eXtremeDB®
In-Memory Database System

High performance, small footprint in-memory database system (IMDS) for embedded devices.

McObject®

"eXtremeDB gave us the performance and flexibility we required to manage the complex data in our applications."

-- Genesis Microchip

---

eXtremeDB, the real-time embedded database for devices that are eXtremely innovative

Overview

The *eXtreme*DB Database System is McObject's core product. It is designed for performance, with a strict memory-based architecture. Data is stored and manipulated exactly in the form used by the application, removing overheads of caching and translation. Typical read and write accesses are at the level of a few microseconds, or less. The engine is reentrant, allowing for multiple execution threads, with transactions supporting the ACID properties for data integrity.

The Runtime Environment

Accelerated Transactions. *eXtreme*DB slashes latency by storing data in main memory, eliminating the need for disk access, caching and other overhead of disk-based DBMSs. Its transaction managers are optimized for ultra-fast processing.

Tiny Footprint. A streamlined design delivers the maximum in performance and features with a code size of 200K or less! This makes it a powerful enhancement to intelligent devices with resource limits that, until now, ruled out the use of a database system.

Direct Data Access. By working with data directly in main memory, *eXtreme*DB eliminates 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.

Highly Scalable. Some real-time systems manage large data stores. The 64-bit *eXtreme*DB edition is proven in terabyte-plus deployments. Advanced memory management, and a Multi-Version Concurrency Control (MVCC) transaction manager, fully leverage multi-threading on multi-core systems.

No Data Translation. *eXtreme*DB stores data in the form used by the application. This eliminates translation tasks, such as mapping a C data element to a relational representation.

High Reliability. For data integrity, *eXtreme*DB transactions support the ACID properties, ensuring that operations grouped into transactions will complete together or the database will be rolled back to the pre-transaction state.

The Development Environment

Developers strive to produce readable, maintainable, efficient code in the shortest possible time. *eXtreme*DB includes several features that boost the developer's capabilities when integrating *eXtreme*DB in demanding real-time applications.

Incorporating third party software often means learning and adopting an API that does not completely fit an application. *eXtreme*DB's **native, project-specific API** for development in C/C++ ensures that each database operation in the API reflects the type of data being manipulated. Optional *eXtreme*SQL supports the widely used SQL standard, and a Java Native Interface (JNI) delivers the ease of working with "plain old Java objects" (POJOs).

McObject offers full source code, to give an in-depth understanding of *eXtreme*DB within an application.

eXtremeDB supports complex data types including structures, arrays, vectors and BLOBs.

eXtremeDB provides extremely efficient indexing for queries. Rather than storing duplicate data, indexes contain only a reference to data, keeping memory requirements to an absolute minimum. Supported indexes include:

- Hash indexes for exact match searches
- Tree indexes for pattern match, range retrieval and sorting
- R-tree indexes for geospatial searches
- KD-tree for spatial and Query-By-Example (QBE)
- Patricia trie indexes for networking & telecom
- Object-identifier references, for direct access
- Custom indexes

For application debugging, the *eXtreme*DB runtime includes **progressive error detection and consistency features.**

For development, the *eXtreme*DB runtime implements many verification traps and consistency checks. Then, when the application is debugged and consistently passes verification tests, developers can employ the optimized *eXtreme*DB runtime with fewer checks, to restore valuable clock cycles.

Precision Data Management

Additional Features

eXtremeDB's many extras help developers and application end-users get the most from the database.

- **Custom Collations**. Specify the character sorting sequence (collation) for text, including collations supporting more than one language.
- **Event Notifications**. Notifies an application when something "of interest" in the database changes. Synchronous and asynchronous modes.
- **Remote Procedure Call (RPC) Mechanism**. Framework enables remote processes (i.e. on another network node) to read/update an eXtremeDB database.
- **Security Features.** Page-level Cyclic Redundancy Check (CRC) detects unauthorized changes, while RC4 encryption blocks both tampering and unauthorized access.
- **Pattern Search**. Use wildcards to search tree index entries for single and multiple character matches.

Supported Platforms

Embedded Platforms:

- VxWorks 5.5, 6.x
- VxWorks 653 RTOS (for avionics)
- INTEGRITY OS
- QNX 6.x
- Various Real-Time Linux distributions
- Lynx OS
- RTXC Quadros, RTXC 3.2
- Microsoft Windows Embedded
- Bare bones boards (no operating system required)

Development environments

- gnu toolchain (gcc 2.96 and higher)
- Tornado 2.0 and 2.2 (GNU and Diab compilers)
- QNX Momentics IDE (C, C++, Embedded C++)
- Eclipse
- XCode
- GreenHills Multi
- Microsoft Visual Studio (C/C++, .NET)

Server and Desktop Platforms:

- Sun Solaris 8, 9 and 10
- HP-UX 11.x
- Linux distributions
- Classic Windows platforms (7/NT/2000/XP/Vista)

Database Specifications

Maximum database size, 32-bit:	3 gigabytes
Maximum database size, 64-bit:	18 exabytes
Maximum classes per database:	32,767
Maximum indexes per database:	32,767
Maximum fields per class:	32,767
Maximum fields per index:	32,767
Maximum elements per vector:	32,767
Memory requirements:	As little as 200K
Maximum simultaneous connections per database:	configurable
Maximum databases open simultaneously:	configurable

Supported Data types

- 1, 2, 4, 8-byte signed/unsigned integers
- float, double
- date, time, timestamp
- Char (fixed length)
- String (variable length)
- Boolean
- Fixed-size array
- Variable-length vector
- Structs (embedded to any depth)
- BLOB
- Autoid (auto-increment)
- User-defined object-id and references
- Unicode

The eXtremeDB Product Family

Building on the edition, combining in-memory and persistent storage in a single hybrid database, McObject offers eXtremeDB editions to meet specialized needs.

- **eXtremeDB-64** - 64-bit edition enables processing of very large databases
- **eXtremeDB High Availability** for the highest level of database fault-tolerance
- **eXtremeDB Transaction Logging** provides recovery capabilities via a highly configurable logging process
- **eXtremeSQL** adds SQL and Open Database Connectivity (ODBC) support to eXtremeDB