Pro SQL Server 2008 XML

Michael Coles

Apress®

Pro SQL Server 2008 XML

ISBN-13: 978-1-59059-983-9

ISBN-10: 1-59059-983-7

ISBN-13 (electronic): 978-1-4302-0630-9

ISBN-10 (electronic): 1-4302-0630-6

Printed and bound in the United States of America 9 8 7 6 5 4 3 2 1

Lead Editor: Jonathan Gennick
Technical Reviewer: Fabio Claudio Ferracchiati
Editorial Board: Clay Andres, Steve Anglin, Ewan Buckingham, Tony Campbell, Gary Cornell, Jonathan Gennick, Matthew Moodie, Joseph Ottinger, Jeffrey Pepper, Frank Pohlmann, Ben Renow-Clarke, Dominic Shakeshaft, Matt Wade, Tom Welsh
Project Manager: Kylie Johnston
Copy Editor: Kim Benbow
Associate Production Director: Kari Brooks-Copony
Production Editor: Liz Berry
Compositor/Artist: Kinetic Publishing Services, LLC
Proofreader: April Eddy
Indexer: Becky Hornyak
Cover Designer: Kurt Krames
Manufacturing Director: Tom Debolski
Distributed to the book trade worldwide by Springer-Verlag New York, Inc., 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax 201-348-4505, e-mail orders-ny@springer-sbm.com, or visit http://www.springeronline.com.

For information on translations, please contact Apress directly at 2855 Telegraph Avenue, Suite 600, Berkeley, CA 94705. Phone 510-549-5930, fax 510-549-5939, e-mail info@apress.com, or visit http://www.apress.com.

Apress and friends of ED books may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Special Bulk Sales–eBook Licensing web page at http://www.apress.com/info/bulksales.

The source code for this book is available to readers at http://www.apress.com.

For Devoné and Rebecca

Contents at a Glance

Contents

Foreword

It is my pleasure to present Michael Coles's book, *Pro SQL Server 2008 XML*, which covers one area of Microsoft® SQL Server™ that I spend a large part of my work at Microsoft designing, influencing, and building: the XML support in SQL Server.

Michael covers this complex topic with the focused understanding of a practitioner and the deep background of an experienced industry observer. He presents this large and, to the general database programmer, often somewhat new and surprising area, both in an easy and logical way—covering the client application, .Net programming with XML, and the database development aspects. Lots of examples provide access to the concepts and technologies, and practical tips about usage and performance add relevance. He even shows how XML is being used with other features in SQL Server, such as the spatial support added in SQL Server 2008, which provides support for a subset of GML, Bulk copy's use of XML and XML for Analysis.

His technical content is presented against a historical background of what XML is about, and also points out that XML is not the cure for all ailments in this world, but that it has its place to address several important scenarios.

When I started to work on the XML support in SQL Server in the SQL Server 2000 release cycle, we set out on a journey to provide extensions to our relational database customers that would enable them to work with XML data to address three different but related scenarios that were united in that they all were using XML at some level.

The first scenario, which we focused on in SQL Server 2000, was the ability to integrate existing relational data into the new world of web service and loosely coupled data exchange that was starting to use XML as their lingua franca of data interchange. Features like FOR XML or the mapping schemas of the SQL/XML component were allowing programmers to map their existing relational data into XML and take structured data from XML into their existing relational database.

Since XML with its tag markup structure is well-suited to describe complex, non-regular data shapes, it also quickly became a preferred way by many to represent data that did not easily fit into the relational mold. XML was used either because the data shape was changing too quickly or was not known a priori, or the decomposition and re-composition costs of the complex properties were too high and XML gave a good compromise between queryability and flexibility. This second scenario, often referred to as the scenario of managing semi-structured data management, got support in SQL Server 2005 with the addition of the XML data type, XML Schema collections, and the support for XQuery to query into the XML structure and unlock the information within it. SQL Server 2008 has now even added relational functionality—such as sparse columns and column sets, themselves based on XML—to provide more "relational" support for semi-structured data.

