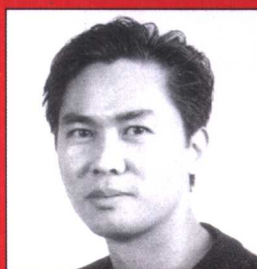
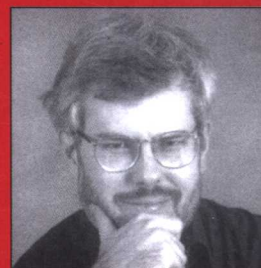
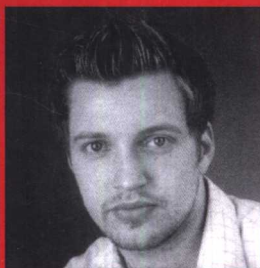
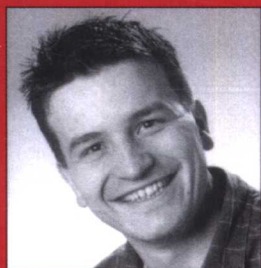




PROGRAMMER TO PROGRAMMER™

Professional XSL

Wrox 程序员参考系列



XSL

高级编程

(美) Kurt Cagle 等著

韩平 程永敬 董启雄 等译

p2p.wrox.com 提供在线支持



机械工业出版社
China Machine Press



乐思

Wrox 程序员参考系列

XSL 高级编程

(美) Kurt Cagle 等著

韩 平 程永敬 董启雄 等译

程永敬 董启雄 审校



机械工业出版社
China Machine Press

本书以一种实用的、教程样式的方法来讲解 XSLT、XPath 和 XSL-FO 规范的核心基础。主要内容包 括：如何创建结构合理的模块化样式表来生成所需要的输出文档，如何对数据进行更改、过滤及排序，以及与其他内容进行合并而显示出来。

对于非常熟悉 XML 数据及其结构的开发人员，以及转换数据和将样式表应用于 B2B 和 Web 应用程序的人员来说，这是一本有益的参考书。

Kurt Cagle, et al: Professional XSL.

Authorized translation from the English language edition published by Wrox Press.

Original copyright © 2001 by Wrox Press.

All rights reserved.

Chinese simplified language edition published by China Machine Press.

本书中文简体字版由英国乐思出版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：01-2001-4402

图书在版编目（CIP）数据

XSL 高级编程/（美）卡格尔（Cagle, K.）等著；韩平等译. - 北京：机械工业出版社，2002.4

（Wrox 程序员参考系列）

书名原文：Professional XSL

ISBN 7-111-09947-8

I .X… II .①卡…②韩… III . 可扩充语言, XSL-程序设计 IV . TP312

中国版本图书馆 CIP 数据核字（2002）第 013208 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：朱 砂 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 4 月第 1 版第 1 次印刷

787mm×1092mm 1/16·42.75 印张

印数：0 001-4 000 册

定价：69.00 元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换

译者序

首先，我们非常荣幸地将这些最新的资料翻译过来，介绍给大家。我们也真心希望我们的工作能够帮助你迅速掌握最新的技术。

我们知道，XML 是一种为实现与平台无关的数据存储和交换、实现将内容与显示分离而建立的标准。它的普及要归功于这种语言的灵活性和以各种方式对数据进行重用的能力。而我们要介绍的 XSL 是用于 XML 的一项关键技术，它由两部分组成：XSLT 是一种正式语言，它用于将 XML 从一种格式转换为另一种格式，既可以用于对数据进行重构/选择性处理，也可以用于为显示而呈现数据；XSL-FO 是一个提议用语，主要是为了将与文档为表示而进行的排列相关的信息进行合并。XPath 是一种与 XSL 转换紧密相关的标准，这种语言主要用于对 XML 文档中的指定部分进行访问。

本书首先对什么是 XSL、XSL 能干什么进行了解释，然后以一章的篇幅对用于在一个 XML 树中指定位置的 XPath 语言进行了描述。第 3 章和第 4 章中，主要对作为 XSL 的重要组成部分的 XSLT 进行了介绍，包括该语言的核心内容和一些为了增强应用程序而附加的一些功能和技术。此外，本书还对通用的结构化转换以及为了显示而将 XML 进行转换等内容进行了说明和演示。最后本书还介绍了 CSS 和 XSL-FO 的概念和关键元素、XSLT 和 XML 大纲的使用等内容。在本书的附录中，还提供了各种丰富的内容和资料来帮助我们学习。

本书主要由韩平、程永敬、董启雄、张宝玲、张怀利、庄锦山等人翻译，参加翻译的有程永敬、董启雄、韩平、张宝玲、张怀利、庄锦山、叶伟、刘学来、汪洋、唐雪飞、程思羽、于海龙、马君艳、董骥、叶雨涵、吴嵘、任卓涛、李德峰、张永明、胡子飞等。本书涉及的技术和内容较新，其中有些甚至还没有正式的标准；再加上个别章节由于作、译者的水平和理解、阐述的水平等限制，所以本书可能还存在一些这样或那样的问题。我们真心地欢迎各位读者对我们的工作进行批评和指正。我们的邮件地址是：freediscovery@263.net。

