O3 Sensor Hub 2.0

MQTT API Reference Guide

Document version: 1.0

Contents

Introduction

This guide describes the MQTT topics associated with the O3 Sensor Hub 2.0. These topics are used to publish and read data from the sensor hub and to send commands to control outputs.

The information in this guide applies to the following sensor hub models:

- O3-HUB2
- O3-HUB2-2xP

This guide is intended for the developer or technical person who wants to use MQTT to interact with data from the sensor hub through a broker or custom application.

About the O3 Sensor Hub 2.0

The O3 Sensor Hub 2.0 is a ceiling-mounted multisensor device that integrates temperature, humidity, motion, sound, and light sensing. Using sensor fusion technology and machine learning algorithms, the sensor hub delivers fast, accurate feedback on the monitored space.

The sensor hub supports BACnet, MQTT, and Bluetooth® Low Energy (BLE) protocols, allowing it to integrate with almost any system. Select models are equipped with two universal I/O points for controlling or accepting inputs from nearby devices.

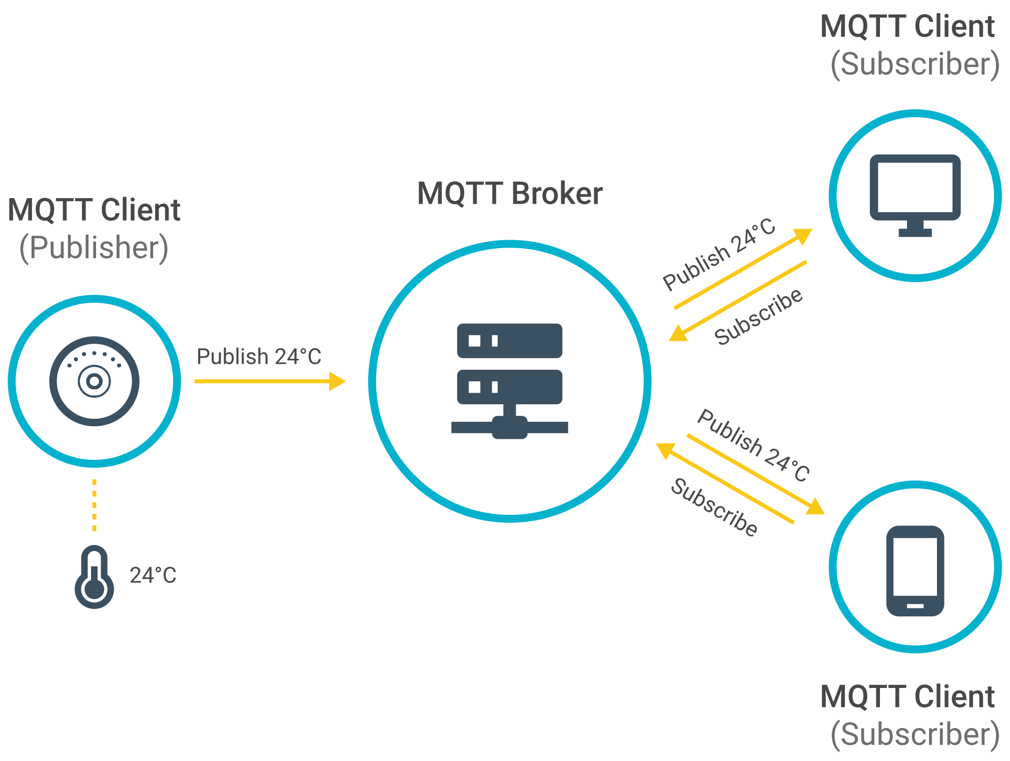
Note: Some features listed in this document may only be available in select models or firmware versions.

For more information about the sensor hub, go to support.o3hub.com.

About MQTT

MQTT is a lightweight publish/subscribe messaging protocol built on TCP/IP. Originally designed for monitoring oil pipelines (as *MQ Telemetry Transport*), MQTT is now the standard messaging and data exchange protocol for the Internet of Things.

An MQTT **client** is any device that runs an MQTT library and connects to an MQTT **broker** (server) over a network. Unlike in the traditional client-server model where clients communicate directly with an endpoint, in the publish/subcribe ("pub/sub") model, clients communicate through a central **broker**. The client that sends a message (the **publisher**) and the client that receives that message (the **subscriber**) never interact. The broker filters incoming messages from the publisher and distributes them to the subscriber(s).



Messages are filtered using topics. A **topic** is a UTF-8 string that the broker uses to to decide which subscriber receives which message. The topic consists of one or more topic levels, with each topic level separated by a forward slash (topic level separator). A topic must be at least one character long and topic names are case-sensitive.

