Voice of the Machine Edge

How to Get Started with Voice of the Machine Edge

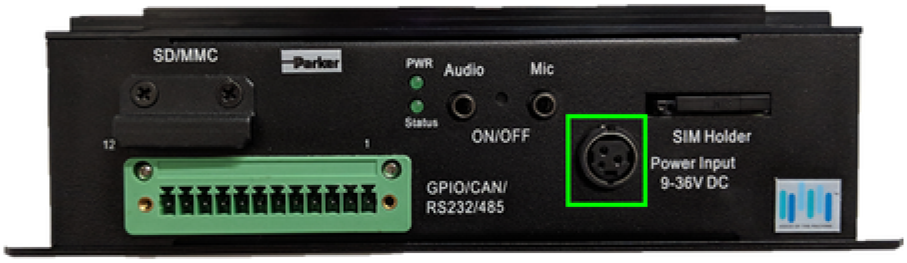
Voice of the Machine Edge is a software that can be installed on any type of industrial computer or IoT gateway to connect, process, and manage SensoGold data. The edge level solution offers pre-processing, filtering and ability to analyze data at the edge level, instead of only at the cloud, which helps to cut costs by only sending useful, synthesized data to the cloud.

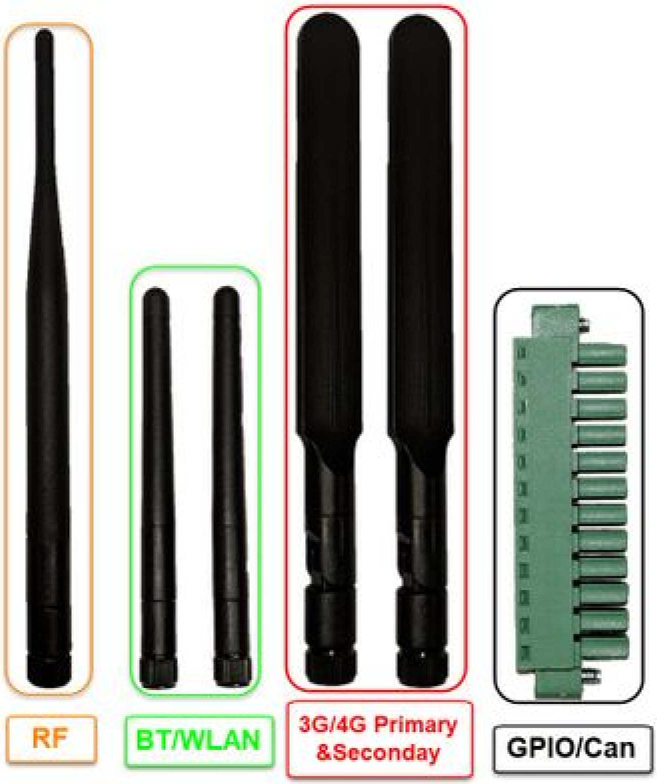
Simply stated, Voice of the Machine Edge has these main functions:

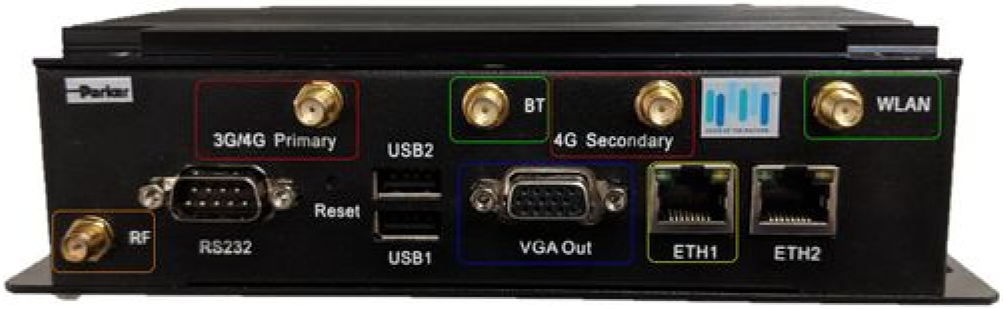
- Enable secure data collection from any industrial device.
- Send data securely to the cloud.
- Run applications on top of the data.
- Manage devices.

