

# Description of the SyncServer and FloeNavi App features

Note: The offline version of this document does not have pictures included.

## Table of Contents

1. A prologue
2. Initial grid configuration
3. Grid and station status messages
4. Connection to the AIS transponder
5. GPS status icon
6. Pin for admin and special security access
7. Support for map images in the grid background
8. Deletion of a grid on the sync server
9. Synchronising additional base stations
10. SyncServer and Tablet Logging and Logfiles
11. Device operations export flags
12. Deletion of device information

## Prologue

The FloeNavi app was developed using concepts and algorithms previously developed for the FloeNavigation project. The admin manual of the predecessor app provides an introduction for the concepts of the grid, and the role of the different base stations (origin, x-axis, additional stations).

The behaviour of the FloeNavi app and SyncServer deviates from the previous implementation, to either fix issues in the original implementation, or to improve the workflows w.r.t to the received feedback, UI changes and new requirements.

## Short list of possible, noticeable and visible differences to the old app behaviour

The nominal use case requires a grid built from an AIS base station at the origin position ( $x=0, y=0$ ) and a station defining the bearing of the x-axis, called x-axis station.

If these two stations are available and have sent their LAT/LON positions at least once, the app **will** compute a grid and display positions in the grid view.

The color of the grid status icon is an indicator for the accuracy (or usability) of x/y positions computed for the grid.

In the nominal case, with an origin station, and an x-axis station, the icon is **GREEN only** when

- both stations have sent a datum containing latitude and longitude (LAT/LON),

- course over ground, and speed over ground (COG/SOG) values, and
- the datum was received not longer than 10min ago.

The app will always use the latest received data from each station. The grid view popup for each station will always show the latest received LAT/LON positions.

#### **NOTE:**

If COG/SOG values for a station are available, the app will extrapolate a new position (LAT/LON) based on the last received datum, and the elapsed time since the last message was received.

If the SOG/COG values are not available, the positions of the stations on the grid view **WILL** visibly jump, with each update, unless the ice is absolutely not moving, or the app is used on solid ground.

For the X/Y positions in the grid view, the app always computes the drift-applied **estimated** origin position and bearing of the grid, before translating a LAT/LON position of a mobile station, or the tablet's own position into X/Y coordinates.

The app does not iteratively compute the positions of stations at fixed intervals, but only at specific events:

- the app receives a message with LAT/LON and optionally SOG/COG for a base-station, typically every 3min
- the app receives a message with LAT/LON and optionally SOG/COG for a mobile station, typically every 3min, maybe faster if the station is moved, or is sending at shorter intervals (mothership).
- the tablet requests an update of its own X/Y position. This is triggered at regular intervals and when the located tablet position has moved a certain distance.

Starting point for the drift-related extrapolation of positions is always the last received datum.

This approach was chosen for multiple reasons:

- event based processing reduces the resource usage in many scenarios.
- the app translates the LAT/LON position the instant it receives an AIS message for a station, and not after some, albeit short, waiting time. The grid view immediately updates and shows the new position of that station.
- removing the errors introduced by re-using the result of previous computations, because of e.g. rounding errors.
- Android is not a real-time system, the interval at which the Android system calls a service is not guaranteed.
- If the tablet or the app is inactive, stopped, sleeping, shut down, resetting, or otherwise interrupted, the app will start from the last received datum, and apply the positional drift correction based on the elapsed time since then. The app will not continue from an old result produced before the interruption.

- Opening the grid view will trigger an update of the grid positions and station positions (since v3.1)

— TBD —

### **App: Initial grid configuration**

To use the FloeNavi app, a grid needs to be configured by initially providing the MMSI of two stations:

- the origin station, which will become the origin of the grid, defining the position  $x=0.0\text{m}$ ,  $y=0.0\text{m}$  on the grid
- the x-axis station, which will have the role to define the bearing of the x-axis of the grid. On the grid it will be located somewhere along the x-axis, with  $y=0.0\text{m}$ , and the x-value computed at each update.

The x-value of the x-axis-station position is the distance of the x-axis station from the origin station. Currently, this distance is only required to render the background map image.

### **Admin view - Initial grid configuration**

The grid is configured by opening the initial grid configuration view on the admin dashboard.

For nominal operation, a grid is configured by providing the MMSI of the origin station, and the MMSI of the x-axis station.