For example, in the topic `events/object/occupantTemperature`, the first level is "events", the second level is "object", and the third level is "occupantTemperature".

When a client subscribes to a topic, it can subscribe to the exact topic of a published message or it can use wildcards to subscribe to multiple topics at once. There are two types of **wildcards**: single-level (+) and multi-level The single-level wildcard replaces one topic level, while the multi-level wildcard replaces multiple topic levels. The multi-level wildcard must always be the last character in the topic string and must be preceded by a forward slash.

For example: `events/object/#`

Note: Wildcards cannot be used when publishing to a topic.

Senders and receivers set a **Quality of Service (QoS)** level that defines how much effort will be expended to ensure that the message is delivered.

There are three QoS levels in MQTT:

Level 0	The message is delivered *at most once*, according to the best efforts of the underlying TCP/IP network. There is no guarantee of delivery. You can think of this approach as send and forget.
Level 1	The message is delivered *at least once*, with multiple retries until the message is acknowledged as received. You can think of this approach as acknowledged delivery.
Level 2	The message is delivered *exactly once*, with the sender and receiver doing a handshake to ensure that the message is received once by the intended recipient. You can think of this approach as guaranteed delivery.

Both the subscriber and publisher can set a QoS level. If the levels don't match, the lower of the two is used. For example, if a client subscribes to a message at a QoS of 1 but the message was published at a QoS of 0, then the client will receive the message at a QoS of 0.

The content of a message is referred to as the **payload**. For example, the payload of a subscribe message will contain a list of topic names to which the client wants to subscribe, as well as the QoS level at which the client wants to receive the messages.

For more information about MQTT, go to https://mqtt.org.

MQTT and the O3 Sensor Hub 2.0

Note: The sensor hub currently supports the MQTT v3.1.1 specification.

The MQTT topics associated with the sensor hub can be used to:

- publish sensor hub data to an external broker (connection settings are configured in O3 Setup mobile app)
- subscribe to sensor hub data, either from an external broker (recommended) or from the hub's internal Eclipse Mosquitto™ broker
- send commands to change the sensor hub's setpoints and outputs

Sensor hub data can be stored in the cloud via MQTT, where it can then be aggregated, analyzed, and integrated with other site data through custom-built apps.

Topic Format

The sensor hub's sensor and device properties are represented by BACnet objects, which are mapped to MQTT topics through an internal BACnet-to-MQTT bridge. The MQTT topics are divided into "events" topics and "commands" topics, indicating the direction of the message. "Events" topics (`events/object/SUBTOPIC`) are used to report BACnet object values (from sensors, outputs, etc.), while "commands" topics (`commands/object/SUBTOPIC`) are used to set BACnet object values.

To learn more about the BACnet objects associated with the sensor hub, refer to the *O3 Sensor Hub 2.0 BACnet Application Guide* on support.o3hub.com.

Message Format

The sensor hub supports key-value content in JSON (JavaScript Object Notation) format as follows:

```
{"key1": "value1"}
```

The key is always a string, while the value can be either a string, number, boolean, object, array, or null. Spaces before or after the colon are optional, as JSON ignores whitespace between elements.

For example, the `events/object/occupantTemperature` topic looks like this:

```
{"Present_Value": 24.51,"Units": "°C","updated": "31-08-2020
23:09:26.21","status": 0}
```

Getting Started

This chapter describes how to connect to a broker and subscribe/publish to topics using a third-party MQTT client application.

