



程序员为程序员编写的图书!



clipse

精通 Eclipse

张大治 应群 编著



清华大学出版社

精通 Eclipse

张大治 应群 编著

清华大学出版社

北 京

内 容 简 介

Eclipse 是当前最流行的基于 Java 的集成开发环境之一,功能非常强大。它是一个开放源代码的、基于 Java 的可扩展开发平台。全书共分 13 章,以 Eclipse 各种实用功能为线索,全面地介绍了 Eclipse 的体系结构特征及其各项功能的使用方法。本书语言通俗易懂、叙述深入浅出、图文并茂、概念结合操作、理论结合实例,方便读者在较短的时间内认识并掌握 Eclipse 平台的使用方法,并能用它作为开发利器用到实际编程工作中去。

本书具有很强的实践性和操作性,内容紧凑、结构严谨,不论对于 Eclipse 的初学者还是已经接触过 Eclipse 的中、高级用户,都有很高的参考价值,同时也是在校学生学习 Java 集成开发环境的理想教材。

版权所有,翻印必究。举报电话:010-62782989 13501256678 13801310933

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

本书防伪标签采用特殊防伪技术,用户可通过在图案表面涂抹清水,图案消失,水干后图案复现;或将表面膜揭下,放在白纸上用彩笔涂抹,图案在白纸上再现的方法识别真伪。

图书在版编目(CIP)数据

精通 Eclipse/张大治 应群 编著. —北京:清华大学出版社,2006.6

ISBN 7-302-12403-5

I. 精… II. ①张…②应… 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 005033 号

出 版 者:清华大学出版社 地 址:北京清华大学学研大厦

<http://www.tup.com.cn> 邮 编:100084

社 总 机:010-62770175 客户服务:010-62776969

组稿编辑:胡辰浩

文稿编辑:袁建华

封面设计:信 京

版式设计:康 博

印 刷 者:北京密云胶印厂

装 订 者:北京市密云县京文制本装订厂

发 行 者:新华书店总店北京发行所

开 本:185×260 印张:28.75 字数:700 千字

版 次:2006 年 6 月第 1 版 2006 年 6 月第 1 次印刷

书 号:ISBN 7-302-12403-5/TP·7946

印 数:1~4000

定 价:39.80 元

前言

Eclipse 作为开源的软件系统,在开源社区的支撑下,不断推出新版本。时至今日,Eclipse 已经成为公认为最为广泛接受的、甚至是最好的开发工具。

本书中各个章节内部耦合紧密,章节之间却比较独立,即使读者抽出若干章节来单独学习,也能顺利进行。因此本书可以作为一本 Eclipse 的参考手册来使用。

与此同时,全书的 13 章体现出 3 个主要层次。其中,第 1 章~3 章是基础知识部分,主要介绍 Eclipse 本身的特点及其开发环境的使用,对 Eclipse 操作比较熟悉的读者可以跳过这 3 章;从第 4 章~9 章属于基于 Eclipse 的项目开发中经常用到的功能;第 10 章~13 章介绍了在 Eclipse 平台上开发 Java 应用时的一些高级特性,比如数据库、高级应用界面、单元测试和重构等。

第 1 章是关于 Eclipse 平台的概述,包括其起源和体系结构。简要讨论 Eclipse 的组成、特点和发展,详细介绍了 Eclipse 平台的安装过程。

第 2 章介绍 Eclipse 平台的体系结构和平台的常用操作。对平台本身的理解将有助于开发的进行。而 Eclipse 工作台的操作虽不复杂,但要做到灵活使用仍需要一定的实践和练习。读者可以在实际操作过程中仔细体会,逐渐熟悉 Eclipse 平台的特性及各种常用工具,从而提高开发效率。

第 3 章涵盖了如何使用 JDT 开发 Java 程序的各个方面,包括 JDT 的基本概念、Java 开发环境的具体使用以及 Java 程序的运行和调试等。

第 4 章是关于 Ant 的介绍以及 Ant 的语法和如何在 Eclipse 中使用 Ant 插件进行构建部署的相关知识。本章还详细讲解如何使用 Ant 插件进行构建部署。

第 5 章从版本控制和协同开发的概念讲起,介绍 CVS 的发展历史和 CVS 的一般特性。随后介绍在 Linux 和 Windows 下 CVS 服务器的安装和配置过程。最后着重介绍如何在 Eclipse 平台上使用 CVS 进行协同开发和版本控制。

第 6 章主要介绍 Eclipse 中 J2EE 开发环境的安装,包括 Tomcat 和 JBoss 的下载、安装以及配置问题,并且介绍如何使用 TomcatPlugin 插件将 Tomcat 环境集成在 Eclipse 中。本章还重点介绍 Eclipse 中 J2EE 的插件 Lombos,并通过一个配置实例介绍如何通过 Lombos 将 JBoss 服务器集成到 Eclipse 中。

第 7 章是关于使用 EclipseUML 插件在 Eclipse 平台上进行 UML 系统建模的介绍,详细叙述了 EclipseUML 插件的相关情况和安装过程。另外本章还通过具体的例子展示利用 EclipseUML 插件进行建模的具体方法。