自由探索工作组

FreeDiscovery WorkGroup

2001 年 11 月

作者简介

Kurt Cagle (库尔特·卡格尔)

Kurt Cagle 是 XML 和相关问题的专门作家和开发人员。他已经写了八部书和一百多篇文章，从 Visual Basic 编程到 Internet 对社会的影响，他的主题都有涉及。他还为 Microsoft、Nordstorm、AT&T 等公司提供咨询。他还帮助发行 Fawcette 的 XML 杂志，并在最近两年成为 DevX、DHTML 和 XML 的专家。

Michael Corning (迈克尔·考宁)

Michael Corning 是 Microsoft 公司的一名 Memetic 工程师，致力于基于 .Net 框架为应用程序中心构建一种软件测试体系结构。夜间他为 Wrox 出版社、ASPToday.com、ActiveWeb Developer 杂志写作，并在 XML Developer 杂志开设了一个每周专栏，叫作“XSLT 狂热者的自白 (Confessions of an XSLT Bigot)”。他的第一本书是《Working with Active Server Pages》，该书 1997 年由 Que 出版社出版。Corning 还在全球大会上发言，讲述基于 Schema (大纲) 编程的好处。闲暇时，他喜欢约上妻子 (他的崇慕者) 外出，或者驾驶他的 West Wight Potter (西部陶工) “Finally” (终结) 号航行，或者读书。他是三位英俊儿子引以自豪的父亲，这三个孩子分别是：Christian、Seth 和 Casey；此外，他还拥有三只猫：Minnie、Dixie、和 Nutmeg。

致谢：首先，我要感谢我亲爱的妻子 Katy。她深深了解作为作家妻子的生活。还要感谢我的老板 Dan Casey 和 Tony Godfrey，感谢他们对我的信任，并让我在白天时间之外做这么多的事，就像写书。最后，感谢我的好朋友和 XML 小组的同事，特别是 Jonathan Marsh、Derek Denny-Brown 和 Chris Lovett，与你们这些人共同工作是我职业生涯中的一种荣耀。

Jason Diamond (杰森·迪曼德)

“Jason Diamond 热爱他的母亲”。

Teun Duynstee (特恩·坦斯特)

Teun Duynstee 是 Macaw 公司 (荷兰的一家一流的 Web 构建和咨询公司) 的一名首席开发人员，专攻在 Windows DNA 平台上构建复杂的企业 Web 应用程序。他最喜欢干的事就是热心为别人提供新技术方面的帮助，还喜欢晚睡。

你可以通过电子邮件与他联系：proxsl-feedback@duynstee.com。

Oli Gauti Gudmundsson (奥利·高提·伽德曼德森)

Oli 为 SALT 公司工作，他是 SALT 系统的两个首席系统构造师之一，还担任纽约的开发主管。他现在的工作是将 XML 和 XSL 整合到 SALT 的编辑和目录管理系统中。他还是冰岛大学计算机科学 Java 课程的导师，而 Java 是他的强项（及兴趣）。作为一种“业余爱好”，他正在努力获得计算机工程的理科学士学位。

他的祖国是冰岛，但他现在和女友 Edda 一起居住在纽约。你可以通过电子邮件和他联系：oli.gauti@salt.is。

Michael Mason (迈克尔·马逊)

Mike 于 1999 年毕业于牛津大学，获计算学士学位，并开始为当地的一家叫做 DecisionSoft 的公司工作。Mike 参与了创建 XML 脚本语言及相关工具的工作。现在，他就职于伦敦一家名为 Digital Rum 的公司，使用 Java、XML 和 XSLT 来开发移动商务应用程序。

Mike 喜欢滚轴溜冰、空手道、在伦敦以外的任何地方驾驶自己的汽车，余暇时间就摆弄 Linux。他也是 Buffy the Vampire Slayer 游戏的超级迷恋者之一，虽然这不是一个时髦的爱好。

可以通过下面的邮件地址来与 Mike 通信：mgm@eskimoman.net。

Mike 感谢 DecisionSoft 公司的所有同仁，尤其是 Philip 和 Ben，因为当 Mike 刚开始工作的时候，他们是很好的工作伙伴，还有 Gareth，感谢他解答了 v2 的问题。Mike 还要感谢他在牛津和伦敦的一帮伙伴感谢他们认为他可以是一名作家，感谢在他写这一章内容时来自他们的鼓励。Wrox 出版社的人们给了 Mike 很多帮助，他要感谢 Marsha，感谢她回答了所有他作为新手所提出的问题。

Jon Pinnock (约翰·比诺克)

