# Z-Way Manual

## A python based static controller for Z-Wave wireless networks, runs as software, in the cloud (Z-Cloud) or our of a hardware box (Z-Box)

Version 1.07.2011, beta

Version: 01.07.2011
Valid for software version: 1.2
Z-Wave-Certification Number for Z-Way: ZC08-11040003

## Table of Content

# PART I: General Information for Users

Z-Way is a fully featured home automation controller supporting Z-Wave as communication technology. It allows to

- Include and exclude devices and configure these devices, manage the network configuration and stability by visualizing the configuration and routing within the network
- Switch actuators such as electrical switches, dimmers, motor controls for sun blind, garage doors or venetian blind, door looks, heating thermostats and many more
- Access sensor data such as motion detection, temperature, CO2, smoke etc.
- Visualization of all functions of the Z-Wave network mapped to the floor plan or as tables simple to read
- Create logical connection between events created by sensors and actions performed by actuators
- Control Heating Schedules
- Realize automation of functions in the home using internal timers, schedulers, logical interconnection of events
- Execute scenes: This can either be a set of defined actions of certain actuators like dimmers or switches or even a script executed at certain times or under certain conditions. This allows to realize the most complex automation rules and a very powerful intelligent home or office,

# 1 Installation and Start

Z-Way is available in three different setups:

- A Stand alone software for PC running Linux, Windows or MAC OSX
- As a cloud installation (Z-Cloud), where the majority of the functionality is delivered by a server in a secure data centre and only a small pieces of hardware and software – the Z-Connector – is plugged into a PC or DSL router box locally availably.
- An all in one solution (Z-Box), where the software is pre installed on a energy efficient controller box installed locally in the home.

All three setups offer the very same user interface and functionality. Configuration work performed in one of the setups can easily be moved into different setups and vice versa.

## 1.1 Z-Box

A Quick Start Guide is part of the scope of delivery of the Z-Box. It has all necessary information to start up the Z-Box.

## 1.2 Z-Cloud

The web site z-cloud.z-wave.me offers all information needed to get started with Z-Way in the cloud setup. The Z-Connector software to be installed either on a PC or a DSL router is available after registration and is already preconfigured with all necessary certifications and configuration files.

## 1.3 Local startup

For the local startup of Z-Way on a PC please refer to the startup document provided together with the software.

# 2 Z-Wave Network Basics

## 2.1 Network and controllers

A Z-Wave network consists of various devices interconnected by a wireless communication protocol. **Thanks to the Z-Wave standard products from different vendors can work together seamlessly.**

Another advantage of Z-Wave is their ability to act as repeaters and forward data packets between nodes not able to communicate directly over the air. This extends the range of a Z-Wave network and improves stability. In order to perform this packet routing and forwarding the particular node needs to be mains powered. Battery operated nodes can't act as repeaters.

Z-Wave differentiates between portable and static controllers to control other devices. **Portable controllers** change their location and they are battery powered. To allow long battery live time they are inactive most of the time and will only communicate with other devices during manual interaction (pressing a button)

**Static controllers** are installed on a fixed location. They are mains powered and therefore able to stay alive all the time to communicate with other devices.

This allows static controllers to automatically update and optimize a Z-Wave network without further user interaction. Z-Wave refers to this function as **SUC/SIS (Static Update Controller/Static ID-Server)** as a special enhanced function of a static controller.

Beside the differentiation between static and portable controllers Z-Wave also distinguishes different roles of controllers in a network. The first controller of a network - regardless if portable or static - is always called **primary controller,** while all further controllers – regardless if portable or static – are referred to as **secondary controllers**.

In a standard network configuration the primary controller will organize the network and include and exclude other devices. Other controllers can be included but they will act as a secondary controller not being able to include further devices.

The **special SUC/SIS function** of a static controller **enables all other controllers** in a network **to include further devices**. Therefore the presence of such a SUC/SIS is highly desirable and Z-Way will always act in this enhanced mode if the Z-Wave hardware supports this (Almost all current USB Sticks support this mode).

If the hardware does not support the SUC/SIS mode the network will run in the standard mode (only the primary controller can include and exclude) and Z-Way will try to turn on the SUC/SIS role at every other static controller which gets included.

Z-Way can be included as a secondary controller in other networks controlled by other Z-Wave controllers. Z-way will then follow the network setup of this other network.

The "Network Management" tab of the user interface tells whether the software works in standard or in SUC/SIS mode.

## 2.2 Inclusion and Configuration of various device types

Z-Wave devices can be mains powered or battery operated. The inclusion process for both devices types is similar, however battery operated devices need special handling.

### 2.2.1 Mains Powered Devices

A Mains powered device is easy to configure after inclusion since the device will receive all configuration commands and execute them immediately. Mains powered devices are always listening to other commands and can repeat commands to other nodes.

### 2.2.2 Battery Operated Devices

The main objective of a battery-operated device is to preserve the battery power and only use as much battery power as needed. Battery powered devices are therefore in a deep-sleep state most of the time. In deep-sleep state they are not able to communicate with other devices.

In order to communicate with other device the battery-operated device needs to be woken up and send to sleep mode right after the communication took place. To maintain a minimal level of "responsiveness" and to allow to configure and to use battery-operated devices Z-Wave offers three basic solutions:

1. Devices with wakeup intervals
2. Frequently listening battery devices
3. Devices with manual wakeup

#### 2.2.2.1 Wakeup Interval

Devices with wakeup interval will wakeup after a defined interval and send out a wakeup notification. Other devices such as the Z-Way controller are able to

communicate with this device and send out messages to this device (The controller know about the status of the battery operated device and will queue messages in a waiting queue.) After all communication is done the controller is supposed to send the device back into deep sleep.
If no communication happens the battery-operated device will go back into deep sleep mode after a defined time (typically some seconds – up to one minute).

The best practice of Z-Wave suggests that battery operated devices stay awake for a defined time right after inclusion and go into deep sleep mode afterwards. This first awake-time is device dependent and varies from 1 minute to one hour.

Hence it is recommended to configure the device right after inclusion to make sure the device is still alive. If the battery operated device is already in sleep state all configuration commands will be queued and executed after the next scheduled wakeup if and only if a valid wakeup interval was configured during the first part of the configuration process. If this was not the case the device needs to be woken up manually.

If the device went already into deep sleep before the configuration was finished it its recommended to wake up the device manually to speed up the configuration process. Otherwise the configuration will happen after the next scheduled wakeup.

Some battery-operated devices may not go into sleep mode at all after inclusion but need to be sent into deep sleep after configuration is finished. This is done by Z-Way automatically.

The configuration of the wakeup interval is a tradeoff between maximum battery life-time (suggest a very long wakeup interval with few wakeup cycles) and some responsiveness of the device in case of a network-reorganization. Typical values are 5min … 5 hours.

#### 2.2.2.2 Frequently Listening Devices

Z-Wave has introduced frequently listening devices (FLIRS). These devices will wakeup at least once in a second and try to receive a message. The trick is that the wakeup is so short, that on average the power consumption of FLIRS devices are low enough to allow battery life times greater than one year. FLIRS devices can be configured without any problems like mains powered devices since every command will always be received latest after one second.
However FLIRS device will not route other devices messages to preserve battery power.

#### 2.2.2.3 Devices with manual wakeup

Remote controls are battery operated as well but they are only awake if a button is pressed. Remote controls will not wakeup regularly to check for queued messages. Hence whenever a remote control is configured from the Z-Way

controller, the device needs to be woken up manually. Please refer to the manual of the remote control for further instructions how to wakeup the device.

## 2.3 Association versus Scenes

**The purpose of the Z-Wave network is to gather sensor information and allow switching of actors caused by manual interaction or as result of certain sensor events triggering or other automation.**

While the collecting and displaying of sensor values is a very straightforward process (include the device + access the sensor data on the relevant pages) the interrelationship between sensor events, manual interaction and switching of certain devices requires more setup work.

### 2.3.1 What is Association ?

To interconnect the different sources of switching events (sensors, remote controls, wall controllers) with real switches (power switches, dimmers, window blinds, thermostats) Z-Wave introduces the concept of **Association**.

An Association is a relationship between a sensor or controller and an actor with the result that any events triggered by the sensor or manually activated by a button on the controller results in sending a switching command to an actor or a group of actors.

Examples:

*In order to switch a simple power switch with a remote control, this certain button of a remote control needs to be associated to the power switch.*

*In order to close the window blind during darkness the luminescence sensor issuing a command when the light level changes below a certain level or raises above a certain level needs to be associated with a window blind.*

Z-Wave devices supporting association – means supporting to control other devices – offer a number of so called association groups**.** Each group refers to a single event within the device, which may cause sending out commands to other Z-Wave device. The user can assign certain actors to these groups and all members of these groups will receive a switching command if this group gets activated. Typically these groups are referred to a button (association group Button 1) or a special event of a device (association group for battery status messages or association group for triggering a door sensor)

The command used for associations is typically a simple “SET to Value X” with X as a definable value. This certainly limits the flexibility of associations but allows a simple and fast setup of interrelations between sensors, controllers and actors.

Z-Way allows to set associations after inclusion of the device and during normal operation of the network. Please refer to the section "Device Configuration" for more details.

### 2.3.2 What are Scenes?

A scene defines the desired switching statues of a series of Z-Wave devices. While devices in association groups will all receive a similar command and switch accordingly the scene can mix device dependent switching commands.

Example:

*"I am away" Scene: all lights in the home are off, but the outside light is on and the security system is activated.*

Scenes can't be defined in ordinary sensors but in controllers only, since there is a need to store a lot of device dependent information. Certain remote controls allow configuring scenes. Z-Way can also define and activate scenes.

### 2.3.3 Association or Scenes

Both associations and scenes are suitable to define interrelationships between sensors, controllers and actors. They can be even mixed. However it is recommended to stay with one concept to make the setup of the network easier.

The following table gives some pros and cons:

| | Associations | Scenes |
|---|---|---|
| Easiness of setup | Very easy | More complicated |
| Switching by remote controls and wall controllers | Easy | May be complicated |
| Activated by timers | Not possible | Possible |
| Mix of different switching status | Not possible | Possible |
| Activated by web interface | Not possible | Possible |

The net-net of this comparison is that associations are easier to use but limited in their functionality. Scenes may be more complicated but give users much more flexibility and power to define interdependencies and automation of the Z-Wave network.

## 2.4 Network Stability

Wireless communication is a challenging task. Therefore Z-Wave has implemented quite a few approaches how to make communication over the air more reliable.

The most important of these approaches is the so called meshing, which means that every mains powered node is able to store and forward packets on behalf of other nodes in case there is no direct wireless communication between transmitter and receiver.

To allow meshing in a network the controller needs to know a map of all available links between nodes. A link between nodes is referred to as stable wireless communication between these two links. Knowing all these links the controller can calculate routes (ways) to communicate to devices and inform other devices accordingly (an association set will cause two nodes to communicate with each other without using the controller).
The routing within a Z-Wave network is based on fixed predefined routes. Each packet, which is supposed to be sent using other nodes, has the full information about the desired route in the packet header. It is not possible to change this route on the fly.

While this sounds quite inflexible this is in fact quite smart. It significantly reduces traffic over the air and attempts to follow wrong routes.

If the attempt to communicate with a certain device over a given route fails (no acknowledge received back) the controller will try two more times the same route and then a number of other possible route to reach the device. This is also very smart because the route may be intercepted by a failed or otherwise not available node but the final destination is still alive and working well.

If the communication attempt fails the controller will try again and again to communicate with the desired target always creating a lot of traffic by testing other the available routes. This behavior is still desired but the increased traffic may slow down other communication in the network and will particularly keep the controller chip busy. A result of this traffic is the delayed executions of other wireless commands such as turning on a light.

The controller will stop to try communicating to this device at a certain time, which depends on the size and complexity of the network. The controller will mark the device as failed and ignore any further communication request to this link unless receiving an unsolicited message from this disappeared device.

The worst-case scenario is that the device is only reachable occasionally since this will always motivate the controller to try as hard as he can to reach the device that again causes a lot of traffic and delay for other command executions.

There are a couple of best practices to minimize these traffic overhead and keep the network in a stable status with minimal wireless traffic and minimal response time to wireless commands sent.

1. **Exclude devices which are not longer needed or which are moved outside the network**. If you take one device - e.g. a wall plug- and bring him outside the network, you need to exclude him from the network. Otherwise this device is not reached anymore and will create overhead traffic until it's marked as failed.
2. If a device is obviously **failed or broken, remove** it using the "Remove Failed Node" function in the "Network Management" tab.
3. If a device is moved within the network you need to **start network reorganization**. This process asks each node to detect its neighbors and reports the list of neighbors back to the controller. This used to update the routing table and recalculate the best routes to the devices.
4. Try to **avoid longer routes**. Check the routes between two nodes using the "routing table" tab and refer to the advice giving in the manual chapter "Routing Table".
5. **Avoid "shaky" links**. The tab "Communication Timing" in the expert's mode in tab "Routing table" gives valuable information about the quality of wireless links.
6. **Reduce Polling intensity**. On default a script " Polling devices" in the network zone will be called every minute and poll a list of command class if they are available on the device. This will create increasing overhead if the network grows. Make sure only to poll what it absolutely needed.
    a. You may want to increase polling interval from one minute to 5 minutes or so.
    b. Don't poll FLIRS devices and don't try t poll devices that are marked as "failed".
    c. Try to enable pushing of sensor values wherever possible. Most metering devices (power, temperature) allow to be configured so that they send sensor updates frequently or when changes occur, Make heavy use of these functions and limit the polling of the corresponding command classes
    d. Meter command classes reporting accumulated values do not need to be polled so often.
    e. If there is already one devices class polled delivering the status of a device – e.g. switch binary command classes for a binary switch – there is no need to poll additional command classes - e.g. the basic command class- to get the very same value.

**Network reorganization is also a good prevention** practice and it is recommended after any change of the network (include device, exclude devices remove failed nodes, move nodes,) it takes a couple of minutes and updates the routing table. Please be aware that changes in the environment such as new furniture may also change the wireless communication environment. A regular network reorganisation is therefore a good practice to keep the network healthy and stable.

## 2.5 Network Configurations with Z-Way

Z-Wave allows using a wireless network in different configurations. Each of these configurations has its own pros and cons.

### 2.5.1 Z-Way is used to build the new network

If Z-Way is used to build a new network the Z-Way controller will be the first node of this network and will most likely have the node ID 1. If the Z-Wave transceiver hardware supports the SUC/SIS function Z-Way will automatically become SUC/SIS controller and will organize the network. Every other controller included into the network will be a secondary controller and receive all network information from Z-Way.

If the firmware of the Z-Wave transceiver hardware does not support the SUC/SIS function Z-Way works the very same way but only Z-Way is allowed to include further devices. It is also not possible to create a new primary controller but Z-Way can hand over the primary role to a new included controller and become a secondary controller. In this role –however- Z-Way will not be able to receive updates about changes (inclusion/exclusion) of the network unless the new primary controller will serve in SUC/SIS mode.

In both modes there are no further controllers needed in the network.

The step by step description of the use is:

1. Start up Z-Way
2. Include and configure all devices using Z-Way or – in case Z-Way is SUC/SIS – any other inclusion controller in the network.
3. Use the network

### 2.5.2 Z-Way as installation controller only

Z-Way is used to setup the network but will be excluded later. Z-Way is used in the very same mode as when building the network. Once the network is finished, the SUC/SIS function needs to be turned off. Now Z-Way needs to include another controller as new primary controller into the network. Turning off the SUC/SIS function makes sure that Z-Way does not become SUC again during this operation but the new primary controller – either another static controller or a portable controller – may assign the SUC function to a different static controller in the network if available.

If Z-Way is used as installation controller only there need to be at least one further controller acting as primary controller once Z-Way leaves the network.

The step by step description of the use is:

1. Start up Z-Way
2. Include and configure all devices using Z-Way or – in case Z-Way is SUC/SIS – any other inclusion controller in the network.
3. Turn off SUC/SIS function
4. Include another controller as primary controller
5. Exclude Z-Way from the network using the new primary controller

*Attention: During inclusion and subsequent auto configuration of new devices Z-Way will put its own node ID into all association groups and will announce itself as the static controller to be informed by battery powered devices when waking up. If Z-Way departs the network all these settings become invalid. The user need to make sure that all associations and the node id setting of the wakeup command class is set correctly before Z-Way is excluded. It is recommended to turn off both SIC/SIS function and the auto configuration in order to avoid these problems. This can be done by editing the file defaults.xml.*

### 2.5.3 Z-Way is included into an existing network as secondary

Z-Way needs to be included into the existing network as secondary controller. If there is no SUC/SIS function in the network, Z-Way will try to become the SUC/SIS function. To avoid this the SUC/SIS function need to be disabled in defaults.xml prior to the inclusion. Otherwise the whole network will be informed about the presence of a new SUC/SIS and this may result in malfunctions of the network once Z-Way is excluded again. After all configuration work is done, Z-Way may be excluded from the network using the primary controller.

