The setup

```
class Lib {
    let path: String
    var handle: UnsafeMutableRawPointer?
    var lastReadTime: Int?
}
```

```swift
extension Lib {
    var lastWriteTime: Int? {
        var fileStats = stat()
        guard stat(path, &fileStats) == 0 else { return nil }
        return fileStats.st_mtimespec.tv_sec
    }

    func load() {
        handle = dlopen(path, RTLD_LAZY)
        assert(handle != nil, dlError)
        lastReadTime = lastWriteTime
    }

    func unload() {
        guard dlclose(handle) == 0 else { fatalError(dlError) }
        self.handle = nil
    }
}
```

```swift
extension Lib {
    var shouldReload: Bool {
        return lastWriteTime != lastReadTime
    }

    func reload() {
        guard shouldReload else { return }
        unload()
        usleep(10)
        load()
        print("\(path.basename) reloaded successfully at \(getTime())")
    }
}
```

Loading Symbols

```swift
extension Lib {
    func symbol(named name: String) -> UnsafeMutableRawPointer? {
        guard handle != nil else { return nil }
        let symbol = dlsym(handle, name)
        return symbol
    }

    func symbol<T>(named name: String, withType: T.Type) -> T? {
        guard let symbol = symbol(named: name) else { return nil }
        return unsafeBitCast(symbol, to: T.self)
    }
}
```

The Loop

```
let lib = DynamicLib(path: "@executable_path/Asteroids")
lib.load()

typealias LoadFunction   = @convention(c) () -> UnsafeMutableRawPointer?
typealias UpdateFunction = @convention(c) (UnsafeMutableRawPointer) -> Void

let setup = lib.symbol(named: "setup", withType: LoadFunction.self)
var memory = setup!()

// The start of the persisted memory must be the an Int representing the size
//   of the currently persisted memory. If the size is 0 then it's the lib indicating it is done.
while memory.assumingMemoryBound(to: Int.self).pointee != 0 {
    if lib.shouldReload {
        lib.reload()
    }
    let loop = lib.symbol(named: "update", withType: UpdateFunction.self)
    loop?(memory)
}
```

The Altering

```
// ...
    if lib.shouldReaload {
        let pre = lib.symbol(named: "preReload", withType: PreFunction.self)
        pre?(memory)

        lib.reload()

        let post = lib.symbol(named: "postReload", withType: PostFunction.self)
        post?(memory)
    }
// ...
```

The Game

```
@_silgen_name("setup")
func setup() -> UnsafeMutablePointer<GameState> {
    InitWindow(640, 480, "Asteroids")
    var gameState = GameState()
    // ...
    let memory = UnsafeMutablePointer<GameState>.allocate(capacity: 1)
    memory.initialize(to: gameState)
    return memory
}

@_silgen_name("update")
func update(_ memory: UnsafeMutablePointer<GameState>) {
    var gameState = memory.pointee; defer { memory.pointee = gameState }
    if WindowShouldClose() {
        gameState.currStateSize = 0
        CloseWindow()
        return
    }
    gameState.update()
    BeginFrame()
    ClearBackground(.black)
    gameState.render()
    EndFrame()
}
```

