
Subject: Re: 8-bit vs. 24-bit color on Windows

Posted by [Phillip & Suzanne](#) on Tue, 09 Feb 1999 08:00:00 GMT

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

Martin Schultz wrote:

>

> Stein Vidar Hagfors Haugan wrote:

>>

>> To them (and me), the "I" in IDL stands for exactly what he
>> described (interactive use of commands and xloadct in unison),
>> and no widget programmer can ever imagine in advance *everything* a
>> scientist would want to do with his data before/during/after
>> displaying them (the scientist doesn't know in advance, either:-),
>> so the interactive command line is an extremely valuable part of IDL.

>>

>> Not having the possibility of exploring an image by tweaking
>> the color table in pseudo-color mode would take away a lot of the
>> original appeal of IDL, IMHO, so this should be taken very seriously
>> by RSI.

>>

>

> So true, so true !

I work in a shop of about 80 people, roughly half programmers and half analysts. The analysts almost all use IDL for a lot of their work, but get by with programmer support from only 4 of us! While the four of us really appreciate all of the wonderful niceties provided by each upgrade, most of our users get more annoyed than anything else.

In particular, direct graphics into a window is the typical M.O. of most of these people. When they figure out what they want, THEN they consider creating a basic widget to automate the process in the future IF they're pretty saavy with IDL.

> Let me unravel a little dream here that I had again lately: Wouldn't it
> be great if a program would have enough intelligence to notice
> inconsistencies in user input? Example: you draw a map and you specify
> both hatched and filled continents. Instead of somehow selecting either
> one, a window would pop up and present you with the possible options.
> Likewise, all parameters and keywords from a program could be displayed
> (and queried) automatically either by setting a special keyword (e.g.
> draw,/query) or as some kind of error handler (e.g. if (n_params() eq 0)
> then query,...). This could be helpful to run programs that you are not
> too familiar with (now, what were these keywords again ??), and it would
> take care of those countless situations when you run a program in
> different contexts or with different data sets where for instance one
> variable is missing that was explicitly defined in the other data set
> (or it is named differently). The mechanism would still be "invisible"

- > if you specify all parameters and keywords correctly, and it would of
- > course have to "know" the keywords and parameters automatically so that
- > you don't have to pass a long name list to query when you call it. The
- > "interactive variable assigner" should allow to either enter values
- > manually, or select from all variables that are "visible" on the caller
- > level (and maybe even globally)...

While I'm not volunteering to do so, it should be simple to write just such a program. Pass the command line for the command you want to a routine that has a case statement to check each of the possible commands, determines if the calling sequence is reasonable, and either invokes the requested command or pops up a Dialog_Message box indicating what options are available. Since most users really only use a small subset of the commands, you wouldn't have to make it complete (unless you're RSI doing this...:-). Instead, only include the commands you have trouble with. Good luck.

Phillip
