

# Android Studio入门指南v0.1

作者：毕小鹏

日期：2013.8.5

博客：<http://blog.csdn.net/wirelessqa>

# 1. Android Studio简介

《Android Studio入门指南》作者：毕小鹏 博客：<http://blog.csdn.net/wirelessqa>

Android Studio是一个基于IntelliJIDEA的新的Android开发环境。与Eclipse ADT插件相似，AndroidStudio提供了集成的Android开发工具用于开发和调试。除了你期望的IntelliJ的能力，AndroidStudio还提供：

- 基于Gradle构建的支持。
- Android专属的重构和快速修复。
- 捕获性能、易用性、版本兼容性问题。
- 支持ProGuard和应用签名。
- 基于模板的向导来创建常见的Android设计和组件。
- 丰富的布局编辑器，允许你拖放UI组件并在多个屏幕上的预览布局，等等。



## 2. 安装Android Studio

1. 从上面下载AndroidStudio安装包。（下载地址：<http://developer.android.com/sdk/installing/studio.html>译者注）
2. 安装Android Studio和SDK工具：

### Windows系统：

1. 启动下载的EXE文件，`android-studio-bundle-<version>.exe`。
2. 按照安装向导安装AndroidStudio。

已知问题： 在某些Windows系统中，没有找到Java的安装启动程序脚本。如果你遇到这样的问题，你需要设置一个环境变量，指向正确的位置。

《Android Studio入门指南》作者：毕小鹏 博客：<http://blog.csdn.net/wirelessqa>

选择“开始”菜单]]> 计算机]]> 系统属性]]> 高级系统属性。然后打开“高级”选项卡]]> 环境变量，并添加新的系统变量JAVA\_HOME，指向你的JDK文件夹，例如C: \ Program Files\JAVA\jdk1.7.0\_21。

## Mac OS X的：

3. 打开下载的DMG文件，`android-studio-bundle-<version>.dmg`。

4. 拖放AndroidStudio到“应用程序”文件夹中。

已知问题： 根据您的安全设置，当你试图打开Android Studio时，你可能会看到一个警告说，安装包损坏、应将其移至垃圾桶。如果发生这种情况，进入系统偏好设置]] ]]> 安全与隐私，在允许下载的应用程序，选择随时随地。然后再次打开AndroidStudio。

## Linux操作系统：

5. 为您的应用程序，解压下载的Tar文件，`android-studio-bundle-<version>.tgz`到一个合适的位置。

6. 要启动Android Studio，在终端定位到`android-studio/bin/`目录，并执行`studio.sh`。

您可能需要添加`android-studio/bin/`到你的PATH环境变量，这样就可以从任何目录启动Android Studio。

这就可以了！你已经准备好开始用Android Studio开发应用程序了。

注：在Windows和Mac，Android Studio个人工具和其他的SDK包保存应用程序的目录内。要直接访问这些工具，使用终端导航到应用程序并定位到`sdk/`目录。例如：

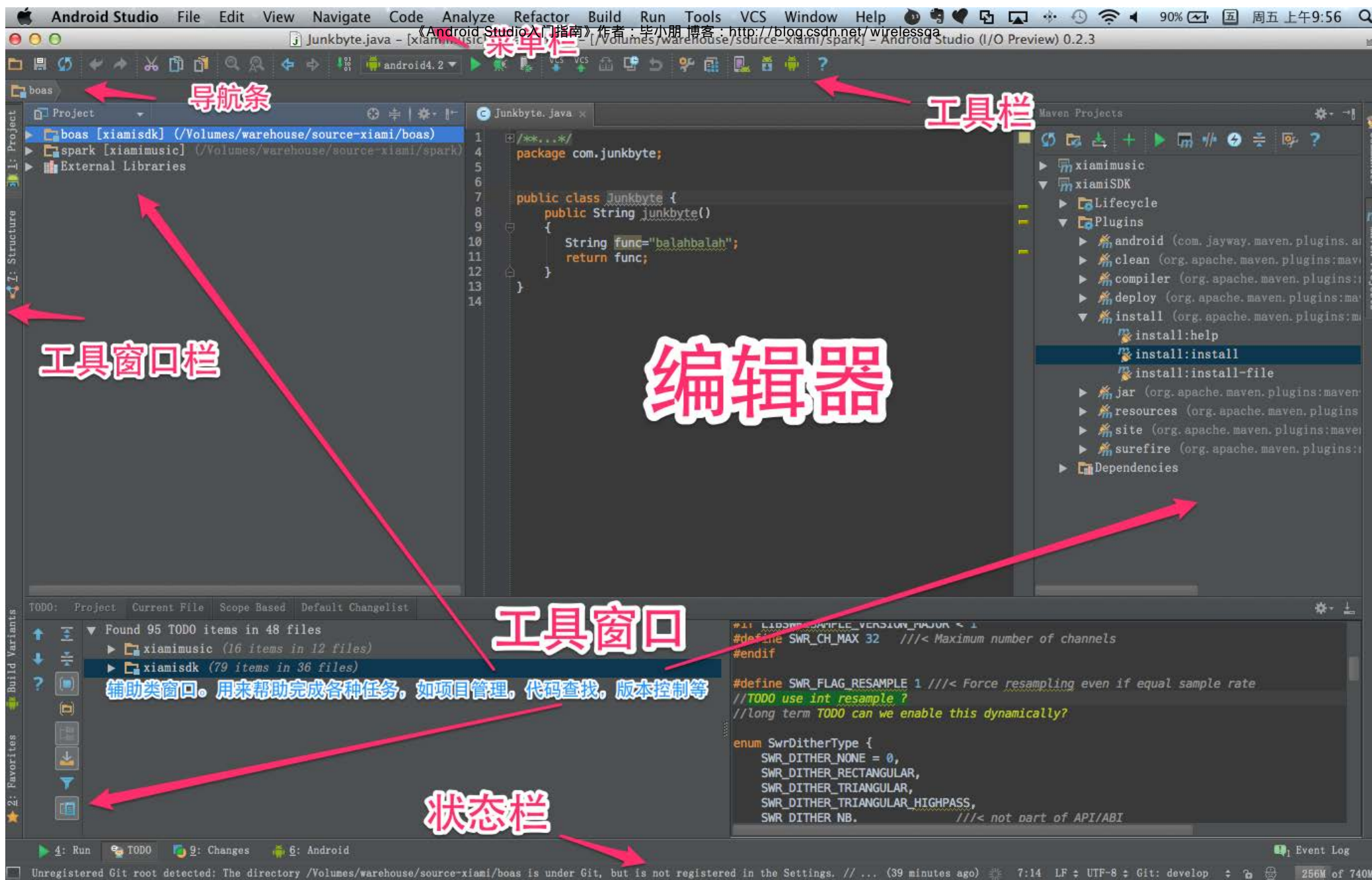
Windows: `\Users\<user>\AppData\Local\Android\android-studio\sdk\`

Mac: `/Applications/Android\Studio.app/sdk/`

对于一些已知的问题，查看[tools.android.com/knownissues](http://tools.android.com/knownissues)。

## 3. Android Studio界面元素

### 3.1 界面简介

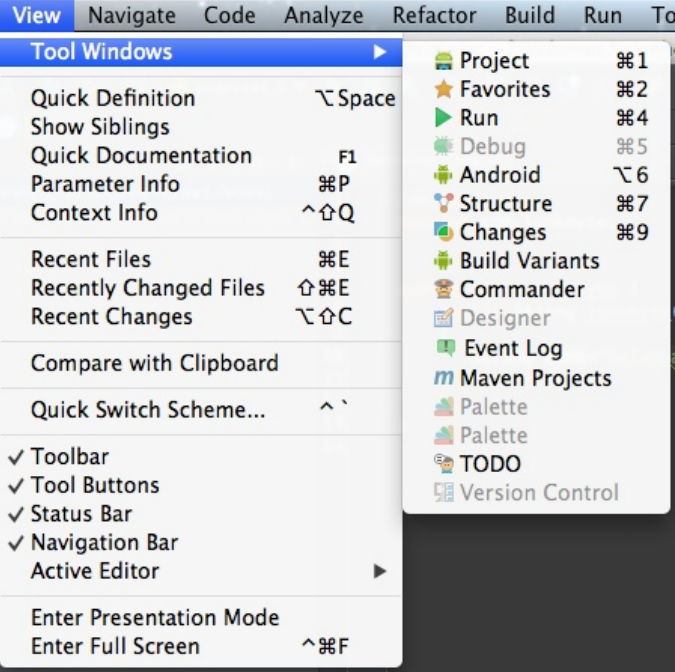


1. 菜单和工具栏: 你懂的
2. 导航条: 编辑文件时帮助定位和导航项目中的文件
3. 状态栏: 显示当前项目, Android Studio 本身的状态, 还有别的一些状态相关的一些信息

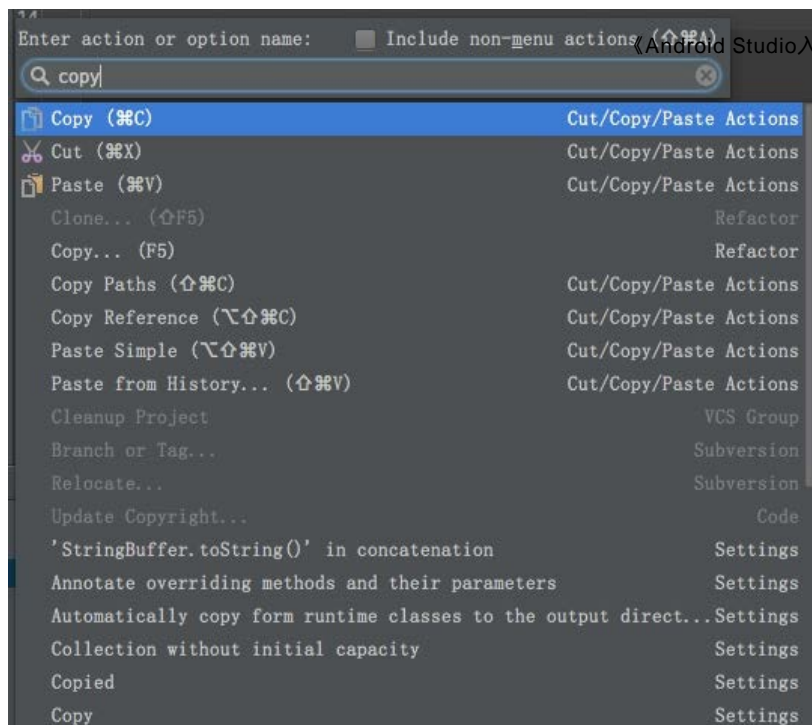
- 4.编辑器： 你懂的
- 5.工具窗口： 辅助类窗口。Android Studio提供了各式各样的辅助窗口来帮助完成各种任务，如项目管理，代码查找，版本控制等，类似eclipse中VIEW的概念
- 6.工具窗口栏:位于IDEA界面的上下左右各有一个，摆放着工具窗口显示和隐藏的按键

### 3.2 关于菜单和工具条

菜单和工具栏包含了Android Studio中的各种有用的命令，如：（1）打开或者隐藏Android Studio中的界面元素，可以使用菜单栏中的VIEW选项



（2）当你想执行一个命令，但忘记去哪里找这个命令的时候，Android Studio提供了一个快捷的方式进行定位，按 command+Shift+A 可以触发一个命令提示界面，输入你想执行命令的前缀，就会列出跟这个前缀相关的一些命令，选择你需要的执行



### 3.3 关于导航条

显示导航条有两种途径:

- (1) 通过VIEW菜单, 选择 Navigation Bar 选项
- (2) 按 Alt+Home. (用该方式打开, 导航条显示为浮动形式(windows))



关闭导航条也有两种途径

- (1) 通过VIEW菜单, 取消选择 Navigation Bar 选项
- (2) 当导航条是浮动状态时候, 按ESC键(windows)

### 3.4 关于状态栏



单击该图标可以控制“工具窗口栏”的显示和隐藏

《Android Studio入门指南》作者：毕小鹏 博客：<http://blog.csdn.net/wirelessqa>

当前在菜单栏或者工具栏或者右键上下文菜单中选择的命令描述和说明

单击该图标将触发后台任务管理器

显示项目的分支

当前光标的位置

AndroidStudio内存堆的使用情况，点击触发垃圾回收

Unregistered Git root detected: The directory /Volumes/w... (today 上午9:16) 14:1 LF UTF-8 Git: develop 194M of 740M

行分隔符

标示当前语法的高亮级别，有Inspections, Syntax, None，可以通过点击该图标进行更改

标示当前文件只读还是可写

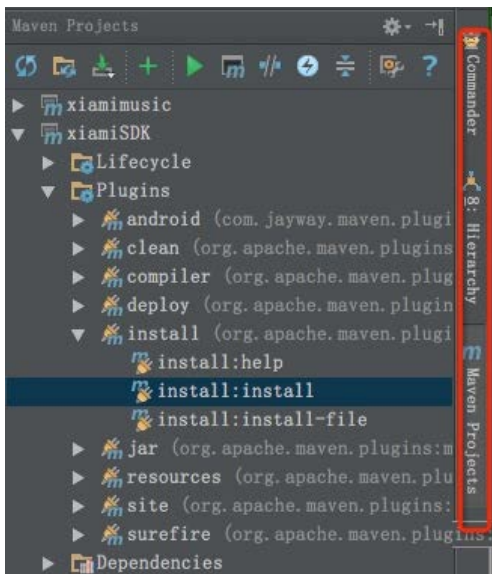
当前文件的编码，若当前文件没有自动选择编码，则可以通过它进行编码的选择

### 3.5 关于工具窗口

每个工具窗口提供了各种不同的视角和编程任务，一些工具窗口总是可见的，一些工具窗口则必须当相关的插件或者facets被激活的时候才是可见的。控制工具窗口显示或者隐藏的按键分布在工具窗口条上，显示和隐藏状态通过颜色区分：



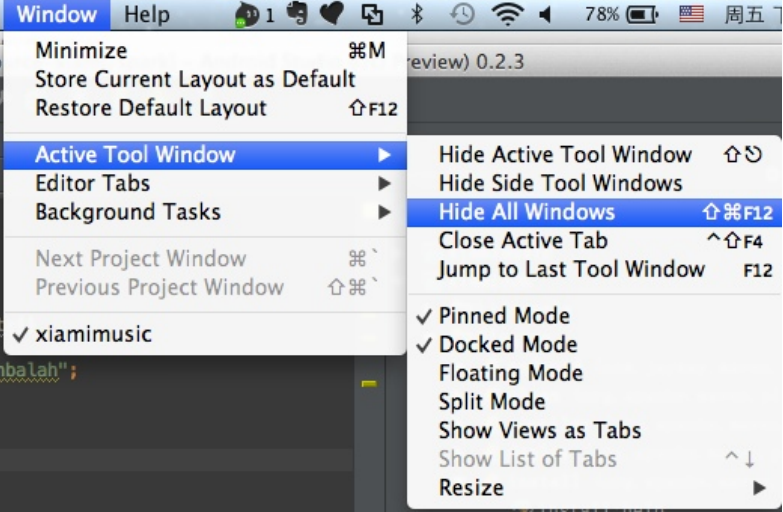
工具窗口的显示与隐藏是基于工具窗口的摆放组的，每个工具窗口条分为2个组（水平的工具窗口条的左右两边，垂直窗口条的上下两边，可以通过拖坠工具窗口按键来进行摆放组的移动），每个分组只有一个显示的工具窗口，当选择组中的另一个窗口时，会自动关闭该组原先的窗口



### 3.6 操纵工具窗口

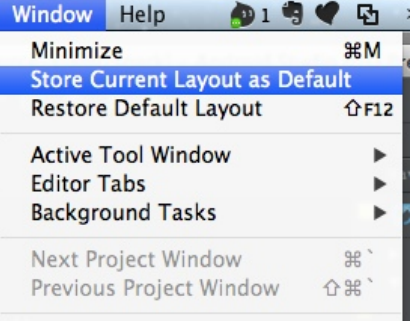
#### 3.6.1 关闭所有的工具窗口

- 1.通过菜单栏中的 Window - Active Tool Window - Hide All Windows
2. command+Shift+F12.
- 《Android Studio入门指南》作者：毕小鹏 博客：<http://blog.csdn.net/wirelessqa>



3.6.2 保存当前的窗口布局：

菜单栏 Window - Store Current Layout as Default  
若需要重置窗口布局选择菜单栏 Window - Restore Default Layout



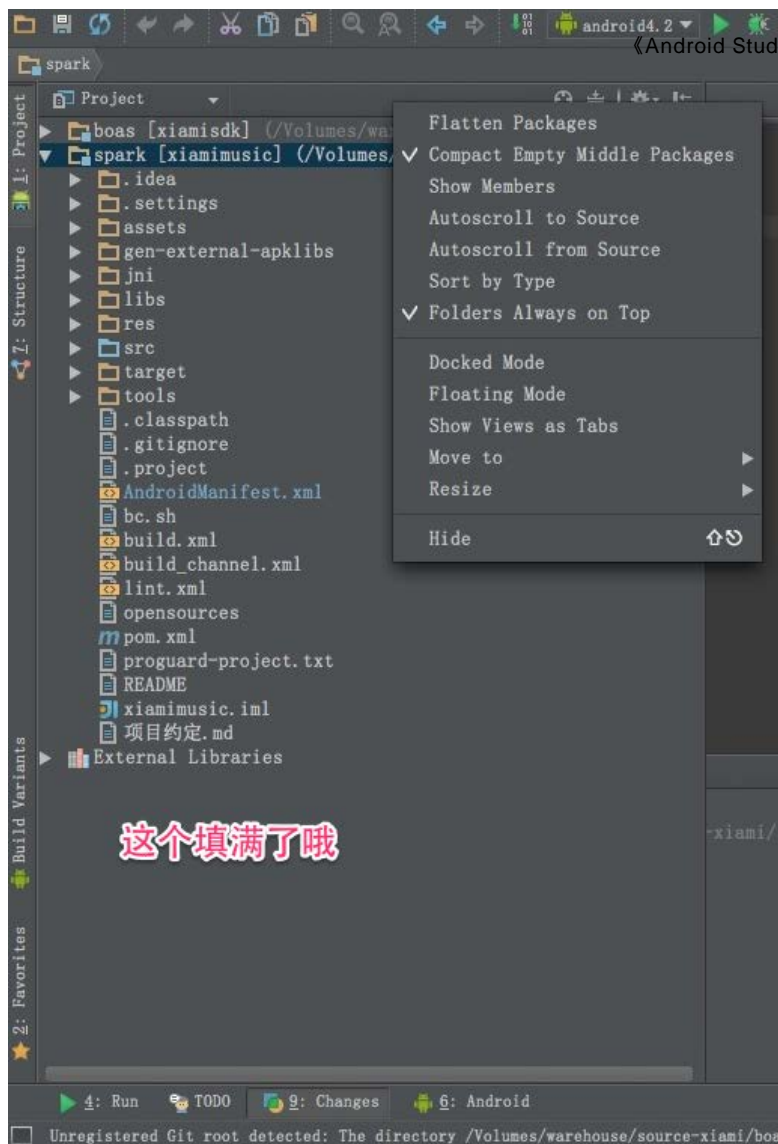
3.6.3 窗口工具的查看模式（Viewing Modes）

工具窗口提供了不同的查看模式，用于控制工具窗口的显示形式和表现方式，帮助使用者快速定位到工具窗口，或者最大化编辑区域

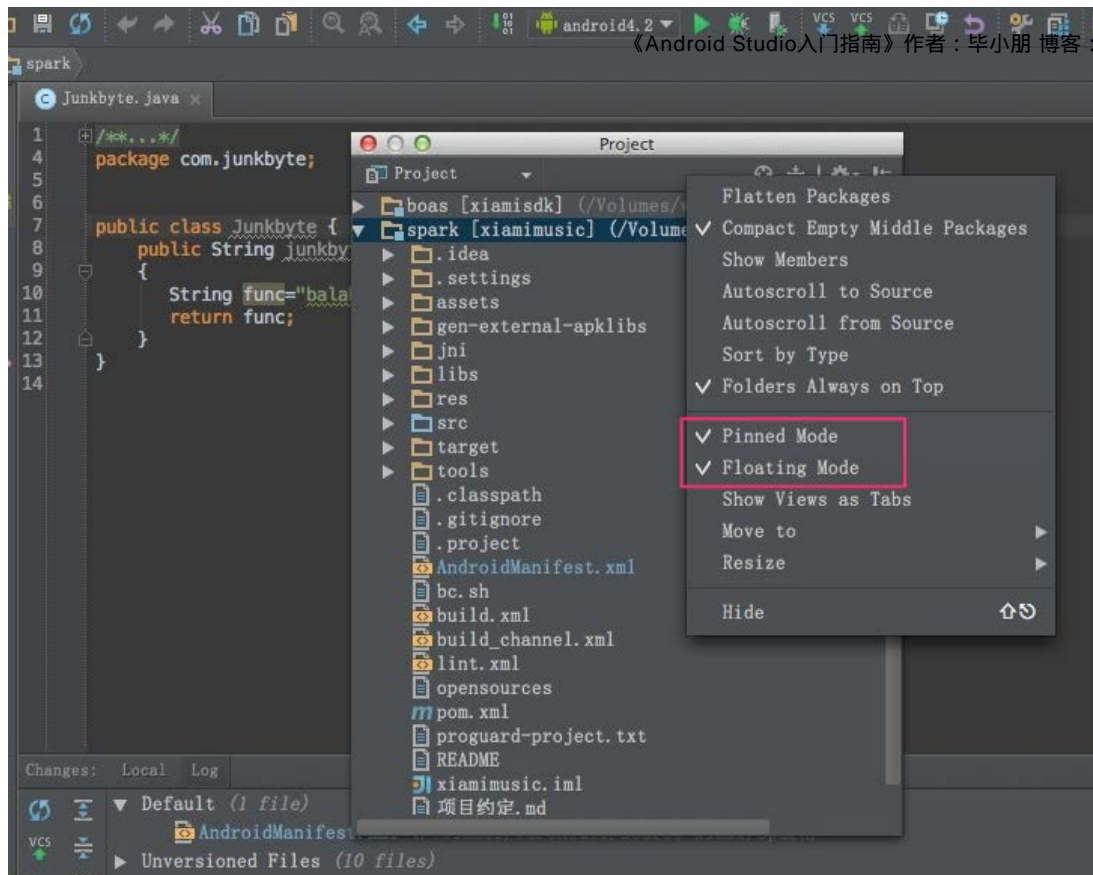
1.固定和浮动模式 Floating Mode

当窗口时固定模式时，若同时窗口的模式还是非停泊模式（docked mode未选中）则窗口会占满它所附着的工具窗口条的长度或者高度（取决于工具窗口条是水平的还是垂直的）