- The configuration of a grid does not require a connection to the AIS-Transponder.
- It is not possible to configure a new grid, unless an old grid is deleted in the app.
- Any MMSI can be configured as the origin or x-axis station MMSI
- The origin station MMSI must be different from the x-axis station MMSI

If, within the last 10min, valid positional data was received for a station, the station MMSI will appear in the MMSI selection list. This list is updated approximately every 15s.

The configuration view itself will also show if there is recent positional data received from the selected station, displayed in the lower half of the screen as LAT/LON values.

However, the admin may decide to accept the configuration at any time, even if the app did not receive any data for the stations yet.

By confirming the configuration, the grid is **CONFIGURED**. The app will start the **INITIALIZATION** in the background, and the grid status icon will reflect the current state of the grid.

### **Notes:**

- In addition to the initial grid configuration, a grid can be downloaded to the app from the sync server using the sync view
  - if there is currently no grid configured in the app. In this case there is already a grid in the app, it needs to be deleted
- If a grid is downloaded from the sync server, the grid will also be **CONFIGURED**, and start with the initialization phase

### The initialization phase

Instead of a one-time initialization phase at the time of grid configuration, the grid is using a time-window based grid quality estimation.

- For release v3.1, a time window of 30min was defined, in which for each base station at least 4 valid AIS messages had to be received, for the grid to be considered GREEN (stable).
- After receiving feedback for release v3.1, the time-window was changed to 10min and 1 valid AIS message. Effectively, there no longer is any initialization phase.

Every time a grid interaction is triggered, i.e. receiving an AIS message with positional/movement data, the grid quality (or accuracy) is measured. Based on the timeliness and content of the messages, the status icon is changed.

However, the grid will always try to keep positions, bearings and drift of the grid and registered stations as accurate as possible, based on the information that is currently available.

### Grid status icon

#### Notes:

- When starting the app, the grid status icon will always be red.
- The status icon shows the **current** state of the grid, it is an indicator for the quality of the data used to compute positions, bearings and drift of the grid and registered stations on the grid.

The following color indicators are supported by the status icon:

- RED: (**LOST** grid)
  - the grid is not yet configured
  - the grid has not received valid data for enough base station on the grid within 10min.
- YELLOW: (**DEGRADED** grid)
  - the grid has not received data for each base station on the grid within 10min, but it has enough data to reliable compute a grid from data that was received within 10min, i.e. because additional stations have been added.

- the grid will not follow the ice drift and stations may jump with each update, because incomplete data (i.e. missing COG/SOG) was received for a base stations (in v3.2 this scenario was considered **LOST** and showing RED icon)
  - It is dangerous to add or remove stations from the grid in this state.
- GREEN: (**NOMINAL** grid)
  - the grid has received complete data for each base station (origin, x-axis and additional) within 10 min.

### What is a valid AIS message

The AIS protocol allows for the fields in the messages to be marked as **invalid** or **unavailable**.

- A value of 360.0 degrees for the course over ground value (COG)
- A value of 102.3 knots for the speed over ground value (SOG)
- A value of 0x6791AC0 (181°) for longitude (LON)
- A value of 0x3412140 (91°) for latitude (LAT)

~~For the estimation of the quality of the grid, a station that is sending positional data (LAT/LON), but not COG/SOG will be considered OFFLINE.~~

For the estimation of the quality of the grid, a station that is sending positional data (LAT/LON), but not COG/SOG will be considered as if sending a full set of data. (since v3.2.1)

Even though the COG/SOG data is not available, the grid can be computed to a certain accuracy with just the positional data (LAT/LON).

The grid will also try to display these positions anyway, as they are **mostly** accurate for a short time after receiving the position data, but they will drift away from their **true** position quickly. This is also indicated by a **RED YELLOW** grid icon (since v3.2.1).

- Typically, COG is not available if SOG is below 0.4kn (0.2m/s).
- For computation, COG and SOG is assumed to be 0 if one of the two values is not available. Otherwise, always assuming a course of e.g 360° would further increase the inaccuracy.
- Typically, without COG/SOG, the expected error is approx. 36m after 3min.

### Advanced Topic: Details of the grid quality utility function

- A basic score/point system is used to measure the quality of a grid.
  - the origin station provides 200 points
  - the x-axis station provides 100 points
  - each additional station provides 100 points

- The expected score is based on the synchronized grid configuration, which is also stored at the sync server.
- The measured score is based on a communication monitor in the app that records a timestamp for the messages received from each station.
- Since v3.2.1, any message containing LAT/LON is contributing to the new score, independent of the availability of COG/SOG.
- Since v3.2.1, a new score based on the availability of COG/SOG is introduced, to distinguish between different types of DEGRADATION, i.e. based on missing COG/SOG, or failed stations in a redundant setup.
- If the measured score is equal to the expected score, the grid icon is colored GREEN.
- If the measured score is below 300 points, the grid icon is colored RED.
- If the measured score is equal or above 300 points, but below the expected score, the grid icon is colored YELLOW. This is called **DEGRADED**, as there is enough information to compute a grid, but not with data from all configured stations.

