

Hi,
I am experiencing a problem in IDL 5.4.

I create an object called 'my_object', and define it in a file called
'my_object__define.pro'

I do the same thing with 'other_object' , and
'other_object__define.pro'.

Both of these objects compile and i can create variables with them.
They both have 'init' functions that do basic initialization stuff.

I know that the 'init' functions are getting called because I can
put...

```
print,'i am here ...'
```

...statements in the init functions, and I will see the output when
i initialize an object of that type.

...

Now, my problem is that if I want 'my_object' to have an instance of
'other_object' as a member object, the init function(of
'other_object')
doesn't get called.

To eleborate...(snipping irrelevant stuff)

```
;-----  
;my_object__define.pro
```

```
pro my_object__define
```

```
    foo: obj_new('other_object')
```

```
end ; my_object
```

```
;-----  
;other_object__define.pro
```

```
function other_object::init
```

```
    print, ' i am here!'
    return, 1
end ; init
```

```
pro other_object__define
```

```
    ;object definition stuff here
```

```
end ; other_object
```

```
;------
```

When type ' x = obj_new('other_object') ' on the command line, I see the ' i am here!'.

When type ' y = obj_new('my_object') ' on the command line, I DON"T see the ' i am here!'.

Why is this?

I don't get any errors otherwise, and according to the documentation on 'obj_new' it seems it should call the 'init' function in BOTH circumstances.

Any ideas greatly appreciated,

Paul Wisehart
wisehart@runbox.com
