

J2EE 应用与 IBM WebSphere

Building J2EE Applications with
IBM WebSphere

[美] Dale R. Nilsson 著
Louis E. Mauget
马竹青 鞠文飞 译



Java 技术丛书

J2EE 应用与 IBM WebSphere

Building J2EE Applications with
IBM WebSphere

[美] Dale R. Nilsson 著
Louis E. Mauget

马竹青 鞠文飞 译

電子工業出版社
Publishing House of Electronics Industry
北京 · BEIJING

内 容 简 介

本书通过循序渐进的指导,帮助读者掌握使用 IBM 的 WebSphere Studio v5 创建 J2EE 应用程序。本书的主要内容包括 J2EE 概述、Web 页内容、JavaBean、Servlet、JSP、JavaScript、J2EE 应用程序的部署、数据访问和数据显示、XML、版本控制、EJB 等方面的诸多知识。书中示例翔实,图形丰富,并且详细阐述了创建 Java 应用程序时的设计决策,本书读者可以从非常基础的 Java 应用程序起步,逐渐登堂入室,同时又能领会到应用程序开发的重要设计思路。

本书适合熟悉面向对象编程和 Java 编程语言的读者,读者应当了解编程、软件开发、Web 浏览器使用的基本知识和 Web 功能。已经熟悉 Java 和 J2EE 的读者也能从书中广泛涵盖的 WSAD 开发工具和不同的示例应用程序中获益。

Dale R. Nilsson, Louis E. Mauget: **Building J2EE Applications with IBM WebSphere.**

ISBN 0-471-28157-3

Copyright © 2003, John Wiley & Sons, Inc.

All Rights Reserved. Authorized translation from the English language edition published by John Wiley & Sons, Inc.

No part of this book may be reproduced in any form without the written permission of John Wiley & Sons, Inc.

Simplified Chinese translation edition Copyright © 2004 by John Wiley & Sons, Inc. and Publishing House of Electronics Industry.

本书中文简体字翻译版由 John Wiley & Sons 授予电子工业出版社。未经出版者预先书面许可,不得以任何方式复制或抄袭本书的任何部分。

版权贸易合同登记号 图字: 01-2002-6462

图书在版编目 (CIP) 数据

J2EE 应用与 IBM WebSphere / (美) 尼尔森 (Nilsson, D. R.) 等著; 马竹青等译. - 北京: 电子工业出版社, 2004.8 (Java 技术丛书)

书名原文: Building J2EE Applications with IBM WebSphere

ISBN 7-121-00133-0

I. J... II. ①尼... ②马... III. JAVA 语言 - 程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2004) 第 070203 号

责任编辑: 赵红燕 贺瑞君

印 刷: 北京智力达印刷有限公司

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

经 销: 各地新华书店

开 本: 787 × 1092 1/16 印张: 30.5 字数: 781 千字

印 次: 2004 年 8 月第 1 次印刷

定 价: 49.00 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。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。

前 言

J2EE 架构提供了一种创建 Java 应用程序的一致框架, 用户编写的 J2EE 组件可以运行在任何装有 Java 虚拟机的平台上。J2EE 架构有很多不同的 API, 它们可以组合使用, 使开发者能够组装出“一次写成、到处运行”的应用程序, J2EE 组件可在不同应用程序间共享和重用。J2EE 是一种方兴未艾的新技术, 它建立在开放工业标准的坚实基础上, 很多开发工具(包括 IBM 的 WebSphere Studio 产品)支持 J2EE 架构, 大量的应用程序和 Web 解决方案均基于 J2EE 组件。

J2EE 应用程序可以有多种形式, 从具备高健壮性用户界面的单机运行客户机应用程序, 到通过 Web 界面将实时事务传递到企业服务器的高级多层应用程序。访问 <http://java.sun.com/j2ee> 可以找到大量 J2EE 的示例, 关于 WebSphere 的详细产品信息可以从 <http://www.ibm.com/software/websphere> 获取。