The score of 300 can be reached by configuring

- an origin station and a x-axis station (300 points)
- an origin station, x-axis-station and one additional base station (400 points)
- ...

#### Notes:

- To add additional stations, the grid needs to be setup with origin and x-axis station. For triangulation, the initial X/Y-position of the additional stations need to be recorded when these are added to the grid.
- The app will deny the “base station recovery” action, if the score would fall below 300 points.
- Not all possible scenarios can be synchronized with the sync server
- Not all possible scenarios are supported by the grid drift-calculation algorithm

#### Example

##### *Setting up a nominal grid*

- Origin station: 200 points
- X-Axis station: 100 points

This is most basic valid configuration, because the configuration provides 300 points. Initially, the grid icon is RED. A few of the possible scenarios:

- If the grid has received valid data for the origin station, and now receives a valid datum for the x-axis station, the color of the grid icon is set GREEN.
- If the grid has received a valid datum for the x-axis station, but the communication monitor has not recorded a valid datum from the origin station within the last 10min, the color of the grid icon is set RED.

The grid is updated in all cases, and will display the stations and icons. However, the accuracy of the computation degrades with the age of the last valid message, as the position of a station is extrapolated from the last received LAT/LON and COG/SOG values.

#### *Adding a base station to the grid*

- Additional Station S1: 100 points

The configuration now provides 400 points. Initially, the grid icon is RED. A few of the possible scenarios:

- If the grid has received valid data for the origin station, and now receives a valid datum for the x-axis station, the color of the grid icon is set YELLOW. The grid is fully functional, but there is no data for station S1.
- If the grid has received valid data for the origin and x-axis station, and now receives a valid datum for the additional station S1, the color of the grid icon is set GREEN.
- If the grid now receives a valid datum for the origin station, but the last recorded message for the x-axis station has been 15min ago and the last recorded message for station S1 has been 10s ago, the grid icon is set YELLOW. Since the data for station S1 is more current than the x-axis station, the grid bearing is reconstructed from the last recorded bearing between origin and S1 and the X/Y-position of S1 on the grid. Additional stations increase the accuracy, especially if they are spaced further apart.

#### **Grid and station messages**

The grid algorithm and communication monitor will emit **ALERTS** that are displayed in the grid view. Such an alert can be triggered by any update to the grid algorithm, i.e. any message received from the AIS transponder, or a scheduled grid re-calculation request to update the grid view and icon positions (typically performed when the tablet moves a certain distance).

The app uses an event-driven mechanism. If an AIS message is received for a specific station, this event also triggers e.g. a full scan of the communication log, and for any station that has not sent a message since 10min, a LOST-event is sent to the grid view, which will react by removing these stations from the grid.

Therefore, the time displayed in the LOST alert message may not be exactly 600s, which is the defined timeout, but any number larger than this value, depending on the activity of the tablet.

#### **Station LOST**

**No data received from AIS Station xxxxxxxx since xxxx seconds**

The grid algorithm has detected, that a station has not sent any communication since 10min. This station is than marked as offline and will no longer be displayed

on the grid until a new message is received.

### **Station not sending COG/SOG**

**Station xxxxxxxxx has not sent a COG/SOG datum**

The station has sent a message including LAT/LON position, but without COG/SOG data. If the station is a base station, the grid will be inaccurate and the grid status is decreased to at least YELLOW (degraded). However, the grid will still be calculated with available data from the LAT/LON position. and data from other stations.

### **Connection to the AIS transponder**

On the tablet, a dedicated service is maintaining the connection to the AIS transponder. This service does appear in the Android top action bar as an Icon with the title “FloeNavi App - AIS service running”. The AIS service is running in the background, independent of the FloeNavi app.

#### **Connection Details**

- Every 2.5s the AIS service is trying to establish a connection to the AIS transponder, if there is no existing connection.
- If the AIS service does not receive a message on the connection to the AIS transponder for 30s, it forcefully terminates the connection and tries to re-connect.
- The AIS service does not have access to the WLAN or Ethernet settings on the tablet.
- The AIS service always connects to port 2000 on ip address 192.168.0.1.

