

Java EE 企业级编程开发实例详解

袁梅宇 王海瑞 著

书问

BOOKASK.COM
清华大学出版社
北 京

内 容 简 介

Java EE 一直是企业级应用开发的主流技术标准和流行的软件开发架构, 最新的 Java EE 6.0 致力于简化复杂技术, 大大降低了开发难度, 使用户容易理解和上手。

本书带领读者由浅入深地学习 Java EE 开发的持久化层、业务逻辑层和 UI 用户表现层的知识, 系统地介绍称为 Java EE “三剑客” 的 JPA 2.0、EJB 3.1 和 JSF 2.0 开发。全书共 5 章, 主要内容包括 Java EE 开发基础、Java 持久化 API、EJB 3.1、JSF 2.0、Java EE 应用安全和综合示例。

本书适合软件开发人员作为技术参考书, 也适合作为计算机专业高年级本科生和研究生的教材或参考书。

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。

版权所有, 侵权必究。侵权举报电话: 010-62782989 13701121933

图书在版编目(CIP)数据

Java EE 企业级编程开发实例详解/袁梅宇, 王海瑞著. —北京: 清华大学出版社, 2013
ISBN 978-7-302-31491-2

I. ①J… II. ①袁… ②王… III. ①JAVA 语言—程序设计 IV. ①T312

中国版本图书馆 CIP 数据核字(2013)第 023859 号

责任编辑: 魏 莹 桑任松

装帧设计: 杨玉兰

责任校对: 周剑云

责任印制:

出版发行: 清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址: 北京清华大学学研大厦 A 座 邮 编: 100084

社 总 机: 010-62770175

邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

课件下载: <http://www.tup.com.cn>, 010-62791865

印 刷 者:

装 订 者:

经 销: 全国新华书店

开 本: 185mm×260mm

印 张: 27.5

字 数: 668 千字

版 次: 2013 年 3 月第 1 版

印 次: 2013 年 3 月第 1 次印刷

印 数: 1~4000

定 价: 49.00 元

产品编号:

BOOKASK.COM

前言

在当今发达的商业世界中，企业需要使用各种应用软件以帮助提升核心竞争力，而应用软件需要进行数据访问、实施业务逻辑、构筑增进用户体验的表示层，并与外部系统通信。公司需要在尽可能减少开支的同时，选择使用标准的、可靠的技术去处理日益加重的信息传输和信息处理的负荷。选择适合企业的工作平台的重要性日益突出。

Java EE(Java Enterprise Edition, Java 企业版)是一个开放的、基于标准的平台，用于开发、部署和管理面向 Web，以服务器为中心的企业级应用。

自 20 世纪 90 年代后，Java EE 一直是企业级应用开发的主流技术标准和流行的软件开发架构，并一直承受各种开源框架的竞争，每一次 Java EE 新版本的推出，都会给用户带来学习新知识的冲击。

长期以来，Java EE 一直被技术人员认为是难以理解、容易误用的“高雅”技术，其前身 J2EE 常常被业界视为重量级的技术(太过于强大而难以使用)——即使实现一个简单的 J2EE 程序，也需要大量的配置文件，尽管这些配置文件并不是必需的。

因为 J2EE 难以学习和掌握，使得很多开发人员转而寻求其他解决方案，各种以 SSH(Struts + Spring + Hibernate)为代表的轻量级架构大行其道。

吸取了轻量级架构的优点，Java EE 的“体重”在 2006 年 5 月正式发布 Java EE 5.0 规范时成功地“减肥”了。最新的 Java EE 6.0 则致力于简化复杂技术，大大降低了开发的难度，使用户容易理解和上手了。

Java EE 主要的优势在于它是开放的技术标准。使用 JPA、EJB、JMS、SOAP 及 RESTful Web 服务技术编写的应用软件容易在不同的应用服务器间移植。

Java EE 的另一优势是开源，大多数 Java EE 6 的参考实现都采用开源许可证，如 GlassFish、EclipseLink、Mojarra、OpenMQ、Metro 及 Jersey 等。

Java EE 并不仅仅是一门技术，而且还是一系列的技术标准和规范。本书探寻了 Java EE 新版本的革新技术，研究了一系列规范和如何将这些技术组织起来开发应用系统。

Java EE 6 包含近 30 项规范，内容极其庞杂，涵盖企业级开发的方方面面，主要包括 EJB 3.1、JPA 2.0 企业层、Servlet 3.0、JSF 2.0 Web 层、SOAP Web 服务、RESTful 服务的互操作规范。想完全掌握 Java EE 的每项技术，既耗时，也无必要。因此，本书主要涵盖称为开发“三剑客”的 JPA 2.0、EJB 3.1 和 JSF 2.0，分别用于构建企业 Web 应用的持久化层、业务逻辑层和表现层，深入浅出地叙述了各项技术的背景、要点和难点、注意事项，并且附上大量的典型示例供读者参考。本书既可以作为教学参考书，供计算机相关专业的本科生和研究生使用，也可以作为技术手册，在遇到类似的技术问题时查阅。

作为 Java EE 的爱好者和实践者，作者深知：如果没有对大量实例的学习和练习，想学好这些内容是不可能的。大量训练和模仿才是通向成功的正确道路。秉承这个思想，书中安排了大量的实例程序，每个实例都经过作者仔细挑选和构建，并经过认真调试，确认没有问题后才写入书中。



作者专门为本书设置了读者 QQ 群，群号是 245295017，欢迎读者加群，下载和探讨书中的实例，抒写读书心得，寻求技术难题解答等。

本书承蒙很多朋友、同事的帮助才得以成文。特别感谢同事尚振宏博士的热心帮助，他提出了不少意见和建议，对本书的改进帮助甚大；衷心感谢周兰江副教授和研究生何佳，难以忘记他们的真诚帮助；感谢朋友李海军博士、史博嘉先生，没有他们的帮助本书难以面世；感谢清华大学出版社的编辑老师在内容组织、排版以及出版方面提出的建设性意见和给予的无私帮助；感谢理解和支持我的家人，他们是我写作的坚强后盾。感谢购买本书的朋友，欢迎大家批评指正，所提出的问题将会受到重视，并在再版中改进。

编 者



BOOKASK.COM

目 录

第 1 章 Java EE 开发基础	1		
1.1 Java EE 6 简介	2		
1.1.1 学习 Java EE 的意义	2		
1.1.2 Java EE 6 版本介绍	2		
1.2 Java EE 6 的开发环境	4		
1.2.1 Java EE 6 开发环境的 安装和配置	4		
1.2.2 Eclipse 集成 GlassFish	12		
1.2.3 Eclipse 集成数据库	13		
1.2.4 EJB 容器数据源的配置	17		
1.3 Java EE 安全	21		
1.3.1 Java EE 基本安全概念	21		
1.3.2 使用 GlassFish 文件域	24		
1.3.3 基于表单的认证	30		
1.3.4 使用 GlassFish 证书域	33		
第 2 章 Java 持久化 API	39		
2.1 JPA 介绍	40		
2.1.1 持久化概念	40		
2.1.2 JPA 2.0 概述	41		
2.2 JPA 环境设置	43		
2.2.1 简单 JPA 程序的完整开发 过程	43		
2.2.2 使用 JUnit 测试 JPA	51		
2.2.3 persistence.xml 文件配置	55		
2.3 JPA 实体映射基础	57		
2.3.1 JPA 管理实体基本概念	57		
2.3.2 实体与数据库表的映射关系	59		
2.3.3 采用 XML 文件管理对象关系 映射	63		
2.3.4 定义复合主关键字	66		
2.3.5 定制属性的映射关系	70		
2.3.6 JPA 的两种访问类型	75		
2.4 JPA 实体关联映射	76		
2.4.1 实体关联基本概念	77		
2.4.2 一对一单向关联	78		
2.4.3 一对一双向关联	82		
2.4.4 一对多单向关联	84		
2.4.5 多对一单向关联	88		
2.4.6 多对一双向关联	90		
2.4.7 多对多双向关联	93		
2.4.8 有额外字段的多对多映射 关联表	96		
2.5 关联关系再讨论	101		
2.5.1 为关联关系选择集合类	101		
2.5.2 对关联实体进行排序	101		
2.5.3 延迟加载 vs 即时加载	104		
2.6 继承关系	106		
2.6.1 概述	106		
2.6.2 继承关系映射示例	106		
2.7 JPQL	112		
2.7.1 JPQL 语句基础	112		
2.7.2 JPQL 查询	116		
2.7.3 命名查询	120		
2.7.4 原生查询	122		
2.7.5 Criteria API	124		
2.8 JPA 高级话题	128		
2.8.1 处理并发	128		
2.8.2 实体的生命周期	132		
2.8.3 JPA 回调方法	142		
第 3 章 EJB 3.1	145		
3.1 EJB 3.1 介绍	146		
3.1.1 EJB 3.1 简介	146		
3.1.2 简单 EJB 的完整开发过程	149		
3.2 依赖注入和 JNDI	156		
3.2.1 基本概念	156		
3.2.2 JNDI 示例	159		



3.3 EJB 调试技术	162	4.4.2 托管 Bean 的范围	254
3.3.1 嵌入式技术调试 EJB	162	4.5 JSF 导航	257
3.3.2 Web 服务测试 EJB	163	4.5.1 页面导航	257
3.3.3 GlassFish 应用程序客户端	166	4.5.2 动作和动作监听器	260
3.4 会话 Bean	169	4.5.3 参数传递	263
3.4.1 无状态 EJB	169	4.6 JSF 转换器	268
3.4.2 有状态 EJB	172	4.6.1 简单的转换器	269
3.4.3 单体 EJB	176	4.6.2 数值转换器格式化输出	271
3.4.4 单体 EJB 初始化与并发	179	4.6.3 日期时间转换器	273
3.4.5 EJB 部署描述文件	182	4.6.4 用户定制的转换器	276
3.5 EJB 进阶	185	4.7 JSF 验证器	280
3.5.1 异步调用 EJB	185	4.7.1 标准的数据验证器	280
3.5.2 定时服务	188	4.7.2 定制验证器的错误信息	282
3.5.3 会话 Bean 的生命周期	193	4.7.3 用户定制的数据验证器	286
3.5.4 EJB 拦截器	198	4.8 JSF 表格	288
3.6 EJB 事务	204	4.8.1 h:dataTable 标签	288
3.6.1 EJB 事务处理	204	4.8.2 ui:repeat 标签	292
3.6.2 EJB 支持的事务	204	4.8.3 在表格中显示多种组件	294
3.6.3 容器管理的事务	206	4.8.4 编辑表格数据	296
3.6.4 Bean 管理的事务	212	4.8.5 添加删除表格数据	299
3.7 EJB 安全	214	4.8.6 表格数据排序	303
3.7.1 EJB 安全概念	214	4.8.7 表格分页显示	308
3.7.2 EJB 声明式安全策略	216	4.9 Ajax	314
3.7.3 EJB 编程式安全策略	220	4.9.1 使用简单的 Ajax	315
第 4 章 JSF 2.0	223	4.9.2 组件的动态联动	318
4.1 JSF 2.0 介绍	224	4.9.3 构建人性化 Web 应用	321
4.2 JSF 的运行环境	226	4.10 Facelets	325
4.2.1 JSF 运行环境的配置	226	4.10.1 Facelets 标签	325
4.2.2 第一个简单的 JSF 程序	228	4.10.2 使用 Facelets 模板	326
4.2.3 JSF 与可视化网页开发	234	4.11 JSF 进阶	328
4.2.4 配置 web.xml 文件	234	4.11.1 JSF 复合组件	328
4.3 JSF 标准组件	235	4.11.2 JavaScript 复合组件	332
4.3.1 JSF 标准组件示例	236	4.11.3 JSF 的生命周期	334
4.3.2 JSF 2.0 标签库	238	4.11.4 定制 JSF 安全	339
4.3.3 JSF 常用标准组件介绍	240	4.11.5 显示动态图像	345
4.4 托管 Bean	252	4.11.6 文件上传	348
4.4.1 JSF 托管 Bean	252	4.11.7 PrimeFaces 多文件上传	355
		4.11.8 在 JSF 中使用 EJB	358

4.12 国际化.....	361	5.6.3 排序和分页处理	399
4.12.1 国际化 JSF 项目	361	5.7 管理员使用的功能	402
4.12.2 页面语言功能选项	365	5.7.1 添加新用户等	402
第 5 章 综合示例	369	5.7.2 用户管理	404
5.1 系统功能概述.....	370	5.7.3 班级管理	410
5.2 数据库设计.....	370	5.7.4 课程管理	411
5.3 企业 Java Bean 层.....	378	5.8 教师使用的功能	412
5.4 用户认证与授权.....	384	5.8.1 录入成绩	412
5.5 JSF 模板.....	386	5.8.2 查询成绩	417
5.5.1 基本模板.....	386	5.9 学生使用的功能	418
5.5.2 管理员模板.....	387	附录 A 源代码使用说明	425
5.5.3 教师、学生模板	391	附录 B 章节与 Java EE 项目名称 对照表	426
5.6 公用功能.....	394	参考文献	429
5.6.1 登出.....	395		
5.6.2 修改密码.....	396		



BOOKASK.COM



BOOKASK.COM



第 1 章

Java EE 开发基础

本章介绍 Java EE 6 的版本及特色，以及 Java EE 6 开发环境安装和配置的各种常见问题，最后讨论 Java EE 的安全，并附有详细的实例来说明如何使用安全域进行验证和授权，以确保应用程序的安全。