第 8 章介绍 Struts 框架的概念、MVC 设计模式、Struts 结构和标签以及如何在 Eclipse 中开发 Struts 项目。Struts 是一个为开发基于 MVC 模式的应用架构的开源框架,是利用 Java Servlet 和 JSP 构建 Web 应用的一项非常有用的技术。

第 9 章介绍使用 Eclipse 进行 J2EE 开发调试的方法。主要讲述 JSP 的基本概念以及在

Eclipse 中编写、调试 JSP 的步骤。另外，还介绍 EJB 的基本概念以及在 Eclipse 中进行 EJB 开发的相关设置和开发、调试步骤。通过本章的学习，使读者初步掌握在 Eclipse 中使用 Lomboz 插件进行 J2EE 项目开发、部署和调试的基本方法。

第 10 章主要介绍数据库连接的基本概念、JDBC 及其驱动类型、MySQL 数据库在 Windows 和 Linux 下的安装和配置、MySQL 数据库的基本操作等，然后介绍 SQLExplorer、QuantumDB 等 Eclipse 数据库插件，包括其安装、配置和数据库操作的方法；最后介绍 EJB 的基本概念和在 Eclipse 下使用 Lomboz 插件开发 J2EE 项目、创建 EJB、建立 EJB 与数据库的连接以及 EJB 的部署方法，还介绍 EJB 的编程模型，以加深读者对于 EJB 的认识和理解。

第 11 章主要介绍 Eclipse 平台上应用界面开发的主要技术——SWT 和 JFace。在介绍了 SWT 的基础知识之后，以一个比较详细的例子作为演示。最后还介绍 JFace 的相关内容。通过本章的学习，相信读者能够使用 SWT 和 JFace 快捷高效地开发出精美友好的应用界面。

第 12 章从软件测试入手，首先简要叙述软件测试和单元测试的相关概念和基本思想，然后具体介绍 JUnit 技术，包括其基本概念和核心 API，最后介绍如何在 Eclipse 中使用 JUnit。

第 13 章先简单介绍重构的一些概念和方法，然后重点讲解如何使用 Eclipse 进行重构，并通过具体例子实践一个重构的过程。

最后在附录中提供关于 Eclipse 插件开发的概述，包括插件开发环境的介绍以及如何使用 PDE 开发调试 Eclipse 插件。本部分将通过一个 PDE 自带的小例子来展示插件开发的机理，使读者进一步了解插件开发技术。

全书内容覆盖利用 Eclipse 平台开发应用程序的各个方面，知识面广泛，逻辑和条理清晰有序，过程步骤清晰完善而且易于操作。

除了封面署名的作者之外，参与本书制作的还有秦众森、王芳、杨绳兰、宋倩、陈云建、胡铁石、曾艳、陈文琪、霍枝兰、刁啓太、梁仙明、余汉喜、于凤先、程仪、陈歌和陈文萍等人。在此，编者对以上人员致以诚挚的谢意！

本书在编写过程中参考了很多宝贵的文献，在此，向这些文献的作者表示真心的感谢！

本书内容详实、结构紧凑、条理清晰、覆盖知识点全面。但由于编写时间较为仓促，书中难免会有疏漏和不足之处，恳请广大读者提出宝贵意见。我们的电子邮箱是 huchenhao@263.com。

编 者

目

第 1 章 Eclipse 简介	1
1.1 何谓 Eclipse	1
1.1.1 概述	2
1.1.2 Eclipse 的组成	4
1.1.3 Eclipse 的特点	6
1.1.4 Eclipse 的发展过程	8
1.1.5 Eclipse 所遵循的开源和 CPL	9
1.2 Eclipse 的安装	10
1.2.1 JDK 1.4.2 的安装	10
1.2.2 Eclipse 3.1 的安装	11
1.2.3 检验 JRE 版本和环境变量	13
1.2.4 在 Linux 下安装 Eclipse 平台	13
1.3 Eclipse 开发环境	15
1.3.1 视图	16
1.3.2 观察窗口	17
1.3.3 编辑窗口	18
1.4 Eclipse 3.1 的新功能和改进	19
1.4.1 平台更改	19
1.4.2 SWT	26
1.4.3 小组/CVS	28
1.4.4 ANT	30
1.4.5 调试	32
1.5 本章小结	32
第 2 章 Eclipse 平台和体系结构	35
2.1 Eclipse 体系结构	35
2.1.1 运行时	36
2.1.2 工作空间	36
2.1.3 工作台	37
2.1.4 团队协作开发	37
2.1.5 帮助系统	38
2.2 Eclipse 工作台	38
2.2.1 概述	39