Included in the box

- 1- Vantron Gateway
- 1- Power Cord
- 2- 3G/4G Primary and Secondary antennas
- 2- BT/WLAN antennas
- 1- RF antenna
- 1- GPIO/CAN



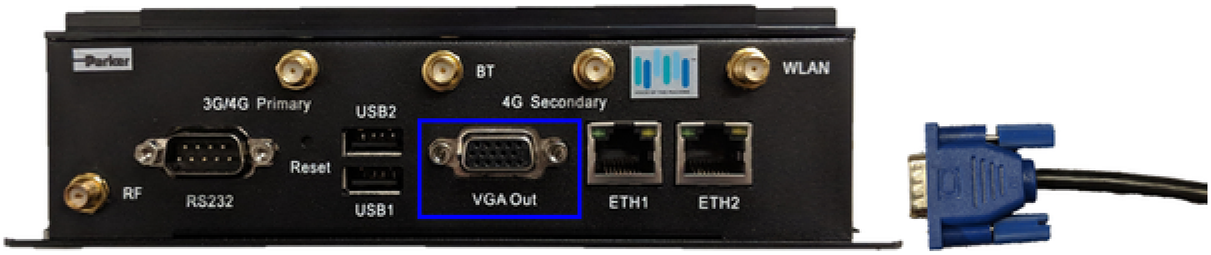




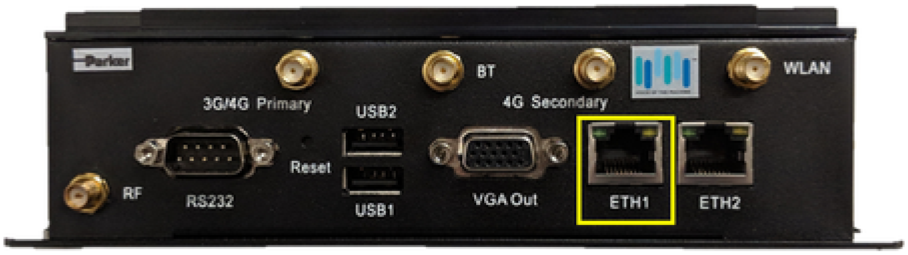
Step 1: Set up Voice of the Machine Gateway

If the system is running Voice of the Machine Edge, it can be discovered using TUI (Terminal User Interface) access.