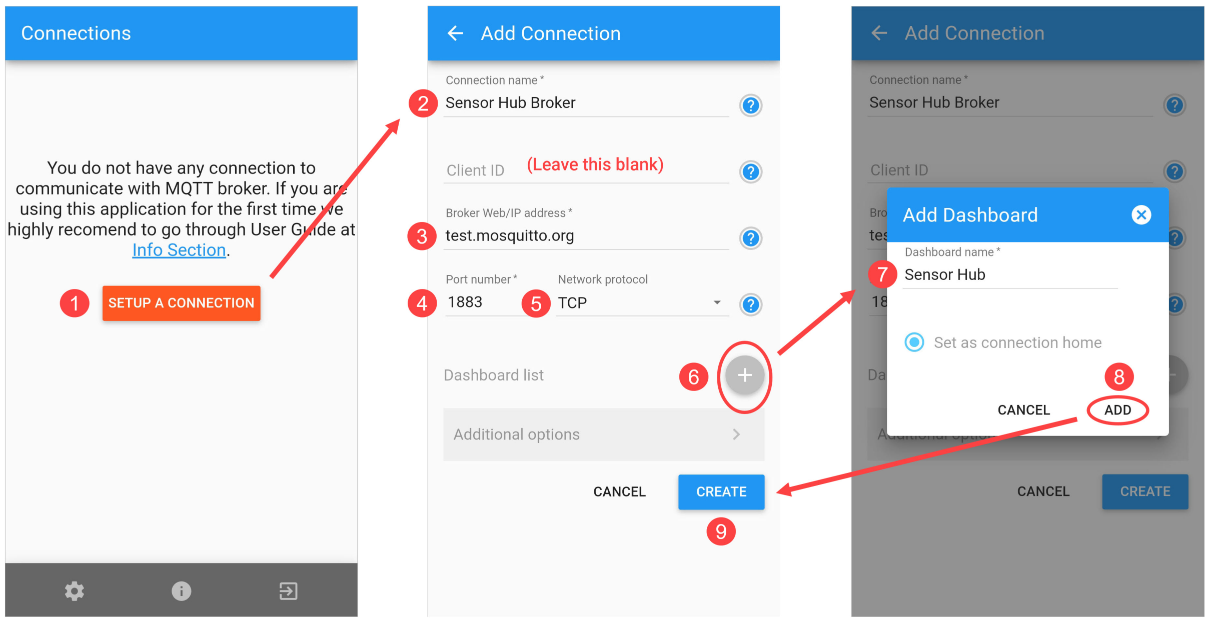
Mobile MQTT Client Example

For this example, we will use IoT MQTT Panel, a free mobile app available on Google Play.

Connect to the broker

Open the IoT MQTT Panel app and tap **Setup a Connection**. Enter the name, address, and port of the MQTT broker that you are connecting to, add a dashboard, and then tap **Create**.

- To connect to the sensor hub's internal MQTT broker, use the sensor hub's IP address.
- The default MQTT port over TCP is **1883**. For TLS connections, use **8883**.
- If you want to publish, you will need to get a user name and password for the broker. You can enter this information under **Additional options**.



You do not have any connection to communicate with MQTT broker. If you are using this application for the first time we highly recomend to go through User Guide at Info Section.

Connection name *: Sensor Hub Broker

Client ID: (Leave this blank)

Broker Web/IP address *: test.mosquitto.org

Port number *: 1883

Network protocol: TCP

Dashboard name *: Sensor Hub

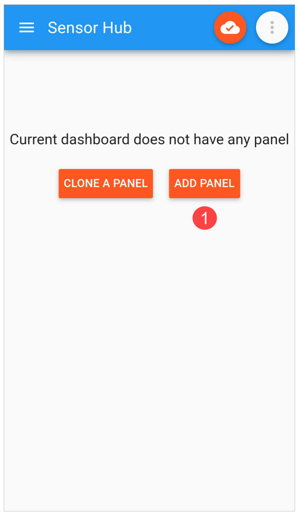
Set as connection home

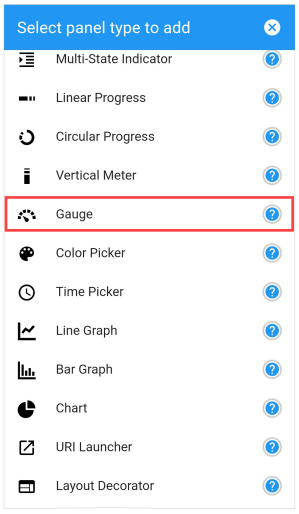
Subscribe to a topic

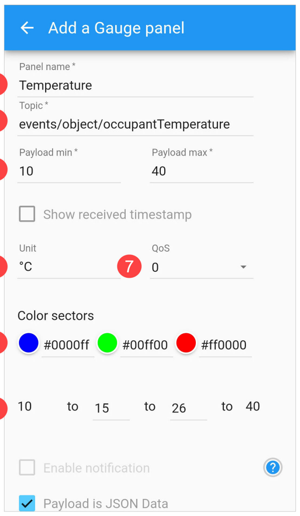
On the Connections page, you should see the connection that you have created. Tap the cloud icon to connect to the broker. The cloud icon changes color and shows a check mark when a connection to the broker is established.

Next, the dashboard page opens and you are prompted to create a panel. Tap **Add Panel**, select a panel type, enter the details for the panel, and then tap **Create**.

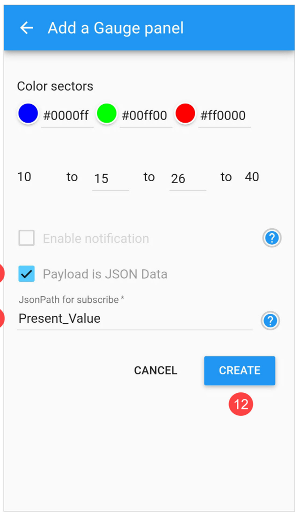
- Subscribe topics on the sensor hub follow the format `events/object/SUBTOPIC`. For example, `events/object/occupantTemperature`.
- The JSON key you want to read is `Present_Value`.





