

MQTT And Azure Client Communication Driver

This document has the specific information related to this driver configuration. For a generic explanation on Device Module, Channels, Nodes and Points configuration, please refer to reference guide.

Contents

Section 1 – Summary Information	2
Section 2 – Channel Configuration	2
Protocol Options	2
Section 3 – Node Configuration.....	2
Station Configuration	2
Station Examples.....	3
Section 4 – Point Configuration	4
Address.....	4
Section 5 – Troubleshoot	4
Section 6 – Common Errors	4
Revision History	5
Append – How to Set up Azure IoT Hub	5

Section 1 – Summary Information

Communication Driver Name: MQTTAzure

Implementation DLL: T.ProtocolDriver.MQTTAzure.dll

Interface: TCPIP

Protocol: MQTT (Message Queuing Telemetry Transport) is a machine-to-machine (M2M)/"Internet of Things" connectivity protocol.

PC Hardware requirements: Ethernet board

Implemented Methods: Connect, Disconnect, Subscribe, UnSubscribe and Publish.

Section 2 – Channel Configuration

Protocol Options

Not used in this driver.

Section 3 – Node Configuration

Station Configuration

Station syntax:

`<URL>; <Port>; <ClientID>; <Username>; <Password>; <SslProtocol>; [X509Certificate]; <PayloadFormat>`

Where:

< URL > = MQTT Broker (Server) name. It must be the same name configured in host Name parameter in Azure IoT Hub.

< Port > = MQTT Broker port. It must be the same *port* is configured in the Broker to be listening. Default value is 8883.

< ClientId > = Device Id configured in Azure IoT Hub.

< Username > = Username defined on MQTT Broker. Is a concatenation of Host Name and Client Id. Syntax: `<URL>/<ClientId>`.

< Password > = Password defined on MQTT Broker. It is the SAS Token defined for the Device Id

< SslProtocol > = IoT Hub uses Transport Layer Security (TLS) to secure connections from IoT devices and services. Three versions of the TLS protocol are currently supported, namely versions 1.0, 1.1, and 1.2.

TLS 1.0 and 1.1 are considered legacy and are planned for deprecation. For more information, see [Deprecating TLS 1.0 and 1.1 for IoT Hub](#). To avoid future issues, use TLS 1.2 as the only TLS version when connecting to IoT Hub.

[**X509Certificate**], *optional* = Path of X509 Certificate using TLS v1.0. This must be the complete path of X509 certificate in the client computer. The certificate must be installed in the computer.

Note: One of the easiest ways to install the certificate on client computer is to use the wizard to import the certificates is through the "Internet Options" to import the certificate in the "Trusted Root Certification Authorities". You need to import the certificate in DER format.

< **Payload Format** > = Format of the message sent to the Broker. By default, the payload is set to follow JSON/SparkplugB format.

Station Examples

URL = CustomHub-MQTT.azure-devices.net

Port = 8883

Client ID = MQTT_Device

Username = CustomHub-MQTT.azure-devices.net/ MQTT_Device

Password = SharedAccessSignature sr=CustomHub-MQTT.azure-devices.net&sig=yglT%2FG8fIFAXhUwGXBlm6WKDK5IhbmVHkUs4atsbXFU%3D&skn=iothubowner&se=1612031715

Ssl Protocol = <TLS>

X059 Certificate =

Payload Format = JSON/SparkplugB

For more information on how to set up a connection with Azure IoT Hub and find the required parameters see the [append](#).

Section 4 – Point Configuration

The MQTT and Azure protocol supports **Write Commands only**.

Address

The Address syntax is.

[Payload] : <Group > ; <Node> ; <Device> ;

Where:

[Payload] = It is a fixed parameter, that assists the user in understanding that the address configuration (*group_id*, *edge_node_id* and *device_id*) are the user-defined parameters in the Payload structure (similar to MQTT + SparkplugB protocol).

< **Group** > = Provides a logical grouping of Edge Node's.

< **Node** > identifies ID of Edge Node.

< **DeviceId** > identifies ID of Device from Edge Node. This field can be empty while accessing main Edge Node.

E.g.: Payload:GroupID;NodeID;DeviceID;
Payload:AzureGroup;Factory1;Motor

Section 5 – Troubleshoot

The status of the driver execution can be observed through the diagnostic tools, which are:

- Trace window
- Property Watch
- Module Information

Status value of 0 (zero) means communication success. Negative values indicate internal driver error.

Section 6 – Common Errors

This section details some errors you might see in your connection to Azure IoT Hub via MQTT protocol and its possible causes.

