目录

[1、xcode软件正常打包【过程略】； 2](#_Toc1505794068)

[2、Payload压缩打包： 2](#_Toc1933604628)

[3、脚本代码打包【重点介绍】 2](#_Toc222952745)

[3.1、照下图进行设置： 2](#_Toc1955304847)

[3.2、定位项目的根目录： 3](#_Toc2013797135)

[3.3、脚本文件： 3](#_Toc1546171225)

[3.4、release模式下真机run的意义： 4](#_Toc1947649875)

[3.5、执行脚本命令（漫长的等待...） 5](#_Toc58217904)

[3.6、如果 PackageApplication 未找到的解决方案【非常重要】 5](#_Toc1363253143)

[3.7、完成： 6](#_Toc692544558)

[4、脚本打包时，如何动态的设置预编译宏？ 6](#_Toc235019566)

[4.1、User-defined： 6](#_Toc751418929)

[4.2、preprocessor Macros 7](#_Toc1894095343)

[4.3、运用： 8](#_Toc1910330320)

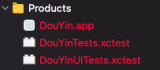
[4.3.1、可以批量动态参数打包 8](#_Toc2041165590)

[4.3.2、可以自定义一个宏 8](#_Toc1966293952)

# 1、xcode软件正常打包【过程略】；

# 2、Payload压缩打包：

相关脚本文件：[Payload压缩打包脚本.sh](相关脚本文件/Payload压缩打包脚本.sh)

拥有一个开发者账号并且真机运行，run成功以后，直接提取项目目录下文件夹下后缀名为.app的文件到一个名为的空文件夹根目录下（注意严格区分大小写Payload），执行压缩操作，压缩完毕修改后缀名为.ipa

# 3、脚本代码打包【重点介绍】

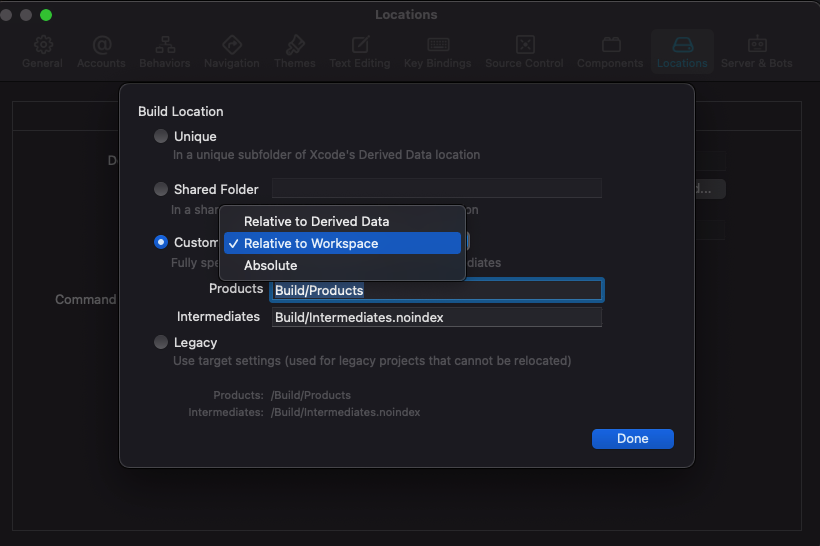
【Payload压缩打包 和 脚本代码打包 的差异化探讨】

两者打出来的包的大小有很大的区别。因为图片是用Xnip2021-04-18_11-35-37，而苹果推崇用Assets.xcassets对图片资源进行管理，因为内部存在所谓的优化，比如某些机型不会使用到2倍图，会自动过滤，而我们Payload压缩打包涉及到具体型号的真机。所以脚本代码打包的兼容性安全性更强，直接完全调用xcode的打包的Api。

