

---

# 实验一

## Android 开发环境配置

2017 年 9 月 24 日

---

### 实验介绍

在学习 Android 程序开发之前，需要首先构建完整的 Android 开发环境。本次试验将分为两大部分：首先是 Android 的开发环境配置，然后我们将介绍如何开发你的第一个 Android 程序。

其中 Android 开发环境的配置包括安装 Java Development Kit (JDK) 和 Android Studio 集成开发环境 (IDE) 等数个步骤，其中 IDE 包括了 Android 软件开发工具包 (SDK)。

*Note*：在 2014 年 12 月前，Android 的官方开发 IDE 为 Eclipse，现在 Google 已经停止对其提供支持，Android Studio 提供向后兼容 (Backward Compatible)，但是在文件结构等方面还是有些区别。

### 1.1 系统要求

- Windows 2003 (32-bit or 64-bit)
- Windows Vista (32-bit or 64-bit)
- Windows 7 (32-bit or 64-bit)
- Windows 8 / Windows 8.1
- Mac OS X 10.8.5 or later (Intel based systems only)
- Linux systems with version 2.11 or later of GNU C Library (glibc)
- Minimum of 2GB of RAM (4GB is preferred)
- 1.5GB of available disk space

### 1.2 安装 Java Development Kit

Android 开发主要采用 Java 语言，所以 JDK 便是 Android 开发最底层的软件组建。开发需要的 JDK 为 6 以上的版本。注意 Java 同时提供 JDK 和 Java Runtime Package (JRE)，不要安装错误。

#### 1.2.1 Windows 上安装 JDK

首先假设你的电脑上从没有安装过 JDK，则你只需要从以下网址中下载最新版的 JDK Windows 版并执行安装过程即可。

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

#### 1.2.2 Mac OS X 上安装 JDK

Java 在 OS X 上并没有被默认安装，你可以在终端中运行一下命令检查：

```
java -version
```

如果 Java 已经被安装则会显示类似以下信息：

```
java version "1.8.0_25"  
Java(TM) SE Runtime Environment (build 1.8.0_25-b17)  
Java HotSpot(TM) 64-Bit Server VM (build 25.25-b02, mixed mode)
```

否则运行java 指令则会出现指令框：

No Java runtime present, requesting install

此时我们点击弹出框的“更多信息”按钮则会指向 Oracle 的 Java 网页，在该网页上下载 Java 的 SDK（为.dmg 镜像文件），装载并双击（类似下图）可安装。



安装完成后重新在终端中执行命令查看安装情况。

### 1.3 下载 Android Studio

首先我们下载 Android Studio 安装包，选择对应的系统。为了方便大家下载，我们提供以下几个链接供选择：

<http://developer.android.com/sdk/index.html>（官网）

<http://www.android-studio.org/>

Note：为了方便安装使用，请下载完整的 IDE 而非仅仅下载 SDK。

#### 1.3.1 安装 Android Studio

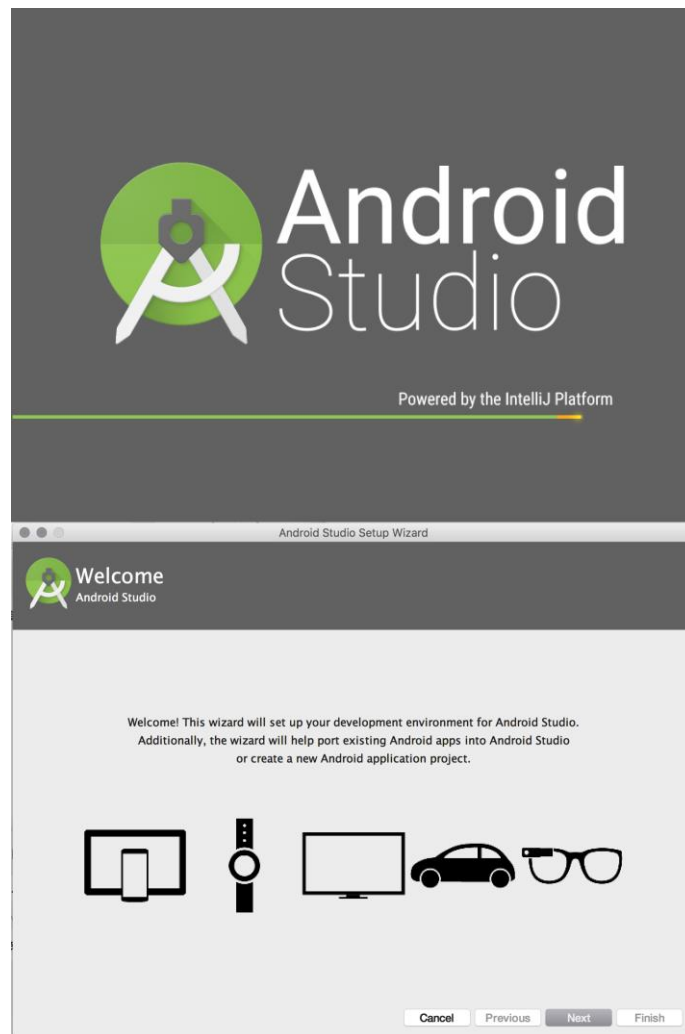
直接执行安装并按照提示即可（Mac 则直接将.dmg 中的程序拖入程序文件夹即可）。

Note：部分 Mac OS X 可能会遇到 JVM 不存在的提示，可能需要安装 Java 6 JRE，参考以下链接。

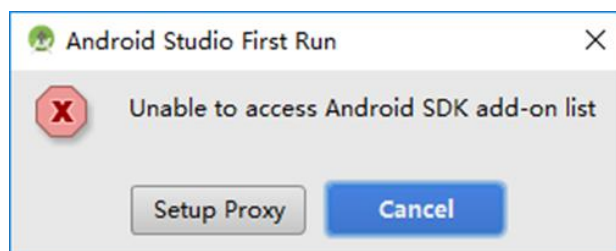
<http://support.apple.com/kb/DL1572>

#### 1.3.2 Android Studio Setup Wizard

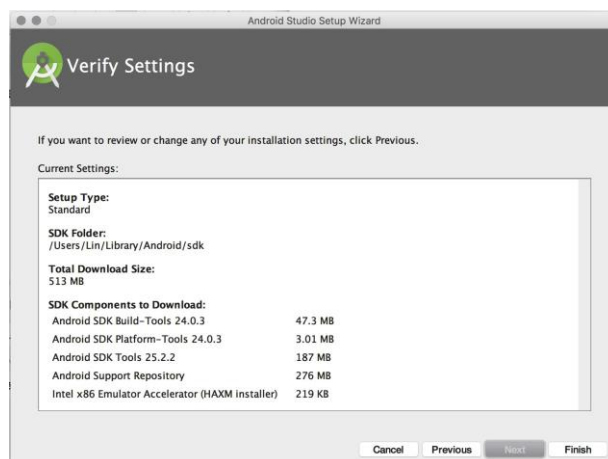
在第一次运行 Android Studio 的时候，首先会进入加载界面，然后会提示你是否导入曾经安装的 Android Studio 文件，如果没有，则会运行 Setup Wizard。



