Mimer SQL Engine

Getting Started

Version 9.2

Mimer SQL Engine, Getting Started, Version 9.2, June 2005

Produced and published by Mimer Information 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 SQL Web Sites:
http://developer.mimer.com
http://www.mimer.com

Contents

Getting Started

Welcome to Mimer SQL Engine. This document describes how to set-up Mimer SQL Engine on Windows 98, ME, NT, 2000 or XP after the installation.

To get the most out of this document, you should be familiar with your Windows environment and know how to use the various Windows system tools.

Licensing Mimer SQL Engine

When you install Mimer SQL Engine, a default development edition license is installed. This license covers basic usage for development purposes and enables 10 concurrent users. You do not need a license for client access.

If you want to use Mimer SQL Engine for any purpose other than development, you must purchase a commercial license. Contact your Mimer SQL distributor to purchase the license you require. Your new license will be sent to you via e-mail. You apply the new license by double-clicking on the e-mail attachment.

For more information, start the Mimer Administrator, located in the Mimer SQL Engine program group, click the **License Key** tab and select **Help**.

Rights and Privileges

On Windows NT, 2000 and XP, you must belong to the administrators group to set-up Mimer SQL Engine clients and servers.

Documentation

The Mimer SQL Engine Documentation Set, JDBC Driver Guide, Release Notes, and Windows specific documentation are available in the Mimer SQL Engine program group, under Online Documentation.

Note: The documentation set (i.e. SQL Reference Manual, Programmer's Manual, System Management Handbook, and User's Manual) will only be available if you chose to install it when you installed Mimer SQL Engine.

If you would like printed copies of the documentation set, contact your Mimer SQL representative.

Useful Links

The Mimer SQL Developer Site contains lots of useful information, like FAQs, How-to's and articles: http://developer.mimer.com

Setting-up a Database Client

If you want to access a Mimer SQL database server (also referred to as a remote database) already installed on your network, all you need to do is set-up a Mimer SQL Engine database client.

You set-up a client using the Mimer Administrator.

1. Start the Mimer Administrator, located in the Mimer SQL Engine program group.
2. Click the **Remote** tab and then click the **Add** button.
3. In the Remote Database Definition dialog box, enter the database name, the node (computer) where the database server is located and which network protocol to use.

   The database name must be the same name used on the remote computer. Ask your system or database administrator if you are unsure of the name.

   The node name is the name of the computer as registered in the network. It is usually entered in lower case letters.

   The network protocol is usually **tcp**. If the remote computer is a Windows NT/2000/XP server, you can also use **NamedPipes**.

   Click **OK**, you are now ready to access the Mimer SQL Engine database from your development tool. An ODBC data source with the same name as the database is created automatically.

Creating Mimer SQL Engine Databases

Creating Mimer SQL Engine databases is easy. All you need to do is:

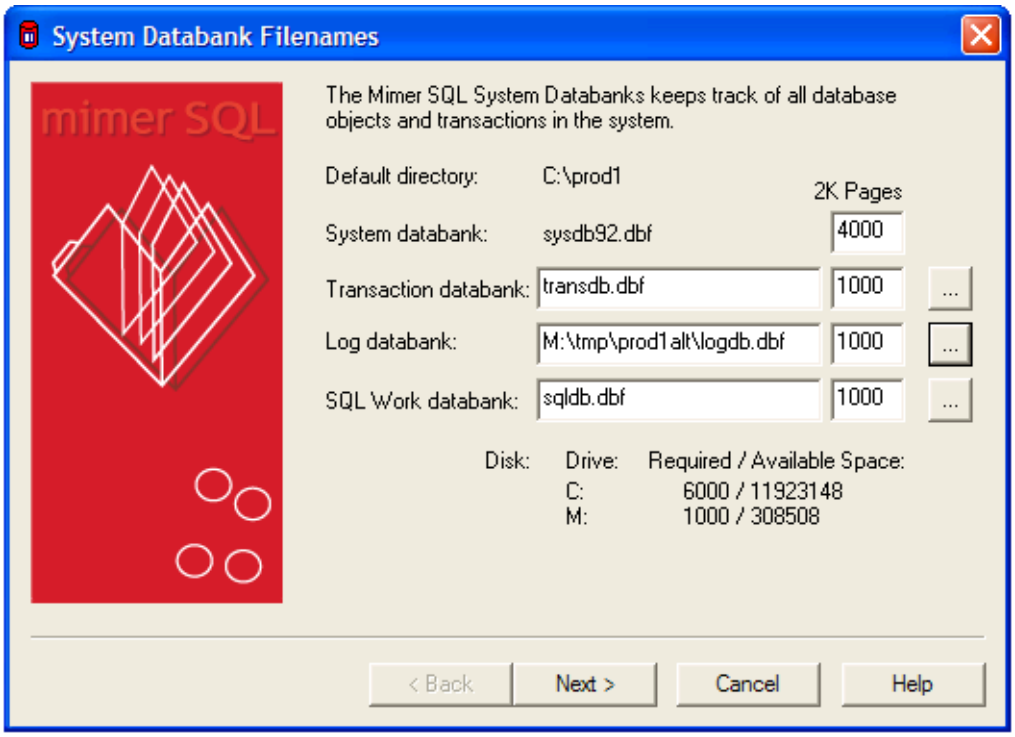
- add a database definition for the database
- create the system databanks
- set the `SYSADM` password
- create a database user
- start the Mimer SQL Engine database server.