【脚本代码打包的流程】

## 3.1、照下图进行设置：

xcode -> preferences -> Locations -> Advanced



## 3.2、定位项目的根目录：



## 3.3、脚本文件：

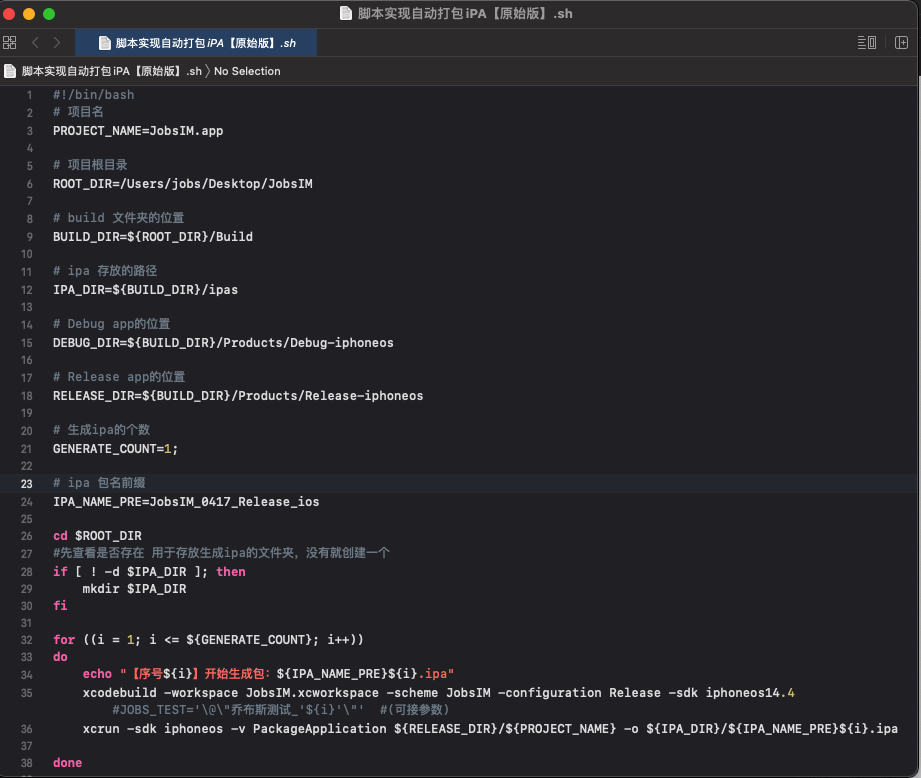
3.3.1、见附件（相关脚本文件）：

[放在项目工程根目录下直接sh运行该脚本实现自动打包iPA + 自动下载PackageApplication【升级版】.sh](相关脚本文件/放在项目工程根目录下直接sh运行该脚本实现自动打包iPA + 自动下载PackageApplication【升级版】.sh)

[脚本实现自动打包iPA【原始版】.sh](相关脚本文件/脚本实现自动打包iPA【原始版】.sh)

其中，放在项目工程根目录下直接sh运行该脚本实现自动打包iPA + 自动下载PackageApplication【升级版】.sh

可以自动下载相关资源文件于指定文件夹，并且允许该自动打包脚本在相关项目工程文件根目录下，直接 *sh 路径* 运行



其中做几点说明：

Xcodebuild 后面有很多参数，这里只做简版操作

-workspace JobsIM.xcworkspace

-scheme JobsIM

-configuration Release

-sdk iphoneos14.4

命令前加前缀- 后面紧跟相关参数

相关【-命令 参数】不分先后次序，但不能提行，只能一行顺序执行

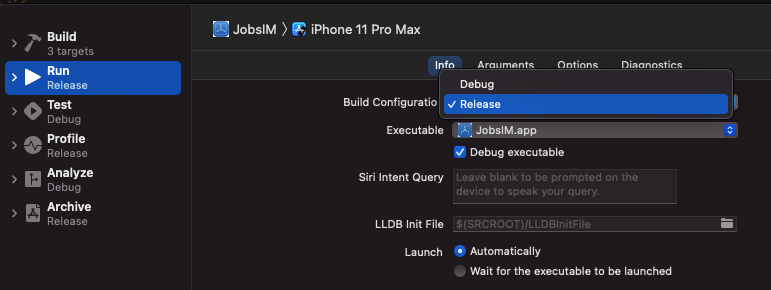
**特别说明：**

**比如：脚本文件里PROJECT\_NAME=JobsIM.app，等号两边不能有空格，否则执行错误**

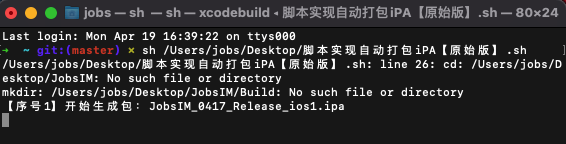
## 3.4、release模式下真机run的意义：

一部分代码在真机是被优化不运行的，比如DEBUG宏;

以前的Mac电脑的芯片是intel，而iPhone芯片是苹果自研芯片，两者指令集不一致，故打包的时候底层代码需要重新编译。



## 3.5、执行脚本命令（漫长的等待...）



## **3.6、如果 PackageApplication 未找到的解决方案【非常重要】**

*xcrun错误：无法找到实用程序“ PackageApplication”，而不是开发人员工具或在PATH中，后面根据对比发现新版的Xcode少了这个PackageApplication（转注：PackageApplication在前几个版本已被标识为废弃，在8.3版本彻底移除了）*