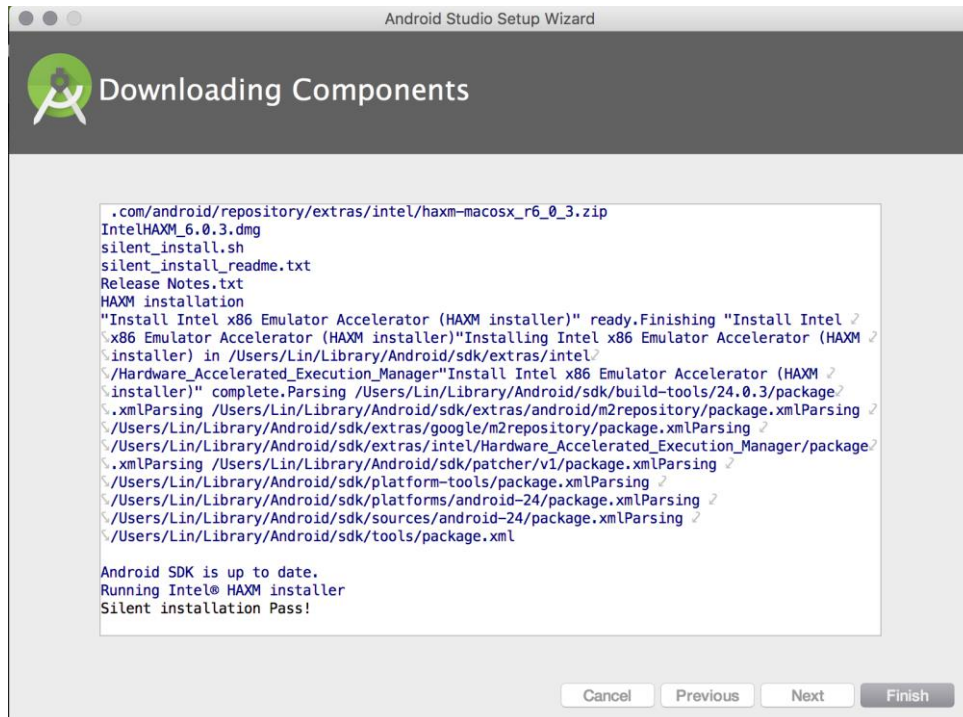
如果你碰到下面这个对话框，选择“Cancel”。



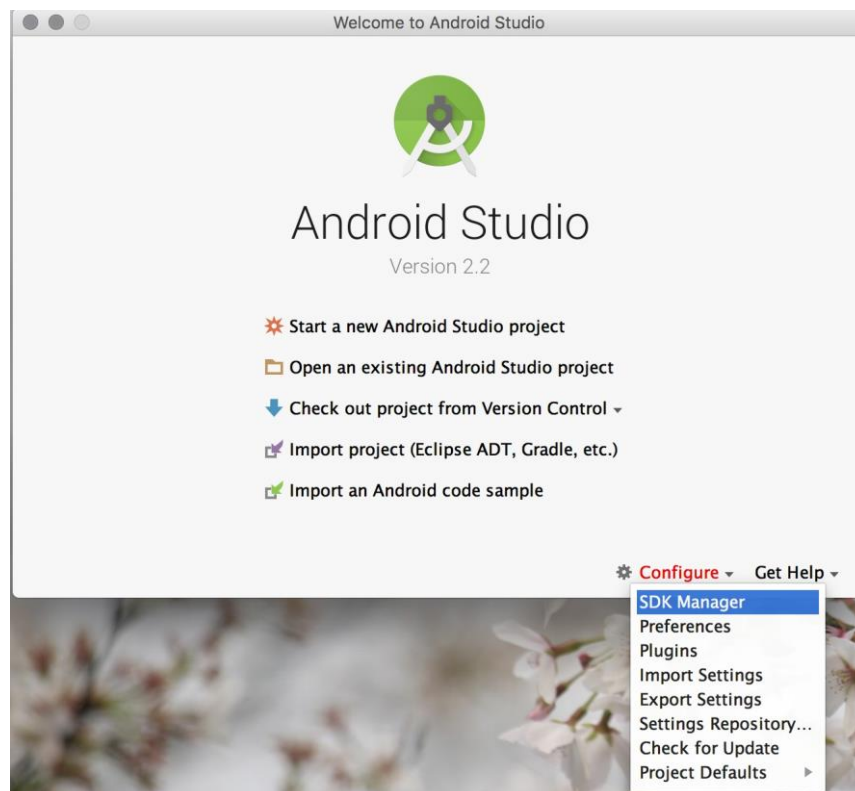
此时你可能会遇到部分需要下载的内容，通常情况下为 SDK 的更新等。



这个时间可能不短，需要耐心等待。

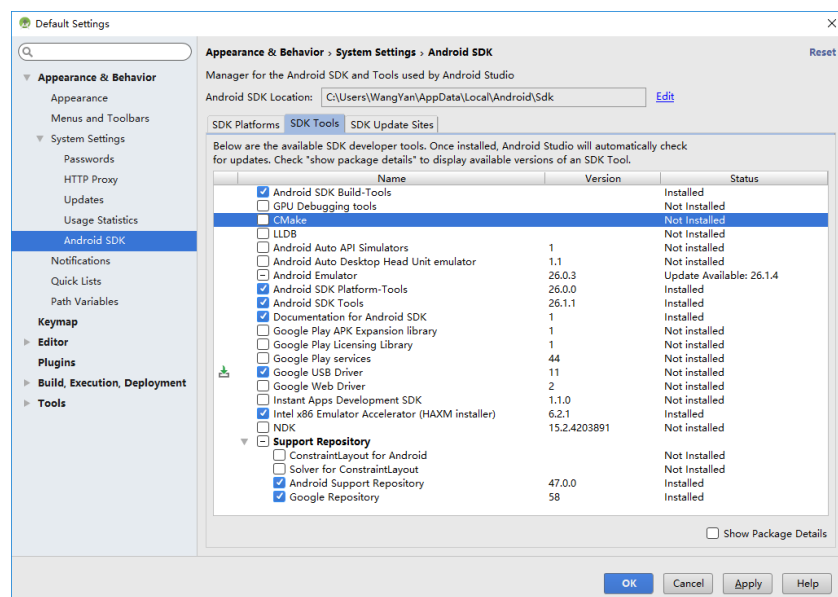
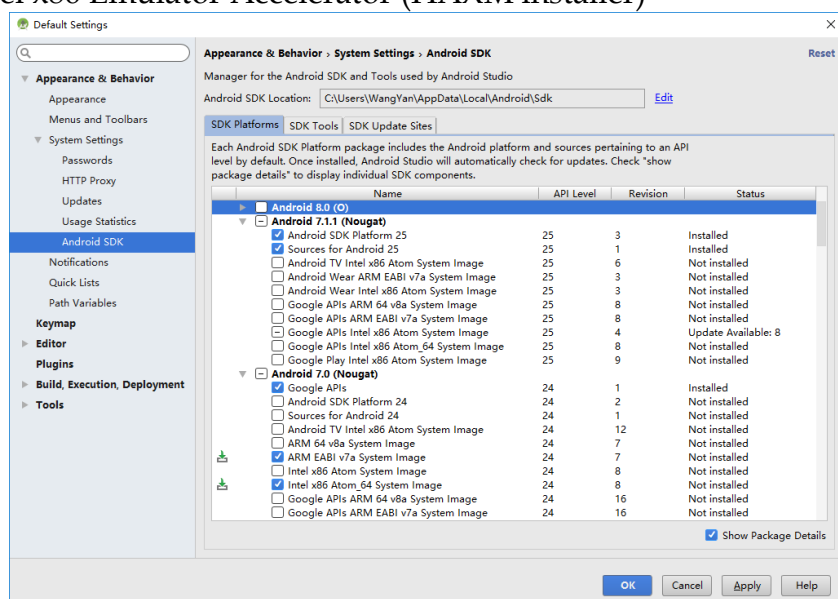


