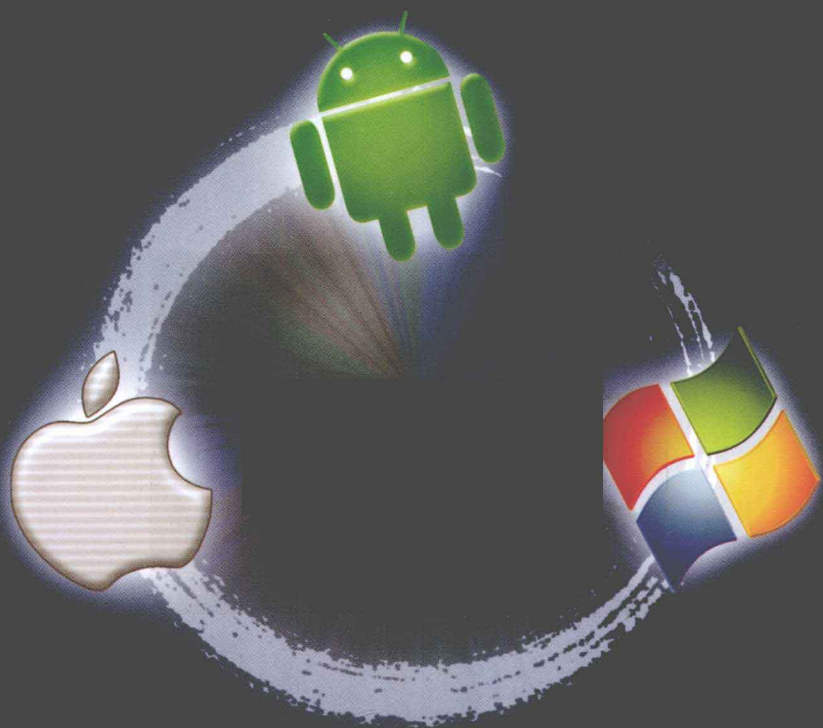


移动开发技术丛书

# 移动开发全平台 解决方案

ANDROID IOS  
WINDOWS PHONE

© 编著/DevDiv移动开发社区

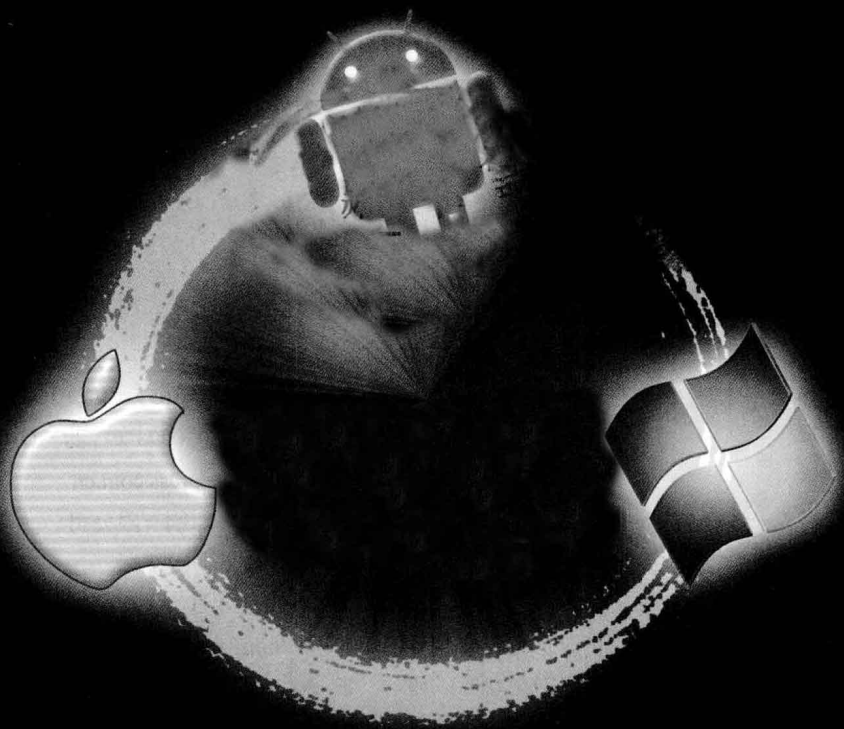


移动开发技术丛书

# 移动开发全平台 解决方案

ANDROID IOS  
WINDOWS PHONE

© 编著/DevDiv移动开发社区



海洋出版社

2011年·北京

## 内 容 简 介

移动开发进入了 Android、iOS、Windows Phone 三足鼎立的时代，第三方软件制造商普遍实行多平台战略，开发人员需要经常在不同平台间转换。为帮助读者在某一个平台快速入门，同时类比学习其他平台，DevDiv 移动开发社区版主、资深会员联合编写本书。

**主要内容：**本书涉及了移动开发中的 25 个经典问题，并分别给出 Android、iOS、Windows Phone 三大平台各自的解决方案，基本涵盖了手机开发人员普遍面临的研发难题。

**适用范围：**本书适合 Android、iOS、Windows Phone 任一平台的初、中级开发者参考使用，同时也可帮助单一平台开发者快速学习其他平台。

### 图书在版编目(CIP)数据

移动开发全平台解决方案：Android/iOS/WindowsPhone/DevDiv 移动开发社区编著. —北京：海洋出版社，2011.8

ISBN 978-7-5027-8065-4

I. ①移… II. ①D… III. ①移动电话机—应用程序—程序设计 IV. ①TN929.53

中国版本图书馆 CIP 数据核字（2011）第 139805 号

总 策 划：吕允英

责任编辑：涂 愷

责任校对：肖新民

责任印制：刘志恒

排 版：海洋计算机图书输出中心 晓阳

出版发行：**海洋出版社**

地 址：北京市海淀区大慧寺路 8 号（716 房间）

100081

经 销：新华书店

技术支持：（010）62100057

发 行 部：（010）62173651（传真）（010）62132549

（010）68038093（邮购）（010）62100077

网 址：[www.oceanpress.com.cn](http://www.oceanpress.com.cn)

