Milesight Sensor

with

Luxriot EVO

Integration & Event Configuration Guide

24 Aug 2026

Table of Contents

1. Overview

This guide describes how to integrate a Milesight environmental sensor with Luxriot EVO using MQTT notifications, create trigger and release events from the sensor JSON payload, map those events to EVO indicator states, and display the resulting states in Luxriot EVO Monitor.

The sensor can generate multiple alarm categories. In the demonstrated configuration, Luxriot EVO can be configured with 18 trigger events and 18 corresponding release events - 36 event definitions in total. Each pair represents the active and cleared state of one sensor condition.

> **Important:** Luxriot EVO uses the ECMAScript regular expression dialect for MQTT event matching. The expressions shown in this guide were validated against the compact JSON payload produced by the Milesight sensor.

Prerequisites

- Milesight sensor with MQTT support and access to its web interface.
- An MQTT broker reachable by both the sensor and the Luxriot EVO server.
- Luxriot EVO configured with access to Events & Actions and Control Panels.
- Network connectivity between the Milesight sensor, MQTT broker, and Luxriot EVO server.

Event model

For each supported sensor category, create two Luxriot EVO MQTT events: one event for the trigger condition and one event for the release condition. The event topic can remain common while the regular expression identifies the specific object and state in the JSON message.

2. Configure MQTT on the Milesight Sensor

Log in to the Milesight sensor web interface and open Platform Integration > MQTT Settings. Enable MQTT, enter the broker connection parameters required for your environment, and save the configuration.

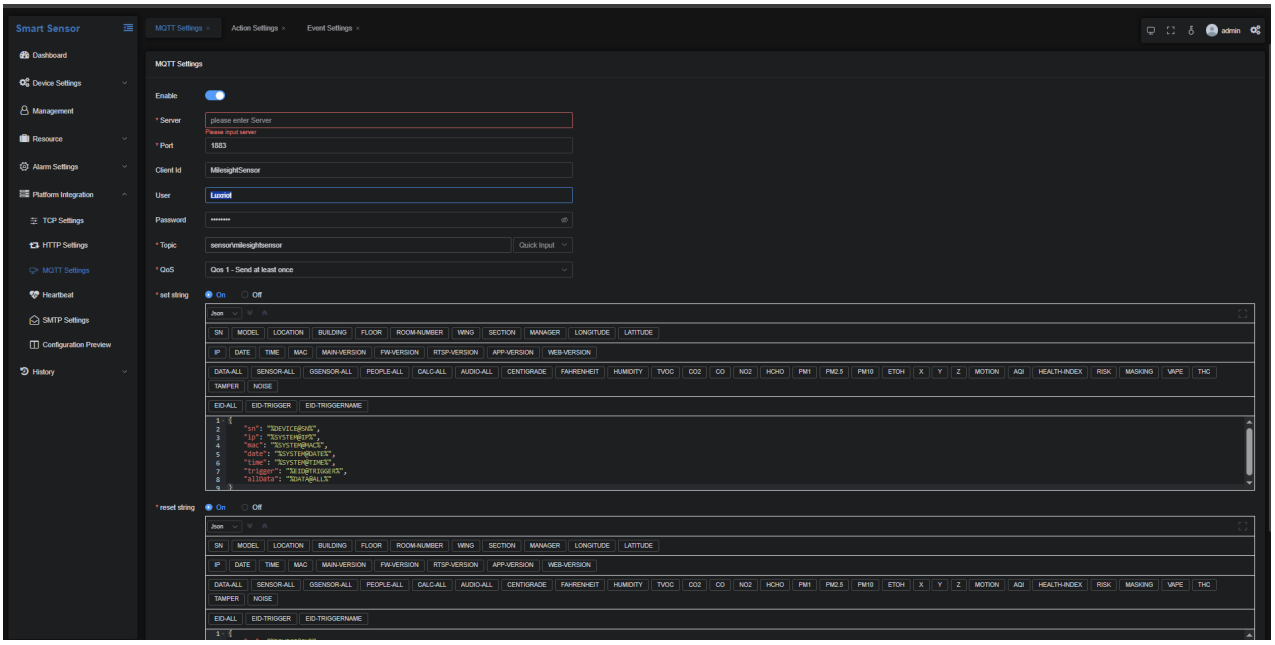

Figure 1. Milesight MQTT settings and event configuration.

> **Tip:** Confirm that the sensor is publishing messages to the expected MQTT topic before creating the EVO events. The Luxriot event definitions must reference the same topic used by the sensor.

3. Create Indicator States in Luxriot EVO

Indicator states provide a visual representation of the current condition in Luxriot EVO. Create one indicator state object for each sensor condition you want to display.

1. In Luxriot EVO Console, go to Events & Actions > Indicator States.
2. Create a new indicator state and give it a meaningful title, for example Milesight Sensor Motion State.
3. Configure the two meaningful states for the sensor condition, such as No Motion and Motion. Unused states can be left as N/A and assigned a neutral color.

