
Subject: Re: Bug in N_PARAMS?

Posted by [David Fanning](#) on Mon, 24 Nov 2008 22:20:44 GMT

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

Kenneth P. Bowman writes:

- > I have been having some very odd problems with the N_PARAMS function lately.
- > ...
- > Am I missing something obvious here?

N_PARAMS only tells you *how many* parameters the command is called with. It tells you nothing about the *type* of the parameter. Since "undefined" is a valid variable type in IDL, it is possible to pass an "undefined" variable and have it counted as a passed *number* of variables by N_PARAMS.

If you are concerned with if a variable is undefined or not, you can't use N_PARAMS to determine this. The only way to tell is to use N_ELEMENTS.

Cheers,

David

--

David Fanning, Ph.D.

Coyote's Guide to IDL Programming (www.dfanning.com)

Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Bug in N_PARAMS?

Posted by [Kenneth P. Bowman](#) on Tue, 25 Nov 2008 14:29:28 GMT

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

In article <MPG.2394d6a934f41609989775@news.giganews.com>, David Fanning <news@dfanning.com> wrote:

- > Kenneth P. Bowman writes:
- >
- >> I have been having some very odd problems with the N_PARAMS function lately.
- >> ...
- >> Am I missing something obvious here?
- >
- > N_PARAMS only tells you *how many* parameters the
- > command is called with. It tells you nothing about
- > the *type* of the parameter. Since "undefined" is
- > a valid variable type in IDL, it is possible to
- > pass an "undefined" variable and have it counted

> as a passed *number* of variables by N_PARAMS.
>
> If you are concerned with if a variable is undefined
> or not, you can't use N_PARAMS to determine this. The
> only way to tell is to use N_ELEMENTS.

Ah. That significantly restricts the utility of

```
SWITCH N_PARAMS() OF  
  0 : ...
```

```
ENDSWITCH
```

to provide default values for optional parameters. I suppose that is the only way it could work with positional parameters, as you might have an undefined parameter in the middle of the list.

Ken

Subject: Re: Bug in N_PARAMS?
Posted by [Carsten Lechte](#) on Wed, 26 Nov 2008 07:53:01 GMT
[View Forum Message](#) <> [Reply to Message](#)

Kenneth P. Bowman wrote:

> to provide default values for optional parameters.

I wrote this little procedure for setting default values that I use a zillion times a day:

```
PRO defaultarg, arg, defval, VALUEGIVEN=valuegiven
```

```
  IF N_ELEMENTS( arg) EQ 0 THEN BEGIN  
    arg = defval  
    valuegiven = 0  
  ENDIF ELSE valuegiven = 1
```

```
END
```

then use it like this:

```
PRO somepro, arg1, arg2, KEY1=key1
```

```
  defaultarg, arg1, 0  
  defaultarg, arg2, 'test'  
  defaultarg, key1, 2343d8
```

...

END

chl

Subject: Re: Bug in N_PARAMS?

Posted by [David Fanning](#) on Wed, 26 Nov 2008 15:49:33 GMT

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

Carsten Lechte writes:

```
> Kenneth P. Bowman wrote:
>> to provide default values for optional parameters.
>
> I wrote this little procedure for setting default values
> that I use a zillion times a day:
>
>
> PRO defaultarg, arg, defval, VALUEGIVEN=valuegiven
>
>   IF N_ELEMENTS( arg) EQ 0 THEN BEGIN
>     arg = defval
>     valuegiven = 0
>   ENDIF ELSE valuegiven = 1
>
>
> END
```

This is a pretty good idea. I can see how it would save some typing! :-)

