# What's New -
# Raima Database Manager 12.0

*RDM Embedded is a high-performance database management system that is optimized for operating systems commonly used within the embedded market. The database engine has been developed to fully utilize multi-core processors, run with minimal memory, and support both in-memory and on-disk storage. It provides a subset of the ANSI/ISO standard SQL that is suitable for running on a wide variety of computers and embedded operating systems which may have limited resources.*

**Key Features:**

- Database Cursors
- Shared Memory Protocol
- Larger Key Size Limit
- Memory Limitation
- Enhanced SQL Optimization Support
- Bulk Insert API
- New Data Types
- Non-Repeatable Reads
- Encryption
- Log Notification API

## Database Cursors

This release introduces a modernized C interface to the RDM engine through database cursors. The addition of cursors to the standard library allows for a natural traversal of records in the database. Cursors facilitate subsequent processing in conjunction with the traversal, such as retrieval, addition and removal of database records.

## Shared Memory Protocol

RDM 12.0 greatly improves performance with remote clients through the addition of shared memory as a transport for communication between the application and the tfserver. In this scenario, a shared memory buffer is established for each connection. This buffer along with some shared mutexes allows data to be transferred between the application and the tfserver on the same machine much quicker than the same communication could occur via TCP/IP.

## Larger Key Size Limit

RDM 12.0 eliminates the legacy limit to key lengths of 240 bytes. With this version developers control the key size through their schema definition.

## Memory Limitation

The runtime library will allocate all of the memory it requires for cache and other necessary storage. On workstation operating systems like Windows and Linux, this rarely causes an over commitment of memory. However, on resource-constrained computers, it is necessary to limit or preallocate all memory that the database process will be allowed to use. This is now facilitated through a new function that specifies a limit, and an optional pointer to preallocated memory.

**Enhanced SQL Optimization Support** This release has enhanced SQL functionality to improve SQL performance through the inclusion of data distribution statistics which will allow the SQL query optimizer to make more intelligent execution plan choices and by allowing the declaration of rowid (i.e., database address) primary and foreign keys which will allow direct access to individual rows of a table without having to incur the overhead of an index.

## Bulk Insert API

In v12.0 the Bulk Insert API is a new performance optimization API to efficiently insert large quantities of data at the Core level.

## New Data Types

RDM v12.0 adds three new data types which, while previously would be stored as binary data, are now stored natively and can be utilized as an index. While some of the new data types exist in RDM's SQL module, RDM v12.0 reorganizes data types in the Core engine so data types can consistently be used across Core, SQL or the other interfaces.

The new data types are as follows:

⇒ **Date/Time/Timestamp**
Binary-code decimal is a standard database representation for financial applications.

⇒ **BCD**
The Decimal type represents a fixed-point real number. We will use the Binary-Coded Decimal (BCD) encoding that RDM Server uses for our support of the Decimal type. A BCD has two modes - packed and unpacked. All application interaction with the BCD type (including SQL) will be unpacked, but when we store the BCD value in the database we will pack it to reduce the space utilized.

⇒ **GUID**
The UUID types represent 128-bit unique identifiers.

**Try it!**

Download a trial version:

raima.com/downloads

## Non-Repeatable Reads

Non-repeatable reads are implemented in RDM 12.0 as safe, lockless reads. A lockless reader always views committed data, but the data may be in the process of being changed by concurrent transactions. Where this is an acceptable degree of isolation, it reduces concurrency and locking conflicts and significantly improves performance by not blocking readers during transaction activity.

## Encryption

RDM now offers database encryption as another layer to the application developer and IT directors defense-in-depth toolkit. RDM optionally supports encryption of the database files using the Rijndael/AES algorithm for encryption or decryption. The algorithm supports the use of 128, 192 or 256 bit keys.

## Log  Notification API

The Log Notification API is a set of functions which enable the programmer the ability to implement applications that monitor RDM databases for changes to specific tables. The intent is to provide the programmer an asynchronous alternative to triggers. The replication system distributes change log files to subscribers. A subscriber can be a customized C/C++ program that uses the new version 12.0 subscriber API that permits requesting log files, opening and scanning log files, and interpreting the log contents. Since the contents may include pre and post images, they may be used for any purpose. One such purpose is change notifications. A customized program using this API will be able to immediately (upon transaction commits) receive and scan logs that identify specific changes made to the database, and act upon them accordingly.

## Industry Partners:

Green Hills SOFTWARE
WIND RIVER
QNX
QNX SOFTWARE SYSTEMS
montavista
LYNUXWORKS™
Windows® Embedded Partner
MITSUBISHI ELECTRIC
NATIONAL INSTRUMENTS
OPEN VIRTUALIZATION ALLIANCE

## Want to know more?

Please visit  our website for the latest news, product downloads and  documentation: www.raima.com  or follow us on Twitter: https://twitter.com/RaimaDB

Headquarters: 720 Third Avenue Suite 1100, Seattle, WA 98104, USA T: +1 206 748 5300
Europe: Stubbings House, Henley Road, Maidenhead, UK SL6 6QLT: +44 1628 826 800

RAIMA