Subject: Re: FSC_PLOT defaults
Posted by David Fanning on Mon, 20 Dec 2010 22:57:59 GMT
View Forum Message <> Reply to Message

Wayne Landsman writes:

> I am updating the plotting routines in the IDL Astronomy library (e.g.
> AL_LEGEND, TVCIRCLE, PLOTERROR in http://idlastro.gsfc.nasa.gov/) to
> use the new Coyote routines FSC_PLOT, and FSC_COLOR(), and for example
> to allow colors to be specified by name. It is mostly going well
> but I do find it jarring going from the white on black of
>
> IDL> plot,indgen(10)
>
> to the black on white of
>
> IDL> fsc_plot,indgen(10)
>
> I'm old-fashioned enough to think that black on white does not show
> up as clearly on the screen, but I did not have a similar problem
> with the black on white of function graphics.
>
> The reason is that
>
> IDL> o = plot(indgen(10))
>
> is more comparable to something like
>
> IDL> fsc_plot,indgen(10),thick=2,xthick=2,ythick=2,charsize = 2, xticklen = !P.ticklen*2,
yticklen=!p.ticklen*2
>
> to give the lines extra thickness (and clarity).
>
> Maybe it would be worth having a similar default display for
> FSC_PLOT (at the expense of having different defaults than PLOT).
> Another default I like in the function PLOT() is having exact X,Y
> ranges (X|Y|Style= 1). --Wayne

The central theme of my upcoming book is that there is
no real need for fancy new graphics. That almost everything
you need is already available to you via traditional graphics
commands IF you do a few simple things. One of those things
is write programs that work both on your display and in a
PostScript file. Of course, it is *possible* to do this
if one has a black background and one has a white background,
but if you use color at all (another theme of the book), then
this is all much harder to do.

Better, I thought, to make everything work the same.
Even old timers with schizophrenic brains catch onto
it fairly quickly and realize how easy it is to make things
work identically in the two different environments. :-)

That said, I've gone to some trouble to make sure these
routines work "naturally". If you set the background color
to "black", the drawing routines should draw in white.
I've just really changed the default background color.

I've no objection to changing the default line thickness
values, although I have tried to make these commands work
as normally as possible, even when "normally" sometimes
means sub-optimally. I really only tried to fix the things
I thought were obvious defects. (I do set exact axis ranges
on the FSC_Contour command, but personally, I don't mind
autoscaling axes on the Plot command most of the time.)

I do note that I *always* change line thicknesses when
I make a PostScript file, so maybe changing it everywhere
is not such a bad idea. Let me play with this a little and
see what I can do. Anything that makes these more useful
is what I am looking for.

We still have a month or so to experiment before
they are cast in stone. :-)

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: FSC_PLOT defaults
Posted by ben.bighair on Tue, 21 Dec 2010 01:43:01 GMT
View Forum Message <> Reply to Message

On 12/20/10 5:57 PM, David Fanning wrote:
> Wayne Landsman writes:
>
>> I am updating the plotting routines in the IDL Astronomy library (e.g.
>> AL_LEGEND, TVCIRCLE, PLOTERROR in http://idlastro.gsfc.nasa.gov/) to
>> use the new Coyote routines FSC_PLOT, and FSC_COLOR(), and for example

>> to allow colors to be specified by name. It is mostly going well
>> but I do find it jarring going from the white on black of
>>
>> IDL> plot,indgen(10)
>>
>> to the black on white of
>>
>> IDL> fsc_plot,indgen(10)
>>
>> I'm old-fashioned enough to think that black on white does not show
>> up as clearly on the screen, but I did not have a similar problem
>> with the black on white of function graphics.
>>
>> The reason is that
>>
>> IDL> o = plot(indgen(10))
>>
>> is more comparable to something like
>>
>> IDL> fsc_plot,indgen(10),thick=2,xthick=2,ythick=2,charsize = 2, xticklen = !P.ticklen*2,
yticklen=!p.ticklen*2
>>
>> to give the lines extra thickness (and clarity).
>>
>> Maybe it would be worth having a similar default display for
>> FSC_PLOT (at the expense of having different defaults than PLOT).
>> Another default I like in the function PLOT() is having exact X,Y
>> ranges (X|Y|Style= 1). --Wayne
>
> The central theme of my upcoming book is that there is
> no real need for fancy new graphics. That almost everything
> you need is already available to you via traditional graphics
> commands IF you do a few simple things. One of those things
> is write programs that work both on your display and in a
> PostScript file. Of course, it is *possible* to do this
> if one has a black background and one has a white background,
> but if you use color at all (another theme of the book), then
> this is all much harder to do.
>
> Better, I thought, to make everything work the same.
> Even old timers with schizophrenic brains catch onto
> it fairly quickly and realize how easy it is to make things
> work identically in the two different environments. :-)
>
> That said, I've gone to some trouble to make sure these
> routines work "naturally". If you set the background color
> to "black", the drawing routines should draw in white.
> I've just really changed the default background color.