Here is a slightly modified version. I'll add this, with documentation, to the Coyote Library later today.

```
PRO SetDefaultValue, argument, defaultValue, BOOLEAN=boolean
```

```
; If BOOLEAN, we need just a 0 or a 1.
IF Keyword_Set(boolean) THEN BEGIN
```

```
    argument = Keyword_Set(argument)
```

```
ENDIF ELSE BEGIN
```

```
; We only need assignment if the argument is undefined.
```

```
IF N_Elements(argument) EQ 0 THEN BEGIN

    ; If the default value is undefined, treat as BOOLEAN.
    ; Otherwise, assign default value to the argument.
    IF N_Elements(defaultValue) EQ 0 THEN BEGIN
        argument = Keyword_Set(argument)
    ENDIF ELSE BEGIN
        argument = defaultValue
    ENDELSE

ENDIF

ENDELSE
```

END

Cheers,

David

--

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

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

Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Bug in N_PARAMS?

Posted by [Mark\[1\]](#) on Thu, 27 Nov 2008 21:53:55 GMT

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

On Nov 26, 3:29 am, "Kenneth P. Bowman" <k-bow...@null.edu> wrote:

> Ah. That significantly restricts the utility of

>

> SWITCH N_PARAMS() OF

> 0: ...

>

> ENDSWITCH

>

> to provide default values for optional parameters.

Indeed. N_ELEMENTS is what you need for that. The only use I have found for N_PARAMS is for writing wrapper routines

```
pro wrapper, P1, P2, P3, P4, _REF_EXTRA=_extra
```

```
;; Do some other stuff
```

```
case n_params() of
```

```
0: wrappee, _STRICT_EXTRA=_extra
```

```
1: wrappee, P1, _STRICT_EXTRA=_extra
```

```
2: wrappee, P1, P2, _STRICT_EXTRA=_extra
3: wrappee, P1, P2, P3, _STRICT_EXTRA=_extra
4: wrappee, P1, P2, P3, P4, _STRICT_EXTRA=_extra
endcase
end
```

Ugly, but sometimes necessary.

Subject: Re: Bug in N_PARAMS?
Posted by [Brian Larsen](#) on Tue, 02 Dec 2008 15:03:24 GMT
[View Forum Message](#) <> [Reply to Message](#)

David, (and others)

this is certainly one of the best things about this newsgroup community. A question gets asked about a bug and there is some helpful discussion then one of our resident experts (who is now a java programmer?) puts up a bit of code that is so obvious that it is incredibly useful. So many of my codes have

```
if n_elements(var) eq 0 then $
    var = 245 ; or some other value
```

that repeat over and over and over, thanks again for providing the clean solution.

Cheers,

Brian

Brian Larsen
Boston University
Center for Space Physics
<http://people.bu.edu/balarsen/Home/IDL>

Subject: Re: Bug in N_PARAMS?
Posted by [David Fanning](#) on Wed, 03 Dec 2008 17:06:06 GMT
[View Forum Message](#) <> [Reply to Message](#)

David Fanning writes:

```
> Carsten Lechte writes:
>
```

```
>> Kenneth P. Bowman wrote:
>>> to provide default values for optional parameters.
>>
>> I wrote this little procedure for setting default values
>> that I use a zillion times a day:
>>
>>
>> PRO defaultarg, arg, defval, VALUEGIVEN=valuegiven
>>
>>   IF N_ELEMENTS( arg) EQ 0 THEN BEGIN
>>     arg = defval
>>     valuegiven = 0
>>   ENDIF ELSE valuegiven = 1
>>
>> END
>
> This is a pretty good idea. I can see how it would
> save some typing! :-)
>
> Here is a slightly modified version. I'll add this,
> with documentation, to the Coyote Library later today.
```

I have a doppelganger at work who takes all my good ideas as soon as I publish them and points out all their deficiencies and the obvious ways in which I could have made them better. :-(

To that end, I've updated the SetDefaultValue program I published earlier to something a good deal simpler that also works in a more intuitive way. You can find the update, if you are interested in this sort of thing, here:

<http://www.dfanning.com/programs/setdefaultvalue.pro>

Cheers,

David

--

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

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

Sepore ma de ni thui. ("Perhaps thou speakest truth.")