#### **Notes:**

- If a connection is interrupted and is not re-established automatically, the system must wait for a timeout to occur.
  - The default timeout on Android is 2h. The FloeNavi app uses a modified timeout of 30s
  - If the AIS network is very quiet, i.e. a window of 30s in which no transponder sends a message, the app may trigger a timeout even though the connection to the AIS transponder is stable.
  - The AIS transponder also has a timeout on TCP connections, the exact length of the timeout is not known.
    - \* The timeout on the AIS transponder seems to be in the range of a few minutes
- It was observed, that Android sometimes disconnects from a WLAN if the access point does not provide internet access, and other WLANs are available. In such a case, Android sends a notification with a request to confirm that the WLAN without internet access should be used anyway.

This decision can be made “permanent” so that in the future the WLAN is automatically used.

- One of the two tablets in the test did not allow to make this decision “permanent” and repeatedly dropped the connection every time a network change occurred, i.e. leaving the WLAN range for a moment

### GPS status icon

Locating the tablet with the tablet built-in GPS receiver is a time and resource consuming operation. Therefore, the tablet uses the GPS only when required, and this is reflected by the GPS status icon.

- The tablet does not constantly try to acquire a GPS fix.
  - Google services provide a helper library for GPS services, but from feedback it was understood that not all tablets have the correct update of the google services installed.
- The GPS status icon is RED by default, the icon will be updated when the app shows a view that requires a GPS fix:
  - the grid view
  - adding logistical installations
  - adding waypoints
  - adding device operations
  - adding a base station
  - initial grid configuration

The Android OS does not have a concept for GPS status. - A request can be made to locate the tablet via GPS. This request is either successful, or runs into a timeout.

The FloeNavi app requests the GPS position of the tablet on views which require a GPS fix. When the position is established, the GPS status icon on that view changes from RED to GREEN. Once in this state, the tablet periodically requests a new GPS fix in the background to update the position.

On pages that do not require the tablet position, i.e. Dashboards, Sync View, List Views, the GPS status icon will permanently stay RED.

### SyncServer: Setting up admin and security PIN

The admin view and grid deletion view are protected by two different 4-digit PIN codes. The codes are configured on the SyncServer.

When syncing the tablet with the SyncServer, the PIN codes are updated in the FloeNavi app.

The example `application.yaml` is provided in the upgrade instructions and in the FloeNavi-Patch archive `etc`-folder.

**Note:** - The initial PIN configured on the FloeNavi app, for both admin and security pin, is **0000**.

## Grid background map image support

Displaying a map image in the app requires two files to be stored in an admin-defined directory on the SyncServer host. The directory used for providing those files is configured in the `application.yaml` file with the keyword `maps`

The example `application.yaml` is provided in the upgrade instructions and in the FloeNavi-Patch archive `etc`-folder.

### 1. The map image file `map.jpg`

- This must be a standard JPEG file.
- The app was tested with a jpg of size 3353x1731 pixel and 4MB. Images with 12Mpx (4000x3000) or higher should also be possible.
- The position of the origin and x-axis station need to be manually locatable on the image.

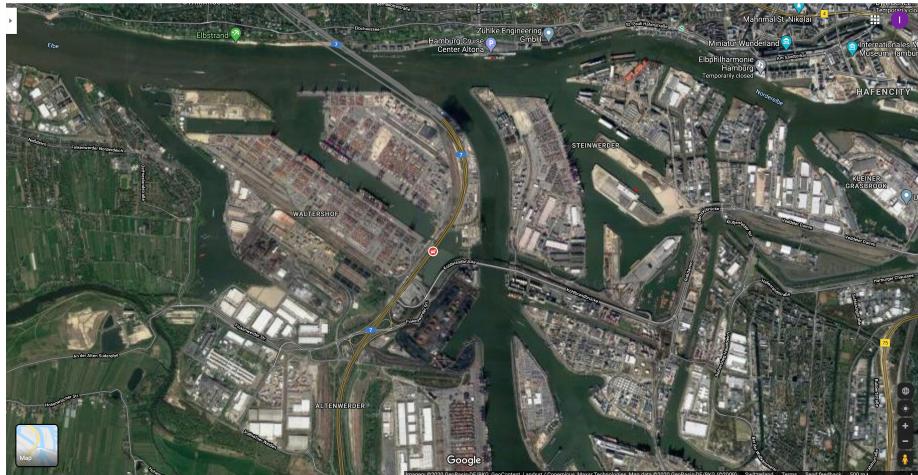


Figure 1: Example Map

On the example map:  
- The origin station is AIDAperla (red dot at Steinwerder Quay, right of center),  
- the x-axis station is Europa2 (red dot at hamburg cruise center, center and top border).