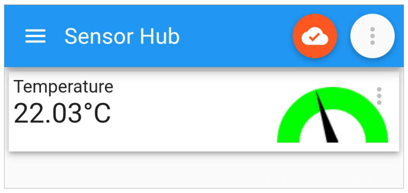


Panel name *: Temperature
Topic *: events/object/occupantTemperature
Payload min *: 10
Payload max *: 40
Show received timestamp: ☐
Unit: °C
QoS: 0
Color sectors: ● #0000ff ● #00ff00 ● #ff0000
10 to 15 to 26 to 40
Enable notification: ☐
Payload is JSON Data: ✓



Color sectors: ● #0000ff ● #00ff00 ● #ff0000
10 to 15 to 26 to 40
Enable notification: ☐
Payload is JSON Data: ✓
JsonPath for subscribe *: Present_Value

The created panel looks like this:



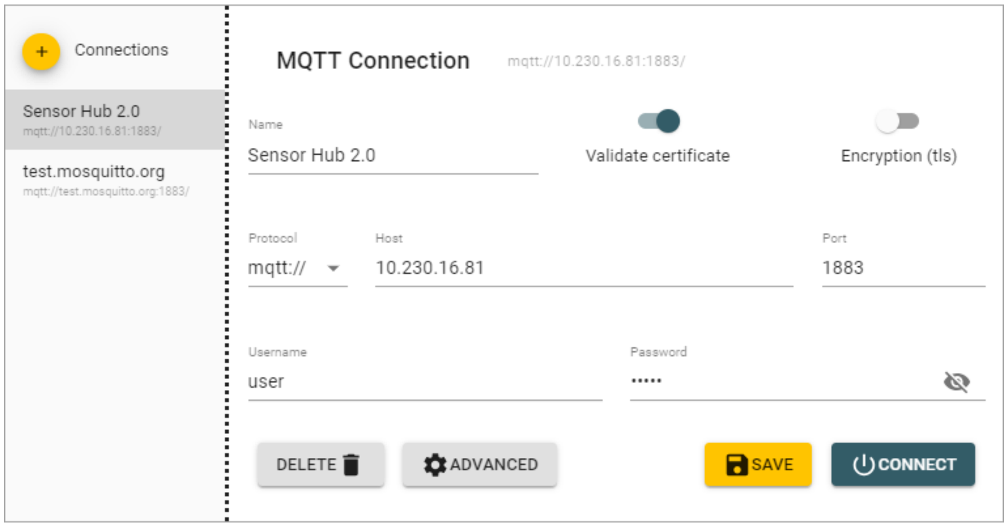
Temperature
22.03°C

Desktop MQTT Client Example

For this example, we will use MQTT Explorer, a desktop MQTT client that lets you visualize, publish, subscribe, and plot topics. You can download it from http://mqtt-explorer.com.

Connect to the broker

Open MQTT Explorer and click the **Connections** button to add a new connection. Enter the name, address, and port of the MQTT broker that you are connecting to. In this example, we will be publishing to a topic, so we will also enter a user name and password. Click **Save** to save the settings, then click **Connect**.



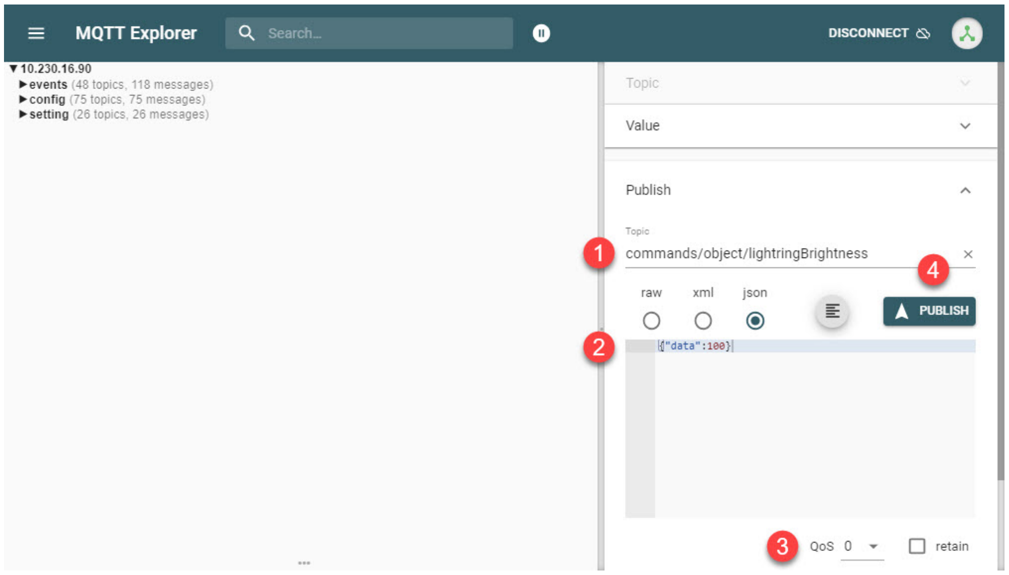
Name: Sensor Hub 2.0
Validate certificate: on
Encryption (tls): off
Protocol: mqtt://
Host: 10.230.16.81
Port: 1883
Username: user
Password: •••••

