Key *eXtreme*DB® Embedded Database Features

In a benchmark of embedded database systems, "eXtremeDB turned in results almost 10 times faster than the next closest competitor."
-- MOD Systems

Welcome to *eXtreme*DB, the high performance database system built from the ground up for intelligent devices, embedded software and scalable real-time systems. These applications demand the ultimate speed, reliability, resource efficiency and development flexibility. *eXtreme*DB meets these needs through the features listed below. Please contact McObject about any feature not included on this list.

*eXtreme*DB Embedded Database Architecture

Core In-Memory Database System (IMDS)
Tiny Footprint (Approximately 150K)
Optional On-Disk or Hybrid Storage

Powerful Runtime Features

ACID Transactions
Transaction Logging
Multi-Version Concurrency Control (MVCC)
Cache Prioritization
Deterministic Rule-Based SQL Optimizer
Security Features
Remote Procedure Calls (RPCs)
XML Import/Export

High Availability
64-Bit Support
Open Replication
Event Notifications
Pattern Search
Binary Schema Evolution
Database Striping/Mirroring
Kernel Mode Deployment

Unmatched Developer Flexibility

C/C++, SQL, Java APIs
Wide Range of Supported Data Types
Custom Collations
Source Code Available
Unmatched Developer Support

Proven Solution
Designed to Prevent Database Corruption
Broad Platform Support
B-, R- and KD-Tree, Patricia Trie and Hash Indexes

The *eXtreme*DB Embedded Database, Feature-by-Feature:

Core In-Memory Database System (IMDS) Design

As an in-memory database system (IMDS), *eXtreme*DB gives your application speed without rewrites or expensive new hardware. It eliminates disk and file I/O, cache management and other sources of latency. By working with data directly in main memory, *eXtreme*DB avoids the overhead of data duplication and transfer inherent in disk-based DBMSs. Databases can be created in shared memory, enabling concurrent access by multiple processes. Read about the performance advantages of IMDS technology in the white papers offered here.

Tiny Footprint (Approximately 150K)

Small code size and minimal overhead (database system memory footprint is as small as 15% of managed data volume) means less RAM is required; *eXtreme*DB's streamlined design permits a lower cost CPU. As a result, your design can use less expensive hardware, resulting in lower manufacturing costs. Or, use the extra CPU speed to offer a snappier user experience than your competition, and use the extra memory to manage more data at the same cost as competing products.

Optional On-Disk or Hybrid Storage

In addition to the core *eXtreme*DB in-memory data storage, McObject's *eXtreme*DB Fusion edition adds hybrid storage: certain record types can be designated for on-disk storage, with flexible caching. Choose the best storage medium based on performance, persistence, cost and form factor.

High Availability

Committed to 99.999% uptime or better? *eXtreme*DB High Availability (HA) edition ensures continuous database operation even in the face of hardware or software failure. *eXtreme*DB HA supports both synchronous (2-safe) and asynchronous (1-safe) replication, with automatic failover. With 2-safe replication, a time-cognizant two-phase commit protocol gracefully handles timeout situations.

ACID Transactions

*eXtreme*DB's transactions support the ACID (Atomic, Consistent, Isolated and Durable) principles, which safeguard data integrity by guaranteeing that updates will complete together or the database will roll back to a pre-transaction state. *eXtreme*DB also enables the developer to prioritize transactions.

64-bit Support

The 64-bit edition of *eXtreme*DB scales to support databases that are terabytes in size, to address the real-time database needs of financial, social network, object-caching and other highly scalable applications.

In a published benchmark test, an *eXtreme*DB-64 database grew to 1.17 terabytes and 15.54 billion rows, with no apparent limits on it scaling further. Performance remained consistent as the database size increased, suggesting nearly linear scalability. For a simple SELECT against the fully populated database, *eXtreme*DB-64 processed 87.78 million query transactions per second using its native application programming interface (API) and 28.14 million transactions per second using a SQL ODBC API.

Multi-Version Concurrency Control (MVCC)

Another important tool for scalability is *eXtreme*DB's optional Multi-Version Concurrency Control (MVCC) transaction manager, which is available as an alternative to the "pessimistic" database locking of the original *eXtreme*DB MURSIW (MUltiple Reader, SIngle Writer) transaction manager. MVCC can dramatically improve scalability and performance, especially in applications with on-disk or hybrid (in-memory and on-disk) database storage; many tasks or processes concurrently modifying the database (versus read-only); and in multi-core systems.

Transaction Logging

*eXtreme*DB Transaction Logging edition adds recoverability by writing database changes into a transaction log on persistent media. Logging may be set to different levels of transaction durability, allowing system designers to make intelligent trade-offs between performance and risk of unrecoverable transactions.

Open Replication

McObject's *eXtreme*DB Data Relay technology facilitates seamless, fine-grained data sharing between real-time systems based on *eXtreme*DB, and external systems such as enterprise DBMSs. As part of the *eXtreme*DB Transaction Logging edition, Data Relay helps developers by simplifying the code that "looks inside" database transactions for changes that should be relayed. It also guarantees maximum efficiency by eliminating the CPU-intensive task of monitoring database activity. Sharing of data can be either synchronous or asynchronous.

Cache Prioritization

*eXtreme*DB Fusion improves on typical Least Recently Used (LRU) cache policies by allowing applications to influence how long certain pages remain in cache, to minimize retrieval overhead for objects used in time-sensitive tasks. When considering whether to remove a page from the cache, *eXtreme*DB's LRU algorithm examines the cache priority property set for the page; the higher the priority, the longer the page remains linked to the LRU list (stays in cache). A caching priority of zero is the default.

