Subject: Re: Another issue with the garbage collector?
Posted by chris_torrence@NOSPAM on Fri, 12 Nov 2010 00:20:09 GMT
View Forum Message <> Reply to Message

Hi,

This is a bit different than the problem that was fixed in IDL 8.0.1. In the other bug (CR60104 - IDL 8.0 fails to dereference unnamed pointer expressions), it was a much simpler case:
print, *(ptr_new(5))

In your case, you are returning a pointer expression from an object that is about to be destroyed. To see this, if you put a breakpoint in your ::cleanup method, when the cleanup gets called, it is right on the "return" in ::getPixelData. So the self.pixelData gets freed during the "return".

I will go ahead and log this bug. In the meantime, I found a workaround. In your ::getPixelData, just put the result into another variable, like this:

```
function myImage::getPixelData
  compile_opt idl2, logical_predicate
  result = ptr_valid(self.pixelData) ? *self.pixelData : -1
  return, result
end
```

That way, when the self.pixelData gets whacked, you've got a "safe" copy in the return value.

As an aside, in your ::cropOutImage, why are you creating the "newData" pointer, only to free it a couple of lines later? I don't think this is actually saving any memory (unless this is just a code snippet).

Hope this helps.

Cheers,
Chris
ITTVIS