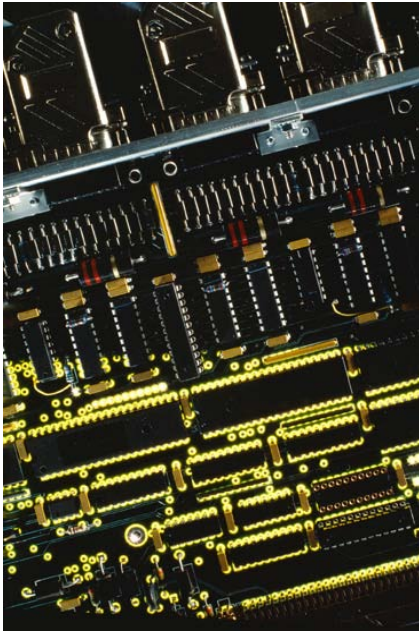


# Mimer SQL Real-Time Edition – White Paper

- Dag Nyström, Product Manager Mimer SQL Real-Time Edition

Mimer SQL Real-Time Edition is a predictable, scalable and zero maintenance database management system for embedded real-time systems, such as automotive, process-control, and robotic applications. By combining predictable hard real-time and non-real-time database access, integrated data management solutions impossible to implement with other DBMS products on the market are made possible.



*Embedded hard real-time databases are needed in many solutions for automotive, process industry and various embedded devices.*

Mimer SQL Real-Time Edition allows data management in embedded real-time applications to be taken to a higher level while still maintaining real-time predictability and fine-grained control.

The uniqueness of Mimer SQL Real-Time Edition comes from its real-time access methods featuring fine-grained in-memory real-time data access with ultra-short, fully deterministic response-times. These real-time algorithms stems from well-proven academic research results gained from years of collaboration with academia (Mälardalen Real-Time Research Center, <http://www.mrtc.mdh.se>).

In Mimer SQL Real-Time Edition, real-time and non-real-time data can be safely modeled and shared by both time-critical and administrative processes, such as diagnostics, statistics and maintenance. A common approach for sharing data between real-time and non-real-time processes is to manage real-time data separately using in-house, specialized data storages. These storage solutions often result in time-consuming development and maintenance, poor system scalability and longer time to market. By using Mimer SQL Real-Time Edition, all data in the system can be modeled in one uniform data storage using standard modeling techniques. Of course, the other strong arguments for using Mimer SQL, such as a footprint of 300kb or less, zero maintenance, multi-user data access, and full support for ISO/ANSI SQL still remains.

## Real-Time Access using Database Pointers

Mimer SQL Real-Time Edition utilizes a unique real-time access concept called *database pointers*, which allow predictable, fine-grained access to data within the database. Database pointers are in-memory application variables that can be initialized to point at individual or sets of data elements in the database. At run-time, these pointers can be read and updated with predictable response-times that are 1000 times faster, or more, than traditional database calls. The API allows database pointers to be dynamically created and deleted during run-time, but could be customized for static use if necessary. Access to data using database pointers need no index-lookups or statement compiling, instead pointers locate the data immediately. The integrity of the database is retained due to that the database pointer API ensures type checking and concurrency. To put it simple; “*Database pointers combine the performance and predictability of shared variables with the power of an SQL database.*”

It is safe to say that Mimer SQL Real-Time Edition once and for all invalidates the misconception that databases are slow, bulky and unpredictable pieces of software.

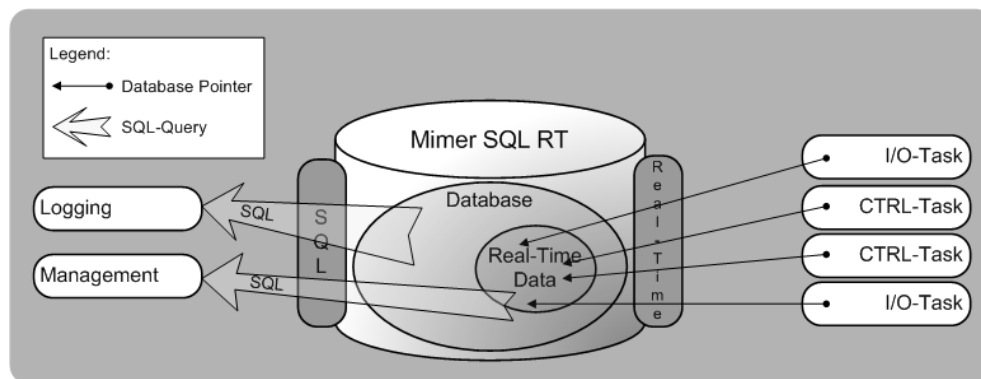


Figure 1: Overview of the Mimer SQL Real-Time Edition API's

## Safe Sharing of Real-Time Data

The true benefit of Mimer SQL Real-Time Edition is its ability to safely share data between both real-time and administrative applications. In Mimer SQL Real-Time Edition, all application data can be stored in one single database, using any relational modeling technique to optimize performance and memory consumption. After that, data that is needed by (hard) real-time applications in the system can be elevated to become real-time data. This implies that the database manager will create an in-memory copy of the data and enable it for real-time access. This data can then be accessed using database pointers for guaranteed real-time predictability, but can also be accessed using standard SQL query management just as any data in the database. A special real-time concurrency-control algorithm ensures that the real-time data access is never blocked or aborted because of conflicts with other database accesses. Furthermore, Mimer SQL Real-Time Edition adopts the scheduling and critical section policies from the underlying real-time operating system, i.e., if your operating system uses the *immediate inheritance protocol*, so will the database pointers. By limiting the maximum blocking times of data access, standard response-time analysis methods can be used.

