Improving the derivatives trading process with DB2 pureXML™

Cynthia M. Saracco
June 2007
Executive Summary

Trading in over-the-counter (OTC) and exchange-based derivatives has grown tremendously during the past several years and shows little sign of abating. As Fig. 1 illustrates, the notional amounts for all OTC derivatives reached $415 trillion in December 2006 – up from $285 trillion in December 2005 and $111 trillion in December 2001.\(^1\) Similarly, the number of exchange-traded futures and options contracts reached nearly 11.9 billion in 2006, up from 10 billion in 2005.\(^2\) A variety of factors are fueling this rapid growth, including market volatility, an increased focus on managing risk, and increased activity from hedge funds, banks, insurance firms, and other parties.

![Outstanding notional value of OTC derivatives worldwide](image)

$ = USD trillion

Fig. 1: Growth of notional value of OTC derivatives from 2001 – 2006

While the growth in derivatives trading offers financial institutions significant opportunities to increase revenue, attract new clients, and improve their competitive position, capitalizing on these opportunities is far from simple. The complex nature of derivatives, coupled with the rapid increase in traded volumes, has led to lengthy confirmation cycles and greater risk, which regulatory bodies are pressuring firms to resolve. Many forward-thinking firms are turning to sophisticated software technologies to help them address these challenges. Frequently, firms use the eXtensible Markup Language (XML) to represent and exchange derivatives data electronically to improve trading cycle automation. Consequently, efficient storage and management of XML data is emerging as a new technology imperative.

---

\(^1\) OTC derivatives market activity reports issued twice yearly by the Bank for International Settlements.  
This paper explores the unique business characteristics of derivatives trading applications, explains their impact on traditional information management systems, and describes how pureXML™ database management technology available in IBM’s DB2 addresses the challenging XML data processing and performance requirements associated with trading derivatives. For those unfamiliar with derivatives, Appendix A provides a very brief introduction.

**Business requirements**

Financial institutions seeking to serve as effective buyers, sellers, or custodians of derivatives need to address several business challenges, including the need to

- manage risk associated with various market and business conditions,
- comply with industry regulations,
- reduce or eliminate manual processing at various stages of the trade,
- cope with rapidly growing trade volumes,
- accommodate the diverse and changing nature of derivatives.

Failure to adequately address these requirements can lead to greater operational risk, legal penalties, and unnecessary expenses.

**Understanding the stages of an OTC derivatives transaction**

An OTC derivatives trade involves multiple stages, each of which requires specific activities to be performed. It’s worth noting that the derivatives trading lifecycle is considerably longer and more complicated than the lifecycle associated with simple equity trades. Indeed, trading derivatives requires creating and maintaining complex legal contracts that may evolve over months or years. This leads to a number of information management challenges, which we’ll discuss shortly.

Table 1 outlines several of the activities that occur throughout the lifecycle of a derivatives trade, which includes pre-trade, trade, and post-trade stages.

<table>
<thead>
<tr>
<th>Stage</th>
<th>Activity</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Pre-trade</td>
<td>Bilateral documentation and internal approvals</td>
<td>Overall parameters of trading activities are established through a bilateral master agreement. Counterparty credit reviews are conducted to establish credit lines and trading limits.</td>
</tr>
<tr>
<td>2. Trade</td>
<td>Trade execution</td>
<td>Parties agree on terms via phone, fax, and/or electronic means.</td>
</tr>
<tr>
<td>3. Post-trade</td>
<td>Trade capture</td>
<td>Trade details are captured for processing and risk management. This may be manual (via trade tickets) or electronic.</td>
</tr>
<tr>
<td></td>
<td>Trade verification</td>
<td>Counterparties may opt to verify key economic details of the trade.</td>
</tr>
</tbody>
</table>
Stage Activity Definition

<table>
<thead>
<tr>
<th>Stage</th>
<th>Activity</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Trade affirmation or matching</td>
<td>Trade details may be provided by one party and affirmed by the other, or each party may exchange records for matching.</td>
</tr>
<tr>
<td></td>
<td>Confirmation</td>
<td>Final confirmation of the trade details are secured and exchanged. Confirmations may be paper- or electronic-based.</td>
</tr>
<tr>
<td></td>
<td>Settlement</td>
<td>Cash or other assets are exchanged per the terms of the contract.</td>
</tr>
</tbody>
</table>

Table 1: Summary of the OTC derivatives lifecycle

Clearly, buyers and sellers need to exchange information, and both parties may need to receive information from or provide information to other sources. For example, buyers and sellers often rely on third parties for market research, pricing information, market volatility data, and other relevant information. Furthermore, firms serving as custodians of the contract need to maintain information about terms and conditions, as well as ensure that these are properly settled. Such information must also be recorded and shared among appropriate parties.

