ATSDB

User Guide

Maintained by Helmut Puhr

Version 0.0.9 *Befuddled Badger*

Short contents

Contents

List of Figures

List of Tables

Introduction

This document has its focus on interaction and working procedures required to make use of the existing functionality. In this introduction, feature highlights are listed, followed by a brief summary of important aspects of ATSDB. In the later section Usage, a functional, task-oriented overview is given.

FEATURE HIGHLIGHTS

The **A**ir **T**raffic **S**urveillance **D**ata**B**ase aims at providing a generalized framework for ATM surveillance data inspection. While its current functionality is somewhat limited, the following features exist:

- Support of multiple database systems, e.g. SCDB
- High performance processing, low memory footprint
- Utilization of application during loading procedure
- Views for data inspection
- Simple custom filter generation
- Supported Database Objects
  - Radar plots
  - System Tracks and Reference Trajectories
  - MLAT & WAM target reports
  - ADS-B target reports
- XML-based configuration files
- Multiple coexisting configurations, usage chosen during runtime
- Based on Open Source libraries
- Runs on generic hardware

GENERAL ASPECTS

ATSDB is a highly specialized surveillance data processing framework, with a strong focus on high-performance and a low memory footprint, to process massive quantities of data. Surveillance data is fetched from a database (limited by a filter system), then processed and displayed using so-called Views (visualization of aspects of the result set).

As storage medium, a database is used. Different database systems are supported, and a flexible read-out system allows for easy adaptation to different database schemas. Data in such a database has to be generated in a previous, separate process. One method would be to use EUROCONTROLS SASS-C Verif V7/8 framework.

When such a previously generated database is opened for the first time, some post-processing is performed, to ease usage and to increase startup speed. When data is loaded using a database query, a filter configuration may restrict the data leading to a result set. Such a result set can be analyzed using Views, e.g. the Listbox view.

Each View defines which parts of the database are required to fulfill its purpose, and only such parts are loaded. During a loading process from the database, subsets of the query result are immediately added to the current result set and all views are updated.

Key Concepts

In this section, a few key concepts are introduced to give a somewhat deeper understanding of ATSDB and to allow the reader to understand some main design choices made by the author. This should also give indications about the strengths and draw-backs of the chosen approach.

Database Systems

A database allows for storage, retrieval and filtering of the data of interest. While SQL has some definitive drawbacks, it was chosen since support of the SASS-C database SCDB was wanted.
Currently two database systems are supported: MySQL and SQLite3. MySQL relies on an independent background process, which holds several databases and can also be accessed over a network connection.
SQLite3 encapsulates one database in a single file container, which is read from a storage medium (e.g. hard drive).

Configuration

At startup, several configuration files are loaded, and at shutdown the current configuration state of ATSDB is saved. But configuration is not just a matter of components having the same parameters, but also what components exist. To give an example: Each existing View is saved, and when the program is started again, the previously active Views are created. The same holds for the filters, or the database interface/schema.
This way, a user can have a specific program configuration for a specific usage situation, which can be instantly reused for a different dataset, using have a completely configurable database schema or filter configuration.
This allows for a high degree of flexibility, but somewhat complicates software development.

Flexible Database Interface

Using such a configuration, a flexible database interface method was implemented to allow general displaying of data in different database systems and schemas. How this was done would require a detailed discussion, which will be skipped for the moment.
To summarize, several database schemas can be stored in configuration files, each of which

is a structured collection of database tables and their logical dependencies. Such information is used in one set of Database Objects (DBOs). In each database system, any database schema can be used.

Database Objects

A Database Object (DBO) is defined by a name and has a collection of variables. For example, radar plots and system tracks are database objects, and each has variables holding time, position, Mode 3/A codes and Mode C heights and so on. From a database, if such a DBO is present, it can be loaded and displayed.

To allow displaying data from different DBOs in the same system, so-called meta-variables were introduced, which hold variables that are present in some or all DBOs (with a possibly different name or unit). For example, there meta-variable 'tod', which is a collection of sub-variables for each existing DBO and the respective "Time of Day" variable.

Data Loading

In ATSDB, a unified data loading process was chosen, meaning that only exactly one common dataset is loaded, which can be inspected using Views. When started, data is incrementally read from the database, stored in the resulting dataset, and distributed to the active Views. Each time such a loading process is triggered, all Views clear their dataset and gradually update.
This makes working with the data somewhat easier to understand, since only one dataset exists, while on the other hand it does not allow several independent datasets (e.g. with different filters) to be loaded.

Generating a Database

A database has to be generated before it can be opened with ATSDB. This section describes how such a process can be performed, however it is by no means a complete guide.

SASS-C Verif

A complete treatment of how to generate a database using the highly sophisticated SASS-C Verif frame work is out of the scope of this document. However, a short summary of the necessary steps will be presented here.

- Import a previous evaluation job
- Set data recording path, e.g. 'somelocation/%iffile.if'
- Run IRIS command
- Run OTR command
- Run CMP command with all but RA
- Run CMP command with just RA

After these steps, a database was generated and filled with data. The name of the database (which will be needed during the opening process) is equivalent to the job name, e.g. ’job_v7_mainsacso_0005’, ’HelloWorld’ etc.

Please **note** that the SASS-C MySQL database name is not case sensitive (’job_v7_mainsacso_0005’ is the same as ’JOB_V7_MAINSACSO_0005’).

Installation

Currently, ATSDB has to be installed as source code and compiled by the user. Any recent Linux installation should work without issues. Regarding software version, no particularly new features were used, so older versions than the verified ones should also work.

The following software has to be installed on the workstation:

Package	**Description**	**Verified version**
g++	C++11 capable C++ compiler	7.2.0
cmake	CMake build tool	3.9.1
qt5-default	Qt5 development package	5.9.1
libboost-dev	Boost development libraries	1.62.0.1
mysql++-dev	MySQL library bindings	3.2.2
libmysqlclient-dev	MySQL development library	1.0.2
libsqlite3-dev	Sqlite3 development files	3.2.2
libgdal-dev	Geospatial data abstraction library	2.2.1
tinyxml2-dev	XML parsing library	5.0.1
log4cpp5-dev	Logging libary	1.1.1
doxygen	Documentation generation	1.8.13

Table 1: Required software/libraries

To install to source code, either download the latest released version from
`https://github.com/hpuhr/ATSDB/releases`
or use the following command (git required) to clone the current repository:

```
git clone https://github.com/hpuhr/ATSDB.git
```

Enter the ATSDB source folder, and execute cmake to create a Makefile:

```
cmake .
```

The output should look like this:

```
sk@golem:~/test/ATSDB$ cmake .
  Path: /home/sk/test/ATSDB
-- The C compiler identification is GNU 7.2.0
-- The CXX compiler identification is GNU 7.2.0
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Check for working CXX compiler: /usr/bin/c++
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
  System: Linux-4.13.0-16-generic
  Install Path: /home/sk/test/ATSDB/dist
  Platform: Linux
-- Looking for pthread.h
-- Looking for pthread.h - found
-- Looking for pthread_create
-- Looking for pthread_create - not found
-- Looking for pthread_create in pthreads
-- Looking for pthread_create in pthreads - not found
-- Looking for pthread_create in pthread
-- Looking for pthread_create in pthread - found
-- Found Threads: TRUE
-- Boost version: 1.62.0
-- Found the following Boost libraries:
--   regex
--   system
--   thread
--   chrono
--   date_time
--   atomic
  Boost_INCLUDE_DIR: /usr/include
  Boost_LIBRARY_DIR:
  CMAKE_MODULE_PATH: /home/sk/test/ATSDB/cmake_modules
-- Found MySQL: /usr/lib/libmysqlpp.so
  MySQLpp_INCLUDE_DIR: /usr/include/mysql++
  MySQLpp_LIBRARY: /usr/lib/libmysqlpp.so
-- Found MySQL: /usr/lib/x86_64-linux-gnu/libmysqlclient.so
  MYSQL_INCLUDE_DIR: /usr/include/mysql
-- Found Sqlite3 header file in /usr/include
-- Found Sqlite3 libraries: /usr/lib/x86_64-linux-gnu/libsqlite3.so
  SQLite3_INCLUDE_DIR: /usr/include
  SQLite3_LIBRARY_DIR: /usr/lib/x86_64-linux-gnu/libsqlite3.so
-- Found Log4CPP
  LOG4CPP_INCLUDE_DIR: /usr/include
  LOG4CPP_LIBRARY: /usr/lib/x86_64-linux-gnu/liblog4cpp.so
```

```
-- Found GDAL: /usr/lib/libgdal.so
  GDAL_INCLUDE_DIRS: /usr/include/gdal
  GDAL_LIBRARIES: /usr/lib/libgdal.so
  TINYXML2_INCLUDE_DIR: /usr/include
  TINYXML2_LIBRARY: /usr/lib/x86_64-linux-gnu/libtinyxml2.so
-- Found Doxygen: /usr/bin/doxygen (found version "1.8.13") found components:  doxygen
CPACK_SOURCE_IGNORE_FILES = /CMakeFiles/;/_CPack_Packages/;/dist/;/.git/;
-- Configuring done
-- Generating done
-- Build files have been written to: /home/sk/test/ATSDB
```

Then, compile the source by executing:

```
make
```

The output should look like this:

```
Scanning dependencies of target atsdb_autogen
[  1%] Automatic MOC for target atsdb
Generating MOC predefs moc_predefs.h
Generating MOC source 2MJGWJB4P3/moc_mainwindow.cpp
...
Generating MOC source 3JYSCE0BDA/moc_viewmanagerwidget.cpp
Generating MOC compilation mocs_compilation.cpp
[  1%] Built target atsdb_autogen
Scanning dependencies of target atsdb
[  2%] Building CXX object CMakeFiles/atsdb.dir/src/atsdb.cpp.o
[  2%] Building CXX object CMakeFiles/atsdb.dir/src/buffer/arraylist.cpp.o
...
[ 93%] Linking CXX shared library dist/lib/libatsdb.so
[ 93%] Built target atsdb
Scanning dependencies of target atsdb_client_autogen
[ 94%] Automatic MOC for target atsdb_client
Generating MOC predefs moc_predefs.h
Generating MOC source 2MJGWJB4P3/moc_mainwindow.cpp
...
Generating MOC source 3JYSCE0BDA/moc_viewmanagerwidget.cpp
Generating MOC compilation mocs_compilation.cpp
[ 94%] Built target atsdb_client_autogen
Scanning dependencies of target atsdb_client
...
[100%] Linking CXX executable atsdb_client
[100%] Built target atsdb_client
```

