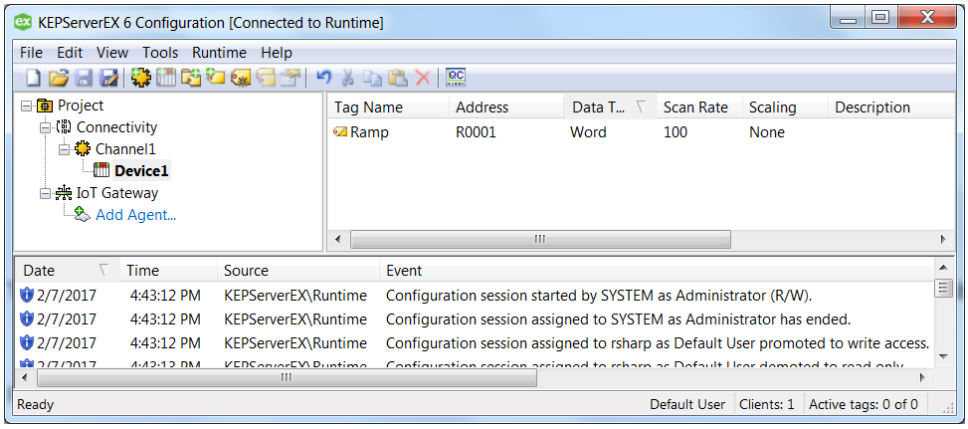
Technical Note

IoT Gateway MQTT Client Agent and Microsoft Azure IoT

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

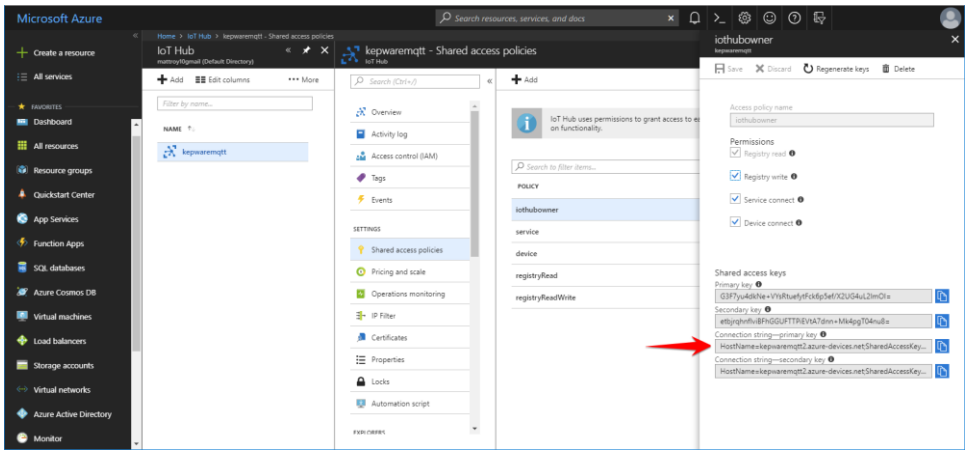
1. Connecting MQTT Client Agent to Azure

1. Open a KEPServerEX® instance with the IoT Gateway advanced plug-in. In this example, one channel and device are configured with the Simulator driver, and there is one tag that ramps up on scan.

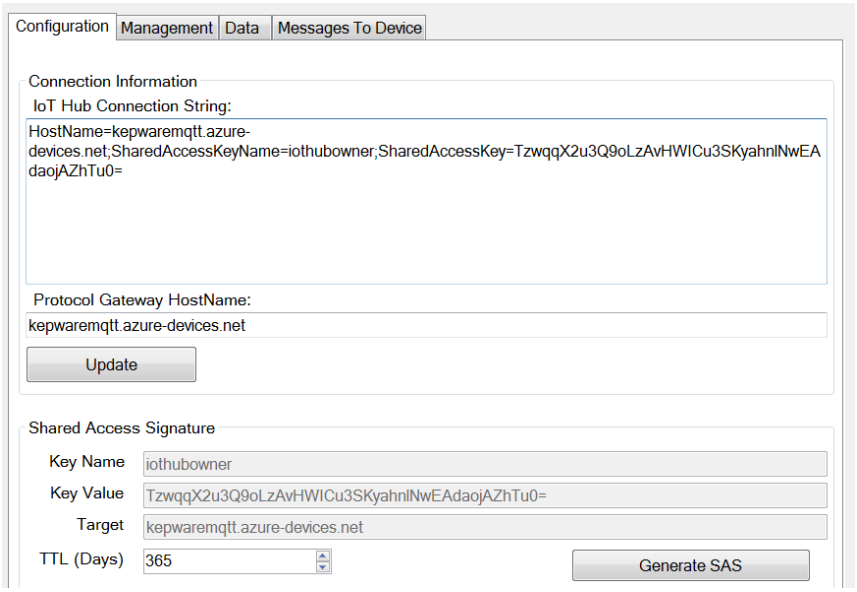


2. Access a Microsoft Azure instance.
3. The third piece of software required is a tool from Microsoft called Device Explorer, which is used to configure the device in the Azure IoT Hub. The application is available to download from https://github.com/Azure/azure-iot-sdk-csharp/tree/master/tools/DeviceExplorer.

