Professional XML

Bill Evjen,
Kent Sharkey,
Thiru Thangarathinam,
Michael Kay,
Alessandro Vernet,
Sam Ferguson

Wiley Publishing, Inc.

Professional XML

Bill Evjen,
Kent Sharkey,
Thiru Thangarathinam,
Michael Kay,
Alessandro Vernet,
Sam Ferguson

BICENTENNIAL 1807 WILEY 2007 BICENTENNIAL

Wiley Publishing, Inc.

Professional XML

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

Published simultaneously in Canada

ISBN: 978-0-471-77777-9

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

Library of Congress Cataloging-in-Publication Data:

Professional XML / Bill Evjen . . . [et al.].
p. cm.
Includes index.
ISBN-13: 978-0-471-77777-9 (paper/website)
ISBN-10: 0-471-77777-3 (paper/website)
1. XML (Document markup language) I. Evjen, Bill.
QA76.76.H94P7638 2007
006.7'4—dc22
2007006214

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 THEN 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.

Credits

Senior Acquisitions Editor
Jim Minatel

Development Editor
Sydney Jones

Technical Editors
Alexei Gorkov
Steve Danielson
Cody Reichenau

Production Editor
William A. Barton

Copy Editors
Mary Lagu
Kathryn Duggan

Editorial Manager
Mary Beth Wakefield

Vice President & Executive Group Publisher
Richard Swadley

Vice President and Publisher
Joseph B. Wikert

Project Coordinator
Erin Smith

Graphics and Production Specialists
Jonelle Burns
Carrie A. Foster
Brooke Graczyk
Denny Hager
Jennifer Mayberry
Barbara Moore
Rashell Smith
Alicia B. South

Quality Control Technicians
Laura Albert
Christy Pingleton

Proofreading and Indexing
Aptara

Anniversary Logo Design
Richard J. Pacifico

To my three little ones—Sofia, Henri, and Kalle.—Bill Evjen

To Babi, for keeping me alive, and putting up with me. Hopefully for a long time to come.—Kent Sharkey

Thanks to my beautiful wife Jacquie for her patience, my cat Peggy for her company (and technical input), and to all my colleagues at API who continue to inspire and challenge me.—Sam Ferguson

About the Authors

Lead Authors

Bill Evjen is an active proponent of .NET technologies and community-based learning initiatives for .NET. He has been actively involved with .NET since the first bits were released in 2000. In the same year, Bill founded the St. Louis .NET User Group (`www.stlnet.org`), one of the world's first such groups. Bill is also the founder and former executive director of the International .NET Association (`www.ineta.org`), which represents more than 450,000 members worldwide.

Based in St. Louis, Missouri, USA, Bill is an acclaimed author (more than 13 books to date) and speaker on ASP.NET and XML Web services. He has written or co-written *Professional C# 2005; Professional VB 2005*; and the best-selling *Professional ASP.NET 2.0,* as well as *ASP.NET Professional Secrets*, *XML Web Services for ASP.NET*, *Web Services Enhancements: Understanding the WSE for Enterprise Applications*, *Visual Basic .NET Bible*, and more. In addition to writing, Bill is a speaker at numerous conferences, including DevConnections, VSLive, and TechEd. Along with these items, Bill works closely with Microsoft as a Microsoft Regional Director and he has received the Microsoft MVP designation for many years.

Bill is the Technical Architect for Lipper (`www.lipperweb.com`), a wholly-owned subsidiary of Reuters, the international news and financial services company. He graduated from Western Washington University in Bellingham, Washington, with a Russian language degree. When he isn't tinkering on the computer, he can usually be found at his summer house in Toivakka, Finland. You can reach Bill at `evjen@yahoo.com`. He presently keeps his weblog at `www.geekswithblogs.net/evjen`.

Kent Sharkey is an independent consultant who lives and codes in the midst of the wilds of Vancouver Island. Before going solo, Kent worked at Microsoft as a Technical Evangelist and Content Strategist, promoting the use of .NET technologies. When not coding or writing, he's off hiking, biking, or canoeing (or exploring the wilds of Azeroth). He shares his house with his wife, Margaret, and two "children," Squirrel and Cica.

Contributing Authors

Thiru Thangarathinam is a Microsoft MVP who specializes in architecting, designing, and developing distributed enterprise class applications using .NET-related technologies. He is the author of the books *Professional ASP.NET 2.0 XML* and *Professional ASP.NET 2.0 Databases* from Wrox press and has coauthored a number of books on .NET-related technologies. He is a frequent contributor to leading technology-related online publications.