此时，我们将会进入以下界面，选择打开 SDK Manager。

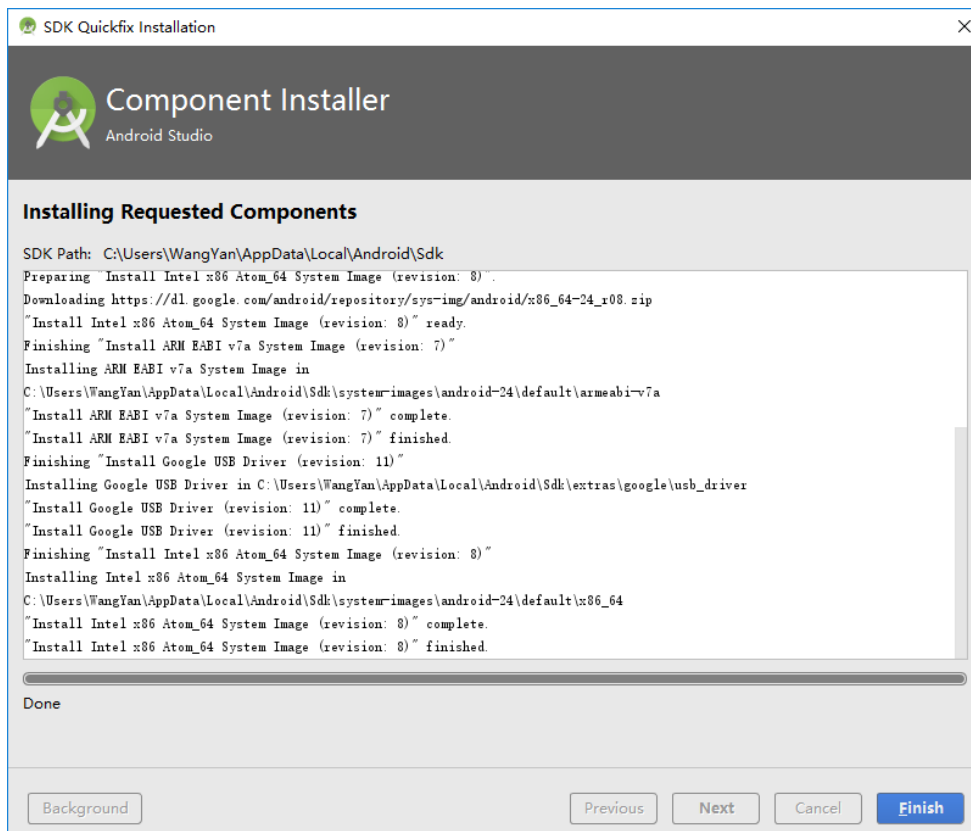
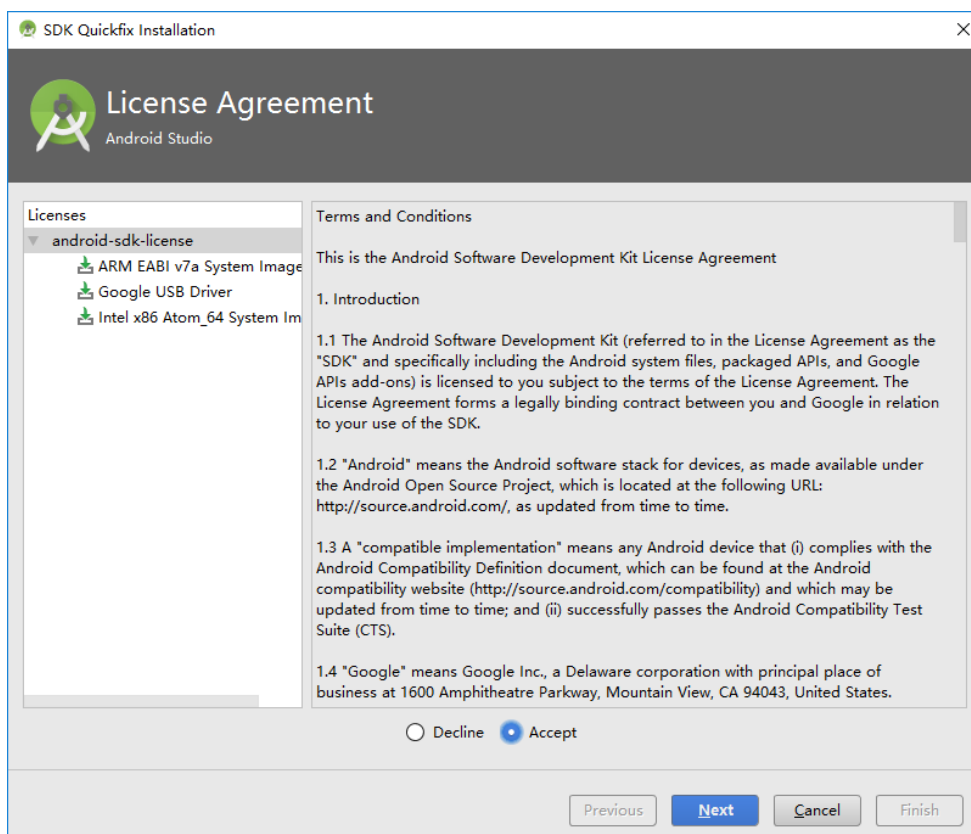


然后如图所示，勾选“Show Package Details”，再勾选需要的 Android 开发工具和版本并安装。（本实验主要基于 Android 5.0.1）。注意，要确保以下部分安装：

- Tools > Android SDK Tools
- Tools > Android SDK Platform-tools
- Tools > Android SDK Build-tools
- SDK Platform (most recent version) > SDK Platform
- SDK Platform (most recent version) > ARM EABI v7a System Image
- SDK Platform (most recent version) > Intel x86 Atom\_64 System Image(更快的虚拟化)
- Extras > Android Support Repository
- Extras > Google Repository
- Extras > Google USB Driver (Required on Windows systems only)
- Extras > Intel x86 Emulator Accelerator (HAXM installer)



