



崔群法 王咏梅 李有军 编著

Eclipse 3.0

从入门到精通



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

TP311.56/408

2008

Eclipse 3.0从入门到精通

崔群法 王咏梅 李有军 编著

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

Eclipse最初由OTI和IBM两家公司的IDE产品开发组创建。在目前所有的可视化开发工具中，Eclipse可以说是最有发展前途的产品之一。本书全面介绍了Eclipse的开发调试环境、SWT、JFace、RCP、Eclipse表单、Web开发和插件开发等。

本书内容结构清晰、由浅入深、注重实用，非常适合Eclipse开发人员使用，尤其适合有一定Java基础的SWT/JFace开发人员、Eclipse插件开发人员、RCP开发人员和Web开发人员使用，是初学者入门的经典图书，是高级程序员的必备参考工具。

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 3.0从入门到精通/崔群法，王咏梅，李有军编著. —北京：电子工业出版社，2008.5

ISBN 978-7-121-06023-6

I. E… II. ①崔… ②王… ③李… III. 软件工具—程序设计 IV. TP311.56

中国版本图书馆CIP数据核字（2008）第018899号

责任编辑：李红玉

印 刷：北京天竺颖华印刷厂

装 订：三河市金马印装有限公司

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

北京市海淀区翠微东里甲2号 邮编：100036

开 本：787×1092 1/16 印张：32.125 字数：820千字

印 次：2008年5月第1次印刷

定 价：58.00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zlts@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

Eclipse最初由OTI和IBM两家公司的IDE产品开发组创建,起始于1999年4月。IBM提供了最初的Eclipse代码基础,包括Platform、JDT和PDE。目前由IBM牵头,多家软件公司围绕着Eclipse项目已经组成了一个庞大的Eclipse联盟,共同参与到Eclipse项目中,其中包括Borland、Rational Software、Red Hat及Sybase,最近Oracle也计划加入到Eclipse联盟中。Eclipse是一个开放源代码的、插件式的、基于Java的可扩展开发平台,已经被应用到Java的各类编程中。

Eclipse是替代IBM Visual Age for Java (IVJ)的下一代IDE(集成开发环境),但它的目标不仅仅是成为专门开发Java程序的IDE,根据Eclipse的体系结构,通过开发插件Eclipse还能扩展到任何语言的开发,甚至能成为图片绘制的工具。目前,Eclipse已经开始提供C语言开发的功能插件。

Eclipse是一个开放源代码的项目,任何人都可以下载Eclipse的源代码,并且在此基础上开发自己的功能插件;同时,可以通过开发新的插件扩展现有插件的功能,例如在现有的Java开发环境中加入Web服务器插件,就可以编写相应的Web程序。Eclipse可以无限扩展,而且有着统一的外观、操作和系统资源管理,这也正是Eclipse的潜力所在。

本书通过大量的实例全面而深入地阐述了Eclipse开发工具的方方面面。

全书共分18章,前16章为理论章节,最后两章为Eclipse项目开发的实践章节。具体内容如下所示:

第1章, Eclipse简介,介绍了Eclipse开发工具的形成和特点,并重点介绍了Eclipse插件的安装方法。

第2章, Eclipse开发环境概述,介绍了Eclipse开发环境的组成,以及相应的功能,是初学者学习Eclipse工具必读的章节。

第3章, 调试代码,以Java代码为例,介绍在Eclipse开发工具中,如何调试编写的程序代码。

第4章, Eclipse敏捷开发,介绍了单元测试及敏捷开发的概念及流程,使用简单的实例来讲述单元测试的重要性和必要性,然后介绍Eclipse怎样对敏捷开发提供支持。

第5章, CVS版本管理,详细介绍CVS的基本概念和cvsnt软件的安装、配置、使用,以及Eclipse平台如何支持软件项目中的源代码版本控制。

第6章, SWT基础,介绍了SWT与AWT/Swing的区别、如何构造SWT运行环境、SWT程序的结构,以及SWT的常用控件和SWT使用的布局管理器,然后运用所介绍的布局管理和控件开发了一个计算器的界面。

第7章, SWT事件,介绍了SWT程序事件,首先介绍事件运行机制和监听器接口,并重点介绍比较常用的事件,如键盘、鼠标和焦点事件等,并在最后以实例的方式介绍一个SWT图形用户界面程序。

第8章, SWT高级应用,介绍了SWT应用的高级部分,如SWT常见的高级组件,并重点介绍比较常用的菜单、对话框和表格等,又介绍了SWT程序中绘制图形和打印的相关信息。

在最后，以实例的形式给出了一个使用SWT编写的记事本。

第9章，JFace基础应用，介绍了Eclipse的另一套图形开发工具——JFace。JFace API是基于SWT API的，它为程序员提供了一些更高层次上的组件，比如视图、操作、对话框和向导。

第10章，JFace高级应用，介绍了MVC设置的表格、树和列表控件，JFace对系统字体、图片和颜色资源的管理，以及JFace的文本处理框架，并在最后应用JFace的文本处理框架创建了一个简单的JSP编辑器。

第11章，RCP技术，介绍了RCP技术的基本概念，RCP开发的基本流程，并在本章后面部分介绍如何开发的实例。

第12章，Eclipse表单，介绍了Eclipse表单，如Eclipse表单中常见的组件，并重点介绍比较常用的表单、滚动表单和超链接等，还介绍了Eclipse表单所具有的布局管理器。在最后，给出了一个使用SWT编写Eclipse表单的实例。

