



# Mimer SQL

## Packaging Guide

### for Linux

Version 8.2  
May 2001

Mimer SQL, Packaging Guide for Linux, Version 8.2  
© Copyright Upright Database 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 property of the respective company. Companies, names and data used in examples herein are fictitious unless otherwise noted.

Produced and published by Upright Database Technology AB, Uppsala, Sweden.  
P.O. Box 1713,  
SE-751 47 Uppsala, Sweden.  
Tel +46(0)18-780 92 00.  
Fax +46(0)18-780 92 40.

Mimer Web Sites:  
<http://www.mimer.com>  
<http://www.upright.se>

# Contents

---

<b>Chapter 1 Introduction .....</b>	<b>1</b>
1.1 Document Overview.....	1
1.2 Definitions and Abbreviations .....	2
1.3 References.....	2
<b>Chapter 2 Creating the Installation Package .....</b>	<b>3</b>
2.1 Adding the Mimer SQL RPM File Information .....	3
2.2 The dbinstall Script.....	4
2.2.1 dbinstall Script Syntax .....	4
2.3 Example of How to Add Mimer SQL .....	6
2.3.1 Licensing Mimer SQL .....	7
2.3.2 Loading Application Data .....	8
<b>Chapter 3 About Previous Mimer SQL Versions .....</b>	<b>9</b>
3.1 Upgrading a Mimer SQL Installation .....	9
3.1.1 Upgrading a Mimer SQL Database .....	10



# Chapter 1

# Introduction

---

This document is intended for Value Added Resellers (VARs) that package and sell Mimer SQL together with their own application.

The purpose of this document is to describe how you can package the relational database system Mimer SQL together with an application.

By including Mimer SQL, end-users can install both the application and Mimer SQL in one single installation.

This document describes in detail how to do this in a Linux environment, with Mimer SQL version 8.2 or later distributed as an RPM file.

## 1.1 Document Overview

Chapter 2, *Creating the Installation Package* on page 3, contains instructions on how to include the necessary Mimer SQL information in your application's installation script.

Chapter 3, *About Previous Mimer SQL Versions* on page 9, explains how to upgrade from older Mimer SQL systems and how to upgrade a Mimer SQL database.

## 1.2 Definitions and Abbreviations

Term	Explanation
RPM	Red Hat Package Manager.  A set of specifications and tools to enable Linux and UNIX suppliers to provide software that is easy to automatically install, upgrade, remove and administer.

## 1.3 References

The Mimer SQL developer's site: <http://developer.mimer.com>

The latest editions of *Mimer SQL System Management Handbook* and *Mimer SQL Release Notes*, both available at:

<http://developer.mimer.com/documentation>

"Maximum RPM" by Edward C. Bailey, ISBN: 1-888172-78-9.

# Chapter 2

# Creating the Installation Package

---

There are two pieces of information you must add to your application's installation script when packaging Mimer SQL with your application:

- an RPM command that handles the Mimer SQL RPM file
- the Mimer SQL `dbinstall` script.

## 2.1 Adding the Mimer SQL RPM File Information

To add the Mimer SQL RPM file, for example, `MimerSQL-8.2.3B-3.i386.rpm`, you must edit your installation script.

To add the RPM file, use the RPM command, for example:

```
rpm -i MimerSQL-8.2.3B-3.i386.rpm
```

For details on RPM, see the publication "Maximum RPM". You can find the details in *References* on page 2.

## 2.2 The dbinstall Script

When run, the `dbinstall` script creates the Mimer SQL database for your application.

The `dbinstall` script performs the following:

- Adds the Mimer SQL database to the `/etc/sqlhosts` registry file.

The `sqlhosts` file contains a list of Mimer SQL database servers and is used for database lookup. If the `/etc/sqlhosts` file is missing, `dbinstall` will create it.

- Creates the initial database, that is, the system databanks and the `multidefs` configuration file.

The `SYSDB` databank and the `multidefs` file are created in the database home directory. The `TRANSDB`, `LOGDB` and `SQLDB` databanks are created in the directories you specify, or if no directories are specified, in the database home directory.

For more information on recommended databank locations, see the section on organizing databank files in the *Mimer SQL System Management Handbook*.

- Creates the license key file `/etc/mimerkey` if it is missing and adds the default key.
- Installs procedures for autostart in `/etc/init.d`, or corresponding.
- Starts the Mimer SQL database server, operating under the given operator user account. If no operator is specified, `root` is used. When the server is started the `mimer.log` server log file is created in the database home directory.

### 2.2.1 dbinstall Script Syntax

The following section describes the `dbinstall` script syntax.

#### dbinstall Usage

```
dbinstall [ -s
           database
           password
           home_directory
           [ transdb_directory
           logdb_directory
           sqldb_directory
           operator ] ]
```

## Options

Option	Explanation
-s	Silent mode. <b>Note:</b> If you specify -s, you must also specify the database, password and database home directory options.
database	database name.
password	database SYSADM password.
home_directory	database home directory.
transdb_directory	TRANSDB databank directory, default is current directory, i.e. the database home directory.
logdb_directory	LOGDB databank directory, default is current directory, i.e. the database home directory.
sqldb_directory	SQLDB databank directory, default is current directory, i.e. the database home directory.
operator	database owner, the default is root.

