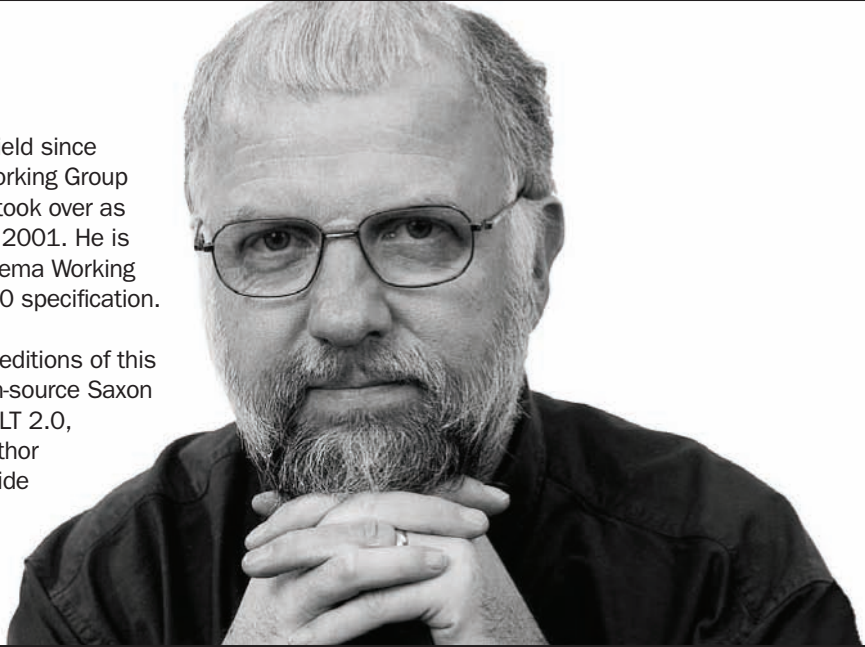


Michael Kay

Michael Kay has been working in the XML field since 1997. He became a member of the XSL Working Group soon after the publication of XSLT 1.0 and took over as editor of the XSLT 2.0 specification in early 2001. He is also a member of the XQuery and XML Schema Working Groups, and is a joint editor of the XPath 2.0 specification.

He is well known not only through previous editions of this book, but also as the developer of the open-source Saxon product, a pioneering implementation of XSLT 2.0, XPath 2.0, and XQuery 1.0. In 2004 the author formed his own company, Saxonica, to provide commercial software and services building on the success of the Saxon technology.



XSLT 2.0 and XPath 2.0

4th Edition

Programmer's Reference

Michael Kay



Updates, source code, and Wrox technical support at www.wrox.com



Programmer to Programmer™

[BROWSE BOOKS](#)

[P2P FORUM](#)

[FREE NEWSLETTER](#)

[ABOUT WROX](#)

Get more Wrox at **Wrox.com!**

Special Deals

Take advantage of special offers every month

Unlimited Access . . .

. . . to over 200 of our books in the Wrox Reference Library (see more details online)

Meet Wrox Authors!

Read running commentaries from authors on their programming experiences and whatever else they want to talk about

Free Chapter Excerpts

Be the first to preview chapters from the latest Wrox publications

Forums, Forums, Forums

Take an active role in online discussions with fellow programmers

Browse Books

.NET
SQL Server
Java

XML
Visual Basic
C# / C++

Join the community!

Sign up for our free monthly newsletter at
newsletter.wrox.com

XSLT 2.0 and XPath 2.0

Programmer's Reference

4th Edition

Part I: Foundations	1
Chapter 1: XSLT in Context	3
Chapter 2: The XSLT Processing Model	41
Chapter 3: Stylesheet Structure	89
Chapter 4: Stylesheets and Schemas	151
Chapter 5: Types	185
Part II: XSLT and XPath Reference	225
Chapter 6: XSLT Elements	227
Chapter 7: XPath Fundamentals	521
Chapter 8: XPath: Operators on Items	571
Chapter 9: XPath: Path Expressions	599
Chapter 10: XPath: Sequence Expressions	633
Chapter 11: XPath: Type Expressions	653
Chapter 12: XSLT Patterns	681
Chapter 13: The Function Library	709
Chapter 14: Regular Expressions	915
Chapter 15: Serialization	929
Part III: Exploitation	951
Chapter 16: Extensibility	953
Chapter 17: Stylesheet Design Patterns	973
Chapter 18: Case Study: XMLSpec	1001
Chapter 19: Case Study: A Family Tree	1049
Chapter 20: Case Study: Knight's Tour	1099
Part IV: Appendices	1115
Appendix A: XPath 2.0 Syntax Summary	1117
Appendix B: Error Codes	1123
Appendix C: Backward Compatibility	1139
Appendix D: Microsoft XSLT Processors	1147
Appendix E: JAXP: The Java API for Transformation	1163
Appendix F: Saxon	1195
Appendix G: Altova	1215
Appendix H: Glossary	1221
Index	1233

XSLT 2.0 and XPath 2.0

Programmer's Reference

4th Edition

Michael Kay



Wiley Publishing, Inc.

XSLT 2.0 and XPath 2.0 Programmer's Reference

4th Edition

Published by
Wiley Publishing, Inc.
10475 Crosspoint Boulevard
Indianapolis, IN 46256
www.wiley.com

Copyright © 2008 by Wiley Publishing, Inc., Indianapolis, Indiana

Published simultaneously in Canada

ISBN: 978-0-470-19274-0

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

Library of Congress Cataloging-in-Publication Data is available from the publisher.

