

# 目录

前言	1.1
MonkeyDev概览	1.2
环境搭建	1.3
初始化MonkeyDev	1.3.1
常见问题	1.3.1.1
用MonkeyDev调试ipa	1.3.2
常见问题	1.3.2.1
自身包含	1.4
class-dump	1.4.1
LLDBTools	1.4.2
心得	1.5
内部脚本逻辑	1.5.1
项目代码结构	1.5.2
附录	1.6
参考资料	1.6.1

# iOS逆向开发：MonkeyDev调试

- 最新版本: v0.8.9
- 更新时间: 20231013

## 简介

整理iOS逆向开发中动态调试和插件tweak开发都会涉及到的工具MonkeyDev。先是概览；然后介绍环境搭建，包括初始化安装MonkeyDev，以如何及用Xcode+MonkeyDev去动态调试YouTube的ipa的过程；然后介绍MonkeyDev内部包含的内容，class-dump、LLDBTools等；然后总结心得，包括内部脚本逻辑、项目代码结构。

## 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\\_monkeydev\\_debug: iOS逆向开发：MonkeyDev调试](#)

### 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逆向开发：MonkeyDev调试 book.crifan.org](#)
- [iOS逆向开发：MonkeyDev调试 crifan.github.io](#)

### 离线下载阅读

- [iOS逆向开发：MonkeyDev调试 PDF](#)
- [iOS逆向开发：MonkeyDev调试 ePUB](#)
- [iOS逆向开发：MonkeyDev调试 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： 2023-10-14 23:23:43

# MonkeyDev概览

iOS逆向开发期间，其中常会涉及到动态调试和写tweak插件，其中有个很好用的工具就是：MonkeyDev

- MonkeyDev
  - 是什么：iOS逆向开发的成套工具
  - 概述：iOSOpenDev的升级版 = 集成XCode和其他各种工具的更强的集成环境
  - 一句话描述：一个基于Xcode模块技术快速开发越狱和非越狱插件的工具，可以自动完成逆向中的固定步骤，一键集成非越狱插件，大大提升逆向分析和开发效率
  - 形式：Xcode的一个插件，可以新建MonkeyDev的相关不同类型的项目，做相关的逆向开发
  - 典型的用途
    - 砸壳出ipa后，用MonkeyDev+Xcode去动态调试
    - 用MonkeyDev去写（iPhone越狱后的）tweak插件
  - 主要包含模块
    - Logos Tweak
      - 使用theos提供的logify.pl工具将.xm文件转成.mm文件进行编译，集成了CydiaSubstrate，可以使用MSHookMessageEx和MSHookFunction来Hook OC函数、C/C++函数或指定地址
    - CaptainHook Tweak
      - 使用CaptainHook提供的头文件进行OC函数的Hook，以及属性的获取
    - Command-line Tool
      - 可以直接创建运行于越狱设备的命令行工具
    - MonkeyApp
      - 自动给第三方应用集成Reveal、Cycrypt和注入dylib的模块，支持调试dylib和第三方应用，支持Pod给第三方应用集成SDK，只需要准备一个砸壳后的ipa或者app文件即可
    - MonkeyPod
      - 将自动开发的非越狱插件制造成Pod以供其它人通过pod的方法来使用
    - MonkeyAppMac
      - 针对Mac逆向开发的模块，可以自动集成substitute，注入以及符号还原工作
- 官网
  - Github
    - AloneMonkey/MonkeyDev: CaptainHook Tweak、Logos Tweak and Command-line Tool、Patch iOS Apps, Without Jailbreak.
      - <https://github.com/AloneMonkey/MonkeyDev>
    - wiki
      - <https://github.com/AloneMonkey/MonkeyDev/wiki>

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：2022-11-08 11:47:50

# 环境搭建

TODO:

- 【记录】研究YouTube广告拦截导致视频从头播放的原因：XCode+MonkeyDev动态调试
  - 【已解决】Xcode调试越狱iPhone6中的YouTube
  - 【记录】恢复iOS逆向Xcode调试YouTube的开发环境
  - 【记录】恢复自己Mac的iOS逆向开发环境
  - 【已解决】自己Mac中恢复和重建Xcode的MonkeyDev开发环境
  - 【未解决】用XCode和MonkeyDev去调试iOS抖音app
- 

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: 2022-11-03 14:59:04

# 初始化MonkeyDev开发环境

## 安装路径/opt不能变

后续的 `MonkeyDev`、`theos` 等的安装路径选择，虽然按道理可以自定义，但是此处内部很多脚本貌似只支持固定的默认的路径

所以，只能安装到默认的固定路径：

- `/opt/MonkeyDev`
- `/opt/theos`

而不能轻易改变路径，否则后续会出现很多诡异的问题

初始化搭建MonkeyDev环境=初始化安装MonkeyDev：

- 下载theos

```
sudo git clone --recursive https://github.com/theos/theos.git /opt/theos
```

- 下载MonkeyDev（到固定位置：`/opt/MonkeyDev`）

```
sudo git clone https://github.com/AloneMonkey/MonkeyDev.git /opt/MonkeyDev
```

- 本地运行脚本去安装

```
cd MonkeyDev/bin
sudo bash md-install
```

## 搭建好的环境，对应目录的文件

```
crifan@licrifandeMacBook-Pro ~ % ll /opt/MonkeyDev
total 88
drwxr-xr-x  7 root  wheel  224B  6 28 22:01 Frameworks
-rw-r--r--  1 root  wheel   34K  6 28 22:26 LICENSE
drwxr-xr-x  3 root  wheel   96B  6 28 22:01 Librarys
drwxr-xr-x  4 root  wheel  128B  6 28 22:01 MFrameworks
-rw-r--r--  1 root  wheel   1.7K  6 28 22:26 README.md
drwxr-xr-x  3 root  wheel   96B  6 28 22:01 Resource
drwxr-xr-x  4 root  wheel  128B  6 28 22:01 Tools
drwxr-xr-x 12 root  wheel  384B  6 28 22:07 bin
-rw-r--r--  1 root  wheel  802B  6 28 22:26 change.log
drwxr-xr-x  4 root  wheel  128B  6 28 22:01 include
drwxr-xr-x 14 root  wheel  448B  6 28 22:03 templates

crifan@licrifandeMacBook-Pro ~ % ll /opt/theos
total 112
-rw-r--r--  1 root  wheel   5.1K  6 28 21:59 CODE_OF_CONDUCT.md
-rw-r--r--  1 root  wheel   35K  6 28 21:59 LICENSE.md
-rw-r--r--  1 root  wheel   1.0K  6 28 21:59 Prefix.pch
-rw-r--r--  1 root  wheel   3.1K  6 28 21:59 README.md
drwxr-xr-x 17 root  wheel  544B  6 28 21:59 bin
drwxr-xr-x  3 root  wheel   96B  6 28 21:59 extras
drwxr-xr-x  3 root  wheel   96B  6 28 21:59 include
drwxr-xr-x  3 root  wheel   96B  6 28 21:59 lib
drwxr-xr-x 28 root  wheel  896B  6 28 21:59 makefiles
drwxr-xr-x  3 root  wheel   96B  6 28 21:59 mod
-rw-r--r--  1 root  wheel  657B  6 28 21:59 package.json
drwxr-xr-x  3 root  wheel   96B  6 28 21:59 sdks
drwxr-xr-x  3 root  wheel   96B  6 28 21:59 templates
drwxr-xr-x  3 root  wheel   96B  6 28 21:59 toolchain
drwxr-xr-x  8 root  wheel  256B  6 28 21:59 vendor
```



