



Mimer SQL

Getting Started on QNX

Version 11.0

Mimer SQL, Getting Started on QNX, Version 11.0, March 2022
© Copyright Mimer Information Technology AB

The contents of this manual may be printed in limited quantities for use at a Mimer SQL installation site. No parts of the manual may be reproduced for sale to a third party.
Information in this document is subject to change without notice. All registered names, product names and trademarks of other companies mentioned in this documentation are used for identification purposes only and are acknowledged as the property of the respective company. Companies, names and data used in examples herein are fictitious unless otherwise noted.

Produced and published by Mimer Information Technology AB, Uppsala, Sweden.

Mimer SQL Web Sites:
<https://developer.mimer.com>
<https://www.mimer.com>

Contents

| | |
|---|-----------|
| | i |
| Getting Started | 1 |
| Introduction | 1 |
| Documentation | 2 |
| Useful links | 2 |
| Target board support..... | 2 |
| Known restrictions..... | 2 |
| Installing Mimer SQL | 3 |
| The distribution package | 3 |
| Unpacking the Mimer SQL for QNX package | 3 |
| Package contents | 3 |
| Programs and libraries provided | 3 |
| License setup | 4 |
| The Database Server | 5 |
| Database Setup and configuration | 5 |
| Initial database - creating system databank files | 5 |
| Registering the database | 5 |
| Database server configuration | 6 |
| Start/Stop Scripts | 7 |
| Automatic start | 7 |
| Shutdown | 7 |
| In-memory Database Server | 8 |
| Time and Date configuration | 8 |
| Development and Example Environments | 9 |
| Developing the database | 9 |
| Locally on the QNX board..... | 9 |
| Remote access..... | 10 |
| On Linux host | 10 |
| Build and execute | 10 |
| Add the Mimer SQL software to the target build environment..... | 10 |

| | |
|---|----|
| Target build procedure setup..... | 11 |
| Configuring the image with Mimer SQL | 11 |
| Build the target image..... | 11 |
| Boot the created image and start the database | 11 |

| | |
|--------------------|-----------|
| Index | 13 |
|--------------------|-----------|

Chapter 1

Getting Started

Welcome to Mimer SQL. This document describes how to install and set-up Mimer SQL on the QNX platform - how to use the installation package for Mimer SQL on QNX.

To get the most out of this document, you should be familiar with your QNX environment.

Introduction

Mimer SQL is an advanced relational database management system (RDBMS) developed by Mimer Information Technology AB.

The main characteristics of Mimer SQL are zero maintenance, small footprint and high performance. These are based on a number of unique technical solutions to handle some of the more complicated functionality that a database management system must provide.

For example, Mimer SQL provides a solution to the problem of allowing simultaneous access to the database without the danger of a deadlock occurring. This greatly simplifies database management and allows truly scalable performance, even during heavy system load.

Another significant technical innovation is the data storage mechanism, which is constantly optimized for the highest possible performance and ensures that no manual reorganization of the database is ever needed.

Mimer SQL offers a uniquely scalable and portable solution, including multi-core support.

The product is available on a wide range of platforms, from small embedded and handheld devices running for example Android and real-time operating systems like QNX, to workgroup and enterprise servers running Linux, Windows, macOS and OpenVMS. This makes Mimer SQL ideally suited for open environments where things like interoperability, portability and small footprint are important.

The database management language Mimer SQL (Structured Query Language) is compatible in all essential features with the currently accepted SQL standards.

Documentation

Detailed information can be found in the Mimer SQL Documentation Set, located at the Mimer SQL developer site (<https://developer.mimer.com/documentation>,) and on the Mimer SQL Documentation Site as https://docs.mimer.com/MimerSqlManual/latest_mimersql.html. The Mimer SQL documentation set includes the following:

- *SQL Reference Manual*
- *Programmer's Manual*
- *System Management Handbook*
- *User's Manual*

For an introduction to the Linux host system, see the guide *Mimer SQL Getting Started on Linux*, located at the Mimer SQL developer site (<https://developer.mimer.com/documentation>,) and on the Mimer SQL Documentation Site as https://docs.mimer.com/MimerLinuxGuide/latest_mimerlinux.html.

Useful links

The Mimer SQL Developer Site, <https://developer.mimer.com>, contains lots of useful information, like news, howto and feature articles.

All manuals for Mimer SQL are found at <https://docs.mimer.com>.

For general information on Mimer SQL, please see <https://www.mimer.com>.

Documentation for QNX 7.0: <https://www.qnx.com/developers/docs/7.0.0/>.

Documentation for QNX 7.1: <https://www.qnx.com/developers/docs/7.1/>.

