



Eclair



Froyo



Gingerbread



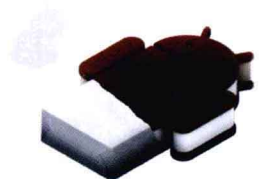
Donut



Cupcake



Honeycomb



IceCreamSandwich

Android

经典应用程序开发

■ 韩超 编著

具有清晰的主线，知识点全面，内容简洁实用
理论，文档和代码三者结合，以通用理念指引Android开发





Android

经典应用程序开发

电子工业出版社
Publishing House of Electronics Industry
北京·BEIJING

内 容 简 介

本书内容理论联系实际，系统化组织。全书由 8 章组成，涵盖了 Android 经典应用程序开发的各个方面，知识量巨大，技术要点清晰，结构紧凑而精炼。拥有本书后，结合 Android 公开的参考文档和示例代码，读者将可以基本掌握在 Android 经典应用层开发领域的所有方面。

本书要求读者具有一定的 Java 语言基础，具有 C/C++ 基础和其他 GUI 设计经验的工程师也可以阅读本书。初级读者可以通过本书逐步学习 Android 应用程序开发的各个方面；高级读者可以关注技巧和细节，并扫除理论上的盲区，提高开发质量。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有，侵权必究。

图书在版编目 (CIP) 数据

Android 经典应用程序开发 / 韩超编著. —北京：电子工业出版社，2012.2
ISBN 978-7-121-15586-4

I. ①A… II. ①韩… III. ①移动终端—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 271294 号

责任编辑：李 冰

印 刷：北京市顺义兴华印刷厂

装 订：三河市双峰印刷装订有限公司

出版发行：电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本：787×1092 1/16 印张：26.75 字数：663 千字

印 次：2012 年 2 月第 1 次印刷

印 数：4000 册 定价：59.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：(010) 88254888。

质量投诉请发邮件至 zltz@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：(010) 88258888。

前 言

Android 经典应用程序开发是 Android 开发的一个主要方面，本书在 Android 系统基本成熟的时刻推出，目标是为了引导和促进 Android 应用程序开发实现系统化、结构化，让尽量多的开发者少走弯路，尽快夯实基础，创造出更多、更好的 Android 程序。同时，本书内容尽量利用 Android 系统现有的资源，帮助大中华地区的读者跨越中-英文的鸿沟，让读者学会高效地使用英文文档的同时，又能继续发挥东方民族思维缜密严谨、结构性强的优势。

本书特点

本书紧紧抓住了 Android 经典应用程序开发。所谓经典应用程序开发，就是基于 Android 的标准 API 在 SDK 的环境下进行应用程序包（APK）的开发。这种开发模式下的开发成果可以作为 Android 系统的第三方的应用程序包，也可以作为 Android 产品中的应用程序。这是一种最基本、适用面最广的 Android 应用程序开发模式。

随着 Android 系统的发展，经历了 1.0、1.1 版本的初露锋芒阶段，1.5 版本里程碑式的程序阶段，1.6 版本的升级，2.0 版本的重要阶段，2.1、2.2 版本的升级，获得了 Android 2.3 版本。目前，Android 3.x 版本为更适合平板电脑的系统，且开源策略与从前相比有所变化。因此 Android 2.3 版本为用于移动系统的相对稳定而高级的版本。Android 4.0 作为 Android 系统最新的开源版本，其开发的技术方面依然和 Android 2.3 非常相似。Android 2.3 中的开发方式和理念可以在 Android 4.0 开发中使用。

本书作者参与了 Android 系统三年多发展过程中的系统开发和产品开发，对 Android 整体架构和发展理念有着深刻的认识。随着 Android 系统趋于稳定，本书作者也获得了一个将成熟的 Android 系统的经典应用开发结构清晰完整地展示给读者的契机。

本书具有以下特点：

- 有一条清晰的主线，从 UI 相关和 UI 无关两大方面进行把握；
- 将 Android 应用开发的细节方面有侧重点地展示给读者；
- 涵盖内容多，简练讲解其精华部分；
- 采用理论知识、API 参考文档和示例代码三者结合的方式；
- 按照使用者需求角度介绍程序开发的各个方面，而不受限于 Android 系统自身；
- 把通用 GUI 设计理念引入 Android 的应用开发中，便于具有其他系统基础的人员进入 Android 系统学习；
- 重视 Android 不同级别 API 层次，区分核心 API 和拓展 API，让读者对程序的兼容性有深刻认识。

本书在风格上与本书作者的其他偏重介绍底层的作品有明显的不同：本书以实用性

为主，不过分探寻原理性的内容。这种风格的选择和应用程序层开发的特点密切相关，作为应用程序的开发者，目的不是研究别人的系统如何实现，而是基于别人的系统开发自己的成果——Android 应用程序包。

