Subject: Re: keyword_set bug or feature
Posted by thompson on Thu, 25 May 1995 07:00:00 GMT
View Forum Message <> Reply to Message

soc@festival.ed.ac.uk (Stephen O'Connell) writes:

> Mark Rivers (rivers@cars3.uchicago.edu) wrote:

> : In article <PHIL.95May12090139@peace.med.ohio-state.edu>,
phil@peace.med.ohio-state.edu (Phil) writes:

> : >
> : > IF NOT(keyowrd_set(key)) THEN BEGIN
> : > ;set a default value here
> : > ENDIF
> : >
> : >The problem comes in if the user wants to set key = 0. If so then it
> : >appears to the above test that the keyword is not set even though in
> : >the function call the user typed
> : >

> : keyword_set is intended for use with switches, i.e. parameters which can either
> : be 0 or 1. 0 or not present means switch not set, 1 means switch set.

> : If you want to detect that the keyword is present, even if the value is zero,
> : then you should use n_elements().

> Keyword is more than just 1/0 switches! ...

I agree. There are basically two kinds of keywords, the kind where you can pass back-and-forth any kind of IDL variable, and those which can be either "set" or "not set" (on/off). This distinction is evident from the typical syntax used for the two kinds. For example, in the IDL statement

```
 IDL> PLOT, X, Y, XRANGE=[100,200], /YNOZERO
```

the keyword XRANGE is of the first type, while YNOZERO is of the second. One could have done the same thing with YNOZERO=1, but using the /YNOZERO notation makes it clearer that this is an on/off kind of parameter.

A keyword can be considered to be "set" if

1. It is equal to a non-zero value (typically 1).

2. It is passed with the /keyword notation

and it is "not set" if

1. It is equal to zero.

2. It is passed as a non-existent variable.

3. It is not passed at all.

KEYWORD_SET was designed to sort out these possibilities. It does exactly what it was designed to do. The fact that it returns 0 if the keyword was set equal to zero is not a bug but an integral and extremely necessary part of its function.

What you're looking for is not whether a keyword was set or not, but whether it was passed. Recently, somebody posted an IDL procedure called KEYWORD_PASSED which does what you want to do.

You also write

> Andd although its true that you can use n_elements, you're still buggered if
> you have more than one keyword...

I don't understand what you mean by that. You can use N_ELEMENTS separately for each keyword passed. Perhaps you're thinking about N_PARAMS?

Cheers,

Bill Thompson
