ReportLoq

Technical User Manual

Revision 2.12, 12th May 2025

Environmental Reporting

1 Contents

2 Preface

This user manual describes how to setup, install and configure ReportLoq and is intended for technical staff only.

The manual explains the administration interface of ReportLoq with the possibilities it brings. It is not intended for ordinary users of ReportLoq. They should instead read the “ReportLoq User Manual” which is embedded in the ReportLoq software and can be obtained using the web interface for ReportLoq.

Administrative personal can use the contents of this manual to commission a ReportLoq Soft installation, or to use the embedded OPC UA server to extract data.

If you are evaluating or reconfiguring an existing system, just read on. If, however you are commissioning a new system, please make sure you have a license key and a configuration ready, or contact Olicem at https://www.olicem.com/en/contact/.

3 Installation

Installing ReportLoq is carried out by executing the ReportLoq installer on a dedicated Windows server. This will, in addition to ReportLoq, install OpenJDK and MongoDB Community server on the target server.

Please note that the server should as minimum have:

- Be Windows 10 / Windows server 2022 or newer
- 16GB memory
- Have at least 20GB free diskspace + 5GB per measured stack
- Have at least 2 cores (4 cores recommended)

The installer will install ReportLoq as default inside %ProgramFiles(x86)%. MongoDB and OpenJDK will be installed in their respective default directories.

MongoDB and ReportLoq will be installed as Windows services, which will launch with Windows.

Note that: The desktop icon will restart ReportLoq with UI and start ReportLoq as a service when closing the UI. This interface is intended for technical staff only. Users will not need access to this UI.

4 First launch

ReportLoq will prompt for the correct start approach on first launch. Three options are available:

- **Setup as new installation**
  First time installations of ReportLoq should choose this option.

- **Restore from local backup**
  Restoring a zipped backup of ReportLoq? Choose this option.

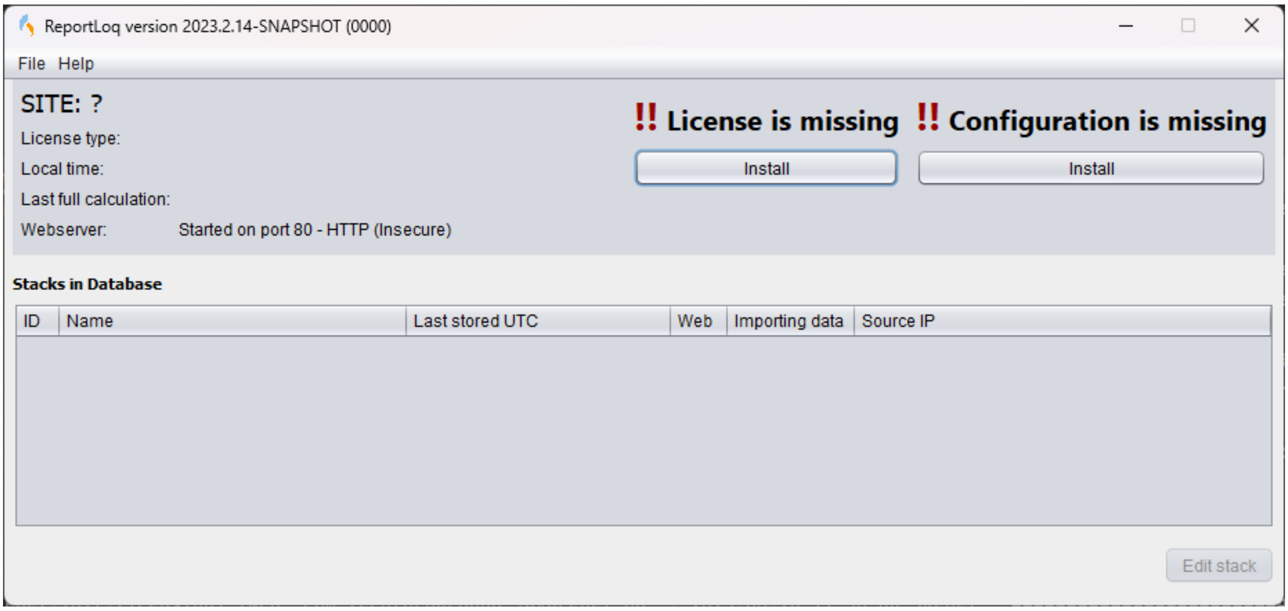
Note: “Installation number” is for advanced users and can be disregarded.

5 Setting up ReportLoq

The first thing to do after installing ReportLoq is to install the missing license and configuration.

5.1 Installing a license

Press “Install” below the license warning and choose “Change License”. This lets you choose from using a trial license or installing an existing license.



ReportLoq version 2023.2.14-SNAPSHOT (0000)

File Help

SITE: ?

License type:

Local time:

Last full calculation:

Webserver: Started on port 80 - HTTP (Insecure)

!! License is missing

!! Configuration is missing

Stacks in Database

ID	Name	Last stored UTC	Web	Importing data	Source IP

Figure 1 – Main window

ReportLoq will not calculate average values or start the embedded webserver if the license is missing, invalid or not activated. You must therefore choose from using a purchased license key, a purchased license text or an evaluation key. Evaluation keys and license keys will need online activation to work. Please make sure the firewall is open outbounds for https://www.reportloq.com

5.1.1 Evaluation keys

ReportLoq comes with a 60 days evaluation key with full functionality. The key can be activated only once per hardware device.

5.1.2 Purchased license keys

License keys are much like Windows keys and will only activate on one device. Once activated, the license key will lock to the hardware is activated on. Unless agreed otherwise, activation is not possible on other devices. **Activated keys cannot be deactivated. Make sure you activate your key on the correct server.**

5.1.3 Purchased license text

License text are keys delivered in a file. This type of key can be installed without online activation. The license text is obtained from Olicem and will require you to forward the hardware ID of the server you installed ReportLoq on. This can be found by using the “Display hardware ID” button in the License Validation popup.

A license text key can be activated only on the hardware device it was generated for.

5.2 Configurations

The configuration contains the environmental setup needed to do calculations in ReportLoq. ReportLoq will not work without a configuration installed.

