# New Features in Oracle XML DB for Oracle Database 11g Release 1

*An Oracle White Paper*
*June 2007*

ORACLE

# New Features in Oracle XML DB for Oracle Database 11g Release 1

**The growing number of XML-based standards and the adoption of XML by office productivity tools is driving a significant growth in the volume of XML organizations have to manage effectively.**

**Organizations need a single platform that can manage highly structured XML, semi-structured XML and unstructured XML.**

## INTRODUCTION

XML has been widely adopted in just about every industry. XML standards are emerging in sectors such as healthcare, financial services, manufacturing, publishing, law enforcement and government. XML and XML Schema also provide the foundation for service-orientated application development. The introduction of standards such as XBRL has led to XML becoming the tool of choice for exchanging information between application systems and the use of XML as a persistence model for mission-critical data is growing.

In addition to the growth in the use of XML for managing highly structured content, there has been an equally significant adoption of XML as the primary format for persisting semi-structured and unstructured content. This trend can be seen by the use of XML-based storage in the current generation of Microsoft Office and Open Office. The movement to XML-based documents will drive a growing demand for XML-based content management (CM) systems. These systems will differentiate themselves from traditional CM solutions by being able to understand both the metadata and the content of the documents that they manage.

This growth in the use of XML has created a requirement to manage XML with the same degree of rigor as is required for other mission-critical data. To meet this need, Oracle developed Oracle XML DB. Oracle XML DB is a high-performance, native XML storage and retrieval technology that is delivered as a part of all versions of Oracle Database. Oracle XML DB provides full support for all of the key XML standards, including XML, Namespaces, DOM, XQuery, SQL/XML and XSLT. Oracle XML DB was first released with Oracle 9iR2, and it has been enhanced in each subsequent major release of the database.

**Only Oracle delivers a performant, hybrid solution allowing efficient XML operations on relational content. and SQL operations on XML content.**

**Oracle Database 11g provides full support for SQL-centric and XML-centric application development. In additional Oracle is unique in providing an XML repository that enables document centric standards and metaphors.**

Unique among mainstream databases, Oracle XML DB also incorporates a high-performance XML repository. This repository makes it possible to organize XML content stored in Oracle Database using the familiar file/folder metaphor. The repository can be accessed directly from SQL or using standard Internet protocols such as HTTP and FTP. This makes it possible to access XML content stored in the database directly from common desktop tools such as Microsoft Office. Oracle XML DB Repository allows Oracle Database 11g to meet the needs of content-centric, as well as data-centric XML application development.

By providing full support for XML standards, Oracle XML DB supports native XML application development. Application developers are able to use XML-centric techniques to store, manage, organize, and manipulate XML content stored in the database. Oracle XML DB also supports the SQL/XML standard, which allows SQL-centric development techniques to be used to publish XML directly from relational data stored in Oracle Database 11g.

Oracle XML DB was the first platform to deliver hybrid relational/XML capability, bringing the full power of the SQL language to bear on XML content and bringing the full power of the XML language to bear on relational data.

With the release of Oracle Database 11g, Oracle extends its industry-leading XML support, ensuring that Oracle remains the best platform for storing, managing, and querying all types of XML content. Oracle Database 11g offers improved performance and scalability and delivers complete support for the flexibility that makes the XML data model so powerful. Oracle Database 11g also includes new functionality that makes it easier to leverage the full

power of the database when developing next-generation service-oriented applications.

### NEW IN ORACLE DATABASE 11G

From an XML perspective, the major goal of Oracle Database 11g was to deliver increased flexibility combined with improved performance. This was achieved by further optimization of XML Schema-based storage and the introduction of additional storage and indexing techniques for non-schema-based XML content.

## IMPROVEMENTS FOR XML –SCHEMA-OPTIMIZED XML STORAGE

New Features for Schema-Optimized XML

In-place Schema Evolution.

Partitioning Support .

