Subject: Re: cgcolor, cgplot, and /WINDOW
Posted by David Fanning on Wed, 23 Mar 2011 04:01:04 GMT
View Forum Message <> Reply to Message

Kwill writes:

> I was excited to learn today of David's Coyote Graphics packages, and
> downloaded them on both my laptop and desktop. I'm getting very
> puzzling results on one machine (but not the other). When I set the /
> WINDOW keyword in cgplot, cgcolor will not work for overplotting in
> color, displaying all subsequent points in a dark grey.
>
> My example:
>
> IDL> cgplot, indgen(10),color=cgcolor("Dodger Blue"), /window
> IDL> cgplot, indgen(10)+1, color=cgcolor("Hot Pink"), /window, /over
>
> This results in one blue and one grey line plotted on the screen. I do
> not have any color problems when the /WINDOW keyword is not set.
>
> I tried an identical series of commands on my laptop: it runs
> perfectly, producing both a blue and pink line. Both of my machines
> run IDL Version 7.1.1, Mac OS X (darwin x86_64 m64) on Snow Leopard
> 10.6.4.
>
> I'm baffled as to why this would be different between my two machines,
> but hopeful that I'm doing something wrong and that CGCOLOR plays
> nicely with the resizeable window in CGPLOT. Any ideas?

What happens if you use ADD instead of WINDOW on the second
command?

```
IDL> cgplot, indgen(10),color=cgcolor("Dodger Blue"), /window
IDL> cgplot, indgen(10)+1, color=cgcolor("Hot Pink"), /add, /over
```

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.")

Subject: Re: cgcolor, cgplot, and /WINDOW

Posted by David Fanning on Wed, 23 Mar 2011 04:07:37 GMT
View Forum Message <> Reply to Message

Kwill writes:

> I was excited to learn today of David's Coyote Graphics packages, and
> downloaded them on both my laptop and desktop. I'm getting very
> puzzling results on one machine (but not the other). When I set the /
> WINDOW keyword in cgplot, cgcolor will not work for overplotting in
> color, displaying all subsequent points in a dark grey.
>
> My example:
>
> IDL> cgplot, indgen(10),color=cgcolor("Dodger Blue"), /window
> IDL> cgplot, indgen(10)+1, color=cgcolor("Hot Pink"), /window, /over

I'll have to look to see what the gray color is about,
but this is the wrong way to call these commands. The
correct way is like this:

> IDL> cgplot, indgen(10),color="Dodger Blue", /window
> IDL> cgplot, indgen(10)+1, color="Hot Pink", /window, /over

I think when you pass Hot Pink through cgColor twice, the
value turns into a long integer, which gets interpreted
incorrectly. It's late here, so I'm not thinking as
clearly as I usually am. :-)

I'll look into 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 thui. ("Perhaps thou speakest truth.")

---

Subject: Re: cgcolor, cgplot, and /WINDOW
Posted by willettk on Wed, 23 Mar 2011 04:09:57 GMT
View Forum Message <> Reply to Message

On Mar 22, 10:01 pm, David Fanning <n...@idlcoyote.com> wrote:
> Kwill writes:
>> I was excited to learn today of David's Coyote Graphics packages, and
>> downloaded them on both my laptop and desktop. I'm getting very

>> puzzling results on one machine (but not the other). When I set the /
>> WINDOW keyword in cgplot, cgcolor will not work for overplotting in
>> color, displaying all subsequent points in a dark grey.
>
>> My example:
>
>> IDL> cgplot, indgen(10),color=cgcolor("Dodger Blue"), /window
>> IDL> cgplot, indgen(10)+1, color=cgcolor("Hot Pink"), /window, /over
>
>> This results in one blue and one grey line plotted on the screen. I do
>> not have any color problems when the /WINDOW keyword is not set.
>
>> I tried an identical series of commands on my laptop: it runs
>> perfectly, producing both a blue and pink line. Both of my machines
>> run IDL Version 7.1.1, Mac OS X (darwin x86_64 m64) on Snow Leopard
>> 10.6.4.
>
>> I'm baffled as to why this would be different between my two machines,
>> but hopeful that I'm doing something wrong and that CGCOLOR plays
>> nicely with the resizeable window in CGPLOT. Any ideas?
>
> What happens if you use ADD instead of WINDOW on the second
> command?
>
>  IDL> cgplot, indgen(10),color=cgcolor("Dodger Blue"), /window
>  IDL> cgplot, indgen(10)+1, color=cgcolor("Hot Pink"), /add, /over
>
> 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.")