承 印：北京东海印刷有限公司

版 次：2011 年 8 月第 1 版

2011 年 8 月第 1 次印刷

开 本：787mm×1092mm 1/16

印 张：26

字 数：520 千字

印 数：1~3000 册

定 价：72.00 元

本书如有印、装质量问题可与发行部调换

## 序 言

随着 3G、云计算等技术从概念转为现实，移动互联网迎来了高速、爆炸式的增长，带动了移动互联网应用的快速发展。

移动互联网应用可针对不同行业 and 不同客户的需求，面向电信、保险、航空等行业，为个人、家庭、企业、政府提供社交、娱乐、办公等服务，这些服务融入了移动通信的特点，从而使用户可以随时随地地获得服务。举例来说，基于 3G 的应用，不仅有视频通话，还包括移动支付、移动商务、RFID、NFC 和二维码等。特别是基于位置的服务，将成为移动互联网业务的亮点之一。随着人们对位置信息需求的日益增加，运营商通过手机地图和 GPS 数据的整合，可以为手机用户提供定位服务；用户还可以实时地看到某街道是否堵塞，精确地搜索到自己最近的餐厅、商场等；商家也可以精确地推送位置相关的广告给特定范围的顾客。

我们注意到，移动互联网应用的核心是手机，虽然移动互联网应用需要服务器作为后台的业务支撑，但手机上的终端服务和应用是吸引用户的关键。

由于智能手机提供了更为绚烂多彩的用户界面、更多的应用程序，因此，其增长速度远远超过了非智能手机，智能手机的市场占有率甚至决定了手机厂商的成败。随着 Symbian、Windows Mobile 的快速没落，智能手机操作系统进入了全新的 Android、iOS、Windows Phone 三足鼎立的时代。Android 因拥有良好的用户界面和开发上的便利而受到各大手机厂商的追捧，成为市场占有率增长最快的操作系统，在可预期的未来必将成为市场占有率最大的操作系统；而装备 iOS 的手机和平板电脑，其用户体验都是最优秀的，至今还没有哪一款手机的销售量能超过 iPhone，其他平台的平板电脑更是没有办法与 iPad 竞争；微软在 Windows Mobile 市场快速衰退之际，推出了 Windows Phone 手机，为用户提供了不同于 Android 与 iOS 的用户体验，强调 Metro 体验，微软认为，随着生活节奏越来越快，简洁、直接的用户体验更加适合现代的用户。近期，诺基亚决定采用 Windows Phone 作为主要的操作系统，各大手机厂商也纷纷开始支持 Windows Phone 手机，业界普遍预测 Windows Phone 将成为主流操作系统。

多平台长期并存的移动互联时代要求第三方软件制造商及个人开发者必须拥有多平台开发技能，这样才能立于不败之地。借助这本《移动开发全平台解决方案》，读者能够迅速从某一个平台入门，然后与其他平台进行类比，并深入学习，进而掌握移动互联时代的先机。

中国科学院博士、博士生导师，软件研究所研究员



2011 年夏

# 前 言

## 全平台

移动互联网已经进入了爆炸式增长的时期，3G、三网合一、云计算等技术也已经从理论变为现实。就移动终端技术来说，随着 Symbian、Windows Mobile 的快速没落，操作系统进入了全新的 Android、iOS、Windows Phone 三足鼎立的时代。

Android 拥有良好的用户界面，受到各大手机厂商的追捧，成为市场占有率增长最快的操作系统，在可预期的将来必定会成为市场占有率最高的操作系统。

iOS 无论是平板还是手机，目前都是最优秀的，没有哪一款手机的销售量能超过 iPhone，其他平台的平板市场更是没有办法与 iPad 竞争。

微软在 Windows Mobile 市场占有率快速下降之际推出了 Windows Phone 手机，提供了不同于 Android 与 iOS 的用户体验，微软强调的是 Metro 体验，认为随着生活节奏越来越快，简洁而直接的用户体验更加适合现代的用户。市场调研也显示，Windows Phone 最终受到了消费者的青睐。近期，诺基亚决定采用 Windows Phone 作为其产品的主要的操作系统，各大手机厂商也都纷纷开始支持 Windows Phone，业界普遍预测 Windows Phone 将成为主流操作系统。

所以未来的移动终端必然是多平台并存的，而且各大厂商出于战略平衡考虑，不会只做单一平台投入，基本上 Android 和 Windows Phone 的手机都会出，甚至会适时推出自己的操作系统，而苹果公司作为行业创新的领头羊，目前还没有哪一款操作系统能超越 iOS 的用户体验，还没有哪一款硬件能够超越 iPhone 和 iPad。所以，未来的移动市场在相当长时间内会是多平台并存的。

