

Robin M. Roos 著
周兆确 译

Java Data Objects

Java 数据对象

Java Data Objects



人民邮电出版社
POSTS & TELECOM PRESS

Java 数据对象

Robin M. Roos 著

周兆确 译

人民邮电出版社

图书在版编目 (CIP) 数据

Java 数据对象 / (美) 鲁斯 (Roos, R. M.) 著; 周兆确译. —北京: 人民邮电出版社, 2004. 2
ISBN 7-115-11804-3

I. J... II. ①鲁...②周... III. JAVA 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2003) 第 127280 号

版 权 声 明

Robin M. Roos: Java Data Objects

ISBN: 0321123808

Copyright ©Pearson Education Ltd 2003

This translation of Java Data Objects, First Edition is published by arrangement with Pearson Education Limited.

本书中文简体字版由 Addison Wesley 出版公司授权人民邮电出版社出版, 未经出版者书面许可, 对书的任何部分不得以任何方式复制或抄袭。

版权所有, 侵权必究。

Java 数据对象

- ◆ 著 Robin M. Roos
- 译 周兆确
- 责任编辑 陈冀康
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
读者热线 010-67132705
北京汉魂图文设计有限公司制作
北京顺义振华印刷厂印刷
新华书店总店北京发行所经销
- ◆ 开本: 787×1092 1/16
印张: 15
字数: 351 千字 2004 年 2 月第 1 版
印数: 1-4 000 册 2004 年 2 月北京第 1 次印刷

著作权合同登记 图字: 01-2003-1493 号

ISBN 7-115-11804-3/TP • 3725

定价: 29.00 元

本书如有印装质量问题, 请与本社联系 电话: (010) 67129223

序

虽然 JDO (Java Data Object, Java 数据对象) 标准已经有比较“悠久”的历史了,但在 JDO 标准的发展过程中,我们对该标准的众多规范也只能略窥一斑! JDO 标准起源于 ODMG (Object Data Management Group, 对象数据管理组) 标准。这个标准很早就尝试在面向对象的编程语言中对以透明方式访问数据库的方法实现标准化。ODMG 标准比 Java 语言本身更早出现。这个标准是在面向对象技术领域中的一次大讨论——究竟是 Smalltalk 还是 C++ 才是面向对象编程语言的主流——中诞生的。事实证明这场争论只不过是一场纯学术领域的争论,Java 语言最终修成正果,成为了编写面向对象应用程序的事实上的标准。

将 ODMG 标准用于 Java 遇到了很多问题。ODMG 标准是在两层结构的应用程序占主导地位时出现的。应用程序服务器的模型化非常困难。因为 ODMG 标准缺乏兼容性测试,所以任何软件提供商,只要其产品表面上看起来和 ODMG 有几分类似,那么他们就能对外宣称其产品是和 ODMG 标准兼容的。

计算机技术发展到了 JCP (Java Community Process) 时代。表面上看, JCP 解决了 ODMG 标准中的一些固有的且非常棘手的问题——对参考实现和兼容性测试工具的需求。人们经过深思熟虑之后认可了该标准,而且业界提交了 Java 技术规范需求;此外,作为一种规范, JCP 也获得了应用程序开发人员的认可。与此同时,这一标准获得了一些主要的数据库厂商、中间件厂商和软件工具提供商的大力支持。

JDO 标准是和其他两种标准——JCA (Java Connector Architecture, Java 连接器体系结构) 和 CMP (Container-Managed Persistence, 容器管理的永久性) 同步发展的。尽管应用程序的易用性和简单性仍然是一个绝对重要的目标,但 JDO 专家组仍然坚信使 JDO 标准和这两个标准同步发展是非常重要的。

JCP 对参考实现和兼容性测试工具的需求成为决定该标准的最终版本出现时间的决定性因素。JCP 的参考实现是由一个专职软件工程师实现的,而兼容性测试工具则是由一位兼职工程师实现的。就通常的工程技术标准而言,这两个项目的人手严重不足。但事实证明这也有好的一面:商业实现的开发是和参考实现及测试工具的开发同步进行的。尽管参考实现发现了这个标准中的很多瑕疵,但试图在其数据库产品中采用这个标准的软件提供商们在这个标准中发现了更多的缺陷。事实上,有好些产品在测试还未完成之前就面世了。这些产品为这个处于发展中的标准提供了更多的支持。