## Real-Time Data and Event Logging

Since most applications today involve some kind of logging and diagnostics, Mimer SQL Real-Time Edition directly supports this. By introducing *multirow and multicolumn database pointers* it has never been easier to record logs and events, as well as handle producer-consumer type applications. By using multirow database pointers, all (or some if desired) records in a table can be circularly accessed. Before each write operation on the database pointer, the pointer is redirected to the same column in the next record of the table. Combining this with multicolumn database pointers, several columns can be written in one operation. In Figure 2, a combination of a multirow and multicolumn database pointer is used to store warnings and errors in a log-table. When the table is full, the pointer continues to the first record again, overwriting that log. During run-time, administrative applications can directly access the entire table using SQL queries under transactional control. Of course, the predictable real-time properties of database pointers are retained in both multirow and multicolumn database pointers.

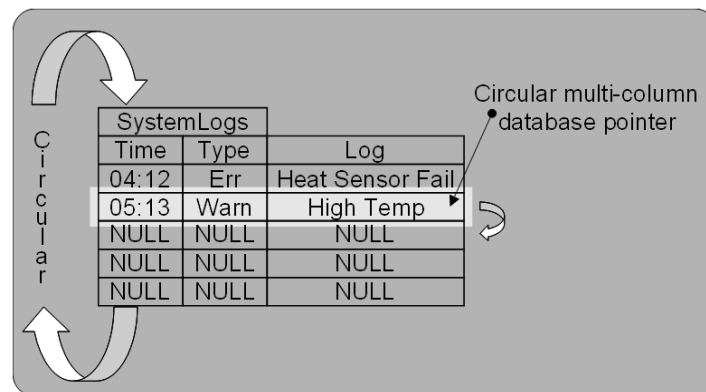


Figure 2: A multirow multicolumn database pointer used in a logging table

## Retained Real-Time Application Control

Another common misconception about databases is that, apart from databases being bulky, you give up control of the application to the database. With Mimer SQL Real-Time Edition, all real-time operations are performed entirely in primary memory using the caller's thread of execution instead of using a client/server architecture. Thus, real-time operations will never yield a context switch for data accesses. If persistency of all or some data is needed, you decide when or if to flush real-time data to secondary storage such as a hard disk drive or a flash memory. Flushing data will, of course, never block real-time data access. To be able to select flushing strategy for individual real-time data elements greatly improves performance, since much real-time data might never need to be saved to persistent storage.

## Achieve True Application Interoperability

Mimer SQL Real-Time Edition has inherited all the functionality from the other members of the Mimer SQL product family, which for example include 100% conformance to international SQL standards. This implies that Mimer SQL Real-Time Edition will work seamlessly with any standard database tool, as well as making porting of your existing applications easier.

All products in the Mimer SQL family are characterized by zero maintenance, high performance and full multilingual support. This gives you a very low total cost of ownership (TCO) for your database solution, disregarding if you distribute a low priced - high volume product or if you deploy a high priced – low volume product.

## Customization to Your Needs

Mimer Information Technology AB has a long experience of customization and porting their products for various requirements and environments. All Mimer's products stem from the same code-base and can be tailored for systems ranging from tiny real-time control-nodes to resource abundant enterprise servers.

Mimer SQL Real-Time Edition can be made available for a range of operating systems, both general operating systems and RTOS's. Evaluation versions for MS Windows, Windows CE, Linux, and VxWorks are currently available upon request. For other platforms, please contact us.

## Mimer SQL Real-Time Performance

Data type	Intel Core2Duo 2.33GHz		Intel XScale, 533 MHz	
	Read	Write	Read	Write
Integer (32bit)	0.14µs	0.11µs	1.6µs	1.2µs
Integer (16bit)	0.10µs	0.10µs	1.1µs	1.0µs
String (100bytes)	0.15µs	0.36µs	1.7µs	2.6µs
Timestamp	0.18µs	0.19µs	2.1 µs	2.3 µs

## Facts About Mimer SQL Real-Time Edition

### Footprint

Full Real-Time Server (Engine)	<20 kb + Ordinary Mimer SQL Engine server (~3.5 Mb)
Embedded Real-Time Server	<10 kb + Ordinary Mimer SQL Embedded server (from ~300kb)
Real-Time Client API	<35 kb + communication package (configurable 2kb – 160kb)

### Datatypes (can be extended on demand)

Integer (32bit), Integer (16bit), String, Timestamp

### Persistence

In-Memory with option to flush to persistent storage

### Real-time operations

<b>Single value database pointers</b>	Read/Write individual database values.  One database pointer per value
<b>Multicolumn database pointers</b>	Atomic Read/Write of multiple columns (attributes) of a database record.  One database pointer per record
<b>Multirow database pointers</b>	Read/Write of database value in multiple database records of a table.  Writes are performed in sequential order  Reads are performed in sequential, backwards or arbitrary order.  Can be combined with multicolumn functionality  One database pointer per table
<b>Database pointer flushing</b>	Flushing of any database pointer to persistent storage.  Will not block real-time operations.

For further information, contact the author:

Dag Nyström,  
Product Manager, Mimer SQL Real-Time Edition  
Mimer Information Technology AB  
Uppsala, Sweden  
Mail: dag.nystrom@mimer.com



or our sales department:  
Mimer Information Technology AB  
Uppsala, Sweden  
mail: info@mimer.com  
Phone: +46 (0)18 780 92 00  
Web: <http://www.mimer.com>