


Thore Thomassen  
Senior Enterprise Architect  
The Storebrand Group

# XML IN FOCUS

AP PHOTO/SARA JOHANNESSEN



DB2 9's "pureXML" technology is speeding development for early customers, including financial-services giant Storebrand. Explore the developer-friendly features behind the radical improvements.

Since IBM introduced object-relational technology with DB2 Universal Database 5.0, Internet technology, distributed computing, and, most recently, Extensible Markup Language (XML) have exerted a major influence on computing. XML turns a spotlight on document-centric computing, new standard formats for office documents, and SQL/XML:2003, the successor to the SQL standard.

Content management and Web-facing applications often involve storing and retrieving XML data. XML provides the underpinnings for data integration, process integration, and enterprise information integration. XML also provides enabling technology for a new distributed computing model that includes Web services, grid services, and service-oriented architectures (SOA).

DB2 9's ability to process both XML and SQL is a substantial benefit. It enables the use of a single database platform for data processing, document processing, and SOA. To someone grounded in SQL and tabular structures, XML opens the door to a structured document mindset and new query technology.

#### pureXML IN ACTION

IBM provided a test drive of DB2 9 to enable early adopters to test their software with the new DB2 SQL/XML platform. In the technology adoption cycle, early adopters are typically adept organizations and independent software vendors (ISVs) developing tools and middleware. Some of these early adopters shared their experiences with the new technology.

One such company is The Storebrand Group, a Norway-based financial services company that offers pension plans, life and health insurance, banking, and asset management. In Norway, pension plans are mandatory; 1,400 companies have entered into pension agreements with Storebrand. In 1996, Storebrand adopted a service-oriented architecture using the Standard Generalized Markup Language (SGML) to format messages. It uses DB2 to store messages for synchronization, audit, and other purposes. Prior to DB2 9, Storebrand decomposed (shredded) the XML data or stored complete documents as CLOB columns. Moving to DB2 9's pureXML technology, which stores XML documents in their native format, provided several advantages.

Storebrand's senior enterprise architect Thore Thomassen notes that storing XML natively reduces the I/O code in applications and requires fewer database skills than the shredding or CLOB method. DB2 9's XML type also provides more flexibility for searching and manipulation. "We have several cases where we know we have business-critical data in XML format in a CLOB, but can't access it without writing an application to do so," Thomassen explains.

The XML Query Language (XQuery) from the World Wide Web Consortium (W3C) supports querying documents based on content and structure. To locate parts of documents, queries involve logic for traversing or matching a pattern in a tree structure. Maintaining the order of rows is not integral to an SQL INSERT query, but preserving the order of tags is essential when storing an XML document in its native format. Preserving document structure makes it possible to use locations in documents as search criteria.

Thomassen says the ability to use XPath and XQuery simplifies application development and provides greater transparency and reuse within pureXML applications. His team has found the ability to mix XML and SQL queries "very powerful."

---

BY KEN NORTH

OpenLink Software, an ISV specializing in middleware, also tested DB2 9. The company's CEO, Kingsley Idehen, explains the value this way: "Native XML data type support in DB2 provides a major contribution to the emerging standardization of SQL and XML integration at the data access middleware layer across ODBC, JDBC, and ADO.Net."

DB2 has supported XML since 1999, so what's so revolutionary about DB2 9?

### **pureXML AND THE DB2 XML EXTENDER**

IBM's early XML support came in the form of the DB2 XML Extender, which treated XML as *xcolumns* or *xcollections*. Next came stored procedures and functions for processing XML in message queues. DB2 9 introduces a new XML storage model.

DB2 XML Extender treats XML as a CLOB or a variable-length character string (VARCHAR), but DB2 9 implements XML as a first-class data type. This means you can use the XML type in Data Definition Language (DDL) statements, stored procedures, and functions, including SQL/XML:2003 XML publishing functions. Although IBM will support DB2 XML Extender going forward, its capabilities are superseded by the pureXML support in DB2 9.

### **pureXML ARCHITECTURE**

A common approach to integrating XML into an SQL platform is to support queries over XML by mapping to relational algebra. This approach uses the

existing relational engine, which DB2 XML Extender has done since DB2 UDB 6.1. In DB2 9, a single engine (optimized for both XML and relational data) processes relational and XML (hierarchical) data; however, the two data types reside in separate storage layers. The new engine treats an XML document as a parsed, annotated tree structure and supports indexing parts of documents.

