

# 第 1 章 Android 概述

## 1.1 Android 简介

### 1. Android 的起源、发展与现状

Android 是当今最为流行的三大（另外两个为苹果 iOS、微软 Windows Phone）智能手机操作系统之一。Android 一词的本意指“机器人”，最初是由 General Magic 公司软件工程师 Andy Rubin（安迪·鲁宾）开发的一款专用于移动终端设备的 OS（操作系统）软件。2003 年 10 月，Andy Rubin 等人创建 Android 公司，并组建 Android 团队。2005 年 8 月 17 日，Google（谷歌）低调收购了成立仅 22 个月的 Android 公司及其团队。现在被称为 Android 之父的 Andy Rubin 在公司被收购之后成为 Google 公司工程部副总裁，继续负责 Android 项目。而原 Android 这个公司名也就作为 OS 产品的名称被保留了下来。

Google 在收购 Android 公司后，继续对 Android 系统进行开发运营。

- 2007 年 11 月 5 日，Google 正式向外界展示了 Android 手机操作系统，并且在这天宣布建立一个全球性的联盟组织——开放手持设备联盟（Open Handset Alliance）来共同研发和改良 Android 系统。该组织由 84 家手机制造商、软件开发商、电信运营商以及芯片制造商共同组成，以支持 Google 发布的手机操作系统及其上的无数应用软件，Google 则以 Apache 免费开源许可证的授权方式，公开了 Android 的源代码。
- 2008 年 10 月，第一部 Android 智能手机发布。随后，Android 逐渐扩展到平板电脑及其他领域，如电视、数码相机、游戏机等。
- 2011 年第一季度，Android 在全球的市场份额首次超过塞班系统，跃居全球第一。
- 2013 年的第四季度，Android 平台手机的全球市场份额已经达到 78.1%。同年 9 月 24 日，Android 迎来其 5 岁生日，全世界采用这款系统的设备数量已经达到 10 亿台！
- 2014 年第一季度，Android 平台已占有所有移动广告流量来源的 42.8%，首度超越了 iOS。

如今 Android 已是全世界市场占有率和普及率最高的智能手机操作系统，最新版本为 Android 7.0。在中国大陆更是家喻户晓，其中文名称为“安卓系统”。

### 2. Android 的系统架构

Android 和其他操作系统一样，采用了分层的架构，如图 1.1 所示。

从架构图可以看出，Android 系统分为四个层次，从高到低分别是应用程序层（Applications）、应用框架层（Application Framework）、系统运行时库（Libraries/Android Runtime）和 Linux 内核层（Linux Kernel）。

#### （1）应用程序层

Android 系统一般会与一系列核心应用程序包一起发布，该应用程序包包括客户端、SMS 短消息程序、日历、计算器、地图、浏览器、联系人管理簿等智能手机最常用的应用（App）。所有的应用都是使用 Java 语言编写的。