最后，点击“Apply”再同意协议即可进行安装。安装完成后对照以上列表，确认所有部分已经被安装，否则勾选未安装部分重新进行安装。

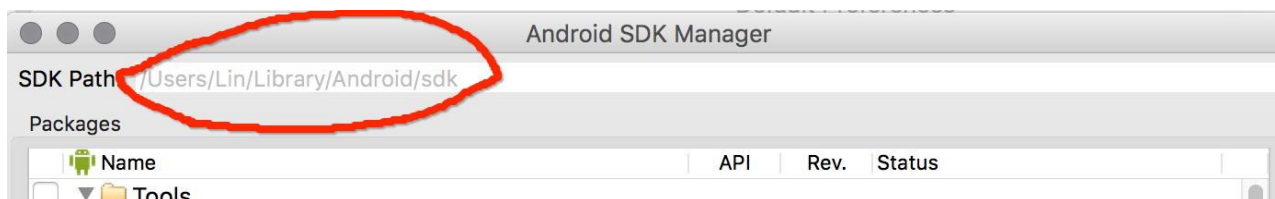


### 1.3.3 命令行工具的使用

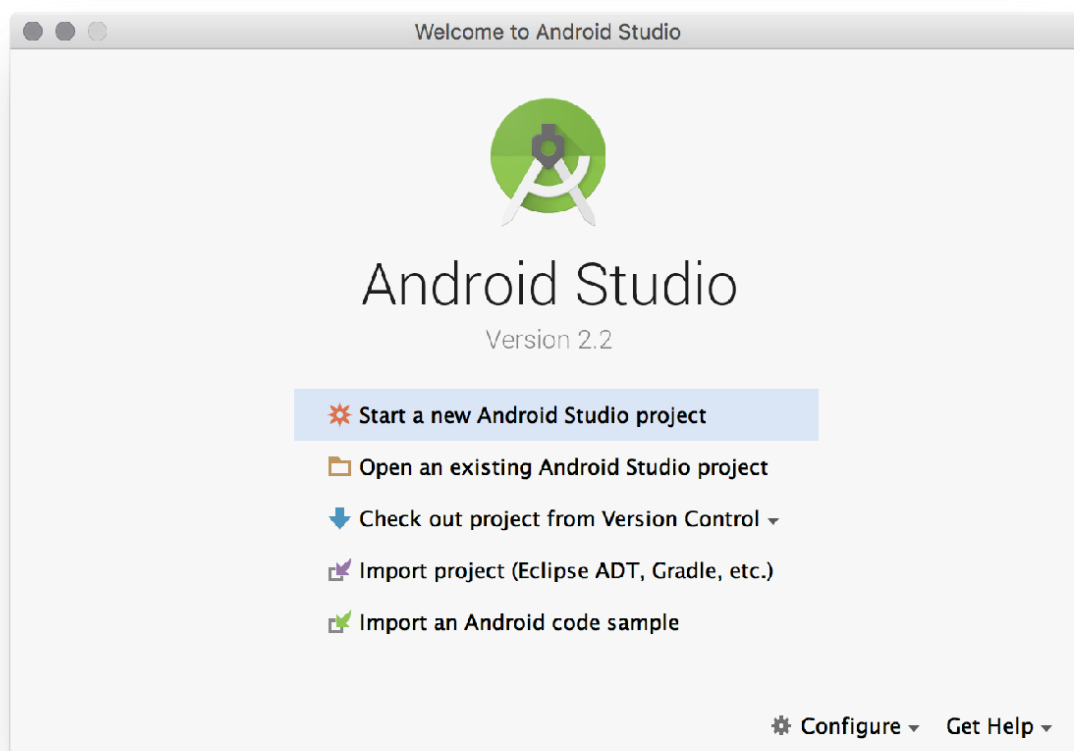
为了让命令行工具可用，我们需要将 SDK 的工具路径加入到 PATH（环境变量）中。

```
<path_to_android_sdk_installation>/sdk/tools  
<path_to_android_sdk_installation>/sdk/platform-tools
```

以上两部分即为需要添加的变量，其中<path\_to\_android\_sdk\_installation>可以在 Android SDK Manager 中看到：



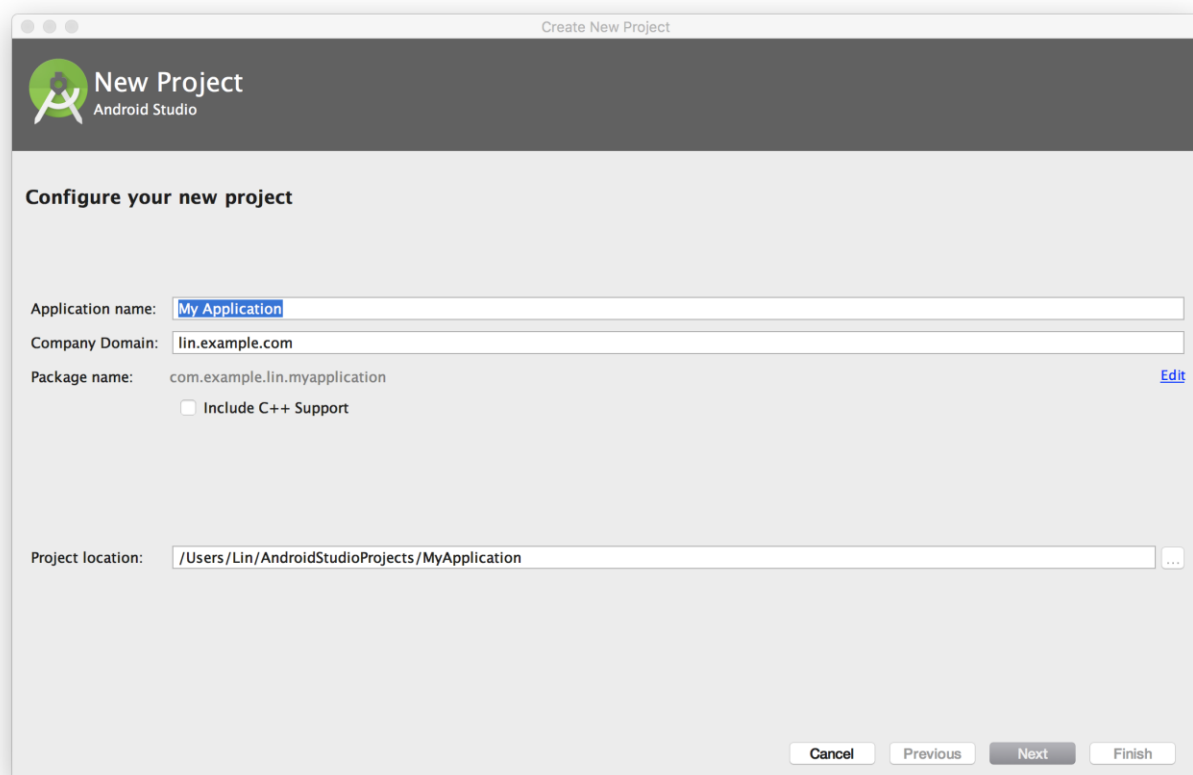
## 2.1 创建一个新的 Android Project



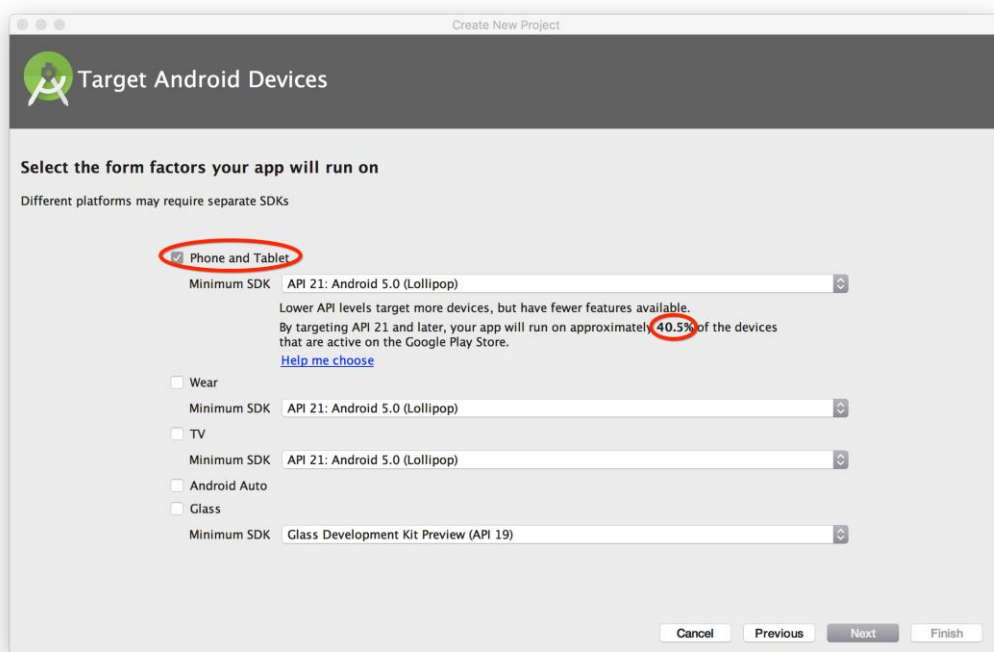


在 Welcome 界面中选择 *Start a new Android Studio project*。进入下面的界面。其中 Application name 为 Android Studio 识别的名称同时也是程序完成后再 Google Play 商店中 销售 的名称。

*Package Name* 为了在 Android 生态系统中唯一的标识一个程序，它应当为你的域名的 逆序加上程序名。



点击下一步，我们会看到项目开发的目标设备，这里我们勾选 Phone and Tablet.

