

Android/OPhone

李 宁 编著

开发完全讲义

- 国内覆盖面最广的Android开发指南
- 国内第一本同时涉及Android和OPhone技术的参考手册
- 全书贯穿Android 1.5至Android 2.1五大Android版本的重要特性
- 超过200个实际案例，提供完美的实战环境
- 大量有价值的代码可以直接用于实际的项目
- 一线开发人员倾力之作，使全书的内容更接近实战
- 完整的综合应用案例，理论与实战相结合



中国水利水电出版社
www.waterpub.com.cn

Android/OPhone 开发完全讲义

李宁 编著



中国水利水电出版社
www.waterpub.com.cn

内 容 提 要

本书是国内第一本同时介绍 Android 和 OPhone 的经典著作,国内著名 Android 社区 eoeandroid 极力推荐。

本书内容全面,详细讲解 Android 框架、Android 用户接口、Android 组件、Android 的数据存储解决方案、Android 的 4 种可跨平台通信的组件 (Activity、Service、Broadcast 和 ContentProvider)、网络、绘图、多媒体、2D 动画、OpenGL ES、资源、国际化、访问 Android 手机的硬件、实时壁纸、实时文件夹、NDK (利用 C/C++ 开发可运行在 Android 上的应用程序)、脚本语言 (Python、Lua、Perl 等)、手势输入、文字语音转换 (TTS)、蓝牙及 OPhone 平台的技术。除此之外,在 OPhone 平台中内嵌了一种新的 SDK: JIL Widget。这种技术可以只使用 HTML、CSS、JavaScript 等 Web 技术来编写可运行在 OPhone 上的程序。为了使读者更早进入实战阶段,在本书的最后给出了两个完整的例子:“万年历”和“知道当前位置的 Google GTalk 机器人”。

本书适合有一定的 Java 基础,想通过 Android 进入移动开发领域的读者;已经有一定的 Android 开发经验,想进一步提高 Android 的开发能力的读者;想将本书作为 Android 的参考手册,随时随地查阅的读者;对 Android 抱有浓厚兴趣的其他手机平台的开发人员;正在学习 Android 的在校大学生以及培训学校的学员。

图书在版编目 (C I P) 数据

Android/OPhone 开发完全讲义 / 李宁编著. — 北京
: 中国水利水电出版社, 2010.6
ISBN 978-7-5084-7527-1

I. ①A… II. ①李… III. ①移动通信—携带电话机—应用程序—程序设计②移动通信—携带电话机—操作系统—程序设计 IV. ①TN929.53

中国版本图书馆 CIP 数据核字 (2010) 第 092850 号

策划编辑: 周春元 责任编辑: 宋俊娥 加工编辑: 俞 飞 封面设计: 李 佳

书 名	Android/OPhone 开发完全讲义
作 者	李宁 编著
出版发行	中国水利水电出版社 (北京市海淀区玉渊潭南路 1 号 D 座 100038) 网址: www.waterpub.com.cn E-mail: mchannel@263.net (万水) sales@waterpub.com.cn 电话: (010) 68367658 (营销中心)、82562819 (万水)
经 售	全国各地新华书店和相关出版物销售网点
排 版	北京万水电子信息有限公司
印 刷	北京蓝空印刷厂
规 格	210mm×285mm 16 开本 30 印张 896 千字
版 次	2010 年 6 月第 1 版 2010 年 6 月第 1 次印刷
印 数	0001—4000 册
定 价	58.00 元 (赠 1CD)

凡购买我社图书,如有缺页、倒页、脱页的,本社营销中心负责调换
版权所有·侵权必究

前 言

2009 年对于中国的移动互联网领域是最重要的一年，因为这一年信息产业部向中国三大运营商（移动、联通、电信）发放了 3G 牌照。这就意味着中国移动互联网 3G 时代已经到来，因此，2009 年也就成为了中国移动互联网 3G 时代的元年。

随着中国移动互联网 3G 时代的到来，很多抢眼的词汇也不断冲击着人们的眼球。3G、Android、OPhone、iPhone、iPad、iPod touch、Google、苹果、HTC、G1、G2、G3 等词汇在各大网站的新闻和评论中频繁出现。出现这种情况的原因只有一个，那就是智能手机和移动互联网终于修成成果，成为完美的一对。

智能手机虽然早在多年前就已出现，但那时的智能手机的功能主要是本地应用，这些手机以 Nokia 的 Symbian 系统为主。但随着移动互联网时代的到来，本地应用已远远无法满足用户的需求了。然而在创新为王的今天，新的技术总在不断地取代旧的技术。以创新闻名的 Google 为了进军移动广告市场，早在 2005 年，就开始研制新的移动操作系统，这也就是我们现在熟知的 Android。经过 2 年多的研发，终于在 2007 年 11 月 5 日发布了 Android 的第一个版本：Android 1.0，而 HTC（宏达电子）也在 10 个月发布了世界上第一部装有 Android 系统的手机：G1。这也标志着 Android 正式成为移动操作系统大家族的成员，而且出身名门。

自从 Android 问世以来，不断有新的运营商、终端厂商、浏览器厂商、软件厂商等加入 Android 阵营，通过合作以及不断地创新，推出了大量基于 Android 的新产品，例如，Adobe 公司已推出 Android 版的 AIR 和 Flash10，并且 Google 宣布在 Android 2.2 中将会全面支持 Flash，这就意味着可以使用 Flash 来编写 Android 应用程序了。

