

[全部经验分类](#)[Android \(/lib/tag/Android\)](#) [iOS \(/lib/tag/IOS\)](#) [JavaScript \(/lib/tag/JavaScript\)](#)[\(/lib/list/all\)](#) [所有分类 \(/lib/list/all\)](#) > [开发语言与工具 \(/lib/list/36\)](#) > [Android Studio开发工具 \(/lib/list/399\)](#)

拥抱 Android Studio 之一：从 ADT 到 Android Studio

[Android Studio \(/lib/tag/Android Studio\)](#) 2015-12-03 08:16:47 发布

您的评价: 5.0

收藏

6收藏

拥抱变化，拥抱新事物

多年前央视有一套电视系列片：大国崛起

(<http://baike.baidu.com/subview/634340/5376026.htm>)，主要讲述那些曾经在站在世界舞台中央的国家的兴衰历程。纵观历史，正是创新使得一个国家得以崛起。

Android 开发因为涉及到代码编辑、UI 布局、打包等工序，最好使用一款 IDE。Google 最早提供了基于 Eclipse 的 ADT 作为开发工具，后于2013年 Google I/O 大会发布基于 IntelliJ IDEA 开发 Android Studio (https://zh.wikipedia.org/wiki/Android_Studio)。后者正式版发布之后，Google 宣布不再持续支持 ADT。Android Studio 有更快的速度，更好的代码完成等特性。

时至今日（2015-11-30），在我们运营 bugtags.com (<https://bugtags.com>)这一款高效的 app 测试服务的时候，在帮助用户集成 Android SDK 的过程中，我们发现很多遗留项目依旧没有从 ADT 迁移过来，依然有很多用户对 Android Studio 怀着恐惧与不信任。我想我们有义务去做一个系列介绍，让更多的人拥抱变化，拥抱新鲜事物。

本文系列一，主要介绍两个工具的比较及其迁移。

Android Studio (IntelliJ IDEA) vs. ADT (Eclipse)

本文并不打算涉及 IDE 的圣战，理论上只要配上插件，用得趁手，都差不多。然而本文也是带有倾向性的，因为工作原因，笔者使用过 Eclipse 多年，也使用过多款 JetBrains 的 IDE，均不辱 XXX 语言开发第一 IDE 之名。下面介绍一些 Android Studio 的优胜的地方。

- 更快的运行速度；
- 更智能的自动补全；

你不用自动补全？那其实你 ADT 也不需要的，vi 或者一个记事本就足够了；

- 更好的重构，IDEA 会帮你找到每一个需要重构的地方；
- 更好的索引与搜索，双击 shift 键搜索，文件搜索，代码搜索，都很方便；
- 更好的版本管理功能（git/SVN）；
- 更灵活强大集成脚本 gradle；
- 原生支持的从项目文件到操作系统的文件浏览器功能；