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as permitted under Sections 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 646-8600. Requests to the Publisher for permission should be addressed to the Legal Department, Wiley Publishing, Inc., 10475 Crosspoint Blvd., Indianapolis, IN 46256, (317) 572-3447, fax (317) 572-4355, or online at <http://www.wiley.com/go/permissions>.

Limit of Liability/Disclaimer of Warranty: The publisher and the author make no representations or warranties with respect to the accuracy or completeness of the contents of this work and specifically disclaim all warranties, including without limitation warranties of fitness for a particular purpose. No warranty may be created or extended by sales or promotional materials. The advice and strategies contained herein may not be suitable for every situation. This work is sold with the understanding that the publisher is not engaged in rendering legal, accounting, or other professional services. If professional assistance is required, the services of a competent professional person should be sought. Neither the publisher nor the author shall be liable for damages arising herefrom. The fact that an organization or Website is referred to in this work as a citation and/or a potential source of further information does not mean that the author or the publisher endorses the information the organization or Website may provide or recommendations it may make. Further, readers should be aware that Internet Websites listed in this work may have changed or disappeared between when this work was written and when it is read.

For general information on our other products and services please contact our Customer Care Department within the United States at (800) 762-2974, outside the United States at (317) 572-3993 or fax (317) 572-4002.

Trademarks: Wiley, the Wiley logo, Wrox, the Wrox logo, Wrox Programmer to Programmer, and related trade dress are trademarks or registered trademarks of John Wiley & Sons, Inc. and/or its affiliates, in the United States and other countries, and may not be used without written permission. All other trademarks are the property of their respective owners. Wiley Publishing, Inc., is not associated with any product or vendor mentioned in this book.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic books.

*To Anyone Who Uses This Book
To Make the World a Better Place*

About the Author

Michael Kay has been working in the XML field since 1997; he became a member of the XSL Working Group soon after the publication of XSLT 1.0, and took over as editor of the XSLT 2.0 specification in early 2001. He is also a member of the XQuery and XML Schema Working Groups, and is a joint editor of the XPath 2.0 specification. He is well known not only through previous editions of this book but also as the developer of the open source Saxon product, a pioneering implementation of XSLT 2.0, XPath 2.0, and XQuery 1.0.

In 2004 the author formed his own company, Saxonica, to provide commercial software and services building on the success of the Saxon technology. Previously, he spent three years with Software AG, working with the developers of the Tamino XML server, an early XQuery implementation. His background is in database technology: after leaving the University of Cambridge with a Ph.D., he worked for many years with the (then) computer manufacturer ICL, developing network, relational, and object-oriented database software products as well as a text search engine, and held the position of ICL Fellow.

Michael lives in Reading, England, with his wife and daughter. His hobbies (reflected in his choice of examples) include genealogy and choral singing, and once included chess. Since completing the previous edition he has found time to improve his croquet handicap to 6.

Credits

Director of Acquisitions

Jim Minatel

Development Editor

Maureen Spears

Technical Editor

Sam Judson

Production Editor

Angela Smith

Copy Editor

Foxxe Editorial Services

Editorial Manager

Mary Beth Wakefield

Production Manager

Tim Tate

Vice President and Executive Group Publisher

Richard Swadley

Vice President and Executive Publisher

Joseph B. Wikert

Project Coordinator, Cover

Lynsey Stanford

Proofreader

Nancy Carrasco

Indexer

Robert Swanson

Acknowledgments

There are two groups of people I must thank: those who contributed to the book, and those who supported me in writing it.

In the first group, I am indebted to readers of previous editions who have pointed out my errors, and have told me what they liked and didn't like. I hope readers of this edition will do the same. Also to the (by now numerous) reviewers and editors engaged first by the original Wrox team in the UK, and more recently by their successors in Wiley, who have done so much of the legwork of testing example code and finding continuity errors, not to mention handling the unseen production processes that turn a heap of word-processed text into a finished book. Then my colleagues on the working groups, who provided the subject matter for me to write about, and those who taught me how to use the language — if you find a programming pearl that you particularly like in this book, the chances are I stole the idea from someone. James Clark in particular, who invented the XSLT language and showed me how it worked.

In the second group, I must once again acknowledge the patience of my family, who sighed resignedly when I suggested the prospect of retreating to my study for half a year to produce a new revision, and the generosity of my past employers who provided the time to get the project off the ground in the first place.

Contents

Introduction	xxix
List of Examples	xxxix

Part I: Foundations

Chapter 1: XSLT in Context	3
What Is XSLT?	3
Why Version 2.0?	5
A Scenario: Transforming Music	5
How Does XSLT Transform XML?	7
XSLT and SQL: An Analogy	8
XSLT Processors	9
A Simple XSLT Stylesheet	10
An XSLT 2.0 Stylesheet	19
The Place of XSLT in the XML Family	21
XSLT and XSL Formatting Objects	22
XSLT and XPath	22
XSLT and XML Namespaces	23
XSLT and CSS	24
XSLT and XML Schemas	24
XSLT and XQuery	26
The History of XSL	26
Prehistory	26
The First XSL Proposal	28
Saxon	30
Beyond XSLT 1.0	30
Convergence with XQuery	31
The Development of XSLT 2.0 and XPath 2.0	32
XSLT 2.0 as a Language	33
Use of XML Syntax	33
No Side Effects	34
Rule-Based	35
Types Based on XML Schema	38
A Two-Language System: XSLT and XPath	39
Summary	40

