

Android™

Android程序员指南

原著: Jerome (J.F.) DiMarzio

译: 张劲锋

› 创建移动设备应用程序

› 精通Eclipse开发环境和Android SDK

› 在Android 模拟器上运行及调试 应用程序

源码实例在网上获取

Mc
Graw
Hill



Android™

程序员指南

原著: *J. F. DiMarzio*

翻译: 张劲锋

zyf19870302@126.com

仅供学习使用
版权归属原著作者



New York Chicago San Francisco

Lisbon London Madrid Mexico City

Milan New Delhi San Juan

Seoul Singapore Sydney Toronto

版权©2008 McGraw-Hill 公司. 保留所有权. U.S 出品. 除被 U.S 1976 版权法令许可之外, 此出版物的任何部分被禁止用任何形式或者电子书店以及复印组织, 没有被原著授予发行权的出版社以任何形式复制、散布、发行。

0-07-129989-4

本电子书的资料也标注有 0-07-129989-4 的版本标题。

所有商标归它们各自的所有者。而不是在任何交易发生时放一个交易名称的商标符号, 我们只是以编辑形式使用了名称, 为了商标所有者的利益, 没有对商标的侵害意图。在本书出现该称号的地方, 它们已经被印刷成首标注。

McGraw-Hill 电子书能以特殊数量的折扣作为奖品和增进促销或者公司培训项目来获得、使用。要获得更多信息, 请联系 George Hoare, Special Sales, 通过 george-hoare@mcgraw-hill.com 或者 (212) 904-4069。

TERMS OF USE

This is a copyrighted work and The McGraw-Hill Companies, Inc. ("McGraw-Hill") and its licensors reserve all rights in and to the work. Use of this work is subject to these terms. Except as permitted under the Copyright Act of 1976 and the right to store and retrieve one copy of the work, you may not decompile, disassemble, reverse engineer, reproduce, modify, create derivative works based upon, transmit, distribute, disseminate, sell, publish or sublicense the work or any part of it without McGraw-Hill' s prior consent. You may use the work for your own noncommercial and personal use; any other use of the work is strictly prohibited. Your right to use the work may be terminated if you fail to comply with these terms.

THE WORK IS PROVIDED "AS IS." McGRAW-HILL AND ITS LICENSORS MAKE NO GUARANTEES OR WARRANTIES AS TO THE ACCURACY, ADEQUACY OR COMPLETENESS OF OR RESULTS TO BE OBTAINED FROM USING THE WORK, INCLUDING ANY INFORMATION THAT CAN BE ACCESSED THROUGH THE WORK VIA HYPERLINK OR OTHERWISE, AND EXPRESSLY DISCLAIM ANY WARRANTY, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. McGraw-Hill and its licensors do not warrant or guarantee that the functions contained in

版权©2008 McGraw-Hill 公司



the work will meet your requirements or that its operation will be uninterrupted or error free. Neither McGraw-Hill nor its licensors shall be liable to you or anyone else for any inaccuracy, error or omission, regardless of cause, in the work or for any damages resulting therefrom. McGraw-Hill has no responsibility for the content of any information accessed through the work. Under no circumstances shall McGraw-Hill and/or its licensors be liable for any indirect, incidental, special, punitive, consequential or similar damages that result from the use of or inability to use the work, even if any of them has been advised of the possibility of such damages. This limitation of liability shall apply to any claim or cause whatsoever whether such claim or cause arises in contract, tort or otherwise.

DOI: 10.1036/0071599886



本书贡献于 *Suzannah, Christian, and Sophia*

关于作者:

J. F. DiMarzio 是一个在网络和程序开发领域拥有超过 15 年经验的程序开发人员并且是 7 本关于计算机科学书籍的作者。他已经成为 IT 界顾问和发展的领导人物。他住在 Central Florida.

关于技术编辑:

Gilbert L. Polo 是一个在电信, 财政, 最近最多是教育机构工作上拥有超过 20 年经验的程序开发人员。他已经用多种语言 (包括 C, C++, Java 和 C#) 项目编程。

目录浏览

1. 什么是 Android?	1
2. 下载和安装 Eclipse.....	9
3. 下载和安装 Android SDK.....	21
4. 浏览 Android SDK.....	35
5. 应用程序: Hello World!	53
6. 使用命令行工具和 Android 模拟器.....	83
7. 使用 Intents 和 Phone Dialer.....	117
8. Lists, Menus 和其他 View.....	149
9. 使用 Cell Phone 的 GPS 功能.....	203
10. 和 GTalk 一起使用 Google API.....	239
11. 应用程序: Find a Friend.....	257
12. Android SDK 工具参考.....	307
索引.....	313



目录

致谢.....	xiii
介绍.....	Xv
1. 什么是 Android?	1
嵌入式设备开发项目简要历程.....	2
开放手机联盟和 Android.....	5
Android 介绍.....	6
2. 下载和安装 Eclipse.....	9
为什么使用 Eclipse.....	11
下载和安装 JRE.....	12
下载和安装 Eclipse.....	18
3. 下载和安装 Android SDK.....	21
下载 Android SDK.....	24
下载和安装 Android Eclipse 插件.....	30
配置 Android Eclipse 插件.....	18
4. 浏览 Android SDK.....	35
Android SDK 中有什么.....	37
Android 文档.....	38
尝试: 运行 APIDemo 例子程序.....	41
Android 工具.....	45
APIS.....	48
应用程序生命周期.....	49
标准 ASP 应用程序生命周期.....	49
Android 应用程序生命周期.....	50
5. 应用程序: Hello World!	53

在 Eclipse 中创建你的第一个 Android 项目	55
细述 Android-生成文件	61
AndroidManifest.xml	62
参考 运行库	62
说明	63
再次 Hello world!	69
使用 Image Hello world!	72
Hello World! Code-Based UI	75
Hello World! XML-Based UI	78
尝试: 使用 TextView 和 ImageView	81
6. 使用命令行工具和 Android 模拟器	83
使用 Windows CLI 创建一个 Shell Activity	84
运行 ActivityCreator.Bat	85
工程结构	88
在 Windows CLI 中创建 HelloWorld! Activity	95
编辑工程文件	95
添加 JAVA_HOME 变量	96
编译和安装应用程序	97
HelloWorld! 在 Linux 中	109
配置 PATH 声明	109
尝试: 在 CLI 下创建 Image-Based Helloworld!	115
7. 使用 Intents 和 Phone Dialer	117
什么是 Intents	119
使用 Dialer	124
从你的 Activity 调用 Call	128
为你的 Activity 添加 Intent	129
编辑 Activity 许可	131
修改 AndroidPhoneDialer	136
添加 Button	136
实现一个 EditTextView	141
尝试: 修改 AndroidPhoneDialer 项目	145
8. Lists, Menus 和其他 View	149
构造 Activities	151
Intent 代码 .xml 文件	152
Intent 代码 .java 文件	154
修改 AndroidManifest.xml	155
使用 Menu	157
为 AutoComplete 创建 Activity	163
Button	173
CheckBox	178



EditText.....	183
RadioGroup.....	189
Spinner.....	195
尝试: 修改更多 View 属性.....	202

9. 使用 Cell Phone 的 GPS 功能 203

使用 Android Location-Based API.....	204
创建一个 kml 文件.....	205
track 文件是什么.....	208
在 Windows 下得到 namea 文件.....	208
在 Linux 下得到 namea 文件.....	210
阅读 GPS Android Location-Based API.....	212
创建 AndroidLBS Activity.....	212
向 Google Maps 传递坐标.....	222
添加图像放大控制.....	226
尝试: 联系 MapView 的标准和 Statellite Views.....	232

10. 和 GTalk 一起使用 Google API 239

为 GTalk 配置 Android 模拟器.....	241
在 Android 中实现 GTalk.....	244
在 GoogleAPI.xml 中创建 Activity 的 Layout	245
向 GoogleAPI.Java 中添加包.....	247
实现 View.OnClickListener.....	248
编译和运行 GoogleAPI.....	252
尝试: 给你的 GoogleAPI Activity 添加一个 Setting Feature.....	255

11. 应用程序: Find a Friend..... 257

创建一个 SQLite 数据库.....	259
创建一个 Custom Content Provider.....	263
编辑 strings.xml.....	263
创建你的 Content Provider.....	265
创建 FindAFriend Activity.....	276
编辑 AndroidManifest.xml.....	276
创建 NameEditor Activity.....	278
创建 LocationEditor Activity.....	283
创建 FriendsMap Activity.....	293
创建 FindAFriend Activity.....	299
运行 FindAFriend Activity.....	302
尝试: 实时本地更新.....	305

12. Android SDK 工具参考 307

Android 模拟器命令集.....	308
---------------------	-----



Android Bebug Bridge 命令集.....	310
13. 索引.....	313

致谢:



我要感谢参与制作这本书的每个人，我的助理，Neil Salkind; Roger, Carly, Janet, Bill, 和 McGraw-Hill 全体人员; Gil Polo; 和 Studio B 的每个人。

我也想感谢我的家人，Suzannah, Christian, and Sophia; Brett, Robert, Roger, Zack, Mark, Kurt, Walter, Walter, Walter, Steve, Steve, Steve, 和 Gary—还有我所有在 Central Florida 的同事，还有我忘记提到的任何一个人。

介绍:



欢迎来到 Android: 程序员指南。本书专门为你通向新的、令人兴奋的开源移动开发世界的最好的第一步做了设计。Android 是最新的移动设备操作系统，并且这是帮助普通程序员转变为专业无阻的 Android 开发者的第一手书籍的其中一部。通过本书的课程，会向你介绍使用开源手机联盟的 Android 平台最基本的移动设备应用程序开发。到了本书的最后，你将能够自信的创建你自己的移动设备项目。

本书的格式将会以逻辑性的风格带你走遍 Android 应用程序开发。本书开始通过检查 Android 作为一个平台的构架，看看它是怎么被开发的，它能在什么上运行，和需要什么样的工具来为 Android 开发项目。在讨论和安装开发工具、Android SDK 和 Eclipse 开发环境（第 2, 3, 4 章节）之后，本书直接跳入设计和创建 Android 应用程序（第 5 章节）。本书包括在现有的 Google 工具（例如：Google Maps（第 9 和 11 章节）、Gtalk（第 10 章节））上尝试你自己的应用程序的教学指导。也包括一个快速参考指导（第 12 章节）。

这本书是一本程序员指南，不是入门指南，意思是，你的却需要拥有一些编程技术通过本书来获得更过技术。在这些技术中最重要的是基本的 java 编程知识。Android 应用程序是用 java 语言开发，并且运行在 Linux2.6 内核上。如果你是一个快速的学习者，你可能明白什么是在一些最基本的目标性经验下进行开发。第 2 章节解析怎样下载和安装受偏爱的，综合的开发环境-Eclipse。本书中所有的代码实例和截屏是用 Eclipse (Europe 版本) 和 Android Eclipse 插件提供的。

有任何建议，疑问，或者意见关于本书的任何资源，可以直接联系作者 jfdimarzio@jfdimarzio.com。

第一章

什么是Android?

关键技术&概念

- 嵌入式设备编程项目历史
 - 解释开放手机联盟
 - 第一次观看 Android 主屏幕
-

能够说，传统的桌面应用程序开发者已经很受宠了。这不是说传统桌面应用程序开发比其他形式的开发更简单。然而，作为一个传统桌面应用程序开发者，我们已经有创建所有我们能够想象的任何一种应用程序的能力。我也指包括我在内，因为我以前就开始桌面项目开发。

一方面，我们已经有能力去和桌面操作系统互动已经使桌面编程开发更易接受，这样的互动是和任何底层的硬件的。相当自由（或者至少有很小的例外）。这种自由、独立地编程，已经不再是那些敢于冒险跳进手机开发黑水中的少数量的开发者能够获得的。

注意：

在这个讨论中我提及到了两种不同的开发者：传统桌面应用程序开发者，他们几乎用所有的编程语言来工作，他们的最终产品、应用程序被构建能运行在任何一种桌面操作系统上；Android 开发者（Java 开发者），为 Android 平台开发。这不是有意图说哪一种开发者在某种意义上比另一种更好或者更坏。然而，这个讨论意图在于：桌面操作系统环境和移动操作系统环境 Android 的开发形式和开发工具的比较。

简短的嵌入式设备开发历史

很长一段时间，手机开发者构成了一个认作是嵌入式设备开发者的小的稍微大点的派别。理解成一个小的具有魅力的桌面家族成员-后来的网络 Web-开发，嵌入式设备开发代表性地得到闻名的短的结局，远至硬件和操作系统特色，因为嵌入式设备制造业在特色支持方面恶名昭彰地吝啬。嵌入式设备制造厂商典型地需要封闭地保护他们硬件的机密，因此，当嵌入式设备开发者试着去与特殊的设备互动时嵌入式设备制造厂商给他们很少的函数库去调用。

嵌入式设备不同于桌面系统是一个嵌入式设备是一个典型的“片上计算机”。例如：考虑你的标准电视机的远程控制。它不是真正看上去像一个无法实现的复杂技术的实现。当任何一个按钮按下，芯片以一种方法解释信号（已经编程嵌入到设备中）。这允许设备从输入设备（键盘）知道用户期望干什么，和如何对这些命令响应（例如，打开电视机）。这是一个来自嵌入式设备的编程例子。



然而，信或不信，诸如这些简单的设备被明确涉及到早期手机设备和开发的根源。

大多数嵌入式设备运行（在一些情况仍然运行）在专有的操作系统上。选择创建一个专有的操作系统而不是使用任何消费者系统的理由是：这是一个产品真正必需的。简单的设备不需要很耐用、完善的操作系统。

作为一种设备产品的发展，大约2年前左右，许多复杂的嵌入式设备，例如早期的 PDAs，家庭安全系统，和 GPSs，被移植到有点标准化的操作系统平台上。小体积的操作系统如 Linux，或甚至嵌入式版本的 Microsoft Windows，已经是在许多嵌入式设备上很流行了。围绕这个时间段的设备发展，手机从其他的嵌入式设备分支成他们自己的路线。当你查看他们的业绩时，这个分支显得很明显。

几乎从他们起初，当运行专有的软件（被制造厂商拥有和控制，几乎也总是被考虑是“封闭”系统）时，手机就已经成为终端设备的范围内。制造厂商使用专有的操作系统的实践开始比其他任何理由更加不必要。这是因为手机制造商代表性的使用完全在内部开发出的硬件。或至少硬件是为运行手机设备特别开发出来的，结果，这些硬件、专有系统不是开发地可获得的，空架的软件包，或者解答（这些都是能够确切与他们硬件互动的）。自制造商想更可靠地保护他们的硬件交易机密（通过允许接近设备软件水平可以显露出来），通常的实践是（在多数情况下仍然是）通过完全专用的和封闭的软件来运行他们的设备。接下来就是，任何想为手机开发能够在上面运行的应用程序的人不得不去揭秘专有系统环境。解决方法是直接向制造商购买昂贵的开发工具。这孤立了很多“业余爱好”的开发者们。

注意：

一种成长的业余爱好开发者文化已经拥抱着手机应用程序开发。“业余爱好者”队伍事实涉及到了那些具有代表性的、不是为手机开发公司工作的开发者和普遍地小生产，利用他们自己时间开发产品的人。

其他的，更多强制的“必需品”是硬件制造商对于“内存对照需求”困境的解答（这使天天进行手机开发的开发者难以入手）。直到最近，手机不是比实现多一点点，接收电话，跟踪你的联系，并且可能的发送和接收文本短信；当今他们不是真正的“瑞士军刀”技术。甚至迟到2002年，带有照相机的手机在消费者的手上一般的也没有见到。

到1997年，小应用程序例如计算器和游戏（俄罗斯方块，举例）缓慢地在手机上爬行他们的发展路线，但是这些无法实现的功能仍然是电话拨号程序本身。手机还没有成为今天它们的多样的，多功能的个人工具。这是可能的，1997年，手机制造商完全没有意识到消费者需求是所有功能集聚一个设备上。不过，即使这个需求是目前的，设备的内存缺乏，存储能力是一个更大的要去克服的障碍。更多的人可能已经想过他们的设备变成聚集所有功能为一身的工具，但是制造商仍然不得不攀爬内存障碍。

为了使问题简单，这使得任何设备上让内存去存储和运行应用程序，包括手机。手机，作为一种设备，直到最近都还没有使手机容易包含额外程序的大量内存。



在最近两年里，内存价格已经达到很低水平了。设备制造商现在有能力以更低的价格包含更多的内存。现在很多手机拥有比 1990 世纪中期普通 PC 机更标准的内存。因此，现在我们有需要、内存，我们可以深入，并且在世界范围里为手机开发酷的应用程序，对不？不是确切的。