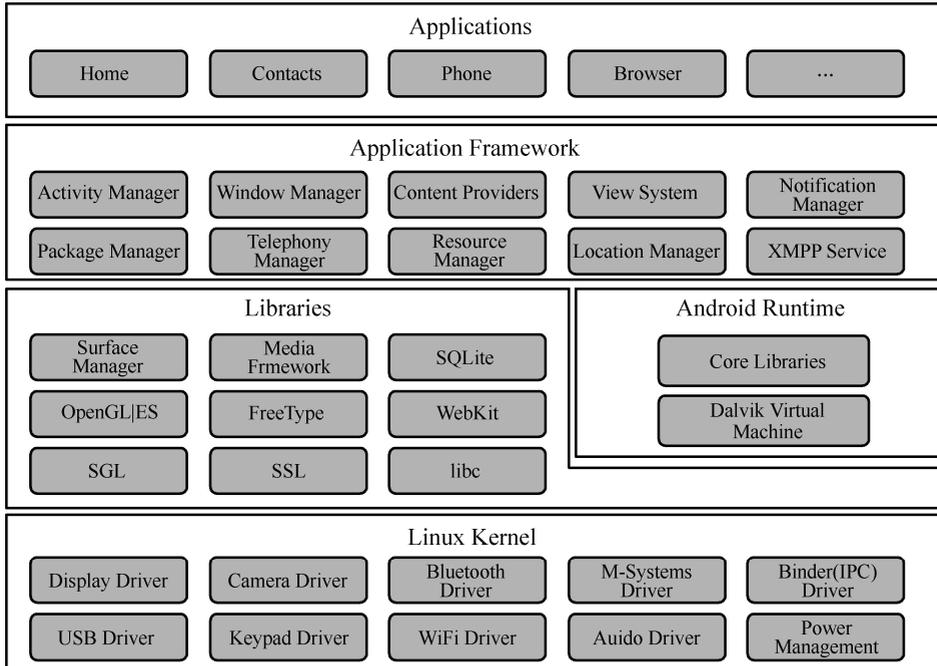


图 1.1 Android 系统的分层架构

## (2) 应用框架层

应用框架层是核心应用程序包所使用的 API（应用编程接口）框架，应用的开发人员可以完全访问。该 API 框架的架构设计简化了组件的重用，任何一个应用都可以发布它的功能块并且任何其他的应用也都可以使用其所发布的功能块（不过得遵循框架的安全性）。同样，这种重用机制也使用户能够非常方便地替换程序组件。

隐藏在每个应用后面的是一系列的服务和系统，其中包括以下几方面。

- **活动管理器 (Activity Manager):** 用来管理应用生命周期并提供常用的导航、回退等功能。
- **内容提供者 (Content Providers):** 使一个应用可以访问另一个应用的数据（如联系人数据库），或者共享它们自己的数据。
- **视图系统 (View System):** 可以用来构建应用，它包括列表 (Lists)、网格 (Grids)、文本框 (Text Boxes)、按钮 (Buttons)，甚至可嵌入 Web 浏览器。
- **通知管理器 (Notification Manager):** 使应用程序可以在状态栏中显示自定义的提示信息。
- **资源管理器 (Resource Manager):** 提供非代码资源的访问，如本地字符串、图形和布局文件 (Layout Files)。

## (3) 系统运行时库

Android 包含一些 C/C++ 库，这些库能被 Android 系统中不同的组件使用。它们通过上层的应用框架为开发者提供服务。以下是一些核心库。

- **Surface Manager (门面管理器):** 对显示子系统的管理，并且为多个应用提供了 2D/3D 图层的无缝整合。
- **Media Framework (媒体框架):** 支持多种常用的音频、视频格式回放和录制，同时支持静态图像文件。编码格式包括 MPEG4、H.264、MP3、AAC、AMR、JPG 和 PNG 等。
- **SQLite:** 一款非常流行的嵌入式数据库，支持 SQL 语言，并且只占很少的内存，尤其适用于

手机这类内存受限的设备。

- **WebKit**: 一个最新的 Web 浏览器引擎, 支持 Android 浏览器和一个可嵌入的 Web 视图。
- **libc (系统 C 库)**: 一个从 BSD 继承来的标准 C 语言函数库, 它是专门为基于 Embedded Linux 的设备定制的。

#### (4) Linux 内核层

Android 是基于 Linux 的操作系统, 其架构的最底层是 Linux 内核, 但并不是通常的 GNU/Linux。因为在一般 GNU/Linux 里支持的功能, Android 大都不支持。另外, Google 为了达到商业目的, 必须移除被 GNU/GPL 授权证所约束的部分, 例如 Android 将驱动程序移到用户空间, 使 Linux 的驱动与内核彻底地分开了。

Android 所依赖的 Linux 内核功能包括安全 (Security)、存储器管理 (Memory Management)、进程管理 (Process Management)、网络堆栈 (Network Stack) 和驱动程序模型 (Driver Model) 等。

### 3. Android 平台的优势

#### (1) 开放性

Android 平台的最大优点首先就是其开放性, 开放的平台允许任何移动终端厂商加入到 Android 联盟中来。显著的开放性可以使其拥有更多的开发者, 随着智能手机的普及和手机应用的日益丰富, Android 平台也很快走向成熟。

开放性对于 Android 的发展而言, 有利于积累人气, 这里的人气包括消费者和厂商, 而对于消费者来讲, 最大的受益正是其丰富的软件资源。开放的平台也会带来更激烈的竞争, 如此一来, 消费者就可以用更低的价位购得心仪的手机。

#### (2) 多硬件兼容

这一点还是与 Android 平台的开放性相关, 由于 Android 的开放性, 众多的厂商会推出千奇百怪、功能各具特色的多种硬件产品。但硬件设备功能上的差异和特色, 却不会影响到数据同步和软件的使用, 如从小米手机改用三星, 同时还可将小米中优秀的软件也带到三星上使用, 联系人等资料更是可以方便地转移。