How to install QNX on a Pi4 could be found here: <https://carleton.ca/rcs/qnx/installing-qnx-on-raspberry-pi-4/>.

Target board support

Currently the Mimer SQL package for QNX is integrated and verified on the following target boards:

- NXP S32V board (QNX 7.0)
- Renesas H3 board
- Raspberry Pi4

The instructions given in this guide are mainly based on the integration made towards these platforms.

Known restrictions

- Currently the F2FS file system is unsupported:

Chapter 2

Installing Mimer SQL

The distribution package

Unpacking the Mimer SQL for QNX package

The Mimer SQL for QNX package is provided as a compressed tar file. It is unpacked as in the following example command:

```
# tar xvzf mimersql_1106_qnx710_aarch64le-37595.tgz
```

The extracted files are placed in a directory named `mimersql` plus the version number, e.g. `mimersql_1106C`.

Package contents

The package contains the following:

- `./arm` - contains pre-built programs and libraries for BSP/arm.
- `./examples` - contains a few examples on how to use the MimerAPI for database access.
- `./include` - contains header files for integration.
- `./misc` - contains miscellaneous files, like the license key file.
- `./x86` - contains pre-built programs and libraries for Emulator/x86 (still not complete).
- `./README` - a readme file.

Programs and libraries provided

The `./arm` and `./x86` directories contain programs used to manage the Mimer SQL database environment. The programs provided is a subset of programs that can be seen in a standard installation of Mimer SQL on Linux. Currently the following programs are included:

- `bsql` - Command line SQL interpreter tool.
- `dbc` - Used to verify the consistency of a databank file.

- `mimcontrol` - Used to manage the Mimer SQL database server program.
- `mimexper` - The standard Mimer SQL database server program.
- `mimhosts` - Used to manage the database registration.
- `miminfo` - Used to analyze a running database server.
- `miminm` - The in-memory Mimer SQL database server program.
- `mimlicense` - Used to manage the license key.
- `sdbgen` - Used to generate the initial database, i.e., system databank files.

These files are located under subfolder `./bin`.

The `./arm` and `./x86` directories also contain libraries for linking QNX applications accessing Mimer SQL databases. Currently the Mimer SQL C API (MimerAPI) is supported:

- `libmimerapi.so` - The Mimer API link library to connect to Mimer SQL.

This file is located under the subfolder named `./lib`.

License setup

To be able to start a Mimer SQL database server, a license key must be in place. The license keys are stored in the `/etc/mimerkey` file, administered using the `mimlicense` program.

For details, see the section *Mimer SQL License Key* in the Mimer SQL Documentation Set.

The QNX board needs to have the license added in the build script. The `MIMER_KEYFILE` variable could be set to use a different file than `/etc/mimerkey`, like, for example (for a file residing in the current user home directory):

```
export MIMER_KEYFILE=~/.my_keyfile
```

Chapter 3

The Database Server

Database Setup and configuration

Initial database - creating system databank files

The initial creation of the database can be done directly on the QNX board, using the provided `sdbgen` program, or the database can be created on a host Linux system, and later on be moved to the target board.

Registering the database

To be able to run a Mimer SQL database server properly, the database name needs to be registered as a LOCAL database in the `/etc/sqlhosts` file, or in a file defined by the environment variable `MIMER_SQLHOSTS`.

The database registry setup file is created via the build script for QNX, so the setup for the database needs to be prepared in advance.

For details in this matter, see the section *The SQLHOSTS File* in the *Mimer SQL Documentation Set*.

Networking

If a database is to be accessed over TCP/IP, settings for port usage must be defined in the `multidefs` file for the database, see *Database server configuration* on page 6. This is the definition telling which port the database server should listen to, the one that will be used for TCP/IP communication. Default is the port number 1360, which is registered to Mimer SQL.

On the client side, for example on a Linux system, the QNX database should be registered as a REMOTE database. For an example QNX database named 'mimerdb', listening to port 1360 and operating on the IP-address 192.168.250.87, the client side REMOTE `sqlhosts` entry may look as follows:

```
mimerdb          192.168.250.87      tcp          ''          1360
```

This database can now be accessed from the client with, for example, `bsql` as follows:

```
# bsql mimerdb
```

Database server configuration

The configuration file for an installed Mimer SQL database server is named `multidefs`, and is located in the database home directory. The settings there can be modified after the database is created, and will be taken into account at the next server startup.

For more information in this area, see the section *The MULTIDEFS Parameter File* in the *Mimer SQL Getting Started on Linux guide*.

