

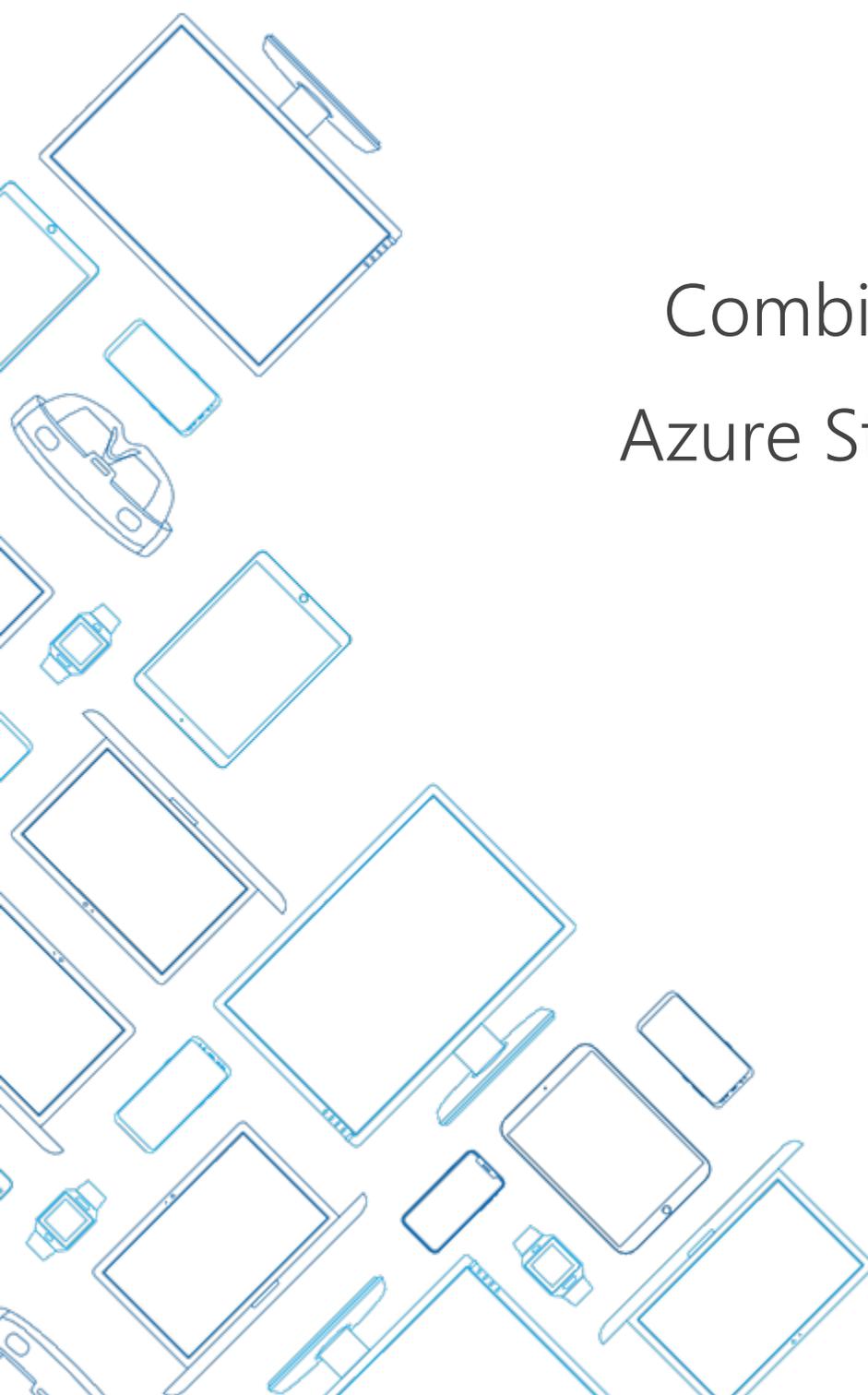


Make the Invisible Visible™



Combining IoTWorX and Azure Stream Analytics on Azure IoT Edge

An ICONICS Whitepaper
December 2021



Combining IoTWorX and ASA on Azure IoT Edge

Contents

- Introduction 4
- Prerequisites 4
- Structure 4
- Overall design 5
- Install IoTWorX on IoT Edge 7
- Create a simple ASA job on IoT Edge 10
- Deploy the ASA job and connect to IoTWorX 12
- View the data 14
- Summary 15

Combining IoTWorX and ASA on Azure IoT Edge

Copyright and Confidentiality

By accessing and using the installation instructions (the “instructions”) you acknowledge and agree, on your behalf and on behalf of the person, entity or other organization on whose behalf you are accessing the instructions, that neither Microsoft or ICONICS, nor any of its service providers, including, without limitation, any system integrator or independent software vendor: (1) makes any representations or warranties of any kind, either express, implied, statutory or otherwise with respect to the instructions, including the accuracy, completeness or usefulness thereof; and (2) shall be liable for damages of any kind, under any legal theory, arising out of or in connection with your election to follow or use, or inability to follow or use, the instructions, including any direct, indirect, incidental, special, punitive or consequential damages, or for loss of use, loss of profits, loss of data, loss of business, or loss of privacy or security, even if foreseeable, arising out of or in connection with your election to follow or use, or inability to follow or use, the instructions. You further acknowledge and agree that your use of the instructions, whether directly or indirectly, is at your own risk and that you expressly assume all risk in connection with your use of the instructions. If you do not agree to the foregoing, you may not access or use the instructions.

