

开发专家

之 Sun ONE

J2EE

应用开发 (WebLogic + JBuilder)



飞思科技产品研发中心
赵强 乔新亮

编著



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

开发专家

之 Sun ONE

J2EE

应用开发

(WebLogic+JBuilder)

J2SE+J2EE+J2ME+
Sun
ONE

飞思科技产品研发中心

赵 强 乔新亮

编著



电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING



0781153

57

内 容 简 介

本书介绍如何将最流行的 J2EE 服务器 WebLogic Server 和最好的 Java 集成环境 JBuilder 结合起来开发 J2EE 应用。内容包括 WebLogic Server 的安装、JBuilder 下的集成环境的配置, 开发 Servlet、JSP、数据访问、远程调用技术、JMS 技术和三类 EJB 组件的开发。本书还通过一个规模适度的 J2EE 应用: 网上购书系统, 介绍了基于 J2EE 平台的企业级应用开发的全貌。

流行的服务器+流行的开发环境+流行的计算平台是本书的特色。飞思在线 <http://www.fecit.com.cn> 的“下载专区”提供本书源代码下载。

从环境搭配到高级应用开发, 本书为那些想用 J2EE 开发应用的初学者和中级读者提供了学习的捷径。

未经许可, 不得以任何方式复制或抄袭本书的部分或全部内容。

版权所有, 侵权必究。

图书在版编目(CIP)数据

J2EE 应用开发(WebLogic+JBuilder)/飞思科技产品研发中心, 赵强, 乔新亮编著. —北京: 电子工业出版社, 2003.7

(开发专家之 Sun ONE)

ISBN 7-5053-8856-8

I. J... II. ①飞...②赵...③乔... III. Java 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2003)第 052398 号

责任编辑: 郭 晶

印刷者: 北京市增富印刷有限责任公司

出版发行: 电子工业出版社 <http://www.phei.com.cn>

北京市海淀区万寿路 173 信箱 邮编: 100036

经 销: 各地新华书店

开 本: 787×1092 1/16 印张: 27.75 字数: 710.4 千字

版 次: 2003 年 7 月第 1 版 2003 年 7 月第 1 次印刷

印 数: 5 000 册 定价: 45.00 元

凡购买电子工业出版社的图书, 如有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系。联系电话: (010) 68279077

前 言

关于本丛书

从来没有任何事物像互联网那样，对人类的活动产生如此深刻的影响，无论是政府、企业，以及个人，莫不如此。与此同时，IT 工业也正面临着一场变革——从传统应用向基于 Internet/Web 的服务模式转化。

翻开历史，我们可以看到互联网的形成和发展就是以分布性、开放性和平台无关性为基础的，这是 Internet 与生俱来的属性。随着互联网应用的发展，又引入了诸如 RPC/COM/CORBA 等技术，但这些技术在实际应用中又存在着很多不足和局限。它们的特定协议也难以通过防火墙，因而不适于在 Web 上的应用开发。为了进一步开发基于 Web 的应用，相继出现了 Sun 公司的 Sun ONE (Open Net Environment, 开放网络环境) 和 Microsoft 公司的 .NET 两大 Web 服务技术体系。其中，Sun ONE 以 Java 技术为核心，更接近或者满足于互联网在智能化 Web 服务上对分布性、开放性和平台无关性的要求，同时其在健壮性、安全性、组件化等方面也更为成熟稳定，获得了众多 IT 厂商和产品的支持，是目前惟一在市场上得到了广泛应用的技术体系。

Sun ONE 体系结构以 Java 语言为核心，包括 J2SE/J2EE/J2ME，并基于一系列开放的流行标准、技术及协议。要特别指出的是，Sun ONE 体系结构本身作为开放式体系结构，在得到 IBM/BEA/Oracle/Sybase 等这些 IT 巨擘支持的同时，更得到了互联网上 Open Source 社区的青睐。我们很容易就能从网上免费获得和使用包括 Java 集成开发环境、Java 数据库，甚至是中间件 (Application Server) 服务器等产品，以及它们的源代码。这对于加速国内中小企业的信息化建设和自有知识产权产品开发、提高企业应用和软件行业的整体水平，无疑是难得的机会。

综观国内的技术发展，广大的 Java 程序开发人员及正在转向 Java 体系进行开发的技术人员虽然已面临这一令人激动和鼓舞的转型期，却苦于没有足够的相关资料和文献，尤其对国内的最新 Java 技术动态和技术现状知之甚少，而图书市场上关于 Java 的书籍尽管汗牛充栋，但精品罕见，能反映出 J2EE 及 Sun ONE 的框架全貌的书籍更是奇缺。