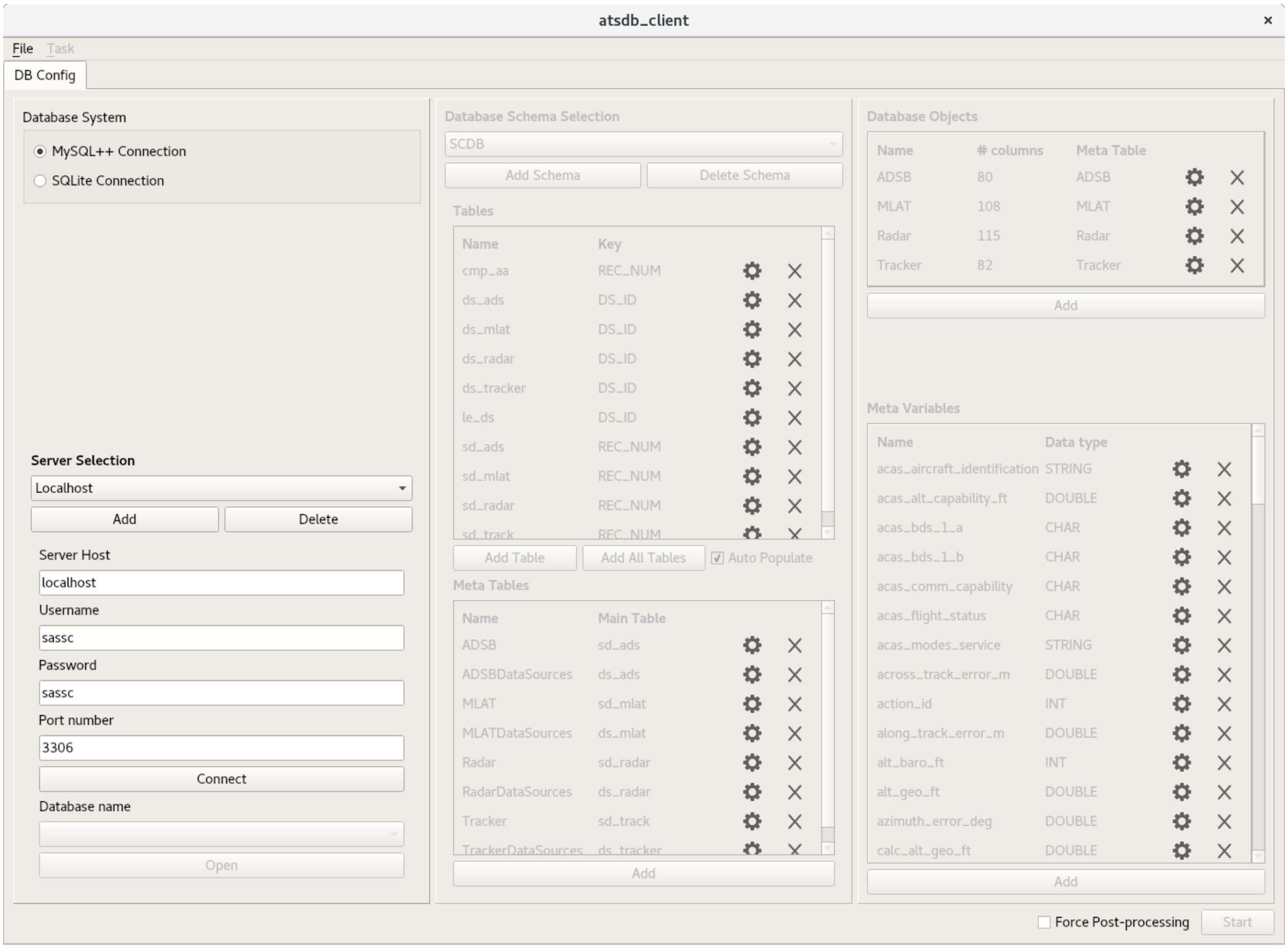
Usage

To start the ATSDB client, enter the ATSDB source/build folder, and execute:

```
./atsdb_client
```

Opening a database

While ATSDB is more a database framework, it comes with a dedicated client. When the ATSDB client is started a dialog for opening a database is shown.



Database System

● MySQL++ Connection

○ SQLite Connection

Server Selection: Localhost

Server Host: localhost

Username: sassc

Password: sassc

Port number: 3306

Database name:

Database Schema Selection: SCDB

Tables

Name	Key
cmp_aa	REC_NUM
ds_ads	DS_ID
ds_mlat	DS_ID
ds_radar	DS_ID
ds_tracker	DS_ID
le_ds	DS_ID
sd_ads	REC_NUM
sd_mlat	REC_NUM
sd_radar	REC_NUM
sd_track	REC_NUM

☑ Auto Populate

Meta Tables

Name	Main Table
ADSB	sd_ads
ADSBDataSources	ds_ads
MLAT	sd_mlat
MLATDataSources	ds_mlat
Radar	sd_radar
RadarDataSources	ds_radar
Tracker	sd_track
TrackerDataSources	ds_tracker

Database Objects

Name	# columns	Meta Table
ADSB	80	ADSB
MLAT	108	MLAT
Radar	115	Radar
Tracker	82	Tracker

Meta Variables

Name	Data type
acas_aircraft_identification	STRING
acas_alt_capability_ft	DOUBLE
acas_bds_1_a	CHAR
acas_bds_1_b	CHAR
acas_comm_capability	CHAR
acas_flight_status	CHAR
acas_modes_service	STRING
across_track_error_m	DOUBLE
action_id	INT
along_track_error_m	DOUBLE
alt_baro_ft	INT
alt_geo_ft	DOUBLE
azimuth_error_deg	DOUBLE
calc_alt_geo_ft	DOUBLE

□ Force Post-processing

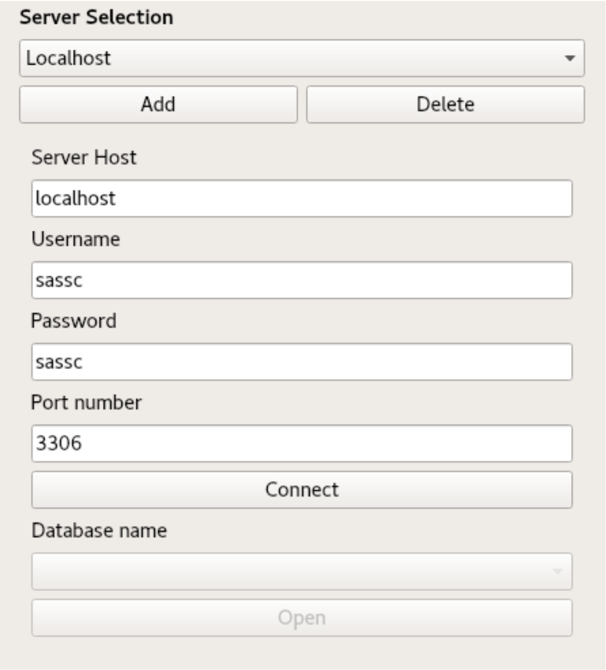
Figure 1: Connecting to a database

On the left-hand side, a database system can be selected. Choices are either MySQL database or a file container with a SQLite3 database.
On the lower left (depending on the database system) either a MySQL server connection can be configured or a list of SQLite3 files is shown.

On the right-hand side a database schema can be selected and edited (editing is only recommended for experienced users).

Database Selection

MySQL database



Server Selection

Localhost

Server Host

localhost

Username

sassc

Password

sassc

Port number

3306

Database name

Figure 2: Connecting to a MySQL server

Several MySQL servers can be defined, each one has a specific set of parameters. To add a new server, press the ’Add’ button and enter a unique server name. To select the currently used server, use the dropdown menu. To delete the currently used server, press the ’Delete’ button.

For connecting to a MySQL database, several parameters have to be entered:

Parameter	**Description**	**Example Values**
Server Host	Network identifier of server	’localhost’, ’10.0.0.123’
Username	MySQL user name	’sassc’, ’root’
Password	MySQL user password	’sassc’, ”
Port Number	MySQL server port	’3306’, ”

Table 2: MySQL server parameters

To connect to a defined MySQL server, press the ’Connect’ button.

If a wrong database name or IP address is used, error messages can be e.g.

```
MySQLConnection: executeSQL: error when executing
```

or

```
MySQLConnection:  init:  DB connection failed
```

If such an error occurs, correcting the server host name and or user/password should solve the problem.

After successful connection, all existing databases in the server are shown in the 'Database name' drop-down menu. The last used one is selected automatically.

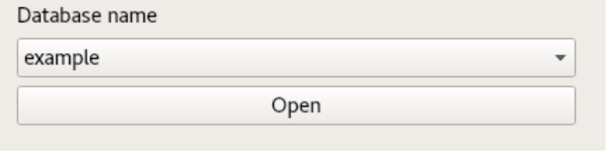
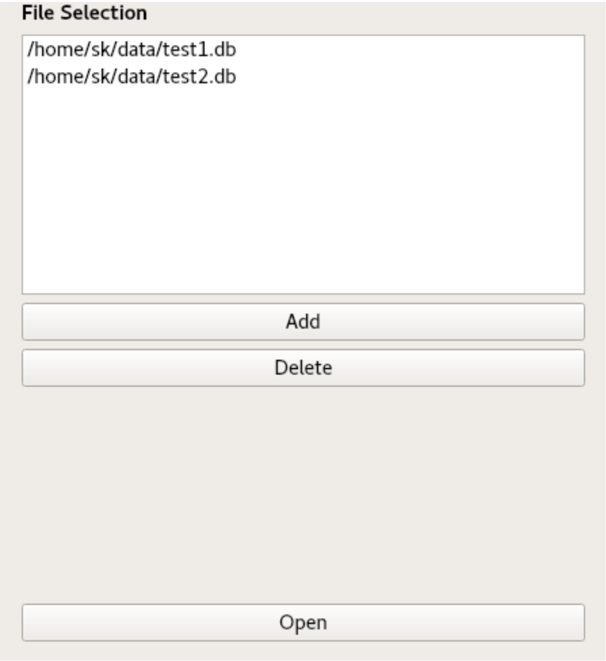

Database name: example

Figure 3: Selecting a MySQL database.

To open a database click the 'Open' button (lower left corner).

SQlite3 File container

For opening a file container, clicking the 'Select' button opens a file selection dialog, in which any SQLite3 file can be selected as data source.

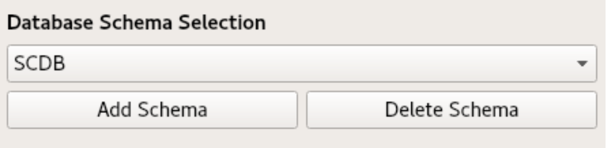

File Selection

/home/sk/data/test1.db
/home/sk/data/test2.db

Figure 4: Opening a SQLite3 file container

Database Schema Selection

For a common user, selection of a pre-configured database schema is recommened. To select a different database schema, please use the ’Schema selection’ drop-down menu.

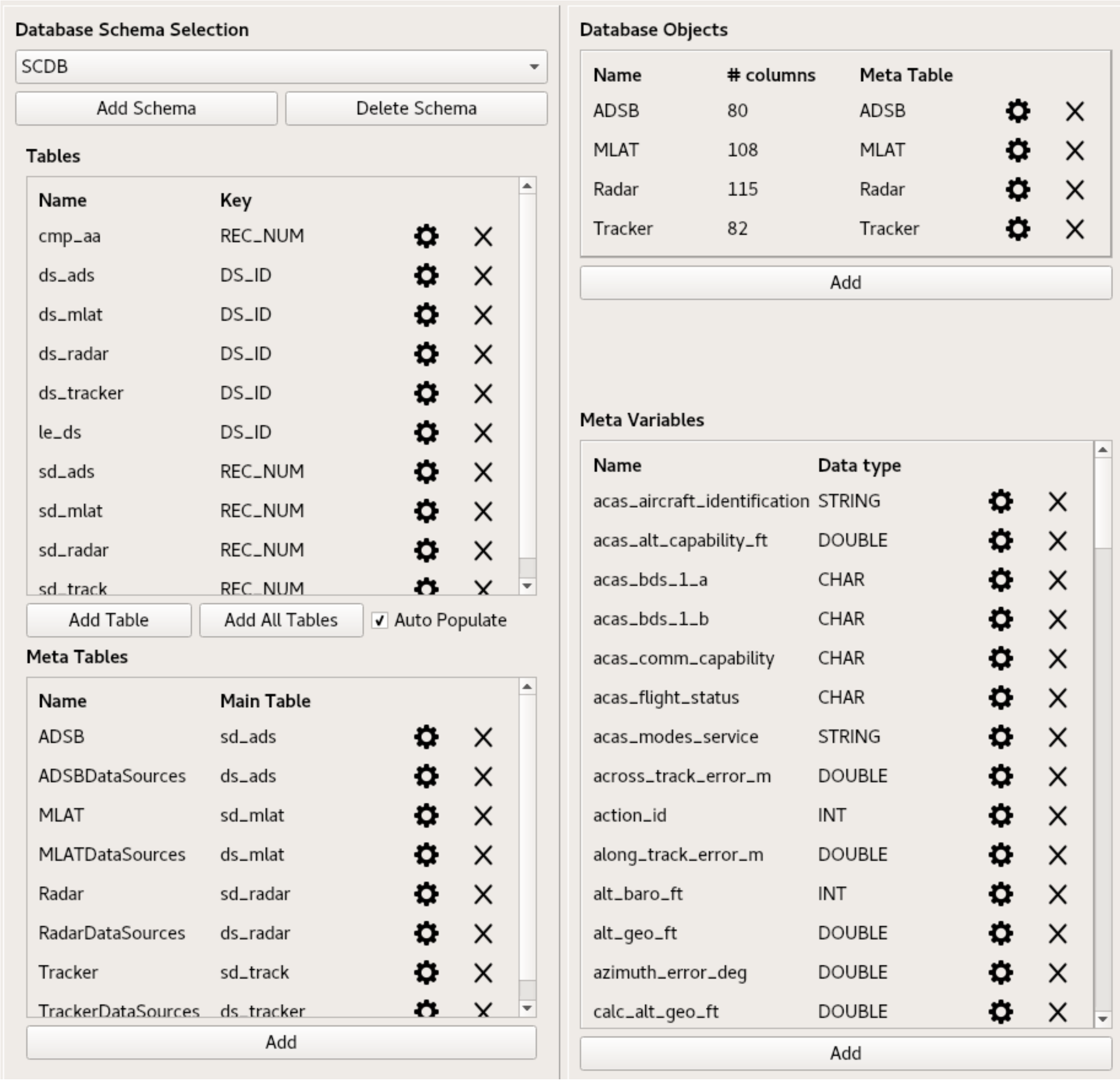


Database Schema Selection

SCDB

Figure 5: Selecting a database schema

For experienced users, on the right-hand side a database schema can be selected and edited, which is currently not recommended (since it is not user friendly and might crash if used in the “wrong” manner).



