

Eclipse 完全手册

——基础 · 进阶 · 高级

周竞涛 主编

赵寒 王明微 李山 等著



光盘包含：
本书所有实例代码



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

一 Java 技术大系

一 **Eclipse 完全手册——基础 · 进阶 · 高级**

阅读此书，**快速掌握 Eclipse** 集成开发环境，**充分开发** 专业应用程序。

一 **基础篇**



第 1 章 Eclipse 入门
第 3 章 Java 开发工具

第 2 章 Eclipse 基础操作
第 4 章 调试 Java 代码

一 **进阶篇**



第 5 章 在 Eclipse 中使用 CVS
第 7 章 Eclipse SWT/JFace 开发
第 9 章 在 Eclipse 中应用 J2EE

第 6 章 JUnit 与单元测试
第 8 章 Web 开发
第 10 章 基于 Struts 的
Web 开发技术

一 **高级篇**



第 11 章 Eclipse 架构综述
第 13 章 Eclipse 扩展基础
第 14 章 拥有个性化的视图、编辑器和透视图
第 15 章 JDT 技术内幕
第 16 章 资源管理中应用 Natures 和 Builders
第 17 章 Eclipse RCP 程序开发

第 12 章 Eclipse 插件开发
第 18 章 插件大全



配书光盘附带了全书所有实例源代码，读者只需导入即可编译运行，有助于读者更快地掌握 Eclipse 集成开发环境。

图书分类：编程语言>Java技术

ISBN 7-121-02807-7



9 787121 028076 >



网上订购：www.dearbook.com.cn
第二书店 · 第一服务



责任编辑：高洪霞
责任美编：张子建
崔云霞

本书贴有激光防伪标志，凡没有防伪标志者，属盗版图书。
ISBN 7-121-02807-7 定价：55.00元（含光盘1张）

Java技术大系

Eclipse 完全手册

——基础·进阶·高级

周竞涛 主编

赵寒 王明微 李山 等著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

全书本着“求全求精”的宗旨,力求全面而系统地介绍 Eclipse 开发环境对各种 Java 应用开发的支持。全书共分三篇,“基础篇”介绍了 Eclipse 及相关插件的安装、基础操作、Java 开发工具以及代码调试等知识;“进阶篇”介绍了 Eclipse 中 CVS、JUnit、SWT/JFace、Web 相关技术的使用及相关插件;“高级篇”介绍了 Eclipse 体系架构、插件开发流程、Eclipse 扩展基础、JDT 技术内幕以及资源管理中 Natures 和 Builders 的应用等知识。

本书适合 Java 程序员及各类软件开发人员阅读和参考,为了便于不同层面的读者根据兴趣进行选择性的阅读,全书的大部分章节都力争做到了相对独立,读者可以将本书作为参考书使用。

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

Eclipse 完全手册:基础·进阶·高级 / 周竞涛主编. —北京:电子工业出版社, 2006.8

(Java 技术大系)

ISBN 7-121-02807-7

I. E… II. 周… III. 软件工具—程序设计 IV. TP311.56

中国版本图书馆 CIP 数据核字(2006)第 068184 号

责任编辑:高洪霞

印 刷:北京天宇星印刷厂

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

经 销:各地新华书店

开 本:850×1168 1/16 印张:32.5 字数:935 千字

印 次:2006 年 8 月第 1 次印刷

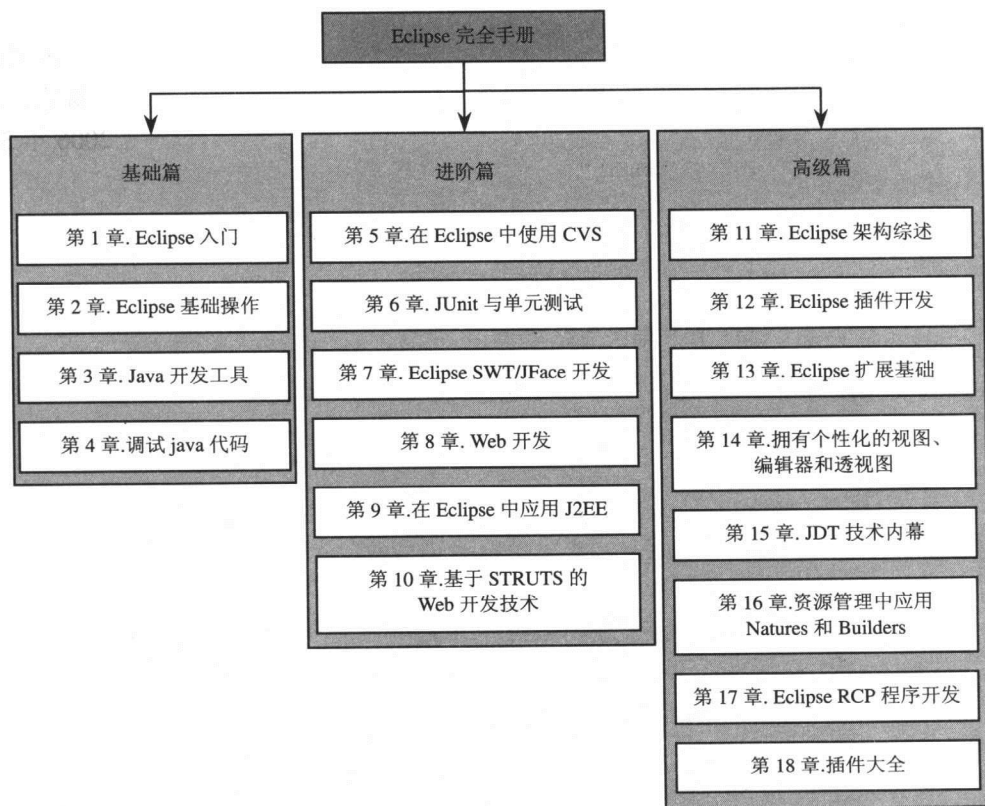
印 数:5000 册 定价:55.00 元(含光盘 1 张)

