一、WestAPP 工程说明

该工程为贵州威思特电子有限公司汽车诊断设备Android平板软件的工程，主要包括以下几个模块：

app: 应用程序主模块

appLibrary: 应用使用到的第三方库，如Android 应用权限请求RxPermission库、PDF展示库

serialPort：串口库，将蓝牙和USB模拟成串口

videoCache：视频缓存

bluetoothchat：此模块为独立的应用程序模块，用于模拟蓝牙

二、程序编译前准备：

该工程为Android Studio 工程，在编译运行前，需要在机器中安装JDK、Android SDK 以及Android Studio。

JDK可访问 <http://www.oracle.com/technetwork/java/javase/downloads/index.html> 选择相应版本下载并安装，需要注意的是JDK版本需要与电脑相对应（64位或32 Windows）；另外在安装JDK完成之后需要配置环境变量，配置过程可在百度中搜索。

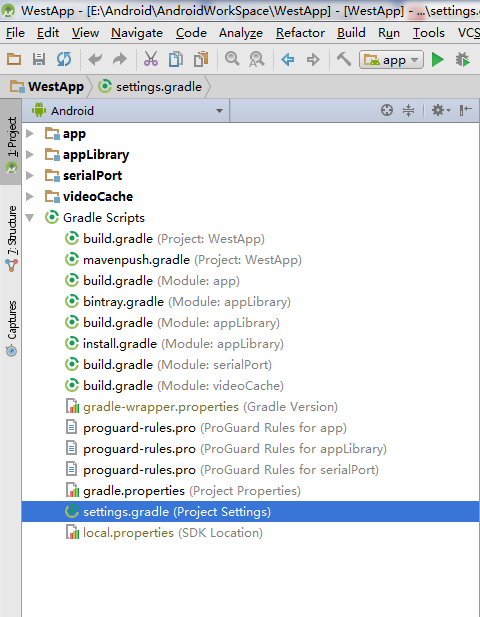
Android SDK 与可以在Android Studio安装过程中一起打包安装，另外，安装后还需要下载与运行目标Android设备的版本相一致的SDK库（现测试Android平板版本为6.0，对应的SDK库版本为23）。

Android Studio 可在<http://www.android-studio.org/>中下载并安装。（该工程开发中使用的版本为2.2.2）

三、工程引入

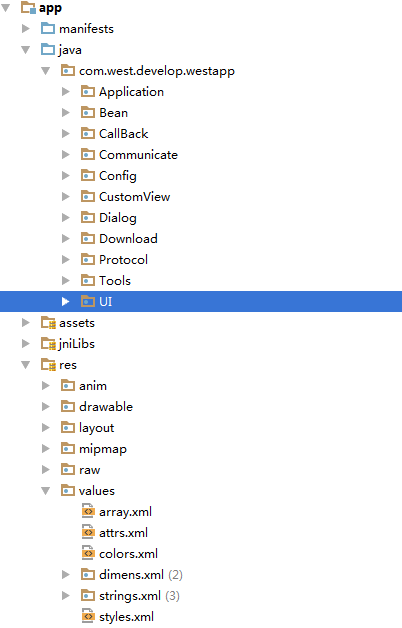
当JDK安装并配置环境变量完成，且Android Studio 和Android SDK均安装完毕后，打开Android Studio引入工程，就可以成功编译了。引入工程方式如下：

1. File→Open→选择该工程保存的跟路径（如：E:\Android\AndroidWorkSpace\WestApp），点击“OK”，需要注意的是，在第一次编译该工程的时候，部分库文件需要从网络下载，因此第一次编译时请连接网络，否则会编译报错。
2. 如果编译成功，将“Project” 栏调节到“Android”模式，显示如下图：