第13章，Web开发，首先介绍Web服务器，然后介绍如何安装myEclipse插件，最后介绍如何在Eclipse中开发Web应用程序。

第14章，Eclipse中开发Struts，介绍了Struts产生的背景和发展、工作原理、体系结构和在Eclipse中使用Struts框架，并给出一个使用MyEclipse开发Struts的实例。

第15章，Eclipse插件开发，介绍了Eclipse的插件体系、插件开发的基本思想和流程，还介绍了如何在Eclipse中创建属于自己的插件。

第16章，常用插件，介绍了一些常用的、下载量非常大的Eclipse插件的下载、安装以及如何在项目开发中使用这些插件。

第17章，公司人事管理系统，介绍了在Eclipse中开发图形用户界面程序的过程。从该系统的需求分析入手，讨论系统设计、数据库设计、系统编码和测试等过程。

第18章，Web图书管理系统，介绍了在Eclipse中开发Web程序的过程。本章从图书管理系统的需求分析入手，逐步分析图书管理系统的需求，围绕系统的需求来建立数据库，并按三层架构来设计与实现图书管理系统。

通过阅读本书，读者将全面了解近年来最热门的开发工具Eclipse的使用。本书适合有一定Java基础的读者阅读，可作为在校学生、中高级技术开发工程师和其他IT技术人员的参考书，也可作为大专院校和培训机构的教学用书。

参加本书编写与制作的人员除封面署名作者以外，还有祝红涛、吴东衡、龙建华、刘海松、陈红旭、朱璟煜、王伟平、张水波、陈军红、王俊伟、唐有明、李振等。在此，对他们表示衷心的感谢。由于编写时间仓促，加之作者水平有限，书中难免会有错误和疏漏之处，恳请广大读者给予批评和指正。

为了方便读者阅读，本书配套资料请登录“华信教育资源网” (<http://www.hxedu.com.cn>)，在“资源下载”频道的“图书资源”栏目下载。

目 录

第1章 Eclipse简介	1	3.2.4 表达式视图	35
1.1 Eclipse的历史	1	3.2.5 代码片段编辑测试窗口	35
1.2 Eclipse体系结构	2	3.3 热替换	39
1.3 安装Eclipse	4	3.4 本地调试	39
1.3.1 安装JDK	4	3.5 远程调试	42
1.3.2 Eclipse安装	5	3.6 调试使用技巧	44
1.4 Eclipse插件安装	6		
1.4.1 复制安装	6	第4章 Eclipse敏捷开发	45
1.4.2 链接安装	7	4.1 单元测试	45
1.4.3 查看插件	7	4.1.1 安装JUnit	46
1.5 Eclipse示例	8	4.1.2 编写测试用例	46
		4.1.3 编写Java类	51
		4.1.4 运行测试用例	53
第2章 Eclipse开发环境概述	11	4.2 重构	54
2.1 Eclipse界面一览	11	4.2.1 重构简述	55
2.1.1 透视图操作	12	4.2.2 重命名	58
2.1.2 编辑器操作	13	4.2.3 提取方法	59
2.2 文件结构	15	4.2.4 移动	63
2.2.1 创建工程文件及类文件	15	4.2.5 下推	64
2.2.2 导入项目使用的包	18		
2.2.3 设置编译方式及运行程序	20	第5章 CVS版本管理	66
2.3 常用的代码编辑功能	20	5.1 CVS介绍	66
2.3.1 添加注释	21	5.2 CVS安装	67
2.3.2 自定义格式化代码	21	5.3 在Windows上配置	
2.3.3 自动生成Getter和Setter代码	22	CVS服务器	69
2.3.4 代码的重构	23	5.3.1 服务器设置	
2.3.5 子类覆盖父类的方法	23	(Server Settings)	69
2.4 代码错误调试	24	5.3.2 配置代码库 (Repository	
2.5 文件查找	25	configuration)	71
2.6 使用快捷键	25	5.4 CVS常用术语	73
		5.5 Eclipse和CVS的连接	73
第3章 调试代码	27	5.6 启动新项目	76
3.1 调试器	27	5.7 检出代码	78
3.2 Eclipse调试视图	29	5.8 基本操作	79
3.2.1 调试视图	30	5.8.1 常用基本操作概述	79
3.2.2 变量视图	30	5.8.2 同步 (Synchronize)	80
3.2.3 断点	31		