Chapter 2: The XSLT Processing Model **41**

XSLT: A System Overview	41
A Simplified Overview	41
Trees, Not Documents	42
Different Output Formats	43
Multiple Inputs and Outputs	44
The XDM Tree Model	45
XML as a Tree	45
Completing the UML Class Diagram	56
Document Order	57
Names and Namespaces	58
IDs and IDREFs	61
Characters in the Data Model	62
What Does the Tree Leave Out?	64
From Textual XML to a Data Model	65
Controlling Serialization	67
The Transformation Process	67
Invoking a Transformation	67
Template Rules	68
Push Processing	74
Controlling Which Nodes to Process	76
Modes	78
Built-In Template Rules	78
Conflict Resolution Policy	79
Error Handling	80
Variables and Expressions	80
Variables	81
Parameters	82
Expressions	82
Context	84
Temporary Documents	85
Summary	88

Chapter 3: Stylesheet Structure **89**

Changes in XSLT 2.0	90
The Modular Structure of a Stylesheet	90
The <xsl:stylesheet> Element	98
The <?xml-stylesheet?> Processing Instruction	99
Embedded Stylesheets	102

Declarations	104
XSLT-Defined Declarations	105
Implementor-Defined Declarations	106
User-Defined Top-Level Elements	107
Instructions	108
XSLT Instructions	108
Extension Instructions	111
Literal Result Elements	112
Attribute Value Templates	122
Simplified Stylesheets	125
Writing Portable Stylesheets	127
Conditional Compilation	127
Version Compatibility	128
Extensibility	134
Whitespace	141
The Effect of Stripping Whitespace Nodes	146
Whitespace Nodes in the Stylesheet	146
Solving Whitespace Problems	147
Summary	148
 Chapter 4: Stylesheets and Schemas	 151
XML Schema: An Overview	151
Simple Type Definitions	152
Elements with Attributes and Simple Content	154
Elements with Mixed Content	155
Elements with Element-Only Content	156
Defining a Type Hierarchy	157
Substitution Groups	158
Declaring Types in XSLT	161
Validating the Source Document	165
Validating the Result Document	170
Validating a Temporary Document	174
Validating Individual Elements	176
Validating Individual Attributes	179
The default-validation Attribute	180
Importing Schemas	180
Using xsi:type	181
Nullability	182
Summary	183

Chapter 5: Types	185
What Is a Type System?	185
Changes in 2.0	186
Sequences	187
Atomic Values	189
Atomic Types	191
The Major Atomic Types	193
The Minor Atomic Types	206
Derived Numeric Types	210
Derived String Types	213
Untyped Atomic Values	215
xs:NMTOKENS, xs:IDREFS, and xs:ENTITIES	217
Schema Types and XPath Types	217
The Type Matching Rules	219
Static and Dynamic Type Checking	221
Summary	224

Part II: XSLT and XPath Reference

Chapter 6: XSLT Elements	227
xsl:analyze-string	230
xsl:apply-imports	237
xsl:apply-templates	240
xsl:attribute	254
xsl:attribute-set	266
xsl:call-template	271
xsl:character-map	280
xsl:choose	282
xsl:comment	285
xsl:copy	287
xsl:copy-of	292
xsl:decimal-format	298
xsl:document	303
xsl:element	306
xsl:fallback	316
xsl:for-each	322
xsl:for-each-group	326
xsl:function	344
xsl:if	353

xsl:import	357
xsl:import-schema	368
xsl:include	372
xsl:key	376
xsl:matching-substring	386
xsl:message	386
xsl:namespace	390
xsl:namespace-alias	394
xsl:next-match	399
xsl:non-matching-substring	402
xsl:number	403
xsl:otherwise	420
xsl:output	420
xsl:output-character	424
xsl:param	425
xsl:perform-sort	437
xsl:preserve-space	439
xsl:processing-instruction	442
xsl:result-document	445
xsl:sequence	452
xsl:sort	455
xsl:strip-space	465
xsl:stylesheet	465
xsl:template	483
xsl:text	492
xsl:transform	495
xsl:value-of	495
xsl:variable	500
xsl:when	515
xsl:with-param	517
Summary	519
Chapter 7: XPath Fundamentals	521
Notation	522
Where to Start	523
Expressions	524
Examples	527
Lexical Constructs	527
Comments	529
Numeric Literals	530

Contents

String Literals	532
Names	534
Operators	537
Primary Expressions	539
Examples	540
Variable References	540
Usage	540
Examples	541
Parenthesized Expressions	542
Changes in XPath 2.0	543
Context Item Expressions	543
Changes in XPath 2.0	544
Usage	544
Function Calls	544
Identifying the Function to be Called	545
Converting the Arguments and the Result	547
Changes in XPath 2.0	549
Side Effects	549
Examples	550
Conditional Expressions	551
Changes in XPath 2.0	552
Examples	553
The XPath Evaluation Context	553
The Static Context	554
The Dynamic Context	563
Summary	568
 Chapter 8: XPath: Operators on Items	 571

Arithmetic Operators	571
Syntax	571
Type Promotion	572
Changes in XPath 2.0	573
Effect	573
Arithmetic Using Numbers	574
Examples of Numeric Arithmetic	576
Arithmetic Using Durations	577
Value Comparisons	581
Permitted Operand Types	582
Type Checking for Value Comparisons	586
Examples of Value Comparisons	587

General Comparisons	588
Changes in XPath 2.0	588
Rules for General Comparisons	589
Existential Comparison	590
Examples of General Comparisons	592
Node Comparisons	593
The «is» Operator	593
The operators «<<» and «>>»	594
Changes in XPath 2.0	594
Boolean Expressions	594
Shortcut Semantics	595
Examples	596
Summary	596
Chapter 9: XPath: Path Expressions	599
Examples of Path Expressions	600
Changes in XPath 2.0	601
Document Order and Duplicates	602
The Binary «/» Operator	602
Syntax	602
Effect	603
Examples of the Binary «/» Operator	604
Associativity of the «/» Operator	605
Axis Steps	606
Syntax of Axis Steps	606
Effect	607
Examples of Axis Steps	608
Axes	609
Node Tests	613
Name Tests	614
Kind Tests	616
Predicates	617
Abbreviated Axis Steps	621
Rooted Path Expressions	625
Syntax	625
Examples of Rooted Paths	626
The «//» Abbreviation	626
Examples Using «//»	627
Comparing «//» with «/descendant::»	628
Combining Sets of Nodes	628
Syntax	629

