Subject: Re: Passing zero as a Parameter/ NOT KEYWORD_SET
Posted by Grady Daub on Mon, 28 Jun 1999 07:00:00 GMT
View Forum Message <> Reply to Message

Will n_elements give a number <= zero if the keyword is called like
"/keyword"

Kinda looks like you can't have a parameter = 0 AND be able to check if
it is set or not.
None of that fancy

on_of_off=num
;where num is 1 for set and 0 for unset.
procedure_name,keyword=on_or_off

:-(

"Martin LUETHI GL A8.1 2-4092" wrote:

> You have two possibilites:
>
> o if param_present(keyword) then ; working in PV-Wave at least
> o if (n_elements(keyword) gt 0) then ; better

Subject: Re: Passing zero as a Parameter/ NOT KEYWORD_SET
Posted by Grady Daub on Mon, 28 Jun 1999 07:00:00 GMT
View Forum Message <> Reply to Message

Ah, I remember doing that. I think I took it out while looking for another
bug, thinking that it didn't work. :-(

Thanks.

> However the best way is just to do this:
>
> IF KEYWORD_SET(keyword) EQ 0 THEN BEGIN
>

Subject: Re: Passing zero as a Parameter/ NOT KEYWORD_SET
Posted by philaldis on Mon, 28 Jun 1999 07:00:00 GMT
View Forum Message <> Reply to Message

On Mon, 28 Jun 1999 09:08:04 -0400, Grady Daub
<gadZOOKS8371@garnet.acns.fsuMMER.edu> wrote:

> How can I pass the number zero as a parameter, without the
> function/subroutine thinking that the parameter is not set?
>
snipped
>
> IF NOT KEYWORD_SET(keyword) then do-this-stuff
>
> not work?
>
> I've had to do the following:
>
> IF KEYWORD_SET(keyword) then a=a else do-this-stuff
>
> ("a=a" acts as some kind of dummy variable thing.)
>
>
> -Grady Daub
>
> (Remove MMER and ZOOKS to reply by e-mail.)
>
>

I presume that in your code your doing an

```
IF Keyword_Set() THEN BEGIN
 .
 .
        define out
 .
 .
ENDIF
```

but Keyword_Set() is more suitable for flags, than parameters. The
best way to do it is use N_Elements(), which returns 0 if a variable
is undefined. So do :

```
IF N_Elements(in) THEN BEGIN
 .
 .
 .
ENDIF
```

NOT when applied to an integer does funny twos complement or something
(I'm not treally sure so look at the IDL help if you're interested).
When NOT operates on floats, however, it just does the logical

inverse. So you could do:

IF NOT(Float(KEYWORD_SET(keyword))) THEN BEGIN

However the best way is just to do this:

IF KEYWORD_SET(keyword) EQ 0 THEN BEGIN

Remeber that KEYWORD_SET is a function and it returns 0 if the keyword
isn't set and 1 if it is. It's easy to get confused looking at the
syntax of IF Keyword_Set(keyword), and think that it is actually
asking if the keyword is set. In fact, IF KEYWORD_SET(keyword) is
merely shorthand for:

 IF KEYWORD_SET(keyword) EQ 1

I hope this helps

Cheers,
Phil

Subject: Re: Passing zero as a Parameter/ NOT KEYWORD_SET
Posted by luthi on Mon, 28 Jun 1999 07:00:00 GMT
View Forum Message <> Reply to Message

Grady

You have two possibilites:

o if param_present(keyword) then   ; working in PV-Wave at least
o if (n_elements(keyword) gt 0) then ; better

Cheers

Martin

--
 ==
Martin Luethi   Tel. +41 1 632 40 92
Glaciology Section Fax. +41 1 632 11 92
VAW ETH Zuerich
CH-8092 Zuerich   mail luthi@vaw.baum.ethz.ch
Switzerland
 ==

Subject: Re: Passing zero as a Parameter/ NOT KEYWORD_SET
Posted by Pavel Romashkin on Tue, 29 Jun 1999 07:00:00 GMT
View Forum Message <> Reply to Message

It is obvious that most IDL users are scientists with experience.
Otherwise, how one practical question on "how to do..." could result in
this much hypothesyzing and discussion? Not only do we want to have the
job done (and not so much we want it done, I guess), but to figure out
what's right and who's wrong. Only PhDs can do that...

Grady Daub wrote:

> How can I pass the number zero as a parameter, without the
> function/subroutine thinking that the parameter is not set?