The step by step description of the use is:

1. Turn off SUC/SIS function in defaults.xml
2. Start up Z-Way
3. Include Z-Way into the existing network by the primary controller as a secondary controller.
4. Perform all configuration work
5. If you don't want to keep Z-way in the network exclude it from the network using the primary controller

*Attention: During inclusion and subsequent auto configuration of new devices Z-Way will put its own node ID into all association groups and will announce itself as the static controller to be informed by battery powered devices when waking up. If Z-Way departs the network all these settings become invalid. The user need to make sure that all associations and the node id setting of the wakeup command class is set correctly before Z-Way is excluded. It is recommended to turn off both SIC/SIS function and the auto configuration in order to avoid these problems. This can be done by editing the file defaults.xml.*

### 2.5.4 Z-Way is included into an existing network as primary controller

The actual primary controller needs to hand over the primary role to Z-Way during inclusion. If there is no SUC/SIS function in the network, Z-Way will try to become SUC/SIS. To avoid this the SUC/SIS function need to be disabled in defaults.xml prior to the inclusion. Otherwise the whole network will be informed about the presence of a new SUC/SIS and this may result in malfunctions of the network once Z-Way is excluded again. After all configuration work is done, Z-Way needs to hand over the primary role back to the original primary controller.

The step by step description of the use is:

1. Turn off SUC/SIS function in defaults.xml
2. Start up Z-Way
3. Hand over primary role from the actual primary controller to Z-Way by including Z-Way as new primary controller
4. Perform all configuration work

*Attention: During inclusion and subsequent auto configuration of new devices Z-Way will put its own node ID into all association groups and will announce itself as the static controller to be informed by battery powered devices when waking up. If Z-Way departs the network all these settings become invalid. The user need to make sure that all associations and the node id setting of the wakeup command class is set correctly before Z-Way is excluded. It is recommended to turn off both SIC/SIS function and the auto configuration in order to avoid these problems. This can be done by editing the file defaults.xml.*

# 3 Web Interface

## 3.1 Basics

The Z-Way web interface can be used in three different modes.

**1. Administrator Mode:** All administration functions are visible for inclusion, exclusion and configuration of the network. This function is designed for installers and technically interested end users to maintain the network

**2. User Mode:** Only a subset of all functions is shown for daily usage of the network. This mode is suitable for end users only.

**3. Experts Mode:** The experts mode is an enhanced version of the administrator mode. These functions are only needed for debugging purposes or for advanced configuration. The use of the expert mode is only recommended if you have extensive knowledge about wireless networks in general and Z-Wave networks in particular.

### 3.1.1 Workflows

The web interface is intended to allow the configuration and the daily usage of a Z-Wave network.

The GUI is used in five different contexts or workflows:

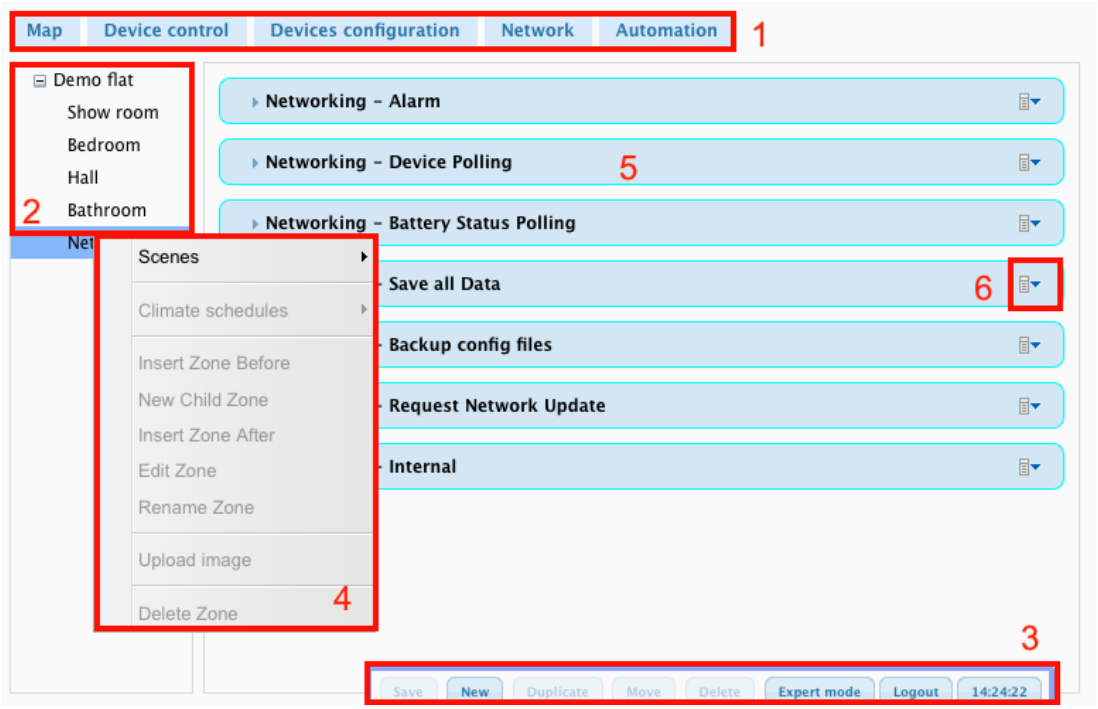
1. **Initial setup of the controller and the network**
2. **Initial setup and configuration of the devices**
3. **Automation setup and usage**
4. **Ongoing maintenance of the network**
5. **Daily usage by the end user**

The following table gives an overview of the different dialogs of the Z-Way web UI and refers to the sections of this manual where they are described:

| **Dialog** | **Described in** | **Also used in context** |
|---|---|---|
| Zones | 3.2 Initial Setup | Usage, Device Setup |
| | | |
| Network Management | 3.3 Device Setup | |
| Device Configuration | 3.3 Device Setup | |
| Scenes | 3.4 Automation | Device Setup |
| Rules | 3.4 Automation | Device Setup |
| Schedulers | 3.4 Automation | |

| | | |
|---|---|---|
| Routing | 3.5 Maintenance | |
| Device Status | 3.5 Maintenance | |
| Zone Status | 3.6 Usage | |
| Switches | 3.6 Usage | |
| Meters | 3.6 Usage | |
| Thermostats | 3.6 Usage | |
| Sensors | 3.6 Usage | |
| Heating Schedule | 3.4 Automation | |
| Expert command | 3.7 Debugging/ Expert Mode | |
| | | |
| Controller Info | 3.7 Debugging/ Expert Mode | |
| | | |

### 3.1.2 General Usage




The total screen estate can be divided into five sections:

**1 The function tabs**: Clicking on the function tabs leads to the different functions of Z-Way.

**2. Left hand side:** Here you either find a list of devices, zones etc. Clicking on these icons opens a dialog on the dialog pane (5).

**3. Bottom Context Menu:** The bottom context menu contains functions, which usually apply to all devices within the selected function tab.

**4. Right Mouse Click Context Menu:** Right clicking on different areas of the screen opens a context menu with functions related to the item clicked.

**5. Main Window:** This area context the main user dialog.

**6. Main Window Context Menu**: For some list items in the Main Window there is a drop down context menu. The very same context menu can be accessed by right clicking into this list item.

## 3.2 The initial setup

The initial setup is done one time only. But all settings can be changed later on.

Go to tab "Zones". Here you can setup your zones (rooms) in your home. There are already three zones:

- All: This is the root zone which includes all other zones. Try to avoid placing scenes etc into this zone but place them into the child zones they belong to.
- Network: This zone is special zone and not related to any real room. Its sole purpose is to keep some scripts that organize the network. You will find some predefined scripts when going to the tab "Scenes" and select the zone "Network".
- Home: This is the zone that is used to place devices and functions into. You may rename this zone and add more child zones to structure your home.

### (1) Load your floor plan

As first step you should replace the default zone map with your own floor plan. However you will be able to use Z-Way with the default floor plan as well.

Click the Button „Upload Image" on the bottom line context menu and choose the new image from your local computer. You may upload image in the following file formats: JPG, PNG, GIF.

All changes will be saved after hitting the save button in the bottom context menu. Make sure to save all your changes!

### (2) Define your zones/rooms

Define all your rooms using the right click context menu on the left hands side tree. You can organize your rooms in a hierarchy or you can place all of them under the zone called „Home".

Whenever you enter a new child zone into the tree you are required to mark the zone (room) in the floor plan on the right hand side.

Just click into the right hand floor plan and mark the corners of the zone. Once you have completed the outer border you can drag the points you just placed. You can also move the whole zone object when click and hold into the middle of the object.

Use the “Enter” key to stop editing and escape to cancel. To save zone without drawing area click enter immediately after entering edit mode. Hold shift to draw straight lines. Stop this mode by clicking on „Stop Editing Zone“ on the bottom end context menu.

For each individual child zone a new floor map can be uploaded. This allows for instance to have one overall map and certain sub parts or to have one map for each floor of the home.

Devices can be placed on each level of the floor plan hierarchy. The internally recognized place of the device is supposed to be the “deepest” child level zone.

Example: A home has two floors that are two subzones. A lamp is in the living room on floor one. The zone shown in the other dialogs of Z-Way is the living room, even if the device was placed in the floor 1 area on the main zone map.

All changes will be saved after hitting the save button in the bottom context menu. Make sure to save all your changes!

**(3) Place your devices**

**The bottom line context menu**

If you have already included devices you can place them into the floor plan. Otherwise just skip this step.

To place devices into the map activate the device list by clicking „Place Devices“ on the bottom context menu. A list of all available and not placed devices will appear on the left hand side below the zone tree. Just drag and drop the device icons into the floor plan. The devices will be assigned to the zone selected in the zone tree regardless of the place in the map. So it is possible to have a device assigned to a zone but not placed into the marked area of this zone.

When finished just click „Stop Placing Devices“. All changes will be saved after hitting the save button in the bottom context menu. Make sure to save all your changes!

**(4) Name your devices**

Once devices are placed on the map they can be renamed. Just click „Show devices in tree“ in the bottom context menu to show all devices in their respective zones. Right clicking on the Device default name („Device“ plus „Node ID) opens a dialog to change the name.

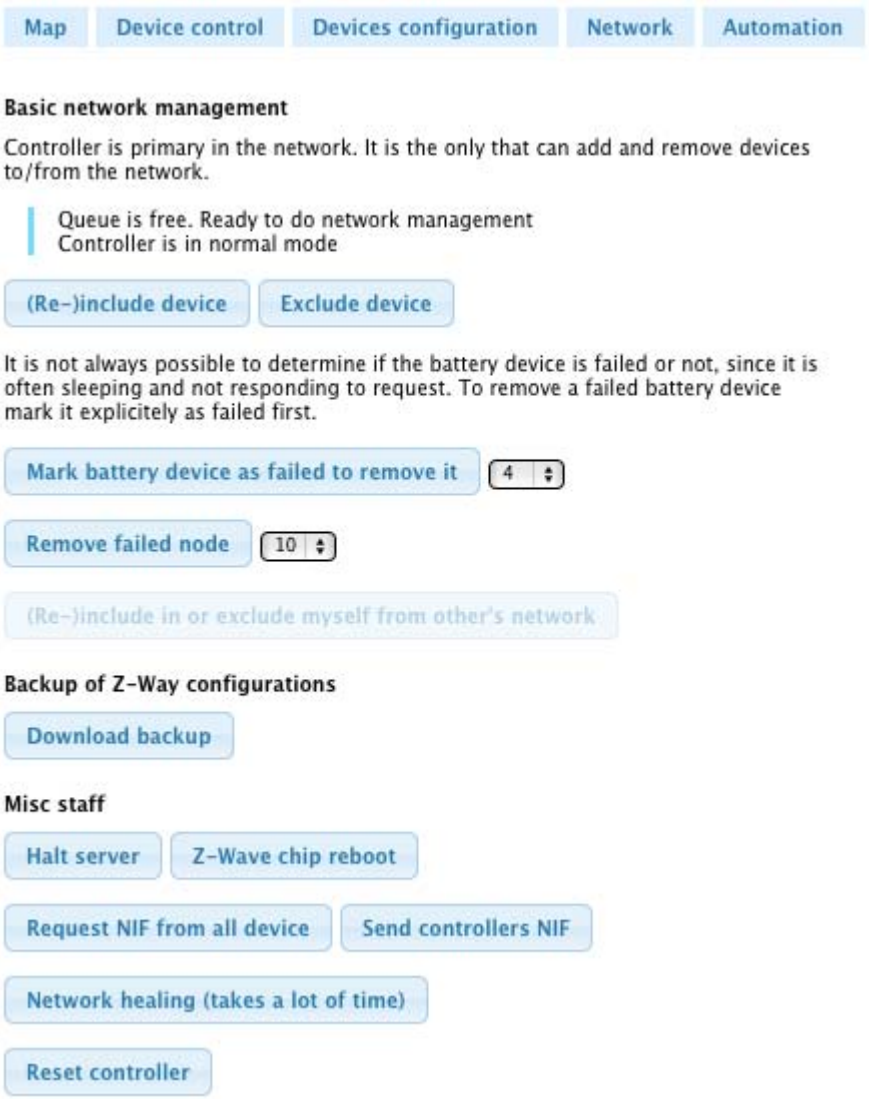
All changes will be saved after hitting the save button in the bottom context menu. Make sure to save all your changes!

## 3.3 The Device Setup and Configuration

For every device included into Z-Way to follow these steps need to be done.

1. In the tab “Network Management” include the device into the network.
2. Check in Device Status Tab if the interview was completed.
3. In the tab “Device Configuration” configure device specific parameters.
4. In the tab “Zones” place the devices in the area they belong to and rename it.
5. In the tabs “Rules, Scheduler and Scene“ check any changes in the automation engine of Z-Way.

### 3.3.1 Network Management




The tab “Network Management” allows including and excluding devices and managing the network.

**Inclusion**

You can include devices by pressing the “Include Device” button. This turns the controller into an inclusion mode that allows including a device. A status information line indicates this status. The inclusion of a device is typically confirmed with a triple press of a button of this particular device. However, please refer to the manual of this particular device for details how to include them into a Z-Wave network. The inclusion mode will time out after about 20 seconds or is aborted by pressing the “Stop Include” button.

If the network has a special controller with SIS function (Z-Way will try to activate such as function on default, hence this mode should always be active if the USB hardware used by Z-Way supports it) the inclusion of further devices can also be accomplished by using the include function of any portable remote control which is already included into the network. A short explanation above the include button will inform about the ways devices can be included.

**Exclusion**

You can exclude devices by pressing the “Exclude Device” button. This turns the controller into an exclusion mode that allows excluding a device. The exclusion of a device is typically confirmed with a triple press of a button of this particular device as well. However, please refer to the manual of this device for details how to exclude them into a Z-Wave network. The exclusion mode will time out after about 20 seconds or is aborted by pressing the “Stop Include” button.
It is **possible to exclude all kind of devices** regardless if they were included in the particular network of the excluding controller.

If a node is not longer in operation it can’t be excluded from the network since exclusion needs some confirmation from the device. Please use the “Remove Failed Node” function in this case.
Please make sure that only failed nodes are moved this way. Removed but still function nodes - called phantom nodes – will harm the network stability.

**Mark Battery powered devices as failed**

This function allows marking battery-powered devices as failed. Only devices marked as failed can be excluded from the network without using the exclusion function. Typically multiple failed communications with a device result in this marking. Battery powered devices are recognized as sleeping in the controller and therefore all communication attempts with this device will be queued until a wakeup notification from this device is received. A faulty battery operated device will never send a wakeup notification and hence there is never a communication, which would result in a failed node status. Battery operated devices can therefore be manually marked as faulty. Make sure to only mark - and subsequently remove – devices that are faulty or have disappeared. A device, which was removed with this operation but is still functioning it may create malfunctions in the network.

**Remove Failed Nodes**

This function allows you to remove nodes, which are not longer responding or which are not available. Please refer to the manual section „Network stability" for further information about why failed nodes should be removed.