四．应用主模块介绍

模块结构如图：



1. manifest 应用声明

包括：应用权限声明、页面声明、公用广播接收器注册、服务声明等

1. java 控制源代码

Application：关于应用

Bean：自定义类

Callback：回调接口

Communicate: 通讯连接，USB连接、蓝牙的搜索和连接

Config: 应用配置

CustomView: 自定义控件

Dialog: 对话框

Download: 文件下载相关（诊断程序升级、文档下载、APP更新下载、固件下载）

Protocol: 诊断协议

Tools: 工具

UI: 界面相关

1). Communicate:

BluetoothService: 打开平板蓝牙后自动搜索附近蓝牙设备，如果蓝牙名称属于蓝牙设备列表，并蓝牙设备包含SPP协议的UUID，则连接蓝牙设备。

USBService:检查到USB串口设备连接后，请求应用的USB权限，当权限请求成功，则连接USB设备。

蓝牙设备连接成功后，检查设备序列号，如果返回正确，则说明设备是与平板电脑绑定的接头设备；如果返回错误，则说明是接头设备，但并非与平板电脑绑定的接头设备；如果没有返回，说明不是本公司生产的接头设备。

COMFunAPI: USB、蓝牙串口的通讯基础封装，其功能包括：获取读取缓冲区大小、打开关闭串口、发送接收单字节、延时、接收发送特定长度的字节数组。

2). Config: 应用配置参数

配置参数主要包括：已使用次数、可使用次数、未注册设备试用次数、语言、绑定设备信息、设备是否注册、设备最后使用时间、偏好设置内容、蓝牙设备名称列表。

3). CustomView:

QuickSideBarView: 诊断程序升级列表右侧字母导航

Radio: 自定义单选按钮

VideoView: 视屏播放控件

ListViewForScrollView: 兼容ScrollView的ListView

RefreshLayout: 可下拉刷新和上拉加载的布局

ScreenView: 程序诊断过程中手持机屏幕控件（仿手持机）

4). Dialog：

BackupDialog: 诊断过程中需备份文件时弹出的对话框

BondDialog: 绑定设备时弹出的对话框

ConnectStatus: 应用下方的状态栏

FileDialog: 诊断过程中读取备份文件时选择文件弹出的对话框

LoadDialog: 有进度条的对话框

MenuDialog: 从下方弹出的按钮列表对话框

SignDialog: 激活设备时弹出的对话框

TipDialog: 提示对话框，应用需要文字提示时调用

5). Download: 诊断程序升级下载、APP应用下载、说明文档下载、固件程序下载、诊断记录上传

ProgramDownload: 诊断程序升级下载（支持断点续传）

Thread: 文件下载线程

APKDownloadThread：平板应用APK下载线程

CompareVersionThread: 获取服务器诊断程序版本后与本地进行比较线程

DocumentDownloadThread: 文档下载线程，用户手册，快速指南

FWDownloadThread: 固件程序下载线程，下载完成后升级手持机

PostReportTask: 诊断记录上传线程

6). Protocol: 诊断协议封装

Drivers:

BaseCMD: 指令常量

BaseDriver: 协议封装超类，发送包、计算校验和、检验校验和、计算实际返回值

DeviceDriver: 设备连接驱动，检查序列号、读取版本

RunningDriver: 诊断过程中通讯封装，接收手持机指令（屏幕显示、文件备份）、操作手持机（键盘）

UpDriver: 下载诊断程序封装，检验序列号、跳转Boot、诊断程序下载、固件程序下载

DISP\_IOPack: 屏幕显示接收器

HolderThread: 发送通讯保持指令（间隔2s）

TIOPack: 通讯基础数据封装

7). Tools: 工具

constants: 常量

RequestCodeConstant: 启动Activity 的RequestCode 常量，对应startActivityForResult()和onActivityResult();

