



Android

经典项目案例开发 实战宝典

11小时高清多媒体教学视频

郭金尚 等编著



**资深Android程序员用心之作，多年开发经验毫无保留，以飨读者
立足实战，通过20个经典项目案例，全面、深入地阐释了Android开发的精髓**

- ✓ **注重实战：**详解20个Android经典项目案例的开发过程，提高实战开发水平
- ✓ **内容全面：**涵盖Android本地开发、网络开发、多媒体影音开发和游戏开发等领域
- ✓ **由浅入深：**从较为简单的案例开始，逐渐加大难度，适合各个层次的读者阅读
- ✓ **技巧丰富：**给出了大量的开发技巧，攻克各种疑点和难点，迅速提高开发水平
- ✓ **代码经典：**每个案例都给出了详细的源代码，并提供了大量的注释，便于读者研读
- ✓ **贴心专栏：**每个案例后都专门设有特色栏目“知识拓展”，以拓宽读者的知识面
- ✓ **视频教学：**每章都提供了高清多媒体教学视频，便于读者更加轻松、直观地学习

清华大学出版社

提供各种书籍的pd电子版代找服务，如果你找不到自己想要的书的pdf电子版，我们可以帮您找到，如有需要，请联系QQ1779903665.

PDF代找说明：

本人可以帮助你找到你要的PDF电子书，计算机类，文学，艺术，设计，医学，理学，经济，金融，等等。质量都很清晰，而且每本100%都带书签索引和目录，方便读者阅读观看，只要您提供给我书的相关信息，一般我都能找到，如果您有需求，请联系我QQ1779903665。

本人已经帮助了上万人找到了他们需要的PDF，其实网上有很多PDF,大家如果在网上不到的话，可以联系我QQ，大部分我都可以找到，而且每本100%带书签索引目录。因PDF电子书都有版权，请不要随意传播，如果您有经济购买能力，请尽量购买正版。

声明：本人只提供代找服务，每本100%索引书签和目录，因寻找pdf电子书有一定难度，仅收取代找费用。如因PDF产生的版权纠纷，与本人无关，我们仅仅只是帮助你寻找到你要的pdf而已。

Android

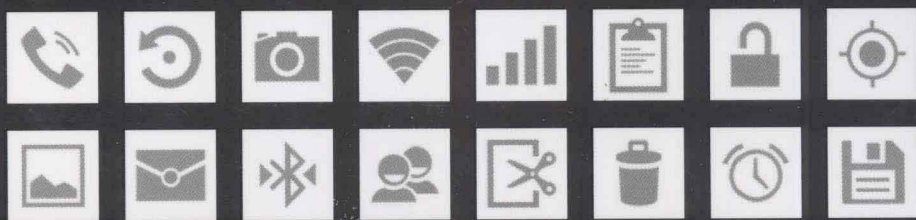
经典项目案例开发实战宝典

本书精华内容

- ✓ 搭建Android开发环境（18分钟视频）
- ✓ Android程序开发基础（31分钟视频）
- ✓ 计算器（38分钟视频）
- ✓ 电子词典（18分钟视频）
- ✓ 文件管理器（51分钟视频）
- ✓ 备忘录（37分钟视频）
- ✓ 短信收发工具（19分钟视频）
- ✓ 通讯录（29分钟视频）
- ✓ 任务管理器（29分钟视频）
- ✓ 软件管理器（19分钟视频）
- ✓ Android公交查询（44分钟视频）
- ✓ 股票查询软件（32分钟视频）
- ✓ Google天气客户端（16分钟视频）
- ✓ RSS新闻阅读器（35分钟视频）
- ✓ Android地图应用（14分钟视频）
- ✓ 新浪微博客户端（52分钟视频）
- ✓ MP3播放器（55分钟视频）
- ✓ Android照相机（13分钟视频）
- ✓ 视频播放器（29分钟视频）
- ✓ 小兔跳铃铛（35分钟视频）
- ✓ 飞行射击游戏（21分钟视频）
- ✓ 3D迷宫游戏（33分钟视频）

下载提示

本书提供配套教学PPT，需要的读者请到清华大学出版社网站（<http://www.tup.com.cn>）上搜索到本书页面后下载。



清华大学出版社数字出版网站

WQBook 书局

www.wqbook.com



DVD-ROM

ISBN 978-7-302-32101-9



9 787302 321019 >

定价：89.00元

附DVD光盘，含11小时教学视频
与20个案例源代码

Android

经典项目案例开发

实战宝典

郭金尚 等编著

清华大学出版社
北 京

内 容 简 介

本书详细介绍了 20 个 Android 项目案例的实际开发过程,以提升读者的实际项目开发水平。本书案例紧贴市场,讲解由浅入深,并注重对实际动手能力的提高,还配以翔实的开发情景截图,同时还将重要的知识点、开发技巧以“知识拓展”的形式呈现给读者。另外,作者专门为本书录制了大量的配套教学视频,以帮助读者更好地学习本书内容,这些视频和书中的实例源代码一起收录于本书的配书光盘中。

全书共分 5 篇。第 1 篇介绍 Android 开发环境及搭建、Android 工程的创建和调试方法;第 2 篇介绍计算器、电子词典、文件管理器、备忘录、短信收发工具、通讯录、任务管理器、软件管理器;第 3 篇介绍 Android 公交查询、股票查询软件、Google 天气客户端、RSS 新闻阅读器、Android 地图应用、新浪微博客户端;第 4 篇介绍 MP3 播放器、Android 照相机、视频播放器;第 5 篇介绍小兔跳铃铛、飞行射击游戏、3D 迷宫游戏。

本书涉及面广,从应用到游戏,从简单到复杂,几乎涉及 Android 项目开发的所有类型。本书适合所有想全面学习 Android 开发技术的人员阅读,也适合各种使用 Android 进行开发的工程技术人员使用。对于经常使用 Android 做开发的人员,本书更是一本不可多得的案头必备参考书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13501256678 13801310933

图书在版编目(CIP)数据

Android 经典项目案例开发实战宝典 / 郭金尚等编著. —北京:清华大学出版社, 2013
ISBN 978-7-302-32101-9

I. ①A… II. ①郭… III. ①移动电话机—应用软件 IV. ①TN929.53

中国版本图书馆 CIP 数据核字(2013)第 082727 号

责任编辑:夏兆彦

封面设计:欧振旭

责任校对:胡伟民

责任印制:刘海龙

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社总机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者:清华大学印刷厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:40.75 字 数:1020 千字

附光盘 1 张

版 次:2013 年 9 月第 1 版

印 次:2013 年 9 月第 1 次印刷

印 数:1~3000

定 价:89.00 元

产品编号:052481-01

前 言

从 2008 年 10 月第一部 Android 智能手机发布到 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, 短短三年时间 Android 可以说是称霸全球。2012 年 11 月数据显示, Android 占据全球智能手机操作系统市场 76% 的份额, 中国市场占有率为 90%, 如此庞大的用户群体, 相信没有一个开发者不会为之心动。

由于 Android 发展迅速, 导致了就业市场对 Android 开发人员的需求量猛增。然而, 很多企业需要的是拥有实践经验的开发人员。刚毕业的大学生一般没有企业要求的实践经验, 而培训机构的高昂培训费又令他们望而却步。尽管可以通过很多 Android 书籍中的小例子积累一些经验, 但这些例子毕竟有限, 有的也不完整, 根本达不到企业所要求的水平。

为了帮助读者提高 Android 实际应用开发水平, 笔者结合自己多年的开发经验和心得体会, 花费近一年的时间写作了本书。希望各位读者能在本书的引领下提高 Android 实际项目开发水平, 成为一名开发高手。

本书详细介绍了 20 个 Android 项目案例的开发过程, 这些案例紧贴市场, 实用价值高, 读者稍加修改便可用于自己的项目当中。为了便于读者高效而直观地阅读本书内容, 笔者还专门为本书录制了大量的配套教学视频。学习完本书后, 读者应该可以具备独立进行项目开发的能力。

本书特色

1. 配备大量多媒体语音教学视频, 学习效果好

作者专门录制了大量的配套多媒体语音教学视频, 以便让读者更加轻松、直观地学习本书内容, 提高学习效率。这些视频与本书源代码一起收录于配书光盘中。

2. 内容全面、系统、深入

本书系统介绍了 Android 开发中常见的几种项目类型, 包括本地应用开发、网络应用开发、多媒体影音开发及游戏开发, 提供了共计 20 个项目案例。这些项目案例可以让读者在实际操作中掌握开发流程, 巩固基础知识, 攻克难点疑点。

3. 讲解由浅入深, 循序渐进

本书从 Android 搭建环境讲起, 前面介绍较为简单的案例, 到后面逐渐加大难度, 让读者能循序渐进, 更好地接受本书的内容。

4. 贯穿大量的开发实例和技巧, 迅速提升开发水平

本书以大量的典型实例贯穿全文, 并给出了大量的开发技巧, 以便让读者更好地理解

各种概念和开发技术，体验实际编程，迅速提高开发水平。

5. 详解典型项目案例开发，提高实战水平

本书的每一个案例都给出了详细的开发过程，并配以大量而详细的代码及注释，让不同层次的读者均可以较好接受，对整个项目学习更全面。

6. 提供技术支持，答疑解惑

读者在阅读本书时有任何疑问，都可以发 E-mail 到 guojinshangb@163.com 或者 bookservice2008@163.com 以获得帮助。

本书内容及体系结构

第1篇 Android起步技术（第1~2章）

本篇主要内容包括：Android 开发环境的搭建、Android 基本应用程序的创建、Eclipse 的基本使用等。通过本篇的学习，读者可以掌握如何搭建 Android 开发环境和如何新建一个 Android 项目。

第2篇 Android典型应用实战案例（第3~10章）

本篇主要内容包括：计算器、电子词典、文件管理器、备忘录、短信收发工具、通讯录、任务管理器、软件管理等。通过本篇的学习，读者可以掌握 Android 应用编程的方法和思路。

第3篇 Android网络应用实战案例（第11~16章）

本篇主要内容包括：Android 公交查询、Android 股票软件、Google 天气客户端、RSS 新闻阅读器、Android 地图应用、新浪微博客户端等。通过本篇的学习，读者可以掌握 Android 网络编程技术。

第4篇 Android影音应用实战案例（第17~19章）

本篇主要内容包括：MP3 播放工具、Android 照相机、视频播放器等。通过本篇的学习，读者可以掌握 Android 多媒体应用的编程。

第5篇 Android游戏开发实战案例（第20~22章）

本篇主要内容包括：小兔跳铃铛、飞行射击游戏、3D 迷宫游戏等。通过本篇的学习，读者可以掌握 Android 游戏编程的知识，独立开发设计 Android 游戏。

本书读者对象

- ☐ Android 初学者；
- ☐ 想全面学习 Android 开发技术的人员；
- ☐ Android 专业开发人员；

- ☐ 利用 Android 做开发的工程技术人员；
- ☐ Android 开发爱好者；
- ☐ 大中专院校的学生；
- ☐ 社会培训班学员；
- ☐ 需要一案头必备手册的程序员。

本书作者

本书由郭金尚主笔编写。其他参与编写的人员有武冬、郅晓娜、孙美芹、卫丽行、尹翠翠、蔡继文、陈晓宇、迟剑、邓薇、郭利魁、金贞姬、李敬才、李萍、刘敬、陈慧、刘艳飞、吕博、全哲、余勇、宋学江、王浩、王康、王楠、杨宗芳、张严虎、周玉、张平、张靖波、周芳、杨罡、于海滨、张晶杰、张利峰、杨景凤、陈锴、郑剑锋、叶佩思。