Finally, over the last few years, more and more documents are being represented in XML, be it custom schemas or some standard document schemas such as the Office OpenXML (not to be confused with the OpenXML function in SQL Server) and others. As such uses for XML become more prevalent, the queryability of the underlying document format, namely XML, will

become more important and the current support for XML in SQL Server 2008 will provide a solid foundation to manage such documents together with the other business data.

Michael's experience, written down in this book, gives you good guidance and insight into how the different SQL Server XML technologies can help you with these and similar scenarios. Whether you are a newcomer to SQL Server's XML Support or a seasoned XML user, you will find lots of value in this book with its practical advice and easy to understand explanations and examples.

Enjoy the book!

Michael Rys
Principal Program Manager, Microsoft

About the Author

MICHAEL COLES has over a dozen years' experience in SQL database design, T-SQL development, and client-server application programming. He has consulted in a wide range of industries, including the insurance, financial, retail, and manufacturing sectors, among others. Michael's specialty is developing and performance-tuning high-profile SQL Server–based database solutions. He currently works as a consultant for a business intelligence consulting firm. He holds a bachelor's degree in information technology and multiple Microsoft and other certifications.

Michael has published dozens of highly rated technical articles online and in print magazines, including *SQL Server Central*, *ASP Today*, and *SQL Server Standard* magazines. Michael is the author of the book *Pro T-SQL 2005 Programmer's Guide* (Apress, 2007) and a contributor to *Accelerated SQL Server* 2008 (Apress, 2008).

About the Technical Reviewer

FABIO CLAUDIO FERRACCHIATI is a senior consultant and a senior analyst/developer using Microsoft technologies. He works for Brain Force (`www.brainforce.com`) in its Italian branch (`www.brainforce.it`). He is a Microsoft Certified Solution Developer for .NET, a Microsoft Certified Application Developer for .NET, a Microsoft Certified Professional, and a prolific author and technical reviewer. Over the past ten years, he's written articles for Italian and international magazines and coauthored more than ten books on a variety of computer topics. You can read his LINQ blog at `www.ferracchiati.com`.

Acknowledgments

Bringing a book like this to you, the reader, is not a solo act. Even though Apress was kind enough to put my name on the cover, there's no way you would be reading these words if not for the entire team at Apress. This book is the product of the work of dozens of my Apress teammates.

With that in mind, I would like to start by thanking my editor Jonathan Gennick, who pulled this project together and oversaw it from first concept to press. I would also like to thank the chief cat-herder, project manager Kylie Johnston, who made sure everyone did what they were supposed to, when they were supposed to. I want to send a special thank you to the technical reviewer and resident LINQ expert, Fabio Claudio Ferracchiati, for keeping me honest. I would also like to thank those team members who spent countless hours copy editing, laying out pages, and contributing in numerous other ways to bring this book to you.

I would like to send an extra special thank you to Michael Rys, project manager for SQL Server XML technology and member of the W3C XML Query Working Group. Thank you, Mr. Rys, for taking the time to answer all of the wild SQL Server XML questions I kept coming up with. I would also like to thank Steve Jones, entrepreneur extraordinaire of *SQL Server Central* fame and SQL Server MVP, and Chuck Heinzelman, editor in chief of *SQL Server Standard* magazine, for their support.

Finally, I would like to thank my family, including my mom, Eric, Jennifer, Chris, Desmond, and Deja. I'd also like to thank my Aunt Linda and her family for their support. And a special thank you to my girlfriend, Donna, for being so supportive and understanding during this process.

And most important, thank you to Devoné and Rebecca—my little angels—for keeping a smile on my face.

Introduction

Back in 1999, I was working for a dot-com when I was first exposed to the interesting new technology known as XML. My imagination ran wild, and I immediately began working on dozens of applications for XML, like custom reporting from our legacy inventory system and Web-based business-to-business ordering. I was as excited as a kid on Christmas morning!