Z-Way allows removing a node, if and only if this node was detected as failed by the Z-Wave transceiver. The network will recognize that communication with a device fails multiple times and the device can't be reached using alternating routes either. The controller will then mark the device as "failed" but will keep it in the current network configuration. Any successful communication with the device will remove the failed mark. Only devices marked as failed can be removed using the "Remove Failed Node" function.

If you want to remove a node that is in operation use the "Exclude" Function.

**Include into / Exclude from a different network**

Z-Way can join a Z-Wave network as secondary controller. It will change its own Home ID to the Home ID of the new network and it will learn all network information from the including controller of the new network. To join a different network, the primary controller of this new network need to be in the inclusion mode.

Z-Way needs to be turned into the learn mode using the button "Start Include in others network". The button "Stop Include in others network" can be used to turn off the Learn mode, which will time out otherwise or will stop if the learning was successful.

Please be aware that all existing relationships to existing nodes will get lost when the Z-Way controller joins a different network. Hence it is recommended to join a different network only after a reset with no other nodes already included.

**Backup and Restore**

The backup and restore function allows to make a backup of the whole configuration of Z-Way into a file. The backup will include all included nodes with all their values, status and configurations as well as all rules, scenes and timer.

The restore function will overwrite all node values and configuration and automation settings. By setting a checkbox the restore function will also overwrite the network topology information stored in the Z-Wave chip itself. This will change the home id of the controller to the home ID stored in the backup file. This means that the new controller is an identical clone of the controller where the backup is from.

This function needs to be handled with extreme care. Running two identical controllers in one network will certainly screw up the settings of both controllers if not doing any further harm. Make sure that there is always only one copy of cloned controllers active.

The backup and restore function can be used to move the network between different implementations of Z-Way, e.g. between a Z-Cloud and a Z-Box

**Halt Server**

The "Halt Server" function stops or restarts the Z-Way software depending in the operating system environment or start script. The network configuration will be kept.

**Z-Wave chip reboot**

This function will perform a soft restart of the firmware of the Z-Wave controller chip without deleting any network information or setting. It may be necessary to recover the chip from a freezing state. A typical situation of a required chip reboot is if the Z-Wave chip fails to come back from the inclusion or exclusion state.

**Request NIF from all devices**

This function will call the Node Information Frame from all devices in the network. This may be needed in case of a hardware change or when all devices where included with a portable USB stick such Aeon Labs Z-Stick. Mains powered devices will return their NIF immediately, battery operated devices will respond after the next wakeup.

**Send controllers NIF**

In certain network configurations it may be required to send out the Node Information Frame of the Z-Way controller. This is particularly useful for the use of some remote controls for scene activation. The manual of the remote control will refer to this requirement and give further information when and how to use this function.

**Network Reorganization**

The „Network Reorganization" allows starting a network maintenance and error detection functions of Z-Way. Please refer to the section „ Network Stability" for further information about the use of these functions.

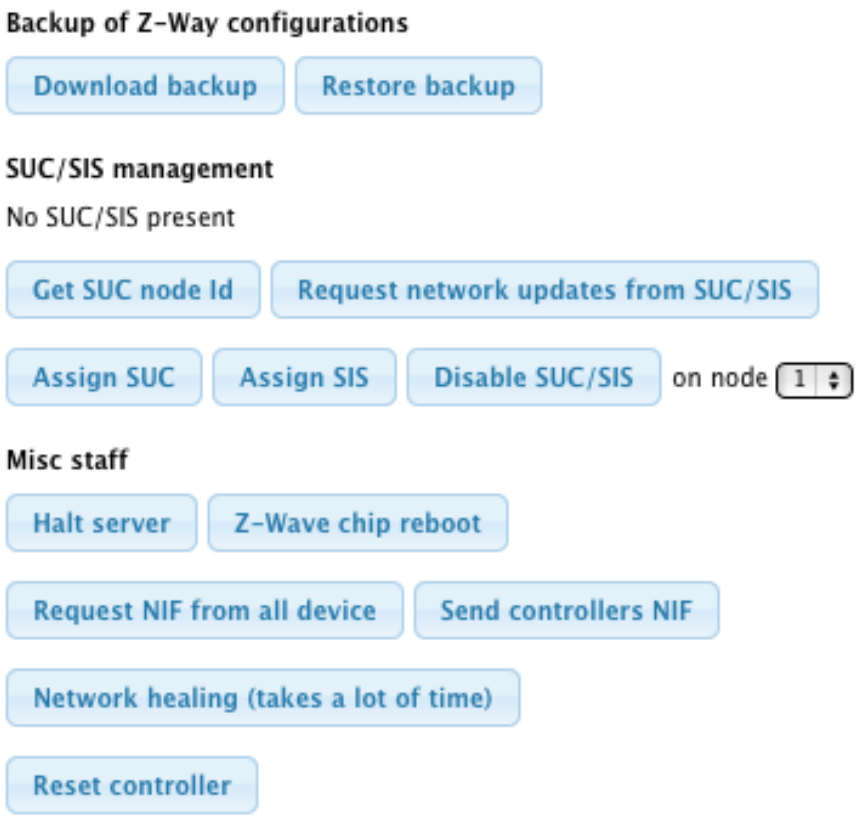
**Reset Controller**

The network configuration (assigned node IDs and the routing table and some other network management specific parameters) is stored in the Z-Wave transceiver chip and will therefore even survive a complete reinstallation of the Z-Way software.

The function “Reset Controller” erases all values stored in the Z-Wave chip and sent the chip back to factory defaults. This means that all network information will be lost without recovery option.

This function may create problems if there are still device included in the network, which are not reset to factory default (excluded) before the controller is reset. These devices may continue to communicate with the controller regardless if their Node IDs are stored in the controller after reset. This can cause all kind of problems. Hence, please handle this function with extreme caution!
Z-Wave-ME hardware does not have this problem anymore!

**Change Controller - Experts Mode only**




The controller change function allows to handover the primary function to a different controller in the network. The function works like a normal inclusion function but will hand over the primary privilege to the new controller after inclusion. Z-Way will become a secondary controller of the network.
This function may be needed during installation of larger networks based on remote controls only where Z-Way is solely used to do a convenient network setup and the primary function is finally handed over one of the remote controls.

**SUC/SIS Management – Experts Mode only**

This interface allows controlling the SUC/SIS function for the Z-Wave network.

Z-Way will – according to the Z-Wave guideline –always try to become SUC/SIS of a Z-Wave network. If Z-Way will remain in the network there is no reason to change the default settings of SUC/SIS. In case Z-Way is used as installation controller only it is recommended to turn off the SUC/SIS function and assign the SUC/SIS role to a different static controller within the Z-Wave network.

### 3.3.2 Interview Process

After the inclusion of a new device Z-Way will interview this very device. The interview is a series of commands Z-Way is sending to the device in order to learn the capabilities and functions of this device.

Depending of the capabilities announced in the Node Information Frame that was received during the inclusion Z-Way will ask further questions to get more detailed information. The interview process may take some seconds since more questions may be required to ask depending in certain answers given.

Since all functions of a device are grouped in so called Command Classes each command class announced in the Node Information frame will typically cause its part of the interview.

The interview will be executes in three different Steps:

1. In case there is a Version Command Class ask for the Version of the device and the versions of all Command Classes announced in the Node Information frame. Otherwise Version 1 is assumed.
2. In case there is a Multi Channel Command class announced, ask for the number of the capabilities of the different channels and repeat Step 3 for each channel.
3. Ask for all capabilities of all command classes.
4. Do some auto configurations if needed.

The „Device Status“ tab will indicate if the interview was successfully completed. The blue information icon shows if the interview was not complete. Clicking on this icon opens a dialog with all command classes and the status of their respective interviews.

A complete interview is important in order to have access to all functions of the device included. Incomplete interviews may also be a reason for malfunctions of the network.

There are several reasons why an interview may not be completed.

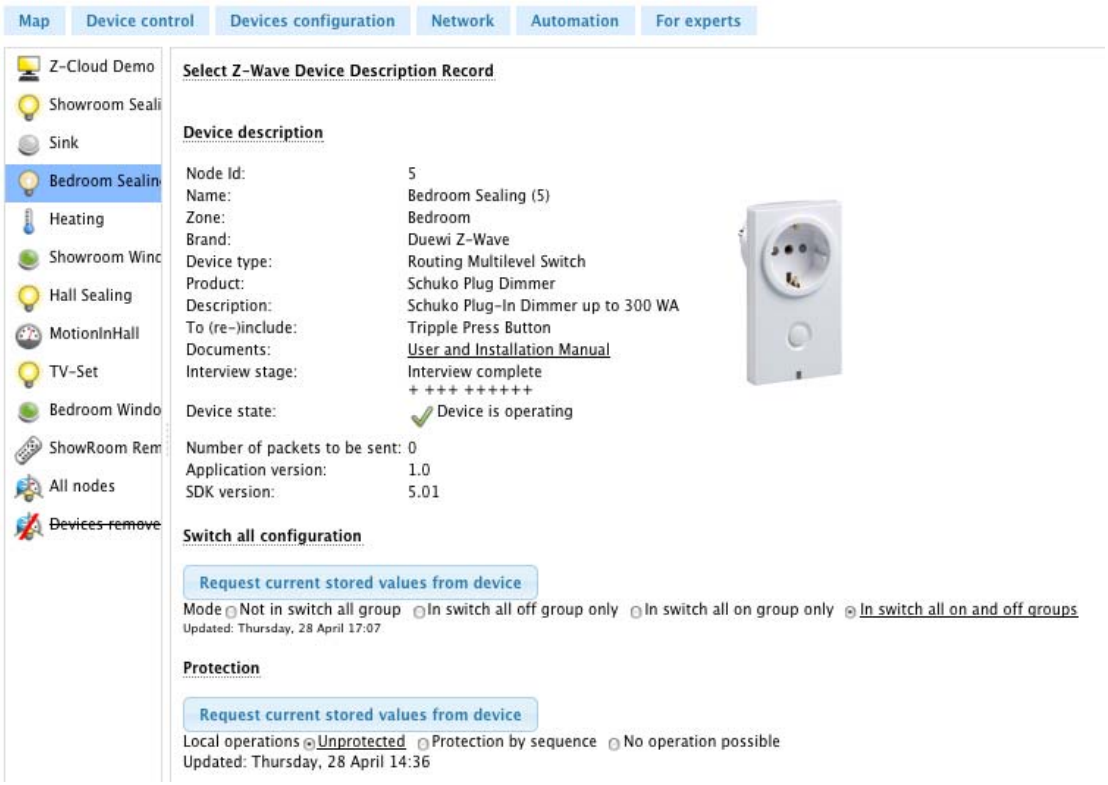
(1) A battery-operated device may be gone into sleep mode too early. In this case its possible to wake up the device manually to complete the interview. Sometimes manual wakeup is needed several times.
(2) The device does not fully comply with the Z-Wave protocol. This is particularly possible for devices that were brought to market before

2008. The current more sophisticated certification process makes sure that device are 100% compatible to the Z-Wave product when they hit the market. Please check online information (wiki, forums) on details and possible ways to fix these kinds of problems.

(3) The device does not have a reliable communication route to the controller. Interview communication typically use longer packets than normal polling communication. This makes the interview communication more vulnerable against weak and instable communication links. Its possible that the controller is able to include a device and even receive confirmation of a polling request but still not being able to complete the interview. However this is a rare case.

(4) The device may be simply broken.

### 3.3.3 Device Configuration




Each Z-Wave device is designed to work out of the box after inclusion without further configuration. However it may be suitable and in certain contexts even required doing device specific configurations.

The device configuration page allows to further configure the device and to access certain additional information about the device. The tab is grouped into several sections. The sections can be toggled from invisible to visible and back by clicking on the headlines:

- Select Z-Wave Device Description Record
- Device Description
- Configurations
- Actions with configurations
- Advanced Actions

**Select Z-Wave Device Description Record**

After a successful inclusion Z-Way will interview the device to gather further information.

Certain information such as names of association group, the brand name of the device and the parameters of further configuration values can't be detected during interview. Z-Way uses a device database with product description files to obtain this information. In order to identify the right device description record certain parameters of the interview are used.
If these parameters match exactly one device description record this very record is loaded and its content is shown on the device configuration page automatically.

If the information from the device is sufficient to select one specific record from the database this section of the tab is hidden. If it's not possible to identify the correct device description record the user can manually choose the correct record. It's also possible to manually change the selection of the device description by unhiding this section and clicking on the "Select Device Description Record" button.

**Device Description**

The upper part of the dialog shows some descriptive values of the device. The Z-Wave device type is the only value generated solely from the interview data. All other data are taken from the device description record.

**Zone**: ... the zone/room the device is assigned to. Will be manually defined in Zone-tab.

**Brand:** … the product code or brand name of the device. This will be taken from the device description record.

**Device Type:** ... the type of Z-Wave device as reported by the device during inclusion.

**Description:** … a verbal description of the function. This will be taken from the device description record.

**Interview Stage:** …shows the progress of the interview process. This information is generated by Z-Way.

**Inclusion Note:** … how to (re-) include the device. This will be taken from the device description record.

**Wakeup Note:** … this will be taken from the device description record.

**Documents:** … If the device description record offers links to manuals or other online documents there are shown here. This will be taken from the device description record.

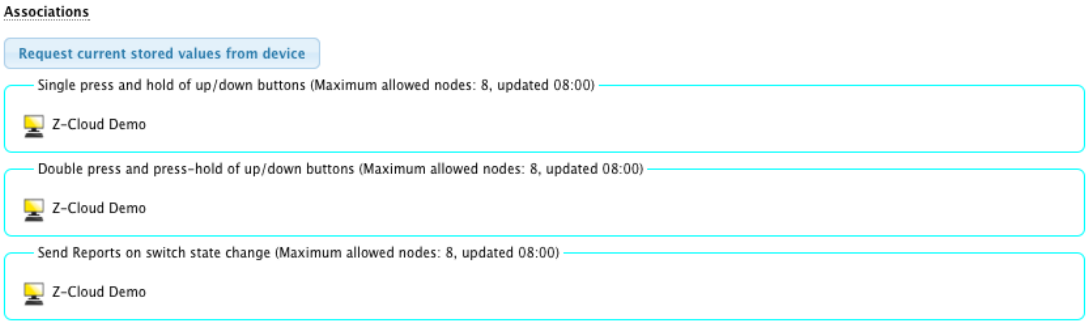
**Device State:** Status of the device plus number of packets queued for this device

There are a couple of reasons why no device description record was found:

1. There is no record for the device available. Since there are always new device on the market Z-Way need to catch up and update its device database. If your device is not found, updating to the most recent version of Z-Way may help.
2. The interview was not finished to the point where enough parameters were detected to identify the correct device description record. You may manually choose the correct device description record using the button "Select Device Description Record". A dialog box will be opened for manual selection of the product (if available). The manual selection of a device description record is only needed if no record was found on default.
3. The interview of the device was completed but the device does not offer enough information to identify the correct device. You may manually choose the correct device description record using the button "Select Device Description Record". A dialog box will be opened for manual selection of the product (if available). The manual selection of a device description record is only needed if no record was found on default.
4. There is more than one device description record matching the information gathered during interview. This is particularly possible if a vendor sells devices with different firmware and functions without properly updating the firmware version information. You may manually choose the correct device description record using the button "Select Device Description Record". A dialog box will be opened for manual selection of the product (if available). The manual selection of a device description record is only needed if no record was found on default.

**Associations:**

Please refer to the manual section "Associations versus Scenes" on how and why to use associations.




If there was no previous device of the same type installed the interface will show the values as read from the device. If there was already a device of the same kind installed there may exist a stored default configuration for this particular device. Then the setup in the device may differ from the default configuration stored in Z-Way.

**Gray Icon:** This Node ID is stored in the device but its not stored in the default configuration of the Z-Way. You can double click this device to store this setting in the Z-Way default configuration of this device type.

**Red Icon:** This Node ID is stored locally but not in the association group of the device yet. Apply the settings to transmit the setting into the device. In case of a battery operated device you need to wakeup the device in order to store the configuration.

**Black Icon:** This Node ID is stored both in the device and in the local configuration of Z-Way.

*Hint: The auto configuration function of Z-Way will place the node ID of the Z-Way controller in all association groups if possible. This allows the activation of scenes from these devices.*

### Device Configuration

This section will – if device description record is loaded – show device specific configuration values including their possible parameters and a short description of the configuration value. You may change these values according to your needs.

### Actions with configurations

The most important action in regard to the configuration is to apply the configuration to the device. This is only done when the button "Apply configuration for this device" is hit. This button is therefore even shown, when the rest of the tab part is hidden.