You do this using the Mimer Administrator and a wizard to guide you.

Creating a Database

1. Start the Mimer Administrator, located in the Mimer SQL Engine program group.
2. In the Mimer Administrator, on the **Local** tab, click **Add** to create a database. The Local Database Definition dialog box opens.
3. Enter the name of the database and the directory in which you want to create it. Click **OK**.

4 The Mimer Administrator asks if you want to create system databanks. In the Create Systems dialog box, click **Yes**. The following wizard starts:



The example above demonstrates the following important points:

- you must specify the file locations of databanks not in the default directory as absolute paths

- to safeguard against data loss in the event of a hard disk failure, place the log databank, `LOGDB`, on a separate disk from the other databanks.

- If possible, also place `TRANSDB` on a separate disk. However, `LOGDB` and `TRANSDB` should not be placed on the same disk. If you only have one alternative disk, use it for `LOGDB`. Mimer SQL Engine examines the available hard disks and suggests how the files should be distributed. (In the above example, C: and M: are on separate hard disks.)

Note: On Windows NT, 2000 and XP, you can use the Disk Administrator in the Administrative Tools menu to check the layout of logical disk partitions (e.g. C:, D:) on the available hard disks.

The file sizes are expressed in 2K blocks. The initial sizes are not critical as the files will be automatically extended as required.

Enter the database information and click **Next**.

5 The next step is to specify the password for the system administrator `SYSADM`. Enter the system administrator's password, and confirm it. Click **Next**.

Note: Keep the `SYSADM` password secure, as `SYSADM` can alter passwords for Mimer SQL Engine users that he/she creates. If you lose the `SYSADM` password, it cannot be retrieved from your Mimer SQL Engine system.

6 When the system databank creation is complete, the wizard asks you what you want to do next. Select both options available and click **Next**.

7 Continue with the wizard to create a development user and start the database server.

After completing the wizard, you can access your database directly from your favorite ODBC tool. You can also access it using Mimer WSQL – available for download for free from our Web site: http://developer.mimer.se/howto/howto_14.htm

If you want to access this database from another computer in the network, you must install and set-up a database client on the other computer. See *Setting-up a Database Client* on page *2*.

Starting and Stopping Database Servers

How you start and stop Mimer SQL Engine database servers depends on the Windows operating system you are using.

However, on all Windows platforms supported by Mimer SQL Engine, you can control local database servers using the Mimer Administrator.

Go to your Mimer SQL Engine program group and select **Mimer Administrator**, Mimer Administrator starts.

Click the **Local** tab and right-click on the database name. A pop-up menu enables you to start and stop the database server.

Starting and Stopping on Windows NT, 2000 and XP

On Windows NT, 2000 and XP, you can start and stop database servers, and enable or disable logins using the Mimer Controller, located in the Mimer SQL Engine program group. You can control both remote and local databases using the Mimer Controller.

The operating system equivalent of the Mimer Controller is the Services dialog box in the Control Panel. You can use Windows Services to start and stop Mimer SQL Engine databases.