My excitement about XML in those early days began to fade as I quickly ran into its limitations, however. For one thing, the basic XML standard did not define a type system. All content in XML is character data—strings. This meant that I needed to not only convert string data to the correct data type in my own applications, but I also had to validate the content and handle boatloads of potential exceptions in each application.

Another thing that bothered me was the strangely cryptic and underpowered Document Type Definition (DTD) system, the primary means of constraining XML document structure and content. Again, the majority of structure and content validation became the responsibility of the application, making it even more complicated. And the fact that querying XML data required inefficient loops or event-based parsing and comparisons didn't help matters much.

Fortunately for us, XML has matured with the introduction of standards for constraining and typing XML data and efficiently querying XML content. Several XML-based standards have also been introduced to allow efficient and convenient data sharing across platforms.

With SQL Server 2008, Microsoft has taken a different approach to the concept of the "database." Rather than confining themselves to a basic relational model, Microsoft has taken the stance that your data should be accessible and manageable in all its glorious (and not so glorious) forms. While relational data remains king, XML is the cornerstone of disparate data integration.

This book provides detailed information on XML from the SQL Server 2008 perspective. I'll discuss several aspects of SQL Server 2008 XML, including the XML functionality built right into SQL Server, as well as how to access SQL Server functionality that is not built into T-SQL. I'll even look at client-side XML technologies that are important to SQL Server/XML–based application design. In all, I hope you find this book enjoyable and useful in designing and implementing your own SQL Server and XML-based applications.

Who This Book Is For

This book is written for SQL Server developers by a SQL Server developer. It's written for anyone who wants to know how to retrieve relational data in XML format, shred XML data back to relational format, use XML Schema to strongly type XML data, use XQuery to query XML data, or perform dozens of other SQL Server XML tasks.

In order to take advantage of SQL Server 2008's XML functionality, you will need a basic understanding of T-SQL. Some of the code samples and concepts in the book utilize the SQL Common Language Runtime (SQLCLR), and some are presented as .NET client code. An understanding of the C# language and the Microsoft .NET Framework is useful, though not required, to utilize these samples.

How This Book Is Structured

Pro SQL Server 2008 XML is written as a guide to SQL Server's built-in XML functionality and tools. This book is written for two types of readers:

- The developer who is not familiar with prior SQL Server XML functionality will get the most out of reading the book cover to cover.
- The developer who is familiar with prior implementations of SQL Server XML functionality may get more out of using the book as a reference guide to new features and specific options.

Each chapter of the book addresses a different XML topic, making it easy to locate specific information if you are using it as a reference guide. In each chapter, I've also attempted to build on concepts introduced in prior chapters so that reading the book from start to finish will prove an engaging exercise. Following are brief summaries of each chapter in the book.

Chapter 1

Chapter 1 provides a brief overview of the history of XML, including an overview of the W3C XML recommendation. It's designed to answer the question "What is XML?" Readers who already have knowledge of XML can skip this chapter.

Chapter 2

Chapter 2 discusses the `FOR XML` clause and other legacy SQL Server XML support, such as the `OPENXML` rowset provider. Readers who are well-versed in `FOR XML` may wish to skip this chapter. For those who are already familiar with legacy XML support, this chapter will prove a valuable reference for features like the `FOR XML PATH` clause.

Chapter 3

Chapter 3 serves as the introduction to the SQL Server `xml` data type. This chapter is particularly useful for those readers who have not yet used the `xml` data type in the prior version of SQL Server. In this chapter, I lay the foundation for the discussion of SQL Server 2008's XML functionality, since most of this functionality centers around the `xml` data type. Those readers who are already familiar with the `xml` data type will still want to at least skim this chapter, as I discuss SQL Server's internal management of XML data.

