*eXtreme*DB®
C# Native Interface

The fastest database solution in C#/.NET.

Access eXtremeDB's powerful features entirely from within C# code, working entirely with "plain old" C# objects.

*eXtreme*DB, the embedded database for applications and devices that are eXtremely innovative.

Overview

McObject provides a C# Native Interface for the *eXtreme*DB In-Memory Database System (IMDS) and related product family, offering the fastest possible DBMS solution in C#. This application programming interface (API) reduces development time and slashes latency in .NET real-time applications (e.g., finance, e-commerce, and social networks), while affording the ease of working with "plain old" C# objects.

The *eXtreme*DB C# Native Interface delivers the speed of compiled C/C++ for performance-intensive database operations, with the convenience of using a familiar language. *eXtreme*DB's breakthrough performance stems from a streamlined architecture that manages data entirely in main memory, eliminating file I/O, cache management and other overhead found in database systems that incorporate disk storage.

```
class Address
{
    public String city;
    public String street;
    public String phone;
    public String fax;

    public Address(String city, String
      street, String phone, String fax) {
        this.city = city;
        this.street = street;
        this.phone = phone;
        this.fax = fax;
    }
}

class Employee
{
    public String name;
    public int age;
    public long salary;
    public long companyID;
    Address address;
    public byte[] photo;
}
```

C# Class Declarations

Figure 1. Class definitions in a C# application.

With the *eXtreme*DB C# interface, developers work with the database entirely from within C#. There is no external data definition language, and no requirement to compile a database schema.

Instead, the API uses C#'s reflection capability to discover database classes and their fields defined in the application, via class definition syntax (see Figures 1 and 2). This streamlines coding, enabling developers to work entirely with C# objects.

Optionally, the C# class definitions can be used to generate a schema that enables C/C++ and C# programs to use the same database.

```
class Address
{
    [Dimension(20)]
    public String city;

    [Dimension(60)]
    public String street;

    [Dimension(20)]
    public String phone;

    [Dimension(20)]
    public String fax;

    public Address(String city, String
      street, String phone, String fax) {
        this.city = city;
        this.street = street;
        this.phone = phone;
        this.fax = fax;
    }

    // default constructor is needed for
    // eXtremeDB
    public Address() {}
}

[Persistent(List=true)]
class Employee
{
    [Indexable(Unique=true)];
    public String name;

    public int age;

    [Indexable(Unique=false)];
    public long salary;

    [Indexable(Unique=false)];
    public long companyID;

    [Optional];
    Address address;

    [Blob];
    public byte[] photo;
}
```

With the *eXtreme*DB C# API, C# class definitions use attributes (defined using brackets) to indicate database characteristics.

Figure 2. *eXtreme*DB's C# API uses reflection to discover classes, their fields and other characteristics at run-time.

Which DBMS in C# and .NET?

What are the database options in C# when an application calls for the highest possible responsiveness? While not an exhaustive list of options, consider the following:

SQL relational database systems. SQL RDBMSs' biggest drawback with C# is the "impedance mismatch" that occurs when going between relational structures and SQL (a set-oriented database access language), on one end, and the object-oriented C# language, on the other. The need to map between object and relational technologies consumes CPU cycles and hurts performance.

"Pure" C# database systems. Embedded object-oriented databases written in C# deliver improved performance by eliminating impedance mismatch. However, as a language that is interpreted at run-time, C#'s performance can't compete with compiled C and C++.

***eXtreme*DB w/ native C# API**. *eXtreme*DB's C# Native Interface uses C# class definitions to define the database schema. Since it relies on the programming language's own syntax, there is (by definition) no impedance mismatch. And all sorting, retrieval, storage and other DBMS functions execute in fast, compiled C code rather than C#.

McObject LLC
33309 1st Way South
Suite A-208
Federal Way, WA 98003

Phone:+1 425 888 8505
info@mcobject.com
www.mcobject.com

LINQ Support

The native C# API adds support in *eXtreme*DB for Language Integrated Query (LINQ), a Microsoft .NET Framework component that brings SQL-like data querying capabilities to .NET languages. In LINQ, developers using the *eXtreme*DB C# interface have a succinct, extensible and increasingly popular approach to database querying, with additional benefits including protection against software bugs caused by data-typing errors.