Hand-in-hand with the new XML data store, DB2 9 supports the SQL/XML:2003 XML type, SQL/XML functions, and XQuery. DB2 9 lets you query XML data using XQuery alone, SQL alone, XQuery that invokes SQL, and SQL/XML functions that execute XQuery expressions.

### **INDEXING**

DB2 9 provides flexible capabilities for indexing XML data. An XML index can be defined on any path or subtree of the document in an XML column by specifying an XML pattern. This approach provides fine control over which data is indexed and which isn't. DB2's XML indexes map path and values to document and node instances.

Alternatively, DB2's Net Search Extender provides indexes for full-text searches by mapping text tokens to the documents that contain them. Full-text indexing and full-text search can be applied to full or partial documents.

### **ENCODING AND SERIALIZATION**

In the DB2 world, it's not uncommon to have a mismatch of character sets for mainframes (EBCDIC), Linux, Unix, and Windows (ASCII), and XML, Java, and .Net applications (Unicode). But Unicode is popping up everywhere in DB2. Since version 8.1, DB2 uses Unicode for catalog information, SQL parsing, and string comparisons for queries involving multiple character sets. Also, the DB2 pre-compiler converts embedded SQL program source code to Unicode before compilation.

When transferring data between machines, DB2 must often do character

set conversions to put the data in a usable form for a receiving machine. The Coded Character Set Identifier (CCSID) definitions are unique values that identify a code page. They provide bidirectional layout transformations in DB2 for Linux, Unix, and Windows. When two machines have different code pages or CCSIDs, DB2 converts input and output character data, such as from a DB2 Connect server code page to a host CCSID and vice versa. The receiving machine performs the conversion on incoming data.

When serializing XML, DB2 may add an encoding declaration at the beginning of the XML output. When storing application variables or encoded XML types in an XML column, the DB2 database manager will check for an encoding declaration and map it to a CCSID.

Support for Unicode and CCSIDs also includes changes to DB2 client/server connectivity and programming APIs, such as the call-level interface (CLI) and embedded SQL. Embedded SQL programs with dynamic SQL use the SQL descriptor area (SQLDA) for host variables. To support XML data, embedded SQL programs must update the *sqlname* field of the SQLVAR to indicate that a base type contains XML data.

### **ANNOTATED XML SCHEMAS**

When executing a CREATE DATABASE command, users can choose whether to include XML support in a DB2 database. If you choose to include database objects, DB2 creates a database with the Unicode character set. Exercising the XML option precludes partitioning a database. A second decision when generating a database is whether to use system-managed space or database-managed space. The best choice, whether for relational only or XML databases, is database-managed space.

In addition to a new XML type for SQL access, DB2 9 includes W3C XML Schema types. The XML Schema Language has more primitive types than SQL and it supports complex types by deriva-

**!** **FROM THE DEVELOPER**

Get the DB2 9 story straight from the source. Go to [DB2mag.com](http://DB2mag.com) for an in-depth Q&A with Anjul Bhambhri, senior development manager for DB2 XML.

tion. Data integrity and type safety are a byproduct of intelligent use of SQL Data Definition Language (DDL). The XML Schema Language provides a similar benefit for XML applications and XQuery, although XQuery can also operate on documents without a schema.

Documents with multiple schemas, including versions of the same schema, can coexist in a single XML column. An XML document can include multiple types and can be validated using XML schemas. DB2 9 provides registration of schemas for validating documents. It maintains an XML schema repository (XSR) in the database catalog, consisting of catalog tables, views, and stored procedures.

When validating schemas, the XML schema registration engine produces type annotations. Therefore, XML-enabled DB2 databases include documents and annotated XML schemas for use during the execution of queries. Using annotated schema decomposition, it's possible to convert XML to relational data. DB2 uses the annotations to map elements and attributes from an XML document to target database tables. To assist with annotated schema decomposition, DB2 9 provides six new stored procedures (xdb-DecompXMLxxxx).

Creating an XML-enabled database and registering XML schemas can impact application memory requirements. Developers may need to increase the application heap (APPLHEAPSZ) for large schemas.