Subject: Re: Passing zero as a Parameter/ NOT KEYWORD_SET
Posted by davidf on Tue, 29 Jun 1999 07:00:00 GMT
View Forum Message <> Reply to Message

J.D. Smith (jdsmith@astrosun.tn.cornell.edu) writes:

> I just took a look at your article again, and I fear I must take issue
> with the very first statement:
>
> "Despite what your colleague may have told you, or what you believe you
> read in the IDL documentation, it is NOT possible to reliably determine
> if a keyword
> was used in a call to your program."
>
> I disagree. Though I would never do this myself, you can easily get
> this behaviour, vz.
>
> 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

Oh, well, of course I meant "impossible with the tools RSI
gives you, but not impossible if you write your own variations
using arcane knowledge of how the tools RSI gives you *really*
work, despite their names". I'm annoyed with myself for having
settled for the shorter paragraph and been found out. :-)

> For it must be either that k is undefined by virtue of not being passed
> at all, or by virtue of being an as-yet undefined (and therefore
> by-reference) variable passed in from above. The former case we can

> detect with n_elements(), the latter case with arg_present(). I
> encourage you to try to find an example of something which is both
> undefined, and also passed by value. This would be the only thing which
> could escape detection in the above algorithm.

Well, this is not such a stretch as you might imagine. I remember
a very strange problem with compound widgets. Let's see, I think
if use Set_Value with the value of 1...Oh, I can't remember exactly
now. But I do remember is was caused by this very thing: setting a
value to a scalar constant when no one in the world would think to
do it except the students in my classes who naively did what I told
them to do. :-(

> When I first urged RSI to give us arg_present(), I had in mind exactly
> the type of application you mention at the end of your page. Your
> strenuous warning might disincline readers from its use, but it really
> has proven invaluable.

Indeed, that is *exactly* what it is meant to be used for. I just
wish they hadn't given it such an unfortunate name. But I would
certainly use it for that purpose myself.

> I *don't* recommend using the above method, since you end up with all
> sorts of undefined variables which are silently created and unused.
> This is not likely what the user expects. It does open the possibility
> for using keyword arguments as strings without being strings, e.g.
>
> mypro, KEY=TRUE
>
> but this is rather silly, I'd say, when you can just as easily use
> /KEY. But since you made such a strong point about it being impossible,
> I couldn't keep it to myself :).

Any article from you, JD, is *always* much appreciated. :-)

> The rest of the page makes good sense though. I hope this doesn't
> qualify as bugging you day and night ;)

I'd make the appropriate change to the article, but I'm afraid
you and I (and perhaps Martin after he has a chance to cogitate
a bit in this newsgroup) will be the only ones to appreciate
the truth of the remark. :-)

Cheers,

David

--

David Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438 E-Mail: davidf@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: Passing zero as a Parameter/ NOT KEYWORD_SET
Posted by J.D. Smith on Tue, 29 Jun 1999 07:00:00 GMT
View Forum Message <> Reply to Message

David Fanning wrote:
> Most people use Keyword_Set for entirely the wrong thing.
> Keyword_Set very clearly returns 0 if the argument is 0 or
> undefined. It returns a 1 if the argument is *anything*
> other than 0 or undefined. It should be used ONLY with
> those keywords that require a Boolean value.
>
> But I welcome your comments on the article. (Just don't
> bug me night and day until I incorporate your comments. :-)
>
> http://www.dfanning.com/tips/keyword_check.html
>

I just took a look at your article again, and I fear I must take issue
with the very first statement:

"Despite what your colleague may have told you, or what you believe you
read in the IDL documentation, it is NOT possible to reliably determine
if a keyword
was used in a call to your program."

I disagree. Though I would never do this myself, you can easily get
this behaviour, vz.

```
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
```

For it must be either that k is undefined by virtue of not being passed
at all, or by virtue of being an as-yet undefined (and therefore
by-reference) variable passed in from above. The former case we can
detect with n_elements(), the latter case with arg_present(). I
encourage you to try to find an example of something which is both
undefined, and also passed by value. This would be the only thing which
could escape detection in the above algorithm.

When I first urged RSI to give us arg_present(), I had in mind exactly the type of application you mention at the end of your page. Your strenuous warning might disincline readers from its use, but it really has proven invaluable.

I *don't* recommend using the above method, since you end up with all sorts of undefined variables which are silently created and unused. This is not likely what the user expects. It does open the possibility for using keyword arguments as strings without being strings, e.g.

mypro, KEY=TRUE

but this is rather silly, I'd say, when you can just as easily use /KEY. But since you made such a strong point about it being impossible, I couldn't keep it to myself :).