录

2.2.2 菜单和工具栏	44
2.2.3 创建第一个项目	45
2.2.4 资源管理	46
2.2.5 搜索及搜索视图	50
2.2.6 任务及任务视图	51
2.2.7 书签及书签视图	52
2.2.8 定制工作台	53
2.3 小结	57
第 3 章 使用 JDT 开发 Java 程序	59
3.1 JDT 基础	59
3.1.1 Java 项目(Java Projects)	59
3.1.2 Java 构建器(Java Builder)	60
3.1.3 Java 透视图(Java Perspectives) 和 Java 视图(Java Views)	60
3.1.4 Java 编辑器(Java Editor)	63
3.1.5 快速修正(Quick Fix)	64
3.1.6 快速协助(Quick Assist)	64
3.1.7 模板(Templates)	65
3.1.8 Java 搜索(Java Search)	66
3.1.9 重构支持 (Refactoring support)	67
3.1.10 构建类路径 (Build Classpath)	67
3.1.11 类路径变量 (Classpath Variables)	67
3.1.12 排斥和包含过滤器(Exclusion and Inclusion Filters)	67
3.1.13 访问规则(Access Rules)	68
3.1.14 调试器(Debugger)	68
3.1.15 代码片断编辑测试窗 (Scrapbook)	68
3.1.16 本地调试(Local Debugging)	68

3.1.17 远程调试 (Remote Debugging)	69
3.1.18 断点(Breakpoints)	69
3.1.19 字符串外部化 (String Externalization)	69
3.2 Java 开发环境的使用	70
3.2.1 创建 Java 元素	70
3.2.2 使用层次结构视图	74
3.2.3 使用 Java 包视图	75
3.2.4 使用 Java 编辑器	77
3.2.5 外部化字符串	85
3.2.6 Java 搜索	87
3.2.7 构建 Java 程序	90
3.3 运行与调试	91
3.3.1 使用构建路径	91
3.3.2 启动 Java 程序	92
3.3.3 启动 Java Applet	93
3.3.4 创建 Java 应用程序配置	94
3.3.5 重新启动程序	96
3.3.6 本地调试	96
3.3.7 远程调试	103
3.4 小结	104
第 4 章 Eclipse 与 Ant 插件	105
4.1 Ant 的介绍	105
4.1.1 Ant 的优点	106
4.1.2 Ant 的安装和配置	106
4.1.3 Ant 支持库	109
4.1.4 Ant 的运行参数介绍	110
4.1.5 Ant 的使用	111
4.1.6 Ant 名词附录	117
4.2 Ant 语法	119
4.3 使用 Ant 插件进行构建部署	126
4.3.1 Ant 构建文件的创建	126
4.3.2 Ant 构建文件的编辑	127
4.3.3 Ant 构建文件的运行	129
4.3.4 定义自己的 Ant 选项	131

4.3.5 打开 Ant 视图	133
4.3.6 使用 build.xml 构建 部署文件	134
4.4 小结	136

第 5 章 Eclipse 下使用 CVS 进行

协同开发	137
5.1 版本控制协同开发和 CVS	137
5.1.1 版本控制和协同开发	137
5.1.2 CVS 的由来	138
5.1.3 CVS 的几个基本概念	139
5.1.4 利用 CVS 软件开发的 典型过程	140
5.1.5 常用的 CVS 命令	141
5.2 Linux 下 CVS 服务器配置	144
5.2.1 CVS 服务确认	144
5.2.2 Linux 下 CVS 服务器安装	144
5.2.3 Linux 下 CVS 服务器配置	145
5.3 Windows 下 CVS 服务器配置	147
5.3.1 CVSNT 的安装	147
5.3.2 CVSNT 的配置	148
5.4 Eclipse 利用 CVS 进行 协同开发	149
5.4.1 Eclipse 与 CVS 服务器 的连接	149
5.4.2 用 CVS 共享 Eclipse 工程	152
5.4.3 在 Eclipse 下检出代码	155
5.4.4 代码的提交	155
5.4.5 代码的更新	157
5.4.6 代码的比较	157
5.4.7 从历史记录中恢复版本	161
5.5 小结	164