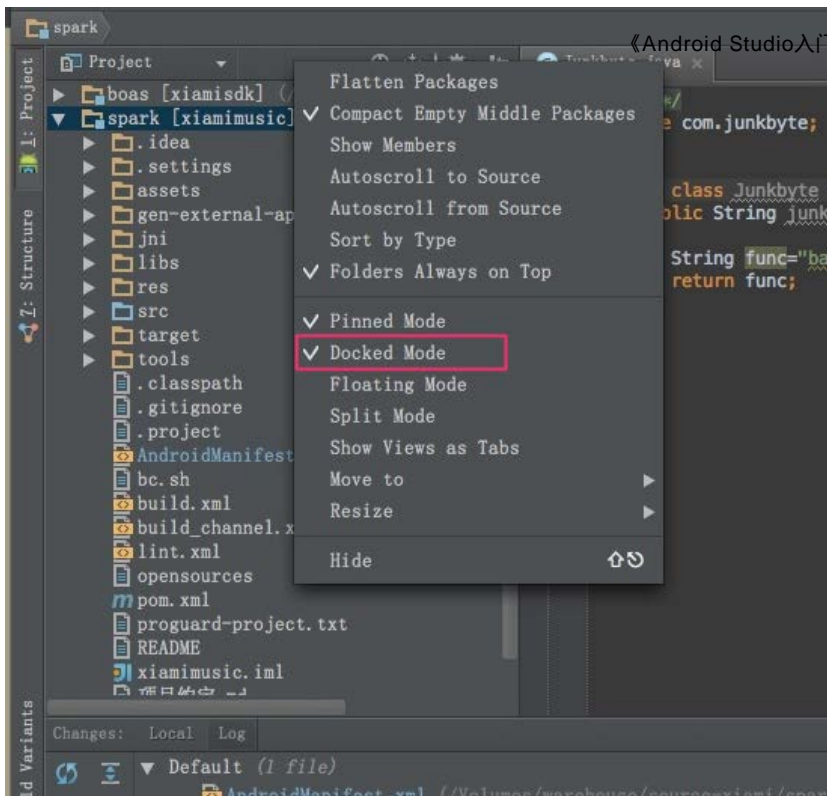




当窗口时浮动模式(**Floating**)时，它可以放在屏幕上的任何位置，当这个窗口处于不活动状态时，若同时窗口时寄托模式（pinned.mode）它会变成半透明状态（透明度取决于Appearance settings中的透明度设置）



## 2. 停靠和脱开模式 Docked Mode



该模式基于窗口时**固定模式(Docked Mode)**才起作用

在停靠模式，跟这个窗口相邻的界面元素将环绕着该窗口，调整该窗口的大小会自适应调整相邻窗口的大小

在脱开模式时。该窗口会变成最上层的界面元素，会盖住其他的元素与其相交的部分（除开工具窗口，无法盖住其他的工具窗口，不会存在相交部分，当与其他工具窗口相邻时，无法调整该相邻的边界），重设脱开模式的窗口大小不会自适应的调整其他元素的大小。

当一个脱开模式的窗口变为不活动状态时，它会自动隐藏。

### 3.寄托与非寄托模式 **Pinned Mode**

该模式决定了工具窗口变成不活跃状态时是隐藏还是可见。

Pinned 模式下是可见，unpinned 模式下将会自动隐藏。当然根据别的模式设定有一些例外情况：

Undocked 模式的窗口不活跃时，总是被隐藏

Floating 模式的窗口不活跃时，将变成半透明

### 4.分离模式 **Split Mode**

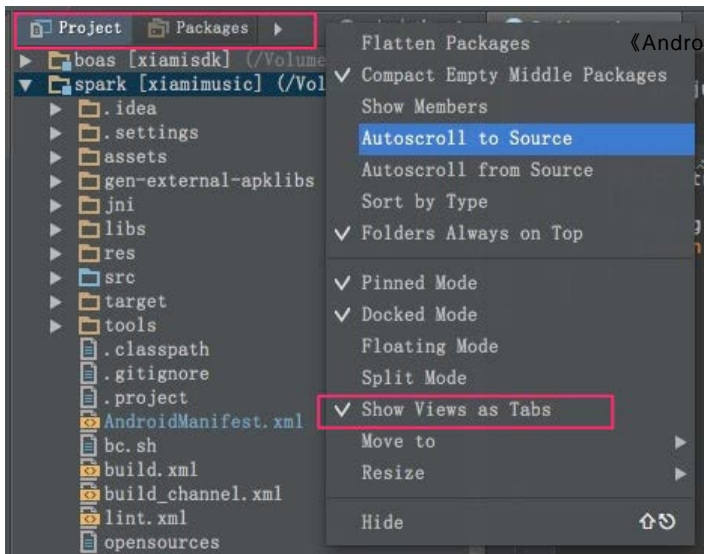
该模式决定了同一个工具窗口条中有多少个停靠模式（docked）的窗口可以同时显示（1还是2）

同时工具窗口依据该模式将依附其上的窗口进行分组，分离模式的为一组，非分离模式的为另一组，任何一个时候每个分组中只有一个窗口是可见的

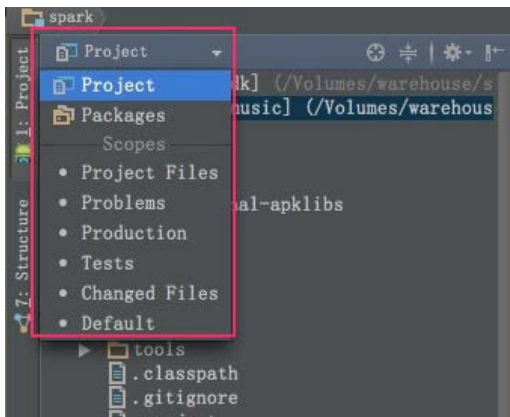
### 5. 作为标签显示 **Show View as Tabs**

该模式用于控制有多个内容层需要展示的窗口以何种形式的方式来选择内容层

Tabbed 模式 使用标签页形式

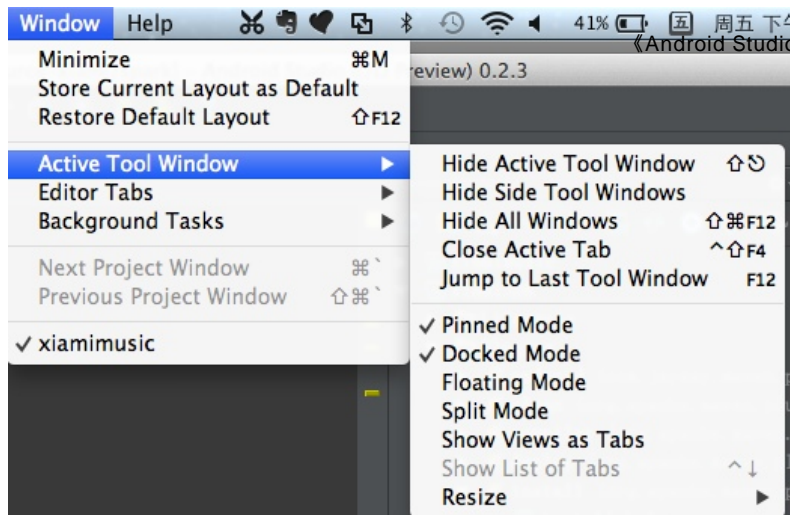


非标签页形式，则使用下拉列表的方式进行选择：



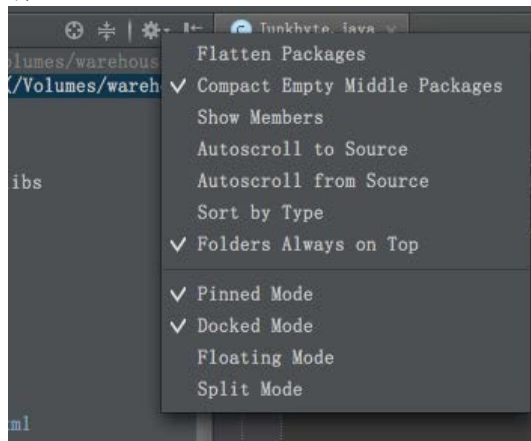
### 3.6.4 开启和关闭模式的方法

开启和关闭该模式可以通过窗口的上下的上下文菜单（右键菜单）或者主菜单栏中 Window - Active Tool Window来转换



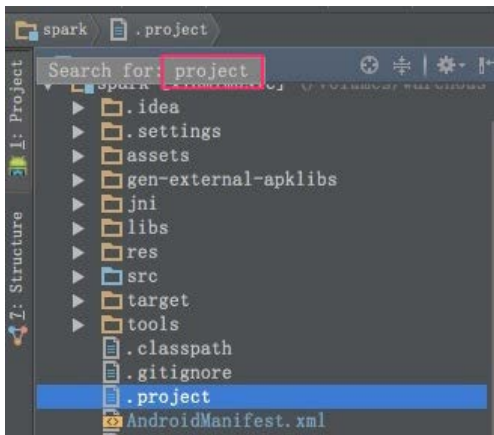
《Android Studio入门指南》作者：毕小鹏 博客：<http://blog.csdn.net/wirelessqa>

或



### 3.6.5 在工具窗口的快速搜索

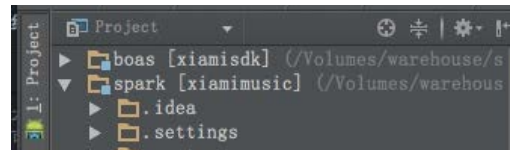
- 1.选择所需的工具窗口
- 2.输入你需要查找内容（文件，类，字段等）的名字，当你输入时，搜索域将出现在该窗口上，并显示你输入的字符
- 3.输入完毕时，按ENTER，窗口上符合要求的结果将被显示和选择. 按ESC隐藏搜索域



## 4. Android Studio基本操作

### 4.1 在编辑器中打开文件

打开Project内的文件进行编辑的几种方式:



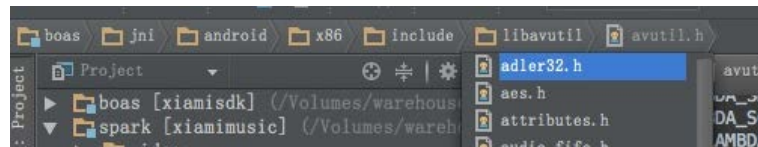
1. 在Project窗口中双击需要编辑的文件。
2. 在Project窗口选择需要编辑的文件，按F4
3. 在Project窗口选择需要编辑的文件，然后右键弹出上下文菜单，选择Jump to Source



4. 使用Go To

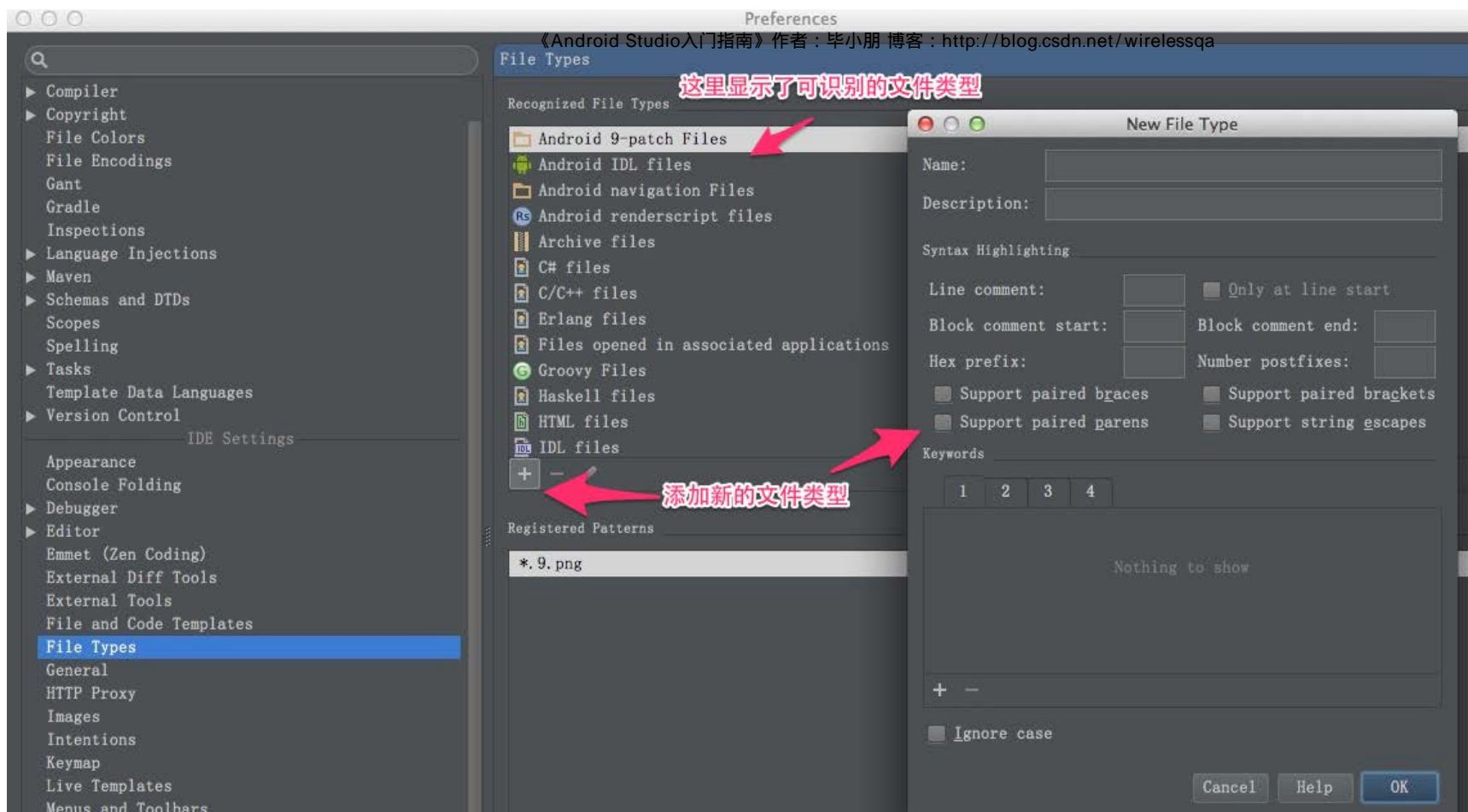


5. 通过导航条，来选择打开文件：



备注：如果文件类型已经注册，则直接使用编辑器打开，（可以在 Settings 中的 File Types查看已经注册的文件类型）。





如果文件类型是关联到外部程序的，则使用外部程序打开，如 (.doc, .chm, .pdf)

如果文件类型没有注册的，IDEA会打开一个对话框，让你注册一个新类型，或者使用相关的外部程序打开

2. 可以使用下面的几种方式外部文件进行编辑

- (1) 通过主菜单 File - Open File
- (2) 将外部文件拖到Android Studio编辑器中

3. 打开最近打开过的文件

- (1) 通过主菜单 View - Recent Files
- (2) 快捷键 Command+E

4. 打开最近更改过的文件

- (1) 通过主菜单 View - Recently Changed Files
- (2) 快捷键 Command+Shift+E

## 4.2. 标签管理/操作

## 1. 标签编辑

《Android Studio入门指南》作者：毕小鹏 博客：<http://blog.csdn.net/wirelessqa>

任一时刻打开一个文件，会以标签页的形式在编辑窗口中出现。该标签页出现在当前活动的标签页的下个位置，并成为新的活动标签页，当编辑窗口的标签页达到上限时，它会根据标签页关闭的优先策略来关闭掉别的标签页。当关闭活动的标签页时。会根据活动标签页选取策略来选择下一个活动标签页  
标签页关闭的优先策略有两种：

没有被修改的文件标签页优先被关闭

使用的最少文件标签页优先被关闭

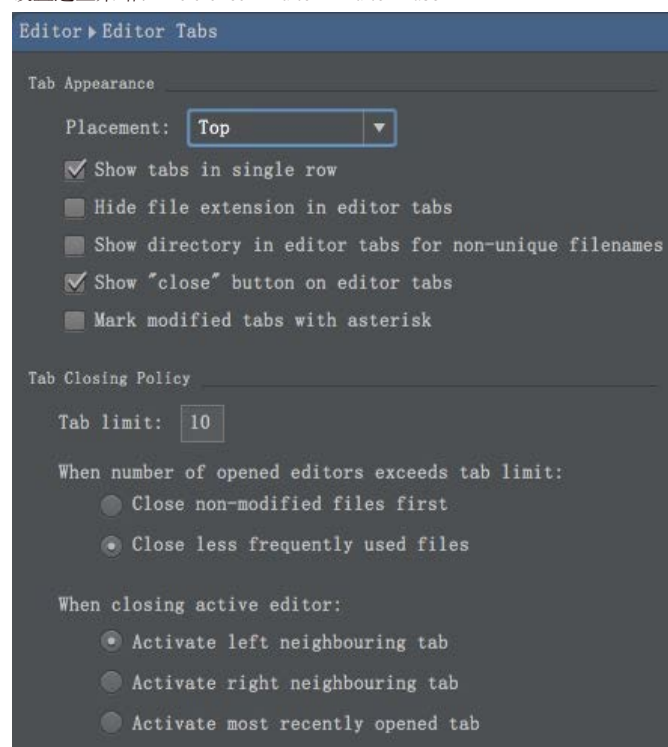
标签页选取策略有三种：

靠近被关闭的活动标签页的左侧标签页

靠近被关闭的活动标签页的右侧标签页

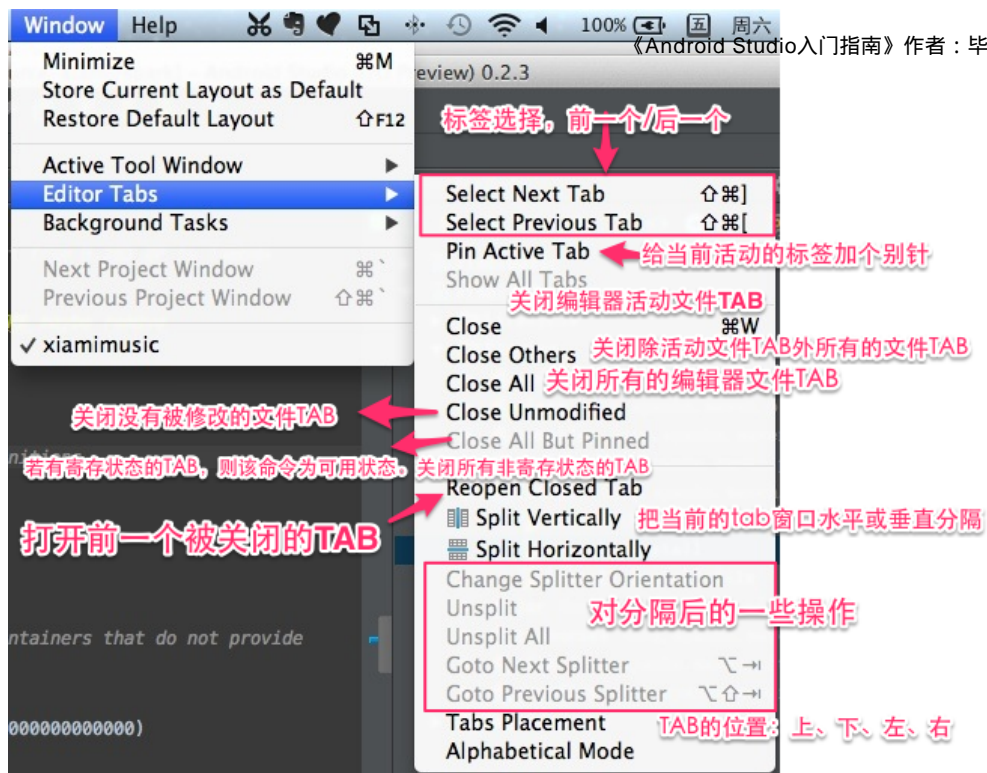
使用得最频繁的标签页

设置这些策略：Preference->Editor->Editor Tabs

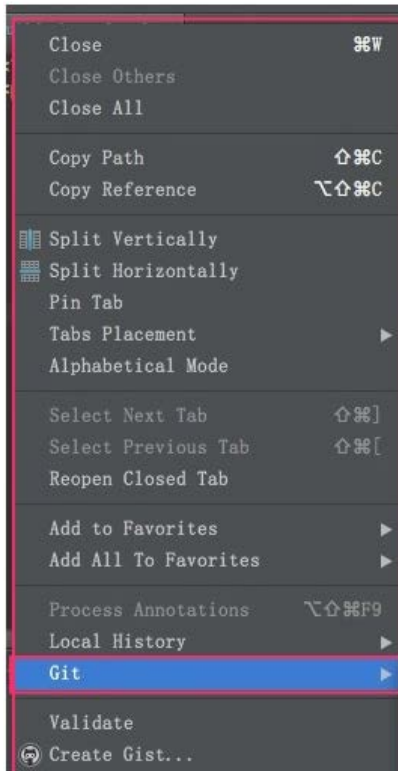


## 2. 标签操作

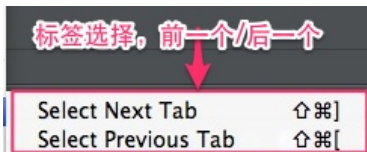
路径1： 通过Window - Editor Tabs 进行操作



路径2: 在tab上右击, 可调出操作菜单

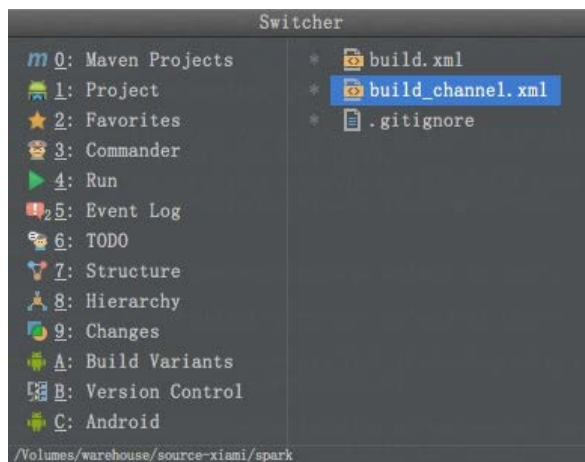


3. 在标签页间进行切换



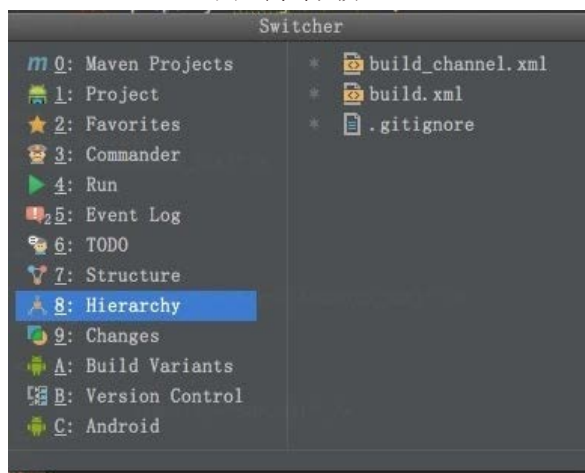
4. 使用Switcher功能

control+Tab 项目文件之间切换



《Android Studio入门指南》作者：毕小鹏 博客：<http://blog.csdn.net/wirelessqa>

control+Shift+Tab 工具之间的切换



5. 在使用过的标签页中前进和后退

command+option+Left(后退), command+option+Right (前进)