凡购买电子工业出版社的图书,如有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系。联系电话:(010) 68279077。质量投诉请发邮件至 zltts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

自序

本书题为“Eclipse 完全手册”，当初确有求全求精之意。与一些流行的商品化 IDE 相比，Eclipse 已经远远超出了 IDE 的范畴，它代表着一种思想、一套模式和一系列软件方法论的实践。Eclipse 所涉及的领域之广，内容之深可谓高山仰止！因此，既要求其“全”，又不能失其“精髓”，一直是撰写本书所必须面对的主要矛盾。求“全”固然困难，求“精”亦是不易。这也就造成了全书各个章节在“全”与“精”上权衡和取舍的偏重不同：有的章节保证论述周全细致，有的章节则力争论述得深刻精辟。当然，其结果还有待读者们的检验和赏评。

由于全书是将 Eclipse 作为 Java IDE 来展开讨论的，因此要求读者阅读本书前具有一定的 Java 开发知识。为了便于不同层面的读者根据兴趣进行有选择性的阅读，全书的大部分章节都力争做到了相对独立，读者可以将本书当成参考书，只在需要时翻阅某一章，因此了解一下本书的组织结构对读者会有所帮助。全书共分三篇，“基础篇”介绍了 Eclipse 及相关插件的安装、基础操作、Java 开发工具以及代码调试等知识；“进阶篇”介绍了 Eclipse 中 CVS、JUnit、SWT/JFace、Web 相关技术的使用及相关插件；“高级篇”介绍了 Eclipse 体系架构、插件开发流程、Eclipse 扩展基础、JDT 技术内幕以及资源管理中 Natures 和 Builders 的应用等知识。



全书是使用 Eclipse 构建 Java 程序的综合性参考书，是为那些希望充分利用 Eclipse 系统特点开发专业级应用程序的开发人员准备的。本书适合处在不同阶段的 Java 程序开发和设计人员以及任何希望用 Eclipse 进行软件开发的人员阅读和参考。书中一些高级主题蕴含了大量 Eclipse 的设计精髓，也可

以作为大型软件设计的有益参考。书中的光盘附带了全书所有的例子，读者只需导入即可编译运行，有助于读者更快地掌握 Eclipse 集成开发环境。

回首编写本书的半年时光，可以说是苦乐参半，个中滋味，颇不足为外人道也。对于作者来说，编写如此大部头的一部 Eclipse 专著，确实是一种考验。但是出于对 Eclipse 中软件模式的热爱和对 Eclipse 前景的看好，还是按计划完成了全书的写作。正如前文所述，Eclipse 已经超出了一种单纯作为 IDE 的范畴。因此，作者力求不落窠臼，将相当一部分笔墨花费到了 Eclipse 本身的结构和对它的扩展之上，力图通过这些斑斑点点，使读者能够窥见其背后的思想、模式和软件方法论的全豹。在这个过程中，作者受益匪浅。如果对读者来说也有一定的裨益，那将使作者颇感欣慰。

本书第 1 章、第 2 章和第 3 章由张超完成编写；第 4 章、第 5 章、第 6 章、第 9 章、第 10 章和第 16 章由董小峰完成编写；第 7 章、第 12 章和第 17 章由王克飞完成编写；第 11 章、第 13 章、第 14 章和第 15 章由赵寒完成编写。王明微和李山负责全书的校对和文字方面的润色。最后由周竞涛完成全书的统稿和修订。本书成书过程中，一直受到西北工业大学张树生教授的全力支持。哈尔滨工业大学硕士研究生田金华，西北工业大学硕士研究生冯赧、杨敏、常潇月帮助调试了本书中用到的部分源代码。本书中采用的参考资料，除了出自 Eclipse 的联机帮助和 www.Eclipse.org 提供的大量资料、技术报告和示例工程之外，其他参考资料来自于 IBM 的 Eclipse 开发者论坛，以及国内外相关的论文、书籍和期刊，在这里谨向这些提供帮助的作者表示衷心的感谢。