Chapter 4

Chapter 4 builds on the discussion of the `xml` data type in Chapter 3, with an in-depth discussion of XML Schema collections and typed XML. This chapter also compares the SQL Server XML Schema implementation to the W3C standard and describes how to implement your own custom XML schemas. This chapter is a must-read for anyone who wants to use XML Schema to constrain the structure and content of their XML data.

Chapter 5

Chapter 5 builds on the `xml` data type discussion that began in Chapter 3. The `xml` data type methods that allow you to query XML data type instances all rely on XQuery expressions. In this chapter, I will look at the SQL Server implementation of the W3C XQuery recommendation, with a thorough discussion of available XQuery expressions, predicates, and the XQuery/XPath Data Model (XDM). This chapter is a must if you plan to query XML or shred XML data into relational format.

Chapter 6

Chapter 6 continues the discussion of XQuery begun in Chapter 5 by introducing the SQL Server–supported XQuery functions and operators, as well as the SQL Server XML Data Manipulation Language (XML DML) extensions to the standard. As always, the W3C reference is used as the basis for the discussion of SQL Server features. If you plan to manipulate XML data with XML DML or if your intent is to write XQuery expressions that require calculations and functions, you should read both Chapters 5 and 6.

Chapter 7

Chapter 7 provides a thorough discussion of XML indexing in SQL Server. In this chapter, I discuss primary and secondary XML indexes as well as XML full-text indexing. If you plan to store large amounts of XML data as `xml` data within SQL Server, this chapter will teach you how to optimize XQuery performance over your XML data.

Chapter 8

Chapter 8 begins the exploration of extending SQL Server's XML functionality via the SQLCLR. In this chapter, you will learn how to access a major piece of XML functionality that is not available directly to the `xml` data type—Extensible Stylesheet Language Transformations (XSLT). I also reference the XSLT standard heavily so that you will be able to create your own basic XSL transformations after reading this chapter.

Chapter 9

Chapter 9 discusses SQL Server 2008 HTTP Simple Object Access Protocol (SOAP) endpoints, an XML-based technology that allows you to expose server-side stored procedures and user-defined functions as web service methods. By using HTTP SOAP endpoints, your client applications/web service consumers can access predefined functionality in your database remotely and securely. If you plan on performing cross-platform or web development with a SQL Server back end, this chapter is definitely worth reading.

Chapter 10

Chapter 10 goes into greater depth discussing .NET XML support. With the .NET XML functionality, you can create SQLCLR routines that access remote XML data and services, perform legacy DTD and XML Data-Reduced (XDR) schema validations, and perform client-side manipulations of your SQL Server XML data. If you plan to write SQLCLR XML manipulation and access

routines or you plan to write client-side code that accesses XML stored on SQL Server, then this chapter is for you.

Chapter 11

Chapter 11 discusses Geography Markup Language (GML) support built into SQL Server 2008's new spatial data types, `geography` and `geometry`. This chapter is an interesting diversion for those interested in the new spatial data types and an absolute must-read for those interested in sharing spatial data in XML format.

Chapter 12

Chapter 12 provides a detailed discussion of the functionality available through the COM-based SQLXML API. SQLXML provides access to an array of functions like XML bulk load, updategrams and diffgrams, client-side `FOR XML` formatting, and XSLT-style querying of relational data. Those who support legacy applications that use this functionality should definitely read this chapter, as should those who are interested in SQLXML's functionality.

Chapter 13

Chapter 13 is an overview and discussion of the new .NET Language-Integrated Query (LINQ to XML). LINQ to XML is a powerful method of querying XML data using a SQL-like syntax directly from C# or Visual Basic. LINQ to XML is definitely an exciting new technology, and I'd recommend that anyone interested in client-side XML development should read this chapter.

Chapter 14

Chapter 14 rounds out the main content of the book with summaries and examples of common XML applications that are useful to SQL Server developers. In this chapter, I discuss SQL Server XML support tools for bulk loading flat files, optimizing SQL queries, and managing SQL Server Analysis Services (SSAS). I also look at support applications that help developers create, edit, and test XML data, like Altova® XMLSpy® XML editor software.

