

Scene Me

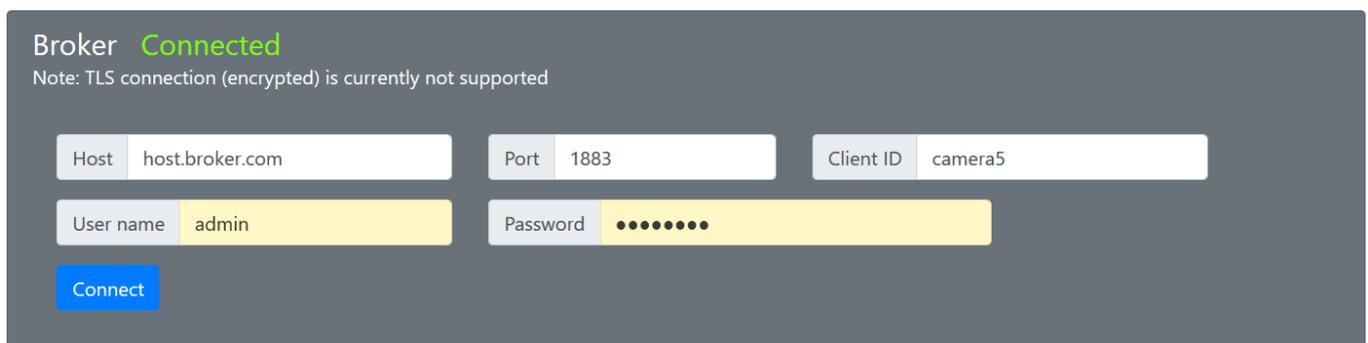
Highlights

- MQTT integration of Axis device events and analytics data
- Automatic MQTT message publish upon any device events
- Publish user defined MQTT messages upon specific events
- Subscribe to user-defined topics to initiate device actions
- Publish video analytics trackers and objects
- Define various specific analytics rules (Scenes) that are published on both MQTT and ONVIF event stream
- Detect and publish “faces in scene” on both MQTT and ONVIF event stream
Applicable for devices based on ARTPEC-6 and ARTPEC-7 using FW 9.20 or later
- Detect and publish “humans in scene” and “vehicles in scene” on both MQTT and ONVIF event stream
Applicable for devices based on ARTPEC-7 using FW 9.30 or later

Prerequisites and recommended tools

- MQTT broker. Install your own broker or use 3:rd party broker services
 - Eclipse Mosquitto <https://mosquitto.org/>
 - CloudMQTT <https://www.cloudmqtt.com/>
 - HIVEMQ <https://www.hivemq.com/>
- MQTT Snooper client for monitoring and testing
 - MQTT.fx (PC Java Client) <http://mqttfx.org/>
- Node-Red, Node.js, PHP or Python for system integration
- Mobile MQTT apps
 - Snoopers
 - Dashboard
 - GPS location publisher

MQTT Broker Configuration



Broker **Connected**
Note: TLS connection (encrypted) is currently not supported

Host	host.broker.com	Port	1883	Client ID	camera5
User name	admin	Password	●●●●●●		

[Connect](#)

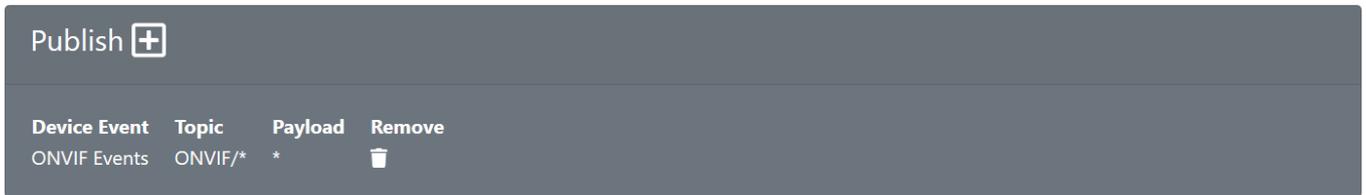
You need to supply broker address, port and a client ID (unique to the broker). If the broker does not require user/password, leave those fields empty.

Note that Scene Me does not support connecting to broker over TLS. This means that the traffic will not be encrypted. MQTT connections over Internet should be used with caution.

If you are unable to connect to your broker, check log for clues.

MQTT Publishing

Publish message on specific device events.



There is one predefined MQTT publish configuration. This will publish all the events detected in the Axis device to MQTT. The topic and payload are based on the ONVIF event stream topics and payload. See [MQTT Topic Message reference]. If there are no subscriber to ONVIF/# in the system, it is recommended to remove this publish by clicking the trashcan.

Add a new user-defined publish by clicking 



1. Select the event you want to trigger a publish
2. Define the topic that a system subscriber will listen to
3. Set the payload (See “Payload” below)
4. Define QoS. It is recommended to only use **QoS 0: Fire and forget** unless you know what you are doing and have subscribers that handles other QoS.
5. Set retained. Only enable this if you know what “retained” means and how it works.

