Subject: About keywords and positional parameters Posted by fengliza on Thu, 07 Nov 2002 16:15:37 GMT

View Forum Message <> Reply to Message

Would you tell me how to distinguish the keywords and positional parameters in IDL procedures and functions?

Thanks in advance!

Subject: Re: About keywords and positional parameters Posted by David Fanning on Thu, 07 Nov 2002 16:57:54 GMT

View Forum Message <> Reply to Message

lily_zhang (fengliza@sina.com) writes:

- > Would you tell me how to distinguish the keywords and positional
- > parameters in IDL procedures and functions?

Positional parameters are those variables that come after the command name. They have a position after the name (1, 2, 3, etc.). Keyword parameters also come after the command name, but they have a KEYWORD=variable (or sometimes a /KEYWORD, which means KEYWORD=1) appearance. Keyword parameters can come in any order and don't affect the count of positional parameters.

You can't tell from looking whether a positional or keyword parameter is an input parameter or an output parameter (or both). You have to read the documentation to discover this.

Cheers,

David

--

David W. Fanning, Ph.D.

Fanning Software Consulting, Inc.

Phone: 970-221-0438, E-mail: david@dfanning.com

Coyote's Guide to IDL Programming: http://www.dfanning.com/

Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: About keywords and positional parameters Posted by R.G. Stockwell on Thu, 07 Nov 2002 18:26:08 GMT

View Forum Message <> Reply to Message

lily zhang wrote:

- Would you tell me how to distinguish the keywords and positional
 parameters in IDL procedures and functions?
 Thanks in advance!
- for instance:

function positionparam, position param, keyword = keyword

-bob stockwell

Subject: Re: About keywords and positional parameters Posted by JD Smith on Thu, 07 Nov 2002 20:37:03 GMT View Forum Message <> Reply to Message

On Thu, 07 Nov 2002 09:57:54 -0700, David Fanning wrote:

- > lily_zhang (fengliza@sina.com) writes:
- >> Would you tell me how to distinguish the keywords and positional
- >> parameters in IDL procedures and functions?
- > Positional parameters are those variables that come after the command
- > name. They have a position after the name (1, 2, 3, etc.). Keyword
- > parameters also come after the command name, but they have a
- > KEYWORD=variable (or sometimes a /KEYWORD, which means KEYWORD=1)
- > appearance. Keyword parameters can come in any order and don't affect
- > the count of positional parameters.
- > You can't tell from looking whether a positional or keyword parameter is
- > an input parameter or an output parameter (or both). You have to read
- > the documentation to discover this.

>

>

>

>

>

I agree that you can't tell if a parameter is *meant to be* an input parameter, an output parameter, or both without reading the documentation, but you certainly can tell whether it is *capable* of passing output:

if arg_present(parameter) then parameter=myungodlyfunction(foo)

ARG_PRESENT() may or many not reveal anything useful, depending on the context. Its primary legitimate use is in avoiding calculations of

JD

Subject: Re: About keywords and positional parameters Posted by David Fanning on Thu, 07 Nov 2002 21:15:18 GMT View Forum Message <> Reply to Message

JD Smith (jdsmith@as.arizona.edu) writes:

- > I agree that you can't tell if a parameter is *meant to be* an input
- > parameter, an output parameter, or both without reading the documentation,
- > but you certainly can tell whether it is *capable* of passing output:

>

> if arg_present(parameter) then parameter=myungodlyfunction(foo)

>

- > ARG_PRESENT() may or many not reveal anything useful, depending on the
- > context. Its primary legitimate use is in avoiding calculations of
- > optional keyword/argument outputs if the caller didn't ask for them.

I have a feeling this piece of helpful advice is more likely to confuse Mr. Zhang than enlighten him, but please don't get me started on the havoc caused by mis-named functions. Remember, if you want to know if a keyword is *used* or not in your program, you have to follow JD's helpful advice:

```
pro testme, KEY=k
if n_elements(k) ne 0 OR arg_present(k) then $
print,'You used KEY!' else $
print,'You neglected KEY!'
end
```

You can find all the nitty-gritty details here:

http://www.dfanning.com/tips/keyword_check.html

Cheers,

David

__

David W. Fanning, Ph.D. Fanning Software Consulting, Inc.