Contents

Examples	630
Usage	630
Set Intersection and Difference in XPath 1.0	631
Sets of Atomic Values	631
Summary	632
Chapter 10: XPath: Sequence Expressions	633
The Comma Operator	634
Examples	635
Numeric Ranges: The «to» Operator	636
Examples	637
Filter Expressions	638
Examples	640
The «for» Expression	640
Mapping a Sequence	641
Examples	642
The Context Item in a «for» Expression	642
Combining Multiple Sequences	643
Example	644
Simple Mapping Expressions	644
The «some» and «every» Expressions	646
Examples	648
Quantification and the «=» Operator	649
Errors in «some» and «every» Expressions	649
Summary	651
Chapter 11: XPath: Type Expressions	653
Converting Atomic Values	654
Converting between Primitive Types	656
Converting between Derived Types	664
Sequence Type Descriptors	668
Matching Atomic Values	669
Matching Nodes	670
Matching Elements and Attributes	672
The «instance of» Operator	677
The «treat as» Operator	678
Summary	680
Chapter 12: XSLT Patterns	681
Patterns and Expressions	681
Changes in XSLT 2.0	682

The Formal Definition	683
Applying the Definition in Practice	684
An Algorithm for Matching Patterns	685
Patterns Containing Predicates	685
An Informal Definition	685
Conflict Resolution	686
Matching Parentless Nodes	688
The Syntax of Patterns	689
Pattern	689
PathPattern	690
RelativePathPattern	693
PatternStep	694
IdKeyPattern	704
Summary	708
 Chapter 13: The Function Library	 709
A Word about Naming	710
Functions by Category	710
Notation	712
Code Samples	714
Function Definitions	714
abs	714
adjust-date-to-timezone, adjust-dateTime-to-timezone, adjust-time-to-timezone	715
avg	718
base-uri	719
boolean	721
ceiling	723
codepoint-equal	724
codepoints-to-string	725
collection	726
compare	727
concat	729
contains	730
count	733
current	734
current-date, current-dateTime, current-time	738
current-group	739
current-grouping-key	740
current-Time	741
data	741
dateTime	743
day-from-date, day-from-dateTime	744

Contents

days-from-duration	745
deep-equal	745
default-collation	748
distinct-values	749
doc, doc-available	750
document	754
document-uri	764
element-available	764
empty	770
encode-for-uri	771
ends-with	773
error	774
escape-html-uri	775
exactly-one	777
exists	778
false	779
floor	779
format-date, format-dateTime, format-time	781
format-number	788
format-time	792
function-available	792
generate-id	797
hours-from-dateTime, hours-from-time	800
hours-from-duration	801
id	802
idref	804
implicit-timezone	806
index-of	807
in-scope-prefixes	808
insert-before	810
iri-to-uri	811
key	812
lang	819
last	820
local-name	824
local-name-from-QName	826
lower-case	827
matches	828
max, min	830
min	832
minutes-from-dateTime, minutes-from-time	832
minutes-from-duration	832

minutes-from-time	833
month-from-date, month-from-dateTime	833
months-from-duration	834
name	835
namespace-uri	837
namespace-uri-for-prefix	839
namespace-uri-from-QName	841
nilled	842
node-name	843
normalize-space	845
normalize-unicode	847
not	850
number	851
one-or-more	853
position	854
prefix-from-QName	857
QName	858
regex-group	860
remove	861
replace	862
resolve-QName	864
resolve-uri	867
reverse	869
root	870
round	870
round-half-to-even	872
seconds-from-dateTime, seconds-from-time	873
seconds-from-duration	874
seconds-from-time	875
starts-with	875
static-base-uri	876
string	877
string-join	879
string-length	880
string-to-codepoints	881
subsequence	882
substring	883
substring-after	885
substring-before	887
sum	889
system-property	890
timezone-from-date, timezone-from-dateTime, timezone-from-time	893

Contents

tokenize	894
trace	896
translate	897
true	899
type-available	899
unordered	901
unparsed-entity-public-id, unparsed-entity-uri	902
unparsed-text, unparsed-text-available	904
upper-case	910
year-from-date, year-from-dateTime	911
years-from-duration	911
zero-or-one	912
Summary	913

Chapter 14: Regular Expressions **915**

Branches and Pieces	916
Quantifiers	916
Atoms	917
Subexpressions	918
Back-References	918
Character Groups	919
Character Ranges	919
Character Class Escapes	920
Character Blocks	922
Character Categories	924
Flags	925
The «i» flag	925
The «m» flag	926
The «S» flag	926
The «X» flag	926
Disallowed Constructs	927
Summary	927

Chapter 15: Serialization **929**

The XML Output Method	929
The HTML Output Method	936
The XHTML Output Method	939
The Text Output Method	940

Using the <xsl:output> declaration	940
Character Maps	941
Usage	942
Choosing Characters to Map	942
Limitations of Character Maps	944
Disable Output Escaping	945
Reasons to Disable Output Escaping	945
Why disable-output-escaping Is Deprecated	946
Using disable-output-escaping to Wrap HTML in CDATA	947
Character Maps as a Substitute for disable-output-escaping	948
Summary	949