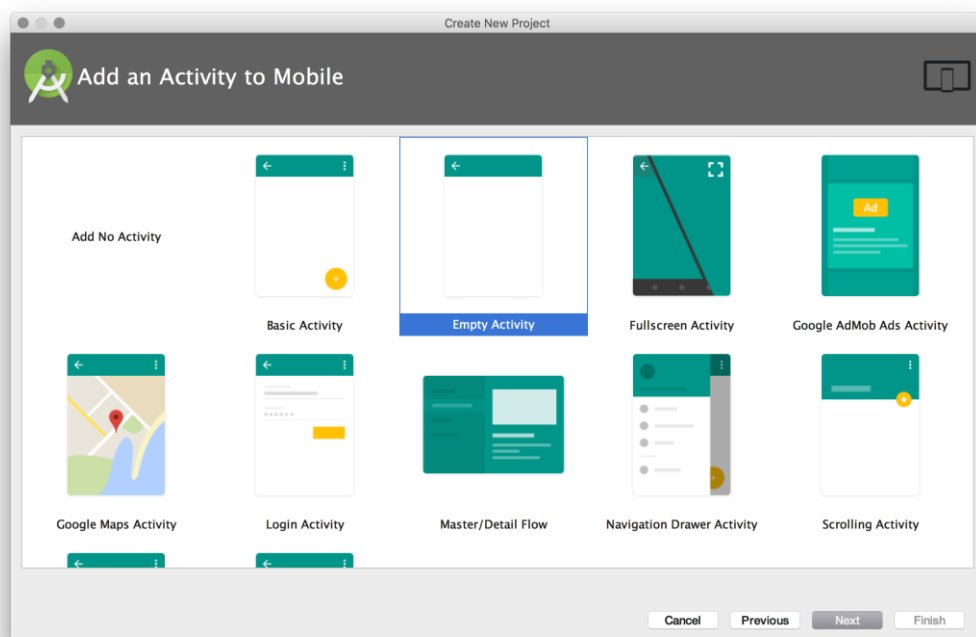




时，你也可以在界面中看到大致可以运行在多少设备中。

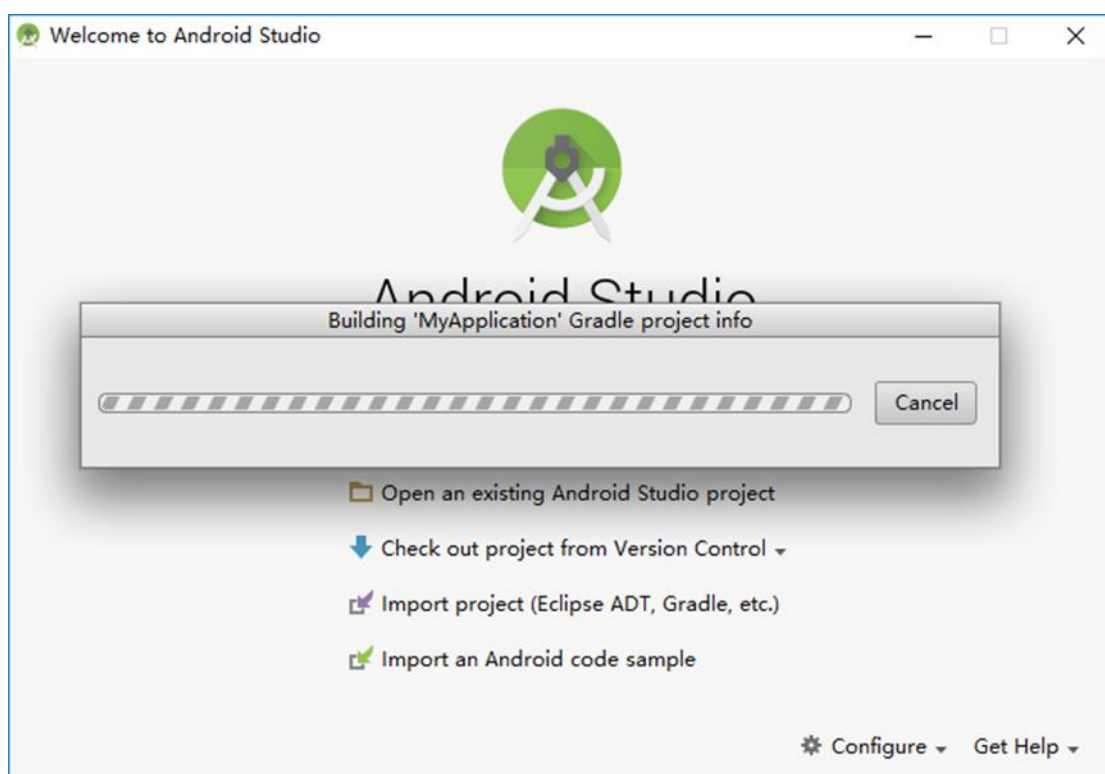
## 2.2 创建一个 Activity

点击在下一步中，我们可以选择程序一个初始的 Activity。



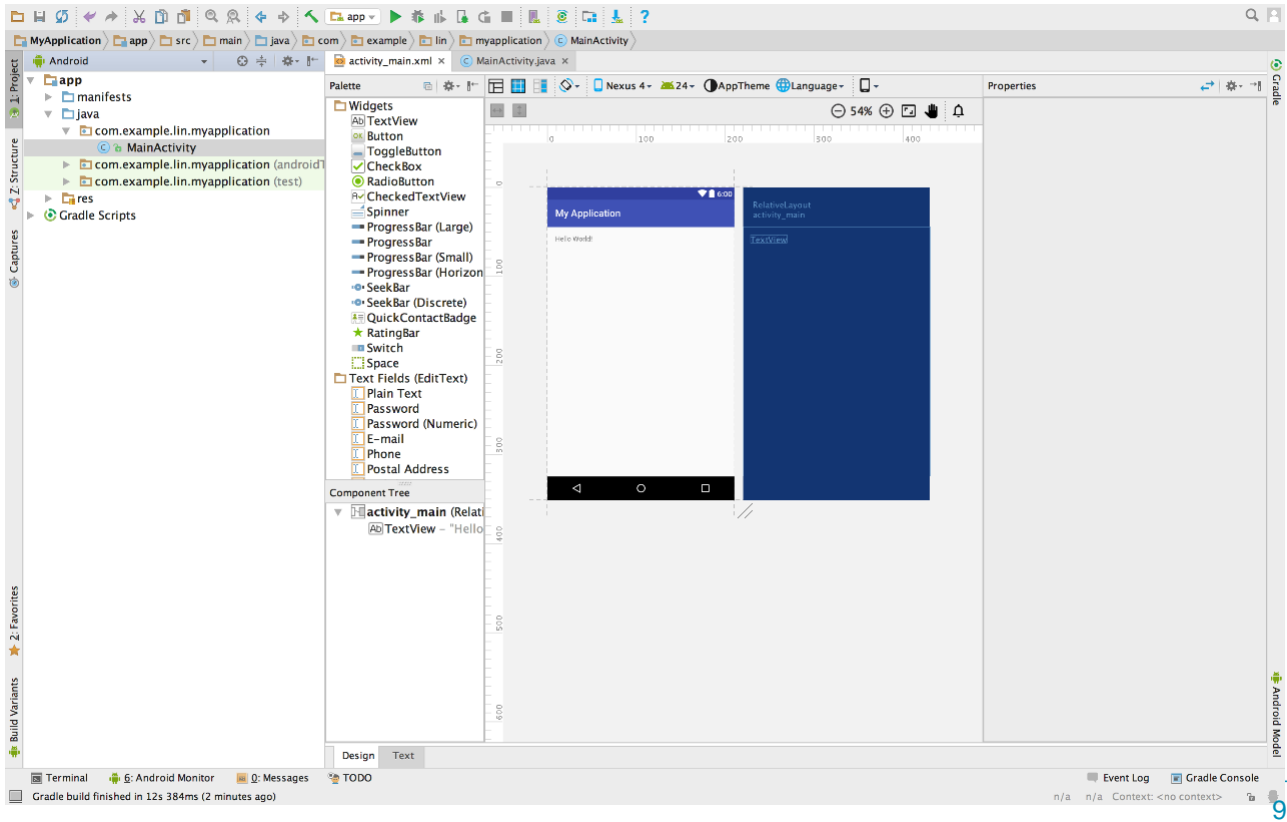
在这里，我们选择 Empty Activity。点击下一步，设置 Activity 和 Layout 的命名，并点击 Finish。