设备制造商仍然封闭地保护运行在他们设备上的操作系统。而一些对这一点已经开放，他们允许一些以 Java 为基础的应用程序在手机上的小环境里运行，许多的都不允许这样做。甚至某些不允许运行一些 Java 应用程序的系统不允许接近标准桌面开发者习惯去拥有的核心系统。

开放手机联盟和 Android



在 2007 年 11 月，当 Google，在开放手机联盟下，发布 Android，这个对于应用程序开发者的障碍开始消失。开放手机联盟是一个硬件和软件开发者组织，它包括 Google、NTT DoCoMo, Sprint Nextel, 和 HTC，它的目标是创建一个更开放的手机环境。在开放手机联盟组织下发布的第一个产品是 Android-移动设备操作系统。

(获得更多关于开放手机联盟的信息，登陆 www.openhandsetalliance.com) 和 Android 发行一起，Google 开发了可获得的主要开发工具和指南去帮助要成为新系统开发者的人。帮助文档、平台软件开发包 (SDK) 和甚至在 Google 的 Android 网站上能被找到的一个开发者社区，<http://code.google.com/android>。这个网站应该是你的入门点，我十分鼓励你去访问该网站。

注意:

Google，为促进新的 Android 操作系统，甚至开始了举办了一个 1000 万的竞赛，寻求新的、令人兴奋地 Android 应用程序。

而运行在 Linux、Windows、甚至 PalmOS 上的手机能轻易找到，到写此书为止，没有硬件平台被宣布专门为 Android 所运行。HTC、LG 电子、Motorola 和 Samsung 都是开放手机联盟的成员，在 Android 已经被发布之下，我们只希望他们为 Android-based 设备在未来有计划实行。在它 2007 年 11 月发行，这系统本身仍然只是一个测试版软件。这对于开发者是一个好消息，因为它给我们一个展望未来系统珍贵的进步发展和一个开始开发将会很快在被发布的硬件上运行的应用程序的机会。

注意:

这个策略清楚地给开放手机联盟一个大优势超过别的手机操作系统开发者，因为对于第一个发布的运行 Android 的设备将会有无数的应用程序可以直接获得。

介绍: Android

Android作为一个系统,是一个运行在Linux2.6内核上的基于Java的操作系统.这个系统非常轻量级的,并且功能丰富.图1-1展示未修改的Android主屏幕.拥有

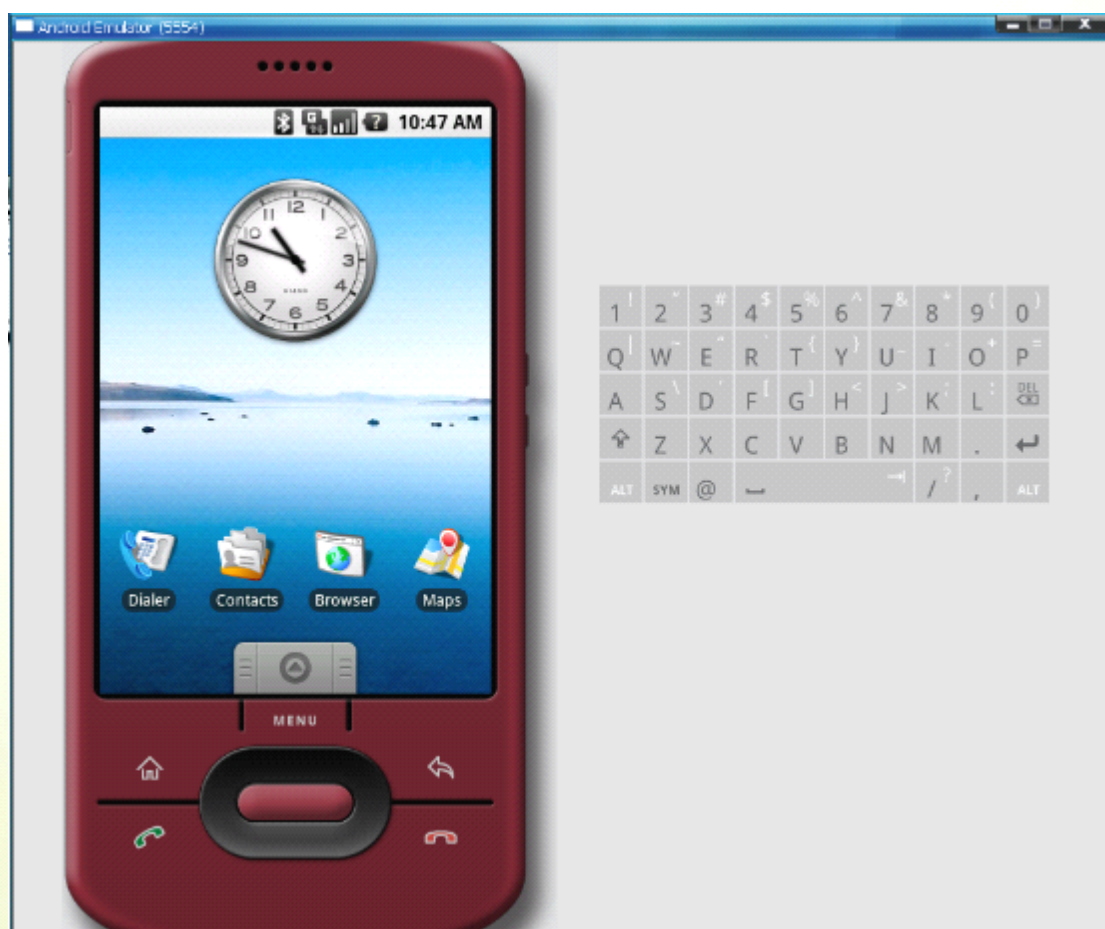


图 1-1 目前版本 Android 主屏幕, 模拟器

Android 应用程序是用 Java 开发的,能够十分简单的装入新的平台。如果你还没有下载 Java 或者不确信需要哪个版本,我在第二章节详述安装开发环境。Android 的其他特色包括加速 3D 绘图引擎(依赖于硬件支持),SQLite 强力支持的数据库,和一个综合完整的 Web 浏览器。

如果你熟练 Java 编程或者是一个任何种类的 OOP 开发者,你可能用过用户界面 (UI) 开发,那是 UI 放置(直接在项目源码中运用的)。Android 能识别和允许为项目的 UI 做开发,也支持更新的,基于 XML 的 UI 编排设计。XML UI 编排设计对于一般桌面开发者是一个相当新的概念。在本书支持的章节里我将涵盖 XML UI Layout 和项目的 UI 开发。

Android 一个最令人兴奋和瞩目的特色是它的架构，三部分运用程序-包括“home-grown”当这些和内核系统绑定，可带有同样的系统优先权执行。这是和大多系统一个主要的违背，它给了嵌入式系统应用程序一个比进程优先级（从第三方开发者创建的应用程序获得）更大的执行权。每一个应用程序在它自己的进程里使用一个很轻量级的虚拟机执行。

除了我们能够获得用来开发的很大的 SDK 和良好的架构库，对于 Android 开发者最令人兴奋的特色是我们现在允许进入任何操作系统所允许的。换句话说，如果你要创建一个拨电话的应用程序，你可接近电话拨号程序，如果你想创建一个利用电话的内在 GPS 的应用程序（如果配备），你可以接近它。对于开发者创建动态的和令人感兴趣的应用程序的可能性现在是宽阔开放了。

在这些特色中最顶级的是从 Android 网站可获得的平衡，Google 已经投放一些它自己非常逗弄的特色。Android 应用程序开发者将能够去把自己的应用程序连接到 Google 提供的已经存在的程序（例如 GoogleMaps、不所不在的 Google 搜索）上。假设你想写一个获取一个打进电话的发出地的 Google 地图，或者你想能够存储你联系的一般搜索结果；利用 Android 可能性的大门已经成为直冲的开放和宽阔。

第二章将开始你的 Android 开发之旅。你将学到怎样和为什么使用特殊的开发环境或综合完整的开发环境(IDE)，并且你将会下载和安装 JavaIDE Eclipse。



专家咨询

Q: Google 和开放手机联盟的区别是什么？

A: Google 是开放手机联盟的成员之一。在 Google 收购原来的 Android 的开发商，在开放手机联盟组织下发布了这个操作系统。

Q: 是否 Android 有能力运行任何 Linux 软件？

A: 不是必需的。我确认将会有方法打破许多任何开源的系统，应用程序需要被编译后利用 Android SDK 在 Android 上运行。主要原因是 Android 应用程序执行文件是一种特殊的格式；这将在后面章节讨论。

第二章

下载和安装Eclipse

关键技术&概念

- 选择一个开发环境
- 下载 Eclipse
- 安装和配置 Eclipse

.....

Android 应用程序是用 Java 开发的。Android 本身不是一种编程语言，而是一个运行应用程序的环境。如此，你可以理论上使用任何你自己支配的配发的或是综合完整的开发工具（IDE）来开始你的开发。实际上，你也可以不选择使用 IDE。

贴士

在本书后面章节，我将给你介绍不用 Android 开发环境或是命令行界面（CLI）来开发 Android 应用程序。然而，我不会使用这个技术来涵盖本书的每个例子，你将获得怎样在 CLI 中开发的基本技能。

如果你使用的一个 Java IDE 更舒适（超过别的任何一个），例如：Borlan JBuilder 或是开源的 NetBeans，自由地使用它。用一种中等的经验水平，你应该仍然能够应用这本书的主要例子跟随本书学习下去。然而，开放手机联盟和 Google 都支持赞同一个 Java IDE 胜过其他的任何一个，它是：Eclipse。

注意：

如果你选择跟随这本书的例子，而不使用 Eclipse，你需要检查你的 IDE 的文件来编译和调试你的 Android 应用程序。本书的例子只是在 Eclipse 环境、Android Eclipse 插件中给了相关的编译和调试的说明。

本章节简明的描述了下载和安装 Eclipse 和必需的 Java 运行环境（JRE）。多的时间，安装指导和用户手册，按自己需求不是缺乏更多技术的技术在此之上的读者可以跳过像这种简单的步骤。我已经建立了那些文本来跳过这些更简单的步骤，或者是查看重要的条目。由于这个原因，本章我将要介绍所有的下载和安装步骤。

为何使用 Eclipse?

为什么 Eclipse 是推荐的 Android 应用程序 IDE? 这有几个对这个特别支持的理由:



- 为了符合开放手机联盟真正开发手机开放市场的主题, Eclipse 是一个最完全该特色、自由、可获得的 Java IDEs 中的一个。Eclipse 也使用非常简单, 最小的学习弯路。这使得 Eclipse 成为一个非常吸引人的坚固的开源 Java 开发 IDE。

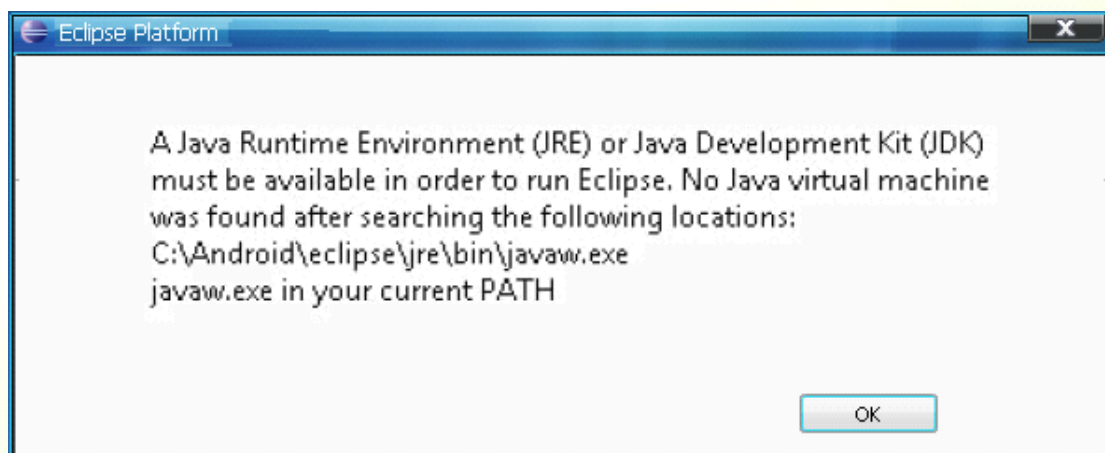
- 开放手机联盟已经发布了一个 Android Eclipse 插件, 它允许你创建 Android 特性的程序项目, 编译它们, 和使用 Android 模拟器来运行和调试它们。当你创建你的第一个 Android 应用程序时, 这些工具和功能将显示出无比的价值。你也能够用其他的 IDEs 来创建 Android 应用程序, 但是 Android Eclipse 插件为你创建必要的安装元素, 比如文件和编译设置。Android Eclipse 插件提供的帮助节省你宝贵的开发时间和巨大的减少学习弯路, 意思就是你可以花更多时间来创建无法想象的应用程序。

注意:

Eclipse 也可以为 Mac 和 Linux 定制获得。具有更大的可用性, 在很多数的操作系统上。意思是几乎任何一个人在任何一台计算机上都能开发 Android 应用程序。而本书的例子和截屏是来自 Microsoft Windows 下的 Eclipse。如果你不是在任何一种 Microsoft 环境下使用 Eclipse, 请记住这点。你的界面会明显的和本书截屏不同, 但是全部的功能应是不变的。假如在 Linux 下有主要的操作变化, 我将会为这个变动包含一个例子。我也会提供一些在 Linux 环境下的例子。主要的这些例子会来自 Linux/Android 命令行环境。

下载和安装 JRE

在你下载和安装 Eclipse 之前，你不得不确保你有了 Java 运行环境（JRE）的下载和安装到你的机器上。因为 Eclipse 是用 java 编写的应用程序，它需要 JRE 来运行。如果 JRE 没有安装，或是不能被它发现，如果你试着打开 Eclipse 开发环境，你将会看到下面的错误：



如果你是一个现行的 java 开发者并且已经安装了 java 在你的电脑上，你将仍然想照这里来进行下去，只要保证你安装的 JRE 的正确版本。

注意：

很多用过网络或是基于网络的应用程序的用户也装过 JRE。JRE 允许你运行基于 java 的应用程序，但它不支持创建 java 程序。要创建 java 程序，你需要下载和安装 java 开发工具包 JDK，它包括了创建 java 应用程序需要的所有工具和类库。假如你不熟悉 java，记住这些区别。对于本书例子，我将下载 JDK，因为它也包含 JRE。即使你不需要 JDK 来运行 Eclipse，你也可以用它来开发本书后面其他的开发。

像下面图中所示，导航进入 Sun 公司开发网（SDN）下载页面：
<http://developers.sun.com/downloads/>，通常你只需要 JRE 来运行 Eclipse，但是为了符合本书目的，你应该下载完整的 JDK，它包含了 JRE。下载完整 JDK 的理由是，在本书中我也将举一些怎样用 JDK 的工具在 Eclipse 之外来创建 Android 应用程序的例子。如果你要跟着这本指导书走的话，你需要完整的 JDK。



Sun Microsystems - Sun Developer Network (SDN) - Internet Explorer provided by Dell

http://developers.sun.com/downloads/

Sun Microsystems - Sun Developer Network (SDN)

Sun Java Solaris Communities My SDN Account Join SDN

Sun Developer Network (SDN)

search tips Search

Developers Home >

Downloads

Download. Develop. Deploy.
Take advantage of the power of Sun's tools and technologies today.

All Downloads New Downloads Top Downloads

From enterprise software to developer tools, Sun offers a comprehensive suite of products that help to create solutions and increase productivity. The latest releases are available for download below.

To see the list of all downloadable software from Sun, please visit the Sun Download Center.