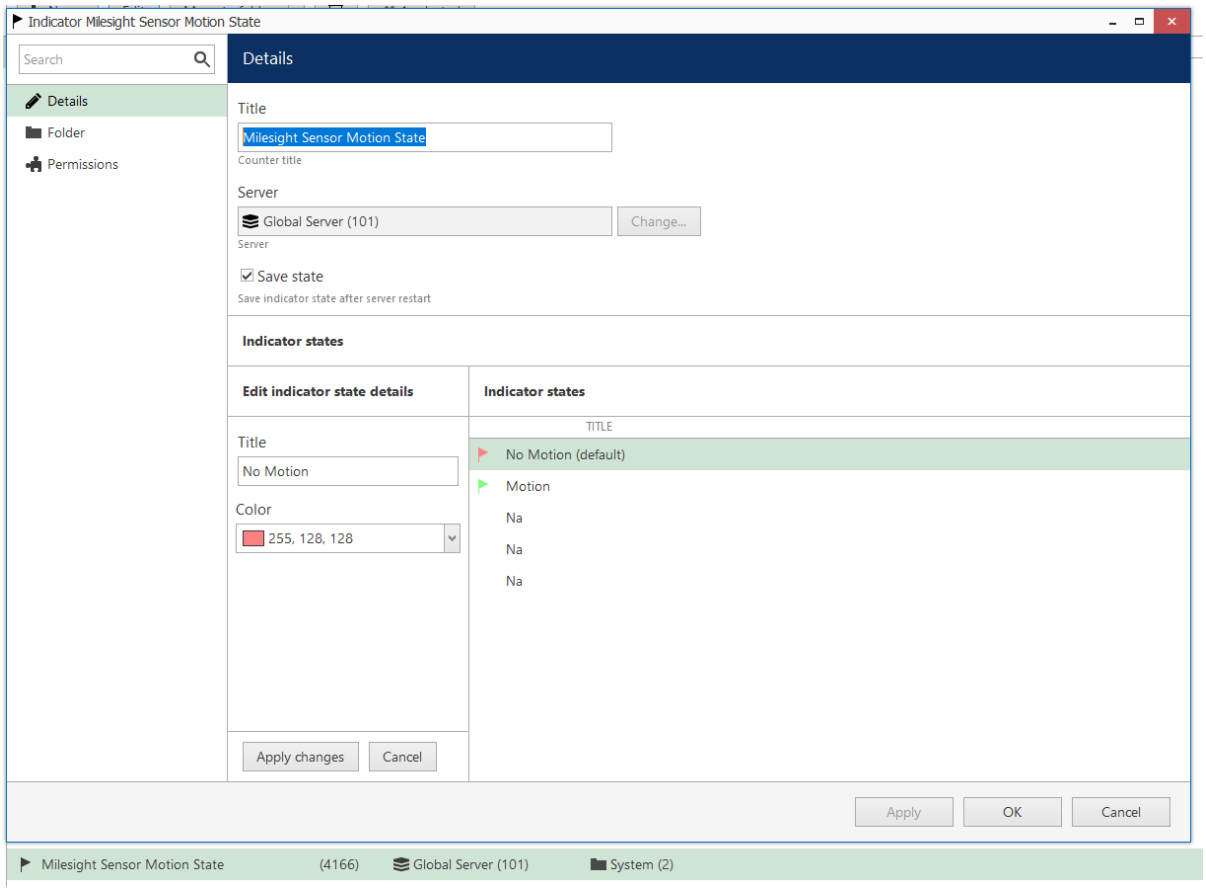

Figure 2. Example indicator state for the Milesight motion condition.

4. Create MQTT Trigger and Release Events

In Events & Actions > Events, create two events for each sensor category. Select the MQTT event type, choose the Milesight MQTT source, enter the sensor topic, enable Regular expression, and define a unique expression for the trigger and release messages.

Motion trigger

Create an event for the active Motion condition. The following ECMAScript expression matches a Milesight payload containing a Motion trigger:

```
.*"trigger":\{"Motion":\{"trigger":true.*
```

