
目录

前言	1.1
导出头文件概述	1.2
ObjC	1.3
class-dump	1.3.1
不同版本	1.3.1.1
nygard/class-dump	1.3.1.1.1
MonkeyDev的class-dump	1.3.1.1.2
0xcd/class-dump	1.3.1.1.3
lechium/classdumpios	1.3.1.1.4
用法	1.3.1.2
举例	1.3.1.2.1
语法help	1.3.1.3
Swift	1.4
dsdump	1.4.1
不同版本	1.4.1.1
DerekSelander/dsdump	1.4.1.1.1
paradiseduo/dsdump	1.4.1.1.2
paradiseduo/resymbol	1.4.1.1.2.1
crifan/dsdump	1.4.1.1.3
用法	1.4.1.2
举例	1.4.1.2.1
最新结论	1.5
对比	1.5.1
问题和心得	1.6
导出头文件是空的	1.6.1
附录	1.7
参考资料	1.7.1

iOS逆向分析：导出头文件

- 最新版本： `v1.0.0`
- 更新时间： `20250101`

简介

介绍iOS逆向之静态分析中的导出头文件相关工具和方法。

源码+浏览+下载

本书的各种源码、在线浏览地址、多种格式文件下载如下：

HonKit源码

- [crifan/ios_re_export_header: iOS逆向分析：导出头文件](#)

如何使用此HonKit源码去生成发布为电子书

详见：[crifan/honkit_template: demo how to use crifan honkit template and demo](#)

在线浏览

- [iOS逆向分析：导出头文件 book.crifan.org](#)
- [iOS逆向分析：导出头文件 crifan.github.io](#)

离线下载阅读

- [iOS逆向分析：导出头文件 PDF](#)
- [iOS逆向分析：导出头文件 ePub](#)
- [iOS逆向分析：导出头文件 Mobi](#)

版权和用途说明

此电子书教程的全部内容，如无特别说明，均为本人原创。其中部分内容参考自网络，均已备注了出处。如发现有侵权，请通过邮箱联系我 `admin` 艾特 `crifan.com`，我会尽快删除。谢谢合作。

各种技术类教程，仅作为学习和研究使用。请勿用于任何非法用途。如有非法用途，均与本人无关。

鸣谢

感谢我的老婆陈雪的包容理解和悉心照料，才使得我 `crifan` 有更多精力去专注技术专研和整理归纳出这些电子书和技术教程，特此鸣谢。

其他

作者的其他电子书

本人 `crifan` 还写了其他 `150+` 本电子书教程，感兴趣可移步至：

[crifan/crifan_ebook_readme: Crifan的电子书的使用说明](#)

关于作者

关于作者更多介绍，详见：

[关于CrifanLi李茂 – 在路上](#)

crifan.org，使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：2025-01-01 23:08:20

iOS逆向之导出头文件

iOS逆向的静态分析中，往往涉及到：从二进制文件中，导出ObjC的类的头文件，简称：**导出头文件**

- 导出头文件
 - 前提
 - iOS的app的二进制是已经**砸壳**后的 = 解密后的
 - 相关机制
 - iOS的二进制文件格式是Mach-O
 - 可以用相关工具，从Mach-O中分析和提取出，ObjC（和Swift）相关类的定义，从而导出头文件
 - 目的
 - 用于iOS逆向时，搞懂相关（ObjC和Swift的）类的属性定义和函数
 - 从而继续优化IDA中伪代码
 - 配合动态调试，直到最终彻底搞懂代码逻辑

