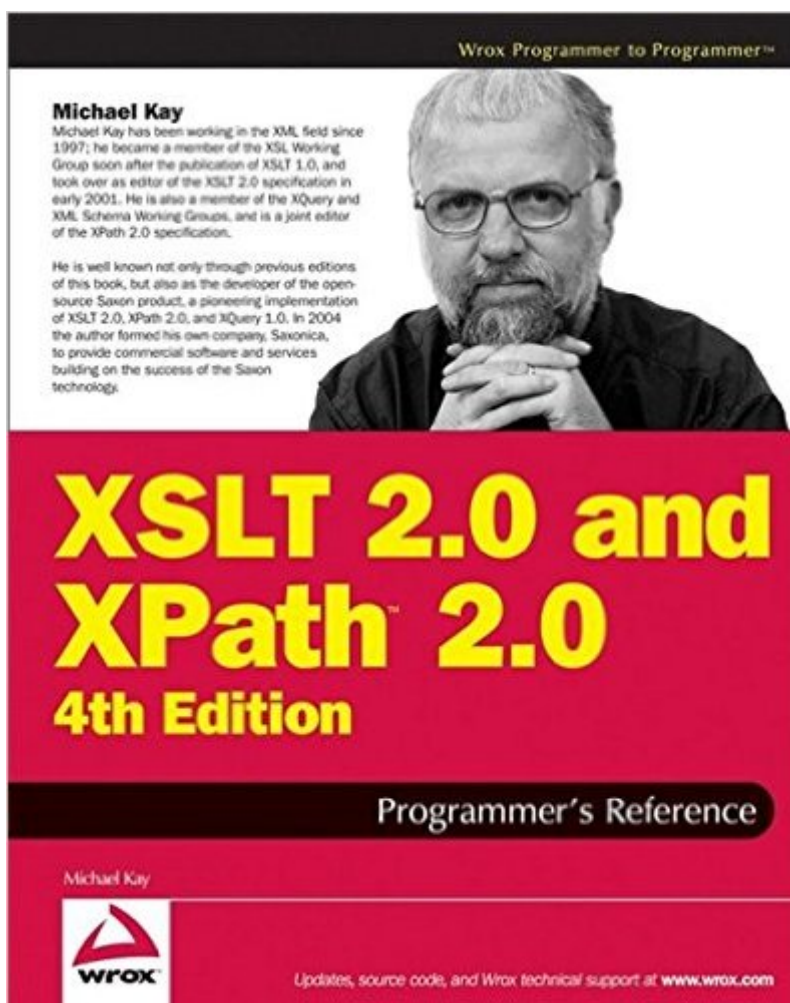


The book was found

XSLT 2.0 And XPath 2.0 Programmer's Reference



Synopsis

This book 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. 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. The book is suitable both for XSLT 1.0 users upgrading to XSLT 2.0, and for newcomers to XSLT. The book is also 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. 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 (the author's 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 the book went to press, too late to 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, the material is 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. 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. You should read these before you start coding. If you ignore this advice, as most people do, then 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 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 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 *œno 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) it's separated into its own chapter. You can control serialization from the stylesheet using an 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 example with semi-structured data—a mixture of fairly complex data and narrative text—that can be presented in many different ways for different audiences. It also shows 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 the book includes 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 its interfaces and extensions are covered 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 Note: CD-ROM/DVD and other supplementary materials are not included as part of eBook file.

Book Information

Hardcover: 1376 pages

Publisher: Wrox; 4 edition (May 5, 2008)

Language: English

ISBN-10: 0470192747

ISBN-13: 978-0470192740

Product Dimensions: 7.6 x 2 x 9.5 inches

Shipping Weight: 4.4 pounds (View shipping rates and policies)

Average Customer Review: 4.4 out of 5 stars See all reviews (27 customer reviews)

Best Sellers Rank: #337,186 in Books (See Top 100 in Books) #50 in Books > Computers & Technology > Programming > Languages & Tools > XML #675 in Books > Computers & Technology > Web Development & Design > Web Design #944 in Books > Computers & Technology > Programming > Web Programming