Michael Kay is widely known in the XML world as the developer of the Saxon XSLT and XQuery processor, and as the editor of the XSLT 2.0 specification. His Wrox books *XSLT 2.0 Programmer's Reference* and *XPath 2.0 Programmer's Reference* are regarded as the definitive guides to these languages. Michael runs his own company, Saxonica, which develops the Saxon technology and provides support and consultancy for XSLT and XQuery users. His background is as a software designer creating database products for a mainframe manufacturer. He is a Fellow of the British Computer Society and a Visiting Fellow at the University of Reading (UK). In his spare time he sings and plays croquet.

About the Authors

Alessandro Vernet co-founded Orbeon in 1999, which makes Orbeon Forms, an open source product to build and deploy sophisticated forms on the Web. He is one of the authors of *Professional Web 2.0 Programming* and is a member of two W3C Working Groups: the XForms and XML Processing Model Working Groups. Before co-founding Orbeon, Alessandro was at Symantec as part of the VisualCafé team, working on their next-generation RAD for web applications. He holds an MS/CS from the Swiss Institute of Technology (EPFL) in Lausanne, Switzerland, and since 1998 lives in the incredibly energetic Silicon Valley.

Sam Ferguson is a Project Manager with API Software, a Microsoft Gold Certified Partner, based in Glasgow, Scotland. Sam, who lives in Ayrshire, specializes in SQL Server, Microsoft Office Server System 2007, .NET, and all XML-related technologies. In what little spare time he has, Sam enjoys playing golf and is an avid fan of the Glasgow Rangers.

Acknowledgments

This book, like most, took an entire team to get out the door. First and foremost, I would like to thank Jim Minatel for providing me the opportunity to write this book. Big thanks go to Sydney Jones, the book’s development editor who kept this book together despite numerous delays.

Thanks to my family for putting up with this second job. I love you all greatly.—Bill Evjen

Contents

Introduction

As many people predicted, XML has changed the world! When XML was introduced, many considered it a revolutionary leap forward in how we look at data representation. Since then, XML has grown considerably and many new technologies have been introduced. The number of new technologies based upon XML is staggering. From Web services, to blogging, to alerts and notifications—there is so much coming out today completely based upon this technology.

This book covers not just the basics of XML and the XML specification, but it also takes a look at the technologies based on XML that are driving the tech industry forward. This book not only introduces these technologies to you, but it also shows you examples of these new technologies in action. So sit back, pull up that keyboard, and let's have some fun!

What You Need for This Book

This book is vendor agnostic as XML can be utilized on any major operating system out there. But this book is also about various technologies that are based upon XML, and for this reason, you will sometimes need specific vendor tools to complete the examples provided in the chapters. In these cases, the locations of the tools or technologies and where you can acquire them are called out in the chapter.

You will find that this book focuses on both Microsoft- and Java-based technologies, and therefore, you are sometimes asked to work with that vendor's specific development tools and environments to complete the examples.

Who Should Read This Book?

This book was written to provide you with the latest and greatest information on XML, and to look at the new technologies and capabilities being built on XML today. We assume you have a general understanding of programming technologies, such as C# or Java. If you understand the basics of these programming languages, then you shouldn't have much trouble following along with this book's content.

If you are brand new to XML, be sure to check out *Beginning XML* by David Hunter and others (published by Wrox; ISBN: 978-0-7645-7077-3) to help you understand the basics.

You may also be wondering whether this book is focused on the Microsoft developer or the Java developer. We're happy to say that it's for both! You will find that there are chapters focused on each of these programming technologies.

What This Book Covers

This book spends its time reviewing the big changes that have occurred in the 2.0 release of ASP.NET. Each major new feature included in ASP.NET 2.0 is covered in detail. The following list tells you something about the content of each chapter.