Jonathan Pinnock 早在其学生时期，就其学校的 PDP 8/e 上，使用着 4KB 的内存，编写 Pal III 的汇编程序。现在，他花大部分的时间来开发和扩展日益成功的平台集成产品，这是他的公司 (JPA) 为开拓面向金融服务社团的市场所做的主要工作。他似乎将他的剩余时间都投入到了为 Wrox 的写作工作中，当然他偶尔也简单地向他长期受苦的妻子和两个孩子问个好。

JPA 的主页是：www.jpassoc.co.uk。

Paul Spencer (保罗·斯本瑟)

作了三年的自由 XML 顾问之后，Paul 在 2000 年底决定不再过一周七天都要工作的日子了，并找到了他现任 CTO 的 alphaXML 有限公司。以 Henley-on-Thames 为基地，他带领一个扩展小组在全世界范围内提供 XML 服务。Paul 是一个 XML 建议者，他为英联邦税务局、政府网关和电子外交办公室提供服务。因此，他实际的工作量并没有减少。当然，他也决定保留空闲时间来航海，在这个领域他试图保持领先的地位。他的航行水平是出名的，这是由于他有一次击败了刚刚击败 Dennis Conner 胜利而归的一个人。

Paul 要感谢所有为他提供他所喜爱的工作的人们，还要感谢 Wrox 出版社将本书的写作安排在非航海的季节里。

Jeff Tang (杰弗·唐)

Tang 是 Tellngo 有限公司的首席软件工程师，在这里，他使用 Nuance 和 Speechworks 技术及 Java、C++，还有 VoiceXML 来设计并开发具有语音功能的企业应用程序。之前，Jeff 在 Sprint、Informix、Cerner 和 Perceptive Vision 等公司做过六年的高级软件工程师、技术带头人和咨询等工作。他帮助设计并开发了许多企业系统，包括医疗信息系统、文档镜像产品、基于 Web 的计算机电话应用程序、Internet 媒体应用程序，以及企业订购管理系统。在此之前，他在中国做了大概五年的软件工程师，参加开发了自然语言处理/机器翻译系统和数据库应用程序。

Andrew Watt (安德鲁·瓦特)

Andrew Watt 是一个独立的咨询工作者，除了探讨那些别人没有试验过的技术问题外，他很少有其他兴趣。在 20 世纪 80 年代中期，他就用 6502 汇编语言和 BBC Basic 语言写了他的第一个程序，当时他已经尝试过 Pascal、Prolog 和 C++ 等语言。最近，他又对 Web 相关的技术（包括 Lotus Domino、Java 和 HTML 等）产生了兴趣。他现在的兴趣是可扩展标记元语言（XMML）的各种应用程序（XMML 在不太严密的情况或误解的情况下也会叫做 XML）。现在他还关注着 SVG、XSL-FO、XSLT、CSS、XLink、XPointer 等等，以及当这些工具真正完全一起工作时令人激动的前景。他还刚刚开始涉足于 XQuery。他的学习范围涉猎如此之广，用他的话说，这叫作“活到老学到老”。

在闲暇时，他也在思考 Web 技术为现实世界中的人们带来的影响。语义学的 Web 将会有什么影响？这些如何不同于一些知识产权的事件呢？

感谢上帝和天堂赋予人类观察、思考和感觉的能力。感谢我的父亲在生活上对我的大量教导。

前 言

作为用来构造和描述 Web 数据、甚至于构造和描述企业应用程序数据的语言，可扩展标记语言（XML）已经得到了广泛的认可。XML 的魅力在于它是基于文本的（因此与平台无关）和可扩展的（允许对底层数据进行更有意义的描述）。它是描述数据的一种方法，而不管数据最终如何使用，同时将内容与表示分离，因而使其非常适用于需要为不同的最终用途而对数据进行重新规划的场合。但当我们想要对这些数据进行某些处理时，例如从其中抽取部分数据或者以一种非 XML 格式来对其进行表示时，我们需要的就不仅仅是单一的 XML 规范了。

可扩展样式表语言——XSL，最初提交给环球网联盟（W3C）时就是为了提供一种对 XML 数据进行格式化的手段。对 XML 进行格式化的过程包括下述内容：

- 用来指定使用源文档的哪一部分的方法。
- 为了得到理想的输出文档，用来描述对被使用的源数据进行何种操作的方法。
- 将命令结合到输出文档中去的方法。

XSL 规范很快就产生了两个分支：XSL 转换（XSLT）和 XSL 格式化对象（XSL-FO），在整个的转换过程中，它们分别起着不同的作用。

第三种相关的规范是 XPath，它是一种用来对 XML 文档中的各部分（及位置）进行标识的一种语言。XML 数据是作为一种层次式、树状节点结构来表示的，而 XPath 提供了一种用来定义表达式和包含函数的语法，它可以深入到 XML 的树结构中直至到达指定的节点。

