

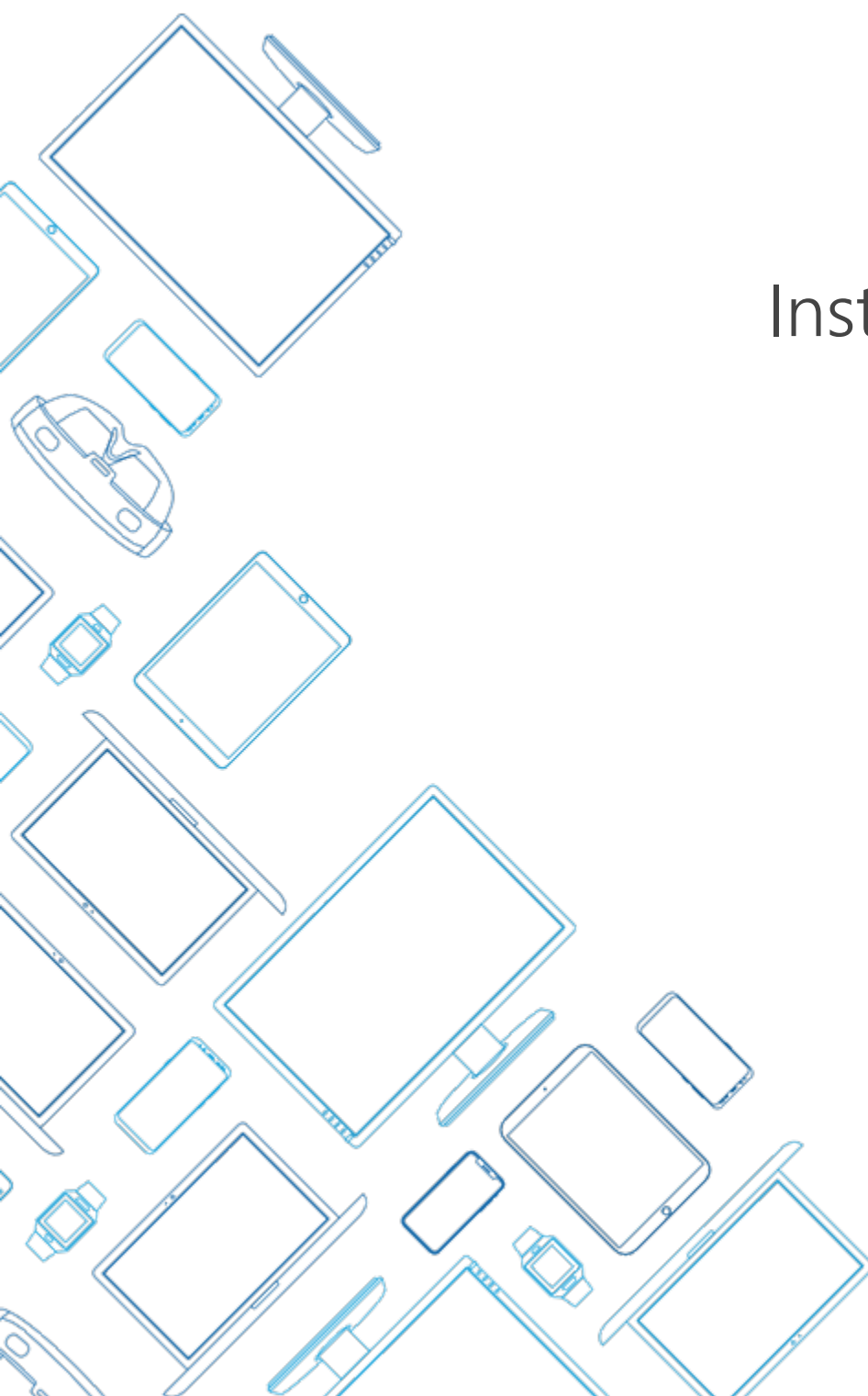


Make the Invisible Visible™



Installing IoTWorX on Azure IoT Edge

An ICONICS Whitepaper
October 2020



Contents

1	Introduction	4
2	Prerequisites	5
3	Installing an IoT Edge supported operating system.....	6
4	Creating and configuring Azure IoT Hub.....	7
5	Installing Azure IoT Edge and IoTWorX pre-requisites	8
6	Create an ICONICS Suite Azure VM.....	9
7	Create a new IoT Project in Workbench	10
8	Associating gateways to IoT project	12
9	Deploy edge modules	13
10	Verify deployment	14
11	Configuration Process Review	15
12	BACnet configuration.....	15
12.1	Channel configuration.....	15
12.2	Channel Nodes configuration	16
12.3	Deploy Channel configuration to gateway device(s)	16
12.4	BACnet device discovery.....	16
12.5	Deploy device discovery results to gateway device(s).....	17
13	Update Subscriber Connections.....	18
14	Publishing to the cloud	18
14.1	Defining a publish list.....	18
14.2	Defining a custom encoder/decoder	19
14.3	Assign a publish list to a publisher connection	20
14.4	Assign a publisher connection to a gateway.....	20
14.5	Deploy publish list configuration to gateway device(s)	21
15	Confirming data acquisition	21
15.1	Verify that the data is being sent to Azure	21
15.2	View a sample of the data being sent to Azure	22
16	Applying ICONICS licenses.....	23
16.1	Applying a license to ICONICS GENESIS64 virtual machine	23
16.2	Applying a license to IoTWorX gateway.....	23

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 © 2020, Microsoft Corporation and ICONICS, Inc. All rights reserved.

