Azure IoT Hub Module Quick Guide

How to connect to Microsoft Azure IOT Hub

There are 2 ways for connecting vNode to Microsoft Azure IOT data Hub:

1. Connecting using username and password.
2. Connecting via X.509 Certificates.

For both scenarios it will needed to download and install Device Explorer using the following URL:

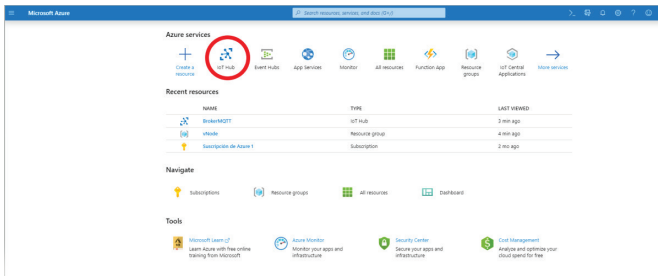
https://github.com/Azure/azure-iot-sdk-csharp/tree/master/tools/DeviceExplorer.

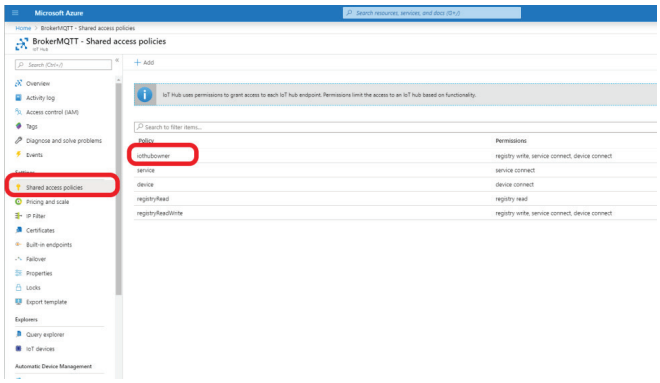
A Connecting using username and password:

The purpose of this quick guide is to show you how to configurate MQTTClient under vNode using a generic broker.

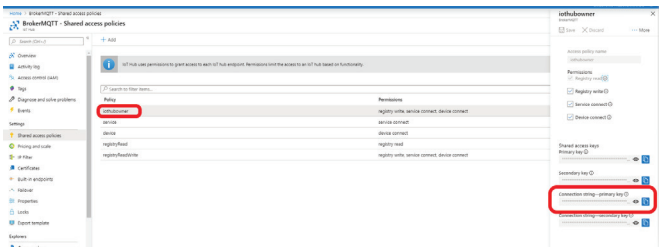
In this case, we have chosen to pick Mosquitto, an open-source and reliable available on the web for free.

1 Enter Microsoft Azure Portal and select or create a new IoT-Hub. Once it's done, click "Shared access policies" then "iothubowner" and copy "Connection string—primary key

