# Millistream Database Application

### Technical Description and Reference Guide

14 February 2024

**Millistream Market Data AB** Box 5243, 402 24 Göteborg, Sweden www.millistream.com Org nr: 556763-4117

### Introduction

The Millistream Database Application (mda) is a application that receives data from the Millistream system via the Millistream Data Feed (libmdf) and stores it in a relational SQL database.

MDA is available for multiple Linux distributions and Windows. The source code is released under GPLv3 and should compile on most POSIX compliant operating systems (and if not we will happily accept patches/bug-reports) if a binary is not available for your specific operating system.

Native database drivers exists for MySQL/MariaDB, PostgreSQL and Microsoft SQL Server as well as a generic ODBC driver for ODBC compliant databases.

Names and availability of tables and/or columns are freely configurable and the SQL schema used is developed and released independent of MDA as such (only exception is that a minimum version of MDA might be required to support a specific table).

MDA operates four queues, each having a separate connection with the SQL server executing on separate threads. The tables handled by each queue are:

• Queue #1

quotes, quotesex. performance, keyratios, noii and greeks

• Queue #2

trades, news, newscoding, newstags and stats

• Queue #3

pricehistory, adjustedpricehistory, corporateactions, instruments, l10n, listmappings, constituents, ticksize, fundamentals, estimates, estimateshistory, brokers, ci, cihistory, priip, mifid, mifidhistory and sectors.

• Queue #4

orderbooks and mbo

Since v0.9.55 MDA will move news, newscoding and newstags from Queue #2 to Queue #4 if orderbooks and mbo are disabled in the config file, leaving Queue #2 to focus on trades.

### **Installation - Windows**

The MDA installer can be downloaded from <u>https://packages.millistream.com/Windows/</u>. To upgrade an existing installation to a new version you simply download and run the installer of the new version (the configuration and tabledef will not be overwritten).

Simply follow the on screen instructions from the installer and check the boxes for MySQL/PostgreSQL support if that is needed for your setup.

If MDA is currently running as a system service then the service will be stopped and then started again after the installation/upgrade is complete. Silent install/upgrade can be performed by using "/S" as an argument to the installer executable.

For convenience the tabledef file is named <u>tabledef.txt</u> on Windows while we name it <u>mda.tabledef</u> on every other operating system as well as in the downloadable SQL schema updates so if you see <u>mda.tabledef</u> just treat that as <u>tabledef.txt</u> on Windows.

A default configuration file, tabledef and SQL setup scripts for the various supported SQL servers are installed at the MDA root which by default is <u>C:\Program Files\Millistream\MDA\</u>.

The installer will add Start Menu entries for editing the configuration file as well as quickly starting MDA in a terminal.

A system service for MDA will also be installed, the start type is set to "manual" so as to not start automatically on boot as a surprise so if you want to run MDA as a system service you will have to open the Services Manager in Windows and change the start type of the MDA service from "manual" to "automatic".

Error and informational logs will be sent to the Event Viewer (and the terminal if MDA is run in the terminal and if not started with "--silent").

MDA hooks into Windows Performance Monitor with a number of performance counters that match the data exported with the **--status** argument.

### **Installation - Linux**

First see the information for your specific Linux distribution at <u>https://packages.millistream.com/Linux/</u> on how to add our deb/rpm repository to your system.

After that you can simply install MDA by typing "apt install mda" or "yum install mda" depending on if you use deb or rpm (you might also have to prepend the statements with "sudo" if you are not root).

Then you need to install the native sql server drivers for the sql server that you plan to use. If you run MDA on the same server as your SQL server then the native drivers are usually already installed and you can skip this step.

The package name of the drivers differ between distributions and e.g some have replaced the mysql libraries with the mariadb equivalents.

MySQL/MariaDB: install either "libmariadb" or "mariadb-libs".

PostgreSQL: install "libpq" or "postgresql-libs".