6. 拆卸编辑标签页

IDEA支持拆卸编辑标签页，将它挪成一个独立的窗口显示。

拆卸标签页：拖动标签页离开编辑窗口，直到有预览的缩略图出现，放下

复原标签页：拖动独立窗口中的标签页进入编辑窗口的标签栏摆放处，放下

```
<!-- The local.properties file is created and updated.
It contains the path to the SDK. It should *NOT*
be checked into Version Control Systems. -->
<property file="local.properties" />

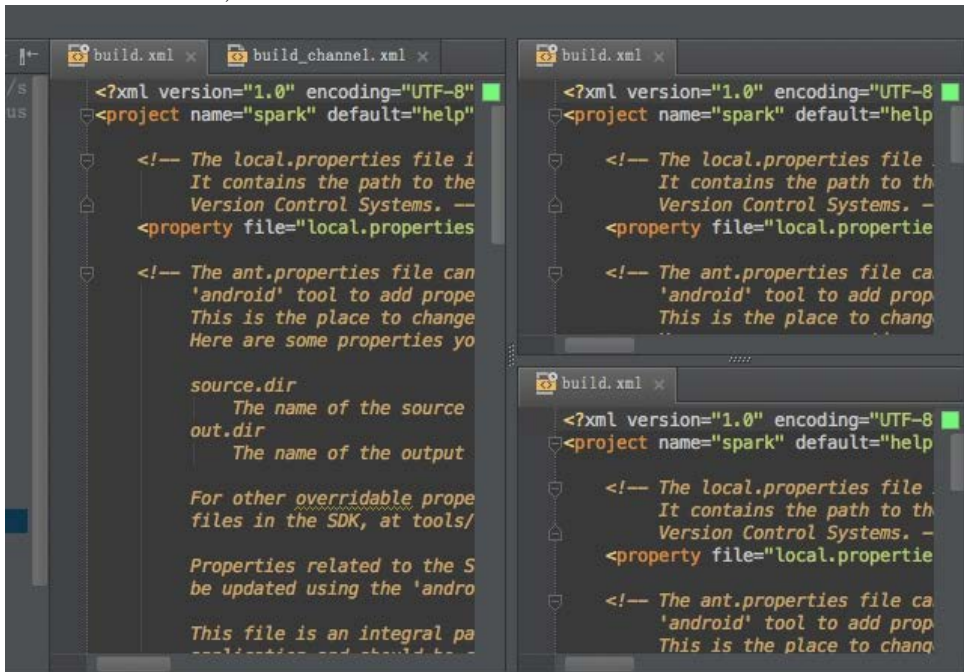
<!-- The ant.properties file can be created by you. It
contains the ant specific properties used by the
'android' tool to add properties to the project.
This is the place to change the ant specific build
properties for the project. Here are some properties you may want to change/
```

## 7. 寄存状态 Pin Active Tab

会给tab加个别针（寄存状态），寄存状态的标签页确保该标签页不会被编辑窗口自动关闭

## 8. 分离编辑窗口 Split Vertically + Split Horizontally

IDEA支持编辑窗口分离，满足同时查看一个文件不同部分的要求



分离一个标签窗口：

右键标签页，打开上下文菜单，选择Split Vertically or Split Horizontally

改变分离窗口的摆放方式：

右键标签页，打开上下文菜单，选择 Change Splitter Orientation

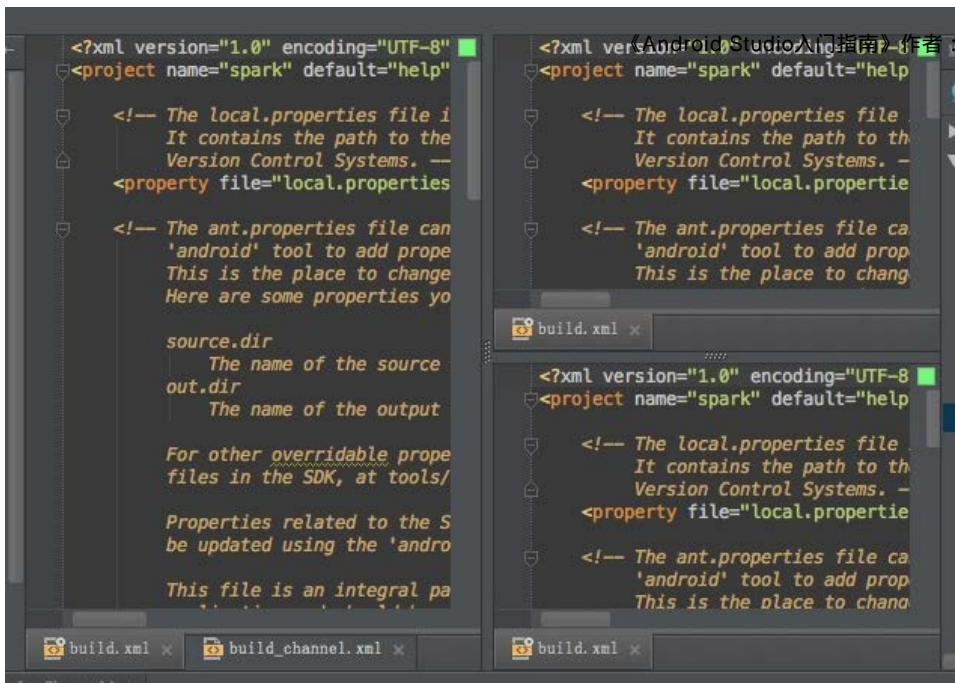
撤销分离：

右键标签页，打开上下文菜单，选择Unsplit（撤销当前活动的标签页的分离窗口） 或Unsplit All（撤销全部标签页的分离窗口）

## 9. 标签位置 Tab Placement – Bottom

build.xml这样的tab 都放到底部了





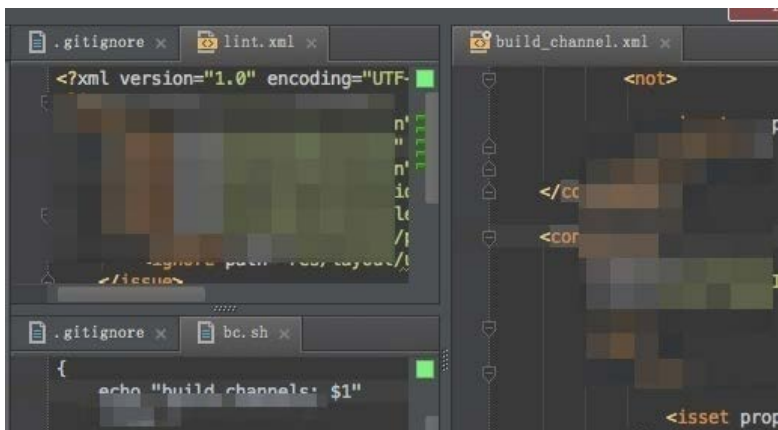
毕小鹏 博客：<http://blog.csdn.net/wirelessqa>

#### 10. 通过标签组的方式支持多文件编辑

IDEA可以通过将标签页分组的方式支持多文件同时编辑，组的数量没有限制

创建一个标签页组;只需要简单的将一个标签页分离，就可以创建一个新的标签页组了

将一个标签页移动到另外一个分组，选择需要移动的标签页，右键打开上下文菜单，选择 Move to Opposite Group



#### 4.3. 保存文件

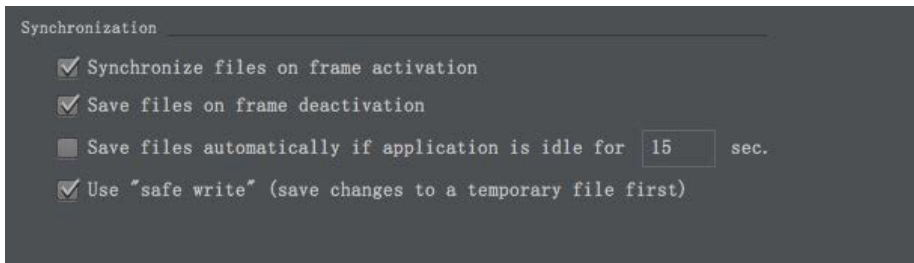
Android Studio支持手动保存和自动保存，支持两种方式的自动保存:

定期保存：以一定的间隔时间，自动保存文件

切换窗口时保存：当从IDEA编辑窗口切换到其他窗口时，自动保存已经修改过的文件（默认选择）

《Android Studio入门指南》作者：毕小鹏 博客：<http://blog.csdn.net/wirelessqa>

设置自动保存 Preference -- General -- Synchronization 勾选自动保存的方式

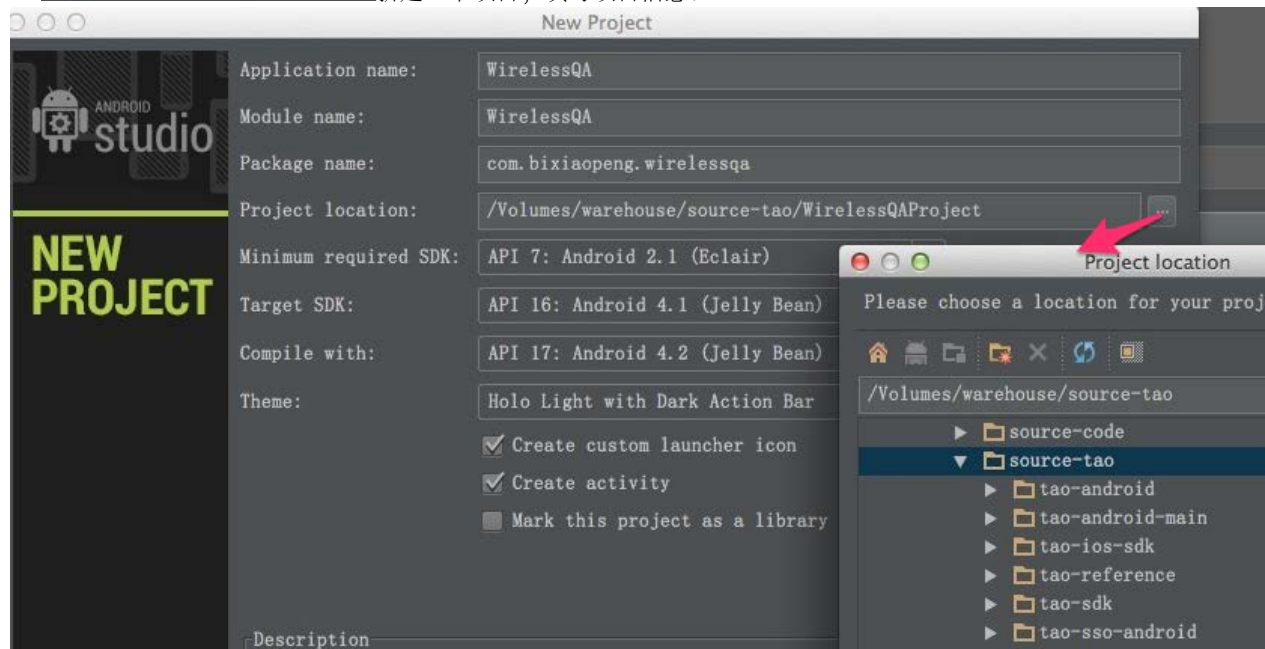


## 5. Android Studio项目操作

### 5.1. 新建一个Android 项目

首先要说一点，在IntelliJ IDEA里面“new Project”就相当于我们eclipse的“workspace”，而“new Module”才是创建一个工程。

1.  New Project... 新建一个项目，填写项目信息：



2. 接下来默认就好了

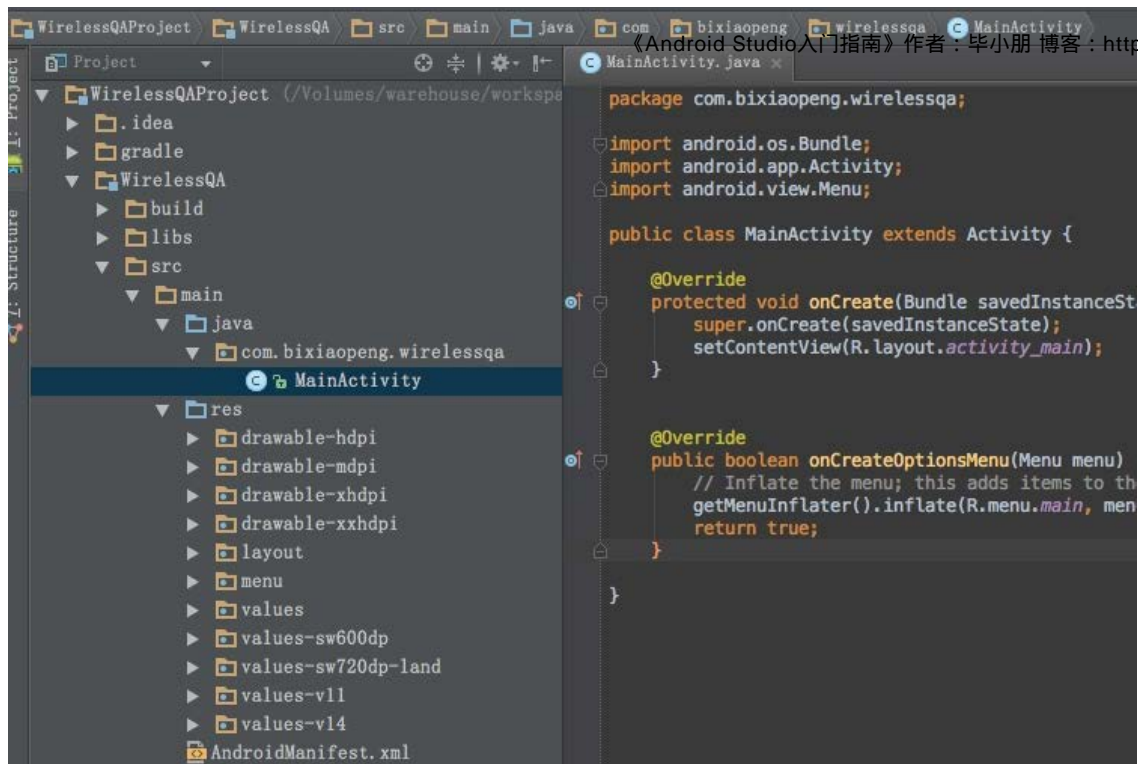


OK

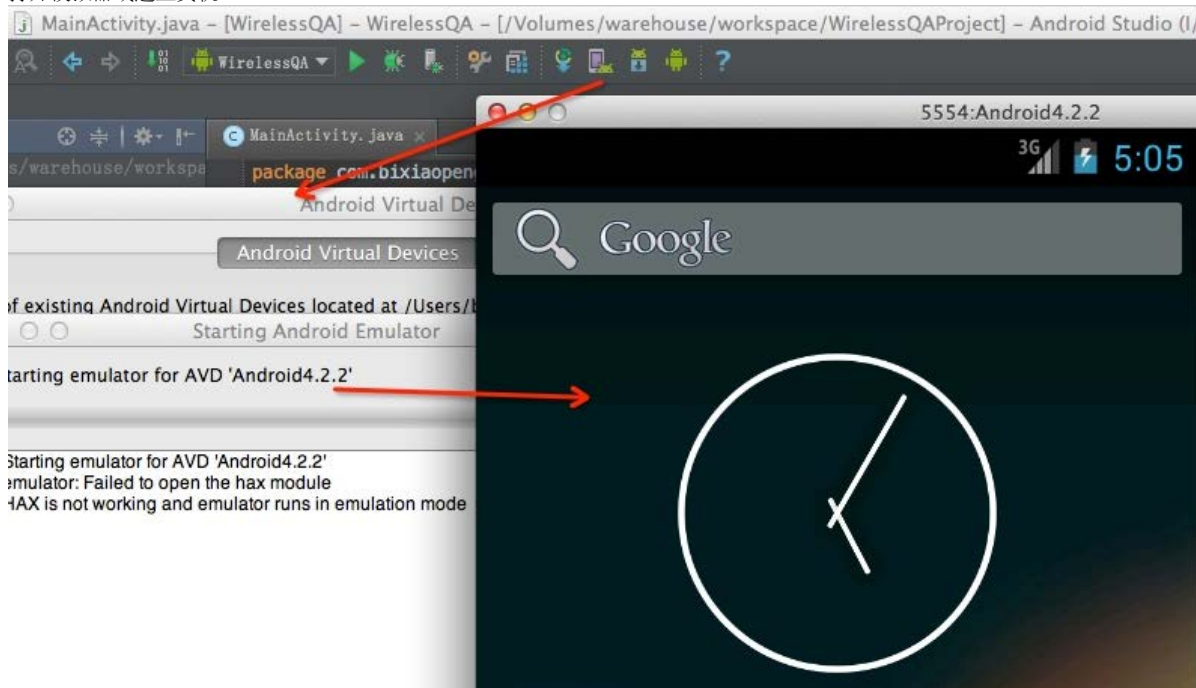


3. 下载完成后，进入主界面。用户可以编写相应代码。

工程的结构和Eclipse上的不同，src下分为java和res



打开模拟器或连上真机

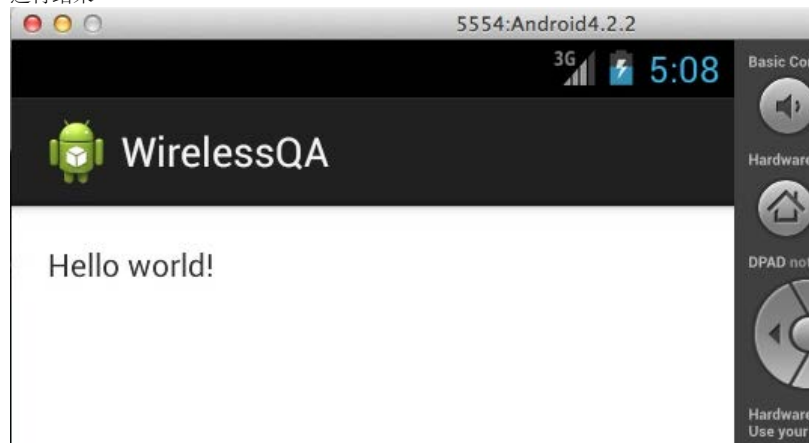


工具栏上点击运行  按钮或快捷键 **control+R** 来运行我们的项目 [《Android Studio入门指南》作者：毕小鹏 博客：http://blog.csdn.net/wirelessqa](http://blog.csdn.net/wirelessqa)

这里如果设置过模拟器，会有一个选择设备的提示，我们可以选择正在运行的设备，当然也可以选择一个新的模拟器来重新启动



运行结果



运行时打出的信息：

Waiting for device.

Target device: emulator-5554 (Android4.2.2) //准备将包安装到的指定的设备

Uploading file //将apk包上传到手机中指定的目录

local path: /Volumes/warehouse/workspace/WirelessQAProject/WirelessQA/build/apk/WirelessQA-debug-unaligned.apk

remote path: /data/local/tmp/com.bixiaopeng.wirelessqa

Installing com.bixiaopeng.wirelessqa //安装包

DEVICE SHELL COMMAND: pm install -r "/data/local/tmp/com.bixiaopeng.wirelessqa"

pkg: /data/local/tmp/com.bixiaopeng.wirelessqa

Success

Launching application: com.bixiaopeng.wirelessqa/com.bixiaopeng.wirelessqa.MainActivity. //启动MainActivity

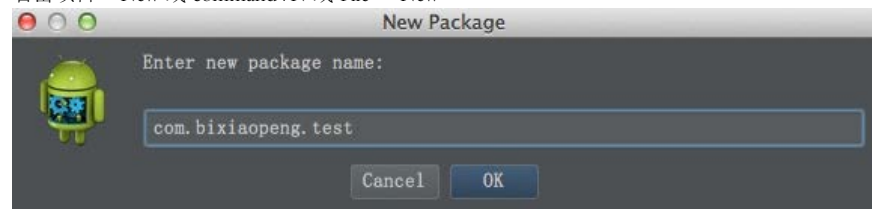
DEVICE SHELL COMMAND: am start -n "com.bixiaopeng.wirelessqa/com.bixiaopeng.wirelessqa.MainActivity" -a android.intent.action.MAIN -c android.intent.category.LAUNCHER

Starting: Intent { act=android.intent.action.MAIN cat=[android.intent.category.LAUNCHER] cmp=com.bixiaopeng.wirelessqa/.MainActivity }

## 5.2. 新建一个包

《Android Studio入门指南》作者：毕小鹏 博客：<http://blog.csdn.net/wirelessqa>

右击项目-- New 或 command+N 或 File -- New

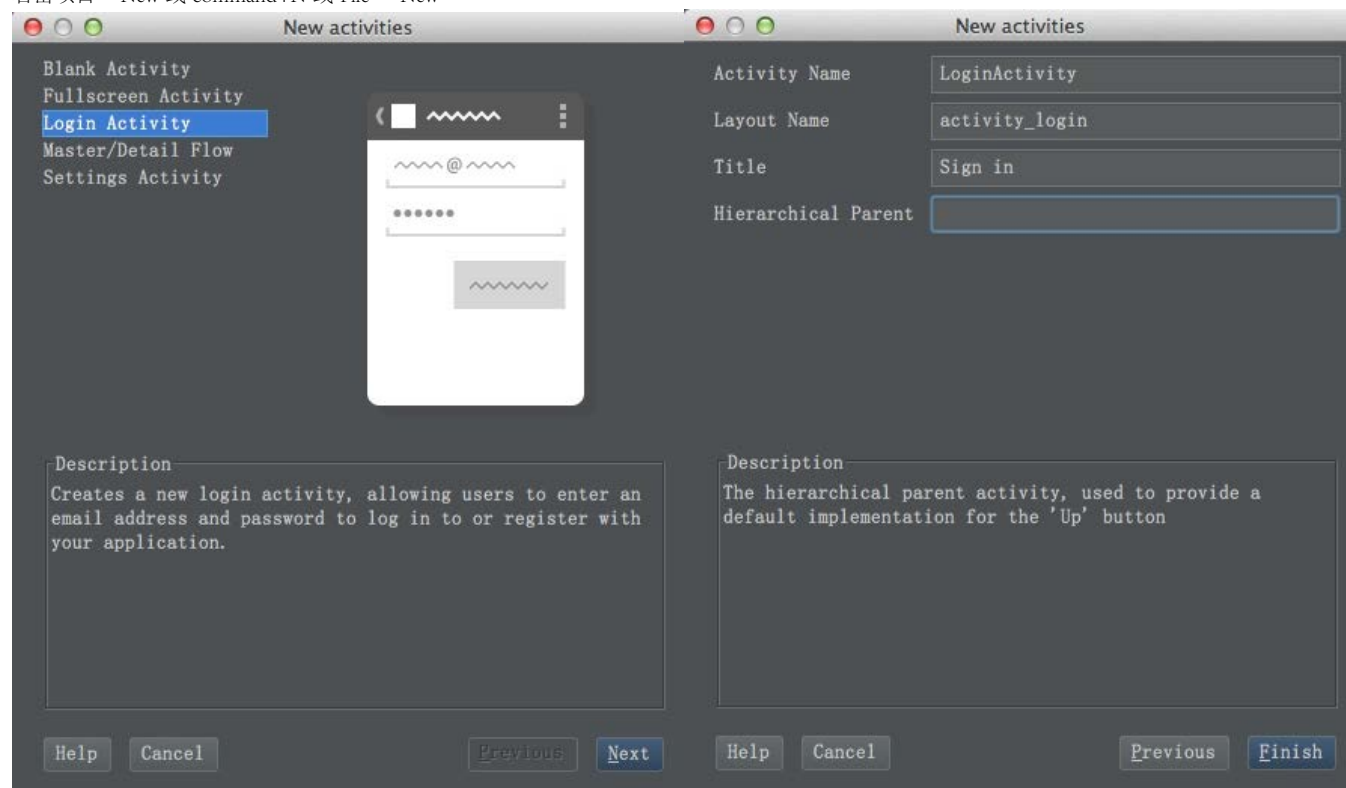


如果包名前面路径相同，会被归到同一个文件夹下



## 5.3. 新建一个Activity

右击项目-- New 或 command+N 或 File -- New



如果新建的文件已存在，会有相应的提示





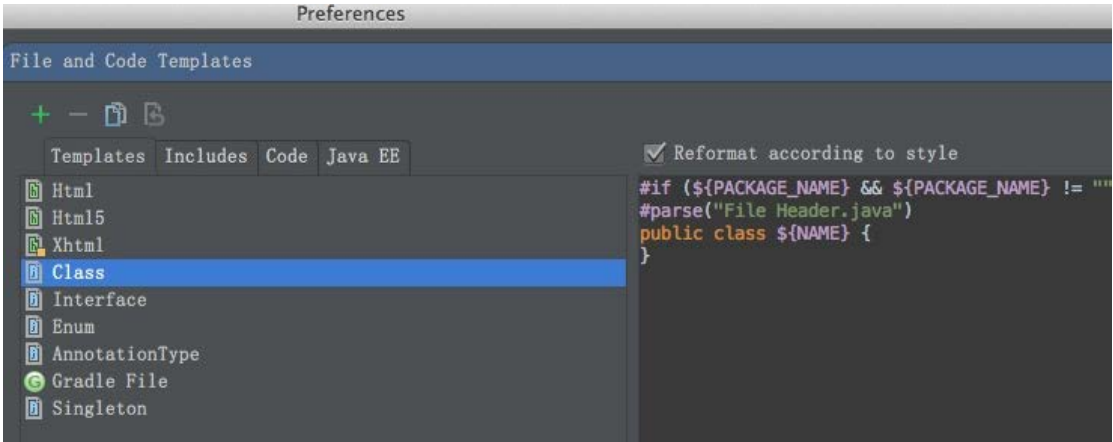
## 5.4. 文件操作

打开新建成功的LoginActivity

### 1. 文件模板

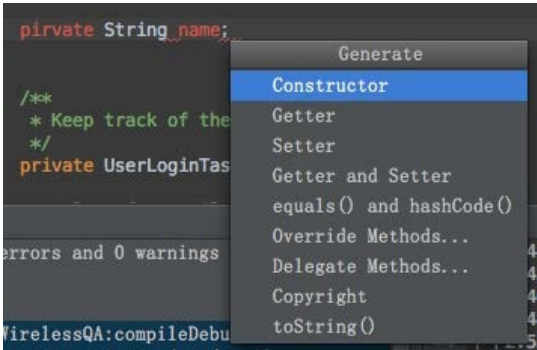
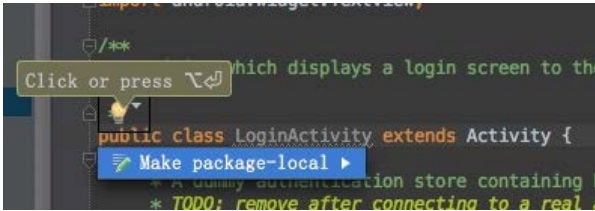
注意文件开头的包申明和类申明.当创建类时, Android Stduo应用文件模板. (IntelliJ IDEA为创建不同的类型文件供给了多少预定义的文件模板)