ADT 可以通过插件支持？问题来了，我为啥要去花宝贵的时间折腾这个插件呢；

-

以上省略一万字。

相信我，对于 Android 开发者，切换到 Android Studio，经过短短学习曲线之后，你会爱上这个 IDE，并且即将由奢入俭难，再也回不去了。

从 ADT 迁移到 Android Studio

如果你看到上面的优点，已经克服了心中的怀疑和恐惧，准备动手迁移，下面提供一些迁移提示。

常用网站

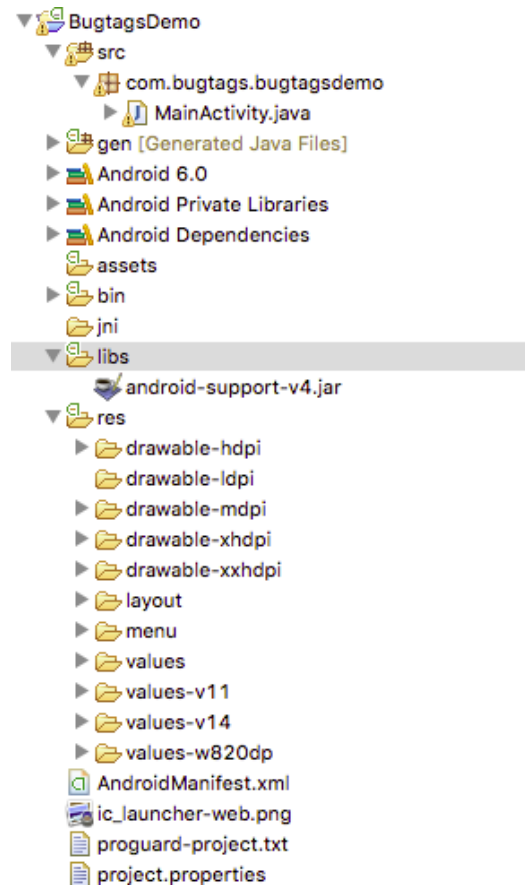
部分需要科学上网

你无法科学上网？赶紧搜一下，自己搭或者买一个吧。世界如此多娇，人生苦短。将你遇到过问题的出错信息用 Google 搜索，第一条来自 stackoverflow 的答案就是你想要的。

- Android Studio 下载页 (<http://developer.android.com/sdk/index.html>)
- Android Studio (<http://developer.android.com/tools/studio/index.html>)：Android Developer 上 Android Studio 的介绍
- Android Tools (<http://tools.android.com/>)：Android 工具首页，有大量的深度的技术文档
- Android Build Sytem (<http://tools.android.com/tech-docs/new-build-system>)：基于 gradle 的 Android 构建系统，最为关键的一篇文章
- 从 ADT 迁移指南 (<http://developer.android.com/sdk/installing/migrate.html>)：官方提供的迁移手册
- Android Studio 国内下载渠道 (<http://androiddevtools.cn/>)：良心网站，你懂的，但是笔者不对 Xcode Ghost之类的事情负责

ADT 与 Android Studio 目录结构对比

ADT 项目结构



([http://7xligz.com1.z0.glb.clouddn.com/15-](http://7xligz.com1.z0.glb.clouddn.com/15-12-1/49630646.jpg)

12-1/49630646.jpg)

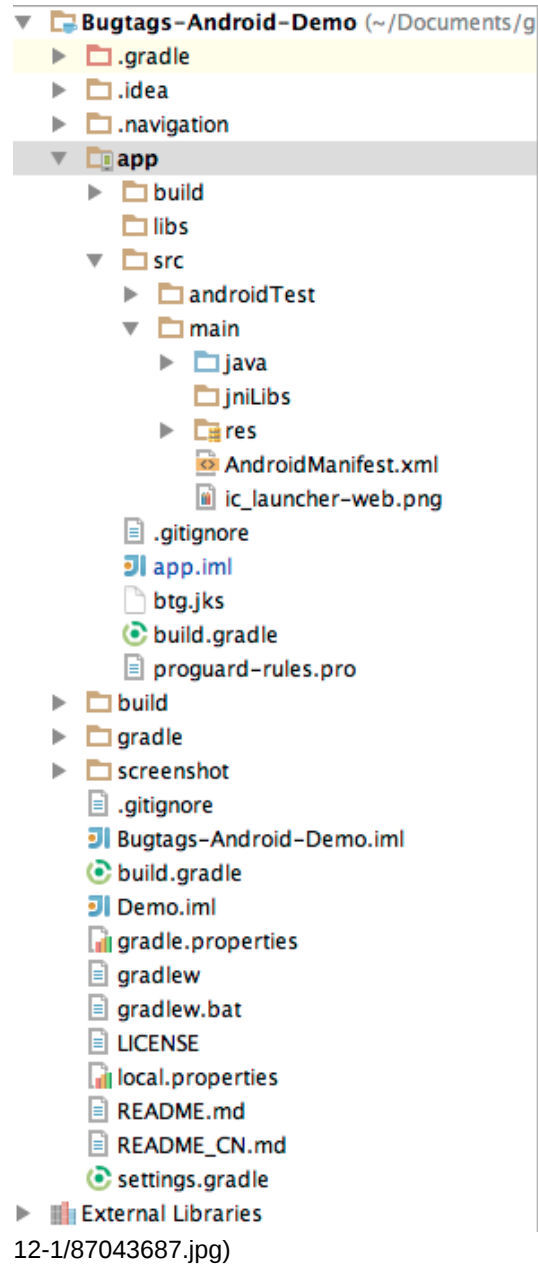
project/src/：放置代码文件

project/libs：放置 .jar 库或者 .so 库

project/res：项目资源文件

project/AndroidManifest.xml：工程 manifest 文件

Android Studio 项目结构



(<http://7xligz.com1.z0.glb.clouddn.com/15-12-1/87043687.jpg>)

