

工程师经验手记

Android应用开发精解

高磊 杨戈 元凯 编著



北京航空航天大学出版社
BEIHANG UNIVERSITY PRESS

内 容 简 介

本书以 Android 应用开发所需要的技术为线索,先后讲解了 SDK 一些工具的使用、UI 开发、Service 开发、数据存储、图形动画、网络开发、硬件操作、结合 Web 开发以及如何对应用进行优化等。本书并没有从基础的 Eclipse 安装、Hello World 等讲起,而是以 Android 应用开发过程中常用的技术作为主线去讲解,很多内容需要读者有一定的开发经验才能够很好地理解。

本书在对各种技术阐述的过程中,并不是仅仅简单地把相关知识点讲完就结束了,而是在各知识点后面分享了大量项目实践的经验。这些都是作者在项目过程中思考的成果,是多年从事软件开发工作的技术沉淀,是项目实践过程中的精华。

本书适合有一定 Java 基础并且想要自学 Android 开发的编程爱好者、想要转行从事 Android 领域开发的程序员,以及正在从事 Android 的初学者阅读。

图书在版编目(CIP)数据

Android 应用开发精解 / 高磊,杨诚,元凯编著. --
北京:北京航空航天大学出版社,2012.9
ISBN 978-7-5124-0918-7

I. ①A… II. ①高… ②杨… ③元… III. ①移动终端—应用程序—程序设计 IV. ①TN929.53

中国版本图书馆 CIP 数据核字(2012)第 195394 号

版权所有,侵权必究。

Android 应用开发精解

高磊 杨诚 元凯 编著

责任编辑 董立娟

*

北京航空航天大学出版社出版发行

北京市海淀区学院路 37 号(邮编 100191) <http://www.buaapress.com.cn>

发行部电话:(010)82317024 传真:(010)82328026

读者信箱:emsbook@gmail.com 邮购电话:(010)82316936

涿州市新华印刷有限公司印装 各地书店经销

*

开本:710×1 000 1/16 印张:19 字数:405 千字

2012 年 9 月第 1 版 2012 年 9 月第 1 次印刷 印数:4 000 册

ISBN 978-7-5124-0918-7 定价:39.00 元

若本书有倒页、脱页、缺页等印装质量问题,请与本社发行部联系调换。联系电话:(010)82317024

前言

如今的 Android 市场异常火爆,Android 手机已经占据智能手机操作系统的半壁江山了。市场的火爆也带动了 Android 软件的发展,越来越多的开发者开始学习 Android 开发,开始涌入到 Android 开发的各个领域,这其中有一些软件开发经验丰富的开发者,也有一些刚刚毕业或者还在学校中的学生。然而,总的来说,大部分开发者在 Android 平台上的开发时间都不算很多,开发经验不足,需要不断在实际项目中学习和沉淀。

如果您也想学习 Android 开发,或者正在学习 Android 开发,您一定对下面的问题最关心:Android 开发到底难不难? 这个问题没有标准的答案,笔者的理解,可以说简单,也可以说难。

为什么说简单? 这是因为 Android 开发主要使用 Java 语言,对于有 Java 基础的开发者入门很容易;即使原来没有基础,Java 语言学习起来相对也容易些。还有 Android 的开发成本相对较低,只要有一台 PC 机就可以用模拟器学习开发了。

为什么说难? 这是因为 Android 市场的复杂性。一个是 Android SDK 的版本兼容问题。由于市场上 Android 设备的 SDK 版本各不相同,开发软件时就不得不考虑不同 SDK 版本的兼容问题。一个是硬件兼容问题。Android 设备的硬件支持五花八门,包括 CPU、内存、屏幕分辨率、是否支持某些硬件特性等,开发软件时就不得不考虑市场上已有的各种 Android 硬件设备的兼容性问题。一个就是定制 ROM 的问题。Android 是开源的,设备厂商可以修改 Android 的代码并且定制在自己的设备中,从而导致了大多数 Android 设备都使用了定制过的 ROM,这就需要开发者花更多的时间和精力来解决定制 ROM 的各种问题。还有一个就是 Android 在国内的开发资料相对较少,很多时候需要去国外网站,或者直接阅读 Android 的源代码来解决某些问题,这就增加了学习 Android 的难度。

