Android Studio使用指南

# 开发环境

Version: Android Studio 1.4

Ubuntu: ubuntu 12.04 LTS

# Android Studio优势

丰富的模板代码，让程序创建更加简单便捷。

强大的智能代码补全，节时省力。

界面更加友好人性化，颜色、图片所见即所得，更好的实时预览效果。

代码搜索、查看和跟踪更方便，只需掌握几个常用的快捷键，能极大提高效率。

多屏预览、截图带有设备框，可随时录制视频。

自动导入所需的包，添加库更快捷。

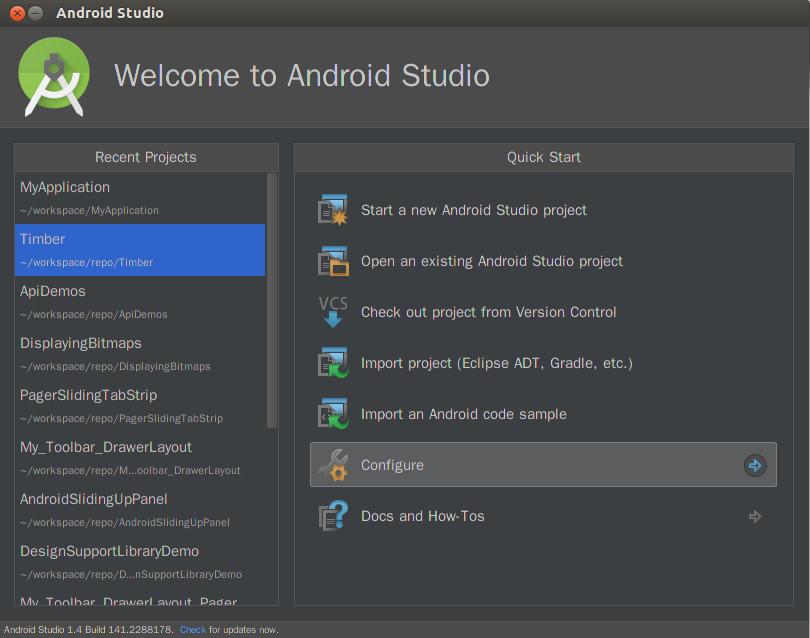
内置终端。

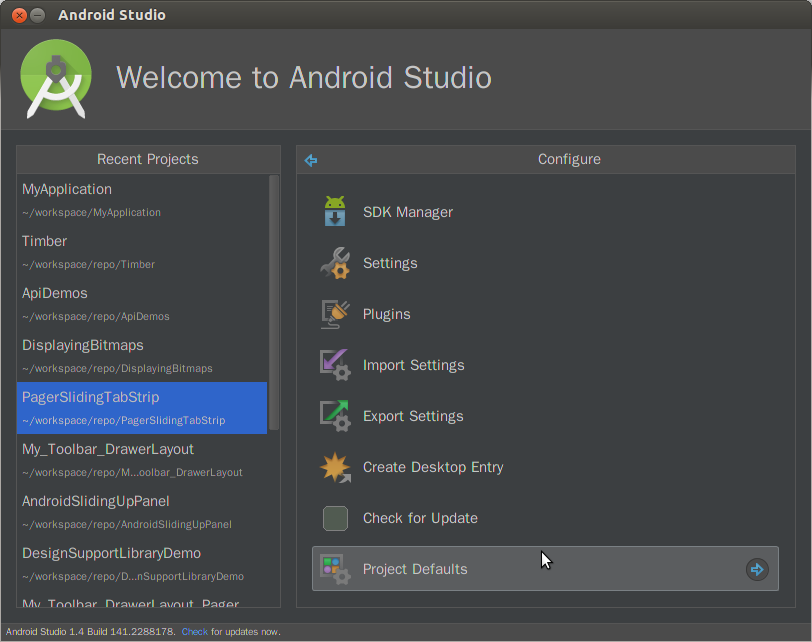
谷歌专为Android量身打造，大势所趋。

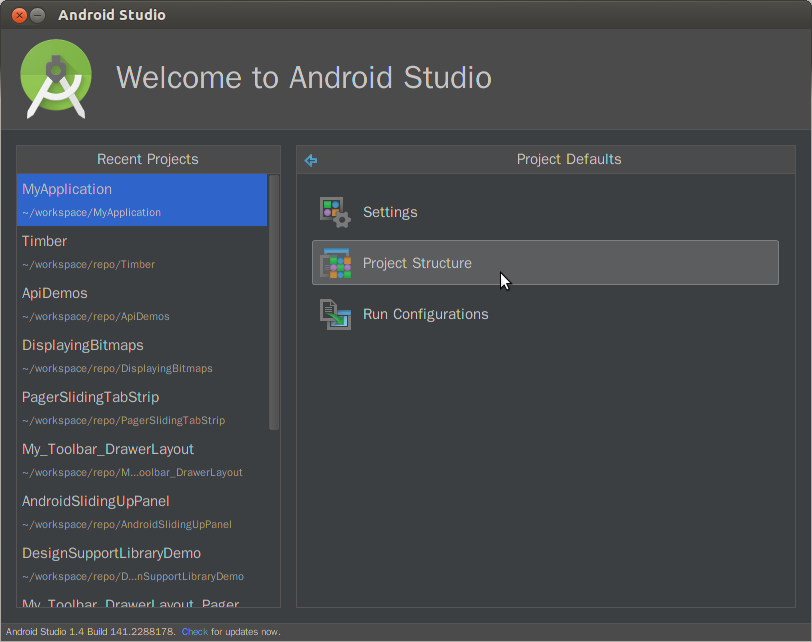
# 安装

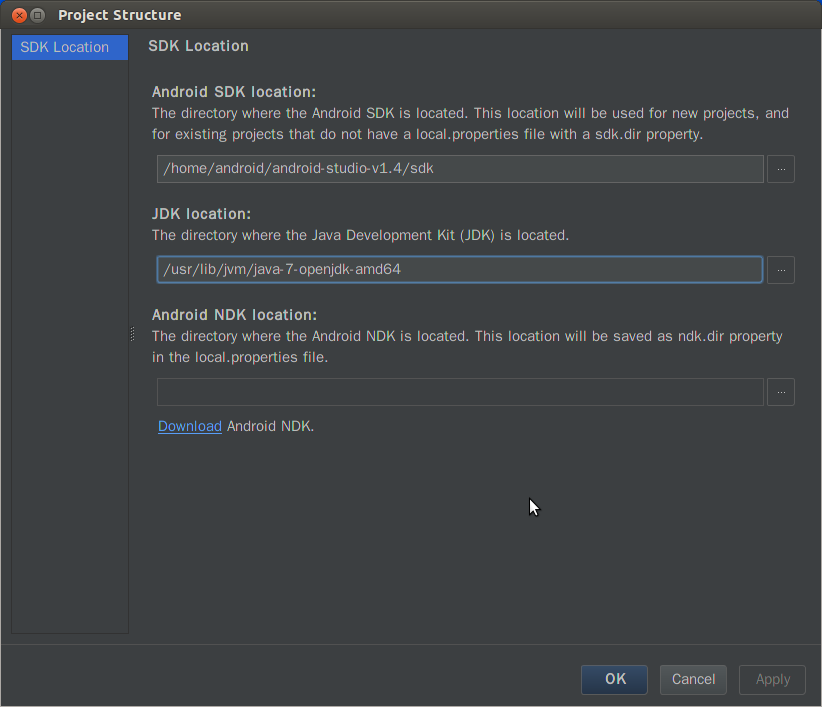
解压之后，运行bin目录下的studio.sh文件。

第一次打开Android Studio(以下简称AS)时，需要配置JDK和SDK。

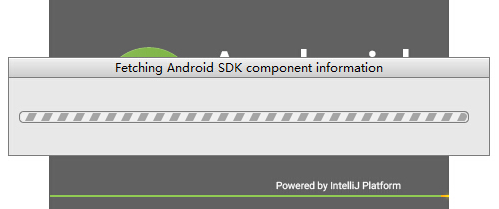








安装好了以后首次运行AS可能一直停在Fetching Android SDK component information。如下界面：



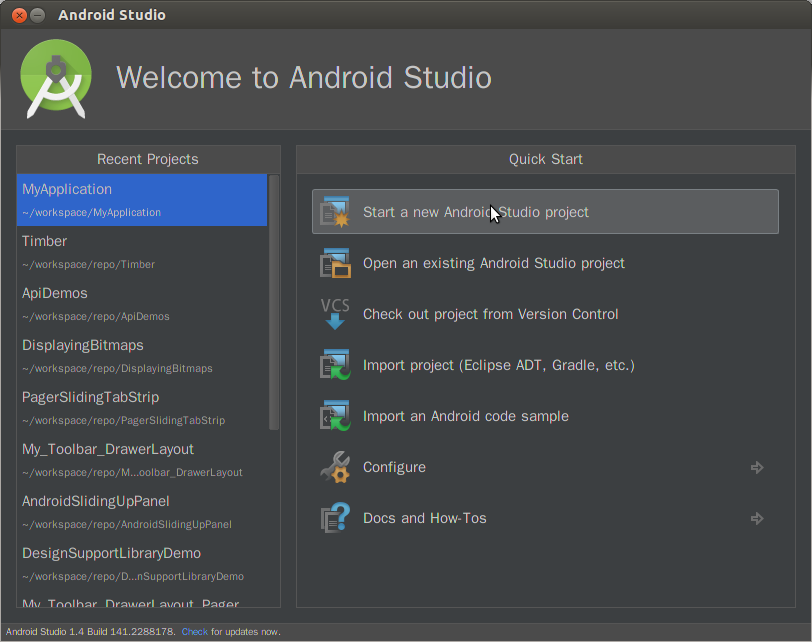
这是因为天朝的墙把首次运行更新SDK给墙了。解决办法就是关闭安装向导，如果无法关闭可以在任务管理器中手动关掉进程，然后打开AS安装目录下的bin目录里面的idea.properties文件，添加一条禁用开始运行向导的配置项：

