
Subject: Re: Contour: difference between IDL 8.0 and IDL 8.4
Posted by [Jim Pendleton](#) on Sun, 19 Feb 2017 03:55:02 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Friday, February 17, 2017 at 10:14:48 AM UTC-7, burkina wrote:

> On Friday, 17 February 2017 17:45:59 UTC+1, burkina wrote:

>> On Friday, 17 February 2017 16:30:47 UTC+1, Jim P wrote:

>>> On Friday, February 17, 2017 at 2:55:48 AM UTC-7, burkina wrote:

>>>> On Thursday, 16 February 2017 12:59:39 UTC+1, burkina wrote:

>>>> > Ciao,

>>>> >

>>>> > I've upgraded from IDL 8.0 to IDL 8.4, and found a strange difference in the behaviour of CONTOUR.

>>>> >

>>>> > Here's a simple test:

>>>> >

>>>> > data = RANDOMU(seed, 9, 9)

>>>> > smooth = CONTOUR(MIN_CURVE_SURF(data), TITLE='Smoothed', RGB_TABLE=1,
/FILL, c_value=[0, 0.2, 0.3, 0.5])

>>>> >

>>>> > If you launch it in IDL 8.0, you get what you (more or less) expect: three filled contours, while the outer part is the white transparent background. I need to include the '0' level, otherwise the inner contour is left blank (BTW, I find it odd, and it is the same for IDL 8.4).

>>>> >

>>>> > On the other hand, if you launch it in IDL 8.4, there's a big difference: also the outer part is filled! Of course, I can give an explicit white color to it, but it's not the same, because it is not treated as a completely transparent background. So, when you put a contour above another, in IDL 8.0 everything is fine, because the outer part is completely transparent, while in IDL 8.4, you see also this outer part. Using 'max_value=0.5' is not an equivalent solution.

>>>> >

>>>> > What changed between 8.0 and 8.4? How can I revert to the 8.0 behaviour? I basically want to draw three filled contours, and leave the white transparent background in the rest of the image.

>>>> >

>>>> > Thanks,

>>>> >

>>>> > Stefano

>>>>

>>>> Hi,

>>>>

>>>> just to be more specific, this is a plot I had in IDL 8.0:

>>>>

>>>> <https://img42.com/MOnas>

>>>>

>>>> and this is what I get (exactly with the same code) in IDL 8.4:

>>>>

>>>> <https://img42.com/KSkRc>

>>>>

>>>> As you can see, in the latter the background of the reddish contour plot is still visible, because it's indeed a filled contour level, while in IDL 8.0 it is simply background. How can I get rid of it?

>>>>

>>>> Something clearly changed after IDL 8.0, I hope there's some kind of switch/parameter that I don't understand that can be set to have the old behaviour. I even thought about reconstructing the contour levels polygons, and then fill them, but, apparently, there's no way to get the polygons in function graphics (!?).

>>>>

>>>> Thanks,

>>>>

>>>> Stefano

>>>>

>>>> P.S. I had a chance to test this code on IDL 8.2, and it behaves like 8.4.

>>>

>>> I don't have a specific solution, but here are a couple things to investigate.

>>>

>>> According to the docs for the CONTOUR function, <http://www.harrisgeospatial.com/docs/CONTOUR.html>, a number of keywords were added in IDL 8.2 that may address this. These include BACKGROUND_COLOR and BACKGROUND_TRANSPARENCY, and probably changed the default behaviors.

>>>

>>> Comparing the contour properties in 8.0 versus later versions may help, as well, e.g.,

>>>

>>> IDL> print, smooth

>>>

>>> For example, the "c_color" property represented a color table index in 8.0 [0, 47, 73, 125], while in later versions c_color contains RGB triplets, [[0, 0, 13], [0, 13, 141], ...].

>>

>> Dear Jim,

>>

>> thank you for your reply.

>>

>> As for the new keywords, BACKGROUND_COLOR and BACKGROUND_TRANSPARENCY, they have no effect in my case, because they simply change the background color and transparency (as expected), but my problem is that in 8.2+ that part is not background, but a filled contour!

>>

>> On the other hand, I had a look at the contour properties, as you suggested, and it seems to me that there are two main differences.

>>

>> 1. IDL 8.0: N_LEVELS = 4 -- IDL 8.4: N_LEVELS = 0

>>

>> If I include a n_levels=3 keyword in the command line, nothing changes for IDL 8.4 (as expected, the c_values are dominant), while in IDL 8.0 the c_values are ignored (which is not what I would expect). I don't know how this could affect my issue

>>

>> 2. As you correctly pointed out, c_color is different. So I changed the keyword in

```

>>
>> c_value=[level_0,level_1, level_2, level_3], c_color=['blue', 'green', 'red']
>>
>> which still reproduces the issue I'm trying to solve, but it seems to me easier to interpret. And
here's a hint:
>>
>> IDL 8.0: print, smooth.c_color
>>
>> 0 0 255
>> 0 127 0
>> 255 0 0
>>
>> IDL8.4: print, smooth.c_color
>>
>> 0 0 255
>> 0 127 0
>> 255 0 0
>> 0 0 255
>> 0 127 0
>>
>> There are indeed two further colors defined! I think this is why the contours are filled also
outside. No matter if I define less colors, always n+1 colors are stored in c_color (repeated
cyclically). So how can I delete these two colors and have only background after the first three
contours?
>>
>> Further ideas?
>>
>> In any case, I found the fill contour implementation in IDL counter-intuitive, because it seems
to me that it actually fills the outside of the contour, not the inside...
>>
>> Thanks,
>>
>> Stefano
>
> As a further test, I re-define c_color after the plot:
>
> print, c2.c_color
>
> help, c2.c_color
>
> c2.c_color=[[0,0, 255],[0 ,127 , 0],[255, 0 , 0]]
>
> print, c2.c_color
>
> help, c2.c_color
>
> But, although now I only have three colors as in IDL8.0, nothing changes in the plot:
>

```

```
> 0 0 255
> 0 127 0
> 255 0 0
> 0 0 255
> 0 127 0
> <Expression>  BYTE    = Array[3, 5]
> 0 0 255
> 0 127 0
> 255 0 0
> <Expression>  BYTE    = Array[3, 3]
```

One other debugging trick is comparing the source code of the two classes IDLitVISContour and IDLitVISContourLevel, between the version 8.0 and 8.X.

I found that if I executed the contour() function in IDL 8.5 first (to ensure the routines were compiled), then recompiled the two class definition files from my IDL 8.0 distribution on the same computer, my IDL 8.5 results now look like my IDL 8.0 results.

They are in the lib\itools\components subdirectory of each distribution.

The challenge, then, is to find the causal difference between the two versions. I don't know if it's necessary to investigate both classes. Only one of the two may be the problem.

If you open two projects in your IDL Workbench, one based in the root of the IDL 8.0 lib directory and one in that of IDL 8.X, you can multi-select the two source files for each class then execute a "compare" via a right click in the Project Explorer window.

Jim P.
