在unity4.x通过xupoter插件来设置unity导出xcode自动配置，到了unity5.x unity自己实现了一套工具，并且比xupoter提供了更完善的操作可能。但是unity的工具集成在了unity编辑器里面，所以如果要想使用只能在unity工程中通过编辑器功能来操作导出xcode工程的自动配置。这样有一个缺点就是对于工程不需要变化但是又要打多个渠道的iOS包的情况下需要每次都要重新从Unity导出到xcode，大项目这是十分浪费时间的。

针对上述情况，Usdk独立出了unity这套自动配置工具，完全脱离了Unity体系，能够单独作为一个外部工具随时随地的操作xcode工程下的工程配置文件和Info.plist配置文件，可以添加或者移除指定库、源码、资源等，操作非常灵活，只要你能想到的操作都能够实现。

并且工具通过unity安装目录下的mono虚拟机进行运行，所以具有跨平台的能力，不管是windows还是mac都能够使用。

所以unity代码或者资源没有改变并且需要打多个渠道或者不同参数的渠道包的时候，只需要导出一次xcode工程即可，通过外部的自动配置工具进行xcode工程的各种修改，修改的时候时候最好要在原始xcode工程上直接修改，而是应该复制一个副本然后进行修改打包，这一切工作Usdk打包工具已经做了，只需要你在渠道下面配置一个XcodeSetting.json的文件即可，如果XcodeSetting.json的功能无法满足你的修改内容，那么请自行修改publish/ios/tools下XcodeProjectSetting源码，然后生成可运行的XcodeSetting.exe。

**XcodeSetting功能介绍**

支持功能：

1、Info.Plist编辑支持

2、系统Lib添加、移除支持

3、系统framework添加、移除支持

4、工程编译属性的添加、更新、移除

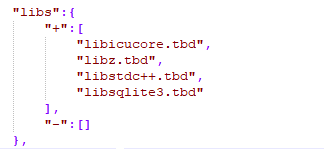
5、文件加入

6、文件夹加入

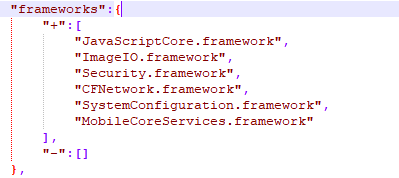
7、文件编译符号更新

**静态库配置：**

ios9以上，lib是以tbd为扩展名的，+号里表示要添加的库，-号是要移除的，一般不需要移除。

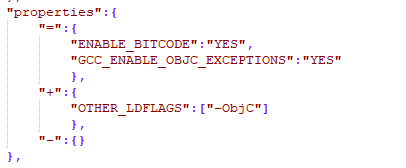


**Frameworks配置：**



**编译属性配置：**

需要注意的是，配置的KEY可能和在Xcode中看到的KEY不一样，每一条属性有一个声明名字和一个描述名字，一般配置下，在Xcode中看到的是描述名字，我们需要的是它的声明名字，方法就是，在Xcode中选中该条属性，点击右边检视面板上的“？”图标，然后Declaration就是我们需要的名字，例如，Enable Bitcode属性它的声明为ENABLE\_BITCODE，所以我们在配置时key为ENABLE\_BITCODE。



A、Bitcode设置为关闭

B、ObjC异常设置为启用

C、其它链接符号中加入符号-ObjC

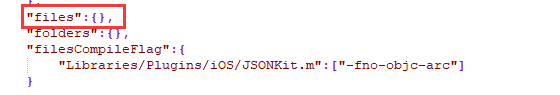
**Info.plist配置：**

分别配置了ShareSDK Schemes、相册权限描述、ShareSDK白名单，同上，它的KEY也是需要声明名字。

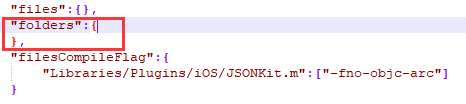


**文件复制配置：**

一般可用文件夹复制代替



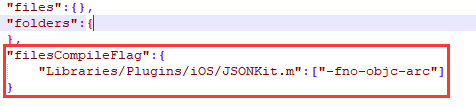
**文件夹复制配置：**



key是源目录的绝对路径。Value是目标目录在Xcode中的位置，是以Xcode工程目录为根目录的相对路径。

目前存在bug是xcode工程复制到其它mac上时，文件夹引用会丢失，需要手动添加一下引用。

**文件编译符号配置：**



这里是对JSONKit.m文件添加了-fno-objc-arc符号，key是目标目录在Xcode中的位置，是以Xcode工程目录为根目录的相对路径。

注意value是数组，也即可以写多个，也可以合成一个string写，但始终是数组形式。