



Make the Invisible Visible™



Air Quality Monitoring Solution with ICONICS and Sysinno

An ICONICS Whitepaper

June 2022

Version 1.4

Contents

1	Introduction	4
2	Architecture	5
3	Azure infrastructure setup.....	6
4	Azure Sphere setup with Sysinno	6
5	IoT Subscriber Connection	8
6	AssetWorX configuration	10
6.1	Create the Equipment class	10
6.2	Instantiate equipment	12
7	AnalytiX-BI configuration	13
7.1	Create the Data Flow	13
7.2	Create the Data Model.....	15
8	BridgeWorX64 configuration	17
8.1	Create the Transaction.....	17
8.2	Configure the Transaction Template	18
8.3	Trigger transaction with IoT Hub messages.....	21
9	Visualize the air quality metrics.....	23
10	Conclusion.....	24

Copyright and Confidentiality

By accessing and using the instructions within (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 ICONICS nor any of its partners mentioned herewith and 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, ICONICS, Inc. All rights reserved.

Authors

- Zhi Wei Li, Director of Innovation and Incubation Solutions
- Spyros Sakellariadis, Global Director of IoT Business Development

1 Introduction

The air quality of the environment we live and work in has a significant effect on our health, wellbeing, and productivity. To understand the air quality in our environment, we need to measure the levels of the different conditions and volume of air pollutants in the air.

As part of ICONICS' commitment to the wellbeing of our customers, we have partnered with Sysinno to create an air quality monitoring solution that will be described in detail in this article to serve as a blueprint for your air quality monitoring needs.

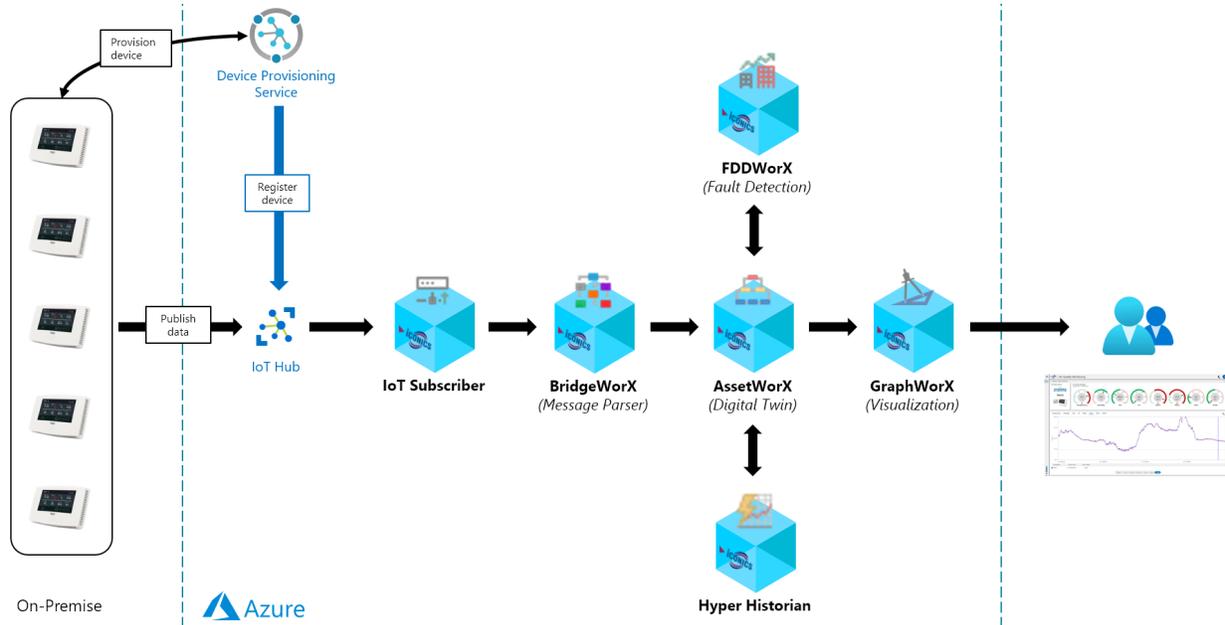
Sysinno is a producer of air quality monitoring devices sold under the iAeris brand. For the solution in this article, we utilized the iAeris14 series of devices. The iAeris2 device provides measurements of the following metrics:

- Temperature
- Humidity
- CO₂
- CO
- PM 2.5
- PM 10
- TVOC
- HCHO

The iAeris 14 device can provide data in multiple ways, ranging from MQTT to Modbus. However, one of the differentiators of the iAeris device, and the focus of this article, is its support for Azure Sphere. By leveraging Azure Sphere, the iAeris device can self-register itself into the Azure IoT Hub. This allows for a significantly faster deployment of the device in the field as all it takes is to power up the device and setup internet connectivity.

2 Architecture

The architecture of this solution is shown in the diagram below:



There are 3 primary components to the architecture:

- Azure IoT Hub
- Azure Device Provisioning Service
- ICONICS Suite Virtual Machine

All ICONICS capabilities are available in the ICONICS Suite Virtual Machine.

The FDDWorX capability shown in the architecture diagram can optionally be added to the solution to provide fault detection of air quality issues. Learn more about FDDWorX here:

https://docs.iconics.com/V10.97/GENESIS64/Help/ICONICS_Product_Help.htm#Apps/WBDT/FDDWorX/FDDWorX_Overview.htm

3 Azure infrastructure setup

For this solution, we need to first set up the following Azure infrastructure components:

- Azure IoT Hub
- Azure Device Provisioning Service
- ICONICS Suite Virtual Machine

To provision the Azure IoT Hub and Device Provisioning Service, please follow the instructions in this Microsoft article:

<https://docs.microsoft.com/en-us/azure/iot-dps/quick-setup-auto-provision>

For this solution, the IoT Hub must be at least of the S1 tier.

To provision the latest version of ICONICS Suite Virtual Machine from the Azure Marketplace, please follow the instructions in this ICONICS article:

<https://iconics.com/Documents/Whitepapers/Creating-Azure-Virtual-Machine-Image-Definition>

Proceed to the next section once the necessary Azure components are successfully created.

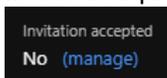
4 Azure Sphere setup with Sysinno

The iAeris device can self-register and publish directly to the IoT Hub by utilizing Azure Sphere. This usually involves a process of exchanging certificates and validation thumbprints.

However, to simplify the process, it is recommended that Sysinno support personnel be given temporary access to the Device Provisioning Service and IoT Hub that is set up in the previous section.