Targeting Real-Time Enterprise Applications

The C# API enhances *eXtreme*DB's value by leveraging .NET's familiarity among enterprise developers, providing a powerful database solution for embedded and Web-based systems in areas including finance, e-commerce and highly scalable social networking applications. Developers can deploy the 64-bit edition of *eXtreme*DB (*eXtreme*DB-64) as an in-memory front-end for an enterprise relational database management system (RDBMS). This dramatically accelerates data management, while providing benefits missing in traditional object caching and NoSQL solutions, such as efficient storage and safeguards on data integrity.

Compared to object caching solutions, *eXtreme*DB offers persistence: cached data can be easily recovered, through features such as transaction logging; a clustered system architecture that builds in redundancy (*eXtreme*DB Cluster); database replication (*eXtreme*DB High Availability), and the optional disk- or flash memory-based storage provided by McObject's *eXtreme*DB hybrid database technology.

McObject's *eXtreme*DB

*eXtreme*DB offers the benefits of a proven database management system used world-wide in millions of applications.

- Speed: core in-memory database system (IMDS) architecture delivers micro-second transactions even on modest hardware
- Optional SQL, XML, Java and type-safe C/C++ interfaces
- Multi-version concurrency control (MVCC) transaction manager and advanced memory management fully leverage multi-threaded, multi-core systems
- Transactions support the ACID (Atomic, Consistent, Isolated and Durable) properties, to safeguard data integrity
- Available source code
- High Availability and Cluster editions
- Transaction Logging edition supports database recovery, Fusion edition supports hybrid in-memory & on-disk storage
- 64-bit edition supports very large databases, with linear performance gains proven in benchmark managing a 15.54 billion row, 1.17 TB in-memory database on 160 cores.

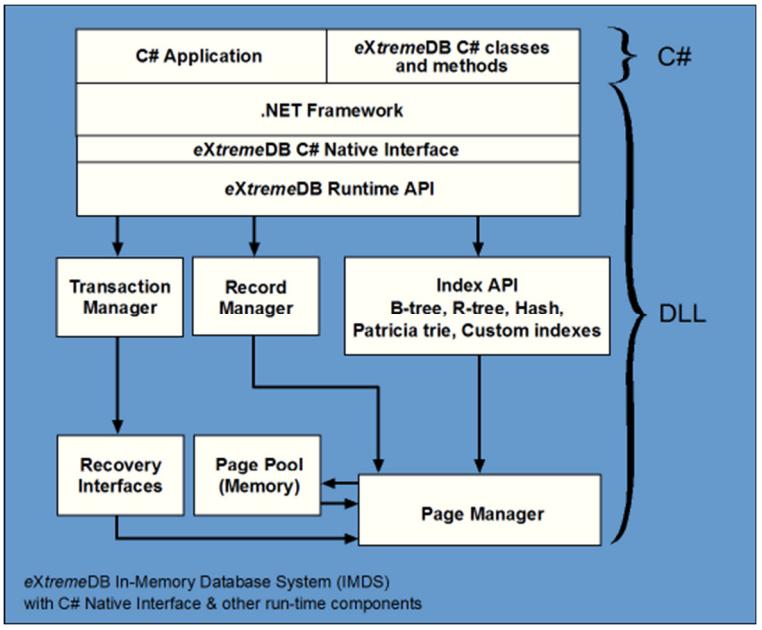

Figure 3. *eXtreme*DB with its C# Native Interface delivers the fastest possible database solution in C#/.NET.

Highly efficient indexing

*eXtreme*DB provides diverse indexes, optimized for a variety of application and data types. Supported indexes include:

- Hash indexes for exact match searches
- Tree indexes for pattern match, range retrieval and sorting
- R-tree indexes for geospatial searches
- Patricia trie indexes for networking & telecom
- Object-identifier references for direct access
- Custom indexes

C# Attributes

*eXtreme*DB JNI uses C# attributes to define database characteristics in application code. Recognized attributes include:

- [Blob] - Uses *eXtreme*DB BLOB type to store a byte array
- [Dimension] - Specifies dimension for fixed size arrays (char, scalar)
- [Encoding] - Specifies encoding of string fields
- [Indexable] - Includes a given field in an index
- [Indexes] - Defines a compound (multi-field) index
- [Optional] - Optional structure field, for example:
  [Optional]
  Address address;
- [Persistent] - Marks class as stored in the database and provides attributes of the class

McObject

McObject LLC
33309 1st Way South
Suite A-208
Federal Way, WA 98003

Phone:+1 425 888 8505
info@mcobject.com
www.mcobject.com