 742 and the right side y axis title is partially cut off

Thank you for your help!
Paul

Subject: Re: Coyote Graphics PS/PDF output size/orientation
Posted by David Fanning on Wed, 14 Aug 2013 02:39:31 GMT
View Forum Message <> Reply to Message

Paul Levine writes:

> I made some example code that simulates the plots I'm trying to make

OK, I'll have to look at this more carefully in the morning, but my PDF files are definitely NOT rotated, as yours are. I'll see if I can figure out why that might be. It seems to me this has to be an ImageMagick thing, but I'll have to check to make sure. My version of ImageMagick is 6.7.6-3.

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.")

Subject: Re: Coyote Graphics PS/PDF output size/orientation
Posted by David Fanning on Wed, 14 Aug 2013 02:43:02 GMT
View Forum Message <> Reply to Message

David Fanning writes:

> It seems to me this has to be an ImageMagick
> thing, but I'll have to check to make sure. My version of ImageMagick is
> 6.7.6-3.

Or, now that I think about it, maybe a Ghostscript thing. PDF files will be handled by Ghostscript, rather than ImageMagick itself, I think. My GhostScript is 9.05, different from yours.

Sigh... Well, lot's of things to check. :-)

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.")

Subject: Re: Coyote Graphics PS/PDF output size/orientation
Posted by David Fanning on Wed, 14 Aug 2013 02:48:44 GMT
View Forum Message <> Reply to Message

David Fanning writes:

> Or, now that I think about it, maybe a Ghostscript thing. PDF files will
> be handled by Ghostscript, rather than ImageMagick itself, I think. My
> GhostScript is 9.05, different from yours.
>
> Sigh... Well, lot's of things to check. :-)

OK, here is something. On Macs, I don't use Ghostscript. I use the built in (I believe) command pstopdf. What happens if you comment out the "DARWIN" section of the case statement in cgps2pdf.pro and use the UNIX case instead. Do you have the same results?

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.")

Subject: Re: Coyote Graphics PS/PDF output size/orientation
Posted by Paul Levine on Wed, 14 Aug 2013 03:24:42 GMT
View Forum Message <> Reply to Message

On 2013-08-14 02:48:44 +0000, David Fanning said:

> David Fanning writes:
>
>> Or, now that I think about it, maybe a Ghostscript thing. PDF files will
>> be handled by Ghostscript, rather than ImageMagick itself, I think. My
>> GhostScript is 9.05, different from yours.
>>
>> Sigh... Well, lot's of things to check. :-)
>
> OK, here is something. On Macs, I don't use Ghostscript. I use the built
> in (I believe) command pstopdf. What happens if you comment out the
> "DARWIN" section of the case statement in cgps2pdf.pro and use the
> UNIX case instead. Do you have the same results?

>
> Cheers,
>
> David

OK, first I commented out the entire "DARWIN" case (lines 190-199) in cgps2pdf and ran my example again, but no PDF files were produced (I did not get any error message, and I did get the "PDF output will be created here:" message, but no PDF files in the directory).

Then, I found an IF case on line 244
```
IF StrUpCase(!Version.OS) EQ 'DARWIN' THEN BEGIN
      cmd = gs_exe + " " + ps_file[0] + " -o " + pdf_file
```

So I changed 'DARWIN' to 'foo' to force it to go else, and ran the example code again. This time, the Resize_600x800.pdf and Resize_800x800.pdf both came out the same as they had previously been (those were the two that were ending up in a portrait orientation from the beginning), as 8.5 x 11 inch pages. However, the other three, Resize_none.pdf, Resize_800x600.pdf, and Resize_800x400.pdf all came out as 11 x 8.5 inch pages, properly oriented, and all framed differently on the page. In other words, they were identical to what I had gotten before after rotating 90 degrees clockwise.

I'm not exactly sure what the intended behavior is; perhaps the results I got forcing the UNIX case is what you had intended, with all PDF files ending up on a US letter sized page in either portrait or landscape orientation (8.5 x 11 inches or 11 x 8.5 inches). From my extremely limited knowledge of this stuff, it would seem that any PS conversion would turn out like this unless it was an encapsulated PS file, in which case it could be any size.

