

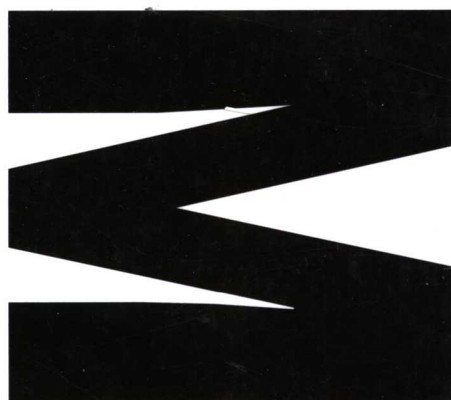
程序天下

知程序 行天下

知程序 行天下

张朝明 等编著

数据标记、处理、共享与分析



开发应用：
开典应

本书的技术要点：

XML Schema XSLT XPath

变量定义 流量控制 EXSLT

XPath的灵魂一轴 JDOM

JAXB XML DOM……



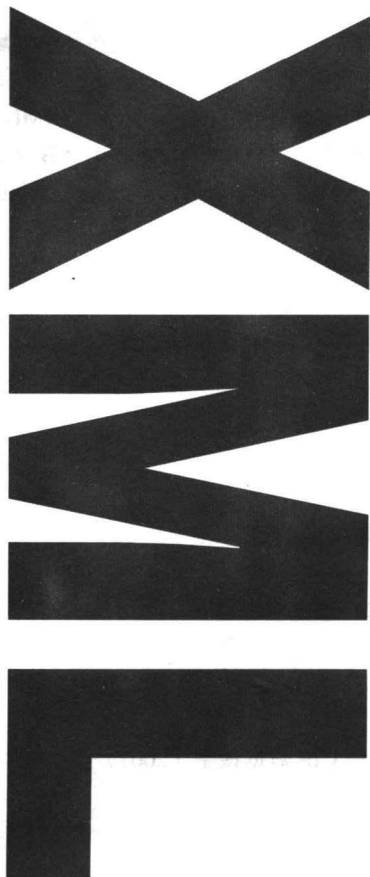
电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn



TP312/2656D

2008

数据标记、处理、共享与分析



开发应用：
开典应

程序天下

张朝明 等编著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

XML 以其强大的发展态势, 迅速在各个领域得到了广泛应用。本书由浅入深、循序渐进地向读者讲述了 XML 的三大技术——XML Schema、XSLT 和 XPath。全书内容包括 XML 基础、XML 架构——XML Schema、XML 样式表转换语言——XSLT、XML 路径语言——XPath, 以及 XSLT 2.0 和 XPath 2.0 的知识, 最后结合 Java 语言, 讲述了 JDOM 和 JAXB 两大应用。

本书既适合 XML 初学者阅读, 也适合专职的 XSLT 程序员、XML 网站开发人员、XML 技术爱好者、数据分析人员作为技术参考资料。

未经许可, 不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有, 侵权必究。

图书在版编目 (CIP) 数据

XML 开发典型应用: 数据标记、处理、共享与分析 / 张朝明等编著. —北京: 电子工业出版社, 2008.1
(程序天下)

ISBN 978-7-121-05369-6

I. X… II. 张… III. 可扩充语言, XML—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2007) 第 176844 号

责任编辑: 高洪霞

印 刷: 北京市通州大中印刷厂

装 订: 三河市鹏成印业有限公司

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

开 本: 787×1092 1/16 印张: 35.25 字数: 872 千字

印 次: 2008 年 1 月第 1 次印刷

印 数: 5000 册 定价: 65.00 (含光盘 1 张)