The following is an example of a default `multidefs` parameter file for QNX:

```
-- Mimer SQL version 11.0.5B Beta Test parameters generated 1970-01-01 00:01
Databanks          100          # Max # of databanks (20-1000)
Tables             4000         # Max # of tables (500-1000000)
ActTrans           20000        # Max # of active trans (500-1000000)
SQLPool            1000         # Initial SQLPool (400-8388607 kb)
RequestThreads     2           # # of request threads (1-100)
BackgroundThreads  3           # # of background threads (1-100)
TcFlushThreads     0           # # of t-cache flush threads (0-20)
Users              50          # Max # of logged in users (1-5000)
DBCCheck           1           # DB check, 0=index, 1=all, 2=immediate,
                          3=im. index, 4=im. all (0-4)
Pages4K            1000        # # of 4K bufferpool pages (11-2147480000)
Pages32K           400         # # of 32K bufferpool pages (7-2147480000)
Pages128K          56         # # of 128K bufferpool pages (0-
                          2147480000)
DelayedCommit      0           # Delayed commit, 0=Off 1=On
                          2=Disabled (0-2)
DelayedCommitTimeout 100       # Delayed commit timeout in milliseconds
                          (0-60000)
GroupCommitTimeout 2           # Group commit timeout in milliseconds
                          (0-20)
Oper               .           # Receivers for messages
DumpPath           .           # Path for dump directory
TCPPort            1360        # TCP/IP port
MaxSQLPool         128000       # SQLPool max size (2400-16777215 kb)
NetworkEncryption  1           # Client/server encryption, 0=None
                          1=Optional, 2=Required (0-2)
MemLock            0           # Lock bpool in memory, 0=No 1=Yes (0-1)
MiniDump           1           # Small bufferpool dump (no page content),
                          0=No 1=Yes (0-1)
BackgroundPriority  0           # Thread priority, 0=Off, 1=On (0-1)
AutoStart          1           # Autostart, 0=No, 1=Yes (0-1)
DumpScript         ./dumper.sh %p # Dump Script
IOQueue            128         # Max # of concurrent I/O requests
                          (0-65535)
ServerType         3           # Server type: 3=mimexper, 7=miminm (3-9)
```

Start/Stop Scripts

It is convenient to have the Mimer SQL database server start and stop handling connected to the target board boot and shutdown procedures. This way the database automatically starts when the board is booted, and a controlled termination of the database system will take place when QNX is taken down. (If the database is brought down carelessly, a database check may take place at the following startup.)

Automatic start

In the build script example `./arm/scripts/mimer.build`, the following section shows how to add a start-up procedure for a Mimer SQL server:

```
# Start Mimer
/etc/mimer-start.sh={
#!/bin/sh
# Setting up something to use for writing and reading database files.
# /dev/sd0t11 could have another adress or be another media than a sd-card.
mount -t dos /dev/sd0t11 /mimer
echo "Starting mimerdb..."
mimcontrol -s mimerdb
}
```

This command setup needs to run during boot, suitably at the end of the boot. For example, add the following line close to the end of the boot script (most commonly named `[+script] .script`):

```
# sh /etc/mimer-start.sh
```

In the scripts for specific boards, located in the `./arm/scripts/` folder, the start-up script statement is added. It will need a database to be created. Also, there is a line as follows that will need an updated to reflect the media the device is using:

```
# mount -t dos /dev/sd0t11 /mimer
```

Shutdown

It is important that the Mimer SQL database server is taken down properly to avoid a database check when starting up again. Therefore, the `mimcontrol -t` command should be used whenever suitable. This also goes for the situation when the QNX board is brought down. During the shutdown procedure, the time between the SIGTERM and SIGKILL signals are sent to the Mimer SQL database server process, the gap is not enough for a proper shutdown to be made. The following solutions have been evaluated for that purpose:

- **Set export SIGKILL_TIMEOUT=300**

The value 300 might not be the appropriate value for every database, depending on what kind of data and the amount of data stored, but it may be a fair value to start with.

- **Add a customized procedure for shutdown**

Add `mimcontrol -t` command to the shutdown script used for the device, or create a new separate script that runs `mimcontrol -t` before running the shutdown procedure.

- **Change the DBCheck option in the multidefs configuration file**
Select an option that reflects the desired behavior at system restart. If the database needs to be quickly in operation after a reboot, set the `DBCheck` flag to 0, to only verify the index pages.

In-memory Database Server

For systems where a relational database and extreme performance is looked for, the Mimer SQL In-memory database server is the choice. This database server works in memory only, thus providing a huge throughput.