虽然笔者对本书中所述内容都尽量核实，并多次进行文字校对，但因时间所限，可能还存在疏漏和不足之处，恳请读者批评指正。

编者著

目 录




第 1 篇 Android 起步技术

第 1 章 搭建 Android 开发环境 (🎥 教学视频: 18 分钟)	2
1.1 Android 的诞生	2
1.1.1 Android 的发展史	2
1.1.2 Android 的发行版本	3
1.2 Android 的系统架构及特性	5
1.2.1 Android 的系统架构	5
1.2.2 Android 的特性	7
1.3 Android 的开发环境的搭建	7
1.4 Android 调试	10
1.5 其他工具的使用	14
1.6 本章小结	15
第 2 章 Android 程序开发基础 (🎥 教学视频: 31 分钟)	16
2.1 Android 项目结构分析	16
2.2 Android 四大组件	19
2.2.1 Activity	19
2.2.2 Service	23
2.2.3 Broadcast	26
2.2.4 Content Provider	29
2.3 Activity 的生命周期	36
2.4 本章小结	42

第 2 篇 Android 典型应用实战案例

第 3 章 计算器 (🎥 教学视频: 38 分钟)	44
3.1 功能分析	44
3.2 界面设计	44
3.3 功能实现	49
3.3.1 定义变量	49
3.3.2 actionPerformed()函数	51
3.3.3 print()函数	54
3.3.4 TipChecker()函数	56

3.3.5 TipShow()函数	60
3.3.6 计算类 calc	62
3.4 知识拓展	70
3.5 本章小结	71
第4章 电子词典 (教学视频: 18 分钟)	72
4.1 功能分析	72
4.2 界面设计	74
4.3 功能实现	75
4.4 知识拓展	80
4.5 本章小结	81
第5章 文件管理器 (教学视频: 51 分钟)	82
5.1 功能分析	82
5.2 界面设计	83
5.3 功能实现	87
5.3.1 声明变量	87
5.3.2 初始化菜单及绑定监听器	88
5.3.3 设置长按监听器	90
5.3.4 显示指定目录内容	92
5.3.5 创建文件夹	93
5.3.6 重命名文件	95
5.3.7 删除文件	96
5.3.8 粘贴文件	97
5.3.9 搜索文件	99
5.3.10 接收器源文件	100
5.3.11 搜索服务	101
5.3.12 广播接收器	103
5.3.13 fileAdapter 代码	105
5.3.14 打开文件	108
5.3.15 系统默认打开文件的方法	110
5.3.16 用编辑器打开文本文件	111
5.3.17 文本编辑器的实现	112
5.3.18 网页浏览器	114
5.4 知识拓展	117
5.5 本章小结	120
第6章 备忘录 (教学视频: 37 分钟)	121
6.1 主界面设计	121
6.2 主界面功能	123
6.3 添加和更新备忘录页面	130
6.4 添加和更新备忘录功能实现	132
6.5 闹钟设置和实现	139
6.6 公共类的实现	144
6.7 知识拓展	145
6.8 本章小结	146

第 7 章 短信收发工具 ( 教学视频: 19 分钟)	147
7.1 显示手机所有信息	147
7.1.1 新建 main.xml	147
7.1.2 设置布局	148
7.1.3 新建文件 MsgListActivity.java	149
7.2 新建信息	152
7.2.1 界面设计	152
7.2.2 实现发送信息功能	153
7.2.3 发送信息	156
7.3 回复信息	157
7.4 知识拓展	161
7.5 本章小结	161
第 8 章 通讯录 ( 教学视频: 29 分钟)	162
8.1 界面设计	162
8.1.1 布局的设置	163
8.1.2 添加“查看联系人”页面	165
8.2 功能实现	167
8.2.1 创建数据库	167
8.2.2 创建 ContactColumn 类	168
8.2.3 为数据库提供操作类	169
8.2.4 ListView 界面的实现	173
8.2.5 创建菜单	173
8.2.6 实现界面的查看	175
8.2.7 添加一个标志变量	178
8.2.8 设置 menu 菜单	181
8.3 知识拓展	183
8.4 本章小结	185
第 9 章 任务管理器 ( 教学视频: 29 分钟)	186
9.1 功能分析	186
9.2 界面设计	187
9.2.1 编写主界面	187
9.2.2 ListView 布局的设置	188
9.2.3 显示程序的详细信息	189
9.3 功能实现	192
9.3.1 初始化变量	192
9.3.2 获取运行的进程	193
9.3.3 获取应用程序	196
9.3.4 存放程序的基本信息	197
9.3.5 取出信息并适配到 ListView 中	198
9.3.6 重写 onItemClick 方法	199
9.3.7 关闭指定进程	201
9.3.8 显示文件详细信息	201
9.3.9 显示程序详细信息	205
9.3.10 更新列表	209
9.3.11 查看程序详细信息	210

9.4 知识拓展	210
9.5 本章小结	214
第 10 章 软件管理器 (教学视频: 19 分钟)	215
10.1 功能分析	215
10.2 界面设计	216
10.2.1 主界面的设置	216
10.2.2 设置 ListView 布局	218
10.2.3 设置 GridView 的子元素布局	219
10.3 功能实现	220
10.3.1 声明变量	221
10.3.2 ListViewAdapter 和 GridViewAdapter	221
10.3.3 实现 getView() 函数	222
10.3.4 入口函数 onCreate()	223
10.3.5 线程 thread	225
10.3.6 AlertDialog	226
10.4 知识拓展	229
10.5 本章小结	232

第 3 篇 Android 网络应用实战案例




第 11 章 Android 公交查询 (教学视频: 44 分钟)	234
11.1 功能分析	234
11.2 界面设计	235
11.3 功能设计	237
11.3.1 数据文件生成和校验	237
11.3.2 显示城市列表	242
11.3.3 公交查询	246
11.4 知识扩展	267
11.5 本章小结	269
第 12 章 股票查询软件 (教学视频: 32 分钟)	270
12.1 功能分析	270
12.2 界面布局	271
12.2.1 主界面的设置	272
12.2.2 设置 ListView 布局	273
12.2.3 设置界面布局	274
12.3 功能实现	278
12.3.1 新建一个类 StockInfo	278
12.3.2 初始化主界面	279
12.3.3 适配器类 QuoteAdapter	281
12.3.4 设置按键监听器	284
12.3.5 数据查询	287
12.3.6 读取股票代码信息	288
12.3.7 股票的更新	291
12.4 知识拓展	293



12.5 本章小结	294
第 13 章 Google 天气客户端 (📺 教学视频: 16 分钟)	295
13.1 功能分析	295
13.2 XML 解析	296
13.2.1 DOM 解析	296
13.2.2 SAX 解析	299
13.2.3 PULL 解析	303
13.3 界面设计	306
13.4 功能实现	311
13.4.1 设置当前天气类	312
13.4.2 设置天气预报类	314
13.4.3 天气预报信息汇总	315
13.4.4 设置主界面	316
13.4.5 ConstData.java 类	320
13.4.6 程序的核心函数	321
13.4.7 存储天气信息	324
13.5 知识拓展	328
13.6 本章小结	332
第 14 章 RSS 新闻阅读器 (📺 教学视频: 35 分钟)	333
14.1 功能分析	333
14.2 登录过程实现	334
14.2.1 界面设置	335
14.2.2 新建 LoginActivity.java	335
14.3 RSS 源的设置	337
14.3.1 RSS 源选择界面设计	337
14.3.2 创建数据库	340
14.3.3 显示每行元素的界面	343
14.3.4 添加 RSS 源界面	343
14.3.5 实现添加 RSS 源界面的功能	345
14.4 读取 RSS 源	348
14.4.1 存放 RSS 信息	348
14.4.2 取出需要的信息	350
14.4.3 对 XML 文件进行解析	351
14.4.4 调用、共享 RSS 信息	353
14.4.5 查看界面	356
14.4.6 详细查看 RSS 信息	360
14.5 知识拓展	361
14.6 本章小结	362
第 15 章 Android 地图应用 (📺 教学视频: 14 分钟)	363
15.1 开发前准备	363
15.2 创建地图应用	364
15.2.1 新建布局文件	364
15.2.2 新建程序管理类	365
15.2.3 地图的主界面	366

15.3 知识拓展	370
15.4 本章小结	370
第 16 章 新浪微博客户端 (教学视频: 52 分钟)	371
16.1 开发前的准备	371
16.1.1 申请微博账号和获得授权	371
16.1.2 OAuth 认证介绍	372
16.1.3 SDK 使用说明	373
16.2 载入界面设计	374
16.3 载入界面功能实现	374
16.3.1 保存用户信息	375
16.3.2 新建数据库	376
16.3.3 “增删改查”数据	377
16.3.4 获取数据库所有的信息	380
16.4 授权功能实现	382
16.5 登录界面设计	386
16.5.1 设置布局结构	387
16.5.2 显示所有用户列表	388
16.6 登录界面功能实现	389
16.6.1 初始化界面图标	390
16.6.2 用户的授权	390
16.6.3 按键监听器的设置	392
16.6.4 用户适配器	392
16.6.5 对话框监听器	393
16.6.6 底部菜单的实现	393
16.6.7 “登录”按钮	395
16.7 用户首页设计	397
16.8 用户首页功能实现	401
16.9 阅读微博界面设计	412
16.10 阅读微博功能实现	416
16.11 知识拓展	422
16.12 本章小结	424

第 4 篇 Android 影音应用实战案例

第 17 章 MP3 播放器 (教学视频: 55 分钟)	426
17.1 主界面设计	426
17.1.1 主界面概览	426
17.1.2 中间切换界面实现	427
17.1.3 底部切换界面实现	432
17.1.4 主界面结构布局	435
17.2 左右界面设计	440
17.2.1 歌曲列表界面布局	440
17.2.2 专辑列表界面布局	442
17.3 中间滑动界面	444

17.3.1	左侧视图——播放动画	444
17.3.2	中间视图——显示专辑	446
17.3.3	右侧视图——显示歌词	447
17.4	主界面功能实现	452
17.4.1	音乐播放界面	452
17.4.2	MusicListView 类	453
17.4.3	新建类 MusicSpecialView	454
17.4.4	对界面初始化	455
17.4.5	图片的设置	461
17.4.6	布局文件 dialog.xml	464
17.5	歌曲信息类	466
17.6	音乐播放服务	470
17.7	知识拓展	478
17.8	本章小结	479
第 18 章	Android 照相机 ( 教学视频: 13 分钟)	480
18.1	调用 Android 相机的两种方式	480
18.1.1	调用系统自带相机	480
18.1.2	根据 Camera API 实现自己的拍照程序	481
18.2	相机界面设计	483
18.3	相机功能实现	484
18.3.1	拍照功能实现	485
18.3.2	照片查看	488
18.4	知识拓展	491
18.5	本章小结	492
第 19 章	视频播放器 ( 教学视频: 29 分钟)	493
19.1	视频播放界面设计	493
19.2	播放器主界面	494
19.3	播放器功能实现	503
19.4	知识拓展	518
19.5	本章小结	519
 第 5 篇 Android 游戏开发实战案例 		
第 20 章	小兔跳铃铛 ( 教学视频: 35 分钟)	522
20.1	功能分析	522
20.2	游戏角色设计	522
20.2.1	小兔类	523
20.2.2	铃铛类	529
20.2.3	小鸟类	532
20.3	游戏背景设计	535
20.3.1	背景音乐	535
20.3.2	背景图片	536

