

Android

程序设计

柯元旦 宋 锐 编著



北京航空航天大学出版社

Android 程序设计

柯元旦 宋 锐 编著

北京航空航天大学出版社

内 容 简 介

本书介绍基于 Android 操作系统的应用程序开发,作者力求从一般程序概念的角度去解析 Android 中的一些新概念,从而让程序员能够快速将已有的程序经验转移到 Android 平台上来。本书内容可分为 3 部分,分别为 Java 基础、Android 系统组件介绍以及开发工具的使用。书中包含了大量的基础性实例,用于阐述一些基础组件的使用方法,这些实例大部分基于 Android 1.5r2 版本,并根据 Android 1.6 版本的更新做了一定的调整和说明。

本书除了作为 Android 应用程序开发者的入门读物外,也可以作为 Android 工程人员的基础设计参考书。同时,本书中的游戏开发章节包含了一般的多线程、多角色游戏架构设计,也适合有意愿开发基于 Android 游戏类应用的爱好者。

图书在版编目(CIP)数据

Android 程序设计/柯元旦,宋锐编著. --北京:
北京航空航天大学出版社,2010.7
ISBN 978-7-5124-0113-6

I. ①A… II. ①柯… ②宋… III. ①移动通信—携带
电话机—应用程序—程序设计 IV. ①TN929.53

中国版本图书馆 CIP 数据核字(2010)第 106696 号

版权所有,侵权必究。

Android 程序设计

柯元旦 宋 锐 编著

责任编辑 刘 星

*

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

北京市媛明印刷厂印装 各地书店经销

*

开本:787×960 1/16 印张:21.5 字数:482 千字

2010 年 7 月第 1 版 2010 年 7 月第 1 次印刷 印数:5 000 册

ISBN 978-7-5124-0113-6 定价:45.00 元(含光盘 1 张)

前言

当说到 Android,很多人的概念是——Google 推出的一套开源操作系统,也有人会想到 Google 手机,也有人说是一个开源联盟。

这些都没有错,Android 本是一家小公司的名字,由这家公司的几位创办人的名字组合而成的。该公司主要做基于移动类电子设备的操作系统,与其他嵌入式操作系统不同,该操作系统是基于 Java 虚拟机的,后来 Google 收购了 Android,Android 的核心开发人员也就加入了 Google,直到现在。

接着,Google 发起了一个 Open Handset Alliance,成员包括芯片制造商、设备生产商、网络运营商以及应用开发商,旨在让人们更廉价、更方便、更有趣地进行信息沟通。所有这些会员开发的产品都会遵循约定的技术标准,主要是以 Android 操作系统为实现载体。芯片制造商需要更好地兼容 Android 操作系统,网络运营商可以在 Android 的基础上定义自己所需要的服务,应用开发商也基于 Android 开发程序。

操作系统并不算是个新的概念,Android 尽管有独特的概念或者定义,但所实现的功能(或者目标)与大多数操作系统相同。因此,本书在每章开始时,会提出一些和应用目标相关的问题;然后再具体介绍所涉及的操作系统相关的技术细节,当读者理解了技术细节后,基本上就可以回答这些问题了;最后,本书会对所提出的问题做一个简要的回答,帮助读者形成一个更清晰的概念。

本书第 1、2 章介绍了 Android 操作系统的整体框架组成。

Android 操作系统是使用 Java 语言编写的,因此第 3 章介绍了一些 Android 中常用的 Java 语法。

对于任何一个具有图形接口的操作系统来讲,GUI 操作原理、存储系统、多线程设计都是重点,这些分别在本书的第 4、11、12 章进行讲解。除此之外,Android 应用程序可以分片段运行,这与其他系统不一样,本书第 5~9 章介绍了这些程序片段的原理与使用。