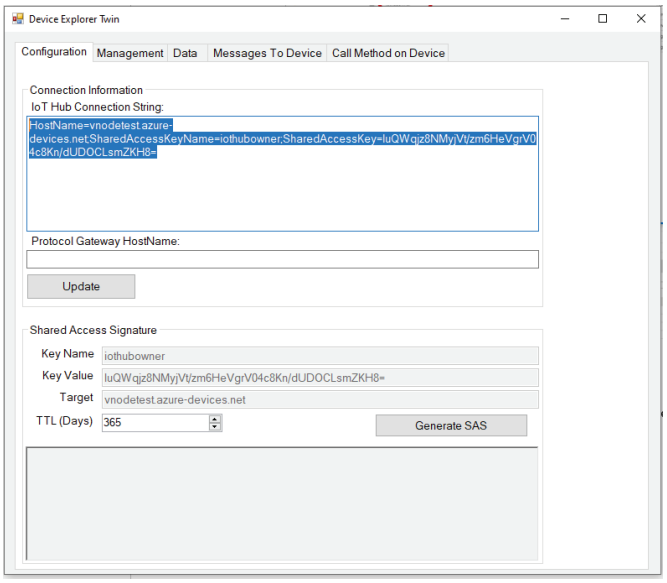




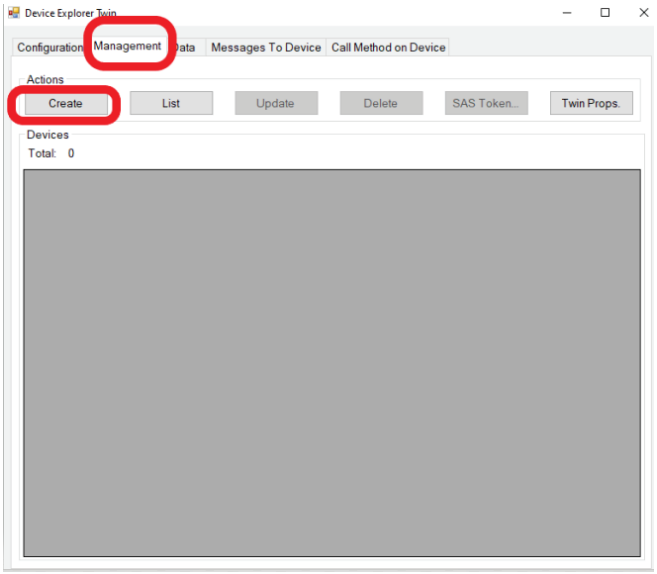
2 Copy the Primary Key connection string

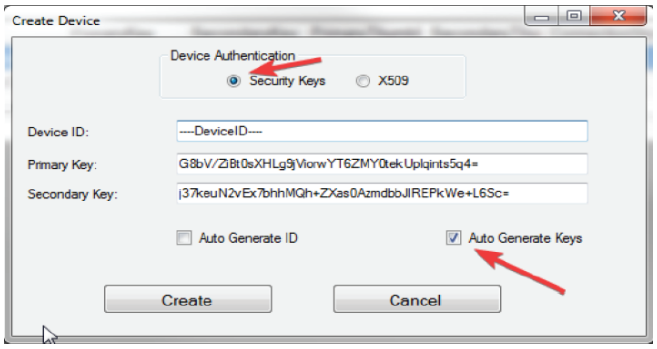


3 Start Device Explorer, and paste the connection string in the "IoT Hub Connection String" text area, and afterwards click Update

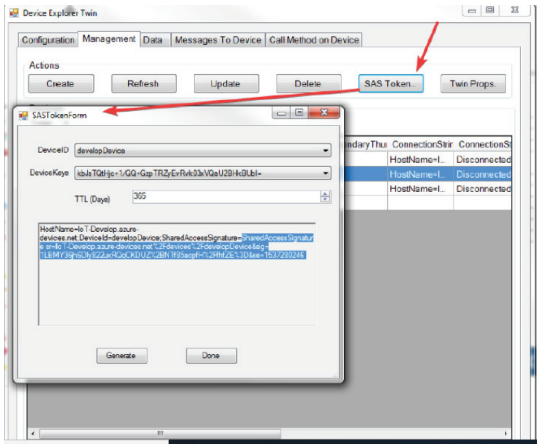


4 Click on management, then “Create”, select “Security Keys” as the authentication method and introduce a Device ID, leaving the “Auto Generate Keys” checkbox checked.





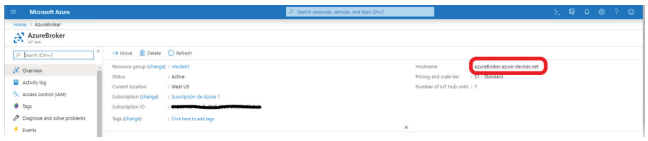
5 Next, click “SAS Token”, select the device you want to connect to, and specify a Time to Live (TTL) for the token. Once created, copy everything after “SharedAccessSignature = “



6 Finally, configure MqttClient with the following options:

URL: `ssl://HostName:8883` or `mqtts://HostName:8883`

- ClientId: **deviceId** (The broker uses the **ClientID** to identify the **client** and the current state of the **client**. Therefore, this **ID** should be unique per **client** and broker.)
- Hostname: IP address or URL from the MQTT Broker



- Username: **HostName/deviceId**
- Password: SAS Token (see previous screenshot)
- Port: 8883
- Protocol: MQTTS