电子工业出版社计算机研发部为进一步推动国内 Java 技术的应用与发展，不失时机地推出了“开发专家之 Sun ONE”系列丛书。

本套丛书以 Sun ONE 整体架构为基础，全面体现了 Sun ONE 的技术核心——Java 的应用开发。丛书从各个角度深入 Java 应用开发的各个层面，涵盖了 Java 技术的所有重要思想和实践，体现了最新的 Java 技术进展和动态，可大幅度提升读者的理论和应用水平。同时，丛书重点突出实用性。书中引入了大量的行业应用范例，使读者不仅能快速掌握开发技能，而且对于开发者进行综合系统分析也有所裨益。

关于本书

本书主要讲述如何将最流行的 J2EE 服务器 WebLogic Server 和最好的 Java 集成开发环境 JBuilder 结合起来进行 J2EE 应用程序的开发。JBuilder 作为优秀的 Java 开发工具可以大大简化 J2EE 应用的开发，而 WebLogic Server 是被广泛应用的 J2EE 应用服务器，当前世界上许多大企业都在使用 WebLogic Server 创建大规模的应用程序。

本书需要读者有 Java 语言的基础，但是不需要有 J2EE 和 WebLogic Server 的经验，通过本书给出的带有图注的开发实例，读者可以很快掌握 J2EE 的开发技术。

此外，本书的定位不仅仅是一本 J2EE 开发入门书，本书还讲述了 J2EE 设计模式、框架、网上购书系统、Sun Pet Store 实例分析等内容，供有一定 J2EE 经验的读者使用。

本书的组织结构

本书是按照 J2EE 技术的分类组织的，即每一章讲述一种 J2EE 技术。具体内容如下：

第 1 章 J2EE 概述

本章将对 J2EE 的历史、J2EE 中包含的技术内容及 J2EE 与应用服务器的关系加以介绍，使读者对 J2EE 有一个总体的了解。

第 2 章 认识 WebLogic Server 与 JBuilder

本章将讲述 WebLogic Server 的技术结构、提供的服务、WebLogic Server 7.0 的新特性，以及 JBuilder 8 的新特性、集成开发环境的使用及对 J2EE 开发的支持。

第 3 章 配置 J2EE 开发环境

本章主要讲述如何安装 WebLogic Server 7.0、WebLogic Server 的一些常用操作，以及 JBuilder 中与 J2EE 开发相关的设置（主要有两项：应用服务器设置和数据库设置）。

第 4 章 WebLogic Server 中的 JDBC

JDBC 是 Java 程序使用的与数据库通信的 API，本章将讲述 JDBC 中的主要接口、WebLogic Server 对 JDBC 的支持（例如事务、连接池、数据源），同时给出两个 WebLogic Server 下 JDBC 的开发实例。

第 5 章 基于 WebLogic Server 的 Servlet 开发

本章主要讲述 Servlet 的基本知识及如何使用 JBuilder 开发 Web 应用程序。

第 6 章 基于 WebLogic Server 的 JSP 开发

本章将介绍 JSP 的基本知识及一些高级话题。例如，JavaBean 的使用、MVC 设计模式、Struts 的应用等。每一部分都将给出一个实例，这些实例都是使用 JBuilder 开发的。

第 7 章 RMI 与 JNDI

RMI 与 JNDI 是 J2EE 的基础服务，也是经常使用的服务，例如在 JDBC 开发中获取数据源就是通过 JNDI 完成的。本章主要讲述 RMI、JNDI 的基本原理及编程结构。

第 8 章 开发 JMS 应用

Java Message Service (JMS) 是在 Java 编程中访问企业消息系统的 API。本章将首先讲述 JMS 开发的基本知识, 然后给出一个 WebLogic Server 中 JMS 的配置实例, 最后以一个实例来讲解如何使用 JBuilder 进行 JMS 开发。

第 9 章 开发 JavaMail 应用

J2EE 应用程序有时需要通过 E-mail 与客户进行交互, 例如电子购物网站通过顾客的 E-mail 给用户发送订单确认信息。利用 JavaMail 可以实现 E-mail 功能。本章讲述在 WebLogic Server 环境下如何进行 JavaMail 开发。