I get the same result (greyness) with /ADD.

- thanks,
KW

---

Subject: Re: cgcolor, cgplot, and /WINDOW
Posted by willettk on Wed, 23 Mar 2011 04:13:26 GMT
View Forum Message <> Reply to Message

On Mar 22, 10:09 pm, Kwill <wille...@gmail.com> wrote:

> On Mar 22, 10:01 pm, David Fanning <n...@idlcoyote.com> wrote:
>
>
>
>
>
>
>
>
>
>
>> Kwill writes:
>>> I was excited to learn today of David's Coyote Graphics packages, and
>>> downloaded them on both my laptop and desktop. I'm getting very
>>> puzzling results on one machine (but not the other). When I set the /
>>> WINDOW keyword in cgplot, cgcolor will not work for overplotting in
>>> color, displaying all subsequent points in a dark grey.
>
>>> My example:
>
>>> IDL> cgplot, indgen(10),color=cgcolor("Dodger Blue"), /window
>>> IDL> cgplot, indgen(10)+1, color=cgcolor("Hot Pink"), /window, /over
>
>>> This results in one blue and one grey line plotted on the screen. I do
>>> not have any color problems when the /WINDOW keyword is not set.
>
>>> I tried an identical series of commands on my laptop: it runs
>>> perfectly, producing both a blue and pink line. Both of my machines
>>> run IDL Version 7.1.1, Mac OS X (darwin x86_64 m64) on Snow Leopard
>>> 10.6.4.
>
>>> I'm baffled as to why this would be different between my two machines,
>>> but hopeful that I'm doing something wrong and that CGCOLOR plays
>>> nicely with the resizeable window in CGPLOT. Any ideas?
>
>> What happens if you use ADD instead of WINDOW on the second
>> command?
>
>>  IDL> cgplot, indgen(10),color=cgcolor("Dodger Blue"), /window
>>  IDL> cgplot, indgen(10)+1, color=cgcolor("Hot Pink"), /add, /over
>
>> 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.")
>
> I get the same result (greyness) with /ADD.
>
> - thanks,
> KW

Whoops - but in David's second post, he correctly identified the problem. It works fine when I don't call cgcolor twice - although I'm still puzzled why it worked on one machine and not the other. Problem solved for me, though.

Thanks!

---

Subject: Re: cgcolor, cgplot, and /WINDOW
Posted by David Fanning on Wed, 23 Mar 2011 05:02:15 GMT
View Forum Message <> Reply to Message

Kwill writes:

> I was excited to learn today of David's Coyote Graphics packages, and
> downloaded them on both my laptop and desktop. I'm getting very
> puzzling results on one machine (but not the other). When I set the /
> WINDOW keyword in cgplot, cgcolor will not work for overplotting in
> color, displaying all subsequent points in a dark grey.
>
> My example:
>
> IDL> cgplot, indgen(10),color=cgcolor("Dodger Blue"), /window
> IDL> cgplot, indgen(10)+1, color=cgcolor("Hot Pink"), /window, /over
>
> This results in one blue and one grey line plotted on the screen. I do
> not have any color problems when the /WINDOW keyword is not set.
>
> I tried an identical series of commands on my laptop: it runs
> perfectly, producing both a blue and pink line. Both of my machines
> run IDL Version 7.1.1, Mac OS X (darwin x86_64 m64) on Snow Leopard
> 10.6.4.
>
> I'm baffled as to why this would be different between my two machines,
> but hopeful that I'm doing something wrong and that CGCOLOR plays
> nicely with the resizeable window in CGPLOT. Any ideas?

