相信很多开发者都知道反编译APK，正常情况下，如果我们想要看看xml，看看java源代码，是看不了的。用压缩工具解压apk得到的文件也都是二进制文件，打开后全是乱码。

    开发者能够用的也仅仅是图片资源，这往往也满足不了开发者的需求。

    下面我就给大家详细讲解一下Android-apk的反编译步骤：

    注意：反编译的前提是：开发者没有使用proguard（java代码混淆的工具）。

**1.下载反编译工具**

    需要下载四个工具：dex2jar、JD-GUI、apktool、apktool-install-windows-r05-ibot

    dex2jar：将apk中的classes.dex转化成Jar文件。

    JD-GUI：反编译工具，可以直接查看Jar包的源代码。

    apktool：反编译工具，通过反编译APK中XML文件，直接可以查看。

apktool-install-windows-r05-ibot：辅助windows批处理工具。

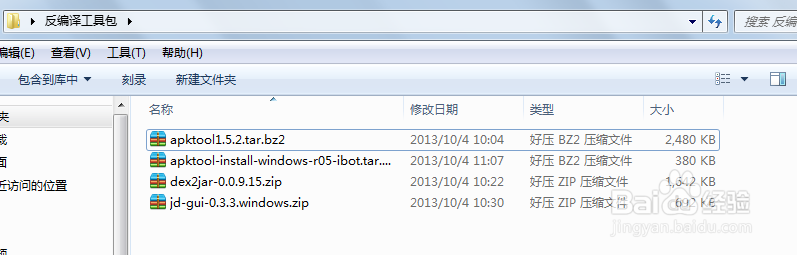
下载地址：（这里工具JD-GUI和apktool.jar,apktool.bat，aapt.exe已在库中）

    dex2jar：http://code.google.com/p/dex2jar/downloads/list

    JD-GUI：http://code.google.com/p/innlab/downloads/lis

    apktool：http://code.google.com/p/android-apktool/downloads/list

    工具截图：



**2.解压缩APK包**

    apk文件也是一种常见的zip，常用的压缩软件就可轻松地解压apk文件。用压缩软件将apk文件中的文件解压到一个文件夹中，得到APK中的资源包括XML文件和classes.dex文件，classes.dex就是java文件编译再通过dx工具打包而成的。

    解压截图：



**3.反编译dex文件**

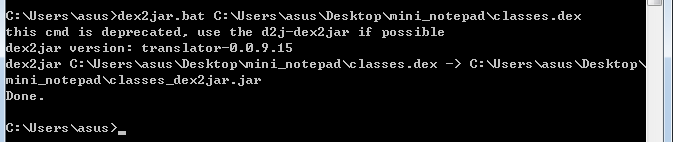
    解压apk文件后，你会发现res目录的图片没有加密，但java源码编译成了一个classes.dex文件，无法用普通的反编译class文件的方法来处理，dex2jar可以将dex文件转换成普通的jar包。

    解压下载的dex2jar，把解压后的文件夹放到系统盘跟目录中并把这个目录加到系统path变量中，就可以直接用:dex2jar.bat或dex2jar.sh转换dex文件了。

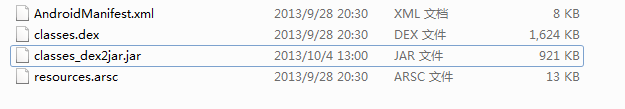
    DOS行命令如下：dex2jar.bat  xx\classes.dex（xx是classes.dex文件所在的路径）。

    将在同目录得到普通jar包：classes\_dex2jar.jar，就可以用处理普通jar包的方法来反编译这个jar包得到原文件。

    编译截图：



   生成jar文件的截图如下：

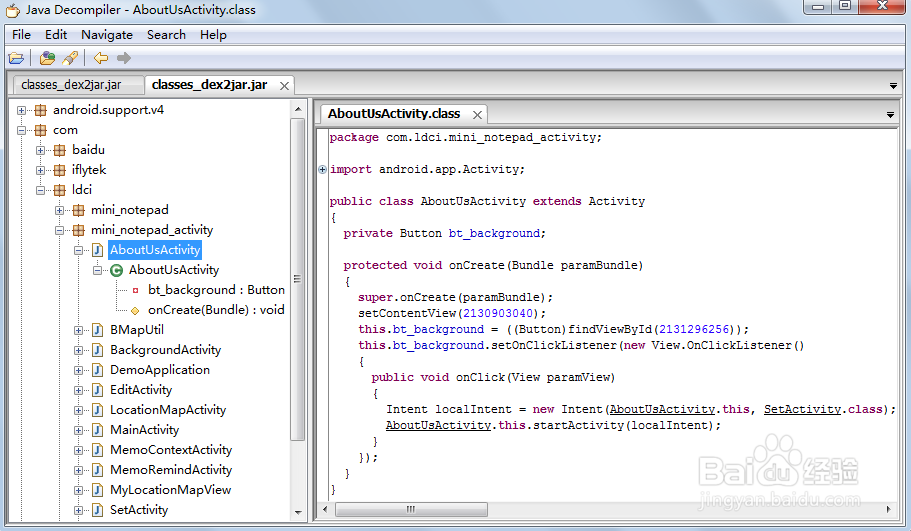


**4.反编译jar包（所有的签名jar包都可以这样反编译）**

    得到jar包后，下载自已喜欢的反编译工具，推荐使用JD-GUI，有比较简单的图形界面，可以反编译单个class文件，也可以反编译jar包，比较方便。

    打开压缩文件JD-GUI，运行jd-gui.exe，打开上面生成的jar包，即可看到源代码了。

    截图如下：



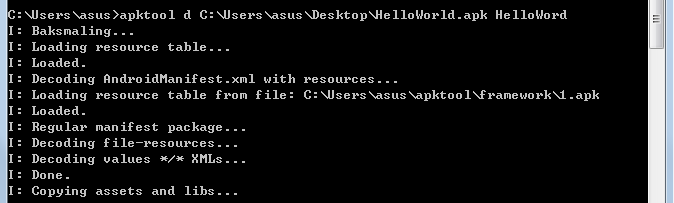
**5.反编译xml文件 （apk在getpublickey（）之前会分析**AndroidManifest.xml**）**

    打开解压的res\layout目录，有很多xml文件，如果你想看下作者是如何设计界面的，你会很失望，因为你看到的是一大堆乱码！这个时候我们需要用到的是apktool。

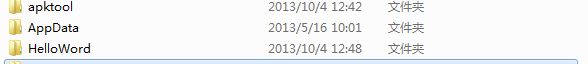
    具体操作方法：将下载的apktool和apktool-install-windows-r05-ibot两个包解压到同一个文件夹下，这个文件夹有三个文件：aapt.exe，apktool.bat，apktool.jar。把这个文件夹也放到系统盘的根目录中并把这个目录加到path变量中。以HelloWord.apk为例：

    在DOS命令行中输入：apktool d C:\....xx\HelloWord.apk HelloWord(我在输入目录是报错，所以干脆最后的参数就没输入，最后的包存放在C:\Users\ronny下，文件夹名字与apk的名字一致)得到HelloWord文件夹，此文件夹中的xml文件就是编译好的可以正常查看的文件。

    截图如下：



   生成的文件：



 这样就得到了可以得到编译的源码和XML资源。

 注意：文件的路径中最好不要出现中文！