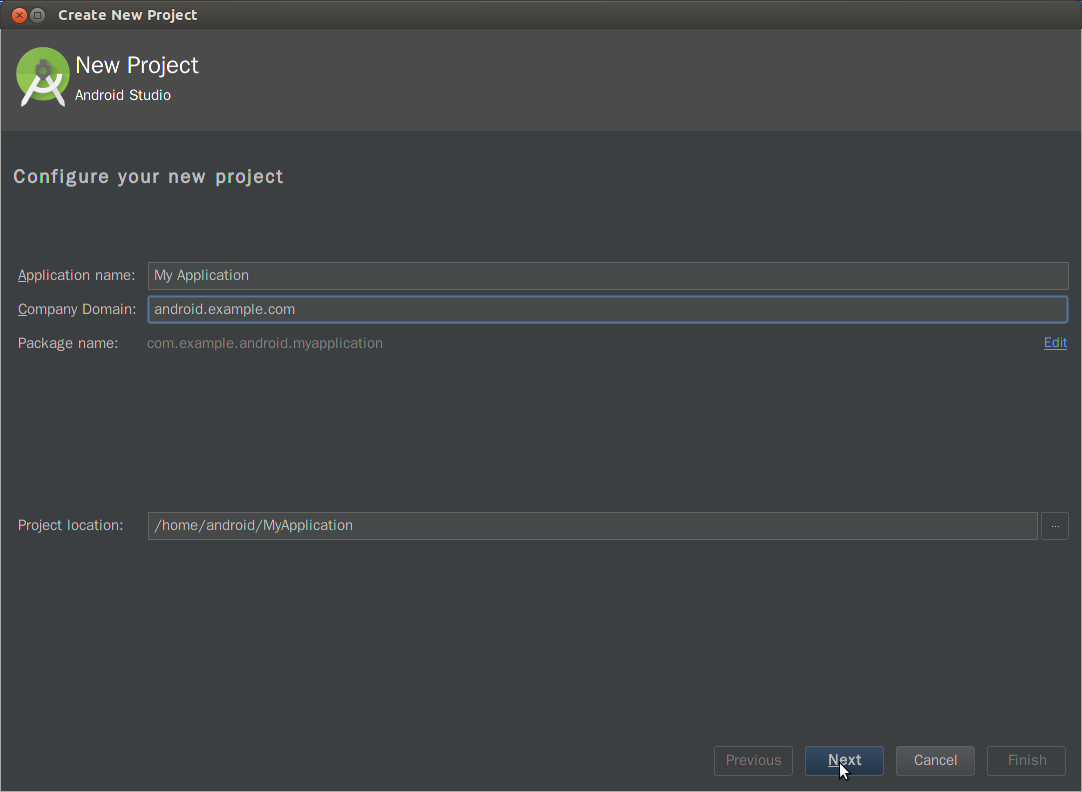
disable.android.first.run=true

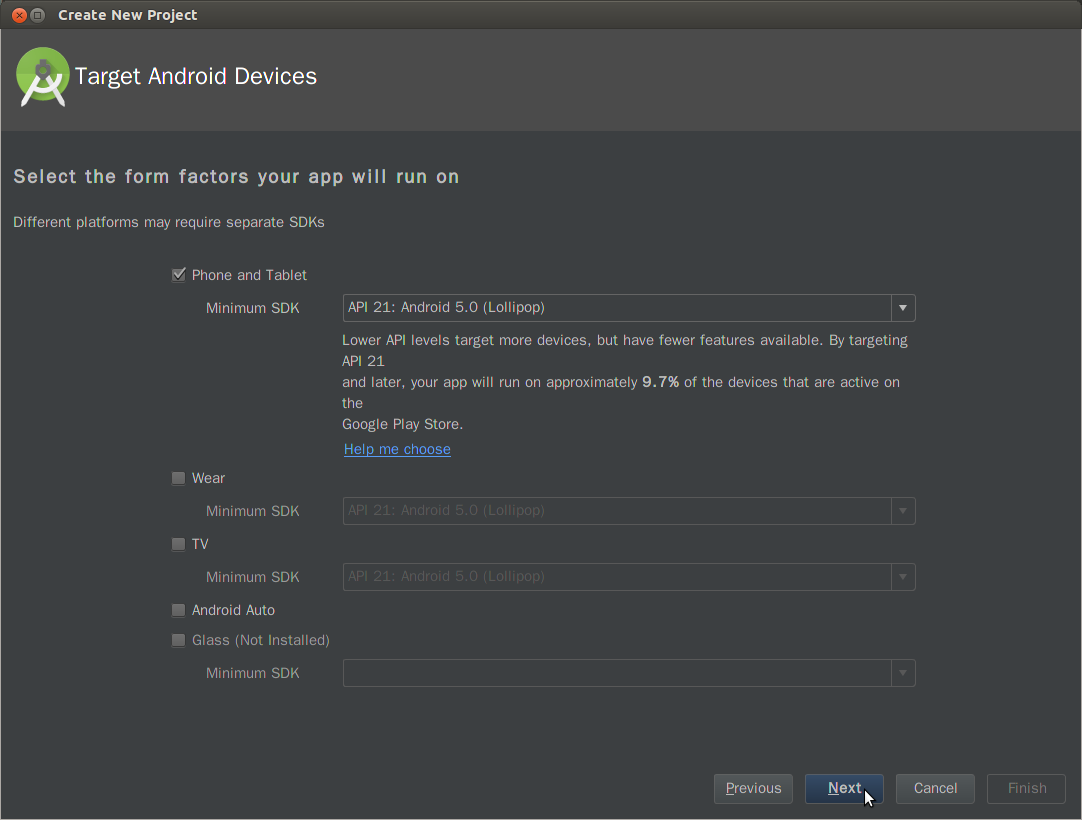
然后再启动程序就会打开项目向导界面。

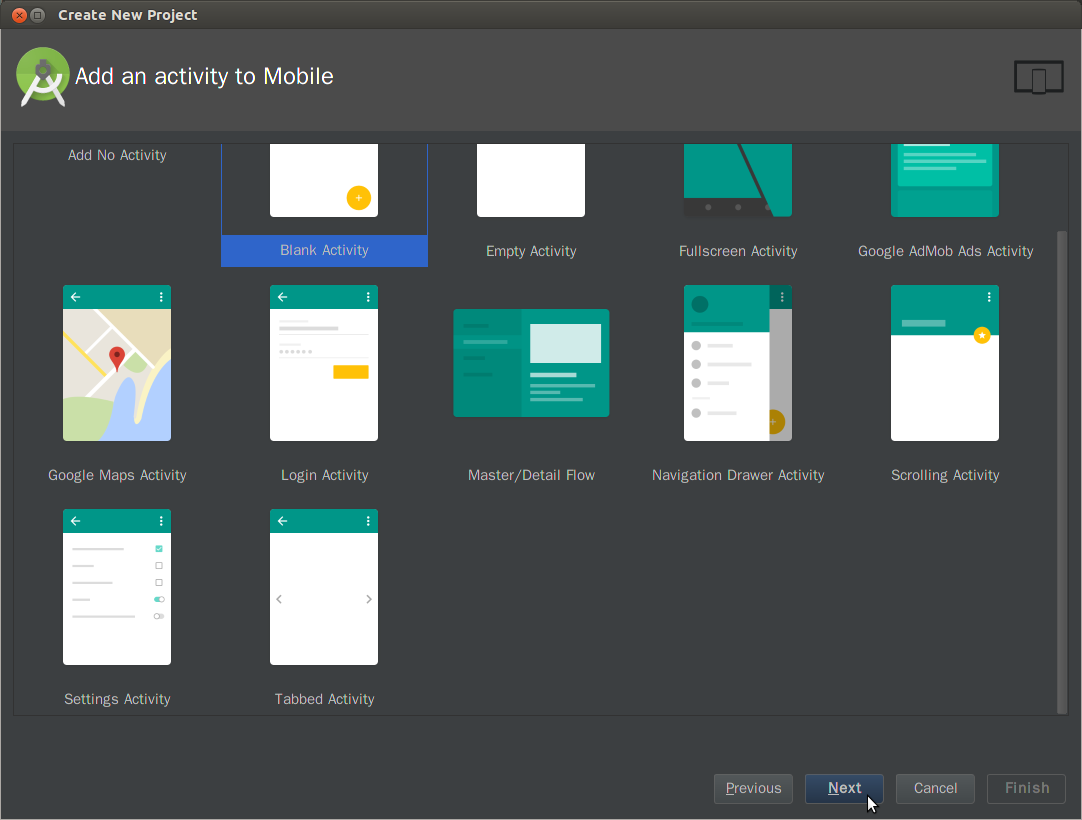
# 新建AS工程

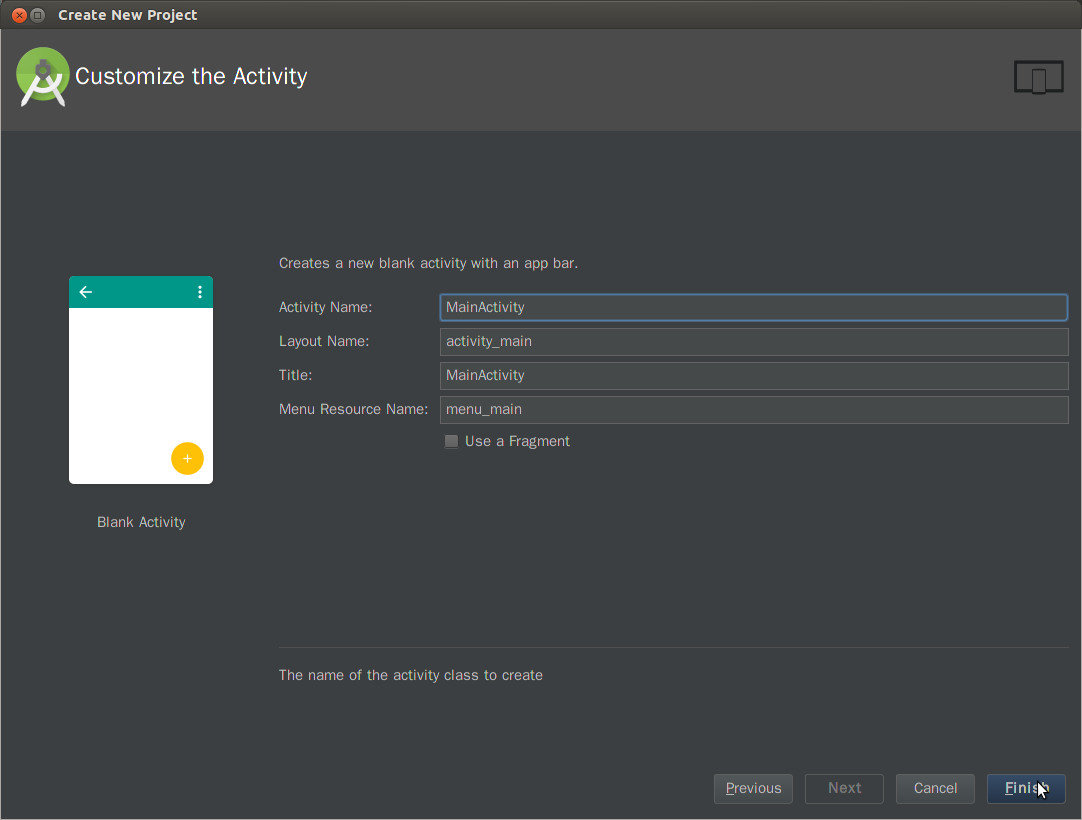
AS的工程类似于Eclipse中的WorkSpace，Module类似于Eclipse中的Project。







