

Android

系统安全和反编译实战

杨峻 编著



**从 Android 系统介绍到综合实例，
全程剖析了 Android 系统安全的核心知识**

Android 系统的内存安全机制，Binder 通信安全机制，Android 虚拟机，DEX 文件详解，编写安全的应用程序，应用安全策略，文件加密，电话系统的安全机制，短信系统的安全机制，网络防火墙系统，文件加密系统



中国工信出版集团

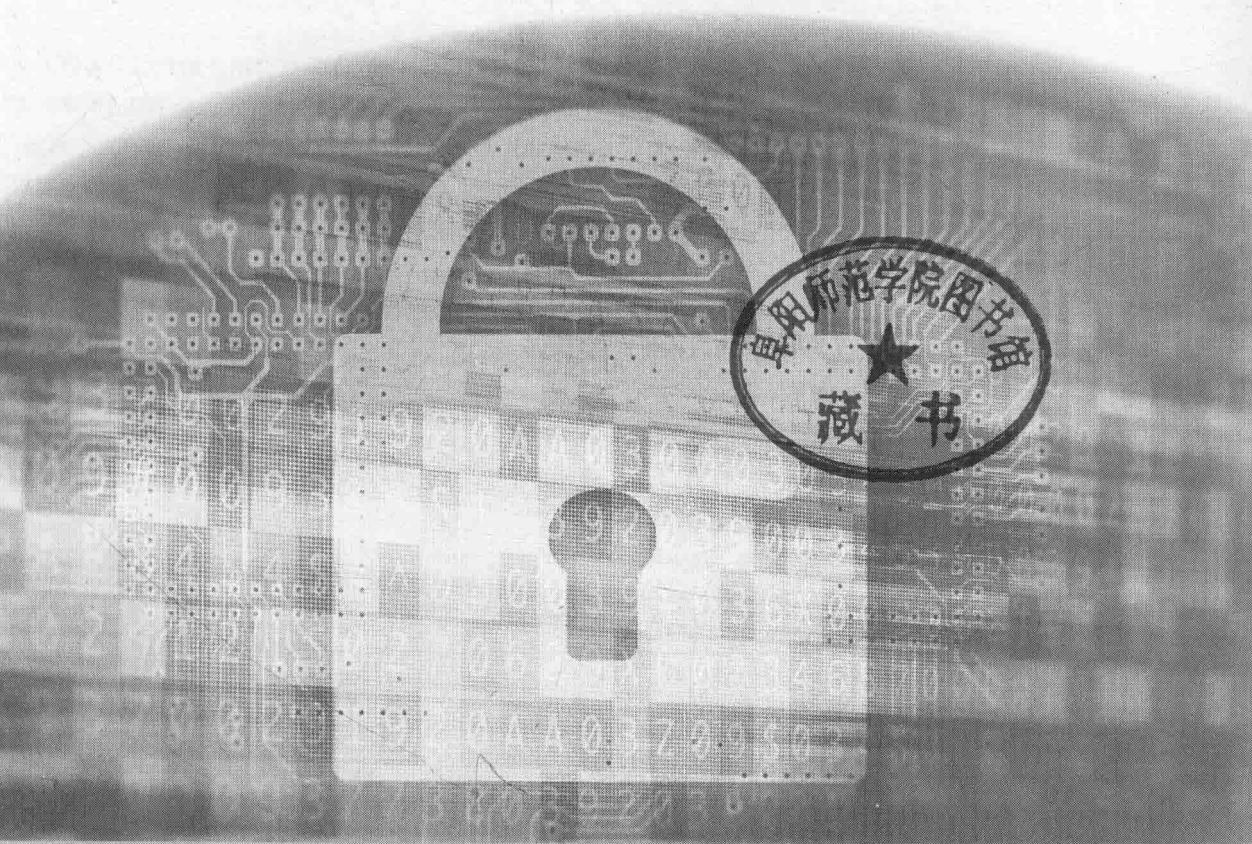


人民邮电出版社
POSTS & TELECOM PRESS

Android

系统安全和反编译实战

杨峻 编著



人民邮电出版社
北京

图书在版编目 (C I P) 数据

Android系统安全和反编译实战 / 杨峻编著. -- 北京 : 人民邮电出版社, 2015.8
ISBN 978-7-115-39303-6

I. ①A… II. ①杨… III. ①移动终端—应用程序—程序设计—安全技术 IV. ①TN929.53

中国版本图书馆CIP数据核字(2015)第152151号

内 容 提 要

本书循序渐进地讲解了 Android 系统安全方面的基本知识,从 Android 系统介绍开始到综合实例的实战过程,全程剖析了 Android 系统安全、应用安全开发和加密、解密方面的核心知识。全书共 17 章,主要内容包括,Android 技术核心框架分析、获取并编译源代码、Android 系统的安全机制、内存安全机制、Binder 通信安全机制、Android 虚拟机基础、Dalvik VM 的运作流程、DEX 文件详解、编写安全的应用程序、应用安全策略、文件加密、电话系统的安全机制、短信系统的安全机制、网络防火墙系统、文件加密系统等核心技术。

本书适合 Android 初学者、Android 应用开发者、Android 安全架构研究人员、Android 底层开发人员和 Android 源代码分析人员学习,也可以作为相关培训学校和大专院校相关专业的教学用书。

-
- ◆ 编 著 杨 峻
责任编辑 张 涛
责任印制 张佳莹 焦志炜
 - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京昌平百善印刷厂印刷
 - ◆ 开本: 787×1092 1/16
印张: 33.75
字数: 935 千字
印数: 1—3 000 册
-

定价: 79.00 元

读者服务热线: (010)81055410 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京崇工商广字第 0021 号

前 言

由于 Android 是开源系统，源程序可以随意浏览和修改，因此，Android 的安全性就受到使用人员的关注。为了更好地帮助读者了解 Android 的安全机制，帮助读者防范黑客的攻击，作者特意撰写了本书。本书对 Android 系统安全和应用开发方面的安全知识进行了细致的分析，本书的主要内容如下。

本书共分 17 章，循序渐进地讲解了 Android 系统安全方面的基本知识，从 Android 系统介绍开始到综合实例的演练，全程剖析了 Android 系统安全、应用开发安全和加密、解密方面的核心知识，几乎涵盖了 Android 系统安全方面的所有主要内容，并且全书写法上言简意赅，讲解通俗易懂，不仅适合有一定基础的读者参考，也特别适合初学者学习使用。