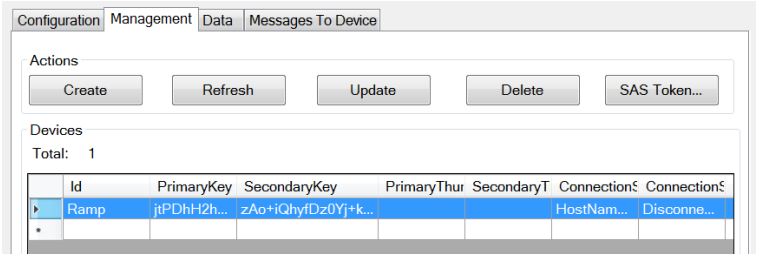
4. Create an IoT Hub in the Azure instance. Assign a unique name and a resource group.
5. Click **Shared access policies** under Settings in the IoT Hub and select the appropriate policy. In this example, use the "iothubowner" policy, a shared access key is generated as well as a Connection string. The **Connection string** for 'iothubowner' includes both the Shared Access Key and the Hostname URL.



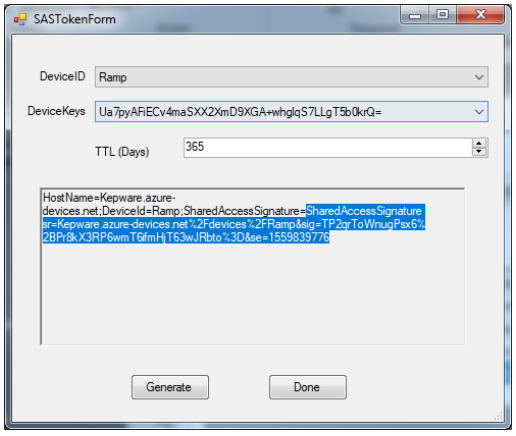
6. Copy the **Connection string** and place it into the "IoT Hub Connection String" field within the Device Explorer application downloaded in a previous step.



7. In Device Explorer, click on "Update", and then create a device by accessing the **Management** tab. Click **Create** and give the device a unique name.



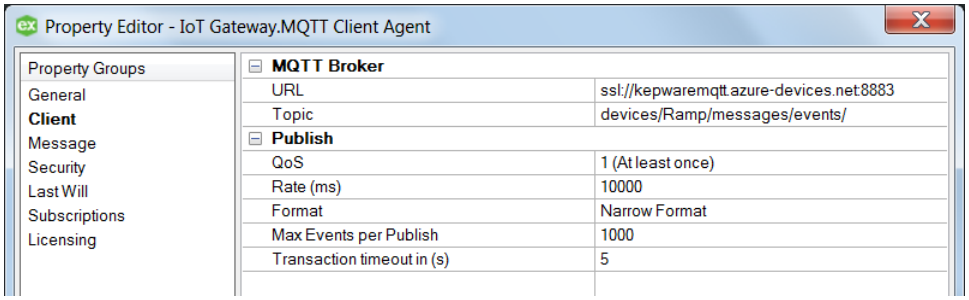
8. Click **SAS Token…** to generate the SAS token in Device Explorer. Part of the string from this dialog needs to be copied (starting with everything AFTER "SharedAccessSignature=") and will be used as the **SAS Key** later. This example will copy the highlighted text.



9. In KEPServerEX, add an IoT MQTT agent. Use the following formats for the indicated properties:
10. URL format: ssl://<**HostName**>:8883
11. Topic format: devices/<**deviceID**>/messages/events/<**property bag**>

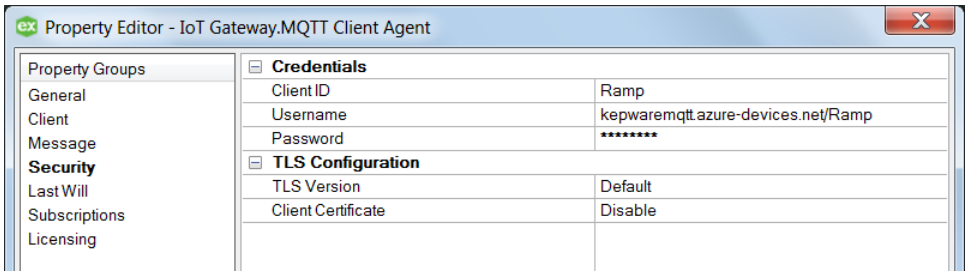
    where:
    <**HostName**> = HostName element in the IoT Hub Connection String
    <**deviceID**> = Name of the device created in the Device Explorer
    <**property bag**> = (optional) Sends each message with additional properties in a url-encoded format. Example: location=abcd&id=12345

    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*.