Also, I am quite interested in your take on the PNG files I wound up with, where the pixel size of the files was quite different from the xwsize and ywsize specified, and some axis titles and labels were cut off. I don't how related this is to the PDF issue, so if you can only deal with one of these issues at a time, then perhaps you need to pass the other one off to Coyote and see where he gets with it ;-)

Thanks,
Paul

Subject: Re: Coyote Graphics PS/PDF output size/orientation
Posted by David Fanning on Wed, 14 Aug 2013 03:51:04 GMT
View Forum Message <> Reply to Message

Paul Levine writes:

> Also, I am quite interested in your take on the PNG files I wound up
> with, where the pixel size of the files was quite different from the
> xwsize and ywsize specified, and some axis titles and labels were cut
> off. I don't how related this is to the PDF issue, so if you can only
> deal with one of these issues at a time, then perhaps you need to pass
> the other one off to Coyote and see where he gets with it ;-)

Yeah, I'll talk to Coyote about this, but he has me drinking gin and
tonics tonight to celebrate something or other. It will have to be in
the morning. Oh, I remember, birthdays! We'll sort it out in the
morning. :-)

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.")

Subject: Re: Coyote Graphics PS/PDF output size/orientation
Posted by Paul Levine on Wed, 14 Aug 2013 04:47:41 GMT
View Forum Message <> Reply to Message

On 2013-08-14 03:51:04 +0000, David Fanning said:

> Paul Levine writes:
>
>> Also, I am quite interested in your take on the PNG files I wound up
>> with, where the pixel size of the files was quite different from the
>> xwsize and ywsize specified, and some axis titles and labels were cut
>> off. I don't how related this is to the PDF issue, so if you can only
>> deal with one of these issues at a time, then perhaps you need to pass
>> the other one off to Coyote and see where he gets with it ;-)
>
> Yeah, I'll talk to Coyote about this, but he has me drinking gin and
> tonics tonight to celebrate something or other. It will have to be in
> the morning. Oh, I remember, birthdays! We'll sort it out in the
> morning. :-)
>
> Cheers,
>
> David

Ah, well, in that case, happy birthday! I hope for your sake that you don't see this message until tomorrow morning, but once you do, I have something interesting to report on PNG conversion. It seems that to a certain extent, cgps2raster creates PNG files of a certain size that is determined by the aspect ratio rather than the wxsize/wysize or the cgControl resize keyword. So, for example, any aspect ratio of 4:3 will produce a PNG of 742 x 556 pixels, and any aspect ratio of 1:1 will prodice a PNG of 573 x 573 pixels.

So when I appended my code with the following lines:

```
cgControl, Resize=1, Output='Resize_1x1.png'
cgControl, Resize=50, Output='Resize_50x50.png'
cgControl, Resize=800, Output='Resize_800x800.png'
cgControl, Resize=8000, Output='Resize_8000x8000.png'
cgControl, Resize=25000, Output='Resize_25000x25000.png'
```

each of the resulting PNG files was 573 x 573 pixels.

Now, here's where it gets really interesting. I said "to a certain extent" before because once the size is above a certain threshold, then a different sized file is produced. Resize=32767 and any number below gives me the 573 x 573, but Resize=32768 and any number above that produces a PNG file that is 742 x 390 pixels . Of course, it can't be a coincidence that the threshold I did pinpoint is the 15th power of 2. Just for laughs, I decided to try finding out whether there was an even higher threshold. So I tried Resize=2.14748e+09 (2^31), which was still 742 x 390, then I tried to go even higher than that and ended up getting a "Floating illegal operand" error from cgPlot. So I guess cgPlot is not able to handle more than about 2.14748 x 2.14748 billion pixels ;-P

Tomorrow, I will hopefully have access to a windows machine, which I can use for testing

---

Subject: Re: Coyote Graphics PS/PDF output size/orientation
Posted by David Fanning on Wed, 14 Aug 2013 12:24:19 GMT
View Forum Message <> Reply to Message

Paul Levine writes:

>
> On 2013-08-14 03:51:04 +0000, David Fanning said:
>
>> Paul Levine writes:
>>