1. Connect a monitor to the gateway (VGA Cable not included)

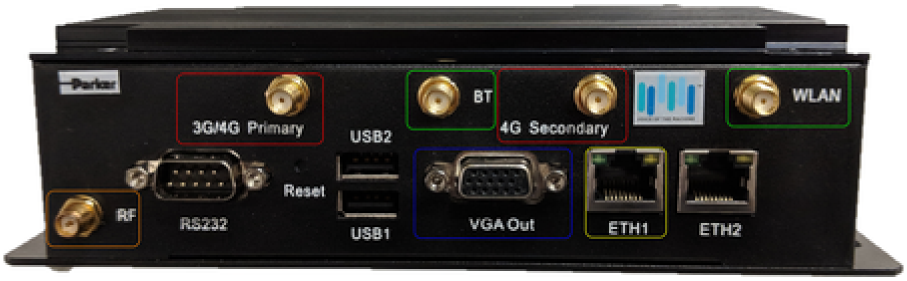


2. Connect to a network. **Always ETH1** (Ethernet Cable not included)

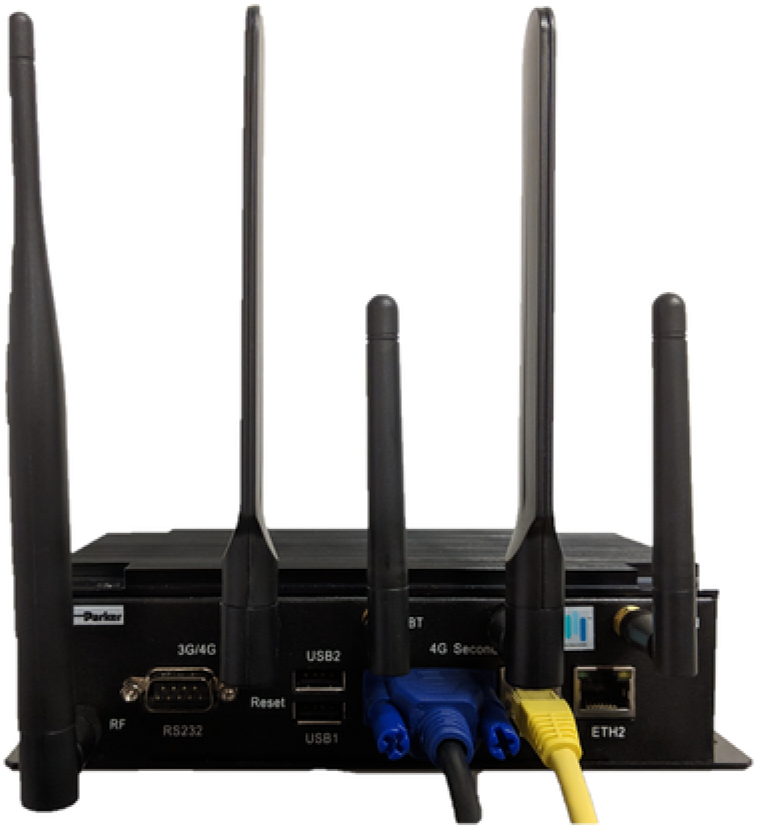


Make sure the network is **not** behind a company firewall as this will not assign an IP address to the device.

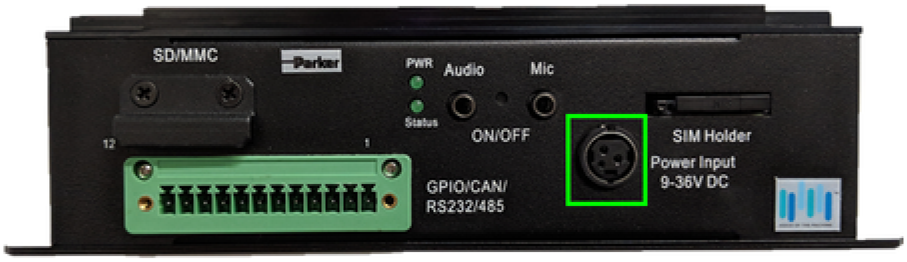
3. Connect all antennas. (Refer to the picture above with antenna labels)



5. The gateway should look like the picture below after you complete steps 1-3

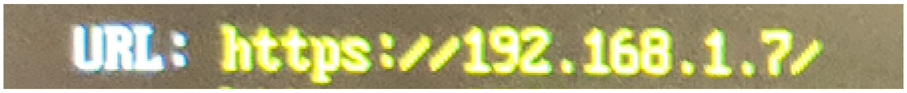


6. Connect the power cable to gateway and an outlet. (Flat side facing the bottom of the device)



Step 2: Obtain the IP address

The gateway will power on and you should hear a long beep. On the screen you will see the TUI and it will display the IP address the gateway was assigned too. (The first line of URL)

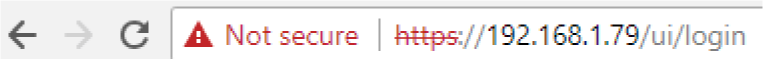


```
URL: https://192.168.1.7/
```

Step 3: Login

Prerequisites
Internet browser must be connected to the same network as the Voice of the Machine Edge. Google Chrome, Mozilla Firefox and Safari recommended.

