

Oracle XML DB vs. IBM DB2 XML Extender

*An Oracle Technical White Paper
February 2003*

Oracle XML DB vs. IBM DB2 XML Extender

OVERVIEW

Oracle XML DB is a high-performance XML storage and retrieval technology available with the Oracle9i Release 2 server. It fully absorbs the W3C XML data model into the Oracle server, and provides new standard access methods for navigating and querying XML. With XML DB, you get all the advantages of relational database technology and XML technology at the same time. XML DB also adds an XML Repository to the database.

IBM's database XML technology is supplied as the DB2 v8.1 XML extender, a loosely coupled plug-in that does not afford the benefits of Oracle's native XML support. IBM also plans to introduce XPERANTO, an XML-wrapped data-integration technology in the mid-tier.

Oracle's XML DB technology is superior to IBM's in terms of

- Native support for the XML data model and other XML standards
- Integrated SQL, XML, Java and Text processing that eliminates extra moving parts and provides high scalability
- Superior functionality and richness of programmatic access
- A comprehensive XML vision, including an XML Repository to support content-oriented XML.

A recent report¹ comparing the major XML data management technologies concluded: *"IBM DB2 comes out disappointingly ... overall, it does not support quite so many features as Oracle's 9i ... Before commencing this Report, given the expertise of the people involved, an expectation was formed as to how the scoring would come out ... The one possibility that was dreaded was that there would be a tie for first place between Oracle and IBM. Not only was this not the case, but the margin of difference was so great, which came as a considerable surprise ..."*

¹ "Database Management Systems -- Managing Relational and XML Data Structures" ISBN 0-9542845-2-6 Published by Butler Direct Ltd., The Butler Group Sep 2002. <http://www.butlergroup.com/>

The table below lists the major points of difference between Oracle XML DB and the IBM DB2 XML Extender. The rest of this paper discusses the differences in more detail.

	Oracle XML DB	IBM DB2 XML Extender
Native support for the XML data model and other XML standards	Yes	No
Native XML Data Type	Yes	No
XML Schema Integrated in database	Yes	No
High-Fidelity XML	Yes	No
Comprehensive XML Standards	Yes	No
Integrated SQL, XML, Java and Text processing	Yes	No
One Engine, for Scalability and Ease-of-Use	Yes	No
Efficient XML Memory Management	Yes	No
Superior Functionality and Richness of Programmatic Access	Yes	No
Database Java and PL/SQL DOMs	Yes	No
Servlet Access	Yes	No
Drag-and-drop XML Load	Yes	No
Rich Programmatic API	Yes	No
Comprehensive XML vision, including an XML Repository	Yes	No
XML Repository	Yes	No
Folding	Yes	No
Access Control Lists	Yes	No
SQL Repository Search	Yes	No
Hierarchical Index	Yes	No

ORACLE XML DB

Oracle XML DB is the term for a set of features in the Oracle 9i Release 2 database that deliver high-performance storage and retrieval of XML. The technology included with the Oracle9i Release 2 database enables the popular Oracle Server to be used as a native XML database, in addition to being the most sophisticated and complete relational database currently available.

Oracle XML DB provides a storage-independent, content-independent and programming-language-independent infrastructure to store and manage XML

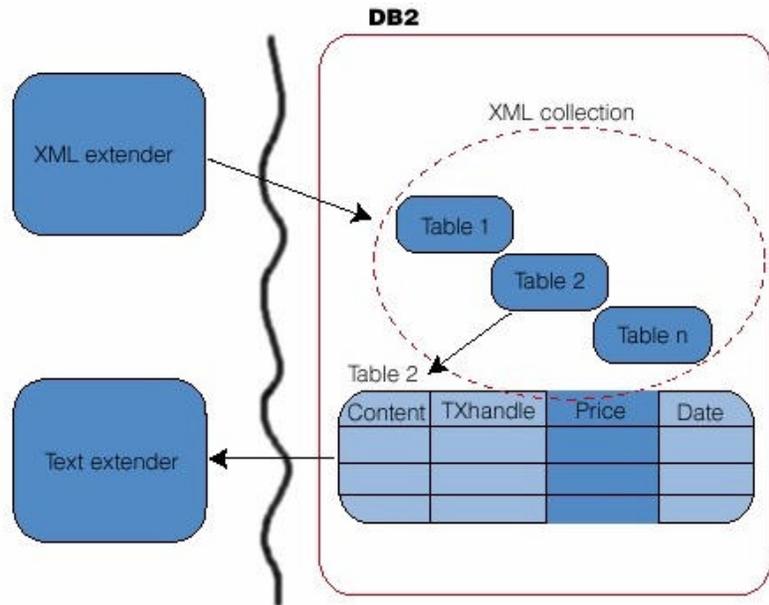
data. It delivers new methods for navigating and querying XML content stored inside the database, and introduces an XML Repository for managing XML content. With Oracle XML DB, you get all the advantages of relational database technology and XML technology at the same time.