既然多平台长期并存，那么无论对各第三方软件制造商还是个人开发者来说，进行平台开发都是很有必要的，而且也是降低风险的有效手段。我们希望通过一本介绍移动多平台解决方案的书籍，帮助大家迅速在某一个平台入门，同时方便类比学习其他平台。

## 作者

本书由 DevDiv 移动开发社区组织编写，参与本书编写的作者均为社区版主或者资深会员，具体分工如下。

作者	社区 ID	作者简介	写作分工	
主编	周智勋	BeyondVincent	超级版主，从 2000 年开始接触编程，先后经历了 Symbian、Windows Mobile、Brew、Android 和 iOS 等开发，对手机行业有非常深入的了解；经常参加一些 IT 界技术沙龙，多次组织翻译一些手机开发文档	项目协调、组织、排版等工作
	吴学友	Vincent	DevDiv 创始人，曾任职微软、诺基亚、Symbian 等公司从事研发工作，具有近 10 年的移动平台研发经验	全部初稿校验及部分 Windows Phone 章节的编写





(续表)

作者	社区 ID	作者简介	写作分工	
Android 平台 作者	劳永超	fluke	Android 版主, 有 3 年移动应用领域的软件开发和交互设计经验, 现就职于广州优视	第 4、6、7、14、17、18 章
	叶晋	kfl56	超级版主, 有 3 年的 J2ME 开发经验, 现主要从事 Android 应用开发	第 1、2、3、16、19 及 25 章部分内容
	张大伟	David_zhang	Android 版主, 有丰富的 Android 研发经验	第 9、12、13、21、24 章
	唐星	starup	Android 版主, 有丰富的 Android 研发经验	第 7、10、11、22 章
	陈鸿图	sodino	Android 版主, 有丰富的 Android 研发经验	第 15、20 及 25 章部分内容
	张兴业	xyz_lmn	Android 版主, 有丰富的 Android 研发经验	第 5 章
	廖甲元	coldjy	Android 版主, 从事软件业 10 年, 具有丰富的系统分析、架构和项目管理经验, 曾主持过电信大型项目开发。他兴趣广泛, 目前正在关注移动互联网的发展和应用	第 9 章
	刘天智	绿腰	曾任职东软集团大连公司, 主要从事 C/C++/Java 开发, 主要研究领域是图像处理和手机应用程序的开发	第 23 章
iOS 平台 作者	赖桂斌	laigb	iOS/iPhone 版主, 具有两年 iPhone 开发经验, 熟悉系统应用, 熟悉 OpenGL、Quartz 2D 及多媒体框架, 目前从事 MFI 外设开发	第 3、4、6、10、11、15、16、17、18、23、24、25 章
	方彬	Gphone	超级版主, 有丰富的 iOS 及 Android 研发经验	第 1、2、13、20、25 章
	许靖昕	cloudhsu	iOS/iPhone 版主, 来自台湾的程序员, 从事开发工作已达 13 年, 项目范围包括金融、CIM、服务器、硬件控制、游戏等。目前在一家硬件设备公司担任软件架构师, 活用设计样式、泛型等技巧, 设计及开发企业中介框架; 同时兼职做 iPhone 游戏, 使用自己设计的游戏引擎来设计产品, 目前已有 Fight Bingo、Ninja Darts、Toilet War 等产品上架	第 5、9、12、14 章
	杜风雷	doors.du	从 2009 年开始从事 iOS、iPhone、MAC 开发, 对 Symbian 和 WM 平台有深刻的了解, 熟悉 Objective-C 语言、Mac OS 和 Cocoa 开发框架。开发过的 iPhone 项目涉及 HTTP 协议、XML 解析、多线程、流媒体等技术	第 7、9、22 章
	王晶	luckywangjing	具有丰富的 iOS/iPhone 开发经验	第 19、21 章
	杨帆	sailing	具有丰富的 iOS/iPhone 开发经验	第 8 章
Windows Phone 平台 作者	吴学友	Vincent	DevDiv 移动开发社区创始人, 曾任职微软、诺基亚、Symbian 等公司, 具有近 10 年的移动平台研发经验	第 1、2、3、4、7、8、9、12、14、19、21、22、23、24、25 章
	黄永雄	wt0731	Windows Phone 版主, 有多年的 Windows Mobile、Windows Phone 研发经验	第 11、13、20 章
	农镇雨	0nongyu0	Windows Phone 版主, 具有丰富的 Windows Phone 研发经验	第 5 章



## 本书结构

本书按照移动开发中可能涉及的 25 个经典问题设置章节。其中，开发环境搭建、模拟器使用、调试方法这三章是学习其他各章的基础。其余各章彼此之间关联度非常小，这样读者可以直接选择学习自己关心的章节。每一章依次介绍 Android、iOS、Windows Phone 平台中相应的实现方法。

## 目标读者

本书适用于 Android、iOS、Windows Phone 任一平台的初、中级开发者；同时也适用于单一平台开发者快速学习其他平台。

## 学习指南

相对于计算机而言，手机屏幕比较小，存储空间有限，而且受到电池续航能力的影响，因此手机程序要求短小精悍、运行高效。

手机程序开发可分为系统开发、系统移植和应用开发几种。系统开发包括操作系统主要组件和内核的开发，真正做操作系统的人很少；国内有一部分人从事系统移植，主要是山寨厂商，以前移植 Windows CE 的人很多，现在移植 Android 的人比较多；还有一些人做系统内置的软件，其实他们是在进行应用开发。

