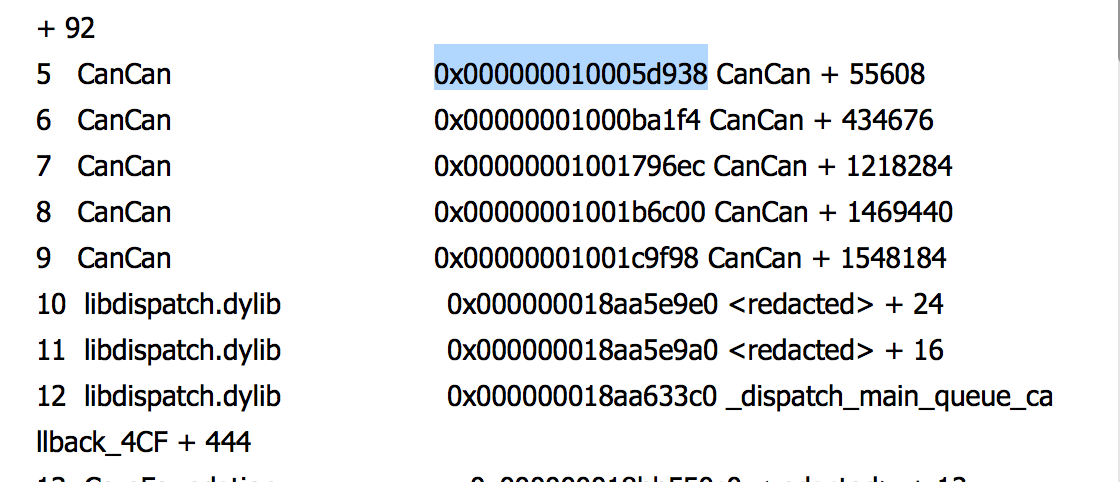
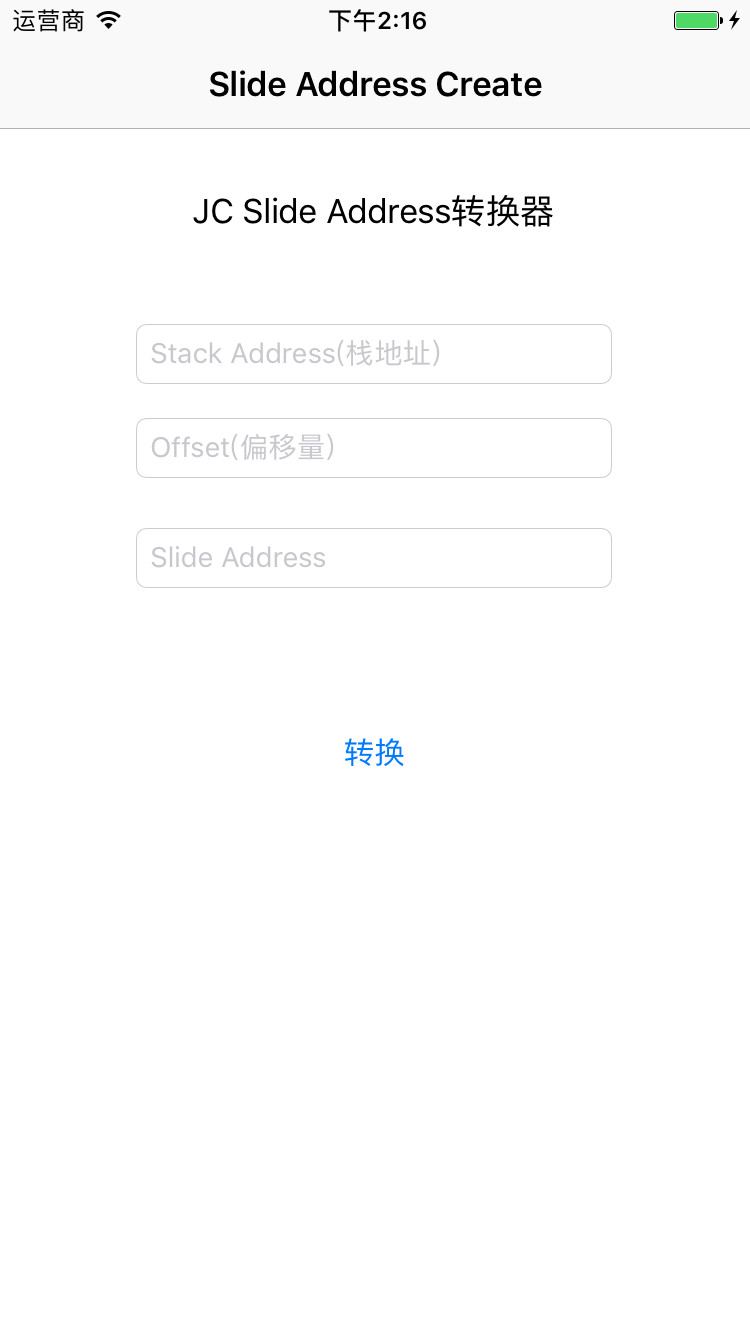
**崩溃日志符号化利器使用说明**