Oracle XML DB provides:

- A native XML data-type, with built-in XML operations.
- The ability to absorb a standard W3C XML Schema data model into the Oracle database.
- XML/SQL duality, allowing XML operations on SQL data and SQL operations on XML content.
- Industry-standard methods for accessing and updating XML documents, including XPath and SQL/XML.
- An XML Repository that allows XML content to be organized and managed using a file / folder / URL metaphor
- Native database implementation of industry-standard, content-orientated, protocols including FTP, HTTP and WebDAV to move XML-content in and out of Oracle.
- Support for multiple, industry-standard APIs for programmatically accessing and manipulating XML content from Java, C and PL/SQL.
- XML-specific optimizations to handle memory management, document-hierarchy traversal, and query-processing.
- The enterprise class management capabilities of the Oracle database -- reliability, availability, scalability and unbreakable security -- on all your XML content.

IBM DB2 XML EXTENDER

Unlike Oracle, DB2 has to be 'extended' to support XML by separate pieces of software called the XML Extender and the Text Extender. These three pieces are developed separately by different organizations worldwide and are poorly integrated, and often inconsistent with each other. Users of such systems not only have to contend with the manageability-headaches arising from poor integration, but also the loose coupling, which proves to be less functional and less performant. This architecture is shown below.



With the DB2 XML Extender, users create a Document Access Definition (DAD) to encode the mapping between an XML document and underlying relational tables. ‘Default’ mappings are not created automatically. Given a mapping, the DB2 XML Extender reads an XML document and loads it into a DB2 database. ‘Mixed content’, i.e. islands of unstructured content interspersed with structured attributes, is handled using CLOBs and ‘side tables’ for indexing structured data contained within text. Due to poor integration between the Text and XML extenders, the handling of complex queries is clumsy.

NATIVE SUPPORT FOR THE XML DATA MODEL AND OTHER XML STANDARDS

Native XML Data Type

Oracle XML DB provides a native XMLType data type conformant to the SQL/XML standard from the ANSI SQLX working group for encapsulating common XML storage and retrieval functionality. DB2 does not provide a native XML data type.

XML Schema Integrated in the Database

Early adopters of XML exploited the standard's core characteristics of self-description and ad-hoc extensibility for the flexibility transportation of messages between applications. The second generation of XML standards such as XML Schema expanded the scope of XML technologies beyond data- or instruction-interchange. XML Schema is the first data model that can be used to represent both unstructured 'documents' and structured 'data'.

Today, applications store data in a relational database and documents or web content in a file system. XML is used mostly as an artifact for transport, generated from a database or a file-system. As the volume of XML being transported grows, and developers consider the costs of constant regeneration of XML documents, there arises the question whether these storage methods can effectively accommodate XML content. From these considerations, it becomes clear that XML Schema is an important model for databases to absorb, so that the core capabilities of strong relational management can be extended to all kinds of data, and also so that both storage and generation of XML can be done with the efficiencies that accrue from understanding the structure of XML.

The W3C Schema Working Group has published a specification of XML Schema to provide a means for defining the structure, content and semantics of XML documents². The XML Schema language is an improvement over DTDs in that it provides strong typing of the elements and attributes, uses XML syntax for its specification, can address content-models (mixed content, exact number of occurrences of elements, named group of elements), is extensible and self-documenting. Its type system is rich, defining 47 scalar data types, and this base set of data types can be extended using techniques like inheritance and extension to define more complex types. Sequences and collections are supported. Namespaces can be used to disambiguate names. XML Schemas can be designed to be variable -- supporting optional attributes, optional and repeated elements, and choices from alternatives of multiple elements.

The XML Schema type system is rich enough to address 'structured' relational data (i.e. where the structure of each item is regular, collections are homogeneous, and the terminal data-items consist of scalar values) as well as 'unstructured' documents (where the structure is flexible, and the document interleaves some data with regular structure and large portions of un-typed annotations or text which has irregular structure.) In addition, XML Schema can be used to specify semi-structured documents (in whom structure exists, but this structure is variable between instances.)

Oracle XML DB is the first mainstream commercial database to natively support the XML Schema data model. This means users can create XML Schemas

² <http://www.w3.org/XML/Schema>

using any popular schema authoring tool and register them to the database. Users can 'annotate' XML Schemas to direct specific layouts of storage, or have the database determine default storage layouts. When XML instances are inserted, Oracle XML DB can check the validity of each instance according to a registered schema. Various kinds of constraints specified in a XML Schema (ordering, cardinality, referential integrity) can be automatically enforced by the database.