>>> Also, I am quite interested in your take on the PNG files I wound up
>>> with, where the pixel size of the files was quite different from the
>>> xwsize and ywsize specified, and some axis titles and labels were cut
>>> off. I don't how related this is to the PDF issue, so if you can only
>>> deal with one of these issues at a time, then perhaps you need to pass
>>> the other one off to Coyote and see where he gets with it ;-)
>>
>> Yeah, I'll talk to Coyote about this, but he has me drinking gin and
>> tonics tonight to celebrate something or other. It will have to be in
>> the morning. Oh, I remember, birthdays! We'll sort it out in the
>> morning. :-)
>>
>> Cheers,
>>
>> David
>
> Ah, well, in that case, happy birthday! I hope for your sake that you
> don't see this message until tomorrow morning, but once you do, I have
> something interesting to report on PNG conversion. It seems that to a
> certain extent, cgps2raster creates PNG files of a certain size that is
> determined by the aspect ratio rather than the wxsize/wysize or the
> cgControl resize keyword. So, for example, any aspect ratio of 4:3
> will produce a PNG of 742 x 556 pixels, and any aspect ratio of 1:1
> will prodice a PNG of 573 x 573 pixels.
>
> So when I appended my code with the following lines:
>
> cgControl, Resize=1, Output='Resize_1x1.png'
> cgControl, Resize=50, Output='Resize_50x50.png'
> cgControl, Resize=800, Output='Resize_800x800.png'
> cgControl, Resize=8000, Output='Resize_8000x8000.png'
> cgControl, Resize=25000, Output='Resize_25000x25000.png'
>
> each of the resulting PNG files was 573 x 573 pixels.
>
> Now, here's where it gets really interesting. I said "to a certain
> extent" before because once the size is above a certain threshold, then
> a different sized file is produced. Resize=32767 and any number below
> gives me the 573 x 573, but Resize=32768 and any number above that
> produces a PNG file that is 742 x 390 pixels . Of course, it can't be
> a coincidence that the threshold I did pinpoint is the 15th power of 2.
> Just for laughs, I decided to try finding out whether there was an
> even higher threshold. So I tried Resize=2.14748e+09 (2^31), which was
> still 742 x 390, then I tried to go even higher than that and ended up
> getting a "Floating illegal operand" error from cgPlot. So I guess
> cgPlot is not able to handle more than about 2.14748 x 2.14748 billion
> pixels ;-P
>

> Tomorrow, I will hopefully have access to a windows machine, which I
> can use for testing

Well, OK, I think I can explain this. Even with a slight headache and a thick tongue.

The aspect ratio is key because the way the PNG file gets made is with a PostScript intermediary file. To maintain the highest resolution, I try to create the largest possible "window" on the PostScript page that maintains that aspect ratio. This "high resolution" PostScript file is then "reduced" in size or resolution when the raster file is created by ImageMagick. The default reduction is to 25% of its original size, which gives me a PNG image, say, that is about the size of a normal IDL graphics window when I open it with my image software (Hypersnap, typically). The amount of size reduction is user controlled by means of the IM_RESIZE keyword in cgWindow_SetDefs and cgControl.

Note that function graphics raster files are always HUGE. This is because they don't apply this size reduction as their default. You have to specify it as a keyword. (One of many things that bug me about function graphics.) Coyote Graphics routines produce comparably-sized files if you set the IM_RESIZE keyword to 100.

http://www.idlcoyote.com/cg_tips/compcont.php

To give a little more control over the absolute dimensions of the raster file output, I have added a WIDTH keyword to PS_END (configurable with the IM_WIDTH keyword for cgWindows). This allows the user to specify a specific width resolution for the raster file. A width of, say, 800 creates a PNG file with a width of 800 pixels and a height based on the aspect ratio of the intermediary PostScript "window". This is handy for me when I am creating PNG output to put on my web page, for example, because I can get the width set exactly as I want it.

I honestly don't know why the resolution of the PNG files change at a certain huge window size. But, I presume it has something to do with the way I compute the largest PostScript "window". I'm probably using short integers to do the computations, since the possibility of someone wanting a graphics window of 32676 pixels never occurred to me. ;-)

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.")