Intelligent Defaults.

Enhanced XQuery support.

Performance Optimizations.

Oracle Database 11g offers a number of improvements for users of Oracle XML Schema-optimized XML storage, including:

- In-place XML schema evolution. This feature makes it possible to make certain kinds of changes to XML schemas, with zero down time. In-place XML schema evolution provides support for most of the common types of change that are made to an XML schema, including, but not limited to, the addition of new elements and attributes.

- Oracle Partitioning support. XML schema-based storage can now be used in conjunction with the Oracle Partitioning option. This lets you use Oracle Database 11g to manage very large volumes of XML documents. Full support is provided for the partition maintenance operations and partition pruning that make Oracle Partitioning the correct solution for managing very large volumes of data.

- New, intelligent defaults for XML schema-optimized storage. These defaults have been implemented based on in-depth knowledge of how best to structure the underlying storage model in order to deliver optimal performance for common XML operations. These new defaults also make it easier to use the capabilities of Oracle Text to perform text-based searches on the contents of specific elements within an XML document.

- Enhanced XQuery support. Two new SQL operators, XMLExists and XMLCast, expand Oracle Database's support for XQuery. Oracle Database 11g also delivers significant additional optimizations for XQuery operations on top of XML schema-based storage. Oracle Database also delivers significant improvements in the area of node-level updates in conjunction with XML schema-based storage.

- When compared with Oracle Database 10g Release 2, Oracle Database 11g delivers noticeably better performance for a number of common XML operations. This includes a 10X improvement in throughput when loading certain types of XML data into XML schema-optimized storage and a 10X improvement for XML publishing operations that make

extensive use of the XMLAgg SQL operator. The implementation of streaming output also delivers significant performance improvements when using XSL to generate large documents.

- Finally, Oracle Database 11g has eliminated the existing 64K limitation on the size of a text node.

## IMPROVEMENTS FOR SCHEMA-LESS XML STORAGE

There are a number of use cases where it is not possible to effectively leverage all the benefits of Oracle's XML schema-based storage model. One of the most exciting new features of Oracle Database 11g is a new binary XML storage model and XML indexing techniques. These each deliver significant improvements to Oracle's ability to optimize operations on non-schema based XML data. They allow Oracle XML DB to deliver high-performance insert, update and query operations in cases where the full flexibility of the XML data model is required and the use of an XML schema is not appropriate.

**New Binary XML storage option provides significant optimizations for schema-Less XML, including:**

**Reduced storage requirements.**

**More Efficient CPU, Network and Memory utilization.**

**Efficient leaf and fragment level extraction.**

**Flexible XML Schema support.**

**Streaming Parsing and Validation.**

**Support for locale-sensitive documents.**

### Binary XML

The primary advantages of Oracle's binary XML XMLType storage format are as follows:

- Reduced storage requirements. Oracle binary XML is a compact representation of an XML document. This significantly reduces the amount of disk space required to store XML documents on disk, even before traditional data-compression techniques are applied. These space savings arise from tokenization of tags as well as conversion from text to native representation for text nodes and attribute values.

- Reduced CPU and memory overhead. Oracle's binary XML format addresses one of the biggest problems associated with large-scale XML deployments, which is the overhead associated with parsing and serializing XML every time the data moves between the different application tiers or before storing it on disk. With Oracle binary XML, the on-disk representation is the same as the in-memory representation and the on-wire representation. Since the same

representation of the XML is shared by all of tiers, this enables efficient exchange of XML content.

- Reduced network overhead. Oracle binary XML also reduces network overhead by using the compact internal format, rather than the traditional serialized text format, to transmit XML on the wire.

- Efficient leaf and fragment extraction. Oracle's binary XML format supports streaming XPath evaluation, allowing multiple leaf nodes or fragments to be extracted in a single pass through an XML document.