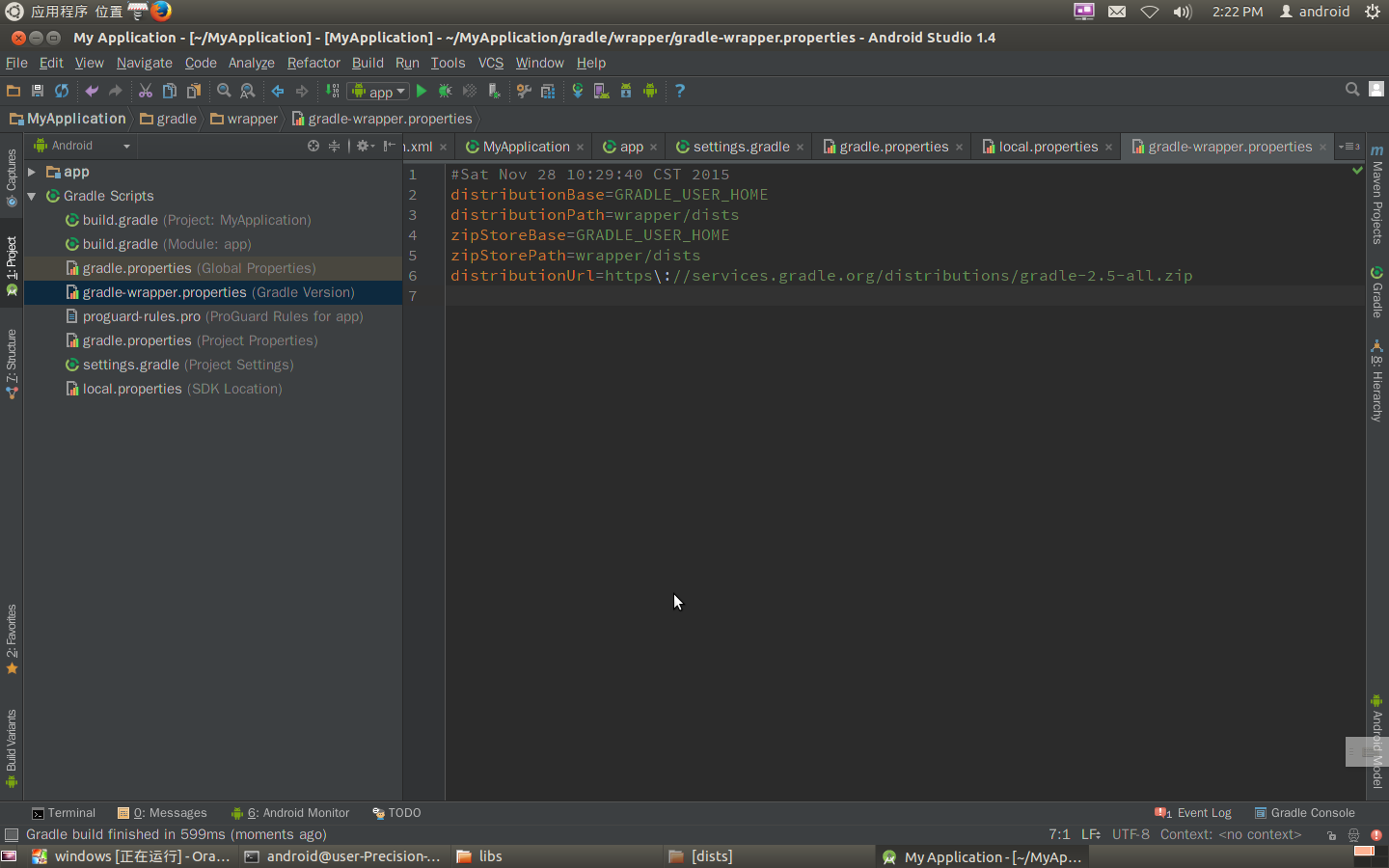




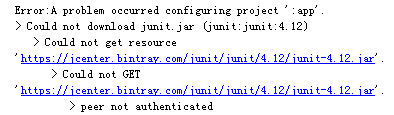
如果是第一次安装Android Studio，会出现“Building “\*\*\*” Grade project info”窗口。是AS正在下载Grade（约50M），由于是国外网站，需要翻墙。因此会一直停在“Loading”状态。

解决办法就是下载离线gradle-\*.\*-all包，并将其添加到/~/.gradle/wrapper/dists目录下，然后重新启动AS。建议下载最新版本的包，兼容性会更好。

图中distributionUrl=https\://services.gradle.org/distributions/gradle-2.5-all.zip这行表示如果我们的工程中没有gradle，AS会根据这个url去下载gradle，这就是为什么第一次打开AS时速度非常慢的原因。



好不容易配置完了，新建了个工程竟然是有错误

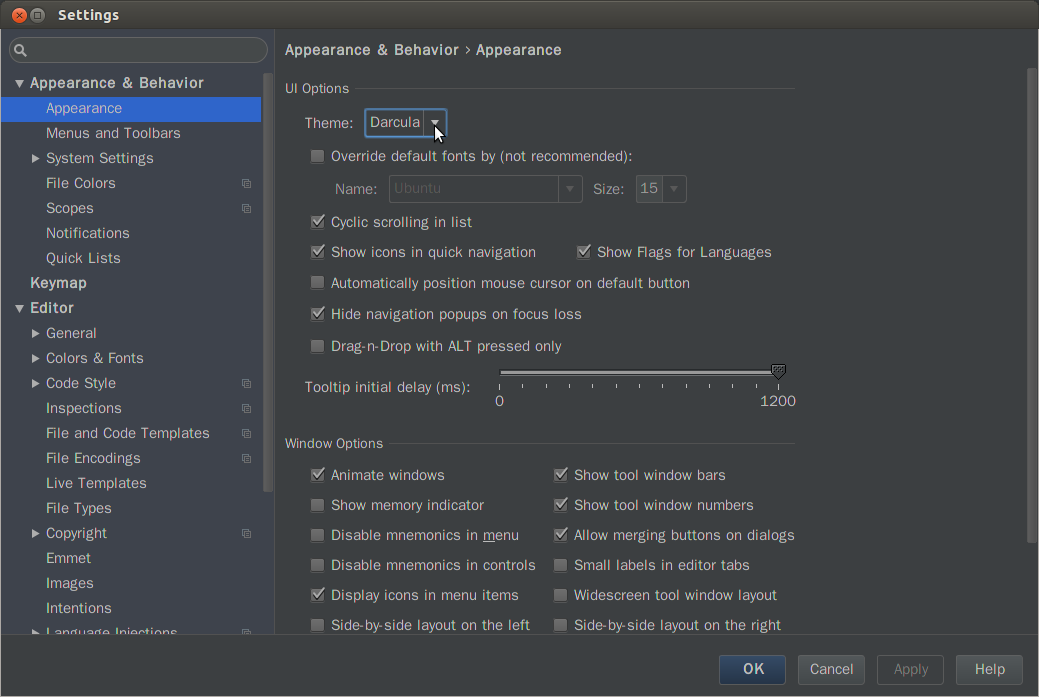


这可能是因为网络问题，在Gradle文件夹下，找到build.gradle(module app), 将里边的dependences { }中的测试依赖项去掉即可。