Payload

Typically you would set some value, string or a JSON object in payload. There are some special controls for dynamic payloads:

Payload `!resolution=640x480` will publish a VGA JPEG snapshot. It is recommended to limit payload data size by keeping resolution to VGA or lower.

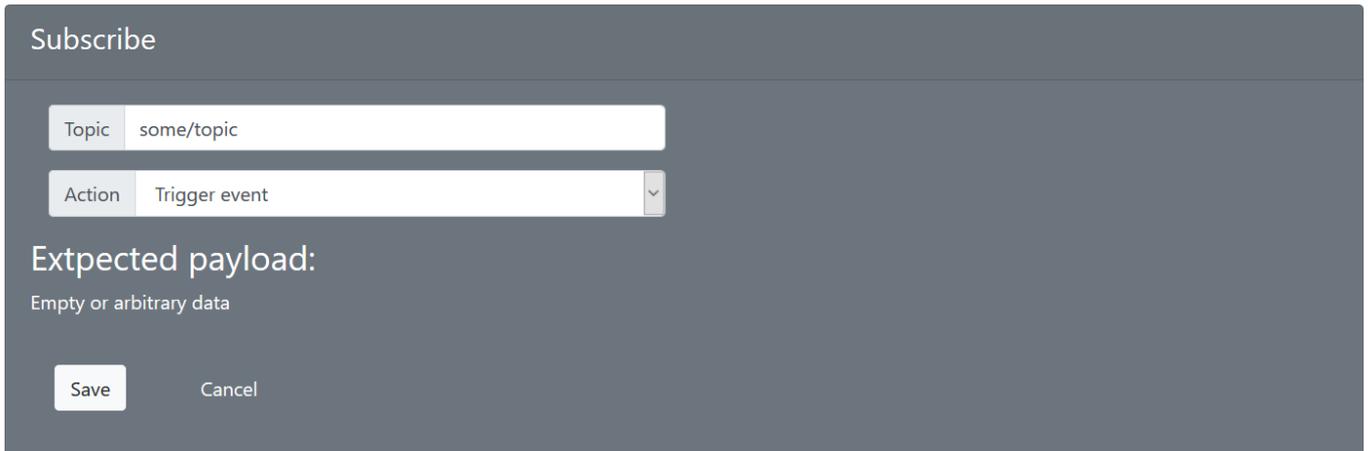
Payload `*` will include the ONVIF event payload

MQTT Subscriptions

Scene Me can subscribe to topics to initiate various actions in the Axis device.



Add a new subscription by clicking 

A dark grey dialog box titled "Subscribe". It contains a text input field for "Topic" with the value "some/topic", and a dropdown menu for "Action" with the value "Trigger event". Below these is the text "Expected payload:" followed by "Empty or arbitrary data". At the bottom are "Save" and "Cancel" buttons.

Subscribe

Topic some/topic

Action Trigger event

Expected payload:
Empty or arbitrary data

Save Cancel

1. Define the topic to subscribe to. It is possible to use wildcards '+' and '#'
2. Pick an action to initiate when that topic is published

Triggered event

This will register a new event in the Axis device. This event can be used to trigger actions in the Axis device. The event will also be published on the ONVIF Event stream, making it possible to initiate actions in the VMS.

The payload in the MQTT message is ignored for Triggered Event.

State Event

Similar to Triggered event with the difference that the event will be stateful (active/inactive). The payload published on MQTT must be "1", "true" or "high" to make the state active and "0", "false" or "low" to deactivate the state.

Image Request

Scene Me will respond with a JPEG snapshot payload on the `subscriber/topic/jpeg` based on the media profile published in the payload e.g. 'resolution=640x480'. It is recommended to limit payload data size by keeping resolution to VGA or lower.

Dynamic Text Overlay

Update image/video text overlay by publishing a string on MQTT. The camera's textoverlay string must be configured with '#D' that will be replaced with the text in the payload.

Note that text overlay will only be updated on an I-frame. If your video stream is configured to 30 P-frames, it may take up to one second before the update is visible.

Analytics

Coordinate system

The coordinate system of all analytics data is 0-1000 for both X and Y axels regardless of aspect ratio of the camera sensor. Origo [0,0] is placed top left corner.

Trackers

Trackers are published when detected objects moves across the scene. These can be used for real-time behavior filters. To increase system performance, trackers should be filtered to only publish relevant detections. A detected object must first pass a “birth area” before tracking is initiated. While inside the birth area, the object must fulfill minimum object size and minimum object age. Once the tracking has been initiated it will track the object outside birth area and all filters will be disabled. Object swaying is used to control how often the tracker will be published when objects move across the scene. Larger values (distance) will reduce the amount of published. This means that a video object that stops will not be published until it passes the swaying/distance limit.

It is recommended to disable publishing trackers if there are no consumers/subscribers in the system. Especially when multiple cameras are connected to the same MQTT broker.

Objects

When a tracker loses the object in video (object left scene) an “object” will be published. Object data includes birth/death coordinates and birth/death timestamps. Objects are primarily targeting post-analytics processing.

Scenes

Scenes are filters applied to trackers. With scenes you can create very specific “motion detectors”. You can have multiple scenes that publish different object behavior.