Copyright © 2021, Microsoft Corporation and ICONICS, Inc. All rights reserved.

Authors

- Spyros Sakellariadis, Global Director, IoT Business Development, ICONICS, Inc.
- Pavel Stahl, Software Architect, ICONICS, Inc.
- Richard Henderson, US QA Manager, ICONICS, Inc.
- Roberto Vercelli, Director of Engineering, ICONICS, Inc.
- Gordon Smith, Senior IoT Technical Specialist, Microsoft Corporation

Combining IoTWorX and ASA on Azure IoT Edge

Introduction

Microsoft's [Azure IoT Edge](#) supports running cloud-native workloads—such as AI, Azure services, or custom business applications – on IoT devices. Many ISVs have created powerful IoT Edge applications, and Microsoft has shown examples of using tools such as Azure Stream Analytics on IoT Edge. These are all excellent examples of siloed applications or modules running on the edge communicating with applications running in the Microsoft Azure cloud. Missing, as of the date of this paper, are examples of integrating production grade applications on IoT Edge communicating with each other to produce even more powerful solutions than any one application can do on its own. In this whitepaper, we will document how to configure one such ISV application, [ICONICS IoTWorX](#), to communicate with another application, Microsoft's [Azure Stream Analytics](#), to bridge this gap.

Prerequisites

To follow the steps in this whitepaper you need the following:

- An on-premises computer. For this whitepaper, we deployed a virtual machine on an existing Intel i7 computer with 8GB RAM, that was being used for other tasks as well.
- A sensor supporting BACnet communications. For this whitepaper, we used a Delta Controls O3 occupancy sensor.
- An Azure subscription.
- ICONICS IoTWorX and GENESIS64 licenses.

Structure

This whitepaper documents how to get IoTWorX to send data to an Azure Stream Analytics (ASA) job on IoT Edge, which will then forward modified data to Azure IoT Hub. The whitepaper presupposes that the reader already has a good working knowledge of both the ICONICS products and with Azure services including Azure Stream Analytics, and it will not include detailed instructions for setting up either of those. However, to follow this whitepaper, the reader does need working instances of both IoTWorX and an Azure Stream Analytics job, so the paper includes links and references to the following documents for those steps:

- [IoTWorX Quick Start](#). Provides instructions for setting up a basic instance of IoTWorX. We will refer to this document as *Quick Start*.
- [Measuring Occupancy with Delta Controls O3 Sense, Azure IoT, and ICONICS](#). Provides information about the occupancy sensor used in this whitepaper. Referred to as *Measuring Occupancy*.

Combining IoTWorX and ASA on Azure IoT Edge

- [Installing IoTWorX on IoT Edge](#). Provides instructions for configuring IoTWorX to access the occupancy sensor. Referred to as *Installing IoTWorX*.
- [Tutorial: Deploy Azure Stream Analytics as an IoT Edge module](#). Provides instructions for setting up a basic instance of ASA on IoT Edge. Referred to as *Tutorial*.

In this whitepaper, we will show you how to:

- ✓ Configure ASA to read the data sent by IoTWorX to the IoT Edge Hub.
- ✓ Configure IoT Hub routes to send data from IoTWorX to ASA, and from ASA to IoT Hub.

While the steps in the documents listed above, and those in the current whitepaper, are generally applicable for most users, results may vary depending upon a user's software and hardware systems.

Overall design

In the *Quick Start*, you will find instructions for how to provision IoTWorX on IoT Edge and configure it to send data to IoT Hub. The data flow is as follows:

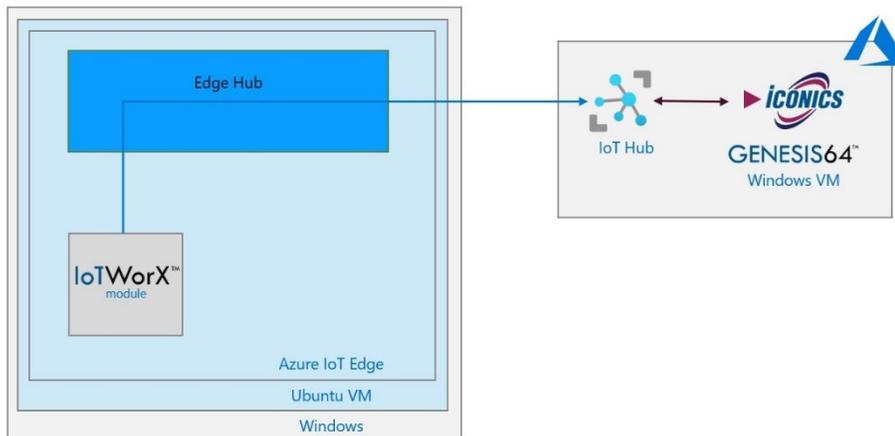


Figure 1: Data flow yielded by Quick Start

The instructions include a simulated temperature sensor running within the IoTWorX module.