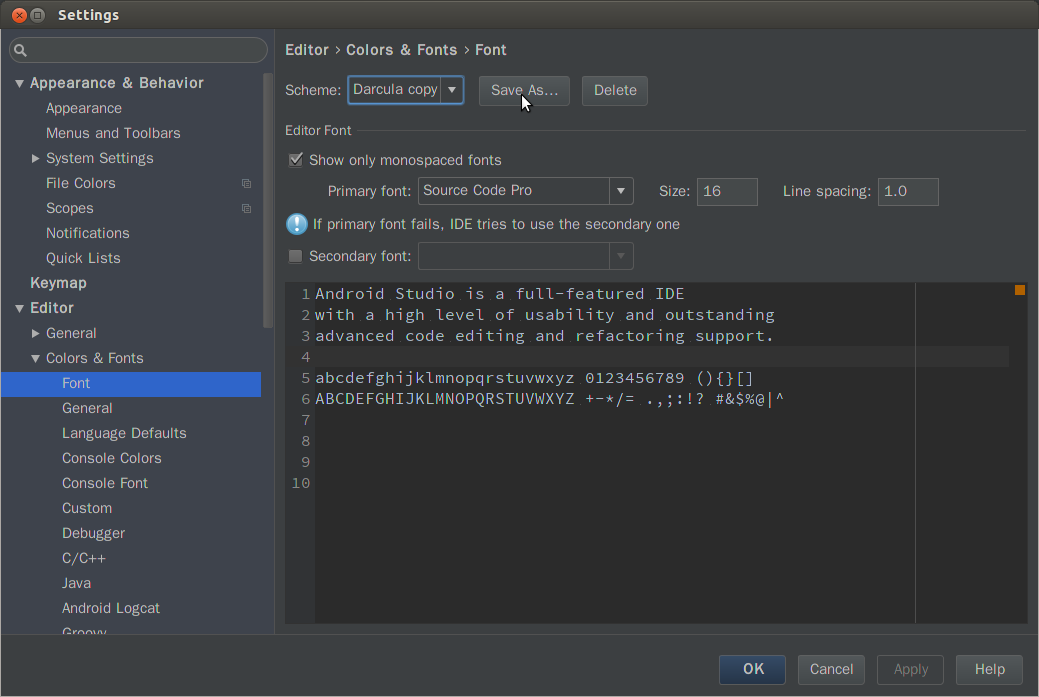
# 设置

打开AS设置对话框，快捷键为Ctrl+Alt+S，不过这个快捷键与Ubuntu系统冲突，可以通过File>Settings打开。

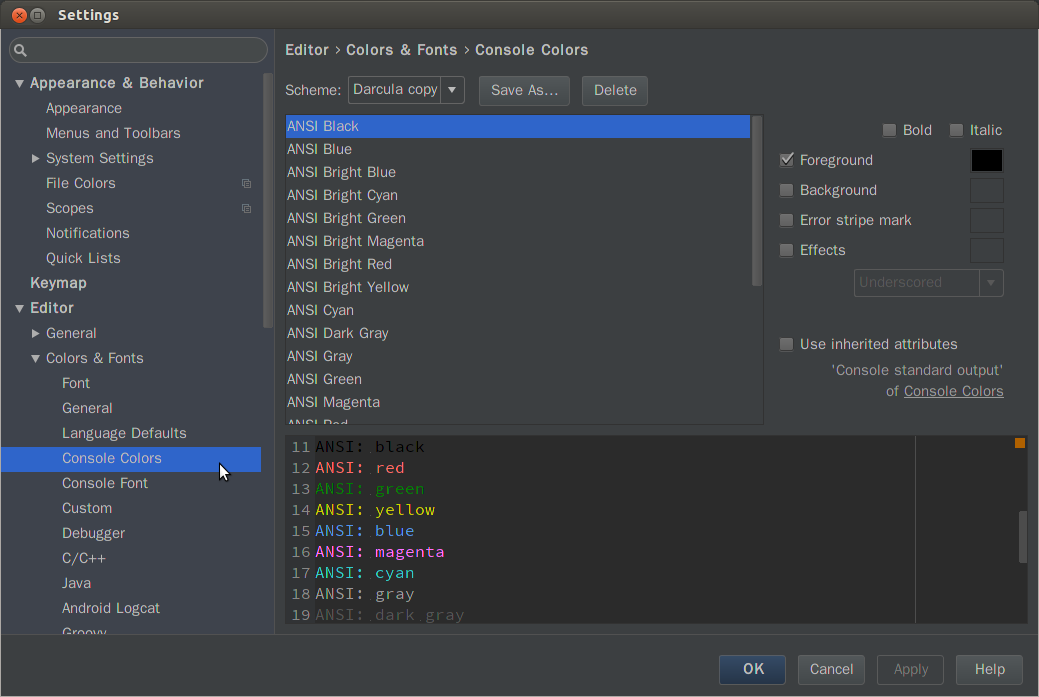
修改主题：



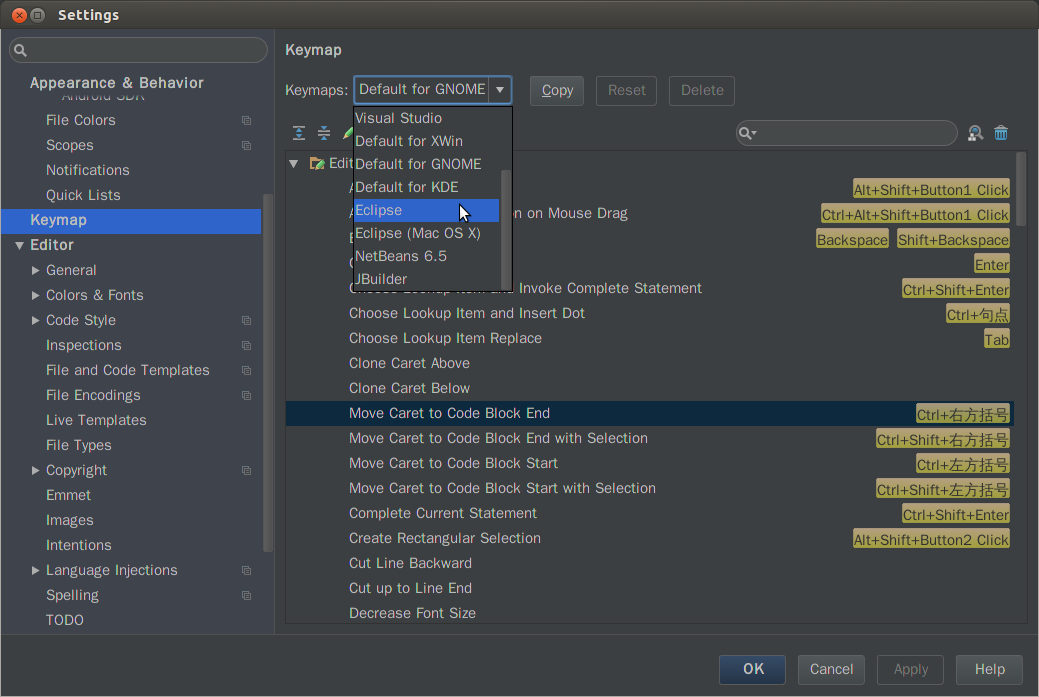
修改字体：



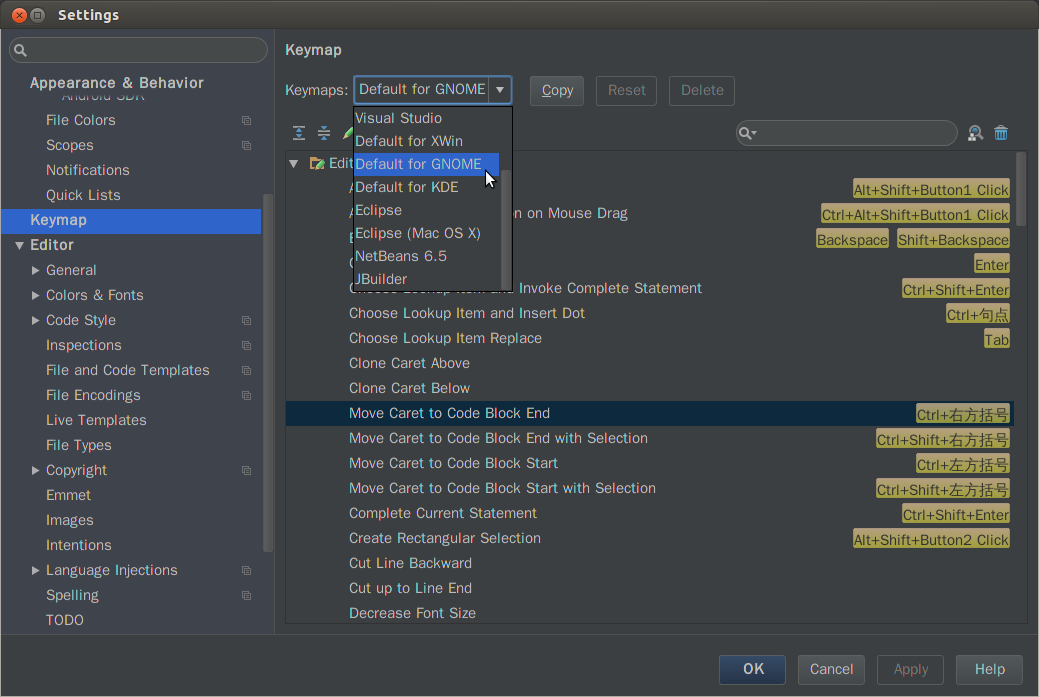
默认的Scheme是不可更改的，点击“Save As”重命名。同样修改“Console Font”字体。