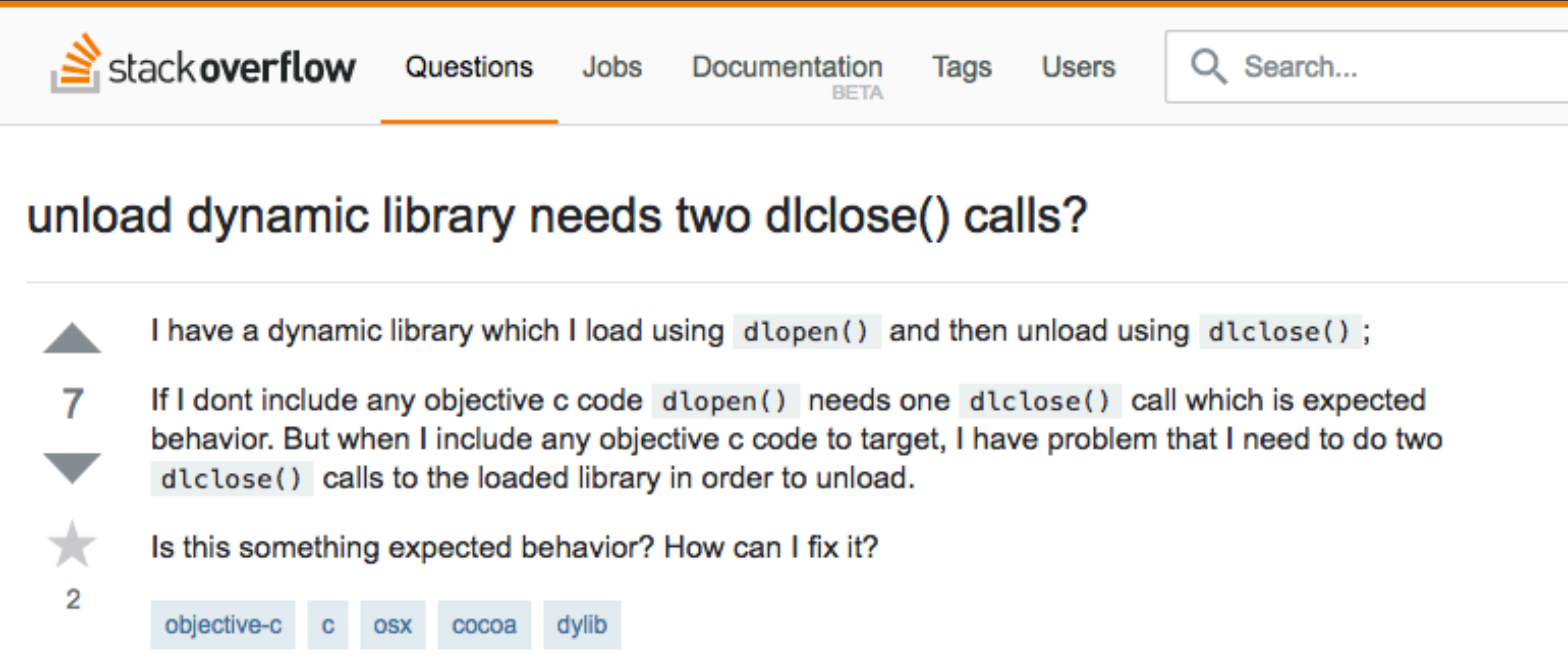
Caveat A

This hasn't worked since Swift snapshot 06-20 last year.

😭

Or so I thought.

To StackOverflow!



unload dynamic library needs two dlclose() calls?

I have a dynamic library which I load using `dlopen()` and then unload using `dlclose()`;

If I dont include any objective c code `dlopen()` needs one `dlclose()` call which is expected behavior. But when I include any objective c code to target, I have problem that I need to do two `dlclose()` calls to the loaded library in order to unload.