- Mains Powered Devices: The settings will become effective immediately after hitting the button.

- Battery Powered Devices: The settings will become effective after the next wakeup of the device, as shown in the Device Status tab.
- Battery Powered Controllers (remote controls or wall controllers): The settings will only become effective if the devices are woken up manually. Refer to the controller manual for more information on how to wake up the device. Appendix B may also give further advice.

If there are many similar devices in a network it is desirable to just apply one working configuration to all these devices. This can be done using the function „Copy from other Device“.

The set of defined configuration values is stored for every device. Therefore its possible to pick a different device and reuse its configuration values for the device to be configured. The „Save“ function of the bottom context menu allows saving the configuration for further use and reuse.

**Functions of Expert Mode**

The expert mode offers some further functions to the users, which are designed for installers or for debugging.

Request NIF – in section “Select Z-Wave Device Description Record”

This function requests the selected device to send its Node Information frame (NIF) to the controller. It can be used instead of triple pressing a button on the device itself that would also instruct the device to send it NIF. The NIF is needed to know device capabilities.

Delete Configuration of this device – in section “Actions with configuration”

This function deletes the stored configuration for this device. This function is for debugging purposes only.

Force Interview – in section “Advanced”

This functions forces to redo the whole interview. All previous interview data will be deleted. This function is for debugging purposes only.

Show Interview Results– in section “Advanced”

This function shows the result of the interview. This function is for debugging purposes only. For information about reasons for incomplete interview please refer to the manual section “Device Status”.

Save Device Description Record

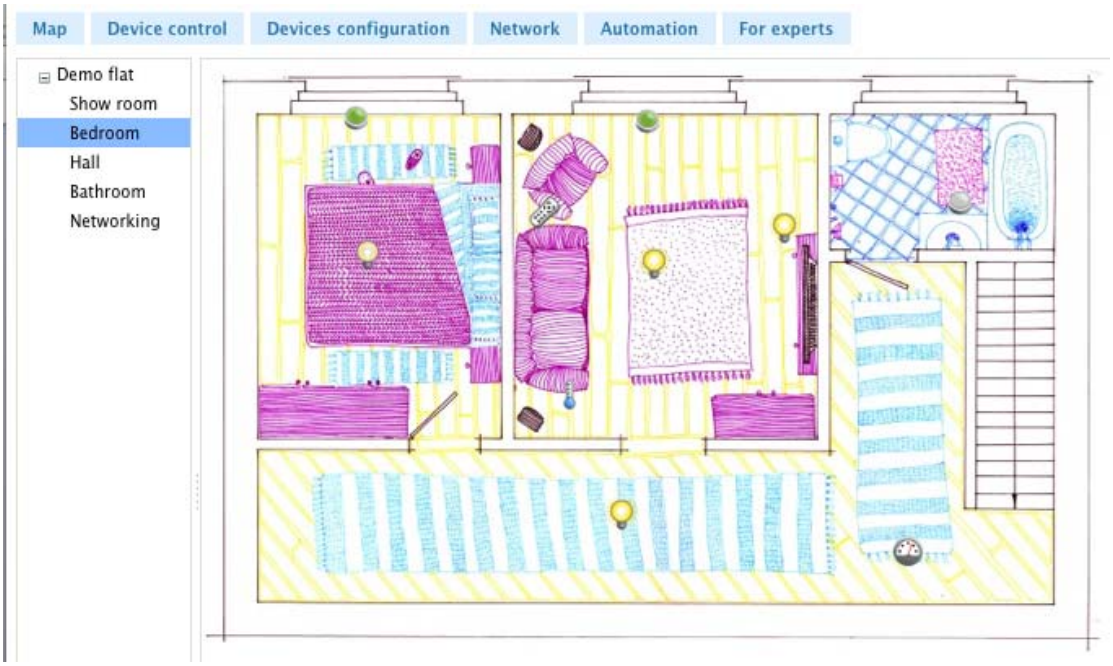
In case there is no device description record available for the given device this function will generate a skeleton of such a record using the data gathered during interview. Please send this file to info@zwave.me for completion and inclusion in the next release of the software.

Switch to raw mode

This function turns the configuration dialog into a generic mode.

The bottom context menu function „Reset Configuration“ deletes all configurations stored in Z-Way, but does not affect devices!

### 3.3.4 Place your devices on the zone map




Consult the “Zones” tab to place the configured device on the floor plan.

To place devices into the map activate the device list by clicking on „Place Devices“ on the bottom context menu. A list of all available and not placed devices will appear on the left hand side below the zone tree. Just drag and drop the device icon into the floor plan.

*Attention: The devices will be assigned to the zone selected in the zone tree regardless of the place in the map. So it is possible to have a device assigned to a zone but not placed into the marked area of this zone.*

When finished just click „Stop Placing Devices“. All changes will be saved after hitting the save button in the bottom context menu. Make sure to save all your changes!

### 3.3.5 Name your devices

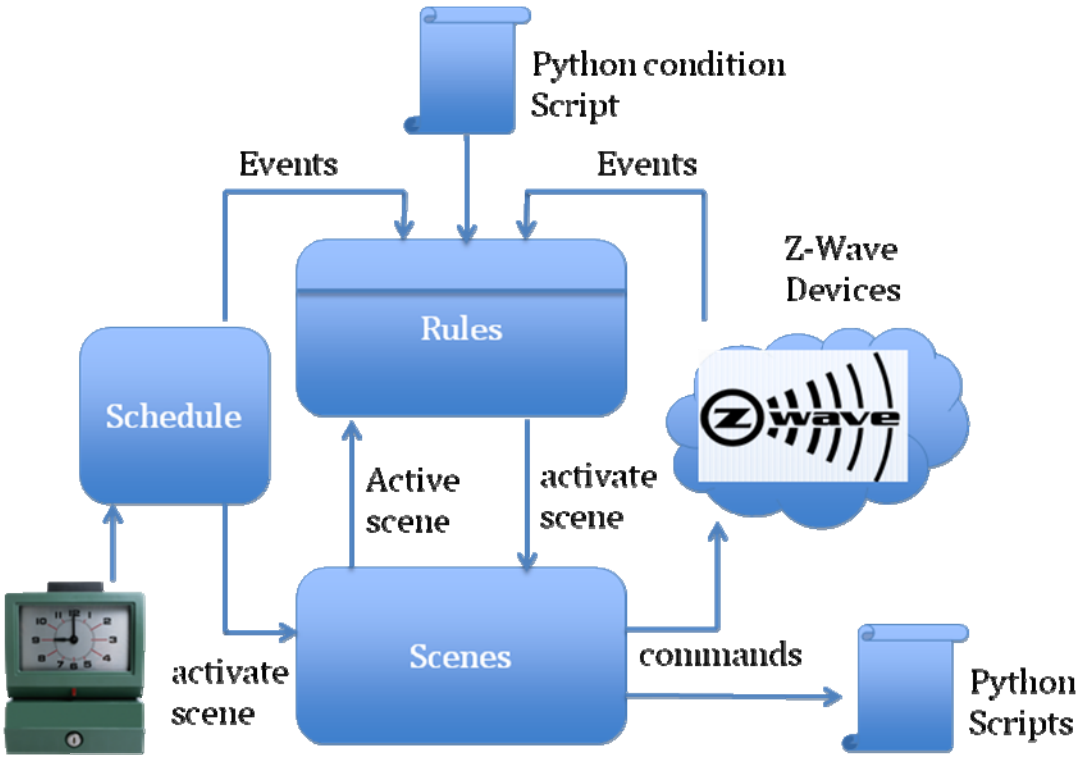
Once devices are placed on the map they can be renamed. Just click on „Show devices in tree“ in the bottom context menu to show all devices in their respective zones. Right clicking on the Device default name („Device“ plus „Node ID) opens a dialog to change the name. All changes will be saved after hitting the save button in the bottom context menu. Make sure to save all your changes!

### 3.3.6 Check automation function

Please refer to the tabs to control the automation if the included device has any impact on the settings or allows further settings. Please refer to the section “Association versus Scenes” for further information about how to setup and use scenes.

## 3.4 Automation

The following section gives you an overview about how to setup and use automation functions in Z-Way. Automation functions refer to the ability of Z-Way to execute different commands dependent on different events. These events can be generated from Z-Wave devices, the user interface or timers or any other logic.

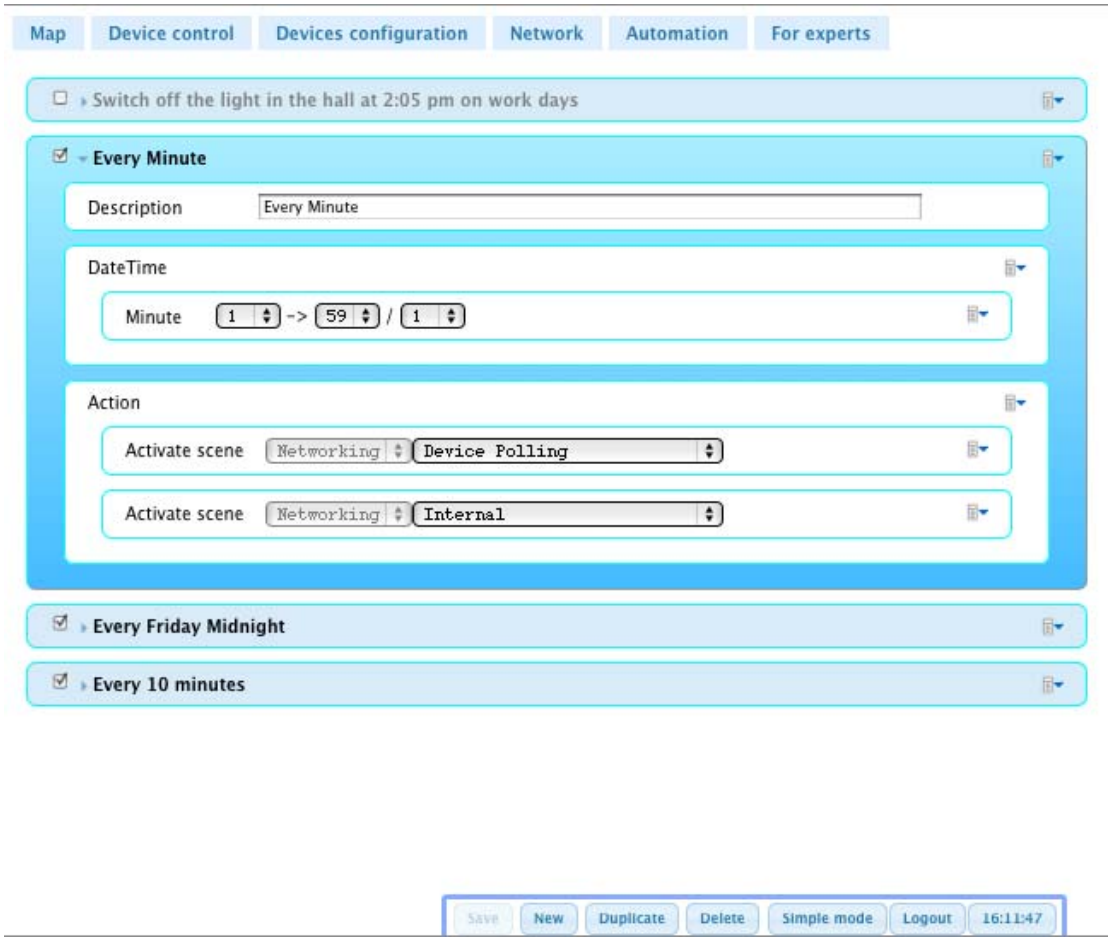


This image shows the logic of the automation engine. As a result of automation functions certain commands are sent to Z-Wave devices or a python script can be executed that gives great flexibility to do things far beyond the functionality of the Z-Wave network.
Commands and scripts are defined in scenes. This makes the scene the basis of the whole automation engine. A scene describes the functions to be executed and scenes are executed depending on certain conditions.

One condition is a timer event. The schedule function allows defining certain timer-based events. It is possible to define repeating event (every hour) or one time event (Monday, 8.00). These timer-based events either activate a scene directly or they are used in the rules engine. The rules engine allows a logical connection between timer events, events generated by different Z-Wave devices and events generated as a result of a python scripted operation. The status of a scene (the last activated on a zone and therefore still "active" scene) can be used as logical input parameter as well.

Rules are based on Boolean logic with AND, OR and NOT. As a result a rules generates an event to activate a scene.

### 3.4.1 Schedules



The Tab “Schedules” allows setting up timers. Timers generate events on a regular basis. The event can be used to activate scenes.

The context menu on the lower end of the browser window allows inserting and deleting schedules. The schedule can be renamed.

Right-Click into the “DateTime” section of the schedule opens a context menu to add timing information.

The scheduler becomes active when all of the selected trigger conditions are true.

Trigger conditions can refer to minutes, hours, days, weekdays, months and the actual time of sunset and sunrises. sunrise and sunset are calculated based on the geographical location of the controller and the day of the year.

Once a trigger condition scale is selected the interface allows setting a start and an end value plus an interval. All missing scales are treated as “every unit”. If more then one scale is defined they are logically connected as AND (if Scale 1 equals xxx AND scale 2 equals yyy). If more than one condition for one scale is defined, they are logically connected as OR (if Scale 1 equals xxx OR scale 1 equals yyy).

*Assuming “hour” is selected as condition scale, start is set to 08:00, end is set to 16:00 and the interval is set to 1, the scheduler will be active every minute on every day between 08:00 and 16:00 as every full hour. Adding a condition scale “minute” with start=0 and end=0 will reduce the events to every full hour between 8 and 16.00. Assuming the event shall happened between 8:00 and 16:00 but 15 minutes after the hour, an additional “minute” scale needs to be added with start=15, end=15, interval=1.*

A timer being active every minute (e.g. to report sensor values) would be:

*Minute: from 0 to 59 in steps of 1.*

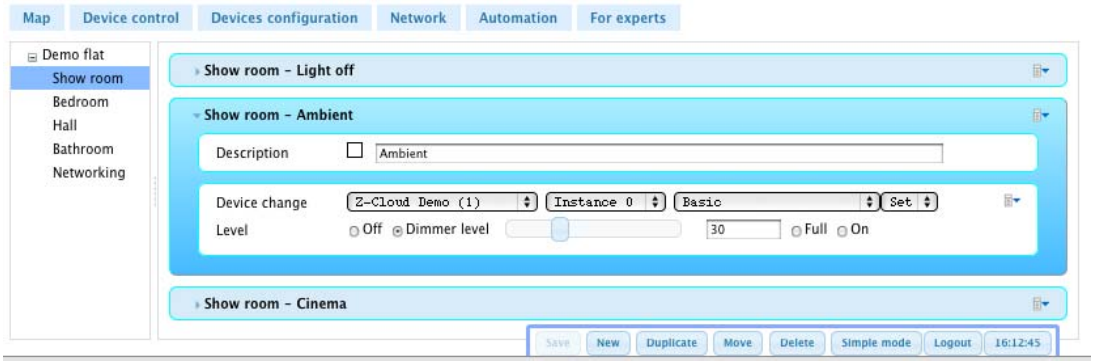
Right-Click into the “Action” sections allows assigning an action to the timer event. Its possible to select between “Activate Scene”, “Timed Scene” and “Timed Event”

Activate Scene: You can choose a defined scene that will be activated when the defined condition(s) become true.

Timed Scene: You can choose a defined scene that will be activated after a defined delay (in seconds) when the defined condition(s) become true.

Timed Event(in expert mode only!): The timer emulates any event that would normally be generated by one of the Z-wave devices in the network. You can select the event type, the source and the destination Z-Wave device of the event and the event value.

### 3.4.2 Scenes




The scene dialog allows defining and activating different scenes for the zone selected. Since the scene concept is so highly flexible it is used beyond just activating a scene status for a given zone.

A scene is a set of actions resulting in a defined state after activating the scene.

*A scene called party may involve to turn the light up in the party room, heat this room and the floors, keep the light at the bathroom always at 50 % to save some energy but allow party guests to find the facilities after having some drinks. Scenes can be applied generally to the whole home ("scene I am away") or to a special zone/room only ("family is watching tv").*

The implemented concept of scenes is very flexible and goes far beyond simply switching lights on and off.

To add a new scene select the zone on the left hand zone tree and click "New" on the bottom context menu. The scene can be renamed.

Right clicking into the blue zone of the zone opens a context menu with the following options:

Activate Scene: This activates the scene regardless of any further conditions set.

