
Subject: Re: heap_gc behaviour
Posted by [davidf](#) on Tue, 04 Apr 2000 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

martin rother (rother@gfz-potsdam.de) writes:

> I detected now a 'heap_gc' behaviour, which I don't *understand*,
> but I'm *not sure* if this is a bug, an annoying feature or ...
>
> A call of 'heap_gc' inside a routine destroys a
> a heap-variable given as parameter.... ?
>
> Yes, I have read the reference manual... but as long I understand
> a call of 'heap_gc' inside a function is not forbidden.
>
> I add here a small demonstration: a 'batch', a tiny function which is
> called and the IDL messages.
>
> any hints?

Humm. Well, I don't know. But since you asked for "hints", which
I read as "advice", here is one.

Heap_GC is an admission of failure.

I'm of the opinion it should only be used after
you have closed the door to your office and sent
your office mate out for a couple of Danish. Putting
it in a program is, to me, a mark of a ... well, an
inexperienced programmer. :-)

Much better to learn how to free pointers and
destroy objects correctly than have to resort to
Heap_GC. About the only time it can legitimately
be used is during program development and before
you have added all your CATCH error handlers.

Cheers,

David

--

David Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438 E-Mail: davidf@dfanning.com
Coyote's Guide to IDL Programming: <http://www.dfanning.com/>
Toll-Free IDL Book Orders: 1-888-461-0155

hi!

just bothered by the spawn bug in IDL 5.3 (formatted I/O in bidirectional pipes crashes IDL...)

I detected now a 'heap_gc' behaviour, which I don't *understand*, but I'm *not sure* if this is a bug, an annoying feature or ...

A call of 'heap_gc' inside a routine destroys a heap-variable given as parameter.... ?

Yes, I have read the reference manual... but as long I understand a call of 'heap_gc' inside a function is not forbidden.

I add here a small demonstration: a 'batch', a tiny function which is called and the IDL messages.

any hints?
thanks in advance,
martin rother.

**** this3.pro

PRO this3, foo

```
;+=====
=====
;
;
; * SCRATCH: TEST HEAP_GC ONLY *
; Sparc, Solaris, IDL 5.3
;
; input:
;
;   foo    pointer?  to heap
;   count  integer   counter, modify foo if zero and
;               call heap_gc
;
;
;   00-Apr-04 Martin Rother (rother@gfz-potsdam.de).
;-=====
=====
```

ON_ERROR, 2

```
print, 'have: ' & help, *foo
print, 'call garbage collection: ' & heap_gc
print, 'have: ' & help, *foo
return
```

```

END
;=====
=====

*** test3

;=====
=====
;
; TEST3
;
; 00-Apr-04
;=====
=====

.r this3.pro

;-----
; simple heap var
;-----

foo = ptr_new(dist(12))

;-----
; CALL
;-----

print, 'top level:' & help, *foo

this3, foo

print, 'top level:' & help, *foo

;=====
=====

**** IDL output:

(X). @test3
top level:
<PtrHeapVar4>  FLOAT   = Array[12, 12]
have:
<PtrHeapVar4>  FLOAT   = Array[12, 12]
call garbage collection:
have:
% Invalid pointer: FOO.
% Error occurred at: THIS3          19

```

```
/home/mt/rother/idl/Test/heap_test/this3.pro
%          $MAIN$
% Execution halted at: $MAIN$
top level:
% Invalid pointer: FOO.
% Execution halted at: $MAIN$
(X).
```

--

Martin Rother (rother@gfz-potsdam.de) 0331/ 288-1272 Division 2.3
GeoForschungsZentrum Potsdam, Germany