URLConstant: 服务器数据接口地址

Diagnosis: 诊断程序

DiagnosisAPI: 诊断程序下载过程，判断固件程序可升级、程序下载、程序下载过程对话框显示

KeyEvent: 诊断过程中键盘事件

Utils: 工具类

FileUtil: 文件工具，诊断程序、帮助文档、诊断记录、备份数据、汽车品牌图标

HexUtil: 16进制工具，字节数组与16进制数的相互转换

LanguageUtil: 设置平板应用语言、获取中文拼音首字母

PixelUtil: dp转像素，平板宽、高

ProxyUtil: 视频缓存

ReportUtil: 诊断报告工具，读写诊断报告、上传报告

SoundUtil: 音频工具，设备断开连接提示音、退出诊断提示音

VolleyUtil: 网络请求工具，利用Volley库进行网络请求

WifiUtil: 判断网络连接，并判断是否为无线WIFI连接

BackHandlerHelper: Fragment返回兼容

CrashHandler: 应用程序一场捕捉

MDBHelper: 数据库方法封装

UI: 用户界面

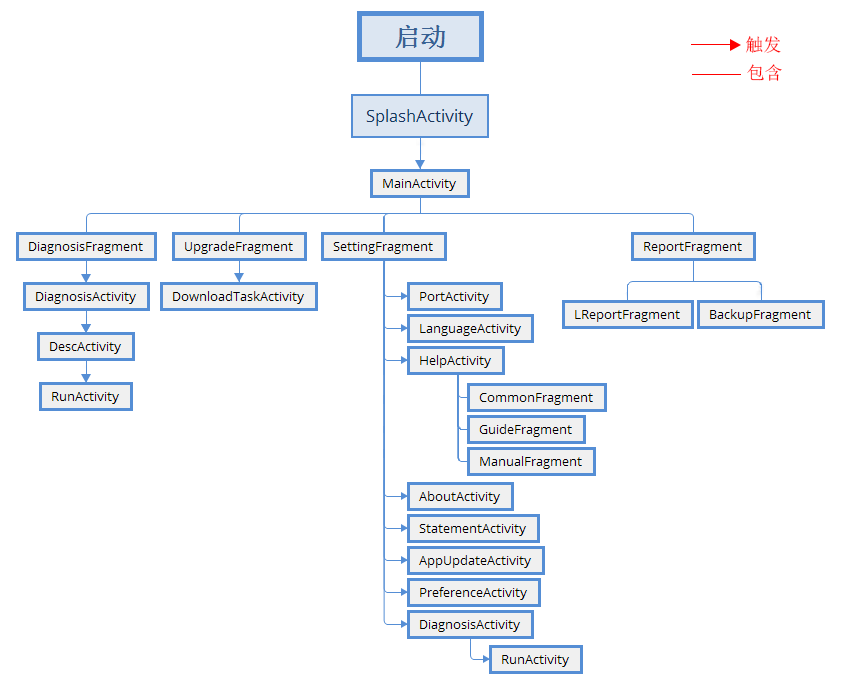
Activity: Activity是用户界面的基本载体，Fragment、Dialog等界面形式都需要依附于Activity，同时Activity也可以是独立的界面展示

