
Subject: Re: Arg_Present

Posted by [J.D. Smith](#) on Wed, 06 Aug 1997 07:00:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

David Fanning wrote:

>

> J.D. Smith writes:

>

>> I'm just full of complaints this week. This particular one is about
>> Arg_Present. It is a lovely addition, and one for which I clamored
>> before the release of IDL v5, but it has something left out... it does
>> not correctly identify *inherited* keyword passed variables as return
>> variables. This may seem like a small detail, but consider this
>> scenario: A subclass' GetProperty method calls its superclass' method
>> with any extra keywords it receives. The superclass' GetProperty method
>> checks if keywords are present as return variables before computing a
>> property for return. But this doesn't work... _EXTRA keywords will not
>> trigger arg_present, and the whole system is broken.

>

> Well, as long as we are griping about Arg_Present, let me
> get my two cents in. :-)

>

> I was under the impression that Arg_Present was introduced
> to solve the sometimes subtle problem of telling whether
> a keyword was *used* or not. And in limited circumstances
> it does this well enough. The danger is in using this
> seemingly useful function in place of the much more
> useful, but completely misnamed (at least for the purpose
> it is being used for), N_Elements.

>

> In particular, Arg_Present returns a 0 (meaning no argument
> present) when a keyword argument is present, but the argument
> is passed by value. It only returns a 1 if the argument is
> present and is passed by reference.

>

> Now, to give RSI credit, this is all documented correctly.
> It is just that most people using Arg_Present will *assume*
> it does what its name suggests it does and end up using it
> incorrectly. It's the Keyword_Set problem all over again.
> It seems to me this whole idea of knowing if and when a
> keyword parameter was used needs more thought and a much
> more consistent (and meaningfully named) interface.

>

You are indeed correct that a better interface would be useful.

Remember, however, that arg_present works on both keywords *and* arguments, and serves to tell you if any variable passed is passed by

reference. As you point out, using `n_elements` or `keyword_set` on an `*undefined*` passed argument (e.g. `mypro, outvariable=undefinedvar`) will be to no avail... you'll not be able to discriminate between this case, and the case of it never having been passed at all, unless of course you give `undefinedvar` a value before passing it, which is not ideal.

But nonetheless, `arg_present` works as advertised. However, the present issue is inherited keywords and the fact that they are always passed by value, which introduces some problems with object oriented methodology. Cf. the my latest article in this thread.

JD
