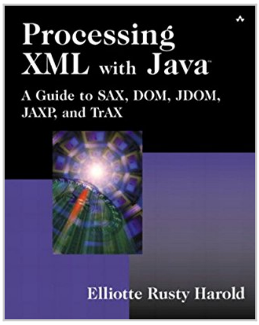

Processing XML with Java¿: A Guide to SAX, DOM, JDOM, JAXP, and TrAX (2 Volume Set)

By Elliotte Rusty Harold

Processing XML with Java¿: A Guide to SAX, DOM, JDOM, JAXP, and TrAX (2 Volume Set) By Elliotte Rusty Harold

-- Author is uniquely qualified to write this book! He is the author of numerous well-received books on Java and has written two of the best-selling books on XML. This is the book that brings his skills together.

-- Harold has a fantastic reputation, is a skilled writer, and has excellent publicity channels.

-- A complete guide to writing Java programs that read and write XML documents.

Java is the ideal language for processing XML documents. Consequently, more XML tools have been written in Java than in any other language. More open source XML tools are written in Java than in any other language. Processing XML with Java fills an immediate need for developers who are working with XML in Java. It is a comprehensive tutorial and reference to the major APIs. This book shows developers how to: save XML documents from their applications written in Java; read XML documents produced by other programs; communicate with network servers that send and receive XML data; validate documents they receive against DTDs, schemas, and business rules; and integrate XSLT into their programs.

Processing XML with Java¿: A Guide to SAX, DOM, JDOM, JAXP, and TrAX (2 Volume Set)

By Elliotte Rusty Harold

Processing XML with Java¿: A Guide to SAX, DOM, JDOM, JAXP, and TrAX (2 Volume Set) By Elliotte Rusty Harold

-- Author is uniquely qualified to write this book! He is the author of numerous well-received books on Java and has written two of the best-selling books on XML. This is the book that brings his skills together.

-- Harold has a fantastic reputation, is a skilled writer, and has excellent publicity channels.

-- A complete guide to writing Java programs that read and write XML documents.

Java is the ideal language for processing XML documents. Consequently, more XML tools have been written in Java than in any other language. More open source XML tools are written in Java than in any other language. Processing XML with Java fills an immediate need for developers who are working with XML in Java. It is a comprehensive tutorial and reference to the major APIs. This book shows developers how to: save XML documents from their applications written in Java; read XML documents produced by other programs; communicate with network servers that send and receive XML data; validate documents they receive against DTDs, schemas, and business rules; and integrate XSLT into their programs.

Processing XML with Java¿: A Guide to SAX, DOM, JDOM, JAXP, and TrAX (2 Volume Set) By Elliotte Rusty Harold Bibliography

- Sales Rank: #1816513 in Books
- Published on: 2002-11-15
- Original language: English
- Number of items: 1
- Dimensions: 9.10" h x 2.30" w x 7.40" l, 3.63 pounds
- Binding: Paperback
- 1120 pages

Download and Read Free Online Processing XML with Java¿: A Guide to SAX, DOM, JDOM, JAXP, and TrAX (2 Volume Set) By Elliotte Rusty Harold

Editorial Review

From the Back Cover

Praise for Elliotte Rusty Harold's *Processing XML with Java™*

"The sophistication and language are very appropriate for Java and XML application developers. You can tell by the way the author writes that he too is a developer. He delves very deeply into the topics and has really taken things apart and investigated how they work. I especially like his coverage of 'gotchas,' pitfalls, and limitations of the technologies."

—**John Wegis**, Web Engineer, Sun Microsystems, Inc.

"Elliotte has written an excellent book on XML that covers a lot of ground and introduces current and emerging technologies. He helps the novice programmer understand the concepts and principles of XML and related technologies, while covering the material at a level that's deep enough for the advanced developer. With a broad coverage of XML technologies, lots of little hints, and information I haven't seen in any other book on the topic, this work has become a valuable addition to my technical library."

—**Robert W. Husted**, Member, Technical Staff, Requisite Technology, Inc.

"The code examples are well structured and easy to follow. They provide real value for someone writing industrial-strength Java and XML applications. The time saved will repay the cost of this book a hundred times over.