第 10 章 WebLogic Server 中的会话 Bean 开发

EJB (Enterprise JavaBean) 是 J2EE 的一部分。通常使用 EJB 组件实现企业的业务规则。本章首先介绍了 EJB 的基础知识, 例如 EJB 的作用、编程模型, 然后重点讲解作为 EJB 的一种类型的会话 Bean, 最后给出两个会话 Bean 开发实例。通过这两个实例, 可以学习如何在 JBuilder 中开发会话 Bean, 以及在 WebLogic Server 环境下的部署。

第 11 章 WebLogic Server 下的实体 Bean 开发

实体 Bean 是代表持久性数据的服务器端组件。本章将讲述以下内容: 实体 Bean 的基本理论; 如何在 JBuilder 中开发实体 Bean; 在实体 Bean 中定义查询 (使用 EJB QL); 建立实体 Bean 关系, 以及通过会话 Bean 访问实体 Bean。每一部分都给出了一个完整的实例。

第 12 章 开发消息驱动 Bean

EJB 2.0 中增加了一种 EJB: 消息驱动 Bean (Message Driven Bean, MDB)。消息驱动 Bean 是一个异步的消息消费者, 客户端不是直接调用 MDB。本章主要介绍 MDB 的基本知识, 并通过一个实例讲解如何使用 JBuilder 开发消息驱动 Bean。

第 13 章 WebLogic Server 的安全性开发

一个企业级应用程序, 特别是要部署到 Internet 上的企业应用程序, 安全性是必须要得到保证的。J2EE 中定义了一个安全性框架, 开发 J2EE 应用程序可以基于这个安全性框架进行安全方面的开发, 以保证应用程序的安全。本章将围绕 WebLogic Server 的安全框架, 通过实例讲述如何保护部署到 WebLogic Server 上的 J2EE 应用程序。为了使没有计算机安全方面基础知识的读者能顺利掌握 J2EE 安全框架, 在本章的 13.1 节给出了计算机安全的基础知识。

第 14 章 J2EE 综合实例

为了使读者能够对于 J2EE 应用程序开发有一个完整的认识, 而不是仅仅停留在一些孤立的知识点上, 本章将给出两个 J2EE 实例, 通过这两个实例, 可以了解如何综合运用所学的 J2EE 技术, 开发出稳定的、易于扩展的、安全的企业应用程序。

本章选用了两个实例, 第一个实例是一个网上购书系统, 第二个实例是 Sun 提供的 Pet Store (应用程序)。此外, 本章还讲述了一些常用的 J2EE 设计模式。

第 15 章 JBuilder 9 简介

本章主要讲解 JBuilder 9 与 J2EE 应用开发相关的一些新特性, 然后通过一个实体 Bean

的开发实例，展示了 JBuilder 9 下的 J2EE 开发。

运行本书中实例所需要的软件环境

如同所有的软件产品一样，WebLogic Server 和 JBuilder 是不断发展的，本书是基于 BEA WebLogic Server 7.0 和 JBuilder 8（企业版）而编写的（本书所讲内容对刚发布的 JBuilder 9 同样适用），读者需要安装以下软件：

- BEA WebLogic Server 7.0。
- JBuilder 8（企业版）。

以下是可选内容：

- 数据库服务器：本书使用的是 JBuilder 8 自带的数据库 JDataStore，如果读者需要使用其他数据库，例如 Oracle，可以自己安装。
- SMTP 兼容的邮件服务器：如果要编写 JavaMail 应用程序的话需要安装邮件服务器。

与我们联系

本书由飞思科技产品研发中心策划并组织编写，赵强、乔新亮主笔，其他参与的人员还有罗春水、蔡庆军、王淼新、韩继宇等人，在此一并表示感谢。本书在编写过程中，编者虽然未敢稍有疏虞，但纰缪和不尽如人意之处仍在所难免，恳请本书的读者提出宝贵的意见或建议，以便修订并使之更加完善。

我们的联系方式如下：

咨询电话：(010) 68134545 68131648

答疑邮件：support@fecit.com.cn

网 址：<http://www.fecit.com.cn> <http://www.fecit.net>

答 疑：<http://www.fecit.com.cn> 的“问题解答”专区

下 载：<http://www.fecit.com.cn> 的“下载专区”

