# 下载安装

1、下载地址

<http://androiddevtools.cn/>

<http://android-mirror.bugly.qq.com:8080/include/usage.html>

注意：如果已经安装了SDK，其实只需要安装AndroidStudio即可，SDK和AndroidStudio是可以分开下载安装的，但是尽量保证SDK是最新的。

2、SDK更新问题

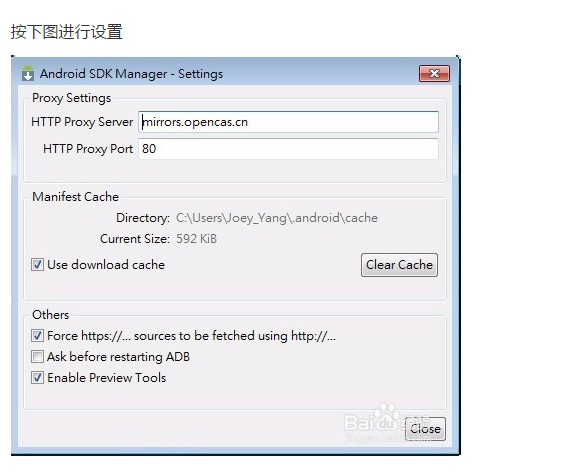
由于天朝对Google访问的限制，需要设置代理才能更新SDK.

设置方法如下：

打开SDK....>options...>ProxySetting添加代理

mirrors.opencas.cn

80



1. 安装完AndroidStudio后，首次启动会弹出检查SDK组件等设置，点击finish会去下载SDK等，如果没有设置代理的情况下，这个界面会卡很久，所以首次启动需要设置不需要检查SDK。设置方法如下：

关闭AndroidStudio，打开AndroidStudio安装目录下的bin文件夹，找到idea.properties并打开，添加:

disable.android.first.run=true

重启AndroidStudio就不会再出现SDK设置检查界面了。

# 常用操作篇

一、创建Project、Module

二、删除Module

删除Module需要两步：

1、从project中移除该Module

在要删除的Module上右键,然后选择Open Module Setting,然后选择要删除的Module，点击左上角的“-”删除。



1. 在Module上再次右键，就会出现delete，这个时候就可以删除了
2. 切换Project

File--->close project 关闭当前project，回到欢迎主界面，选择其他project

File--->open recent 打开最近打开过的project

File--->open 在新窗口中打开新的project

**快捷键大全**

这个网上有很多资料，而且在设置里面可以选择自己习惯的Keymap，比如eclipse，这里点到为止。

列举几个参考链接：

<http://www.cnblogs.com/0616--ataozhijia/p/3870064.html>

**将keymap设置为eclipse后的快捷键：**

F2：查看类说明

F4:查看类继承关系

Ctrl+E:查看最近打开过的文件

Ctrl+Shift+E:查看最近编辑过的文件

Ctrl+G :查看方法或类的调用

Ctrl+H：全局字符内容查找

double Shift ：全局文件查找，只能查找文件,比如打开源码文件

Ctrl+Shift+R:快速定位到你想打开的文件,只能是工程里面的文件,无法打开系统源码文件

Alt+Shift+R:重命名类、方法、变量.