BOOKASK.COM





1.1 Java EE 6 简介

了解一项技术平台是选择和使用这个平台的首要条件。只有在了解 Java EE 技术结构和特点之后,Java EE 爱好者才能根据自身需求,考虑学习和使用哪些技术。

1.1.1 学习 Java EE 的意义

假如程序员要处理对象集合,除非专门为了学习数据结构,显然不会从头来构建自己的链表、队列和堆栈,而是直接使用编程语言提供的集合 API。

同样道理,假如程序员要开发支持事务、安全、互操作和分布式的应用软件系统,也不会去重新开发全部的底层 API,而是直接使用 Java EE 提供的框架。

道理很简单——站在巨人的肩膀上看得更远。就像 Java SE(Java Standard Edition, Java 标准版)提供处理集合的 API 一样,Java EE 提供标准的方式去处理企业级应用,例如,提供 JTA(Java Transaction API, Java 事务 API)以处理事务,提供 JMS(Java Message Service, Java 消息服务)以处理消息,提供 JPA(Java Persistence API, Java 持久化 API)以处理持久化,等等。Java EE 就是专门为企业级应用开发提供的一组规范,可以视其为 Java SE 的扩展,有助于分布式、高可靠性、强大的、高可用性应用系统的开发。

Java EE 提供 Web 服务、组件模型、管理和通信 API 等一系列完整的解决方案,用于实现企业级的面向服务体系架构(Service Oriented Architecture, SOA)和构建 Web 2.0 应用程序。

自 20 世纪 90 年代开始,Java 就为用户广泛接受。Java 技术推动了 Web 的迅速发展。由于 Java 出现得较早,并且能持续地推出创新技术,因此被广泛应用于很多领域,拥有跨平台、动态 Web 以及分布式的计算能力,使学习和使用 Java 成为一种时尚。

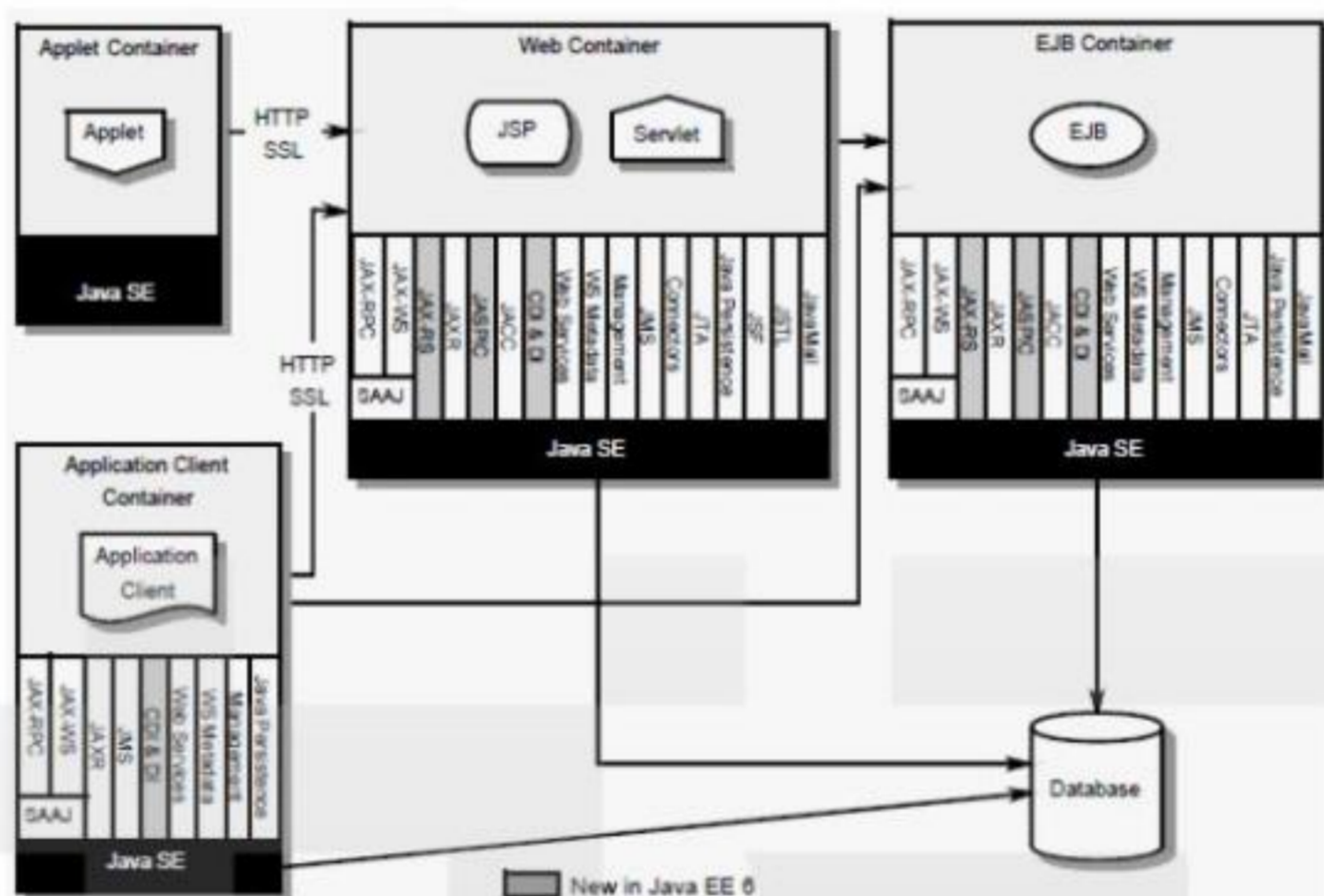
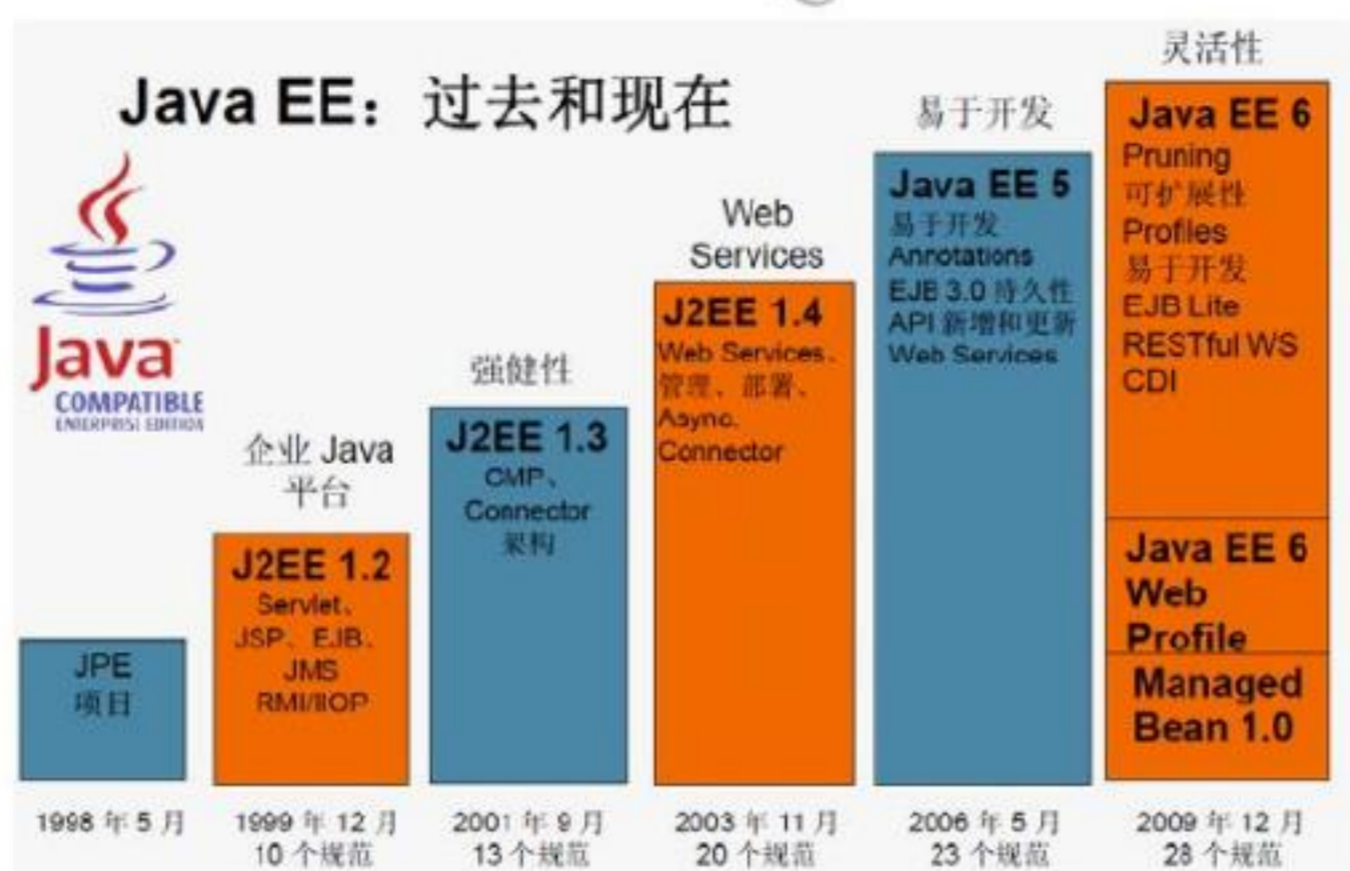
综上所述,学习和使用 Java EE,可以使开发人员能够紧跟 IT 技术的潮流,充分地享受先进技术带来的强大功能、高生产率和低复杂性,在未来的竞争中立于不败之地。

1.1.2 Java EE 6 版本介绍

Java EE 从版本 1.0 开始,最初的版本称为 JPE(Java Professional Edition, Java 专业版),于 1998 年 5 月发布,至今已经走过十多年了,经历了多次变更,每次都增加了很多新的规范,最新版是 Java EE 6,于 2009 年 12 月发布。各版本的发布时间以及规范概要如图 1.1 所示。

Java EE 6 包括 28 个规范和如图 1.2 所示的 4 种不同功能的容器(Container),涵盖了企业级开发技术的方方面面。在上一个版本基础上,Java EE 6 取得了不少进展,添加了大量的新技术,相对于以往的 Java EE 版本,Java EE 6 进一步简化了平台,扩展了可用性。

据 Oracle 公司提供的材料,Java EE 6 比 Java EE 5 更加简化了编程。在基于 POJO、JPA、REST 的简单用例中,与 Java EE 5 相比,Java EE 6 的 Java 类减少了 25%,代码行减少了 50%,XML 码行减少了 80%。





cron 风格的定时服务, 支持通过使用 `@Asynchronous` 标注来对会话 Bean 方法进行异步调用。EJB 3.1 的新特性还包括直接将 EJB 打包成 WAR 文件, 可以运行在嵌入式容器中, 便于在 Java SE 环境进行 JUnit 测试, 使用统一的标准化全局 JNDI 命名方式。

- ❑ JSF 2.0: JSF 2.0 正式将 Facelets 作为默认的视图技术, 为适应 Ajax 而改变了自身的生命周期, 使得 JSF 与 Ajax 组合更加自然。
 - ❑ Servlet 3.0: 提供异步处理方式, 大量使用标注, 简化了 XML 部署描述文件。
- 总之, Java EE 6 简化了平台的使用, 但功能更为强大。

1.2 Java EE 6 的开发环境

本节介绍 Java EE 6 开发环境的安装和配置, 如何在 Eclipse 中集成 GlassFish 和数据库, 以及 EJB 容器数据源的配置。

1.2.1 Java EE 6 开发环境的安装和配置

使用 Java EE 6 进行开发, 首要的任务是能够安装和配置开发环境。虽然 EJB 容器、集成开发环境和数据库系统单独的安装配置并不困难, 但三者的联合工作还需要一定的技巧, 有时一个小小的问题都会让开发人员困惑很长时间。

1. Java EE 开发环境的组成

使用 Java EE 进行开发, 离不开三大组成部分——EJB 容器、集成开发环境 IDE 和数据库系统。

(1) EJB 容器按是否收费, 分为收费版和免费版。收费版主要是各种商用应用服务器, 如 WebLogic、WebSphere、Oracle Application Server 等。收费版昂贵, 且不开源, 不适合初学者; 免费版主要有 JBoss AS 和 GlassFish, 两者都是开源的 Java EE 应用服务器, 前者目前市场占有率居第一位, 后者由 Oracle 旗下的 Sun 主推, 有很好的发展前景, 因此本书使用 GlassFish 3.1 作为 EJB 容器。注意, 广受欢迎的 Tomcat 和 Resin 并不是 EJB 容器, 只是一般的 Java Web 应用服务器, 只支持 JSP 和 Servlet 开发, 不支持 EJB 开发。

(2) 集成开发环境 IDE 也有多种选择, 尽管 Oracle 公司发布的 Java EE 6 示例程序都直接支持 NetBeans, 但由于历史原因, 多数用户更为钟爱 Eclipse, 因此本书选用 Eclipse 作为集成开发环境。

(3) 数据库系统已经非常成熟, 如果使用 JPA 2.0 作为持久化方案, 数据库系统间的差异大部分都已经被屏蔽, 使用何种数据库已经基本没有差别。本书选用用户喜爱的 MySQL 和 JDK 自带的 JavaDB 作为示例数据库。

2. 典型 Java EE 开发环境的安装配置

(1) EJB 容器的安装配置

首先到 Oracle 官方网站下载新版 Java EE 6 SDK, 本书下载的是 Java EE 6 SDK Update

3 with JDK 7, 网址是 <http://www.oracle.com/technetwork/java/javaee/downloads/index.html>。注意不要选择 Web Profile 版本, 要选择完整版。