**ODBC:** install both "unixodbc" as well as the ODBC driver for your SQL server. For connecting with Microsoft SQL Server, see here for information on how to install the ODBC driver: <u>https://docs.microsoft.com/en-us/sql/connect/odbc/linux-mac/installing-the-microsoft-odbc-driver-for-sql-server?</u> <u>view=sql-server-2017</u>

Depending on distribution either a SysV Init script, Upstart Script or systemd unit file will be installed for running MDA as a daemon. On systemd the unit file is disabled by default and have to be enabled by "systemctl enable mda.service" if you want MDA to start automatically after boot.

If your distribution uses Apparmor, a apparmor profile for MDA that will further lock down which files MDA can access on your system. This can be important to remember in case you decide to change the name of the configuration file, tabledef file or other file locations since the apparmor profile will deny access to those unless you also change the apparmor profile to allow the new names.

A default configuration file and tabledef will be installed to <u>/etc/mda.config</u> and <u>/etc/mda.tabledef</u> and SQL setup scripts for the various supported SQL servers will be installed to <u>/usr/share/doc/mda/</u>.

Man pages for "mda" and "mda.config" are also installed for convenience.

### **Installation - From Source**

The source code can be downloaded from <u>https://packages.millistream.com/source/</u>. Build dependencies are libexpatdev, libhashit-dev, libmdf-dev, libmysqlclient-dev, libpq-dev and unixodbc-dev.

Everything is built with gcc (on Windows we build everything on MSYS2) except mod\_mssql which requires Visual Studio due to it being OLE DB.

### Invocation

To list the available options to mda simply execute "mda --help". The supported options are:

Option	Description	
config	Specify the name and path of the configuration file. On Linux the default is to first look for <u>mda.config</u> in the current directory and the for <u>/etc/mda.config</u> . On Windows the default is to first look for <u>mda.config</u> in the current directory and then for <u>mda.config</u> in the install directory as set in the registry under "HKLM\Software\Millistream\MDA\Install Directory".	
check	Connect with the SQL server and check if all tables and columns defined in the tabledef are present on the SQL server.	
check-types	Behaves likecheck but also controls the data types of the columns.	
upgrade-db	Execute 'ALTER TABLE' statements on the SQL server in order to try and fix the problems reported by <b>check</b> or <b>check-types</b> . Missing tables and indexes cannot be fixed by mda so for that you will have to run the SQL scripts instead. Must be used together with either <b>check</b> or <b>check-types</b> to inform mda which type of problems it should try and correct.	
import	Specify the path and name of a file to read SQL statements from. This can be used in order to execute SQL dumps if your SQL server does not provide a good tool for doing so. MDA supports reading from Gzipped files (.gz) and if "-" is given as path/name then MDA will read the SQL statements from <b>stdin</b> .	
silent	Causes mda to redirect <b>stdin</b> , <b>stdout</b> and <b>stderr</b> to <u>/dev/null</u> . Used when mda is running as a system daemon and where logging to syslog/EventViewer is sufficient.	
daemon	Causes mda to background itself (silent is also enabled). Only available on Linux.	
status	Write various internal status variables and statistics from a running mda to <b>stdout</b> and exit.	
map-name	Specify the name of the memory mapped file used to store the statistics data. This must be used to set a different name for each instance if multiple instances of MDA is running on the same server. The default is <u>/millistream-mda-map</u> on Linux and <u>Global\Millistream Database</u> <u>Application Perfcounters</u> on Windows.	

## Configuration

The file format of the configuration file is similar to other Unix configuration files; comments begin with a "#" character and extends to the end of the line and blank lines are ignored. Initial whitespaces are ignored but are present in the supplied default configuration file for readability.

Configuration commands consist of an initial keyword followed by a value, separated by whitespace(s). Commands may not be continued over multiple lines. An empty keyword (i.e one that have no value) defines a **section** and are case insensitive while keywords with a value defines a **setting** and are case sensitive.

As of v0.9.52 if the first character in a value is a **\$** then MDA will lookup the actual value from a environment variable with the same name as that of value minus the **\$** character. So "username **\$**UNAME" will fetch the value for username from the environment variable named UNAME.

