第一图书网, tushu007.com

<<Eclipse RCP与Spring O>>

图书基本信息

书名：<<Eclipse RCP与Spring OSGi>>

13位ISBN编号：9787111402183

10位ISBN编号：7111402189

出版时间：2012-12-15

出版时间：机械工业出版社华章公司

作者：陆阳

页数：394

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：http://www.tushu007.com

前言

为什么要写这本书几年前笔者就有编写一本关于Eclipse RCP和OSGi技术的书籍的想法，但几次提笔又放弃了，因为未能很好地解决在开发实践中遇到的很多问题，笔者不希望将一些不成熟的想法和开发“玩具”程序的经验传播给其他开发者。
最近笔者完成了一个综合性的项目，觉得自己已有了足够的经验，已可以完成这样一本书了。
此时WindowBuilder界面设计工具已经被Google公司收购并免费开放给开发者使用了，使用该工具时不再有软件版权问题，于是笔者真正开始动笔写作本书了。
随着现代软件开发规模愈来愈庞大、愈来愈复杂，模块化开发会成为一种必然趋势。
Eclipse RCP 和OSGi技术正是解决这一系列问题的一剂良药。
使用Eclipse RCP和OSGi技术开发程序所面临的最大挑战不是编写代码，而是如何规划插件的功能边界，如何开发松耦合、可复用的插件系统。
另外，开发数量庞大的插件在项目管理和版本控制上也是一个挑战，有时候开发者不得不在是否应当细化插件功能的颗粒度和由此造成过多的插件数量之间权衡。
无论是桌面系统开发还是Web系统开发，在国外Eclipse RCP和OSGi都早已有非常广泛的应用，但是近年来在国内依然没能得到有效的推广。
鉴于国内Spring技术书籍已经非常丰富，开发者对Spring技术也相对熟悉，所以本书结合Eclipse RCP和Spring OSGi以实践的方式来传播新技术，这样读者可以在已有的知识结构上学习新东西，以缩短学习周期，迅速掌握技术要点。
本书中的技术既能够满足Eclipse客户端软件的开发者，又能够兼顾Java Web应用程序的开发者。
笔者非常赞同Eclipse技术社区中所倡导的软件开发精神：贡献、分享、邀请，每个人都需要别人的关怀和帮助，每个人同样有义务关怀和帮助别人。
Eclipse与OSGi技术在实践开发中应用起来相对困难，需要整合很多第三方技术与辅助工具，开发环境的配置也较为烦琐。
本书中的开发经验大多来自于一个具有上百个插件的综合性软件平台，笔者将其中一些有益的设计经验和技巧抽离出来，尽可能多地向读者展示开发环境的搭建、工具的配置、编程经验、解决问题的思路以及如何获取更多的技术资源等知识。
Eclipse本身的设计非常复杂，项目庞大，更多的编程经验还需要读者在实践开发中不断积累。
阅读Eclipse的源码和下载Eclipse的CVS示例程序都是学习相关技术的有效途径。
而就在笔者完成本书的时候，新一代的Eclipse应用平台核心运行框架Eclipse 4 Juno版本也如期发布，更多新的技术特性得到支持，包括：跨产品平台、自定义的依赖注入（Define custom DI-Annoations）、新的CSS外观技术、向后兼容性等。
而基于企业级的OSGi 5.0规范也将会在EclipseCon 2012会议和OSGiCon 2012会议后得到广泛应用，一系列新的技术特性将在新版本中发布。
我们虽然不能通过一本书就越过那道门，但是至少可以开启一扇窗户，看到门外的世界。
对于那些看上去是障碍和壁垒的技术，亲自接触一下，也是有价值的，哪怕只有几秒，也比旁观几百次有效。
读者对象本书面向的读者是使用Eclipse RCP和Spring OSGi技术开发应用系统的程序设计人员，也包括之前就已经使用Java编程语言编写过桌面或Web应用程序的开发者。
对于准备使用Spring OSGi开发Web应用程序的开发者，本书的大部分概念和知识也同样适用。
本书假设读者已经熟悉了Java的语法和特点，并对Java设计模式有了一定了解，例如适配器模式、监听器模式、模板模式等。
本书内容编排的基本原则是任何具有基本Java知识的人都可以理解。
除此之外，开发者应该掌握一些常用的开源技术，比如Spring开发框架、Hibernate ORM开发框架。
如何阅读本书本书分为三大部分：第一部分为基础篇，介绍了Eclipse RCP开发所使用的常用界面编程技术，涵盖了SWT、JFace、Forms、Nebula和WindowBuilder设计器的使用方法。
读者应当熟练掌握布局和各种控件的使用方法，并且通过访问这些技术的官方网站来获取更多的编程