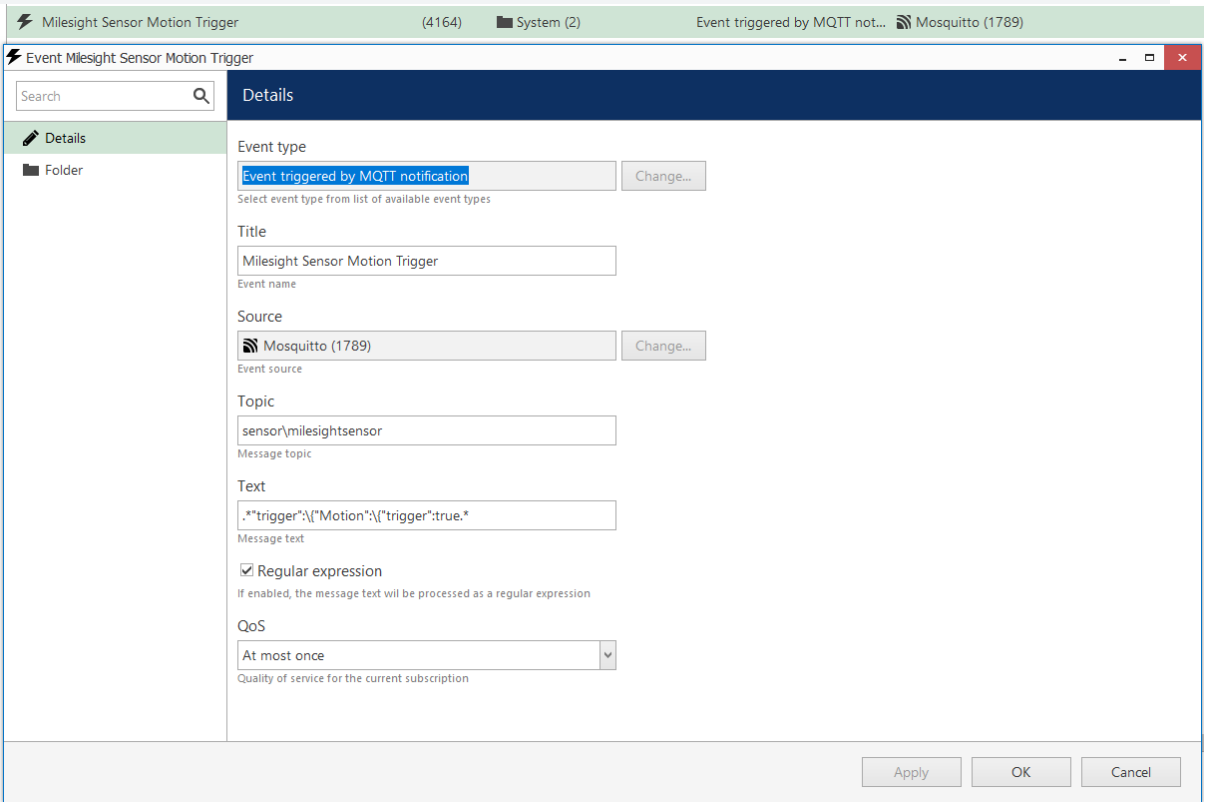

Figure 3. Motion trigger event configured as an MQTT notification.

Motion release

Create a second event for the cleared Motion condition. The following expression matches a Milesight payload containing a Motion release:

```
.*"trigger":\{"Motion":\{"release":true.*
```

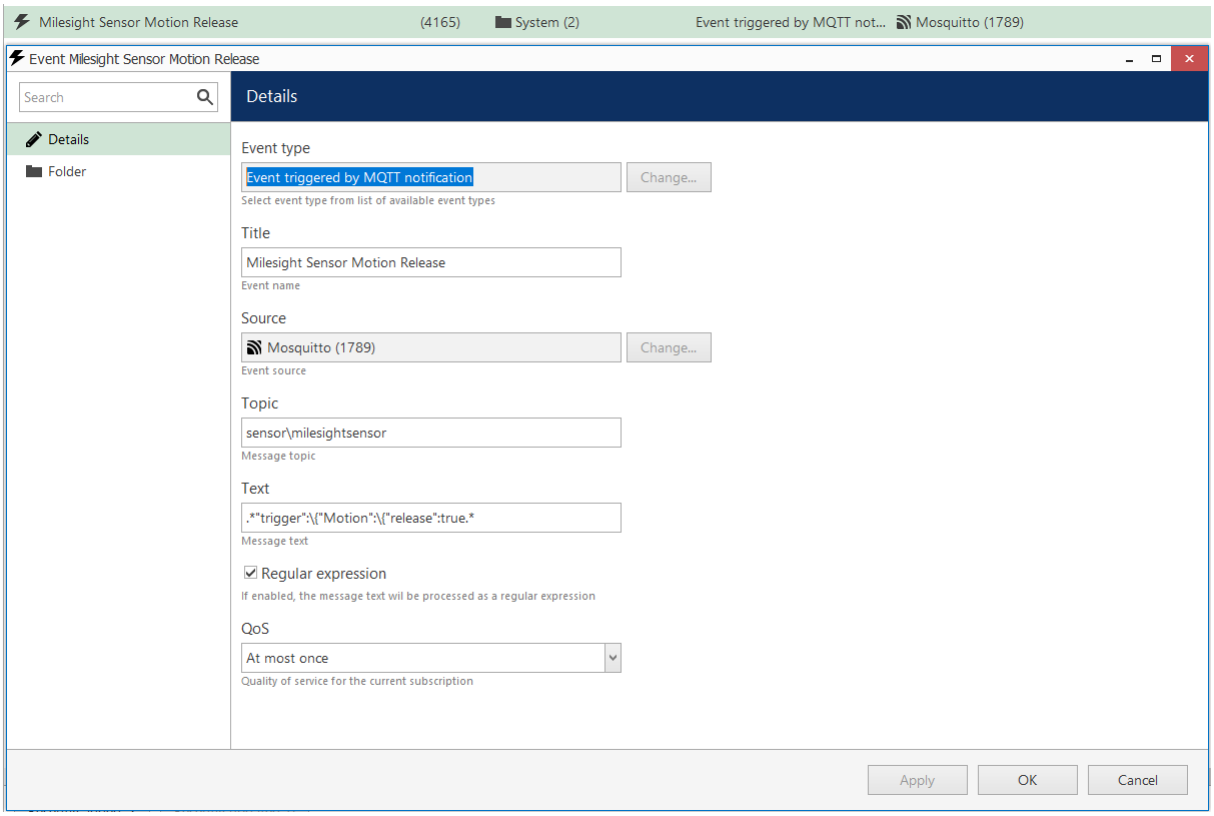

Figure 4. Motion release event configured as an MQTT notification.

Pattern reuse: For other sensor categories, keep the same expression structure and replace only Motion with the exact object name present in the incoming JSON payload.

5. Event Pattern Reference

The table below shows the 36 unique event filters corresponding to the 18 configured sensor categories. Each filter is specific to both the sensor object and its trigger or release state.