Appendix A

Appendix A is a list of W3C recommendations and other standards that I referenced throughout this book. This short list is useful if you need to quickly locate an XML-specific standard.

Appendix B

Appendix B describes the SQL Server implementation of the W3C XQuery/XPath Data Model and all the built-in data types available to XML Schema and XQuery. This appendix is designed to act as a quick reference, useful if you are creating XML Schema collections or writing XQuery expressions against typed `xml` instances.

Appendix C

Appendix C is a reference to the XML elements that compose an XML schema. Descriptions of all SQL Server–supported XML schema elements are provided, with their available attributes and default values.

Appendix D

Appendix D is a quick reference to XQuery, XPath, and XML DML, as implemented in SQL Server. This appendix is designed to be referenced when you are writing XQuery expressions or XPath location paths for use with the `FOR XML PATH` clause.

Appendix E

Appendix E is a quick reference to XSLT 1.0 and XPath 1.0, as implemented by the .NET Framework. This appendix is designed for use when you are creating custom XSLT stylesheets.

Appendix F

Appendix F is a glossary of the terminology used throughout the book. For those new to XML, many of the terms may be new. Likewise, you may want to follow up and get more information about a specific term than was provided in the inline description. The glossary provided me an opportunity to further expand on many terms, allowing me to add more information than could easily fit in the inline text.

Appendix G

Appendix G rounds out the book with a sampling of selected T-SQL and .NET code listings, chosen specifically to demonstrate key concepts. Where possible, I have added additional descriptive text to the code listings to further explain the options and settings selected, as well as various design decisions.

Conventions

To help make reading *Pro SQL Server 2008 XML* a more enjoyable experience, and to help you get as much out of it as possible, I've used standardized formatting conventions throughout the book.

C# code is shown in a special code font. Note that C# code is case sensitive.

```
while (i < 10)
```

T-SQL source code is shown in code font, with keywords capitalized. Note that data types in the T-SQL code are lowercased to help improve readability.

```
DECLARE @x xml;
```

XML code is shown in code font with attribute and element content in boldface for readability. Note that some code samples and results have been reformatted in the book for easier reading. XML ignores white space so the significant content of the XML has not been altered.

```
<book publisher = "Apress">Pro SQL Server 2008 XML</book>
```

Note Notes, tips, and warnings are displayed in a special font with solid bars placed over and under the content.

SIDEBARS

Sidebars include additional information relevant to the current discussion and other interesting facts. Sidebars are shown on a gray background.

Prerequisites

To make the most of this book, you should have access to SQL Server 2008 and SQL Server Management Studio (SSMS). Alternatively you can use the SQLCMD utility to execute code on SQL Server. You should also download and install the SQL Server 2008 AdventureWorks 2008 database from `http://www.codeplex.com`. All code samples in this book are designed to run on the AdventureWorks 2008 database, unless otherwise stated.

To run sample client applications and to compile and deploy SQLCLR samples, you will need C# 2005 or C# 2008. Note that the Express Editions will compile and run the client code samples, but they will not compile or deploy the SQLCLR source code. For the best overall experience, I highly recommend compiling, deploying, and executing C# code samples from within Visual Studio 2005 or the 2008 IDE.

Downloading the Code

This book contains several code samples demonstrating the concepts presented. All of these code samples are available for download in one ZIP file from the Source Code/Downloads section of the Apress web site. To get the download, go to `www.apress.com`, click on the Quick Links option on the menu, and then click on Source Code/Downloads.

Contacting the Author

The author and the Apress team have made every effort to ensure that this book is free from errors and defects. Unfortunately, the occasional error does slip past us, despite our best efforts. In the event that you find an error in the book, please let us know! You can submit errors to Apress by visiting `www.apress.com`, locating the book page for this book, and clicking Submit Errata. Alternatively, feel free to drop a line directly to the author at `michaelco@optonline.net`.