1. Once the IP address is obtained, navigate to it in order to get started with Voice of the Machine Edge.

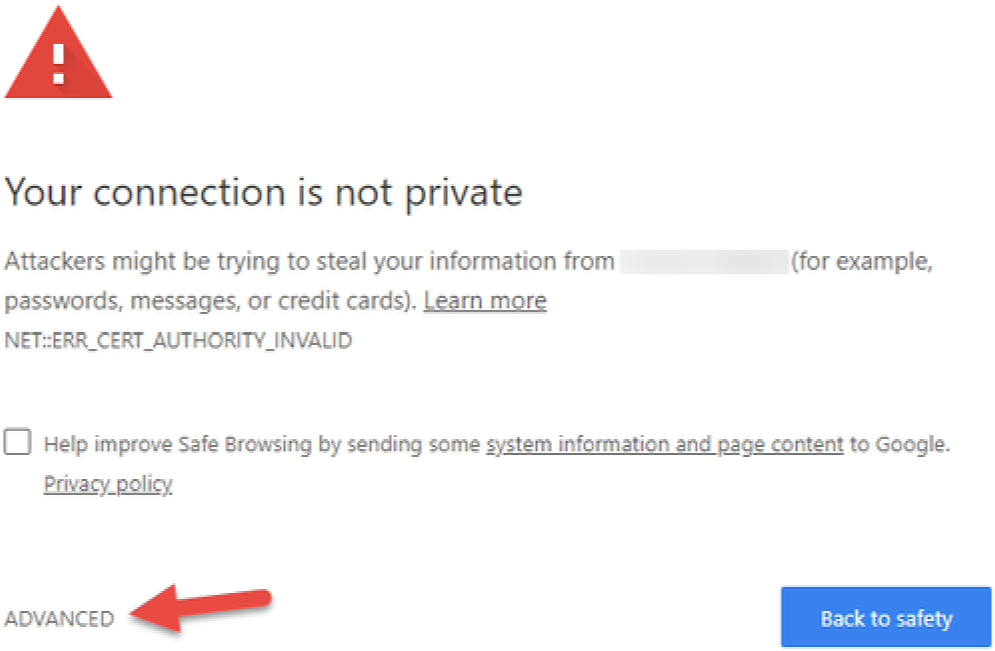


Not secure | ~~https~~://192.168.1.79/ui/login

The browser may not connect immediately due to its security settings.

(Chrome Only Instructions. If using another browser refer to How to Get Around Browser Security Settings)

2. If you get the alert below click **Advanced**



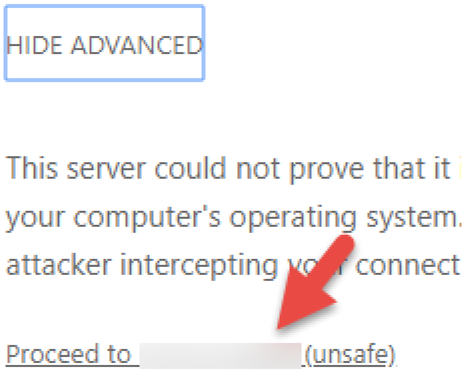
Your connection is not private

Attackers might be trying to steal your information from [redacted] (for example, passwords, messages, or credit cards). Learn more

NET::ERR_CERT_AUTHORITY_INVALID

□ Help improve Safe Browsing by sending some system information and page content to Google. Privacy policy

3. Click **Proceed to [IP Address](unsafe)**



This server could not prove that it
your computer's operating system.
attacker intercepting your connect

Proceed to [redacted] (unsafe)

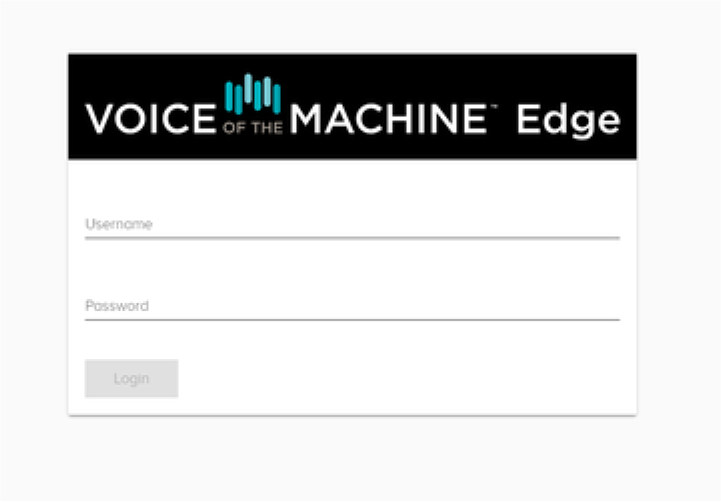
4. Provide the default details.