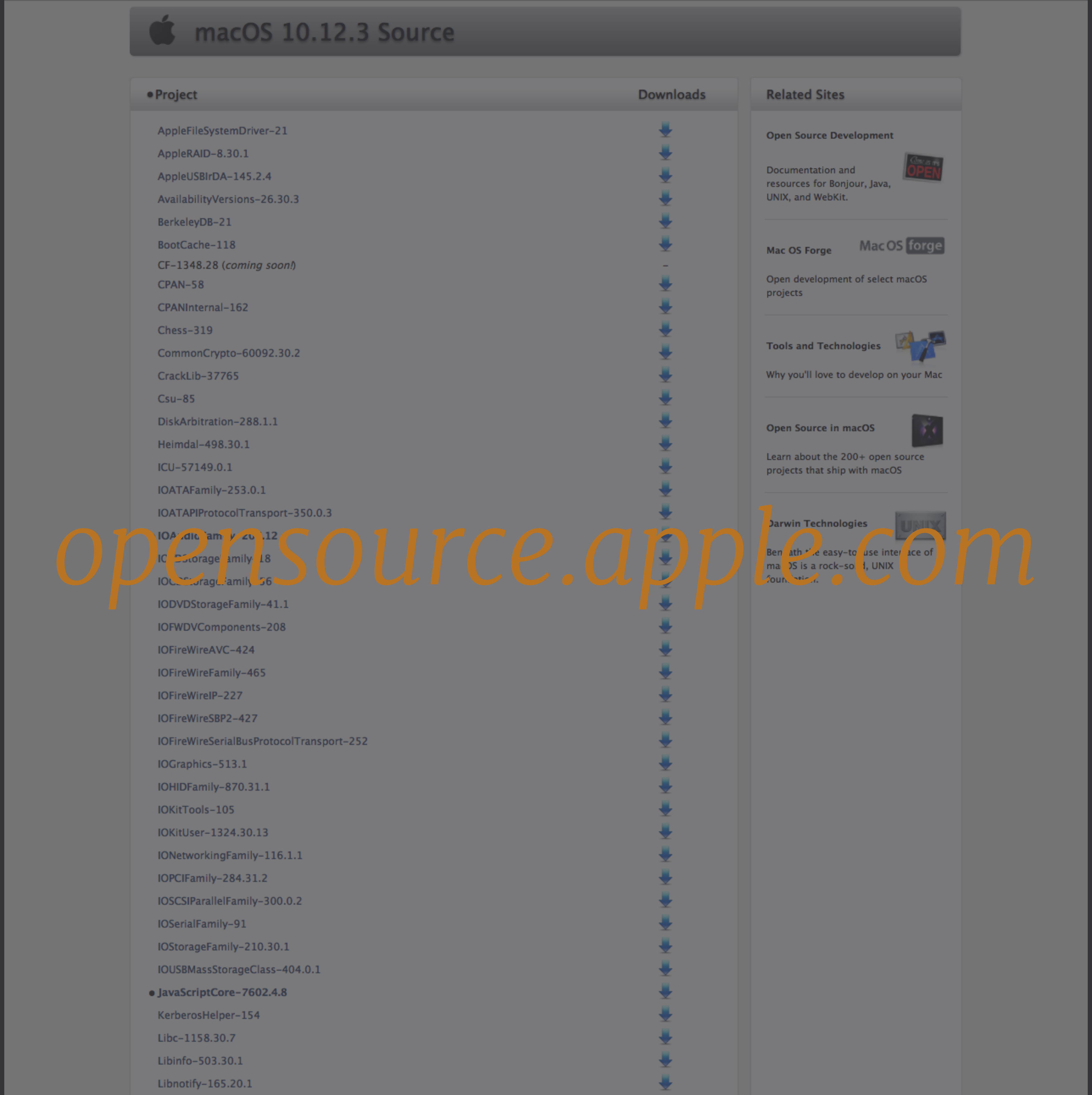
Is this something expected behavior? How can I fix it?

… there is no way to unload Cocoa bundles once loaded due to a runtime limitation.

```
// dylibs are not allowed to unload
// ...except those with image_info and nothing else (5359412)
if (hi->mhdr->filetype == MH_DYLIB  &&  _hasObjcContents(hi)) {
    dlopen(hi->fname, RTLD_NOLOAD);
}
```

Dear `_hasObjcContents`

Dear _hasObjcContents



macOS 10.12.3 Source

● Project	Downloads
AppleFileSystemDriver-21	⬇
AppleRAID-8.30.1	⬇
AppleUSBIrDA-145.2.4	⬇
AvailabilityVersions-26.30.3	⬇
BerkeleyDB-21	⬇
BootCache-118	⬇
CF-1348.28 *(coming soon!)*	–
CPAN-58	⬇
CPANInternal-162	⬇
Chess-319	⬇
CommonCrypto-60092.30.2	⬇
CrackLib-37765	⬇
Csu-85	⬇
DiskArbitration-288.1.1	⬇
Heimdal-498.30.1	⬇
ICU-57149.0.1	⬇
IOATAFamily-253.0.1	⬇
IOATAPIProtocolTransport-350.0.3	⬇
IOAudioFamily-205.12	⬇
IOBDStorageFamily-18	⬇
IOCDStorageFamily-56	⬇
IODVDStorageFamily-41.1	⬇
IOFWDVComponents-208	⬇
IOFireWireAVC-424	⬇
IOFireWireFamily-465	⬇
IOFireWireIP-227	⬇
IOFireWireSBP2-427	⬇
IOFireWireSerialBusProtocolTransport-252	⬇
IOGraphics-513.1	⬇
IOHIDFamily-870.31.1	⬇
IOKitTools-105	⬇
IOKitUser-1324.30.13	⬇
IONetworkingFamily-116.1.1	⬇
IOPCIFamily-284.31.2	⬇
IOSCSIParallelFamily-300.0.2	⬇
IOSerialFamily-91	⬇
IOStorageFamily-210.30.1	⬇
IOUSBMassStorageClass-404.0.1	⬇
● **JavaScriptCore-7602.4.8**	⬇
KerberosHelper-154	⬇
Libc-1158.30.7	⬇
Libinfo-503.30.1	⬇
Libnotify-165.20.1	⬇

Related Sites

Open Source Development

Documentation and resources for Bonjour, Java, UNIX, and WebKit.