本书特色

本书内容丰富，分析细致。作者的目标是通过一本图书，提供多本图书的价值，读者可以根据自己的需要有选择地阅读。在内容的编写上，本书具有以下特色。

(1) 结构合理

从用户的实际需要出发，合理安排知识结构，内容由浅入深，叙述清楚。全面阐述了与 Android 系统安全和应用安全开发有关的知识。

(2) 遵循“理论介绍—演示实例—综合演练”这一主线

为了使广大读者彻底弄清楚 Android 系统安全和应用安全开发的每一个知识点，在讲解时按照剖析基本理论、演示实例、综合实战演练这个主线进行，希望帮助读者达到学以致用目标。

(3) 易学易懂

本书内容讲解上条理清晰、语言简洁，可以帮助读者快速掌握每个知识点。读者既可以按照本书编排的章节顺序进行学习，也可以根据自己的需求对某一章节进行有针对性的学习。

读者对象

Android 编程的自学者

Android 应用开发人员

Android 源代码研究者

Android 架构师

Android 系统安全和应用安全开发者

Android 编程爱好者

相关培训机构的老师和学员

在编写本书过程中，得到了人民邮电出版社工作人员的大力支持，正是各位编辑的求实、耐心和高效，才使得本书在这么短的时间内得以出版。另外，也十分感谢我的家人，在我创作本书时给予的巨大支持。另外告知各位读者，本人水平毕竟有限，纰漏和不尽如人意之处在所难免，恳请读者提出意见或建议，以便修订使之更臻完善。本书源程序下载和答疑网站为：www.toppr.net。编辑联系邮箱为：zhangtao@ptpress.com.cn。

编 者

目 录

第 1 章 Android 技术概述.....1	2.5.1 先看进程.....29
1.1 智能手机系统介绍.....1	2.5.2 再看线程.....30
1.2 搭建 Android 应用开发环境.....2	2.5.3 应用程序的生命周期.....30
1.2.1 安装 Android SDK 的 系统要求.....2	第 3 章 获取并编译源码.....33
1.2.2 安装 JDK.....2	3.1 在 Linux 系统中获取 Android 源码.....33
1.2.3 获取并安装 Eclipse 和 Android SDK.....5	3.2 在 Windows 平台获取 Android 源码.....34
1.2.4 安装 ADT.....7	3.3 分析 Android 源码结构.....36
1.2.5 设定 Android SDK Home.....8	3.3.1 应用程序.....38
1.2.6 验证开发环境.....9	3.3.2 应用程序框架.....39
1.2.7 创建 Android 虚拟设备 (AVD).....9	3.3.3 系统服务.....39
1.2.8 启动 AVD 模拟器.....10	3.3.4 系统程序库.....41
1.2.9 解决搭建环境过程中 的常见问题.....12	3.3.5 系统运行库.....44
第 2 章 Android 技术核心框架分析.....15	3.3.6 硬件抽象层.....44
2.1 简析 Android 安装文件.....15	3.4 编译源码.....45
2.1.1 Android SDK 目录结构.....15	3.4.1 搭建编译环境.....46
2.1.2 android.jar 及内部结构.....16	3.4.2 开始编译.....46
2.1.3 阅读 SDK 帮助文档.....16	3.4.3 在模拟器中运行.....47
2.1.4 常用的 SDK 工具.....17	3.4.4 常见的错误分析.....48
2.2 演示官方实例.....19	3.4.5 实践演练——演示两种编译 Android 程序的方法.....49
2.3 剖析 Android 系统架构.....23	3.5 编译 Android Kernel.....52
2.3.1 Android 体系结构介绍.....23	3.5.1 获取 Goldfish 内核代码.....52
2.3.2 Android 应用工程 文件组成.....25	3.5.2 获取 MSM 内核代码.....55
2.4 简述五大组件.....27	3.5.3 获取 OMAP 内核代码.....55
2.4.1 用 Activity 来表现界面.....27	3.5.4 编译 Android 的 Linux 内核.....55
2.4.2 用 Intent 和 Intent Filters 实现切换.....28	3.6 编译源码生成 SDK.....57
2.4.3 Service 为你服务.....28	第 4 章 Android 系统的安全机制.....61
2.4.4 用 Broadcast Intent/Receiver 发送广播.....29	4.1 Android 安全机制概述.....61
2.4.5 用 Content Provider 存储数据.....29	4.1.1 Android 的安全机制模型.....61
2.5 进程和线程.....29	4.1.2 Android 的安全框架概述.....62
	4.2 分析 Linux 系统的安全机制.....63
	4.2.1 Linux 用户权限基础.....63
	4.2.2 进程.....68
	4.3 分析 Android 系统的安全机制.....70

