Subject: Re: All I get is gray scale (and IDL 5.5)
Posted by Ruediger Kupper on Fri, 01 Feb 2002 13:15:41 GMT
View Forum Message <> Reply to Message

David Fanning wrote:

> Yes, something is not right. I'd slip a
>
> DEVICE, TRUE_COLOR=24
>
> into an IDL start-up file. I don't think you
> want a Direct Color visual. :-(

Hi Gary and David,

now, perhaps there is a possibility for me to understand
this issue, since I tried several times and never managed.
I encountered the same problem as Gary, and I
(heuristically) found the same solution for it, i.e.,
keeping IDL from allocating a Direct-Color visual by
explicitely requesting a True-Color one.
Then, upgrading to IDL5.5, all my colors were scrambled
again. After a series of tests, I discovered that now I need
an additional DEVICE, BYPASS_TRANSLATION=0 call to turn on
IDL's internal color translation table, AFTER the first
window has been opened. (The translation table is enabled
right after requesting the True-Color visual, but it
switches to bypassed when the first window opens. Although
this was atready the case with IDL 5.4, the TV command
obviously didn't bother - It does with IDL 5.5.)

Okay. Now, if anyone of you can help me with the following
questions, I would greatly appreciate:

First question: If I "don't want a Direct-Color visual", why
does IDL want it? As this is the order in which IDL tries to
allocate: Direct-Color first.

Second question: I did not find any way how to use the
Direct-Color visual at all, but perhaps I still don't
understand it right - What exact difference is there between
Direct-Color and True-Color, and how would IDL access the
two different modes? Reading the IDL documentation I get the
impresseion that everything should work fine - why doesn't it?

Third question (related to this): How and when is IDL's internal translation table used on a 24-bit display? The documentation is not very helpful, as it refers to 8-bit displays (in which case I do understand what it means). Actually, the documentation says:
(1) "By default, the translation tables are used with shared and static color tables. When using displays with private color tables, the translation tables are bypassed." While also stating
(2) "When a private or static color table is initialized, the bypass flag is cleared. It is set when initializing a shared color table."
Now this gives us something to think about :-).

Fourth question: Even if we don't understand why it does, what it does - is there any way to determine a configuration that works, without having to try them all out? We use IDL in our group, with a central startup file, and on many different machines (Windows NT, Intel Linux, DEC Alpha Unix, DEC Alpha Linux, 8-bit and 24-bit graphics cards), and, for licensing reasons, IDL versions ranging from 5.5 back to 5.2. Do I really have to ask every single user to try, until he finds a configuration that works on his system? I would expect a high-end visualisation tool like IDL to work without every single user needing to know the internals of his graphics hardware.

Fith question: Does RSI give any statement on this (sorry to say so) absolutely weird color management on X systems?

Please excuse me sounding a bit frustrated (or "stupid", as Gary put it), but I actually feel like it :-(.

I'm using DECOMPOSED=0, by the way, but this should not really matter here.

Thank you in advance for any helpful comments,
Regards,
Rüdiger.