Authors

- Spyros Sakellariadis, Industry Innovation, Microsoft Corporation
- Zhi Wei Li, Director of Innovation and Incubation Solutions, ICONICS
- Rich Henderson, US QA Manager, ICONICS

1 Introduction

We have documented in other articles the reasons for, and process of, deploying an on-premises gateway to acquire data from building and environmental sensors, and there is no need to repeat that information in this article. We will refer to those articles, however, which can be downloaded from the following locations:

- [Using IoTWorX as a Gateway in Microsoft Azure Deployments](#), and will be referred to in this article as *Using IoTWorX*.
- [Writing to Devices from Microsoft Azure Using IoTWorX](#), referred to as *Writing to Devices*.
- [Sending Equipment Data to the Cloud](#), referred to as *Sending*.
- [Connecting a Delta VFD to Microsoft Azure using ICONICS IoTWorX](#), referred to as *Connecting a VFD*.

While we believe these steps above, and those following, are generally applicable for most users, results may vary depending upon a user's software and hardware systems.

The process of installing an on-premises gateway described in *Using IoTWorX* involves installing Microsoft Windows on a computer and then installing the ICONICS gateway software on that computer. Though installing and operating a few on-premises gateways as described is manageable, doing so for more than a few gateways is not optimal. An alternative solution is to deploy those gateways remotely on an Azure IoT Edge device and manage those gateways remotely. The following table summarizes the key differences between the two approaches:

Functionality	Gateway application running directly on Operating System	Gateway application running on IoT Edge
Managing BACnet module	RDP to on-premises computer running BACnet module. Requires inbound ports to gateway.	RDP or Bastion access to cloud VM to access Workbench. No inbound ports to gateway.
Managing multiple gateways	Each on-premises gateway managed one-at-a-time.	Single configuration interface for multiple gateways. Template based configuration, deployment, and management.
Running multiple gateway applications on same physical hardware	Requires gateway applications to be running in VMs.	Runs gateway applications in different containers.
Running multiple applications in addition to gateway application on same physical hardware	May require running in VMs	Runs in different containers
Running analytics on device telemetry	Limited to gateway application capabilities	Supplements gateway application capabilities with AI/ML functionality
Sending data to Azure IoT Hub	Supports publishing different sets of objects to different IoT Devices on an IoT Hub	Limited to publishing a single set of objects to a single IoT Edge Device on an IoT Hub

Managing an on-premises gateway deployed directly on Windows requires using the Remote Desktop client (RDP) and opening inbound IP port 3389 to each computer. This is regarded by many organizations as a security risk, and they may have a security policy that prohibits opening up inbound ports to any on-premises computers.

Installing IoTWorX on Azure IoT Edge

Even if an organization permits opening up the relevant inbound IP ports, this traditional method of deployment is a management challenge as each gateway needs to be accessed and configured separately – not just when being installed but also for any routine or unscheduled maintenance.

A solution to these problems is to deploy the gateway on the Azure IoT Edge platform, which runs on both Windows or Linux operating systems. Doing this allows deployment and management of the gateway software remotely without opening any inbound ports to the on-premises computer, as well as allowing multiple gateways to be managed as a single configuration. This article documents how to do this.

Deploying IoTWorX on IoT Edge involves several distinct steps:

1. Installing an IoT Edge supported operating system on the gateway computer
2. Creating and configuring Azure IoT Hub for an IoT Edge device
3. Installing Azure IoT Edge and IoTWorX pre-requisites on the gateway computer
4. Defining the IoT Edge device connection string on the gateway computer
5. Creating an ICONICS Suite Azure VM
6. Creating an IoT Project in ICONICS Workbench
7. Deploying IoTWorX modules to gateway computer
8. Verifying deployment

These steps are documented in the following sections.

2 Prerequisites

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

1. An on-premises computer. For sizing purposes, we have the following configuration deployed, polling about 10,000 data points every 5 minutes:
 - Hardware: Dell Edge 5000 Intel Atom CPU E3827 with 8 GB RAM
2. An Azure subscription
3. ICONICS IoTWorX and GENESIS64 licenses to cover the number of devices that will be polled

3 Installing an IoT Edge supported operating system

First, you will need to install an operating system and IoT Edge on an on-premises computer, such as that referenced in the Pre-requisites section above.

IoT Edge is supported on multiple operating systems. One of the most popular choice is Ubuntu version 18.04. Note that Ubuntu is not affiliated with MSFT or ICONICS. This paper is not intended to be an endorsement of any third-party products, either express or implied.

Ubuntu version 18.04 can be downloaded from the Ubuntu repository at <http://releases.ubuntu.com/18.04/>. The version used in the installation described in this paper is the “Desktop image for 64-bit PC (AMD64) computers (standard download)”.

For those unfamiliar with installing Ubuntu, there are many guides published on the Internet. To have a point of reference for this article at the time of writing, we followed broadly the steps outlined in the following publication:

<https://linuxtechlab.com/step-by-step-guide-to-install-ubuntu-18-04/>

Since this is simply a link to a live website, we cannot confirm that the content will be the same at the time anyone else accesses, nor that the configuration steps are suitable for production use – we are simply providing as an example that might be helpful.

As you create the Ubuntu desktop, we recommend using only lowercase for the computer name and username, as some of the remote access tools seem to expect lower case names.

Example: Ubuntu desktop name = “IoTWorXonEdge01”, username = “administrator”

4 Creating and configuring Azure IoT Hub

Azure IoT Hub is the Azure service needed to manage all IoT Edge devices. To create an Azure IoT hub in your Azure subscription, follow the instructions published here:

<https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-create-through-portal>

For the purposes of this article, you need only to follow these steps.

1. Login to the Azure portal.
2. Create a new IoT Hub of S1 tier or above that will receive published data. Give it a name.

Example: IoT Hub name = "ICONICS-IoTEdge-Hub"

3. Open the configuration blade of the IoT Hub.
4. Click on IoT Edge in the Automatic Device Management section in the list of options on the left.
5. In the IoT Edge devices blade, select 'Add an IoT Edge device' at the top to create a new device. Give it a unique name amongst all the devices on this hub.

Example: Device ID = "IoTEdgeDevice01".

