## Subject: Re: keyword_set([0])
## Posted by JD Smith on Mon, 20 Mar 2006 20:24:14 GMT

View Forum Message <> Reply to Message

On Mon, 20 Mar 2006 10:04:49 -0800, vlk.astro wrote:

```
> Hello,
>
> We are finally being forced to migrate from 5.3 to 6.2, and
> somewhere along the way IDL has bowled a googly with
> keyword_set.
>
> With 5.3, an array [0] was recognized as having been set:
> IDL> print, !version
> { sparc sunos unix 5.3 Nov 11 1999}
> IDL> a=0 & b=[a] & print, keyword_set(a), keyword_set(b)
>       0       1
> IDL> a=1 & b=[a] & print, keyword_set(a), keyword_set(b)
>       1       1
>
> With 6.2, 0 and [0] are indistinguishable:
> IDL> print, !version
> { sparc sunos unix Solaris 6.2 Jun 20 2005      64      64}
> IDL> a=0 & b=[a] & print, keyword_set(a), keyword_set(b)
>       0       0
> IDL> a=1 & b=[a] & print, keyword_set(a), keyword_set(b)
>       1       1
>
> Now, I am not one to rail against progress, especially when
> the new behavior matches the documentation.  But nevertheles,
> it is damned inconvenient, because I think I have around a
> hundred off procedures that depended on the 0/[0] dichotomy.
>
> My question : is there a simple way to replace the old calls to
> keyword_set() with one- or two-liners that will work in both 5.3
> and 6.2 and one that will know the difference between a scalar 0
> and a vector 0?  I don't want to roll my own function because of
> issues of speed (some of the keyword_set's are deeply nested,
> and I'd rather not have the extra overhead of a new function call)
> and aesthetics (i.e., as much native functionality as possible).
> Right now all I have is an ugly concoction that involves size(),
> n_elements() _and_ keyword_set().
>
> Thanks,
> Vinay
```

Try,

```
IDL> print, size(b,/N_DIMENSIONS) gt 0 OR keyword_set(b)
```

The scalar vs. single element vector dichotomy is one which the RSI engineers wish didn't exist, since it no doubt forces many many lines of special purpose code into IDL which do nothing other than maintain this artificial distinction. I think this change in KEYWORD_SET in IDL 5.6 was a limited nod towards reducing the most egregious annoyances which result from this distinction. Here's the blurb:

> The KEYWORD_SET function returns true if its argument is defined and is nonzero, and false (0) otherwise. The specific rules by which the value is determined are given in the IDL Reference Guide. With IDL 5.6, there has been a small change to these rules, designed to make KEYWORD_SET useful in a larger number of cases. Previously, KEYWORD_SET would return true if it's argument was an array, regardless of the value. This behavior has been changed: Arrays with more than 1 element are treated as before, but 1-element arrays are treated in the same way as scalar arguments, and the value returned by KEYWORD_SET depends on the value of the element.

Personally, I would change the logic so that a single element vector isn't required to be true by default, or isn't ever passed in for boolean arguments/keywords. Changing the logic in this way should be compatible across IDL 5.3/6.2, without new and hard to maintain tests like this.

JD

## Subject: Re: keyword_set([0])
## Posted by Andrew Cool on Mon, 20 Mar 2006 21:19:01 GMT

View Forum Message <> Reply to Message

Vinay,

"IDL has bowled a googly" is a lovely metaphor for us cricket-playing nations,
but you'll have the Yanks scratching their heads... ;-)

Cheers,

Andrew,
(Adelaide, Australia)

## Subject: Re: keyword_set([0])
## Posted by vlk.astro on Mon, 20 Mar 2006 21:46:02 GMT

View Forum Message <> Reply to Message

JD Smith wrote:

> 
> IDL> print, size(b,/N_DIMENSIONS) gt 0 OR keyword_set(b)

That is simple and elegant!

Perfect!