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2025-01-01 21:59:35

ObjC

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新: 2025-01-01 22:07:54

class-dump

- class-dump
 - 一句话描述: iOS逆向中导出ObjC的头文件的常用工具
 - 用于处理 Objective-C 的 Mach-O 文件信息的命令行工具, 可以导出类的定义、分组和协议。
 - command-line utility for examining the Objective-C segment of Mach-O files
 - 说明
 - 和 `otool -ov` 导出的信息是一样的
 - 但是显示为 Objective-C 定义, 更易读
 - 原理
 - 利用了 Objective-C 语言的运行时的特性
 - 将存储在 Mach-O 文件中的头文件信息提取出来, 并生成对应的 .h 文件
 - 用途
 - 查看闭源的 应用、frameworks、bundles
 - 查看其中的头文件信息
 - 对比一个 APP 不同版本之间的接口变化
 - 通过导出不同版本的库的头文件的对比看出来
 - 对一些私有 frameworks 做些有趣的试验

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新: 2025-01-01 22:28:06

不同版本的class-dump

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新: 2025-01-01 22:46:19

nygard/class-dump

- nygard/class-dump
 - 官网的版本
 - 安装包
 - [class-dump-3.5.dmg](#)
 - 相关资料
 - GitHub
 - nygard/class-dump: Generate Objective-C headers from Mach-O files.
 - <https://github.com/nygard/class-dump>
 - 官网
 - class-dump - Steve Nygard
 - <http://stevenyard.com/projects/class-dump/>
 - 源码
 - [class-dump-3.5.tar.gz](#)
 - 或: [class-dump-3.5.tar.bz2](#)
 - 二进制
 - [class-dump-3.5.dmg](#)

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新: 2025-01-01 22:17:38

MonkeyDev的class-dump

- MonkeyDev 中的 class-dump
 - 概述
 - 升级版=优化版的class-dump: 支持 swift 和 objc 混淆
 - 下载
 - <https://github.com/AloneMonkey/MonkeyDev/blob/master/bin/class-dump>

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新: 2025-01-01 22:38:36

0xcd/class-dump

- 0xcd/class-dump
 - 概述
 - 疑似是上述的MonkeyDev的二进制 `class-dump` 的源码来源
 - 也属于升级版=优化版的class-dump: 支持 `swift` 和 `objc` 混淆
 - 最后更新时间: 2016年 -》 也的确有点太老了
 - Github
 - <https://github.com/0xcd/class-dump>
 - swift分支=swift版本
 - <https://github.com/0xcd/class-dump/tree/swift-binaries>
 - 其他引用
 - <https://github.com/tobefuturer/restore-symbol>
 - <https://github.com/0xcd/class-dump/tree/a8877b6695f317816322134944a410de09da4911>
 - <https://github.com/HeiTanBc/restore-symbol>
 - <https://github.com/HeiTanBc/class-dump>
 - <https://github.com/HeiTanBc/class-dump/tree/f6af44053366f27670d62e446c80c6380de6706e>
 - 应该是: HeiTanBc根据原版0xcd/class-dump, 去更新修复而得

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新: 2025-01-01 22:50:27

lechiu/classdumpios

- lechiu/classdumpios
 - 概述
 - 改进版的class-dump: 支持最新的 Load Command (比如: LC_DYLD_CHAINED_FIXUPS) 的版本 == 支持 LC_DYLD_CHAINED_FIXUPS 的class-dump
 - 目的: 旧版 class-dump 对于新的 Mach-O 会报错: Unknown load command: 0x00000032, 对应的需要, 支持最新的 LC_DYLD_CHAINED_FIXUPS 的 Load Command
 - Github
 - <https://github.com/lechiu/classdumpios>

如何得到 classdumpc

- 直接下载
 - <https://github.com/lechiu/classdumpios/releases>
 - 4.2.0 RELEASE
 - https://github.com/lechiu/classdumpios/releases/download/4.2.0-RELEASE1/classdump-c_4.2.0-RELEASE1.zip
 - 下载后, 解压得到二进制文件: classdump-c_4.2.0-RELEASE1/Release/classdumpc
- 自己编译
 - 下载 macos 的branch的源码:
 - <https://github.com/lechiu/classdumpios/tree/macos>
 - 自己用Xcode编译 (出二进制: classdumpc)

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新: 2025-01-01 22:22:01

class-dump的用法

class-dump 系列的典型用法:

```
class-dump --arch <arch> -H -o outputFolder inputBinaryFile
classdumpe --arch <arch> -H -o outputFolder inputBinaryFile
```