Insert Scene After and Insert Scene Before. Both commands create a new scene but allow putting the scenes in a desired order on the screen.

Delete Scene: Allows deleting the selected scene.

Insert Action: This command opens a sub menu with three different actions to be tied to the scene:

Activate Scene: This will add a dropdown menu to select another scene to be selected. Activating a scene from another scene is useful to structure different

scenes in different zones. An example would be to call the “all off” scene in all zones from an “all off” scene in the home area.

Device Change: This will allow sending a specific command to a selected device. Drop down menus allow to select the device and the command and –depending on the command selected –set certain parameters. A typical command to turn on a lamp would be a Basic set with a value of “on” or “255”.

Please keep in mind that all actions of the scene will be executed after each other. Hence, to turn one light on and one other light off send the On command before the off command, to allow one light turning on before the other one will be turned off not leaving people in dark.

Run Python script: This is the by far most powerful and flexible method to execute a scene. It is possible to write a python script using Z-Wave API functions and the Z-Way data model. Please refer to Chapter 9 “Scripting” for further information.

Please keep in mind that all actions of the scene will be executed after each other.

**Special Network Maintenance Scripts**

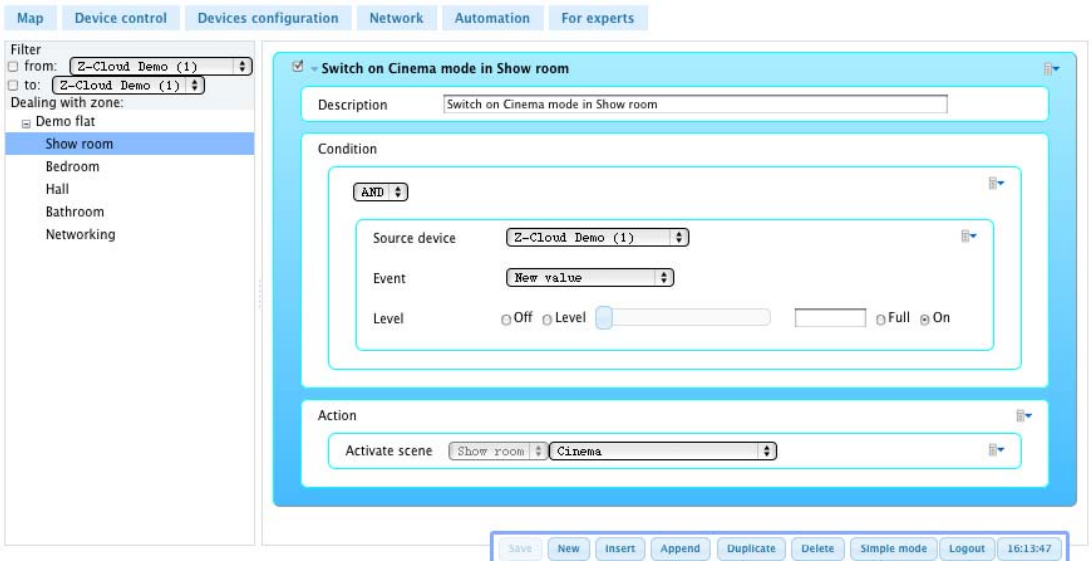
The zone “Network” already contains some scenes with python scripts, which are called regularly. These scenes contain scripts to perform certain tasks within the network. The idea behind placing these functions as scenes and not hidden in the backend is to allow users to modify and adapt these critical functions according to their needs.

Battery Polling: This script polls all battery status values. It should be called once per week or even less often.

Device Polling: This script polls all devices for their sensor value or status information. Users may restrict the polling interval or exclude or include certain functions or devices to be polled or not. Refer to the chapter “Network Stability” for more information how to tweak this script.

Data Save: This scene saves all Z-Way data.

### 3.4.3 Rules




Rules are the engine to logically connect events. It allows applying Boolean logic (AND, OR). You can create a new rule using the “New” button in the bottom context menu.

Each rule has conditions that the rule turns into “true” and an action that is executed when all the conditions become true.

Right clicking into the action and the condition field opens a context menu to add conditions respective actions

**Conditions**

Each condition is a logical combination of events or other conditions. When adding rules there is already one – empty – condition pre set. It is possible to add more conditions to the rule. All conditions are logically “AND” connected. This means that the rules only becomes “True” and the actions are executed if all conditions are True.
An empty “AND” or empty “NOT” means true, empty “OR” is false.

Right clicking into the condition field opens a context menu to add different sub conditions: The following sub conditions are available:

*Logical Operation:*

Repeats the condition logic, which allows having different layers of logical connections of conditions.  An example is

IF (A OR B) AND (A AND B OR C) THEN do something

Note that logical condition are checked until they can't satisfy the event

parameters any more. This is a common practice in modern programming languages and save CPU power during condition checks. So, to speed up automation for big installations check the most important condition (most narrowing) first. For example, check source node Id first and the time as second, since this will skip time checks for all other nodes.

*Check active Scene:*

Makes the rules dependent on which scene was executes last in a given zone.

*Check Event:*

This option allows testing for a specific event sent from a specific device. Selecting this option creates and empty field. Right clicking into this field opens another context menu for further specifying the kind and type of event.

Check Source Device:

Picking a source node narrows the events to those sent by the selected source node.

Check Source Instance:

Picking a certain instance of the source node narrows the events to those sent by the selected instance of the selected source node.

Check Destination Device:

Picking a destination node narrows the events to those sent by the selected source node.

The controller will typically only receive commands sent to him or sent as broadcast. In case the controller supports virtual nodes these virtual nodes are usable as destination device too.

Check Destination Instance:

Picking a certain instance of the source node narrows the events to those sent by the selected instance of the selected source node.

The instance selection can be done even without selecting a source node. In this case every event sent from any device, which has an instance of the chosen number will generate a "True" condition.

Event Type:

Picking a certain event type further narrows down the number of commands, which will generate a "True" condition.

Evaluate python script:

Its possible to write a python script that evaluates certain conditions within or outside the Z-Way event logic. As an example it would be possible to check whether a certain PC address can be pinged or not.

It is possible to write a python script using Z-Wave API functions and the Z-Way data model. Please refer to manual chapter "Scripting" for further information.

**Actions**

Right-Click into the "Action" sections allows assigning an action to the timer event. Its possible to select between "Activate Scene", "Timed Scene" and "Timed Event"

Activate Scene: You can choose a defined scene, which will be activated when the defined condition(s) become true.

Timed Scene: You can choose a defined scene, which will be activated after a defined delay (in seconds) when the defined condition(s) become true.

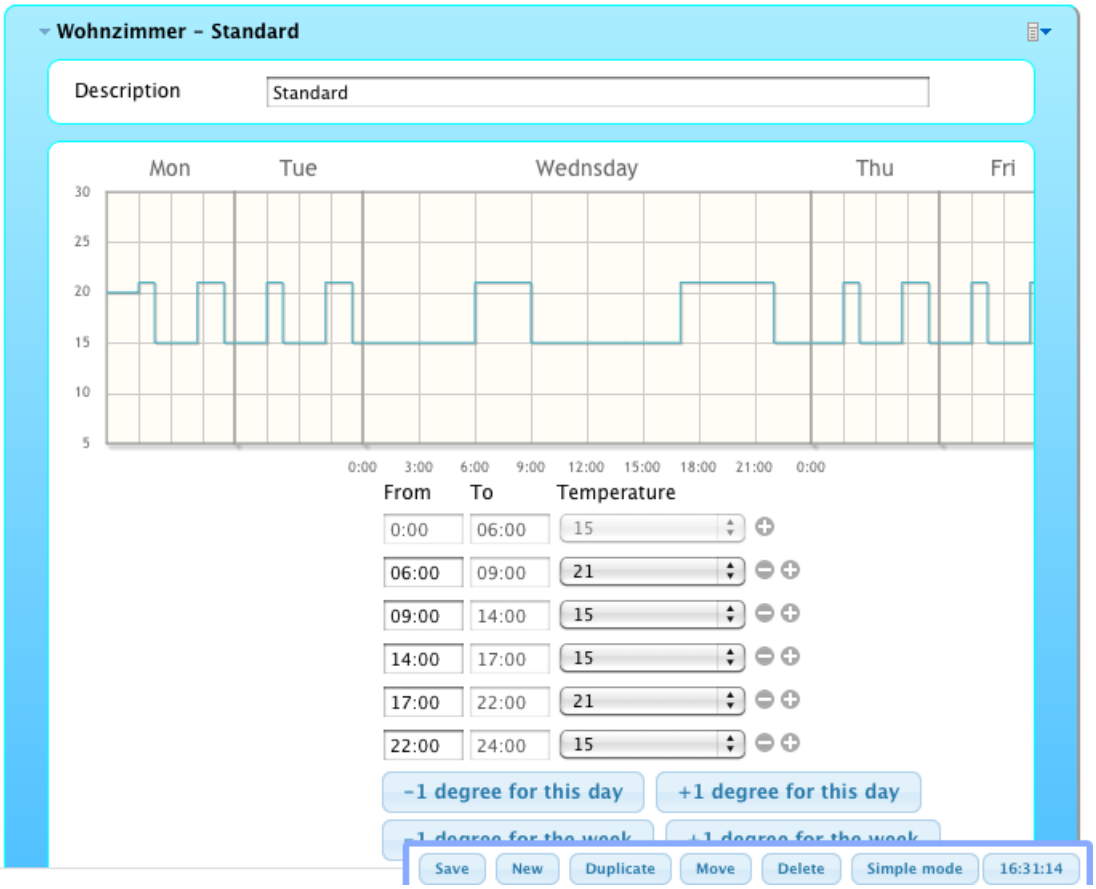
Timed Event (in experts mode only): The timer emulates any event, which would normally be generated by one of the Z-wave devices in the network. You can select the event type, the source and the destination Z-Wave device of the event and the event value. The event will be used by rules.

### 3.4.4 Climate Schedules

**Relationship between Climate Control Schedules and Zones**

Z-Way allows setting up climate control schedules. A climate control schedule is always referred to a zone. Every thermostat assigned to this zone can apply a climate control schedule of this zone. A zone can have multiple climate control schedules (Living everyday, Vacation, School off, ...). There is exactly one climate schedule active per zone. The activation of a climate control schedule and the switching between climate control schedules can be done right on the climate control schedule dialog using the right click context menu. It is also possible to switch climate schedule via the scene automation logic or manually on the dialog zone status described below.

It is possible to create multiple schedules ("I am working day shift", "I am working night shift") for a zone but only one schedule is active at a given time in a given zone.




**Creating and Editing a Climate Control Schedule**

After creating and naming a new schedule a viewgraph of all seven weekdays and the target ambient temperature is shown. Clicking into the schedule of one weekday will zoom out this weekday and open a control panel below this view graph to edit the viewgraph.

The control schedule is defined by switching points. A switching point is a certain time of a weekday and the temperature the thermostat will control from this moment on. This temperature will stay active until the next switch point changes this temperature again.
The left input field of the schedule editor allows setting a switch point. The second time is for information only for how long this schedule will stay active. A drop down menu allows setting the desired temperature. The (+) and (-) buttons allow to add or remove switch points.

Some self describing further buttons below allow changing all switch points into one direction or copy the switch points from one day to other days.

**Overwrites**

It is the desire not to change the climate schedules too often but setting them up once and keep them in the devices and the software.

However in real live there may be a need to change the desired temperature regardless of the predefined schedules.

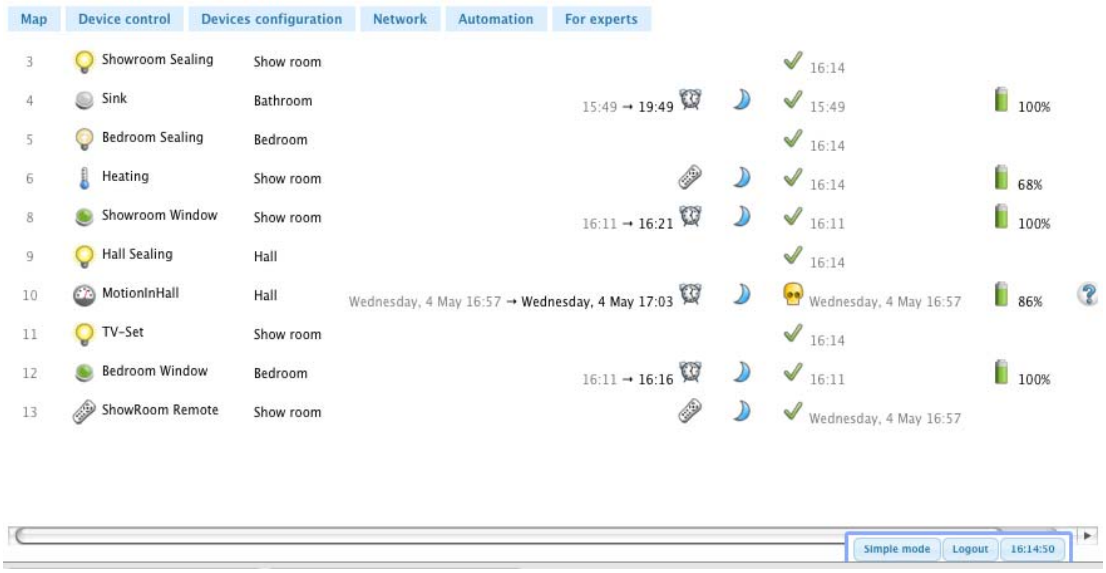
This is always possible turning the device back into the manual mode or by using the overwrite function of the climate control schedule. The overwrite function allows to move the whole schedule by certain degrees warmer or colder. The schedule with its temperature change points will remain the same but the target temperature changes.
A overwrite of +2 degree will increase all defined temperatures by two degrees. This new temperature will be kept until it is reversed. This is called „permanent overwrite“.

There is an overwrite per heating zone and a overwrite per device. The overwrite per heating zone is set for the whole zone and is applied in the dialog „zone state“ of the menu automation. A overwrite for the device can be applied in the dialog „Thermostats“.

Because of the two possible overwrites the resulting target temperature for a given device is the temperature set in the climate schedule plus the change applied for the zone plus the change applied for this device.

## 3.5 Maintenance

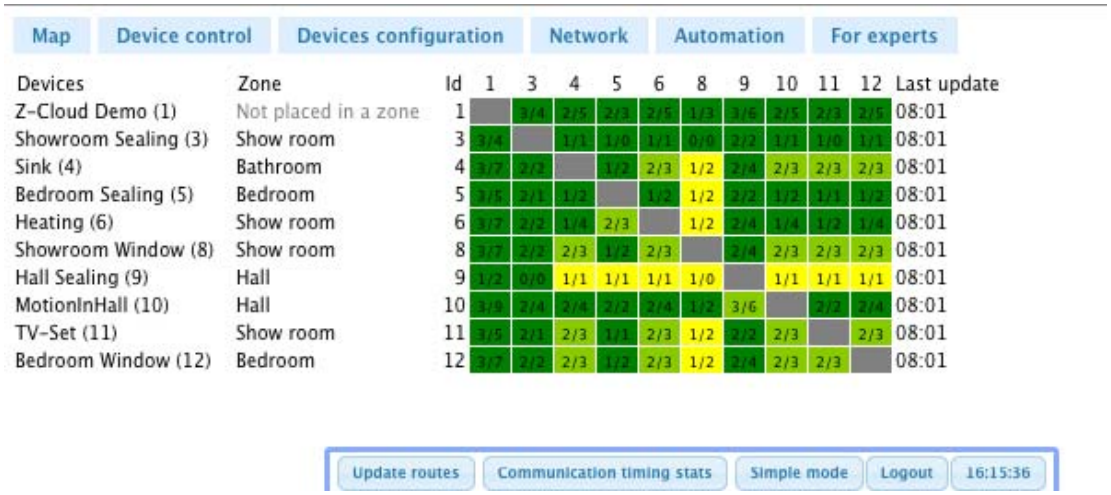
### 3.5.1 Device Status overview




This tab gives an overview of the network status and the availability of each device. It shows the time stamp of the last interaction between the controller and the device. For battery powered devices the battery charging status, the time of the last wakeup and the estimated time for the next wakeup is shown.
An info icon indicates when the interview of a device was not completed. Clicking on this device opens a window showing the interface status by command class.

Please refer to the manual section “Interview” for more information about the interview process.