**Automating derivatives trading**

Quite naturally, firms are striving to automate the exchange of data related to derivatives trading. Doing so can help shorten lengthy phases of the trading cycle, thereby reducing risk and improving efficiency. Automation also helps address regulatory concerns about backlogs at certain points in the trade cycle.

For example, in 2005 the UK Financial Services Authority contacted major derivatives dealers in London about the growing confirmation backlogs of credit derivatives, which the International Swaps and Derivatives Association (ISDA) found to be averaging 23 trading days. Representatives of various UK and American firms discussed how to improve the situation, and process automation emerged as a key initiative. The Depository Trust & Clearing Corporation (DTCC) began offering electronic services to minimize the inherent risk of confirming trades via fax or phone. Today, the percentage of electronically confirmed trades has doubled, significantly reducing confirmation backlogs for credit derivatives trades. However, backlogs still exist and remain at undesirably high levels for certain types of derivatives.

To promote the exchange of derivatives data and help firms automate their processing, industry consortiums have turned to XML, a standard in the technology industry developed by World Wide Web Consortium (W3C). Known for its flexibility, XML enables firms to model business data in a self-describing format. XML is widely used for various applications; for example, Web pages are written in the hypertext mark-up language (or HTML), which is one form of XML.

---

Financial institutions that belong to the ISDA have defined an XML format for OTC derivatives transactions. This format, the Financial Products Markup Language or FpML, has become an increasingly popular means for representing and exchanging many types of derivatives data. A number of software vendors now support FpML, and several industry organizations have recommended or adopted it. These include the Society for Worldwide Interbank Financial Telecommunication (SWIFT), the Asset Managers Forum (AMF), and the International Securities Associate for Institutional Trade Communication (ISITC). Of course, derivatives data can be captured in other XML formats, including those proprietary to a given software vendor or financial institution.

**Information management requirements**

As more derivatives data is being captured electronically and exchanged in XML formats, the need for effective XML data management technology is increasing. Quite simply, firms are seeking software that can efficiently store and operate on this data. Unfortunately, XML data poses several challenges for traditional database management systems (DBMSs).

**Flexibility and adaptability**

XML formats for derivatives, such as FpML, consist of complex hierarchical structures. Because derivatives contracts can vary widely, the hierarchies modeling these contracts must vary as well. For example, an equity derivatives trade represented in XML needs to contain elements for the ticker symbol, share price, and so on. Such information is irrelevant to a credit default swap contract, which needs to capture data about credit events, periodic payments between the buyer and seller, and so on.

The hierarchical and varying nature of XML formats that model derivatives presents challenges for many DBMSs. Often, only two fundamental modeling options are available: decomposing (or “shredding”) the data across numerous table columns or storing the data intact within a character or binary large object (CLOB or BLOB) column of a table. Each technique has disadvantages.

Storing the derivatives data as large objects makes searching, updating, and retrieving portions of contracts expensive because the database system doesn’t understand the internal structures of these objects. Decomposing the contracts and transforming the data into non-XML data types can involve complex, labor-intensive mappings. Resulting database schemas are often unwieldy, requiring numerous tables to capture all possible attributes of interest, including those that are important to only a small number of contracts. Such a design leads to sparsely populated tables, wasting space and creating greater administrative overhead. Writing the code necessary to retrieve (or query) information stored in numerous tables is often labor-intensive and error-prone.

Furthermore, hard-coded XML-to-table mappings are costly to change, inhibiting the flexibility required by derivatives trading applications. Over the life of a derivatives contract, terms or conditions may be added, deleted, or modified. Reflecting such changes in a traditional database design can require new tables, new indexes, new
integrity constraints, new user access privileges, and new application code. Implementing these changes can be expensive and time-consuming.

