# Fusion Database User Manual

*Version 1.3.0*

***Table of Contents***

# 1 Document Introduction

## 1.1 Document Purpose

This document explains installation procedures for the Fusion Database module.

## 1.2 Document Audience

This document is targeted at database administrators.

## 1.3 Document History

| Version | Editor | Date | Changes |
|---|---|---|---|
| 1.1.1 | Fredrik Gratte | 25-Feb-09 | Initial public version. |
| 1.1.2 | Fredrik Gratte | 23-Mar-09 | Added max_allowed_packet hint. |
| 1.1.3 | Morten Simonsen | 29-Sep-09 | Version updated. |
| 1.2.0 | Morten Simonsen | 22-Oct-10 | See version history |
| 1.2.1 (2011R1) | Morten Simonsen | 05-Jan-11 | See version history |
| 1.2.2 (2012R1) | Morten Simonsen | 21-Nov-11 | See version history |
| 1.3.0 (2013R1) | Morten Simonsen | 22-Jan-13 | See version history |

## 1.4 Version history

### 1.4.1 v1.3.0 (2013R1)

Many new tables:

- trigger, trigger_event, trigger_release (Trigger/Alaram)
- heartbeat (Heartbeat detection)
- script_execution (Fusion Script execution support)
- unit_param_session (Inspection/Extraction support)
- test_case, test_case_param, test_case_files, test_history (TR-069 client test system)

Some change in the existing tables:

- added admin flag to user_ table (improved permisssion system)
- added owner to filestore table (make script execute with correct permissions)
- added filestore_id to syslog_event (allow sylsog event script execution)

### 1.4.2 v1.2.2 (2012R1)

Name change: The name of the project has changed from xAPS to Fusion. The change has not been completed, thus "xAPS" or "xaps" is still found in many places.

Database support: No longer officially supports Oracle.

Only minor table changes since 1.2.1:

- Group parameter table changed primary key, and added two columns to handle operator and data type (the changes are necessary to support advanced group searches)
- Syslog table changed slightly with a longer field for facility_version (48)

- Syslog table changed some indexes for better performance

### 1.4.3 v1.2.1 (2011R1)

Only minor changes since 1.2.0:

- Job table expanded to support repeat_interval and repeat_counter
- Group table expanded to support time rolling group
- Report tables revision, removed some unused tables, changed some names

### 1.4.4 v1.2.0

Since 1.1.3 we have added 9 tables and changed many others. The changes are:

- added reports to the system (6 tables)
- added user/permissions (2 tables)
- added certificates (1 table)
- changed syslog_event table structure (added syslog delete limit)
- changed group_, added group_count
- changed unit_param and syslog, changed the indexes
- changed job, loosened some restrictions in job setup

## *1.5 References*

| Document | |
|---|---|
| [1] | Fusion Shell User Manual |
| [2] | Fusion Installation |

# 2 Introduction

The Fusion Database module contains the database schema needed to set up the Fusion Database. The schema is implemented as SQL scripts which must be executed by the database server. The database server/engine must be MySQL 5.5 (latest update).

Please refer to documentation provided by MySQL 5.5 for installation and configuration instructions of the database server. However, an extremely simple setup is provided in the [2] (chapter 4 or 5).

The next chapters provide installation from scratch (chapter 3) and migration from 2012R1 till 2013R1 (chapter 4).

# 3 Installation from scratch

These instructions will create a new MySQL user with its own database, and load the Fusion Database schema into it. The following key parameters are used:

User: user
Password: password
Database: xaps
Host: localhost

Since we assume a situation where everything is done on localhost, we can skip using the -h option for the mysql and mysqldump commands. If you want to setup and/or manage the Fusion Database remotely, just add –h<host-name> to the "mysql" commands in the examples below (not the SQL commands).

