
Subject: Re: How to load a customized color table?

Posted by [David Fanning](#) on Sat, 10 Jan 2009 17:57:48 GMT

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

ICBM0926 writes:

```
> I've tried to load a customized color table with "TVLCT" in a
> procedure. I called this procedure from another one, but the plot
> generated uses B-W color table instead of a red-blue one.
> I can't get this thing work. Can anyone tell me where go wrong?
>
> The code is attached below.
> pro ctloader
> ;loadct,25
> ;COMMON COLORS, R_Orig, G_Orig, B_Orig, R_curr, G_curr, B_curr
> COMMON COLORS, r, g, b, R_curr, G_curr, B_curr
> print,"This is ctloader."
> r=dblarr(256)
> r[0:127]=reverse(2.0*findgen(128))
> r[128:255]=0.0
> R_curr=r
> b=dblarr(256)
> b[0:127]=0.0
> b[128:255]=2.0*findgen(128)
> B_curr=b
> g=dblarr(256)
> g=0.0
> G_curr=g
> tvlct,r,g,b
> end
```

When you set g=0, you are effectively loading only a single color. The only color your load is at index 0.

You can see this if you look at your color table with CINDEX:

<http://www.dfanning.com/programs/cindex.pro>

If you take that line out, you get a red-blue color table. Is that what you were going for?

Cheers,

David

--

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

Subject: Re: How to load a customized color table?
Posted by [russell.grew](#) on Mon, 12 Jan 2009 09:26:52 GMT
[View Forum Message](#) <> [Reply to Message](#)

g[*] = 0.

may be more appropriate.

Subject: Re: How to load a customized color table?
Posted by [Spon](#) on Mon, 12 Jan 2009 18:43:41 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Jan 12, 9:26 am, RussellGrew <russell.g...@gmail.com> wrote:

> g[*] = 0.

>

> may be more appropriate.

Also completely unnecessary, as IDL will set all elements of a newly created array to 0 by default unless you use the /NoZero keyword. I'm also not sure what that common block is doing there, or the double precision numbers that get turned into bytes anyway.

This is how I'd write such a procedure:

```
; OldRGB is an output variable that
; returns the previous colour table values
PRO LOADREDBLUE, OldRGB
; Store current colour table
TVLCT, OldRGB, /GET
; Generate new colour table values
R = [2B*REVERSE(BINDGEN(128)), BYTARR(128)]
G = BYTARR(256)
B = [BYTARR(128), 2B*BINDGEN(128)]
; And load them
TVLCT, R, G, B
PRINT, 'Loading table: Red-Black-Blue'
RETURN
END
```

So if you cared about turning your old colour table back on after your call, you could use the procedure like in this example:

```
IDL> Device, Decomposed = 0
```

```
IDL> Image = Dist(256)
```

```
IDL> LOADREDBLUE, Previous ; load your custom table. The 'Previous' variable contains the old table's values.
```

```
IDL> TVSCL, Image ; display the image
```

```
IDL> TVLCT, Previous ; reload the old table
```

```
IDL> Window, /Free & Plot, Image ; using previous colour table in a new window
```

And if you don't care what the previous colour table looked like, just type:

```
IDL> LOADREDBLUE
```

and you're away (assuming that, again, you've turned off decomposed colours)

```
IDL> TV, Bindgen(256,256)
```

Regards,
Chris

Subject: Re: How to load a customized color table?

Posted by [David Fanning](#) on Mon, 12 Jan 2009 18:51:23 GMT

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

Spon writes:

> On Jan 12, 9:26=A0am, RussellGrew <russell.g...@gmail.com> wrote:

>> g[*] =3D 0.

>>

>> may be more appropriate.

>

> Also completely unnecessary, as IDL will set all elements of a newly
> created array to 0 by default unless you use the /NoZero keyword. I'm
> also not sure what that common block is doing there, or the double
> precision numbers that get turned into bytes anyway.

>

> This is how I'd write such a procedure:

>

> ***

>

> ; OldRGB is an output variable that
> ; returns the previous colour table values
> PRO LOADREDBLUE, OldRGB
> ; Store current colour table
> TVLCT, OldRGB, /GET

```

> ; Generate new colour table values
> R =3D [2B*REVERSE(BINDGEN(128)), BYTARR(128)]
> G =3D BYTARR(256)
> B =3D [BYTARR(128), 2B*BINDGEN(128)]
> ; And load them
> TVLCT, R, G, B
> PRINT, 'Loading table: Red-Black-Blue'
> RETURN
> END
>
> ***
>
> So if you cared about turning your old colour table back on after your
> call, you could use the procedure like in this example:
> IDL> Device, Decomposed =3D 0
> IDL> Image =3D Dist(256)
> IDL> LOADREDBLUE, Previous ; load your custom table. The 'Previous'
> variable contains the old table's values.
> IDL> TVSCL, Image ; display the image
> IDL> TVLCT, Previous ; reload the old table
> IDL> Window, /Free & Plot, Image ; using previous colour table in a
> new window
>
> And if you don't care what the previous colour table looked like, just
> type:
> IDL> LOADREDBLUE
> and you're away (assuming that, again, you've turned off decomposed
> colours)
> IDL> TV, Bindgen(256,256)

```

Yeah, I'd do it like this:

```
IDL> CTLoad, 22, /Brewer
```

Of course, that goes through white rather than black, but that's better for the...uh, old folks who might be looking at your plot. :-)

Cheers,

David

--

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: How to load a customized color table?
Posted by [Spon](#) on Tue, 13 Jan 2009 11:07:43 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Jan 12, 6:51 pm, David Fanning <n...@dfanning.com> wrote:

> Yeah, I'd do it like this:

>

> IDL> CTLoad, 22, /Brewer

>

> David

Heh, well, I was (perhaps foolishly) assuming the black had been chosen for a reason ;-)

Chris
