
Subject: Re: axis problem

Posted by [JD Smith](#) on Mon, 21 Jul 2003 15:25:45 GMT

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

On Sun, 20 Jul 2003 13:42:36 -0700, Reimar Bauer wrote:

> David Fanning wrote:

>

>> Reimar Bauer writes:

>>

>>> David I can't agree

>>>

>>> It can't be that the user program has to test what keywords values are
>>> set as default by a routine and if it has this value then it must be
>>> killed from the _extra structure if it is there.

>>>

>>> With all the other keywords it works as supposed. It would be very bad
>>> if it is somewhere defined that's the user is not able to pass default
>>> values by _extra. It must be possible to switch back to the whatever
>>> default value by submitting 0 for example.

>>>

>>> At the moment I believe there is a bug with the querying of xyz minor.
>>> They used keyword_set() instead of n_elements() and ...

>>

>> I'm not so sure of my answer that I would bet a whole lot of money on
>> the "no bug" theory, but still...

>>

>> Think of how you would do this. A keyword has a value of 5 by default.
>> If the keyword is set to 0, you which to set the value to 5. You would
>> write the program like this:

>>

```
>> PRO MyPlot, KEY=key, _Extra=extra  
>> IF N_Elements(key) EQ 0 THEN key = 5  
>> IF key EQ 0 THEN key = 5
```

>>

```
>> PLOTSOMETHING, Key=key, _Extra=extra  
>> END
```

>>

>> Now, if you pass a value in with the keyword, you encounter the
>> "processing".

>>

```
>> IDL> MyPlot, KEY=0
```

>>

>> If you pass it in via the _EXTRA mechanism, you bypass the processing:

>>

```
>> IDL> MyPlot, _Extra={KEY:0}
```

>>

>> This seems quite reasonable to me. The alternative would be to put

```

>> something like this into your program:
>>
>> IF N_ELEMENTS(extra) NE 0 THEN BEGIN
>>   tagnames = Tag_Names(extra)
>>   index = WHERE(tagnames EQ 'K', count) IF count GT 0 THEN IF
>>   extra.(index) EQ 0 THEN key = 5 index = WHERE(tagnames EQ 'KE',
>>   count) IF count GT 0 THEN IF extra.(index) EQ 0 THEN key = 5 index
>>   = WHERE(tagnames EQ 'KEY', count) IF count GT 0 THEN IF
>>   extra.(index) EQ 0 THEN key = 5
>>   ENDIF
>>
>> Think what would happen if you wrote a long keyword name, or if you had
>> multiple keywords defined that you had to chase down like this. You
>> would spend all your time writing code and no time at all drinking
>> beer. :-(
>>
>> Cheers,
>>
>> David
>
> Dear David,
>
> I think they have probably written a function like this,
>
> FUNCTION is_keyword,keyword,names
>
> ix=where(strpos(names,keyword) eq 0 ,count_ix) if count_ix eq 1 then
> return,names[ix]$
> else message,'Ambiguous keyword abbreviation '+keyword,/cont
>
> end
>
> And with routine_info you get the names of the keywords of the routine
> But I can't do this myself with plot
>
> print,is_keyword('xmin',['xminor','xaxis']) xminor
> print,is_keyword('x',['xminor','xaxis'])
> print,is_keyword('x',['xminor','xaxis']) % IS_KEYWORD: Ambiguous keyword
> abbreviation x
>   0
>

```

That's quite a creative set of theories, but the explanation is much simpler:

Compare:

```
axis,xaxis=0,xticks=4,color=2,xminor=0
```

and

```
axis,xaxis=0,xticks=4,color=2,xminor=0,xstyle=1
```

XSTYLE=1, combined with XTICKS=4, is the source of your trouble. You're forcing IDL to divide the axis range into non-whole units, which means that the minor units cannot be simply specified. Rather than approximate minor tick positions, IDL just skips minor ticks altogether. You can specify the minor tickmark count yourself with, e.g., XMINOR=5, to get the effect you want. You might consider this a bug that IDL doesn't do this for you. I'd probably just drop the XTICKS specification.

The inherited vs. direct issue was a red herring, since you didn't compare apples to apples, specifying many more parameters in the _EXTRA version. As far as I know, all _EXTRA processing, abbreviation expansion, keyword ambiguity checking, etc., is done at a higher level, such that individual routines have no information on how their keywords arrived. This is a good thing, since otherwise you'd have all kinds of differing treatments of abbreviations, keyword inheritance, etc.

JD
