Subject: Re: Another small V8.0 bug
Posted by penteado on Tue, 27 Jul 2010 15:33:21 GMT
View Forum Message <> Reply to Message

On Jul 27, 11:39 am, Paul van Delst <paul.vande...@noaa.gov> wrote:
> Crikey. I hope the documentation clearly states how to turn that default behaviour OFF in one's idl_setup.pro file.
>
> Because, you know, ITTVIS *did* make this behaviour user selectable, right? RIGHT?

As far as I know, this cannot be turned off.

> I dislike the
> x = object_name(args)
> alias for the regular
> x = obj_new('object_name', args)
> because it now means you should include a comment in the code telling the future maintainers what is happening. That is,
> rather than doing something like:
>
> x = obj_new('list',args)
>
> you'd do
>
> ; Create a list object
> x = list(args)
>
> I'm all for syntactic sugar, but this is more like aspartame (groan :o) -- it's obfuscating what was before, IMO, a
> clear indication of what was happening. Now users will have to maintain (or, worse, debug) both the object creation
> *and* the comment.

I find the new syntax better. obj_new() is too verbose. With its string class name it looks more like a dynamic evaluation function, such as execute(), more of a workaround for object use in the lack of object syntax. The class name notation makes it more clear that it is the creation of a variable of a certain type, and is similar to what is used in other languages. For that reason people occasionally already made a wrapper to make the object through a function call anyway (as in yesterday's thread by mankoff).

Most uses of the variable in the code will make it visible that it is an object. To me it seems that to comment that x=list(args) creates an object is just as necessary as saying that x=where(args) creates a 1D integer (maybe long) array, or that x=min(args) creates a scalar of the same type as args, or that x=dist(args) creates a 2D array of floats, or that x=dblarr(args) makes an array of doubles. None of

these is saying much anyway, it is not saying what the variable will contain. If I just see an uncommented obj_new() of some class I am not familiar with, it tells me no more of what that variable will do than if I see only an uncommented assignment from a function I am not familiar with. The object can be anything, just as the function return value can be anything. In both cases I would have to either keep reading the code to see how it is used, or look up that function/class to find out what that variable will be.

I find that more informative and needed comments would actually say the purpose of that specific variable, which would not be conveyed by just the obj_new() call: things such as "create a list with file names that are referenced", or "list of targets with observations above the threshold", or "hash for the keywords set in the parameter file".

Those who find obj_new() more clear can keep using it, but I do not see any assured clarity improvement just from its use. Since IDL is dynamically typed, and function names are resolved at runtime, statically looking at code never does completely tell what will happen anyway.

---