OK, this explanation is complicated. :-)

First of all. This is the wrong way to pass a color to cgPlot. The color does NOT have to go through cgColor,

since it will go through cgColor internally in cgPlot.
By calling the command the way Kwill has here, a string
is turned into a number, then the number is passed to
cgPlot.

In itself, this is not a problem. In fact, had Kwill just
issued these two commands and had them display in a normal
IDL graphics window, there would have been no problem. He
would have gotten the result he expected. And, of course,
as he reports, he has no problem if he uses the software
correctly.

He would also have gotten the result he expected in cgWindow
had he been in decomposed color mode. I suspect the reason
one machine worked and the other didn't is that one machine
was in decomposed color mode and the other wasn't. One of
my goals (probably not possible in my life time) is to convince
everyone to run IDL in decomposed color mode. In fact, I would
strongly support IDL eliminating indexed color mode entirely.
It would make me a rich man. :-)

BUT, he was in indexed color mode and he wanted to display
the commands in cgWindow. So, as I say, things are complicated. :-)

In indexed color mode, cgColor loads a color in the color table
and returns an index to the loaded color. So it is this number,
say, 138 that is passed into cgPlot. cgPlot does ALL its
graphics in decomposed color, so it has to convert that index
number to a 24-bit value. As you can see, it did this
correctly with the first command, and incorrectly with the
second command. How come? This is the mystery.

It turns out that to make cgWindow work properly in an
unknown IDL color environment, I have to exercise pretty
tight control over the color space. Basically, what I do
is save the current color table when I enter a program, and
restore it after I leave. What I do between entering and
leaving is no business of yours.

This works pretty well, until there are two commands to
execute in a resizeable graphics window. What happens
when the program was called the way Kwill called it was
that between the first and second commands, the color
table was "restored" to the way it was when the program
was first called. This had the effect of "erasing" the
hot pink color, so that the color at 138 was a gray color,
not hot pink. In other words, the sequencing of the loading
and restoring the color tables went awry.

So, the take home message is this. (1) Call the programs
they way they were meant to be called, or (2) Swear off
indexed color forever. I can assure you if you choose (2)
life will be very, very pleasant for you. You may even
come to understand how color works in IDL. :-)

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.")

---

Subject: Re: cgcolor, cgplot, and /WINDOW
Posted by Foldy Lajos on Wed, 23 Mar 2011 11:12:13 GMT
View Forum Message <> Reply to Message

Hi David,

On Tue, 22 Mar 2011, David Fanning wrote:

> So, the take home message is this. (1) Call the programs
> they way they were meant to be called, or (2) Swear off
> indexed color forever. I can assure you if you choose (2)
> life will be very, very pleasant for you. You may even
> come to understand how color works in IDL. :-)

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.

---

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.")

Subject: Re: cgcolor, cgplot, and /WINDOW
Posted by Michael Galloy on Wed, 23 Mar 2011 23:49:30 GMT
View Forum Message <> Reply to Message

On 3/23/11 6:32 AM, David Fanning wrote:
> 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.

You can pass either a single index or a 3-element array for a color in object graphics. If you pass a single index, then the object uses its PALETTE property to lookup the actual color. Of course, now in IDL 8.0, you can say !color.red.

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

---

Subject: Re: cgcolor, cgplot, and /WINDOW
Posted by David Fanning on Thu, 24 Mar 2011 03:22:31 GMT
View Forum Message <> Reply to Message

Michael Galloy writes:

> You can pass either a single index or a 3-element array for a color in
> object graphics. If you pass a single index, then the object uses its
> PALETTE property to lookup the actual color. Of course, now in IDL 8.0,
> you can say !color.red.

Yes, this is what Coyote Graphics does, too, if you pass a color palette. And red, of course, is just passed as "red". ;-)

Cheers,

David

--
David Fanning, Ph.D.