通用网址：计算机图书、FECIT、飞思教育、飞思科技、飞思

飞思科技产品研发中心

目 录

第 1 章 J2EE 概述	1
1.1 J2EE 的历史	1
1.2 J2EE 的优势	2
1.3 J2EE 应用程序的结构	3
1.3.1 J2EE 应用服务器	4
1.3.2 客户层组件	4
1.3.3 Web 层组件	4
1.3.4 业务逻辑层组件	5
1.3.5 企业信息系统层	5
1.4 J2EE 的技术体系	5
1.4.1 JDBC (Java 数据库连接)	5
1.4.2 Servlet 与 JSP	6
1.4.3 JNDI (Java Naming & Directory Interface)	6
1.4.4 RMI (Remote Method Invocation)	7
1.4.5 JMS (Java Message Service)	7
1.4.6 Java Mail	7
1.4.7 EJB	7
1.5 对 J2EE 初学者的建议	8
第 2 章 认识 WebLogic Server 与 JBuilder	9
2.1 认识 WebLogic Server	9
2.1.1 WebLogic Server 简介	9
2.1.2 BEA WebLogic Server 的结构	9
2.1.3 WebLogic Server 提供的服务	10
2.1.4 WebLogic Server 7.0 的新增功能简介	12
2.1.5 如何获取 WebLogic Server 的资源	14
2.2 JBuilder 8 简介	14
2.2.1 JBuilder 8 的新特性	14
2.2.2 集成开发环境介绍	19
2.2.3 JBuilder 8 中的 J2EE 开发	27
第 3 章 配置 J2EE 开发环境	31
3.1 WebLogic Server 7.0 的安装、配置与使用	31
3.1.1 安装 WebLogic Server 7.0	31
3.1.2 WebLogic Server 7.0 的常用操作	43

3.2	JBuilder 8 的 J2EE 开发环境设置	51
3.2.1	应用服务器设置	51
3.2.2	数据库设置	55
第 4 章	WebLogic Server 中的 JDBC	59
4.1	WebLogic Server 支持的 JDBC	59
4.1.1	JDBC 简介	59
4.1.2	JDBC 的结构	60
4.1.3	使用 JDataStore 数据库	63
4.1.4	WebLogic Server 中的 JDBC 配置	67
4.2	WebLogic Server 下的 JDBC 开发实例	80
4.2.1	建立连接	80
4.2.2	执行 SQL 语句	81
4.2.3	处理结果集	81
4.2.4	关闭连接	82
4.2.5	开发实例	82
4.3	WebLogic Server 中的事务处理	86
4.3.1	事务的基本理论	86
4.3.2	JDBC 中的简单事务	90
4.3.3	WebLogic Server 中的事务	91
4.4	JDBC 的高级话题	96
4.4.1	使用 PreparedStatement	96
4.4.2	使用 JDBC 存储和调用大对象	97
4.5	JDBC 开发时的注意事项	107
4.5.1	适时地关闭资源	107
4.5.2	尽可能使查询灵活、准确	108
4.5.3	尽可能快地提交或回滚事务	108
第 5 章	基于 WebLogic Server 的 Servlet 开发	109
5.1	理解 Servlet	109
5.1.1	Servlet 的工作方式	109
5.1.2	何时使用 Servlet	110
5.1.3	Servlet 实现与配置	111
5.1.4	Servlet 的生命周期	114
5.2	Servlet 开发	114
5.2.1	简单的 Servlet 处理 GET 请求	114
5.2.2	使用 Servlet 处理 POST 请求	115
5.3	JBuilder 8 下的 Servlet 开发实例	116
5.3.1	建立 Web 应用程序	117
5.3.2	通过 Servlet Wizard 建立 Servlet	118
5.3.3	部署 Web 应用程序到 WebLogic Server	121