Section: MILLISTREAM		
Setting	Value	
username	Username to authenticate with the Millistream distribution node.	
password	Password to authenticate with the Millistream distribution node.	
server	Millistream distribution node to connect to in "host:port" format. host can be a host name, a fully qualified domain name, a IPv4 address or a IPv6 address. If host is a IPv6 address then it has to be enclosed in square brackets such as: [2001:db8::1428:57ab]. If multiple "host:port" pairs are defined (comma separated list), then mda will try each in order when connecting.	
reconnectinterval	The number of seconds to wait between reconnect attempts, default 0.	
instrumenthashsize	mda maintains an internal hash table for the <b>orderbooks</b> table in memory. The currently used number of items in this hash table can be seen by the instrument-hash-size value from the <b>status</b> option to mda where the output is "current/maximum" and with this setting the "maximum" part can be set, default 100000.	
queuehashsize	mda maintains a hash table per queue in order to quickly lookup the location of a insref in the queue. The currently used number of items can be seen by the qX-queue value from the <b>status</b> option to mda since most items on the queue occupies one item in the hash, default 100000.	
requestXXXX	The requestXXXX settings tell mda which messages to request and subscribe to from the Millistream distribution node, if mda can request the message(s) or not is controlled by the entitlements on the account used to log on. Also there have to be a definition for the message in question in the tabledef file or mda will not perform the request regardless of the settings. The default value for all these are "1", i.e to perform the request. So the main use of these settings are to disable data for various tables if you so desire (by setting the value to "0").	
requestrades	Subscribe to and populate the <b>trades</b> table.	
requestorderbooks	Subscribe to and populate the <b>orderbooks</b> table.	
requestquotes	Subscribe to and populate the <b>quotes</b> table with Last-Sale info.	
requestquotesbbo	Subscribe to and populate the <b>quotes</b> table with BBO info.	
requestquotesex	Subscribe to and populate the <b>quotesex</b> table with Extra Session info.	
requestpricehistory	Subscribe to and populate the <b>pricehistory</b> and <b>adjustedpricehistory</b> tables. For the prices in the <b>adjustedpricehistory</b> table to be adjusted you must also enable <i>requestcorporateactions</i> .	
requestnews	Subscribe to and populate the <b>news</b> , <b>newscoding</b> and <b>newstags</b> tables.	
requestbasicdata	Subscribe to and populate the <b>instruments</b> table.	
requestcorporateactions	Subscribe to and populate the <b>corporateactions</b> table.	

requestfundamentals	Subscribe to and populate the <b>fundamentals</b> table.	
requesttradestate	Subscribe to and populate the <b>tradestate</b> column in the <b>instruments</b> table.	
requestperformance	Subscribe to and populate the <b>performance</b> table.	
requestkeyratios	Subscribe to and populate the <b>keyratios</b> table.	
requestestimates	Subscribe to and populate the <b>estimates</b> and <b>estimateshistory</b> tables.	
requestnoii	Subscribe to and populate the <b>noii</b> (Net Order Imbalance Indicator) table.	
requestl10n	Subscribe to and populate the <b>l10n</b> (Localization) table.	
requestci	Subscribe to and populate the <b>ci</b> and <b>cihistory</b> (Corporate Information) tables.	
requestpriip	Subscribe to and populate the <b>priip</b> (European PRIIPs Template EPT) table.	
requestmifid	Subscribe to and populate the <b>mifid</b> (European MiFID Template EMT) and <b>mifidhistory</b> tables.	
requestmappings	Subscribe to and populate the <b>sectors</b> table.	
requestmbo	Subscribe to and populate the <b>mbo</b> table.	
requestgreeks	Subscribe to and populate the <b>greeks</b> table.	
insrefs	A comma separated list of the insrefs to subscribe to, if set to "*" or if commented out, mda will subscribe to all available insrefs.	

