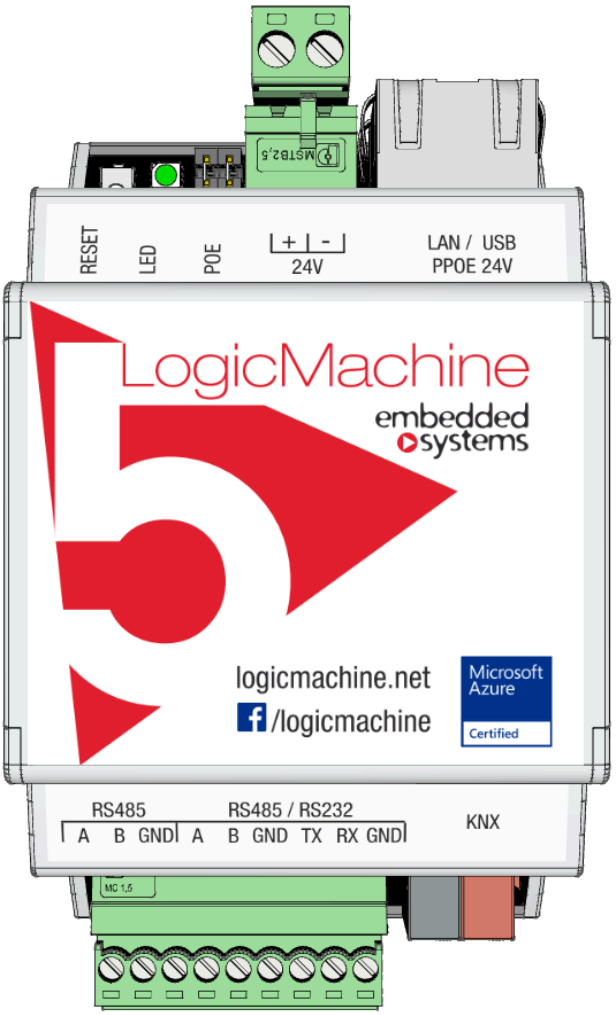
Embedded Systems SIA
47. Katolu str., Riga, LV 1003, LATVIA
Phone: +371 67648888, e-mail: sales@openrb.com

LogicMachine5 Lite

Product Manual



December, 2025

Applicable firmware version: 2025.12

Technical support:
support@openrb.com
forum.logicmachine.net

Copyright

Notice

Embedded Systems SIA reserves the right to modify the information contained herein as necessary. Embedded Systems SIA assumes no responsibility for any errors which may appear in this document. Information in this document is provided solely to enable system and software implementers to use LogicMachine products.

Trademarks

LogicMachine is a trademark of Embedded Systems SIA. All other names and trademarks are the property of their respective owners and are hereby acknowledged.

Introduction

LogicMachine (LM) is the easiest way to implement complex logic in KNX, Modbus, BACnet, ZigBee, EnOcean and other networks. LM enables efficient building automation process customization, providing virtually unlimited flexibility to the end users in a cost-effective way.

LM5 is an embedded platform with an integrated Ethernet, USB, KNX/TP and RS-485/RS-232 serial interfaces. LM can be used as a cross-standard gateway (Modbus, BACnet/IP), logic engine, visualization platform, KNX/IP Router. It can be integrated with various cloud/web services and 3rd party devices. Scripts (logic engine) allows LM to simultaneously act as a thermostat, security panel, lighting controller, etc. Additional applications can be installed to further extend the device functionality.

Technical support

Any faulty devices should be returned to Embedded Systems.

For any other technical questions use our forum at forum.logicmachine.net

Firmware updates are available at openrb.com/firmwares/

Caution
Security advice

Risk of damage to property and personal injury due to wrong installation.

Electrical installation can only be ensured if the person can prove knowledge in the following areas:

• Installation of networks

• Mounting electric cables

• Installation of KNX networks

These skills are possessed by certified specialists who are trained in electrical installation technology. If these requirements are not met, you are personally liable for any damage to property or personal injury.

Electrical connection

The devices are constructed for the operation of protective low voltage (SELV). Grounding of devices is not needed. When switching the power supply on or off, power surges must be avoided.

Table of contents

1. Security recommendations

It is recommended to install LM on a secure network without public access. Where possible separate network segments either physically, by using VLANs or firewalls.

If communication between several LMs is required in one building - provide a KNX Backbone key and set the "Enable only secure communication" option.

Disable KNX/IP features after the project has been fully commissioned if KNX/IP Routing is not needed.

LM cloud can be used for remote control. For remote commissioning and diagnostics either ZeroTier or OpenVPN should be used. Do not use port forwarding.

Use HTTPS instead of HTTP where possible. Install the *SSL certificate* application from the LM app store to obtain a valid certificate for your LM. Using HTTP over ZeroTier or OpenVPN is allowed as the tunneling connection is already encrypted.

Perform LM project backups periodically and store them in a safe place.

In case you find cyber security incidents or vulnerabilities, please contact us through this page: logicmachine.net/contact-us

Embedded Systems SIA cannot be held responsible for performance problems and incompatibilities caused by applications, services or devices from third-party providers. Failure to follow these instructions can result in equipment damage.

2. Quick startup guide

2.1. Connection

- Mount the device on the DIN rail
- Connect the KNX/TP bus cable
- Connect 24V power supply to the device (red pole to *24V+*, grey pole to *GND*)
- Connect LM to the local network using an Ethernet cable

2.2. Default network and access configuration

Login name	admin
Password	admin
IP address	192.168.0.10
Network mask	255.255.255.0

Make sure that your PC is on the same sub-network as LM or the connection will not be possible.

The device can be accessed by opening a web browser (Chrome, Firefox, Safari are supported) and entering IP of the device http://IP

Secure access to the device is available via https://IP

LM comes with a self-signed certificate for which the browser will display a warning. Install the *SSL certificate* application from the LM app store to obtain a valid certificate for your LM or accept the warning.

2.3. Discover LogicMachine IP address

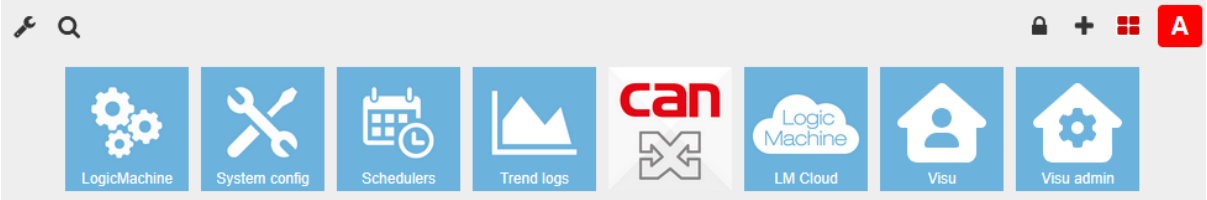
LM has a built-in mDNS/zeroconf support. *LM Home* application for Android and iOS can be used to discover LM devices on the network.

2.4. Firmware upgrade and update installation

See *System → Upgrade firmware* and *Utilities → Install updates*.

3. Graphical user interface

After a successful login the main page appears:

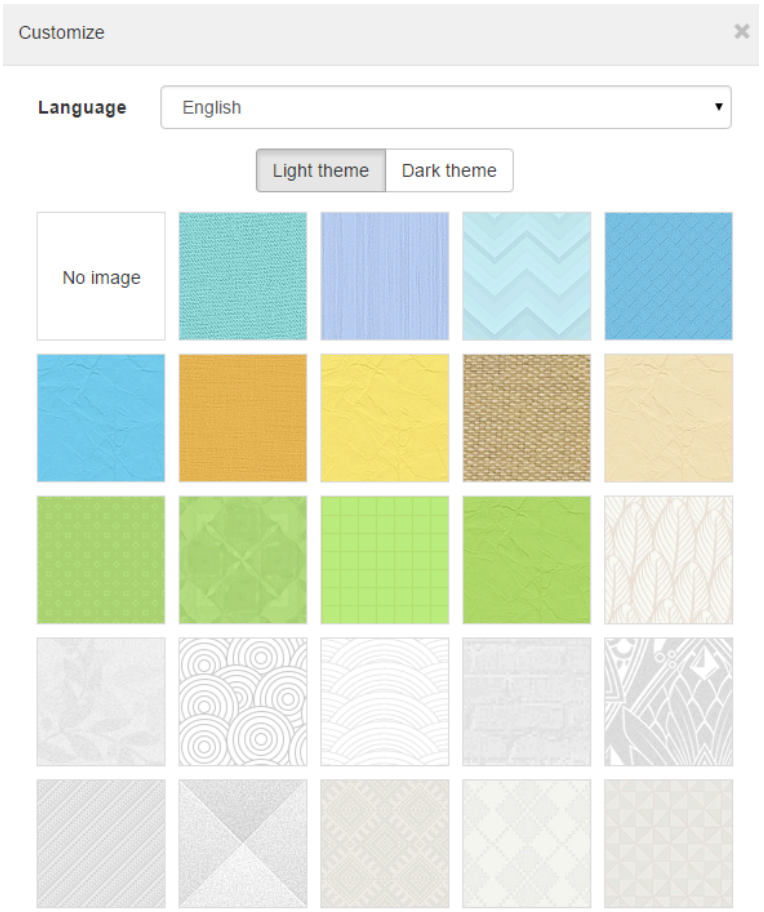


- *LogicMachine* - objects and object logs, scripts, schedulers, trends, visualization editor, user access, alerts and error logs
- *System config* - network, KNX and other built-in service configuration, package management and firmware upgrade, general system status
- *Visu* - graphical visualization (user mode)
- *Visu admin* - graphical visualization (editor mode)
- *Schedulers* - user-defined schedulers
- *Trend logs* - chart view for time-series data

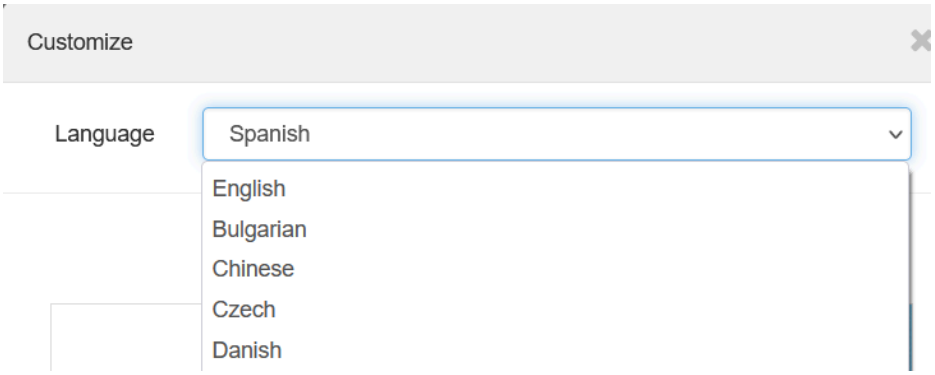
The main screen of the interface is a list of installed applications. It is possible to change the application order, hide selected applications. Each user can customize the background, light/dark theme and interface language. The admin user can install and update applications.

3.1. Customizing the background / Language

Click *Customize*  to change the interface language, color theme and the background image.



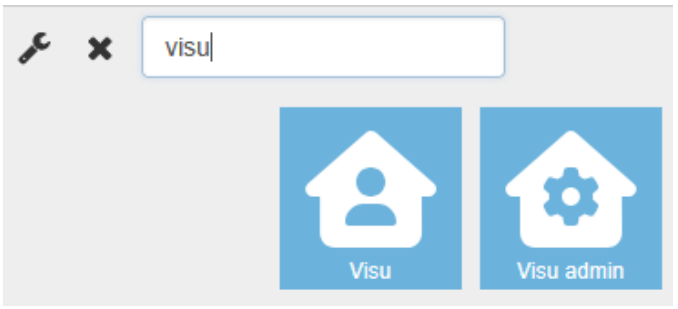
The interface language can be chosen by clicking the *Language* drop-down menu.



The background image is applied automatically when chosen.

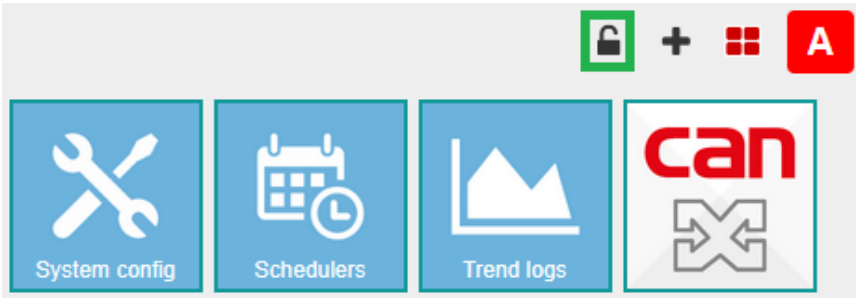
3.2. Search function

Click *Search* to open the search box to find applications containing the given search phrase.

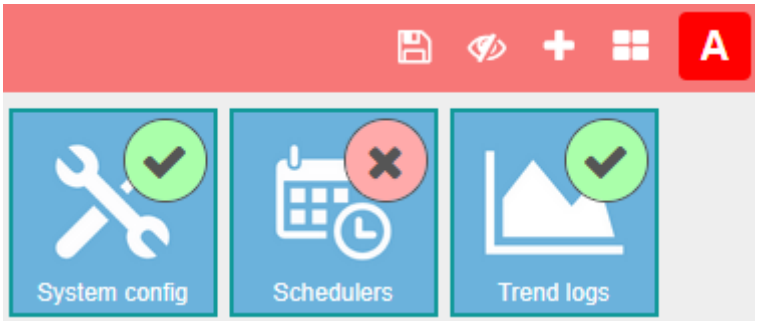


3.3. Unlock the application list

Click *Unlock* to change the application order via drag & drop.



To hide certain applications for users click *Grid* and then click *Visibility*.

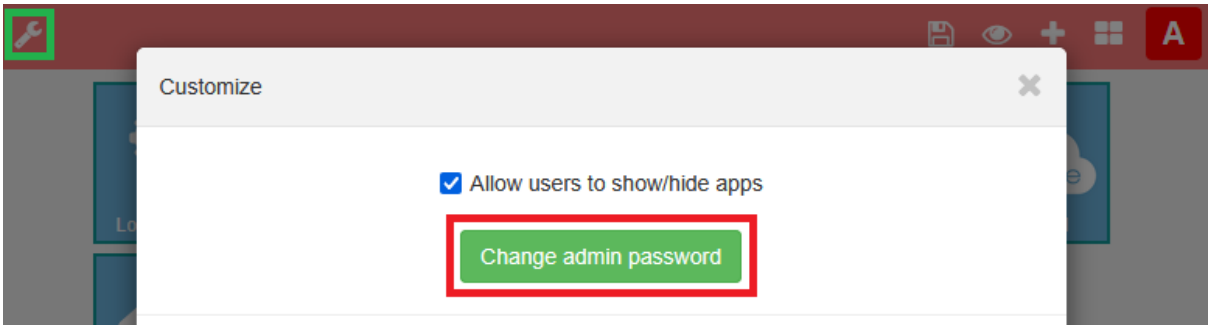


After selecting which apps will be visible to users click *Save*.

3.4. Admin mode

3.4.1. Admin mode settings

Click *Grid* then *Customize* and click *Change admin password.*



Allow users to show/hide apps enables/disables the possibility for non-admin users to to show or hide apps themselves.

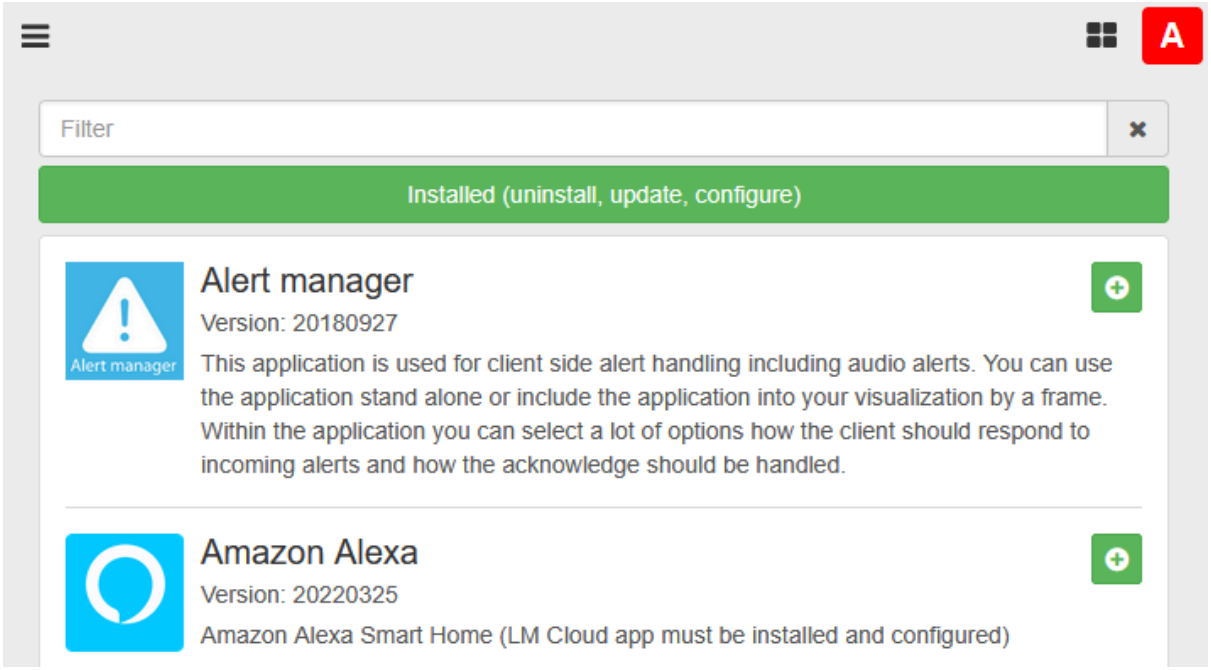
3.4.2. Change default page view for users

While in admin mode it is possible to change the default view for all users - the background image, hide/unhide/sort apps the same way as described in 3.3. Once the all the necessary changes have been made click *Save* .

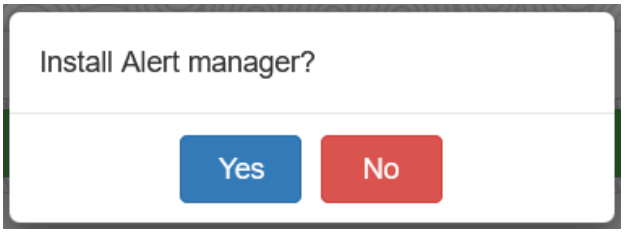
3.4.3. Add or remove applications

Click *Plus*  to enter the application administration page. If an error message appears, provide valid DNS settings as described in *System Configuration → Network → Interfaces*.

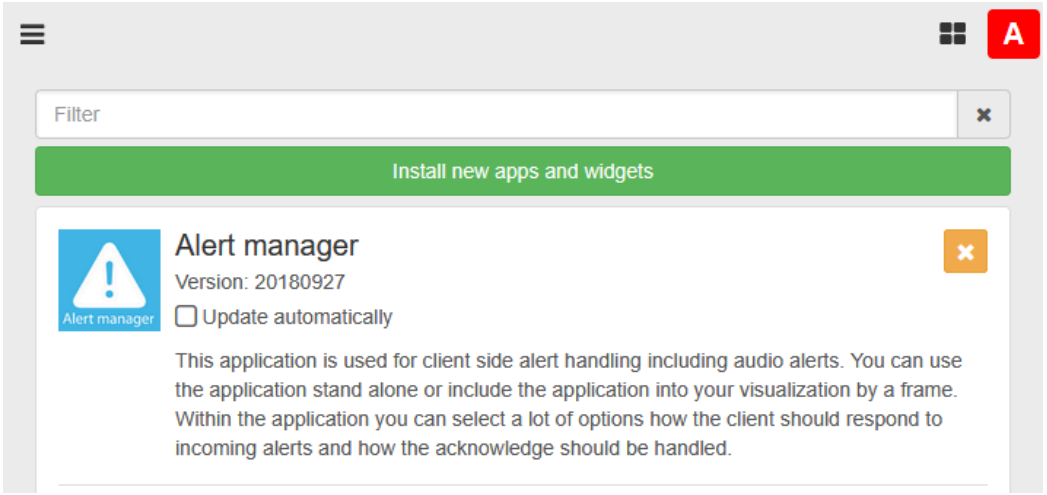
The list of available applications is displayed.



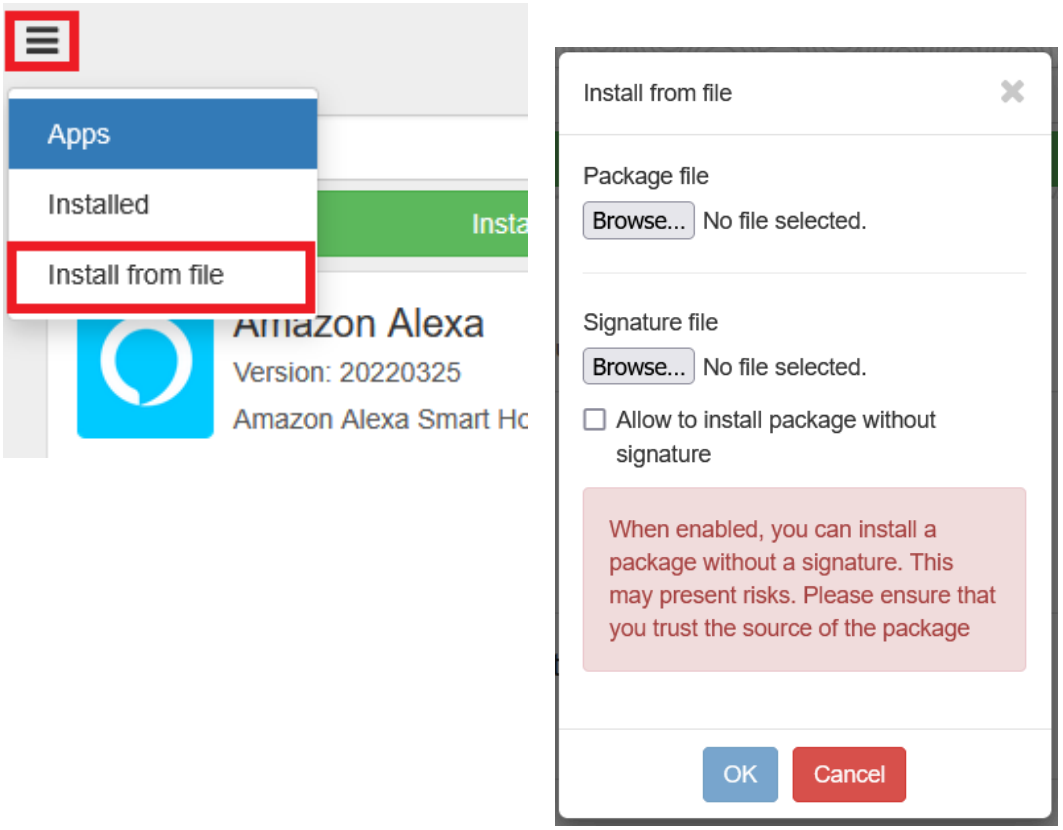
Click *Install*  to install a given application into LM.



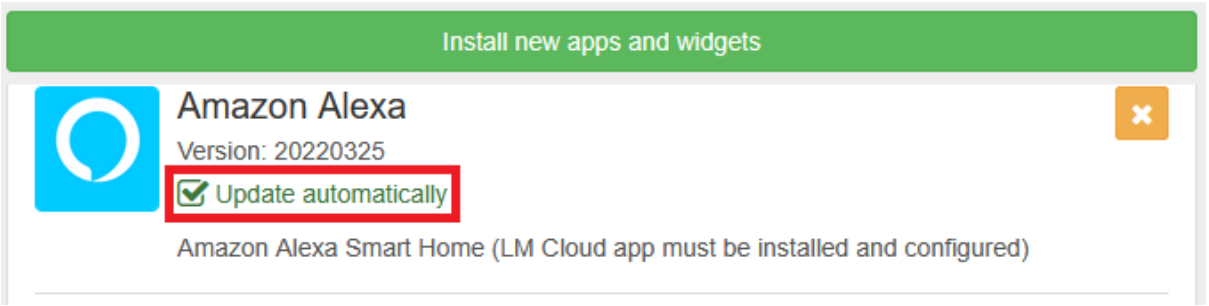
Click *Installed* to view a list of currently installed applications on LM.



Applications can be installed manually by providing an appropriate application package file.



Mark option “Update automatically” for automatic updates of the selected application.



Click *Update* to install a newer version of a selected application.

Click *Grid* to return to the main page.

3.4.4. Exit admin mode

Click A to logout.

3.5. Application development

Visit our forum: forum.logicmachine.net

4. LogicMachine configuration

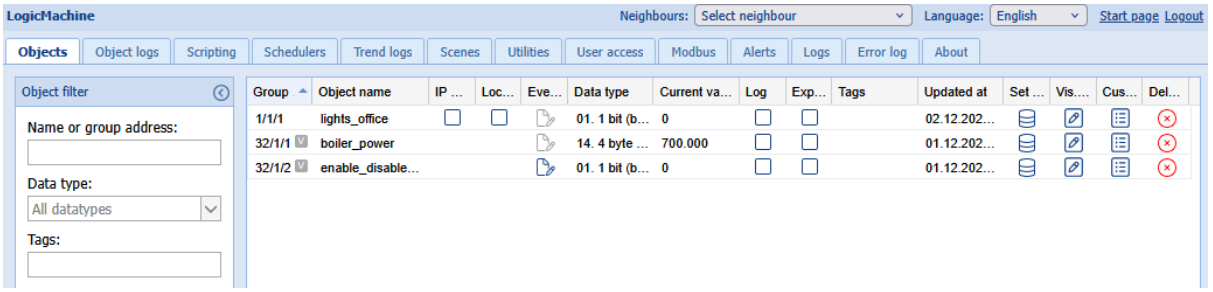
This is the main page of the LogicMachine configuration and management. It consists of the following tabs:

- *Objects* - object management
- *Object logs* - object logs
- *Scripting* - scripting repository management
- *Schedulers* - user scheduler managements
- *Trend logs* - time/series value log managements
- *Scenes* - scenes for object control
- *Utilities* - utilities including import from ETS, reset, backup, restore
- *User access* - user management and access logs
- *Modbus* - Modbus mapper
- *Alerts* - user and system alert messages
- *Logs* - script logs
- *Error log* - script and system error messages

4.1. Objects

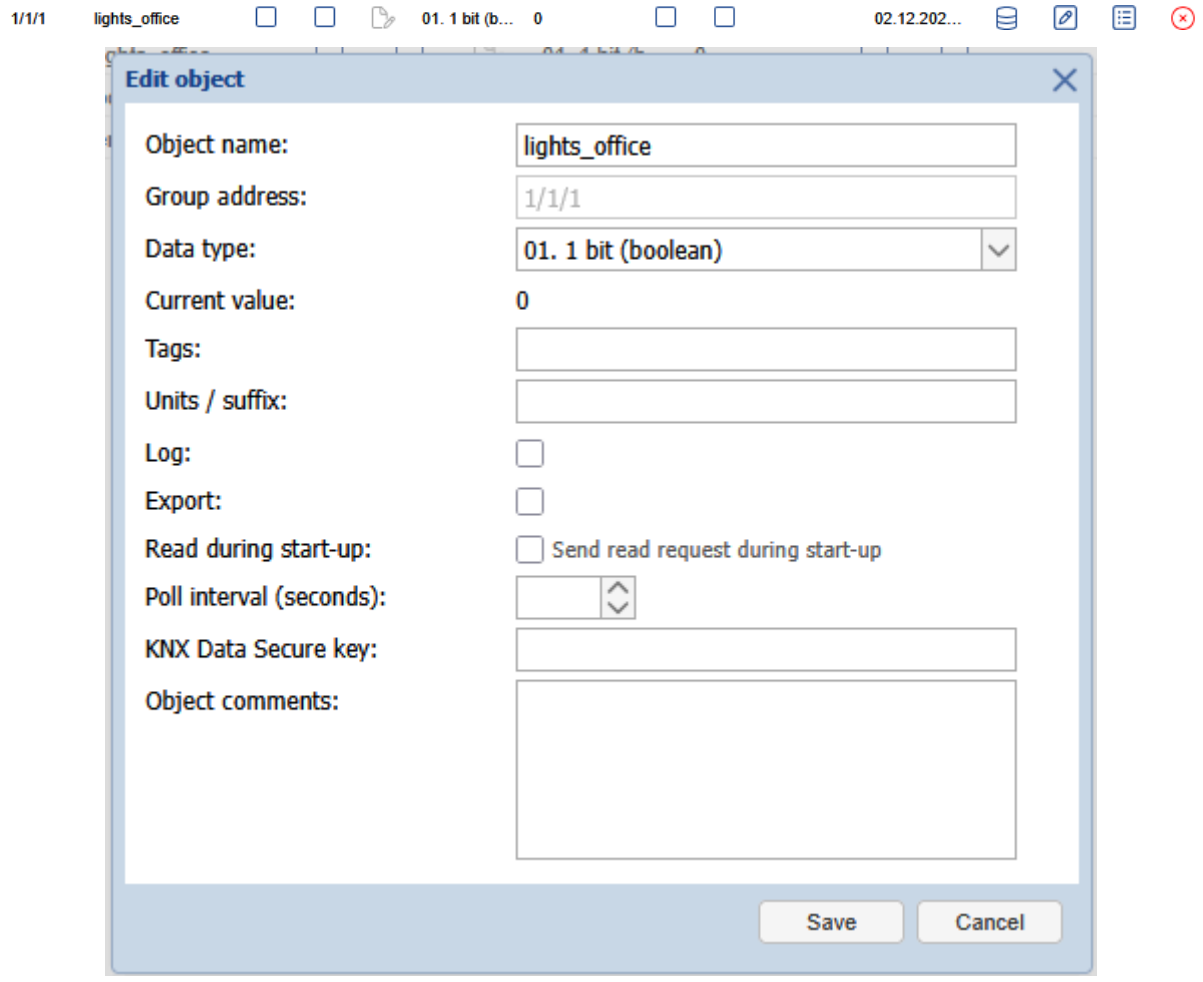
Objects can be added to this list in several ways:

- Manually by clicking *Add new object*
- New group addresses seen on the bus are added automatically (if *Bus sniffer* enabled in *Utilities → General configuration*)
- Importing KNXPROJ in *Utilities*



4.1.1. Object parameters

Click the object name or group address to edit object parameters.

