How to connect Volvo on Call to Domoticz

Table of Contents

1 Introduction....................1
2 Preparation....................1
3 Node-Red flow VoC to Domoticz....................4
4 Node-Red flow Domoticz to VoC....................28
5 Temperature and Location....................32

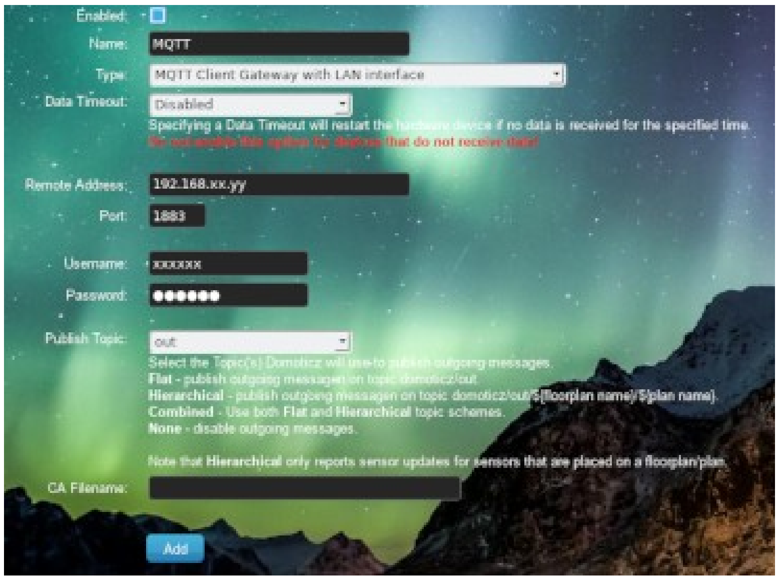
1 Introduction

It is assumed that you have installed the following programs and that they are functioning correctly:

- Volvo on Call python script.
  To download it, see: https://github.com/molobrakos/volvooncall
- MQTT broker. It is preferred to use Mosquitto.
  To download it, see: https://mosquitto.org/
- Node Red.
  To download it, see: https://nodered.org/

2 Preparation

In Domoticz install in “Hardware” MQTT Client Gateway with LAN interface



Enabled: ☐
Name: MQTT
Type: MQTT Client Gateway with LAN interface
Data Timeout: Disabled
Specifying a Data Timeout will restart the hardware device if no data is received for the specified time.
Do not enable this option for devices that do not receive data!

Remote Address: 192.168.xx.yy
Port: 1883

Username: xxxxxx
Password: ●●●●●●

Publish Topic: out
Select the Topic(s) Domoticz will use to publish outgoing messages.
Flat - publish outgoing messages on topic domoticz/out.
Hierarchical - publish outgoing messages on topic domoticz/out/$(floorplan name)/$(plan name).
Combined - Use both **Flat** and **Hierarchical** topic schemes.
None - disable outgoing messages.

Note that **Hierarchical** only reports sensor updates for sensors that are placed on a floorplan/plan.

CA Filename:

Configure the IP address of the MQTT broker (server). If Domoticz and the MQTT server are installed on the same hardware you can fill in “localhost” (without “ ”). It is recommended to leave the port as default (1883). If you have secured your MQTT broker with a user name and a password (recommended) fill in these in the corresponding fields. Leave ”Publish Type” as “out”.

The Volvo on Call script will publish various sensors, both binary and analogue. Beside that we want to send some commands to our Volvo. Therefore we will need one or more switches

In order to create these sensors and switches create in “Hardware” a Dummy (Does nothing, use for virtual switches only).

Each user has to check, which sensors and commands are supported for his/her Volvo.
To do so, go to the command line and issue the following command: “*voc print*”.
The result will be that all all available sensors and functions are listed.

We have to create several virtual sensors. For the binary sensors, which will present 2 values (ON/OFF, Open/Close, Safe/Unsafe) we have the option to create an “Alert” sensor or a simple “Text” sensor. The analogue sensors are “Custom” sensors, with the correct index on the y- axis. The most important difference between the “Alert” sensor and the “Text” sensor is that the “Alert” sensor is able to send notifications and that it makes use of different colours, to attract attention. The user can decide for themselves which function requires an “Alert” or a “Text” sensor.

In this document we will use the “Alert” sensor for the windows and the doors. The other binary sensors are ”Text” sensors.

So we have to create the following virtual sensors:

- 4 “Alert” sensors for the front left window, the rear left window, the front right window and the rear right window.
- 6 “Alert” sensors for the front left door, the rear left door, the front right door, the rear right door, the hood and the tailgate.
- 4 “Text” sensors for the front left tyre pressure, the rear left tyre pressure, the front right tyre pressure and the rear right tyre pressure.
- 8 “Text” sensors for the following functions: brake fluid, washer fluid, service warning, bulb failures, latest trip, car locked, heater(if applicable) and engine running.

The next sensors to configure are the other (analogue) sensors, as follows:

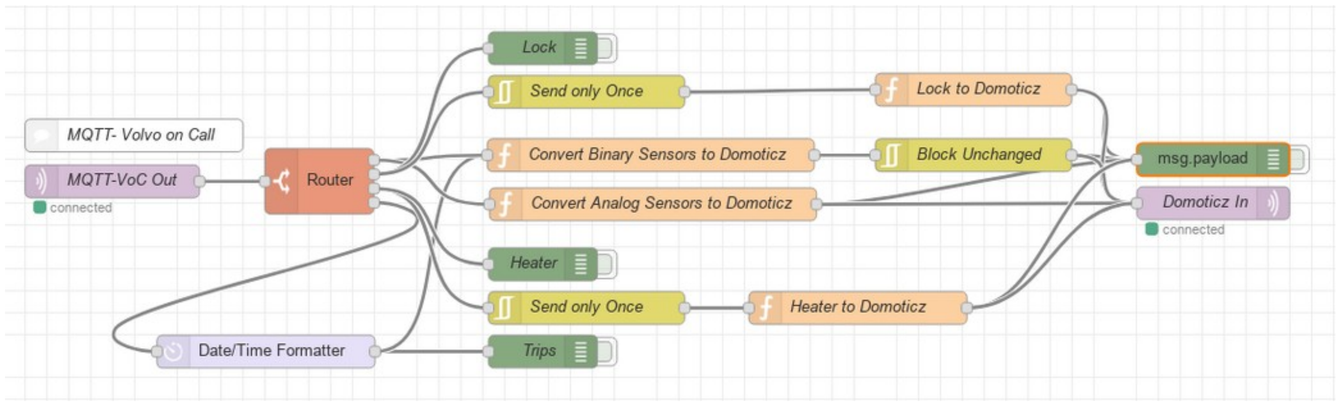
- Custom sensor for “Fuel Amount”, with axis label L
- Percentage sensor for “Fuel Amount Level”
- Custom sensor for “Fuel Consumption”, with axis label L/100 km
- Custom sensor for “Average Speed”, with axis label km/h
- Custom sensor for “Distance to Empty”, with axis label km
- Custom sensor for “Odometer”, with axis label km

If your car provides more than these analogue sensors, create the corresponding sensors accordingly.

Go to “Setup” and then to “Devices”. Make a note of the Idx number of the newly created devices, as we will need these later.

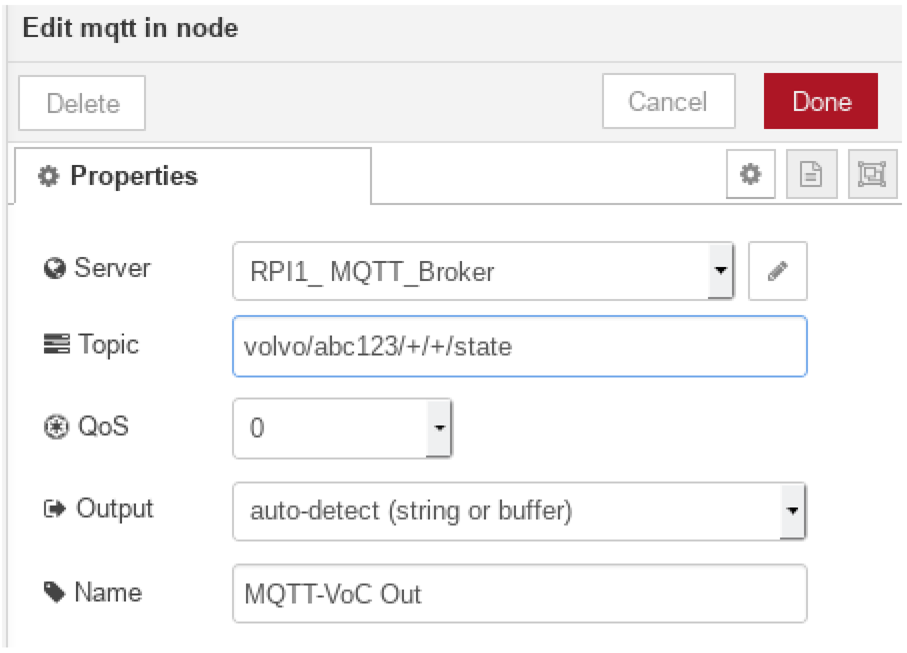
3 Node-Red flow VoC to Domoticz

In the picture below you will see the total flow



All nodes used, are core Node-Red nodes, except the node, called “Router” and the node, called “Date/Time Formatter”.
The node “Router” has to be installed from, either the “Palette” of Node Red or directly from the command line. In “Palette” search for a node called: “node-red-contrib-routing”, and install this node. For more information, see: https://flows.nodered.org/node/node-red-contrib-routing.
The node “Date/Time Formatter” has to be installed as well.
Search for: “node-red-contrib-moment” and install this node.
For more information, see: https://flows.nodered.org/node/node-red-contrib-moment