6. Keep all other options as default.
7. Click on Save to create the new IoT Edge device.

Having set up the Hub and IoT Edge device you need to copy the information about how to access it:

1. Select the newly created IoT Edge device.
2. In the details page, look for **Primary Connection String**.
3. To the right of that field is a button to copy the connection string - click on the copy button to copy the connection string to the clipboard.
4. Store the connection string somewhere temporarily until it is needed in the subsequent sections.

Finally, you will also need information about the IoT Hub in general:

1. From the IoT Hub configuration blade, click on **Shared access policies**.
2. Click on **iothubowner**.
3. From the blade that opens on the right, look for **Connection string – primary key** field.
4. To the right of the field is a button to copy the connection string – click on the copy button to copy the connection string to the clipboard.
5. Store the connection string somewhere temporarily until it is needed in the subsequent sections.

5 Installing Azure IoT Edge and IoTWorX pre-requisites

The following article describes how to install Azure IoT Edge runtime on an Ubuntu desktop:

<https://docs.microsoft.com/en-us/azure/iot-edge/how-to-install-iot-edge-linux>

This is a complex process, and in addition, there are some IoTWorX files that need to be installed on the gateway. It is much easier to use a simple installation script provided by ICONICS, that installs Azure IoT Edge and sets up the necessary IoTWorX pre-requisites. The steps below walk you through the necessary steps.

1. Log on locally to the Ubuntu desktop.
2. Launch a browser window and navigate to the ICONICS IoTWorX website here:
<https://iconics.com/Products/IoTWorX>
3. Request a Trial of IoTWorX.
4. Download the IoTWorX ISO package to the desktop.
5. Extract the **SharedFolder_Setup** folder.
6. Open the file browser and browse to the **SharedFolder_Setup** folder. You should see a file called **IoTWorX_inst_amd64.sh** in the folder.
7. Right click on the empty space in the file browser and select **Open in Terminal**.
8. In the Terminal window, type in these commands (hit Enter between each command):

```
sudo chmod +x ./IoTWorX_inst_amd64.sh
sudo ./IoTWorX_inst_amd64.sh
```
9. The script will install Azure IoT Edge and set up the necessary IoTWorX pre-requisites.
10. During the installation of Azure IoT Edge, it will ask you for the device connection string and an SSL certificate.
 - a. For the device connection string, enter the **Primary Connection String** of the IoT Edge device created in the previous section.
 - b. For the certificate, hit Enter if you do not have one and the system will create a self-signed certificate.
11. The installation is complete when you see the message "**ICONICS IoTWorX sharedfolder package... done.**"

Once you have completed these steps, IoT Edge should be running on the computer. You can verify whether the installation has been successful with the following command in the Terminal window.

```
sudo iotedge list
```

The result list should show an **edgeAgent** with running status and uptime.

6 Create an ICONICS Suite Azure VM

Azure IoT Edge modules are designed for remote administration. For IoTWorX, we need to use the Workbench tool in ICONICS GENESIS64 to configure and administer IoTWorX gateways.

We can easily create a virtual machine in Azure that has ICONICS GENESIS64 by leveraging the ICONICS Suite virtual machine offer in Azure Marketplace.

To create an ICONICS Suite virtual machine, do the following:

1. Log on to Azure portal
2. Search for **ICONICS** from the search bar at the top
3. In the results, under Marketplace, select **ICONICS Suite 10.96.1**
4. In the offer page, click on **Create**
5. Enter the requested information to create an Azure VM

Example:

VM Size = Standard D4s v3 (4 vcpus, 16 GiB memory)

Operating System = Windows (Windows Server 2019 Datacenter)

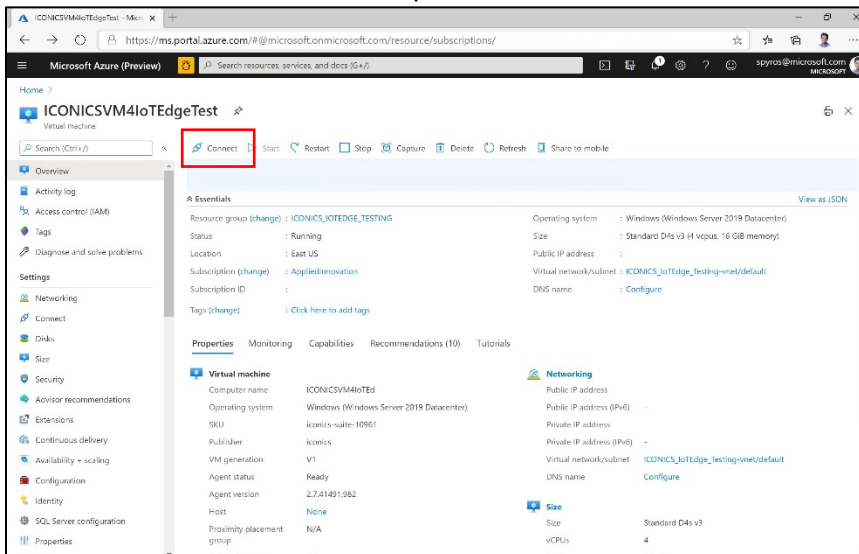
VM name = "ICONICSV4IoTEdgeTest"

Computer name = "ICONICSV4IoTEd"

Username = "IoTEdgeTester"

Once the VM is created, access the VM with RDP or Bastion. An easy way to do this is through the Azure portal:

1. Log on to Azure portal
2. Select the Virtual Machine created above
3. Click on the **Connect** icon in the top menu bar:



4. Select **RDP** or **Bastion** from the menu and enter the credentials you created above.

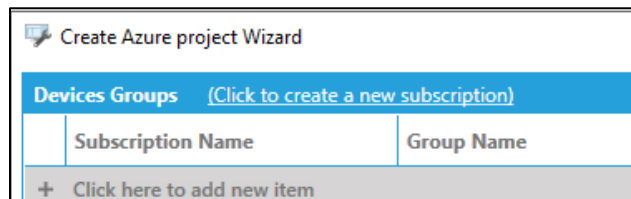
7 Create a new IoT Project in Workbench

Before we can start interacting with the IoTWorX gateway, we have to set up a new IoT project to communicate with IoT Hub which then allows us to interact with the IoTWorX gateway. Follow the steps to create a new IoT project.

