
Subject: Re: IDL 8.0 compile_opt changes
Posted by [JDS](#) on Fri, 08 Jan 2010 21:42:38 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Jan 6, 1:35 pm, pp <pp.pente...@gmail.com> wrote:

> I realised now that I had not thought this through:

>

> On Jan 5, 9:54 pm, pp <pp.pente...@gmail.com> wrote:

>

>>> 1. Overloading structure dereference and method invocation breaks the
>>> ability to semantically parse a.b.

>> That is a good point that I had not considered. I only noticed that
>> with the assumed idl2 option, there would be no ambiguity in complete,
>> correct lines. But as you point out, there would be in the middle of
>> writing a line. Though, as you indicate in (3), the ambiguity would
>> only occur between methods and fields of self. In other cases, the
>> interpreter should know whether you typed a structure or an object.

>

> This is not actually a problem. It would just be the same kind of
> ambiguity that happens in the middle of typing a name, when there are
> more than one that begin with that same characters. At that point,
> using completion should give the user both options. In the case of the
> function/member ambiguity, the options would be "a.b(" and "a.b", as
> the first is the function, and the second is the structure member.

If you are the IDL interpreter, that's fine, because you know `a' is
either an object or a structure and can act accordingly. If you can
only presume that `a' may be some unknown structure, or some object,
the space of completions just ballooned by a factor of 2 at minimum.
If IDL had adopted the overloaded "." operator initially, the saved
keystroke of this syntactic might be worth the added ambiguity. Given
that it didn't, the pain associated with switching horses mid-streams
makes it a questionable move, in my view.

JD