Autostart – NT, 2000 and XP

By default, Mimer SQL Engine database servers are created with the startup setting set to **Autostart**. Mimer SQL Engine database servers will start automatically whenever the machine reboots.

You can view and change the Autostart setting using the Mimer Administrator or Windows Services.

Starting and Stopping on Windows 98 and ME

On Windows 98/ME, database servers are started automatically whenever they are needed by an application. The database servers remain active until they are explicitly shut down.

Logging Database Events

Windows NT, 2000 and XP

On Windows NT, 2000 and XP, Mimer SQL Engine writes events to the Windows event log. Use the Event Viewer, located under Administrative Tools, to examine the event log.

The source for Mimer SQL Engine events is the name of the database service they relate to, and they are logged in the Application log.

Windows 98 and ME

On Windows 98/ME, information about any database server errors that occur during database startup can be found in the `MIMER.LOG` file located in the database's home directory.

Information about the latest startup is also displayed in the Mimer SQL Engine database server's main window. If you receive any error messages, check this window for more information on the error.

Running Mimer BSQL and Other Utilities

In order to run most of the Mimer SQL Engine utilities from a command prompt window, you must specify which database to access. You can do this in different ways:

- Enter the database name on the command line, e.g. `BSQL database_name`
- Use the environment variable `MIMER_DATABASE`, e.g.
  `SET MIMER_DATABASE=database_name`
- Use an ODBC default data source. Specify **Default** as the data source name in the Mimer Administrator.

The order of the three methods is significant as the first methods override the later ones. For example, specifying the database on the command line overrides the setting of the `MIMER_DATABASE` environment variable.

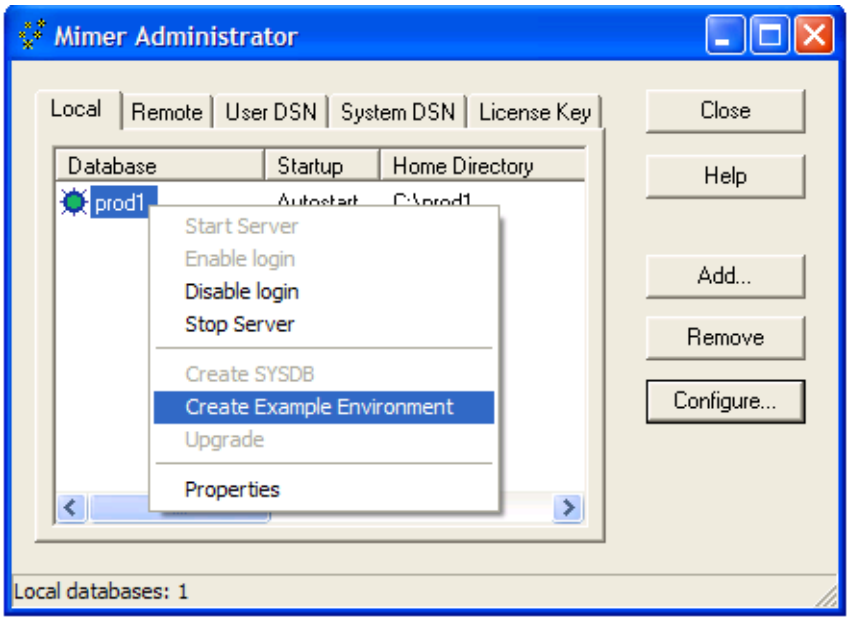
The Developer and Example Environments

If, when installing Mimer SQL Engine, you chose to install the development and sample files, you can choose to set-up a development environment and an example environment.