目前，手机软件开发多数是面向个人用户的，面向企业级的应用还没有大规模上市，如保险、办公自动化等均处于初级阶段。在个人消费领域，Android、iOS、Windows Phone 无论在应用、游戏还是社交、定位技术服务等方面都没有问题。相对来说，iOS 面向高端用户，Android 在中高端用户群体中占有较大市场，Windows Phone 目前只支持较为高端的用户。

从开发方面来说，三个平台差不多，开发工具都比较完善，开发语言难度也差不多。相对来说，Windows Phone 的开发工具更成熟一些，界面设计工具也更为出色；Android 能实现的功能要多一些，所以学习周期要比 iOS 和 Windows Phone 稍长。三个平台的开发思路有很多类似的地方，读者在学习的过程中可以慢慢体会。

本书基本覆盖了移动开发涉及的主要知识点和技术方法。学习之初，可以按照兴趣或者项目需求有选择性地进行学习，对于书中提供的解决方案，可以快速移植到自己的项目中去。有些章节涉及比较通用的技术，例如双缓冲绘图，在进行项目优化时可以使用。

书中很多章节总结了开发过程中容易犯的常见错误，希望大家阅读相关章节时给予特别关注。如果项目调试遇到问题，建议先查找一下书中是否有类似的经验可以借鉴。





## 例程代码

本书中的代码均可在 [http://www.devdiv.com/books/allplatform/mobile\\_dev\\_multi\\_plat\\_code.rar](http://www.devdiv.com/books/allplatform/mobile_dev_multi_plat_code.rar) 免费下载，DevDiv 移动开发社区（[www.devdiv.com](http://www.devdiv.com)）负责代码的维护工作，若有更新会第一时间发布。

编 者



# 目 录

<b>第 1 章 开发环境搭建</b> .....1	
1.1 Android 开发环境搭建.....1	
1.1.1 相关软件准备.....1	
1.1.2 安装与设置.....1	
1.2 iOS 开发环境搭建.....4	
1.2.1 相关软件准备.....5	
1.2.2 安装与设置.....7	
1.2.3 Xcode 整体布局.....8	
1.3 Windows Phone 开发环境搭建.....9	
1.3.1 相关软件准备.....9	
1.3.2 安装与设置.....10	
1.3.3 环境搭建涉及的其他问题.....11	
1.3.4 其他必备工具或方法.....11	
<b>第 2 章 模拟器的使用</b> .....13	
2.1 Android 模拟器.....13	
2.1.1 Android 模拟器的创建.....13	
2.1.2 Android 模拟器的使用.....14	
2.2 iOS 模拟器.....15	
2.2.1 iOS 模拟器简介.....15	
2.2.2 清理界面.....17	
2.2.3 硬件模拟需求.....18	
2.2.4 模拟内存警告.....18	
2.2.5 多点触摸操作.....18	
2.3 Windows Phone 模拟器.....18	
2.3.1 Windows Phone 模拟器界面.....18	
2.3.2 Windows Phone 模拟器的使用.....19	
<b>第 3 章 调试方法</b> .....21	
3.1 Android 调试.....21	
3.1.1 Android 调试桥.....21	
3.1.2 查询模拟器、设备.....22	
3.1.3 发送 ADB 命令.....22	
3.1.4 向特定模拟器、设备实例发送命令.....22	
3.1.5 安装应用程序.....23	
3.1.6 转发端口.....23	
3.1.7 模拟器、设备与文件传输.....23	
3.1.8 ADB 命令列表.....24	
3.1.9 执行 shell 命令.....25	
3.1.10 启用 logcat 日志.....27	
3.1.11 停止 ADB 服务.....29	
3.2 iOS 调试.....29	
3.2.1 终端输出.....30	
3.2.2 窗口提示.....30	
3.2.3 使用 Leaks 工具查看对象与内存.....31	
3.3 Windows Phone 调试.....32	
3.3.1 Windows Phone 工程的构成.....32	
3.3.2 调试 Windows Phone 程序.....36	
3.3.3 打印 Log.....36	
3.3.4 常见问题及解决办法.....36	
<b>第 4 章 触摸屏</b> .....41	
4.1 Android 触摸屏.....41	
4.1.1 Android 输入处理.....41	
4.1.2 Android 多点触摸与手势.....47	
4.2 iOS 触摸屏.....48	
4.2.1 iOS 输入处理.....49	
4.2.2 iOS 多点触摸与手势.....49	
4.3 Windows Phone 触摸屏.....50	
4.3.1 Windows Phone 输入处理.....50	
4.3.2 Windows Phone 多点触摸与手势.....56	





