Subject: Re: do I really need to use loops on objects?
Posted by Brad Gom on Wed, 01 Dec 1999 08:00:00 GMT
View Forum Message <> Reply to Message

philaldis@yahoo.com wrote:

> I don't think that this isn't particularly consistent with the IDL
> philsophy. I can see what you're saying but how would go about
> implementing your suggestion being as in your case the objects may be
> all of the same type, but in other cases the objects may all be
> different. What would IDL do then when some of the objects do have the
> method called and others don't?
>

In the case where an object in an array didn't have the required method,
IDL could spit out an error to the effect of:
%Attempt to call undefined method: 'OBJECT::METHOD' when a method was
called on a whole array. It would be up to the programmer to make sure the
array was filled with the appropriate class of objects.
At any rate, I still have to make sure all the objects in my array have the
required method when I use a for loop on an object array!

> What is more consistent with IDL philosophy is the fact that a
> procedure like Obj_Destroy() can work on an entire array of object
> references so you can destroy a whole bunch of them in one go. However
> IDL does not continue with this fully. What's always riled me is the
> fact that Obj_Class() does not work on a objArr and you can't get it to
> return a string array with the object's classes.

I'll agree with that.

> I'm sure lots of people will disagree but I think on this occaison IDL
> is correct.

Well, I guess I am a disagree-er. The fact that after learning the basics
of IDL I would intuitively try to use object arrays in the same style that
I use all the other array types, suggests to me that there is an
inconsistency. Of course I have the same gripe about pointer arrays, but
then I'm used to programming in C, where I am allowed to crash the computer
in all sorts of creative ways by mis-casting variables.

Thanks for the opinion,

Brad Gom

Subject: Re: do I really need to use loops on objects?
Posted by Craig Markwardt on Wed, 01 Dec 1999 08:00:00 GMT
View Forum Message <> Reply to Message

bjackel@phys.ucalgary.ca writes:

> Brad Gom wrote:
>>
>> It seems odd to me that I can't treat object arrays with the same
>> elegance that is possible with all other array types. If I have a large
>> list of objects of the same type, and I want to call the same method on
>> each of them, do I really have to use a for loop? Wouldn't it be more
>> consistent with the IDL philosophy to write: object_array->method()
>> instead of: for i=0,10 do object_array[i]->method()
>
> I agree completely, but most people on this group don't appear to.
>
> The counter argument seems to be that an object array can contain
> different kinds of objects, so you can't be sure that any particular
> method will work for every element of the array. While true, this
> seems (to me) to be a reason why object arrays should contain only
> similar elements *JUST LIKE EVERY OTHER IDL ARRAY TYPE*. If you
> want different kinds of objects wrapped up together then use an
> array of pointers, or a structure with different object members.

One of the points of object oriented programming is *polymorphism*.
Which means, in short, that the same function can be applied to
different object types (possibly with differing implementations for
each). That's the great idea of keeping a array of heterogeneous
objects around.

So, object_array->method() *should* call "method" for each object in
object_array, irregardless of the type of the object. If one of the
objects doesn't define "method" then an exception should be raised.
[Following Smalltalk or Objective C there should be some way to query
an object to see if it responds to a particular method].

Not that I use objects much, but this is my sense of rightness.

Craig

--
-- --------------
Craig B. Markwardt, Ph.D. EMAIL: craigmnet@cow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
-- --------------

Subject: Re: do I really need to use loops on objects?
Posted by bjackel on Wed, 01 Dec 1999 08:00:00 GMT
View Forum Message <> Reply to Message

Brad Gom wrote:
>
> It seems odd to me that I can't treat object arrays with the same
> elegance that is possible with all other array types. If I have a large
> list of objects of the same type, and I want to call the same method on
> each of them, do I really have to use a for loop? Wouldn't it be more
> consistent with the IDL philosophy to write: object_array->method()
> instead of: for i=0,10 do object_array[i]->method()

I agree completely, but most people on this group don't appear to.

The counter argument seems to be that an object array can contain different kinds of objects, so you can't be sure that any particular method will work for every element of the array. While true, this seems (to me) to be a reason why object arrays should contain only similar elements *JUST LIKE EVERY OTHER IDL ARRAY TYPE*. If you want different kinds of objects wrapped up together then use an array of pointers, or a structure with different object members.

However, it is unlikely that RSI will make any significant change to object arrays, so we might as well get used to more loops. Ugh.

Brian

Subject: Re: do I really need to use loops on objects?
Posted by philaldis on Wed, 01 Dec 1999 08:00:00 GMT
View Forum Message <> Reply to Message

I don't think that this isn't particularly consistent with the IDL philsophy. I can see what you're saying but how would go about implementing your suggestion being as in your case the objects may be all of the same type, but in other cases the objects may all be different. What would IDL do then when some of the objects do have the method called and others don't?

What is more consistent with IDL philosophy is the fact that a procedure like Obj_Destroy() can work on an entire array of object references so you can destroy a whole bunch of them in one go. However IDL does not continue with this fully. What's always riled me is the fact that Obj_Class() does not work on a objArr and you can't get it to return a string array with the object's classes.

I'm sure lots of people will disagree but I think on this occaison IDL

is correct.

Cheers,
Phil

Phil Aldis,
Gonville and Caius College,
Cambridge University,
CB2 1TA.
E-Mail : philaldis@yahoo.com

In article <38447341.28FBAACB@hotmail.com>,
Brad Gom <b_gom@hotmail.com> wrote:
> It seems odd to me that I can't treat object arrays with the same
> elegance that is possible with all other array types. If I have a
large
> list of objects of the same type, and I want to call the same method
on
> each of them, do I really have to use a for loop? Wouldn't it be more
> consistent with the IDL philosophy to write: object_array->method()
> instead of: for i=0,10 do object_array[i]->method()
>
> I'm just getting back into IDL after a short hiatus, and back to
object
> programming in particular.. is there something I've missed?
>
> Brad
>
>