Subject: IDL 8.0 garbage collector issue.
Posted by Bubba on Wed, 20 Oct 2010 12:47:35 GMT

View Forum Message <> Reply to Message

I have found something that doesn't seem quite right to me with
regards to the IDL 8.0 garbage collector.

If you have a function that returns a pointer array and you try to
tuck that call into another function call (n_elements for instance)
the pointer gets freed before the second function call completes.
Here is an example program that I wrote to show my problem...

```
function createPointerArray
  compile_opt idl2, logical_predicate

  data = ptrarr(3)
  data[0] = ptr_new(indgen(10))
  data[1] = ptr_new(indgen(10))
  data[2] = ptr_new(indgen(10))

  return, data
end

pro testGarbageCollector
  compile_opt idl2, logical_predicate

  ; The return from createPointerArray gets destroyed before the
n_elements call is complete
  print, n_elements(*(createPointerArray())[0]) ; Prints 0

  data = createPointerArray()

  print, n_elements(*data[0]) ; Prints 10
end
```

Am I missing something? Is this expected behavior? It seems to me
that you should be able to do this. It would just be extra unneeded
code to store the return into a variable just to be able to access
it. I use code like this a lot and I can't afford to update it
everywhere.

Please help!
Thanks