Error: Exception connecting to the broker

If you see this message on the Trace Window Logs (with modules Debug and Devices enabled) check the port number parameter in *Devices - Nodes - PrimaryStation*.

Device.Node.Status = -1

If you get this error code in `Devices.Node.<NodeName>.Status` check if the URL defined in *Devices - Nodes - PrimaryStation* is correct.

Error Connecting to Broker. Status: 5

If you see this message on the Trace Window Logs (with modules Debug and Devices enabled) check the following parameters in *Devices-Nodes-PrimaryStation*:

- . Username
- . ClientID

MQTT driver connection lost: System.EventArgs

If you see this message on the Trace Window Logs (with modules **Debug** and **Devices** enabled) check the **QoS** setting in *Devices - Nodes - PrimaryStation*. As stated in a previous section, IoT Hub will automatically disconnect your client if the Qos value is 2 (Exactly Once).

Another reason might be that someone else tried to connect to your IoT Device using the same ClientID. IoT Hub only supports one active MQTT connection per device. Any new MQTT connection on behalf of the same device ID causes IoT Hub to drop the existing connection.

Revision History

Revision	Version	Description	Date
A	1.0.0.0	Initial Revision	January 2021
B	1.0.0.2	Revision Update	March 2021

Append – How to Set up Azure IoT Hub

Below is listed the requirements for a successful configuration.

- Microsoft Azure IoT Hub Account;
- Visual Studio Code with an Azure IoT Hub extension.

Configuration Settings

In this section we will describe how to configure each program.

- Visual Studio Code

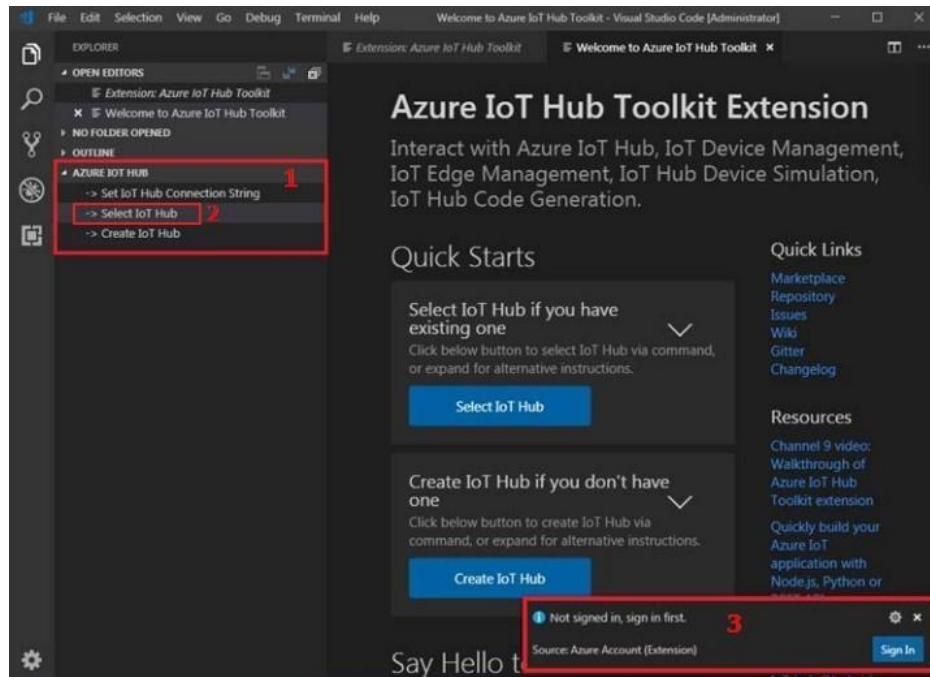
Visual Studio Code is an open-source, streamlined code editor with support for development operations like debugging, task running, and version control. It can be downloaded [here](#).

To download the extension that allows interaction with Azure IoT Hub and IoT Device Management click [here](#).

Note: This guide assumes that an IoT Hub already exists within the Azure Portal account.

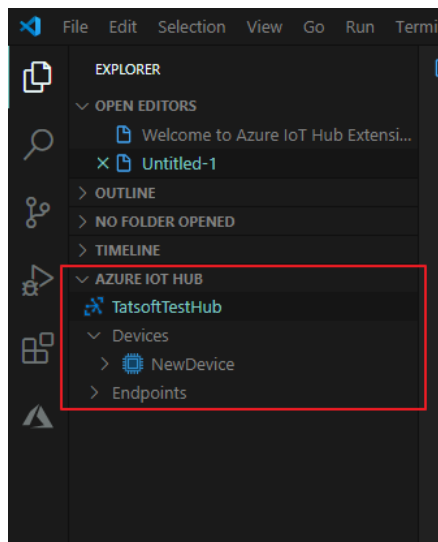
After installing the extension, open VSCode application. In Explorer of VS Code, click "Azure IoT Hub" tab in the bottom left corner (1) and Click "Select IoT Hub" (2) in context menu.