>
> I've no objection to changing the default line thickness
> values, although I have tried to make these commands work
> as normally as possible, even when "normally" sometimes
> means sub-optimally. I really only tried to fix the things
> I thought were obvious defects. (I do set exact axis ranges
> on the FSC_Contour command, but personally, I don't mind
> autoscaling axes on the Plot command most of the time.)
>
> I do note that I *always* change line thicknesses when
> I make a PostScript file, so maybe changing it everywhere
> is not such a bad idea. Let me play with this a little and
> see what I can do. Anything that makes these more useful
> is what I am looking for.
>
> We still have a month or so to experiment before
> they are cast in stone. :-)

Hi,

I wonder if you might consider having a per-user default preference stored in a system variable - it's really just swiping the idea from IDL's !P, !D system variables but with improved functionality. The preferences could be stored in a FSC_Defaults.pref save file that is loaded at least once per session (somehow). I think I would make the initial setup similar to IDL's PREF_SET, PREF_GET, etc. That would allow the user to decide what they really want as defaults if different than what you provide. There could even be a default per device.

So, instead of this...

```
; Check the keywords.
    IF N_Elements(sbackground) EQ 0 THEN $
 background = 'WHITE' ELSE $
 background = sbackground
```

It might be this...

```
; Check the keywords.
    IF N_Elements(sbackground) EQ 0 THEN $
 background = FSC_Default("background") ELSE $
 background = sbackground
```

And the function FSC_Default could be simple a wrapper around !FSC_Default object that maintains the settings.

```
Function FSC_Default, what, value, SET = set, _REF_EXTRA = extra

  IF (KEYWORD_SET(set)) THEN $
    return, !FSC_Default->Set(what, value, _EXTRA = extra) ELSE $
    return, !FSC_Default->Get(what, _EXTRA = extra)

END;
```

The object that !FSC_DEFAULT references would have two methods, Get and Set, that redirect to "Get_PS", "Get_Win", "Set_PS", "Set_X", etc. depending upon the current device state. Like this...

```
Function FSC_DEFAULT::Set, what, value, SAVE = save

 theMethod = "Set_" + !D.Name
 oldValue  = self->CallMethod(theMethod, what, value)
 if (save) then self->Save
 return, oldValue

END

Function FSC_DEFAULT::Get, what, COUNT = count

 theMethod = "Get_" + !D.Name
 return(self->CallMethod(theMethod, what, COUNT = count))

END
```

This might be a whacky idea and not worth the effort. But it is easy to suggest since I wouldn't be doing the work!

On one hand it would put the default settings in the user's hands on the other it would take a good deal of mind-numbing effort to switch all of the keyword tests. The consolation is it only has to be done once - forever (maybe).

Cheers,
Ben

Subject: Re: FSC_PLOT defaults
Posted by penteado on Tue, 21 Dec 2010 02:03:41 GMT
View Forum Message <> Reply to Message

On Dec 20, 11:43 pm, Ben Tupper <ben.bigh...@gmail.com> wrote:

> I wonder if you might consider having a per-user default preference
> stored in a system variable - it's really just swiping the idea from
> IDL's !P, !D system variables but with improved functionality.

One of the main reasons I do not use object graphics is their reliance on global variables - especially those that do not get reset with a .reset_session. I usually object to global variables on any situation, as they make the results dependent on the environment, not just the code the user writes / receives (and, particularly, dependent not just on the routine calls). Also, it is awkward to use simultaneous instances of things that rely on (the same) global variables.

Subject: Re: FSC_PLOT defaults
Posted by Marc Buie on Tue, 21 Dec 2010 13:58:28 GMT
View Forum Message <> Reply to Message

Just one warning on x/y/z.style. I have always used style=3 since I don't like the ticks getting involved with the data. I know that others have their own preferences and that's fine. But, the _biggest_ objection I have to the new plot() function is that the style=3 functionality is not supported. You get style=1 whether you want it or not. That one deficiency alone is enough for me to turn my back on an otherwise promising new method of plotting.

However, I do agree with David. Changing the default background to white for all devices is something that I hated at first but quickly grew to like. If you use color you have to use completely different color values when switching from a black to white background and I really do value saving a plot that can be used anywhere (without having to plan ahead). As a global "standard", I find a white background more universally useful.

Subject: Re: FSC_PLOT defaults
Posted by David Fanning on Tue, 21 Dec 2010 14:45:13 GMT
View Forum Message <> Reply to Message

Marc Buie writes:

> Just one warning on x/y/z.style. I have always used style=3
> since I don't like the ticks getting involved with the data.
> I know that others have their own preferences and that's fine.
> But, the _biggest_ objection I have to the new plot() function
> is that the style=3 functionality is not supported. You get
> style=1 whether you want it or not. That one deficiency alone is
> enough for me to turn my back on an otherwise promising new
> method of plotting.

Wow! 25 years working with IDL and I didn't even *know*

there was a [XYZ]Style=3. The things you learn... :-)

Neat!

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: FSC_PLOT defaults
Posted by David Fanning on Tue, 21 Dec 2010 14:51:43 GMT
View Forum Message <> Reply to Message

Paulo Penteado writes:

> One of the main reasons I do not use object graphics is their reliance
> on global variables - especially those that do not get reset with
> a .reset_session. I usually object to global variables on any
> situation, as they make the results dependent on the environment, not
> just the code the user writes / receives (and, particularly, dependent
> not just on the routine calls). Also, it is awkward to use
> simultaneous instances of things that rely on (the same) global
> variables.