总体来说,开发一个 Android 应用并不难,开发出一个优秀的能够经受住市场考验的 Android 应用却并不容易,需要开发者有丰富的 Android 开发经验。无论您是想从头开始学习 Android 应用开发,还是想学习积累经验,本书都是适合的。

本书特点

现在市场上的 Android 书籍,要么以讲解知识点为主,点到为止;要么是以某虚拟项目为主,介绍项目的同时附上大量的源代码,很少会把项目过程中积累的实际经验写进去。本书最大的特点就是除了讲解 Android 应用开发的各知识点以外,还分享了大量的实际经验;这些经验一般都无法从现有的书籍或者网络上获得,这些都是在项目过程中思考的成果,是多年从事软件开发工作的技术沉淀。

本书作者

本书的第 1~4 章由元凯编写,第 5~9 章由杨诚编写,第 10~12 章由高磊编写,全书由高磊审阅。由于本书涉及知识较多,而作者水平有限,很难全部精通,难免有疏漏之处,如果读者朋友发现错误,请批评指正(作者邮箱:gaolei021@gmail.com),非常感谢。

鸣 谢

最后在此感谢现在和曾经一起日夜奋战的兄弟姐妹们,感谢出版社的各位同仁,也感谢家人默默的理解和支持。特别的,感谢刘雪莲、禅延玲、李洁,谢谢你们。

编 者
2012.4



第 1 章 工欲善其事 必先利其器—Android SDK 工具	1
1.1 巧妇难为无米之炊—Android SDK 的安装	1
1.2 设备管理工具—调试桥(ADB)	3
1.2.1 ADB 简介	3
1.2.2 ADB 常用命令	4
1.3 没有真机一样开发—Android 模拟器	7
1.4 Android 调试—调试工具 DDMS	9
1.5 UI 布局分析工具—视图工具(Hierarchy Viewer)	14
1.6 Log 打印—Log 输出工具 logcat	17
1.6.1 启动 logcat	17
1.6.2 过滤日志输出	17
1.6.3 控制日志输出格式	18
1.6.4 查看可用日志缓冲区	19
1.7 图片拉伸不失真—九宫格绘制工具 Draw 9 - Patch	20
1.7.1 什么是“点九”文件	20
1.7.2 点九文件的制作	20
第 2 章 吸引你的眼球—UI 编程	24
2.1 UI 基础—常用 UI 组件	24
2.1.1 文本显示—文本框(TextView)	24
2.1.2 按钮(Button)	27
2.1.3 文本编辑—编辑框(EditText)	29
2.1.4 图片显示—图片视图(ImageView)	32
2.1.5 多项选择—多选框(CheckBox)和单项选择—单选框(RadioButton)	34
2.1.6 图片拖动—拖动效果(Gallery)	38
2.1.7 列表组件(ListView)	41
2.2 彰显你的个性—自定义 UI 组件	48
2.3 简单明了的消息提示框(Toast)和对话框(Dialog)	51
2.3.1 Toast 提示	51

2.3.2	Dialog 提示	52
2.4	Menu 键的呼唤—Menu 菜单	56
第 3 章	界面 UI 的基石—UI 布局	61
3.1	用户界面的基本单元—View 视图	61
3.2	百花齐放—各种 Layout 布局	64
3.2.1	Layout 布局的简单介绍	64
3.2.2	线性布局(LinearLayout)	64
3.2.3	相对布局(RelativeLayout)	67
3.2.4	框架布局(FrameLayout)	70
3.2.5	表单布局(TableLayout)	72
3.2.6	绝对布局(AbsoluteLayout)	74
3.3	样式(Style)和主题(Theme)的使用	76
3.3.1	样式(Style)的使用	76
3.3.2	主题(Theme)的使用	77
第 4 章	Android 开发三大基石—Activity、Service 和 Handler	79
4.1	应用程序的接口—Activity 窗口	79
4.1.1	Activity 生命周期	79
4.1.2	Activity 栈	81
4.1.3	Activity 的创建	81
4.1.4	Activity 的 4 种加载模式	82
4.1.5	Activity 交互—Activity 跳转	83
4.1.6	Activity 中数据传递	86
4.2	千变万化的服务—Service 开发	87
4.2.1	Service 的生命周期	87
4.2.2	Service 的启动和停止	88
4.2.3	我的服务我来用—本地服务开发	89
4.2.4	开机自启动的服务	93
4.3	Android 线程间的通信—消息机制	95
4.3.1	消息的传递—Handler 的使用	95
4.3.2	Android 中消息机制的详细分析	97
第 5 章	以数据为中心—数据存取	100
5.1	文件操作	100
5.1.1	读写一般的文本文件	100
5.1.2	结构性的文件—读写 XML 文件	103
5.1.3	自由操作,随心所欲—序列化和反序列化	113