5.3.4	运行 Web 应用程序	123
5.4	WebLogic Server 中 Web 应用程序的配置	124
5.4.1	卸载 Web 应用程序	125
5.4.2	部署 Web 应用程序	125
第 6 章	基于 WebLogic Server 的 JSP 开发	129
6.1	JSP 编程	129
6.1.1	JSP 基础知识	129
6.1.2	JSP 生命周期	134
6.1.3	WebLogic Server 配置	135
6.1.4	JBuilder 下的 JSP 开发实例	142
6.2	高级话题	147
6.2.1	标准动作	147
6.2.2	使用 JavaBean	149
6.2.3	使用自定义标记 (TAGLIB)	155
6.3	Struts 应用开发实例	158
6.3.1	什么是 Struts	159
6.3.2	建立 Struts 工程	160
6.3.3	部署及运行 Struts 工程	167
第 7 章	RMI 与 JNDI	169
7.1	RMI 编程	169
7.1.1	RMI 介绍	169
7.1.2	RMI 结构	169
7.1.3	RMI 开发实例	171
7.2	WebLogic Server 下的 JNDI	173
7.2.1	JNDI 介绍	173
7.2.2	了解 JNDI 接口	174
7.2.3	绑定到 WebLogic Server JNDI TREE	175
7.2.4	JNDI 开发实例	176
7.3	JNDI 与集群	181
第 8 章	开发 JMS 应用	183
8.1	WebLogic Server 中的 JMS 服务	183
8.1.1	JMS 基础	183
8.1.2	JMS 消息	185
8.1.3	持久订阅	190
8.2	在 JMS 中使用事务	191
8.2.1	事务性会话	191
8.2.2	JTA 事务	192
8.3	WebLogic Server 中的 JMS 配置实例	193
8.4	JMS 开发实例	197

8.4.1	使用 JMS 队列的 JMS 实例	197
8.4.2	使用 JMS 主题的 JMS 实例	207
第 9 章	开发 JavaMail 应用	213
9.1	JavaMail 介绍	213
9.1.1	介绍	213
9.1.2	协议	213
9.1.3	关键类	215
9.2	在 WebLogic Server 中配置 JavaMail	219
9.3	JavaMail 开发实例	220
9.3.1	创建 JavaMail 工程	221
9.3.2	使用 JavaMail 发送消息	221
9.3.3	使用 JavaMail 接收消息	222
第 10 章	WebLogic Server 中的会话 Bean 开发	225
10.1	EJB 简介	225
10.1.1	组件及 JavaBeans 简介	226
10.1.2	EJB 技术的设计目标	227
10.1.3	使用 EJB 技术的好处	227
10.1.4	EJB 结构	228
10.1.5	EJB 与 JNDI	228
10.1.6	EJB 与 JTA	229
10.1.7	EJB 的类型	229
10.2	EJB 编程模型	231
10.2.1	Bean 类	231
10.2.2	对外接口	233
10.2.3	Home 接口	234
10.3	部署和使用 EJB 组件	235
10.3.1	部署描述文件	235
10.3.2	EJB-jar 文件	236
10.3.3	使用 EJB 组件	237
10.4	会话 Bean 的基本理论	238
10.5	会话 Bean 开发实例	239
10.5.1	开发无状态会话 Bean	239
10.5.2	有状态会话 Bean 开发实例	257
第 11 章	WebLogic Server 下的实体 Bean 开发	277
11.1	实体 Bean 的基本理论	277
11.1.1	什么是实体 Bean	277
11.1.2	实体 Bean 的子类型	278
11.1.3	实体 Bean 的组成	278
11.2	实体 Bean 开发实例	284

11.2.1	概述.....	284
11.2.2	开发步骤.....	285
11.3	使用 EJB QL 开发查询.....	299
11.3.1	EJB QL.....	299
11.3.2	查询方法开发实例.....	300
11.4	建立实体 Bean 的关系.....	303
11.4.1	实体 Bean 的关系概述.....	303
11.4.2	实体 Bean 关系开发实例.....	304
11.5	通过会话 Bean 访问实体 Bean	310
11.5.1	EJB 引用.....	310
11.5.2	EJB 引用实例.....	311
第 12 章	开发消息驱动 Bean.....	317
12.1	什么是消息驱动 Bean	317
12.1.1	优势.....	317
12.1.2	关于消息.....	317
12.1.3	消息驱动 Bean 的生命周期.....	318
12.2	开发和配置消息驱动 Bean.....	318
12.3	并发控制.....	320
12.4	使用事务.....	320
第 13 章	WebLogic Server 的安全性开发.....	323
13.1	计算机安全的基础知识.....	323
13.1.1	加密/解密.....	323
13.1.2	身份验证.....	324
13.1.3	数字签名.....	324
13.1.4	数字证书.....	325
13.1.5	SSL.....	325
13.2	WebLogic Server 的安全框架.....	325
13.2.1	验证 (Authenticate) 与授权 (Authorization)	325
13.2.2	用户和用户组.....	325
13.2.3	角色 (Role)	326
13.2.4	主体 (Principal)	326
13.2.5	凭证 (Credential)	326
13.2.6	说明性的安全.....	326
13.2.7	编程性安全.....	326
13.2.8	访问控制列表 (ACL)	327
13.2.9	安全区 (Security Realm)	327
13.3	创建安全的 Web 应用程序.....	329
13.3.1	基本验证方式 (Basic Authentication)	329
13.3.2	表单验证方式.....	332

