# Master Password

如果是第一次使用Github检出代码到AS，会要求输入Master Password，这个密码是用来保护存储在AS数据库中的密码，也就是说有一些密码保存在了AS的数据库中，比如签名的密码，而Master Password这个密码则用于访问这个数据库。

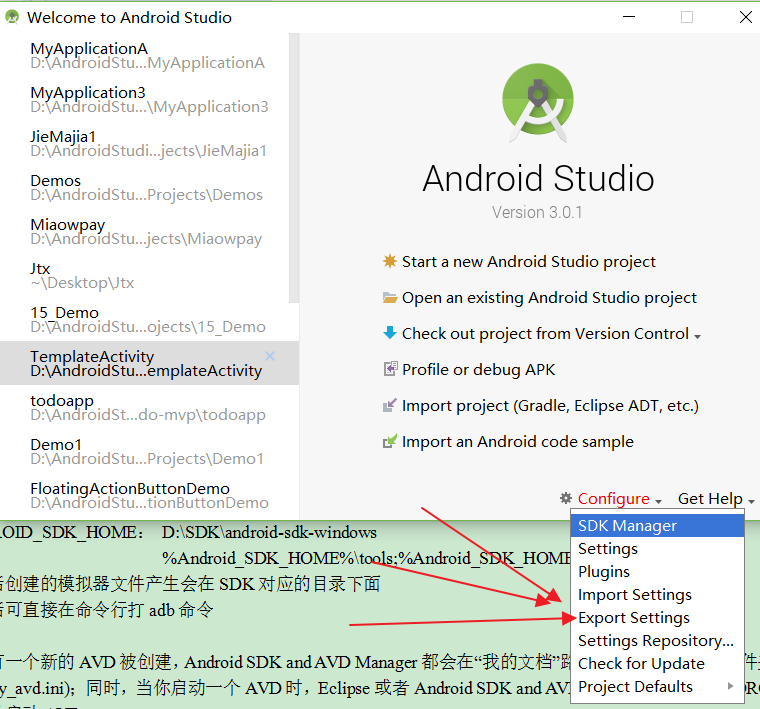
# 第1章 初识AndroidStudio

2013年5月16日Google在I/O大会上发布AndroidStudio的第1个预览版本

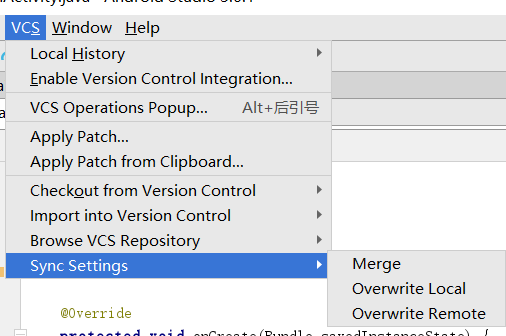
谷歌至2015年底已停止对Eclipse Android开发工具的一切支持，包括ADT插件、Ant构建系统、DDMS、Traceview与其他性能和监控工具。

Android 6.0 Marshmallow或者更高的系统映像支持对称式多重处理，对模拟器和ADB进行了重大的改进，当使用ADB推送文件时速度可以比真机快五倍。

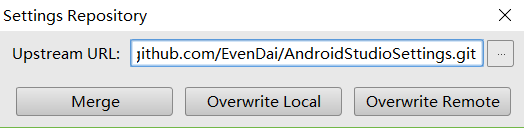
## 设置仓库



也可在菜单中选择设置仓库，如下：

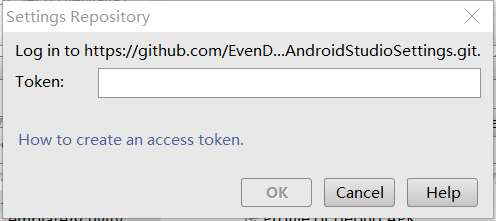


Settings Repository（设置仓库）使用GitHub/Gitlab来存储设置，可以共享 设置文件给所有基于JetBrains平台的产品（如AndroidStudio、Intellij IDEA、Pycharm等），打开这个设置窗口如下：

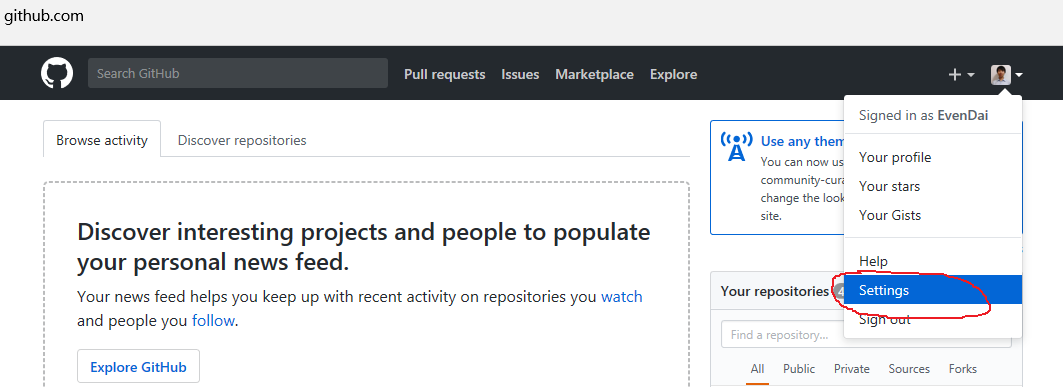


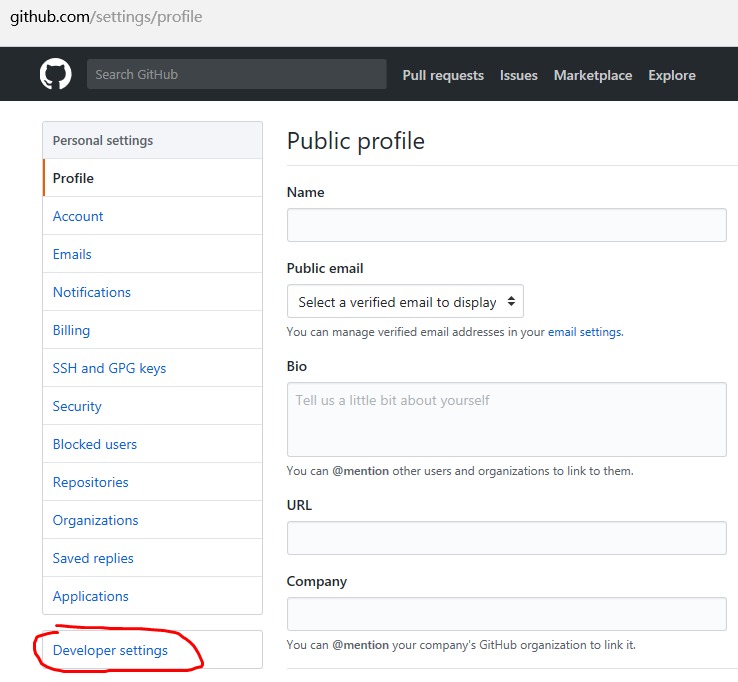
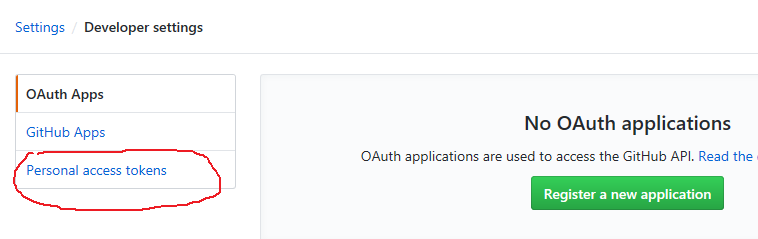
在Upstream URL中输入用于共享 设置的Git仓库的地址。当需要同步的时候可以选择本地的覆盖远程（Overwrite Remote），或远程的覆盖本地（Overwrite Local），也可以合并本地和远程的设置（Merge）。