More Info

- Development Tools
 - Sun Studio Compilers & Tools
 - Java Studio Creator
 - Java Studio Enterprise
 - NetBeans and Add-on Packs
- Java EE
 - Java ME
 - Java GC
- Database
- Infrastructure Software
 - Identity Management
 - Web Server
 - Directory Server
 - Portal Server
 - Web Proxy Server
 - Sun Java System Active Server Pages
 - Application Server
- Networking Technologies
 - Solaris

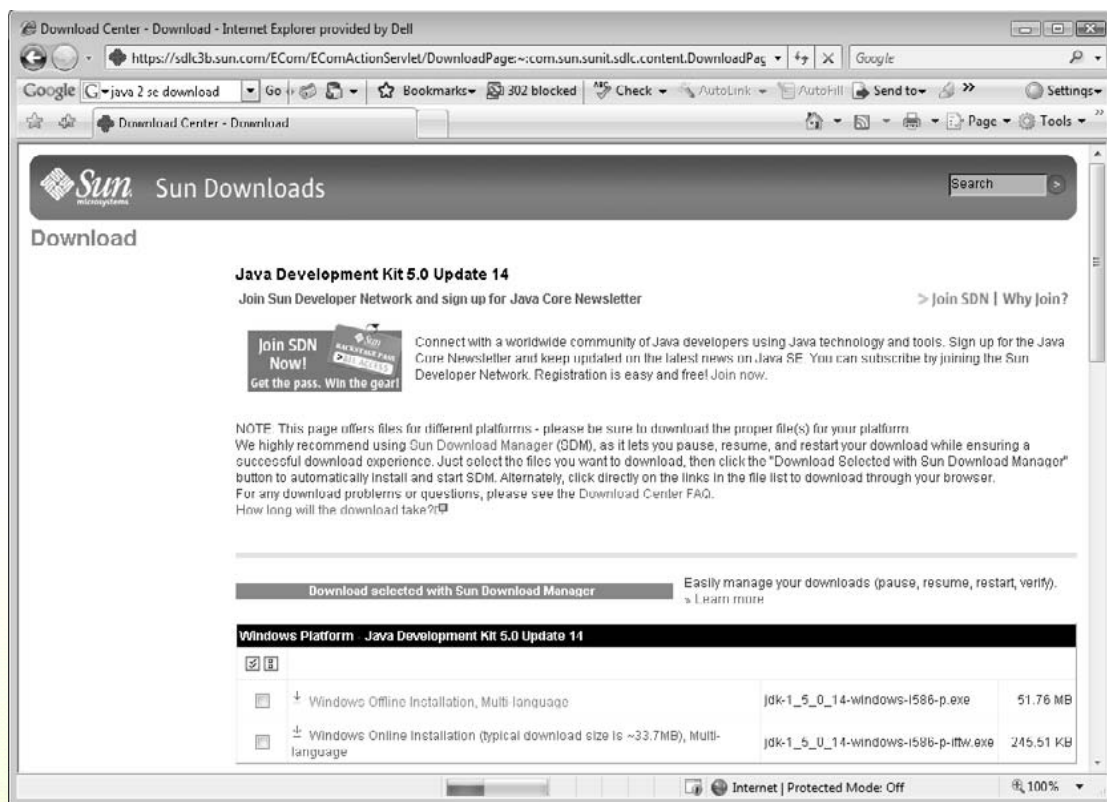
Vote Total
43

SDN SHARE - Vote, Post, Comment, Earn rewards.
Participate Now

Getting Started

- New to Java Center
- New to Solaris Center
- Java Studio Creator

从 SDN 下载页面导航进入合适的 JDK 的下载区。像下图所示，选择并开始下载。



由于本书例子，我选择了用 Java5 JDK pdate 14 来进行。因为它是在 Eclipse 文档中被明确定义的被支持的 Java 版本。要下载 JDK5，选择你想要下载的平台。假如你选择了 JDK6 的话，你也能够轻松地跟着学习下去。然而，假如你要下载更老版本的 JDK5，你需要向下面显示的那样点击以前版本的链接。



注意:

在你开始下载之前，你必须同意并接受本页中 Sun 公司的许可协议。

在 JavaSE 以前版本下载页面上，点击 J2SE5.0 下载链接，接下来点击 JDK5.0 Update x 的下载按钮，这里的 x 是最新的更新号码（14 是写这本书时候的，但和你读这本书时候的可能不一样）。

假如你正在下载到 Microsoft Windows 环境下, 当你看到下图的提示, 点击 Run 来开始安装 JDK。



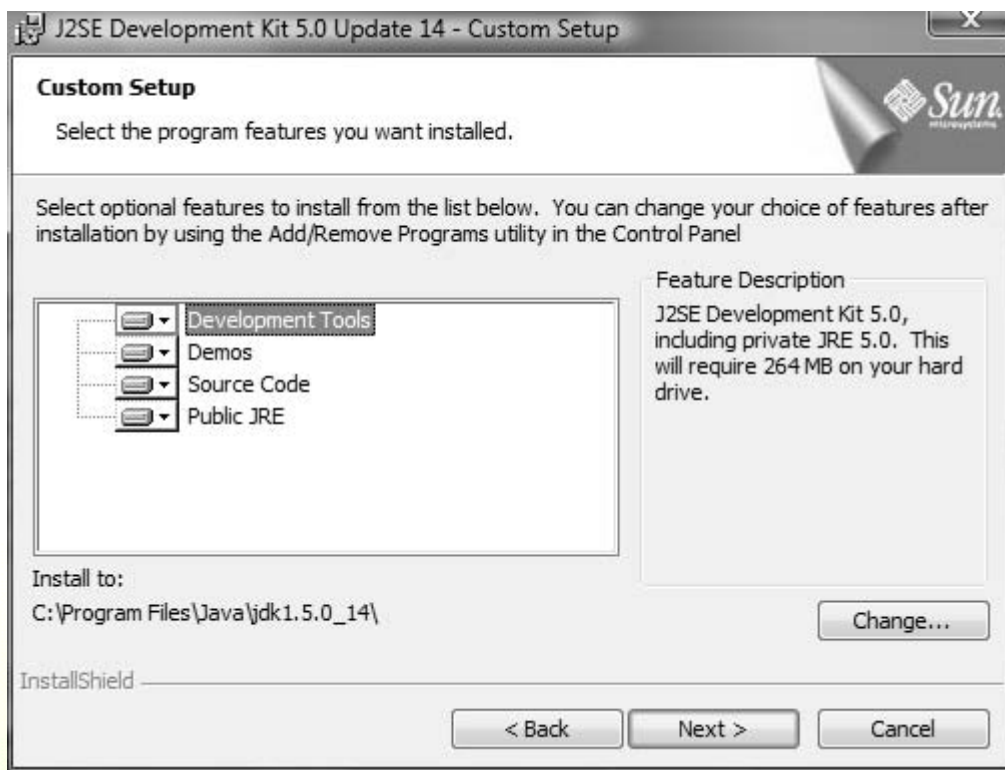
警告:

如果你要保留 JDK 的安装包的复本, 请点击 Save, 而不是 Run。然而, 假设你选择了下载 JDK, 请确保你记得下载到哪里。下载完之后, 你需要导航进入下载目录手工执行安装包。

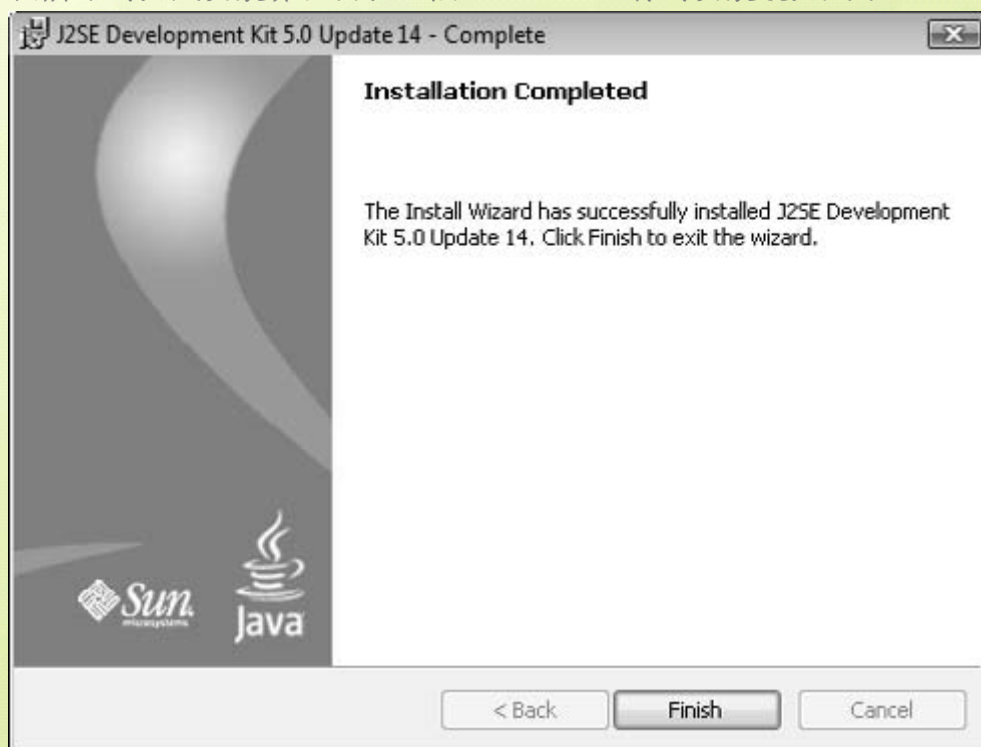
在安装期间, 你会被催促去阅读并且接受下面所示的许可协议。在同意这标准协议后点击 Next, 你就能够选择你的定制安装选项。



这里只有很少的地方需要改动，除非你是一个经验很丰富的 Java 老手并有让你舒适而自由选择的一些你想要的特殊选择。下图显示了 Java JDK 的定制安装界面。



为了让过程简洁和十分标准化，你应该接受推荐的包（默认每个都被选择），通过点击 Next 来继续下一步安装。再强调，如果特殊的更改让你舒适的话，尽管做就是了。而如果你在后面章节遇到麻烦，你会想改变你的安装选项。当下图所示的完成安装界面出现，点击 Finish，你的安装就完成了。



一旦你完成了 Java JDK 的安装-默认的 JRE 也被安装了-你可以开始安装 Eclipse 了。

下载和安装 Eclipse

导航进入下图所示的 Eclipse 下载页面: www.eclipse.org/downloads。作为开篇,之前的篇幅你就注意到了,在 Eclipse 开发需要 JRE(推荐 Java 5 JRE)。从此网站下载 Java 开发者专用的 Eclipse IDE。安装包十分小(79M),并且下载也会相当快。确保你不要下载 Java EE 开发者专用的 Eclipse IDE,因为这个 IDE 稍微有点不一样,我也不会涉及到它的使用。



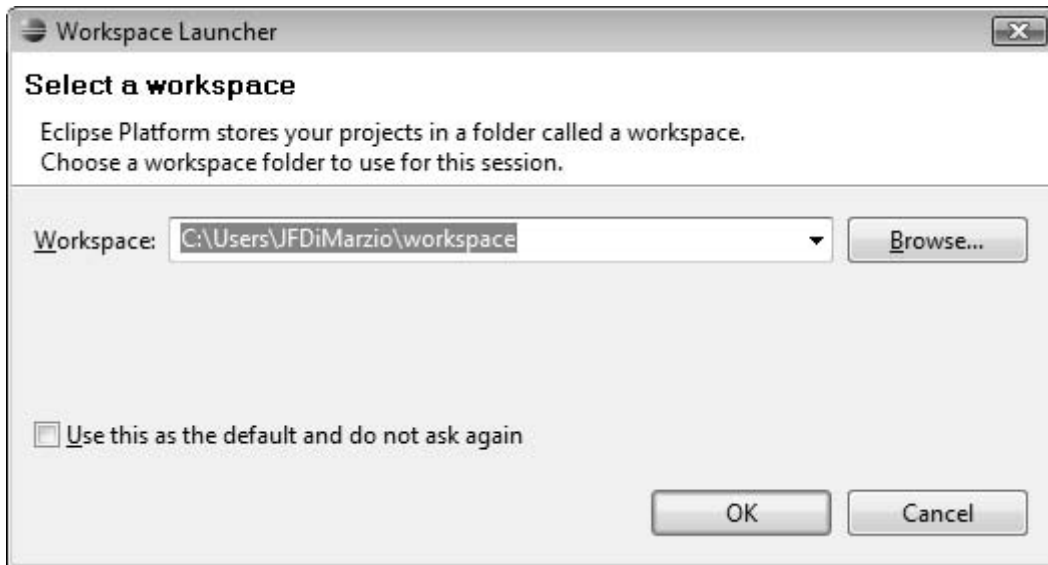
在你已经下载好 Eclipse 后,是时候安装了。导航进入到你下载 Eclipse 安装包的目录。正在写本书时,Microsoft Windows 平台下的最想的 Eclipse 安装包是 eclipse-java-europe-fall12-win32.zip。解压安装包,运行 eclipse.exe,在 Microsoft Windows 下,Eclipse 默认安装到你用户目录下,但你可能会想安装到你的 Program Files 目录下。这会让你的应用程序可用,也允许你设置不同的工作空间。下图显示了运行后 Eclipse 的主标题界面。



注意:

如果你没有看到图中的弹出界面,试试重启你的电脑。假如重启也不起效,就只有重新下载安装 Java 5 JRE。

一旦 Eclipse 安装开始,你会被催促去创建一个默认的工作空间,或是文件夹。正如多数的开发环境,工程在工作空间里创建和保存。如下图所示,默认的工作空间路径是你的用户目录。要选择不同的路径,点击 Browse,并且导航进入来设置它。



我推荐你选取默认你所有工程都在特殊工作空间的复选框。通过选取这个复选框，你将不必担心任何一件事，当你创建一个新的工程，你总会知道在哪个目录结构中找到你的源文件。在本书中，有时你将会在 Android 开发环境之外进入工程文件中工作，因此确切的知道它们在哪里是很有帮助的。

在你选择了你工作空间路径后，点击 OK。

到这，你的开发环境被下载和安装好了。而 Eclipse 的安装看上去快得出奇，但在你能创建你的第一个 Android 工程前，你仍然需要去做一些配置工作。许多需要你去完成的配置工作重点是在 Android SDK 和 Android Eclipse 插件上。

下一步你需要下载和安装 Android SDK、Android Eclipse 插件，还有配置 Eclipse 设置。到了第三章结束，你将会有一个完整配置好的开发环境（在这个环境里你就能够开始创建你的应用程序了）。接下来在第五章你会去浏览一下 Android SDK 并开始创建你第一个 HelloWorld 程序。

专家咨询

Q: Eclipse 是用来开发 Java 应用程序的，但 Android 能不能运行其他编程语言写的应用程序？

A: 当写本书的时候，还没找得到有别的 SDK 或是模拟器允许 Android 用除 Java 外任何别的语言开发。

Q: 你能否用带有其他版本的 JRE 而不是第 5 版本的 Eclipse 和 Android SDK？

A: 技术上讲你可以使用带有 JRE5 的 Eclipse 和更新一点的。而最新版本的 Eclipse 只在 Java5 JRE 上测试过。

第三章

下载和安装Android SDK

关键技术&概念

- 下载 Android SDK
 - 使用 Eclipse 更新特性
 - 下载、安装和配置 Android Eclipse 插件
 - 检查 PATH 变量
-



在前面的章节里，你下载和安装了你主要的开发环境 Eclipse。现在，你开始的开发环境被建立了，使用 Eclipse 作为你的 Java IDE，你可以使用它来开发 Java 应用程序了，但是在你开始创建手机应用程序前，还有一个额外的步骤。你必须以一定的方式配置它，这会使开发 Android 变得容易。

由于 Eclipse 是一个 Java 开发环境，你可以极度轻松的创建和编辑 Java 项目。然而，只是给了它，你没有用来理解 Android 应用程序是如何工作的运行库，你不能够开发任何运行在基于 Android 设备的应用程序。为了开始创建 Android 工程，你需要下载并安装 Android SDK。你接下来必须下载相关的 Android Eclipse 插件在 Eclipse IDE 中使用 Android SDK。在这些就绪，你就可以开始你的开发了。

如果你有任何开发经验，你会非常熟悉使用 SDK 的过程，使用 SDK 来创建即将运行在他们正开发的预期系统上的应用程序。Android SDK 包含所有的 Java 核心库，这些核心库在创建具体运行在 Android 平台上的运用程序时是必须的，这和其他的 SDK 没有什么不同之处。SDK 也包含帮助文档，Android 模拟器和一些其他主要的开发调试工具。

注意:

章节四会深入地涵盖 Android SDK 的很多功能。

为了开始，你将要从 Google Android 开发网站下载 Android SDK，登录 <http://code.google.com/android>。Google Android 开发主页包含了很多关于

为 Android 平台开发的有价值的工具和文档，包括 Android 开发者论坛（或社区）的超链接。

贴士：

假如你正在开发 Android 应用程序时遇到困难，首要的找到答案的地方是 Android 开发者论坛：<http://code.google.com/android/groups.html>。那有专门为新手、开发者和黑客们设立的讨论组。给了这个，Android 是一个十分新的平台，Android 开发者论坛是能找到全面、可靠的关于正在为 Android 产品开发的信息的其中一个地方。