## 3.1 Create a new database

**If you have Fusion installed already, and want to upgrade, read chapter 3.3 first.**

To create a new database on the MySQL Server, perform the following steps:

1. Login to mysql as root (using the MySQL root password (not to be confused with the “password” password in this example) when prompted):
   ```
   mysql -uroot -p
   ```

2. Create a user for the Fusion Server modules to use when connecting. Skip the first user if you will only allow traffic from localhost – most secure, but then all Fusion modules must be installed on localhost, which may not be recommended for other security reasons and performance reasons.
   ```
   CREATE USER 'user' IDENTIFIED BY 'pass';
   CREATE USER 'user'@'localhost' IDENTIFIED BY 'password’;
   ```

3. Create a database for the Fusion server modules to use:
   ```
   CREATE DATABASE xaps;
   ```

4. Switch to the newly created database:
   ```
   USE xaps;
   ```

5. Grant permissions for the database to the user (see comment on 2.)
   ```
   GRANT ALL ON xaps.* TO 'user';
   GRANT ALL ON xaps.* TO 'user'@'localhost';
   ```

6. Commit the changes:
   ```
   COMMIT;
   ```

## 3.2 Load database schema

To load the Fusion Database schema, first unzip and change directory

```
>unzip xapsdatabase.zip
>cd mysql
>mysql -uuser -ppassword xaps < install2013R1.sql
```

You can log into the mysql database to see that the 40 tables were created:

```
>mysql -uuser -ppassword xaps
mysql> show tables from xaps;
+-----------------------+
| Tables_in_xaps        |
+-----------------------+
| certificate           |
| filestore             |
| group_                |
| group_param           |
| heartbeat             |
| job                   |
| job_param             |
| message               |
| monitor_event         |
| permission_           |
| profile               |
| profile_param         |
| report_gateway_tr     |
| report_group          |
| report_hw             |
| report_hw_tr          |
| report_job            |
| report_prov           |
| report_syslog         |
| report_unit           |
| report_voip           |
| report_voip_tr        |
| script_execution      |
| syslog                |
| syslog_event          |
| test_case             |
| test_case_files       |
| test_case_param       |
| test_history          |
| trigger_              |
| trigger_event         |
| trigger_release       |
| unit                  |
| unit_job              |
| unit_param            |
| unit_param_session    |
| unit_type             |
| unit_type_param       |
| unit_type_param_value |
| user_                 |
+-----------------------+
40 rows in set (0.00 sec)

mysql>
```

# 4 Migration from 2012R1 to 2013R1

Every new release of Fusion brings about a new version of the database. This is due to the continuously development of Fusion. The goal of this chapter is to get you safely through a migration of the data from your old version to the new version. Needless to say, this requires a careful approach supported by a healthy number of backups. It is of out-most importance to pay attention to details to avoid possible loss of data. Another situation where this is needed, is the event of moving database to another server, or copying from production server to test server.

## *4.1 Database backup*

### 4.1.1 Raw backup

On a MySQL server with default configuration you will probably find one big file in your file system: /var/lib/mysql/ibdata1 (The path may vary depending on OS and distributions). This path usually contains all data necessary to run mysql. You could just backup the whole path if you want.

### 4.1.2 Mysqldump

Another approach is to use the mysqldump tool:

```
>mysqldump -uuser -ppassword xaps > database_backup.sql
```

To backup the entire MySQL database – which will also backup any users created (they are stored in the "mysql" database, not in "xaps" database), run this command (you need the MySQL root password):

```
>mysqldump -uroot -p -all-databases > all_databases_backup.sql
```

## *4.2 The migration*

### 4.2.1 Export

Install the latest version of Fusion Shell (read at least some of [1]). Perform the following command:

```
>java -jar xapsshell.jar -export ALL -user user -password password -url
jdbc:mysql://localhost:3306/xaps
```

The entire content of your Fusion database should now be exported to files found in the directory "ALL". This command must end without any exception or error, so check all the output of the command (search for case-insensitive "error").

### 4.2.2 Make new database

Create the database tables from scratch. This will drop/delete all tables in your database and create everything anew.

```
>mysql -uuser -ppassword xaps < install2013R1.sql
```

It's also possible to run another sql to upgrade just those tables touched by the change from 2012R1 to 2013R1. However, then you need to know which tables are dropped and you might have to export and delete certain parts of the database in order for the drop/create table to complete. Then you must do an import of those exported parts again. All of this requires expert knowledge, and is **not recommended** for most users.

```
>mysql -uuser -ppassword xaps < upgrade2013R1.sql
```

### 4.2.3 Import

Then import the data we previously exported:

```
>java -jar xapsshell.jar -import ALL -user user -password password -url
jdbc:mysql://localhost:3306/xaps
```

Migration is completed if no errors nor exceptions occur. Keep in mind that old versions of Fusion Shell will fail, always use the most recent version.