如果你卡在以下界面较长时间，可以到群里下载文件 gradle-3.3-all.zip，然后拷贝到 C:\Users\XXX\gradle\wrapper\dists\gradle-3.3-all\55gk2rcmfc6p2dg9u9ohc3hw9 目录下，最后，关掉 Android Studio (可能需要强制结束进程)，再重新创建一个项目或者打开刚才创建失败的项目。

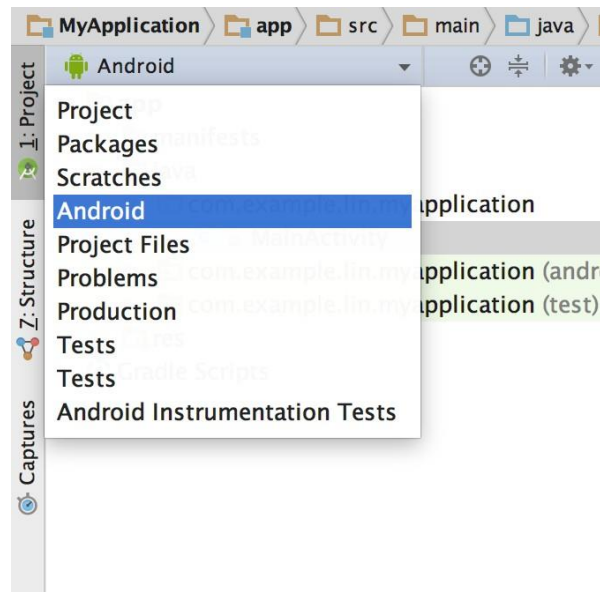


## 2.3 修改示例程序

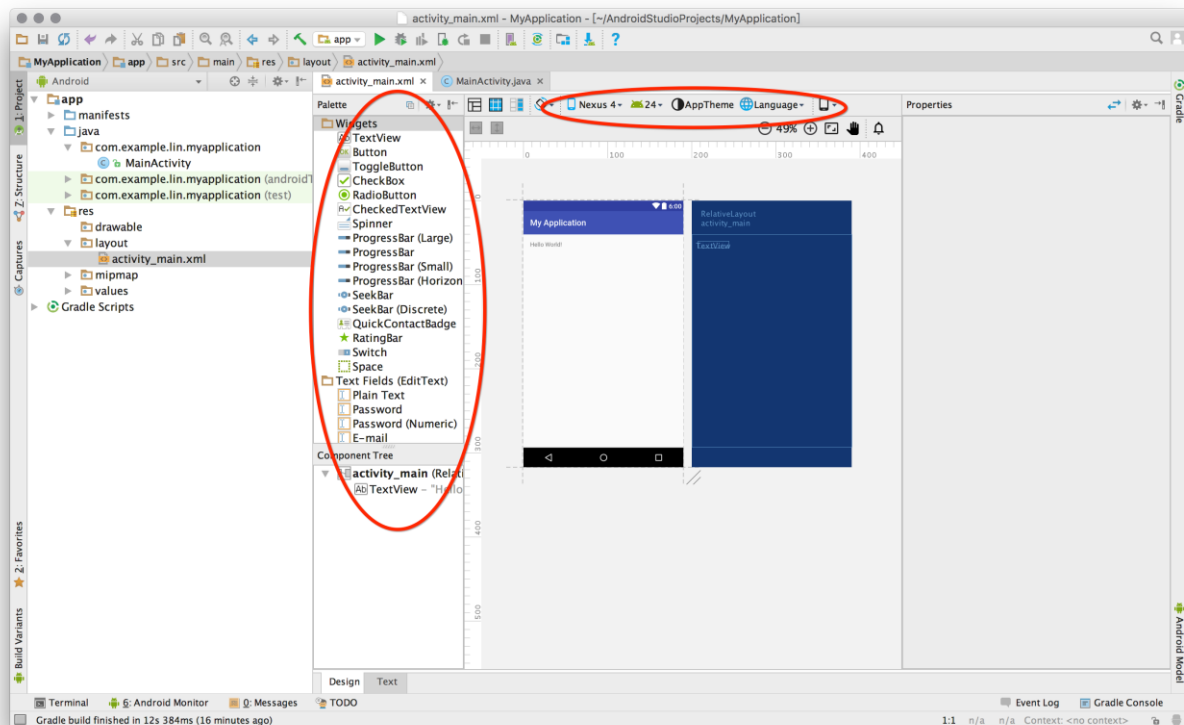
此时，我们已经创建了一个初始的程序，并显示主界面。




与程序相关的文件将在左边栏中显示，同时，ProjectTool 窗又默认将为 Android 模式，如果不是，请将其改为 Android 模式：

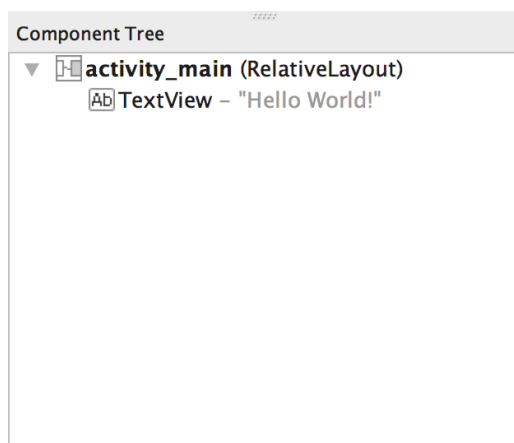


此时，我们创建的程序中已经包括了一个 label“Hello World”。我们的 Activity 的 UI 设计在文件 activity\_main.xml 中（如果你没有改过名字的话），位于 app->res->layout 中。双击该文件，界面中将会在中心显示 UI 设计工具。其中包括了许多可添加的元素包括开关，搜索栏，勾选框等等。