If you have not signed in to Azure, a pop-up will show in the bottom right corner to let you sign into Azure (3).



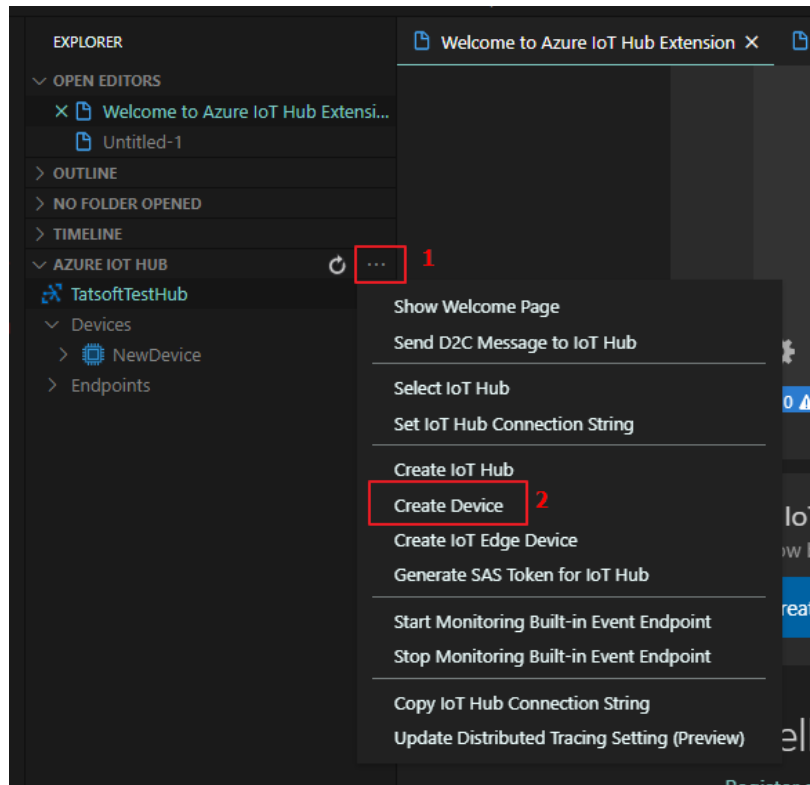
Azure IoT Extension Setup.

After you sign in, your Azure Subscription list will be shown, then select Azure Subscription and IoT Hub. The devices and endpoints list will be shown in "Azure IoT Hub" tab in a few seconds.



Devices in IoT Hub.

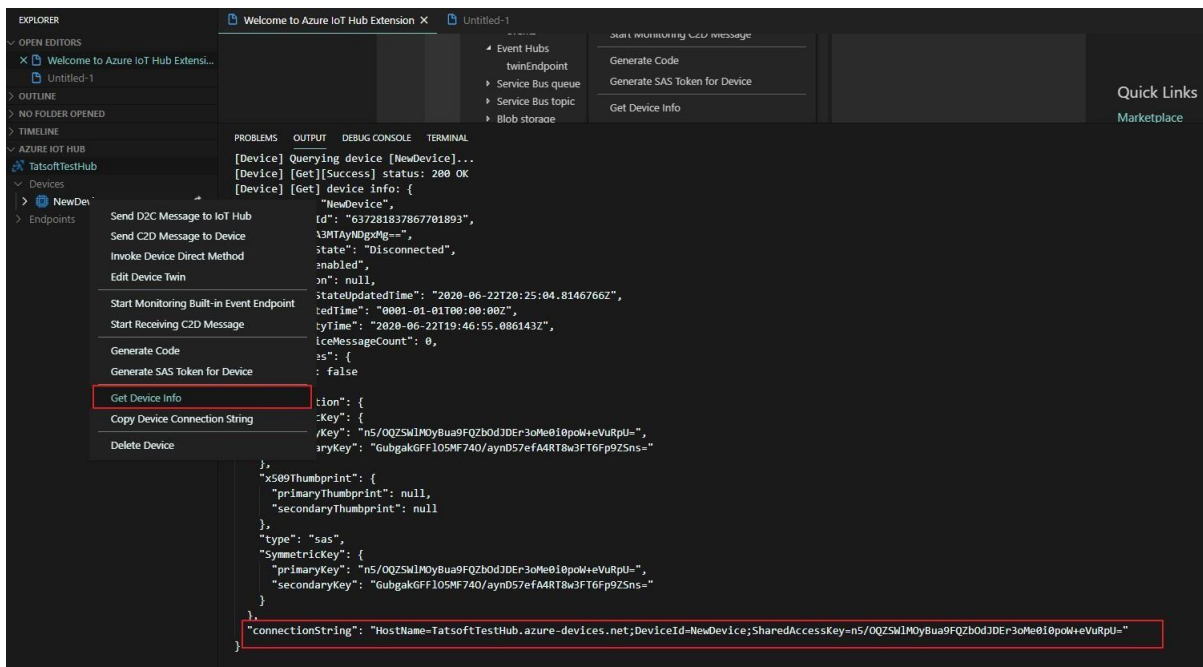
New IoT Devices can be created through this extension. From the context menu (1), click Create Device (2) and enter a Device ID for the new IoT device.