J2EE 架构由 Sun Microsystems 负责管理, 同时也得到很多业界伙伴的积极帮助和支持, IBM 的 WebSphere 产品团队为帮助实现 J2EE 提供了世界一流的工具和运行时服务, 为了支持 J2EE 编程模式, WebSphere v5 在开发和运行时服务方面有了显著的增强, WebSphere 拥有的工具和服务可以帮助用户利用 J2EE API 完成从 applet 到 Web 服务的开发和运行。

Dale Nilsson 和 Lou Mauget 清晰而深入地剖析了 J2EE 架构的精髓, 使之能被大多数程序员轻松地消化、理解, 本书通过大量示例向读者展示了如何使用 J2EE API, 如何利用 WebSphere 的效率和强大功能。

Greg Clark

WebSphere 开发与支持主管

IBM 公司

导 读

Java应用的爆炸性发展,已经使之成为Web开发的标准。IBM的战略开发环境WebSphere Studio Application Developer (WSAD) 于2001年推出,它替代了Visual Age for Java。WSAD版本5具备J2EE兼容性,其众多工具能帮助用户创建大量不同类别的应用程序。

本书通过循序渐进的指导,帮助读者学习如何使用WSAD创建各种各样的示例性J2EE应用程序。书中还提供了实际WSAD产品的众多截屏图,可以让读者在开发Java应用程序的过程中得到直观的视觉反馈。

本书搁笔之际,市面上已经出版了数百种Java书籍,这些书籍大多将焦点集中在某种特定的Java API上,如Servlet, JavaScript和XML等,它们并没有向开发者展示如何在一个应用程序中结合使用这些不同技术。大量Java编程书籍的另一大缺陷是页复一页地列出API代码,仅仅是为了增加页数。本书避免了这些缺点并提供了一个独特的角度,让开发者在构建Java应用程序时通过设计决策来逐步登堂入室。

内容概览

本书提供了一套在真实环境中使用WSAD的实例集合。作者在IBM的应用程序开发软件实验室和WebSphere服务领域工作多年,现在他们作为顾问,帮助客户开发和部署J2EE应用程序。

读者将学习如何使用WSAD设计、开发、调试和部署J2EE应用程序。本书以J2EE技术的良好编程结构和设计标准为指导思想,讲解了如何实现示例应用程序并介绍其设计思路,书中介绍了Web应用程序的部署和配置信息,不过并没有着重讲述WebSphere Application Server的维护。

本书是两名作者真正通力合作的智慧结晶,诸多章节都展现了作者多年积累的知识和经验,为读者提供了全面、广泛的视角,使读者可以通过各种不同的途径利用WSAD的强大功能。书中还提供了从VisualAge for Java和WSAD v4向新版本WSAD进行迁移的信息,作为一本紧跟技术潮流的书籍,书中提供了更多使用WSAD版本5的新的实例和素材。

认证

本书可以帮助读者准备IBM WebSphere认证考试,该认证是JCert Initiative的组成部分。读者将通过创建示例性的Web应用程序学习如何使用WSAD工作台及其工具。使用WSAD进行真正的动手演练可以帮助读者获取WSAD认证测试所需的相关知识。

读者还可以从www.ibm.com/certify和www.jcert.org获取关于不同认证等级和认证需求的更多信息。本书不提供认证题目,也没有给出关于认证测试的任何材料,读者可以在认证Web站点上进行样例测试。

内容组织

书中每章的开始部分是该章的内容提要，结尾为小结部分。书中通过一些小型的应用程序来展示各种 J2EE 应用程序开发主题。本书从简单的 Java 应用程序入手，一步步讲到 J2EE Web 应用程序，最后介绍了 EJB 和 Web 服务。本书的行文从前到后说明逐渐趋于简明，示例应用程序的功能逐渐增加。