Section: DATABASE		
Setting	Value	
instance	Specify the name and path of the SQL plugin to use, currently there exists plugins for PostgreSQL, MySQL/MariaDB, ODBC and Blackhole (a plugin that does nothing, used internally for benchmarking). By default the plugins are installed to <i>/usr/lib/mda/</i> . Note that the MySQL/MariaDB plugin can read options from the mda group in the <i>/etc/mysql/my.cnf</i> file.	
updateinterval	Specifies the minimum time between two updates for the same instrument to the same table for Queue #1 (Queues #2 and #3 are hardcoded to 0) in number of seconds.	
connectionstring	The connection string to use for the SQL plugin.	
domain	The domain to use if Windows Domain Account Authentication is to be used and a different account than the one mda is started on should be used. A dot is used as an alias for "the local computer". Only available on Windows.	
domainuser	The username to use if Windows Domain Account Authentication is to be used and a different account than the one mda is started on should be used. Only available on Windows.	
domainpassword	The password to use if Windows Domain Account Authentication is to be used and a different account than the one mda is started on should be used. Only available on Windows.	
keepdeleted	If set to "1" mda will set the <b>tradestate</b> column in the <b>instruments</b> table to "9" when it receives a <b>Instrument Delete</b> message in the feed instead of deleting it from the database, default value is "1".	
checkdatabase	If set to "1" mda will perform the same check as if executed with the <b>check</b> option upon start and if the check fails for whatever reason mda will immediately terminate, default value is "1".	
localtime	If set to "1" then timestamps are converted from UTC to the local time zone of the machine where mda runs. Do note that the timestamps in tables <b>pricehistory</b> , <b>adjustedpricehistory</b> and <b>corporateactions</b> are never converted since the time column there are strictly to enforce order of events and are not to be used as the time of the closing prices, default value is "0".	
deletelimit	Upon start and if <i>requestbasicdata</i> is enabled, mda will delete instruments that have been deleted during the time mda where disconnected. Only insrefs with a value < limit will be	

	deleted so with this setting you can protect internal/private insrefs from being deleted, also setting limit to "0" effectively disables this functionality. As a protection mda will not perform the delete if the number of insrefs to delete exceeds <i>maxdeletes</i> . Default value is "18446744073709551615".
maxdeletes	Sets the safety limit for the <i>deletelimit</i> functionality described above.
transactions	Maximum number of SQL statements to use per transaction. If the value is < 2 then transactions are disabled, default value is "1".
truncatetime	How to handle timestamps with subsecond resolution. If "0" allow full "HH:MM:SS:mmmuuu", if "1" truncate to "HH:MM:SS" and if "2" truncate to "HH:MM:SS:mmm". Default value is "0".
aggtrades	If set to "1" then mda will aggregate trades on a one trade per minute basis into the <b>aggtrades</b> table, the source trades in the <b>trades</b> table will be left untouched. Useful if you want to create data to draw intraday ticks on but don't want to be swarmed by millions of updates per second.
disableupsert	Since MDA v0.9.56, MDA will perform upsert statements for tables that are defines as being insertfirst=yes in the tabledef. Setting this to "1" or "Y" will revert back to the old behavior where MDA would first try with an INSERT and then execute a UPDATE if the INSERT failed.

Section: APPLICATION		
Setting	Value	
tabledefinitions	Specify the name and path of the tabledef file to use, default <u>/etc/mda.tabledef</u> on Linux and to <u>\$INSTALLPATH\tabledef.txt</u> on Windows.	
pidfile	Specify the name and path of the file used to record the mda process ID. Main use is for users of the SysV Init scripts since neither Upstart nor systemd needs this. Only available on Linux.	
user	Specify a Unix account user to switch to. Only available on Linux.	
group	Specify a Unix account group to switch to. Only available on Linux.	

Section: LOG		
Setting	Value	
logsqlpath	If set mda will log all executed SQL statements to files at \$PATH/sqYYMMDD.log, default is to disable logging.	
facility	Defines the syslog facility to log as, default is to disable syslog (i.e no facility).	
host	Syslog server to send the syslog messages to. The default value is localhost. Only available on Windows.	
debug	If set to "1" then debug logging is enabled (mostly orderbook related errors), default is "0".	