Device creation.

Now that we have our Devices created, we will need to get our Connection String and SAS Token.

Right-Click on the Device and select "Get Device Info". You should see some information displayed at the Output Window.

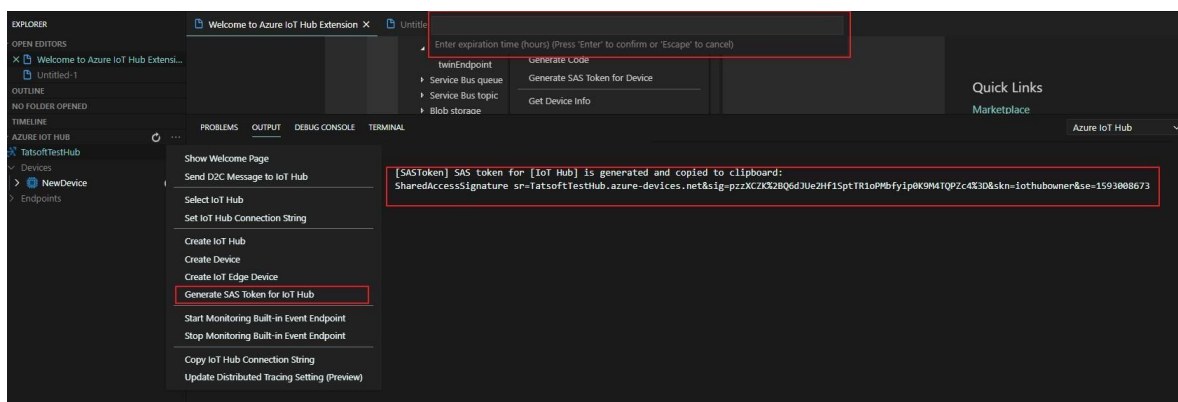


Connection String.

Review the Connection String created for the device and record the following pieces of information from this string:

HostName = TatsoftTestHub.azure-devices.net
DeviceId = NewDevice

Again, right-click the device and select "Generate SAS token for device" and enter the expiration time. You should see some information displayed at the Output Window.