XSLT 是用来对找到那些指定节点时所将采取的动作进行声明的一种语言。它用于书写样式表，这些样式表由一些模板组成，当在源文档中找到指定的大纲时，将会应用这些模板——当找到匹配的节点时，接着就会在模板中找到相应的指令，例如选取并存储输出的数据。通过这种方式，我们可以生成输出文档。XSLT 语言有一组单元及函数，用于完成诸如复制一个单元、将文本内容或属性值抽取到输出文档中，以及输出一个节点的排序列表等任务。XSLT 可以用于许多应用程序，这些应用可以是任何需要对 XML 数据在结构上进行修改或转换成指定格式的场合。

XSL-FO 定义了一组格式化对象，这些对象可以包含到输出文档中，以精确地指出数据应该如何被表示。目前，虽然有些工具可以将 XSL-FO 翻译成其他格式，但可以理解和正确翻译 XSL-FO 的 Web 浏览器却还很少。通常 Web 应用程序并不使用 XSL-FO，而是将 HTML（或者 CSS、WML、VoiceXML、SVG、JavaScript 等）插入到输出文档中在“浏览器”中显示。

在许多类型的应用程序中 XML 数据并不需要以用户友好的形式呈现。例如，在商务对商务（BtoB）的应用场合，不同的商业合作伙伴使用的应用程序相互之间需要通信并共享数据，但每个合作者都各有其首选的数据格式。XSLT 可以完成这些转换，从而使得企业可以高效而广泛地和其他合作伙伴进行合作。

即使对于开发人员来讲，虽然他们不需要呈现或交换数据，例如那些纯粹处理数据存储的

人，但他们仍然会发现 XSL 的用处。依据 schema（大纲）来验证数据以及从大纲中生成 XML 文档，这正是使用 XSLT 来帮助实现自动化并加速开发进程的两个例子。

本书的内容范围

本书将为你讲解所有 XSL 技术“家族”中三个部分（XSLT、XSL-FO，以及相关的 XPath）的目的及语法。更为重要的是，本书使用大量的应用程序来演示如何在实际工作中使用 XSL，包括将由多个提供商提供的产品信息转换为统一的产品目录、为移动用户动态创建 WML Web 页面，以及生成 XML 内容的图形化表示等。当你读完了这本书的时候，你将清楚地理解什么是 XSL、它的用途，以及如何着手将其使用到你已有的或将来开发的应用程序中。

本书的读者对象

本书的读者应是那些已经熟悉 XML 的程序员。如果你对 XML 知之甚少，那么推荐你阅读《Beginning XML》（ISBN 1-861003-41-2）（XML 初级编程）或《Professional XML》（ISBN 1-861003-11-0）（XML 高级编程），这两本书都已经由 Wrox 出版社出版。

本书适合下列两类开发人员：

- 需要将数据转换为 XML 格式或者要将 XML 格式的数据转换为其他格式的。
- 需要将样式应用到其 XML 数据的。

可以在其中使用 XSL 的应用程序几乎数不胜数，且覆盖了许多领域，包括商务对商务的交换、基于 Web 的目录发布以及 XML 数据存储器的管理等。

本书的结构

本书的各章节是这样安排的：在深入了解 XSL 典型的实际应用之前，首先讲解 XSL 的基础知识。全书中提供了大量的小例子来对所讲的概念进行演示。内容组织如下：

- 第 1 章对 XML 知识的复习回顾，并详尽解释了之所以开发一种转换语言来处理 XML 的背景。该章介绍了 XSL 技术家族的每一个成员。初步介绍了 XSLT 样式表的基本概念及其处理方式，并提供了几个关于所介绍 XSL 的实例，用来说明 XSL 可以用于哪些地方。
- 第 2 章提供了一份 XPath 语言语法的完整参考，同时以实例来演示其用法。当使用 XSLT 时，XPath 是必不可少的，它用来定位并识别给定 XML 文档的指定部分，同时，善于使用 XPath 可以提高 XSLT 样式表的性能。
- 第 3 章对 XSLT 语言的核心部分进行了解释并加以演示——即为创建应用于多种应用程序的能完成各种功能的样式表所需掌握的内容。
- 第 4 章学习使用 XSLT 和用来提高样式表的能力和效率的更高级的技术。该章内容中包括了变量和参数的使用。
- 第 5 章继续介绍 XSLT 的高级技术，学习如何插入外部模板。在该章的结尾处，对 XSLT 规范的未来版本中所做的一些改进进行介绍。
- 第 6 章从不使用 XSL 而转换 XML 的观点出发，对 XML 文档对象模型（DOM）以及用于 XML 的简单 API（SAX）这两者进行讨论。这些编程接口可以用作 XSLT 的替代或者作