20.4	游戏辅助界面	538
20.4.1	开场画面	538
20.4.2	帮助界面	539
20.4.3	声音设置界面	539
20.4.4	结束界面	541
20.5	游戏过程	543
20.6	知识拓展	559
20.7	本章小结	559
第 21 章	飞行射击游戏 ( 教学视频: 21 分钟)	560
21.1	功能分析	560
21.2	子弹和敌机类的实现	561
21.3	场景的绘制	566
21.4	知识拓展	574
21.5	本章小结	576
第 22 章	3D 迷宫游戏 ( 教学视频: 33 分钟)	577
22.1	游戏地图绘制方法	577
22.2	游戏地图的绘制	578
22.2.1	3D 绘图基本知识	578
22.2.2	地板	580
22.2.3	墙壁	582
22.2.4	小球	591
22.2.5	圆形洞	595
22.2.6	数字	597
22.3	游戏菜单	599
22.3.1	界面布局	600
22.3.2	主菜单功能	606
22.4	游戏进行	620
22.5	知识拓展	637
22.6	本章小结	638

第 1 篇 *Android* 起步技术

▶▶ 第 1 章 搭建 Android 开发环境

▶▶ 第 2 章 Android 程序开发基础

第 1 章 搭建 Android 开发环境

Android 是基于 Linux 内核的软件平台和操作系统，是 Google 在 2007 年 11 月 5 日公布的手机系统平台，早期由 Google 开发，后由开放手持设备联盟（Open Handset Alliance）开发。

1.1 Android 的诞生

Android 操作系统最初由 Andy Rubin 开发，刚开始主要支持手机，被 Google 收购后，对 Android 进行了改良，使其可以使用于平板电脑等其他领域。

1.1.1 Android 的发展史

Android 的发展历史如下：

- ❑ 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 低调收购了成立仅 22 个月的高科技企业 Android 及其团队。Andy Rubin 成为 Google 公司工程部副总裁，继续负责 Android 项目。
- ❑ 2007 年 11 月 5 日，Google 公司正式向外界展示了这款名为 Android 的操作系统，并且在这天 Google 宣布建立一个全球性的联盟组织，该组织由 34 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，并与 84 家硬件制造商、软件开发商及电信运营商组成开放手持设备联盟（Open Handset Alliance）来共同研发改良 Android 系统，这一联盟将支持 Google 发布的手机操作系统以及应用软件，Google 以 Apache 免费开源许可证的授权方式，发布了 Android 的源代码。
- ❑ 2008 年，在 Google I/O 大会上，Google 提出了 Android HAL 架构图，在同年 8 月 18 日，Android 获得了美国联邦通信委员会（FCC）的批准，在 2008 年 9 月，Google 正式发布了 Android 1.0 系统，这也是 Android 系统最早的版本。
- ❑ 2009 年 4 月，Google 正式推出了 Android 1.5 这款手机。从 Android 1.5 版本开始，Google 开始将 Android 的版本以甜品的名字命名，Android 1.5 命名为 Cupcake（纸杯蛋糕）。该系统与 Android 1.0 相比有了很大的改进。
- ❑ 2009 年 9 月，Google 发布了 Android 1.6 正式版，并且推出了搭载 Android 1.6 正式版的手机 HTC Hero（G3）。凭借着出色的外观设计以及全新的 Android 1.6 操作系统，HTC Hero（G3）成为当时全球最受欢迎的手机。Android 1.6 也有一个有趣的甜品名称，它被称为 Donut（甜甜圈）。
- ❑ 2010 年 2 月，Linux 内核开发者 Greg Kroah-Hartman 将 Android 的驱动程序从 Linux

内核“状态树”（staging tree）上除去，从此，Android 与 Linux 开发主流分道扬镳。在同年 5 月，Google 正式发布了 Android 2.2 操作系统。Google 将 Android 2.2 操作系统命名为 Froyo，翻译成中文名为冻酸奶。

- ❑ 2010 年 10 月，Google 宣布 Android 系统达到了第一个里程碑，即电子市场上获得官方数字认证的 Android 应用数量已经达到了 10 万个。Android 系统的应用增长非常迅速，在 2010 年 12 月，Google 正式发布了 Android 2.3 操作系统 Gingerbread（姜饼）。
- ❑ 2011 年 1 月，Google 称每日的 Android 设备新用户数量达到了 30 万部，到 2011 年 7 月，这个数字增长到 55 万部，而 Android 系统设备的用户总数达到了 1.35 亿，Android 系统已经成为智能手机领域占有量最高的系统。
- ❑ 2011 年 8 月 2 日，Android 手机已占据全球智能手机市场 48% 的份额，并在亚太地区市场占据统治地位，终结了 Symbian（塞班系统）的霸主地位，跃居全球第一。
- ❑ 2011 年 9 月，Android 系统的应用数目已经达到了 48 万，而在智能手机市场，Android 系统的占有率则达到了 43%，继续排在移动操作系统首位。在 9 月 19 日，Google 发布全新的 Android 4.0 操作系统，这款系统被 Google 命名为 Ice Cream Sandwich（冰激凌三明治）。
- ❑ 2012 年 1 月 6 日，Google Android Market 已有 10 万开发者推出超过 40 万活跃的应用，大多数的应用程序为免费。Android Market 应用程序商店目录在新年首周末突破 40 万基准，距离突破 30 万应用仅 4 个月。在 2011 年早些时候，Android Market 从 20 万增加到 30 万应用也花了 4 个月。

1.1.2 Android 的发行版本

Android 在正式发行之之前，最开始拥有两个内部测试版本，并且以著名的机器人名称来对其进行命名，它们分别是：阿童木（Android Beta）、发条机器人（Android 1.0）。后来由于涉及到版权问题，Google 将其命名规则变更为用甜点作为它们系统版本的代号，如图 1.1 所示。甜点命名法开始于 Android 1.5 发布的时候。作为每个版本代表的甜点的尺寸越变越大，然后按照 26 个字母顺序：纸杯蛋糕（Android 1.5），甜甜圈（Android 1.6），松饼（Android 2.0/2.1），冻酸奶（Android 2.2），姜饼（Android 2.3），蜂巢（Android 3.0），冰激凌三明治（Android 4.0），根据最新消息，新一代 Android 版本将命名为果冻豆（Jelly Bean）。

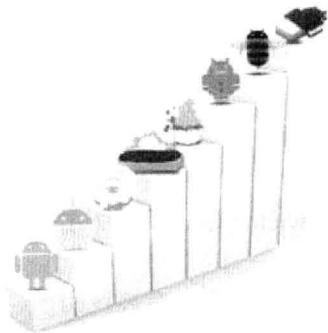


图 1.1 Android 发行版本

Android 各个发行版本及更新如下:

❑ Android 1.1: 2008 年 9 月发布的 Android 第一版。

❑ Android 1.5 Cupcake (纸杯蛋糕): 2009 年 4 月 30 日发布。主要的更新如下:

拍摄/播放影片, 并支持上传到 Youtube; 支持立体声蓝牙耳机, 同时改善自动配对性能; 最新的采用 WebKit 技术的浏览器, 支持复制/粘贴和页面中搜索; GPS 性能大大提高; 提供屏幕虚拟键盘; 主屏幕增加音乐播放器和相框 widgets; 应用程序自动随着手机旋转; 短信、Gmail、日历、浏览器的用户接口大幅改进, 如 Gmail 可以批量删除邮件; 相机启动速度加快, 拍摄图片可以直接上传到 Picasa; 来电照片显示。

❑ Android 1.6 Donut (甜甜圈): 2009 年 9 月 15 日发布。主要的更新如下:

重新设计的 Android Market 手势; 支持 CDMA 网络; 文字转语音系统(Text-to-Speech); 快速搜索框; 全新的拍照接口; 查看应用程序耗电; 支持虚拟私人网络(VPN); 支持更多的屏幕分辨率; 支持 OpenCore 2 媒体引擎; 新增面向视觉或听觉困难人群的易用性插件。

❑ Android 2.0/2.0.1/2.1 Eclair (松饼): 2009 年 10 月 26 日发布。主要的更新如下:

优化硬件速度; Car Home 程序; 支持更多的屏幕分辨率; 改良的用户界面; 新的浏览器的用户接口和支持 HTML 5; 新的联系人名单; 更好的白色/黑色背景比率; 改进 Google Maps 3.1.2; 支持 Microsoft Exchange; 支持内置相机闪光灯; 支持数码变焦; 改进的虚拟键盘; 支持蓝牙 2.1; 支持动态桌面的设计。

❑ Android 2.2/2.2.1 Froyo (冻酸奶): 2010 年 5 月 20 日发布。主要的更新如下:

整体性能大幅度提升; 3G 网络共享功能; Flash 的支持; App2sd 功能; 全新的软件商店; 更多的 Web 应用 API 接口的开发。

❑ Android 2.3.x Gingerbread (姜饼): 2010 年 12 月 7 日发布。主要的更新如下:

增加了新的垃圾回收和优化处理事件; 原生代码可直接存取输入和感应器事件、EGL/OpenGL ES、OpenSL ES; 新的管理窗口和生命周期的框架; 支持 VP8 和 WebM 视频格式, 提供 AAC 和 AMR 宽频编码, 提供了新的音频效果器; 支持前置摄像头、SIP/VOIP 和 NFC (近场通讯); 简化界面、速度提升; 更快更直观的文字输入; 一键文字选择和复制/粘贴; 改进的电源管理系统; 新的应用管理方式。

❑ Android 3.0 Honeycomb (蜂巢): 2011 年 2 月 2 日发布。主要更新如下:

针对平板优化; 全新设计的 UI 增强网页浏览功能; n-app purchases 功能。

❑ Android 3.1 Honeycomb (蜂巢): 2011 年 5 月 11 日发布。版本主要更新如下:

经过优化的 Gmail 电子邮箱; 全面支持 Google Maps; 将 Android 手机系统与平板系统再次合并从而方便开发者; 任务管理器可滚动, 支持 USB 输入设备(键盘、鼠标等); 支持 Google TV, 可以支持 XBOX 360 无线手柄; widget 支持的变化, 能更加容易地定制屏幕 widget 插件。

❑ Android 3.2 Honeycomb (蜂巢): 2011 年 7 月 13 日发布。版本更新如下:

支持 7 英寸设备; 引入了应用显示缩放功能。

❑ Android 4.0 Ice Cream Sandwich (冰激凌三明治): 2011 年 10 月 19 日发布。版本主要更新如下:

全新的 UI; 全新的 Chrome Lite 浏览器, 有离线阅读、16 标签页、隐身浏览模式等; 截图功能; 更强大的图片编辑功能; 自带照片应用堪比 Instagram, 可以加滤镜、加相框,

进行 360° 全景拍摄，照片还能根据地点来排序；Gmail 加入手势、离线搜索功能，UI 更强大；新功能 People：以联系人照片为核心，界面偏重滑动而非点击，集成了 Twitter、Linkedin、Google+ 等通讯工具，有望支持用户自定义添加第三方服务；新增流量管理工具，可具体查看每个应用产生的流量，限制使用流量，到达设置标准后自动断开网络。