Android 主要用于消费类电子产品,因此,多媒体处理和游戏中应用很普遍,第 15 章介绍了多媒体处理、网络通信、游戏开发等基本原理与使用。

Android 亦是面向个人信息的电子产品,因此,其安全性特别重要。Android 采取了多种机制保证个人信息安全,第 13 章介绍了 Android 的安全机制。

编写 Android 应用程序时,除了使用 Java 语言外,还可以使用 xml 语法编写一些辅助类文件。这些文件由相应的编译工具编译,以方便程序的开发。这些辅助类文件的编写在第 10、14 章介绍。

程序开发需要好的调试环境和工具,Android 提供了丰富的调试工具,在第 16 章中进行介绍。

本书读者对象要求具有一定的程序设计基础,熟练使用 C 语言或者 Java 语言的任何一种。

感谢徐琳、王国锋、舒伟、黄景楠、赵晓睿、毛世杰等同志,他们对此书进行了校阅;同时感谢我的妻子给予的关怀,使我在整个写作过程中能够专心致志,心无杂念。

由于时间与水平所限,本书难免会有错误,望谅解,并与作者联系指出错误,本人邮箱:yuandanke@gmail.com;也可以发送电子邮件到:emsbook@gmail.com,与本书策划编辑进行交流。

柯元旦

2010 年 3 月

目 录

第 1 章 Android 架构	1	3.7 synchronized 同步关键字	27
本章问题	1	3.8 new 的使用	29
1.1 Android 系统框架	1	3.9 字节流与字符串之间的编码 转换	30
1.2 Android 开发的分类	4	3.10 系统时间与设备时间	31
1.3 Android 应用程序开发环境的 建立	5	本章问题答案	31
本章问题答案	7	第 4 章 图形用户接口	33
第 2 章 Android 应用程序的组成	8	本章问题	33
本章问题	8	4.1 布 局	33
2.1 建立一个模拟器	8	4.1.1 使用 xml 创建一个布局 ...	34
2.2 Hello Android 程序	10	4.1.2 视图的大小与位置	41
2.3 Android 程序基础	12	4.1.3 动态创建 View 和 ViewGroup	47
2.4 源文件	15	4.1.4 常用 ViewGroup(layout)	53
2.5 资源文件	15	4.1.5 网页视图	60
2.6 自动生成的文件	16	4.2 添加用户事件响应	62
2.7 AndroidManifest.xml 文件	17	4.2.1 用户消息派发流程	62
本章问题答案	18	4.2.2 添加回调函数	64
第 3 章 Java 语法在 Android 中的使用 ...	20	4.3 菜 单	66
本章问题	20	4.3.1 菜单概况	66
3.1 interface 的使用	20	4.3.2 创建 Option Menu	67
3.2 abstract class 的使用	22	4.3.3 创建 Context Menu	71
3.3 interface 与 abstract class 的 区别	22	4.3.4 子菜单	76
3.4 for 循环的使用	24	4.3.5 单选与复选菜单	77
3.5 Map 类的使用	25	4.3.6 在 xml 文件中描述菜单 ...	78
3.6 int 与 String 之间的转换	26	4.4 对话框	80

