



J2EE Web Services — The Ultimate Guide

J2EE Web Services

高级编程

(美) Richard Monson-Haefel 著
崔洪斌 王爱民 译



清华大学出版社

J2EE Web Services 高级编程

(美) Richard Monson-Haefel 著

崔洪斌 王爱民 译

清华大学出版社

北 京

Simplified Chinese edition copyright © 2004 by **PEARSON EDUCATION ASIA LIMITED and TSINGHUA UNIVERSITY PRESS.**

Original English language title from Proprietor's edition of the Work.

Original English language title: J2EE Web Services, by Richard Monson-Haefel, Copyright © 2004 by John Wiley & Sons, Inc.

EISBN: 0-321-14618-2

All Rights Reserved.

Published by arrangement with the original publisher, Pearson Education, Inc., publishing as Addison-Wesley.

This edition is authorized for sale only in the People's Republic of China (excluding the Special Administrative Region of Hong Kong and Macao).

本书中文简体翻译版由 **Pearson Education** 授权给清华大学出版社在中国境内(不包括中国香港、澳门特别行政区)出版发行。

北京市版权局著作权合同登记号 图字: 01-2003-8071

版权所有, 翻印必究。举报电话: 010-62782989 13501256678 13801310933

本书封面贴有 **Pearson Education** (培生教育出版集团) 激光防伪标签, 无标签者不得销售。

图书在版编目(CIP)数据

J2EE Web Services 高级编程/(美)蒙松-哈菲(Monson-Haefel, R.)著; 崔洪斌, 王爱民译.

—北京: 清华大学出版社, 2005.4

书名原文: J2EE Web Services

ISBN 7-302-09434-9

I. J… II. ①蒙… ②崔… ③王… III. JAVA 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2004)第 090864 号

出 版 者: 清华大学出版社

<http://www.tup.com.cn>

社 总 机: 010-62770175

地 址: 北京清华大学学研大厦

邮 编: 100084

客户服务: 010-62776969

组稿编辑: 曹 康

文稿编辑: 王 军

封面设计: 康 博

版式设计: 康 博

印 刷 者: 北京市昌平环球印刷厂

装 订 者: 三河市新茂装订有限公司

发 行 者: 新华书店总店北京发行所

开 本: 185×260 印张: 41.25 字数: 1056 千字

版 次: 2005 年 4 月第 1 版 2005 年 4 月第 1 次印刷

书 号: ISBN 7-302-09434-9/TP·6586

印 数: 1~4000

定 价: 79.80 元

目 录

第 1 章 J2EE 1.4 Web Services 概述	1
1.1 J2EE 平台	1
1.2 Web 服务技术	2
1.2.1 WS-I Basic Profile 1.0	4
1.2.2 XML	5
1.2.3 SOAP	5
1.2.4 WSDL	6
1.2.5 UDDI	7
1.3 J2EE Web Services API	7
1.3.1 JAX-RPC	8
1.3.2 SAAJ	8
1.3.3 JAXR	8
1.3.4 JAXP	9
1.4 小结	9

第 I 部分 XML

第 2 章 XML 基础知识	13
2.1 XML 入门	13
2.1.1 XML 文档实例	15
2.1.2 剖析 XML 文档	15
2.1.3 处理 XML 文档	18
2.2 XML 命名空间	20
2.2.1 使用命名空间示例	20
2.2.2 默认命名空间、前缀和限定名称	22
2.3 小结	24
第 3 章 W3C XML 模式语言	25
3.1 XML 模式基础知识	25
3.1.1 Web 服务中 XML 模式优于 DTD 的原因	25
3.1.2 XML 模式文档	26
3.1.3 简单类型	27
3.1.4 复杂类型	31
3.1.5 在模式中声明全局元素	37

3.1.6	限定元素与非限定元素	40
3.1.7	指定模式和确定模式的位置	42
3.2	高级 XML 模式	44
3.2.1	复杂类型的继承性	44
3.2.2	简单类型的继承性	49
3.2.3	列表类型与联合类型	52
3.2.4	匿名类型	54
3.2.5	导入模式和包含模式	55
3.3	小结	59