Follow the steps below to provide Sysinno support personnel temporary access to the necessary Azure resources:

1. Request the Sysinno support personnel's email address
2. On the Azure portal, open the configuration page of the Device Provisioning Service that was set up in the previous section
3. From the left menu, select **Access control (IAM)**
4. In the right dialog, click on **+ Add** from the top
5. A **Add role assignment** blade opens up on the right
6. In the blade, set the following options:
 - a. Set the **Role** field to be **Contributor**
 - b. Enter the email address of the Sysinno support personnel
 - c. Select the user with the email you entered
 - d. The user should now be under the **Selected members** section
 - e. Click on **Save**
7. Back in the **Access control** configuration page, click on **Role assignments**
8. Select the newly added user
9. In the user profile page, look for the **Invitation accepted** field



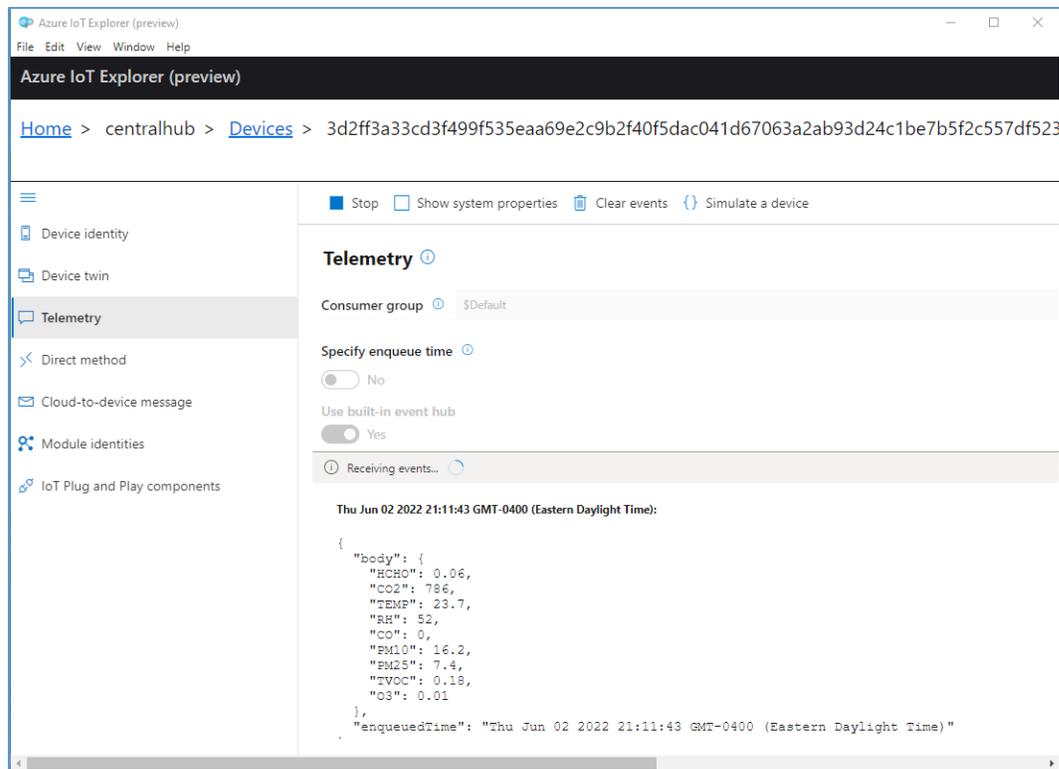
10. Click on the **(manage)** link next to **No**
11. A **Manage invitations** blade shows up on the right
12. For the **Resend invite?** option, select the **Yes** radio button
13. Confirm
14. A new field called **Invitation URL** shows up below
15. Copy the URL in the Invitation URL field
16. Send the copied URL to the Sysinno support personnel

Once the Sysinno support personnel receives the invitation URL, they would be able to accept your invite to set up the Device Provisioning Service. The setup primarily involves adding Sysinno's certificate to the Device Provisioning Service such that all future iAeris devices can self-register to the IoT Hub linked to the Device Provisioning Service.

By leveraging the Device Provisioning Service, all you have to do to set up the device on site is to provider power and setup the WiFi settings to give the device internet access. Please refer to the iAeris device user manual on setting up WiFi connectivity.

To verify that the data is being published by the iAeris device you can use the Microsoft Azure IoT Explorer:

1. Launch Azure IoT Explorer
2. Add the IoT Hub to which the iAeris publishes, using the Shared access policy key and Connection string. To write this whitepaper we created and used an IoT Hub we named **CentralHub**, but you can use any valid name you wish.
3. Click on **View devices in this hub**
4. Click on the **Device ID** for the iAeris device. This will be a long string of letters and numbers created by the Device Provisioning Service and the iAeris device.
5. Select **Telemetry** from the left navigation menu
6. Click **Start** at the top. After a few minutes, you will start to see the data coming into IoT Hub:



5 IoT Subscriber Connection

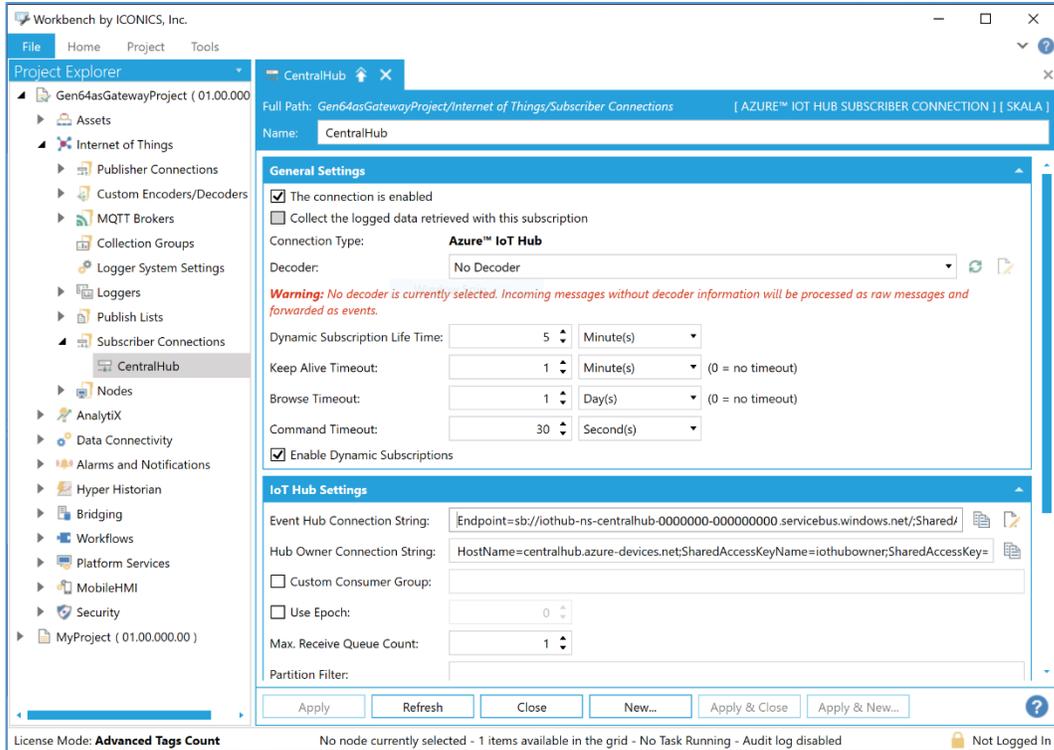
To be able to receive data published by iAeris into IoT Hub, we must subscribe to the IoT Hub from Genesis64.