4.4.1 对话框概况	80	6.2.1 直接启动 Activity	155
4.4.2 提示对话框	81	6.2.2 通过菜单启动 Activity ...	156
4.4.3 进度对话框	83	6.3 使用小工具	157
4.4.4 日期、时间对话框	85	6.3.1 Widget 原理	158
4.4.5 自定义对话框	86	6.3.2 创建 Widget	159
4.5 绑定视图和数据	89	6.3.3 在 Widget 中启动 Activity ...	161
4.5.1 Adapter 与 Adapter View ...	89	6.3.4 使用 Widget 进行信息提醒	162
4.5.2 下拉列表框	91	6.4 使用 Notification	164
4.5.3 带自动完成的下拉列表框 ...	94	6.4.1 创建 Notification	164
4.5.4 格子视图	95	6.4.2 在通知栏中启动 Activity ...	167
4.5.5 Gallery	100	6.5 保存活动状态	168
4.5.6 标签页	103	6.6 Activity 之间的数据传送	169
4.5.7 列表视图	106	6.7 重新定义 Back 键——Activity 与 Task	172
4.6 自定义 View	111	本章问题答案	174
4.6.1 View 的工作原理	112	第 7 章 使用 Service	176
4.6.2 编写 View 类	113	本章问题	176
4.6.3 为 View 类增加属性	115	7.1 创建 Service	176
4.6.4 绘制屏幕	119	7.2 在 Activity 中启动 Service	179
4.6.5 响应用户消息	134	7.3 使用 Widget 启动 Service	180
4.6.6 自定义回调函数	136	本章问题答案	183
本章问题答案	139	第 8 章 使用 Broadcast Receiver	185
第 5 章 Intent 与 Intent Filter	141	本章问题	185
本章问题	141	8.1 Receiver 概况	185
5.1 概 况	142	8.2 静态创建 Receiver	187
5.2 Intent 数据结构	143	8.3 动态创建 Receiver	188
5.3 响应 Intent——Intent 过滤器 ...	145	本章问题答案	190
5.3.1 Action 测试	146	第 9 章 使用 Content Provider	192
5.3.2 Category 测试	148	本章问题	192
5.3.3 Data 测试	149	9.1 Provider 基础	193
本章问题答案	151	9.2 读取 Provider	194
第 6 章 使用 Activity	152	9.3 修改 Provider	197
本章问题	152	9.3.1 添加数据项	197
6.1 Activity 概况	152		
6.2 启动 Activity	155		

9.3.2 更改数据项	201	11.2.2 读/写文件	230
9.3.3 删除数据项	203	11.3 数据库存储	232
9.4 创建 Provider	203	11.4 程序参数存储	236
9.4.1 Provider 接口函数定义 ..	204	11.4.1 写入参数值	237
9.4.2 实现一个 Provider	204	11.4.2 读取参数值	237
9.4.3 使用自定义 Provider	206	本章问题答案	239
9.4.4 调试自定义 Provider	208	第 12 章 多线程设计	240
9.5 Provider 中 Uri 格式总结	209	本章问题	240
本章问题答案	209	12.1 多线程引述	241
第 10 章 资源文件	211	12.2 线程定义	243
本章问题	211	12.2.1 Thread	243
10.1 资源文件概述	211	12.2.2 Runnable	246
10.2 资源类型	212	12.2.3 Thread 与 Service	248
10.2.1 简单数值类资源	212	12.3 使用 Handler	249
10.2.2 “可绘制”类资源	214	12.3.1 完成定时任务	249
10.2.3 动画	216	12.3.2 在线程之间传递数据	252
10.2.4 菜单	216	12.4 使用 Looper	255
10.2.5 布局	216	12.5 使用 HandlerThread	259
10.2.6 自定义视图	217	12.6 线程间的同步	259
10.2.7 元数据文件资源	218	12.6.1 wait()/notify() 的使用	259
10.2.8 样式	218	12.6.2 join() 的使用	262
10.2.9 主题	219	本章问题答案	266
10.3 assets 管理器	220	第 13 章 系统安全	267
10.4 国际化	222	本章问题	267
本章问题答案	224	13.1 系统安全概述	267
第 11 章 Android 本地存储系统	225	13.2 程序签名	268
本章问题	225	13.2.1 创建签名	269
11.1 Android 内部文件夹结构	225	13.2.2 执行签名	270
11.1.1 系统文件夹	226	13.3 安全调用	270
11.1.2 文件访问权限	227	13.3.1 系统功能调用	271
11.1.3 文件路径	228	13.3.2 启动 Activity	272
11.1.4 程序私有文件	228	13.3.3 Broadcast Receiver	
11.2 文件存储	229	发送与接收	274
11.2.1 遍历文件夹	229	13.3.4 读/写 Content Provider ..	274