点击任意的一个同步后需要输入一个令牌，如下：

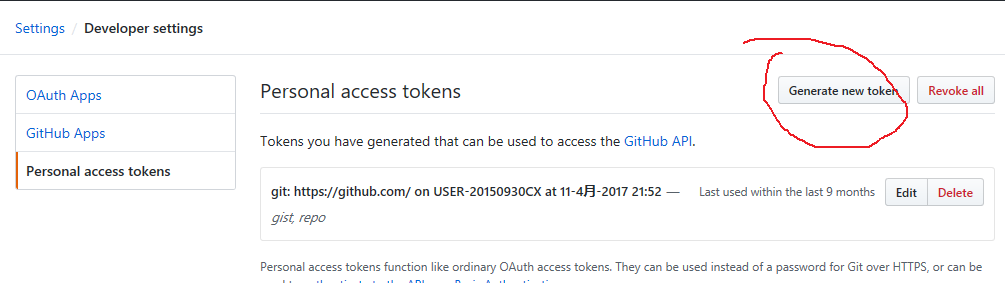


可以点击“How to create an access token”来查看如何创建一个令牌，具体网址是：https://help.github.com/articles/creating-a-personal-access-token-for-the-command-line/，创建令牌的步骤如下：

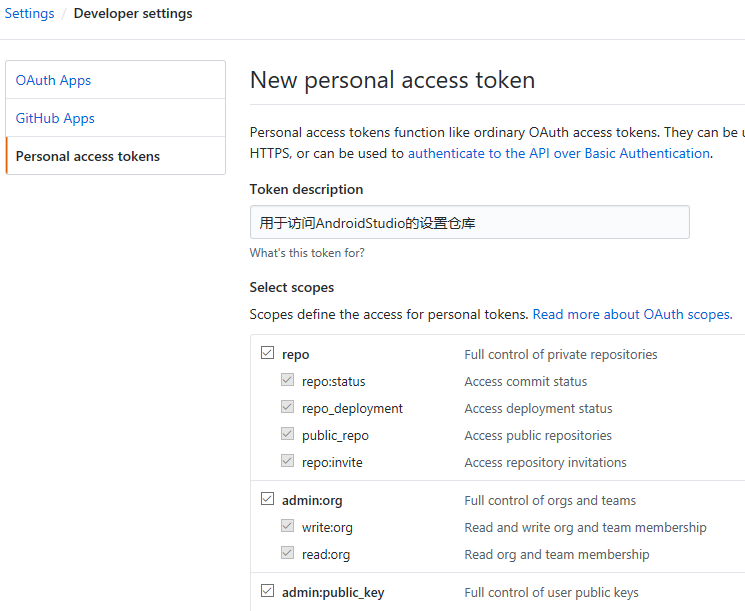
点击Github右上角个人头像，出现下拉列表，选择Settings  
 

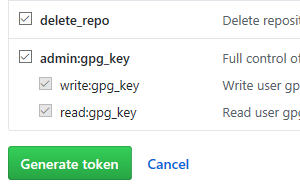
点击Settings后出现如下界面：  
   
 点击 Developer settings后出现如下界面：  
 

再点击 “Personal access tokens”，如下：

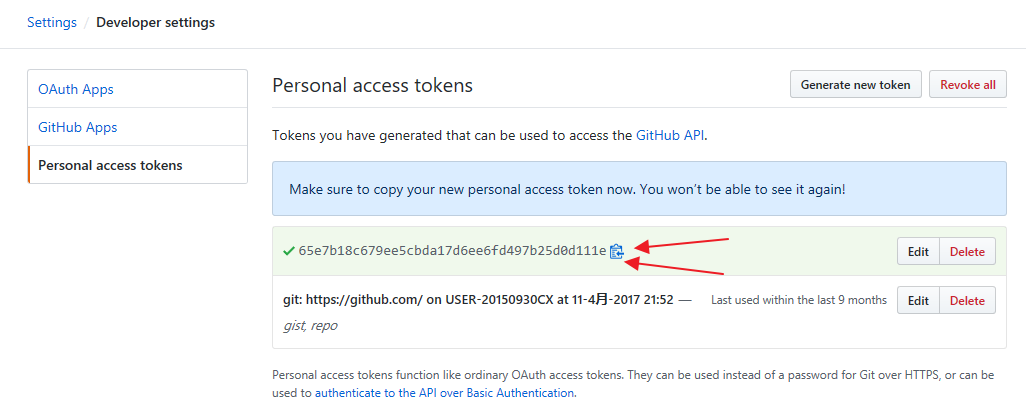


点击 “Generate new token”后如下：





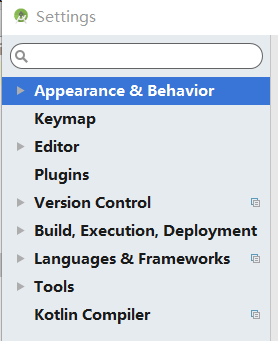
把勾全打上，这是代表选择这个token所拥有的权限，然后点击最下方的Generate token按钮，这时出现新界面如下：