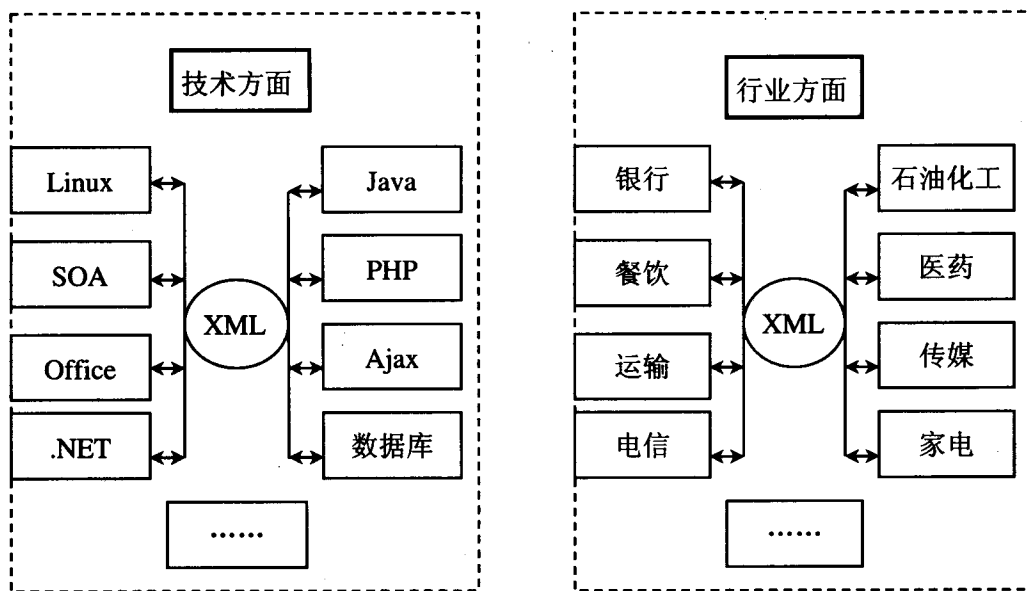
凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

前言：作者的提示

XML 作为国际标准，表现出了强大的生命力。不管是新的技术架构还是软件厂商对 XML 重视都反映了这一点。可以说，XML 是不同技术之间数据交换的基石。下图反映了它的重要程度：



由此可见 XML 功能之强大，应用之广泛。可以说以后任何软件都离不开 XML。对于这么重要的技术，书店里却没有一本很好的入门书籍。笔者在学习的时候尝到了各种苦头。为了帮助大家快速掌握这一强大技术，笔者萌生了写一本书的想法。

本书的特点

1. 配有学习论坛 <http://www.rzchina.net>，读者可以在上面讨论技术，笔者会及时回答读者提问，并提供各种技术文章，帮助读者提高开发水平。

2. 内容翔实，实例丰富：本书就 XML Schema、XSLT 和 XPath 三大技术，进行了详细的讲解，并辅以大量实例，使读者可以边学技术，边进行实践。本书对 XSLT 2.0 和 XPath 2.0 的知识也做了介绍，帮助读者实现从 1.0 到 2.0 的跨越。

3. 注重理解，把握全局：本书对许多模糊的概念进行了详尽的描述，同时记录了许多笔者的技术经验和心得体会，以帮助读者思考问题、解决问题。本书注重从全局上把握知识框架，在讲解宏观知识的同时穿插讲述语法。

4. **注重实际，应用为王：**本书注重实际应用，尤其在讲述 XSLT 1.0 和 XSLT 2.0 的时候，都是从应用的角度出发进行讲解，并将应用模块化，对每个模块都总结了典型实例和应用场合，既方便初级读者进行理解，也方便中、高级读者进行查阅。

5. **分析本质，注重呼应：**XML 技术中很多知识都是相互呼应的。本书在讲解时，注意知识体系的前后呼应。例如在讲解 XML Schema 中复杂类型的时候，首先讲述定义复杂类型，然后讲述复杂类型的默认速记规则，便于读者了解来龙去脉。再比如在讲解 XPath 时，首先讲解了许多 XPath 的知识，最后利用轴这个概念来统一所有的概念和语法。通过类似的讲述方式，使读者不但知其然，而且知其所以然。

6. 光盘中赠送 XML 技术要点学习课件。

本书的内容

本书由 21 章组成，内容分布如下：



关于百家争鸣

众所周知，春秋战国时期的百家争鸣是中国历史上思想和文化最为辉煌灿烂、智慧闪光的时代。诸子百家或为了实现政治抱负，或为了富国强兵，或为了兼济天下，甚至为了富贵荣华，纷纷著书立说，四处游说。而百家之间又相互竞争、相互诘难，思想的火花便被激发了。但反观现在的技术领域，环境宽松，条件优越，却很难出现思想的火花。

笔者认为，百家之所以繁荣，最主要的原因是他们都在说自己的话。

关于写书

首先，写书要保证内容正确。而很多作者可能并没有太多时间去验证技术方面的问题，只好人云亦云，这样既省时省力，又不会出错。另外，大家探讨的比较大的问题，一般也是大家都会遇到的问题，自然重复的几率就会高很多。

