

AUTHORIZED ORACLE PRESS™—EXCLUSIVELY FROM OSBORNE

Oracle9i XML 手册

Oracle9i XML Handbook

Ben Chang
(美) Mark Scardina 著 陈维军 张静贤 郜竹香 等译
Stefan Kiritzov



本书前一版曾在影响力很大的《XML Magazine》
评选当中，荣获“最佳图书”大奖！

OFFICIAL • AUTHORIZED
Oracle Press
ONLY FROM OSBORNE

138OR



机械工业出版社
China Machine Press



OSBORNE

Oracle技术系列丛书

Oracle9i XML手册

Ben Chang

(美) Mark Scardina 著

Stefan Kiritzov

陈维军 张静贤 郜竹香 等译



机械工业出版社
China Machine Press

本书介绍如何在Oracle9i平台上利用XML建立高效的、可靠的互联网应用。主要内容
包括：XML介绍，Oracle9i数据库开发应用程序，利用Oracle的XML使能技术案例分析等
等。并在附录中介绍了W3C XML、DOM、SAX和XSLT规范。

本书内容翔实、实例众多，是一本Oracle XML组件的用户指南。本书所附光盘包括
Oracle9i产品版的Java、C、C++和PL/SQL语言的Oracle XDK及众多演示实例。

Ben Chang, Mark Scardina and Stefan Kiritzov:Oracle9i XML Handbook (ISBN:0-07-
213495-x).

Copyright © 2001 by The McGraw-Hill Companies, Inc.

Original language published by The McGraw-Hill Companies, Inc. All Rights reserved.
No part of this publication may be reproduced or distributed in any means, or stored in a
database or retrieval system, without the prior written permission of the publisher.

Simplified Chinese translation edition jointly published by McGraw-Hill Education (Asia)
Co. and China Machine Press.

本书中文简体字版由美国麦格劳－希尔教育出版公司授权机械工业出版社出版，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。

版权所有，侵权必究。

本书封面贴有McGraw-Hill公司防伪标签，无标签者不得销售。

本书版权登记号：图字：01-2002-0804

图书在版编目（CIP）数据

Oracle9i XML手册/（美）常（Chang, B.）等著；陈维军等译.－北京：机械工业出版社，
2002.3

（Oracle技术系列丛书）

书名原文：Oracle9i XML Handbook

ISBN 7-111-09844-7

I. O… II. ①常… ②陈… III. 关系数据库－数据库管理系统，Oracle9i－技术手册
IV.TP311.138

中国版本图书馆CIP数据核字（2002）第008153号

机械工业出版社（北京市西城区百万庄大街22号 邮政编码 100037）

责任编辑：于杰琼 张鸿斌

北京忠信诚胶印厂印刷·新华书店北京发行所发行

2002年3月第1版第1次印刷

787mm×1092mm 1/16·22.75印张

印数：0 001-4 000册

定价：48.00元（附光盘）

凡购本书，如有倒页、脱页、缺页，由本社发行部调换

译者序

可扩展标记语言（Extensible Markup Language, XML）是一种元标记语言。如同在万维网协会（World Wide Web Consortium, W3C）的XML1.0 规范中所说明的那样，XML可以让用户定义自己的标记语言，从而在XML文件中描述并封装数据。XML的独特之处在于其简单性，并且能够集成内嵌数据提供语义的用户定义的标示标记。XML起源于1986年由国际标准化组织批准的标准通用标记语言。虽然在文档世界中SGML仍然是广为使用的标准，HTML作为万维网上数以万计的Web页面的基础依然用途广泛，但XML仍迅速得到了广泛的认同。因为它与现有的标记语言相比，在数据的交换、存储、描述等方面都具有优势。自从W3C在1998年2月发布了XML1.0版以来，XML已经被广泛认为是电子商务在语言和数据交换方面的重要选择。

ARB把Oracle的XML战略归结为简洁的一句话：“向开发者提供最好的平台以使他们高效地建立和部署可靠的、可伸缩的、利用XML的互联网应用。”很重要的一点是，这个声明提出以发展为核心，而不是以某种产品为核心。从整体上来说这是具有一致性的，因为XML并不是应用，在这一点上它与其他标记语言不同。XML是一种使能技术，正因为如此，从解决方案的上下文才能看到其重要性。不仅仅是Oracle内部的开发组对XML感兴趣，合作者以及Oracle的客户群体提出了借助于XML把分离的系统整合起来的需求，其中大多数这些系统是私有的。XML及其相伴技术XSL非常适于作为提供链接这些系统的基础。

本书由Oracle XML开发组成员编写，详细阐述了Oracle的XML策略、Oracle在XML领域的最新发展以及Oracle所提供的XML使能产品。本书还包含大量与XML和Oracle XML相关的信息和链接，对于那些想借助于Oracle最新产品进行开发和研究的人，以及创建XML使能的应用程序的人而言，都是一本难得的参考书。