The configuration contains information about

- The company you are installing ReportLoq for
- The number of lines (stacks) needed
- Language files needed to support the locale you intend to use.

Trial customers can use any of the provided example files in the "\ReportLoq\examples" folder. This will start up ReportLoq with a predefined set of measurement points. This approach is suitable for evaluation purposes only and will demonstrate how ReportLoq can be configured.

Customers with a purchased license should contact Olicem A/S and ask for a signal clarification sheet. This must be filled in and returned to Olicem. It will be returned as a configuration which can be imported in ReportLoq.

5.3 Edit stack

The entry point to importing data to ReportLoq is in the "Edit stack" option of Figure 1 – Main window. Opening this is possible only after selecting an existing stack. An example window is displayed on Figure 2 - Stack setup. This example logs 22 components every 10 seconds using OPC UA.

The window presents several configuration options. These are explained in the following section:

- Components
  The number of components you would like to log. Please note that changing this will require a new configuration.

- Unique ident
  Only used for ReportLoq+ Cloud customers.

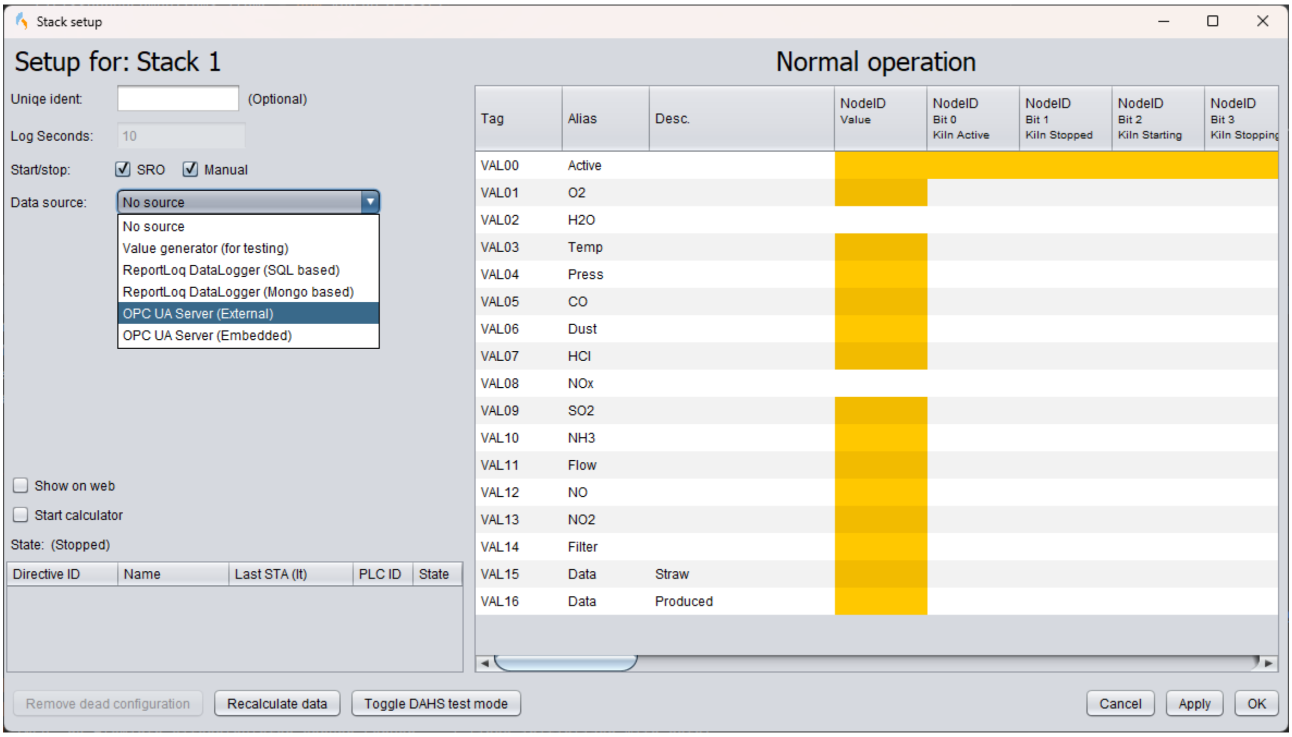
- Log Seconds
  How often to log. Please note that this MUST add up to one minute and not be more often that every 10 seconds. Valid values are 10,20,30,60.

- Start/stop
  How should ReportLoq handle operation? *SRO* means the command to start/stop reporting is transmitted from the control system. *Manual* means the user will handle start/stop in the user interface. Selecting both will allow the user to disregard the signal from the control system. Selecting none is not allowed.

- Show on web
  Displays this line in the web-interface. The line can collect and calculate without being visible for

the end user, as long as this option is deselected.

- Start import and calculator
  Selection this option will initiate import from the selected data source and begin calculating environmental data, such as STA and LTA averages and check for exceedances.



Stack setup

Setup for: Stack 1

Uniqe ident: (Optional)

Log Seconds: 10

Start/stop: ☑ SRO ☑ Manual

Data source: No source

- No source
- Value generator (for testing)
- ReportLoq DataLogger (SQL based)
- ReportLoq DataLogger (Mongo based)
- OPC UA Server (External)
- OPC UA Server (Embedded)

☐ Show on web

☐ Start calculator

State: (Stopped)

Directive ID	Name	Last STA (lt)	PLC ID	State

Normal operation

Tag	Alias	Desc.	NodeID Value	NodeID Bit 0 Kiln Active	NodeID Bit 1 Kiln Stopped	NodeID Bit 2 Kiln Starting	NodeID Bit 3 Kiln Stopping
VAL00	Active						
VAL01	O2						
VAL02	H2O						
VAL03	Temp						
VAL04	Press						
VAL05	CO						
VAL06	Dust						
VAL07	HCl						
VAL08	NOx						
VAL09	SO2						
VAL10	NH3						
VAL11	Flow						
VAL12	NO						
VAL13	NO2						
VAL14	Filter						
VAL15	Data	Straw					
VAL16	Data	Produced					

Figure 2 - Stack setup

5.3.1 Data sources

There are several available data sources. ReportLoq DataLoggers are hardware devices sold by Olicem that will enable external data logging which increases up-time during ReportLoq down time. Logging data directly in ReportLoq is supported by the remaining items in the list.