Password has been changed during initial set up at Parker.

Refer to the note in your box or contact your Parker representative.

Username: **admin**

Password: ***Unique*** Refer to note in box



What's Next?

Connect the PRN and Sensors to the Gateway

Related articles

- Voice of the Machine Edge
- How to Activate an Offline License Key
- How to Complete a SensoNODE Gold Site Survey (Video)
- How to Use Terminal User Interface with Voice of the Machine Edge
- How to Read Sensor Data String Outputs

How to Use Terminal User Interface with Voice of the Machine Edge

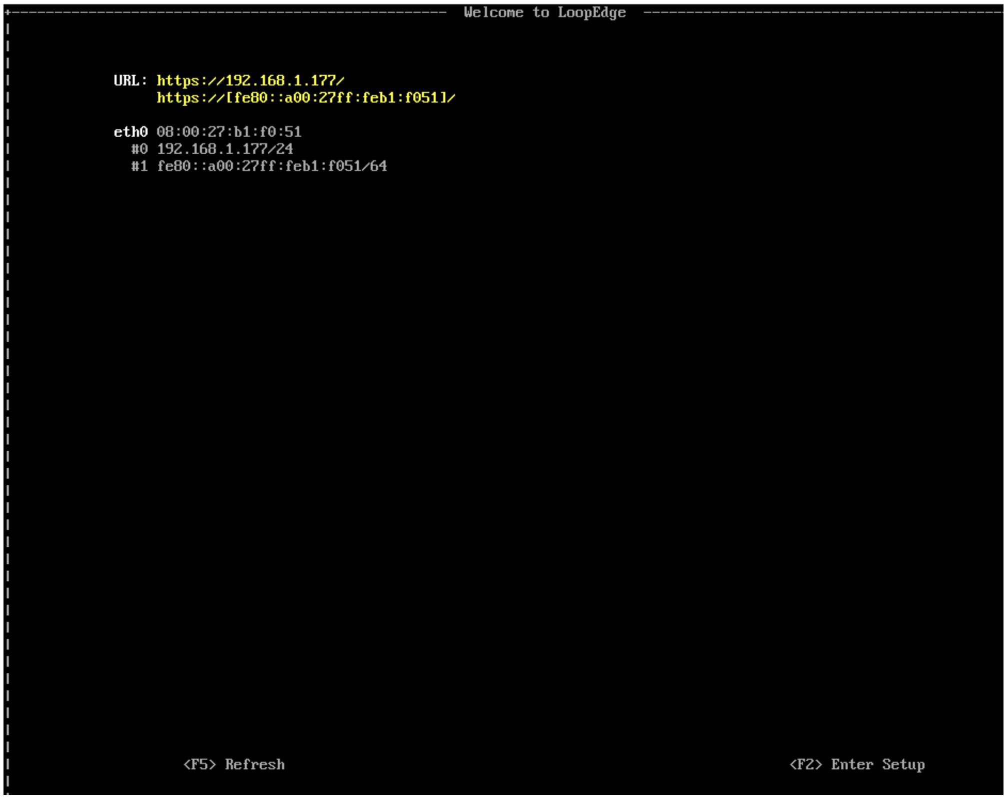
Connect a Keyboard and Monitor

Gateway devices have rear-panel connectors for a monitor and a keyboard, to enable access to the configuration options of the installed Voice of the Machine Edge

1. Connect a keyboard (for example, a USB connection)
2. Connect a monitor (for example, an HDMI, Micro HDMI, or VGA connection)