**Performance and scalability**

Larger trade volumes and increased automation require an infrastructure that offers superior runtime performance and scalability. Certain database management design features are critical. For example, because parsing XML data is an expensive operation, it’s best for systems to avoid such activity at query runtime. Furthermore, indexing technology must be tailored to the unique characteristics of XML data structures, including those that contain repeating elements or attributes. Administrators should be able to index only those portions of their XML data of interest to their applications, both to conserve disk space and to avoid unnecessary write operations for index maintenance.

**Standard interfaces and formats**

Integration with existing IT infrastructures and popular software offerings is also critical to most derivatives trading applications. As such, supported query languages, application programming interfaces (APIs), and integrated development environments (IDEs) are worth consideration. Indeed, a DBMS that leverages widely available database administration and application development skills can reduce labor costs and prevent project delays.

Furthermore, because software to support derivatives trading is evolving rapidly, a database management infrastructure must be able to accommodate a variety of XML formats for modeling derivatives. Industry consortiums – such as the ISDA – develop new versions of existing standards as business needs change. Proprietary XML formats for derivatives also remain in use and evolve over time. Thus, a derivatives trading database must be able to cope with XML data based on a variety of schemas and do so with minimal administrative overhead.

**The DB2 pureXML™ solution**

IBM’s DB2 hybrid database server, which supports both relational and pureXML™ technologies, was designed with challenging applications such as derivatives trading in mind. The product provides firms with a common interface and database management platform for traditional corporate data (modeled in tables) as well as complex derivatives trading data (modeled in XML), as shown in Fig. 2. Indeed, IBM even offers free software to help firms quickly deploy a sample derivatives trading database complete with FpML data, associated database objects, pre-built queries, and stored procedures.
Internal studies and early customer experiences have shown that DB2’s hybrid architecture can reduce labor requirements, shorten development cycles, and provide strong runtime performance. For example, Storebrand, a Norwegian financial services firm, compared DB2’s pureXML support with relational technology for various situations.\(^4\) It found that pureXML enabled them to

- Generate a report in less than 10 minutes instead of more than 1 day.
- Implement a schema change in a few minutes instead of requiring a full day to prototype and test the change.
- Reduce the I/O portions of select Web services by 65%.

IBM benchmarks also confirm that pureXML technology can provide 2- to 5-times faster performance for concurrent insert operations and up to 40 times faster performance for certain types of queries when compared with relational alternatives.

The following sections explore how DB2 pureXML addresses the information management challenges associated with derivatives trading.

**Flexibility and adaptability**

DB2 enables firms to store any well-formed XML document in a single column of type “XML.” Because of this, derivatives data represented in different XML formats, or different versions of a given XML format (such as FpML), can be written to DB2 without costly database design changes. This saves time and labor. Furthermore, administrators can register one or more XML schemas with DB2 and instruct DB2 to automatically validate XML data against such schemas before inserting or updating data.

DB2’s approach is more flexible than certain other approaches. For example, DB2 doesn’t mandate that all data in a single XML column conform to a given schema, which

is often a requirement of systems that decompose or shred XML data behind the scenes. Furthermore, DB2 supports evolving XML schemas (varied structures) by enabling different XML documents within a given column to be validated against different versions of the same schema. This minimizes administrative overhead and labor.

**Performance and scalability**

Performance and scalability were critical design aspects of DB2’s pureXML technology. To ensure superior query performance, DB2 parses XML data only once (when it is written to the database) rather than at runtime for each query (as some systems do). New indexing technology and query optimization techniques also minimize unnecessary I/O and processing overhead. In particular, DB2’s XML indexing techniques enable administrators to index only the XML nodes they wish, and DB2 fully supports indexing repeating elements (such as multiple payment calculation periods). Products that lack such options may suffer from unwieldy or inefficient indexing structures.

DB2 features a shared query processing engine for its relational and XML data, providing firms with high performance for queries that involve either type of data (or both). It automatically evaluates all queries using cost-based optimization techniques so that it can select an efficient data access strategy. Query language translation never occurs in DB2. In other words, DB2 does not transform a query written in the standard XML query language (XQuery) into the standard relational query language (SQL) before processing. This approach saves time and avoids semantic inconsistencies that can otherwise result.