Create a subscription with the following steps:

1. From the IoT Hub that was set up in section 3, obtain the following:
 - a. Event Hub Connection String
 - found in the **Built-in endpoints** section of the Hub → **Event Hub-compatible endpoint**
 - b. Hub Owner Connection String
 - found in the **Shared access policies** section of the Hub → **iothubowner** → **Connection string-primary key**
2. Open ICONICS Workbench
3. In the Workbench project, expand **Internet of Things**
4. Right click on **Subscriber Connections** → **Add Subscriber Connection**
5. Name the Subscriber Connection the same as the IoT Hub created earlier. In the example used for this whitepaper, the IoT Hub was called **CentralHub**, so we named the subscriber Connection **CentralHub**.
6. Ensure the connection type is **Azure IoT Hub**
7. Change the **Default Decoder** to be **No Decoder**
8. In the **IoT Hub Settings** section, enter the **Event Hub Connection String** and **Hub Owner Connection String** information from step 1
9. Click **Apply**

Start the **Subscriber** service from the **Home** ribbon

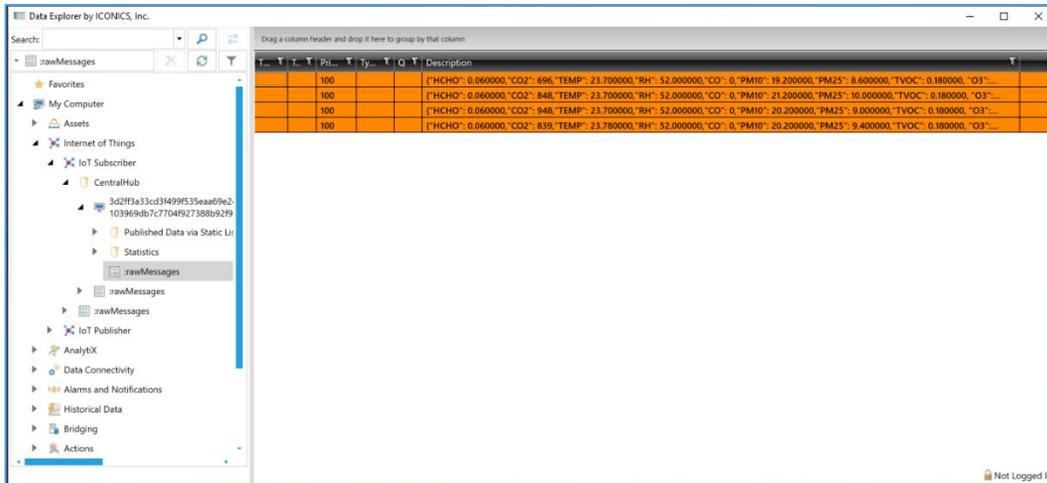
The configuration of IoT Subscriber Connection is now complete. Using our **CentralHub** as the IoT Hub, the properties of the Subscriber Connection look like this:



At this stage, you can verify that the subscriber connection is working correctly by launching the ICONICS Data Explorer from the Windows Start Menu:

1. Launch Data Explorer
2. Browse to My Computer → Internet of Things → CentralHub → {iAeris Sphere ID} → :rawMessages

After a while you will see the raw messages as received by the Subscriber from IoT Hub:



Note that in Azure IoT Explorer, the raw data is in a nested JSON message. The ICONICS Subscriber shows this same nested JSON message. The next sections in this paper show how to create a ICONICS AssetWorX digital twin to store this data, and how to use ICONICS BridgeWorX to parse this incoming JSON string into a format that can be stored in AssetWorX.

6 AssetWorX configuration

Before we can utilize the data published by the iAeris device, we need to create a digital twin of the device in AssetWorX, ICONICS' digital twin platform. This provides a place to store the most recent value for each published metric.

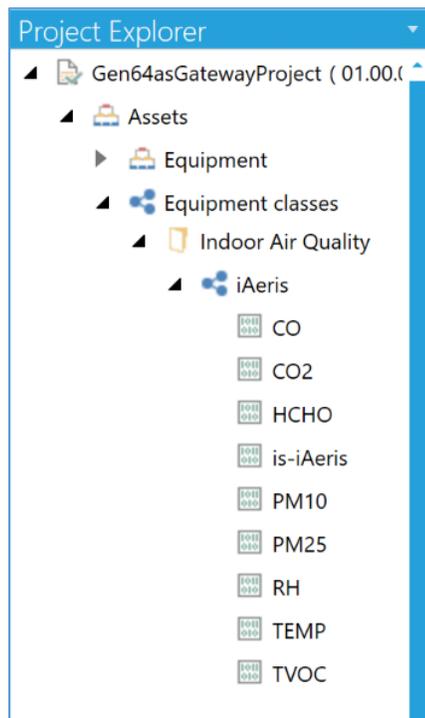
6.1 Create the Equipment class

We start by creating an equipment class that serves as the template of the iAeris device. Follow the steps below to create an equipment class of an iAeris device:

1. In the ICONICS Workbench project, expand **Assets**
2. Right click on **Equipment classes** → **Add Folder**
3. Give the folder a name. We named the folder **Indoor Air Quality**. Click **Apply**
4. Right click on the newly created folder and select **Add Equipment class**
5. Give the equipment class a name. We called it **iAeris**
6. In the Display Name field, enter **iAeris**
7. Click **Apply**
8. Right click on the **iAeris** equipment class and select **Add Equipment class property**
9. Give this property a name, for example **is-iAeris**
10. Select the **Real Time Data** tab and configure the following fields:
 - a. Set the **Source type** to **Static value**
 - b. Set the **Data type** to **Boolean**
 - c. Set the **Value** to **True**
11. Click **Apply**
12. Right click on the **iAeris** equipment class and select **Add Equipment class property**
13. Name this property **CO**
14. Select the **Real Time Data** tab and configure the following fields:

- a. Set the **Source type** to be **Static value**
 - b. Set the **Data type** field to be **Float32**
 - c. Set the **Value** field to be **0**
 - d. Check the **Use database cache** checkbox
15. To easily log any value change in Hyper Historian, select the **Historical Data** tab and configure the following fields:
 - a. Set the **Source type** to **Hyper Historian Tag**
16. Click **Apply**
17. Right click on the **CO** equipment class property and select **Multiply**
18. In the **Multiply Options** dialog:
 - a. Set the **Number of copies** to be **7**
 - b. Click **OK**
19. You should now see 7 additional copies of the **CO** equipment class property

20. Rename each of them to match one of the names of the published metrics:
- | | |
|-------------------------------|-------------------------------|
| <input type="checkbox"/> CO | <input type="checkbox"/> CO2 |
| <input type="checkbox"/> HCHO | <input type="checkbox"/> PM10 |
| <input type="checkbox"/> PM25 | <input type="checkbox"/> RH |
| <input type="checkbox"/> TEMP | <input type="checkbox"/> TVOC |
21. Update the parameter definitions of the iAeris equipment class with the following steps:
- Right click on the **iAeris** equipment class
 - Select **Update parameter definitions**
 - Uncheck **Simulate** operation
 - Click OK
 - Once the update process is complete, click **Close**
22. If you used the same names as we did for this paper, you would have an equipment class that looks like this:

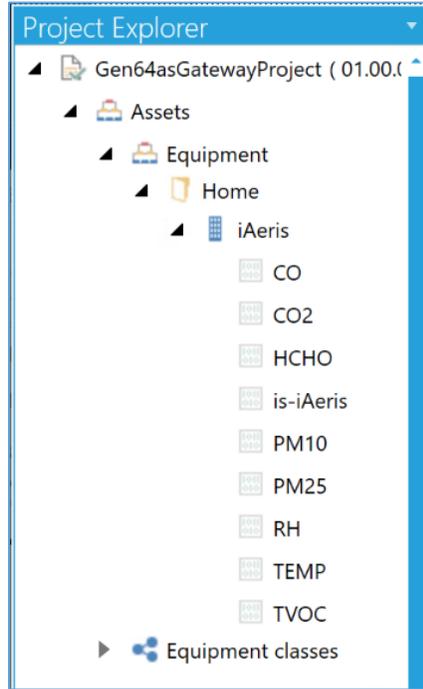


6.2 Instantiate equipment

Now that we have the equipment class that describes the iAeris device, we can instantiate the iAeris device into the AssetWorX equipment hierarchy. Follow the steps below to create equipment based on the iAeris equipment class

- Under the **Assets** provider tree, right click on **Equipment** → **Add Equipment**
- Name the equipment something descriptive. We named it **Home**.
- Right click on **Home** and select **Add Equipment from library**
- In the **Equipment Class Library** dialog, browse to and select the **iAeris** equipment class in the **Indoor Air Quality** folder
- In the **Instantiation Options** dialog:
 - Click on **Instantiate**
 - Click **Close** once successful

6. You should now see the equipment created with the desired Device ID name with the equipment properties as defined in the class:



The values of each equipment property will be updated by a BridgeWorX transaction that parses the published JSON message created later in this whitepaper.

The configuration of AssetWorX is now complete.

7 AnalytiX-BI configuration

In order to map the metrics published by each iAeris device to its respective digital twin in AssetWorX, we need to match the iAeris Device ID in the incoming message with the AssetWorX equipment whose name matches the Device ID.

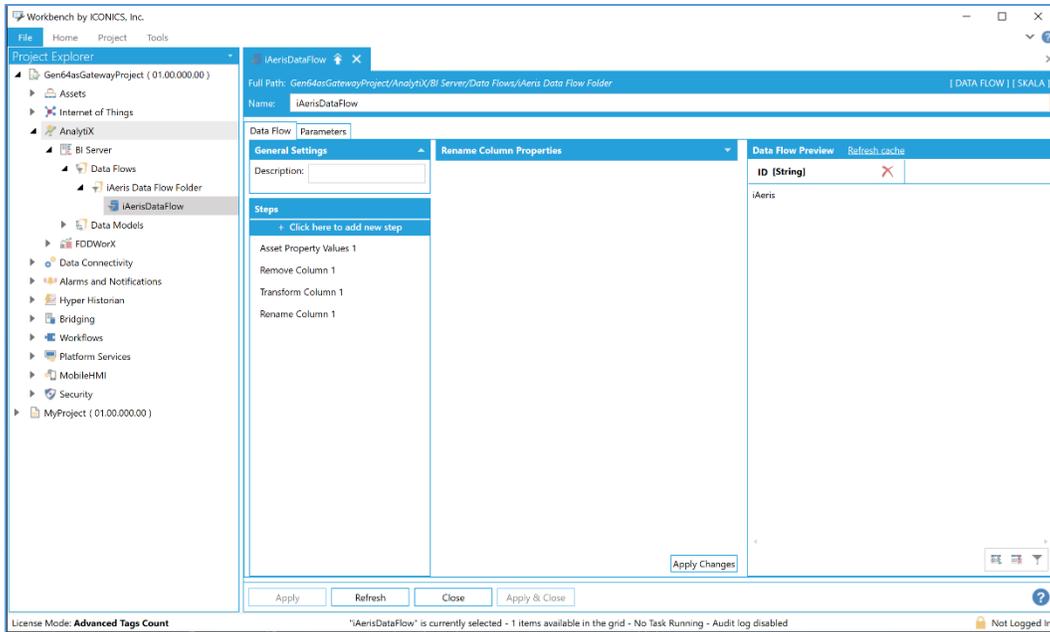
To achieve this, we will use AnalytiX-BI to extract the list of AssetWorX equipment that are iAeris devices. The extracted list will later be used by the message parsing transaction in BridgeWorX64 to match the iAeris Device ID in the published message to the respective AssetWorX equipment. The transaction will then update the equipment property per device that matches the metric parsed from the message.

7.1 Create the Data Flow

We start by creating a data flow to bring in the list of equipment that contain the iAeris equipment property. Follow the steps below to configure the necessary data flow:

1. In the ICONICS Workbench project, expand **AnalytiX → BI Server**
2. Right click on **Data Flows → Add Data Flow Folder**