由于时间仓促，加上作者水平所限，本书中还存在许多不足。诚恳地欢迎读者批评指正。

本书编写组
执笔人：周竞涛
2006 年 3 月于西安

目 录

第一篇 Eclipse 基础

第 1 章 Eclipse 入门2

1.1 什么是 Eclipse2

1.2 Eclipse 大事记3

1.3 Eclipse 的安装5

1.3.1 JDK 下载安装与配置5

1.3.2 Eclipse 下载与安装7

1.4 本章小结8

第 2 章 Eclipse 基础操作9

2.1 透视图操作 (Perspective)9

2.1.1 打开透视图10

2.1.2 定制透视图10

2.1.3 第一个项目11

2.2 编辑器操作12

2.3 视图操作14

2.3.1 导航器视图 (Navigator view)14

2.3.2 任务视图16

2.3.3 书签视图17

2.4 资源管理18

2.4.1 资源概述18

2.4.2 本地历史记录20

2.4.3 比较和替换资源21

2.4.4 恢复已删除资源21

2.4.5 工作集22

2.4.6 查找资源22

2.4.7 导入/导出资源24

2.4.8 移动资源26

2.5 本章小结27

第 3 章 Java 开发工具28

3.1 JDT 的基本操作28

3.1.1 查看 JRE 安装和类路径变量28

3.1.2 创建 HelloWorld 项目29

3.1.3 创建 JUnit 项目32

3.2 JDT 相关视图基本操作34

3.2.1 包资源管理器视图34

3.2.2 层次结构视图35

3.3 Java 编辑器的基本操作36

3.3.1 打开 Java 编辑器37

3.3.2 添加新方法38

3.3.3 使用内容辅助功能39

3.3.4 标识代码中的问题40

3.3.5 使用快速修正功能41

3.3.6 使用模板编程42

3.3.7 组织导入语句44

3.3.8 创建和查看 Javadoc 文档45

3.4 重构支持48

3.4.1 抽取新方法48

3.4.2 重命名 Java 元素49

3.4.3 移动和复制 Java 元素50

3.5 Java 搜索51

3.5.1 使用搜索对话框执行 Java 搜索51

3.5.2 使用弹出菜单执行 Java 搜索51

3.5.3 搜索操作实践52

3.6 运行 Java 应用程序54

3.6.1 运行程序54

3.6.2 对代码段和表达式求值56

3.7 使用 JRE59

3.8 本章小结61

第 4 章 调试 Java 代码62

4.1 初识 Java 调试器62

4.2 Eclipse 调试视图64

4.2.1 调试视图64

4.2.2 变量视图65

4.2.3 断点66

4.2.4 表达式和显示视图69

4.2.5 代码片段编辑测试窗70

4.3 热替换72

4.4 本地调试73

4.4.1 调试启动配置73

4.4.2 暂挂线程75

4.4.3 单步遍历	75
4.5 远程调试	76
4.6 调试使用技巧	77
4.6.1 控制控制台	77
4.6.2 详细信息格式化程序	78
4.6.3 命令行详细信息	78
4.6.4 堆栈跟踪超链接	78
4.7 调试其他语言	78
4.8 本章小结	79

第二篇 Eclipse 进阶篇

第 5 章 在 Eclipse 中使用 CVS	82
5.1 资源控制的必要性	82
5.2 Eclipse 团队支持	82
5.2.1 Eclipse 团队支持的基本原则	82
5.2.2 团队支持配置	83
5.2.3 Eclipse 本地版本控制	84
5.3 CVS 概述	86
5.3.1 CVS 基本原理	86
5.3.2 CVS 之不能	88
5.4 CVS 服务器	89
5.4.1 安装服务器端	89
5.4.2 配置 CVS 服务器	89
5.5 在 Eclipse 中使用 CVS	91
5.5.1 设置 CVS 首选项	91
5.5.2 使用 CVS	94
5.5.3 在 CVS 的控制下工作	102
5.5.4 版本标签和分支	114
5.6 第三方资源管理控制程序	116
5.7 本章小结	117
第 6 章 JUnit 与单元测试	118
6.1 软件测试基础理论	118
6.1.1 什么是测试	118
6.1.2 测试驱动的开发与极限 编程 (XP)	118
6.1.3 单元测试	119
6.1.4 JUnit 简介	122
6.2 在 Eclipse 中使用 JUnit	144
6.3 本章小结	146

第 7 章 Eclipse SWT/JFace 开发	147
7.1 Java GUI 开发现状	147
7.1.1 Java 的不足	147
7.1.2 AWT 与 Swing 简介	147
7.2 Eclipse SWT 界面工具包	148
7.2.1 SWT 的诞生	148
7.2.2 第一个 SWT 程序	149
7.2.3 SWT 程序剖析	154
7.2.4 SWT 常用部件	155
7.2.5 SWT 布局方式管理	162
7.2.6 系统资源管理	167
7.3 SWT 与多线程	168
7.3.1 多线程编程	168
7.3.2 Java 线程安全	173
7.3.3 SWT 线程安全性新机制	175
7.4 Eclipse JFace 界面工具包	177
7.4.1 JFace 是什么	177
7.4.2 JFace 程序剖析	177
7.4.3 JFace 事件处理机制	179
7.5 SWT/JFace 高级特性	181
7.5.1 OLE 与 ActiveX	181
7.5.2 OLE 文档应用程序实例	185
7.6 本章小结	190
第 8 章 Web 开发	191
8.1 Web 服务器	191
8.1.1 Web 服务器的实现	191
8.1.2 Tomcat 安装	192
8.1.3 Tomcat 配置	192
8.1.4 在 Eclipse 中使用 Tomcat	197
8.2 Eclipse 中开发 Web 应用	201
8.2.1 新建 Tomcat 项目	201
8.2.2 Servlet 开发	202
8.2.3 Lombok 插件	206
8.2.4 JSP 开发	211
8.3 本章小结	219
第 9 章 在 Eclipse 中应用 J2EE	220
9.1 J2EE 和 Eclipse	220
9.2 J2EE 应用服务器	220
9.2.1 服务器浏览	221
9.2.2 下载和安装 JBoss	222