I "sorta" agree with this. I don't like to rely on
global variables either, with one very pronounced
exception. I do set up my PostScript output (via
PS_START) with global variables for plot thickness,
character size, font selection, etc. It is just so
much easier to do it this way.

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: FSC_PLOT defaults
Posted by David Fanning on Tue, 21 Dec 2010 15:06:39 GMT
View Forum Message <> Reply to Message

Ben Tupper writes:

> I wonder if you might consider having a per-user default preference
> stored in a system variable - it's really just swiping the idea from
> IDL's !P, !D system variables but with improved functionality. The
> preferences could be stored in a FSC_Defaults.pref save file that is
> loaded at least once per session (somehow). I think I would make the
> initial setup similar to IDL's PREF_SET, PREF_GET, etc. That would
> allow the user to decide what they really want as defaults if different
> than what you provide. There could even be a default per device.

There are some good suggestions here, Ben, but I think
I am content to leave these traditional graphics commands
alone to act, well, traditionally. I'll save these ideas
for my new simple, *function* graphics system that I'll
be introducing with the book I'm writing next. It's working
title is "Function Graphics Even a Dullard Like Coyote Can
Understand!." :-)

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: FSC_PLOT defaults
Posted by David Fanning on Tue, 21 Dec 2010 15:19:00 GMT
View Forum Message <> Reply to Message

Wayne Landsman writes:

> I'm old-fashioned enough to think that black on white does
> not show up as clearly on the screen, but I did not have a
> similar problem with the black on white of function graphics.
>
> The reason is that
>
> IDL> o = plot(indgen(10))
>

> is more comparable to something like
>
> IDL> fsc_plot,indgen(10),thick=2,xthick=2,ythick=2,charsize = 2, $
> xticklen = !P.ticklen*2, yticklen=!p.ticklen*2
>
> to give the lines extra thickness (and clarity).
>
> Maybe it would be worth having a similar default display for
> FSC_PLOT (at the expense of having different defaults than PLOT).

OK, I've implemented this for FSC_Plot, but I'm not sure about
it. It's, uh, pretty bold. :-(

Maybe it grows on you. :-)

Anyway, I thought I would put it out there for people to play
with and test. If we decide we like it, we can make the change
permanent. You can find the test program here:

  http://www.dfanning.com/misc/fsc_plot_test.pro

You would run it like this:

  IDL> FSC_Plot_Test, findgen(11)

I have different defaults for the display and for PostScript. I try
to use the ratio of display*1.7 for PostScript defaults. I think this
works well, but appreciate your input with these defaults, too.

My biggest problem is preserving the ability to use system variables
to set plot properties. This is especially a problem with things like
tick length. These can be set with !P.Ticklen as well as !X.Ticklen
and !Y.Ticklen. So, if I am going to set a default tick length, I have
to make sure none of those other variables are set, etc.

Anyway, the more I look at this plot this morning, the more I like it.
So maybe this is the way to go. Let me know what you think! :-)

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: FSC_PLOT defaults
Posted by Paul Van Delst[1] on Tue, 21 Dec 2010 15:39:56 GMT
View Forum Message <> Reply to Message

David Fanning wrote:
> The central theme of my upcoming book is that there is
> no real need for fancy new graphics.

Wha..? Seriously?

I will agree that you can plot everything you need to plot with either direct or function (aka "new") graphics, but the
ability to easily manipulate the function graphic result, outside of its creation routine, after it's been created is a
huge step forward. And this step is built upon existing tools (object graphics), involving more packaging and interface
updates than actual new functionality. I find function graphics way more useful than iTools. Go figure.

I've never been a booster of the ITTVIS development process (we pay lotsa $$ for licenses and that's all the ego
massaging I'm prepared to dole out), but with function graphics they have, for the first time since v6 (I'm a command
line guy who's never been into IDEs so v7 was mostly lost on me), provided an update that earned the cost of the product.

My measure of success is if using new stuff in IDL makes it easier to for me to visualise my data and to communicate its
content to others via a plot/surface/whatever.

In that regard, I reckon function graphics beats direct hands down. The effort and time to learn how to use them has
paid off (and my knowledge of them is still rudimentary at best).

Since I got v8 installed, I have not written any new code that uses direct graphics. It's all been function graphics.
And I've been happy to do so despite my reservations about picture quality when embedding generated graphics in
documents -- word, powerpoint, latex, whatever. (others have posted in c.l.i-p about how to solve that problem too).

There are still some things that need to be addressed[*] but function graphics, for the most part, work the way I expect
them to. E.g. I *want* my legends to be associated with the data ... not something added afterwards.

Disclaimer: Various regulations I'm subject to require me to state that the above is an expression of my personal
opinion only and does not represent any official position or policy of my employers (any of 'em)

cheers,

paulv

[*] For a start:
1) better documentation/examples on how to do stuff
2) See #1
3) speed issues when plotting lost and lots of data (also see #1)
4) the ability to copy+paste objects in a graphic (e.g. added annotations/arrows etc)
5) better (i.e. not so 80's looking) properties interface... although that may be a motif/X windows thing on linux.