读者对象

本书适用于熟悉面向对象编程和 Java 编程语言的读者。阅读本书对编程、软件开发、Web 浏览器的使用和 Web 功能的全面理解非常有益。本书并未专注于介绍 Java 编程，现在学习 Java 的资源很多，比如自学书籍、交互式 CD-ROM 和正式的课程等。

本书旨在向新近接触 WSAD 的读者或希望从这一全能开发平台中挖掘出最大效能的人员提供巨大的帮助。读者将学会如何使用 WSAD 工作台和开发工具，从非常基础的 Java 应用程序起步，到使用 EJB 开发和部署 Web 应用程序。本书提供了很多示例，展示了如何使用 WSAD 构建工具进行正确的面向对象实现。即使是已经熟悉 Java 和 J2EE 的读者也能从本书中的 WSAD 开发工具指南和不同的示例应用程序中获益。

Web 站点的内容

本书的支持网站是 www.wiley.com/compbooks/nilsson，站点上包括了 WSAD v5 测试版的链接，使用这一版本可以完成本书中所有的示例应用程序，不过更妥当的做法是取得该产品的使用授权。关于 Web 站点上可安装程序组件的最新信息以及最近新闻，请阅读此 Web 站点上的 readme.txt 文件。

大多数章节中列举了实例解决方案的 Java 源代码，这些文件可从此站点下载，用于帮助读者解决问题或者输入代码。

网站上包括了本书中所有示例程序的代码。这些应用程序包括 Java 代码和其他 Web 资源，如 HTML 文件、JSP 文件、.gif 图形文件和属性文件。如果读者已经安装了 WSAD，仍需从此站点下载示例文件，以使书中的某些项目正常运作。

小结

WSAD 是一种具备大量特性和功能、紧密集成的强大开发环境 (IDE)，很多企业已经采用 WebSphere 产品系列进行关键任务的 Java 开发。这种具有丰富功能的 IDE，可能让已经熟悉 Java 开发工具包 (JDK) 中基于文件的编辑器、编译器和调试器的读者有点望而生畏，但本书将帮助读者掌握 WSAD IDE，让读者在编写自己的 Java 解决方案时树立自信，提高效率。衷心希望读者喜欢本书。

目 录

第 1 章 J2EE 概述	1
1.1 本章内容提要	1
1.2 什么是 J2EE	1
1.3 J2EE 运行时	7
1.4 J2EE 不包含什么	9
1.5 小结	9
第 2 章 WSAD 入门	11
2.1 本章内容提要	11
2.2 安装注意事项	11
2.3 工作区设置	12
2.4 使用帮助系统	13
2.5 IDE 基础	15
2.6 编写 Java 代码	18
2.7 小结	25
第 3 章 创建一个简单的 Java 应用程序	26
3.1 本章内容提要	26
3.2 开发一个 Java 应用程序	26
3.3 测试应用程序	38
3.4 更新代码	40
3.5 重新测试应用程序	41
3.6 小结	45
第 4 章 使用 WSAD 进行调试	46
4.1 本章内容提要	46
4.2 调试简介	46
4.3 WSAD 调试器	49
4.4 剪贴簿	59
4.5 远程调试	60
4.6 小结	65
第 5 章 Web 页面内容	66
5.1 本章内容提要	66

5.2	HTML 结构	66
5.3	自定义 HTML 设置	73
5.4	简单表格	74
5.5	测试 HTML 页面	76
5.6	编辑字体	77
5.7	编辑 Web 页面样式	79
5.8	小结	81
第 6 章	链接 Web 页面内容	83
6.1	本章内容提要	83
6.2	添加表格行和列	83
6.3	HTML 链接	86
6.4	构建 HTML 表单	89
6.5	小结	98
第 7 章	建立 JavaBeans	99
7.1	本章内容提要	99
7.2	JavaBeans 基础	99
7.3	Bean 类型	105
7.4	其他 JavaBeans 需求	107
7.5	构建 JavaBeans	109
7.6	小结	117
第 8 章	使用 JUnit 进行单元测试	118
8.1	本章内容提要	118
8.2	单元测试	118
8.3	运行测试组	124
8.4	结论	130
8.5	小结	130
第 9 章	构建 Servlet	131
9.1	本章内容提要	131
9.2	什么是 Servlet	131
9.3	创建第一个 Servlet	136
9.4	过滤器	149
9.5	小结	149
第 10 章	含有 JavaBeans 的 Servlet	150
10.1	本章内容提要	150