### Tabledef / SQL Schema

The tabledef file works as a mapping between the Millistream Data Feed (libmdf) and the SQL database, in short it describes the SQL schema as far as mda is concerned. Tables and/or columns that exist in the SQL server that is not described in the tabledef is completely ignored by mda (and is not checked on --**check** or --**check-types**) allowing users to customize the mda tables and columns to their liking.

The format is plain XML and should normally not be edited by end users but can be used to rename columns and tables as well as remove unwanted columns/tables or to insert new tables/columns as part of upgrading the SQL schema.

The first line in the file describes the version of the SQL schema defined in the file:

<definitions version="94">

This example tells that the tabledef file describe version 94 of the SQL schema, this can also be seen in **--status** in the <u>templates-version</u> option where the value is in the format: "**A**/**B**". **A** here is the version from the tabledef file that mda loaded at startup and **B** is the value sent by the feed to indicate the current SQL schema version available.

New versions of the SQL schema are published as ZIP archives at

<u>https://packages.millistream.com/documents/MDA\_schema\_XX.zip</u> where XX is the version number of the new SQL schema. The archive contains both a new tabledef file as well as upgrade scripts for the SQL servers, consult the <u>README.txt</u> file in the archive for further information.

Upgrading the SQL schema is only needed if you find the new tables/columns useful. There is no need to upgrade the SQL schema when upgrading MDA or for any other reason than to access the new data.

Tables and columns are defined like this (this is an example of the **ticksize** table):

```
<message mref="24" table="ticksize" >
<field tag="insref" column="insref" key="1" type="bigint" nullable="0" unsigned="1" />
<field tag="1" column="threshold" key="1" type="double" nullable="0" />
<field tag="2" column="increment" type="double" nullable="0" />
```

</message>

Tables are defined with the <message> elements and the colums for that table with <field> elements. In order to change the name of the table/column used in the SQL server you can changed the table and column attributes.

The mref and tag attributes are used by mda to cross reference with the MDF MessageReference (MDF\_M\_) and FieldReference (MDF\_F\_) definitions. These values can be used when looking up message and field definitions from the MDF Fields Reference Guide found at <a href="https://packages.millistream.com/documents/MDF Fields Reference.pdf">https://packages.millistream.com/documents/MDF Fields Reference.pdf</a>

The key attribute informs mda which columns it should use in the WHERE clause for a UPDATE query while type, nullable and unsigned are used only by --check-types when verifying that the data types in the SQL server matches those from the tabledef.

### **SQL Server Drivers / Plugins / Modules**

In order to support different SQL servers, mda uses a plugin mechanism of modules in the form of shared libraries (.so files on Linux and .dll on Windows). Which module to use is determined by the <u>instance</u> setting in the **DATABASES** section of the configuration file.

### mod\_mysql (MySQL / MariaDB):

The mod\_mysql module connects to a MySQL/MariaDB SQL server using the native libmysqlclient library. Since the libmysqlclient library does not handle connection strings, the <u>connectionstring</u> setting is parsed by mda according to the format specified at <u>https://docs.microsoft.com/en-us/previous-versions/windows/desktop/ms722656(v=vs.85)</u>.

Please note that we have included two SQL schema scripts for MySQL/MariaDB in the MDA install. One for MySQL versions prior to v5.6.4 and one for v5.6.4 or newer (which includes all versions of MariaDB). The reason for this is that MySQL prior to v5.6.4 did not support *time* columns with subsecond values and would not accept the *time(6)* syntax so a script with just *time* was required.