3. Give the folder a name. We called it **iAeris Data Flow Folder**.
4. Right click on **iAeris Data Flow Folder** and select **Add Data Flow**
5. Give the data flow a name. We called it **iAerisDataFlow**
6. In the **Data Flow** tab, under **Steps**, click on the top where it says **+ Click here to add new step**
7. From the menu, select **Data Sources → Asset Property Values**
8. In the **Asset Property Values Properties** dialog in the middle, define the configuration fields as follows:
 - a. Set the **Property-Read batch size** to **500**
 - b. Set the **Property-Read timeout** to **5** seconds
 - c. Set the **Property Name Filter** to **is-iAeris**
9. In the **Data Sources** dialog in the middle:
 - a. Click on **Click here to add new item**
 - b. Click on the tag browser button, , to the right of the highlighted field
 - c. Browse to the folder that contains the iAeris equipment under **My Computer → Assets → Home**. Double click on **iAeris** and click OK.
 - d. In the **Data Sources** panel, under **Asset Path**, you will see **ac:Home/iAeris/**. Change that to **ac:Home/iAeris/*** so that all the items under this folder will be brought in.
10. Click on **Apply Changes**.
11. From the **Steps** section, add a **Remove Column** step.
12. In the **Remove Column Properties** section, uncheck all the columns except the **AssetPropertyPath**
13. Click on **Apply Changes**.
14. From the **Steps** section, add a **Transform Column** step
15. In the **Transform Column Properties** dialog, set the following:
 - a. Set the **Column Name** to be **AssetPropertyPath**
 - b. Click on the **Configure the expression** link
 - c. Enter this expression to extract the Device ID used as the equipment name:
`getat(split({AssetPropertyPath}),1,"/"),1)`
 - d. Click **OK**
16. Click on **Apply Changes**
17. In the **Data Flow Preview** table on the right, it should show a single column table with the column header called **AssetPropertyPath**.
18. From the **Steps** section, add a **Rename Column** step.
19. In the **Rename Column Properties** dialog, set the following:
 - a. Set the **Column Name** to be **AssetPropertyPath**
 - b. Set the column name to **ID**
20. Click on **Apply Changes**
21. Click **Apply** to save this data flow. The data flow should look like this:

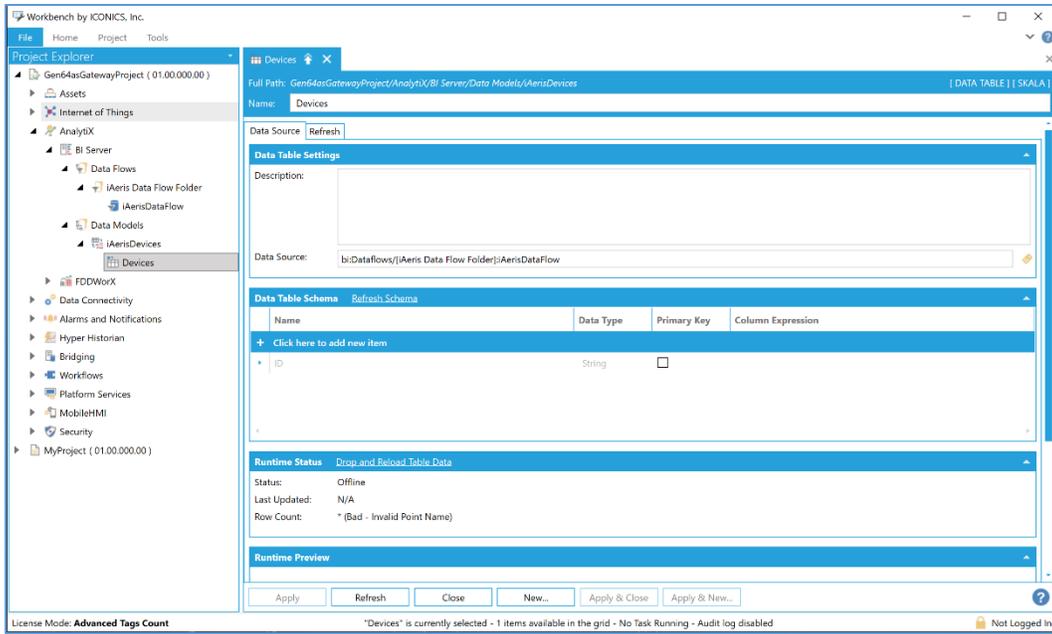


7.2 Create the Data Model

The Data Flow above provides a list of the iAeris Device IDs that are configured in AssetWorX. To allow the message parsing transaction to query this list, it must first be cached in a Data Model. Create the Data Model with the following steps:

1. In the ICONICS Workbench project, expand **AnalytiX** → **BI Server**
2. Right click on **Data Models** → **Add Data Model**
3. Give the data model a name. We called it **iAerisDevices**
4. Click **Apply**
5. Right click on the new Data Model → **Add Data Table**
6. Give the data table a name. We used **Devices**
7. In the Data Table configuration dialog, enter the following information:
 - a. For the **Data Source** field, click on the data browser icon, , on the right
 - b. Browse to the **iAerisDataFlow** Data Flow created before under **My Computer** → **AnalytiX** → **BI Server** → **Data Flows**
 - c. Select the Data Flow
 - d. Click **OK**
 - e. Click away from the **Data Source** field
 - f. Select the **Refresh** tab
 - g. Click **Apply**

The Data Model should look like this:



Start the **Point Manager** service from the **Home** ribbon

The configuration of AnalytiX-BI is now complete.

8 BridgeWorX64 configuration

BridgeWorX64 is ICONICS' transactional workflow module. In this solution, a transaction will be configured in BridgeWorX64 to parse the incoming messages from iAeris devices.

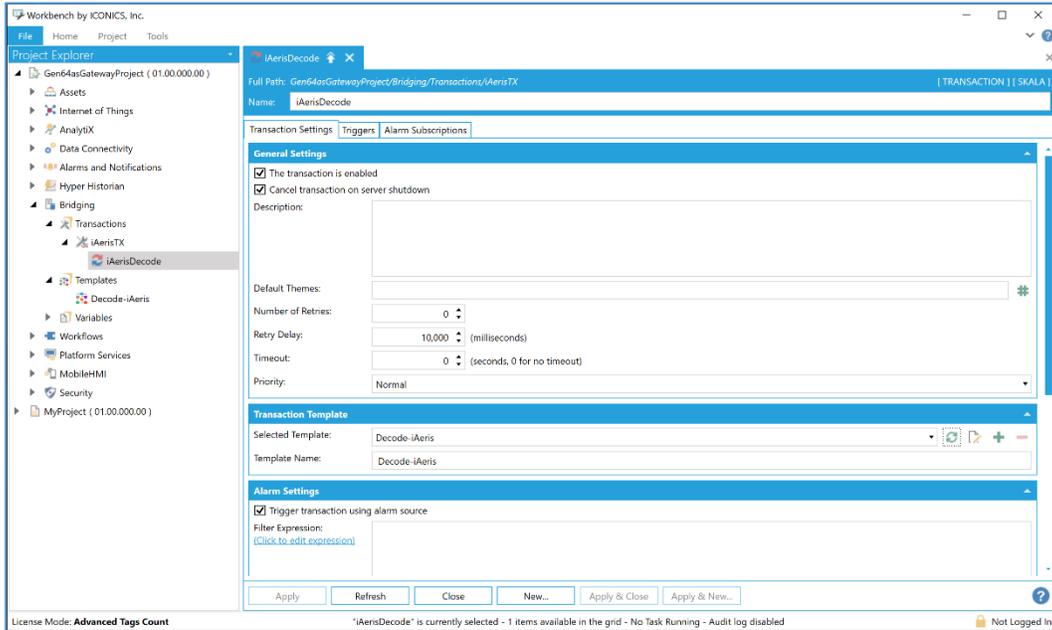
The messages published by iAeris devices will be first received by the IoT Subscriber Connection created in a prior section. The message parsing transaction would then subscribe to the IoT Subscriber Connection to receive the published messages and write them to the model.