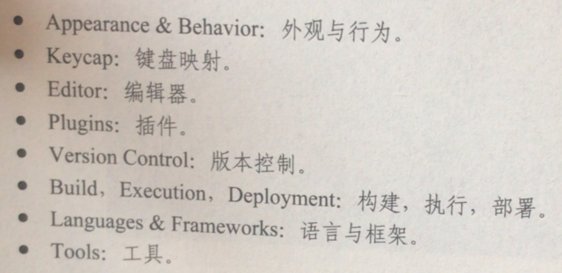


点击 Copy按钮将令牌复制到您的剪贴板。出于安全原因，在离开页面之后，您将无法再看到该令牌。

## 设置

，如下：

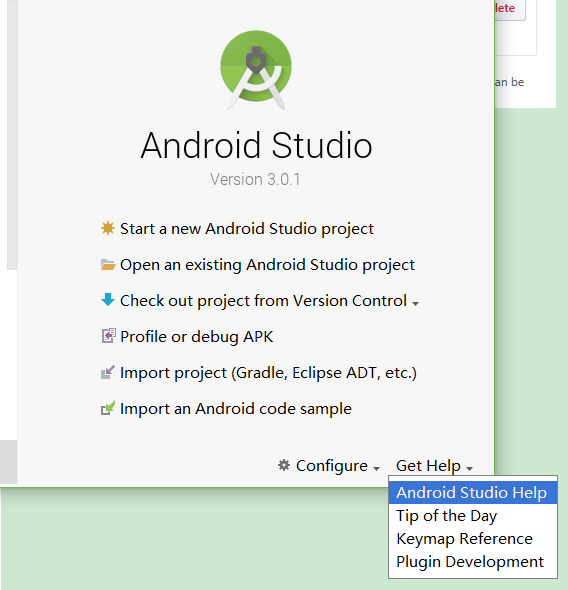




## 默认设置



## 帮助和教程



第一个是官方的帮助文档，介绍如何使用AndroidStudio

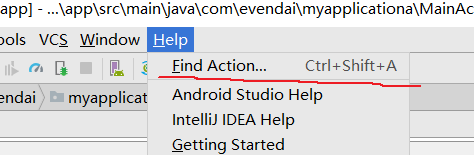
第二个是用于每日提示，显示一些AS的使用技巧

第三个是显示所有的快捷键

第四个是插件开发，如果想为AS开发插件，可以考虑这个网站

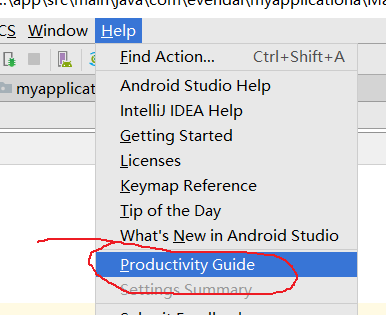
## 全局查找操作

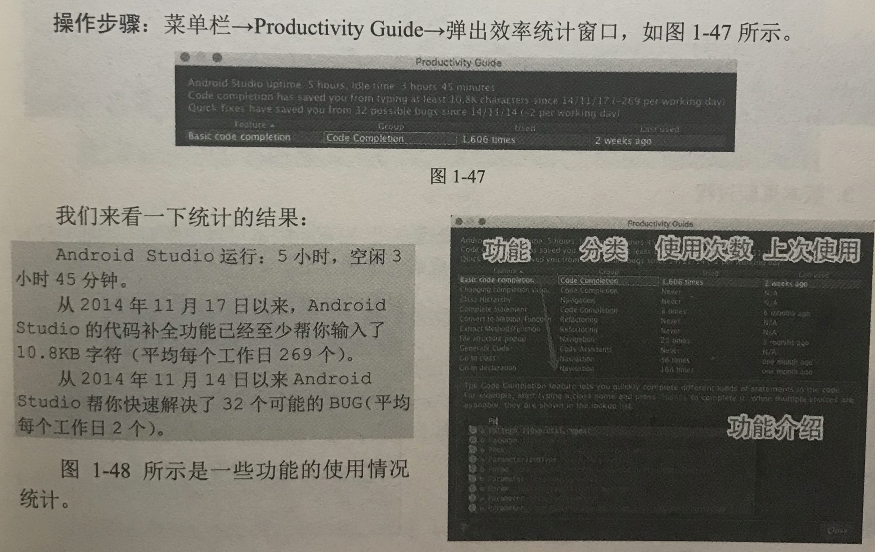
如果忘记某个菜单项在哪时，可以通过搜索来查找，如下：



Find Actin会搜索AS中所有能匹配的操作，包括菜单项、设置、工具等，对于没有设置快捷键或者忘记快捷键的操作通过输入名字可快速调用。

## 效率指南





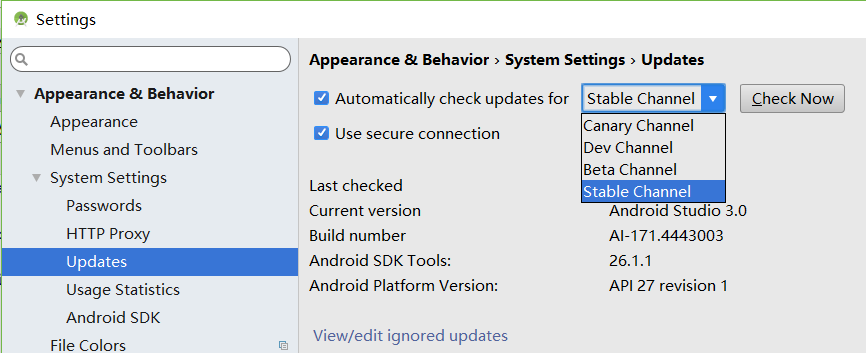
## 查看AS版本和使用的JDK版本

Help-->About



## 配置AS更新通道

File-->Settings-->Appearance & Behavior-->System Settings-->Updates，如下：



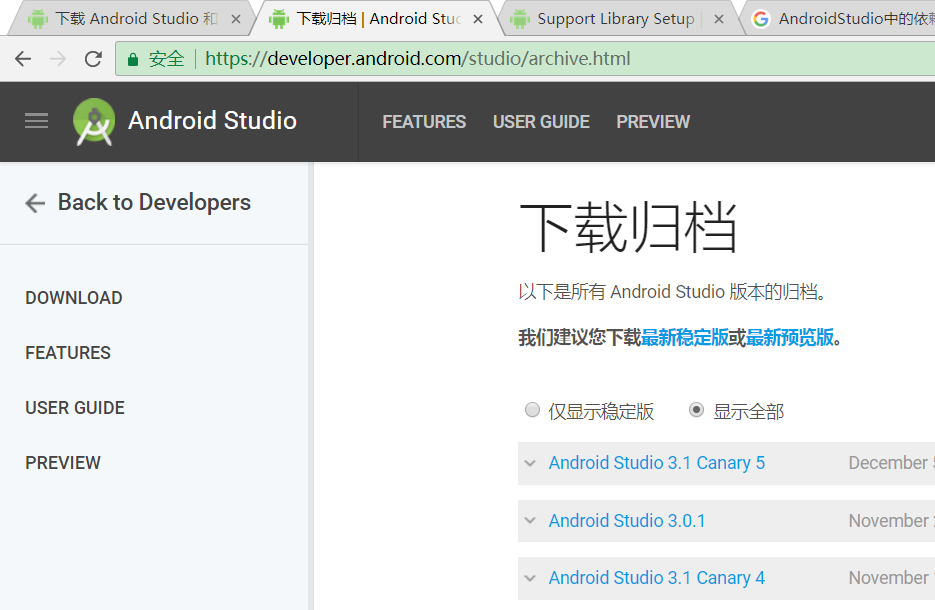
4种版本的区别：

* Stable Channel 稳定版本
* Beta Chnanel 测试版本
* Dev Channel 开发版本
* Canary Channel 金丝雀版本

这4种版本更新频率从Canary往上逐渐递减：Cannary大概1或2周更新一次，Beta则为相对稳定的发布版本，Stable则是正式版本。稳定性能与其更新频率相反。如果不是特别需要最好是使用稳定版本，如果想尝鲜可使用Canary Channel版本，可通过这个网站获取：https://developer.android.com/studio/preview/index.html

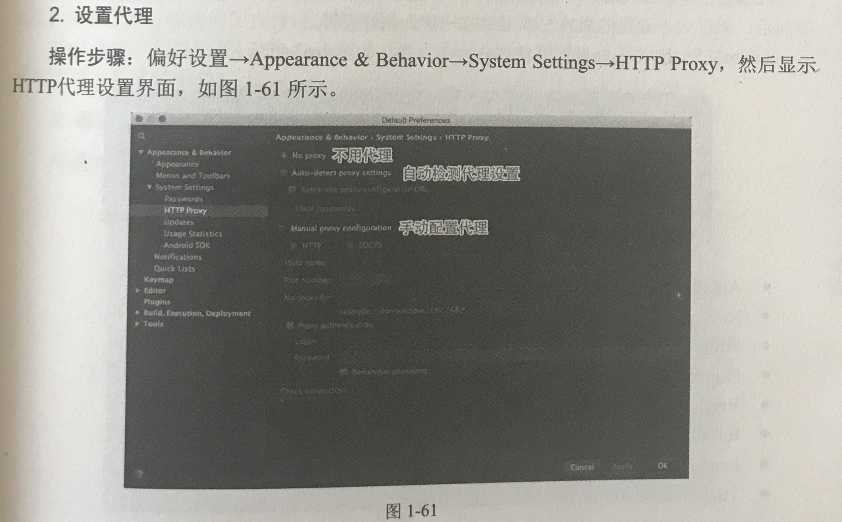
可能过这个网站下载各种版本：

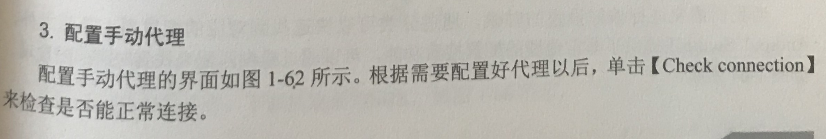
https://developer.android.com/studio/archive.html