Database Schema Selection

SCDB

Tables

Name	Key		
cmp_aa	REC_NUM	⚙	✕
ds_ads	DS_ID	⚙	✕
ds_mlat	DS_ID	⚙	✕
ds_radar	DS_ID	⚙	✕
ds_tracker	DS_ID	⚙	✕
le_ds	DS_ID	⚙	✕
sd_ads	REC_NUM	⚙	✕
sd_mlat	REC_NUM	⚙	✕
sd_radar	REC_NUM	⚙	✕
sd_track	REC_NUM	⚙	✕

☑ Auto Populate

Meta Tables

Name	Main Table		
ADSB	sd_ads	⚙	✕
ADSBDataSources	ds_ads	⚙	✕
MLAT	sd_mlat	⚙	✕
MLATDataSources	ds_mlat	⚙	✕
Radar	sd_radar	⚙	✕
RadarDataSources	ds_radar	⚙	✕
Tracker	sd_track	⚙	✕
TrackerDataSources	ds_tracker	⚙	✕

Database Objects

Name	# columns	Meta Table		
ADSB	80	ADSB	⚙	✕
MLAT	108	MLAT	⚙	✕
Radar	115	Radar	⚙	✕
Tracker	82	Tracker	⚙	✕

Meta Variables

Name	Data type		
acas_aircraft_identification	STRING	⚙	✕
acas_alt_capability_ft	DOUBLE	⚙	✕
acas_bds_1_a	CHAR	⚙	✕
acas_bds_1_b	CHAR	⚙	✕
acas_comm_capability	CHAR	⚙	✕
acas_flight_status	CHAR	⚙	✕
acas_modes_service	STRING	⚙	✕
across_track_error_m	DOUBLE	⚙	✕
action_id	INT	⚙	✕
along_track_error_m	DOUBLE	⚙	✕
alt_baro_ft	INT	⚙	✕
alt_geo_ft	DOUBLE	⚙	✕
azimuth_error_deg	DOUBLE	⚙	✕
calc_alt_geo_ft	DOUBLE	⚙	✕

Figure 6: Configuring a database schema

Starting

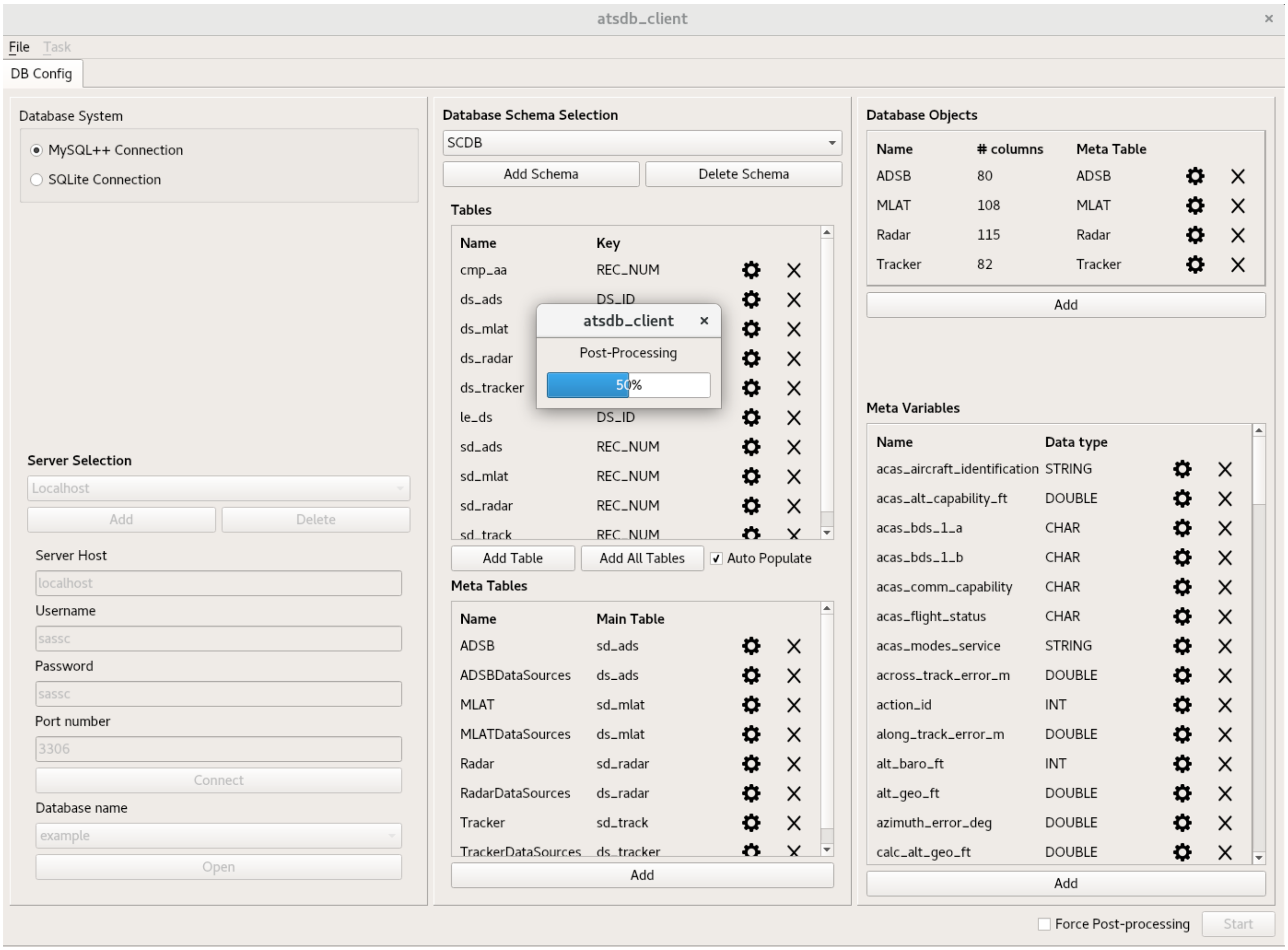
After the previous steps have been completed, the 'Start' button can be pressed to continue.

When a database is opened the first time, a post-processing has to be performed.

Postprocessing

When a database is generated by a previous process, some information that eases usage of the software does not exist. This information is generated once during a post-processing step, which is automatically performed. If wanted, it can always performed using the 'Force post-processing' checkbox.

Please **note** that during post-processing the application will not react to user input.



Database System

● MySQL++ Connection

○ SQLite Connection

Server Selection: Localhost

Server Host: localhost

Username: sassc

Password: sassc

Port number: 3306

Database name: example

Database Schema Selection: SCDB

Tables

Name	Key
cmp_aa	REC_NUM
ds_ads	DS_ID
ds_mlat	
ds_radar	
ds_tracker	
le_ds	DS_ID
sd_ads	REC_NUM
sd_mlat	REC_NUM
sd_radar	REC_NUM
sd_track	REC_NUM

☑ Auto Populate

atsdb_client ×

Post-Processing

50%

Meta Tables

Name	Main Table
ADSB	sd_ads
ADSBDataSources	ds_ads
MLAT	sd_mlat
MLATDataSources	ds_mlat
Radar	sd_radar
RadarDataSources	ds_radar
Tracker	sd_track
TrackerDataSources	ds_tracker

Database Objects

Name	# columns	Meta Table
ADSB	80	ADSB
MLAT	108	MLAT
Radar	115	Radar
Tracker	82	Tracker

Meta Variables

Name	Data type
acas_aircraft_identification	STRING
acas_alt_capability_ft	DOUBLE
acas_bds_1_a	CHAR
acas_bds_1_b	CHAR
acas_comm_capability	CHAR
acas_flight_status	CHAR
acas_modes_service	STRING
across_track_error_m	DOUBLE
action_id	INT
along_track_error_m	DOUBLE
alt_baro_ft	INT
alt_geo_ft	DOUBLE
azimuth_error_deg	DOUBLE
calc_alt_geo_ft	DOUBLE

□ Force Post-processing

Figure 7: Post-processing a database

The following information is generated and stored in the database:

- List of all active data sources for all DBOs

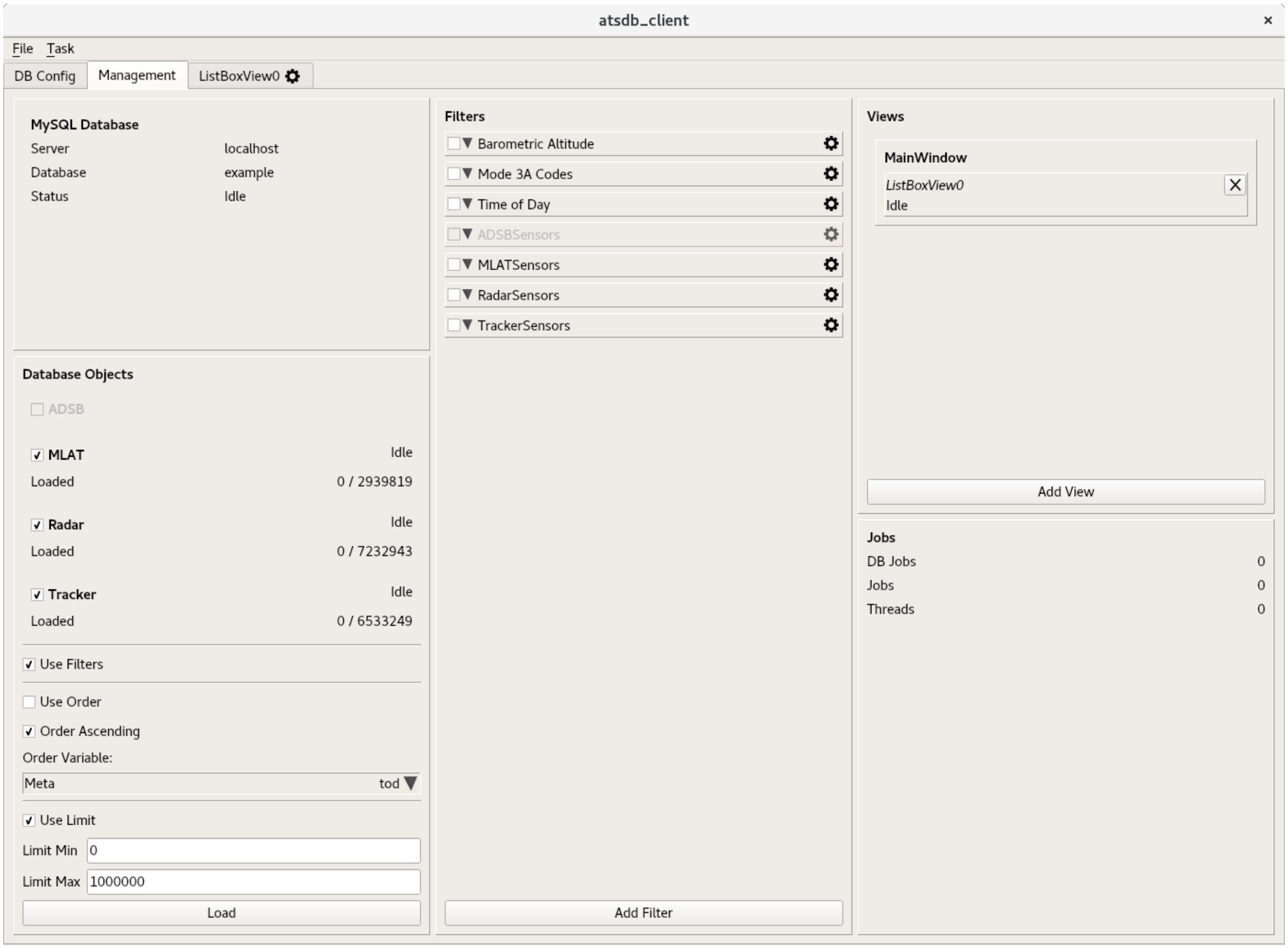
- List with all minima/maxima for all variables of all DBOs

Please **note** that this step has to be performed only once for each database, and may take up to a few minutes for large datasets.