Note: If you are publishing to the sensor hub's internal MQTT broker for the first time, use the following login credentials: username = **user**, password = **admin**. The username and password are case-sensitive. After logging in, change the default password in setting/mqttPassword.

Publish to a topic

The main window is divided into two panes. On the left is a tree view of topics. On the right are panels for Topic, Value, Publish, and Stats. Expand the Publish panel and enter the topic that you want to publish to. Select **json** and enter the key-value pair that you want to modify. The value should be the new value that you want to publish. Select a **QoS** level, then click **Publish**.

- Publish topics on the sensor hub follow the format `commands/object/SUBTOPIC`. For example, `commands/object/lightringBrightness`.
- The JSON key you want to write to is `data`.



```
▼10.230.16.90
  ▶events (48 topics, 118 messages)
  ▶config (75 topics, 75 messages)
  ▶setting (26 topics, 26 messages)
```

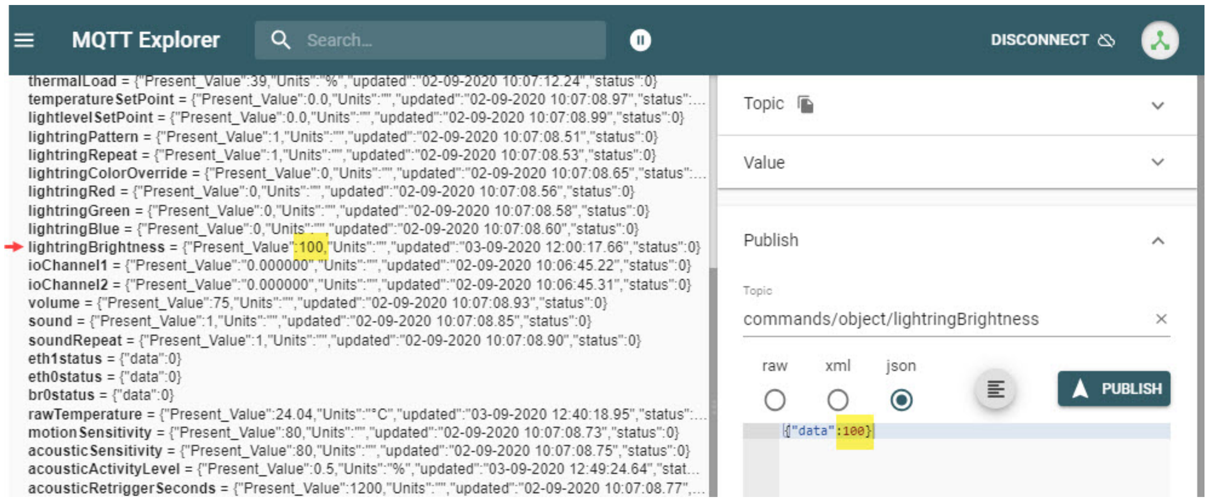
Topic: commands/object/lightringBrightness