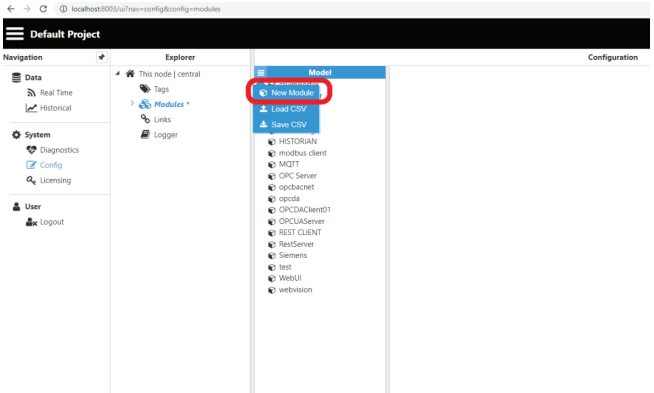
- Topic: devices/**deviceId**/messages/events/**properties** **(it could be any topic, properly from the broker or even created in the client too.)**

Properties are a list of key-value pair in URL encoded format, for example:

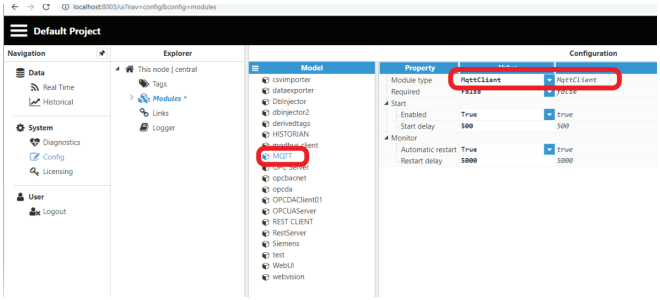
`devices/TestDevice/messages/events/sensor=Temperature&unit=Celsius`

vNode MQTT Client configuration

1 Under vNode configuration, create New Module and give it a name.



2 Select MQTT client module from the Drop-Down list

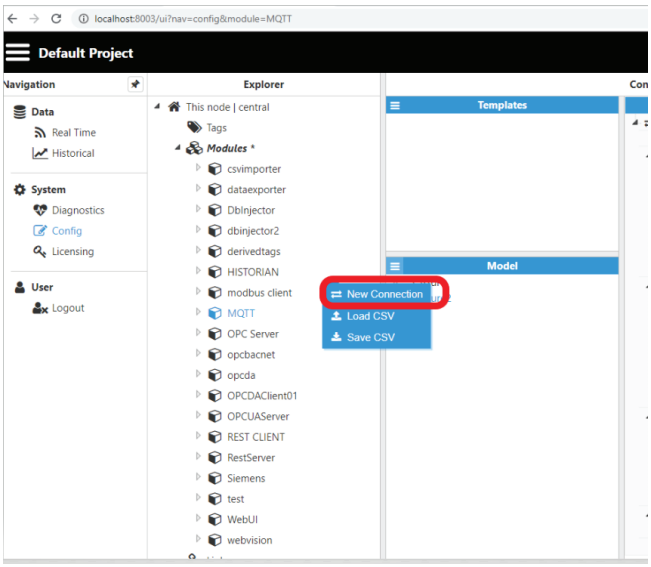


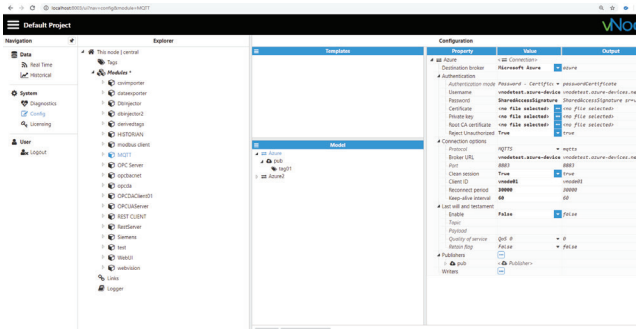
3 Click on Save Bottom.

