

XSLT

技术内幕

Inside XSLT

(美) Steven Holzner 著

闻道工作室 译



机械工业出版社
China Machine Press

New
Riders

Internet新技术丛书

XSLT技术内幕

(美) Steven Holzner 著

闻道工作室 译



机械工业出版社
China Machine Press

本书全面地介绍了XSLT的各方面知识,读者不仅可以了解完整的XSLT语法,还可以深入学习XSLT的应用方法。主要内容包括:XSLT的基础知识,样式单的创建与使用,模板的创建与使用,匹配形式的创建,数据排序,XPath的应用与理解,XSLT和XPath函数的使用,模板、参数和变量的命名,XSLT处理程序API的使用等等。本书还提供了数以百计的工作实例,以便使读者更易掌握XSLT。

本书内容全面、组织合理、实例丰富,并包含有其他书籍中不多见的上百个在线资源、技巧和分析参考。本书不仅可作为初学者入门指导,还可以作为程序人员的完整参考手册。

Steven Holzner: Inside XSLT.

Authorized translation from the English language edition published by New Riders, an imprint of Macmillan Computer Publishing U.S.A.

Copyright © 2001 by New Riders Publishing. All rights reserved.

Chinese simplified language edition published by China Machine Press.

Copyright © 2002 by China Machine Press.

本书中文简体字版由美国麦克米兰公司授权机械工业出版社独家出版。未经出版者书面许可,不得以任何方式复制或抄袭本书内容。

版权所有,侵权必究。

本书版权登记号: 图字: 01-2001-4283

图书在版编目(CIP)数据

XSLT技术内幕/(美)霍兹纳(Holzner, S.)著;闻道工作室译.-北京:机械工业出版社, 2002.1

(Internet新技术丛书)

书名原文: Inside XSLT

ISBN 7-111-09633-9

I. X… II. ①霍… ②闻… III. 可扩展语言, XSLT-程序设计 IV. TP312

中国版本图书馆CIP数据核字(2001)第091260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:白红莉 张鸿斌

北京第二外国语学院印刷厂印刷·新华书店北京发行所发行

2002年1月第1版第1次印刷

787mm×1092mm 1/16·31.75印张

印数:0 001-4 000册

定价:49.00元

凡购本书,如有倒页、脱页、缺页,由本社发行部调换

译者序

XSLT（可扩展样式单语言变换）是与XML相关联的、且同样由W3C推出的一项标准。

在当今软件界，能像XML这样同时得到以微软和Sun为代表的两大阵营共同支持和成功应用，实属罕见，也许只有作为通信协议的TCP/IP标准可以与之比肩。这一切皆源于XML在描述（标记、置标）信息时的可扩展性、跨平台性和自描述性，和当今网络时代对跨平台信息交换的迫切需求，其中XSLT技术可谓功不可没。

关于XML的任何著作中几乎都不可避免地要讨论XSLT，这难免不让人产生XSLT为XML标准之一部分的误解。目前在市面上关于XML的书籍多如牛毛，但像本书这样专门介绍XSLT的专著则不多见。

首先，本书力求最全面地介绍XSLT。包括从XSLT 1.0规范、XSLT 1.1工作草案到XSLT 2.0需求的所有版本；与现有介绍XSLT的书籍不同，本书不仅讨论XML至HTML的变换，也探讨XML至XML、XHTML、SQL、JavaScript、普通文本乃至PDF文档的变换。

其次，本书力求清晰易懂地介绍XSLT。作为鼎鼎大名的畅销技术书籍作家，本书作者极尽其组织素材、阐释疑难问题之长，捐一己之艰辛以释读者求索之苦。实践出真知，几乎对每一技术要点，作者都提供了实例。此外，还提供了大量的相关网上资源的URL。

本书还超越了入门教程和完备参考手册之间的界限，不仅初学者可以用作学习XSLT的教材，中高级用户也将从中受益匪浅。

闻道工作室的多位同仁参与了本书的翻译、核对和录入，为力求准确，大家付出了巨大的努力，但由于我们学知有限，错误之处仍难避免，欢迎读者批评指正。

2001年11月

前言

这是一本关于可扩展样式单语言变换的著作。本书试图在一本书的篇幅之内尽可能全面且清晰易懂地介绍XSLT。XSLT完全是关于转换XML文档为其他的XML文档，或者整个地转换为其他类型的文档，并且正变为极端热门的话题。本书是关于XSLT的、最完备的著作，而且我确信与其他有关书籍相比，本书提供了对XSLT的更深入的探讨。这是一个快速扩张的领域，也是一个复杂的领域——我已经在此投入了艰辛的劳动，为的是使读者避免这样做。