## 常见问题

此处整理MonkeyDev环境初始化期间的常见问题。

### curl: (7) Failed to connect to raw.githubusercontent.com port 443: Connection refused

```
curl: (7) Failed to connect to raw.githubusercontent.com port 443: Connection refused
Failed to download https://raw.githubusercontent.com/AloneMonkey/frida-ios-dump/3.x/dump.py to /opt/MonkeyDev/bin/dump.py
```

解决办法：

另外单独下载 frida-ios-dump：

```
git clone https://github.com/AloneMonkey/frida-ios-dump.git
```

然后把其中的 dump.py 和 dump.js 拷贝到 /opt/MonkeyDev/bin/

->

- /opt/MonkeyDev/bin/dump.py
- /opt/MonkeyDev/bin/dump.js

### Failed to extract /xxx/md-install.gvGnDuMp/file.tar.gz to

```
Failed to extract /var/folders/zz/zyxvpvxq6csfxvn_n000000000000/T/md-install.gvGnDuMp/file.tar.gz to /var/folders/zz/zyxvpvxq6csfxvn_n000000000000/T/md-install.KQ1lUKhp
```

解决办法：

自己新建一个临时目录：

```
mkdir -p /tmp/md_install/tempdirs
```

改 bin/md-install 为：

```
# export tempDirsFile=`mktemp -d -t $scriptName`/tempdirs"
export tempDirsFile="/tmp/md_install/tempdirs"
```

### Failed to echo into

错误现象：

```
line 82行: Failed to echo into
```

解决办法：

注释掉

```
# echo "$tempDir" >> "$tempDirsFile" || \
```

```
#     panic $? "Failed to echo into $tempDirsFile"
```

## File /xxx/Specifications/MacOSX Package Types.xcspec not found

```
→ bin sudo bash md-install
...
File /Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/Library/Xcode/Specifications/MacOSX Package Types.xcspec not found
```

解决办法：

- Xcode <13
  - 背景：存在 MacOSX Package Types.xcspec，只是路径不对
  - 解决办法：改动路径或换用软链接
- Xcode 13+
  - 背景：不存在 MacOSX Package Types.xcspec（和 MacOSX Product Types.xcspec），所以要去网上下载后，再去：改动路径或换用软链接
    - 下载 MacOSX Package Types.xcspec 和 MacOSX Product Types.xcspec
      - qbs/share/qbs/modules/bundle at master · qbs/qbs 中，下载
      - <https://github.com/qbs/qbs/blob/master/share/qbs/modules/bundle/MacOSX-Package-Types.xcspec>
        - 保存为：MacOSX Package Types.xcspec
      - <https://github.com/qbs/qbs/blob/master/share/qbs/modules/bundle/MacOSX-Product-Types.xcspec>
        - 保存为：MacOSX Product Types.xcspec
    - 拷贝到（旧版Xcode中对应的）目录
      - 录：/Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/Library/Xcode/PrivatePlugIns/IDEOSXSupportCore.ideplugin/Contents/Resources

然后继续去操作：

- 【推荐】方法1：使用软链接

```
sudo ln -s /Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/Library/Xcode/PrivatePlugIns/IDEOSXSupportCore.ideplugin/Contents/Resources /Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/Library/Xcode/Specifications
```

- 方法2：（修改 md-install 脚本）改动路径

修改 /opt/MonkeyDev/bin/md-install

修改路径，改为：

```
# macosxSDKSpecificationsPath=$macosSdkPlatformPath/Developer/Library/Xcode/Specifications
# packageTypesForMacOSXPath="$macosxSDKSpecificationsPath/MacOSX Package Types.xcspec"
# productTypesForMacOSXPath="$macosxSDKSpecificationsPath/MacOSX Product Types.xcspec"
macosxSDKSpecificationsPath "$macosSdkPlatformPath/Developer/Library/Xcode/PrivatePlugIns"
packageTypesForMacOSXPath="$macosxSDKSpecificationsPath/IDEOSXSupportCore.ideplugin/Contents/Resources/MacOSX Package
Types.xcspec"
productTypesForMacOSXPath="$macosxSDKSpecificationsPath/IDEOSXSupportCore.ideplugin/Contents/Resources/MacOSX Product
Types.xcspec"
```

最后重新运行：

```
sudo bash md-install
```

即可

## File /xxx/IDEiOSSupportCore.ideplugin/xxx/Embedded-Device.xcspec not found

- 问题：

Xcode 14.3.1的Mac中，报错：

```
→ bin sudo bash md-install
...
File /Applications/Xcode.app/Contents/PlugIns/IDEiOSSupportCore.ideplugin/Contents/Resources/Embedded-Device.xcspec not found
```

- 原因：xcode 13+之后，部分路径变化了，所以找不到对应路径
- 解决办法：从Xcode中搜索到Embedded-Device.xcspec的实际位置，然后拷贝到报错的路径（如果不存在，先创建对应目录）即可
- 具体步骤

(1) 找到Embedded-Device.xcspec

```
→ ~ cd /Applications/Xcode.app/Contents
→ Contents find . -name Embedded-Device.xcspec
./Developer/Library/Xcode/Plug-ins/XCBSpecifications.ideplugin/Contents/Resources/Embedded-Device.xcspec
```

找到：

- ./Applications/Xcode.app/Contents/Developer/Library/Xcode/Plug-ins/XCBSpecifications.ideplugin/Contents/Resources/Embedded-Device.xcspec