本书内容

本书分成 8 章，各章的内容如下表所示。

章 次	主要 内容	主要 目标
第 1 章 Android 应用开发基础	应用程序的开发模式，IDE 开发环境的构建，Android 应用程序的结构，API 参考文档的使用	学会构建 IDE 环境，了解基本程序结构，学会使用参考文档
第 2 章 UI 程序的结构	基本 UI 程序的结构、活动组件和其内容视图的结合、控件的事件响应，设备事件的响应，屏幕间跳转，样式	深入了解 UI 部分的程序主体结构及编程方法
第 3 章 控件和布局	控件类的层次结构，自定义控件，视图组的结构，UI 界面的布局方式，自适应视图的使用，标签组织多界面	学会控件的使用和界面组织
第 4 章 图形接口	2D 图形系统结构和绘制，3D 图形系统的结构和绘制，可绘制内容，动画	学会使用比较底层自由图形绘制
第 5 章 非 UI 组件和结构	活动、广播接收器、内容提供者三种非 UI 的组件，工程文件、资源资产	深入了解非 UI 部分的程序主体结构及相关编程
第 6 章 Android 应用开发的细节	全局介绍 Android 所有接口，安全性和权限，数据存储，硬件相关接口的适应性，程序通知方法	学会主体结构之外的几个重要的程序要点
第 7 章 Android 应用的设计思想	程序的组件设计，程序的通信，UI 设计，进程线程设计，程序结构性设计	从通用 GUI 的设计实现角度学习 Android 的系统
第 8 章 Android 的应用层插件	在 Android 应用程序层的多种标准化插件的构建方法	学会 Android 特定的集成架构

在知识层次方面，第 1 章为基础部分，第 2 章～第 6 章为主体部分，第 7 章和第 8 章为提高部分。

在知识结构方面，第 1 章提供基本方法，第 2 章～第 4 章为 UI 相关的主干部分，第 5 章为 UI 不相关的主干部分，第 6 章是根据 Android 特点对前 5 章内容的补充，最后两章则站在通用的程序设计的角度来介绍 Android 应用层程序的开发。

本书读者

本书要求读者具有一定的 Java 语言基础的工程师，具有 C/C++ 基础和其他 GUI 设计经验的工程师也可以学习本书。

Android 第三方应用程序的开发者，开放应用方面的 Android 系统开发者，有利于构建一个完整的知识 Android 应用开发的知识结构。

对于基础较浅的读者，可能感觉本书的内容偏多，此时应采取的阅读方法是，根据目

录、小标题把握主干，首先掌握宏观内容。相信本书可以快速地帮助这类读者快速熟悉 Android 应用程序的开发。

对于有一定 Android 开发经验的读者，可能感觉本书的目录较简单。作者的目的是为了通过有条理的目录结构，将清晰的开发结构展现给读者，而不是故弄玄虚地制造晦涩的内容。在这些看似简单的目录背后，在每一章节的正文中，具有很多技巧、细节和注意点，这些微观的方面对有经验的读者提升技能也是有一定帮助的。

为了使本书在理论上尽量覆盖全面，本书的编写风格大都为列出理论纲要，再列出实例。对于希望先具有感性认识的读者，可以先学习实例，再回过头阅读前面的理论，最后再探索细节。

本书需要使用 Android 的应用程序开发环境和部分典型的示例代码，建议读者使用 Android 官方提供的版本。随着版本的升级，内容可能略有不同。建议读者参考如下网站：<http://developer.android.com/>。

本书作者

本书的编写工作由中国大陆兼具产品和理论经验的工程师韩超领衔，内容来源于开发团队在不同领域的开发经验的总结和归纳。韩超完成了本书内容的主要部分，参与本书编写的还有来自 Android 开发领域的梁泉、崔海斌、马若劫、于仕林等，机锋网(www.gfan.com)的张宇、张超、赵家维、黄亮、沈楨、徐威特等、国软时代(www.gredu.org.cn)的孙丽娜，以及杨钰、曹道刚等。广大社区开发者也对本书的成稿作出了贡献。