### 3.5.2 Routing




The routing table of the Z-Wave network is shown. It indicates how two devices of the Z-Wave network can communicate with each other. If two devices are in direct range (the can communicate without the help of any other node) the cross point of the two devices in the table is marked as **dark green**. The color **light green** indicates that the two nodes are not in direct range but have more than one alternating routes with one node between. This is still considered as a stable connection.
The **yellow color** indicates that there are less than two “one-hop” routes available between the two nodes. However there may be more routes but with more nodes between and therefore considered as less stable.

A **red indicator** shows that there are no good short connections between the two nodes. This does not mean that they are unable to communicate with each other but any route with more than 2 routers between Z-Way is considering as not reliable, even taking into account that Z-Wave supports routes with up to four devices between. **Grey cells** indicate the connection to the own Node ID.

**The general rule of thumb is: “The greener the better”**

The table lists all nodes on the y-axis and the neighborhood information on the x-axis. On the right hand side of the table a timestamp shows when the neighborhood information for a given node was reported.

In theory the table should be totally symmetric, however different times of the neighborhood detection may result in different neighborhood information of the two devices involved.

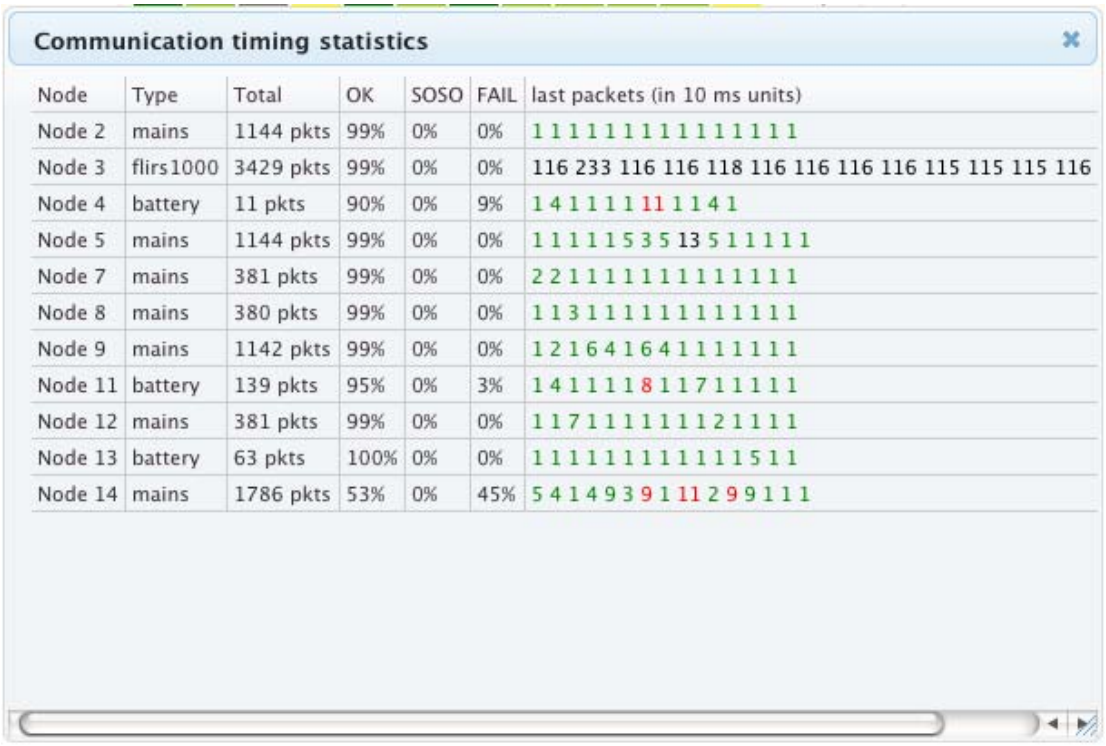
The neighbor information of the controller works with an exception. The Z-Wave implementation used in current Z-Wave transceiver does not allow requesting

an update of the neighbor list for the controller itself. The neighborhood information displayed for the controller s therefore simply wrong.

Battery powered devices will report their neighbors when woken up and report their mains powered neighbor correctly. However mains powered devices will report battery-powered devices as neighbors only when routes are updated twice. This is less critical because battery powered devices can't be used as routers and are therefore not relevant for calculating route between two nodes anyway.

The context menu command "Network Reorganization" allows re-detecting all neighborhood information (battery powered devices will report after their next wakeup!) Please refer to the manual section „ Network Stability" for further information about the use of this function.

**Communication Timing Statistics – Expert Mode only**




| Node | Type | Total | OK | SOSO | FAIL | last packets (in 10 ms units) |
|---|---|---|---|---|---|---|
| Node 2 | mains | 1144 pkts | 99% | 0% | 0% | 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 |
| Node 3 | flirs1000 | 3429 pkts | 99% | 0% | 0% | 116 233 116 116 118 116 116 116 116 115 115 115 116 |
| Node 4 | battery | 11 pkts | 90% | 0% | 9% | 1 4 1 1 1 1 11 1 1 4 1 |
| Node 5 | mains | 1144 pkts | 99% | 0% | 0% | 1 1 1 1 1 5 3 5 13 5 1 1 1 1 1 |
| Node 7 | mains | 381 pkts | 99% | 0% | 0% | 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 |
| Node 8 | mains | 380 pkts | 99% | 0% | 0% | 1 1 3 1 1 1 1 1 1 1 1 1 1 1 1 |
| Node 9 | mains | 1142 pkts | 99% | 0% | 0% | 1 2 1 6 4 1 6 4 1 1 1 1 1 1 1 |
| Node 11 | battery | 139 pkts | 95% | 0% | 3% | 1 4 1 1 1 1 8 1 1 7 1 1 1 1 1 |
| Node 12 | mains | 381 pkts | 99% | 0% | 0% | 1 1 7 1 1 1 1 1 1 1 1 2 1 1 1 1 |
| Node 13 | battery | 63 pkts | 100% | 0% | 0% | 1 1 1 1 1 1 1 1 1 1 1 1 1 5 1 1 |
| Node 14 | mains | 1786 pkts | 53% | 0% | 45% | 5 4 1 4 9 3 9 1 11 2 9 9 1 1 1 |

In expert mode the bottom context menu offers "Communication Timing Stats". This opens an extra window to show some communication timing statistics that are very useful to find communication problems.

Z-Way measures the time between sending a command from the controller to a certain node and receiving a final acknowledgement from this node. These timings are shown on the right hand side for the different nodes. The minimum time between sending and receiving a confirmation is 10 ms (equals „1" in the table). If the packet needs to be resent this time may increase. As long as the time value is still marked green there was only a collision in the air or any other need for retransmission. A black entry indicates that the controller was not able to

reach the given node with three attempts and tried a different route that finally worked. A red entry indicates that the communication finally failed after some re-routing attempts.

For FLIRS devices (Frequently listening battery nodes) the minimum timing is about 1 second since a wakeup beam need to wakeup the device first. Hence the timings for these nodes are generally longer.

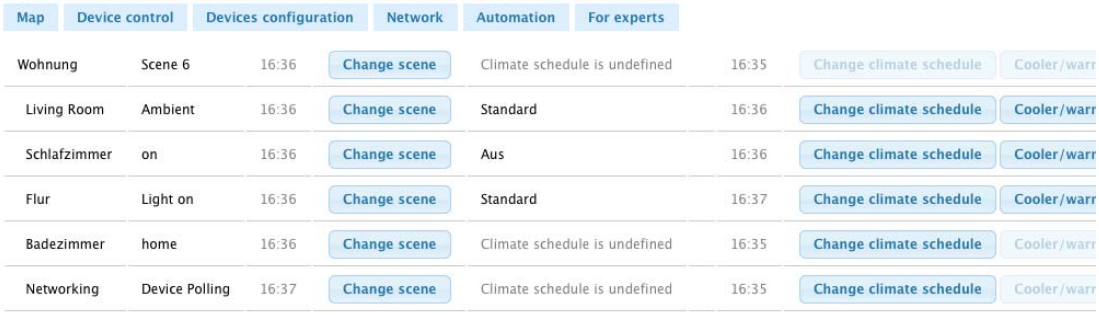
The example of a real network shown above allows the following conclusions:

1) Most nodes communicate most of the time without any problems.
2) The FLIRS node 2 shows a longer communication time but due to the nature of FLIRS this is acceptable.
3) Node 4 and Node 11 had one failed communication. For battery powered devices it sometimes happens that the final „go back to sleep“ command is sent out too late and the device is already back in sleep mode. In this case the particular communication may fail which is not necessarily an indication for a network problem.
4) Node 14 shows failed communication and a lot of frames that were only successfully transmitted after several attempts. Since no other routes succeeded it can be concluded that the device is just at the edge of the wireless range of the controller and using other routes do not offer any advantage. Here it would be recommended to place another node between node 14 and any other node which may act as router in this case. The fact, that sometimes the controller can reach Node 14 (indicated by „1“) directly without routing suggests that one additional device between the controller and Node 14 will certainly improve the communication.

## 3.6 Daily Usage

This section describes interface tabs to access functions of the Z-Wave devices.

### 3.6.1 Zone states

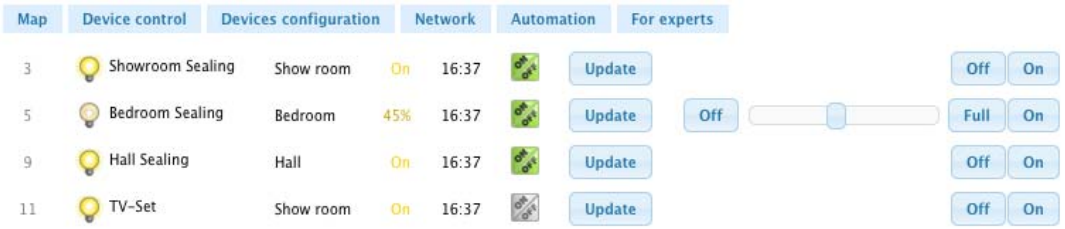



The zone state tab gives an overview of the current status of the different zone. Status refers to the last activated scene and the active climate schedule.

A drop down menu allows activating all scenes defined for the zone selected.

A second drop down list allows changing the climate control schedule per zone. The cooler/warmer buttons allows moving all set temperatures of the active climate control schedule in one of the other direction. This overwrite remains active until it is reversed. For further information about climate control schedule and overwrite refer to the chapter “Climate Control Schedule”.

### 3.6.2 Switch Overview




This page gives a table style overview of all actuators of the Z-Wave network. Actuators are devices with some kind of switching function such as

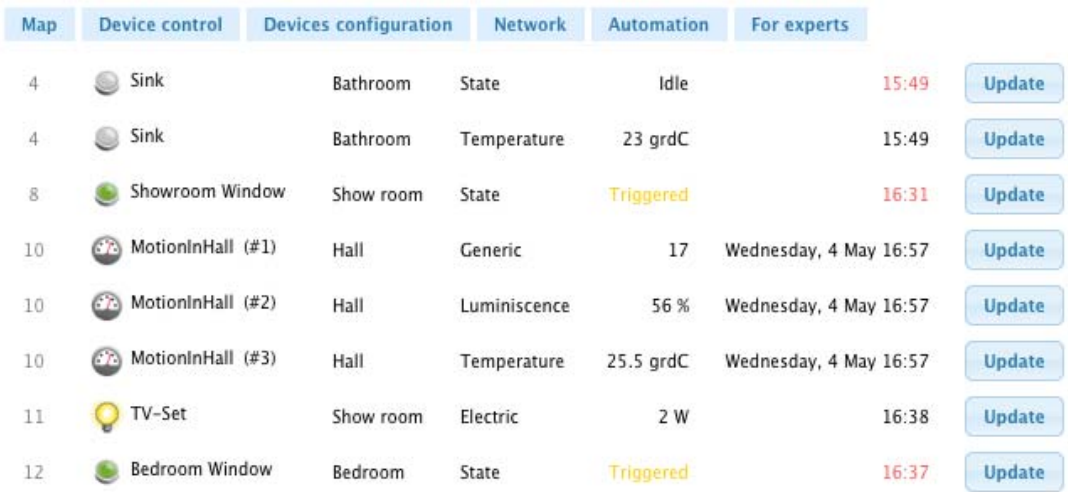
- Digital (on/off) switches,
- Light Dimmer,
- Motor Controls for Venetian blinds, window bind,
- Motor Control to open/close doors and windows.

Beside the name of the device, the location and the type of device the actual status and the timestamp of this status are shown.

Of course it is possible to switch the devices and to update the status of the device.

A little icon indicates how the device will react to a “switch all devices” command (will switch, will not switch, will react to off command only or to on command only).

### 3.6.3 Sensor Overview



This page gives a table style overview of all sensors of the Z-Wave network. Sensors are devices able to report measured values. Sensors can report binary or analog values. Beside the name of the device, the location and the type of sensor the actual sensor value and the timestamp of this value are shown. It is possible to ask for an update of the sensor value.

### 3.6.4 Meter Overview

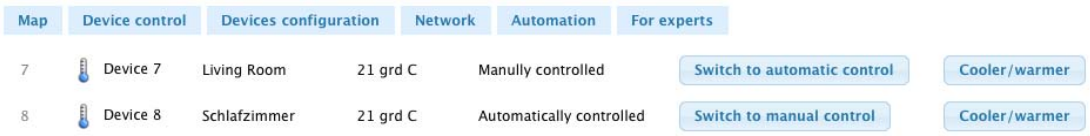
This page gives a table style overview of all meters of the Z-Wave network. Meters are devices able to report accumulated values. Beside the name of the device, the location and the type of meter the actual meter value and the timestamp of this value are shown. It is possible to ask for an update of the meter value.

### 3.6.5 Thermostats

Z-Way allows setting the desired temperature of the thermostat in two basic modes:

- manually,
- using climate control schedules.

The dialog „Thermostats“ in the menu „Device Control“ shows all thermostats in the network with its actual target temperature and the actual controlling mode.




After inclusion the thermostat is always in the manual mode. The mode of the device is indicated in the thermostat dialog for each individual device. This means that the thermostat will control the heat to reach the temperature shown. The button „Cooler/Warmer“ allows changing the target temperature.

*Attention: If the target temperature is changed this change will not be shown immediately in the user interface, since this change is not sent to the sleeping device but stored in Z-Way only. This status is indicated by a red temperature value. Once the new value is stored in the device – after the next wakeup – the value will be updated and turned back into back.*

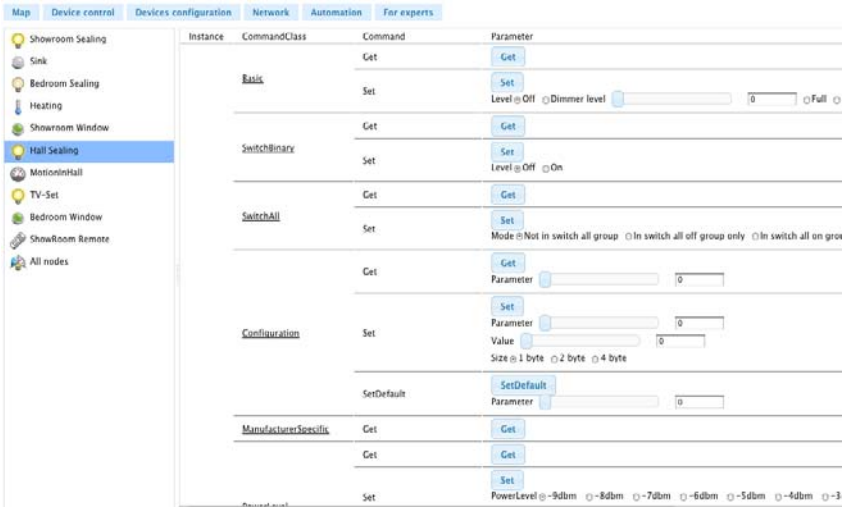
A second button on the thermostat device interface allows to switches to a mode, where the thermostat is controlled by a climate control schedule. This will only work, if a climate control schedule is defined for a particular zone and the respective thermostat is assigned to this zone.

In case the thermostat is controlled by a climate control schedule the “cooler/warmer” button can be use to overwrite the temperatures set by the schedule. For more information about climate control schedules and overwrite refer to the chapter “Climate Control Schedule”

## 3.7 Debugging / Expert Mode

The „Expert Mode“ button in the bottom context menu switches between a standard mode and expert mode. In expert mode some more technically oriented dialogs are available. It is recommended not to use the expert mode unless you are a Z-Wave expert and know what you do.

### 3.7.1 Expert Commands




The expert command tabs allows accessing all command class related data of the Z-Way controller as well as executing all command class related commands in a generic way. This interface overlay the more function-oriented tabs for switches, sensor etc. and can be used for debugging and testing purposes.