5.8.3 解决冲突与合并 (Solve conflict and merge)	81	7.5 窗口控制事件	135
第6章 SWT基础	83	7.6 选择组件事件	137
6.1 SWT入门	83	7.6.1 选择事件常用属性	137
6.1.1 SWT简介	83	7.6.2 选择事件实例	138
6.1.2 构建SWT运行环境	86	7.7 其他常用事件	139
6.1.3 Display类	86	7.7.1 帮助请求事件	139
6.1.4 Shell类	90	7.7.2 文本修改事件	140
6.1.5 SWT包类结构	93	7.8 基金收益计算器	143
6.1.6 在SWT中使用Swing控件	94	第8章 SWT高级应用	148
6.2 SWT常用控件	96	8.1 SWT容器	148
6.2.1 SWT控件类概述	96	8.1.1 面板容器Composite	148
6.2.2 Button控件	98	8.1.2 分组容器Group	150
6.2.3 Label控件	100	8.1.3 选项卡 (TabFolder)	152
6.2.4 Text控件	101	8.1.4 分割框	154
6.2.5 List控件	103	8.2 菜单和工具栏	156
6.2.6 Combo控件	104	8.2.1 菜单	156
6.3 SWT布局管理	106	8.2.2 工具栏	161
6.3.1 概述	106	8.3 进度条和对话框	164
6.3.2 FillLayout布局	107	8.3.1 进度条	164
6.3.3 RowLayout布局	109	8.3.2 对话框	166
6.3.4 GridLayout布局	110	8.4 Tree控件	169
6.3.5 FormLayout布局	115	8.5 Table控件	172
6.3.6 StackLayout布局	118	8.6 SWT绘图	175
6.3.7 自定义布局	119	8.6.1 颜色和字体	175
6.4 开发一个简单的计算器	121	8.6.2 绘制对象GC	176
第7章 SWT事件	124	8.6.3 绘制图形	177
7.1 SWT事件介绍	124	8.7 SWT线程	179
7.1.1 事件模型概述	124	8.7.1 线程简介	179
7.1.2 事件简单例子	126	8.7.2 SWT UI线程	181
7.1.3 事件类型、监听器接口和适配器	127	8.7.3 线程实例	183
7.2 鼠标事件	129	8.8 SWT打印	184
7.2.1 鼠标事件常用属性	129	8.9 实现记事本功能	186
7.2.2 鼠标事件实例	130	第9章 JFace基础应用	191
7.3 键盘事件	132	9.1 JFace概述	191
7.3.1 键盘事件常用属性	132	9.1.1 JFace与SWT	191
7.3.2 键盘事件实例	133	9.1.2 创建JFace程序	192
7.4 焦点事件	134	9.2 应用程序窗口	194
		9.2.1 Window和WindowManager类	194

9.2.2 ApplicationWindow类	197	11.4.1 RCP应用程序开发	275
9.3 JFace对话框	200	11.4.2 RCP程序部署和发布	292
9.3.1 MessageDialog	200	11.4.3 导出RCP应用程序	294
9.3.2 InputDialog	204		
9.3.3 ErrorDialog	206	第12章 Eclipse表单	295
9.3.4 ProgressMonitorDialog	208	12.1 Forms概述	295
9.3.5 自定义对话框	210	12.1.1 Eclipse表单介绍	295
9.4 向导式对话框	213	12.1.2 SWT程序中使用表单	297
9.4.1 向导式对话框概述	213	12.2 常用控件	298
9.4.2 简单的向导式对话框	215	12.2.1 Form和ScrolledForm表单	298
9.4.3 复杂的向导式对话框	218	12.2.2 ExpandableComposite 折叠面板	301
9.5 首选项	221	12.2.3 内容区Section	304
9.5.1 保存首选项设置	222	12.2.4 超链接Hyperlink	306
9.5.2 首选项页面	223	12.2.5 表单文本	310
9.5.3 首选项页面树	225	12.3 布局管理器	315
第10章 JFace高级应用	229	12.3.1 表单布局	316
10.1 MVC的表格、树和列表	229	12.3.2 列布局	317
10.1.1 表格组件TableViewer	230	12.4 开发实例	319
10.1.2 树控件组件TreeView	240	第13章 Web开发	322
10.1.3 列表控件组件ListView	246	13.1 Web服务器	322
10.2 JFace的工具类	248	13.1.1 什么是Web服务器	322
10.2.1 字体资源管理类	248	13.1.2 Web服务器工作原理	323
10.2.2 图片资源管理	249	13.1.3 Tomcat服务器	324
10.2.3 颜色资源管理类	251	13.1.4 WebLogic服务器	325
10.3 文本处理	251	13.1.5 Resin服务器	326
10.3.1 IDocument接口	251	13.2 Eclipse中开发Web应用	327
10.3.2 TextViewer类	253	13.2.1 MyEclipse插件	327
10.4 JFace开发实例	254	13.2.2 新建Tomcat项目	329
10.4.1 代码着色	255	13.2.3 Servlet开发	334
10.4.2 主窗口	257	13.2.4 JSP开发	339
10.4.3 内容助手	259		
10.4.4 撤销与恢复	262	第14章 Eclipse中开发Struts	343
10.4.5 文件的打开与保存	263	14.1 Struts简介	343
10.4.6 Constants类	265	14.1.1 MVC模式简介	343
第11章 RCP技术	267	14.1.2 Struts框架	345
11.1 RCP的简介	267	14.1.3 Struts框架基本组件包	349
11.2 建立一个RCP的应用	268	14.2 Struts框架实例	350
11.3 RCP组成	273	14.3 Struts框架组件	357
11.4 RCP开发	275	14.3.1 控制器组件	357