In the *Tutorial*, you will find instructions for how to set up an ASA job on IoT Edge that takes data from another simulated temperature sensor, this time running as a separate ASA module, and sends them to IoT Hub. The data flow is as follows, including the IoT Hub Message Routes:

Combining IoTWorX and ASA on Azure IoT Edge

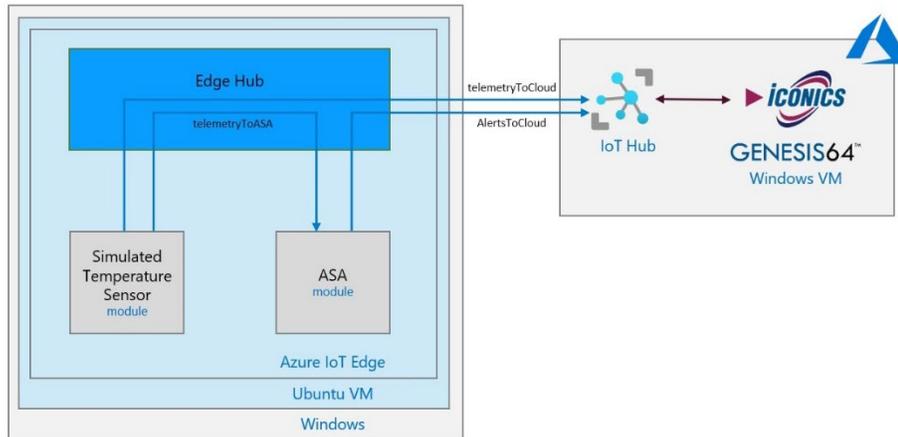


Figure 2: Data flow yielded by Tutorial

If you follow the instructions in these documents, you end up with two working applications running on IoT Edge: IoTWorX and ASA working side by side but independently and without communicating with each other. By getting those applications to work together, we can significantly increase the edge computing functionality available to the end user. For example:

Documentation	IoT Edge module	Data source	Output
<i>Quick Start</i>	IoTWorX	Simulated sensor	Temperature data
<i>Tutorial</i>	ASA	Simulated sensor	Temperature data and alerts
<i>Measuring Occupancy</i>	IoTWorX	Physical device	Occupancy data
This whitepaper	IoTWorX + ASA	Physical device	Reformatted and augmented occupancy data

The simplest combination of the *Quick Start* and *Tutorial* modules described in this whitepaper simply replaces the two simulated sensors with a physical device, using IoTWorX to get data from the device and ASA to send that unchanged to the cloud with a simple SQL query. However, now that this data flow is in place, we could modify the ASA SQL query to support many different scenarios without changing the IoTWorX configuration, for example:

- Issue alerts based upon occupancy, such as if the room is occupied at a time that it should not be, or for longer than it should be.
- Calculate averages over time, such as the average amount of time the room is occupied on different days of the week.
- Combine occupancy data with data from ASA inputs and issue instructions to the building's HVAC system, for example to increase the heating if the room is occupied and the temperature is below some threshold.

To do this, we need to get the applications to process data from sensors sequentially. That is, we need to get ASA to read the data from IoTWorX, apply whatever transformations are needed, and to send

Combining IoTWorX and ASA on Azure IoT Edge

them to IoT Hub. The following diagram shows this flow, also adding a real sensor instead of using simulated data:

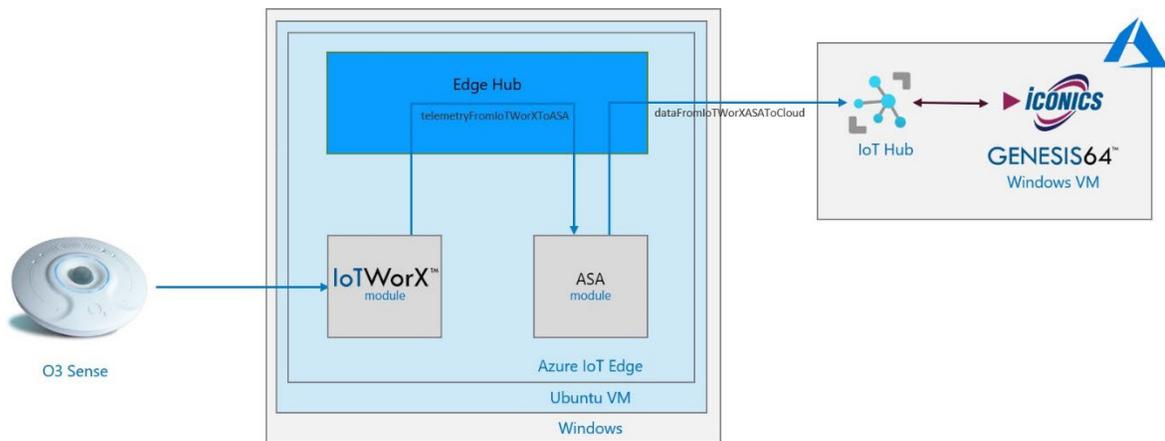


Figure 3: Data flow yielded by following this whitepaper

Specifically:

1. IoTWorX queries an O3 occupancy sensor over the BACnet protocol.
2. IoTWorX sends the data to the Edge Hub.
3. ASA reads the IoTWorX data from the Edge Hub and performs any preprogrammed transformations.
4. ASA sends the data to the Edge Hub.
5. The Edge Hub forwards the data to IoT Hub, where it is picked up by any Hub client such as ICONICS GENESIS64 for analysis.

