

# [Android] V1 多渠道打包

做法

Android 專案

參考資訊

## 做法

1. 生成APK后，通过解压apk，在META-INF里面添加空的文件（文件名标识渠道）来完成；
2. 通过apk末尾的一个 File comment 区域来添加信息

### 大写的注意:

第二，三种方式在7.0一下是没有问题的。7.0以后，这样做就无法安装了

android 7.0 引入一项新的应用签名方案 APK Signature Scheme v2，它能提供更快的应用安装时间和更多针对未授权 APK 文件更改的保护。在默认情况下，Android Studio 2.2 和 Android Plugin for Gradle 2.2 会使用 APK Signature Scheme v2 和传统签名方案来签署您的应用。——官网

我们需要在gradle里面禁用新版签名模式

```
1  android {  
2      ...  
3      defaultConfig { ... }  
4      signingConfigs {  
5          release {  
6              storeFile file("myreleasekey.keystore")  
7              storePassword "password"  
8              keyAlias "MyReleaseKey"  
9              keyPassword "password"  
10             v2SigningEnabled false  
11         }  
12     }  
13 }
```

位现象，建议100%大小显示。

## Android 專案

- 請使用 kotlin 語言，Java 語法可以使用 Android Studio 貼上做轉換即可。

- 讀取 channel 的方式

```
fun getChannel(context: Context): String? {
    val appinfo: ApplicationInfo = context.getApplicationInfo()
    val sourceDir = appinfo.sourceDir
    var ret = ""
    var zipfile: ZipFile? = null
    try {
        zipfile = ZipFile(sourceDir)
        val entries: Enumeration<*> = zipfile.entries()
        while (entries.hasMoreElements()) {
            val entry: ZipEntry = entries.nextElement() as ZipEntry
            val entryName: String = entry.getName()
            // 檔名與python指令碼定義的相匹配即可
            if (entryName.startsWith("META-INF/tdchannel")) {
                ret = entryName
                break
            }
        }
    } catch (e: Exception) {
        e.printStackTrace()
    } finally {
        if (zipfile != null) {
            try {
                zipfile.close()
            } catch (e: IOException) {
                e.printStackTrace()
            }
        }
    }
    val split = ret.split("_").toTypedArray()
    return if (split.size >= 2) {
        ret.substring(split[0].length + 1)
    } else {
        ""
    }
}
```

- 讀取 apk comment 的方式

```
// 做讀取的動作
fun readAPK (filename:String) : String? {
    //获取缓存的 APK 文件
    val file = File(this.getPackageCodePath())
    var bytes: ByteArray
    var accessFile: RandomAccessFile? = null
    // 从指定的位置找到 WriteAPK.java 写入的信息
    try {
        accessFile = RandomAccessFile(file, "r")
```

```

        var index = accessFile!!.length()
        bytes = ByteArray(2)
        index = index - bytes.size
        accessFile.seek(index)
        accessFile.readFully(bytes)
        val contentLength: Short = this.stream2Short(bytes, 0)
        bytes = ByteArray(contentLength.toInt())
        index = index - bytes.size
        accessFile.seek(index)
        accessFile.readFully(bytes)
        return String(bytes, charset("utf-8"))
    } catch (e: IOException) {
        e.printStackTrace()
    } finally {
        if (accessFile != null) {
            try {
                accessFile.close()
            } catch (ignored: IOException) {
                ignored.printStackTrace()
            }
        }
    }
    return null
}

fun stream2Short(stream: ByteArray, offset: Int): Short {
    val buffer: ByteBuffer = ByteBuffer.allocate(2)
    buffer.order(ByteOrder.LITTLE_ENDIAN)
    buffer.put(stream[offset])
    buffer.put(stream[offset + 1])
    return buffer.getShort(0)
}

```

## 參考資訊

- [https://blog.csdn.net/android\\_jianbo/article/details/53643826](https://blog.csdn.net/android_jianbo/article/details/53643826)
- 完整的做法 / 程式碼參考
  - <https://www.twblogs.net/a/5b8bfcfb2b717718832f9d41>
- 寫入 comment 的做法
  - [https://blog.csdn.net/yancey\\_blog/article/details/52743675](https://blog.csdn.net/yancey_blog/article/details/52743675)