Note: You can only choose one data source for every stack. It is not possible to use several data sources for the same stack.

5.3.1.1 Value generator

Selecting the "Value generator" will allow you to start up the system with test-data, which can be ramping, static or linear drops. Note that all data generated in this mode is marked as test-data and can therefore not be used for average value calculations and comparison towards limit values.

5.3.1.2 OPC UA server (External)

ReportLoq contains an embedded OPC UA Client, which can be used by selecting the "OPC UA Server (External)" as data source. This will connect ReportLoq to you own OPC UA server, where ReportLoq will frequently poll data. Note that OPC UA data sources does not support catch-up and needs therefore to be connected and running for reporting to work.

The option empowers you to collect data from an existing OPC UA server in an easy way, and to write back sum alarms to your own server.

Read more in section 5.3.1.2.1 - Connecting tags using OPC UA

5.3.1.2.1 Connecting tags using OPC UA

Connections to OPC UA servers are carried out by specifying [IP]:[PORT]. OPC UA servers usually run on port 4840, meaning the connection string will like something like: 10.10.1.2:4840. Connections support but does not require SSL. Username/password is not supported.

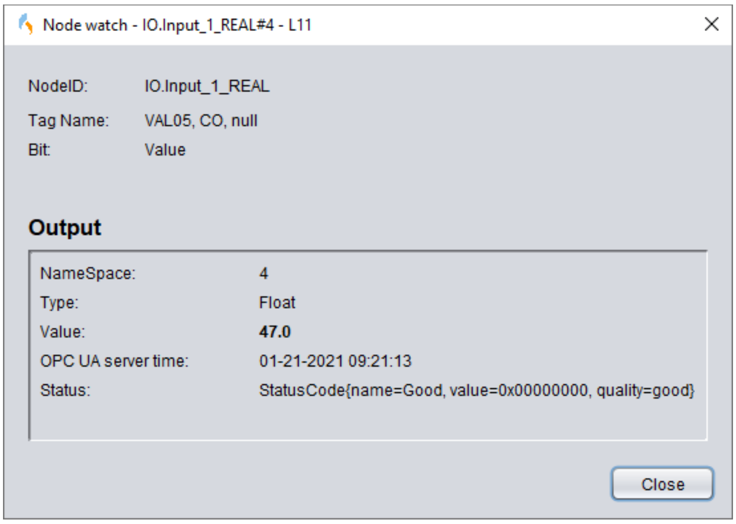
Successful connections will cause ReportLoq to display “Connected to: 10.10.1.2:4840”. This will not start data collection, but it will allow signal setup and verification for all tags.

A tag in ReportLoq should be thought of as a single measurement point. Every tag is saved along with a 32-bit Integer, containing information regarding the tag. This means tags must fetched long with the status information for the tag from which the tags value is validated. This could be invalidation of the measurement caused by malfunction or calibration on the analyser. This setup is handled in the graphical from the table displaying all tags in the first column.

The reserved columns cannot be used and is reserved for future usage.

Each tag and bit must be matched from the signal clarification Excel sheet.

The fields in the table can be double clicked, and their content changed. Fields should contain the tag in the OPC sever which should be recorded. Tags in the table can be verified for their contents on the OPC UA server by selecting a cell and clicking the button “Query Selected NodeID”. This will open a window as displayed in Figure 3 - Node watch. The node watch window is auto updated every 500ms and can be used to verify whether the cell is connected to the correct tag in the OPC server.


Node watch - IO.Input_1_REAL#4 - L11

NodeID: IO.Input_1_REAL
Tag Name: VAL05, CO, null
Bit: Value

Output

NameSpace: 4
Type: Float
Value: 47.0
OPC UA server time: 01-21-2021 09:21:13
Status: StatusCode{name=Good, value=0x00000000, quality=good}

Figure 3 - Node watch

5.3.1.2.2 Namespace

Setting namespace on tags can be done by appending a "#" followed by the name space number. E.g. IO.Input_1_REAL#4 for namespace no. 4.

5.3.1.2.3 Columns 1-3 (Descriptive columns)

Note that it is not recommended to change the values in the first tree columns (Tag, Alias, Description). These are all auto-generated during import of a configuration and changing them may lead to unexpected behaviour.

5.3.1.2.4 Columns 4-36 (Input columns)

These are all data collection columns. The first column (value) can be a Real. The rest of the columns should preferably be Booleans which evaluate to 0 or 1 (low or high). Other data types are supported but are only evaluated to 1 if the value is exactly 1.

5.3.1.2.4.1 NodeID, Value

The only tag in the input series which can be Real. The value will be stored with 3 decimals in precision.

5.3.1.2.4.2 NodeID, Bit 0 to Bit 3, Kiln signals + Bit 20 Kiln in SCADA

The four bits 0, 1, 2, 3 are only needed for the Kiln or Active tag (whichever is present). Setting these bits on other tags has no effect.

Only one of these should be high at any given time. At no time, must they all be low. Not complying with this rule will cause calculations to be erroneous.

- Bit 0 will evaluate emission limits and do mass emission calculations
- Bit 1 will not do emission calculations nor mass emission calculations
- Bit 2 will do mass emission calculations but not evaluate emission limits
- Bit 3 will do mass emission calculations but not evaluate emission limits

Setting bit 20, Kiln in SCADA high will cause ReportLoq's UI to display that the active signal is controlled by the control system.

Note: The correct configuration for OPC UA setups is always to set this to high, and to deselect Start/Stop "Manual". This configuration makes sure the user cannot select any other operation status than the one coming from the control system

5.3.1.2.4.3 NodeID, Bit 5-10, Malfunction

Setting any of these bits high will set the measurement in malfunction and cause it not to be used for environmental calculations.

5.3.1.2.4.4 NodeID, Bit 11, Testing

Setting this bit high indicates that the value is recorded during test. This allows collected test data to be disregarded from calculated values. Note that the value will appear to be valid in the user interface which allows makes it possible to see that the setup is working as expected although in test.