The rest of the page makes good sense though. I hope this doesn't qualify as bugging you day and night ;)

JD

--
J.D. Smith |*| WORK: (607) 255-5842
Cornell University Dept. of Astronomy |*| (607) 255-6263
304 Space Sciences Bldg. |*| FAX: (607) 255-5875
Ithaca, NY 14853 |*|

Subject: Re: Passing zero as a Parameter/ NOT KEYWORD_SET
Posted by J.D. Smith on Tue, 29 Jun 1999 07:00:00 GMT
View Forum Message <> Reply to Message

Martin Schultz wrote:
>
> J.D. Smith wrote:
>>
>>
>> That's a bit dangerous. [...]
> Indeed ;-)
>>
>> The best way to proceed is pretend keyword_set() was really
>> named is_defined_and_non_zero(). Forget that it's called
>> keyword_set().
> In fact it is "is_defined_and_uneven" !
> Just try to pass var=2 into a routine and print keyword_set(var). Hope,
> David will take notice of this in his article.

I get 1 printed if var is anything but zero or undefined. This sounds
like a bug on your version/system, if it's really occurring. So I'll
stick with is_defined_and_non_zero(): a bit of a mouthful, but much less
misleading.

> Another marginal point about setting default values: I recently learned
> from someone's code (cannot remember whose), to use
>   if (n_elements(var) ne 1) then var=default
> instead of
>   if (n_elements(var) eq 0) then var=default
>
> The advantage being that you can prevent program crashes when someone
> passes a vector or array in what is supposed to be a scalar.
>

Unless, of course, you *are* interested in a vector or array. You're
mixing two types of checks here: "is the argument there at all?", vs.
"is the argument of the type I want?"... (n_elements(var) eq 0) will
work for any type of expected variable; (n_elements(var) ne 1) only
works if you expect a scalar. You are of course free to do this type of
mixing where appropriate (though you don't have to), but for the benefit
of those just learning, I thought I should try to make this clear.

> And, finally: Use keyword_set when you want to make sure the value of a
> boolean flag is defined:
>   flag = keyword_set(flag)
> Then, later in the code, it's just
>   if (flag) then ...
> Or value = x+y*(flag), etc. which would crash otherwise.

Yes, another good use of keyword _set().

JD

--
J.D. Smith |*| WORK: (607) 255-5842
Cornell University Dept. of Astronomy |*| (607) 255-6263
304 Space Sciences Bldg. |*| FAX: (607) 255-5875
Ithaca, NY 14853 |*|

Subject: Re: Passing zero as a Parameter/ NOT KEYWORD_SET
Posted by davidf on Tue, 29 Jun 1999 07:00:00 GMT

View Forum Message <> Reply to Message

Martin Schultz (mgs@io.harvard.edu) writes:

>> The best way to proceed is pretend keyword_set() was really

>> named is_defined_and_non_zero(). Forget that it's called
>> keyword_set().

> In fact it is "is_defined_and_uneven" !
> Just try to pass var=2 into a routine and print keyword_set(var). Hope,
> David will take notice of this in his article.

Most people use Keyword_Set for entirely the wrong thing.
Keyword_Set very clearly returns 0 if the argument is 0 or
undefined. It returns a 1 if the argument is *anything*
other than 0 or undefined. It should be used ONLY with
those keywords that require a Boolean value.

But I welcome your comments on the article. (Just don't
bug me night and day until I incorporate your comments. :-)

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

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438 E-Mail: davidf@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: Passing zero as a Parameter/ NOT KEYWORD_SET
Posted by Martin Schultz on Tue, 29 Jun 1999 07:00:00 GMT
View Forum Message <> Reply to Message

J.D. Smith wrote:
>
>
> That's a bit dangerous. [...]
Indeed ;-)
>
> The best way to proceed is pretend keyword_set() was really
> named is_defined_and_non_zero(). Forget that it's called
> keyword_set().
In fact it is "is_defined_and_uneven" !
Just try to pass var=2 into a routine and print keyword_set(var). Hope,
David will take notice of this in his article.

Another marginal point about setting default values: I recently learned
from someone's code (cannot remember whose), to use

```
   if (n_elements(var) ne 1) then var=default
instead of
   if (n_elements(var) eq 0) then var=default
```

The advantage being that you can prevent program crashes when someone passes a vector or array in what is supposed to be a scalar.

And, finally: Use keyword_set when you want to make sure the value of a boolean flag is defined:

```
    flag = keyword_set(flag)
```

Then, later in the code, it's just

```
    if (flag) then ...
```

Or value = x+y*(flag), etc. which would crash otherwise.