10.2	从 Web 页面读参数	150
10.3	使用会话	154
10.4	调试 Servlets	161
10.5	集中引用	166
10.6	小结	170
第 11 章	JavaServer Page	171
11.1	本章内容提要	171
11.2	什么是 JSP	171
11.3	使用 JSP	176
11.4	JSP 隐含对象	188
11.5	调试 JSP	189
11.6	JSP 2.0 的变化	190
11.7	新隐含对象	191
11.8	小结	191
第 12 章	JavaScript	193
12.1	本章内容提要	193
12.2	什么是 JavaScript	193
12.3	Web 应用程序里的 JavaScript	198
12.4	作为 JavaScript 工具的 WSAD	199
12.5	在 WSAD 中创建 JavaScript	203
12.6	JavaScript 应用	204
12.7	小结	209
第 13 章	使用继承	210
13.1	本章内容提要	210
13.2	继承类型	210
13.3	重构	211
13.4	最佳实践	221
13.5	小结	222
第 14 章	美化 Web 页面	223
14.1	本章内容提要	223
14.2	框架集	223
14.3	添加图片	228
14.4	制作导航栏	234
14.5	集成声音	241
14.6	显示天气情况	242

14.7	测试更新的 Web 页面	243
14.8	小结	244
第 15 章	创建 Applet	245
15.1	本章内容提要	245
15.2	Java Applet	245
15.3	Java 插件	246
15.4	使用 Applet 的场合	248
15.5	柱形图 Applet	248
15.6	Web 应用程序中的 Applet	261
15.7	小结	261
第 16 章	J2EE 应用程序部署	262
16.1	本章内容提要	262
16.2	可部署的归档	262
16.3	Web 和企业项目	264
16.4	在 WSAD 中导入 WAR 文件	265
16.5	从 WSAD 中导出 WAR 文件	268
16.6	WSAD Web 应用程序打包	269
16.7	J2EE 打包	270
16.8	从 WSAD 中导出 EAR 文件	275
16.9	在 WAS 中定义企业应用程序	276
16.10	导入到 WAS	278
16.11	在集成开发环境之外运行应用程序	280
16.12	小结	280
第 17 章	Struts	282
17.1	本章内容提要	282
17.2	什么是 Struts	282
17.3	构建一个 Struts 应用程序	284
17.4	测试 Struts	300
17.5	Web 图表工具	301
17.6	小结	302
第 18 章	数据访问	303
18.1	本章内容提要	303
18.2	JDBC 概述	303
18.3	WSAD 数据库工具	311
18.4	导出数据定义	323

18.5	执行 SQL	324
18.6	由 SQL 创建 JavaBean	325
18.7	小结	329
第 19 章	显示数据	330
19.1	本章内容提要	330
19.2	数据客户机	330
19.3	设计选项	331
19.4	模型 1	331
19.5	使用 DB 标记的 JSP 模型 1	337
19.6	JSP 模型 2	341
19.7	增强	351
19.8	小结	355
第 20 章	WSAD 中的 XML	356
20.1	本章内容提要	356
20.2	WSAD XML 工具	356
20.3	基于 XML 的 Web 项目	357
20.4	小结	376
第 21 章	代码版本控制	377
21.1	本章内容提要	377
21.2	独立开发	377
21.3	版本化概述	379
21.4	版本控制系统产品	379
21.5	并行版本系统	380
21.6	WSAD CVS 集成	382
21.7	使用资源库	389
21.8	小结	400
第 22 章	企业 JavaBeans	402
22.1	本章内容提要	402
22.2	EJB 概述	403
22.3	销售项目	403
22.4	客户机应用程序	420
22.5	小结	421
第 23 章	实体 EJB	422
23.1	本章内容提要	422