### 2. The image configuration file `map.info`

- The file contains the pixel coordinates of the two stations

For the example map:

```
origin 2300 681  
x-axis 1774 201
```

- origin is at x=2300, y=681

- x-axis is at x=1774, y=201

The position x=0, y=0 is the top-left corner of the image

### 3. Removing the map image

- Remove the files from the map directory on the SyncServer
- Perform a sync (download) from the SyncServer

If there is no image in the map directory on the SyncServer, the map image on the app will be deleted.

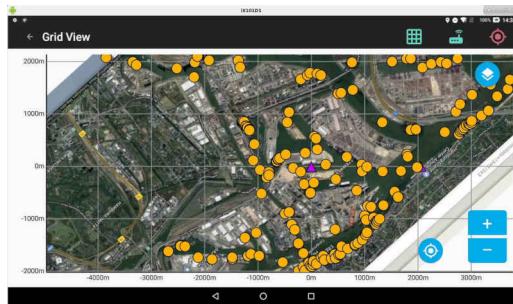


Figure 2: Example Screenshot

### Grid Deletion

If there is a grid configured on the SyncServer, and the grid on the app has been deleted with the security pin and a new grid has been initialised, the SyncServer must be prepared to receive the new grid from the app.

A specific url has to be called on the server and the admin token has to be provided.

The following `curl` command can be used to reset the grid on the SyncServer:

```
curl -X DELETE -H "Authorization: Bearer <token>" http://10.0.2.2:8080/grid
```

`<token>` needs to be replaced by the `base64` encoded `adminToken` that is specified also in the SyncServer `application.yaml` file.

The `base64` encoded token for the default admin token is `MDEyMzQ1Njc4OWFiY2R1Zg==`.

The example `application.yaml` is provided in the upgrade instructions and in the FloeNavi-Patch archive `etc`-folder.

### Encoding the admin token with base64 on Windows

The `certutil` provided by Windows can be used to encode text. For this, the admin token needs to be saved in a file, i.e. `in.txt`.

The command `certtool -encode in.txt out.b64` will create the file `out.b64` with the encoded content of `in.txt`

Newlines and linefeeds are encoded, too. Therefore care needs to be taken when creating the infile.

In the outfile, the encoded text will be wrapped between the lines -----BEGIN CERTIFICATE----- and -----END CERTIFICATE-----.

### Synchronization of additional base stations

Modifications to an existing grid can only be performed from one tablet. When a tablet is making changes to a grid, i.e. adding or removing base stations, these changes must be done from a single tablet.

The SyncServer will accept only one update to the additional base stations of a grid. Any further modifications to the additional stations need to be authorized similar to the deletion of a grid.

```
curl -X DELETE -H "Authorization: Bearer <token>" http://10.0.2.2:8080/grid/additional-stations
```

<token> needs to be replaced by the base64 encoded adminToken that is specified also in the SyncServer `application.yaml` file.

The base64 encoded token for the default admin token is `MDEyMzQ1Njc4OWFiY2RlZg==`.

*Note:*

- This authorization workflow blocks concurrent modifications of the grid done by multiple tablets. By allowing only one source of modifications to the grid ensures that concurrent changes done on multiple tablets do not destroy the grid.
- “Deleting” the additional stations does not destroy the existing grid.

## Logging and Logfiles

### SyncServer audit log

The SyncServer writes message about events, i.e. synchronization of device operations, into a dedicated log file.

On the SyncServer host, the admin must specify a directory to contain the log files. This setting is available in the file `application.yaml` in the server `etc` directory.

The corresponding configuration option is called `logging/file/path`.

```
logging:  
  file:  
    path: "<insert absolute path of log directory here>"
```

The upgrade instructions provide an example `application.yaml` file, as is the `etc` directory in the FloeNavi-Patch archive.

**Note:** - On Windows, the backslashes in the directory name need to be escaped as \\ when adding them to the configuration file.  
- C:\Program Files\FloeNavi\Logs needs to be written as C:\\Program Files\\FloeNavi\\Logs

### SyncServer tablet log

Every time a tablet is synchronized with the SyncServer, the tablet will transfer up to 10MB of the latest recorded log messages to the SyncServer.