Phone: 970-221-0438, E-mail: david@dfanning.com

Coyote's Guide to IDL Programming: http://www.dfanning.com/

Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: About keywords and positional parameters Posted by JD Smith on Fri, 08 Nov 2002 19:45:28 GMT

View Forum Message <> Reply to Message

On Thu, 07 Nov 2002 14:15:18 -0700, David Fanning wrote:

```
> JD Smith (jdsmith@as.arizona.edu) writes:
>
>> I agree that you can't tell if a parameter is *meant to be* an input
>> parameter, an output parameter, or both without reading the
>> documentation, but you certainly can tell whether it is *capable* of
>> passing output:
>>
    if arg_present(parameter) then parameter=myungodlyfunction(foo)
>>
>> ARG PRESENT() may or many not reveal anything useful, depending on the
>> context. Its primary legitimate use is in avoiding calculations of
>> optional keyword/argument outputs if the caller didn't ask for them.
>
> I have a feeling this piece of helpful advice is more likely to confuse
> Mr. Zhang than enlighten him, but please don't get me started on the
> havoc caused by mis-named functions. Remember, if you want to know if a
> keyword is *used* or not in your program, you have to follow JD's
> helpful advice:
>
    pro testme, KEY=k
>
    if n elements(k) ne 0 OR arg_present(k) then $
>
      print, 'You used KEY!' else $
>
      print,'You neglected KEY!'
>
>
    end
>
```

Just to be clear, I've never actually used this method (using arg_present() essentially as an input test) in real-life code... it's not really all that helpful. In fact, I wrote it just to get under your skin, what with your braggadocious claim that you could never know if a keyword is *used*;). From my perspective, there are only three real cases worth keeping track of:

- 1. An input keyword: Use n_elements() to see if it's present, or keyword_set() if it's intended to be a boolean.
- An output keyword: Use arg_present() if computing the output is actually optional in your program, and you want to save the trouble of calculating it if the caller didn't ask for it.
- 3. An input/output keyword: Use a combination of the above two, reserving n_elements/keyword_set for input tests, and arg_present for output tests.

Subject: Re: About keywords and positional parameters Posted by David Fanning on Fri, 08 Nov 2002 20:32:19 GMT View Forum Message <> Reply to Message

JD Smith (jdsmith@as.arizona.edu) writes:

- > Just to be clear, I've never actually used this method (using
- > arg_present() essentially as an input test) in real-life code... it's
- > not really all that helpful. In fact, I wrote it just to get under
- > your skin, what with your braggadocious claim that you could never
- > know if a keyword is *used* ;). From my perspective, there are only
- > three real cases worth keeping track of:

>

- > 1. An input keyword: Use n_elements() to see if it's present, or
- > keyword_set() if it's intended to be a boolean.
- > 2. An output keyword: Use arg_present() if computing the output is
- > actually optional in your program, and you want to save the
- > trouble of calculating it if the caller didn't ask for it.
- > 3. An input/output keyword: Use a combination of the above two,
- > reserving n_elements/keyword_set for input tests, and arg_present
- > for output tests.

JD and I don't really need to be having this conversation, because both of us know perfectly well how to handle keywords correctly (and neither of us would probably ever have a need to know if a keyword was "used" or not). But for the sake of completeness, and for the edification of the 63.58% of IDL programmers (according to an informal survey I've taken) that use KEYWORD_SET incorrectly in their programs, I reiterate: KEYWORD_SET [capitals deliberate] should be used ONLY for keywords that have Boolean values. :-)

Cheers,

David

P.S. Let's just say that if everyone ripped KEYWORD_SET out of their code and substituted N_ELEMENTS the world would probably be a better place. (Or, at the very least, more functional.)

_-

David W. Fanning, Ph.D.

Fanning Software Consulting, Inc.

Phone: 970-221-0438, E-mail: david@dfanning.com

Coyote's Guide to IDL Programming: http://www.dfanning.com/

Toll-Free IDL Book Orders: 1-888-461-0155