Subject: Re: keyword_set bug or feature
Posted by peter on Fri, 12 May 1995 07:00:00 GMT
View Forum Message <> Reply to Message

Phil (phil@peace.med.ohio-state.edu) wrote:
: Thanks to those who responded to my ? concerning common blocks.

: I just can across another 'feature' and was wondering if others felt
: that it is really a 'bug' or if there is some way to over come it. It
: has to do with keywords in functions or procedures. I use several and
: check to see if the are set using

: 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
:
: Result = somefunction(var,key=0)

: What gives? Can I just add key = key + 1 and key = key - 1 around the
: testing if statement?

This is the specified behavior of keyword_set, and yes, it has caused
problems for others (i.e. me!). The solution to your problem is to
used n_elements(key) to test whether any value was set, or no value
at all. n_elements will return 0 on an undefined variable, and 1
on any scalar, including 0.

Peter

Subject: Re: keyword_set bug or feature
Posted by rivers on Sat, 13 May 1995 07:00:00 GMT
View Forum Message <> Reply to Message

In article <PHIL.95May12090139@peace.med.ohio-state.edu>, phil@peace.med.ohio-state.edu
(Phil) writes:
> Thanks to those who responded to my ? concerning common blocks.
>
> I just can across another 'feature' and was wondering if others felt
> that it is really a 'bug' or if there is some way to over come it. It
> has to do with keywords in functions or procedures. I use several and
> check to see if the are set using
>

>  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().


Mark Rivers (312) 702-2279 (office)
CARS (312) 702-9951 (secretary)
Univ. of Chicago (312) 702-5454 (FAX)
5640 S. Ellis Ave. (708) 922-0499 (home)
Chicago, IL 60637 rivers@cars3.uchicago.edu (Internet)


---

Subject: Re: keyword_set bug or feature
Posted by chase on Mon, 15 May 1995 07:00:00 GMT
View Forum Message <> Reply to Message

>>>> > "Rob" == Stephen O'Connell <soc@festival.ed.ac.uk> writes:

Rob> Keyword is more than just 1/0 switches! And if you are using them
Rob> to pass values, you just might mistakenly set one equal to zero.
Rob> Andd although its true that you can use n_elements, you're still
Rob> buggered if you have more than one keyword...so it is a bug
Rob> really, in my opinion. Maybe ver. 4.0 has it sorted - does
Rob> anyone know ?

It is not a bug. KEYWORD_SET works exactly as documented in the "IDL Reference Guide". Furthermore, the guide states: "This function is especially useful in user-written procedures and functions that process keywords that are interpreted as being either true (keyword present and nonzero) or false (keyword was not used, or was set to zero)." Thus, its purpose is not for checking if a keyword is used to pass values. Just because KEYWORD_SET does not do what you wish does not mean it has a bug. Use

N_ELEMENTS(var) NE 0

to check for defined keywords. You can make your own function

"KEYWORD_DEFINED()" if you think it is more convenient than
"N_ELEMENTS(var) NE 0".

Chris

--
==========
Bldg 24-E188
The Applied Physics Laboratory
The Johns Hopkins University
Laurel, MD 20723-6099
(301)953-6000 x8529
chris.chase@jhuapl.edu