For the tablet log files, a dedicated folder **tablets** is created in the log directory. For each tablet, the name of the logfile is generated from the unique Android-ID of the tablet and a timestamp.

The files are plain text files with timestamped messages, which can easily be split into smaller parts and compressed with ZIP or 7-Zip with a very high compression ratio.

The following list contain messages indicating typical issues.

### Connection issues

- The tablet is on a network where host 192.168.0.1 is available, but is not responding to requests on port 2000:

AisServiceRunnable: java.net.ConnectException: failed to connect to /192.168.0.1 (port 2000 refused)

- The tablet is not connected to a network, or connected to a local (un-routed) network without access to 192.168.0.1:

AisServiceRunnable: java.net.ConnectException: failed to connect to /192.168.0.1 (port 2000 refused)

- The tablet has not received any messages on the connection to the AIS transponder located at 192.168.0.1 for 30s:

AisServiceRunnable: java.net.SocketTimeoutException

- The connection to the SyncServer failed:

SynchronisationViewModel: java.lang.RuntimeException: java.net.ConnectException: Failed to connect to 192.168.0.1:2000 (port 2000 refused)

### Received AIS messages

- A **good** message
  - This transponder probably connects to a compass and log, and does not use GPS for speed and course
  - The SOG of 0.0kn is a correct value, in AIS a value of 102.3 means unavailable, the decoder will log a value of null instead.

AisServiceRunnable: Received line: !AIVDM,1,1,,B,139KSs0v@00eWWjN';A'D8NN00RB,0\*24

AisServiceRunnable: Decoded geo info mmsi: 211215340, position: (lat:53.5258766667 lon:9.965555555555555)

- A message missing COG and SOG

```
AisServiceRunnable: Received line: !AIVDM,1,1,,A,139a8jPP010eU=JN`p8>4?vL0<3g,0*69
AisServiceRunnable: Decoded geo info mmsi: 211437770, position: (lat:53.5450133333 lon:9.957
```

- A message missing LAT/LON and COG and SOG

```
Received line: !AIVDO,1,1,,,B39>ab03wk?8mP=18D3Q3wwQhP00,0*7D
Decoded geo info mmsi: 211003816, position: (lat:91.0 lon:181.0), movement: (sog:null cog:nu
```

- A **good** message containing voyage data (vessel name)

```
AisServiceRunnable: Received line: !AIVDM,2,1,2,B,539q8nT00000@S037<0<58j0HD@@E9<Dp000001S1
```

```
AisServiceRunnable: Decoded name info mmsi: 211699930, name: CARL FEDDERSEN
```

- Some messages cannot be decoded, but this should not be an issue
  - This seems to be AIS transponder housekeeping message

```
AisServiceRunnable: Received line: $AIALR,083852,002,A,V,AIS: Antenna VSWR exceeds limit*78
AisServiceRunnable: de.awi.floenavi.ais.decoder.AISDecoderException: Message TalkerID is not
```

```
AisServiceRunnable: Received line: $AIALR,100002,007,A,V,AIS: UTC sync invalid*2D
```

```
AisServiceRunnable: de.awi.floenavi.ais.decoder.AISDecoderException: Message TalkerID is not
```

```
AisServiceRunnable: Received line: $AIALR,100003,026,A,V,AIS: no position sensor in use*71
AisServiceRunnable: de.awi.floenavi.ais.decoder.AISDecoderException: Message TalkerID is not
```

```
AisServiceRunnable: Received line: $AIALR,100003,029,A,V,AIS: no valid SOG information*74
AisServiceRunnable: de.awi.floenavi.ais.decoder.AISDecoderException: Message TalkerID is not
```

### SyncServer export flags

To support the requirement of blocking device operations from export to the DSHIP proxy system, the database table managing the device operations supports an additional field: `export_status` in table `operation`.

The allowed values for this field are:

- **PENDING**: The device operation was synced from the tablet and is ready for export
- **EXCLUDED**: This device operation will not be exported
- **EXPORTED**: The device operation was retrieved by the DSHIP proxy and will not be exported again.

### Deleting device information

Device information stored on the sync server can be deleted (flagged as **DELETED**), and as such they are no longer visible in the FloeNavi app.

These device information do not get physically deleted from the database, therefore the integrity of the database stays intact.

Using the device information update workflow, any device for which an update is received will be undeleted.

### **Example**

Deleting the device with the device id 1:

```
curl --verbose --request DELETE --url http://10.0.2.2:8080/devices/1
```