Install IoTWorX on IoT Edge

First, follow the steps in the *Quick Start*, linked above, to create a working deployment of IoTWorX on IoT Edge. The *Quick Start* describes deploying on a physical computer, but as noted in the previous section, in this whitepaper, we describe using an Ubuntu virtual machine on an on-premises computer rather than directly on a physical machine. To be consistent with the names in this whitepaper, provide the following values as you follow the steps in the *Quick Start*:

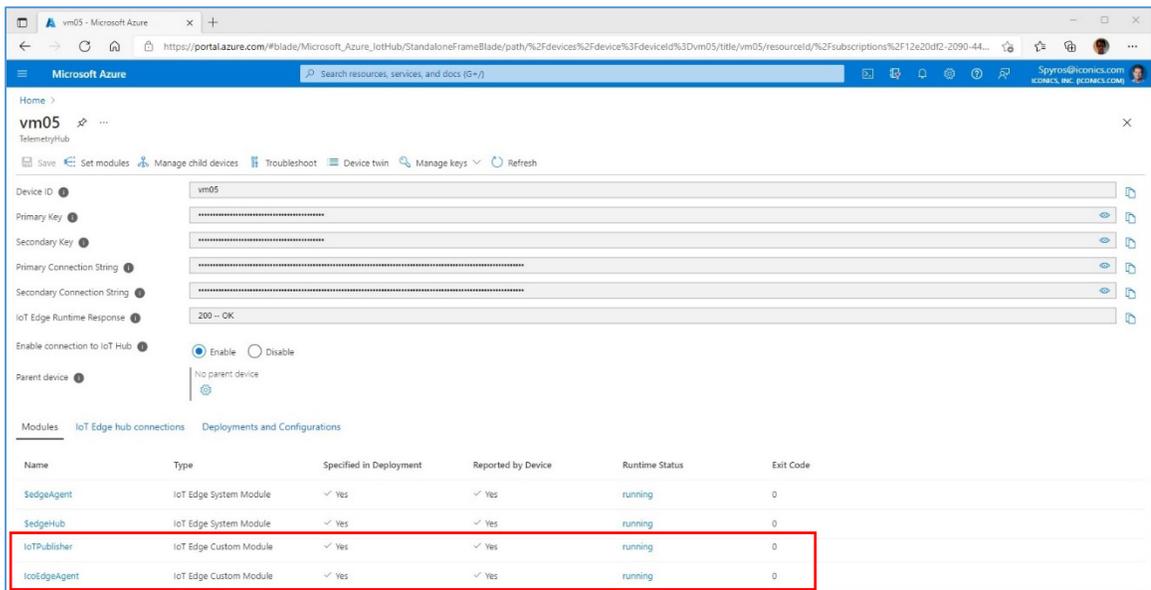
Field	Value
Azure IoT Hub name	TelemetryHub
IoT Edge Device name	vm05
IoT project name	iotworx

Combining IoTWorX and ASA on Azure IoT Edge

Stop when you get to **Deploying Configurations to the Device**, Step 4, *Add points to be published*. The *Quick Start* adds a simulator module to create data to analyze, and you can continue to follow those steps if you would like. However, for this whitepaper, we used a real physical device, the O3 occupancy sensor, instead of the simulator. To configure accessing this device over BACnet, follow the instructions in *Installing IoTWorX*, starting at **BACnet configuration**, Step 12. Provide the following values as you follow the steps in *Installing IoTWorX*:

Field	Value
BACnet Device name	O3 Hub 2
IoT Edge Device name	vm05
Publish List name	VM05 JSON Publist
Custom Encoder name	Gateway JSON Encoder
Publisher Connection	vm05

After deploying the configuration to the virtual machine **vm05**, check in the Azure portal that the IoTWorX modules, IoTPublisher and IcoEdgeAgent, have deployed correctly and are running.



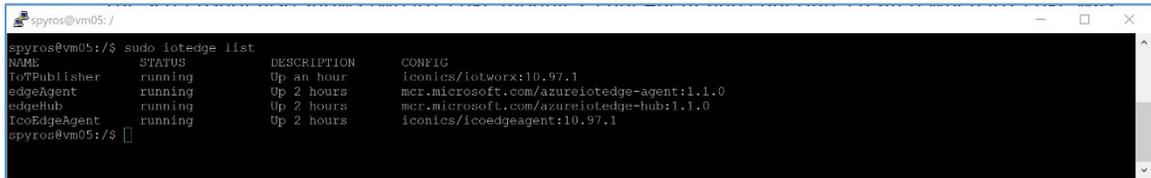
The screenshot also shows two IoT Edge modules, *edgeAgent* and *edgeHub*, created when IoT Edge was installed.