我非常高兴地看到 JDO 不仅在用户方面而且在软件提供商方面都获得了认同。JDO 在两层结构和多层结构的应用程序以及不同数据库平台上的实现为 Java 对程序员的承诺“一次写成,处处可用”增加了新的含义。

内 容 提 要

本书是对 JDO (Java Data Object, Java 数据对象) 规范和技术的具体介绍和学习指南。

全书共 13 章, 分别介绍了 JDO 的体系结构、对象生命周期状态、基本接口和类、事务管理、JDOQL 语言、异常、与 J2EE 的集成以及 JDO 实现方面的知识。附录部分给出了 JDOHelper、PersistenceManagerFactory 接口以及 JDOQL 语法等有用的参考资料。

本书适用于具有一定的 Java 基础知识的 Java 应用程序开发人员, 以及对 JDO 技术感兴趣的程序设计人员和程序架构设计人员。

前 言

在过去两周内，JCP 的执行委员会一直在考虑 JDO 的技术规范问题，并对 JDO 技术规范进行表决。现在，JDO 已经被当作了一种标准，在笔者写作本书时，该标准的 1.0 版本已经发布在 Yahoo!Groups 的 Java DataObjects 和 JDOcentral.com 上了。

我相信 JDO 将会对我们规划、设计和实现 Java 应用程序产生深远的影响。据保守估计，使用 JDO 开发企业应用程序至少能节约 20% 的开发时间（从而也至少能节约 20% 的费用），某些分析人员认为使用 JDO 实际能节约的时间可能比这个估计的时间更多。

业界为这个标准提供的支持是相当多的；目前，市场上已经有大量关于 JDO 的产品了，绝大部分产品的目标都是在不久的将来完全满足 JDO 1.0 版的技术规范。

JDO 是什么？

JDO 是 Java 语言基于接口的对象永久性定义，这个定义描述了应用程序在数据存储过程中间上存储、查询和检索对象的方式。

凭借 JDO 所支持的透明永久性的概念，JDO 是一种非常具有竞争力的标准。我们可以将透明永久性特性简单地归纳如下：

- JDO 透明地处理 JDO 实例和底层数据存储空间上的数据映射关系，也就是非 JDO 环境中所谓“对象—关系”之间的阻抗失配问题。
- 对具有永久性属性的 Java 对象而言，JDO 是透明的；应用程序开发人员不必为实例的永久性而向 Java 类增加任何特定的方法或属性，或者修改 Java 类的数据成员的可视性修饰符。对 JDO 而言，使用 `private` 修饰符的数据成员或者没有使用 `get/set`（读取/设置数据成员）方法的数据成员，都不会有任何问题。
- JDO 可被用于多种经典的数据存储方式，这些方式包括（但不局限于）关系数据库、对象数据库、文件系统和 XML 文档。在不远的将来，业界将会推出用于访问现有应用程序中的永久性数据的 JDO 实现，这种 JDO 实现将被用于企业应用程序集成领域。
- 因为 JDO 对数据存储方式本身也是透明的，所以只要选择一种适当的 JDO 实现，我们就能将同一个应用程序移植到不同的数据存储空间上。JDO 技术规范所保证的 JDO 实例二进制代码的兼容性，意味着程序员在不需要对应用程序进行重新编译的情况下就能实现应用程序在不同 JDO 实现之间的可移植性，更不用说对应用程序源代码进行任何修改了。
- 如果一个应用程序引用了一个永久性对象，并且该应用程序在内存中对这个永久性对象的状态进行了某种修改，那么当该对象所在的事务被提交时，JDO 实现将以透明方式对数据存储空间上的数据进行更新。这个特性使得应用程序开发人员不必再重复性地编写用于保存应用程序对数据所做操作的代码。

JDO 将受到哪些人的青睐？

因为 Java 应用程序的结构设计人员可以利用 JDO 创建灵活并且能与 J2EE 无缝连接的应用程序体系结构，所以 JDO 将会受到 Java 领域中这类人员的青睐。他们不仅会把 JDO 的可移植性应用到关系数据库、对象数据库和其他数据存储方式中，而且还会将 JDO 的广泛适应性用于企业级应用程序和简单的“客户端—服务器”体系结构的应用程序中，以及在不远的将来应用到 J2ME 的环境中。