根据操作系统, 选择要下载的文件, 本书假定使用 Windows 操作系统, 所做的选择如图 1.3 所示。

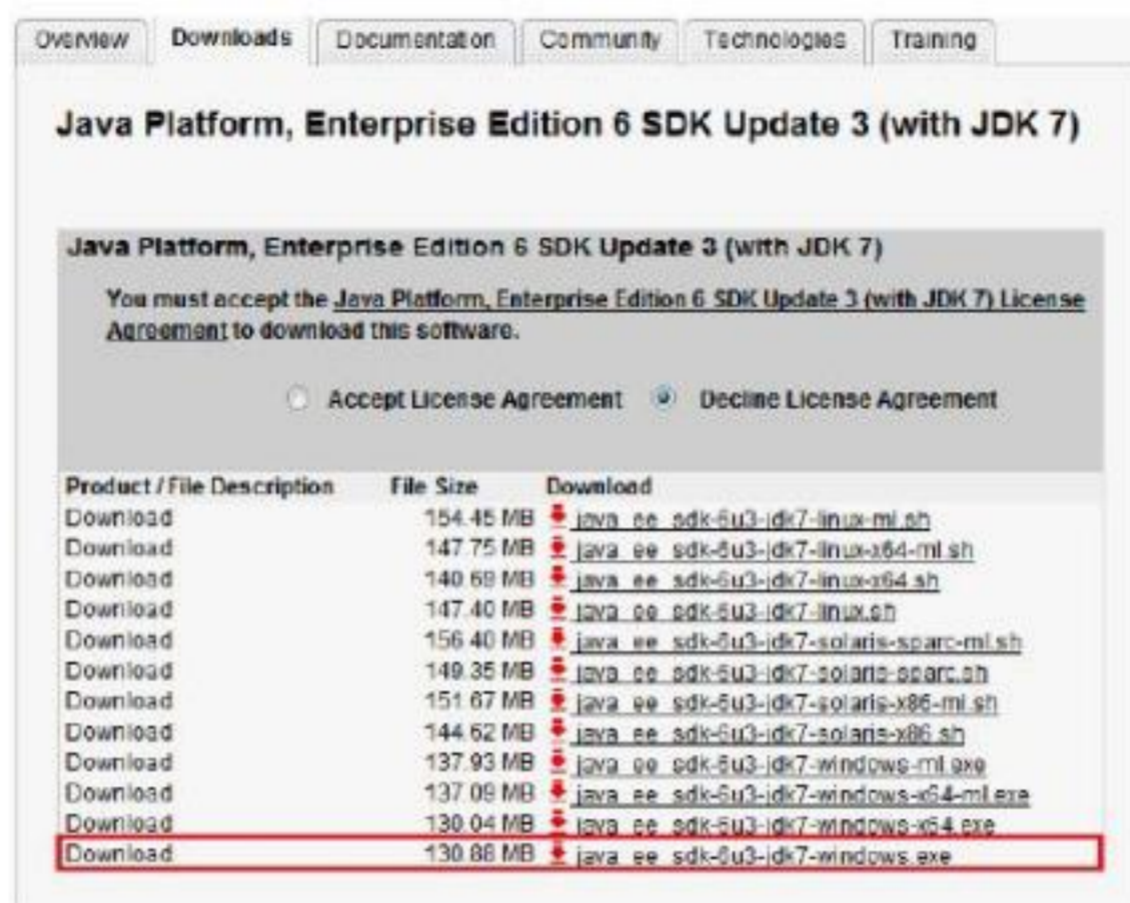


图 1.3 下载 Java EE 6 SDK

运行下载的安装文件, 一步一步地按照提示做, 就能顺利安装, 由于较为简单, 就不再赘述。安装向导如图 1.4 所示。



图 1.4 安装向导

注意, 完整版本的 Java EE 6 SDK 附带 JavaDB、JDK 7 等工具, 不用另外专门安装。

(2) Eclipse 安装配置

到 Eclipse 官方网站(<http://www.eclipse.org/downloads/>)下载最新的 Eclipse IDE for Java EE Developers, 这里下载的是 eclipse-jee-indigo-win32.zip 文件, 共 212MB。下载后, 用解



压缩软件释放到读者想要的文件夹中，例如，解压缩到“C:\jee6learning\eclipse”文件夹中。然后双击 eclipse.exe 文件启动，如图 1.5 所示。

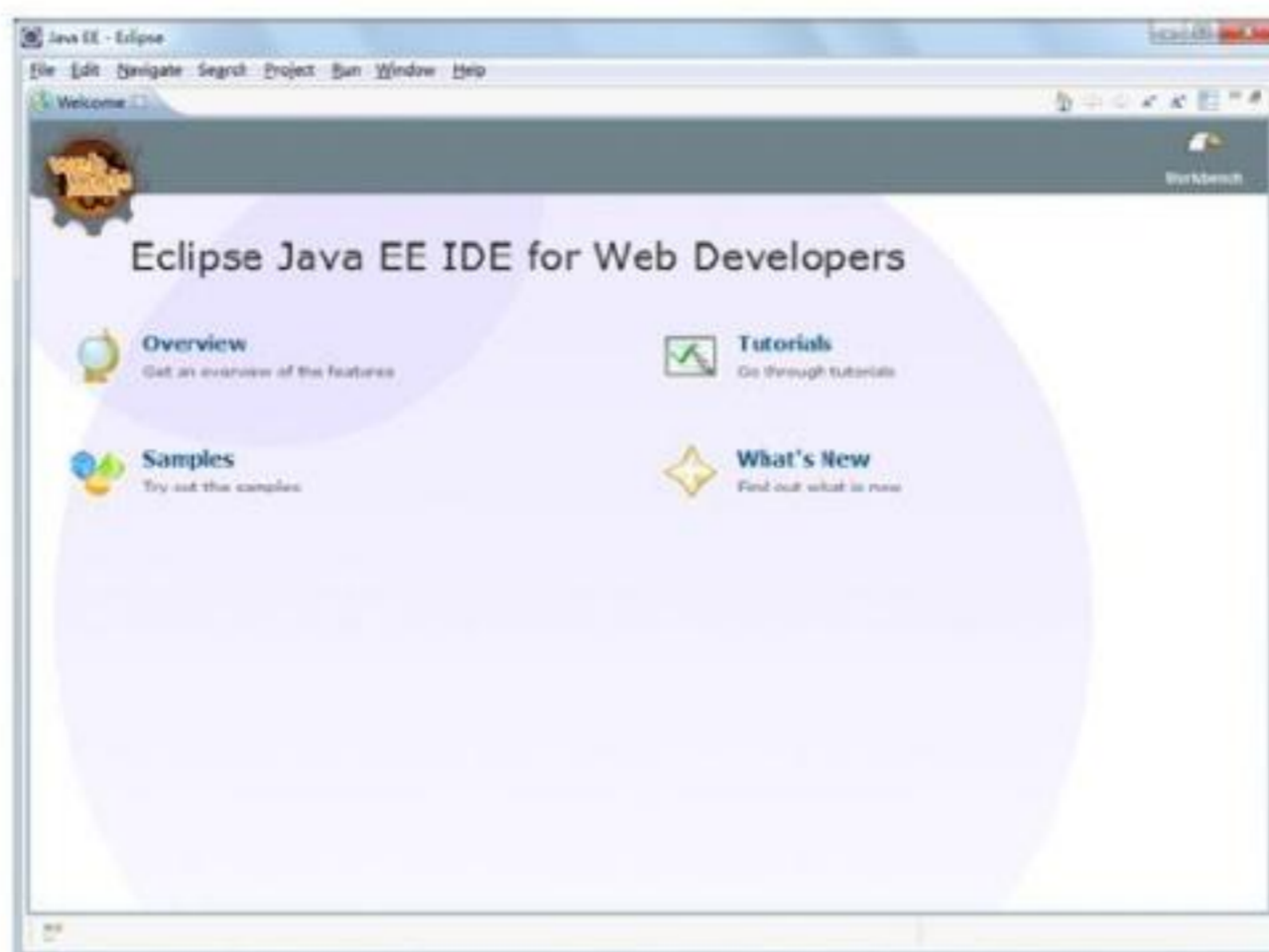
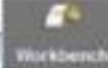


图 1.5 启动 Eclipse

单击欢迎页面右上角的  进入 Eclipse 工作台。

Eclipse 默认没有配置对 GlassFish 服务器的支持，只能手动进行配置。注意确保计算机已连上 Internet。首先在 Eclipse IDE 中，选择 File→New→Other，在弹出的向导窗口中选择 Server→Server，然后单击 Next 按钮，如图 1.6 所示。在出现的 New Server 界面中，单击 Download additional server adapters 链接，下载额外的服务器适配器，如图 1.7 所示。

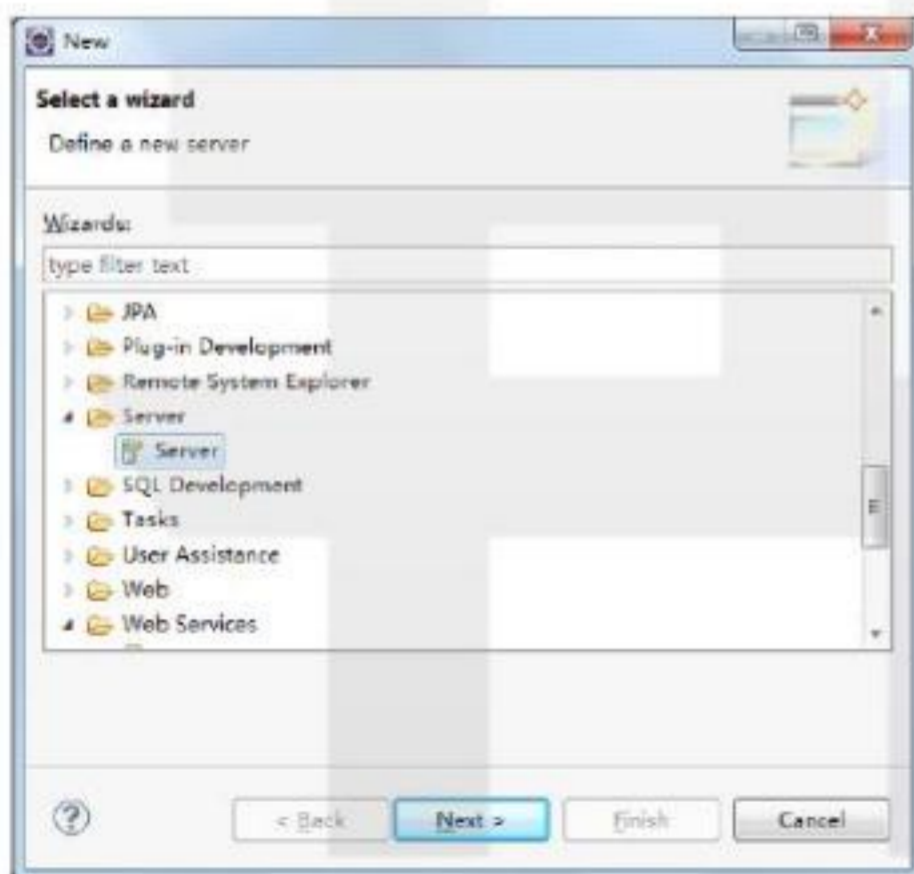


图 1.6 安装 GlassFish 服务器支持工具

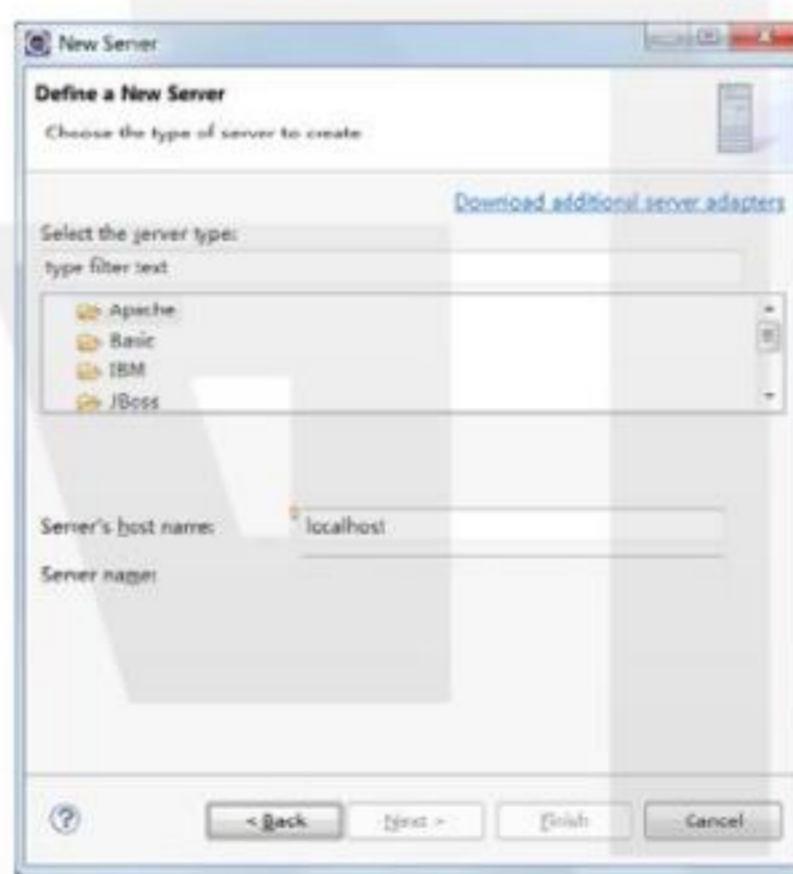


图 1.7 下载额外的服务器适配器

选择欲安装的服务器扩展，即 Oracle GlassFish Server Tools，如图 1.8 所示。然后，选中第一个单选按钮，同意安装服务器的许可协议，如图 1.9 所示。

