
Subject: Re: Pointers and Objects

Posted by [Robbie](#) on Wed, 21 Mar 2007 02:13:12 GMT

[View Forum Message](#) <> [Reply to Message](#)

If you're concerned that a block of code is leaking heap variables, then you could try counting the number of objects on heap, before and after the code is executed. I use the following code in my unit tests:

```
; Example of using the rkbemptyheap procedure
rkbemptyheap, /mark ; Mark number of heap variables
; ...
a = ptr_new(1)
; ...
rkbemptyheap ; Check that the number of heap variables hasn't changed
since last mark
```

```
pro rkbemptyheap, DUMP=dump, MARK=mark
common rkbemptyheap, m_pointers, m_objects
on_error, 2
```

```
; Extract a dump of heap variables from the help procedure
help, /heap, OUTPUT=output
n_pointers = 0l
n_objects = 0l
for i=0l,n_elements(output)-1 do begin
  ppos = STRPOS( output[i], 'Pointer')
  if (ppos gt 0) then n_pointers = long(strmid(output[i],ppos+8))
  ppos = STRPOS( output[i], 'Object')
  if (ppos gt 0) then n_objects = long(strmid(output[i],ppos+8))
endfor
```

```
slevel = SCOPE_LEVEL()-1 ; The level of the calling procedure
```

```
; Ensure that the array has size equal to the level of the calling
procedure
if (n_elements(m_pointers) eq 0) then m_pointers = lonarr(slevel)
if (n_elements(m_objects) eq 0) then m_objects = lonarr(slevel)
if (n_elements(m_pointers) lt slevel) then m_pointers =
[m_pointers,lonarr(slevel-n_elements(m_pointers))]
if (n_elements(m_objects) lt slevel) then m_objects =
[m_objects,lonarr(slevel-n_elements(m_objects))]
```

```
if (keyword_set(mark)) then begin
; If marking then cache the number of pointers and objects at this
level
m_pointers[slevel-1] = n_pointers
m_objects[slevel-1] = n_objects
```

```
endif else begin
; If checking then see if the number of pointers or objects has
increased for this level
if (n_pointers gt m_pointers[slevel-1]) then begin
if (keyword_set(dump)) then $
for i=0l,n_elements(output)-1 do $
print, output[i]
if (m_pointers[slevel-1] gt 0) then $
message, LEVEL=-1, 'There is still at least one new pointer since
last mark' $
else $
message, LEVEL=-1, 'There is still at least one pointer on heap'
endif
if (n_objects gt m_objects[slevel-1]) then begin
if (keyword_set(dump)) then $
for i=0l,n_elements(output)-1 do $
print, output[i]
if (m_objects[slevel-1] gt 0) then $
message, LEVEL=-1, 'There is still at least one new object since
last mark' $
else $
message, LEVEL=-1, 'There is still at least one object on heap'
endif
m_pointers[slevel-1:*] = 0l ; Wipe all information about pointers
m_objects[slevel-1:*] = 0l ; Wipe all information about objects
endelse
end
```
