Subject: Re: Legend object without border
Posted by Paul Van Delst[1] on Wed, 23 Sep 2015 14:49:12 GMT
View Forum Message <> Reply to Message

Hello again,

On 09/22/15 22:55, laura.hike@gmail.com wrote:
> Both of the suggested techniques work in the sense that you don't see
> a border when they are used, however, they don't accomplish my goal.
> The problem I have is that the legend overlaps some of my plotted
> lines, but wouldn't if there wasn't an actual box (with or without a
> border). When I use either of these techniques, the white corner of
> the legend box still overlaps the lines, causing a gap. I tried
> overplotting the lines again, but the box will not be overwritten, so
> I'm still at a loss. Is there no way to make the legend without
> creating a box at all?

Ohhhh.... I see.

Well, then, as alx stated, the TRANSPARENCY keyword (set to 100% transparency) is what you want.

(I also tested the legend ORDER method with /SEND_TO_BACK but that didn't do what you wanted.)

> I think a large portion of my frustration with IDL is that the
> documentation is poor. When I try something new, I look at the
> appropriate Exelis web page, but so many details aren't covered, like
> why there is a legend outline in one exaple but not the other. And
> yes, things often aren't intuitive. For example, on the page
> describing "legend," it does say that "linestyle" is a valid
> property, but I assumed that this would modify the lines in the
> legend, not the _outline_ of the legend.

I think your question highlighted one of the difficulties of writing documentation. Stating things clearly is quite hard. I certainly misunderstood.

With regards to the IDL docs, I think they suffer from not enough cross-referencing and too few examples. Especially for the more esoteric uses of FG.

I find myself visiting David Fannings Graphics Gallery (http://www.idlcoyote.com/gallery) quite often to see if what I want to do can actually be done (in DG or FG). That gives me the fortitude to soldier on (and, in some cases, the actual FG code where he has dual DG and FG examples!).

> I switched to object graphics because I liked the way I could
> manipulate them until I get what I want and because there are more
> built-in functions than for the old graphic routines, but a lot of
> things are turning out to be difficult. (I used a user-written
> routine called "legend" previously -- I think it originated at MIT?
> -- and it seemed more intuitive and included many options.)

No argument here. It's a learning process. An archival search of this newsgroup will reveal a lot of grousing from me about object graphics when they first arrived (pre-Function Graphics). I suspect RSI (back in the day) softened us users up with iTools and Insight and LIVE Tools (cripes - remember Insight? And LIVE? Argh!). After all that, Object and Function graphics was the proverbial manna from on high. (ha ha)

But, similar to alx, I have found the capabilities of FG over DG *vastly* outweigh the frustrations[*]. In my case I had to learn to stop thinking the DG-way and shift to thinking FG. Of course, it would've been nice if that process was less orthogonal. :o\

cheers,

paulv

[*] At least until I have to start plotting millions of data points again...then, unless OG/FG speed has increased several orders of magnitude in the last few releases, I'll probably have to use DG again. (insert sad trombone music here)