1. Remote desktop to the ICONICS 10.96.1 VM created in Azure
2. Launch **Workbench** from the Start menu
3. Create a new IoT project with **File** → **New IoT Project**
4. Enter a project name

Example: Name = "IoTEdge on Ubuntu test".

5. Confirm **Template Version** as **10.96.1 (or any latest version)**
6. Enter a **Storage Connection**
 - a. Follow the instructions here to create a new storage account:
<https://docs.microsoft.com/en-us/azure/storage/common/storage-account-create?tabs=azure-portal>
 - b. In the storage account, select **Access keys** from the **Settings** section
 - c. Copy the **Connection string** field under **key1**
 - d. Enter it into the **Storage Connection** field of the IoT Project
7. Click **Next**
8. You will now configure the subscription to your IoT Hub:
 - a. From the **Devices Groups** header, click on the **Click to create a new subscription** link.

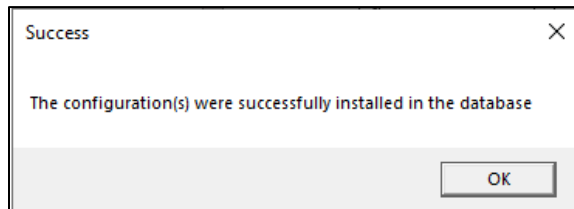


- b. Enter a subscription name, IoT Hub event hub endpoint and IoT Hub owner connection string.
 - c. Click **OK** and the dialog will close.
9. You will now configure the device grouping definition and storage information for the gateway(s) configuration file
 - a. Click in the **Subscription Name** column
 - b. Click on the drop down
 - c. Select the subscription connection you just created
 - d. Enter a desired **group name** and **storage folder** name
 - i. The group name is used to logically group your gateways

Example: Name = "Group1".



Installing IoTWorX on Azure IoT Edge

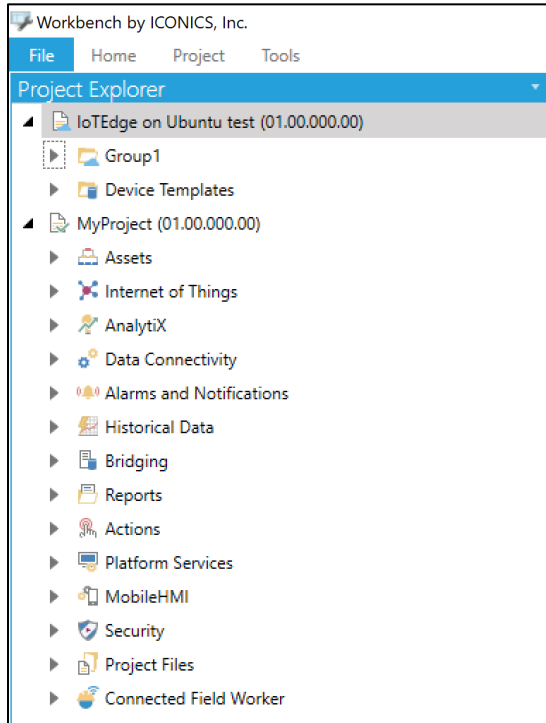
- ii. The storage folder is a storage container in Azure Blob Storage that is used to store the configurations that the gateway should download. If the storage folder doesn't exist, it will be automatically created
 - e. Hit Enter after entering the **Storage Folder** to commit the changes
 - f. Click **Next**
- 10. You will now configure a new local configuration database for the IoT project
 - a. Enter or select a local SQL Server
 - b. Select the appropriate authentication method and enter the necessary information
 - c. Enter a new database name
 - d. Uncheck **Create a backup of the database**
 - e. Click **Next**
 - f. Click **Next**
- 11. You will now decide which applications your IoT project should contain
 - a. Check off the **Install/Overwrite** checkbox for each application that you wish to configure and use in your gateways
 - b. You can right click in the column to get additional options
 - c. Click **OK**
- 12. The information you entered is now used to create the following:
 - a. IoT Hub subscription
 - b. Storage folder
 - c. Local configuration database
 - d. Application configuration definition in the configuration database
- 13. Upon successful creation of the above, a confirmation dialog appears like so



- 14. Click **OK** on the success dialog to close

8 Associating gateways to IoT project

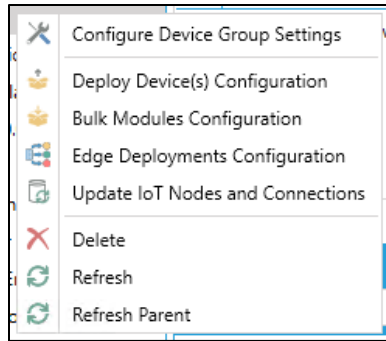
Upon successful IoT project creation, your workbench project explorer should now have at least 2 root items, a standard GENESIS64 project identified with the  icon, and an IoT project identified with the  icon. In our example, the GENESIS64 project is labeled “MyProject” and the IoT project labeled “IoTEdge on Ubuntu Test”:



To interact with the gateway(s), we must first associate the gateway(s) to an appropriate device group in our IoT project.