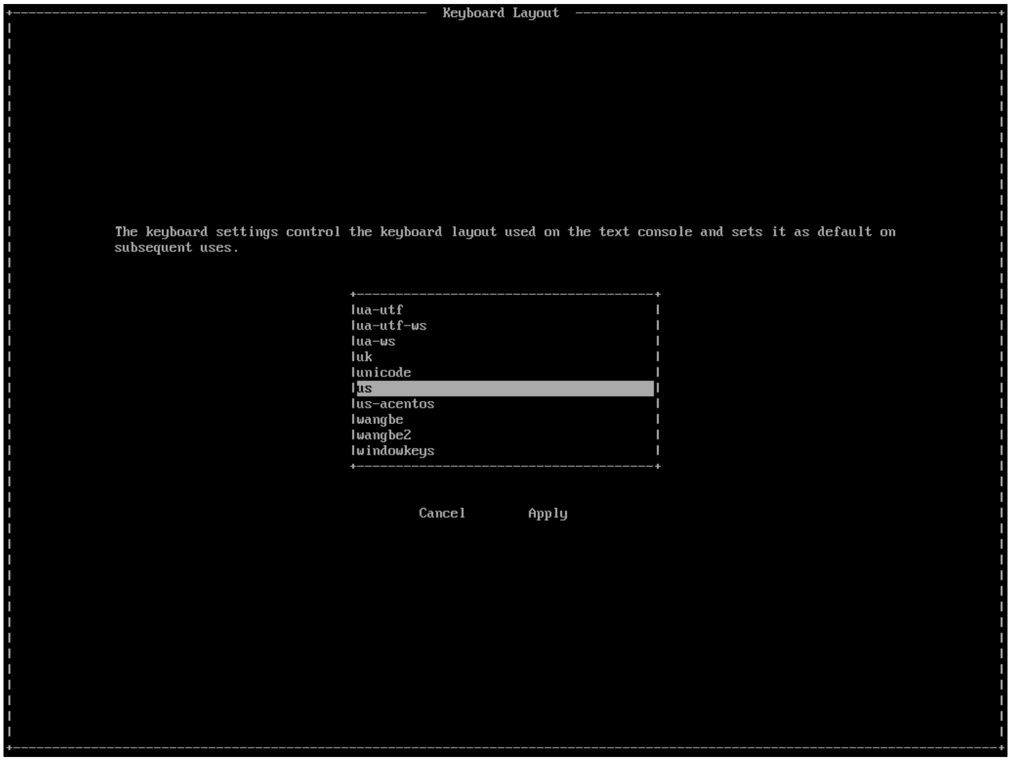
Log in after Initial Bootup

Once the system has finished booting up, the TUI welcome screen will display the IP addresses and network interface settings.
The eth0 Ethernet interface is required for first-time bootup.



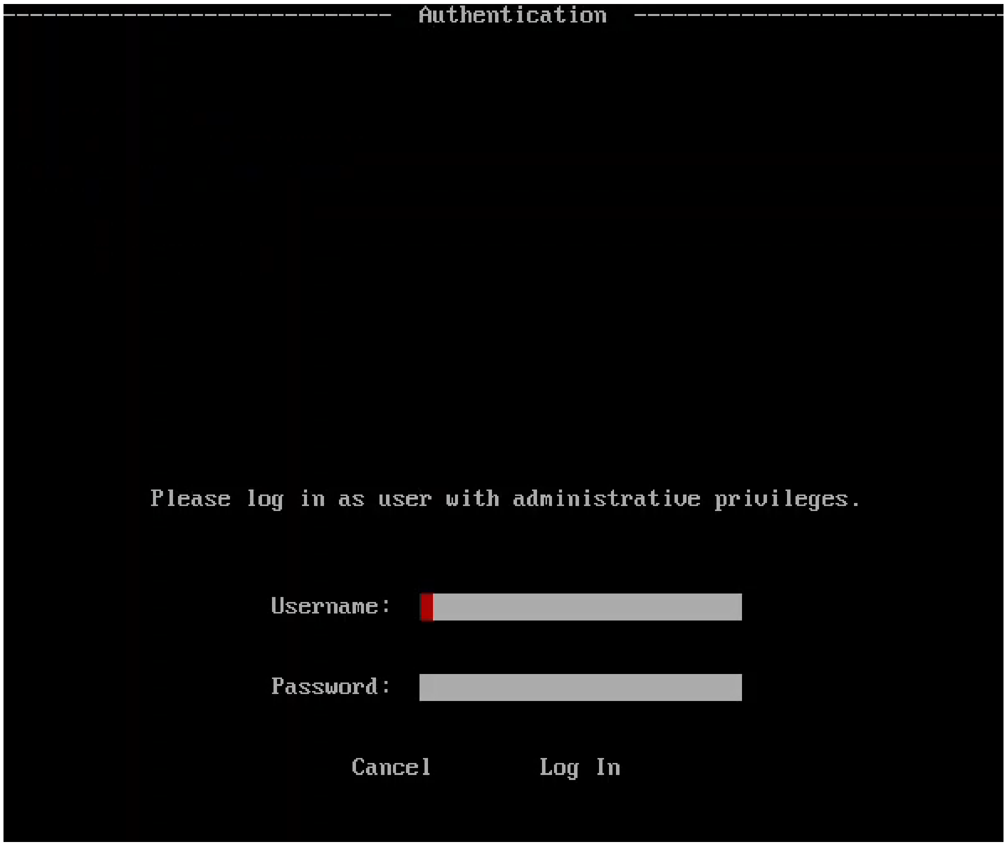
```
Welcome to LoopEdge

URL: https://192.168.1.177/
     https://[fe80::a00:27ff:feb1:f051]/

eth0 08:00:27:b1:f0:51
  #0 192.168.1.177/24
  #1 fe80::a00:27ff:feb1:f051/64

<F5> Refresh                                        <F2> Enter Setup
```