快捷键修改：

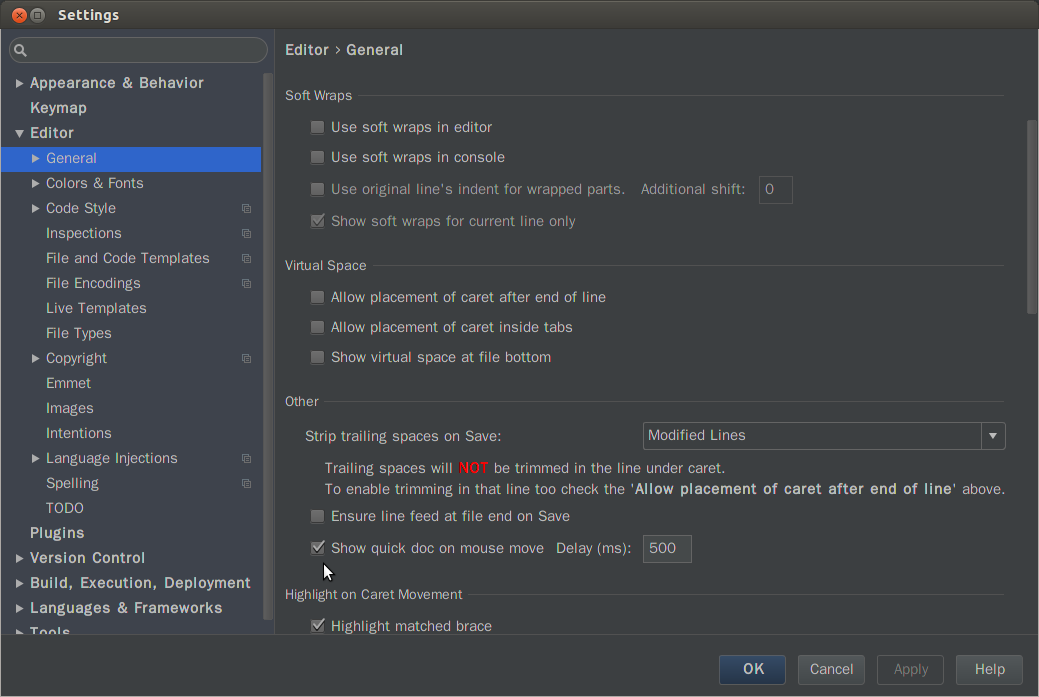


习惯Eclipse快捷键的童鞋可以选择此项。当然，还是建议使用AS默认的快捷键，习惯之后发现比Eclipse的快捷键强大很多。



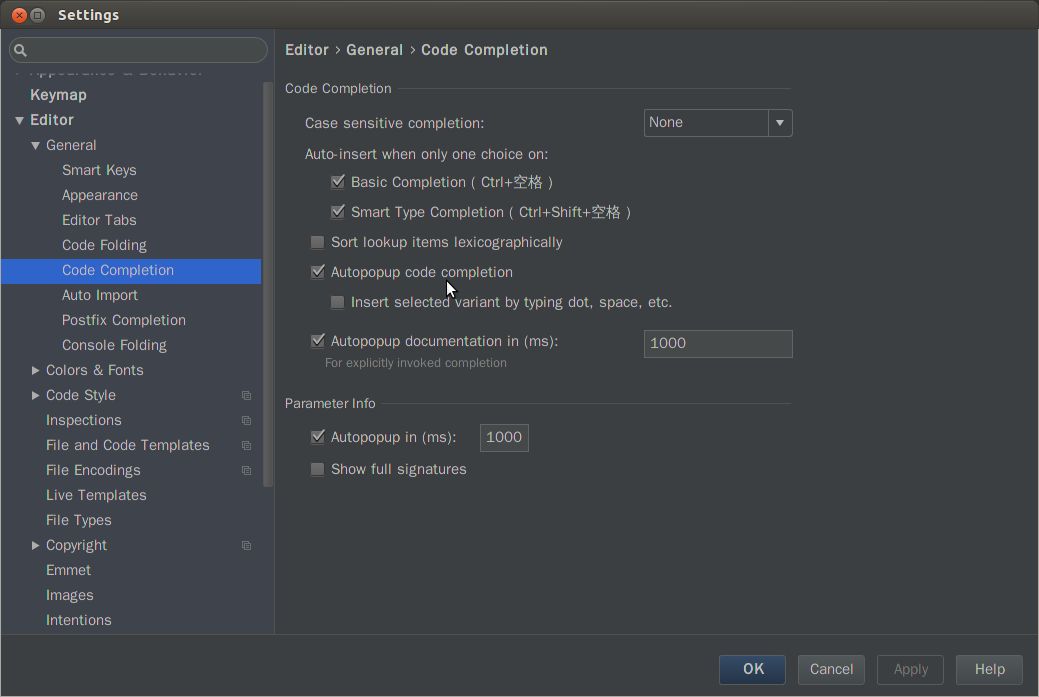
添加API文档悬浮显示：

AS默许是要通过快捷键Ctrl+Q来悬浮显示API文档的，如果要像Eclipse那样鼠标悬停处显示API文档，设置如下。

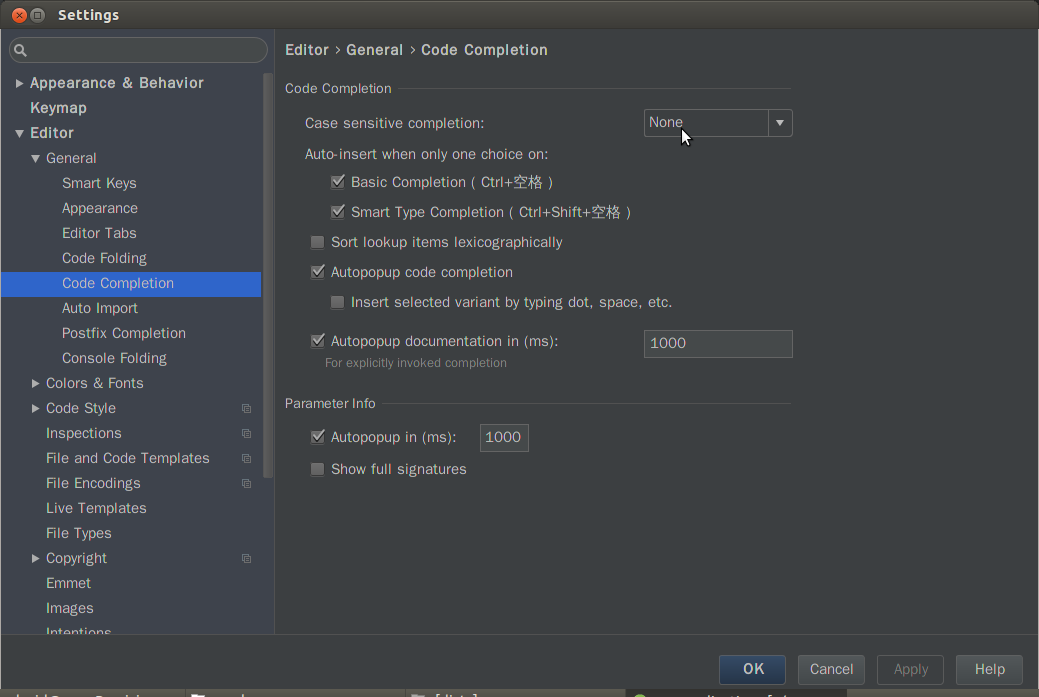


配置代码自动提示：

AS中默认是打开代码自动提示的，而且非常智能。你只要输入方法或字符串当中的某些字符，AS会自动给出最优的匹配。亲测非常实用。

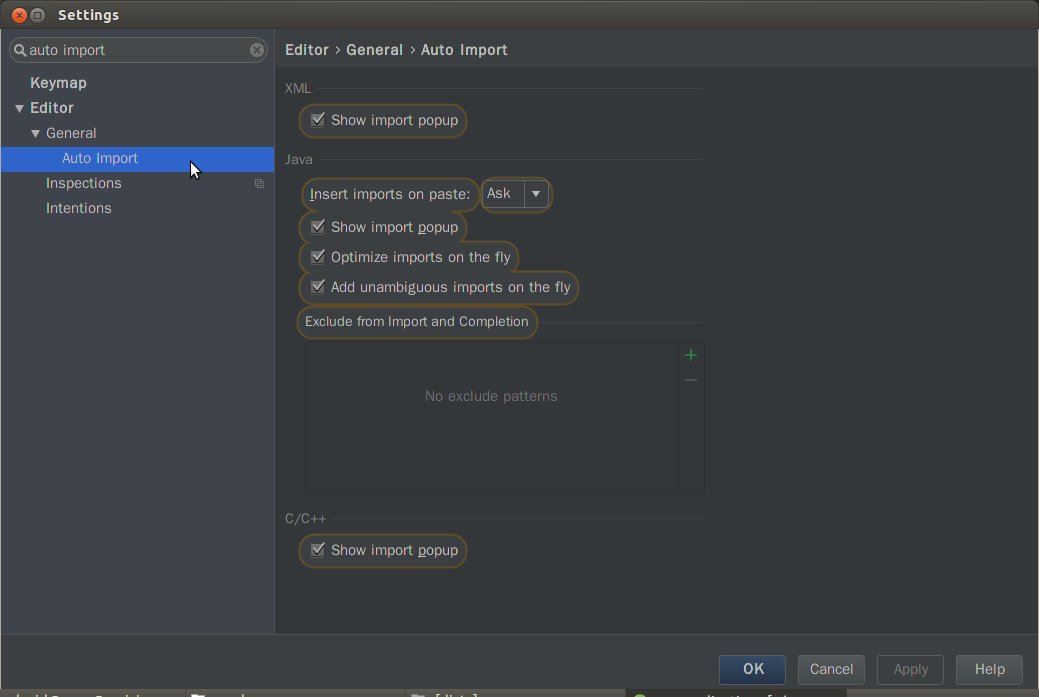


AS默认的代码提示是对大小写敏感的，如果想要AS对大小写不敏感，可以如下设置。

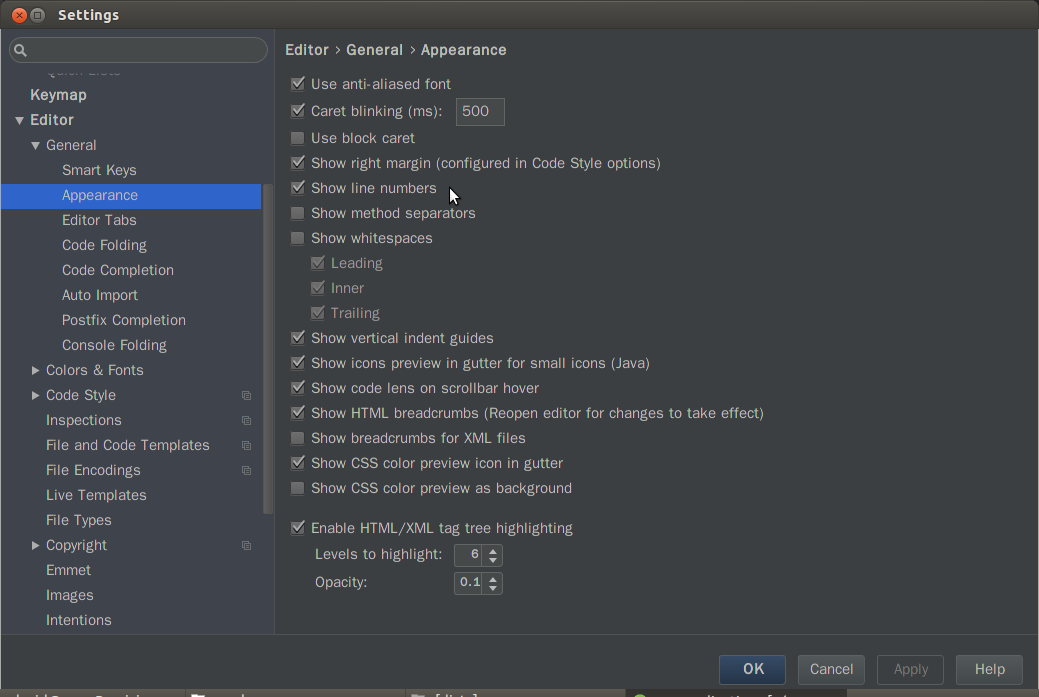


设置自动导包：

Eclipse中，我们要引用一些类时必须要导包。而AS可以自动导包和优化掉无用包。



显示代码行数：

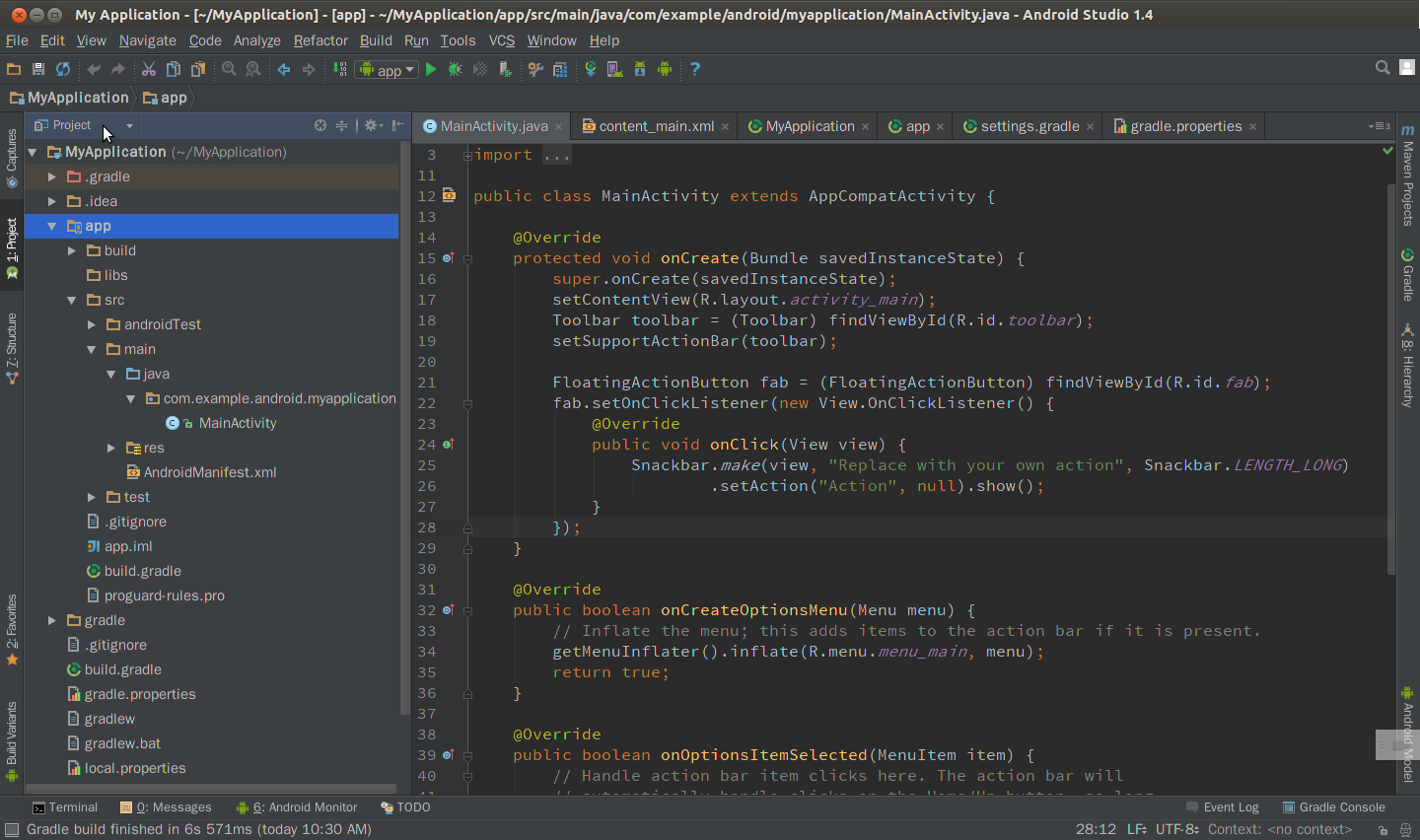


AS中的工程结构和Eclipse有所不同。

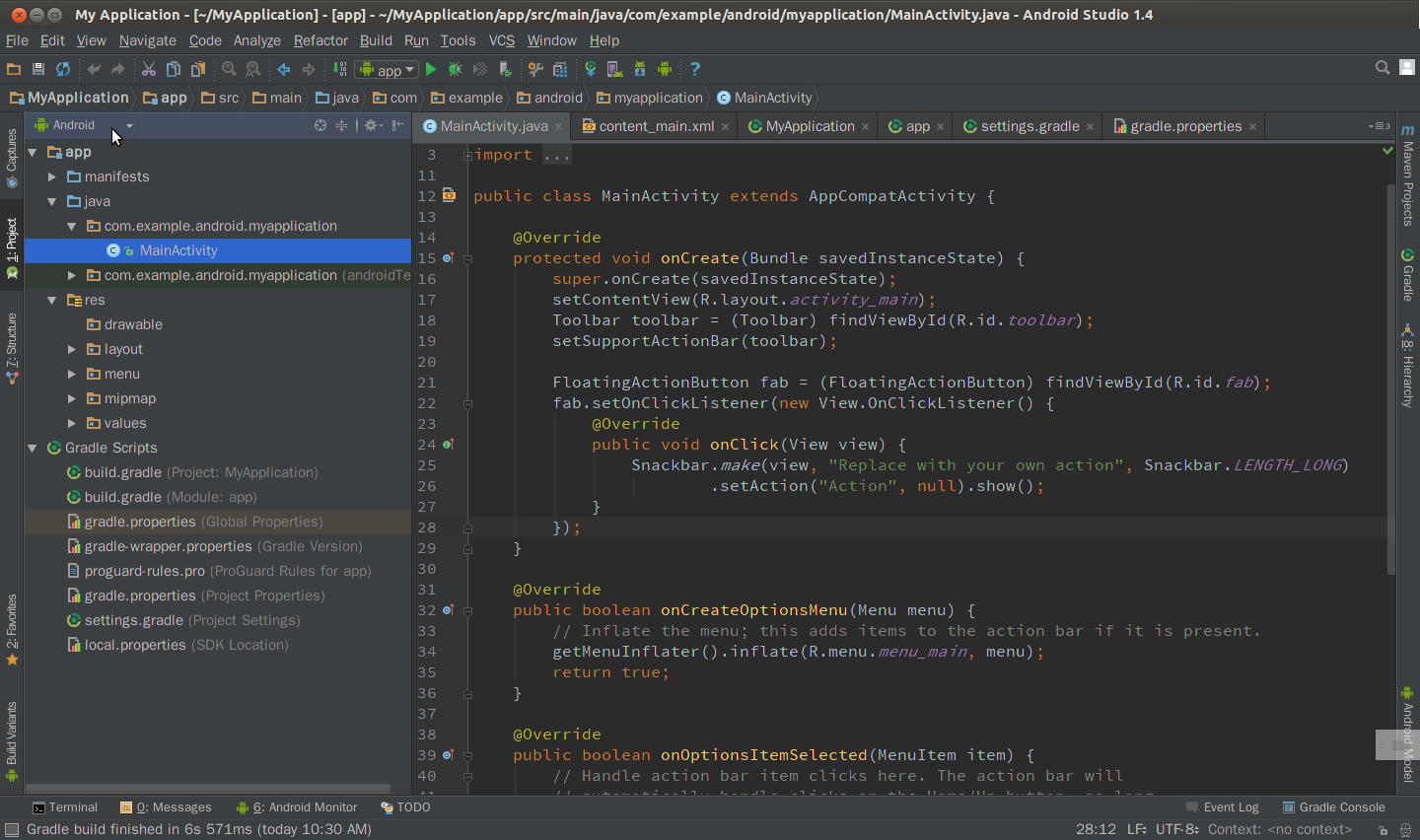
|  |  |  |
| --- | --- | --- |
|  | Eclipse | AS |
| 工作区 | Workspace | Project |
| 项目 | Project | Module |
| 引用 | Preference | Module dependency |

新建工程后默认会生成一个名为app的Module(类似Eclipse中的Project)。新建Module有两种方式：1、File-New-New Module；2、在左侧导航栏内选中Module按F4，打开Project Structure对话框，点击左上角绿色的“+”号。Module会有一个手机图标。

AS的显示方式：在左侧导航栏选择Project栏，再切换到Project显示方式，这种显示方式与Eclipse基本相同。



但是这种显示方式文件目录都堆叠在一起，显得杂乱没有重点，推荐使用Android显示方式。



Android显示方式分为两个部分：上面部分显示各个Module，下面部分显示Grade Scripts。

至此，AS的基本配置就差不多了，可以开始码代码了。但为了提高工作效率，掌握一些常用快捷键是很有必要的。

# 常用快捷键(http://blog.csdn.net/wuqilianga/article/details/51003867)

|  |  |
| --- | --- |