本书的核心作者团队大都来自开发一线，在工作之余完成本书。为了使本书尽早和读者见面，编写过程略显仓促。虽然经过审校，可能依然存在一些错误，敬请读者谅解。

目 录

第 1 章 Android 应用开发基础 1	
1.1 Android 应用开发基础概述..... 1	
1.1.1 Android 应用开发的结构..... 1	
1.1.2 Android 应用程序 开发的目的..... 2	
1.2 基于 Android SDK 的 IDE 开发环境..... 2	
1.2.1 Android IDE 开发环境的 组成和结构..... 2	
1.2.2 Windows 中的 Android SDK 开发环境安装..... 3	
1.2.3 Linux 环境中的 Android SDK 开发环境安装..... 10	
1.2.4 Android 中运行仿真器 环境..... 11	
1.2.5 Android 中建立工程..... 13	
1.2.6 在 IDE 中使用各种 Android 工具..... 20	
1.2.7 其他 Android 工具..... 26	
1.3 Android 应用程序的结构..... 27	
1.3.1 Android 应用程序的代码 组成..... 27	
1.3.2 Android 应用示例..... 28	
1.3.3 应用程序生成运行过程..... 31	
1.4 Android 应用开发的方法..... 33	
1.4.1 API 参考文档的使用..... 33	
1.4.2 Android 应用开发模式..... 37	
第 2 章 UI 程序的结构 39	
2.1 Android 中的界面组成..... 39	
2.1.1 屏幕（活动 Activity）..... 40	
2.1.2 屏幕中的内容 （视图 View）..... 41	
2.1.3 二者的结合..... 41	
2.2 屏幕中内容的控制和响应..... 42	
2.2.1 基本响应方法..... 43	
2.2.2 变化的响应方法..... 45	
2.2.3 控件响应方法比较..... 47	
2.3 设备事件的响应..... 48	
2.3.1 键盘事件的响应..... 48	
2.3.2 运动事件的处理..... 51	
2.4 屏幕切换..... 56	
2.4.1 启动活动的方法..... 57	
2.4.2 带有返回结果的启动活动..... 58	
2.5 菜单、对话框和标题栏..... 61	
2.5.1 菜单的使用..... 61	
2.5.2 对话框的使用..... 65	
2.5.3 标题栏的使用..... 73	
2.6 样式和主题的使用..... 74	
2.6.1 控件中的样式..... 75	
2.6.2 全局性质的主题..... 75	
第 3 章 控件和布局 81	
3.1 控件..... 81	
3.1.1 Android 中的控件..... 81	
3.1.2 文本类控件..... 84	
3.1.3 图像类控件..... 88	
3.1.4 进度条类控件..... 91	
3.1.5 继承 View 实现自定义控件..... 96	
3.1.6 继承控件实现自定义控件..... 101	
3.2 视图组和屏幕的布局..... 104	
3.2.1 Android 中的视图组..... 104	
3.2.2 独立使用的视图组..... 108	

3.2.3	作为简单容器使用的视图组	113
3.2.4	几种布局类的使用	116
3.2.5	自适应视图 AdapterView	131
3.2.6	Tab 的使用	147
第 4 章	图形接口	158
4.1	2D 图形接口的使用	158
4.1.1	使用 2D 图形接口的程序结构	158
4.1.2	画布和画笔结合的基本绘制	159
4.1.3	路径 (Path)	164
4.1.4	文本 (Text)	166
4.1.5	位图 (Bitmap)	171
4.1.6	颜色效果	174
4.1.7	画布状态和区域操作	177
4.1.8	记录	182
4.2	OpenGL 3D 图形接口的使用	184
4.2.1	使用 OpenGL 图形接口的程序结构	184
4.2.2	单纯实现的渲染器动画	185
4.2.3	具有交互的 OpenGL 绘制程序	189
4.2.4	背景和透明效果	192
4.2.5	另一种 OpenGL 的接口方式	196
4.3	可绘制内容	198
4.3.1	可绘制内容的概念	198
4.3.2	使用 XML 文件表示基本的可绘制内容	199
4.3.3	使用 XML 文件表示扩展的可绘制内容	202
4.3.4	使用代码构建的可绘制内容	207
4.4	动画	210
4.4.1	动画的概念和分类	210
4.4.2	补间动画	210
4.4.3	帧动画	215
4.4.4	屏幕切换的动画	216
第 5 章	非 UI 组件和结构	219
5.1	服务	219
5.1.1	服务综述	219
5.1.2	本地服务的调用者	222
5.1.3	远程服务的使用	228
5.1.4	服务的应用场景	236
5.2	广播接收器	238
5.2.1	广播接收器综述	238
5.2.2	不同形态的广播接收器	239
5.2.3	结合 PendingIntent 使用广播接收器	241
5.3	内容提供者	244
5.3.1	内容提供者综述	244
5.3.2	简单内容提供者的实现	248
5.3.3	URI 的处理	251
5.3.4	内容提供者的打开文件功能	252
5.4	工程描述文件、资源和资产	255
5.4.1	Android 中的工程描述文件	255
5.4.2	Android 中的资源	259
5.4.3	Android 中的资产	265
第 6 章	Android 应用开发的细节	268
6.1	Android API 层次结构	268
6.1.1	Android API 层次结构	268
6.1.2	Android 的 API 级别	268
6.1.3	Android 的各个基本的核心包	270
6.1.4	Android 1.5 增加的内容	274
6.1.5	Android 1.6 增加的内容	275
6.1.6	Android 2.0-2.1 增加的内容	276
6.1.7	Android 2.2 增加的内容	277