raw ( ) xml ( ) [✓ json]

```
{"data":100}
```

QoS 0 ☐ retain

After you click **Publish**, you can look up the corresponding subtopic under `events/objects` and verify that the `Present_Value` has changed.



```
thermalLoad = {"Present_Value":39,"Units":"%","updated":"02-09-2020 10:07:12.24","status":0}
temperatureSetPoint = {"Present_Value":0.0,"Units":"","updated":"02-09-2020 10:07:08.97","status":...
lightlevelSetPoint = {"Present_Value":0.0,"Units":"","updated":"02-09-2020 10:07:08.99","status":0}
lightringPattern = {"Present_Value":1,"Units":"","updated":"02-09-2020 10:07:08.51","status":0}
lightringRepeat = {"Present_Value":1,"Units":"","updated":"02-09-2020 10:07:08.53","status":0}
lightringColorOverride = {"Present_Value":0,"Units":"","updated":"02-09-2020 10:07:08.65","status":...
lightringRed = {"Present_Value":0,"Units":"","updated":"02-09-2020 10:07:08.56","status":0}
lightringGreen = {"Present_Value":0,"Units":"","updated":"02-09-2020 10:07:08.58","status":0}
lightringBlue = {"Present_Value":0,"Units":"","updated":"02-09-2020 10:07:08.60","status":0}
lightringBrightness = {"Present_Value":100,"Units":"","updated":"03-09-2020 12:00:17.66","status":0}
ioChannel1 = {"Present_Value":"0.000000","Units":"","updated":"02-09-2020 10:06:45.22","status":0}
ioChannel2 = {"Present_Value":"0.000000","Units":"","updated":"02-09-2020 10:06:45.31","status":0}
volume = {"Present_Value":75,"Units":"","updated":"02-09-2020 10:07:08.93","status":0}
sound = {"Present_Value":1,"Units":"","updated":"02-09-2020 10:07:08.85","status":0}
soundRepeat = {"Present_Value":1,"Units":"","updated":"02-09-2020 10:07:08.90","status":0}
eth1status = {"data":0}
eth0status = {"data":0}
br0status = {"data":0}
rawTemperature = {"Present_Value":24.04,"Units":"°C","updated":"03-09-2020 12:40:18.95","status":...
motionSensitivity = {"Present_Value":80,"Units":"","updated":"02-09-2020 10:07:08.73","status":0}
acousticSensitivity = {"Present_Value":80,"Units":"","updated":"02-09-2020 10:07:08.75","status":0}
acousticActivityLevel = {"Present_Value":0.5,"Units":"%","updated":"03-09-2020 12:49:24.64","stat...
acousticRetriggerSeconds = {"Present_Value":1200,"Units":"","updated":"02-09-2020 10:07:08.77",...
```

Topic: commands/object/lightringBrightness

raw ( ) xml ( ) [✓ json]

```
{"data":100}
```

MQTT Topics

This chapter lists the MQTT topics associated with the O3 Sensor Hub 2.0. You can subscribe to "events" topics and publish to "commands" topics.

Subscribe Topics

Object Topics

You can subscribe to read-only object topics that contain sensor data and device configurations. Subscribe using the format `events/object/SUBTOPIC` and read the `Present_Value` JSON key. To subscribe to all object topics, use `events/object/#`.

TABLE 1: SENSOR DATA OBJECT SUBSCRIBE TOPICS

Topic	Description
events/object/occupantTemperature	Temperature at 1 m (3 ft) above the floor. This is a composite value derived from the sensor hub's internal temperature sensors and the IR temperature sensor. Range: 0°C to 59°C (32°F to 138°F).
events/object/irTemperature	Average temperature of surfaces in the sensor hub's field of view. Range: 0°C to 59°C (32°F to 138°F).
events/object/internalTemperature	Temperature at ceiling height. Range: 0°C to 59°C (32°F to 138°F).
events/object/rawTemperature	Uncalibrated occupant temperature. Range: 0°C to 59°C (32°F to 138°F).
events/object/occupantHumidity	Humidity at 1 m (3 ft) above floor. This is calculated from the occupant temperature and internal humidity using psychrometrics. Range: 0% to 100%.
events/object/internalHumidity	Humidity at ceiling height. Range: 0% to 100%.
events/object/combinedOccupancy	Combined (motion + sound) occupancy signal. Active state when motion and sound is detected. Range: 0 (not occupied), 1 (occupied).
events/object/motion	Motion occupancy signal. Active state when motion is detected. Range: 0 (not occupied), 1 (occupied).

TABLE 1: SENSOR DATA OBJECT SUBSCRIBE TOPICS

Topic	Description
events/object/acousticActivityLevel	Audio level after certain frequencies are filtered out. Range: 0 to 65535.
events/object/acousticOccupancy	Acoustic occupancy signal. Active state when audio level is above acoustic occupancy threshold. Range: 0 (not occupied), 1 (occupied).
events/object/motionSensitivity	Controls the sensitivity of the PIR sensor to changes in movement levels within the detection area. Range: 0% to 100%. 100% = maximum sensitivity.
events/object/acousticSensitivity	Controls the sensitivity of the acoustic occupancy sensor to changes in audio levels within the detection area. Range: 0% to 100%. 100% = maximum sensitivity.
events/object/acousticRetriggerSeconds	The amount of time that activity sounds can cause the sensor hub to remain in the occupied state after motion is detected. Default value is 1200 seconds (20 minutes). Measured from most recent motion detection event.
events/object/occupancyInactivitySeconds	The amount of time it takes the sensor hub to return to the unoccupied state when no motion and no audio activity is detected. Default value is 30 seconds (5 minutes).
events/object/acousticBGUpdateSeconds	Update period for the baseline microphone levels to adjust to environmental changes when no occupants are present. Default value is 30 seconds.
events/object/lightLevel	Brightness of ambient light (lux or foot-candle). Range: 0 to 65535 (lux) or 0 to 6088 (foot-candle).
events/object/colorTemperature	Color temperature of ambient light (K). Range: 0 to 65535.
events/object/colorRed	Red component of ambient light. Range: 0 to 65535.

