
Subject: Re: `_EXTRA` inheritance crashing IDL
Posted by [Mark Hadfield](#) on Thu, 11 Apr 2002 21:51:48 GMT
[View Forum Message](#) <> [Reply to Message](#)

"Rob Preece" <Rob.Preece@msfc.nasa.gov> wrote in message
news:Rob.Preece-1104021521040001@biggamma.nsstc.nasa.gov...

>
> I have a *very* good reason for doing it this way (otherwise, why
> bother! ;) I have a set of keywords to pass on to an embedded plot
> routine, and I can't know at the time what their values will be. I
> have a plot zooming function in a widget (object) that simply calls
> the object's 'PLOT' method, passing new x and y ranges. The plot
> method does some stuff, and then hands it all off to the IDL
> (direct) PLOT routine, adding in the 'XRANGE' and 'YRANGE' keywords
> so that they can be overridden when the 'ZOOM' method is
> invoked. Very much like 'test1' in my sample code. Since I followed
> the documentation as Mark H. mentioned, is this a valid bug in IDL?
> How to proceed?

Yes it's a bug. Report it to IDL.

Re the need to do it this way, if you're passing specific keywords
then you don't *need* to use inheritance. Eg this...

```
PRO test1, _REF_EXTRA=extra
  test2, INDGEN (10), _EXTRA = ['TITLE','XTITLE','YTITLE', 'XRANGE']
END
```

...can be written...

```
PRO test1, TITLE=title, XTITLE=xtitle, YTITLE=ytitle, XRANGE=xrange
  test2, INDGEN (10), TITLE=title, XTITLE=xtitle, YTITLE=ytitle, $
  XRANGE=xrange
END
```

(I note that your original test1 was a little more complicated than
the one I have written. Was there a reason for that?)

If you want to direct different keywords to different routines you can
do this.

```
PRO test1, TITLE=title, XTITLE=xtitle, YTITLE=ytitle, XRANGE=xrange
  test2, INDGEN (10), TITLE=title, XTITLE=xtitle,
  test3, YTITLE=ytitle, XRANGE=xrange
END
```

or even

```
PRO test1, TITLE=title, XTITLE=xtitle, YTITLE=ytitle, XRANGE=xrange, $
  _EXTRA=extra
  test2, INDGEN (10), TITLE=title, XTITLE=xtitle,
  test3, YTITLE=ytitle, XRANGE=xrange
  test4, _EXTRA=extra
END
```

Defaults can be supplied and overridden for explicit or inherited keywords, eg:

```
PRO test1, KEY1=key1, _EXTRA=extra
  if n_elements(key1) eq 0 then key1 = 33
  test2, indgen(10), KEY1=key1, KEY2=33, _EXTRA
END
```

The default for key1 is obvious, What's not so obvious is that if I call

```
IDL> test1, KEY2=66
```

then this value is passed to test2, overriding 33.

Warning: if you use this last technique with IDL 5.4 you must use inheritance by value not reference because there was a bug in 5.4.

Of course with the different bugs in different versions you may well decide to leave your code as it is and stick with 5.4!

--

Mark Hadfield
m.hadfield@niwa.co.nz Ka puwaha et tai nei
http://katipo.niwa.co.nz/~hadfield Hoesa tatou
National Institute for Water and Atmospheric Research (NIWA)