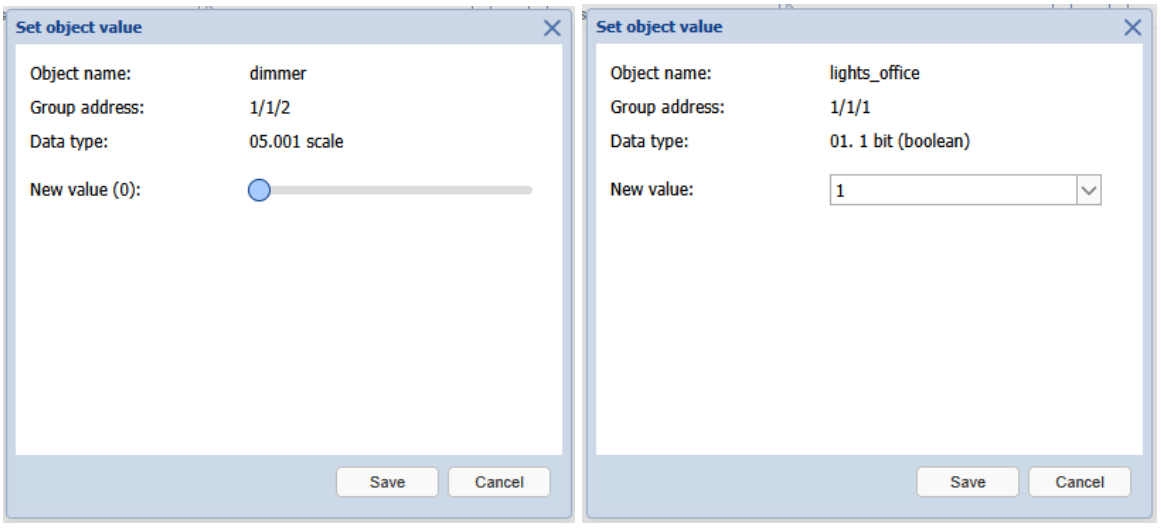


- *Object name* - name of the object
- *Group address* - group address of the object (cannot be changed once object is created)
- *Data type* - KNX data type of the object
  Note: in some cases objects added via bus sniffer or ESF import might have incorrect data type
- *Current value* - current value of the object
- *Tags* - allows grouping several objects via common tags which then can be used in scripting
- *Units / suffix* - text value that appears after the object value. Some data types have units by default (%, °C etc.)
- *Log* - enable logging for this object. Logs will appear in the *Objects logs* tab
- *Export* - this will allow object access from BACnet/IP (if enabled). Can also be used to limit objects that can be accessed via Remote services (configurable)
- *Read during start-up* - send a read request to this object when the system starts.
- *Poll interval (seconds)* - periodically send read requests to this KNX object
- *KNX Data Secure key* - KNX Secure key of the object for KNX communication encryption (32 hexadecimal characters)
- *Object comments* - used comments about this object

The object list can be sorted by one of the following parameters - Name, Group address, Data type, Current value, Tags, Comments.

4.1.2. Change the object value

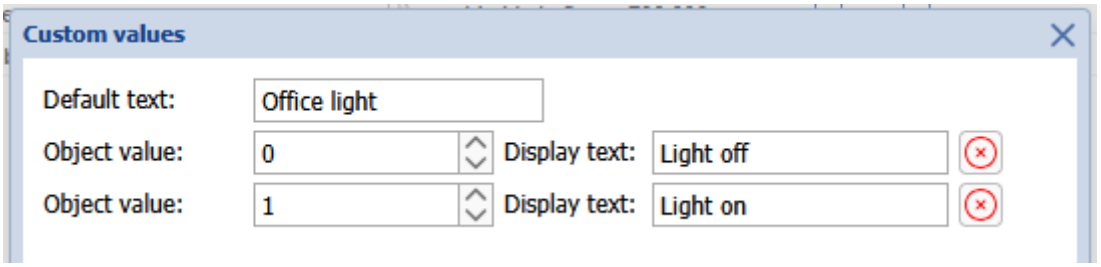
Click  to change the object value. Control elements depend on the object data type and visualization parameters.



4.1.3. Custom values

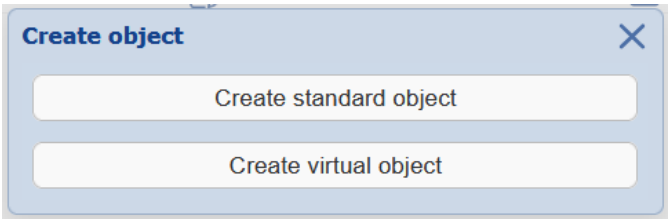
Click to map textual values to certain numerical object values. When the *Default text* is set it will be shown when no matching object value has been found. Otherwise raw object value is displayed.

Custom values are available only for Boolean and integer data types. For Boolean data type use *0* for *off/false* and *1* for *on/true*.



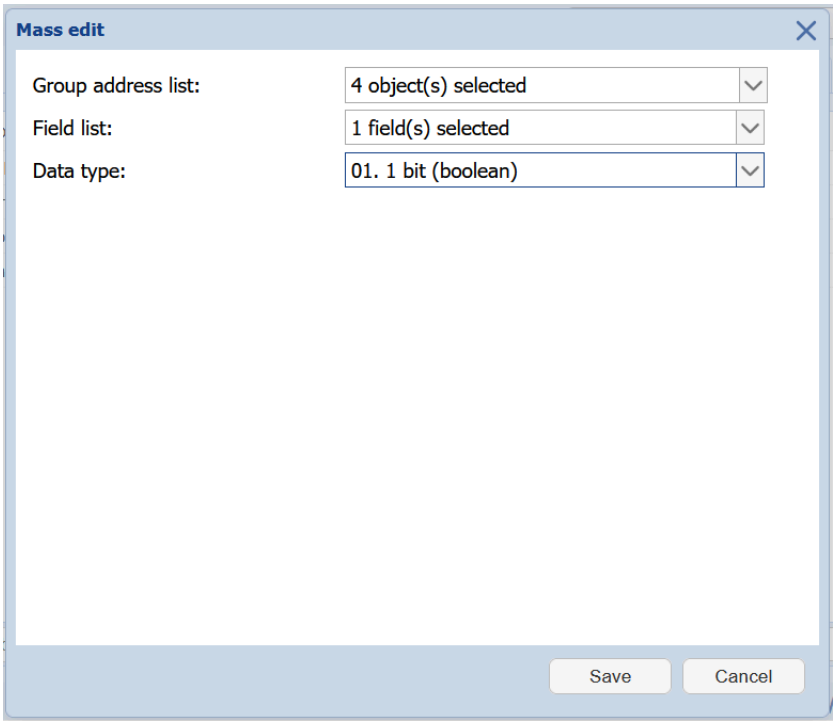
4.1.4. Object control bar

- *Add new object* - manually add new object to the list



  Virtual objects cannot appear on KNX/TP and KNX/IP, but can be accessed via BACnet and Remote services. Use virtual objects for values that are internal to the LM.

- *Auto update enabled* - whether the object list is updated automatically or not when object values are changed
- *Clear* - clear the list of group addresses
- *Mass edit* - edit certain parameters of multiple objects at once:
  - object properties
  - visualization parameters
  - custom values

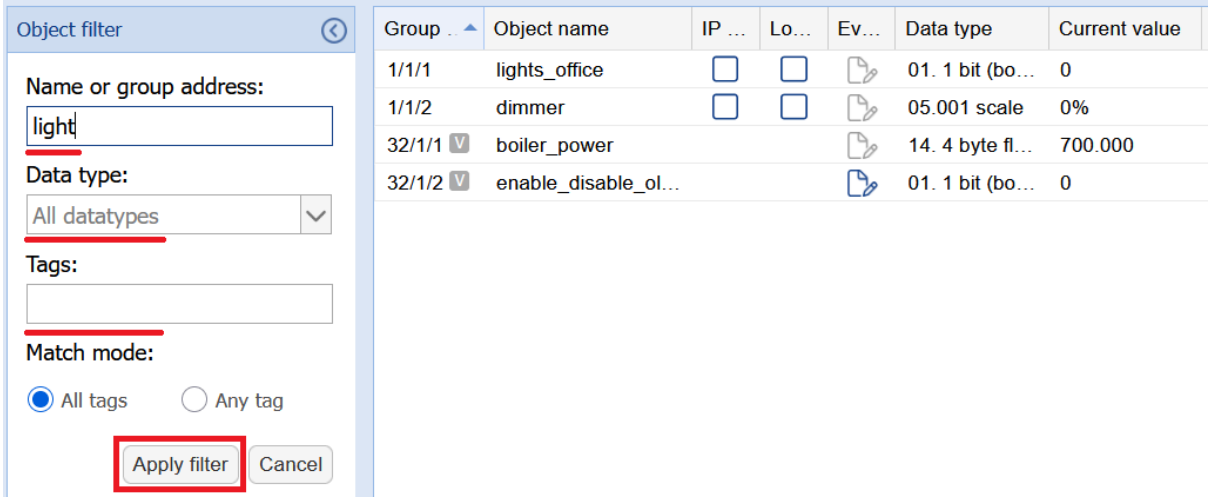


- *Mass delete* - delete all objects that either have no name set, no data type set or all objects matching the current filter

4.1.5. Object filter

Object list can be filtered by name/group address, data type and tags.

Wildcard(for example 1/1/*) can be used for filtering multiple groups addresses. Several filters can be specified, separated by comma.



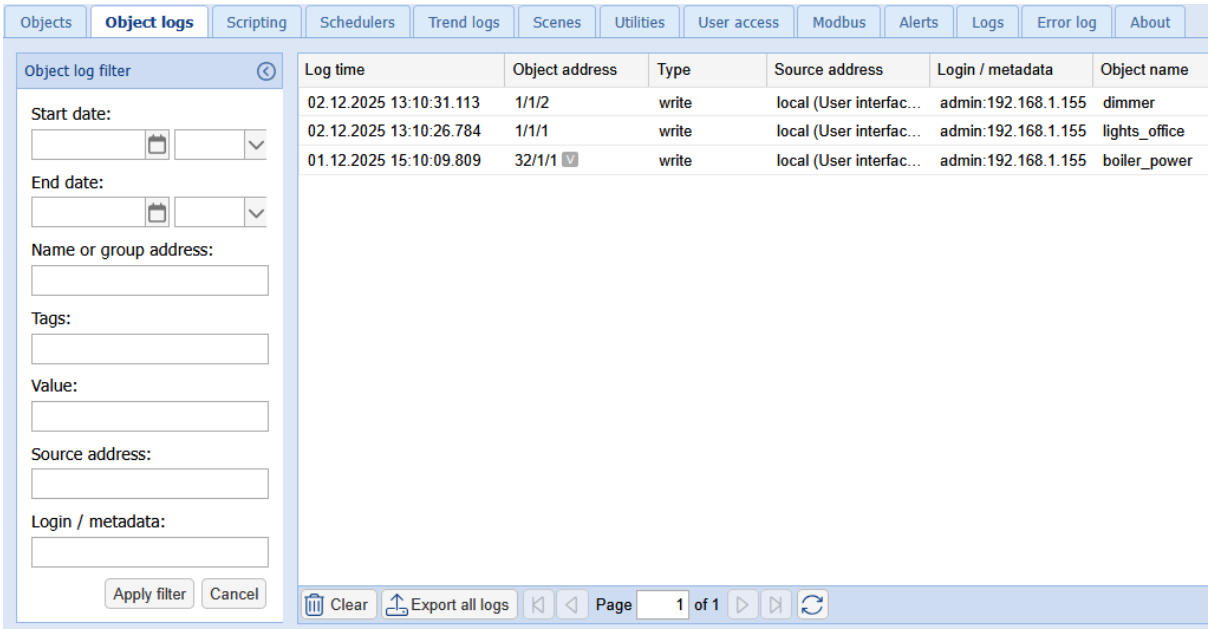
Match mode:

All tags - represents *AND* function when all tags must match

Any tag - represents *OR* function when at least one of tags must match

4.2. Object logs

Telegrams from objects that have the *Log* property enabled are available in the *Object logs* tab.



Logs can be filtered by the following criteria:

- *Start date* - start date and time
- *End date* - end date and time
- *Name or group address* - specific name or group address of the object (* wildcard allowed)
- *Tags* - object tags
- *Value* - object value
- *Source address* - source address when the telegram comes from KNX/TP or KNX/IP, *local* otherwise
- *Login / metadata* - additional telegram information. For example: login and IP address if the value has been changed by a local or a cloud user

Click *Clear* to remove all object logs.

Object log size can be changed in *LogicMachine → Utilities → General Configuration*

4.3. Scripting

The Lua programming language is used for scripting (LuaJIT, compatible with Lua 5.1). Most of the Lua language aspects are covered in the first edition of "Programming in Lua" which is freely available at www.lua.org/pil/

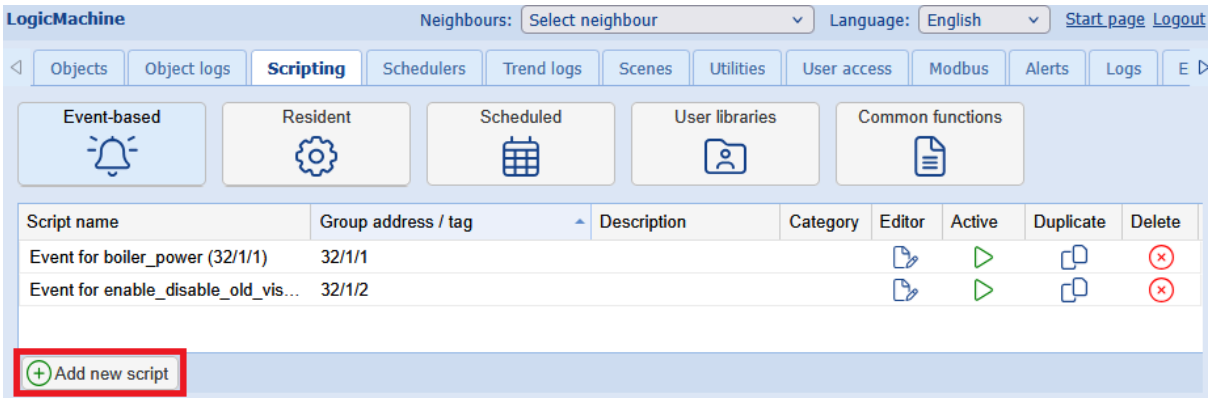
Note! the latest Lua reference manual for LogicMachine can be found at kb.logicmachine.net/libraries/lua/

Scripts types:

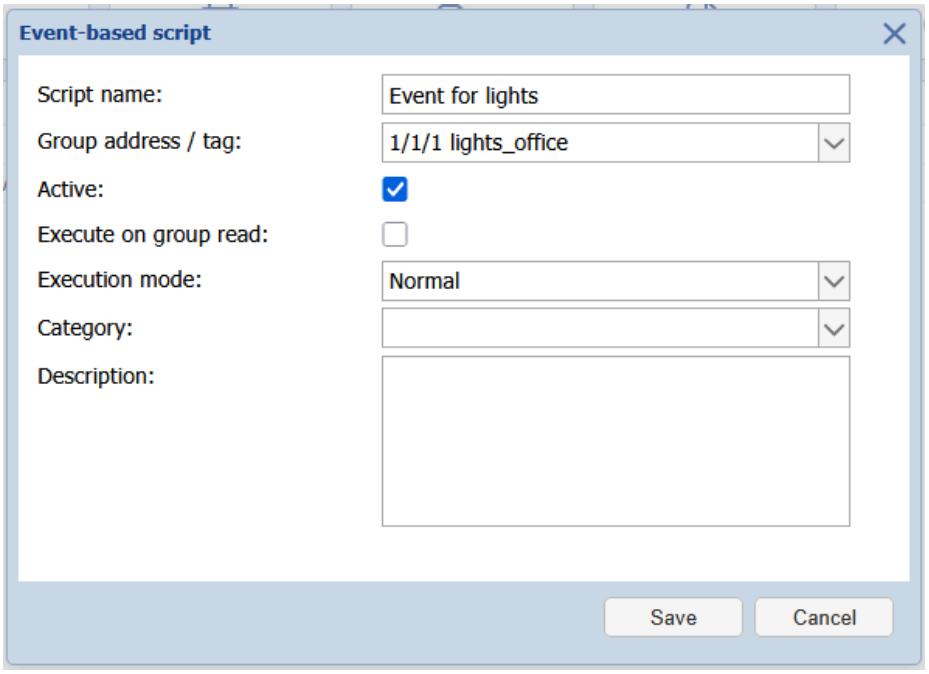
- *Event-based* - executed when a group event occurs (read/write/response)
- *Resident* - function executed in an infinite loop with a defined sleep time between each iteration
- *Scheduled* - executed at a defined time and date
- *User libraries* - custom function libraries that are used in other scripts
- *Common functions* - common functions that are used by other scripts
- *Start-up (init) script* - executed when the system starts

4.3.1. Adding a new script

Click *Add new script* at the bottom of the *Event-based*, *Resident* or *Scheduled*



4.3.1.1. Event-based



- *Script name* - name of the script
- *Group address / Tag* - specific group address or a tag name which triggers the script
- *Active* - whether the script is active (green circle) or disabled (red circle)
- *Execute on group read* - whether the script is executed when a group read telegram is received. By default scripts are triggered by group write/response telegrams
- *Execution mode* - there are three possibilities of event script execution behaviours:
    - *Normal* - if the event is triggered multiple times in quick succession, then script will execute for each event trigger
    - *First instance only* - a new script instance is not executed if a script instance is already running when a new event is triggered
    - *Last instance only* - an existing script instance is stopped if it is running and a new script instance is executed when a new event is triggered
- *Category* - new or existing category for the script. This allows filtering scripts by category, it is also shown in the *Tools* → *Print* script listings page
- *Description* - description of the script

Event-based scripting can be used to implement custom logic for group address or tag events. User-defined function is executed when a “group write/response” or “group read” (if enabled) event occurs for a given group address. Event information is stored in the global *event* variable. Variable contents:

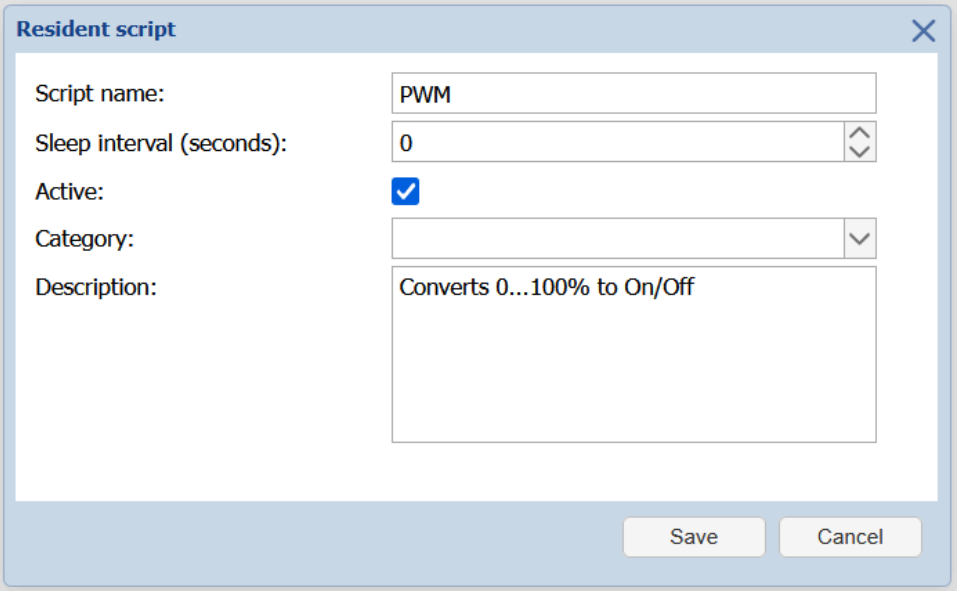
- *dstraw* (integer) - raw destination group address
- *srcraw* (integer) - raw source individual address

- *dst* (string) - decoded destination group address (for example: 1/1/4)
- *src* (string) - decoded source individual address (for example: 1.1.2)
- *type* (string) - type of the event, either 'groupwrite', 'groupread', 'groupresponse'
- *dataraw* (string) - event data as a binary string
- *datahex* (string) - event data as a hex-encoded string

To get the event value use the following command:

```
value = event.getvalue()
```

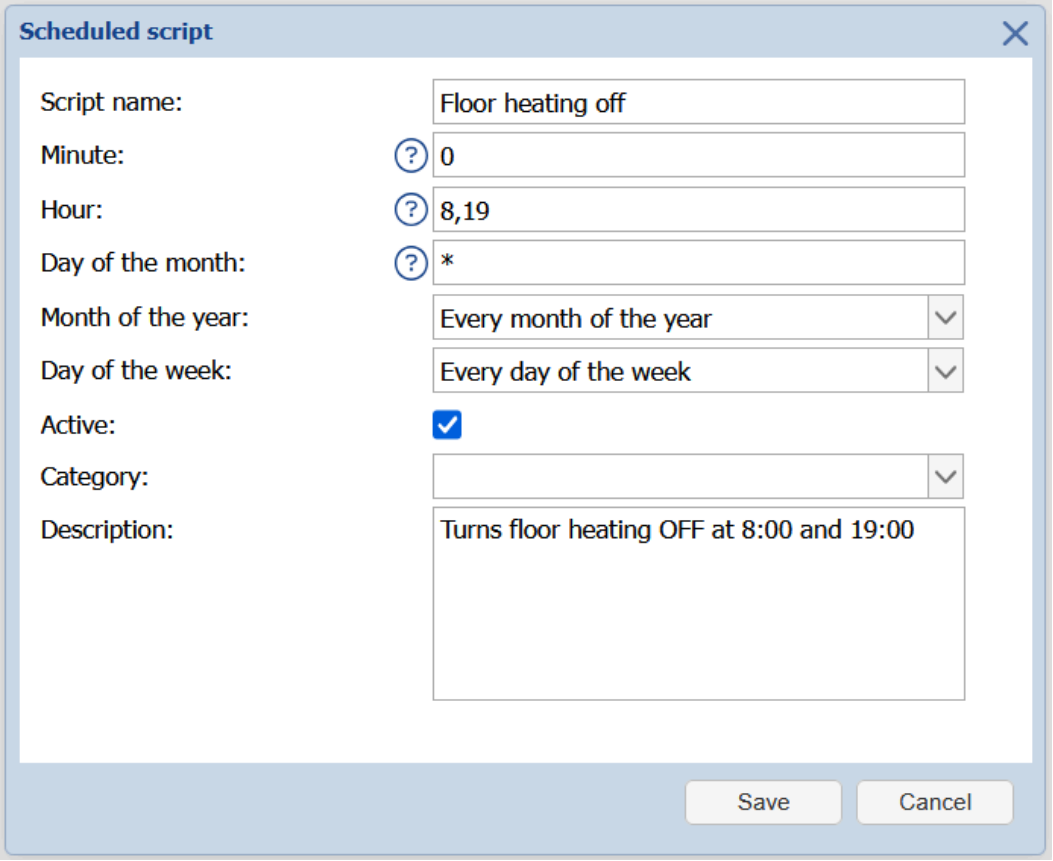
4.3.1.2. Resident



- *Script name* - name of the script
- *Sleep interval (seconds)* - delay between each script execution
- *Active* - whether the script is active (green circle) or disabled (red circle)
- *Category* - new or existing category for the script. This allows filtering scripts by category, it is also shown in the *Tools* → *Print* script listings page
- *Description* - description of the script

Note! Even though resident scripts are executed in parallel they should not have internal infinite loops or it will not be possible to reload scripts after editing.

4.3.1.3. Scheduled



- *Script name* - the name of the script
- *Minute, Hour, Day of the month, Month of the year, Day of the week* - specifies when the script is executed
- *Active* - whether the script is active (green circle) or disabled (red circle)
- *Category* - new or existing category for the script. This allows filtering scripts by category, it is also shown in the *Tools* → *Print* script listings page
- *Description* - description of the script

Scheduled scripts are executed when the system date/time matches the specified script start date/time. Scheduled script is run only once for each matching date/time.

Scheduled scripts use standard *cron* format for date/time parameters. Valid values are:

*** — execute script every minute, hour or day.

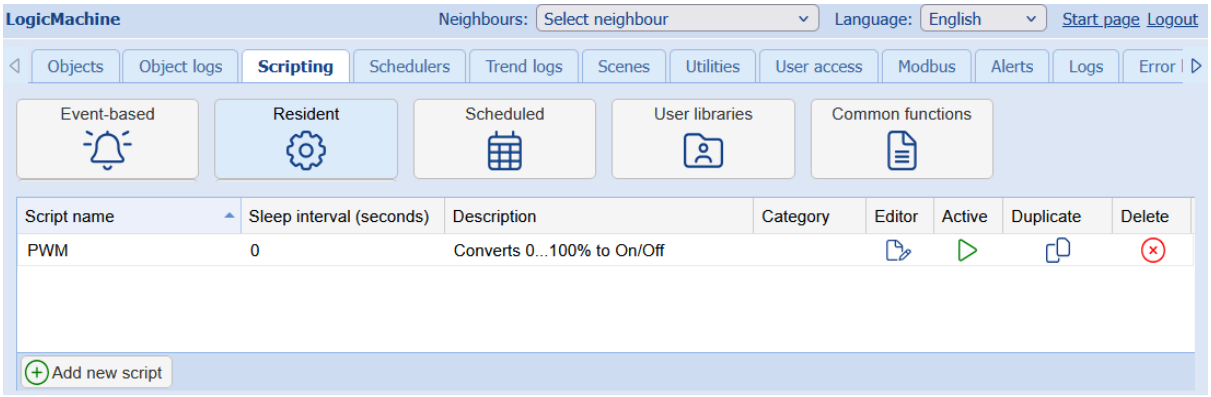
**/N* — execute script every N minutes, hours or days. N is an integer, script is executed when the current value divided by N gives 0 in modulo. For example, script with hour parameter set to **/8* will be executed when hour is 0, 8 and 16.

N — execute script exactly at Nth minute, hour or day.

N-K — execute script when minute, hour or day in the N-K range (inclusive).

N,K — it is possible to specify several *N* and *N-K* type parameters separated by a comma. For example, script with minute parameter set to 15,50-52 will get executed when minute is 15, 50, 51 and 52

4.3.2. List of scripts



Available actions:

- *Editor* - open scripting editor interface
- *Active* - activate (green) or deactivate (red) the selected script
- *Duplicate* - duplicate the selected script
- *Delete* - delete the selected script

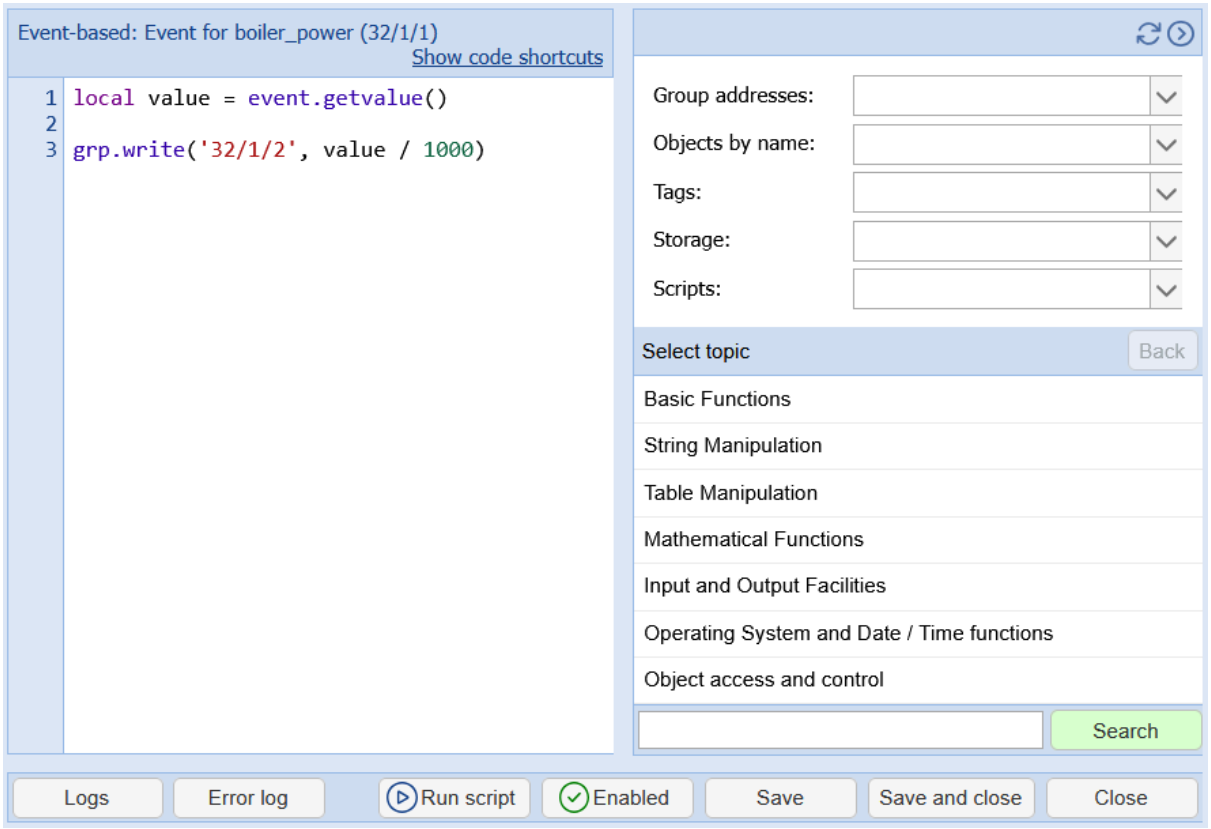
4.3.3. Script editor

Click  to open the editor.

4.3.3.1. Left sidebar

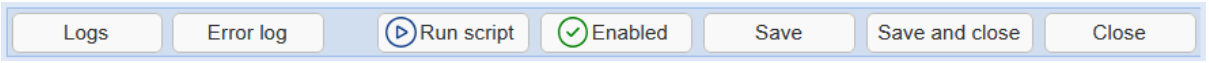
- *Helpers* - predefined code snippets categorized by the use case, click an entry to insert it into the script
- *Data types* - list of available data type constants, click an entry to insert it into the script
- *Scripts* - list of all scripts for quick switching

4.3.3.2. Right sidebar



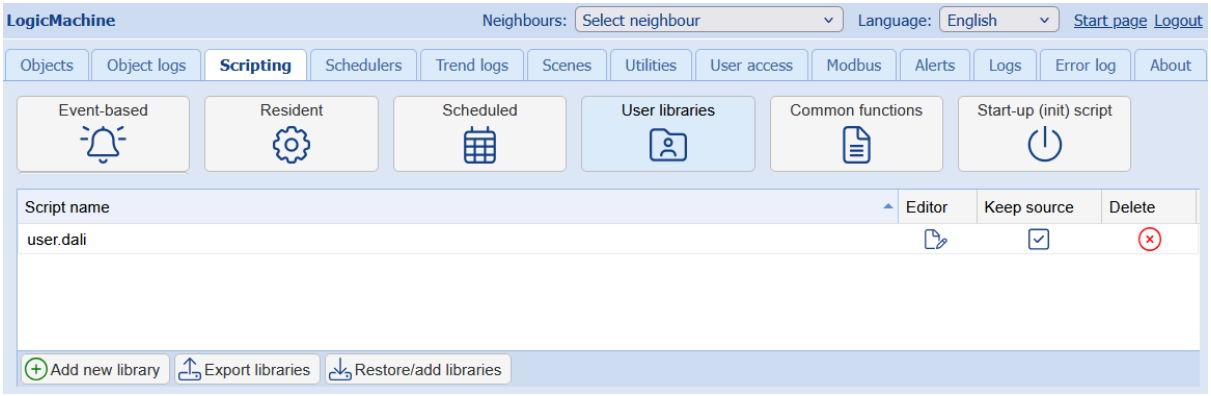
- *Group addresses, Objects by name, Tags, Storage, Scripts* - drop-down lists of all relevant values, click an entry to insert it into the script
- *Built-in help* - Lua function documentation, click plus to insert a code snippet into the script

4.3.3.3. Bottom toolbar

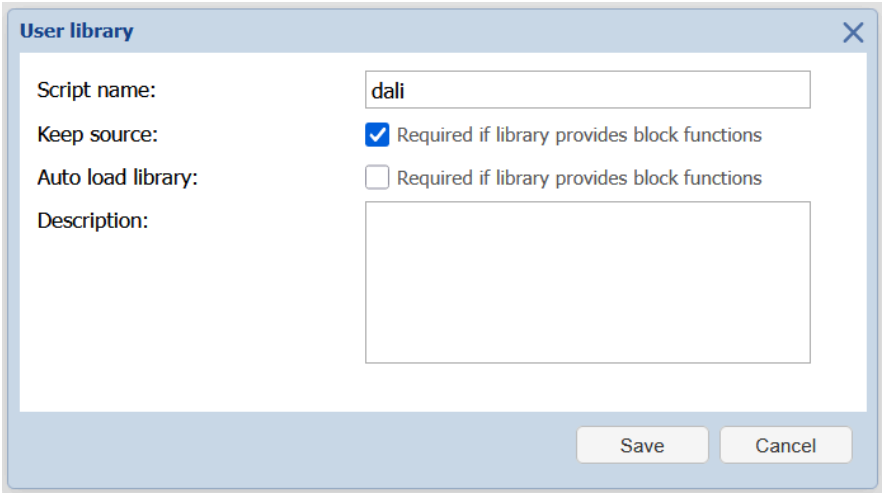


- *Logs* - display a floating window containing Error logs
- *Error logs* - display a floating window containing Error logs
- *Run script* - execute the script (not available for resident and libraries)
- *Enabled/Disabled* - toggle script status
- *Save* - save the script and continue editing
- *Save and close* - save the script and close the editor window
- *Close* - close the editor window without saving the script

4.3.4. User libraries



User libraries contain custom functions that are used in other scripts.