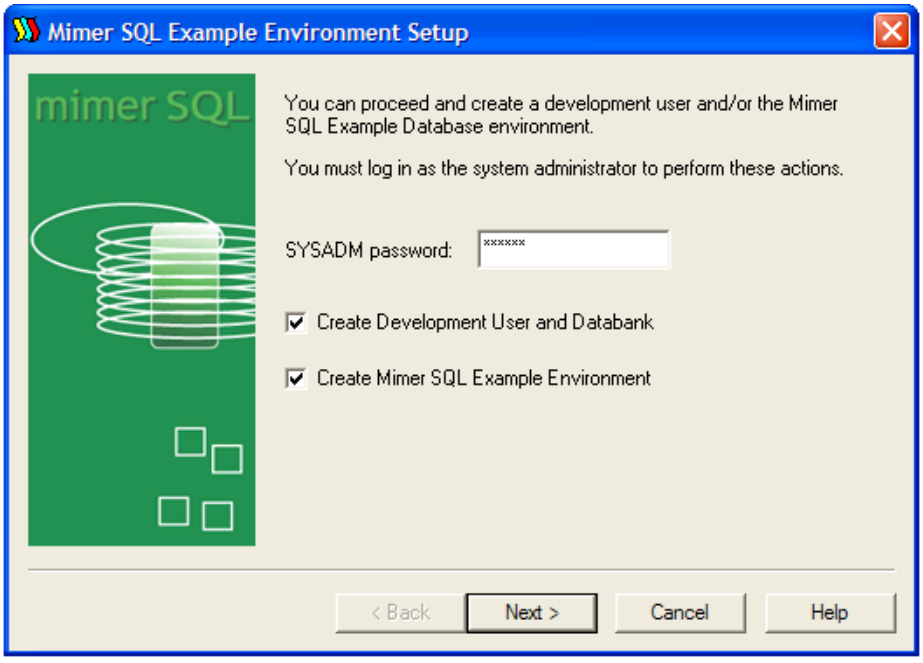
To install the environments:

1 Start Mimer Administrator, located in the Mimer SQL Engine program group.

2 Select the **Local** tab, and right-click on the database server, for example:



3 Select **Create Example Environment**, the following dialog box opens:



4 Select the environment(s) you want to create and click **Next**. A wizard will guide you through the necessary steps.

Accessing the Example Environment

To access the example environment, you can use:

- Mimer BSQL, available in the Mimer SQL Engine program group.
- Any ODBC-based SQL tool.
- WSQL, an unsupported SQL tool, available for download from http://developer.mimer.com/downloads

Building the Development Environment

Note: You must have the Microsoft Visual Studio or equivalent development kit installed.

To build the environment:

1. Open a command prompt window.
2. Change the current directory to the development kit home directory, under the installation directory, e.g. `\Program Files\Mimer SQL 9.2\DEV`.
3. Enter the following commands:

```
NMAKE /F makefile.mak
DSQL
OSQL
NMAKE /F makefile.mak clean
```

You must have a path which includes the `NMAKE` program.

The sample programs created accept SQL statements and display selected results.

Note: To build and run the OSQL (ODBC sample program) you must have defined a data source.

You can use the `makefile` as a starting point for writing your own `makefiles`.

An alternative to the above steps is to open the `makefile` in Microsoft Visual Studio and then compile the programs.

Mimer PSM Debugger

Mimer SQL Engine also contains a Java-based graphic debugger for PSM routines. The Mimer PSM Debugger supports watching variables, step-wise execution and setting breakpoints.

You can debug procedures and functions.

Mimer PSM Debugger requires a Java 2 (version 1.2 or later) compatible Java runtime environment.

You can download the files from: http://java.sun.com/j2se/downloads.html

Starting the Mimer PSM Debugger

You start the PSM debugger with this command:

```
java -jar psmdebug.jar
```

The syntax for the database URL in the login dialog is:

```
hostname[:port]/database
```

If the database resides on your local machine, then specify `localhost` as the host name.

Examples:

```
localhost/testdb

my_node.mimer.se/supplier

my_node.mimer.se:1365/supplier_temp
```

Packaging Mimer SQL Engine with a Windows Application

If you want to package your application with Mimer SQL Engine, you should read the Mimer SQL Packaging Guide for Windows: http://developer.mimer.com/documentation/Mimer_SQL_Packaging_Guide_Windows/Mimer_SQL_Packaging_Guide_Windows.htm

ODBC Trace

Mimer ODBC Trace is a tool for getting the details out of an ODBC session. It can for example be used for debugging, monitoring and analyzing purposes. For each ODBC statement the tool can trace:

- the elapsed time in server
- the SQL statement used (if appropriate)
- the ODBC function used
- the timestamp for the ODBC call
- the ODBC handle used
- the calling application process ID and name
- the return code for the ODBC call

This software consists of two parts, the Mimer ODBC Trace Administration tool and the Mimer ODBC Trace DLL that substitutes the standard Microsoft ODBC Trace DLL.