- 参数说明

- --arch : 指定架构
 - 最常见的值: arm64
 - 默认可以省略
 - 但如果是 FAT = 胖二进制 时, 则必须指定对应架构
- -H : 输出头文件
- -o : 输出目录

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新: 2025-01-01 22:52:46

class-dump用法举例

TODO:

- 【基本解决】 砸壳抖音ipa后导出iOS抖音头文件

- AwemeCore

```
MonkeyDev/bin/class-dump --arch arm64 -H AwemeCore -o /Users/crifan/dev/DevRoot/iOSReverse/Aweme/class_dump_output
```

- Aweme

```
AloneMonkey/MonkeyDev/bin/class-dump --arch arm64 -H Aweme -o /Users/crifan/dev/DevRoot/Aweme/classDumpResult/17.8.0/Aweme
```

- YouTube

```
class-dump --arch arm64 -H . /ipa/YouTube_17.08.2_dumped/Payload/YouTube.app/YouTube -o .
```

- MusicallyCore

```
./class-dump --arch arm64 -H ipa/Payload/TikTok.app/Frameworks/MusicallyCore.framework/MusicallyCore -o tiktok_headers_26.8.0
```

- Apple Store

```
class-dump --arch arm64 -H AppleStore/ipa/Payload/Apple Store.app/Apple Store -o AppleStore_headers_5.18.0.910
```

- mobileactivationd

```
classdump-c_4.2.0-RELEASE1/Release/classdumpc --arch arm64 -H -o ../staticAnalysis/headers mobileactivationd
```

- 其他

- o class-dump AppKit

- class-dump /System/Library/Frameworks/AppKit.framework

- o class-dump UIKit

- class-dump /Developer/Platforms/iPhoneOS.platform/Developer/SDKs/iPhoneOS4.3.sdk/System/Library/Frameworks/UIKit.framework

- o class-dump UIKit and all the frameworks it uses

- class-dump /Developer/Platforms/iPhoneOS.platform/Developer/SDKs/iPhoneOS4.3.sdk/System/Library/Frameworks/UIKit.framework -r --sdk-ios 4.3

- o class-dump UIKit (and all the frameworks it uses) from developer tools that have been installed in /Dev42 instead of /Developer

- class-dump /Dev42/Platforms/iPhoneOS.platform/Developer/SDKs/iPhoneOS5.0.sdk/System/Library/Frameworks/UIKit.framework -r --sdk-root /Dev42/Platforms/iPhoneOS.platform/Developer/SDKs/iPhoneOS5.0.sdk

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新: 2025-01-01 22:52:51

class-dump的语法help

```
class-dump 3.5 (64 bit)
Usage: class-dump [options] mach-o-file

where options are:
-a          show instance variable offsets
-A          show implementation addresses
--arch arch choose a specific architecture from a universal binary (ppc, ppc64, i386, x86_64)
-C <regex> only display classes matching regular expression
-f <str>    find string in method name
-H          generate header files in current directory, or directory specified with -o
-I          sort classes, categories, and protocols by inheritance (overrides -s)
-o <dir>    output directory used for -H
-r          recursively expand frameworks and fixed VM shared libraries
-s          sort classes and categories by name
-S          sort methods by name
-t          suppress header in output, for testing
--list-arches list the arches in the file, then exit
--sdk-ios   specify iOS SDK version (will look in /Developer/Platforms/iPhoneOS.platform/Developer/SDKs/iPhone
OS version .sdk
--sdk-mac   specify Mac OS X version (will look in /Developer/SDKs/MacOSX version .sdk
--sdk-root  specify the full SDK root path (or use --sdk-ios/--sdk-mac for a shortcut)
```