Android 是 Google 进军移动领域最具杀伤力的武器之一。在此之前，苹果推出的 iPhone 在智能手机和移动互联网领域刮起了首轮风暴，并且赢得了数以百万计的忠实“粉丝”。而 Android 与 iPhone 不同。iPhone 与苹果以往的产品相同，都是在封闭状态下发展的，而且限制太多。Android 则在这方面有着绝对的优势，Android 不仅免费，而且开源，并且 Google 没有限制使用什么语言或技术在 Android 上开发软件。这就意味着任何企业、组织和个人都可以使用 Android 系统，而且不需要付给 Google 一分钱。正是因为这一点，支持 Android 的终端厂商不断增加，这也使 Android 的市场占用率节节攀升。甚至在 2010 年第一季度，Android 在美国的占有率首次超过 iPhone，成为占有率居第二位的移动操作系统。

在与美国相隔万里的中国，Android 也受到相当的关注和重用。国内不仅在短时间内涌现了大量的 Android 社区（中国移动开发者社区、eoeandroid 等），而且很多运营商和企业也以 Android 为基础开发出了很多定制的移动操作系统。其中国内最早的定制 Android 系统就是中国移动和播思通讯联合开发的 OPhone 系统，也称为 OMS。除此之外，还有联想的 LePhone。联通和电信也在研发自己的定制 Android 系统。这些充分证明了 Android 在世界上的几个主要市场（中国、美国等）都已成为竞相追逐的梅花鹿，在可预期的未来，将会在全球范围内上演一场群雄逐鹿的大戏。

既然 Android 无论在国内还是国外都是如此的火爆，如此的重要，那么作为开发人员的我们是不是应该立刻开始学习 Android 呢？如果您正在阅读本书的前言，那么说明您已经给出了肯定的答案，而本书正是打开 Android 神秘大门的钥匙。有了这把钥匙，就可以尽享 Android 中的宝藏。那么我们还等什么呢？Let's go. 现在就让我们继续阅读本书的精彩内容，以获取更多的宝藏吧！

编 者
2010 年 5 月

如何使用本书的例子

本书的所有例子代码都在光盘的 src 目录中。其中每一章的例子代码被单独放在了相应的目录中。例如，第 10 章的例子代码在 src\ch10 目录中。读者在运行程序之前，建议先将光盘中的源代码复制到硬盘中的指定目录，再使用 Eclipse 导入相应的工程。导入的方法是单击 Eclipse 的【File】>【Import】菜单项，打开【Import】对话框，选择【Existing Projects into Workspace】节点，如图 1 所示。单击【Next】按钮进行下一个页面后，单击【Browse...】按钮选择要导入的 Android 工程，如图 2 所示。最后单击【Finish】按钮即可导入 Android 工程。

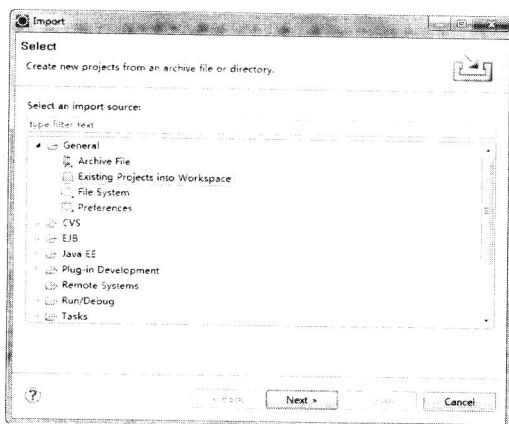


图 1 选择【Existing Projects into Workspace】节点

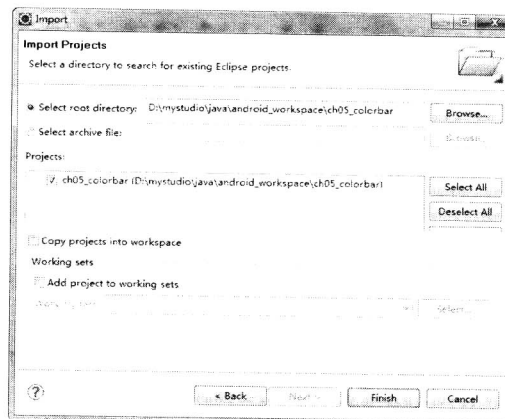


图 2 选择要导入的工程

在成功导入 Android 工程后，工程目录可能会出现错误。原因是该工程引用了作者本机的一些 jar 包，而在读者的机器上没有这些 jar 包。读者可以在工程属性对话框中重新设置这些 jar 包的路径。要注意的是，有一些和硬件相关的程序，如 GPS、录音、照相机等，需要直接在手机上进行测试，这些程序可能在 Android 模拟器中成功运行，但却会显示异常的运行结果。读者应尽量在手机上测试这些程序。

目 录

前言

如何使用本书的例子

第一部分 准备篇——大军未动，粮草先行

第 1 章 Android 入门	2
1.1 Android 的基本概念	2
1.1.1 Android 简介	2
1.1.2 Android 的系统构架	4
1.2 Android 开发环境的搭建	5
1.2.1 开发 Android 程序需要些什么	5
1.2.2 安装 Android SDK	5
1.2.3 安装 Eclipse 插件 ADT	6
1.2.4 测试 ADT 是否安装成功	6
1.3 Android SDK 中的常用命令行工具	8
1.3.1 启动和关闭 ADB 服务 (adb start-server 和 adb kill-server)	8
1.3.2 查询当前模拟器/设备的实例 (adb devices)	8
1.3.3 安装、卸载和运行程序 (adb install、adb uninstall 和 am)	8
1.3.4 PC 与模拟器或真机交换文件 (adb pull 和 adb push)	9
1.3.5 Shell 命令	9
1.3.6 创建、删除和浏览 AVD 设备 (android)	10

1.3.7 创建 SD 卡	10
1.4 PC 上的 Android	10
1.4.1 Android LiveCD	11
1.4.2 AndroidX86	11
1.5 Android 的学习资源	12
1.6 应用程序商店	12
1.6.1 Android Market	12
1.6.2 Mobile Market (MM)	12
1.6.3 其他应用程序商店	13
1.7 本章小结	13