Mac OS Forge

Open development of select macOS projects

Tools and Technologies

Why you'll love to develop on your Mac

Open Source in macOS

Learn about the 200+ open source projects that ship with macOS

Darwin Technologies

Beneath the easy-to-use interface of macOS is a rock-solid, UNIX foundation.

opensource.apple.com

```
// Look for an __objc* section other than __objc_imageinfo
static bool segmentHasObjcContents(const segmentType *seg)
{
    if (seg) {
        for (uint32_t i = 0; i < seg->nsects; i++) {
            const sectionType *sect = ((const sectionType *)(seg+1))+i;
            if (sectnameStartsWith(sect->sectname, "__objc_")  &&
                !sectnameEquals(sect->sectname, "__objc_imageinfo"))
            {
                return true;
            }
        }
    }

    return false;
}

// Look for an __objc* section other than __objc_imageinfo
bool
_hasObjcContents(const header_info *hi)
{
    const segmentType *data =
        getsegbynamefromheader(hi->mhdr, "__DATA");
    const segmentType *data_const =
        getsegbynamefromheader(hi->mhdr, "__DATA_CONST");
    const segmentType *data_dirty =
        getsegbynamefromheader(hi->mhdr, "__DATA_CONST");

    return segmentHasObjcContents(data)
        || segmentHasObjcContents(data_const)
        || segmentHasObjcContents(data_dirty);
}
```

`otool` is awesome

List dependencies (-L)

```
otool -L bin/Asteroids
bin/Asteroids:
    /System/Library/Frameworks/CoreFoundation.framework/Versions/A/CoreFoundation (compatibility version 150.0.0, current version 1349.64.0)
    /usr/local/opt/glfw/lib/libglfw.3.dylib (compatibility version 3.0.0, current version 3.2.0)
    @executable_path/libmuse.dylib (compatibility version 0.0.0, current version 0.0.0)
    /usr/lib/libobjc.A.dylib (compatibility version 1.0.0, current version 228.0.0)
    /usr/lib/libSystem.B.dylib (compatibility version 1.0.0, current version 1238.50.2)
    @rpath/libswiftCore.dylib (compatibility version 1.0.0, current version 802.0.53)
    @rpath/libswiftDarwin.dylib (compatibility version 1.0.0, current version 802.0.53)
    @rpath/libswiftSwiftOnoneSupport.dylib (compatibility version 1.0.0, current version 802.0.53)
```

Display load commands (-l)

```
otool -l bin/Asteroids | grep sectname
  sectname __text
  sectname __stubs
  sectname __stub_helper
  sectname __cstring
  sectname __swift3_typeref
  sectname __swift3_capture
  sectname __const
  sectname __swift3_reflstr
  sectname __swift3_fieldmd
  sectname __swift2_proto
  sectname __swift3_assocty
  sectname __ustring
  sectname __swift2_types
  sectname __swift3_builtin
  sectname __objc_classname
  sectname __objc_methname
  sectname __objc_methtype
  sectname __unwind_info
  sectname __eh_frame
  sectname __nl_symbol_ptr
  sectname __got
  sectname __la_symbol_ptr
  sectname __const
  sectname __objc_nlclslist
  sectname __objc_protolist
  sectname __objc_imageinfo
  sectname __objc_const
  sectname __objc_selrefs
  sectname __objc_protorefs
  sectname __objc_data
  sectname __data
  sectname __bss
  sectname __common
sectname __objc_nlclslist
sectname __objc_protolist
sectname __objc_imageinfo
sectname __objc_const
sectname __objc_selrefs
sectname __objc_protorefs
sectname __objc_data
```

```
swiftc --help | grep objc
  -emit-objc-header-path <path>
  -emit-objc-header       Emit an Objective-C header file
```