Next, SSH to the IP address of the virtual machine to check the deployment on the device itself. We used the open-source tool [PuTTY](#) to SSH into the virtual machine, but you could use any SSH client. At a command prompt, issue the following command:

Combining IoTWorX and ASA on Azure IoT Edge

```
sudo iotedge list
```

The modules you see running here should be the same as were seen in the Azure portal:



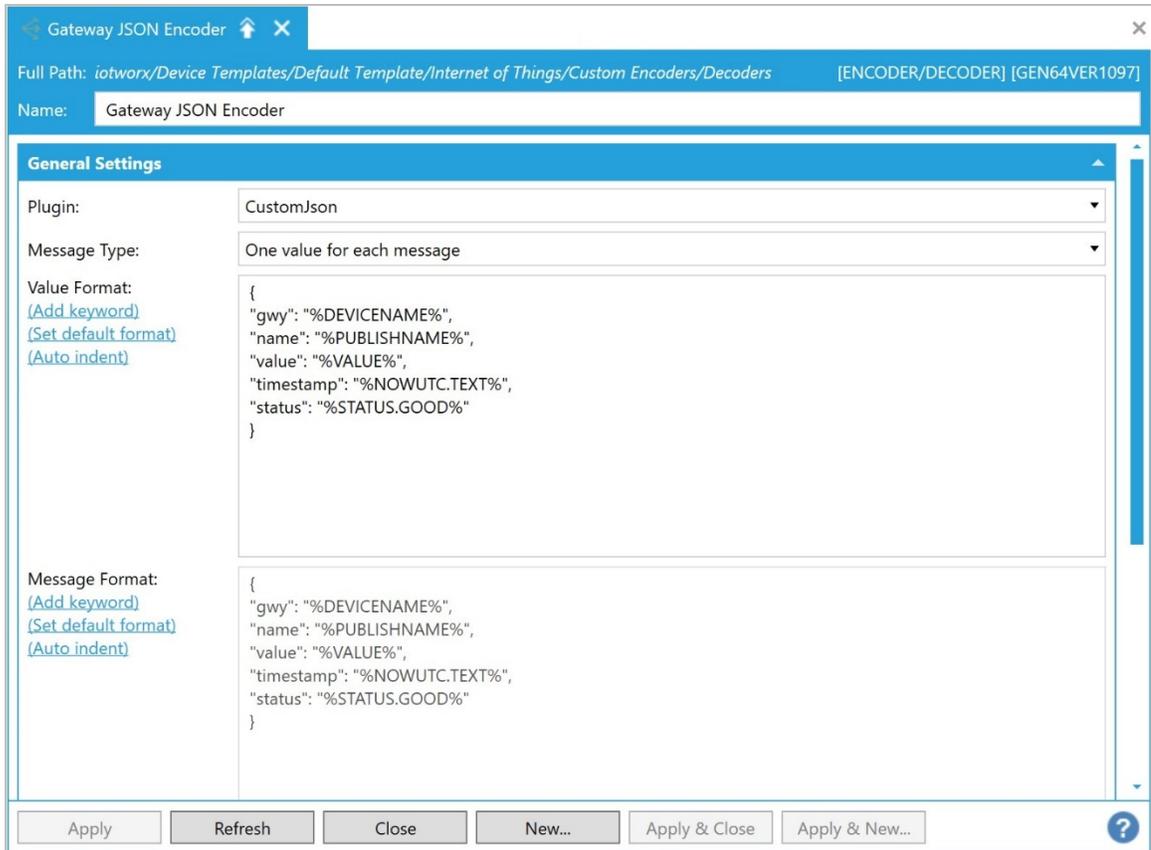
```
spyros@vm05:/ $ sudo iotedge list
NAME          STATUS      DESCRIPTION          CONFIG
IoTPublisher  running    Up an hour          iconics/iotworx:10.97.1
edgeAgent     running    Up 2 hours          mcr.microsoft.com/azureiotedge-agent:1.1.0
edgeHub       running    Up 2 hours          mcr.microsoft.com/azureiotedge-hub:1.1.0
IcoEdgeAgent  running    Up 2 hours          iconics/Icoedgeagent:10.97.1
spyros@vm05:/ $
```

At this point, you have IoTWorX querying the O3 occupancy sensor and sending the data to Azure IoT Hub. You can see the data coming into Azure IoT Hub using the [Azure IoT Explorer](#):

```
{
  "gwy": "vm05",
  "name": "Sound_volume",
  "value": 75.0,
  "timestamp": "2021-11-01T23:37:51.430Z",
  "status": true,
  "EventProcessedUtcTime": "2021-11-01T23:41:48.2392251Z",
  "PartitionId": 0,
  "EventEnqueuedUtcTime": "2021-11-01T23:37:51.9930000Z",
  "IoTHub": {
    "MessageId": null,
    "CorrelationId": null,
    "ConnectionDeviceId": "vm05",
    "ConnectionDeviceGenerationId": "637714035708812611",
    "EnqueuedTime": "2021-11-01T23:37:51.8530000Z"
  }
}
```

Note that the fields shown are those that we specified in the Gateway JSON Encoder that was created following the steps in *Installing IoTWorX*:

Combining IoTWorX and ASA on Azure IoT Edge



For now, that is all we need to do to configure IoTWorX. Next, we create an Azure Stream Analytics job on IoT Edge.

