

友盟多渠道打包

1.配置清单文件

2.在mode的buildGradle文件里配置

3.在Java代码中设置获取渠道标识，发送个服务器的逻辑代码（简单的吐了司，表示拿到渠道标识）

4.进行签名打包

一、渠道包就是要安装包中添加渠道信息，也就是channel,对应不同的渠道

列如：360，应用宝，机锋，小米，豌豆芽市场

多渠道打包的作用：apk---发布到市场上，当我们在这些APK安装包中已经添加了不同的渠道标识应用在请求网络的时候会携带渠道信息（渠道就是可以下载apk的地方，），方便后台做运营统计（这就是添加渠道信息的左右）

总结：如果不多渠道打包的话，一个APK包投放到数十个市场，你就无法自动哪一个市场下载量最大---多渠道打包主要是为了运营，让他们根据数据去玩。

多渠道打包的核心：就是给每一个APK根据要投放的市场，打上独有的渠道标识（一般来讲，这个渠道标识会放在清单文件里，然后通过java代码获取，发送到我们服务器上）

二、多渠道打包基本步骤：

1、为APK添加标记（渠道的标识）

2、运行APK的时候取出标记

3、上传标记给服务器统计

三、程序打包：就是根据签名和其他标识生成安装包（注意：app的apk名字不能包含中文）

四、APP签名：在Android应用文件中保存的一个特别字符串（用来表示不用的应用开发者：开发者A.B）

签名的作用：

1， 安卓系统要求每一个android应用程序必须要经过数字签名，才可以安装到系统中（我们平常没有配置，依然能运行到程序上，是编译工具使用了 默认签名， debug,keystore）（也就是说程序没有签名是安装不到Android系统上）

五、提示：

- 1.Debug签名的应用程序不能在安卓市场上发布，它会强制你使用自己的签名
- 2.不用电脑使用文件生成的签名文件不一样，那就意味着你换机器，也就换了签名文件
- 2.主要是为了后期APP的升级，能无缝升级，故一定要保存好你的签名文件，丢了就找不回来（Android系统靠签名文件识别程序的拥有者）。
- 3.用来标识不同的应用开发者：开发者A,开发者B，防止盗版，应用更新时要用同一个签名，否则就不是更新而是安装新应用了

提示：千万不要把签名文件弄丢了（jks是studio签名产生的文件）

签名文件丢失的后果：

- 1.用户安装时必须先卸载才能成功安装，
- 2.应用市场有下载量排名，如果重新签名，应用市场上的排行就会从0开始

建议：所有应用程序，使用一个签名证书：方便升级，模块化，代码或数据的共享

六、友盟多渠道打包优点：安全，效率较高，效率取决于设备性能，市面上百分之八十公司采用友盟多渠道打包

七、360多渠道打包优点：1秒能打300个包，可以在下载APK的同事，服务端可以写入一些信息，列如邀请码，分享信息等等

缺点：渠道信息较容易修改，虽然可以加密，但只是提供了修改的门槛，故有百分之十的公司会采用此方法，了解即可。

360多渠道打包原理：apk文件本质就是zip文件，利用zip文件可以添加comment的数据结构特点，在文件的末尾写入任意数据，而不是从重新解压zip文件,我们就可以将渠道信息写入摘要区（复制）

//AndroidManifest.XML

<meta-data

android:name="UMENG_CHANNEL"

android:value="{\$UMENG_CHANNEL_VALUE}"

/>

//build.gradle下

//设置APK的渠道信息

```
productFlavors {
    wandoujia {}
    xiaomi {}
}
//分别进行打包，固定格式
productFlavors.all {
    flavor ->
        flavor.manifestPlaceholders = [UMENG_CHANNEL_VALUE: name]
}
//根据渠道自定义apk的名称,自带版本号,渠道名
applicationVariants.all { variant ->
    variant.outputs.each { output ->
        output.outputFile = new File(
            output.outputFile.parent,
            "xxxx(apk的名字)-${variant.buildType.name}-
${defaultConfig.versionName}-
${variant.productFlavors[0].name}.apk".toLowerCase())
    }
}
```