(2) 拷贝到报错目录

先新建该目录

```
sudo mkdir -p /Applications/Xcode.app/Contents/PlugIns/IDEiOSSupportCore.ideplugin/Contents/Resources/
```

再去拷贝：

```
sudo cp ./Applications/Xcode.app/Contents/Developer/Library/Xcode/Plug-ins/XCBSpecifications.ideplugin/Contents/Resources/Embedded-Device.xcspec /Applications/Xcode.app/Contents/PlugIns/IDEiOSSupportCore.ideplugin/Contents/Resources/
```

确认文件的确存在：

```
→ PlugIns ll /Applications/Xcode.app/Contents/PlugIns/IDEiOSSupportCore.ideplugin/Contents/Resources/
total 8
-rw-r--r--@ 1 root  wheel  437B 10 12 15:34 Embedded-Device.xcspec
```

最后重新去操作：

```
sudo bash md-install
```

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：2023-10-13 14:12:52

## 用MonkeyDev调试ipa

### 用Xcode+MonkeyDev去调试砸壳后的YouTube的ipa

- 概述

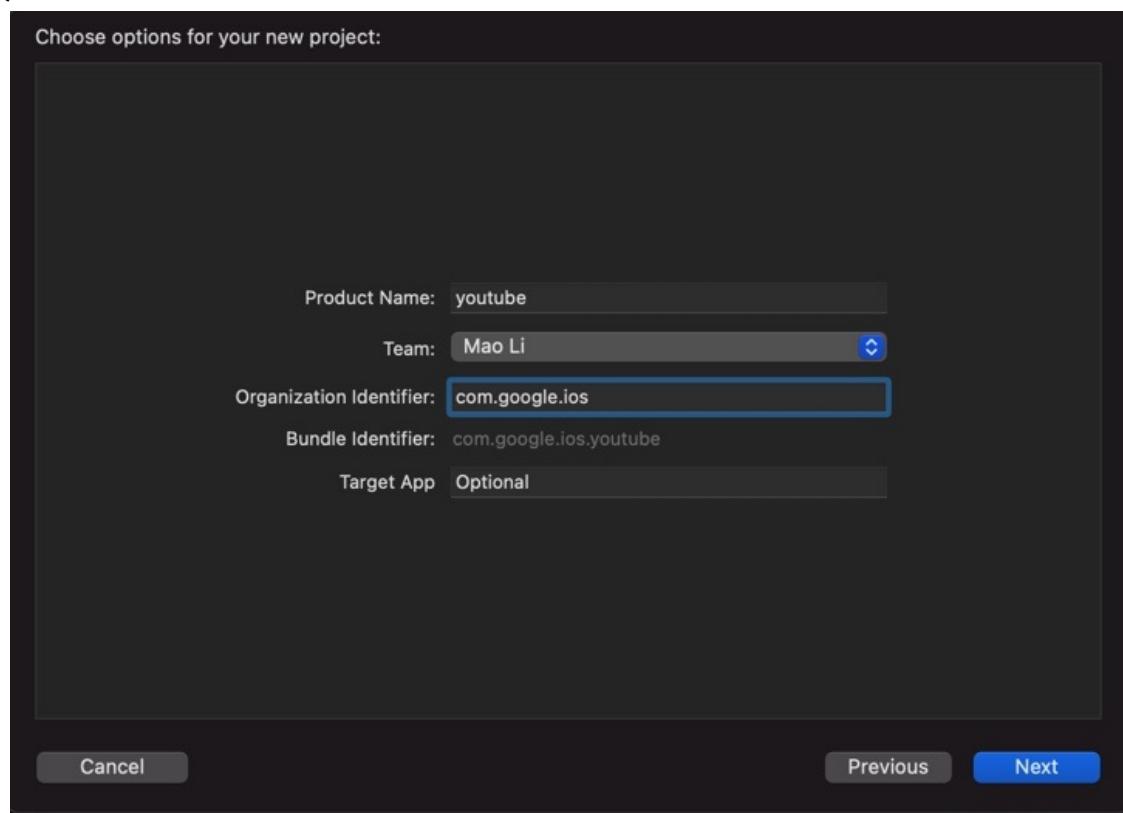
- (1) xcode 新建 MonkeyDev 的 MonkeyApp 项目
- (2) 设置基本参数
  - Product : youtube
  - Organization Identifier : com.google.ios
  - 自动生成包名: com.google.ios.youtube
    - 记得要和app真实包名是一致的
- (3) 右键 TargetApp -> Add Files to youtube ->选择YouTube的 ipa
  - 注意勾选:
    - Destination : Copy Items if needed
      - 表示将ipa拷贝过来, 而不是只是建立引用 (链接)
    - Added folders : Create groups
- (4) 确保已设置合适的目标部署iOS版本
  - 尽量让 PROJECT 和 TARGETS 中的iOS目标的版本一致
    - PROJECT -> ProjectName -> Info -> Deployment Target -> iOS Deployment Target , 比如设置为 iOS 12.0
    - TARGETS -> ProjectName -> General -> Minimum Deployment , 比如设置为 iOS 12.0
- (5) 确保 Targets 是 youtube (而不是youtubeDylib) , 点击▶按钮去启动调试, 即可正常调试
- 如果遇到各种问题
  - Unable to install
    - Could not inspect the application package
    - There was an internal API error
  - 可以:
    - 多试试几次
    - 或 Xcode -> Clean Build Folder , 一般均可解决问题

- 详解:

### 新建MonkeyDev项目

- Xcode中新建项目, 选 MonkeyDev -> MonkeyApp

- - 填写项目信息
    - 效果



- 包名: com.google.ios.youtube
  - Product Name : youtube
  - Organization Identifier : com.google.ios
    - 自动生成包名: com.google.ios.youtube
  - Target App : Optional