如果想修改, 请到Preferences -- File and Code Templates下面修改



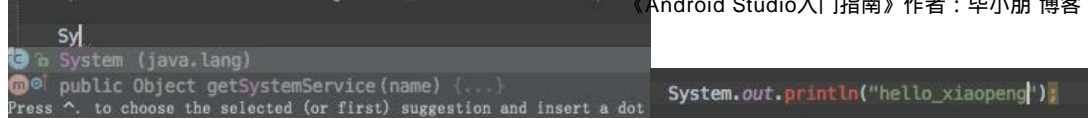
### 2. 建议列表

注意一下黄色灯泡.它指示Android Studio对于此上下文有一些建议.点击灯泡,或者按下option+Enter来查看建议列表



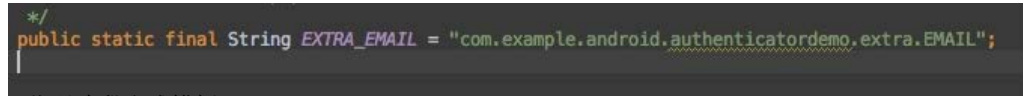
### 3. 智能提示

《Android Studio入门指南》作者：毕小鹏 博客：<http://blog.csdn.net/wirelessqa>



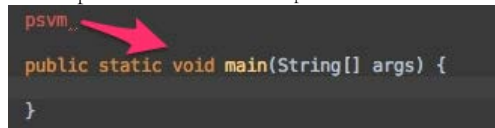
### 4. 新建一行

不管光标位置在哪,按下Shift+Enter开始一个新行并且能保持前面那行的完整性



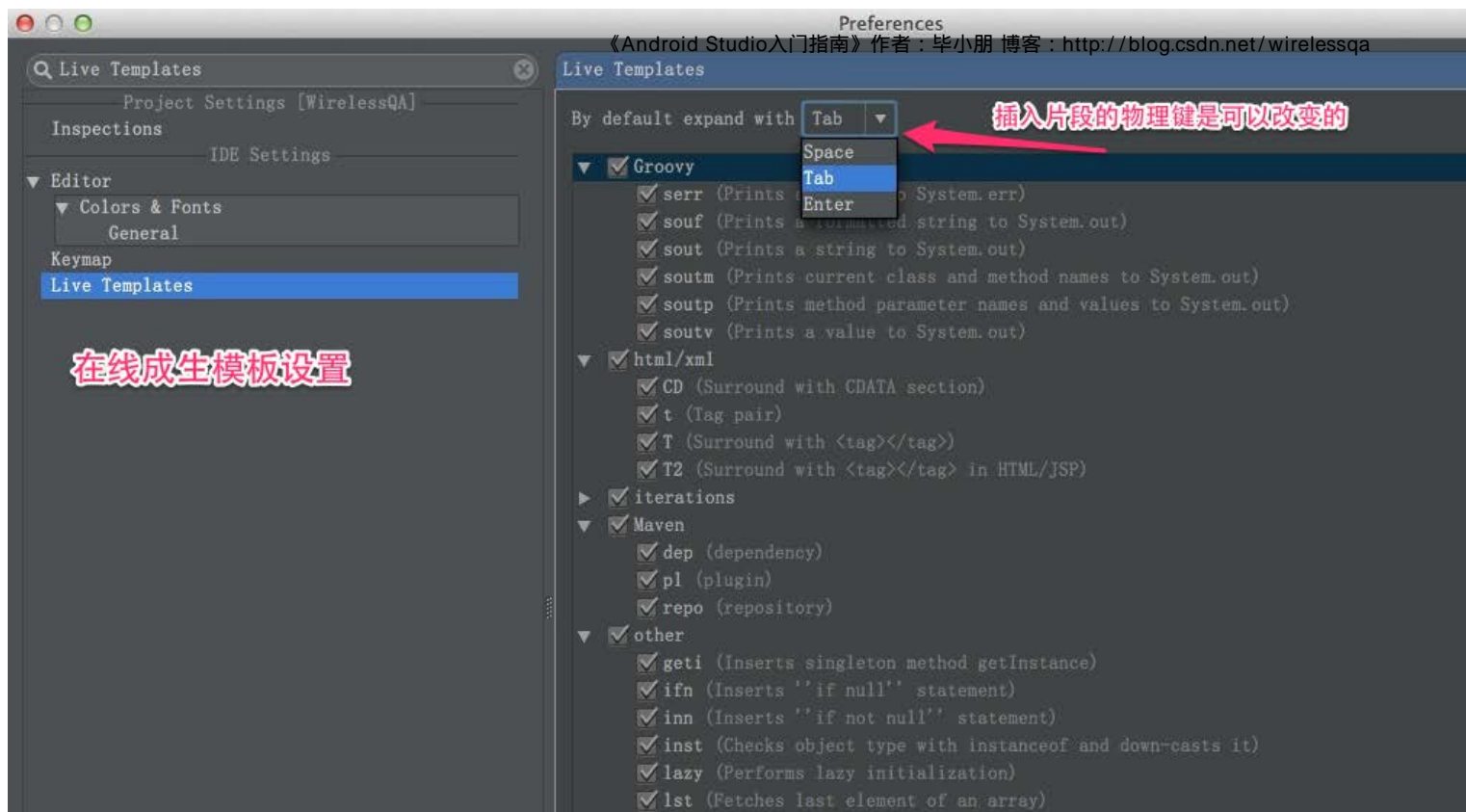
### 5. 代码片段生成模板

举例: psvm + Tab 即: 输入psvm然后点击Tab就可以生成



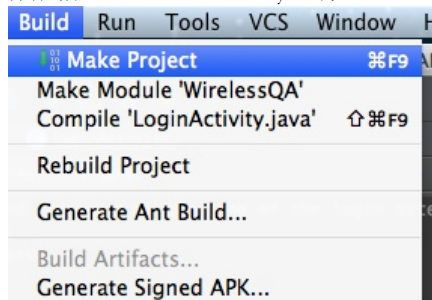
我们刚才应用的是基于在线模板的代码生成机制.被相应机制应用的在线模板是能够被插入到你的代码中的代码片段.

一个在线模板有一个缩写形式,一个字符串标识这个模板(这个例子中是psvm).还有一个扩展键,键盘上按下以插入片段的物理键(这里是Tab)



## 5.5. 项目构建

操作路径：Build -- Make Project 或 command + F9



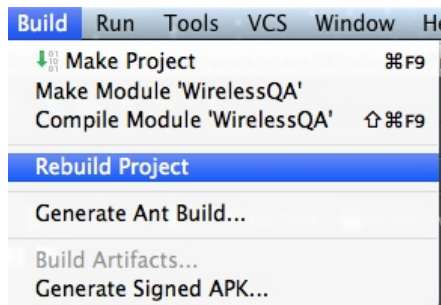
开始构建 % Gradle build using tasks: [:WirelessQA:assembleDebug]

构建结果: Compilation completed with 16 errors and 0 warnings in 3 sec (a minute ago)

## 5.6. clean = 重新构建项目

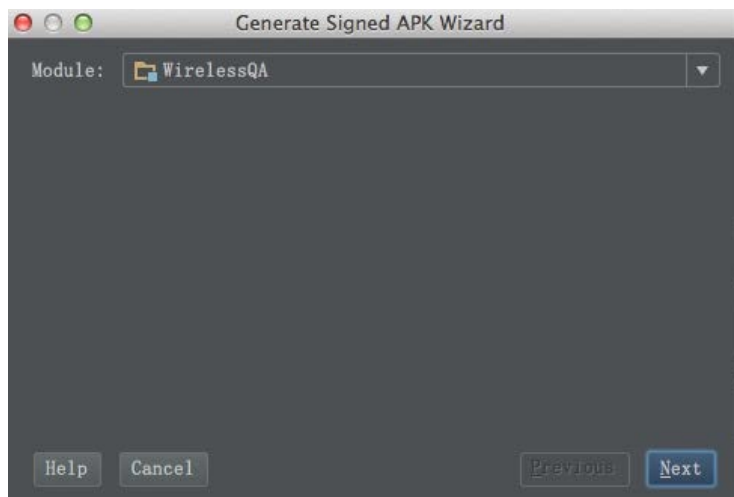
《Android Studio入门指南》作者：毕小鹏 博客：<http://blog.csdn.net/wirelessqa>