Sensor object	Trigger expression	Release expression
Vape	.*"trigger":\{"Vape":\{"trigger":true.*	.*"trigger":\{"Vape":\{"release":true.*
THC	.*"trigger":\{"THC":\{"trigger":true.*	.*"trigger":\{"THC":\{"release":true.*
Masking	.*"trigger":\{"Masking":\{"trigger":true.*	.*"trigger":\{"Masking":\{"release":true.*
Aggression	.*"trigger":\{"Aggression":\{"trigger":true.*	.*"trigger":\{"Aggression":\{"release":true.*
Tamper	.*"trigger":\{"Tamper":\{"trigger":true.*	.*"trigger":\{"Tamper":\{"release":true.*
Motion	.*"trigger":\{"Motion":\{"trigger":true.*	.*"trigger":\{"Motion":\{"release":true.*
Health_Index	.*"trigger":\{"Health_Index":\{"trigger":true.*	.*"trigger":\{"Health_Index":\{"release":true.*
AQI	.*"trigger":\{"AQI":\{"trigger":true.*	.*"trigger":\{"AQI":\{"release":true.*
PM2.5	.*"trigger":\{"PM2.5":\{"trigger":true.*	.*"trigger":\{"PM2.5":\{"release":true.*
PM10	.*"trigger":\{"PM10":\{"trigger":true.*	.*"trigger":\{"PM10":\{"release":true.*

Sensor object	Trigger expression	Release expression
TVOC	.*"trigger":\{"TVOC":\{"trigger":true.*	.*"trigger":\{"TVOC":\{"release":true.*
CO2	.*"trigger":\{"CO2":\{"trigger":true.*	.*"trigger":\{"CO2":\{"release":true.*
Humidity	.*"trigger":\{"Humidity":\{"trigger":true.*	.*"trigger":\{"Humidity":\{"release":true.*
HCHO	.*"trigger":\{"HCHO":\{"trigger":true.*	.*"trigger":\{"HCHO":\{"release":true.*
ETOH	.*"trigger":\{"ETOH":\{"trigger":true.*	.*"trigger":\{"ETOH":\{"release":true.*
Noise	.*"trigger":\{"Noise":\{"trigger":true.*	.*"trigger":\{"Noise":\{"release":true.*
Temp_C	.*"trigger":\{"Temp_C":\{"trigger":true.*	.*"trigger":\{"Temp_C":\{"release":true.*
Temp_F	.*"trigger":\{"Temp_F":\{"trigger":true.*	.*"trigger":\{"Temp_F":\{"release":true.*

Payload naming: The object name in the regular expression must match the JSON exactly, including capitalization and punctuation. If a sensor firmware version publishes a different object name, use the name shown in the received MQTT payload.

6. Build Events & Actions Rules

After the MQTT events and indicator states are created, use the Events & Actions Configurator to connect each event to the appropriate indicator-state action.

4. Open Luxriot EVO Console > Events & Actions > Open Configurator.
5. Locate the trigger and release events in the Events pane on the left.
6. Create one rule for the trigger and one rule for the release. Events can be moved into the Rules pane by double-clicking or by using the arrows between panes.
7. Add the corresponding Change Indicator State action from the Actions pane on the right to the highlighted rule.

The center pane always represents the rules. In the example below, one rule changes the indicator to Motion when the trigger event occurs and another changes it back to No Motion when the release event occurs.