Typical connection string when the MySQL/MariaDB is running on the same machine as MDA would be: connectionstring host=localhost; dbname=millistream; user=<username>; password=<password>;

#### Keyword Option MYSQL\_OPT\_SSL\_KEY sslkey MYSQL\_OPT\_SSL\_CERT sslcert MYSQL\_OPT\_SSL\_CA sslrootcert MYSQL\_OPT\_SSL\_CAPATH sslcapath MYSQL\_OPT\_SSL\_CIPHER sslcipher MYSQL\_OPT\_SSL\_CRL sslcrl MYSQL\_OPT\_SSL\_CRLPATH sslcrlpath sslmode MYSQL\_OPT\_SSL\_MODE MYSQL\_OPT\_CONNECT\_TIMEOUT connect\_timeout MYSQL\_OPT\_CONNECT\_TIMEOUT connection timeout MYSOL OPT WRITE TIMEOUT write\_timeout MYSQL\_OPT\_WRITE\_TIMEOUT write timeout

### Supported keywords used for mysql\_options() from

https://dev.mysql.com/doc/refman/8.0/en/mysql-options.html

# Supported keywords used for mysql\_real\_connect() from https://dev.mysql.com/doc/refman/8.0/en/mysql-real-connect.html

Keyword	Argument
host	host
server	host
port	port or unix_socket if the value starts with a "/" character
dbname	db
database	db
data source	db

# MILLISTREAM :::::•

datasource	db
user	user
uid	user
user id	user
user name	user
username	user
password	passwd
pwd	passwd

### mod\_postgresql (PostgreSQL):

The mod\_postgresql module connects to a PostgreSQL SQL server using the native libpq library. The connection string is sent unmodified to libpq and the supported format and keywords can be found at https://www.postgresql.org/docs/current/static/libpq-connect.html#LIBPQ-CONNSTRING

Typical connection string when password authentication is used and the PostgreSQL server is on the same machine as MDA would be: connectionstring host=localhost dbname=millistream user=<username> password=<password>

A common issue on PostgreSQL is that people create a *mda* user on PostgreSQL for MDA but then execute the SQL script with the *postgres* user leading to *postgres* being the owner of the tables and the *mda* user not having the necessary rights to access the tables in the *millistream* database.

If you encounter this problem (MDA will exit with "**Database check failed. Error '(null)', Code: 0**" which indicates that it got back 0 rows from it's **SELECT** to *information\_schema.columns*) then you have to execute this SQL statement for each and every table in the *millistream* database:

**GRANT SELECT, INSERT, UPDATE, DELETE TABLE** instruments **TO USER** mda;

### **mod\_mssql** (Microsoft SQL Server using OLE DB): (*legacy driver, should not be used*).

The mod\_mssql module connects to a Microsoft SQL Server server using the OLE DB interface. This module is only available on Windows and to connect to a MSSQL server on Linux we recommend the ODBC module instead, also on Windows we highly recommend that you use the ODBC module instead since the OLE DB interface is a legacy interface and that this module have no deadlock detection (so all updates in a transactions will be completely lost in case a deadlock occurs).

The connection string is sent unmodified to the OLE DB interface and must contain a "Provider=<xxx>;" where <xxx> tells which OLE DB provider to use, the current most recent version is MSOLEDBSQL but the availability of OLE DB providers depend on which providers that are installed on the machine in question.

If Windows Domain Account Authentication is used over SQL Authentication you must add "Trusted\_Connect=yes;" to the connection string to inform OLE DB which form of authentication to use.

Typical connection string when the MSSQL server is running on the same machine as MDA would be: connectionstring Provider=MSOLEDBSQL; Data Source=.; Initial Catalog=millistream; uid=<username>; pwd=<password>;

### mod\_odbc (Any ODBC compliant SQL server):

The mod\_odbc module connects to a ODBC compliant SQL server using the ODBC interface. While it has been tested against MySQL, MariaDB and PostgreSQL we highly recommend to use the native modules for those servers instead since they are both more efficient and stable than their ODBC drivers. The main use for the ODBC module is to connect to a Microsoft SQL Server.

The connection string is sent unmodified to the ODBC interface and numerous examples of connection strings for

various SQL servers can be found at https://www.connectionstrings.com/.

For convenience, DSN:s can be setup using the ODBC Data Source Administrator tool on Windows or in the <u>/etc/odbc.ini</u> and <u>/etc/odbcinst.ini</u> files on Linux. Without a DSN the connection string must contain a *Driver* keyword that specifies which ODBC driver that is to be used, e.g "Driver={ODBC Driver 17 for SQL Server};".

