Subject: Another issue with the garbage collector?
Posted by Bubba on Thu, 11 Nov 2010 16:29:08 GMT
View Forum Message <> Reply to Message

I am having an issue with the garbage collector once again. I have an "image" object that contains a pointer to a two dimensional data array. It has two pretty simple functions, crop and get. The crop function crops the data and then returns another instance of the image class with the cropped data. The get function simply returns the two dimensional data array. If I try to call the crop function and the get function all in one line, it appears as thought the garbage collector cleans up my new image object before it gets to calling the get function. Here is an example...

```
; Image class definition

function myImage::init, pixelData
  compile_opt idl2, logical_predicate

  ;store image data
  self.pixelData = ptr_new(pixelData, /NO_COPY)

  return, 1
end

pro myImage::cleanup
  compile_opt idl2, logical_predicate

  if ptr_valid(self.pixelData) then ptr_free, self.pixelData
end

function myImage::cropOutImage, bottomLeftX, bottomLeftY,
suggestedWidth, suggestedHeight
  compile_opt idl2, logical_predicate

  ;get resized pixel data
  oldData = self.pixelData
  newData = ptr_new((*oldData)[bottomLeftX: bottomLeftX +
suggestedWidth - 1, $
                        bottomLeftY: bottomLeftY +
suggestedHeight - 1])

  ;create the new image
  newImage = obj_new('myImage', *newData)
  ptr_free, newData

  return, newImage
end
```

```
function myImage::getPixelData
  compile_opt idl2, logical_predicate

  return, ptr_valid(self.pixelData) ? *self.pixelData : -1
end

pro myImage__define
  compile_opt idl2, logical_predicate

  define = {myImage, pixelData: ptr_new()}
end

; test code
pro testGarbageCollector_Objects
  compile_opt idl2, logical_predicate

  data = findgen(200,100)

  workingImage = obj_new('myImage', data)

  newImage = workingImage->cropoutimage(0, 0, 10, 3)
  print, newImage->getPixelData() ; Prints the expected 10 x 3 array

  print, (workingImage->cropoutimage(0, 0, 10, 3))->getPixelData() ;
Crashes with an undefined variable
end
```

I thought this was fixed in IDL 8.0.1, am I missing something?