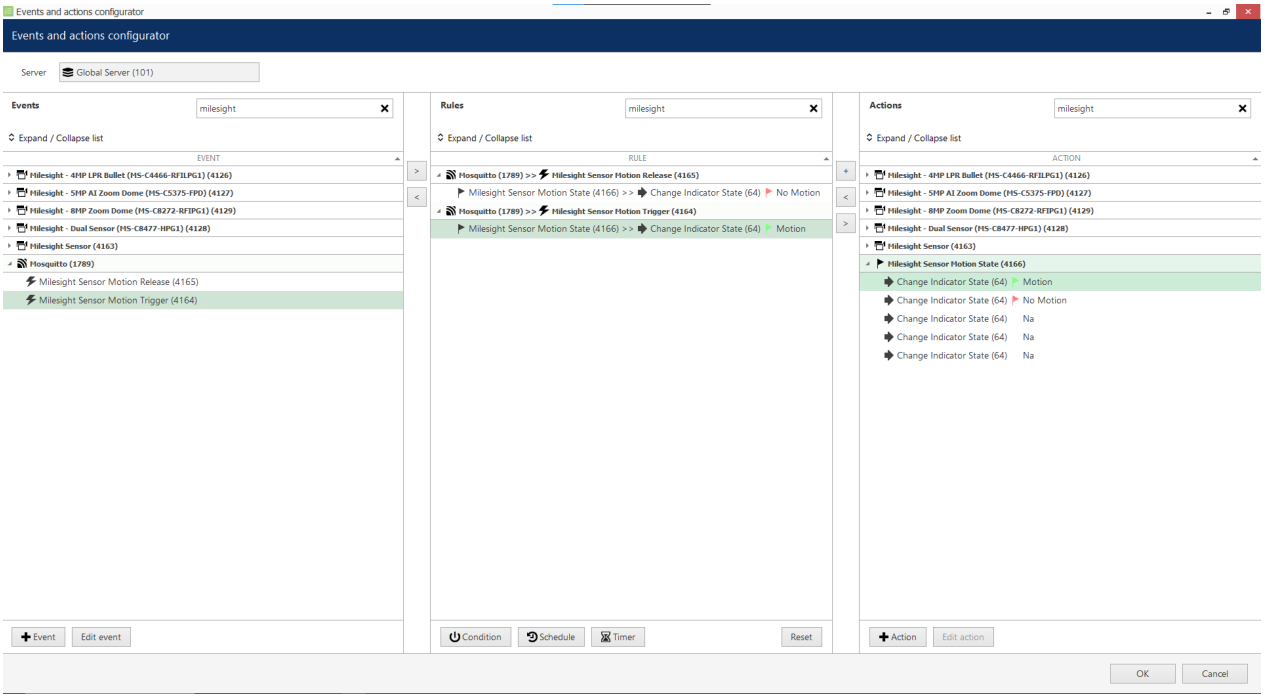

Figure 5. Trigger rule for the Motion condition.

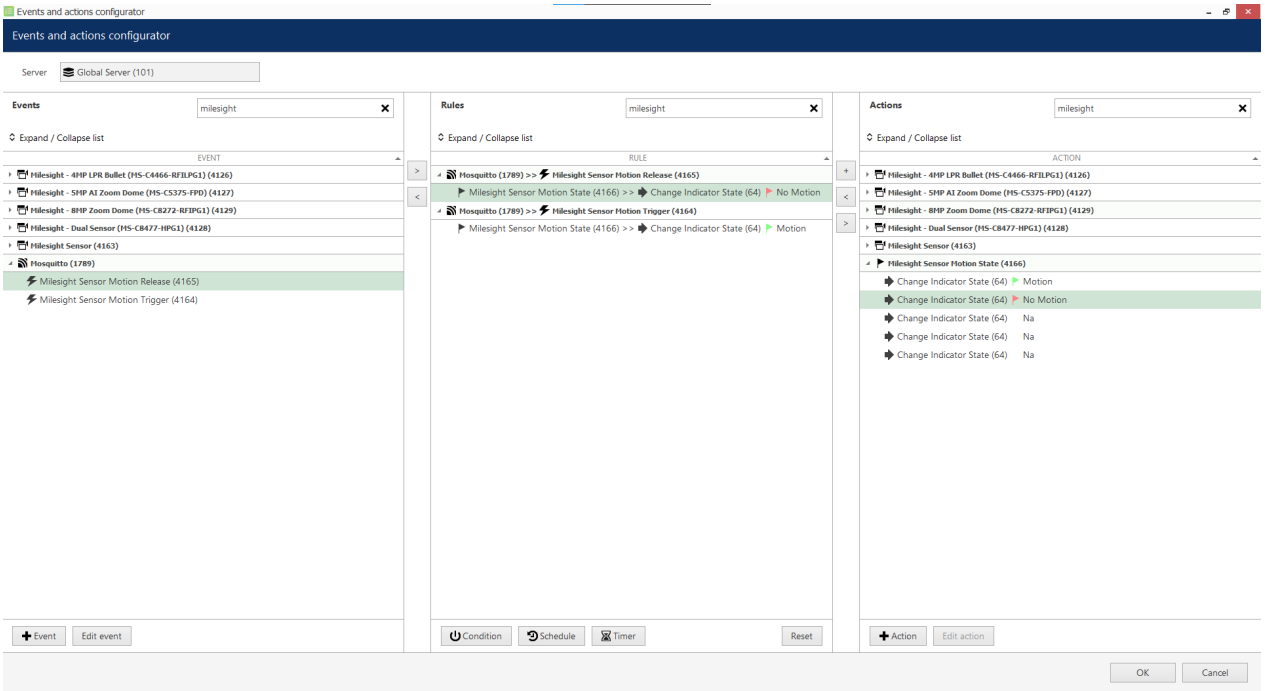

Figure 6. Release rule for the Motion condition.

7. Add the Indicator to a Control Panel

Control Panels can be used to present sensor states directly in Luxriot EVO Monitor.