9.2.3 在 Eclipse 中配置 JBoss	223
9.3 创建 J2EE 项目	223
9.4 开发 EJB	226
9.4.1 EJB 简介	226
9.4.2 Xdoclet 介绍	226
9.4.3 配置 Lomboz	226
9.4.4 更改服务器	228
9.4.5 生成 EJB 文件	229
9.4.6 添加 EJB 方法	231
9.4.7 构建 EJB 接口类	232
9.4.8 EJB 部署	234
9.4.9 运行测试	235
9.5 本章小结	237
第 10 章 基于 Struts 的 Web 开发技术	238
10.1 Struts 简介	238
10.1.1 什么是 Struts	238
10.1.2 为什么用 Struts	238
10.2 Struts 框架	239
10.2.1 MVC 模式	240
10.2.2 Struts 体系结构	240
10.2.3 Struts 框架基本组件包	241
10.3 构建 Struts	242
10.3.1 构建模型	242
10.3.2 构建视图	242
10.3.3 构建控制器	244
10.4 Struts 应用的配置	244
10.5 Struts 控制流	249
10.6 基于 Struts 框架的开发流程	249
10.7 Struts 分析	250
10.7.1 强项	250
10.7.2 弱项	250
10.8 在 Eclipse 中开发 Struts	250
10.8.1 StrutsIDE 插件	251
10.8.2 创建一个 Struts 项目	251
10.8.3 Struts 示例	253
10.9 本章小结	259

第三篇 Eclipse 高级篇

第 11 章 Eclipse 架构综述	262
11.1 Eclipse 体系架构	262

11.1.1 组成结构	262
11.1.2 RCP 核心组件	263
11.1.3 RCP 可选组件	265
11.1.4 集成开发环境组件	266
11.2 Eclipse 插件模型	266
11.2.1 插件内部	266
11.2.2 Eclipse 运行时	267
11.2.3 Eclipse 3.0 插件和 OSGi Bundle	267
11.2.4 插件组织结构	270
11.2.5 扩展中的参与者	272
11.2.6 插件和扩展对象的关系	274
11.2.7 扩展点模式	274
11.2.8 扩展处理过程	276
11.2.9 清单详解	278
11.3 本章小结	282
第 12 章 Eclipse 插件开发	283
12.1 初识插件	283
12.1.1 what 和 why	283
12.1.2 插件的构成	283
12.2 插件的 Helloworld	285
12.3 使用 PDE 开发插件	289
12.3.1 PDE 简介	289
12.3.2 PDE 插件开发实例	298
12.4 Eclipse 插件的高级特性	303
12.4.1 插件片段	303
12.4.2 插件功能部件	304
12.4.3 定义插件的扩展点	304
12.5 本章小结	310
第 13 章 Eclipse 扩展基础	311
13.1 用户界面基础	311
13.2 工作台扩展点	312
13.3 扩展操作基础	313
13.3.1 操作 API	313
13.3.2 操作的共有特征	314
13.3.3 菜单和工具条属性	315
13.3.4 使用过滤器	317
13.4 工作台菜单和工具栏	319
13.4.1 定义为透视图的成员	319
13.4.2 通过用户指定	319
13.4.3 独立于透视图的菜单和工具栏	321

13.5	视图菜单和工具栏	322	14.4	本章小结	368
13.6	编辑器菜单和工具栏	323	第 15 章	JDT 技术内幕	369
13.7	视图和编辑器的上下文菜单	324	15.1	JDT 介绍	369
13.7.1	视图上下文菜单	324	15.2	Java 模型	370
13.7.2	添加上下文菜单到视图 中的对象	325	15.3	JDT 模型核心要素	371
13.7.3	编辑器上下文菜单	326	15.3.1	JavaCore 类	371
13.8	本章小结	327	15.3.2	JavaUI 类	374
第 14 章	拥有个性化的视图、编辑器 和透视图	328	15.3.3	ToolFactory 类	375
14.1	视图	328	15.4	分析 Java 源代码	375
14.1.1	视图基础	328	15.5	修改 Java 源代码	386
14.1.2	一个简单的示例	330	15.5.1	使用 AST	386
14.1.3	创建视图	332	15.5.2	使用工作副本	387
14.1.4	声明视图扩展	333	15.6	JDT 对 Eclipse 的扩展	389
14.1.5	创建视图模型	334	15.7	扩展 JDT 用户界面	389
14.1.6	创建视图类	335	15.7.1	视图操作	390
14.1.7	创建内容提供者	336	15.7.2	编辑器操作	390
14.1.8	创建标记提供者	337	15.7.3	Java 元素上下文菜单	391
14.1.9	关联视图与模型	338	15.8	本章小结	391
14.1.10	用户选择	338	第 16 章	资源管理中应用 Natures 和 Builders	392
14.1.11	添加操作	339	16.1	Eclipse 中的资源管理	392
14.1.12	视图交互	343	16.1.1	定制工程进程	392
14.2	编辑器	344	16.1.2	扩展工作空间的资源进程框架	393
14.2.1	编辑器基础	344	16.1.3	工程和工程描述	393
14.2.2	创建编辑器	344	16.2	应用 Natures	394
14.2.3	清单文件	345	16.2.1	创建 Nature 支持工程配置	395
14.2.4	向导	347	16.2.2	在工程中添加 Nature	396
14.2.5	编辑器类	349	16.2.3	使用图示标识项目性质	397
14.2.6	初始化编辑器	350	16.3	应用 Builder	397
14.2.7	修改和保存编辑内容	350	16.3.1	定义构建器	398
14.2.8	添加编辑器操作	351	16.3.2	构建过程	399
14.2.9	添加语法着色	353	16.3.3	Listener 和 Builder	399
14.2.10	添加内容辅助	355	16.3.4	关联 Nature 和 Builder	400
14.2.11	首选项页面	357	16.4	本章小结	401
14.2.12	属性页	360	第 17 章	Eclipse RCP 程序开发	402
14.2.13	编辑器和内容大纲窗口	363	17.1	认识 RCP	402
14.3	透视图	364	17.1.1	什么是 RCP	402
14.3.1	自定义透视图	365	17.1.2	RCP 能做什么	402
14.3.2	扩展已有透视图	367	17.2	RCP 如何运行	403