5.2 通用的数据操作方式—数据库	116
5.2.1 SQLite 数据库介绍	116
5.2.2 创建并打开数据库	116
5.2.3 添加、删除和修改操作	117
5.2.4 游标的操作—使用 Cursor	120
5.3 安全方便简单—使用 SharedPreferences	121
5.4 我的数据大家用—ContentProvider、ContentResolver	123
第6章 一张白纸好作画—Canvas 画布	126
6.1 Canvas 画布简介	126
6.1.1 View Canvas—使用普通 View 的 Canvas 画图	126
6.1.2 Bitmap Canvas—使用普通 Bitmap 的 Canvas 画图	128
6.1.3 SurfaceView Canvas—使用 SurfaceView 的 Canvas 画图	128
6.2 Canvas 常用绘制方法	131
6.3 对 Canvas 进行变换	133
6.4 Canvas 绘制的辅助类	134
6.4.1 画笔 android.graphics.Paint	134
6.4.2 字体 android.graphics.Typeface	135
6.4.3 颜色 android.graphics.Color	136
6.4.4 路径 android.graphics.Path	137
6.4.5 路径的高级效果 android.graphics.PathEffect	139
6.4.6 点类 android.graphics.Point 和 android.graphics.PointF	141
6.4.7 形状类 android.graphics.Rect 和 android.graphics.RectF	142
6.4.8 区域 android.graphics.Region 与 Region.Op	144
6.4.9 千姿百态,矩阵变换 android.graphics.Matrix	145
第7章 实现炫酷效果—图像和动画	149
7.1 Android 的几种常用图像类型	149
7.2 图片的基础—Bitmap(位图)	150
7.2.1 如何获取位图资源	150
7.2.2 如果获取位图的信息	151
7.2.3 位图的显示与变换	152
7.3 变化多端—Drawable(绘图类)	154
7.3.1 Drawable 的一些常用子类	154
7.3.2 BitmapDrawable	154
7.3.3 点九图片—NinePatchDrawable	155
7.3.4 会动的图片—StateListDrawable	156

7.3.5	颜色填充的另一种方法—PaintDrawable	157
7.3.6	超炫的特效—ShapeDrawable	157
7.3.7	简单的帧动画—AnimationDrawable	165
7.4	轻量级图片—Picture	167
7.5	Drawable、Bitmap、byte[]之间的转换	167
7.6	Android 提供的几种动画效果(Animation)	168
7.7	渐变动画—Tween Animation	169
7.7.1	Tween Animation 简介	169
7.7.2	Tween Animation 共同的属性	169
7.7.3	淡进淡出—AlphaAnimation	170
7.7.4	忽大忽小—ScaleAnimation	171
7.7.5	平移—TranslateAnimation	172
7.7.6	旋转—RotateAnimation	173
7.7.7	实现一个自己的 TweenAnimation	174
7.8	渐变动画—Frame Animation	176
7.9	随意组合动画效果—AnimationSet	177
7.10	加速的工具—Interpolator	178
第8章	网络的时代—网络开发	180
8.1	Android 中网络开发概述	180
8.2	直接基于 Socket 编程	181
8.2.1	Socket 编程简介	181
8.2.2	基于 TCP 协议的 Socket 编程	183
8.2.3	基于 UDP 协议的 Socket 编程	184
8.3	基于最成熟的 Web 协议—HTTP 协议编程	185
8.3.1	HTTP 协议简介	185
8.3.2	使用 URL 类读取 HTTP 资源	187
8.3.3	使用 HttpURLConnection 类访问 HTTP 资源	189
8.3.4	使用 Apache 的 HttpClient	190
8.4	耗时操作的通用方式—多线程与异步处理	192
8.4.1	多线程和异步处理简介	192
8.4.2	Handler 方式	193
8.4.3	AsyncTask 类实现后台任务的处理	197
8.5	轻量级的数据交换格式—JSON	199
8.5.1	客户端与服务器端的数据交互	199
8.5.2	XML 格式与 JSON 格式的比较	201