The administration tool has two main functions:

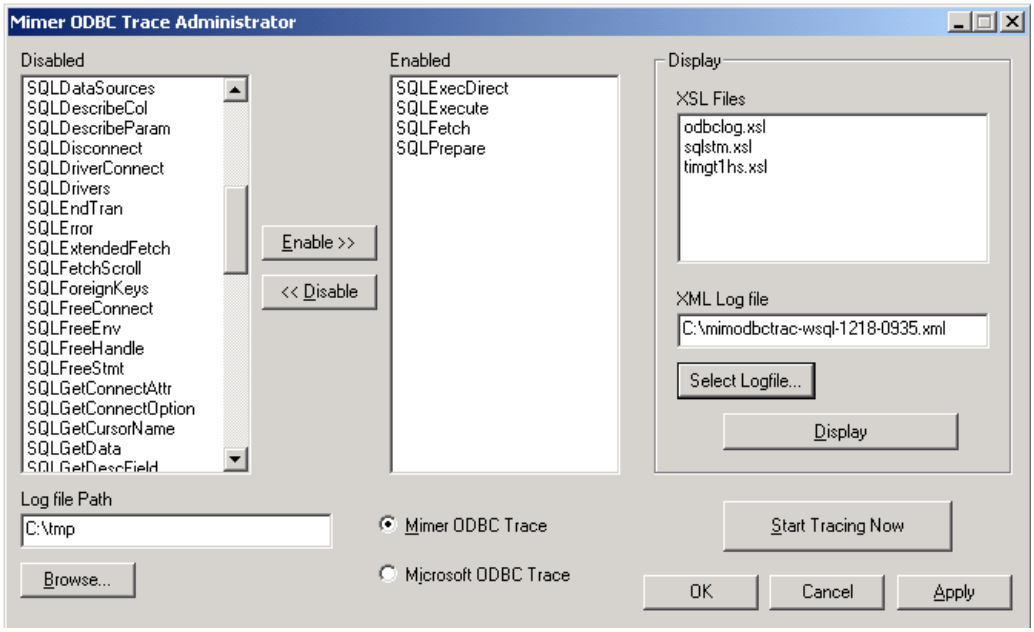
- Configuration - to setup prerequisites for the trace logging and to start/stop logging
- Presentation - to select and display the records from a logging file.

To use the Mimer ODBC Trace DLL, it must first be activated using the administration tool.

Before using this software however, you should read *Important Notes* on page *13*.

Using the Administration Tool

When starting the administration tool, the following dialog is displayed.



The Configuration Part

The left side of the window is the **Configuration** part, which is used to change the settings for the logging.

The two list boxes are used to manage which functions calls are to be logged, and which are not. Calls to functions present in the left list box will not be logged, and calls to functions present in the right list box will be logged.

The **Enable>>** button moves any selected function(s) in the left list box over to the right, thus activating logging of calls to that/those function(s).

The **<<Disable** button moves any selected function(s) in the right list box over to the left, thus deactivating logging of calls to that/those function(s).

The **Log file Path** edit field is where you specify the location of the generated log file(s). The path must exist for logging to be conducted successfully.

The radio button **Mimer ODBC Trace** specifies that the Mimer ODBC Trace DLL should be used instead of the default Microsoft ODBC Trace DLL. There is also a radio button **Microsoft ODBC Trace**, which can be used to switch back. See also *Important Notes* on page *13*.

The **Apply** button saves any changes to the fields described above.

The **Cancel** button dismisses the dialog without saving any changes you have made and that are not applied.

The **OK** button save any changes you might have made, and dismisses the dialog.

Start Logging

After pressing the **Start Tracing Now** button (renames to **Stop Tracing Now**), logging will take effect for all applications attaching to the DLL. As the settings only will be read when the DLL is loaded, already started applications will not be affected.

Each started application will have a log file of its own, placed in the location specified by Log file Path and named `mimodbctrac-`*`applname-mmdd-hhmm`*`.xml`.