TABLE 1: SENSOR DATA OBJECT SUBSCRIBE TOPICS

Topic	Description
events/object/colorGreen	Green component of ambient light. Range: 0 to 65535.
events/object/colorBlue	Blue component of ambient light. Range: 0 to 65535.
events/object/soundLevel	Level of ambient noise (dB SPL). Unfiltered audio level across the entire spectrum. Range: 0 to 120.
events/object/thermalLoad	*Not currently supported.*

TABLE 2: DEVICE CONFIGURATION OBJECT SUBSCRIBE TOPICS

Topic	Description
events/object/temperatureSetPoint	User-entered temperature. Measured by user at occupant height. Offset is calculated by the sensor hub.
events/object/lightlevelSetPoint	User-entered light level. Records the light level read by the hub (AI12) when the lighting in the space is set to the desired brightness. This setpoint can be retrieved later by the control system to set the feedback loop, etc.
events/object/lightringPattern	Pre-defined light ring pattern (1-13). Default value is 1 (Off).
events/object/lightringRepeat	Number of times light ring pattern repeats. Default value is 1.
events/object/lightringRed	Red component of light ring RGB value. Range: 0 to 100. Default value is 0 (Off).
events/object/lightringGreen	Green component of light ring RGB value. Range: 0 to 100. Default value is 0 (Off).
events/object/lightringBlue	Blue component of light ring RGB value. Range: 0 to 100. Default value is 0 (Off).
events/object/lightringColorOverride	Custom light ring color defined by lightringRed, lightringBlue, and lightringGreen subtopics. Range: 0 (Off), 1 (On). When set to 1 (On), it overrides lightringPattern.
events/object/lightringBrightness	Overall brightness of light ring. Range: 0 to 100. Default value is 50.

TABLE 2: DEVICE CONFIGURATION OBJECT SUBSCRIBE TOPICS

Topic	Description
events/object/soundinfo	Lists the names and index numbers of all the sound files and their total size. There are 25 default sounds and you can create up to 25 custom sounds. For more details about creating custom sounds, see the *O3 Sensor Hub 2.0 BACnet Application Guide* on support.o3hub.com. This topic is updated automatically whenever a custom sound file is added, changed, or removed.
events/object/volume	Speaker volume. Range: 0 to 100. Default value is 0 (Off).
events/object/soundRepeat	Number of times a sound is played. Default value is 1.
events/object/startupSoundEnable	Startup sound. Range: 0 (Off), 1 (On). Default value is 0 (Off).
events/object/bleMac	Displays the MAC address of Bluetooth LE beacon.
events/object/firmwareInfo	Displays the installed firmware version.
events/object/ioChannel1	*2xP models only:* Displays the xP1 universal point value.
events/object/ioChannel2	*2xP models only:* Displays the xP2 universal point value.

Publish Topics

Object Topics

Publish to an object topic using the format `commands/object/SUBTOPIC` and write to the `data` JSON key.

For example, to change the brightness of the sensor hub's LED light ring from 50% to 100%, publish `{"data": 100}` to `commands/object/lightringBrightness`. The corresponding subscribe topic `events/object/lightringBrightness` will then show `{"Present_Value": 100}`.

TABLE 3: OBJECT PUBLISH TOPICS

Topic	Description
commands/object/lightringPattern	Plays pre-defined light ring pattern (1-13). Default value is 1 (Off).
commands/object/lightringRepeat	Sets number of times light ring pattern repeats. Default value is 1.
commands/object/lightringRed	Sets red component of light ring RGB value. Range: 0 to 100. Default value is 0 (Off).
commands/object/lightringGreen	Sets green component of light ring RGB value. Range: 0 to 100. Default value is 0 (Off).
commands/object/lightringBlue	Sets blue component of light ring RGB value. Range: 0 to 100. Default value is 0 (Off).
commands/object/lightringColorOverride	Activates custom light ring color defined by lightringRed, lightringBlue, and lightringGreen subtopics. Set to 1 (On) or 0 (Off). When set to 1 (On), it overrides **commands/object/lightringPattern**.
commands/object/lightringBrightness	Sets overall brightness of light ring. Range: 0 to 100. Default value is 50.
commands/object/indicateStatus	Displays power-on self-test (POST) status. Set to 1 (On) or 0 (Off). POST status is indicated by the following colors: green = OK, red = sensor problem, blue = Bluetooth problem, and yellow = Ethernet problem. Default value is 0 (Off).