Regards,
Martin.

 ||||||||||||||\\\\\\\\\\\\\\----------////////////// //||||||||||||||
Martin Schultz, DEAS, Harvard University, 29 Oxford St., Pierce 109,
Cambridge, MA 02138 phone (617) 496 8318 fax (617) 495 4551
e-mail mgs@io.harvard.edu web http://www-as/people/staff/mgs/

Subject: Re: Passing zero as a Parameter/ NOT KEYWORD_SET
Posted by R.Bauer on Tue, 29 Jun 1999 07:00:00 GMT

View Forum Message <> Reply to Message

```
<!doctype html public "-//w3c//dtd html 4.0 transitional//en">
<html>
<head>
   <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
   <meta name="description" content="1.7 ï¿½bungsaufgaben">
   <meta name="keywords" content="idl_kurs_doc">
   <meta name="resource-type" content="document">
   <meta name="distribution" content="global">
   <meta name="GENERATOR" content="Mozilla/4.5 [en] (WinNT; I) [Netscape]">
   <title>FZJ--BHB-007: 1.7 &Uuml;bungsaufgaben</title>
<!--Converted with LaTeX2HTML 97.1 (release) (July 13th, 1997)
 by Nikos Drakos (nikos@cbl.leeds.ac.uk), CBLU, University of Leeds
* revised and updated by:  Marcus Hennecke, Ross Moore, Herb Swan
* with significant contributions from:
  Jens Lippman, Marek Rouchal, Martin Wilck and others -->
<link REL="STYLESHEET" HREF="idl_kurs_doc.css">
<link REL="previous" HREF="node30.html">
<link REL="up" HREF="node2.html">
<link REL="next" HREF="node33.html">
</head>
<body bgcolor="#FFFFFF">
```


<h1>
<a NAME="SECTION00270000000000000000">1.7 TRAINING</h1>

Look carefully at the given functions:

<table BORDER WIDTH="80%" BGCOLOR="#FFFFAA" >
<tr>
<td>
<pre>FUNCTION test1,minimum=min_val

    IF KEYWORD_SET(min_val) THEN RETURN,1 ELSE RETURN,0

  END

  </pre>
</td>
</tr>

<caption ALIGN=BOTTOM>Example 1.21: test1</caption>
</table>

<table BORDER WIDTH="80%" BGCOLOR="#FFFFAA" >
<tr>
<td>
<pre>FUNCTION test2,minimum=min_val

    IF ARG_PRESENT(min_val) THEN RETURN,1 ELSE RETURN,0

  END

  </pre>
</td>
</tr>

<caption ALIGN=BOTTOM>Example 1.22: test2</caption>
</table>

<table BORDER WIDTH="80%" BGCOLOR="#FFFFAA" >
<tr>
<td>
<pre>FUNCTION test3,minimum=min_val

    IF N_ELEMENTS(min_val) GT 0 THEN RETURN,1 ELSE RETURN,0

  END

  </pre>
</td>
</tr>

<caption ALIGN=BOTTOM>Example 1.23: test3</caption>
</table>

<p>
Your results:

<table BORDER CELLPADDING=3 BGCOLOR="#FFFFAA" >
<tr VALIGN=TOP>
<td ALIGN=LEFT NOWRAP>call:</td>

<td ALIGN=LEFT NOWRAP>test1</td>

<td ALIGN=LEFT NOWRAP>test2</td>

<td ALIGN=LEFT NOWRAP>test3</td>
</tr>

<tr VALIGN=TOP>
<td ALIGN=LEFT NOWRAP><tt>PRINT, testX( )</tt></td>

<td ALIGN=LEFT NOWRAP> </td>

<td ALIGN=LEFT NOWRAP> </td>

<td ALIGN=LEFT NOWRAP> </td>
</tr>

<tr VALIGN=TOP>
<td ALIGN=LEFT NOWRAP><tt>PRINT, testX(minimum=0)</tt></td>

<td ALIGN=LEFT NOWRAP> </td>

<td ALIGN=LEFT NOWRAP> </td>

<td ALIGN=LEFT NOWRAP> </td>
</tr>

<tr VALIGN=TOP>
<td ALIGN=LEFT NOWRAP><tt>PRINT, testX(minimum=10)</tt></td>

<td ALIGN=LEFT NOWRAP> </td>

<td ALIGN=LEFT NOWRAP> </td>

<td ALIGN=LEFT NOWRAP> </td>
</tr>

<tr VALIGN=TOP>
<td ALIGN=LEFT NOWRAP><tt>PRINT, testX(minimum=-10)</tt></td>

<td ALIGN=LEFT NOWRAP> </td>

<td ALIGN=LEFT NOWRAP> </td>

<td ALIGN=LEFT NOWRAP> </td>
</tr>

<tr VALIGN=TOP>
<td ALIGN=LEFT NOWRAP><tt>mv=0 & PRINT, testX(minimum=mv)</tt></td>