现在，我们很荣幸能够有机会承担本书的翻译工作。在翻译过程中，我们经常为一句话、一个术语进行反复的讨论，到处查找资料，力图使本书的翻译能正确、贴切地反映原文的意思，同时注意使句子、段落符合中国人的语言习惯。出版社编辑具有高度的职业精神，在许多方面对翻译工作进行规范，是本书翻译工作得以顺利完成的另外一个关键因素。我们真挚地希望你能够从本书中有所收获，这是作者的初衷，也是我们良好的愿望！

本书由陈维军、张静贤、郇竹香等组织翻译，万方工作室的全体同仁都参加了本书的翻译、校正和输入等工作。具体参加本书翻译、录排、校对工作的其他人员为：陈远春、周昊、龚志翔、万静、李向白、韩冰、张文革、王林、文瑶、王惠英、腾达、谢笑、陈佩、王晋强、易辉、马志宏、林建泉、潘德宏、付文涛、杨建峰、杨晓静、刘坤、胡忻、周鹏、梁晓明、邓超等。本书的出版是集体劳动的结晶，由于时间仓促，且译者经验和水平有限，译文难免有不妥之处，恳请读者批评指正！

2001年11月

作者简介

Ben Chang在Oracle公司工作了12年，在公司里，他负责服务器技术部门中的CORE和XML开发小组的工作。除了负责Oracle6到Oracle9i版本的工作外，他还是Oracle 8.0的版本管理者，并且是拥有该职位最长时间的人，该期间发布了五个版本。他也有3年时间担任Oracle C编码标准委员会主席，在五大洲的许多会议上对XML做过介绍，他也是W3C DOM工作委员会的成员。以前Ben Chang曾在IBM(美国国际商用机器公司)、太平洋Bell公司、Bellcore公司和GE R&D通用电气公司工作过。他持有斯坦福大学的电子工程（计算机系统）理学硕士学位和伯克利学院的电子工程和计算机理学学士学位。

Mark Scardina是Oracle公司XML服务器产品的倡导者，也是CORE和XML开发小组产品经理，负责在整个Oracle产品中建立XML基础构架组件，包括XML开发者工具包。Mark是Oracle公司在W3C XSL工作组的代表。他在Socket通信和ACE技术有多年工作经验，持有旧金山大学的信息系统管理学士学位。

Stefan Kiritzov是Oracle公司的CRM技术与结构（CRM Technology & Architecture）组的开发经理，在开发系统软件方面有19年的经验。在加入Oracle以前，他曾在美国电话电报公司、NCR、SHL Systemhouse和ICT工作过。Stefan Kiritzov持有保加利亚Kliment Ohridsky大学的数学学士学位和硕士学位。

前 言

在1999年3月，Philip Greenspun在Oracle公司总部做了一次讲座，在本次讲座中，他鼓励我们编写本书的第一版。他的讲座涉及因特网、XML、如何创建以数据库为基础的Web站点、为什么Oracle数据库在业界应用如此广泛，以及为什么Oracle开发组成员应该为Oracle出版社多编写图书。他以惯有的调侃和强调方式反复说明了最后一点。

我们把这个任务看作一个不小的挑战，接下来开始了紧张的准备工作，把我们在Oracle上关于XML所做的工作整理了一个提纲，联系Oracle出版社的Jeremy Judson表达了我们的意向，签署了必要的合同，制定了严格的时间进度表，向Philip Greenspun发邮件，请他撰写前言，最后开始了紧张有序的编写工作。我们就本书对问题范围的组织方式多次征询Philip的意见，而不是向技术部门或者管理部门。

回想2000年中期，我们一边在Oracle公司长时间工作，还一边尽力保证本书的进度，终于完成了《Oracle XML Handbook》（该书中译本《Oracle XML开发手册》已由机械工业出版社出版。——编者注）。对于那段经历，我们作者、我们的家庭以及Oracle出版社的编辑们都不会忘记！这段辛勤劳动终于收获了丰硕的果实，该书在全世界销售了成千上万本，被翻译成多种语言，并且获得2000年《XML杂志》的“Best Book”奖，其中在评奖过程中获得XML业界大约5000名读者的投票。我们向ArsDigita Foundation和America's Promise捐献了\$2000。

再回想2001年，我们制定本书第二版的进度表，最终在第二版中把业界最新进展和Oracle9i容纳进去。我们希望读者能够从本书浅显易懂的范例中获益，并且能够学习到一些最新的Oracle XML功能。

XML万岁！

注意 本书的部分销售额会捐献给ArsDigita Foundation和America's Promise以支持他们在提高教育方面所做的工作。他们的网址分别是<http://arsdigita.org>和<http://www.americaspromise.org>。