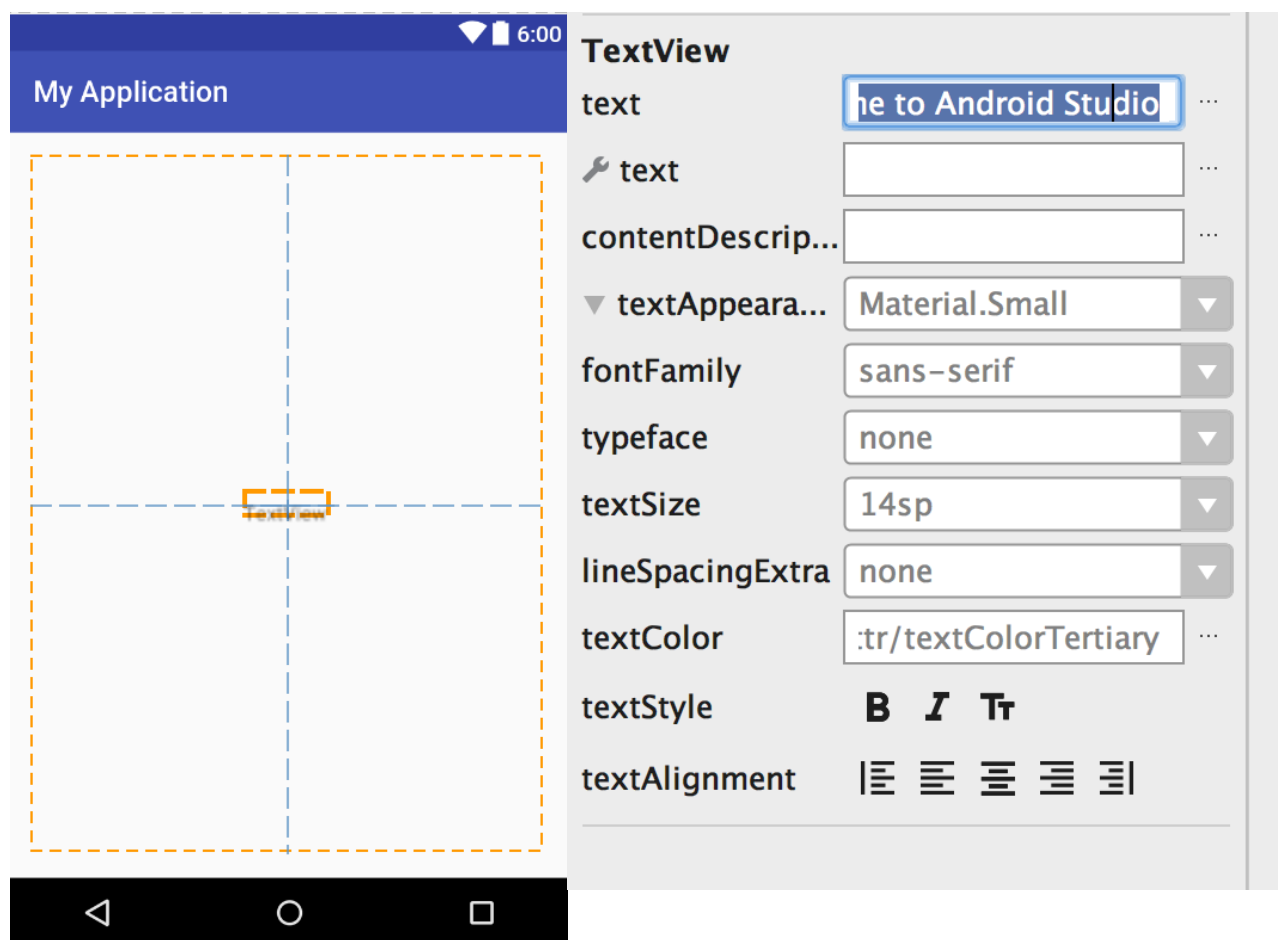


同时在上方的工具栏中，我们看到目前设计已经选为 Nexus 4，下拉菜单中还有很多设备供选择。同时你也可以通过  选项来调节横屏和竖屏。

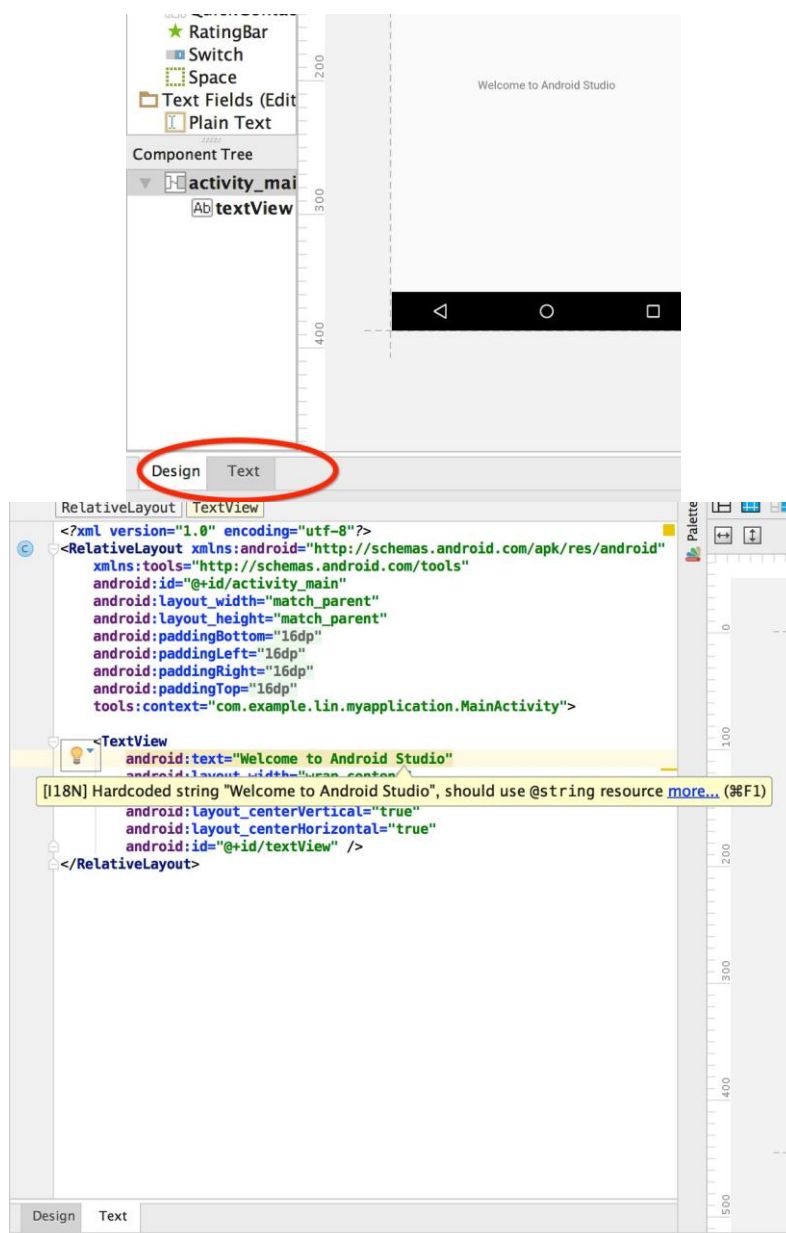
在界面的中间你可以预览目前的布局，但是并不是所有的部件对于开发者来说都是可见的，这时你需要通过界面的中下部的 Component Tree 来查看。目前程序采用的为相对布局。



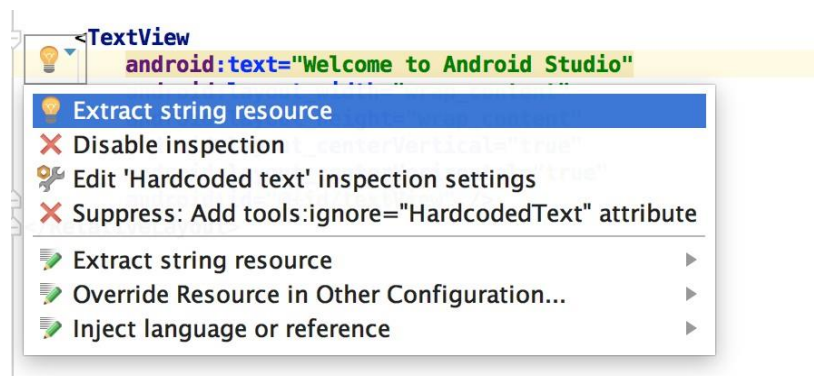
在改造过程中，我们首先在 Component Tree 中删除 TextView。然后在中间的 Palette 中单击并拖拽 TextView 组件放置于屏幕的中间。然后将 text 修改为“Welcome to Android Studio”



2.4 检查 Layout 和资源文件 在上一步中，我们通过工具提供的菜单删除添加了交互组件，但实际上这只是 IDE 提供的对 activity\_main.xml 的一个用户友好的方式，本质上还是对 activity\_main.xml 的内容进行了修改。点击下图的按钮，将图形化的设计界面改为代码。



注意到，在上图中你会看到一个 I18N 提示，I18N 代表 internationalization，以 I 开头以 N 结尾，中间有 18 个字母。这个警告提示我们无论什么时候，都应该尽量将属性和值例如 text strings 保存为 resource 的形式。这里，我们点击左边的灯泡提示，选择 Extract string resource 提示，输入 welcomestring 并点击 OK。