13.4	使用 JAAS 进行安全验证	334
13.4.1	JAAS 简介	335
13.4.2	WebLogic Server 下的 JAAS 开发	335
13.5	EJB 的访问控制	346
第 14 章	J2EE 综合实例	349
14.1	开发网上购书系统	349
14.1.1	需求	349
14.1.2	设计模式的应用	350
14.1.3	网上购书系统设计	359
14.1.4	网上购书系统的实现	361
14.2	Pet Store 简介	395
14.2.1	安装 Pet Store 应用程序	396
14.2.2	Pet Store 的使用	396
14.2.3	获得 Pet Store 的源代码	402
14.3	Pet Store 的设计思想	402
14.3.1	控制模块	404
14.3.2	用户登录和客户注册模块	408
14.3.3	异步消息模块	413
14.3.4	产品目录模块	414
14.3.5	客户模块	416
14.3.6	购物车模块	417
14.4	Pet Store 的源代码分析	418
14.4.1	EJB	418
14.4.2	JSP 页面	419
14.4.3	Pet Store 使用的过滤器 (filter)	420
14.4.4	Pet Store 使用的 xml 文件	420
第 15 章	JBuilder 9 简介	421
15.1	JBuilder 9 的新特性	421
15.1.1	工程管理	421
15.1.2	应用程序生成 (Build)	422
15.1.3	Web 应用程序开发	422
15.1.4	EJB Designer	424
15.1.5	J2EE 应用服务器	424
15.2	JBuilder 9 下的实体 Bean 开发实例	425

第 1 章 J2EE 概述

J2EE 是 Java 2 Platform, Enterprise Edition 的缩写, 它是 Sun 公司提出的一种分布式企业级应用开发的技术架构。目前经过实践的检验, 证明 J2EE 技术是一种可以信赖的企业级软件开发技术, 受到许多 IT 业的大厂商的支持, 具有十分广阔的前景。

本章将介绍 J2EE 的历史、J2EE 中包含的技术内容及 J2EE 与应用服务器的关系, 使读者对 J2EE 有一个总体的了解。

1.1 J2EE 的历史

Sun Microsystems 公司发起了一项称为 Java 2 Platform, Enterprise Edition (J2EE) 的技术创新, 旨在将 Java 平台的范围扩展到大规模服务器环境:

“1997 年 4 月 12 日, Sun 宣布了一项为企业环境开发 Java 平台的创新成果。使用开放式的 Java Community Process, Sun 促进了一组标准的 Java 扩展的开发, 称为 Enterprise Java API。这些应用程序编程接口 (API) 为各种各样的中间件的实现提供了不依赖供应商的编程接口。Enterprise Java API 的要点是 Enterprise JavaBeans API, 后者为 Java 应用程序服务器定义了一个服务器端组件模型, 以及一个不依赖供应商的编程接口。”(见 Thomas 的 “Java 2 Platform, Enterprise Edition: Ensuring Consistency, Portability, and Interoperability”)

J2EE 为 Enterprise JavaBeans 技术提供了工作环境。事实上, Sun 把若干项软件技术都设想为这样的构件块, 它们将使大型企业能够把以任务为关键的业务系统移植到 Java 环境中, 而 Enterprise JavaBeans 技术不过是这些技术之一。EJB 组件是按它们自己的规范定义的, 但 EJB 技术并不是一项独立的技术。它建立在其他 Java 技术之上, 这些技术由 Sun 和其他 IT 公司联合规定, 它们一起提供了这个框架的内容, 该框架就称为 Java 2 Platform, Enterprise Edition。

J2EE 中包括以下技术:

- Enterprise JavaBeans (EJB)
- Java Interface Definition Language (IDL)
- Java Message Service (JMS)
- Java Naming and Directory Interface (JNDI)
- Java Remote Method Invocation (RMI) 和 Object Serialization
- Java Servlet API
- Java Transaction API (JTA)
- Java Transaction Service (JTS)
- Java Server Pages (JSP)
- JDBC 数据库访问 API

参与到这个企业 Java 框架中,并不意味着每项技术都依赖于所有其他技术。单独的规范文档指出每项技术的相关性。例如,Enterprise JavaBeans 规范就指明了在定位各个组件时与 JNDI 的相关性,以及在编程中启动和停止事务处理时与 JTA 的相关性。