- Flexible XML Schema support. Oracle binary XML supports both XML schema-based and non-schema-based data. Unlike the XML schema-optimized storage model, the binary XML model allows XML documents that are associated with multiple XML schemas to be stored in the same table or column. It also eliminates the majority of problems that can arise when changes occur to an XML schema.

- Streaming XML parsing and validation. Oracle binary XML introduces a new pull parser to ingest XML and a new streaming validator for XML schema validation. Both the parser and the validator avoid using a DOM tree, which leads to significant reductions in the amount of CPU and memory required for XML processing. Binary XML also reduces the overhead needed to validate XML documents during update operations, by allowing fragment-level, rather than document-level validation. This means that only the parts of an XML document that are modified need to be re-validated following an update.

- Support for language-sensitive operations. Oracle's binary XML supports the W3C XLIFF standard, allowing content to be managed in a locale-sensitive manner. If a document contains language-sensitive content, then only content appropriate to the current locale is returned when the document is accessed.

- Tight integration with Oracle Database 11g SecureFiles. Oracle's binary XML storage model leverages all of the

benefits of Oracle SecureFiles, to ensure maximum throughput when storing and retrieving XML documents. Binary XML also leverages the SecureFiles sliding insert feature, to efficiently perform node-level insert, delete, and update operations on XML content.

Along with the introduction of the new binary XML format, Oracle Database 11g adds support for streams-based replication of text-based XMLType data. This allows XMLType to be used in situations where high-availability is needed.

## XML Indexing

New XML Indexing capabilities:

Fully Integrated with XQuery and the Binary XML format.

Path Subsetting allows you to choose which fragments to index.

Synchronous and Asynchronous modes of operation.

Supports partial re-indexing for update operations.

Integrated with Oracle Text for full text indexing.

One of the primary design goals for the Oracle binary XML format is to enable efficient path-based indexing of XML content. The binary XML format was designed from the ground up to allow optimization of XQuery evaluation and extraction of XML leaves and fragments. The new XML indexing capabilities of Oracle Database 11g take full advantage of this aspect of the binary XML format. The major benefits of Oracle's new XMLIndex index include the following:

- Full integration with XQuery. Oracle's XQuery capability, first introduced in Oracle Database 10gR2, has been tightly integrated with binary XML storage and XMLIndex. This means that developers using XQuery have complete access to the full power of Oracle's XML indexing when working with Oracle's binary XML

- Oracle's XMLIndex index type supports path subsetting. As the name implies, path subsetting lets developers use XPath expressions to limit which nodes in a XML document should be indexed. This makes it possible to balance query performance with application performance, index size, and index maintenance overhead. Path subsetting can be specified on an inclusive or exclusive basis at index creation time and it can be modified without recreating the entire index.

- XML Index has synchronous and asynchronous modes of operation. In synchronous mode, an XMLIndex index operates just like any other database index: the index maintenance takes place as part of the transaction that modifies the data. In asynchronous mode, the index maintenance takes place separately from the transaction that modifies the data. Applications thus avoid being impacted by the overhead associated with full XML indexing. Asynchronous mode is fully automatic: there is no requirement for an application or a DBA to force re-synchronization of the index and the index does not suffer from the fragmentation issues that are often seen with other kinds of asynchronous index.

- Oracle's XMLIndex is also optimized for efficient re-indexing in the presence of node and fragment level updates. When a node-level or fragment-level update occurs, only the parts of a document affected by the update are re-indexed. This can lead to much higher performance improvements than can be achieved with systems that are forced to completely re-index the document in these circumstances.

- One other unique feature of the Oracle XMLIndex is that it can be used a source of metadata. It is possible to use the XMLIndex to discover information about the structure of the documents that have been indexed. This can be extremely useful when attempting to perform analysis of large amounts of unstructured XML content.

- The Oracle XML Index is also tightly integrated with Oracle's full-text indexing capability allowing full-text indexes to be created on the contents of the leaf level text-nodes and attributes in the XML documents. This enables efficient full-text searches to be performed across XML content.