在出现的界面中单击 OK 按钮确认安装服务器扩展，如图 1.10 所示。然后，Eclipse 开

始从网络下载并进行安装，如图 1.11 所示。



图 1.8 选择欲安装的服务器扩展



图 1.9 同意许可协议

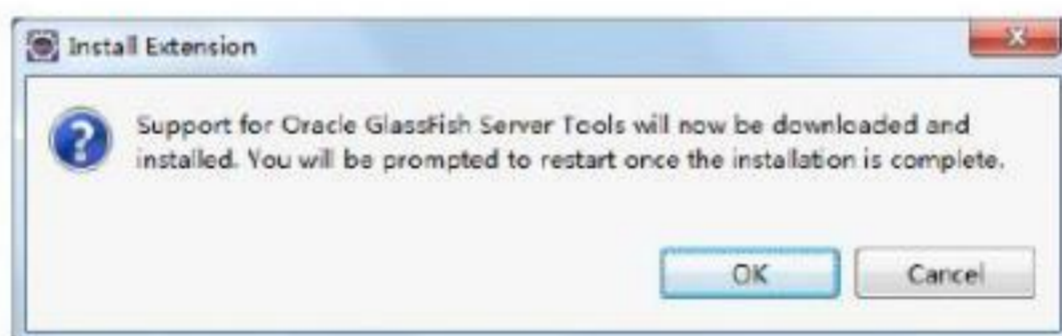


图 1.10 确认安装

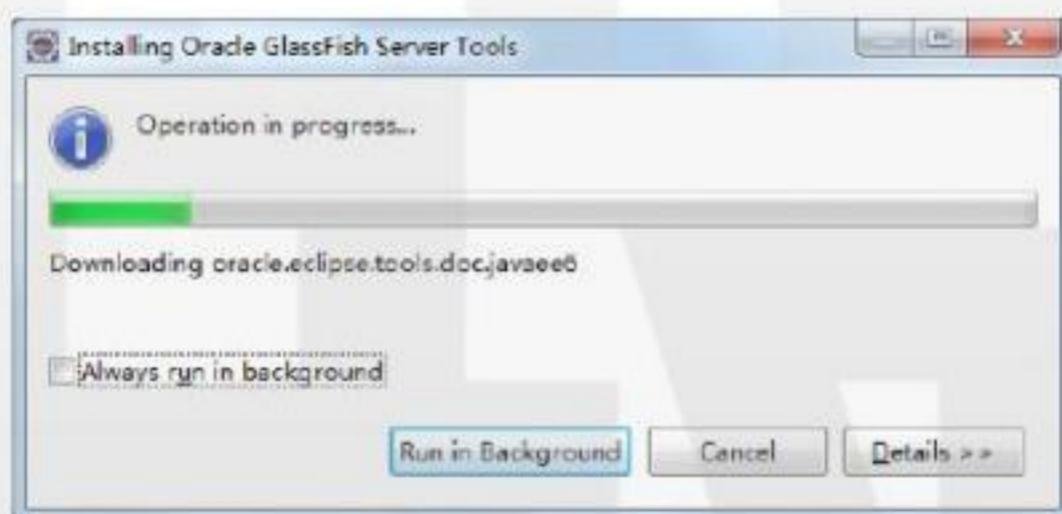


图 1.11 进行安装

安装完毕后，提示需要重新启动 Eclipse 以使安装生效，单击 Restart Now 按钮重新启动，如图 1.12 所示。

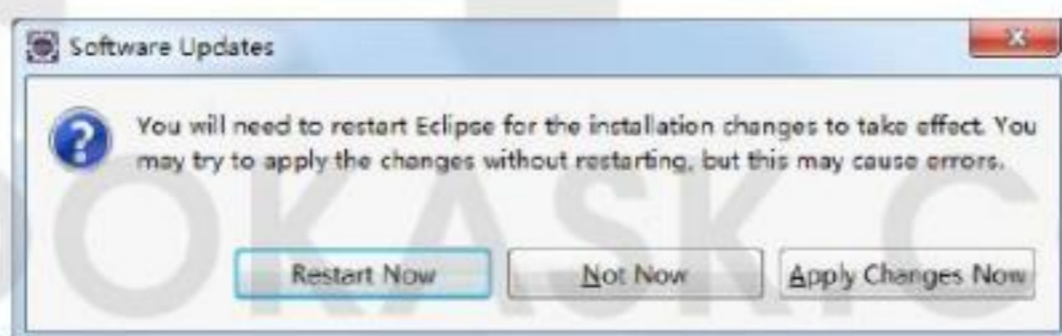


图 1.12 安装完毕重新启动



重新启动后, Eclipse 欢迎页面显示 GlassFish 应用服务器扩展的帮助信息, 如图 1.13 所示。单击对应的超链接, 可以查看 GlassFish 服务器扩展说明文档, 如图 1.14 所示。

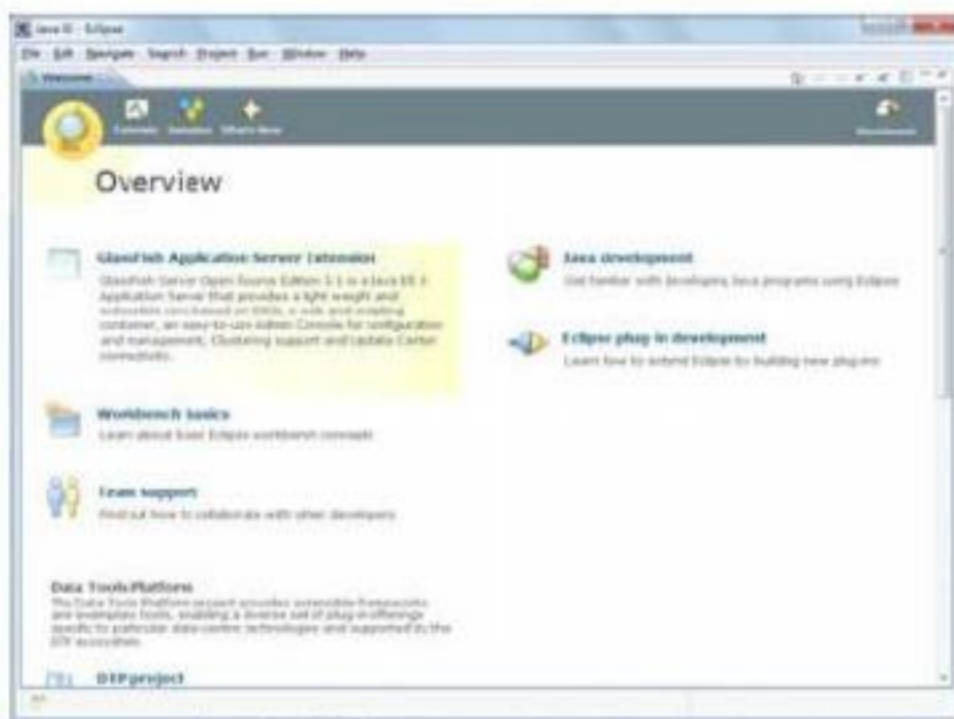


图 1.13 已安装的 GlassFish 服务器扩展



图 1.14 GlassFish 服务器扩展说明

关闭欢迎页面。至此, 已经完成 GlassFish 应用服务器扩展的安装。

(3) 安装配置 MySQL

MySQL 数据库是应用最广泛的开源数据库, 具有执行性能高、运行速度快、容易使用等特点, 其官方主页地址为 <http://www.mysql.com/>, 产品分为 MySQL Community Server (MySQL 社区版) 和 MySQL Enterprise (MySQL 企业版) 两类, 前者免费开源, 后者用于商业, 需要付费。

在编写本书时下载的 `mysql-5.5.15-win32.msi` 为 Windows 环境下的完全安装版本。下载后双击该文件进行安装, 在如图 1.15 所示的安装向导中, 单击 Next 按钮进入下一步。然后, 单击 Typical 按钮选择典型安装, 如图 1.16 所示。



图 1.15 MySQL 安装向导



图 1.16 选择典型安装

选择详细配置, 即选中 Detailed Configuration 单选按钮, 如图 1.17 所示, 单击 Next 按钮进入下一步。

选择开发者模式, 即选中 Developer Machine 单选按钮, 如图 1.18 所示, 单击 Next 按钮进入下一步。



图 1.17 选择详细配置



图 1.18 选择开发者模式

选择多功能数据库，即选中 Multifunctional Database 单选按钮，如图 1.19 所示，然后单击 Next 按钮进入下一步。然后，选择表空间存储位置，默认使用 C 盘存储表空间，保持默认设置不变，如图 1.20 所示，单击 Next 按钮。



图 1.19 选择多功能数据库



图 1.20 选择表空间存储位置

下一步选择“决策支持/OLAP”，即选中“Decision Support(DSS)/OLAP”单选按钮，如图 1.21 所示，然后单击 Next 按钮。设置网络端口号可以使用默认配置，只要不与本机端口号冲突，可以使用默认端口 3306，如图 1.22 所示，单击 Next 按钮继续。

下一步设置多语言字符集，MySQL 默认的字符集只适合英文和西欧语言，使用中文时需要设置多语言字符集，一般选用 UTF8 以支持多语言，确保选中 Best Support For Multilingualism 单选按钮，如图 1.23 所示，单击 Next 按钮继续。选中 Install As Windows Service 复选框，将 MySQL 安装为 Windows 服务；如果想自动加载 MySQL，应选中 Launch the MySQL Server automatically 复选框；默认的服务名为 MySQL，如果想改变，也在这一步进行设置，如图 1.24 所示，单击 Next 按钮继续。

下一步设置数据库管理员密码，确保两次输入的管理员密码是完全一致的，如图 1.25 所示，然后单击 Next 按钮。最后一步是执行数据库实例配置，等待四个配置步骤全部完成后，会出现四个蓝色的勾，单击 Finish 按钮结束 MySQL 的安装，如图 1.26 所示。

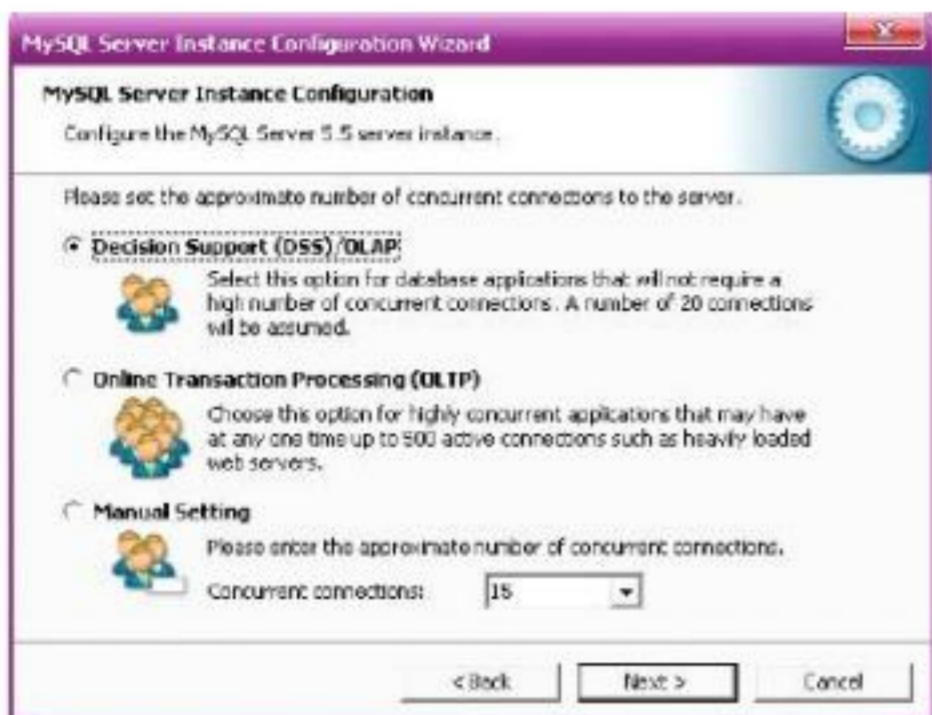


图 1.21 选择“决策支持/OLAP”