> That almost everything
> you need is already available to you via traditional graphics
> commands IF you do a few simple things. One of those things
> is write programs that work both on your display and in a
> PostScript file. Of course, it is *possible* to do this
> if one has a black background and one has a white background,
> but if you use color at all (another theme of the book), then
> this is all much harder to do.
>
> Better, I thought, to make everything work the same.
> Even old timers with schizophrenic brains catch onto
> it fairly quickly and realize how easy it is to make things
> work identically in the two different environments. :-)
>
> That said, I've gone to some trouble to make sure these
> routines work "naturally". If you set the background color
> to "black", the drawing routines should draw in white.
> I've just really changed the default background color.
>
> I've no objection to changing the default line thickness
> values, although I have tried to make these commands work
> as normally as possible, even when "normally" sometimes
> means sub-optimally. I really only tried to fix the things
> I thought were obvious defects. (I do set exact axis ranges
> on the FSC_Contour command, but personally, I don't mind
> autoscaling axes on the Plot command most of the time.)
>
> I do note that I *always* change line thicknesses when
> I make a PostScript file, so maybe changing it everywhere
> is not such a bad idea. Let me play with this a little and
> see what I can do. Anything that makes these more useful
> is what I am looking for.
>
> We still have a month or so to experiment before
> they are cast in stone. :-)

>
> Cheers,
>
> David
>

Subject: Re: FSC_PLOT defaults
Posted by David Fanning on Tue, 21 Dec 2010 15:58:12 GMT
View Forum Message <> Reply to Message

Paul van Delst writes:

> Wha..? Seriously?
>
> I will agree that you can plot everything you need to plot with either direct or function (aka "new") graphics, but the
> ability to easily manipulate the function graphic result, outside of its creation routine, after it's been created is a
> huge step forward. And this step is built upon existing tools (object graphics), involving more packaging and interface
> updates than actual new functionality. I find function graphics way more useful than iTools. Go figure.

Paul, take it easy!

I'd probably be a fan of function graphics, too, if I could get IDL 8 to work on my computer. But I'm stuck in IDL 7 for the time being, waiting for the problems to be sorted out. If I judge the people sending me e-mail and browsing my web page correctly, I'm not the only one wishing for some easy alternative.

These are simple, they are fast, they produce the PostScript output you expect. You can actually write programs with them (as opposed, say, to just display data). There are just a lot of things to like about them. But, you don't have to use them if you don't want to. :-)

Anyway, this is just a stop-gap until I can get IDL 8 running. Then I have some ideas for function graphics commands that work the way I want them to. Maybe you will like those better. ;-)

But, yes, seriously! I don't feel any great need for anything fancier than direct graphics. If you can make a PostScript file and you can install ImageMagick,

most of the things I have complained about for years
are solved. :-)

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: FSC_PLOT defaults
Posted by wlandsman on Tue, 21 Dec 2010 16:19:54 GMT
View Forum Message <> Reply to Message

On Tuesday, December 21, 2010 10:19:00 AM UTC-5, news wrote:

> OK, I've implemented this for FSC_Plot, but I'm not sure about
> it. It's, uh, pretty bold. :-(

Yeah, the problem is that at least on my display the THICK keywords don't work in fractional amounts. XTHICK = 1.5 (which I would prefer) is the same as XTHICK=2.

I was trying to get the output to look like function graphics, but perhaps this requires further fine tuning. --Wayne

---

Subject: Re: FSC_PLOT defaults
Posted by David Fanning on Tue, 21 Dec 2010 16:37:26 GMT
View Forum Message <> Reply to Message

Wayne Landsman writes:

> Yeah, the problem is that at least on my display the
> THICK keywords don't work in fractional amounts.
> XTHICK = 1.5 (which I would prefer) is the same as XTHICK=2.

Yes, I think in direct graphics, THICK will define the
number of pixels used for the lines. By itself, this isn't
too bad, but then the problem of no anti-aliasing raises
its ugly head. Things start to look a little "blocky".

I've just tried this on a contour plot, and the contour

line labels really start to bother me at this thickness.

I guess I am still preferring the original plots with
their single line thickness. I mostly use the display for
"quick looks" at the plot. Most of my output these days
is really produced automatically from PostScript files
and ImageMagick, which handles the anti-aliasing of
text, etc. for me. Fine tuning, of course, is trivial
in the PostScript file.

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: FSC_PLOT defaults
Posted by ben.bighair on Tue, 21 Dec 2010 17:53:48 GMT
View Forum Message <> Reply to Message

On 12/20/10 9:03 PM, Paulo Penteado wrote:

Hi,

> One of the main reasons I do not use object graphics is their reliance
> on global variables - especially those that do not get reset with
> a .reset_session. I usually object to global variables on any
> situation, as they make the results dependent on the environment, not
> just the code the user writes / receives (and, particularly, dependent
> not just on the routine calls).