#### (3) 方便开发

Android 平台提供给第三方开发者一个十分宽泛、自由的环境, 不会受到任何条条框框的限制。而且 Android 平台支持二次开发, 开发者可以将现成的互联网应用, 如百度地图、新浪微博、微信等功能集成进自己的应用, 定制出更为强大的功能, 可想而知, 会有多少新颖别致的软件源源不断地诞生。

## 1.2 Android 开发平台

### 1. 主流开发平台

#### (1) Eclipse + ADT 方式

传统的 Android 开发使用 Eclipse, 它是一个开放源代码、基于 Java 的可扩展开发平台。就其本身而言, Eclipse 只是一个框架和一组服务, 用于通过插件组件构建开发环境。要开发 Android 程序, 就必须安装专为 Android 平台定制的 ADT (Android Development Tools) 插件。在 Eclipse 上安装 Android 开发环境, 首先需要安装支持 Java 应用程序运行的 Java 开发工具包 JDK (Java Development Kit), 然后再按顺序安装 Eclipse、Android SDK 和 ADT。由于 ADT 本身并未集成 Android 的 API, 需要用户根据实际情况自己上网选择更新所需 API 的 SDK 版本, 常常会出现安装的开发环境中各组件之间的兼容性问题, 搭建这样的环境对开发者的要求较高, 初学者往往不易上手。

近年来，Eclipse 官方也推出了专门针对 Android 开发的 Eclipse for Android Developers 版本，较之分立的 Eclipse + ADT 环境使用起来要方便一些，有兴趣的读者可以去 Eclipse 官网下载来试用。

## （2）Android Studio

2013 年 5 月 16 日，Google 推出新的 Android 开发环境——Android Studio，它基于 IntelliJ IDEA，类似 Eclipse ADT，但它的内部集成了 Android SDK，并提供了集成的 Android 开发工具用于开发和调试，无论是安装还是使用都比 Eclipse + ADT 方式要方便得多。目前最新版本为 Android Studio 2.2.3。

Android Studio 在 2014 年 12 月 8 日正式成为开发 Android 应用程序的官方集成开发环境(Integrated Development Environment, IDE) 工具，Android 开发官网也很明确地指出，原先 Eclipse + ADT 的开发模式已经被 Android Studio 取代，建议开发者尽早改用 Android Studio 以获得最好的支持。基于此，本书采用 Android Studio 作为书中 Android 实例程序的开发环境。

## 2. 软硬件要求

开发 Android 程序对计算机系统的要求较高，Android 各个版本的 SDK 包普遍较大，会占用一定的磁盘空间，且运行 Android 模拟器要耗费大量内存。鉴于此，强烈建议读者学习本书前先升级自己的计算机至较高配置。笔者所用计算机的配置如下。

处理器（CPU）：Intel(R) Core(TM) i5-4200U CPU @ 1.60GHz 2.30GHz，建议至少在 i5 或以上。

内存：8.00GB（7.89GB 可用）。

硬盘：700GB，其中 C 盘 180GB，D 盘 250GB，且 C 盘、D 盘均为高速固态硬盘。

操作系统：64 位 Windows 7 及以上版本。

## 1.3 Android Studio 2.x 的安装

### 1.3.1 在 Windows 上安装 Java 开发工具包

Android Studio 使用 Java 工具构建，因此在开始使用 Android Studio 之前，需要确保已经在自己的电脑上安装了 Java 开发工具包（JDK）。

#### 1. 下载 JDK

可以从 Oracle 官网下载到最新版本的 JDK，网址为 <http://www.oracle.com/technetwork/java/javase/downloads/index.html>，单击页面上的“Java DOWNLOAD”按钮，如图 1.2 所示。



图 1.2 JDK 下载页面

下一步要求选中“Accept License Agreement”单选按钮以接受许可协议。接着，需要选择适合自己操作系统的 JDK。笔者使用的是 64 位 Windows 7，故单击 Windows x64 标签右侧的文件链接“jdk-8u121-windows-x64.exe”，如图 1.3 所示（Oracle 经常会发布 JDK 的更新版本，到本书出版的时候，JDK 应该已经有了更新的版本，因此务必下载最新版），等待安装文件下载完成。下载得到的文件名为“jdk-8u121-windows-x64.exe”，这个文件的大小通常为 200MB 左右，所以下载过程不会花费太长时间。