- Area of Interest. The detected objects center point must overlap this area
- Minimum object size
- Minimum vertical movement
- Minimum horizontal movement
- Type. If “face”, “human” and “car” is supported
- Minimum age

The scene will be published and active as soon the above criteria is fulfilled. The scene becomes inactive when the tracker loses the object and a an “inactive scene” will be published.

A configured scene will also fire an Axis device event than enables VMS or device rules to trigger without using MQTT.

Object Classifications

Axis cameras based on ARTPEC-6 can detect/classify faces. Axis device using ARTPEC-7 can detect/classify faces, humans and vehicles. “Faces In Scene”, “Cars In Scene” and “Humans In Scene” will be published on both MQTT and as device events.

Note: Object classification is default disabled. To enable, use a browser and make the following request:

```
http://<address>/local/sceneme/settings?set=vod&json={"enabled": true,"active": true}
```

Scene Me must be restarted.

MQTT Topic Message reference

Client Announcement

Published when connecting to broker

Topic: `connect/<clientID>` [RETAINED]

Payload:

```
{
  "connected": true,
  "client": "clientID",
  "model": "AXIS Pxxx",
  "type": "Network Camera",
  "serial": "ACCC8Exxxxxx",
  "mac": "AC:CC:8E:xx:xx:xx",
  "IPv4": "1.2.3.4",
  "firmware": "9.30.1",
  "aspect": "16:9",
  "rotation": 270
}
```

Last-Will Testament

Published by broker when device disconnects

Topic: `connect/<clientID>` [RETAINED]

Payload:

```
{
  "connected": false,
  "client": "clientID"
}
```

ONVIF Device Events

All events detected in the Axis device will be automatically published. It is recommended to remove this publishing if you do not have any clients/subscribers. Specific device events can be added individually with a user-defined topic/payload.

Topic: `ONVIF/<clientID>/[ONVIF topic path]`

Payload:

```
{
  "client": "clientID",
  "device": "ACCC8Exxxxxx",
  "topic": "[ONVIF TOPIC Path]",
  "payload": "ONVIF event payload",
  "timestamp": UTC milliseconds
}
```

Scenes

Published when user-defined scenes (analytics rules) occur

Topic: `scene/<clientID>/<sceneID>/<state>`

The `<state>` will be "true" or "false"

Payload: JSON object depending on what initiated the scene

Sensors

The Axis device CPU load will automatically publish every 15 minutes.

Topic: `sensor/<clientID>/cpu` Payload: number (Linux loadavg 15)

The Axis device bandwidth produced will automatically publish every 3 minutes.

Topic: `sensor/<clientID>/network` Payload: number (Average kbps integrated over 3 minutes)

Tracker

Analytics object tracked when moving in the scene

Topic: `tracker/<clientID>`

Payload:

```
{
  "id": number,           Unique tracker ID
  "timestamp": number,   UTC milliseconds
  "x": 557,              Left 0-1000
  "y": 248,              Top
  "w": 139,              Width 0-1000
  "h": 481,              Height 0-1000
  "cx": 626,             Center X of the object 0-1000
  "cy": 488,             Center Y of the object 0-1000
  "dead": false,        Dead = true means the object is lost (exit the scene)
  "sx": 565,             Center X birth coordinate
  "sy": 517,             Center Y birth coordinate
  "px": 565,             Previous X center position (used to create paths)
  "py": 772,             Previous Y bottom position (used to create paths)
  "dx": 61,              Total X distance travelled
  "dy": -29,            Total Y distance travelled
  "birth": 1568833494879, Birth Timestamp
  "human": 0,            If supported, number of classification hits
  "car": 0,              If supported, number of classification hits
  "age": 382,            Total age in milliseconds (timestamp - birth)
}
```

If the Axis device is a Radar (D2050), the following properties are included in tracker

```
"class": 2,             Radar object classification
"speed": 9.3,           Object speed on m/s
"direction": 261.6,     Object direction angle in degrees
"distance": 13.1,       Object distance from Radar in meter
"angle": 34,            Object location from Radar in degrees
}
```

Object

Analytics object published when scene object (tracker) is lost.

Topic: `object/<clientID>`

Payload:

```
{
  "id": 648040,           Unique tracker ID
  "x": 823,              Left position when first detected
  "y": 229,              Top position when first detected
  "w": 77,               Width when first detected
  "h": 364,              Height when first detected
  "sx": 861,             Center X position when first detected
  "sy": 411,             Center Y position when first detected
  "cx": 12,              Center X position when tracker lost (last known position)
  "cy": 747,             Center Y position when tracker lost (last known position)
  "dx": -849,            Total distance X traveled
  "dy": 336,             Total distance Y traveled
  "human": 0,            If available, the number of times classified as human
  "car": 0,              If available, the number of times classified as car
  "birth": 1568833564319, UTC birth timestamp
  "death": 1568833569724, UTC death timestamp
  "age": 5405,           Total age (death - birth)
  "pause": 600,          Longest pause during tracking
  "tx": 799,             Center X of the longest pause
  "ty": 418,             Center Y of the longest pause
}
```