The first node in the flow is a MQTT input node.
The configuration of the node is as follows:


Edit mqtt in node

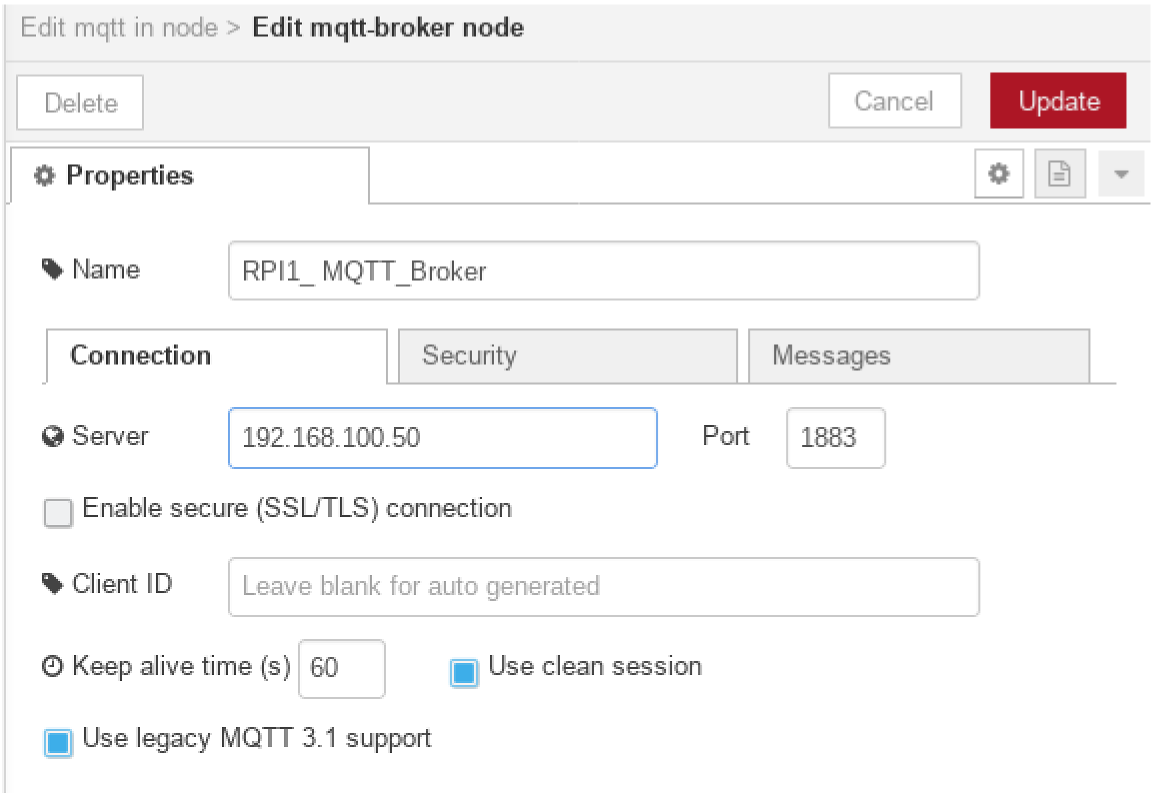
Server: RPI1_ MQTT_Broker
Topic: volvo/abc123/+/+/state
QoS: 0
Output: auto-detect (string or buffer)
Name: MQTT-VoC Out

The name of the Server is up to the user and can be anything, but will be used later, as well. The VoC python script publishes many topics to the configured MQTT server. Many are intended to be used with Home Assistant, in order to create the various devices. These topics will not be used in Node Red and Domoticz. We are only interested in the various statuses and therefore the Topic will be: “volvo/abc123/+/+/state”.

Note 1: Do not use a leading forward slash, such as “/volvo”.
Note 2: Replace “abc123” with your license plate number (on all occurrences in the flow).

The Quality of Service (QoS) is up to the user, but can be left default.
The Output, you can leave default as .well
The Name is up to the user, but it is recommended to use a name, that represents the function.

Next we have to configure the MQTT broker. To do so, press the pencil, right of the Server name. We start with the “Connection” tab.



Name: RPI1_ MQTT_Broker

Server: 192.168.100.50 Port: 1883

☐ Enable secure (SSL/TLS) connection

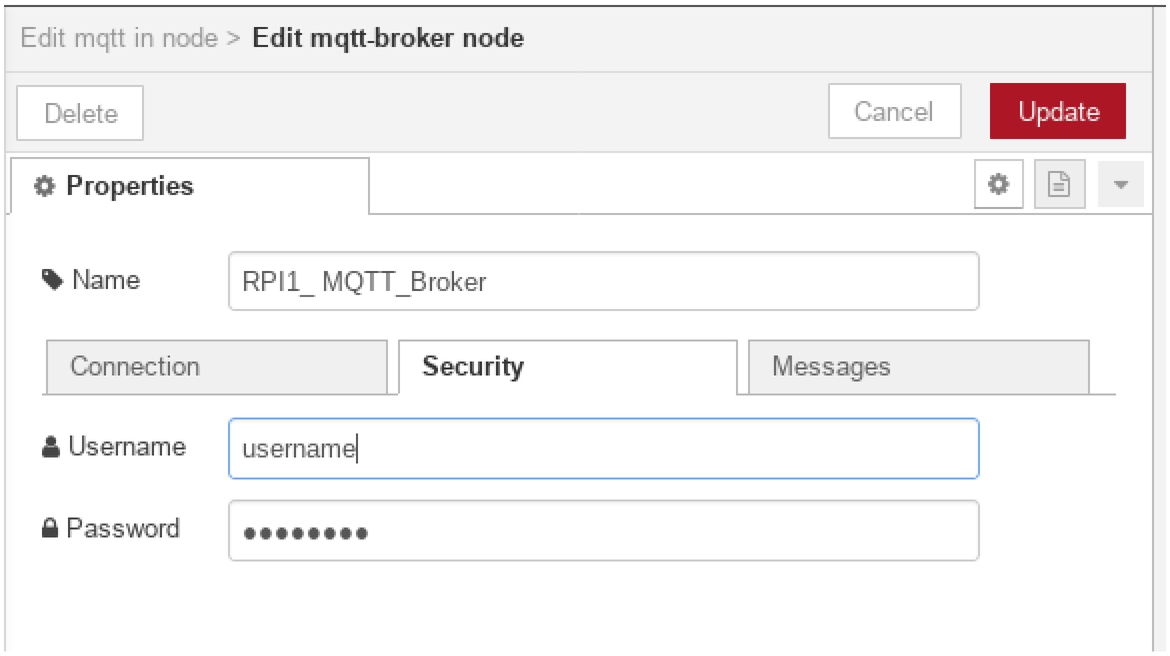
Client ID: Leave blank for auto generated

Keep alive time (s): 60 ✓ Use clean session

✓ Use legacy MQTT 3.1 support

First configure the IP address of the Server. If Node-Red and the MQTT server are installed on the same hardware you may use “localhost”. All other properties you can leave as indicated. Finally you have to press: “Update”.

The next tab is the Security tab.

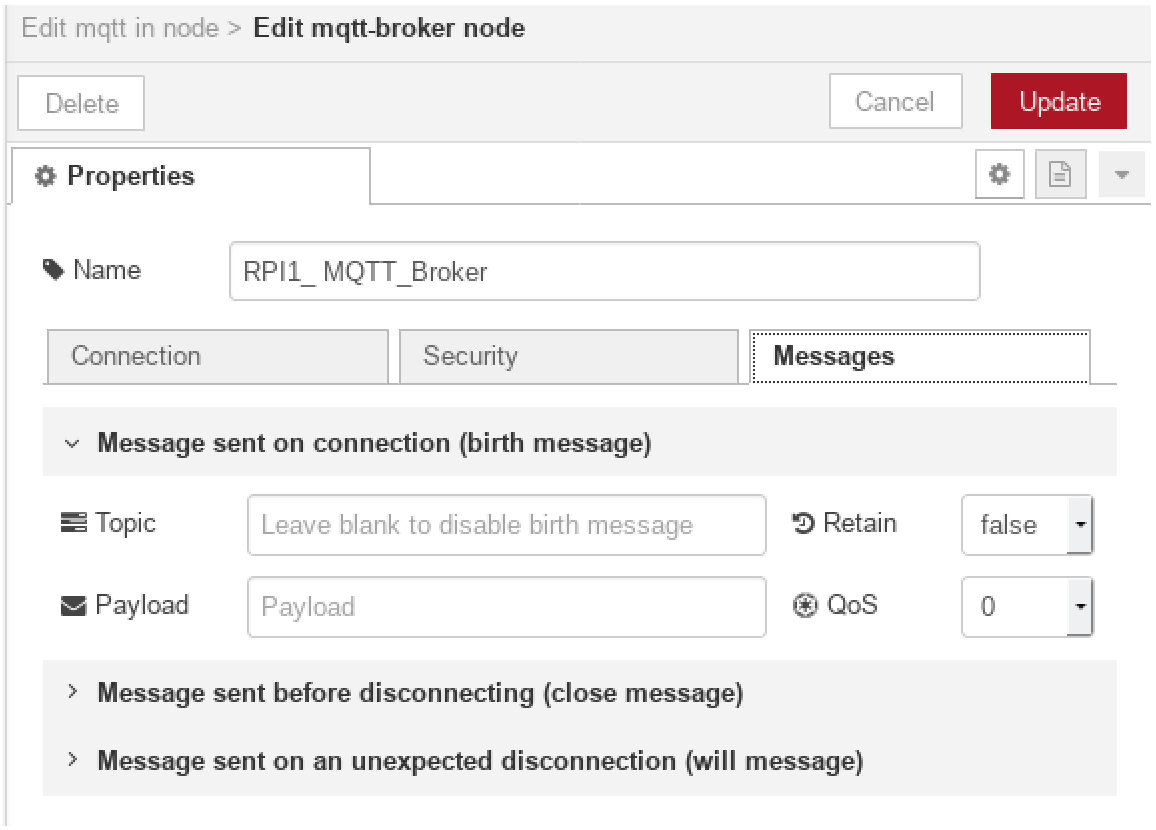


Name: RPI1_ MQTT_Broker

Username: username

Password: ●●●●●●●●●

If you have secured your MQTT broker with, at least, with a user name and a password (recommended) you should fill in your chosen Username and Password.
Press “Update” again,
The Messages tab you can leave as default.