I see what you mean. I wonder if this is a case we can resolve that
automatically - have the FSC_Defaults function check for the system
variable !FSC_Defaults and if it isn't there then load it. I can't
imagine that it adds much to overhead except the first time it is called
each session.

```
Function FSC_Default, what, value, SET = set, _REF_EXTRA = extra

  DEFSYSV, "!FSC_Default", EXISTS = ok
```

```
  IF (~ok) THEN BEGIN
RESTORE, <path-to-FSC_Defaults.pref>, RESTORED_OBJECTS = objs
DEFSYSV, "!FSC_Default", objs[1]
  ENDIF

  IF (KEYWORD_SET(set)) THEN $
    return, !FSC_Default->Set(what, value, _EXTRA = extra) ELSE $
    return, !FSC_Default->Get(what, _EXTRA = extra)

END;
```

> Also, it is awkward to use
> simultaneous instances of things that rely on (the same) global
> variables.

I'm a little fuzzy on this. How might this occur?

Cheers,
Ben

P.S. Isn't it grand thinking up things for David to do? Puts me in the holiday spirit!

Subject: Re: FSC_PLOT defaults
Posted by David Fanning on Tue, 21 Dec 2010 17:58:48 GMT
View Forum Message <> Reply to Message

Ben Tupper writes:

> P.S. Isn't it grand thinking up things for David to do? Puts me in the
> holiday spirit!

The Christmas dinner has already been rescheduled for later in the afternoon. :-(

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: FSC_PLOT defaults
Posted by Marc Buie on Tue, 21 Dec 2010 18:53:10 GMT
View Forum Message <> Reply to Message

Ben writes:

>> Also, it is awkward to use
>> simultaneous instances of things that rely on (the same) global
>> variables.
>
> I'm a little fuzzy on this. How might this occur?

This is a very serious issue, especially if you use any kind of widget programming (doesn't have to be object oriented stuff, either). With widgets you can have multiple copies of the same tool each looking at different data. You can also have different tools running at once. Each of these can run asynchronously and at any time. If you have a set of graphics tools that all depend on a global variable for its behavior (and perhaps modify it) then you are stuck with every tool taking on the same behavior. Sometimes that might be ok but mostly it's not what you want.

For this reason, I never, ever use the global variables to control plots unless it's in a throwaway script. In a program, I always use the graphics keywords to set the behavior. There are ways to make the global variables work for you but you have to be _very_ careful. For me, the price is always too high.

Subject: Re: FSC_PLOT defaults
Posted by ben.bighair on Tue, 21 Dec 2010 20:38:46 GMT
View Forum Message <> Reply to Message

On 12/21/10 1:53 PM, Marc Buie wrote:
> Ben writes:
>
>>> Also, it is awkward to use simultaneous instances of things that
>>> rely on (the same) global variables.
>>
>> I'm a little fuzzy on this. How might this occur?
>

Hi,

> This is a very serious issue, especially if you use any kind of
> widget programming (doesn't have to be object oriented stuff,
> either). With widgets you can have multiple copies of the same tool
> each looking at different data. You can also have different tools
> running at once. Each of these can run asynchronously and at any
> time. If you have a set of graphics tools that all depend on a
> global variable for its behavior (and perhaps modify it) then you are
> stuck with every tool taking on the same behavior. Sometimes that

> might be ok but mostly it's not what you want.

Hmm. I think every tool should take on the same default behavior, which programmers override with keywords as they do now.

> For this reason, I never, ever use the global variables to control
> plots unless it's in a throwaway script. In a program, I always use
> the graphics keywords to set the behavior. There are ways to make
> the global variables work for you but you have to be _very_ careful.
> For me, the price is always too high.

Yes, I can see how that would be a serious issue if the user modified the system variable. Even so, if the purpose is to define configurable default values then presumably the defaults are, well, the defaults when nothing else is specified. If one wanted something other than the default one would do as you do... use the graphics keyword to explicitly set the value.

The fatal flaw you point out is that the user will try to use the system variable so define the graphic state for Graphics A and also try to use the same variable to define the state for Graphics B which might be different. I guess I hadn't thought of that. I was thinking of David literally allowing the user to define fresh-session default values. If the programmer changes the default value between the realizations of Graphics A and Graphics B, well, then so be it. The default has been redefined.

But won't most programmers follow your suit by explicitly stating the graphics keywords?

Now that I stew on this, I am beginning to think I am more right than wrong. The default for Graphics A and Graphics B should point to the same defaults using David's FSC_* routines. They do now since David defines background = "white" as the default. Using a system variable as proposed, the default value for background isn't hardwired in the code as it is now. That is the only change.

Man-o-man! He could even make two or three different default starter kits available form his website:

"The Traditionalist Starter Kit"
Background = "Black"
Color = "White"

"The Fanning Starter Kit"
Background = "Ivory"
Color = "Charcoal"

"The Geezer Starter Kit"
 Background = "Black"
 Color = "White"
 Charsize = 20

"The Holiday Starter Kit"
 Background = "Green"
 Color = "Red"

Maybe another potential flaw might arise when I write code for someone else forgetting that the end-user may have different defaults than me. Well, I guess that's another reason to explicitly set the value of graphics keywords one code I'll be sharing with others.

Cheers,
Ben

Subject: Re: FSC_PLOT defaults
Posted by David Fanning on Tue, 21 Dec 2010 21:56:22 GMT
View Forum Message <> Reply to Message

Ben Tupper writes:

> The fatal flaw you point out is that the user will try to use the system
> variable so define the graphic state for Graphics A and also try to use
> the same variable to define the state for Graphics B which might be
> different. I guess I hadn't thought of that. I was thinking of David
> literally allowing the user to define fresh-session default values. If
> the programmer changes the default value between the realizations of
> Graphics A and Graphics B, well, then so be it. The default has been
> redefined.