❑ Android 4.1 Jelly Bean（果冻豆）。新特性如下：

更快、更流畅、更灵敏；特效动画的帧速提高至 60fps，增加了三倍缓冲；增强通知栏；全新搜索，搜索带来全新的 UI、智能语音搜索和 Google Now 三项新功能；桌面插件自动调整大小；加强无障碍操作；语言和输入法扩展；新的输入类型和功能；新的连接类型。

1.2 Android 的系统架构及特性

1.2.1 Android 的系统架构

如图 1.2 可以看出，Android 操作系统的体系结构可分为 4 层，由上到下依次是应用程序、应用程序框架、核心类库和 Linux 内核，其中第三层还包括 Android 运行时的环境。下面分别来讲解各个部分。

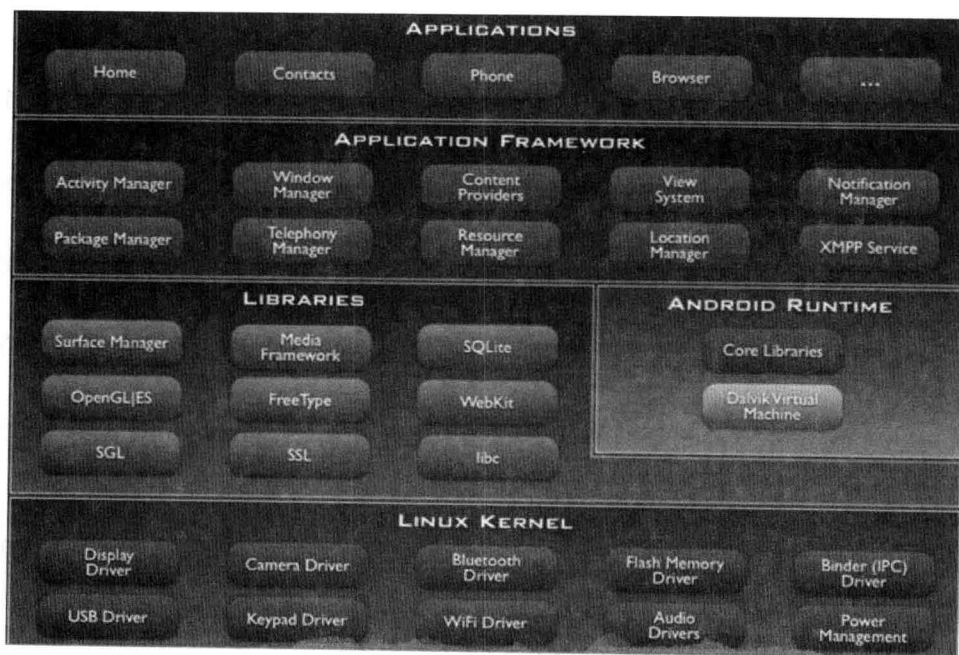


图 1.2 Android 系统架构

1. 应用程序

Android 连同一个核心应用程序包一起发布，该应用程序包包括 E-mail 客户端、SMS 短消息程序、日历、地图、浏览器、联系人管理程序等。所有的应用程序都是用 Java 编写的。

2. 应用程序框架

开发者完全可以访问核心应用程序所使用的 API 框架。该应用程序框架架构用来简化组件软件的重用，任何一个应用程序都可以发布它的功能块，并且任何其他的应用程序都可以使用其发布的功能块（不过得遵循框架的安全性限制）。该应用程序重用机制使得组件可以被用户替换。

所有的应用程序都由一系列的服务和系统组成，包括：

(1) 一个可扩展的视图 (Views)，可以用来创建应用程序，包括列表 (lists)、网络 (grids)、文本框 (text boxes)、按钮 (buttons)，甚至是一个可嵌入的 Web 浏览器。

(2) 内容管理器 (Content Providers)，使得应用程序可以访问另一个应用程序的数据（如联系人数据库），或者共享它们自己的数据。

(3) 一个资源管理器 (Resource Manager)，提供非代码资源的访问，如本地字符串、图形和分层文件 (layout files)。

(4) 一个通知管理器 (Notification Manager)，使得应用程序可以在状态栏中显示客户通知信息。

(5) 一个活动类管理器 (Activity Manager)，用来管理应用程序生命周期并提供常用的导航回退功能。

3. Android程序库

Android 包括一个被 Android 系统中各种不同组件所使用的 C/C++ 集库，该库通过 Android 应用程序框架为开发者提供服务。以下是一些主要的核心库：

(1) 系统 C 库：一个从 BSD 继承来的标准 C 系统函数库 (libc)，专门为基于 Embedded Linux 的设备定制。

(2) 媒体库：基于 PacketVideo OpenCORE。该库支持录放，并且可以录制许多流行的音频视频格式以及静态映像文件，包括 MPEG4、H.264、MP3、AAC、JPG、PNG。

(3) Surface Manager：对显示子系统的管理，并且为多个应用程序提供 2D 和 3D 图层的无缝融合。

(4) LibWebCore：一个最新的 Web 浏览器引擎，用来支持 Android 浏览器和一个可嵌入的 Web 视图。

(5) SGL：一个内置的 2D 图形引擎。

(6) 3D libraries：基于 OpenGL ES 1.0 API 实现。该库可以使用硬件 3D 加速（如果可用）或者使用高度优化的 3D 软加速。

(7) FreeType：位图 (bitmap) 和向量 (vector) 字体显示。

(8) SQLite：一个对于所有应用程序可用、功能强劲的轻型关系型数据库引擎。

4. Android运行库

Android 包括了一个核心库，该核心库提供了 Java 编程语言核心库的大多数功能。

每一个 Android 应用程序都在它自己的进程中运行，都拥有一个独立的 Dalvik 虚拟机实例。Dalvik 是针对同时高效地运行多个 VM 实现的。Dalvik 虚拟机执行 .dex 的 Dalvik 可执行文件，该格式文件针对最小内存使用做了优化。该虚拟机是基于寄存器的，所有的类

都经由 Java 编译器编译, 然后通过 SDK 中的 DX 工具转化成.dex 格式由虚拟机执行。

Dalvik 虚拟机依赖于 Linux 的一些功能, 如线程机制和底层内存管理机制。

5. Linux内核

Android 的核心系统服务依赖于 Linux 内核, 如安全性、内存管理、进程管理、网络协议栈和驱动模型。Linux 内核也同时作为硬件和软件栈之间的硬件抽象层。

1.2.2 Android 的特性

Android 平台有如下特性:

(1) 应用程序框架支持组件的重用与替换。

这样我们可以把系统中不喜欢的应用程序删除, 安装我们喜欢的应用程序。

(2) Dalvik 虚拟机专门为移动设备进行了优化。

Android 应用程序将由 Java 编写、编译的类文件, 通过 DX 工具转换成一种后缀名为.dex 的文件来执行。Dalvik 虚拟机是基于寄存器的, 相对于 Java 虚拟机速度要快很多。

(3) 内部集成浏览器基于开源的 WebKit 引擎。

有了内置的浏览器, 这将意味着 WAP 应用的时代即将结束, 真正的移动互联网时代已经来临, 手机就是一台“小电脑”, 可以在网上随意遨游。

(4) 优化的图形库包括 2D 和 3D 图形库, 3D 图形库基于 OpenGL ES 1.0。

强大的图形库给游戏开发带来了福音。在 3G 时代最为重要的应用莫过于手机上网和手机游戏。

(5) SQLite 用作结构化的数据存储。

(6) 多媒体支持包括常见的音频、视频和静态映像文件格式, 如 MPEG4、H.264、MP3、AAC、AMR、JPG、PNG、GIF。

(7) GSM 电话(依赖于硬件)功能。

(8) 蓝牙(Bluetooth)、EDGE、3G、WiFi(依赖于硬件)功能。

(9) 照相机、GPS、指南针和加速度计(依赖于硬件)功能。

(10) 丰富的开发环境, 包括设备模拟器、调试工具、内存及性能分析图表和 Eclipse 集成的开发环境插件。

(11) Google 提供了 Android 开发包 SDK, 其中包含了大量的类库和开发工具, 以及针对 Eclipse 的可视化开发插件 ADT。

1.3 Android 的开发环境的搭建

在开始 Android 开发之前, 首先要做的就是搭建开发环境, 主要包括 JDK 的安装、Eclipse 的安装、ADT 的安装、Android SDK 的安装及创建 AVD。

1. java JDK安装

进入网页 <http://java.sun.com/javase/downloads/index.jsp>, 看到如图 1.3 所示的 JDK 下载界面, 选择最新的 JDK7。

Java Platform, Standard Edition		
Java SE 7u6 JavaFX 2.2 is now bundled with the JDK on Windows, Mac and Linux x86/x64. We also include bug fixes and enhancements. Learn more ▶	JDK	JRE
	DOWNLOAD ▶	DOWNLOAD ▶
"What Java Do I Need?" You must have a copy of the JRE (Java Runtime Environment) on your system to run Java applications and applets. To develop Java applications and applets, you need the JDK (Java Development Kit), which includes the JRE.	JDK 7 Docs	JRE 7 Docs
	Installation Instructions ReadMe ReleaseNotes Oracle License Java SE Products Third Party Licenses Certified System Configurations	Installation Instructions ReadMe ReleaseNotes Oracle License Java SE Products Third Party Licenses Certified System Configurations
JDK 7 and JavaFX Demos and Samples Demos and samples of common tasks and new functionality available on JDK 7. The source code provided with samples and demos for the JDK is meant to illustrate the usage of a given feature or technique and has been deliberately simplified.	Demos and Samples DOWNLOAD ▶	
Java SE 6 Update 34 This release includes tools useful for developing and testing programs written in the Java programming language and running on the JavaTM platform. Learn more ▶	JDK	JRE
	DOWNLOAD ▶	DOWNLOAD ▶
	JDK 6 Docs	JRE 6 Docs

图 1.3 JDK 下载

下载完之后直接安装即可。我们检查一下是否已经正确安装。进入 cmd 窗口，输入 `java-version`，看到如图 1.4 所示的显示，表明 JDK 已经安装正确。

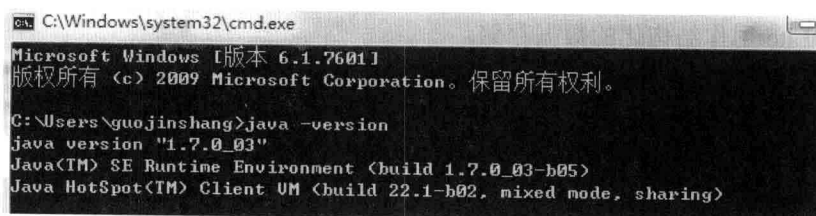


图 1.4 查看系统 Java 版本

2. Eclipse 安装

进入网页 <http://www.eclipse.org/downloads/>，看到 Eclipse 下载界面，如图 1.5 所示。选择第一个 Java EE 下载。



图 1.5 Eclipse 下载

3. ADT 安装

打开 Eclipse IDE，单击 Help|Install New Software 命令，单击 Add...按钮，弹出对话框要求输入 Name 和 Location。Name 自己随便取，Location 输入 <http://dl-ssl.google.com/android/eclipse/>。确定返回后，在 work with 后的下拉列表中选择我们刚才添加的 ADT，我们会看到下面有 Developer Tools，展开它会有 Android DDMS 和 Android Development Tools，勾选它们，如图 1.6 所示。然后按提示一步一步单击 next 按钮，重启 Eclipse 之后可以看到 Eclipse 工具栏多了 Android SDK Manager 和 Android Virtual Device Manager 两个按钮。

