Subject: Re: plot and contour inconsistency
Posted by Matthew Argall on Sun, 10 Mar 2013 06:24:06 GMT
View Forum Message <> Reply to Message

Also, the documentation for the C_SPACING keyword says,

"If the FILL or CELL_FILL keyword is set, this keyword can be used to control the distance, in centimeters, between the lines used to fill the contours"

but c_spacing works even if those keywords are not set...

c = dist(256)
contour, c, FILL=0, CELL_FILL=0, C_SPACING=0

Same goes with C_ORIENTATION

contour, c, FILL=0, CELL_FILL=0, C_ORIENTATION=45

Subject: Re: plot and contour inconsistency
Posted by Jeremy Bailin on Sun, 10 Mar 2013 15:37:48 GMT
View Forum Message <> Reply to Message

On 3/10/13 12:24 AM, Matthew Argall wrote:
> Also, the documentation for the C_SPACING keyword says,
>
> "If the FILL or CELL_FILL keyword is set, this keyword can be used to control the distance, in centimeters, between the lines used to fill the contours"
>
> but c_spacing works even if those keywords are not set...
>
>
> c = dist(256)
> contour, c, FILL=0, CELL_FILL=0, C_SPACING=0
>
> Same goes with C_ORIENTATION
>
> contour, c, FILL=0, CELL_FILL=0, C_ORIENTATION=45
>

Hmmp. I'd call that a bug, personally - you shouldn't be able to over-ride explicit fill settings just by changing the parameters of what would get filled. Certainly not if it doesn't say so in the documentation!

-Jeremy.

Subject: Re: plot and contour inconsistency
Posted by David Fanning on Sun, 10 Mar 2013 15:46:58 GMT
View Forum Message <> Reply to Message

Jeremy Bailin writes:

> Hmmp. I'd call that a bug, personally - you shouldn't be able to
> over-ride explicit fill settings just by changing the parameters of what
> would get filled. Certainly not if it doesn't say so in the documentation!

Yes, and there will be no chance of this getting fixed, so I think you
are just going to have to learn to live with it. If it is something you
need badly, I could fix it in cgContour (by undefining C_Spacing if
either of the fill keywords was set). Let me know.

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: plot and contour inconsistency
Posted by Matthew Argall on Sun, 10 Mar 2013 17:02:53 GMT
View Forum Message <> Reply to Message

> If it is something you
> need badly, I could fix it in cgContour (by undefining C_Spacing if
> either of the fill keywords was set). Let me know.

I have done that in this program, so it is not necessary yet. I might take you up on your offer in the future, though, when I start upgrading this thing to use the cg library ;-)

I also changed the default MAX_VALUE and MIN_VALUE to +/- !values.f_infinity and it works fine (for any of those who were wondering). I figured !values.f_nan would work because NaN's are suppose to be ignored.

Subject: Re: plot and contour inconsistency
Posted by David Fanning on Sun, 10 Mar 2013 17:10:51 GMT
View Forum Message <> Reply to Message

Matthew Argall writes:

> I also changed the default MAX_VALUE and MIN_VALUE to +/- !values.f_infinity and it works fine (for any of those who were wondering). I figured !values.f_nan would work because NaN's are suppose to be ignored.

This is kind of an odd way to write a program. What are you hoping to accomplish by doing this?

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: plot and contour inconsistency
Posted by Matthew Argall on Sun, 10 Mar 2013 20:09:27 GMT
View Forum Message <> Reply to Message

> This is kind of an odd way to write a program. What are you hoping to
> accomplish by doing this?

There is a program that makes survey plots of satellite data that has a window that allows me to switch between time periods. I wanted to be able to change a few of the default attributes of the plots for when the range of the data changes by a lot. Also, the bad values are not always weeded out, so there is often a few -1e-31 that throw everything off.

In order to set the default attributes, I filled a structure with a small set of keywords accepted by plot then made a setProperties method to change them. Every time the plot is updated, PLOT now gets called something like

PLOT, x, y, _EXTRA=extra

where "extra" is the structure of properties I want to change.

In for this to work, though, I have to default to the IDL defaults. MAX_VALUE and MIN_VALUE do not have defaults, so I chose +/- !values.f_infinity. Later, I can changed them so that the -1e-31's do not show up.