The solution to *all* these problems, of course, is to have graphics routines that are objects. Then any individual object "knows" what it is suppose to do. The image objects in the Catalyst Library are the perfect examples of this.

It's all coming. But Christmas dinner has already been pushed back to the 28th. (Christmas shopping on the 27th now.) And Coyote is NO help! He's off to Costa Rica for two weeks with some young secretary he met at one of his "Christmas Parties". Sheesh.

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: FSC_PLOT defaults
Posted by Bruce Bowler on Wed, 22 Dec 2010 13:22:19 GMT
View Forum Message <> Reply to Message

On Tue, 21 Dec 2010 14:56:22 -0700, David Fanning set fingers to keyboard
and typed:

> (Christmas shopping on the 27th now.)

My neighbours, when I was growing up, always celebrated on the 31st. They
saved *lots* of money on wrapping paper etc :-)

Bruce

Subject: Re: FSC_PLOT defaults
Posted by lecacheux.alain on Wed, 22 Dec 2010 13:51:01 GMT
View Forum Message <> Reply to Message

On 21 déc, 16:58, David Fanning <n...@dfanning.com> wrote:
> Paul van Delst writes:
>> Wha..? Seriously?
>
>> I will agree that you can plot everything you need to plot with either direct or function (aka "new") graphics, but the
>> ability to easily manipulate the function graphic result, outside of its creation routine, after it's been created is a
>> huge step forward. And this step is built upon existing tools (object graphics), involving more packaging and interface
>> updates than actual new functionality. I find function graphics way more useful than iTools. Go figure.
>
> Paul, take it easy!
>
> I'd probably be a fan of function graphics, too, if
> I could get IDL 8 to work on my computer. But I'm

> stuck in IDL 7 for the time being, waiting for the
> problems to be sorted out. If I judge the people
> sending me e-mail and browsing my web page correctly,
> I'm not the only one wishing for some easy alternative.
>
> These are simple, they are fast, they produce the
> PostScript output you expect. You can actually
> write programs with them (as opposed, say, to just
> display data). There are just a lot of things to like
> about them. But, you don't have to use them if you
> don't want to. :-)
>
> Anyway, this is just a stop-gap until I can get IDL
> 8 running. Then I have some ideas for function graphics
> commands that work the way I want them to. Maybe you
> will like those better. ;-)
>
> But, yes, seriously! I don't feel any great need for
> anything fancier than direct graphics. If you can
> make a PostScript file and you can install ImageMagick,
> most of the things I have complained about for years
> are solved. :-)
>
> 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.")

Regarding change of graphic defaults at the plot level or not, it can
be done, I guess, by using the STYLE_NAME keyword in ITOOLS (then
likely available in NG).
alx.

Subject: Re: FSC_PLOT defaults
Posted by David Fanning on Wed, 22 Dec 2010 17:11:37 GMT
View Forum Message <> Reply to Message

Wayne Landsman writes:

> I'm old-fashioned enough to think that black on white does
> not show up as clearly on the screen.

Here is another test program. Maybe this will help. If you
prefer black backgrounds in your display graphics windows,
you can set the TRADITIONAL keyword on FSC_Plot:

IDL> FSC_PLot, LoadData(17), /TRADITIONAL

This will still give you white backgrounds in PostScript.

To make this work (since this group convinced me to make the
BACKGROUND color work *everywhere*), I have to "assume" you
do not ever *really* want black backgrounds in PostScript. If
you *do* want a black background in PostScript, then you
will also have to specify the COLOR and AXISCOLOR keywords,
otherwise I won't believe you. :-)

The test program is here:

http://www.dfanning.com/misc/fsc_plot_test.pro

Let me know what you think. Does this solve your problem?

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: FSC_PLOT defaults
Posted by wlandsman on Thu, 23 Dec 2010 16:38:33 GMT
View Forum Message <> Reply to Message

On Wednesday, December 22, 2010 12:11:37 PM UTC-5, David Fanning wrote:

> Here is another test program. Maybe this will help. If you
> prefer black backgrounds in your display graphics windows,
> you can set the TRADITIONAL keyword on FSC_Plot:
>
> IDL> FSC_PLot, LoadData(17), /TRADITIONAL
>

Thanks. My personal preference is for direct graphics to look like function graphics (black on white background with thicker and larger lines and characters), especially since I am currently

spending about half my time with each graphics system. But if I am going to drop-in replace PLOT with FSC_PLOT in legacy programs, then I would use the /TRADITIONAL keyword. I have found that otherwise FSC_PLOT can introduce conflicts with program that for example assume a default color of !D.TABLE_SIZE-1.

-Wayne

Subject: Re: FSC_PLOT defaults
Posted by David Fanning on Thu, 23 Dec 2010 18:08:07 GMT
View Forum Message <> Reply to Message

Wayne Landsman writes:

> I have found that otherwise FSC_PLOT can introduce conflicts
> with program that for example assume a default color of
> !D.TABLE_SIZE-1.