13.3.5 启动 Service	274	15.3.2 Socket 编程	304
本章问题答案	275	15.3.3 Http 编程	307
第 14 章 AndroidManifest.xml	276	15.4 通信开发	309
本章问题	276	15.4.1 发送短信	309
14.1 文件结构	277	15.4.2 接收短信	313
14.2 元素定义规范	283	本章问题答案	315
14.2.1 声明元素	283	第 16 章 Android 常用调试工具的使用	316
14.2.2 指定元素属性	284	本章问题	316
14.2.3 指定类名称	284	16.1 调试环境组成	316
14.2.4 多值定义	284	16.2 安装设备驱动	318
14.2.5 资源引用	285	16.3 Emulator	320
14.2.6 字符串值	285	16.4 ddms	321
14.2.7 图标与标签	285	16.5 adb	321
本章问题答案	286	16.6 Logcat	321
第 15 章 多媒体与网络开发基础	287	16.7 draw9patch	322
本章问题	287	16.8 为工程添加 jar 包	324
15.1 媒体播放	287	本章问题答案	327
15.1.1 媒体播放原理	287	附录 A 光盘文件结构	329
15.1.2 播放音乐	289	附录 B 相关资源信息	330
15.1.3 播放视频	291	B.1 获取应用程序源码	330
15.1.4 录音	295	B.2 获取 Android 内核源码	331
15.2 Surface 的使用	297	附录 C 综合例程	333
15.2.1 Surface 原理	298	C.1 生日与性格	333
15.2.2 游戏开发	299	C.2 俄罗斯方块	333
15.2.3 游戏中的多线程	301	C.3 星际飞船	334
15.3 网络服务开发	304	参考文献	336
15.3.1 网络服务概况	304		

第 1 章

Android 架构

本章问题

1. Android 操作系统都包含哪些部分?
2. Google 都实现了操作系统的哪些部分?
3. 对应用程序开发者而言,需要了解的部分都包含哪些?

1.1 Android 系统框架

Android 操作系统是一个开源的软件平台,基于 Linux Kernel,定义并实现了一套系统服务框架和图形用户接口。

Android 操作系统组成如图 1-1 所示。系统组成包含以下几个部分。

第 1 部分:Linux Kernel

Android 目前是基于 Linux 2.6 版本,选择 Linux Kernel 的主要原因是:

- ① Linux 有着良好的驱动模型以及已存在的、广泛的、运行良好的各种硬件驱动。
- ② 可靠的内存管理模块、安全机制、网络以及良好的操作系统内核框架。
- ③ Linux 在越来越多设备上的使用,证明了 Linux 的可靠性。

第 2 部分:Libraries

Libraries 采用 C/C++ 语言编写,主要为 Android GUI、Android 系统服务提供一些底层实现。Applications 不能直接访问 Libraries,而只能通过 Application Framework 访问。

Surface Manager:Surface 是 Android 定义的一种视图,子程序可以通过该视图直接在屏幕上绘图,而不需要 GUI 的干涉,该模块用于管理 Surface 视图。