Android Studio 的一个 project (项目)下可以包含多个独立的 module (模块)，图中 Bugtags-Android-Demo 便是一个项目，app 便是一个 module。结构如下。

project/build.gradle：项目的 build.gradle 文件，主要是全局的远程库配置、gradle 插件版本等信息

project/app：名字为 app 的模块

project/build.gradle：模块的 build.gradle 文件，**是最关键的配置文件**，后续会有文章继续专门介绍 gradle build system

project/app/src：源码文件夹

project/app/src/java：放置代码文件

project/app/src/jniLibs：默认的.so 库放置的位置，可以在模块的 build.gradle 中通过 jniLibs.srcDirs 值来修改所在位置

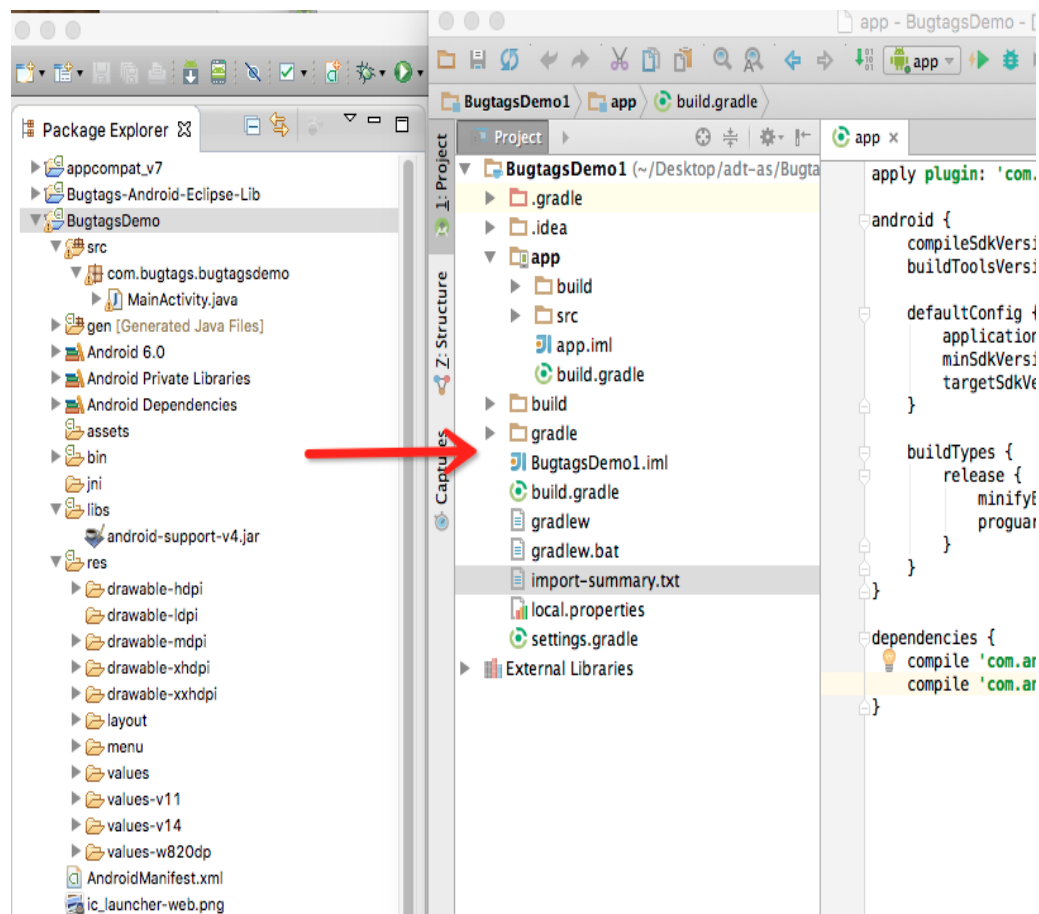
project/app/libs：放置 .jar 库

从 ADT 迁移

本文主要参考这篇迁移指南 (<http://developer.android.com/sdk/installing/migrate.html>)。Android Studio 提供了非常友好的导入向导。