5.3.1.2.4.5 NoteID, Bit 12, Calibration

Setting this bit high will cause data taken during calibration not to be used for environmental calculations. It will however be possible for the user to see the value, and that the measurement was taken during a calibration.

5.3.1.2.4.6 NodeID, Bit 16-18, Manual overrides

Setting any of these bits indicates that an operator has decided to override the normal signal coming from the analyser. Setting these bits are possible from ReportLoq’s user interface and is therefore normally not configured.

5.3.1.2.4.7 NodeID, Bit 21, Wire break

Setting this high will cause the measurement not to be used for environmental calculations, and should be used whenever the control system detects a wire break on the measured value

5.3.1.2.4.8 NodeID, Bit 24-31, Available bits

These bits can be used for additional events which can be displayed for the user. These can be alarms or events but will not cause the measured value to be discarded.

5.3.1.2.5 Columns 37-45 (Output columns)

Data can be transferred back for the control system using these columns. Note that the input data needs to be calculated before it can be written back to the control system. This means there is a minor delay from the time the data is collected to the time it is written.

The OPC UA server must accept data of type Real for data to be written. Note that

- QAL3 FLD = First Level Data. Measurements collected every x seconds corrected for QAL3 entries.
- QAL2 FLD = QAL3 FLD corrected for QAL2 entries.
- SFLD = Standardised First Level Data. QAL2 entries corrected for peripheral parameters.
- VFLD = Validated First Level Data. SFLD entries deducted confidence interval.
- SSTA = Standardised Short Term Average. This is the running average corrected for peripheral parameters and will be reset whenever the average starts over.
- VSTA = Validated Short Term Average. This is the SSTA value deducted confidence interval.
- LTA Day = Long Term average, Daily. This is the running average made from VSTA values during the day. It will be reset at midnight.
- LTA Month = Long Term average, Daily. This is the running average made from VSTA values during the month. It will be reset at every 1st day of month.
- Set point = This is the value the customer should hold the VFLD at to avoid exceeding A-limit values on STA daily limit values. Note that this does not take monthly limit values into account.

5.3.1.2.6 General output

ReportLoq can generate summarised alerts and has a watchdog which can used to monitor if ReportLoq is collecting data as expected. This is be accessed using the “OPC UA Output” button.

5.3.1.3 OPC UA Server (Internal)

ReportLoq embeds an Eclipse Milo OPC UA Server for data collection and exposure. The server is automatically configured with the required UPC UA nodes based on the configuration provided during installation.

You'll need to set the setting "opc-ua.server.enabled" to true, to use this strategy. By default this will only enable data exposure over OPC-UA. To use it for data collection, you must set "OPC UA Server (Embedded) as "Data source" in the stack configuration.

Note that the path "Root -> Objects -> ReportLoq -> Output" of the OPC UA server supplies various output values from ReportLoq. The default value (-) of the "PLC-ID" field in the Stack setup will reduce the outputs to a minimum. Deleting the value or prefixing it will increase the outputs.

5.3.1.3.1 Security

The embedded OPC-UA server generates a server certificate at first startup, that must be trusted by the client. This ensures SSL cryptography.

It allows anonymous access as a default. User login can be enabled by creating an OPC_UA user in the User Administration UI. The server will enforce login after being rebooted.

5.3.1.3.2 Getting values over OPC-UA (Reading from the DAHS)

The DAHS can provide many different types of output. This includes FLD, STA, and LTA data. To read these off the embedded OPC UA server you must navigate to "Root -> Objects -> ReportLoq -> Output".

See 5.12 (Values from DAHS to Central Control System) for the full list of readable values.

Output Watchdog

Every stack in the OPC UA server includes a Boolean output node for ensuring that data in the OPC UA server is recent. This will stop toggling if data age goes above 5 minutes and its OPC StatusCodes will be marked as "Bad (0x80000000)"

5.3.1.3.3 Forcing output values (For signal testing)

It is a perquisite for forcing output values that you have enabled the embedded OPC-UA server.

You can open "OPC UA output" from the "Stack setup" window. In here, you can force values using the "Force value" column. This allows you to toggle data during signal test. Forced data will be marked with the "Good_LocalOverride" allowing the client to see that this data is simulation data in the DAHS.

Note that the window is empty if the DAHS has not received FLD data since the last startup.

Tip: For Boolean values you can numerically input "1" (for true) and "2" (for false).
Tip: For all values, you can input "=" to copy the current value to the forced field.

5.3.1.3.4 Setting data in the DAHS from using the embedded OPC-UA server

ReportLoq is capable of reading data using its embedded OPC UA server. If this strategy is chosen, all data connected to the stack must be supplied over this channel. The strategy is chosen by selecting "OPC UA Server (Embedded)" in the stack setup.

Once this is set, ReportLoq will monitor the input nodes of the embedded OPC UA server and log data when present. The nodes can be found in the OPC UA server at “Root -> Objects -> ReportLoq -> Input”.

All tags listed in the Input-node must have their values updated by an OPC UA client. Make sure to make an end-to-end signal test for all values and status signal during commissioning.

Note: During commissioning, the DAHS should always be in test-mode to avoid reporting test-data.

Input watchdog
The “Watchdog” input-node (DataType: Double) should be updated every time new data has been written to the nodes. Failing to do this, will mark all data points connected to the stack with error after 5 minutes. At startup, the DAHS will not start logging data before the Watchdog has been toggled.

Nodes with other OPC StatusCodes than “Good (0x00000000)” will be marked with error. For this reason, we recommend that you make sure to write all values before toggling updating the Watchdog at startup. *Note that the DAHS will wait 5 minutes after system startup for the watchdog to update before logging data. This prevents logging FLD data with erroneous values. Setting the DAHS into test mode will disable the 5 minutes delay. Not updated fields in the OPC UA server will be marked with wire break. Note that test mode marks all logged data with a test-flag that prevents it from being used in average calculations.*

Signal loss to AMS from the OPC UA client

It is a DAHS requirement that data is logged every 10 seconds. The OPC UA standard does however lean towards a strategy of not updating values that hasn’t changed. This means there is a risk that BIT values, and very stable signals are not updated in long periods. Toggling the OPC UA servers input Watchdog takes care of this by allowing the OPC UA server to collect the data although the OPC SourceTimestamps are old.

