**快速掌握 Android Studio 中 Gradle 的使用方法**

2014-12-08      [**1**](javascript:void(0);)[个评论](http://www.2cto.com/kf/201412/359002.html#SOHUCS)    来源：码农的自留地

[收藏](javascript:;)    [[http://www.2cto.com/statics/images/dl.jpg](http://www.2cto.com/index.php?m=member&c=content&a=publish&modid=1&siteid=1)我要投稿](http://www.2cto.com/index.php?m=member&c=content&a=publish&modid=1&siteid=1)

Gradle是可以用于Android开发的新一代的 **Build System**， 也是 Android Studio默认的build工具。

Gradle脚本是基于一种JVM语言 -- Groovy，再加上DSL（领域特定语言）组成的。

因为Groovy是JVM语言，所以可以使用大部分的Java语言库。所谓DSL就是专门针对Android开发的插件，比如标准Gradle之外的一些新的**方法（Method）**、**闭包（Closure）**等。

由于Gradle的语法足够简洁，而且可以使用大部分的java包，当之无愧地成为新一代 Build System。

使用Android Studio新建一个工程后，默认会生成两个**build.gralde**文件，一个位于工程根目录，一个位于app目录下。还有另外一个文件 --**settings.gradle**。

根目录下的脚本文件是针对**module**的全局配置，它的作用阈所包含的所有 module 是通过settings.gradle来配置。

**app**文件夹就是一个module，如果在当前工程中添加了一个新的module -- **lib**，就需要在settings.gralde文件中包含这个新的module。

**gradle脚本的基本结构**

用我现在的工程举例来说，根目录的build.gradle的内容如下所示：

[?](http://www.2cto.com/kf/201412/359002.html)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23 | // Top-level build file where you can add configuration options common to all sub-projects/modules.    buildscript {      repositories {          jcenter()      }      dependencies {          classpath 'com.android.tools.build:gradle:1.0.0-rc4'            // NOTE: Do not place your application dependencies here; they belong          // in the individual module build.gradle files      }  }    allprojects {      repositories {          jcenter()            maven {              url '<http://mvnrepo.xxx.com/mvn/repository>'          }      }  } |

classpath 'com.android.tools.build:gradle:1.0.0-rc4'就是Android特有的插件，maven仓库位于通过方法**jCenter()** 获取，这也是默认的maven仓库。当然也可以添加额外的maven仓库地址，例如以上文件中的

[?](http://www.2cto.com/kf/201412/359002.html)

|  |  |
| --- | --- |
| 1  2  3 | maven {      url '<http://mvnrepo.xxx.com/mvn/repository>'  } |

然后是 settings.gradle 文件：

[?](http://www.2cto.com/kf/201412/359002.html)

|  |  |
| --- | --- |
| 1 | include ':app' |

app就是项目包含的一个module，如果有多个module，可以在为 include 方法添加多个参数。

最后是app/build.gradle

[?](http://www.2cto.com/kf/201412/359002.html)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70 | apply plugin: 'com.android.application'    android {      compileSdkVersion 21      buildToolsVersion "21.1.1"        compileOptions {          sourceCompatibility JavaVersion.VERSION\_1\_7          targetCompatibility JavaVersion.VERSION\_1\_7      }          defaultConfig {          applicationId "your.application.id"          minSdkVersion 14          targetSdkVersion 21          versionCode 2          versionName "2.0.0"      }        signingConfigs {          release {              storeFile file('release.keystore')              storePassword "yourstorepassword"              keyAlias "yourkeyalias"              keyPassword "yourkeypassword"          }          debug {              storeFile file('debug.keystore')          }      }        buildTypes {          release {              minifyEnabled true              proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'              signingConfig signingConfigs.release          }          debug {              signingConfig signingConfigs.debug          }      }        productFlavors {          inner {              applicationId "your.application.inner.id"              versionName "2.0.0"          }            market {            }      }  }    repositories {      flatDir {          dirs 'libs'      }  }    dependencies {        // 通用      compile name: 'volley', ext: 'aar'      compile 'com.nostra13.universalimageloader:universal-image-loader:1.9.3'      compile 'com.alibaba:fastjson:latest.integration'        // 项目相关（已删除）  } |