第 II 部分 SOAP 与 WSDL

第 4 章	SOAP	63
4.1	SOAP 的基本结构	65
4.2	SOAP 命名空间	68
4.3	SOAP 头	73
4.3.1	actor 属性	75
4.3.2	mustUnderstand 属性	78
4.3.3	WS-I 一致性文件头	80
4.3.4	有关文件头的进一步说明	81
4.4	SOAP 体	81
4.5	SOAP 消息传递模式	82
4.5.1	Document/Literal 消息传递模式	82
4.5.2	RPC/Literal 消息传递模式	83
4.5.3	消息传递模式与消息传递交换模式	84
4.5.4	其他消息传递模式	85
4.6	SOAP 错误	85
4.6.1	faultcode 元素	86
4.6.2	faultstring 元素	89
4.6.3	faultactor 元素	90
4.6.4	detail 元素	90
4.6.5	关于错误的进一步说明	91
4.7	SOAP 与 HTTP	92
4.7.1	传输带 HTTP POST 消息的 SOAP	92
4.7.2	HTTP 响应代码	93
4.7.3	进一步说明 HTTP	94
4.8	小结	95
第 5 章	WSDL	96
5.1	WSDL 的基本结构	97

5.2	WSDL 声明: definitions 元素、types 元素和 import 元素	100
5.2.1	XML 声明	100
5.2.2	definitions 元素	100
5.2.3	types 元素	101
5.2.4	import 元素	102
5.3	WSDL 抽象接口: message 元素、portType 元素和 operation 元素	103
5.3.1	message 元素	103
5.3.2	portType 元素	107
5.3.3	operation 元素	108
5.4	WSDL 消息传递交换模式	109
5.4.1	请求/响应消息传递	110
5.4.2	单向消息传递	110
5.4.3	通知和恳求/响应消息传递	111
5.5	WSDL 实现: binding 元素	111
5.6	WSDL 实现: service 元素和 port 元素	121
5.7	WS-I 一致性声明	122
5.8	小结	122

第III部分 UDDI

第 6 章	UDDI 数据结构	125
6.1	businessEntity 结构	127
6.1.1	businessEntity 元素和 businessKey 属性	129
6.1.2	discoveryURL 元素	129
6.1.3	name 元素	130
6.1.4	description 元素	130
6.1.5	contacts 元素	130
6.1.6	businessServices 元素	131
6.1.7	identifierBag 元素	132
6.1.8	categoryBag 元素	133
6.2	businessService 结构和 bindingTemplate 结构	136
6.2.1	businessService 结构	137
6.2.2	bindingTemplate 结构	138
6.3	tModel 结构	140
6.3.1	用于 WSDL 文档的 tModel	141
6.3.2	用作分类法标识符的 tModel	142
6.3.3	uddi-org:types tModel	144
6.3.4	校验 tModel 与非校验 tModel	145
6.3.5	tModel XML 模式	146
6.4	publisherAssertion 结构	149

6.5	UUID 密钥	150
6.6	WS-I 一致性声明	151
6.7	小结	151
第 7 章	UDDI 查询 API	152
7.1	UDDI SOAP 消息传递概述	152
7.2	查询操作	154
7.2.1	查找操作	154
7.2.2	取出操作	170
7.3	小结	176
第 8 章	UDDI 发布 API	178
8.1	操作定义与有效负载	178
8.1.1	授权操作	179
8.1.2	保存操作	182
8.1.3	删除操作	189
8.1.4	取出操作	195
8.2	错误消息	202
8.3	小结	205