最后，写自己的心得，谈自己的经验，出错的几率会比较大。因为一个人的智力有限，思维的宽度和广度也有限，对于某个概念，稍有不慎，在理解上就可能会有偏差。

于是，大家都不说自己的话。我想，这大概也是当前为什么会有那么多雷同内容的书籍出现的原因了吧。

思考才是最重要的

思考有时候是一件痛苦的事情，为了理解一个概念、解决一个问题，绞尽脑汁，可不是一件轻松好玩的事情。但是，一旦理解了、解决了，就会有一种“更上一层楼”的感觉。但是问题在于，很多时候，我们并没有去思考是因为没有人提醒我们该去想哪些问题。因此，笔者觉得，书籍，尤其是技术性书籍，首要目标就是启发读者思考，当然，还要说出自己的理解，以便读者分析对比。这样才能真正对读者产生帮助。字典式的语法书就像一杯白开水，读者喝了，没有什么感觉，很快也就忘掉了，根本不能起到提高技术水平的目的。能引发读者思考的图书则是五味俱全，它能让读者尝尽酸甜苦辣，真正提高读者对技术的理解和体会。

关于本书

笔者在编写本书时，想到网友的评论，总是战战兢兢，如履薄冰。因此笔者还是决定逆流而上——说自己的话。

本书是笔者在工作中的经验总结和学习心得，并融合了在网上解答的网友提问。在本书中，读者会发现笔者将概念的理解看得很重要，而将语法的讲解放在了次要的位置。尽管本书可能存在着这样那样的不足，但是笔者仍然决定将自己的想法和理解说出来，以引

发读者的思考。当然，这些理解都经过了笔者仔细求证。

超值光盘内容

内容详见附录 A、附录 B。关心应用本书实例代码的读者请详细阅读。

适合的读者

- 专职的 XSLT 程序员
- XML 技术爱好者
- 程序开发人员
- 基于 XML 的网站开发者
- 数据库管理人员 DBA
- 数据分析人员

编 者

2007 年 11 月

目 录

第一篇 XML 简介

第 1 章 XML 概述	2
1.1 什么是 XML	3
1.2 比较 XML 和 HTML	4
1.3 XML 的用途	4
1.4 本章小结	4
第 2 章 XML 基础知识	5
2.1 XML 语法	6
2.1.1 理解 XML 文档的本质	6
2.1.2 文档类型声明	7
2.1.3 处理指令	8
2.1.4 文档注释	9
2.1.5 元素	10
2.1.6 属性	12
2.1.7 实体引用	14
2.1.8 CDATA 段	15
2.1.9 字符引用	16
2.2 XML 文件	16
2.2.1 XML 开发工具	16
2.2.2 如何书写 XML 文档	17
2.2.3 格式良好和有效的 XML 文档	19
2.2.4 我的 XML 出了什么错	20
2.2.5 本节小结	23
2.3 开发前奏: XML 和 Schema, XSLT 的关系	23
2.3.1 什么是 Schema 和 XSLT	23
2.3.2 没有 Schema 和 XSLT, XML 世界将变成什么样子	24
2.4 本章小结	24

第二篇 XML 模式——Schema

第 3 章 浅析 Schema 开发	26
3.1 Schema 简介	27
3.1.1 Schema 的本质	27
3.1.2 比较 Schema 和 DTD	27
3.2 使用 Schema 验证 XML 文档	29
3.2.1 分析一个 Schema	29
3.2.2 用 Schema 验证 XML 文档有效性	30
3.3 名称空间	31
3.3.1 什么是名称空间	31
3.3.2 如何理解名称空间	33
3.3.3 无名称空间和默认名称空间	34
3.3.4 名称空间的作用范围	35
3.3.5 本节小结	37
3.4 重新分析 Schema 文件	38
3.5 Schema 内置数据类型	39
3.5.1 xs:string (字符型)	39
3.5.2 xs:decimal (小数型)	39
3.5.3 xs:integer (整型)	40
3.5.4 xs:boolean (布尔型)	40
3.5.5 xs:date (日期型)	41
3.5.6 xs:time (时间型)	41
3.5.7 xs:dateTime (日期时间型)	41
3.5.8 xs:gYearMonth (特定年中的特定月)	42
3.5.9 xs:gMonthDay (每年都会出现的某一天)	42
3.5.10 xs:gDay (每月都会出现的某一天)	42
3.5.11 xs:gMonth (每年都会出现的某个月份)	43