Java SE Development Kit 8u121		
You must accept the Oracle Binary Code License Agreement for Java SE to download this software.		
Thank you for accepting the Oracle Binary Code License Agreement for Java SE; you may now download this software.		
Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	77.86 MB	<a href="#">jdk-8u121-linux-arm32-vfp-hflt.tar.gz</a>
Linux ARM 64 Hard Float ABI	74.83 MB	<a href="#">jdk-8u121-linux-arm64-vfp-hflt.tar.gz</a>
Linux x86	162.41 MB	<a href="#">jdk-8u121-linux-i586.rpm</a>
Linux x86	177.13 MB	<a href="#">jdk-8u121-linux-i586.tar.gz</a>
Linux x64	159.96 MB	<a href="#">jdk-8u121-linux-x64.rpm</a>
Linux x64	174.76 MB	<a href="#">jdk-8u121-linux-x64.tar.gz</a>
Mac OS X	223.21 MB	<a href="#">jdk-8u121-macosx-x64.dmg</a>
Solaris SPARC 64-bit	139.64 MB	<a href="#">jdk-8u121-solaris-sparcv9.tar.Z</a>
Solaris SPARC 64-bit	99.07 MB	<a href="#">jdk-8u121-solaris-sparcv9.tar.gz</a>
Solaris x64	140.42 MB	<a href="#">jdk-8u121-solaris-x64.tar.Z</a>
Solaris x64	96.9 MB	<a href="#">jdk-8u121-solaris-x64.tar.gz</a>
Windows x86	189.36 MB	<a href="#">jdk-8u121-windows-i586.exe</a>
Windows x64	195.51 MB	<a href="#">jdk-8u121-windows-x64.exe</a>

图 1.3 接受许可协议并单击适用于 Windows 的链接

## 2. 执行 JDK 向导

导航到浏览器下载安装文件的位置，并双击执行该文件。一旦安装开始，将会看到安装向导，如图 1.4 所示。单击“下一步”按钮，系统进入指定安装目录对话框。在 Windows 中，JDK 安装程序的默认路径为 C:\Program Files\Java\。要更改安装目录的位置，可单击“更改”按钮。本书安装到默认路径，参见图 1.5。



图 1.4 Windows 中的 JDK 安装向导



图 1.5 选择 JDK 的安装目录

记住安装 JDK 的位置。单击“下一步”按钮，向导提示安装 Java 运行时环境（Java Runtime Environment, JRE）的目标文件夹（C:\Program Files\Java\jre1.8.0\_121），默认是与 JDK 相同的目录，同样可单击“更改”按钮选择其他位置。本书保持默认，单击“下一步”按钮，如图 1.6 所示。



图 1.6 选择 JRE 的目标文件夹

开始安装进程，直到显示安装完成对话框，单击“关闭”按钮，完成安装。

### 3. 配置 Windows 环境变量

配置 Windows 环境变量的目的是为了后面安装的 Android Studio 开发平台能够找到 JDK。

在桌面上右击“计算机”图标，从弹出的菜单中选择“属性”，打开“系统”窗口，如图 1.7 所示。



图 1.7 Windows 系统窗口

单击“高级系统设置”选项，系统显示“系统属性”对话框，如图 1.8 所示。单击“环境变量”按钮，系统显示当前环境变量的情况，如图 1.9 所示。

#### (1) JAVA\_HOME 环境变量新建/编辑

在底部列出的“系统变量”列表中，如果 JAVA\_HOME 项不存在，单击“新建”按钮创建它。系统显示新建系统变量对话框，在“变量名”文本框中输入 JAVA\_HOME，在“变量值”文本框中，输入之前安装 JDK 的位置 C:\Program Files\Java\jdk1.8.0\_121，单击“确定”按钮，如图 1.10 所示。