1. Expand the IoT project (e.g. IoTEdge on Ubuntu test) and you should see a device group with your chosen name (e.g. Group1), and a **Device Templates** folder
2. Right click on the device group
3. Select **Configure Device Group Settings**
4. Scroll to the **Registry Management** section
5. Check the **In Group** column checkbox for the gateway that you want to include in this device group
6. Expand the device group and you should see a list of gateways that are connected to the IoT Hub
 - a. Devices with a green dot indicate they are online
 - b. All devices in a device group is associated with the same device template
7. If the **IoT Hub connection**, **Storage Folder connection** or **Device Template** association needs to be edited, right click on the device group and choose **Configure Device Group Settings**

Installing IoTWorX on Azure IoT Edge



8. Later, we will also deploy device configurations to gateways via this same right click menu

9 Deploy edge modules

Before we can use IoTWorX on the gateway, we must first deploy the necessary edge modules. We would use the Workbench tool in ICONICS GENESIS64 to deploy the necessary modules.

1. Expand the IoT project in **Workbench**
2. Expand the device group to see your list of gateways
3. Right click on the gateway and select **Configure Modules**
4. On the dialog that comes up, you will see the status of any Edge modules running on the device.
5. In the **Available Modules** header, click on the **Click to configure modules** link
6. Under the **Deployment Options** section, selection an option from the **Version** and **Architecture** dropdowns that are applicable to your deployment.
7. Check the **Deploy** checkbox for each ICONICS module you would like to add to your IoT device.
Note: The minimum required modules are **FrameWorX Server**, **IoT Publisher** and **Ico Edge Agent**.
8. Click **Deploy**. You should see a **Deploy successfully completed** message, indicating that a new deployment has been queued for the device. Click **Close**.
9. At this point, ICONICS modules will begin downloading to the IoT device. Depending on the speed of your internet, this may take some time (ICONICS modules total about 1.6 GB).
10. After modules have finished downloading and have started on the device, if you refresh your device group in **Workbench**, you will see that the device has a green status icon indicating that it is online and running.

10 Verify deployment

The easiest way to verify that the edge modules were successfully deployed is with the green status icon of each gateway in the Workbench IoT project device group.

The state of each container in the gateway can also be checked by logging in to the Ubuntu box and run this command in a Terminal window:

sudo iotedge list

If you have logged on as administrator to a computer with the name IoTWorXonEdge01, you should see something similar to the following (your version and config path may differ):

```
administrator@IoTWorXonEdge01:~$ sudo iotedge list
```

NAME	STATUS	DESCRIPTION	CONFIG
edgeAgent	running	Up 20 hours	mcr.microsoft.com/azureiotedge-agent:1.0.8.2
IoTPublisher	running	Up 6 hours	icotechpreview/iotpublisher:10.96.1TP1-amd64
BACnet	running	Up 6 hours	icotechpreview/bacnet:10.96.1TP1-amd64
edgeHub	running	Up 20 hours	mcr.microsoft.com/azureiotedge-hub:1.0.8.2
IoTVisualizer	running	Up 6 hours	icotechpreview/visualizer:10.96.1TP1-amd64
FwxServer	running	Up 6 hours	icotechpreview/fwxserver:10.96.1TP1-amd64
IcoEdgeAgent	running	Up 6 hours	icotechpreview/icoedgeagent:10.96.1TP1-amd64

11 Configuration Process Review

The configuration process for IoTWorX is slightly different than standard GENESIS64.

The primary difference is that IoTWorX uses a device template approach to configuration and deployment of configurations to gateways. This means most of our configuration activities will be in the Device Templates folder and then deployed to related gateways.

Additionally, due to the nature of IoT, every configuration change that needs to be applied to the gateway must first be packaged, stored in the Storage Folder and the gateway will pick up the new configuration and then apply it locally. If a configuration sequence requires the gateway to have certain configuration settings before the next step, a configuration deployment must first be performed before the next step is taken.

12 BACnet configuration

The BACnet module in IoTWorX allows for the configuration of multiple gateways from a single interface and then deploy configurations specific to each gateway. Two important configuration structures make this possible, **Channels** and **Channel Nodes**.

A **Channel** represents an isolated communication path and is used to segregate different BACnet networks and devices.

A **Channel Node** defines the relationship between each gateway and one or more **Channels** it should use to communicate with the relevant BACnet networks and devices.

The overall configuration of BACnet is going to be split into two parts:

- The first part is to configure the channel with the appropriate foreign device definition if necessary. A foreign device definition allows the gateway to discover and connect to devices in a different subnet.
- The second part is to discover the devices and its associated objects then adding them into our configuration.

In between the two parts is a necessary configuration deployment such that the gateway has the channel information before we attempt to do a device discovery.

12.1 Channel configuration

Each gateway device has a **Default Channel** created as part of initial deployment. If the gateway only needs to communicate with a single BACnet network, the default channel can be used. If the gateway needs to communicate with multiple BACnet networks, then a channel should be created for each BACnet network.

To configure a channel, do the following:

1. Expand the IoT project → **Device Templates** → **Default Template** → **Data Connectivity** → **BACnet** → **Channels**
2. Right click on a channel, i.e. **Default Channel**
3. Select **Edit**
4. In the **General** tab, if you wish to assign a specific BACnet Device ID to the gateway, enter it in the **Local Device ID** field. Otherwise, the gateway will generate a random BACnet device ID
5. In the **Port** tab, review the default settings

Installing IoTWorX on Azure IoT Edge

6. In the **Foreign Devices** tab, if the gateway needs to communicate with a different subnet than the subnet it is in, do the following:
 - a. Check **Enabled**
 - b. In the **Foreign Devices** list, click on **Click here to add new item**, this adds a new foreign device entry in the list
 - c. Click on the **IP Address** column of the newly entered item
 - d. Enter the IP address of the BBMD in the different subnet and confirm the port the BBMD is communicating at
7. Click **Apply** to confirm the changes in the channel

12.2 Channel Nodes configuration

To ensure each gateway is assigned the correct channel and associated device configuration, we need to first define an association between a gateway device and channel. Follow the steps below to associate a gateway device and channel.