为 XSL 转换的前身，该章中还包含何时使用哪项技术最为合适的有关讨论。

- 第 7 章包含 Microsoft 的 MSXML3 以及 .Net 架构产品有关 XSL 特性的一些资料。该章中的示例代码使用 MSXML 来对数据进行分析，并使用 .Net 架构的基本类来对 XML 数据进行流操作、定位、选择及进行转换。
- 第 8 章基于作者的实际经验，通过对一个实例的研究，指出一些在开发样式表时应注意的问题，并对如何提高 XSLT 样式表的性能给出了一些建议。
- 第 9 章讨论由 W3C 所建立的 XML 样式化技术。由于级联样式表（CSS）得到现代浏览器的广泛支持，我们首先对其进行讨论。XSL-FO 也是基于类似的原理，但其目标是更多的输出格式。该章对这两者都进行了详细的解释，从而使你能够在你的转换中使用它们。
- 第 10 章介绍 XSLT 的典型应用，而不涉及为显示而对输出进行样式化，换句话说，就是对 XML 数据进行结构上的转换（例如抽取、修改、添加及合并）。这些内容包括商务词汇间的转换、使用样式表生成新的样式表以及使用私有数据库中的数据。
- 第 11 章演示以较为通用的方法来为 XML 数据加入描述信息（换句话说，就是使用 HTML 而不是 XSL-FO）。该章所开发的示例程序展示了如何使用样式表，结合 CSS，并通过恰当地在服务器端或客户端进行转换，来动态生成 Web 页面的 HTML 及基于 WML 的表示法。
- 第 12 章讲解当前图形格式的背景，尤其是可缩放矢量图形（SVG）语言。然后演示了如何编写生成 SVG 的样式表来产生图形、格式化文本及描绘各种形状。
- 第 13 章介绍 VoiceXML 语言的语法，这是一种用于有声音的 Web 应用程序的新的基于 XML 的方法。该章介绍了相关的开发工具。该章的例子演示了如何使用 XSLT 来加速生成新的 VoiceXML 的过程，当应用程序是基于不断变更的数据源时，这就更加有用了。
- 第 14 章将对一些有用的方法进行演示，根据这些方法，可以将 XSLT 和 XML 大纲进行组合，例如，使用 XSLT 来从一个大纲生成一个 XML 文档实例（或反之），创建样式表来依据一个大纲验证 XML 文档，或使用 XSLT 规则表达式来规范数据完整性。
- 第 15 章给出了关于 XSLT 转换的最后一个例子，这次的基础是 Petri 网计算机。Petri 网是一种用于图形化和数学化建模系统的工具，在本章 XSLT 用来将 Petri 网标识语言（PNML）转换为用于图形表示的 SVG。为使本书更加完整，该章还提供了略微不同的思路来考虑 XSL 编程，为你未来的编程提供一些启发。XSL 是基于大纲编程（这是一个基于 XML 应用程序开发的术语）的关键部分，该章的讨论将对为什么 XSL 对于使用 XML 的开发人员来说是至关重要的这一问题进行进一步的补充。
- 第 16 章演示了一个典型的 XSLT 实际应用程序，其中使用了许多本书各处所提供的技术。它展示了 XSLT 是如何能够用在 Web 应用程序的核心部分，向最终用户提供底层内容的友好界面，并提供了实现在文档各部分间轻松定位的功能。该程序既可以成批转换来生成静态 HTML，又可以根据需要来实时转换生成一些特定的内容。此外，该程序还进一步巩固了前面各章中的知识，样式表的组件化和参数化促进了程序的可重用

性和可维护性。

在本书的附录中，我们还提供了一些相关的参考资料：

- 附录 A 提供了完整的 XPath 规范 1.0 版。
- 附录 B 提供了完整的 XSLT 规范 1.0 版。
- 附录 C 包括了 DOM 标准 2 核心的所有接口，包括用于处理 XML 文档的基础接口和扩展接口。
- 附录 D 列出并描述了 SAX 2.0 的所有接口和方法。
- 附录 E 内容包括如何获取并安装（可免费得到的）通用 XSL 处理程序，其中有许多是运行本书所提供示例代码所需的。本附录还介绍了其他一些可免费得到而有用的工具的样例，例如帮助你理解复杂样式表的工具。
- 附录 F 介绍了几种用来加速 XSLT 样式表开发的途径，并详细讨论了两个这样的工具：XSLTScript（一种用来快速生成 XSLT 的工具）和 XML Script（XSLT 的一种替代工具）。
- 附录 G 提供了一些在线资源和将来可能会用到的其他书籍。

本书中所用到的技术

由于 XML 是一种平台无关的语言（或者更确切地说是一种平台无关的语法），我们将尽可能地保持书中示例的平台无关性。尽管在一些章节中使用了 HTML、Java、Visual Basic、Visual C++、C#、ASP、JSP、JavaScript、Windows Script Host、Oracle 以及 XSQL 来演示一些特殊的编程技术，但这些方法可以毫不费力地移植为别的语言。尽管我们采用了一些常用的编程知识，但所有的代码在需要时还是给出了解释。

本书中提供的示例（除非特别声明）可用于常用的以及通用的处理程序，这将在本书的适当位置讲到。有关如何获取和安装这些处理程序的详细情况，参见本书附录。

用户支持

虽然我们已经努力使本书尽可能地准确并有趣味性，但实际上你仍然可能发现一些不尽人意的地方。请让我们知道你对本书的看法，你的反馈意见可以通过书后的反馈卡片寄给我们，也可以通过电子邮件联系，我们的邮件地址是：feedback@wrox.com。