图 1.8 系统属性

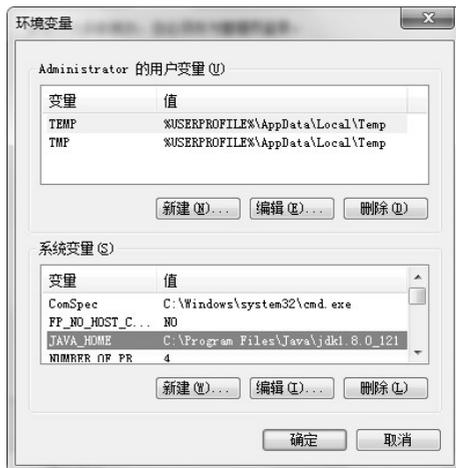


图 1.9 环境变量



图 1.10 新建 JAVA\_HOME 环境变量

以后可以选择 JAVA\_HOME 项，单击“编辑”按钮进行修改。

## (2) Path 环境变量编辑

Path 环境变量系统已经存在，需要编辑 Path 环境变量，参见图 1.11。将光标置于“变量值”文本框中内容的末尾并输入以下内容：

```
;C:\Program Files\Java\jdk1.8.0_121\bin
```

或者

```
;%JAVA_HOME%\bin
```

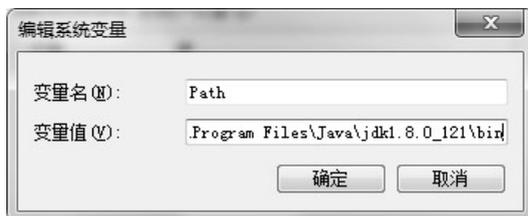


图 1.11 编辑 Path 环境变量

连续三次单击“确定”按钮，Windows 接受这些修改并返回到最初的“系统”窗口。这样，系统就在原来的 Path 路径上增加了一个查找路径。

### 注意：

在 Windows 10 系统中，编辑 Path 环境变量的界面如图 1.12 所示。

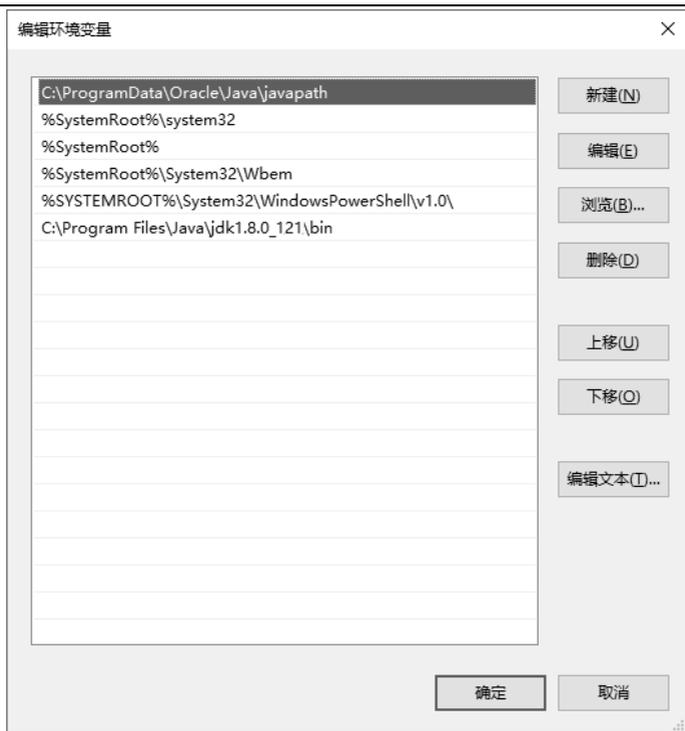


图 1.12 在 Windows 10 系统中编辑 Path 环境变量

### （3）测试 JDK 是否正确安装

为了测试新的 JDK 是否已正确安装，单击 Windows 系统的“开始”→“运行”命令，在弹出的对话框中输入 cmd，接着按 Enter 键打开命令行。在命令行窗口中输入以下命令并按 Enter 键：

```
java -version
```

如果得到如图 1.13 所示的响应，表示已经正确地安装了 JDK。



图 1.13 确认 JDK 安装正确

## 1.3.2 安装 Android Studio

对于广大国内用户来说，获取和学习 Android Studio 最便捷的途径是访问 Android 的中文社区（网址：<http://www.android-studio.org/>），其与 Android 官网（国外）是同步更新的，旨在让 Android 对中文开发者更友好。