许多XSLT书籍只集中探讨XML至HTML的变换，但是，相比之下，本书则深入得多。XML至XML的变换正变得非常重要，而且随着具备不同特色的XML应用的增加，这种变换不久将比XML至HTML的变换更重要。

实际上，在本书中，我们将推出完整的解决方案，考察从XML至XML、HTML、XHTML、RTF（多信息文本格式文档）、普通文本、JavaScript、基于SQL的数据库和XSL-FO（可扩展样式单语言格式化对象文档）的XSLT变换。

为了完整地介绍XSLT，本书逐个地描述所有可以得到的XSLT元素，以及它们的所有属性。本书还介绍了每种可用于XSLT样式单的XSLT和XPath函数。

读者还将看到本书中有数以百计的工作实例，毫无疑问，它是理解XSLT如何工作的最佳方式。如果我没有通过文字解释清楚某些内容，读者还可以参考这些实例，本书试图对几乎所有内容，都介绍了一种实用的、完整的实例。

所有这些实例都可以运行。本书使用可以得到的最流行的XSLT处理程序，它们都可以从因特网上免费下载：Xalan、XT、Saxon、Oracle的XSLT处理程序，甚至包括内置于微软的Internet Explorer的处理程序。本书通篇都将介绍从因特网上的什么位置得到这些软件，以及如何使用它。

如同编写XML一样，编写XSLT同样不是一件普通和单调的任务：它会启发艺术修养、爱心、热情、快乐和执着——恼怒和挫折自不待言。笔者试图在本书中尽可能地真实体现这种精神并最大限度地捕捉XSLT的激动人心之处和威力。

本书包含哪些内容

本书提供了完整的XSLT知识。读者不仅可以了解完整的XSLT语法——从最基础的到最高级的，而且还将深入了解许多当今XSLT的应用方法。

本书介绍了数以百计的实际应用论题，诸如在服务器端执行XSLT变换，连接至数据库，以及用浏览实时地实施变换。

下面是本书所介绍的一些论题。注意，这些论题都包含许多内容（太多而无法逐一列举）：

- XSLT 1.0规范
- XSLT 1.1工作草案

- XSLT 2.0需求
- XPATH 1.0规范
- XPath 2.0需求
- XSL 1.0候选规范
- 使用Xalan、Saxon XT和Oracle的XSLT处理程序
- 创建XML、HTML、RTF和文本输出
- 重复的模板处理
- 缺省模板规则
- 处理空白
- 禁止输出转义
- 选择适用的模板
- 创建匹配形式
- 在匹配形式中使用断言
- 匹配元素、子元素、元素后代、属性等
- 通过ID匹配
- XPath缩写语法
- If条件
- For-each结构
- 作出条件性“选择”决策
- 简单元素排序
- 处理多重选择
- 嵌套模板
- 样式单参数
- 简化的样式单
- 编写用于Internet Explorer的样式单
- 创建样式单规则
- 使用扩展元素
- 使用扩展函数
- 创建文字结果元素
- 使用属性值模板
- 删除空白
- 创建新元素、属性和注释等
- XSLT型态（上下文指定的格式化）
- 结果树片断
- XPath数据格式
- XPath表达式
- 所有的XSLT和XPath函数

- 创建属性集
- 调用已命名模板
- 关键字、单一和多值
- 创建多输出文档
- 在Java代码中，通过API调用来使用Oracle、Saxon、XML4Java和XT
- 为Internet Explorer中的XSLT使用JavaScript
- XSL-FO格式化对象
- XSL-FO格式化属性
- 用Java服务器页面（JSP）、活动服务器页面（ASP）和Java Servlet实现服务器端XSLT变换。

本书完整地介绍由World Wide Web联盟（W3C：XSLT 1.0规范、XSLT 1.1工作草案和XSLT 2.0需求——关于这些用语的具体含义，请参见第1章）创建的正式XSLT规范。正如第1章所介绍的，W3C已声称XSLT 1.1工作草案将不会成为正式规范，因为他们决心推出XSLT 2.0（XSLT 2.0目前正处于需求阶段，这是一种出现在工作草案阶段之前的阶段）。但是，由于XSLT 1.1工作草案的内容将会被吸收进XSLT 2.0——而且由于许多XSLT处理程序将实现XSLT 1.1的已知部分——因此，本书也将完整地介绍XSLT 1.1工作草案，并且在处理XSLT 1.1工作草案的专属材料时明确指明。