Create a simple ASA job on IoT Edge

Follow the steps in the *Tutorial*, linked above, to create an Azure Stream Analytics job on IoT Edge. Provide the following values as you follow the steps in the *Tutorial*:

Field	Value
ASA job name	ASAjobOnIoTworXData
Input	Telemetryfromiotworx
Output	datatoiothub

Stop when you get to **Configure your job**, Step 11, *Creating the ASA query*. Instead of the code shown in the *Tutorial*, replace the default text with the following query:

Combining IoTWorX and ASA on Azure IoT Edge

```
SELECT
    'ASAjobOnIoTWorXData' AS Sender,
    gwy AS Gateway,
    name AS Object,
    value AS Value,
    CONCAT (YEAR(TimeStamp),'-',MONTH(TimeStamp), '-', DAY(TimeStamp))
        AS Date,
    CONCAT( DATEPART(hh, DATEADD(hour,-7,TimeStamp)),':', DATEPART
        (mi, TimeStamp), ':', DATEPART(ss, TimeStamp)) AS TimePST
INTO
    datatoiothub
FROM
    telemetryfromiotworx
```

IoTWorX sends data in the following format, based upon the Custom Encoder we created:

```
"gwy": "vm05",
"name": "Sound_volume",
"value": 75.0,
"timestamp": "2021-11-01T23:37:51.430Z",
"status": true
```

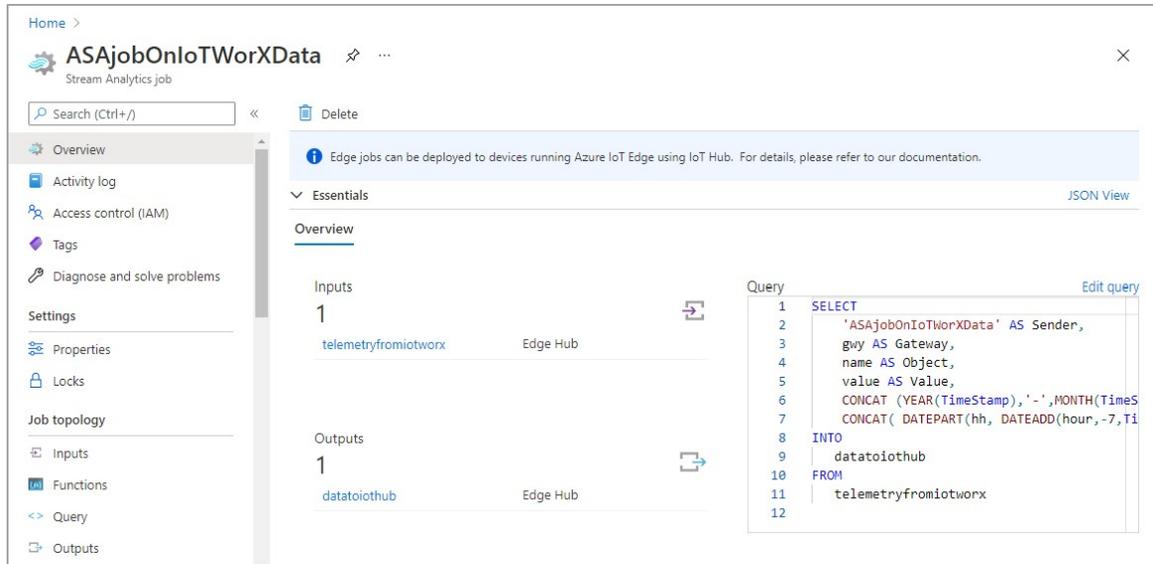
The ASA job above renames several of the fields and reformats the date and time, so that after the ASA job is executed, the data will look like the following:

```
"Sender": "ASAjobOnIoTWorXData",
"Gateway": "vm05",
"Object": "Sound_volume",
"Value": 75.0,
"Date": "2021-11-01",
"TimePST": "16:37:51"
```

The purpose of this whitepaper is to show how to connect IoTWorX and ASA, and for that all we need is this simple ASA job. In a production environment, the ASA query could be as complex as needed, including combining data from multiple sources, and fully utilizing all the functions supported by ASA.

The finished job should look like this in the Azure portal:

Combining IoTWorX and ASA on Azure IoT Edge



Deploy the ASA job and connect to IoTWorX

Having created the ASA job, the next step is to deploy it to the IoT Edge box and configure routes for the data from IoTWorX to the ASA input and from the ASA output to the IoT Edge Hub. To do this, continue following the instructions in the *Tutorial*, starting at **Deploy the job**, but using the field names shown above.

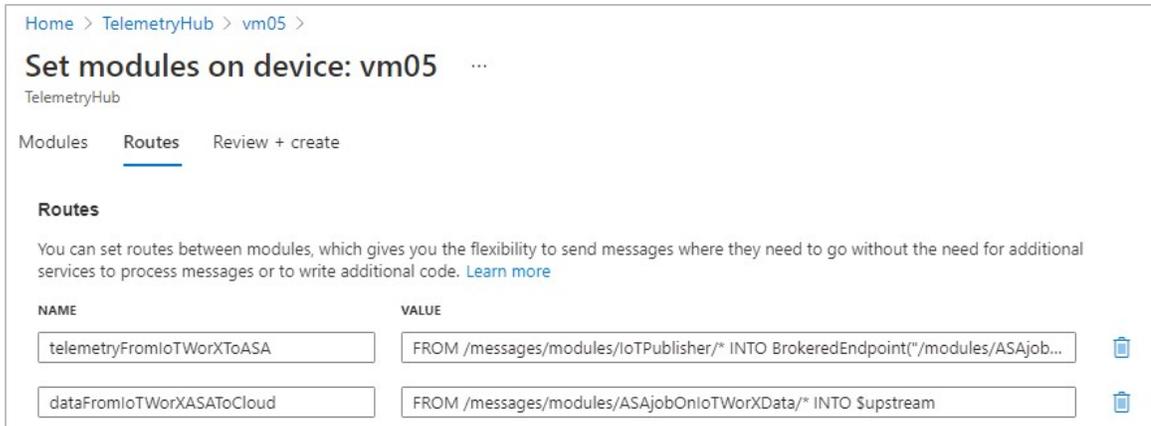
In **Deploy the job** Step 4, ignore adding the **SimulatedTemperatureSensor**, and in Step 5, select **ASAjobOnIoTWorXData** from the drop-down list.