## 1. 下载 Android Studio

在 Android 中文社区首页上，单击“下载 ANDROID STUDIO”按钮开始下载。

### 注意:

下载得到的文件名形如“android-studio-bundle-xxx.xxxxxxx-windows.exe”（其中 x 代表数字），文件大小在 1.6GB 左右（此为集成了 Android SDK 的版本）。我们下载的文件为“android-studio-bundle-145.3537739-windows.exe”。

如果只有几百兆，说明其中不含 Android SDK，用户还要单独去下载 Android SDK 来安装，随之而来的各种兼容性问题将十分棘手！故强烈建议初学者要下载使用集成好 Android SDK 的 Android Studio 版本。

## 2. 执行安装向导

下载完毕后，双击执行得到的文件，启动安装向导。注意，因为在 Android Studio 的安装过程中需要时刻到网络获得所需的各种文件，为了防止出现麻烦，建议关闭 Windows 防火墙。

在安装向导启动后，单击“Next”按钮向前推进界面，一直到达“Choose Components”界面。在这里，选中所有组件复选框，如图 1.14 所示。

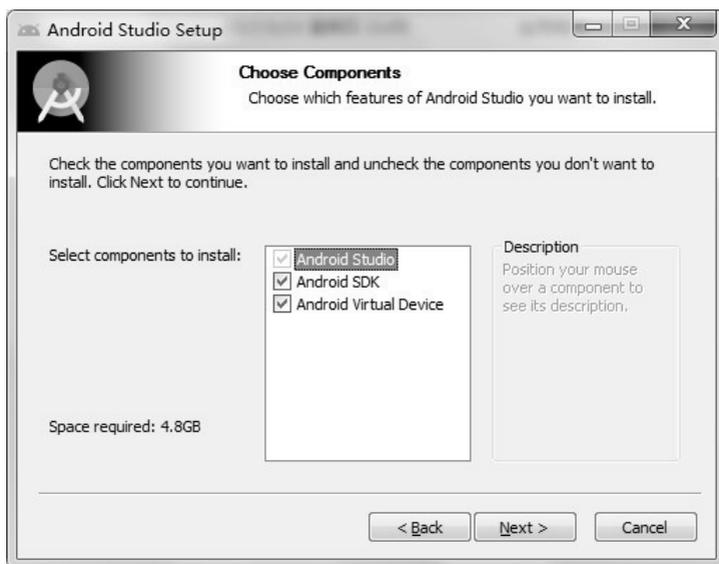


图 1.14 选择组件

单击“Next”按钮，继续向前推进安装界面。在“LicenseAgree”对话框单击“I Agree”按钮接受各种条款。当到达“Configuration Settings Install Locations”界面时，如图 1.15 所示，选择 Android Studio 和 Android SDK 的安装位置，本书采用默认。

单击“Next”按钮，系统显示“Choose Start Menu Folder”对话框，让用户选择是否在系统的开始菜单中创建“Android Studio”快捷菜单项。这里选择默认创建快捷菜单项。

如图 1.16 所示，单击“Install”按钮，系统开始安装，直到安装完成，显示“Installation Complete”。单击“Next”按钮，到达“Completing Android Studio Setup”界面，如图 1.17 所示。“Start Android Studio”复选框能够让 Android Studio 在单击“Finish”之后启动。确保选中了该复选框，接着继续单击“Finish”，Android Studio 将会启动。此后，将通过桌面图标或开始菜单来启动 Android Studio。



图 1.15 为 Android Studio 和 SDK 选择安装位置



图 1.16 “开始”菜单“Android Studio”菜单项



图 1.17 完成 Android Studio 的安装

### 3. 第一次启动

当 Android Studio 第一次启动时，它会检查用户的系统之前是否安装过早期版本，并询问用户是否要导入先前版本 Android Studio 的设置，如图 1.18 所示。一般初学者建议使用初始设置，选中下面一个单选按钮后，单击“OK”按钮。

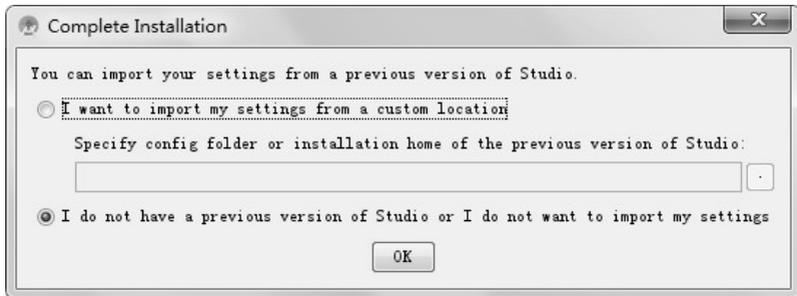


图 1.18 Android Studio 的初始设置

接着出现启动画面，首先弹出的是 Android Studio Setup Wizard 的 Welcome（欢迎）界面，如图 1.19 所示。Setup Wizard 将会分析用户的系统，查找已有 JDK（例如之前安装的那个）以及 Android SDK 的位置。单击“Next”按钮。