17.2.1	Eclipse 与 OGSi 框架	403	18.10.1	RamanVideoPlayer 播放器插件	454
17.2.2	RCP 应用程序模型	404	18.10.2	Player 插件套	455
17.2.3	RCP 应用程序的.plugin 文件	404	18.10.3	SweatAmp 插件	457
17.2.4	设置 RCP 程序的外观	405	18.10.4	Eclipse games 插件	458
17.3	创建 RCP 应用	406	18.11	Continuous Testing 插件	460
17.3.1	建立工程	406	18.12	Eclipse UML2 插件	463
17.3.2	创建 Product 配置文件	408	18.13	Log4E 插件	472
17.3.3	创建产品图标	410	18.14	Flow4J 插件	473
17.3.4	创建产品 About 图片和文字	412	18.14.1	插件介绍	473
17.3.5	替换闪屏图片	414	18.14.2	Flow4J 使用方法	474
17.3.6	定制产品的风格	414	18.15	Pattern Box 插件	477
17.3.7	发布产品	415	18.16	JspFormat	480
17.3.8	选择 JRE	416	18.17	XMLBuddy 插件	480
17.3.9	创建自己的启动程序	416	18.18	EMF 插件	482
17.4	一个更复杂的例子	417	18.18.1	插件介绍	482
17.4.1	RCP 程序的部署和发布	428	18.18.2	使用方法	483
17.4.2	导出 RCP 应用程序	429	18.19	EclipseHTMLEditor 插件	486
17.5	本章小结	430	18.20	JBossIDE 插件	489
第 18 章	插件大全	431	18.20.1	插件介绍	489
18.1	Ant Build file Creator 插件	432	18.20.2	环境	489
18.2	Doclipse 插件	433	18.20.3	安装	489
18.3	viPlugin 插件	436	18.20.4	在 Eclipse 中应用	489
18.4	ColorEditor 插件	437	18.21	DbEdit 插件	500
18.5	CDT 插件	439	18.21.1	插件介绍	500
18.6	EclipseME 插件	445	18.21.2	环境	500
18.7	Code Analysis 插件	449	18.21.3	安装	500
18.8	Designer 插件	450	18.21.4	在 Eclipse 中的应用	500
18.9	Visual Editor 插件	451			
18.10	娱乐类插件	454			



Eclipse 基础

- ◇ 第 1 章 Eclipse 入门
- ◇ 第 2 章 Eclipse 基础操作
- ◇ 第 3 章 Java 开发工具
- ◇ 第 4 章 调试 Java 代码

Eclipse 入门

1.1 什么是 Eclipse

Eclipse 是一个开放源码的、可扩展的应用开发平台，该平台为编程人员提供了一流的 Java 集成开发环境。作为一套开源工具，可用于构建 Web Services、J2EE 等各种类型的应用，其所提供的功能不亚于、甚至要超过由专业的集成环境供应商所提供的商业化产品，如 JBuilder。

Eclipse 最有魅力的地方就在于它的插件体系结构。在这个体系中重要的概念是扩展点（extension points），也就是为插件提供的接口。每一个插件都是在现有的扩展点上开发的，并可能还留有自己的扩展点，以便在这个插件上继续开发。

由于有了插件，Eclipse 系统的核心部分在启动的时候要完成的工作十分简单：启动平台的基础部分和查找系统的插件。在 Eclipse 中实现的绝大部分功能是由相应的插件完成的，比如 WrokBench UI 插件完成界面的外观显示，Resource Management 插件完成维护或生成项目或文件等资源管理工作，而 Version and Configuration Management（VCM）插件则负责完成版本控制功能，等等。虽然以上提到的每一个功能都是绝大多数 IDE 环境所必备的功能，Eclipse 却把它们都做成了插件模式，甚至用来开发 Java 程序的开发环境（Java development tooling, JDT），也只不过是 Eclipse 系统中的一个普通插件而已。整个 Eclipse 体系结构就像一个大拼图，可以不断地向上加插件，同时，现有插件上还可以再加插件。