第 2 章 第一个 Android 程序	14
2.1 编写用于显示当前日期和时间的程序	14
2.1.1 新建一个 Android 工程	14
2.1.2 界面组件的布局	15
2.1.3 编写实际代码	15
2.2 调试程序	18
2.3 签名和发布应用程序	18
2.3.1 使用命令行方式进行签名	19
2.3.2 使用 ADT 插件方式进行签名	19
2.4 DDMS 透视图	20
2.5 本章小结	21

第二部分 基础篇——Android 世界的精彩之旅

第 3 章 Android 应用程序架构	23
3.1 Android 应用程序中的资源	23
3.1.1 资源存放在哪里	23
3.1.2 资源的种类	23
3.1.3 资源的基本使用方法	24
3.2 Android 的应用程序组件	25
3.2.1 活动 (Activity) 组件	25
3.2.2 服务 (Service) 组件	25
3.2.3 广播接收者 (Broadcast receivers) 组件	26
3.2.4 内容提供者 (Content providers) 组件	26

3.3 AndroidManifest.xml 文件的结构	26
3.4 本章小结	27
第 4 章 建立用户接口	28
4.1 建立、配置和使用 Activity	28
4.1.1 建立和配置 Activity	28
4.1.2 Activity 的生命周期	30
实例 1: 演示 Activity 的生命周期	31
4.2 视图 (View)	33
4.2.1 视图简介	34
4.2.2 使用 XML 布局文件控制视图	34
4.2.3 在代码中控制视图	35
实例 2: 混合使用 XML 布局文件和代码	

来控制视图	37	实例 14: 霓虹灯效果的 TextView	77
4.2.4 定制组件 (Widget) 的三种方式	39	4.6.2 线性布局 (LinearLayout)	78
实例 3: 定制组件——带图像的 TextView	39	实例 15: 利用 LinearLayout 将按钮放在屏幕 的四角和中心位置	80
实例 4: 定制组件——带文本标签的 EditText	42	4.6.3 相对布局 (RelativeLayout)	81
实例 5: 定制组件——可更换表盘的指针 时钟	44	实例 16: 利用 RelativeLayout 实现梅花效果 的布局	81
4.3 使用 AlertDialog 类创建对话框	49	4.6.4 表格布局 (TableLayout)	82
4.3.1 AlertDialog 类简介	49	实例 17: 计算器按钮的布局	82
4.3.2 【确认/取消】对话框	49	4.6.5 绝对布局 (AbsoluteLayout)	83
实例 6: 创建询问是否删除文件的 【确认/取消】对话框	50	4.7 本章小结	83
4.3.3 带 3 个按钮的对话框	51	第 5 章 组件详解	84
实例 7: 创建【覆盖/忽略/取消】对话框	51	5.1 显示和编辑文本的组件	84
4.3.4 简单列表对话框	52	5.1.1 显示文本的组件: TextView	85
4.3.5 单选列表对话框	53	实例 18: 在 TextView 中显示 URL 及不同 字体大小、不同颜色的文本	86
4.3.6 多选列表对话框	53	实例 19: 带边框的 TextView	88
实例 8: 创建 3 种选择省份的列表对话框	54	5.1.2 输入文本的组件: EditText	89
4.3.7 水平进度对话框和圆形进度对话框	58	实例 20: 在 EditText 中输入特定的字符	90
实例 9: 水平进度对话框和圆形进度 对话框演示	58	实例 21: 按回车键显示 EditText	91
4.3.8 自定义对话框	61	5.1.3 自动完成输入内容的组件: AutoCompleteTextView	92
实例 10: 创建登录对话框	61	5.2 按钮与复选框组件	93
实例 11: 使用 Activity 托管对话框	63	5.2.1 普通按钮组件: Button	93
实例 12: 创建悬浮对话框和触摸任何位置 都可以关闭的对话框	64	实例 22: 异形 (圆形、五角星、螺旋形 和箭头) 按钮	94
4.4 Toast 和 Notification	67	5.2.2 图像按钮组件: ImageButton	95
4.4.1 用 Toast 显示提示信息框	67	实例 23: 同时显示图像和文字的按钮	96
4.4.2 Notification 与状态栏信息	68	5.2.3 选项按钮组件: RadioButton	96
4.5 菜单	70	5.2.4 开关状态按钮组件: ToggleButton	97
4.5.1 创建选项菜单	70	5.2.5 复选框组件: CheckBox	97
4.5.2 设置与菜单项关联的图像和 Activity	71	实例 24: 利用 XML 布局文件动态创建 CheckBox	98
4.5.3 响应选项菜单项单击事件的 3 种方式	71	5.3 日期与时间组件	100
4.5.4 动态添加、修改和删除选项菜单	72	5.3.1 输入日期的组件: DatePicker	100
4.5.5 创建带复选框和选项按钮的子菜单	73	5.3.2 输入时间的组件: TimePicker	100
4.5.6 创建上下文菜单	74	实例 25: DatePicker、TimePicker 与 TextView 同步显示日期和时间	100
4.5.7 菜单事件	75	5.3.3 显示时钟的组件: AnalogClock 和 DigitalClock	101
实例 13: Activity 菜单、子菜单、上下文 菜单演示	75	5.4 进度条组件	102
4.6 布局	76	5.4.1 进度条组件: ProgressBar	102
4.6.1 框架布局 (FrameLayout)	76	5.4.2 拖动条组件: SeekBar	103