第 6 章 Eclipse 与 J2EE 开发环境

6.1 Tomcat 与 JBoss	165
6.1.1 Tomcat 简介	165
6.1.2 Tomcat 的安装	166
6.1.3 Tomcat 的运行	168

6.1.4 插件 tomcatPlugin	171	第 9 章 Eclipse 进行 J2EE 开发调试	243
6.1.5 JBoss 简介	174	9.1 创建 J2EE 项目	243
6.1.6 JBoss 的安装及运行	175	9.1.1 J2EE 基本概念	243
6.2 Lomboz 插件介绍	177	9.1.2 创建 J2EE 项目	246
6.2.1 Lomboz 概述	177	9.2 编写和调试 JSP	248
6.2.2 Lomboz 安装及配置	178	9.2.1 配置 JSP 开发调试环境	249
6.3 小结	192	9.2.2 编写 JSP	250
第 7 章 使用 EclipseUML 插件		9.2.3 调试 JSP	252
进行建模	193	9.2.4 编写 JSTL	255
7.1 EclipseUML 插件	193	9.3 编写和调试 EJB	258
7.2 使用 EclipseUML 插件进行		9.3.1 开发 EJB	260
系统建模	195	9.3.2 调试 EJB	269
7.2.1 准备工作	196	9.4 小结	270
7.2.2 绘制 UML 图	198	第 10 章 Eclipse 数据库开发	271
7.3 小结	204	10.1 数据库连接	271
第 8 章 Eclipse 下的 Struts 框架应用	205	10.1.1 数据库管理系统	271
8.1 MVC 与 Struts	205	10.1.2 JDBC	272
8.1.1 何谓 MVC 模式	206	10.1.3 MySQL 的安装与配置	273
8.1.2 将 MVC 概念映射到		10.1.4 MySQL 编程初步	278
Struts 组件中	206	10.2 Eclipse 数据库插件	282
8.2 Struts 结构与框架	208	10.2.1 SQLExplorer 插件	282
8.2.1 Struts 体系结构	208	10.2.2 QuantumDB 及其他插件	288
8.2.2 详细分析 Struts	209	10.3 EJB 与数据库的连接	293
8.2.3 从 Struts 的组件来看		10.3.1 EJB 概念回顾	293
Struts 的工作原理	213	10.3.2 EJB 与数据库的连接	294
8.2.4 何谓自定义的 tag	214	10.4 小结	301
8.2.5 Struts 标记库	215	第 11 章 Eclipse 下的应用界面开发	303
8.3 开发 Struts 项目	229	11.1 SWT 简介	303
8.3.1 用 Eclipse 编写一个简单的		11.1.1 SWT 概述	304
Struts 程序	229	11.1.2 SWT 的基本内容	305
8.3.2 myeclipse 插件简介	236	11.2 SWT 高级主题	308
8.3.3 使用 myeclipse 开发		11.2.1 布局	308
Struts 项目	236	11.2.2 线程问题	309
8.4 小结	242	11.2.3 错误处理	312
		11.2.4 图形	312
		11.3 使用 SWT 开发应用界面	315

11.3.1 示例概述	315	13.2.8 更改方法特征符	387
11.3.2 开发过程	316	13.2.9 提取局部变量	388
11.4 JFace 用户界面框架	329	13.2.10 提取常量	388
11.4.1 查看器	330	13.2.11 自包括值域	389
11.4.2 操作和添加项	334	13.2.12 将成员上拉至父类	389
11.4.3 用户界面资源	335	13.2.13 将成员下推至子类	391
11.4.4 长时间运行的操作	338	13.2.14 在各个类型之间移动 静态成员	391
11.5 小结	339	13.2.15 将实例方法移至组件	392
第 12 章 Eclipse 和 JUnit 测试技术	341	13.2.16 将局部变量转换为值域	392
12.1 单元测试与 JUnit	341	13.2.17 从类型中提取接口	393
12.1.1 软件测试概述	342	13.2.18 将表达式替换为 方法参数	394
12.1.2 单元测试概述	343	13.2.19 将构造函数调用替换 为工厂方法调用	394
12.1.3 JUnit 概述	345	13.2.20 撤销/重做重构操作	395
12.2 JUnit API	351	13.3 使用 Eclipse 进行重构	395
12.2.1 JUnit 核心概述	351	13.3.1 源程序介绍	395
12.2.2 JUnit 核心类简述	356	13.3.2 提取 switch 语句	398
12.3 在 Eclipse 中使用 JUnit	357	13.3.3 移动函数 amountFor()	402
12.3.1 测试的准备	358	13.3.4 提取会员积分计算的 代码	405
12.3.2 测试的建立	361	13.3.5 移动 getCharge() 函数	406
12.3.3 测试的运行	364	13.3.6 重构后的代码	409
12.3.4 使用 TestSuite 进行测试	365	13.4 小结	412
12.4 小结	368	附录 Eclipse 插件开发技术	413
第 13 章 Eclipse 重构技术	369	附录 A 插件开发环境 PDE	413
13.1 重构	369	A.1 Eclipse 的构架	413
13.1.1 重构的时机	370	A.2 PDE 简介	414
13.1.2 重构的优点	371	A.3 PDE 中的概念	416
13.1.3 重构的方法	373	附录 B 利用 PDE 开发插件	418
13.1.4 代码坏味道	378	B.1 Eclipse 插件的结构和 声明文件	418
13.2 Eclipse 中的重构支持	382	B.2 插件的部署和启动	419
13.2.1 提取方法	382	B.3 介绍一个 Eclipse 自带的 插件例子程序	420
13.2.2 重命名包	383		
13.2.3 重命名编译单元	383		
13.2.4 重命名类型	384		
13.2.5 重命名方法	385		
13.2.6 重命名值域	385		
13.2.7 重命名局部变量	386		

B.4 自己开发一个简单的 视图插件	424	C.4 发布插件	439
B.5 为 B3 中的插件编写片段	430	附录 D Eclipse 3.1 插件迁移指南 ..	442
附录 C 调试运行插件	435	D.1 Eclipse 3.0 与 Eclipse 3.1 之间的不兼容性	443
C.1 对运行插件的选择	435	D.2 采用 3.1 机制和 API 时 所需进行的更改	448
C.2 运行时调试跟踪	437		
C.3 采用调试方式运行插件	439		

第1章

Eclipse 简介

Eclipse 是一个开放源代码的、基于 Java 的可扩展开发平台，它的目标是简化跨平台软件的开发过程。

本章是关于 Eclipse 平台的概述，包括 Eclipse 的起源和体系结构。本章首先简单介绍 Eclipse 的组成、特点和发展，接着详细介绍 Eclipse 平台的安装过程，然后通过一个简单的例子来展示 Eclipse 的开发环境，最后集中介绍 Eclipse 3.1 的新功能和改进。