Nothing to help us here …

```
swiftc --help-hidden | grep objc
  -emit-objc-header-path <path>
  -emit-objc-header       Emit an Objective-C header file
  -import-objc-header <value>
  -no-link-objc-runtime   Don't link in additions to the Objective-C runtime
```

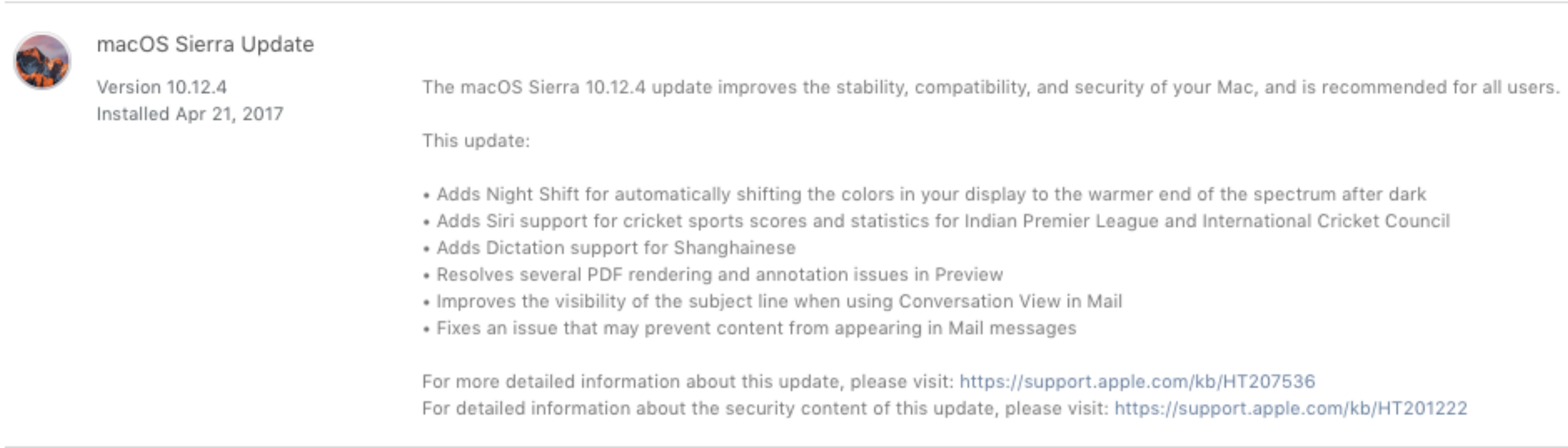
Hidden no longer.

```
set -e

c_flags="-Xcc -I/usr/local/include"
swiftc_flags="-Xswiftc -no-link-objc-runtime"
linker_flags="-Xlinker -L/usr/local/lib -Xlinker -lglfw"

swift build $c_flags $swiftc_flags $linker_flags

target_dir="$(pwd)/bin"
mkdir -p ${target_dir}
cp -f .build/debug/libmuse.dylib      ${target_dir}
cp -f .build/debug/Asteroids          ${target_dir}
cp -f .build/debug/LoopDynamic        ${target_dir}
```

```
otool -l bin/Asteroids | grep sectname
  sectname __text
  sectname __stubs
  sectname __stub_helper
  sectname __cstring
  sectname __swift3_typeref
  sectname __swift3_capture
  sectname __const
  sectname __swift3_reflstr
  sectname __swift3_fieldmd
  sectname __swift2_proto
  sectname __swift3_assocty
  sectname __ustring
  sectname __swift2_types
  sectname __swift3_builtin
  sectname __unwind_info
  sectname __eh_frame
  sectname __nl_symbol_ptr
  sectname __got
  sectname __la_symbol_ptr
  sectname __const
  sectname __objc_imageinfo
  sectname __data
  sectname __bss
  sectname __common
```

Solved.