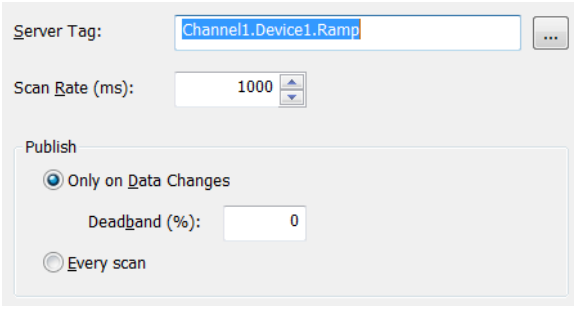


12. Create security credentials. The expected formats are as follows:
13. Client ID: <**deviceID**>
14. Username: <HostName>/<deviceID>
15. Password: <**SAS Key**>

    where:
    <**SAS Key**> = SharedAccessSignature element when the SAS token was created with Device Explorer



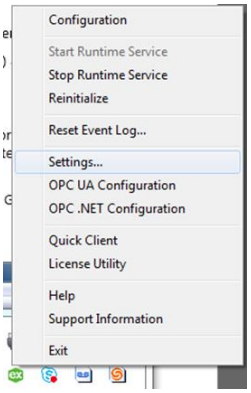
16. Add an IoT item to the MQTT Client Agent.



17. 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:

Date	Time	Level	Source	Event
2/10/2017	2:41:35 PM	Information	KEPServerEX\Runtime	MQTT agent 'Ramp_2_Cloud' is connected to broker 'ssl://kepwaremqtt.azure-devices.net:8883'

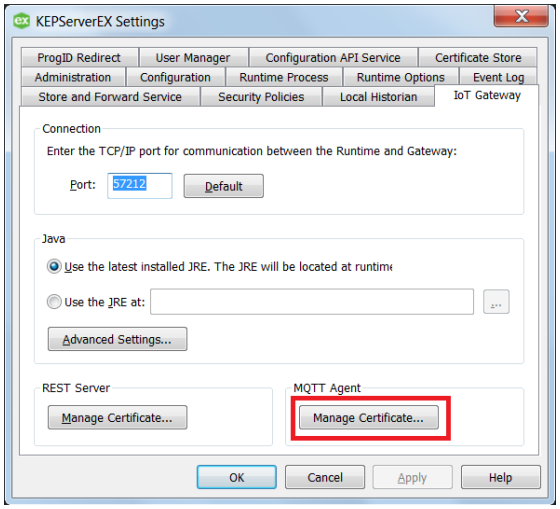
2. Connecting with Self-Signed X.509 Certificates



Microsoft Azure IoT Hub also supports self-signed certificates for authorization, which keeps a certificate and private key on the local machine and stores only a certificate thumbprint on the hub. *For more information on how to connect with self-signed certificates, refer to https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-devguide-security*. The following information describes how to connect IoT Gateway to the hub:

1. Generate self-signed certificates. The output of this command is a new generated certificate and private key:

   ```
   $ openssl req -x509 -sha256 -nodes -days 365 -newkey rsa:2048 -keyout privateKey.key -out certificate.crt
   ```



2. To import the certificate and private key to IoT Gateway, right click KEPServerEX Administration and select **Settings**. Access the IoT Gateway tab, then click **Manage Certificate...**.
3. Click **Import New Certificate...** and browse to the certificate created (certificate.crt).
4. Once the certificate is imported, the Import dialogue box immediately reopens and prompts to import the private key (privateKey.key).
5. Once the private key is imported, you'll receive a popup requesting a private key password.

   Note: The files required for this step depend on the format and contents of the file(s) being imported. For example, if a PFX file is selected, that contains both

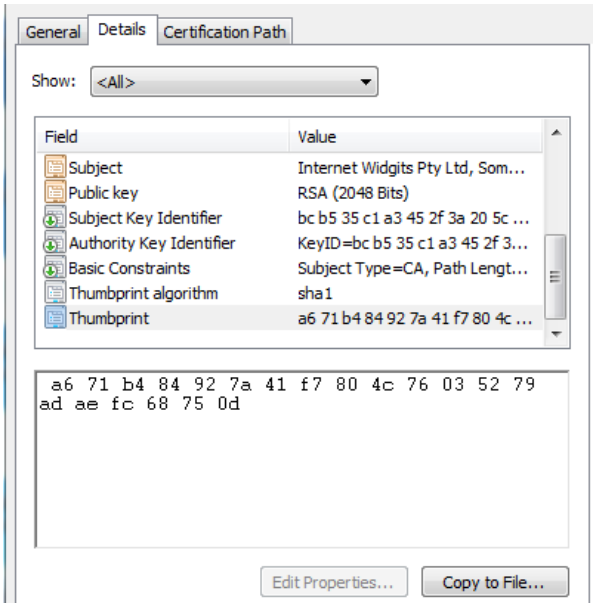
the certificate and private key, no additional files are required. The OpenSSL creation process used in this example creates both a xxx.crt and xxx.key file and both need to be imported independently *(see steps 4-6 above)*.