8. Go to Configuration > Control Panels (beta).
9. Create a new control panel and enable it.

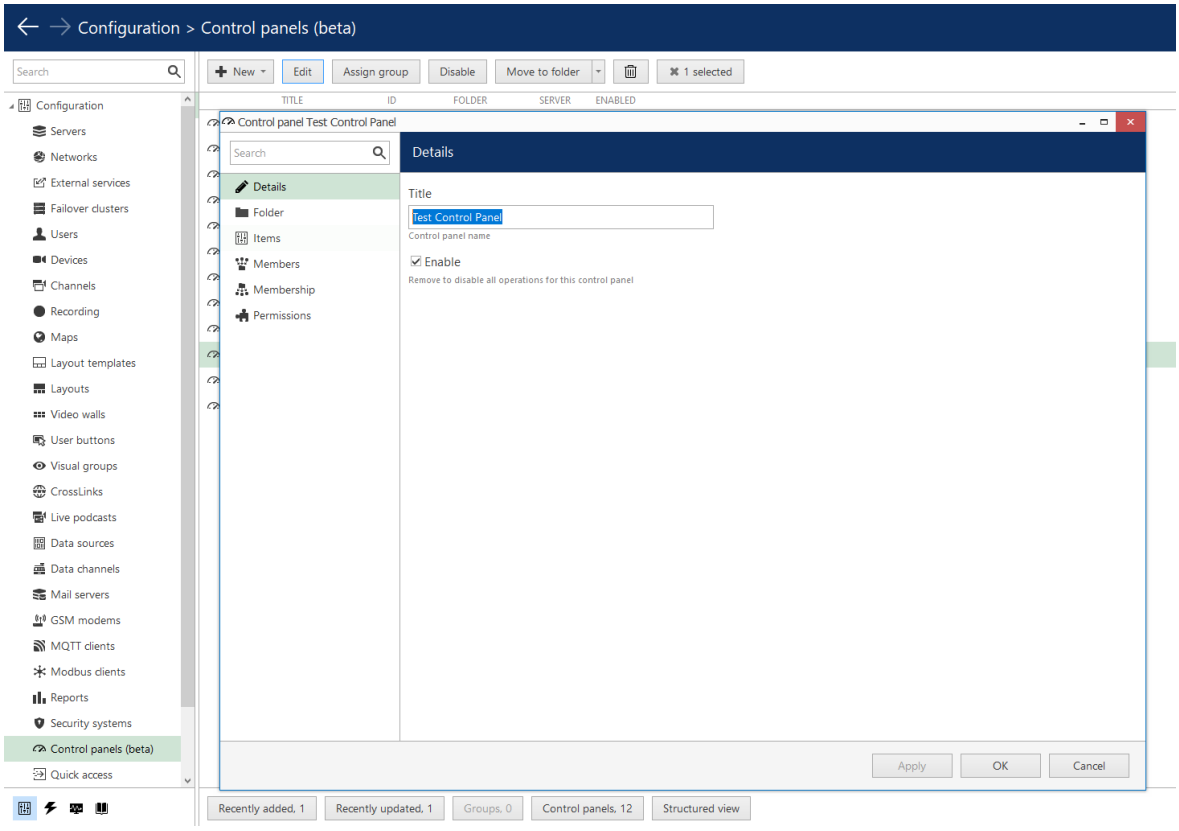

Figure 7. Creating a new control panel.

10. Open the Items section and add the indicator state created earlier.

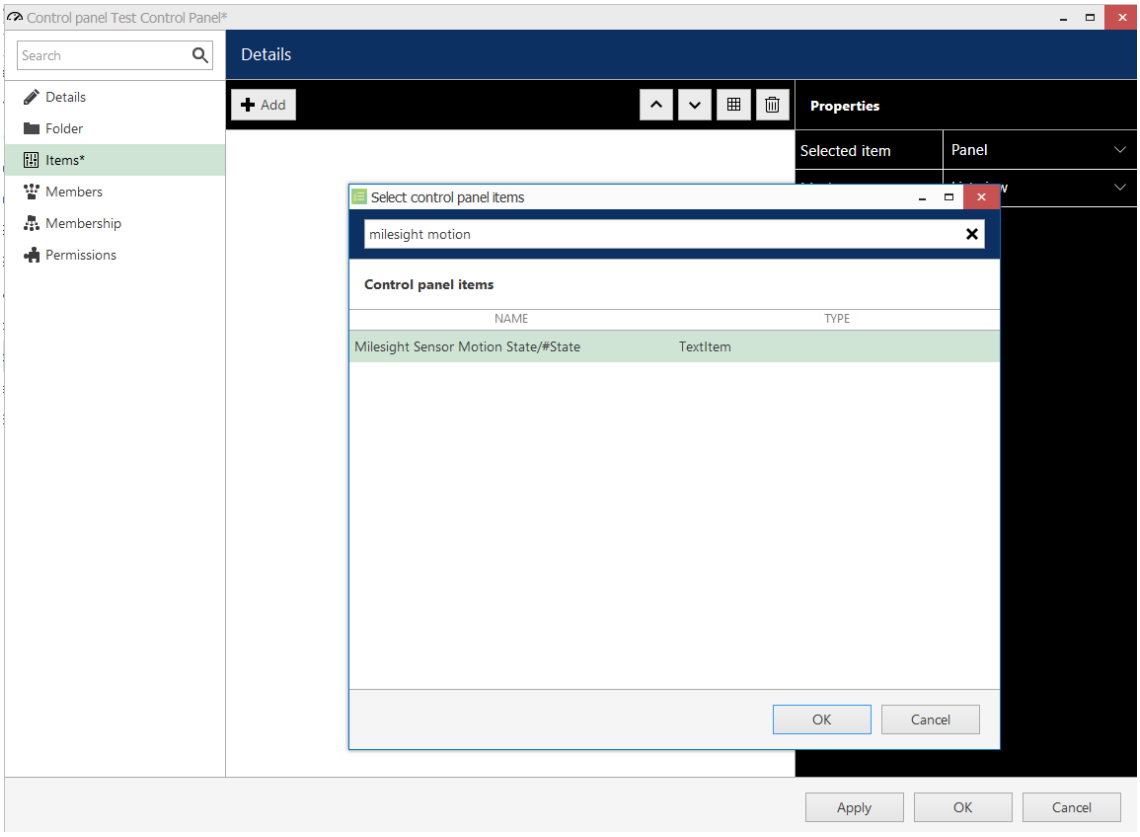

Figure 8. Selecting the Milesight indicator state as a control panel item.