在编写一个XSLT样式单时，必须指明所使用的XSLT版本。因为版本1.1决不会超越工作草案阶段或被正式发布，本书的实例的版本设定为1.0，在讨论XSLT 1.1工作草案的专属材料期间除外，在此期间，本书显式地设置为版本1.1。

当今在因特网上存在非常丰富的XSLT资源，因此本书包含大量的这些资源的URL，包括本书所使用的所有软件，它们是可以免费下载的。（不过，存在一种必须说明的风险——因特网上的URL变动频繁，因此，如果在浏览它们时，遇到其中一些URL改变了的情况不要大惊小怪。）

本书的完整代码清单是可以在线得到的资源之一。读者可以在New Riders网站www.newriders.com查找到它们。

适于阅读本书的读者

本书是为任何想要学习XSLT以及想了解如何在当今的现实生活中使用XSLT的人们而编写的。本书所作的惟一假设是，读者具备某些用XML和HTML编写文档的知识。本书的姐妹篇是New Riders出版的《Inside XML》（也由本人编写），如果读者需要很好地理解XML，可以翻阅那本书。这些主题方面的知识终究是有帮助的，它是读者真正需要的全部预备知识。

如果读者想要使用从网上免费下载的XSLT处理程序运行本书的实例，那么还应该在自己的系统中安装Java（它也是免费的），而且第1章还将完整地讨论这方面的内容。

本书是在什么层次上编写的

本书是按照从初级到高级的几种不同的层次进行编写的，因为XSLT领域太广泛。笔者试图尽可能多地介绍XSLT，因此包括从入门到高级层次的所有内容。如上所述，假设读者具备某些XML和HTML的知识。

本书不要求读者具备任何程序设计的知识。但第10章除外，该章专门讨论如何借助Java来利用XSLT的功能，如果读者想要学习这一章，则应该具备某些Java程序设计的经验。

由于当今存在众多的XSLT用法，因此本书使用了几种不同的软件包。笔者所使用的所有软件包都是从因特网上免费下载的，而且本书将告诉读者从哪里找到它们。

至此，万事俱备。如果读者有任何建议，可以给我写信，这是关心New Riders之举。本书计划成为介绍XSLT的新标准，确实比此前任何有关书籍更全面更易懂。如果读者认为本书欠缺什么内容，请告诉我，我将予以补充，因为我希望本书永保第一。