14.3.2	视图组件	362	16.3.2	EclipseME插件	424
14.4	Struts应用配置	364	16.3.3	ColorEditor插件	427
14.4.1	web.xml	364	16.3.4	DbEdit插件	427
14.4.2	struts-config.xml	366	16.3.5	Eclipse UML2插件	429
14.5	Struts标签库	370	16.3.6	Log4E插件	431
14.5.1	HTML标签库	371	16.3.7	CDT插件	433
14.5.2	Bean标签库	379	16.3.8	Flow4J插件	437
第15章	Eclipse插件开发	385	16.3.9	Ant Build file Creator 插件	440
15.1	Eclipse基于插件的体 系结构	385	16.4	插件使用	441
15.1.1	Eclipse体系概述	385	16.4.1	JBossIDE插件使用实例	441
15.1.2	Eclipse插件模型	386	16.4.2	EMF插件使用	445
15.2	开发一个简单的视图 插件HelloWorld	388	第17章	公司人事管理系统	449
15.3	插件开发环境简介	392	17.1	系统概述	449
15.3.1	PDE透视图清单 配置编辑器	392	17.1.1	人事管理系统开发背景	449
15.3.2	PDE透视图代码编辑	395	17.1.2	系统需求和设计	450
15.4	使用PDE创建插件	396	17.2	数据库设计	451
15.4.1	创建插件	396	17.3	核心代码实现	453
15.4.2	插件文件分析	398	17.3.1	实现主界面	454
15.5	调试和运行插件	401	17.3.2	公共代码实现	458
第16章	常用插件	405	17.4	员工基本信息管理模块	460
16.1	Java开发插件	405	17.4.1	新员工加入代码实现	460
16.1.1	Doclipse插件	405	17.4.2	员工离职代码实现	463
16.1.2	Visual Editor插件	406	17.5	考勤管理模块	466
16.1.3	Pattern Box插件	407	17.6	出差管理模块	469
16.1.4	viPlugin插件	409	17.7	工资管理模块	472
16.1.5	Continuous Testing插件	409	第18章	Web图书管理系统	477
16.1.6	Code Analysis插件	410	18.1	需求分析	477
16.1.7	Designer插件	411	18.2	数据库设计	478
16.2	Web开发插件	414	18.3	系统公共类	482
16.2.1	HTMLEditor插件	414	18.4	图书管理模块	490
16.2.2	XMLBuddy插件	415	18.5	会员管理模块	497
16.2.3	JBossIDE插件	418	18.6	图书借阅管理模块	500
16.2.4	JspFormat插件	419	18.7	信息查询模块	502
16.2.5	CSSEditor	421	18.8	脚本源码	503
16.3	其他插件	422			
16.3.1	娱乐类插件	423			

第1章 Eclipse简介



内容摘要 | Abstract

目前Java开发领域的各种集成开发环境（IDE）呈现出百花齐放的局面，从Borland的JBuilder、Oracle的JDeveloper、WebGain的Visual Cafe，到开放源代码的Eclipse、NetBeans等共有十余种。在目前所有的IDE中，Eclipse可以说是最有发展前途的产品之一。Eclipse最初由OTI和IBM两家公司的IDE产品开发组创建。

Eclipse是替代IBM Visual Age for Java (IVJ) 的下一代IDE，但它的目标不仅仅是成为专门开发Java程序的IDE，而且根据Eclipse的体系结构，通过开发插件Eclipse还能扩展到任何语言的开发中，甚至能成为图片绘制的工具。目前，Eclipse已经开始提供C语言开发的功能插件。

另外一个重要点是，Eclipse是一个开放源代码的项目，任何人都可以下载Eclipse的源代码，并且在此基础上开发自己的功能插件；同时，可以通过开发新的插件扩展现有插件的功能，例如在现有的Java开发环境中加入Tomcat服务器插件。由于其可以无限扩展，而且有着统一的外观、操作和系统资源管理，所以可以说Eclipse具有巨大的发展潜力。



学习目标 | Objective

- 了解Eclipse的历史
- 了解Eclipse体系结构
- 正确安装Eclipse
- 掌握Eclipse插件的复制安装
- 掌握Eclipse插件的连接安装
- 了解如何查看Eclipse环境配置
- 使用Eclipse创建一个Java应用程序

1.1 Eclipse的历史

2001年11月，IBM向开源社区捐献了Eclipse代码，自此Eclipse逐渐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。

Eclipse起源的一个非常重要的原因是，当时IBM的开发工具Visual Age for Java和WebSphere Studio很难集成到一起，并且底层的技术也比较脆弱，这就使得进一步发展非常艰难，无法满足应用开发的需求。

因此，1998年，IBM开始了下一代开发工具技术探索之路，成立了一个项目开发小组，经过两年的发展到2000年，IBM内部给新一代的开发工具项目命名为Eclipse。这时候从商业

目标考虑, IBM希望Eclipse项目能够吸引更多的开发人员, 从而发展成为一个强大而又充满活力的商业合作伙伴社区, 同时也意识到需要用它来对抗Microsoft Visual Studio的发展。此外, IBM推出了Eclipse试用计划, 允许IBM以外的开发人员使用Eclipse, 从此Eclipse被广泛地得到应用, Eclipse也就成为这个项目的正式名称。

在早期的1.0版本中, Eclipse纯粹是一个Java集成开发平台, 类似JBuilder。为更好地适应用户需求, Eclipse被重新设计和定位, 并于2002年推出了2.x版本。从此, Eclipse开始转变自身角色, 从一个单一的集成开发环境转变成为一个开放的可扩展的集成平台。它可将单独开发的工具融合到精心设计的套件中, 并且可以很容易地将现有工具移植到平台中。

