# Gradle 总结

## 一、Gradle用来干什么的？

一个android项目（包括资源文件和源代码）需要经过构建编译最终输出可执行的apk，而Gradle就是一个高级的构建工具用来自动执行和管理构建流程。同时也允许您定义灵活的自定义构建配置。 每个构建配置均可自行定义一组代码和资源，同时对所有应用版本共有的部分加以重复利用。

## 二、android项目的构建流程

构建流程涉及许多将您的项目转换成 Android 应用软件包 (APK) 的工具和流程。

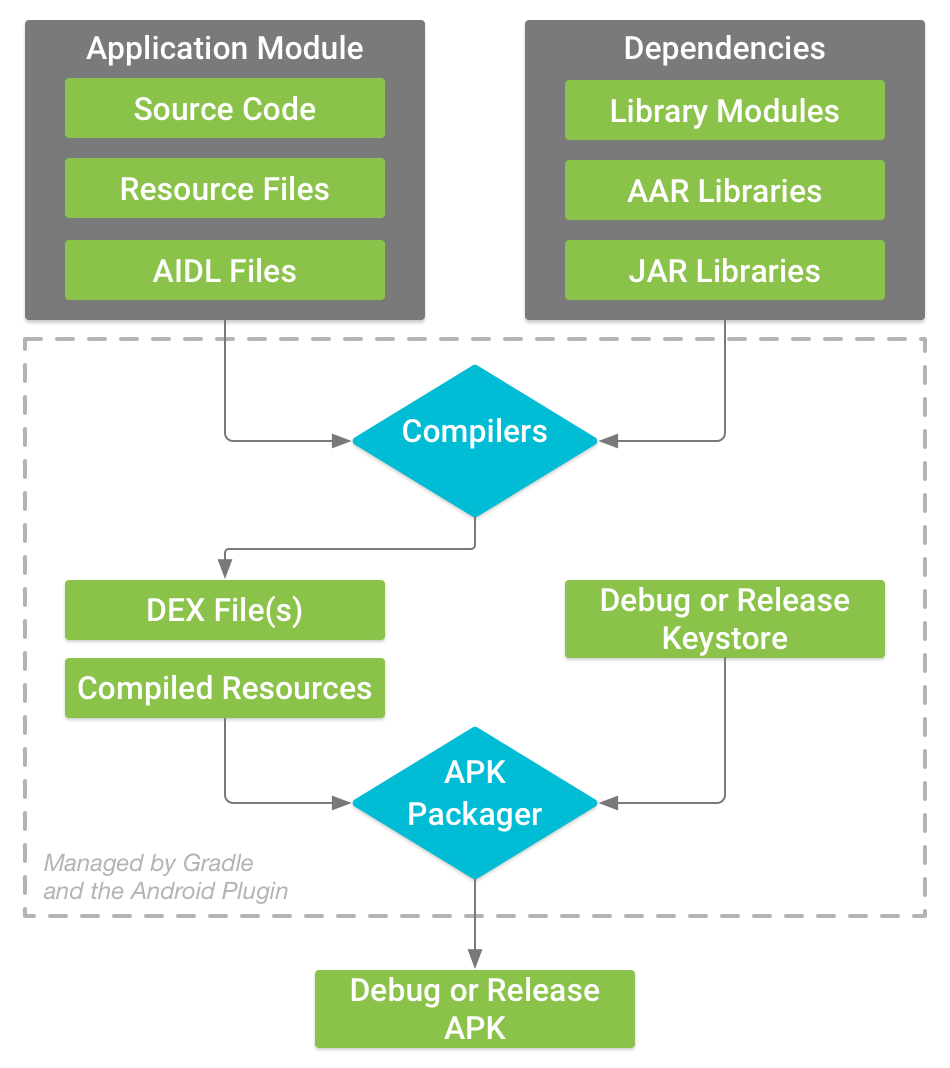


图 1. 典型 Android 应用模块的构建流程。

如图 1 所示，典型 Android 应用模块的构建流程通常依循下列步骤：

1、编译器将您的源代码转换成 DEX（Dalvik Executable) 文件（其中包括 Android 设备上运行的字节码），将所有其他内容转换成已编译资源。