因为 JDO 最终将使用各种优秀的对象建模技术（包括接口和继承）——以前，这些技术曾经为实现对象的永久性导致了应用程序的巨大复杂性——所以 JDO 将会受到 Java 应用程序设计人员的欢迎。

因为 JDO 不仅非常直观而且能通过域对象模型的层次上——不需要这些域对象为其应用程序实现永久性基础结构——JDO 还将受到 Java 应用程序开发人员的重视。此外，应用程序开发人员还非常喜欢 JDO 提供的类似于 Java 语言语法的新型查询语言 JDOQL。

最后，因为 JDO 能使应用程序开发过程更加流畅，所以软件开发人员和项目经理也会喜欢使用 JDO。应用程序的设计人员和开发人员可将自己的精力完全集中到应用程序需要实现的业务功能方面，而不需要花费大量的人力物力来实现和调试应用程序中非功能性的永久性基础结构。

本书适合哪些读者阅读？

尽管本书主要是为 Java 应用程序开发人员编写的，但本书所涉及的 JDO 也将引起应用程序设计人员和应用程序结构设计人员的巨大兴趣。

本书的读者需要具有 Java 语言方面的丰富知识，并熟悉 Java 语法。

过去，Java 应用程序是通过使用 JDBC 和 SQL 来将数据存储到关系数据库中的。JDO 可以彻底取代这两种技术，因此，读者不需要为学习 JDO 而了解关于 JDBC 和 SQL 的任何知识。当然，如果读者有在数据库中存储数据的知识和经验，那么这些背景知识对于读者学习 JDO 还是有帮助的。

我们可在 J2EE 应用程序服务器中和 J2EE 应用程序服务器外成功地使用 JDO。基于 JDO 的应用程序既可以是标准的 Java 客户端，也可以是 Web 部件（服务器端小程序和 JSP）和 EJB（会话 Bean、消息驱动 Bean 和实体 Bean）。

因为 JDO 被大量用于域对象，所以笔者在本书中花费了大量的篇幅来向读者介绍如何灵活地设计域对象模型。Java 应用程序设计人员将会对这部分内容感兴趣，但他们可能会忽略关于语法知识方面的例子。

JDO 与 J2EE 部件的集成将会成为 Java 应用程序结构设计人员和应用程序开发人员的兴趣之所在，因此，本书专门用了一章的篇幅（第 11 章）来介绍这方面的内容。本书第 11 章首先向不熟悉 J2EE 的读者介绍了 J2EE 的各个组成部件。因此，虽然具有 J2EE 知识的读者肯定会比不具备这方面知识的读者更加深入地理解这部分内容，但就本书的学习而言，即使