1.2 J2EE 的优势

J2EE 是一种利用 Java 2 平台简化企业解决方案的开发、部署和管理复杂问题的体系结构。J2EE 技术的基础就是核心 Java 平台或 Java 2 平台的标准版, J2EE 不仅巩固了标准版中的许多优点,例如“编写一次、随处运行”的特性、方便存取数据库的 JDBC API、CORBA 技术,以及能够在 Internet 应用中保护数据的安全模式等,同时还提供了对 EJB (Enterprise JavaBeans)、Java Servlets API、JSP (Java Server Pages) 及 XML 技术的全面支持。其最终目的就是成为一个能够使企业开发者大幅缩短投放市场时间的体系结构。

J2EE 体系结构提供中间层集成框架用来满足无需太多费用而又需要高可用性、高可靠性及可扩展性的应用的需求。通过提供统一的开发平台, J2EE 降低了开发多层应用的费用和复杂性,同时提供对现有应用程序集成强有力的支持,完全支持 Enterprise JavaBeans,有良好的向导支持打包和部署应用,添加目录支持,增强了安全机制,提高了性能。

J2EE 目前得到了广泛的应用,其优势主要是为搭建具有可伸缩性、灵活性、易维护性的业务系统提供了良好的机制。下面对其优势做一详细叙述。

● 保留现存的 IT 资产

由于企业必须适应新的商业需求,利用已有的企业信息系统方面的投资而不是重新制定全盘方案就变得很重要。这样,一个以渐进的(而不是激进的、全盘否定的)方式建立在已有系统之上的服务器端平台机制是公司所需求的。J2EE 架构可以充分利用用户原有的投资,如一些公司使用的 BEA Tuxedo、IBM CICS、IBM Encina、Inprise VisiBroker 及 Netscape Application Server。这之所以成为可能,是因为 J2EE 拥有广泛的业界支持和一些重要的“企业计算”领域供应商的参与。每一个供应商都对现有的客户提供了不用废弃已有投资即可进入可移植的 J2EE 领域的升级途径。由于基于 J2EE 平台的产品几乎能够在任何操作系统和硬件配置上运行,现有的操作系统和硬件也能被保留使用。

● 高效的开发

J2EE 允许公司把一些通用的、很繁琐的服务器端任务交给中间件供应商去完成,这样开发人员可以集中精力在如何创建业务逻辑上,相应地缩短了开发时间。高级中间件供应商提供以下这些复杂的中间件服务:

- 状态管理服务: 让开发人员写更少的代码,不用关心如何管理状态,这样能够更快地完成程序开发。
- 持续性服务: 让开发人员不用对数据访问逻辑进行编码就能编写应用程序,能生成更轻巧且与数据库无关的应用程序,这种应用程序更易于开发与维护。
- 分布式共享数据对象缓存服务: 让开发人员编制高性能的系统,极大地提高了整体部署的伸缩性。

● 支持异构环境

J2EE 能够开发部署在异构环境中的可移植程序。基于 J2EE 的应用程序不依赖任何特定操作系统、中间件、硬件，因此设计合理的基于 J2EE 的程序只需开发一次就可部署到各种平台。这在典型的异构企业计算环境中是十分关键的。J2EE 标准也允许客户订购与 J2EE 兼容的第三方的现成的组件，把它们部署到异构环境中，节省了由自己制订整个方案所需的费用。

- 可伸缩性

企业必须选择一种服务器端平台，这种平台应能提供极佳的可伸缩性去满足那些在他们系统上进行商业运作的大批新客户。基于 J2EE 平台的应用程序可被部署到各种操作系统上。例如可被部署到高端 UNIX 与大型机系统，这种系统单机可支持 64~256 个处理器。J2EE 领域的供应商提供了更为广泛的负载平衡策略，能消除系统中的瓶颈，允许多台服务器集成部署。这种部署可达数千个处理器，实现可高度伸缩的系统，满足未来商业应用的需要。

- 稳定的可用性

一个服务器端平台必须能全天候运转以满足公司客户、合作伙伴的需要。因为 Internet 是全球化的、无处不在的，即使在夜间按计划停机也可能造成严重损失。若是意外停机，那会有灾难性后果。J2EE 部署到可靠的操作环境中，支持长期的可用性。一些 J2EE 部署在 Windows 环境中，客户也可选择健壮性能更好的操作系统，如 Sun Solaris、IBM OS/390。