本章要点：

- Eclipse 的组成、特点和发展
- Eclipse 平台的安装
- Eclipse 开发环境
- Eclipse 3.1 的新功能和改进

1.1 何谓 Eclipse

Eclipse 是当前最流行的集成开发环境之一。它是一个开放源代码的、基于 Java 的可扩展开发平台。如图 1-1 所示的是 Eclipse 3.1 的启动界面。

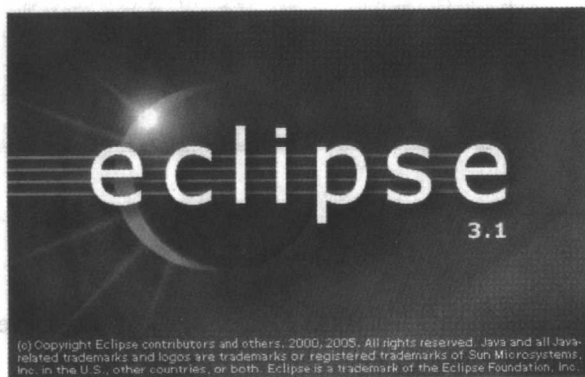


图 1-1 Eclipse 3.1 的启动界面

1.1.1 概述

过去, Java 程序员遇到的最大难题是当使用 javac 编译程序时所出现的整屏的错误信息, 这些错误信息可能只是因为类中缺少了一条 import 语句, 或者少了一个分号, 或者是在修改代码的过程中没有注意区分变量名称的大小写。javac 只是一个编译器, 而不是编辑器——程序员需要的是一个在编写代码的过程中就能检测到这些低级语法错误的自动化工具。

集成开发环境(Integrated Development Environment, IDE)能有效地解决以上所说的问题, 并且已经有多种 Java IDE 正在被程序员使用, 例如 Borland 的 Jbuilder、IBM 的 Visual Age for Java 与 WebSphere Studio、Oracle 的 Jdeveloper、Sun 的 Forte for Java、WebGain 的 Visual Café、TogetherSoft 的 Together, 还有开放源代码的 Eclipse、NetBeans 等, 它们通常包含了编辑器、调试器、项目工具和大量的用户偏好设置。目前最流行的也是功能最强大的一种 Java IDE 就是本书所介绍的 Eclipse。

实际上, Eclipse 被设计成了 IDE 的开发工具。

如图 1-2 所示的是 Eclipse 基本界面。本书所介绍的 Eclipse IDE 为英文版本。

Eclipse 提供了可扩展的、灵活的插件应用程序接口, 开发者可以通过插件组件来构建开发环境。

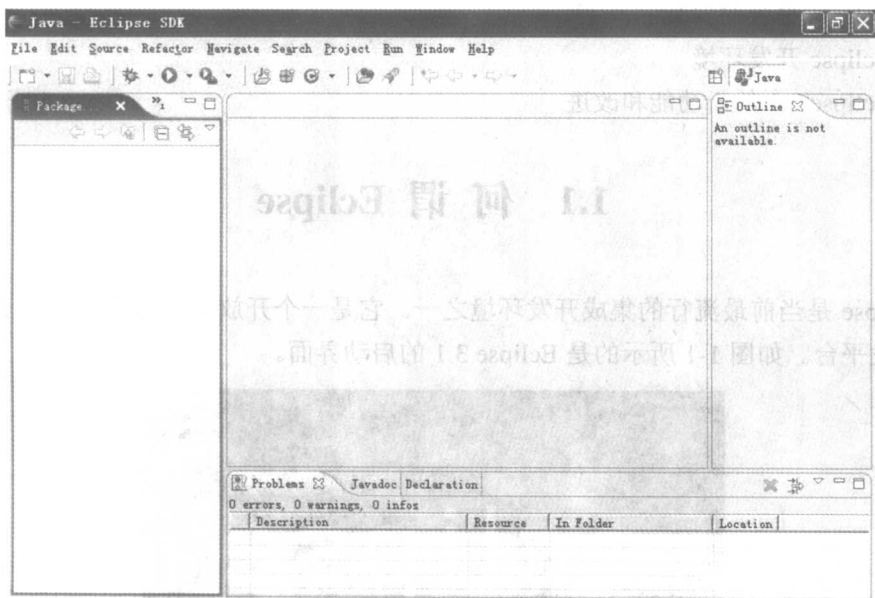


图 1-2 Eclipse 的基本界面

Eclipse 附带有—个标准的插件集, 包括 Java 开发工具(Java Development Tools, JDT)和插件开发环境(Plug-in Development Environment, PDE)。

PDE 主要是针对希望扩展 Eclipse 的软件开发人员而设计的, 因为它允许软件开发人员构建与 Eclipse 环境无缝集成的工具。由于 Eclipse 中的每种元素都是插件, 对于向 Eclipse