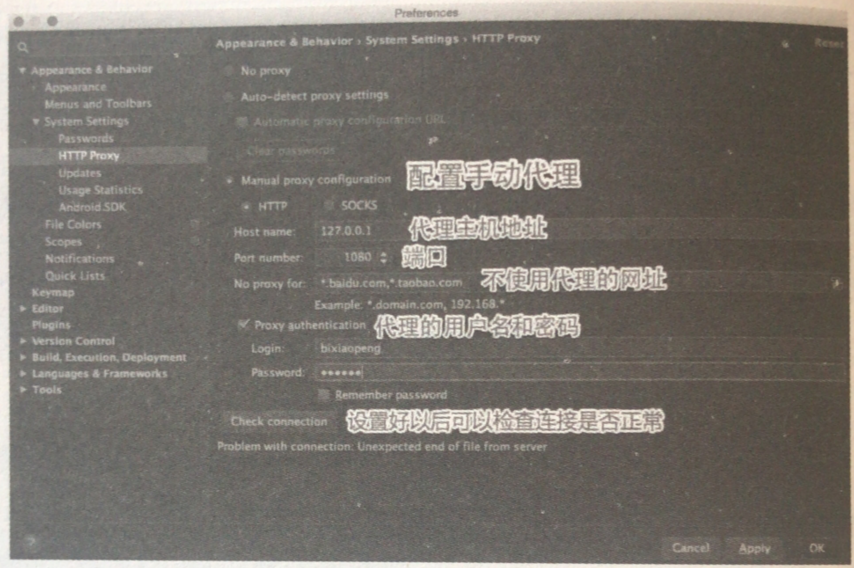


## 在AndroidStudio中使用代理

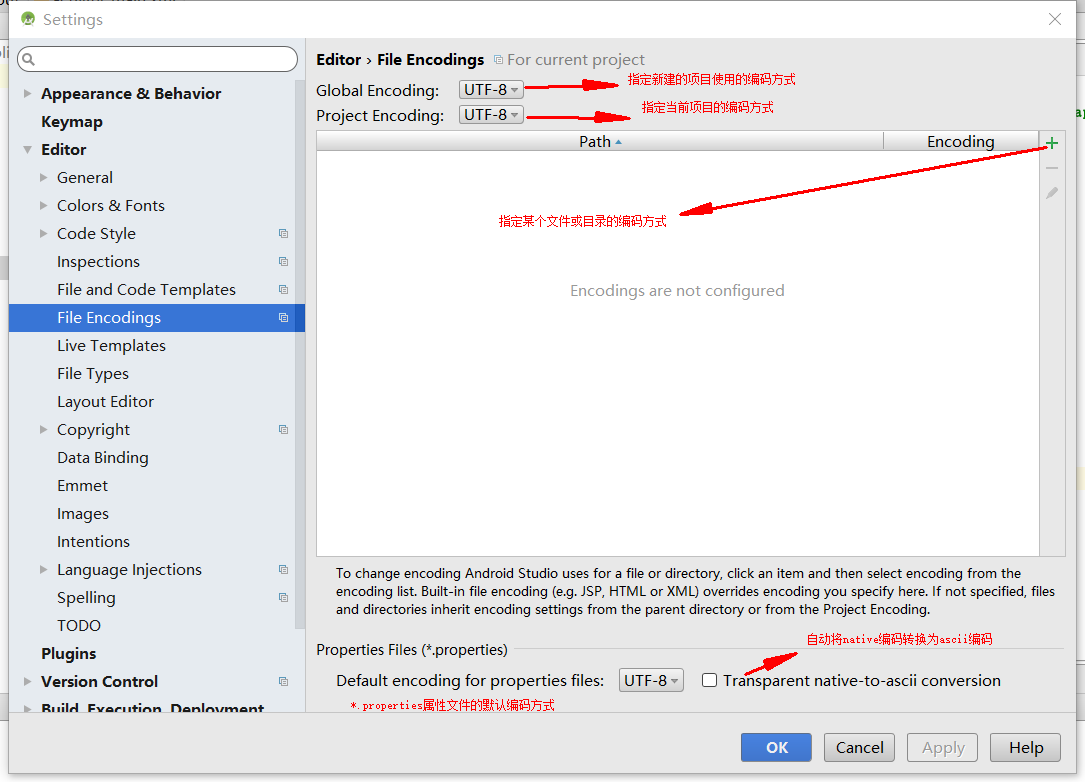




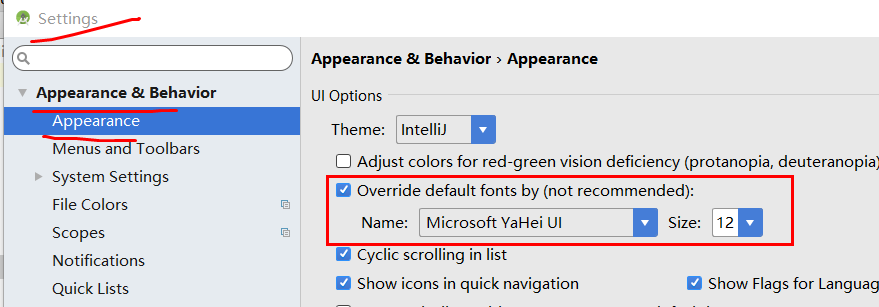




## 编码设置



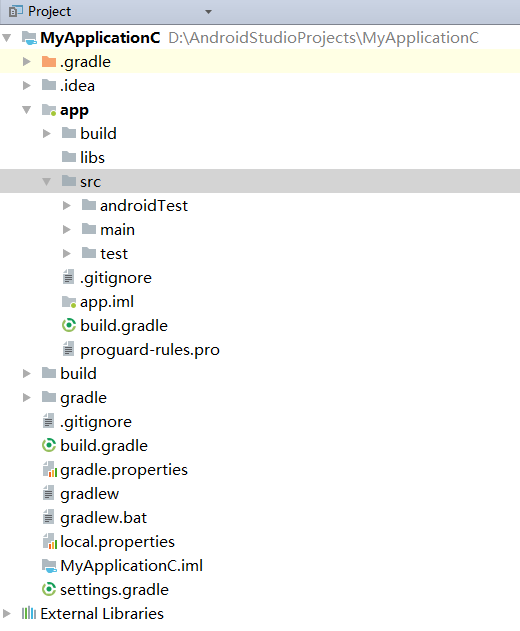
## 字体设置



# 项目与模块

## 2.1.2 基本的项目结构

Project模式会展示全部的文件信息，文件的位置是真实的物理结构，因为在查看文件的时候建议切换到Project模式。



MyApplicationC\.idea 自动生成的用于存放AndroidStudio配置文件的目录，包括版权、检查配置、jar包信息、项目名、编译、编码、gradle、模块等