3.5.12	xs:timeDuration (时间段)	43	5.1.2	区分 simpleContent 和 complexContent	70
3.5.13	xs:anyURI (统一资源标识符)	43	5.2	声明 simpleContent 元素	71
3.6	XML 元素类型	43	5.2.1	如何声明 simpleContent 元素	71
3.7	本章小结	44	5.2.2	关于类型派生——restriction 和 extension	73
第 4 章	Schema 声明简单元素	45	5.2.3	属性声明	74
4.1	基本数据类型的使用	46	5.2.4	使用 use 定义属性	76
4.1.1	字符型的使用	46	5.2.5	使用 default 给属性赋予默认值	78
4.1.2	日期型的使用	46	5.2.6	使用 fixed 限制属性	80
4.1.3	数值型的使用	47	5.3	声明 complexContent 元素	81
4.1.4	布尔型的使用	47	5.3.1	任意类型 anyType	81
4.2	自定义简单类型的使用	48	5.3.2	如何定义 complexContent	82
4.2.1	枚举型——enumeration	48	5.3.3	complexContent 定义的缩略模式	83
4.2.2	限制长度——minLength、maxLength	50	5.3.4	定义子元素顺序	84
4.2.3	取值范围——minInclusive、maxInclusive	51	5.3.5	定义子元素出现次数	87
4.2.4	小数位数限定——totalDigits 和 fractionDigits	53	5.3.6	顺序指示符及元素出现次数综合使用实例	89
4.2.5	模式匹配——pattern	54	5.3.7	顺序指示符和元素出现次数小结	89
4.2.6	原子类型——atomic	54	5.3.8	综合使用子元素和属性声明	90
4.2.7	列表类型——list	55	5.3.9	simpleContent 和 complexContent 的几种理解	92
4.2.8	联合类型	56	5.4	特殊的复杂类型	95
4.2.9	分析 type 属性值的特殊性	58	5.4.1	anyType 的约束	95
4.2.10	type 属性与名称空间	58	5.4.2	anyType 的约束派生	99
4.2.11	使用比喻来解释 type 属性和名称空间关系	62	5.4.3	空元素的定义	99
4.2.12	简单数据类型小结	64	5.4.4	混合内容类型的定义	101
4.3	元素的默认值和固定值	65	5.5	ref 引用	102
4.3.1	元素默认值	65	5.5.1	ref 引用元素声明	103
4.3.2	元素默认值的用途	66	5.5.2	元素组的使用	105
4.3.3	元素固定值	67	5.5.3	属性组的使用	106
4.4	本章小结	68	5.6	全局声明与局部声明	109
第 5 章	Schema 声明复杂元素	69	5.6.1	什么是全局声明和局部声明	109
5.1	复杂元素的分类: simpleContent 和 complexContent	70	5.6.2	目标名称空间 (targetNamespace)	111
5.1.1	理解内容 (content)	70	5.6.3	限定与非限定	112
			5.6.4	在 Schema 中声明名称空间	115
			5.6.5	全局声明和局部声明的比较	118
			5.6.6	无名称空间元素 (属性) 如何验证	119
			5.7	本章小结	121

第6章 Schema 使用技巧 122

6.1 指定唯一性——使用 unique	123
6.1.1 定义元素唯一性	123
6.1.2 定义属性唯一性	124
6.1.3 慎用默认名称空间	126
6.2 key 和 keyref 的使用	127
6.3 include 的使用	129
6.4 import 的使用	132
6.5 使用 xml:lang	135
6.6 本章小结	136

第三篇 XML 转换——XSLT

第7章 浅析 XSLT 开发 138

7.1 XSLT 简介	139
7.1.1 XSLT 的本质	139
7.1.2 XSL 和 XSLT 的关系	139
7.1.3 XSLT 的专用名称空间	140
7.1.4 XSLT 处理器	140
7.2 第一个 XSLT 文档	140
7.2.1 编写第一个 XSLT 文档	140
7.2.2 查看第一个 XSLT 的输出结果	142
7.2.3 为 XSLT 转换选择处理器	142
7.2.4 修改 XSLT 文档的参数	143
7.2.5 分析第一个 XSLT 文档	144
7.2.6 在浏览器中使用 XSLT	145
7.2.7 编写 XSLT 文档的注意事项	146
7.3 本章小结	146