第IV部分 JAX-RPC

第 9 章	JAX-RPC 概述	209
9.1	服务器端编程模型	209
9.1.1	JSE	209
9.1.2	Enterprise JavaBean 端点	210
9.2	客户端编程模型	211
9.2.1	生成的占位程序	212
9.2.2	动态代理	214
9.2.3	DII	214
9.3	其他 JAX-RPC 事宜	215
9.4	SAAJ	215
9.5	小结	215
第 10 章	JAX-RPC 服务端点	216
10.1	简单 JSE 示例	216
10.2	JSE 运行时环境	217
10.2.1	Servlet: JSE 的基础	218
10.2.2	JNDI 环境命名上下文	219
10.2.3	ServletEndpointContext 接口与 ServiceLifecycle 接口	223
10.3	多线程与 JSE	230

10.4	小结	231
第 11 章	JAX-RPC EJB 端点	232
11.1	Enterprise JavaBean 基础知识	232
11.1.1	事务简介	232
11.1.2	EJB	234
11.1.3	小结	239
11.2	Enterprise JavaBean Web 服务	239
11.2.1	简单示例	239
11.2.2	EJB 运行时环境	242
11.3	小结	247
第 12 章	JAX-RPC 客户 API	248
12.1	生成的占位程序	248
12.1.1	端点接口	249
12.1.2	生成的占位程序	250
12.1.3	服务接口	251
12.1.4	在 J2EE 中使用生成的占位程序	252
12.2	动态代理	256
12.2.1	使用动态代理	257
12.2.2	深层探索	260
12.3	DII	261
12.3.1	与 WSDL 文档一起使用 DII	261
12.3.2	不通过 WSDL 文档使用 DII	264
12.3.3	与 DII 一起使用单向消息传递	265
12.3.4	JAX-RPC 标准属性与常数	265
12.4	小结	267
第 13 章	SAAJ	269
13.1	简单 SAAJ 示例	270
13.2	创建 SOAP 消息	272
13.2.1	MessageFactory 类	272
13.2.2	SaaJOutputter 类	274
13.2.3	SOAPMessage 类	275
13.3	使用 SOAP 文档	276
13.3.1	SOAPPart 类型与 SOAPEnvelope 类型	277
13.3.2	SOAPFactory 类与 Name 类型	278
13.3.3	SOAPElement 类型	280
13.3.4	Node 类型	282
13.3.5	SOAPHeader 类型	283
13.3.6	SOAPHeaderElement 类型	286

13.3.7	SOAPBody 类型	287
13.3.8	SOAPBodyElement 类型	287
13.3.9	Text 类型	288
13.3.10	SOAPConstants 类	289
13.3.11	SOAPException 类	289
13.3.12	SOAPFactory 类型与 SOAPElement 类型	290
13.4	SOAP 错误	291
13.4.1	SOAPFault 类型	292
13.4.2	Detail 类型	295
13.4.3	SOAPFaultElement 类型	295
13.4.4	DetailEntry 类型	296
13.5	与 SAAJ 一起传输 SOAP 消息	296
13.6	SAAJ 1.2 与 DOM 2	297
13.7	小结	300
第 14 章	消息处理程序	301
14.1	简单示例	301
14.1.1	定义消息处理程序类	301
14.1.2	定义 WSDL 文档	305
14.1.3	生成服务接口与端点接口	306
14.1.4	配置消息处理程序	307
14.1.5	在 J2EE 组件中使用消息处理程序	308
14.2	处理程序链与处理顺序	309
14.2.1	返回值与处理顺序	313
14.2.2	异常与处理顺序	315
14.3	处理程序运行时环境	316
14.3.1	无状态与多线程	316
14.3.2	JNDI 环境命名上下文	317
14.3.3	消息处理程序的生命周期	319
14.3.4	MessageContext 类型	321
14.4	小结	323
第 15 章	将 Java 映射到 WSDL 和 XML	324
15.1	将 WSDL 映射到 Java	324
15.1.1	WSDL 与端点接口	325
15.1.2	声明多个部分	326
15.1.3	定义多个操作	327
15.1.4	单向消息传递	328
15.2	将 XML 映射到 Java	329
15.2.1	XML 模式内置简单类型	329
15.2.2	XML 模式复杂类型	331