Eclipse 2.x在应用过程中达到了其所提供的目标, 但是该版本忽略应用的发布, 即当开发一套工具时, 可能需要将整个Eclipse集成开发环境都给用户打包。因此, Eclipse社团内部启动了一个新的研究项目——基于Eclipse的集成开发环境技术, 用来实现非集成开发环境的应用。也就是将Eclipse可扩展架构进行重用, 开发出具备Eclipse特性的应用, 最终成为富客户机程序(Rich Client)。

富客户机程序曾在20世纪90年代风靡一时, 但是随着Internet和基于Web的应用程序的不断发展, 瘦客户机程序(Thin Client)成为一项通用的技术。瘦客户机程序可以解决富客户机程序带来的诸如管理不便、升级成本高昂等问题。瘦客户机程序以牺牲用户界面的特色和高速的用户交互为代价, 降低了应用的部署和维护的费用, 因此而大受欢迎。但是瘦客户机程序从根本上说是一种功能和性能上的倒退。瘦客户机程序采用请求-应答模式, 所以需要大量的网络容量以确保最佳的交互效果。

随着应用和用户本身越来越复杂, 以及大量新需求的出现, 瘦客户机对这些应用就显得无能为力, 于是富客户机的需求变得越来越强烈。但是富客户机本身固有的部署和维护问题如何解决——Eclipse 3.x为富客户机程序提供了全新的解决方案。它可以充分利用Eclipse插件化的特点, 彻底地将与集成开发环境相关的依赖项从中剥离, 同时更多的用户界面组件被开发并允许个性化的定制。

1.2 Eclipse体系结构

由于Eclipse提供了一个开放灵活的框架, 随着Eclipse技术的广为流行, 基于Eclipse构架的应用也会越来越多。Eclipse的特点是: 跨平台; 提供了GUI和非GUI; 语言中立; 基于插件的开放的、可扩展的平台。本节将对Eclipse的结构进行了剖析, 并分析Eclipse使用的技术及其实现原理。

应用软件在Eclipse中被称为“插件”, 它们在架构中互相提供服务。Eclipse的结构体系如图1-1所示。

Eclipse类似于“总线”的体系结构, 其核心部分Platform Runtime类似于一条“即插即用”的“总线”, 它提供了许多类似于“插槽”的扩展点。其余部分都可看成类似于外部设备的“插件”, 可随时加载和卸载。它更为灵活的是每个插件又提供了扩展点可继续安装其他的插件。

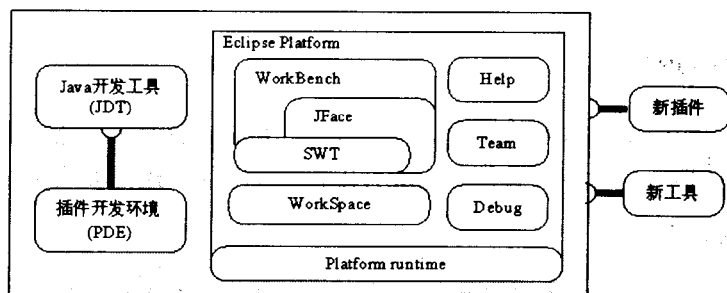


图1-1 Eclipse体系结构图

在整个Eclipse框架中，可以分为Eclipse平台、JDT、PDE、其他的插件等几个部分。其中Eclipse平台是整个系统的基础与核心。JDT是提供了用于编辑、查看、编译、调试和运行Java代码的专门插件。插件开发环境PDE是由Eclipse提供的一个完整的环境，以帮助开发、测试插件并对插件进行打包。PDE还创建了模板代码，用户可以使用这些模板代码来构建自己的插件。在为插件实现所需的接口和文件时，这些代码可以节省大量的开发时间和精力。

Eclipse平台是一个通用的基础架构，它由平台运行时（Platform runtime）、工作空间（WorkSpace）、工作台（WorkBench）、帮助（Help）、小组（Team）、调试（Debug）构成。平台在结构上大致可分为核心和用户界面UI两层。核心包含平台运行时和插件管理、工作空间和资源管理以及版本和配置管理。其中平台运行时又是Eclipse平台的核心。

1. 平台运行时（Platform Runtime）

它相当于一个微内核，是插件的核心引擎。它位于平台核心中最低级别的部分，由org.eclipse.osgi和org.eclipse.core.runtime两个插件组成，其他所有的插件都依赖于这两个插件。

平台运行时（Platform Runtime）的功能主要包括：定义插件的结构以及它们包含的实现详细信息；查找和执行主要的Eclipse应用程序以及维护插件的注册表、扩展及扩展点；提供实用程序。

平台运行时（Platform Runtime）管理着一个插件注册表，当Eclipse启动时，运行时的内核先定位JRE的位置，然后启动startup.jar，扫描plugins和features目录下的插件配置文件，对插件进行初始化注册，并保存配置文件中的信息；之后查找清单文件中声明的扩展点和扩展，将二者匹配，并保存插件的依赖关系，最后再启动应用。