MyApplicationC\app\libs 存放Module依赖的jar包

MyApplicationC\app\src\androidTest 存放Android单元测试代码

MyApplicationC\app\src\test 存放单元测试代码

MyApplicationC\app\src\main\res\mipmap 存放启动图标

MyApplicationC\app\src\main\jniLibs 存放so文件

MyApplicationC\app\src\main\assets 资产目录

MyApplicationC\app\app.iml 模块配置文件

MyApplicationC\app\build.gradle 模块构建配置文件

MyApplicationC\build.gradle 项目构建配置文件

MyApplicationC\app\proguard.pro 代码混淆配置文件

MyApplicationC\app\build 模块编译后的文件存放目录

MyApplicationC\build 项目编译目录

MyApplicationC\gradle gradle 目录

MyApplicationC\app\.gitignore 模块中Git忽略配置文件

MyApplicationC\.gitignore 项目中Git的忽略配置文件

MyApplicationC\gradle.properties gradle配置文件

MyApplicationC\gradlew gradlew配置文件（Lunix）

MyApplicationC\gradlew.bat gradlew配置文件（Windows）

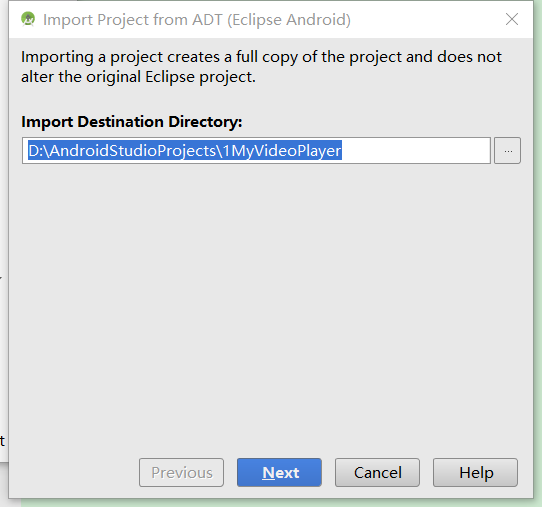
MyApplicationC\local.propeties 属性配置文件

MyApplicationC\settings.gradle 全局配置文件

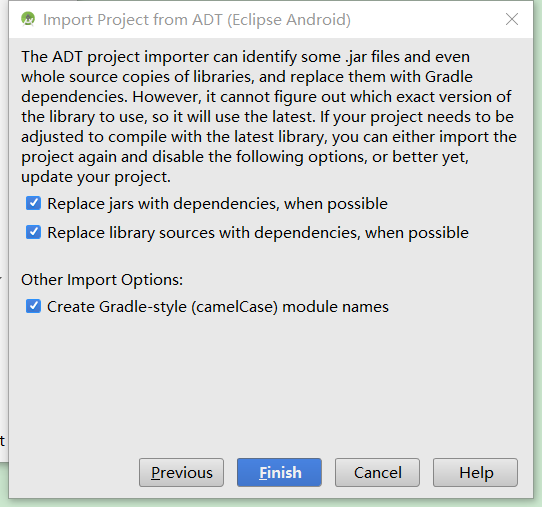
External Libraries 项目中使用到的依赖库存放目录

## 2.2 导入项目和模块

在导入Eclipse项目时，AS会复制一份，所以在选择了Eclipse项目的所在目录然后点下一步之后，会出现如下界面：



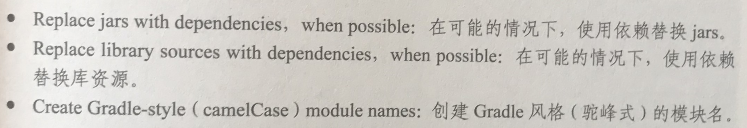
这用于指定把导入的项目存放在哪里，然后点击Next，出现如下界面：



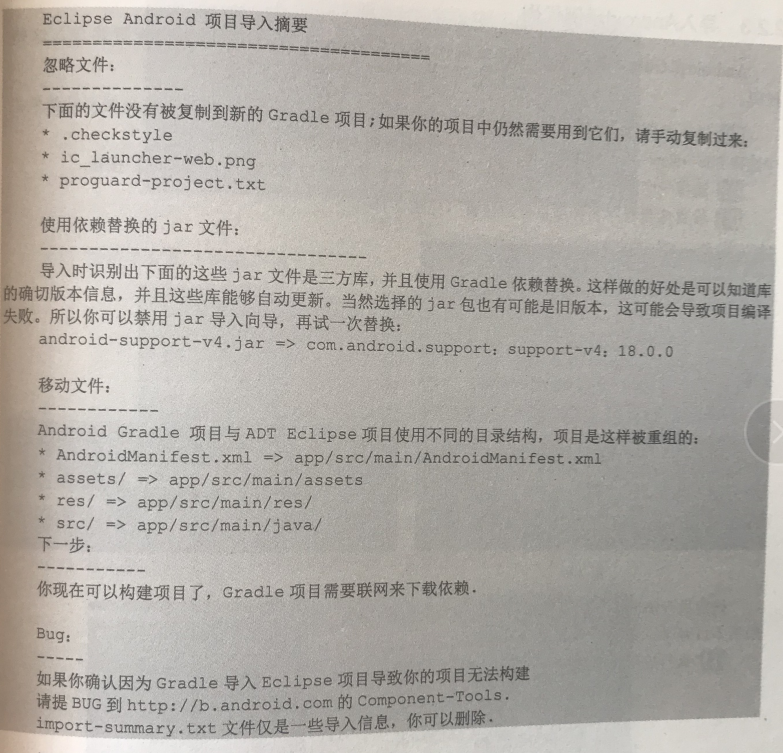
上面界面的英文翻译：

从ADT（Eclipse Android）导入项目

ADT项目导入器能识别一些jar文件和库，并且可以替换为Gradle依赖，但是无法确定要使用的库的版本号，因此替换为最新版本。



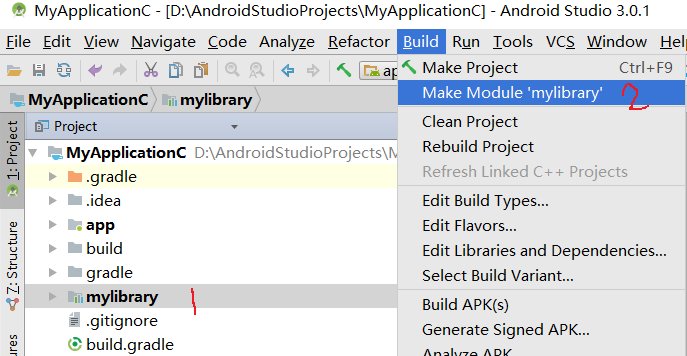
项目导入成功后会自动打开import-summary.txt文件，该文件记录了项目导入的摘要，通过这份摘要可以看出Eclipse项目是如何被转换成AS所需要的Gradle项目的。摘要简单翻译如下：



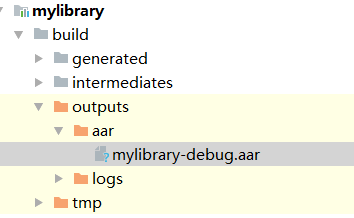
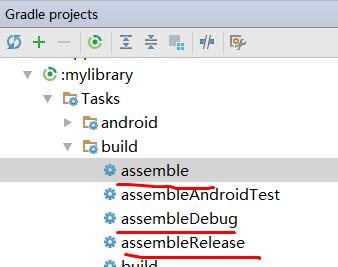
## 2.2.5 导入.jar/.AAR

.arr格式的包是Android独有的第三方库（Android Library），先说.aar是如何创建的：

先在AS中New一个库类型的Module，然后左侧视图中选择它，然后执行菜单命令，如下图步骤1、2：



然后在如下位置会生成.arr，生成的是一个debug的aar，也可双击Gradle任务中的assemble任务，这会生成release和debug两种，双击assembelDebug和assembleRelease只生成一个。