第8章 XSLT 基础 147

8.1 进一步思考 XSLT 的本质	148
8.1.1 XSLT 是一门编程语言	148
8.1.2 XSLT 相对其他语言的特点	148
8.2 XML 文档结构描述——XPath	148
8.2.1 什么是 XPath	149
8.2.2 XPath 表达式简介	149
8.2.3 XPath 和 XSLT 的关系	150

8.2.4 典型实例	150
8.3 常用 XSLT 函数概述	152
8.3.1 current()函数——返回当前节点	152
8.3.2 document()函数——使用外部 XML 文档	153
8.3.3 element-available()函数——判断元素可用性	154
8.3.4 format-number()函数——格式化数字	156
8.3.5 function-available()函数——判断函数可用性	157
8.3.6 generate-id()函数——生成节点 id	158
8.3.7 key()函数——返回索引中的节点	158
8.3.8 system-property()函数——返回系统属性	158
8.4 XPath 节点集函数	159
8.4.1 count()函数——统计节点个数	159
8.4.2 position()函数——返回节点位置	160
8.4.3 last()函数——返回节点总数	161
8.4.4 name()函数——返回节点名称	162
8.5 XPath 字符串函数	162
8.5.1 concat()函数——连接字符串	163
8.5.2 contains()函数——判断包含关系	164
8.5.3 normalize-space()函数——格式化空白符	165
8.5.4 starts-with()函数——判断开头字符串	166
8.5.5 string-length()函数——返回字符串长度	167
8.5.6 substring()函数——截取字符串	167
8.5.7 substring-before()函数——向前截取字符串	168
8.5.8 substring-after()函数——向后截取字符串	169
8.5.9 translate()函数——替换字符	170
8.6 XPath 数值型函数	171
8.6.1 ceiling()函数——上取整	171
8.6.2 floor()函数——下取整	172

8.6.3	number()函数——数值转换	173
8.6.4	round()函数——四舍五入取整	174
8.6.5	sum()函数——求和	175
8.7	XPath 布尔型函数	176
8.7.1	boolean()函数——计算布尔值	176
8.7.2	false()函数——返回 false	177
8.7.3	lang()函数——判断 xml:lang 属性	177
8.7.4	not()函数——取反操作	178
8.7.5	true()函数——返回 true	180
8.8	XPath 函数小结	180
8.9	XSLT 运算符	180
8.9.1	算术运算符	180
8.9.2	逻辑运算符	180
8.9.3	比较运算符	181
8.10	概念分析	181
8.10.1	当前节点和上下文节点	181
8.10.2	XPath 表达式与字符串	183
8.11	本章小结	184

第 9 章 理解模板 185

9.1	定义和调用模板	186
9.1.1	模板定义语法	186
9.1.2	调用节点模板 (xsl:apply-templates)	186
9.1.3	调用命名模板	190
9.2	模板冲突	193
9.2.1	最基本的模板冲突	193
9.2.2	mode 属性的使用	195
9.2.3	设置模板优先级——priority 属性的使用	197
9.2.4	包含模板	201
9.2.5	导入模板	205
9.3	XSLT 执行过程	206
9.3.1	XSLT 程序的入口点	206
9.3.2	XSLT 程序哪些模板将被执行	207
9.3.3	模板调用顺序跟 XML 文档没有任何关系	209
9.4	内嵌模板	210
9.5	模板递归	211

9.5.1	节点模板递归	211
9.5.2	命名模板递归	214
9.6	本章小结	218

第 10 章 XSLT 应用——输出 219

10.1	输出文档控制——<xsl:output>	220
10.1.1	<xsl:output>元素的使用语法	220
10.1.2	定义输出格式——method 属性	221
10.1.3	设置输出编码格式——encoding 属性	222
10.1.4	设置自动换行和缩进——indent 属性	224
10.1.5	输出文档控制小结	225
10.2	输出文本——<xsl:value-of>	226
10.2.1	<xsl:value-of>语法	226
10.2.2	典型实例 1——输出复杂元素	227
10.2.3	典型实例 2——输出节点集合	228
10.3	节点复制——<xsl:copy>、<xsl:copy-of>	229
10.3.1	<xsl:copy>元素的使用	229
10.3.2	<xsl:copy-of>元素	230
10.4	生成元素——<xsl:element>	231
10.4.1	<xsl:element>语法	231
10.4.2	典型实例 1——动态生成元素名	232
10.4.3	典型实例 2——特殊名称空间元素	234
10.4.4	生成元素小结	235
10.5	生成属性——<xsl:attribute>	235
10.5.1	<xsl:attribute>语法	235
10.5.2	典型实例 1——动态生成属性	235
10.5.3	典型实例 2——属性值难以获得	237
10.5.4	典型实例 3——所依附的元素为 <xsl:element>	239
10.5.5	典型实例 4——生成 XSLT 文档	241
10.5.6	生成属性小结	243
10.6	输出指令——<processing-instruction>	243
10.6.1	<processing-instruction>语法	243
10.6.2	典型实例——创建处理指令	243
10.7	输出注释——<xsl:comment>	245

