



## Foreword

---

*by Roger Wolter*

AS I SAID IN THE foreword to *A First Look at SQL Server 2005 for Developers*, I have been a Bob Beauchemin fan for the past seven years. Bob is one of the few people I know who not only knows a lot about database programming, but also is very good at teaching other people to be great database programmers. Bob wrote the book on ADO programming and has many years of experience teaching database programming classes and giving presentations at conferences and seminars.

In addition to reviewing several chapters of this book multiple times as they were changed to keep pace with changes to the beta content, I sat in on several classes that Bob and Dan taught about SQL Server 2005 programming. They both have a unique talent for simplifying complex topics and explaining them in a way that is relevant to developers. This talent is obvious in the content of this book, and I highly recommend it to anyone who wants to learn how to take advantage of the new features of SQL Server 2005 in his or her applications.

About three years ago, I sat for a couple of hours at a Starbucks in Seattle with Bob to talk about the Service Broker chapter of this book. In those three years, this book has gone through countless reviews and rewrites, until now, it's the most authoritative book about SQL Server 2005 programming available. Many people on the SQL Server development team have provided input and reviewed many of the chapters several times. I have

been involved in various ways throughout this process, so I'm very excited to see the RTM version of this book reach completion.

The thing that makes this book truly unique is that Bob and Dan taught SQL Server 2005 programming to several hundred developers in the Microsoft Ascend and TAP programs while they were writing the book, so the content of the book is heavily influenced by what real developers found useful and the questions they asked. There's no better way to learn a new technology than trying to explain it to someone else, so after teaching this material many times all over the world, Dan and Bob have a deep understanding of what developers need to know and how best to explain it to them. This is evident in both the clear, lucid text and the excellent sample code included in this book. Examples are the key elements in any programming book, and the examples in this book are exceptionally clear and relevant.

SQL Server 2005 is the most feature-rich version of SQL Server ever released, and many of the most significant features were developer oriented, so a book on the developer features of SQL Server 2005 is critical. Dan and Bob do an excellent job of explaining the most significant new developer-centric features and how to use them to improve your database applications. I was heavily involved in the Service Broker chapter, so naturally I think it's the best one, but the sections on XML and CLR integration are also very good. These three features together make SQL Server 2005 a serious contender as an application platform for data-intensive applications. Understanding these new technologies is a key part of being a successful SQL Server 2005 developer, and this book is an excellent place to start learning.

I wish you luck as you embark on your journey into the next generation of database development and hope that you enjoy this book as much as I have.

*Roger Wolter  
Solutions Architect  
SQL Server Team  
Microsoft Corporation  
Redmond, Washington  
January 2006*



## Foreword

---

*by Gert E. R. Drapers*

THE SQL SERVER landscape has changed dramatically over the years. When I started working on SQL Server in 1988, while still at Ashton-Tate, all we had were the base RDBMS-level functionality inherited from Sybase, no real extensibility, some arcane tools like ISQL and BCP, and a single database access API named DB-Library. You had to write your applications in C, squeeze all memory out of your MS-DOS based client to be able to load the network stack, and enable named pipes so you could communicate with the Intel 80286-based PC running OS/2 with LAN Manager and the Ashton-Tate/Microsoft SQL Server 1.0. Twenty-two TPC-B transactions per minute later, we proved that you could build client-server-based database applications on top of a PC platform. In reality, it would take many more years before the environment was ready to host real customer applications.

In 1996, I moved to Redmond, Washington, to join the SQL Server development team. As Lead Software Design Engineer in Test for the Storage Engine, I was responsible for tasks that included the validation of the on-disk conversion from 2K to 8K pages. The new page format was just one of many architectural changes that drove the SQL Server 7 release. Changes ranged from row-level locking, which triggered a different logging and recovery scheme, to the brand-new query processor. These were just a few of the key aspects that laid the foundation for SQL Server as we know it today.

While working on SQL Server 7, we were very cautious not to make too big a leap between SQL Server 6.x and SQL Server 7. We tried to keep

the DBA and developer experience as stable as possible. Granted, data access APIs came and went—DAO, RDO, ADO, OLE-DB, ODBC, and ADO.NET—but from a server perspective, we kept the changes to a minimum. The amount of new syntax in the form of Data Definition Language (DDL) and Data Manipulation Language (DML) and the number of new data types were pretty limited.

SQL Server 2000 introduced limited XML support, which for many set-based purists was a real shock or maybe even an insult, depending on their point of view. This was just the tip of the iceberg waiting to be revealed by the introduction of SQL Server 2005, however. The changes between SQL Server 6.x and 7 or 2000 were small compared with the huge new feature set of SQL Server 2005.

SQL Server 2005 expands in almost every possible dimension: server-side extensibility through the SQL-CLR integration; support for queue-based and message-based constructs and operations in the form of Service Broker; first-class XML integration as a storage object, but also through query extensions in the form of XQuery; and new T-SQL constructs like exception handling, common table expressions, output clauses, and new data types. On the client programming side, the list is analogous, because a lot of the new server-side features have corresponding supporting infrastructure to make them accessible to the developer.

All this new, changed, and enhanced functionality is great but yields one very important question: Which technology do we use, when, where, how, and why? What is the appropriate use of SQL-CLR, XML, Service Broker, Web Service endpoints, and so on? These and many more questions are the ones you will—or should—ask yourself when designing and/or implementing SQL Server 2005-based applications.

This is exactly why this book will be playing such a crucial role in building new systems that leverage SQL Server 2005. Not only does it provide you the overview that every (database) developer and even DBA should understand on how to leverage the new developer-oriented features and functionality, but it also provides you much-needed guidance in decisions like T-SQL versus SQL-CLR usage, how to leverage message-based processing, and how best to use XML inside the server.

Most important, this book has already proved itself. When this book first appeared during the beta stages of the project, it quickly became the de facto standard for developers who needed to educate themselves on SQL Server 2005. Bob and Dan have done a great job of updating the book—incorporating user feedback and reflecting the many changes between the first book and the final release of the product. The many samples do a great job of explaining the usage models and hidden aspects behind the features. Last but not least, many colleagues at Microsoft helped by reviewing the content to ensure its correctness and completeness so that you can use it as *the* guideline for designing and implementing your SQL Server–based systems.

Bottom line, this truly is *A Developer's Guide to SQL Server 2005*, providing you the knowledge, examples, and use cases to make the SQL Server 2005 implementation really happen!

*Gert E. R. Drapers  
Development Manager  
Visual Studio Team System  
Microsoft Corporation  
Redmond, Washington  
January 2006*