- *Script name* - unique library name
- *Keep source* - library is converted to a binary form that cannot be edited when this option is disabled. Make sure to keep a backup of the source code when disabling this option
- *Auto load library* - automatically load this library in all scripts
- *Description* - description of the library

User libraries are manually included in other scripts by calling *require('user.library_name')* unless *Auto load library* is enabled.

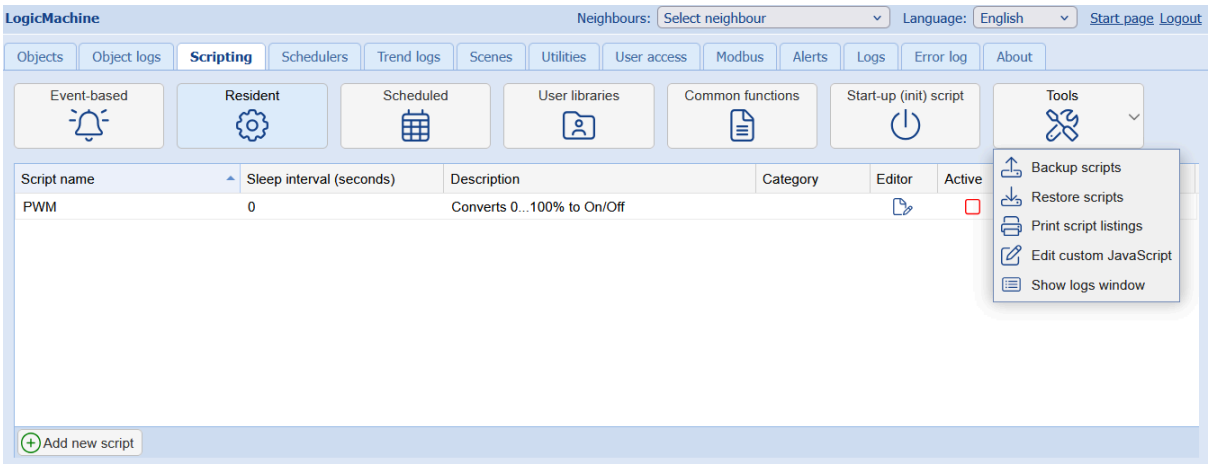
4.3.5. Common functions

Common functions is a library that is automatically included in all other scripts. Functions like *sunrise/sunset*, *email* are included by default.

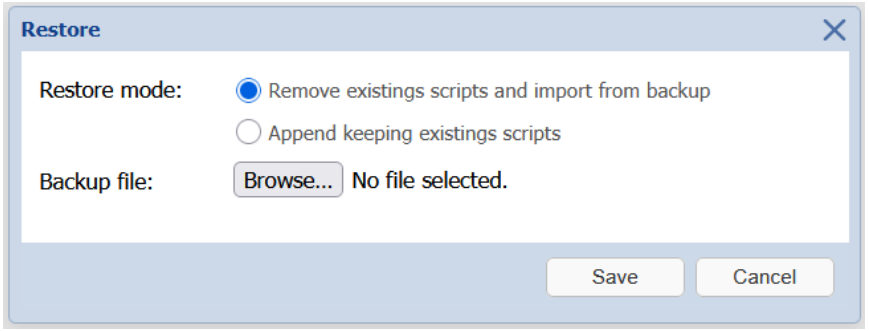
4.3.6. Start-up (init) script

Init script is run once each time the system starts.

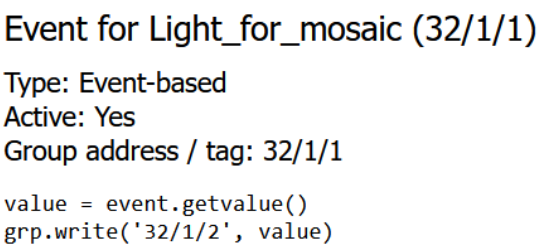
4.3.7. Tools



- *Backup scripts* - backup all scripts in *.gz file, *Common function* and *Start-up script* can be included in the backup if needed
- *Restore scripts* - restore scripts from a backup



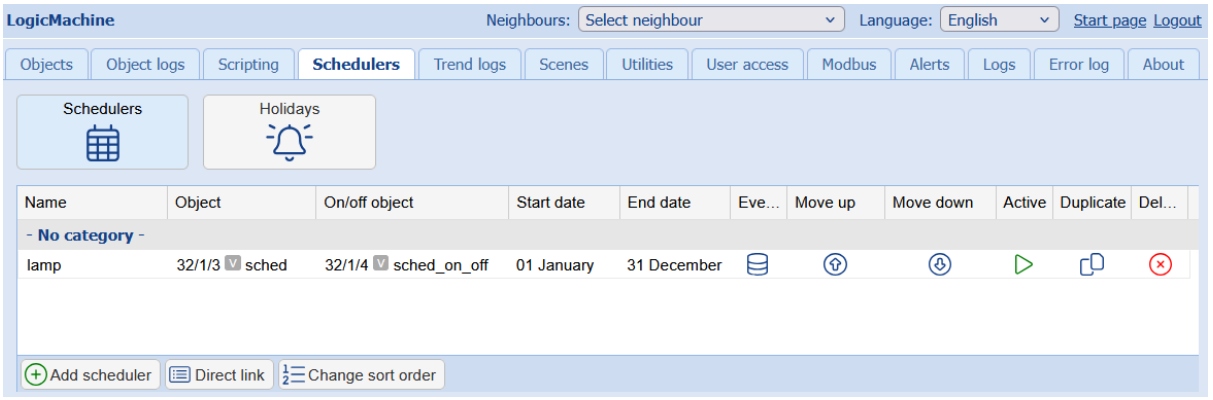
- *Print script listing* - shows all scripts in a single page ordered by Category



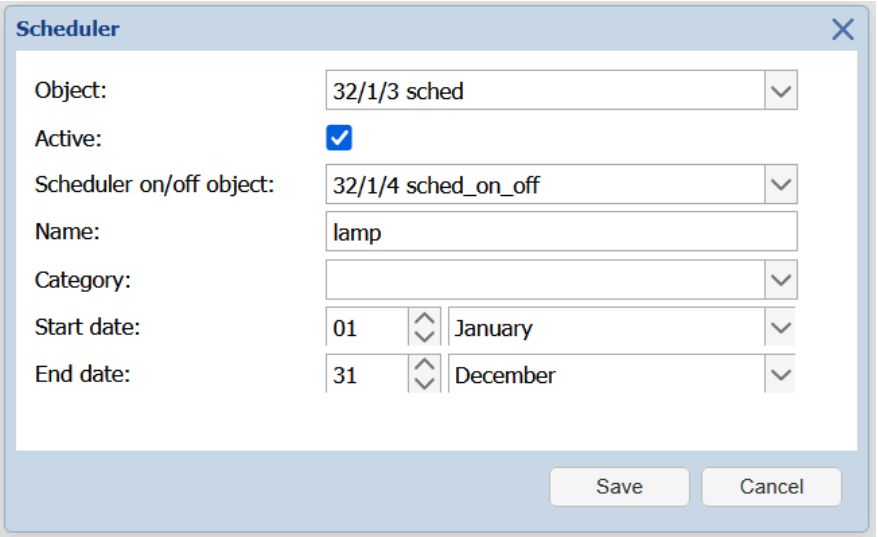
- *Edit custom JavaScript* - allows adding certain actions for the user Visualization (deprecated), schedulers and trends that are not possible by the built-in functionality. See examples at forum.logicmachine.net/showthread.php?tid=275
- *Show logs window* - show script logs in a separate floating window

4.4. Schedulers

Schedulers are used to specify events based on date/time when an object should be set to a predefined value. Correct date, time and timezone must be set in *Utilities*. Location coordinates can be provided to make sunrise and sunset event time more accurate. It is recommended to enable time synchronization (NTP).



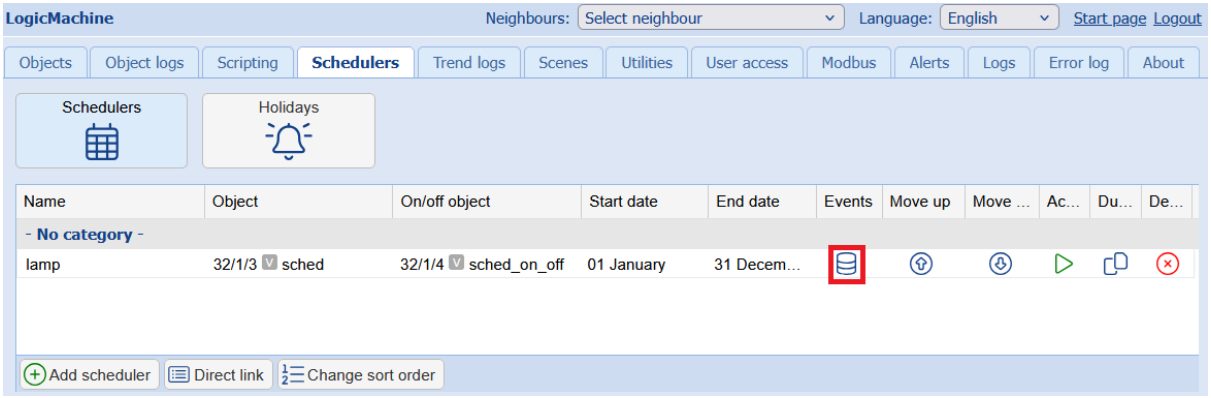
4.4.1. Add scheduler (admin interface)

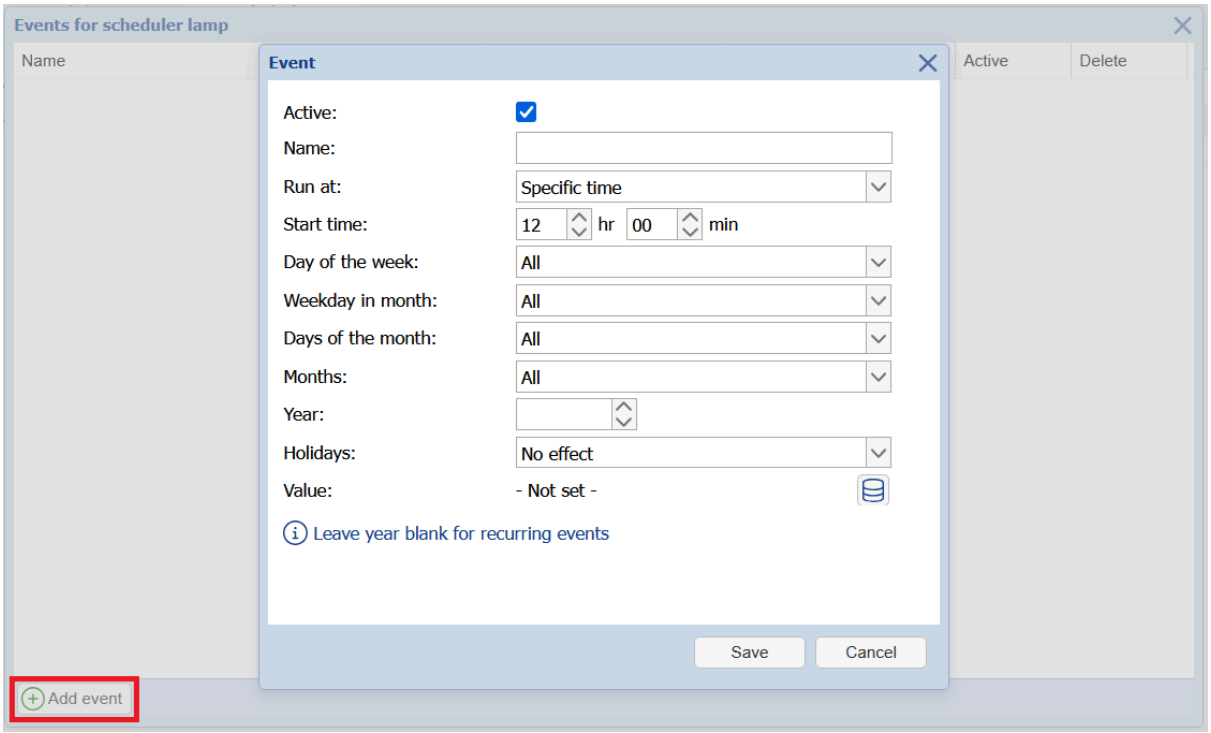


- *Object* - the object group address which will be controlled by scheduler
- *Active* - whether the scheduler is active or not
- *Scheduler on/off object* - object that can be used to enable/disable this scheduler
- *Name* - name of the scheduler
- *Category* - category of the scheduler
- *Start date* - start date of the scheduler
- *End date* - end date of the scheduler

4.4.2. Scheduler events (admin interface)

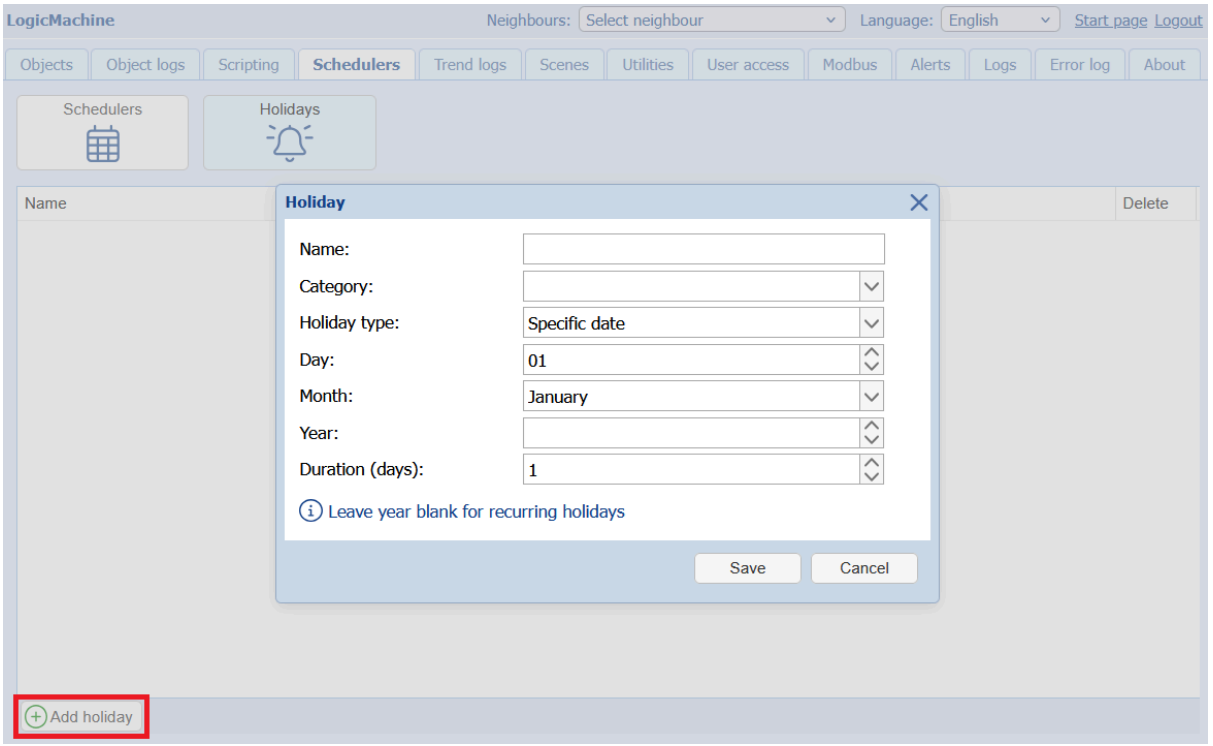
Events can be added both in the admin and the end user interfaces.





- *Active* - whether the event is active or not
- *Name* - name of the event
- *Run at* - specific time, sunrise, sunset
- *Start time* - start time for the event
- *Days of the week* - days of the week when the event will be triggered
- *Weekday in month* - weekdays of the month when the event will be triggered
- *Days of the month* - days of the month when the event will be triggered
- *Months* - months of the year when the event will be triggered
- *Year* - year when the event will be triggered
- *Holidays* - *"do not run on holidays"* or *"run only on holidays"*
- *Value* - value to send to the group address when the event is triggered

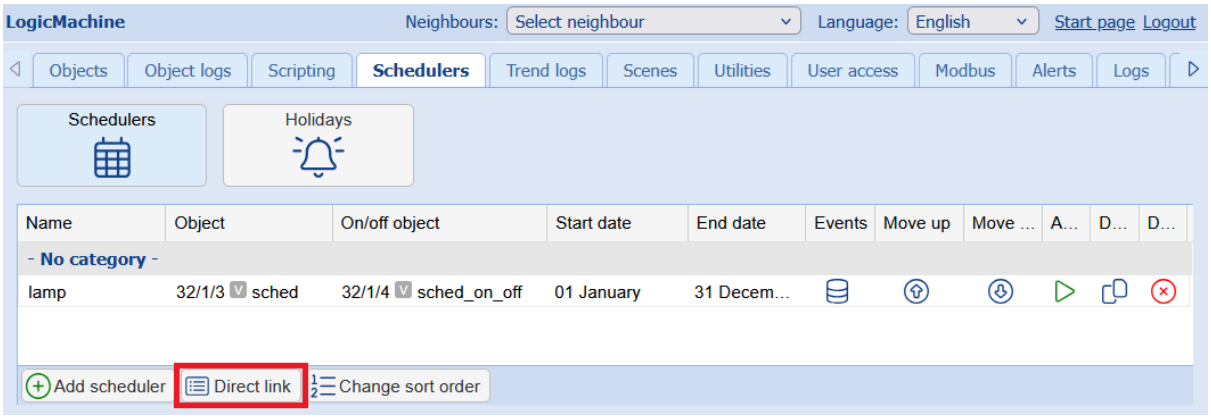
4.4.3. Scheduler holidays (admin interface)

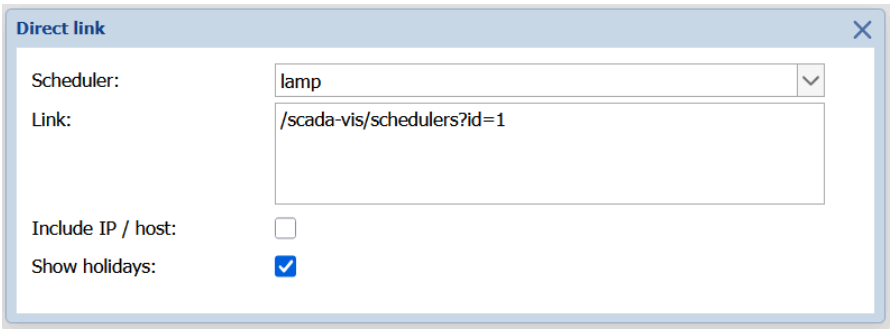


- *Name* - holiday name
- *Holiday type* - either *Specific date* or *Day of the week*
    - *Specific date:*
        - *Day* - day of the month value
    - *Day of the week:*
        - *Day of the week* - specific day of the week (e.g. 2nd Monday)
- *Month* - holiday month value
- *Year* - holiday year value, leave blank when a holiday recurs every year
- *Duration (days)* - holiday length in days

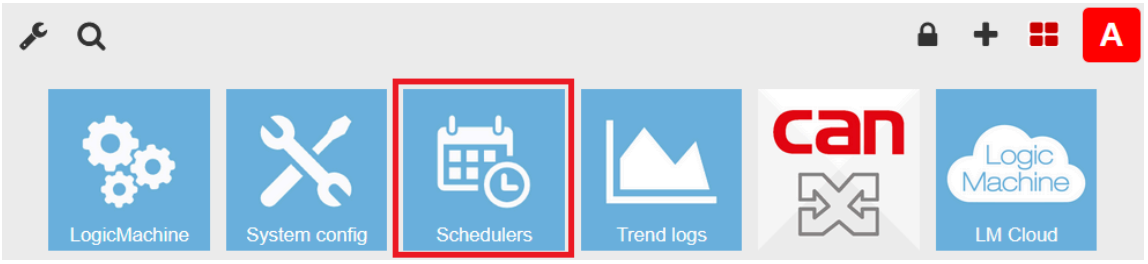
4.4.4. Direct link (admin interface)

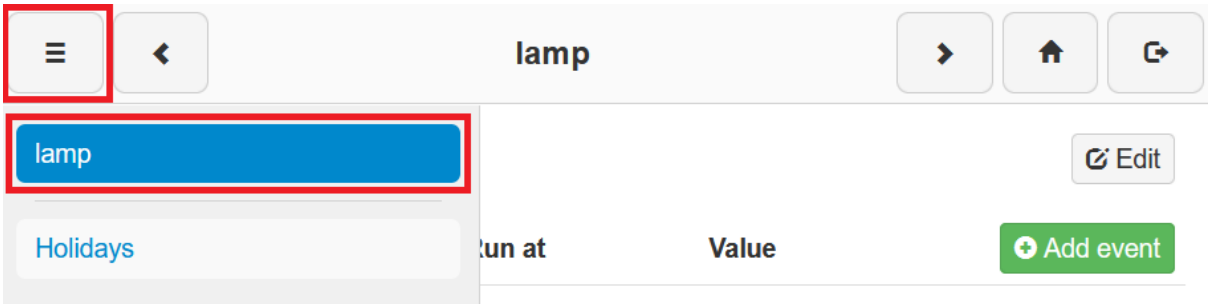
To get a direct link to a specific scheduler click the *Direct link* button. This link can be used to include scheduled in the *Visu* via the *Frame* widget.

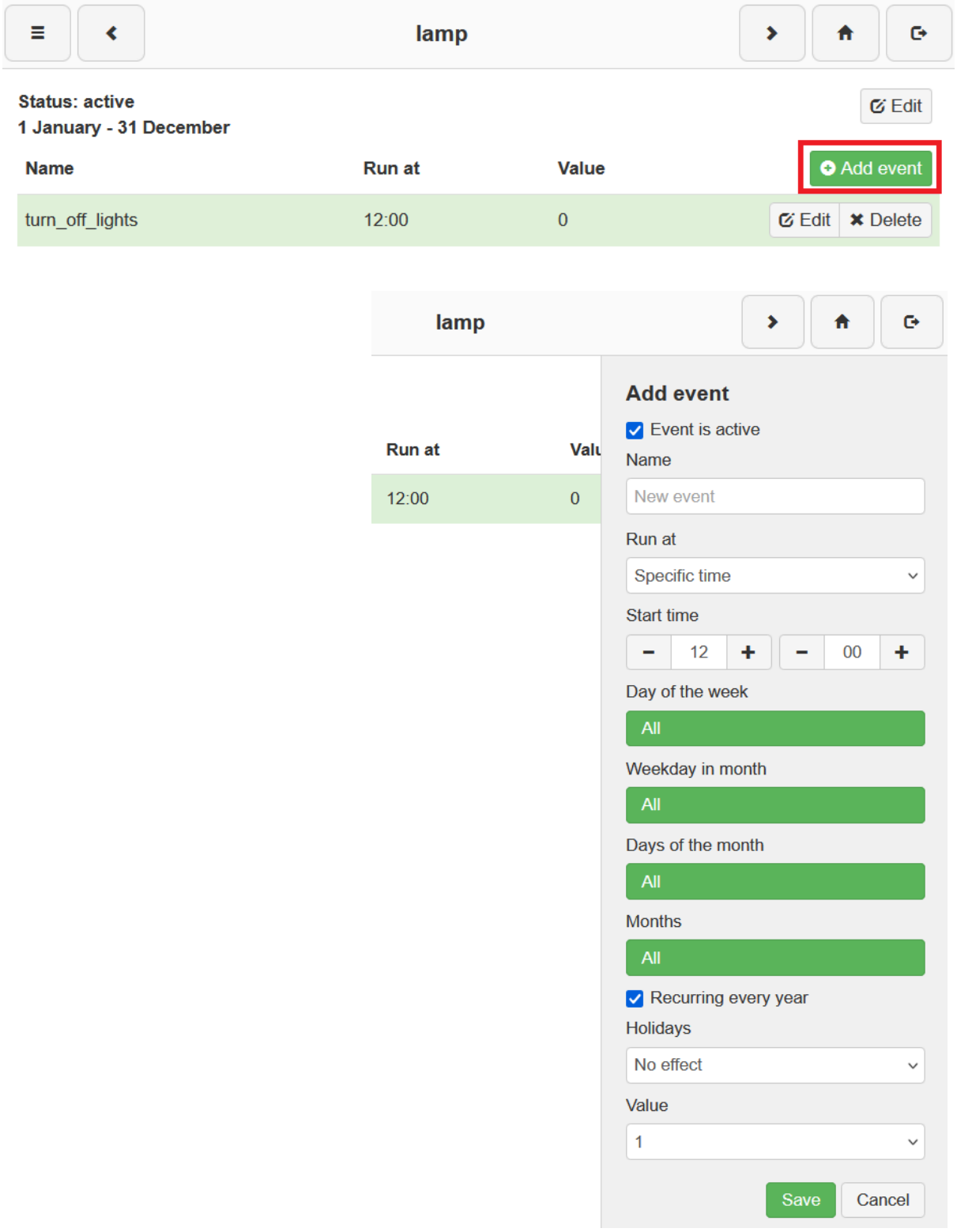




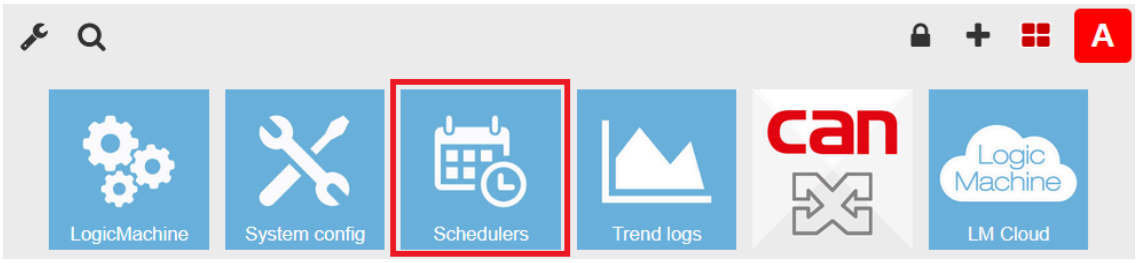
4.4.5. Scheduler events (user interface)

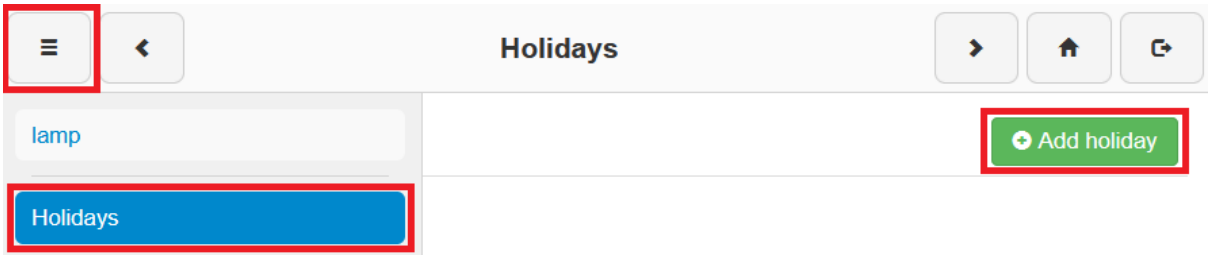


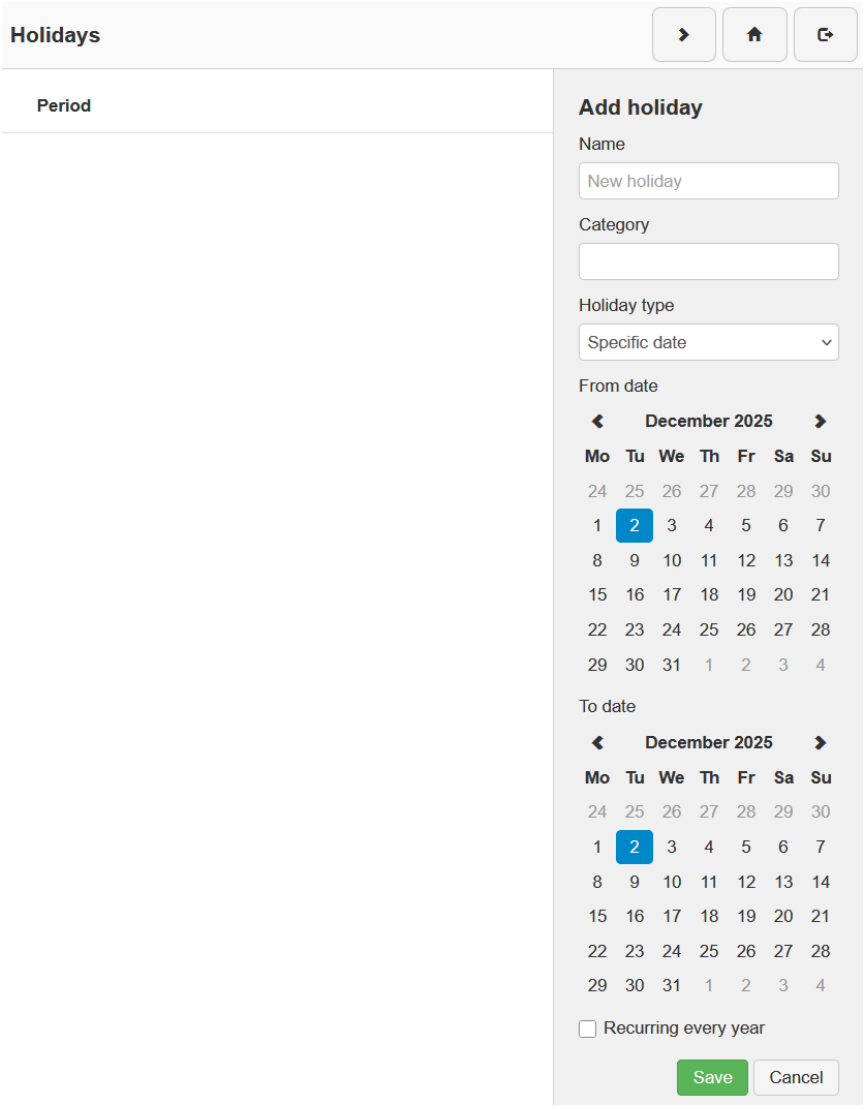



lamp
Status: active
1 January - 31 December
Edit
Name
Run at
Value
Add event
turn_off_lights
12:00
0
Edit
Delete
lamp
Run at
Valu
12:00
0
Add event
Event is active
Name
New event
Run at
Specific time
Start time
12
00
Day of the week
All
Weekday in month
All
Days of the month
All
Months
All
Recurring every year
Holidays
No effect
Value
1
Save
Cancel