| Alt+Enter | 选择导入包；自动修正错误。 |
| Ctrl+N | 查找类 |
| Ctrl+Shift+N | 查找文件 |
| Ctrl+Shift+Alt+N | 查找方法 |
| 双击Shift | 全范围查找 |
| Ctrl+E | 最近打开文件 |
| Ctrl+P | 方法参数提示 |
| Ctrl+D | 复制行 |
| Ctrl+X/Y | 删除行 |
| Ctrl+H | 显示类结构图 |
| Ctrl+Q | 显示API文档 |
| Ctrl+O | 选择父类方法实现 |
| Alt+Up/Down | 在方法间快速移动 |
| F2 | 高亮错误或警告快捷定位 |
| F3 | 向下搜索 |
| Shift+F3 | 向上搜索 |
| Ctrl+Shift+F7 | 高亮关键字 |
| Ctrl+B/Click | 快速打开光标处类或方法 |
| Ctrl+Alt+B | 查看抽象方法的实现 |
| Ctrl＋[或] | 跳到大括号的开头结尾 |
| Ctrl＋Shift＋Backspace | 跳转到上次编辑的地方 |
| Shift+Enter | 在当前行下方添加一行，光标跳转到下一行。 |
| Ctrl+F12 | **调出当前文件的大纲** |
| Ctrl+Alt+h | 查看某个方法的调用路径。 |
| ctrl+shift+i | **快速查看定义** |
| shift+f6 | **重命名变量或者方法名。重构神技。** |
| Alt+Shift+Insert之后拖框选择 | **列编辑** |
| Ctrl+Shift+a输入Analyze Data Flow to Here | 分析某个值的来源 |
| Ctrl+Shift+a输入Analyze StackTrace | 分析堆栈信息（当出现FATAL EXCEPTION时，可以点击报错信息进行查找） |
| Ctrl + Shift + “+”/” | 折叠/展开代码块 |
| Ctrl + Shift + Insert | 选择性插入 |
| Alt + Q | 上下文信息 |
| Ctrl + U | 定位到父类 |
| Ctrl + 点击Tab | 在外部打开文件 |
| Alt + Ｊ | 识别当前选中字符串，选择下一个同样的字符串，同时进行操作。 |
| Ctrl + Shift + Enter | 语句补全 |
| Ctrl + Alt + M/V/C/P  Menu → Refactor → Extract → | 提取方法/变量/常量/参数 |
| Ctrl + Shift + J | 合并行和文本 |
| Ctrl + Shift + Up/Down | 上下移动行  上下移动方法 |
| Ctrl + Alt + T | 包裹代码  跟Ubuntu快捷键冲突  Menu → Code → |
| Ctrl + Shift + Delete | 移除包裹 |