本书英文原书书名：Inside XSLT

英文原书书号：ISBN 0-7357-1136-4

英文原书出版社网站：www.newriders.com

目 录

译者序

前言

第1章 XSLT基础1

1.1 XSL = XSLT + XSL-FO1

1.2 简单背景1

1.2.1 XSLT—XSL变换2

1.2.2 XSL-FO: XSL格式化对象2

1.2.3 W3C规范3

1.2.4 XSLT版本3

1.3 XML文档4

1.3.1 合式XML文档6

1.3.2 有效的XML文档6

1.4 XML在浏览器中看起来如何9

1.5 XSLT变换9

1.6 执行XSLT变换12

1.7 使用独立的XSLT处理程序12

1.7.1 使用一种Java XSLT处理程序14

1.7.2 James Clark的XT15

1.7.3 Saxon16

1.7.4 Oracle XSLT16

1.7.5 Xalan16

1.8 用浏览器变换XML文档18

1.9 在Internet Explorer中使用XSLT和

JavaScript21

1.10 位于Web服务器的XSLT变换22

1.11 XML至XML的变换23

1.12 XML至XHTML的变换26

1.13 XSLT资源29

1.13.1 XSLT规范、教程和实例29

1.13.2 XSLT编辑器30

1.13.3 XSLT实用工具32

1.14 XSL格式化对象: XSL-FO33

1.15 XSL-FO资源33

1.16 格式化XML文档34

1.17 XSLT样式单35

1.18 变换文档为格式化对象形式37

1.19 创建格式化文档39

第2章 创建与使用样式单41

2.1 树与节点41

2.2 XSLT树模型与信息集模型45

2.3 XSLT元素46

2.4 <?xsl:stylesheet?>处理指令46

2.5 <xsl:stylesheet>元素48

2.6 顶层样式单的元素50

2.7 <xsl:template>元素51

2.8 模板体51

2.8.1 XSLT指令52

2.8.2 扩展元素52

2.8.3 文字结果元素52

2.8.4 模板中的匹配元素53

2.9 <xsl:apply-templates>元素54

2.10 访问节点值57

2.11 支持XML基础规范62

2.12 选择输出方法63

2.12.1 输出方法: HTML64

2.12.2 输出方法: XML66

2.12.3 输出方法: 文本67

2.13 简易的样式单68

2.14 内嵌样式单69

2.15 <xsl:include>元素72

2.16 <xsl:import>元素73

2.17 <xsl:apply-imports>元素74

2.18 利用Internet Explorer变换XML文档76

第3章 创建与使用模板	80	4.10.2 断言: 布尔型	135
3.1 创建一个模板	80	4.10.3 断言: 数字	138
3.2 处理子节点	81	4.10.4 字符串	139
3.3 访问节点值	83	4.10.5 断言: 结果树片断	142
3.4 创建匹配形式	84	4.11 通过ID匹配	143
3.5 选择应用模板	85	4.12 通过关键字匹配	144
3.6 读属性值	89	4.13 使用Or操作符	146
3.7 <xsl:text>元素	90	4.14 形式实例	148
3.8 禁止输出转义	92	第5章 数据选择与排序	157
3.9 写属性值	95	5.1 <xsl:if>元素	157
3.10 属性值模板	96	5.2 <xsl:choose>、<xsl:when>和 <xsl:otherwise>元素	161
3.12 <xsl:strip-space>和<xsl:preserve-space> 元素	99	5.3 <xsl:for-each>元素	168
3.13 自动缩进	101	5.4 排序元素	172
3.14 缺省模板规则	104	5.5 <xsl:number>元素	177
3.15 删除内容	106	5.5.1 单层编号方式	178
3.16 模板冲突的解决方案	108	5.5.2 任意层面编号方式	181
3.17 <xsl:copy>元素	111	5.5.3 多层面编号方式	184
3.18 <xsl:copy-of>元素	113	5.6 XSLT可扩展性	186
3.19 <xsl:message>元素	114	5.7 扩展函数	187
第4章 创建匹配形式	117	5.7.1 <xsl:script>元素	189
4.1 匹配根节点	118	5.7.2 使用function-available函数	194
4.2 匹配元素	118	5.7.3 外部对象	194
4.3 匹配子节点	118	5.8 扩展元素	194
4.4 匹配元素后代	119	5.9 <xsl:fallback>元素	197
4.5 匹配属性	119	第6章 XML变换为XML、HTML、XHTML、 RTF和修改文档内容	199
4.6 正式定义的匹配形式	121	6.1 <xsl:output>元素	200
4.7 步骤形式的第1部分: 形式轴	123	6.1.1 输出方法: HTML	201
4.8 步骤形式的第2部分: 节点测试	126	6.1.2 输出方法: XML	205
4.8.1 匹配注释	127	6.1.3 输出方法: Text	209
4.8.2 用node()匹配节点	128	6.1.4 输出XHTML	211
4.8.3 用text()匹配文本节点	129	6.2 基于输入的改变文档结构	215
4.8.4 匹配处理指令	130	6.3 <xsl:element>元素: 运行时创建新元素	215
4.9 步骤形式的第3部分: 断言	130	6.4 <xsl:attribute>元素: 创建新属性	218
4.10 创建断言	133	6.5 <xsl:comment>元素: 生成注释	219
4.10.1 断言: 节点集	133		