> Personally, I would change the logic so that a single element vector
> isn't required to be true by default, or isn't ever passed in for
> boolean arguments/keywords.  Changing the logic in this way should be
> compatible across IDL 5.3/6.2, without new and hard to maintain tests
> like this.

yes, I've learnt my lesson, at least in this case!

Thanks,
Vinay

## Subject: Re: keyword_set([0])
## Posted by vlk.astro on Mon, 20 Mar 2006 21:52:57 GMT

View Forum Message <> Reply to Message

andrew.cool@dsto.defence.gov.au wrote:
> Vinay,
>
> "IDL has bowled a googly" is a lovely metaphor for us cricket-playing
> nations,
> but you'll have the Yanks scratching their heads... ;-)
>
> Cheers,
>
> Andrew,
> (Adelaide, Australia)

And my next question deals with the keyword parameters to the function
LBW.PRO :)

(Actually, the Yanks should know all about googlies if they've been
following their President's recent adventures with a tennis ball in
Pakistan.)

Vinay

Subject: Re: keyword_set([0])
Posted by vlk.astro on Mon, 20 Mar 2006 23:49:14 GMT
View Forum Message <> Reply to Message

JD Smith wrote:

>
> IDL> print, size(b,/N_DIMENSIONS) gt 0 OR keyword_set(b)

Unfortunately, upon further investigation, this doesn't quite work.
The behavior I am looking for is (sorry I wasn't clearer before): if
A=0 and B=[0], then expr(A)=0 and expr(B)=1 in all versions of IDL (for
backwards compatibility and all that)

instead, 5.3 and 6.2 never match up properly --

| v5.3 | v6.2 | expr() |
|---|---|---|
| 0 | 0 | keyword_set(A) |
| 1 | 0 | keyword_set(B) |
| 1 | 0 | size(A,/dim) gt 0 OR keyword_set(A) |
| 1 | 1 | size(B,/dim) gt 0 OR keyword_set(B) |

i.e., if it works in 5.3, it doesn't in 6.2, and vice versa.

Vinay

Subject: Re: keyword_set([0])
Posted by JD Smith on Tue, 21 Mar 2006 19:24:01 GMT
View Forum Message <> Reply to Message

On Mon, 20 Mar 2006 15:49:14 -0800, vlk.astro wrote:

> JD Smith wrote:
>
>>
>> IDL> print, size(b,/N_DIMENSIONS) gt 0 OR keyword_set(b)
>
> Unfortunately, upon further investigation, this doesn't quite work.
> The behavior I am looking for is (sorry I wasn't clearer before): if
> A=0 and B=[0], then expr(A)=0 and expr(B)=1 in all versions of IDL (for
> backwards compatibility and all that)
>

```
> instead, 5.3 and 6.2 never match up properly --
>
> v5.3   v6.2   expr()
> 0      0      keyword_set(A)
> 1      0      keyword_set(B)
>
> 1      0      size(A,/dim) gt 0 OR keyword_set(A)
> 1      1      size(B,/dim) gt 0 OR keyword_set(B)
>
> i.e., if it works in 5.3, it doesn't in 6.2, and vice versa.
```

Not /DIM, /N_DIM (actually I'm not sure if that exists in 5.3, but in
that case (size(a))[0] should work). Your version with /DIM should work
for scalars and single element vectors anyway, so I'm not sure why it
doesn't here. In IDL5.5 (the oldest I have):

```
IDL> a=0
IDL> print,size(a,/dim) gt 0 OR keyword_set(a)
       0
```

and of course the same in 6.2. I'm very surprised v5.3 is different --
in fact I can't imagine how it could be.

JD

## Subject: Re: keyword_set([0])
## Posted by vlk.astro on Tue, 21 Mar 2006 21:18:48 GMT

View Forum Message <> Reply to Message

JD Smith wrote:

> Not /DIM, /N_DIM (actually I'm not sure if that exists in 5.3, but in
> that case (size(a))[0] should work).

whoops! teaches me to always cut-and-paste!

yes, /n_dim does exist in 5.3, and it works as advertised.

Thanks again!
Vinay