本书目标

也许你阅读本书的目的是准备在真实的应用程序中使用XML。这非常好，因为XML不是一个应用程序，而是一种使能技术。为了辅助你创建XML使能应用程序，我们在合适的地方提供了许多不同的应用程序场景和特定代码范例，以及范例XML和XSL文件。你可以免费使用这些代码，并且我们希望这些代码能够引导你更有效地使用XML以显著增强自己的应用程序。

注意 如果你想对本书或者光盘部分观点发表自己的意见，或者发现其中的错误，请给我们发电子邮件：oraclexmlhandlebook@yahoo.com。

读者对象

本书的编写目的是提供Oracle XML组件的用户指南。本书不是完整的参考手册，因为所有组件API文档都有电子版，如果本书完全介绍，则篇幅会翻倍。在Oracle技术网上你可以找到参考文档。本书的读者对象是那些开始为程序开发而使用XML的程序员，或者那些对学习如何在Oracle产品中使用XML感兴趣的读者。尽管许多范例使用Java语言，但是借助于组件命令行接口和简单的XML、XSL和XSQL文件的编辑器。你能够了解和认识大多数的XML功能。

如何使用本书

在阅读本书时，你不必按照顺序从头逐章学习。本书的第1章和第2章是XML和Oracle的XML技术的介绍性材料，所以如果对这些知识已经非常熟悉，完全可以跳过去。对于那些创建Oracle9i XML使能数据库应用程序的读者来说，我们建议你仔细学习第3、7、8和9章，最好对第10章也比较重视，以便于深刻理解OTN应用程序。如果你计划使用Oracle应用程序服务器、因特网文件系统或者Oracle Text，则应该阅读第4、5、6章的内容。第10章是我们对XML未来的展望。最后，附录A讨论了部分W3C规范。附录B讨论W3C XML规范，而附录C讨论其他W3C规范。接下来是每章的概述。

第1章更新和引入了在XML业界Oracle所做的工作，解释了XML基本概念和术语、XML存在的原因，以及Oracle在业界的XML策略。该章还提供Oracle技术网及其XML链接、Oracle的XML使能产品和XML组件的概述，以及一个范例，图书销售XML使能应用程序。

第2章更新和引入了Oracle XML开发者工具箱（XDK）的组件，讨论了Java/C/C++/PLSQL XML解析器和XSLT处理器、Java/C++ XML模式处理器、Java/C++类生成器，以及Java TransViewer Beans。该章还包含访问和操作XML/XSL文件的范例代码。

第3章更新和引入了Oracle9i内置的Java虚拟机和Oracle的Java XML组件。该章着重介绍使用XDK的XSQL Servlet如何在数据库中保存XML数据，以及怎样从数据库中检索XML数据，以及图书销售XML使能应用程序的设计和实现。

第4章更新和讨论了Oracle应用程序服务器体系结构，以及到XML和XDK组件的链接。该章还讨论怎样把书店应用程序创建为应用程序Servlet。

第5章更新和引入了Oracle的下一代文件系统（iFS）。该章讨论了体系结构、iFS如何充当XML文件系统，以及iFS如何在内部使用XML以实现把文件的数据映射为关系型模式。该章还包含多个常见文件操作的Java范例代码。

第6章更新和讨论了Oracle9i的文本搜索引擎，即Oracle Text，讨论了其体系结构，以及其搜索存储XML文档的能力。该章扩展了书店应用程序，通过范例代码以创建索引和执行XML使能的搜索。

第7章是新增的，解释了基于XML的服务和事件框架，其中借助于该框架可以开发、部署、管理和执行B2B应用程序集成的基于Web的服务。该章还讨论了基于SOAP的技术，并且解释了Oracle 电子商务应用程序技术是如何使用本框架的。

第8章更新和讨论了在端对端解决方案中如何组装XML组件。该章提供了某XML使能数据

库应用程序的设计，包括支持XML的Web站点和商务-客户消息系统。该章还包含了XML和XSL范例文件和范例代码。

第9章是一个新增的内容，深刻解释了怎样借助于Oracle产品和代码列表来创建功能强大的XML应用程序，即FAQ应用程序。该应用程序演示了如下技术的重要功能：XSLT处理器、XML类生成器、XSQL Servlet、XML SQL 工具、Oracle Text，以及新的XML数据类型和操作。

第10章更新和引入了Oracle技术网（OTN）的XML Web站点，以及其上的XML示范程序。该章还讨论了如何安装和运行这些示范程序，包括特定XML功能的细节，以及每个示范程序所演示的功能。

第11章更新和讨论W3C和OASIS对XML标准所做的工作，以及这些组织今后基于XML的研究方向。该章还讨论了XML及其相关技术当前在XML业界几个主要公司中的使用情况。

附录A更新和讨论了主要的XML规范。还包括对XML文档接口的两个W3C标准（即DOM和SAX）的讨论。讨论了名字空间、XPath、XSL转换的其他相关的XML技术标准。