一、利用ios App ->JCCrashTest获取崩溃日志的Slide Address

Slide Address 不用多说了，详情可参见

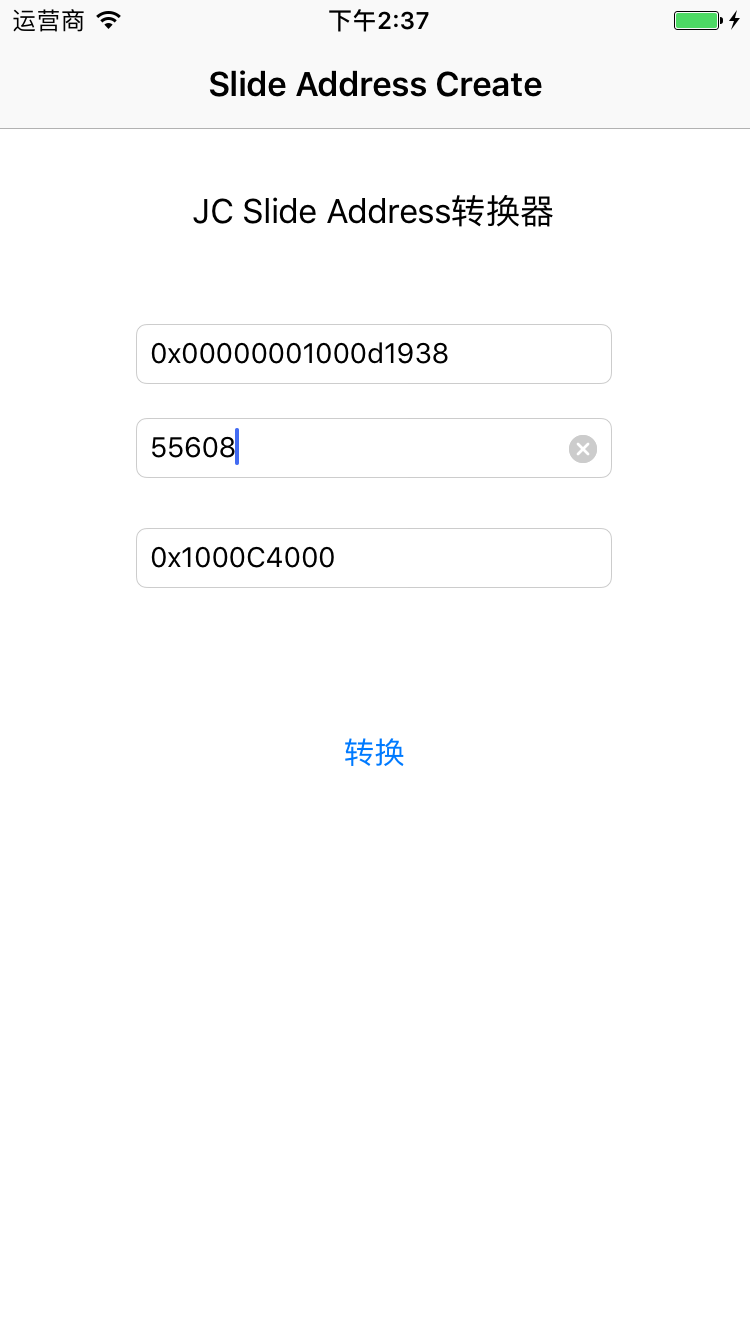
———[http://www.jianshu.com/p/746184ea5662](http://www.jianshu.com/p/746184ea5662)