Please also **note** that during this step, no DBO data itself is changed, but only additional information is generated and stored in separate database tables.

Management

After pressing the 'Start' button, a management window is shown.



MySQL Database

Server	localhost
Database	example
Status	Idle

Database Objects

□ ADSB

☑ MLAT Idle
Loaded 0 / 2939819

☑ Radar Idle
Loaded 0 / 7232943

☑ Tracker Idle
Loaded 0 / 6533249

☑ Use Filters

□ Use Order

☑ Order Ascending

Order Variable: Meta tod

☑ Use Limit

Limit Min 0

Limit Max 1000000

Filters

- □ Barometric Altitude
- □ Mode 3A Codes
- □ Time of Day
- □ ADSBSensors
- □ MLATSensors
- □ RadarSensors
- □ TrackerSensors

MainWindow

ListBoxView0
Idle

Jobs

DB Jobs	0
Jobs	0
Threads	0

Figure 8: Main management window

In the uppermost part, the current tab can be selected. The following tabs exist:

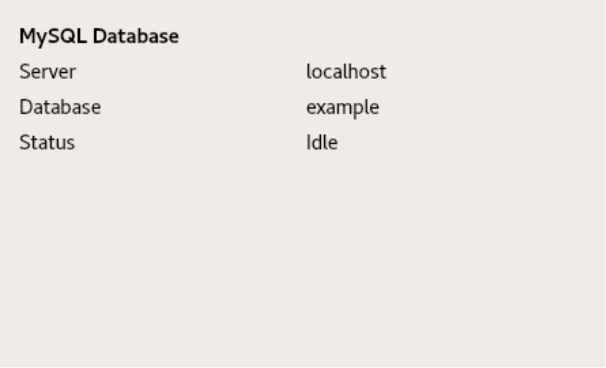
Tab	Description
DB Config	Database configuration, used during startup
Management	All elements to manage loading and inspection of data
...	Additional tabs with Views

Table 3: Main window tab list

Located in the main window, a management tab exists. It shows general database information, a list of database objects with loading functions, a filter system and so on.

Database Information

In this widget, general information about the database is given.



MySQL Database

Server	localhost
Database	example
Status	Idle

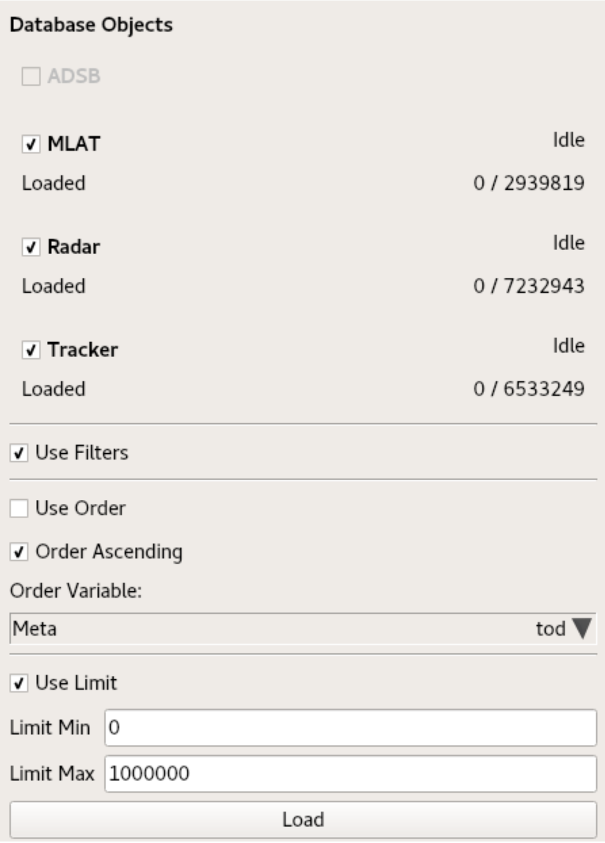
Figure 9: Management: Database Information

This also includes the working status, which can be:

- Idle: Nothing to do at the moment
- Working: Database read/write in progress

Database Objects

In this widget, information about the DBO's, the loaded dataset, and the loading parameters are given.



Database Objects

□ ADSB

☑ MLAT Idle

Loaded 0 / 2939819

☑ Radar Idle

Loaded 0 / 7232943

☑ Tracker Idle

Loaded 0 / 6533249

☑ Use Filters

□ Use Order

☑ Order Ascending

Order Variable: Meta tod

☑ Use Limit

Limit Min 0

Limit Max 1000000

Figure 10: Management: Database Objects

Each existing DBO is listed, and active ones have the following items:

- Name checkbox: Defines whether data from this DBO should be loaded
- Loading status information
  - Idle: Nothing to do at the moment
  - Loading: DBO read/write in progress
- Loaded data size: Number of loaded items / Number of existing items

If a DBO exists, but has no data in the database, it is shown as inactive (greyed out, like ADSB in the screenshot).

Additionally, parameters which configure the loading process exist:

- “Use Filters” checkbox: Whether filtering should be performed
- “Use order” checkbox: Whether the dataset should be ordered by a DBO variable
- “Use Ascending“ checkbox: If ordered, defines if it should be ascending or descending

- ”Order Variable“ selection: If ordered, what variable should be used
- ”Use Limit“ checkbox: If the data size should be limited
- ”Limit Min“: If limited, the data set will start at the n-th index given here. If 0, it will be loaded from the beginning, if e.g. 100, the first 100 entries will be skipped and the 101st entry will be the first in the dataset.
- ”Limit Max”: If limited, the number given here will be the number of loaded entries.

Using the “Load” button , a loading process using the current configuration is started.

Filters

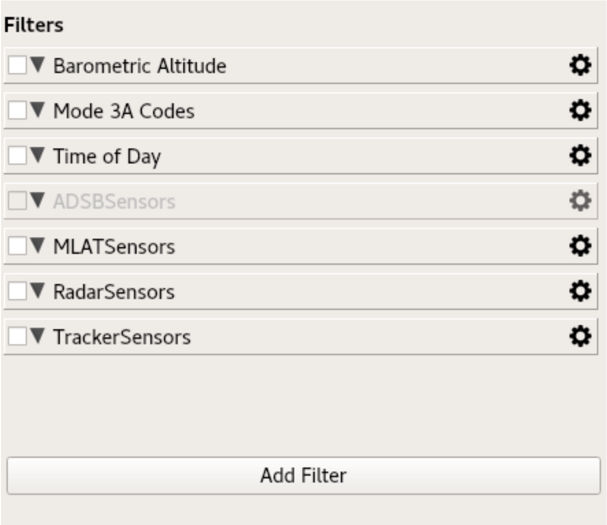


Figure 11: Management: Filters

Each filter consists of a checkbox, defining if a filter is active (contributes to the search query), a triangle-button (to show/hide the filter configuration elements), a unique name, and a manage button (activates a context menu). At the bottom an ’Add filter’ button exists, which can be used to add new filters.

Please **note** that the filter configuration will be saved at program shutdown, which is also true for new filters. At startup, all filters from the configuration are generated and restored to their previous state. However, when the database was changed (usage of different data source), all filters are reset to an initial state (since their previous configuration may be senseless).

Please also **note** that active filters, at the moment, are always combined with a logical AND. Therefore, when two filters are active, only the intersection of data which both filters allow is loaded. A logical combination of filters using an OR operation is planned, but was not implemented yet.

Sensor Filters

For each DBO with a sensor list, a sensor filter is generated which can not be edited or deleted. For each sensor a checkbox exists, which is only active if the sensor was active in the database. If checked, the data generated by the sensor is loaded, and vice versa.

Custom filters

A custom filter does not differ in general usage, but the inner workings are different. Also, it can be generated using the 'Add filter' button. It can also be reset (to the original values) edited and deleted using the manage button.
A custom filter consists of one or more filter conditions. Such a condition involves a DBO (or meta) variable, an operator, and a value. When active, the filter restricts all loaded DBO data to be fullfill all filter conditions. As an example, the "Time of Day" filter limits the loaded data to a specific time window, to load only time slices of the dataset. The "Mode 3A Codes" filter restricts to a list of (comma-separated) mode 3/A codes, to single out specific flights.

For more information about filtering, please refer to Section Filtering.

Views

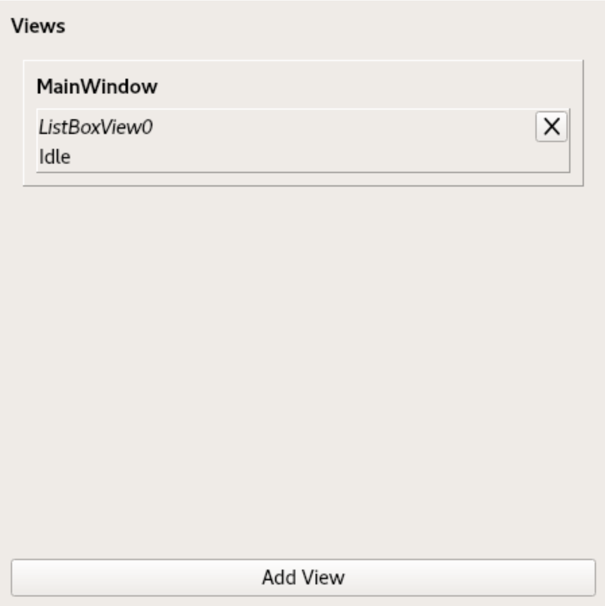


Figure 12: Management: Views

This element allows generation and management of all active Views and windows. Each View is contained in a tab within a parent window. At startup, only the main window exists ('MainWindow'), which also holds the management tab. If the main window is closed, the ATSDB client shuts down. New Views can be added using the 'Add View' button, which opens a pull-down menu. Each View can either be added to the main window ('MainWindow') or into a new window. When added, a new tab exists in the containing window, and controlling elements are added for any new windows or views.

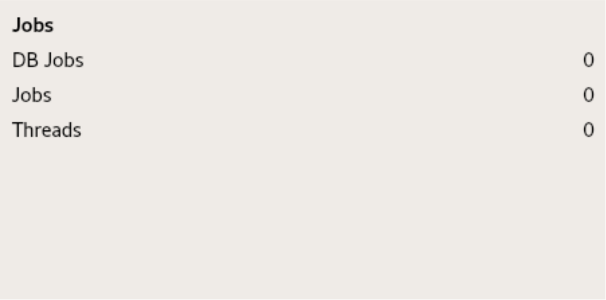
Currently, only the Listbox View exist, for more information about this View, please refer to Section Listbox View.

New Views can be added either to currently existing windows as new tabs, or to a newly opened window. A window can be closed either by the close button in the window decoration, which discards all contained Views within the window. To delete a single View, one can use the close button in the GUI, which frees up all its allocated resources. Each View adds its required variables to the loading list for the database. During a loading process, the loading status of a View is shown in the management tab.

For more information about Views, please refer to Section Inspection.

Jobs

The ATSDB framework supports multi-threading. A number of processing steps ("Jobs") can be exectuted on a number of parallel threads. Since multi-threading on a database creates limited benefit, only one thread is used specifically for database jobs. All non-database jobs are exectuted on a dynamic number of other threads, which are increased/decreased in number depending on the application's needs.



