XODB®

MADE BY ARCITECTA

High performance, massively scalable NoSQL binary-XML object database engine for Mediaflux.

XODB is the powerful binary-XML object database engine embedded at the heart of Mediaflux, which enables the flexibility and performance needed to handle data management problems of any size.

Designed with non-IT users in mind and to run on any hardware, XODB can easily adapt to changing data requirements as they evolve over time.

Keeping ahead of evolving data

Managing data is not a static event. Data structures and metadata evolve over time as infrastructures grow, new use cases arise, and new technology becomes available. XODB was designed with this reality in mind, as the powerful data management engine at the core of Mediaflux. It brings the power and flexibility to metadata and data management that is unattainable with any relational database or stand alone proprietary database.

XODB is embedded directly into Mediaflux, which removes the context switching and network overhead between the application and database, and allows the application to function as a single executable process. This greatly simplifies deployment and maintenance. Designed for zero administration, XODB enables end-users full capability to customise data schemas on the fly in live systems with complete data protection and without the need for a dedicated database administrator.

Flexible, extensible, portable

Valuable data inevitably lasts longer than any system it currently lives on. The problem as is that data and metadata may originate in proprietary environments that can result in stranded and/or stale data as use cases and infrastructures evolve. Written in pure JAVA for maximum platform flexibility, XODB stores all metadata in XML to ensure that it is fully portable and compatible with open standard environments. As a document-oriented system that manages both structured and unstructured data with validation, XODB is designed to bridge across any data type, on any hardware or storage system.

ARCITECTA
OPERATING SYSTEMS FOR META+DATA

www.arcitecta.com | talk@arcitecta.com

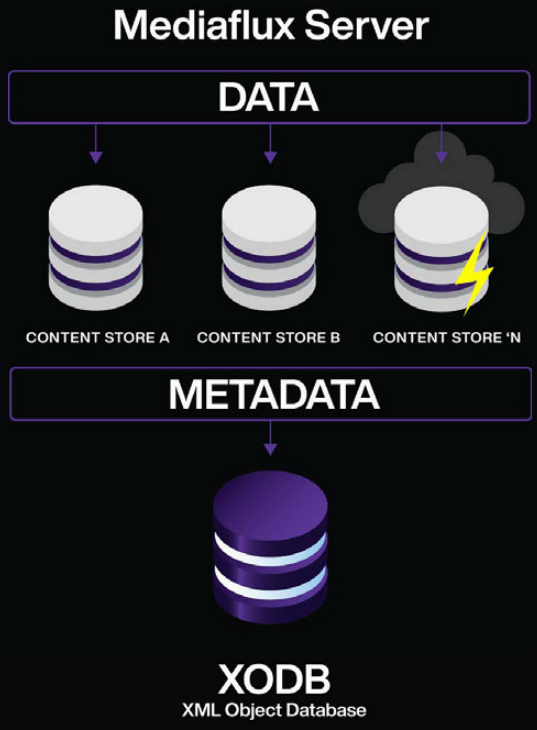
The best of both worlds

XODB is a hybrid that combines the best attributes of object databases (clustering, complex objects, pointer de-referencing) with XML (structured, typed, versioned, with easy schema evolution). XODB is an object database where everything exists as related objects and where the object “state” is persisted as compact XML.

- All XML within XODB is binary encoded using a combination of numeric tokenisation and compression to produce a binary document that is:
- Significantly smaller than the original text XML - (reducing data footprint)
- Can be rapidly evaluated (in binary form) - (increasing performance)
- Is easily reinflated into the original XML - (ensuring portability)

The result is all data for a given object is stored in a binary compressed XML format which minimises the amount of memory and disk space required to store an object, while retaining the extensibility benefits of XML.

XML eliminates the need for joins that would be normally be required in a relational database – all data for an object is stored in the one record.



Key XODB features and benefits

- Clusters all data for an object in a compressed binary XML format.
- Allows incremental, on-demand, schema evolution.
- Controls data accessibility for improved authentication and authorisation, compliance and classification.
- Supports federation and replication. Supports partial query evaluation.
- Supports complex XPath expressions.
- Supports custom metadata through an extensible framework of services.
- Provides high levels of concurrency.
- Supports billions of assets per instance.
- May be used as a single node, or multiple instances clustered for extreme scalability.
- Written entirely in Java and does not require the installation of additional database software.
- Requires no administration except periodic backup.

Key XODB features and benefits (continued)

Dynamic collections

Mediaflux XODB supports the creation of dynamic collections where membership is determined based on arbitrary query. Objects are automatically added or removed dependent upon them satisfying the query constraints of the collection at inception, update or deletion. Collections significantly improve the evaluation performance of very large data sets.

Indexes

Small databases (a few million objects) often require no indexes, as the sequential XML evaluation performance is sufficient. This further decreases the overall disk space required by the database. Indexes can also be dynamically added for any XPath element or attribute.

Concurrency

Mediaflux XODB uses a 2-stage cache that allows both reads and updates concurrently for the same object. Objects that are part of an update transaction will not be locked unless they will be updated, significantly improving concurrency. Mediaflux XODB includes graph-based deadlock detection.

Transaction performance

Mediaflux XODB uses asynchronous I/O and I/O scheduling to maximise disk I/O performance. A given session can indicate the nature of update transactions allowing the independent control of transaction performance and recoverability on a per-session basis.

Advanced query constructs

Mediaflux XODB enables multi-level qualified XPath expressions. Mediaflux XODB also supports advanced queries that are not normally found in relational database technology, such as:

- Full regular expression pattern matching.
- Plug-in code to extend the types of expressions (e.g. content evaluation as part of the query).
- Range set queries and ‘contains’, ‘contains any’, ‘contains all’.
- Spatial queries and indexes; Relative temporal queries and indexes.
- Extensible XPath-like Functional Query Language (FAQL).

Query optimisation

XODB query performance is several orders of magnitude faster than relational databases. This gap increases non-linearly the more complex the query. The query engine employs fuzzy logic to produce candidate sets, which are incrementally refined to produce a result set. Partial query execution scales to billions or more objects.

XODB evaluates dynamically, in real-time, result sets that would typically require pre-computation and caching in most other systems.

The Mediaflux XODB query optimiser organises query execution based on the available indexes, dynamic collections, specified sub-sets and cost of execution.

Our primary objective is to provide highly scalable millisecond querying for people, and to provide an experience similar to that of Internet search engines.

Installation and administration

Mediaflux XODB is part of a standard Mediaflux server software installation. No other database software is required. Consequently, a Mediaflux server can be installed and fully operational within half an hour.

Other than periodic backup, a Mediaflux XODB database requires no other administration. If configured for dynamic replication, then backup is not required either. Mediaflux XODB also supports online backup.

Platform support

Mediaflux XODB is written entirely in Java and is available on a diverse range of platforms including Linux, Mac OS X and Windows.

Hardware requirements

Deployments for a few users can be supported by a laptop or desktop computer. Workgroup deployments typically require a small server with a few processing cores and at least 4GB of available RAM.

Systems can be deployed in multi-node clusters to scale performance and I/O. In addition, systems can be distributed geographically for global deployments in federated or replicated implementations.

Software requirements

Mediaflux is a self-sufficient and cost-effective solution. It requires only an operating system and Java Runtime Environment (JRE) version 1.7+. No other software is required.