In Step 9, we create the routes, which define how messages are passed between modules and the IoT Hub. Do not create the routes listed. Instead, create only these two routes:

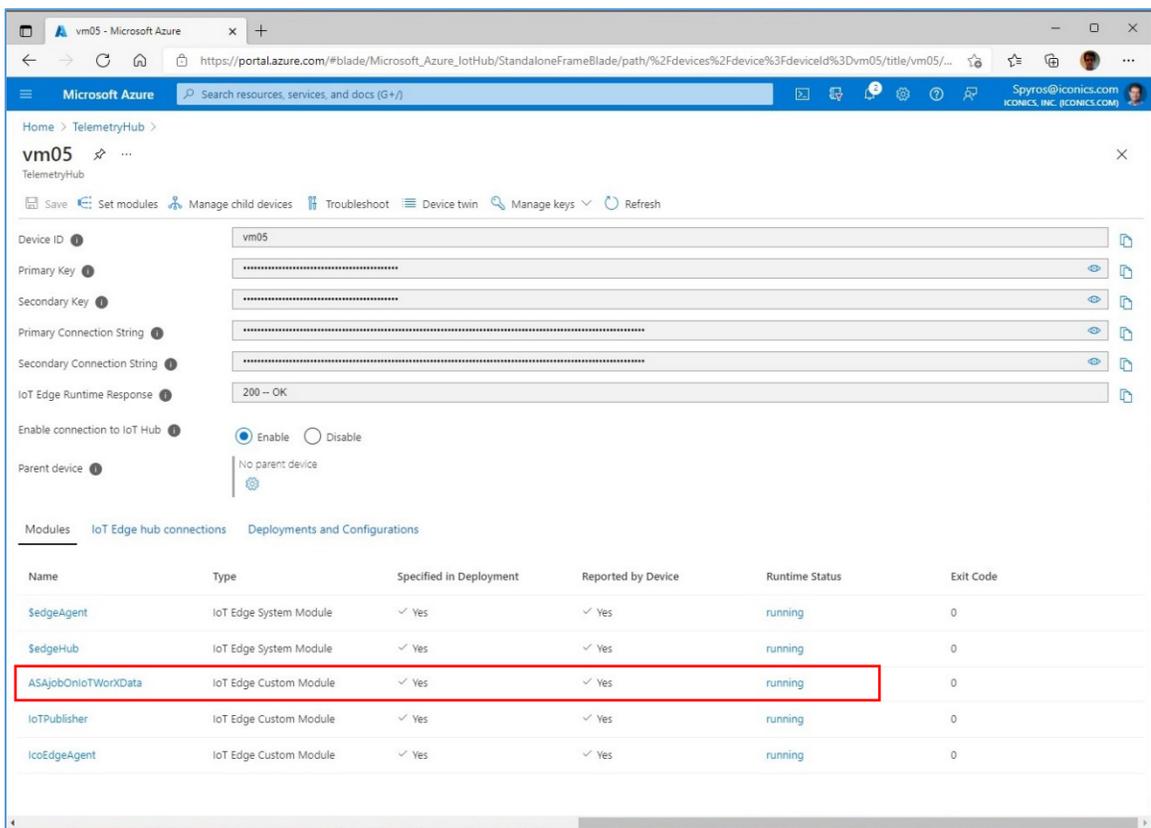
Name	Value
telemetryFromIoTWorXToASA	FROM /messages/modules/IoTPublisher/* INTO BrokeredEndpoint("/modules/ASAjobOnIoTWorXData/ inputs/telemetryfromiotworx")
dataFromIoTWorXASAToCloud	FROM /messages/modules/ASAjobOnIoTWorXData/* INTO \$upstream

The routes should look like this in the portal:

Combining IoTWorX and ASA on Azure IoT Edge



After creating the routes, the ASA job is deployed to the device. First check that the ASA module is running in the portal:



One note of caution – if you delete the ASA job but leave the routes configured, or if you create the routes without creating the ASA job, you will not see the data being published to the vm05 device in Azure IoT Hub. This is because the route to the vm05 device in Azure sends the data from IoTWorX to

Combining IoTWorX and ASA on Azure IoT Edge

the ASA module, and if the ASA module does not exist, it has no way to get to IoT Hub. To avoid this problem, follow the steps above in the sequence described.

