Subject: Re: Keyword precedence
Posted by Mark Hadfield on Mon, 28 Aug 2000 23:32:06 GMT
View Forum Message <> Reply to Message

"J.D. Smith" <jdsmith@astro.cornell.edu> wrote in message
news:39AA8AFE.CDBAB7E6@astro.cornell.edu...

> ...Just as with normal
> positional parameters, the programmer must be sure to define in advance
how each
> will be used: for input values, for return values, or for both. This
affects
> their usage!...

It's interesting that David Fanning and Martin Shultz have both recommended the following idiom for establishing overridable defaults

```
pro my_plot, COLOR=color, _EXTRA=extra
   if n_elements(color) eq 0 then color = 12
   plot, COLOR=12, _EXTRA=extra
end
```

This has the effect, unintended and normally irrelevant, that if the following call is made with the COLOR keyword set to an undefined variable

```
my_plot, COLOR=color
```

then this variable is set to 12 on output. It isn't too hard to imagine a situation (successive calls to different routines with different default colours) where this will bite an unwary programmer, though in several years of using this idiom I have seldom thought about this side-effect and have very seldom been bitten.

My point: in many situations IDL programmers are pretty relaxed about whether values are modified on output because it has no effect on how their programs operate. As far as possible the language should avoid punishing them for this.

> ...More scary is the notion of:
>
> mgh_example_keywords_reference_wrapper, COLOR=color, _EXTRA=extra
>
> putting a return value somewhere other than in the variable "color"...
maybe not
> even on this level... maybe n levels up somewhere.

Yes, that would be the effect of my proposals for precedence. And it is the current situation (which acts exactly the way I am proposing except in a specific, albeit common, case). I think you have to accept that by putting _EXTRA or _REF_EXTRA in your code you are passing power, and responsibility to the next level up.

> ...I
> think you are imagining cases in which you don't have control over the
> inheritance chain, and may not know you are using _REF_EXTRA.

Yes. And I was bitten in a case where I did have control over the inheritance chain but had forgotten which particular mechanism I had used.

> In any case, hopefully an RSI person or two will get the basic notion that
this
> needs to be straightened up. And for those of you who have resolved never
to
> include _REF_EXTRA in your programs, please be assured that this really
affects
> only a very limited subset of cases of use.

Agreed.

I can't resist adding another tidbit: CALL_PROCEDURE passes keywords through in both directions, but it doesn't behave like my "reference wrapper". It always gives precedence to extra keywords, irrespective of whether the names match exactly.

Mark Hadfield
m.hadfield@niwa.cri.nz http://katipo.niwa.cri.nz/~hadfield/
National Institute for Water and Atmospheric Research
PO Box 14-901, Wellington, New Zealand