I am trying to read in files of a priori undefined organization within IDL.

The files are "tabular" binary files (an a priori unknown number of columns of an a priori unknown type with an a priori unknown number of rows). "a priori unknown" means documented in a file header.

What I wish to have is a procedure   myread,'myfile.name',s

which returns a structure s

This structure will have a number of members   s.keyname corresponding to keywords, and a member s.data.

The member s.data is created as: data:replicate(template,nrows)

where template is constructed building up a string containing a statement like the following and doing an "test=execute(statement)"

template= {record,TIME:0L, THETA:0L, PHI:0L, ENERGY:0 , undef:0B}
          ^^^^^^

where the names and types of the tags are constructed by my procedure according to information in the file itself. So later I can access the individual columns as s.data.time, etc.

I MUST use a named structure, otherwise I'm getting an error "cannot have an anonymous structure as a member of another structure").

This is fine the first time, or if I read more files with the same "record structure", but if I try to read a file with a different structure, I get an error % Wrong number of tags defined for structure: RECORD.

Is there any way I can REMOVE this structure from being defined (note that it does not correspond to any true variable in use), and re-use it again ?

PS
I'm using IDL 2.2.1 (and I'm not likely to update it for the time being)

```
 ---------------------------------------------------------------- -----------------
     A member of  G.ASS : Group for Astronomical Software Support
 ---------------------------------------------------------------- -----------------
Lucio Chiappetti - IFCTR/CNR Milano       |      U N I C U I Q U E
via Bassini 15 - I-20133 Milano - Italy   |      System  Manager
Internet: LUCIO@IFCTR.MI.CNR.IT           |        FORTUNAE SUAE
```

Decnet: IFCTR::LUCIO (39610::LUCIO) |
Bitnet: <please do not use any more> |
 --- ----------------

Subject: Re: undefining structures ?
Posted by thompson on Mon, 06 Dec 1993 22:30:57 GMT

View Forum Message <> Reply to Message

lucio@ifctr.mi.cnr.it (Lucio Chiappetti) writes:

> I am trying to read in files of a priori undefined organization within IDL.

> The files are "tabular" binary files (an a priori unknown number of columns
> of an a priori unknown type with an a priori unknown number of rows).
> "a priori unknown" means documented in a file header.

> What I wish to have is a procedure   myread,'myfile.name',s

> which returns a structure s

> This structure will have a number of members   s.keyname corresponding to
> keywords, and a member s.data.

> The member s.data is created as: data:replicate(template,nrows)

> where template is constructed building up a string containing a statement
> like the following and doing an "test=execute(statement)"

> template= {record,TIME:0L, THETA:0L, PHI:0L, ENERGY:0 , undef:0B}
>   ^^^^^^

> where the names and types of the tags are constructed by my procedure
> according to information in the file itself. So later I can access the
> individual columns as s.data.time, etc.

> I MUST use a named structure, otherwise I'm getting an error "cannot have
> an anonymous structure as a member of another structure").

> This is fine the first time, or if I read more files with the same "record
> structure", but if I try to read a file with a different structure, I get
> an error % Wrong number of tags defined for structure: RECORD.

> Is there any way I can REMOVE this structure from being defined (note that
> it does not correspond to any true variable in use), and re-use it again ?

> PS
> I'm using IDL 2.2.1 (and I'm not likely to update it for the time being)

There's only one way I know around this. One could, at least in principal,
create a new named structure variable every time. For example,

```
 PRO MAKE_MY_STRUC, TAGS, TEMPLATE
;
;  The input variable TAGS is a character string containing a list of tag
;  statements, e.g. 'TIME:0L, THETA:0L, PHI:0L, ENERGY:0 , undef:0B'.  The
;  output TEMPLATE is a structure with a unique name containing these tags.
;
;  The common block MY_STRUC_COMMON keeps track of the STRUC_NUM variable from
;  call to call.
;
 COMMON MY_STRUC_COMMON, STRUC_NUM
;
;  If STRUC_NUM hasn't been defined yet, then initialize it to zero.
;  Otherwise, increment it by one.
;
 IF N_ELEMENTS(STRUC_NUM) EQ 0 THEN STRUC_NUM = 0 ELSE $
  STRUC_NUM = STRUC_NUM + 1
;
;  Form the command to execute.  Each time this routine is called, the
;  structure name will be different.
;
 TEST = EXECUTE('TEMPLATE = {REC'+STRTRIM(STRUC_NUM,2)+','+TAGS+'}')
;
 END
```

I should warn you, however, that I haven't actually tried the above approach.
I think it should work, though.

In any case, it is clumsy. I don't see any reason why one shouldn't be able to
put anonymous structures within other structures. Let us hope that RSI hears
us and makes this change.

If anyone can think of a simpler or better way to accomplish what I and Dr. Chiappetti want to do, or if one of the latest upgrades to IDL solves or addresses this problem, please post details.

Bill Thompson