*先去找个旧版的Xcode里面复制一份过来*

放到下面这个目录：

**/Applications/Xcode.app/Contents/Developer/Platforms/iPhoneOS.platform/Developer/usr/bin/**

然后执行命令：

**【命令1】sudo xcode-select -switch /Applications/Xcode.app/Contents/Developer/**

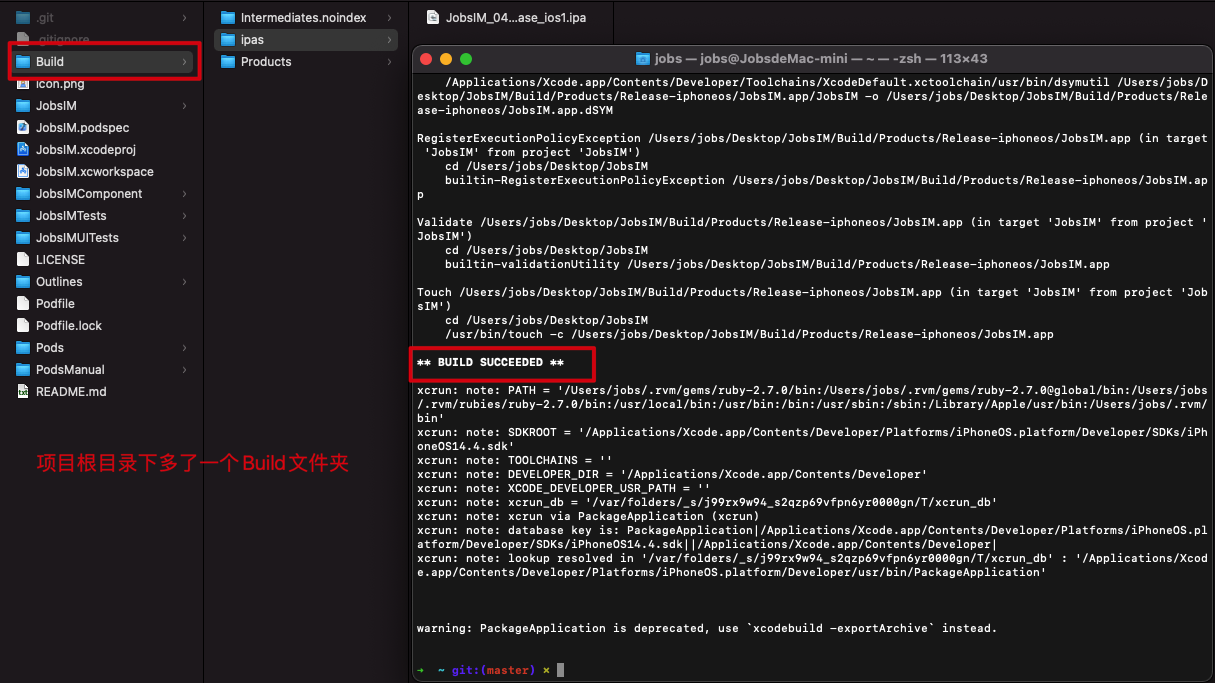
**【命令2】chmod + x /Applications/Xcode.app/Contents/Developer/Platforms/iPhoneOS.platform/Developer/usr/bin/PackageApplication**