- 选择项目保存路径

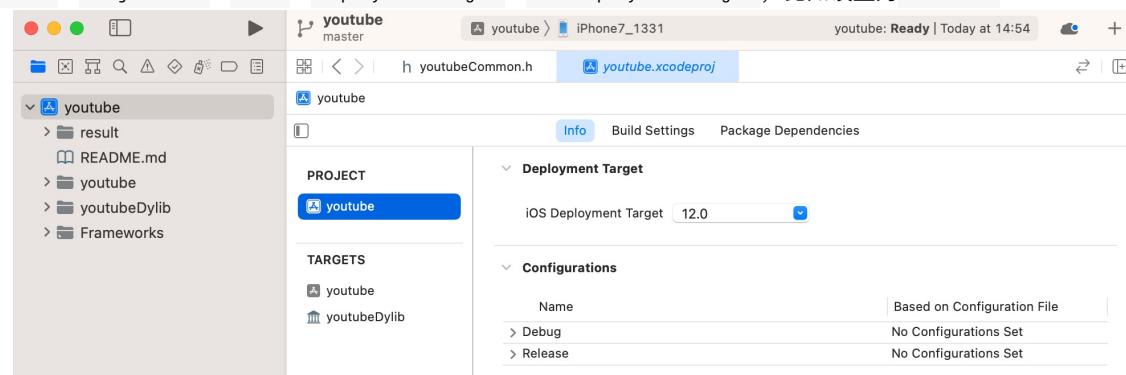
- 此处: /Users/crifan/dev/DevRoot/YoutubeAdsFilter/Xcode/YouTube\_1708
- 新建好了 Xcode + MonkeyDev 的项目

◦

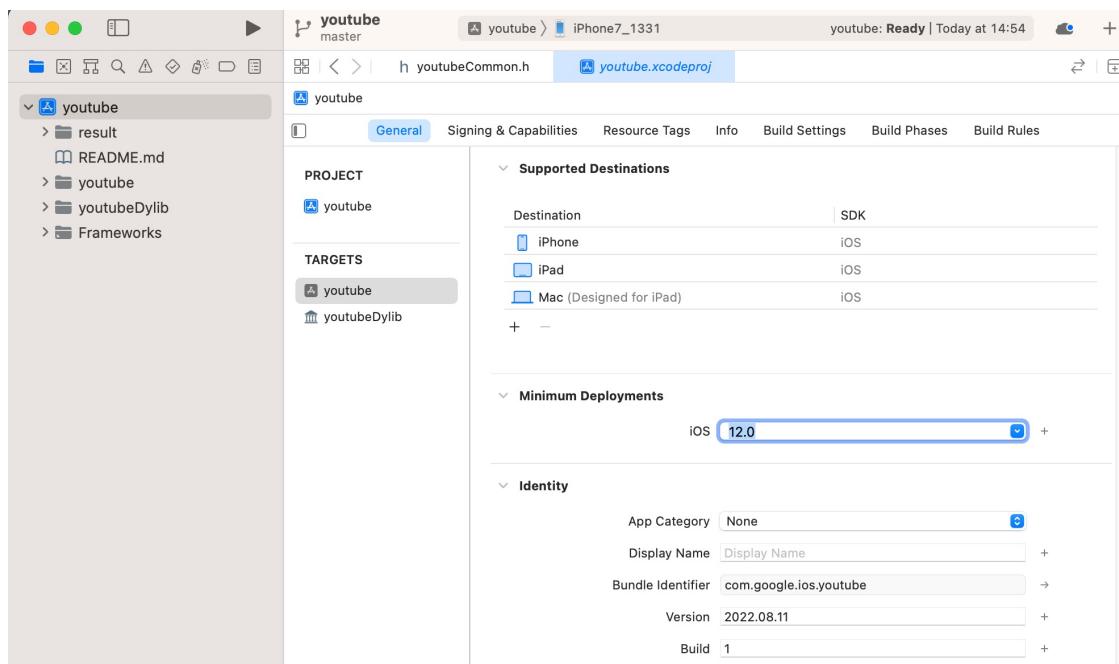
## 确保已设置合适的目标部署iOS版本

- 尽量让 PROJECT 和 TARGETS 中的iOS目标的版本一致

- PROJECT -> ProjectName -> Info -> Deployment Target -> iOS Deployment Target , 比如设置为 iOS 12.0



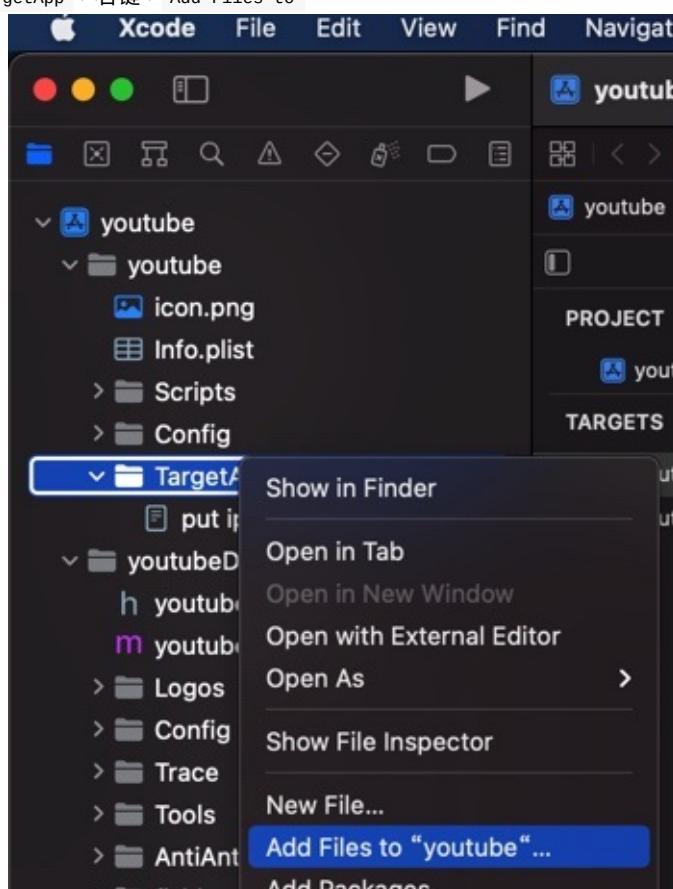
- TARGETS -> ProjectName -> General -> Minimum Deployment , 比如设置为 iOS 12.0