虽然大多数用户很乐于将 Eclipse 当做 Java IDE 来使用，但 Eclipse 的目标不仅限于此。Eclipse 平台为工具提供者（Tools Provider）提供一套使用机制和一组需要遵循的规则，从而使得开发出的工具之间可以实现无缝的集成。这些机制通过定义良好的 API 接口、类和方法提供给用户使用，平台同样为新的工具的开发提供强有力的组件支持（如 Plug-in Development Environment, PDE——插件开发环境）。主要针对希望扩展 Eclipse 的软件开发人员，因为它允许他们构建与 Eclipse 环境无缝集成的工具。由于 Eclipse 中的每样东西都是插件，对于给 Eclipse 提供插件，以及给用户提供一个一致和统一的集成开发环境而言，所有工具开发人员都具有同等的发挥场所。

这种平等和一致性并不仅限于 Java 开发工具。尽管 Eclipse 是使用 Java 语言开发的，但它的用途并不限于 Java 语言；例如，支持诸如 C/C++、COBOL 和 Eiffel 等编程语言的插件已经可用，或预计会推出。Eclipse 框架还可用来作为与软件开发无关的其他应用程序类型的基础，比如内容管理系统。

基于 Eclipse 的应用程序的突出例子是 IBM 的 WebSphere Studio Workbench，它构成了 IBM Java

开发工具系列的基础。例如，WebSphere Studio Application Developer 添加了对 JSP、Servlet、EJB、XML、Web 服务和数据库访问的支持。

尽管大多数开发人员不会使用 Eclipse 来开发插件，或创建基于 Eclipse 的新产品，但是 Eclipse 的开放源代码性质所意味的，并不只是它使得 Eclipse 免费可用（尽管便于商业化的许可证意味着插件可能要花钱）。开放源代码鼓励创新，并激励开发人员（甚至是商业开发人员）为公共开放源代码库贡献代码。为这个项目作贡献的开发人员越多，这个项目就会变得对每个人越宝贵。随着这个项目变得更加有用，更多的开发人员将会使用它，并围绕它形成一个社区，就像那些围绕 Apache 和 Linux 形成的社区一样。

Eclipse.org 协会管理和指导 Eclipse 正在进行中的开发。据说 IBM 花了 4000 万美元开发 Eclipse，并把它作为一个开放源代码项目发布。之后，Eclipse.org 协会吸收了许多软件工具提供商，包括 Borland, Merant, Rational, RedHat, SuSE, TogetherSoft 和 QNX。从那以后还有其他公司相继加入，包括 Hewlett Packard, Fujitsu, Sybase。

1.2 Eclipse 大事记

如图 1-1 所示（摘自 Eclipse 官方网站），自从 2001 年发布第一个版本开始，Eclipse 逐渐地被越来越多的开发人员所采纳，其功能和需求也在不断地更新和变化中。

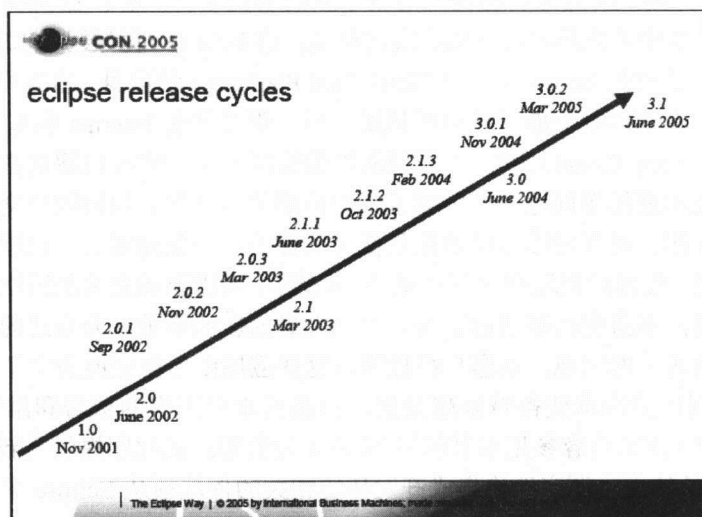


图 1-1 Eclipse 发展历程

1.0 版本的目的纯粹就是作为一个 Java 集成开发平台，就如 JBuilder 和 VisualAge 那样。

在 1.0 版本的应用过程中，由于 Eclipse 的开源特性，Eclipse 社区不断地从其广大的 Fans 那里得到新的灵感，特别是一些面向最终用户的技术支持人员，他们往往会面对客户这些提问：

- 为何你所提供的产品不与其他公司提供的工具集成？
- 为何不能把某个工具产生的数据导入到其他工具中去？
- 为何在不同程序之间进行导入和导出时遇到了问题？
- 为何程序在执行相似的任务时却有着完全不同的用户界面？
- 为何不将 Web 站点设计工具与脚本编制程序集成？

为了解决以上的用户需求，Eclipse 被重新设计和定位，并于 2002 年推出了 2.x 版本。Eclipse 转变了自身的角色，从一个单一的集成开发环境转变为一个开放的可扩展的集成平台。它能将单独开发