4.4.6. Scheduler holidays (user interface)

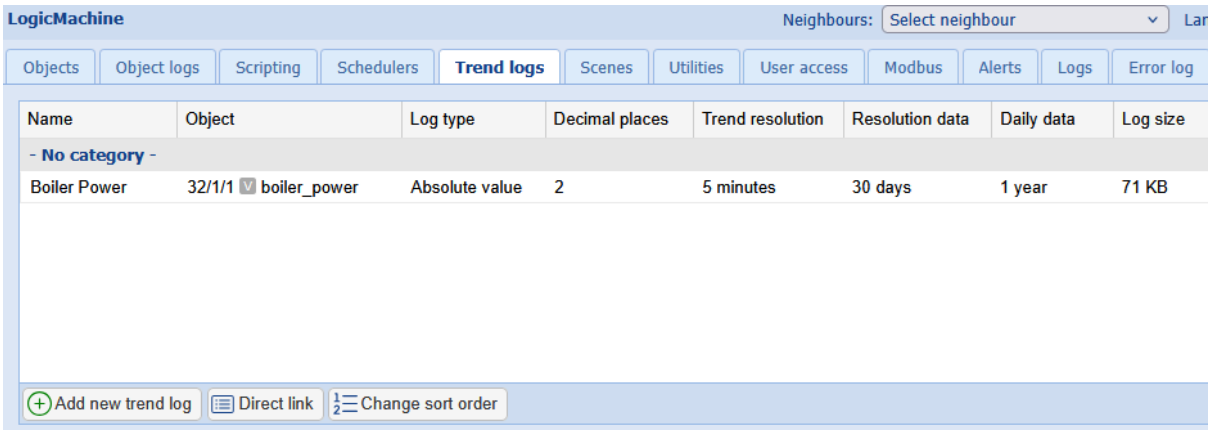




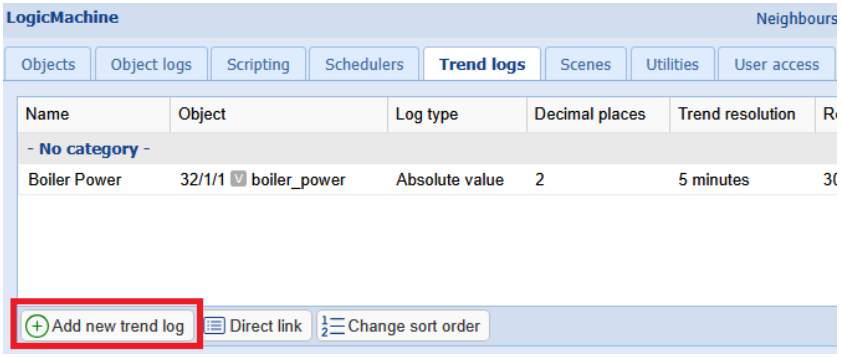


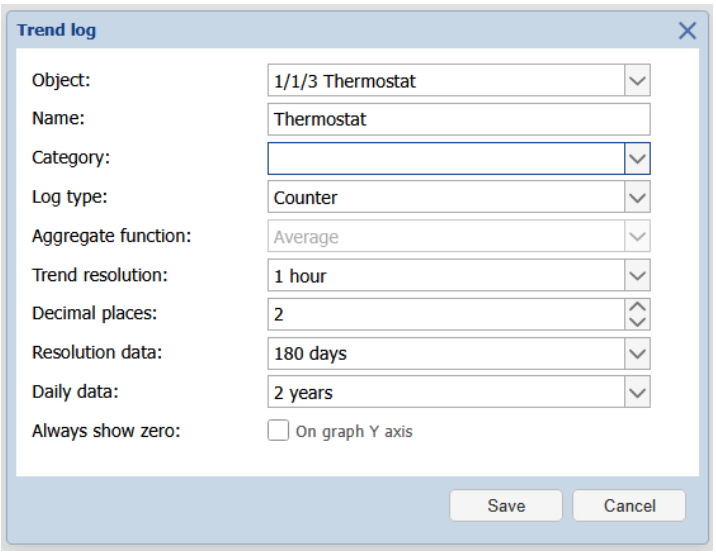
4.5. Trend logs

Trend logs store object data for a certain period of time with a predefined interval between each value.



4.5.1. Add new trend log (admin interface)





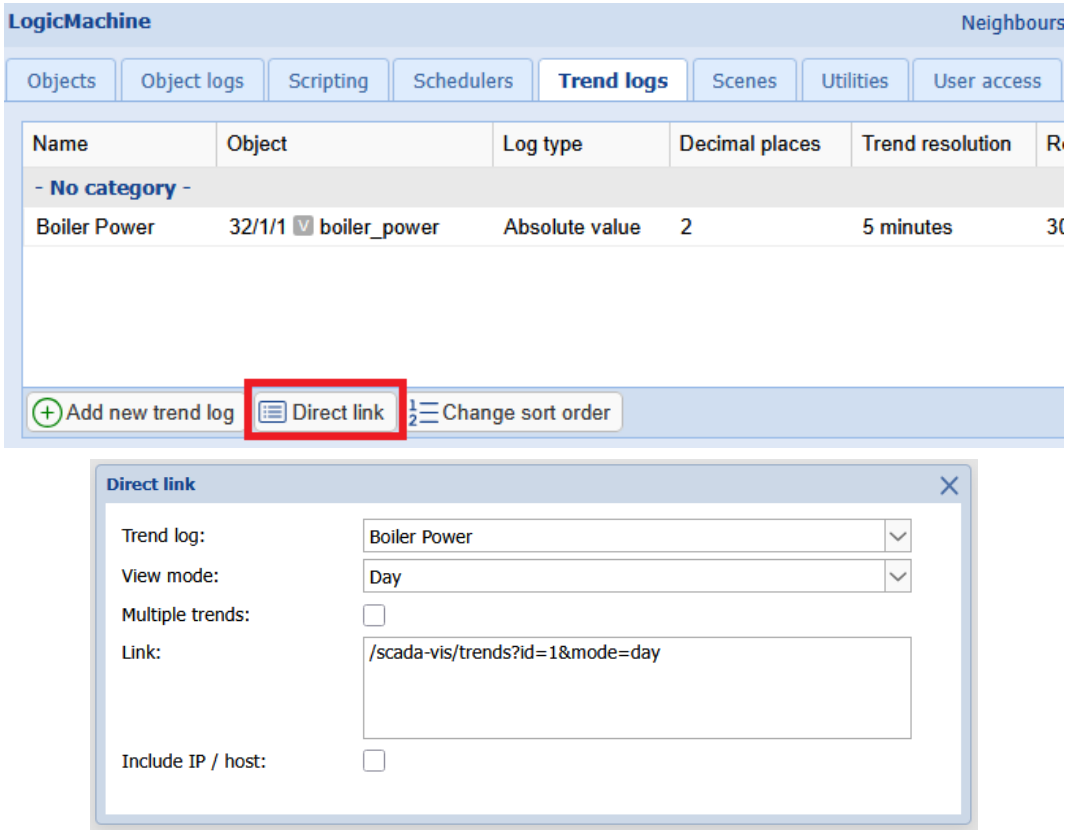
- *Object* - object which value is used as trend data source
- *Name* - name of the trend
- *Category* - category of the trend
- *Log type [Counter, Counter with negative delta, Absolute value]* - type of the log. *Counter* type is used for metering data which is always growing (electricity,

water, gas), *Absolute value* is used for temperature, humidity sensor data. *Counter with negative delta* is used for metering data that can change both ways (solar installations connected to the grid).

- *Aggregate function [Average, Minimum, Maximum, Last value]* - function which aggregates trend data into lower resolution, only available for the *Absolute value* type.
- *Trend resolutions [5 min .. 1 hour]* - how often the trend value is stored
- *Decimal places* - decimal places for the object value
- *Resolution data* - time period for which the data at the defined resolution is kept
- *Daily data* - time period for which the daily data is kept
- *Always show zero* - whether to always show zero on the graph Y axis

4.5.2. Direct link (admin interface)

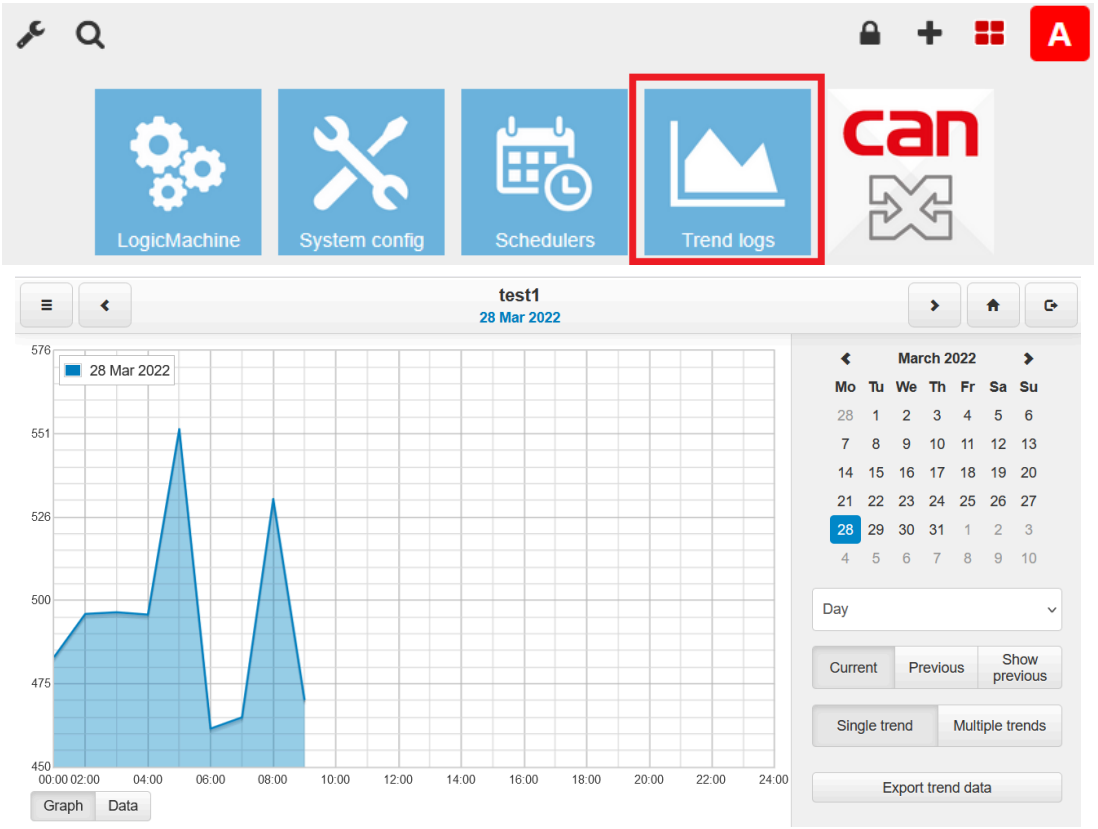
To get a direct link to a specific trend log click *Direct link*.



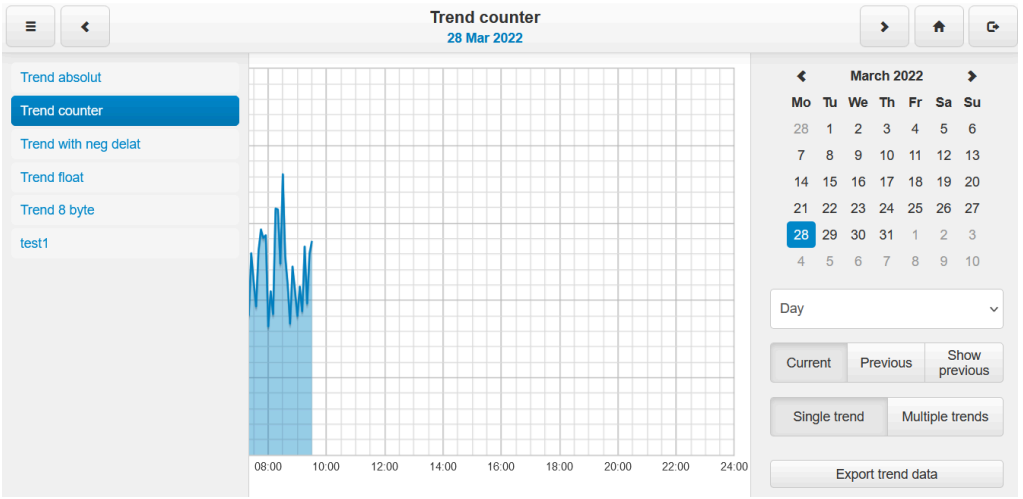
4.5.3. Trend log functions for scripts

kb.logicmachine.net/libraries/trends/

4.5.4. Trend log example (user interface)

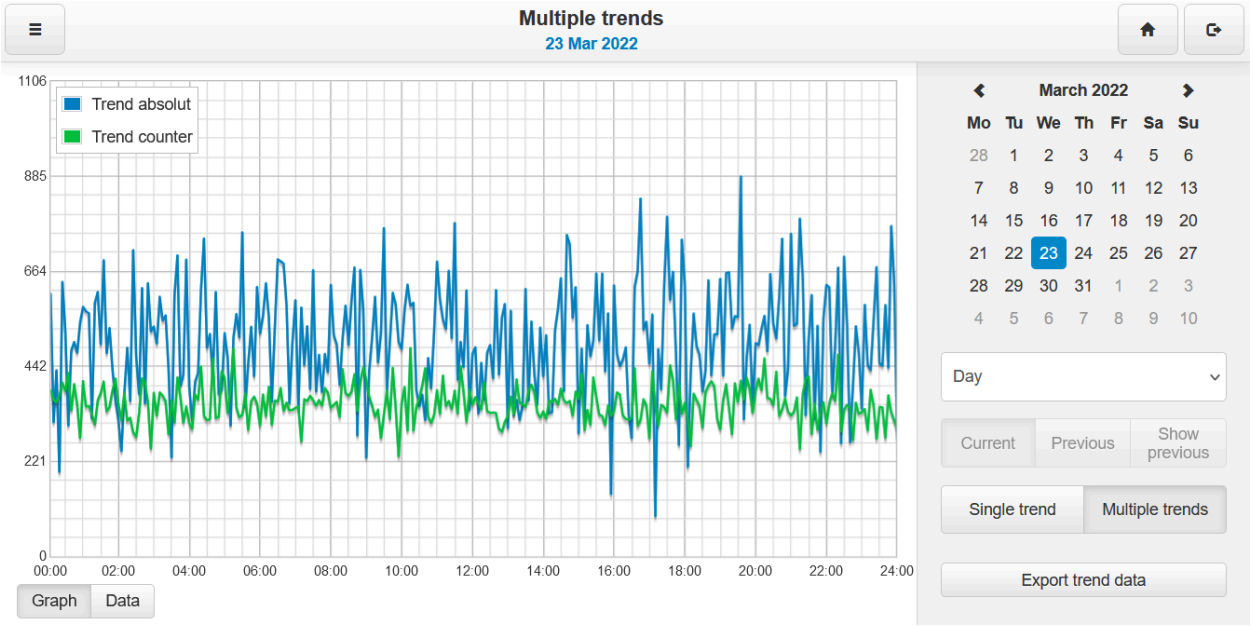


Click *Menu* ☰ to select which trend log to view. In *Multiple trends* view click each trend to toggle its display.

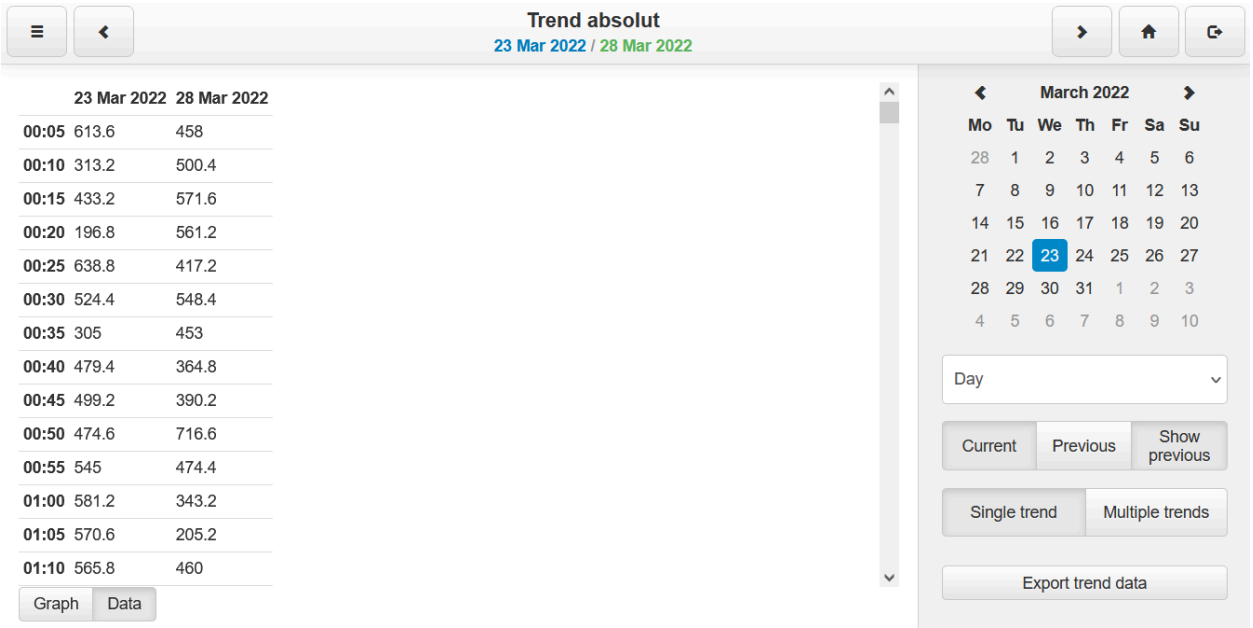


- *Day/Week/Month/Year* - change between different view
- *Current* - select the date for the current data
- *Previous* - select the date for the previous data
- *Show previous* - toggle previous data display, current and previous data can be shown together for comparison

- *Single trend / Multiple trends* - toggle between single and multiple trend log display
- *Export trend data* - export selected trend data into a CSV file

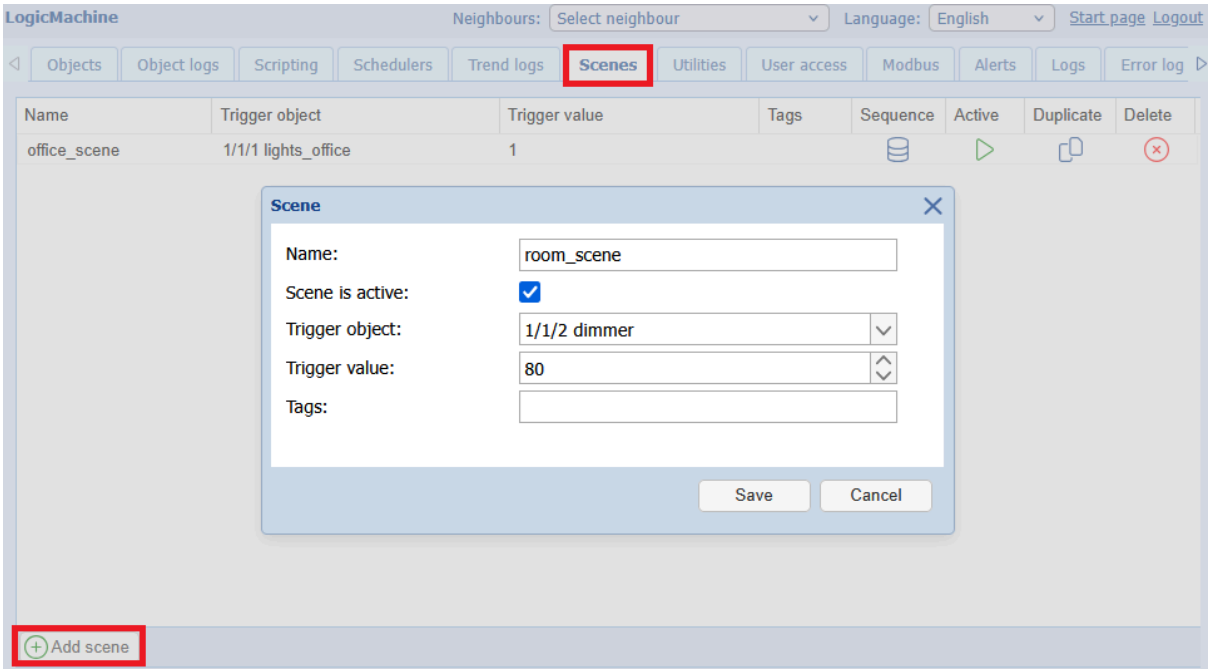


Click *Data* to view trend data in a table view.



4.6. Scenes

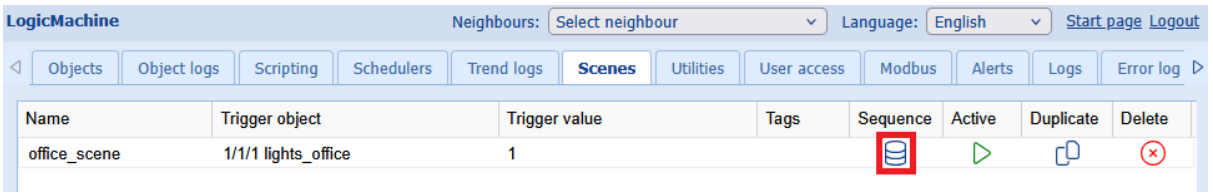
4.6.1. Add a scene

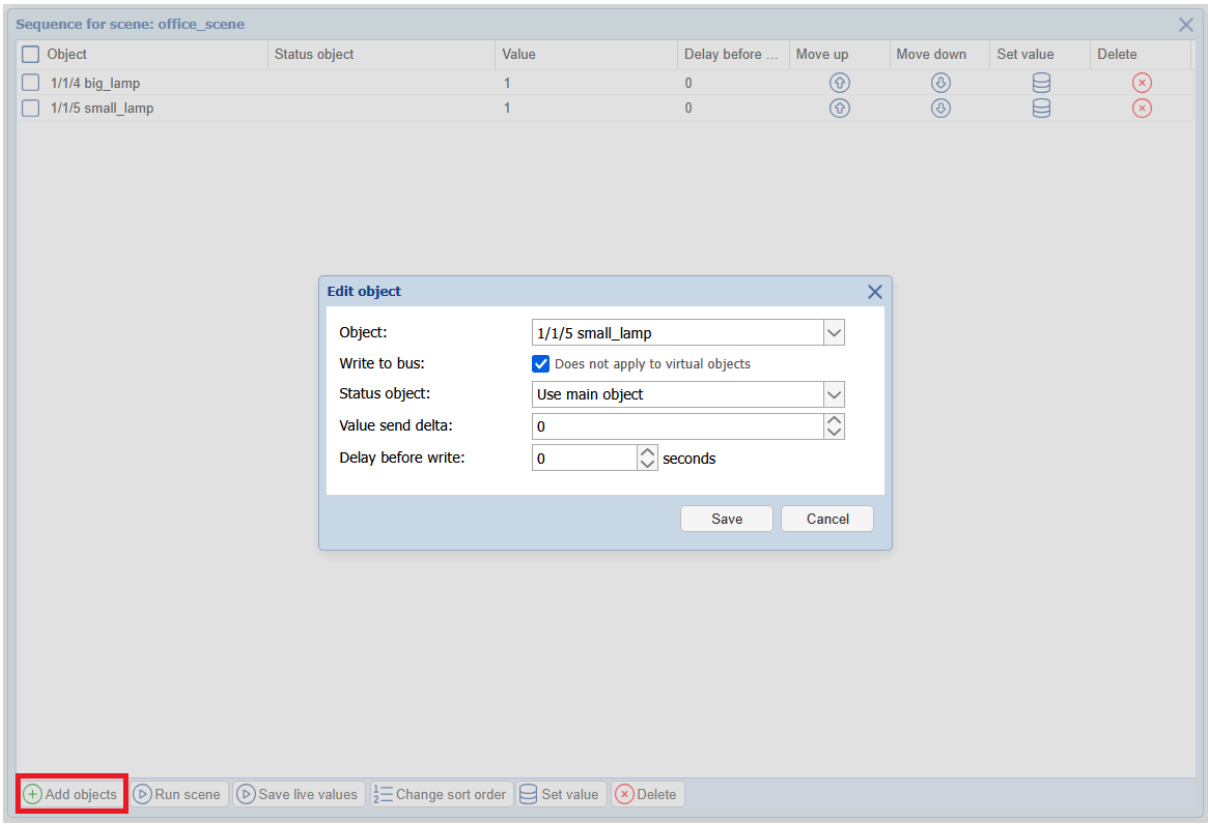


- *Name* - name for the scene
- *Scene is active* - whether the scene is active
- *Trigger object* - object that triggers the scene
- *Trigger value* - object value that triggers the scene
- *Tags* - tags of the scene, can be used in scripts to run multiple scenes

4.6.2. Add objects to the scene sequence

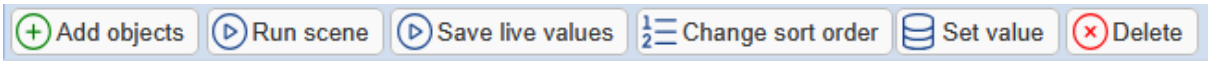
Click *Sequence*  to add objects that will be controlled by the scene





- *Object* - sequence object
- *Write to bus* - where the write will be sent to KNX/TP bus
- *Status object* - optional object that can be used to get the value when saving live values
- *Value send delta* - a new value is only sent if the absolute difference between it and the last updated value is greater than this delta
- *Delay before write* - delay in seconds before the object value is written

4.6.3. Scene sequence toolbar



- *Run scene* - execute this scene
- *Save live values* - use current object value as sequence values
- *Change sort order* - change sequence object order via drag&drop
- *Set value* - set value for a selected objects
- *Delete* - delete selected objects from the sequence

4.7. Visualization structure

See 2024 manuals

4.8. Visualization

See 2024 manuals

4.9. Visualization graphics

See 2024 manuals

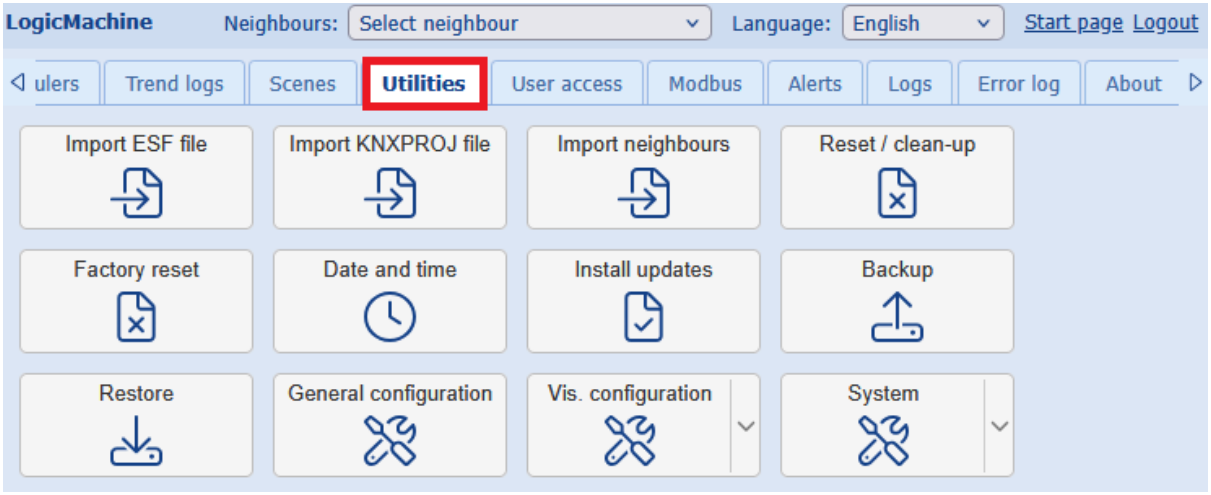
To **enable** *Visualization structure / Visualization / Visualization graphics* following script should be executed once:

```
require('uci')
uci.delete('genohm-scada.core.visdisabled=0')
uci.commit('genohm-scada.core')
```

To **disable** *Visualization structure / Visualization / Visualization graphics* following script should be executed once:

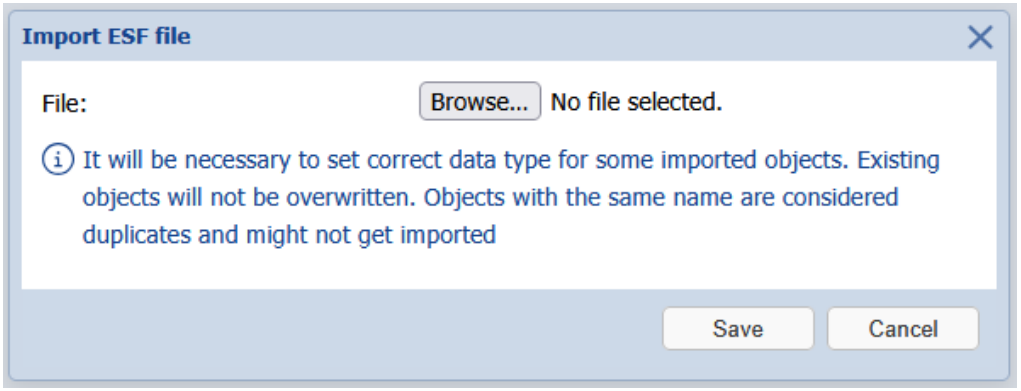
```
require('uci')
uci.set('genohm-scada.core.visdisabled=1')
uci.commit('genohm-scada.core')
```

4.10. Utilities

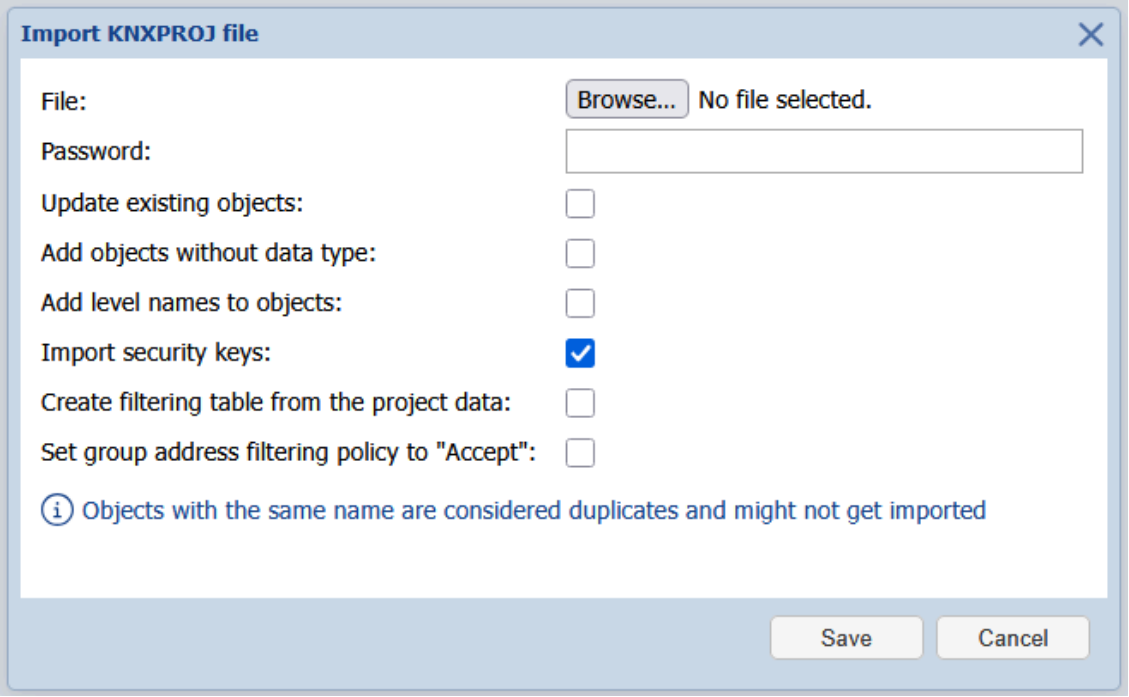


4.10.1. Import ESF file

Where possible KNXPROJ files should be imported instead. ESF does not provide full data type description of objects meaning that some data types will have to be corrected manually.