在 Android Studio 导入

File->New->Import Project，找到你的 ADT 的 workspace 下的项目，一路点击 YES，会生成一个全新的项目。



(<http://7xligz.com1.z0.glb.clouddn.com/15-12-1/89986096.jpg>)

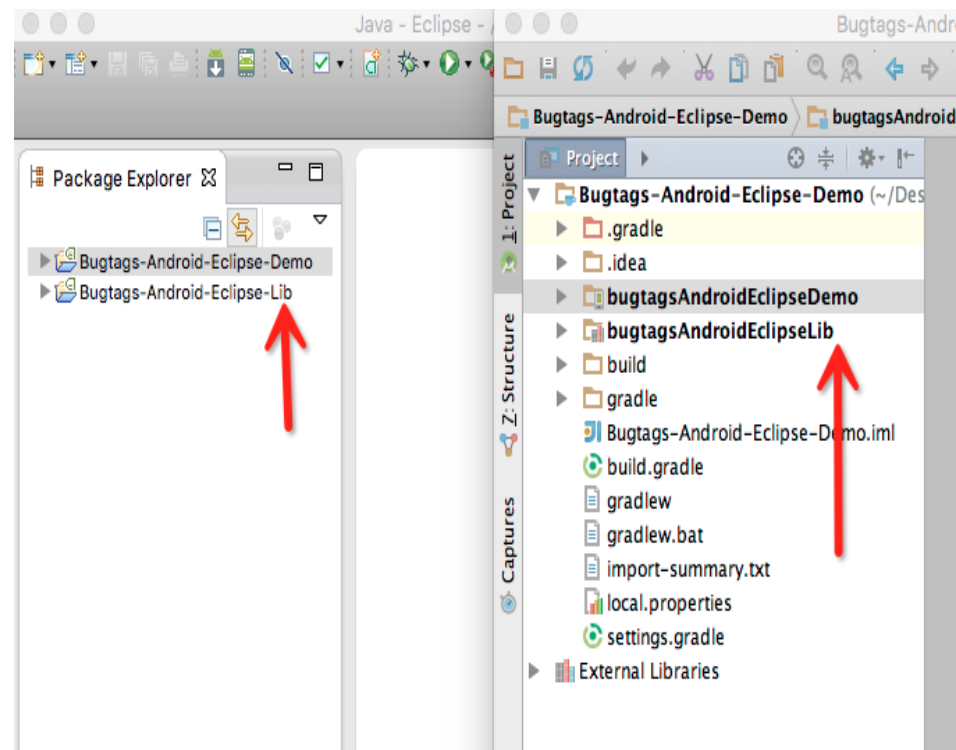
问题

- 项目之间有依赖

直接导入使用依赖的项目，被依赖的项目就会自动被导入到同一个项目中，作为一个模块。

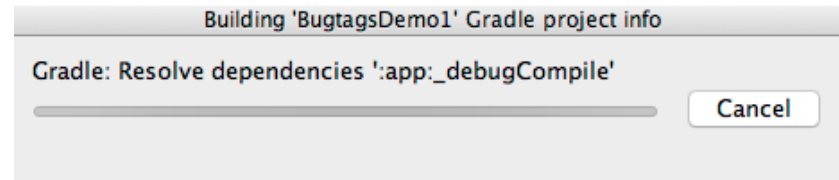
下文中的左侧是 ADT 项目，右侧是 Android Studio 项目。

其中 Bugtags-Android-Eclipse-Demo 使用 Bugtags-Android-Eclipse-Lib 作为依赖。



(<http://7xligz.com1.z0.glb.clouddn.com/15-12-1/41683475.jpg>)