OpenGL|ES, SGL:提供了一套 OpenGL(3D)、SGL(2D)绘图函数库。

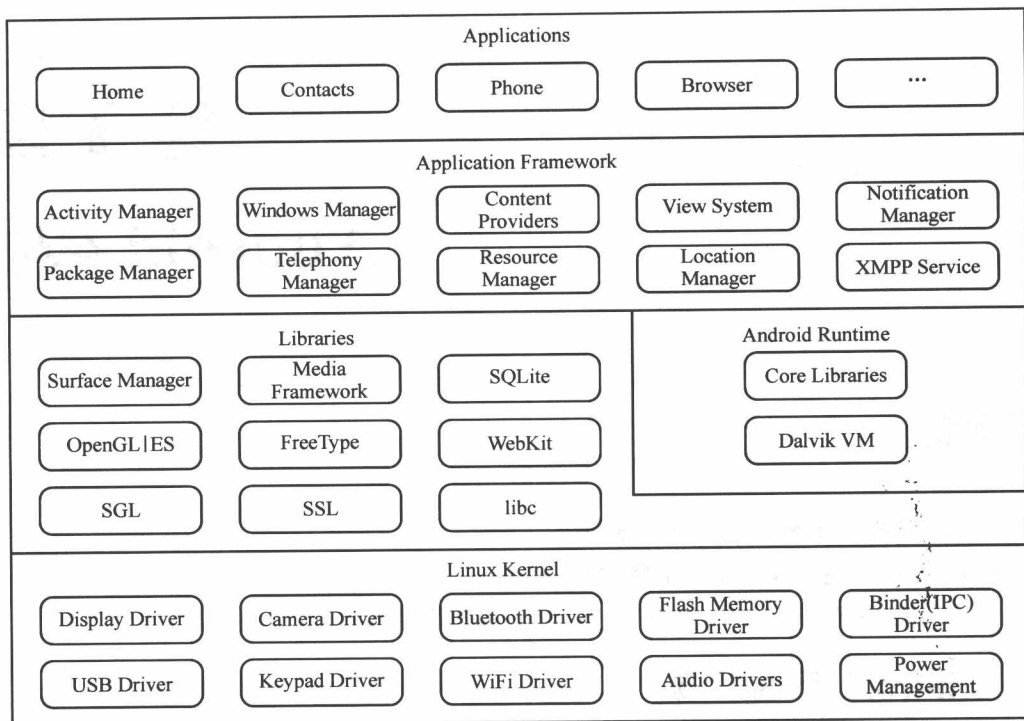


图 1-1 Android 操作系统组成

Media Framework: 一套音视频处理库, 包括 MPEG4、H. 263、MP3、AAC、AMR、JPG、PNG 等。

FreeType: 位图以及向量字体显示。

SSL: SSL 协议库。

SQLite: Android 系统提供了一个 SQLite 数据库引擎。

WebKit: 一个开源的浏览器引擎。

libc: 标准 C 库。

第 3 部分: Android Runtime

Android 应用程序是用 Java 语言编写的, Google 开发了一个自己的 Java 虚拟机。和 Sun 公司的 Java 虚拟机相比, Android 的虚拟机采用的是 Dalvik 虚拟机, 该虚拟机对执行嵌入式 Java 程序进行优化, 主要针对有限的内存、有限的 CPU 处理能力以及能耗方面。

Core Libraries: 提供了 Dalvik 虚拟机运行时所需要的一些核心 Java 类库。

Dalvik VM: 能够执行 Java 字节码的嵌入式 Java 虚拟机。Dalvik VM 是整个 Android 操作系统中唯一没有开源的部分。

第4部分:Application Framework

该部分全部使用 Java 语言编写,实现了 Android 的系统框架和 GUI。

Activity Manager: Activity 是 Android 应用程序的一个最小执行单元,该模块用于管理 Activity 的生命期以及相关的行为。

Windows Manager: Android 应用程序包含很多 Windows, 该模块用于管理 Windows。

Package Manager: 每一个 Android 应用程序都是一个 Package, 该模块用于管理应用程序的安装以及卸载。

Telephony Manager: 手机一般都是由一个应用处理器加一个调制解调器(Moderm)组成,应用处理器用于界面操作和内容管理,而调制解调器用于打电话、发送短信等通信工作。该模块用于通信管理,包括收发短信、接打电话、GPRS 上网等。

Content Providers: Android 系统中,每个应用程序都运行在不同的 Linux User ID 中,因此应用程序之间彼此不能看见对方拥有的文件,该模块用于在不同的应用程序间传递数据。

