
Subject: Re: cgcolor, cgplot, and /WINDOW

Posted by [David Fanning](#) on Wed, 23 Mar 2011 12:32:13 GMT

[View Forum Message](#) <> [Reply to Message](#)

FÖLDY Lajos writes:

```
> I usually use 'BBGGRR'x numbers for 24-bit colors, as suggested by the IDL
> documentation. I use decomposed mode, but cgPlot still has some problems
> with it:
>
> IDL> device, /decomposed
> IDL> cgplot, dist(10), color='FF0000'x, /window ; blue, OK
> IDL> cgplot, dist(10), color='00FF00'x, /window ; green, OK
> IDL> cgplot, dist(10), color='0000FF'x, /window ; black, ???
> IDL> cgplot, dist(10), color='005050'x, /window ; error
>
> Traceback Report from CGPLOT:
>
> % Attempt to subscript R with C2 is out of range.
> % Execution halted at: COLORSAREIDENTICAL 121 colorsareidentical.pro
> %          CGPLOT          384 cgplot.pro
> %          FSC_WINDOW_COMMAND::DRAW 1512 cgwindow.pro
> %          FSC_CMDWINDOW::EXECUTECOMMANDS 399 cgwindow.pro
> %          CGWINDOW        2017 cgwindow.pro
> %          CGPLOT          248 cgplot.pro
> %          $MAIN$
> cgPlot, p1, COLOR=value, NOERASE=value
>
> It seems to be that cgPlot uses color tables even in decomposed mode. Or I
> am breaking your (1) rule simply :-)
>
> regards,
> Lajos
>
> ps: after writing the above, I have found the solution: 24-bit colors must
> be specified as long numbers, '0000FF'xl and '005050'xl works.
```

Yes, the **proper** way to specify colors in Coyote Graphics routines is to use color names (there are more than 200).

But, I am well aware there are anarchists among us (mostly Mac users, I think). So the programs are written so that they will accept other ways of specifying colors.

But, by bending the rules like this, I am forced to live in ambiguity. Suppose, for example, I get a "color" value of 248. What am I to make of this? Is this an index into the current color table? If so, I can grab the color vectors, extract the RGB values at index 248 and make a 24-bit color

from it. But who is to say this is not **already** a 24-bit value for a reddish color?

The only way I can "tell" is by the type of number it is. If the number is an INT or a BYTE, I simply assume it is an index to the color table. If it is a LONG, I assume it is something I can decompose.

You found the right colors if you passed longs. Yesterday, I had a user with the opposite problem. She was passing longs, but meant to be passing ints. She could get nothing but red colors on her display.

Object graphics **forces** you to pass unambiguous colors by requiring three-element arrays. But, this is so "unnatural" to many users, that I think it has delayed the acceptance of object graphics routines. I happen to think color names are the "natural" way to specify colors, but I am willing to live with some ambiguity if it means more users will migrate to using better and more useful software.

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.")