1. Expand the IoT project → **Device Templates** → **Default Template** → **Data Connectivity** → **BACnet**
2. Right click on **Channel Nodes**
3. Select **Edit**
4. Click on **Click here to add new item**
5. Enter the name of the gateway device in the **Machine Name** column
6. Click in the **Channel** column and select the appropriate channel in the drop down
7. Click away from the row to commit the changes
8. Click on **Apply** to save your configuration

12.3 Deploy Channel configuration to gateway device(s)

The channel configuration changes need to be deployed to the gateway device before it can be used for further device discovery. To deploy the channel configuration to the respective gateway device, perform the following steps:

1. Expand your IoT project
2. Right click on the device group that contains the gateway you wish to deploy the updated device template configuration
3. Select **Deploy Device(s) Configuration**
4. Check off the device(s) where the updated device template configuration should be deployed
5. Click **OK**
6. This creates a task to generate the necessary configuration package for the selected gateway device(s)
7. The progress of the configuration package generation can be monitored in the **Recent Tasks** panel in Workbench
8. Once the configuration package generation is complete, it will be sent to the **Storage Folder**. The gateway device(s) will then download the configuration package and apply it.

12.4 BACnet device discovery

After the channel configuration is deployed to the gateway device, we are now ready to perform network discovery of BACnet devices. Perform the steps below to discover devices:

Installing IoTWorX on Azure IoT Edge

1. Expand your IoT project → **Device Templates** → **Default Template** → **Data Connectivity** → **BACnet** → **Channels**
2. Right click on **Default Channel**
3. Select **Network Discovery**
4. A list of Available Gateways shows up
5. Select the gateway you wish to use to perform the network discovery
6. Click **Ok**
7. In the next **Discover BACnet Devices** dialog, click on **Start Discovery**
8. This sends a command to the selected gateway to initiate a BACnet discovery process
9. Once the discovery is complete, you will see a list of BACnet devices the gateway discovered
10. Uncheck any device that should not be included in your configuration
 - a. Right clicking on the list provides additional options
11. Click **Next** at the bottom to move on to the **Object Scan** dialog
12. Click on the **Start Scan** button to initiate the object scan process of the selected device(s)
13. Once the scan is complete, the list of objects from the device(s) will show up
14. By default, all discovered objects are selected. Uncheck any object you do not wish to include.
15. Click on **Finish** to trigger a task to add the device and object to your configuration
16. Once the task is complete, you should see the added BACnet device(s) and objects by expanding the appropriate channel, i.e. **Default Channel**

12.5 Deploy device discovery results to gateway device(s)

To update the gateway device configuration to reflect the discovered BACnet device(s) and objects, perform the following steps:

1. Expand your IoT project
2. Right click on the device group that contains the gateway you wish to deploy the updated device template configuration
3. Select **Deploy Device(s) Configuration**
4. Check off the device(s) where the updated device template configuration should be deployed
5. Click **OK**
6. This creates a task to generate the necessary configuration package for the selected gateway device(s)
7. The progress of the configuration package generation can be monitored in the **Recent Tasks** panel in Workbench
8. Once the configuration package generation is complete, it will be sent to the Storage Folder. The gateway device(s) will then download the configuration package and apply it.

13 Update Subscriber Connections

To improve the browsing experience when defining the publish list later, we need to tweak some settings of the subscriber connection. Perform the following steps:

1. Expand your GENESIS64 project → **Internet of Things** → **Subscriber Connections**
2. Right click on the subscriber connection
3. Select **Edit**
4. In the **General Settings** section, change the **Browse Timeout** to 1 Minute(s)

14 Publishing to the cloud

Now that the gateway device is updated with the necessary BACnet configuration, we can define a publish list to select the object attributes that should be continuously published to the cloud. We will first define the publish list and then associate the publish list to the appropriate gateway device.

14.1 Defining a publish list

1. Expand your IoT project → **Device Templates** → **Default Template** → **Internet of Things**
2. Right click on **Publish Lists**
3. Select **Add Publish List**
4. Enter a name for the new publish list
5. In the **Default Collection Group** field, click on the '+' sign to the right of it to create a new collection group
6. Click on the **Published Points** tab to start adding points to be published
7. Click on the **Click here to add multiple tags** link at the top
8. This brings up the data browser
9. Expand My Computer → **Internet of Things** → Subscriber Connection → Gateway → **All Available Data** → **My Computer** → **Data Connectivity** → **BACnet** → **Default Channel**
10. Expand the desired BACnet device, object
11. Select the desired attribute to be published, i.e. presentValue
 - a. Hold the **Ctrl** key and multi-select other attributes to be published
12. Click **OK** to add the select list of data points to be published
13. Click **Apply**

Installing IoTWorX on Azure IoT Edge

14.2 Defining a custom encoder/decoder

ICONICS IoTWorX supports publishing data with either the standard ICONICS binary or JSON format, or with your own custom JSON format. If you intend to consume the published data using a non-ICONICS client, you may want to use your own custom JSON format.

Note that while a custom JSON format can be used to publish data, for version 10.96.1, data browsing from an ICONICS client will stop working. This limitation may be improved in newer versions.

To publish data with a custom JSON format, you have to first define a custom encoder with the following steps:

1. Expand your IoT project → **Device Templates** → **Default Template** → **Internet of Things**
2. Right click on **Custom Encoders/Decoders**
3. Select **Add Encoder/Decoder**
4. Give this encoder a name
5. In the **General Settings** section, for the **Plugin** field, select **CustomJson**
6. In the **Message Type** field, choose the option that best meets your needs
7. In the **Value Format** field, you can click on the **Set default format** to get a starting point of defining your encoder format
8. Click on **Add keyword** to see the list of available keywords for data that can be published
9. Select your desired keyword and click **OK** to add it to your encoder format
10. Scroll down on the encoder form to view the **Write Format**
11. Check off **Use different format for writing** if you want to customize the write format
12. Click **Apply** when done

Here is an example of a simple custom JSON format:

General Settings

Plugin: CustomJson

Message Type: One value for each message

Value Format:
[\(Add keyword\)](#)
[\(Set default format\)](#)
[\(Auto indent\)](#)

```
{
  "gwy": "%DEVICENAME%",
  "name": "%PUBLISHNAME%",
  "value": "%VALUE%",
  "timestamp": "%NOWUTC.TEXT%",
  "status": "%STATUS.GOOD%"
}
```

Message Format:
[\(Add keyword\)](#)
[\(Set default format\)](#)
[\(Auto indent\)](#)

```
{
  "gwy": "%DEVICENAME%",
  "name": "%PUBLISHNAME%",
  "value": "%VALUE%",
  "timestamp": "%NOWUTC.TEXT%",
  "status": "%STATUS.GOOD%"
}
```

Use different format for writing

Write Format:
[\(Add keyword\)](#)
[\(Set default format\)](#)
[\(Auto indent\)](#)

```
{
  "id": "%PUBLISHNAME%",
  "v": "%VALUE%",
}
```

14.3 Assign a publish list to a publisher connection

Before data tags defined in a publish list can be published, it must be assigned to a publisher connection, which defines how the publishing is done. Each gateway in the device group has a publisher connection created for it. So, all we have to do is assign the desired publish list to the publisher connection for the gateway.

To assign a publish list to a publisher connection, perform the following steps:

1. Expand your IoT project → **Device Templates** → **Default Template** → **Internet of Things** → **Publisher Connections**
2. Right click on the desired gateway
3. Select **Edit**
4. In the **General Settings** section, look for the **Publish List** field
5. Click on the drop down
6. Select the desired publish list
7. Click **Apply**

If a custom encoder is to be used, do the following additional steps:

1. Uncheck **Enable compatibility with ICONICS clients**
2. In the **Encoder** drop down, select your custom encoder
3. Click **Apply**

Note that if a custom encoder is used to publish data, as of version 10.96.1, data browsing from ICONICS clients will stop working. As such, it is advised that publishing data with a custom encoder be done after the configuration of the publish list is complete.

The data browsing ability can be re-enabled by checking the **Enable compatibility with ICONICS clients** checkbox and re-deploying the device configuration according to the instructions in Section 12.5 above.

14.4 Assign a publisher connection to a gateway

To instruct a gateway to use a particular publisher connection (which then associates the publish list), a node definition has to be made to relate the gateway to the publisher connection. By default, each gateway in the device group already has a node definition associated to its default publisher connection. IoTWorX supports the ability to define multiple publisher connections for a gateway by defining multiple node definitions for the same gateway, with each definition associated to a different publisher connection.

To associate a publisher connection to a gateway, perform the following steps:

1. Expand your IoT project → **Device Templates** → **Default Template** → **Internet of Things** → **Nodes**
2. Right click on Publisher Nodes
3. Select **Edit**
4. Enter the gateway name in the **Machine Name** column and the desired publisher connection name in the **Publisher Connection** column
5. Click **Apply**

Installing IoTWorX on Azure IoT Edge

14.5 Deploy publish list configuration to gateway device(s)

To update the gateway device configuration with the assigned publish list, perform the following steps:

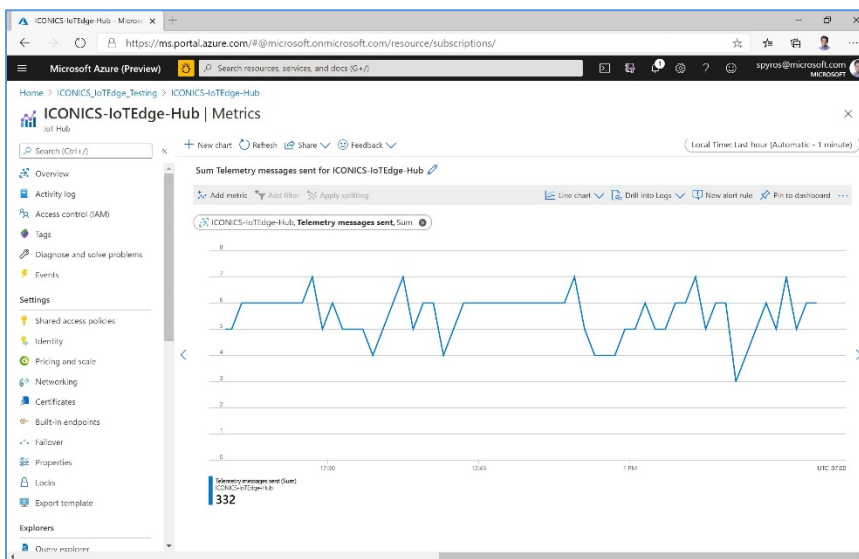
1. Expand your IoT project
2. Right click on the device group that contains the gateway you wish to deploy the updated device template configuration
3. Select **Deploy Device(s) Configuration**
4. Check off the device(s) where the updated device template configuration should be deployed
5. Click **OK**
6. This creates a task to generate the necessary configuration package for the selected gateway device(s)
7. The progress of the configuration package generation can be monitored in the **Recent Tasks** panel in Workbench
8. Once the configuration package generation is complete, it will be sent to the Storage Folder. The gateway device(s) will then download the configuration package and apply it.

15 Confirming data acquisition

You can easily check the data that is coming into Azure IoT Hub or view a sample of the data as it comes into Azure.

15.1 Verify that the data is being sent to Azure

1. Open the Azure portal and navigate to the IoT Hub
2. In the Monitoring section, click on Metrics.
3. In the filter bar at the top, pick “Telemetry Messages Sent” from the drop-down list.
4. If data is being sent from the gateway this should be visible in the chart. Since we configured messages to be sent every 5 minutes the chart should reflect batches of activity.