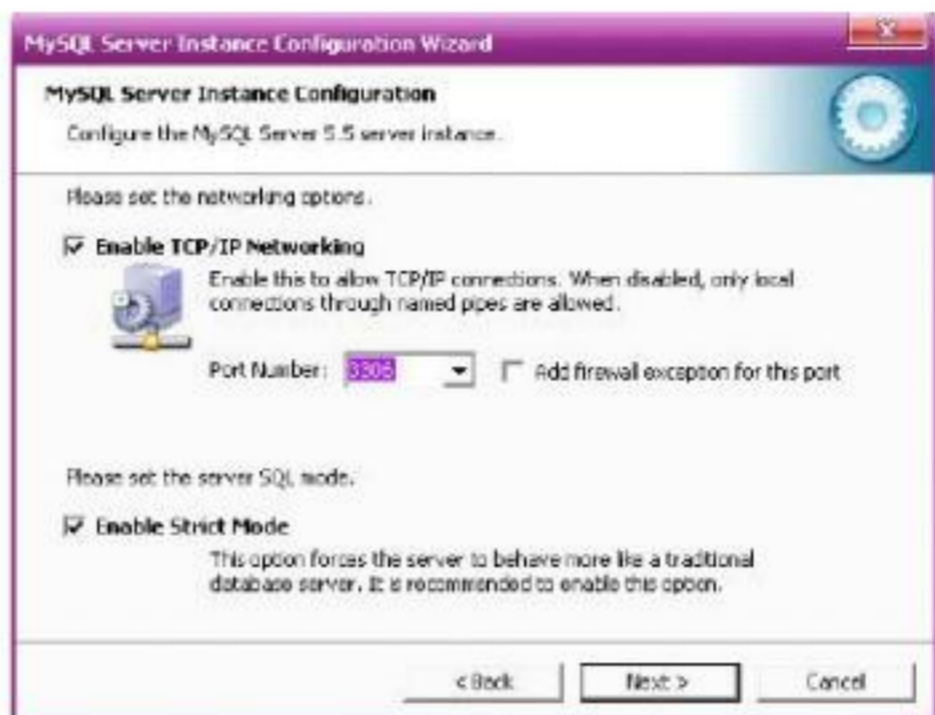


图 1.22 设置网络端口号

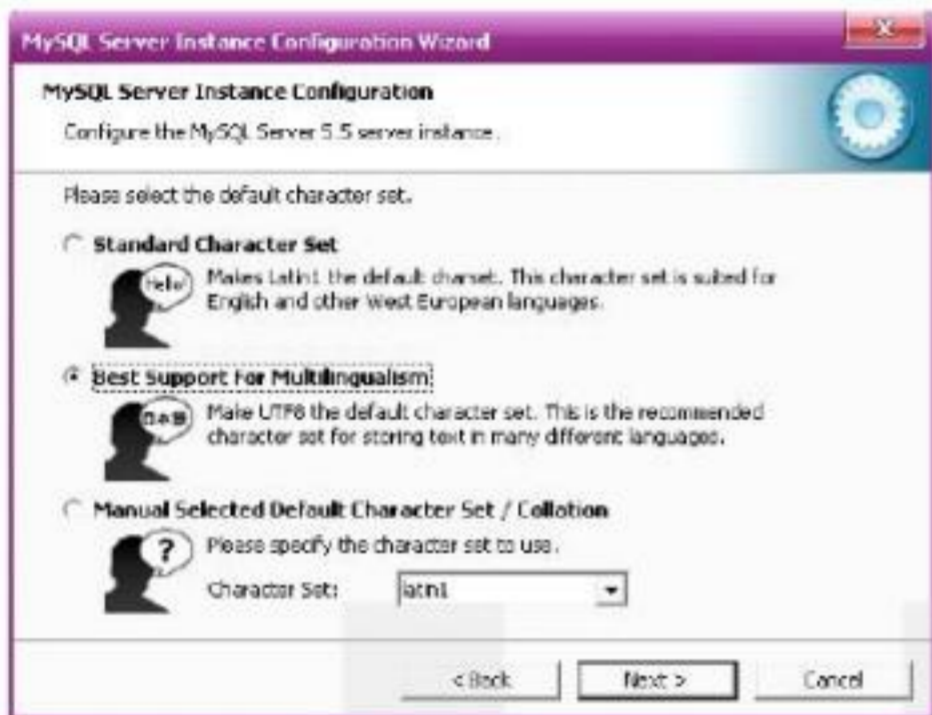


图 1.23 设置多语言字符集



图 1.24 设置服务名



图 1.25 设置数据库管理员密码



图 1.26 执行数据库实例配置

由于安装好的 MySQL 仅提供称为“MySQL 5.5 命令行客户端”的命令行窗口，不方便使用，如图 1.27 所示，为此，这里下载使用 Navicat for MySQL 工具，以方便查询、修改数据库中的数据，还提供导入、导出数据以及转储 SQL 文件等功能，可以轻松地管理数

数据库，如图 1.28 所示。

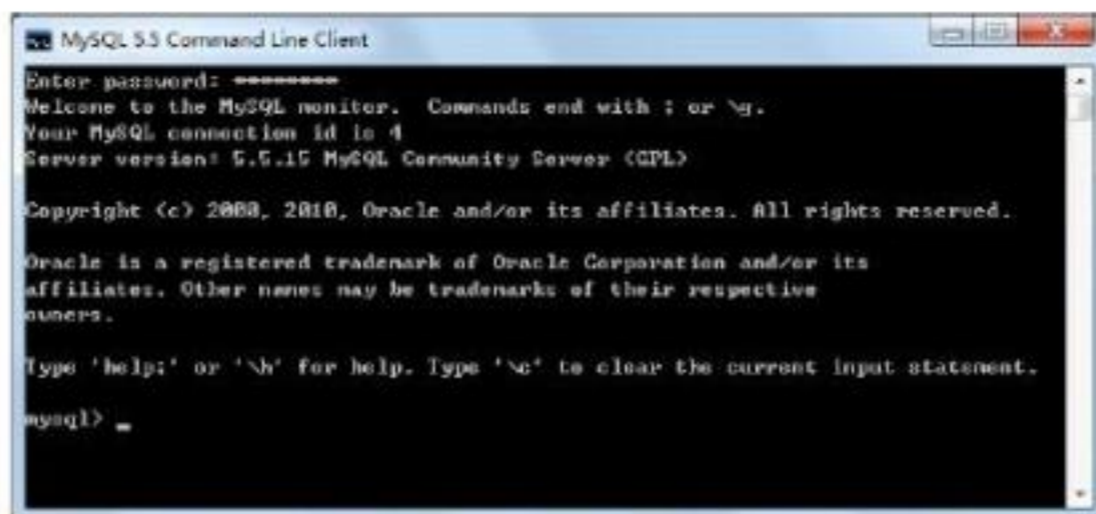


图 1.27 不方便使用的命令行窗口

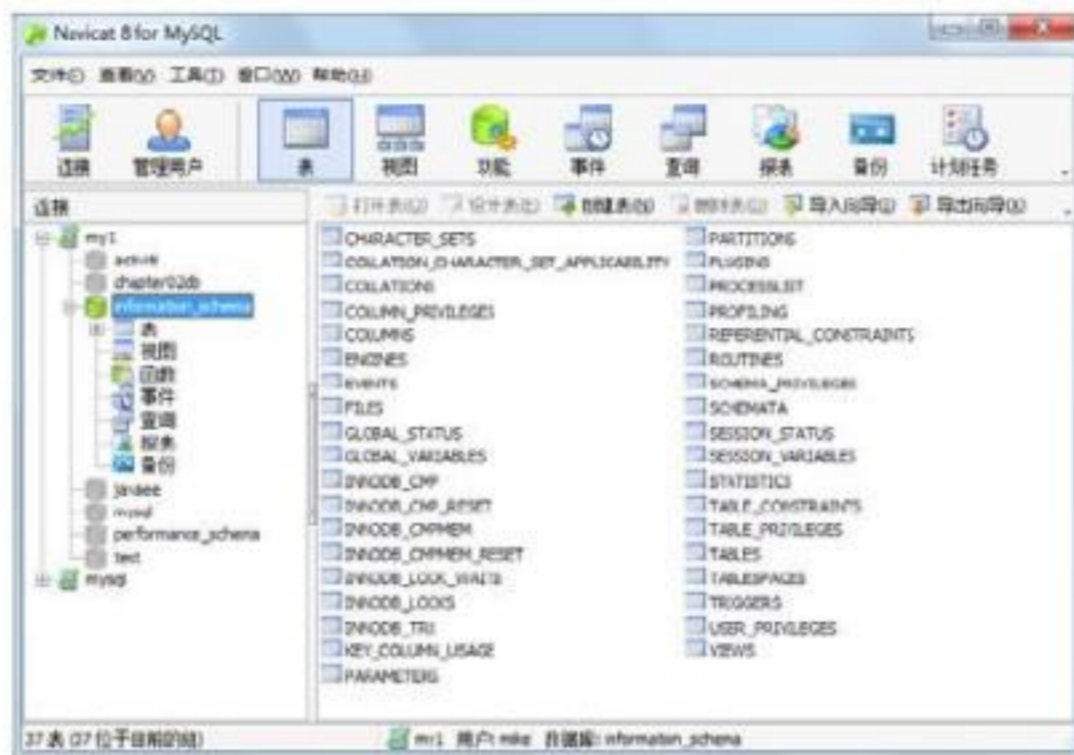


图 1.28 图形化的 MySQL 客户端

3. GlassFish 和 MySQL 的启动和停止

(1) GlassFish 的启动和停止

打开 Windows 资源管理器，在 GlassFish 的安装目录下，找到 bin 目录，位置类似于本书的 C:\glassfish3\glassfish\bin。

该目录中有两个批处理文件：startserv.bat 和 stopserv.bat，用鼠标双击前者，启动 GlassFish，同时打开一个命令行窗口，如图 1.29 所示；双击 stopserv.bat 停止 GlassFish。

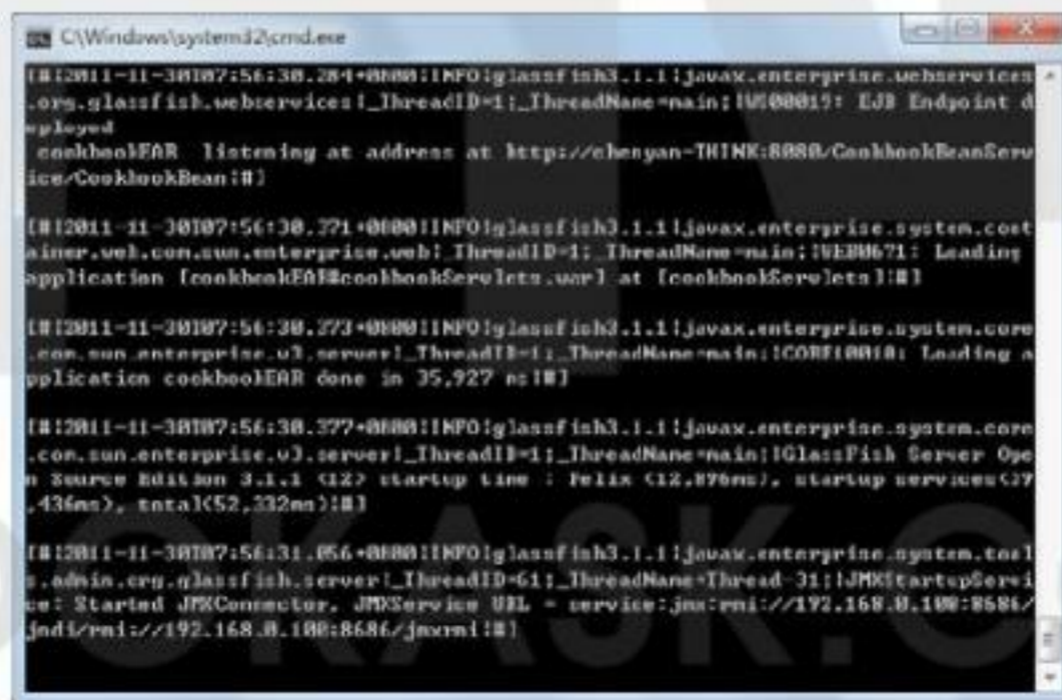
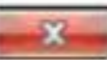




图 1.29 GlassFish 的命令行窗口

注意如果要关闭 GlassFish 的命令行窗口，一定要使用 stopserv.bat 来关闭，不能随便单击命令行窗口的 ，否则可能造成数据丢失或数据库崩溃等不可预料后果。

(2) MySQL 的启动和停止

如果安装时将 MySQL 配置为 Windows 服务，就可以使用“计算机管理”工具启动或停止 MySQL 服务。本书将 MySQL 服务的启动类型设置为“手动”，以获得更快的操作系统启动速度。


打开“计算机管理”窗口，如图 1.30 所示，选中 MySQL 服务，工具栏  的四个图标分别表示启动服务、停止服务、暂停服务和重新启动服务，单击相应按钮可以操纵 MySQL 服务。



图 1.30 启动或停止 MySQL

1.2.2 Eclipse 集成 GlassFish

在 Eclipse 中集成 GlassFish，可以直接在 IDE 中启动或停止 GlassFish，发布或取消发布 Java EE 程序，方便开发。

1. 集成 GlassFish

并不是所有的 Eclipse 透视图都支持 Web 容器或 EJB 容器，最好在 Java EE 透视图下集成 GlassFish。步骤是，通过 Window → Open perspective 菜单命令打开 Java EE 透视图，在 IDE 的右下角切换到 Servers 标签页，如图 1.31 所示。

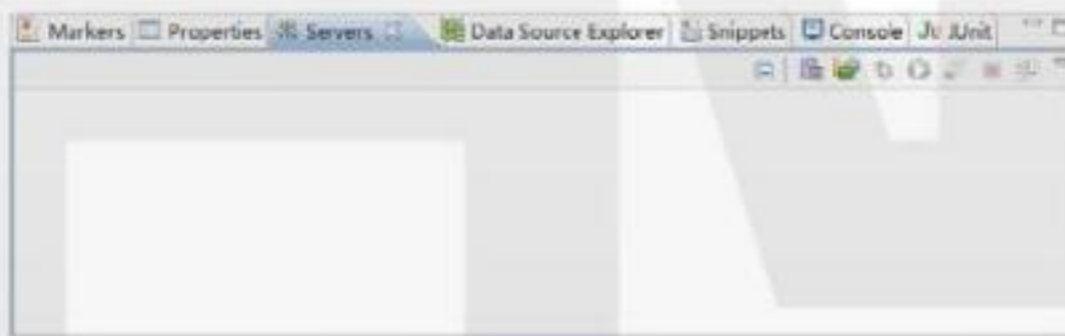


图 1.31 Servers 标签页

在空白处单击鼠标右键，从弹出的快捷菜单中选择 New→Server 命令，在弹出的对话框中选择服务器类型“GlassFish 3.1”，然后单击 Next 按钮继续配置，如图 1.32 所示。如果找不到想要设置的服务器类型，那就是还没有配置过服务器适配器，可以单击 Download additional server adapters 超链接进行配置。

下一步是配置用户密码。GlassFish 的初始管理员名称为 admin，密码为空。如果设置

了 GlassFish 的其他管理员或者修改了初始密码，可以在这一步进行设置。完成后单击 Finish 按钮结束设置，如图 1.33 所示。



图 1.32 选择服务器类型



图 1.33 配置用户密码

2. 操纵 GlassFish 服务器

上一步完成后，如果在 Servers 标签页中已经看到玻璃鱼图标，表明 GlassFish 已经与 Eclipse 成功集成，如图 1.34 所示。

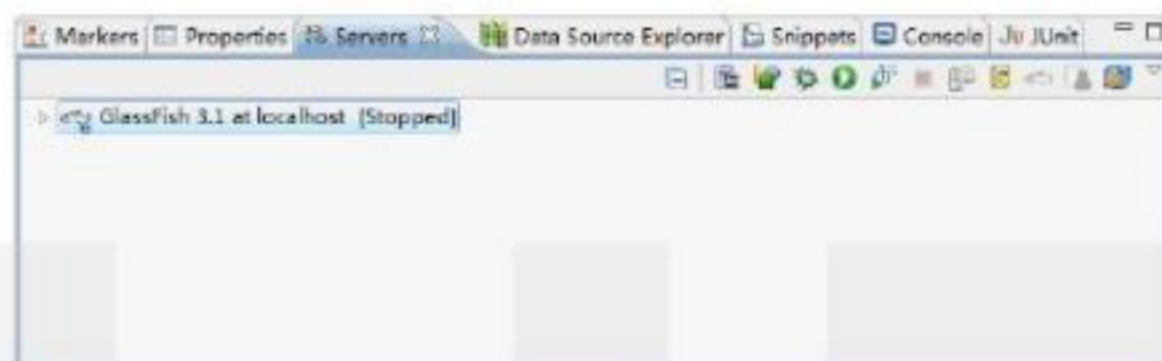


图 1.34 集成了 GlassFish

在玻璃鱼图标上用鼠标右击，在弹出的快捷菜单中选择相关的命令，就可以对服务器进行操纵。快捷菜单中有若干命令选项，常用的功能如表 1.1 所示。

表 1.1 快捷菜单的常用命令选项

菜单命令选项	功 能
Start	启动服务器
Debug	以 Debug 方式启动服务器
Stop	停止服务器
Add and Remove...	添加或删除项目
Publish	发布
Clean...	清除所有已发布资源，重新发布