Until this Sunday night …



macOS Sierra Update

Version 10.12.4
Installed Apr 21, 2017

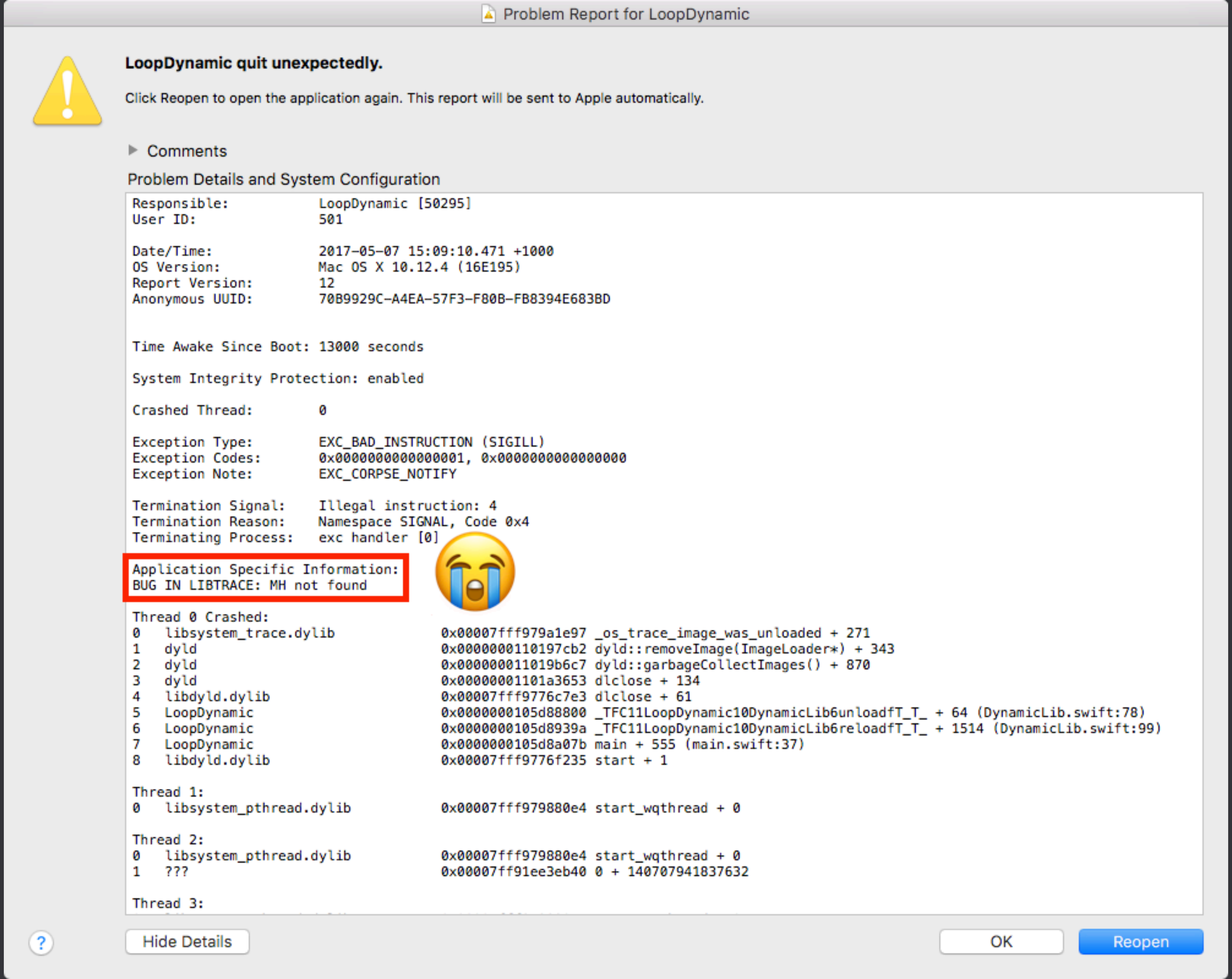
The macOS Sierra 10.12.4 update improves the stability, compatibility, and security of your Mac, and is recommended for all users.

This update:

- Adds Night Shift for automatically shifting the colors in your display to the warmer end of the spectrum after dark
- Adds Siri support for cricket sports scores and statistics for Indian Premier League and International Cricket Council
- Adds Dictation support for Shanghainese
- Resolves several PDF rendering and annotation issues in Preview
- Improves the visibility of the subject line when using Conversation View in Mail
- Fixes an issue that may prevent content from appearing in Mail messages

For more detailed information about this update, please visit: https://support.apple.com/kb/HT207536
For detailed information about the security content of this update, please visit: https://support.apple.com/kb/HT201222

They forgot to mention it will break dlclose.