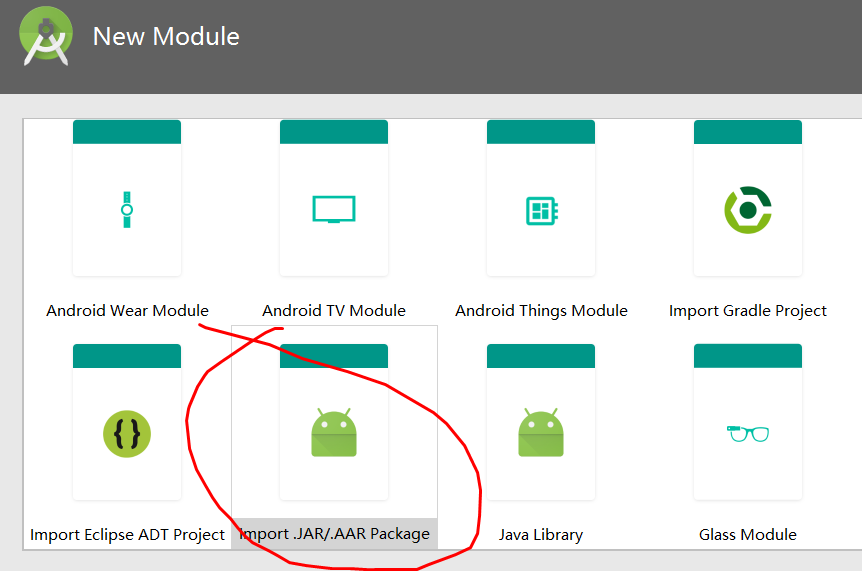
 

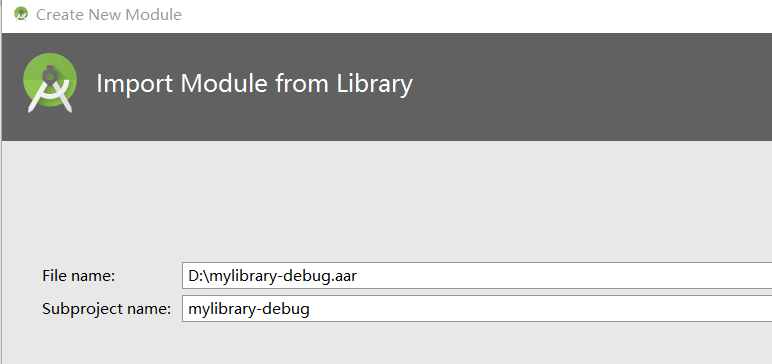
可以这样理解：应用类型的Module打包后是apk文件，库类型的Module打包后是aar文件。

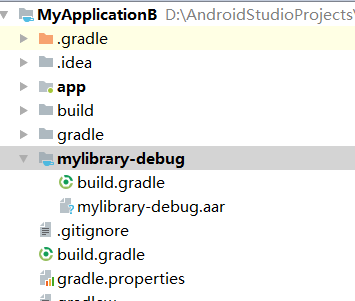
可以把这个.aar与别人共享，使用winRAR打开这个aar文件如下：



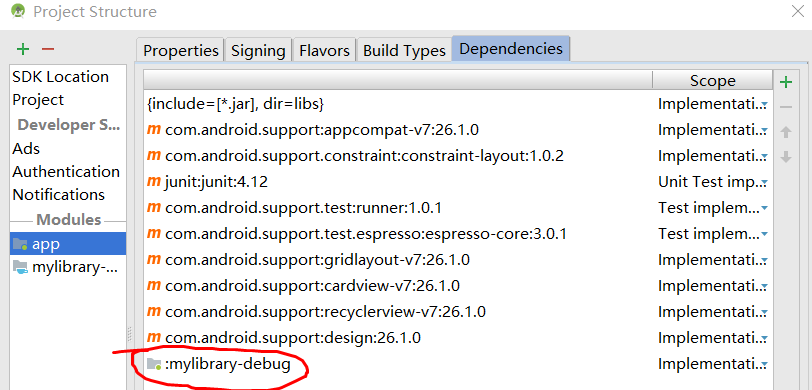
别人如何导入这个aar呢，执行New-Module，选择如下：



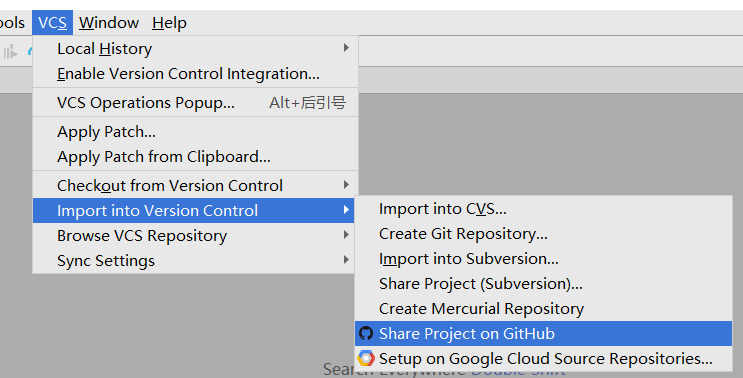


导入后效果如下：  
 

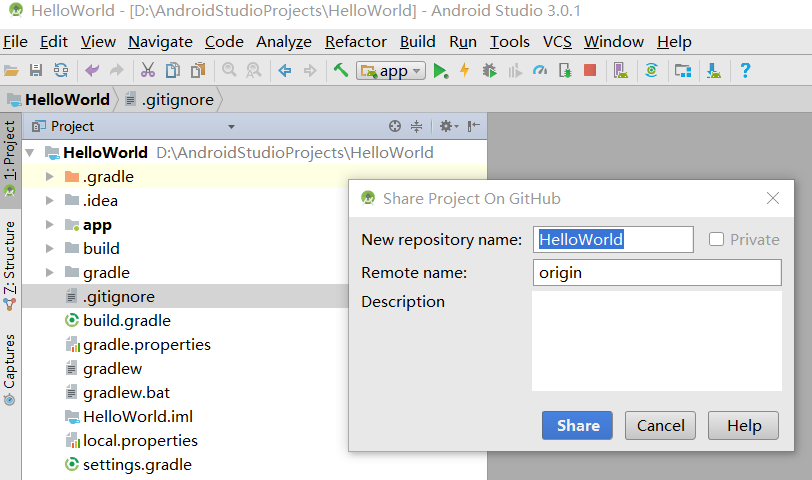
导入后还不能使用，需要给app这个module添加依赖，这跟之前的那些依赖一样添加即可，如下：



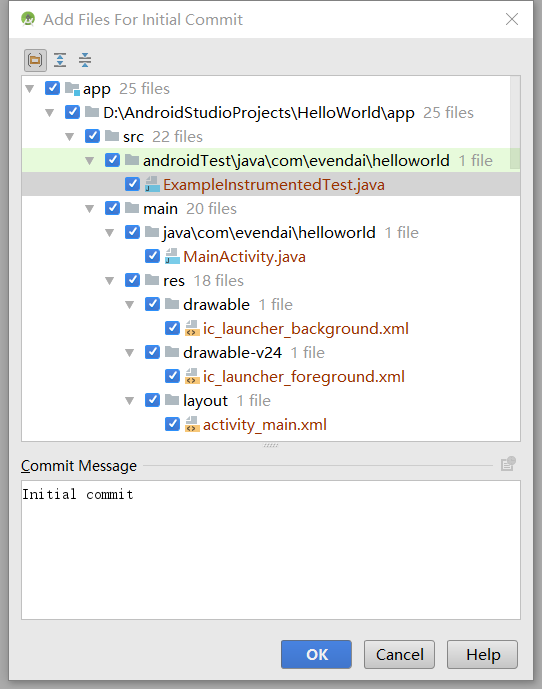
## 把项目分享到Github



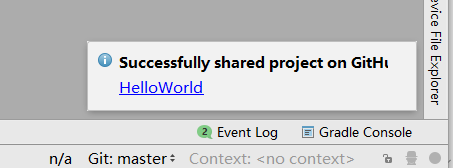
如果是第一次访问Github会要求输入Github的访问token，输入后点击Next，出现如下界面：