提供插件以及向用户提供一致和统一的集成开发环境而言，所有工具的开发人员都具有同等的发挥空间和机会。

目前，Eclipse 的插件已经涉及到建模工具、业务规则管理程序、业务过程设计器、支持 C++ 的嵌入式应用开发环境、HTML 编辑工具和内容管理系统等。

Eclipse 是使用 Java 语言开发的，但它的用途并不仅限于 Java 语言。通过开发插件，它能扩展到任何语言的开发，甚至能成为图形的绘制工具，例如，支持 C/C++、C#、PL1 和 COBOL 等编程语言的插件已经可用。

Eclipse 是跨平台的开发工具，它能在多种操作系统上运行。在网站 www.eclipse.org 上，可以了解到最新的 Eclipse 3.2 M1 版本已经通过如表 1-1 所示的所有平台的测试。

表 1-1 Eclipse 3.2 M1 支持的平台

操作系统	处理器	窗口系统	Java 2 平台
Microsoft Windows XP	Intel x86	Win32	Sun Java 2 Standard Edition 5.0 Update 4 for Microsoft Windows
Microsoft Windows XP	Intel x86	Win32	IBM 32-bit SDK for Windows, Java 2 Technology Edition 5.0
Red Hat Enterprise Linux WS 4	Intel x86	GTK	Sun Java 2 Standard Edition 5.0 Update 4 for Linux x86
Red Hat Enterprise Linux WS 4	Intel x86	GTK	IBM 32-bit SDK for Linux on Intel architecture, Java 2 Technology Edition 5.0
SLES 9	Intel x86	GTK	Sun Java 2 Standard Edition 5.0 Update 4 for Linux x86
SLES 9	Intel x86	GTK	IBM 32-bit SDK for Linux on Intel architecture, Java 2 Technology Edition 5.0
Sun Solaris 10	SPARC	Motif	Sun Java 2 Standard Edition 5.0 Update 2 for Solaris SPARC
HP HP-UX 11i	hp9000 PA-RISC	Motif	HP-UX JDK for the Java 2 Platform Standard Edition for 5.0 01
IBM AIX 5L Version 5.2	PowerPC	Motif	IBM 32-bit SDK for AIX, Java 2 Technology Edition 5.0
Apple Mac OS X 10.4	PowerPC	Carbon	Java 2 Platform Standard Edition (J2SE) 5.0 Release 1 for Tiger

实际上，除了表 1-1 中所列出的平台之外，Eclipse 还支持其他版本的操作系统平台，例如，Win32 系列的 Windows 98、Windows Me、Windows NT、Windows 2000、Windows

XP 和 Windows 2003 Server 等, 而且 Eclipse 2 甚至支持流行的 QNX 嵌入式实时操作系统, 但是 Eclipse 的最新版本 3.2 M1 并未支持 QNX, 这是因为 QNX 尚未支持 3.2 M1 版本所需要的 J2SE 5.0。

1.1.2 Eclipse 的组成

Eclipse 包含 Eclipse 项目、Eclipse 工具项目、Eclipse 技术项目、Eclipse Web 工具平台项目、Eclipse 测试和性能工具平台项目、业务情报和报表工具(BIRT)项目、数据工具平台项目(DTP)和设备软件开发平台(DSDP)等多个项目。每个项目都由项目管理委员会进行检查和管理, 并且都使用 CPL(Common Public License) 1.0 版本许可协议。下面将分别介绍其中的 5 个项目的任务和内容。

(1) Eclipse 项目

Eclipse 项目是一个开放源代码的软件开发项目, 它专注于向高度集成的工具的开发提供一个健壮的、全功能的、具有商业品质的工业平台。它由 3 个子项目组成: Eclipse 平台、Java 开发工具(JDT)和插件开发环境(PDE)。Eclipse 项目的任务是适应并发展 Eclipse 平台和相关联的工具, 以达到工具建造委员会和用户的需求, 以这种方式, Eclipse 作为一个工业平台的设想将很快得到实现。

Eclipse 平台子项目定义了框架和通用服务的集合, 该集合是 Eclipse 作为组件模型、rich client platform (RCP)和全面的工具集成平台的基础。这些服务和框架包括了基本的运行平台、OSGI (Open Service Gateway Initiative)框架实现、一个标准的工作台用户界面模型、移动本地工具集、资源管理的项目模型、针对增量式编译器的自动资源 delta 管理、语言独立的调试功能底层结构以及支持分布式多用户版本资源管理的底层结构等。

JDT 子项目向 Java 程序员提供了支持开发任何 Java 应用的 Java IDE, 包括对 Eclipse 插件开发的支持。它提供了 Java 项目的视图以及 Java 透视图、若干查看器、编辑器、向导、编译器、代码合并以及代码重构工具。JDT 使得 Eclipse 支持对它自身的开发, 而 JDT 插件本身也可以通过其他的工具来扩展。

PDE 子项目提供了若干视图和编辑器来简化开发 Eclipse 插件的过程。通过使用 PDE, 开发人员可以创建插件的 manifest 文件(plugin.xml), 指定插件的运行时刻所需要的插件, 定义扩展点, 基于其他插件的扩展点进行扩展等。PDE 使得插件的集成变得非常容易。

(2) Eclipse 工具项目

Eclipse 工具项目的主要作用是促进 Eclipse 平台上的各种工具的创建, 为不同的工具创建者提供一个焦点, 以保证为 Eclipse 平台创建出最好的工具。它规定了工具开发的重点和目标, 起到了协调和管理的作用, 防止生成相同的插件, 降低了开发资源的重复和浪费, 并在不同的工具之间定义了无缝的可操作性。