6.6 <xsl:processing-instruction>元素: 生成处理指令	220	7.5 XPath缩写语法	271
6.7 <xsl:document>元素: 生成多个输出文档	222	7.6 检查XPath表达式	272
6.8 <xsl:namespace>元素: 生成名字空间声明	224	7.7 XPath 2.0	272
6.9 <xsl:attribute-set>元素: 生成属性组	224	第8章 使用XSLT和XPath函数	274
6.10 省略XML声明和生成XML片段	226	8.1 XSLT函数	275
6.11 使用generate-id创建唯一的标识符	228	8.1.1 current()	275
6.12 创建CDATA节	232	8.1.2 document()	277
6.13 设置字符编码	233	8.1.3 element-available()	279
6.14 模式: 上下文指定格式	234	8.1.4 format-number()	280
第7章 应用和理解XPath	242	8.1.5 function-available()	283
7.1 理解XPath	243	8.1.6 generate-id()	284
7.2 XPath数据类型	246	8.1.7 key()	286
7.2.1 XPath节点组	246	8.1.8 system-property()	288
7.2.2 XPath数字	249	8.1.9 unparsed-entity-uri()	289
7.2.3 XPath字符串	250	8.2 XPath 节点组函数	292
7.2.4 XPath布尔值	254	8.2.1 count()	292
7.3 创建XPath位置路径	255	8.2.2 id()	294
7.3.1 XPath定位步骤第1部分: 轴	255	8.2.3 last()	295
7.3.2 XPath定位步骤第2部分: 节点测试	256	8.2.4 local-name()	296
7.3.3 XPath定位步骤第3部分: 判定	257	8.2.5 name()	296
7.3.4 使用XPath轴	257	8.2.6 namespace-uri()	296
7.3.5 使用ancestor轴	257	8.2.7 position()	298
7.3.6 使用ancestor-or-self轴	258	8.3 XPath字符串函数	298
7.3.7 使用descendant轴	259	8.3.1 concat()	299
7.3.8 使用descendant-or-self轴	261	8.3.2 contains()	300
7.3.9 使用following轴	261	8.3.3 normalize-space()	302
7.3.10 使用following-sibling轴	263	8.3.4 starts-with()	304
7.3.11 使用namespace轴	264	8.3.5 string()	306
7.3.12 使用parent轴	265	8.3.6 string-length()	308
7.3.13 使用preceding轴	266	8.3.7 substring()	310
7.3.14 使用preceding-sibling轴	268	8.3.8 substring-after()	311
7.3.15 使用self轴	269	8.3.9 substring-before()	311
7.4 位置路径举例	270	8.3.10 translate()	311
		8.4 XPath数字函数	312
		8.4.1 ceiling()	312
		8.4.2 floor()	312
		8.4.3 number()	314

8.4.4 round()	314	10.9 Oracle XSLT处理程序与Java接口	379
8.4.5 sum()	314	10.10 XT与Java接口	382
8.5 XPath布尔函数	315	10.11 变换XML为SQL数据库	385
8.5.1 boolean()	315	10.12 与Active Server Pages 一起使用XSLT	389
8.5.2 false()	315	10.13 与Java Server Pages一起使用XSLT	391
8.5.3 lang()	316	10.14 与Java Servlet一起使用XSLT	393
8.5.4 not()	317	第11章 使用XSLT创建XSL-FO文档： 文本和表格	396
8.5.5 true()	319	11.1 XSL格式化	396
8.6 <xsl:decimal-format>元素： 创建数字格式	319	11.2 XSL-FO格式化对象	397
8.7 XSLT和XPath 2.0中的新函数	322	11.3 XSL-FO格式化属性	400
第9章 命名模板、参数和变量	324	11.4 格式化XML文档	405
9.1 <xsl:variable>元素：创建变量	324	11.5 使用XSLT样式单向XSL-FO变换	406
9.1.1 变量作用域	325	11.6 创建文档根：<fo:root>	413
9.1.2 工作中的变量	328	11.7 创建主文件集布局： <fo:layout-master-set>	414
9.2 <xsl:call-template>元素： 使用已命名模板	332	11.8 创建页面主文件： <fo:simple-page-master>	414
9.3 <xsl:param>和<xsl:with-param>元素： 创建参数	335	11.9 创建区域	415
9.3.1 递归调用模板	342	11.10 创建页面序列： <fo:page-sequence>	417
9.3.2 默认模板值	343	11.11 创建流：<fo:flows>	418
9.3.3 在命令行指定模板值	345	11.12 创建块级内容：<fo:block>	419
9.4 <xsl:key>元素：关键字匹配	345	11.13 创建表格	422
9.5 <xsl:document>元素： 生成多个结果文档	352	11.14 创建表格：<fo:table>	429
9.6 <xsl:namespace-alias>元素： 生成样式单	355	11.15 创建表格列：<fo:table-column>	430
第10章 使用XSLT处理程序API	358	11.16 创建表格主体：<fo:table-body>	431
10.1 Internet Explorer中的XSLT和 JavaScript	358	11.17 创建表格行：<fo:table-row>	432
10.2 处理解析错误	360	11.18 创建表格单元：<fo:table-cell>	433
10.3 Internet Explorer和动态样式	363	第12章 使用XSLT创建XSL-FO文档： 列表、图形、列和定位	436
10.4 Internet Explorer和XML数据岛	368	12.1 创建XLS-FO列表	436
10.5 从XSLT处理程序直接调用Java	371	12.1.1 创建列表：<fo:list-block>	440
10.6 XSLT与Java API接口	372	12.1.2 创建列表项：<fo:list-item>	441
10.7 Xalan与Java接口	373	12.1.3 创建列表项标签： <fo:list-item-label>	442
10.8 Saxon与Java接口	375		