技巧。
第二部分为高级篇，着重讲解了Eclipse RCP软件产品的组成部分，读者应当充分理解RCP开发中关于运行平台和目标平台、扩展和扩展点的概念。
掌握这些技术点有助于读者构建一个结构完整的Eclipse RCP软件，解决软件开发过程中的故障。
第三部分为实战篇，详细讲解了Eclipse RCP与Spring OSGi框架、Hibernate ORM框架、JPA规范、Maven工具的整合。
如果读者在此之前从未接触过Spring和Hibernate技术，建议阅读一些与此相关的资料。
对于Eclipse RCP和Spring OSGi技术的初学者，建议采用循序渐进的方式逐章学习。
对于已经具备一定开发经验的读者，可以直接选择感兴趣的内容阅读。
本书的技术点都尽可能地与业务逻辑剥离，以减少无关的程序代码实现，从而使读者更好地关注技术本身，减少对读者学习的干扰。

内容概要

本书由资源的Eclipse专家亲自执笔，并得到了Eclipse官方技术社区的强烈推荐，权威性毋庸置疑！
内容全面，系统讲解了利用Eclipse RCP和Spring OSGi开发大规模Java应用的核心技术；实战性强，包含大量易于操作的案例和最佳实践。

全书共分3个部分：基础篇（第1~5章）详细介绍了与Eclipse RCP相关的一系列核心概念、Eclipse RCP开发环境的搭建，以及SWT、JFace、Forms、Nebula和WindowBuilder等Eclipse RCP开发所常用的界面编程技术；高级篇（第6~12章）系统讲解了Eclipse RCP应用开发的基础知识、Eclipse RCP软件产品各个组成部分的构建方法，以及Eclipse RCP扩展的使用和扩展点的开发，掌握这些技术知识的读者将能构建一个结构完整的Eclipse RCP软件，并解决软件开发过程中遇到的故障；实战篇（第13~15章）详细讲解了Eclipse RCP与Spring OSGi框架、Hibernate ORM框架、JPA规范、Maven工具的整合，以及它与Java的模块化设计。

作者简介

陆阳，资深Eclipse技术专家，致力于OSGi和Eclipse技术的研究和推广，积累了十分丰富的经验。精通Java语言和各种开源的企业级Java技术框架的整合与开发；擅长Eclipse RCP 富客户端系统框架的设计、基于OSGi的BS/CS系统的架构、基于Java工作流中间件的设计及其应用，以及基于MDA
可视化建模软件的开发和设计。
活跃于Eclipse（www.eclipse.org.org）技术社区，是Eclipse
技术社区Hibernate/JPA 设计器的开发者，被提名Eclipse2012 Community-Top Newcomer Evangelist Award。

书籍目录

推荐序
前言
第一部分　基础篇
第1章　初识Eclipse RCP
1.1　Eclipse平台概述
1.2　Eclipse RCP概述
1.3　Eclipse RCP的应用
1.4　OSGi与Equinox
1.5　SWT与JFace
1.6　小结
第2章　搭建Eclipse RCP开发环境
2.1　JDK的安装与配置
2.1.1　下载JDK安装程序
2.1.2　安装JDK运行程序
2.1.3　配置JDK运行环境
2.1.4　测试JDK安装状态
2.1.5　卸载JDK应用程序
2.2　安装Eclipse IDE
2.2.1　Eclipse开发包
2.2.2　Eclipse版本更新
2.2.3　下载Eclipse
2.2.4　安装Eclipse
2.2.5　技术资源
2.3　Maven的安装与配置
2.3.1　Maven介绍
2.3.2　下载Maven
2.3.3　安装Maven
2.3.4　测试Maven
2.3.5　修改Maven的配置文件
2.3.6　技术资源
2.4　Maven和Eclipse的集成
2.4.1　安装Maven Eclipse插件
2.4.2　配置Maven Eclipse插件
2.4.3　技术资源
2.5　安装Spring IDE Plugin
2.5.1　Spring IDE介绍
2.5.2　安装Spring IDE插件
2.5.3　技术资源
2.6　安装WindowBuilderProPlugin
2.6.1　WindowBuilder介绍
2.6.2　安装WindowBuilder插件
2.7　小结
第3章　SWT界面设计详解
3.1　SWT介绍
3.2　WindowBuilder插件初识

