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