

Jürgen Aurisch  
Markus Stausberg  
Wolfgang Auer

SAP AG

# **MaxDB Synchronization Manager**

## **Message Server**

April, 20th, 2005

Version 7.6

© SAP AG 2005

Important Note .....	3
1 Installation .....	3
1.1 Prerequisites .....	3
1.2 Message Server Database Instance .....	3
1.3 Creating the Message Server Database User .....	3
1.4 Classpath and Configuration Files .....	4
1.4.1 Example Command Line for starting the Server .....	4
1.4.2 Example Command Line for stopping the Server .....	4
1.4.3 User Names and Passwords on the Command Line .....	4
1.5 Messaging Client Configuration and Network Settings .....	5
1.6 SyncMan Access Key .....	5

## Important Note

The information in this document describes configuration and deployment of the Synchronization Manager suite programs when run from the command line using java. This approach should only be taken in special, complex scenarios by advanced users.

The recommended way to configure and run the Synchronization Manager programs is to use the wrapper scripts which are provided in '`<independent program path>/bin`' .

For documentation, refer to the official MaxDB documentation, available at <http://www.mysql.com/products/maxdb/> .

For a simple walkthrough example of Synchronization Manager, see '`<independent program path>/app/syncman/example`' .

## 1 Installation

### 1.1 Prerequisites

The following external Java software is needed to run the Message Server :

- Java runtime  $\geq$  1.4
- *j2ee.jar* (1.3) from the Java 2 Enterprise SDK, available at <http://java.sun.com/j2ee/> .
- *sapdbc.jar* , the SAP DB/MaxDB JDBC implementation, available at <http://www.mysql.com/products/maxdb/> (or a version which matches the deployed MaxDB database software version)

The latter jar files could be placed in the *SyncMan/lib* directory.

### 1.2 Message Server Database Instance

Install the MaxDB database software (recommended  $\geq$  version 7.6.00.07).

This software can be obtained from <http://www.mysql.com/products/maxdb/> .

Refer to the installation instructions.

Create a database instance. This instance could run on a different machine.

### 1.3 Creating the Message Server Database User

If not installed with the database instance create a database user which will be used by the Message Server.

The user needs to be a DBA user. You can create the user with the DBMGUI or by executing a SQL Command with e.g. SQL Studio.

Example:

```
create user <user> password <password> dba not exclusive
```

## 1.4 Classpath and Configuration Files

Edit *MSGSERVER.xml* according to your local settings. In the *MSGSERVER.xml* set the database url to the created instance and to the chosen user name and password. (There is no user name and password needed for the transient database.) Alternatively you can use command line options for the server node, the database, the database user and the database password.

A password to administrate the Message Server can be set in the section AdminPassword or by the command line option `-msgserver_password`. Alternatively the Database credentials and the Message Server Admin Password can be obtained with the option `-A <access_key>` from the SyncMan Access File.

### 1.4.1 Example Command Line for starting the Server

Copy the `j2ee.jar` and the `sapdbc.jar` in the `SyncMan/lib` directory.

To start the Synchronization Process in the SyncMan directory, use:

```
java -classpath .\lib\sapdbc.jar;.\lib\msgserver.jar;.\lib\j2ee.jar
      com.sap.sdb.msgServer.Server -f .\MSGSERVER.xml
```

Under Linux the elements of the class path are separated by a colon.

Alternatively a batch file with the predefined class path may be used: `msgserver`

### 1.4.2 Example Command Line for stopping the Server

To stop the server gracefully, use the following command line:

```
java -classpath .\lib\msgserver.jar com.sap.sdb.msgServer.AdminClient
      [-host <server>] -port 7220 -password <your_admin_password> shutdown
```

`-host` denotes the machine where the server runs (default: localhost)

`-port` and `-password` parameters should match the settings in *MSGSERVER.xml*

Alternatively the Message Server credentials can be obtained with the option `-A <access_key>` from the SyncMan Access File.

Alternatively a batch file with the predefined class path may be used:

```
msgserverstop [-host <server>] -port 7220 -password <your_admin_password>
```

### 1.4.3 User Names and Passwords on the Command Line

The MaxDB database and all database tools adhere to a special policy concerning case sensitivity of user names and passwords which apply as well to all user names and passwords supplied to the messaging software. This is as follows:

If such a value is provided without being surrounded with double quote signs, it will implicitly be converted to upper case. To supply a value containing lowercase letters the value has to be surrounded with double quote signs. Note that in most shells it is required to escape the quote signs in order to pass them to the invoked program. This is usually done by using preceding backslash characters. Example :

```
java -classpath [...] -password \"secret\"
```

This also applies to respective values supplied in XML configuration files, where escaping is done using a CDATA section.

## 1.5 Messaging Client Configuration and Network Settings

A messaging (i.e. JMS) client is configured by the *jni.properties* file in the unnamed, top level Java package. The TCP/IP ports configured there should match the settings in *MSGSERVER.xml*. For convenience, a default *jni.properties* is already included in *msgserver.jar*.

## 1.6 SyncMan Access Key

In a SyncMan Access Key database and Message Server credentials can be stored. The command line help provides all necessary information.

To display the command line help you can call:

```
java -classpath .\lib\syncmanacc.jar  
com.sap.sdb.syncMan.acc.SyncManACC -?
```