Jobs

DB Jobs	0
Jobs	0
Threads	0

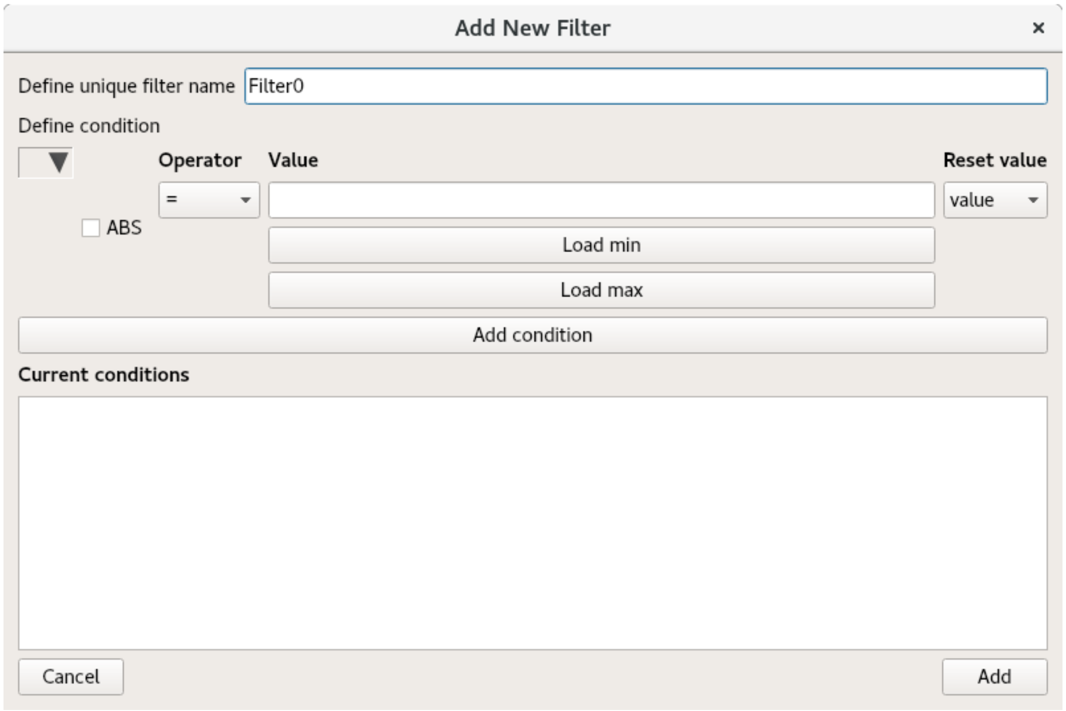
Figure 13: Management: Jobs

In the Job element the number of database jobs is listed under "DB Jobs", the number of other Jobs is listed under "Jobs". The number of active processing threads is listed under "Threads".

FILTERING

Adding a New Filter

When clicking the 'Add filter' button, a dialog is opened.



Define unique filter name: Filter0

Operator: =

Reset value: value

□ ABS

Figure 14: Adding a filter

First, one has to give the filter a new (unique) name. Then, conditions have to be defined and added. A condition consists of a DBO variable, an operator, a value, and a reset value.

When the triangular button is clicked, a sub-menu is opened, where one can choose a DBO variable. The selected variable restricts data of all DBOs if it is of type ’Meta’, or just data from one DBO if it is not.Additionally, the mathematical operator ’ABS’ can be selected. If so, not the value of the variable but the absolute value of the variable is used: ’ABS(var)>value’ is equivalent to ’var>value OR var<-value’.

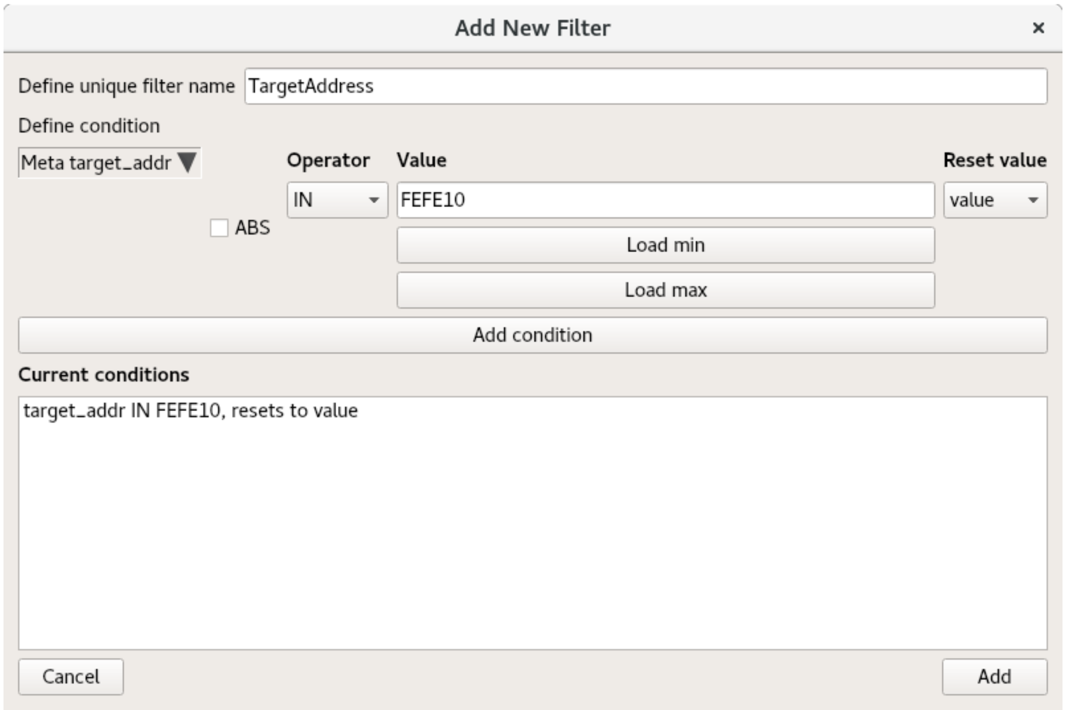
An operator can be chosen with the drop-down menu, the supplied operators are common SQL operators.

Operator	Description
=	Equal
!=	Not equal
>	Greater than
>=	Greater than or equal
<	Less than
<=	Less than or equal
IN	Matches a value in a list
LIKE	Pattern matching with % and _
IS	Value NULL: No value exists
IS NOT	Value NULL: Value exists

Table 4: SQL operators

In the 'Value' field one can set a value manually, or load the minimum or maximum values of the selected DBO variable from the database using the 'Load min'/'Load max' buttons . A reset value also has to be supplied, which can be the chosen value or a minimum/maximum value set from the database. Whenever a database different from the previous one is opened, all filters are reset, since previous values may have become invalid.

After a condition is defined, it has to be added using the 'Add condition' button. Existing conditions are shown in the 'Current conditions' list. Please note that added conditions can not be removed in this dialog, but have to be removed as described in the Section Editing a Filter.



Define unique filter name TargetAddress

Define condition

Meta target_addr

□ ABS

Operator IN

Value FEFE10

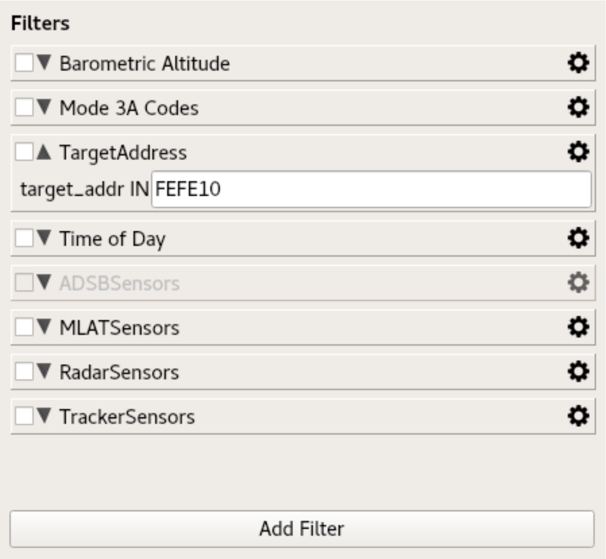
Reset value value

Current conditions

target_addr IN FEFE10, resets to value

Figure 15: Filled out filter dialog

Now the described process can be repeated until a usable filter emerges, which is added using the 'Add' button. The process of adding a new filter can be canceled by using the 'Cancel' button, which discards all settings. When added, a new filter shows up immediately in the filter list and is saved to the configuration for persistence.



Filters

□ ▼ Barometric Altitude

□ ▼ Mode 3A Codes

□ ▲ TargetAddress

target_addr IN FEFE10

□ ▼ Time of Day

□ ▼ ADSBSensors

□ ▼ MLATSensors

□ ▼ RadarSensors

□ ▼ TrackerSensors

Add Filter

Figure 16: Filter added

Editing a Filter

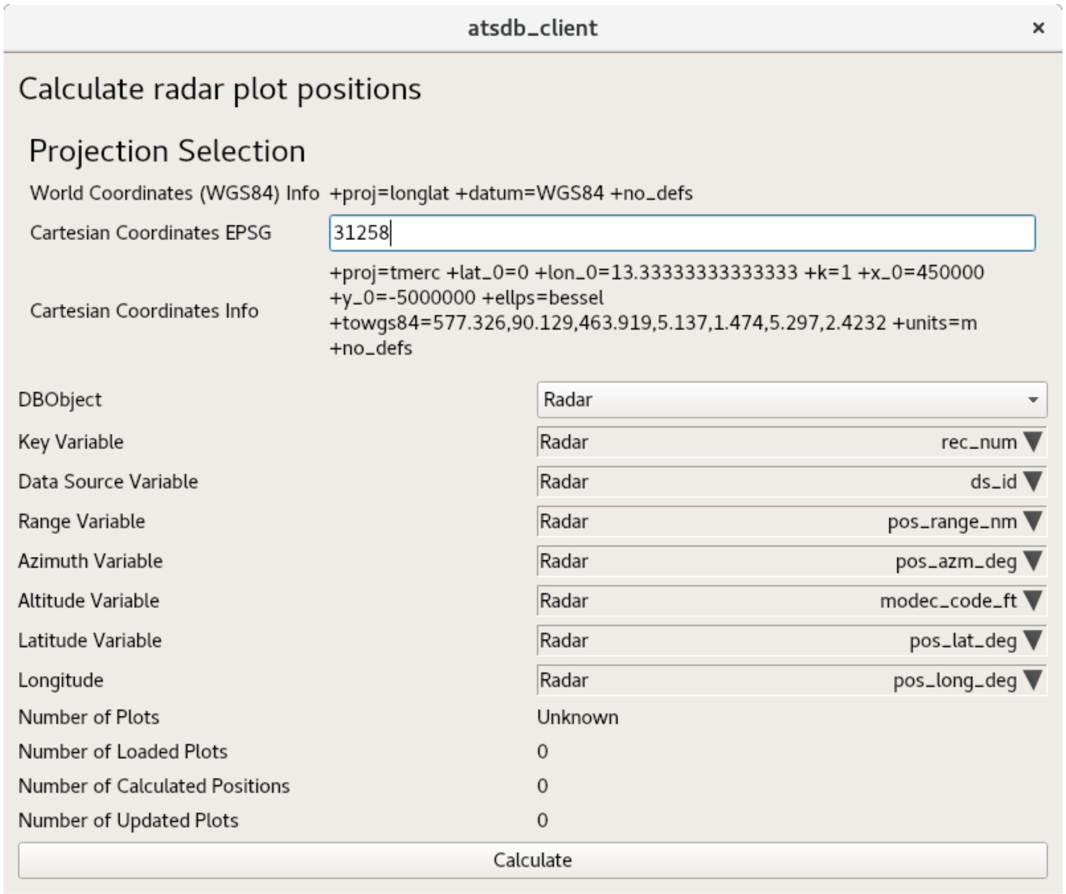
Currently, filter editing has been disabled and will be added at a later version.

TASKS

Since in SASS-C Verif radar plot coordinates are not given as latitude/longitude, which are the main coordinates for all processing in ATSDB, optionally these coordinates can be re-calculated and set in the database using the ”Calculate Radar Plot Position“ Task.

Calculate Radar Plot Position

To execute this task select ”Task“-> ”Calculate Radar Plot Position‘ in the top menu bar.



Calculate radar plot positions

Projection Selection

World Coordinates (WGS84) Info +proj=longlat +datum=WGS84 +no_defs

Cartesian Coordinates EPSG 31258

Cartesian Coordinates Info +proj=tmerc +lat_0=0 +lon_0=13.33333333333333 +k=1 +x_0=450000 +y_0=-5000000 +ellps=bessel +towgs84=577.326,90.129,463.919,5.137,1.474,5.297,2.4232 +units=m +no_defs

DBObject	Radar	
Key Variable	Radar	rec_num
Data Source Variable	Radar	ds_id
Range Variable	Radar	pos_range_nm
Azimuth Variable	Radar	pos_azm_deg
Altitude Variable	Radar	modec_code_ft
Latitude Variable	Radar	pos_lat_deg
Longitude	Radar	pos_long_deg
Number of Plots	Unknown	
Number of Loaded Plots	0	
Number of Calculated Positions	0	
Number of Updated Plots	0	

Figure 17: Calculate radar plot position task

The EPSG code for the projection has to be chosen according to your needs, please refer to `http://spatialreference.org/ref/epsg/` for a list of possible codes.