附录B是新增的内容，从较高层次上概述了新的W3C XML模式规范，并且演示其在多个范例中的使用。

附录C在较高层次上概述了新的W3C XML查询、XML协议和SOAP规范，并且演示了其在多个范例中的使用。

术语表包含一系列XML术语，以及其在本书中的定义。

本书附带的光盘

在本书随带的光盘中，你会找到Oracle9i产品版的Java、C、C++和PL/SQL语言的Oracle XDK，包括XML解析器、XSL处理器、TransViewer Java bean和XSQL Servlet，以及许多演示程序。从OTN站点，你可以下载更新版本的Oracle XDK。

你可以从Oracle出版社Web站点下载本书的所有源代码，地址为<http://www.osborne.com/oracle>。从Oracle技术网Web站点<http://technet.oracle.com/tech/xml>，也许可以下载最新的XML组件和Oracle9i。

目 录

译者序	
作者简介	
前言	
第1章 Oracle 与XML	1
1.1 XML基本概念和术语	1
1.1.1 序言	2
1.1.2 文档类型定义	3
1.1.3 文档的主体	4
1.1.4 文档对象模型API	4
1.1.5 XML简单应用程序接口API	7
1.1.6 名字空间API	11
1.1.7 解析器API	15
1.1.8 可扩展样式表语言变换API	15
1.1.9 XML模式API	15
1.2 为何使用XML	16
1.3 Oracle的XML战略	17
1.4 Oracle在XML产业中的努力	17
1.4.1 Oracle在W3C工作组委员会中的工作	17
1.4.2 Oracle的XML开发者包	18
1.5 Oracle技术网与XML链接	19
1.6 XML使能的Oracle产品概览	22
1.6.1 提供XML API的Oracle产品	22
1.6.2 XML用于数据交换的Oracle产品	23
1.6.3 使用XML进行配置的Oracle产品	23
1.6.4 使用XML进行内容管理和发布的Oracle产品	24
1.7 Oracle XML组件的使用概览	24
1.7.1 文档创建与发布	25
1.7.2 个性化信息发送服务	25
1.7.3 易于自定义的数据驱动应用程序	25
1.7.4 使用XML购物车的电子商务	25
1.7.5 B2B互联网消息机制	25
1.7.6 通过XML消息机制的应用程序集成	25
1.8 实例与应用程序	25
第2章 Oracle XML核心技术	29
2.1 Java V2的XML解析器	29
2.1.1 对SAX的支持	29
2.1.2 对DOM的支持	36
2.1.3 对XSLT的支持	43
2.1.4 对XML模式的支持	47
2.2 Java类生成器	52
2.2.1 输入的DTD	52
2.2.2 处理DTD来生成Java类	53
2.2.3 从Java类创建有效的XML文档	54
2.2.4 由Java应用程序生成的XML文档	55
2.2.5 输入的XML模式	55
2.3 用Java浏览和转换XML	58
2.3.1 DOMBuilder bean	59
2.3.2 XSLTransformer bean	62
2.3.3 XMLSourceView bean	62
2.3.4 XMLTreeView bean	65
2.3.5 XMLTansformPanel bean	66
2.3.6 DBView bean	68
2.3.7 DBAccess bean	68
2.4 PL/SQL版本的XML解析器	68
2.5 C版本的XML解析器和XSLT处理器	72
2.5.1 独立的解析器和集成的XSLT处理器	72
2.5.2 解析器/XSLT 处理器库	73
2.5.3 文档对象模型API	76
2.5.4 XML简单应用程序接口	77
2.5.5 XSLT支持	82

