Subject: Re: User selectable lower array bound?
Posted by Paul van Delst on Fri, 03 Aug 2001 14:08:24 GMT
View Forum Message <> Reply to Message

Jeff Guerber wrote:
>
> 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.)

Well, maybe WHERE could work as it does now, but for cases where the start index is not zero, a function like the Fortran 90 intrinsic LBOUND() could be used.

BTW, I never check the WHERE result either, always the COUNT value.

paulv

--
Paul van Delst A little learning is a dangerous thing;
CIMSS @ NOAA/NCEP Drink deep, or taste not the Pierian spring;
Ph: (301)763-8000 x7274 There shallow draughts intoxicate the brain,
Fax:(301)763-8545 And drinking largely sobers us again.
Alexander Pope.