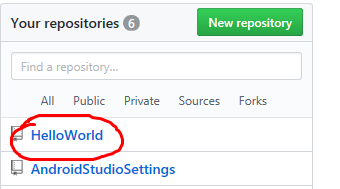
点击 “Share”，然后会出现如下界面：



点击 “OK”，一会在软件右下角会显示分享成功的提示，如下：

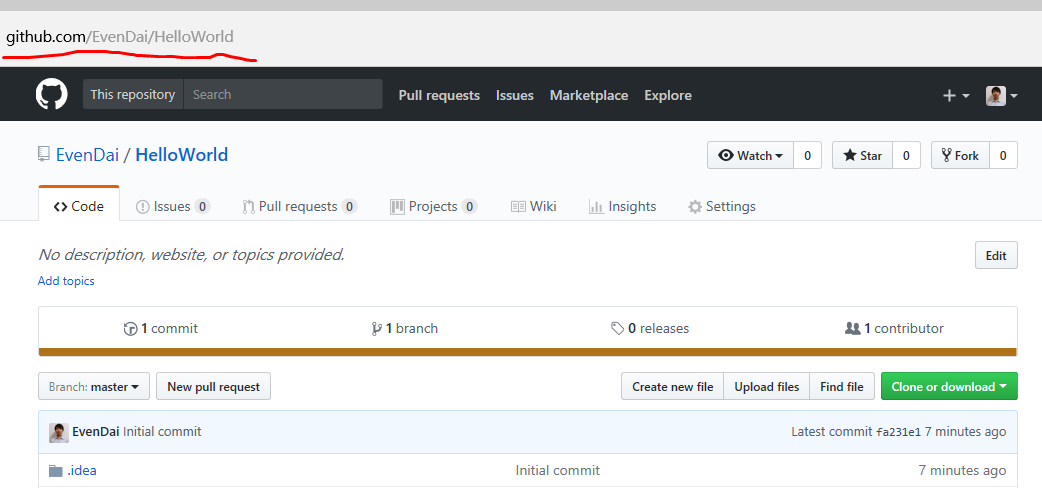


这时打开Github，则可看到刚刚分享的HelloWorld项目，如下：

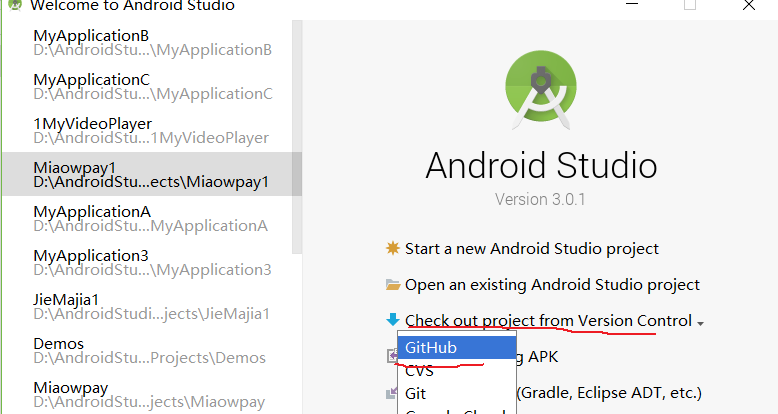


## 导入Github项目

首先在网页上打开Github项目，如下：

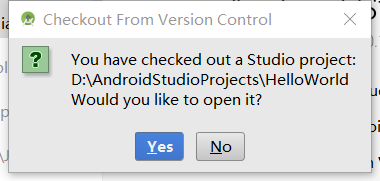


把浏览器中的url复制，如上图。



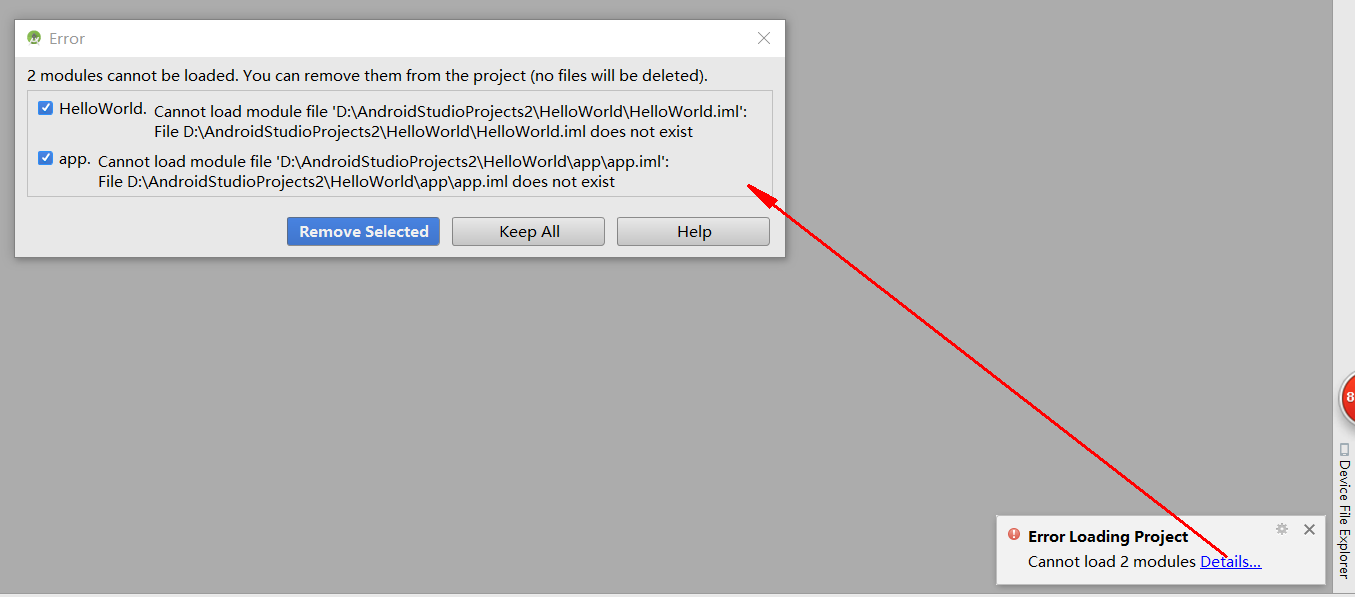


在点击“Clone”之前可以点击“Test”测试一下项目是否存在，克隆完毕后出现如下对话框：



大概意思是：你有捡出一个Studio的项目：HelloWorld，你想要打开它吗？

点击Yes后，一会在软件右下角出现提示如下：



英语解释：两个模块不能加载。您可以从项目中删除它们(不会删除任何文件)。HellowWorld。不能加载模块文件D:.....\HelloWorld.iml：文件HelloWorld.iml不存在。