Ctrl+F:当前文件内搜索内容,Ctrl+K查找下一个，Ctrl+Shift+K查找上一个

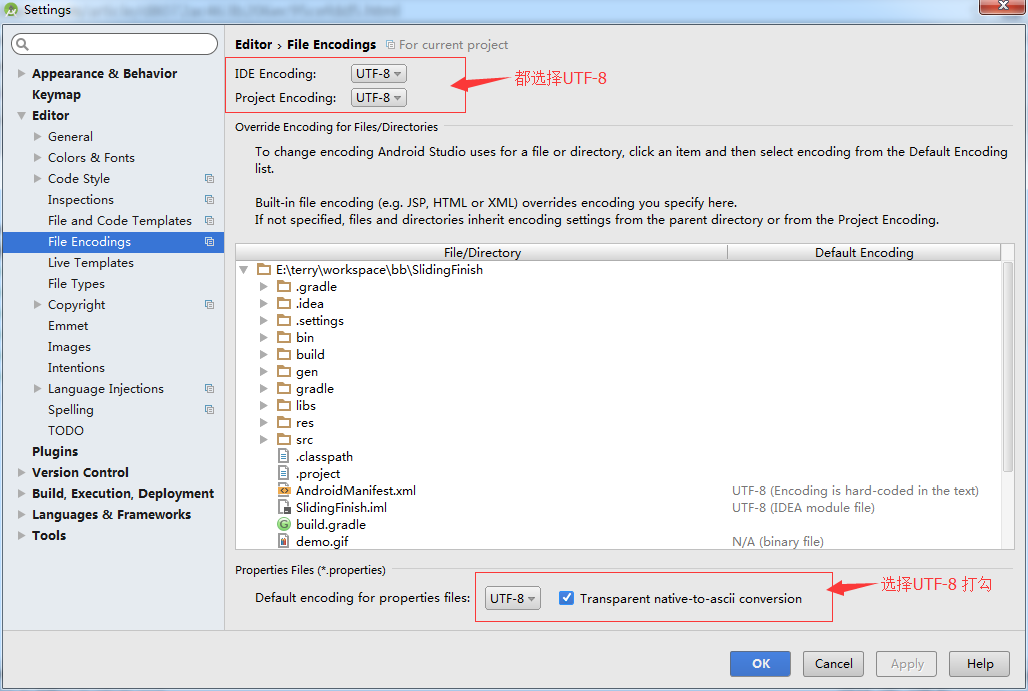
Ctrl+C:复制光标所在行代码到下一行

**修改快捷键**

<http://jingyan.baidu.com/article/90808022c416d9fd91c80f98.html>

**设置字符编码**

File----->Settings...>Editor--->File Encodings--选择utf-8



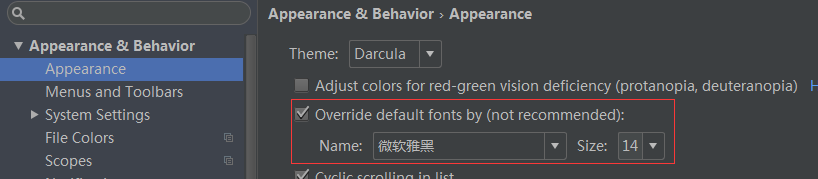
**显示行号**

Settings--->Editor---->General---->Appearance--->Show line number

**修改字体大小**

Settings---->Editor----->Colors&Fonts--->Font.....>Save As另存一份，然后才能编辑

**修改侧边栏字体大小**



**设置界面风格**

Settings---->Appearance&Behavior---->Theme-->Darcula或IntelliJ

**创建代码模板**

Settings---->Editor----->LiveTemplates

AndroidLog：logd、logi、loge....

output：sout、serr....

自定义模板：<http://my.oschina.net/djone/blog/493865>

<http://www.jb51.net/article/69923.htm>

**添加代码注释**

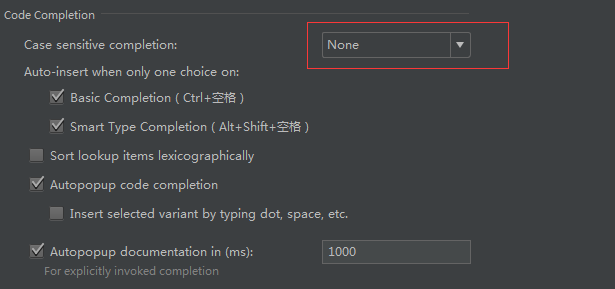
Settings---->Editor---->File and Code Templates-->选择Includes,编辑FileHeader

**错误修正提示列表**（Eclipse中的Quick Fix，Ctrl+1）  
Keymap->Other->Show Intention Action：默认是Alt+Enter

**智能提示首字母匹配**

如String和string，默认情况输入首字母s是不会匹配String的。

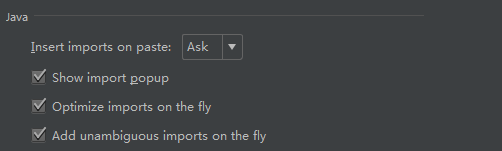
Settings->IDE Settings->Editor->General->Code Completion->Case sensitive completion->设置为None



**设置自动导包**

位置: Files ->Settings -> Editor -> General -> Auto Import

选择: Add unambiguous imports on the fly, 即可.