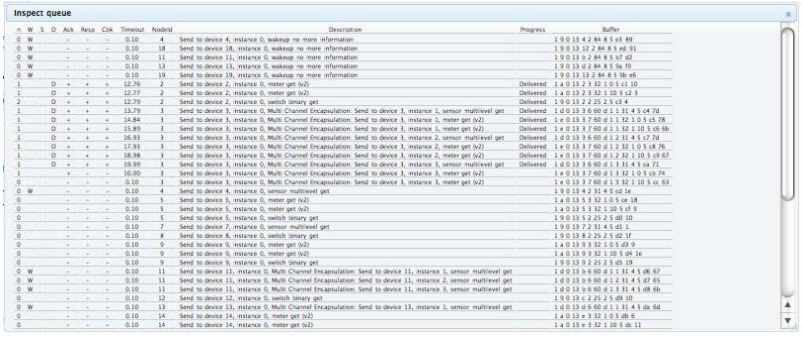
Just select the device on the left hand side and access the command interface on the right hand side. Clicking on the names of the Command Classes allows accessing the full data structure of this particular command class of the device chosen.

### 3.7.2 Controller Information


The controller information tab shows all conttoller information. The buttons “Show Controller Data” shows the internal Z-Way data structure related to the specific controller function of the controller device. The button “Show controller device data” show the generic device related data of the controller device.

The information given on this page is only relevant for advanced Z-Wave developers and for debugging.

### 3.7.3 Queue Inspection



In expert mode the Network management tab shows a button “Inspect queue” allows monitoring the work of the job execution of the Z-Way backend. Every communication with the Z-Wave transceiver is scheduled into a job and queued that it can transmitted over the serial hardware interface.

The table shows the active jobs with their respective status and additional information.

n: This column shows the number of sending attempts for a specific job. Z-Way tries three times to dispatch a job to the transceiver.

W,S,D: This shows the status of the job. If no indicator is shown the job is in active state. This means that the controller just tries to execute the job. W- states indicated that the controller believes that the target device of this job is in deep sleep state. Jobs in “W” state will remain in the queue to the moment when the target devices announces its wakeup state by sending a wakeup notification to the controller. Jobs in “S” state remain in the waiting queue to the moment the security token for this secured information exchanged was validated.
“D” marks a job as done. The job will remain in the queue for information purposes until a job garbage collection removed it from the queue.

ACK: … shows if the Z-Wave transceiver has issued an ACK message to confirm that the message was successfully received by the transceiver. This ACK however does not confirm that the message was delivered successfully. A successful delivery of a message will result in a “D” state of this particular job.

If the ACK field is blank, then no ACK is expected. A “.” indicates that the controller expects an ACK but the ACK was not received yet. A “+” indicates that an ACK was expected and was received.

RESP: ... shows if a certain command was confirmed with a valid response. Commands are either answered by a response or a callback.

If the RESP field is blank, then no Response is expected. A “.” indicates that the controller expects a Response but the Response was not received yet. A “+” indicates that a Response was expected and was received.

Cbk: … shows that the z-wave transceiver has finally reported the status of the delivery. This may either be a success - resulting in a “D” state right away or a failure, which would either trigger a retransmission or an abort of the job.

If the Cbk field is blank, then no callback is expected. A “.” indicates that the controller expects a Callback but the Callback was not received yet. A “+” indicates that a Callback was expected and was received.

Timeout: …shows the time left until the job is de queued.

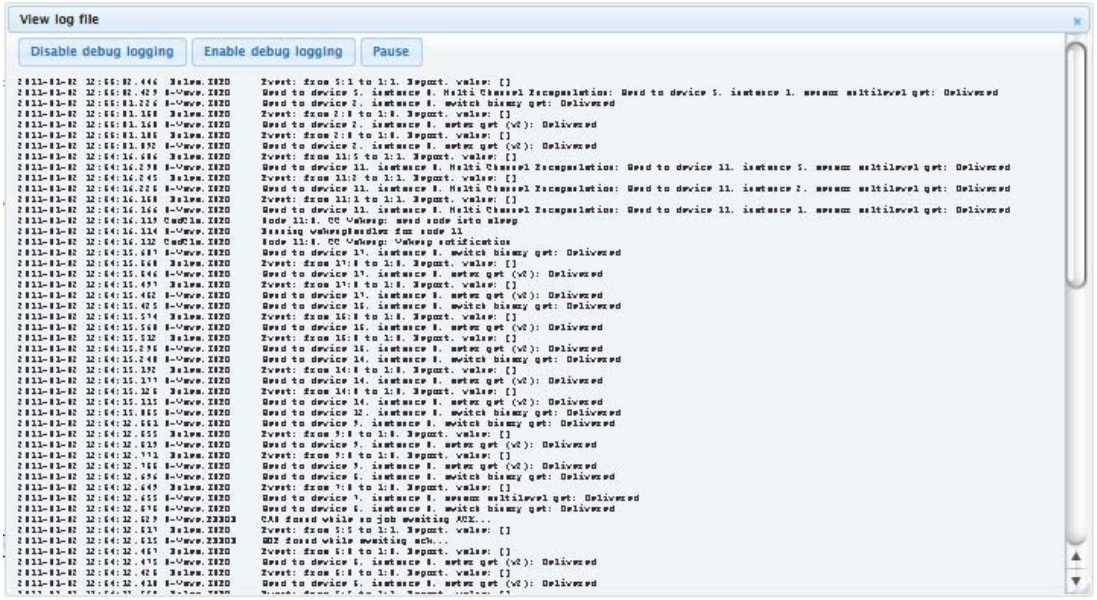
Node Id … shows the id of the target node. Communication concerning the network – like inclusion of new nodes – will have the controller node id as target node ID. For command classes command the node ID of the destination Node is shown. For commands directed to control the network layer of the protocol, the node id is zero.

Description: shows a verbal description of the job

Progress: shows a success or error message depending on the delivery status of the message. Since Z-Way tries three times to deliver a job up to 3 failure messages may appear.

Buffer: … shows the hex values of the command sent within this job.

### 3.7.4 Z-Way console




In expert mode the Network management tab shows a button “View Log file” allowing the access to the system logging.

Three more buttons on top of the console log window allow turning on and off the debugging mode of the log file that shows very detailed information about the information exchange between Z-Way and the Z-Wave transceiver chip. The “pause” resp. “continue” button allows controlling the data output of the window.

# PART II: Advanced Technical Information

The second part of this Manual is targeted to experienced Z-Wave professionals, the certification entity and software developers with an intention to use and enhance the code.

## 4 Supported Command Class

Z-Way fully complies with the Z-Wave standard.

The device will announce itself with the following device classes:

- Basic: Controller
- Generic: Static Controller
- Specific: PC Controller

This chapter describes the command classes used by Z-Way.

Attention: Certain Z-Wave hardware transceivers truncate the length of the Node Information Frame sent out. Certain of the supported command classes may not be announced correctly by the Z-Way controller into the Z-Wave net. This may cause malfunctions of the network. Z-Wave.Me hardware does not have this limitation.

### 4.1 Basic – Version 1

The basic device class is supported to allow receiving BASIC SET events from devices in order to allow scene activation.

Both SET and REPORT commands generate an event that is displayed on the console and can be used to trigger an action or activate a scene. The event contains information about the sending node, the value transmitted by REPORT or SET and the channel in case the multi channel emulation is used.

### 4.2 Manufacturer Specific – Version 1

The manufacturer specific command class reports IDs to identify the product within a Z-Wave Network:

| | |
|---|---|
| Vendor ID: | 0x115 (Z-Wave.Me) |
| Product ID: | 1 |
| Product Type ID: | 1 |

### 4.3 Version – Version 1

The version command class reports the versions of all other command classes plus its own firmware version.

The version command class furthermore reports the version of the Z-Wave library used in the Z-Wave serial interface used.

Additionally the version of the Z-Way code is reported as application software version:

Application Version Major: 1
Application Version Minor: 0

### 4.4 Multi Channel – Version 2

The multi channel command class is supported to allow receiving BASIC SET events from devices in order to allow scene activation and to allow differentiating between scene commands received from controller devices able to support multi channel devices. Please refer to manual chapter "Scene Switching Events" for more information.

The software supports 16 different channels; all supporting the BASIC command class only.

### 4.5 Switch Multilevel – Version 1

The controller emulates a multilevel switch to receive commands, which will be used for scene switching.

Both SET and REPORT commands generate an event, which is displayed on the console and can be used to trigger an action or activate a scene. The event contains information about the sending node, the value transmitted by REPORT or SET and the channel in case the multi channel emulation is used.

### 4.6 Clock – Version 1

The command class clock is supposed to provide the system time of the controller to all devices in the network able to ask for clock update.

### 4.7 Node Naming - Version 1

The node naming allows assigning a name and a location to a specific node. Z-way will report "Gateway" as default node name and an empty string as location.

### 4.8 Multi Command - Version 1

This command class announces that Z-Way is able to handle multi command encapsulated commands.

### 4.9 Simple A/V Class – Version 1

The implementation of the Simple A/V class allows controlling the Z-Way software from a remote capable of controlling Audio/Video devices. This makes particularly sense if the Z-Way software runs on a PC, where it can directly control Audi Video software such as Winamp. Special Scenes are needed to translate the AV control commands into control command of the respective software application.

### 4.10 Timing Command – Version 1

The timing command class is used to synchronize calendar information with devices in the network.

### 4.11 Climate Schedule

This command class is needed for full support of Danfoss Thermostat Values

# 5 Controlled Command Classes:

Z-Way controls Z-Wave command classes in a couple of different contexts:

## 5.1 Interview:

Right after inclusion of a device the controller interviews the device. The objective is to gain as much knowledge about the device characteristics as possible. Several command classes are only used during Interview. Other command classes offer specific commands to get information about capabilities that are used in the interview process. The interview process can be repeated by user interaction ("Force Interview" in Device Configuration Tab).

## 5.2 Auto Configuration:

Right after initial interview of the device the Z-Way controller will do some default setup actions, which are described in Chapter "Auto configuration". Auto configuration will only be executed if the interview was completed.
The auto configuration function can be turned off using a flag in /config/Defaults.xml

## 5.3 Device Configuration:

The user configuration dialog allows to manually configuring the device. This configuration includes the use of the configuration command class and other configuration options of the device. This also includes the configuration of associations.

## 5.4 User Actions

The usage interface allows getting and setting values, switching devices or turn devices into a desired status.

## 5.5 Events

Certain command class communications trigger a Z-Way event that may be used to activate scenes. Events are displayed on the console and can be access by the automation logic

## 5.6 Command Classes Usage

The following overview specifies the contexts in which the different command classes are controlled.

| | Interview | Auto Conf | User Conf | Action | Event |
|---|---|---|---|---|---|
| ALARM | X | X | | | X |
| ASSOCIATION | X | X | X | | |
| ASSOCIATION_COMMAND_CONF | X | X | X | | |
| BASIC | | | | X | X |
| BASIC_WINDOW_COVERING | | | | X | |
| BATTERY | | | | X | |
| CLIMATE_CONTROL_SCHEDULE | X | | | X | |
| CLOCK | | | | X | |
| CONFIGURATION | | | X | | |
| HAIL | | | | | |
| INDICATOR | | | | X | |
| MANUFACTURER_SPECIFIC | X | | | | |
| METER | X | | | X | |
| METER_PULSE | | | | X | |
| MULTI_CMD | | | | | |
| MULTI_INSTANCE | X | | | | |
| MULTI_INSTANCE_ASSOCIATION | X | X | X | | |
| NODE_NAMING | | | X | | |
| POWERLEVEL | | | X | | |
| PROTECTION | X | X | | X | |
| SCENE_ACTIVATION | | | | | X |
| SCENE_ACTUATOR_CONF | | | X | | |
| SCENE_CONTROLLER_CONF | | X | X | | |
| SENSOR_ALARM | | | | | |
| SENSOR_BINARY | | | | X | X |
| SENSOR_CONFIGURATION | | | | X | |
| SENSOR_MULTILEVEL | X | | | X | X |
| SILENCE_ALARM | | | X | | |
| SWITCH_ALL | | X | X | X | |
| SIMPLE_AV | X | | | X | X |
| SWITCH_BINARY | | | | X | X |
| SWITCH_MULTILEVEL | X | | | X | X |
| SWITCH_TOGGLE_BINARY | | | | X | |
| THERMOSTAT_MODE | X | | | X | |
| TIME | | | X | | |
| TIME_PARAMETERS | | | X | | |
| VERSION | X | | | | |
| WAKE_UP | X | X | X | | |

# 6 Scene Switching Events

Static controllers such as Z-Way support definition and activation of scenes. Beside timers or UI actions causing the activation of a scene the scene activation shall also be possible from other Z-Wave devices in the network such as wall controllers or remote controls. In order to do this the controller need to identify, which scene the user is intended to activate. There are a couple of approached in Z-Way how to achieve this.

## 6.1 Scene Activation

Certain Z-Wave controllers explicitly support the activation of scenes. Their different buttons can be configured to send enumerated scene activation commands, which are received by the controller. Controllers with scene switching capability can be used for scene switching without further work and constrains.

## 6.2 Associations

Associations are used to setup switching relationship between a controlling and a controlled Z-Wave device. Typically controlling devices send a BASIC SET command to perform a switching function in the device controlled.

If these BASIC switching commands are sent to the controller these commands can be “misused” and interpreted as scene switching command. The big challenge is that the BASIC SET command does not allow sending any scene number information. Rather the BASIC SET command typically only supports the values 0x00 and 0xff. As long as the sender only has one push button or just one association group supported, a received BASIC SET COMMAND from one particular device can be referred to a scene easily.

If the controlling device has more than one association group the controller is not able to differentiate between the different groups – all sending the very same BASIC Set command.

However some tricks can help to solve this problem.

A) Multi Channel Association

   If the controlling device can send multi channel commands it can be configured in a way to send different scene activation commands (still as BASIC SET or BASIC REPORT with values 0x00 or 0xff) to different channels helping the receiving controller to distinguish different commands and activate different scenes accordingly. In order to use this “trick” the controlling device must support multi channel association.

B) Virtual Nodes

The controller is able to act as multiple virtual nodes in the network. It will then not receive commands for one single Node ID but for various Node IDs. Associations can be set to these different Node IDs that are all received by the same physical device – the controller. The controller is now able to distinguish different commands and activate different scenes accordingly. In order to use this "trick" the hardware used for the controller (USB Stick) must support bridges devices. Only few USB interface devices do this at the moment.

C) Association Configuration

Certain devices have the capability to allow configuring which command to be sent in case of an association group event. If the standard BASIC set can be changed to a command that includes information about the association group that causes sending a command the controller is able to use such a command to activate scenes. In order to use this "trick" the controlling hardware needs to support the association configuration command class.

The following table gives an overview how to use certain devices for scene switching.

| Method | Vendor/Products |
|---|---|
| Scene Configuration | • QEES<br>• Remotek |
| Multi Channel Association | • Merten<br>• Z-Wave.Me |
| Association Configuration | • Aeon Labs |
| Virtual Devices | • ACT dual paddle |
| No Scene Switching possible | • Tricklestar<br>• Duwi Remote Control |
| Standard association sufficient | • ACT single paddle<br>• All common motion detectors and door/window sensors |

## 6.3 Auto Configuration for Scene Activation

The auto configuration function makes sure that users will receive scene activation events without further setup and configuration work. Therefore several settings are done during the auto configuration process

**Association (devices will use BASIC command class):**

The Node ID of the controller is set into every association group of the device included. Therefore every SET/REPORT command sent to a specific group will be received by the controller but not necessarily referred to a specific group.

**Multi Channel Association:**
The Node ID of the controller is set into every association group and the target instance equals the number of the association group. Therefore the controller is able to distinguish different received SET and REPORT commands according to the group, which initiated this command.

**Multilevel Switch:**
The controller receives SET and REPORT commands for its multi level switch emulation and can use the multi level value to distinguish different events based on this value. However the sending device must be able to precisely set a specific multilevel switch value for the emulated device in the controller.

**Scene Controller Configuration:**
The controller will receive scene activation events defined by the scene controller configuration command class. The command class assigns scene IDs according to the group IDs used in the configured device (Group ID = Scene ID).

**Association Command Class Configuration:**
The controller configures the association commands sent to its own node ID as scene activation commands with the group ID set as scene ID. Association commands sent to other nodes than the controller itself are left untouched.