工具栏的常用功能：

在AS的工具栏有几个常用的按钮，分别是Gradle同步、AVD Manager、SDK Manager、DDMS。

Gradle同步：在你项目运行或者更改Gradle配置的时候都要点击下这个按钮，会下载相应的依赖。

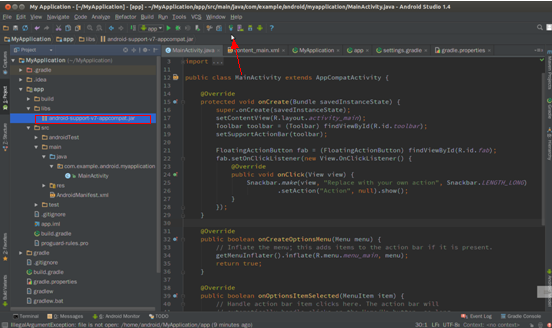
AVD Manager：模拟器管理

SDK Manager：SDK管理

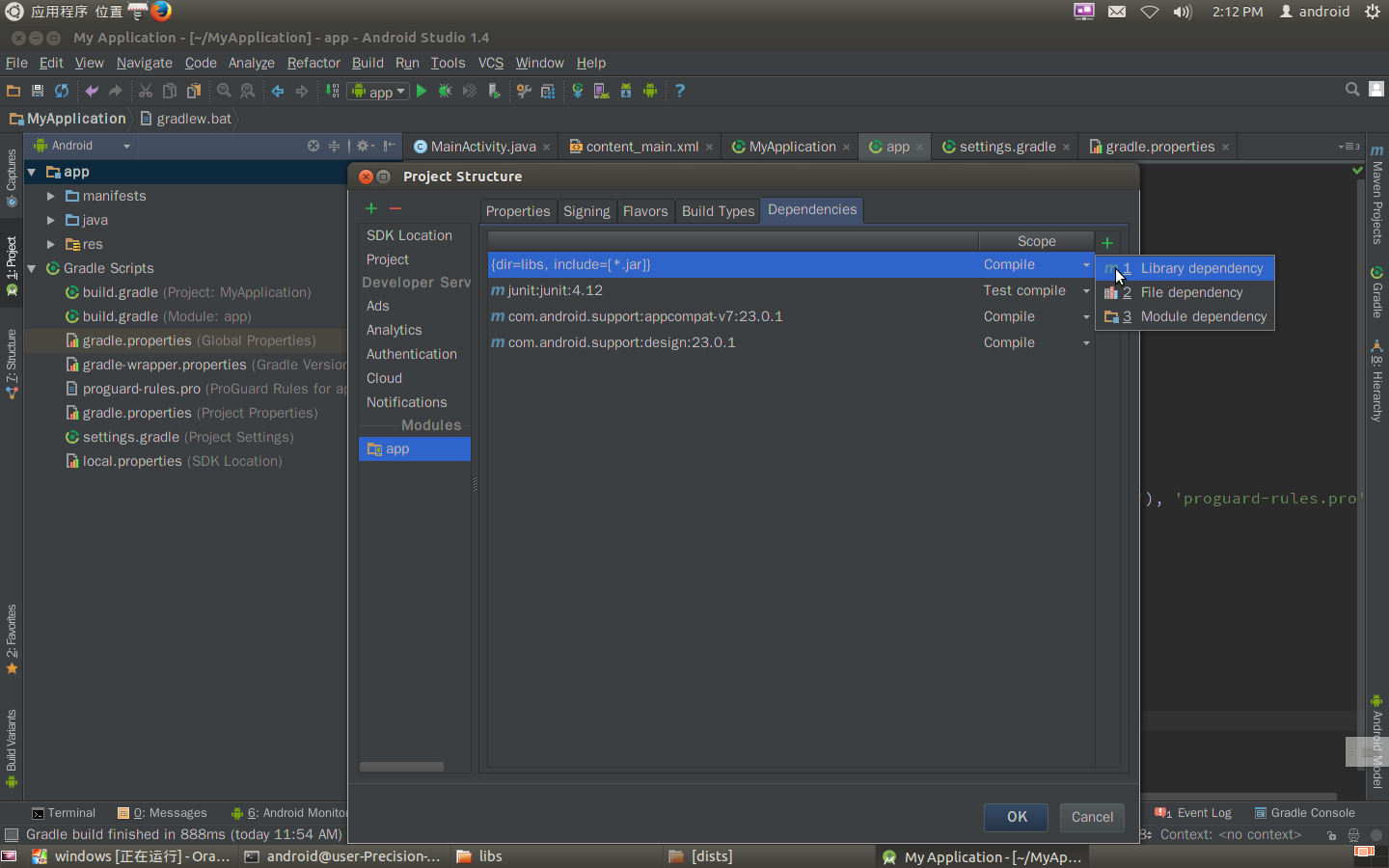
DDMS：即Dalvik Debug Monitor Service，Dalvik调试监控服务。

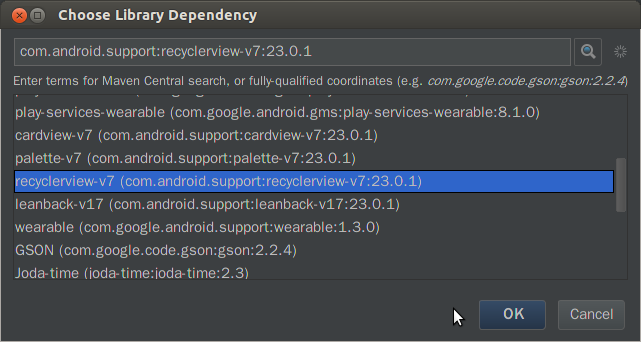
# 导入第三方jar包：

方法一：将jar文件拷贝到Project视图下工程文件的libs目录下，导入之后，选中jar包，然后同步。



方法二：打开Project Structure。





# [使用Android Studio导入源码](http://blog.csdn.net/yanbober/article/details/48846331)

在开始编译idegen模块前，需要先全局编译出out目录及相关文件，然后我们通过如下命令编译idegen模块：

mmm development/tools/idegen/

这行命令的意思是编译idegen这个模块项目，然后生成idegen.jar文件。运行完以后如果看到如下信息则说明编译OK：

......

#### make completed successfully (7 seconds) ####

接着执行如下脚本：

development/tools/idegen/idegen.sh

这行命令的意思是在根目录生成对应的android.ipr、android.iml IEDA工程配置文件。等待片刻得到类似如下信息说明OK：

Read excludes: 21ms

Traversed tree: 194799ms

通过如上操作我们就完成了基本的源码配置工作。

大家都知道使用Eclipse导入源码很慢，Android Studio导入源码时也慢，所以建议修改android.iml文件（将自己不用的代码去掉），然后再导入Studio。

就像下面摘取的android.iml文件1887行开始的这些一样：

<sourceFolder url="file://$MODULE\_DIR$/./sdk/testapps/userLibTest/src" isTestSource="true"/>

<sourceFolder url="file://$MODULE\_DIR$/./tools/external/fat32lib/src/main/java" isTestSource="false"/>

<excludeFolder url="file://$MODULE\_DIR$/out/eclipse"/>

<excludeFolder url="file://$MODULE\_DIR$/.repo"/>

<excludeFolder url="file://$MODULE\_DIR$/external/bluetooth"/>

<excludeFolder url="file://$MODULE\_DIR$/external/chromium"/>

<excludeFolder url="file://$MODULE\_DIR$/external/icu4c"/>

<excludeFolder url="file://$MODULE\_DIR$/external/webkit"/>

<excludeFolder url="file://$MODULE\_DIR$/frameworks/base/docs"/>

<excludeFolder url="file://$MODULE\_DIR$/out/host"/>

<excludeFolder url="file://$MODULE\_DIR$/out/target/common/docs"/>

<excludeFolder url="file://$MODULE\_DIR$/out/target/common/obj/JAVA\_LIBRARIES/android\_stubs\_current\_intermediates"/>

