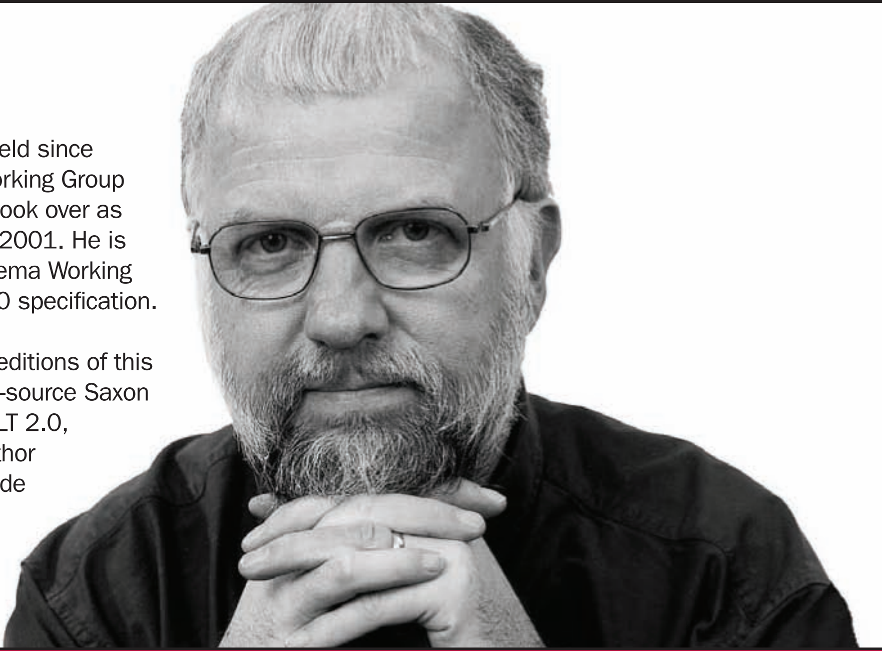
Wrox Programmer to Programmer™



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

wrox™

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

XSLT 2.0 and XPath 2.0
Programmer's Reference
4th Edition

XSLT 2.0 and XPath 2.0

Programmer's Reference

4th Edition

Michael Kay

WILEY

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

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.

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.

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

Part I: Foundations

Part II: XSLT and XPath Reference

Part III: Exploitation

Part IV: Appendices

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 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 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.

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.