15.2.3	数组	334
15.2.4	枚举	334
15.2.5	SOAPElement: 在 Document/Literal 编码中支持非标准类型	336
15.2.6	SOAPElement: xsd:any 元素	341
15.2.7	Nilable 元素	342
15.3	Holder	343
15.3.1	复制传递: IN 参数	344
15.3.2	引用传递: INOUT 参数和 OUT 参数	344
15.3.3	Holder 类: 在 JAX-RPC 中支持 INOUT 参数和 OUT 参数	346
15.3.4	从 WSDL 映射 Holder 类型	347
15.4	错误与 Java 异常	354
15.5	小结	359

第 V 部分 JAXR

第 16 章	JAXR 基础知识	363
16.1	使用 UDDI 测试注册表	363
16.2	与 UDDI 注册表建立连接	364
16.2.1	获得 ConnectionFactory	365
16.2.2	配置 ConnectionFactory	366
16.2.3	连接到 UDDI 注册表	368
16.2.4	对 UDDI 注册表进行身份验证	368
16.2.5	在 J2EE 中获得 JAXR 连接	369
16.3	使用 RegistryService 对象与 BusinessLifeCycleManager 对像	370
16.3.1	RegistryService 接口	372
16.3.2	使用 BusinessLifeCycleManager 对象	374
16.4	BulkResponse 类型	377
16.4.1	BulkResponse 接口	378
16.4.2	在示例中处理 BulkResponse	379
16.5	异常	379
16.6	小结	379
第 17 章	JAXR 商务对象	381
17.1	RegistryObject 接口	382
17.2	Organization 信息对象	385
17.2.1	businessEntity	388
17.2.2	ExternalLinks	389
17.2.3	Name 和 Description	390
17.2.4	User	394
17.2.5	分类	402

17.2.6	外部标识符	411
17.2.7	服务	416
17.3	小结	416
第 18 章	JAXR 技术对象	417
18.1	Service 信息对象与 ServiceBinding 信息对象	417
18.1.1	创建新 Service 对象	419
18.1.2	创建新 ServiceBinding 对象	420
18.1.3	Service 接口	421
18.1.4	ServiceBinding 接口	422
18.2	Concept 信息对象	423
18.2.1	创建新 Concept	425
18.2.2	Concept 接口	428
18.3	SpecificationLink 信息对象	432
18.3.1	使用 Specification 对象	432
18.3.2	SpecificationLink 接口	435
18.4	Association 信息对象	436
18.5	预定义枚举	440
18.5.1	AssociationType 枚举	441
18.5.2	URLType 枚举	441
18.5.3	ExtensibleObject 接口与 Slot 接口	442
18.6	小结	443
第 19 章	JAXR 查询 API 与发布 API	444
19.1	将 JAXR 映射到 UDDI 查询 API	444
19.1.1	使用搜索准则	448
19.1.2	findXXX() 方法	454
19.2	将 JAXR 映射到 UDDI 发布 API	458
19.3	小结	459

第 VI 部分 JAXP

第 20 章	SAX2	463
20.1	用 SAX 解析: XMLReaderFactory 类与 XMLReader 类	464
20.2	ContentHandler 接口与 DefaultHandler 接口	465
20.2.1	startDocument() 方法与 endDocument() 方法	465
20.2.2	startElement() 方法与 endElement() 方法	469
20.2.3	startPrefixMapping() 方法与 endPrefixMapping() 方法	473
20.2.4	characters() 方法	474
20.2.5	其他 ContentHandler 方法	475
20.2.6	其他 SAX2 侦听器接口	476

20.3	用 W3C XML 模式检验	477
20.4	小结	478
第 21 章	DOM 2	479
21.1	用 DOM 解析: DocumentBuilderFactory 类与 DocumentBuilder 类	480
21.2	节点	481
21.2.1	依赖于类型的特性	484
21.2.2	XML 名称方法	486
21.2.3	属性方法	488
21.2.4	所有者(Owner)文档	489
21.2.5	导航节点树	489
21.2.6	用于子对象管理的方法	491
21.2.7	其他方法	491
21.3	构建 DOM 文档	492
21.4	复制节点	498
21.5	小结	500