High-Fidelity XML

Storing and retrieving XML Schema-based documents in a relational or object-relational database presents a number of novel challenges.

- **Preservation of the XML Document Object Model:** The tuples in a relational system have to inherent ordering. However, the relative ordering of elements in an XML documents (say paragraphs in a chapter) can be an important part of the semantics of the document. In addition, constructs like namespaces cannot be easily mapped to relational tables. These differences between what XML Schemas can represent and what basic relational models allow, can result in loss of fidelity as part of storing XML Schema-based documents in relational storage. Oracle XML DB provides DOM fidelity – the XML DOM stored is the XML DOM retrieved. DB2 does not provide high-fidelity XML storage and retrieval. DB2 breaks down XML documents into XML Collections, which are fully relational, not maintaining DOM fidelity. This means information stored can come out in wrong order unless applications explicitly maintain order. Other information kept in a DOM (e.g. namespaces) can also be lost.
- **Application of constraints and semantic rules:** XML Schemas can specify not only the structure but also semantics and business rules of certain kinds. It is important to be able to constrain Schema-based documents to all the semantics in the Schema, and not just the structural ones. DB2 does not allow native XML Schema-based constraining.

Comprehensive XML Standards

Oracle provides a fuller implementation of XML standards, including support for SAX in the database, SQL/XML, JNDI, Java Bean access etc. This creates an open, rich programming environment for XML. DB2 supports none of these standards for XML.

INTEGRATED SQL, XML, JAVA AND TEXT PROCESSING

One Engine, for Scalability and Ease-of-Use

With Oracle XML DB, XML and SQL are fully integrated. (In Oracle8i, Oracle had earlier integrated Text and Java with SQL inside the server). Oracle provides a native SQL datatype, XMLType, which stores XML and supports a wide variety of operations. It is possible to perform SQL operations on XML data, and also XML operations on SQL data. We call this 'XML SQL Duality'.

The SQL operators to operate on XML data in Oracle conform to the SQL/XML standard developed by the SQLX standards group in ANSI. The XML operations belong to various W3C standards such as XPath and XSL-T.

DB2 provides poor integration of XML and Java. You cannot perform Java processing on XML inside the server. XML cannot be parsed into an efficient Java DOM inside the server.

DB2 provides poor integration of XML and Text processing. The DB2 Net Search Extender and Text Information Extender do not work with the XML Extender. The Text index runs in a separate process outside the database. The loose integration results not only in manageability problems, but also in poor query optimization. (When a query involves XPath traversal, a text keyword match, and a SQL range search, DB2 cannot decide which to apply first. Oracle's Cost-Based Optimizer, on the other hand can apply the most restrictive clause first, resulting in superior performance and throughput.)

With the non-integrated architecture, building an XML application for DB2 is very complex. A developer has to

- Install XML Extender. Specifically enable database and relevant tables for the XML Extender. (You cannot use the XML extender on just any table.)
- Install Text Extender. Specifically enable database and relevant tables for the Text Extender
- Create an IBM proprietary DAD and store it in the file system to specify the XML-Relational mapping.
- Create DTDs to specify validation rules for XML.
- Explicitly manage separate Text and XML processing, with separate set of API.
- Hand-optimize queries on 'mixed content'.
- Pull XML into the mid-tier for any Java programming with XML

Efficient XML Memory Management

Oracle XML DB is more efficient for XML storage and materialization and as a result more scalable than DB2. XML document instances are relatively large for the amount of information they contain (due to the extra overhead of markup and conversion of all information into characters), and the Document Object Model relatively inefficient in terms of memory consumption. For such documents to be scalably stored and processed it is important to use the information latent in XML schemas for efficient storage and retrieval. Oracle XML DB can lazily materialize an XML DOM inside the database or at a Java client.

DB2 has to wait till the entire DOM is constructed from an XML parse, and laid out in memory. Oracle's virtual DOM can be deployed at the client or in the Server JVM, leading to far superior performance and scalability across multiple deployment configurations. DB2 does not provide efficient memory management of XML, which results in poor scalability with large XML document. Any Java processing can only be done in the mid-tier, adding to the overhead of moving pieces for even small amounts of processing logic on XML.

SUPERIOR FUNCTIONALITY AND RICHNESS OF PROGRAMMATIC ACCESS

Oracle XML DB provides extra functionality as well as richer programmatic access compared to DB2.