**Groovy 的基本语法**

**方法调用**

[?](http://www.2cto.com/kf/201412/359002.html)

|  |  |
| --- | --- |
| 1 | apply plugin: 'com.android.application' |

以上语句中的apply是一个方法，给它传递了一个参数plugin，plugin 的值是'com.android.application'。

如果有多个参数，则以逗号隔开，例如

[?](http://www.2cto.com/kf/201412/359002.html)

|  |  |
| --- | --- |
| 1 | <code>compile name: 'volley', ext: 'aar'</code> |

**闭包**

Groovy中花括号包含的部分成为一个闭包（Closure）。例如下面的代码

[?](http://www.2cto.com/kf/201412/359002.html)

|  |  |
| --- | --- |
| 1  2  3  4 | compileOptions {      sourceCompatibility JavaVersion.VERSION\_1\_7      targetCompatibility JavaVersion.VERSION\_1\_7  } |

compileOptions 是一个 Method， 它的参数是一个闭包，这个闭包内依次执行了两个方法 --sourceCompatibility 和targetCompatibility， 参数都是JavaVersion.VERSION*1*7。

闭包也可以嵌套包含

[?](http://www.2cto.com/kf/201412/359002.html)

|  |  |
| --- | --- |
| 1  2  3  4  5 | repositories {      flatDir {          dirs 'libs'      }  } |

**常见使用方法**

**包依赖（aar）**

使用aar时可以分为两种情况

**① aar位于本地目录**

首先在 android 的参数闭包中添加调用方法 repositories

[?](http://www.2cto.com/kf/201412/359002.html)

|  |  |
| --- | --- |
| 1  2  3  4  5 | repositories {      flatDir {          dirs 'libs'      }  } |

然后在 dependencies 的参数闭包中添加

[?](http://www.2cto.com/kf/201412/359002.html)

|  |  |
| --- | --- |
| 1 | compile name: 'volley', ext: 'aar' |

**② aar位于远程仓库**

这里以maven为例，当然也可以使用其他类型的仓库，例如 **Ivy**。

只需要在jar包引用方式后面添加一个@aar就可以了

[?](http://www.2cto.com/kf/201412/359002.html)

|  |  |
| --- | --- |
| 1 | compile 'com.alibaba:fastjson:latest.integration@aar' |

**包依赖（jar）**

[?](http://www.2cto.com/kf/201412/359002.html)

|  |  |
| --- | --- |
| 1 | compile group: 'com.alibaba', module: 'fastjson', version: 'latest.integration' |

可以简写成

[?](http://www.2cto.com/kf/201412/359002.html)

|  |  |
| --- | --- |
| 1 | compile 'com.alibaba:fastjson:latest.integration' |

latest.integration可以替换成具体的版本号，这里是获取服务器上的最新版本。

**去掉重复依赖**

[?](http://www.2cto.com/kf/201412/359002.html)

|  |  |
| --- | --- |
| 1  2  3 | compile 'com.alibaba.fastjson.latest.integration' {      exclude module: 'annotations', group: 'com.google.android'  } |

**使用 Java7**

[?](http://www.2cto.com/kf/201412/359002.html)

|  |  |
| --- | --- |
| 1  2  3  4 | compileOptions {      sourceCompatibility JavaVersion.VERSION\_1\_7      targetCompatibility JavaVersion.VERSION\_1\_7  } |

**productFlavors**

针对不同的APP分发渠道，我们可以定义不同的 **product flavor**。也可以定义**内部版本**和**外部版本**，内部版本中包含了一些调试代码，这些代码在发布时并不会被编译进最后的APP中。而且可以分别为内部版本和外部版本指定不同的ApplicationId，这样在同一个设备上可以同时安装两个版本以方便调试。

**命令行执行Gradle脚本**

在Android工程根目录下会自动生成一个shell脚本 - **gradlew**，执行之前记得加上x属性 -chomod +x gradlew

gradle脚本中包含了很多 **task**，可以通过task名来指定需要执行的task。

./gradlew build./gradlew assemble./gradlew assembleInnderDebug

**总结**

不得不说，Gradle实在太好用了！虽然 Gradle 可以与 Ant 或 maven 配合使用，但是其简洁和功能性远远超过其他两个。我现在开发的项目普遍使用的是 maven，不知道什么原因，使用Gradle时经常会遇到一些无法获取远程依赖包的问题，最简单的解决办法就是把依赖包[下载](http://www.2cto.com/soft)的本地。