Martin Rother wrote:
>   Hi Gurus,
>
>   finally something, where I don't know,
>   if it's odd idl syntax or not...
>
>   something about structures.
>
> FUNCTION test_struct
>
>   s = [{ m : [1L, 2L, 3L], n :1L }]
>
>   FOR i = 1L, 2L DO BEGIN
>   ;
>   s = [s, { m : [1L, 2L, 3L], n : i }]
>   ;
>   ENDFOR
>
>   return, s
>
> END
>
>   this creates an array of structures:
>
> (X). s = test_struct()
> (X). help,/struct,s
> ** Structure <fa330>, 2 tags, length=16, data length=16, refs=1:
>   M   LONG   Array[3]
>   N   LONG   1
> (X). print, n_elements(s)
>   3
> (X).
>
>   so far ok, I guess. but
>
> (X). help,s.m
> <Expression>   LONG   = Array[3, 3]
> (X). print, s.m
>   1   2   3
>   1   2   3
>   1   2   3
> (X). print, s.m[0]
>   1   1   1
> (X). print, s.m[1]

```
>           2        2        2
>  (X). print, s.m[2]
>           3        3        3
>
>        is a *bit* confusing. isn't it?
>
>
>        best regards,
>        martin.
```

Hi,

Perhaps you are looking to get the entire m array of the ith element in vector s?

```
IDL> s = [{ m : [1L, 2L, 3L], n :1L }]
IDL> FOR i = 1L, 2L DO s = [s, { m : [1L, 2L, 3L], n : i }]
IDL> print, s[0].m
        1        2        3
```

Which is different than asking for the ith elements of the m array in ALL the structures in the vector s.

```
IDL> print, s.m[0]
        1        1        1
```

Or how about the ith element of the m array in the jth element of the vector s?

```
IDL> print, s[0].m[0]
        1
```

I agree it can be confusing (wait till you have pointers in there!) but is very handy sometimes, too.

Ben