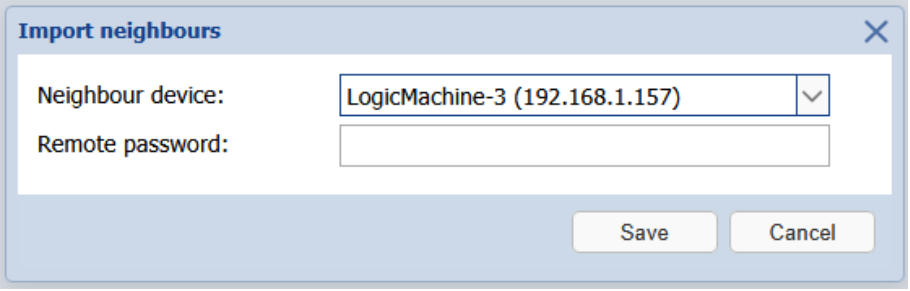
4.10.2. Import KNXPROJ file



- *Password* - ETS project password (optional)
- *Update existing objects* - updates existing objects
- *Add objects without data type* - whether to import objects that do not a have a data type assigned
- *Add level names to objects* - when enabled the object name is formatted as *Main group name - Middle group name - Group address name*
- *Import security keys* - for KNX secure communication
- *Create filtering table from the project data* - creates a filtering table to control which group address telegrams are allowed to pass
- *Set group address filtering policy to "Accept"* - telegrams with group addresses not included in the filtering table may be blocked, depending on the filtering configuration

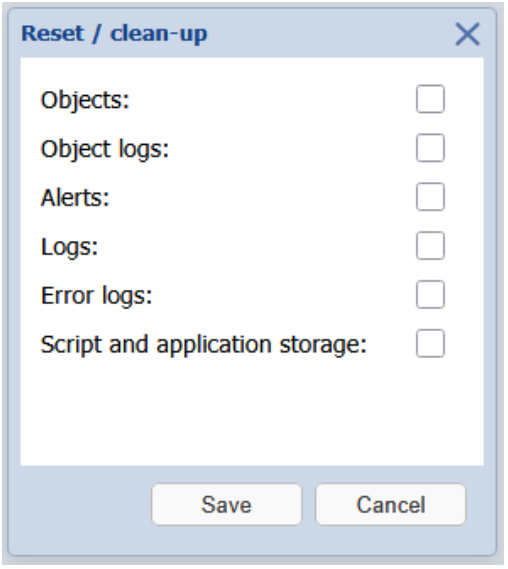
4.10.3. Import neighbours

Import objects from another LogicMachine on the network. Remote services must be enabled on the neighbour device.



4.10.4. Reset / clean-up

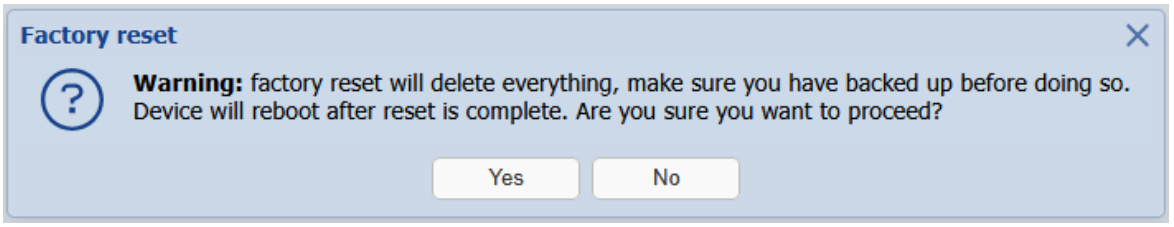
Use *Reset / clean-up* to delete certain system



Warning: clearing script and application storage will delete configuration for installed applications.

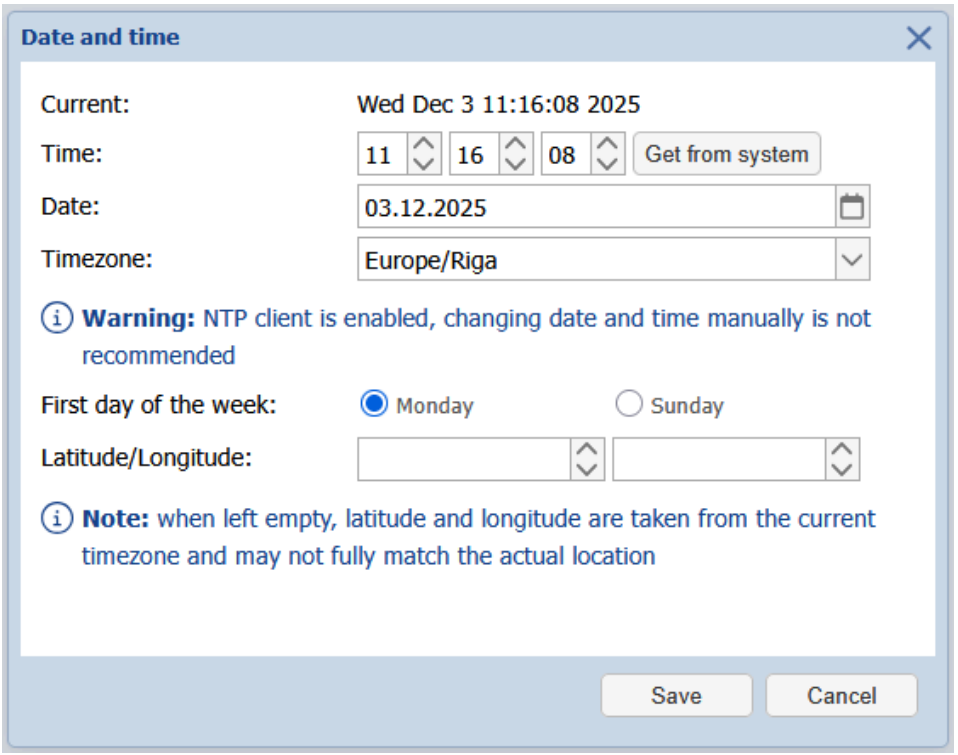
4.10.5. Factory reset

Delete current configuration and return to factory defaults. Does not affect System configuration settings.



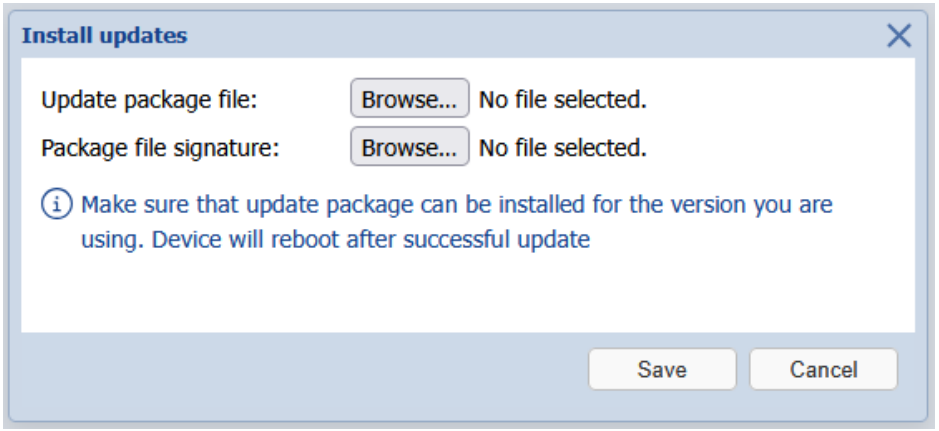
4.10.6. Date and time

For better precision of *Scheduler* sunrise/sunset functionality it is recommended to provide exact coordinates of the Latitude/Longitude.



4.10.7. Install updates

Updates are provided via *.lmup package files. Depending on an update an automatic system reboot might be performed after installation.



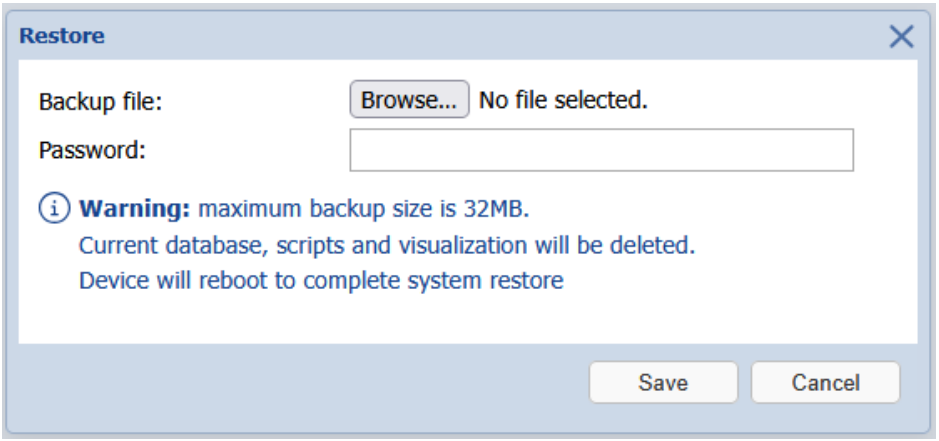
4.10.8. Backup

Perform project backup as a single archive.

An optional password for the backup can be added.

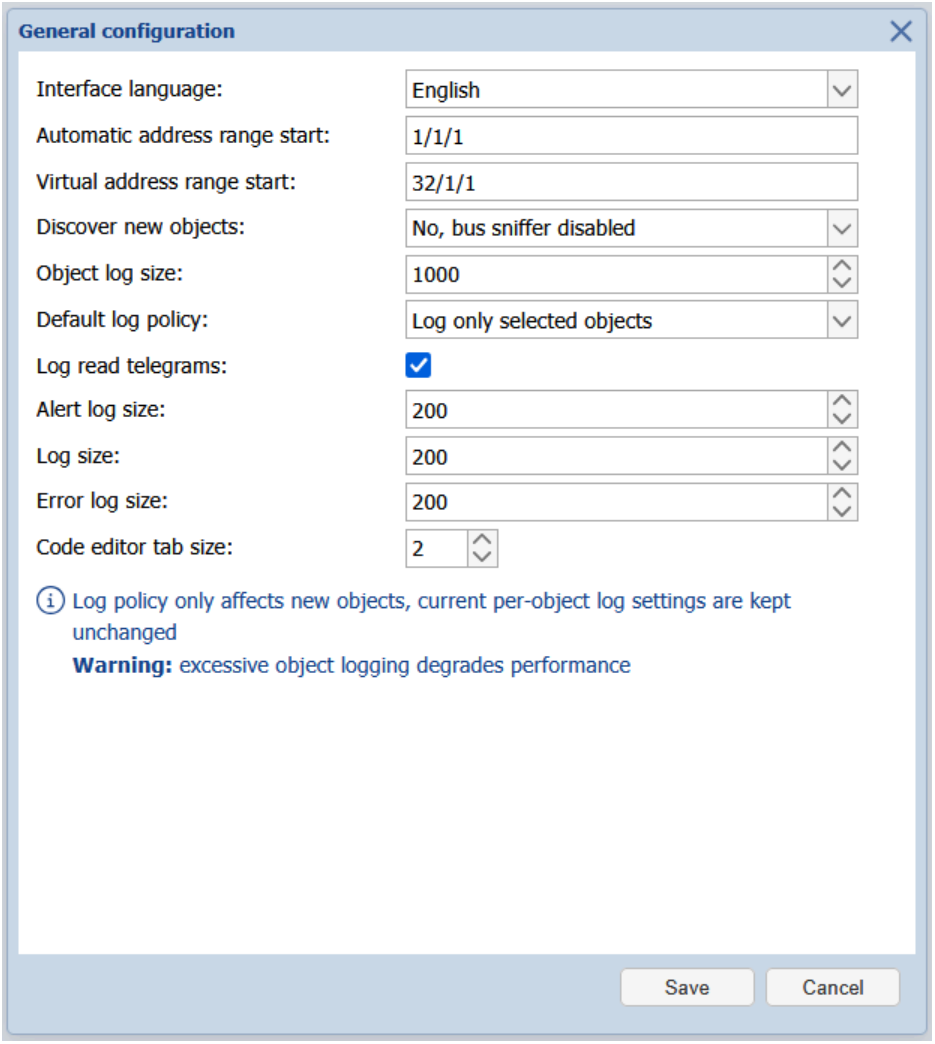
4.10.9. Restore

Restore project from a backup. The current project configuration will be overwritten.



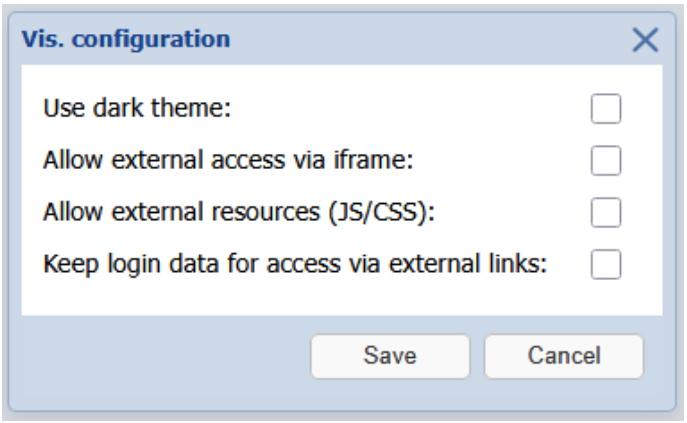
If your backup is protected with a password, then you need to fill the *Password* field.

4.10.10. General configuration

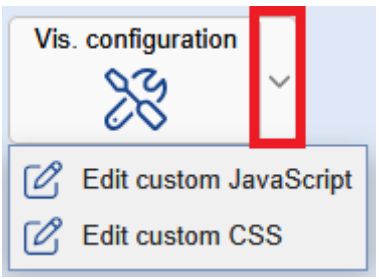


- *Interface language* - interface language
- *Automatic address range start* - starting group address for standard objects
- *Virtual address range start* - starting group address for virtual objects
- *Discover new objects* - whether to enable KNX bus sniffer to automatically add newly discovered objects
- *Object log size* - maximum number of entries for *Object logs*
- *Default log policy* - whether to enable the *Log* property automatically for newly discovered objects
- *Log read telegrams* - whether to log read telegrams, otherwise only write and response telegrams are logged
- *Alert log size* - maximum number of entries for *Alerts*
- *Log size* - maximum number of entries for *Logs*
- *Error log size* - maximum number of entries for *Error logs*
- *Code editor tab size* - number of spaces to insert when pressing *Tab* in the scripting editor

4.10.11. Visualization configuration



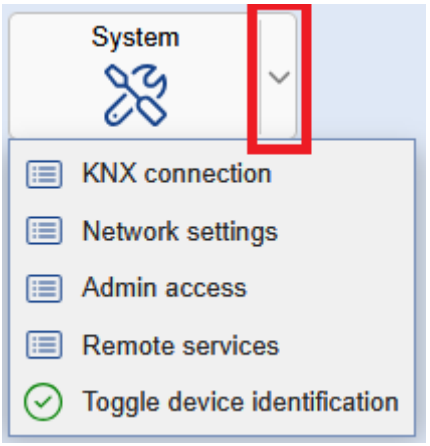
- *Use dark theme* - dark theme for Visu
- *Allow external access via iframe* - allow the pages of one LogicMachine to be embedded into the Visu of another LogicMachine
- *Allow external resources (JS/CSS)* - external resources (JS/CSS) for Visu
- *Keep login data for access via external links* - allows saving of login credentials when LM is accessed via an external link (for example from Visualization on another LM)



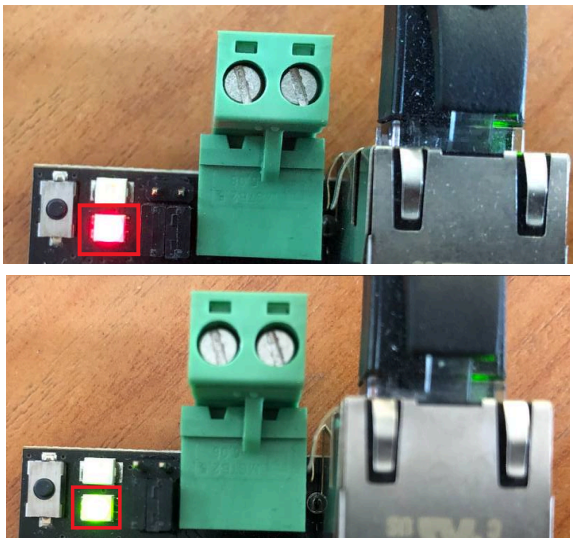
- *Edit custom JavaScript* - custom JavaScript for Visu
- *Edit custom CSS* - custom CSS for Visu

4.10.12. System

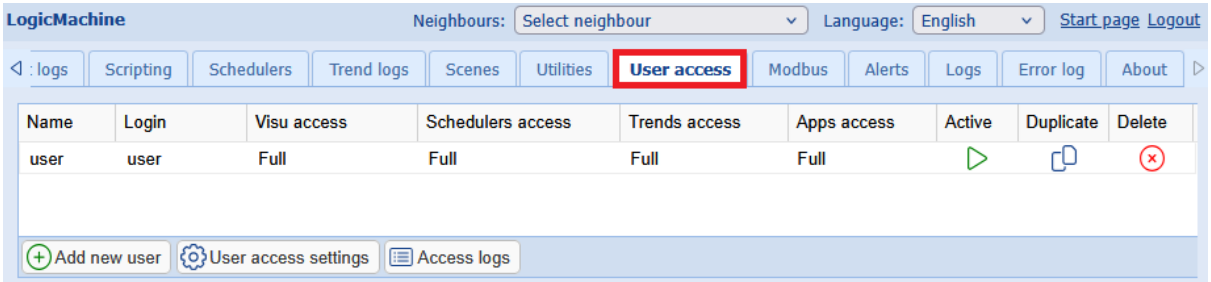
Quick access to certain *System configuration* settings.



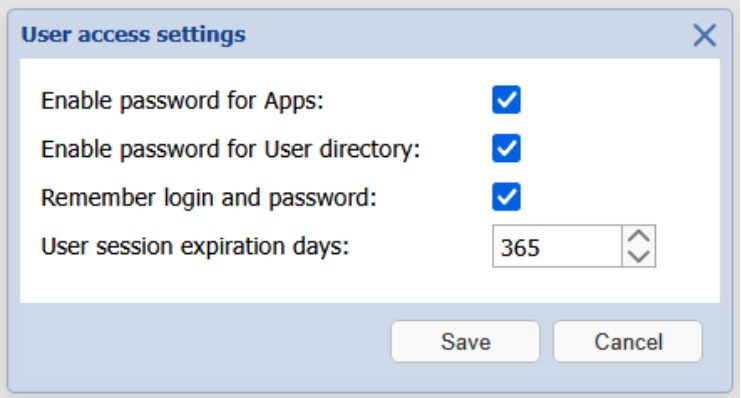
When *Toggle device identification* is clicked, LED2 starts blinking red and green. Click it again to stop this process.



4.11. User access



4.11.1. User access settings

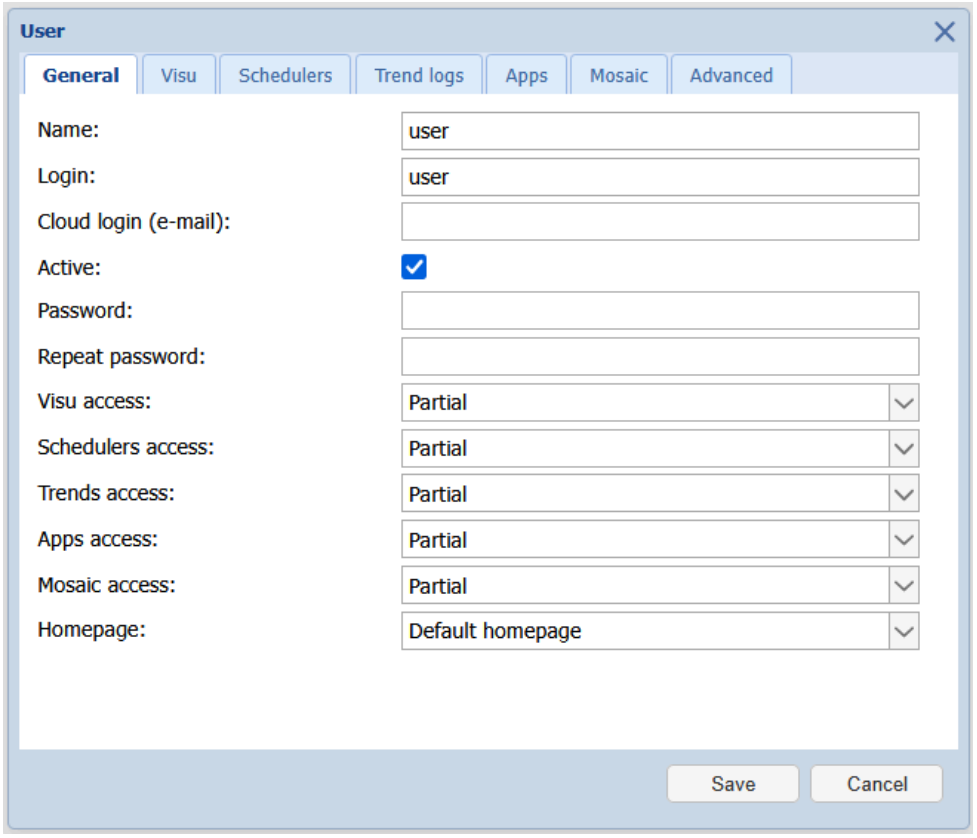


- *Enable password for Apps* - enable password for the main application page
- *Enable password for User directory* - enable password access for the *user* directory
- *Remember login and password* - whether to save credentials on the client device. Does not apply to the *admin* user. Credentials become invalid if client IP address changes
- *User session expiration days* - how many days the saved credentials are stored

4.11.2. User directory

HTTP server-side scripts (.lp) files can be placed into the *user* directory to provide additional functionality. FTP server must be enabled to upload files to this directory. Visit our forum for examples: forum.logicmachine.net

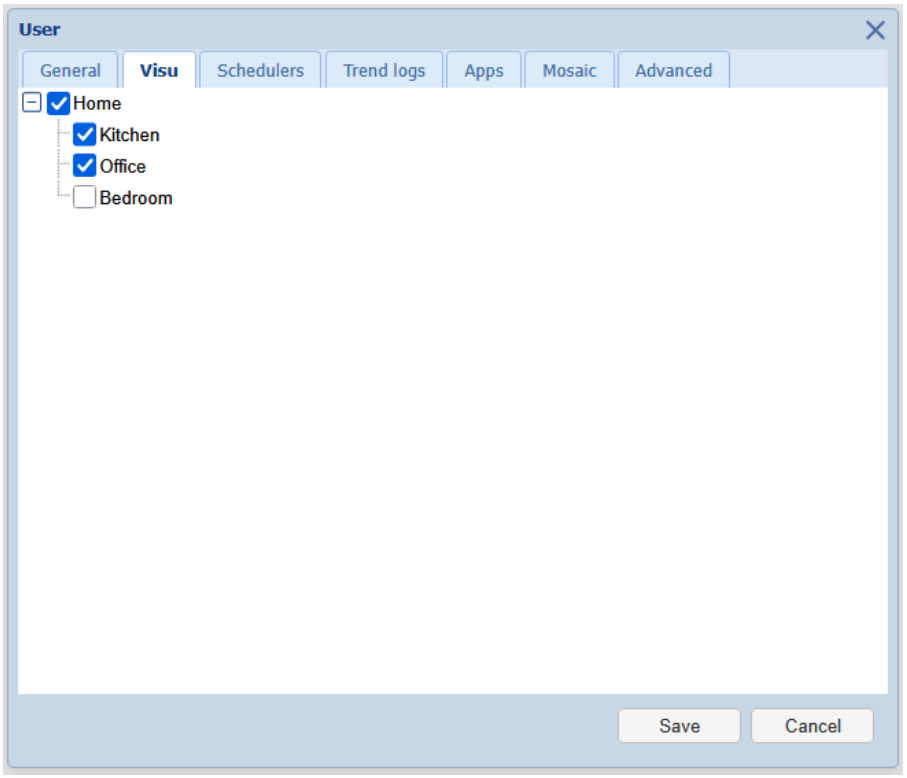
4.11.3. Adding users

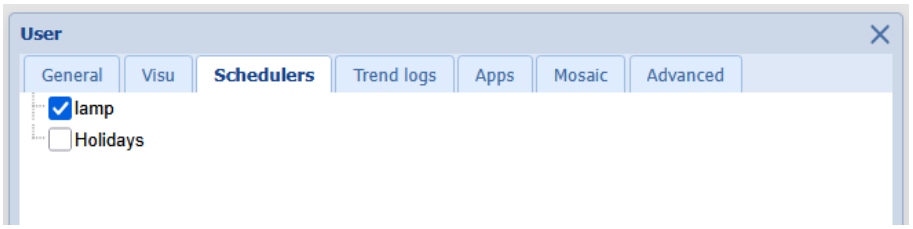


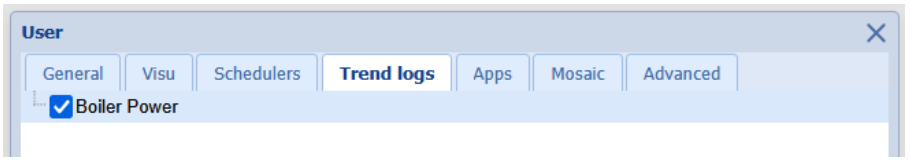
- *Name* - name of the user
- *Login* - user login name
- *Cloud login (e-mail)* - login for cloud (e-mail address)
- *Active* - whether the user is active or not. Inactive users can't access the system
- *Password* - user password
- *Visu access [None, Partial, Full]* - *Visu* access rights
- *Schedulers access [None, Partial, Full]* - *Schedulers* access rights
- *Trends access [None, Partial, Full]* - *Trends* access rights
- *Apps access [None, Partial, Full]* - *Apps* access rights
- *Mosaic access [None, Partial, Full]* - *Mosaic* access rights
- *Homepage* - default page that is shown after a user is logged in; it can be the *Default homepage*, *Visu*, *Visu (admin)*, *Schedulers*, *Trend logs* or any of the installed applications

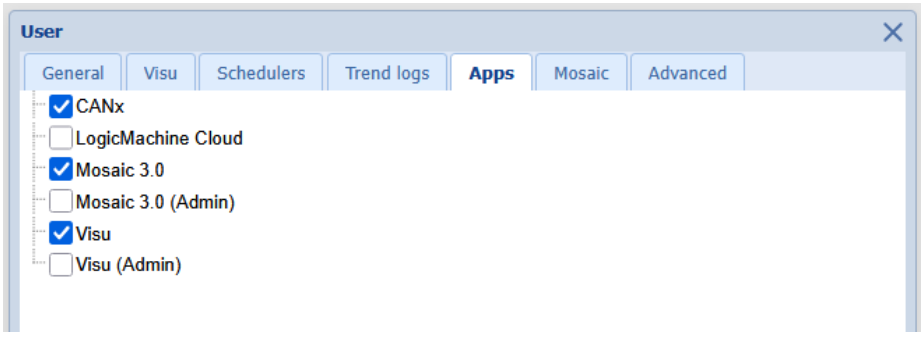
Access rights:

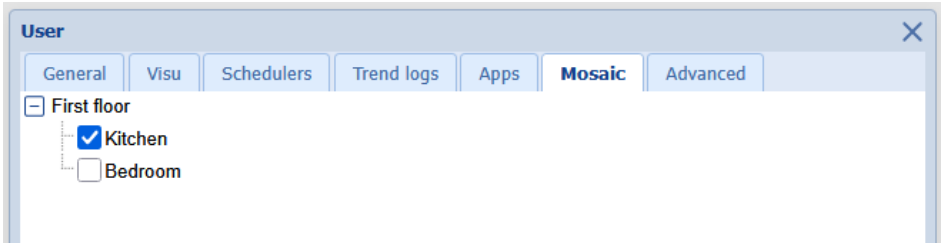
- *None* - no access
- *Partial* - access to specific elements is defined in a relevant tab
- *Full* - full unrestricted access


User
General
Visu
Schedulers
Trend logs
Apps
Mosaic
Advanced
Home
Kitchen
Office
Bedroom
Save
Cancel

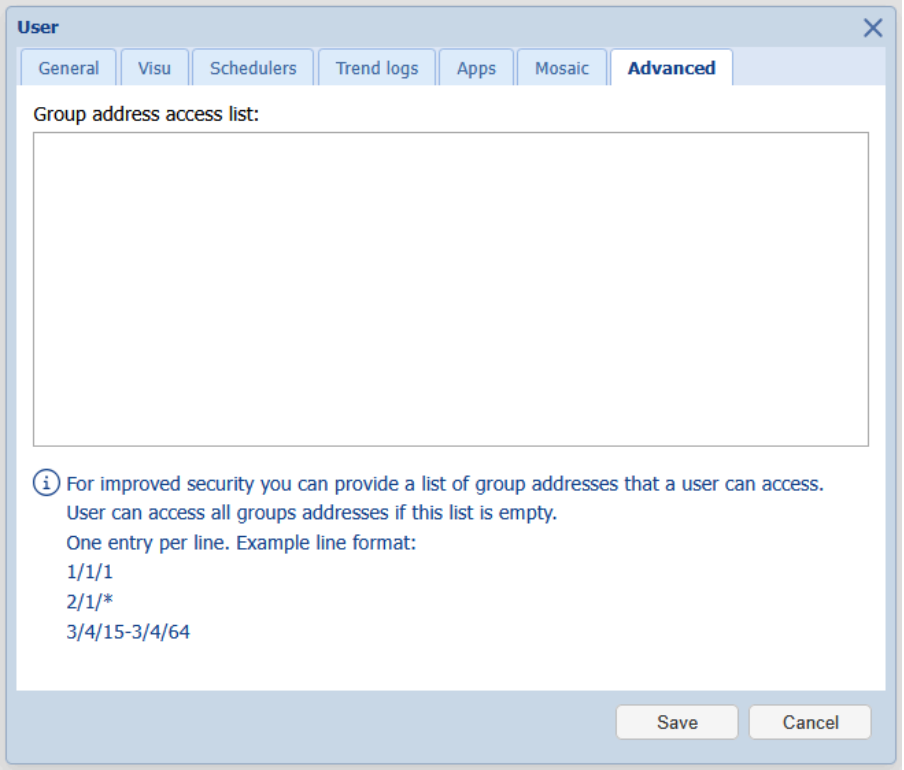

User
General
Visu
Schedulers
Trend logs
Apps
Mosaic
Advanced
lamp
Holidays


User
General
Visu
Schedulers
Trend logs
Apps
Mosaic
Advanced
Boiler Power


User
General
Visu
Schedulers
Trend logs
Apps
Mosaic
Advanced
CANx
LogicMachine Cloud
Mosaic 3.0
Mosaic 3.0 (Admin)
Visu
Visu (Admin)

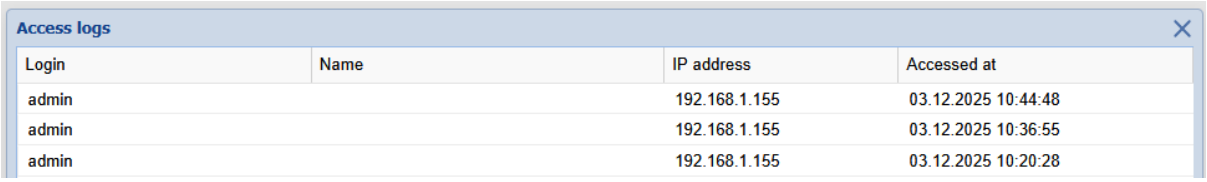

User
General
Visu
Schedulers
Trend logs
Apps
Mosaic
Advanced
First floor
Kitchen
Bedroom

Advanced tab allows specifying group address access filter for each user. This feature is recommended for improved security especially when a single LogicMachine is shared by multiple independent clients.