注意，只有当需要使用插件时，运行时才会将其调入内存；不需要时则选择适当的时机清除出内存。

2. 用户界面UI

Eclipse用户界面（UI）的三个主要部分组成是：Workbench、JFace和Standard Widget Toolkit (SWT)。Workbench是包含和集成了Eclipse中所有的视图、工具栏、菜单栏和编辑器的主窗口。JFace是构建于SWT之上的高级用户界面（UI）小部件库；SWT定义了用于处理UI编程任务和操作的常用小部件。SWT定义了一组UI小部件，而这些小部件是Eclipse中所有UI工具的基础。

1.3 安装Eclipse

目前, Eclipse的最新版本为3.x系列, 需要注意, Eclipse仅是一个集成开发平台, 它本身并不包含语言编译器。因此, 在使用Eclipse开发Java程序前, 还需要安装JDK。

1.3.1 安装JDK

JDK是许多Java程序开发人员最初使用的开发环境, 尽管许多编程人员已经使用第三方的集成开发工具, 但是JDK仍被认为是Java开发的重要工具。JDK包含了所有编译、运行Java程序所需的工具: 基本库、Java编译器、Java虚拟机和Applet浏览器。

安装JDK的第一步是下载安装文件, Sun公司的官方下载地址为: <http://java.sun.com/javase/downloads/index.jsp>。本书所使用的JDK为1.6版本, 通常情况下直接双击下载的文件即可开始安装JDK。

在安装JDK时可以选择安装的项目, 如图1-2所示, 依次为开发工具 (Development Tools)、演示程序 (Demo)、API源代码 (Source Code)、公共JRE与Java DB。其中, 开发工具是必需的; 演示程序可以供日后编写程序时参考; 通过API源代码可以了解所使用的API实际上是如何编写的; 而JRE是执行Java程序所必要的; 另外, JDK 6中还提供了对开源Apache数据库的支持。

在安装JDK时要注意JDK的安装位置, 该信息会在设置系统环境变量时用到, 可以通过单击“更改”按钮改变JDK的安装位置。单击“下一步”按钮后开始安装JDK, 完成JDK安装后, 接着会安装“公共JRE”。

同样应注意图1-3中关于JRE的安装地址, 单击“更改”按钮可以改变JRE的安装路径。

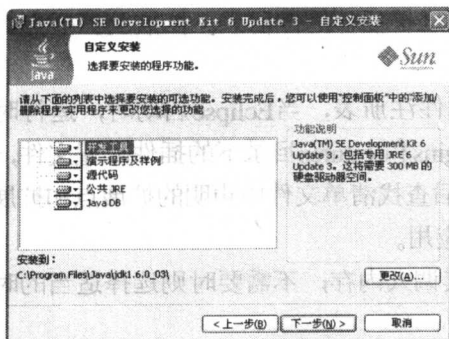


图1-2 安装JDK的界面

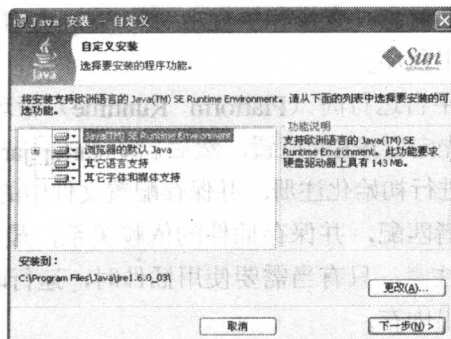


图1-3 安装JRE

安装完JDK和JRE后, 还需要设置计算机系统的环境变量。与JDK相关的环境变量包括path和classpath, 其中环境变量path指定JDK中所用工具的存放位置, classpath指定JDK中基本类的存放位置。

在安装好JDK程序后, 在JDK安装目录下的bin目录中会提供一些开发Java程序时必备的工具程序, 所以设置path环境变量的值就应该是到该bin目录下的路径。在Windows 2000/XP环境下, 对path变量最简单的设置是选择桌面上的“我的电脑”, 并右击, 从弹出的菜单中

选择“属性”，切换属性页到“高级”页面，单击下方的“环境变量”按钮，在环境变量对话框中编辑Path变量，如图1-4所示。

双击Path变量就可以打开如图1-5所示的“编辑系统变量”对话框，在该对话框的变量值文本框中输入“JDK安装目录\bin”，注意在最后应使用一个分号作为路径设置的分隔。

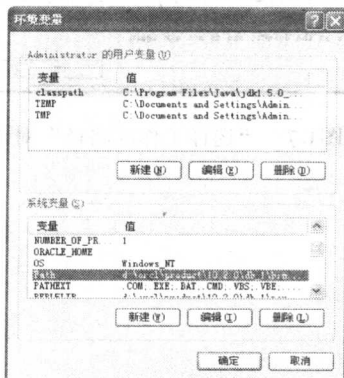


图1-4 设置Path变量

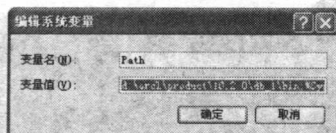


图1-5 编辑Path变量

也可以在命令提示符下执行以下的命令，直接设置当前的环境变量，包括path变量。注意这种设置方法在关闭当前命令提示符窗口后将会失效。

```
set Path=C:\Program Files\Java\jdk1.6.0\bin;
```

与path变量类似，设置classpath变量最简单的方法是在系统变量中新增一个变量，classpath变量的值为：JDK安装目录\lib。其中，lib目录下存放了一些.jar文件，这些文件是zip压缩格式，其中包括其本类库的.class文件。总之，classpath变量的目的是告诉Java执行环境，在哪些目录下可以找到执行Java程序所需的类库。