<td ALIGN=LEFT NOWRAP> </td>

<td ALIGN=LEFT NOWRAP> </td>

<td ALIGN=LEFT NOWRAP> </td>
</tr>

<tr VALIGN=TOP>
<td ALIGN=LEFT NOWRAP><tt>mv=10 & PRINT, testX(minimum=mv)</tt></td>

<td ALIGN=LEFT NOWRAP> </td>

<td ALIGN=LEFT NOWRAP> </td>

<td ALIGN=LEFT NOWRAP> </td>
</tr>

<tr>
<td><tt>PRINT, testX(minimum=mv)</tt></td>

<td> </td>

<td> </td>

<td> </td>
</tr>
</table>

<address>
</address>

</body>
</html>

File Attachments

1) node32.html, downloaded 209 times

Subject: Re: Passing zero as a Parameter/ NOT KEYWORD_SET
Posted by davidf on Tue, 29 Jun 1999 07:00:00 GMT
View Forum Message <> Reply to Message

J.D. Smith (jdsmith@astrosun.tn.cornell.edu) writes:

> That's a bit dangerous. IDL may have strange semantics and constructs,
> but that's no reason not to learn them. I recommend peeking at a bit of
> IDL source code available all over the web. You'll find common
> constructs such as:
>
> ; give foo a default value if it is undefined.
> if n_elements(foo) eq 0 then foo=5
> ; perform some action if flag is defined and non-zero
> if keyword_set(flag) then compute_something
> ; return something to the caller if var is available to them (by
> reference)
> if arg_present(var) then var=compute_something_else()

Ah, finally, some sense on this topic! Thanks, JD.

I lost my news feed yesterday and the article I posted on this topic got eaten by DejaNews (probably fortunately). But nothing causes more confusion for users than these three totally misnamed functions: N_Elements, Keyword_Set, and Arg_Present.

I'm in the process of writing an article on the subject for my web page right this minute. (Why I haven't done it before is a mystery. It's the single most misunderstood programming feature I see in my travels.) It will be on my site sometime this afternoon (Colorado time).

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

Cheers,

David
--

David Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438 E-Mail: davidf@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: Passing zero as a Parameter/ NOT KEYWORD_SET
Posted by J.D. Smith on Tue, 29 Jun 1999 07:00:00 GMT
View Forum Message <> Reply to Message

Andy Sowter wrote:
>
> Hmmm! These solutions are all very elegant, aren't they? Gives me a
> headache, though! There's something in my IDL manual about avoiding IF
> statements.....
>
> For the less accomplished among us, and I'm included, if the parameter
> you're trying to pass is an unsigned integer, add 1 to it before you pass it
> to the subroutine and then subtract 1 when you get in. Easy. And you
> didn't have to spend ages with the manual either (actually, there's 8
> manuals with my IDL and it's a nightmare to find things - it was better when
> there were only 2!) :)
>

That's a bit dangerous. IDL may have strange semantics and constructs, but that's no reason not to learn them. I recommend peeking at a bit of IDL source code available all over the web. You'll find common constructs such as:

```
; give foo a default value if it is undefined.
if n_elements(foo) eq 0 then foo=5
; perform some action if flag is defined and non-zero
if keyword_set(flag) then compute_something
; return something to the caller if var is available to them (by
reference)
if arg_present(var) then var=compute_something_else()
```

The best way to proceed is pretend keyword_set() was really named is_defined_and_non_zero(). Forget that it's called keyword_set(). It really has nothing to do with keywords necessarily, and RSI has managed to confound many new users with this unfortunate appellation. I use it in many other places. It's good if the caller may want to explicitly turn *off* an option by passing a value, as with a flag. Here's a simple guide to the use of the keyword_set vs. n_elements methods.

1. IDL> foo, /FLAG \ same results for "n_elements(flag) ne 0"

2. IDL> foo, FLAG=0 / \
3. IDL> foo / same results for "keyword_set(flag)"

And using an optional argument instead of keyword:

1. IDL> foo, 1 \ same results for "n_elements(arg) ne 0"
2. IDL> foo, 0 / \
3. IDL> foo / same results for "keyword_set(arg)"

Once you look past the naming and see what these functions do, you can easily set up any kind of argument/keyword processing you might want.

Good Luck,

JD

--
J.D. Smith |*| WORK: (607) 255-5842
Cornell University Dept. of Astronomy |*| (607) 255-6263
304 Space Sciences Bldg. |*| FAX: (607) 255-5875
Ithaca, NY 14853 |*|

Subject: Re: Passing zero as a Parameter/ NOT KEYWORD_SET
Posted by Andy Sowter on Tue, 29 Jun 1999 07:00:00 GMT
View Forum Message <> Reply to Message

Hmmm! These solutions are all very elegant, aren't they? Gives me a headache, though! There's something in my IDL manual about avoiding IF statements.....

