

---

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  
> { sparcsunos 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  
> { sparcsunos 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 nevertheless,  
> 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

---