10.7.1	<xsl:comment>使用语法	246
10.7.2	典型实例——存储报价	246
10.8	输出消息——<xsl:message>	248
10.8.1	<xsl:message>语法	248
10.8.2	典型实例 1——文本提示	248
10.8.3	典型实例 2——指令模板	250
10.9	替换名称空间——<xsl:namespace-alias> 元素	251
10.9.1	<xsl:namespace-alias>语法	251
10.9.2	典型实例——生成 XSLT 文档	252
10.10	<xsl:decimal-format>元素	254
10.10.1	<xsl:decimal-format>语法	254
10.10.2	典型实例——常用属性使用	255
10.10.3	使用模式分隔符——pattern-separator 属性	257
10.11	阻止名称空间的输出	258
10.12	空白符对输出的影响	260
10.12.1	无意义空白符	261
10.12.2	有意义空白符	264
10.12.3	XML Schema 与空白符	265
10.13	本章小结	265

第 11 章 XSLT 应用——变量定义和 流程控制 266

11.1	定义变量——<xsl:variable>	267
11.1.1	使用语法	267
11.1.2	典型实例——select 属性赋值	267
11.1.3	使用 content 赋值	270
11.1.4	select 属性和 content 赋值的区别	271
11.1.5	临时结果树片段	273
11.1.6	变量的有效范围	275
11.1.7	定义变量小结	277
11.2	条件控制——<xsl:if>	277
11.2.1	<xsl:if>语法	277
11.2.2	分析 test 属性	277
11.2.3	字符串类型	278
11.2.4	数值类型	279
11.2.5	布尔类型	280

11.2.6	节点集合类型	281
11.2.7	典型实例	282
11.2.8	条件控制小结	283
11.3	多条件选择——<xsl:choose>	283
11.3.1	使用语法	284
11.3.2	典型实例	284
11.4	循环处理——<xsl:for-each>	287
11.4.1	<xsl:for-each>语法	287
11.4.2	典型实例	287
11.4.3	循环嵌套	288
11.4.4	分析<xsl:for-each>与 <xsl:apply-templates>的关系	290
11.4.5	条件约束	292
11.4.6	多条件约束	293
11.4.7	循环处理小结	296
11.5	异常处理——<xsl:fallback>	296
11.5.1	<xsl:fallback>语法	297
11.5.2	典型实例	297
11.5.3	关于用 element-available ()函数代替 <xsl:fallback>	299
11.6	本章小结	300

第 12 章 XSLT 应用——排序、索引 与编号 301

12.1	排序处理——<xsl:sort>	302
12.1.1	<xsl:sort>语法	302
12.1.2	典型实例——单关键字排序	302
12.1.3	典型实例——多关键字排序	304
12.1.4	典型实例——求极值	306
12.2	建立索引——<xsl:key>	308
12.2.1	<xsl:key>语法	308
12.2.2	典型实例 1——一对一映射	308
12.2.3	典型实例 2——一对多映射	310
12.2.4	典型实例 3——多对多映射	312
12.3	生成编号——<xsl:number>	315
12.3.1	使用语法	315
12.3.2	典型实例 1——默认输出	316

