
Subject: Re: User selectable lower array bound?
Posted by [Jeff Guerber](#) on Thu, 09 Aug 2001 06:39:38 GMT
[View Forum Message](#) <> [Reply to Message](#)

On 3 Aug 2001 bennetsc@NOSPAMucs.orst.edu wrote:

>> It seems like such a simple thing to be able to do (with default
>> action being start at
>> index 0) although I'm sure the amount of work required to
>> implement this would be
>> horrendous. Still, it shur would be nice.....
>>
> That depends upon how IDL already keeps track of arrays
> internally. In PL/1, for example, one declared an array with the
> boundaries for each dimension in the form lowerbound:upperbound,
> where specification of the lower bound and the colon were optional.
> If only the upper bound were specified, then the lower bound defaulted
> to 1. In its internal representation of arrays, IIRC, PL/1 kept
> the lower and upper boundaries of each dimension as part of a control
> block preceding the actual array memory. If a language implementation
> doesn't already store both boundaries, or equivalently, the lower
> boundary and number of elements, for each dimension, then yes, adding
> such support might well be a major headache.

Well, IDL does perform bounds checking, even for arrays passed into a procedure as arguments, so it must already store at least either the upper bound or the number of elements (which are equivalent since the lower bound is fixed). It's likely that this is only done in one place, so implementing lower bounds in the IDL core might not be all that much work. HOWEVER...

Having thought about this further, I now think the more serious problem would be all the library procedures (and not just RSI's!) that assume you can loop over the elements of any array by going from 0 to `n_elements(array)-1`. (Aiiigh!) Unless the bounds are lost across procedure calls (as Paul pointed out that Fortran does), which can sometimes be useful but which kind of defeats the point of having definable bounds, if you ask me.

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