Subject: Re: Recursive Objects
Posted by Martin Downing on Tue, 09 Oct 2001 15:51:15 GMT
View Forum Message <> Reply to Message

"Francesco Belletti" <guenhwyvar@libero.it> wrote in message
news:e7b4ed30.0110090547.79d8e345@posting.google.com...
> Hello,
> I'm a very beginner with IDL so I'm sorry if my question is too
> stupid.
> I need to store a object reference in an object structure:
>
> pro my_obj__define
> struct = {my_obj, another_obj:OBJ_NEW()}
> end
>
> After another_obj is zeroed during object creation, it couldn't no
> more be used as an object reference!!
> I've tried to redeclare it
>
> another_obj=OBJ_NEW()
>
> but the problem remains.
> It's an IDL bad limit, or my error?
>
> Thank you,
>
> Francesco Belletti

Francesco,
Objects are handled using OBJREFs, which are very much like pointers. What you have declared is a reference to an object, ie an OBJREF, which you have set to the NULLOBJ. What you need to do next is to point variable to a valid object, eg in your INIT method:

```
----------------------------------------
function my_obj::init, other = other
 ; see if other is a valid object
 if obj_valid(other) then begin
   ; ok so set pointer to it
   self.another_obj = other
   print, "setting another"
 endif
 return, 1
end

function my_obj::GET_ANOTHER
 return, self.another_obj
end
```

```
pro my_obj__define
   struct = {my_obj, another_obj:OBJ_NEW()}
end
-----------------------------------------------
```

Now when you create some objects:

IDL> obj_a = obj_new('my_obj')

IDL> obj_b = obj_new('my_obj',other =obj_a)

setting another

IDL> help, obj_a

OBJ_A OBJREF = <ObjHeapVar1(MY_OBJ)>

IDL> help, obj_b

OBJ_B OBJREF = <ObjHeapVar2(MY_OBJ)>

IDL> help, obj_b->get_another()

<Expression> OBJREF = <ObjHeapVar1(MY_OBJ)>

The second call to obj_new has not created another copy of obj_a but has stored its address (OBJREF) within obj_b.

Hope this helps, good luck with IDL!

Martin