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新: 2025-01-01 22:41:23

Swift

其他

- Swift
 - <https://github.com/neil-wu/SwiftDump>

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新: 2025-01-01 22:44:00

dsdump

- dsdump
 - 概述: (另外一个) 支持 Swift 的版本
 - 目的: 和 `class-dump` 输出结果 (导出的头文件), 有时候不完全一样 (比如Swift的信息输出略有不同)
 - 此时可以通过2套结果, 互相对比, 便于逆向时查看细节信息

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2025-01-01 23:05:42

不同版本的dsdump

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新: 2025-01-01 22:48:11

DerekSelander/dsdump

- DerekSelander/dsdump
 - 概述
 - 原始版本的dsdump
 - Github
 - <https://github.com/DerekSelander/dsdump>

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新: 2025-01-01 22:24:21

paradiseduo/dsdump

- `paradiseduo/dsdump`
 - 概述
 - 新版的优化后的dsdump
 - 评价
 - 输出内容带类的属性字段的大小和偏移量，效果不错
 - 结论
 - 但是有些细节做的不够好，尤其是部分类包含其他类的内容，所以建议换用：
 - [crifan/dsdump](#)
 - Github
 - <https://github.com/paradiseduo/dsdump>

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新: 2025-01-01 22:55:46

paradiseduo/resymbol

- paradiseduo/resymbol
 - 概述
 - 用于替代paradiseduo/dsdump
 - 评价
 - 但是输出内容只是单个的类似于Swift类的定义的头文件，并没有包含属性字段的大小和偏移量
 - 结论
 - 不建议作为主要输出，仅建议作为辅助输出作为参考
 - Github
 - <https://github.com/paradiseduo/resymbol>

如何得到二进制的 resymbol

- 下载代码

```
git clone https://github.com/paradiseduo/resymbol.git
```

- 自己编译

- Apple Silicon = ARM

```
./build-macOS_arm.sh
```

- Intel = X86

```
build-macOS_x86.sh
```

- 编译输出二进制文件: resymbol

resymbol用法

```
resymbol inputMachoFile > outputSingleHeaderFile.h
```

举例

```
resymbol ../../../../input/mobileactivationd > mobileactivationd_headers_resymbol.h
```

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新: 2025-01-01 22:54:24

crifan/dsdump

- `crifan/dsdump`
 - 概述
 - Crifan基于[paradiseduo/dsdump](#)优化后的版本的dsdump
 - Github
 - <https://github.com/crifan/dsdump>

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新: 2025-01-01 22:27:29

dsdump用法

下载:

```
git clone https://github.com/crifan/dsdump.git
```

使用:

```
python3 dsdump.py -d -1 machOFile -o outputFolder
```

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新: 2025-01-01 22:52:18

dsdump用法举例

```
python3 dsdump.py -d -i /Users/crifan/dev/dev_root/iosReverse/WhatsApp/ipa/Payload/WhatsApp.app/WhatsApp -o WhatsApp_swift_headers
```

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新: 2025-01-01 22:52:22

最新结论

- 导出头文件的最新结论
 - ObjC
 - 推荐：支持Swift和ObjC混淆的版本
 - MonkeyDev 中的 `class-dump`
 - <https://github.com/AloneMonkey/MonkeyDev/blob/master/bin/class-dump>
 - 如果报错 `Unknown load command: 0x00000032`，则要换成：支持 `LC_DYLD_CHAINED_FIXUPS` 的：`classdumpc`
 - https://github.com/lechium/classdumpios/releases/download/4.2.0-RELEASE1/classdump-c_4.2.0-RELEASE1.zip
 - Swift
 - 用我自己基于 `paradiseduo/dsdump` 继续优化的 `crifan/dsdump`
 - 原因：能显示字段属性的大小和偏移量 -> 方便辅助调试搞懂类的字段定义 -> 快速写出类的定义 -> 加到 IDA 中 -> 优化伪代码 -> 尽快搞懂代码含义

