Subject: Re: Interesting Rant
Posted by Maarten[1] on Wed, 15 Nov 2006 13:25:54 GMT
View Forum Message <> Reply to Message

David Fanning wrote:
> I disagree with much of what he has to say, but I
> can't tell whether this is because I am old enough
> to remember what things were like *before* IDL, or
> whether I've just become inured over the years. :-(

On the contrary, I'd like to imagine what things can become *after* IDL. While I would list a different set of annoyances, I'd say you have become inured to the IDL idiosyncrasies over the years. Lovely word, "inured", had to look it up though (can you tell I'm not a native speaker?)

* Arrays in structures can not be resized. This also applies to objects.
* You can have resizable arrays in structures, if you use pointers. The way to access data in the array changes, you'll have to rewrite your code.
* Pointers that can get lost in a scripted language. If I want a memory leak, I'll use C, Fortran, ...
* Default integer size is 16 bits (how long have computers been at 32bits by now; how much code will break if you change that to 32 (or even 64 bits)? How can code break on such a change? I think the code was broken in the first place if it relies on this bit-size.
* Individual floating point constants are float, not double.
* Direct graphics seem to be dead, object graphics are not practical for interactive use. Hello, the I in idl stands for interactive, right?
* Procedure arguments as output parameter and array elements.
* Brain dead for loops. Nice to have a vector engine (which doesn't work in all cases, idl_validname() only accepts strings, not string arrays, so for loops cannot be avoided in all cases), but sometimes an explicit for loop aids readability (histogram jugling, anyone?).
* Stupid end of line behaviour, especially on interactive use.
* Logical test on least significant bit only
* The UI on Mac OS X. Using X11 on Mac OS X does not make a port.
* Special characters in graphs, combined with the butt ugly Hershey fonts. The graphs look old-fashioned, and like they were made with some homebrew software, rather than a professional, expensive tool. Yes, you can do better in IDL, many don't.

So while IDL probably is an improvement over what came before it (I'm too young to really tell), those programs are truly dead and buried. IDL seems to survive, despite better solutions, especially for interactive use. The only things that keep idl alive are the dinosaurs that use it, and the legacy code that has been written for it. Others

have suggested to use Python as a basis, and I think I agree, although some array indexing issues will be as annoying as IDL. At least there is a large community behind it, and the core of python is free.

Maarten