4.11.4. Access logs

Access logs contain information on successful and unsuccessful login attempts into the system. Login, IP address and access time is provided for all entries.



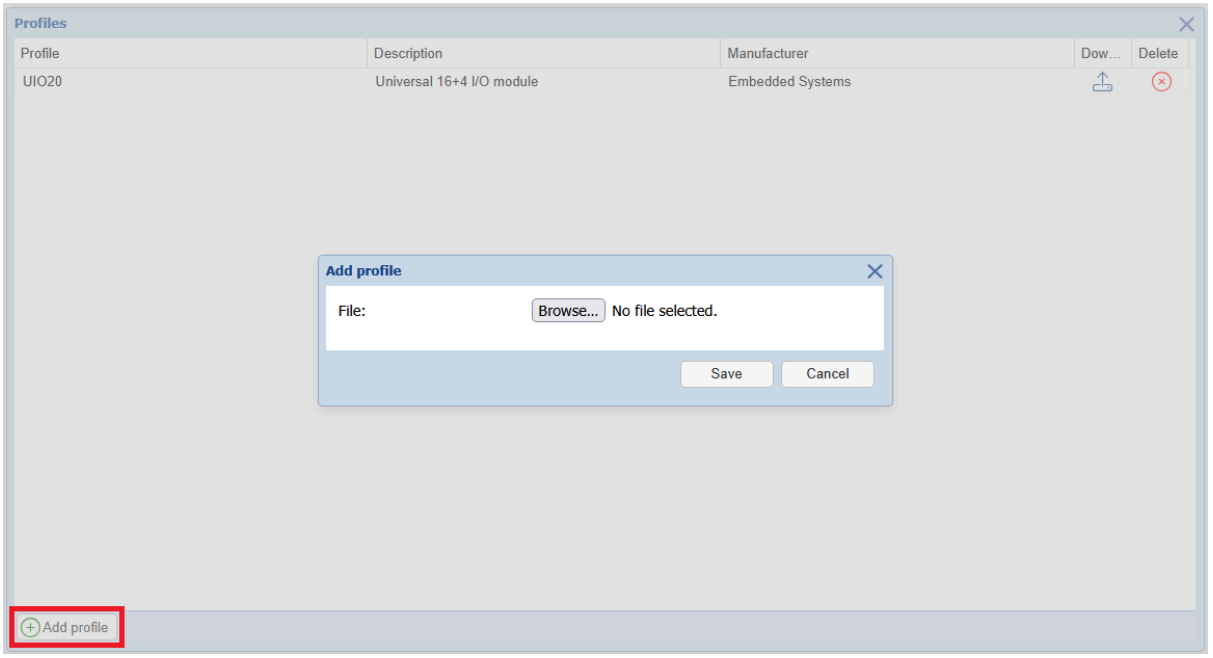
4.12. Modbus master (RTU/TCP)

4.12.1. Modbus devices profile

Each Modbus device requires a JSON profile which specifies the list of available Modbus points and their respective data formats.

Profile format description: kb.logicmachine.net/misc/modbus-profile/

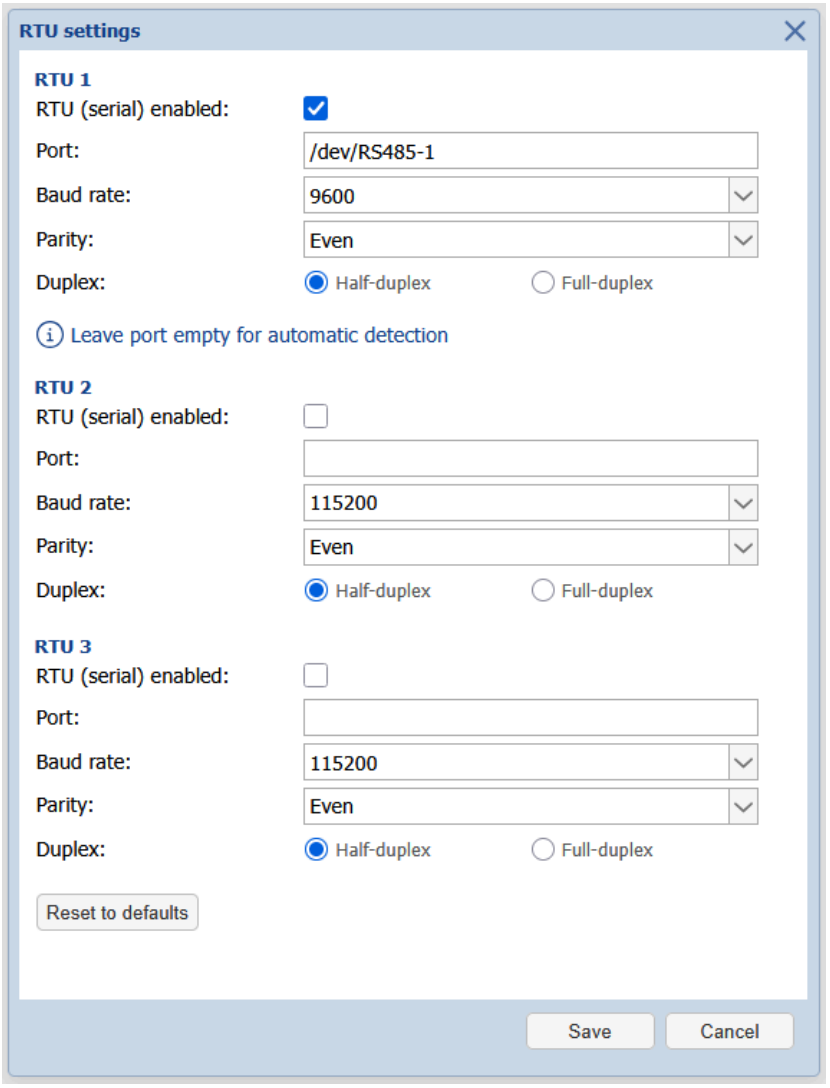
Click *Profiles* and then click *Add profile* to upload a new profile.



Note! Device entry must be created from scratch when an updated profile is added.

4.12.2. RTU settings

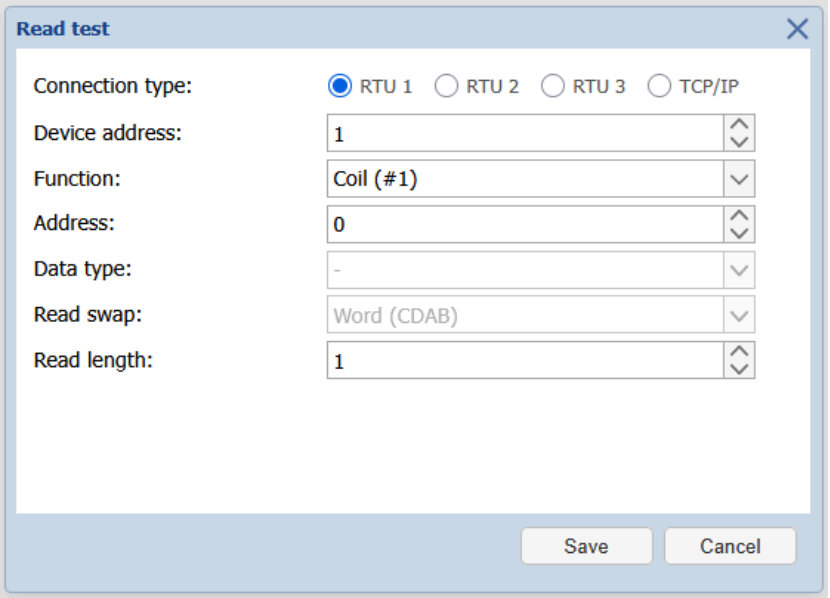
Up to three different RTU connections can be defined.



- *RTU (serial) enabled* - whether to enable this connection
- *Port (/dev/RS485-1; /dev/RS485-2)* - serial port name or leave blank for automatic detection
- *Baud rate (1200..500000)* - serial baud rate.
- *Parity (None 1 stop bit; Odd; Even; None 2 stop bits)* - parity/stop bits
- *Duplex* - either use half or full duplex communication

4.12.3. Read test

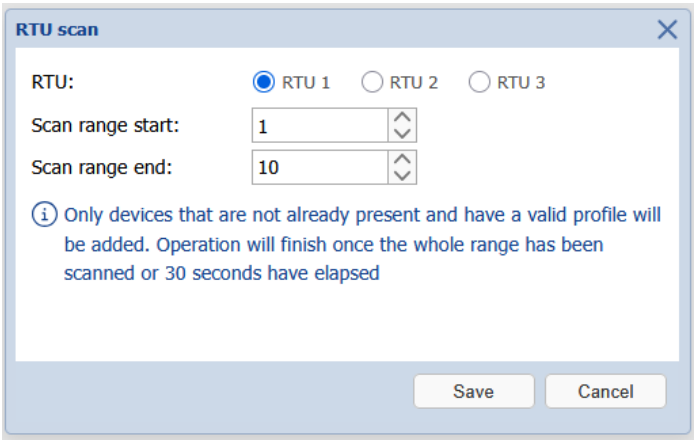
Click *Read test* to read values from a Modbus RTU/TCP device.



- *Connection type* - either *RTU 1*, *RTU 2*, *RTU 3* or *TCP/IP*
- *Device address* - Modbus device address
- *Function (Coil, Discrete input, Holding register, Input register)* - Modbus function
- *Address* - starting data address
- *Data type* - data type, only available for registers
- *Read swap (None (ABCD); Word (CDAB), Byte (BADC), Byte and word (DCBA))* - sets word/byte order
- *Read length* - number of registers/coil to read, disabled when a predefined data type is selected

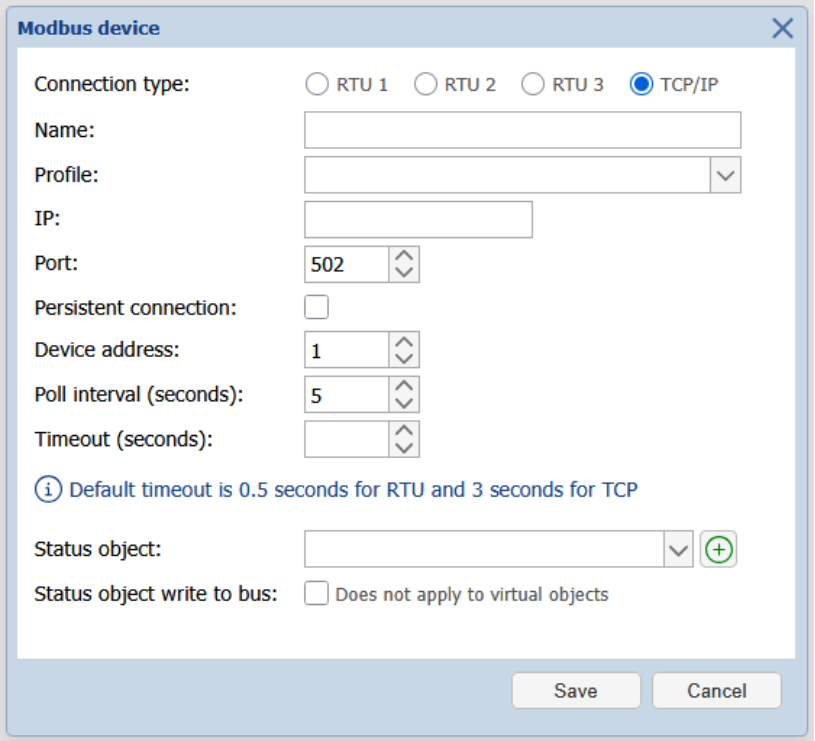
4.12.4. RTU Scan

Click *RTU Scan* to scan one of RTU ports for new devices in a selected address range. Only devices that are not already present and have a valid profile will be added. Operation will finish once the whole range has been scanned or 30 seconds have elapsed.



4.12.5. Adding Modbus device

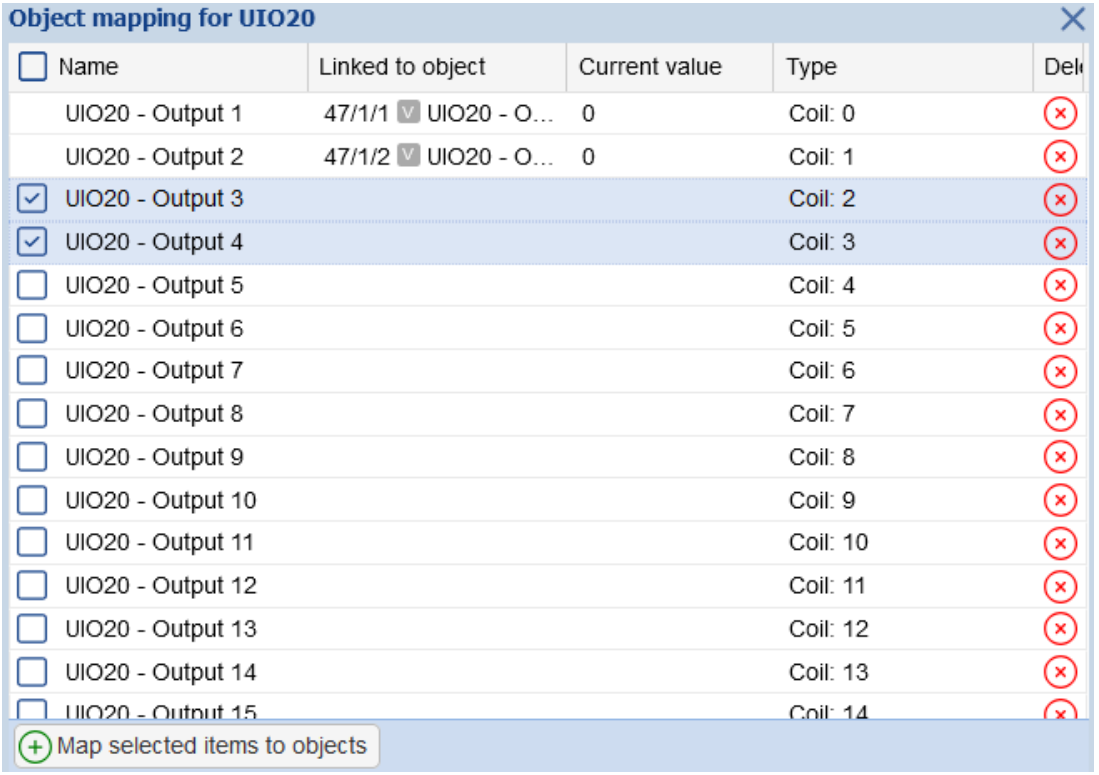
Click *Add device* to add a new Modbus device with a predefined profile.



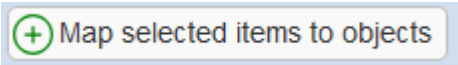
- *Connection type* - either one of RTU ports or a TCP/IP connection
- *Name* - device name
- *Status object* - status object of Modbus device (online/offline)
- *Write to bus* - whether to write status to KNX/TP
- *Profile* - device profile
- TCP/IP properties:
  - IP - IP address of the Modbus device
  - Port - communication port of the Modbus device
  - *Persistent connection* - when enabled the connection is not closed after each read cycle
- *Device address* - slave ID of the Modbus device
- *Poll interval (seconds)* - number of seconds between each read cycle
- *Timeout (seconds)* - time to wait for a reply from the device

4.12.6. Object mapping

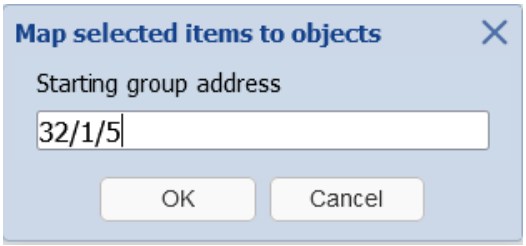
Clicking the  icon to map Modbus data points to objects.



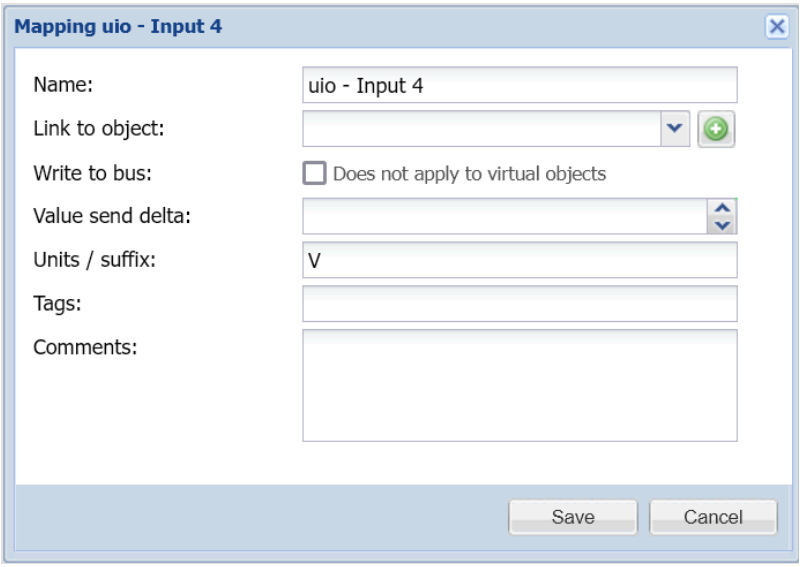
You can mass map the objects, by selecting the required data points and clicking



Select the starting group address. The data points will be mapped to objects using the same sequence as in the mapping table.



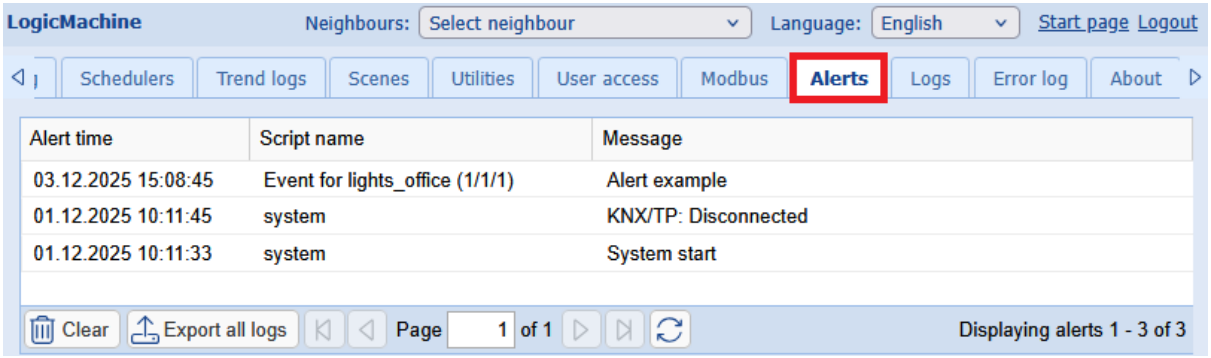
Click on a specific object to perform mapping and configuration.



- *Name* - mapping entry name
- *Link to object* - link this value to a new or existing object
- *Write to bus* - whether to write object value to KNX/TP bus
- *Value send delta* - (only applies to registers) value is sent when the difference between the current and previously sent values is larger than the defined delta. Leave blank to always send the new value
- *Units / suffix* - (only applies to registers) object units
- *Tags* - object tags
- *Comments* - entry comments

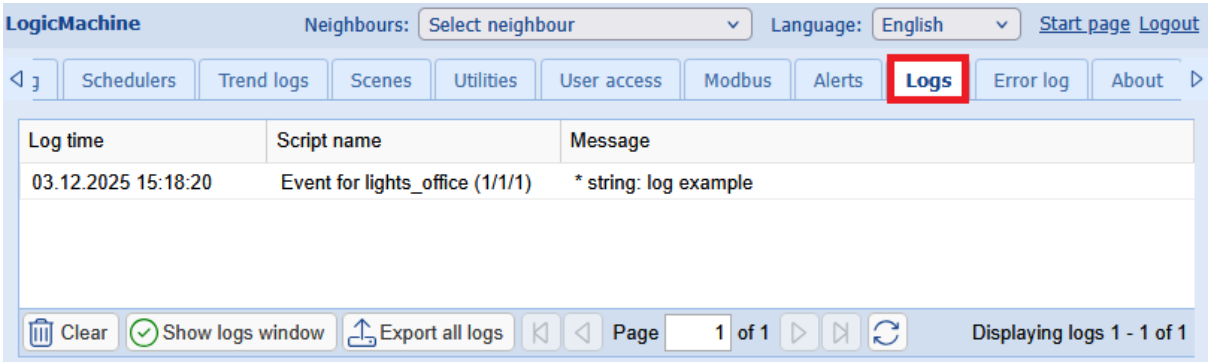
4.13. Alerts

Contains a list of alert messages from scripts (*alert()* function), KNX connection status messages and system start messages.

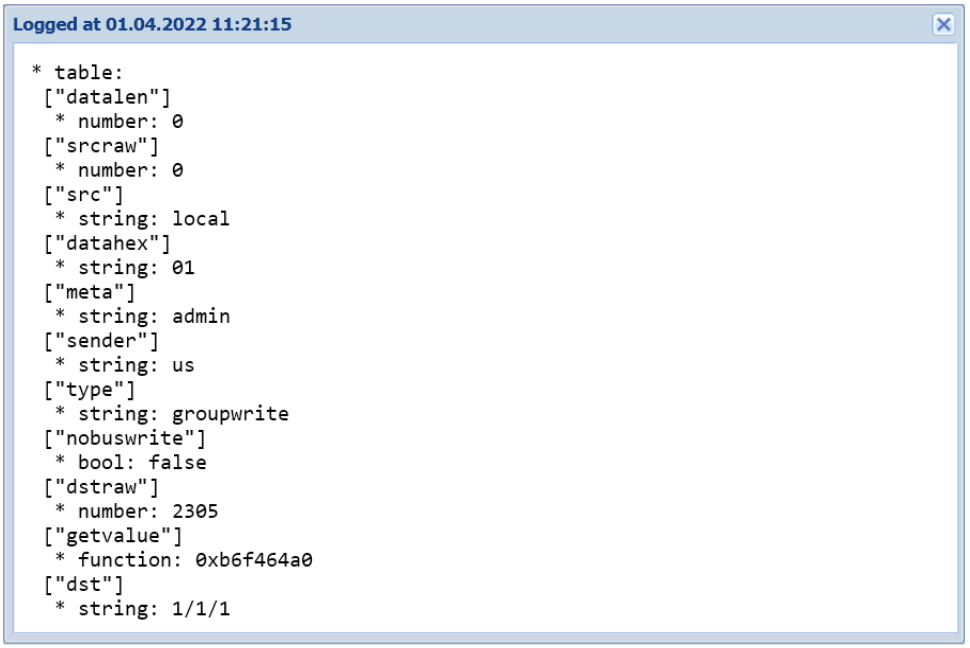


4.14. Logs

Contains a list of log messages from scripts (*log()* function).

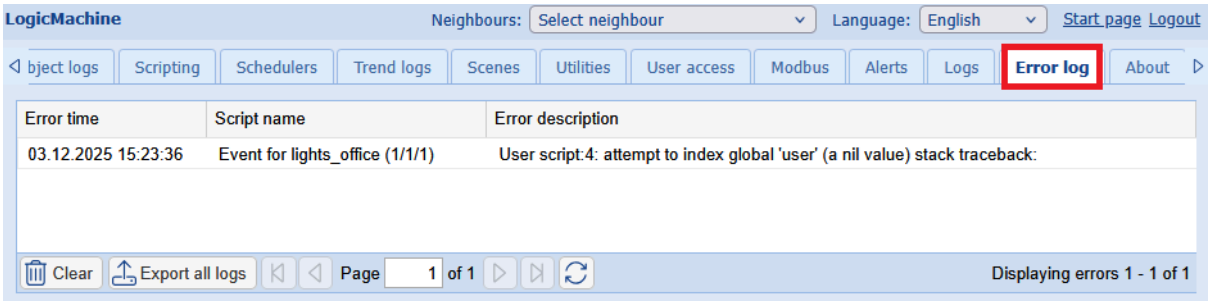


Click an entry to show a window with full log text in a readable format.



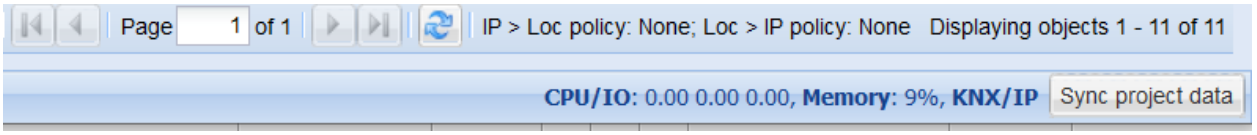
4.15. Error log

Error messages from scripts and applications are displayed in the *Error log* tab.



Click an entry to show a window with full error log text.

4.16. User Interface status information



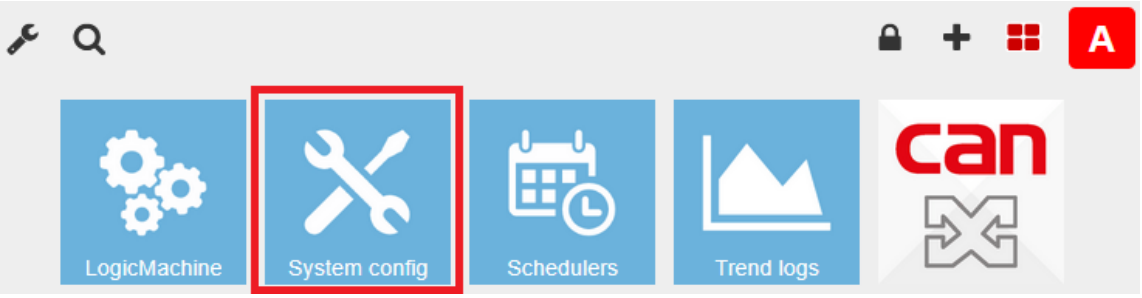
- *CPU/IO* - Load average. The load average represents the average system load over a period of time. It conventionally appears in the form of three numbers which represent the system load during the last one-, five-, and fifteen-minute periods. The lower a number the better.
  Note! Inspect your running tasks if the load exceeds a level of 0.70!

  More on UNIX style load calculation can be found here:
  http://en.wikipedia.org/wiki/Load_(computing)#Unix-style_load_calculation

- *Memory* - RAM usage in %
- *KNX/IP / KNX/TP* - type of the KNX bus connection. Current connection status is provided for KNX/TP mode
- *Sync project data* - save all project data to the internal storage. The project is synchronised automatically every 30 minutes, or when *Reboot* or *Shutdown* commands are executed
- *KNX/TP load graph* - shows average KNX bus load, click the graph to open full KNX/TP statistics

5. System configuration

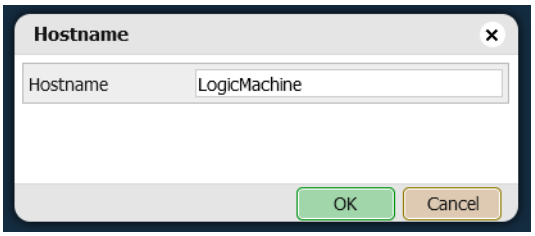
System configuration allows changing system parameters, configure services, upgrade firmware and check system status.