Subject: Re: plot and contour inconsistency
Posted by David Fanning on Sun, 10 Mar 2013 20:31:07 GMT
View Forum Message <> Reply to Message

Matthew Argall writes:

>
>> This is kind of an odd way to write a program. What are you hoping to
>> accomplish by doing this?
>
> There is a program that makes survey plots of satellite data that has a window that allows me to switch between time periods. I wanted to be able to change a few of the default attributes of the plots for when the range of the data changes by a lot. Also, the bad values are not always weeded out, so there is often a few -1e-31 that throw everything off.
>
> In order to set the default attributes, I filled a structure with a small set of keywords accepted by plot then made a setProperties method to change them. Every time the plot is updated, PLOT now gets called something like
>
> PLOT, x, y, _EXTRA=extra
>
> where "extra" is the structure of properties I want to change.
>
> In for this to work, though, I have to default to the IDL defaults. MAX_VALUE and MIN_VALUE do not have defaults, so I chose +/- !values.f_infinity. Later, I can changed them so that the -1e-31's do not show up.

Ah, then let me suggest a different way of doing this. The cgGraphicsKeywords object can keep track of of the "graphics keywords" required by Plot, Contour, etc. But, the nice thing about this object is that when asked for the keywords, if a keyword hasn't been defined, then the keyword value for that keyword will be an undefined variable. In other words, you don't *have* to define a default value for each and every keyword. This is because all the keywords are stored as pointers, and a pointer can return an undefined variable.

I use this to keep track of plot keywords in cgZPlot, for example. It works really well. This was the first step required to turn the Coyote Graphics routines into objects. The cgZPlot program was the first proof of concept program. Since then, I haven't had much time to work on it, and no one seems to be clamoring for Coyote Graphics objects, so I haven't worried too much about it. But, you might be able to use to good effect. :-)

See the DrawPlot method in cgZPlot__Define to see how I use it.

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: plot and contour inconsistency
Posted by Matthew Argall on Sun, 10 Mar 2013 21:16:15 GMT
View Forum Message <> Reply to Message

On Sunday, March 10, 2013 4:31:07 PM UTC-4, David Fanning wrote:
> Matthew Argall writes:
>
>
>
>>
>
>>> This is kind of an odd way to write a program. What are you hoping to
>
>>> accomplish by doing this?
>
>>
>
>> There is a program that makes survey plots of satellite data that has a window that allows me to switch between time periods. I wanted to be able to change a few of the default attributes of the plots for when the range of the data changes by a lot. Also, the bad values are not always weeded out, so there is often a few -1e-31 that throw everything off.
>
>>
>
>> In order to set the default attributes, I filled a structure with a small set of keywords accepted by plot then made a setProperties method to change them. Every time the plot is updated, PLOT now gets called something like
>
>>
>
>> PLOT, x, y, _EXTRA=extra
>
>>
>
>> where "extra" is the structure of properties I want to change.
>
>>
>
>> In for this to work, though, I have to default to the IDL defaults. MAX_VALUE and MIN_VALUE do not have defaults, so I chose +/- !values.f_infinity. Later, I can changed them so

that the -1e-31's do not show up.
>
>
>
> Ah, then let me suggest a different way of doing this. The
>
> cgGraphicsKeywords object can keep track of of the "graphics keywords"
>
> required by Plot, Contour, etc. But, the nice thing about this object is
>
> that when asked for the keywords, if a keyword hasn't been defined, then
>
> the keyword value for that keyword will be an undefined variable. In
>
> other words, you don't *have* to define a default value for each and
>
> every keyword. This is because all the keywords are stored as pointers,
>
> and a pointer can return an undefined variable.
>
>
>
> I use this to keep track of plot keywords in cgZPlot, for example. It
>
> works really well. This was the first step required to turn the Coyote
>
> Graphics routines into objects. The cgZPlot program was the first proof
>
> of concept program. Since then, I haven't had much time to work on it,
>
> and no one seems to be clamoring for Coyote Graphics objects, so I
>
> haven't worried too much about it. But, you might be able to use to good
>
> effect. :-)
>
>
>
> See the DrawPlot method in cgZPlot__Define to see how I use it.
>
>
>
> 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.")

That sounds amazing, actually, and will put it to good use.

Subject: Re: plot and contour inconsistency
Posted by Matthew Argall on Sat, 20 Apr 2013 03:08:21 GMT
View Forum Message <> Reply to Message

> cgGraphicsKeywords object can keep track of of the "graphics keywords"
>
> required by Plot, Contour, etc.
>
>
>
> I use this to keep track of plot keywords in cgZPlot, for example.
>
> But, you might be able to use to good
>
> effect. :-)
>
> See the DrawPlot method in cgZPlot__Define to see how I use it.

This is quite great! I had been doing something similar, but the /ALLOCATE_HEAP keyword eluded me...

I was wondering, in the cgZPlot::DrawPlot method, is there a reason you call cgGraphicsKeywords::GetProperty? Why can't you get all of the properties by doing simply, e.g., *self.max_value, since cgGraphicsKeywords is inherited?

Also, why do you always draw to the pixmap then copy to the display?