**格式化代码** Ctrl+Alt+L或 Ctrl+Shift+F

File-->Setting-->Keymap-->Reformat Code

或者在Keymap右上的搜索栏搜 Reformat

**导入和导出设置**

当你设置了大量快捷键和配置后，可保存当前设置，以后方便使用，

导出设置：File---->export settings--->选择你想导出的模块

导入设置：File---->import settings--->选择你想导入的模块

**添加assets目录**

assets目录位于src/main，与java平级

**将Studio的Module设置为library**

1. 新建Module的时候可以选择以Android Library的方式新建。
2. 如果是要修改已有的Module为library，则需要修改Module的build.gradle

把 apply plugin:’com.android.application’修改为

apply plugin:’com.android.library’

然后去掉applicationId (一个library不需要这个)

**多工程引用同一个library项目配置方法**

http://www.tuicool.com/articles/IjMVVrF

**添加jar包**

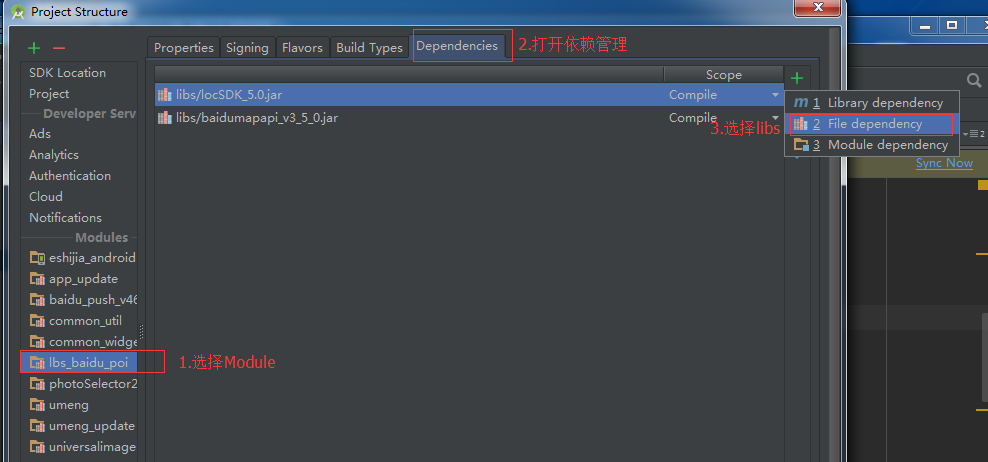
首先将jar包拷贝到libs目录下

方式1

在该jar包上右键，然后add as library ，在弹出框中选择Module

方式2

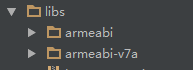
打开Project Structure



**添加so文件**

方式1

将so文件目录拷贝到libs目录下

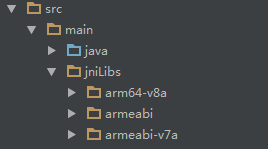


打开Module的build.gradle文件，添加

|  |
| --- |
| sourceSets {  main {  jniLibs.srcDir 'libs'  }  } |

方式2

在项目 src/main目录下创建一个名叫jniLibs文件夹。将对应的so文件拷贝到该目录下



**Studio编译NDK**

**项目签名打包**

**项目运行、Debug**

**多渠道打包**

主要是配置build.gradle

# Eclipse项目导入到AndroidStudio中

Google为了让我们更加方便的使用AndroidStudio，提供了两种导入Eclipse代码的方式：一种兼容Eclipse，一种是全新的Android Gradle Project.