Edit mqtt in node > **Edit mqtt-broker node**

Name: RPI1_ MQTT_Broker

Message sent on connection (birth message)

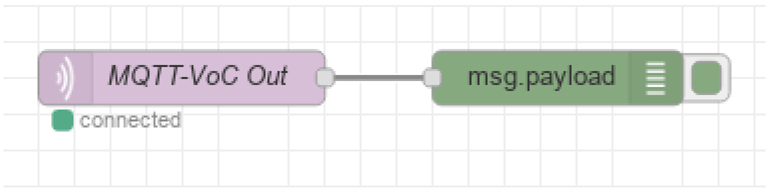
Topic: Leave blank to disable birth message — Retain: false

Payload: Payload — QoS: 0

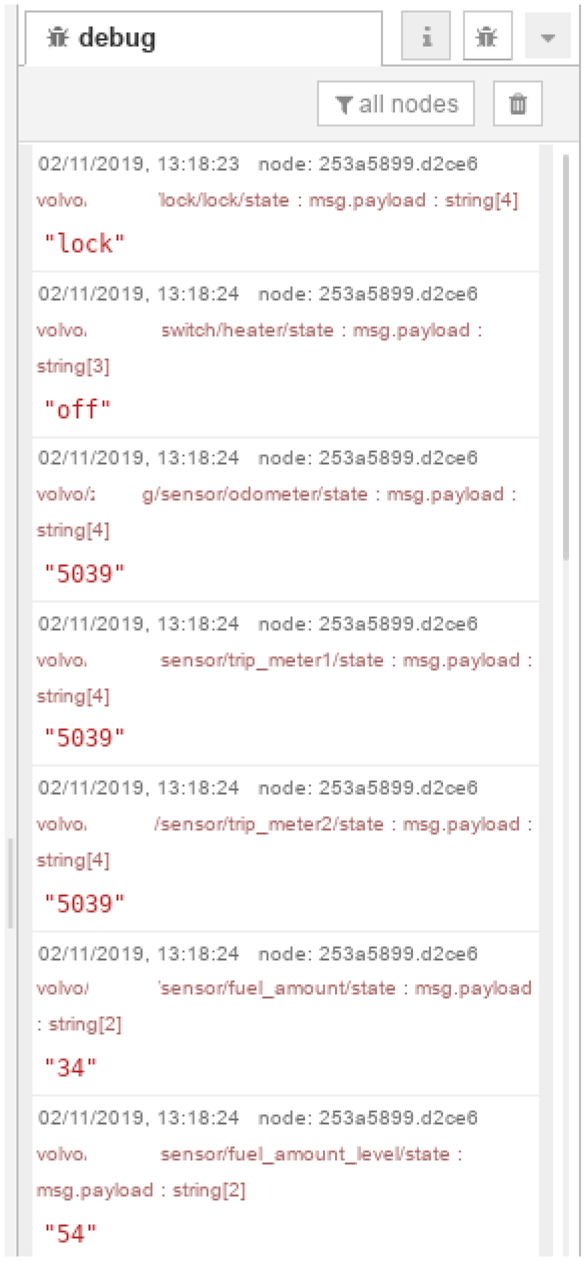
Message sent before disconnecting (close message)

Message sent on an unexpected disconnection (will message)

If everything is configured correctly and you press Deploy in the right top corner, you should see that the MQTT input node connects to the MQTT server. If you connect a “debug” node” directly to its output, you will see the result in the right debug pane.

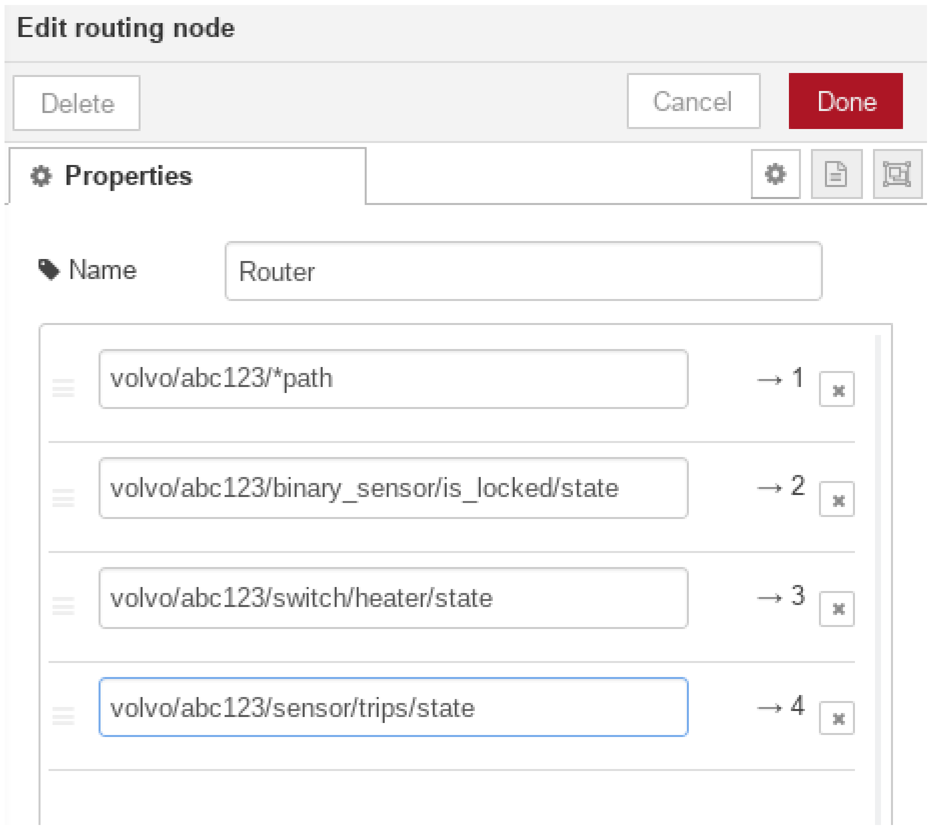


Check carefully that all the topics, you need, are listed in the debug pane and that you created all virtual sensors for it.



```
debug

all nodes

02/11/2019, 13:18:23   node: 253a5899.d2ce6
volvo.          lock/lock/state : msg.payload : string[4]
"lock"

02/11/2019, 13:18:24   node: 253a5899.d2ce6
volvo.          switch/heater/state : msg.payload :
string[3]
"off"

02/11/2019, 13:18:24   node: 253a5899.d2ce6
volvo/:      g/sensor/odometer/state : msg.payload :
string[4]
"5039"

02/11/2019, 13:18:24   node: 253a5899.d2ce6
volvo.          sensor/trip_meter1/state : msg.payload :
string[4]
"5039"

02/11/2019, 13:18:24   node: 253a5899.d2ce6
volvo.          /sensor/trip_meter2/state : msg.payload :
string[4]
"5039"

02/11/2019, 13:18:24   node: 253a5899.d2ce6
volvo/          'sensor/fuel_amount/state : msg.payload
: string[2]
"34"

02/11/2019, 13:18:24   node: 253a5899.d2ce6
volvo.          sensor/fuel_amount_level/state :
msg.payload : string[2]
"54"
```

The output of de MQTT input node is connected to the “Router Node”. Any (green) debug node is optional and can be removed safely.
The task of the “Router node” is to split different topics to different output, because some topics require extra and different “treatment”.



Name: Router

volvo/abc123/*path	→ 1
volvo/abc123/binary_sensor/is_locked/state	→ 2
volvo/abc123/switch/heater/state	→ 3
volvo/abc123/sensor/trips/state	→ 4

The first path everything, except the three below, to output 1
So insert: volvo/abc123/*path (mind the asterisk).

As we will insert later a push button to lock and unlock the car we will route the lock information to output 2. So insert: volvo/abc123/binary_sensor/is_locked/state.

The same applies for the heater (if applicable) and we will route this to output 3.
So insert: volvo/abc123/switch/heater/state.

As we want to modify our date and time, before we send it further, we will route the latest trip information to output 4. So insert: volvo/abc123/sensor/trips/state.

