## 需求

针对咪咕音乐客户端APK做拆包，拆出所有的文件清单，跟上一个版本比较change，然后跟GIT 和JIRA打通，确认安装包改动的变化，包括大小、修改日期，修改人员，关联的故事是谁

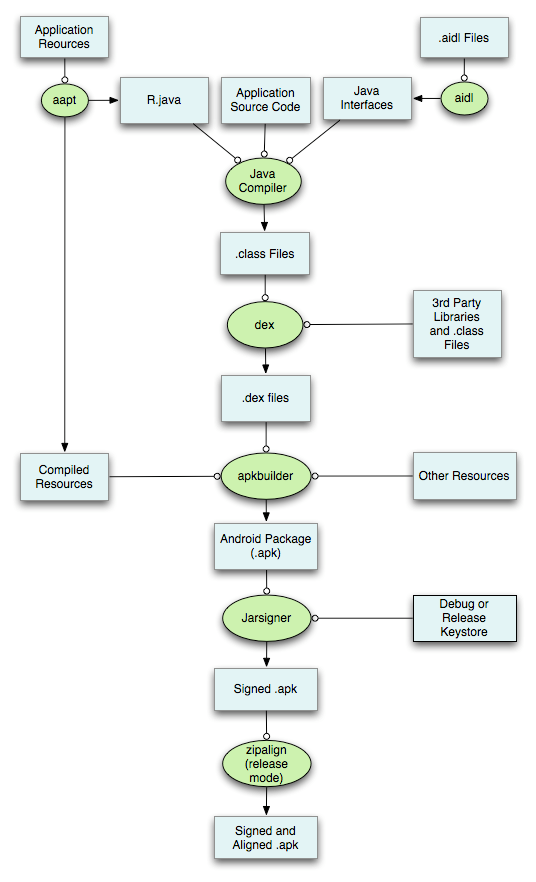
## 相关技术：

1. [Apk编译打包流程](https://mp.weixin.qq.com/s?__biz=MzI1NjEwMTM4OA==&mid=2651232113&idx=1&sn=02f413999ab0865e23d272e69b9e6196&scene=1&srcid=0831gT4p6M0NFG5HTTeRHTUC#wechat_redirect)
2. [反编译](https://blog.csdn.net/gaozhan_csdn/article/details/51984056)

## 编译打包概要：

1. 资源合并（有support包或者是aar库时，先解压后与本地资源合并，含有assets、res、AndroidMainfest.xml）
2. **第三方依赖中的assets或res文件与本地文件有冲突时，会优先选用本地文件**；res/values略有不同，此目录下的strings.xml、color.xml、styles.xml等文件会被整合到一个叫values.xml的文件中去，后与各第三方依赖中的values.xml进行内容上的合并，不会像res其它子目录文件一样直接舍弃第三方冲突文件
3. Androidmanifest.xml的合并
4. AAPT打包（生成R.java以及resources.arsc，assets不做任何处理，res/raw分配id后与assets一起打包进应用中，其他xml文件则被编译为二进制）
5. 源码编译（含有aidl或者renderscript时首先调用llvm-rs-cc，生成java文件和相应的.bc文件;.bc文件直接打包进apk，java代码和源代码一起javac编译为字节码）🡪混淆处理
6. 生成DEX（主要是项目中四大组件、Application、Instrumentation、注解类等的java代码编译生成的）
7. 打包签名生成apk
8. zipalign文件对齐[非必要，可提高系统访问安装包资源的效率]

## 编译打包图示：



**简单来说就是：反编译apk后上传git与历史版本对比得到改动文件列表changeList，再参照历史提交列表commitList获取当前关注的文件改动详情**

**目前开发流程：**1.jira故事🡪2.创建分支🡪修改，开发，提交🡪3.删除分支

**优化流程如下：**1.Jira故事🡪2.创建分支（此时jira已经与git产生了关系，关联了起来，记录下创建者和开发者）🡪3.修改，开发，提交（修改日期，人员已关联）\*N次，（由Git可以得到N个历史改动详情[类、资源等]，生成列表commitList）🡪4.编译打包apk🡪5.解压apk上传git与历史版本对比（获取apk已改动文件[资源等]清单changeList）[如果需要对比java类文件的改动，需要反编译dex文件]🡪6.根据changeList查看我们需要关注的文件及其改动，此外，筛选出来此文件在commitList中的对应的改动历史，进而就获得了我们需要的信息（如res目录中某图片为新增[打包后的图片命名未改变]，我们只需要根据此文件名找到commitList中此图片对应的改动历史）{说明：如果有文件为自动生成，参照gradle打包步流程骤，生成文件与生成此文件的目录进行反追溯（如资源索引表arsc文件，它是根据合并资源后的xml文件编译生成的，合并的资源文件来自于本地文件、依赖库或者是依赖文件，那么我们就可以从这些文件或者库进行反追溯）}

ChangeList和CommitList得到之后，可以根据changeList中改变的文件找到对应的CommitList中对应的文件或资源并取出来。Bingo！

JIRA故事

创建git分支，

并记录开发者，创建者

Git分支

历史版本apk

开发，修改，提交，

解压apk，得到文件目录 统计修改详情（人员，日期，文件等）

Apk文件

commitList（提交改变的文件与目录）

编译打包

历史文件目录

得到提交历史详情，

开发者，日期，修改

解压apk，得到文件目录， 的文件

再上传Git并对比，

得到changeList

ChangeList（改变的文件与目录）

需要关注的

文件或目录

查询commitList得到关注的改动历史

ResultChangeList