原因就是我们在提交的时候并没有提示\*.iml文件，点击 “Remove Selected”，然后点击AS中的同步按钮，然后AS会自动生成所需的iml文件。

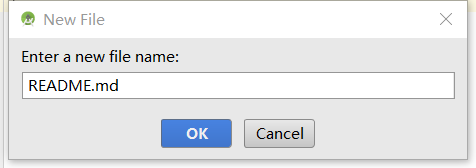
如果上面的提示框消息了，可以从下面的位置打开：



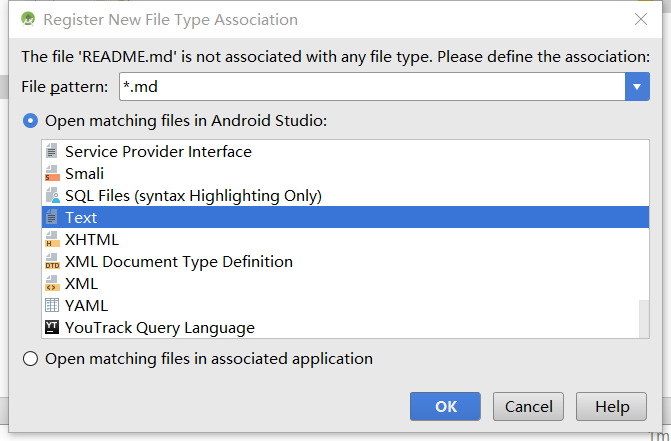
## 新建其他格式的文件

新建的文件格式如果AS不能识别，如下：

New-->File



点击OK后如下：



翻译：标题：注册新文件类型关联

‘README.md' 文件没有关联任何的文件类型，请定义关联

在AS中打开匹配的文件（如果选择Text，则代表AS会把\*.md的文件格式当成txt那样去打开）

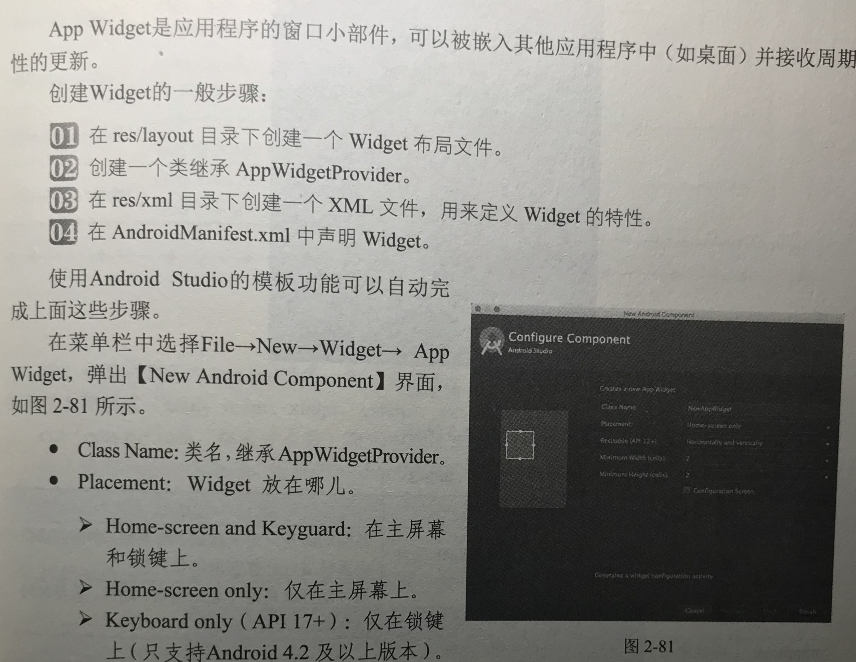
在关联的应用中打开文件（创建好后，双击文件则会使用关联的应用程序打开，如果安装了MarkDown会自动使用MarkDown打开）

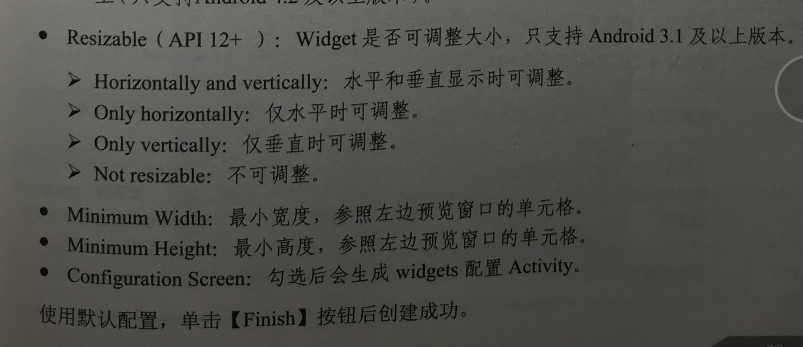
## Service的一些属性

Exported 表示是否支持其他应用调用当前组件

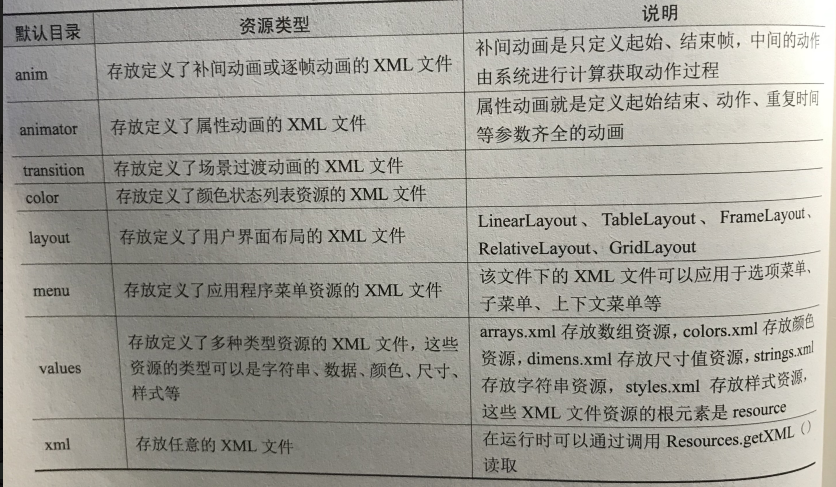
Enable 表示该服务是否能够被实例化

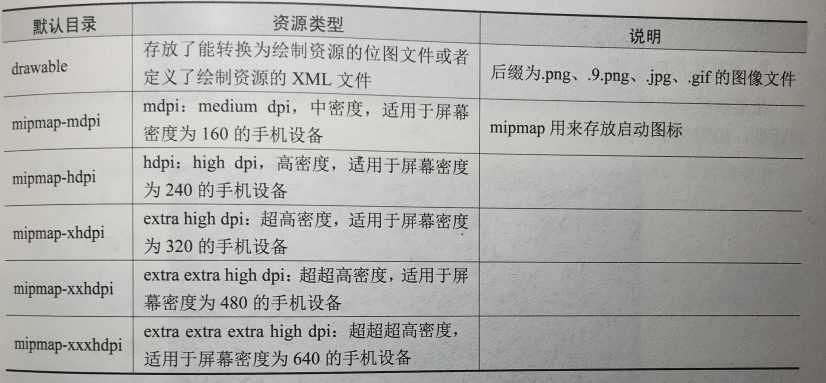
## 创建App Widget





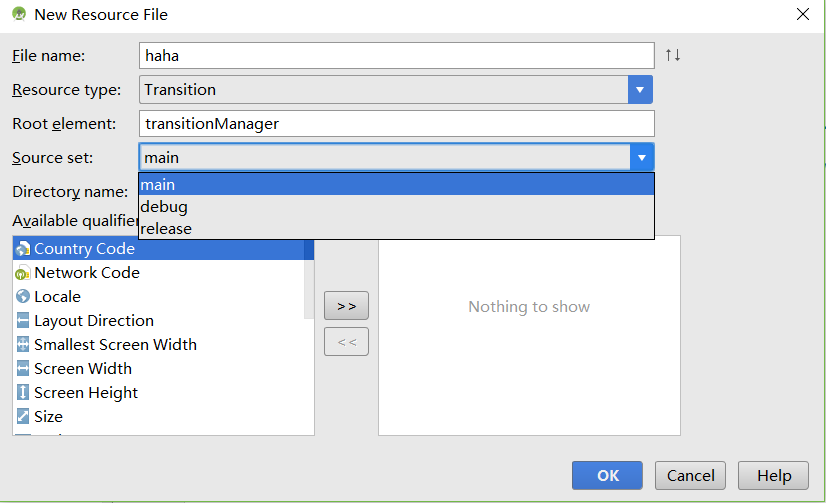
## 资源文件





## 创建资源文件

在res目录右击，选择New -->Android resouce file，如下：

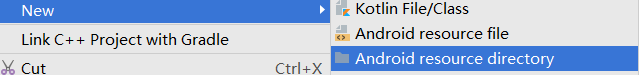


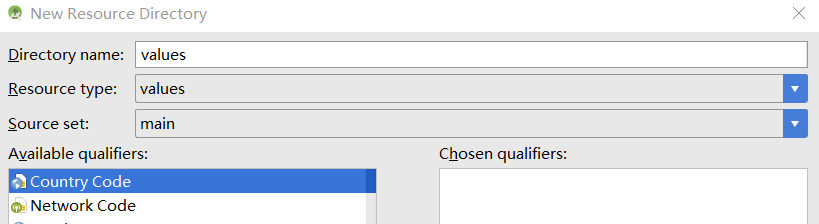
Source set中有3个选项，如果还有其他变体这里也会现变体，在打没的包时会引用不同的资源。

也可以在指定的res目录上右击，如在anim目录上右击，则出现的New菜单为对应的动画资源的命令：



新建资源目录：





这仅仅是创建目录。当与别人合作开发时，怕冲突的话可以创建自己的资源目录，如：values-evendai

### 资源限定符