第Ⅶ部分 部署

第 22 章	J2EE 部署	503
22.1	J2EE 部署过程概述	503
22.2	J2EE Web Services 部署	504
22.2.1	用 J2EE 端点开始 Web 服务	504
22.2.2	用 WSDL 开始 Web 服务	506
22.2.3	JAX-RPC 映射文件	508
22.2.4	用于 J2EE 组件的部署描述器	508
22.3	部署 JSE	509
22.3.1	将 JSE 打包到 WAR 文件	509
22.3.2	web.xml 文件	510
22.4	部署 EJB 端点	515
22.4.1	说明性事务属性	515
22.4.2	说明性安全属性	517
22.5	服务引用	519
22.5.1	service-ref-name 元素	521
22.5.2	service-interface 元素	521
22.5.3	wsdl-file 元素与 service-qname 元素	522
22.5.4	jaxrpc-mapping-file 元素	523
22.5.5	port-component-ref 元素	524
22.5.6	显示元素	525
22.5.7	handler 元素	525

22.6	小结	531
第 23 章	Web 服务描述器	532
23.1	wsdl-file 元素与 wsdl-port 元素	533
23.2	port-component-name 元素	534
23.3	service-endpoint-interface 元素	535
23.4	service-impl-bean 元素	536
23.5	jaxrpc-mapping-file 元素	537
23.6	handler 元素	538
23.6.1	handler-name 元素	539
23.6.2	handler-class 元素	539
23.6.3	init-param 元素	539
23.6.4	soap-header 元素	540
23.6.5	soap-role 元素	541
23.6.6	通过处理程序进行处理	542
23.6.7	显示元素	542
23.7	小结	542
第 24 章	JAX-RPC 映射文件	543
24.1	轻量级 JAX-RPC 映射文件的条件	543
24.2	轻量级映射示例	544
24.3	重量级映射示例	546
24.4	映射文件剖析	549
24.4.1	java-wsdl-mapping 元素	549
24.4.2	package-mapping 元素	550
24.4.3	java-xml-type-mapping 元素	552
24.4.4	exception-mapping 元素	554
24.4.5	service-interface-mapping 元素	556
24.4.6	service-endpoint-interface-mapping 元素	563
24.5	小结	535

第VIII部分 附 录

附录 A	XML DTD	567
附录 B	XML 模式正则表达式	571
B.1	符号集	571
B.2	限定符	572
B.3	其他元符号	572
B.4	应用示例	573
B.4.1	ISBN(国际标准书号)	573

B.4.2	美国邮政编码	574
B.4.3	美国电话号码	574
B.4.4	英国邮政编码	574
附录 C	Base64 编码	575
附录 D	SOAP RPC/Encoded	576
D.1	soap:encodingStyle 特性	576
D.2	操作结构	579
D.3	简单类型	580
D.4	复杂类型	581
D.5	数组类型	582
D.5.1	数组的大小	585
D.5.2	数组的其他功能	586
D.6	引用	586
D.7	小结	589
附录 E	SwA	590
E.1	MIME 基础知识	590
E.2	在 SOAP 中使用 MIME	593
E.3	小结	595
附录 F	SAAJ 附件	597
F.1	JAF	598
F.1.1	DataHandler 类	599
F.1.2	DataContentHandler 类	600
F.1.3	DataSource 类	603
F.2	SAAJ 与 JAF: AttachmentPart 对象	605
F.2.1	数据对象	607
F.2.2	createAttachmentPart() 方法	607
F.2.3	setContent() 方法与 setDataHandler() 方法	608
F.2.4	getContent() 方法	609
F.2.5	MIME 文件头方法	610
F.3	SOAPPart 对象	613
F.3.1	XSLT	615
F.3.2	TrAX	615
F.3.3	使用 StreamSource	615
F.3.4	DOMSource	616
F.3.5	SAXSource	616
F.4	SOAPEnvelope	617
F.4.1	getHeader() 方法、getBody() 方法与 addBody() 方法	617
F.4.2	createName() 方法	617