实例 26: 改变 ProgressBar 和 SeekBar 的颜色	104
5.4.3 评分组件: RatingBar	105
5.5 其他重要组件	106
5.5.1 显示图像的组件: ImageView	106
实例 27: 可显示图像指定区域的 ImageView 组件	107
实例 28: 动态缩放和旋转图像	108
5.5.2 列表组件: ListView	109
实例 29: 可以单选和多选的 ListView	111
实例 30: 动态添加、删除 ListView 列表项	112
实例 31: 改变 ListView 列表项选中状态的背景颜色	115
5.5.3 封装 ListView 的 Activity: ListActivity	116
实例 32: 使用 SimpleAdapter 建立复杂的列表项	116
实例 33: 给应用程序评分	117
5.5.4 可展开的列表组件: ExpandableListView	120
5.5.5 下拉列表组件: Spinner	121
5.5.6 垂直滚动视图组件: ScrollView	123
5.5.7 水平滚动视图组件: HorizontalScrollView	123
实例 34: 可垂直和水平滚动的视图	124
5.5.8 网格视图组件: GridView	124
5.5.9 可循环显示和切换图像的组件: Gallery 和 ImageSwitcher	125
5.5.10 标签组件: TabHost	127
5.6 本章小结	128
第 6 章 移动存储解决方案	129
6.1 最简单的数据存储方式: SharedPreferences	129
6.1.1 使用 SharedPreferences 存取数据	129
6.1.2 数据的存储位置和格式	131
实例 35: 存取复杂类型的数据	131
6.1.3 设置数据文件的访问权限	133
6.1.4 可以保存设置的 Activity: PreferenceActivity	134
6.2 文件的存储	137
6.2.1 openFileOutput 和 openFileInput 方法	137
实例 36: SD 卡文件浏览器	138
实例 37: 存取 SD 卡中的图像	142

6.2.2 SAX 引擎读取 XML 文件的原理	143
实例 38: 将 XML 数据转换成 Java 对象	144
6.3 SQLite 数据库	147
6.3.1 SQLite 数据库管理工具	147
6.3.2 创建数据库和表	148
6.3.3 模糊查询	149
6.3.4 分页显示记录	149
6.3.5 事务	149
6.4 在 Android 中使用 SQLite 数据库	149
6.4.1 SQLiteOpenHelper 类与自动升级数据库	150
6.4.2 SimpleCursorAdapter 类与数据绑定	150
实例 39: 带照片的联系人管理系统	153
6.4.3 将数据库与应用程序一起发布	156
实例 40: 英文词典	156
6.5 抽象的标准——内容提供者 (ContentProvider)	159
实例 41: 将联系人管理和英文字典集成到自己的应用中	160
6.6 本章小结	165
第 7 章 应用程序之间的通讯	166
7.1 Intent 与 Activity	166
7.1.1 用 Intent 启动 Activity, 并在 Activity 之间传递数据	166
7.1.2 调用其他应用程序中的 Activity (拨打电话、浏览网页、发 Email 等)	169
7.1.3 定制自己的 Activity Action	172
实例 42: 将电子词典的查询功能共享成一个 Activity Action	174
7.2 接收和发送广播	175
7.2.1 接收系统广播	175
实例 43: 开机可自动运行的程序	176
实例 44: 收到短信了, 该做什么	176
实例 45: 显示手机电池的当前电量	178
7.2.2 在自己的应用程序中发送广播	179
实例 46: 接收联系人系统中发送的添加联系人广播	180
7.3 本章小结	181
第 8 章 Android 服务	182
8.1 Service 起步	182
8.1.1 Service 的生命周期	182
8.1.2 绑定 Activity 和 Service	185
8.1.3 在 BroadcastReceiver 中启动 Service	187

8.2 系统服务	188
8.2.1 获得系统服务	188
实例 47: 监听手机来电	189
实例 48: 来电黑名单	190
8.2.2 在模拟器上模拟重力感应	191
实例 49: 手机翻转静音	192
8.3 时间服务	194
8.3.1 计时器: Chronometer	194
8.3.2 预约时间 Handler	196
8.3.3 定时器 Timer	197
8.3.4 在线程中更新 GUI 组件	198
8.3.5 全局定时器 AlarmManager	200
实例 50: 定时更换壁纸	201
实例 51: 多次定时提醒	203
8.4 跨进程访问 (AIDL 服务)	205
8.4.1 什么是 AIDL 服务	206
8.4.2 建立 AIDL 服务的步骤	206
实例 52: 建立 AIDL 服务	206
实例 53: 传递复杂数据的 AIDL 服务	209
8.5 本章小结	213
第 9 章 网络	214
9.1 可装载网络数据的组件	214
实例 54: 将网络图像装载到 ListView 组件中	215
实例 55: Google 图像画廊 (Gallery)	217
9.2 WebView 组件	221
9.2.1 用 WebView 组件浏览网页	221
实例 56: 手机浏览器	221
9.2.2 用 WebView 组件装载 HTML 代码	222
实例 57: 将英文词典整合到 Web 页中 (JavaScript 调用 Java 方法)	224
9.3 访问 HTTP 资源	225

9.3.1 提交 HTTP GET 和 HTTP POST 请求	225
9.3.2 HttpURLConnection 类	227
实例 58: 上传文件	228
实例 59: 远程 Apk 安装器	231
9.3.3 调用 Webservice	232
实例 60: 通过 Webservice 查询产品信息	233
9.4 本章小结	236
第 10 章 多媒体	237
10.1 图形	237
10.1.1 图形绘制基础	237
实例 61: 绘制基本的图形和文本	239
10.1.2 绘制位图	243
实例 62: 用两种方式绘制位图	243
10.1.3 设置颜色的透明度	244
实例 63: 可任意改变透明度的位图	245
10.1.4 旋转图像	246
实例 64: 旋转动画	246
10.1.5 扭曲图像	247
实例 65: 按圆形轨迹扭曲图像	248
10.1.6 拉伸图像	251
实例 66: 拉伸图像演示	251
10.1.7 路径	253
实例 67: 沿着路径绘制文本	256
实例 68: 可在图像上绘制图形的画板	258
10.2 音频和视频	262
10.2.1 使用 MediaPlayer 播放 MP3 文件	262
10.2.2 使用 MediaRecorder 录音	263
10.2.3 使用 VideoView 播放视频	264
10.2.4 使用 SurfaceView 播放视频	264
10.3 本章小结	265