Can you provide me with more details on this. I don't
see how this is possible. :-(

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: FSC_PLOT defaults
Posted by David Fanning on Thu, 23 Dec 2010 19:10:09 GMT
View Forum Message <> Reply to Message

Folks,

I have updated FSC_Plot, FSC_Contour, and FSC_Surf to
have a TRADITIONAL keyword that will use the "traditional"
graphics colors of white on black for graphics windows
and black on white for PostScript.

While this provides *some* benefit, it pretty much defeats
the purpose of making it easy to write programs that work
the same everywhere, which is the secret (I suppose it can

be told now) of making direct graphics commands extremely
useful for web display, Powerpoint presentations, and
sharing information with colleagues.

But, I aim to please. :-)

   http://www.dfanning.com/programs/fsc_plot.pro
   http://www.dfanning.com/programs/fsc_surf.pro
   http://www.dfanning.com/programs/fsc_contour.pro

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: FSC_PLOT defaults
Posted by lecacheux.alain on Thu, 23 Dec 2010 20:34:56 GMT
View Forum Message <> Reply to Message

On 23 déc, 17:38, wlandsman <wlands...@gmail.com> wrote:
> On Wednesday, December 22, 2010 12:11:37 PM UTC-5, David Fanning wrote:
>> Here is another test program. Maybe this will help. If you
>> prefer black backgrounds in your display graphics windows,
>> you can set the TRADITIONAL keyword on FSC_Plot:
>
>>   IDL> FSC_PLot, LoadData(17), /TRADITIONAL
>
> Thanks. My personal preference is for direct graphics to look like function graphics (black on white background with thicker and larger lines and characters), especially since I am currently spending about half my time with each graphics system. But if I am going to drop-in replace PLOT with FSC_PLOT in legacy programs, then I would use the /TRADITIONAL keyword.
  I have found that otherwise FSC_PLOT can introduce conflicts with program that for example assume a default color of !D.TABLE_SIZE-1.
>
> -Wayne

Please try:
   IDL> p1 = plot(/TEST, STYLE_NAME='IDL classic')
   IDL> p1.save, 'p1.ps'
   IDL> p2 = plot(/TEST, STYLE_NAME='IDL standard')
   IDL> p2.save, 'p2.ps'

then compare p1, p2 plots and PostScript files.
You can change nearly everything by using the style editor and create
your own style in a consistent way.
alx.

Subject: Re: FSC_PLOT defaults
Posted by Mark[1] on Thu, 23 Dec 2010 20:35:13 GMT
View Forum Message <> Reply to Message

On Dec 21, 3:03 pm, Paulo Penteado <pp.pente...@gmail.com> wrote:
> On Dec 20, 11:43 pm, Ben Tupper <ben.bigh...@gmail.com> wrote:
>
> One of the main reasons I do not use object graphics is their reliance
> on global variables

Are you sure you mean object graphics?

Subject: Re: FSC_PLOT defaults
Posted by penteado on Thu, 23 Dec 2010 20:53:14 GMT
View Forum Message <> Reply to Message

On Dec 23, 6:35 pm, Mark <mark.h...@gmail.com> wrote:
> On Dec 21, 3:03 pm, Paulo Penteado <pp.pente...@gmail.com> wrote:
>
>> On Dec 20, 11:43 pm, Ben Tupper <ben.bigh...@gmail.com> wrote:
>
>> One of the main reasons I do not use object graphics is their reliance
>> on global variables
>
> Are you sure you mean object graphics?

Thanks for catching that. I must had been asleep. I meant direct
graphics, of course.

Subject: Re: FSC_PLOT defaults
Posted by wlandsman on Thu, 23 Dec 2010 23:42:44 GMT
View Forum Message <> Reply to Message

Fresh session on Linux or Mac IDL

IDL> fsc_plot,indgen(10),color=255

shows only the axes and not the plot. A lot of legacy plotting code has code fragments like the following:

pro test,color=color

if N_elements(color) EQ 0 then color = !D.TABLE_SIZE-1
plot,x,y,color=color

--Wayne

On Thursday, December 23, 2010 1:08:07 PM UTC-5, David Fanning wrote:
> Wayne Landsman writes:
>
>> I have found that otherwise FSC_PLOT can introduce conflicts
>> with programsthat for example assume a default color of
>> !D.TABLE_SIZE-1.
>
> Can you provide me with more details on this. I don't
> see how this is possible. :-(

Subject: Re: FSC_PLOT defaults
Posted by wlandsman on Fri, 24 Dec 2010 00:02:08 GMT
View Forum Message <> Reply to Message

On Thursday, December 23, 2010 3:34:56 PM UTC-5, alx wrote:

>
> Please try:
> IDL> p1 = plot(/TEST, STYLE_NAME='IDL classic')
> IDL> p1.save, 'p1.ps'
> IDL> p2 = plot(/TEST, STYLE_NAME='IDL standard')
> IDL> p2.save, 'p2.ps'
> then compare p1, p2 plots and PostScript files.
> You can change nearly everything by using the style editor and create
> your own style in a consistent way.