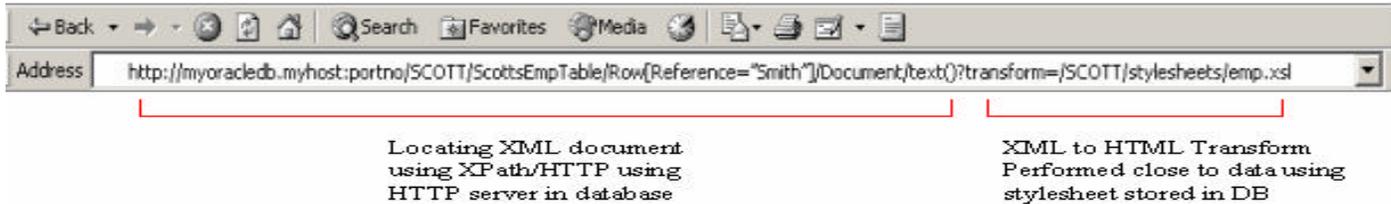
Database XML Java & PL/SQL DOM

With Oracle XML DB, XMLType data can be read into a Java or PL/SQL DOM at the server for running data-intensive logic close to the data. (Further, this DOM is highly efficient with 'lazy materialization' as discussed above.) DB2 does not provide a DOM inside the server, and as a result cannot provide the same functional XML processing capabilities as Oracle.

Servlet Access

You can manipulate XML data in the Oracle XML DB server with the standard servlet API, in addition to processing XML with PL/SQL or Java. There is no such capability in DB2.

With support for a HTTP protocol server as part of Oracle XML DB, as well as integrated storage/invoke facilities for XSL sheets, generation of XML from HTTP queries over the web is possible from a browser. This integrated XML processing with no moving pieces other than Oracle XML DB and a standard web browser is not possible with DB2.



Drag-and-Drop XML Load

Oracle provides multiple ways of loading XML documents, including easy to use techniques like drag-and-drop using FTP and WebDAV. Without these features, using DB2 in interactive, end-user focused applications is not feasible.

Rich Programmatic API

Oracle XML DB supports a much richer set of programmatic API compared to DB2, including

- Content-oriented storage and access through WebDAV and FTP
- Programmatic dynamic access via DOM and SAX.
- Programmatic static access through Java Beans
- Hierarchical access via JNDI

DB2 supports none of these API.

COMPREHENSIVE XML VISION, INCLUDING AN XML REPOSITORY

Oracle XML DB can be used for a large number of application areas, including the emerging content-management area. The XML data model includes unstructured as well as structured information, and a host of

industries from desktop publishing to document archives use XML as a content model.

XML Repository

Oracle XML DB provides specific capabilities for content-oriented access to XML. Over the years, databases have introduced data management services such as schemas, roles, views etc. Content-management requires different services to organize and deploy applications. Key to these services is an Internet XML Repository.

DB2's vision of XML is limited to mapping-to-and from relational data; there is no vision of high-fidelity, high-performance XML, nor one of XML as a vehicle of managing content.

Some of the important aspects of Oracle XML DB's full-spectrum XML support include