4.3.1	沙箱模型介绍	70	6.2.2	分析设备初始化	150
4.3.2	应用程序的安全机制	72	6.2.3	打开 Binder 设备文件	151
4.3.3	分区加载机制	73	6.2.4	内存映射	152
第 5 章	内存安全机制——匿名共享		6.2.5	释放物理页面	156
	内存系统	74	6.2.6	分配内核缓冲区	157
5.1	分析 Ashmem 驱动程序	74	6.2.7	释放内核缓冲区	158
5.1.1	基础数据结构	74	6.2.8	查询内核缓冲区	160
5.1.2	初始化处理	75	第 7 章	Binder 通信安全机制（下）	162
5.1.3	打开匿名共享内存		7.1	Binder 封装库	162
	设备文件	76	7.1.1	类 BBinder	163
5.1.4	内存映射	78	7.1.2	类 BpRefBase	165
5.1.5	实现读写操作	79	7.1.3	类 IPCThreadState	166
5.1.6	锁定和解锁	81	7.2	初始化 Java 层 Binder 框架	168
5.1.7	回收内存块	86	7.3	分析 MediaServer 的通信机制	170
5.2	分析 C++访问接口层	87	7.3.1	MediaServer 的入口函数	170
5.2.1	接口 MemoryBase	87	7.3.2	ProcessState	171
5.2.2	客户端实现	90	7.3.3	defaultServiceManager	173
5.2.3	接口 MemoryBase	94	7.3.4	注册 MediaPlayerService	179
5.3	分析 Java 访问接口层	97	7.3.5	分析 StartThread Pool 和 join Thread Pool	188
5.4	内存优化机制	100	7.4	总结进程通信机制的安全性	190
5.4.1	sp 和 wp 简析	100	7.4.1	进程多线程安全	190
5.4.2	详解智能指针	102	7.4.2	远程过程调用机制 (RPC)	191
5.5	Android 内存系统的安全		7.4.3	实现线程安全方法	192
	机制分析	118	7.4.4	Binder 中的安全策略	193
5.5.1	Ashmem 匿名共享内存 的机理	119	第 8 章	Android 虚拟机基础	194
5.5.2	使用 Low Memory Killer 机制实现安全和高效	119	8.1	Dalvik VM 和 JVM 的差异	194
第 6 章	Binder 通信安全机制（上）	121	8.2	Dalvik 虚拟机的主要特征	195
6.1	Binder 机制基础	121	8.3	Dalvik VM 架构	196
6.1.1	选择 Binder 机制的 原因——简洁快速、 低耗内存、更加安全	121	8.3.1	Dalvik 虚拟机的代码结构	196
6.1.2	Binder 安全机制的 必要性	122	8.3.2	dx 工具	198
6.1.3	Android 的进程间通信 (IPC) 机制 Binder	122	8.3.3	Dalvik VM 的进程管理	198
6.1.4	Service Manager 是 Binder 机制的上下文管理者	123	8.3.4	Android 的初始化流程	198
6.1.5	Service Manager 服务	137	8.4	Dalvik VM 控制 VM 命令详解	199
6.2	分析 Binder 驱动程序	140	8.4.1	基本命令	199
6.2.1	分析数据结构	140	8.4.2	扩展的 JNI 检测	199
			8.4.3	断言	200
			8.4.4	字节码校验和优化	200
			8.4.5	Dalvik VM 的运行模式	201
			8.4.6	死锁预测	201
			8.4.7	dump 堆栈追踪	202

8.4.8	dex 文件和校验	202	9.4.2	启动第一个 Activity	255
8.4.9	产生标志位	202	9.5	加载 class 类文件	255
8.5	Dalvik VM 进程管理	202	9.5.1	DexFile 在内存中的映射	255
8.5.1	Zygote 基础	202	9.5.2	ClassObject——Class 在 加载后的表现形式	257
8.5.2	Dalvik 的进程模型	211	9.5.3	加载 Class 并生成相应 ClassObject 的函数	257
8.5.3	Dalvik 的进程通信	215	9.5.4	加载基本类库文件	258
8.6	Zygote (孕育) 进程	218	9.5.5	加载用户类文件	258
8.6.1	Zygote 基础	218			
8.6.2	分析 Zygote 的启动过程	219			
第 9 章 Dalvik VM 的运作流程		233	第 10 章 DEX 文件详解		259
9.1	Dalvik VM 相关的可执行程序	233	10.1	DEX 文件介绍	259
9.1.1	dalvikvm、dvz 和 app_process 简介	233	10.2	DEX 文件的格式	259
9.1.2	对比 app_process 和 dalvikvm 的执行过程	234	10.2.1	map_list	260
9.2	初始化 Dalvik VM	236	10.2.2	string_id_item	262
9.2.1	开始虚拟机的准备工作	236	10.2.3	type_id_item	266
9.2.2	初始化跟踪显示系统	237	10.2.4	proto_id_item	266
9.2.3	初始化垃圾回收器	237	10.2.5	ield_id_item	267
9.2.4	初始化线程列表和 主线程环境参数	237	10.2.6	method_id_item	267
9.2.5	分配内部操作方法的 表格内存	238	10.2.7	class_def_item	268
9.2.6	初始化虚拟机的指令码 相关的内容	238	10.3	DEX 文件结构	271
9.2.7	分配指令寄存器状态 的内存	239	10.3.1	文件头 (File Header)	271
9.2.8	分配指令寄存器状态 的内存和最基本用的 Java 库	239	10.3.2	魔数字段	272
9.2.9	初始化使用的 Java 类库 线程类	240	10.3.3	检验码字段	272
9.2.10	初始化虚拟机使用的 异常 Java 类库	241	10.3.4	SHA-1 签名字段	274
9.2.11	初始化其他对象	242	10.3.5	map_off 字段	274
9.3	启动 Zygote	250	10.3.6	string_ids_size 和 off 字段	275
9.3.1	在 init.rc 中配置 zygote 启动参数	250	10.4	DEXFile 接口详解	277
9.3.2	启动 Socket 服务端口	250	10.4.1	构造函数	277
9.3.3	加载 preload-classes	251	10.4.2	公共方法	278
9.3.4	加载 preload-resources	252	10.5	DEX 和动态加载类机制	280
9.4	启动 SystemServer 进程	253	10.5.1	类加载机制	280
9.4.1	启动各种系统服务线程	253	10.5.2	具体加载	281
			10.5.3	代码加密	283
			10.6	动态加载 jar 和 DEX	283
第 11 章 编写安全的应用程序		284			
11.1	开发第一个 Android 应用程序	284	11.1	开发第一个 Android 应用程序	284
11.1.1	新创建 Android 工程	284	11.1.1	新创建 Android 工程	284
11.1.2	编写代码和代码分析	284	11.1.2	编写代码和代码分析	284
11.1.3	调试	285	11.1.3	调试	285
11.1.4	运行项目	287	11.1.4	运行项目	287
11.2	声明不同的权限	287	11.2	声明不同的权限	287