For MySQL either the "*Return matched rows instead of affected rows*" setting when configuring the DSN must be enabled or "FOUND\_ROWS=1;" must be added to the connection string. For MariaDB the setting must be done when configuring the DSN since this setting is not possible to set with a connection string.

## **Tables and Columns**

Since it has proven to be unsustainable to maintain a complete list of available tables and columns in parallel documents we have decided to omit a complete table/column definition here. Instead we refer all users to the MDF Fields Reference Guide from <a href="https://packages.millistream.com/documents/MDF Fields Reference.pdf">https://packages.millistream.com/documents/MDF Fields Reference.pdf</a> which contains detailed information on each individual field in the Millistream Data Feed, for the vast majority of fields the column name matches the field name (e.g lastprice is named MDF\_F\_LASTPRICE) and where it does not the **mref** and **tag** is available in the tabledef.

The MDF Messages Reference Guide from <u>https://packages.millistream.com/documents/MDF Messages Reference.pdf</u> can also be used in order to see which fields are available for each message, in mda a message is usually handled as a table and a field is handled as a column.

The ci and cihistory tables are further described in a separate document at

<u>https://packages.millistream.com/documents/Company Information.pdf</u> and the format of the **text** column from the **news** table is described in this document <u>https://packages.millistream.com/documents/News Format.pdf</u>

MDA Table Name	MDF Message	Minimum version of MDA
instruments	MDF_M_BASICDATA, MDF_M_TRADESTATE	0.0.8
quotes	MDF_M_QUOTE	0.0.8
trades	MDF_M_TRADE	0.0.8
aggtrades	Built using sql statements to the trades table.	0.9.55
pricehistory	MDF_M_PRICEHISTORY	0.0.8
orderbooks	MDF_M_BIDLEVELINSERT, MDF_M_ASKLEVELINSERT, MDF_M_BIDLEVELDELETE, MDF_M_ASKLEVELDELETE, MDF_M_BIDLEVELUPDATE, MDF_M_ASKLEVELUPDATE	0.0.8
news	MDF_M_NEWSHEADLINE, MDF_M_NEWSCONTENT	0.0.8
corporateactions	MDF_M_CORPORATEACTION	0.0.8
fundamentals	MDF_M_FUNDAMENTALS	0.2.0
estimates	MDF_M_ESTIMATES	0.9.24
estimateshistory	MDF_M_ESTIMATESHISTORY	0.9.25
performance	MDF_M_PERFORMANCE	0.2.00
keyratios	MDF_M_KEYRATIOS	0.3.00, 0.9.47 for the new calculated columns in keyratios from SQL schema v100.
noii	MDF_M_NETORDERIMBALANCE	0.9.27
ci	MDF_M_CI	0.9.33
cihistory	MDF_M_CIHISTORY	0.9.33
l10n	MDF_M_L10N	0.9.28
priip	MDF_M_PRIIP	0.9.41
mifid	MDF_M_MIFID	0.9.41
mifidhistory	MDF_M_MIFIDHISTORY	0.9.44

sectors	MDF_M_MAPPINGS	0.9.45
greeks	MDF_M_GREEKS	0.9.50
mbo	MDF_M_MBOADD, MDF_M_MBOUPDATE, MDF_M_MBODELETE	0.9.50
quotesex	MDF_M_QUOTEEX	0.9.51

The **adjustedpricehistory** table is a mda specific table where mda combines the prices from the **pricehistory** table with the adjustment factors from the **corporateactions** table in order to provide easy access to adjusted price history. Added in v0.4.0.

The **performance** table contains mda specific columns as **diffXX** and **diffXXprc** which are calculated from **quotes.lastprice** (or **quotes.lastyield** for the Yield columns and **quotes.lastnav** for the Price columns if **instruments.pricetype** is "3" for the instrument in question) and the corresponding

**performance.closepriceXX/performance.closeyieldXX** columns. This is done to calculate the realtime net change and net change in percentage values. Added in v0.2.0.

The **keyratios** table contains mda specific columns (they have the **calc="1**" attribute set in the tabledef) which are calculated from **quotes.lastprice**. These columns require at least v0.9.47 to work and where introduced in SQL schema v100.