3.2.1 设计视图
3.2.2 源码视图
3.2.3 结构视图
3.3 SWT设计面板
3.4 SWT设计准备
3.4.1 Element
3.4.2 ElementFactory
3.5 Shell使用入门
3.5.1 Shell样式
3.5.2 Shell TRIM
3.5.3 Shell模态类型
3.6 容器使用入门
3.6.1 Composite
3.6.2 Group
3.6.3 ScrolledComposit
3.6.4 SashForm
3.6.5 TabFolder与TabItem
3.6.6 CTabFolder与CTabItem
3.6.7 ViewForm
3.6.8 CBanner
3.7 容器使用进阶
3.7.1 checkSubclass方法
3.7.2 界面复用
3.7.3 添加自定义的Component
3.7.4 背景模式
3.7.5 带分隔条的SashForm
3.8 布局使用入门
3.8.1 AbsoluteLayout
3.8.2 FillLayout
3.8.3 GridLayout
3.8.4 FormLayout
3.8.5 RowLayout
3.8.6 StackLayout
3.8.7 GroupLayout
3.8.8 FlowLayout
3.8.9 BoxLayout
3.8.10 BorderLayout
3.9 布局使用进阶
3.9.1 使用AbsoluteLayout
3.9.2 使用GridLayout
3.9.3 StackLayout
3.10 控件使用入门
3.10.1 Label
3.10.2 CLabel
3.10.3 Separator
3.10.4 Text
3.10.5 StyledText

3.10.6 Combo
3.10.7 CCombo
3.10.8 List
3.10.9 Button
3.10.10 DateTime
3.10.11 Spinner
3.10.12 Slider
3.10.13 Scale
3.10.14 Link
3.10.15 ToolBar与ToolItem
3.10.16 CoolBar与CoolItem
3.10.17 ExpandBar与ExpandItem
3.10.18 ProgressBar
3.10.19 Browser
3.10.20 Canvas
3.10.21 TrayItem
3.10.22 Table
3.10.23 Tree
3.10.24 TableTree
3.11 控件使用进阶
3.11.1 创建Simple Table
3.11.2 创建Lazy Table
3.11.3 创建Virtual Table
3.11.4 Virtual Table注意事项
3.11.5 Virtual Table排序
3.11.6 创建Lazy Tree
3.11.7 创建Virtual Tree
3.11.8 创建TreeVirtualDynamic
3.11.9 创建POJOComposite
3.11.10 如何与浏览器交互
3.12 菜单使用入门
3.12.1 Menu Bar组件
3.12.2 Popup Menu
3.12.3 MenuItem
3.13 菜单使用进阶
3.14 技术资源
3.15 小结
第4章 JFace界面设计详解
4.1 MVC设计思想
4.1.1 MVC模式
4.1.2 Java与MVC模式
4.2 JFace与MVC模式
4.2.1 JFace Viewer框架概述
4.2.2 JFace Viewer框架机制
4.3 JFace常用组件入门
4.3.1 ComboViewer
4.3.2 ListViewer