The WGS84 latitude/longitude coordinates are then calculated using the radar positions in the database, the range and the azimuth. Press “Calculate” to start the calculation process, which will take a few minutes depending on the data size.

Please **note** that currently the various “Number of” labels are not set correctly, and no status indication exists. This will be fixed in a later version.

Messages like these will be printed in the text console, the last one indicates completion of the task:

```
...
[INFO] RadarPlotPositionCalculatorTask: loadingDoneSlot: starting calculation
[INFO] RadarPlotPositionCalculatorTask: loadingDoneSlot: writing update_buffer
```

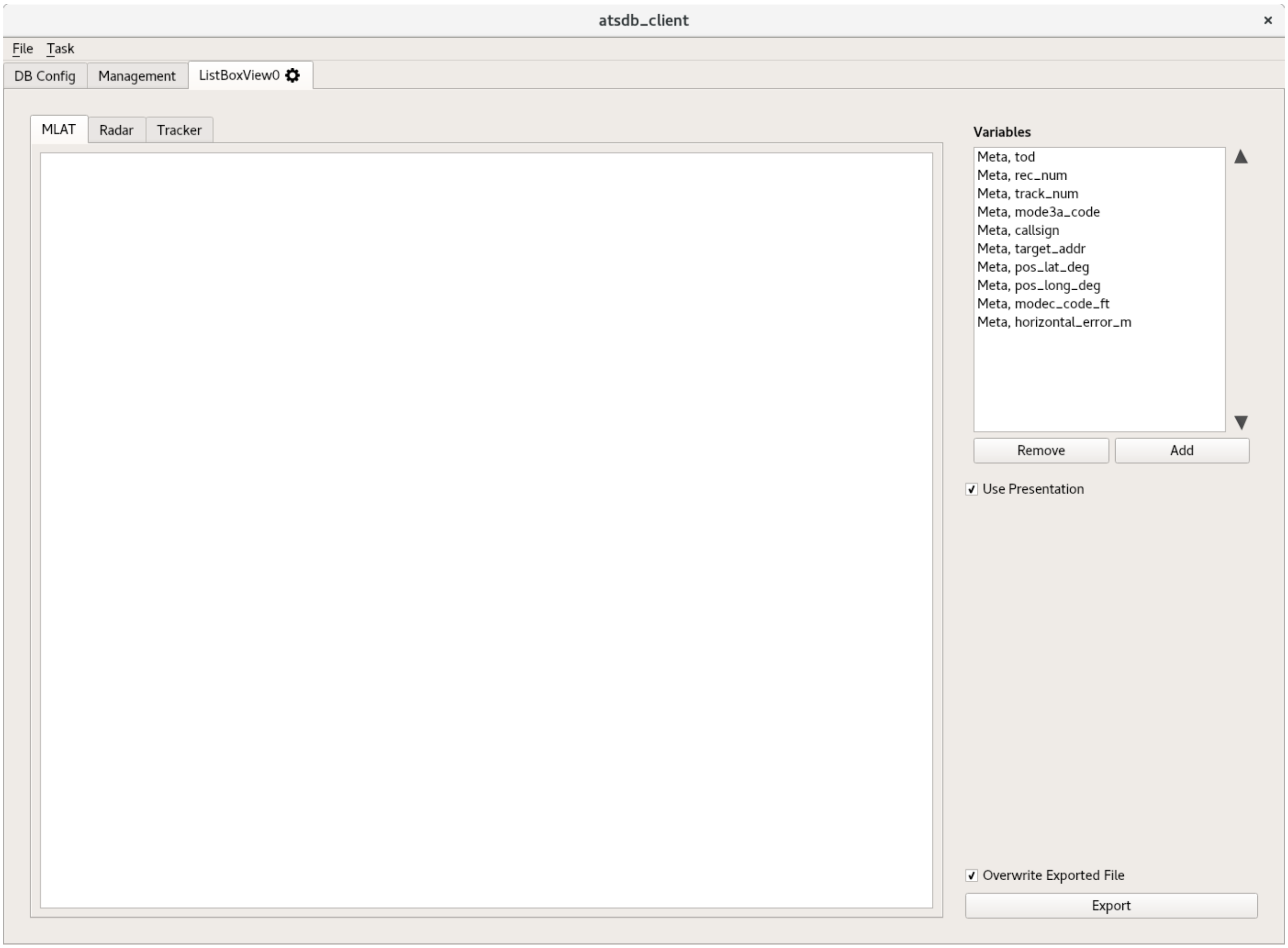
```
[INFO] RadarPlotPositionCalculatorTask: loadingDoneSlot: update_buffer size 7230527
[INFO] RadarPlotPositionCalculatorTask: loadingDoneSlot: end
[INFO] UpdateBufferDBJob: run: start
[INFO] UpdateBufferDBJob: run: writing object Radar key rec_num size 7230527
[INFO] BufferWriter: update: buffer size 7230527 into table sd_radar
[INFO] SQLGenerator: createDBUpdateStringBind: idvar name REC_NUM
[INFO] BufferWriter: update: preparing bind statement
[INFO] BufferWriter: update: starting inserts
[INFO] BufferWriter: update: bind transactions cnt 0
[INFO] BufferWriter: update: bind transactions cnt 100000
...
[INFO] BufferWriter: update: bind transactions cnt 7200000
[INFO] BufferWriter: update: ending bind transactions
UpdateBufferDBJob: run: buffer write done (411.33 s).
```

After running this task once (per database), the radar plots also have a set latitude/longitude. This task can be re-run with different projections if wanted.

Inspection

Listbox View

A Listbox View displays the DBO records as text in tables, to allow full data inspection. When started, it presents itself in the following manner.



Variables

Meta, tod
Meta, rec_num
Meta, track_num
Meta, mode3a_code
Meta, callsign
Meta, target_addr
Meta, pos_lat_deg
Meta, pos_long_deg
Meta, modec_code_ft
Meta, horizontal_error_m

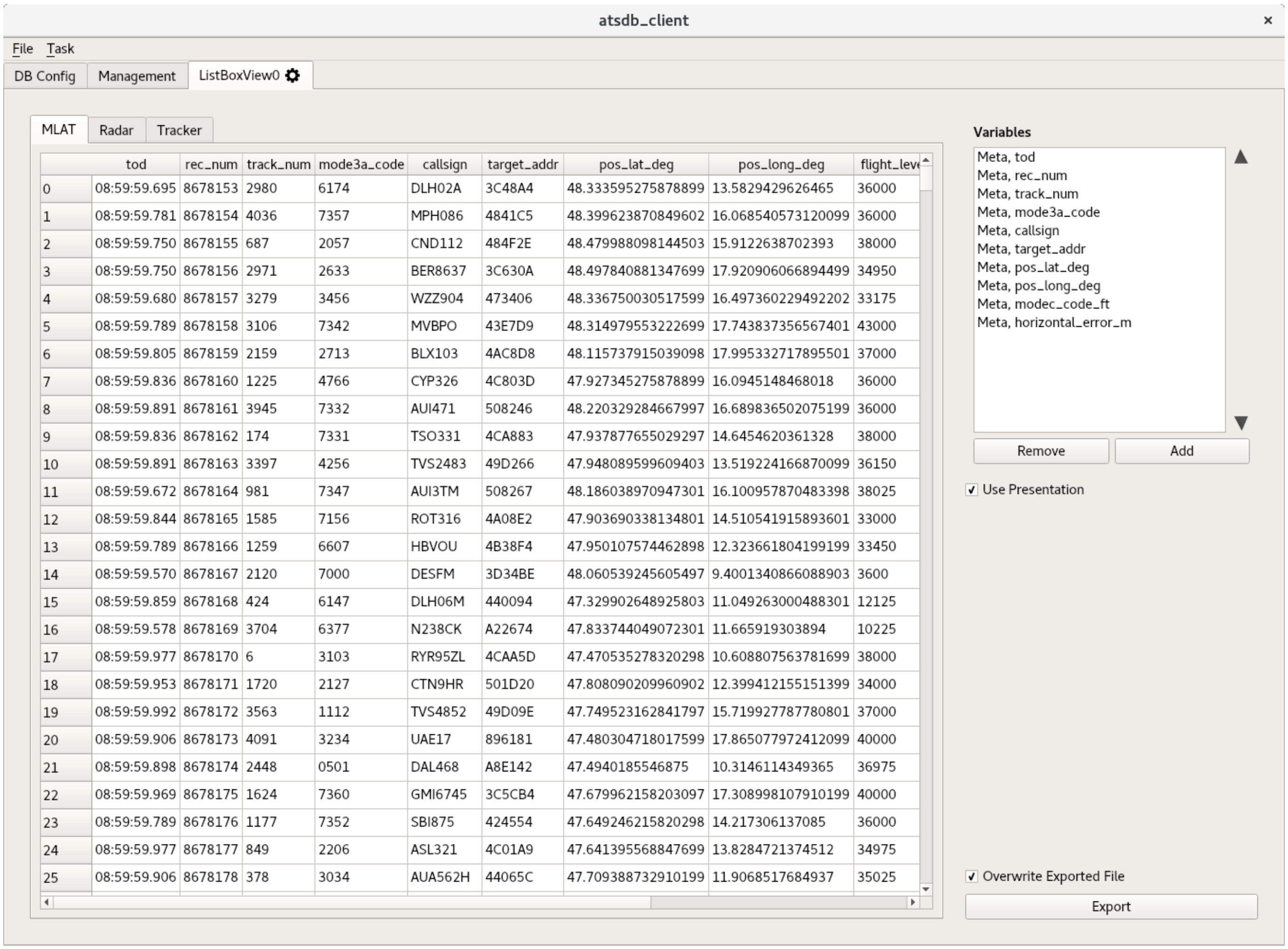
☑ Use Presentation

☑ Overwrite Exported File

Figure 18: Listbox View startup

On the left side, a number of tabs exist for each active DBO, each of which contains a table. On the right side, a configuration area exists. A number of variables is displayed in the 'Variables' list. One can change the order, remove and add variables to be inspected.

To limit, order based on a variable or load the dataset, the mechanism described in Section Database Objects can be used. To filter the dataset, the mechanism described in Section Filtering can be used.



	tod	rec_num	track_num	mode3a_code	callsign	target_addr	pos_lat_deg	pos_long_deg	flight_leve
0	08:59:59.695	8678153	2980	6174	DLH02A	3C48A4	48.333595275878899	13.5829429626465	36000
1	08:59:59.781	8678154	4036	7357	MPH086	4841C5	48.399623870849602	16.068540573120099	36000
2	08:59:59.750	8678155	687	2057	CND112	484F2E	48.479988098144503	15.9122638702393	38000
3	08:59:59.750	8678156	2971	2633	BER8637	3C630A	48.497840881347699	17.920906066894499	34950
4	08:59:59.680	8678157	3279	3456	WZZ904	473406	48.336750030517599	16.497360229492202	33175
5	08:59:59.789	8678158	3106	7342	MVBPO	43E7D9	48.314979553222699	17.743837356567401	43000
6	08:59:59.805	8678159	2159	2713	BLX103	4AC8D8	48.115737915039098	17.995332717895501	37000
7	08:59:59.836	8678160	1225	4766	CYP326	4C803D	47.927345275878899	16.0945148468018	36000
8	08:59:59.891	8678161	3945	7332	AUI471	508246	48.220329284667997	16.689836502075199	36000
9	08:59:59.836	8678162	174	7331	TSO331	4CA883	47.937877655029297	14.6454620361328	38000
10	08:59:59.891	8678163	3397	4256	TVS2483	49D266	47.948089599609403	13.519224166870099	36150
11	08:59:59.672	8678164	981	7347	AUI3TM	508267	48.186038970947301	16.100957870483398	38025
12	08:59:59.844	8678165	1585	7156	ROT316	4A08E2	47.903690338134801	14.510541915893601	33000
13	08:59:59.789	8678166	1259	6607	HBVOU	4B38F4	47.950107574462898	12.323661804199199	33450
14	08:59:59.570	8678167	2120	7000	DESFM	3D34BE	48.060539245605497	9.4001340866088903	3600
15	08:59:59.859	8678168	424	6147	DLH06M	440094	47.329902648925803	11.049263000488301	12125
16	08:59:59.578	8678169	3704	6377	N238CK	A22674	47.833744049072301	11.665919303894	10225
17	08:59:59.977	8678170	6	3103	RYR95ZL	4CAA5D	47.470535278320298	10.608807563781699	38000
18	08:59:59.953	8678171	1720	2127	CTN9HR	501D20	47.808090209960902	12.399412155151399	34000
19	08:59:59.992	8678172	3563	1112	TVS4852	49D09E	47.749523162841797	15.719927787780801	37000
20	08:59:59.906	8678173	4091	3234	UAE17	896181	47.480304718017599	17.865077972412099	40000
21	08:59:59.898	8678174	2448	0501	DAL468	A8E142	47.4940185546875	10.3146114349365	36975
22	08:59:59.969	8678175	1624	7360	GMI6745	3C5CB4	47.679962158203097	17.308998107910199	40000
23	08:59:59.789	8678176	1177	7352	SBI875	424554	47.649246215820298	14.217306137085	36000
24	08:59:59.977	8678177	849	2206	ASL321	4C01A9	47.641395568847699	13.8284721374512	34975
25	08:59:59.906	8678178	378	3034	AUA562H	44065C	47.709388732910199	11.9068517684937	35025

Meta, tod
Meta, rec_num
Meta, track_num
Meta, mode3a_code
Meta, callsign
Meta, target_addr
Meta, pos_lat_deg
Meta, pos_long_deg
Meta, modec_code_ft
Meta, horizontal_error_m

☑ Use Presentation

☑ Overwrite Exported File

Figure 19: Listbox View after loading

Once updated, the tables are filled with text representing the values of the DBO variables. If a value is undefined its cell is empy.