使用 Build -- Rebuild Project

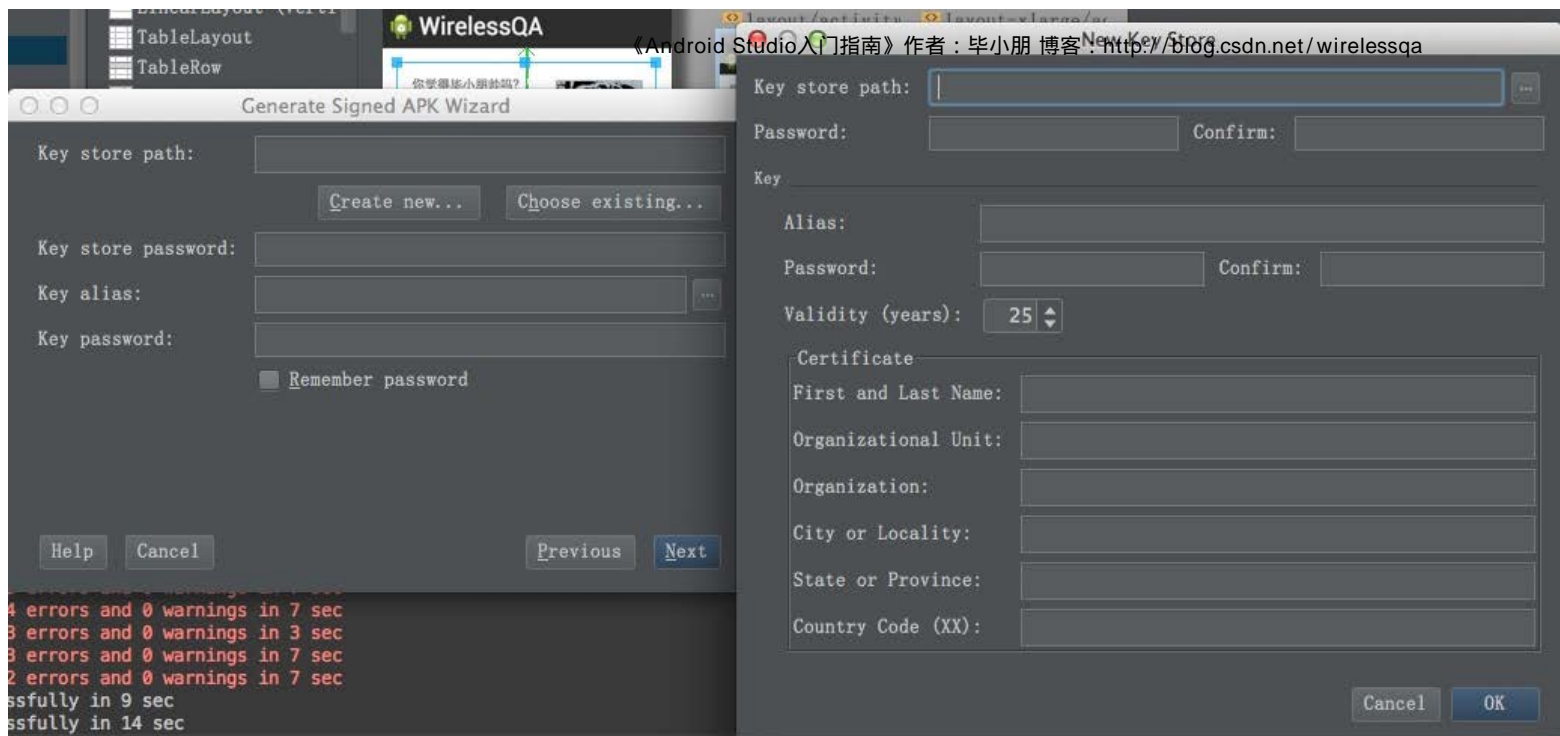


## 5.7. 签名并导出APK

Build -- Generate Signed Apk

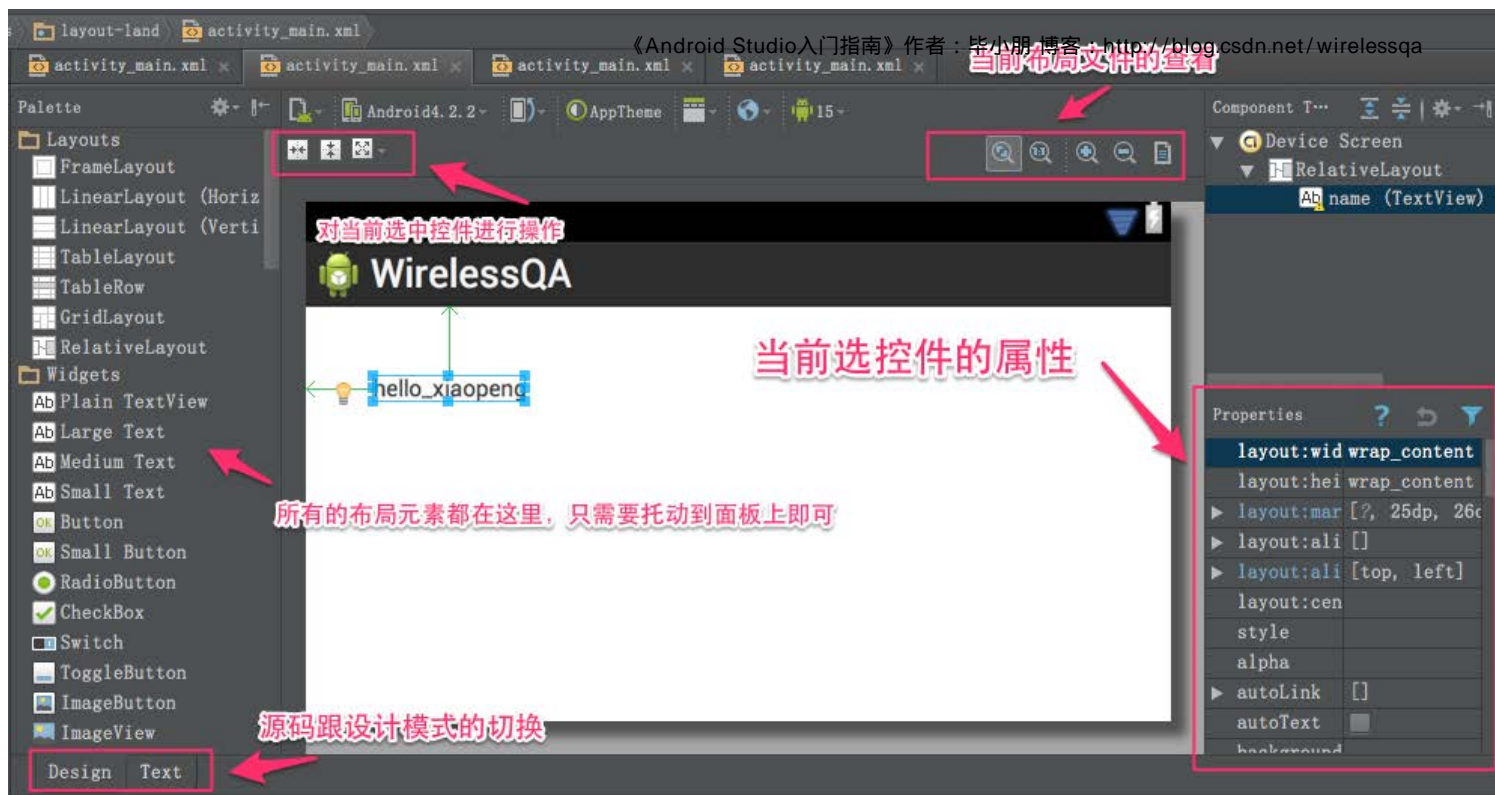


这里可以新建一个keystore,也可以选择一个已存在的keystore,这个很简单，不多介绍了。



## 6. Android Studio 布局文件

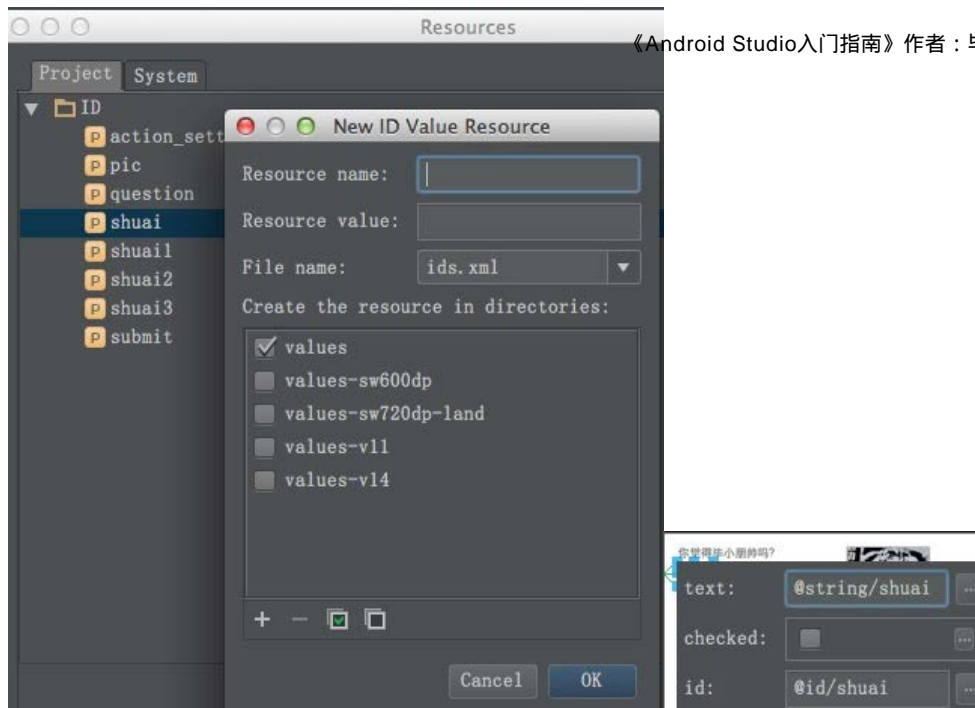
### 6.1 概览



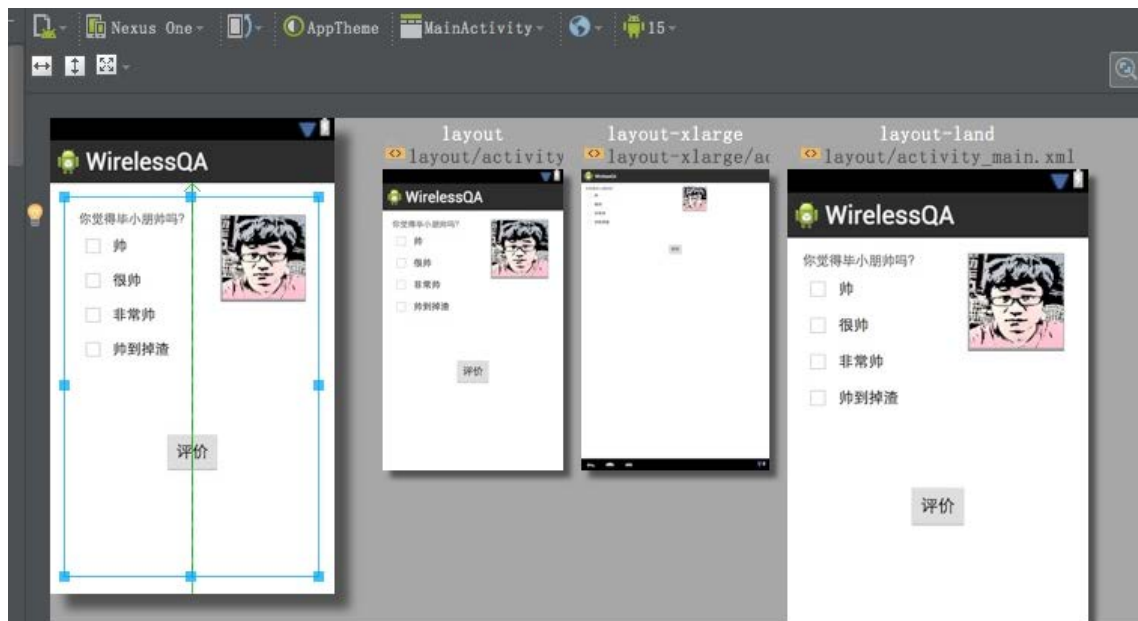
## 6.2 修改id/text

双击选中控件可以修改text和id

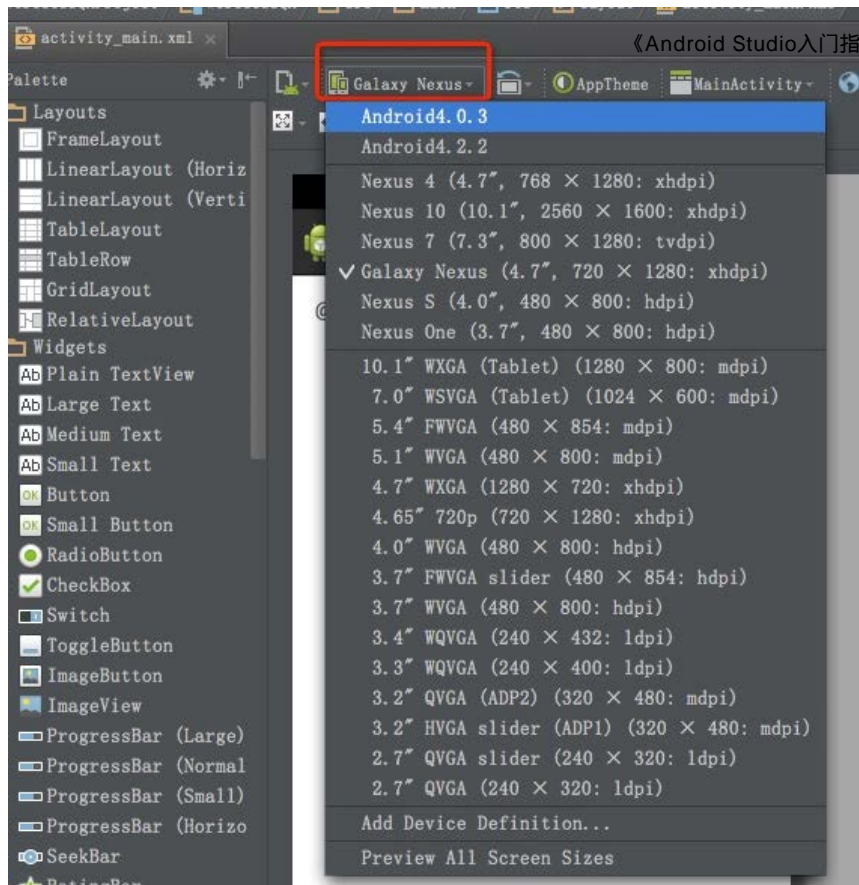




### 6.3 布局适配



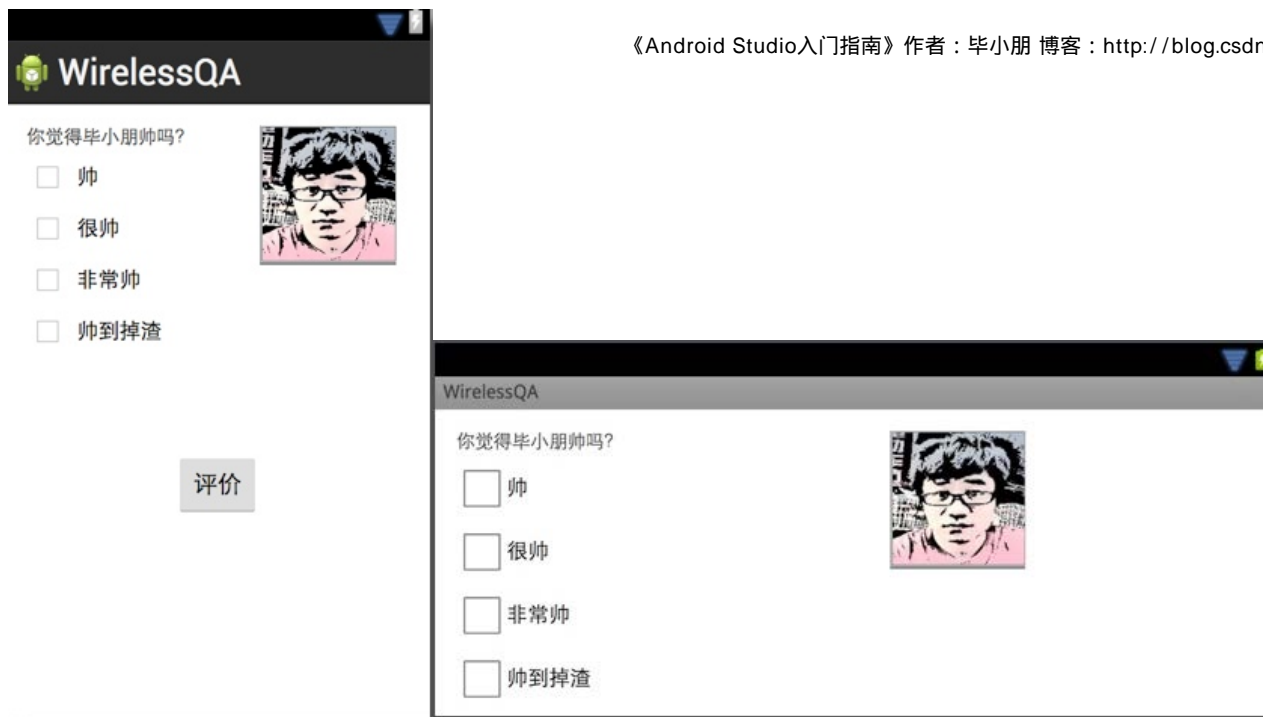
### 6.4 机型适配



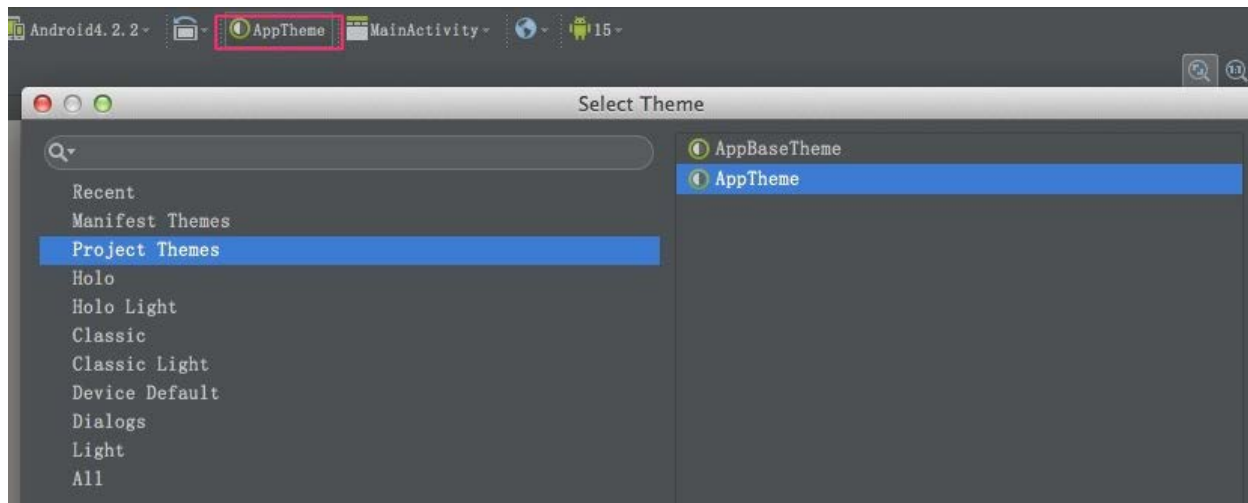
太帅了，有木有



## 6.5 横竖屏切换



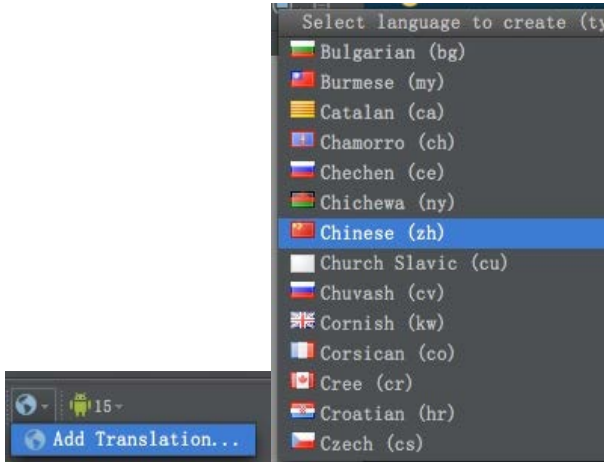
## 6.6 主题选择



## 6.7 版本适配



## 6.8 语言适配



运行看一下效果：



## 7.真机/模拟器调试

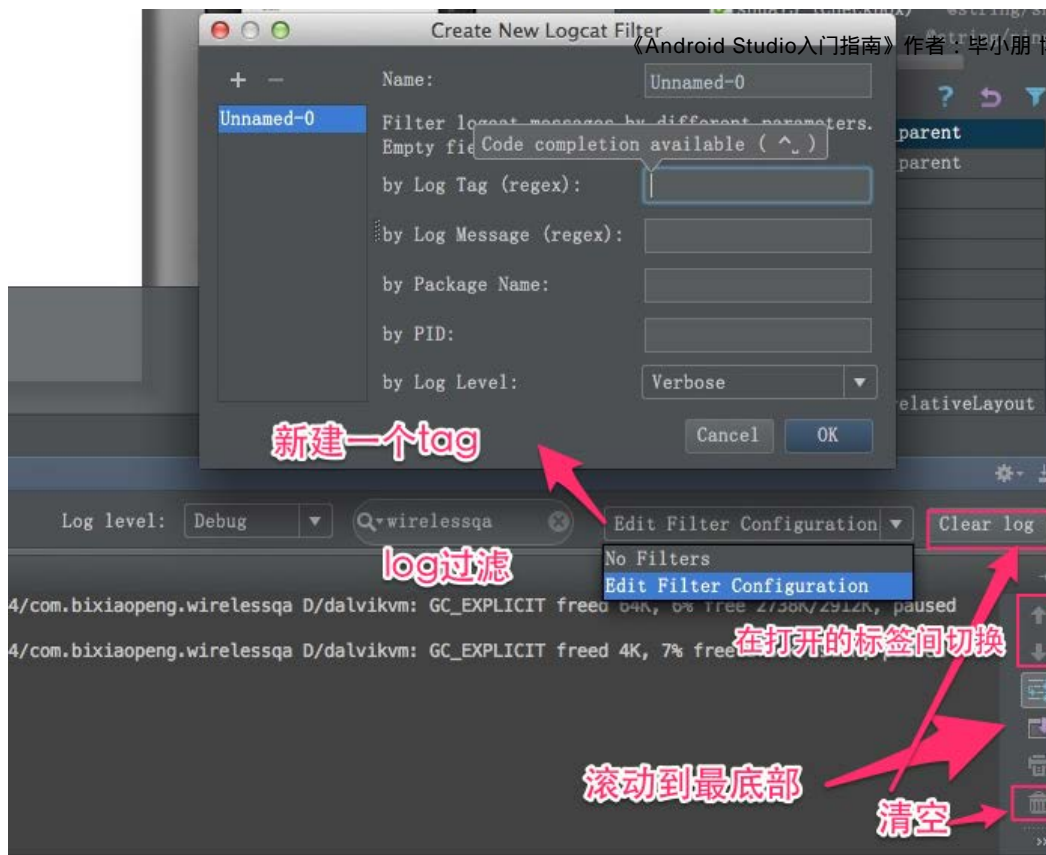
### 7.1 打开调试界面

打开Android调试界面: View -- Tool Windows -- Android 或使用option + 6

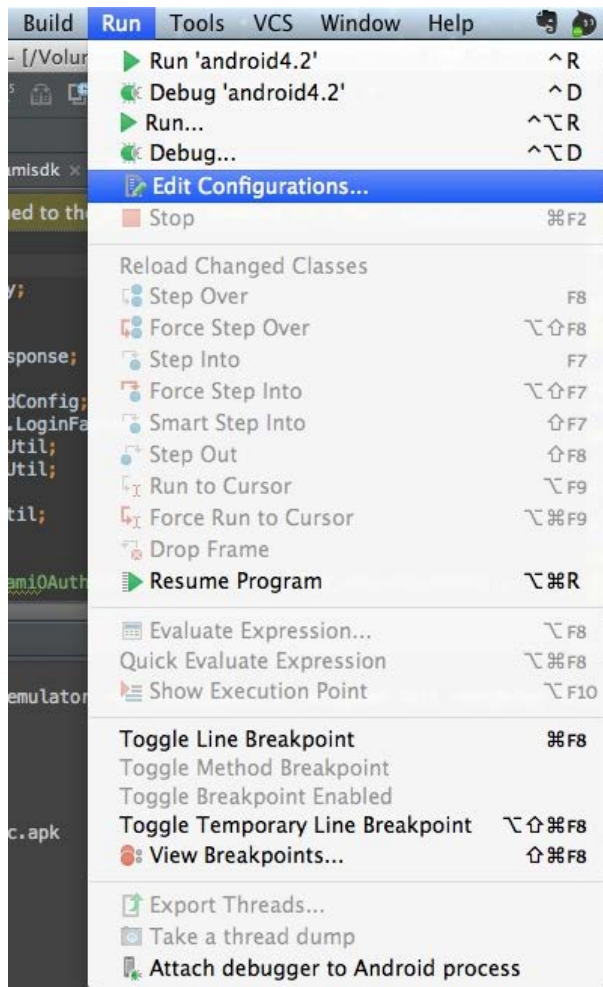


对于log的过滤等操作

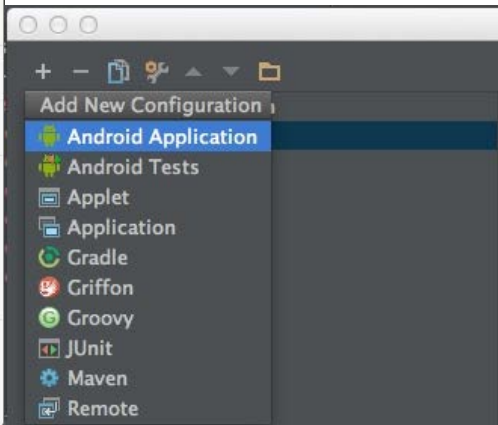


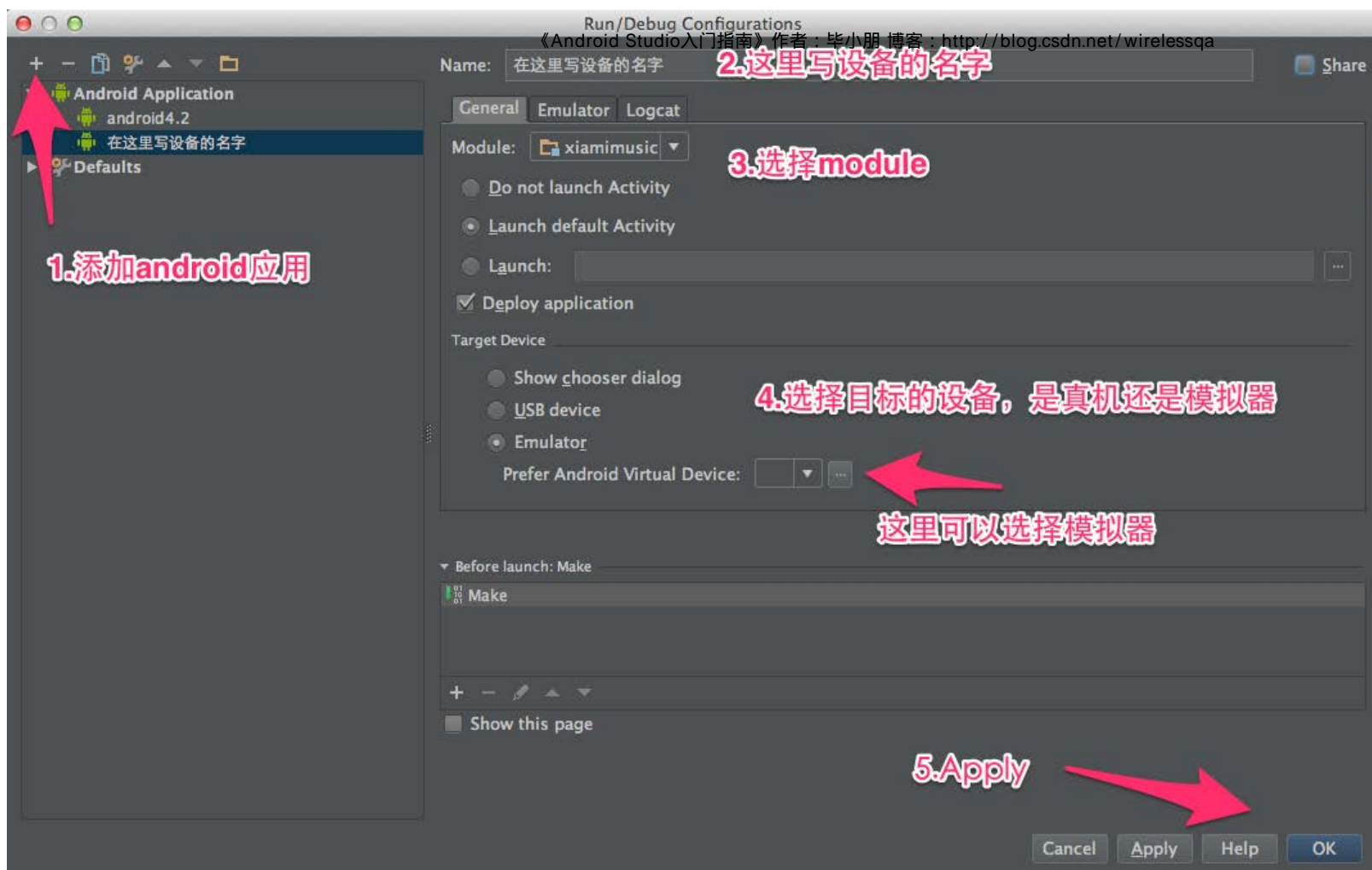


## 7.2 添加调试设备

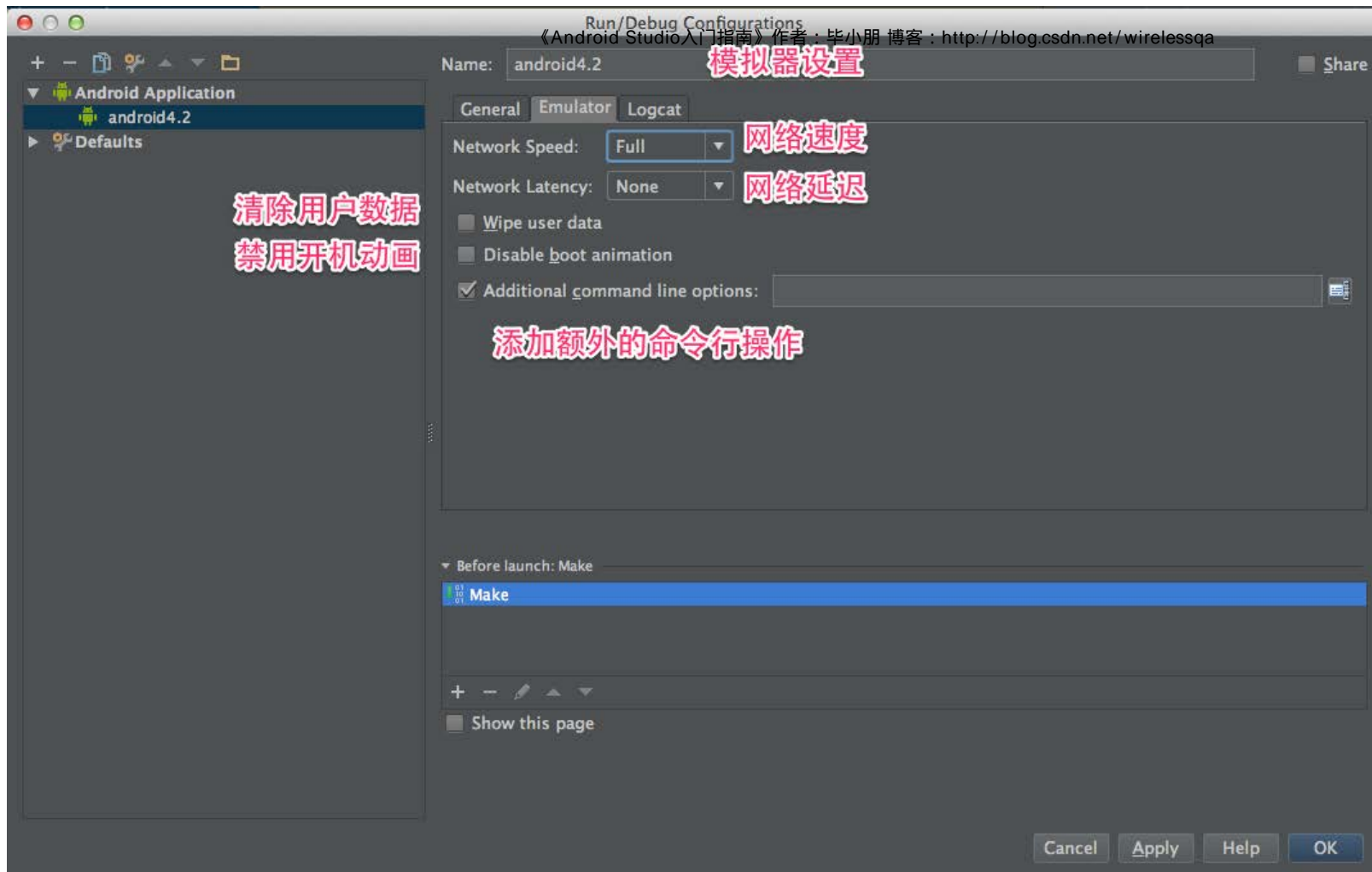


《Android Studio入门指南》作者：毕小鹏 博客：<http://blog.csdn.net/wirelessqa>



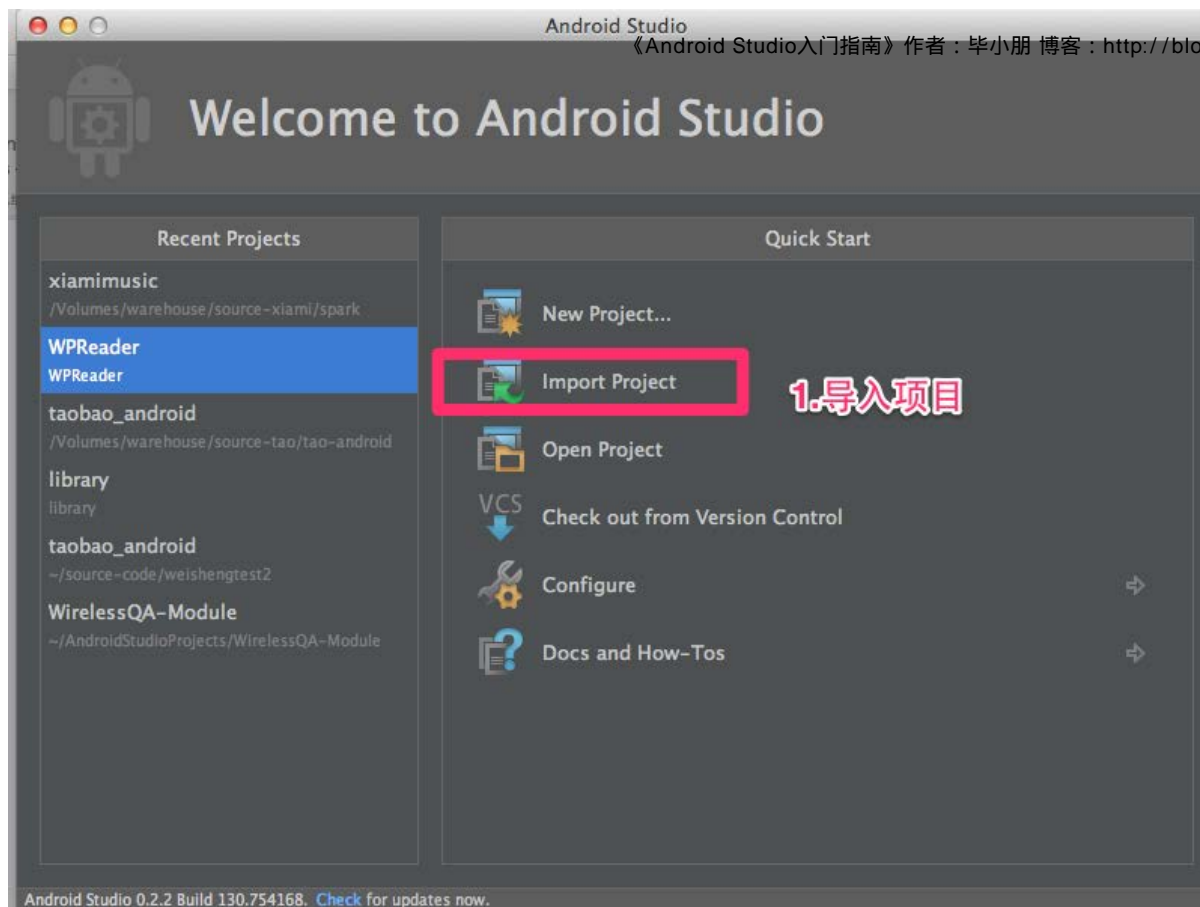


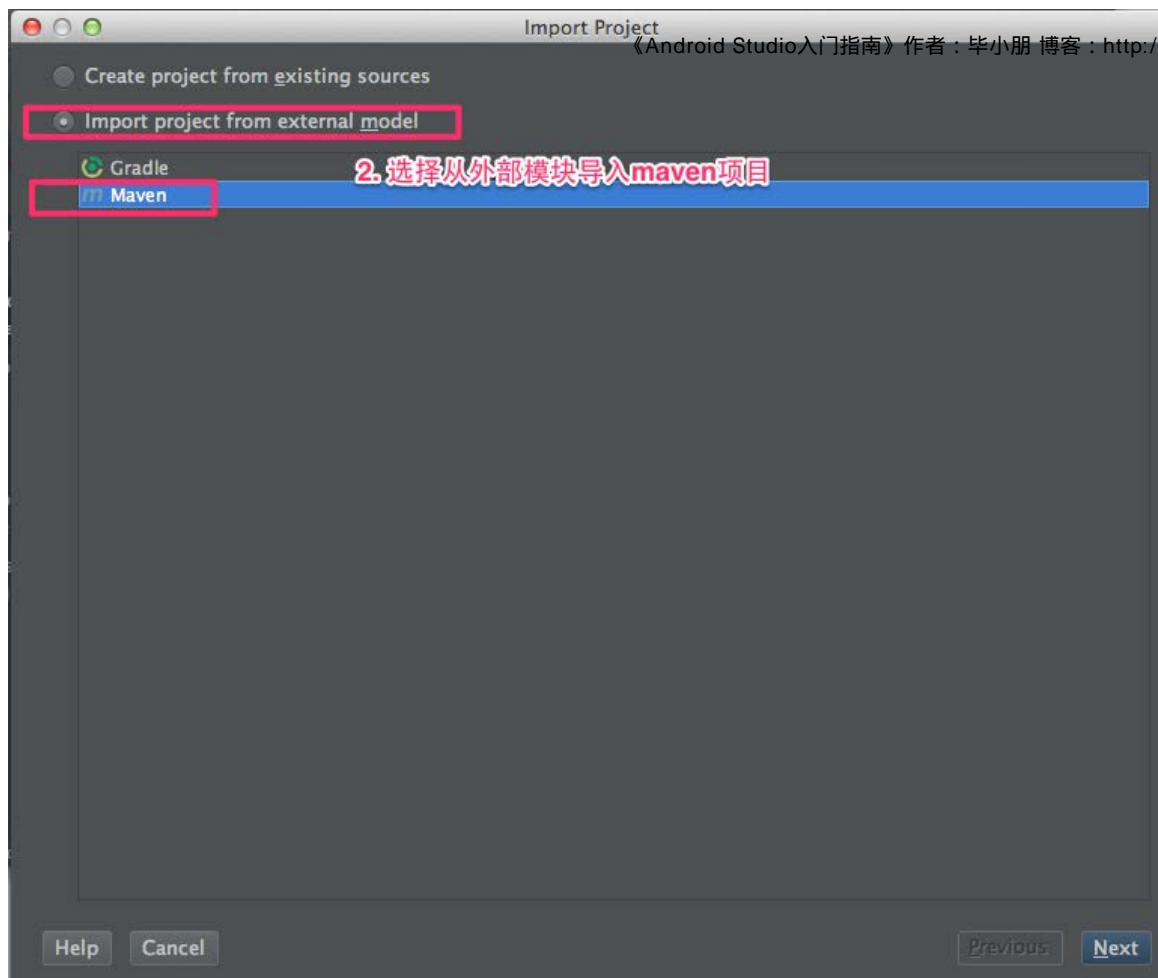
### 7.3 模拟器的配置



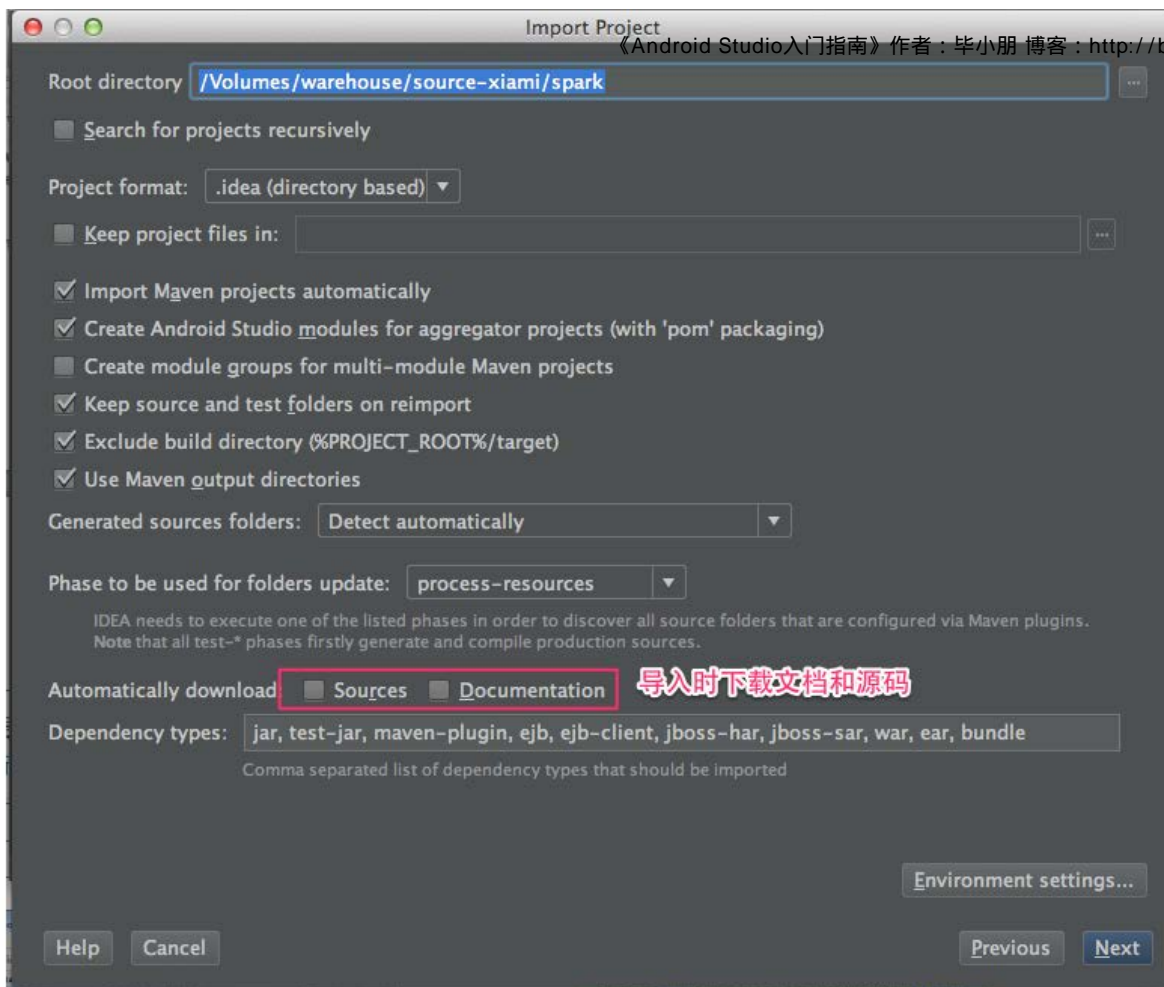
## 8. 导入项目

### 8.1. 导入maven项目



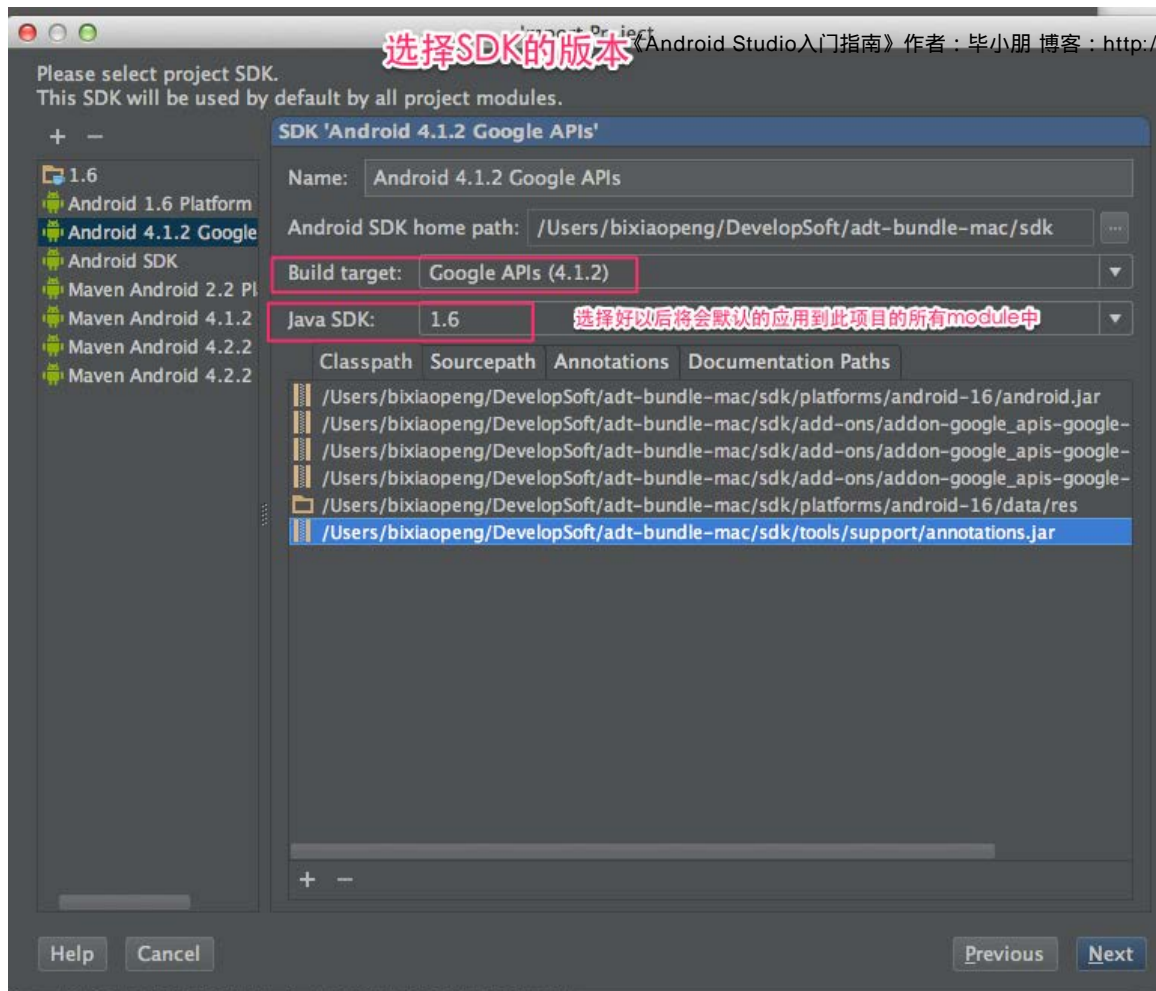