11.2.1	AndroidManifest.xml 文件基础	288	12.2.3	分析 WebKit 的 操作过程	344
11.2.2	声明获取不同的权限	288	12.2.4	WebView 讲解	347
11.2.3	自定义一个权限	292	12.2.5	Android 浏览器的安全 漏洞分析	349
11.3	发布 Android 程序生成 APK	293	12.3	数据库和 SQL 注入	350
11.3.1	什么是 APK 文件	293	12.3.1	Android 系统存储数据 的方式	351
11.3.2	申请会员	294	12.3.2	恶意获取存储卡信息	354
11.3.3	生成签名文件	296	12.3.3	SQLite 注入漏洞分析	354
11.3.4	使用签名文件	301	12.4	软件更新安全策略	357
11.3.5	发布到市场	303	12.4.1	设置软件实现自动更新	358
11.4	反编译 APK 文件	303	12.4.2	自动更新的安全隐患	365
11.4.1	使用 dex2jar 和 jdgui.exe 进行反编译	303	第 13 章	文件加密	366
11.4.2	Smali 指令破解	305	13.1	Dmccrypt 加密机制介绍	366
11.5	防止 APK 文件被反编译	306	13.1.1	Linux 密码管理机制	366
11.6	操作实战——演示反编译和 编译的完整过程	306	13.1.2	Dmccrypt 加密机制分析	371
11.6.1	分析反编译后的文件	307	13.1.3	使用 dmccrypt 机制构建 加密文件系统	378
11.6.2	分析 smali 文件	309	13.2	Vold 机制介绍	380
11.6.3	分析内部类	316	13.2.1	Vold 机制基础	380
11.6.4	分析监听器	321	13.2.2	Vold 的主要功能	384
11.6.5	分析注解类	322	13.2.3	Vold 处理 SD/USB 的流程	389
11.6.6	Android 独有的自动类	324	13.2.4	Vold 的加密机制	410
11.7	Android 应用组件的安全机制	325	第 14 章	电话系统的安全机制	412
11.7.1	设置组件的可访问性	325	14.1	Android 电话系统详解	412
11.7.2	Intent 组件的安全机制	325	14.1.1	电话系统简介	412
11.7.3	Activity 组件的安全机制	329	14.1.2	电话系统结构	413
11.7.4	Content Provider 组件 的权限机制	330	14.1.3	需要移植的内容	415
11.7.5	Service 组件的安全机制	332	14.1.4	驱动程序介绍	416
11.7.6	Broadcast Receiver 组件 的安全机制	333	14.1.5	RIL 接口	417
第 12 章	应用安全策略	334	14.1.6	分析电话系统的 实现流程	420
12.1	应用程序安装安全策略	334	14.2	分析 Andorid 音频系统	425
12.1.1	Android 应用程序权限 的背景	334	14.2.1	音频系统结构	425
12.1.2	Android 应用程序 的权限	334	14.2.2	分析音频系统的层次	427
12.1.3	控制 Android 应用权限 的工具	336	14.3	电话系统的安全机制	433
12.2	Android 浏览器安全策略	338	14.3.1	防止电话监听	433
12.2.1	WebKit 源码分析	338	14.3.2	VoIP 语音编码和 安全性分析	437
12.2.2	C/C++ 层框架	342	14.3.3	SIP 协议控制	440

14.3.4 在 Android 平台实现 SIP 协议栈	443	16.3.5 登录验证	495
14.3.5 通话加密技术	444	16.3.6 打开/关闭某一个 实施控件	496
第 15 章 短信系统的安全机制	446	第 17 章 文件加密系统实现	499
15.1 Android 短信系统详解	446	17.1 文件加密系统功能模块介绍	499
15.1.1 短信系统的主界面	446	17.2 系统主界面	499
15.1.2 发送普通短信的过程	449	17.3 文件加密模块	502
15.1.3 发送彩信的过程	459	17.3.1 选择加密文件	502
15.1.4 接收短信	467	17.3.2 加密文件	503
15.2 短信加密机制的设计模式	471	17.4 文件解密模块	507
15.2.1 短信编码设计模式	472	17.4.1 选择解密文件	507
15.2.2 DES 短信息加密 解密算法	472	17.4.2 解密文件	507
第 16 章 网络防火墙系统实现	474	17.5 数字签名模块	511
16.1 防火墙系统需求分析	474	17.5.1 数字签名主界面	511
16.2 编写布局文件	475	17.5.2 设置数字签名	512
16.3 编写主程序文件	476	17.5.3 签名验证	516
16.3.1 主 Activity 文件	477	17.5.4 生成密钥	518
16.3.2 帮助 Activity 文件	485	17.6 报文鉴别模块	520
16.3.3 公共库函数文件	486	17.6.1 报文鉴别主界面	520
16.3.4 系统广播文件	494	17.6.2 生成鉴别码	521
		17.6.3 鉴别文件	524
		17.7 编辑文件模块	527

第1章 Android 技术概述

Android 是一种智能手机操作系统，是建立在 Linux 开源系统基础之上的，能够迅速搭载手机软件的解决方案。Android 系统功能十分强大，已经成为当前软件行业的一股新兴力量。本章将简单介绍 Android 的发展过程和背景，并介绍搭建 Android 应用开发环境的基本知识，为学习本书后面的知识打下基础。

1.1 智能手机系统介绍

在 Android 系统诞生之前，智能手机这个新鲜事物大大丰富了人们的生活，得到了广大手机用户的青睐。各大手机厂商在利益的驱动下，纷纷开发出了各种智能手机操作系统，Android 系统就是在这个背景下诞生的。

1. 何谓智能手机

智能手机是指具有像个人电脑那样强大的功能，拥有独立的操作系统，用户可以自行安装应用软件、游戏等第三方服务商提供的程序，并且可以通过移动通信网络接入到互联网中的移动设备。在 Android 系统诞生之前已经有很多优秀的智能手机系统，例如家喻户晓的 Symbian 系列和微软的 Windows Mobile 系列等。

2. 当前主流的智能手机系统

市面上最主流的智能手机系统当属微软的 Windows Mobile、苹果的 iOS 和本书的主角 Android 系统。

① 微软的 Windows Mobile

Windows Mobile 是微软公司的一款杰出产品，Windows Mobile 将熟悉的 Windows 桌面扩展到了个人设备中。使用 Windows Mobile 操作系统的设备主要有手机、PDA、随身音乐播放器等。Windows Mobile 操作系统有 3 种，分别是 Windows Mobile Standard、Windows Mobile Professional、Windows Mobile Classic。当前的最新版本是 Windows Phone 8。

② iOS

iOS 作为苹果移动设备 iPhone 和 iPad 的操作系统，在 App Store 的推动之下，成为了世界上引领潮流的操作系统之一。原本这个系统名为“iPhone OS”，直到 2010 年 6 月 7 日 WWDC 大会上宣布改名为“iOS”。iOS 的用户界面的基础是能够使用多点触控直接操作。控制方法包括滑动、轻触开关及按键。与系统交互包括滑动（Swiping）、轻按（Tapping）、挤压（Pinching，通常用于缩小）及反向挤压（Reverse Pinching or unpinching，通常用于放大）。

③ Android

Android 是我们本书的主角，于 2007 年 11 月 5 日宣布的基于 Linux 平台的开源手机操作系统的名称，该平台由操作系统、中间件、用户界面和应用软件组成，号称是首个为移动终端打造的真正开放和完整的移动软件。