- 停留在界面上不动



(<http://7xligz.com1.z0.glb.clouddn.com/15-12-1/84638123.jpg>)

这个状态是 gradle 构建脚本尝试去下载 gradle 的 wrapper，这个 wrapper 大概 3M，**需要科学上网**，早期很多人就是在这个页面卡住，放弃了 Android Studio。确保你已经科学上网，并且耐心等待，又或者点击取消，先进项目看看。

Android Studio 的几个关键概念

如果上面没有问题，你已经能够正常的看到 Android Studio 的项目工程了。

有问题？在文章下留言或者加 qq 群：453503476，希望能帮到你。

IDE 使用、快捷键什么的，笔者认为没什么可写的。自己熟悉吧。下面介绍一个对于 ADT 而言全新的概念。

build.gradle

在项目的 build.gradle 文件中，有如下内容：

```
// Top-level build file where you can add configuration options common to
buildscript {
    repositories {
        jcenter()//这是构建插件所使用的远程库，后续会有一篇介绍库的含义
    }
    dependencies {
        classpath 'com.android.tools.build:gradle:2.0.0-alpha1'//这是项目
    }
}

allprojects {
    repositories {
        jcenter()//这是使用远程库依赖的时候，的远程库
    }
}
```

在每个模块的 build.gradle 文件中，初始会有如下内容：

```
apply plugin: 'com.android.library' //com.android.library标示是一个库模块；
//com.android.application标示这是一个应用模块

android {
    compileSdkVersion 19//编译 sdk 版本
    buildToolsVersion "23.0.2"//构建工具版本

    defaultConfig {
        minSdkVersion 9 targetSdkVersion 19 }

    buildTypes {//打包类型
        release {
            minifyEnabled false//是否混淆
            proguardFiles getDefaultProguardFile('proguard-android.txt')
        }
    }

    dependencies {//依赖管理
        compile 'com.bugtags.library:bugtags-lib:1.0.8'//远程
        compile fileTree(dir: 'libs', include: ['*.jar'])//本地 jar 依赖
    }
}
```

依赖管理

远程依赖：

```
dependencies {  
    compile 'com.bugtags.library:bugtags-lib:1.0.8' }
```

Android Studio 使用 gradle 进行依赖管理，兼容了 maven 的库，一个经典的 maven 库的配置如下：

```
<dependency>  
    <groupId>com.bugtags.library</groupId>  
    <artifactId>bugtags-lib</artifactId>  
    <version>1.0.8</version>  
</dependency>
```

对比之后，你会发现 compile 的内容中，分为三部分 groupId:artifactId:version

com.bugtags.library对应groupId

bugtags-lib对应 artifactId

1.0.8对应 version

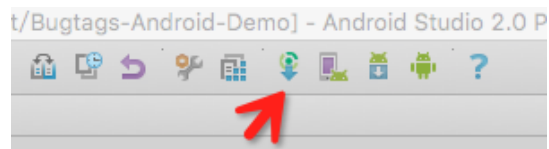
本地依赖

```
dependencies {  
    compile fileTree(dir: 'libs', include: ['*.jar'])//本地 jar 依赖  
}
```

Android Studio 工程创建的时候，默认会有如上依赖，意义是把放在与该 build.gradle 文件同一层的 libs 文件夹中所有的 jar 文件都作为库依赖。

同步依赖

一般情况下，你配置完了build.gradle之后，该编辑框的上方会出现正在 sync 的标记。耐心等待完成即可。如果发现没有同步成功，可以使用下面这个按钮：



(<http://7xligz.com1.z0.glb.clouddn.com/15-12-1/19304918.jpg>)

如果长时间未成功，则可以在控制台使用：

```
./gradlew build --info
```

以上命令来看控制台输出。

build variant

Android Studio 内置了多渠打包功能，就是所谓的 build variant，可以新建若干个 productflavor 来达到目的。大致可以这样：



(<http://7xligz.com1.z0.glb.clouddn.com/15-12-1/21680795.jpg>)