TABLE 3: OBJECT PUBLISH TOPICS

Topic	Description
commands/object/soundfile	Plays a sound by sound file name. For example, **{"data": "(1) Power On.wav"}**. See **commands/object/soundinfo**.
commands/object/volume	Sets the speaker volume. Range: 0 to 100. Default value is 0 (Off).
commands/object/soundRepeat	Sets the number of times a sound is played. Default value is 1.
commands/object/soundinfo	Lists the names and index numbers of all the sound files and their total size (in bytes). No JSON content needs to be entered. Simply enter 0 and then publish.
commands/object/ioChannel1	*2xP models only:* Sets the xP1 universal point value, if applicable.
commands/object/ioChannel2	*2xP models only:* Sets the xP2 universal point value, if applicable.

Firmware and Reboot Topics

TABLE 4: FIRMWARE AND REBOOT PUBLISH TOPICS

Topic	Description
commands/fw/updateFw	Initiates a firmware upgrade. Indicate the firmware version using the format **{"data": {"version": "*FIRMWARE_VERSION*"}}**.
commands/reboot	Reboots the sensor hub using the following message: **{"data": {"source": "enteliWeb", "message": "command coldstart"}}**.

Setting Topics

If you are connected to the sensor hub's internal MQTT broker, you can directly modify the hub's configuration file.

Modify a setting using the format `setting/TOPIC/SUBTOPIC` and write to the `data` JSON key. The change is then applied to `config/TOPIC/SUBTOPIC`.

For example, to change the unit for temperature from the default Celsius (°C) to Fahrenheit (°F), publish `{"data": "F"}` to `setting/units/temp`. To change it back to Celsius, publish `{"data": "C"}`. (Do not include the degree symbol °)

TABLE 5: SETTING TOPICS

Topic	Description
setting/bacnet/eth/Enable	Enables BACnet/Ethernet when set to true. Default value is true.
setting/bacnet/bnip/Enable	Enables BACnet/IP when set to true. Default value is false.
setting/bacnet/bnip/IpMode	*Not currently supported.* The topic **config/bacnet/bnip/IpMode** is read-only. To set up the sensor hub as a foreign device, you must use a BACnet front-end application. For more details, see the *O3 Sensor Hub 2.0 BACnet Application Guide* on support.o3hub.com.
setting/bacnet/bnip/udpPort	Sets the BACnet/IP UDP port. Default value is 47808.
setting/bacnet/device_name	Sets the BACnet device name.
setting/bacnet/id	Sets the BACnet device ID. Must be a unique value between 1 and 4194302. Default value is based on Ethernet MAC address.
setting/ble/enable	Enables the Bluetooth LE communication when set to true. Default value is true.
setting/ble/pin	Sets the 6-digit authentication code for read/write access via Bluetooth. Default value is 000000. Reboot the device after changing the PIN.
setting/ble/txPower	Sets the Bluetooth maximum transmit power. Must be a value between 1 and 8, corresponding to the following states: -40, -20, -16, -12, -8, -4, 0, 4 dBm.
setting/fw/urlFw	Sets the location of the server where firmware upgrades are stored.

TABLE 5: SETTING TOPICS

Topic	Description
setting/io/1/setup	*2xP models only:* Sets the xP1 universal point type (analog-input, analog-output, binary-input, binary-output) and configuration (0-5-volt, 0-10-volt, 10kohm-thermistor). For example, **{"data": {"type": "analog-input", "option": "0-5-volt"}}**.
setting/io/2/setup	*2xP models only:* Sets the xP2 universal point type (analog-input, analog-output, binary-input, binary-output) and configuration (0-5-volt, 0-10-volt, 10kohm-thermistor). For example, **{"data": {"type": "analog-output", "option": "0-10-volt"}}**.
setting/mqttPassword	Sets the password to access the internal MQTT broker. Default value is admin. *This password must be changed after initial login to prevent unauthorized use of the broker.* The password is case-sensitive and must be 5 to 31 characters in length. Special characters are allowed. Reboot the device after changing the password.
setting/mqttPort	Sets the port used by the internal MQTT broker. Default value is 1883 (TCP). For TLS connections, set the value to 8883.
setting/NTP	Enables Network Time Protocol service when set to true, causing the sensor hub to display UTC time. Default value is true.
setting/units/light	Sets the illuminance unit of measurement: lx (lux) or ft-c (foot-candle). Default value is lx.
setting/units/temp	Sets the temperature unit of measurement: C (Celsius) or F (Fahrenheit). Default value is C.

Revision History

Version	Date	Description
1.0	January 8, 2021	New document.