根据国际数据公司（IDC）公布的新数据，在 2013 年第一季度，Android 和 iOS 系统的装机量占到所有智能手机出货量的 92.3%。在 2013 年第一季度，安装 Android 系统的新智能手机数量跃升至 1.621 亿部，大大超过上年同期的 9030 万部。这意味着，在运往世界各地的所有新智能手机中，谷歌的移动操作系统的市场占有率已经达到 75%。

到本书截稿之时，Android 的最新版本是 Android 5.0。

12

搭建 Android 应用开发环境

“工欲善其事，必先利其器”出自《论语》，意思是要想高效地完成一件事，需要有一个好用的工具。对于安卓开发人员来说，开发工具同样至关重要。作为一项新兴技术，在进行开发前首先要搭建一个对应的开发环境。本节将详细讲解搭建 Android 应用开发环境的基本知识。

1.2.1 安装 Android SDK 的系统要求

在搭建之前，一定先确定基于 Android 应用软件开发环境的需求参数，具体如表 1-1 所示。

表 1-1 开发系统所需参数

项 目	版本要求	说 明	备 注
操作系统	Windows XP 或 Vista Mac OS X 10.4.8+Linux Ubuntu Drapper	根据自己的电脑自行选择	选择自己最熟悉的操作系统
软件开发包	Android SDK	选择最新版本的 SDK	截止到目前，最新手机版本是 5.0
IDE	Eclipse IDE+ADT	Eclipse3.3 （ Europa ） ， 3.4 （ Ganymede ） ADT （ Android Development Tools ） 开发插件	选择 “for Java Developer”
其他	JDK Apache Ant	Java SE Development Kit 5 或 6 Linux 和 Mac 上使用 Apache Ant 1.6.5+， Windows 上使用 1.7+版本	不能单独使用 JRE，必须要有 JDK

Android 应用工具是由多个开发包组成的，具体说明如下。

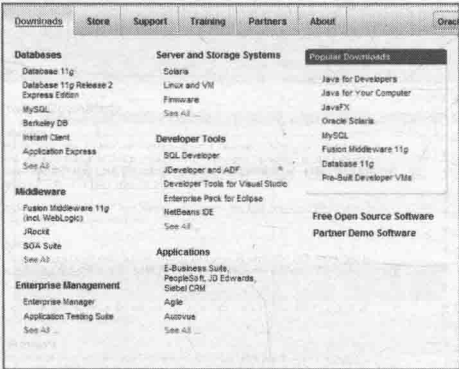
- JDK：可以到网址 <http://java.sun.com/javase/downloads/index.jsp> 下载。
- Eclipse（Europa）：可以到网址 <http://www.eclipse.org/downloads/> 下载 Eclipse IDE for Java Developers。
- Android SDK：可以到网址 <http://developer.android.com> 下载。
- 还有对应的开发插件。

1.2.2 安装 JDK

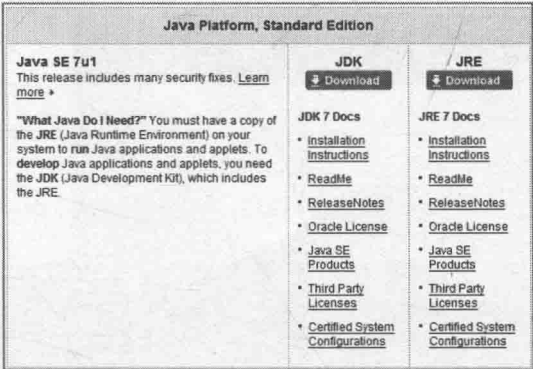
JDK（Java Development Kit）是整个 Java 的核心，包括了 Java 运行环境、Java 工具和 Java 基础的类库。JDK 是学好 Java 的第一步，是开发和运行 Java 环境的基础，当用户要对 Java 程序进行编译的时候，必须先获得对应操作系统的 JDK，否则将无法编译 Java 程序。在安装 JDK 之前需要先获得 JDK，获得 JDK 的操作流程如下所示。

（1）登录 Oracle 官方网站，网址为 <http://developers.sun.com/downloads/>，如图 1-1 所示。

(2) 在图 1-1 中可以看到有很多版本，在此选择当前最新的版本 Java 7，下载页面如图 1-2 所示。



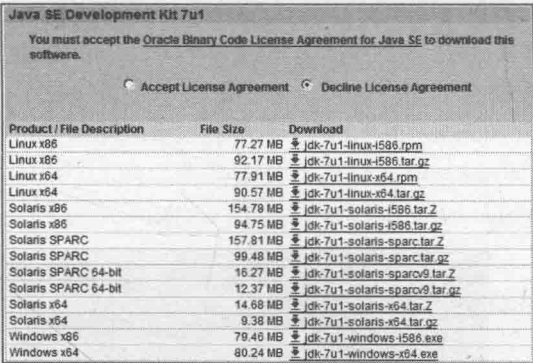
▲图 1-1 Oracle 官方下载页面



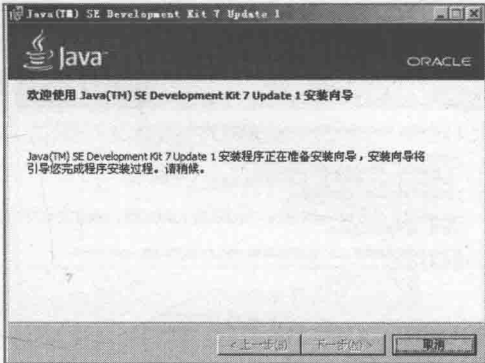
▲图 1-2 JDK 下载页面

(3) 在图 1-2 中单击 JDK 下方的【Download】按钮，在弹出的新界面中选择将要下载的 JDK，在此选择 Windows X86 版本，如图 1-3 所示。

