提包文档--自动篇

## 关于文档

本片文档只会介绍关于自动化化工具相关内容，换皮等相关知识请查看提包文档--手动篇，进行了解。

## 关于AutoSubmitAppTools工具

AutoSubmitAppTools是为了简化提包流程，优化提包速度，减少不必要的重复性劳动和代替云主机Apploaduper而进行制作。主要功能有自动化创建bundleID、证书、描述文件及导出p12，自动化创建app及填写资料上传美宣，自动化提交ipa到app store商店

## 工具结构

AutoSubmitAppTools\_Ver1.5

* + - 使用文档.md： 简单的使用文档
    - autoSubmitAppTool.rb： 工具的入口，主程序
    - create\_app.rb： 创建app及填写资料模块
    - create\_cert\_appStore.rb： 创建及下载证书、描述文件、p12模块
    - exportAppStoreInfo.sh： 导出ipa和AppStoreInfo.plist
    - ExportOptions\_copy.plist： ExportOptions.plist的模版，导出AppStoreInfo.plist文件时使用
    - info.rb： 解析info.plist内容的模块
    - info模版.plist： info.plist的模版，需要复制一份重命名为info.plist后在里面填写提包所需要的资料
    - installEnv.sh： 自动检测mac/Linux系统安装工具所需环境
    - IpaUpload.sh： 自动检测mac/Linux系统使用iTMSTransporter工具进行提交ipa的模块
    - mobileprovision-read： 获取描述文件中的信息导出AppStoreInfo.plist文件时使用
    - plist2json.py： 把plist文件解析成json文件模块

## 使用方法

* + - 复制info模版.plist并重命名为info.plist，进行填写资料
      * info.plist文件内容：
      * 
      * 在info.plist模版中已经清晰的写了各个字段的涵义，下面说注意事项：
        + 在Primarylanguage字段填写的是主语言，常见语言如下：

美国英文： English

英国英文： English\_UK

加拿大英文： English\_CA

简体中文： Simplified Chinese

繁体中文： Traditional Chinese

* + - * + languages的值、appNames、keyword、promotional、description等的键所需要填写的都是语言的代码常见语言代码如下：

美国英文： en-US

英国英文： en-GB

加拿大英文： en-CA

简体中文： zh-Hans

繁体中文： zh-Hant

* + - * + rating是应用的评级。key是各个选项，value是级别，0是无、1是一般、2是频繁激烈，一般默认是4+，也就是都为0，其他根据具体情况进行填写
        + phoneNumber这里是电话号码。根据账号地区写一个格式正确的电话号码
        + xcodePath是Xcode项目的路径如：/Users/mac/pack/build/**jsb-default**
    - 工具目录下打开终端输入：**sh installEnv.sh** 来检测所需的环境，会自动进行安装。
    - 环境安装好后打开终端输入ruby autoSubmitAppTool.rb 然后回车，界面如下：
      * 
      * 1、会创建证书描述文件bundleID导出p12文件会自动下载至path路径下
      * 2、会把info.plist文件的资料上传至app store，期间如果出现问题可以进行修改info.plist文件然后重新输入指令不必重新打开程序
      * 3、提交ipa到商店，会提示输入ipa到路径，拖进入即可，Linux如路径错误可以手动输入
      * 4、确保项目已经构建完毕，且是最后一次构建的项目。会使用最后一次构建的项目导出ipa和AppStoreInfo.plist文件，然后会自动打开文件夹
      * 0、退出

## 工具原理

* + - 工具主要使用ruby语言进行编写，shell、python进行辅助
    - 证书及填写资料部分使用fastlane spaceship工具完成
    - 上传ipa部分使用iTMSTransporter工具完成
    - 解析plist使用python的plistlib完成
    - 使用shell进行检测并安装所需环境