第三部分 进阶篇——深入 Android 世界的腹地

第 11 章 2D 动画	267
11.1 帧 (Frame) 动画	267
AnimationDrawable 与帧动画	267
实例 69: 通过帧动画方式播放 Gif 动画	269
实例 70: 播放帧动画的子集	271
11.2 补间 (Tween) 动画	273
11.2.1 移动补间动画	273
实例 71: 循环向右移动的 EditText 与上下弹跳的球	275

11.2.2 缩放补间动画	277
实例 72: 跳动的心	277
11.2.3 旋转补间动画	279
实例 73: 旋转的星系	279
11.2.4 透明度补间动画	280
实例 74: 投掷炸弹	281
11.2.5 振动效果	283
11.2.6 自定义动画渲染器 (Interceptor)	284
11.2.7 以动画方式切换 View 的组件	

ViewPager	285	第 14 章 访问 Android 手机的硬件	322
11.3 本章小结	287	14.1 在手机上测试硬件	322
第 12 章 OpenGL ES 编程	288	14.1.1 安装 Android USB 驱动	322
12.1 OpenGL 简介	288	14.1.2 在手机上测试程序	324
12.2 构建 OpenGL ES 的基本开发框架	289	14.1.3 在手机上调试程序	324
12.3 2D 图形绘制	290	14.2 录音	325
12.3.1 多边形	290	14.3 控制手机摄像头 (拍照)	326
12.3.2 颜色	292	14.3.1 调用系统的拍照功能	326
12.3.3 旋转	293	14.3.2 实现自己的拍照 Activity	328
12.4 3D 图形绘制	294	14.4 传感器在手机中的应用	332
12.4.1 旋转立方体	294	实例 76: 电子罗盘	333
12.4.2 在立方体上显示纹理	295	实例 77: 计步器	334
12.4.3 光照下的立方体	296	14.5 GPS 与地图定位	335
12.4.4 透明的立方体	298	14.5.1 Google 地图	336
12.5 本章小结	298	14.5.2 用 GPS 定位到当前位置	339
第 13 章 资源、国际化与自适应	299	14.6 WIFI	340
13.1 Android 中的资源	299	14.7 本章小结	342
13.1.1 Android 怎么存储资源	300	第 15 章 放在桌面上的小玩意	343
13.1.2 资源的种类	300	15.1 窗口小部件 (App Widget)	343
13.1.3 资源文件的命名	301	15.1.1 在 Android 桌面上添加 App Widget	343
13.2 定义和使用资源	301	15.1.2 开发 App Widget 的步骤	344
13.2.1 使用系统资源	301	实例 78: 数字时钟	346
13.2.2 字符串 (String) 资源	302	15.1.3 AppWidgetProvider 类	348
13.2.3 数组 (Array) 资源	303	15.1.4 向 App Widget 添加配置 Activity	349
13.2.4 颜色 (Color) 资源	305	实例 79: 可以选择风格的数字时钟	349
13.2.5 尺寸 (Dimension) 资源	306	15.2 快捷方式	353
13.2.6 类型 (Style) 资源	307	15.2.1 向快捷方式列表中添加快捷方式	353
13.2.7 主题 (Theme) 资源	308	15.2.2 直接将快捷方式放到桌面上	354
13.2.8 绘画 (Drawable) 资源	309	15.3 实时文件夹 (LiveFolder)	354
13.2.9 动画 (Animation) 资源	310	15.4 本章小结	356
13.2.10 菜单 (Menu) 资源	310	第 16 章 NDK 编程	357
13.2.11 布局 (Layout) 资源	313	16.1 Android NDK 简介	357
13.2.12 属性 (Attribute) 资源	313	16.2 安装、配置和测试 NDK 开发环境	358
实例 75: 改进可显示图标 of IconTextView 组件	315	16.2.1 系统和软件要求	358
13.2.13 XML 资源	316	16.2.2 下载和安装 Android NDK	358
13.2.14 RAW 资源	317	16.2.3 下载和安装 Cygwin	359
13.2.15 ASSETS 资源	317	16.2.4 配置 Android NDK 的开发环境	361
13.3 国际化和资源自适应	318	16.2.5 编译和运行 NDK 自带的例子	362
13.3.1 对资源进行国际化	318	16.3 Android NDK 开发	364
13.3.2 Locale 与国际化	320	16.3.1 JNI 接口设计	364
13.3.3 常用的资源配置	320	16.3.2 编写 Android NDK 程序的步骤	365
13.4 本章小结	321	实例 80: 将文件中的小写字母转换成	