这个功能十分的灵活和强大，举个例子，你想要有两个版本，一个收费版，一个免费版，在若干代码文件和资源文件上，都不大一样。

以前你可能选择在代码中判断当前版本，加载不同的代码和资源。或者使用 git 的 branch。

现在你可以使用 Android Studio 直接使用多个 productflavor，在对应的 flavor 放置那些有区别的代码和文件。一次打包，即可完成。在下一篇系列文章中，笔者将会详细介绍这个功能。

总结

新版本的 Android Studio 还有更多强大的功能，包括最新 preview 的 instant build，将会极大提升构建速度。后文将会详细介绍。

一个工程从 ADT 迁移到 Android Studio 只需要很短的时间，连带熟悉基本概念，半天基本可以完成。如果还没有进行迁移的同学，要抓紧了，不要被世界抛弃了。

笔者 kvh (<http://kvh.io>)先在这里挖一个系列坑，希望这个系列能够让读者快速掌握 Android Studio 与 gradle，请持续关注。

- 拥抱 Android Studio 之二：Android Studio 与 gradle 深入
- 拥抱 Android Studio 之三：maven 库，本地库与发布到 bintray 或者 mavenCentral
- 拥抱 Android Studio 之四：gradle 插件使用与开发

来自：<http://my.oschina.net/stephenzl/blog/537975>

扩展阅读

拥抱Android Studio：从Eclipse到Android Studio的完美迁徙

(</lib/view/open1452320531401.html>)

Android 开发小贴士 (</lib/view/open1453898074230.html>)

拥抱Android Studio（二）：Android Studio与Gradle深入

(</lib/view/open1450683325183.html>)

Visual Studio 2015 和 Apache Cordova 跨平台开发入门(一)

(</lib/view/open1438774248754.html>)

Android Studio使用教程图文详解怎么用Android Studio (</lib/view/open1434591422958.html>)

为您推荐

Android类库框架汇总：awesome-android (</lib/view/open1442664464415.html>)

在已有项目上构建持续集成平台 (</lib/view/open1452421062417.html>)

Dojo Mobile 1.7 入门 (</lib/view/open1355831701916.html>)

WEB前端规范 (</lib/view/open1449061887764.html>)

【译】60个有用CSS代码片段 (</lib/view/open1471501464485.html>)

更多

Android Studio (<https://www.baidu.com/s?wd=site:open-open.com Android Studio>)

Android Studio开发工具 (<https://www.baidu.com/s?wd=site:open-open.com Android Studio 开发工具>)

1. Android开发周报：谷歌推出Android Wear界面定制应用 (<http://www.open-open.com/news/view/1a51887>)
2. 解读2015之Android篇：横向扩张、平稳发展 (<http://www.open-open.com/news/view/a8a06b>)
3. 是时候抛弃Eclipse转向IntelliJ了 (<http://www.open-open.com/news/view/1464631>)
4. 如何开发一个App（Android） (<http://www.open-open.com/news/view/2ea38a>)
5. Android Studio 1.3 Preview 1 发布 (<http://www.open-open.com/news/view/1a95a54>)
6. Android Studio 1.3 Preview 3 发布 (<http://www.open-open.com/news/view/1e9778e>)

同类热门经验

1. Android Studio教程从入门到精通 (/lib/view/open1433387390635.html)
2. 最强 Android Studio 使用小技巧 and 快捷键 (/lib/view/open1458715872710.html)
3. 最全面的Android Studio使用教程 (/lib/view/open1416883124753.html)
4. Android Studio - 基于IntelliJ IDEA全新的Android开发工具 (/lib/view/open1368686066578.html)
5. Android Studio项目目录结构介绍 (/lib/view/open1436323668817.html)
6. Android Studio 新手完全指引 (/lib/view/open1434029865551.html)

阅读目录

拥抱变化，拥抱新事物

Android Studio（IntelliJ IDEA）vs. ADT（Eclipse）

从 ADT 迁移到 Android Studio

常用网站

ADT 与 Android Studio 目录结构对比

ADT 项目结构

Android Studio 项目结构

从 ADT 迁移

在 Android Studio 导入