4.3.3 TableViewer
4.3.4 TreeViewer
4.3.5 CheckboxTableViewer
4.3.6 CheckboxTreeViewer
4.3.7 TreeTableViewer
4.3.8 TableViewer Composite
4.3.9 Table Composite
4.3.10 TreeViewer Composite
4.3.11 Tree Composite
4.4 JFace组件使用进阶
4.4.1 数据绑定
4.4.2 基于EditingSupport的可编辑表格
4.4.3 基于ICellModifier的可编辑表格
4.4.4 对话框编辑器
4.4.5 其他表格标签
4.4.6 Viewer浮动提示控件
4.5 小结
第5章 Eclipse Forms与Nebula界面设计
5.1 Eclipse Forms项目介绍
5.2 Eclipse Forms使用入门
5.2.1 Composite
5.2.2 ExpandableComposite
5.2.3 Label与Text
5.2.4 Hyperlink与ImageHyperlink
5.2.5 Composite Separator
5.2.6 Section
5.2.7 Form与ScrolledForm
5.3 Eclipse Forms使用进阶
5.3.1 FormEditor与FormPage
5.3.2 MasterDetailsBlock 与IDetailsPage
5.4 Eclipse Forms技术资源
5.5 Nebula项目介绍
5.6 安装Eclipse Nebula Plugin
5.6.1 安装Eclipse Nebula Main Widgets
5.6.2 安装Eclipse Nebula incubation Widgets
5.6.3 加载Eclipse Nebula控件到WindowBuilder Pro Plugin中
5.7 Nebula Main Widgets设计初识
5.8 Nebula Incubation widgets设计初识
5.9 Nebula设计经验
5.10 Nebula技术资源
5.11 小结
第二部分 高级篇
第6章 Eclipse RCP 开发基础
6.1 Eclipse概念
6.1.1 欢迎界面
6.1.2 工作台
6.1.3 透视图

6.1.4 视图
6.1.5 编辑器
6.2 RCP应用程序设计
6.2.1 RCP插件创建向导
6.2.2 Plug-in Project
6.2.3 Plugin Content
6.2.4 Templates
6.2.5 Sample Intro
6.2.6 产品初次运行
6.3 插件清单编辑器
6.3.1 插件概述
6.3.2 插件依赖项
6.3.3 插件运行时
6.3.4 插件扩展
6.3.5 插件扩展点
6.3.6 插件构建
6.3.7 其他页面
6.4 RCP Activator与Advisor类解析
6.4.1 WorkshopAppActivator
6.4.2 Application
6.4.3 ActionBarAdvisor
6.4.4 WorkbenchAdvisor
6.4.5 WorkbenchWindowAdvisor
6.5 小结
第7章 构建产品帮助系统
7.1 创建帮助插件向导
7.1.1 帮助插件的 Content向导页面
7.1.2 帮助插件的Templates向导页面
7.1.3 帮助插件的Sample Help Table of Contents向导页面
7.2 维护Help Content
7.2.1 帮助系统配置与资源文件的目录
7.2.2 帮助系统配置与资源文件的关系
7.2.3 帮助系统扩展
7.3 添加帮助的菜单项
7.4 帮助文件的创建
7.5 小结
第8章 构建产品欢迎页面
8.1 产品绑定欢迎页面
8.1.1 欢迎页面扩展
8.1.2 欢迎页面定义
8.1.3 欢迎页面首页
8.1.4 页面跳转链接
8.2 欢迎页面请求
8.2.1 创建IIntroAction
8.2.2 添加IIntroAction链接
8.3 添加动态内容
8.3.1 动态页面创建

8.3.2　动态数据创建
8.4 创建页面扩展
8.5 小结
第9章　构建产品多国语言
9.1　插件清单的多国语言
9.2　Java程序的多国语言
9.2.1　Java程序的外部化字符串
9.2.2 Java程序的多国语言包
9.2.3 产品多国语言进阶
9.3 帮助系统的多国语言
9.4 Eclipse界面汉化
9.5 切换软件系统语言
9.5.1 添加自定义产品的config文件
9.5.2 添加语言切换扩展点
9.5.3 语言切换处理器
9.6 小结
第10章　产品的运行与导出
10.1 产品配置
10.2 产品配置编辑器
10.3 导出向导
10.3.1 产品导出
10.3.2 跨平台导出
10.3.3　插件导出
10.4　运行与导出进阶
10.5 小结
第11章　Eclipse RCP扩展的使用
11.1 ViewPart 视图
11.1.1　创建新插件
11.1.2　视图向导
11.1.3　视图代码
11.1.4 视图扩展
11.2　Perspective透视图
11.2.1　透视图向导
11.2.2　透视图代码
11.2.3　透视图扩展
11.2.4　透视图布局
11.2.5　打开透视图
11.3　EditorPart编辑器
11.3.1　创建数据表单
11.3.2　创建编辑器的输入
11.3.3　创建编辑器
11.3.4　创建视图树形组件
11.3.5　编辑器扩展
11.4　编程实现Action
11.4.1　IAction
11.4.2　ToolBar工具栏
11.4.3　CoolBar工具栏