## 2.3 Example of How to Add Mimer SQL

The following is an example of how you can include the Mimer SQL RPM file and the `dbinstall` script in your application's installation script to silently install Mimer SQL.

**Note:** Any script that is written to install Mimer SQL must be executed by `root`.

```
.
.
# -----
# Installation part for Mimer SQL
#

# Define a log file and abort actions
LOG=$$mimsqlilog
trap "stty echo; rm $LOG; exit" 2 3 15

# Execute rpm to install software. OK if log file is empty
rpm -i MimerSQL-8.2.3B-3.i386.rpm > $LOG 2>&1
if [ $? -ne 0 ]
then
    echo "Mimer SQL installation failed: rpm -i: `cat $LOG`"
    exit 1
fi

# Installing the initial Mimer SQL database server
#     Database name is:   appdb
#     Password:           Given as argument to script ($1)
#     Home is:            /usr/local/opt/MimerSQL
#     Transdb directory: /disk1
#     Logdb directory:   /disk2
#     Sqldb directory:   same as home
#     Database operator: root
#
PASSWD=$1
/usr/bin/dbinstall -s appdb $PASSWD /usr/local/opt/MimerSQL
/disk1 /disk2 "" root > $LOG 2>&1
if [ $? -ne 0 ]
then
    echo "Mimer SQL installation failed: dbinstall: `cat $LOG`"
    exit 1
fi

#
# Mimer SQL installation done, ready to continue!
# -----
.
.
.
```

## 2.3.1 Licensing Mimer SQL

By default, `dbinstall` installs a free, built-in Test & Development license key.

However, if you want to use Mimer SQL in production, you must have a production license key. Contact Upright Database Technology, see [www.mimer.com](http://www.mimer.com) for details on how to acquire the configuration file which contains the production license key.

Once you have the file, you must edit your application's installation script to install the configuration file.

The following is an example of how to edit your application's installation script. `appdb_key.cfg` is the configuration file used.

```
.  
. .  
# Execute mimlicense to update the key file  
/usr/bin/mimlicense -f appdb_key.cfg > $LOG 2>&1  
  
# Verify the mimlicense session  
if [ $? -ne 0 ]  
then  
    echo "mimlicense failed, `cat $LOG`"  
    exit 1  
fi  
. .  
.
```

For more information on `mimlicense`, see the *Mimer SQL System Management Handbook*.

## 2.3.2 Loading Application Data

At this stage, the Mimer SQL database server is running and ready for multi-user access.

If you need to load application data you can do so by using the read command in the BSQL tool.

The following example shows how to load data using a shell script:

```
.
.
.
# Execute bsql to create schema and load application data
/usr/bin/bsql <<-% > $LOG
sysadm
$PASSWORD
READ 'appschema.sql';
READ 'appdata.sql';
EXIT;
%

# Verify the bsql session
if [ $? -ne 0 ]
then
    echo "bsql failed, `cat $LOG`"
    rm $LOG
    exit
fi
.
.
.
```

**Note:** In the example above, we are assuming that database is the system default database. If it is not, you must specify the database name as an argument in BSQL.

You can use the command: "mimhosts -d database\_name" to set a database as the system default.

# Chapter 3

# About Previous Mimer SQL Versions

---

This chapter discusses how to upgrade older Mimer SQL installations and how to upgrade a Mimer SQL database. For more information on upgrading, see the current version of the Mimer SQL Release Notes available at <http://developer.mimer.com>

## 3.1 Upgrading a Mimer SQL Installation

Basically, when upgrading, your application must use the new Mimer SQL shared library, called `libmimer.so`, and the Mimer SQL database server must be able to locate the new executable programs and the new shared library.

Although it is possible to run Mimer SQL versions in parallel, the most straight forward and recommended solution is to first uninstall the previous version and then install the new version.

You can use the following example in a shell script to check for and uninstall a Mimer SQL installation:

```
.
.
.
# Check if Mimer is installed. OK if log file is empty
INSTALLED=`rpm -qa | grep MimerSQL`

if [ ! $INSTALLED = "" ]
then
    rpm -e $INSTALLED > $LOG 2>&1
    if [ $? -ne 0 ]
    then
        echo "Mimer SQL uninstall failed: rpm -e: `cat $LOG`"
        exit 1
    fi
fi
.
.
.
```

**Note:** This script above checks for and uninstalls Mimer SQL software, it will not affect the Mimer SQL database.

### 3.1.1 Upgrading a Mimer SQL Database

When upgrading between major releases, for example from 7.3 or 8.1 to 8.2, you have to upgrade existing databases using a database upgrade utility.

To upgrade a Mimer SQL database on Linux, you use the command:

```
sdbgen -u database_name
```

From a shell script, you can do as follows:

```
.
.
.
# Upgrade a Mimer SQL database
/usr/bin/sdbgen -u appdb > $LOG 2>&1
if [ $? -ne 0 ]
then
    echo "Mimer SQL database upgrade failed: sdbgen: `cat $LOG`"
    exit 1
fi
.
.
.
```

**Caution:** Always remember to carry out a proper backup before upgrading a database.

Mimer SQL version 8.2 supports a number of new data types and other features, that can easily be brought into use through the ALTER TABLE statement.

See the *Mimer SQL Reference Manual* for further information about ALTER TABLE.