图 1.19 Setup Wizard 欢迎界面

在接下来的对话框中，选择安装类型为 Standard（标准），如图 1.20 所示。单击“Next”按钮。在最终确认界面上，汇总显示了开发环境将要下载安装的全部 Android SDK 组件的详细信息，如图 1.21 所示。单击“Finish”按钮。

Setup Wizard 会下载在 Android Studio 中开发应用需要的所有组件，如图 1.22 所示。稍等一会儿，待完成后单击“Finish”按钮，关闭 Setup Wizard。

至此，Android Studio 开发环境全部安装完成，可以使用了。

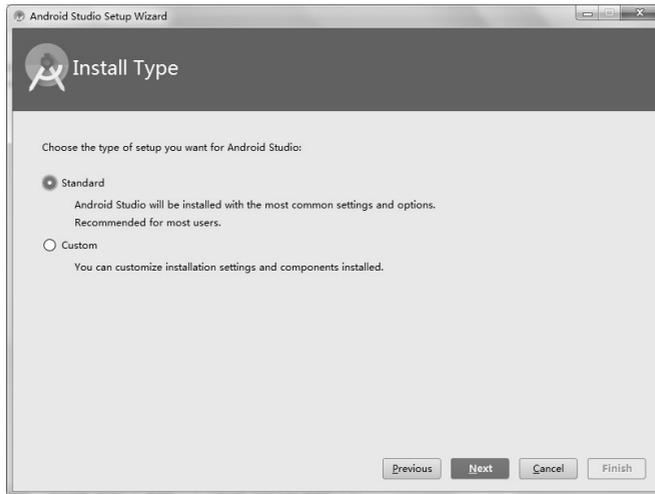


图 1.20 选择安装类型

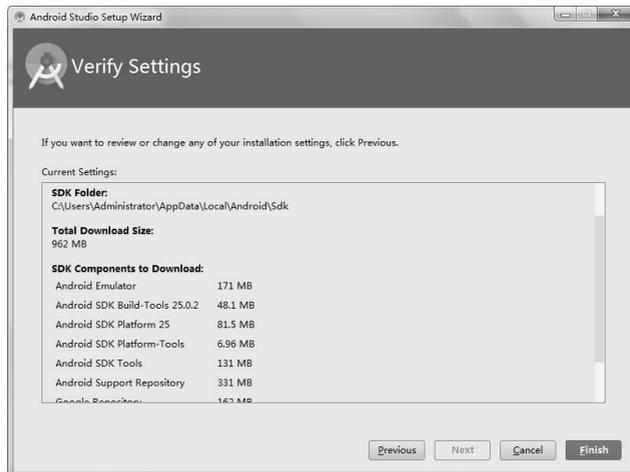


图 1.21 将要下载安装的 Android SDK 组件

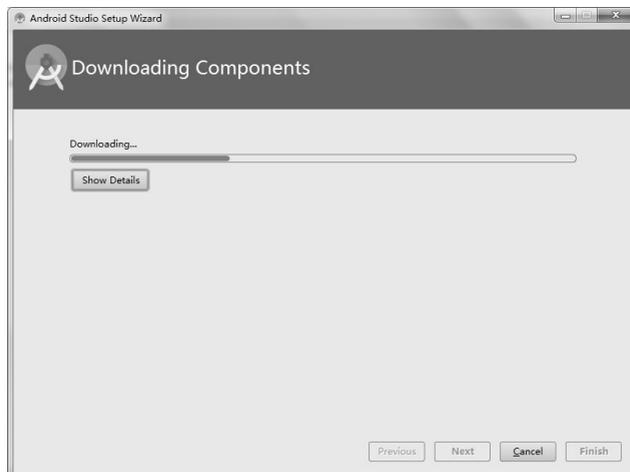


图 1.22 Setup Wizard 下载组件