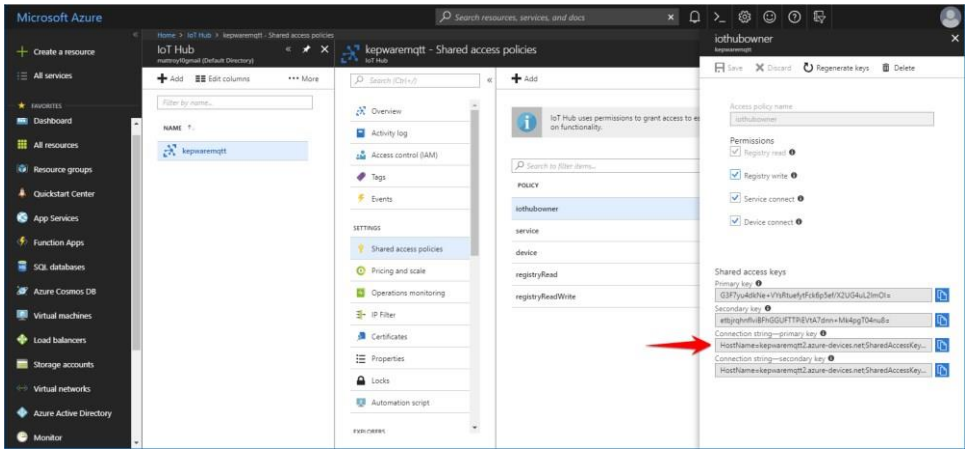
Technical Note

IoT Gateway MQTT Client Agent and Microsoft Azure IoT and ThingWorx Kepware Edge

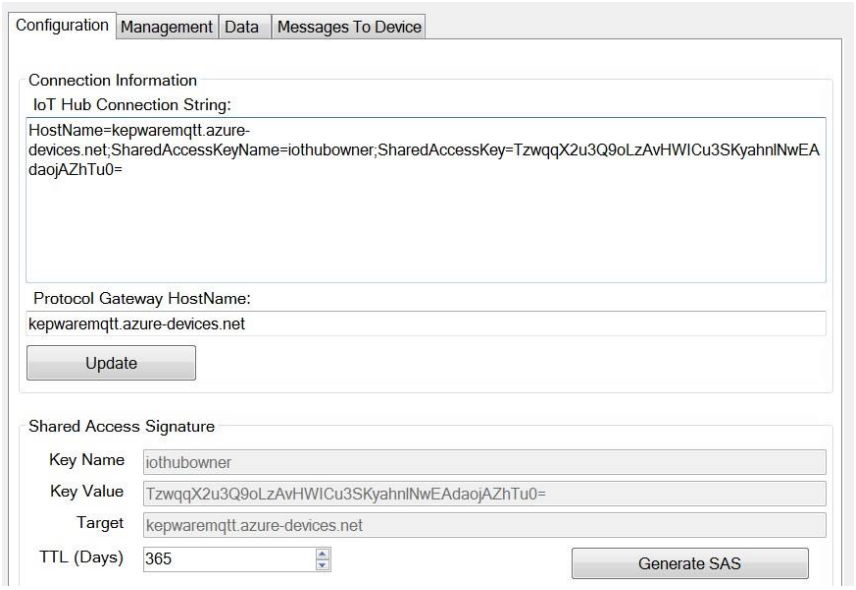
This document facilitates connecting an MQTT client to a Microsoft Azure IoT Hub with ThingWorx Kepware Edge. To read more about the Azure IoT Hub's MQTT support, refer to the following documentation from Microsoft for more information: https://docs.microsoft.com/enus/azure/iot-hub/iot-hub-mqtt-support.

Connect MQTT Client Agent to Azure with an SAS Token

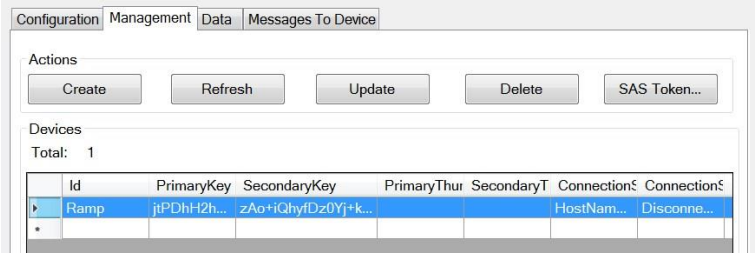
1. Install Microsoft's Device Explorer. Device Explorer is used to configure the IOT Hub device and the SAS token is required to complete the setup. The application is available to download from https://github.com/Azure/azure-iot-sdkcsharp/tree/master/tools/DeviceExplorer.
2. Access a Microsoft Azure instance.
3. Create an IoT Hub in the Azure instance with a unique name and resource group.
4. Click **Shared access policies** under Settings in the IoT Hub and select the appropriate policy. In this example, using the iothubowner policy, a shared access key is generated as well as a connection string. The **Connection string** for iothubowner includes both the shared access key and the hostname URL.



5. Copy the connection string and place it into the **IoT Hub Connection String** field into the Device Explorer application downloaded in a previous step.

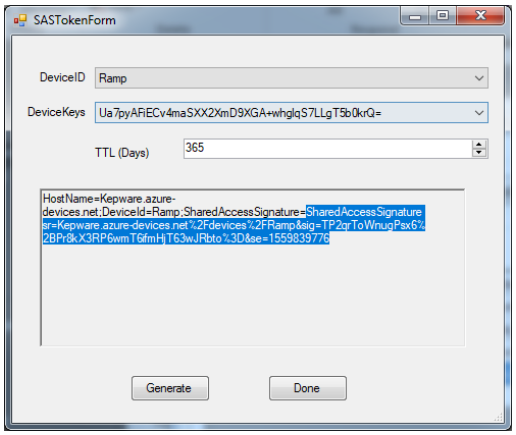


6. In Device Explorer, click **Update**.
7. Create a device by accessing the **Management** tab in the Device Explorer.
8. Click **Create** in the **Actions** area.



