Subject: Passing objects and parenthesis... doing something wrong here?
Posted by Helder Marchetto on Wed, 09 Jan 2013 12:17:51 GMT
View Forum Message <> Reply to Message

Hi,
I've made a program that makes extensive use of objects. I ran perfectly on one pc and is now giving me problems on the laptop (laptop with IDL 8.2... most likely also the pc, but I can't bet until I have a look at it).
It basically has to do with the way I address objects. On the laptop I need some extra parenthesis that I didn't need before. To make things easier to understand I wrote some code that reproduces the error. Below is the code that generates the error on the laptop, where the error is:

```
% Expression must be a structure in this context: <OBJREF
(<ObjHeapVar1(MYFIRSTTESTOBJECT)>)>.
% Execution halted at: MYSECONDTESTOBJECT::INIT    6 F:\Test\testobjects.pro
%                OBJ_NEW
%                MYFIRSTTESTOBJECT::INIT   18 F:\Test\testobjects.pro
%                OBJ_NEW
%                TESTOBJECTS       30 F:\Test\testobjects.pro
%                $MAIN$
```

Before I go around adding parenthesis in the code, can anybody offer an explanation for this difference?
Tonight late I will try the code on the pc and see how it runs there, but it would be nice to get the code to run as it is (without the extra parenthesis).

Thanks,
Helder

PS: code below... too lazy for destroy methods... clean up with a .reset_session after running!

```
FUNCTION MySecondTestObject::Init, FirstObj=FirstObj
self.FirstObj = FirstObj
HELP, self.FirstObj, /STRUCTURES
HELP, self.FirstObj, /OBJECTS
PRINT, (self.FirstObj).Var1, (self.FirstObj).Var2   ;this works on the laptop
PRINT, self.FirstObj.Var1, self.FirstObj.Var2       ;this gives an error on the laptop
RETURN, 1
END

PRO MySecondTestObject__Define
void={MySecondTestObject,$
    FirstObj:Obj_New()}
END

FUNCTION MyFirstTestObject::Init
self.Var1 = 1
self.Var2 = 1
```

```
obj = OBJ_NEW('MySecondTestObject',FirstObj=self)
RETURN, 1
END

PRO MyFirstTestObject__Define
void={MyFirstTestObject,$
     Var1:0,$
     Var2:0}
END


PRO TestObjects
obj = OBJ_NEW('MyFirstTestObject')
END
```
