His comments about the 7/2 by themselves tell me that he doesn't know
jack about programming. I know for sure that at least 2 of the 4
competing languages (the C family, Java and possibly Maltab) all do the
exact same thing as IDL. EXACT! (Okay, maybe Matlab doesn't, I
haven't used it in 8 months.) The fourth is Maple, and it may still do
the same thing as IDL (it's been so long since I did simple math in
Maple as well). His complaint is utterly baseless.

I'd also like to point out that Java isn't entirely consistent in how
it passes variables to functions. If memory serves, structures and
objects, and pointers by extension are the only things passed by
reference, whereas everything else is passed by value. This caused
much fussing on my part during my first year programming courses, since
I was coming from high-school where I learned C++. I still resent Java
to this day for that reason. IDL is much more consistent. None of the
stupid "This is an int, so it's by value, that's an object, so it's by
reference," which makes sense from an engineering standpoint, but not
from a programming standpoint.

Richard Edgar wrote:
> Paul van Delst wrote:
>
>>> Someone sent me a link to this interesting IDL rant this morning:
>>>
>>>   http://www.sccs.swarthmore.edu/users/07/sstvinc2/research/st upid.html
>>
>> Hee hee. I thought it was pretty funny. I probably would've done
>> something similar back when I was one of dem young whippersnappers
>> (well, I would've if anything but Fortran was available....)
>>
>> Apart from the fact that some of the info was just wrong, the rant shows
>> the writers lack of experience with programming languages in general. To
>> say nothing of exiting college and entering the real world where being
>> able to distinguish and effectively handle the differences between the
>> the way things *should* be and the way they actually *are* are a
>> definite plus.
>
> If I were to write my list of IDL annoyances, I think mine would be
> somewhat different to this.
>
> I'd comment that the first one, about 7/2 vs 7/2.0 is a place where I'd
> say that IDL definitely does the right thing. And comment 12 about
> comparisons contradicts the first comment (and seems to show a lack of
> understanding of floating point arithmetic).

>
> Row-major vs column major is a silly point... you just need to know
> which way the language does it, and that's the end of the matter. And
> I'd dispute the bit about 'every other language in the history of
> mankind' too ;-)
>
> There is some basis for the comments about the 'compiler' and need for
> recompilation. I'd say that this is due to a bit of bad nomenclature on
> IDL's part, since IDL is more an interpreted than a compiled language.
>
> The complaint about inconsistency in how variables are passed to
> routines is fair enough. ISTR my encounter with this was based on
> passing a structure vs. passing elements of the structure. However, the
> passing method is irrelevant. All I should have to do is declare whether
> I intend to modify the variables or not, and whether they should be
> defined on entry. Something like INTENT(IN), INTENT(OUT) and
> INTENT(INOUT) would be ideal ;-)
>
> Richard