2.5.6 XML模式支持	83	4.3.1 Oracle 数据库客户开发者工具包	119
2.5.7 XML模式处理器库	84	4.3.2 Oracle XML开发者工具包	119
2.6 C++版本的XML解析器、XSLT 处理器及XML模式处理器	84	4.3.3 Oracle LDAP客户工具包	119
2.7 C++类生成器	85	4.4 作为OAS Servlet的Bookstore 应用程序	119
第3章 开发Oracle9i数据库的应用	87	4.4.1 BookstoreServlet 应用	122
3.1 Oracle9i——XML使能的数据库	87	4.4.2 注册BookstoreServlet 应用程序和cartridge	124
3.2 Oracle的JServer和Java XML组件	88	4.4.3 调用BookstoreServlet应用程序	125
3.2.1 JServer基本知识	88	4.4.4 数据库访问	125
3.2.2 Java XML组件	89	4.4.5 使用事务服务	129
3.2.3 发布和调用Java XML组件	91	4.4.6 调用OAS组件	129
3.3 数据库模式与XML文档	93	4.5 作为iAS Servlet的Bookstore应用	130
3.3.1 把XML文档映射到数据库模式	94	4.5.1 配置Apache	130
3.3.2 把数据库模式映射为虚拟XML 文档	96	4.5.2 配置JServ	130
3.3.3 排序和检索XML数据	98	第5章 Oracle因特网文件系统	131
3.4 XSQL——XSLT/SQL服务器页面	99	5.1 特性	131
3.4.1 XSQL页面的体系结构	101	5.1.1 表存储	131
3.4.2 XSQL Servlet的安装	102	5.1.2 解析器	132
3.4.3 SQL查询生成的动态XML文档	102	5.1.3 调整器	132
3.4.4 在XSQL中对条件SQL命令 的支持	104	5.1.4 覆盖	132
3.5 图书销售范例	105	5.1.5 多协议	132
3.5.1 设计数据库模式	105	5.2 优点	132
3.5.2 使用XSQL来设计Web站点	106	5.3 组件	133
第4章 Oracle应用程序服务器开发	111	5.4 XML	133
4.1 Oracle应用程序服务器体系结构	112	5.5 媒介文本/Oracle文本	134
4.1.1 HTTP监听器	113	5.6 文档模型	134
4.1.2 OAS组件	113	5.7 文档属性	135
4.1.3 应用程序cartridge	113	5.7.1 无版本属性	135
4.2 Oracle因特网应用程序服务器体系	114	5.7.2 标准属性	135
4.2.1 iAS 通信服务	115	5.7.3 链接属性	136
4.2.2 iAS 表示服务	116	5.7.4 用户属性	136
4.2.3 iAS 商业逻辑服务	117	5.8 文档处理	136
4.2.4 iAS数据管理服务	118	5.9 定义文档类型	137
4.2.5 iAS 系统服务	118	5.9.1 类型定义范例	137
4.3 iAS客户组件	119	5.9.2 标准类型属性	138
		5.9.3 定制类型属性	139

5.9.4 属性特性	140	7.3.2 Web服务	166
5.10 文件扩展	142	7.3.3 服务组	166
5.11 使用iFS	144	7.3.4 关键集成点	166
5.11.1 例1: 创建并保存HelloWorld	144	7.3.5 调用点	166
5.11.2 例2: 创建一个更好的 Hello World	145	7.3.6 调用记录	167
5.11.3 例3: 操作文件	145	7.3.7 事件	167
5.11.4 例4: 搜索文件	146	7.3.8 事件预定者	167
5.12 使用iFS操作XML文件	147	7.4 XML服务和SOAP	167
5.12.1 存储被解析的XML文件	147	7.4.1 SOAP是什么	167
5.12.2 存储未解析的XML文件	148	7.4.2 SOAP如何工作	168
5.12.3 显示XML文件	148	7.4.3 SOAP客户机做什么	169
5.13 关于XML文件的其他重要事项	149	7.4.4 SOAP服务器做什么	169
第6章 用Oracle Text搜索XML文档	150	7.5 服务组指导方针	170
6.1 下一代文本搜索引擎Oracle Text	150	7.6 服务指导方针	170
6.2 Oracle Text索引结构	151	7.7 安全模型	171
6.2.1 数据存储	151	7.8 服务执行细节	172
6.2.2 过滤器	152	7.9 调用指导方针	172
6.2.3 分段器	152	7.10 服务调用例子	173
6.2.4 词法分析器	154	7.11 事件指导方针	174
6.3 使用Oracle Text	154	7.12 发出事件信号的例子	175
6.3.1 数据存储	156	7.13 给过滤的预定者发出 事件信号的例子	176
6.3.2 字段和区域段	158	7.14 事件预定者服务	178
6.3.3 终止段	161	7.15 部署一种新的服务	178
6.3.4 属性段	161	7.16 创建认证配置	189
6.3.5 PATH_SECTION_GROUP 中的XPATH搜索	163	7.17 创建一个调用记录	190
6.3.6 动态添加段	163	7.18 运行范例服务	192
第7章 Oracle电子商务XML服务	165	7.19 调用和事件API	195
7.1 XML服务综述	165	7.19.1 oracle.apps.jtf.services.invocation. Client类	195
7.2 XML服务组件	165	7.19.2 oracle.apps.jtf.services.invocation. Param类	197
7.2.1 管理员接口	165	7.19.3 oracle.apps.jtf.services.invocation. ServiceResult类	200
7.2.2 SOAP服务器	165	第8章 Oracle和XML应用	202
7.2.3 客户机API	166	8.1 Oracle XML SQL Utility	202
7.2.4 服务和事件库	166	8.1.1 检索XML格式的数据	202
7.3 术语	166		
7.3.1 XML服务	166		