1. Press **F2** to enter setup.
2. If this is your first time logging in, you will have to choose a keyboard layout. The **US** layout is sufficient for most English keyboards.



The keyboard settings control the keyboard layout used on the text console and sets it as default on subsequent uses.

3. Press **Enter** to **Apply** the layout and display the login screen.
   - If this is the first time anyone is logging in to Voice of the Machine Edge, the credentials will be provided by Parker.



Please log in as user with administrative privileges.

4. **Tab** to the Username field.
5. **Tab** to move from one field to the next to enter credentials.
6. **Tab** to **Log In** and press **Enter**.

Navigate the TUI

- The TUI will log you out if no activity occurs within a few minutes.
- Press **F2** to log in again.

The terminal user interface is limited to keyboard entries. Since most users are accustomed to using a mouse to navigate, here are a few tips to use the keyboard to navigate the TUI.

- **F2**: Enter setup to configure Voice of the Machine Edge settings.
- **F5**: Refresh the screen.
- **Tab**: Move the cursor from one field to the next.
- **Arrows**:
  - Move the cursor to select an item in a list, such as the TUI Configuration Options.
  - Use the Arrow to select an item, tab to **Next** at the bottom of the screen, and then press **Enter** to go to the selected option's screen.
- **Enter**:
  - Accept the data and move to the next field.
  - Submit the form for processing and then display the next screen in the workflow.

