Subject: Re: Recursive Objects
Posted by Karl Schultz on Wed, 10 Oct 2001 14:52:01 GMT
View Forum Message <> Reply to Message

"Mark Hadfield" <m.hadfield@niwa.cri.nz> wrote in message news:001501c150fd\$d00b6440\$d938a8c0@Hadfield...

- > From: "Karl Schultz" <kschultz@researchsystems.com>
- >>> pro my\_obj\_\_define
- >>> struct = {my\_obj, another\_obj:OBJ\_NEW()}
- >>> end
- >> ....
- >> Also note that you do not need to explicitly destroy this instance of
- >> "SomeOtherClass" in the my\_obj::Cleanup method. Since the objref is in > the
- >> class struct, IDL will find and destroy it for you.
- > Not in my ovnorioned
- Not in my experience!
- > When IDL destroys a my\_obj instance it will erase an object reference stored
- > in the my\_obj class structure. It will \*not\* destroy the heap variable that
- > this reference refers to. So to avoid a memory leak you need
- > pro my\_obj::cleanup
- > obj\_destroy, self.another\_obj
- > ; Other cleanup tasks
- > end

Oops, you are right. The object I was thinking about was a container object (inherits from a container object) where the Init method added an object to the container right after creating it. So, when the container gets destroyed, the added object gets destroyed too. Sorry about that.