Part III: Exploitation

Chapter 16: Extensibility **953**

What Vendor Extensions Are Allowed?	954
Extension Functions	955
When Are Extension Functions Needed?	955
When Are Extension Functions Not Needed?	956
Calling Extension Functions	956
What Language Is Best?	957
Client-Side Script	957
Binding Extension Functions	957
XPath Trees and the DOM	963
Calling External Functions within a Loop	965
Functions with Uncontrolled Side Effects	967
Keeping Extensions Portable	970
Summary	971

Chapter 17: Stylesheet Design Patterns **973**

Fill-in-the-Blanks Stylesheets	973
Navigational Stylesheets	976
Rule-Based Stylesheets	980
Computational Stylesheets	985
Programming without Assignment Statements	985
So Why Are They Called Variables?	989
Avoiding Assignment Statements	989
Summary	1000

Chapter 18: Case Study: XMLSpec	1001
Formatting the XML Specification	1002
Preface	1004
Creating the HTML Outline	1008
Formatting the Document Header	1012
Creating the Table of Contents	1019
Creating Section Headers	1023
Formatting the Text	1024
Producing Lists	1028
Making Cross-References	1029
Setting Out the Production Rules	1033
Overlay Stylesheets	1041
diffspec.xsl	1041
REC-xml.xsl	1044
Stylesheets for Other Specifications	1044
xslt.xsl	1045
xsltdiff.xsl	1046
funcproto.xsl	1046
xsl-query.xsl	1047
xmlspec.xsl	1047
Summary	1047
Chapter 19: Case Study: A Family Tree	1049
Modeling a Family Tree	1050
The GEDCOM Data Model	1050
Creating a Schema for GEDCOM 6.0	1053
The GEDCOM 6.0 Schema	1054
Creating a Data File	1058
Converting GEDCOM Files to XML	1059
Converting from GEDCOM 5.5 to 6.0	1063
Displaying the Family Tree Data	1072
The Stylesheet	1073
Putting It Together	1085
Summary	1098
Chapter 20: Case Study: Knight's Tour	1099
The Problem	1099
The Algorithm	1100
Placing the Knight	1104
Displaying the Final Board	1105

Finding the Route	1106
Finding the Possible Moves	1107
Trying the Possible Moves	1109
Selecting the Best Move	1111
Running the Stylesheet	1112
Observations	1112
Summary	1113

Part IV: Appendices

Appendix A: XPath 2.0 Syntax Summary

1117

Whitespace and Comments	1118
Tokens	1118
Syntax Productions	1119
Operator Precedence	1122

Appendix B: Error Codes

1123

Functions and Operators (FO)	1124
XPath Errors (XP)	1126
XSLT Errors (XT)	1127

Appendix C: Backward Compatibility

1139

Stage 1: Backward-compatibility Mode	1140
Deprecated Facilities	1140
Error Handling	1140
Comparing Strings	1141
Numeric Formats	1141
Other XPath Changes	1142
Serialization Changes	1142
Stage 2: Setting version="2.0"	1142
The First Node Rule	1142
Type Checking of Function Arguments	1143
Comparison Operators	1143
Arithmetic	1143
The Empty Sequence	1143
Error Semantics for «and» and «or»	1144
Other XSLT Differences	1144
Stage 3: Adding a Schema	1145
Summary	1145

Appendix D: Microsoft XSLT Processors **1147**

MSXML	1147
Objects	1148
IXMLDOMDocument and IXMLDOMDocument2	1148
IXMLDOMNode	1150
IXMLDOMNodeList	1151
IXMLDOMParseError	1151
IXMLDOMSelection	1152
IXSLProcessor	1152
IXSLTemplate	1153
Putting it Together	1154
Restrictions	1158
System.Xml	1158
XPathDocument	1158
XmlNode	1159
XPathNavigable	1159
XPathNavigator	1159
XslTransform	1159
Summary	1161

Appendix E: JAXP: The Java API for Transformation **1163**

The JAXP Parser API	1164
JAXP Support for SAX	1164
JAXP Support for DOM	1166
The JAXP Transformation API	1169
Examples of JAXP Transformations	1187
Example 1: Transformation Using Files	1187
Example 2: Supplying Parameters and Output Properties	1188
Example 3: Holding Documents in Memory	1188
Example 4: Using the <?xml-stylesheet?> Processing Instruction	1189
Example 5: A SAX Pipeline	1190
Summary	1193

Appendix F: Saxon **1195**

Using Saxon from the Command Line	1196
Using Saxon from a Java Application	1199
Using Saxon via JAXP Interfaces	1199
The s9api Interface	1202
Using Saxon from a .NET Application	1203
Saxon Tree Models	1205

Extensibility	1205
Writing Extension Functions in Java	1206
Writing Extension Functions under .NET	1206
Collations	1207
Extensions	1208
Serialization Extensions	1208
Extension Attributes	1209
Extension Instructions	1209
Extension Functions	1209
The evaluate() Extension	1210
Summary	1214
Appendix G: Altova	1215
Running from within XMLSpy	1215
Conformance	1216
Extensions and Extensibility	1217
The Command Line Interface	1217
Using the API	1218
The COM API	1218
The Java API	1219
The .NET API	1220
Summary	1220
Appendix H: Glossary	1221
Index	1233

Introduction

This book, as the title implies, is primarily a practical reference book for professional XSLT developers. It assumes no previous knowledge of the language, and many developers have used it as their first introduction to XSLT; however, it is not structured as a tutorial, and there are other books on XSLT that provide a gentler approach for beginners.

Who This Book Is For

The book does assume a basic knowledge of XML, HTML, and the architecture of the Web, and it is written for experienced programmers. There's no assumption that you know any particular language such as Java or Visual Basic, just that you recognize the concepts that all programming languages have in common.