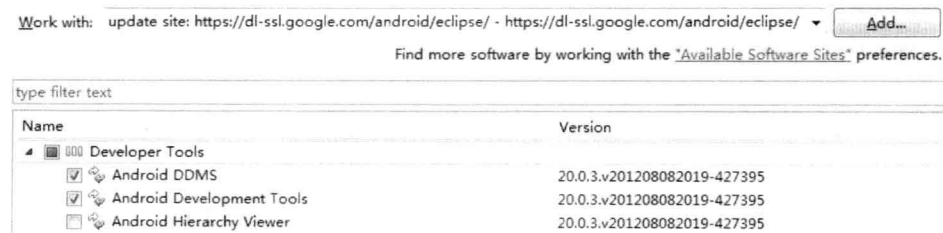


图 1.6 ADT 下载

4. SDK 下载

打开 Android SDK Manager，显示当前可用的 SDK 版本，选择你需要的 SDK 版本进行下载，这里我们勾选上 2.3.3、2.3.1 和 2.2 一并下载下来，如图 1.7 所示。当然实际上本书我们只用到 2.3.3 版本，多下载也没什么影响，关键时刻想用一下就方便了。

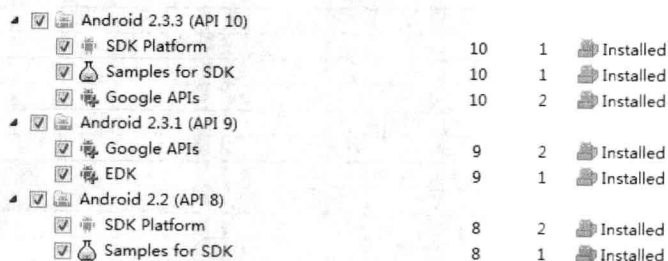


图 1.7 SDK 下载

下载完之后，单击 Windows|Preferences 命令，选择 Android，在右侧的 SDK Location 中选择 SDK 安装的位置，单击 OK 按钮即可将 SDK 导入。

5. 创建AVD

打开 Android Virtual Device Manager，Name 可以随便设置，Target 选择 API Level 10，SD Card 选择 100M，Skin 选择默认的就行，如图 1.8 所示，最后单击 Create AVD 即可完成创建 AVD。

6. 新建一个HelloAndroid程序

单击 File|New |Android Project 命令，建立新项目 HelloAndroid，如图 1.9 所示。



图 1.8 创建 AVD

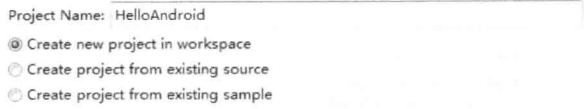


图 1.9 填写项目名称

然后单击 Next 按钮，在 Build Target 中选择“Android 2.3.3”，单击 Next 按钮。填写 Package Name，其他的默认就行，如图 1.10 所示。单击 Finish 按钮。

最后右键单击刚才生成的项目，单击 Run As|Android Application 命令，出现如图 1.11 所示的界面，表明 AVD 启动成功。至此，我们的 Android 开发环境就搭建好了。



图 1.10 填写包名

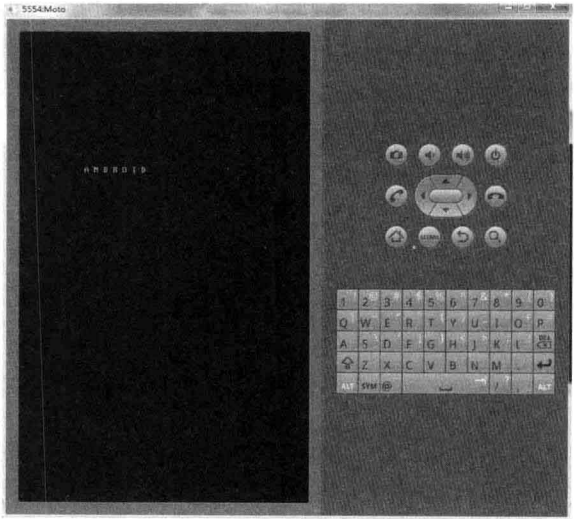


图 1.11 AVD 启动

1.4 Android 调试

Eclipse 为我们提供了方便的调试工具——DDMS，DDMS 的全称是 Dalvik Debug Monitor Service，它为我们提供了多种功能，例如，为测试设备截屏、针对特定的进程查

看正在运行的线程以及堆信息、Logcat、广播状态信息、模拟电话呼叫、接收 SMS、虚拟地理坐标等等。

DDMS 工具存放在 {ANDROID-SDK}/tools 路径下，启动 DDMS 的方法如下：

- ❑ 直接在 tools 目录下双击 ddms.bat。
 - ❑ 在 Eclipse 调试程序的过程中启动 DDMS，相应选项在 Eclipse 界面中的位置如图 1.12 所示。
- 启动 DDMS 后，界面如图 1.13 所示。

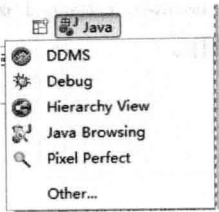


图 1.12 DDMS 工具位置

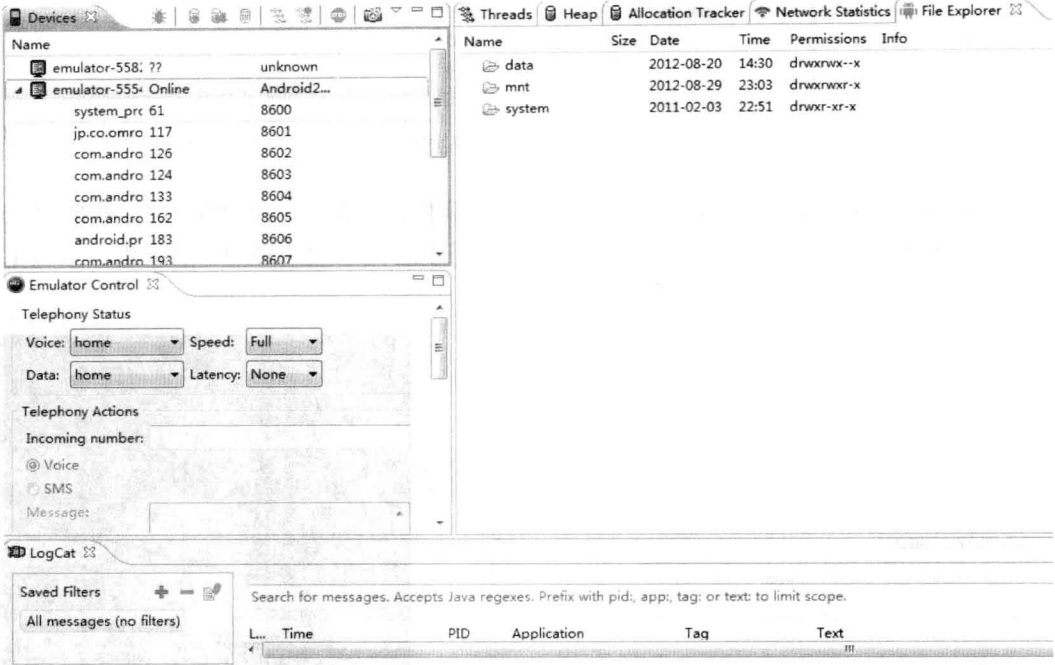


图 1.13 DDMS 界面

DDMS 将搭建起 IDE 与测试终端（Emulator 或者 connected device）的链接，它们应用各自独立的端口监听调试器的信息，DDMS 可以实时监测到测试终端的连接情况。当有新的测试终端连接后，DDMS 将捕捉到终端的 ID，并通过 ADB 建立调试器，从而实现发送指令到测试终端的目的。

如图 1.14 所示，DDMS 可以监听虚拟机中的进程。DDMS 监听的第一个终端 APP 进程的端口为 8600，第二个 APP 进程将分配 8601，如果有更多终端或者更多 APP 进程，将按照这个顺序依次类推。DDMS 通过 8700 端口（base port）接收所有终端的指令。在面板的右上角有一排很重要的按键，它们分别是 Debug the selected process、Update Threads、Update Heap、Stop Process 和 ScreenShot。

在开发过程中，有时需要用到短信、电话功能等，Eclipse 也为我们提供了这些功能的调试工具。通过这个面板的一些功能可以非常容易地使测试终端模拟真实手机所具备的一些交互功能，比如接听电话、根据选项模拟各种不同网络情况、模拟接受 SMS 消息和发送

虚拟地址坐标用于测试 GPS 功能等。下面我们分别来看看模拟打电话和发送短信功能如何操作。

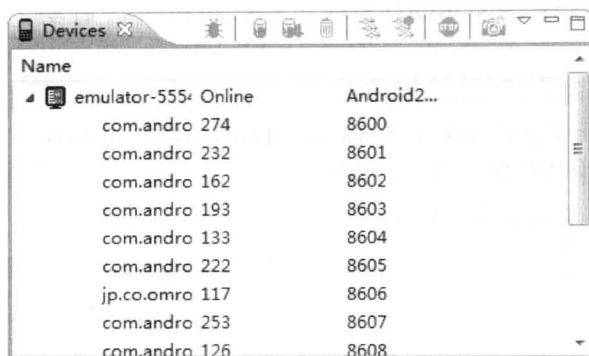


图 1.14 监听 APP 端口

选择要呼叫的模拟器，然后输入呼叫的号码，我们这里输入“12345678”，选择 Voice，单击 Call 按钮，如图 1.15 所示。在模拟器一端就可以看到“12345678”的来电，如图 1.16 所示。

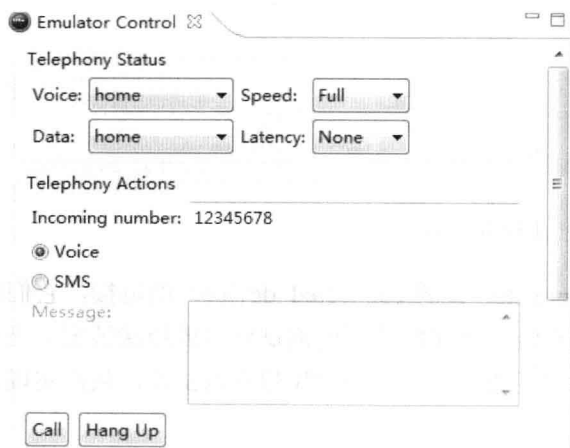


图 1.15 呼叫模拟器



图 1.16 收到来电

发送短信时方法类似，如图 1.17 所示，选择 SMS，输入要发送的信息，单击 Send 按钮。这时在模拟器端可以接收到来自“12345678”的短信“Hello Android!”，如图 1.18 所示。

调试的时候使用最多的要数 logcat 了，logcat 是 Android 的一个命令行工具，可以得到程序的 log 信息。logcat 的功能是由 Android 的类 android.util.Log 决定的，在程序中 log 的使用方法如下所示：