一、兼容模式，即Eclipse和Studio都可以开发

1）eclipse：文件file --》 导出 export

2）弹出窗口里面选择android-->>generate gradle build files

3）导出选择你想导出的工程就行了，点击finish。

4）此时会在你的那个项目主目录下面生成一个build.gradle文件，这个就是studio所需要的配置文件。

5）打开studio，import project(作为project导入)或者import module(作为module导入)。

6）选择那个生成的build.gradle，点击ok。

注意，在导入后可能会遇到无法运行项目的问题，一般是gradle版本问题造成的，解决办法：

1、修改build.gradle

在studio中打开项目的build.gradle,修改dependencies的值为当前Studio的gradle版本

dependencies {  
 classpath **'com.android.tools.build:gradle:2.0.0-alpha5'**}

2.0.0版本是Studio2.0对应的gradle版本，请使用你自己Studio的gradle版本

2、修改gradle-wrapper.properties

在studio中打开项目gradle/gradle-wrapper.properties文件，修改

**distributionUrl**=**https\://services.gradle.org/distributions/gradle-2.10-all.zip**

为当前studio的gradle版本

具体可参考：

<http://www.open-open.com/lib/view/open1419597450031.html>

二、 以 Android Gradle Project方式导入到Studio

1.如果只有一个项目，则直接选择Eclipse工作空间中的项目导入即可

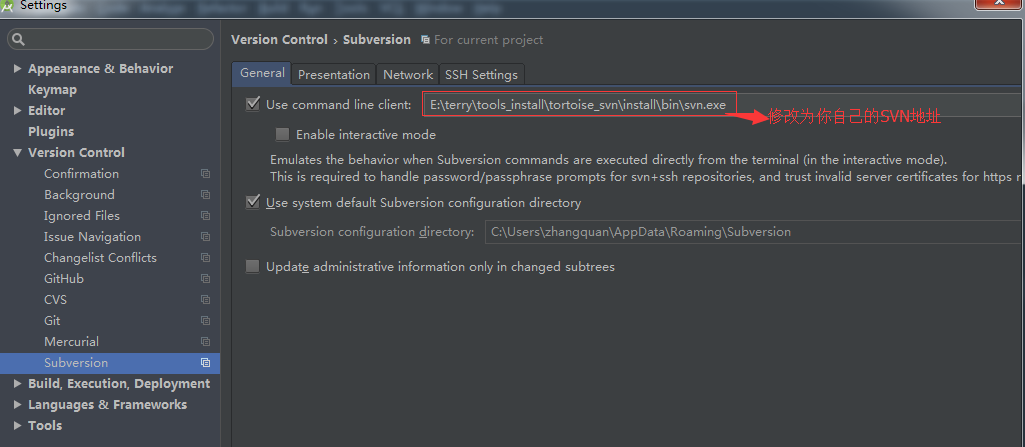
2.如果项目含有库工程，在Studio中选择Eclipse 工作空间中的主工程导入。

# 版本控制

## Git

## SVN

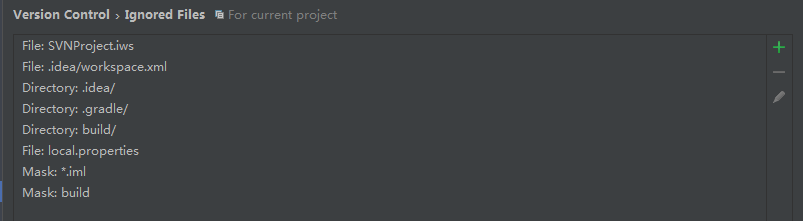
配置SVN



1. 共享项目到SVN服务器

在共享项目到SVN服务器之前，请配置要忽略的文件,一般需要忽略的文件有：  
1、.idea文件夹  
2、.gradle文件夹  
3、所有的build文件夹  
4、所有的.iml文件  
5、local.properties文件。

