
Subject: How to get numbers into passed structure elements
(pass-by-value/reference problem).

Posted by [Paul Van Delst\[1\]](#) on Wed, 27 Feb 2008 15:02:46 GMT

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

Hello,

This subject comes around every now and again, but I can't find anything useful searching.

I'm calling a function, `CRTM_Read_Atmosphere_Record()`, like so:

```
Atm = PTRARR(n_Profiles)
FOR m = 0, n_Profiles-1 DO BEGIN
  Atm[m] = PTR_NEW({CRTM_Atmosphere})
  result = CRTM_Read_Atmosphere_Record( FileID, *Atm[m], DEBUG=Debug )
  ...check result....
ENDFOR
```

That function in turns calls another function, `CRTM_Read_Cloud_Record()`, like so:

```
FUNCTION CRTM_Read_Atmosphere_Record, FileID    , $ ; Input
      Atm    , $ ; Output
      DEBUG=Debug    ; Optional input
.....
FOR n = 0, Atm.n_Clouds-1 DO BEGIN
  result = CRTM_Read_Cloud_Record( FileID, (*Atm.Cloud)[n], DEBUG=Debug )
  ...check result....
ENDFOR
```

And inside the `CRTM_Read_Cloud_Record()` function I do the following:

```
FUNCTION CRTM_Read_Cloud_Record, FileID        , $ ; Input
      Cloud    , $ ; Output
      DEBUG=Debug    ; Optional input
.....
Type = Cloud.Type
READU, FileID, Type, $
      *Cloud.Effective_Radius, $
      *Cloud.Effective_Variance, $
      *Cloud.Water_Content
Cloud.Type = Type
```

(and similar for an Aerosol structure and I/O function. Also, assume the pointer component

allocation has been done to the correct array size))

As you have probably guessed, by the time I inspect all the data that is returned in the original Atm pointer array, everything is fine *except* the cloud type flag. It is zero.

So, I realise this is one of those pass-by-reference or pass-by-value things, but how does one get around it? Do I:

- a) make the Type component of the Cloud structure a pointer? (Yuk!)
- b) change the way I pass the Cloud structure into the CRTM_Read_Cloud_Record() fn?
i.e. not reference the cloud structure array via the index [n].

I would much prefer (b), but will that entail copying entire structures? The number of "Clouds" associated with any particular "Atm[m]" profile is variable.

Thanks for any info.

cheers,

paulv

p.s. FWIW, the structure definitions are:

```
PRO CRTM_Atmosphere__Define
void = { CRTM_Atmosphere, $
    n_Layers      : 0L, $
    n_Absorbers   : 0L, $
    n_Clouds      : 0L, $
    n_Aerosols    : 0L, $
    Climatology   : 0L, $
    Absorber_ID   : PTR_NEW(), $
    Absorber_Units : PTR_NEW(), $
    Level_Pressure : PTR_NEW(), $
    Pressure      : PTR_NEW(), $
    Temperature   : PTR_NEW(), $
    Absorber      : PTR_NEW(), $
    Cloud         : PTR_NEW(), $
    Aerosol       : PTR_NEW() }
END
```

and

```
PRO CRTM_Cloud__Define
void = { CRTM_Cloud, $
    n_Layers      : 0L, $
    Type          : 0L, $
    Effective_Radius : PTR_NEW(), $
    Effective_Variance : PTR_NEW(), $
```

Water_Content : PTR_NEW() }
END

Subject: Re: How to get numbers into passed structure elements
(pass-by-value/reference problem).
Posted by [Paul Van Delst\[1\]](#) on Thu, 28 Feb 2008 15:19:33 GMT
[View Forum Message](#) <> [Reply to Message](#)

David Fanning wrote:

> Paul van Delst writes:

>
>> I'll repeat my comment in my reply to David Fanning's post in this thread:
>> I find it a fundamental flaw of IDL that the argument passing mechanism (reference
>> or value) is so exposed to the user that they have to tailor their code to prevent
>> data loss.

>
> I find this equivalent to dynamic tying and structuring:
> powerful, but dangerous. Like keeping a loaded 357 Magnum
> in the drawer of the night stand. In the right hands ... etc.

Hmm. As a colleague pointed out to me just yesterday, if you don't have a 357 Magnum
(loaded or otherwise) in the drawer of your nightstand there is no associated danger.

:o)

However, I do think you have a point with regard to the dynamics of IDL. Maybe we could
employ a slightly less aggressive analogy? Such as the old omelette and breaking eggs
chestnut? :o)

cheers,

paulv

Subject: Re: How to get numbers into passed structure elements
(pass-by-value/reference problem).
Posted by [Paul Van Delst\[1\]](#) on Thu, 28 Feb 2008 15:20:25 GMT
[View Forum Message](#) <> [Reply to Message](#)

David Fanning wrote:

> David Fanning writes:

>
>> I find this equivalent to dynamic tying and structuring:
>
> Whoops, should be "typing", of course.

Either one works.

:o)

>

> Cheers,

>

> David

>