对JDK进行正确安装和配置是在Eclipse下创建并运行Java程序的前提条件。

1.3.2 Eclipse安装

由于Eclipse是一个开放源代码的项目，可以到其官方网站www.eclipse.org上免费下载Eclipse的最新版本。本书所使用的Eclipse为Windows平台下的3.2版。

虽然Eclipse本身是用Java语言编写的，但下载的压缩包中并不包含Java运行环境，需要用户自己另行安装JRE，并且要在操作系统的环境变量中指明JRE中bin的路径。

安装Eclipse的步骤非常简单：只需将下载的压缩包按原路径直接解压即可。需注意如果有了更新的版本，要先删除老的版本重新安装，不能直接解压到原来的路径覆盖老版本。在解压缩之后可以到相应的安装路径中去找Eclipse.exe运行。启动Eclipse，会出现一个由弯月图片构成的闪屏（如图1-6所示）。随后出现一个如图1-7所示的“选择工作区路径”对话框，Eclipse会将编辑的所有文件存放在工作区指定的路径下。

确定工作区路径后单击“OK”按钮，打开如图1-8所示Eclipse的欢迎主界面。

如果在未安装JDK前启动Eclipse，系统会出现图1-9所示的错误信息。提示用户先安装JDK，并正确配置后再重新启动。

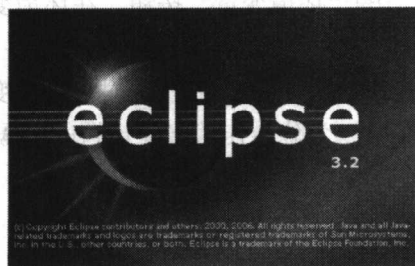


图1-6 Eclipse启动界面

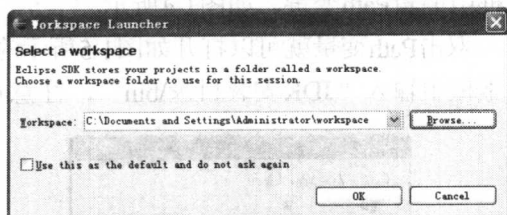


图1-7 “选择工作区路径”对话框

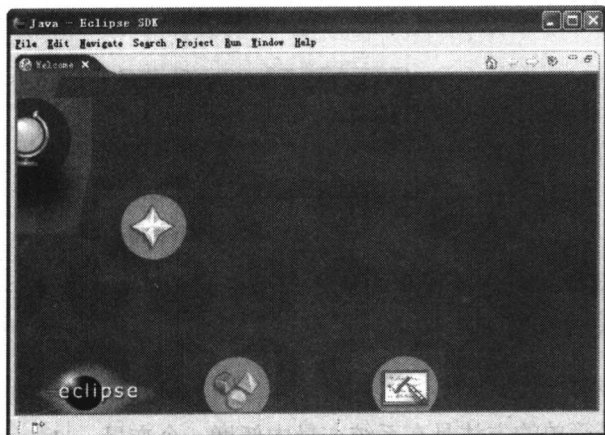


图1-8 Eclipse欢迎主界面

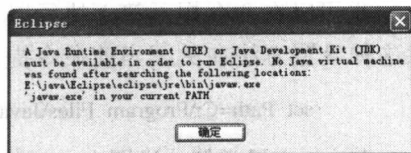


图1-9 “未发现JDK”对话框

1.4 Eclipse插件安装

Eclipse是一个开放源代码的、基于Java的可扩展开发平台。虽然Eclipse主要被用来当作Java IDE使用,但Eclipse的目标不仅限于此。就其本身而言,它只是一个框架和一组服务,用于通过插件组件构建开发环境。即通过安装Eclipse插件,可以无限扩展Eclipse平台。Eclipse插件的安装可以分为两种方式:一种是内置方式安装,即复制安装;另一种是链接安装。下面以语言汉化包的安装为例,介绍Eclipse插件的两种安装方式。

1.4.1 复制安装

完成Eclipse安装后,在Eclipse的主目录下应包括plugins和feature目录,复制安装方法只需要下载插件后解压缩,然后把解压后插件的plugins和features目录里面的内容分别复制到Eclipse安装目录的plugins和feature目录中。下面是汉化语言包的复制安装方法:

(1) 下载合适版本的汉化语言包。注意,在所有的插件安装时,必须注意插件与Eclipse版本的匹配问题。本书使用的Eclipse版本号为3.2版,所以下载的汉化包应为: NLpack1-eclipse-SDK-3.2-win32.zip。

(2) 解压缩NLpack1-eclipse-SDK-3.2-win32.zip文件。解压缩后的文件将包含一个名为eclipse的目录,在该目录中找到plugins和features目录。

(3) 将解压缩后的plugins和features目录下的文件复制到Eclipse安装目录下plugins和features的相应文件夹下。

(4) 重新启动Eclipse, 可以发现Eclipse平台已经变为中文界面。

这种方法安装比较简单, 但是问题也比较多, 因为所有的插件的文件都混合到了plugins和feature目录里面, Eclipse会变得越来越臃肿, 而且当不再需要使用某个插件时, 卸载比较麻烦。所以对插件的安装推荐使用链接安装。