I have tried to make the book suitable both for XSLT 1.0 users upgrading to XSLT 2.0, and for newcomers to XSLT. This is easier to do in a reference book, of course, than a tutorial. I have also tried to make the book equally suitable whether you work in the Java or .NET world.

As befits a reference book, a key aim is that the coverage should be comprehensive and authoritative. It is designed to give you all the details, not just an overview of the 20 percent of the language that most people use 80 percent of the time. It's designed so that you will keep coming back to the book whenever you encounter new and challenging programming tasks, not as a book that you skim quickly and then leave on the shelf. If you like detail, you will enjoy this book; if not, you probably won't.

But as well as giving the detail, this book aims to explain the concepts, in some depth. It's therefore a book for people who not only want to use the language but who also want to understand it at a deep level. Many readers have written to me saying that they particularly appreciate these insights into the language, and it's my sincere hope that after reading it, you will not only be a more productive XSLT programmer, but also a more knowledgeable software engineer.

What This Book Covers

The book aims to tell you everything you need to know about the XSLT 2.0 language. It gives equal weight to the things that are new in XSLT 2.0 and the things that were already present in version 1.0. The book is about the language, not about specific products. However, there are appendices about Saxon (my own implementation of XSLT 2.0), about the Altova XSLT 2.0 implementation, and about the Java and Microsoft APIs for controlling XSLT transformations, which will no doubt be upgraded to handle XSLT 2.0 as well as 1.0. A third XSLT 2.0 processor, Gestalt, was released shortly before we went to press, too late for us describe it in any detail. But the experience of XSLT 1.0 is that there has been a very high level of interoperability between different XSLT processors, and if you can use one of them, then you can use them all.

In the previous edition we split XSLT 2.0 and XPath 2.0 into separate volumes. The idea was that some readers might be interested in XPath alone. However, many bought the XSLT 2.0 book without its XPath

Introduction

companion and were left confused as a result; so this time, we've brought the material back together. The XPath reference information is in self-contained chapters, so it should still be accessible when you use XPath in contexts other than XSLT.

The book does not cover XSL Formatting Objects, a big subject in its own right. Nor does it cover XML Schemas in any detail. If you want to use these important technologies in conjunction with XSLT, there are other books that do them justice.

How This Book Is Structured

This book contains twenty chapters and eight appendixes (the last of which is a glossary) organized into four parts. The following section outlines what you can find in each part, chapter, and appendix.

Part I: Foundations

The first part of the book covers essential concepts. I recommend reading these before you start coding. If you ignore this advice, as most people do, then I suggest you read them when you get to that trough of despair when you find it impossible to make the language do anything but the most trivial tasks. XSLT is different from other languages, and to make it work for you, you need to understand how it was designed to be used.

Chapter 1: XSLT in Context

This chapter explains how XSLT fits into the big picture: how the language came into being and how it sits alongside other technologies. It also has a few simple coding examples to keep you alert.

Chapter 2: The XSLT Processing Model

This is about the architecture of an XSLT processor: the inputs, the outputs, and the data model. Understanding the data model is perhaps the most important thing that distinguishes an XSLT expert from an amateur; it may seem like information that you can't use immediately, but it's knowledge that will stop you from making a lot of stupid mistakes.

Chapter 3: Stylesheet Structure

XSLT development is about writing stylesheets, and this chapter takes a bird's-eye view of what stylesheets look like. It explains the key concepts of rule-based programming using templates, and explains how to undertake programming-in-the-large by structuring your application using modules and pipelines.

Chapter 4: Stylesheets and Schemas

A key innovation in XSLT 2.0 is that stylesheets can take advantage of knowledge about the structure of your input and output documents, provided in the form of an XML Schema. This chapter provides a quick overview of XML Schema to describe its impact on XSLT development. Not everyone uses schemas, and you can skip this chapter if you fall into that category.

Chapter 5: The Type System

XPath 2.0 and XSLT 2.0 offer strong typing as an alternative to the weak typing approach of the 1.0 languages. This means that you can declare the types of your variables, functions, and parameters, and use this information to get early warning of programming errors. This chapter explains the data types available and the mechanisms for creating user-defined types.

Part II: XSLT and XPath Reference

This section of the book contains reference material, organized in the hope that you can easily find what you need when you need it. It's not designed for sequential reading, though if you're like me, you might well want to leaf through the pages to discover what's there.

Chapter 6: XSLT Elements

This monster chapter lists all the XSLT elements you can use in a stylesheet, in alphabetical order, giving detailed rules for the syntax and semantics of each element, advice on usage, and examples. This is probably the part of the book you will use most frequently as you become an expert XSLT user. It's a "no stone unturned" approach, based on the belief that as a professional developer you need to know what happens when the going gets tough, not just when the wind is in your direction.

Chapter 7: XPath Fundamentals

This chapter explains the basics of XPath: the low-level constructs such as literals, variables, and function calls. It also explains the context rules, which describe how the evaluation of XPath expressions depends on the XSLT processing context in which they appear.

Chapter 8: XPath: Operators on Items

XPath offers the usual range of operators for performing arithmetic, boolean comparison, and the like. However, these don't always behave exactly as you would expect, so it's worth reading this chapter to see what's available and how it differs from the last language that you used.

Chapter 9: XPath: Path Expressions

Path expressions are what make XPath special; they enable you to navigate around the structure of an XML document. This chapter explains the syntax of path expressions, the 13 axes that you can use to locate the nodes that you need, and associated operators such as union, intersection, and difference.

Chapter 10: XPath: Sequence Expressions

Unlike XPath 1.0, in version 2.0 all values are sequences (singletons are just a special case). Some of the most important operators in XPath 2.0 are those that manipulate sequences, notably the «for» expression, which translates one sequence into another by applying a mapping.

