Subject: Named and Anonymous structures ?
Posted by robk on Fri, 06 Oct 1995 07:00:00 GMT
View Forum Message <> Reply to Message

I could use a refreshing idea to get me out of a problem related to
structures in IDL:

I must read binary data with variable format and had no problem to
write an extraction programme that reads one file per IDL session,
but my main problem now is that named structures cannot be changed once
defined, and at the same time, anonymous structures (which can be
changed) cannot be part of other anonymous structures.

At first I wrote a procedure that works fine on one file:
Below is a simplified version, reduced to the essential elements of
the problem.

```
 ..................................................................... ......
 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;; ;;;;;;
PRO an_extraction.pro, struc_back_to_main
 openr,unit,'huge_satellite_data_file_with_complex_structure' ,/get_lun
struc1 = {named_struc                    , $  ; first part of data, always
                (1):          0        , $  ; same format
                (2):          0L       , $
                (...):        0.       , $
                (5):    string(0)        $
        } ;
readu,unit,struc1

struc2 = {another_named_struc              , $
                (1):          0L      , $  ; always same format
                (2):          0L      , $
                (...):        0.      , $
                number_of_rows:0       , $
                nmbr_o_records:0        $
        };
readu,unit,struc2    ; get values for number_of_rows and nmbr_o_records



struc3 = { made_to_measure_struc,
                first_array: fltarr(number_of_rows, 10), $ ; dynamic
                another_arr: intarr(number_of_rows, 25), $ ; arrays
        };

array_of_struc3s = replicate({file_specific_struc},struc2.nmbr_o_records)
readu,unit,array_of_struc3s
free_lun,unit
struc_back_to_main = { pret_a_porter_struc       $
```

```
                first: struc1          , $
                secnd: struc2          , $
                  (3): ......          , $
                  (4): ......          , $
                fifth: array_of_struc3s , $
                  (6): ......          $
         };
end
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;; ;;;;;;;;;;;;;;;
```

But here's my problem. I can make "struc_back_to_main" an anonymous structure, and this allows me to read files with different "nmr_o_records" but as soon as there's a file with a different "number_of_rows" I'm in trouble: "struc3" should then also be anonymous, but unfortunately anonymous structures cannot be part of other anonymous structures as I've just found out after writing several main programs that call this procedure :-(
Once this struc_back_to_main is passed to the caller, struc3 can be destroyed, as far as I'm concerned, but the only way I've managed to do that, is by exiting the IDL session.

Is there another way to destroy named structures?
Can the TEMPORARY function help me out of this trouble?
I have an IDL version (4..) that's newer than my manual (3.1) , are there new tricks that could help me out, e.g. "C"-style pointers, things like "malloc" etc??

Any help is appreciated, thank you in advance,

```
    _/_/_/          _/
   _/   _/         _/
  _/_/_/    _/_/   _/_/
 _/   _/  _/  _/  _/ _/  Koopman, Air Research Lab., RIVM,The Netherlands
_/    _/  _/_/   _/_/
```

Subject: Re: Named and Anonymous structures ?
Posted by hahn on Sun, 05 Oct 1997 07:00:00 GMT
View Forum Message <> Reply to Message

On Fri, 6 Oct 1995 17:09:15 GMT, you wrote:

Hi Rob,

If I'm not totally mistaken you could use the procedure create_struct to nest anonymous structures. This wasn't possible before IDL 3.6.1 !

Why do you define a structure inside a procedure while it is returned via the parameter list. Can't you define it before calling the procedure ? This would allow to use different code to access fields available in one structure while skipping the code for a different structure at a different call. The following worked for me:

```
pro testit, rqm, s

s.n = 100

print, rqm, s.n, size(s.fx1)

; Note: The variable s.fl1 is only present when rqm=1
;
if  rqm eq 1  then  print, -1, -1, size(s.fl1)

end
;
 s1 = { ABC, n:100L, fx1:LonArr(100) }
 s2 = { XYZ, n:100L, fx1:LonArr(100), fl1:fltarr(100) }

 a1 = { ABC }
 a2 = { XYZ }

 testit, 0, a1
 testit, 1, a2

end
```

Maybe I missed the point ...

Hope, this helps
Norbert Hahn