6.1.8	Android 2.3.x 增加的内容	278	7.4.2	线程	337
6.2	安全性和权限	280	7.4.3	Android 中的任务	342
6.2.1	进程的用户 id 级别的权限	280	7.5	UI 设计	348
6.2.2	许可	281	7.5.1	布局的组合	348
6.3	数据存储	283	7.5.2	组合型控件	352
6.3.1	Android 数据存储的方式	283	7.5.3	布局的动态使用	355
6.3.2	文件和网络	284	7.5.4	不同屏幕的适应性	357
6.3.3	数据库的使用	285	7.6	Android 的应用程序设计	
6.3.4	参数选项的使用	292		实现理念	359
6.4	硬件相关的程序	300	7.6.1	Android 程序的设计和实现	
6.4.1	硬件相关的类	301		的通用方法	359
6.4.2	硬件相关的特性	301	7.6.2	Android 程序的要点	361
6.4.3	电源管理器	303	7.6.3	Android 中几种典型的程序	
6.4.4	警报器的使用	304		结构	363
6.4.5	音频系统的使用	304	第 8 章	Android 应用程序层插件	366
6.4.6	定位系统的使用	305	8.1	Android 应用层插件的功能	
6.4.7	电话系统的使用	307		与作用	366
6.4.8	传感器的使用	310	8.1.1	应用层插件的特点	366
6.4.9	振动器的使用	312	8.1.2	应用层插件的结构	367
6.4.10	无线局域网的使用	313	8.1.3	主要应用层插件	368
6.4.11	蓝牙的使用	314	8.2	快捷方式 (Shortcut)	368
6.4.12	照相机的使用	316	8.2.1	Shortcut 的概念和用途	368
6.4.13	电池和电源信息的获取	318	8.2.2	Shortcut 的程序组成	369
6.5	用户通知方法	318	8.2.3	Shortcut 的实现	370
6.5.1	Toast 通知	318	8.3	活动目录 (LiveFolder)	372
6.5.2	Status Bar 通知	321	8.3.1	LiveFolder 的概念和用途	372
第 7 章	Android 应用的设计思想	325	8.3.2	LiveFolder 的程序组成	373
7.1	程序的组件模式	325	8.3.3	LiveFolder 的实现	374
7.2	系统的脉络: Intent	327	8.4	应用小部件 (App Widget)	377
7.2.1	Intent 的作用	327	8.4.1	App Widget 的概念和用途	377
7.2.2	代码中 Intent 的使用	328	8.4.2	App Widget 程序的组成	378
7.3	远程调用: Binder 和 AIDL	332	8.4.3	App Widget 的实现	380
7.3.1	应用层 IPC 的实现方式	332	8.5	输入法 (InputMethod)	383
7.3.2	AIDL 文件的使用	333	8.5.1	InputMethod 的概念和	
7.4	进程、线程和任务	334		用途	383
7.4.1	进程	335	8.5.2	InputMethod 的程序组成	384

8.5.3	InputMethod 的实现	388	8.8.1	Account 和 Sync 的概念和 用途	405
8.6	活动墙纸 (Live Wallpaper) ..	393	8.8.2	Account 和 Sync 的程序 组成	405
8.6.1	Live Wallpaper 的概念和 用途	393	8.8.3	Account 和 Sync 的实现	408
8.6.2	Live Wallpaper 的程序 组成	393	8.9	语音识别器 (SpeechRecognizer)	412
8.6.3	Live Wallpaper 的实现	395	8.9.1	SpeechRecognizer 的概念 和用途	412
8.7	可搜索程序 (Searchable)	397	8.9.2	SpeechRecognizer 的程序 组成	413
8.7.1	Searchable 的概念和用途	397	8.9.3	语音识别器实现	414
8.7.2	Searchable 的程序组成	399			
8.7.3	Searchable 的实现	400			
8.8	账户和同步 (Account 和 Sync)	405			

第1章

Android 应用开发基础

1.1 Android 应用开发基础概述

1.1.1 Android 应用开发的结构

Android 是 Google 开发的基于 Linux 平台的、开源的智能手机操作系统。Android 包括操作系统、中间件和应用程序。OHA（Open Handset Alliance，开放手机联盟）为 Google 与 33 家公司联手为 Android 移动平台系统的发展而组建的一个组织。

