# 

# AutopackingAndroid

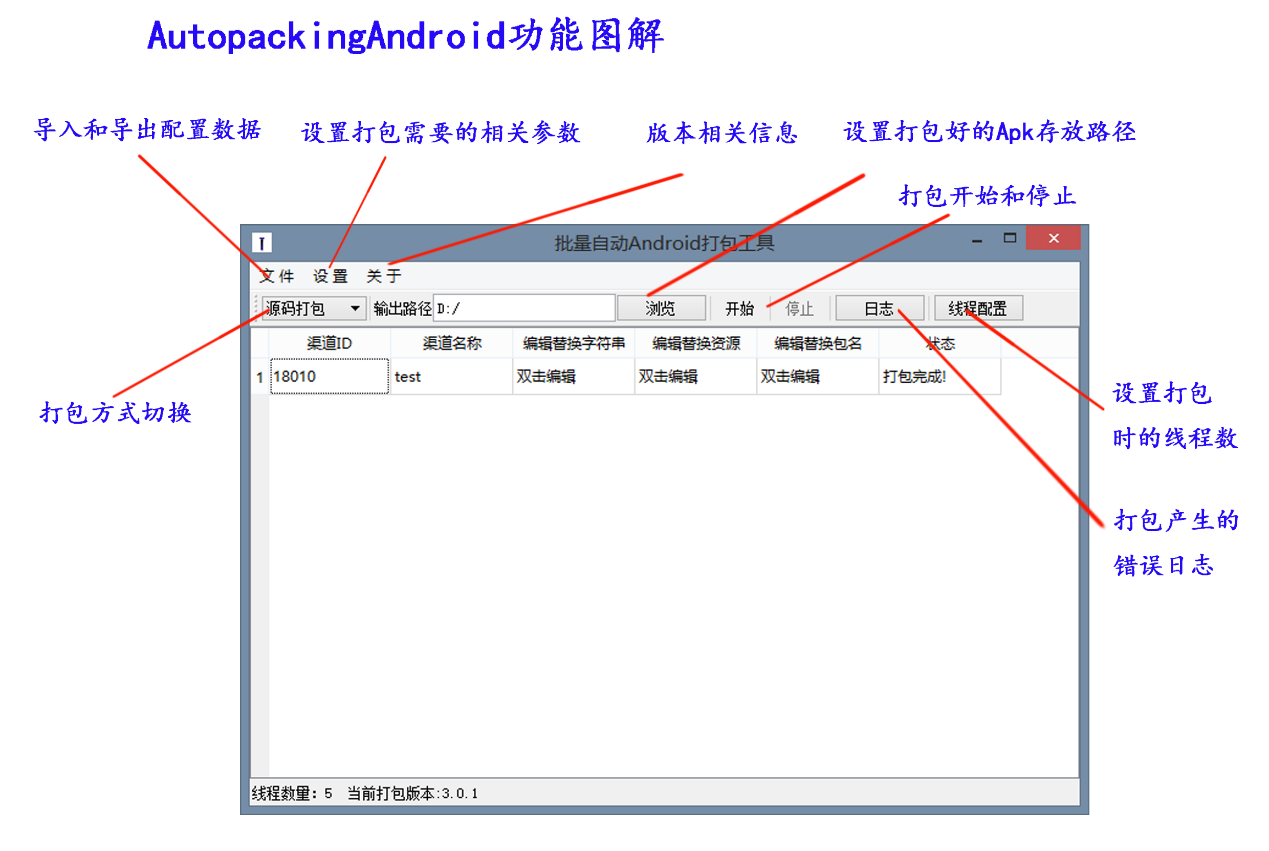
# 使用手册

# 一、概述

## 1. 关于AutopackingAndroid

AutopackingAndroid是一款专注于android的批量打包工具，主要定位于一次配置，重复、快速的批量生成Android 的Apk安装包。

2.功能介绍

****

**特别说明：**

**线程配置是配置同时开多少线程去打包，如果要打包的任务数少于线程数，则最多只能开启到任务数量的线程去打包。**

**日志是指错误日志，只显示打包过程中出现的错误，如果错误日志为空，就是打包过程中未产生错误，是正确的打包。**

**注意：打包开始时一定注意看现在的打包方式是源码打包还是反编译打包。避免造成打包错误的问题。**

## 3. 安装要求

安装前请保证机器已经安装好并配置好Android开发环境，其中包括：java环境运行时环境的配置、JAVA\_HOME环境变量的配置。JAVA\_HOME的环境变量一定要配置，否则在源码打包时会报错误。

在以下环境中测试通过

测试环境：

平台：Windows8.1 64位

Adt版本：adt-bundle-windows-x86\_64-20140702

Java版本：jdk1.8.0\_11

# 二、开始使用

AutopackingAndroid支持两种打包方式，一种是通过源码的打包，与Eclipse生成Apk的方式相同。在这种模式下，支持包名替换、渠道号替换、资源文件替换、支持混淆。是最安全的批量打包方式。另一种是通过反编译打包，反编译打包是把原来的包反编译之后进行一些渠道号的替换、资源的替换之后，重新生成Apk，签名，zip4字节优化后生成最终Apk。这种模式下，支持渠道号替换、资源文件替换。但不支持包名替换。所以请选择合适的打包方式。下面将分别介绍用法。

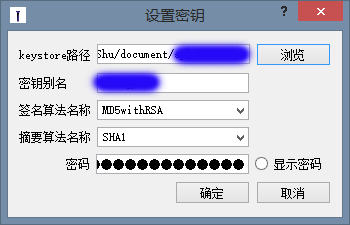
无论采用哪种打包，开始之前都必须设置相关信息，如果配置信息不全则不能开始打包。

两种方式都必须设置的信息

Jdk路径

无论采用源码打包还是反编译打包，都必须设置jdk路径。正常情况下，软件会自动识别你的java环境变量并配置jdk路径，如果软件识别的不准确或未识别到的，请手动修改到正确位置。

密钥信息



如图，

keystore路径就是签名密钥的文件路径。

密钥别名就是申请密钥时候设置的别名，如果没有可留空。

签名算法有两种主流的可供选择，MD5withRSA和SHA1withDSA，如果选项中没有您使用的，请手动填入您的签名算法名称。（支持手动编辑）。

摘要算法名称目前只内置了一个，如果不是您采用的，请手动填入您的签名算法。

密码就是您签名包所用的密码。

选择性配置的信息

版本号是选择性配置信息，可配置可不配置。配置的目的是为了生成包名的时候区分版本的。

最后生成包名的命名规则

最后的包名生成规则是渠道名称+渠道ID +　版本号。



如图所示配置，最后生成的包名为：test\_18010\_3.0.1.apk

1. 源码打包

首先在进行源码打包之前，请先保证你的项目能在Eclipse中无错误可打包。如果在Eclipse无法进行打包的，在AutopackingAndroid中同样无法打包，会报错误。

# 五、注意事项

# 六、问题反馈