## 导入ipa

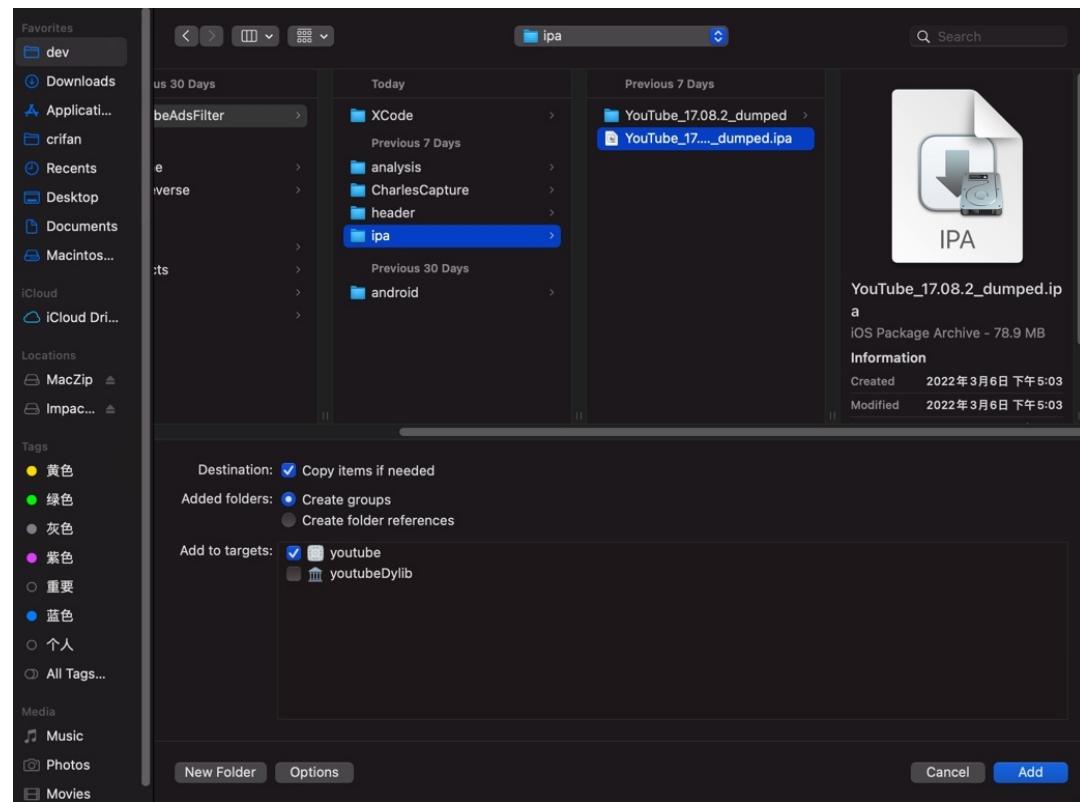
- 添加导入（砸壳后的）ipa

- TargetApp ->右键-> Add Files to



- 选择ipa文件

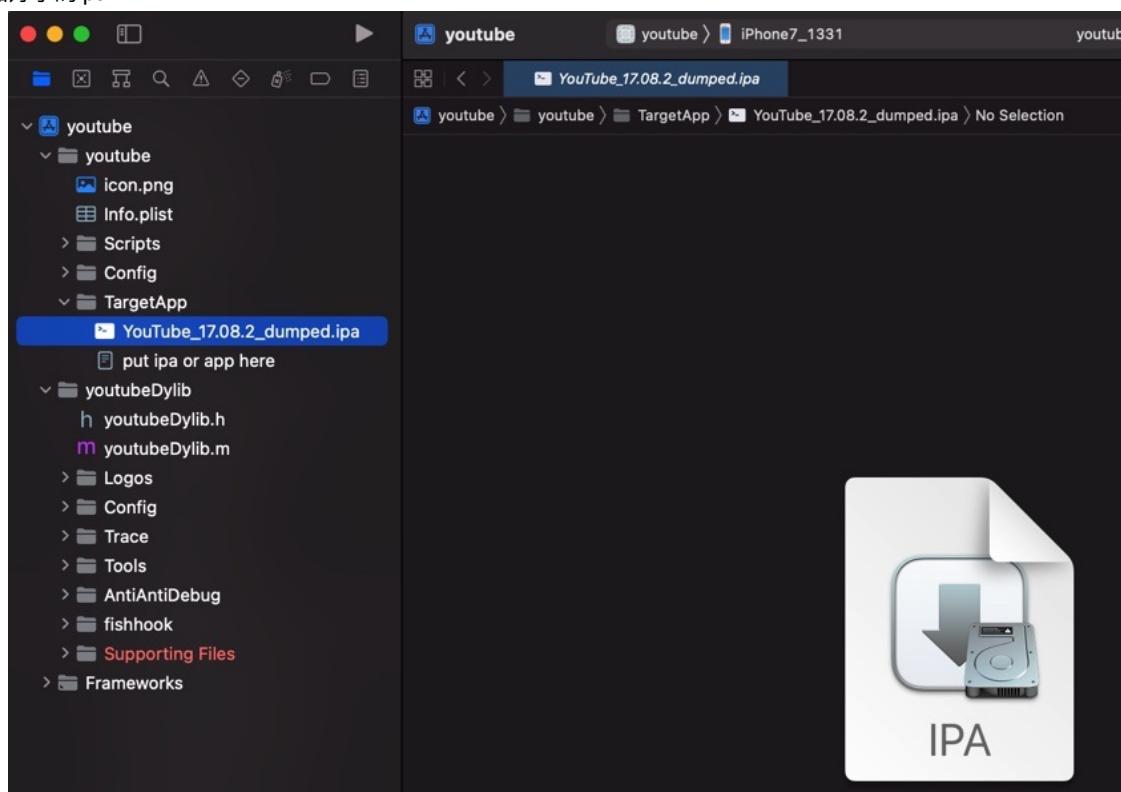
- 图



#### ■ 参数

- Destination : Copy Items if needed
- Added folders : Create groups

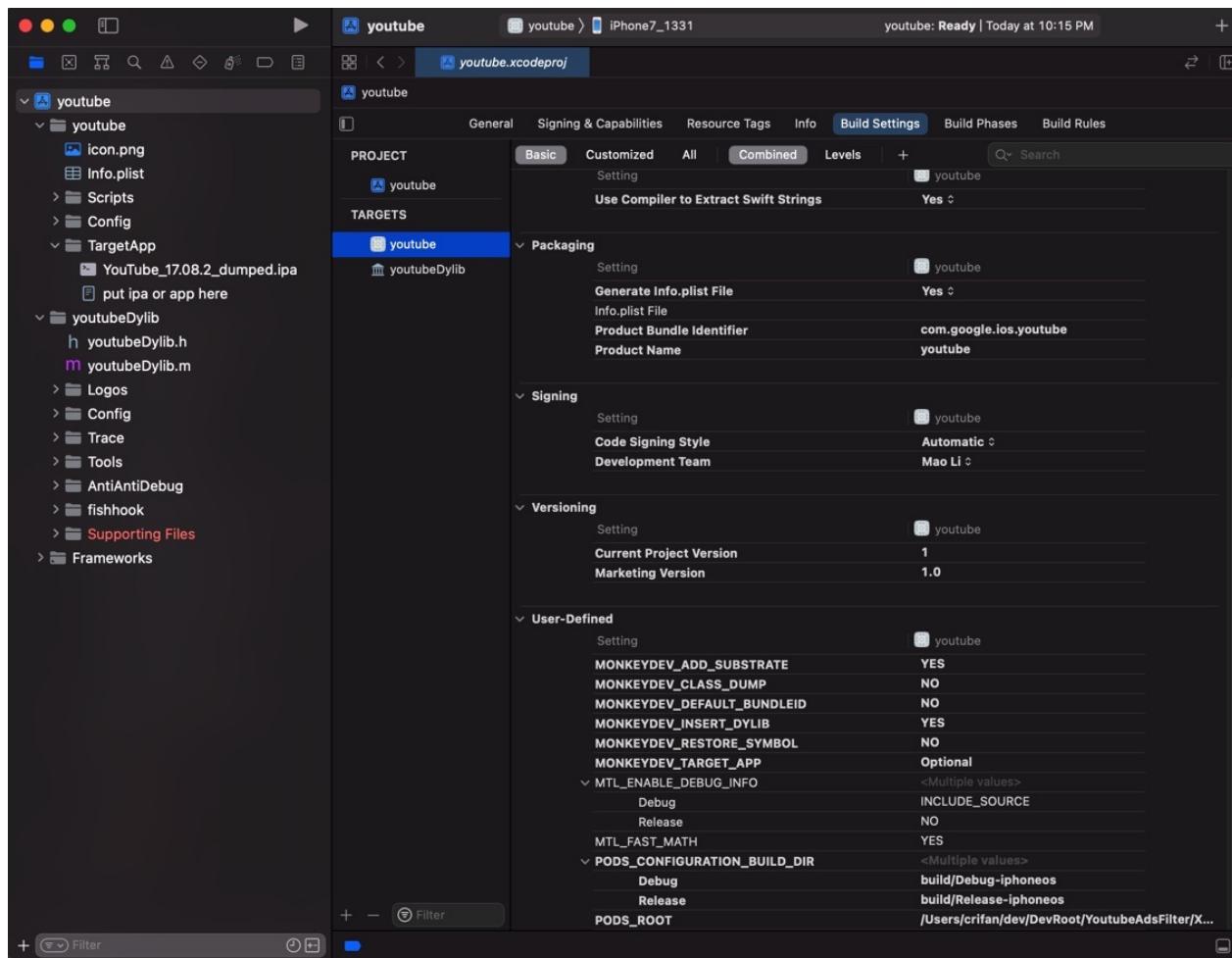
#### ◦ 添加好了的ipa



## 确认（调整）MonkeyDev配置参数

注意：默认的 TARGETS 是 youtubeDylib，要先去切换过去 TARGETS -> youtube，才能看到配置。

去 TARGETS -> youtube 中确认此处MonkeyDev的配置参数（是你所希望的）：



此处参数配置值（多数是默认值）是：

- MONKEYDEV\_ADD\_SUBSTRATE = YES
- MONKEYDEV\_CLASS\_DUMP = NO
- MONKEYDEV\_DEFAULT.BundleID = NO
- MONKEYDEV\_INSERT\_DYLIB = YES
- MONKEYDEV\_RESTORE\_SYMBOL = NO