For the less accomplished among us, and I'm included, if the parameter you're trying to pass is an unsigned integer, add 1 to it before you pass it to the subroutine and then subtract 1 when you get in. Easy. And you didn't have to spend ages with the manual either (actually, there's 8 manuals with my IDL and it's a nightmare to find things - it was better when there were only 2!) :)

Andy
Brute Force School of Programming

Grady Daub wrote in message <37777E20.353D94CA@garnet.acns.fsuMMER.edu>...
>
> Ah, I remember doing that. I think I took it out while looking for another
> bug, thinking that it didn't work. :-(
>
> Thanks.

>
>> However the best way is just to do this:
>>
>> IF KEYWORD_SET(keyword) EQ 0 THEN BEGIN
>>
>
>

Subject: Re: Passing zero as a Parameter/ NOT KEYWORD_SET
Posted by philaldis on Tue, 29 Jun 1999 07:00:00 GMT
View Forum Message <> Reply to Message

On Mon, 28 Jun 1999 10:02:49 -0400, Grady Daub
<gadZOOKS8371@garnet.acns.fsuMMER.edu> wrote:

> Will n_elements give a number <= zero if the keyword is called like
> "/keyword"
>

No.

N_Elements() returns the number of elements in a variable. If that variable was a (10,10) array then N_Elements would return 100. If however the variable is undefined (which of course is its own special variable type in IDL), then N_Elements returns 0.

If the keyword is set, but set to a negatove number then N_Elements will return will return a poitive number of elements still. If the parameter is simply a flag then you're much better off with the Keywrod_Set() EQ 0 idea.

Cheers,
Phil

Subject: Re: Passing zero as a Parameter/ NOT KEYWORD_SET
Posted by J.D. Smith on Wed, 30 Jun 1999 07:00:00 GMT
View Forum Message <> Reply to Message

"R.Bauer" wrote:
>
> Martin Schultz wrote:
>
>> Martin Schultz wrote
>>
>> and D. Fanning, R. Bauer and J.D. Smith replied

>>
>> [...]
>>
>> oh well! Just when I for once thought I knew IDL ;-(
>> (1) I confess: I confused IF with KEYWORD_SET. It is still sometimes
>> enstranging that
>> IF (2) THEN PRINT,'TRUE'
>> will not print anything...
>
> Did you know that's
>
> if (3) then print,'true'
>
> will print 'true'
>
> All odd number gives true.
>

For non-floating numbers, only the least significant bit is examined.
Note that if (-1) also gives true. For floating numbers, only 0.0 is
false.

JD

--
J.D. Smith |*| WORK: (607) 255-5842
Cornell University Dept. of Astronomy |*| (607) 255-6263
304 Space Sciences Bldg. |*| FAX: (607) 255-5875
Ithaca, NY 14853 |*|

Subject: Re: Passing zero as a Parameter/ NOT KEYWORD_SET
Posted by Vapuser on Wed, 30 Jun 1999 07:00:00 GMT
View Forum Message <> Reply to Message

Martin Schultz <mgs@io.harvard.edu> writes:

> J.D. Smith wrote:
>>
>>
>> That's a bit dangerous. [...]
> Indeed ;-)
>>
>> The best way to proceed is pretend keyword_set() was really
>> named is_defined_and_non_zero(). Forget that it's called
>> keyword_set().

> In fact it is "is_defined_and_uneven" !

HuH?!

```
PRO junk,b=b
  print,keyword_set(b)
END

IDL> junk
       0
IDL> junk,b=1
       1
IDL> junk,/b
       1
IDL> junk,b=0
       0
IDL> junk,b=2
       1

IDL> print,!version
{ mipseb IRIX unix 5.1.1 Jul 20 1998}
```

What version of IDL are you using?

> Just try to pass var=2 into a routine and print keyword_set(var). Hope,
> David will take notice of this in his article.
>

> Another marginal point about setting default values: I recently learned
> from someone's code (cannot remember whose), to use
>   if (n_elements(var) ne 1) then var=default
> instead of
>   if (n_elements(var) eq 0) then var=default
>
> The advantage being that you can prevent program crashes when someone
> passes a vector or array in what is supposed to be a scalar.
>

If I require a scalar I always do this...

IF n_elements(var) EQ 0 THEN var=default else var=var[0]

> And, finally: Use keyword_set when you want to make sure the value of a
> boolean flag is defined:
>   flag = keyword_set(flag)
> Then, later in the code, it's just

> if (flag) then ...
> Or value = x+y*(flag), etc. which would crash otherwise.
>

This, in fact, is the only time I use keyword_set. On all other keywords where values are being passed, I use n_elements() for input and and arg_present for output.