"The book also contains more of the pearls of wisdom we've come to expect from Elliotte Rusty Harold—the kind of pointers that will save developers weeks, if not months, of time."

—**Ron Weber**, Independent Software Consultant

Written for Java programmers who want to integrate XML into their systems, this practical, comprehensive guide and reference shows how to process XML documents with the Java programming language. It leads experienced Java developers beyond the basics of XML, allowing them to design sophisticated XML applications and parse complicated documents.

Processing XML with Java™ provides a brief review of XML fundamentals, including XML syntax; DTDs, schemas, and validity; stylesheets; and the XML protocols XML-RPC, SOAP, and RSS. The core of the book comprises in-depth discussions on the key XML APIs Java programmers must use to create and manipulate XML files with Java. These include the Simple API for XML (SAX), the Document Object Model (DOM), and JDOM (a Java native API). In addition, the book covers many useful supplements to these core APIs, including XPath, XSLT, TrAX, and JAXP.

Practical in focus, ***Processing XML with Java™*** is filled with over two hundred examples that demonstrate how to accomplish various important tasks related to file formats, data exchange, document transformation, and database integration. You will learn how to read and write XML documents with Java code, convert legacy flat files into XML documents, communicate with network servers that send and receive XML data, and much more. Readers will find detailed coverage of the following:

- How to choose the right API for the job
- Reading documents with SAX
- SAX filters
- Validation in several schema languages
- DOM implementations for Java
- The DOM Traversal Module
- Output from DOM
- Reading and writing XML documents with JDOM
- Searching XML documents with XPath
- Combining XSLT transforms with Java code
- TrAX, the Transformations API for XML
- JAXP, the Java API for XML Processing

In addition, the book includes a convenient quick reference that summarizes the major elements of all the XML APIs discussed. A related Web site, located at http://www.cafeconleche.org/books/xmljava/, contains the entire book in electronic format, as well as updates and links referenced in the book.

With thorough coverage of the key XML APIs and a practical, task-oriented approach, ***Processing XML with Java™*** is a valuable resource for all Java programmers who need to work with XML.

About the Author

Elliotte Rusty Harold is an internationally respected writer, programmer, and educator. He is an Adjunct Professor of Computer Science at Polytechnic University in Brooklyn, where he lectures on Java and object-oriented programming. His Cafe con Leche Web site has become one of the most popular sites for information on XML. In addition, he is the author and coauthor of numerous books, the most recent of which are *The XML Bible* (John Wiley & Sons, 2001) and *XML in a Nutshell* (O'Reilly, 2002).

0201771861AB06062003

One night five developers, all of whom wore very thick glasses and had recently been hired by Elephants, Inc., the world's largest purveyor of elephants and elephant supplies, were familiarizing themselves with the company's order processing system when they stumbled into a directory full of XML documents on the main server. "What's this?" the team leader asked excitedly. None of them had ever heard of XML before, so they decided to split up the files among them and try to figure out just what this strange and wondrous new technology was. The first developer, who specialized in optimizing Oracle databases, printed out a stack of FMPXMLRESULT documents generated by the FileMaker Pro database where all the orders were stored, and began poring over them. "So this is XML! Why, it's nothing novel. As anyone can see who's able, an XML document is nothing but a table!"

"What do you mean, a table?" replied the second developer, well versed in object-oriented theory and occupied with a collection of XMI documents that encoded UML diagrams for the system. "Even a Visual Basic programmer could see that XML documents aren't tables. Duplicates aren't allowed in a table relation, unless this is truly some strange mutation. Classes and objects are what these documents are. Indeed, it should be obvious on the very first pass. An XML document is an object, and a DTD is a class."

"Objects? A strange kind of object, indeed!" said the third developer, a web designer of some renown, who had loaded the XHTML user documentation for the order processing system into Mozilla."I don't see any types at all. If you think this is an object, then it's your software I refuse to install. But with all those stylesheets there, it should be clear to anyone not sedated that XML is just HTML updated!"