(4) 下载完成后双击下载的“.exe”文件进行安装，将弹出“安装向导”对话框，在此单击【下一步】按钮，如图 1-4 所示。



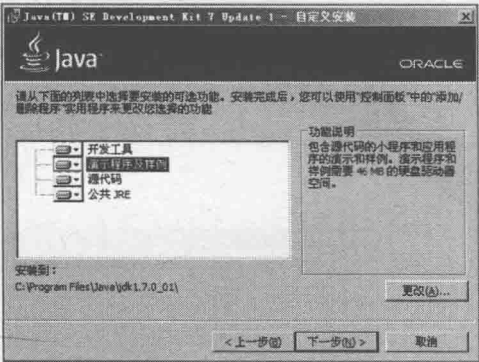
▲图 1-3 选择 Windows X86 版本



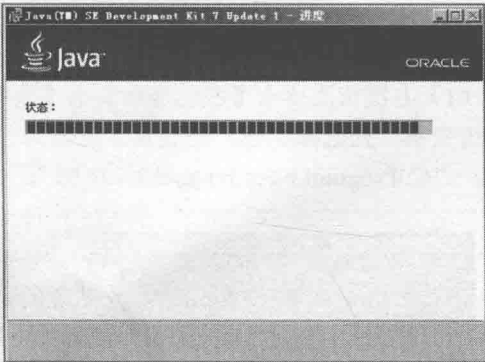
▲图 1-4 “许可证协议”对话框

(6) 弹出“安装路径”对话框，在此选择文件的安装路径，如图 1-5 所示。

(7) 在此设置安装路径为“E:\jdk1.7.0_01\”，然后单击【下一步】按钮开始在安装路径下解压缩下载的文件，如图 1-6 所示。



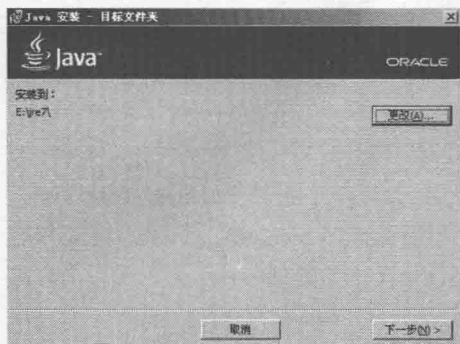
▲图 1-5 “安装路径”对话框



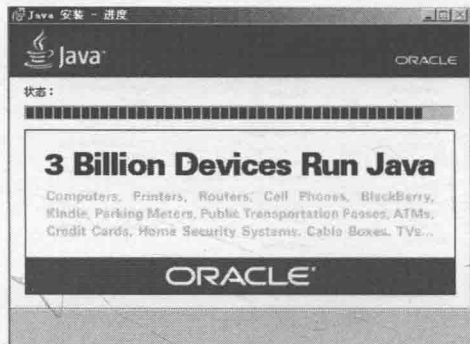
▲图 1-6 解压缩下载的文件

(8) 完成后弹出“目标文件夹”对话框，在此选择程序要安装的位置，如图 1-7 所示。

第 9 步：单击【下一步】按钮后开始正式安装，如图 1-8 所示。



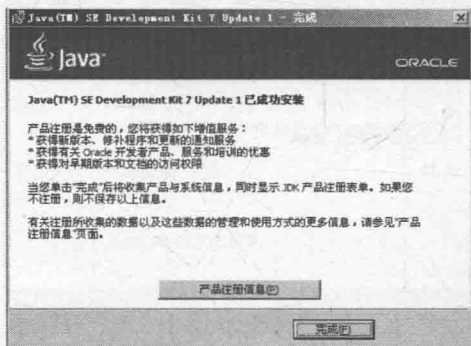
▲图 1-7 “目标文件夹”对话框



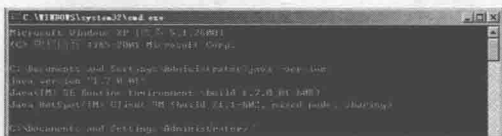
▲图 1-8 继续安装

第 10 步：完成后弹出“完成”对话框，单击【完成】按钮后完成整个安装过程，如图 1-9 所示。

完成安装后可以检测是否安装成功，检测方法是依次单击【开始】|【运行】，在运行框中输入“cmd”并按下回车键，在打开的 CMD 窗口中输入“java -version”，如果显示如图 1-10 所示的提示信息，则说明安装成功。



▲图 1-9 完成安装



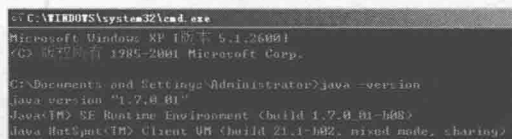
▲图 1-10 CMD 窗口

注意

完成安装后可以检测是否安装成功，方法是依次单击【开始】|【运行】，在运行框中输入“cmd”并按下回车键，在打开的 CMD 窗口中输入“java -version”，如果显示如图 1-11 所示的提示信息，则说明安装成功。

如果检测没有安装成功，需要将其目录的绝对路径添加到系统的 PATH 中。具体做法如下所示。

(1) 右键依次单击【我的电脑】|【属性】|【高级】，单击下面的“环境变量”，在下面的“系统变量”处选择新建，在变量名处输入“JAVA_HOME”，变量值中输入刚才的目录，比如设置为“C:\Program Files\Java\jdk1.7.0_01”，如图 1-12 所示。



▲图 1-11 CMD 窗口



▲图 1-12 设置系统变量

(2) 再次新建一个变量，取名为 `classpath`，其变量值如下所示：

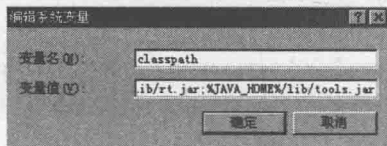
```
.;%JAVA_HOME%/lib/rt.jar;%JAVA_HOME%/lib/tools.jar
```

单击【确定】按钮找到 `PATH` 的变量，双击或单击编辑，在变量值最前面添加如下值：

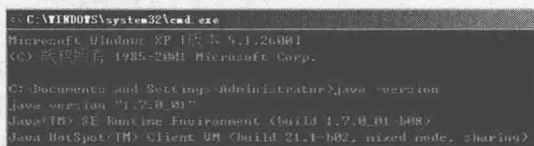
```
%JAVA_HOME%/bin;
```

具体如图 1-13 所示。

(3) 再依次单击【开始】|【运行】，在运行框中输入“`cmd`”并按下回车键，在打开的 `CMD` 窗口中输入“`java -version`”，如果显示如图 1-14 所示的提示信息，则说明安装成功。