> Regards,
> Martin.
>
> |||||||||||||||\\\\\\\\\\\\\\\----------////////////// //|||||||||||||||
> Martin Schultz, DEAS, Harvard University, 29 Oxford St., Pierce 109,
> Cambridge, MA 02138 phone (617) 496 8318 fax (617) 495 4551
> e-mail mgs@io.harvard.edu web http://www-as/people/staff/mgs/

--
William Daffer: 818-354-0161: vapuser@catspaw.jpl.nasa.gov

Subject: Re: Passing zero as a Parameter/ NOT KEYWORD_SET
Posted by R.Bauer on Wed, 30 Jun 1999 07:00:00 GMT
View Forum Message <> Reply to Message

Martin Schultz wrote:

> Martin Schultz wrote
>
> and D. Fanning, R. Bauer and J.D. Smith replied
>
> [...]
>
> oh well! Just when I for once thought I knew IDL ;-(
> (1) I confess: I confused IF with KEYWORD_SET. It is still sometimes
> enstranging that
> IF (2) THEN PRINT,'TRUE'
> will not print anything...

Did you know that's

if (3) then print,'true'

will print 'true'

All odd number gives true.

Reimar

>
>
> (2) Of course, n_elements(var) eq 1 only makes sense when you expect a
> scalar. Guess, I wasn't clear enough here. It's just that I sometimes
> found I was running into trouble with ambiguous keywords such as COLOR
> which can occasionally be a vector (see SPLOT), but must be scalar on
> (most) other occasions. Now the test 'EQ 1' is of course only suitable
> for the 'other occasions'.
>
> (3) Reimar Bauer makes a good point that you want to define a variable
> locally if you want to make sure not to pass it back (however, I often
> find that passing previously undefined variables back is useful)
>
> Gotta go back and learn some more ;-)
>
> Martin
>
> |||||||||||||||\\\\\\\\\\\\\\\----------////////////// //|||||||||||||||
> Martin Schultz, DEAS, Harvard University, 29 Oxford St., Pierce 109,
> Cambridge, MA 02138 phone (617) 496 8318 fax (617) 495 4551
> e-mail mgs@io.harvard.edu web http://www-as/people/staff/mgs/

Subject: Re: Passing zero as a Parameter/ NOT KEYWORD_SET
Posted by Martin Schultz on Wed, 30 Jun 1999 07:00:00 GMT
View Forum Message <> Reply to Message

Martin Schultz wrote

and D. Fanning, R. Bauer and J.D. Smith replied

[...]

oh well! Just when I for once thought I knew IDL ;-(
(1) I confess: I confused IF with KEYWORD_SET. It is still sometimes
enstranging that
IF (2) THEN PRINT,'TRUE'
will not print anything...

(2) Of course, n_elements(var) eq 1 only makes sense when you expect a
scalar. Guess, I wasn't clear enough here. It's just that I sometimes
found I was running into trouble with ambiguous keywords such as COLOR
which can occasionally be a vector (see SPLOT), but must be scalar on
(most) other occasions. Now the test 'EQ 1' is of course only suitable

for the 'other occasions'.

(3) Reimar Bauer makes a good point that you want to define a variable
locally if you want to make sure not to pass it back (however, I often
find that passing previously undefined variables back is useful)

Gotta go back and learn some more ;-)

Martin

||||||||||||||||\\\\\\\\\\\\\\\\----------////////////// //||||||||||||||||
Martin Schultz, DEAS, Harvard University, 29 Oxford St., Pierce 109,
Cambridge, MA 02138 phone (617) 496 8318 fax (617) 495 4551
e-mail mgs@io.harvard.edu web http://www-as/people/staff/mgs/

Subject: Re: Passing zero as a Parameter/ NOT KEYWORD_SET
Posted by Liam Gumley on Wed, 30 Jun 1999 07:00:00 GMT
View Forum Message <> Reply to Message

Martin Schultz wrote:
> And, finally: Use keyword_set when you want to make sure the value of a
> boolean flag is defined:
> flag = keyword_set(flag)
> Then, later in the code, it's just
> if (flag) then ...

Hi Martin,

I've done this many times myself. However it occurs to me that if FLAG
is defined on entry to the procedure, this method has the potential to
change the size and type of FLAG. For example, in the following cases,
all initial values of FLAG are valid 'true' values as far as KEYWORD_SET
is concerned:

```
flag = indgen(10)
flag = keyword_set(flag)
help, flag
FLAG            INT       =        1

flag = 'TEXT'
flag = keyword_set(flag)
help, flag
FLAG            INT       =        1

flag = [0]
flag = keyword_set(flag)
help, flag
```

```
FLAG            INT       =       1
```

In all these cases, the size, type, and value of FLAG was changed and passed back to the caller. This could potentially lead to problems, unless the caller is explicitly made aware that input keyword values can be changed. I think a better method is to follow the advice DavidF gives in his keyword usage article, which is to embed the KEYWORD_SET call in the call to any procedures or functions where Boolean keywords are used, e.g.

```
myprog, x, y, flag=keyword_set(flag)
```

which guards against changing the size, type, or value of the input keyword FLAG.