大写字母 (NDK 版本)	365
16.3.3 配置 Android.mk 文件	367
16.3.4 Android NDK 定义的变量	368
16.3.5 Android NDK 定义的函数	369
16.3.6 描述模块的变量	369
16.3.7 配置 Application.mk 文件	370
16.4 本章小结	371
第 17 章 整合 Android 与脚本语言	372
17.1 Android 脚本环境简介	372
17.2 Android 脚本环境安装	373
17.3 编写和运行 Android 脚本	374
17.4 本章小结	376
第 18 章 Android 平台的新特性展示	377
18.1 Android 平台的新特性	377
18.1.1 Android 1.6 的新特性	377
18.1.2 Android 2.0 的新特性	379
18.2 Android Framework API 演变	379
18.2.1 Android 1.6 Framework API 的变化	379
18.2.2 Android 2.x Framework API 的变化	380
18.3 实时壁纸 (Live Wallpapers, Android 2.1)	380
18.4 补间动画渲染器 (Android 1.6)	387
18.4.1 Anticipate 渲染器	388
18.4.2 Overshoot 渲染器	388
18.4.3 Anticipate/Overshoot 渲染器	388
18.4.4 Bounce 渲染器	388
18.5 本章小结	389
第 19 章 另类的输入输出	390
19.1 手势 (Gesture)	390
19.1.1 创建手势文件	390
19.1.2 通过手势输入字符串	391
19.1.3 通过手势调用程序	393
19.1.4 编写自己的手势创建器	394
19.2 让手机说话 (TTS)	395
19.3 本章小结	396
第 20 章 蓝牙	397
20.1 蓝牙简介	397
20.2 打开和关闭蓝牙设备	398
20.3 搜索蓝牙设备	398
20.4 蓝牙 Socket	400
20.5 OBEX 服务器	402
20.6 本章小结	403

第四部分 OPhone 篇——进入 OPhone 世界

第 21 章 OPhone 入门	405
21.1 OPhone 平台概述	405
21.1.1 OPhone 的系统架构	405
21.1.2 JIL Widget 的运行环境	406
21.1.3 OPhone 应用程序展示	406
21.2 OPhone 开发环境搭建	407
21.2.1 安装 ODT 和 WDT	407
21.2.2 测试一下 ODT 是否安装成功	407
21.2.3 测试一下 WDT 是否安装成功	407
21.3 在真机上调试程序	408
21.4 本章小结	410
第 22 章 OPhone 的 API 扩展	411
22.1 视频电话	411
22.1.1 拨打视频电话	411
22.1.2 挂断视频电话	411
实例 81: 可拨打视频电话的程序	412
22.2 搜索 API	413
22.3 本章小结	414
第 23 章 JIL Widget 开发详解	415
23.1 编写第一个 JIL Widget 程序	415
23.2 多媒体	417
23.2.1 播放音频	417
23.2.2 播放视频	418
23.3 操作文件	418
23.3.1 获得与文件相关的信息	419
23.3.2 操作文件	419
23.3.3 操作文件夹	421
23.4 获得系统信息	422
23.4.1 获得电池信息	422
23.4.2 获得手机信号信息	423
23.4.3 获得设备信息	424
23.4.4 获得设备状态信息	424
23.5 控制硬件	425
23.5.1 拍照和摄像	426
23.5.2 打电话	426
23.5.3 手机振动	426
23.6 本章小结	427

第五部分 综合实例篇——实践是检验真理的唯一标准

第 24 章 万年历.....	429	24.3.5 在广播接收器中显示提醒界面	449
24.1 主界面设计与实现.....	429	24.3.6 访问数据库: DBService	450
24.1.1 万年历的核心类: Calendar	429	24.4 本章小结.....	454
24.1.2 日历内容元素的基类: CalendarParent	431	第 25 章 知道当前位置的 Google GTalk 机器人	455
24.1.3 绘制万年历边框: Border 类	431	25.1 GTalk 的通讯协议和技术.....	455
24.1.4 绘制周名称: Week 类	432	25.1.1 Jabber 和 XMPP	455
24.1.5 绘制日期和网格: Grid 类	433	25.1.2 XMPP 客户端库: Smack 和 Asmack	456
24.1.6 日历视图: CalendarView 类	440	25.2 登录 GTalk: Login 类	456
24.1.7 生成万年历的主界面	442	25.3 联系人信息	458
24.2 选项菜单功能	443	25.3.1 显示联系人列表	458
24.2.1 切换到今天的日期	443	25.3.2 监听联系人是否上线	460
24.2.2 指定日期	443	25.4 联系人之间的通讯	461
24.3 可以写日记和提醒的万年历	444	25.4.1 发送聊天信息	462
24.3.1 显示所有的记录信息: AllRecord 类	444	25.4.2 接收聊天信息	462
24.3.2 添加和修改记录: Record 类	446	25.4.3 自动回复当前的位置 (GPS 定位)	465
24.3.3 设置提醒时间	447	25.5 本章小结	466
24.3.4 启动服务	448		



Android 入门

Google 于 2005 年并购了成立仅 22 个月的高科技企业 Android，展开了短信、手机检索、定位等业务，同时基于 Linux 的 Android 平台也进入了开发阶段。Google 在 2007 年 11 月 5 日发布了 Android 的第一个版本。在刚发布之初，Android 并没有引起业界太多的关注。但随着 Google 组建的开放手机联盟不断有新力量加入，Android 这个初出茅庐的小子已成为与 iPhone 分庭抗礼的生力军。

在作者写作本书时，至少有数十家不同规模的手机厂商宣布加入 Android 阵营。基于 Android 的手机也是琳琅满目。现在让我们进入时空隧道，回到 2008 年 9 月 23 日（北京时间 2008 年 9 月 23 日 22:30）的美国纽约，Google 和运营商 T-Mobile 共同发布了世界上第一款安装 Android 系统的手机 T-Mobile G1。由于这款手机的出色表现，使 Android 真正成为了万众瞩目的焦点。正是因为 Android 和其他几项创新，在 17 个月后的 2010 年 2 月 25 日，美国著名商业杂志《Fast Company》评选的 2010 年全球最具创新力公司 50 强中，Google 位列移动领域十大最具创新力公司榜首。



本章内容

- Android 的系统构架
- 搭建 Android 开发环境
- Android SDK 中的常用命令行工具（包括 adb、android 和 mkcdsard）
- 可以在 PC 上运行的 Android 系统（用于在没有真机的情况下测试程序）
- 应用程序商店

