Subject: Re: IDL 8.0 compile_opt changes
Posted by wallabadah on Thu, 07 Jan 2010 00:30:18 GMT
View Forum Message <> Reply to Message

I agree with JD:

> 3. IDL is not Python. IDL enforces strict encapsulation of object
> data, i.e. all object data must be accessed through a method (except
> within the object's methods themselves). Python has no object data
> encapsulation. In Python it is natural to mix method invocation with
> data access. In IDL this only occurs only in an object's own
> methods. Which is clearer?
> self->limit, self.limit
> self.limit, self.limit

As Ken has pointed out, it was necessary to change to square brackets for array indexing to avoid ambiguity. Introducing a new ambiguity by allowing the dot operator to access both structure elements and object functions is definitely a Bad Idea. Apart from breaking code completion as pointed out by JD, it makes code much harder to read - whoever wrote "I still find it clear from the syntax the meaning of the second line" is having a joke with us - this is step 1 of code obfuscation. I also think the -> operator visually depicts what it does, making the intention of the author absolutely clear when reading code. I don't think changing the language's syntax so that it is more like another language is a valid reason. Assuming introduction of the dot operator in IDL8.0 doesn't break ->, we're in a situation where there are two operators to do the same thing. Will ITTVIS deprecate -> in IDL10.0, requiring yet another update to everyone's code libraries. Regardless, with operator overloading those die-hards that want '.' to mean '->' could implement it in their own objects themselves.

On the subject of updating parentheses in old libraries - if the IDL interpreter/compiler is capable of discerning the different intent of () when parsing idl code into byte code - shouldn't it be possible to use this information to update the libraries (with zero human intervention)? I don't have a great understanding of how byte code is generated, but if the code can be compiled and run correctly by IDL, then IDL 'knows' whether the parentheses in 'old' code should be changed to square brackets. If this assumption is correct, updating () to [] in old code should be built in to IDL 8.0.

Will.