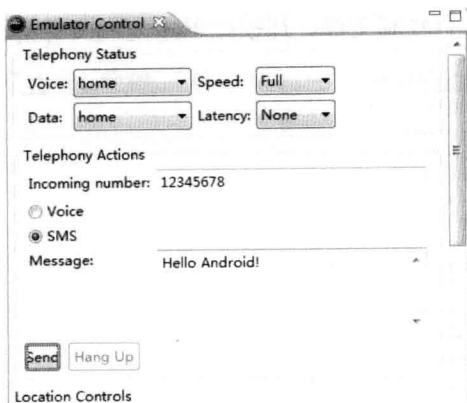


图 1.17 发送短信



图 1.18 查看短信

```

Log.v() ----- VERBOSE
Log.d() ----- DEBUG
Log.i() ----- INFO
Log.w() ----- WARN
Log.e() ----- ERROR

```

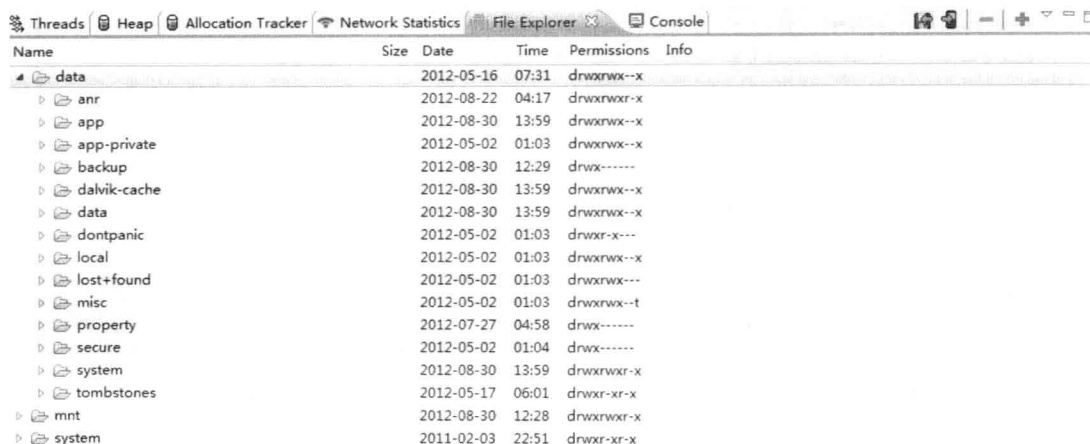
以上 log 的级别依次升高, DEBUG 信息应当只存在于开发中, INFO、WARN、ERROR 这 3 种 log 将出现在发行版中。在使用过程中我们通常声明一个字符串常量 TAG, 通过它来区分不同的 log。例如我们在 HelloAndroid 可以定义 `private static final String TAG="HelloAndroid"`, 这样所有在 HelloAndroid 中使用的 log, 均以“HelloAndroid”开头。我们在 HelloAndroid 的 `onCreate()` 函数中打了一个 log——`Log.i(TAG,"onCreate!")`, 运行后可以看到 log 信息, 如图 1.19 所示。

Time	PID	Application	Tag	Text
08-30 13:59:53.812	2559		AndroidRuntime	>>>>> AndroidRuntime START com.android.internal.os.RuntimeIn
08-30 13:59:53.812	2559		AndroidRuntime	CheckJNI is ON
08-30 13:59:54.512	2559		AndroidRuntime	Calling main entry com.android.commands.pm.Pm
08-30 13:59:54.542	2559		AndroidRuntime	Shutting down VM
08-30 13:59:54.552	2559		dalvikvm	GC_CONCURRENT freed 101K, 71% free 297K/1024K, external 0K/0K
08-30 13:59:54.552	2559		dalvikvm	Debugger has detached; object registry had 1 entries
08-30 13:59:54.593	2559		dalvikvm	JNI: AttachCurrentThread (from ???) failed
08-30 13:59:54.593	2559		AndroidRuntime	NOTE: attach of thread 'Binder Thread #3' failed
08-30 13:59:55.002	2571		AndroidRuntime	>>>>> AndroidRuntime START com.android.internal.os.RuntimeIn
08-30 13:59:55.002	2571		AndroidRuntime	CheckJNI is ON
08-30 13:59:55.642	2571		AndroidRuntime	Calling main entry com.android.commands.am.Am
08-30 13:59:55.662	61	system_process	ActivityManager	Starting: Intent { act=android.intent.action.MAIN cat=[android
08-30 13:59:55.732	2571		AndroidRuntime	Shutting down VM
08-30 13:59:55.742	2571		dalvikvm	GC_CONCURRENT freed 103K, 69% free 319K/1024K, external 0K/0K
08-30 13:59:55.762	2571		dalvikvm	Debugger has detached; object registry had 1 entries
08-30 13:59:55.792	2571		dalvikvm	JNI: AttachCurrentThread (from ???) failed
08-30 13:59:55.792	2571		AndroidRuntime	NOTE: attach of thread 'Binder Thread #3' failed
08-30 13:59:55.862	2503	com.supermario.helloandroid	HelloAndroid	onCreate!
08-30 13:59:56.542	61	system_process	ActivityManager	Displaying com.supermario.helloandroid/.HelloAndroidActivity:
08-30 13:59:57.152	142	com.android.launcher	dalvikvm	GC_EXPLICIT freed 141K, 49% free 3079K/5959K, external 4979K/
08-30 14:00:05.111	142	com.android.launcher	dalvikvm	GC_EXPLICIT freed 47K, 50% free 3032K/5959K, external 4928K/5

图 1.19 onCreate log 信息

还有一个在调试过程中使用频率很高的工具是 File Explore, 如图 1.20 所示, 通过这个工具我们可以方便地查看模拟器的文件系统。我们可以查看文件及文件夹的名称、文件的大小、文件最后修改的日期、文件最后修改的时间、文件的读写权限等。我们也可以通过

右上角的按钮分别实现新建/删除文件、从模拟器中复制文件、往模拟器中复制文件等。



Name	Size	Date	Time	Permissions	Info
data		2012-05-16	07:31	drwxrwx--x	
anr		2012-08-22	04:17	drwxrwxr-x	
app		2012-08-30	13:59	drwxrwx--x	
app-private		2012-05-02	01:03	drwxrwx--x	
backup		2012-08-30	12:29	drwx-----	
dalvik-cache		2012-08-30	13:59	drwxrwx--x	
data		2012-08-30	13:59	drwxrwx--x	
dontpanic		2012-05-02	01:03	drwxr-x---	
local		2012-05-02	01:03	drwxrwx--x	
lost+found		2012-05-02	01:03	drwxrwx---	
misc		2012-05-02	01:03	drwxrwx--t	
property		2012-07-27	04:58	drwx-----	
secure		2012-05-02	01:04	drwx-----	
system		2012-08-30	13:59	drwxrwxr-x	
tombstones		2012-05-17	06:01	drwxr-xr-x	
mnt		2012-08-30	12:28	drwxrwxr-x	
system		2011-02-03	22:51	drwxr-xr-x	

图 1.20 File Explore

1.5 其他工具的使用

除了使用 Eclipse 的 DDMS 进行调试外，我们还可以使用 ADB 工具来进行调试。ADB 的全称为 Android Debug Bridge，借助这个工具，我们可以管理设备或手机模拟器的状态，还可以进行以下的操作：

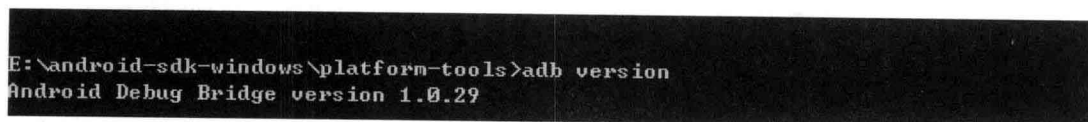
- ❑ 快速更新设备或手机模拟器中的代码，如应用或 Android 系统升级；
- ❑ 在设备上运行 Shell 命令；
- ❑ 管理设备或手机模拟器上的预定端口；
- ❑ 在设备或手机模拟器上复制或粘贴文件。

ADB 的工作方式比较特，殊采用监听 Socket TCP 5554 等端口的方式让 IDE 和 Qemu 通信，默认情况下 ADB 会监听相关的网络端口，所以当我们运行 Eclipse 时 ADB 进程就会自动运行，在 Eclipse 中通过 DDMS 来调试 Android 程序，也可以通过手动方式调用。

ADB 工具的功能很多，下面就几个常用的功能略作说明。

(1) 查看当前 ADB 的版本

在终端输入 `adb version`，可以查看到当前 ADB 的版本，如图 1.21 所示。



```
E:\android-sdk-windows\platform-tools>adb version
Android Debug Bridge version 1.0.29
```

图 1.21 查看 ADB 版本

(2) 查看当前运行的模拟器

在终端输入 `adb devices`，可以看到当前运行的模拟器，如图 1.22 所示。

```
E:\android-sdk-windows\platform-tools>adb devices
List of devices attached
emulator-5554    device
```

图 1.22 查看当前运行的模拟器

(3) 安装软件

命令格式为 `adb install <apk 文件路径>`，这个命令将指定的 `apk` 文件安装到设备上。

(4) 卸载软件

命令格式为 `adb uninstall <软件名>` 或者 `adb uninstall -k <软件名>`，`-k` 参数，表示卸载软件但是保留配置和缓存文件。

(5) 进入设备或模拟器的 shell

命令格式为 `adb shell`，通过这个命令，就可以进入设备或模拟器的 `shell` 环境中。在这个 `Linux Shell` 中，你可以执行各种 `Linux` 的命令，另外如果只想执行一条 `shell` 命令，可以采用以下方式：

```
adb shell [command]
```

例如，`adb shell ls` 会显示出当前目录下的文件信息。

(6) 发布端口

可以设置任意的端口号，作为主机向模拟器或设备的请求端口。如：

```
adb forward tcp:5555 tcp:8000
```

(7) 从电脑上发送文件到设备

命令格式为 `adb push <本地路径> <远程路径>`，用 `push` 命令可以把本机电脑上的文件或者文件夹复制到设备（手机）。

(8) 从设备上下载文件到电脑

命令格式为 `adb pull <远程路径> <本地路径>`，用 `pull` 命令可以把设备（手机）上的文件或者文件夹复制到本机电脑。

(9) 记录无线通讯日志

一般来说，无线通讯的日志非常多，在运行时没必要去记录，但我们还是可以通过 `adb shell` 和 `logcat -b radio` 命令来设置记录。

1.6 本章小结

本章主要介绍了 `Android` 开发环境的搭建，以及如何利用 `Eclipse` 自带的 `DDMS` 工具及 `SDK` 中的 `ADB` 工具调试程序。本章我们学会了如何去编写一个最简单的 `Android` 应用程序——`HelloAndroid`，以及如何在模拟器中查看运行的结果，在 `logcat` 工具中查看自己写的 `log`。

第 2 章 Android 程序开发基础

上一章我们讲述了 Android 平台的框架和特性以及 Android 开发环境的搭建，这部分虽然繁琐，但却是学习 Android 开发必须首先了解的。接下来这章我们要讲解 Android 程序开发必备的一些基础知识。

2.1 Android 项目结构分析

我们通过 Eclipse 创建 Android 工程的时候，Eclipse 会自动为我们生成一个目录结构，这个目录结构是我们今后开发所有应用程序的“原型”，因此我们有必要深入了解一下这个目录结构的各个部分。