8.1 Create the Transaction

Create the message parsing transaction with the following steps:

1. In the ICONICS Workbench project, expand **Bridging → Transactions**
2. If there is no existing configuration under Transactions, create a new configuration with the following steps:
 - a. Right click on **Transactions → Add Configuration**
 - b. Give the configuration a name. We called it **iAerisTX**
 - c. Check **Active Configuration**
 - d. For the **Temporary Directory** field, provide a file location such as **C:\Program Files\ICONICS\Temp**
 - e. Click **Apply**
3. Right click on the active configuration and select **Add Transaction**
4. Give the transaction a name. We called it **iAerisDecode**.
5. Check **The transaction is enabled**
6. In the **Transaction Template** section's **Selected Template** field, click on the + button to the right to create a new transaction template
7. Give the transaction template a name in the **Template Name** field. We called it **Decode-iAeris**.
8. Click **Apply**

9. The transaction settings should look like this:



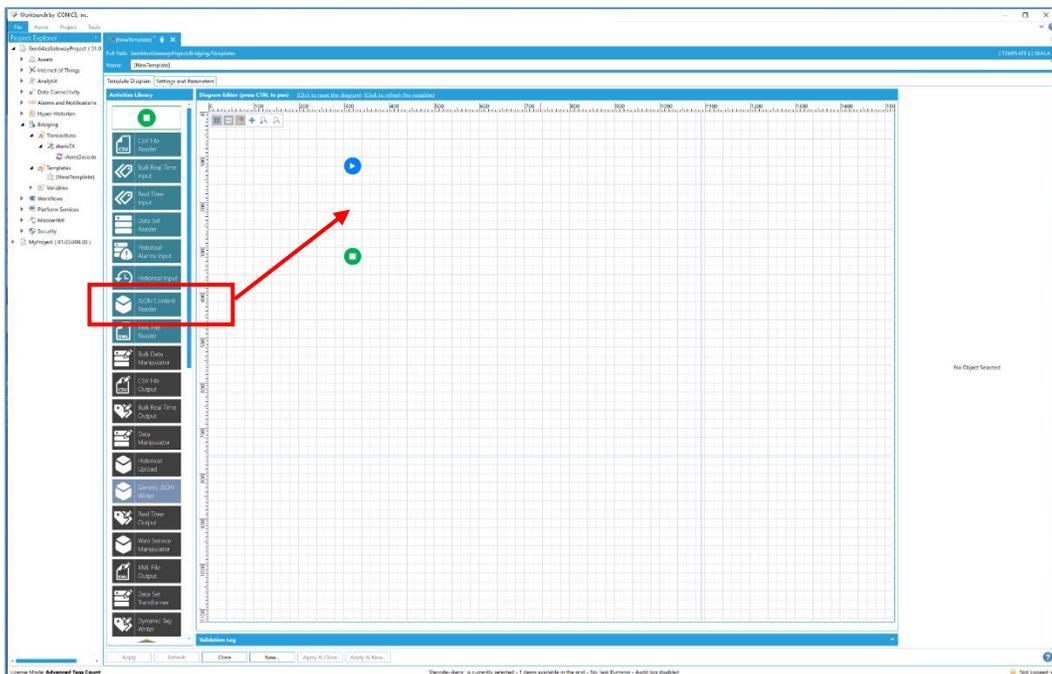
10. In the **Transaction Template** section, click on the edit button  to edit the template.

8.2 Configure the Transaction Template

The Transaction Template is where the logic of the transaction workflow is defined.

Follow the steps below to define the logic necessary to parse the incoming messages for iAeris devices:

1. From the **Activities Library** on the left, drag in a **JSON Content Reader** activity:



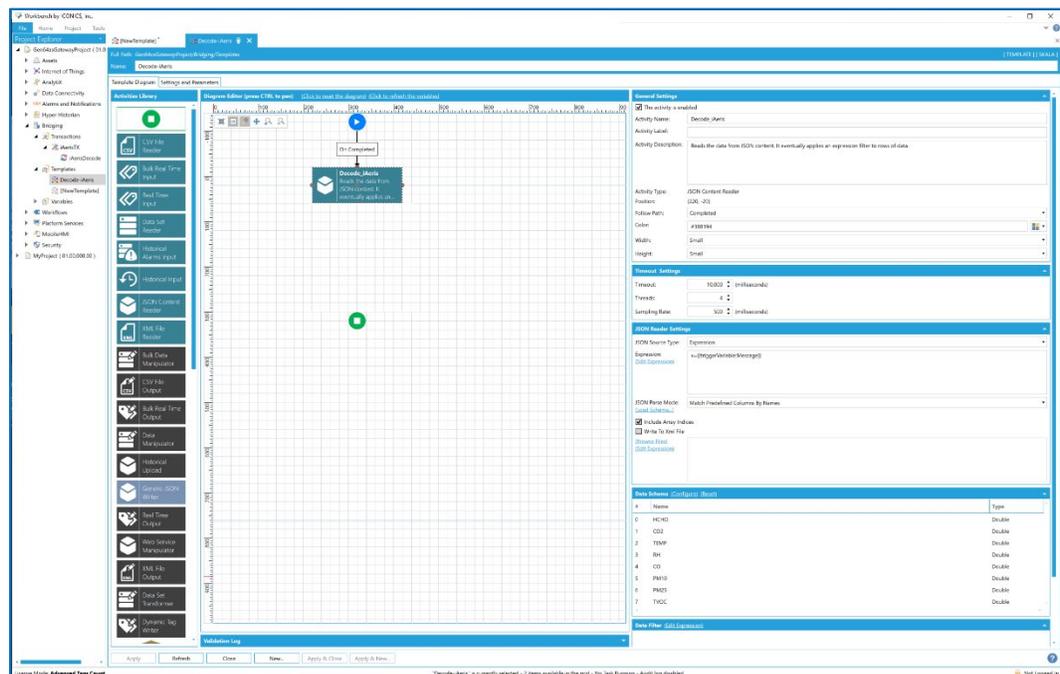
2. Select the new activity and edit its properties as follows:

- a. Give the activity a name, for example **Decode_iAeris**
- b. In the **JSON Reader Settings** section, set the following settings:
 - i. Set the **JSON Source Type** to be **Expression**
 - ii. Click on the **Edit Expression** link
 - iii. In the expression editor, paste in this expression:

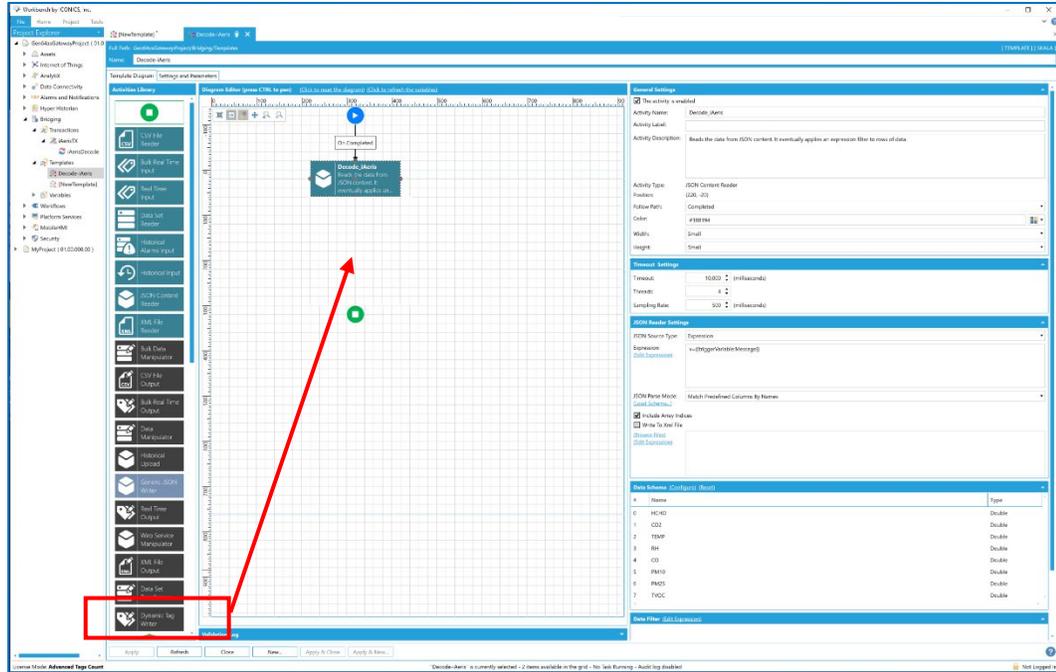

```
{triggerVariable:Message}
```
 - iv. Click **OK**
 - v. Set the **JSON Parse Mode** to be **Match Predefined Columns By Names**
- c. In the **Data Schema** section, click on the **Configure** link
- d. Define the columns list as follows:

Configure the columns						
#	Name	Display Name	Data Type	Output	Output E	
+ Click here to add new item						
0	HCHO		Double	<input checked="" type="checkbox"/>		
1	CO2		Double	<input checked="" type="checkbox"/>		
2	TEMP		Double	<input checked="" type="checkbox"/>		
3	RH		Double	<input checked="" type="checkbox"/>		
4	CO		Double	<input checked="" type="checkbox"/>		
5	PM10		Double	<input checked="" type="checkbox"/>		
6	PM25		Double	<input checked="" type="checkbox"/>		
7	TVOC		Double	<input checked="" type="checkbox"/>		

- e. Click OK. The completed activity should look like this:



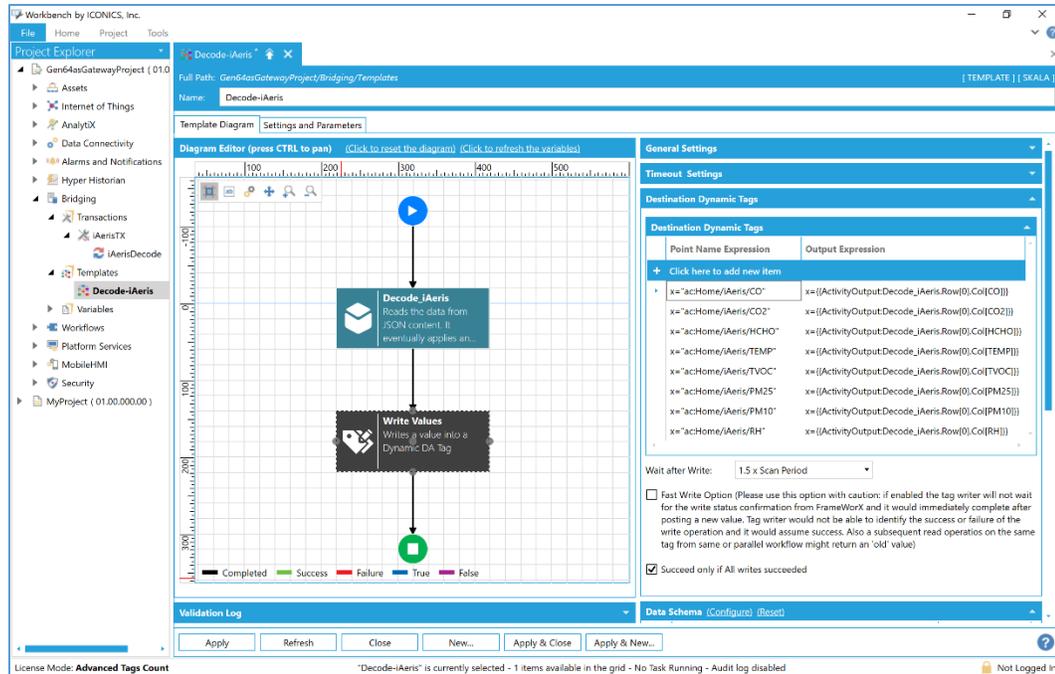
3. From the **Activities Library** on the left, drag in a **Dynamic Tag Writer** activity:



4. Select the new activity and edit its properties as follows:
 - a. Give the activity a name, for example **Write Values**
 - b. In the **Destination Dynamic Tags** section, add the list of tags as follows:

Point Name Expression	Output Expression
+ Click here to add new item	
x="acHome/iAeris/CO"	x={{ActivityOutput.Decode_iAeris.Row[0].Col[CO]}}
x="acHome/iAeris/CO2"	x={{ActivityOutput.Decode_iAeris.Row[0].Col[CO2]}}
x="acHome/iAeris/HCHO"	x={{ActivityOutput.Decode_iAeris.Row[0].Col[HCHO]}}
x="acHome/iAeris/TEMP"	x={{ActivityOutput.Decode_iAeris.Row[0].Col[TEMP]}}
x="acHome/iAeris/TVOC"	x={{ActivityOutput.Decode_iAeris.Row[0].Col[TVOC]}}
x="acHome/iAeris/PM25"	x={{ActivityOutput.Decode_iAeris.Row[0].Col[PM25]}}
x="acHome/iAeris/PM10"	x={{ActivityOutput.Decode_iAeris.Row[0].Col[PM10]}}
x="acHome/iAeris/RH"	x={{ActivityOutput.Decode_iAeris.Row[0].Col[RH]}}