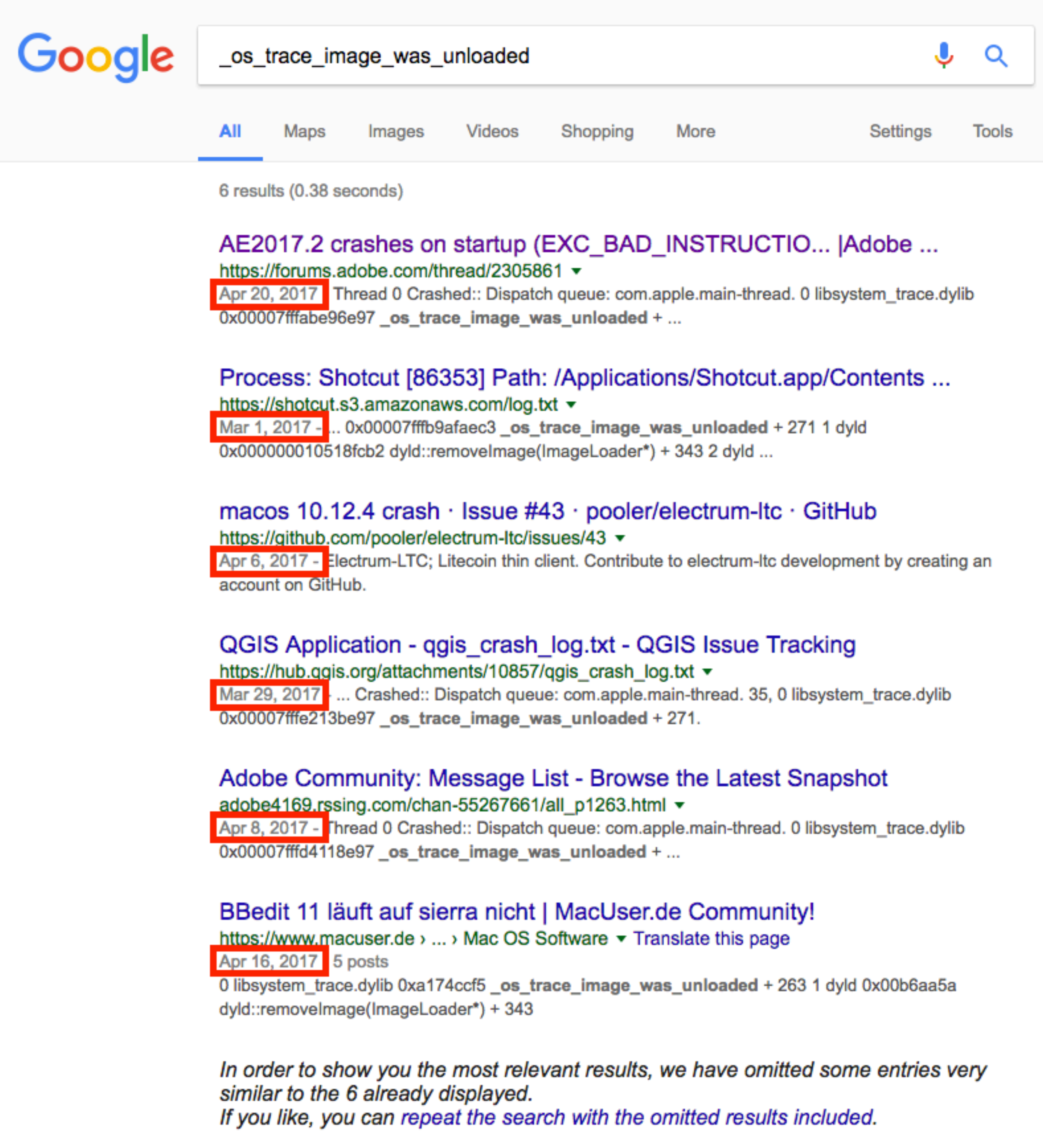
Problem Report for LoopDynamic

LoopDynamic quit unexpectedly.

Click Reopen to open the application again. This report will be sent to Apple automatically.

