**第一步：得到源程序Apk文件和脱壳程序的Dex文件**

运行源程序和脱壳程序项目，之后得到这两个文件(记得将classes.dex文件改名ForceApkObj.dex)，然后使用加壳程序进行加壳：

这里的ForceApkObj.apk文件和ForceApkObj.dex文件是输入文件，输出的是classes.dex文件。

**第二步：替换脱壳程序中的classes.dex文件**

我们在第一步中得到加壳之后的classes.dex文件之后，并且我们在第一步运行脱壳项目的时候得到一个ReforceApk.apk（需要在eclipse中签名导出）文件，这时候我们使用解压缩软件进行替换：

**第三步：我们在第二步的时候得到替换之后的ReforceApk.apk文件，这个文件因为被修改了，所以我们需要从新对他签名，不然运行也是报错的。**

工具下载：<http://download.csdn.net/detail/jiangwei0910410003/9102767>

下载之后的工具需要用ReforeceApk.apk文件替换ReforceApk\_des.apk文件，然后运行run.bat就可以得到签名之后的文件了。下载的文件中有了force文件，这个是签名文件。

run.bat文件的命令如下：

**cd C:\Users\i\Desktop\forceapks  
jarsigner -verbose -keystore forceapk -storepass 123456 -keypass 123456 -sigfile CERT -digestalg SHA1 -sigalg MD5withRSA -signedjar ReforceApk\_des.apk ReforceApk.apk jiangwei  
del ReforceApk.apk**

这里最主要的命令就是中间的一条签名的命令,关于命令的参数说明如下：

**jarsigner -verbose -keystore 签名文件 -storepass 密码  -keypass alias的密码 -sigfile CERT -digestalg SHA1 -sigalg MD5withRSA  签名后的文件 签名前的apk alias名称  
  
eg:  
jarsigner -verbose -keystore forceapk -storepass 123456 -keypass 123456 -sigfile CERT -digestalg SHA1 -sigalg MD5withRSA -signedjar ReforceApk\_des.apk ReforceApk\_src.apk jiangwei  
  
签名文件的密码：123456  
alais的密码：123456**

所以这里我们在得到ReforceApk.apk文件的时候，需要签名，关于Eclipse中如何签名一个Apk的话，这里就不多说了，自己google一下吧：

那么通过上面的三个步骤之后我们得到一个签名之后的最终文件：**ReforceApk\_des.apk**

# APK逆向

### [apk反编译和重新打包](http://ye-good-job.iteye.com/blog/1849727)

**博客分类：**

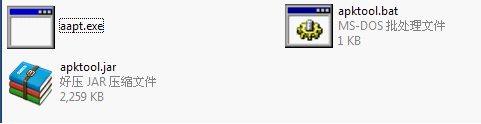
* [android](http://ye-good-job.iteye.com/category/241759)

相信每位玩机的人对APK文件都不陌生。你可能每天都与APK文件打交道，无论是安装和卸载有用的应用工具、插件、好玩的游戏等等。。。你可曾知道这些每天都伴随着你的APK文件是什么吗？怎样对它们作些修改呢？比如说：对英文版进行汉化、修改功能、修改文字描述、去掉广告等等。本文介绍APK的基本知识、结构、APK文件的解包、打包及签名，以及对APK文件的常规修改。

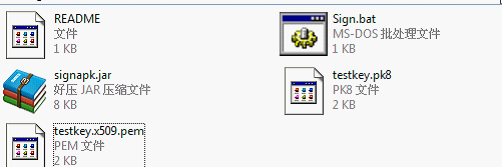
**1.APK文件简介**  
  
APK是Android Package的缩写，即即Android application package文件或Android安装包。每个要安装到Android平台的应用都要被编译打包为一个单独的文件，后缀名为.apk。APK文件是用专业软件eclipse编译生成的文件包，其中包含了应用的二进制代码、资源、配置文件等。通过将APK文件直接传到Android手机中执行即可安装。APK文件其实就是zip格式，但其扩展名被改为apk，用解压软件可以直接打开。通过WinRAR或UnZip解压后，你会看到有几个文件和文件夹。一个典型的APK文件通常有下列内容组成：  
AndroidManifest.xml 程序全局配置文件  
classes.dex    Dalvik字节码  
resources.arsc    编译后的二进制资源文件  
META-INF\ 该目录下存放的是签名信息  
res\  该目录存放资源文件  
assets\ 该目录可以存放一些配置文件  
  
下面对这些文件和目录做些基本的注释和介绍。  
• AndroidManifest.xml  
该文件是每个应用程序都必须定义和包含的文件，它描述了应用程序的名字、版本、权限、引用的库文件等等信息。需要解包后才能加以阅读。  
• classes.dex文件   
classes.dex是java源码编译后生成的java字节码文件。dex是Dalvik VM executes的全称，即Android Dalvik执行程序，并非Java ME的字节码而是Dalvik字节码。  
• resources.arsc   
编译后的二进制资源文件。  
• META-INF目录   
META-INF目录下存放的是签名信息，用来保证apk包的完整性和系统的安全。在eclipse编译生成一个apk包时，会对所有要打包的文件做一个校验计算，并把计算结果放在META-INF目录下。这就保证了apk包里的文件不能被随意替换。比如拿到一个apk包后，如果想要替换里面的一幅图片，一段代码， 或一段版权信息，想直接解压缩、替换再重新打包，基本是不可能的。如此一来就给病毒感染和恶意修改增加了难度，有助于保护系统的安全。  
• res目录   
res目录存放资源文件。包括图片，字符串等等。  
解包后，几乎所有可能的修改和编辑工作基本都在这里。  
• assets目录  
assets目录可以存放一些配置文件，这些文件的内容在程序运行过程中可以通过相关的API获得。   
  