2、APK 打包器将 DEX 文件和已编译资源合并成单个 APK。 不过，必须先签署 APK，才能将应用安装并部署到 Android 设备上。

3、APK 打包器使用调试或发布密钥库签署您的 APK：

A、如果您构建的是调试版本的应用（即专用于测试和分析的应用），打包器会使用调试密钥库签署您的应用。 Android Studio 自动使用调试密钥库配置新项目。

B、如果您构建的是打算向外发布的发布版本应用，打包器会使用发布密钥库签署您的应用。 要创建发布密钥库，请阅读在 Android Studio 中签署您的应用。

4、在生成最终 APK 之前，打包器会使用 zipalign 工具对应用进行优化，减少其在设备上运行时占用的内存。

构建流程结束时，您将获得应用的调试 APK 或发布 APK，您可使用它们进行部署、测试，或向外部用户发布。

## gradle、gradle wrapper、Android Plugin for Gradle区别

● Gradle是个构建系统，能够简化你的编译、打包、测试过程。熟悉Java的同学，可以把Gradle类比成Maven。

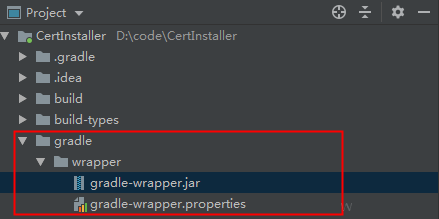
● Gradle Wrapper的作用是简化Gradle本身的安装、部署。不同版本的项目可能需要不同版本的Gradle，手工部署的话比较麻烦，而且可能产生冲突，所以需要Gradle Wrapper帮你搞定这些事情。Gradle Wrapper是Gradle项目的一部分。

● Android Plugin for Gradle是一堆适合Android开发的Gradle插件的集合，主要由Google的Android团队开发，Gradle不是Android的专属构建系统，但是有了Android Plugin for Gradle的话，你会发现使用Gradle构建Android项目尤其的简单。

另外需要说明的一点是Gradle、Gradle Wrapper与Android Plugin for Gradle不一定要和Android Studio一起使用，你可以完全脱离Android Studio，使用三者独立进行Android项目的构建。

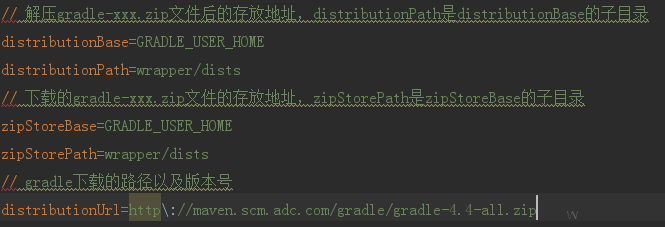
### gradle wrapper

以Android中项目为例，新建一个Android项目时，其目录结构：



在项目目录下会有一个gradle文件夹，gradle-wrapper.jar是Gradle Wrapper的主体功能包，

在Android Studio安装过程中产生gradle-wrapper.jar（如果默认安装的话会在C:\Program Files\Android\AndroidStudio\plugins\android\lib\templates\gradle\wrapper\gradle\wrapper\gradle-wrapper.jar）。然后每次新建项目，会将gradle-wrapper.jar拷贝到你的项目的gradle/wrapper目录中。gradle-wrapper.properties文件主要指定了该项目需要什么版本的**Gradle**，从哪里下载该版本的Gradle以及存放位置。

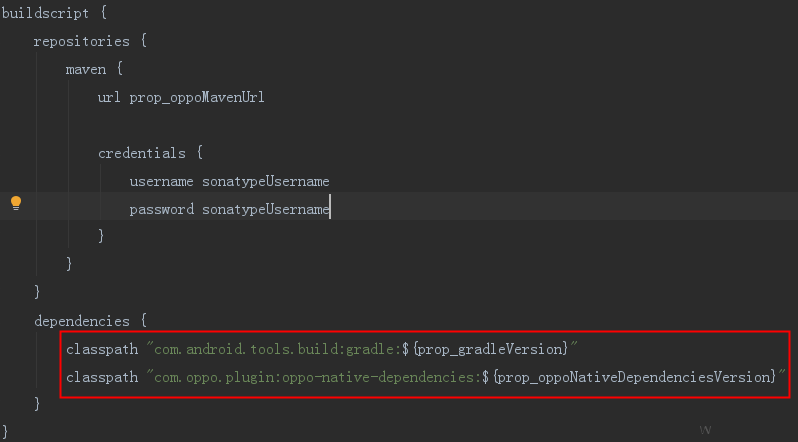


GRADLE\_USER\_HOME：用户的gradle目录，window下的路径为C:\User\用户名\.gradle

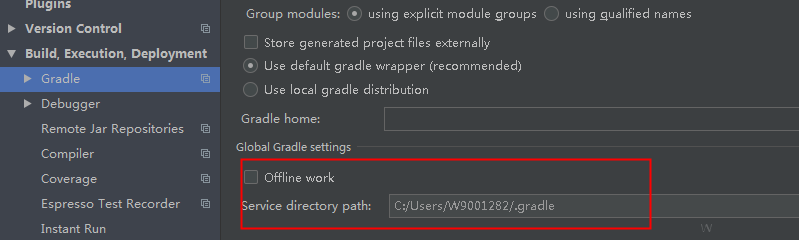
另外，Android Studio默认也带有gradle，路径为/Applications/Android Studio.app/Contents/gradle/gradle-4.1。

### Android Plugin for Gradle

Gradle对应版本下载完成之后，Gradle Wrapper的使命基本完成了，Gradle会读取build.gradle文件，该文件中指定了该项目需要的Android Plugin for Gradle版本是什么，从哪里下载该版本的Android Plugin for Gradle。如下图所示：