11. If using Grid view, select the target grid cell before clicking Add. If no grid cell is selected, the Add button remains unavailable. In List view, use New and select the indicator state.

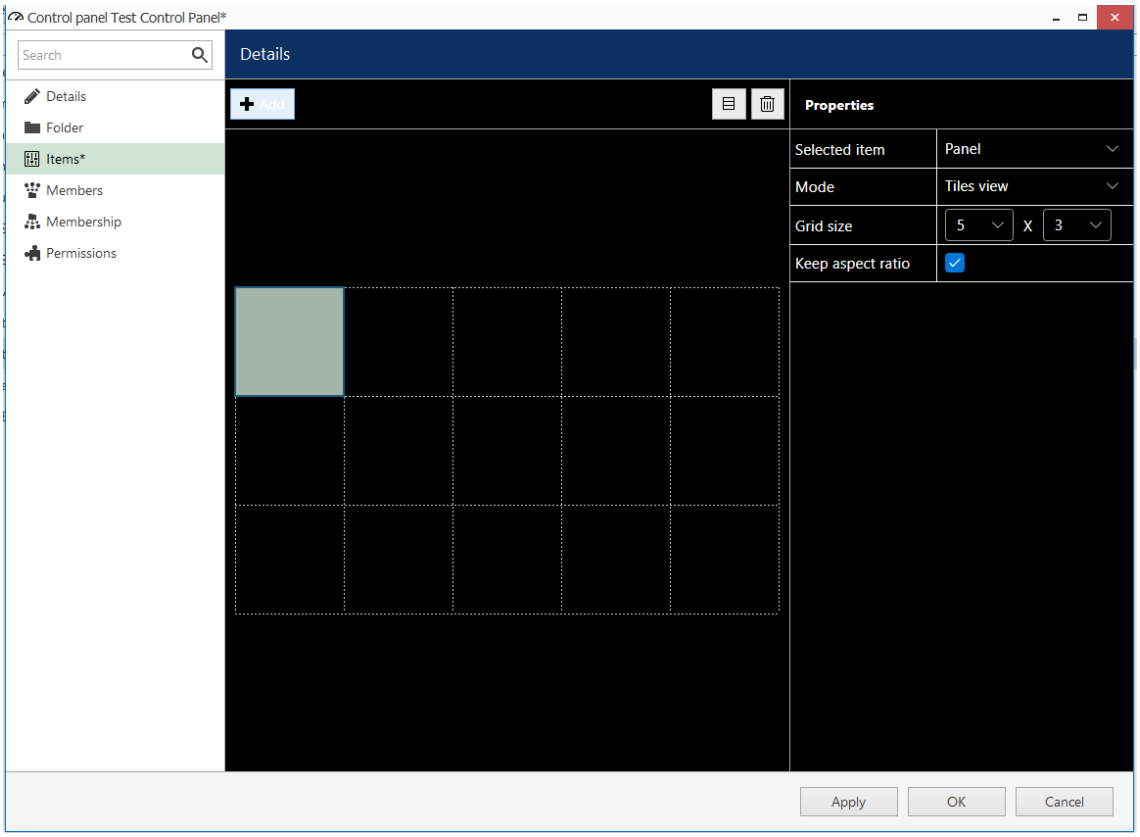

Figure 9. Selecting a target cell in Grid view.