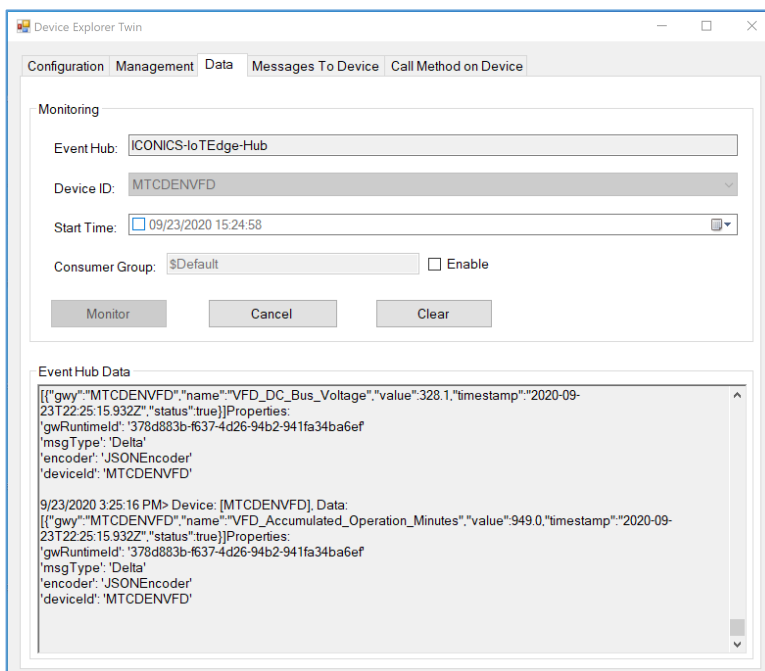


Installing IoTWorX on Azure IoT Edge

15.2 View a sample of the data being sent to Azure

1. Launch an application such as Azure Device Explorer, available as an [MSI](#) or as source code from [GitHub Device Explorer repo](#).
2. In the Configuration tab, copy the IoT Hub connection string of the IoT Hub you got in Section 2.3 into the IoT Hub Connection String field, and click Update.
3. In the Management tab, click List.
4. In the Data tab, select the IoTWorX gateway in the Device ID drop-down list.

Click Monitor to see the data as it comes in to IoT Hub. If you have set a custom encoder to send individual messages, the data stream should look something like that shown in the figure below, depending upon the format in the encoder. (If you selected “Enable compatibility with ICONICS clients” in the publisher, the data is in a binary format and you will not see these records.)



16 Applying ICONICS licenses

The ICONICS software in the virtual machine and IoTWorX software on the gateway both come up demo licenses that expire after 12 hours. To use these components for longer than the demo period, a valid license from ICONICS must be applied to the respective components. The following steps describe the process of applying the necessary licenses to the respective components.

16.1 Applying a license to ICONICS GENESIS64 virtual machine

1. Contact ICONICS to purchase the appropriate ICONICS GENESIS64 license. ICONICS' contact information can be found here: <https://iconics.com/About/Contact-Us>
2. You will receive an email with your purchased license information that contains a Product Registration number and Customer Key for each product purchased.
3. Go to <https://licensing.iconics.com>
4. On the ICONICS licensing website, create a new account with the available link or log in with your existing ICONICS licensing account.
5. Once logged in, you should see 3 options, Software, Hardware and Cloud.
6. Select the Cloud option as we are running ICONICS software in the cloud.
7. From the top menu, select New License.
8. Enter the Product Registration number and Customer Key from the email you received for the products to be licensed on the virtual machine.
9. Click Next.
10. Choose an existing end user or enter a new end user information.
11. Click Next.
12. You should now see a list of products that are available for you to license.
13. Check off the products that you want to license.
14. Click Next.
15. Review the summary of products that you are creating a license for.
16. When ready, click on Generate Key.
17. Once the key is generated, you should see a License Pool field with a string of characters.
18. Copy the string of characters next to License Pool.
19. Access the ICONICS virtual machine that the license should be applied to.
20. From the Windows Start menu, search, or look for Platform Services Configuration.
21. In the configuration dialog, select the License tab.
22. Select the radio button for Cloud License.
23. Paste the license pool character string copied before.
24. Leave the password field blank.
25. Click Apply.
26. Restart the ICONICS License Service service.

16.2 Applying a license to IoTWorX gateway

1. Contact ICONICS to purchase the appropriate ICONICS GENESIS64 license. ICONICS' contact information can be found here: <https://iconics.com/About/Contact-Us>
2. You will receive an email with your purchased license information that contains a Product Registration number and Customer Key for each IoTWorX product purchased.
3. Go to <https://licensing.iconics.com>

Installing IoTWorX on Azure IoT Edge

4. On the ICONICS licensing website, create a new account with the available link or log in with your existing ICONICS licensing account.
5. Once logged in, you should see 3 options, Software, Hardware and Cloud.
6. Select the Cloud option as we are running ICONICS software in the cloud.
7. From the top menu, select New License.
8. Enter the Product Registration number and Customer Key from the email you received for the IoTWorX product.
9. Click Next.
10. Choose an existing end user or enter a new end user information.
11. Click Next.
12. You should now see a list of products that are available for you to license.
13. Check off the products that you want to license.
14. Click Next.
15. Review the summary of products that you are creating a license for.
16. When ready, click on Generate Key.
17. Once the key is generated, you should see a License Pool field with a string of characters.
18. Copy the string of characters next to License Pool.
19. Launch Workbench in the ICONICS virtual machine used to manage the IoTWorX gateways.
20. Expand the IoT project → Device group.
21. Right click on the gateway.
22. Select Configure Application(s) settings.
23. In the General Settings section, check off the Override Cloud License Pool option.
24. Paste the license pool character string copied before.
25. Click Apply.
26. Right click on the gateway with the license.
27. Select Deploy Device(s) Configuration.



Founded in 1986, ICONICS is an award-winning independent 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

© 2020 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, BizViz, PortalWorX, MobileHMI and their respective modules, OPC-to-the-Core, and Visualize Your Enterprise 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

