
Subject: Re: structure into structre

Posted by [Paul Van Delst\[1\]](#) on Fri, 12 Oct 2007 18:12:32 GMT

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

natha wrote:

> Hi Gurus,

>

> I want to declare an object structe with 4 structures inside. Is it

> possible ?

>

> My class has 4 structures with 10 to 15 variables.

> I declare the object like this example:

>

> PRO objectA__define

> struct = { objectA, \$

> struct_a={ a1:0L, \$

> a2:0L \$

> }, \$

> struct_b={ b1:0L, \$

> b2:0L \$

> } \$

> }

> END

>

> But IDL can't compile...

> Maybe I need to declare everyone as a pointer and redefine every

> structure in the Init function but I don't want to do this.

With

```
PRO objectA__define
```

```
void = { objectA, $
```

```
    struct_a: { a1:0L, a2:0L }, $
```

```
    struct_b: { b1:0L, b2:0L } $
```

```
}
```

```
END
```

I get:

```
IDL> .run objecta__define
```

```
% Compiled module: OBJECTA__DEFINE.
```

```
IDL> x={objecta}
```

```
% Named structures can't contain anonymous structure members
```

```
% Execution halted at: OBJECTA__DEFINE 3
```

```
/export/lnx374/wd20pd/scratch/objecta__define.pro
```

```
%           $MAIN$
```

So what about something like:

```

PRO objectA__define
  void = { objectA, $
          struct_a: {struct_a, a1:0L, a2:0L }, $
          struct_b: {struct_b, b1:0L, b2:0L } $
        }
END

```

```

IDL> x={objecta}
% Compiled module: OBJECTA__DEFINE.
IDL> help, x, /struct
** Structure OBJECTA, 2 tags, length=16, data length=16:
  STRUCT_A    STRUCT  -> STRUCT_A Array[1]
  STRUCT_B    STRUCT  -> STRUCT_B Array[1]

```

.....OR.....

```

PRO mystruct__define
  void = { mystruct, $
          x1:0L, $
          x2:0L $
        }
END

```

```

PRO objectA__define
  void = { objectA, $
          struct_a: {mystruct}, $
          struct_b: {mystruct} $
        }
END

```

```

IDL> .run objecta__define
% Compiled module: MYSTRUCT__DEFINE.
% Compiled module: OBJECTA__DEFINE.
IDL> x={objecta}
IDL> help, x, /struct
** Structure OBJECTA, 2 tags, length=16, data length=16:
  STRUCT_A    STRUCT  -> MYSTRUCT Array[1]
  STRUCT_B    STRUCT  -> MYSTRUCT Array[1]

```

Of course, you would define the various XXX__define procedures as needed if each is a different structure.

cheers,

paulv