1.4.2 链接安装

插件的另外一种安装方法就是被称为使用外挂的方法, 即Links安装。下面仍以Eclipse3.2汉化包的安装加以说明:

(1) 首先把插件放在一个单独的目录中(如E:\plug-in), 然后进行解压缩。例如将NLpack1-eclipse-SDK-3.2-win32.zip解压成NLpack1-eclipse-SDK-3.2-win32目录, 最终的目录结构为: E:\plug-in\NLpack1-eclipse-SDK-3.2-win32\eclipse\plugins。这样Eclipse将会到指定的目录下去查找eclipse\features目录和eclipse\plugins目录, 查看是否有合法的功能部件和插件。也就是说, 目标目录必须包含一个\eclipse目录。

(2) 然后在Eclipse安装目录中建立一个子目录links, 在links目录中建立一个link文件。link文件用于指定添加到Eclipse平台的插件所在位置。例如, 新建文件LanguagePackFeature.link, 其内容指定解压后的插件目录: path=E:\plug-in\NLpack1-eclipse-SDK-3.2-win32。

(3) 重新启动Eclipse。

使用links安装插件后, 如果想要重新安装某一个插件, 只要将文件替换或者修改link文件的路径即可。使用links安装插件时需要注意以下几点:

- 插件可以分别安装在多个自定义的目录中。
- 一个自定义目录可以安装多个插件。
- link文件的文件名及扩展名可以取任意名称, 例如ddd.txt, myplugin都可以。
- link文件中path=插件目录的path路径的分隔符要用反双斜杠“\\”或正斜杠“/”。
- 在links目录下也可以有多个link文件, 每个link文件中的path参数都将生效。
- 插件目录可以使用相对路径。

1.4.3 查看插件

在完成Eclipse插件安装后, 用户如何知道当前Eclipse平台都安装了哪些插件, 以及插件是否安装正确? 为此Eclipse提供了一个查看安装插件的功能。

要查看已安装插件, 可以通过Eclipse平台上的“帮助”|“关于Eclipse SDK”菜单命令, 打开“关于Eclipse SDK”对话框, 单击其中的“功能部件详细信息”按钮, 打开如图1-10所示的“关于 Eclipse SDK功能部件”对话框。

图1-10所示是安装汉化工具包后显示的Eclipse中已经完成安装的功能部件。图1-11所示显示了安装新插件后在Eclipse平台中可以使用的功能部件。

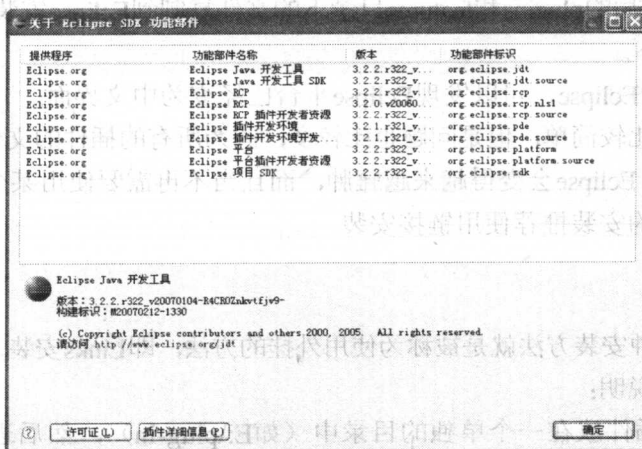


图1-10 Eclipse中已经完成安装的功能部件

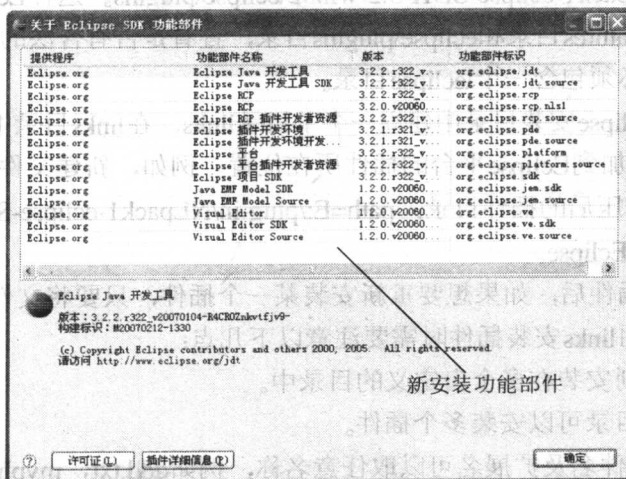


图1-11 查看新安装的功能部件

1.5 Eclipse示例

Java开发工具（JDT）是随Eclipse平台一起交付的插件。JDT与其他插件一样是对工作台的一组扩展，它允许用户编辑、编译和运行Java程序。在本节中将使用JDT创建一个简单的Java程序。

启动Eclipse，关闭欢迎界面，进入如图1-12所示的Eclipse主界面。

在Eclipse工作区中，对任何程序都是以项目为单位进行管理的。项目也是工作区使用的最大结构化单元，项目中包含文件夹和文件，并且可以打开、关闭或构造文件夹和文件。

新建Java项目的步骤如下：

(1) 从菜单栏中选择“文件”|“新建”|“项目”，打开“新建项目”对话框，在其中选择“Java”|“Java项目”，如图1-13所示。