1.1 Android 的基本概念

Android 的中文意思是“机器人”。但在移动领域，大家一定会将 Android 与 Google 联系起来。Android 本身就是一个操作系统，只是这个操作系统是基于 Linux 内核的。也就是说，从理论上，基于 Linux 的软件移植到 Android 上是最容易的。Android 是一个由 30 多家科技公司和手机公司组成的“开放手机联盟”共同研发的，而且完全免费开源，这将大大降低新型手机设备的研发成本，甚至 Android 已成了“山寨”机的首选。

1.1.1 Android 简介

Android 作为 Google 最具创新的产品之一，正受到越来越多的手机厂商、软件厂商、运营商及个人开发者的追捧。目前 Android 阵营主要包括 HTC（宏达电）、T-Mobile、高通、三星、LG、摩托罗拉、ARM、

软银移动、中国移动、华为等。虽然这些机构有着不同的性质，但它们都在 Android 平台的基础上不断创新，让用户体验到最优质的服务。尤其要提一下的是中国移动与播思公司联合研发的基于 Android 的 OMS 系统已取得了不俗的业绩(关于 OMS 和 OPhone 将在第 4 篇详细介绍)。下面欣赏几款具有代表性的 Android 手机。第一款毫无疑问，就是世界上第一部 Android 手机 T-Mobile G1，如图 1.1 所示。这款手机带有一个物理键盘(硬键盘)，可以通过侧划拉出。第二款是创下了销售奇迹的 HTC Hero，也称为 G3，如图 1.2 所示。这款手机的显著特征是下方有一个突起的小“下巴”。除此之外，HTC Hero 绚丽的 Sense 界面也成为 Android 手机中一道亮丽的风景。最后一款创下了配置之最，这就是 2009 年底发布的几款拥有 1GHz CPU 的手机之一：SonyEricsson X10，如图 1.3 所示。

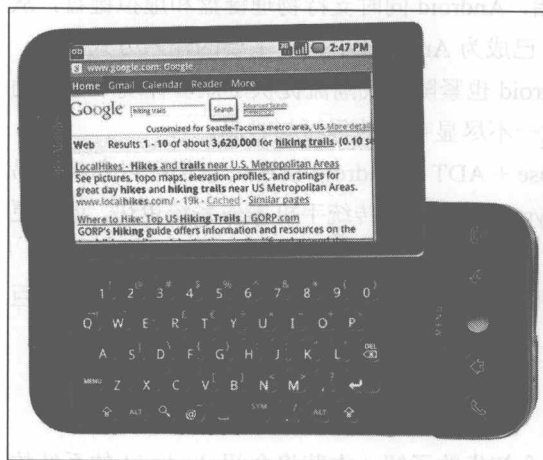


图 1.1 T-Mobile G1



图 1.2 HTC Hero



图 1.3 拥有 1GHz CPU 的 SonyEricsson X10

欣赏完这么多“超酷”的手机，现在来看一下 Android 到底有什么魔力，可以让众多的粉丝为之疯狂。据粗略统计，Android 至少有如下 8 件制胜法宝：

- 开放性。Android 平台是免费、开源的。而且 Google 通过与运营商、设备制造商、开发商等机构形成的战略联盟，希望通过共同制定标准使 Android 成为一个开放式的生态系统。
- 应用程序的权限由开发人员决定。编写过 Symbian、Java ME 程序的读者应该能体会到这些程序在发布时有多麻烦。如果访问到某些限制级的 API，不是出现各种各样的提示，就是根本无法运行。要想取消这些限制，就得向第三方的认证机构购买签名，而且价格不菲。而 Android 平台的应用程序就幸福得多。要使用限制级的 API，只需要在自己的应用程序中配置一下即可，完全是 DIY。这也在某种程度上降低了 Android 程序的开发成本。
- 我的平台我作主。Android 上的所有应用程序都是可替换和扩展的，即使是拨号、Home 这样的

核心组件也是一样。只要有足够的想象力，就可以缔造出一个独一无二、完全属于自己的 Android 世界。

- 应用程序之间的无障碍沟通。应用程序之间的通信一直令人头痛，而在 Android 平台上无疑是一种享受。在 Android 平台上应用程序之间至少有 4 种沟通方式。很难说哪一种方式更好，但它们的确托起了整个 Android 的应用程序框架。
- 拥抱 Web 的时代。如果想在 Android 应用程序中嵌入 HTML、JavaScript，那真是再容易不过了。基于 Webkit 内核的 WebView 组件会完成一切。更值得一提的是，JavaScript 还可以和 Java 无缝地整合在一起（见第 9 章的实例 57 中的介绍）。
- 物理键盘和虚拟键盘双管齐下。从 Android 1.5 开始，Android 同时支持物理键盘和虚拟键盘，从而可大大丰富用户的输入选择。尤其是虚拟键盘，已成为 Android 手机中主要的输入方式。
- 个性的充分体现。21 世纪是崇尚个性的时代。Android 也紧随时代潮流提供了众多体现个性的功能。例如，Widget、Shortcut、Live WallPapers，无一不尽显手机的华丽与时尚。
- 舒适的开发环境。Android 的主流开发环境是 Eclipse + ADT + Android SDK。它们可以非常容易地集成到一起，而且在开发环境中运行程序要比 Symbian 这样的传统手机操作系统更快，调试更方便。

虽然 Android 的特点还有很多，但这已经不重要。重要的是现在 Android 已经成为万众瞩目的国际巨星，她的未来将令人充满期望。

1.1.2 Android 的系统构架

通过上一节的介绍，我们对 Android 的特点已经有了一个初步的了解。本节将介绍 Android 的系统构架。先来看看 Android 的体系结构，如图 1.4 所示。