- In addition to supporting binary XML, XML Index can be also be used with text-based XMLType. This allows an XML Index to be created on fragments within a Schema-Optimized storage model that has been mapped into CLOB based XMLType.

## DATABASE NATIVE WEB SERVICES

**Database native web-services allow Oracle Database 11g to directly participate in a service-orientated architecture**

**PL/SQL packages, procedures and functions automatically exposed as web-services.**

**Generic SQL and XQuery query service.**

Web services have become a popular way of developing applications. The W3C's Simple Object Access Protocol (SOAP) standard has become the most common way for applications to interact with Web services. SOAP makes extensive use of XML. The Web Services Description Language (WSDL) and the request and response documents that form the foundation of the protocol are all XML-based. Until now, creating a Web service has required hand-coding or generating the programs necessary to expose a SOAP end-point, and then deploying this code using an application server. These programs need to address all aspects of a Web service's usage, including generating the WSDL document, parsing and processing

the SOAP request, and marshaling the required data into the correct XML format to provide the SOAP response.

Database native Web services, delivered as part of Oracle Database 11g, eliminates all of this complexity for Web services that provide access to data stored in an Oracle Database. Database native Web services let you expose PL/SQL stored procedures, functions, and packages as Web services with zero coding and zero –deployment effort. The database native Web services feature also includes a Web service that supports execution of dynamic SQL queries and XQuery expressions. The database HTTP server, provided as part of Oracle XML DB Repository, allows these Web services to be accessed using HTTP and HTTPS, without any additional application server infrastructure.

Database native Web services are very efficient. They leverage the powerful XML capabilities of Oracle XML DB to automatically generate the WDSL, parse and process the SOAP request, and generate the SOAP response.

## ORACLE XML DB REPOSITORY IMPROVEMENTS

**Oracle XML Repository is now programmable. PL/SQL and Java code can be associated with operations on files and folders..**

**Compound document supports enables managing logical XML documents as a set of separate components,**

**Support for JSR-170 allows development of repository-independent content management applications.**

In Oracle Database 11g, Oracle XML DB Repository continues to deliver an infrastructure uniquely suited to solving the problems of XML content management (XCM). Specific improvements in Oracle Database 11g include

- Using the new binary XML storage model and XML indexing features of Oracle Database 11g to provide significant improvements in the management of document-centric XML data. In particular, in Oracle Database 11g, you can create a repository-wide XMLIndex index. This index combines metadata and content, and it greatly improves the repository's ability to effectively manage queries over content and metadata.

- A comprehensive repository event model, which automatically executes PL/SQL and Java procedures when operations take place on content stored in the Oracle XML DB repository. This makes the repository programmable in the same manner that database triggers make the database programmable. Using repository events makes it easy to develop workflow and other kinds of advanced content-

centric applications: the simple event of operating on a document can automatically trigger the required processing.

Events can be triggered based on a number of different factors, including location and content type. This allows specific processing to be associated with a particular folder tree, or a particular kind of document. Examples of this include moving a purchase requisition into an `approved` folder, thus initiating an ordering process, or creating a repository-wide procedure that automatically catalogs digital pictures based on the image metadata.

- Oracle Database 11g incorporates compound-document support based on the XInclude and XLink standards. Compound document supports allows a single logical XML document to be mapped to a series of smaller component documents. When the compound document is inserted into Oracle XML DB Repository or updated, the necessary changes are automatically propagated to the underlying components. When the compound document is retrieved, its content is automatically generated from the individual components. This has the advantage of allowing for reuse, since a given component can be used with more than one document, and also makes it easier for a team of authors to work on a single logical document.