Customer Reviews

You won't easily find a better book on XSLT 2.0 and XPath 2.0. The reason is very simple: Michael Kay is THE expert on the subject, he has been publishing xslt references for almost a decade now. Currently he is the editor of the XSLT 2.0 specification at W3C. He is also involved the XQuery and the XML Schema Working Groups, also at W3C. He has developed the Saxon XSLT processor. This book (like its predecessors were and its future versions will be) is the definitive reference on

XSLT. The third edition was in two separate books, one on XSLT 2.0 (XSLT 2.0 Programmer's Reference (Programmer to Programmer)) and one on XPath 2.0 (XPath 2.0 Programmer's Reference (Programmer to Programmer)). If you are serious about XSLT 2.0, you also need the information on XPath 2.0, as it is a sub language of XSLT 2.0. So you would need both books at hand. The current book contains all of the material available in the two predecessors, and more. I could have survived on the two books, but tired of taking them from my office to my home and vice versa, I ordered the new edition and I am enjoying it very much. After using the new edition for a week or so I have come upon quite a few improvements, for instance the chapter on regular expressions contains more information and is better structured. Examples have been updated and as have been the appendices covering the processors. There is a new appendix on the Altova processor.

It's tricky to judge the quality of a book in a field with which you're unfamiliar. One technique I've used is using common sense to find various kinds of errors-- and a good sign of this book is that each time I thought I'd caught the author in a mistake, further study always proved me wrong. I eventually ended up reading this book practically cover to cover and soaking up as much as I could. This book has no fluff or cutesy humor. The writing is precise, to the point, and anticipated my questions. There is some repetition (for example, on the topic of namespaces), making the book in my mind very useful as a reference. The author's obvious familiarity with the standard itself and its various implementations reveals itself in the many clarifying annotations. Some of the examples are very elaborate and thorough and serve to clarify the complexities of the technology where needed-- for example, regarding the subtleties of the 'for' and 'number' family of instructions. If there are any deficiencies in the book at all, they are minor. One irritation is the author's prejudice towards Java (granted that XSLT and XPath themselves seem more comfortable in that domain); for example, calling C++ "procedural" or claiming that Java popularized IEEE 754 (neglecting phenomenal implementations such as SANE that predated Java by a decade). This didn't really distract from the value of the book, other than that an entire appendix is devoted to JAXP while there is not a single mention of libxslt2-- making me wonder for a while whether the technologies described in this book were applicable to the problem I was trying to solve (they were).

[Download to continue reading...](#)

XSLT 2.0 and XPath 2.0 Programmer's Reference XSLT 2.0 Programmer's Reference (Programmer to Programmer) XML: A Beginner's Guide: Go Beyond the Basics with Ajax, XHTML, XPath 2.0, XSLT 2.0 and XQuery Definitive XSLT and XPath XSL: The Personal Trainer for XSLT, XPath and

XSL-FO Professional JSP: Using JavaServer Pages, Servlets, EJB, JNDI, JDBC, XML, XSLT, and WML to Create Dynamic and Customizable Web Content (Programmer to Programmer) XSLT Cookbook: Solutions and Examples for XML and XSLT Developers, 2nd Edition Microsoft Win32 Programmer's Reference: Introduction Platforms, and Index (Microsoft Windows Programmer's Reference Library) Microsoft Win32 Programmer's Reference Library: Multimedia (Microsoft Windows Programmer's Reference Library) Java Programmer's Reference: Programmer's Reference XSLT 1.0 Pocket Reference (Pocket Reference (O'Reilly)) Microsoft Win32 Programmer's Reference: Window Management and Graphics Device Interface (Microsoft Professional Reference) Microsoft Win32 Programmer's Reference: System Services, Multimedia, Extensions, and Application Notes (Microsoft Professional Reference) Microsoft Win32 Programmer's Reference: Messages, Structures, and Macros (Microsoft Professional Reference) Microsoft Win32 Programmer's Reference: Functions A-G (Microsoft Professional Reference) Microsoft Win32 Programmer's Reference: Functions H-Z (Microsoft Professional Reference) Microsoft Visual C++ Run Time Library Reference, Part 3 (Microsoft Visual C++ 5.0 Programmer's Reference Set) Beginning ASP.NET 4.5.1: in C# and VB (Wrox Programmer to Programmer) Professional Jini (Programmer to Programmer) Professional ASP.NET 2.0 AJAX (Programmer to Programmer)

[Dmca](#)