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新： 2025-01-01 23:05:08

对比

dsdump vs class-dump

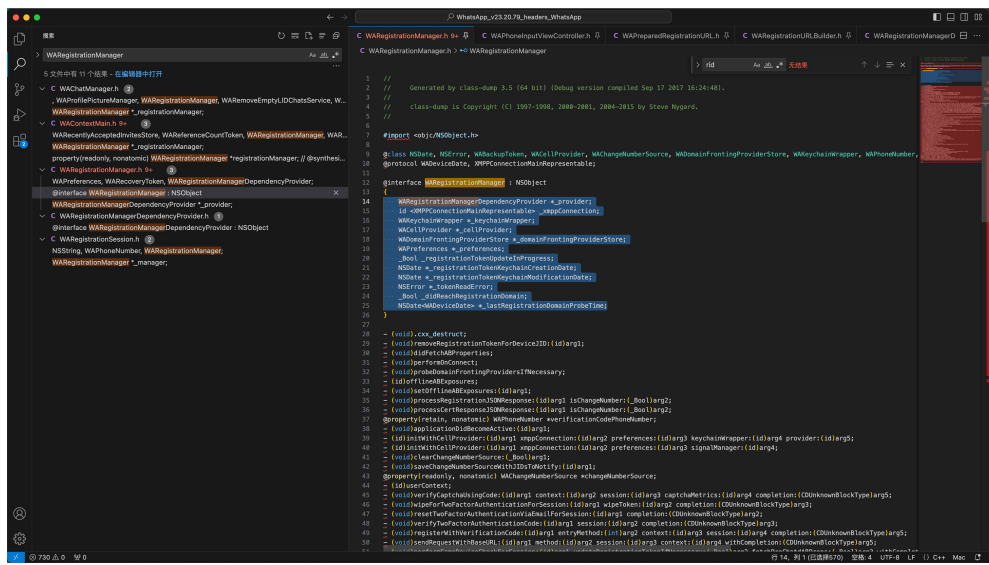
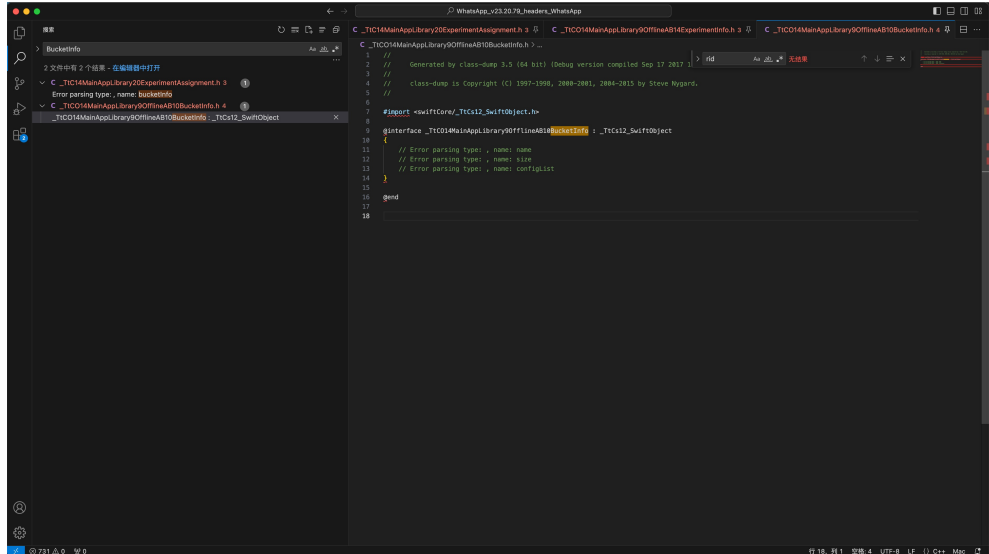
- dsdump vs class-dump