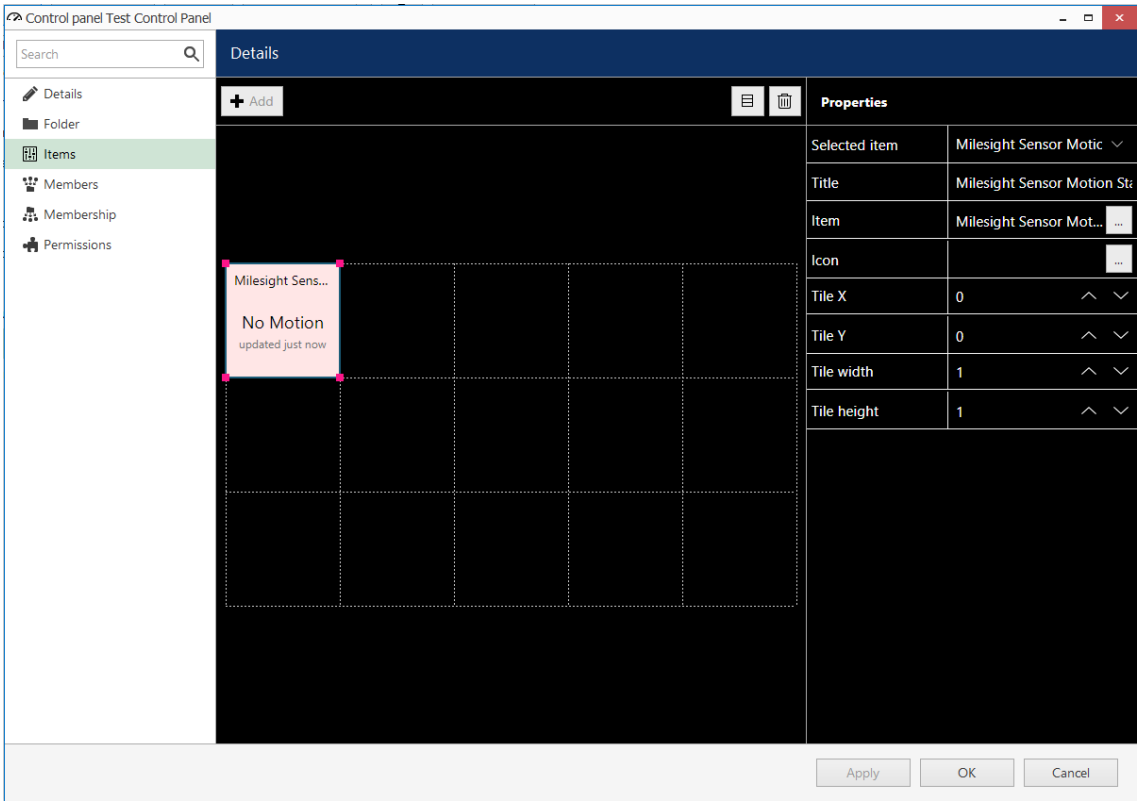

Figure 10. Motion indicator placed in the control panel.

8. Use the Integration in Luxriot EVO Monitor

Once the control panel is configured, the indicator state becomes available in Luxriot EVO Monitor. When the corresponding MQTT trigger event is received, the rule changes the displayed state. When the release event is received, the rule returns the indicator to its cleared state.

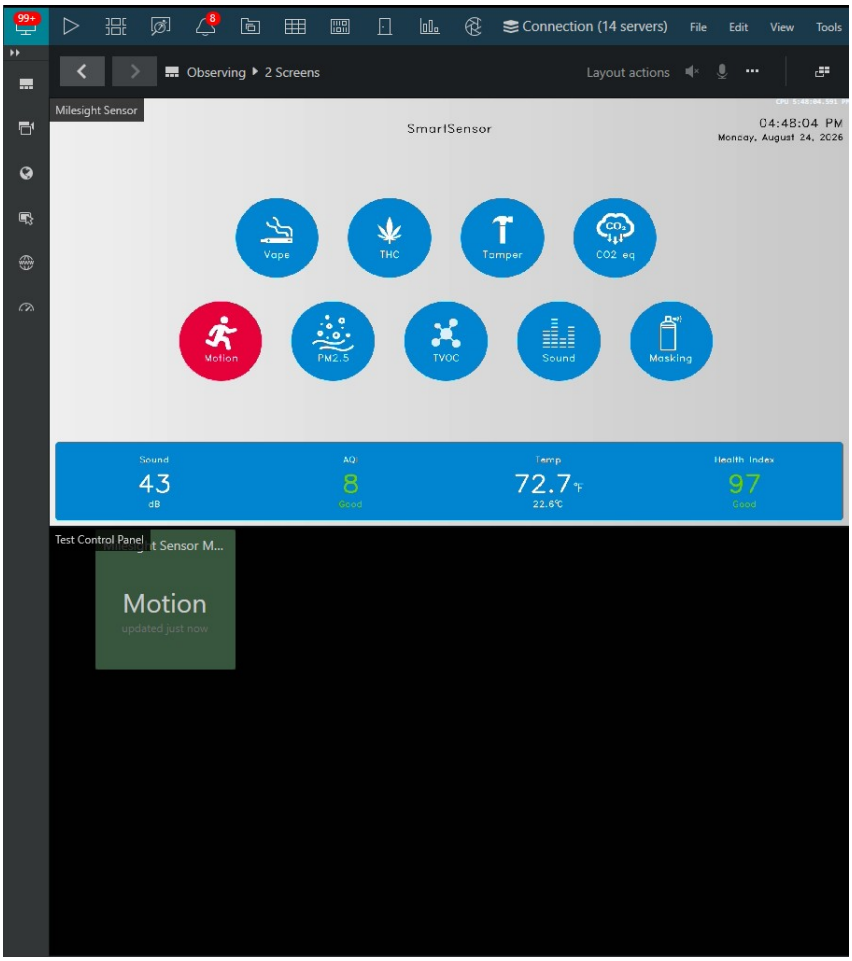

Figure 11. Milesight sensor information and Motion state displayed in Luxriot EVO Monitor.

Sensor video output

The Milesight sensor also provides an RTSP video output through ONVIF. To add it to Luxriot EVO, go to Devices > Add Device and use the ONVIF driver. The device exposes a single video channel that can be used alongside the sensor state information.

> **Scalability:** The same integration method can be repeated for any number of states required by the deployment. Create the needed indicator states, MQTT trigger/release events, and matching rules for each condition you want to display.

9. MQTT Payload Example and Troubleshooting

A compact Milesight Motion trigger payload contains the following structure within the JSON message:

```
{"trigger":{"Motion":{"trigger":true,"value":"1","threshold":"","data_source":"motion"}}}
```

A Motion release contains the same object name but reports release instead of trigger:

{"trigger":{"Motion":{"release":true,"value":"0","threshold":"","data_source":"motion"}}}

If an event does not fire

- Verify that the MQTT topic configured in the EVO event matches the topic used by the sensor.
- Confirm that Regular expression is enabled in the EVO event definition.
- Confirm the exact sensor object name in the received JSON. The match is sensitive to differences in the object text.
- Test the filter against the compact JSON actually received by EVO. The validated Motion filters are .*"trigger":\{"Motion":\{"trigger":true.* and .*"trigger":\{"Motion":\{"release":true.*.
- Confirm that the event is connected to the correct Change Indicator State action in Events & Actions Configurator.