Subject: Re: pointer to object confusion (C++ programmer, IDL n00b)
Posted by Aram Panasenco on Tue, 04 May 2010 15:01:15 GMT
View Forum Message <> Reply to Message

Matt Francis wrote:
> Hi All, I'm new to IDL but am a reasonable C++ coder. I'm trying to
> set up some object classes in IDL and am having some trouble.
>
> I can create custom objects and use them okay, but I can't seem to get
> a custom object to use another custom object within it. So say I have
> defined allready a class FOO, and now I want another class FOO2 which
> stores within it an instance of FOO:
>
> PRO FOO2__DEFINE
> struct = {FOO2, ..., FOO:<???>, ...}
> END
>
> What I want to know is what goes in<???>. I can't use OBJ_NEW because
> I don't know yet what arguments will be fed to FOO when it gets
> instantiated in some method of FOO2. I tried using simply OBJ_NEW() to
> get a null pointer, but then when I try something like
>
> PRO FOO2::some_method
> ...
> FOO = OBJ_NEW('FOO',[ARGS])
> END
>
> I get an error. I've tried various combinations of *FOO etc to try and
> get the above to work without success.
>
> Can anyone help me? I'm probably thinking too much like a C++
> programmer here, but I can't see that I'm trying to do something crazy
> so there must be a way to do this. Any hints?

Hi Matt!
What's going on here is: In IDL it takes two routines to initialize an object. One is CLASSNAME__DEFINE, and the other is CLASSNAME::INIT. The CLASSNAME__DEFINE procedure simply creates the object's class structure. All properties of the object are initially either zeroes, or null strings, or empty objects, or etc. CLASSNAME::INIT initializes the properties. All arguments passed in [ARGS] in FOO = OBJ_NEW('CLASSNAME', [ARGS]) are arguments to the INIT function. The INIT function initializes the object's properties and returns 1 if everything went A-ok (and 0 if the object couldn't be initialized). For example, the CLASSNAME file could look something like this:

function CLASSNAME::INIT, arg1, arg2, arg3=arg3

```
  if ( (n_elements(arg1) eq 0) or (n_elements(arg2) eq 0) ) then $
    return, 0

  self.arg1 = arg1
  self.arg2 = arg2
  if (n_elements(arg3) gt 0) then begin
    self.arg3 = arg3
  endif else begin
    self.arg3 = Obj_New('SomeClass')
  endelse

  return, 1
end

pro CLASSNAME__DEFINE
  struct = {CLASSNAME, $
    arg1:0, $
    arg2:'', $
    arg3:Obj_New() $
  }
end
```

And then you could create a new CLASSNAME object:

```
myObject = Obj_New('CLASSNAME', 2, 'Custom String')
  or
myObj2 = Obj_New('SomeClass', arg1,arg2,arg3)
myObject = Obj_New('CLASSNAME', 2, 'Custom String', arg3 = myObj2)
```

Hope that helped you understand object creation better

~Aram Panasenco
