Subject: Re: Top 10 IDL Requests
Posted by Jeff Guerber on Fri, 21 Jul 2000 07:00:00 GMT
View Forum Message <> Reply to Message

Wow, it looks like we're _finally_ able to post again! And it only took them four months!

Anyway, after thinking about it for a while, here's _my_ IDL wishlist. Some of the things have been mentioned by other people, but I'm including them anyway to add my vote. Beyond the first few, they're not necessarily in any particular order of preference.

1) Definable lower array bounds. I often have to translate algorithms originally developed in Fortran, and changing everything to 0-based indexing is a MAJOR pain (not to mention bug-prone). Besides, _my_ fingers start at "one", not "zero"! :-) In some cases I'd like to go from, say, -10:+10; yeah, I know I can use an extra array, but what about -10000:+10000 ? Fortran's had this for DECADES... why not IDL?

2) It's probably waaaay too late to change this, but: I find it really annoying that pointer dereferencing is a low-precedence prefix operator; I'd much prefer a postfix operator at the same precedence as structure-field references and array subscripts. So instead of having to write, for example,

```
(*(*statep).plotIDSp)[i]
```

(which I actually took from one of my programs), I could simply say,

```
statep*.plotIDSp*[i]
```

which IMHO is MUCH more readable -- no excess parentheses and you don't have to jump clear back to the front of the expression to see that "plotIDSp" is a dereferenced pointer. Even "(statep*).(plotIDSp*)[i]" would be better than the current form. For dereferencing an array of pointers, the syntax would be "array[i]*".

3) Class data, public data, and private methods in objects.

4) Initialization in structure definitions. I'd like to be able to write, for example,

```
pro struct__define
void = { struct, a:2, b:4.5, c:"Fanning's book is great" }
return
end
```

and have a new {struct} automatically be initialized to {a:2, b:4.5,

c:"Fanning's book is great"} instead of {a:0, b:0.0, c:""}.

5) Specifying colors by name... preferably the whole X11 palette from /etc/X11/rgb.txt. I've sometimes toyed with the idea of writing an object that would read rgb.txt, load colors _on demand_... and keep track of where in the color table each went, so that any one color doesn't get loaded into multiple table entries.

6) A standard, and _comprehensive_, I/O package for FITS files. Yes, I know there are about 5 packages available in Wayne's (excellent!) idlastro library; but they all have different capabilities, and until recently there was no good way of figuring out which was best for an application. (I think there's now a document that discusses the strengths and weaknesses of each package.) Given IDL's background and popularity in astronomy, I've always been puzzled by its lack of built-in FITS support.

7) Recalling the graphics system variables on a window-by-window basis, so we don't have to restore them manually when changing windows.

8) String subscripting.

9) BREAK and CONTINUE (or CYCLE) statements for loops.

10) Complement of WHERE.

11) Subscript shorthand for the upper-bound array index.

Sorry, I guess that's more than 10, isn't it? Oh well, some of them probably don't have the proverbial snowball's chance of getting into the language anyway!

Jeff Guerber

Raytheon ITSS
NASA Goddard Space Flight Center
Oceans & Ice Branch (code 971)

Any opinions here are my own. Well, in a few cases I borrowed them from other people, but they're certainly not Raytheon's or NASA's!