Further MDA contains a number of mda specific tables that are maintained from the tabular field values or just as cross reference tables for convenience.

Table: <b>listmappings</b> Cross reference table between lists and instruments. Added in v0.0.8		
Column	Description	
insref	The insref of the instrument.	
list	The insref of the list.	

Table: constituents
Cross reference table between indexes and their constituents. Added in v0.9.27

Column	Description
insref	The insref of the index.
constituent	The insref of the instrument.
weight	Optional weight in percentage of the instrument in the index.

able: <b>ticksize</b> cross reference table between tick size tables and the actual tick sizes. Added in v0.9.27	
Column Description	
insref	The insref of the tick size table.
threshold	The price threshold at which the increment takes effect.
increment	The minimum price increment at this price threshold.

### Table: **brokers**

Table over the available brokers for a marketplace. Added in v0.9.31		
Column	Description	

insref	The insref of the marketplace.
symbol	The symbol/id for the broker as used by the marketplace as buyer/seller in trades.
name	Name of the broker

### Table: newscoding

Cross reference between news items and instruments/companies. Added in v0.5.0

Column	Description
newsref	The newsref/insref of the news agency.
newsid	The newsid of the news item.
isin	The ISIN of an instrument. A single news item can have multiple ISINs each having their own row.
company	The insref of a company instrument, values of "0" should be ignored. A single news item can have multiple company insrefs each having their own row.
date	The date of the news item.

Table: newstags     Cross reference table between news items and their corresponding tags. Added in v0.9.42	
Column Description	
newsref	The newsref/insref of the news agency.
newsid	The newsid of the news item.
tag	A news tag defined for the news item. A single news item can have multiple news tags each having their own row.
date	The date of the news item.

### Table: underlyings

Cross reference table between instruments and underlying instruments, used since an instrument can have multiple underlying instruments. Added in v0.9.41

Column	Description
insref	The insref of the instrument.
isin	The ISIN of the underlying instrument.
currency	The currency of the underlying if known to us.

Table: <b>sectors</b> Cross reference table between instruments and classification sectors. Added in v0.9.45		
Column	Description	
insref	The insref of the instrument	
sector	The insref of the classification sector	
marketplace	The insref of the marketplace of the instrument, mostly present for internal use by the MDA.	

Table: <b>combolegs</b> Specifies the legs for a Combo or Strategy contract. Added in v0.9.51	
Column	Description
insref	The insref of the Leg
Leg Ratio	Ratio of lots for the leg
Leg Side	0 – Buy, 1 – Sell

### Table: stats

Various statistics updated by mda every 5 seconds (if the value have changed since last update). Added in v0.1.0

Column	Description
name	Name of the statistic
value	Value of the statistic

### List of statistics from the **stats** table:

Name	Description
Connected	"1" if mda is connected to a Millistream distribution node and "0" if not.
Databasequeue	Total combined number of items in all queues.
Databaseupdates	Total combined updates done since start from all queues to the SQL server.
Databaseupdates 1	The number of updates done since start by Queue #1 to the SQL server.
Databaseupdates 2	The number of updates done since start by Queue #2 to the SQL server.
Databaseupdates 3	The number of updates done since start by Queue #3 to the SQL server.
Failed databaseupdates	Total combined number of failed writes by all queues to the SQL server.
Last write	Local time and date of when any queue performed an update to the SQL server.
Queue 1	Number of queued items in Queue #1.
Queue 2	Number of queued items in Queue #2.
Queue 3	Number of queued items in Queue #3.
Running	Set to "1" when mda starts and "0" when it shuts down.
Startup time	Local time and date of when mda was started.
Strangled	The total combined number of deduplicated items from all queues since startup.
Templates Version (Feed)	The current SQL Schema version as advertised by the Millistream Data Feed (libmdf).
Templates Version (Loaded)	The current SQL Schema version as loaded by mda from the tabledef.
Transactions	The total received number of messages from the Millistream distribution node since startup.