<b>第 5 章 UI 控件</b> .....57	
5.1 Android 常用控件.....57	
5.1.1 Button 控件.....57	
5.1.2 CheckBox 控件.....58	
5.1.3 ListView 控件.....59	
5.1.4 容器控件.....70	
5.2 iOS 常用控件.....74	
5.2.1 Button 控件.....74	
5.2.2 输入控件.....75	
5.2.3 Label 控件.....76	
5.2.4 Switch 控件.....77	
5.2.5 Slider 控件.....77	
5.2.6 定制 UISlider 的 On/Off 显示 文字.....78	
5.3 Windows Phone 常用控件.....84	
5.3.1 Button 控件.....84	
5.3.2 CheckBox 控件.....87	
5.3.3 RadioButton.....88	
5.3.4 Slider 控件.....89	
5.3.5 容器控件.....90	
5.3.6 输入控件.....94	
5.3.7 图形控件.....96	
5.3.8 其他基础控件.....102	
5.3.9 Paviot (枢轴控件).....106	
5.3.10 Panorama (全景视图控件) ...108	
5.3.11 WebBrowser (浏览器控件) ...109	
5.3.12 自定义控件.....110	
5.3.13 扩展控件 (Siverlight Toolkit 控件库).....115	
5.3.14 广告控件.....119	
<b>第 6 章 双缓冲画图</b> .....123	
6.1 Android 双缓冲画图.....123	
6.1.1 Android 画图基础.....123	
6.1.2 双缓冲画图的实现.....127	
6.2 iOS 双缓冲画图.....129	
6.2.1 iOS 中通过双缓冲实现截屏.....129	
6.2.2 双缓冲的应用.....131	
6.3 Windows Phone 双缓冲画图.....131	
<b>第 7 章 线程操作</b> .....132	
7.1 Android 线程操作.....132	
7.1.1 Thread 类实现多线程.....132	
7.1.2 AsyncTask 类实现多线程.....135	
7.1.3 多线程通信原理.....136	
7.2 iOS 线程操作.....137	
7.2.1 线程创建及通信.....137	
7.2.2 范例详解.....140	
7.2.3 iOS 线程的注意事项及测试.....142	
7.3 Windows Phone 线程操作.....142	
7.3.1 Dispatcher 调度与 Rendering Thread 线程原理.....143	
7.3.2 Dispatcher 属性.....144	
7.3.3 多线程实例.....144	
7.3.4 多线程的其他应用.....145	
<b>第 8 章 网络应用</b> .....146	
8.1 Android 网络应用.....146	
8.1.1 Android 的 HTTP 访问类.....146	
8.1.2 使用 WAP 方式访问 HTTP.....149	
8.2 iOS 网络应用.....149	
8.2.1 iOS 的 HTTP 访问类.....150	
8.2.2 Socket 应用.....154	
8.3 Windows Phone 网络应用.....159	
8.3.1 Windows Phone 的 HTTP 访问 类.....159	
8.3.2 WebClient 与 HttpWebRequest 的使用.....160	
<b>第 9 章 多语言支持与编码格式</b> .....162	
9.1 编码相关知识.....162	
9.1.1 编码原理及发展历史.....162	

9.1.2 常见字符集标准 .....	162	11.1.2 assets 下资源图片的使用 .....	202
9.1.3 常见编码标准 .....	163	11.1.3 nine patch 图片的使用 .....	202
9.1.4 BOM .....	163	11.2 iOS 资源图片的使用 .....	206
9.2 Android 多语言支持与编码格式 .....	164	11.3 Windows Phone 资源图片的使用 .....	207
9.2.1 资源文件 .....	165	11.3.1 图片格式选择 .....	207
9.2.2 通过资源文件实现多语言支持 .....	166	11.3.2 图片文件选择 .....	207
9.2.3 设置应用的 Locale (显示语言) .....	168	11.3.3 图片大小限制 .....	207
9.3 iOS 多语言支持与编码格式 .....	172	11.3.4 BuildAction 选项设置 .....	208
9.3.1 iOS 多语言支持 .....	172	11.3.5 释放内存 .....	208
9.3.2 iOS 编码格式 .....	176		
9.4 Windows Phone 多语言支持与编码格式 .....	179	<b>第 12 章 文件、SD 卡操作 .....</b>	<b>210</b>
9.4.1 Windows Phone 多语言支持 .....	179	12.1 Android 文件及 SD 卡操作 .....	210
9.4.2 Windows Phone 编码格式 .....	182	12.1.1 I/O 操作 .....	210
<b>第 10 章 获取手机信息 .....</b>	<b>184</b>	12.1.2 私有文件操作 .....	214
10.1 Android 获取手机信息 .....	184	12.1.3 SD 卡操作 .....	216
10.1.1 获取手机 IMEI 和 SIM 卡相关信息 .....	184	12.2 iOS 文件及 SD 卡操作 .....	217
10.1.2 获取系统属性 .....	185	12.2.1 获取 Home Directory .....	217
10.1.3 获取手机的 CPU 信息和内存信息 .....	188	12.2.2 枚举文件夹 .....	217
10.1.4 获取已经安装的 APK 信息 .....	190	12.2.3 文件夹的建立及删除 .....	218
10.2 iOS 获取手机信息 .....	193	12.2.4 创建与读取文件 .....	219
10.3 Windows Phone 获取手机信息 .....	194	12.2.5 读取与写入 plist 文件 .....	220
10.3.1 获取用户信息 .....	195	12.3 Windows Phone 文件及 SD 卡操作 .....	224
10.3.2 获取设备信息 .....	195	12.3.1 IsolatedStorage .....	224
10.3.3 获取用户和设备信息封装类 DeviceInfo.cs .....	197	12.3.2 IsolatedStorageSettings .....	226
10.3.4 获取用户和设备信息的应用 .....	198	12.3.3 SD 卡访问 .....	228
<b>第 11 章 资源图片的使用 .....</b>	<b>200</b>	12.3.4 关于 IsolatedStorage 的一个常见误解 .....	229
11.1 Android 资源图片的使用 .....	200	<b>第 13 章 数据库操作 .....</b>	<b>230</b>
11.1.1 drawable 下资源图片的使用 .....	200	13.1 Android 数据库操作 .....	230
		13.1.1 Android 集成 SQLite 数据库 .....	231
		13.1.2 Android 数据库实例分析 .....	233
		13.1.3 SQLite 管理工具 .....	236
		13.2 iOS 数据库操作 .....	237
		13.2.1 iOS 中 SQLite 的使用及数据库文件的准备 .....	237