12.3.3 典型实例 2——指定编号	317	14.7 父轴——parent	377
12.3.4 典型实例 3——指定计数层次和 节点	318	14.7.1 属性节点的父轴	378
12.3.5 典型实例 4——指定计数开始位置	324	14.7.2 上下文节点的父轴	380
12.3.6 典型实例 5——指定编号格式	326	14.8 属性轴——attribute	382
12.4 本章小结	329	14.8.1 属性轴的方向	383
第 13 章 XSLT 应用——分组	330	14.8.2 属性轴的缩略模式	385
13.1 什么是分组	331	14.9 本身轴——self	386
13.2 解决分组问题的思路	332	14.10 后代轴——descendant	390
13.3 解决分组问题的办法	332	14.10.1 后代轴的方向	391
13.4 MuenChian 方法分组	335	14.10.2 后代轴的缩略模式	392
13.5 较为复杂的分组	338	14.11 后代或本身轴——descendant-or-self	394
13.6 本章小结	341	14.11.1 后代或本身轴的方向	395
第 14 章 XPath 的灵魂——轴	343	14.11.2 后代或本身轴的应用	396
14.1 什么是轴	344	14.12 祖先轴——ancestor	397
14.1.1 轴的本质	344	14.12.1 祖先轴的方向	398
14.1.2 轴的分类	344	14.12.2 祖先轴的应用——输出 XPath	400
14.2 前驱轴——preceding	345	14.13 祖先或本身轴——ancestor-or-self	401
14.2.1 前驱轴应用 1——分组	346	14.13.1 祖先或本身轴的方向	402
14.2.2 前驱轴应用 2——统计节点	348	14.13.2 祖先或本身轴应用——输出 XPath	402
14.2.3 preceding 轴只与源文档有关	351	14.14 名称空间轴——namespace	403
14.2.4 preceding 轴的方向	352	14.14.1 名称空间轴的默认节点	404
14.3 前驱兄弟轴——preceding-sibling	354	14.14.2 名称空间的继承性	404
14.3.1 前驱兄弟轴的应用——分组	355	14.14.3 名称空间轴上节点的名称	405
14.3.2 前驱兄弟轴的应用——统计节点	357	14.15 常见问题	405
14.3.3 综合应用实例	357	14.15.1 轴的分类问题	405
14.4 后继轴——following	361	14.15.2 节点测试	407
14.4.1 典型实例	363	14.15.3 重新理解 XPath 表达式	408
14.4.2 following 轴的方向	365	14.15.4 重新理解 “/” 和 “//”	408
14.5 后继兄弟轴——following-sibling	366	14.16 本章小结	410
14.5.1 典型示例	368	第四篇 EXSLT (扩展 XSLT) 和 XSLT 2.0	
14.5.2 综合应用实例	370	第 15 章 EXSLT——扩展 XSLT	412
14.6 子轴——child	372	15.1 EXSLT 简介	413
14.6.1 子轴的方向	373	15.1.1 EXSLT 介绍	413
14.6.2 子轴和前驱、后继轴	375	15.1.2 EXSLT 的前景	413

15.2 应用方向一——获得日期和时间	414
15.3 应用方向二——将字符串当做 XPath 表达式进行计算	417
15.3.1 典型实例 1——计算节点	417
15.3.2 典型实例 2——变量传递	418
15.4 本章小结	422

第 16 章 XPath 2.0 423

16.1 XPath 2.0 简介	424
16.1.1 XPath 2.0 的使用环境	424
16.1.2 对 XML Schema 内建数据类型 的支持	424
16.1.3 增加了 XPath 数据类型名称空间	426
16.1.4 注意非整数天数的处理	429
16.2 序列概念的引入	430
16.2.1 什么是序列	430
16.2.2 XPath 1.0 表达式如何与 XPath 2.0 兼容起来	431
16.2.3 典型实例——实现循环次数	431
16.3 序列运算符	432
16.3.1 循环操作	432
16.3.2 条件判断	435
16.4 字符串函数	437
16.4.1 compare() 函数——比较字符串	437
16.4.2 ends-with() 函数——判断结尾 字符串	438
16.4.3 starts-with() 函数——判断开头 字符串	439
16.4.4 string-join() 函数——串联序列对象	439
16.4.5 subsequence() 函数——获取子序列	441
16.4.6 matches() 函数——匹配正则表达式	441
16.4.7 tokenize() 函数——分割字符串为 序列	443
16.4.8 replace() 函数——替换字符串	443
16.4.9 lower-case() 函数——返回小写字符	446
16.4.10 upper-case() 函数——返回大写 字符	447
16.5 数值型函数	448

16.5.1 avg() 函数——获取平均值	448
16.5.2 max() 函数——获取最大值	449
16.5.3 min() 函数——获取最小值	449
16.6 日期型函数	451
16.7 本章小结	454