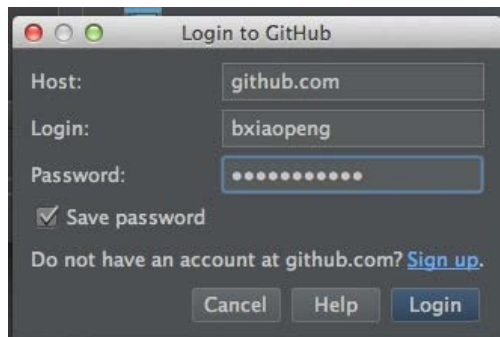
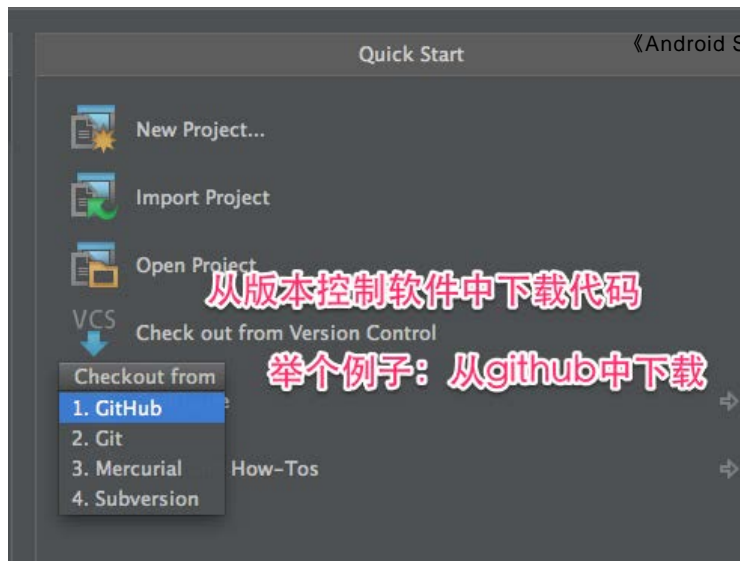


选择SDK的版本

《Android Studio入门指南》作者：毕小鹏 博客：<http://blog.csdn.net/wirelessqa>



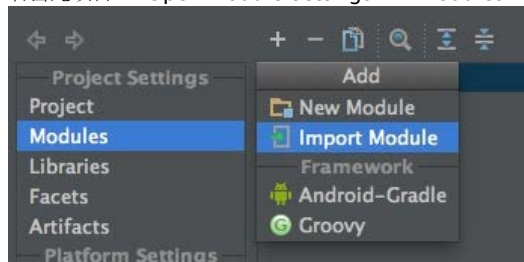
## 8.2. 从代码仓库中导入

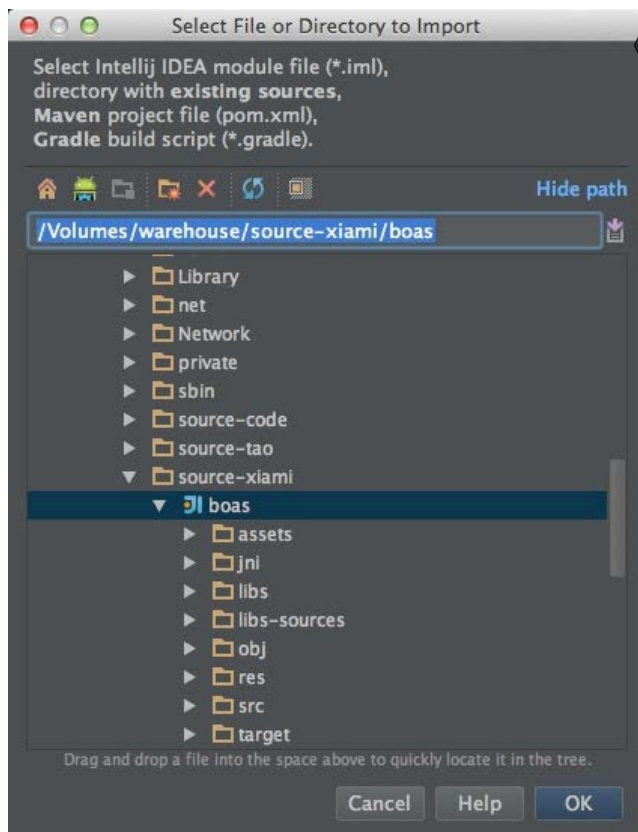


### 8.3. 导入module

说明： Android Studio的目录结构代表一个workspace,一个workspace里面可以依赖Module. module是什么?module就相当于eclipse里依赖的项目.

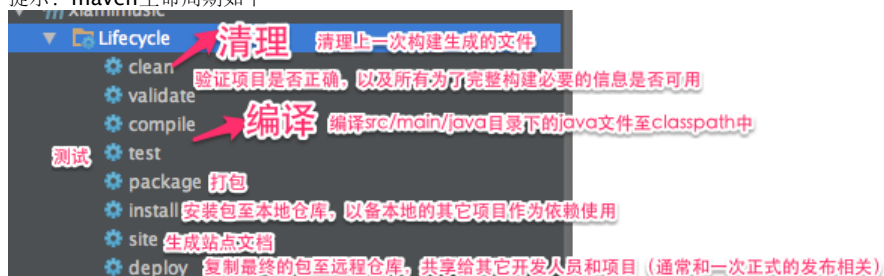
右击此项目-- Open Module Settings -- Modules -- import Module ,然后选择module





导入module后执行maven install ,将jar包安装到本地仓库，以后项目调用

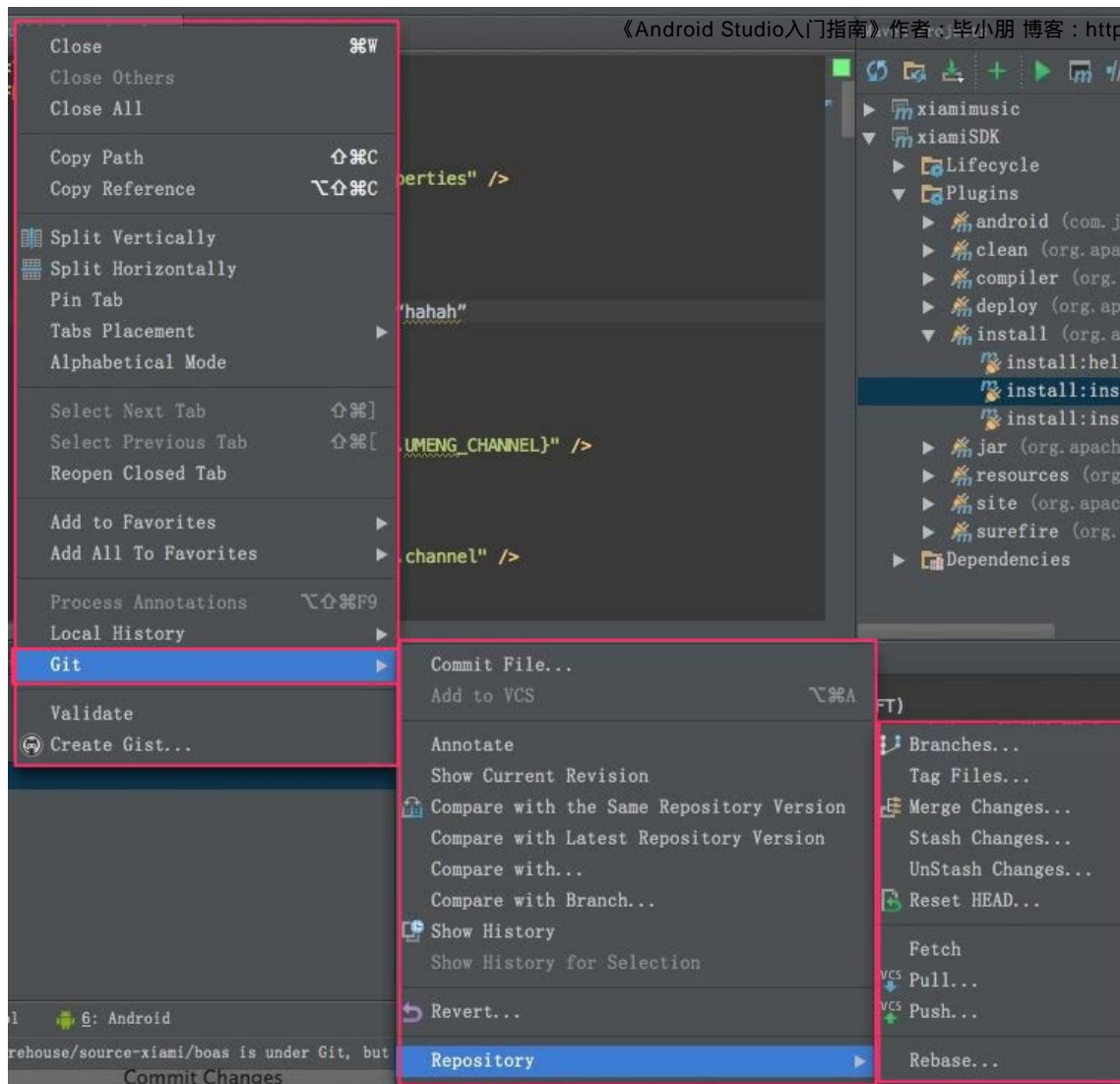
提示：maven生命周期如下



## 9. Android Studio Git操作

### 9.1. Git在哪里？

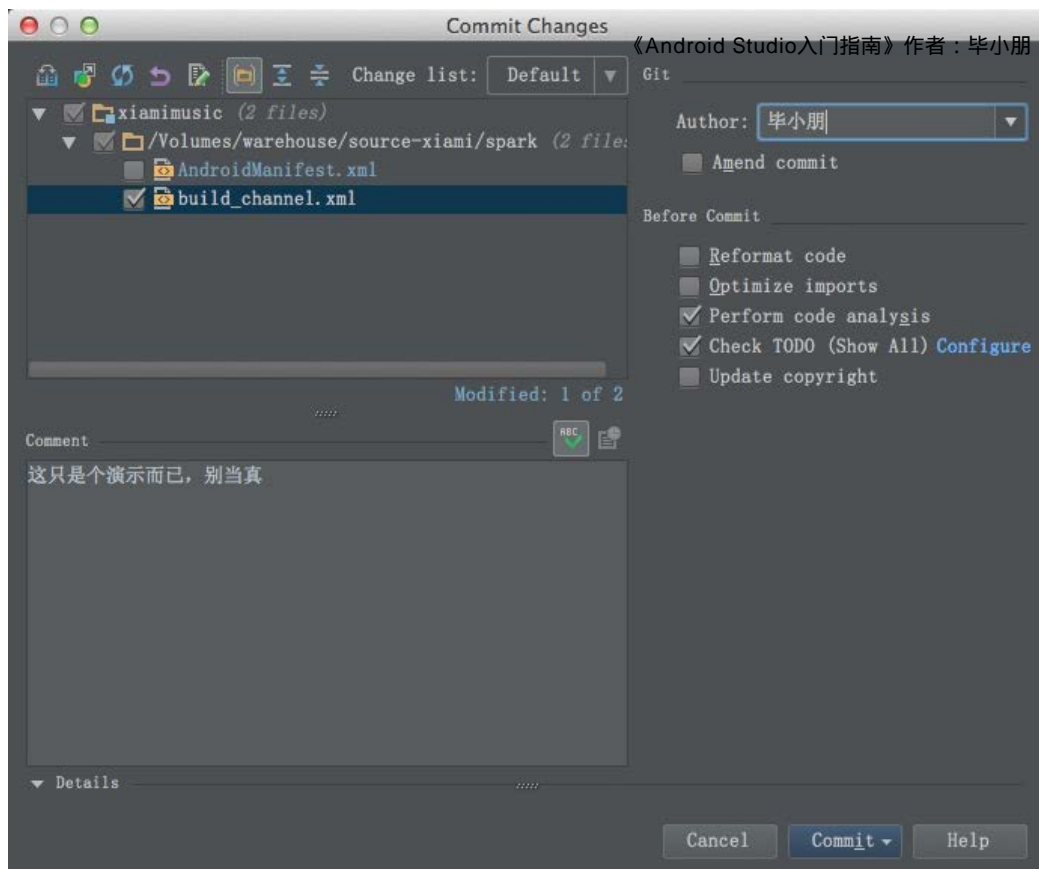
右击文件或项目可以找到Git



## 9.2. git有哪些功能

### 1. Commit File

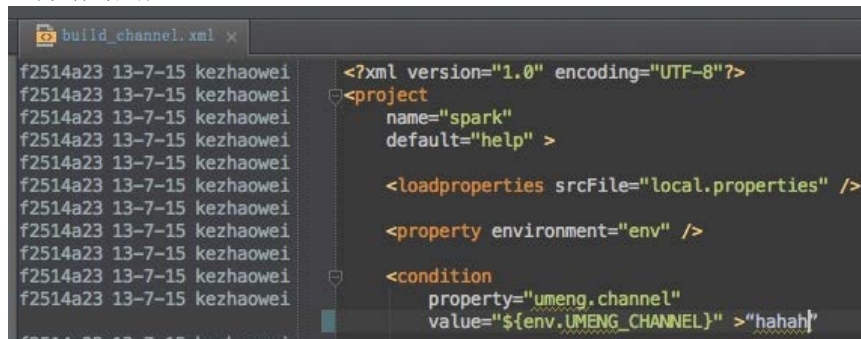
git commit命令，将索引内容添加到仓库中



《Android Studio入门指南》作者：毕小鹏 博客：<http://blog.csdn.net/wirelessqa>

## 2. Annotate

显示文件的注释



## 3. Show Current Revision

显示该文件最近一次改动



```
build_channel.xml

Current version is f2514a23fd9e5f0879233a2722falb0795980363.
Modified by kezhaowei
13-7-15 下午5:21
initial import
```

## 4. Compare

差异比较

Compare with the Same Repository Version  
Compare with Latest Repository Version  
Compare with...  
Compare with Branch...