读者没有关于 J2EE 方面的预备知识也是无关紧要的。

本书的内容结构

第 1 章 理解对象永久性

这一章向读者概要介绍了如何将对象存储在数据空间上，重点讨论了 JDO 技术和传统技术之间的差异。

第 2 章 一个基于 JDO 的简单应用程序

在这一章中，我们向读者逐一介绍开发一个基于 JDO 的简单应用程序所需要的步骤。

第 3 章 JDO 的体系结构

本章向读者介绍了 JDO 体系结构中的一些核心概念，比如环境、事务性和永久性实例的身份标识等。

第 4 章 JDO 实例的生命周期状态

这一章的主要内容是向读者介绍 JDO 实例的生命周期中涉及到的各种状态，以及应用程序能调用哪些方法来实现 JDO 实例生命周期中不同状态之间的转换和过渡。

第 5 章 永久性对象模型

本章向读者详细介绍 JDO 是如何处理域对象模型的。

第 6 章 JDO 的基本接口和类

本章将从应用程序的角度向读者介绍 JDO 的核心内容。

第 7 章 事务管理

本章向读者介绍 JDO 如何管理事务，重点介绍了乐观事务策略。

第 8 章 使用 JDOQL 语言查询数据

本章向读者介绍一门新型的查询语言：JDOQL。

第 9 章 JDO 异常

本章向读者介绍 JDO 定义的异常类层次结构和产生这些异常的条件。

第 10 章 永久性描述符

本章详细介绍基于 XML 格式的永久性描述符文档的结构。

第 11 章 JDO 与 J2EE 的集成

本章介绍如何在 EJB 部件（会话 Bean、消息驱动 Bean 和实体 Bean）和 Web 部件（服务器端小程序和 JSP）的基础上使用 JDO。

第 12 章 JDO 实现

本章向读者介绍目前业界提供的 JDO 实现。业界为 JDO 技术规范和技术规范的改进提供了很好的实现。

第 13 章 结语

本章向读者介绍 JDO 1.0 版技术规范之外的一些内容，重点考虑了未来的 JDO 技术规范可能实现的某些特性。

附录 A JDOHelper 启动属性

这部分内容介绍为初始化 JDO 所需的一些有用属性的名称。

附录 B JDO 实现所支持选项的字符串

这个附录向读者介绍 JDO 所支持的一些可选特性和这些可选特性对应的字符串常量。

附录 C JDO 永久性描述符的 DTD 文件

这个附录包含了永久性描述符的一个 DTD 文件。

附录 D PersistenceManagerFactory 接口

这个附录列出了 PersistenceManagerFactory 接口提供的一些方法——我们没有在本书正文部分向读者介绍这些方法。

附录 E JDOQL 语法

这部分内容向读者介绍 JDOQL 的语法符号。

此外，本书还向读者提供了一些有用的参考书目。

本书附带光盘上的内容

本书附带光盘提供了 JDO 实现的如下一些试用版本：

- Poet 公司的 FastObjects；
- Solarmetric 公司的 Kodo JDO；
- LiBeLIS 公司的 LiDO；
- Prism Technologies 公司的 OpenFusion JDO。

本书第 12 章对所有上述产品进行了介绍。关于这些 JDO 实现的进一步细节，请读者参考本书附带光盘中的 readme.pdf 文件。

读者可从笔者的 Web 站点上获得本书各章中重要应用程序示例的源代码。笔者的 Web 站点的地址为：

<http://www.OgilviePartners.com>

JDO 的版本信息

本书介绍的是 JDO 1.0 版技术规范的内容。

Java 应用程序示例是与 JDK 1.3.1_01 和 J2EE 1.3 兼容的。

关于本书的作者

Robin Roos 曾经在南非的 St. Andrews 学院和 Rhodes 大学求学，于 1988 年获得计算机科学和物理学学士学位，并于 1989 年获得计算机科学荣誉学士学位。在 1996 年开始学习 Java 之前，Robin Roos 在关系数据库和各种过程化语言和面向对象语言（包括 Forte 语言）方面就积累了丰富的经验。1996 年之后，Robin Roos 参与过多个 Java 和 J2EE 方面的项目。根据在 JDBC 和实体 Bean 的 Java 平台上访问数据库的经验，Robin Roos 坚信在 Java 平台上一定还有另一种访问数据库的更好方式。正是在这个信念的鼓舞下，他于 2001 年加入了 JDO Expert Group（JDO 专家组）。

Robin 是英国 Ogilvie Partner 公司的首席顾问, 该公司的主要业务是为日益增长的遍布世界各地的客户群提供技术培训、项目指导和咨询服务。Ogilvie Partner 公司在核心 Java J2EE 技术上的实际项目背景使其在 JDO 方面的发展受益匪浅。

迄今为止, Robin 已经在 JDO 用户组和英国、欧洲和美国的很多有关 JDO 的会议上作过多次发言。Robin 的联系方式是:

Robin@OgilviePartner.com

<http://www.OgilviePartner.com>

目 录

第 1 章 理解对象永久性	1
1.1 什么是对象的永久性	2
1.2 目前所采用的对象永久性技术	2
1.2.1 关系数据库	3
1.2.2 文件系统	4
1.2.3 对象数据库	4
1.2.4 实体对象 bean (Entity Bean)	5
1.3 使用 JDO 实现对象的永久性	5
1.4 JDO 在各种不同的永久性技术中的地位	8
1.5 下一章内容	9
第 2 章 一个基于 JDO 的简单应用程序	10
2.1 定单处理应用程序的域对象模型	10
2.2 讨论	15
2.2.1 改进过程和对象的永久性特性	16
2.2.2 JDOBootstrap 和 PersistenceManagerFactory 类	17
2.2.3 事务 (Transactions)	17
2.2.4 临时性和永久性的比较	17
2.2.5 以循环方式处理位于 Extent 空间中的所有永久性对象	18
2.3 应用程序源代码	18
2.4 下一章内容	20
第 3 章 JDO 的体系结构	21
3.1 JDO 的实现和 JDO 的软件提供商	21
3.2 JDO 实例	22
3.3 JDO 环境	22
3.3.1 非管理性环境	22
3.3.2 管理性环境	23
3.3.3 JDO 两种使用环境之间的比较	24
3.4 对象永久性和临时性之间的比较	24
3.5 事务性操作和非事务性操作的比较	25
3.6 JDO 对事务性 / 永久性实例的支持	26
3.6.1 具有临时性和事务性特性的实例	26
3.6.2 具有永久性和非事务性特性的实例	27
3.7 JDO 身份标志	27

3.7.1 数据存储身份标志	28
3.7.2 应用程序身份标志	29
3.7.3 非永久性 JDO 身份标志	31
3.7.4 几种不同类型的 JDO 身份标志的比较	32
3.8 下一章内容	32
第 4 章 JDO 实例的生命周期状态	33
4.1 判断一个实例的状态	33
4.2 必须的生命周期状态	34
4.2.1 临时性生命周期状态 (Transient)	34
4.2.2 实例初次实现永久化的状态 (Persistent-New)	35
4.2.3 新创建的实例在同一事务中被删除时的状态 (Persistent-New-Deleted)	35
4.2.4 空状态 (Hollow)	35
4.2.5 永久性干净状态 (Persistent-Clean)	36
4.2.6 永久性脏状态 (Persistent-Dirty)	36
4.2.7 永久性删除状态 (Persistent-Deleted)	36
4.3 JDO 实现必须提供的生命周期状态转变特性	37
4.3.1 永久化一个处于临时性状态的实例	37
4.3.2 在一个事务中创建 / 删除永久性实体对象	38
4.3.3 读取实例的数据成员的值	38
4.3.4 将实例清除出内存 (Eviction)	39
4.3.5 更新永久性实例数据成员的值	39
4.3.6 刷新永久性实例数据成员的值	40
4.3.7 删除一个永久性实例	41
4.3.8 删除一个永久性脏实例	42
4.3.9 将一个实例的状态设置为临时性状态	43
4.3.10 所有必须的状态转化	44
4.4 可选生命周期状态	45
4.4.1 永久性非事务状态 (Persistent-Nontransactional)	45
4.4.2 临时性干净状态 (Transient-Clean)	45
4.4.3 临时性脏状态 (Transient-Dirty)	45
4.5 可选生命周期状态之间的转换	45
4.5.1 乐观事务	45
4.5.2 在事务之外对永久性数据成员的访问	46
4.5.3 对临时性实例的事务性访问	47
4.6 InstanceCallbacks 接口	49
4.6.1 数据成员加载后 (Post Load)	49
4.6.2 数据存储之前 (Pre Store)	50
4.6.3 数据清除之前 (Pre Clear)	50

4.6.4 实例删除之前 (Pre Delete)	50
4.7 下一章内容	51
第 5 章 永久性对象模型	52
5.1 永久性的透明性	52
5.2 JDO 实例	54
5.3 第一类对象和第二类对象	55
5.3.1 对象变化的可见性	57
5.3.2 具有永久性属性的类作为第二类对象	58
5.4 “第三类对象”——数组	59
5.5 JDO 对永久性数据成员所能使用的数据类型的限制	60
5.5.1 具有非永久性和非事务性属性的数据成员	60
5.5.2 具有事务性属性和非永久性属性的数据成员	60
5.5.3 具有永久性属性的数据成员	60
5.6 继承	78
5.7 附加部分: 定单处理系统的 GUI	84
5.8 下一章内容	85
第 6 章 JDO 的基本接口和类	86
6.1 JDOHelper 类	86
6.1.1 JDOHelper 类中与判断永久性实例状态有关的方法	86
6.1.2 JDOHelper 类中与管理功能有关的方法	87
6.1.3 JDOHelper 类中与 JDO 实现的引导功能相关的方法	90
6.2 关于 JDO 的属性说明	92
6.2.1 ConnectionFactory 属性	94
6.2.2 启动 JDO 实现	94
6.3 PersistenceManagerFactory 接口	98
6.4 PersistenceManager 接口	100
6.4.1 缓冲区管理	100
6.4.2 实例的生命周期	101
6.4.3 使用 JDO 身份标志	104
6.4.4 管理功能	105
6.4.5 获得实例	106
6.4.6 查询工厂类方法	107
6.5 Extent (类的范围) 接口	107
6.6 PersistenceCapable 接口	110
6.7 下一章内容	110
第 7 章 事务管理	111
7.1 事务	111
7.2 Transaction 接口	111