- MONKEYDEV\_TARGET\_APP = Optional

## 开始调试ipa

注意：默认的 TARGETS 是 youtubeDylib，要先去切换过去 TARGETS -> youtube，才能正常运行，安装ipa，开始调试。

然后Xcode中即可去调试运行ipa：

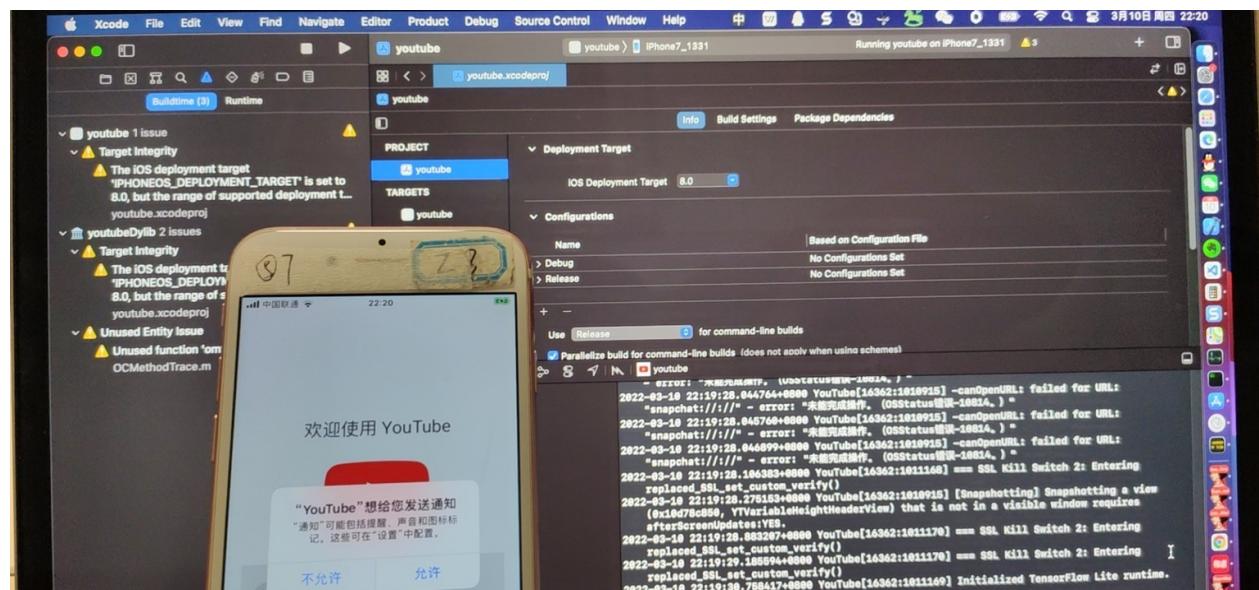
- Building

- - Installing

- - Running

◦

然后可以在 iPhone 真机上调试 YouTube 了：



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: 2023-10-13 15:05:16

# 常见问题

此处整理MonkeyDev调试编译运行ipa期间的常见问题。

## Id: file not found: /usr/lib/libstdc++.dylib

- 问题

MonkeyDev编译链接时报错：

```
ld: file not found: /usr/lib/libstdc++.dylib
```

- 原因：Xcode 10+ 之后=新版XCode，没了 /usr/lib/libstdc++.dylib
- 解决办法：网上找到缺失的 /usr/lib/libstdc++.dylib，再安装拷贝到对应目录即可。
- 具体步骤