- 概述: 总体算各有千秋
- 细节

- 多数类

- class-dump

- 能导出对应的类和私有属性, 但是不显示偏移量和大小



- dsdump

- 类中属性, 除了属性名和类型, 还能显示出偏移量和空间占用大小》很方便, 利于调试和逆向

```

C:\OfflineAB\BucketInfo.h
1 BucketInfo
2 {
3     @+80010 name (NSString)
4     @+80020 size (NSNumber)
5     @+80030 configList (NSArray)
6 }
7
8 @+801835c108 _TTC14MainAppLibraryOfflineABInfo : Swift_SwiftObject @path/libswiftCore.dylib
9 {
10     @+80010 name (NSString)
11     @+80020 size (NSNumber)
12     @+80030 configList (NSArray)
13 }
14
15 @+801835c108 _TTC14MainAppLibraryOfflineABInfo : Swift_SwiftObject @path/libswiftCore.dylib
16 {
17     @+80010 name (NSString)
18     @+80020 startTIme (NSNumber)
19     @+80020 endTime (NSNumber)
20     @+80030 bucketList (NSArray)
21     @+80040 userInfo (NSMutableDictionary)
22 }
23
24 @+801835c108 _TTC14MainAppLibraryOfflineABInfo : Swift_SwiftObject @path/libswiftCore.dylib
25 {
26     @+80010 userInfo (NSMutableDictionary)
27     @+80020 mimeType (NSString)
28     @+80040 configList (NSArray)
29 }
30
31 @+801835c108 _TTC14MainAppLibraryOfflineABInfo : Swift_SwiftObject @path/libswiftCore.dylib
32 {
33     @+80010 name (NSString)
34     @+80020 mimeType (NSString)
35     @+80030 experimentList (NSArray)
36     @+80040 userInfo (NSMutableDictionary)
37 }
38
39 @+801835c148 _TTC14MainAppLibraryOfflineABInfo : Swift_SwiftObject @path/libswiftCore.dylib
40 {
41     @+80010 list (NSArray)
42 }
43
44 @+801835c148 _TTC14MainAppLibraryOfflineABInfo : Swift_SwiftObject @path/libswiftCore.dylib
45 {
46     @+80010 onUpdate (NSNumber)
47     @+80020 abProperties (NSMutableDictionary)
48     @+80030 cellDelegate (NSObject)
49     @+80040 presentationController (UIViewController)
50 }
51
52 @+801835c148 _TTC14MainAppLibraryOfflineABInfo : Swift_SwiftObject @path/libswiftCore.dylib

```

```

C:\OfflineAB\WRegistrationManager.h
1 WRegistrationManager : NSObject @usr/lib/libobjc.A.dylib
2 {
3     @+80008 @WRegistrationManagerDependencyProvider *provider (NSObject)
4     @+80008 @WRegistrationManagerDependencyProvider *connection (NSObject)
5     @+80008 @WRegistrationManagerDependencyProvider *keychainWrapper (NSObject)
6     @+80008 @WRegistrationManagerDependencyProvider *cellProvider (NSObject)
7     @+80008 @WRegistrationManagerDependencyProvider *deviceFrontingProviderStore (NSObject)
8     @+80008 @WRegistrationManagerDependencyProvider *preferences (NSMutableDictionary)
9     @+80008 @WRegistrationManagerDependencyProvider *registrationUpdateProgress (NSNumber)
10     @+80008 @WRegistrationManagerDependencyProvider *registrationUpdateDate (NSDate)
11     @+80008 @WRegistrationManagerDependencyProvider *lastRegistrationUpdateDate (NSDate)
12     @+80008 @WRegistrationManagerDependencyProvider *lastRegistrationUpdateDate (NSDate)
13     @+80008 @WRegistrationManagerDependencyProvider *lastRegistrationUpdateDate (NSDate)
14     @+80008 @WRegistrationManagerDependencyProvider *lastRegistrationUpdateDate (NSDate)
15 }
16 @property (readonly, nonatomic) WKChangeObserver *changeObserver
17 @property (readonly, nonatomic) WKChangeObserver *registrationToken
18 @property (readonly, nonatomic) NSError *error
19 @property (strong, nonatomic) WRegistrationManager *wRegistrationManager
20
21 // instance methods
22
23 @+801825755A - (void)requestProvisionalPushNotificationRegistration:(id)arg1 userContext:(id)arg2 completion:(id)arg3
24 @+801825755A - (void)requestProvisionalPushNotificationRegistration:(id)arg1 data:(id)arg2 error:(id)arg3 userContext:(id)arg4 completion:(id)arg5
25 @+8018254608 - (id)backupToken
26 @+8018254608 - (id)restoreToken
27 @+8018254608 - (id)restoreToken
28 @+8018254608 - (id)restoreToken
29 @+8018254608 - (id)restoreToken
30 @+8018254608 - (id)restoreToken
31 @+8018254608 - (id)restoreToken
32 @+8018254608 - (id)restoreToken
33 @+8018254608 - (id)restoreToken
34 @+8018254608 - (id)restoreToken
35 @+8018254608 - (id)restoreToken
36 @+8018254608 - (id)restoreToken
37 @+8018254608 - (id)restoreToken
38 @+8018254608 - (id)restoreToken
39 @+8018254608 - (id)restoreToken
40 @+8018254608 - (id)restoreToken
41 @+8018254608 - (id)restoreToken
42 @+8018254608 - (id)restoreToken
43 @+8018254608 - (id)restoreToken
44 @+8018254608 - (id)restoreToken
45 @+8018254608 - (id)restoreToken
46 @+8018254608 - (id)restoreToken
47 @+8018254608 - (id)restoreToken
48 @+8018254608 - (id)restoreToken
49 @+8018254608 - (id)restoreToken
50 @+8018254608 - (id)restoreToken
51 @+8018254608 - (id)restoreToken
52 @+8018254608 - (id)restoreToken