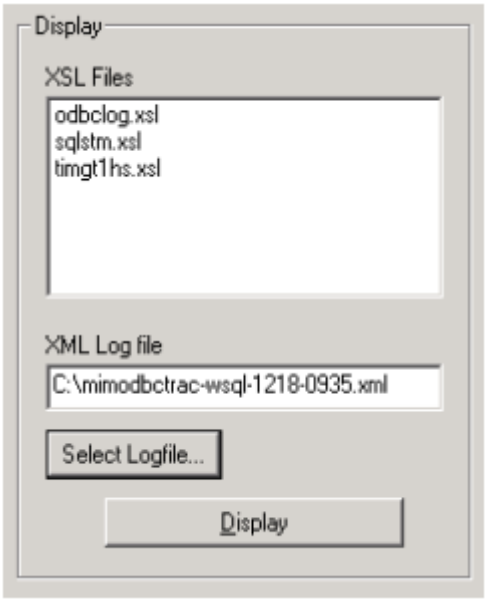
Stop Logging

Pressing the **Stop Tracing Now** button (renames to **Start Tracing Now**) means that no logging will take place for applications started afterwards.

Note: Already running applications will continue logging until they quit. When an application ends, its log file is closed and becomes available for presentation.

The Presentation Part

The right side of the main window is the **Presentation** part (as showed below). It is used to present results from a selected log file, that has been produced by the Mimer ODBC Trace DLL.



First of all you have to find a Mimer ODBC Trace log file by using the button **Select Logfile**. Then you get a new window `Select Log File`, where you should choose a file, which is named according to the output from the DLL, i.e. `mimodbctrac-`*`applname-mmdd-hhmm`*`.xml`. When selected the name will be displayed in the field **XML Logfile**.

Next a presentation form should be selected from the **XSL files** menu. When a XSL file is selected the **Display** button should be pressed and the presentation will be shown.

Presentation Formats

Currently there are 3 different predefined presentations:

odbclog.xsl

Shows all information from the log file. Functions, which used more time than 0.01 seconds are displayed with the time in red color.

sqlstm.xsl

Shows all SQL statements found in the log file.

timgt1hs.xsl

Shows all functions, which used more time than 0.01 seconds.

Example odbclog.xsl:

Timestamp	Function	UsedTime (sec)	Application (Processid:Threadid)	Handle [id]	Return code	SQL-statement
2003-12-11 15:28:29.60	SQLAllocStmt	0.0001	wsql (fff60ad9 :fff49a99)	HDBC[0x028f0280]	SQL_SUCCESS	
2003-12-11 15:28:29.93	SQLPrepare	0.2296	wsql (fff60ad9 :fff49a99)	HSTMT[0x028f0acc]	SQL_SUCCESS	select P from JSP

Example sqlstm.xsl:

SQL-statements
`SELECT DISTINCT P FROM SPJ, S, J  WHERE SPJ.S = S.S AND SPJ.J = J.J AND S.CITY = J.CITY`

Example timgt1hs.xsl:

Timestamp	Function	UsedTime (sec)	SQL-statement
2003-12-11 15:28:29.93	SQLPrepare	0.1675	SELECT DISTINCT P FROM SPJ, S, J WHERE SPJ.S = S.S AND SPJ.J = J.J AND S.CITY = J.CITY

Customized Presentations

Additional XSL files can be added to the default XSL file location.

The distributed XSL files may, for example, be modified, and replaced or stored as a new XSL file in the same location. Maybe other time limits are desired, or other colors, or others.

If for example, other time limits are wanted, the distributed XSL files may be modified and replaced or stored as a new XSL file in the same location.

Important Notes

- When changing any settings using the administration tool, you should restart any application for which you want the changes to be used. The reason for this is that the settings will only be read when the DLL is loaded by the application, and thus the DLL has to be reloaded for any changes to take effect.
- The administration tool changes the registry entries in `HKEY_CURRENT_USER\SOFTWARE\ODBC\ODBC.INI\ODBC`

  If there is any problem look there in order to see if your changes are applied.
- The "original" ODBC Administrator may also change those registry entries. Currently there is a problem concerning the cooperating use between these two administration tools. Watch out which DLL is selected, as both administrators may start or stop tracing with anyone of the DLL:s.