# 7. Controllers Architecture

## 7.1 Startup

The code starts (Main.py) by activating the ZWaveAPI, which will first initialize the serial API and detect the various capabilities of the API. Depending on these capability reported the ZWaveAPI will load and instantiate the function classes from FunctionClasses/*.py

As next step the code will read the list of current nodes from the Z-Wave hardware chip and instantiate their data structures. The most important property of a node is the list of command classes supported.

After activation of the ZwaveAPI the code will activate two more services.

The web server (webserver.py) is started and bound to a defined IP Port. This server is able to receive commands via JSON and reports back values via JSON

An automation service (Rulesmanager.py, Scheduler.py) is started which is able to trigger actions based on different conditions.

## 7.2 User Interface

All User interaction is done via the JSON interface of the web server. The user interaction includes

- Access to API function classes such as “AddNodeToNetwork” or “SetDefault”
- Access to higher layer back end functions such as “HealNetwork” or “LoadData”
- Access to all exposed command class functions such as “Basic Get” or “Configuration Set”
- Access to all controller data (API versions, library used, … )
- Access to all device specific data of each node (node ID, generic Device Class, SDK, … )
- Access to all command class specific data (Configuration value, sensor data, … )
- Access to definition of polling and scenes handling

The default user interface is intentionally designed with very basic graphics but exposes all JSON functionality for testing and usage.

Users are invited to write and share their own UI based on the services mentioned.

## 7.3 Scripting

Please refer to the section Scripting.

## 7.4 Inclusion Process

The inclusion process – regardless if a controller or a device is to be included – is started by the user interface (Button "Include Start"). The Z-Wave hardware is turned into an inclusion mode and waits for a device to respond. After the successful inclusion of a node the corresponding device structure is generated and the command classes information from the inclusion process is used to load all command classes.
The following interview process has four steps. As first step the Version command class is interviewed to get all version information for the command classes to be interviewed. If no version command class is announced this step is skipped and all versions are assumed to be "1". The second step is to interview the manufacturer specific command class. This allows identifying the Device Data Record already after step 2.
The third step will interview the multi channel or multi instance command class if such a command class was announced. As a result the number of channels and the structure of the channels are determined and data structures for each instance/channel is created.
The final step interviews all command classes of all instances and stores the reported values in the command class data structures.

The user need to make sure that the device is awake during the interview process in order to complete the interview. Battery operated devices may go to sleep during the interview process leaving this process uncompleted.
Every wakeup notification of the device (NIF received are used as Wakeup notifications as well) will resume the interview process until its finished.

To avoid infinite loops by broken devices with protocol violations the interview process will break up after 10 unsuccessful attempts to complete.

## 7.5 File Overview

| | |
|---|---|
| CommandClasses/*: | Implementation of Z-Wave Classes |
| config/ | |
| Defaults.xml: | Definition of Default Values |
| Rules.xml | Definitions of Scenes, Scenes Events and Actions |
| FunctionClasses/* | Implementation of ZWave API functions |
| htdocs/* | Webserver root |
| serial/* | Py class for serial handling |
| translations/ | |
| Alarms.xml | String translations of Alarm Types |
| DeviceClasses.xml | String translations of Device Classes |

| | |
|---|---|
| Scales.xml | String translations of Meter and Sensor types/scales |
| SDKIds.xml | String translations of SDK versions |
| VendorIds.xml | Vendor Information |
| ZDDX/ | |
| generated/ | |
| HID_NID.xml | generated ZDDX files of devices |
| ID_DeviceData.xml | central value storage file |
| managed/* | |
| Common.py | Library of commonly used functions |
| DeviceClasses.py | DeviceClass handling |
| Logging.py | Central logging class |
| Main.py | Startup code |
| PollManager.py | Manager of Command Class polls |
| RulesManager.py | Manager for Rules/Scenes |
| Scheduler.py | Scheduler |
| SerialAPI.py | Access to Serial Interface |
| Tracking.py | Usage tracking |
| Translations.py | Handles the XML translation files |
| WebServer.py | The JSON and Web server |
| XMLObject.py | XML File handling |
| ZDDX.py | Device Description File handling |
| ZWaveAPI.py | Defines Devices, Instances , etc. |

# 8 Scripting

Z-Way allows executing user written python scripts when a scene is activated. This gives huge flexibility to enhance the Z-Wave network with Internet based function or more intelligent logic.

For more information about the scripting language python please refer to www.python.org or the online tutorial at docs.python.org/tutorial.

Beside general capabilities of the language it is of course possible to access all functionalities of the Z-Way backend. This refers to access to all variables of the data model and the execution of all Command Class Commands and API Function calls. Please note that it is strictly forbidden to use the time.sleep() function of python inside a script since it will block the rules manager and interfere with all other unhidden and hidden processes of the Z-Way.

**Attention: Wrong usage of Z-Wave API Calls or Z-Wave Command Class Commands may result in malfunctions of the controller and/or the whole network. Hence – handle with care!**

## 8.1 Access to data

The additional UI information given in the expert mode is a great reference to the data model of Z-Way.

All data is organized in a hierarchical tree of devices, instances of devices and command class of the instances of the devices.

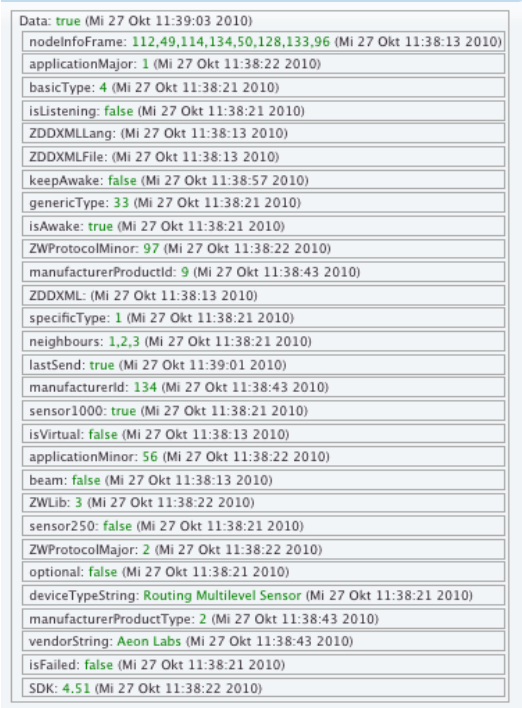
Each device, each instance and each command class has an object called data. This object data contains all accessible variables.
Examples

```
ZWaveAPI.devices[nodeId].data
# this accesses the data object of node id nodeId

ZWaveAPI.devices[nodeId].instances[0].data
# this accesses the data object of the instance 0 of node
id nodeId

ZWaveAPI.devices[nodeId].instances[0].commandClass[0x31].
data
# this accesses the data object of command class
Multilevel sensor (0x31) of the instance 0 of node id
nodeId
```

Some remarks:

- Devices without Multi Channels capabilities - which are most of the current devices on the market - only have instances[0] instantiated
- Please refer to the Annex A for the IDs of the different command classes
- Please refer to the Zensys Documentation SDS11060 for information about the different command classes and their behavior, calling parameters, values and constrains.
- Do **NEVER** use the time.sleep() function to make a delay in scene script - this will hang up the whole Z-Way automation engine. Instead use Z-Way 'timed scene'

The UI tab “Expert Commands” gives you insight on the values each of the data objects contains. Just click on the node ID or the instance ID or the command class ID for a pop up window that visualizes the respective data object.



Data Model of a Device

The names of the different variables are self-explaining. Each variable is stored in an object with the following properties.

Variable.value = the value itself
Variable.type = type of value
Variable.updated = flag which shows valitity
Variable.updateTime = (UNIX) timestamp of the last change
Variable.invalidateTime = (UNIX) timestamp of the last invalidation

The following two examples show how to access a variable

```
ZWaveAPI.devices[nodeId].data.vendorString.value
```

return the vendor string of this device.

```
ZWaveAPI.devices[nodeId].instances[0].commandClasses[0x31
].data.val.value
```

returns the sensor value of the sensor of the device Node Id using the command class 0x31 (multilevel sensor)

A test

```
if ZWaveAPI.devices[nodeId].instances[0].commandClasses[0x31].data.val.updated:
        Do_something
```

Will only execute do_something if the value of the sensor is valid.

## 8.2 Execution of Command Class Commands

The command class commands can be called using the same hierarchy as for the data. Instead of accessing the data object the public command class function is called directly. The expert page "Expert Commands" has a generic interface to all command class commands. All results of the commands are accessible in the data model.

Most public functions of the command classes are either SET() or GET(), with or without additional parameter. Example:

```
ZWaveAPI.devices[nodeId].instances[0].commandClasses[0x31
].Get()
```

will send a get command to a multilevel sensor with node ID nodeID. The node will (hopefully) generate a Multilevel sensor report which will update the

```
ZWaveAPI.devices[nodeId].instances[0].commandClasses[0x31
].data.val
```

value.

Please be aware that every get() command will invalidate the corresponding values in the data object and a received report command will validate and update the values again. With this flag mechanism its possible to track if devices have successfully updates their corresponding values in the data objects.

## 8.3 Execution of Z-Wave API function calls

As mentioned above the usage of Z-Wave API function calls is very dangerous and may result in complete malfunction of the network and Z-Way itself. You

should only use these functions if you are a Z-Wave expert and know what you do.

The tab “Controller Info” lists all available API calls of the given Z-Wave Hardware with their names and ID. Please refer to the Zensys document “Z-Wave ZW301 Application Programmers Guide INS10247” for information about calling parameters, return values and other constrains.

The document INS10247 is available for owners of a Z-Wave SDK only.

Calls to API functions are quite simple.

Example: Call to request a Node NeighborUpdate, one parameter is the node ID of the nodes requested

```
#start script
ZWaveAPI.RequestNodeNeighborUpdate(nodeId)
```

The result of the call is found in the data model of ZWaveAPI after the function was completed.

## 8.4 Examples

On default Z-Way has a couple of scripts, which poll devices and record device values. They may be used as example on how to access devices and public function calls from the scene-scripting interface. For more examples of scripts and recipes refer to the online user forum of Z-Way

# 9 What's left

There is a couple of tasks left for future versions of Z-Way:

- Network Wide Power Level Test and generating of a 2D network map based on link quality information
- Secure Command Class
- Firmware Update CC supported in Frontend
- User Dialog for initial setup of Temperature display, etc.
- Advanced Energy Framework CCs (DCP List Configuration DCP List Monitoring Meter Table Configuration Meter Table Monitoring Meter Table Push Configuration Prepayment Encapsulation Rate Table Configuration Rate Table Monitoring Tariff Table Configuration Tariff Table Monitoring, HRV*)
- Audio Video Framework CCs (Content Directory, Search_NMD, Renderer_status, Tagging_MD, Screen Attributes, Screen_MD)
- Sensor Value History
- Some Thermostat CCs (Operating State, Setback, Fan_Mode, Fan_State, Heating, Chimney Fan)
- Some rare CCS (6LoPAN, ZIP, User_Code, Remote Association, MTB_Window, Language, Group-Naming, Geographic Location, Replication)
- Door/Lock CCs(Lock, Schedule Entry Lock, Door_Lock, Door_Lock_Logging)

## Annex A: Z-Wave Command Classes

```
#define COMMAND_CLASS_NO_OPERATION                      0x00
#define COMMAND_CLASS_ALARM                             0x71
#define COMMAND_CLASS_BASIC                             0x20
#define COMMAND_CLASS_CONTROLLER_REPLICATION            0x21
#define COMMAND_CLASS_APPLICATION_STATUS                0x22
#define COMMAND_CLASS_SWITCH_BINARY                     0x25
#define COMMAND_CLASS_SWITCH_MULTILEVEL                 0x26
#define COMMAND_CLASS_SWITCH_ALL                        0x27
#define COMMAND_CLASS_SWITCH_TOGGLE_BINARY              0x28
#define COMMAND_CLASS_SWITCH_TOGGLE_MULTILEVEL          0x29
#define COMMAND_CLASS_CHIMNEY_FAN                       0x2A
#define COMMAND_CLASS_SCENE_ACTIVATION                  0x2B
#define COMMAND_CLASS_SCENE_ACTUATOR_CONF               0x2C
#define COMMAND_CLASS_SCENE_CONTROLLER_CONF             0x2D
#define COMMAND_CLASS_SENSOR_BINARY                     0x30
#define COMMAND_CLASS_SENSOR_MULTILEVEL                 0x31
#define COMMAND_CLASS_METER                             0x32
#define COMMAND_CLASS_METER_PULSE                       0x35
#define COMMAND_CLASS_THERMOSTAT_HEATING                0x38
#define COMMAND_CLASS_THERMOSTAT_MODE                   0x40
#define COMMAND_CLASS_THERMOSTAT_OPERATING_STATE        0x42
#define COMMAND_CLASS_THERMOSTAT_SETPOINT               0x43
#define COMMAND_CLASS_THERMOSTAT_FAN_MODE               0x44
#define COMMAND_CLASS_THERMOSTAT_FAN_STATE              0x45
#define COMMAND_CLASS_CLIMATE_CONTROL_SCHEDULE          0x46
#define COMMAND_CLASS_THERMOSTAT_SETBACK                0x47
#define COMMAND_CLASS_BASIC_WINDOW_COVERING             0x50
#define COMMAND_CLASS_MTP_WINDOW_COVERING               0x51
#define COMMAND_CLASS_MULTI_INSTANCE                    0x60
#define COMMAND_CLASS_DOOR_LOCK                         0x62
#define COMMAND_CLASS_USER_CODE                         0x63
#define COMMAND_CLASS_CONFIGURATION                     0x70
#define COMMAND_CLASS_MANUFACTURER_SPECIFIC             0x72
#define COMMAND_CLASS_POWERLEVEL                        0x73
#define COMMAND_CLASS_PROTECTION                        0x75
#define COMMAND_CLASS_PROTECTION_V2                     0x75
#define COMMAND_CLASS_LOCK                              0x76
#define COMMAND_CLASS_NODE_NAMING                       0x77
#define COMMAND_CLASS_FIRMWARE_UPDATE_MD                0x7A
#define COMMAND_CLASS_GROUPING_NAME                     0x7B
#define COMMAND_CLASS_REMOTE_ASSOCIATION_ACTIVATE       0x7C
#define COMMAND_CLASS_REMOTE_ASSOCIATION                0x7D
#define COMMAND_CLASS_BATTERY                           0x80
#define COMMAND_CLASS_CLOCK                             0x81
#define COMMAND_CLASS_HAIL                              0x82
#define COMMAND_CLASS_WAKE_UP                           0x84
#define COMMAND_CLASS_ASSOCIATION                       0x85
#define COMMAND_CLASS_VERSION                           0x86
#define COMMAND_CLASS_INDICATOR                         0x87
#define COMMAND_CLASS_PROPRIETARY                       0x88
#define COMMAND_CLASS_LANGUAGE                          0x89
#define COMMAND_CLASS_TIME                              0x8A
#define COMMAND_CLASS_TIME_PARAMETERS                   0x8B
#define COMMAND_CLASS_GEOGRAPHIC_LOCATION               0x8C
#define COMMAND_CLASS_COMPOSITE                         0x8D
#define COMMAND_CLASS_MULTI_INSTANCE_ASSOCIATION        0x8E
```

```
#define COMMAND_CLASS_MULTI_CMD                                0x8F
#define COMMAND_CLASS_ENERGY_PRODUCTION                        0x90
#define COMMAND_CLASS_MANUFACTURER_PROPRIETARY                 0x91
#define COMMAND_CLASS_SCREEN_MD                                0x92
#define COMMAND_CLASS_SCREEN_ATTRIBUTES                        0x93
#define COMMAND_CLASS_SIMPLE_AV_CONTROL                        0x94
#define COMMAND_CLASS_AV_CONTENT_DIRECTORY_MD                  0x95
#define COMMAND_CLASS_AV_RENDERER_STATUS                       0x96
#define COMMAND_CLASS_AV_CONTENT_SEARCH_MD                     0x97
#define COMMAND_CLASS_SECURITY                                 0x98
#define COMMAND_CLASS_AV_TAGGING_MD                            0x99
#define COMMAND_CLASS_IP_CONFIGURATION                         0x9A
#define COMMAND_CLASS_ASSOCIATION_COMMAND_CONFIGURATION        0x9B
#define COMMAND_CLASS_SENSOR_ALARM                             0x9C
#define COMMAND_CLASS_SILENCE_ALARM                            0x9D
#define COMMAND_CLASS_SENSOR_CONFIGURATION                     0x9E
#define COMMAND_CLASS_MARK                                     0xEF
#define COMMAND_CLASS_NON_INTEROPERABLE                        0xF0
```