根据上边的那个链接，了解完Slide Address之后，利用JCCrashTest利器，求出崩溃日志的 Slide Address(将崩溃日志最前边的那个崩溃的栈地址填写，偏移量同时写入，点击转换，即可在Slide Address输入框获取到你想要的Slide Address)

例如上边，第五行，崩溃的栈地址就是 0x000000010005d938，偏移量就是55608。输入之后，点击转换，即可获取Slide Address。

结果如图

二、获取Slide Address之后，利用DSYMTools，来进行单行分析，总体分析，导出符号化的崩溃日志等等功能(在原作者基础之上添加了总体分析和导出符号化崩溃日志两个功能，同时把分析结果的控件区域变大了，不然看起来很别扭)

单条分析使用步骤:

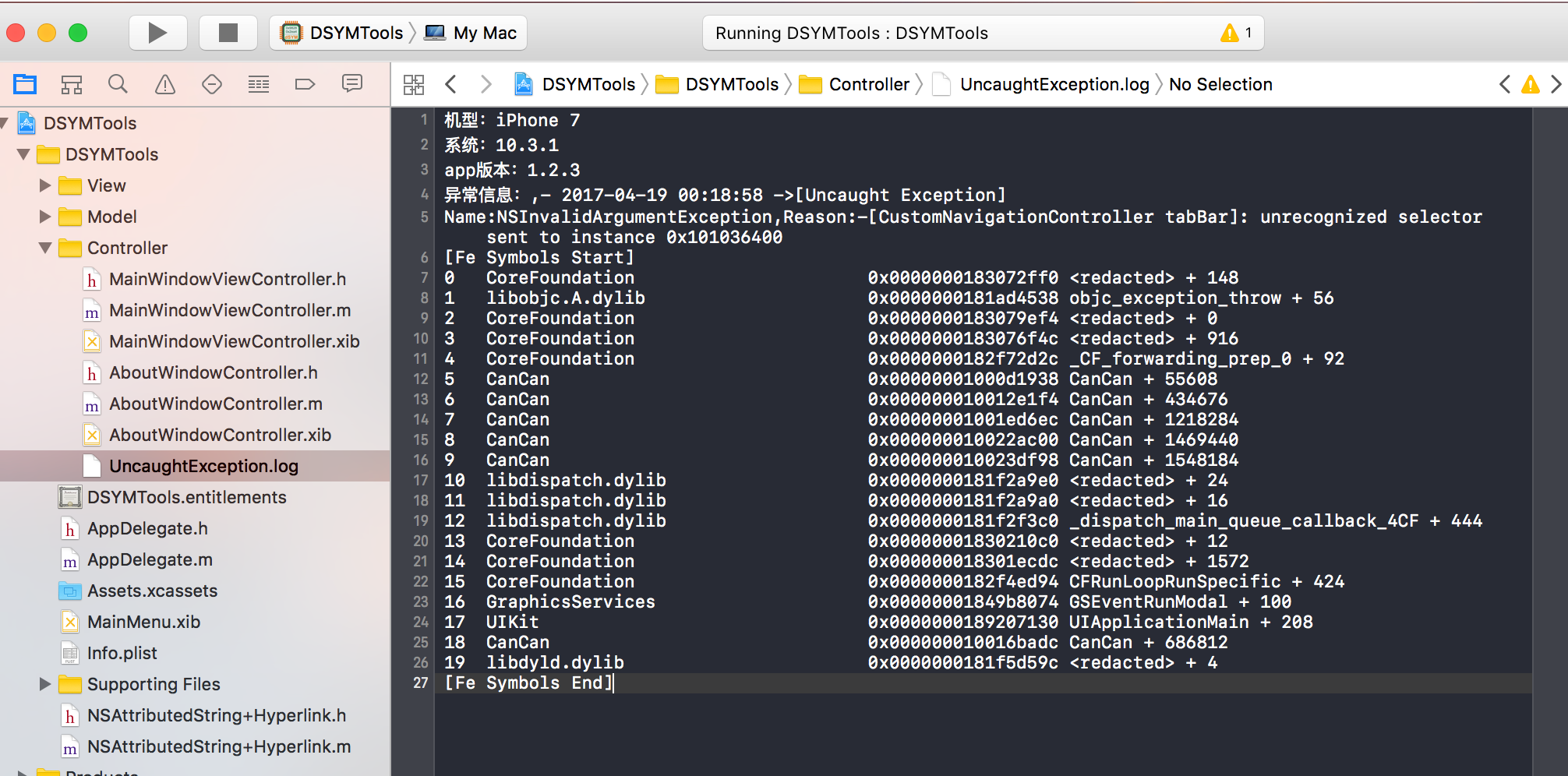
1.将打包发布软件时的xcarchive文件(或者DSYM文件也可以)拖入软件窗口内的任意位置(支持多个文件同时拖入，注意：文件名不要包含空格)