正是具有了以上优点，J2EE 才会受到企业客户及软件开发者的青睐，成为当前软件开发领域的一大技术主流。

1.3 J2EE 应用程序的结构

Sun 设计 J2EE 的初衷是为了解决两层模式 (client/server) 的弊端。在传统的两层模式中，客户端担当了过多的角色而显得臃肿。在这种模式中，第一次部署的时候比较容易，但难于升级或改进，可扩展性也不理想，而且经常基于某种专有的协议——通常是某种数据库协议，它使得重用业务逻辑和界面逻辑非常困难。现在，J2EE 的多层企业级应用模型将两层化模型中的不同层面切分成许多层，一个多层化应用能够为不同的服务提供一个独立的层。

J2EE 典型的结构分为 4 层 (如图 1-1 所示)：

- 运行在客户端机器上的客户层组件 (Client tier)。
- 运行在 J2EE 服务器上的 Web 层组件 (Web tier)。
- 运行在 J2EE 服务器上的业务逻辑层组件 (Business tier)。
- 运行在 EIS 服务器上的企业信息系统 (Enterprise information system) 层软件 (EIS tier)。

其中第二层和第三层合称为中间层 (Middle tier)。

下面首先介绍一下 J2EE 应用服务器的概念。

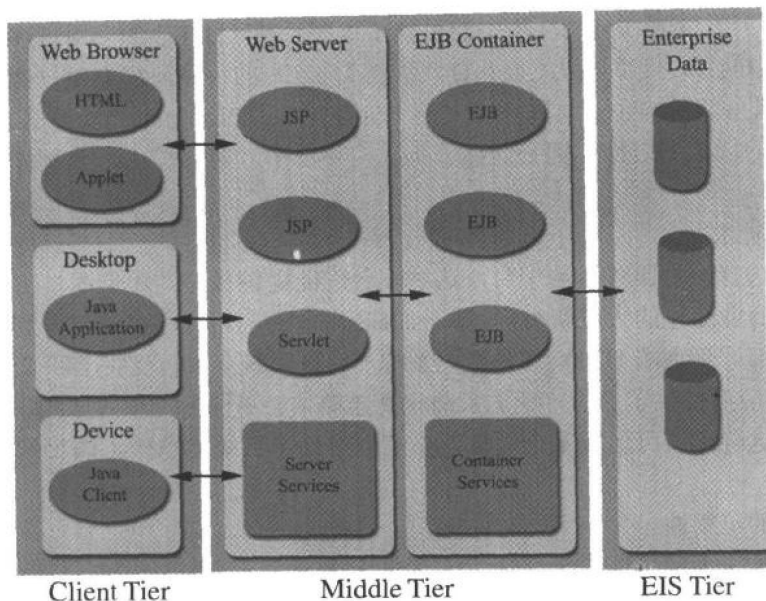


图 1-1 J2EE 的结构

1.3.1 J2EE 应用服务器

J2EE 提供了一个框架（一组标准的 API）用来开发分布式的企业级应用。J2EE 架构中的各项技术需要一个实现者，这个实现者要按照 J2EE 的规范提供 J2EE 应用程序所需的服务，主要是 Web 服务和业务组件服务。J2EE 应用服务器就是充当这个 J2EE 技术标准实现者的角色。目前有许多 J2EE 应用服务器，一些厂商着重于实现 J2EE 体系中的某些组件，例如 Apache 的 Tomcat 为 JSP 和 Servlet 提供支持。BEA 公司的 WebLogic Server 作为 J2EE 应用服务器，实现了 J2EE 的全部特性。J2EE 服务器可以处理一些常见的编程任务，包括提供事务服务、安全认证服务、命名和目录服务、数据库访问和连接池、线程池、负载均衡和容错等。

1.3.2 客户层组件

这一层对应于 J2EE 应用程序的客户端。J2EE 应用程序的客户端可以是基于 Web 方式的（例如 Web 页面、Applet），可以是基于传统方式的（例如 Java 应用程序），也可以是移动设备（例如手机、掌上电脑）。

1.3.3 Web 层组件

J2EE Web 层组件可以是 JSP 页面或 Servlets。按照 J2EE 规范，静态的 HTML 页面和 Applets 不算是 Web 层组件。Web 层可能包含某些 JavaBean 对象处理用户输入，并把输入发送给运行在业务层上的 Enterprise Bean 进行处理。