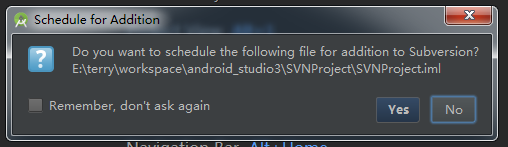
忽略文件设置：



1. checkout检出项目

从Studio中File-->close project 回到欢迎页面，

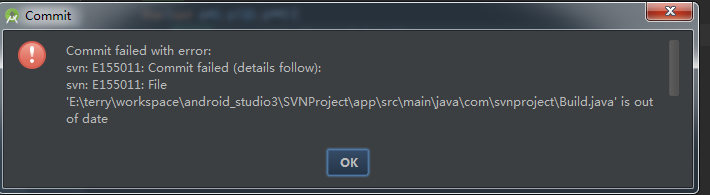
检出项目时会提示你是否将\*.iml文件加入到SVN控制中，请选择NO



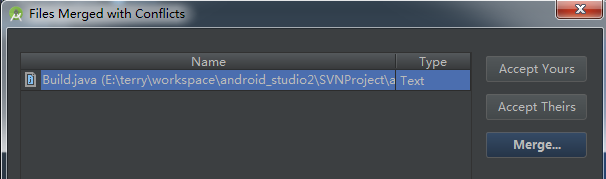
三、解决项目冲突

提交文件出现冲突时，会提示你某某文件is out of date 过时了，则先update，解决冲突后再commit

文件过时提示



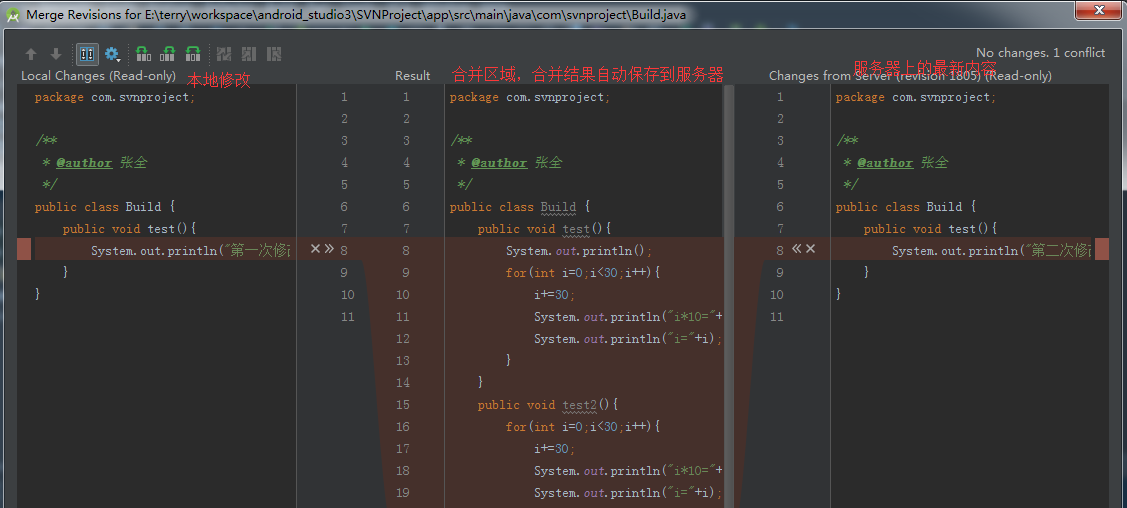
update提示解决冲突



Accept Yours：替换成你的本地文件,即以你的文件为准

Accept Theirs：替换成服务器上的文件，即以服务器上的最新文件为准

Merge：合并代码，将服务器上的文件和本地文件进行整合,



四、断开与SVN的关系

1、将.idea构建目录下的vcs.xml中的

<mapping directory="" vcs="svn" />

 还原回

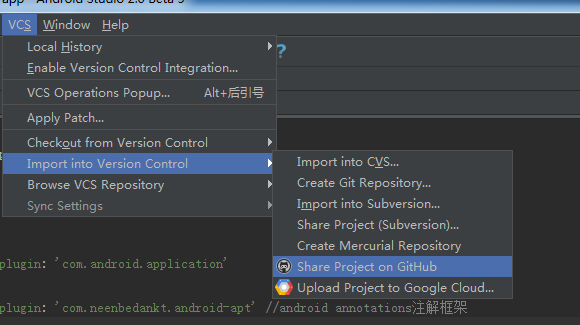
<mapping directory="" vcs="" />

 2、删除工程目录下的.svn目录

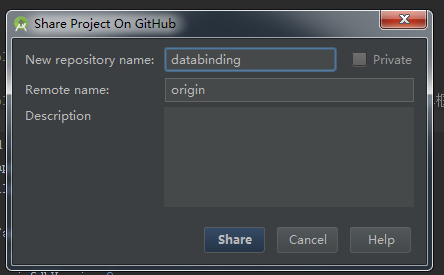
## Github

参考 ： <http://www.th7.cn/Program/Android/201511/704911.shtml>

1. 提交项目到Github

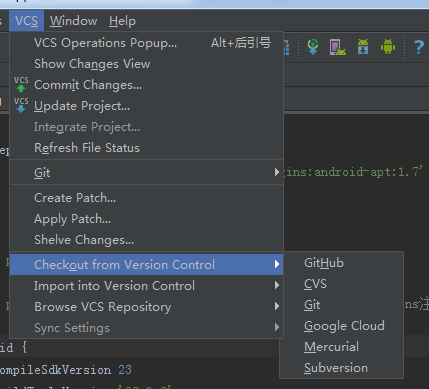


填写respository name ，点击share



如果出现失败，请注意检查respository名称是否正确，注意不能和其他人的respository名称相同。

1. 从github 检出项目



# 优化编译速度

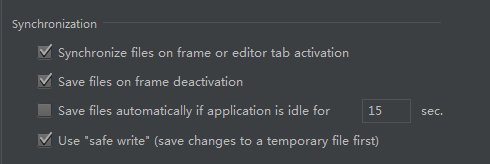
1、C:\Users\zhangquan\.gradle 中增加gradle.properties

org.gradle.jvmargs=-Xmx2048m -XX:MaxPermSize=512m -XX:+HeapDumpOnOutOfMemoryError -Dfile.encoding=UTF-8

org.gradle.daemon=true

project.android.dexOptions.preDexLibraries=false