If we have more functions available, e.g. remote start engine, it might be necessary to create additional outputs and insert their corresponding topics.

The first output of the “Router” node is connected to 2 Function nodes. In principal it can be 1 function node.
In order to avoid that the Domoticz log is quickly filled with the same message for every binary sensor (Alarm or Text) only a message will be send, if the message is different from the previous message. This saves a lot of log entries in Domoticz.
In principal this can also be done with the analogue sensor, but a few sensors, such as “Fuel Consumption” and “Average Speed” will not change much. Especially if the car makes more kilometers or miles, the impact on the fuel consumption per kilometer or average speed becomes less and less.

The contents of the first function node, called “Convert Binary Sensors to Domoticz” is as follows:

```
// Function node for Binary Sensors
// Declare variables
var number_plate = "abc123";

// Binary Sensors which will give alert

if (msg.topic == "volvo/" + number_plate + "/binary_sensor/doors_hood_open/state"){
if (msg.payload == "close") {
msg.payload = {"command":"udevice","idx":204,"nvalue":1,"svalue":"Closed"};
} else {
msg.payload = {"command":"udevice","idx":204,"nvalue":4,"svalue":"Open"};
}
return msg;
}

if (msg.topic == "volvo/" + number_plate + "/binary_sensor/doors_front_left_door_open/state"){
if (msg.payload == "close") {
msg.payload = {"command":"udevice","idx":200,"nvalue":1,"svalue":"Closed"};
} else {
msg.payload = {"command":"udevice","idx":200,"nvalue":4,"svalue":"Open"};
}
return msg;
}

if (msg.topic == "volvo/" + number_plate + "/binary_sensor/doors_front_right_door_open/state"){
if (msg.payload == "close") {
msg.payload = {"command":"udevice","idx":203,"nvalue":1,"svalue":"Closed"};
} else {
```

```
msg.payload = {"command":"udevice","idx":203,"nvalue":4,"svalue":"Open"};
}
return msg;
}

if (msg.topic == "volvo/" + number_plate + "/binary_sensor/doors_rear_left_door_open/state"){
if (msg.payload == "close") {
msg.payload = {"command":"udevice","idx":201,"nvalue":1,"svalue":"Closed"};
} else {
msg.payload = {"command":"udevice","idx":201,"nvalue":4,"svalue":"Open"};
}
return msg;
}

if (msg.topic == "volvo/" + number_plate + "/binary_sensor/doors_rear_right_door_open/state"){
if (msg.payload == "close") {
msg.payload = {"command":"udevice","idx":202,"nvalue":1,"svalue":"Closed"};
} else {
msg.payload = {"command":"udevice","idx":202,"nvalue":4,"svalue":"Open"};
}
return msg;
}

if (msg.topic == "volvo/" + number_plate + "/binary_sensor/doors_tailgate_open/state"){
if (msg.payload == "close") {
msg.payload = {"command":"udevice","idx":205,"nvalue":1,"svalue":"Closed"};
} else {
msg.payload = {"command":"udevice","idx":205,"nvalue":4,"svalue":"Open"};
}
return msg;
}
```

```
if (msg.topic == "volvo/" + number_plate + "/binary_sensor/windows_front_left_window_open/state"){
if (msg.payload == "close") {
msg.payload = {"command":"udevice","idx":196,"nvalue":1,"svalue":"Closed"};
} else {
msg.payload = {"command":"udevice","idx":196,"nvalue":4,"svalue":"Open"};
}
return msg;
}

if (msg.topic == "volvo/" + number_plate + "/binary_sensor/windows_front_right_window_open/state"){
if (msg.payload == "close") {
msg.payload = {"command":"udevice","idx":199,"nvalue":1,"svalue":"Closed"};
} else {
msg.payload = {"command":"udevice","idx":199,"nvalue":4,"svalue":"Open"};
}
return msg;
}

if (msg.topic == "volvo/" + number_plate + "/binary_sensor/windows_rear_left_window_open/state"){
if (msg.payload == "close") {
msg.payload = {"command":"udevice","idx":197,"nvalue":1,"svalue":"Closed"};
} else {
msg.payload = {"command":"udevice","idx":197,"nvalue":4,"svalue":"Open"};
}
return msg;
}

if (msg.topic == "volvo/" + number_plate + "/binary_sensor/windows_rear_right_window_open/state"){
if (msg.payload == "close") {
msg.payload = {"command":"udevice","idx":198,"nvalue":1,"svalue":"Closed"};Convert Binary Sensors
to Domoticz
```

```
} else {

msg.payload = {"command":"udevice","idx":198,"nvalue":4,"svalue":"Open"};

}

return msg;

}


// Binary Sensors which will give text


if (msg.topic == "volvo/" + number_plate +
"/binary_sensor/tyre_pressure_front_left_tyre_pressure/state"){

if (msg.payload == "safe") {

msg.payload = {"command":"udevice","idx":206,"nvalue":0,"svalue":"Safe"};

} else {

msg.payload = {"command":"udevice","idx":206,"nvalue":0,"svalue":"Unsafe"};

}

return msg;

}


if (msg.topic == "volvo/" + number_plate +
"/binary_sensor/tyre_pressure_front_right_tyre_pressure/state"){

if (msg.payload == "safe") {

msg.payload = {"command":"udevice","idx":207,"nvalue":0,"svalue":"Safe"};

} else {

msg.payload = {"command":"udevice","idx":207,"nvalue":0,"svalue":"Unsafe"};

}

return msg;

}


if (msg.topic == "volvo/" + number_plate +
"/binary_sensor/tyre_pressure_rear_left_tyre_pressure/state"){

if (msg.payload == "safe") {

msg.payload = {"command":"udevice","idx":208,"nvalue":0,"svalue":"Safe"};
```

```
} else {
msg.payload = {"command":"udevice","idx":208,"nvalue":0,"svalue":"Unsafe"};
}
return msg;
}


if (msg.topic == "volvo/" + number_plate +
"/binary_sensor/tyre_pressure_rear_right_tyre_pressure/state"){
if (msg.payload == "safe") {
msg.payload = {"command":"udevice","idx":209,"nvalue":0,"svalue":"Safe"};
} else {
msg.payload = {"command":"udevice","idx":209,"nvalue":0,"svalue":"Unsafe"};
}
return msg;
}


if (msg.topic == "volvo/" + number_plate + "/binary_sensor/washer_fluid_level/state"){
if (msg.payload == "safe") {
msg.payload = {"command":"udevice","idx":218,"nvalue":0,"svalue":"Safe"};
} else {
msg.payload = {"command":"udevice","idx":218,"nvalue":0,"svalue":"Unsafe"};
}
return msg;
}


if (msg.topic == "volvo/" + number_plate + "/binary_sensor/brake_fluid/state"){
if (msg.payload == "safe") {
msg.payload = {"command":"udevice","idx":217,"nvalue":0,"svalue":"Safe"};
} else {
msg.payload = {"command":"udevice","idx":217,"nvalue":0,"svalue":"Unsafe"};
}
return msg;
```

```
}

if (msg.topic == "volvo/" + number_plate + "/binary_sensor/service_warning_status/state"){
if (msg.payload == "safe") {
msg.payload = {"command":"udevice","idx":224,"nvalue":0,"svalue":"Safe"};
} else {
msg.payload = {"command":"udevice","idx":224,"nvalue":0,"svalue":"Unsafe"};
}
return msg;
}

if (msg.topic == "volvo/" + number_plate + "/binary_sensor/bulb_failures/state"){
if (msg.payload == "safe") {
msg.payload = {"command":"udevice","idx":226,"nvalue":0,"svalue":"Safe"};
} else {
msg.payload = {"command":"udevice","idx":226,"nvalue":0,"svalue":"Unsafe"};
}
return msg;
}

// Binary Sensors not in use

//if (msg.topic == "volvo/" + number_plate + "/binary_sensor/any_door_open/state"){
//if (msg.payload == "close") {
//msg.payload = {"command":"udevice","idx":xxx,"nvalue":1,"svalue":"Closed"};
//} else {
//msg.payload = {"command":"udevice","idx":xxx,"nvalue":4,"svalue":"Open"};
//}
//return msg;
//}
```

```
//if (msg.topic == "volvo/" + number_plate + "/binary_sensor/any_window_open/state"){
//if (msg.payload == "close") {
//msg.payload = {"command":"udevice","idx":yyy,"nvalue":1,"svalue":"Closed"};
//} else {
//msg.payload = {"command":"udevice","idx":yyy,"nvalue":4,"svalue":"Open"};
//}
//return msg;
//}

if (msg.topic == "volvo/" + number_plate + "/binary_sensor/is_engine_running/state"){
if (msg.payload == "off") {
msg.payload = {"command":"udevice","idx":230,"nvalue":0,"svalue":"Off"};
} else {
msg.payload = {"command":"udevice","idx":230,"nvalue":0,"svalue":"On"};
}
return msg;
}

if (msg.topic == "volvo/" + number_plate + "/lock/lock/state"){
if (msg.payload == "lock") {
msg.payload = {"command":"udevice","idx":228,"nvalue":0,"svalue":"Locked"};
} else {
msg.payload = {"command":"udevice","idx":228,"nvalue":0,"svalue":"Unlocked"};
}
return msg;
}

if (msg.topic == "volvo/" + number_plate + "/switch/heater/state"){
if (msg.payload == "off") {
msg.payload = {"command":"udevice","idx":229,"nvalue":0,"svalue":"Off"};
} else {
```

```
msg.payload = {"command":"udevice","idx":229,"nvalue":0,"svalue":"On"};

}

return msg;

}


if (msg.topic == "volvo/" + number_plate + "/sensor/trips/state"){

msg.payload = {"command":"udevice","idx":227,"nvalue":0,"svalue":(msg.payload)};

return msg;

}
```

The user has to replace the value of idx with its own idx numbers.
For the Binary Sensors which will give alert, the “nvalue” represents the colour.
(0 = grey, 1 = green, 2 = yellow, 3 = orange, 4 = red) The “svalue” represents the text to display. You can translate it to your own language, if desired.