11.4.4　ContentMenuBar菜单栏
11.4.5　工作台Action
11.4.6　视图Action菜单栏
11.4.7　视图Action工具栏
11.4.8　编辑器Action工具栏
11.5　扩展实现Action
11.5.1　工作台Action扩展
11.5.2　视图Action扩展
11.5.3　编辑器Action扩展
11.6　Action的键盘绑定
11.6.1　Commands扩展
11.6.2　Bindings扩展
11.7　Eclipse RCP设计进阶
11.7.1　视图编辑器之间通信
11.7.2　创建状态栏项
11.7.3　创建属性配置页面
11.7.4　创建登录对话框
11.8　小结
第12章　Eclipse RCP扩展点开发
12.1　扩展点简介
12.2　创建扩展点
12.3　扩展点编辑器
12.3.1　扩展点概述
12.3.2　扩展点定义
12.3.3　扩展点源码
12.4　扩展点处理
12.4.1　扩展点对象
12.4.2　扩展点注册
12.4.3　扩展点获取
12.5　添加扩展
12.6　使用扩展
12.7 小结
第三部分　实战篇
第13章　Eclipse RCP与Spring OSGi
13.1　OSGi技术介绍
13.1.1　软件开发面临的问题
13.1.2　动态模块化解决方案
13.1.3 OSGi框架
13.1.4　选择合适的OSGi实现
13.2　Spring OSGi项目所需要的开发包
13.2.1　下载Spring DM的开发包
13.2.2　解压缩Spring DM的开发包
13.2.3　创建Spring DM目标平台
13.3　Spring OSGi项目的Maven管理
13.3.1　创建Maven POM项目
13.3.2　修改Maven项目类型
13.4　Spring OSGi项目的模块化设计

13.4.1 创建Spring DM业务项目
13.4.2 设计Spring DM业务项目
13.5 Spring OSGi项目转为插件项目
13.6 Spring OSGi插件项目发布接口
13.6.1 创建接口发布配置
13.6.2 添加包的路径导出
13.7 Eclipse RCP获取Spring OSGi服务
13.7.1 创建Eclipse RCP UI 插件
13.7.2 获取Spring OSGi服务接口
13.7.3 调用Spring OSGi服务接口
13.8 Eclipse RCP与Spring OSGi的启动
13.8.1 Eclipse 应用程序启动选项
13.8.2 启动Eclipse RCP产品配置
13.8.3 启动基于Spring OSGi的RCP程序
13.9 在Eclipse RCP中获取ApplicationContext和Bean
13.10 Spring OSGi的Maven插件项目打包
13.10.1 PDE插件导出向导
13.10.2 Maven插件项目的打包配置
13.10.3 执行Maven插件打包
13.11 小结
第14章 Eclipse RCP与数据访问
14.1 SpringOSGi与Hibernate
14.1.1 Hibernate简介
14.1.2 数据访问插件目标
14.1.3 数据访问插件设计
14.2 插件如何引入第三方开发包
14.2.1 如何获取第三方开发包
14.2.2 如何引用第三方开发包
14.3 将普通JAR转成插件格式
14.3.1 导入向导
14.3.2 BND工具
14.4 创建数据库插件
14.4.1 Maven POM配置文件
14.4.2 properties配置文件
14.4.3 Spring XML配置文件
14.4.4 MAINFEST.MF配置文件
14.5 创建连接池插件
14.5.1 Maven POM配置文件
14.5.2 properties配置文件
14.5.3 Spring XML配置文件
14.5.4 Java类设计
14.5.5 MAINFEST.MF配置文件
14.6 创建连接池Feature插件
14.6.1 修改DBCP的MAINFEST.MF文件
14.6.2 创建DBCP的Fragment插件
14.7 创建Hibernate插件
14.7.1 Maven POM配置文件