第 17 章 XSLT 2.0 455

17.1 XSLT 2.0 简介	456
17.1.1 XSLT 2.0 的地位	456
17.1.2 如何使用 XSLT 2.0	456
17.1.3 XSLT 2.0 增加的新内容	456
17.2 XSLT 2.0——分组	457
17.2.1 分组的使用	457
17.2.2 分组排序	459
17.2.3 分组中的上下文节点	461
17.2.4 分组嵌套	462
17.2.5 动态分组	465
17.2.6 利用 group-adjacent 属性进行 相邻分组	468
17.2.7 利用 group-starting-with 和 group- ending-with 属性指定分组条件	470
17.3 多文件输出——<xsl:result-document>	472
17.3.1 <xsl:result-document> 语法	472
17.3.2 典型实例——分割 XML 文档	473
17.4 字符映射——<xsl:character-map> 元素	475
17.4.1 <xsl:character-map> 语法	475
17.4.2 典型实例——生成 JSP 文档	475
17.5 分析字符串——<xsl:analyze-string> 元素	477
17.5.1 <xsl:analyze-string> 元素简介	477
17.5.2 典型实例——获得文件名	478
17.5.3 关于 regex 属性中的 “{}”	479
17.5.4 正则表达式分组——regex-group() 函数的使用	480
17.6 定义变量 <xsl:variable> 和创建序列 <xsl:sequence>	482
17.6.1 <xsl:variable> 元素的变化	482
17.6.2 <xsl:sequence> 元素	483

17.6.3 序列和临时结果树的优劣比较	485
17.7 本章小结	485

第 18 章 正则表达式 486

18.1 正则表达式在 XML 中的应用	487
18.1.1 在 XML Schema 中的应用	487
18.1.2 在 XPath 2.0 中的应用	487
18.1.3 在 XSLT 2.0 中的应用	487
18.2 XML 正则表达式简介	488
18.2.1 元字符和普通字符	488
18.2.2 量词	488
18.2.3 字符转义与字符类	489
18.2.4 字符组的使用	490
18.2.5 正则表达式分支	490
18.3 本章小结	490

第五篇 XML 在 Java 中的典型应用

第 19 章 用 JDOM 解析、处理 XML 文档 492

19.1 用 JDOM 解析 XML 文档	493
19.1.1 准备工作	493
19.1.2 创建 Java 类	494
19.2 用 JDOM 处理 XML 文档	496
19.2.1 创建 XML 文档	496
19.2.2 删除和修改节点	498
19.3 本章小结	500

第 20 章 用 JAXB 处理 XML 文档 501

20.1 准备工作	502
20.1.1 下载与安装 JAXB	502
20.1.2 XJC 简介	503
20.1.3 JXL 简介	504
20.1.4 查看用来映射的 XML Schema 文档	504
20.2 项目开发	505
20.2.1 创建项目	505
20.2.2 利用 XJC 生成 Java 类	505

20.2.3 存储了字典表的 Excel 文档	509
20.3 本章小结	512

第六篇 XML 在 C# 中的典型应用

第 21 章 使用 XML DOM 操作节点 514

21.1 C# 中的 XML DOM	515
21.1.1 XML DOM 的操作对象 XmlDocument	515
21.1.2 使用 XML 文件分析 XmlDocument 中的对象	515
21.1.3 使用 DOM 对象获取 XML 文件	516
21.1.4 使用 DOM 对象获取 XML 文件中的指定节点	517
21.1.5 使用 DOM 对象改变 XML 文件的数据顺序	519
21.2 XML 文件读取器——XmlReader	520
21.2.1 XmlReader 的作用	520
21.2.2 对 XML 的验证	521
21.2.3 使用 XmlReader 读取 XML 文件的一部分	521
21.2.4 使用 XmlTextReader 读取整个 XML 文件	523
21.3 XML 文件编写器——XmlWriter	524
21.3.1 XmlWriter 的作用	524
21.3.2 XmlWriter 对 XML 文件的验证	525
21.3.3 用 XmlWriter 创建并编辑 XML 文件	525
21.4 XML 与 DataSet 的交互	527
21.4.1 将 XML 文件转化为 DataSet 数据集	527
21.4.2 将 DataSet 数据集转换为 XML 文件	528
21.5 本章小结	531

附录 A 本书光盘应用注意事项 532

附录 B 光盘代码说明 533



程序天下

第一篇

XML 简介

第 1 章

XML 概述

XML 技术现在已广泛应用于数据转换、科学计算、图形图像、无线通信、Web 开发等技术领域。尤其是 Internet 的迅猛发展，不仅成就了 Java，同时也成就了 XML。XML 已经成为 Web 开发中不可或缺的技术之一。本章将介绍这一流行技术的概貌。