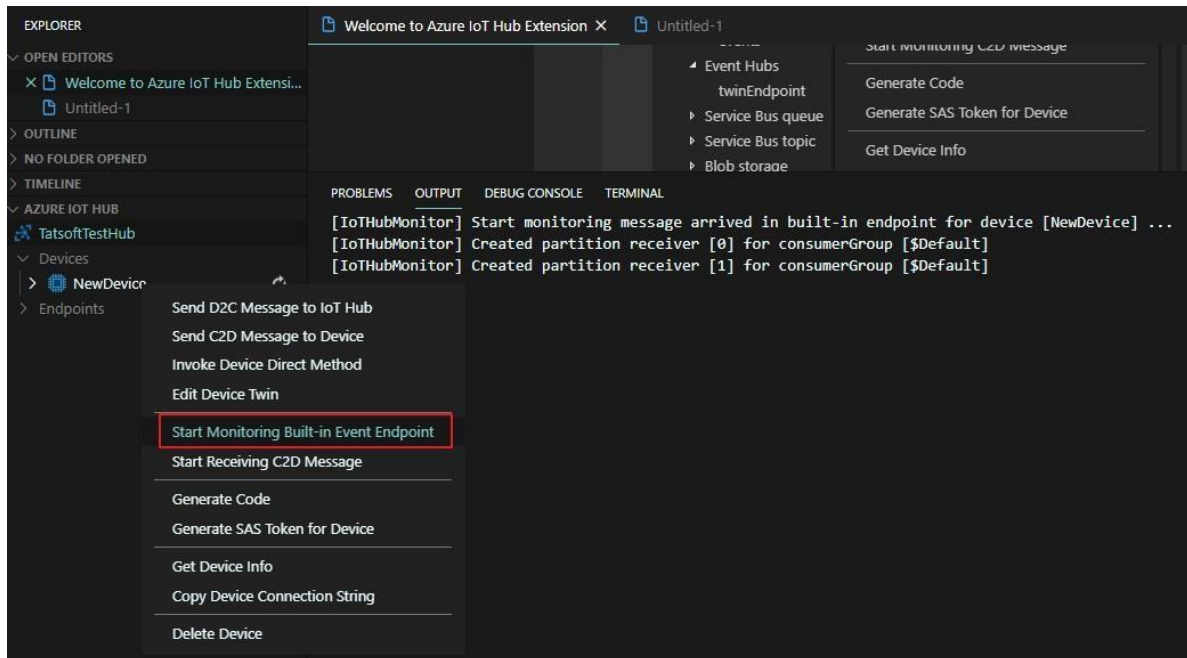


SAS Token.

Record the following pieces of information from this string:

Password = SharedAccessSignature sr=TatsoftTestHub.azure-devices.net&sig=pzzXCZK%2BQ6dJue2Hf1SptTR1oPMBfyip0K9M4TQPZc4%3D&skn=iothubowner&se=1593008673

To verify that data is flowing from MQTT Driver to the cloud-based Azure IoT Hub, right-click the device entry and select "Start Monitoring Built-In Event Endpoint".



```
[IoTHubMonitor] [5:04:19 PM] Message received from [MyDevice]:
{
  "Topic": {
    "Namespace": "spBv1.0",
    "GroupId": "GroupID",
    "EdgeNodeId": "NodeID",
    "DeviceId": "DeviceID"
  },
  "Payload": {
    "timestamp": "1614888259453",
    "metric": [
      {
        "name": "FS_Simulator_Sine",
        "timestamp": "1614888259142",
        "datatype": 3,
        "properties": {
          "keys": [
            "Quality"
          ],
          "values": [
            {
              "type": 12,
              "stringValue": "{ \"type\": 3, \"intValue\": 192 }"
            }
          ]
        },
        "intValue": 12
      }
    ],
    "seq": "1"
  }
}
```

Messages received in VSCode

Info 1	Info 2	Message
MQTTAzure		Device OnStart - Finalizing initialization
MQTTAzure		Device Module initialized Successfully
MQTTAzure		Connected successfully to URL: CustomHub-MQTT.azure-devices.net 8883
MQTTAzure		Publishing: [{"Topic": {"Namespace": "spBv1.0", "Groupid": "GroupID", "EdgeNodeid": "NodeID", "Deviceid": "DeviceID"}, "Payload": [{"timestamp": "1614888242943", "metric": [{"name": "FS_Simulator_Sine", "timestamp": "1614888242380", "datatype": 3, "properties": {"keys": ["Quality"], "values": [{"type": 3, "intValue": 64}, {"type": 0}]}]}]}]
MQTTAzure		Topic 'devices/MyDevice/messages/events/' was published successfully [{"Topic": {"Namespace": "spBv1.0", "Groupid": "GroupID", "EdgeNodeid": "NodeID", "Deviceid": "DeviceID"}, "Payload": [{"timestamp": "1614888242943", "metric": [{"name": "FS_Simulator_Sine", "timestamp": "1614888242380", "datatype": 3, "properties": {"keys": ["Quality"], "values": [{"type": 3, "intValue": 64}, {"type": 0}]}]}]}]
MQTTAZURE1	ok	ID: 5 - Command Success (ignore) (Master) GroupID:NodeID:DeviceID Value: 12
MQTTAZURE1	ok	ID: 6 - Command Success (ignore) (Master) GroupID:NodeID:DeviceID Value: 12
MQTTAzure		Topic 'devices/MyDevice/messages/events/' was published successfully {"Topic":{"Namespace":"spBv1.0","Groupid":"GroupID","EdgeNodeid":"NodeID","Deviceid":"DeviceID"},"Payload":[{"timestamp":"1614888259453","metric":[{"name":"FS_Simulator_Sine","timestamp":"1614888259142","datatype":3,"properties":{"keys":["Quality"],"values":[{"type":12,"stringValue":{"type":3,"intValue":192}}]}]}]},"intValue":12},"seq":"1"}]

Trace Window logs.

For the correct topic, see the Address section of this document.