Android 的开发者可以在完备的开发环境中进行开发，Android 的官方网站也提供了丰富的文档、资料。这样有利于 Android 相关程序开发和运行在一个良好的生态环境中。

Android 应用程序开发是 Android 开发中最上面的一个层次。Android 应用程序的基础是 Android 提供的各个 Java 类，这些类也提供了 Android 平台级的 API。位于 Android 框架层之上的部分，为 Android 系统的应用程序，它们通常被组织成 Android 应用程序包（APK），可以运行于各个 Android 设备或者 Android 仿真器之上，如图 1-1 所示。

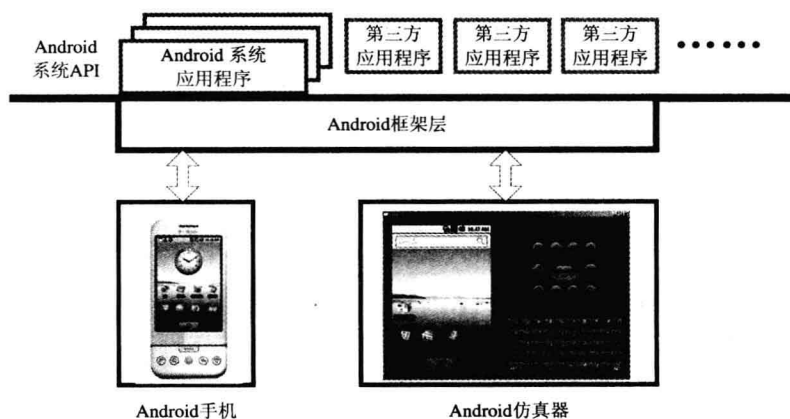


图 1-1 Android 应用的开发结构

Android 应用程序可以基于两种环境来开发：Android SDK 和 Android 源代码。Android SDK 环境可以采用 IDE 的开发环境，方便开发的流程。而使用源代码开发 Android 应用程序则可以获得更多的功能。

完全基于 Android 标准的 API 开发的 Android 应用程序包，可以被称为“Android 经典应用程序”，这种程序具有标准的兼容性，一般可以任意运行于支持了某个级别 API 的 Android 系统之上。

👉 1.1.2 Android 应用程序开发的目的


由于 Android 系统本身是一个开发的系统，而且具有标准的 API，因此 Android 应用程序包的开发主要有以下两种情况。

- 第一种：第三方应用程序；
- 第二种：具体系统的应用程序。

第一种情况是将某个具体的功能以应用程序包（APK）的方式实现，安装到各个不同的 Android 设备上。以这种模式开发的应用程序，就是第三方应用程序包，其特点是同样的一个应用程序包可以运行于不同的目标设备之上，因此尤其需要考虑兼容性问题。这种情况开发的应用程序包可能出于以下几种目的：发布到 Market 上可以作为交易的产品，某种功能的具体客户端，作为广告的载体，爱好者共享自己的开发程序。

第二种情况是随着某个具体的系统（一般是某个 Android 设备）所实现的应用程序包。这种情况实现的应用程序包不需要放到其他 Android 系统上运行，因此不需要考虑兼容性问题，只要能在当前系统中运行即可。另一方面，这种情况开发的应用程序可能依赖于具体系统的特定功能。这种情况的目的是让某个 Android 设备具有更多的功能。

无论是第三方应用程序还是具体系统的应用程序，都可以基于 SDK 或者源代码两种开发环境开发。相比较而言，后者使用源代码环境开发的情况更多。基于标准的 SDK 开发或者基于源代码但是只使用标准 API 的开发方式，属于经典 Android 应用程序。与之相对，如果使用经过改动的源代码生成的非标准 SDK，或者直接使用源代码环境开发，则可能使用到标准 API 以外的接口，这在带来更多功能的同时，也带来了兼容性的问题。

 **提示：**从 Android 应用程序开发的角度，无论使用何种环境（SDK 或者源代码），基于标准 API 开发的情况属于通常情况，也是 Android 应用程序开发的基本方式。

1.2 基于 Android SDK 的 IDE 开发环境

Android 的 SDK 开发环境使用预编译的系统，开发者可以基于 Android 的标准平台 API 进行应用程序层次的开发。在 SDK 的开发环境中，还可以使用 Eclipse 等作为 IDE 开发环境。

