Subject: object memory management
Posted by Anonymous on Wed, 12 Nov 2008 17:31:43 GMT
View Forum Message <> Reply to Message

Originally posted by: Demitri

Quick on the heels of my previous question about empty arrays... I have a question about memory management.

Let's say I have a function that will return an array, but as it can be empty, I'd like to return an IDL_Container instead. No problem:

```
FUNCTION f
 container = NEW_OBJ('IDL_CONTAINER')
 container->add, NEW_OBJ('my_obj')
 container->add, NEW_OBJ('my_obj')

 return, container
END
```

(Let's ignore the memory management of the 'my_obj's for the moment.) Another method calls this and gets the container, but now the responsibility to destroy that object is in the hands of the calling routine, where it's not obvious (or maybe depending on the type it is?) that it will need to be freed by hand.

<Mac programmers only>
In Obj-C, this problem solved by the autorelease / retain messages, which of course IDL doesn't have. But that's the first thing I thought of.
</Mac programmers only>

Is this something that should be published in my class' API and the responsibility is passed to anyone using the function? It seems that calling OBJ_DESTROY will also destroy the objects within the container, and I may not want that. Should I ignore it and call HEAP_GC occasionally (*cough*hack!*cough*)? What is the IDL convention here?

Cheers,

Demitri