8.1.2 保存XML格式的数据	204	9.7.1 用SYS_XMLGEN 和SYS_XMLAGG生成XML	237
8.1.3 用XML SQL Utility执行 更新操作	206	9.7.2 用Extract()和ExistsNode() 从XMLType中提取数据	237
8.1.4 用XML SQL Utility执行 删除操作	207	9.7.3 用XMLType建立相关FAQ	237
8.1.5 安装XML SQL Utility	208	9.8 创建Web应用程序	240
8.1.6 扩展XML SQL Utility	209	9.9 提交FAQ	241
8.2 Oracle XSQL Servlet	210	9.10 检索FAQ	248
8.2.1 XSQL页面	210	9.11 用HASPETH和INPATH 进行XMLType检索	250
8.2.2 安装XSQL Servlet	211	9.12 利用函数索引提高检索性能	250
8.2.3 向XSQL Servlet提交查询	212	9.13 用URI-Refs直接链接到数据库内容	251
8.2.4 用样式表来转换XSQL输出	213	9.14 建立术语表	251
8.2.5 用XSQL Servlet插入XML文档	215	9.15 扩展应用程序	253
8.2.6 用XSQL Servlet更新数据	216	第10章 OTN提供的基于XML的 应用程序	255
8.3 一个支持XML的Web站点	218	10.1 访问XML应用程序	255
8.3.1 XML使能解决方案	218	10.2 什么是XML应用程序示范	257
8.3.2 设计要求	218	10.3 XML应用程序	257
8.3.3 体系结构	219	10.3.1 Hello World应用程序	257
8.3.4 实现范例	219	10.3.2 Employee Page应用程序	261
8.3.5 扩展范例	221	10.3.3 Insurance Claim应用程序	263
8.3.6 Oracle Portal-to-Go	221	10.3.4 Invalid Classes应用程序	267
8.4 支持XML的电子商务通信	222	10.3.5 XSQL演示程序索引	268
8.4.1 XML使能解决方案	222	10.3.6 Do You XML?站点	270
8.4.2 设计要求	223	10.3.7 Employee/Department对象 视图应用程序	272
8.4.3 体系结构	223	10.3.8 Airport Code Validation 应用程序	272
8.4.4 范例实现	224	10.3.9 Airport Code Display 应用程序	278
8.4.5 扩展本范例	230	10.3.10 Ad Hoc Query Visualization 应用程序	280
8.4.6 Oracle集成服务器	230	10.3.11 XML Document Demo	280
第9章 使用Oracle的XML使能技术 的案例研究	232	10.3.12 XML Insert Request Demo	281
9.1 XML使能的FAQ Web站点	232	10.4 安装和运行XML应用程序	285
9.2 应用程序需求	232	第11章 未来趋势	289
9.3 应用程序设计	233		
9.4 数据库模式设计	233		
9.5 生成XML模式	235		
9.6 生成Java类	236		
9.7 将XML文档存为XMLType	236		

11.1 标准团体的作用	289	模式成果的一个例子	299
11.1.1 W3C的作用	290	11.3 XML对因特网的影响	300
11.1.2 OASIS的作用	295	附录A W3C XML、DOM、SAX和	
11.2 产业范围的XML模式	296	XSLT规范	304
11.2.1 XML 模式/DTD竞技场中		附录B W3C XML模式规范	330
的主要参与者	297	附录C 其他W3C规范	335
11.2.2 产业专用的DTD/XML		术语表	342

第1章 Oracle与XML

可扩展标记语言（Extensible Markup Language，XML）是一种元标记语言。如同在万维网协会（World Wide Web Consortium，W3C）的XML1.0 规范中所说明的那样，XML可以让用户定义自己的标记语言，从而在XML文件中描述并封装数据。这些文件可以在诸如Netscape Navigator和Microsoft Internet Explorer等浏览器中显示，通过因特网在应用程序和业务之间进行交换，或者从数据库中读取数据和写入数据。XML的独特之处在于其简单性，是开放标准的一部分，并且能够集成给内嵌数据提供语义的用户定义的标记。

XML起源于1986年由国际标准化组织（International Standards Organization，ISO）批准的标准通用标记语言（Standard Generalized Markup Language，SGML），基于SGML的超文本标记语言（Hypertext Markup Language，HTML）诞生于1990年。虽然在文档世界中SGML仍然是广为使用的标准，HTML作为万维网上数以万计的Web页面的基础依然用途广泛，但XML仍迅速得到了广泛的认同。因为它与现有的标记语言相比，在数据的交换、存储、描述等方面都具有优势。自从W3C在1998年2月发布了XML1.0版以来，XML已经被广泛认为是电子商务在语言和数据交换方面的重要选择。

1.1 XML基本概念和术语

与其他所有的标准一样，XML同样也有众多的概念和技术术语需要解释。由于开发XML是用来传输数据的，因此我们介绍一个相关的例子，看一下标准数据库中书籍列表的数据记录。一个复杂的SQL查询通常会返回下述格式的数据：

```
History of Interviews, Juan, Smith, 99999-99999, Oracle Press, 2000.
```