View System: 一个编辑框、按钮等都是一个 View, 该模块就是 Android GUI 的核心模块。

Resource Manager: Android 应用程序可以通过一种叫做 Resource 的类型访问诸如字符串、位图等资源,该模块用于管理此类资源。

Location Manager: GPS 等定位设备的服务模块,为应用程序提供位置服务。

Notification Manager: Notification 用于事件通知,比如有新短信,该模块用于管理各种事件通知。

XMPP Service: XMPP 是一种应用层的通信协议, Gtalk 就是基于 XMPP 协议设计的, XMPP 是一种标准协议,可用于即时通信、RSS 等多种 Web 服务。该模块提供了 XMPP 协议的底层实现,应用程序可以调用该服务模块。

第5部分:Applications

Application 是千变万化的,Android 系统包含了一些常用的应用程序,其中包括:

Email 邮件收发;

SMS Program 短信收发;

Calendar 日历;

Maps 地图服务;

Browser 浏览器;

Contacts 电话簿。

Android 所提供的这些典型应用程序和设计者开发的程序是平等的,用户完全可以像卸载普通程序一样卸载掉短信收发软件、打电话软件等。这在很多其他手机上是不可以的,因为这些程序拥有“系统程序”的等级;而在 Android 上,大家都一样,用户甚至可以重新定义桌面程序。

1.2 Android 开发的分类

从 Android 系统组成的角度来讲,Android 技术开发一般包含 5 个层面。

第 1 层面:Linux Kernel 开发

主要工作是将 Linux 移植到不同的硬件平台上,并提供相应的硬件驱动。对于大多数硬件设备而言,比如 USB 接口、网卡、声卡等设备,都有现成的驱动;但对于一些自定义的硬件设备,则需要开发自定义的驱动,同时对于这些专有设备的使用,还必须提供一套 Java 应用程序可以使用的接口,也就是说,需要与 Libraries 开发人员协同工作。

第 2 层面:Libraries 开发

Libraries 为 Application Framework 提供服务,主要是调用 Linux 驱动完成一些硬件操作,并包含一些高效的 C/C++ 函数库。

如前所述,如果给一个 Android 设备增加了一个硬件模块,那么除了要写一个该硬件模块的 Linux 驱动外,还得用 C/C++ 写一个 Libraries 来调用驱动,再用 Java 写一个 Application Framework 来调用该 Libraries,最终,Java 应用程序才能使用该硬件设备。

第 3 层面:Dalvik VM

由于 Google 并没有对 Dalvik VM 开源,所以,一般不能直接去开发 Dalvik,也很少需要去开发 Dalvik。

第 4 层面:Application Framework 的开发

这部分内容主要由 Google 开发,Google 已经定义了 GUI 和系统服务的架构,因此,对于这部分更多的开发是扩展系统服务,而不是重写已有的 GUI 或者系统服务。

第 5 层面:应用程序开发

该部分内容包括如何创建一个窗口,如何给窗口中添加一些文字、按钮等。根据用户的输入完成不同的功能,比如播放音乐、存储数据、启动不同的程序、发送短信、拨打电话等,这些都是应用程序干的事情。本书所涵盖的内容仅在应用程序开发。

“Android 操作系统”这个叫法往往有点含糊,从 Android 整个系统组成来讲,它包含了 Linux Kernel 和 Dalvik 虚拟机部分;而从应用程序开发的角度来讲,程序员关心最多的还是 Google 所定义的 GUI 和系统服务框架部分。因此,为了更明确表述,本书以后所称的“Android 操作系统”或者“Android 系统”均指 GUI 和 Android 系统服务部分,而不包括 Dalvik 虚拟机和 Linux Kernel。更确切的讲,本书只关心 Application Framework 部分,当然,我们并不是关心它内部是如何实现的,而是关心它给 Application 提供了哪些接口,定义了哪些概念和规则。

1.3 Android 应用程序开发环境的建立

Android 应用程序是在 Eclipse 下开发的, Google 提供了一个 Eclipse 的插件和一个 Android SDK。

Android SDK 包括一个 Android 手机模拟器、调试工具以及 Android 系统的库文件。

手机模拟器如图 1-2 所示, 该模拟器在 PC 上运行, 在调试程序时, 它与真实设备几乎一样, 虽然不能进行通信、照相等需要真实物理设备支持的功能, 但是却可以使用 Android 提供的调试工具模拟这些真实设备。