## SQL DDL, EXTERNAL ROUTINES, UDFs, STORED PROCEDURES, AND TRIGGERS

Because DB2 9 implements XML, you can use it in SQL DDL statements, stored procedures, and functions, including SQL/XML functions. DB2 stores the value of an XML column in an internal format but the XMLSERIALIZE function can convert it to a string value.

You can create triggers on XML columns and use the XML type with external functions and stored procedures, although you can't reference the

## DATA PAGE

**Company:** The Storebrand Group

**Inception:** 1767

**Primary businesses:** Life insurance, asset management, and banking

**Q1, 2006 group profits:** NOK 460 million

**Key DB2 9 Results:**

- Reduced query run time from 36 hours to 10 minutes or less
- Shrank programming search processes from up to eight hours down to 30 minutes
- Accomplished XML schema changes in response to business change in five minutes compared to one week with the previous shredding method
- Reduced I/O code in services by 65 percent

Read the Storebrand case study online at [ibm.com/software/success/cssdb.nsf/CS/HSAZ-6PW3MW?OpenDocument&Site=db2software](http://ibm.com/software/success/cssdb.nsf/CS/HSAZ-6PW3MW?OpenDocument&Site=db2software).

before or after value of the XML column in the trigger. To use the XML type with stored procedures, you include parameters of type XML in CREATE PROCEDURE parameter signatures. The process is similar for a scalar or table function using CREATE FUNCTION, although the XML type is not valid for an external OLE DB function. DB2 implicitly parses and serializes XML used as parameters passed to external routines. You can also use an XMLPARSE function to explicitly parse XML data.

You can use the XML type with programs written in C, C++, C#, Cobol, Java, and Visual Basic (VB). C and C++ programmers can access the XML type with embedded SQL or CLI; C# and VB programmers, with the DB2 .Net Data Provider; and Java programmers, with JDBC or SQLJ. If you're writing external routines used by stored procedures and functions, you use the XML type in the same way you use a CLOB. You declare external routine parameters to be type XML and use CREATE PROCEDURE and CREATE FUNCTION statements to register them with the database. They specify that the XML type is stored as a CLOB and DB2 serializes the XML before passing it to an external routine.

## TYPES AND DATA ACCESS

C/C++ and scripting solutions such as PHP and Perl often use an ODBC interface to SQL data. The DB2 CLI extends the ODBC 3.51 API with support for additional types, including the new XML type.

When writing CLI programs, the symbolic SQL data type for XML columns is SQL\_XML and the default symbolic C data type is SQL\_C\_BINARY. Permissible conversions include SQL\_C\_CHAR, SQL\_C\_WCHAR, and SQL\_C\_DBCHAR. The default precision for XML

columns is 0, but it's a defined length for external routines.

IBM is deprecating the type 2 JDBC driver for DB2 9. It suggests moving to the IBM DB2 Driver for JDBC and SQLJ, a single driver that combines type 2 and type 4 features. It supports the Java Transaction Service (JTS), Java Transaction API (JTA), JDBC 3.0 API, and SQLJ capabilities that have JDBC counterparts.

To access DB2 from Java, developers can use static SQL with SQLJ and dynamic SQL with JDBC. Java does not have an XML type, so invoking the JDBC GetTypeInfo metadata method will report the type of an XML column as java.sql.Types.OTHER.

Using JDBC ResultSet.getXXX methods, you can retrieve an entire XML column or a sequence from the column. When you retrieve XML data using a JDBC getXXX method, it's in a serialized string format. The getXXX method will output the result in the format corresponding to the method name (for example, getAsciiStream). In the case of ResultSet.getObject, you can cast the object to the DB2Xml type, assign it to a DB2Xml object and use a DB2Xml.getDB2XXX method to output the data to the desired type.

An enhanced set of add-ins for

Microsoft Visual Studio 2005 include support for building and testing .Net applications using XML data stored in DB2 9. DB2 includes providers for OLE DB and ADO.Net, including the DB2 .Net Data Provider, the OLE DB .Net Data Provider and the DB2 Data Provider for .Net Framework 2.0. The OLE DB Provider and OLE DB .Net Data Provider don't support the new XML type. To access an XML column with use the ODBC .Net Data Provider, you map it to SQL\_LONGVARCHAR.