/Volumes/warehouse/source-xiami/spark/build\_channel.xml

Ignore whitespace: Do not ignore ?

f2514a23fd9e5f0879233a2722falb0795980363 (Read-only) Local

| Repository | Local |
|------------|-------|
| 8          | 8     |
| 9          | 9     |
| 10         | 10    |
| 11         | 11    |
| 12         | 12    |
| 13         | 13    |
| 14         | 14    |
| 15         | 15    |
| 16         | 16    |
| 17         | 17    |
| 18         | 18    |
| 19         | 19    |

```
<property environment="env" />

<condition
  property="umeng.channel"
  value="${env.UMENG_CHANNEL}" >
  <and>
    <not>
      <isset property="${env.UMENG_CHANNEL}" />
    </not>
  </condition>
```

```
<property environment="env" />

<condition
  property="umeng.channel"
  value="${env.UMENG_CHANNEL}" >"hahah"
  <and>
    <not>
      <isset property="${env.UMENG_CHANNEL}" />
    </not>
  </condition>
```

1 difference Deleted Changed Inserted

## 5. Show History

/Volumes/warehouse/source-xiami/spark/build\_channel.xml

Ignore whitespace: Do not ignore ?

(Read-only) f2514a23fd9e5f0879233a2722falb0795980363 (Read-only)

| Repository | Local |
|------------|-------|
| 1          | 1     |
| 2          | 2     |
| 3          | 3     |
| 4          | 4     |
| 5          | 5     |
| 6          | 6     |
| 7          | 7     |
| 8          | 8     |
| 9          | 9     |
| 10         | 10    |
| 11         | 11    |
| 12         | 12    |

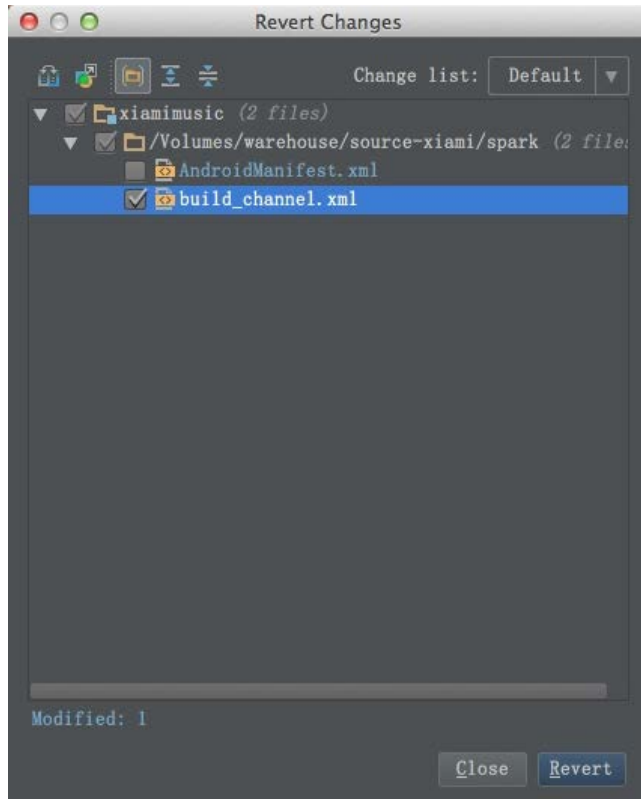
1 difference Deleted Changed Inserted

Version Control: Console File build\_channel.xml History

| Version | Date           | Author    | Commit Message |
|---------|----------------|-----------|----------------|
| f2514a2 | 13-7-15 下午5:21 | kezhaowei | initial import |

## 6. Revert

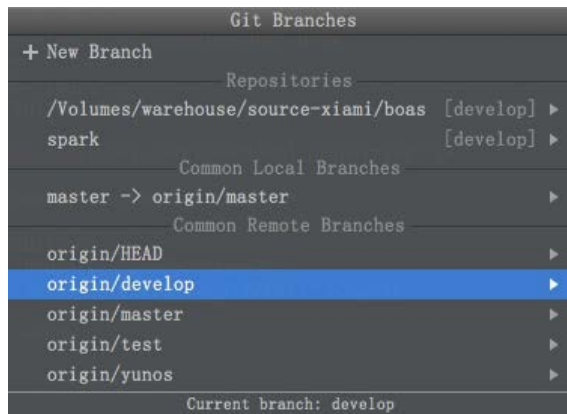
还原文件



## 7. Repository

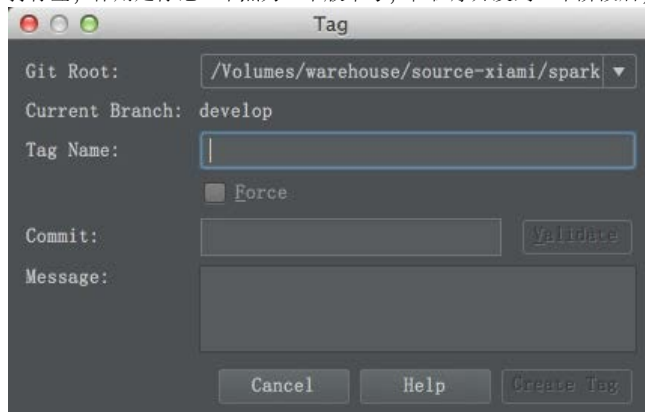
### 1. Branches

对Branches进行各种操作



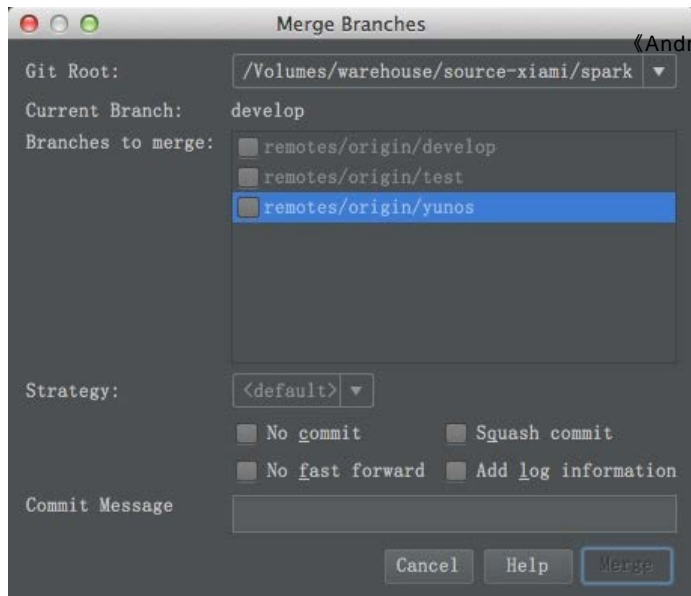
## 2. Tag

打标签，作用是标记一个点为一个版本号，在程序开发到一个阶段后，我们需要打个标签，发布一个版本。



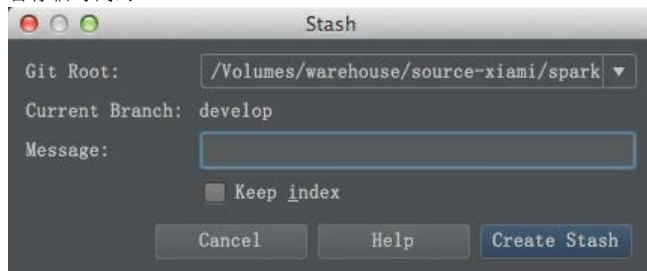
## 3. Merge Branches

合并分支



#### 4. Stash

暂存临时代码



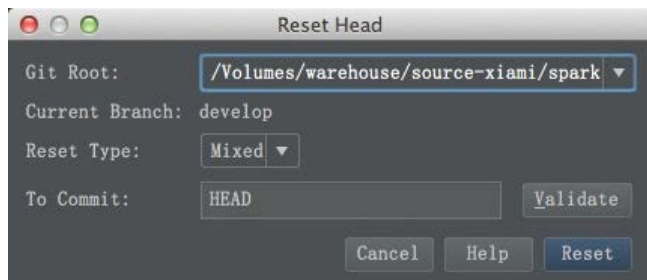
使用场景：

当你不想提交当前完成了一半的代码、但是却不得不修改一个紧急Bug，那么使用‘git stash’就可以将你当前未提交到本地(和服务器的)代码推入到Git的栈中，这时候你的工作区间和上一次提交的内容是完全一样的，所以你可以放心的修 Bug，等到修完Bug，提交到服务器上后，再使用‘git stash apply’将以前一半的工作应用回来。

#### 5. Reset Head

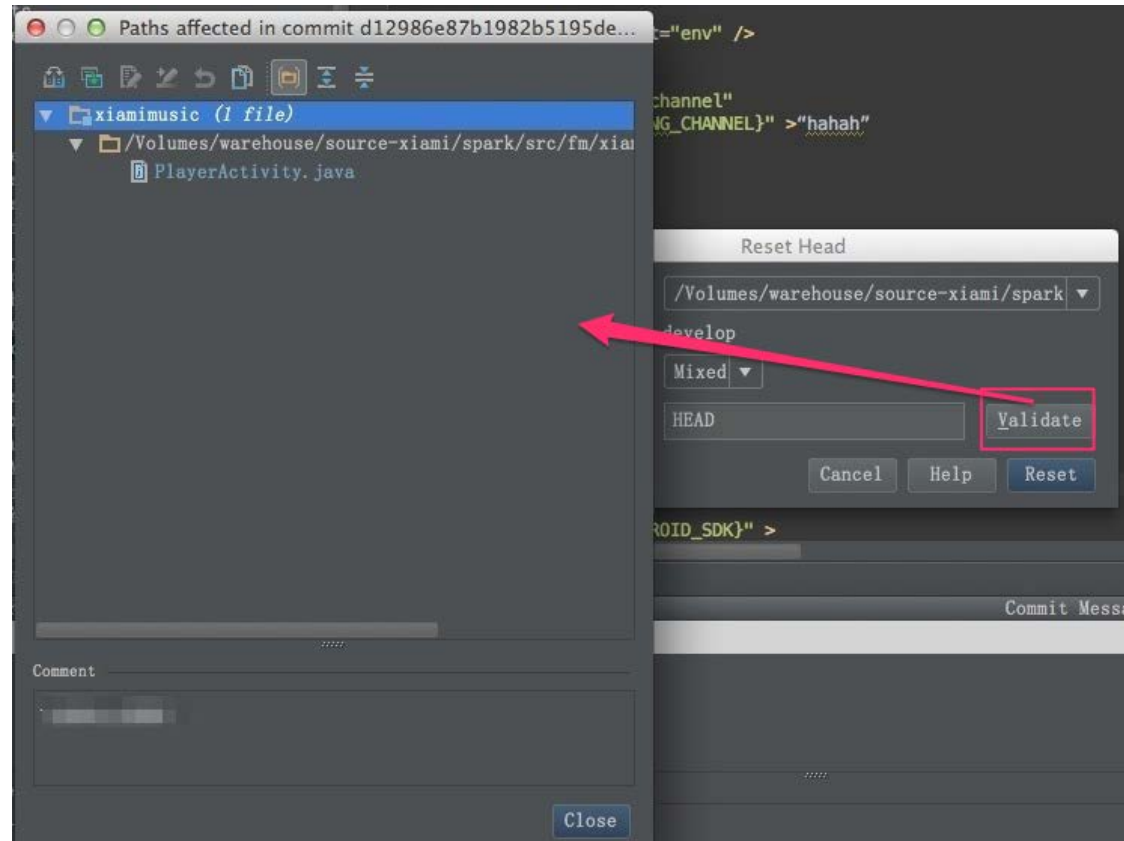
##### 1. Reset Head简介

将当前的分支重设为最新的一次提交

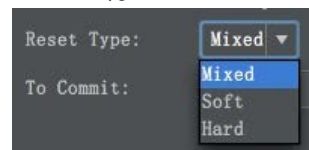


在git的一般使用中，如果发现错误的将不想staging的文件add进入index之后，想回退取消，则可以使用命令：`git reset HEAD <file>...`，同时git add完毕之后，git也会做相应的提示。HEAD 即 最新的一次提交  
《Android Studio入门指南》作者：毕小鹏 博客：<http://blog.csdn.net/wirelessqa>

Validate 可以验证最后一次提交



## 2. Reset Type 重设类型



hard: [reset index和working directory](#)

自从<commit>以来在working directory中的任何改变都被丢弃，并把HEAD指向<commit>。

soft: [index和working directory中的内容不作任何改变，仅仅把HEAD指向<commit>](#)

这个模式的效果是，执行完毕后，自从<commit>以来的所有改变都会显示在git status的"Changes to be committed"中。

mixed: [仅reset index，但是不reset working directory](#)

这个模式是默认模式，即当不显示告知git reset模式时，会使用mixed模式。

这个模式的效果是，working directory中文件的修改都会被保留，不会丢弃，但是也不会被标记成"Changes to be committed"，但是会打出什么还未被更新的报告。  
《Android Studio入门指南》作者：毕小鹏 博客：<http://blog.csdn.net/wirelessqa>

### 3. Git几个基本概念

#### 1. Working Directory（工作目录）

Git的工作目录是保存当前正在工作的文件所在的目录，在这个目录中的文件可能会在切换branch时被Git删除或者替换。  
这个目录是个临时目录，临时存储你从Git库中取出的文件，这些文件一直会被保存，直到下次提交。

#### 2. Git Directory（Git库目录）

项目的所有历史提交都被保存在了Git库目录中，只要你作回滚操作，它应该不会丢失。

#### 3. Git Index（Git索引）

Git index 可以看作是工作目录和Git库目录之间的暂存区，和staging area是相同的意思。  
可以使用Git index构建一组你准备一起提交的改变。

Git Index和Git Staging area是同一个意思，都是指已经被add的但尚未commit的那些内容所在的区域。  
最简单的查看目前什么内容在index中的方法是使用git status命令。

#### 4. 扩展阅读：

命令中“Changed but not updated”中所列的内容是在Working Directory中的内容，add之后将进入Index。

命令中“Untracked files”中所列的内容是尚未被Git跟踪的内容，add之后进入Index。

命令中“Changes to be committed”中所列的内容是在Index中的内容，commit之后进入Git Directory。

哪些操作能够改变git index中的内容？

A). **git add <path>...**会将working directory中的内容添加进入git index。

B). **git reset HEAD <path>...**会将git index中path内容删除，重新放回working directory中。

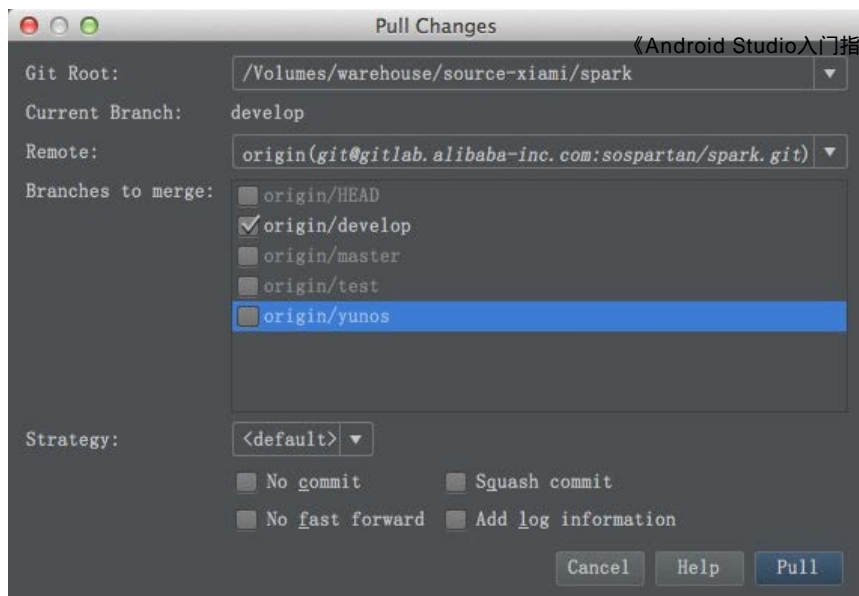
### 6. Fetch

git fetch：相当于从远程获取最新版本到本地，不会自动merge

#### 7. Pull

git pull：相当于从远程获取最新版本并merge到本地





《Android Studio入门指南》作者：毕小鹏 博客：<http://blog.csdn.net/wirelessqa>

## 8. Fetch VS Pull

举例说明：下载最新的代码并进行合并

**Fetch**命令：

`git fetch origin master`

`git log -p master..origin/master`

`git merge origin/master`

以上命令的含义：

- 1.首先从远程的origin的master主分支下载最新的版本到origin/master分支上
- 2.然后比较本地的master分支和origin/master分支的差别
- 3.最后进行合并

上述过程其实可以用以下更清晰的方式进行：

`git fetch origin master:tmp`

`git diff tmp`

`git merge tmp`

以上命令的含义：

从远程获取最新的版本到本地的test分支上

比较不同后再进行比较合并

**Pull**命令：

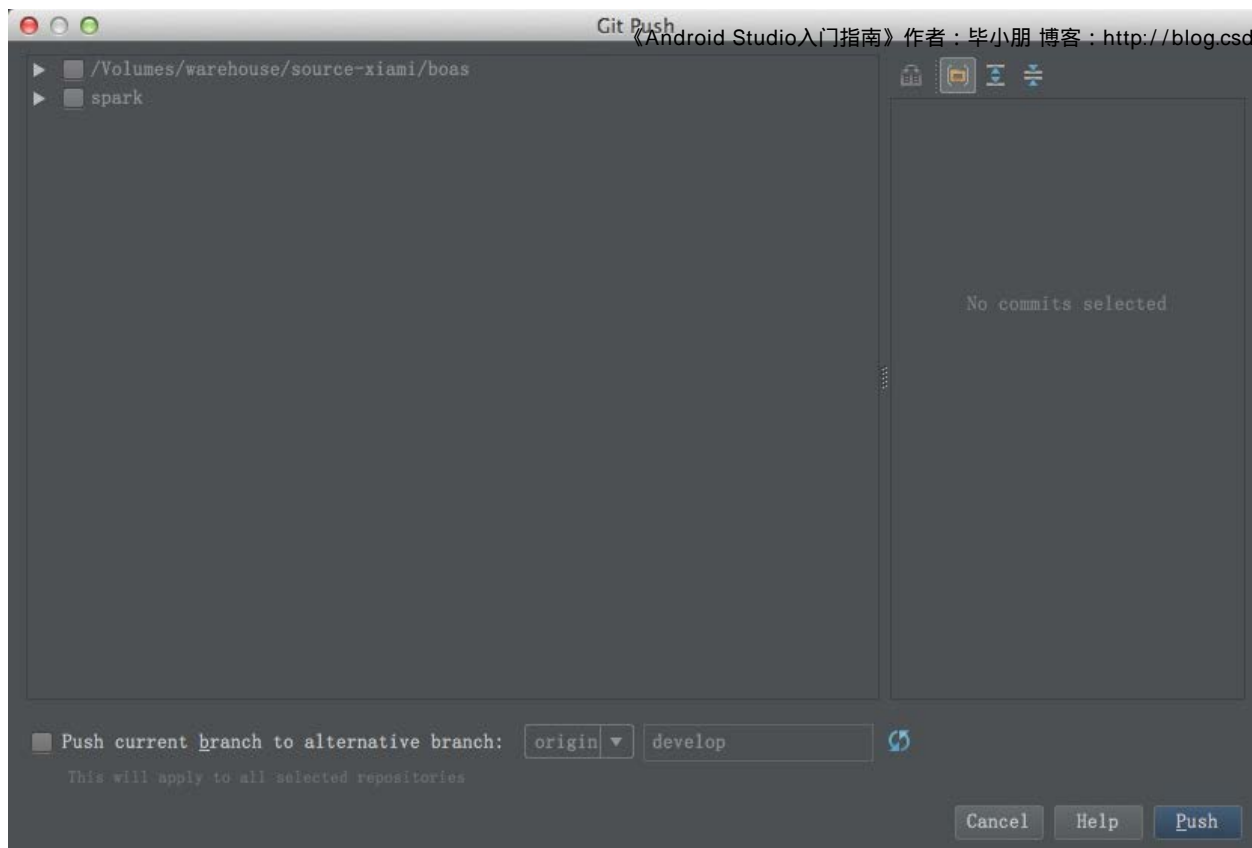
`git pull origin master`

上述命令其实相当于git fetch 和 git merge

在实际使用中，git fetch更安全一些，因为在merge前，我们可以查看更新情况，然后再决定是否合并

## 9. Push

通过git push把本地仓库的更新推到服务器仓库



扩展阅读：

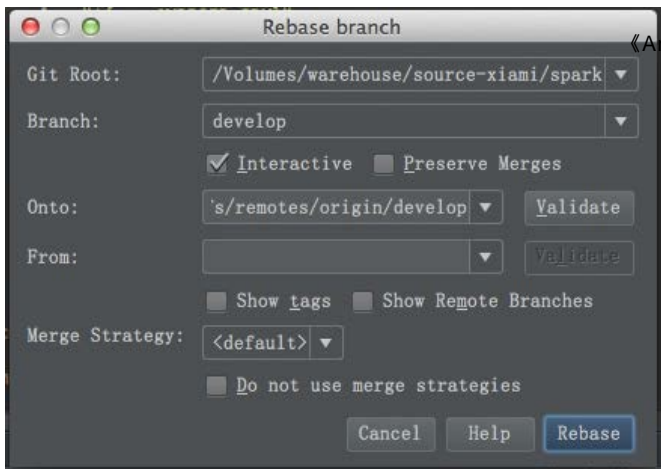
```
$ git push ssh://git@dev.lemote.com/rt4ls.git master // 把本地仓库提交到远程仓库的master分支中
$ git push origin test:master // 提交本地test分支作为远程的master分支
$ git push origin test:test // 提交本地test分支作为远程的test分支
```