14.7.2 properties配置文件
14.7.3 Spring XML配置文件
14.7.4 扩展点设计
14.7.5 Java类设计
14.7.6 MAINFEST.MF配置文件
14.8 创建业务插件
14.8.1 Spring XML配置文件
14.8.2 Java类设计
14.8.3 数据库SQL设计
14.8.4 MAINFEST.MF配置文件
14.8.5 添加BundleAnnotationSessionFactoryBean扩展
14.9 启动OSGi Framework程序
14.10 其他整合数据库的方式
14.11 基于数据访问插件的RCP程序启动
14.12 小结
第15章 Eclipse与Java的模块化设计
15.1 基于编程语言的模块化
15.1.1 Java语句的模块化设计
15.1.2 Java方法的模块化设计
15.1.3 Java类的模块化设计
15.1.4 Java接口的模块化设计
15.2 基于编程方法的模块化
15.2.1 IoC编程的模块化设计
15.2.2 AOP编程的模块化设计
15.3 基于程序物理文件的模块化
15.3.1 Package路径的模块化设计
15.3.2 XML文件的模块化设计
15.3.3 Properties文件的模块化设计
15.3.4 JAR组件的模块化设计
15.4 基于技术架构的模块化
15.4.1 OSGi的模块化设计
15.4.2 Eclipse插件的模块化设计
15.4.3 JMS的模块化设计
15.5 模块化的技术移植
15.6 小结

章节摘录

第一部分基础篇第1章 初识Eclipse RCP第2章 搭建Eclipse RCP开发环境第3章 SWT界面设计详解第4章 JFace界面设计详解第5章 Eclipse Forms与Nebula界面设计第1章 初识Eclipse RCP1.1 Eclipse平台概述Eclipse最初是由IBM公司开发的替代商业软件Visual Age for Java的下一代IDE开发环境，它于2001年11月贡献给开源社区，现在由非营利软件供应商联盟Eclipse基金会（Eclipse Foundation）管理。
2003年，Eclipse 3.0选择OSGi服务平台规范为运行时架构。
2007年6月，稳定版本3.3发布；2008年6月发布代号为Ganymede的3.4；2009年6月发布代号为Galileo的3.5；2010年6月发布代号为Helios的3.6；最近发布的Eclipse是代号为indigo的3.7。
Eclipse用户界面采用的技术是IBM公司开发的SWT，这是一种基于Java的窗口组件，类似Java本身提供的AWT和Swing窗口组件，不过IBM声称SWT比其他Java窗口组件更有效率。
Eclipse的用户界面还使用了GUI中间层JFace，从而简化了基于SWT的应用程序的构建。
Eclipse的插件机制是轻型软件组件化架构。
在富客户端平台上，Eclipse使用插件来提供所有的附加功能，例如支持Java以外的其他语言。
已有的分离插件能够支持C/C++（CDT）、PHP、Perl、Ruby、Python、telnet和数据库开发。
插件架构能够支持将任意的扩展加入到现有环境中，例如配置管理，而决不仅限于支持各种编程语言。
Eclipse的设计思想是：一切皆为插件。
Eclipse核心很小，Eclipse的所有功能都以插件的形式附加于Eclipse核心之上。
Eclipse的基本内核包括图形API（SWT/JFace）、Java开发环境插件（JDT）和插件开发环境（PDE）等。

媒体关注与评论

毫无疑问，Eclipse是全球最受欢迎的Java客户端软件开发IDE，Spring则在众多的Java Web开发框架中独树一帜。
随着现代软件开发的规模越来越大，结构越来越复杂，模块化是当今解决这一问题的最佳方案之一。Eclipse RCP和Spring OSGi分别是开发大规模的Java客户端应用和Java Web应用的主流技术，本书十分详尽地讲解了Eclipse RCP和Spring OSGi的技术细节，以及利用它们编写模块化程序的思路和最佳实践。
——51CTO 中国领先的IT技术网站

编辑推荐

《Eclipse RCP与Spring OSGi:技术详解与最佳实践》编辑推荐：资深的Eclipse专家亲自执笔，Eclipse官方技术社区推荐，权威性毋庸置疑。
内容全面，系统讲解了利用Eclipse RCP和Spring OSGi开发大规模Java应用的核心技术；实战性强，包含大量易于操作的案例和最佳实践。

第一图书网, tushu007.com

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:http://www.tushu007.com