如果XML用作输出表单，则该记录的每个数据项都将具有附加的上下文环境，如下所示：

```
<book>
  <title>History of Interviews</title>
  <author>
    <firstname>Juan</firstname>
    <lastname>Smith</lastname>
  </author>
  <ISBN>99999-99999</ISBN>
  <publisher>Oracle Press</publisher>
  <publishyear>2000</publishyear>
  <price type="US">1.00</price>
</book>
```

本例中的一些值得注意的地方稍后将进行详细的讨论。可以注意到文件具有对称性，且每个数据块都被其上下文所包围，形式类似于<context> ... </context>。括弧及其内部文本称为标

记符 (tag), 每组标记符及其包围的数据称为元素。这种关系可以认为类似于数据库中表的一列, 其中标记符的内容相当于列标题, 标记符之间的文本相当于该列中某一行的数据。在上例中, “title” 应该是列名, 而 “History of Interviews” 应当是某一行中的数据。

还可以注意到, 有几个标记符中包含的并非数据, 而是标记符。这是XML一个很重要的特征, 它允许数据的嵌套以更好地定义关系。回到与数据库的类比上来, <author>标记符可以看作一个表, 它的列为<firstname>和<lastname>。在XML的术语中, 这些列标记符被认为是父标记符Author的子标记符。

再看<price>标记符, 可以发现它包含了名字 = “值” 形式的内容。这些 “名-值” 对称为属性, 在开始标记符中可以包含一个或多个这样的属性。但属性出现在结束标记符中是非法的 (例如, </tag name= “foo” >)。

下面是最后一个值得注意的术语, 整个XML实例是被<book>...</book>所包围的, 这种标记符定义为文档的根, 在整个文档中只能有一个。若XML文档遵循了下面的规则: 只有一个根标记符并且所有打开的标记符都被正确关闭, 则称该文档格式正确。

XML的基本概念和术语都很简单, 并且已经形式化为开放的因特网标准。如同W3C的XML1.0规范中所说的那样, “XML文档由称为实体的存储单元组成, 实体中包含解析过或未解析的数据。解析过的数据由字符组成, 其中一些形成字符数据, 另一些形成标记符。标记符对文档的存储布局和逻辑结构进行描述”。XML文档同时具有物理结构和逻辑结构。XML文档的物理结构简单地指向XML文件本身以及可能引入的其他文件, 而XML文档的逻辑结构则指向文档的序言(Prolog)和主体(body)。

book例子中XML代表一个XML文档的主体, 但它缺少了有助于识别其本质的重要信息。该信息在文档的序言中, 在下面一小节中, 我们将对其进行定义。

1.1.1 序言

序言由XML的声明组成, 包括版本号、可能的语言编码、其他属性 (形如名字 = “值” 的对)、可选的文档类型定义 (Document Type Definition, DTD) 等, 其中DTD可以是内部的, 即包括在XML文档中, 也可以是外部的, 即指向另一个文件。例如:

```
<?xml version="1.0"encoding="UTF-8"standalone="no"?>
<!DOCTYPE book SYSTEM"book.dtd">
```

请注意, 形如<?...?>的行是XML处理指令 (processing instruction, PI) 的范例, 将在本章中稍后进行讨论。在本例中, XML处理指令的名字就是xml。另外, 本例中所支持的字符集编码是Unicode的压缩版本, 称为UTF-8。最后的standalone属性是指处理器是否需要包含或引入其他的外部文件。

序言的第二行指DOCTYPE, 在这里XML文档声明数据模型。它为什么这么重要? 请记住, XML文件同时具有物理表示和逻辑表示。在某些应用中, 在不清楚信息是否缺失的条件下, 也足以处理XML。但在大多数的情况下, 应用程序需要验证其收到的XML文档的合法性。这样做, 我们应该需要知道哪些元素是必需的, 哪些可以有子标记, 哪些可以有属性等等。用XML的术语来说, 该数据模型被称为文档类型定义。文档类型定义可以包含在XML文件中, 也可以简单

地顺次引用，这样处理器就可以对其进行定位。

1.1.2 文档类型定义

文档类型定义（Document Type Definition, DTD）是借助于SGML而引入到XML中的，它并不是XML语法的一部分。根据W3C XML模式工作组所做的工作定义了一种利用XML语法的新模型定义，并且扩展了DTD的功能以支持各种数据类型。在上例中，DTD包含在外部文件中。它也可通过统一资源定位器（Uniform Resource Locator, URL）的形式进行说明，类似于<http://www.foobar.com/book.dtd>。DTD还可以在XML的文件内部声明。例如，在XML文件内部，声明类似于下面的代码。

```
<!-- DTD bookcatalog may have a number of book entries -->
<!DOCTYPE bookcatalog [
<!ELEMENT bookcatalog (book)*>
<!-- Each book element has a title, 1 or more authors, etc. -->
<!ELEMENT book (title, author+, ISBN, publisher, publishyear, price)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT author (firstname, lastname)>
<!ELEMENT firstname (#PCDATA)>
<!ELEMENT lastname (#PCDATA)>
<!ELEMENT ISBN (#PCDATA)>
<!ELEMENT publisher (#PCDATA)>
<!ELEMENT publishyear (#PCDATA)>
<!ELEMENT price (#PCDATA)>
<!-- ATTENTION: price type (US|CAN|UK|EURO) #REQUIRED -->
]>
```