  
**2.APK文件的反编译和打包**

首先要下载两个工具：dex2jar和JD-GUI，前者是将apk中的classes.dex转化成Jar文件，而JD-GUI是一个反编译工具，可以直接查看Jar包的源代码。以下是下载地址：dex2jar：http://laichao.googlecode.com/files/dex2jar-0.0.7-SNAPSHOT.zip JD-GUI：http://laichao.googlecode.com/files/jdgui.zip

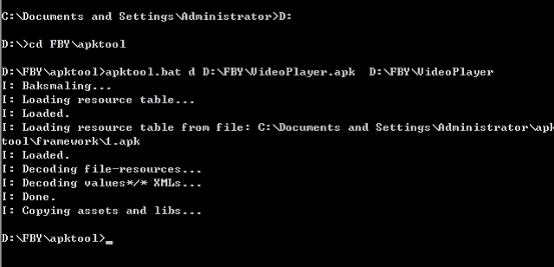
工具：apktool



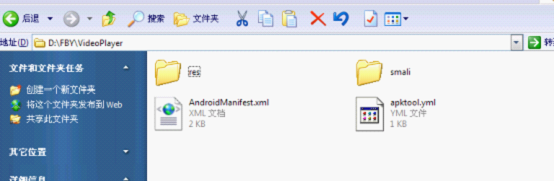
签名工具：Auto-sign.rar



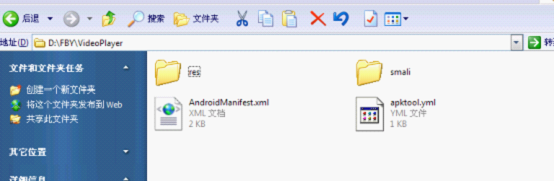
1. 进入到apktool 把要修改的apk通过如下



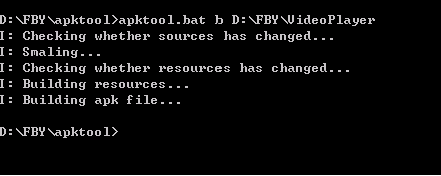
把apk反编译出来结果如下



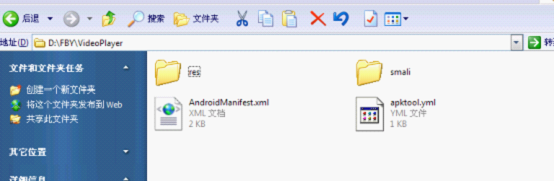
1. 修改要修改的地方 修改后结果还是



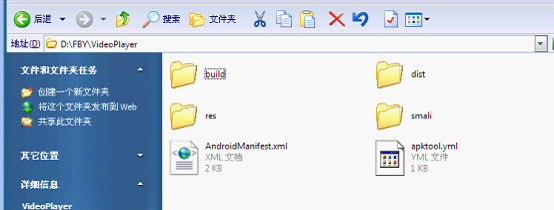
再通过如下的操作进行修改后的apk进行打包为apk



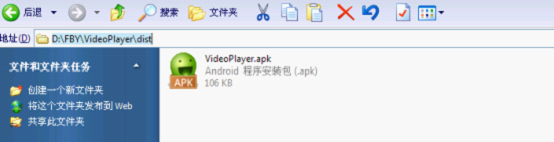
进行上面操作后由原来的



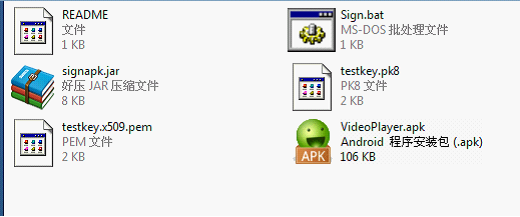
变为



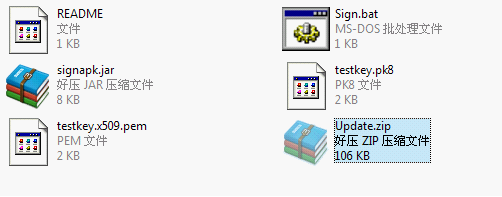
打包好的apk在D:\FBY\VideoPlayer\dist的目录下



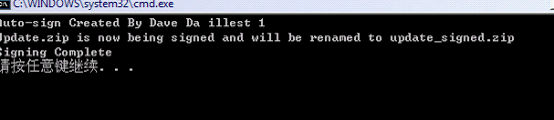
1. 解压签名工具：Auto-sign.rar把打包好的apk复制到 Auto-sign目录下如



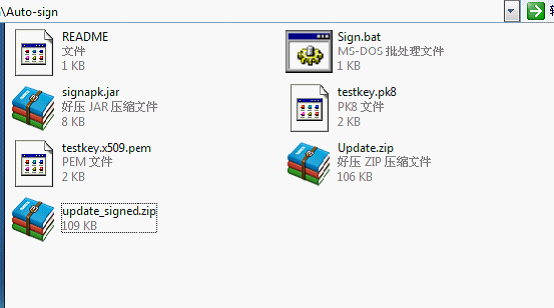
1. 修改 VideoPlayer.apk为 Update.zip



1. 执行Sign.bat 进行签名如下



1. 结果生成了一个update\_signed.zip的文件



把update\_signed.zip修改为XXXXX.apk

完成了！