Subject: Re: User selectable lower array bound?
Posted by John-David T. Smith on Mon, 06 Aug 2001 20:52:13 GMT
View Forum Message <> Reply to Message

Paul van Delst wrote:
>
> "Pavel A. Romashkin" wrote:
>>
>> Craig Markwardt wrote:
>>>
>>> Well, as grumpy as I have been in the past about IDL wishes, this is
>>> one thing I do not want to have in IDL now!
>>
>> I am with you Craig. Besides, for the purists of array indexing, I think
>> it is unfair to dasignate a *lower* array bounds. We don't designate the
>> *upper* one.
>
> In the context of initially declaring an array in IDL, sure you do:
>
> x = fltarr(10)
>
> declares the upper bound as 9. We also designate a lower bound: 0. The difference between
> the two is that I can change the former.
>
>> To be exact, we need a zero point fixed
>
> why?
>
>> and the ability to
>> extend an array in both directions. This way, I can add data in both
>> positive and negative directions.
>
> Why would this functionality be any different to what exists now? And, to me at least,
> allowing -ve indices would make this sort of data manipulation easier to understand, i.e.
> extend array in -ve direction => negative indices.

The most annoying thing about IDL arrays to me is the need always to
test whether they exist or not when concatenating onto them. The idea
of extending arrays in both directions would be neatly summed up by
allowing:

a=[b,a] & a=[a,b] even if a doesn't (yet) exist.

Either that, or IDL needs a list type which allows such operations.
Wasn't that just me ranting about special case functionality leading to
inconsistency?

But anyway, my $2e-2.

JD