Broken signals can be handled in two ways in the DAHS:

1. **Using OPC SourceTimestamps (default)**
   This ensures that all nodes in the OPC UA server with a OPC SourceTimestamp older than 5 minutes (as default) is marked as old (Error). The time can be reduced by changing opc-ua.input.value.timeout.minutes if required. The DAHS will ignore values above 5 minutes.
   You:
   a. … can skip updating nodes with the broken signal. These will be logged with error after 5 minutes.
   b. … must mark nodes for broken signals with OPC StatusCode “Bad (0x80000000)”. The DAHS will log this instantly and mark it with error.

   The DAHS will additionally mark all signals with error if the Watchdog is not timely toggled.

   Skipping the datapoints due to signal loss and toggling the watchdog is only allowed if the setting is enabled.

2. **Using the watchdog only**
   When setting opc-ua.input.value.timeout.minutes to zero (0), the DAHS will not use the nodes OPC SourceTimestamp. This can be used in scenarios where the OPC UA standard enforces that e.g.

Boolean signals cannot be updated unless toggled.

You must mark nodes for broken signals with OPC StatusCode “Bad (0x80000000)”. The DAHS will such values with -1 and mark it with wire break.

The DAHS will additionally mark all signals with error if the Watchdog is not timely toggled.

5.3.1.4 ReportLoq Data Logger

The data logger options can only be used if you have a hardware ReportLoq DataLogger from Olicem. Please contact your vendor if you need one or have one that you need to reconfigure.

5.3.2 Table with directive information

The table displaying Conf. ID, Name, Last interval LT, PLC ID and State is populated whenever a new configuration is imported.

- Directive ID
  The ID of the imported configuration. This is provided from Olicem and is unique. Every directive has its own set of environmental parameters.

- Name
  The data set from an imported line can have one to many directives. This makes it possible to import data once, and to calculate separate environmental directives which is derived from the same data set.

- Last STA (lt)
  This displays the local time for the last calculated STA value in the directive.

- PLC ID
  Output data for the control system depends on the PLC ID.
  Directives with “-“ will NOT generate any output.
  - OPC UA Server data sources will write data from any found directive not marked with “-“
  - ReportLoq DataLogger data sources can write more than one directive to the control system. The value written in the PLC ID field is used as PLC suffix.

- State
  OK = This directive is present in the database and can calculate values.
  DEAD = This directive is no longer present in the database and can be removed.

5.3.3 Recalculating data

It is possible to delete calculated data and mark exceedances found as deleted. This will force the system to rerun all calculations and can come in handy if e.g. a configuration has been imported with date that lies before the current date. Exceedances which is found again at the same time as before the recalculations are unmarked as deleted and updated with the exceeded value if necessary. Comments are preserved for unmarked exceedances.

Please be aware that data recalculation may cause heavy CPU usage, and lead to delays in data logging on small systems.

5.3.4 Starting data collection

Selecting "Start import and calculator" will cause ReportLoq to begin data collection of all parameters. Any parameters not connected will be recorded with the value -1 and marked with bit 5 (malfunction). This is because ReportLoq needs all values to be present for calculations to run.

Collected data is stored in the embedded Mongo database.

5.3.5 Data gaps causing 'Not Valid' STA values

Data collection over OPC-UA will stop when ReportLoq is not running. This will typically invalidate the STA if 1/3 of the FLD values are missing. For waste incinerators with 30 min. STA this means that you need to turn the DAHS back on after 10 minutes to avoid STA invalidation (NV[1]). None of ReportLoq's OPC UA options support OPC-HDA[2].

Data collection from a ReportLoq DataLogger will not cause invalidation if ReportLoq is not running. The Data Collector has 30 days of data store internally. This means that offline periods are safe for a longer period of time. Once the DAHS is back up, it will buffer up and calculate the necessary data.

5.4 Webserver

ReportLoq comes with an embedded Jetty webserver, which serves as ReportLoq's user interface. The interface is as default accessed by end-users on port 80[3] (HTTP) or port 443 (SSL).

Note: ReportLoq is not intended to run on public IP addresses as some interfaces are publicly available without login. Users who desire external access should consider ReportLoq+ Cloud which offers secure access using www.reportloq.com.

The webserver is stopped as running by default. This default setting can be changed[4], and the webserver can manually be toggled in File→Webserver

5.4.1 SSL

ReportLoq supports SSL certificates in .pfx key stores and can out-of-the-box generate self-signed certificates. Existing keystores can be added in File→Webserver→SSL certificate setup.

The "Validate certificate" function will look up "Key alias" in the keystore using the supplied password. This will return "Certificate found" if found. Testing if the certificate is working should be done by starting the webserver and browsing ReportLoq using a browser.

Note: Keystore files are not a part of the backup functionality in ReportLoq and should be backed up manually.

[1] NV = Not Valid
[2] OPC UA HDA = Historian OPC UA with data buffering over OPC
[3] Can be changed to another port in Settings → webserver.port
[4] Can be changed to autostart in Settings → webserver.autostart by setting value = true

Note: Passwords are stored in the password-protected MongoDB.

5.4.1.1 Existing certificates

The .pfx file must be placed in ReportLoq's installation folder. "Keystore password" and "Key alias" must grant access to the keystore and should point to the certificate in the keystore which should be used for the webserver.

Keytool.exe can be used for generation of PFX files. Please see:
https://docs.oracle.com/javase/9/security/java-secure-socket-extension-jsse-reference-guide.htm#JSSEC-GUID-3D26386B-BC7A-41BB-AC70-80E6CD147D6F

5.4.1.2 Test certificates

The user UI additionally has support for generation of test certificates. Doing this will generate a test-certificate which is valid for 365 days and store it in the suggested keystore. The default password is "password". Test certificates pointing to non-existing keystores will be stored in a new keystore created with the suggested password.