网上有人弄了个仓库，专门干这事。所以去下载代码和运行对应脚本即可。

```
git clone https://github.com/devdawei/libstdc+.git
cd libstdc+
chmod +x install-xcode_11+.sh
./install-xcode_11+.sh
```

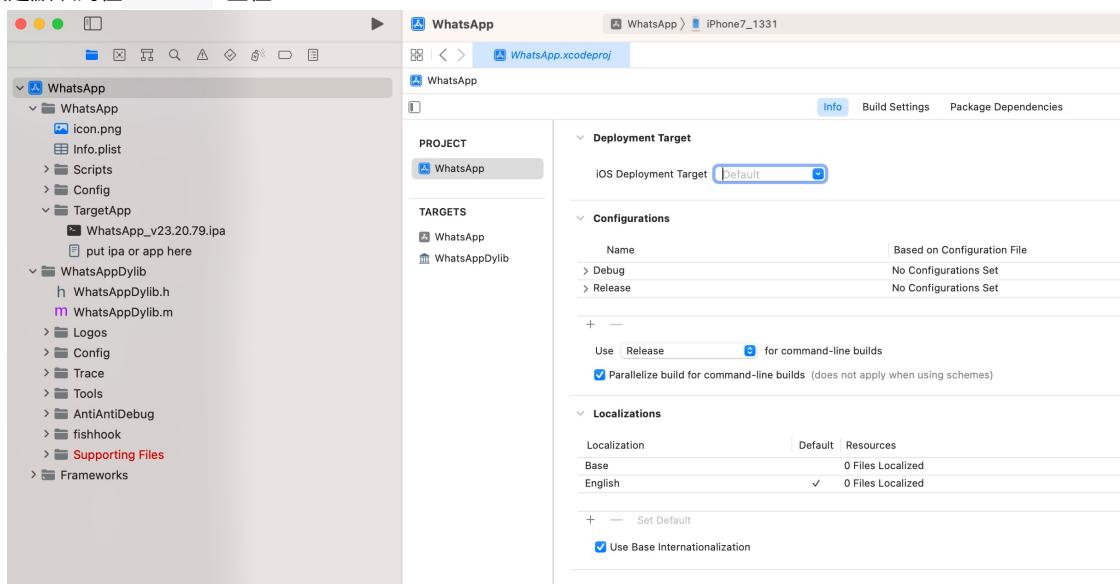
## File not found: /xxx/arc/libarclite\_iphoneos.a

- 问题

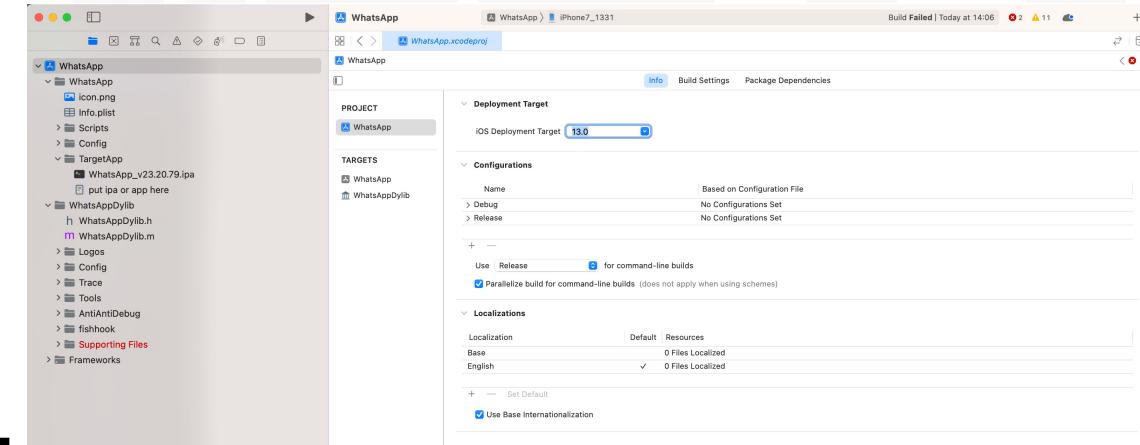
MonkeyDev调试ipa报错：

```
File not found: /Applications/Xcode.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/lib/arc/libarclite_iphoneos.a
```

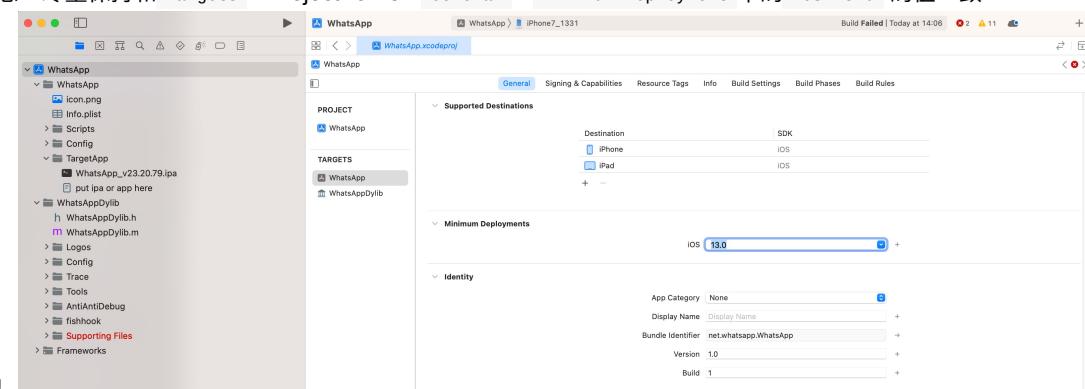
- 原因：没有设置PROJECT中的iOS部署目标的iOS版本
- 导致是默认的值 Default =空值



- 解决办法：去设置PROJECT中的iOS部署目标的iOS版本
- 具体操作
  - 把项目的 PROJECT -> Info -> Deployment Target -> iOS Deployment Target 从默认的 default =空值，改为 iOS 13.0



■ 注意：尽量保持和 Targets ->ProjectName-> General -> Minimum Deployment 中的 iOS 13.0 的值一致



# 自身包含

TODO:

- 要加上其他的?
  - AntiAntiDebug ?
  - trace?

