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

phil@peace.med.ohio-state.edu (Phil) writes:

> 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?

I always use a statement like the following:

```
  if n_elements(key) eq 0 then key=default_value
```

This is not something you would guess as the obvious solution but n_elements returns 0 if its argument is undefined. You only need a default value when your keyword is undefined so this works. It also works the same for ordinary positional arguments.

What is missing is a way to determine if an output keyword was requested in the call. Often the variable in such a keyword is undefined since it is an output of the routine. I don't know anyway to find out if the caller actually requested the output keyword or not so any such keywords must always be set. Occasionally it may take a lot of processing to find the desired output. It would be nice if you could determine if it was really requested and only do it then.

Ray Sterner   sterner@tesla.jhuapl.edu
The Johns Hopkins University   North latitude 39.16 degrees.
Applied Physics Laboratory   West longitude 76.90 degrees.
Laurel, MD 20723-6099
WWW Home page: ftp://fermi.jhuapl.edu/www/s1r/people/res/res.html