# 5 Tuning to increase performance

To actually tune a database to perform optimally is not a task you will master in one day or two. There is a whole range of tools (mytop, innotop, iostat, MySQL Workbench, etc), that you may need to understand. Furthermore, you must understand the relationship between disk/buffer/query cache/select/insert etc.

Non the less, we can try to guide you on the right track. The number 1 issue when it comes to bad performance on MySQL running Fusion is the syslog table. It is **very** easy to let this table grow to an enormous table, suddenly bogging down your system. To address this issue we have provided Fusion Core Server, which will try to keep the syslog table in check, by deleting after certain rules.

Another way to go about it, is to reduce the load of incoming, non-essential messages. You may specify syslog events to identify such non-essential messages and discard them or even count duplicates (and only store one of them). To set up such a syslog event, use Fusion Web or Fusion Shell.

Next you need to consider the settings in my.cnf. Those that come as default are usually inadequate for a medium/large sized database. There are many settings which you should look into, but perhaps the most important is this:

```
innodb_buffer_pool_size = 128M
```

The buffer pool size is recommended to be 80% of the memory on a dedicated MySQL Server. However, setting it too high may cause your database not to start the innodb-engine. And it will do it silently!!!

Luckily MySQL usually provides various my.cnf files, so you could try to exchange the default my.cnf with another suited for medium or large databases (search for my-large.cnf on your system).

Another consideration is hardware. If you are running a system with maximum 1 GB memory reserved for MySQL, but your table size amounts to 10-20GB, it's like asking for trouble. Ideally the memory should at least be big enough to contain all indexes in the databases, and the indexes for the syslog table could easily amount to some GB (for a large table of course). Fast disks are also important. We have not run with SSD yet, but it sounds like a good idea.

Lastly, the syslog database/table could be split out from the Fusion database. Then the responsiveness of Fusion could be kept intact, while all the syslog load is contained on another server. All the Fusion modules support this in their configuration, just specify another host for the syslog connection.

# 6 Backup routines

To entirely delete the Fusion Database, perform the following steps:

- Backup to file. This backup is your lifeline if everything else fails.

```
>mysqldump -uuser -ppassword xaps > database_backup.sql
```

- Optional: To backup the entire MySQL database – which will also backup any MySQL users created (they are stored in the "mysql" database, not in "xaps" database), run this command (you need the MySQL root password):

```
>mysqldump -uroot -p -all-databases > all_databases_backup.sql
```

- Read chapter 5.2 or [1] and make an export of your entire database. This is the main migration path, since Fusion Shell is able to both read old and new databases, and will make a swift and clean upgrade.

## *6.1 Restore from backup*

A simple restore of the database can later be done (if needed) by running this mysql command:

```
>mysql -uuser -ppassword xaps < database_backup.sql
```

The correct restore (if doing an upgrade) is to use Fusion Shell (read chapter 6.2 or [1]) and import the data which exported in the previous chapter.

## *6.2 Daily Backup*

This is not really something this document must explain, but we add it since it is fairly straight forward. Make a script called "backup.sh" (only for Unix-OS'):

```
>mysqldump --single-transaction -uroot -pROOT-PASSWORD --all-databases >
/your_backup_dir/backup-`date +"%F_%T"`.sql
```

Then add a crontab job (you need to be root):

```
>crontab -e
```

add this to crontab file (runs at 2 am to avoid other Fusion jobs running at midnight):

```
0 2 * * * /your_backup_script_dir/backup.sh
```

This backup operation will halt any inserts/updates to the database. If the database is very large (many GB) it could take many minutes to complete. Everything about these commands could be changed, both how much of the database is backed up, when it's done and so on. This will be a task for the DBA to tune in.