For the Binary Sensors which will give text, the “nvalue” is always 0. The “svalue” represents the text to display. You can translate it to your own language, if desired.

Two binary sensors are not in use (any_door_open and any_window_open). Feel free to use them, by removing //. They will behave as an “Alert” sensor.

The contents of the second function node, called “Convert Analog Sensors to Domoticz” is as follows:

```
// Function node for Analog Sensors

// Declare variables

var number_plate = "abc123";


// Sensors Analogue values


if (msg.topic == "volvo/" + number_plate + "/sensor/odometer/state"){

msg.payload = {"command":"udevice","idx":222,"nvalue":0,"svalue":(msg.payload)};

return msg;

}


if (msg.topic == "volvo/" + number_plate + "/sensor/fuel_amount/state"){
```

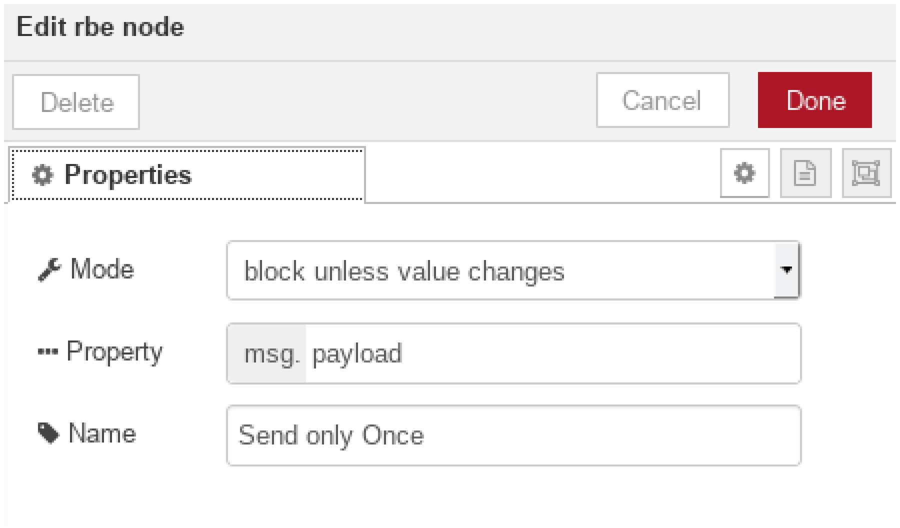
```
msg.payload = {"command":"udevice","idx":212,"nvalue":0,"svalue":(msg.payload)};
return msg;
}

if (msg.topic == "volvo/" + number_plate + "/sensor/fuel_amount_level/state"){
msg.payload = {"command":"udevice","idx":213,"nvalue":0,"svalue":(msg.payload)};
return msg;
}

if (msg.topic == "volvo/" + number_plate + "/sensor/average_fuel_consumption/state"){
msg.payload = {"command":"udevice","idx":214,"nvalue":0,"svalue":(msg.payload)};
return msg;
}

if (msg.topic == "volvo/" + number_plate + "/sensor/distance_to_empty/state"){
msg.payload = {"command":"udevice","idx":216,"nvalue":0,"svalue":(msg.payload)};
return msg;
}

if (msg.topic == "volvo/" + number_plate + "/sensor/average_speed/state"){
msg.payload = {"command":"udevice","idx":215,"nvalue":0,"svalue":(msg.payload)};
return msg;
}

// Sensors not in use

//if (msg.topic == "volvo/" + number_plate + "/sensor/trip_meter1/state"){
//msg.payload = {"command":"udevice","idx":aaa,"nvalue":0,"svalue":(msg.payload)};
//return msg;
//}
```

```
//if (msg.topic == "volvo/" + number_plate + "/sensor/trip_meter2/state"){

//msg.payload = {"command":"udevice","idx":bbb,"nvalue":0,"svalue":(msg.payload)};;

//return msg;

//}
```

The user has to replace the value of idx with its own idx numbers.

Two analogue sensors are not in use (trip_meter1 and trip_meter2). Feel free to usethem, by removing //.

The second output of the "Router" node is connected to a RBE (Report by Exception), called "Send only Once". This node blocks identical payloads and it avoids that a loop will exist, because later on we will add a push button for "Lock/Unlock".



Mode: block unless value changes
Property: msg. payload
Name: Send only Once

The configuration is the default configuration and we don't have to change anything.

The output of this node has to be connected to another function node.
The contents is as follows:

```
if (msg.payload == "on") {

msg.payload = {"command":"udevice","idx":210,"nvalue":1,"svalue":"0"};

} else {

msg.payload = {"command":"udevice","idx":210,"nvalue":0,"svalue":"0"};

}

return msg;
```

The third output of the "Router" node is also connected to a RBE (Report by Exception), called "Send only Once", identical to the previous one. This node blocks also identical payloads and it avoids that a loop will exist, because later on we will add a push button for the heater "On/Off".

The output of this node has to be connected to another function node.
The contents is as follows:

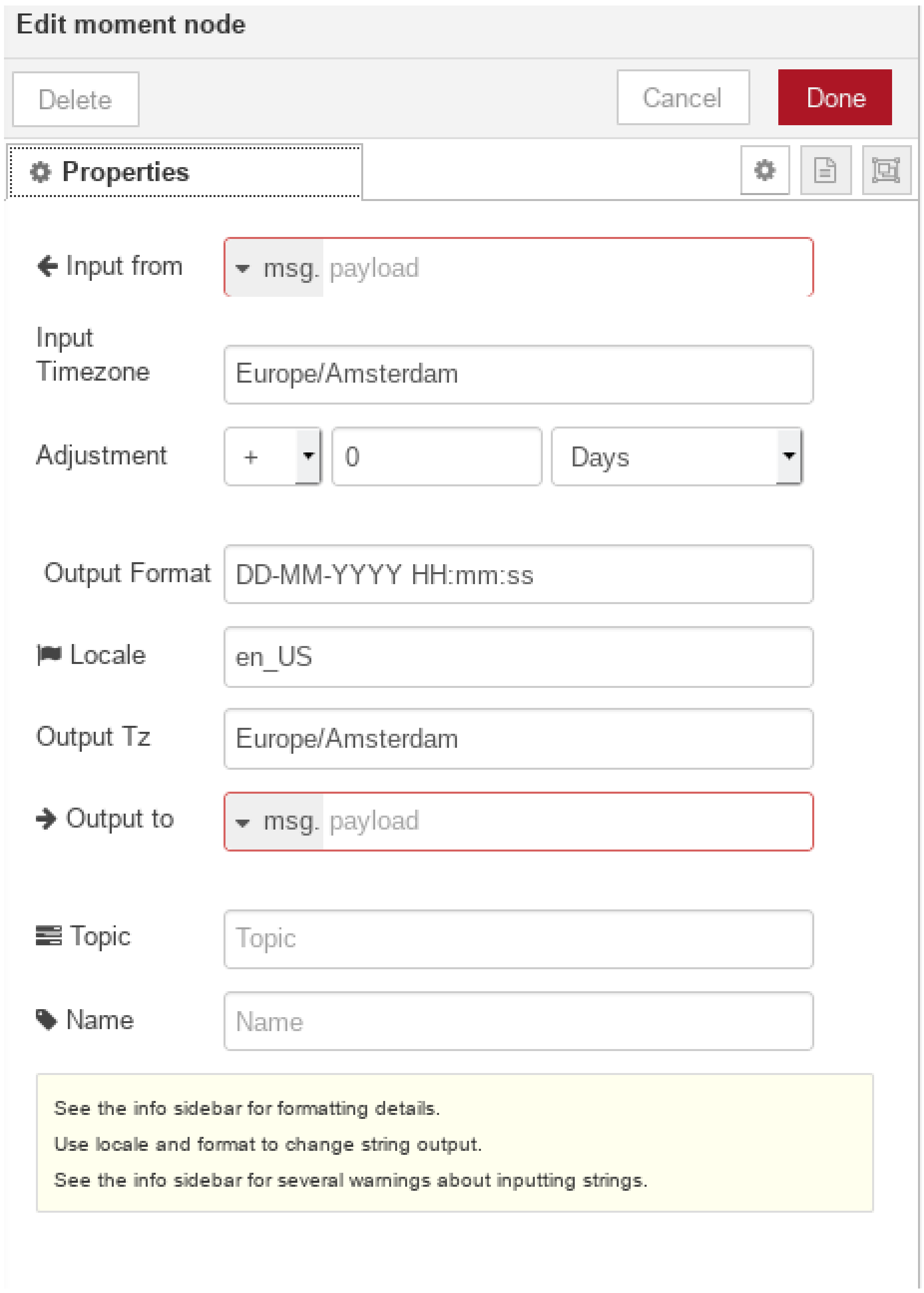
```
if (msg.payload == "off") {

msg.payload = {"command":"udevice","idx":211,"nvalue":0,"svalue":"0"};

} else {

msg.payload = {"command":"udevice","idx":211,"nvalue":1,"svalue":"0"};

}

return msg;
```

Also here the user has to replace the value of idx with its own idx numbers.