8.5.3 解析 JSON 格式数据	202
第 9 章 多语言环境的支持和多屏幕的适配	205
9.1 Android 程序的资源文件	205
9.1.1 资源文件的目录结构	205
9.1.2 资源文件目录的修饰语	206
9.1.3 程序加载资源文件的步骤	208
9.2 国际化和本地化的支持	209
9.3 多屏幕的适配	210
9.3.1 屏幕参数的基本概念	210
9.3.2 屏幕参数的各种单位及相互转换	211
9.3.3 处理屏幕自适应的方法	212
9.3.4 详细说明 Density	214
第 10 章 利用手机特性—结合硬件进行开发	217
10.1 炫酷的人机交互—触摸和手势	217
10.1.1 实现滑动翻页—使用 ViewFlipper	217
10.1.2 支持多个手指一起操作—实现多点触摸	222
10.1.3 识别手势—使用 GestureDetector	227
10.2 我在哪里—使用定位功能	230
10.3 电话拨打和短信发送	233
10.3.1 调用系统的电话拨打功能	233
10.3.2 实现发送短信的功能	234
10.4 拍照和摄像	235
10.5 使用传感器	238
10.5.1 传感器概述	238
10.5.2 加速度传感器	241
10.5.3 方向传感器	242
10.5.4 其他传感器	243
第 11 章 避重就轻—结合 Web 开发 Android 应用	245
11.1 Android 上的 Web 应用概述	245
11.2 使用 WebView 载入 Web 页面	247
11.2.1 Webkit 引擎和 WebView 控件	247
11.2.2 浏览基本的 Web 页面	247
11.2.3 开启对于 JavaScript 的支持	249
11.2.4 监听 Web 页面的载入	250
11.2.5 让 WebView 支持文件下载	250

11.3 本地代码与 Web 页面交互	252
11.3.1 向 Web 页面传递数据	252
11.3.2 本地代码调用 Web 页面 JavaScript 方法	254
11.3.3 Web 页面调用本地 Java 方法	254
11.4 Web 页面的 JavaScript 调试	259
11.5 常用移动设备 Web 开发框架	260
11.5.1 jQuery Mobile 框架简介	260
11.5.2 Sencha Touch 框架简介	262
11.5.3 PhoneGap 开发平台简介	262
第 12 章 细节决定成败—Android 应用程序的优化	264
12.1 对应用内存的优化	264
12.1.1 Android 程序的内存概述	264
12.1.2 追踪内存—使用内存优化辅助工具	266
12.1.3 吃内存大户—Bitmap 的优化	270
12.1.4 想回收就回收—使用软引用和弱引用	274
12.1.5 注重细节—从代码角度进行优化	277
12.2 对界面 UI 的优化	281
12.2.1 多利用 Android 系统的资源	281
12.2.2 抽取相同的布局	284
12.2.3 精简 UI 层次	285
12.2.4 界面延迟加载技术	287
12.3 留条后路—对 Crash 进行处理	288
12.3.1 为什么需要捕获 Crash	288
12.3.2 如何捕获和处理 Crash	289
参考文献	292

第 1 章

工欲善其事 必先利其器 —Android SDK 工具

在工作和生活中,合理运用手头上的工具往往能够使我们要做的事情达到事半功倍的效果。Android 应用程序的开发也一样,善于使用辅助工具,可以使开发过程更加清晰、流畅,可以更快地发现程序中存在的问题,写出更加高效、合理的代码。本章将从介绍一些常用的 Android SDK 工具开始,和您一起体验 Android 应用开发学习之旅。

1.1 巧妇难为无米之炊—Android SDK 的安装

由于 Android 开发和模拟器的运行等都需要在 Java 环境下才能运行,因此先要搭建好 Java 环境(Java 环境的搭建这里就不介绍了,读者可参考相关图书)。

搭建好 Java 环境之后就开始下载和安装 Android SDK 了(SDK 可以从 <http://developer.android.com/sdk/index.html> 网站下载)。下载好 Android SDK 之后把它解压到某个目录,如:“E:\android-sdk”(目录不要有中文),然后把“E:\android-sdk\tools”完整路径加入到系统 Path 变量(这一步主要用于以后安装 apk 软件时方便调用)。