👉 1.2.1 Android IDE 开发环境的组成和结构

Android 的 IDE 开发环境需要以下的内容。

- JDK 环境（Java 开发包）；

- Eclipse 集成开发环境;
- ADT (Android Development Tools) 插件;
- Android SDK。

Android SDK 在 IDE 环境中使用的结构如图 1-2 所示。

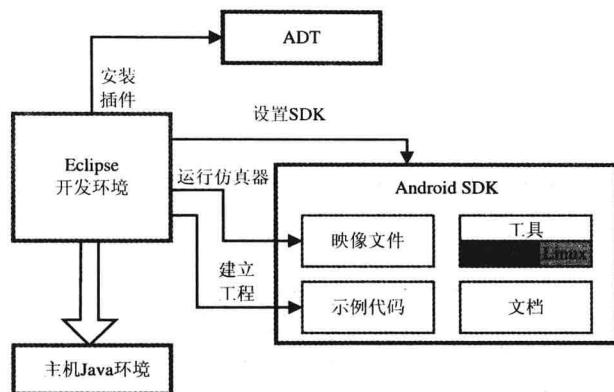


图 1-2 Android SDK 在 IDE 环境中使用的结构

Android 的 SDK 具有 Windows、Linux 和 MAC 三种不同版本，其中的基础映像文件、示例代码和文档都是相同的，它们之间的主要差别是主机的工具不同。例如，很多主机运行的工具在 Window 环境中为 .exe 文件，而在 Linux 环境中为 Elf 可执行程序；很多脚本在 Window 环境中是 .bat 批处理脚本，在 Linux 环境中是 shell 脚本。

1.2.2 Windows 中的 Android SDK 开发环境安装

1. 安装 JDK 基本 Java 环境

Eclipse 的运行需要依赖 JDK (Java Develop Kit, Java 开发包)，因此需要下载使用 JDK 的包并进行安装。JDK 1.6 版本使用一个可执行程序来安装，例如 `jdk-6u10-rc2-bin-b32-windows-i586-p-12_sep_2008.exe`，双击直接进行安装即可。

2. 安装 Eclipse

Eclipse 集成开发环境是开放的软件，可以到 Eclipse 的网站上去下载：

<http://www.eclipse.org/downloads/>

在 Android 系统发展的早期，各个版本的 Eclipse 在 Android 中均有所使用。经历了 Android 版本的多次升级过程后，目前推荐使用 Eclipse 3.5 以上的版本，较低版本 Eclipse 可能不能使用。

3. 在 Eclipse 中安装 ADT

ADT 的含义为 Android Development Tools (Android 开发工具包)，它主要具有以下一些功能：

- 可以从 Eclipse IDE 内部访问其他的 Android 开发工具。例如，ADT 可以直接从 Eclipse

访问 DDMS 工具的很多功能——屏幕截图、管理端口转发 (port-forwarding)、设置断点、观察线程和进程信息。

- 提供了一个新的项目向导 (New Project Wizard)，帮助快速生成和建立起新 Android 应用程序所需的最基本文件。
- 使构建 Android 应用程序的过程变得自动化、简单易行，提供了一个 Android 代码编辑器，可以帮助完成 Android 中各种源文件的编写。