从图示项目中可以知道我们要使用3.1.4版本，从google和jcenter处下载，那么下载到我们本地的哪里呢？它会下载到~\.gradle\caches\modules-2\files-2.1\com.android.tools.build中。有时候大家网络装填不好，选择下图中的Offline work时可能出现"No cached version of com.android.tools.build:gradle:xxx available for offline mode"问题，此时你只要将对应版本的Android Plugin for Gradle下载到本地的C:\Program Files\Android\Android Studio\gradle\m2repository\com\android\tools\build中即可。



## Android项目中的Gradle

创建一个新的Android项目，可以看到这里有4处关于gradle的配置文件。

### app/build.gradle

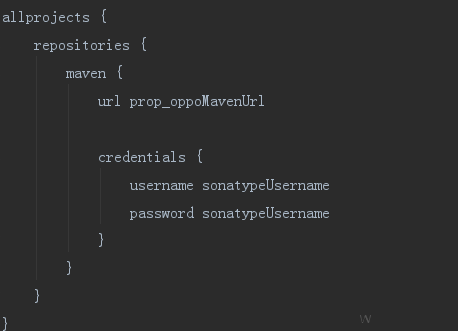
app的gradle配置文件，关于该应用的配置几乎都在这个文件中，平时关注最多的也是这个文件。包含编译的SDK版本，不同编译类型的配置，项目依赖等。

### gralde/wrapper/gradle-wrapper.properties

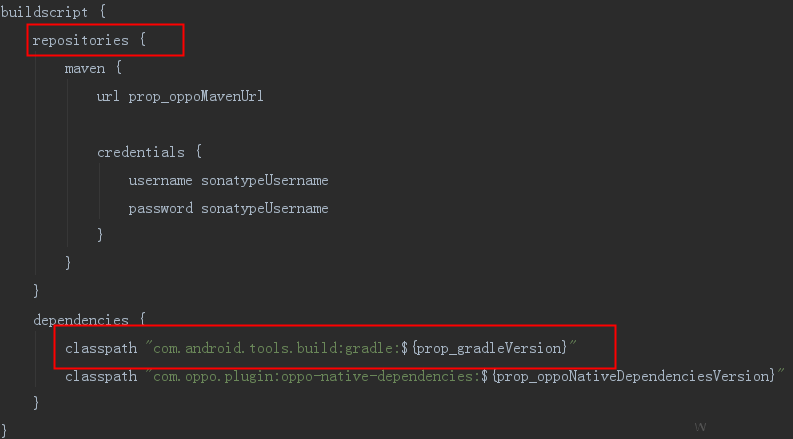
定义Gradle Wrapper的一些信息，请求下载的路径，版本号，以及本地存放的位置。

### Build.gradle

整个项目的gradle配置文件，定义依赖仓库的地址



一般不在module中定义依赖仓地址，allprojects{...}可以用来定义各个模块的默认（共有）属性。

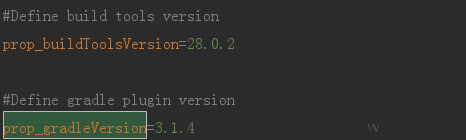


buildscript{...}代码块下的dependencies里面指明编译所用到时的插件，repositories里面指明编译所用到的插件从哪个仓库下载。

Gradle支持多种仓库形式，有远程仓库，本地仓库，以及本地目录形式作为仓库。OPPO编译时，这里会换成OPPO指定的仓库。

**gradle.properties**

在该文件夹定义设计gradle相关的一些属性。在使用gradle的task时候，经常会在此定义一些全局变量，然后直接去调用。

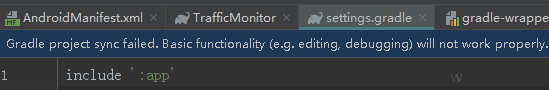


在build.gradle中可以使用这些全局变量。

### Settings.gradle

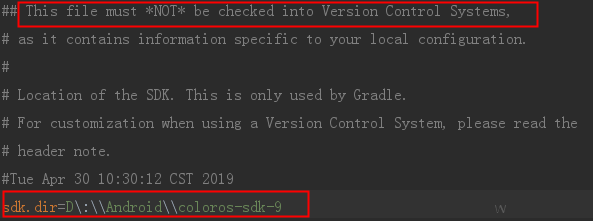
Setting文件可以说是子项目（也可以说是Module）的配置文件，声明整个项目（主工程）中需要gralde的module（子工程）

Android studio 项目 允许 多项目构建，一个主工程对应多个子工程，子工程要在主工程中能够使用，需要在setting.gradle中声明。



### Local.properties

这个文件通过定义sdk.dir和ndk.dir，来指定Gradle编译所用到的Android SDK路径和Android NDK路径。Windows平台需要使用\来转义冒号和路径分隔符，否则无法识别。



不能加入版本控制系统