13.2.2	iOS 中数据库的操作.....	239	15.3.1	启动器(Launchers)调用短信、 电话、邮件功能 .....	275
13.2.3	Core Data 在数据库中的应 用.....	240	15.3.2	选择器(Choosers)调用短信、 电话、邮件功能 .....	280
13.3	Windows Phone 数据库操作.....	243	15.3.3	启动器和选择器的区别及使 用方法 .....	289
13.3.1	Isolated Storage .....	243			
13.3.2	第三方嵌入式数据库.....	243	<b>第 16 章</b>	<b>开机自启动.....</b>	<b>290</b>
13.3.3	SQLite Client for Windows Phone .....	244	16.1	Android 开机自启动.....	290
13.3.4	Windows Phone 中使用数据 库的注意事项.....	247	16.2	iOS 开机自启动 .....	291
			16.3	Windows Phone 开机自启动.....	291
<b>第 14 章</b>	<b>第三方库调用.....</b>	<b>248</b>	<b>第 17 章</b>	<b>日程与闹钟.....</b>	<b>292</b>
14.1	Android 第三方库调用.....	248	17.1	Android 日程与闹钟.....	292
14.1.1	Android 中创建第三方库.....	248	17.1.1	日程 .....	292
14.1.2	Android 中调用第三方库.....	249	17.1.2	闹钟 .....	296
14.2	iOS 第三方库调用.....	251	17.2	iOS 日程与闹钟 .....	298
14.2.1	iOS 中创建第三方库.....	251	17.3	Windows Phone 日程与闹钟.....	300
14.2.2	创建 Device 与 Simulator 通 用的 Static Library .....	254	<b>第 18 章</b>	<b>墙纸、屏保、Widget.....</b>	<b>301</b>
14.2.3	iOS 中调用第三方库.....	258	18.1	Android 墙纸、屏保、Widget.....	301
14.3	Windows Phone 第三方库调用.....	259	18.1.1	墙纸 .....	301
14.3.1	Windows Phone 中创建第三 方库 .....	259	18.1.2	屏保 .....	301
14.3.2	Windows Phone 中调用第三 方库 .....	261	18.1.3	Widget.....	305
			18.2	iOS 墙纸、屏保、Widget.....	308
			18.3	Windows Phone 墙纸、屏保、 Widget .....	308
<b>第 15 章</b>	<b>短信、电话、邮件.....</b>	<b>264</b>	<b>第 19 章</b>	<b>Timer 机制.....</b>	<b>309</b>
15.1	Android 短信、电话、邮件 .....	264	19.1	Android Timer 机制.....	309
15.1.1	Android 短信功能 .....	264	19.1.1	计划任务.....	309
15.1.2	Android 电话功能 .....	267	19.1.2	UI 更新 .....	310
15.1.3	Android 邮件功能 .....	272	19.2	iOS Timer 机制.....	312
15.2	iOS 短信、电话、邮件.....	272	19.2.1	NSTimer 简介 .....	312
15.2.1	iOS 短信功能.....	273	19.2.2	NSTimer 创建方法.....	312
15.2.2	iOS 电话功能.....	273	19.2.3	NSTimer 应用实例.....	312
15.2.3	iOS 邮件功能.....	273			
15.3	Windows Phone 短信、电话、邮件 .....	274			



19.3	Windows Phone Timer 机制.....	313	22.3	Windows Phone 定位技术 .....	350
19.3.1	DispatcherTimer 的使用方法.....	314	22.3.1	Windows Phone 定位技术基 础.....	350
19.3.2	XNA 中的 Timer .....	315	22.3.2	Windows Phone 定位技术实 例.....	351
19.3.3	以视频刷新频率为 Timer.....	315			
<b>第 20 章</b>	<b>XML 相关知识 .....</b>	<b>317</b>	<b>第 23 章</b>	<b>传感器.....</b>	<b>353</b>
20.1	Android 中 XML 相关知识 .....	317	23.1	Android 传感器.....	353
20.1.1	解析 JSON 数据.....	319	23.1.1	Android 传感器的类型.....	353
20.1.2	解析 XML 数据 .....	319	23.1.2	加速度传感器.....	355
20.2	iOS 中 XML 相关知识.....	322	23.1.3	其他传感器.....	357
20.2.1	XML 在 iOS 开发中的使用 .....	322	23.1.4	Android 中传感器的相关类 .....	357
20.2.2	String 在 iOS 开发中的使用.....	326	23.2	iOS 传感器.....	358
20.2.3	XML 和 String 之间的简单 转换.....	328	23.2.1	加速器 .....	358
20.3	Windows Phone 中 XML 相关知识 .....	329	23.2.2	三轴陀螺仪.....	359
20.3.1	XML 简介.....	329	23.3	Windows Phone 传感器 .....	359
20.3.2	Windows Phone 中字符串与 XML 的使用.....	329	23.3.1	加速器 API .....	359
			23.3.2	Windows Phone 传感器实例 分析 .....	360
<b>第 21 章</b>	<b>解压缩.....</b>	<b>332</b>	<b>第 24 章</b>	<b>动画 .....</b>	<b>362</b>
21.1	Android 解压缩.....	332	24.1	Android Animation.....	362
21.2	iOS 解压缩.....	336	24.1.1	动画类型.....	362
21.2.1	数据压缩与解压 .....	336	24.1.2	加载 XML 动画文件.....	368
21.2.2	通过第三方库解压缩.....	338	24.2	iOS Animation .....	368
21.3	Windows Phone 解压缩.....	338	24.2.1	UIView 页面切换动画 .....	368
			24.2.2	CATransition 动画 .....	369
<b>第 22 章</b>	<b>定位技术 .....</b>	<b>341</b>	24.2.3	CABasicAnimation 动画.....	371
22.1	Android 定位技术 .....	341	24.3	Windows Phone Animation.....	371
22.1.1	LocationManager 定位.....	341	24.3.1	变换、动画简介及分类.....	372
22.1.2	基站和 WiFi 定位 .....	342	24.3.2	基于定时器的动画.....	372
22.1.3	模拟经纬度坐标 .....	346	24.3.3	利用 ObjectAnimation 实现 动画 .....	374
22.2	iOS 定位技术.....	348	24.3.4	利用 ImageTool Gif 解码库 播放动画.....	378
22.2.1	CLLocationManager 类的使 用.....	348	24.3.5	页面切换动画 .....	380
22.2.2	CLLocationManagerDelegate 类的使用.....	349			