---

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: 2022-11-03 11:57:59

# class-dump

TODO:

- 【记录】支持iOS的Swift和ObjC混编的class-dump
- 【已解决】MonkeyDev安装失败: Failed to download AloneMonkey/frida-ios-dump/3.x/dump.py
- 【已解决】Mac中用class-dump导出YouTube头文件

- class-dump : 是编译好的二进制支持swift混淆的版本
  - 对应路径: /opt/MonkeyDev/bin/class-dump
  - 版本信息

```
→ ~ class-dump --version
class-dump 3.5 (64 bit) (Debug version compiled Sep 17 2017 16:24:48) compiled Sep 17 2017 16:24:48
```

## 让MonkeyDev的class-dump全局可用

此次，之前已安装好 `iOSOpenDev` 的环境和设置了相关的环境变量：

- ~/.zshrc
 

```
export iOSOpenDevPath /opt/iOSOpenDev
export iOSOpenDevDevice
export PATH /opt/iOSOpenDev/bin:$PATH
```

使得此处找到的 `class-dump` 是 `iOSOpenDev` 版本的：

```
→ ~ which class-dump
/opt/iOSOpenDev/bin/class-dump
```

此处想要，把全局的，命令行中找到的 `class-dump` 换成（支持Swift和ObjC混淆的） `MonkeyDev` 的

可以去：设置PATH环境变量，加上MonkeyDev的路径

编辑 `~/.zshrc`，在最末尾加上：

```
export MonkeyDevPath /opt/MonkeyDev
export MonkeyDevDeviceIP
export PATH /opt/MonkeyDev/bin:$PATH
```

保存退出。重启终端，即可实现我们的效果：

```
→ ~ which class-dump
/opt/MonkeyDev/bin/class-dump
```

# LLDBTools

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: 2022-11-03 11:54:46

# 心得

TODO:

- 【整理】iOS越狱插件开发工具：MonkeyDev
- 【未解决】Mac中安装和搭建MonkeyDev+XCode的开发环境
- 【已解决】MonkeyDev安装失败：Failed to download AloneMonkey/frida-ios-dump/3.x/dump.py
- 【已解决】MonkeyDev安装报错：tar Error Failed to extract md-install file.tar.gz
- 【已解决】MonkeyDev的XCode项目编译报错：codesign\_allocate error failed with exit code 34304 errno No such file or directory
- 【已解决】MonkeyDev的XCode编译：始终弹框安装codesign\_allocate命令行工具
- 【已解决】XCode启动崩溃：Failed to register spec from DEiOSSupportCore.ideplugin couldn't register specification malformed property list dictionary required key Identifier not present
- 【已解决】MonkeyDev的XCode项目编译报错：Unable to install This application's application-identifier entitlement does not match that of the installed application
- 
- 【记录】用XCode和MonkeyDev调试Logos越狱插件代码的效果
- 【已解决】用XCode和MonkeyDev去调试iOS抖音app
- 【未解决】给MonkeyDev的pack.sh加上echo的log日志调试分析运行逻辑
- 【记录】分析XCode+MonkeyDev编译抖音ipa详细过程的log
- 【未解决】XCode+MonkeyDev调试iOS的ipa除了首次外后续调试均会异常
- 【基本解决】Mac中用MonkeyDev+XCode去调试抖音脱壳ipa

- 
- 每次调试
    - 先Clean再Build：绕过bug，否则导致调试ipa会崩溃
      - 详见：
        - 【已解决】XCode+MonkeyDev调试18.9.0抖音的崩溃问题：先Clean后再调试
  - Xcode中，新增.xm文件的流程
    - 先新增.xm文件，再Build出.mm，再把.mm加到要编译的文件列表
      - 好像还要做一个什么映射还是关联？以便确保 自动从.xm生成.mm？

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：2023-10-13 14:32:30

# 内部脚本逻辑

TODO:

整理下面多个帖子

---

MonkeyDev内部有一套自己的脚本，执行对应的预处理、编译、链接等等流程和逻辑。

下面介绍其中相关内容。

## pack.sh

- 【未解决】XCode+MonkeyDev调试iOS的ipa除了首次外后续调试均会异常
- 【未解决】研究MonkeyDev的XCode中/opt/MonkeyDev/Tools/pack.sh脚本的内部逻辑
- 【未解决】给MonkeyDev的pack.sh加上echo的日志调试分析运行逻辑
- 【记录】研究MonkeyDev中pack.sh中为何info.plist异常缺失图标等字段
- 

## md

- 【已解决】Xcode调试报错：/opt/MonkeyDev/bin/md No such file or directory

## md-install

- 【已解决】Mac中MonkeyDev搭建环境运行md-install报错：File Xcode/Specifications/MacOSX Package Types.xcspec not found
- 【已解决】MonkeyDev安装报错：tar Error Failed to extract md-install file.tar.gz

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：2022-11-03 14:39:29

# 项目代码结构

TODO:

- 【已解决】MonkeyDev的Xcode项目代码优化：新增独立文件youtubeCronet.xm
  - 【已解决】MonkeyDev的Xcode项目代码优化：把公共部分提取到youtubeCommon.h
  - 【已解决】MonkeyDev的Xcode项目代码优化：把hook代码移动到独立文件
  - 【记录】优化MonkeyDev的YouTube代码：把Error部分提取到单独文件
- 

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：2022-11-03 12:01:30

## 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: 2022-11-03 11:53:00

## 参考资料

- 【已解决】XCode+MonkeyDev动态调试YouTube的ipa
- 【已解决】用MonkeyDev和XCode去调试17.8.0的抖音ipa
- 【已解决】Mac中安装和搭建MonkeyDev+XCode的开发环境
- 【已解决】MonkeyDev初始化报错：File  
/Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/Library/Xcode/Specifications/Mac OSX Package Types.xcspec not found
- 【已解决】M2的Mac中给Xcode安装MonkeyDev出错：File  
/Applications/Xcode.app/Contents/Plugins/IDEiOSSupportCore.ideplugin/Contents/Resources/Embedded-Device.xcspec not found
- 【已解决】MonkeyDev的XCode编译报错：ld file not found /usr/lib/libstdc++.dylib
- 【已解决】MonkeyDev编译运行报错：File not found arc libarclite\_iphoneos.a
- 

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：2023-10-13 14:32:17