Folding

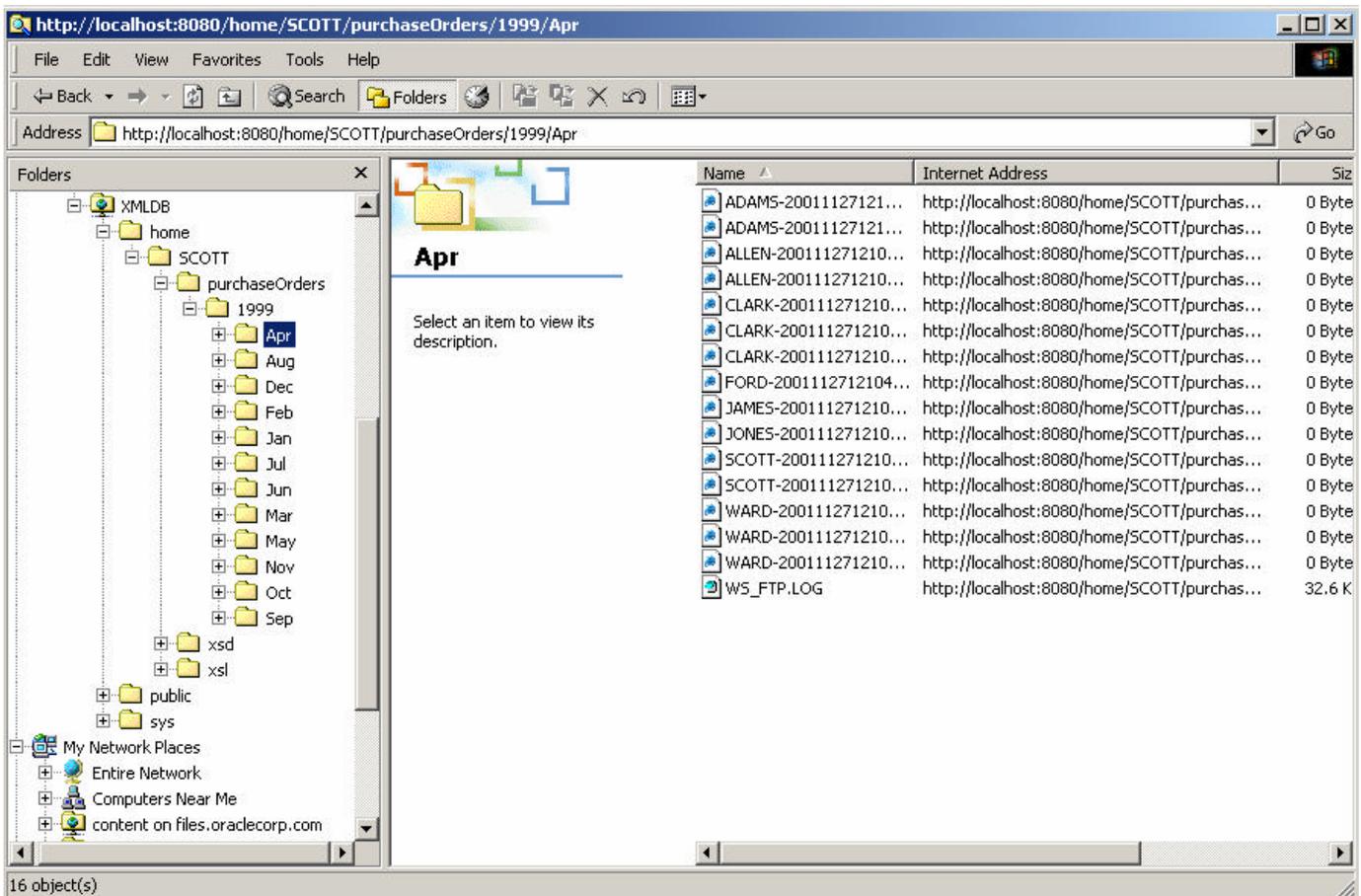
You can organize database XML content into folders, and expose the whole database as a directory-like hierarchy of folders. These folders can be navigated from common desktop tools. Also, you can map existing relational rows into URLs, providing access control, modification date tracking, and other metadata management for those rows.

Access Control Lists

You can create high-performance access control lists (ACLs) for any XMLType object, and define your own privileges in addition to the system-defined ones. You thus get fine-grained security on XMLType.

Integrated Protocol Servers

Oracle XML DB has built-in protocol servers (HTTP, FTP, WebDAV) to help move XML documents in and out without low-level relational loading/unloading. You can also access any foldered XMLType row via WebDAV and FTP, say for viewing XML documents in a Microsoft Windows Explorer™ or similar WebDAV desktop interface.



SQL Repository Search

Operators like UNDER_PATH and DEPTH, allow applications to search folders, or file metadata like owner and creation date, as well as file contents via SQL

Hierarchical Index

A special hierarchical index to speed pathname resolution and folder search. Additionally, you can automatically map hierarchical data in relational tables into folders.

DB2 has no comparable functionality.

CONCLUSION

The significant advantages that Oracle XML DB has over IBM DB2 makes it the most natural technology for XML. Not only does Oracle have a more comprehensive XML vision that envisages XML being used to represent content as well as data, but Oracle also supports a broader set of XML standards more natively, with integrated performance and scalability that the IBM DB2 XML Extender cannot match.

FURTHER INFORMATION

- The Oracle XML DB homepage on Oracle Technology Network:
<http://otn.oracle.com/tech/xml/xmlldb/content.html>.
- Oracle XML DB on Oracle.com:
http://www.oracle.com/ip/index.html?xmldbcm_intro.html



Oracle XML DB vs. IBM DB2 XML Extender White Paper
February 2003
Author: Sandeepan Banerjee
Contributing Authors: Vishu Krishnamurthy, Eric Sedlar, Mark Drake

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
www.oracle.com

Oracle Corporation provides the software
that powers the internet.

Oracle is a registered trademark of Oracle Corporation. Various
product and service names referenced herein may be trademarks
of Oracle Corporation. All other product and service names
mentioned may be trademarks of their respective owners.

Copyright © 2003 Oracle Corporation
All rights reserved.