12.1.4 创建列表项体:	
<fo:list-item-body>	443
12.2 使用块容器定位文本:	
<fo:block-container>	444
12.3 内置级格式化对象	449
12.3.1 创建内置区域: <fo:inline>	450
12.3.2 分别处理字符: <fo:character>	451
12.3.3 创建页号: <fo:page:number>	453
12.3.4 插入图形: <fo:external-graphic>	454
12.3.5 格式化首行:	
<fo:initial-property-set>	456
12.4 创建脚注: <fo:footnote>和	
<fo:footnote-body>	457
12.5 创建链接: <fo:basic:link>	460
12.6 创建列	463
12.7 页面次序和页面编号	465
附录A XSLT DTD	474
附录B XSL-FO格式化属性	483

第1章 XSLT 基础

欢迎进入可扩展样式单语言变换 (EXTENSIBLE STYLESHEET Language Transformations XSLT) 的领域。本书是进入该领域的指南, 该领域非常广泛, 并且每时每刻都以不可预知的方式在扩展。本书将让读者熟悉该领域。需要讨论的课题很多, 因为当前XSLT正在以最惊人的方式演变为最引人注目的领域。在本书中, 读者将领略到这一切。

XSLT完全是关于XML文档内容的处理与格式化 (New Riders于2000年出版的《Inside XML》一书是该书的姐妹篇)。XML已成为最热门的话题, 而现在该轮到XSLT了。XML允许在文档中结构化地组织数据, 而XSLT允许对XML文档的内容进行处理——管理此内容并且创建别的文档, 例如, 排序一个XML雇员记录数据库或者将该数据保存于HTML文档, 以及按照明确的方式格式化该数据。

可以通过编写至XML解析器应用程序的接口程序处理XML文档的内容, 但是, 这涉及编写自己的代码。在另一方面, 通过使用XSLT可以完成同样的任务, 却不需要任何编程。只是使用XSLT定义想要完成的任务, 然后余下的一切由一个XSLT处理程序完成, 而不是通过编写Java、Visual Basic或C++代码处理XML文档的内容。这正是XSLT所要实现的一切, 而且这将成为XML领域中第2个重大事件。

1.1 XSL = XSLT + XSL-FO

XSLT本身实际上是一个较大规范——可扩展样式单语言, 或XSL——的组成部分。XSL完全致力于定义文档的精确格式, 具体至毫米级。XSL的格式化部分远大于XSLT规范, 它是基于特定的格式化对象, 而且该部分通常称为XSL-FO (或者XSL: FO, 或XSLFO)。XSL-FO是一个复杂的论题, 因为用格式化对象设计文档样式是一个错综复杂的过程。实际上, XSLT最先是为了便于转化XML文档至基于XSL-FO格式化对象的文档而补充至XSL的。

本书是专门讨论XSLT的, 但也对XSL-FO做了简介, 包括如何使用XSLT变换文档至XSL-FO形式, 毕竟, XSLT首先是为便于处理XSL-FO而引入的。作为开始, 本章概括地考察XSLT和XSL-FO。

1.2 简单背景

XSL本身是World Wide Web联盟 (W3C, www.w3.org) ——最先由Tim Berners-Lee发起的一种联盟——的一项创新。W3C是发布规范的机构, 诸如本书所使用的XSL。XML和XSL即是由它们所提出。

W3C与样式单语言

读者可以在www.w3.org/Style/History处读到W3C处理样式单语言的历史。了解付出了多少的努力, 以及这些年样式单语言经历了哪些变化是有趣的。

W3C最先在20世纪80年代开发了XML的鼻祖：SGML（Standard Generalized Markup Language标准通用标记语言）。但是，它太过复杂而未得以广泛应用。实际上，XML（类似如HTML）是一种简化版本的SGML。为与SGML配合使用，W3C还创建了一种称为DSSSL（即Document Style Semantics and Specification Language）的样式语言。如同XML是派生自SGML，XSL则是基于早先的DSSSL。正如同W3C所述：“XSL用于在屏幕中显示文档的模式是建立在许多年工作的基础——即一种称为文档样式语法和规范语言（DSSSL）的复杂ISO标准样式语言——之上的。”

