Subject: Re: Is a dynamically sized pointer array object component possible?
Posted by Michael Galloy on Fri, 22 May 2009 16:45:44 GMT
View Forum Message <> Reply to Message

Paul van Delst wrote:
> That's what my first example above does (it was the only way I could
> make it work). I was trying to avoid that if possible to avoid the
> double dereferencing that would require in the object methods - as in
> your "get" and "set" method:
>
>> function blah::get, m, n
>> compile_opt strictarr
>>
>> return, (*(*self.y)[m])[n]
>> end
>>
>> pro blah::set, m, n, value
>> compile_opt strictarr
>>
>> (*(*self.y)[m])[n] = value
>> end
>
> And that is what I am doing now in my code. For example, my "set" method
> does
>
> *(*self.Frequency)[_Band] = Frequency
> *(*self.Response)[_Band] = Response
>
> (where Frequency and Response are vectors.[*])
>
> I just wanted to avoid the *(*.self.y) double dereference (DD) if
> possible. It has zero impact on the user, of course - I want to avoid
> the DDing for my own benefit (insert sheepish grin here)
>
> Thanks for taking the time to write the code. It's a nice teaching example.

Another way to do it would be for "::allocate, n" to just create a
pointer to a single vector of size TOTAL(n) and also save the n array,
then for "::get, i, j" to use TOTAL(n, /CUMULATIVE) to find the correct
value(s). I'm not sure that would be simpler, but it would eliminate the
double dereference.

> [*] BTW, note also my use of "_Band". I have now adopted your
> methodology for things like,
>
> ; Check band keyword argument
> IF (N_ELEMENTS(Band) GT 0) THEN _Band = LONG(Band[0])-1 ELSE _Band = 0L
>

> based on your post a few days ago. I've noticed that these type of
> small, incremental changes to create more robust code (like the snippet
> above) eventually leads to shifts in other people's perceptions about
> writing clean code (e.g. no side effects). Nothing earth shattering in
> this little post scriptum, of course, but still neato.

Cool!

Mike
--
www.michaelgalloy.com
Associate Research Scientist
Tech-X Corporation
