Subject: Re: objects, crashes, and negative memory of my Posted by Paolo Grigis on Fri, 01 Dec 2006 08:51:02 GMT

View Forum Message <> Reply to Message

```
Pete Warner wrote:
```

- > Hopefully one day I'll join the crowd of you helpful people who answer
- > guestions instead of coming up with them, but that day is not today.

>

- > I've got an object that should hold and organize a large number of
- > arrays and pointers and arrays of pointers and then clean them up when
- > I'm done. I know I have enough memory for the data in question because
- > I can load it without using pointers. I want the objects and pointers
- > for flexibility and to make the overall program easier to work with.

>

- > The problem is that I'm crashing IDL. I create the object, and then
- > destroy it to check for leaks. Using help, /heap_variables I show 0
- > pointers and objects. Then if I create and destroy it again IDL locks
- > up or crashes to desktop with some memory errors. I figured there must
- > be crazy leaking going on but I can't find it.

>

- > 1. Restart computer
- > 2. Load IDL
- > 3. Load project with objects
- > 4. Compile project
- > 5. IDL> print, memory(/I64, /current)
- 845185 >
- > 6. IDL> test = obj_new('mtgv_lodat', 'model.txt',
- > 'c:/rsi/projects/gicvu idl/system models/')
- > 7. IDL> print, memory(/I64, /current)
- 412234 >
- > 8. IDL> obj destroy, test
- > 9. IDL> print, memory(/I64, /current)
- -1077816

Maybe is not negative, but just overflowing for 18446744073708473800LL... now that's a nice rig you have, just one megabyte short of 4 exabytes ;-)

Ciao, Paolo

- If I repeat the process starting at 5 IDL blows up after a repetition
- > or two. If anyone has had this problem before and knows a solution I'd
- > be grateful. I couldn't find a solution searching the board, the IDL
- > help, or the ITTVIS IDL memory allocation FAQs.

> On the other hand, if anyone wants to buy a program that uses negative

> memory this is your lucky day.

Page 2 of 2 ---- Generated from comp.lang.idl-pvwave archive