Problem Details and System Configuration

```
Responsible:           LoopDynamic [50295]
User ID:               501

Date/Time:             2017-05-07 15:09:10.471 +1000
OS Version:            Mac OS X 10.12.4 (16E195)
Report Version:        12
Anonymous UUID:        70B9929C-A4EA-57F3-F80B-FB8394E683BD


Time Awake Since Boot: 13000 seconds

System Integrity Protection: enabled

Crashed Thread:        0

Exception Type:        EXC_BAD_INSTRUCTION (SIGILL)
Exception Codes:       0x0000000000000001, 0x0000000000000000
Exception Note:        EXC_CORPSE_NOTIFY

Termination Signal:    Illegal instruction: 4
Termination Reason:    Namespace SIGNAL, Code 0x4
Terminating Process:   exc handler [0]

Application Specific Information:
BUG IN LIBTRACE: MH not found

Thread 0 Crashed:
0   libsystem_trace.dylib         0x00007fff979a1e97 _os_trace_image_was_unloaded + 271
1   dyld                          0x0000000110197cb2 dyld::removeImage(ImageLoader*) + 343
2   dyld                          0x000000011019b6c7 dyld::garbageCollectImages() + 870
3   dyld                          0x00000001101a3653 dlclose + 134
4   libdyld.dylib                 0x00007fff9776c7e3 dlclose + 61
5   LoopDynamic                   0x0000000105d88800 _TFC11LoopDynamic10DynamicLib6unloadfT_T_ + 64 (DynamicLib.swift:78)
6   LoopDynamic                   0x0000000105d8939a _TFC11LoopDynamic10DynamicLib6reloadfT_T_ + 1514 (DynamicLib.swift:99)
7   LoopDynamic                   0x0000000105d8a07b main + 555 (main.swift:37)
8   libdyld.dylib                 0x00007fff9776f235 start + 1

Thread 1:
0   libsystem_pthread.dylib       0x00007fff979880e4 start_wqthread + 0

Thread 2:
0   libsystem_pthread.dylib       0x00007fff979880e4 start_wqthread + 0
1   ???                           0x00007ff91ee3eb40 0 + 140707941837632

Thread 3:
```



_os_trace_image_was_unloaded

6 results (0.38 seconds)

AE2017.2 crashes on startup (EXC_BAD_INSTRUCTIO... |Adobe ...
https://forums.adobe.com/thread/2305861 ▾
Apr 20, 2017 Thread 0 Crashed:: Dispatch queue: com.apple.main-thread. 0 libsystem_trace.dylib 0x00007fffabe96e97 **_os_trace_image_was_unloaded** + ...

Process: Shotcut [86353] Path: /Applications/Shotcut.app/Contents ...
https://shotcut.s3.amazonaws.com/log.txt ▾
Mar 1, 2017 - ... 0x00007fffb9afaec3 **_os_trace_image_was_unloaded** + 271 1 dyld 0x000000010518fcb2 dyld::removeImage(ImageLoader*) + 343 2 dyld ...

macos 10.12.4 crash · Issue #43 · pooler/electrum-ltc · GitHub
https://github.com/pooler/electrum-ltc/issues/43 ▾
Apr 6, 2017 - Electrum-LTC; Litecoin thin client. Contribute to electrum-ltc development by creating an account on GitHub.

QGIS Application - qgis_crash_log.txt - QGIS Issue Tracking
https://hub.qgis.org/attachments/10857/qgis_crash_log.txt ▾
Mar 29, 2017 - ... Crashed:: Dispatch queue: com.apple.main-thread. 35, 0 libsystem_trace.dylib 0x00007fffe213be97 **_os_trace_image_was_unloaded** + 271.

Adobe Community: Message List - Browse the Latest Snapshot
adobe4169.rssing.com/chan-55267661/all_p1263.html ▾
Apr 8, 2017 - Thread 0 Crashed:: Dispatch queue: com.apple.main-thread. 0 libsystem_trace.dylib 0x00007fffd4118e97 **_os_trace_image_was_unloaded** + ...

BBedit 11 läuft auf sierra nicht | MacUser.de Community!
https://www.macuser.de › ... › Mac OS Software ▾ Translate this page
Apr 16, 2017 5 posts
0 libsystem_trace.dylib 0xa174ccf5 **_os_trace_image_was_unloaded** + 263 1 dyld 0x00b6aa5a dyld::removeImage(ImageLoader*) + 343

In order to show you the most relevant results, we have omitted some entries very similar to the 6 already displayed.
If you like, you can repeat the search with the omitted results included.