这2句分开执行

【附上PackageApplication下载地址】

**https : //github.com/JackSteven/PackageApplication/blob/master/PackageApplication.zip**

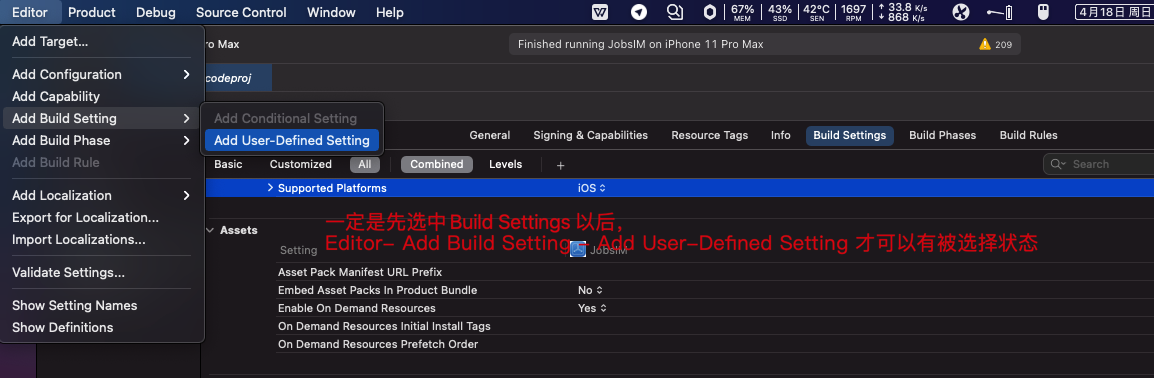
## 3.7、完成：



# 4、脚本打包时，如何动态的设置预编译宏？

## 4.1、User-defined：

首先通过xcode的工具栏中:Editor->Add build Setting添加User-defined值：



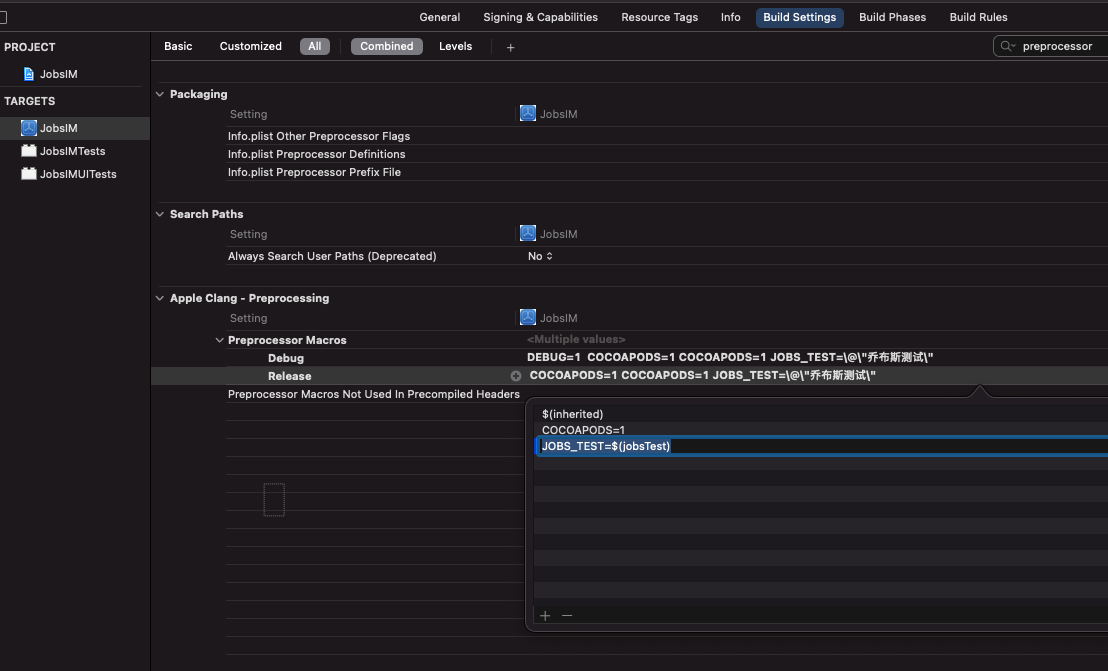


**【注意写法】：如果是字符串，写法规则一定是 \@\"您需要输出的值\"，如果不是字符串，就直接赋值，如: jobsTest = 2**

## 4.2、preprocessor Macros

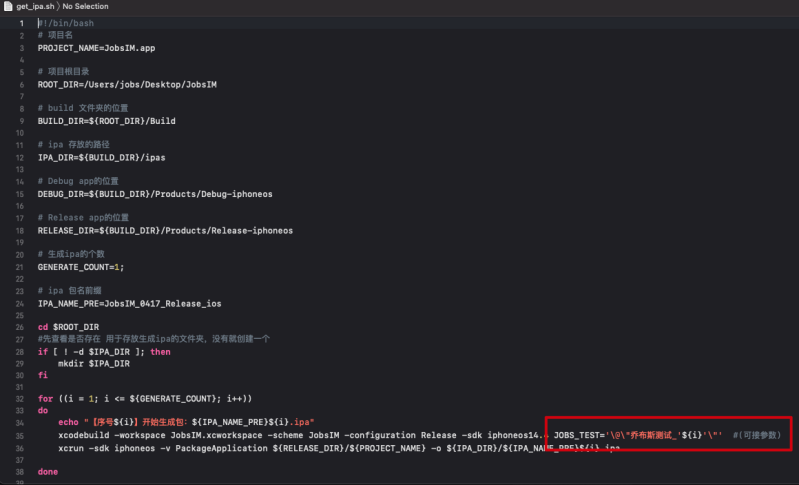
在Build Setting中找到:preprocessor Macros选项，并加入一个变量；

这里也是系统定义DEBUG的地方：



## 4.3、运用：

### 4.3.1、可以批量动态参数打包



### 4.3.2、可以自定义一个宏