的工具融合到精心设计的套件中；它可以很容易地将现有工具移植到平台中；它是开放式的，让人容易理解，并且功能强大，不需要额外的努力就可以支持集成；它提供工具从而有助于使常见的任务自动化；它足够稳定，可以在它上面构建业界领先的工具。

2.x 版本的 Eclipse 平台可以达成以下目标：

- 支持用于应用开发的各种工具的构建。
- 支持非受限的工具提供者，包括独立的软件提供商。
- 支持用于操作任意类型的文件（HTML，Java，C，JSP，EJB，XML，GIF 等）的工具。
- 推动各种工具的无缝集成。
- 支持 GUI（图形用户界面）和非 GUI 的应用开发环境。
- 运行于多种操作平台（Windows，Linux 和 Solaris）。
- 利用 Java 编程语言的普及来推动应用工具的开发。

2.x 版本在应用过程中的确达到了其所提出的目标，但是新的问题随之而来，举个例子来说，基于 Eclipse 我们开发出了一套工具，但是如何提交给客户呢？要将整个 Eclipse 集成开发环境都给用户打包过去吗？那太荒唐了，客户可不需要包含一个集成开发环境的产品。另外，许多插件并不是集成开发环境所专有的，它们完全可以脱离 Eclipse 而单独使用，最主要的是：先进的桌面应用有许多相似之处：帮助系统、升级管理、配置管理、开放的架构，等等，Eclipse 集成开发环境的整个架构经历了无数的测试，已经证明是健壮的和一流的，这些东西为什么不能提供给用户用于桌面开发呢？

因此，自 2.1 版本开始，Eclipse 社团内部启动了新的研究项目：基于 Eclipse 的集成开发环境构建技术。主要用来实现非集成开发环境的应用，换句话说，将 Eclipse 可扩展架构进行重用，开发出具备 Eclipse 新特征的应用。这就是 Eclipse RCP（Rich Client Platform）的前身。富客户机程序（Rich Client）并不是一个新的名词，在 20 世纪 90 年代曾经风靡一时，但是随着 Internet 和基于 Web 的应用的不断发展，瘦客户机程序（Thin Client）成为了一项通用的解决技术。它可以解决富客户机程序所带来的诸如管理不便和升级成本高昂等问题。以放弃了用户界面的特色和高速的用户交互为代价，降低了部署和维护企业应用的费用。降低费用和简单化是很受欢迎的，但是向瘦客户机程序迁移在根本上是一种功能和性能上的倒退。瘦客户机程序采用请求-应答模型，所以要求更大的网络容量以确保最佳的交互效果。随着应用和用户本身变得越来越复杂，以及大量的新的需求（分布式的业务逻辑、操作移动设备、非互联的客户端等）的出现，瘦客户机程序对这些应用就显得无能为力了。

因此，富客户机程序的需求变得越来越强烈，但是其本身固有的部署和维护问题怎么解决呢？Eclipse 3.x 版本的 RCP（详细内容参见本书的第 17 章）为富客户机程序提供全新的解决方案，它充分利用 Eclipse 插件化的特点，彻底地将集成开发环境相关的依赖项从 Eclipse 平台底层剥离，同时更多的用户界面组件被开放并允许个性化的定制。采用基于 OSGi（Open Service Gateway Initiative）的平台运行时，从而实现动态的插件安装、移除和升级机制。3.x 版本具有的以下特性解决了富客户机程序所固有的问题。

1. 组件化

Eclipse 包含了一套健壮的组件模型，基于 Eclipse 的系统通过组合这些称之为“插件”的组件来实现自身功能。插件是有版本编号的，可以在多个应用中共享，相同插件的多个版本可以并行安装，通过配置，来运行其所指定的版本，通过添加或者替换组件可以实现应用的不断完善和扩充。

2. 基础设施

组件模型之上是一套框架和工具，用于帮助实现客户端应用的开发，提供可以扩展的用户界面规范，帮助支持、上下文敏感帮助、网络升级、错误控制等。

3. 本地用户体验

相对于瘦客户机程序而言,许多用户希望得到丰富、舒适的本地用户体验。Eclipse 标准小窗口工具箱 (Standard Widget Toolkit) 提供图形化的用户界面, SWT 与本地的操作系统紧密地联系在一起,使得应用程序与操作系统在用户界面上风格统一。

4. 轻便

瘦客户机程序的一个显著特点就是可以在任何地方运行。Eclipse 支持各种异构的操作系统和客户机环境——从服务器到个人计算机、移动设备、嵌入式设备等。只要配备了 Java 虚拟机(JVM)和 J2ME™ 基础库 (或者更多如 J2SE™ 1.4), 你的应用都可以运行。

5. 智能性

在部署和维护富客户机应用的支出中,安装和升级问题占据了花费的大部分。Eclipse 的组件架构使得插件的部署和升级通过各种方式 (HTTP、Java Web Start、集中式的升级站点、简单的文件拷贝或者是企业管理系统) 在一种可控的状态下自动进行。

6. 非互联操作

富客户机程序在一台本地计算机上运行,可以脱离互联网,与瘦客户机程序相比,这是主要的优势。各种应用可以通过本地缓存、副本、存储转发机制等来解决互联中断的问题。

7. 开发工具和组件库支持

Eclipse 提供第一流的 Java 集成开发环境,覆盖产品开发、测试到发布的全过程。Eclipse 提供丰富的组件支持,包括可插入的用户界面、安装和升级支持、文本编辑、图形编辑框架、模型框架等。