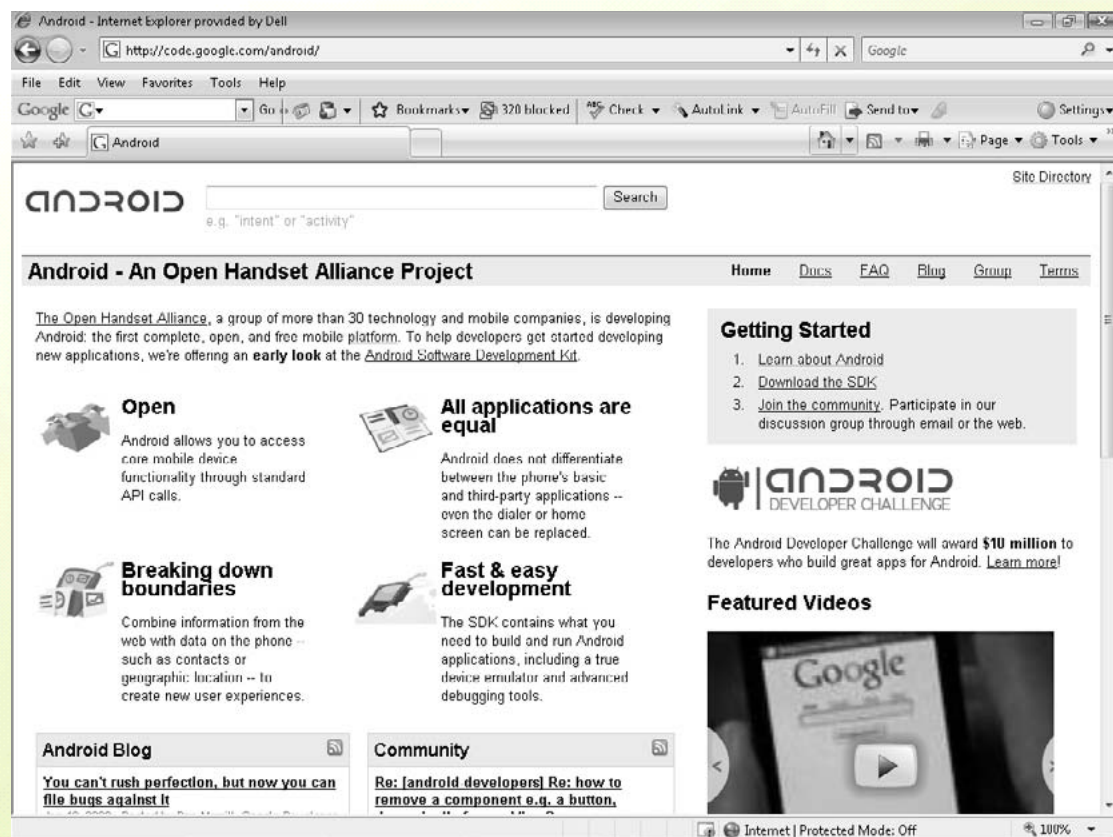


图 3-1 Google Android 开发主页

下载 Android SDK



Android SDK 很容易从 <http://code.google.com/android> 页面得到。从开发主页，点击“Getting Start”下面的 SDK 下载链接。在你同意 Android 许可协议条款后，你将看到 Android SDK 下载页面。Android SDK 是一个 79M 的下载包（对于 Windows）并且下载会相当快。点击对应你操作系统的包开始下载。

注意：

下载大小对于操作系统可以有差异。

对于 Android SDK，没有“setup”或是安装步骤涉及到。相反，你必须遵照一系列的步骤来把 SDK 和你的 Eclipse 开发环境结合起来。这些步骤的第一步是获得 Android Eclipse 插件，这之后你要配置它。

下载和安装 Android Eclipse 插件

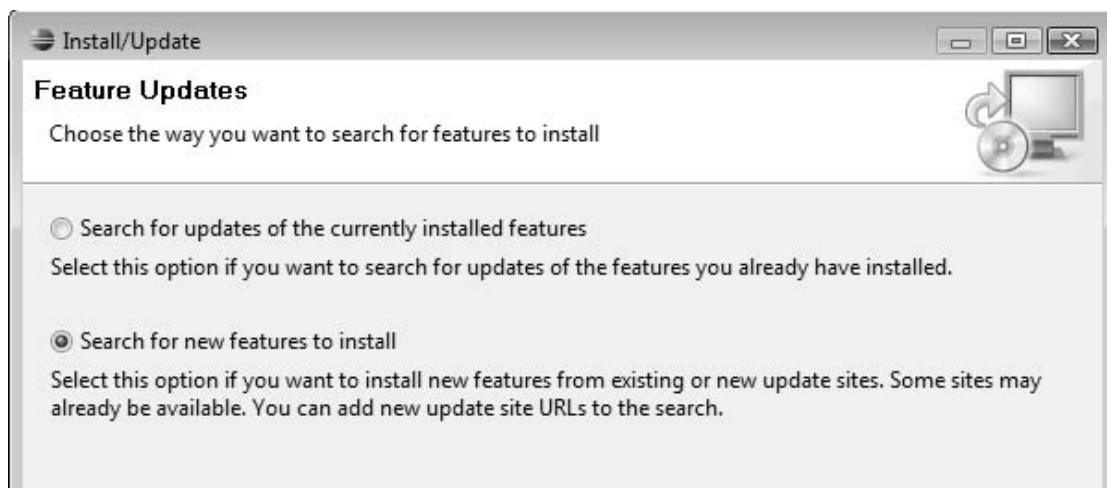
把 Android SDK 设置到 Eclipse 开发环境中的第一步是下载和安装 Android Eclipse 插件。下载和安装插件的任务可以同时进行，并且相当的简单。

1. 打开 Eclipse 应用程序。你将在 Eclipse 里面下载 Android Eclipse 插件。

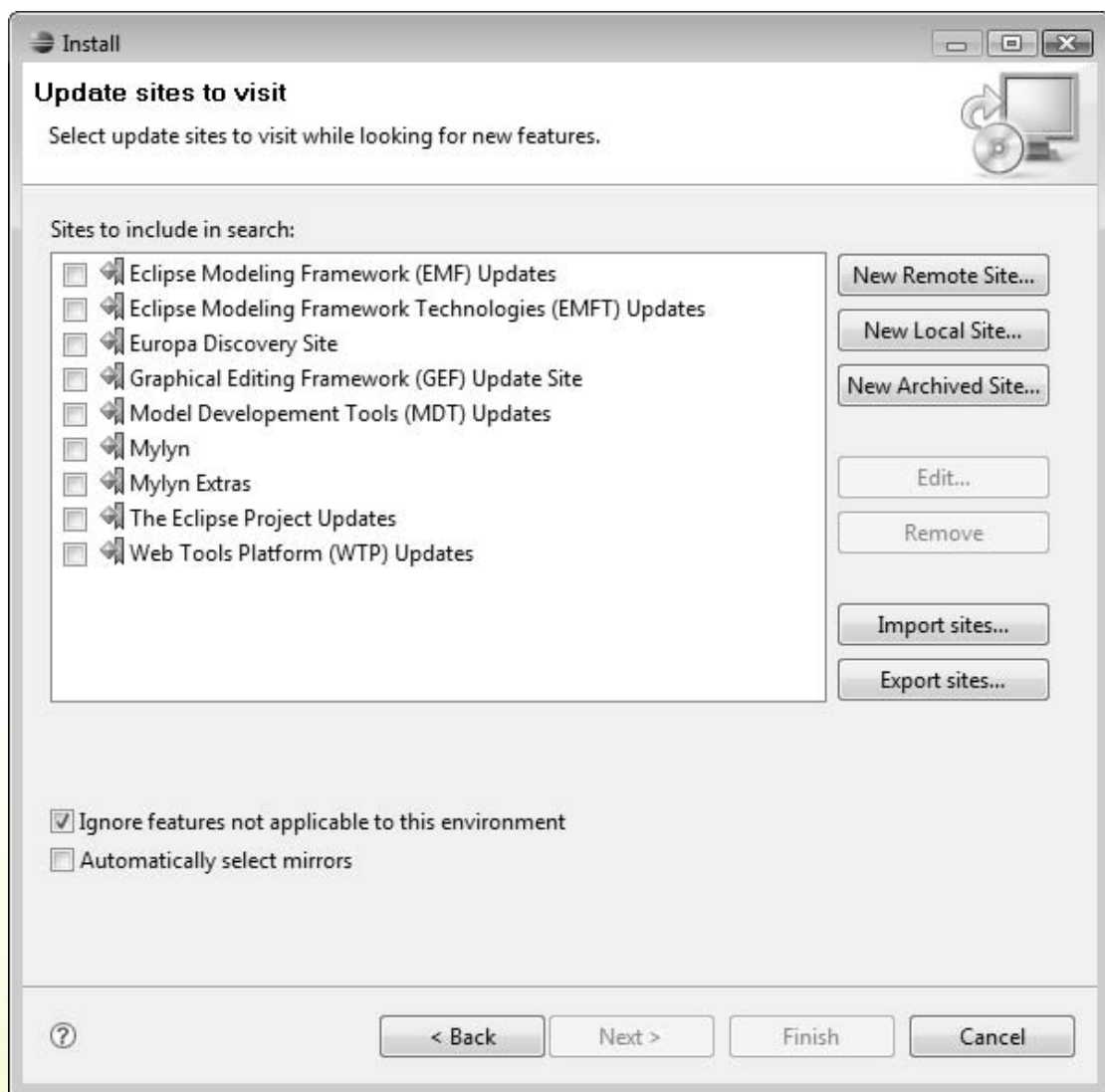
2. 选择 Help|Software Updates|Find and Install。



3. 在 Install/Update 窗口中，它允许你开始下载和安装任何一个对于你的 Eclipse 可以获得的插件的步骤，点击“Search for New Features to Install”点选按钮，接着点击 Next。

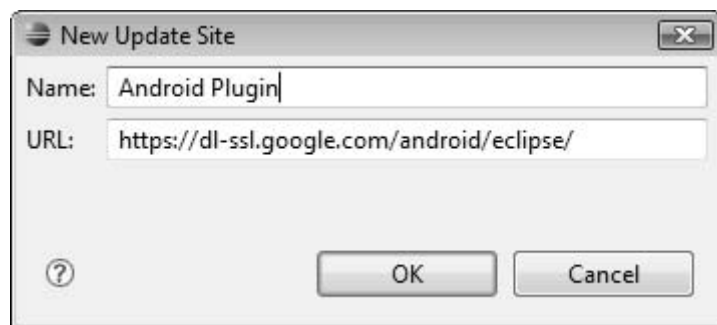


4. 安装窗口中的“Update Sites to View”页面，在下面显示，列出了所有已经得到的 Eclipse 插件的默认网址。然而，你想要的 Android Eclipse 插件，在默认网址里没有。为了下载 Android 插件，你必须告诉 Eclipse 在哪里可以找到它，因此，点击“New Remote Sites”按钮。



5. 在 "New Update Site" 对话框中，下面显示的，你必须键入两个信息来继续：新站点的名称和可用的 URL 链接。名称只是为了显示用的，对下载的插件没有影响。在名称域中，填入 Android Plugin，URL 域中键入 Eclipse 能够从该 URL 获得可用插件信息的 URL：

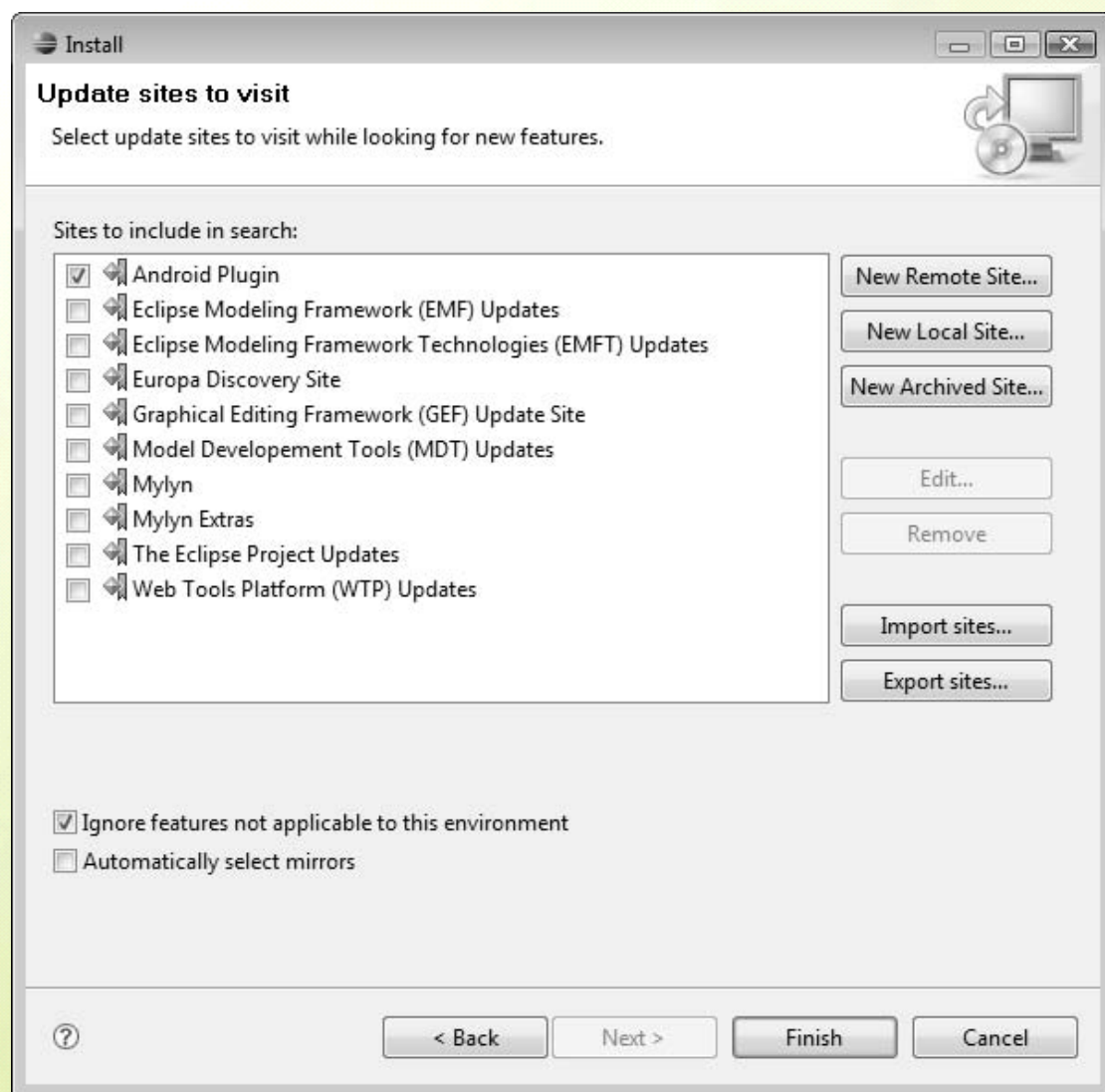
<http://dl-ssl.google.com/android/eclipse/>。点击 OK。



注意:

你站点的名称可以是任何你想要的名称，只要它能够帮你辨别它是什么链接。自由的选用其他的，而不是 Android Plugin。

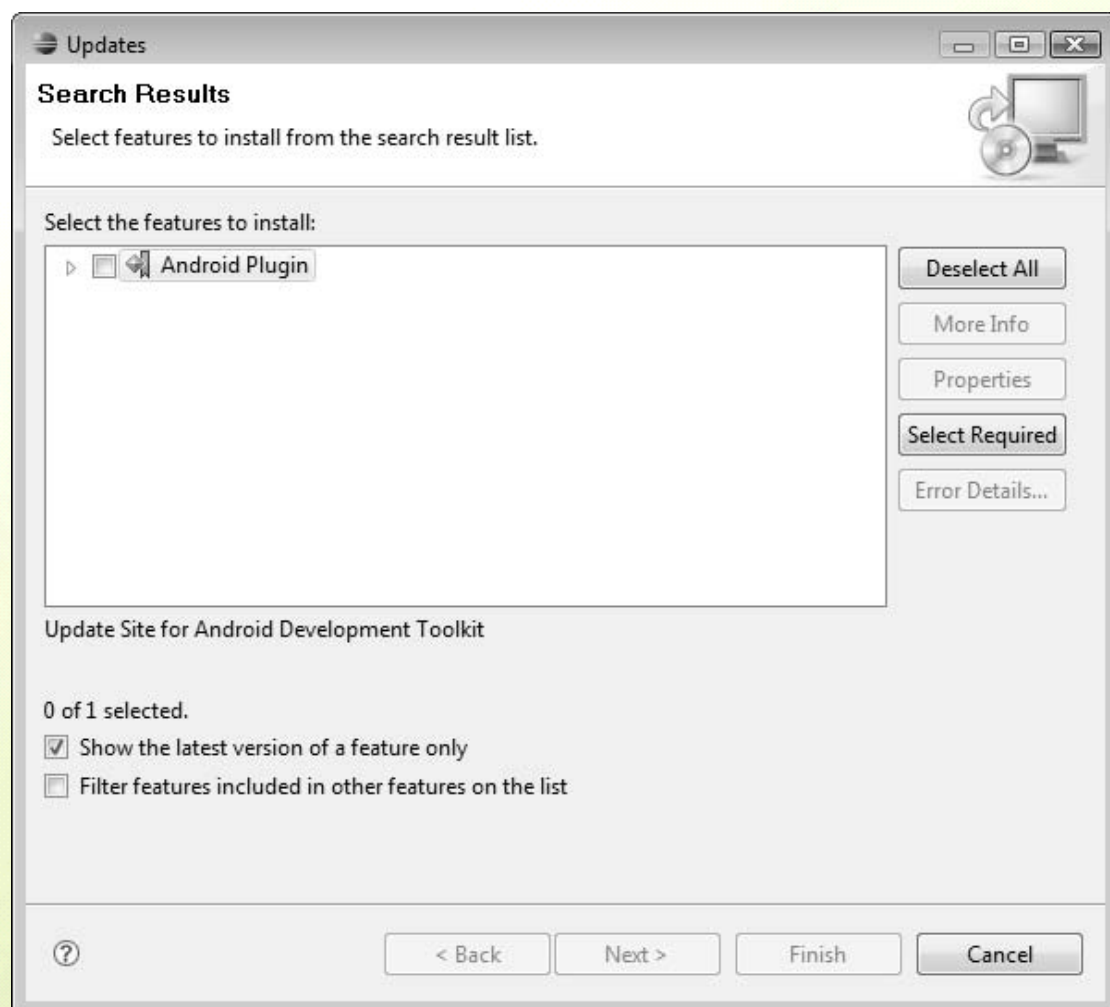
6. 一个叫 Android Plugin 的新站点名现在应该在列表中可以找到。