Performance concerns often prompt firms to demand specific proof points, usually in the form of published benchmarks. Because of this, IBM ran comparative tests involving various database design (and storage) alternatives for DB2. While results for any given workload can vary, it’s helpful to review some baseline data. Significant results of IBM’s study\(^5\) include:

- Concurrent insert operations can be 2 to 2.5 slower with character large objects (CLOBs) than with an XML column.
- Bulk inserts or import operations involving XML columns may run 4 to 5 times faster than equivalent operations involving decomposition or “shredding.”
- Queries over complex XML data can be 40 times faster than corresponding queries over CLOB columns. (This is largely because XML data must be parsed at query runtime with CLOBs, while such work is unnecessary if the data is stored in an XML column.)
- Queries over complex XML data can be 50 to 100 times faster over XML columns than queries over “shredded” data. (This is largely due to the overhead of reconstructing the data as XML, which can require extensive tagging or conversion operations as well as expensive joins.)

---

Furthermore, separate scalability studies conducted by Intel\textsuperscript{6} and IBM\textsuperscript{7} on different hardware platforms revealed positive results for DB2’s pureXML technology. Both studies involved simulations of more than 100 concurrent users accessing FIXML data stored natively in DB2. (FIXML is the XML version of the Financial Information eXchange messaging standard for securities transactions.) Results demonstrated near-linear scalability for read-only workloads as well as high levels of scalability for read/write workloads. Throughput rates involved thousands of queries and inserts per second – significant results for the hardware platforms tested.

Although FIXML and FpML differ in many respects, they impact a database management system in similar ways. Both include numerous XML schema documents, hundreds of type definitions, and thousands of elements and attributes. Many of these elements and attributes are optional, and only a relatively small number may be present in any given FIXML or FpML message. These are some of the characteristics that make such data well-suited to DB2 pureXML technology.

Indeed, a large securities processing firm recently evaluated DB2’s pureXML capabilities for managing derivatives trading data. The firm concluded that DB2 pureXML provided certain performance benefits and enabled it to avoid creating complex, fragile mappings between its FpML data and tabular structures. It also found DB2’s interoperability between XML and SQL to be “impressive.”

**Standard interfaces and formats**

Securing skilled labor and integrating a new technology into an existing IT infrastructure can be challenging. As a result, firms often insist that purchased software provide standard interfaces. DB2 supports two industry-standard query languages: SQL and XQuery. SQL has been widely used for decades by database professionals, while XQuery (which includes XPath) is often preferred by application developers familiar with XML.

With DB2, firms can access their data using either or both of these query languages. By contrast, some systems require programmers to “wrap” their XQueries in SQL before the database system will process them. This can introduce added complexity and may present challenges for skilled XPath developers who have minimal SQL experience. While DB2 supports “hybrid” queries that embed XQueries in SQL or vice versa, it doesn’t mandate their use. Instead, developers are free to express their queries in the language – or language combination – they find most convenient and appropriate for their work.

Finally, DB2 pureXML supports familiar tools, utilities, and programming languages to help administrators and programmers become productive quickly. Administrators can work with common facilities to monitor and tune performance, import and export data, and

\textsuperscript{6} DB2 pureXML scalability on Intel Xeon MP Platforms using IBM N Series Storage, Intel paper, 2006.

back up and recover data, and so on. Programmers can access XML data stored in DB2 through Java, .Net, PHP, C, and other popular interfaces.

**Free DB2 software bundles for FpML, other financial data**
To help firms quickly exploit DB2’s pureXML technology and prototype applications tailored to their business needs, IBM has developed free industry-specific software bundles. As of this writing, free packages are available for derivatives trading data (modeled as FpML), securities transaction data (modeled as FIXML), mortgage data (modeled as MISMO), and other data formats.

Each industry-specific package contains code to create database objects that store and index XML data, register an industry-specific XML schema (such as FpML), populate the database with sample XML data that conforms to the registered schema, and create stored procedures that query the sample XML data. The source code for all scripts and stored procedures is provided for easy customization. See the “References” section for details on where to download these packages.