1.2.3 Eclipse 集成数据库

Eclipse 为数据库开发定制了数据库开发透视图，开发人员直接使用 Eclipse IDE 就可以通过 SQL 语句操纵数据库对象，使用起来十分方便。



1. 配置数据库驱动

Eclipse 支持大多数关系数据库，但用户必须提供数据库驱动程序。

首先，选择 Window→Preferences 菜单命令，在弹出的 Preferences 窗口中，从左边栏导航至 Data Management → Connectivity → Driver Definitions，可以在右上部的 Filter 提示的下拉列表框中选择 MySQL 以过滤掉不用的数据库驱动，这时应该还没有定义任何 MySQL 数据库驱动，如图 1.35 所示。

然后单击 Add 按钮，在弹出的 New Driver Definition 对话框中，这时的默认选项卡应该是 Name/Type，在中上部的 Vendor Filter 提示的下拉列表框中选择 MySQL 以过滤掉其他数据库驱动提供商，再选择版本 5.1，这时，应该出现一个错误提示，提示 Eclipse 找不到“mysql-connector-java-5.1.0-bin.jar”驱动文件，如图 1.36 所示。

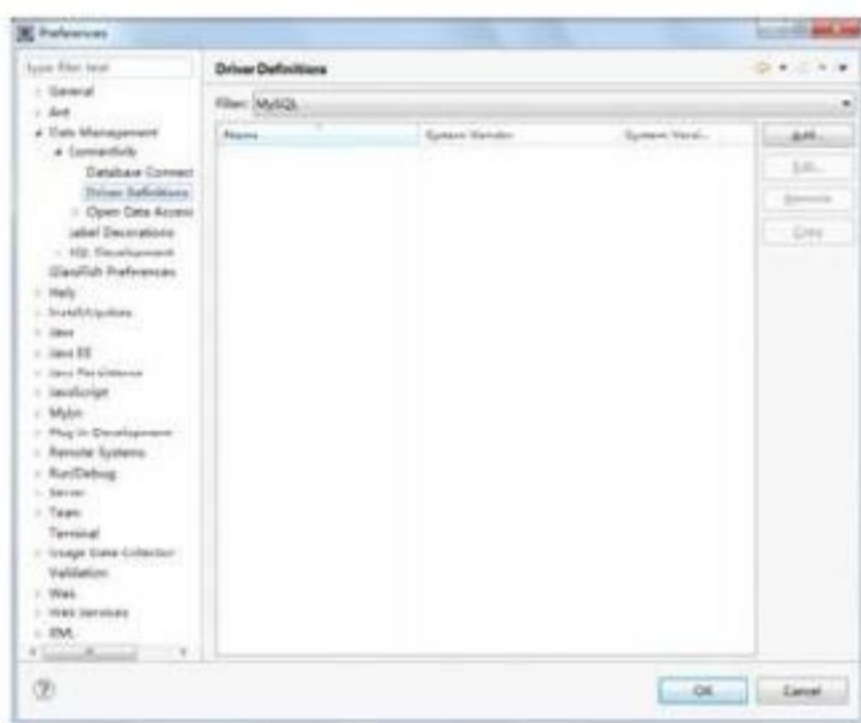


图 1.35 定义数据库驱动

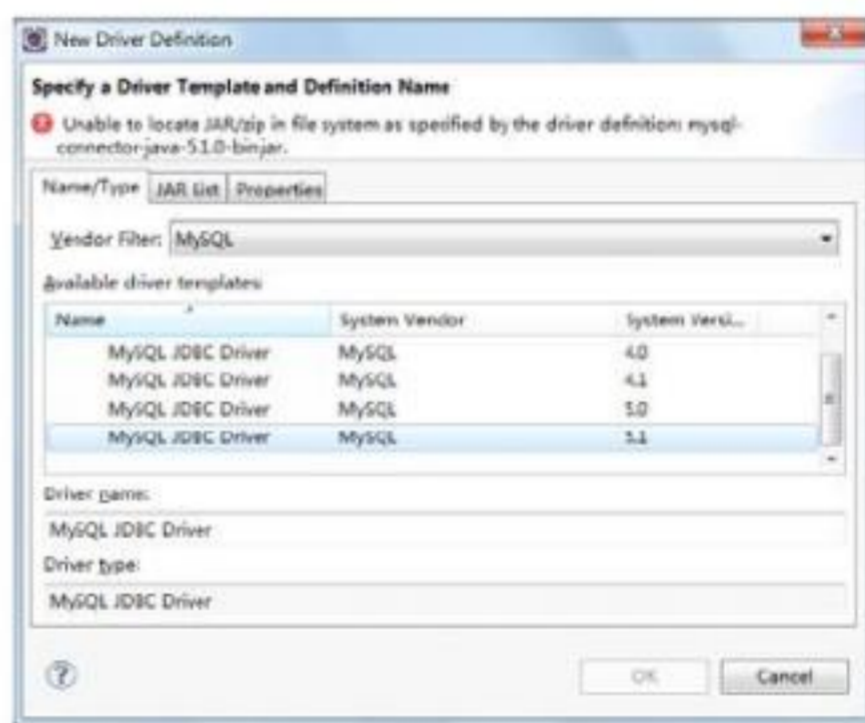


图 1.36 添加数据库驱动

单击 JAR List 标签切换选项卡，使用 Remove JAR/Zip 按钮删除找不到的驱动，如图 1.37 所示。然后使用 Add JAR/Zip 按钮添加数据库驱动，如图 1.38 所示。

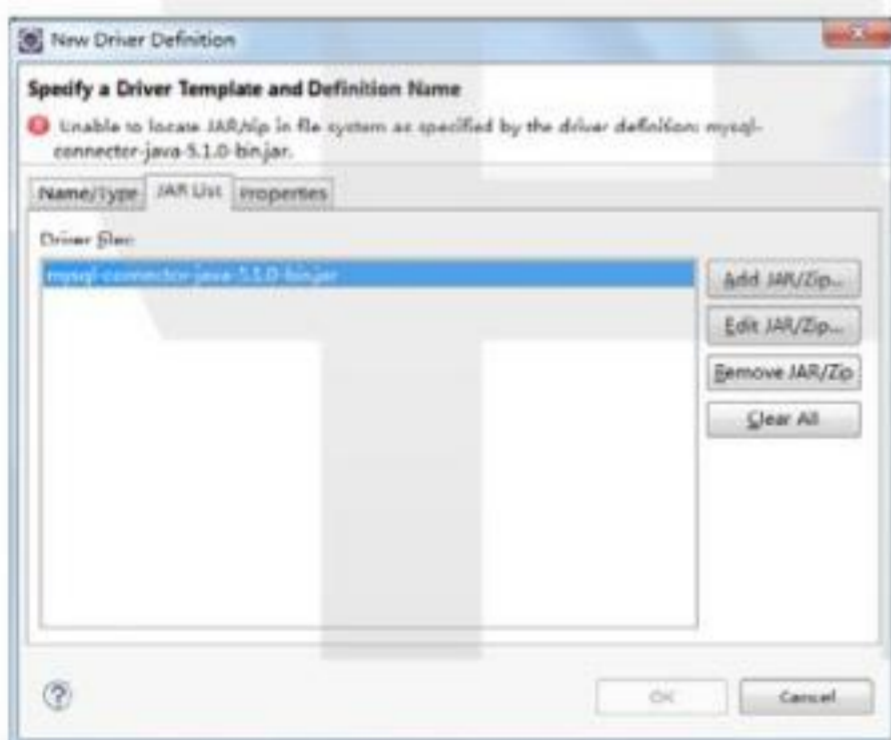


图 1.37 删除找不到的驱动

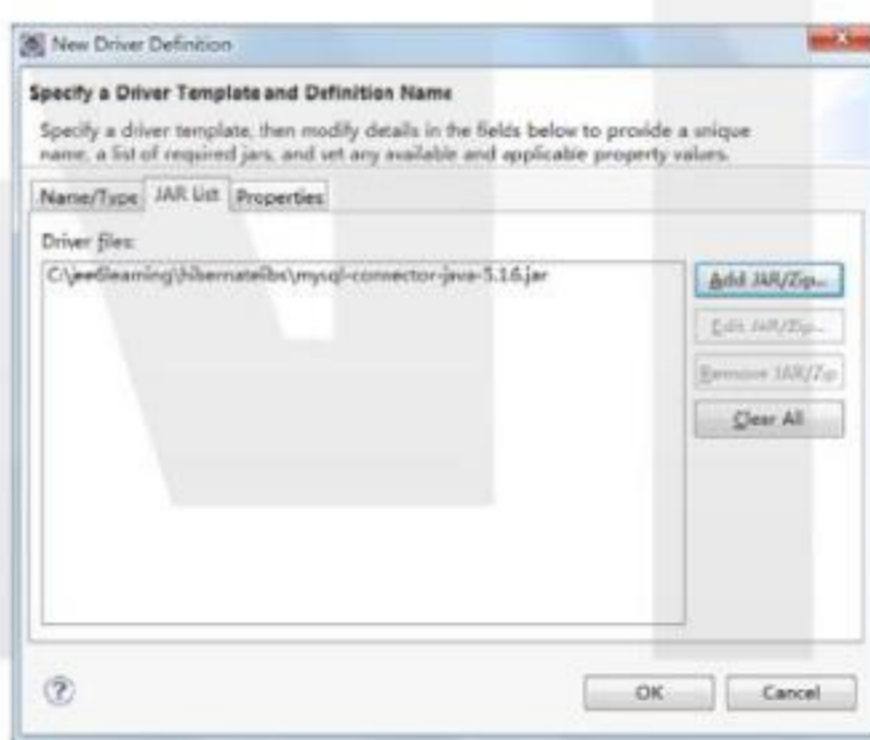


图 1.38 添加驱动

最后，单击 OK 按钮确认添加，现在可以看到已经添加了驱动，如图 1.39 所示。再次单击 OK 按钮关闭 Preferences 窗口。

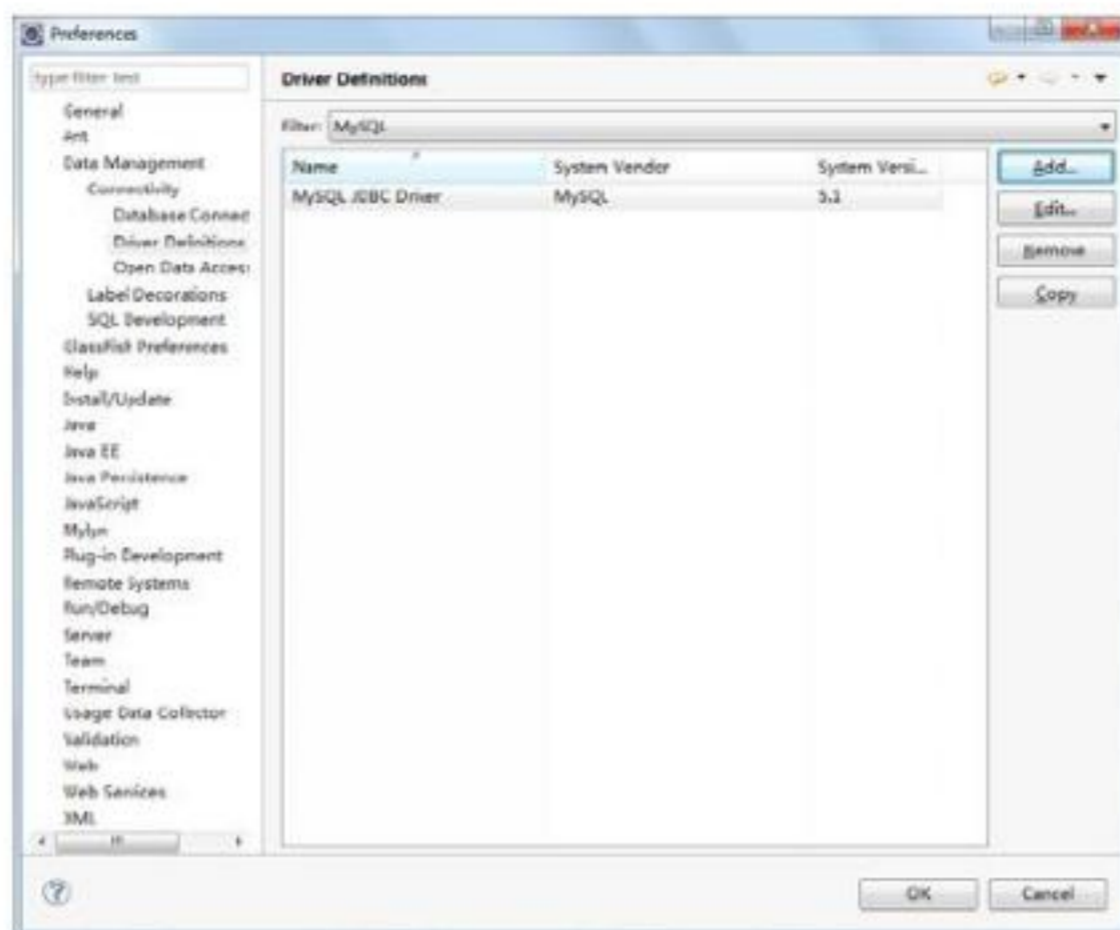


图 1.39 已经添加了驱动

2. 连接数据库

从菜单栏中选择 Window → Open Perspective → Other → Database Development, 切换到数据库开发透视图(见图 1.40)。