然而，XSL的原始部分——即XSL-FO——并没有证实是足够易于应用而被广泛采用的，因此，XSLT被引入以便简化由XML文档至XSL-FO形式的转换。随着它的退出，XSLT就真正取而代之，因为它提供一种完整的变换语言，它支持在无需编写程序代码的情况下处理XML文档的内容，转换这些文档至另一种XML文档、HTML或其他的基于文本的格式。这种甚至出乎W3C预料的、了不起的成功故事就是XSLT。

1.2.1 XSLT—XSL变换

XSLT允许人们直接处理XML文档的内容。例如，读者可能拥有一个巨大的XML文档，它保存最新棒球赛季的全部统计资料，但是，你可能只对投手的统计资料感兴趣。为了抽取投手的数据，可以用支持XML解析器的Java、Visual Basic或者C++语言编写自己程序。解析器是一种特殊的软件包，它阅读XML文档并且将文档中的全部数据逐渐地传递至你自己的代码。然后读者可以编写一个新的XML文档pitchers.xml，它只包含关于投手的数据。

这种处理方式可行，但是，它涉及相当大量的编程任务，以及大量的时间和测试的投资。XSLT是设计用于解决类似这样的问题。XSLT可以由XSLT处理程序阅读，XSLT处理程序替你处理XML文档——你需要做的不过是创建一个XSLT样式单，它规定你希望应用于从一种文档转换为另一种文档的规则。无需编程——这正是它吸引许多人的原因，甚至是资深的程序员。对于棒球的例子，你必须做的不过是编写一个规定想要做什么的XSLT样式单，而让XSLT处理程序完成余下的工作。

除了转换一个XML文档至另一个XML文档，也可以转换XML文档至其他类型的文档，诸如HTML文档、多格式文本（RTF）文档和使用XSL-FO的文档等。也可以转换XML文档至其他基于XML的语言，诸如MathML、MusicML、VML和XHTML等——全部不需要编程。

按照多种方式，XSLT可以发挥类似于像SQL（Structured Query Language，结构化查询语言，著名的数据库访问语言）的数据库语言的功能，由于它支持从XML文档中提取自己想要的数据，非常类似于应用一个SQL语句于数据库。某些人甚至视XSLT为网上SQL语言，而且，如果你熟悉SQL，它向你提供一些XSLT可以访问的无限范围的概念。例如，使用一个XSLT样式单，可以从一个XML文档中提取一个数据子集，创建一个长文档的完整内容表格，查找与特定测试匹配的全部元素——诸如位于特定邮政代码区域的消费者——等等。而且你可以一步完成所有的工作！

1.2.2 XSL-FO：XSL格式化对象

XSL的其他组成部分是XSL-FO，它是XSL的格式化语言部分，而且读者将会在本书中尝试

XSL-FO。可以使用XSL-FO定义在XML文档中如何显示数据,具体到页边距的大小、字体、对齐方式、标题和脚标大小以及页面宽度。在格式化一个XML文档时,存在数以百计的项目需要考虑,因此,XSL-FO远大于对应的XSLT。

在另一方面,由于其极端复杂性,XSL-FO仍然不是很普及,自然无法与XSLT相提并论。目前支持XSL-FO的软件不多,而比较完整地实现XSL-FO的则一个都不存在。正如XSLT的最常见应用是转换XML为HTML,XSL-FO的最常见应用则是转换XML为格式化的PDF(Portable Data Format可移植的数据格式),即Adobe Acrobat所使用的格式。本章结尾将给出这样的一个例子,第11章也包含这样的一个例子。

1.2.3 W3C规范

XML和XSL规范都是由W3C发布的,而这些规范正是本书将要介绍的。根据国际协议,W3C规范不被称为标准,而标准则是由政府认可的机构创建的。相反,W3C以发布一种新规范的需求为开始。需求是目标,并且列出一个关于此规范将要处理内容的分类评述,但是,在此阶段并不编写规范。接下来,W3C先以工作草案的形式发布规范,任何人都可以对此工作草案作进行评论,然后才作为候选规范,它仍然要接受评论,此后才最终成为规范,至此才告结束。

下面的列表包含本书将要使用的、与XSLT相关的W3C规范,以及读者可以查找它们的网址:

- 完整的XSL候选规范 www.w3.org/TR/xsl/。这是一个很大的文档,它描述XSL的所有候选规范。
- XSL变换1.0规范 www.w3.org/TR/xslt。XSLT的功能是转换XML文档的内容为其他的文档,正是它让XSL变得如此普及。
- XSLT 1.1工作草案 www.w3.org/TR/xslt11。这是XSLT 1.1工作草案,这将被升级为规范——W3C计划将XSLT 1.1的功能添加至XSLT 2.0。
- XSLT 2.0需求 www.w3.org/TR/xslt20req。W3C已经发布XSLT 2.0的一组目标,包括对XML大纲的更多支持。
- XPath 1.0规范 www.w3.org/TR/xpath。在XML文档中,XPath可以用于定位并且指向特定的节和元素,以便对它们进行处理。
- XPath 2.0需求 www.w3.org/TR/xpath20req。XPath正被升级以更多地支持XSLT 2.0。

1.2.4 XSLT版本

与作为一个整体的XSL规范相比,对于XSLT规范的考虑更为积极。XSLT 1.0规范是在1999年11月16日最终得以确认,而正是该版本形成了当今XSLT的基石。

接下来出现的是XSLT 1.1工作草案,尽管最初它是计划继续发展并成为一种新的规范,但是,W3C的一些成员已经着手准备XSLT 2.0。一段时间之后,W3C决定取消XSLT 1.1规范,这意味着XSLT 1.1工作草案将不会继续发展——它将一直保持为工作草案的形式,而不会成为一种规范。换言之,将不会存在正式的1.1版XSLT。

不过,W3C也声称,它将集成XSLT 1.1工作草案所实现的许多功能至XSLT 2.0中,因此,笔者将在本书中对XSLT 1.1工作草案进行一番考察。在讨论某些在XSLT 1.1工作草案中新引入

的内容时，笔者将这些材料标识为“XSLT 1.1工作草案专有”。

下面给出的是在XSLT 1.1工作草案中对XSLT 1.0所作的改动。注意，给出此列表只不过是参考，对于读者而言，这些材料中的大部分可能毫无意义：

- 在XSLT 1.0中支持的结果树片断数据类型被去掉。
- 输出方法不再完全拥有添加名字空间节点的自由，因为自动采用了一种名字空间处理修正版本。
- 增加了对XML基础的支持。
- 现在通过<xsl:document>元素支持多输出文档。
- 现在允许<xsl:apply-imports>元素拥有参数。
- 现在可以用<xsl:script>功能定义扩展功能。
- 现在允许扩展功能返回外部对象，此外部对象不对应于任何XPath数据类型。

本书既讨论XSLT 1.0规范，也讨论XSLT 1.1工作草案。实际上，W3C已经更领先一步发布了XSLT 2.0的需求。而且本书将介绍笔者了解的XSLT 2.0的内容。下面列表给出对XSLT 2.0目标的一个概述：

- 增加对通过XSLT使用XML大纲类型的内容的更多支持。
- 简化字符串内容的操作。
- 简化XSLT的使用。
- 改善国际化支持。
- 保持与XSLT 1.0的向后兼容。
- 支持改进的处理程序效率。

尽管XSLT 2.0不会在短时间内出台，但是，在讨论相关论题时，笔者仍将尽力去介绍它。例如，W3C为HTML准备的后继者是基于XML的XHTML。在XSLT 1.0和XSLT 1.1工作草案中，不存在对由XML至XHTML变换的支持，因此，本书将从零开始创建该变换。不过，在XSLT 2.0中提供了这种支持，而且本书将在讨论XHTML时说明此事实。

上面为我们提供了一个概述，并且做了准备。现在该是切入正题的时候了，XSL是设计用于处理XML文档的，因此，这里首先将概述一下XML文档的结构。读者将要处理XML文档，但是XSL样式单本身实际上也是XML文档，这正是读者在编写XSL时应该记住的一点。本书假设读者同时具备XML和HTML方面的一些知识。（正如前面讲过的，本书是New Riders出版的《Inside XML》一书的姐妹篇。）

1.3 XML文档

对于读者而言，了解XML文档如何运作很重要，因此，利用这一节给读者补补课。下面是一个XML文档的例子，对其进行考察：

```
<?xml version="1.0" encoding="UTF-8"?>
<DOCUMENT>
  <GREETING>
    Hello From XML
  </GREETING>
```