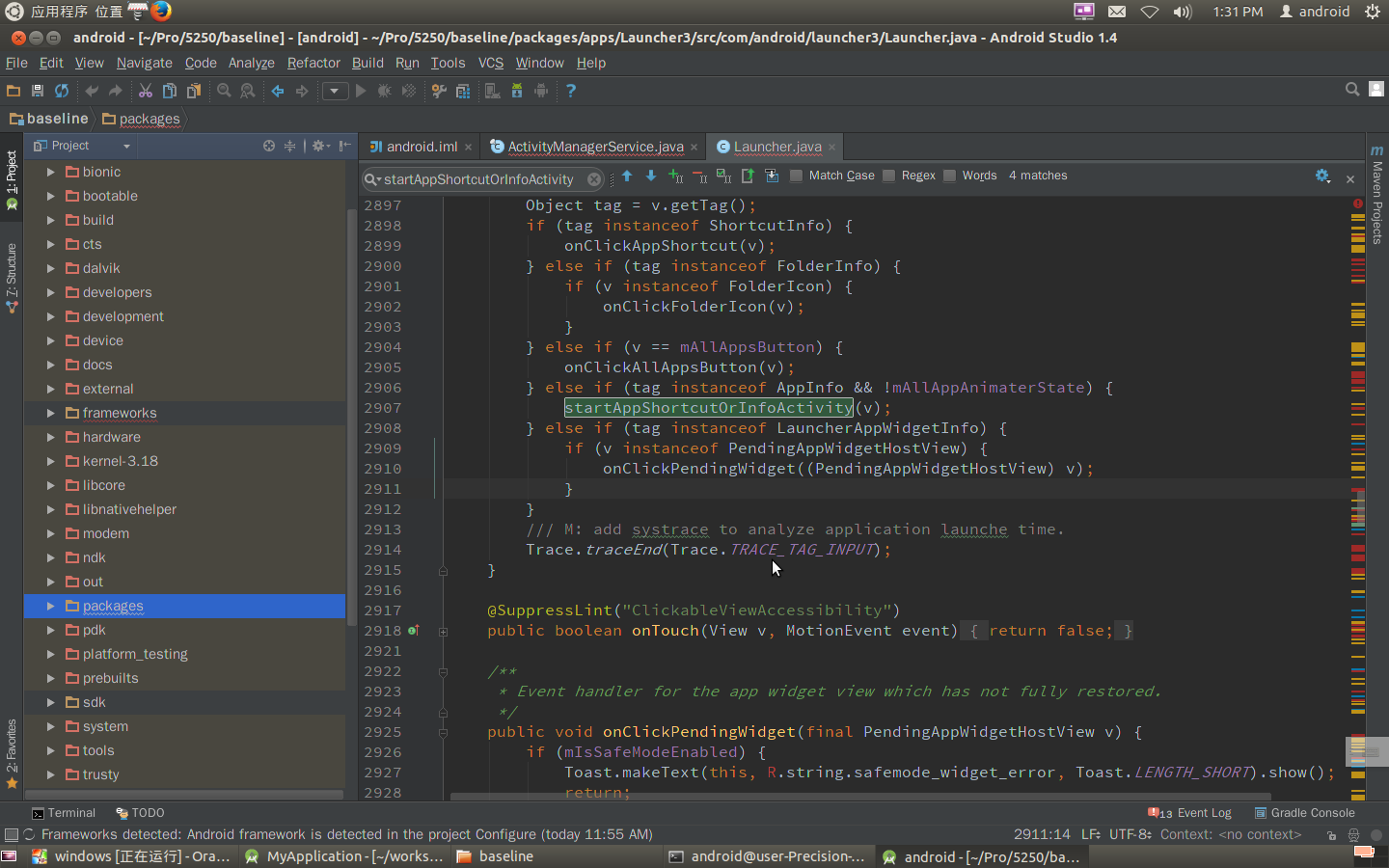
<excludeFolder url="file://$MODULE\_DIR$/out/target/product"/>

<excludeFolder url="file://$MODULE\_DIR$/prebuilt"/>

我们可以仿照上面这段代码的<excludeFolder url="file://$MODULE\_DIR$/.repo"/>写法一样过滤掉不需要的内容，这样在导入时就会快很多。

也可以通过Android Studio的Project Stucture 打开左侧Modules，然后将右侧Sources中一些目录Excluded掉。

有了如上的这些操作以后，我们打开Android Studio，然后选择打开一个现有的Android Studio项目，然后选择打开源码根目录下的android.ipr文件。



首次导入会出现很多”cannot resolve method”错误。选择File—Project Structure—modules—Dependences，将Dependences全部删除即可。

# Gradle

每个Module都有一个build.gradle文件，它是对应Module的配置文件。build.gradle中有一段代码：

dependencies {

compile fileTree(dir: 'libs', include: ['\*.jar'])

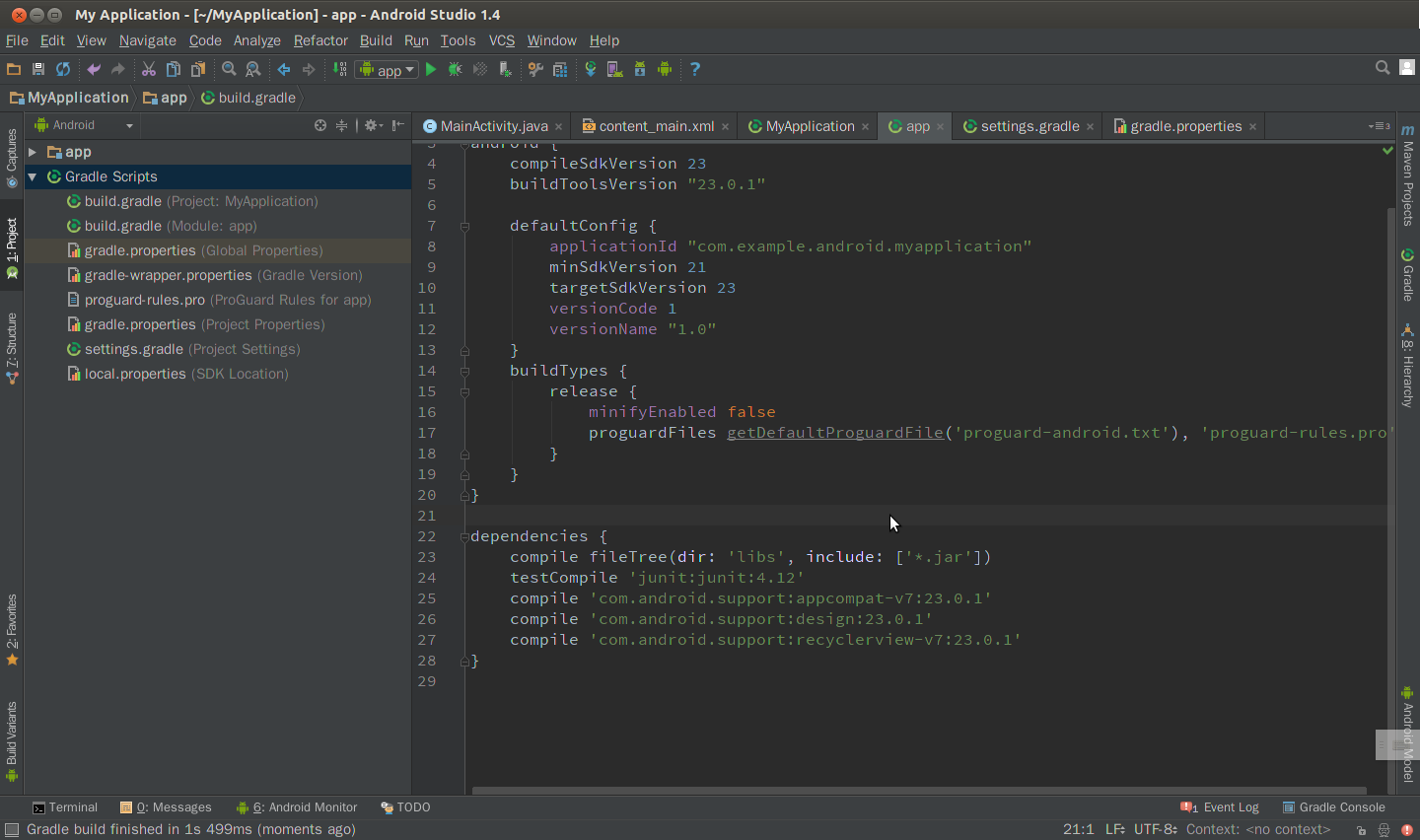
testCompile 'junit:junit:4.12'

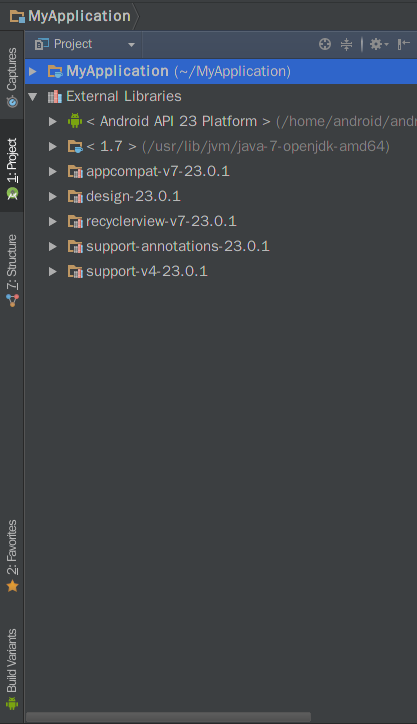
compile 'com.android.support:appcompat-v7:23.0.1'

compile 'com.android.support:design:23.0.1'

}

表示该Module引用了哪些包。在Project Structure中添加Library Dependencies后，在Module的build.gradle中会多一行代码，表示引入了recycleview-v7这个包。此时，recycleview-v7这个jar包不再是出现在libs这个文件夹下了，而是出现然最下方的External Libraries中。





通过Gradle导入的jar包都是从<http://bintry.com/bintray/jcenter>这个中央仓库上download下来的。

Gradle导入jar包更加方便，不需要自己去官方下载。而且如果官方将jar包更新了，我们只需要在build.gradle中改一下版本号就行了。不过需要有网络才能构建Gradle。

# 常见错误

1. The current **inotify**(7) watch limit is too low. [More detail](https://confluence.jetbrains.com/display/IDEADEV/Inotify+Watches+Limit)

主要有三个步

1.在 /etc/sysctl.conf文件中增加一行   
fs.inotify.max\_user\_watches = 524288

2.让改变生效   
sudo sysctl -p

3.重启IDE

参考资料：

第一次使用Android Studio时你应该知道的一切配置。

[**Android Studio入门到精通**http://blog.csdn.net/yanbober/article/details/45306483](http://blog.csdn.net/yanbober/article/details/45306483)

[**使用Android Studio导入源码**http://blog.csdn.net/yanbober/article/details/48846331](http://blog.csdn.net/yanbober/article/details/48846331)

倍数提高工作效率的 Android Studio 奇技http://android.jobbole.com/81687/