Chapter 11: XPath: Type Expressions

The type system was explained in Chapter 5; this chapter explains the operations that you can use to take advantage of types. This includes the «cast» operation which is used to convert values from one type to another. A big part of this chapter is devoted to the detailed rules for how these conversions are done.

Chapter 12: XSLT Patterns

This chapter returns from XPath to a subject that's specific to XSLT. Patterns are used to define template rules, the essence of XSLT's rule-based programming approach. The reason for explaining them now is that the syntax and semantics of patterns depends strongly on the corresponding rules for XPath expressions.

Chapter 13: The Function Library

XPath 2.0 includes a library of functions that can be called from any XPath expression; XSLT 2.0 extends this with some additional functions that are available only when XPath is used within XSLT. The library

Introduction

has grown immensely since XPath 1.0. This chapter provides a single alphabetical reference for all these functions.

Chapter 14: Regular Expressions

Processing of text is an area where XSLT 2.0 and XPath 2.0 are much more powerful than version 1.0, and this is largely through the use of constructs that exploit regular expressions. If you're familiar with regexes from languages such as Perl, this chapter tells you how XPath regular expressions differ. If you're new to the subject, it explains it from first principles.

Chapter 15: Serialization

Serialization in XSLT means the ability to generate a textual XML document from the tree structure that's manipulated by a stylesheet. This isn't part of XSLT processing proper, so (following W3C's lead) we've separated it into its own chapter. You can control serialization from the stylesheet using an `<xsl:output>` declaration, but many products also allow you to control it directly via an API.

Part III: Exploitation

The final section of the book is advice and guidance on how to take advantage of XSLT to write real applications. It's intended to make you not just a competent XSLT coder, but a competent designer too. The best way of learning is by studying the work of others, so the emphasis here is on practical case studies.

Chapter 16: Extensibility

This chapter describes the "hooks" provided in the XSLT specification to allow vendors and users to plug in extra functionality. The way this works will vary from one implementation to another, so we can't cover all possibilities, but one important aspect that the chapter does cover is how to use such extensions and still keep your code portable.

Chapter 17: Stylesheet Design Patterns

This chapter explores a number of design and coding patterns for XSLT programming, starting with the simplest "fill-in-the-blanks" stylesheet, and extending to the full use of recursive programming in the functional programming style, which is needed to tackle problems of any computational complexity. This provides an opportunity to explain the thinking behind functional programming and the change in mindset needed to take full advantage of this style of development.

Chapter 18: Case Study: XMLSpec

XSLT is often used for rendering documents, so where better to look for a case study than the stylesheets used by the W3C to render the XML and XSLT specifications, and others in the same family, for display on the Web? The resulting stylesheets are typical of those you will find in any publishing organization that uses XML to develop a series of documents with a compatible look-and-feel.

Chapter 19: Case Study: A Family Tree

Displaying a family tree is another typical XSLT application. This time we're starting with semi-structured data — a mixture of fairly complex data and narrative text — that can be presented in many different ways for different audiences. We also show how to tackle another typical XSLT problem, conversion of the data into XML from a legacy text-based format. As it happens, this uses nearly all the important

new XSLT 2.0 features in one short stylesheet. But another aim of this chapter is to show a collection of stylesheets doing different jobs as part of a complete application.

Chapter 20: Case Study: Knight's Tour

Finding a route around a chessboard where a knight visits every square without ever retracing its steps might sound a fairly esoteric application for XSLT, but it's a good way of showing how even the most complex of algorithms are within the capabilities of the language. You may not need to tackle this particular problem, but if you want to construct an SVG diagram showing progress against your project plan, then the problems won't be that dissimilar.

Part IV: Appendices

Appendix A: XPath 2.0 Syntax Summary

Collects the XPath grammar rules and operator precedences into one place for ease of reference.

Appendix B: Error Codes

A list of all the error codes defined in the XSLT and XPath language specifications, with brief explanations to help you understand what's gone wrong.

Appendix C: Backward Compatibility

The list of things you need to look out for when converting applications from XSLT 1.0.

Appendix D: Microsoft XSLT Processors

Although the two Microsoft XSLT processors don't yet support XSLT 2.0, we thought many readers would find it useful to have a quick summary here of the main objects and methods used in their APIs.

Appendix E: JAXP: The Java API for XML Processing

JAXP is an interface rather than a product. Again, it doesn't have explicit support yet for XSLT 2.0, but Java programmers will often be using it in XSLT 2.0 projects, so we decided to include an overview of the classes and methods available.

Appendix F: Saxon

At the time of writing Saxon (developed by the author of this book) provides the most comprehensive implementation of XSLT 2.0 and XPath 2.0, so we decided to cover its interfaces and extensions in some detail.

Appendix G: Altova

Altova, the developers of XML Spy, have an XSLT 2.0 processor that can be used either as part of the development environment or as a freestanding component. This appendix gives details of its interfaces.

Appendix H: Glossary

Index

What You Need to Use This Book

To use XSLT 2.0, you'll need an XSLT 2.0 processor: at the time of writing that means Saxon, AltovaXML, or Gestalt, though Gestalt appeared on the scene too late for us to give it much coverage. You can run these products in a number of different ways, which are described as part of the "Hello World!" example in Chapter 1 (pages 11–18).

If in doubt, the simplest way to get started is probably to download Kernow (<http://kernowforsaxon.sourceforge.net/>), which has Java SE 6 as a prerequisite. Kernow comes complete with the Saxon XSLT engine. The only other thing you will need is a text editor.

Conventions

To help you get the most from the text and keep track of what's happening, we've used a number of conventions throughout the book.