图 1-2 Android 手机模拟器

调试工具主要包括单步跟踪和实时输出, 类似于 C 语言中的 `printf()`, 在 Android 中则使用 `Log()` 函数。

库文件主要指 Android 架构中的 Libraries、Application Framework 以及 Dalvik VM。

在 Windows 下建立 Android 开发环境, 请按以下步骤进行。对于 Linux 和 MacOS, 请参照 Android 官方网站向导。

① 下载 Eclipse, 网址为: <http://www.eclipse.org/downloads>, 选择 Eclipse Classic 类别。Eclipse 是不需要安装的, 直接解压后即可运行。

② 下载并安装 JDK5 或者 JDK6, 网址为: <http://java.sun.com/javase/downloads/index.jsp>。

③ 下载 Android SDK,网址为:http://developer.android.com/sdk/1.5_r2/index.html。SDK 也不需要安装,解压即可使用。到目前为止,SDK 已经正式发行了 3 个版本,分别为 1.1、1.5r1 和 1.5r2。这里使用最新版本 1.5r2,尽管如此,该版本中仍未包含一些新的功能。要获得最新的 Android 平台资源,需要去 <https://sites.google.com/a/android.com/opensource/download> 下载最新的源码并编译,但这是以后的事情,1.5r2 版本对于了解 Android 应用程序设计已足够了。

④ 运行 Eclipse,下载 Android Eclipse 插件,名字为 ADT(Android Developer Tools)。

在菜单中选择 Help→Software Update,在弹出的对话框中选择 Available Software 选项卡并单击 Add Site,如图 1-3 所示。

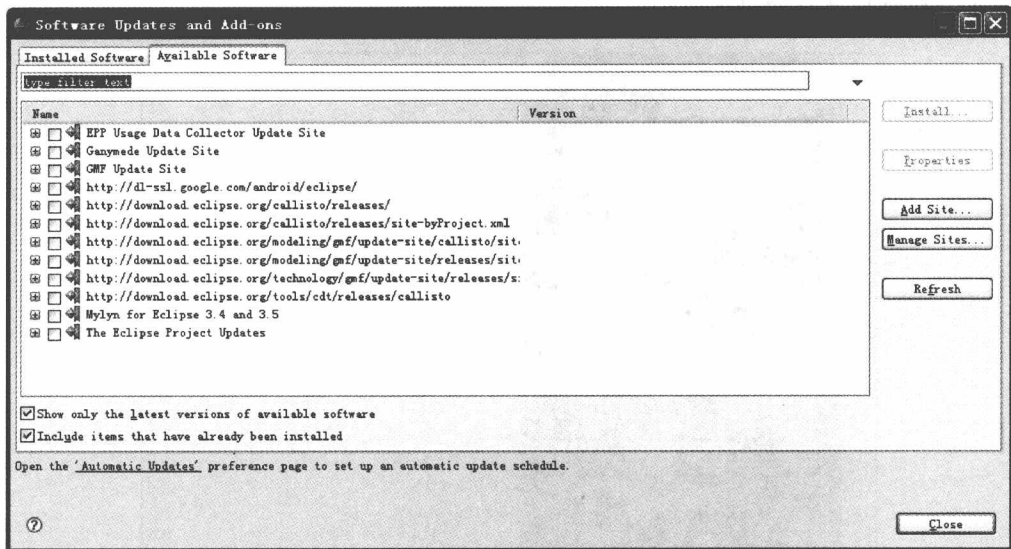


图 1-3 为 Eclipse 添加 ADT 插件

在单击 Add Site 后弹出的对话框中输入 ADT 的网址:<http://dl-ssl.google.com/android/eclipse/>,然后确定。