The fourth output of the “Router” node is connected to a node, called Date/Time Formatter and is actually a “moment” node.
The output of the “Router” node presents the date and time in the following format:

```
"2019-10-25 11:57:14+02:00"
```

If this is not what you want, you can convert it easily, as follows:



Input from: msg. payload

Input Timezone: Europe/Amsterdam

Adjustment: + 0 Days

Output Format: DD-MM-YYYY HH:mm:ss

Locale: en_US

Output Tz: Europe/Amsterdam

Output to: msg. payload

Topic: Topic

Name: Name

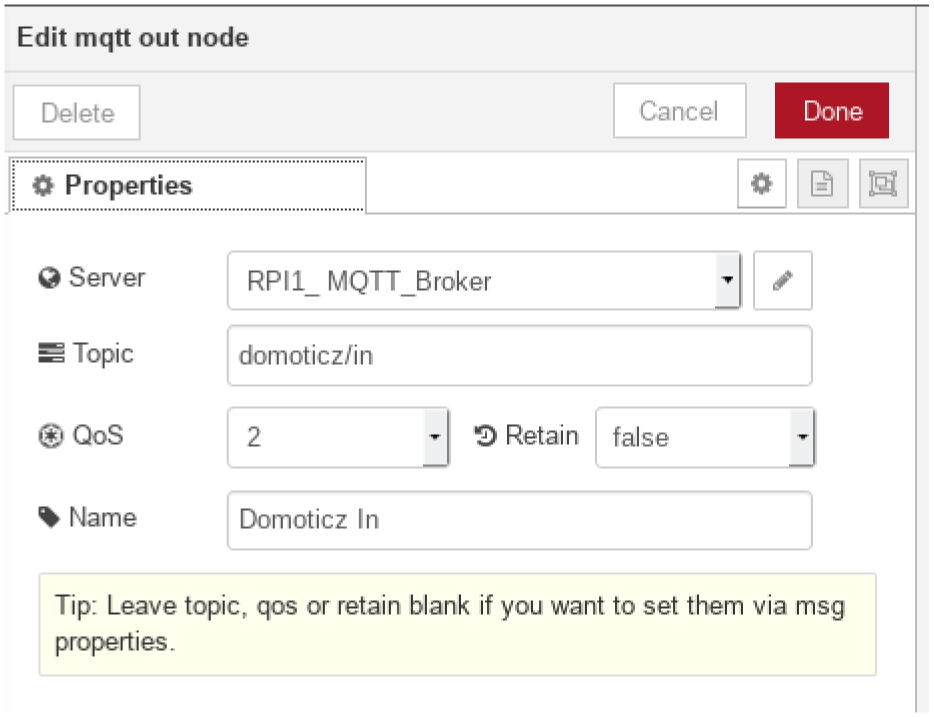
See the info sidebar for formatting details.
Use locale and format to change string output.
See the info sidebar for several warnings about inputting strings.

The output of this “Date/Time Formatter” node has to be connected with the input of the “Convert Binary Sensors to Domoticz” node.

The output of the “Convert Binary Sensors to Domoticz” is connected to another RBE (Report by Exception), called “Block Unchanged”, in order to prevent, that the Domoticz log is filled with identical messages quickly.

The output of this RBE node is, together with the outputs of the three other function nodes connected to a MQTT output node, called “Domoticz In”

The configuration of this node is almost identical to the MQTT input node, except for the topic.


Edit mqtt out node

Server: RPI1_ MQTT_Broker

Topic: domoticz/in

QoS: 2

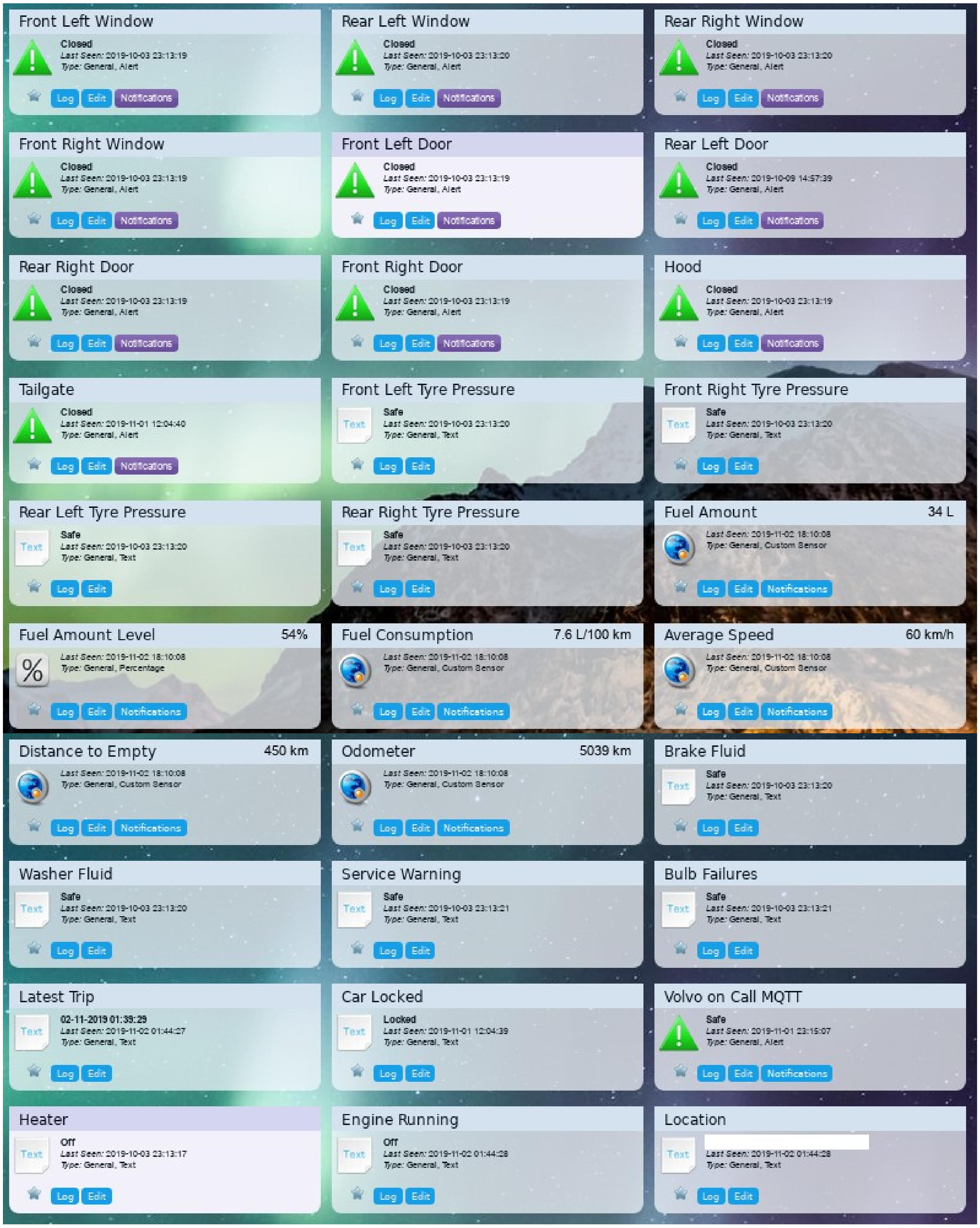
Retain: false

Name: Domoticz In

Tip: Leave topic, qos or retain blank if you want to set them via msg properties.

The Topic has to be: domoticz/in.
All other properties can be left as default.

If everything has been configured successfully all the sensors will appear in Domoticz under the “Utility” tab. Other sensors can be added easily, if you split it into binary and analogue s sensors and copy a code block in either the “Convert Binary Sensors to Domoticz” node or “Convert Analog Sensors to Domoticz node”.