7.3 事务策略	112
7.3.1 “悲观”（数据存储空间）事务策略	113
7.3.2 “乐观”事务策略	114
7.3.3 乐观事务和刷新	116
7.3.4 使用明确的刷新操作的解决方案	119
7.4 高级事务选项	119
7.5 为提高应用程序效率的事务模式	119
7.5.1 RestoreValues 模式	119
7.5.2 RetainValues 模式	120
7.6 与 JDO 实例的同步化	120
7.7 下一章内容	122
第 8 章 使用 JDOQL 语言查询数据	123
8.1 Query 接口的体系结构	123
8.2 构造 Query 对象（实现了 Query 接口的类的对象）	125
8.3 Query 接口	126
8.3.1 Query 对象的执行	128
8.3.2 关闭 Query 对象的返回结果	128
8.4 关于 Query 对象的应用程序实例	128
8.4.1 不使用过滤器的 Query 对象	128
8.4.2 使用实例排序规则的 Query 对象	130
8.4.3 使用过滤器的 Query 对象	130
8.5 查询对象的过滤器表达式	130
8.5.1 JDO 支持的逻辑操作符	131
8.5.2 JDOQL 支持的关键字	132
8.5.3 JDOQL 和 Java 操作符之间的区别	132
8.6 更加复杂的例子	134
8.6.1 参数化	134
8.6.2 在单个数据成员之间的导航	136
8.6.3 集合数据成员之间的导航	136
8.7 非约束性查询变量	137
8.8 动态查询窗口（Dynamic Query Window）	137
8.9 下一章内容	139
第 9 章 JDO 异常	140
9.1 JDO 异常类层次结构	140
9.2 基本异常类	141
9.2.1 JDOException 异常类	141
9.2.2 JDOFatalException 异常类	142
9.2.3 JDOCanRetryException 异常类	142

9.2.4	JDOUserOptionException 异常类	142
9.2.5	JDOUnsupportedOptionException 异常类	143
9.2.6	JDOFatalUserException 异常类	143
9.2.7	JDOFatalInternalException 异常类	143
9.2.8	JDODataStoreException 异常类	143
9.2.9	JDOFatalDataStoreException 异常类	143
9.3	应用程序异常	143
9.4	下一章内容	144
第 10 章	永久性描述符	145
10.1	XML 文档概述	145
10.2	永久性描述符的命名规范	146
10.3	永久性描述符的元素	146
10.3.1	<extension>标记	147
10.3.2	<jdo>标记	147
10.3.3	<package>标记	147
10.3.4	<class>标记	147
10.3.5	<field>标记	148
10.3.6	<collection>标记	150
10.3.7	<map>标记	151
10.3.8	<array>标记	152
10.4	实例——永久性描述符文件“op.jdo”	152
10.5	JDO 的软件提供商对永久性描述符所作的特定扩展	154
10.6	实例——永久性描述符中的删除语法	155
10.7	下一章内容	156
第 11 章	JDO 与 J2EE 的集成	157
11.1	管理性环境	157
11.2	J2EE 简介	157
11.2.1	EJB 层	157
11.2.2	Web 层	160
11.3	JDO 实例的序列化	161
11.4	JDO 中的事务和 J2EE 中的事务的比较	162
11.5	JDO 与 EJB 的集成	163
11.6	JDO 与无状态会话 Bean 的集成	164
11.7	JDO 与有状态会话 Bean 的集成	167
11.8	JDO 和实体 Bean 之间的集成	171
11.9	JDO 和消息驱动 Bean 的集成	179
11.10	JDO 与 Web 层的集成	181
11.10.1	服务器端小程序	182

