Subject: Re: IDL objected oriented question
Posted by btt on Tue, 08 Apr 2003 16:53:35 GMT
View Forum Message <> Reply to Message

Sabir Pasha wrote:

> Basically
>
> define = { ClassA,  $
>
>    ObjectB:  Obj_New()}
>
> END
>
> the object gets defined in
> ObjectB =  Obj_New("ClassB")
>
> And somewhere we define ObjectA
>
> ObjectA = Object_New("ClassA)
>
> and now in an event handler far far away
>
> Sinfo.objectA.objectB->member function
>
> doesnt' work because we cannot access Objects A's member variables
> only member functions.

Hi,

I think you have a couple of choices.  The easiest and 'safest' is to provide access to the properties (member variables) of ObjA via the GetProperty method.

ObjA->GetProperty, ObjectB = ObjB

ObjB->DoMyMemberMethodThing

You would include, in this case, an OBJECTB = OBJECTB keyword in ObectA's GetProperty method.

```
PRO OBJECTA::GetProperty, OBJECTB = OBJECTB, etc=etc

 ;did the caller ask for objectB?
If ARG_PRESENT(ObjectB) then ObjectB = ObjectB

;other stuff
```

END ; GetProperty

This does make you go through two (or more programmatic steps) but it does the job nicely.

You know, if object A is behaving like a container object, then perhaps you could make OBJECTA inherit the properties and methods of IDL_CONTAINER? Then you would have easy access to all of the objectBs, objectCs, etc. Makes cleanup and transport issues a snap. And if you like that idea, try using Martin Schultz's flavor of IDL_CONTAINER (called MGS_CONTAINER) which has a handy method for finding contained objects by name (assuming each object contained has a name, of course.)

Cheers,
Ben

---