5. Now that all necessary activity blocks are configured, link the activities into a workflow as shown below:

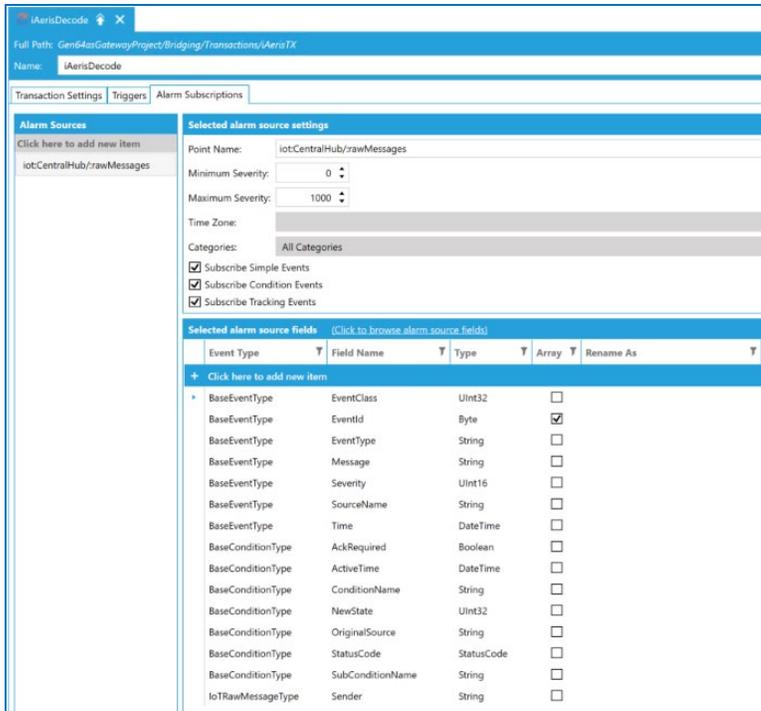


6. Click **Apply** when complete

8.3 Trigger transaction with IoT Hub messages

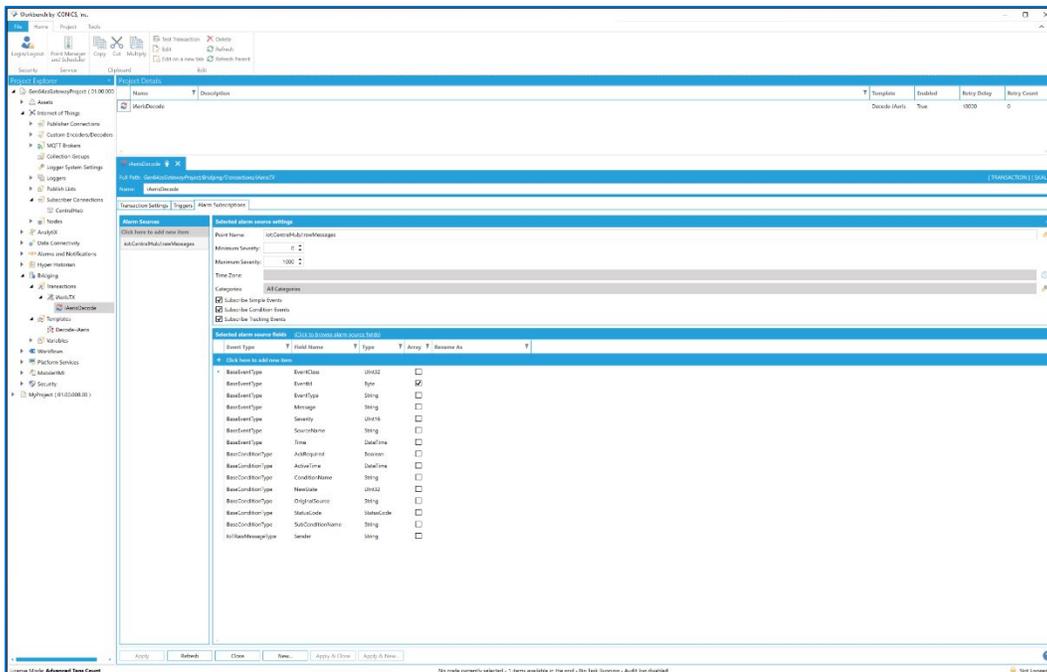
To trigger the transaction based on messages published to IoT hub, it needs to be configured to subscribe to the IoT Hub via the IoT Subscriber Connection defined before. Follow the steps below to connect the transaction to the IoT Hub:

1. Expand the active configuration, **iAerisTX**, under Bridging → Transactions
2. Right click on the message parsing transaction item **iAerisDecode** created earlier and select **Edit**.
3. In the **Transaction Settings** tab, **Alarm Settings** section, check **Trigger transaction using alarm source**.
4. In the **Alarm Subscriptions** tab click on **Click here to add new item**
5. Click on the browse button, 
6. In the **Data Browser**, browse under My Computer → Internet of Things and select **:rawmessages** under **CentralHub**, the subscriber connection configured earlier
7. Click OK
8. Click on the **Click to browse alarm source fields** link in the **Selected alarm source fields** section
9. Scroll through the various folders and select fields to match the following configuration:



Start the **Point Manager and Scheduler** service from the **Home** ribbon.

The configuration of BridgeWorX64 is now complete. The completed configuration should look like this:



When the iAeris device publishes its data to IoT Hub, it will be received by BridgeWorX64 and trigger the transaction to parse the JSON message and update the appropriate equipment property in AssetWorX for the respective device.

9 Visualize the air quality metrics

With the air quality metrics now available via AssetWorX equipment properties, we can easily create an air quality dashboard utilizing GraphWorX and its collection of templates, controls, and symbols. The examples below illustrate what a dashboard that visualizes the current value and historical trend of each air quality metric could look like:



10 Conclusion

Having an air quality monitoring solution as part of the overall building monitoring system is increasingly critical as we gain a better understanding of how the quality of the air around us affect our health, wellbeing, and productivity.

Leverage ICONICS GENESIS64, Sysinno iAeris and Azure IoT technologies to easily deploy an air quality monitoring solution and gain a better understanding of your environment.



Founded in 1986, ICONICS, a group company of Mitsubishi Electric Corporation, 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

© 2022 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.

A Group Company of Mitsubishi Electric Corporation

Gold

Microsoft Partner

Ten-time Microsoft Partner of the Year