▲图 1-13 设置系统变量



▲图 1-14 CMD 界面



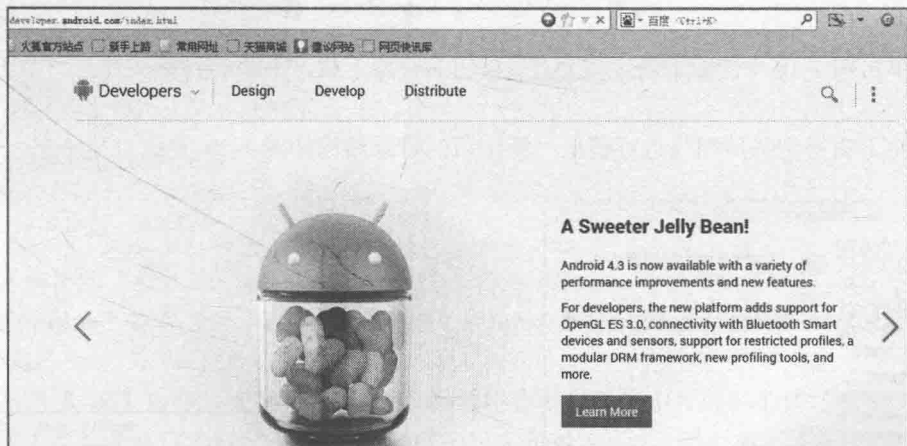
注意

上述变量设置中，是按照笔者计算机上的安装路径设置的，安装的 `JDK` 的路径是 `C:\Program Files\Java\jdk1.7.0_01`。

1.2.3 获取并安装 Eclipse 和 Android SDK

在安装好 `JDK` 后，接下来需要安装 `Eclipse` 和 `Android SDK`。`Eclipse` 是进行 `Android` 应用开发的一个集成工具，而 `Android SDK` 是开发 `Android` 应用程序必须具备的框架。在 `Android` 官方公布的最新版本中，已经将 `Eclipse` 和 `Android SDK` 这两个工具进行了集成，一次下载即可同时获得这两个工具。获取并安装 `Eclipse` 和 `Android SDK` 的具体步骤如下所示。

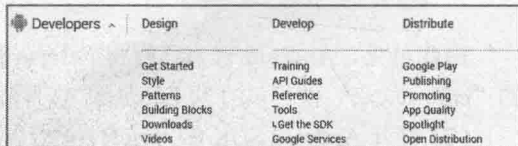
(1) 登录 `Android` 的官方网站 <http://developer.android.com/index.html>，如图 1-15 所示。



▲图 1-15 Android 的官方网站

(2) 单击图 1-15 左上方“`Developers`”右边的“`∨`”符号，在弹出的界面中单击“`Get the SDK`”链接，如图 1-16 所示。

(3) 在弹出的新页面中单击【`Download the SDK`】按钮，如图 1-17 所示。

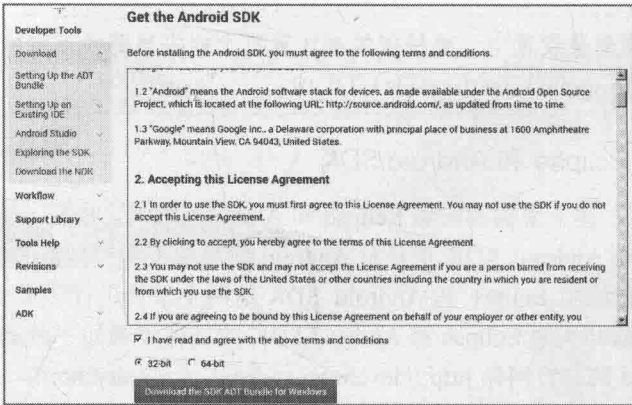


▲图 1-16 单击“`Get the SDK`”链接



▲图 1-17 单击“Download the SDK”按钮

(4)在弹出的“Get the Android SDK”界面中勾选“I have read and agree with the above terms and conditions”前面的复选框，然后在下面的单选按钮中选择系统的位数。例如笔者的机器是 32 位的，所以在此勾选“32-bit”前面的单选按钮，如图 1-18 所示。



▲图 1-18 “Get the Android SDK”界面

(5)单击图 1-18 中的“Download the SDK ADT Bundle for Windows”按钮后开始下载，下载的目标文件是一个压缩包，如图 1-19 所示。

(6)将下载得到的压缩包进行解压，解压后的目录结构如图 1-20 所示。



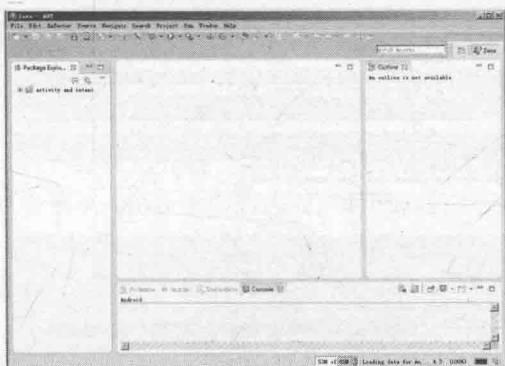
▲图 1-19 开始下载目标文件压缩包

eclipse	2013/10/11 18:28	文件夹
sdk	2013/7/26 18:16	文件夹
SDK Manager.exe	2013/5/22 18:57	应用程序

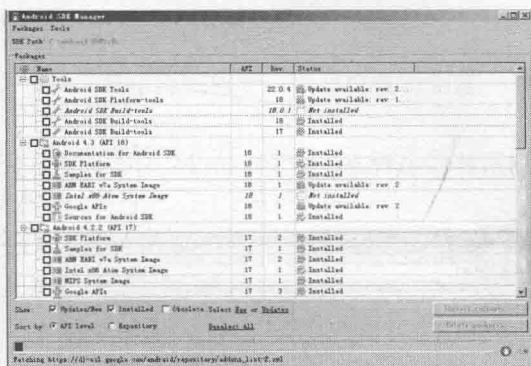
▲图 1-20 解压后的目录结构

由此可见，Android 官方已经将 Eclipse 和 Android SDK 实现了集成。双击“eclipse”目录中的“eclipse.exe”便可以打开 Eclipse，打开后的界面效果如图 1-21 所示。

(7)打开 Android SDK 的方法有两种，第一种是双击下载目录中的“SDK Manager.exe”文件，第二种是在 Eclipse 工具栏中单击图标，打开后的效果如图 1-22 所示。