"HTML? You must be joking" said the fourth, a computer science professor on sabbatical from MIT, who was engrossed in an XSLT stylesheet that validated all of the other documents against a Schematron schema. "Look at the clean nesting of hierarchical structures, each tag matching its partner as it should. I've never seen HTML that looks this good. What we have here is S-expressions, which is certainly nothing new. Babbage invented this back in 1882!"

"S-expressions?" queried the technical writer, who was occupied with documentation for the project, written in DocBook. "Maybe that means something to those in your learned profession. But to me, this looks just like a FrameMaker MIF file. However, locating the GUI does seem to be taking me a while."

And so they argued into the night, none of them willing to give an inch, all of them presenting still more examples to prove their points, but none bothering to look at the others' examples. Indeed, they're probably still arguing today. You can even hear their shouts from time to time on xml-dev. Their mistake, of course, was in trying to force XML into the patterns of technologies they were already familiar with rather than taking it on its own terms. XML can store data, but it is not a database. XML can serialize objects, but an XML document is not an object. Web pages can be written in XML, but XML is not HTML. Functional (and other) programming languages can be written in XML, but XML is not a programming language. Books are written in XML, but that doesn't make XML desktop publishing software.

XML is something truly new that has not been seen before in the world of computing. There have been precursors to it, and there are always fanatics who insist on seeing XML through database (or object, or functional, or S-expression) colored glasses. But XML is none of these things. It is something genuinely unique and new in the world of computing; and it's possible to understand it only when you're willing to accept it on its own terms, rather than forcing it into yesterday's pigeonholes.

There are a lot of tools, APIs, and applications in the world that pretend XML is something more familiar to developers--that it's just a funny kind of database, or just like an object, or just like remote procedure calls. These APIs are occasionally useful in very restricted and predictable environments; however, they are not suitable for processing XML in its most general format. They work well in their limited domains, but they fail when presented with XML that steps outside the artificial boundaries they've defined. XML was designed to be extensible, but sadly many of the tools designed for XML aren't nearly as extensible as XML itself.

This book is going to show you how to handle XML in its full generality. It pulls no punches. It does not pretend that XML is anything except XML, and it shows you how to design your programs so that they handle real XML in all its messiness: valid and invalid, mixed and unmixed, typed and untyped, and both all and none of these at the same time. To that end, this book focuses on APIs that don't try to hide the XML. In particular, there are three major Java APIs that correctly model XML, as opposed to modeling a particular class of XML documents or some narrow subset of XML. These are

- SAX, the Simple API for XML
- DOM, the Document Object Model
- JDOM, a Java native API

These APIs are the core of this book. In addition, I cover a number of preliminaries and supplements to the basic APIs, including

- XML syntax
- DTDs, schemas, and validity
- XPath
- XSLT and the TrAX API
- JAXP, a combination of SAX, DOM, and TrAX with a few factory classes

And, since we're going to need a few examples of XML applications to demonstrate the APIs, I also cover XML-RPC, SOAP, and RSS in some detail. However, the techniques this book teaches are hardly limited to those three applications.

Who You Are

This book is written for experienced Java developers who want to integrate XML into their systems. Java is the ideal language for processing XML documents. Its strong Unicode support in particular made it the preferred language for many early implementers. Consequently, more XML tools have been written in Java than in any other language. More open source XML tools are written in Java than in any other language. More developers process XML in Java than in any other language.

Processing XML with Java™ will teach you how to

- Save XML documents from applications written in Java
- Read XML documents produced by other programs
- Search, query, and update XML documents
- Convert legacy flat data into hierarchical XML
- Communicate with network servers that send and receive XML data
- Validate documents against DTDs, schemas, and business rules
- Combine functional XSLT transforms with traditional imperative Java code

This book is intended for Java developers who need to do anything with XML. It teaches the fundamentals and advanced topics, leaving nothing out. It is a comprehensive course in processing XML with Java that takes developers from having little knowledge of XML to designing sophisticated XML applications and parsing complicated documents. The examples cover a wide range of possible uses, including file formats, data exchange, document transformation, database integration, and more.

