Subject: Why are objects global?
Posted by bjackel on Tue, 18 Jul 2000 07:00:00 GMT
View Forum Message <> Reply to Message

Subject line says it all.

I've been playing with objects (not object graphics) for a
while now, and can see how they might be useful. However,
having to call "obj_destroy" manually at the end of functions
and routines has been the source of many headaches.

For example, consider the procedure below. It creates
a "vector3" object which is used for various useful
calculations, but is of no further use outside "example".
Once we exit "example", variables "a" and "b" are automatically
cleaned up. However, "c" gets leaked, unless we explicitly
destroy it.

Previously, variables in routines would be local unless
1) linked to a parameter or keyword
2) placed in a common block
3) attached to a pointer

Why must objects be any different?

I would argue that they should be treated the same as
any other variables: cleaned up at the end of a procedure
or function, unless one of the three conditions mentioned
above is met. The current behaviour adds complexity to
IDL, with no obvious advantages.

Brian

```
;=================================================
 PRO example

  a= 1
  b= FLTARR(3)
  c= OBJ_NEW('vector3')

  <<hypothetical complicated useful code>>

 RETURN
 END
```