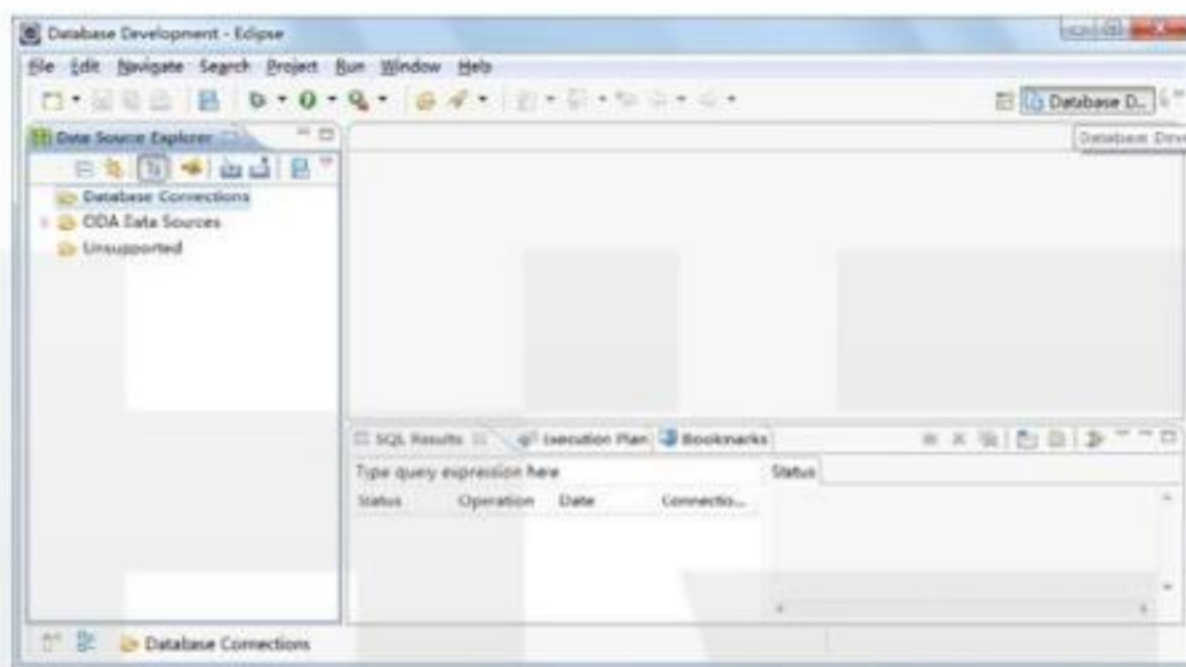


图 1.40 数据库开发透视图

在左边的 Data Source Explorer 栏中, 用鼠标右击 Database Connections 项, 从弹出的快捷菜单中选择 New 命令, 打开 New Connection Profile 窗口。可以在 Connection Profile Types 下用键盘输入“my”, 然后按 Enter 键, 以快速定位所需数据库类型。选择 MySQL 数据库, 在 Name 提示下可以看到 Eclipse 已经为数据库连接取了一个名字, 用户可以自行修改名字及描述, 如图 1.41 所示。

单击 Next 按钮进入下一步。根据读者自身的数据库设置, 填写欲连接的数据库名、URL、用户名、密码等连接数据库所需的信息, 如图 1.42 所示。

填写完毕后, 最好测试一下是否填写正确, 单击 Test Connection 按钮, 如果一切正常, 应该出现如图 1.43 所示的 Success 对话框。如果不正确, 应检查数据库是否已正常启动, 数据库连接的细节信息是否正确。



图 1.41 选择数据库类型



图 1.42 填写数据库连接细节



图 1.43 数据库连接正常

3. 测试

经过上面的步骤，在 Database Connections 下已经有了一个名为“New MySQL”的数据库连接，展开后得到如图 1.44 所示的数据库 Schemas。

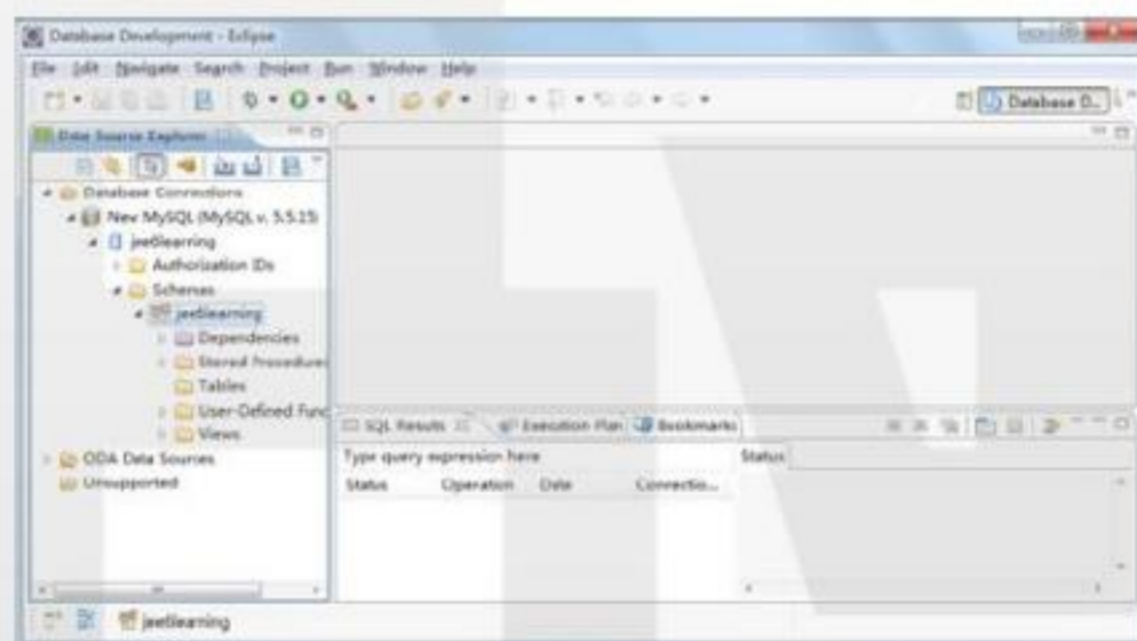


图 1.44 数据库的 Schemas

现在添加数据库表进行测试。选择 New MySQL 并以鼠标右击，从弹出的快捷菜单中选择 Open SQL Scrapbook 菜单命令，在 Connection profile 区域选择数据库，确保连接数据库的正确性。如图 1.45 所示，输入建立测试表的 SQL 语句，在窗口中用鼠标右击，从弹出的快捷菜单中选择 Execute All 命令来执行 SQL 语句，刷新数据库后，可以看到数据库表已经建立。读者可自行进行数据库表的插入、删除、修改等其他操作的测试。

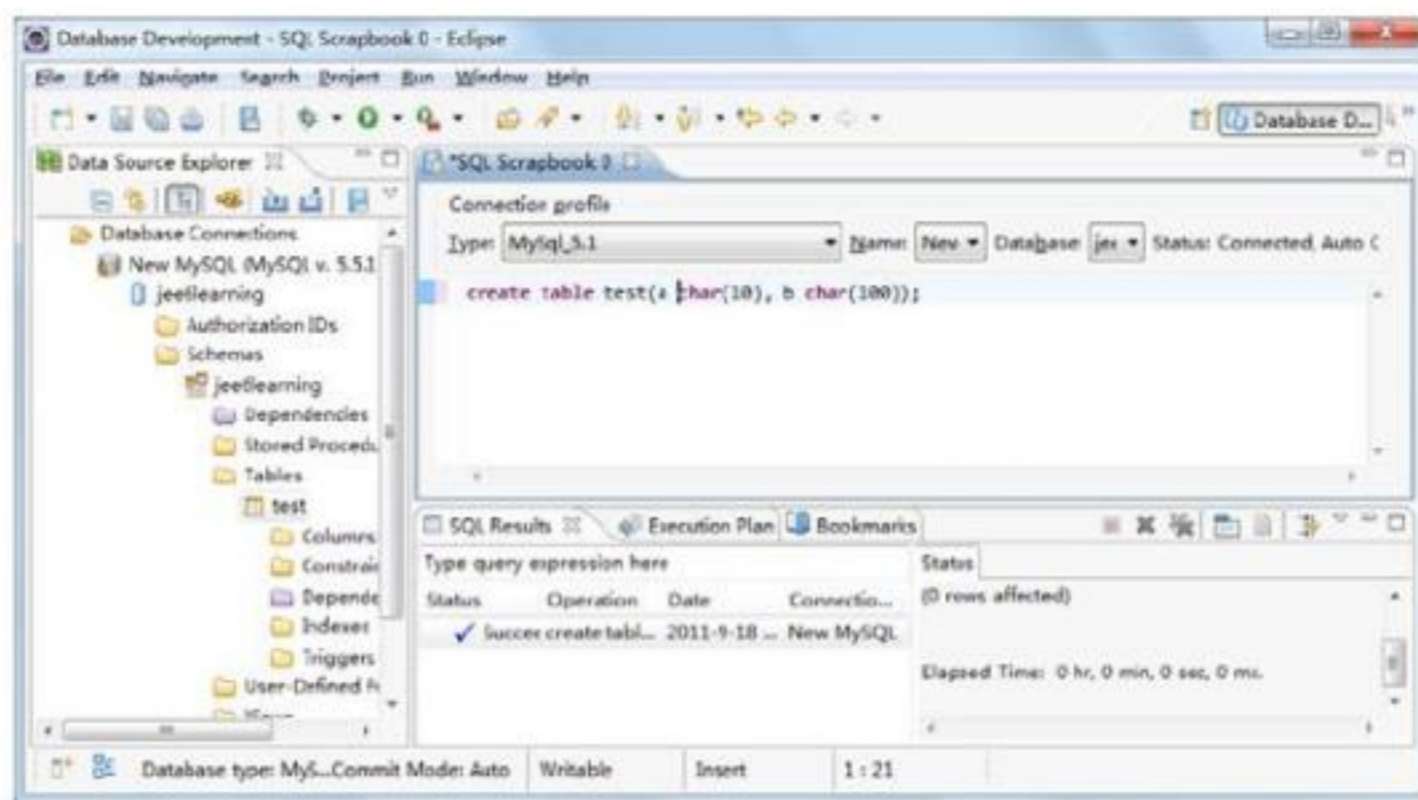


图 1.45 测试

1.2.4 EJB 容器数据源的配置

上一节已经将数据库集成到 Eclipse 中，在 Eclipse 环境下能够访问数据库了。但 EJB 容器还无法访问数据库，必须为 EJB 容器配置数据源，这是开发 EJB 的重要步骤。如果不会配置数据源，就无法在 Java EE 环境下使用数据库。

1. 概述

在 EJB 容器中配置数据源是一项重要的任务。

在 GlassFish 中配置数据源包括如下几个步骤。

(1) 把数据库的 JDBC 驱动拷贝到 GlassFish 安装目录的 lib 目录下。由于 GlassFish 自带 JavaDB Derby 的驱动，因此，如果使用 Derby 数据库，这一步可以省略。

(2) 创建 JDBC 连接池。

(3) 创建引用 JDBC 连接池的 JNDI 名称。

值得一提的是，GlassFish 本身就带有 TopLink 的 JPA 实现，如果对 JPA 实现没有特殊的要求，可以直接使用自带 TopLink，没有必要使用其他 JPA 实现。

第一步比较简单，本书使用 MySQL，就以 MySQL 驱动进行说明。MySQL 的 JDBC 驱动文件名称类似于 mysql-connector-java-5.1.6.jar，将该文件拷贝到 GlassFish 安装目录的 lib 目录下，如 C:\glassfish3\glassfish\lib。

接下来的两步有两种方式——使用 GlassFish 管理控制台或使用 asadmin 命令行，两者都能达到同样的目的，读者可根据自己的喜好进行选择。不管采用哪种方式，都要确保首先启动 GlassFish 服务器。

2. 使用 GlassFish 管理控制台配置数据源

使用 GlassFish 管理控制台配置数据源非常直观。首先打开浏览器，在地址栏输入“http://localhost:4848”，然后按 Enter 键，打开 GlassFish 管理控制台，导航至 Resources → JDBC → JDBC Connection Pools，如图 1.46 所示。