What You Need to Know

This is not an introductory book with respect to either Java or XML. I assume you have substantial prior experience with Java and preferably some experience with XML. On the Java side, I freely use advanced features of the language and its class library without explanation or apology. Among other things, I assume you are thoroughly familiar with the following:

- Object-oriented programming, including inheritance and polymorphism.
- Packages and the CLASSPATH. You should not be surprised by classes that do not have main() methods or that are not in the default package.

- I/O including streams, readers, and writers. You should understand that System.out is a horrible example of what really goes on in Java programs.
- The Java Collections API including hash tables, maps, sets, iterators, and lists.

In addition, in one or two places in this book I use some SQL and JDBC. These sections are relatively independent of the rest of the book, however, and chances are if you aren't already familiar with SQL, then you don't need the material in these sections anyway.

What You Need to Have

XML is deliberately architecture, platform, operating system, GUI, and language agnostic (in fact, more so than Java). It works equally well on Mac OS, Windows, Linux, OS/2, various flavors of Unix, and more. It can be processed with Python, C++, Haskell, ECMAScript, C#, Perl, Visual Basic, Ruby, and of course Java. No byte-order issues need concern you if you switch between PowerPC, X86, or other architectures. Almost everything in this book should work equally well on any platform that's capable of running Java.

Most of the material in this book is relatively independent of the specific Java version. Java 1.4 bundles SAX, DOM, and a few other useful classes into the core JDK. However, these are easily installed in earlier JVMs as open source libraries from the Apache XML Project and other vendors. For the most part, I used Java 1.3 and 1.4 when testing the examples; therefore, it's possible that a few of the classes and methods used are not available in earlier versions. In most cases, it should be fairly obvious how to backport them. All of the basic XML APIs except TrAX should work in Java 1.1 and later. TrAX requires Java 1.2 or later.

How to Use This Book

This book is organized as an advanced tutorial that can also serve as a solid and comprehensive reference. Chapter 1 covers the bare minimum material needed to start working with XML, although for the most part this is not intended as a comprehensive introduction, but more as a review for readers who already have read other, more basic books. Chapter 2 introduces RSS, XML-RPC, and SOAP, the XML applications used for examples throughout the rest of the book. This is followed by two chapters on generating XML from your own programs (a subject all too often presented as a lot more complicated than it actually is). Chapter 3 covers generating XML directly from code, and Chapter 4 covers converting legacy data in other formats to XML. The remaining bulk of the book is devoted to the major APIs for processing XML:

- The event-based SAX API
- The tree-based DOM API
- The tree-based JDOM API
- XPath APIs for searching XML documents
- The TrAX API for XSLT processing

Finally, the book finishes with an appendix providing quick references to the main APIs.

If you have limited experience with XML, I suggest that you read at least the first five chapters in order. From that point forward, if you have a particular API preference, you may begin with the part that covers the major API you're interested in:

- Chapters 6 to 8 cover SAX.
- Chapters 9 to 13 cover DOM.
- Chapters 14 and 15 cover JDOM.

Once you're comfortable with one or more of these APIs, you can read Chapters 16 and 17 on XPath and

XSLT. However, those APIs and chapters do require some knowledge of at least one of the three major APIs.

The Online Edition

The entire book is available online in plain-vanilla HTML at my Cafe con Leche web site. You can find it at http://www.cafeconleche.org/books/xmljava/. Every word of this book is there. Nothing has been held back or left out. I do hope you also find the printed book useful and choose to buy it--it's certainly cheaper than the paper and toner you'd use up printing out all 1,120 pages from your laser printer--but you are by no means obligated to do so. My goal is to make this material as broadly available and useful as possible.

The online version has no protection other than copyright law and your own good will. You don't need to register to read it, or to download some special electronic key that becomes invalid when you buy a new laptop (and that probably wouldn't run on Linux or a Mac in the first place). I want people to read and use this book. I do not want to put up silly roadblocks that make it less useful than it could be. I do ask, as a courtesy, that you do not republish the online edition on your own server. Doing so makes it extremely difficult for me to keep the book up to date. If you want to save a few pages on your laptop so you can read this book on an airplane, I don't really mind. But please don't pass out your own copies to anyone else. Instead, refer your friends and colleagues to the web site or the printed book.