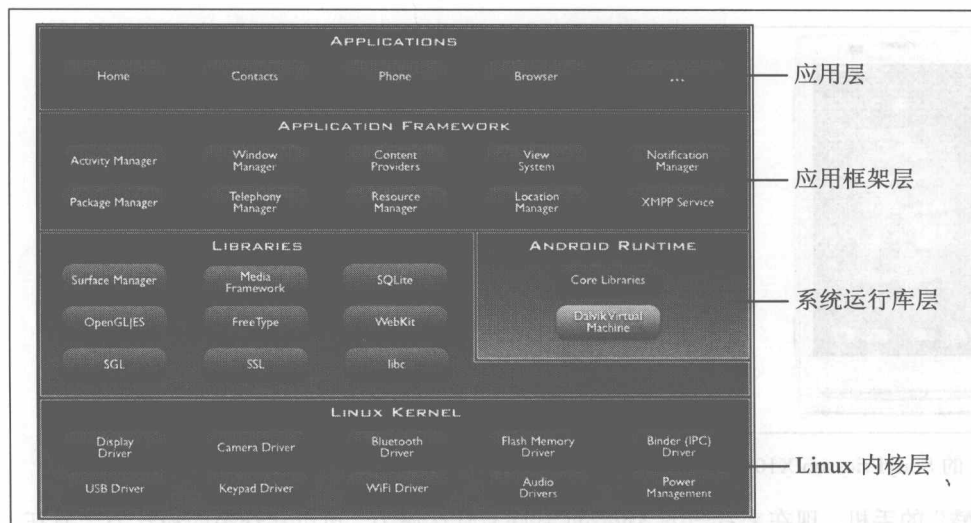


图 1.4 Android 的体系结构

从图 1.4 可以看出 Android 分为 4 层，从高到低分别是应用层、应用框架层、系统运行库层和 Linux 内核层。下面将对这 4 层进行简单介绍。

- 应用层。该层由运行在 Dalvik 虚拟机（为 Android 专门设计的基于寄存器的 Java 虚拟机，运行 Java 程序的速度更快）上的应用程序（主要由 Java 语言编写）组成。例如，日历、地图、浏览器、联系人管理，都属于应用层上的程序。
- 应用框架层。该层主要由 View、通知管理器(Notification Manager)、活动管理器(Activity Manager)等由开发人员直接调用的组件组成。
- 系统运行库层。Java 本身是不能直接访问硬件的。要想让 Java 访问硬件，必须使用 NDK 才可以。

NDK 是一些由 C/C++ 语言编写的库。这些程序也是该层的主要组成部分。该层主要包括 C 语言标准库、多媒体库、OpenGL ES、SQLite、Webkit、Dalvik 虚拟机等。也就是说，该层是对应用框架层提供支持的层。

- Linux 内核层。该层主要包括驱动、内存管理、进程管理、网络协议栈等组件。目前 Android 的版本基于 Linux 2.6 内核。

1.2 Android 开发环境的搭建

工欲善其事，必先利其器。开发 Android 应用程序总不能直接用记事本开发吧（那些超级大牛除外）。找到合适的开发工具是学习 Android 开发的第一步。而更多地了解 Android 的开发环境将会对进一步学习 Android 保驾护航。

1.2.1 开发 Android 程序需要些什么

开发 Android 程序至少需要如下工具和开发包：

- JDK（建议安装 JDK1.6 及其以上版本）
- Eclipse
- Android SDK
- ADT（Android Development Tools，开发 Android 程序的 Eclipse 插件）

其中 JDK 的安装非常简单，读者可以在官方网站下载 JDK 的最新版，并按着提示进行安装。Eclipse 下载后直接解压即可运行。在 1.2.2 节和 1.2.3 节将介绍 Android SDK 和 ADT 的安装。

1.2.2 安装 Android SDK

读者可以从下面的两个地址下载 Android SDK 的最新版本：

地址 1

<http://developer.android.com/intl/zh-CN/sdk/index.html>

地址 2

<http://androidappdocs.appspot.com/sdk/index.html>

该版本可以同时安装 6 个 Android SDK 版本（1.1 至 2.1）。要注意，Android SDK 是在线安装。在安装 Android SDK 之前，要保证有稳定而快速的 Internet 连接。如果完全安装 Android SDK，安装时间会比较长，读者需要耐心等待。

如果安装 Android SDK 时下载文件失败，可以将如图 1.5 所示的安装界面右下角的第 1 个复选框选中，然后重新安装 Android SDK。如果安装过程顺利，将会出现如图 1.6 所示的下载界面。

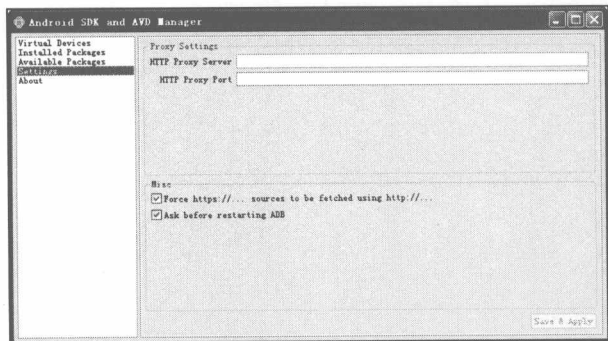


图 1.5 使用 http 下载文件

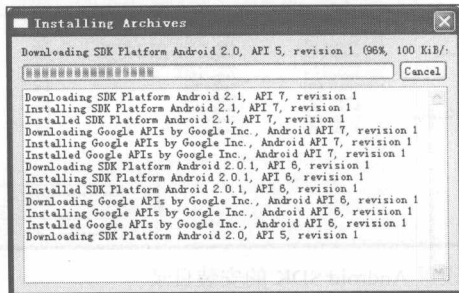


图 1.6 安装过程的下载界面

Android SDK 安装成功后，会看到如图 1.7 所示的 Android SDK 根目录结构。platforms 目录包含当前 Android SDK 支持的所有版本，如图 1.8 所示。