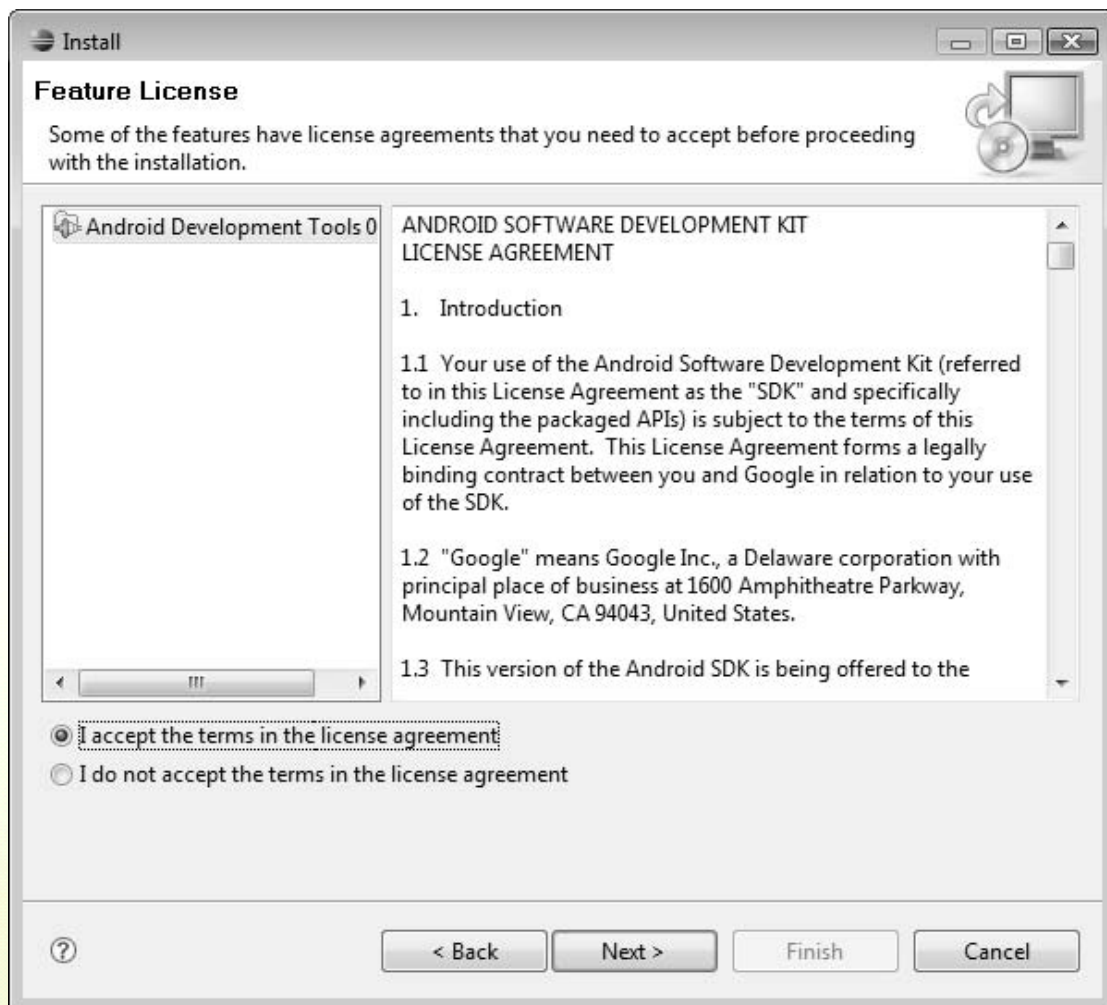
到这里 Eclipse 还没有查找插件，这只是当查找新的插件来安装时，你叫 Eclipse 来查找的路径列表。

7. 检查 Android Plugin 边上的复选框并点击 Finish。Eclipse 利用这个带有 Android 插件站点的 URL 来寻找任何可用的插件。

8. 在更新窗口中的搜索结果页面中，选择 Android Plugin 并点击 Finish。



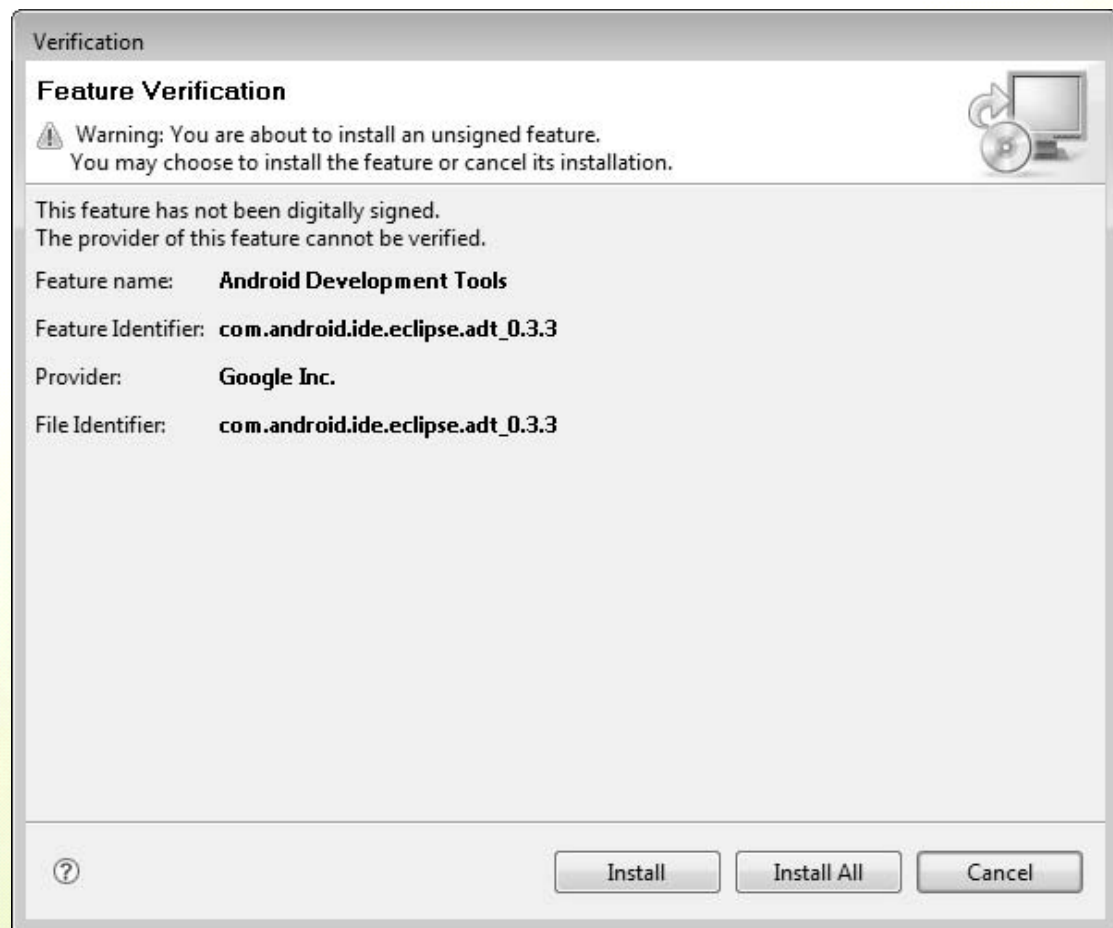
9. 在安装窗口中的特色许可页面上, 下面所示, 为 Android 开发工具接受许可协议并点击 Next。



注意:

记住一点, 所有的 Eclipse 插件都安装在 %安装路径%/eclipse/plugins 目录下。如果你要查找这些组成 Android 插件的文件, 这个信息将会对你有帮助的。

10. Eclipse 下载 Android 插件。在写本书的时候，这个插件的版本是 0.4.0.200802081635。在安装页面的最后一页，特色验证，点击“Install All”来完成 Android 插件的安装。



伴随着 Android 插件被安装，最后你不得不执行的一步是配置插件。

配置 Android Eclipse 插件

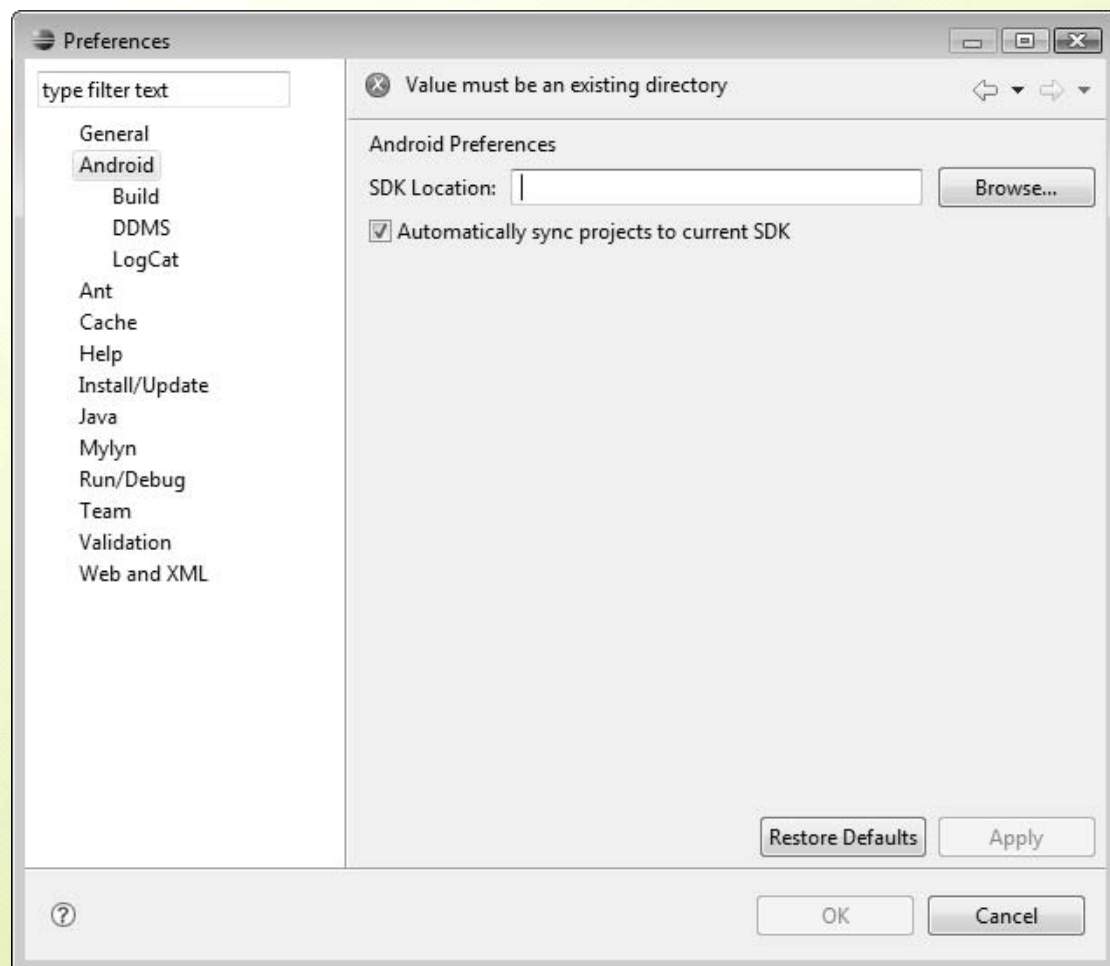
在安装好 Android Eclipse 插件后，Eclipse 将会提示你重启。假如它没有提示你，现在就重启 Eclipse。重启 Eclipse 将确保这些工程有一个带着这个插

件被安装的更新来重新开始。为了减少错误配置的机率，确保跟随这里的配置步骤是很重要的。

Android Eclipse 插件通过 Eclipse 的配置窗口来进行配置的。步骤如下：

1. 从 Eclipse 主窗口选择 Window|Preference 。

2. 在“Preferences”（偏好）窗口中，如下所示，在左边菜单中选择 Android。在窗口右边，点击 Browse，找到你硬盘中 Android SDK 的位置，并把它选入到 SDK Location 域中。Eclipse 需要这个信息才能够使用所有 Android 支持的工具，例如模拟器。



3. 选择 “Automatically Sync Project to Current SDK ” 复选框并点击 Apply。

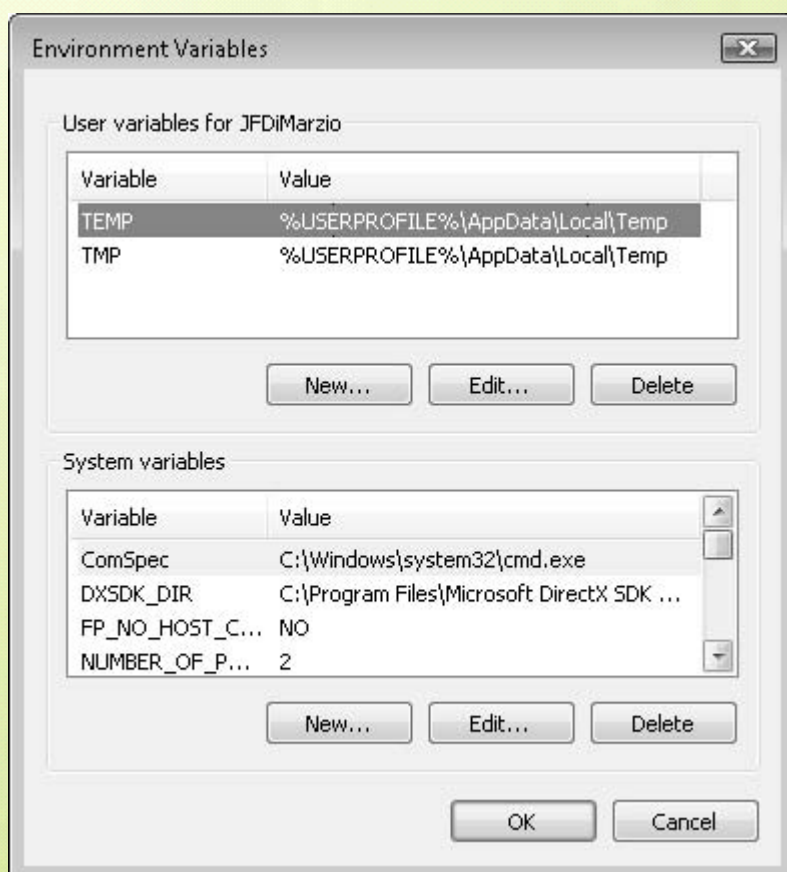
注意:

Windows 下 Android 插件是以一个 zip 文件的方式进行传送的, 该文件包含了一个很长的目录名的目录: android-sdk-m5-rc14-win32。重命名这个目录使它容易管理, 可能在后面章节中, 尤其是当用命令行编程时这会很有用。你可能也会想把 SDK 取出到工程文件目录。



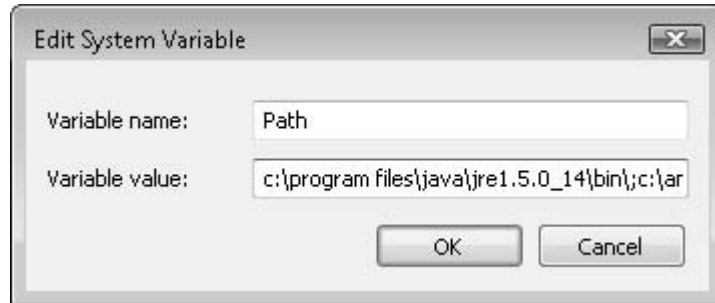
4. 安装 Android SDK 最后一步是把它设入你的 PATH 变量中。如果你在使用 microsoft windows 机器, 右击 “电脑” (或 “我的电脑”, 依赖于你的 Windows 版本), 并选择 “属性” 来打开系统属性对话框。点击 “高级” 标签。