源代码及其更新

你使用本书代码时，可以选择手工输入所有的代码。许多读者都采取这种方式，因为这是熟悉这些代码的一个好办法。

当然，我们已经将本书的这些源代码放在了我们的网站上，网址如下：

http://www.wrox.com/Books/Book_Down.sap?isbn=1861003579

如果你属于那种喜欢自己输入代码的读者，那么可以使用我们的文件来检查你所获得的结果，如果你觉得你可能输入得有错，那么这是你第一个用来确定是否错误的方法。如果你属于那种不愿意输入代码的读者，那么你必须从我们的站点下载这些源代码。

无论是哪种方式，这些源代码都有助于你修改和调试程序。

勘误表

我们付出了全部努力来确保在全书的文本和源代码中不出现错误。然而，人总是会犯错误的，因此我们觉得需要通知你一些已经发现并纠正的错误。全书的勘误表可以在 <http://www.wrox.com> 找到。如果你发现了任何还没有报告的错误，请让我们知道。

我们的站点承担着收集信息和支持的任务，另外还包括我们所出版的所有图书中的源代码、图书的一些样章、将要出版的图书的预览以及相关主题的文章和观点。

P2P

在线论坛 p2p.wrox.com 提供了程序员对程序员的广泛技术支持。你可以订阅其邮件列表并周期性接收摘要，或者可以查找其存档文件来找出可解决你所遇到编程问题的解决方案。

目 录

译者序	
作者简介	
前言	
第 1 章 XSL 简介	1
1.1 XML 概述	2
1.1.1 DTD 和 Schema	3
1.1.2 XML 名字空间	5
1.1.3 转换/格式化的必要性	6
1.2 XSL 简介	7
1.2.1 XSLT 样式表	8
1.2.2 XSLT 处理模型	8
1.3 工具	10
1.4 转换 XML 的其他手段	11
1.5 XSL 的应用	11
第 2 章 XPath —— XML 路径语言	13
2.1 XPath 简介	13
2.1.1 XPath 的应用目标	13
2.1.2 XPath 的工作原理	14
2.1.3 XPath 数据模型	16
2.1.4 其他的 XPath 概念	19
2.1.5 XPath 函数	23
2.2 XPath 语法	24
2.2.1 非简写 XPath 语法	25
2.2.2 简写语法	32
2.2.3 两种语法的注意事项	35
2.3 实例	36
2.3.1 下载并安装 Instant Saxon	36
2.3.2 XML 源文档	36
2.4 XPointer、XLink 和 XQuery	43
2.4.1 XPointer	44
2.4.2 XLink	44
2.4.3 XQuery	45
2.5 本章小结	45
第 3 章 XSLT 基础	46
3.1 XSLT 处理	46
3.1.1 XSLT 处理模型	46
3.1.2 处理一份文档	47
3.1.3 推模型和拉模型	51
3.2 关于名字空间的话题	59
3.3 XSLT 元素	61
3.3.1 < xsl: stylesheet >	61
3.3.2 < xsl: output >	62
3.3.3 < xsl: template >	65
3.3.4 < xsl: apply-templates >	66
3.3.5 < xsl: value-of >	66
3.3.6 < xsl: copy > 和 < xsl: copy-of >	67
3.3.7 控制流程元素	70
3.3.8 < xsl: sort >	76
3.3.9 < xsl: number >	79
3.3.10 < xsl: text >	80
3.3.11 < xsl: element >	81
3.3.12 < xsl: attribute >	83
3.4 默认模板	84
3.5 模板匹配冲突	86
3.6 XSLT 函数	86
3.6.1 position () 和 last ()	87
3.6.2 name ()	89
3.6.3 count ()	90
3.7 本章小结	90
第 4 章 模块化的 XSLT	91
4.1 变量的取值	92
4.1.1 < xsl: variable > 的 select 属性	94
4.1.2 变量选择和节点集	96
4.2 参数的功能	100
4.2.1 参数和外部程序的关系	101
4.2.2 建立命名模板	109
4.3 本章小结	120
第 5 章 外部 XSLT	121
5.1 文档和超空间世界	122