F.5 小结	618
附录 G JAX-RPC 和 SwA	619
G.1 JAF: DataContentHandler 类型和 DataSource 类型	619
G.2 简单示例	620
G.3 将 MIME 类型映射到 Java	623
G.3.1 multipart/* MIME 类型	624
G.3.2 text/xml 类型与 application/xml MIME 类型	626
G.3.3 用作返回类型、INOUT 和 OUT 参数的 MIME 附件	627
G.4 使用 DataHandler 类型和 DataSource 类型	630
G.5 小结	632
附录 H 在没有 WSDL 文档的情况下使用 JAX-RPC DII	633
参考资料	638

第 1 章 J2EE 1.4 Web Services 概述

简单地说, J2EE Web 服务是关于互操作性的, 读者学习本书时要掌握的最重要的事情大概就是互操作性了。这里说的互操作性不只是指不同 J2EE 应用程序服务器之间的互操作性, 而且还包括 J2EE 服务器与其他各种 Web 服务平台之间的互操作性, 这些平台包括 .NET Framework、Perl、Apache Axis、Python 以及 C++ 等。Web 服务技术使 J2EE 能够与其他各种系统进行真正的互操作, 这是因为 Web 服务技术是与平台无关的技术。换言之, 用于通信的媒介与所使用的编程语言、操作系统以及硬件无关。

J2EE Web 服务互操作性以及通用 Web 服务互操作性的核心是由 WS-I(Web Services Interoperability Organization, Web 服务互操作性组织)发布的 Basic Profile 1.0(简称 BP)。BP 提供了一套用于确定应用程序如何使用公共 Web 服务技术的规则, 从而使每个人均可使用相同的语言。BP 使 Web 服务的互操作性更为实用, 这也是本书要介绍的重要内容。

虽然本书假设读者希望开发符合 WS-I 的 Web 服务, 但只要求 J2EE 1.4 供应商支持 BP, 并不强迫用户使用 BP。事实上, 还要求 J2EE 1.4 Web Services API 支持不符合 BP 的各种技术, 如带附件的 SOAP 消息和 RPC/Encoded 消息传递。本书附录介绍了这两种不符合 BP 的技术。

本章将重点介绍 J2EE 平台的体系结构、Web 服务技术以及 J2EE 1.4 Web Services API。读者在学习本章之前应该先读一下本书的序言, 序言中介绍了有关本书涉及的内容以及本书结构等方面的重要信息。

1.1 J2EE 平台

J2EE(Java 2 Platform, Enterprise Edition, Java 2 平台企业版)规范描述了如何将各种 Enterprise Java API 集成到一个完整的平台上。J2EE 1.4 规范特别说明了应用服务器如何配置和部署 Enterprise JavaBean(EJB)组件、Java Servlet 2.4 组件与 JavaServer Pages(JSP)2.0 组件, 以及如何在运行时管理它们。J2EE 规范还说明了这些服务器端组件如何相互以及与应用服务器与像 JDBC(Java Database Connectivity, Java 数据库连接)和 JMS(Java Message Service, Java 消息服务)这样的资源 API 相互作用。图 1-1 说明了 J2EE 规范支持的各种组件与 API。

图 1-1 省略了许多细节, 但该图确实概要说明了 J2EE 体系结构。EJB 和 servlet/JSP 组件位于它们自己的容器中, 这些容器要管理组件的生命周期、接收和发送消息、事务处理、安全性以及其他服务内容。EJB 和 servlet/JSP 组件可以访问所有的资源 API, 如 Web 服务、J2EE 连接器(包括 JDBC)、JMS、JavaMail、Security、Transactions 以及 J2EE Management API 等。J2EE 的整个平台建立在 J2SE(Java 2 Platform, Standard Edition, Java 2 平台标准版)1.4 版本之上, 并在一个或多个 Java 虚拟机中运行。