上一章我们建立了一个 HelloAndroid 的项目，现在我们把它全部展开，如图 2.1 所示。

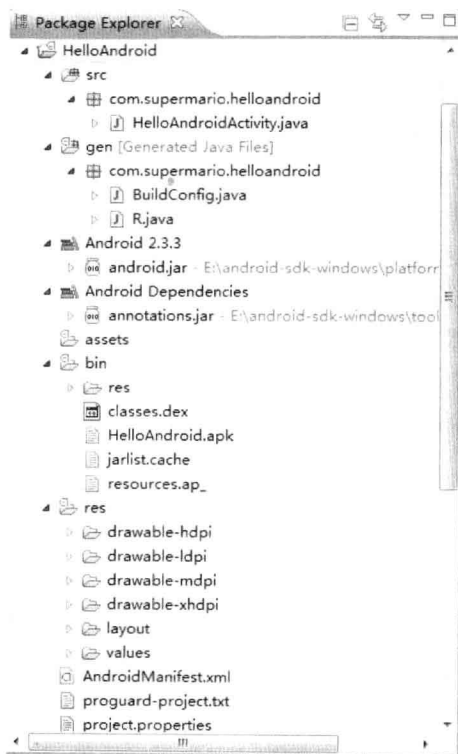


图 2.1 HelloAndroid 项目结构

(1) **src**: 和普通的 Java 项目一样，这是项目所有的包和源文件存放的目录。

(2) res: 资源 (Resource) 目录。

在这个目录中我们可以存放应用使用到的各种资源, 如 xml 界面文件、图标或常量。

❑ res/drawable——专门存放图标文件。

❑ res/layout——专门存放 xml 界面文件, xml 界面文件和 HTML 文件一样、主要用于用户界面显示。

❑ res/values——专门存放应用使用到的各种常量, 作用和 struts 中的国际化资源文件一样。

(3) gen: gen 目录中存放所有由 Android 开发工具自动生成的文件。目录中最重要的就是 R.java 文件, 这个文件是由 Android 开发工具自动产生的。Android 开发工具会自动根据你放入 res 目录的 xml 界面文件、图标与常量, 同步更新修改 R.java 文件。正因为 R.java 文件是由开发工具自动生成的, 所以我们应避免手工修改 R.java。R.java 在应用中起到了字典的作用, 它包含了界面、图标、常量等各种资源的 id, 通过 R.java, 应用可以很方便地找到对应资源。另外编译器也会检查 R.java 列表中的资源是否被使用到, 没有被使用到的资源不会编译进软件中, 这样可以减少应用在手机中占用的空间。

HelloAndroid 的 R.java 文件代码如下所示, 里面包含类 drawable、layout、string, 分别对应了 res 目录中相应的 drawable、layout、values 中的文件。

```
01  /* AUTO-GENERATED FILE. DO NOT MODIFY.
02  *
03  * This class was automatically generated by the
04  * aapt tool from the resource data it found. It
05  * should not be modified by hand.
06  */
07
08  package com.supermario.helloandroid;
09  <!-- Eclipse 自动生成-->
10  public final class R {
11      public static final class attr {
12      }
13      public static final class drawable {
14          public static final int ic_launcher=0x7f020000;
15      }
16      public static final class layout {
17          public static final int main=0x7f030000;
18      }
19      public static final class string {
20          public static final int app_name=0x7f040001;
21          public static final int hello=0x7f040000;
22      }
23  }
```

(4) AndroidManifest.xml: 功能清单文件。

如以下代码所示, 这个文件列出了应用程序所提供的所有功能。每当你添加一个新的 Activity 时, 就需要在这个文件中作相应的配置, 否则应用程序就无法识别和使用这个 Activity。在这里, 你也可以指定应用程序需要用到的服务、接收器等, 以及它们对应的 <Intent-filter>, 用于通过 intent 的方式打开指定服务或接收器。此外, 程序需要的权限也需要在这里设置。

```
01  <?xml version="1.0" encoding="utf-8"?>
02  <manifest xmlns:android="http://schemas.android.com/apk/res/android">
```



```

03     package="com.supermario.helloandroid"
04     android:versionCode="1"
05     android:versionName="1.0" >
06     <!--使用的 SDK 最低版本-->
07     <uses-sdk android:minSdkVersion="10" />
08
09     <application
10         android:icon="@drawable/ic_launcher"
11         android:label="@string/app_name" >
12     <!--主体应用程序-->
13         <activity
14             android:name=".HelloAndroidActivity"
15             android:label="@string/app_name" >
16             <intent-filter>
17                 <action android:name="android.intent.action.MAIN" />
18     <!--设置为默认类别-->
19                 <category android:name="android.intent.category.
20                     LAUNCHER" />
21             </intent-filter>
22         </activity>
23     </application>
24 </manifest>

```

表 2.1 详细描述了 AndroidManifest 部分标签及元素的作用。

表 2.1 AndroidManifest.xml 元素功能介绍

manifest	根节点，描述了 package 中所有的内容
xmlns:android	包含命名空间的声明。xmlns:android=http://schemas.android.com/apk/res/android，使得 Android 中各种标准属性能在文件中使用，提供了大部分元素中的数据
application	包含 package 中 application 级别组件声明的根节点。此元素也可包含 application 的一些全局和默认的属性，如标签、icon、主题、必要的权限，等等。一个 manifest 能包含零个或一个此元素（不能大于一个）
android:icon	应用程序图标
android:label	应用程序名字
Activity	用来与用户交互的主要工具。Activity 是用户打开一个应用程序的初始页面，大部分被使用到的其他页面也由不同的 activity 所实现，并声明在另外的 activity 标记中。注意，每一个 activity 必须有一个<activity>标记对应，无论它给外部使用或是只用于自己的 package 中。如果一个 activity 没有对应的标记，你将不能运行它。另外，为了支持运行时查找 Activity，可包含一个或多个<intent-filter>元素来描述 activity 所支持的操作
android:name	应用程序默认启动的 activity
intent-filter	声明了指定的一组组件支持的 Intent 值，从而形成了 IntentFilter。除了能在此元素下指定不同类型的值，属性也能放在这里来描述一个操作所需的唯一的标签、icon 和其他信息
action	组件支持的 Intent action
category	组件支持的 Intent Category。这里指定了应用程序默认启动的 activity
uses-sdk	该应用程序所使用的 SDK 版本信息

(5) default properties: 系统默认信息，一般不需要修改此文件。

(6) assets: Android 系统为每个新设计的程序提供了 assets 目录，这个目录保存的文件可以打包在程序里。它与 res 目录不同点在于，Android 不为 assets 下的文件生成 id，如

果要使用 `assets` 下的文件，需要指定文件的路径和文件名。

(7) `bin`: 存放每次编译生成的文件，编译生成的 `HelloAndroid.apk` 就存放在这个目录。

2.2 Android 四大组件

Android 开发四大组件分别是：

- ❑ 活动 (Activity)：用于表现功能。
- ❑ 服务 (Service)：后台运行服务，不提供界面呈现。
- ❑ 广播接收器 (BroadcastReceiver)：用于接收广播。
- ❑ 内容提供商 (ContentProvider)：支持在多个应用中存储和读取数据，相当于数据库。

2.2.1 Activity

在 Android 中，Activity 是所有程序的根本，所有程序的流程都运行在 Activity 之中，Activity 可以算是开发者遇到的最频繁，也是 Android 当中最基本的模块之一。在 Android 的程序当中，Activity 一般代表手机屏幕的一屏。如果把手机比作一个浏览器，那么 Activity 就相当于一个网页。在 Activity 当中可以添加一些 Button、Check box 等控件。可以看到 Activity 概念和网页的概念相当类似。

一般一个 Android 应用是由多个 Activity 组成的。这多个 Activity 之间可以进行相互跳转，例如，按下一个 Button 按钮后，可能会跳转到其他的 Activity。和网页跳转稍微有些不一样的是，Activity 之间的跳转可能有返回值。例如，从 Activity A 跳转到 Activity B，那么当 Activity B 运行结束的时候，有可能会给 Activity A 一个返回值。这样做在很多时候是相当方便的。

当打开一个新的屏幕时，之前一个屏幕会被置为暂停状态，并且压入历史堆栈中。用户可以通过回退操作返回到以前打开过的屏幕。我们可以选择性地移除一些没有必要保留的屏幕，因为 Android 会把每个应用从开始到当前的每个屏幕都保存在堆栈中。

Android 用 Intent 这个特殊类实现在 Activity 与 Activity 之间的切换。Intent 类用于描述应用的功能。在 Intent 的描述结构中，有两个最重要的部分：动作和动作对应的数据。典型的动作类型有：MAIN、VIEW、PICK、EDIT 等，而动作对应的数据则以 URI 的形式表示。例如，要查看一个人的联系方式，需要创建一个动作类型为 VIEW 的 Intent，以及一个表示这个人的 URI。

通过解析各种 Intent，从一个屏幕导航到另一个屏幕是很简单的。当向前导航时，Activity 将会调用 `StartActivity(Intent my Intent)` 方法。然后，系统会在所有已安装的应用程序中定义的 `IntentFilter` 中查找，找到最匹配 `myIntent` 的 Intent 对应的 Activity。新的 Activity 接收到 `MyIntent` 的通知后，开始运行。当 `StartActivity` 方法被调用时，将触发解析 `MyIntent` 的动作，该机制提供了两个关键好处：

- ❑ Activity 能够重复利用从其他组件中以 Intent 的形式产生的请求。
- ❑ Activity 可以在任何时候被具有相同 `IntentFilter` 的新的 Activity 取代。

下面我们举例来说明两个 Activity 之间的切换。在 Eclipse 中新建一个项目 IntentTest, 新建两个 Activity: IntentTest Activity 和 AnotherActivity。IntentActivity.java 代码如下所示, 定义了一个按钮并为这个按钮绑定监听事件, 当用户单击这个按钮的时候, 将会切换到 AnotherActivity, 同时关闭当前的 Activity。

```

01 package com.supermario.intenttest;
02 import android.app.Activity;
03 import android.content.Context;
04 import android.content.Intent;
05 import android.os.Bundle;
06 import android.view.View;
07 import android.view.View.OnClickListener;
08 import android.widget.Button;
09 public class IntentTestActivity extends Activity {
10     /** Called when the activity is first created. */
11     @Override
12     public void onCreate(Bundle savedInstanceState) {
13         super.onCreate(savedInstanceState);
14         //设置显示main 的布局
15         setContentView(R.layout.main);
16         //取得布局main 中的 button1 按钮
17         Button btn=(Button)findViewById(R.id.button1);
18         //取得当前程序的上下文
19         final Context context=this;
20         //设置 btn 的按键监听器
21         btn.setOnClickListener(new OnClickListener() {
22             @Override
23             public void onClick(View v) {
24                 // TODO Auto-generated method stub
25                 //新建一个 Intent 对象
26                 Intent it=new Intent();
27                 //指定 it 要启动的类
28                 it.setClass(context, AnotherActivity.class);
29                 //启动 it 指定的类
30                 startActivity(it);
31                 //关闭当前的 Activity
32                 finish();
33             }
34         });
35     }
36 }

```

AnotherActivity.java 的代码如下所示, 同样定义了一个按钮并为该按钮绑定监听器, 当用户单击这个按钮的时候, 就会启动 IntentTestActivity, 并关闭当前的 Activity。

```

01 package com.supermario.intenttest;
02 import android.app.Activity;
03 import android.content.Context;
04 import android.content.Intent;
05 import android.os.Bundle;
06 import android.view.View;
07 import android.view.View.OnClickListener;
08 import android.widget.Button;
09 public class AnotherActivity extends Activity {
10     /** Called when the activity is first created. */
11     @Override
12     public void onCreate(Bundle savedInstanceState) {

