Xcode Make 使用说明

## 导出流程

1. 环境搭建
2. 证书准备
3. 配置make\_config
4. 使用命令行，进入runtime-src/\_\_mack\_config目录
5. sh make.sh
6. 静候导出完毕，在runtime-src/package中有对应的包

注：

1. 为提高xcode\_make适用性，请确保ios工程目录中包含/ios/info.plist和/ios/cConfig.json
2. 因为Xcodebuild需要ruby环境，所以请在正确的ruby环境下使用exporterall.sh（目前使用 rvm use system使用系统ruby环境，但sh必须使用source xx.sh）

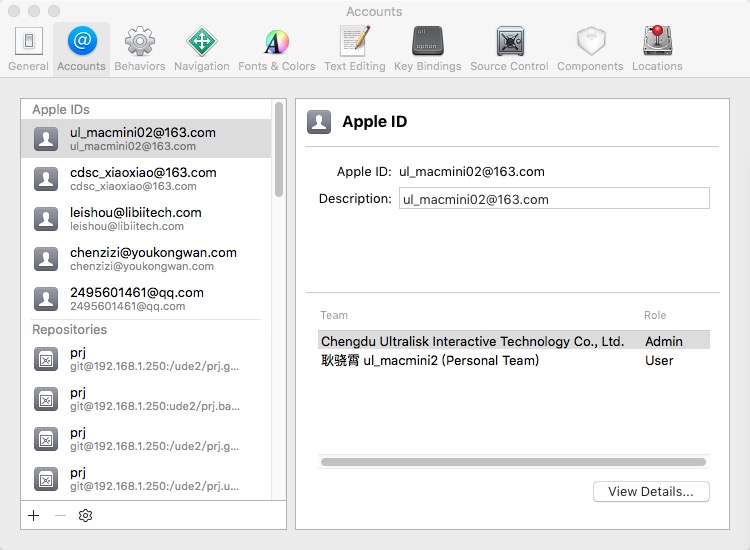
## 环境搭建

1. 运行ude2\prj.ulsdk\UISDK\xcode\_exporter\运行我生成xcode\_make环境.sh
2. 按照提示，输入项目的“根路径”，比如cocos项目中是 client/framework/runtime-src/。这个路径是proj.ios\_mac的伤及路径。
3. 完成后，将在根路径下创建 \_\_make\_config目录，包含make\_config.json和make.sh两个文件。

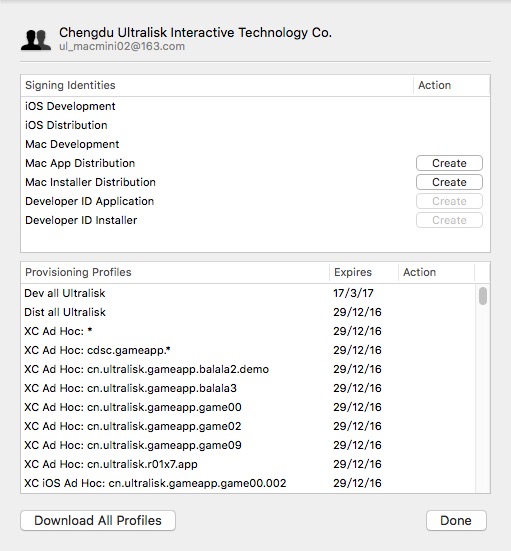
## 2. 证书准备

导出器是基于xcodebuild完成的，对应打包所需的证书，需要提前在xcode中配置好。

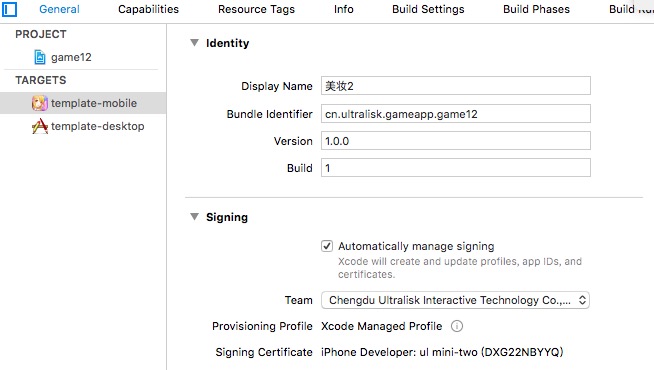
1. 在Xcode-Preferences...-Accounts中，登录对应的AppleID



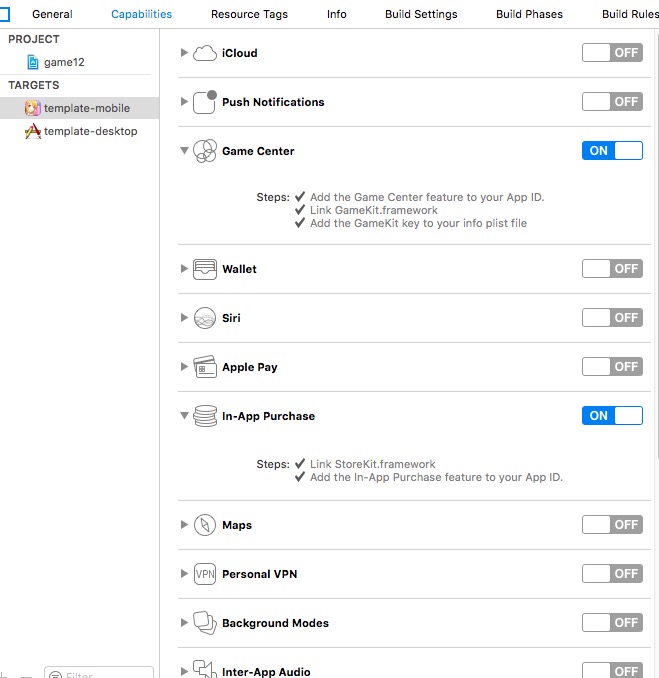
1. 进入账号，查看Signing Identities是否有【iOS Development】和【iOS Distribution】，具体配置方法看\*附录1