Some Grammatical Notes

The rules of English grammar were laid down, written in stone, and encoded in the DNA of elementary school teachers long before computers were invented. Unfortunately, this means that sometimes I have to decide between syntactically correct code and syntactically correct English. When forced to do so, English normally loses. This means that sometimes a punctuation mark appears outside a quotation mark when you'd normally expect it to appear inside, a sentence begins with a lowercase letter, or something similarly unsettling occurs. For the most part, I've tried to use various typefaces to make the offending phrase less jarring. In particular, please note the following:

- *Italicized* text is used for emphasis, the first occurrence of an important term, titles of books and other cited works, words in languages other than English, words as words themselves (for example, Booboisie is a very funny word), Java system properties, host names, and resolvable URLs.
- `Monospaced` text is used for XML and Java source code, namespace URLs, system prompts, and program output.
- *`Italicized monospace`* text is used for pieces of XML and Java source code that should be replaced by some other text.
- **`Bold monospaced`** text is used for literal text that the user types at a command line, as well as for emphasis in code.

It's not just English grammar that gets a little squeezed, either. The necessities of fitting code onto a printed page rather than a computer screen have occasionally caused me to deviate from the ideal Java coding conventions. The worst problem is line length. I can fit only 65 characters across the page in a line of code. To try to make maximum use of this space, I indent each block by two spaces and indent line continuations by one space, rather than the customary four spaces and two spaces respectively. Even so, I still have to break lines where I otherwise would prefer not to. For example, I originally wrote this line of code for Chapter 4:

```
result.append(" " + amount + " ");
```

To fit it on the page, however, I had to split it into two pieces, like this:

```
result.append(" ");
result.append(amount +" ");
```

This wasn't too bad, but sometimes even this wasn't enough and I had to remove indents from the front of the line that would otherwise be present. This occasionally forced the indentation not to line up as prettily as it otherwise might, as in this example from Chapter 3:

```
wout.write(
"xmlns='http://namespaces.cafeconleche.org/xmljava/ch3/' "
);
```

The silver lining to this cloud is that sometimes the extra attention I give to the code when I'm trying to cut down its size results in better code. For example, in Chapter 4, I found I needed to remove a few characters from this line:

```
OutputStreamWriter wout = new OutputStreamWriter(out, "UTF8");
```

On reflection I realized that nowhere did the program actually need to know that wout was an OutputStreamWriter as opposed to merely a Writer. Thus I could easily rewrite the offending line as follows:

```
Writer wout = new OutputStreamWriter(out, "UTF8");
```

This follows the general object-oriented principle of using the least-specific type that will suit. This polymorphism makes the code more flexible in the future should I find a need to swap in a different kind of Writer.

Contacting the Author

I always enjoy hearing from readers, whether with general comments, specific ways I could improve the book, or questions related to the book's subject matter. Because this book is being published in its entirety online, it is possible for me to reprint at least the online edition much faster than can be done with a traditional paper book. Thus corrections and errata are especially helpful because I have a real chance to fix them. Before sending in a correction, please do check the online edition to see if I have already fixed the problem.

Please send all comments, inquiries, bouquets, and brickbats to elharo@ metalab.unc.edu. I get a lot of e-mail, so I can't promise to answer them all; but I do try. It's helpful if you use a subject line that clearly identifies yourself as a reader of this book. Otherwise, your message may accidentally get misidentified as spam I don't want or bulk mail I don't have time to read and be dropped in the bit bucket before I see it. Also, please make absolutely sure that your message uses the correct reply-to address and that the address will be valid for at least several months after you send the message. There's nothing quite as annoying as taking an hour or more to compose a detailed response to an interesting question, only to have it bounce because the reader sent the email from a public terminal or changed their ISP. But please do write to me. I want to hear from you.

