Subject: Object Funkiness
Posted by J.D. Smith on Wed, 30 Jul 1997 07:00:00 GMT
View Forum Message <> Reply to Message

OK OOP experts. Here's a conundrum...

here's a procedure to test dynamic binding of methods in IDL...

```
pro testdb
   a=obj_new('eeke','a')
   b=obj_new('eeke','b')
   list=[a,b]
   ran=fix(randomu(sd) ge .5)
   list[ran]->Message, ran, 1-ran
   obj_destroy,list
   return
end
```

and the file eek__define.pro contains:

```
pro eeke::Message, ran, ran2
   print,self.val,' random: ',ran, ran2
   return
end
function eeke::Init,val
   self.val=val
   return,1
end
pro eeke__define
   struct={eeke,val:''}
   return
end
```

When I run it I get

```
IDL> testdb

   list[ran]->Message, ran, 1-ran
                          ^
% MESSAGE: Incorrect number of arguments.
  At: /u/jdsmith/idl/pro/mylib/testdb.pro, Line 6
% Compiled module: TESTDB.
% Attempt to call undefined procedure/function: 'TESTDB'.
% Execution halted at:  $MAIN$
```

If I change the name of Message to something else, or take away the arguments to message in both the method definition and the call, the error disappears. Is 'message' somehow different than other names? I thought it could be shadowing the idl built-in message, and screwing up the dynamic binding, but I've tried tons of other built-in names (like 'print', 'xmanager', etc.) with nary a problem. Perhaps 'Message' is some IDL internal method for all objects... anyway it is very troublesome, and I'd like to figure out the cause. Any help would be greatly appreciated.

JD

P.S. Except for this problem, dynamic binding seems to work well.

Subject: Re: Object Funkiness
Posted by J.D. Smith on Mon, 04 Aug 1997 07:00:00 GMT
View Forum Message <> Reply to Message

Phil Williams wrote:
>
> J.D. Smith wrote:
>>
>> J.D. Smith wrote:
>>>
>>> OK OOP experts. Here's a conundrum...
>>>
>>> here's a procedure to test dynamic binding of methods in IDL...
>>>
>>> pro testdb
>>> a=obj_new('eeke','a')
>>> b=obj_new('eeke','b')
>>> list=[a,b]
>>> ran=fix(randomu(sd) ge .5)
>>> list[ran]->Message, ran, 1-ran
>>> obj_destroy,list
>>> return
>>> end
>>>
>>> and the file eek__define.pro contains:
>>>
>>> pro eeke::Message, ran, ran2
>>> print,self.val,' random: ',ran, ran2
>>> return
>>> end
>>> function eeke::Init,val
>>> self.val=val

```
>>>     return,1
>>>  end
>>>  pro eeke__define
>      ^^^^^^^^^^^^^^^^^
>  The __define method should be a FUNCTION.  I'm sure once this is done
>  everything should work fine.
>
>>>     struct={eeke,val:''}
>>>     return
>>>  end
>>>
>
>  phil
```

I think perhaps you're thinking of the INIT class method -- __defines
are in fact procedures, I believe. It's really a problem with the
compiler parsing method names incorrectly for methods which shadow IDL
builtin's, with some conditions affecting whether the error will occur.
I've alerted RSI Tech Support and they're working on it, but for now,
avoid method names such as 'Message', 'Wait', 'Wset', and likely several
more.

JD
