Subject: Re: Curious Keywords
Posted by davidf on Thu, 27 Feb 1997 08:00:00 GMT
View Forum Message <> Reply to Message

JD Smith writes in a follow-up to this thread:

> The question is really about the mechanism of Keyword Checking for built-ins
> vs. non-built-ins. That is, if *I* wrote a routine that took the "THICK"
> keyword, a user could pass an undefined value through THICK and I would never
> know that he did. I could not use n_elements() or keyword_set() or any other
> mechanism I know of to discern in any way that he has used "THICK". And so,
> consequently, I would not be able to issue an error message in the case he
> passed an undefined variable. It is as if he never passed it at all! Not so
> for Plot and other built-in routines (for *some* of their keywords). They
> somehow "know" that I used e.g., "THICK", even when I pass them an undefined
> variable.
>
> My point was that this would be a useful feature to have (although it would
> possibly result in some subtle and harmful programmatic issues). For
> instance, if I pass a variable through a keyword into which I'd like to put
> the result of some calculation, I have to give that variable a value before
> passing it, in order to test whether to go through the bother of doing the
> calculation at all. There are, of course, other ways to do this (e.g.
> optional *parameters*), but I've found myself wishing for this particular
> mechanism on some occassions, when the other techniques had limitations.

JD is absolutely right about this. You cannot tell in an IDL procedure
if a keyword is *used* or not. You can only tell if the argument to the
keyword is *defined* or not.

(Most of the programs I look at, by the way, mistakenly think this
is what KEYWORD_SET is doing for them. Sigh...)

Since knowing if a keyword is *used* is often quite useful for
exactly the reasons JD mentions, I think it should be added to IDL.
Obviously IDL knows if the keyword is being used.

I think, if I remember correctly, that PV-Wave has a function
entitled KEYWORD_USED (or something like it) that does this very thing.

Cheers!

David

David Fanning, Ph.D.
Fanning Software Consulting
2642 Bradbury Court, Fort Collins, CO 80521

Phone: 970-221-0438 Fax: 970-221-4762
E-Mail: davidf@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com

Subject: Re: Curious Keywords
Posted by J.D. Smith on Thu, 27 Feb 1997 08:00:00 GMT
View Forum Message <> Reply to Message

Kevin Spencer wrote:

> The reason for this behavior is that the default value of UNDEF in both
> cases is 0. For the POLAR keyword, supplying a 0 means don't do anything,
> so it doesn't. But supplying 0 to THICK generates an error because the
> line has to have *some* thickness to it.
>
> Dig?

O Contraire!

Please try:

```
plot, findgen(100), findgen(100)/99.*2*!PI, THICK=0
```

Works pretty nicely.

The question is really about the mechanism of Keyword Checking for built-ins vs. non-built-ins. That is, if *I* wrote a routine that took the "THICK" keyword, a user could pass an undefined value through THICK and I would never know that he did. I could not use n_elements() or keyword_set() or any other mechanism I know of to discern in any way that he has used "THICK". And so, consequently, I would not be able to issue an error message in the case he passed an undefined variable. It is as if he never passed it at all! Not so for Plot and other built-in routines (for *some* of their keywords). They somehow "know" that I used e.g., "THICK", even when I pass them an undefined variable.

My point was that this would be a useful feature to have (although it would possibly result in some subtle and harmful programmatic issues). For instance, if I pass a variable through a keyword into which I'd like to put the result of some calculation, I have to give that variable a value before passing it, in order to test whether to go through the bother of doing the calculation at all. There are, of course, other ways to do this (e.g. optional *parameters*), but I've found myself wishing for this particular mechanism on some occassions, when the other techniques had limitations.

JD

Subject: Re: Curious Keywords
Posted by kspencer on Thu, 27 Feb 1997 08:00:00 GMT
View Forum Message <> Reply to Message

John-David Smith <jdsmith@astrosun.tn.cornell.edu> writes:

> Hi folks,

> A Keyword question:

> I was curious how IDL built-ins can distiguish a passed keyword with an
> undefinded variable from a keyword never passed at all, and why this
> feature is not implemented in the language itself.

> E.G. :

> I can say (for some undefined variable UNDEF):

> plot, findgen(100), findgen(100)/99.*2*!PI,POLAR=UNDEF

> and get a straight line. (Or I can set UNDEF=1 and get a spiral o'
> Archimedes).

> I cannot, however, say:

> plot, findgen(100), findgen(100)/99.*2*!PI, THICK=UNDEF

> Since this generates an "a undefined" error message. This implies that
> plot *knows* I am using the keyword "THICK", even when I am passing an
> undefined variable -- a possibility which most of us discovered the hard
> way is not implementable in IDL code. That is, if IDL's plot routine
> were written in IDL code, the previous call would be equivalent, for all
> practical purposes, to a call in which the keyword were ommitted.

> Any thoughts?

> JD

The reason for this behavior is that the default value of UNDEF in both
cases is 0. For the POLAR keyword, supplying a 0 means don't do anything,
so it doesn't. But supplying 0 to THICK generates an error because the
line has to have *some* thickness to it.

Dig?

Kevin

Kevin Spencer
Cognitive Psychophysiology Laboratory and Beckman Institute
University of Illinois at Urbana-Champaign
kspencer@p300.cpl.uiuc.edu

Subject: Re: Curious Keywords
Posted by thompson on Mon, 03 Mar 1997 08:00:00 GMT
View Forum Message <> Reply to Message

davidf@dfanning.com (David Fanning) writes:

> JD Smith writes in a follow-up to this thread:

(stuff deleted)

> JD is absolutely right about this. You cannot tell in an IDL procedure
> if a keyword is *used* or not. You can only tell if the argument to the
> keyword is *defined* or not.

> (Most of the programs I look at, by the way, mistakenly think this
> is what KEYWORD_SET is doing for them. Sigh...)

> Since knowing if a keyword is *used* is often quite useful for
> exactly the reasons JD mentions, I think it should be added to IDL.
> Obviously IDL knows if the keyword is being used.

> I think, if I remember correctly, that PV-Wave has a function
> entitled KEYWORD_USED (or something like it) that does this very thing.

On the other hand, I find it quite frustrating that I can't use a phrase like

```
IDL> plot, findgen(100), findgen(100)/99.*2*!PI, THICK=THICK
```

and not worry about whether or not THICK was defined. In some sense this is what _EXTRA does, but that has its own limitations.

Of course, there have been instances when I wanted to do exactly what JD Smith was talking about. The most obvious example is when I wanted error messages to be returned to the user rather than printed on the screen. The only way to do this was to force the calling routine to define the keyword before calling the program, e.g.

```
ERRMSG = ''
MYROUTINE, ERRMSG=ERRMSG, ...
IF ERRMSG NE '' THEN ...
```

As pointed out, this is a pain.

In summary, I guess what would be best would be if the built-in routines were more forgiving about passing undefined keywords, and also if there were a KEYWORD_USED (or whatever) routine that would allow .PRO files to figure out for themselves whether or not a keyword was really passed, even if undefined. That way, we could eat our cake and have it too.

I would also like to echo David's comment about KEYWORD_SET(). I also find it upsetting when people use this when they should have used N_ELEMENTS() instead. (Double sigh.)

Bill Thompson