The KEYWORD_SET examples above also reveal that KEYWORD_SET accepts as 'true' some pretty odd values (e.g. [0]). For this reason, I was thinking that it might be useful to construct a function that truly tests for Boolean true/false keyword values. In this case, I mean a strict test for a value of zero or one, not the broad IDL definition of true/false. That is, the keyword must pass the following tests to be classified as 'true':
(1) It must be defined,
(2) It must not be a string,
(3) The first element of the keyword must be equal to 1L when converted to LONG.

Here's a rough cut at the function (sans prolog):

```
;---cut here---
FUNCTION KEYWORD_BOOLEAN, KEYWORD

;- Check number of arguments

if n_params() ne 1 then message, 'Usage: RESULT =
KEYWORD_BOOLEAN(KEYWORD)'

;- Return false if keyword is undefined, a string, or first element ne 1
;- (otherwise return true)

case 1 of
  n_elements(keyword) eq 0          : return, 0
  size(keyword, /tname) eq 'STRING' : return, 0
  long(keyword[0]) ne 1L            : return, 0
  else                              : return, 1
endcase

END
```

;---cut here---

This function appears to display more 'correct' behavior, e.g.

```
print, keyword_boolean(indgen(10))
      0
print, keyword_boolean('TEXT')
      0
print, keyword_boolean([0])
      0
```

Cheers,
Liam.

--
Liam E. Gumley
Space Science and Engineering Center, UW-Madison
http://cimss.ssec.wisc.edu/~gumley

Subject: Re: Passing zero as a Parameter/ NOT KEYWORD_SET
Posted by philaldis on Wed, 30 Jun 1999 07:00:00 GMT
View Forum Message <> Reply to Message

I expect Grady didn't realise the can of worms he was opening.

I find that I tend to use :

Keyword_Set() for flags : See if the user wants an option on or off

N_Elements() for numeric values : Test to see if they have defined
it. If they have then test its validity. If anything fails then set
the default.

Arg_Present for returning : e.g. in a getProperty or setProperty
method see if the user wants a particular property returned.

This discussion shows, though, that while we often spend a lot of time
disussing really complex problems, it is still the very simple, yet
fundamental questions which provoke the most interest.

Cheers,
Phil Aldis

Subject: Re: Passing zero as a Parameter/ NOT KEYWORD_SET
Posted by R.Bauer on Wed, 30 Jun 1999 07:00:00 GMT

Martin Schultz wrote:

> J.D. Smith wrote:
>>
>>
>> That's a bit dangerous. [...]
> Indeed ;-)
>>
>> The best way to proceed is pretend keyword_set() was really
>> named is_defined_and_non_zero(). Forget that it's called
>> keyword_set().
> In fact it is "is_defined_and_uneven" !
> Just try to pass var=2 into a routine and print keyword_set(var). Hope,
> David will take notice of this in his article.
>
> Another marginal point about setting default values: I recently learned
> from someone's code (cannot remember whose), to use
> if (n_elements(var) ne 1) then var=default
> instead of
> if (n_elements(var) eq 0) then var=default
>

Hallo Martin,

this is not the best way to set variables because

e.g

```
pro test,var1
 if (n_elements(var1) ne 1) then var1=2
help,var1
end
```

test,var1 & test,var1 & help,var1

You see after executing test, var1 has a value.

Better is:

```
pro test,var1
if n_elements(var1) gt 0 then in_var1=var1  else in_var1=2
help,in_var1
end
```

R.Bauer

>
> The advantage being that you can prevent program crashes when someone
> passes a vector or array in what is supposed to be a scalar.
>
> And, finally: Use keyword_set when you want to make sure the value of a
> boolean flag is defined:
> flag = keyword_set(flag)
> Then, later in the code, it's just
> if (flag) then ...
> Or value = x+y*(flag), etc. which would crash otherwise.
>
> Regards,
> Martin.
>
> |||||||||||||||\\\\\\\\\\\\\\\----------////////////// //|||||||||||||||
> Martin Schultz, DEAS, Harvard University, 29 Oxford St., Pierce 109,
> Cambridge, MA 02138 phone (617) 496 8318 fax (617) 495 4551
> e-mail mgs@io.harvard.edu web http://www-as/people/staff/mgs/