- ❑ **Chapter 1, "XML Syntax."** This first chapter gives a good grounding in the XML specification. This chapter looks at the reasoning to include XML and its related technologies within an application's architecture. In addition to this introduction, this chapter will also look at the syntactical rules of the XML markup language.
- ❑ **Chapter 2, "XML Editors."** This chapter takes a look at the XML tools that can be used for working with XML and its related technologies. These are also tools used in the chapters throughout the book. This chapter introduces you to the various tools out there, where they can be found, as well as some basics for working with them.
- ❑ **Chapter 3, "XHTML and CSS."** The next set of chapters looks at presenting XML and XML-based presentation technologies. This first chapter focuses on the popular XHTML specification and how it can be used to present content within a browser. In continuing on with the presentation theme, this chapter looks at how to present XML documents visually in a browser using Cascading Style Sheets.
- ❑ **Chapter 4, "XSL-FO."** This chapter takes a look at presenting XML using an XML-based presentation markup language—Extensible Stylesheet Language Formatting Objects (XSL-FO), also known as simply XSL.
- ❑ **Chapter 5, "Document Type Definitions (DTDs)."** This chapter takes a look at defining an XML structure using DTDs which have been around for quite awhile. Though it is preferred to use XML Schema today, DTDs should still be understood as developers may encounter legacy XML documents.
- ❑ **Chapter 6, "XML Schemas."** XML Schemas are the latest and most preferred way to define the structure and data types of an XML document. This is an important topic to understand as it is referenced throughout the book. This chapter takes a deep look into XML Schemas and how to build them.
- ❑ **Chapter 7, "RELAX NG."** RELAX NG is a new XML specification that allows you to validate an XML structure as well as make a link to datatype libraries. This standard was born to simplify what was perceived as difficult and complicated about XML Schema. This chapter looks at how to work with RELAX NG.
- ❑ **Chapter 8, "XSLT."** This chapter looks at transforming XML documents in a multitude of ways using XSLT. Starting with the basics of XSLT, this chapter gives you an understanding of how and when to use this technology.
- ❑ **Chapter 9, "XPath."** XPath allows for the searching and manipulation of particular subsets of an XML document. This chapter takes a look at this popular technology and will focus on the latest release of XPath—version 2.0.
- ❑ **Chapter 10, "XQuery."** XQuery is another search and manipulation technology that often competes with XPath. This chapter takes a look at XQuery and what makes it different than XPath. By the end of chapters 9 and 10, you should have a good understanding of both XPath and XQuery and when to use which technology.

- ❑ **Chapter 11, "XML in the Data Tier."** In the continuing look at XML as data, this chapter focuses on the use of XML in the data tier. With focuses on Microsoft's SQL Server 200 and SQL Server 2005, as well as Oracle, MySQL, and more, this chapter shows what XML capabilities there are for working with the various data storage technologies.
- ❑ **Chapter 12, "XML Document Object Model (DOM)."** This chapter will take a look at using the XML-DOM to program your XML documents. Included in this chapter is an introduction to the XML Document Object Model as well as information about how to parse XML using the XML-DOM.
- ❑ **Chapter 13, "Simple API for XML (SAX)."** Another method to use to program your XML documents is through the use of SAX. By the end of this chapter, you should understand the differences between working with the XML-DOM and SAX and when to use which technology.
- ❑ **Chapter 14, "Ajax."** One of the more talked about programming technologies of 2005 and 2006 has been programming using XMLHTTP, also known as Ajax. Google has made this programming style popular through its use in various Google applications; the use of this asynchronous JavaScript and XML programming technique is now a sought after feature. This chapter takes a look at XMLHTTP and how to use this object to build truly unique applications.
- ❑ **Chapter 15, "XML and .NET."** This chapter takes a look at using the `System.Xml` namespace and other XML capabilities that are provided with Microsoft's .NET Framework. Also covered is XML in the Microsoft development space and reading and writing XML using the classes provided via the .NET Framework 2.0.
- ❑ **Chapter 16, "XML and Java."** This chapter takes a look at using XML with the Java language and what XML documents can do within a Java application environment.
- ❑ **Chapter 17, "Dynamic Languages and XML."** This chapter takes a look at how to use XML with a PHP application. Also reviewed will be XML with Perl and Ruby.
- ❑ **Chapter 18, "RSS and Atom."** One big use of XML as of late has been in regards to content syndication. Both RSS and Atom (competing standards) offer the ability to expose content for aggregation purposes. This chapter takes a close look at both of these technologies and how they can be effectively used.
- ❑ **Chapter 19, "Web Services."** Getting beyond the hype, this chapter takes an introductory look at Web services and what it actually means to expose content and logic as SOAP in this disparate world.
- ❑ **Chapter 20, "SOAP and WSDL."** Digging deeper into Web services, this chapter takes a look at the main specification in the Web services world—SOAP. This chapter will focus on SOAP documents including looking at SOAP headers and SOAP faults. Furthering the discussion around SOAP-based Web services, this chapter looks at the WSDL and UDDI specifications. WSDL is used to define a SOAP interface, whereas UDDI is used to locate services. Both of these specifications will be discussed in detail.
- ❑ **Chapter 21, "Advanced Web Services."** This chapter takes a look at these advanced specifications and what they do for your Web services. In addition to examining the specifications, this chapter also describes implementing these specifications in your applications today.
- ❑ **Chapter 22, "REST."** REST, a competing standard to SOAP, is heavily used in the UNIX world. In fact, companies such as Amazon have seen considerable success in using REST compared to using SOAP. This chapter will take a look at exposing data and services using "the other" standard.