通过引入 RCP 的概念, Eclipse 被重新定义为以下描述: Eclipse 是一种支持任何应用的、开放的、可扩展的平台。

1.3 Eclipse 的安装

Eclipse 的当前版本是 3.x 系列,与其他的集成开发环境相似, Eclipse 同样要求适配的 JDK, JDK 是进行 Java 开发的基础,它包含了所有编写、运行 Java 程序需要的工具: Java 基本组件、库、Java 编译器、Java 解析器、小应用程序浏览器等。

本书使用的是 Eclipse 3.0 版本和相应的简体中文语言包, JDK 采用 Sun 公司的 JDK 1.4.2 版本, 在应用过程中证明该组合是稳定的。

1.3.1 JDK 下载安装与配置

1. 下载安装 JDK

(1) 访问下面的网址进入 JDK 1.4 下载页面。 <http://java.sun.com/j2se/1.4.2/download.html>。

(2) 单击“Download J2SE SDK”, 如图 1-2 所示。

(3) 单击“Accept”继续, 如图 1-3 所示。

(4) 如图 1-4 所示, 列出了各个平台下的 JDK 版本, 包括 Windows 版、Linux 版、Solaris 版等。本书中介绍的内容都是在 Windows XP 系统下进行的, 请你下载安装 Windows 版。Windows 版有两种安装方式, 一种是完全下载后再安装, 一种是在线安装。在线安装过程中, 可能会出现网络中断等问题, 影响安装, 建议选择第一种。

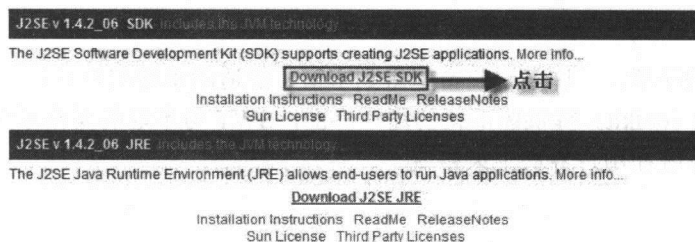


图 1-2 点击下载 JDK

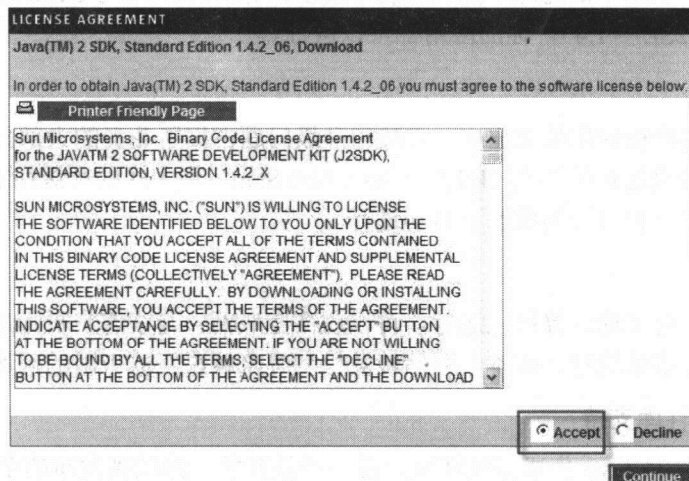


图 1-3 接受注册协议

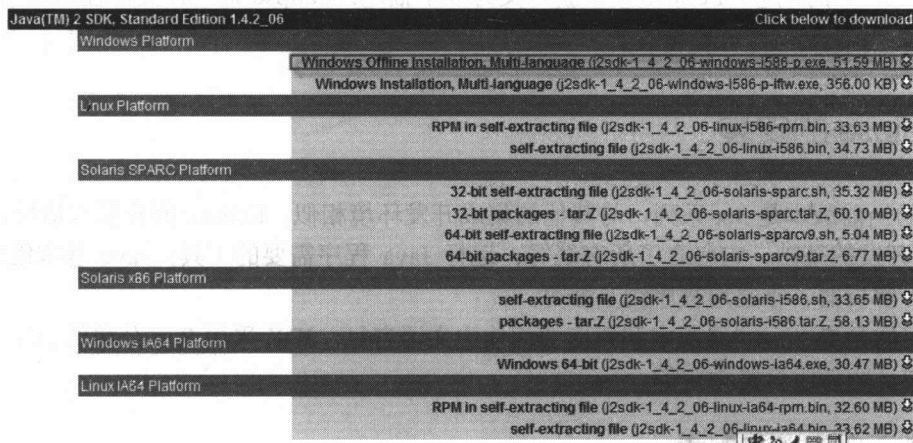


图 1-4 下载 Windows 版 JDK

(5) 下载完成后，双击图标进行安装，安装过程中可以自定义安装目录等信息，例如可以选择安装目录为 D:\jdk1.4。

2. 配置环境变量

安装完 JDK 后，接下来需要配置计算机系统的环境变量，以便 Java 程序能找到已安装的 JDK 和其他配置信息。步骤如下。

(1) 在“我的电脑”上单击鼠标右键，选择“属性”，在弹出的对话框中选择“高级”→“环境变量”，就可以看到环境变量对话框，如图 1-5 所示。