Adapter: 列表中相同布局的项

base: Activity与Fragment的超类，抽象类。封装其公用方法

Fragment: 同一个Activity中需要展示多个不同的用户UI。

应用UI结构如图：



1. assets: 供浏览的文件，如pdf文件、html文件

1). document: 快速指南与用户手册的pdf说明文档

2). Statement.html: 免责申明

1. jniLibs: 集成的已经打包的第三方库文件，如jar、so，包括百度地图定位SDK、gson、volley网络请求工具
2. res：资源文件

anim：动画效果

drawable：利用代码构建的图片资源

layout：页面布局

mipmap: 已经编辑完成的png、jpg的图片文件

raw：音频文件，断开连接提示音，退出诊断警报

values: 值，如颜色、字符串、属性

array: 数组

attrs: 第三方控件的自定义属性声明

color: 颜色

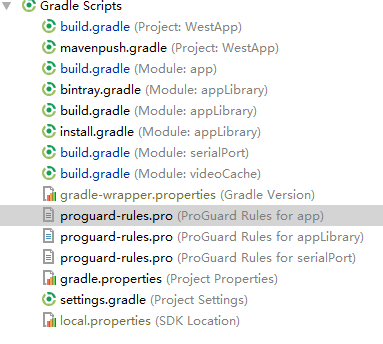
dimens: 尺寸

strings: 字符串；所有的字符串都需要在此资源中声明，便于多语言的实现。

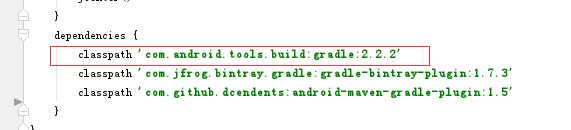
style: 通用样式

五. 应用基础配置

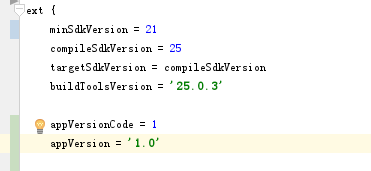
应用基础配置包括项目编译配置、APP版本等，如图：



1). build.gradle(Project:WestAPP)：位于项目根目录



红色选框部分后为Android Studio 版本



minSdkVersion: 供使用的最低SDK版本(21对应的为Android 5.0)

compileSdkVersion: 编译调用的SDK版本（25对应Android 7.1）

targetSdkVersion: 编译的目标设备SDK版本

buildToolsVersion: 编译调用SDK 工具版本

appVersionCode: 打包的APK的版本代码

appVersion: 打包的APK的版本名称

2). build.gradle(Module:app)：位于app模块根目录



compileSdkVersion: 编译的SDK版本（在项目 build.gradle中已配置）

buildToolsVersion: 编译的SDK工具版本（在项目build.gradle中已配置）

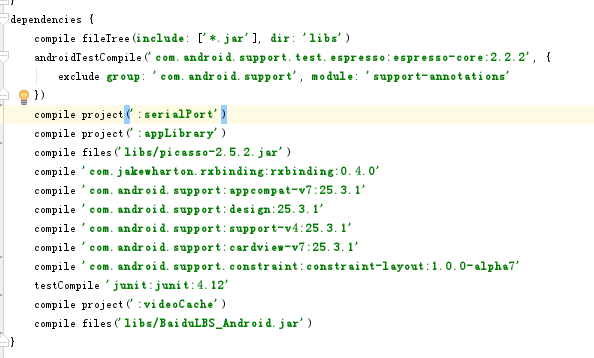
applicationId: 应用的包名

minSdkVersion: 使用APP应用的最小Android版本

targetSdkVersion: 使用的目标Android版本

versionCode: APP应用的版本代码（在项目build.gradle中已配置）

versionName: APP应用的版本名称（在项目build.gradle中已配置）



dependencies: 引用到的第三方库

3). setting.gradle(Project Settings): 位于项目根目录

声明工程中需要使用的库模块（appLibrary/serialPort/videoCache）

六、编译打包成APK（Android安装包）

当工程已经正确导入后，发布应用之前需要将工程编译成Android安装包(APK)，在Android Studio 中打包的方法为：

Build →Generate Signed APK →Choose existing →选择项目根目录下 “westAPP.jks” 文件 →输入下方的密码，密码记录在 jks 文件同级目录的 “jksInfo.txt”文件中。

输入完成后，点击“Next”, 此处 APK Destination Folder编辑打包后APK文件存放的目标路径，默认为工程目录下 app目录；Build Type为编译类型，release为发布模式，debug为调试模式，此处选择release模式。点击Finish。

待打包完成后，Android Studio 会有一个小弹窗(右上角或右下角)，如图：

C:\Users\Dell\Desktop\未标题-2.jpg

点击 “Show In Explorer” 可定位到编译的Apk文件的目录。

说明：由于工程引用了百度地图的定位功能，在编译打包时会出现如下图的提示，该并非工程报错。