该项目也包含了一系列的子项目, 例如 CDT(C/C++ Development Tools)子项目、GEF(Graphical Editor Framework)子项目、EMF 子项目、COBOL 子项目、VE(Visual Editor)子项目和 UML2 子项目等。

CDT 子项目的目标是提供一个功能完备的基于 Eclipse 平台的 C/C++ 集成开发环境。目前 CDT 项目主要运行在 Linux 平台上, 而且发布的目标平台也是 Linux 平台。

GEF 子项目使开发者能够利用现有的应用程序模型, 并且方便地创建出一个强大的图形编辑器。GEF 允许开发者将任何现有的模型快速映射到图形编辑环境, 该图形环境是基于 SWT 的绘制插件——Draw2D。开发者可以使用 GEF 提供的常用操作接口, 并且可以为特定的应用领域扩展这些接口。GEF 适用于创建多种应用程序, 包括: 流程绘制工具、GUI 绘制工具、UML 图编辑器, 甚至是所见即所得的文本编辑器, 如 HTML 编辑器。

EMF 子项目是一个 Java/XML 框架, 其目的是基于简单的类模型来生成工具和其他应用程序。EMF 帮助开发者将模型快速转换成高效的、正确的、且易定制的 Java 代码。它利用了形式化模型所带来的益处, 但是却不需要开发者对形式化方法了解得很深入。除了代码生成之外, 它还能够将对象保存为 XML 文档, 用于在不同工具和应用程序之间交换。模型可以使用有注解的 Java、XML 文档或者建模工具(如 Rational Rose)来建立, 然后将它导入到 EMF 中。代码生成器将模型转换成一系列的 Java 类。这些类是可扩展的, 程序员可以加入自定义的方法和变量。当模型改变的时候, 可以重新生成这些类, 而用户修改的部分将会被保留下来。反之, Java 代码的修改也将引起对模型的更新。

COBOL IDE 子项目的目标是基于 Eclipse 平台建立一个功能完备的集成 COBOL 的 IDE。目前该项目的开发和发布主要是在 Windows/Solaris/Linux 平台上进行。

VE 子项目是一个框架, 它的目标是为 Eclipse 建立一个 GUI 项目编译连接工具。它包含了 Swing/JFC 和 SWT GUI 编译连接工具的实现方式, 并扩展到其他语言例如 C/C++, 包括 Java 不支持的特性。

UML2 子项目是为 Eclipse 平台开发的 UML 2.0 元模型的基于 EMF 的实现。该项目的目标是提供元模型的一个可用的实现, 用于支持模型工具的开发, 一个通用的 XMI schema 用于实现语义模型间的交换, 作为验证规格说明的方法的测试用例, 以及作为定义和强制兼容性水平的方法的验证规则。

(3) Eclipse 技术项目

Eclipse 技术项目的任务是为开放源代码开发者、研究者、学院和教育者提供新的管道, 以参与将来 Eclipse 的演化。它按照 3 个相关的项目流来组织, 即研究、培育和教育。研究项目在 Eclipse 相关领域诸如编程语言, 工具和开发环境方面进行探索和研究。培育项目是小型的、未正式结构化的项目, 它为 Eclipse 软件基础添加了新的能力。教育项目聚焦于教育材料的开发、教学帮助和课件。

Eclipse 技术项目包含大量的子项目, 例如应用生命周期管理、AspectJ 开发工具(AJDT)子项目、AspectJ、CME (Concern Manipulation Environment)子项目、ECESIS(Eclipse Community Education Project)子项目、ECF(Eclipse Communications Framework)子项目、Equinox 子项目、ERCP(Embedded Rich Client Platform)子项目、图形化模型框架子项目、Lepido、Mylar、Phoenix、Pollinate、并行工具平台和语音工具平台子项目等。有关这些子项目的详细资料可以参阅 <http://eclipse.org/technology/index.html>。

(4) Eclipse Web 工具平台项目

Eclipse Web 工具平台项目的目标是建立一个通用的、可扩展的和基于标准的工具平台, 软件供应商可以基于该工具平台建立定制的 J2EE 和关于 Web 的应用。它包含了以下工具: HTML、CSS、JavaScript、JSP、SQL、XML、DTD、XSD 和 WSDL 代码编辑器; J2EE 项目管理、编译连接工具、模型和 J2EE 浏览器; Web 服务向导和浏览器以及 WS-I 测试工具; 以及数据库访问与查询工具和模型。

Eclipse Web 工具平台项目包括 3 个子项目: Web 标准工具(WST)、J2EE 标准工具(JST)和 JavaServer Faces 工具(JSF)。

WST 子项目的目标是向所有基于 Eclipse 开发并允许网络访问的应用程序的开发环境提供通用的底层结构。其内部设计包括 3 层结构(表现层、事务层与数据逻辑层)开发工具以及与特定系统功能对应的系统表现形式。外部则是针对特定语言或者网络结构的技术, 这是其他子项目的工作, 例如 J2EE 网络工具子项目。WST 所提供的工具包括适用于利用各种标准语言(例如 HTML/XHTML、Web Services、XQueries、SQL 等)开发出来的产品的编辑器、验证器和文档生成器。用于支持的内部结构可能包含一个能在目标服务器环境之间发布、运行、启动、停止网络程序代码功能的特制的工作平台。