<b>第 25 章 权限、打包、签名及安装、卸载</b> .....	381
25.1 Android 权限、打包、签名及安装、卸载.....	381
25.1.1 Android 权限、打包、签名 .....	381
25.1.2 Android 安装、卸载 .....	387
25.2 iOS 权限、打包、签名及安装、卸载 .....	389
25.2.1 iOS 权限、打包、签名.....	389
25.2.2 iOS 安装、卸载.....	396
25.3 Windows Phone 权限、打包、签名及安装、卸载.....	399
25.3.1 Windows Phone 权限、打包、签名 .....	399
25.3.2 Windows Phone 安装、卸载 .....	400
<b>参考文献</b> .....	401

# 第 1 章 开发环境搭建

搭建开发环境是进行软件开发的前提,一般手机操作系统软件供应商都会为我们提供比较完备的继承开发环境和软件开发包,软件供应商也尽量降低环境搭建的难度;不过由于很多新手对相应的操作系统、集成开发环境、软件开发包不是很熟悉,搭建过程中难免会遇到各种各样的问题,因此,本章着重介绍一下如何进行环境的搭建,让大家在最短的时间内把准备工作做好。

## 1.1 Android 开发环境搭建

Android 开发环境的搭建在三大系统中算是比较复杂的,需要经过好几步配置。配置之前先来了解一下需要准备的软件及其下载地址。

### 1.1.1 相关软件准备

需要准备的相关软件及其下载地址如下:

(1) Java JDK

下载地址: <http://www.oracle.com/technetwork/java/javase/downloads/index.html>

(2) Eclipse

这里采用 Eclipse IDE for Java Developers

下载地址: <http://www.eclipse.org/downloads/>

(3) Android SDK 工具

这里采用 Android-sdk\_r10-windows.zip

下载地址: <http://developer.Android.com/sdk/>

### 1.1.2 安装与设置

所需的软件准备好后,接下来就可以开始一步一步搭建开发环境了。

**★提示** 开始前先确认当前系统用户名是否只由英文与数字组成,若用户名中含有中文或其他特殊符号,可能会导致模拟器安装或运行失败。

具体步骤如下:

**1** 安装好 JDK,并把 Eclipse 与 Android SDK 工具解压出来。建议把解压后的 eclipse 文件夹与 android sdk 文件夹放在同一目录下(注意路径名中不要含有中文),并在此目录下创建一个 workspace 文件夹备用,如图 1-1 所示。

**2** 运行 Eclipse,首次运行时会让用户指定一个默认的工作目录(这里选择步骤 2 中创建好的 workspace 文件夹)。

**3** 在 Eclipse 中选择菜单栏上的 Help→Install New Software

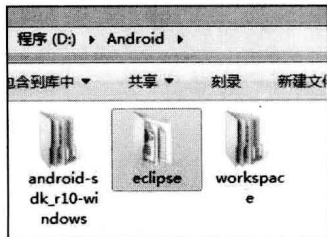


图 1-1



命令，在弹出的 Install 窗口（见图 1-2）中单击右边的【Add】按钮。

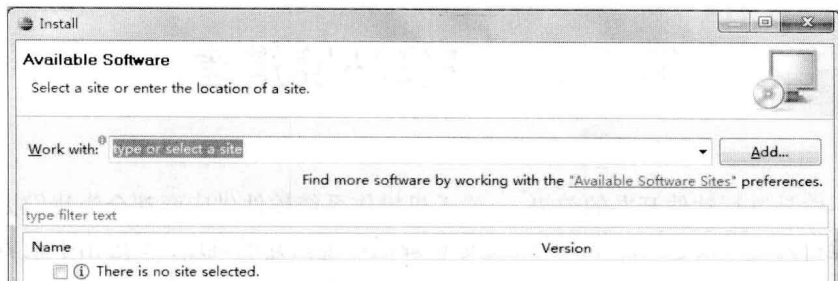


图 1-2

4 如图 1-3 所示，在弹出的 Add Repository 窗口的【Name】栏输入“ADT”，在【Location】栏输入“https://dl-ssl.google.com/android/eclipse/”，单击【OK】按钮。