如果想删除远程的分支呢？类似于上面，如果左边的分支为空，那么将删除右边的远程的分支。

```
$ git push origin :test // 刚提交到远程的test将被删除，但是本地还会保存的，不用担心
```

## 10. rebase

git rebase，顾名思义，就是重新定义（re）起点（base）的作用，即重新定义分支的版本库状态。



扩展阅读：

要搞清楚git rebase这个东西，要先看看版本库状态切换的两种情况：

- 1.在某个分支上，我们可以通过git reset，实现将当前分支切换到本分支以前的任何一个版本状态，即所谓的“回溯”。即实现了本分支的“后悔药”。也即版本控制系统的初衷。
2. 还有另一种情况，当我们的项目有多个分支的时候。我们除了在本地开发的时候可能会“回溯”外，也常常会将和自己并行开发的别人的分支修改添加到自己本地来。这种情况下很常见。作为项目管理员，肯定会不断的合并各个子项目的补丁，并将最新版本推送到公共版本库，而作为开发人员之一，提交自己的补丁之后，往往需要将自己的工作更新到最新的版本库，也就是说把别的分支的工作包含进来。

举个例子来说吧！假设我们的项目初期只有一个master分支，然后分支上作过两次提交。这个时候系统只有一个master分支，他的分支历史如下：

```
master0（初始化后的版本）  
  
||  
  
v  
  
master1（第一次提交后的版本）  
  
||  
  
v  
  
master2（第二次提交后的版本）
```

这个时候，我们可以通过git reset将master分支（工作目录、工作缓存或者是版本库）切换到master1或者master0版本，这就是前面所说的第一种情况。

假设我们这里把master分支通过git reset回溯到了master1状态。那么这个时候系统仍然只有一个master分支，分支的历史如下：

■

master0（初始化后的版本）

《Android Studio入门指南》作者：毕小鹏 博客：<http://blog.csdn.net/wirelessqa>

||

v

master1（第一次提交后的版本）

然后，我们在这里以master1为起点，创建了另一个分支test。那么对于test分支来说，他的第一个版本test0就和master1是同一个版本，此时项目的分支历史如下：

master0（初始化后的版本）

||

v

master1（第一次提交后的版本）==test0（test分支，初始化自master分支master1状态）

这个时候，我们分别对master分支、test分支作两次提交，此时版本库应该成了这个样子：

master0（初始化后的版本）

||

v

master1==test0==>test1==>test2

||

v

master2==>master3

1. 这个时候，通过第一种git reset的方式，可以将master分支的当前状态（master3）回溯到master分支的master0、master1、master2状态。也可已将test分支当前状态（test2）回溯到test分支的test0、test1状态，以及test分支的父分支master的master0、master1状态。
2. 那么。如果我要让test分支从test0到test2之间所有的改变都添加到master分支来，使得master分支包含test分支的所有修改。这个时候就要用到git rebase了。

首先，我们切换到master分支，然后运行下面的命令，即可实现我们的要求：git rebase test

这个时候，git做了些什么呢？

1. 先将test分支的代码checkout出来，作为工作目录
2. 然后将master分支从test分支创建起的所有改变的补丁，依次打上。如果打补丁的过程没问题，rebase就搞定了
3. 如果打补丁的时候出现了问题，就会提示你处理冲突。处理好了，可以运行git rebase -continue继续直到完成
4. 如果你不想处理，你还是有两个选择，一个是放弃rebase过程（运行git rebase -abort），另一个是直接取用test分支的取代当前分支的（git rebase -skip）。

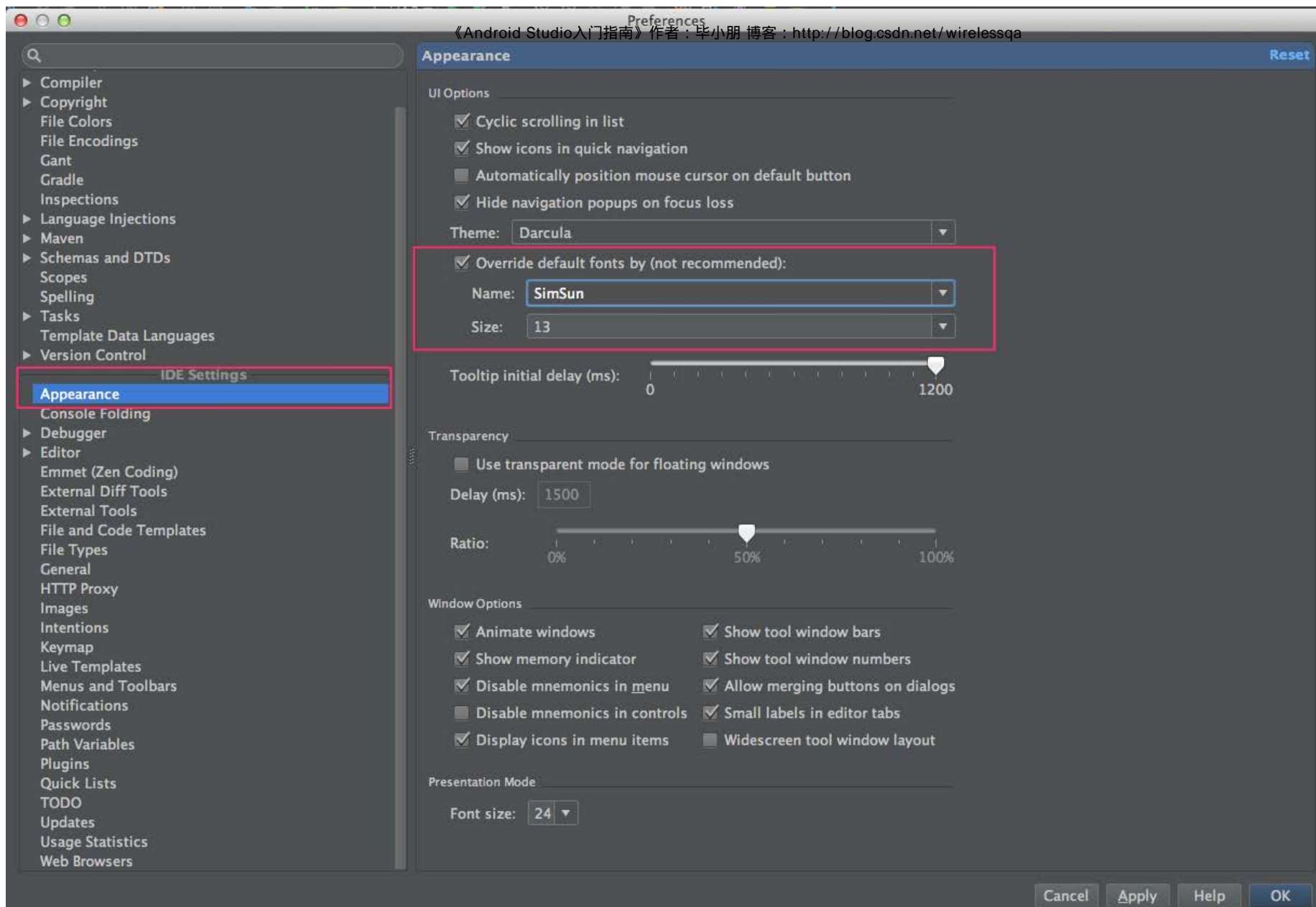
此外，rebase还能够让你修订以前提交，这个功能日后再说。

《Android Studio入门指南》作者：毕小鹏 博客：<http://blog.csdn.net/wirelessqa>

## 10. Android Studio 常用设置

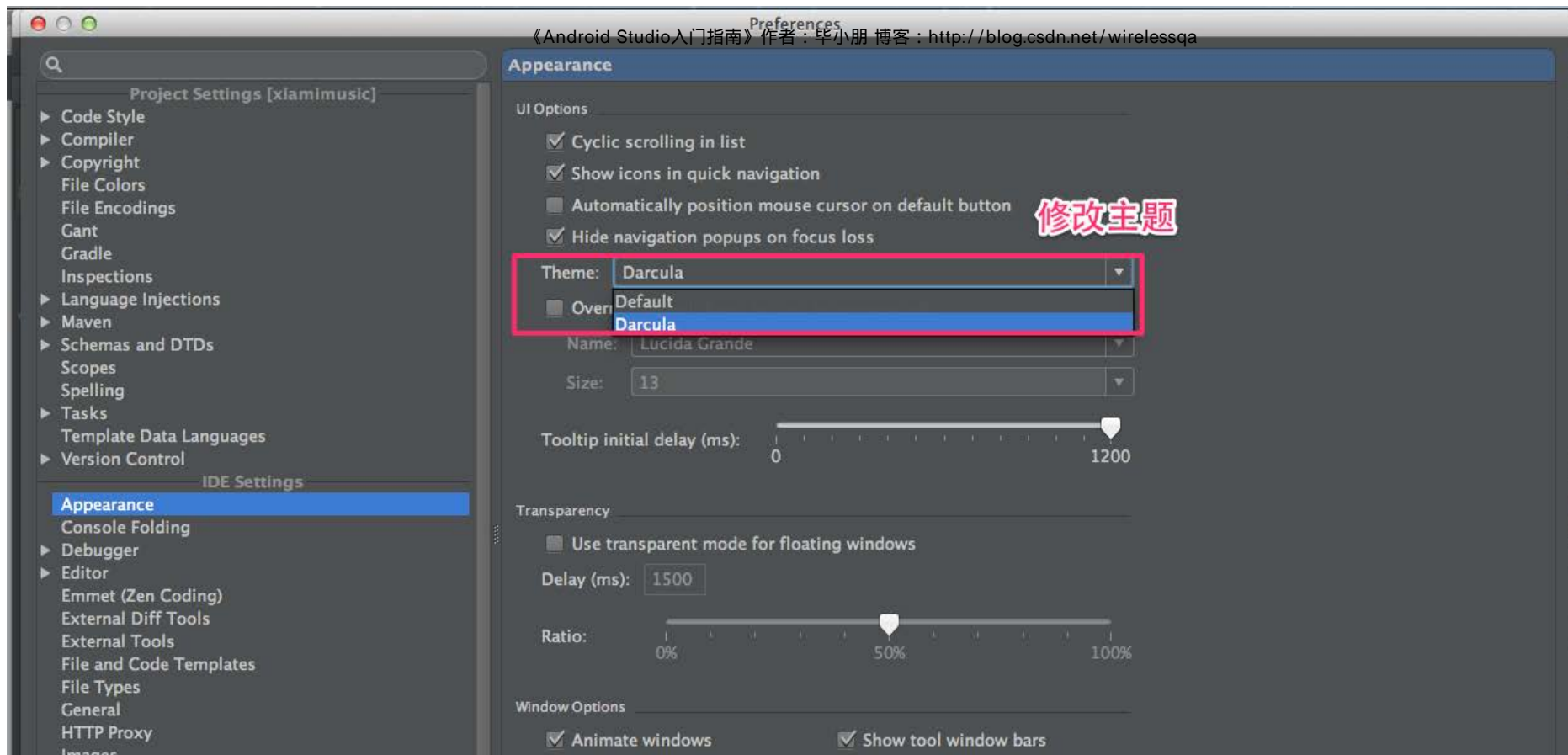
### 10.1 中文乱码

Android Studio -- > Preference -- > IDE Settings -- > Appearance，在右侧勾选上“Override default fonts by”，然后在第一个下拉框中选择字体为“simsun”，然后apply，重启IDE。



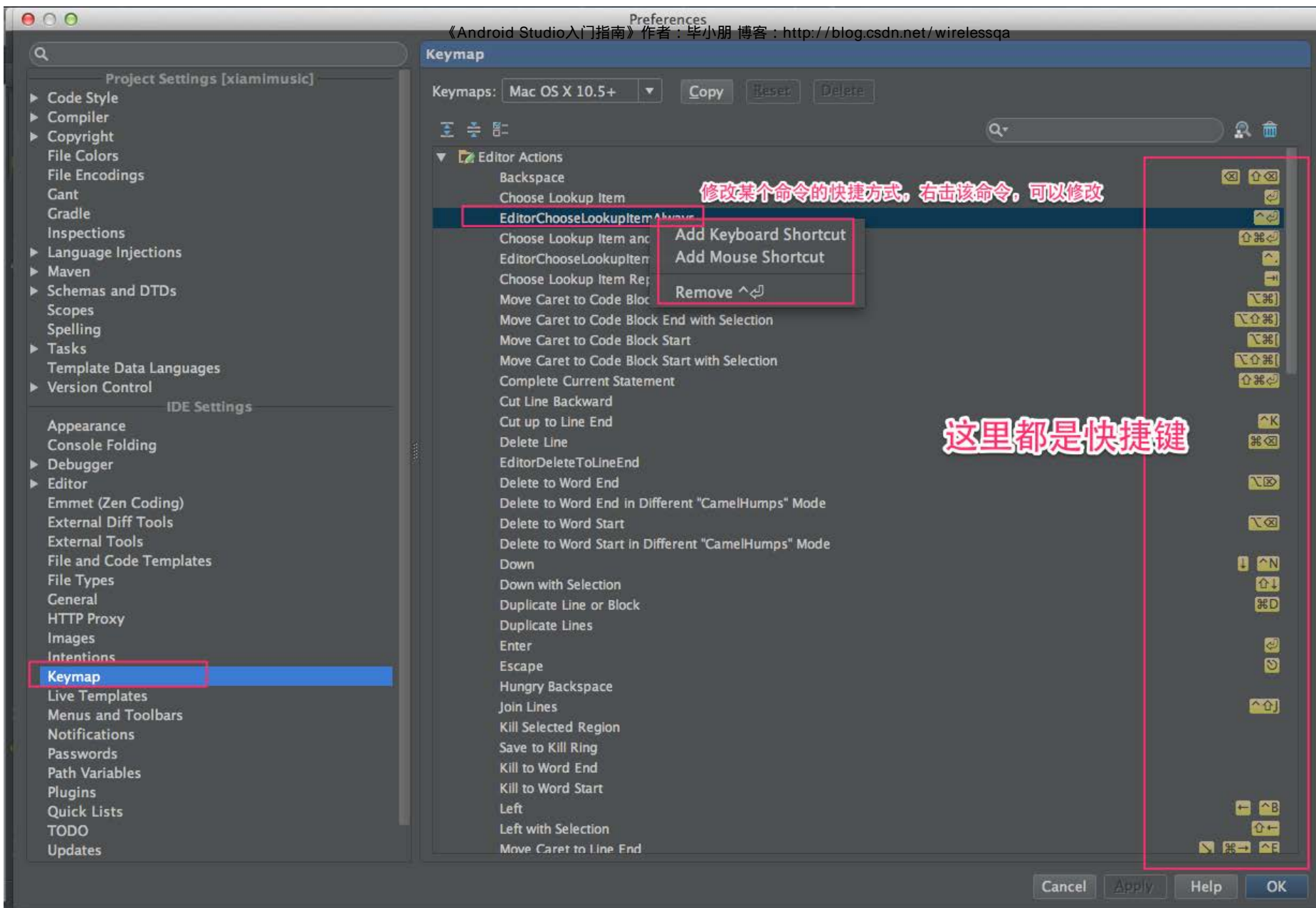
## 10.2 修改主题

Android Studio --> Preference --> IDE Settings --> Appearance --> Theme



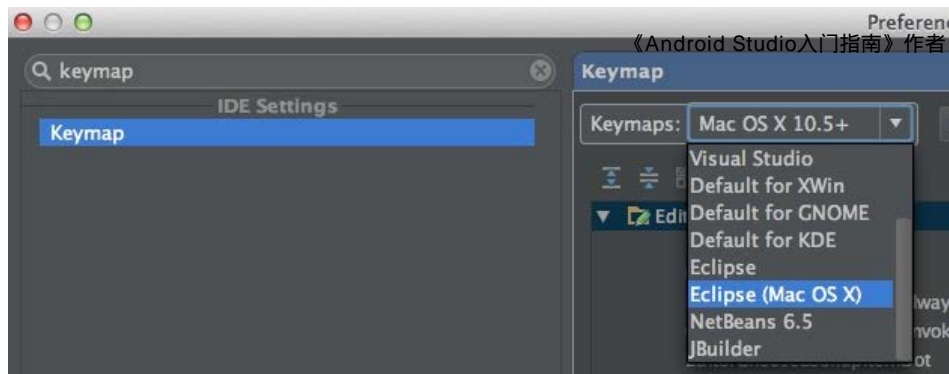
### 10.3. 更改常用操作的快捷键





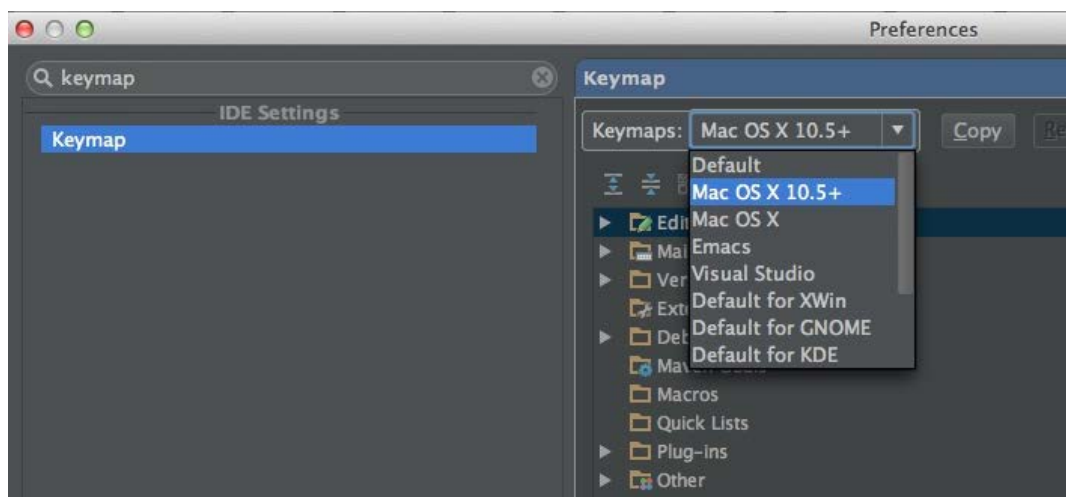
## 10.4 Android Studio设置成Eclipse快捷键

Android Studio > Preferences > Keymap>选择Eclipse



## 10.5. 常用快捷键

提示: mac用户先更改一下keymap,路径在Android Studio > Preferences > Keymap.



### 1. 常用编程快捷键

| 行为             | Android Studio快捷键                                      |
|----------------|--|
| 命令查找 (自动完成命令名) | Ctrl + Shift + A                                       |
| 项目快速修复         | ALT + ENTER  |
| 格式化代码          | CTRL + ALT + L (Win)<br>OPTION + CMD + L (Mac)         |
| 显示所选API的文档     | CTRL + Q (Win)<br>F1 (Mac)                             |
| 显示所选方法的参数      | CTRL + P   |
| 生成方法           | ALT + INSERT (Win)<br>CMD + N (Mac)                    |
| 跳转到源           | F4 (Win)<br>CMD + 下箭头 (Mac)                            |
| 删除一行           | CTRL + Y (Win)<br>CMD + 退格键 (Mac)                      |
| 按符号名称搜索        | CTRL + ALT + SHIFT + N (Win)<br>OPTION + CMD + O (Mac) |

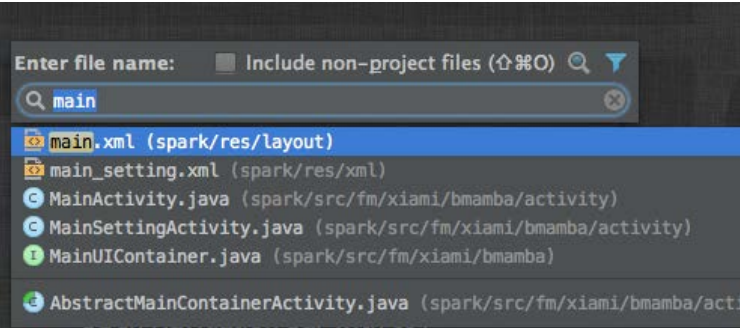
## 2. 常用项目和编辑器快捷键

《Android Studio入门指南》作者：毕小鹏 博客：<http://blog.csdn.net/wirelessqa>

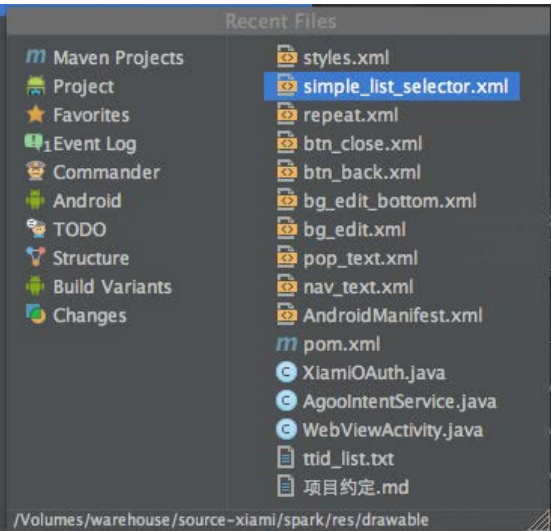
| 行为      | Android Studio 快捷键           |
|---------|------------------------------|
| 构建      | CTRL + F9 (Win)              |
|         | CMD + F9 (Mac)               |
| 构建并运行   | SHIFT + F10 (Win)            |
|         | Ctrl + R (Mac)               |
| 切换项目可视性 | ALT + 1 (Win)                |
|         | CMD + 1 (Mac)                |
| 导航打开的标签 | ALT + 左箭头; ALT + 右箭头 (Win)   |
|         | CTRL + 左箭头; CTRL + 右箭头 (Mac) |

## 3. 常用命令举例

### 1. 快速打开某个文件: **command+shift+o**



### 2. 最近打开文件: **command + E**



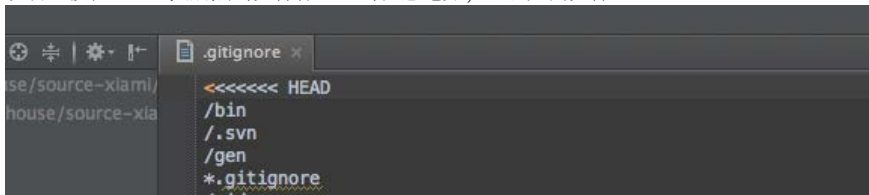
### 3. 将光标移动到导航栏: **command+ 向上的箭头**



《Android Studio入门指南》作者：毕小鹏 博客：<http://blog.csdn.net/wirelessqa>

#### 4.关闭文件: **shift + click**

说明：按住**shift** 然后点击文件标签上的任意地方，可以关闭文件



## 11. 联系作者

此版本为入门指导，写的不是很详细，也可能有错误的地方，如果您发现了任何错误或者您有一些好的技巧，联系我

作者：毕小鹏

博客：<http://blog.csdn.net/wirelessqa>