Subject: Bug in N_PARAMS?
Posted by Kenneth P. Bowman on Mon, 24 Nov 2008 19:31:37 GMT

View Forum Message <> Reply to Message

I have been having some very odd problems with the N_PARAMS function lately.

Here is a simple example with a main program and a function.

```
PRO N_PARAMS_MAIN, a, b
COMPILE_OPT IDL2
PRINT, 'N_PARAMS_MAIN - Number of parameters : ', N_PARAMS()
HELP, a, b
value = N_PARAMS_BUG(a, b)
END


FUNCTION N_PARAMS_BUG, a, b
COMPILE_OPT IDL2
PRINT, 'N_PARAMS_BUG  - Number of parameters : ', N_PARAMS()
HELP, a, b
RETURN, 0
END
```

Here is the output from a new IDL session:

```
stretch> idl
IDL Version 6.4.1, Mac OS X (darwin i386 m32). (c) 2007, ITT Visual Information Solutions

IDL> N_PARAMS_MAIN, 13
% Compiled module: N_PARAMS_MAIN.
N_PARAMS_MAIN - Number of parameters :           1
A               LONG      =           13
B               UNDEFINED = <Undefined>
% Compiled module: N_PARAMS_BUG.
N_PARAMS_BUG  - Number of parameters :           2     <-- returns 2, but b is undefined
A               LONG      =           13
B               UNDEFINED = <Undefined>
```

In the main program N_PARAMS_MAIN, N_PARAMS correctly returns the number of defined arguments (one).

In the function N_PARAMS_BUG, the argument count is incorrectly given as two.

If I call the function N_PARAMS_BUG from the command line, it works correctly.

```
IDL> PRINT, N_PARAMS_BUG(10)
N_PARAMS_BUG  - Number of parameters :           1
A               LONG      =           10
B               UNDEFINED = <Undefined>
           0
```

Am I missing something obvious here?

Ken

Subject: Re: Bug in N_PARAMS?
Posted by R.Bauer on Fri, 28 Nov 2008 10:02:33 GMT

View Forum Message <> Reply to Message

The usecase for me is to call a helper routine if not enough parameters were used.

e.g.

```
 if n_params() lt 2 then begin
      MESSAGE,call_help(),/cont
      return
 endif
```

The user get's shown the CALLING SEQUENCE description

cheers
Reimar

Mark schrieb:
> 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 Mark[1] on Sun, 30 Nov 2008 22:06:51 GMT

View Forum Message <> Reply to Message

On Nov 28, 11:02 pm, Reimar Bauer <R.Ba...@fz-juelich.de> wrote:

```
>  The usecase for me is to call a helper routine if not enough parameters
>  were used.
>
>  e.g.
>
>   if n_params() lt 2 then begin
>        MESSAGE,call_help(),/cont
>        return
>   endif
>
>
```

Sure, but even if 2 parameters were supplied, you still don't know if the variables they're associated with are defined. If you're going to check for that anyway, with N_ELEMENTS, the N_PARAMS check is redundant.

Eg.

```
pro mypro, a, b

if n_elements(a) eq 0 || n_elements(b) eq 0 then begin
       MESSAGE,call_help(),/cont
       return
endif

end
```

Subject: Re: Bug in N_PARAMS?
Posted by Jeremy Bailin on Mon, 01 Dec 2008 14:11:07 GMT
View Forum Message <> Reply to Message

On Nov 30, 5:06 pm, Mark <mark.h...@gmail.com> wrote:
> On Nov 28, 11:02 pm, Reimar Bauer <R.Ba...@fz-juelich.de> wrote:
>
>> The usecase for me is to call a helper routine if not enough parameters
>> were used.
>
>> e.g.
>
>> if n_params() lt 2 then begin
>>   MESSAGE,call_help(),/cont
>>   return
>> endif
>
> Sure, but even if 2 parameters were supplied, you still don't know if
> the variables they're associated with are defined. If you're going to
> check for that anyway, with N_ELEMENTS, the N_PARAMS check is
> redundant.
>
> Eg.
>
> pro mypro, a, b
>
> if n_elements(a) eq 0 || n_elements(b) eq 0 then begin
>   MESSAGE,call_help(),/cont
>   return
> endif
>
> end

Except that there are valid reasons for passing in an undefined
variable, if it's used for output. So it's perfectly legitimate to
want to know whether it was passed in without caring whether it is
defined.

-Jeremy.

Subject: Re: Bug in N_PARAMS?
Posted by Mark[1] on Mon, 01 Dec 2008 22:55:34 GMT
View Forum Message <> Reply to Message

On Dec 2, 3:11 am, Jeremy Bailin <astroco...@gmail.com> wrote:
> Except that there are valid reasons for passing in
> an undefined variable, if it's used for output. So

> it's perfectly legitimate to want to know whether
> it was passed in without caring whether it is
> defined.

To check that an argument has been passed, and can be written to, you can use ARG_PRESENT.

To check that a argument has been passed, and is a defined variable, you can use N_ELEMENTS > 0.

This leaves N_PARAMS for the cases where you want to know if an argument has been passed, but otherwise don't care about its status. In my experience. this is pretty rare, except in what I would call wrapper routines.

Subject: Re: Bug in N_PARAMS?
Posted by David Fanning on Tue, 02 Dec 2008 15:10:52 GMT

View Forum Message <> Reply to Message

Brian Larsen writes:

> 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, thanks again for providing the
> clean solution.

The obvious solutions are the most difficult. It takes
real talent to come up with them. Especially for people
who have a Ph.D. at the end of their name. ;-)

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 R.Bauer on Wed, 03 Dec 2008 17:59:59 GMT
View Forum Message <> Reply to Message

David Fanning schrieb:
> 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

hehe

you need a german keyboard or is this also an english word
"doppelganger" <-> doppelgï¿½nger?

cheers
Reimar

> 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

Subject: Re: Bug in N_PARAMS?
Posted by David Fanning on Wed, 03 Dec 2008 18:49:58 GMT

View Forum Message <> Reply to Message

Reimar Bauer writes:

> you need a german keyboard or is this also an english word

I need a German keyboard AND this is an English
word. Well, it is in an English dictionary, but
obviously stolen from the Germans. :-)

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 pgrigis on Wed, 03 Dec 2008 19:15:45 GMT

View Forum Message <> Reply to Message

David, you don't really need all these keyboards,
it's enough to play a bit with the keyboard settings
and select other languages (this should be possible
on our triad of IDL OSes).

öäü

All written with an american keyboard!
Of course the hard part sometimes may be
finding out the meaning of what you just wrote ;-)

Ciao,
Paolo

David Fanning wrote:
> Reimar Bauer writes:
>
>> you need a german keyboard or is this also an english word

>
> I need a German keyboard AND this is an English
> word. Well, it is in an English dictionary, but
> obviously stolen from the Germans. :-)
>
> Cheers,
>
> David
>
> --
> David Fanning, Ph.D.
> Coyote's Guide to IDL Programming (www.dfanning.com)
> Sepore ma de ni thui. ("Perhaps thou speakest truth.")