Windows 操作系统下双击“SDK Setup. exe”就可以开始在计算机上运行 Android 模拟器,SDK 会从 Google 的服务器检查可更新的套件。如果看到 SSL 错误信息,则找到后台运行的 Android SDK and AVD Manager 窗口,并单击左侧的设置标签。这里取消“Force https://...sources to be fetched using http://...”选项,如图 1-1 所示,单击“确定”,然后重新打开安装程序。

此时将会安装一些包。我们可以有选择性地安装一些包,然后单击 Install 按钮安装 Android SDK,如图 1-2 所示。

默认情况下,所有的 SDK 平台、例程、API 等都会被安装,这可能需要相当长的时间来下载所有可用的 Android 版本。如果仅仅是为了体验一下 Android,则选择需

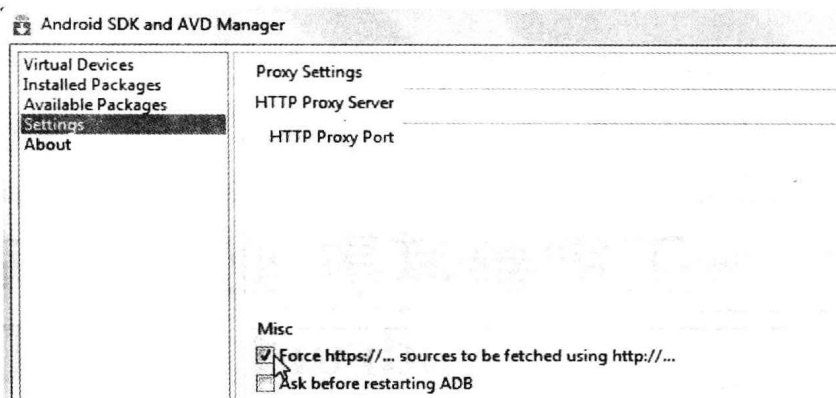


图 1-1 取消默认选择

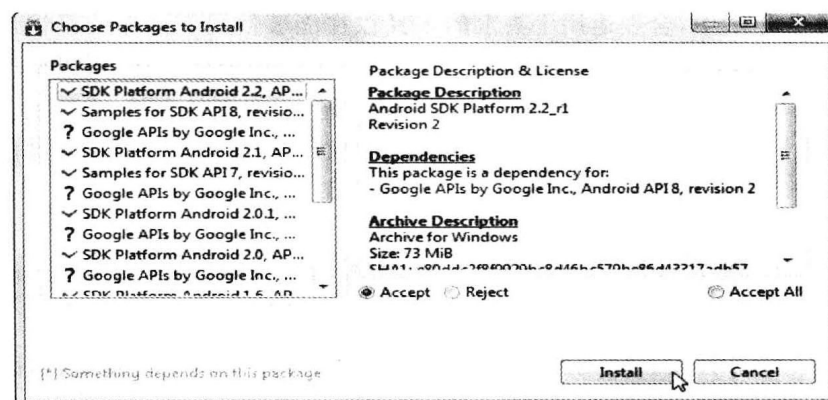


图 1-2 选择安装包

要安装版本即可。然后选择其他不需要安装的包，并单击 Reject 按钮，如图 1-3 所示。单击 Install 按钮进行安装。当然，可以下载我们需要的版本进行下载安装。

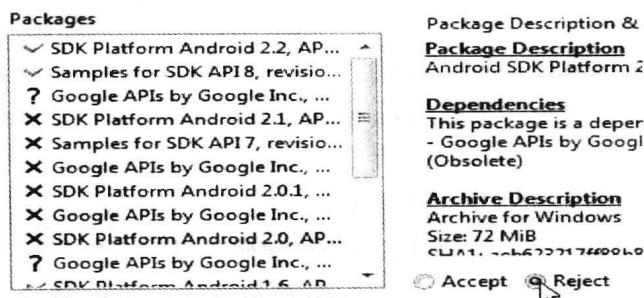


图 1-3 选择安装版本

开始安装后会出现一个窗口，显示下载和安装的进度，如图 1-4 所示。该过程

可能花费比较长的时间,请耐心等待。

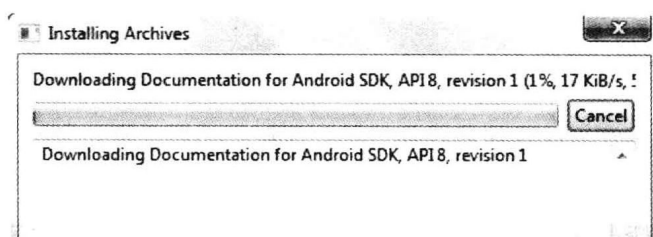


图 1-4- 安装进度界面

下载完成后,SDK 就安装成功了,就能在计算机上进行 Android 开发或者是安装模拟器来测试 Android 了。

经验分享:

除了需要安装 Android SDK 以外,建议开发者同时也下载 Android 的源码,这是因为开发过程中可能经常查看源码。可以在 <http://source.android.com/> 站点使用 Git 工具下载最新的源代码。

1.2 设备管理工具—调试桥(ADB)

1.2.1 ADB 简介

ADB 全称是 Android Debug Bridge,是 Android SDK 里自带的一个工具,用这个工具可以直接操作管理 Android 模拟器(Emulator)或者是真实的 Android 设备。

ADB 的功能很多,我们主要用到的功能有:

- ① 运行设备的 shell 命令行;
- ② 管理模拟器或设备的端口映射;
- ③ 计算机和设备之间上传/下载文件;
- ④ 将本地 apk 软件安装至模拟器或 Android 设备。

ADB 是一个客户端—服务器端程序,其中客户端是用来操作的计算机,服务器端是 Android 设备。要使用 ADB 工具,就先要把手机上的 USB 调试项打开,具体的操作步骤是:设置→应用程序→开发→USB 调试(可能设备不同,具体的操作步骤也略有差别)。

1.2.2 ADB 常用命令

1) 查看设备

adb devices

这个命令是查看当前连接的设备,显示当前连接到计算机的所有 Android 设备和模拟器,如图 1-5 所示。

```
D:\tools\android-sdk\android-sdk\platform-tools>adb devices
List of devices attached
192.168.28.110:5555    device
800A354043045403914  device
```

图 1-5 查看设备的命令窗口

2) 安装软件

adb install <apk 文件路径>

这个命令将指定的 apk 文件安装到设备上,如图 1-6 所示。

```
D:\tools\android-sdk\android-sdk\platform-tools>adb install myveat~1_signed.apk
530 KB/s (491852 bytes in 0.906s)
pkg: /data/local/tmp/myveat~1_signed.apk
Success
```

图 1-6 安装软件命令窗口

3) 卸载软件

adb uninstall <软件包名>

adb uninstall -k <软件包名>

如果加 -k 参数,为卸载软件但是保留配置和缓存文件,如图 1-7 所示。

```
D:\tools\android-sdk\android-sdk\platform-tools>adb uninstall com.weather.manyo
re13
Success
```

图 1-7 卸载软件命令窗口

这里特别需要注意的是,安装时后面跟的参数是 apk 文件路径,而卸载时参数则为软件包名。

4) 登录设备 shell

adb shell

adb shell <command 命令>

这个命令将登录设备的 shell,后面加<command 命令>将是直接运行设备命令,相当于执行远程命令,如图 1-8 所示。


```
D:\tools\android-sdk\android-sdk\platform-tools>adb shell ls
config
cache
sdcard
acct
mnt
```

图 1-8 登陆设备 shell 命令窗口

5) 从计算机复制文件到设备

adb push <本地路径> <远程路径>

用 push 命令可以把本台计算机上的文件或者文件夹复制到设备,如图 1-9 所示。

```
D:\tools\android-sdk\android-sdk\platform-tools>adb push d:/123.txt mnt/sdcard
```

图 1-9 复制文件或者文件夹命令窗口

6) 从设备上下载文件到计算机

adb pull <远程路径> <本地路径>

用 pull 命令可以把设备上的文件或者文件夹复制到本台计算机,如图 1-10 所示。

```
D:\tools\android-sdk\android-sdk\platform-tools>adb pull mnt/sdcard/123.txt d:/
```

图 1-10 从设备下载文件命令窗口

7) 显示帮助信息

adb help

这个命令将显示帮助信息,如图 1-11 所示,要是有些命令及参数不是很熟悉,可以从帮助信息中找到答案。

```
D:\tools\android-sdk\android-sdk\platform-tools>adb help
Android Debug Bridge version 1.0.26

-d                                - directs command to the only connected USB device
e                                returns an error if more than one USB device is
present.
```

图 1-11 显示帮助信息命令窗口

8) 连接设备

adb connect <设备 IP>

如图 1-12 所示。

```
D:\tools\android-sdk\android-sdk\platform-tools>adb connect 192.168.28.110  
connected to 192.168.28.110:5555
```