此时，这个 string 将会保存在 app -> res -> values -> strings.xml 中。

Android Studio 一个非常强大的功能在于修改 xml 文件时，右侧将会实时预览显示 UI 的设计情况。

在左侧的 xml 文件中加入红圈中的字，观察出现的情况。



此时你应该注意到，右侧的预览中主体部分已经变成了红色，同时代码左侧也出现了一个红色的方块。这在编写代码时非常有用。



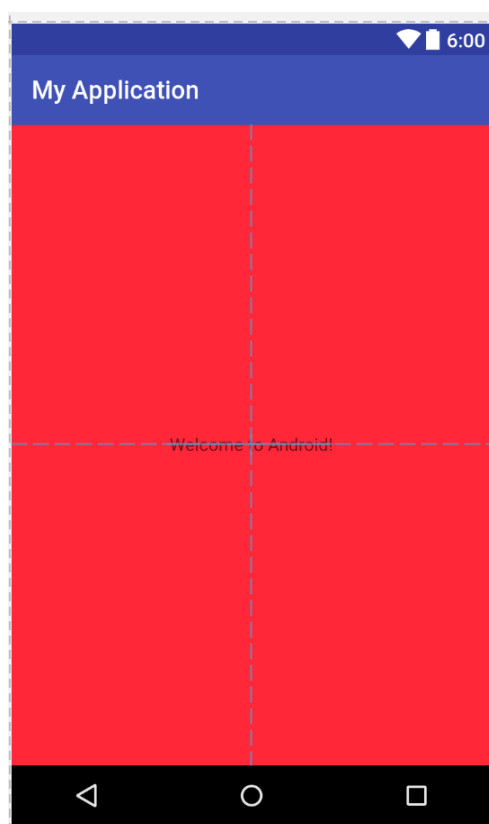
```
tools:context="com.example.lin.myapplication.MainActivity"
android:background="#ff2438">
```

最后，我们打开 app -> res -> values -> strings.xml 文件。

```
<resources>
  <string name="app_name">My Application</string>
  <string name="welcomestring">Welcome to Android Studio</string>
</resources>
```

修改 welcomestring 的 value，然后回到设计界面，你会发现对应的值已经改变。

```
<resources>
  <string name="app_name">My Application</string>
  <string name="welcomestring">Welcome to Android!</string>
</resources>
```



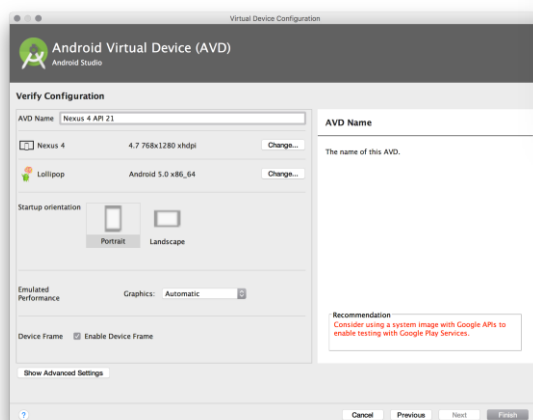
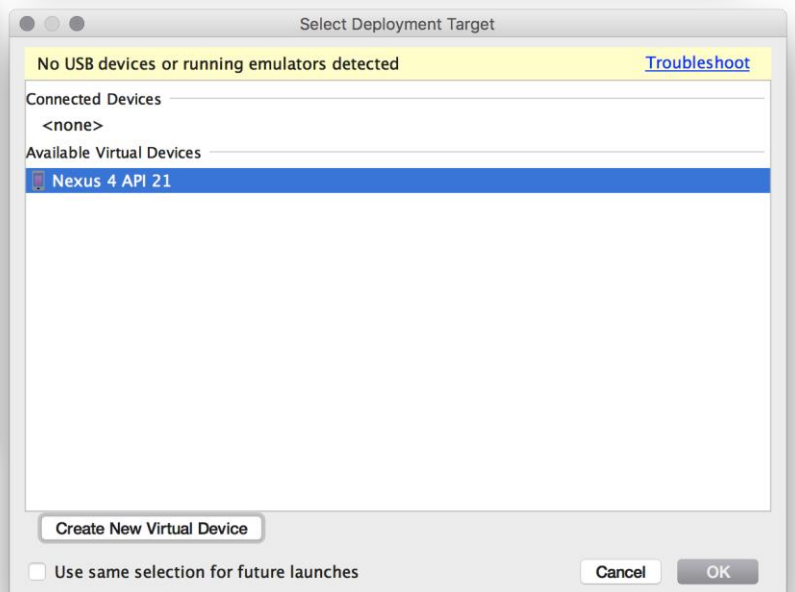
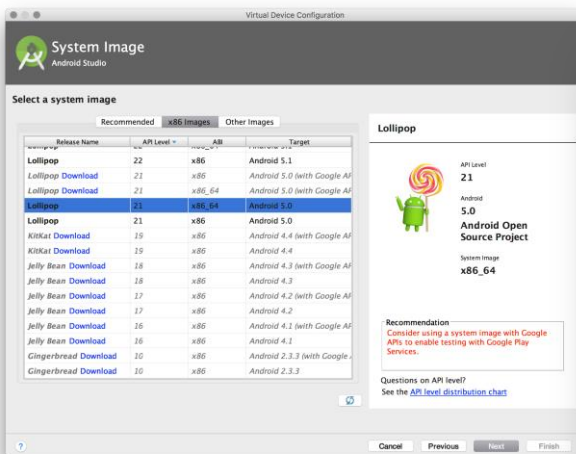
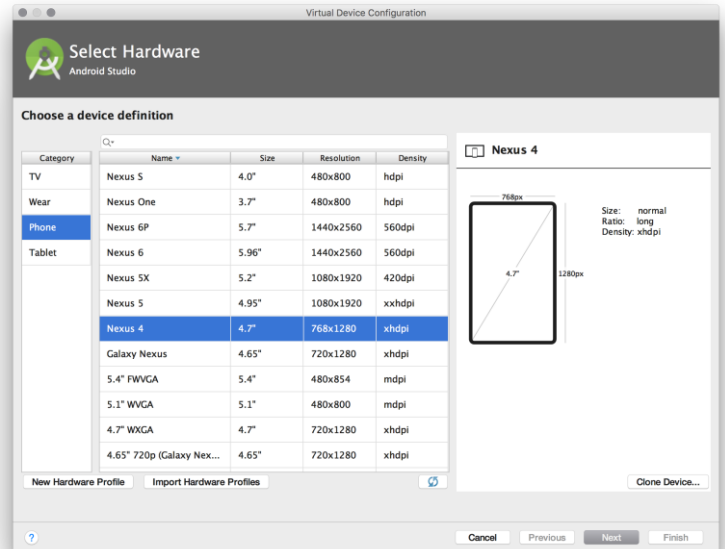
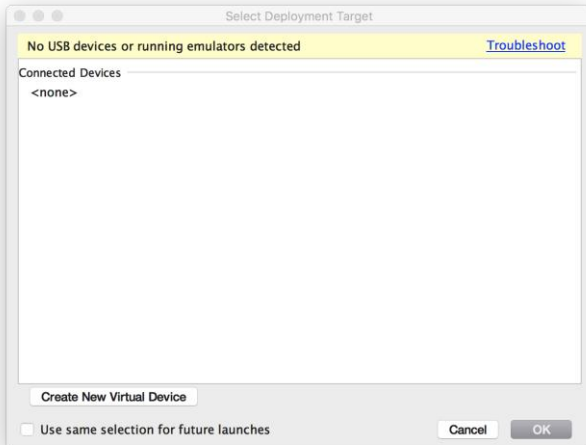
此外，定位资源的另一个快捷的方式为，在 Text 模式下，点击 '@string/welcomestring'，当字符亮起时，按快捷键 Ctrl+B（Mac：command+B），AndroidStudio 将会自动定位至目标。



最后，我们在 Android 中模拟运行该程序。点击运行栏中的  图标，此时因为第一次运行，需要创建一个新的虚拟设备。点击 Create New Virtual Device，选择目标设备，这



里我们选择 Phone 和 Nexus 4，最后选择镜像，推荐的镜像目前主要为 5.1 及以上的，我们点击 x86 Images 并选择 Lollipop ( with Google APIs 的镜像可以测试 Google Play 服务 )。



最后选择刚刚创建的虚拟设备，点击 OK 即可运行。运行结果如下图。