View the data

You can use Azure IoT Explorer to check whether data is flowing to IoT Hub, but to see a larger swath of data, we can set up another ASA job in Azure to output some of the data to a blob. Here is the query we used:

```
SELECT
    Object, Value, Date, TimePST
INTO
    bloboutput
FROM
    iothub
WHERE
    Sender='ASAjobOnIoTWorXData'
```

Depending upon the refresh and publish rates configured for the Publish List, this data can be seen shortly in the blob created in the Azure portal storage account:

```
{"Object": "Acoustic_occupancy", "Value": "0", "Date": "2021-11-2", "TimePST": "11:19:33"}
{"Object": "Internal_humidity", "Value": 45.59548, "Date": "2021-11-2", "TimePST": "11:13"}
{"Object": "IR_temperature", "Value": 19.89952, "Date": "2021-11-2", "TimePST": "11:19:33"}
{"Object": "Light_level", "Value": 164.0, "Date": "2021-11-2", "TimePST": "11:19:33"}
{"Object": "Motion_sensor", "Value": "0", "Date": "2021-11-2", "TimePST": "11:19:33"}
{"Object": "Occupancy", "Value": "0", "Date": "2021-11-2", "TimePST": "11:19:33"}
{"Object": "Occupancy_inactivity_period", "Value": 300.0, "Date": "2021-11-2", "TimePST": "11:19:33"}
{"Object": "Occupant_temperature", "Value": 19.94736, "Date": "2021-11-2", "TimePST": "11:19:33"}
{"Object": "Occupant_humidity", "Value": 48.97536, "Date": "2021-11-2", "TimePST": "11:19:33"}
{"Object": "Sound_level", "Value": 30.59922, "Date": "2021-11-2", "TimePST": "11:19:33"}
{"Object": "Sound_volume", "Value": 75.0, "Date": "2021-11-2", "TimePST": "11:19:33"}
```

Combining IoTWorX and ASA on Azure IoT Edge

Summary

The steps shown in this paper illustrate how to combine two applications, ICONICS IoTWorX and Microsoft Azure Stream Analytics, on IoT Edge. To do this, we followed these steps:

1. Configured IoTWorX to send data to Azure IoT Hub.
2. Created an ASA job with an input of Edge Hub, and an output of Edge Hub.
3. Created a route for data from the IoTWorX IoTPublisher module to the named input in ASA.
4. Created a route for data from the named output in ASA to the IoT Edge Hub.

This inter-module communication opens the possibility of creating very powerful solutions on IoT Edge, beyond the capabilities of any one application. Both IoTWorX and ASA have large sets of functions that can be programmed. However, there are many that are supported in IoTWorX but not ASA, and vice versa. The combination of the two increases edge computing capabilities and can therefore support more complex intelligent edge scenarios.



Founded in 1986, ICONICS is an award-winning global software provider offering real-time visualization, HMI/SCADA, energy management, fault detection, manufacturing intelligence, MES, and a suite of analytics solutions for operational excellence. ICONICS solutions are installed in 70 percent of the Global 500 companies around the world, helping customers to be more profitable, agile and efficient, to improve quality, and to be more sustainable.

ICONICS is leading the way in cloud-based solutions with its HMI/SCADA, analytics, mobile and data historian to help its customers embrace the Internet of Things (IoT). ICONICS products are used in manufacturing, building automation, oil and gas, renewable energy, utilities, water and wastewater, pharmaceuticals, automotive, and many other industries. ICONICS' advanced visualization, productivity, and sustainability solutions are built on its flagship products: GENESIS64™ HMI/SCADA, Hyper Historian™ plant historian, AnalytiX® solution suite, and MobileHMI™ mobile apps. Delivering information anytime, anywhere, ICONICS' solutions scale from the smallest standalone embedded projects to the largest enterprise applications.

ICONICS promotes an international culture of innovation, creativity, and excellence in product design, development, technical support, training, sales, and consulting services for end users, systems integrators, OEMs, and channel partners. ICONICS has over 375,000 applications installed in multiple industries worldwide.

ICONICS Sales Offices



World Headquarters

100 Foxborough Blvd.
Foxborough, MA, USA, 02035
+1 508 543 8600
us@iconics.com



European Headquarters

Netherlands
+31 252 228 588
holland@iconics.com

Australia

+61 2 9605 1333
australia@iconics.com

China

+86 10 8494 2570
china@iconics.com

Czech Republic

+420 377 183 420
czech@iconics.com

France

+33 4 50 19 11 80
france@iconics.com

Germany

+49 2241 16 508 0
germany@iconics.com

India

+91 265 6700821
india@iconics.com

Italy

+39 010 46 0626
italy@iconics.com

Singapore

+65 6667 8295
singapore@iconics.com

UK

+44 1384 246 700
uk@iconics.com



For more, visit iconics.com

© 2021 ICONICS, Inc. All rights reserved. Specifications are subject to change without notice. AnalytiX and its respective modules are registered trademarks of ICONICS, Inc. GENESIS64, GENESIS32, Hyper Historian, IoTWorX, KPIWorX, CFSWorX, MobileHMI, WebHMI and their respective modules, OPC-to-the-Core, Make the Invisible Visible, and ICONICS company logo, are trademarks of ICONICS, Inc. Other product and company names mentioned herein may be trademarks of their respective owners.

Gold

Microsoft Partner

Six-time Partner of the Year Winner