5.2 过于繁复的样式表	130	7.2.3 XmlNavigator	230
5.2.1 导入的重要性	130	7.2.4 XslTransform	234
5.2.2 OO XML 的起点	132	7.2.5 .NET 中的 MSXML	236
5.2.3 超级应用导入	135	7.3 本章小结	238
5.2.4 Import 和 Include, 谁更好	137	第 8 章 优化性能	239
5.2.5 可扩展性	138	8.1 性能陷阱	239
5.3 样式表的将来: XSLT1.1	147	8.2 第一阶段: 转换中的脚本	240
5.3.1 废除 node-set ()	147	8.2.1 克服 XSLT 语言限制	240
5.3.2 多个输出流	148	8.2.2 遗漏操作符的情况	241
5.3.3 通用脚本块和扩展	149	8.2.3 从有脚本到无脚本的转换	242
5.3.4 导入模板的参数化	150	8.2.4 早期的 XSLT	243
5.4 XSLT 2.0 以及其他	151	8.2.5 中间 XSLT	249
5.5 本章小结	153	8.2.6 简单小结第一阶段的性能 问题	252
第 6 章 使用 DOM 和 SAX 进行 转换	155	8.3 第二阶段: 低劣的设计、功能延伸和 野心	253
6.1 什么时候使用 DOM 或者 SAX 来代替 XSLT	155	8.3.1 事倍功半的做法	254
6.2 XML DOM	157	8.3.2 静态代码分析	256
6.2.1 快速了解对象和接口	157	8.3.3 第二个性能改进	257
6.2.2 文档对象模型	158	8.3.4 第三个性能改进	259
6.2.3 使用 Java 进行 DOM 编程	159	8.3.5 第二阶段性能问题的小结	261
6.2.4 使用 VBScript 进行 DOM 编程	166	8.4 第三阶段: 低劣的体系结构	261
6.3 SAX	169	8.5 补充	264
6.3.1 基础知识	169	8.6 一些惊奇	265
6.3.2 SAX 的简短历史	170	8.7 综述	266
6.3.3 什么时候使用 SAX	171	8.8 本章小结	267
6.3.4 使用 Java 开发一个 SAX 2.0 转换 应用程序	171	第 9 章 CSS 和 XSL-FO	268
6.3.5 使用 Microsoft 的 SAX 产品	188	9.1 将样式应用到 XML 的技术	268
6.3.6 过滤和编写 XML	195	9.2 CSS 原理	271
6.3.7 SAX 的其余内容	203	9.2.1 关于级联样式表的说明	271
6.4 本章小结	205	9.2.2 CSS 格式化模型	281
第 7 章 Microsoft XSL 技术	206	9.3 将 CSS 应用到 XML	283
7.1 MSXML3	206	9.3.1 将 XML 链接到样式表	283
7.1.1 安装	208	9.3.2 SVG 中的 CSS	292
7.1.2 XML 文档	209	9.3.3 与 XSLT 一起使用 CSS	294
7.1.3 高级转换	217	9.4 XSL-FO —— XSL 格式化对象	297
7.2 .NET	224	9.4.1 XSL-FO 术语	297
7.2.1 XmlReader	226	9.4.2 XSL-FO 概况	298
7.2.2 XmlDocument	229	9.4.3 XSL-FO 网页	299
		9.5 XSL-FO 文档的结构	301

9.5.1 利用 XSLT 创建一个 XSL-FO 文档	305	12.6 本章小结	403
9.5.2 使用 XSL-FO 创建 PDF; FOP	312	第 13 章 转换为 VoiceXML	405
9.5.3 展望	314	13.1 具有声音能力的应用程序	405
9.6 本章小结	315	13.2 VoiceXML 解决方案	407
第 10 章 通用结构转换	316	13.2.1 VoiceXML 示例文档	408
10.1 转换 XML 树	316	13.2.2 VoiceXML 元素	412
10.1.1 创建产品目录	317	13.2.3 建立环境	414
10.1.2 交付购货订单	322	13.3 一个更高级的 VoiceXML 应用程序	416
10.2 以样式表生成样式表	332	13.3.1 定义语法	417
10.3 从关系型数据库中转换数据	335	13.3.2 基于 Web 的应用程序	418
10.3.1 Oracle XSQL Servlet	336	13.3.3 通话流程	418
10.3.2 从 XSQL 页中创建 XML 结果集	336	13.4 用于 VoiceXML 的 XSLT	423
10.3.3 将 XML 数据插入到 Oracle 数据库中	338	13.4.1 XSLT: 开放源工具	424
10.4 本章小结	339	13.4.2 XSLT: 生成 VoiceXML 文档	426
第 11 章 以 XSLT 增加样式	341	13.5 本章小结	438
11.1 应用程序的要求	341	第 14 章 XSLT 与 XML 大纲	439
11.2 原型 1: 基本目录	342	14.1 XML 大纲概述	439
11.3 原型 2: 控制显示的详细内容	348	14.2 从大纲生成 XML 实例	441
11.4 版本 1: 按作者进行选择	357	14.2.1 术语	441
11.4.1 样式表的变化	357	14.2.2 大纲设计	442
11.4.2 脚本的变化	359	14.2.3 转换	446
11.5 版本 2: 服务器端版本	360	14.3 从 XML 文档生成大纲	450
11.5.1 ASP 代码	361	14.4 生成用于大纲验证的样式表	454
11.5.2 样式表	365	14.5 实施数据完整性约束条件	459
11.6 版本 3: 走向移动	366	14.5.1 规则表达式简介	460
11.6.1 WAP 和 WML	368	14.5.2 大纲中的规则表达式	461
11.6.2 list.xml	369	14.5.3 Xalan Java 扩展	462
11.6.3 book.xml	371	14.5.4 验证大纲	463
11.6.4 ASP 页	372	14.6 本章小结	464
11.7 应用程序集成	374	第 15 章 XSLT 和 XML 展望	465
11.8 本章小结	377	15.1 打破 XSLT 的框框	465
第 12 章 转换到 SVG	378	15.1.1 六个简单的物理机器	466
12.1 PGML、VML 和 SVG	380	15.1.2 六种简单的抽象机械	467
12.2 SVG 中的简单图形	381	15.1.3 XML 与其他计算技术的关系	472
12.3 以 XSLT 分解文本	391	15.2 第一个复杂的抽象机械	474
12.4 制作星图案	397	15.2.1 PNML 介绍	474
12.5 SVG 的将来	402	15.2.2 了解你的大纲	476
		15.2.3 变量的值	476
		15.2.4 最困难的部分	478