1. 进入TARGETS-Gerneral-Signing中，打开【Automatically mangage signing】，选择对应的证书，让xcode自动配置好一切



1. 进入TARGETS-Capablities，打开GameCenter和In-App Purchase



## 3. make\_config配置

make\_config本体是一个json文件，需要参照json的规则进行。

list和dict结尾的元素不能带逗号！[1, 2, 3]

list和dict结尾的元素不能带逗号！[1, 2, 3]

list和dict结尾的元素不能带逗号！[1, 2, 3]

字段说明：

每个渠道会对应一套配置

|  |  |  |
| --- | --- | --- |
| 字段 | 解释 | 范例 |
| b\_skip | 是否忽略，标记为true，则不解析这个渠道的信息 |  |
| b\_export | 是否导出，标记为true，则只生成子工程，不导出ipa包（仅用于调试用） |  |
| project\_name | proj.ios\_mac目录下xxx.xcodeproj中xxx部分 | game12 |
| src\_project\_path | 模板工程的路径，默认为proj.ios\_mac | proj.ios\_mac |
| version | 对应ipa包的version字段，直接修改Info.plist | 1.0.0 |
| build\_version | 对应ipa包的build字段，这个字段最好采用自增的设计。每次上传到appstore的包，build字段不能重复 | 1 |
| team\_id | 打包用的证书对应的team\_id  直接查看make\_config.json头部的rem字段 | KSV9CN5G9G |
| scheme | 对应xcode中的TARGTES的名称 | template-mobile |
| export\_methods | 导出哪些包，目前支持【appstore】【adhoc】 | adhoc |
| resources | 需要导入到resources中的目录。  对应cocos项目，应该包含4个文件：  pack/src  pack/res  pack/config.json  cConfig.json |  |
| 3rdparts | 需要导入到3rdparts目录中的文件，一般为ulsdk所需的目录 |  |
| prepeocessors | 需要添加的宏 |  |
| frameworks | 需要导入的库文件，直接使用名称即可，如【WebKit.framework】 |  |
| header\_search\_paths | 头文件的搜索路径 | 参照模板 |
| framework\_search\_path | framework的搜索目录 | 参照模板 |
| libaray\_search\_path | 库文件的搜索目录 | 参照模板 |
| other\_link\_flags | 需要添加的other link flag字段 | -arc -ObjC |
| info\_plist\_mod | 对Info.plist进行自定义修改  采用字段直接替换的方式  比如需要新增一个key为 my\_name，值为hello的字段，则 mod = {“my\_name”:”hello”} |  |

## 导出

导出器的原理为：

1. 基于模板工程（如proj.ios\_mac），复制出一个子工程（如\_prj.ios\_chance）
2. 根据配置文件，修改子工程（添加文件、修改字段等）
3. 根据导出配置，调用xcode\_exporter导出ipa文件

目前整个make过程直接由make\_config文件配置，如果只需要导出一个渠道的包，则需要手动修改其余渠道的配置，将b\_skip设置为true，操作比较繁琐。

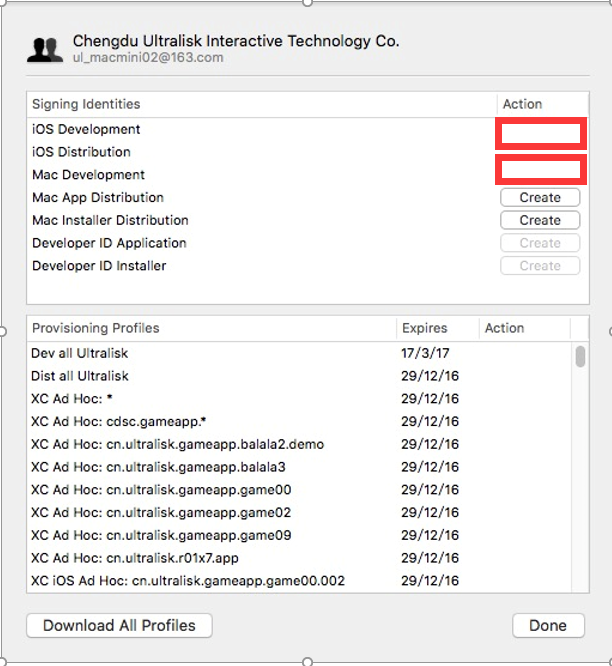
TODO：看看如何方便的使用配置文件。

## 附录1：xcode配置申请方法：

iOS Development

1. 自动创建

在Xcode-Preferences...-Accounts中，直接进入账号配置的地方：



红框的地方会出现【Reset】或【Create】按钮，点击后由xcode自动创建

iOS Distriburtion

由于发布用cer证书，每个Apple开发组只能拥有一个

发布证书需要从ulmacmini02上导出p12文件。

1. 导出，在ulmacmini02中，打开上面截图的证书界面，右键点击iOS Distriburtion-Export，保存在磁盘上
2. 导入：将导出的p12文件弄到需要导入的mac上，双击导入即可