Subject: Re: STRUCT_ASSIGN
Posted by rivers on Sat, 22 Aug 1998 07:00:00 GMT
View Forum Message <> Reply to Message

In article <6rn2tknne1@hammer.msfc.nasa.gov>, mallors@crazyhorse.msfc.nasa.gov (Robert S. Mallozzi) writes:

> I have a structure defined as follows:
>
> a = { f1: 0, f2: { x: 0, y: 0}}
>
> I want to initialize the substructure f2
> in a subroutine by having an instance of the f2
> structure, then copying it field-by-field to
> a.f2. STRUCT_ASSIGN seemed ideal for this,
> except one problem - it doesn't work :-)
>
> Assuming I have initialized an instance of
> f2 (called data) with some values
>
> data = {x: 10, y: 20}

The problem is that data is NOT an instance of f2, but rather another anonymous structure, which IDL does not realize is the same as f2. If IDL did realize this then you could simply type:
a.f2 = data

There are 2 ways to get this to work:

1) Use named structures rather than anonymous ones.

```
a = { f1: 0, f2: {f2, x: 0, y:    0}}
data = {f2, x: 10, y: 20}
a.f2 = data
```

2) Create data from a

```
a = { f1: 0, f2: { x: 0, y: 0}}
data = a.f2 & data.x=10 & data.y=20
a.f2 = data
```


Mark Rivers (773) 702-2279 (office)
CARS (773) 702-9951 (secretary)
Univ. of Chicago (773) 702-5454 (FAX)
5640 S. Ellis Ave. (708) 922-0499 (home)
Chicago, IL 60637 rivers@cars.uchicago.edu (e-mail)

or:
Argonne National Laboratory (630) 252-0422 (office)
Building 434A (630) 252-0405 (lab)
9700 South Cass Avenue (630) 252-1713 (beamline)

Subject: Re: STRUCT_ASSIGN
Posted by Martin Schultz on Sun, 23 Aug 1998 07:00:00 GMT
View Forum Message <> Reply to Message

Robert S. Mallozzi wrote:
>
> Hi all,
>
> I have a structure defined as follows:
>
> a = { f1: 0, f2: { x: 0, y: 0}}
>
> [...]

> So, must I resort to field-by-field copy? The problem
> is that "a" is actually an object ("self"), which has
> some member variables that are large structures.
> A field-by-field copy would be tedious, as each
> structure has on the order of 30 fields, or so.
> I wanted to write a generic "set" method that will
> initialize the object's structures with some data.
> If I must do a field-by-field copy, I would then
> have to have several "set" methods, each of which is
> specialized for each of the different stuctures
> within the object.
>

As Mark points out, the problem here is that you are actually using two different anonymous structures. There has been a recent discussion in this newsgroup about a similar problem: If you type help,a.f2,data,/stru you will se something like :

```
** Structure <10304508>, 2 tags, length=4, refs=2:
   X               INT             0
   Y               INT             0
** Structure <10304708>, 2 tags, length=4, refs=1:
   X               INT            10
   Y               INT            20
```

The number in <> identifies the structure "type"...

> So, must I resort to field-by-field copy? The problem
> is that "a" is actually an object ("self"), which has
> some member variables that are large structures.
> A field-by-field copy would be tedious, as each

> structure has on the order of 30 fields, or so.
> I wanted to write a generic "set" method that will
> initialize the object's structures with some data.
> If I must do a field-by-field copy, I would then
> have to have several "set" methods, each of which is
> specialized for each of the different stuctures
> within the object.
>

What you describe here sounds like a typical application for pointers.
Nesting structures within structures within structures ... can usually
only be achieved with the help of these things. Here is an example:

```
  a = { f1:0, f2:ptr_new() }
  data = { x:10, y:20 }
  a.f2 = ptr_new(data,/NO_COPY)   ; careful: no_copy means
                                  ; "data" will be lost afterwards!

  help,*a.f2,/stru
```

You can test the contents of your sub-structure with tag_names():

```
  print,tag_names(*a.f2)
```

Hope, this gives a hint in the right direction,
Martin.

--

 --- -------
Dr. Martin Schultz
Department for Earth&Planetary Sciences, Harvard University
109 Pierce Hall, 29 Oxford St., Cambridge, MA-02138, USA

phone: (617)-496-8318
fax : (617)-495-4551

e-mail: mgs@io.harvard.edu
Internet-homepage: http://www-as.harvard.edu/people/staff/mgs/
 --- -------