5.5 User administration

The technical user interface provides an easy user administration interface where new users can be spawned. The 'Default user' provides a quick way to create the first user with the username/password reportloq/reportloq. It goes without saying that this user should be changed or deleted once the system is ready to run.

The web interface has a similar structure for user creation and can be used by regular users after commissioning. The technical interface does however provide a safe "backdoor" for creating new users, if the current users are all unable to login.

5.6 Backup

ReportLoq can automate a daily backup and to upload this to an FTP server. This does however require that the server has enough resources to carry out the backup in a timely manner. Small machines can have difficulties carrying out a large backup over time.

The backup created is a "all on a bundle" backup, which can be restored from a fresh installation of ReportLoq (in the same version). Do however note that restoring a backup on another server can cause license validation errors.

Automating backup is done in the Settings dialog by configuring backup.directory and optionally by setting ftp.* if upload of the backup to FTP is desired.

5.7 Time synchronization

EN 17255 stipulates that time must be synchronized on servers running a DAHS system. This can be done using regular Windows NTP. This can however be risky if the server suddenly synchronizes time over longer periods, as data loss may occur. ReportLoq does for this reason offer an alternative NTP option. Setting AutoSyncTime in Settings to true and setting timeserver.ip to a company NTP server will sync time slowly in place. ReportLoq synchronizes time with 2 seconds per hour meaning there will not be any large gaps with missing data. ReportLoq does not set time if time drift is less than 2 seconds.

The setup can be verified in File→Console and running the command “SyncTime”.

5.7.1 Strict time synchronization

Enabling strict time synchronization will force ReportLoq to fully synchronize time at every DAHS startup. This setting will additionally set time every hour regardless of the time drift detected.

5.8 Console

It is possible to use advanced commands using the console. It recommended to backup the DAHS system if you are unsure how the console is working.

5.9 Combined views

Building combined view is a great feature for displaying values from more company lines on the same webpage. It will display only the most important information to the user, and in addition to this, allow an extended alarm list to be displayed next to the real-time values. This feature is often used on big surveillance monitors.

Configured combined views will automatically appear to users of the webserver.

5.10 Event logger

ReportLoq logs every major event in the DAHS system in the internal event log. This includes loading new configurations, major time adjustments and other significant events.

5.11 The active signal and OTNOC

The operating signals in ReportLoq are quality assured by the DAHS system before logging. The following rules apply and will affect the logged value and its status values.

Bit 0 = Within R-EOT

Reporting is active. The plant is running and limit values are active. The logged value is set to 1.

Bit 1 = Outside EOT

Reporting has stopped. The plant is shut down and reporting is irrelevant. The logged value is set to 0.

Bit 2 = Outside R-EOT (Phase 1)

The plant is being started up. Limit values are not active. Mass emissions are calculated. The logged value is set to 0.

Bit 3 = Outside R-EOT (Phase 2)

The plant is being shut down. Limit values are not active. Mass emissions are calculated. The logged value is set to 0.

Only one of the above 4 bits can be high at the time. One must be high. Bit 0 takes precedence. Bit 1 is enforced in case no bits are set high.

Bit 4 = OTNOC (Optional)

This setting is particularly interesting for waste incinerators. Bit 4 is quality assured before logging and always will be:

- Low within EOT (bit 1).
- High outside R-EOT (bit 2 or 3)

During operation (bit 0), the OTNOC signal can extend start-up and shutdown so that OTNOC can be reported during start-up in phase 2 and shutdown in phase 1. The signal can also be used within R-EOT if OTNOC occurs during runtime.

It is only while bit 0 is high that bit 4 will affect the logged value.

Bit 4 will not affect ReportLoq's calculation of STA/LTA values.

5.12 Values from DAHS to Central Control System

The following values can be made available over OPC UA or over a DataLogger using a custom I/O.

5.12.1 Stack information and sum alarms

DataLogger Name (X=Line no.)	OPC name	Type	Description
.X_SRO_REPORT_ACTIVE	REPORT_ACTIVE_VARIABLE	Boolean	Is plant active?
.X_SRO_REPORT_STOPPED	REPORT_STOPPED_VARIABLE	Boolean	Is plant stopped?
.X_SRO_PLANT_SHUT_DOWN	REPORT_SHUT_DOWN_VARIABLE	Boolean	Is plant shutting down?
.X_SRO_PLANT_STARTUP	PLANT_STARTUP_VARIABLE	Boolean	Is plant starting up?
.X_SRO_PLANT_SRO	PLANT_SRO_VARIABLE	Boolean	Is plant in SRO mode?
.X_SRO_K_T	K_T_VARIABLE	Double	Correction for temperature
.X_SRO_K_P	K_P_VARIABLE	Double	Correction for pressure
.X_SRO_K_H2O	K_H2O_VARIABLE	Double	Correction for water
.X_SRO_K_O2	K_O2_VARIABLE	Double	Correction for oxygen
.X_SRO_EXCEEDANCE_4H_HOURLV	EXCEEDANCE_4H_HOURLV_VARIABLE	Integer	Number of unacknowledged 4-Hour exceedances
.X_SRO_EXCEEDANCE_A_HOURLV	EXCEEDANCE_A_HOURLV_VARIABLE	Integer	Number of unacknowledged A-Hour exceedances
.X_SRO_EXCEEDANCE_DAYLV	EXCEEDANCE_DAYLV_VARIABLE	Integer	Number of unacknowledged Day exceedances
.X_SRO_EXCEEDANCE_HOUR_LV_MOM	EXCEEDANCE_HOUR_LV_MOM_VARIABLE	Integer	Number of unacknowledged momentary exceedances
.X_SRO_ LAST_STA_LT_STRING	LAST_STA_LT_STRING_VARIABLE	String	Local time of last calculated STA
.X_SRO_ LAST_STA_LT_SECONDS	LAST_STA_LT_SECONDS_VARIABLE	Int64	Local time of last calculated STA.
.X_SRO_ LAST_STA_UTC_STRING	LAST_STA_UTC_STRING_VARIABLE	String	UTC time of last calculated STA. Seconds since 1970-01-01.
.X_SRO_ LAST_STA_UTC_SECONDS	LAST_STA_UTC_SECONDS_VARIABLE	Int64	UTC time of last calculated STA. Seconds since 1970-01-01.
.X_SRO_NOTIFICATION_ALARM	NOTIFICATION_ALARM_VARIABLE	Integer	Number of unacknowledged active alarms
.X_SRO_NOTIFICATION_EXCEEDANCE	NOTIFICATION_EXCEEDANCE_VARIABLE	Integer	Total number of unacknowledged exceedances
.X_SRO_NOTIFICATION_HIGH_HARDWARE	NOTIFICATION_HIGH_HARDWARE_VARIABLE	Integer	Number of high hardware alarms and wirebreak
.X_SRO_NOTIFICATION_HIGH_PREDICTION	NOTIFICATION_HIGH_PREDICTION_VARIABLE	Integer	Number of high predictions
.X_SRO_ NOTIFICATION_SUM	NOTIFICATION_SUM_VARIABLE	Integer	Total number of unacknowledged Exceedances AND alarms
.X_SRO_NOTIFICATION_WARNING	NOTIFICATION_WARNING_VARIABLE	Integer	Number of unacknowledged warnings (predictions, wirebreaks, DAHS system alerts and Auto QAL3 requests)
.X_SRO_ PREDICTED_STA	PREDICTED_STA_VARIABLE	Double	Predicted STA exceedances
.X_SRO_ PREDICTED	PREDICTED_VARIABLE	Double	Predicted LTA exceedances