4 Once the module is created, it will appear under Modules pane.

5 Select the module created and create a Model by selecting “New Connection”

6 Provide a name for the new connection model.





7 Once the model is created, select the Destination Broker from the Drop Down, and select Microsoft Azure.

8 Fill up the communication parameters provided by the broker.
Authentication
Username: Broker URL / deviceID
Password: created with the device Explorer on part A of this document.

Property	Value	Output
Azure2	<⇄ Connection>	
Destination broker	Microsoft Azure	azure
Authentication		
Authentication mode	Password - Certi	passwordCertificate
Username	AzureBroker.azure-(	AzureBroker.azure-devices.ne
Password	SharedAccessSignatu	SharedAccessSignature sr=Azu
Certificate	<no file selecte	<no file selected>
Private key	<no file selecte	<no file selected>
Root CA certificate	<no file selecte	<no file selected>
Reject Unauthorized	True	true

Connection Options

Broker URL: provided by Azure IoT Hub

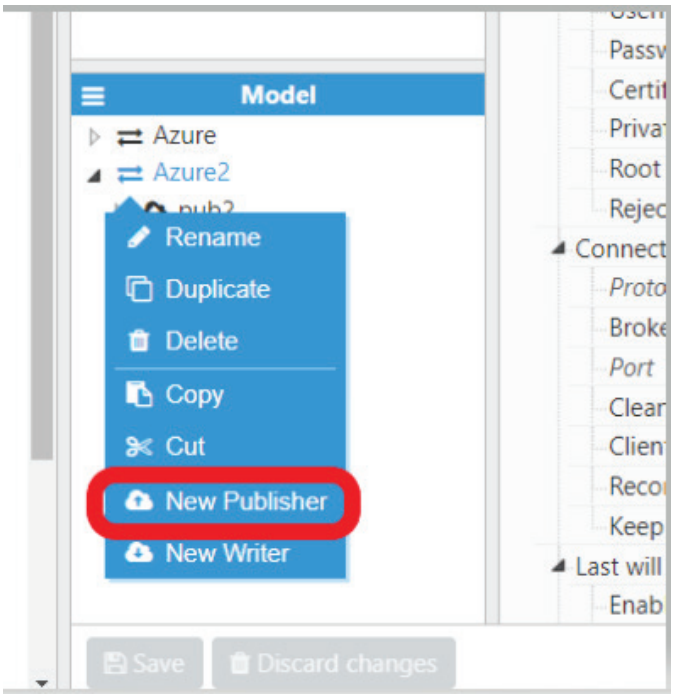
Connection options		
Protocol	MQTTS	mqtts
Broker URL	AzureBroker.azure-(	AzureBroker.azure-devices.n
Port	8883	8883
Clean session	True	true
Client ID	vnode01	vnode01
Reconnect period	30000	30000
Keep-alive interval	60	60

Client ID: MQTT Client identifier
Port: 8883

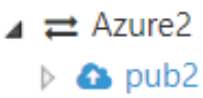
9 Depending of what you need to do, you can decide if vNode will act as a Publisher for sending data to the Azure IoT, o writer if you want to read data from Azure

For this case, we`ll going to send data to Azure, so let’s create a Publisher.

10 Right click over the model created, and select: New Publisher



11 Give a name for the publisher

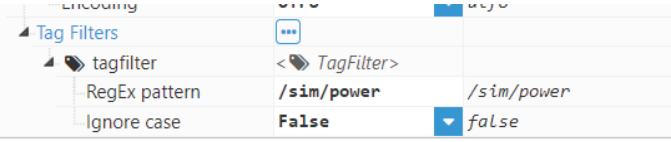


12 Type the topic in which you want, to write data to the broker.
(the topic can be whatever you want or you can use one of the predefined in the broker)

Property	Value	Output
pub2	<Publisher>	
Push interval	60000	60000
Topic	devices/vnode01/me	devices/vnode01/messages/ev
QoS	QoS 0	0
Retain flag	False	false

13 According to payload options, the default parameters match with Azure IoT Hub.

14 The Tag filter is where you select the tags will be sent to the broker
There's a Regex Pattern, to type the tag path from vnode: /group name/tag name