To save a database state to continue from at a later stage, an online backup can be executed (see the Online Backup section in the System Management Handbook part of the Mimer SQL Documentation Set). An online backup will, while the system is running, produce a complete and consistent database file setup written to disk, i.e. a set of files that the in-memory database server can use to start from at the next start-up.

Please note that because all data is stored and managed exclusively in main memory, all data will be lost when the server is stopped, or upon a process or server failure. Thus, it is recommended having a procedure of doing Mimer SQL Online Backups if start-up data should be maintained.

Time and Date configuration

If the time and date are not set on the QNX device, time related functionality will not be available. However, the database will still be usable.

Chapter 4

Development and Example Environments

Developing the database

There are several possible ways to access and develop the database structure and contents, as described in the following sections.

Locally on the QNX board

The database can be accessed locally on the QNX board, for example, using the `bsql` command towards the 'mimerdb' database as follows:

```
# bsql mimerdb
```

In the package, some example programs are provided to show simple application development using the Mimer API. See `./qnx/examples` where a makefile and three example files are located. Preferable read the C-code to see what the files do before running the examples.

Before starting to create the example programs for the MimerAPI, prepare the environment by executing the following:

```
# source qnxsdp-env.sh
```

The header files in `./qnx/include` directory should be copied into the `${QNX_BASE}/target/qnx7/usr/include` as follows:

```
# cp -r ./qnx/include ${QNX_BASE}/target/qnx7/usr
```

Go into the `./qnx/examples` folder and do the following:

```
# make all
```

This should create three binary files `mimerapiex1`, `mimerapiex2` and `mimerapiex3`. To run the example program files they need to be executables. To do this, add the files to the build-script for the board. Or, copy the files from your storage media to a folder (like `/usr/tmp`) to be able to do a `chmod` command to make the files executable inside the QNX environment, as follows:

```
# cd /usr/tmp
# chmod +x mimerapi*
```

Remote access

The database can be accessed and further developed remotely from any host system, for example, using `bsql` or the `DbVisualizer` database tool as described under *Networking* on page 5.

On Linux host

The database can also be completely created and developed on a Linux host system.

In that case the database is created using the `dbinstall` command found in the Mimer SQL installation for Linux. With that command, a database can be up and running in seconds. The database development can then be continued using a suitable database access tool, like `DbVisualizer`, `bsql` and/or `mimload`. For further details, see the *Mimer SQL Getting Started on Linux* guide.

When settled, the databank files are copied to the QNX board, to the database home directory for the database server on the board. The Mimer SQL database server on the board is (re)started and the database will be up and running.

Build and execute

Add the Mimer SQL software to the target build environment

Depending on the target image to build, the corresponding directory in the package is used.

For example, in the case of building an image for the ARM based NXP board, the contents of the `./arm` directory is copied to the target QNX build environment, e.g.

```
/target/qnx7/aarche64le.
```

```
# cd ./qnx
# cp -r ./arm/bin /target/qnx7/aarche64le/bin
# cp -r ./arm/lib /target/qnx7/aarche64le/lib
```

The programs and libraries described above will now be integrated into the BSP (Board Support Package) build structure.

It is essential to have the `bin` and `lib` directories from the Mimer SQL installation package in the proper target folders, respectively.

Target build procedure setup

In the `./qnx/arm/scripts` folder there are a few build files, holding sample build procedures for different devices.

- For NXP S32V - The `evb.build` file must be updated/verified.
- For Raspberry Pi4 - The `rpi4.build` file must be updated/verified.
- For Renesas H3ulcb - The `rcar_h3ulcb.build` file must be updated/verified.

The respective build file holds the default definitions needed to include the Mimer SQL package in the build procedure.

The makefile setup in the target build environment should include the build file.

Configuring the image with Mimer SQL

In the build file there are possibilities to configure the setup for the Mimer SQL database server environment. For example, there are sections specifying Mimer SQL executables and shared libraries as mentioned earlier.

These predefined definitions are using the database name 'mimerdb'.

Note: There is a file called `mimer.build` where the Mimer SQL setup parts are collected, if needed for integration in another build procedure.

Build the target image

Make the target image in the ordinary way, including the prepared build scripts as described above.

Boot the created image and start the database

Boot the image in the ordinary way.

When the QNX system starts up, storage media is set up so that `/mimer/mimerdb` is available. When that is done, run the following command to start the database:

```
# mimcontrol -s mimerdb
```


Index

D

database
 establishing 5
date 8

F

file systems 2

I

in-memory database 8
installing
 Mimer SQL 3

L

license key 4

M

Mimer SQL 1
 installing 3
multidefs 6

P

package contents 3

S

sdbgen 5
sqlhosts 5

T

target boards 2
TCP/IP 5
time 8