11.10.2 JSP	184
11.11 在管理性环境中启动 JDO	186
11.12 下一章内容	186
第 12 章 JDO 的实现	187
12.1 Versant 公司的 enJin 产品	189
12.1.1 关于 Versant 公司	189
12.1.2 关于 enJin	189
12.1.3 enJin 支持的数据存储方式	189
12.1.4 enJin 支持的应用程序服务器	189
12.2 Poet Software 公司的 FastObjects 产品	189
12.2.1 关于 FastObjects 公司	190
12.2.2 关于 FastObjects 公司的产品	190
12.2.3 FastObjects 公司的产品家族	190
12.2.4 FastObjects 支持的数据存储方式	191
12.2.5 FastObjects 支持的应用程序服务器	191
12.3 ObjectFrontier 公司的 FrontierSuite for JDO 产品	191
12.3.1 关于 ObjectFrontier 公司	191
12.3.2 关于 FrontierSuite for JDO 产品	191
12.3.3 ObjectFrontier 公司的产品家族	191
12.3.4 FrontierSuite for JDO 支持的数据存储方式	192
12.3.5 FrontierSuite for JDO 支持的应用程序服务器	193
12.4 Signsoft 公司的 IntelliBO 产品	193
12.4.1 关于 Signsoft 公司	193
12.4.2 关于 IntelliBO 产品	193
12.4.3 IntelliBO 支持的数据存储方式	194
12.4.4 IntelliBO 支持的应用程序服务器	194
12.5 Hemisphere Technologies 公司的 JDO Genie 产品	194
12.5.1 关于 Hemisphere Technologies 公司	194
12.5.2 关于 JDO Genie 产品	195
12.5.3 JDO Genie 支持的数据存储方式	195
12.5.4 JDO Genie 支持的应用程序服务器	195
12.6 Object Industries 公司的 JRelay 产品	195
12.6.1 关于 Object Industries 公司	195
12.6.2 关于 JRelay 产品	196
12.6.3 JRelay 支持的数据存储方式	196
12.6.4 JRelay 支持的应用程序服务器	196
12.7 SolarMetric 公司的 Kodo JDO 产品	196
12.7.1 关于 SolarMetric 公司	196

12.7.2 关于 Kodo JDO 产品	196
12.7.3 产品家族	197
12.7.4 Kodo JDO 支持的数据存储方式	197
12.7.5 Kodo JDO 支持的应用程序服务器	197
12.8 LiBeLIS 公司的 LiDO 产品	197
12.8.1 关于 LiBeLIS 公司	198
12.8.2 关于 LiDO 产品	198
12.8.3 与 LiBeLIS 相关的永久性描述符的扩展	198
12.8.4 LiBeLIS 公司的产品家族	198
12.8.5 LiDO 支持的数据存储方式	199
12.8.6 LiDO 支持的应用程序服务器	199
12.9 PrismTechnologie 公司的 OpenFusion JDO 产品	200
12.9.1 关于 PrismTechnologie 公司	200
12.9.2 关于 OpenFusion 产品	200
12.9.3 OpenFusion 支持的数据存储方式	201
12.9.4 OpenFusion 支持的应用程序服务器	201
12.10 Orient Technologie 公司的 Orient 产品	201
12.10.1 关于 Orient Technologie 公司	201
12.10.2 关于 Orient 产品	202
12.10.3 Orient 的产品家族	202
12.10.4 Orient 支持的数据存储方式	202
12.10.5 Orient 支持的应用程序服务器	202
12.11 HYWY Software 公司的生产环境系统 PE:J	202
12.11.1 关于 HYWY Software	202
12.11.2 关于 PE:J 产品	203
12.11.3 PE:J 的产品家族	203
12.11.4 PE:J 产品支持的数据存储方式	203
12.11.5 PE:J 支持的应用程序服务器	204
第 13 章 结语	205
13.1 对 JDO 1.0 版技术规范的改进	205
13.1.1 序列 (Sequence)	205
13.1.2 范围 (Extent) 对象的分类	206
13.1.3 JDOQL	206
13.1.4 预先检索模式 (Pre-fetch Pattern)	208
13.1.5 乐观事务	209
13.1.6 标准化的 O—R (对象—关系) 映射	209
13.1.7 事件驱动的“反作用”实例	209
13.2 进一步的信息源	209