9. In the Create Device window, verify that **Security Keys** authentication and **Auto Generate Keys** are selected.
10. Provide a unique name in the Device ID field.
11. Click **Create** to close the creation window.

12. Click the **SAS Token...** button in the Actions area to generate the SAS token in Device Explorer.



13. Copy the part of the string from this dialog starting with everything AFTER "SharedAccessSignature=" into a file for use in a later step.
14. The **SAS Key** copied will be provided as the MQTT Client Password in the server project in a later step.

    Note: Set **TTL (Days)** to a reasonably high number. This controls how long the SAS token is valid.

15. Click **Generate**.
16. In the JSON project file, add or modify an IoT MQTT agent with the following formats for the indicated properties.
17. Edit the MQTT Client URL with the correct <**HostName**>. <**HostName>** = HostName element in the IoT hub connection string: "iot_gateway.MQTT_CLIENT_URL": "ssl://<HostName>:8883".
18. Edit the MQTT Client Topic with the correct <**deviceID**> and <**property bag**>.
    <**deviceID>** = Name of the device created in the Device Explorer.
    <**property bag>** = (optional) Sends each message with additional properties in a url-encoded format. Example: location=abcd&id=12345:
    "iot_gateway.MQTT_CLIENT_TOPIC":
    "devices/<deviceID>/messages/events/<property bag>".

    Refer to the following documentation for more information about the property bag: https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-mqtt-support.

19. Create security credentials. The expected formats are as follows:

    Client ID: <**deviceID>** "iot_gateway.MQTT_CLIENT_CLIENT_ID": "<deviceID>"

    Username: <**HostName**>/<**deviceID**>
    "iot_gateway.MQTT_CLIENT_USERNAME": "<HostName>/<deviceID>"

    Password: <**SAS Key>** where the <SAS Key> = SharedAccessSignature element when the SAS token was created with Device Explorer
    "iot_gateway.MQTT_CLIENT_PASSWORD": "<SAS Key>"

20. Add an IoT item to the MQTT Client Agent.
21. Verify that the MQTT client agent has connected to the Azure IoT Hub by observing the event log has an entry similar to the following:

    { "timestamp": "2019-02-20T18:10:05.066", "event": "Information", "source": "TKEdge\\Runtime", "message": "MQTT agent 'MQTT Client' is connected to broker 'ssl://Edge.azure-devices.net:8883'" }

The following is an example of the MQTT agent JSON blob:

```
"common.ALLTYPES_NAME": "MQTT Client",
"iot_gateway.AGENTTYPES_TYPE": "MQTT Client",
"iot_gateway.AGENTTYPES_ENABLED": true,
"iot_gateway.MQTT_CLIENT_URL": "ssl://<HostName>:8883",
"iot_gateway.MQTT_CLIENT_TOPIC": "devices/<deviceID>/messages/events/<property bag>",
"iot_gateway.MQTT_CLIENT_QOS": 1,
"iot_gateway.AGENTTYPES_RATE_MS": 5000,
"iot_gateway.AGENTTYPES_PUBLISH_FORMAT": 0,
"iot_gateway.AGENTTYPES_MAX_EVENTS": 1000,
"iot_gateway.AGENTTYPES_TIMEOUT_S": 5,
"iot_gateway.AGENTTYPES_MESSAGE_FORMAT": 0,
"iot_gateway.AGENTTYPES_STANDARD_TEMPLATE": "timestamp: |SERVERTIMESTAMP|\r\ndate:
|SERVERDATE|\r\nvalues: |VALUES|\r\n",
"iot_gateway.AGENTTYPES_EXPANSION_OF_VALUES": "id: |TAGNAME|\r\nv: |TAGVALUE|\r\nq:
|TAGQUALITY|\r\nt: |TAGTIMESTAMP|\r\n",
"iot_gateway.AGENTTYPES_ADVANCED_TEMPLATE": "{\r\n \"timestamp\": |SERVERTIMESTAMP|,\r\n
\"values\": [\r\n   |#each VALUES|\r\n    {\"id\": \"|TAGNAME|\", \"v\": |VALUE|, \"q\":
|QUALITY|, \"t\": |TIMESTAMP| } |#unless @last|,|/unless|\r\n   |/each|\r\n  ]\r\n}",
"iot_gateway.MQTT_CLIENT_CLIENT_ID": "<deviceID>",
"iot_gateway.MQTT_CLIENT_USERNAME": "<HostName>/<deviceID>",
"iot_gateway.MQTT_CLIENT_PASSWORD": "<SAS Key>",
"iot_gateway.MQTT_TLS_VERSION": 0,
"iot_gateway.MQTT_CLIENT_CERTIFICATE": false,
"iot_gateway.MQTT_CLIENT_ENABLE_LAST_WILL": false,
"iot_gateway.MQTT_CLIENT_LAST_WILL_TOPIC": "",
"iot_gateway.MQTT_CLIENT_LAST_WILL_MESSAGE": "",
"iot_gateway.MQTT_CLIENT_ENABLE_WRITE_TOPIC": false,
"iot_gateway.MQTT_CLIENT_WRITE_TOPIC": "iotgateway/write",
"iot_items":
```

- *See the included demonstration project for the proper syntax for tags under the MQTT agent.*