23.2	事务	422
23.3	自定义 Find 方法	426
23.4	业务层 J2EE 模式	427
23.5	Web 客户机	436
23.6	部署与测试	449
23.7	高级主题	449
23.8	小结	451
第 24 章	Web 服务	452
24.1	本章内容提要	452
24.2	概念	452
24.3	开发 Web 服务	455
24.4	Web 服务浏览器	470
24.5	小结	475

第1章 J2EE 概述

1.1 本章内容提要

过去几年中, J2EE 规范已经被业界广泛认可。目前 J2EE 已经建立了良好的 Java 开发标准, 对 J2EE 概念有基本理解是十分重要的。本书从 J2EE 规范的概述开始, 该规范是本书其余部分的基础。

本章将学习构成 J2EE 的各类组件, 理解 J2EE 这个软件工业术语的含义可使读者更有信心, 读者将了解 J2EE 包含哪些方面而不包含哪些方面。本章将涵盖如下的 J2EE 概念:

架构——J2EE 应用程序的基本构建模块。

J2EE API——哪些 Java API 和 API 版本是 J2EE 的组成部分。

打包——运行时组织 Java 软件的标准一致性方法。

角色——在软件开发过程中, 对各参与者的正式标准定义。

安全——J2EE 应用程序所需服务类型的描述及其工作原理。

对所有 J2EE 主题的讨论都将有助于理解 Java 开发中的词汇和术语。

1.2 什么是 J2EE

Sun 公司的 J2EE 规范长达 174 页, 它还引用了其他 Java 规范和 API。J2EE 的简单定义是: 一个庞大、复杂的 Java 架构。不过这对 J2EE 可不够公平, J2EE 的更恰当描述是: 一组开发客户机/服务器 Web 应用程序的 Java API 集合。假如想浏览完整的 J2EE 规范, 可以到 Java 网站 <http://java.sun.com/j2ee/download.html> 下载。

Java 的发展历经反复。初始的 Java 开发工具 (JDK) 具备基本类, 在很大程度上是面向客户机/服务器编程的。随着初始 Java API 更加成熟和可靠, 许多开发者编写了大量的 Java 程序并使其产品化。Java 运行时环境的目标是保持向后兼容, 即运行在某个 API 版本上的应用程序应该也能运行在它之后的版本上, 反之则不行, 即 Java API 不能向前兼容。

随着 Internet 的流行与普及, Sun 公司不断改进 Java 及其 API, 增加了许多 Web 接口, 为更好地进行面向对象设计而重新设计了 API, 这导致了某些 Java API 与其他 API 不兼容的情形, 一些稳定的 API 也要修正错误, 而新的 API 不断变化且不是很稳定。使用 Java 的开发者和公司需要得到指导来确定哪些 Java API 是稳定和兼容的。

Sun 公司决心涤荡混沌重建秩序, 为此定义了一组 Java API 及其版本, 称为 Java 2 企业版 (J2EE), 实际上就是 JDK v1.2。后来 J2EE API 升级到更高版本 1.3 和 1.4。1.4 版本基于 1.3 版本, 它在 Java 优化编程方面没有根本改变。本章结束部分将阐述 J2EE 未来可能的改进。

1.2.1 使用 J2EE 的理由

尽管 J2EE 标准初看起来有点复杂，实际上用它开发真实环境的 Java 应用程序却非常简便。我们不能说它可以使 Java 开发更容易，因为存在着大量不同的 Java API，且 J2EE 架构是将个别应用程序代码嵌入到分散的单元中，使其遵循模块-视图-控制器（MVC）架构。J2EE 的优点如下：