Neat! The only problem is that I couldn't find STYLE_NAME in the documentation for PLOT(). (It is in the documentation for IPLOT but even there I couldn't a list of the names of the system plot styles.) --Wayne

---

Subject: Re: FSC_PLOT defaults
Posted by David Fanning on Fri, 24 Dec 2010 02:03:48 GMT
View Forum Message <> Reply to Message

wlandsman writes:

> IDL> fsc_plot,indgen(10),color=255

>
> shows only the axes and not the plot. A lot of legacy plotting code has code fragments like the following:
>
> pro test,color=color
>
> if N_elements(color) EQ 0 then color = !D.TABLE_SIZE-1
> plot,x,y,color=color

Ah, right. I guess the equivalent would
be asking for the take-out pizza to be
put in a box. :-)

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: FSC_PLOT defaults
Posted by Michael Galloy on Fri, 24 Dec 2010 05:05:10 GMT
View Forum Message <> Reply to Message

On 12/23/10 5:02 PM, wlandsman wrote:
> On Thursday, December 23, 2010 3:34:56 PM UTC-5, alx wrote:
>
>>
>> Please try:
>> IDL> p1 = plot(/TEST, STYLE_NAME='IDL classic')
>> IDL> p1.save, 'p1.ps'
>> IDL> p2 = plot(/TEST, STYLE_NAME='IDL standard')
>> IDL> p2.save, 'p2.ps'
>> then compare p1, p2 plots and PostScript files.
>> You can change nearly everything by using the style editor and create
>> your own style in a consistent way.
>
> Neat! The only problem is that I couldn't find STYLE_NAME in the documentation for PLOT(). (It is in the documentation for IPLOT but even there I couldn't a list of the names of the system plot styles.) --Wayne

The two system plot styles are 'IDL Classic' (white on black) and 'IDL

Standard' (black on white):

```
IDL iplot, findgen(11), style_name='idl classic'
```

or:

```
IDL> iplot, findgen(11), style_name='idl standard'
```

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

Subject: Re: FSC_PLOT defaults
Posted by David Fanning on Fri, 24 Dec 2010 18:55:24 GMT
View Forum Message <> Reply to Message

Wayne Landsman writes:

> Fresh session on Linux or Mac IDL
>
> IDL> fsc_plot,indgen(10),color=255
>
> shows only the axes and not the plot. A lot of legacy plotting code has code fragments like the following:
>
> pro test,color=color
>
> if N_elements(color) EQ 0 then color = !D.TABLE_SIZE-1
> plot,x,y,color=color

OK, I have solved this problem. It required a new program ColorsAreIdentical, which I still need to document, etc.

I also decided that whenever possible (6.4 and above in Z, 7.1 and above in PostScript, any version of anything else) I would draw graphics using decomposed color so I don't ever load colors in the color table.

Anyway, a fair number of changes and some new programs. I'll be testing in the next day or so to be sure everything still works the way I think it is suppose to work. Look for the updates soon. :-)

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: FSC_PLOT defaults
Posted by David Fanning on Sat, 25 Dec 2010 06:01:36 GMT
View Forum Message <> Reply to Message

Wayne Landsman writes:

> Fresh session on Linux or Mac IDL
>
> IDL> fsc_plot,indgen(10),color=255
>
> shows only the axes and not the plot. A lot of legacy plotting code has code fragments like the following:
>
> pro test,color=color
>
> if N_elements(color) EQ 0 then color = !D.TABLE_SIZE-1
> plot,x,y,color=color

All of the Coyote Graphics commands have now been
updated to handle the situations described here
correctly. This required a couple of new programs,
and modifications to a number programs. Best to get
a fresh copy of the Coyote Library, I think. :-)

http://www.dfanning.com/programs/coyoteprograms.zip

Merry Christmas to all!

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: FSC_PLOT defaults
Posted by lecacheux.alain on Mon, 27 Dec 2010 08:51:59 GMT
View Forum Message <> Reply to Message

On 24 déc, 01:02, wlandsman <wlands...@gmail.com> wrote:
> On Thursday, December 23, 2010 3:34:56 PM UTC-5, alx wrote:
>
>> Please try:
>> IDL> p1 = plot(/TEST, STYLE_NAME='IDL classic')
>> IDL> p1.save, 'p1.ps'
>> IDL> p2 = plot(/TEST, STYLE_NAME='IDL standard')
>> IDL> p2.save, 'p2.ps'
>> then compare p1, p2 plots and PostScript files.
>> You can change nearly everything by using the style editor and create
>> your own style in a consistent way.
>
> Neat! The only problem is that I couldn't find STYLE_NAME in the documentation for PLOT(). (It is in the documentation for IPLOT but even there I couldn't a list of the names of the system plot styles.) --Wayne

That is the bad point. I understand NG as built on ITOOLS with simpler and more appealing syntax. Unfortunately NG's documentation is quasi non existing: you must rely on "Itools user guide from 7.1" (in particular on chapter on ITools procedural interface). Furthermore, some ITOOLS capabilities have disappeared in NG! For instance, you still can use the undocumented STYLE_NAME keyword, but the "style editor" is no longer available (afaik) in NG (you must open an ITools to get it). You still can create a new (personal) style for NG, like in v7.1, but you are no longer able to select it from v8 preferences... A mess, which might definitely discourage NG users !
alx.