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 |*|