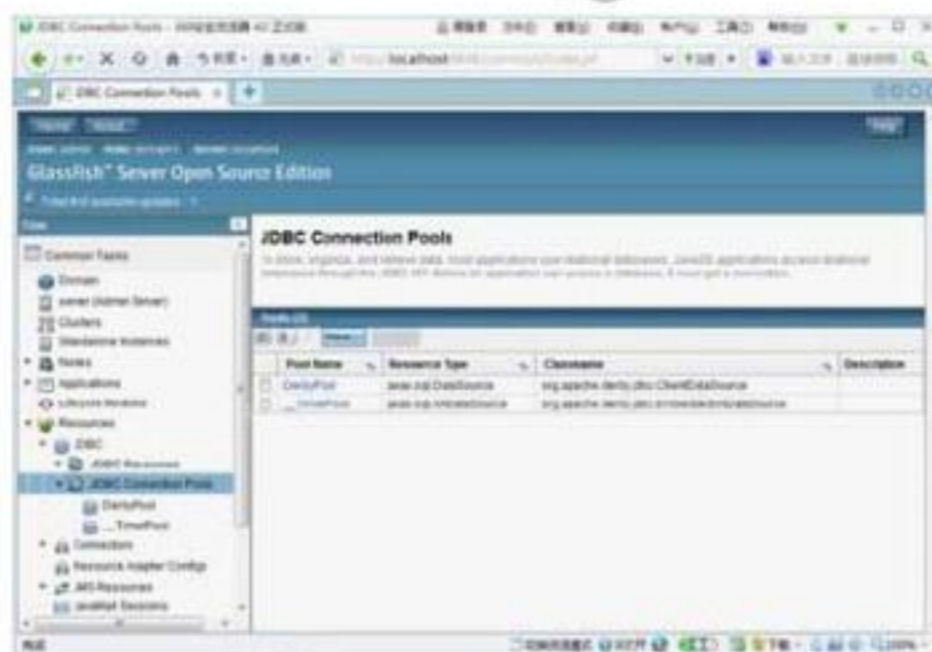


图 1.46 显示已配置的 JDBC 连接池

由图 1.46 可见，GlassFish 默认已配置好两个 JDBC 连接池。单击 New 按钮创建新连接池。进入第一步配置，如图 1.47 所示。

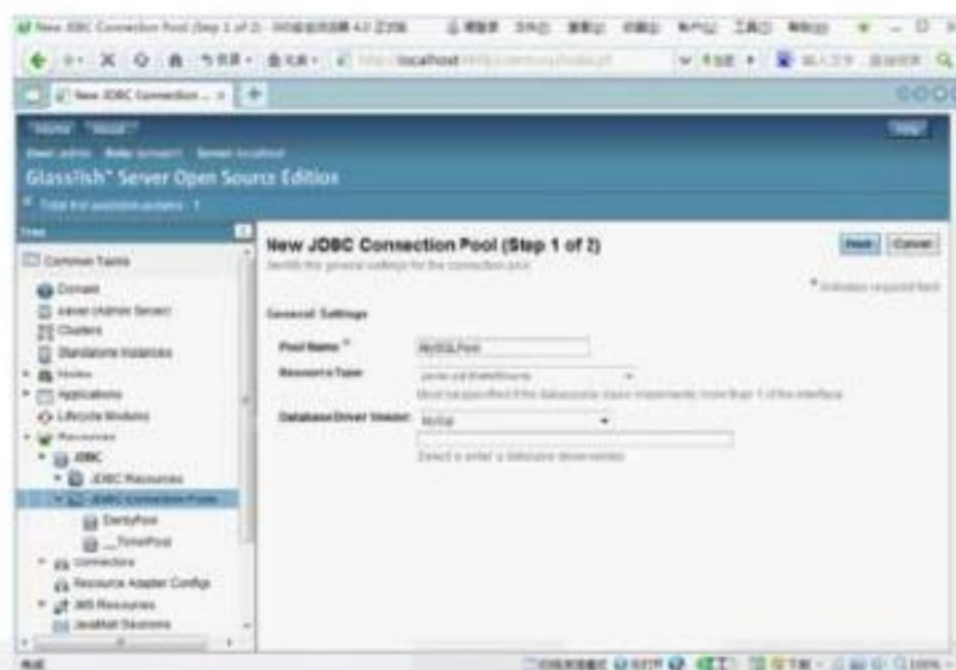


图 1.47 配置 JDBC 连接池第一步

输入连接池名称为“MySQLPool”，选择资源类型为 `javax.sql.DataSource`，选择数据库供应商为 MySQL，单击 Next 按钮进入第二步，如图 1.48 所示。



图 1.48 配置 JDBC 连接池第二步

参考表 1.2，数据库用户名和密码根据自身数据库设置进行配置，完成后单击 Finish 按钮结束。



表 1.2 数据库连接选项

名 称	释 义	值
ServerName	服务器名称	localhost
DatabaseName	数据库名称	jee6learning
User	用户名	APP
Password	密码	APP
PortNumber	端口号	3306
CharacterEncoding	字符编码	utf8
ConnectionAttributes	连接属性	
URL	连接 URL	jdbc:mysql://localhost:3306/jee6learning

可以 Ping 数据库以测试配置是否正确。选中这个 MySQLPool 连接池，在 General 选项卡中单击 Ping 按钮进行测试，如图 1.49 所示。



图 1.49 Ping 数据库

下一步需配置 JDBC 源。导航至 Resources → JDBC → JDBC Resources，如图 1.50 所示。

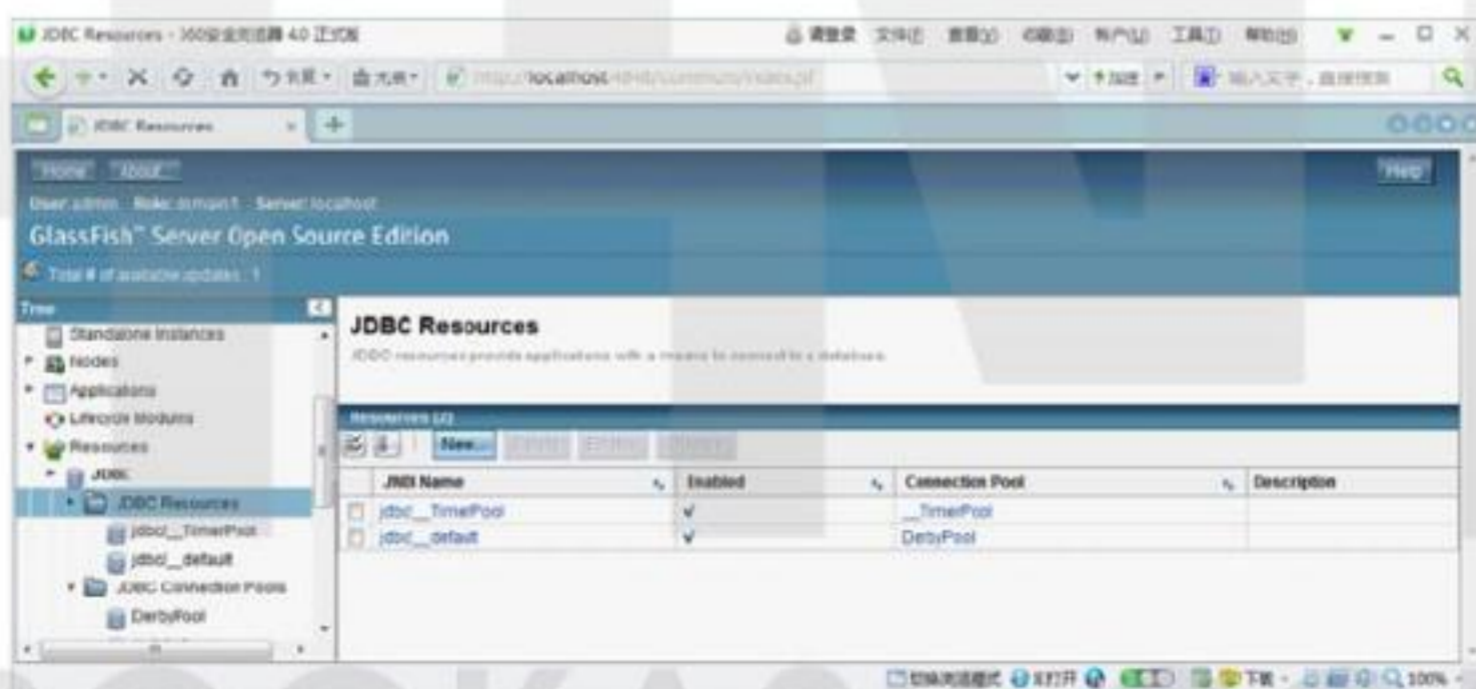


图 1.50 已配置的 JDBC 源

单击 New 按钮创建新的 JDBC 源。进入新 JDBC 源配置，如图 1.51 所示。



图 1.51 创建新 JDBC 源

输入 JNDI 名称(JNDI Name)为“jdbc/MySqlDB”，选择连接池名称(Pool Name)为上一步建立的连接池名称，如 MySQLPool。单击 OK 按钮结束配置。

3. 使用 asadmin 命令行配置数据源

使用 asadmin 命令行不但能够完成与 GlassFish 管理控制台一样的工作，而且还能进行更细微的控制，这种方法适合编程人员，对于初学者来说比较难。

首先，通过“开始”→“运行”命令打开“运行”对话框，并输入“cmd”命令打开命令行窗口。然后，在命令提示符下输入如下命令并按 Enter 键：

```
asadmin --port 4848 --user admin create-jdbc-connection-pool --
datasourceclassname=com.mysql.jdbc.jdbc2.optional.MysqlDataSource --
restype=javax.sql.DataSource --property
portNumber=3306:password=APP:user=APP:serverName=localhost:databaseName=
javaee MySQLPool
```

上述命令将创建名称为 MySQLPool 的 JDBC 数据库连接池。

输入如下命令查看 JDBC 数据库连接池：

```
asadmin --port 4848 --user admin list-jdbc-connection-pools
```

输入如下命令 Ping 数据库连接池：

```
asadmin --port 4848 --user admin ping-connection-pool MySQLPool
```

输入如下命令创建 JDBC 数据源：

```
asadmin --port 4848 --user admin create-jdbc-resource --connectionpoolid
MySQLPool jdbc/MySqlDB
```

输入如下命令查看 JDBC 数据源：

```
asadmin --port 4848 --user admin list-jdbc-resources
```

上述方法每次都需要重复输入“asadmin --port 4848 --user admin”字符，效率低下，更好的方法是在命令行下首先输入如下命令：

```
asadmin --port 4848 --user admin
```

进入到 asadmin 的交互模式，然后再一条一条地输入 create-jdbc-connection-pool 等命令，最后输入 exit 退出交互模式。如图 1.52 所示。

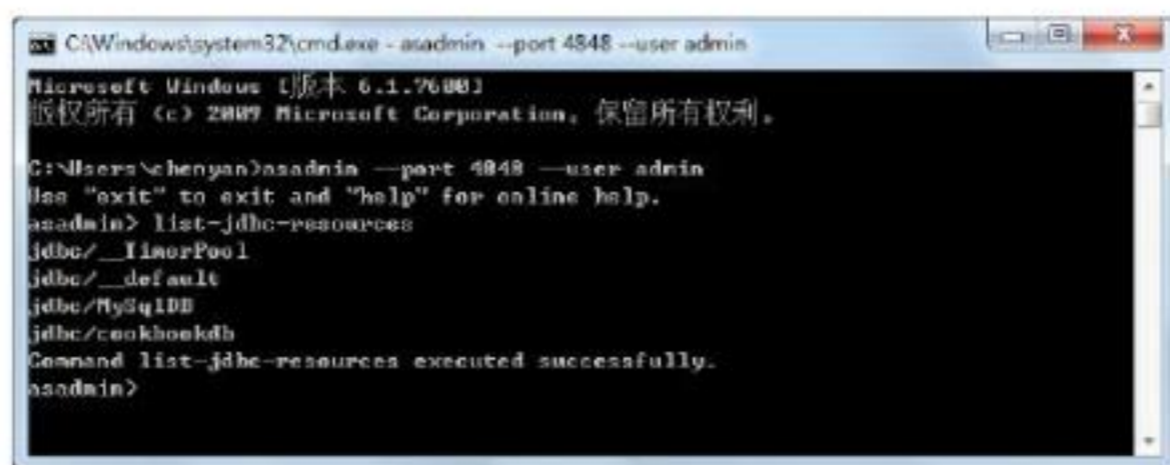


图 1.52 asadmin 交互模式

1.3 Java EE 安全

身处当今的全面信息化网络时代，信息安全问题是广受关注的社会问题，无论如何强调 Web 应用安全的重要性也不过分。Java EE 服务器的一项核心功能就是为应用提供安全控制。通过支持定义良好的、可扩展的 API，将认证、授权与应用逻辑相分离。程序开发人员只要专注业务逻辑的实现，系统安全服务完全由应用服务器负责。

本章后续部分讲述 Java EE 安全，并且提供实例进行说明，如果对安全概念与使用不感兴趣，可跳过这些章节。

1.3.1 Java EE 基本安全概念

1. 概述

大多数应用程序都需要实现登录和访问控制，JAAS(Java Authentication and Authorization Service, Java 认证与授权服务)通过对运行程序的用户的进行验证，从而达到保护系统的目的。JAAS 为 Java EE 中的 EJB 组件、Web 组件等安全提供了良好的基础。

实现 Java EE 应用程序安全性的编码量并不大，在大多数情况下，通过为应用服务器的安全域设置用户和用户组，然后配置应用程序，使之依赖于特定安全域进行验证和授权，就可以确保应用程序的安全。

JAAS 是对 JCE(Java Cryptography Extension, Java 密码扩展)安全框架的重要补充，通过提供认证用户和对用户授权来增强 Java 解决方案的动态安全性，使资源能够得到很好的保护和控制。JAAS 使用动态的安全策略来定义权限，而不是使用静态的方式将其嵌入到代码中。

JAAS 采用的是插件的运行方式，从一开始就被设计成可插拔的，根据应用需要，只要配置一下 JAAS 的配置文件，即可将这些组件包含在应用程序中。使用 JAAS 包接口，开发人员可以开发一些组件或者 Bean 来实现登录认证，或者通过与使用者或外部系统进行交互来获取认证信息。JAAS 提供了一组用于鉴别用户的类和接口，还提供了另一组用于用户授权的类和接口。

虽然各类应用服务器底层验证系统存在差异，但 JAAS 会屏蔽这些差异，能够使应用



本书试读到此结束啦！



[Java EE企业级编程开发实例详解](#)

作者：袁梅宇, 王海瑞, 著

出版社：清华大学出版社

通过以下方式阅读更多 [Powered by 书问](#)



● 扫码分享到朋友圈可阅读更多



立即扫码



● 还不过瘾？购买书库畅读卡全本畅读此书！



[查看全部书库](#)



● 购买纸书也可畅读全本哦！

[当当网](#)

[中国图书网](#)

[广购书城](#)