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