
Subject: Re: Generating keyword parameters from strings
Posted by [Paul Van Delst\[1\]](#) on Mon, 27 Jul 2009 14:39:41 GMT
[View Forum Message](#) <> [Reply to Message](#)

David Fanning wrote:

> Paul van Delst writes:

>

>> Of course it would be nicer if I didn't have to remember it at all. Passing
>> arguments/keywords by reference or value is an implementation detail the user shouldn't be
>> concerned with. Having to consider it in IDL OO programming is mildly ironic.

>

> I think you are forgetting the 80-year history
> of IDL. (Or is it 100? I can never remember.)

Oh no, I'm not forgetting. FORTRAN (which necessarily has a longer history than IDL) had the same issue although it was a tacit understanding about passing mechanism. I assume IDL has that restriction because FORTRAN did (IIRC, wasn't IDL originally written in f77 [or f66?] on a VMS system? But you would know more about that than me.).

Now that we have Fortran (not all caps, f90 and later) the passing mechanism can vary based on situation - the compiler can choose the better method (for a suitable definition of "better").

If you write f90+ code that depends on a particular passing mechanism it's not illegal (in many cases at least), but you're asking for chunks to be removed from your read-end in the future. It can suck for some MPI/F90+ amalgams since the former relies on passing by reference whereas the latter does not.

So, Fortran grew out of the de-facto need to be particular about passing mechanism. I see no reason why IDL can't. It certainly would make it easier for new users who wonder what they've done wrong until they discover you can't pass arguments *out* from procedures/functions when the actual arguments in the calling routine are array elements or structure components.

cheers,

paulv