6. Obtain the certificate thumbprint by doing the following:

   a. Click **View Certificate**.

   b. Access the **Details** tab, and from the drop-down menu, select **<All>**.

   c. Select the thumbprint and copy the string.



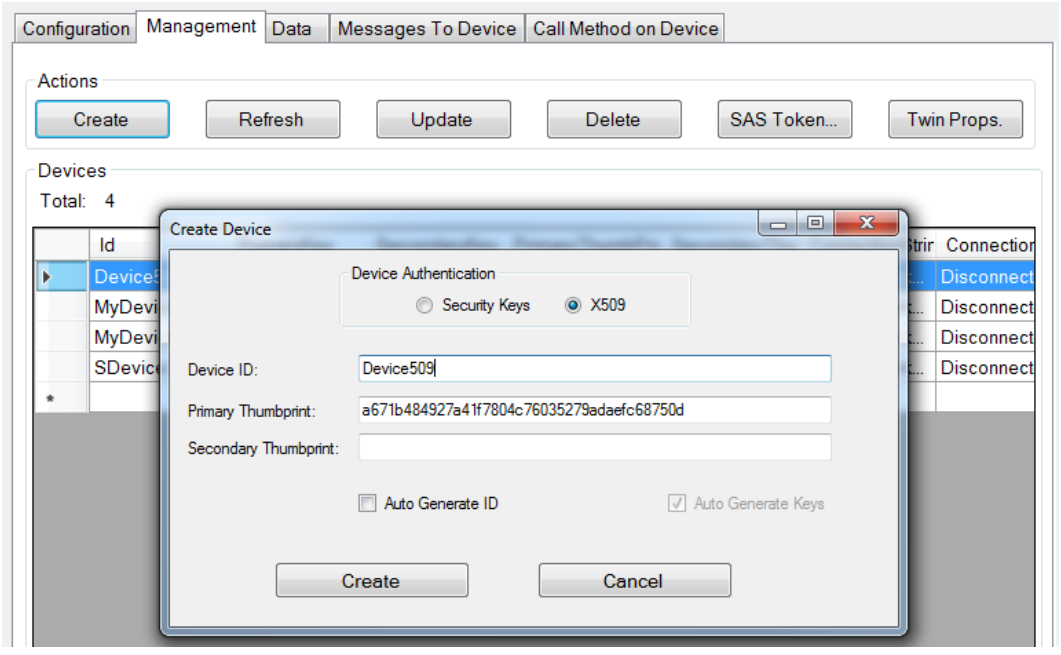
7. Create device using Device Explorer.

   a. Open Device Explorer, access the **Management** tab, and select **Create**.

   b. Select the **X509** option.

   c. Enter **Device ID** and paste the string into the **Primary Thumbprint** field.

   d. Click **Create**.



8. Create new MQTT agent. The following property groups should be configured per the guidelines below:

   a. Client

      i. URL: ssl:// <**HostName**>:8883

      ii. Topic: devices/<**deviceID**>/messages/events/<**property bag**>

b. Security

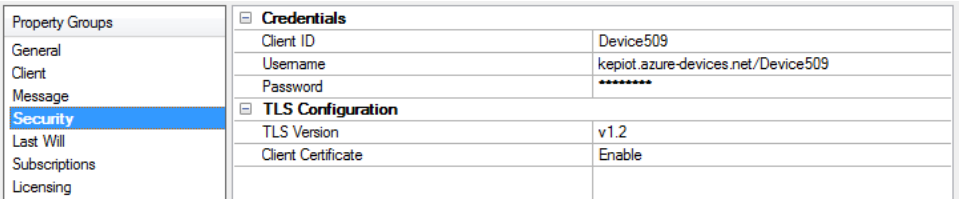
i. Client ID: <**deviceID**>

ii. Username: **<HostName>/<deviceID>**

iii. Password: "HostName= <**HostName**>; **DeviceID** = <**deviceID**>;x509=true"

iv. TLS Version: v1.2

v. Client Certificate: Enable



9. Add tags to MQTT agent. Event log messages will indicate if the agent was successfully connected to the broker.

10. To monitor messages from IoT Gateway, open Device Explorer and access the **Data** tab.

11. Select the device from the drop-down menu and click **Monitor**.