Subject: keyword_set([0])
Posted by vlk.astro on Mon, 20 Mar 2006 18:04:49 GMT

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