JST 子项目最初的设想是要提供一个在适用 J2EE 的应用服务器上开发基于 J2EE 1.4 的程序的基本 Eclipse 插件, 以及一个一般性的 J2EE 加工结构, 以便其他基于 Eclipse 的开发产品使用。其内部设计是一个在适用 JCP 的服务器上提供开发、配置、测试、调试 J2EE 应用程序的平台, 以及为至少一个适用 JSR-88 的 J2EE 服务器提供某个插件的可重现执行。同时还包括一系列使用 EJB、Servlet、JSP、JCA、JDBC、JTA、JMS、JMX、JNDI、Web Services 等 J2EE API 来简化开发过程的工具。这一结构的设计考虑了高级开发过程中的扩展性, 在标准 J2EE 的水平之上提供设计思路与技术提取的构造上的分离。

关于这些子项目的更详细的介绍参阅<http://eclipse.org/webtools/index.html>。

(5) Eclipse 测试和性能工具平台项目

该项目的目标是建立一个通用的、可扩展的、基于标准的工具平台, 软件开发者可以基于该工具平台建立定制的测试和性能评估工具。该项目将会通过测试和性能工具来扩展 Eclipse 平台对软件开发周期和软件的测试、跟踪、调协、日志、检测、分析、自治和管理过程的支持, 并支持多种类型的平台。

1.1.3 Eclipse 的特点

Eclipse 得以流行的原因正是因为它具备有多种能够深深吸引程序员的优点, 以下列举主要的几项优点。

(1) 强大的 IDE

尽管 Eclipse 的目标是通用性, 但是它最广泛的应用仍然是 Java IDE。在这套强大的集成开发环境中, 除了具备常用的编辑、编译、调试等基本功能之外, 还具备了开发者精心为程序员打造的项目管理、重构和 UML 建模等扩展功能。

在版本控制方面,为了支持团队开发,Eclipse 提供了对并发版本控制系统(Concurrent Versions System, CVS)的内置支持,还提供了强大的个人版本管理机制,使得复杂的编程过程更为安全。

(2) 灵活可扩展的框架

在 Eclipse 中实现的绝大部分功能是由相应的插件完成的,例如 WrokBench UI 插件完成界面的外观显示,Resource Management 插件完成维护项目、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,用于开发 Java 程序的开发环境(Java development tooling, JDT)等。整个 Eclipse 体系结构可以不断地向上添加插件,同时,在现有的插件上还可以再添加插件。

Eclipse 框架的灵活性来源于扩展点(extension points),它们是在 XML 文件中所定义的已知接口,并充当插件的耦合点。扩展点的范围包括从用在常规表述过滤器中的简单字符串到一个 Java 类的描述。每一个插件都是在现有的扩展点上开发,并可能还保留有自己的扩展点,以便在这个插件上继续开发。任何 Eclipse 插件定义的扩展点都能够被其他的插件使用,反之,任何 Eclipse 插件也可以遵从其他插件定义的扩展点。插件不知道它们通过扩展点提供的服务将被如何使用。

因此,Eclipse 可以集成不同软件开发工具供应商的产品,任何开发工具厂商都能将他们的开发工具或者组件加入到 Eclipse 平台,用户可以通过相同的接口使用不同的工具。这样,实际上 Eclipse 为用户提供了一整套的开发工具。建模、开发、测试和调试等工作都能在统一的平台上完成,避免了多种工具的不兼容等问题。

(3) 图形界面 API

Java 语言默认的图形界面开发包 AWT 和 SWING 的速度和外观都不够令人满意,Eclipse 的 SWT 开发包提供了一个更为高效和美观的图形开发包。

(4) 开放源代码

Eclipse 是一个开放源代码的软件,是以 Common Public License (简称 CPL)形式发布的,因此它是免费的开发工具。虽然 Eclipse 是一个开放源代码的项目,但是它由 IBM 这样一个商业主体领导着项目的开发,这一点与普通的开源软件有所区别。

公开源代码的结果是 Eclipse 开发者一直在听取用户的意见,实际上 Eclipse 开发者本身就是 Eclipse 最忠实的使用者。它能接受由 Java 开发者自己编写的开放源代码插件,这类似于微软公司的 Visual Studio 和 Sun 微系统公司的 NetBeans 平台。Eclipse 为工具开发商提供了更好的灵活性,使他们能更好地控制自己的软件技术。

(5) 跨平台

虽然昂贵的商业开发工具往往也提供了多种操作系统上的不同版本,但是免费的 Eclipse 在跨平台的特性上毫不逊色。这主要归功于 Java 语言的跨平台特性,利用 Java 开发的 Eclipse 平台可以在不同操作系统和不同处理器上运行。

Eclipse 支持几乎所有支持 JRE 的平台,包括主流的 Windows 和 Linux 系列等。它支持的平台详细列表如表 1-1 所示。