这时 Available Software 中多了一个如上网址的选项,单击该选项的复选框,并选择出现的子选项 Developer Tools,然后单击右边的 Install 按钮。

这是 Eclipse 安装插件的标准方法。安装完 ADT 插件后重启 Eclipse,此时会发现 Eclipse 的工具栏中多了如图 1-4 所示的图标。



图 1-4 安装完 ADT 后的图标

图 1-4 中,手机图标是用于管理模拟器的;DDMS 是一个调试工具,其用途后面再讲,如

如果你的工具栏中没有 DDMS,可单击其左边那个带加号的图标,在下拉菜单中选择 Other,如图 1-5 所示,并在出现的对话框中选择 DDMS。

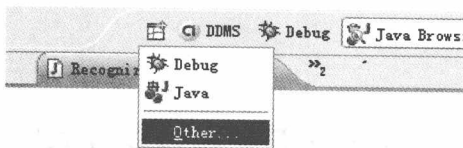


图 1-5 增加 DDMS 视图

⑤ 设置 Eclipse 中 Android SDK 的路径。

在 Eclipse 菜单中,选择 Window→Preferences,在出现的对话框中单击左边列表中的 Android,并在右面板中输入 SDK 的路径,比如“c:\android”。

至此就完成了 Android 开发环境的配置,从下一章我们将开始真正的程序之旅。如果在安装中出现什么问题,可参照 <http://developer.android.com> (该网站有时会出现链接不上 的问题,大家可以去访问它的另一个镜像站点 <http://androidappdocs.appstop.com>) 或者用 Google 搜索一下。

本章问题答案

1. Android 操作系统都包含哪些部分?

答:包含了 5 个部分,分别是 Linux Kernel、Libraries、Android Runtime(Dalvik VM 和 Core Libraries)、Application Framework 以及 Applications。

2. Google 都实现了操作系统的哪些部分?

答:Google 实现了 Application Framework、Dalvik VM 以及一些典型的 Application。

3. 对应用程序开发者而言,需要了解的部分都包含哪些?

答:需了解 Application Framework,该部分定义了 Android 的 GUI 和系统服务框架。

第 2 章

Android 应用程序的组成

本章问题

1. Android 程序的入口点在哪里？
2. 编写的代码都包含哪些文件类型，它们的作用是什么？
3. Android 程序的结构与 Windows 程序的结构一样吗？

Android 程序运行需要一个目标机，目标机可以是一个 Android 设备，也可以是一个模拟器。可能你手头还没有 Android 设备，那么，就先来创建一个模拟器。

2.1 建立一个模拟器

运行 Eclipse，单击工具栏中的“模拟器管理”(类似手机)图标，弹出如图 2-1 所示的对话框。请看 Create AVD 选项区域的内容。

Name: Android 的模拟器可以有多个配置文件，每个配置文件都可以启动一个模拟器，这就意味着可以同时启动多个模拟器。此项是模拟器配置文件的名称，文件所在路径如图 2-1 所示，一般为 C:\Documents and Setting\yourname\.android\avd\xxx.avd，当创建好配置文件后，还可以手工修改该配置文件，从而添加更多其他属性。

Target: 指定该模拟器所运行的 SDK 版本，由于 Android 已经推出了多个版本，就像早期的 Windows 系统一样，以后也会有新的版本不断推出，所以，必须为模拟器指定其所运行的版本号。

SDCard: 手机一般都会有 SD 卡，该选项指定卡的容量大小，单位为兆(M)字节，比如 256 MB。

Skin: 模拟器都会有一个屏幕，屏幕大小如表 2-1 所列，默认选择 HVGA，因为现在市面上的 Android 手机大多数为 HVGA 屏幕。后缀 P 和 L 分别指 Portrait 和 Landscape。Portrait 为肖像的意思，肖像一般都是竖着放；Landscape 为风景的意思，风景一般都横着放。