5. 点击环境变量来显示同名的变量窗口, 下图所示。这是你能够编辑 PATH 变量的地方。



6. 在系统变量中，找到 PATH 变量并双击它。

7. 在系统变量编辑对话框中，下面所示的，添加 Android SDK 的路径 (注意: Android SDK 下的 tools 文件夹的路径)-用分号来和存在的路径分隔-并点击 OK 来应用你的更改，再次点击环境变量窗口上的 OK。



Android SDK、Eclipse 和 Android Eclipse 插件现在完整的为开发配置和准备好了。在下一章节，你将浏览 Android SDK，并学习它的特性。Android SDK 包含了很多的工具来帮助你开发完整特色的手机应用程序，下一章提供了一个好的概述。

专家咨询

Q: Android SDK 对于任何其他编程语言而不是 Java，是可用的吗？

A: 不是，Android 应用程序只是用 Java 语言来开发的。

Q: 是不是会有 Android SDK 的升级？

A: 是的，甚至在撰写本书时，一个 SDK 的更新被发布了，更新地址有可能在这个平台中发布。要获得最新的更新，我建议经常查看一下开发网页面。

Q: 假如一个更新发布了，我怎样来升级我的 SDK？

A: 升级 SDK 应该十分谨慎。当一个新的 SDK 发布，一个新的插件也会被发布。在写本书期间，一个新的 SDK 和新的插件都发布了。我试图去使用提供的升级工具来改变版本。然而，这被证明没有结果，并使我处于两个冲突的版本中，没有一个正常工作。我最终不得不卸载两个版本并且只重装最新的版本。这个最新的 SDK 接下来正常工作了。我建议任何人可能面临从一个版本的 SDK 或插件升级到另一个版本的时候，使用这个相同的步骤：简单的卸载老版本，并安装新的一个版本，而不是升级。

第四章

浏览Android SDK

关键技术&概念

- 使用 Android SDK 文档
 - 使用 Android SDK 工具
 - 使用例子程序
 - 学习 Android 应用程序生命周期
-



现在你有了你建立好的开发环境了，你准备去浏览一下 Android SDK，它包含了许多特意帮你设计和开发运行在 Android 平台上的应用程序的文件和工具。这些工具设计很棒并能够帮你编写出一些难以置信的应用程序。在你开始编程之前，你需要准备去熟悉一下 Android SDK 和它的工具。

Android SDK 也包含一些用来在核心的 Android 特性中（例如利用电话功能一打和接电话，GPS 功能，和文本短信功能）测试你应用程序的类库。这些类库组成了 Android SDK 的核心部分并且是你经常会用到的一些，因此，花一些时间来学习所有的这些核心类库。

本章节涵盖了 Android SDK 中包含的重点。本章最后，熟悉了 Android SDK 内容后，你将很轻松舒适地开始写应用程序了。就像任何事一样，在你深入进行练习之前，你必须熟悉它的内容和使用方法。

警告：

我不会每一分钟都详述 Android SDK；Google 在它的文档里做了很好的工作。为例避免花过多时间来讨论事件工作而不是把它们如何工作展示给你，我试着来让讨论尽可能简要。我只会涵盖多数重要主题和条款，剩下让你自己用你的速度自由地深入浏览。

Android SDK 中有什么？

Android SDK 以一个简单的 zip 压缩包来下载（如第三章描述）。Android SDK 的体积，以数字来算文件的话，包括了文档编程 APIs，工具和一些组成剩下部分的例子。本篇提供了近距离的正确审视 Android SDK 中包含了什么。

贴士：

第三章建议你吧 Android SDK 解压到 "Program Files" 目录下，以便跟踪更简单。由于你使用了默认的解压设置，如果你在找 SDK 时有困难，它应该在以下的文件夹中：

`/%下载目录%/android-sdk-m5-rc14-windows/android-sdk-m5-rc14-windows.`

进入你解压 Android SDK 的地方，以便你能开始浏览里面的文件结构。有一些文件在根目录，像 `android.jar`（一个被编译过的 Java 应用程序，它包含核心 SDK 类库和 APIs）和一些发布记录，Android SDK 余下部分被分隔进三个主文件夹中：

● Doc

包含所有 Android 的伴随文档

注意：

在 Docs 文件夹下找得到的很多文档在 <http://code.google.com/android> Android 开发网站中也能找到。

● Samples

包含你可以在 Eclipse 中编译，测试的六个例子程序。

● Tools

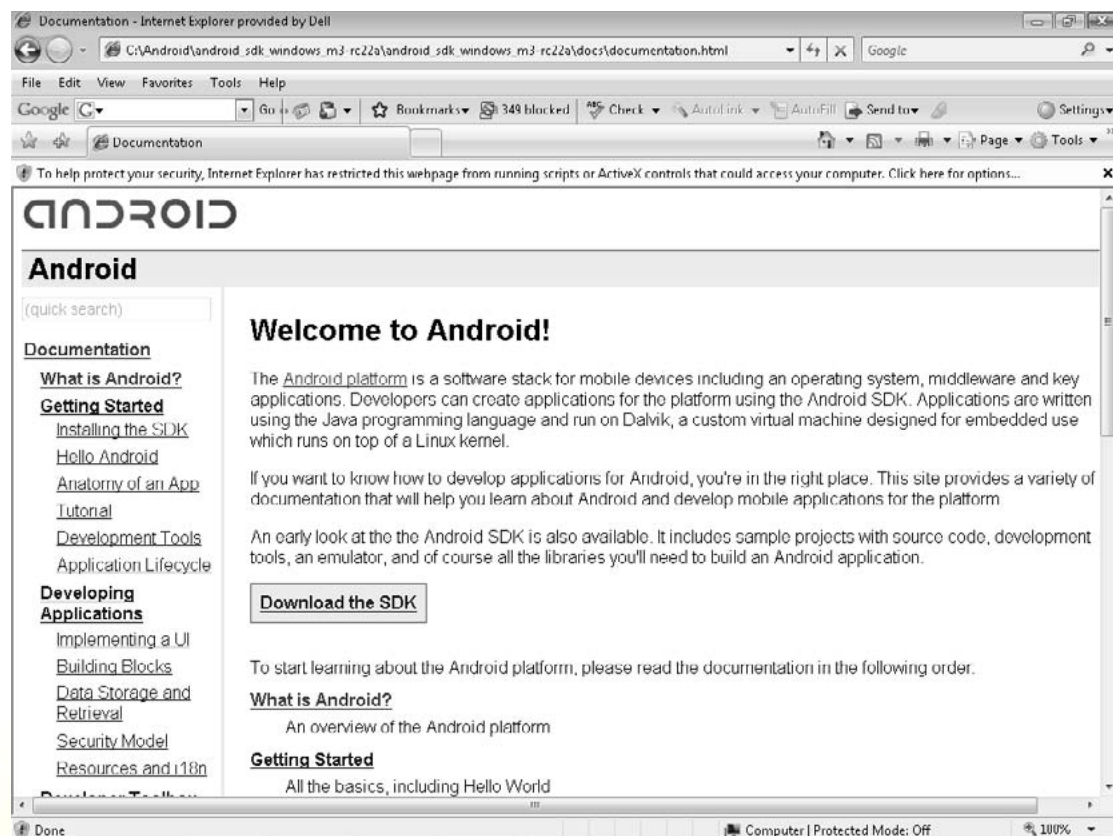
包含所有开发、编译和调试工具，在你开发 Android 应用程序过程中你会需要它们。

关于每个 SDK 文件夹中包含什么，接下来的篇章讨论更多细节。为了展示 Android 的功能，每个 API Demo 被编译和运行。当你学怎样用 Microsoft Windows 和 Linux 的命令行来创建和编译应用程序时，许多工具会在后面章节被讨论和展示。



Android 文档

Android 文档位于 Android SDK 里的 Docs 文件夹中在 %sdk 文件夹%/DOCS 下。和 Android SDK 一起被提供的 Android 文档包括下载和安装 SDK 的步骤、快速开始开发应用程序的步骤、包的定义。这个文档是 HTML 格式的并能在 SDK 文件夹根目录下通过 documentation.html 来使用。下图描述了 Android SDK 文档的主页面。



通过 `documentation.html` 中的超链接，你可以导航到 Android SDK 中包含的所有文档。

警告：

当你进入 Android SDK，当你点击一些链接时，你可能会想一些页面好像是丢失链接或是丢失了，因为右边的屏幕可能是空的。然而，如果你滚动下来你会看到只是没有排好版的页面。

使用 Android SDK 来工作，我发现文档的一些篇幅我比其他更偏爱。对我来说，Android 文档最有价值的部分是下面第一一些：（正如在导航栏中出现的）

- 参考信息

- Class 索引
- 许可列表
- 资源类型清单

- 常见问题

- 故障、维修表

当你开始出差错时，故障、维修表（文档的一小部分）将会十分地有帮助。当你通过本书来继续，并开始开发你自己的 Android 应用程序时，你将会发现文档的这参考信息部分更有帮助。例如，当你照本书处理创建更复杂应用程序的篇幅前进时，许可列表部分将会对你很有用，而现在它对你没有一点用处。花一些时间来熟悉 Android SDK 文档和提供给你的成百的文档。



Android 示例

示例文件夹，`/sdk 文件夹%/SAMPLES`，包含六个展示一个好的 Android 功能横截面的例子程序。

- API Demo (API 样例)
- Hello, Activity!
- Lunar Lander (月球发射)

- NotePad (记事本)
- Skeleton APP (框架程序)
- Snake (贪食蛇)

这些用来给你一个怎样开发 Android 应用程序的快速思路的例子程序由 Google 提供。每个示例程序都展示了 Android 功能的不同面。你可以在 Eclipse 中来打开和运行这些程序。下面是每个程序的一个简单描述。

API Demo

API Demo 是一个以单个 Activity 来展示多种 API 功能的主程序。

注意:

一个 Activity 是一个 Android 应用程序。Activity 在后面章节涵盖得更深入。

API Demo 程序, 如下图所示, 包含多种、更小的 Android 不同功能的示例。



这些许多被包含在 API Demo 例子中的程序包含了 3D 图像转换、列表、前进对话框和一个触摸屏画图样例。

尝试: 运行 API Demo 例子程序

使用 Eclipse 把 API Demos 作为新的 Android 工程项目加载。要这样进行, 从 Eclipse 菜单栏中选择 File|New|Project; 一个新的 Android 工程项目向导就打开了。从现在不要担心在这个向导中的选项了。选择“从已有的资源创建工程”并浏览加载有 API Demo 运用程序的文件夹到里面。当工程已经加载, 选择“Run”来看看它在 Android 模拟器中执行。

你自己导航到超过四十多个不同程序来观看演示。使用每一个程序来让你熟悉一下每一个 API 工具所展示的技术和功能。

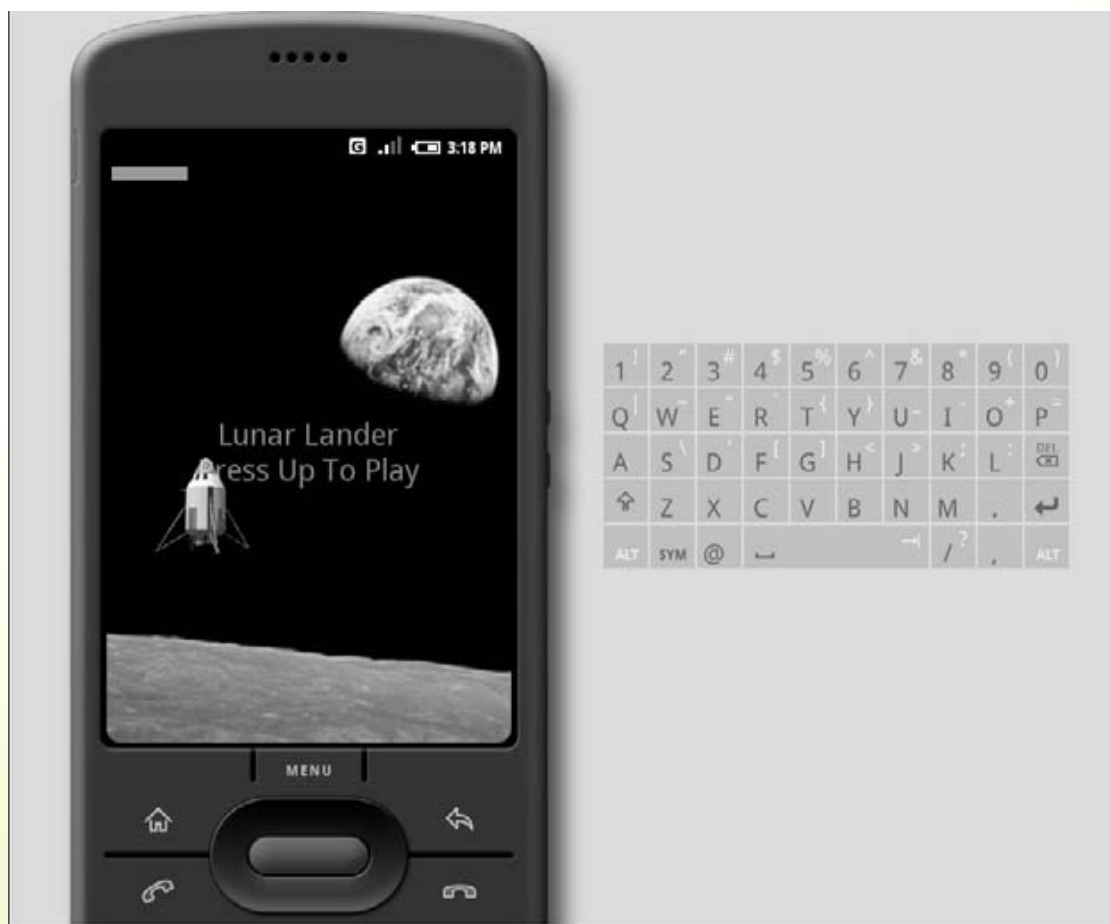
Hello, Activity !

在下图展示的图中的 Hello, Activity! 运用程序是一个简单的 HelloWorld! 一典型的例子。虽然它内部设计简单, Hello, Activity! 很好的展示出了这个平台的能力。你将会在下一章节中创建你自己的 HelloWorld! (经典程序)



Lunar Lander(月球登陆)

下一图中展示的“月球登陆”是一个在 Android 模拟器上运行的小游戏。“月球登陆”展示了 2D 游戏在 Android 上是如何工作的。控制非常简单，并且这个游戏也不是非常复杂。而给了这些基本的东西，它对于游戏开发来说是一个很好的入门点。



“月球登陆”实现了一个简单的控制方法（上，下，左，右）。这个游戏也显示了这个平台给出的给人深刻印象的相对地流动图和样式。如碰撞检测这样复杂的游戏理论是用一个简单的方法实现的。虽然本书没有涵盖为 Android 平台开发游戏，假如你很感兴趣做这个，你可能会想看看“月球登陆”来获得一些启发。

Note Pad (记事本)

如下图所示的记事本，它允许你打开、创建和编辑一些小记录。NotePad 不是一个全功能的文字编辑器，因此，不要期望它能够做一些事来和 Windows Mobile 的 Word 媲美。然而，它作为一个用来展示相对少量的代码可能完成的程序的示范工具是做得很好、有效的。



Skeleton APP (框架程序)

下图所示的框架程序是一个应用程序框架。这是一个更简单的展示了一对不同的应用程序特性功能（如字体、按钮、图片、表单）的基础程序。如果你只是通过它自己来运行框架程序，你实际上不会得到它之外多的东西。通过参考“Skeleton APP 框架程序”，它作为怎样实现特别条目的资源，你将会被更好的服务。



Snake (贪食蛇)

最后一个包含在 AndroidSDK 中的例子是"贪食蛇", 这是一个比起"月球登陆"更单纯化的"SNAPU"风格的小游戏。这幅图展示了"贪食蛇"运行的界面。



注意:

如果你导航进入到每一个例子程序的根文件夹中, 你将看到一个命名为 `src` 的文件夹。这是给出例子的源代码文件夹。你可以用这个来看、编辑和为任何应用程序来重编译这个代码。利用这个源代码来学习一些关于 Android 平台的技巧和条目。

Android 工具



Android SDK 为开发者支持了一系列强大和有用的工具。到本书结尾, 你将只是很少的直接来使用它们。本篇幅只是来快速浏览一下这些工具中的一些, 当你进入命令行开发时, 这些工具在以后章节中将会涵盖得更深层。

注意:

更多关于 Android SDK 中包含的其他工具的信息, 查阅 Android 文档。

emulator.exe

毋庸置疑, Android SDK 中包含的一个最重要的工具是 `emulator.exe`。`emulator.exe` 运行 Android 模拟器。Android 模拟器用来在 Android 仿真环境下运行你的应用程序。当写这本书时, 还没有 Android 平台的硬件设备发布, `emulator.exe` 将会是你唯一“本地”平台测试应用程序的途径。

你可以在命令行下运行 Android 模拟器或是从 Eclipse 中执行它。在本书中, 你通常是让 Eclipse 来为你运行 Android 模拟器环境。然而, 在 Eclipse 外用 Android SDK 开发, 在兴趣上给你了所有你需要的信息, 当你创建你的 HelloWorld! 程序时第六章涵盖了命令行使用 `emulator.exe`

当使用 Android 模拟器来调试你的应用程序时, 你有两种导航用户界面的选择。第一个, 模拟器自带的可用按钮, 如图 4-1 所示。你能使用这些按钮来导航 Android 和所有你为这平台开发的应用程序。

贴士:

电源开/关、音量升/减按钮被隐藏在虚拟设备的边上。当你把鼠标放到它们上面时, 它们就会标示出来。

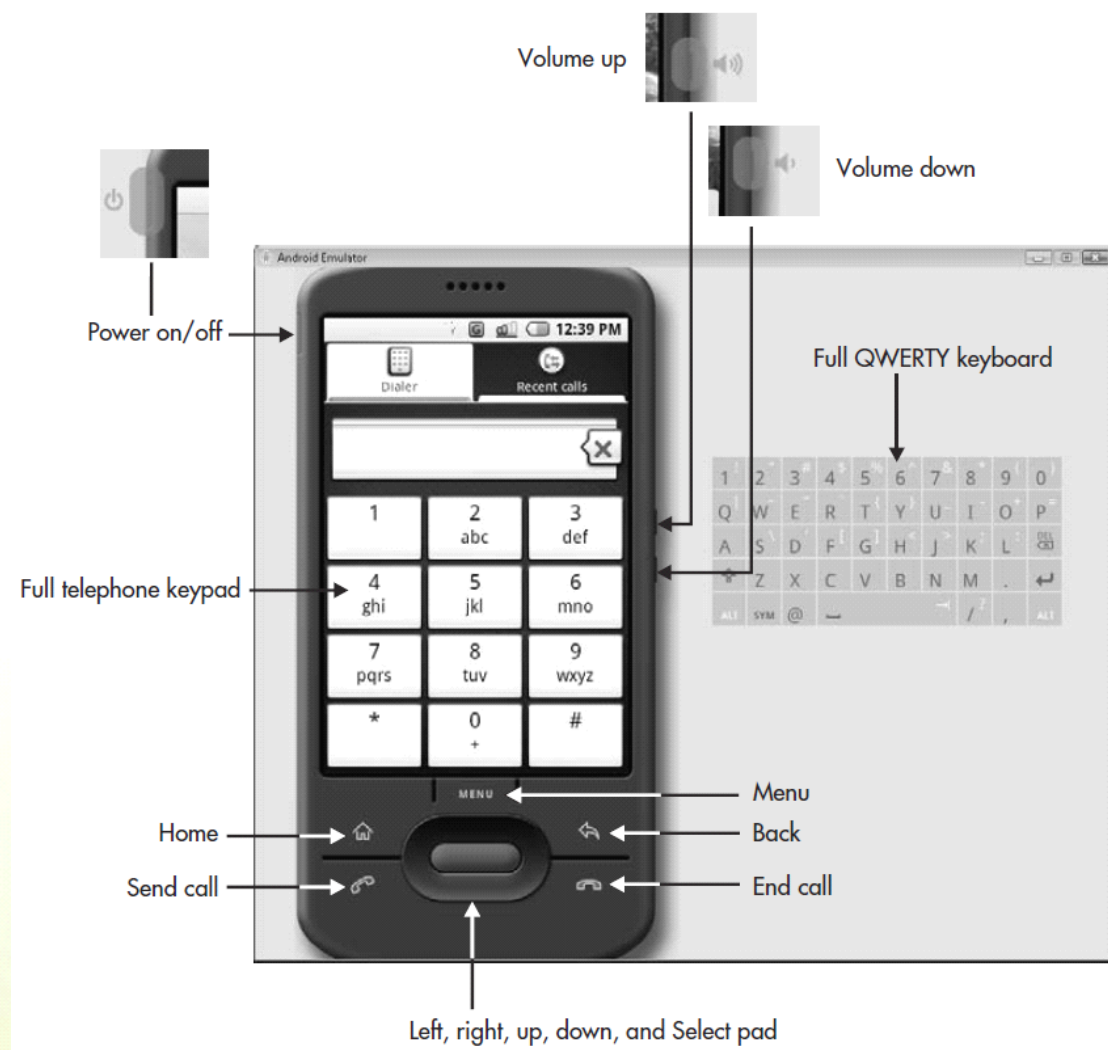


图 4-1 Android 模拟器导航

现在很多高端手机包含了触摸屏，当你使用模拟器时你拥有的第二种输入选择是仿真触摸屏。你把鼠标当做触摸笔。通过使用鼠标，在模拟器屏幕上的物件将会有交互反映。

adb.exe

另一个当你正在使用命令行开发时将会变得非常有用的工具是 Android 调试桥或者 adb (adb.exe)。这个工具允许你对 Emulator.exe 工具发出命令。当你正在命令行开发环境下，adb 工具允许你做以下的：

- 启动和关闭服务
- 安装和卸载应用程序
- 从模拟器中移出文件或把文件移进模拟器

MKSDCARD.exe

假如你将测试一个需要从一张插入移动设备的 SD 存储卡中读或写文件的应用程序, MKSDCARD.exe 是一个非常有用的工具。MKSDCARD.exe 在你的硬盘上创建一个将装有、保存测试文件的小硬盘分区。模拟器将把这个分区当做一个真正的 SD 存储卡。

DX.exe

DX.exe 是 Android SDK 的编译器。当再次运行你的 Java 文件, DX.exe 将生成后缀名为 .dex 的文件—Dalvik 可执行格式。这些文件是被解析、运行在 Android 设备上的正确格式。

注意:

Android 可执行文件被叫做 Dalvik 可执行文件, 作为 Android 用来运所有应用程序的 Dalvik 虚拟机相关的文件。Dalvik 虚拟机在每个应用程序自己的进程中使用和 Android 核心应用程序相同的优先级运行这个应用程序。

activityCreator(.bat 或 .pn)

activityCreator 是一个用来安装一个基本的开发环境的简单命令行工具。当从命令行中运行它, activityCreator 将会安装创建一个基本 Android 应用程序必需的框架文件。如果你没使用 Eclipse, 拥有这些文件是尤其有用的。当你创建一个新的工程时, Android Eclipse 插件通过调用 activityCreator 来为你搭建这些框架文件。

你将会看到 activityCreator 表现为不同的脚本文件, 这依赖你正运行什么样的环境。假如你在 windows 环境, 它将会是一个 .bat 文件; 否则它会是一个 python(.pn) 脚本。你简单地执行这个脚本, 它带着正确参数来轮流地调用真实的 activityCreator 进程。

APIs

API, 或应用程序编程接口, 是 Android SDK 的核心。一个 API 是功能、方法、属性、类和库的集合, 它被应用程序开发者用来创建运行在特殊平台上的工程。Android API 包含了所有你创建能工作和交互的基于 Android 应用程序时所需的特征信息。

Android SDK 也包含了两个补充的 API—Google APIs 和可选 APIs。后面章节中, 当你使用这些 API 开始写应用程序时将会更多关注它们。现在, 快速看一下它们包含了什么, 以便你熟悉他们的用法。

Google APIs



Google APIs 被包含在 Android SDK 中, 并由允许你把你的应用程序绑定到已有的 Google 服务中的编程参照组成。假如你要写一个应用程序并允许你的使用者通过你的程序来使用 Google 服务, 你就需要包含 Google API。

打开 android.jar 文件, Google API 包含在 com.google.*包中。有很多的包含 Google API 的包。许多装有 Google API 的包中包含那些为图形、便捷、联系、日历的公用程序。而在本书中将主要关注于 Google Maps 使用的包。

使用 com.google.android.maps 包, 你可以创建和已经常用的 Google Maps 界面无缝交互的应用程序。这一系列包开辟了有用的程序只是等待被开发的整个世界。

Google API 也包含了一系列有用的包, 这些包允许你利用更新的可扩展信息传输和出席协议 (Extensible Messaging and Presence Protocol) (XMPP)—由 Jabber 开源社区开发。使用 XMPP, 一个应用程序可以快速成为其他客户的出席和发信息、通信的可获得的意识。假如你要创建一个使用手机消息功能的“聊天”形式的应用程序, 用 XMPP 处理的 API 包非常有用。

可选 APIs

Android SDK 包含了大量的可选 API, 这些可选 APIs 涵盖了标准 Android API 没有的功能。这些可选 API 被考虑成可选的, 因为它们处理的功能可能在现有的手机上没有。这就是说许多为 Android 平台开发的设备可能包含其他设备没有的升级和特色; 当你尝试在你的 Android 应用程序中使用这些特色时, 可选 API 涵盖了你的编程选项。

这些特色的其中之一 (你将在后面章节中使用) 是一个基于手机的 GPS。Android LBS (基于本地服务, Location-Based Services) API 处理需要从一个设备的 GPS 模块中接收和使用信息的功能。(结合 Android LBS API 和 Google Maps API 中的信息, 你可能拥有一个非常有用的应用程序。它能够及时的自动显示你任意给出的点的地图。)

其他的可选 API 包括使用蓝牙、Wi-Fi、MP3 播放, 和支持 3D-OpenGL, 依赖于硬件。

应用程序生命周期

如果你作为一个应用程序开发者有一定的经验的话，你会很熟悉应用程序生命周期的概念。一个应用程序生命周期由应用程序从执行到终止过程必须遵照的步骤组成。每一个应用程序，不管它用什么语言编写，都有一个明确的生命周期，Android 应用程序也不例外。这篇阐述一个 ASP 程序的生命周期并和 Android 应用程序的生命周期作比较。

标准 ASP 应用程序生命周期

一个标准 ASP 应用程序的生命周期和 Android 程序生命周期足够相似，就用这个来做个好的比较。ASP 应用程序从开始到处理经过五个不同的步骤。这些步骤要求所有的 ASP 程序都要实现，并真正定义了一个 ASP 程序是什么。这些步骤按顺序是：

1. 应用程序-启动
2. 事件
3. HTTP 应用程序. 初始化
4. 处理、分配 (Dispose)
5. 应用程序-终止

注意：

许多 ASP 参考资料把处理、分配 (Disposal) 和应用程序-终止考虑成生命周期的一个步骤。然而，Disposal 调用在它经过应用程序-终止之前可以被拦截。这就能允许应用程序在它真正销毁前执行特殊的函数。

当应用程序被服务器请求时，应用程序-启动被调用。这个过程依次进入到事件过程。当所有联系的应用程序组件加载好，HTTP 应用程序. 初始化被调用。程序就执行它自己的事件，当用户试图关闭时，Dispose 被调用。Dispose 接下来进入到关闭程序的应用程序-终止过程。

这是一个很标准的应用程序生命周期。大多数程序遵循相似的生命周期：一个程序被创建，加载，拥有事件，销毁。下篇幅展示怎样和 Android 应用程序生命周期来比较。

Android 应用程序生命周期

在许多程序的生命周期中，Android 应用程序生命周期中系统控制方面是独一无二的。所有 Android 程序，或是 Activity，是在它们自己进程中允许的。

所有正在运行的进程被 Android 监视并依赖于这个 Activity 是如何运行（当前的 Activity，一个已分配的 Activity，后台 Activity，或是第四种），Android 可能会选择终止这个 Activity 来收回需要的资源。

注意：

当决定一个 Activity 是不是应该关闭时，Android 考虑许多因素，如用户输入、内存使用和进程时间。

一些在 Android Activity 生命周期中调用的特殊方法是：

- onCreate
- onStart
- Process-specific 事件（如开启 Activity 或使用数据库）
- onStop
- onDestroy

遵循和其他应用程序生命周期相似的逻辑，一个 Android 应用程序被创建，进程被启动，事件被发出，过程停止，接下来程序销毁。虽然有点不同的地方，但多数应用程序开发者感觉这些在生命周期的步骤应该是舒适的。

专家咨询

Q: Google 曾升级过 Android SDK 吗？

A: 是的，从我写这本书开始，Google 已经升级 Android SDK 三次了。当 Android SDK 升级可以获得时，Google 会发布在 Android 网站上。

Q: API Demo 展示程序会不会出现在最终的产品中？

A: 可能不会。API Demo 被创建用来展示出 Android 功能。虽然有核心的特征程序，它可能包含了一些在 API Demo 中找得到的元素。但我们可能不会在最终版本中看到“月球登陆”。

第五章

应用程序：HelloWorld

关键技术&概念

- 创佳一个新的 Android 工程
 - 用 View 视图来工作
 - 使用 TextView
 - 修改 main.xml 文件
 - 在 Android 模拟器上运行应用程序
-



在本章，你将创建你的第一个 Android Activity。本章细查了构建应用程序从开始到结束的过程。我将向你展示如何在 Eclipse 中创建一个 Android 工程，并在 Android 模拟器中运行编写完的应用程序。最后的应用会是一个运行在 Android 平台上完整功能的项目。

实际上，当你来到这个章节时，你将创建更多的 Android Activity。计算机编程传统规定你的第一个应用程序是经典的 Hello World! 程序，因此，在第一部分你将创佳一个标准的只有一个空背景和"Hello World!"文本的 Hello World! 程序。这将给你一个好的 Android 控制和内部工作介绍。

注意:

你将经常看到"Application"和"Activity"被交替使用。它两个不同处是一个"application"可一个由多个 Activity 组成，但是一个"application"必须至少含有一个 Activity。你应用程序每一个"窗口"或是屏幕是一个单独的 Activity。因此，如果你创建一个十分简单的只带一个数据屏幕的应用程序(像本章中的 Hello World! 程序)，那就只是一个 Activity。在以后章节中你将会创建有多个 Activity 的应用程序。

为了确保你对 Android 开发编程有个全面的审视，第六章节将使用 Microsoft Windows 和 Linux 的 Android SDK 命令行环境创建这些程序。换句话

说, 本章节涵盖在 Eclipse 中创建的过程, 第六章节涵盖了命令行工具的创建过程。因此, 在继续之前, 你应该检查一下你的 Eclipse 环境被正确配置。回顾第三章为 Android SDK 设置 PATH 声明的步骤。你也应该确保 JRE 在你的 PATH 声明中正确无误。

贴士

当你试着去用任何一个命令行例子开发时, 如果你有配置相关的问题, 试着参考第二、三章节的配置步骤; 并看一下 Android SDK 文档。



在 Eclipse 中创建你的第一个 Android 项目

要开始你第一个 Android 工程项目, 打开 Eclipse。当你第一次打开 Eclipse 时, 它开启一个空的开发环境(参看 图 5-1), 那就是你想要开始的地方。第一任务是为你的程序设定和命名工作空间。选择 File|New|Android Project, 这将开启新 Android 项目向导。

警告

不要从新菜单中选择 Java Project。虽然 Android 应用程序是用 Java 编写, 并且你在 Java 项目中进行你所有的开发, 但这个选项会创建一个标准的 Java 应用程序。选择 Android Project 使你能够创建具有 Android 特性的应用程序。

如果你没有看见 Android Project 选项, 这就表示 Android Eclipse 插件损坏或没有正确安装。回顾第三章给 Eclipse 安装 Android Eclipse 插件过程来修复该问题。

Android Project 向导为你创建两个东西:

- 一个绑定到 Android SDK 的框架程序, 通过使用 android.jar 文件并把工程绑定到 Android 模拟器中。这将允许你使用所有的 Android 类库和包来编写代码, 也让你能在适当的环境中调试你的应用程序。