5.12.2 Component based stack calculations

DataLogger Name (X=Line no.) [Array]	OPC name [XX] is tag number	Type	Description
Last Complete STA calculations			
.X_SRO_LAST_STA_EM_VSTA	**VAL[XX]_LAST_STA_EM_VSTA**	Double	Last complete VSTA value
.X_SRO_LAST_STA_EM_VT	**VAL[XX]_LAST_STA_EM_VT**	String	Last complete VSTA validation text
.X_SRO_LAST_STA_EM_RELEVANT	**VAL[XX]_LAST_STA_EM_RELEVANT**	Boolean	Was VSTA used for comparing emission limit values (Not EN or NV)?
.X_SRO_LAST_STA_GR_KG	**VAL[XX]_LAST_STA_GR_KG**	Double	Last complete calculated 'kg'
.X_SRO_LAST_STA_GR_SSTA	**VAL[XX]_LAST_STA_GR_SSTA**	Double	Last complete SSTA value
.X_SRO_LAST_STA_GR_STA	**VAL[XX]_LAST_STA_GR_STA**	Double	Last complete STA value
.X_SRO_LAST_STA_GR_VT	**VAL[XX]_LAST_STA_GR_VT**	String	Last complete SSTA validation text
.X_SRO_LAST_STA_GR_RELEVANT	**VAL[XX]_LAST_STA_GR_RELEVANT**	Boolean	Was SSTA used for mass emission calculation (Not EN or NV)?
FLD based calculations			
.X_SRO_RAW	**VAL[XX]_RAW**	Double	FLD (RAW)
.X_SRO_Q3	**VAL[XX]_Q3**	Double	FLD (QAL3 value)
.X_SRO_Q2	**VAL[XX]_Q2**	Double	FLD (QAL2 value)
.X_SRO_C	**VAL[XX]_C**	Double	FLD (Corrected)
.X_SRO_Q	**VAL[XX]_Q**	Double	FLD (Validated)
.X_SRO_CONF	**VAL[XX]_CONF**	Double	FLD-based confidence interval
.X_SRO_CS	**VAL[XX]_CS**	String	Correction factor string
.X_SRO_CM	**VAL[XX]_CM**	Boolean	Is FLD the chosen measurement?
.X_SRO_OMR	**VAL[XX]_OMR**	Boolean	Is FLD outside the measuring range?
.X_SRO_STATUS	**VAL[XX]_STATUS**	String	FLD Validation code
Live STA/LTA calculations			
.X_SRO_AVG6	**VAL[XX]_AVG6**	Double	6-minute VSTA
.X_SRO_AVG10	**VAL[XX]_AVG10**	Double	10-minute VSTA
.X_SRO_AVG[STA_MIN]	**VAL[XX]_AVG[STA_MIN]**[5]	Double	STA-minute VSTA
.X_SRO_AVG[STA_MIN]_EM_VT	**VAL[XX]_AVG[STA_MIN]_EM_VT**	String	VSTA validation text
.X_SRO_AVG[STA_MIN]_EM_RELEVANT	**VAL[XX]_AVG[STA_MIN]_EM_RELEVANT**	Integer	Is VSTA used for reporting (Not EN or NV)?
.X_SRO_AVG[STA_MIN]_GR_VT	**VAL[XX]_AVG[STA_MIN]_GR_VT**	String	SSTA validation text
.X_SRO_AVG[STA_MIN]_GR_RELEVANT	**VAL[XX]_AVG[STA_MIN]_GR_RELEVANT**	Boolean	Is SSTA used for reporting (Not EN or NV)?
.X_SRO_AVG[STA_MIN]_PREDICTED	**VAL[XX]_AVG[STA_MIN]_PREDICTED**	Double	Predicted VSTA
.X_SRO_AVG[STA_MIN]_PREDICTED _EXCEEDANCE	**VAL[XX]_AVG[STA_MIN]_PREDICTED _EXCEEDANCE**	Boolean	Has predicted VSTA exceedance?
.X_SRO_AVG_DAY1	**VAL[XX]_AVG_DAY1**	Double	Daily VLTA
.X_SRO_AVG_DAY1_EM_VT	**VAL[XX]_AVG_DAY1_EM_VT**	String	VLTA validation text
.X_SRO_AVG_DAY1_EM_RELEVANT	**VAL[XX]_AVG_DAY1_EM_RELEVANT**	Integer	Is VLTA used for reporting

[5] Depends on the STA minutes used for the installation. Is usually AVG[XX]_AVG60 for combustion or AVG[XX]_AVG30 for waste incinerators.