2、Studio中，File----Settings--->Appearance&Behavior--->System Settings 同步优化



# 常用插件

插件安装

Settings - Plugins - Browser repositories - 弹出对话框 - 选中某一插件 - 右边Install plugin。

常用插件

http://www.jianshu.com/p/6f5f818afe4b?hmsr=toutiao.io&amp;utm\_medium=toutiao.io&amp;utm\_source=toutiao.io

# 创建JAVA工程

new module---->选择Java Library

# Gradle命令

一些命令如 ./gradlew -v ./gradlew clean ./gradlew build, 这里注意是./gradlew, ./代表当前目录，gradlew代表 gradle wrapper，意思是gradle的一层包装，大家可以理解为在这个项目本地就封装了gradle，即gradle wrapper， 在你的{Project}/gradle/wrapper/gralde-wrapper.properties文件中声明了它指向的目录和版本。只要下载成功即可用grdlew wrapper的命令代替全局的gradle命令。

理解了gradle wrapper的概念，下面一些常用命令也就容易理解了。

./gradlew -v 版本号

./gradlew clean 清除{Project}/app目录下的build文件夹

./gradlew build 检查依赖并编译打包

这里注意的是 ./gradlew build 命令把debug、release环境的包都打出来，如果正式发布只需要打Release的包，该怎么办呢，下面介绍一个很有用的命令 \*\*assemble\*\*, 如

./gradlew assembleDebug 编译并打Debug包

./gradlew assembleRelease 编译并打Release的包

除此之外，assemble还可以和productFlavors结合使用

./gradlew installRelease Release模式打包并安装

./gradlew uninstallRelease 卸载Release模式包

# 扩展知识

1、applicationId和PackageName

参考：<http://blog.csdn.net/maosidiaoxian/article/details/41719357>

简单来说，applicationId用来标示app的唯一性，packageName用来命名资源类的包，比如R类,将二者进行区分，使得修改applicationId就可以发布不同的app版本，而不用去重新引入R类和修改Java源文件.