```

- 以及：函数（实例函数等），也带：「二进制内偏移地址」也方便动态调试时直接计算函数地址
- 有些类
 - class-dump
 - 看不出类型
 - 看不出偏移量和大小
 - dsdump
 - 也看不出类型
 - 但能看出偏移量和大小

cifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新： 2025-01-01 23:06:29

常见问题

TODO:

- 【已解决】class-dump导出Framework二进制AwemeCore报错: Cannot find offset for address in dataOffsetForAddress
- 【未解决】Mac中无法删除临时目录出现没有权限Operation not permitted
- 【已解决】砸壳后抖音ipa安装失败: DeviceNotSupportedByThinning

使用心得

从 **Generated by class-dump** 可以看出原始用到了 **class-dump**

之前从WebDriverAgent的源码中看到很多头文件的头部都有: `Generated by class-dump`

举例:

```
refer/WebDriverAgent/PrivateHeaders/XCTest/XCTestDriver.h
```

```
//  
//   Generated by class-dump 3.5 (64 bit).  
//  
//   class-dump is Copyright (C) 1997-1998, 2000-2001, 2004-2013 by Steve Nygard.  
//
```

-》说明这些文件都是通过 `class-dump` 从库文件中导出生成的。

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新: 2025-01-01 22:56:41

class-dump导出头文件是空的

- 现象：抖音的二进制 `AwemeCore`，用原版 `class-dump` 导出头文件是空的
- 原因：抖音内部代码应该是 `Swift` 和 `ObjC` 混编代码，原版 `class-dump` 只支持 `ObjC` 的，不支持 `Swift` 和 `ObjC` 混编，所以导出是空的。
- 解决办法：找支持Swift的版本的class-dump去导出，即可

- 比如

-

- 概述

- 用 `MonkeyDev`的`class-dump`

- <https://github.com/AloneMonkey/MonkeyDev/blob/master/bin/class-dump>

```
MonkeyDev/bin/class-dump --arch arm64 -H AwemeCore -o crifan/Aweme/class_dump_output
```

- 详解

- [MonkeyDev的class-dump](#)

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新: 2025-01-01 22:38:38

附录

下面列出相关参考资料。

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新: 2025-01-01 22:00:14

参考资料

- 【记录】支持iOS的Swift和ObjC混编的class-dump
- 【已解决】iOS逆向：导出头文件信息对比：dsdump vs class-dump
-

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新: 2025-01-01 22:45:25