Subject: Re: Coyote Graphics PS/PDF output size/orientation
Posted by David Fanning on Wed, 14 Aug 2013 12:59:15 GMT
View Forum Message <> Reply to Message

Paul Levine writes:

> Also, I am quite interested in your take on the PNG files I wound up
> with, where the pixel size of the files was quite different from the
> xwsize and ywsize specified, and some axis titles and labels were cut
> off. I don't how related this is to the PDF issue, so if you can only
> deal with one of these issues at a time, then perhaps you need to pass
> the other one off to Coyote and see where he gets with it ;-)

Unfortunately, as much as I try, conversion from IDL's graphics window output to PostScript output is not *exactly* WYSIWYG. One of the difficulties is on the right-hand side of plots. The default IDL position of plots does not leave enough room in the graphics "window" to accommodate axis annotation, etc. This is why part of your plot is getting cut off. Some of it extends outside this "window".

I have tried to give Coyote Graphics routines a better default position in the window (ie, position = [0.125, 0.125, 0.925, 0.9]), but there is no way for me to know when I do this whether or not you are going to do something different on the right-hand side of the plot. You have to be responsible for that yourself.

The solution is simple enough: use the POSITION keyword on your plot command to give yourself a little more space on the right-hand side of the plot to accommodate axis annotation:

```
  cgPlot, data, Position=[0.125, 0.125, 0.85, 0.925], ...
```

Cheers,

David

P.S. I had to figure this out on my own, since Coyote won't be awake for another 3-4 more hours, at least.

--
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.")

Subject: Re: Coyote Graphics PS/PDF output size/orientation
Posted by David Fanning on Wed, 14 Aug 2013 17:10:44 GMT

David Fanning writes:

> OK, here is something. On Macs, I don't use Ghostscript. I use the built
> in (I believe) command pstopdf. What happens if you comment out the
> "DARWIN" section of the case statement in cgps2pdf.pro and use the
> UNIX case instead. Do you have the same results?

So the use of pstopdf turns out to be the culprit in the mysterious "rotation" problem. Unfortunately, pstopdf doesn't appear to support any rotation options, so I have elected to have Macs use the same ImageMagick/Ghostscript method other UNIX flavors use. I left the old code intact, but commented out, however, in case anyone prefers the old methods.

You can find an updated program that does the rotation the way I expect it to be done here:

http://www.idlcoyote.com/programs/cgps2pdf.pro

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.")

Subject: Re: Coyote Graphics PS/PDF output size/orientation
Posted by Paul Levine on Wed, 14 Aug 2013 17:28:35 GMT

On 2013-08-14 17:10:44 +0000, David Fanning said:

> David Fanning writes:
>
>> OK, here is something. On Macs, I don't use Ghostscript. I use the built
>> in (I believe) command pstopdf. What happens if you comment out the
>> "DARWIN" section of the case statement in cgps2pdf.pro and use the
>> UNIX case instead. Do you have the same results?
>
> So the use of pstopdf turns out to be the culprit in the mysterious

> "rotation" problem. Unfortunately, pstopdf doesn't appear to support any
> rotation options, so I have elected to have Macs use the same
> ImageMagick/Ghostscript method other UNIX flavors use. I left the old
> code intact, but commented out, however, in case anyone prefers the old
> methods.
>
> You can find an updated program that does the rotation the way I expect
> it to be done here:
>
>   http://www.idlcoyote.com/programs/cgps2pdf.pro
>
> Cheers,
>
> David

Excellent, thanks for the update. It seems that like many things
Apple, the original UNIX way provides the best option.

Subject: Re: Coyote Graphics PS/PDF output size/orientation
Posted by Fabzi on Wed, 14 Aug 2013 19:09:00 GMT
View Forum Message <> Reply to Message

Hi,

On 08/14/2013 07:10 PM, David Fanning wrote:
> Unfortunately, pstopdf doesn't appear to support any
> rotation options

I didn't follow the whole conversation so I don't know if that helps,
but: epstopdf has a --Autorotate=All option which works for me most of
the time. And epstopdf also works with ps files...

cheers