5.1. System

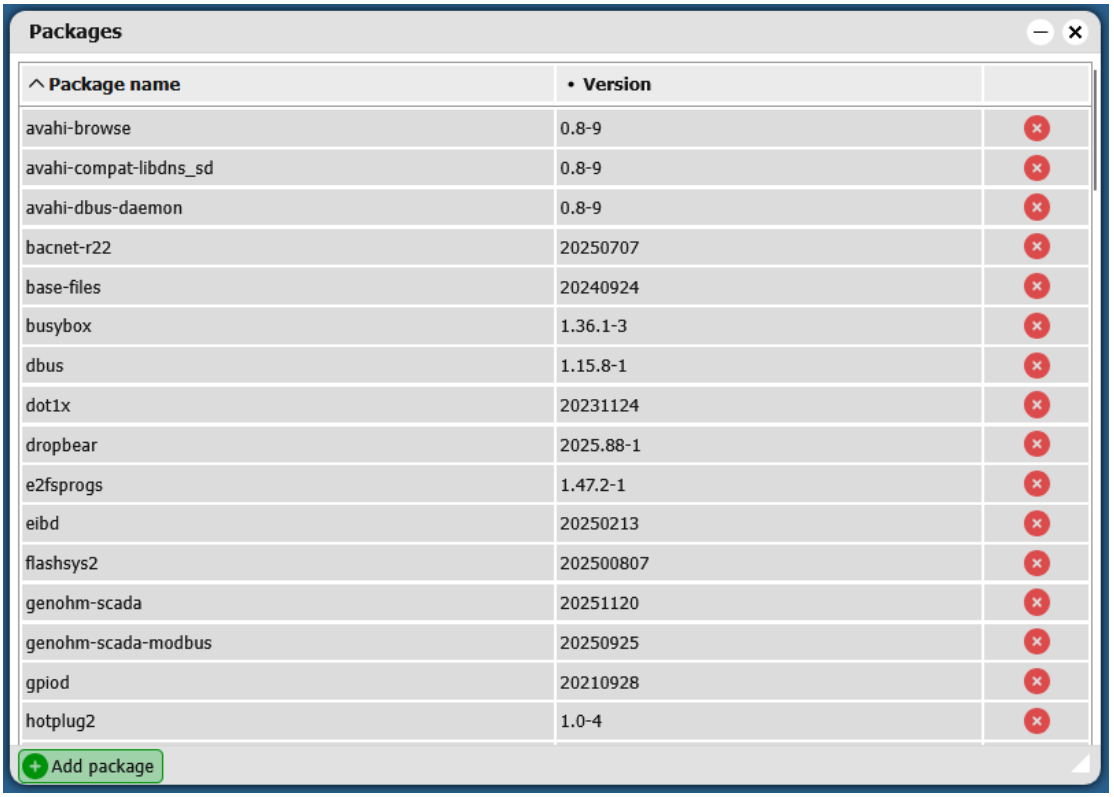
5.1.1. Hostname

Hostname can be changed in *System* → *Hostname*. This name is displayed in the browser title, it is also added to the backup file names. It appears when searching for the device using *LogicMachine* applications.



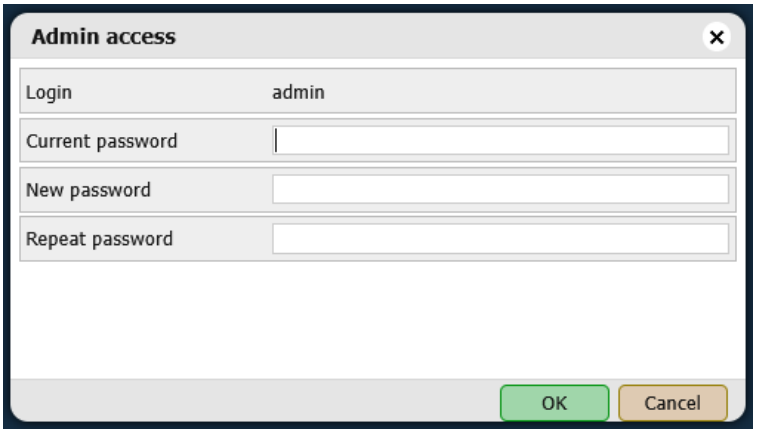
5.1.2. Packages

Displays a list of currently installed packages. Click  to add new packages.



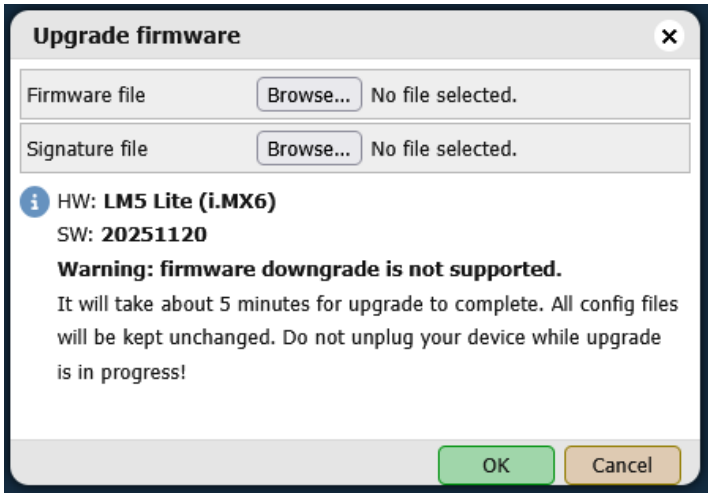
5.1.3. Admin access

Allows changing the *admin* user password.



5.1.4. Upgrade firmware

Used to perform a full system upgrade (both OS and LogicMachine parts).

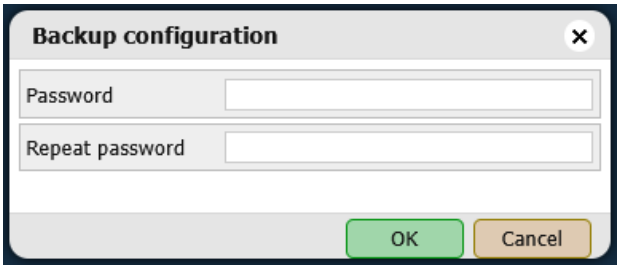


Note! It is recommended to perform a project backup in *LogicMachine → Utilities → Backup* before upgrading the firmware.

Make sure that the new firmware matches the hardware model that is being used.

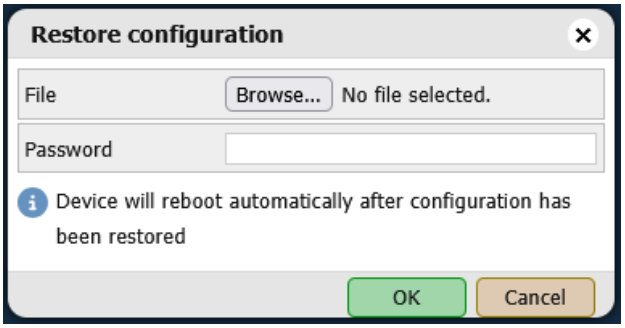
5.1.5. Backup configuration

System configuration backup contains all LM service configuration files (including password) and the KNX filtering table.



5.1.6. Restore configuration

Allows restoring a system configuration backup. Note that this might change the device IP address.



5.1.7. Reboot

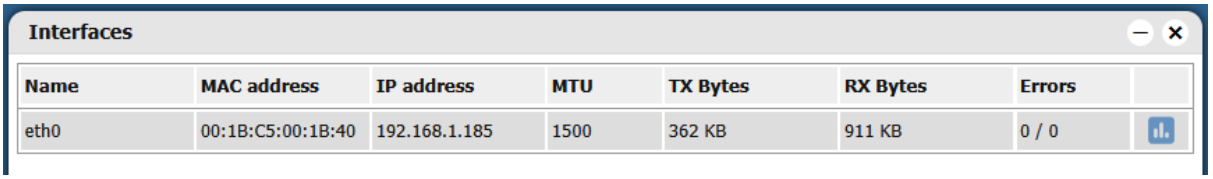
Performs project save to disk and reboots the system.

5.1.8. Shutdown

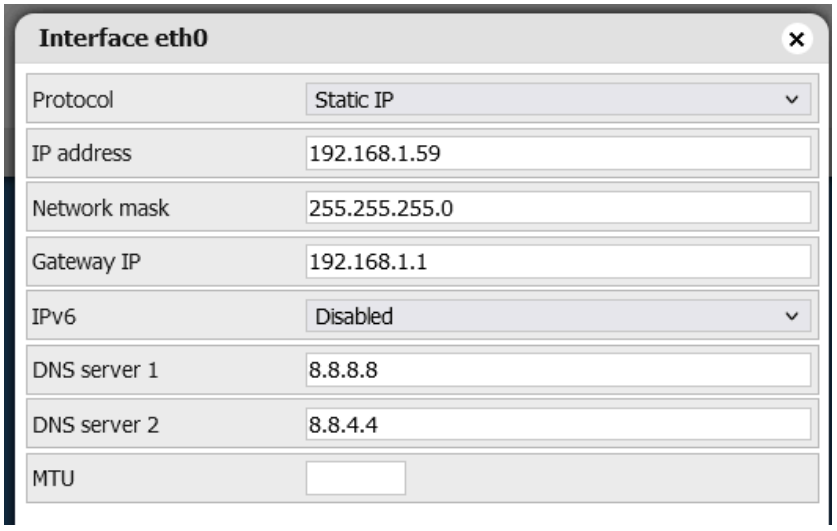
Performs project save to disk and shuts down the system. Power can be safely removed once LED1 stops blinking and LED2 turns off. To prevent accidental shutdowns the system will be restarted automatically after several minutes.

5.2. Network

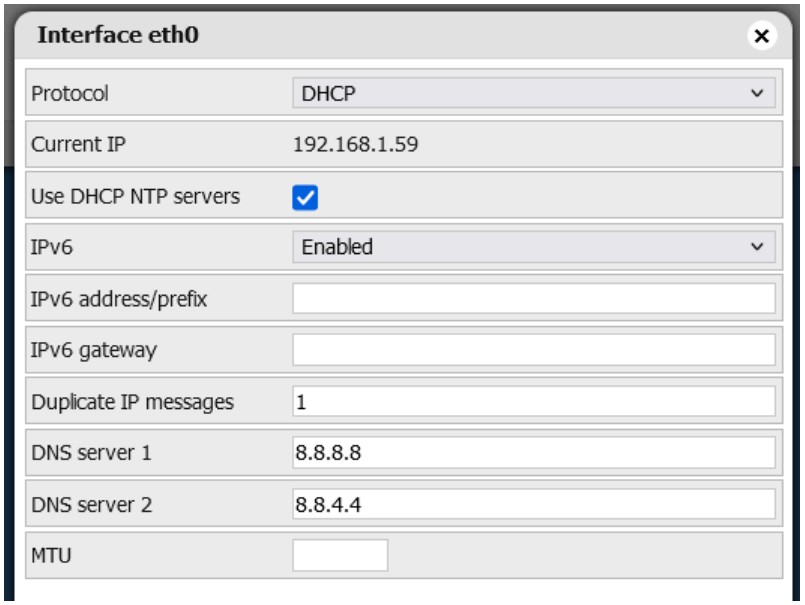
5.2.1. Interfaces



Click the interface name to change parameters.

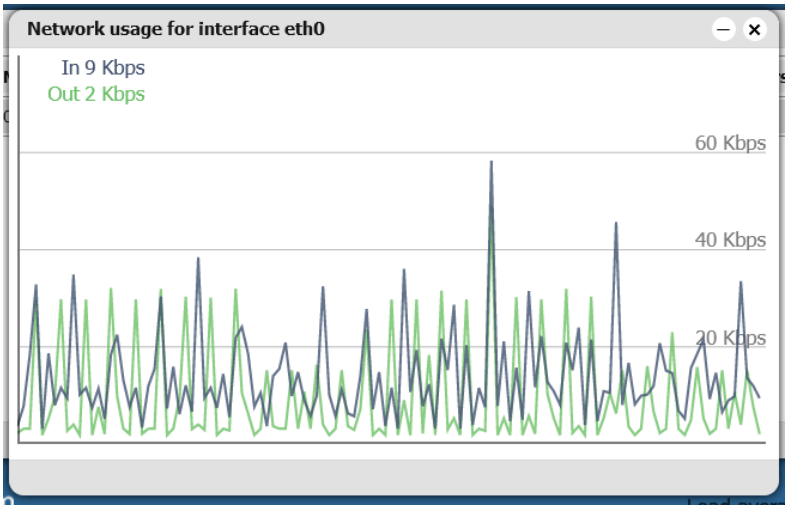


- *Protocol:*
  - *Static IP* - static IP address
  - *DHCP* - use DHCP protocol to get IP configuration automatically
- *IP address* - static IP address
- *Network mask* - network mask. (255.255.255.0 by default)
- *Gateway IP* - gateway IP address
- *IPv6* - enable or disable IPv6
- *DNS server* - DNS server IP address
- *MTU* - maximum transmission unit, the largest size of the packet which could be passed in the communication protocol (1500 by default)
- Use DHCP NTP servers - accept NTP server addresses from the DHCP server (only in DHCP mode)



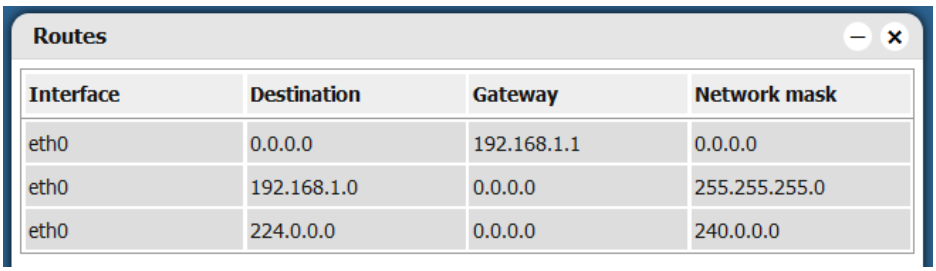
- IPv6 address/prefix - static IPv6 address and subnet prefix
- IPv6 gateway - gateway IPv6 address
- Duplicate IP messages - how many times the system checks for duplicate addresses before determining if the IPv6 address is unique

Click  to view a real-time graph of the interface traffic flow.



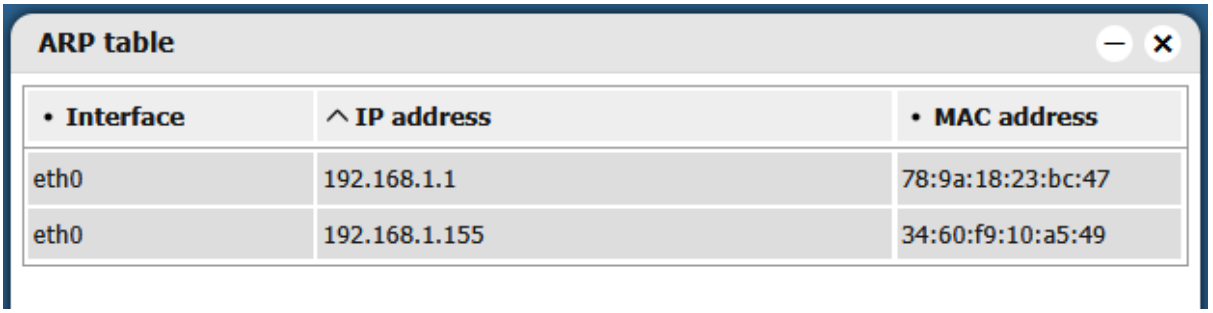
5.2.2. Routes

Displays a list of current network routes.

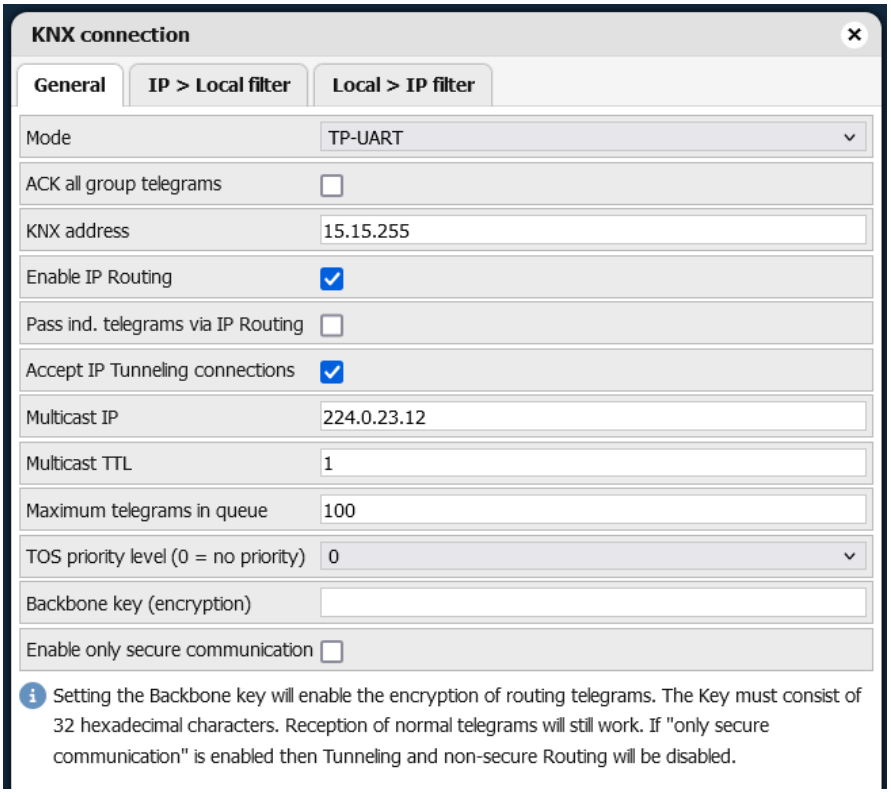


5.2.3. ARP table

Displays a list of known IP and MAC addresses.



5.2.4. KNX connection



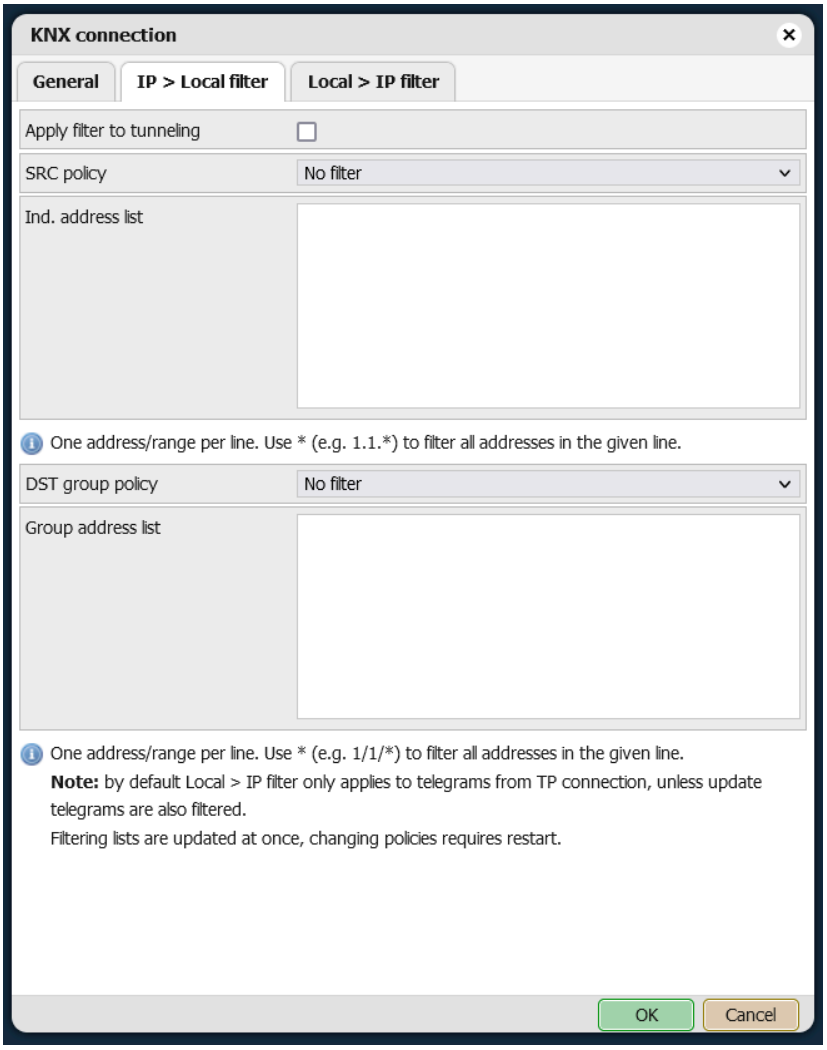
5.2.4.1. General tab

- *Mode* - KNX connection mode:
  - *TP-UART* - KNX/TP bus connection over a built-in TP-UART interface. KNX/IP communication is still possible with this mode when KNX IP features are enabled
  - *IP Routing* - KNX/IP Routing multicast mode with unacknowledged data exchange

    - *IP Tunneling* - KNX/IP Tunneling connection to an external router. Router IP address must be specified in this mode, router port can be added using IP:PORT format. This is a unicast mode with acknowledged data exchange
    - *IP Tunneling (NAT mode)* - same as *IP Tunneling* but allows connecting to a router outside of the LM subnetwork.
- *ACK all group telegrams* - acknowledge reception of all group telegrams received via KNX/TP
- *KNX address* - physical KNX address of the device
- *Enable IP Routing* - turns on routing of KNX group telegrams over IP
- *Pass ind. telegrams via IP Routing* - allows individual device-specific telegrams to be routed over IP; Only telegrams belonging to the same line are accepted (e.g. if LM address is 1.0.255 then only telegrams from the 1.0.x address range are accepted)
- *Accept IP Tunneling connection* - enables external clients to connect to the KNX bus via IP using tunneling
- *Multicast IP* - multicast IP address for KNX/IP Routing
- *Multicast TTL* - Time-To-Live for multicast telegram (maximum number of hops)
- *Maximum telegrams in queue* - maximum number of telegrams that can be queued
- *TOS priority level (0 = no priority)* - Type-Of-Service priority for KNX/IP telegrams, requires a router/switch with TOS support
- *Backbone key (encryption)* - backbone key for secured telegrams for KNX/IP Routing
- *Enable only secure communication* - disables KNX/IP Tunneling and non-secure KNX/IP Routing, only encrypted KNX/IP Routing is supported in this mode

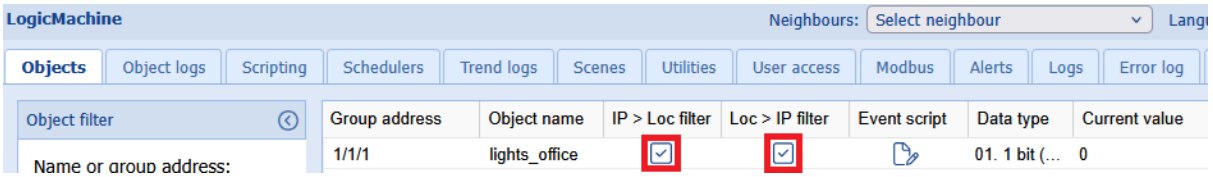
5.2.4.2. IP > Local filter

Filtering table for incoming telegrams from KNX/IP..



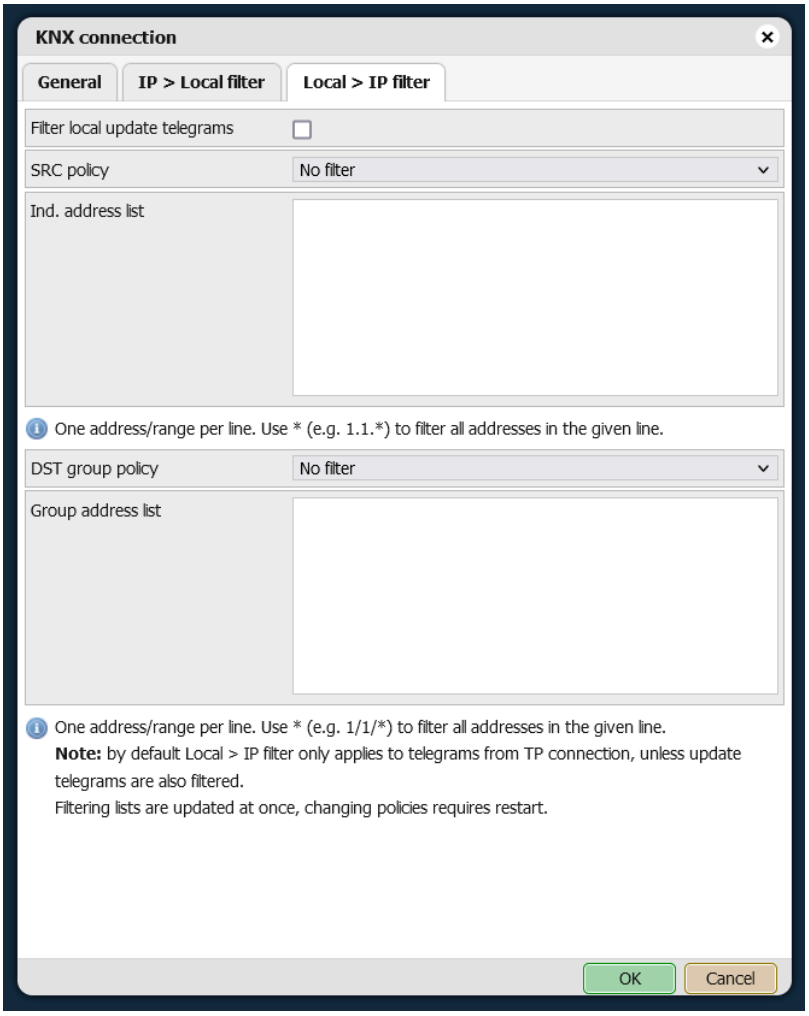
- *Apply filter to tunneling* - whether to apply filter policy to telegrams in tunneling mode. If ETS is used it is recommended to turn this feature off
- *SRC policy [No filter / Accept selected individual addresses / Drop selected individual addresses]* - policy for source individual addresses
- *Ind. address list* - list of individual addresses. One address/range per line. Use * (e.g. 1.1.*) to filter all addresses in the given line
- *DST group policy [No filter / Accept selected group addresses / Drop selected group addresses]* - policy for destination group addresses
- *Group address list* - list of group addresses. One address/range per line. Use * (e.g. 1/1/*) to filter all addresses in the given line

It is also possible to set filtering per-object in *LogicMachine → Objects* tab:



5.2.4.2. Local > IP filter

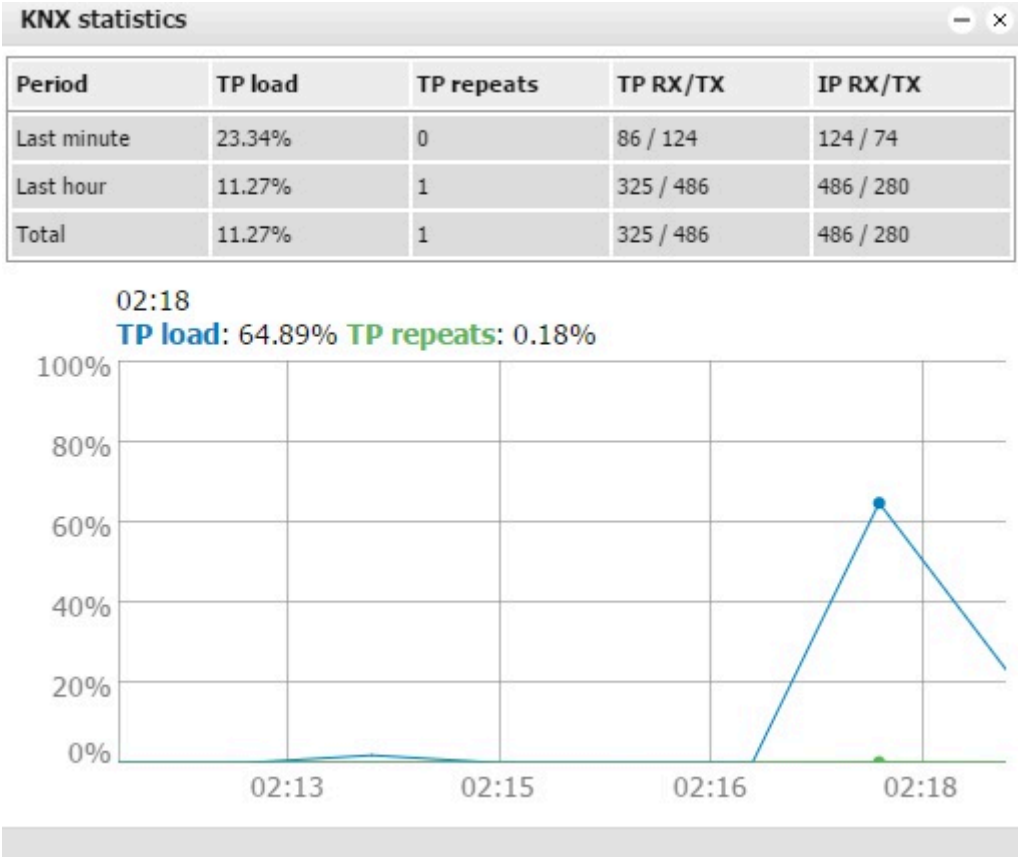
Filtering table for outgoing telegrams to KNX/IP.



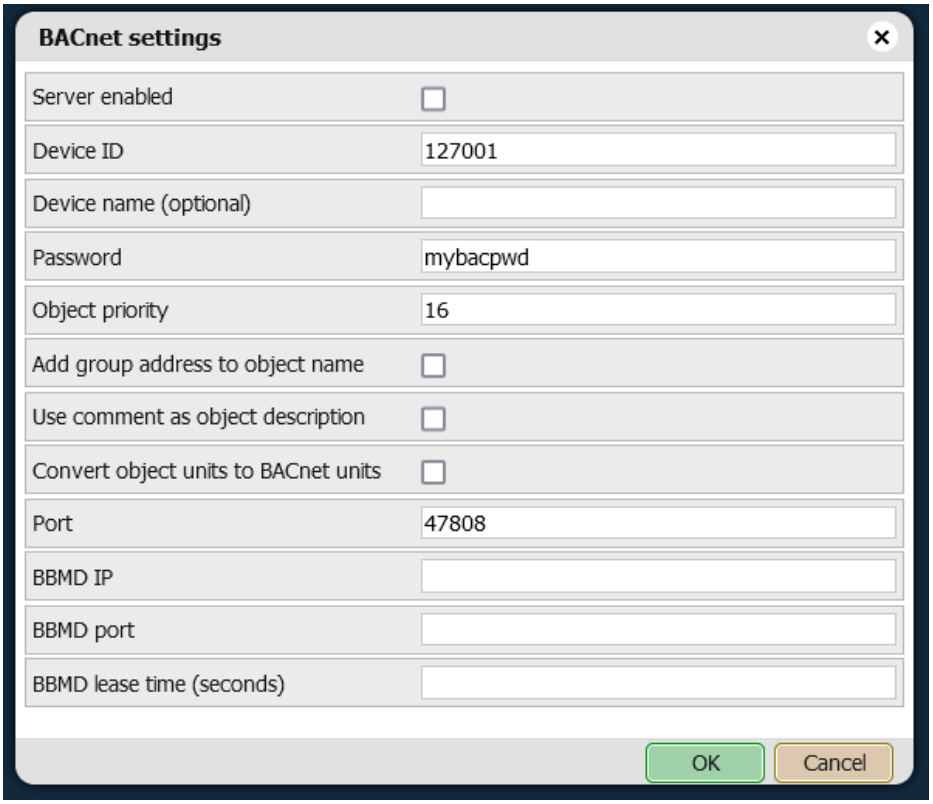
- *Filter local update telegrams* - whether to apply filter rules to update telegrams (called by grp.update() or when *Write to bus* is disabled)
- *SRC policy [No filter / Accept selected individual addresses / Drop selected individual addresses]* - policy for source individual addresses
- *Ind. address list* - list of individual addresses. One address/range per line. Use * (e.g. 1.1.*) to filter all addresses in the given line
- *DST group policy [No filter / Accept selected group addresses / Drop selected group addresses]* - policy for destination group addresses
- *Group address list* - list of group addresses. One address/range per line. Use * (e.g. 1/1/*) to filter all addresses in the given line

5.2.5. KNX statistics

Displays KNX/TP and KNX/IP statistics, including TP bus load, number of TP repeat telegrams and a number of sent and received KNX/TP and KNX/IP telegrams.