**Summary**
Rapid growth of derivatives trading is prompting firms to seek greater levels of automation throughout the trade cycle to keep pace with market demands, reduce risk, and respond to regulatory pressures. As a result, firms are adopting XML as the preferred format for exchanging electronic data. XML’s flexible, self-describing nature makes it well suited for coping with the diverse and complex characteristics inherent in derivatives. Unfortunately, this very flexibility poses challenges for traditional database management systems.

DB2 pureXML addresses several common problems associated with managing derivatives data. It accommodates evolving data structures with ease so administrators don’t have to redesign their database schemas when new business requirements (such as contract changes) emerge. It features specialized XML indexing, storage management, and query optimization techniques for strong runtime performance and scalability. It supports industry-standard query languages and application programming interfaces. And free, industry-specific software bundles are available to help firms get off to a quick start.

To learn more about how DB2 can help with your derivatives trading application, consult the materials cited in the “References” section or contact your IBM account representative.
About the author
Cynthia M. Saracco is a senior solutions architect at IBM’s Silicon Valley Laboratory who specializes in emerging technologies and database management topics. She has more than 20 years of software industry experience, has written 3 books and more than 50 technical papers, and holds 6 patents. Her email address is saracco@us.ibm.com.

Acknowledgments
Thanks to Suzanne Dence, Zohar Hod, Sean Linehan, Susan Malaika, Matthias Nicola, and Philip Schwartz for their help with this paper.
Appendix A: About derivatives

A derivative is a financial instrument based upon (or derived from) some asset, such as a stock or stock index (in the case of an equity derivative). Two parties agree to exchange cash or something of value based upon conditions affecting the underlying asset. Typically, one party uses the trade as a way to mitigate (or hedge against) risk; the other party uses the trade as a way to gain immediate income (through fees or premiums) and/or to speculate that future market conditions will provide profits.

Two short examples may help clarify these concepts. An equity option derivative might give Party A the option to sell 1000 shares of IBM stock to Party B at $90 per share in one year’s time. Party A might seek such a contract if he believes IBM’s share price, currently trading at $93, will drop below $90. To hedge against this, he’s willing to pay a fee for the right to sell the stock the following year for $90 per share. Party B might speculate that IBM stock will exceed $90 per share by the contract settlement date and seek to profit from the up-front fees.

A credit default swap derivative could involve two financial institutions agreeing that one institution will assume the credit risk for a loan. Perhaps Bank A loaned $10 million to an airline. The loan provides Bank A with a favorable income stream, but it also carries risk. Officials at Bank A may fear that rising fuel prices, labor unrest, and competitive pressures could force the airline to default on its loan. To hedge against this type of credit event, Bank A negotiates with Bank B to purchase protection. In essence, Bank A agrees to pay Bank B periodic fees (premiums) to insure itself against the airline’s failure to pay. If a default occurs, Bank B will pay a specified sum to Bank A.

Different types of derivatives are traded privately (over-the-counter or OTC) or through a public exchange (exchange-traded). Growth in derivatives trading is typically measured in the outstanding notional amount of such contracts, i.e., the nominal or face value used to calculate payments.

Derivatives can be divided into three general classes: futures/forwards, which involve buying or selling an asset at a future date; options, which give one party the option to sell an asset at a future date; and swaps, which enable two parties to exchange cash or other assets. The assets underlying different classes of derivatives can vary but often involve equities, interest rates, credit events, commodities, or foreign exchange rates.

It’s important to recognize that derivatives are complex financial instruments that vary considerably. Their life span often ranges from months to years, and their terms and conditions may change over time. Indeed, even the parties involved may change over the life of a derivatives contract, as one party may assign its position to another (novate).
References and free software downloads

About derivatives


Monetary and Economic Department: OTC Derivatives market activity reports, issued twice yearly by the Bank for International Settlements.


About DB2 pureXML


Saracco, C. M. and Don Chamberlin, Rav Ahuja. DB2 pureXML: Overview and fast start, IBM Redbook # SG24-7298-00, July 2006.


These books and papers, as well as other technical materials, are available from the DB2 pureXML Wiki at http://www-03.ibm.com/developerworks/wikis/display/db2xml/Home