			(Not EN or NV)? 1 = Yes
.X_SRO_AVG_DAY1_PREDICTED	**VAL[XX]_AVG_DAY1_PREDICTED**	Double	Predicted daily VLTA
.X_SRO_AVG_DAY1_PREDICTED _EXCEEDANCE	**VAL[XX]_AVG_DAY1_PREDICTED _EXCEEDANCE**	Boolean	Has predicted VLTA exceedance?
.X_SRO_AVG_DAY1_RECOVERY	**VAL[XX]_AVG_DAY1_RECOVERY**	Double	Daily set-point
.X_SRO_AVG[STA_MIN]_A_HOUR_LV_MIN	**VAL[XX]_AVG[STA_MIN]_A_HOUR_LV_MIN**	Double	The current minimum hourly A-LV (if any) A = strict LV
.X_SRO_AVG[STA_MIN]_A_HOUR_LV_MAX	**VAL[XX]_AVG[STA_MIN]_A_HOUR_LV_MAX**	Double	The current maximum hourly A-LV (if any) A = strict LV
.X_SRO_AVG[STA_MIN] _A_HOUR_MOM_LV_MIN	**VAL[XX]_AVG[STA_MIN] _A_HOUR_MOM_LV_MIN**	Double	The current minimum hourly A-MOM-LV (if any) A-MOM = Momentary LV
.X_SRO_AVG[STA_MIN] _A_HOUR_MOM_LV_MAX	**VAL[XX]_AVG[STA_MIN] _A_HOUR_MOM_LV_MAX**	Double	The current maximum hourly A-MOM-LV (if any) A-MOM = Momentary LV
.X_SRO_AVG[STA_MIN]_B_HOUR_LV_MIN	**VAL[XX]_AVG[STA_MIN]_B_HOUR_LV_MIN**	Double	The current minimum hourly B-LV (if any) B = fraction LV
.X_SRO_AVG[STA_MIN]_B_HOUR_LV_MAX	**VAL[XX]_AVG[STA_MIN]_B_HOUR_LV_MAX**	Double	The current maximum hourly B-LV (if any) B = fraction LV
.X_SRO_AVG1440_A_DAY_LV_MIN	**VAL[XX]_AVG1440_A_DAY_LV_MIN**	Double	The current minimum daily A-LV (if any) A = strict LV
.X_SRO_AVG1440_A_DAY_LV_MAX	**VAL[XX]_AVG1440_A_DAY_LV_MAX**	Double	The current maximum daily A-LV (if any) A = strict LV
.X_SRO_AVG1440_A_DAY_97_LV_MIN	**VAL[XX]_AVG1440_A_DAY_97_LV_MIN**	Double	The current minimum daily 97% LV (if any) 97% = fraction LV
.X_SRO_AVG1440_A_DAY_97_LV_MAX	**VAL[XX]_AVG1440_A_DAY_97_LV_MAX**	Double	The current maximum daily 97% LV (if any) 97% = fraction LV
.X_SRO_AVG_MONTH	**VAL[XX]_AVG_MONTH**	Double	Monthly VLTA
.X_SRO_AVG_ MONTH_EM_VT	**VAL[XX]_AVG_MONTH_EM_VT**	String	VLTA validation text
.X_SRO_AVG_ MONTH_EM_RELEVANT	**VAL[XX]_AVG_MONTH_EM_RELEVANT**	Boolean	Is VLTA used for reporting (Not EN or NV)?
.X_SRO_AVG_ MONTH_PREDICTED	**VAL[XX]_AVG_MONTH_PREDICTED**	Double	Predicted monthly VLTA
.X_SRO_AVG_ MONTH_PREDICTED _EXCEEDANCE	**VAL[XX]_AVG_MONTH_PREDICTED _EXCEEDANCE**	Integer	Has predicted VLTA exceedance? 1 = Yes
.X_SRO_AVG_A_MONTH_LV_MIN	**VAL[XX]_AVG_A_MONTH_LV_MIN**	Double	The current minimum monthly A-LV (if any) A = strict LV
.X_SRO_AVG_A_MONTH_LV_MAX	**VAL[XX]_AVG_A_MONTH_LV_MAX**	Double	The current maximum monthly A-LV (if any) A = strict LV
.X_SRO_AVG_YEAR	**VAL[XX]_AVG_YEAR**	Double	Yearly VLTA
.X_SRO_AVG_YEAR_EM_VT	**VAL[XX]_AVG_YEAR_EM_VT**	String	VLTA validation text
.X_SRO_AVG_YEAR_EM_RELEVANT	**VAL[XX]_AVG_YEAR_EM_RELEVANT**	Boolean	Is VLTA used for reporting (Not EN or NV)?

.X_SRO_AVG_YEAR_PREDICTED	**VAL[XX]_AVG_YEAR_PREDICTED**	Double	Predicted yearly VLTA
.X_SRO_AVG_YEAR_PREDICTED _EXCEEDANCE	**VAL[XX]_AVG_YEAR_PREDICTED _EXCEEDANCE**	Boolean	Has predicted VLTA exceedance?
.X_SRO_AVG_A_YEAR_LV_MIN	**VAL[XX]_AVG_A_YEAR_LV_MIN**	Double	The current minimum yearly A-LV (if any) A = strict LV
.X_SRO_AVG_A_YEAR_LV_MAX	**VAL[XX]_AVG_A_YEAR_LV_MAX**	Double	The current maximum yearly A-LV (if any) A = strict LV

6 Contact information

Olicem A/S
Majsmarken 1
DK-9500 Hobro · Denmark
Support e-mail: support@olicem.com
VAT-ID: DK 3995 8708
www.olicem.com

7 Document revision

Rev.	Date	Author	Description
2.3	1/11-2020	KVIN	Created
2.3.1	11/12-2020	KVIN	Updated
2.4	1/1-2021	KVIN	Updated
2.5	1/7-2023	KVIN	Updated
2.6	8/12-2023	KVIN	Updated
2.7	16/1-2024	KVIN	Updated
2.8	24/5-2024	KVIN	Updated
2.9	8/9-2024	KVIN	Updated with OTNOC
2.10	20/11-2024	KVIN	Updated with OMR
2.11	20/1-2025	KVIN	5.12.1 Updated with RELEVANT
2.12	12/5-2025	KVIN	Updated regarding embedded OPC-UA