5.2.6. BACnet server settings

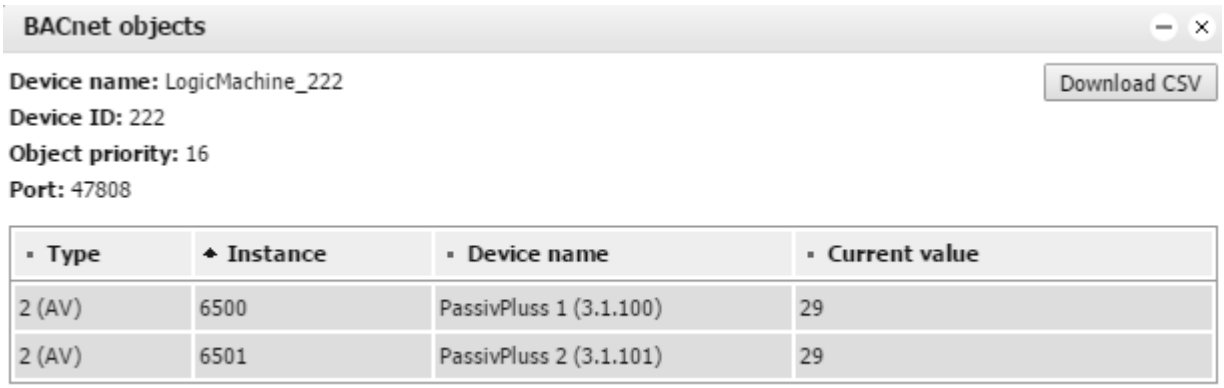


- *Server enabled* - whether the BACnet server is enabled
- *Device ID* - BACnet device ID
- *Password* - device password
- *Object priority* - priority array position for values that are written from any other source than BACnet
- *Add group address to object name* - append group address (X/Y/Z) to object names
- *Use comment as object description* - use object comment field value as BACnet object description
- *Convert object units to BACnet units* - whether to convert textual object units to BACnet units automatically
- *Port* - server port number
- *BBMD IP* - BACnet router IP. When a router IP and port are set, LogicMachine will act as a foreign device and will attempt to register with a BACnet router
- *BBMD port* - BACnet router port. When router IP and port are set, LogicMachine will act as a foreign device and will attempt to register with a BACnet router
- *BBMD lease time (seconds)* - registration resend interval

Only binary and numeric objects with *Export* enabled can be accessed via BACnet.

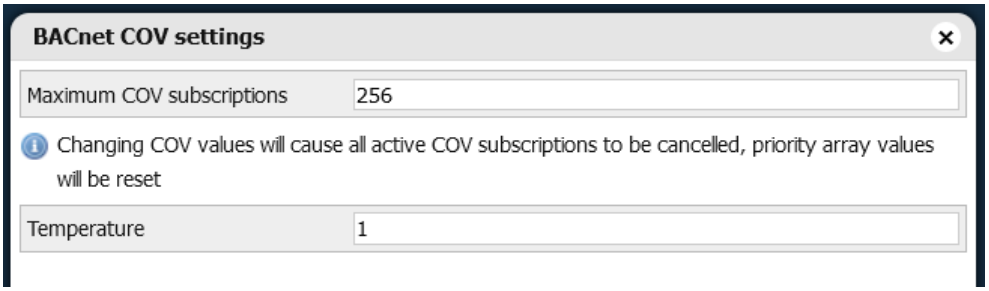
5.2.7. BACnet objects

Displays a list of exported BACnet objects. It is possible to download a CSV report containing all objects.



5.2.8. BACnet COV settings

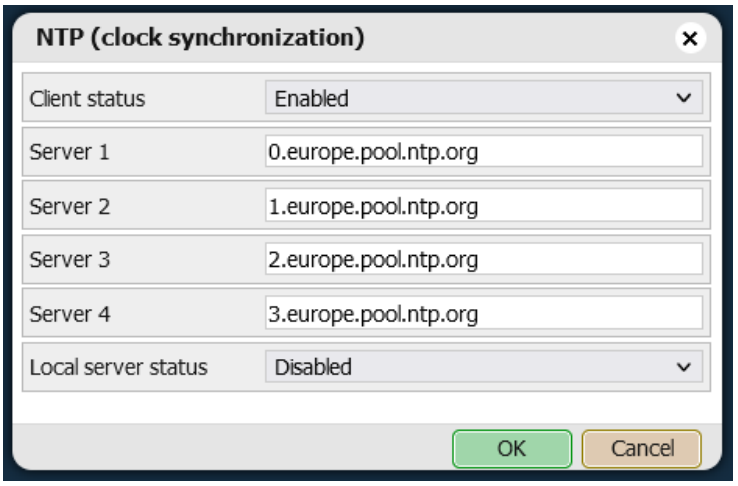
Change Of Value (COV) delta can be set for each numeric object. The maximum number of COV subscriptions can be increased up to 4096.



5.3. Services

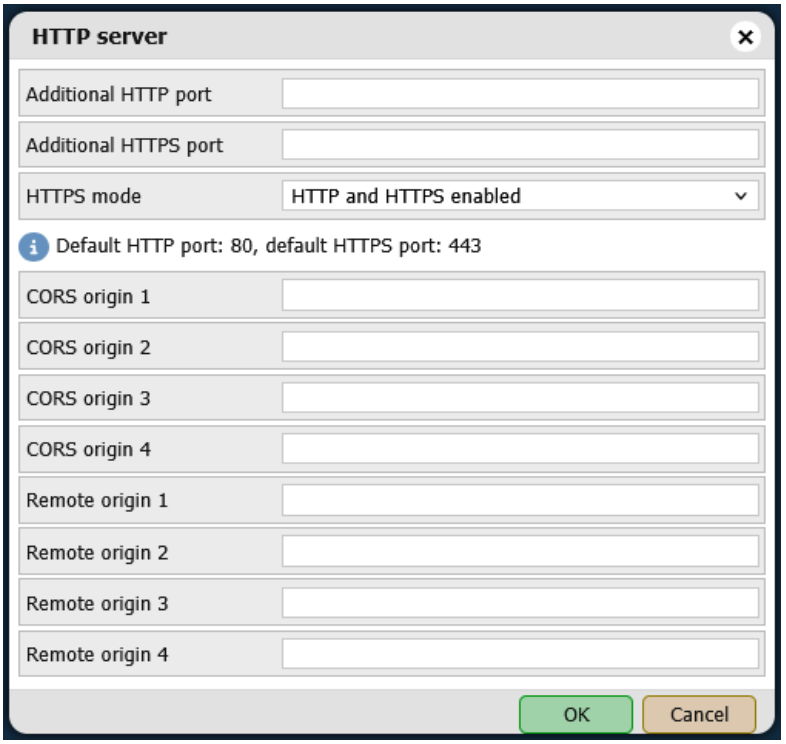
5.3.1. NTP client/server

Network Time Protocol (NTP) service synchronizes LM date and time with external servers. Up to four NTP servers can be specified. LM can also act as an NTP server for other devices on the same network.



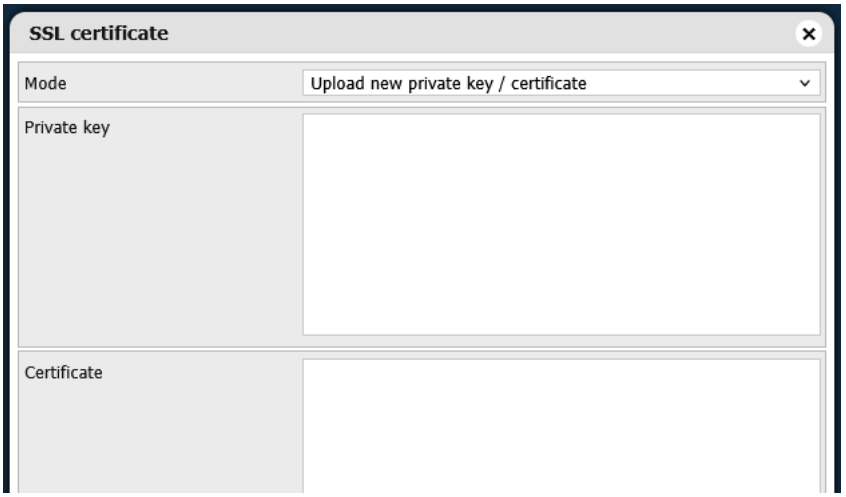
5.3.2. HTTP server

Allows adding an additional HTTP and HTTPS port. Unsecure HTTP can be disabled. Up to four CORS client IP addresses can be provided to allow certain applications to make external requests to the LM via HTTP(s).

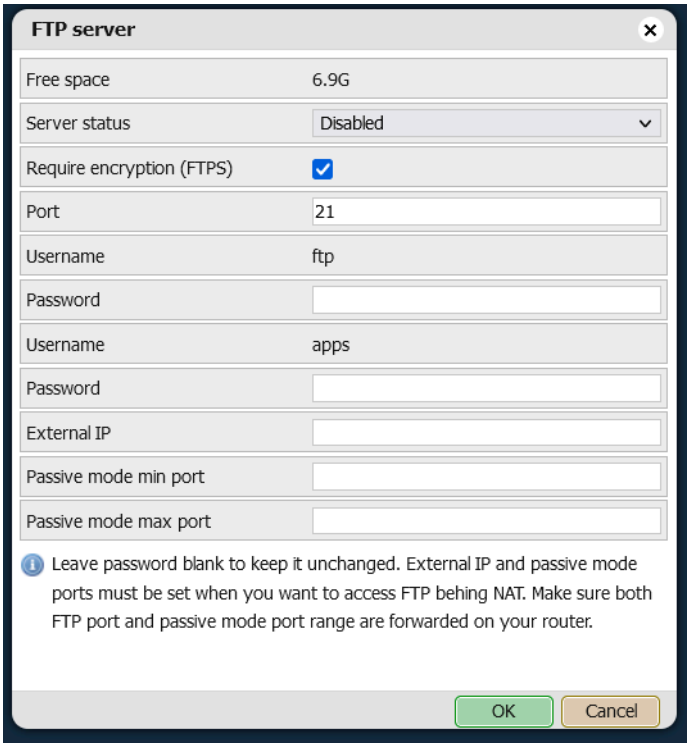


5.3.3. SSL certificate

Allows setting a custom private key and certificate. It is also possible to generate a new self-signed key/certificate pair.

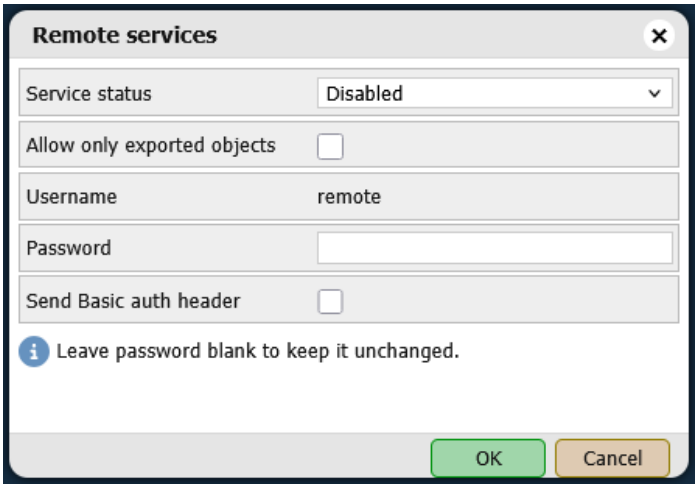


5.3.4. FTP server



- *Server status* - whether the FTP server is enabled
- *Require encryption (FTPS)* - whether to force FTP encryption
- *Port* - FTP server port
- *Password* - password for the *ftp* and *apps* users
- *Passive mode min port, Passive mode max port* - port range for external clients to use in passive mode

5.3.5. Remote services

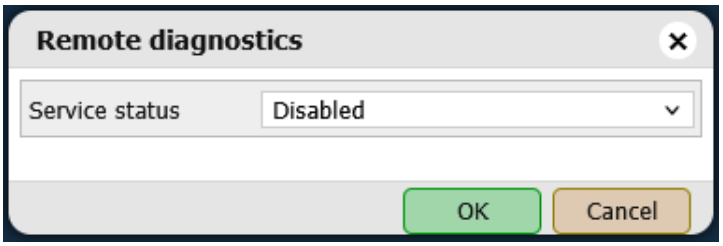


- *Service status* - whether the remote services are enabled
- *Allow only exported objects* - if enabled then only objects with *Export* mark can be access via remote services
- *Password* - password for the *remote* user

Request parameters and examples: kb.logicmachine.net/misc/remote/

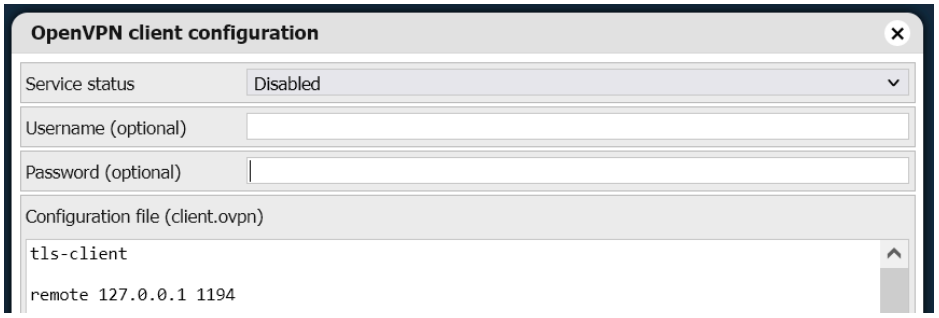
5.3.6. Remote diagnostics

Should only be enabled for remote support provided by Embedded Systems.



5.3.7. OpenVPN client

OpenVPN can be used for secure external access to your LogicMachine. An external OpenVPN server is required.



- *Service status* - whether OpenVPN client is enabled
- *Username / Password* - optional client credentials
- *Configuration file* - contents of *client.ovpn* configuration file

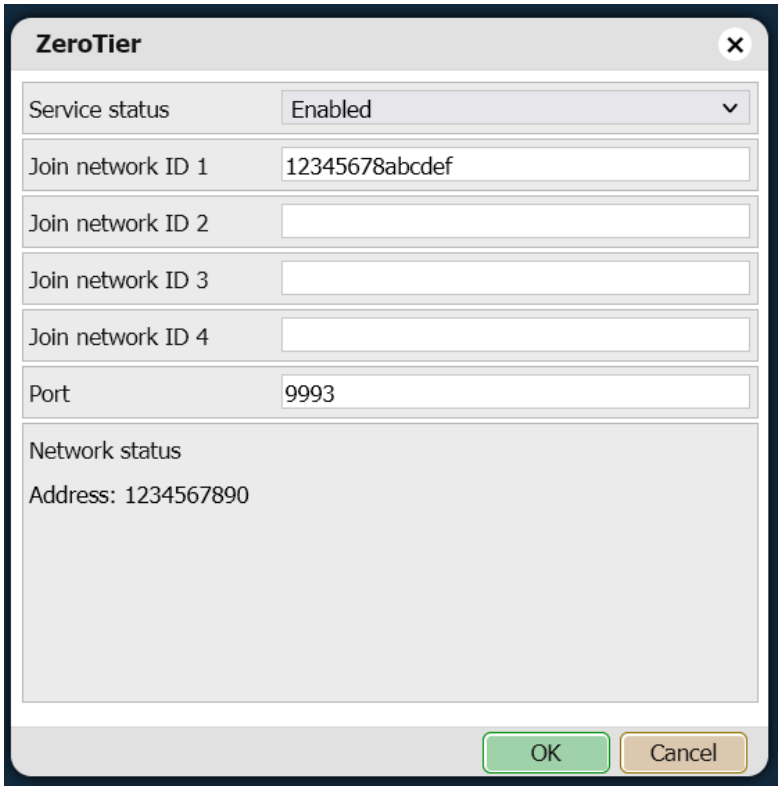
5.3.8. OpenVPN status

Displays the OpenVPN client connection logs.

5.3.9. ZeroTier

ZeroTier is a technology for secure external access to your LogicMachine.

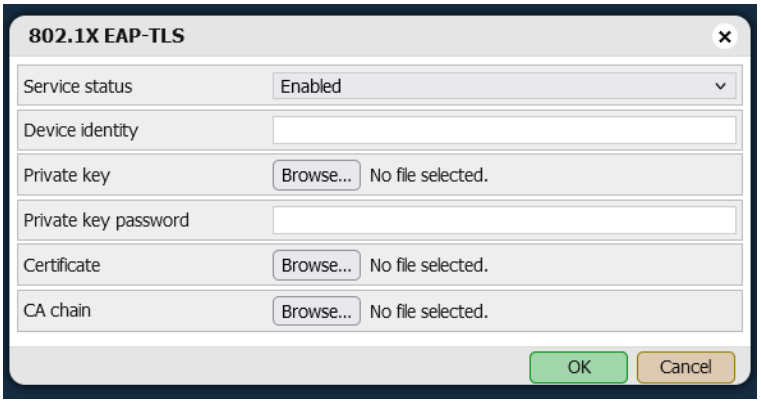
ZeroTier set-up example: kb.logicmachine.net/misc/zerotier/



- *Service status* - whether ZeroTier is enabled
- *Join network ID 1..4* - network IDs to join
- *Port* - UDP port to use
- *Network status* - LogicMachine ZeroTier address and a list of statuses for each configured network

5.3.10. 802.1X EAP-TLS

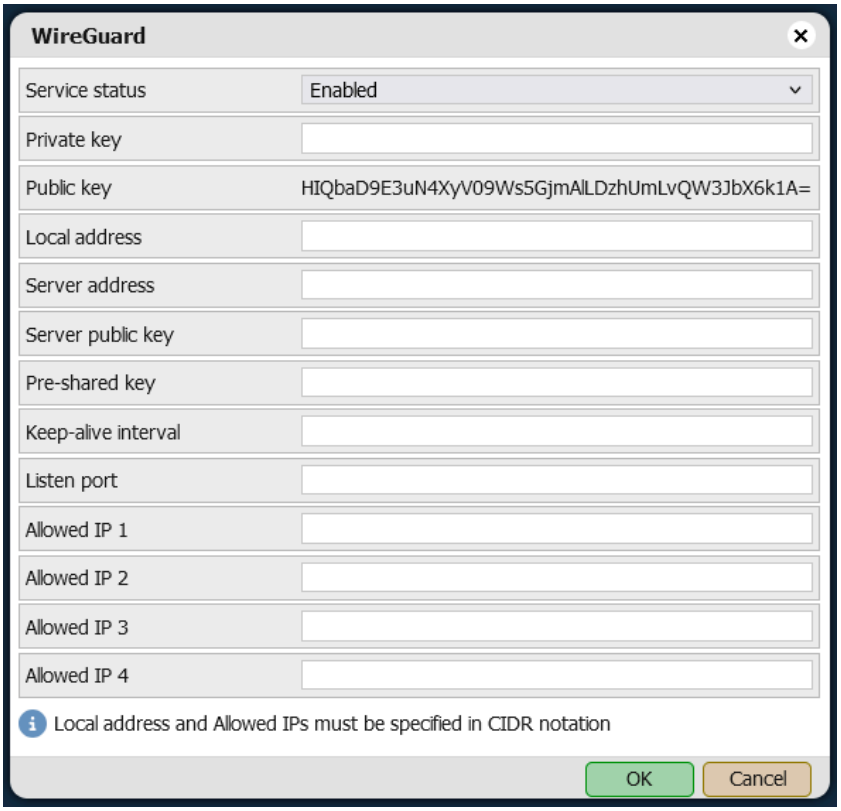
802.1x EAP-TLS provides certificate-based authentication for secure network access.



- *Service status* - enable/disable service
- *Device identity* - identifier that the client presents during the authentication
- *Private key* - client private key in PEM format
- *Private key password* - optional password to protect the *Private key*
- *Certificate* - client certificate key in PEM format
- *CA chain* - certificate authority (CA) certificate chain in PEM format

5.3.11. WireGuard

WireGuard is a highly secure VPN protocol.

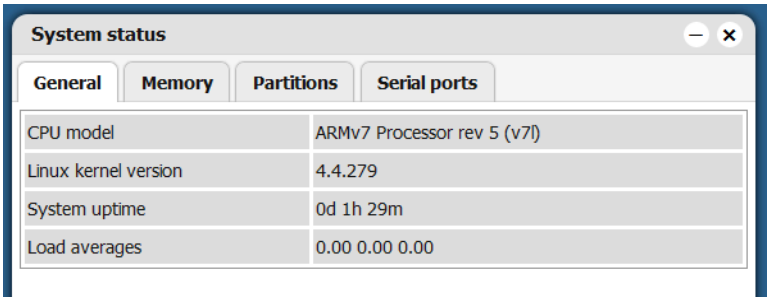


- *Service status* - enable/disable service
- *Private key* - private key for authentication and encryption
- *Public key* - public key, shared with peers for secure communication
- *Local address* - internal IP address inside the VPN network in CIDR notation
- *Server address* - public IP of the WireGuard server in host:port format
- *Server public key* - server's public key used for encryption data sent to the server
- *Pre-shared key* - optional additional layer of encryption between the client and server
- *Keep-alive interval* - frequency at which keep-alive packets are sent to maintain the connection
- *Listen port* - the UDP port number where WireGuard listens for incoming VPN traffic
- *Allowed IP* - single IP address or an IP address range in CIDR notation with which the client should be able to communicate

5.4. Status

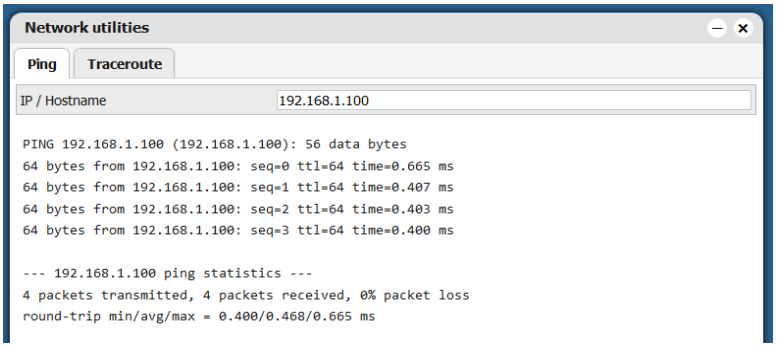
5.4.1. System status

Displays general system information including CPU usage, Memory, Partitions and Serial port list.



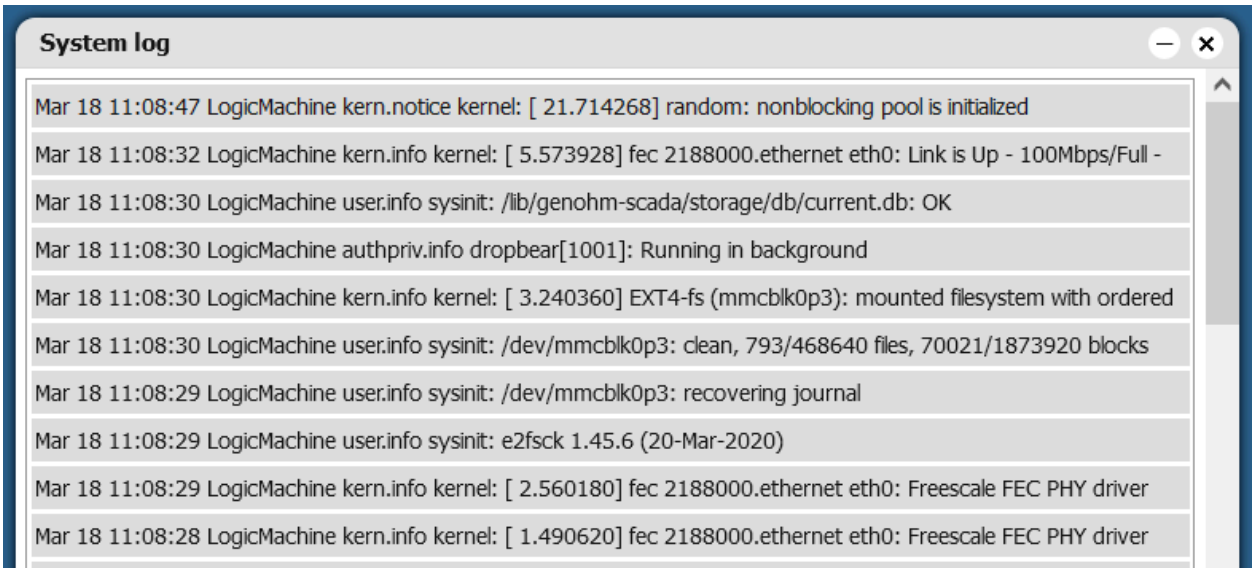
5.4.2. Network utilities

Ping and *Traceroute* utilities are available. Both IP addresses and DNS names are accepted.



5.4.3. System log

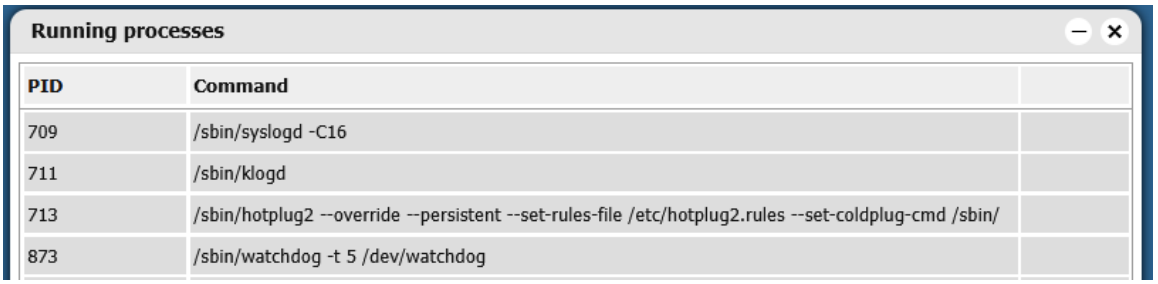
Displays the operating system log. Log entry data/time is in UTC format.

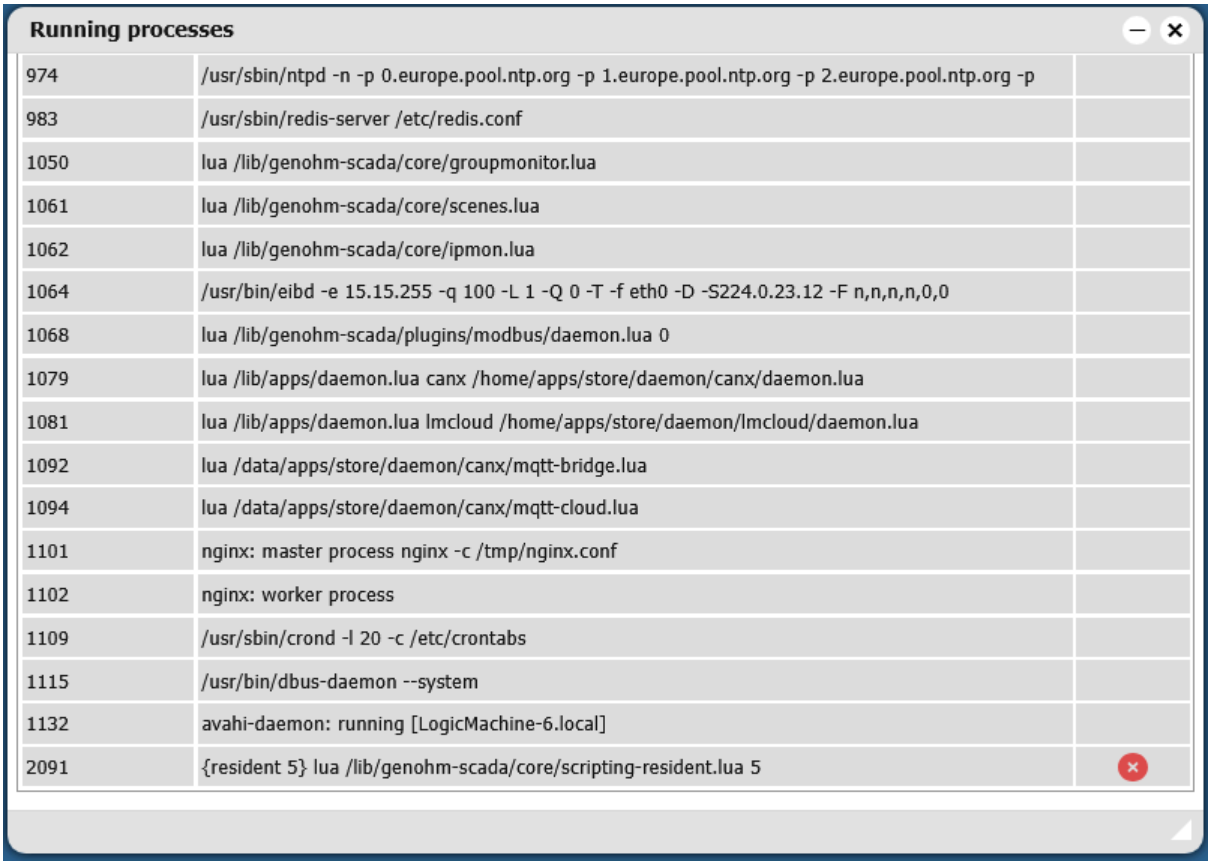


5.4.4. Running processes

Displays currently running system processes.

Only script processes can be stopped.




Running processes
974 /usr/sbin/ntpd -n -p 0.europe.pool.ntp.org -p 1.europe.pool.ntp.org -p 2.europe.pool.ntp.org -p
983 /usr/sbin/redis-server /etc/redis.conf
1050 lua /lib/genohm-scada/core/groupmonitor.lua
1061 lua /lib/genohm-scada/core/scenes.lua
1062 lua /lib/genohm-scada/core/ipmon.lua
1064 /usr/bin/eibd -e 15.15.255 -q 100 -L 1 -Q 0 -T -f eth0 -D -S224.0.23.12 -F n,n,n,n,0,0
1068 lua /lib/genohm-scada/plugins/modbus/daemon.lua 0
1079 lua /lib/apps/daemon.lua canx /home/apps/store/daemon/canx/daemon.lua
1081 lua /lib/apps/daemon.lua lmcloud /home/apps/store/daemon/lmcloud/daemon.lua
1092 lua /data/apps/store/daemon/canx/mqtt-bridge.lua
1094 lua /data/apps/store/daemon/canx/mqtt-cloud.lua
1101 nginx: master process nginx -c /tmp/nginx.conf
1102 nginx: worker process
1109 /usr/sbin/crond -l 20 -c /etc/crontabs
1115 /usr/bin/dbus-daemon --system
1132 avahi-daemon: running [LogicMachine-6.local]
2091 {resident 5} lua /lib/genohm-scada/core/scripting-resident.lua 5

6. Other examples

Various examples, 3rd party protocol integration and other useful applications can be found in our knowledge base and forums:

kb.logicmachine.net

forum.logicmachine.net