Subject: Re: User selectable lower array bound?
Posted by Jeff Guerber on Fri, 03 Aug 2001 00:39:50 GMT
View Forum Message <> Reply to Message

On Thu, 2 Aug 2001, Paul van Delst wrote:

> Is is just me, or would anyone else find useful the ability to define
> arrays in IDL such that the lower bound is *not* always zero? Sorta
> like:
>
> x = FINDGEN(11, LOWER = -5)
> or
> y = DBLARR(100, LOWER = 1)
>
> so that accessing elements such as x[-4] or y[100] are o.k.? [...]

Here, here!! This was #1 on my (13-item) contribution to last summer's "Top 10 IDL Requests" discussion. As I pointed out then, Fortran's had this capability for decades. (And IDL is expressly a data-analysis language, like Fortran, not a systems-programming language like C.) The biggest problem I see is that certain IDL intrinsics, like WHERE(), return -1 to indicate an invalid index. Perhaps WHERE could return (lowerbound-1) instead, on the presumption that existing programs would be using 0-based arrays? Of course it's much better to check the COUNT= keyword anyway. (This would also be a good application for some sort of "undefined value" type.)

(IMHO, the two worst features IDL picked up from (presumably) C are starting arrays at 0 (which makes some sense in C, due to the tight coupling of arrays and pointers, but this isn't the case in IDL), and prefix syntax for the array dereferencing operator. Concerning the latter, I've since learned that even Dennis Ritchie apparently now thinks it was a mistake; from his "The Development of the C Language" (http://cm.bell-labs.com/who/dmr/chist.pdf), page 12:

> An accident of syntax contributed to the perceived complexity of the language. The indirection operator, spelled * in C, is syntactically a unary prefix operator, just as in BCPL and B [C's predecessor languages]. ... Sethi [Sethi 81] observed that many of the nested declarations and expressions would become simpler if the indirection operator had been taken as a postfix operator instead of prefix, but by then it was too late to change.

Oh, make that THREE worst features: Also the use of integers for Boolean data, instead of having a true logical or Boolean data type. VERY confusing.)

Of course, these opinions are my own and don't reflect those of

Raytheon or NASA.

Jeff Guerber
Raytheon ITSS
NASA Goddard Space Flt Ctr
Oceans & Ice Branch (971)