2.选中任意一个版本的xcarchive文件(或者DSYM文件也可以)，右边会列出该xcarchive文件支持的CPU类型，选中错误对应的CPU类型。

3.对比错误给出的UUID和工具界面中给出的UUID是否一致。

4.将错误地址以及 Slide Address 输入工具的文本框中，点击单条分析。

总体分析概图



第一步，很重要

把你的错误信息command+c复制(建议是单个运行时的错误信息，最后有注释)，然后command+v粘贴到DSYMTools里边的 UncaughtException.log文件里边，这是最重要的，为什么呢？

因为:

总体分析，总的原理是通过读取UncaughtException.log的内容，然后遍历其中的错误信息，找出错误的栈地址，遍历进行分析，并进行转换。所以总体分析之前，必须要把你想要分析的错误信息复制到UncaughtException.log里边。这是分析的数据源，记住哦！单条分析和这个文件就没有关系了，因为单条分析的话，你是必须在软件界面输入错误信息的栈地址的，单条分析一次，输入一次错误信息的栈地址。(单条分析是手动输入错误信息栈地址，总体分析是靠遍历UncaughtException.log文件内容进行总体分析)



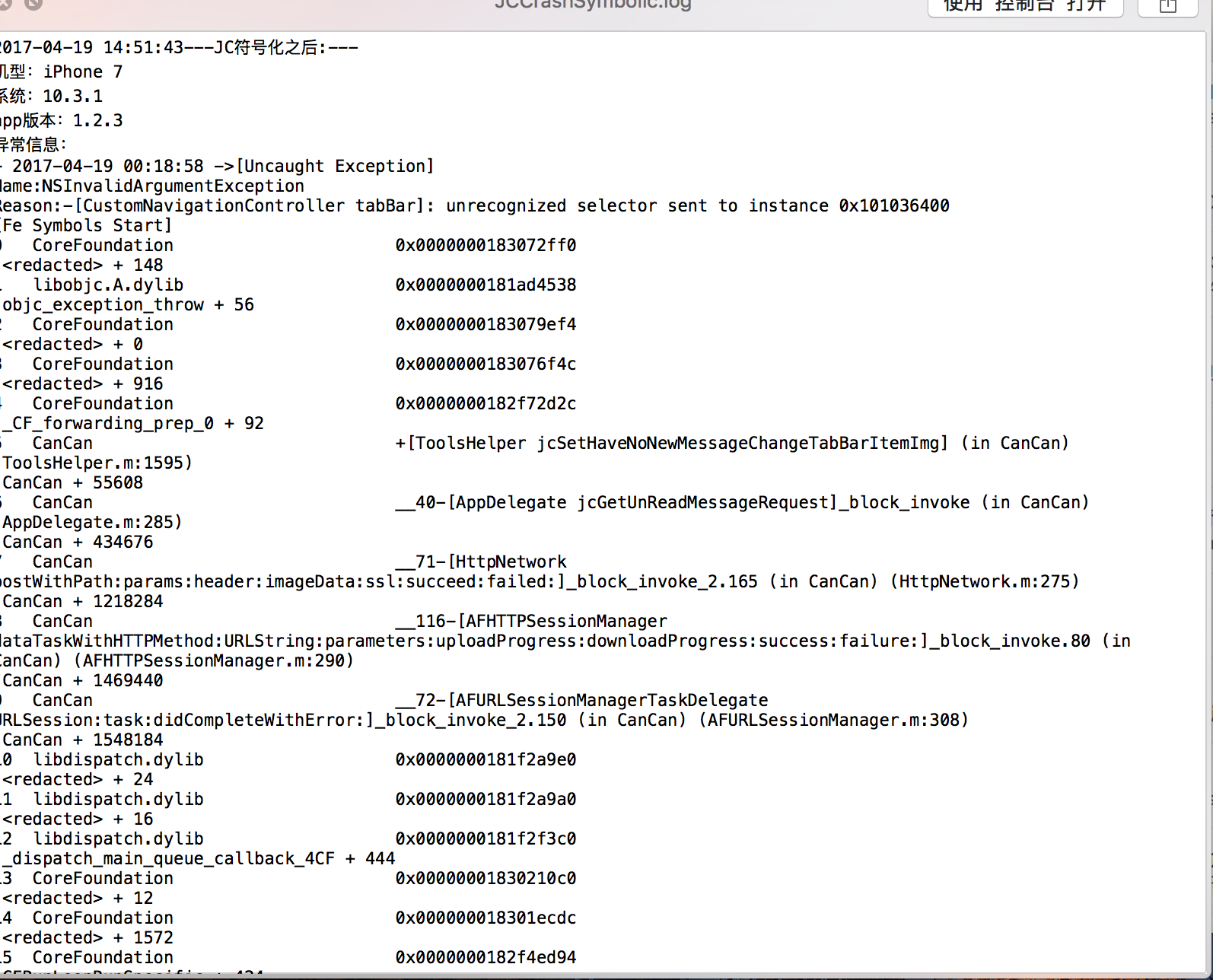
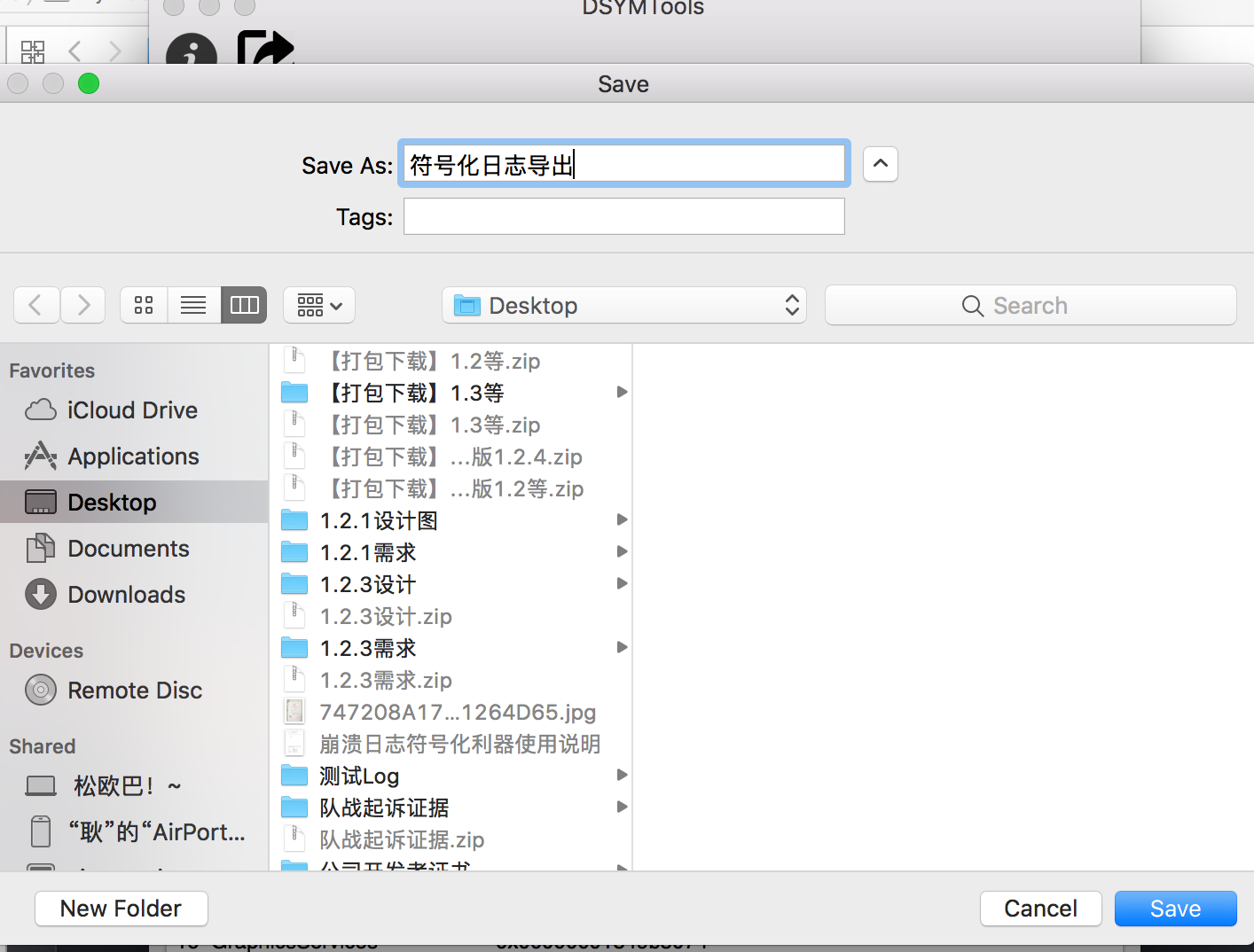
总体分析使用步骤:

1 输入Slide Address

2 点击总体分析

注:总体分析，错误信息内存地址不是必填的，有没有错误信息内存地址都是可以进行分析的。但是单条分析的话，必须把想要分析的那个错误信息内存地址填入，再点击单条分析，不填写错误信息内存地址，单条分析无法正常工作。

三、符号化崩溃日志导出

废话少说，直接上图

可以看到，导出到桌面的日志，已经符号化完毕了，so easy!

总体注意:不论是单条分析，还是总体分析，记住分析的Slide Address地址是会变的，这个地址是在运行时变的，所以每次运行一次App,这个值就变一次。分析崩溃信息的话，一个时间段的崩溃信息，或者可以这样说，一个运行时内的崩溃信息，Slide Address是相同的。

[Fe Symbols Start]

0 CoreFoundation 0x0000000183072ff0 <redacted> + 148

1 libobjc.A.dylib 0x0000000181ad4538 objc\_exception\_throw + 56

2 CoreFoundation 0x0000000183079ef4 <redacted> + 0

3 CoreFoundation 0x0000000183076f4c <redacted> + 916

4 CoreFoundation 0x0000000182f72d2c \_CF\_forwarding\_prep\_0 + 92

5 CanCan 0x00000001000d1938 CanCan + 55608

6 CanCan 0x000000010012e1f4 CanCan + 434676

7 CanCan 0x00000001001ed6ec CanCan + 1218284

8 CanCan 0x000000010022ac00 CanCan + 1469440

9 CanCan 0x000000010023df98 CanCan + 1548184

10 libdispatch.dylib 0x0000000181f2a9e0 <redacted> + 24

11 libdispatch.dylib 0x0000000181f2a9a0 <redacted> + 16

12 libdispatch.dylib 0x0000000181f2f3c0 \_dispatch\_main\_queue\_callback\_4CF + 444

13 CoreFoundation 0x00000001830210c0 <redacted> + 12

14 CoreFoundation 0x000000018301ecdc <redacted> + 1572

15 CoreFoundation 0x0000000182f4ed94 CFRunLoopRunSpecific + 424

16 GraphicsServices 0x00000001849b8074 GSEventRunModal + 100

17 UIKit 0x0000000189207130 UIApplicationMain + 208

18 CanCan 0x000000010016badc CanCan + 686812

19 libdyld.dylib 0x0000000181f5d59c <redacted> + 4

[Fe Symbols End]

一个[Fe Symbols Start]->[Fe Symbols End]为一个运行时吧，个人理解。如果你的崩溃日志里边有多个[Fe Symbols Start]，建议分开来计算Slide Address,同理，总体分析的时候，复制崩溃信息到工程的时候，建议只复制单个运行时的崩溃信息，同时用一个Slide Address分析多段运行时的崩溃信息，可能会出错，导致崩溃信息符号化定位不准。