- **Chapter 23, "XML Form Development."** Forms, as popular as they are on the Web, can now be defined using XForms. This chapter takes a close look at XForms and how it can be used, along with other XML-based technologies, to produce various types of forms.
- **Chapter 24, "The Resource Description Framework (RDF)."** The RDF specification allows for the relation of metadata to presentation content. RDF is a framework for describing and interchanging metadata and is introduced in this chapter as well as demonstrated through some examples.
- **Chapter 25, "XML in Office Development."** This chapter takes a close look at using XML in Office development with a particular focus on Microsoft Office.
- **Chapter 26, "XAML."** This chapter takes a look at XAML—the new way Microsoft is promoting the presentation of a GUI. XAML provides for presentation that is fluid and enriching. This chapter takes a look at the basics of XAML and how to build some basic XAML applications.

Conventions

This book uses a number of different styles of text and layout to help differentiate among various types of information. Here are examples of the styles used and an explanation of what they mean:

- New words being defined are shown in *italics*.
- Keys that you press on the keyboard, such as Ctrl and Enter, are shown in initial caps and spelled as they appear on the keyboard.
- File and folder names, file extensions, URLs, and code that appear in regular paragraph text are shown in a `monospaced` typeface.

When we show a block of code that you can type as a program and run, it's shown on separate lines, like this:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Employee>
   <FirstName>Bill</FirstName>
   <LastName>Evjen</LastName>
   <JobTitle>Technical Architect</JobTitle>
   <Company>Lipper</Company>
   <StartDate>10/04/2001</StartDate>
   <WorkLocation>St. Louis, Missouri</WorkLocation>
   <NumberOfDependents>3</NumberOfDependents>
</Employee>
```

or like this:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Employee>
   <FirstName>Bill</FirstName>
   <LastName>Evjen</LastName>
   <JobTitle>Technical Architect</JobTitle>
   <Company>Lipper</Company>
   <StartDate>10/04/2001</StartDate>
```

```
      <WorkLocation>St. Louis, Missouri</WorkLocation>
      <NumberOfDependents>3</NumberOfDependents>
   </Employee>
```

Sometimes you see code in a mixture of styles, like this:

```
   <?xml version="1.0" encoding="UTF-8" ?>
   <Employee>
      <FirstName>Bill</FirstName>
      <LastName>Evjen</LastName>
      <JobTitle>Technical Architect</JobTitle>
      <Company>Lipper</Company>
      <StartDate>10/04/2001</StartDate>
      <WorkLocation>St. Louis, Missouri</WorkLocation>
      <NumberOfDependents>3</NumberOfDependents>
   </Employee>
```

When mixed code is shown like this, the code with no background represents code that has been shown previously and that you don't need to examine further. Code with the gray background is what you should focus on in the current example.

We demonstrate the syntactical usage of methods, properties, and so on using the following format:

```
   <?[target] [data]?>
```

Here, the italicized parts indicate *placeholder text:* object references, variables, or parameter values that you need to insert.

Most of the code examples throughout the book are presented as numbered listings that have descriptive titles, like this:

Listing 1-8: Creating an XML file

Each listing is numbered (for example: *1-8*) where the first number represents the chapter number and the number following the hyphen represents a sequential number that indicates where that listing falls within the chapter. Downloadable code from the Wrox Web site (`www.wrox.com`) also uses this numbering system so that you can easily locate the examples you are looking for.

Source Code

As you work through the examples in this book, you may choose either to type all the code manually or to use the source code files that accompany the book. All the source code used in this book is available for download at `www.wrox.com`. When you get to the site, simply locate the book's title (either by using the Search box or one of the topic lists) and click the Download Code link. You can then choose to download all the code from the book in one large zip file or download just the code you need for a particular chapter.

> *Because many books have similar titles, you may find it easiest to search by ISBN; this book's ISBN is 978-0-471-77777-9.*

After you download the code, just decompress it with your favorite compression tool. Alternatively, you can go to the main Wrox code download page at `www.wrox.com/dynamic/books/download.aspx` to see the code available for this book and all other Wrox books. Remember, you can easily find the code you are looking for by referencing the listing number of the code example from the book, such as "Listing 1-8." We used these listing numbers when naming the downloadable code files.

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 if you'd tell us about it. By sending in errata, you may spare another reader hours of frustration; at the same time, you are helping us provide even higher-quality information.

To find the errata page for this book, go to `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 already 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 technologies and to interact with other readers and technology users. The forums offer a subscription feature that enables you to receive e-mail on topics of interest when new posts are made to the forums. Wrox authors, editors, other industry experts, and your fellow readers are represented in 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. Supply the information required to join, as well as any optional information you want to provide, and click Submit.