Event Notifications

Similar to triggers in a relational database management system, this feature enables *eXtreme*DB to notify an application when something "of interest" in the database changes. It is available in synchronous and asynchronous modes.

Deterministic Rule-Based SQL Optimizer

Cost-based SQL optimizers must collect sample data, generate statistics and analyze thousands -- sometimes hundreds of thousands -- of possible execution plans; these requirements make cost-based optimizers' operation CPU-intensive and unpredictable. In contrast, *eXtreme*SQL (*eXtreme*DB's SQL interface) uses a highly efficient and predictable rule-based optimizer.

Pattern Search

For flexibility in querying, *eXtreme*DB supports the use of wildcards to search tree index entries for single and multiple character matches.

Security Features

*eXtreme*DB protects your database. Cyclic Redundancy Check (CRC) on the database page level detects whether unauthorized modification to stored data has occurred, while RC4 encryption employs a user-provided cipher to prevent access or tampering. In addition to safeguarding sensitive data (in financial or defense systems, for example) page-level CRC can be used in devices such as portable audio players to help enforce digital rights management.

Binary Schema Evolution

*eXtreme*DB can save a database as a binary image and then restore it with a changed schema. This can accomplish design changes more quickly, using less memory and storage, than the alternative approach using XML import/export.

Remote Procedure Calls (RPCs)

*eXtreme*DB generates an interface that enables remote processes (i.e. on another network node) to read/update an *eXtreme*DB database.

Database Striping/Mirroring

*eXtreme*DB Fusion can exploit multi-disk (solid state or spinning) configurations with its support for RAID-like data striping and data mirroring. Striping accelerates performance when working with two or more disks by parallelizing access to data. Mirroring provides continuous backup by replicating data onto separate disks (if disk A fails, identical records are available from disk B).

XML Import/Export

*e*X*treme*DB's schema compiler, mcocomp, can generate XML interface functions for each class, providing the means to retrieve an object encoded as XML, to create or replace (update) an object in the database from the content of an XML string, to generate the XML schema for each class in the database in order to facilitate sharing data with other XML-enabled systems, and to accomplish simple *e*X*treme*DB schema evolution.

Kernel Mode Deployment

McObject pioneered the concept of an in-memory database that deploys in the operating system kernel. *e*X*treme*DB Kernel Mode leverages the high priority, zero-latency responsiveness afforded to kernel tasks—and meets the data sorting, access and retrieval needs of applications that place key functions there.

C/C++, SQL, Java APIs

*e*X*treme*DB provides the developer with multiple application programming interfaces (APIs). Access *e*X*treme*DB using a fast, native navigational API consisting of C/C++ functions. This is provided as both a type-safe and intuitive application-specific API (functions are generated based on the data design), and as a uniform data access (UDA) API for a consistent interface across all projects. *e*X*treme*DB's SQL implementation includes both an ODBC API for interoperability, as well as a more succinct and easier to use proprietary API. *e*X*treme*DB's Java Native Interface (JNI) offers the fastest possible DBMS solution in Java, and the ability to access *e*X*treme*DB while working entirely with plain old Java objects (POJOs).

Designed To Prevent Database Corruption

The *e*X*treme*DB native API is type-safe: errors in data-typing are caught at compile time, to eliminate database corruption. In addition, the *e*X*treme*DB runtime implements many verification traps and consistency checks. After application debugging, the optimized version of the *e*X*treme*DB runtime can be used, removing traps and internal checks, and restoring valuable clock cycles.

In addition, cyclic redundancy check is built into the backup-and-restore feature for in-memory databases, executing automatically when a file is loaded to ensure the database was written in its entirety when saved, and has not been corrupted.

B-Tree, R-Tree, Patricia Trie, KD-Tree and Hash Indexes

*e*X*treme*DB provides a wide range of database indexes, to boost application performance and minimize footprint. *e*X*treme*DB offers R-trees for geospatial data, Patricia tries for IP/telecom, KD-trees for multi-dimensional data and Query-by-Example (QBE), B-trees, hash indexes and more. For in-memory databases, rather than storing duplicate data, indexes contain only a reference to data, keeping memory requirements to an absolute minimum.

Wide Range of Supported Data Types

*e*X*treme*DB supports a wide range of data types -- including structures, arrays, vectors and BLOBs -- for maximum coding efficiency. Data can be stored in the same complex form in which it is used in the application, or as normalized relations.

Custom Collations

*e*X*treme*DB includes hooks that enable developers to provide a desired character sorting sequence (collation) for data stored as text, including collation that supports a particular language or combination of languages. A single application can support search and other text-processing functions in multiple languages.

Broad platform support

*e*X*treme*DB operates on multiple platforms including all major server and real-time operating systems (RTOSs), and can even run without an OS in "bare bones boards" configurations. It is designed to minimize reliance on OS-specific features, in order to simplify the task of porting to new operating systems.

Source code available

Available source code provides the ultimate transparency for developers using *e*X*treme*DB. Use it for porting, learning, or to add a needed function.

Proven Solution

Compared to self-developed (‘homegrown’) data management, *e*X*treme*DB offers a proven solution that slashes months from development, QA and ongoing support. Why take on the complexity and cost of developing a database management system from scratch, when an off-the-shelf solution meets your requirements?

Outstanding Developer Support

McObject technical support engineers are database and real time application experts who answer customer questions in detail. Whether embedded systems beginners with basic configuration questions or seasoned developers looking for hints on optimization customers get prompt, informative answers and continued follow-up to speed production and get the most from *e*X*treme*DB.