15.2.5 开发的最后阶段	480	16.4.3 创建一个索引	511
15.3 基于大纲的编程	485	16.4.4 家族树	514
15.3.1 基于大纲编程的三个级别	485	16.5 一个实时转换方案	514
15.3.2 XSLT 中的基于大纲的编程	486	16.5.1 ASP 页	515
15.3.3 Petri 网中基于大纲的编程	487	16.5.2 样式表	516
15.4 本章小结	488	16.6 结论	520
第 16 章 案例研究——在线		16.7 本章小结	520
内容发布	490	附录 A XPath 参考	521
16.1 源格式	490	附录 B XSLT 参考	537
16.2 一个简单的转换	492	附录 C XML 文档对象模型	569
16.3 添加独立样式表功能	495	附录 D SAX2.0: XML 简单应用	
16.3.1 第一次导入	496	编程接口	578
16.3.2 创建一个具有指向各章链接的		附录 E XSLT 处理程序和工具	616
目录	497	附录 F XSLT 便捷工具	627
16.4 将内容拆分到几个文件	501	附录 G 资源	660
16.4.1 拆分文件	502		
16.4.2 创建术语表	509		

第1章 XSL 简介

每个熟悉可扩展标志语言——XML的人都应该知道这样的事实，即XML是从数据的表示层分离数据的。而这种特性也正是使得XML如此有用并如此广泛使用的原因之一。XML语言的可扩展性使得开发人员可以以其自己定义的格式来描述和构造数据，这可通过定义其自己的词汇而实现，换句话说，XML的可扩展性允许开发人员创建自己的标记语言。此外，由于XML数据是基于文本的，因而是与平台无关的。这些特性共同提供了在Web和信息管理应用程序方面的巨大潜力。XML可以作为一种有用的手段，在应用程序和组织之间进行数据交换，或者为不同的应用程序进行数据的重用。

但是，单独使用XML还是有许多局限性。这是因为XML其自身是没有为显示而格式化的，并且，不同的商务应用使用不同的自定义格式，这些格式之间需要转换，我们需要一种方式，将数据从一种格式转换为另一种格式、选择并重构一份XML文档的相关部分、添加新数据、或在需要时将数据进行格式化显示。这就是可扩展样式表语言——XSL要完成的任务。

XSL是一种基于XML的语言，它通过描述表示样式表来对XML数据进行控制。最初的XSL规范仅仅是为了格式化XML，例如为了在Web上显示；但人们很快意识到这种处理过程涉及到了两种截然不同的步骤，而且都非常有用，这两个步骤是：将数据结构转换为其他XML格式，以及添加有关数据在某一特定设备上如何显示的信息。这样，当前形式的XSL主要由两个部分组成：

- XSL转换（XSLT），一种用于描述对XML数据进行结构化转换的语言。XSLT用来将一种XML格式转换到另一种格式。
- XSL格式化对象（XSL-FO），一种对格式化的语义进行描述的语言。XSL-FO用来描述当一份文档向用户显示的时候其将要如何呈现，而根本不对格式化的数据进行操作。XSL-FO处理程序将应用正确的格式。

与XSLT有关的另一个标准是：

- XPath，一种选择语法，允许我们在一份XML文档中选择节点。XPath可以用在XSLT中，来指定XML文档的哪一部分用于转换，或者用来选中XML内容来包含在输出文档中。

从1999年后期开始，XSLT和XPath就由W3C推荐，而XSL-FO仍然还处于草案拟定阶段。XSLT远比其他两项技术要成熟，现在已经有工具的支持了。

通过阅读本书，你将学习XSLT和XPath语言的语法，以及如何编写样式表来转换XML数据，这种转换可以是结构化的，也可以是转换为许多适合于不同终端应用的格式。尽管XSL-FO规范发展得没有XSLT那么快（在没有XSL-FO处理程序的情况下，人们将使用XSLT来生成HTML，以用于在Web页面上显示），并且工具很有限（但经常看到），我们还是要简要地介绍一下XSL-FO格式化对象，并学习如何使用它来定义我们想要的文档样式。