第一步：进入 Eclipse 环境，选择 “Help” → “Install New Software”，准备安装插件，如图 1-3 所示。

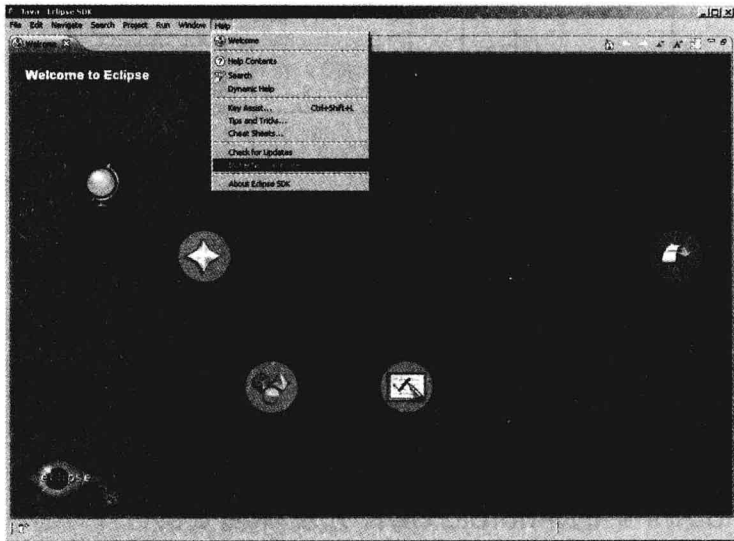


图 1-3 Eclipse 中选择安装新软件

第二步：选择后将出现软件升级对话框，如图 1-4 所示。

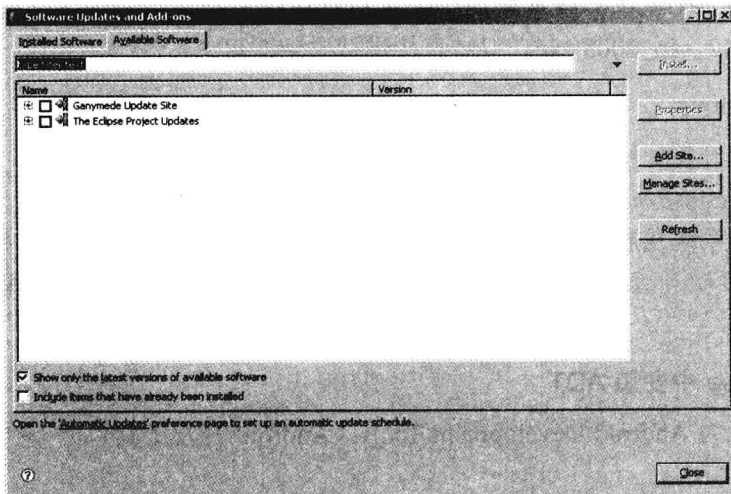


图 1-4 Eclipse 的软件升级对话框

以上界面出现后，单击右侧自上而下的第 3 个按钮“Add Site”，表示准备增加一些路径，在这个路径中获得插件，单击后界面如图 1-5 所示。

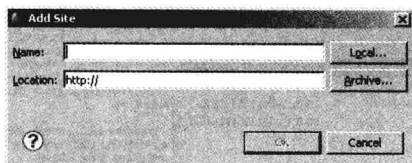


图 1-5 Eclipse 中增加 ADT 插件的路径

在“Add Site”对话框中，在 Location 文本框中输入 Android 插件的路径：

<https://dl-ssl.google.com/android/eclipse/>

以上的方式是通过一个网络地址获取 ADT 插件。除了这种方式以外，另外一种方式是使用本地文件。单击“Archive...”按钮，使用这种方式将直接指定磁盘中的一个 ADT 包文件（例如 ADT-12.0.0.zip 等）。

获得 ADT 插件文件包的方法，可以从如下的网站上下载：

<http://developer.android.com/sdk/eclipse-adt.html>

ADT 文件包的本质是一个 Eclipse 中使用的插件，由于是 Java 插件，因此在 Windows 和 Linux 环境中使用的 ADT 包相同。

ADT 文件本身是一个 ZIP 包，经过解压缩之后，其结构如下所示：

```
ADT-<version>/
|-- features           [特性 jar 的目录]
|-- plugins           [插件 jar 的目录]
|-- index.html        HTML 格式的网页描述文件
|-- site.xml          XML 格式的网站管理文件
`-- web               [关联网页]
```

ADT 文件包的 plugins 目录中包含了 com.android.ide.eclipse.adt_<version>.jar 等 Java 插件包。

在大多数情况下，由于 ADT 对其他包有依赖，即使使用本地磁盘中的 ADT 文件包，依然不能脱离网络环境完成安装。

第三步：ADT 相关的设置完成后，回到软件升级对话框，work with 的路径变为了 <https://dl-ssl.google.com/android/eclipse/>。后面的列表则变成了“Developer Tools”，即（项目就表示了）Android 插件包中的工具，其中包含了以下几个项目。

- Android DDMS;
- Android Development Tools ;
- Android Hierarchy Viewer;
- Android Traceview。

其中的主工具为 Android Development Tools，其他为辅助工具。一般情况下它们应该被一起安装。选择它们后，继续进行安装，出现的界面如图 1-6 所示。