Front Left Window
Closed
Last Seen: 2019-10-03 23:13:19
Type: General, Alert

Rear Left Window
Closed
Last Seen: 2019-10-03 23:13:20
Type: General, Alert

Rear Right Window
Closed
Last Seen: 2019-10-03 23:13:20
Type: General, Alert

Front Right Window
Closed
Last Seen: 2019-10-03 23:13:19
Type: General, Alert

Front Left Door
Closed
Last Seen: 2019-10-03 23:13:19
Type: General, Alert

Rear Left Door
Closed
Last Seen: 2019-10-09 14:57:39
Type: General, Alert

Rear Right Door
Closed
Last Seen: 2019-10-03 23:13:19
Type: General, Alert

Front Right Door
Closed
Last Seen: 2019-10-03 23:13:19
Type: General, Alert

Hood
Closed
Last Seen: 2019-10-03 23:13:19
Type: General, Alert

Tailgate
Closed
Last Seen: 2019-11-01 12:04:40
Type: General, Alert

Front Left Tyre Pressure
Safe
Last Seen: 2019-10-03 23:13:20
Type: General, Text

Front Right Tyre Pressure
Safe
Last Seen: 2019-10-03 23:13:20
Type: General, Text

Rear Left Tyre Pressure
Safe
Last Seen: 2019-10-03 23:13:20
Type: General, Text

Rear Right Tyre Pressure
Safe
Last Seen: 2019-10-03 23:13:20
Type: General, Text

Fuel Amount 34 L
Last Seen: 2019-11-02 18:10:08
Type: General, Custom Sensor

Fuel Amount Level 54%
Last Seen: 2019-11-02 18:10:08
Type: General, Percentage

Fuel Consumption 7.6 L/100 km
Last Seen: 2019-11-02 18:10:08
Type: General, Custom Sensor

Average Speed 60 km/h
Last Seen: 2019-11-02 18:10:08
Type: General, Custom Sensor

Distance to Empty 450 km
Last Seen: 2019-11-02 18:10:08
Type: General, Custom Sensor

Odometer 5039 km
Last Seen: 2019-11-02 18:10:08
Type: General, Custom Sensor

Brake Fluid
Safe
Last Seen: 2019-10-03 23:13:20
Type: General, Text

Washer Fluid
Safe
Last Seen: 2019-10-03 23:13:20
Type: General, Text

Service Warning
Safe
Last Seen: 2019-10-03 23:13:21
Type: General, Text

Bulb Failures
Safe
Last Seen: 2019-10-03 23:13:21
Type: General, Text

Latest Trip
02-11-2019 01:39:29
Last Seen: 2019-11-02 01:44:27
Type: General, Text

Car Locked
Locked
Last Seen: 2019-11-01 12:04:39
Type: General, Text

Volvo on Call MQTT
Safe
Last Seen: 2019-11-01 23:15:07
Type: General, Alert

Heater
Off
Last Seen: 2019-10-03 23:13:17
Type: General, Text

Engine Running
Off
Last Seen: 2019-11-02 01:44:28
Type: General, Text

Location
[redacted]
Last Seen: 2019-11-02 01:44:28
Type: General, Text

4 Node-Red flow Domoticz to VoC

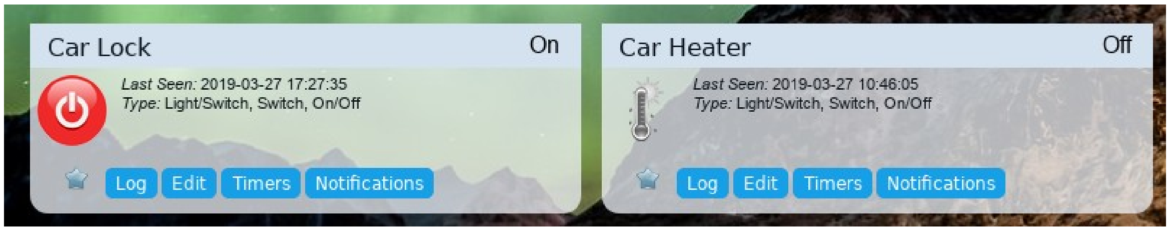
Before we start we should test the functionality from the command line.
So go to the command line on the device, where the VoC script is running and issue the following commands, if your car supports it and check that the commands are carried out:

- $ *voc lock*
- $ *voc unlock*
- $ *voc heater on*
- $ *voc heater off*
- $ *voc engine on*
- $ *voc engine off*

Note: The VoC help file indicate start and stop. If the commands on/off doe not work, try start/stop instead.

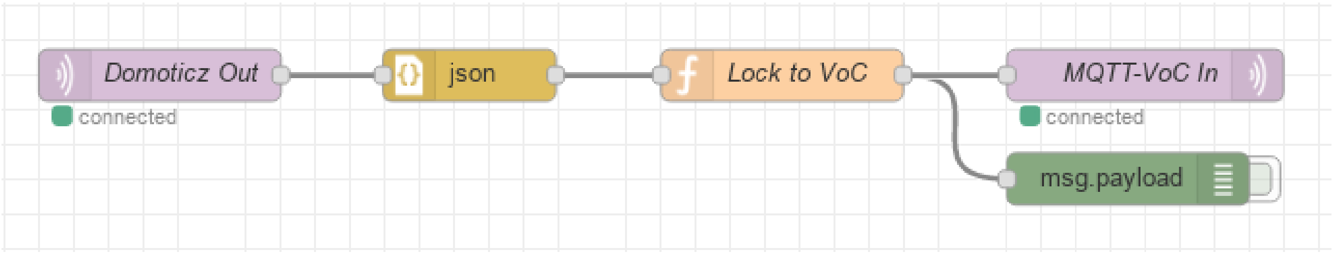
We have to create several virtual switches.
In order to create these switches, if not done already, create in "Hardware" a Dummy (Does nothing, use for virtual switches only).

Each user has to check, which commands are supported for his/her Volvo.

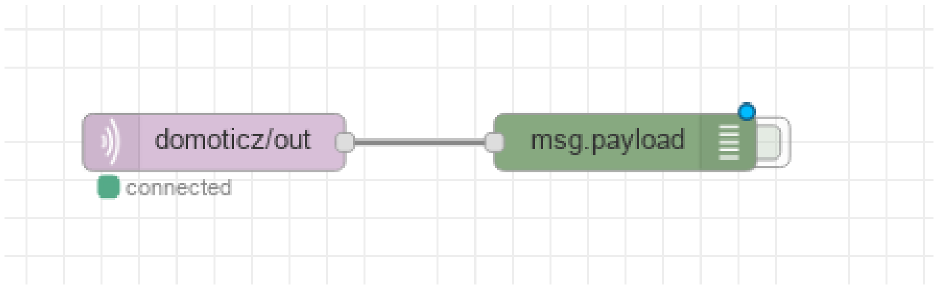

Car Lock
On
Last Seen: 2019-03-27 17:27:35
Type: Light/Switch, Switch, On/Off
Log
Edit
Timers
Notifications
Car Heater
Off
Last Seen: 2019-03-27 10:46:05
Type: Light/Switch, Switch, On/Off
Log
Edit
Timers
Notifications

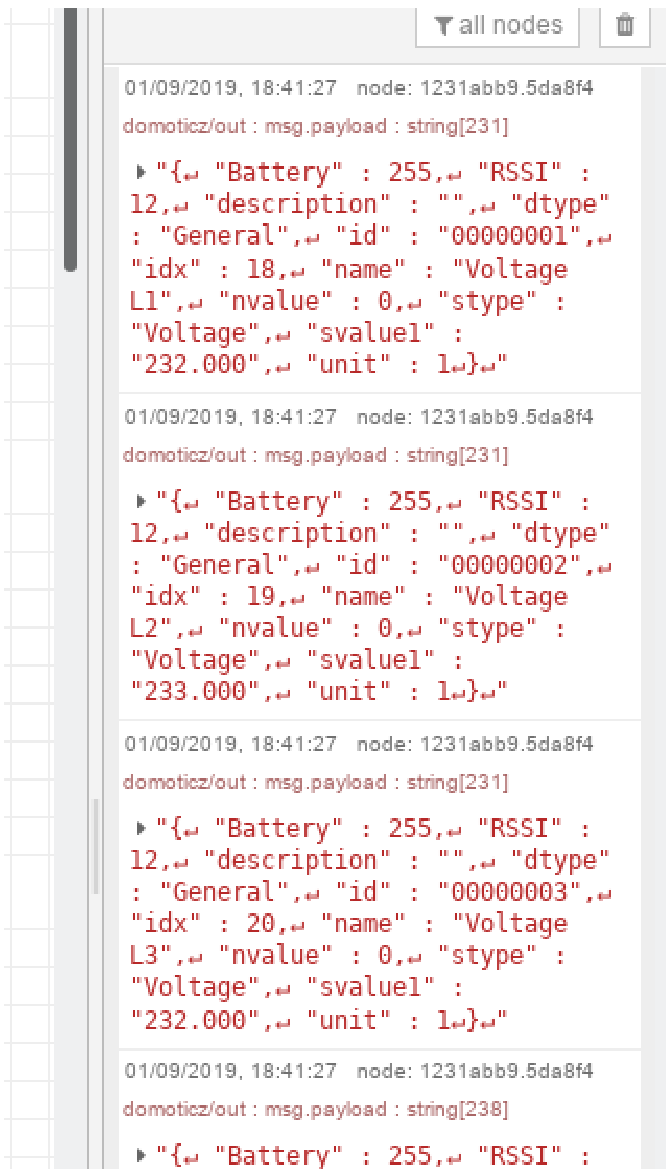
Go to "Setup" and then to "Devices". Make a note of the Idx number of the newly created switches, as we will need these later.

In the picture below you will see the total flow.