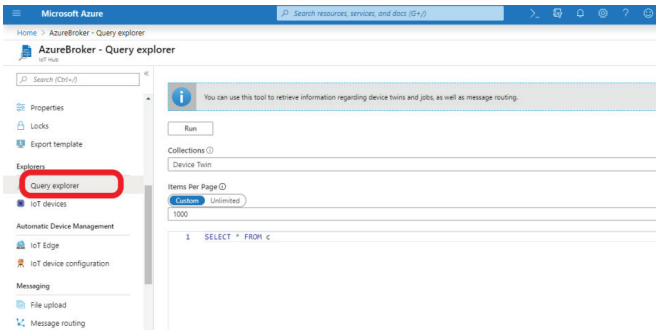


Tag Filters		
tagfilter	<TagFilter>	
RegEx pattern	/sim/power	/sim/power
Ignore case	False	false

15 Once everything has done, click on save bottom.

How to know if we are connected to Azure IoT Hub?

- Go to you Azure Panel, under your broker
- Select Query Explorer tab at your left side panel



- Hit the RUN bottom

Also you can check on vNode logs in which the communication will display like this:

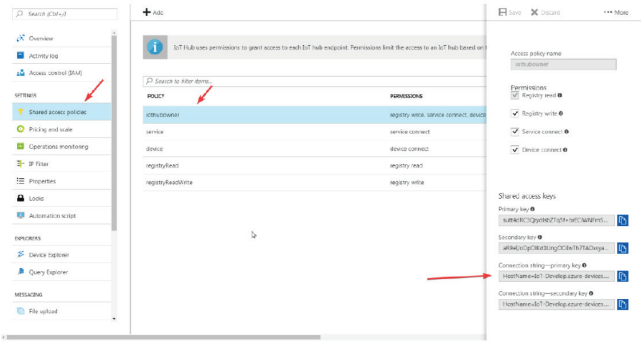
20:55:30.120 | INFO | Connection:Azure2 -> Connected to broker
20:55:30.120 | DEBUG | Connection:Azure2 -> {"cmd":"connack","retain":false,"qos":0,"dup":false,"length":2,"topic":null,"payload":null,"sessionPresent":false,"returnCode":0}
20:55:30.168 | TRACE | Publisher:Azure/pub -> Tag updated /sim/power -> {"value":53729,"quality":192,"ts":"2019-12-11T20:55:29.493Z"}
20:55:30.168 | TRACE | Publisher:Azure2/pub2 -> Tag updated /sim/power -> {"value":53729,"quality":192,"ts":"2019-12-11T20:55:29.493Z"}
20:55:30.345 | DEBUG | Publisher:Azure2/pub2 -> 0 packets in memory, 9 events in buffer
20:55:30.345 | DEBUG | Publisher:Azure2/pub2 -> Reading S&F directory
20:55:30.345 | DEBUG | Publisher:Azure2/pub2 -> 90 files in directory
20:55:30.346 | DEBUG | Publisher:Azure2/pub2 -> Reading file 1576092219118.dat
20:55:30.347 | DEBUG | Publisher:Azure2/pub2 -> File read OK
20:55:30.347 | DEBUG | Connection:Azure2 -> Publishing packet to devices/vnode01/messages/events

B Connecting via X.509 Certificates.

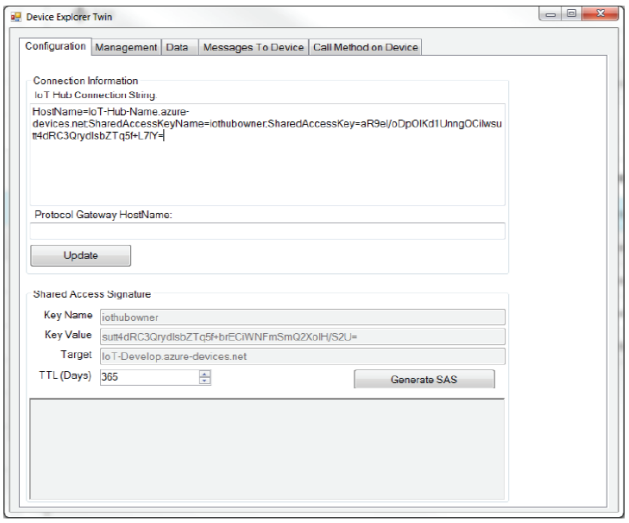
1 Download Device Explorer from using the following URL: https://github.com/Azure/azureiot-sdk-csharp/tree/master/tools/DeviceExplorer

