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

Posted by [daveremba](#) on Sun, 19 Jun 2011 22:50:55 GMT

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

I encountered a similar problem.
Not a crash, but a leak detected in UPP/heapdbg.cpp
I get "Heap leaks detected!" on exit.

Here is what I found:

a stack trace showed a MacOSX carbon function
from Xft... calling
UPP Core/operator new(!)
maybe the order of constructors in UPP
is calling Xft before it is initialized?

Here is a temp fix: in Draw/Font.cpp:
(telling UPP heap debugger to ignore this leak)

```
const CommonFontInfo& Font::Fi() const
{
// add:
MemoryIgnoreLeaksBlock __;

if(lastStdFont != AStdFont.AsInt64()) {
    lastFiFont = INT_MIN;
    lastStdFont = AStdFont.AsInt64();
}
if(AsInt64() == lastFiFont)
    return lastFontInfo;
// known leak on MacOSX here: getAllCarbonLazyValues2000 calls Core.h op new()
// should not call UPP op new()
// from GetFontInfo() ... XftFontOpenPattern() ... getAllCarbonLazyValues2000() -> new()
lastFontInfo = GetFontInfo(*this);
lastFiFont = AsInt64();
return lastFontInfo;
}
```

A stack trace from gdb is attached to this message.
(stack frames 0-5 are from a temp gets() to force a halt
while gdb is attached to the process)

I haven't tried debugging UPP in UPP yet!

I think the better fix is to include some Xft header

in Font.cpp after Core.h ?

```
Core/Core.h:201:inline void *operator new(size_t size) throw(std::bad_alloc) { void *ptr =  
UPP::MemoryAlloc(size); return ptr; }
```

Dave

File Attachments

1) [stack_trace_at_996.txt](#), downloaded 752 times