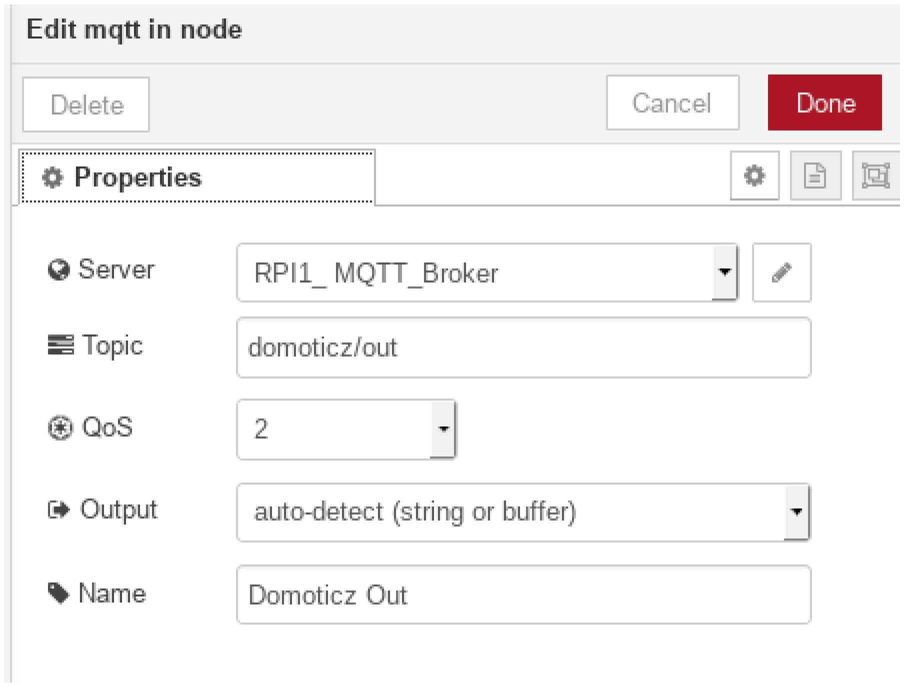
Every event in Domoticz is published to domoticz/out. So if you press on one of the newly created switched a message is published.
If you connect a debug node to the output of "Domoticz Out" you will see a lot of messages published. We have to filter the right one and therefore we need the idx number oft he switch.





```
01/09/2019, 18:41:27   node: 1231abb9.5da8f4
domoticz/out : msg.payload : string[231]
▸"{↵ "Battery" : 255,↵ "RSSI" : 12,↵ "description" : "",↵ "dtype" : "General",↵ "id" : "00000001",↵ "idx" : 18,↵ "name" : "Voltage L1",↵ "nvalue" : 0,↵ "stype" : "Voltage",↵ "svalue1" : "232.000",↵ "unit" : 1↵}↵"

01/09/2019, 18:41:27   node: 1231abb9.5da8f4
domoticz/out : msg.payload : string[231]
▸"{↵ "Battery" : 255,↵ "RSSI" : 12,↵ "description" : "",↵ "dtype" : "General",↵ "id" : "00000002",↵ "idx" : 19,↵ "name" : "Voltage L2",↵ "nvalue" : 0,↵ "stype" : "Voltage",↵ "svalue1" : "233.000",↵ "unit" : 1↵}↵"

01/09/2019, 18:41:27   node: 1231abb9.5da8f4
domoticz/out : msg.payload : string[231]
▸"{↵ "Battery" : 255,↵ "RSSI" : 12,↵ "description" : "",↵ "dtype" : "General",↵ "id" : "00000003",↵ "idx" : 20,↵ "name" : "Voltage L3",↵ "nvalue" : 0,↵ "stype" : "Voltage",↵ "svalue1" : "232.000",↵ "unit" : 1↵}↵"

01/09/2019, 18:41:27   node: 1231abb9.5da8f4
domoticz/out : msg.payload : string[238]
▸"{↵ "Battery" : 255,↵ "RSSI" :
```

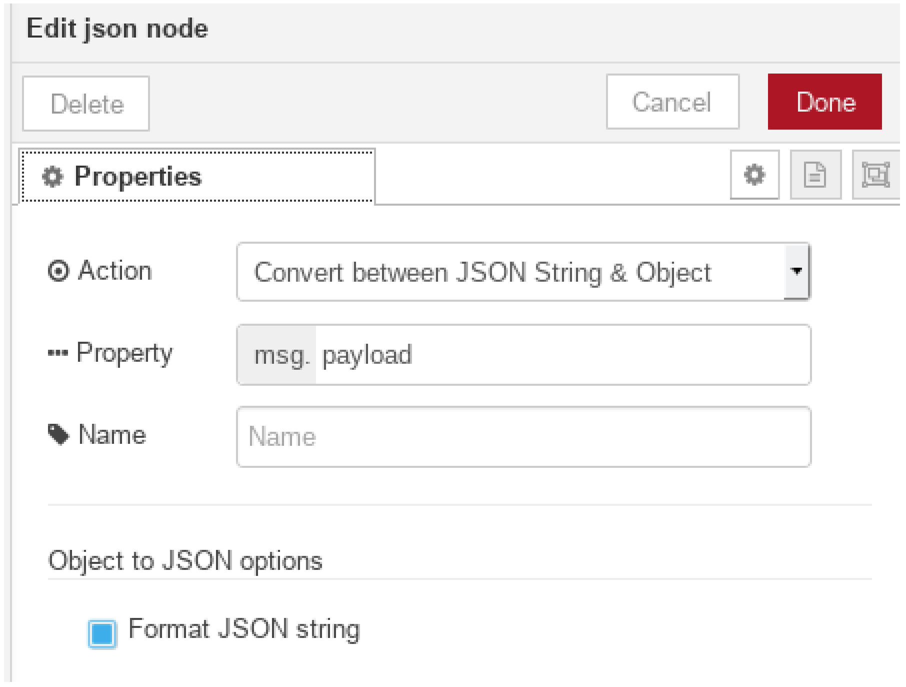
The first node for this flow in a MQTT input node, called “Domoticz Out”
The configuration is identical to the previous used MQTT nodes, except the Topic.



Server: RPI1_ MQTT_Broker
Topic: domoticz/out
QoS: 2
Output: auto-detect (string or buffer)
Name: Domoticz Out

The Topic has to be configured as: domoticz/out

As we can see the output is a javascript string, but we like to see a JSON message.
Therefore the second node is a JSON function node, which converts between a JSON string and an object.
The configuration is as follows:



Action: Convert between JSON String & Object
Property: msg. payload
Name: Name

Object to JSON options

Format JSON string

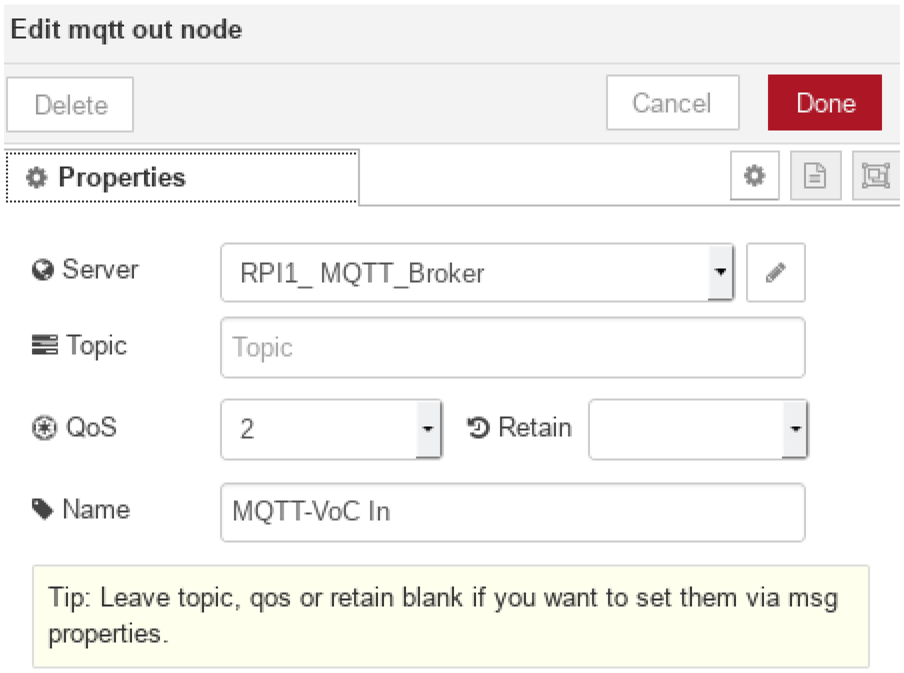
The output of the JSON function node is connected to the input of a funtion node.
This function node takes care for selecting the right idx number of the switch and converts the messages from Domoticz to a message suitable to be handled by the VoC python script.

The contents is as follows:

```
// Declare variables
var number_plate = "abc123";


// Lock
if (msg.payload.idx == "210") {
   msg.topic = "volvo/" + number_plate + "/lock/lock/cmd";
   if (msg.payload.nvalue == 1) {
   msg.payload = "lock";
   }else{
   msg.payload = "unlock";
   }
return msg;
}


//Heater
//Not available anymore
//if (msg.payload.idx == "211") {
//    msg.topic = "volvo/" + number_plate + "/switch/heater/cmd";
//    if (msg.payload.nvalue == 1) {
//    msg.payload = "on";
//    }else{
//    msg.payload = "off";
//    }
//return msg;
//}
```

As the heater is not available any more, this section is commented. If your car is capable to start the heater remotely, simply uncomment this section. If your car is capable to start the engine remotely, simply copy a section. Replace the idx numbers with your numbers.

The output of the function node “Lock to VoC” is connected to the input of a MQTT output node, called MQTT-VoC In.
The configuration is identical as the previous MQTT nodes, except for the Topic.



Server: RPI1_ MQTT_Broker
Topic: Topic
QoS: 2
Retain:
Name: MQTT-VoC In

Tip: Leave topic, qos or retain blank if you want to set them via msg properties.

The Topic should be left empty, as we have set already the topic in the previous function node to volvo/abc123/lock/lock/cmd or volvo/abc123/switch/heater/cmd.
In case you have added a switch for remote engine start/stop you can test it yourself, remembering that the VoC python script has subscribed to the command with /cmd.

It is now time to test the functionality.

5 Temperature and Location

To do