硬件独立——Java 语言号称“一次写成，到处运行”，指 J2EE 可以运行在不同的硬件或软件平台上。

代码分离——J2EE 的设计能够清晰地分割业务、视图和控制器对象，这种分割允许一组开发人员同一时间分别开发不同的部分。

兼容性强——可用的 Java API 非常多，J2EE 规范列出了 Java API 的具体版本来保证兼容性。

可扩展性好——J2EE 架构可以用来构建部署在多个服务器上的嵌入组件。

工具独立——J2EE 是一个含有大量工具 API 的开放性结构，J2EE 的不同部分都有众多的工具厂商提供工具。

J2EE 规范有助于 Java 开发环境走向正常秩序，它统一了众多的 API，为工具开发者和应用程序开发者提供了良好的开发基础。

1.2.2 J2EE 层

J2EE 架构旨在同时解决应用客户机和 Web 结构的问题，J2EE 被设计为处理多个层，此结构有时称为三层结构，包含客户机、服务器和企业组件，如图 1.1 所示。此图是大量复杂系统的简化视图。经常还会存在很多中间层服务器，有时还存在多个后端企业服务器。



图 1.1 J2EE 层

应用程序客户机结构称为“胖客户机结构”，因为它依赖于在客户机上运行代码，此方式适用于高性能的图形界面应用程序，并假定 PC 客户机具有达到性能所需的足够内存和磁盘空间。WebSphere Studio Application Developer（WSAD）就是一个应用程序客户机应用的示例，该程序具有图形用户界面，完全运行在一个系统上。

J2EE 支持的 Web 结构一般被称为“瘦客户机结构”。第 11 章和第 19 章将给出设计 J2EE 模型一和模型二的示例，说明如何使用 J2EE API 建立健壮和可扩展的 Web 应用程序。

1.2.3 J2EE 架构

不同的层使用不同的J2EE组件。J2EE的重点放在运行于J2EE服务器或与此服务器通信的组件上。图 1.2 显示了三个 J2EE 层：客户机层、J2EE 服务器层和企业层。每层含有特定的 J2EE API，来保证开发出开放、可扩展的 Web 应用程序。下一节将阐述各种必需和可选的 J2EE API。

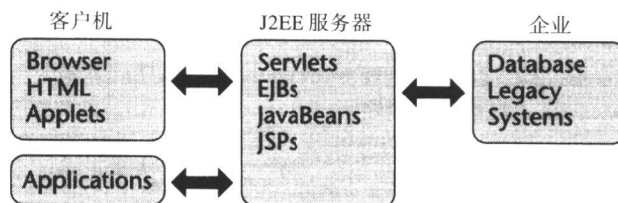


图 1.2 J2EE 层

1.2.4 J2EE API

Java 有很多不同版本的 API，有一些 Java API 版本是不兼容的，J2EE 规范指明哪些 API 版本可以共同工作，它列出了 API 版本的清单以保证兼容性。J2EE API 也被称为 J2EE 服务，所有的 J2EE 服务器都必须支持这些 API。IBM WebSphere Application Server v4 是最先通过 J2EE 认证的应用服务器之一。

J2EE API 分为两大类：必需 API 和可选 API。必需 API 是最基本的，如果读者进行过很多 Java 编程工作就可能认为使用它们是理所当然的。必需 API 十分重要，因为它们具有很多 J2EE 应用程序中需要的通用特性。应用程序必须支持这些 API 才能符合 J2EE 规范。“必需 API”小节里列出了它们的清单。可选 API 是不同 API 的集合，可以解决特定应用程序的问题。

