
Subject: Fatal Upp Core memory management (heap/malloc) interventions in AppKit/Cocoa?

Posted by [fudadmin](#) on Sun, 16 Jan 2011 19:53:07 GMT

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

While working with various upp Cocoa things I became very annoyed with my apps randomly/constantly crashing. Especially when custom menu is enabled. With a message:

which according to some sources:

Quote: is most likely to occur as a result of over-releasing or premature release of an object.

showing any of your code) is mostly useless. The debugger is equally useless as it has no way of identifying the deallocated object.

I've spent many hours during last 2 weeks trying to figure that out. At first I thought it was my Objective-C code at fault. But then, following my instincts, I tried the same apps without Upp::Core and they ran smoothly!

Because, I guess, without those top 4 lines intervention:

```
#0 0x1000110af in Upp::Heap::DivideBlock --crash
#1 0x100010eb4 in Upp::Heap::LAlloc
#2 0x100011bf3 in Upp::Heap::Alloc
#3 0x10000e5b7 in operator new
#4 0x7fff829a48d7 in HIMenuBarView::Construct
#5 0x7fff829a221f in HIView::EventHandler
#6 0x7fff8298d22b in HIObject::Construct
#7 0x7fff8298cdd9 in HIObject::Create
#8 0x7fff8298ccc4 in HIObjectCreate
#9 0x7fff829a469a in HIMenuBarView::Create
#10 0x7fff829a3f48 in HIMenuBarFrameView::Initialize
#11 0x7fff8299204b in HIObject::HandleClassHIObjectEvent
#12 0x7fff82991f19 in HIObject::EventHook
#13 0x7fff82991997 in DispatchEventToHandlers
#14 0x7fff82990ee6 in SendEventToEventTargetInternal
#15 0x7fff82990d57 in SendEventToEventTargetWithOptions
#16 0x7fff8298cdfa in HIObject::Create
#17 0x7fff8298ccc4 in HIObjectCreate
#18 0x7fff829a1d09 in NewWindowCommon
#19 0x7fff829a181f in _HIWindowCreateWithCGWindow
#20 0x7fff829a0aec in _GetMenuBarWindow
#21 0x7fff829a094c in GetMenuBarView
#22 0x7fff829a15d4 in MBarMenuRgn
#23 0x7fff829a1406 in ShowBar
#24 0x7fff829a10d0 in SetSystemUIMode
#25 0x7fff863d57d5 in -[NSApplication finishLaunching]
#26 0x7fff863d5350 in -[NSApplication run]
#27 0x10000104f in main at main.mm:51
```

The question is: how to eliminate/prevent or to work THAT (mostly genius) memory intervention to our/upp advantage?

Possible solutions:

1. Compiling conditions:

a) I tried to change in Core
`#if defined(UPP_HEAP) && !defined(__OBJC__)`
`#include <new>`
....

but HIObect and other things in Cocoa might be `__cplusplus__`, I guess. Hence, no help.

b) make `#include cocoa`
and then `#include Core/core.h`
nut we need something from AppKit inside Core.h? ...

2. Upp flag `USE_MALLOC` - also crash...

3. ...?

Any ideas, corrections for my thinking?