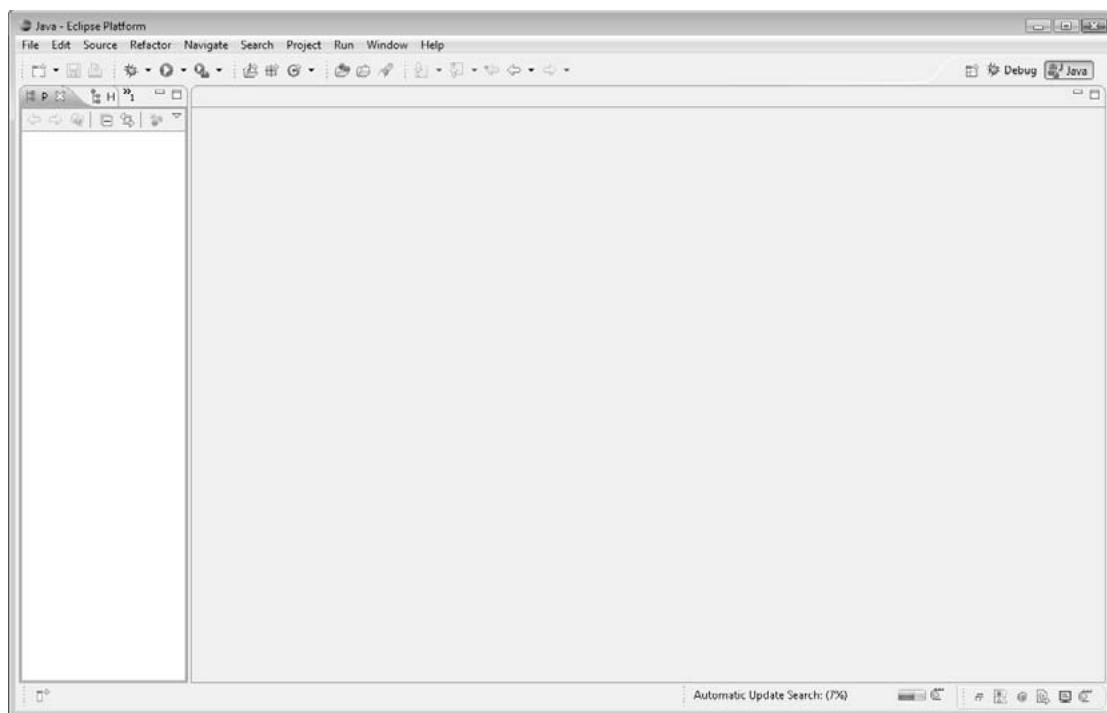
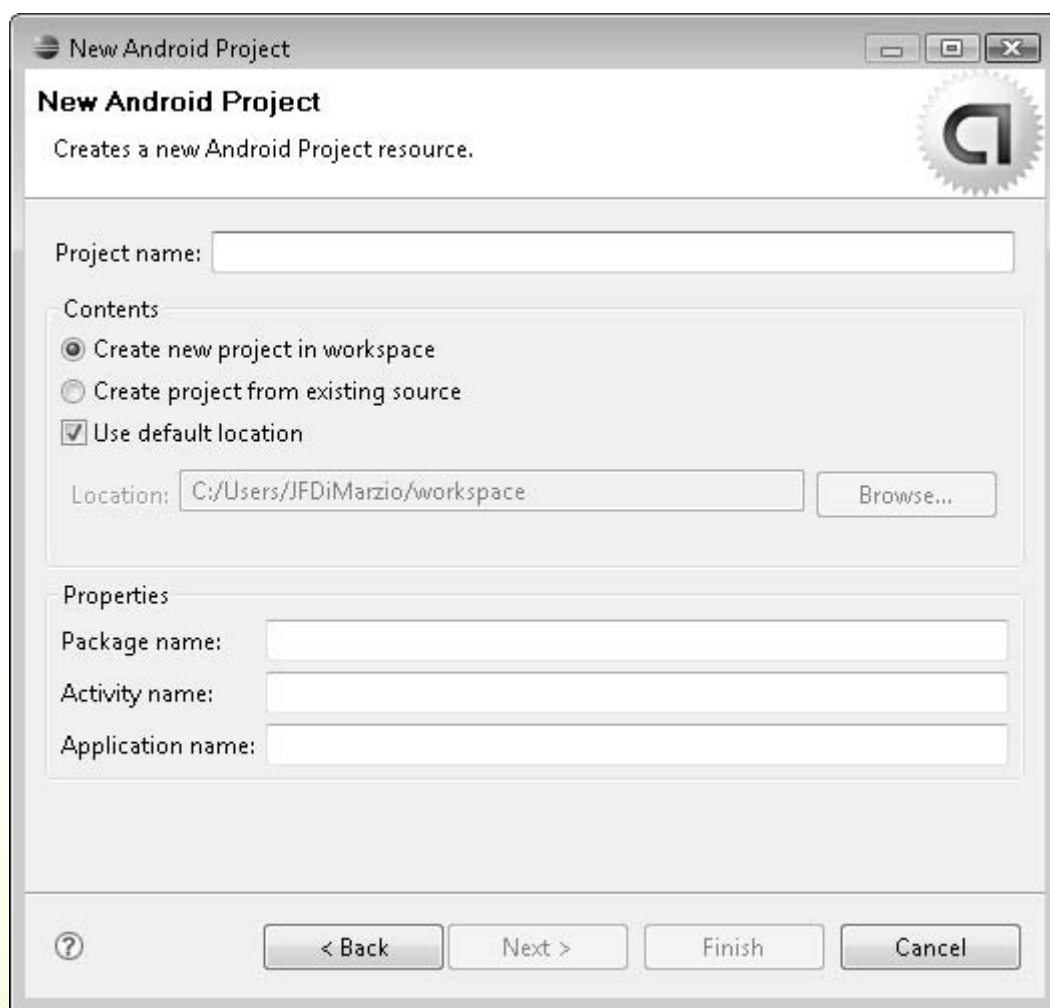


图 5-1 空的 Eclipse 开发环境

- 你第一个新工程的框架文件。这些框架文件包含一些重要的应用程序架构，在这些架构之上你将可以构建你的工程。和许多方式类似，像在 Visual Studio 中创建一个 Microsoft .NET 应用程序的方式在你的文件里自动生成一些 Windows 创建工程代码，使用 Eclipse 中的 Android Project 向导来自动生成你的初始工程文件和一些 Android 创建的代码。
新 Android 项目向导。

另外，新 Android Project 向导包括一些选项，下面所示，你必须设置它们来开始 Android 项目开发。



对于 Project Name 输入框域, 由于本例的目的, 使用 `HelloWorldText` 标题。这个名称足以让这个 HelloWorld! 和其他将在本章中创建的工程区分开。

在 Content 区域, 保持默认选项: "Create New Project in Workspace" 单选按钮应该被选中, 并且 "Use Default Location" 复选框应该被选择。这将允许 Eclipse 在你默认工作空间中创建你的工程项目。保持默认选项的好处是你的工程可以被放在一个使排序、管理、查找这些工程十分容易的重要地方。例如, 假如你在基于 Unix 环境下开发, 这个路径指向你的 \$HOME 目录。

如果你在 Microsoft Windows 环境开发, 工作空间的路径会是:
C: /Users/<username>/workspace, 如上图所示。

然而, 由于任何原因, 你可能不想选择 "Use Default Location" 复选框并为你工程选择一个不同的地方, 一个原因, 如果你想为这个与别的 Android 项目区别开的特殊项目选择一个位置, 你可能想要在这里指定一个不同的简单位置。例如, 你可能想让你在本书中创建的这些项目保存在一个和你将来自己创建的工程存放位置不同的地方。如果是这样, 简单地重写位置选项来为你的工程指定你自己的自定义位置目录。

另一方面, 如果你不选择 "Use this as Default" 并且不想在 Eclipse 启动 (如第二章最后部分的建议) 期间再被对话框问起: 选择一个默认工作空间; 你可能被请求指定一个工程位置。在 Eclipse 启动期间选择那个复选框则默认所有新的项目保存到 workspace 目录中 (并且在新 Android 工程向导中的 "Location" 输入框域中显示默认位置)。假如你在 Eclipse 启动过程期间没有选择这个复选框, 现在你需要为你的新项目选择一个路径, 通过点击 "Browse" 按钮并选入新路径。

新 Android Project 向导中最后的三个选项是在属性域中。这些属性定义了你的项目如何集成到 Android 环境中。在 "Package Name" 输入域中, 你给你的应用程序套件包指定名字空间。例如, android.app.Activity 或者 com.google.android.map.MapActivity。

警告:

Package 名称遵循标准 Java 包命名规则, 它使两个包用同名称发布的风险降低了。最顶级包名是公司的域标示符 (例如: com、org 和 net)。这被域名所遵循, 例如 google。最后, 包的内容描述标题就被提供了。由于本章的目的, 我 Hello World 应用程序的包名将省略 "com" 来识别它是一个文本应用程序并不是意味要被发布。所有未来在本书中创建的包将会被发布并使用 "com" 来识别。

对于 Hello world 文本应用程序, 使用包名 android-programmers-guide.HelloWorldText。这个名称唯一地标识这个代码属于这个应用程序并且让这个测试程序和别的你将在本书中开发的应用程序区别开来。

注意:

假如你正关注屏幕, 如你正在尝试一样, 当你键入包名时, 你将注意到一个错误信息出现在向导的顶部, 说明你必须填满所有的这些输入框来进行。这个错误信息过早并令人困惑, 因为你还没有尝试去填满其他在属性域中的输入框。如果你看到这样的错误信息, 只是别理它并继续, 完成下面两个属性域中的输入框。



下一个属性输入框，Activity 名称，是必须的，因为它是你应用程序主屏幕的引用。那就是，把 Activity 想成在你应用程序中显示的"窗口"。没有 Activity，你的应用程序将不能做任何事。然而，由于 Android 应用程序可以用多个 Activity 来创建，"New Android Project wizard"需要知道哪个 Activity 将会是默认的。Activity 名称是必须的并且没有默认值，因此，你必须提供一个来继续进行（像前面警告所指出的）。由于本示例的目的，使用 "HelloWorldText"。这个保持应用程序简单并只是如它需求一时的描述。

最后一个属性域，应用程序名称，指定你应用程序的名称。当它安装到设备上时，这个是被用来管理你的应用程序的名字。再次地，未来保持事情简单的缘故，用 "HelloWorldText" 作为这个程序的名称。下图显示了已完成填写的 "New Android Project wizard"。



New Android Project

Creates a new Android Project resource.

Project name: HelloWorldText

Contents

- Create new project in workspace
- Create project from existing source
- Use default location

Location: C:/Users/JFDiMarzio/workspace/HelloWorldText Browse...

Properties

Package name: android_programmers_guide.HelloWorldText

Activity name: HelloWorldText

Application name: HelloWorldText

< Back Next > Finish Cancel

贴士:

应用程序名称和 Activity 名称输入域不是必须相一致的。实际上,许多项目使用了更旧的习惯:一个应用程序"启动"屏幕通常被叫做 Main 或 Startup,使用任何你舒适的习惯。由于示范目的,本章假设你使用了建议的名称。

点击"Finish"来结束创建过程。向导运行一个后台进程,帮助一些必须文件自动生成,并创建支持 Android 应用程序项目需要的目录结构。当该过程完成,你将拥有你第一个 Android 应用程序项目,如图 5-2 所示。

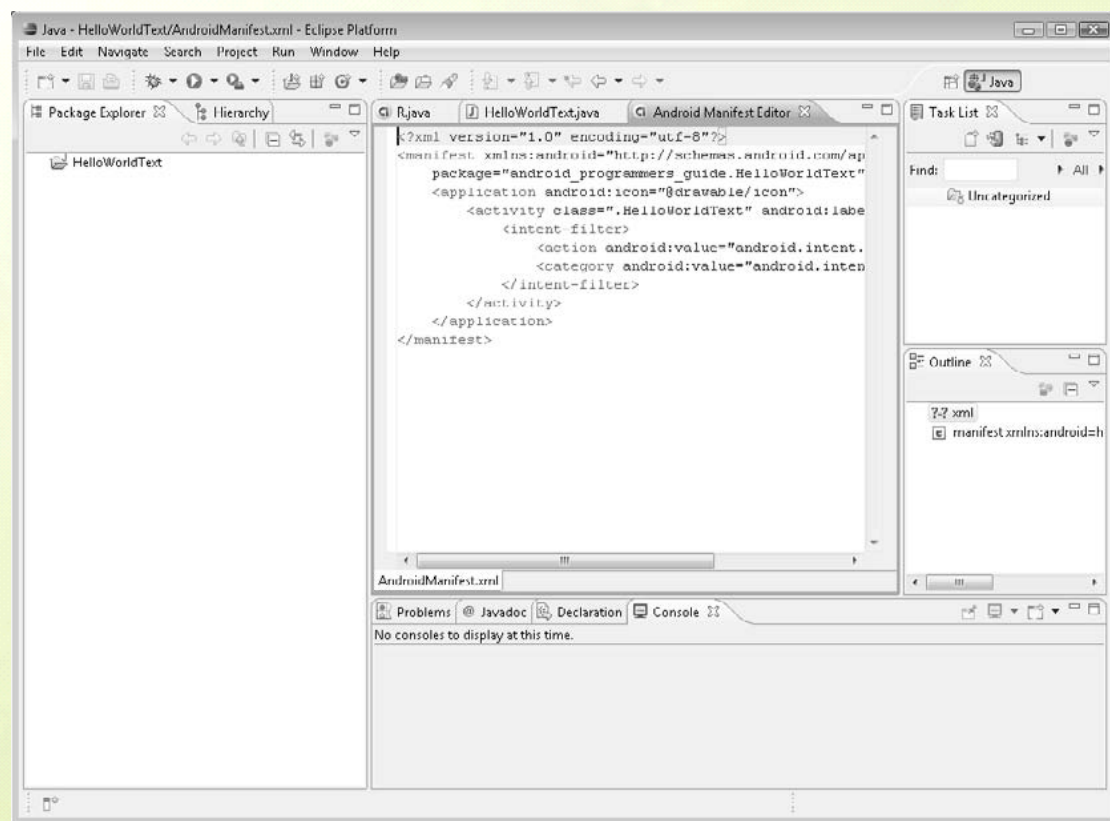


图 5-2 你第一个 Android 应用程序项目

贴士:

如果"Finish"按钮不能使用,你可能在这些属性域中的输入框里犯了错误。确保属性域被正确地填写,如果任何一个你键入的可能导致问题,Eclipse将不会允许你结束该过程。

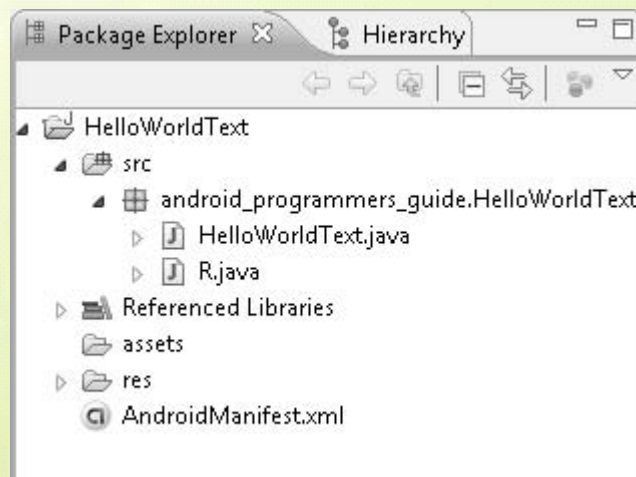
下一部分细述自动生成的 Android 文件内容和许多框架项目对于你的程序的目的。



细述 Android-生成文件

本部分讨论 Android 已经为你创建的新文件。一个相当健壮的架构已经为你创建了,并且如果你不知道你将要寻找什么,你可能最后在不应该放代码的地方添加了代码。有一些 Android 提供的文件你需要修改,也有一些你不应该修改;明白了区别可能会把你从不得不重新创建工程中拯救出来。

打开了你新程序,看一下包浏览器"Package Explorer",两个标签中的一个,位于主开发区域左边方框中。下图展示"Package Explorer"的外观。

**注意:**

如果"Package Explorer"没有被打开,你可以通过选择 Window| Show View|Package Explorer 来实现打开。

你应该看一下根目录，在本例中名叫 HelloWorldText。根目录是你所有工程文件的 home 或是容器。你的用户创建文件和 Android 自动生成文件将都被放到这个目录中，从 "Package Explorer" 中轻易访问得到。当前，应该有一些项目在根目录中。一个 AndroidManifest.Xml 文件，一个包含了相关类库的包，和三个目录 (res, assets 和 src)。这些项目在下一轮中讨论。

AndroidManifest.xml



AndroidManifest.xml 文件是你全局设定的地方。如果你是一个 ASP.NET 开发者，你能把 AndroidManifest.xml 文件想成 Web.config 和 Global.asax 中的一个。(假如你不是 ASP.NET 开发者，这意思是 AndroidManifest.xml 文件是存储设置的地方。) AndroidManifest.xml 文件包括像应用程序许可、Activity、和 IntentFilter 这样的设置。

标准的 AndroidManifest.xml 文件应该包含以下信息：

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android=http://schemas.android.com/apk/res/android
  package="testPackage.HelloWorldText">
  <application android:icon="@drawable/icon">
    <activity class=".HelloWorldText" android:label="@string/app_name">
      <intent-filter>
        <action android:value="android.intent.action.MAIN" />
        <category android:value="android.intent.category.LAUNCHER"
          />
      </intent-filter>
    </activity>
  </application>
</manifest>
```

当你创建未来的应用程序，你将会向这个文件中添加信息。记住，你提供的 package 名称在这里被列出了，你 Activity 将要捕获的 action 也列出了。

后面章节正在翻译中