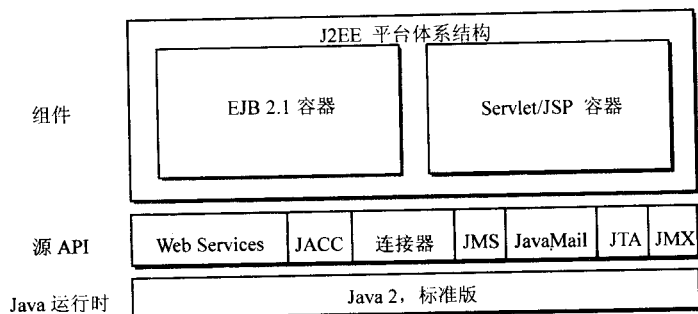


图 1-1 J2EE 平台

J2EE 规范依赖于其他多个规范。例如，在单独的 EJB 规范中描述了 EJB 组件的编程模型与性能；servlet、JSP 组件、J2EE 连接器、JDBC、JMS、JavaMail、JMX 以及其他资源 API 也有类似的单独规范。J2EE 规范的工作就是将所有这些规范联合起来。在本书序言中曾介绍过，本书只详细介绍与 Web 服务有关的 J2EE 组件或 API，因为 J2EE 平台非常庞大。本书将详细介绍 J2EE Web 服务组件与 API。

Web 服务组件与 API 实际上由用于 J2EE 1.1 规范的 Web 服务描述。WS-J2EE 1.1 要求与 WS-I Basic Profile 1.0 相一致。本章后面将介绍 BP。

J2EE 的主要目标是组件以及开发人员技术的便携性。由于每一个 J2EE 供应商必须支持完全相同的服务器端组件模型(EJB、servlet 和 JPS)，因此用户希望为 J2EE 品牌 A 开发的服务器端组件能够在 J2EE 品牌 B 上运行——运行时只需要做很少的工作。但遗憾的是，J2EE 中的组件便携性并不像原来想像的那样简单，因为每一个 J2EE 供应商提供了完全不同的配置和部署功能。因此，J2EE 中的便携性远比 J2EE 之前产品(包括 CORBA 组件和 DCE 组件)中的便捷性简单得多。

除组件便携性外，J2EE 还提供了开发人员便携性。在某一供应商的 J2EE 平台上开发组件的开发人员能够轻易地将所用技术转移到另一供应商提供的平台。当然，开发人员也许还需要学习用于部署和服务器管理的新过程，但是 J2EE 开发人员的技术可以适用于任何牌子的 J2EE。

虽然便携性是 J2EE 的主要内容，但互操作性(特别是某一品牌 J2EE 应用服务器与其他品牌 J2EE 应用服务器通话的能力)近来变得尤为重要。过去，互操作性以 CORBA IIOP 协议为基础，从理论上讲这样做是好的，但实际中却难以正常工作。CORBA 互操作性在处理纯 CORBA 产品时从未体现出特别的优势，而且对 J2EE 产品也继续如此。

然而，随着 J2EE 1.4 的推出，互操作性以 Web 服务的形式进入了一个新时代，这也是本书要介绍的重点内容。本书将重点介绍 4 个主流 Web 服务标准(XML、SOAP、WSDL 和 UDDI)以及 J2EE 1.4 Web Services API(JAX-RPC、SAAJ、SOAP 和 JAXP)。下面的两节将概要介绍 Web 服务、Web 服务标准以及 J2EE Web Services API。

1.2 Web 服务技术

Web 服务这一术语有许多不同版本的定义。传媒和 Web 服务专家给出的一些定义并不很明确，因而也没有多大的用处。但这并不是试图定义 Web 服务这一术语的人的过错，因为他们所描述的某些内容直到现在仍然属于框架概念。