必需 API

符合 J2EE 规范的应用程序必须使用表 1.1 中列出的必需 API：

表 1.1 符合 J2EE 应用程序兼容性要求的必需 API

API	描述
HTTP 和 HTTPS	HTTP 和 HTTPS 是 J2EE 的必需协议，尽管有许多不同的协议，但为了使用 Web 浏览器进行查看，支持 HTTP 是最基本的
Java 接口定义语言 (IDL)	IDL 定义了 CORBA 对象的 Java 接口，CORBA 由对象管理组 (OMG，参见站点 www.omg.org) 管理
Java 数据库连接 (JDBC) 核心	JDBC 是 DB2 和 Oracle 等关系数据库的 Java 接口，它出现已经有一段时间，参见第 18 章的示例
远程方法调用 - 因特网和对象关系数据库间协议 (RMI-IIOP)	RMI-IIOP 用一个对象调用运行于另一台机器上的某个对象的方法
Java 命名与目录接口 (JNDI)	JNDI 具有两个功能：服务提供者使用 JNDI 关联命名和服务目录提供者，应用组件提供者使用 JNDI 访问命名和目录服务

可选 API

可选 API 的数量远远超过必需 API，其中某些 API 也许从不需要使用。一些 API 相当新，可能还会更新，其他 API 已经很成熟，并在今后只会有很小的改变。表 1.2 给出了 J2EE 可选 API 的清单，以及每个 API 的简单描述。

表 1.2 符合 J2EE 应用程序兼容性要求的可选 API

Java 数据库连接 (JDBC) v2	JDBC 的更新, 允许在一个结果集和新数据类型中向后滚动, 也可以使用 Java 方法来更新数据库
EJB v2.	EJB 被设计为可以封装存有数据和业务规则的业务对象。EJB 是一项关键的 J2EE 技术, 参见第 22 章和第 23 章。EJB 与 JavaBeans 类似, 但工作在分布式环境中
Servlet v2.3	这是 Servlet API 的更新版本, 它一直在服务器编程中担当重任, 请参阅第 9 章和第 10 章。Servlet API 提供用来处理客户机请求的服务器端功能
JSP v1.2	JSP 是 J2EE 主要技术之一, 它具有动态 Web 页面功能。此更新版本允许使用标记库, 详细信息请参见第 11 章。JSP 提供动态产生响应的方法
Java 消息服务 (JMS) v1.0	JMS 是一个允许点对点消息传送、发布 - 订阅消息的 API
Java 事务 API (JTA) v1.0	EJB 调用 JTA, 它们被用来对容量所用的 EJB 事务进行事务划分
JavaMail v1.2	简化实现 E-mail 消息和文件传输的 API
JavaBean 激活框架 (JAF) v1.0	JavaMail API 使用 JAF, JAF 支持 MIME 类型
用于 XML 匹配的 Java API (JAXP) v1.1	JAXP 为 SAX 和 DOM 编码提供接口, 它也支持 XSLT
J2EE 连接器结构 v1.0	这在企业层 (EIS) 事务中称为 SPI (服务提供者接口)
Java 认证和授权服务 (JAAS) v1.0	JAAS 提供接口以支持基于用户的安全性

表 1.2 中所有的 API 都有其独立的文档和示例。它们可能看起来太多, 最佳策略是一次学习一种 API, 尽管它们都能一起使用。在编程过程中, 虽然不会使用全部的 API, 也应当知道它们属于 J2EE, 当需要时就可以调用它们。犹如没有必要重新发明汽车轮子一样, 我们可以尽可能地利用 J2EE 服务器提供的服务。

1.2.5 打包

了解 Java 应用程序如何组织和打包是十分重要的。在 J2EE 规范形成前, Java 应用程序的打包方法受到应用程序服务器提供商或开发者使用约定的影响, 打包缺乏一致性, 导致已部署的应用程序的使用和维护遇到了大量困难, 例如, 切换应用服务器就有困难, 让相同的应用程序运行在不同的应用服务器上也同样如此。为了解决此类问题, J2EE 中增加了打包要求。工具开发商已经为诸如 WebSphere Application Server 和 WebSphere Studio Application Developer 这样的平台上的打包约定提供了支持。

