
Subject: Re: User selectable lower array bound?

Posted by [Paul van Delst](#) on Mon, 06 Aug 2001 16:06:51 GMT

[View Forum Message](#) <> [Reply to Message](#)

David Fanning wrote:

>

> Ben Tupper writes:

>

>> I just whipped up an object to allow you to do just this kind of

>> pseudo-indexing for 1d vectors.

>

> Oh, sure, you can do it with objects. But

> isn't that, like, cheating? In any case,

> scientists are too busy to learn yet one

> more useful programming practice.

Mostly, yes, :o) but, with all due respect to Ben and his rather natty and cool code snippet (I'm facing north towards Maine, bowing :o), I would rephrase it more like:

"some scientists would like variable-bounds indexing to be indistinguishable from current command line vector/matrix operations without having to learn :

```
;-----START HERE
```

```
; EXAMPLE
```

```
; IDL> x = obj_new('findgen', 11, lower = -5)
```

```
; IDL> print, x->GetData([-4, 0, 5])
```

```
; 1.00000 5.00000 10.0000
```

```
; IDL> obj_destroy, x
```

```
;-----
```

```
; GetData
```

```
;-----
```

```
FUNCTION FINDGEN::GetData, Indices
```

```
Return, (*Self.Data) [Indices - Self.Lower]
```

```
END; GetData
```

```
;-----
```

```
; SetProperty
```

```
;-----
```

```
PRO FINDGEN::SetProperty, N = N, Lower = Lower
```

```
If n_elements(Data) NE 0 Then *Self.Data = Data
```

```
If n_elements(N) NE 0 Then Begin
```

```
    ;1d vectors only
```

```
    Self.N = N[0]
```

```
    ;either 'redefine' the variable or 'undefine' it
```

```
    If Self.N GT 0 Then *Self.Data = Findgen(Self.N) Else $
```

```
    Dummy = Size(Temporary(*Self.Data))
```

```
EndIf
```

```
If n_elements(Lower) NE 0 Then Self.Lower = Lower[0]
```

```

END    ;SetProperty
;-----
;    GetProperty
;-----
PRO FINDGEN::GetProperty, Data = Data, N = N, Lower = Lower
If Arg_present(Data) Then Data = *Self.Data
N = Self.N
Lower = Self.Lower
END    ;GetProperty
;-----
;    Initialization
;-----
FUNCTION FINDGEN::INIT, N, LOWER = lower
If n_elements(N) EQ 0 Then Self.N = 0L Else Self.N = 0L > N[0]
If n_elements(lower) EQ 0 Then Self.Lower = 0L Else Self.Lower = Lower[0]
If Self.N GT 0 Then Self.Data = Ptr_NEW(Findgen(Self.N))    Else $
    Self.Data = Ptr_NEW(/Allocate)
Return, 1
END    ;Init
;-----
; Cleanup
;-----
PRO FINDGEN::Cleanup
If Ptr_Valid(Self.Data) Then Ptr_Free, Self.Data
END    ;Cleanup
;-----
;    Definiton
;-----
PRO FINDGEN__DEFINE
Struct = {FINDGEN, $
    Data: ptr_new(), $    ; the data array
    N: 0L, $                ;this handles only 1d arrays right now
    Lower: 0L}              ;the indexed address of the lower bound
END
;-----END HERE"

```

```

--
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.

```