```

```

13      super.onCreate(savedInstanceState);
14      //设置当前的布局为 main2
15      setContentView(R.layout.main2);
16      //取得布局 main2 中的 button1 按钮
17      Button btn=(Button)findViewById(R.id.button1);
18      //取得当前程序的上下文
19      final Context context=this;
20      //设置 btn 的按键监听器
21      btn.setOnClickListener(new OnClickListener(){
22          @Override
23          public void onClick(View v) {
24              // TODO Auto-generated method stub
25              //新建一个 Intent 对象
26              Intent it=new Intent();
27              //指定 it 要启动的类
28              it.setClass(context, IntentTestActivity.class);
29              //启动 it 指定的类
30              startActivity(it);
31              //关闭当前的 Activity
32              finish();
33          }
34      });
35  }
36 }

```

布局文件 main.xml 代码如下，定义了一个 TextView 和 Button。

```

01 <?xml version="1.0" encoding="utf-8"?>
02 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/
    android"
03     android:layout_width="fill_parent"
04     android:layout_height="fill_parent"
05     android:orientation="vertical" >
06     <!--用于显示文字-->
07     <TextView
08         android:layout_width="fill_parent"
09         android:layout_height="wrap_content"
10         android:text="第一个 Activity! " />
11     <!--切换按钮-->
12     <Button
13         android:id="@+id/button1"
14         android:layout_width="wrap_content"
15         android:layout_height="wrap_content"
16         android:text="切换到第二个 Activity" />
17 </LinearLayout>

```

布局文件 main2.xml 代码如下，同样是定义了一个 TextView 和 Button。

```

01 <?xml version="1.0" encoding="utf-8"?>
02 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/
    android"
03     android:layout_width="fill_parent"
04     android:layout_height="fill_parent"
05     android:orientation="vertical" >
06     <!--用于显示文字-->
07     <TextView
08         android:layout_width="fill_parent"
09         android:layout_height="wrap_content"
10         android:text="第二个 Activity! " />

```



```

11     <!--用于显示按钮-->
12     <Button
13         android:id="@+id/button1"
14         android:layout_width="wrap_content"
15         android:layout_height="wrap_content"
16         android:text="切换到第一个 Activity" />
17 </LinearLayout>

```

在 Androidmanifest.xml 中声明使用 IntentTestActivity 和 AnotherActivity，如下所示：

```

01 <?xml version="1.0" encoding="utf-8"?>
02 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
03     package="com.supermario.intenttest"
04     android:versionCode="1"
05     android:versionName="1.0" >
06     <!--使用的最低 SDK 版本-->
07     <uses-sdk android:minSdkVersion="10" />
08     <application
09         android:icon="@drawable/ic_launcher"
10         android:label="@string/app_name" >
11         <!--主体 Activity-->
12         <activity
13             android:name=".IntentTestActivity"
14             android:label="@string/app_name" >
15             <intent-filter>
16                 <action android:name="android.intent.action.MAIN" />
17                 <category android:name="android.intent.category.LAUNCHER" />
18             </intent-filter>
19         </activity>
20         <!--声明另一个 Activity-->
21         <activity android:name=".AnotherActivity">
22             <intent-filter>
23                 <action android:name="android.guo.action.anotheractivity" />
24             </intent-filter>
25         </activity>
26     </application>
27 </manifest>

```

最后我们运行该程序，得到如图 2.2 所示的效果，单击“切换到第二个 Activity”，可以切换到第二个 AnotherActivity，如图 2.3 所示。再单击“切换到第一个 Activity”可以切换回 IntentTestActivity。

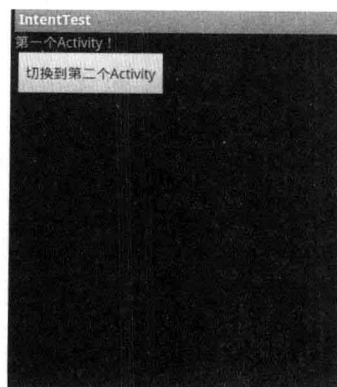


图 2.2 IntentTestActivity

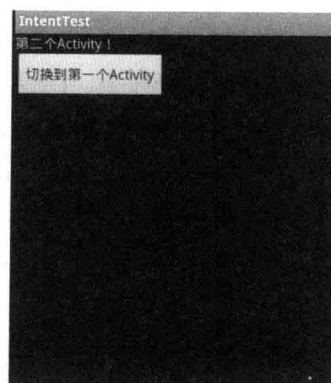


图 2.3 AnotherActivity

2.2.2 Service

Service 是 Android 系统中的一种组件，它跟 Activity 的级别差不多，但是它不能自己运行，只能后台运行，并且可以和其他组件进行交互。Service 是没有界面的长生命周期的代码。Service 是一种程序，它可以运行很长时间，但是它却没有用户界面。这么说有点枯燥，来看个例子，打开一个音乐播放器的程序，这个时候若想上网了，那么，我们打开 Android 浏览器，这个时候虽然我们已经进入了浏览器程序，但是，歌曲播放并没有停止，而是在后台继续一首接着一首地播放。其实这个播放就是由播放音乐的 Service 进行控制的。

当然这个播放音乐的 Service 也可以停止，例如，当播放列表里边的歌曲都结束，或者用户按下了停止音乐播放的快捷键等。Service 可以在很多场合的应用中使用，比如播放多媒体的时候用户启动了其他 Activity，这个时候程序要在后台继续播放，比如检测 SD 卡上文件的变化，再或者在后台记录用户地理信息位置的改变等等。总之服务嘛，总是藏在后头的。

开启 Service 有两种方式。

(1) Context.startService(): Service 会经历 onCreate→onStart (如果 Service 还没有运行，则 Android 先调用 onCreate()然后调用 onStart(); 如果 Service 已经运行，则只调用 onStart(), 所以一个 Service 的 onStart 方法可能会重复调用多次); stopService 的时候直接 onDestroy, 如果是调用者自己直接退出而没有调用 stopService 的话，Service 会一直在后台运行。该 Service 的调用者再启动起来后可以通过 stopService 关闭 Service。注意，多次调用 Context.startService()不会嵌套（即使会有相应的 onStart()方法被调用），所以无论同一个服务被启动了多少次，一旦调用 Context.stopService()或者 stopSelf(), 它都会被停止。补充说明：传递给 startService() 的 Intent 对象会传递给 onStart()方法。调用顺序为：onCreate→onStart (可多次调用)→onDestroy。

(2) Context.bindService(): Service 会经历 onCreate()→onBind(), onBind 将返回给客户端一个 IBinder 接口实例，IBinder 允许客户端回调服务的方法，比如得到 Service 运行的状态或其他操作。这个时候调用者 (Context, 如 Activity) 会和 Service 绑定在一起，Context 退出了，Service 就会调用 onUnbind→onDestroy 相应退出。所谓绑定在一起就共存亡了。

下面我们以一个小例子来学习一下 Service 的用法。在这个例子中，我们在一个 Activity 中设置两个按钮，一个按钮用来启动 Service，一个按钮用来停止 Service。在 Service 中使用相应的函数实现音乐的播放和停止，项目的目录结构如图 2.4 所示。

如下是 PlayMusicActivity.java 的代码，在代码中定义了两个按钮，并分别为这两个按钮绑定了监听器，在这两个监

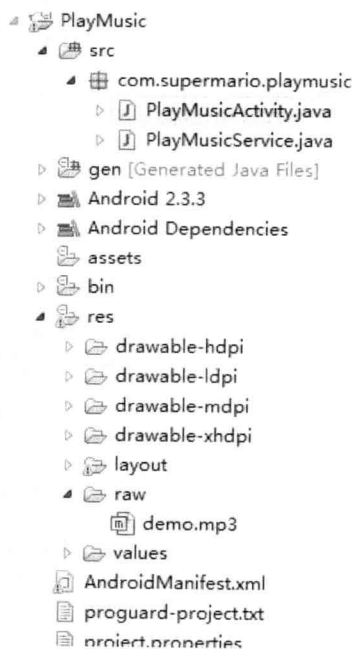


图 2.4 PlayMusic 目录结构

听器中分别打开和关闭指定的 Service。

```

01 package com.supermario.playmusic;
02 import android.app.Activity;
03 import android.content.Intent;
04 import android.os.Bundle;
05 import android.view.View;
06 import android.view.View.OnClickListener;
07 import android.widget.Button;
08 public class PlayMusicActivity extends Activity {
09     /** Called when the activity is first created. */
10     @Override
11     public void onCreate(Bundle savedInstanceState) {
12         super.onCreate(savedInstanceState);
13         setContentView(R.layout.main);
14         //开始按钮
15         Button start=(Button)findViewById(R.id.button1);
16         //停止按钮
17         Button stop=(Button)findViewById(R.id.button2);
18         //用于启动和停止 Service 的 Intent
19         final Intent it=new Intent("android.guo.service.playmusic");
20         //为"开始"按钮绑定监听器
21         start.setOnClickListener(new OnClickListener() {
22             @Override
23             public void onClick(View v) {
24                 // TODO Auto-generated method stub
25                 //启动服务
26                 startService(it);
27             }
28         });
29         //为"停止"按钮绑定监听器
30         stop.setOnClickListener(new OnClickListener() {
31             @Override
32             public void onClick(View v) {
33                 // TODO Auto-generated method stub
34                 //停止服务
35                 stopService(it);
36             }
37         });
38     }
39 }

```

如下是 PlayMusicService.java 的代码，在服务启动的时候将会执行 onStart()，在关闭的时候将会执行 onDestroy()，因此，我们在 onStart()函数中开始播放音乐，在 onDestroy()中停止音乐。程序中用到的音乐是我们预先放到 res/raw/目录下的一个 demo.mp3 文件。

```

01 package com.supermario.playmusic;
02 import android.app.Service;
03 import android.content.Intent;
04 import android.media.MediaPlayer;
05 import android.os.IBinder;
06 public class PlayMusicService extends Service{
07     //定义音乐播放器
08     private MediaPlayer mMediaPlayer;
09     @Override
10     public IBinder onBind(Intent arg0) {
11         // TODO Auto-generated method stub
12         return null;
13     }

```

提供各种书籍的pd电子版代找服务，如果你找不到自己想要的书的pdf电子版，我们可以帮您找到，如有需要，请联系QQ1779903665.

PDF代找说明：

本人可以帮助你找到你要的PDF电子书，计算机类，文学，艺术，设计，医学，理学，经济，金融，等等。质量都很清晰，而且每本100%都带书签索引和目录，方便读者阅读观看，只要您提供给我书的相关信息，一般我都能找到，如果您有需求，请联系我QQ1779903665。

本人已经帮助了上万人找到了他们需要的PDF，其实网上有很多PDF,大家如果在网上不到的话，可以联系我QQ，大部分我都可以找到，而且每本100%带书签索引目录。因PDF电子书都有版权，请不要随意传播，如果您有经济购买能力，请尽量购买正版。

声明：本人只提供代找服务，每本100%索引书签和目录，因寻找pdf电子书有一定难度，仅收取代找费用。如因PDF产生的版权纠纷，与本人无关，我们仅仅只是帮助你寻找到你要的pdf而已。