### QUERYING WITH SQL AND XQUERY

As Storebrand's Thomassen mentioned, DB2 9 provides the ability to use SQL, XQuery, or a combination of the two. A detailed comparison of SQL and XQuery isn't a trivial exercise, so you can imagine the complexity of integrating the two languages. To understand XQuery, you must also grasp concepts defined in a variety of specifications (recommendations) from the W3C, including namespaces, schemas, and XPath expressions.

SQL and XQuery have similarities, such as join operations and user-defined functions, but there are essential differences. XQuery involves two-valued logic but the possibility of a null value puts SQL in the three-valued logic arena. XQuery names are more complex than SQL, and XML schemas can evolve. XML schemas provide type checking, but XQuery can operate on documents without a schema. Update capabilities and full-text searching are a work-in-process for a future version of the XQuery specification.

DB2 supports XQuery and SQL with separate parsers and there is no translation from XQuery to SQL. There is a common query compiler, and queries from either language produce an execution plan using a query graph to model queries. The DB2 cost-based optimizer can use both XML and relational indexes when generating an execution plan.

The SQL/XML standard introduced a group of functions (XMLQUERY, XML EXISTS, and XMLTABLE) that operate on instances of XML data. DB2 9 also uses these SQL/XML functions: XMLParse, XMLValidate, XML Serialize, XMLCast. The SQL/XML standard introduced functions for converting relational data to XML. The publishing functions in DB2 9 include XMLElement, XMLAttributes, XMLForest, and XMLAgg.

### WEB SERVICES

Since it unveiled Web services technology in 2000, IBM has participated in developing specifications and software solutions to spur SOA adoption. Web services and grid services

exchange XML-encoded messages between providers and consumers, using SOAP. Web services and grid services use an XML vocabulary known as the Web Services Description Language (WSDL) to describe services. DB2 includes a Web service provider and it can consume Web services.

Because XML is integral to the operation of Web services and grid services, the overhead of XML processing can incur a performance penalty. DB2's native XML engine, parsed documents, and annotated schemas will diminish that overhead by

reducing the amount of parsing and validation required to exchange XML-encoded messages and to query them on disk and in message queues.

DB2 includes an embedded application server to run the Web services packaged with the product. IBM has also developed software, the Web Object Runtime Framework (WORF), which enables the use of DB2 as a Web services provider. WORF can generate the WSDL for a service and it supports accessing the service with HTTP GET, POST, and SOAP bindings.

WORF supports multiple modes for retrieving data from DB2 databases. WORF users create a Document Access Definition Extension (DADX) file, an XML file describing mappings between XML document nodes and relational data. The DADX file supports SQL operations and operations with XML collections. If you don't use WORF, you'll have to write code to create a provider, including generating the WSDL service descriptions.

### THE LEARNING CURVE

Developers and DBAs adopting DB2 9 will experience new functionality that affects data access APIs, document queries, SQL grammar, query optimization and so on. There will be a learning curve, but the added capabilities provide power, flexibility, and ease of development for SOA, Web services, grid services, and document management applications. ■

### Ken North

is a consultant, industry analyst, and editor of [www.SQLSummit.com](http://www.SQLSummit.com). He teaches Expert Series seminars, frequently speaks at conferences, and has chaired the XML DevCon 200x conferences.

### RESOURCES

#### DB2 9

[ibm.com/db2/v9](http://ibm.com/db2/v9)

#### The Storebrand Group Case Study

[ibm.com/software/success/cssdb.nsf/CS/HS AZ-6PW3MW?OpenDocument&Site=db2software](http://ibm.com/software/success/cssdb.nsf/CS/HS AZ-6PW3MW?OpenDocument&Site=db2software)

### BY THE NUMBERS

# 5

Number of years of development behind DB2 9

# 8

Factor of performance improvement using DB2 9 rather than the DB2 XML Extender, according to tests conducted by Dr. Andreas Birkendorf of Douglas Informatik & Service GmbH

# 261

Number of topics in the DB2 9 online forum (as of July 10) at [ibm.com/db2](http://ibm.com/db2)

# 750

Number of IBM developers who contributed to DB2 9