Please **note** that since some variable might only exist in some DBOs, the number of columns for different DBOs may differ.

Variable List

All DBO variables which are loaded from the database are shown in the 'Variables' list. This list is ordered, and like all configuration elements persistent. Ordering can be changed by selecting (clicking on) a variable and using the up/down triangle buttons. When pressing the ′Remove′ button, a selected variable is removed. Pressing the ′Add′ button allows appending a variable to the list using a context-menu.

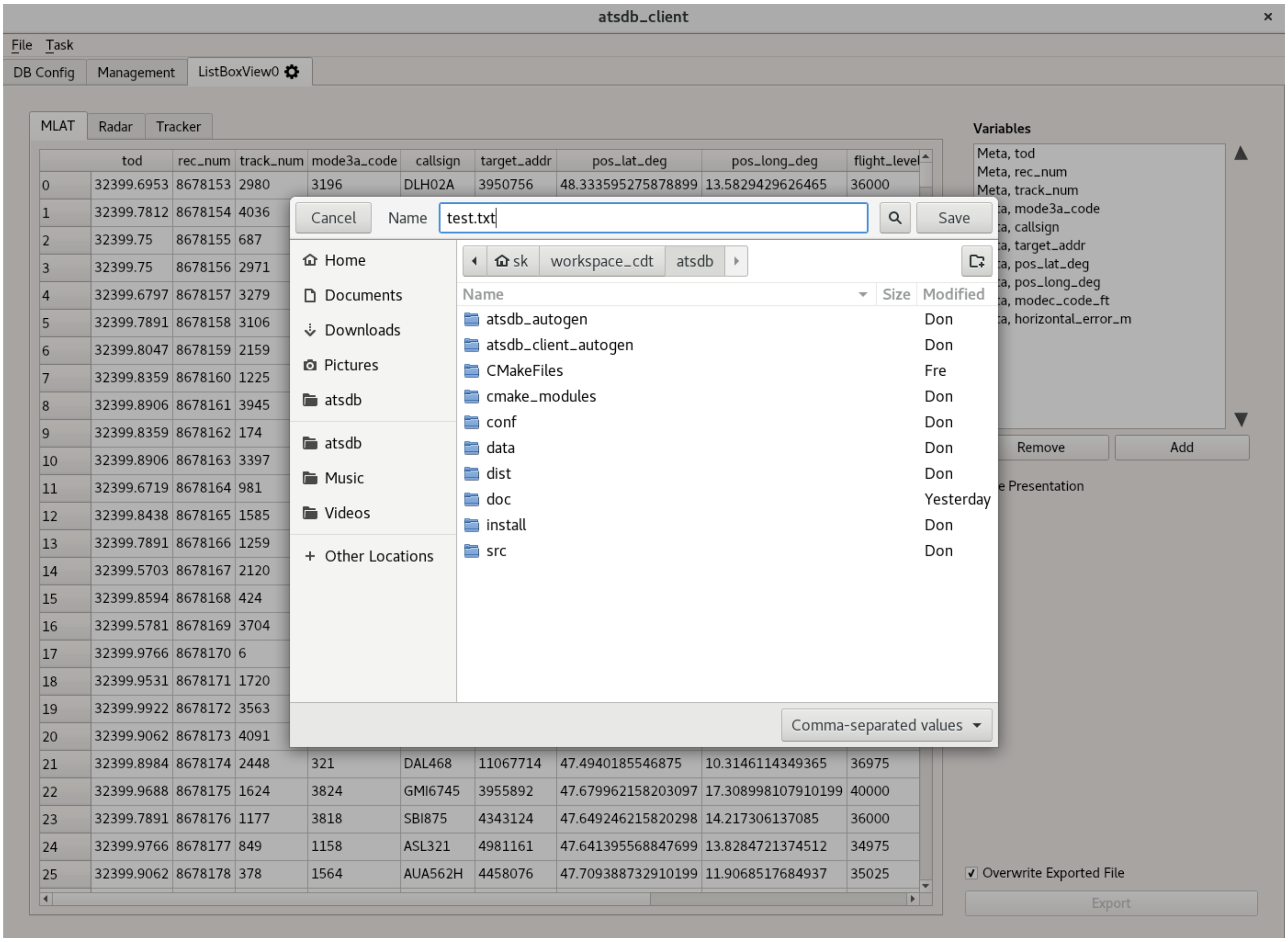
Use Presentation

When this checkbox is checked, the so called presentation mode is used. In the database, the variables might have different units or a data representation which is not easy to read. For this purpose, a presentation mode was introduced to e.g. show a Mode A code as octal, or a Time of Day not in seconds since midnight but in HH::MM:SS.SS format. When the ″Use Presentation“ checkbox is not checked, the original database values are presented (and exported).

Exporting

The data from the current DBO table can be exported to a comma-separated value (CSV) text file.

When pressing the ″Export“ button, a dialog is opened.



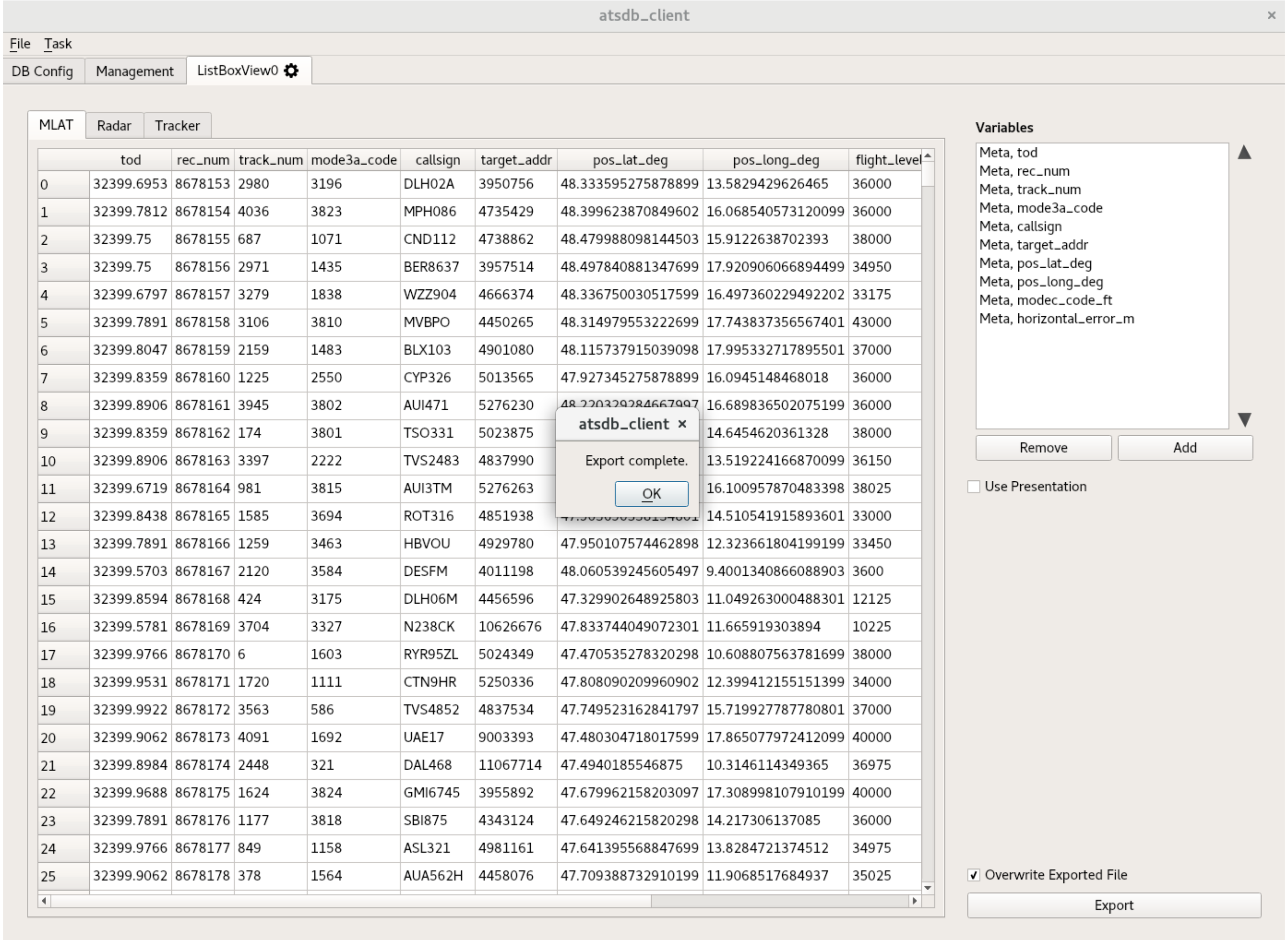
Name: test.txt

☑ Overwrite Exported File

	tod	rec_num	track_num	mode3a_code	callsign	target_addr	pos_lat_deg	pos_long_deg	flight_level
0	32399.6953	8678153	2980	3196	DLH02A	3950756	48.333595275878899	13.5829429626465	36000
1	32399.7812	8678154	4036						
2	32399.75	8678155	687						
3	32399.75	8678156	2971						
4	32399.6797	8678157	3279						
5	32399.7891	8678158	3106						
6	32399.8047	8678159	2159						
7	32399.8359	8678160	1225						
8	32399.8906	8678161	3945						
9	32399.8359	8678162	174						
10	32399.8906	8678163	3397						
11	32399.6719	8678164	981						
12	32399.8438	8678165	1585						
13	32399.7891	8678166	1259						
14	32399.5703	8678167	2120						
15	32399.8594	8678168	424						
16	32399.5781	8678169	3704						
17	32399.9766	8678170	6						
18	32399.9531	8678171	1720						
19	32399.9922	8678172	3563						
20	32399.9062	8678173	4091						
21	32399.8984	8678174	2448	321	DAL468	11067714	47.4940185546875	10.3146114349365	36975
22	32399.9688	8678175	1624	3824	GMI6745	3955892	47.679962158203097	17.308998107910199	40000
23	32399.7891	8678176	1177	3818	SBI875	4343124	47.649246215820298	14.217306137085	36000
24	32399.9766	8678177	849	1158	ASL321	4981161	47.641395568847699	13.8284721374512	34975
25	32399.9062	8678178	378	1564	AUA562H	4458076	47.709388732910199	11.9068517684937	35025

Figure 20: Listbox View export

Choose a filename, and press ”Save“ to save the data. If the ”Overwrite Exported File“ checkbox was checked, an existing file is automatically overwritten. Please **note** that exporting might take some time for larger datasets, and currently no status indication is given.
After export, a dialog is shown indicating that the export was completed.