在DTD的DOCTYPE声明后面是根元素的声明，bookcatalog的<!ELEMENT>。元素的组成很简单，由开始标记符（例如<foo>）对应的结束标记符（如</foo>）以及两者之间的文本组成。在一个XML文档中，只可以存在一个根元素。根元素标示着文档的开始，并被视为所有其他元素的父元素，其他的元素都嵌套在它的开始标记符和结束标记符之间。根据这个DTD，如果一个XML文档被认为是合法的，其主体开始部分的第一个元素必须是根元素bookcatalog。

接下来是元素的声明，它约定了嵌套在根元素bookcatalog中的子元素必须符合根元素的内容模型。注意到bookcatalog的所有子元素都在它的元素声明中显式列出，而且author有后缀“+”。这是扩展的巴科斯-诺尔范式（Extended Backus-Naur Format, EBNF）的一个实例，可以用来描述内容模型。可用的后缀有：

- ? 出现零次或一次。
- * 出现零次或多次。
- + 出现一次或多次。

还可以注意到，我们使用#PCDATA来声明元素中文本必须是非标示的文本，另外也应该显式声明价格的必需属性值。CDATA与PCDATA的区别在于解析器将简单地忽略CDATA段，且不进行格式检查，因此CDATA可视为“未解析字符数据”。

因此,验证XML解析器通过按照DTD中指定的规则对XML文档解析,尽力确定该文档是否符合DTD(合法),也就是结构化关系和顺序是否一致。

1.1.3 文档的主体

根元素出现在序言之后,它包括了XML文档的剩余部分。这一部分由元素、处理指令、内容、属性、注释、实体引用等组成。正如前面提到的那样,元素必须有开始标记符和对应的结束标记符,而且嵌套的顺序要正确,否则XML文档格式就是不正确的,XML解析器有可能因此而报错。元素也可以具有属性,或“名-值”对,例如`<author firstname = "Juan" lastname = "Smith">`。另外还有由XML 1.0规范所定义的内置属性,例如`xml:space = "preserve"`,表示元素之间的数据用于表示预留的空格。实体引用类似于在实体中定义过的宏,引用它们就可以了。例如`&nameofentity`,就可以在其整个定义域中使用。举例来说,可以声明`<!ENTITY Copyright "Copyright 2000 by Smith, Jones, and Doe - All rights reserved">`;这样,`&Copyright`实体就可以作为快捷方式在XML文档中到处使用。XML解析器必须能够识别在DTD中定义的实体,即使为此也许需要关闭合法性检查。再强调一次,XML1.0规范也定义了内置实体,例如“&”符号——`&`、省略号——`&apos`、小于号——`<`等等。而注释则通过包围在形如`<!-- -->`结构中来识别。

还存在其他一些与XML解析器应用程序编程接口(API)相关的基本术语,可以按照如下领域进行分类:

- 文档对象模型(Document Object Model, DOM)。
- XML简单应用程序接口(Simple API for XML, SAX)。
- 名字空间。
- 解析器。
- 可扩展样式表语言变换(Extensible Stylesheet Language Transformation, XSLT)。
- XML模式。

重要的是,这些API是W3C规定的,所以应用程序开发者可以使用标准的编程接口。请注意,如果有些功能未指定,则供应商也许会单独实现对规范的增强功能。

1.1.4 文档对象模型API

由于开始标记符均有对应的结束标记符,并且这些标记符以有序的方式嵌套,因此XML文档是结构化的。由于其结构,XML文档可视为树结构,其节点由标记符以及处于标记符之间和对应于标记符的信息组成。实际上,当XML解析器解析XML文档时,在内存中会形成该文档的语法表示,称之为文档对象模型(Document Object Model, DOM)。

W3C已经创建了一组DOM API,可用于存取及访问文档树。文档树的组成包括文档的根元素、子节点及兄弟节点、元素、属性、表示元素或属性文本内容的文本结点。其他组件还包括用来标识文本块区分标记的字符数据(CDATA)段、注释、实体引用、处理指令等等。能够提供所有DOM API的XML解析器据说与W3C的DOM推荐标准兼容。

下面的Java代码实例演示了对解析器和DOM API的简单使用。输入到应用程序的XML文档