开发 J2EE 应用程序时需要定义一个企业归档 (Enterprise Archive) 文件 (EAR), 用于包含应用程序的信息。J2EE 打包约定在分离的包中管理包含 HTML 和 JSP 文件的 Web 内容部分以及 EJB 部分, 在第 22 章和第 23 章中读者将学到更多 EJB 的知识, 现在需要了解的是, EJB 是在 Web 内容文件之外被单独打包和管理的。由 Servlet, JavaBean, JSP 和静态 Web 内容构成的 Web 内容及其部署描述符被打包在一个 Web 归档文件 (WAR) 中, 如图 1.3 所示。EJB 文件及其部署描述符被放置在独立的 JAR (Java 归档) 文件中。应用程序的部署配置是十分灵活的, 它不但能管理 Web 内容, 也能管理 EJB 内容, 这种设计允许在一个企业归档中组合多个 EJB JAR 文件或者多个 WAR 文件。

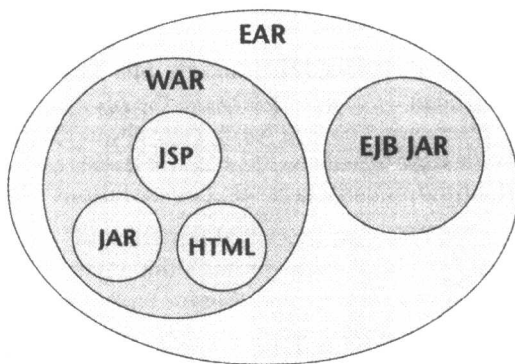


图 1.3 J2EE 打包

EAR, WAR 和 JAR 文件类似于常用的压缩文件, 它将多个文件压缩在一起, 并保留原文件路径结构。这些文件可以在基于文件的版本系统 (例如 PVCS, CVS 或 ClearCase) 中版本化管理, 多数公司依靠这类版本系统部署商业软件, 以便维护已投入使用的系统。

EAR 文件打包起来后会被部署到 J2EE 服务器上。在 WebSphere AppServer 服务器上部署 EAR 文件时, EAR 和 WAR 里面的文件也会在 J2EE 服务器上展开或解压缩。进入到文件系统就能够看到这些文件。读者将从第 16 章学到更多关于部署以及如何在 WebSphere 中进行部署的知识。新的打包约定使部署新的 Java 应用程序更加便捷, 但是在现有的应用程序中却有不少问题, 许多应用程序产品使用了早期 API 的 JSP, Servlet 或 EJB, 它们的开发者和设计师在部署的时候使用了各种不同的约定, 出于成本方面的考虑, 许多企业反对将运行中的应用程序向 J2EE 打包进行转换。转换到 J2EE 打包意味着以下工作:

- 升级应用服务器
- 可能需要更新其他软件, 如关系数据库
- 学习进行 J2EE 打包的新工具
- 打包应用程序
- 部署应用程序
- 测试应用程序

所有这些任务都需要花费时间, 需要进行协调, 这需要经费。在当前的情形下, 许多管理人员抱定了“不坏就不修”的宗旨, 正因为如此, 要将现有的 Java 应用程序转为 J2EE 打包还须假以时日。

1.2.6 开发角色

软件开发项目有若干个不同的参与者, J2EE 对在 Java 项目中履行各种任务的不同角色给出了正式的描述。对于较小的 Java 项目或个人咨询公司, 可能会是一个人担当所有的角色; 而对于大公司, 这些角色将分配给不同的人, 有时还需要分配到不同的部门。最关键的问题是要确定在开发项目中每个人应该扮演的角色。

在讨论 J2EE 角色的各类文章和书籍中, 角色名常常各不相同或者有重叠。无论是否在项目承担多个角色, 都应该力求统一使用正确的角色名, 以避免混乱。J2EE 规范中描述了 6 个角色, 下面将逐一讨论每种角色。