- In order to make Oracle XML DB Repository more accessible to Java developers, Oracle Database 11g includes Oracle Content Connector. Oracle Content Connector is an implementation of the JSR 170 standard. This standard defines an open API that can interact with different content management systems, in the same way that JDBC defines an open API that can be used interact with different database systems. The JSR 170 support in Oracle Database 11g lets developers implement XCM systems that avoid being locked into a particular content repository, but which are still capable of leveraging the full benefits of Oracle XML DB.

## ENHANCED ACL SECURITY

**Oracle XML DB ACL base security enhancements..**

**ACL Inheritance allows definition and enforcement of organization wide security policies.**

In Oracle Database 11g, the Oracle XML DB ACL-based security model has been enhanced in a number of ways, including the following:

- ACL Inheritance. ACL inheritance simplifies the process of defining, managing, and enforcing a common set of security policies across all of the documents stored in Oracle XML DB Repository. These rules can be organization-wide policies or policies specific to certain types of documents. These rules are specified by creating one or more master ACLs. With ACL inheritance it is possible to ensure that all new ACLs must be based on an existing ACL. This ensures that the newly created ACL inherits all of security policies defined by the ACL it is derived from, ensuring that the policies defined by the parent ACL are enforced whenever the new ACL is used.

- DAV ACL Compliance. The Oracle XML DB ACL model has been enhanced to provide more complete support for the DAV ACL specification. This will allow improved interaction with clients that provide support for the DAV ACL security model.

- User defined ACLs. In Oracle Database 11g the set of permissions defined by Oracle XML DB can be extended to allow the ACL based security model to be used to secure other kinds of database object.

- Time-sensitive ACLs. In Oracle Database 11g it is possible to create ACLs that enforce access control polices in a time-sensitive manner. This can be used to automatically publish and then expire content, based on rules defined by the ACL.

## ORACLE XML DEVELOPER'S KIT IMPROVEMENTS

**Oracle XDK Enhancements**

**Pull Parser and Steaming Validator enable much more efficient XML processing**

**Full support for Oracle Binary XML in Java and 'C'**

**New XMLDiff and XMLPatch capabilities..**

Oracle has continued to invest in the Oracle XML Developer's Kit (XDK). There are a number of new features in the Oracle XDK for Oracle Database 11g:

- The pull parser and streaming validator developed for Oracle's binary XML storage model are available to all users of the C-language version of Oracle XDK.

- The Java version of Oracle XDK has been enhanced to offer tight integration with the Oracle binary XML format and a new Scaleable DOM. This also allows extremely efficient parsing, processing and serialization of binary XML format in a pure Java environment using the standard DOM API defined by the W3C. Scaleable DOM enables DOM based operations on very large XML documents without incurring the memory overheads associated with typical DOM implementations.

- Oracle XDK now includes new XMLDiff and XMLPatch capabilities. XMLDiff generates an XML representation of the differences between two XML documents, XMLPatch can take the output from XMLDiff and apply it to a source document, transforming that into a document that is compliant with the target document. This makes it possible to accurately identify the differences between two XML documents and to create applications that can replicate changes to XML documents in a very efficient manner. XMLDiff and XMLPatch are available as SQL operators inside the database, as well as standalone utilities and callable 'C' APIs.

Oralce Database 11g Oracle XML DB
June 2007
Author: Mark Drake
Contributing Authors: Oracle XML DB Development Team

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
oracle.com

Copyright © 2007, Oracle. All rights reserved.
This document is provided for information purposes only and the
contents hereof are subject to change without notice.
This document is not warranted to be error-free, nor subject to any
other warranties or conditions, whether expressed orally or implied
in law, including implied warranties and conditions of merchantability
or fitness for a particular purpose. We specifically disclaim any
liability with respect to this document and no contractual obligations
are formed either directly or indirectly by this document. This document
may not be reproduced or transmitted in any form or by any means,
electronic or mechanical, for any purpose, without our prior written permission.
Oracle is a registered trademark of Oracle Corporation and/or its affiliates.
Other names may be trademarks of their respective owners.