▲图 1-21 打开 Eclipse 后的界面效果



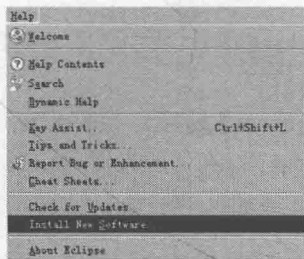
▲图 1-22 打开 Android SDK 后的界面效果

1.2.4 安装 ADT

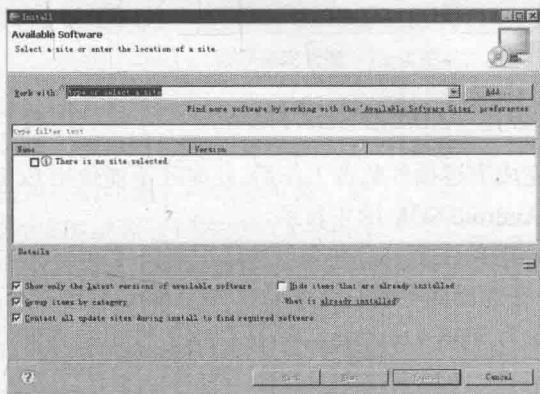
Android 为 Eclipse 定制了一个专用插件 Android Development Tools (ADT)，此插件为用户提供了一个强大的开发 Android 应用程序的综合环境。ADT 扩展了 Eclipse 的功能，可以让用户快速地建立 Android 项目，创建应用程序界面。要安装 Android Development Tools plug-in，首先需要打开 Eclipse IDE，然后进行如下操作。

(1) 打开 Eclipse 后，依次单击菜单栏中的【Help】|【Install New Software...】选项，如图 1-23 所示。

(2) 在弹出的对话框中单击【Add】按钮，如图 1-24 所示。



▲图 1-23 添加插件



▲图 1-24 添加插件

(3) 在弹出的“Add Site”对话框中分别输入名字和地址，名字可以自己命名，例如“123”，但是在“Location”中必须输入插件的网络地址“http://dl-ssl.google.com/Android/eclipse/”，如图 1-25 所示。

(4) 单击【OK】按钮，此时在“Install”界面将会显示系统中可用的插件，如图 1-26 所示。

(5) 选中“Android DDMS”和“Android Development Tools”，然后单击【Next】按钮来到安装界面，如图 1-27 所示。

(6) 选择“I accept...”选项，单击【Finish】按钮，开始进行安装，如图 1-28 所示。

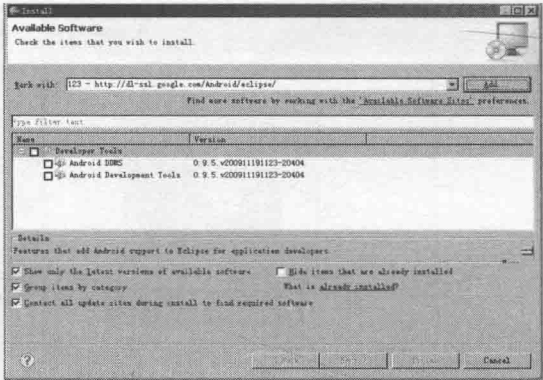


注意

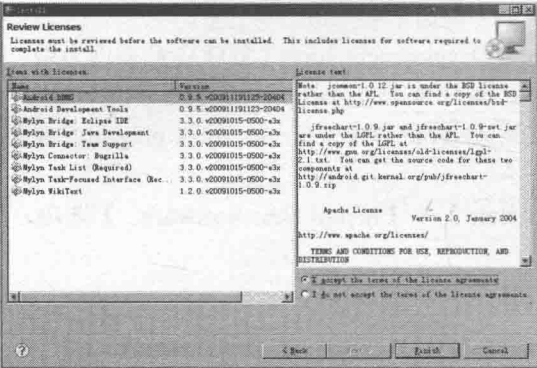
在上个步骤中，可能会发生计算插件占用资源情况，过程有点慢。完成后会提示重启 Eclipse 来加载插件，等重启后就可以用了。不同版本的 Eclipse 安装插件的方法和步骤是不同的，但是都大同小异，读者可以根据操作提示自行解决。



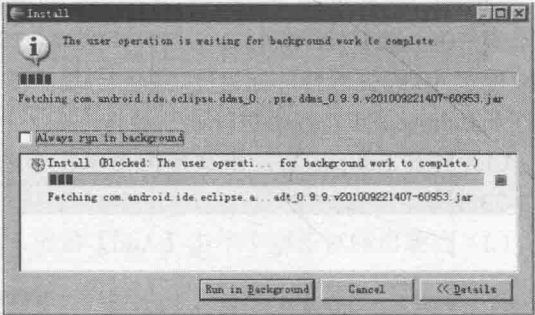
▲图 1-25 设置地址



▲图 1-26 插件列表



▲图 1-27 插件安装界面

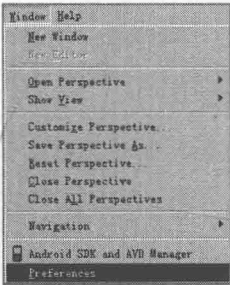


▲图 1-28 开始安装

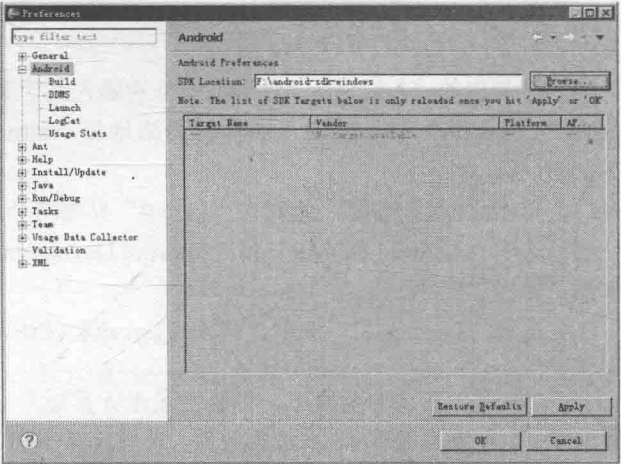
1.2.5 设定 Android SDK Home

当完成上述插件装备工作后,此时还不能使用 Eclipse 创建 Android 项目,我们还需要在 Eclipse 中设置 Android SDK 的主目录。

- (1) 打开 Eclipse, 在菜单中依次单击【Windows】|【Preferences】选项, 如图 1-29 所示。
- (2) 在弹出的界面左侧可以看到“Android”根目录, 选中 Android 后, 在右侧设定 Android SDK 所在目录为 SDK Location, 单击【OK】按钮完成设置, 如图 1-30 所示。



▲图 1-29 Preferences 项



▲图 1-30 Preferences 对话框