
Subject: Re: Keyword checking

Posted by [Paul Van Delst\[1\]](#) on Wed, 29 May 2002 14:15:47 GMT

[View Forum Message](#) <> [Reply to Message](#)

Randall Skelton wrote:

>
> I have run into a great deal of trouble checking keywords in IDL and I
> thought I would relay my thoughts and frustrations. I am trying to
> prevent a user from passing an undefined variable in a keyword...

>
> Imagine,

>
> pro test, A=a, B=b
> print, n_params()
> print, arg_present(a), keyword_set(a), n_elements(a), size(a,/type)
> print, arg_present(b), keyword_set(b), n_elements(b), size(b,/type)
> end

>
> If I call this routine as:

>
> IDL> .reset
> IDL> test, a=1, b=undefvar
> % Compiled module: TEST.
> 0
> 0 1 1 2
> 1 0 0 0

> So, in order to prevent a user from passing junk in a keyword, I have:

>
> pro test, B=b
> if n_elements(b) eq 0 and arg_present(b) eq 1 then \$
> message, 'You passed an undefined variable as a keyword'
> if n_elements(b) gt 0 then b = 'passed'
> end

>
> Now,

>
> IDL> .reset
> IDL> test, b=junk
> % Compiled module: TEST.
> % TEST: You passed an undefined variable as a keyword

>
> Surely there must be something simpler than my 2-step approach ;)

Why the `arg_present`? Do you specifically want to tell the user that what they passed was undefined? I thought `arg_present` was for using when you wanted to return something in the variable? If the data is for input only, won't `n_elements()` suffice?

me confused.

paulv

--

Paul van Delst Religious and cultural
CIMSS @ NOAA/NCEP/EMC purity is a fundamentalist
Ph: (301)763-8000 x7274 fantasy
Fax:(301)763-8545 V.S.Naipaul