2 Download OpenSSL from https://www.npcglib.org/~stathis/blog/precompiled-openssl/

3 Enter Microsoft Azure Portal and select or create a new IoT-Hub. Once it's done, click "Shared access policies" then "iothubowner" and copy "Connection string—primary key"



4 4. Start Device Explorer and paste the connection string in the "IoT Hub Connection String" text area, and afterwards click Update.



5 Generate the necessary certificate and private key or use the following commands to create one:

```
openssl req -newkey rsa:2048 -nodes -keyout private.key -x509 -days
1000 -out certificate.crt
```

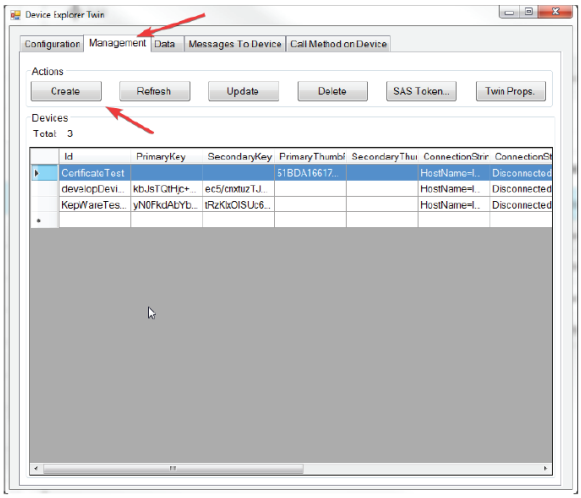
6 Obtain the certificate thumbprint by running the following command:

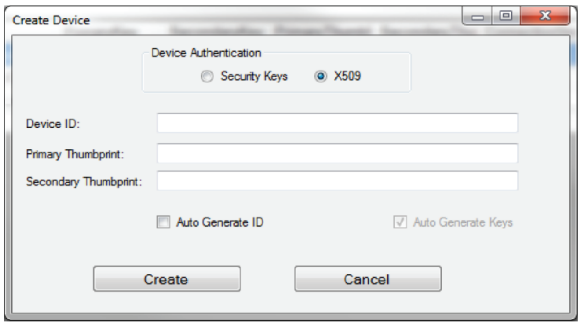
```
openssl x509 -in certificate.crt -fingerprint -noout >
fingerprint.txt
```

7 The thumbprint is generated in a textfile called "fingerprint.txt". The format of the thumbprint is: xx:xx:xx:xx:...

8 Remove all the ':' and copy the resulting thumbprint.

9 In device explorer, click on management, then "Create", select "X509" as the authentication method. Introduce a device ID and paste the certificate thumbprint as the primary thumbprint.





10 Afterwards, configure your MQTT client with the following options:

- URL: `ssl://HostName:8883` or `mqtts://HostName:8883`
- ClientId: **`deviceId`**
- Username: **`HostName/deviceId`**
- Password: `HostName=HostName;DeviceID=deviceId;x509=true`
- Topic: `devices/deviceId/messages/events/properties`

Properties are a list of key-value pair in URL encoded format, for example:
`devices/TestDevice/messages/events/sensor=Temperature&unit=celsius`

This topic sends a message to the TestDevice device and has two properties, sensor and unit, which are equal to Temperature and celsius respectively. This can be used to route messages to other end points via Routes.

Example of password:
`"HostName=test-hub.azure-devices.net;DeviceID=TestDevice;x509=true"`

Contact us:

Main office:
United States,
1549 NE 123St,
North Miami,
ZIP 33161.
+1 (754) 755-0009

Regional offices:
United Kingdom:
(+44) 161 660 3241

Email:
info@vnodeautomation.com.
support@vnodeautomation.com
sales@vnodeautomation.com,
saleseurope@vnodeautomation.com
supporteurope@vnodeautomation.com

Website:
vnodeautomation.com