There are two kinds of code examples in this book: *code fragments* and *worked examples*.

Code fragments are incomplete and are not intended to be executed on their own. You can build them into your own stylesheets if you find them useful, but you will have to retype the code.

Worked examples are provided in the form of complete stylesheets, accompanied by sample source XML documents to which they can be applied, and an illustration of the output that they are expected to produce. You can download these examples and try them out for yourself. They generally appear in a box like this:

A Specimen Example

Source

This section gives the XML source data, the input to the transformation. If the filename is given as `example.xml`, you will find that file in the archive that you can download from the Wrox website at <http://www.wrox.com/>, generally in a subdirectory holding all the examples for one chapter.

```
<source data="xml"/>
```

Stylesheet

This section describes the XSLT stylesheet used to achieve the transformation. Again, there will usually be a filename such as `style.xsl`, so you can find the stylesheet in the Wrox download archive.

```
<xsl:stylesheet...
```

Output

This section shows the output when you apply this stylesheet to this source data, either as an XML or HTML listing, or as a screenshot.

```
<html...</html>
```

Occasionally, for reasons of space, we haven't printed the whole of the source document or the stylesheet in the book, but instead refer you to the website to fetch it.

Boxes like this one hold important, not-to-be forgotten information that is directly relevant to the surrounding text.

Notes, tips, hints, tricks, and asides to the current discussion are offset and placed in italics like this.

As for styles in the text:

- ❑ We *highlight* new terms and important words when we introduce them.
- ❑ We show keyboard strokes like this: Ctrl+A.
- ❑ We show filenames, URLs, and code within the text like so: `persistence.properties`.
- ❑ We show code within the text as follows: Element names are written as `<html>` or `<xsl:stylesheet>`. Function names are written as `concat()` or `current-date()`. Other names (for example of attributes or types) are written simply as `version` or `xs:string`. Fragments of code other than simple names are offset from the surrounding text by chevrons; for example, `«substring($a, 1, 1)='X'»`. Chevrons are also used around individual characters or string values, or when referring to keywords such as `«for»` and `«at»` that need to stand out from the text. As a general rule, if a string is enclosed in quotation marks, then the quotes are part of the code example, whereas if it is enclosed in chevrons, the chevrons are there only to separate the code from the surrounding text.
- ❑ We present code in two different ways:

For blocks of code we usually use gray highlighting.

But for individual lines of code we sometimes omit the highlighting.

Downloading the Code

All of the source code referred to in this book is available for download at <http://www.wrox.com>. Once at the site, simply locate the book's title (either by using the Search box or by using one of the title lists) and click the Download Code link on the book's detail page to obtain all the source code for the book.

Introduction

Because many books have similar titles, you may find it easiest to search by ISBN; this book's ISBN is 978-0-470-19274-0.

Once you download the code, just decompress it with your favorite compression tool. Alternately, you can go to the main Wrox code download page at <http://www.wrox.com/dynamic/books/download.aspx> to see the code available for this book and all other Wrox books.

Errata

We make every effort to ensure that there are no errors in the text or in the code. However, no one is perfect, and mistakes do occur. If you find an error in one of our books, such as a spelling mistake or faulty piece of code, we would be very grateful for your feedback. By sending in errata you may save another reader hours of frustration and at the same time you will be helping us provide even higher-quality information.

To find the errata page for this book, go to <http://www.wrox.com> and locate the title using the Search box or one of the title lists. Then, on the book details page, click the Book Errata link. On this page you can view all errata that have been submitted for this book and posted by Wrox editors. A complete book list, including links to each book's errata, is also available at www.wrox.com/misc-pages/booklist.shtml.

If you don't spot "your" error on the Book Errata page, go to www.wrox.com/contact/techsupport.shtml and complete the form there to send us the error you have found. We'll check the information and, if appropriate, post a message to the book's errata page and fix the problem in subsequent editions of the book.

p2p.wrox.com

For author and peer discussion, join the P2P forums at p2p.wrox.com. The forums are a Web-based system for you to post messages relating to Wrox books and related technologies and interact with other readers and technology users. The forums offer a subscription feature to e-mail you topics of interest of your choosing when new posts are made to the forums. Wrox authors, editors, other industry experts, and your fellow readers are present on these forums.

At <http://p2p.wrox.com> you will find a number of different forums that will help you not only as you read this book but also as you develop your own applications. To join the forums, just follow these steps:

1. Go to p2p.wrox.com and click the Register link.
2. Read the terms of use and click Agree.
3. Complete the required information to join as well as any optional information you wish to provide, and click Submit.
4. You will receive an e-mail with information describing how to verify your account and complete the joining process.

You can read messages in the forums without joining P2P but in order to post your own messages, you must join.

Once you join, you can post new messages and respond to messages other users post. You can read messages at any time on the Web. If you would like to have new messages from a particular forum e-mailed to you, click the Subscribe to this Forum icon by the forum name in the forum listing.

Here are some tips for writing a question if you want a good answer:

1. Choose your subject line carefully. Not just “XSLT question”.
2. Don’t use text shorthand. Not everyone has English as their first language, but if you take care over writing your question, it’s much more likely that someone will take care over answering it.
3. Show a complete source document, a complete example of your required output, and if you want to know why your code doesn’t work, your complete code — but only after paring the problem down to its essentials. Don’t ask people to debug code that they can’t see.
4. If you tried something and it didn’t work, say exactly what you tried and exactly how it failed (including details of what products you are using).

For more information about how to use the Wrox P2P, be sure to read the P2P FAQs for answers to questions about how the forum software works as well as many common questions specific to P2P and Wrox books. To read the FAQs, click the FAQ link on any P2P page.