	tod	rec_num	track_num	mode3a_code	callsign	target_addr	pos_lat_deg	pos_long_deg	flight_level
0	32399.6953	8678153	2980	3196	DLH02A	3950756	48.333595275878899	13.5829429626465	36000
1	32399.7812	8678154	4036	3823	MPH086	4735429	48.399623870849602	16.068540573120099	36000
2	32399.75	8678155	687	1071	CND112	4738862	48.479988098144503	15.9122638702393	38000
3	32399.75	8678156	2971	1435	BER8637	3957514	48.497840881347699	17.920906066894499	34950
4	32399.6797	8678157	3279	1838	WZZ904	4666374	48.336750030517599	16.497360229492202	33175
5	32399.7891	8678158	3106	3810	MVBPO	4450265	48.314979553222699	17.743837356567401	43000
6	32399.8047	8678159	2159	1483	BLX103	4901080	48.115737915039098	17.995332717895501	37000
7	32399.8359	8678160	1225	2550	CYP326	5013565	47.927345275878899	16.0945148468018	36000
8	32399.8906	8678161	3945	3802	AUI471	5276230	48.220329284667997	16.689836502075199	36000
9	32399.8359	8678162	174	3801	TSO331	5023875	[illegible]	14.6454620361328	38000
10	32399.8906	8678163	3397	2222	TVS2483	4837990	[illegible]	13.519224166870099	36150
11	32399.6719	8678164	981	3815	AUI3TM	5276263	[illegible]	16.100957870483398	38025
12	32399.8438	8678165	1585	3694	ROT316	4851938	[illegible]	14.510541915893601	33000
13	32399.7891	8678166	1259	3463	HBVOU	4929780	47.950107574462898	12.323661804199199	33450
14	32399.5703	8678167	2120	3584	DESFM	4011198	48.060539245605497	9.4001340866088903	3600
15	32399.8594	8678168	424	3175	DLH06M	4456596	47.329902648925803	11.049263000488301	12125
16	32399.5781	8678169	3704	3327	N238CK	10626676	47.833744049072301	11.665919303894	10225
17	32399.9766	8678170	6	1603	RYR95ZL	5024349	47.470535278320298	10.608807563781699	38000
18	32399.9531	8678171	1720	1111	CTN9HR	5250336	47.808090209960902	12.399412155151399	34000
19	32399.9922	8678172	3563	586	TVS4852	4837534	47.749523162841797	15.719927787780801	37000
20	32399.9062	8678173	4091	1692	UAE17	9003393	47.480304718017599	17.865077972412099	40000
21	32399.8984	8678174	2448	321	DAL468	11067714	47.4940185546875	10.3146114349365	36975
22	32399.9688	8678175	1624	3824	GMI6745	3955892	47.679962158203097	17.308998107910199	40000
23	32399.7891	8678176	1177	3818	SBI875	4343124	47.649246215820298	14.217306137085	36000
24	32399.9766	8678177	849	1158	ASL321	4981161	47.641395568847699	13.8284721374512	34975
25	32399.9062	8678178	378	1564	AUA562H	4458076	47.709388732910199	11.9068517684937	35025

atsdb_client ×

Export complete.

OK

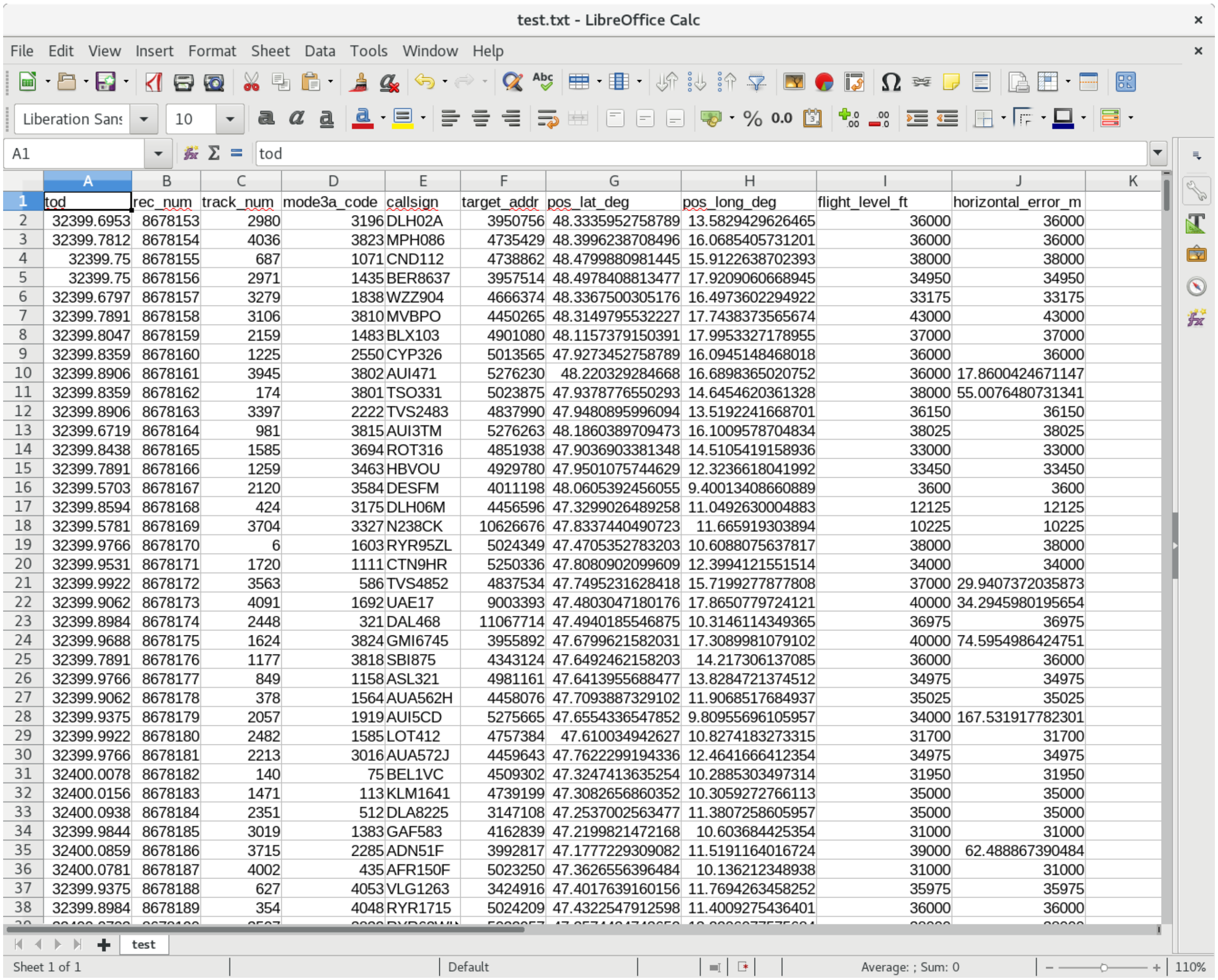
Meta, tod
Meta, rec_num
Meta, track_num
Meta, mode3a_code
Meta, callsign
Meta, target_addr
Meta, pos_lat_deg
Meta, pos_long_deg
Meta, modec_code_ft
Meta, horizontal_error_m

□ Use Presentation

☑ Overwrite Exported File

Figure 21: Listbox View export done

The exported file can be opened in any editor, or for example imported into LibreOffice Calc.


test.txt - LibreOffice Calc
File Edit View Insert Format Sheet Data Tools Window Help
Liberation Sans
10
A1
tod
test
Sheet 1 of 1
Default
Average: ; Sum: 0
110%

	A	B	C	D	E	F	G	H	I	J	K
1	tod	rec_num	track_num	mode3a_code	callsign	target_addr	pos_lat_deg	pos_long_deg	flight_level_ft	horizontal_error_m	
2	32399.6953	8678153	2980	3196	DLH02A	3950756	48.3335952758789	13.5829429626465	36000	36000	
3	32399.7812	8678154	4036	3823	MPH086	4735429	48.3996238708496	16.0685405731201	36000	36000	
4	32399.75	8678155	687	1071	CND112	4738862	48.4799880981445	15.9122638702393	38000	38000	
5	32399.75	8678156	2971	1435	BER8637	3957514	48.4978408813477	17.9209060668945	34950	34950	
6	32399.6797	8678157	3279	1838	WZZ904	4666374	48.3367500305176	16.4973602294922	33175	33175	
7	32399.7891	8678158	3106	3810	MVBPO	4450265	48.3149795532227	17.7438373565674	43000	43000	
8	32399.8047	8678159	2159	1483	BLX103	4901080	48.1157379150391	17.9953327178955	37000	37000	
9	32399.8359	8678160	1225	2550	CYP326	5013565	47.9273452758789	16.0945148468018	36000	36000	
10	32399.8906	8678161	3945	3802	AUI471	5276230	48.220329284668	16.6898365020752	36000	17.8600424671147	
11	32399.8359	8678162	174	3801	TSO331	5023875	47.9378776550293	14.6454620361328	38000	55.0076480731341	
12	32399.8906	8678163	3397	2222	TVS2483	4837990	47.9480895996094	13.5192241668701	36150	36150	
13	32399.6719	8678164	981	3815	AUI3TM	5276263	48.1860389709473	16.1009578704834	38025	38025	
14	32399.8438	8678165	1585	3694	ROT316	4851938	47.9036903381348	14.5105419158936	33000	33000	
15	32399.7891	8678166	1259	3463	HBVOU	4929780	47.9501075744629	12.3236618041992	33450	33450	
16	32399.5703	8678167	2120	3584	DESFM	4011198	48.0605392456055	9.40013408660889	3600	3600	
17	32399.8594	8678168	424	3175	DLH06M	4456596	47.3299026489258	11.0492630004883	12125	12125	
18	32399.5781	8678169	3704	3327	N238CK	10626676	47.8337440490723	11.665919303894	10225	10225	
19	32399.9766	8678170	6	1603	RYR95ZL	5024349	47.4705352783203	10.6088075637817	38000	38000	
20	32399.9531	8678171	1720	1111	CTN9HR	5250336	47.8080902099609	12.3994121551514	34000	34000	
21	32399.9922	8678172	3563	586	TVS4852	4837534	47.7495231628418	15.7199277877808	37000	29.9407372035873	
22	32399.9062	8678173	4091	1692	UAE17	9003393	47.4803047180176	17.8650779724121	40000	34.2945980195654	
23	32399.8984	8678174	2448	321	DAL468	11067714	47.4940185546875	10.3146114349365	36975	36975	
24	32399.9688	8678175	1624	3824	GMI6745	3955892	47.6799621582031	17.3089981079102	40000	74.5954986424751	
25	32399.7891	8678176	1177	3818	SBI875	4343124	47.6492462158203	14.217306137085	36000	36000	
26	32399.9766	8678177	849	1158	ASL321	4981161	47.6413955688477	13.8284721374512	34975	34975	
27	32399.9062	8678178	378	1564	AUA562H	4458076	47.7093887329102	11.9068517684937	35025	35025	
28	32399.9375	8678179	2057	1919	AUI5CD	5275665	47.6554336547852	9.80955696105957	34000	167.531917782301	
29	32399.9922	8678180	2482	1585	LOT412	4757384	47.610034942627	10.8274183273315	31700	31700	
30	32399.9766	8678181	2213	3016	AUA572J	4459643	47.7622299194336	12.4641666412354	34975	34975	
31	32400.0078	8678182	140	75	BEL1VC	4509302	47.3247413635254	10.2885303497314	31950	31950	
32	32400.0156	8678183	1471	113	KLM1641	4739199	47.3082656860352	10.3059272766113	35000	35000	
33	32400.0938	8678184	2351	512	DLA8225	3147108	47.2537002563477	11.3807258605957	35000	35000	
34	32399.9844	8678185	3019	1383	GAF583	4162839	47.2199821472168	10.603684425354	31000	31000	
35	32400.0859	8678186	3715	2285	ADN51F	3992817	47.1777229309082	11.5191164016724	39000	62.488867390484	
36	32400.0781	8678187	4002	435	AFR150F	5023250	47.3626556396484	10.136212348938	31000	31000	
37	32399.9375	8678188	627	4053	VLG1263	3424916	47.4017639160156	11.7694263458252	35975	35975	
38	32399.8984	8678189	354	4048	RYR1715	5024209	47.4322547912598	11.4009275436401	36000	36000	

Figure 22: Listbox View export in LibreOffice Calc