Elliotte Rusty Harold
Brooklyn, New York
June 7, 2002

0201771861P10222002

Users Review

From reader reviews:

Lois Yale:

What do you about book? It is not important with you? Or just adding material when you really need something to explain what your own problem? How about your free time? Or are you busy individual? If you don't have spare time to perform others business, it is make one feel bored faster. And you have free time? What did you do? All people has many questions above. They should answer that question simply because just their can do which. It said that about guide. Book is familiar in each person. Yes, it is proper. Because start from on kindergarten until university need that Processing XML with Java¿: A Guide to SAX, DOM, JDOM, JAXP, and TrAX (2 Volume Set) to read.

Kimberly Spradlin:

This Processing XML with Java¿: A Guide to SAX, DOM, JDOM, JAXP, and TrAX (2 Volume Set) book is just not ordinary book, you have after that it the world is in your hands. The benefit you obtain by reading this book is information inside this book incredible fresh, you will get info which is getting deeper an individual read a lot of information you will get. This Processing XML with Java¿: A Guide to SAX, DOM, JDOM, JAXP, and TrAX (2 Volume Set) without we realize teach the one who reading through it become critical in considering and analyzing. Don't be worry Processing XML with Java¿: A Guide to SAX, DOM, JDOM, JAXP, and TrAX (2 Volume Set) can bring any time you are and not make your case space or bookshelves' come to be full because you can have it in your lovely laptop even cell phone. This Processing XML with Java¿: A Guide to SAX, DOM, JDOM, JAXP, and TrAX (2 Volume Set) having excellent arrangement in word and also layout, so you will not sense uninterested in reading.

Pedro Gonzales:

Beside this specific Processing XML with Java¿: A Guide to SAX, DOM, JDOM, JAXP, and TrAX (2 Volume Set) in your phone, it can give you a way to get more close to the new knowledge or information. The information and the knowledge you will got here is fresh from the oven so don't be worry if you feel like an older people live in narrow community. It is good thing to have Processing XML with Java¿: A Guide to SAX, DOM, JDOM, JAXP, and TrAX (2 Volume Set) because this book offers to you personally readable information. Do you at times have book but you rarely get what it's interesting features of. Oh come on, that will not happen if you have this with your hand. The Enjoyable option here cannot be questionable, such as treasuring beautiful island. Use you still want to miss this? Find this book along with read it from at this point!

Ali Ellison:

E-book is one of source of information. We can add our information from it. Not only for students but native or citizen want book to know the update information of year to be able to year. As we know those books have many advantages. Beside we add our knowledge, can bring us to around the world. With the book Processing XML with Java¿: A Guide to SAX, DOM, JDOM, JAXP, and TrAX (2 Volume Set) we can consider more advantage. Don't someone to be creative people? To be creative person must choose to read a book. Merely choose the best book that appropriate with your aim. Don't always be doubt to change your life with this book Processing XML with Java¿: A Guide to SAX, DOM, JDOM, JAXP, and TrAX (2 Volume Set). You can more desirable than now.

Read Processing XML with Java¿: A Guide to SAX, DOM, JDOM, JAXP, and TrAX (2 Volume Set) By Elliotte Rusty Harold for online ebook

Processing XML with Java¿: A Guide to SAX, DOM, JDOM, JAXP, and TrAX (2 Volume Set) By Elliotte Rusty Harold Free PDF d0wnl0ad, audio books, books to read, good books to read, cheap books, good books, online books, books online, book reviews epub, read books online, books to read online, online library, greatbooks to read, PDF best books to read, top books to read Processing XML with Java¿: A Guide to SAX, DOM, JDOM, JAXP, and TrAX (2 Volume Set) By Elliotte Rusty Harold books to read online.

Online Processing XML with Java¿: A Guide to SAX, DOM, JDOM, JAXP, and TrAX (2 Volume Set) By Elliotte Rusty Harold ebook PDF download

Processing XML with Java¿: A Guide to SAX, DOM, JDOM, JAXP, and TrAX (2 Volume Set) By Elliotte Rusty Harold Doc

Processing XML with Java¿: A Guide to SAX, DOM, JDOM, JAXP, and TrAX (2 Volume Set) By Elliotte Rusty Harold Mobipocket

Processing XML with Java¿: A Guide to SAX, DOM, JDOM, JAXP, and TrAX (2 Volume Set) By Elliotte Rusty Harold EPub