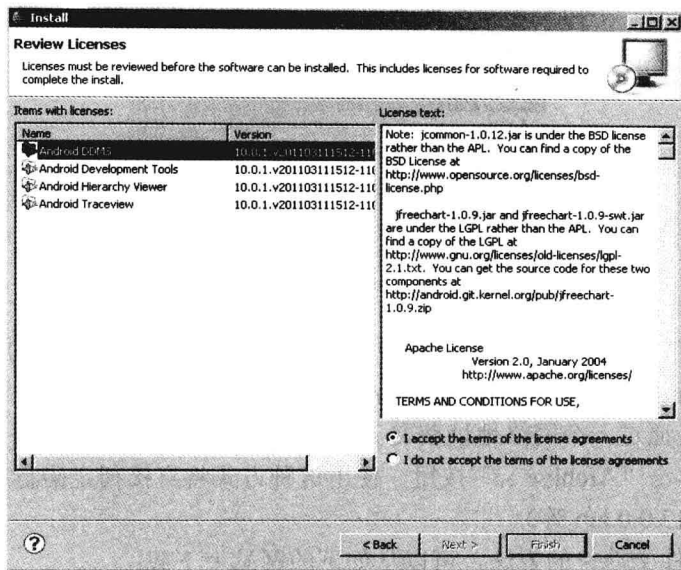


图 1-6 Eclipse 中选择安装 ADT

选择接受 (accept) 并且选择 Finish 完成安装之前的配置, 然后将进入安装 Android 组件的阶段。

在安装的过程中, 可能出现缺少某些包的情况, 如图 1-7 所示。

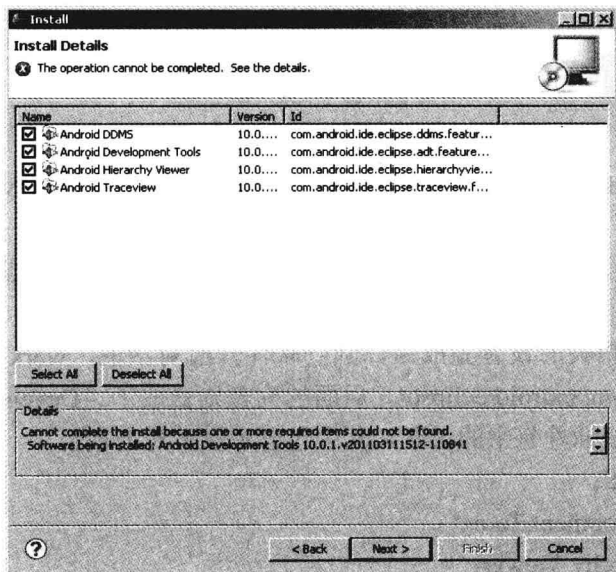


图 1-7 安装 Android 的插件

这时, 可以选择 “Window” → “Preference”, 从左侧的列表中选择 Install/Update 项目, 如图 1-8 所示。

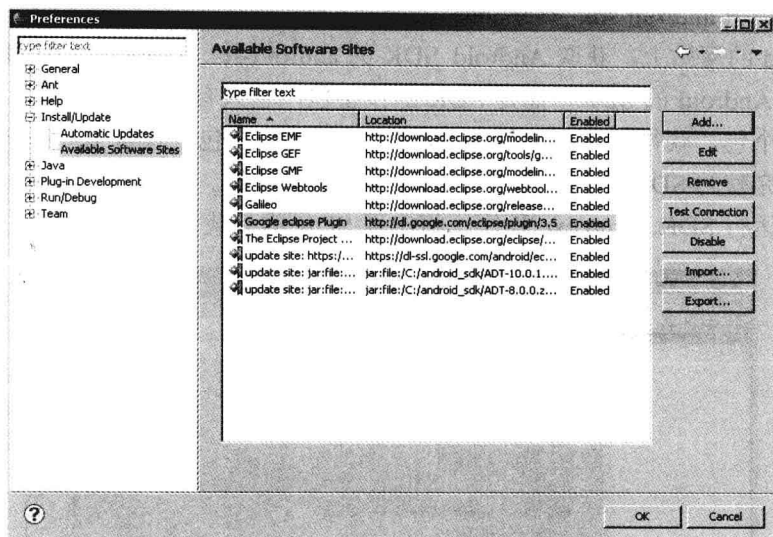


图 1-8 ADT 相关的插件

在项目中，增加以下内容：

Eclipse GEF，地址为：

<http://download.eclipse.org/tools/gef/updates/releases/>

Eclipse EMF，地址为：

<http://download.eclipse.org/modeling/emf/updates/releases/>

Eclipse GMF，地址为：

<http://download.eclipse.org/modeling/gmf/updates/releases>


Eclipse Webtools，地址为：

<http://download.eclipse.org/webtools/updates/>

Google eclipse Plugin，地址为：

<http://dl.google.com/eclipse/plugin/3.5>

第四步：安装完成，关闭并重新启动 Eclipse。再次进入 Eclipse 后，将发现 ADT 已被安装。

 提示：ADT 包是 Android 官方升级较快的内容，从最初 2009 年 10 月的 ADT 0.9.4 到 2011 年的 12.0.0，其间推出过十余个版本。新版本的功能更强，且新版本也可以支持旧版本 SDK。因此，无论 SDK 的版本是哪个，一般情况下都推荐使用最新版本的 ADT。

4. 获得 Android SDK

Android SDK 可以在 Android 的网站上进行下载，相关部分的网址为：

<http://developer.android.com/sdk/>

早期 Android 的 SDK 获得方式是使用直接下载包，这个包通常有几百兆的大小。在目前的 Android 开发中，使用下载工具包+在线升级的方式。