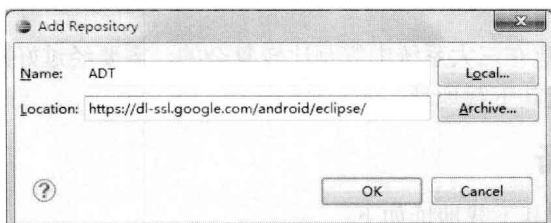


图 1-3

5 加载完毕后，把出现的“Developer Tools”复选框勾选上，如图 1-4 所示。然后依次单击【Next】按钮进入下一步操作。

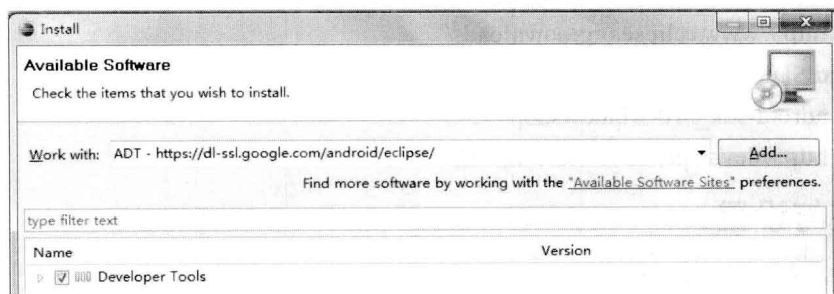


图 1-4

6 当出现提示许可询问时，选择“I accept the terms of the license agreements”并单击【Finish】按钮。

7 在安装 ADT 的过程中可能会弹出安全警告提示（见图 1-5），不用理会，直接单击【OK】按钮继续安装即可。

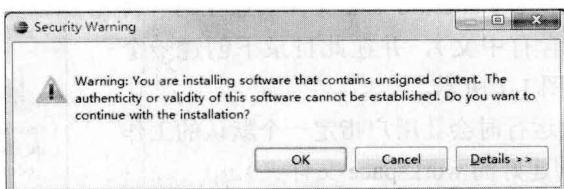


图 1-5



8 安装完毕后，会询问是否重启并生效，单击【Restart Now】按钮立即重启，如图 1-6 所示。

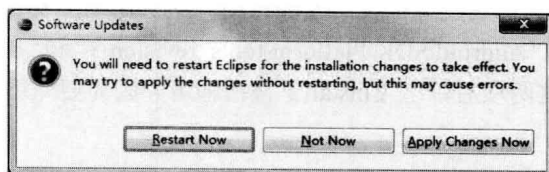


图 1-6

9 Eclipse 重启后，选择菜单命令 Windows→Preferences，进入图 1-7 所示的窗口，选中左边的【Android】，然后在右边的“SDK Location”一栏单击【Browse】按钮，选择之前解压出来的 Android SDK 工具文件夹，单击【OK】按钮。

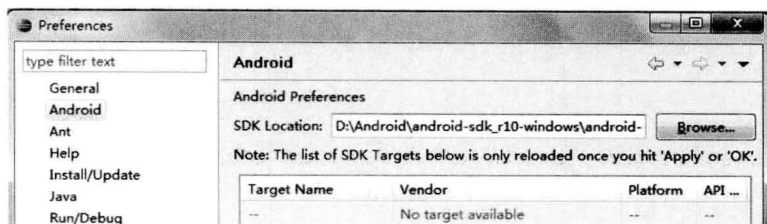


图 1-7

10 弹出一个 Android SDK Verification 窗口（见图 1-8），提示 SDK 平台工具组件丢失，没关系，这是因为还没有安装，直接单击【OK】按钮即可。

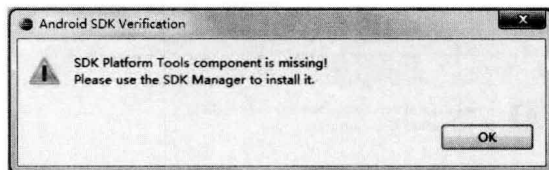


图 1-8

11 安装 SDK，选择菜单栏的 Window→Android SDK and AVD Manager 命令，打开 Android SDK AVD 管理器，先选中左边的“Install packages”选项，如图 1-9 所示。

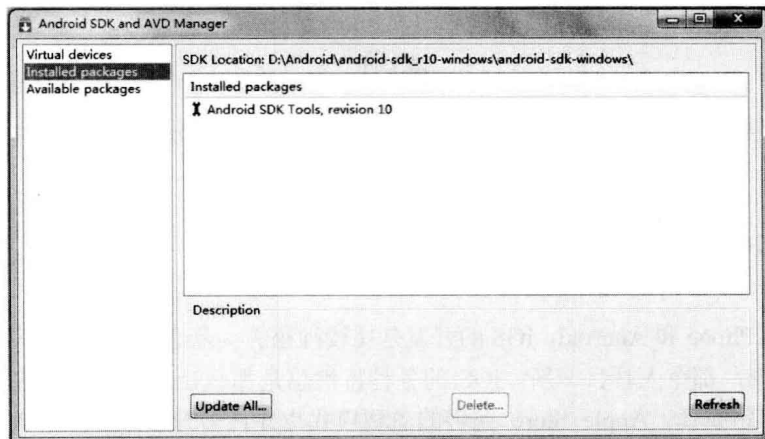


图 1-9