图 1-12 连接设备命令窗口

9) 断开当前连接

```
adb kill-server
```

如图 1-13 所示。

```
D:\tools\android-sdk\android-sdk\platform-tools>adb kill-server
```

图 1-13 断开连接命令窗口

经验分享：

有的时候,我们可能并不需要 ROM 自带的一些软件,想把它卸载。可是 Android 系统并没有卸载 ROM 自带软件的功能,这个时候也可以用 ADB 来卸载这些软件,步骤如下:

- ① 取得手机 root 权限;
- ② 下载 Android_db.rar,解压到 %windir/%System32 下;
- ③ 手机连接数据线,在计算机上打开 cmd,然后输入命令:

```
adb remount  
adb shell  
su
```

执行完成之后,则看到:

```
* daemon not running. starting it now *  
* daemon started successfully *
```

- ④ 接着就是 Linux 命令行模式了,输入:

```
cd system/app
```

则发现没什么变化,然后输入 ls 回车。

这时候列表显示了 system/app 里面的所有文件,也就是 ROM 集成的一些软件了。

- ⑤ 删除命令:

```
rm 文件名
```

另外,需要特别注意的是,对于那些并不了解的文件,请不要随意地删除,避免手机出现问题。

经验分享：

在使用 Eclipse 开发 Android 应用过程中,有时候调试过程中可能会发现报错,错误信息大意是连接不上 adb Server 了。此时可以在任务管理器中结束 adb.exe 进程,然后重新启动 Eclipse。

1.3 没有真机一样开发—Android 模拟器

有些时候,我们手头上可能并没有符合要求的 Android 设备,那这时候是不是对调试或者开发就一筹莫展了呢?当然不是。因为我们有 Android 模拟器!这里先介绍一下 Android SDK 自带的模拟器。

Android 模拟器是 Android SDK 自带的移动设备模拟器,是一个可以运行在计算机上的一个虚拟设备,可以模拟除了接听和拨打电话外所有移动设备上的典型功能和行为,可以让你不需要使用物理设备就能简单地预览、开发和测试 Android 应用程序。

之前已经介绍过了如何安装 Android SDK,现在就安装模拟器来测试 Android。选择左侧导航菜单的 Virtual Devices,再右击 New 按钮,这时弹出创建框,输入模拟器的名字,并从下拉菜单选择所需的 Android 版本,这里只会显示安装时选择了的 Android 版本。输入 SD 卡大小,这只是一个虚拟的 SD 卡,实际上是将你的设置和文件存储到一个 IMG 文件。然后,选择屏幕大小,默认方式显示,设置完成后,单击 Create AVD 按钮,如图 1-14 所示。

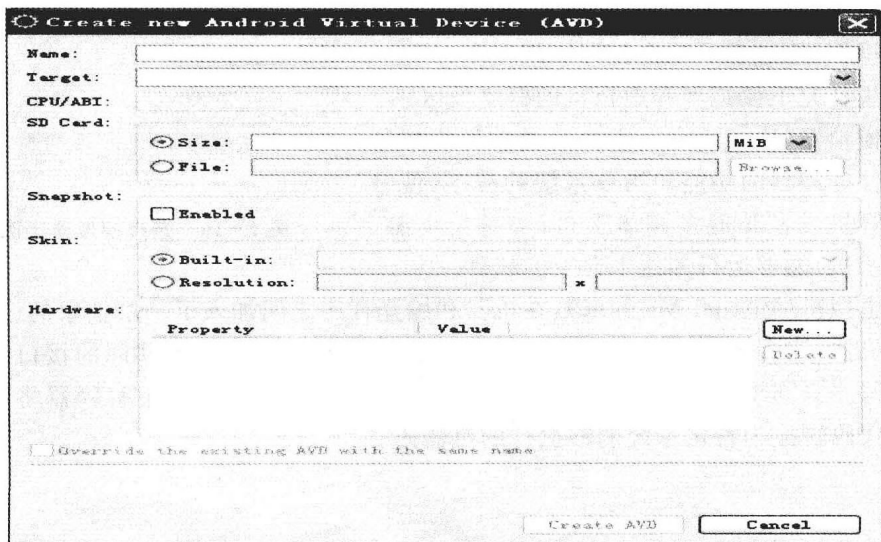


图 1-14 创建 AVD