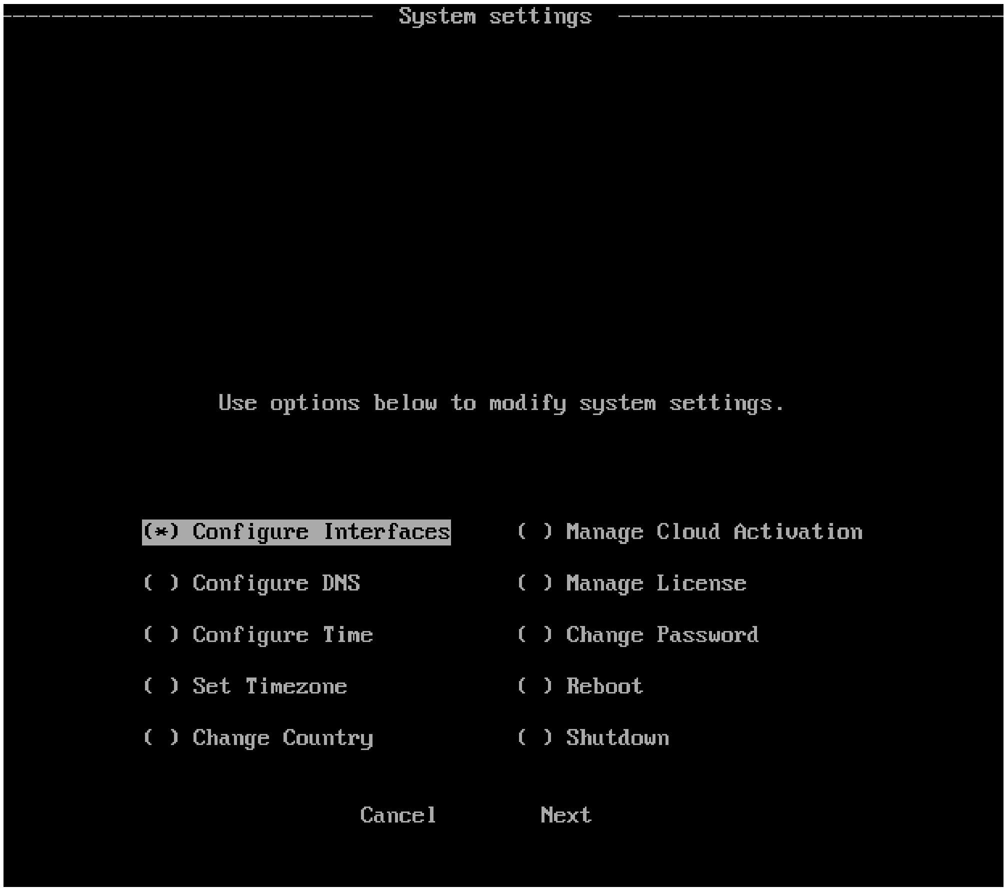
Example of TUI Navigation

When selecting an option:

- Use the arrow keys to move through the list of TUI Configuration Options.
- Use the Tab key to select the item highlighted by the cursor. The cursor then moves to the Next button.
- Press Enter when the cursor is on the Next button, to go to the selected option's screen.

Configure Voice of the Machine Edge via the TUI

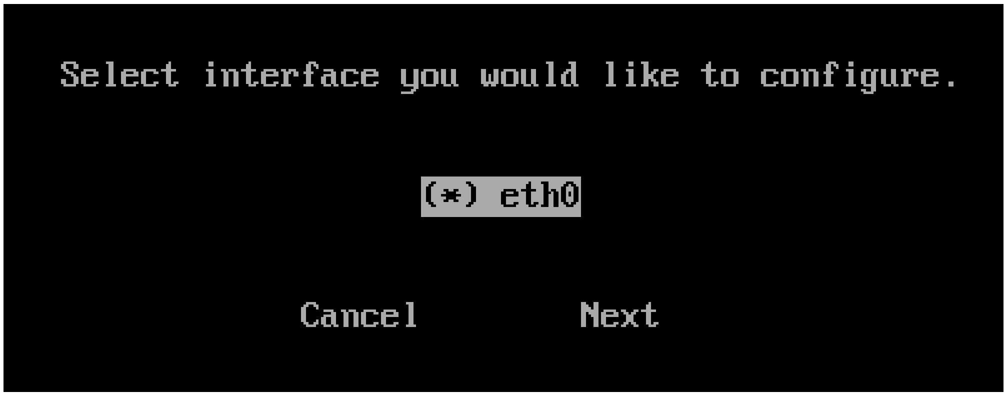
The Voice of the Machine Edge TUI offers many of the same options that are available via the web browser interface. Note that while many of these settings should be configured initially, when deploying a Voice of the Machine Edge device, some of these options, such as Manage Cloud Activation, may be configured at a later time.



```
-----------------------------  System settings  ------------------------------

        Use options below to modify system settings.

     (*) Configure Interfaces     ( ) Manage Cloud Activation

     ( ) Configure DNS            ( ) Manage License

     ( ) Configure Time           ( ) Change Password

     ( ) Set Timezone             ( ) Reboot

     ( ) Change Country           ( ) Shutdown

                    Cancel        Next
```

Configure Interfaces

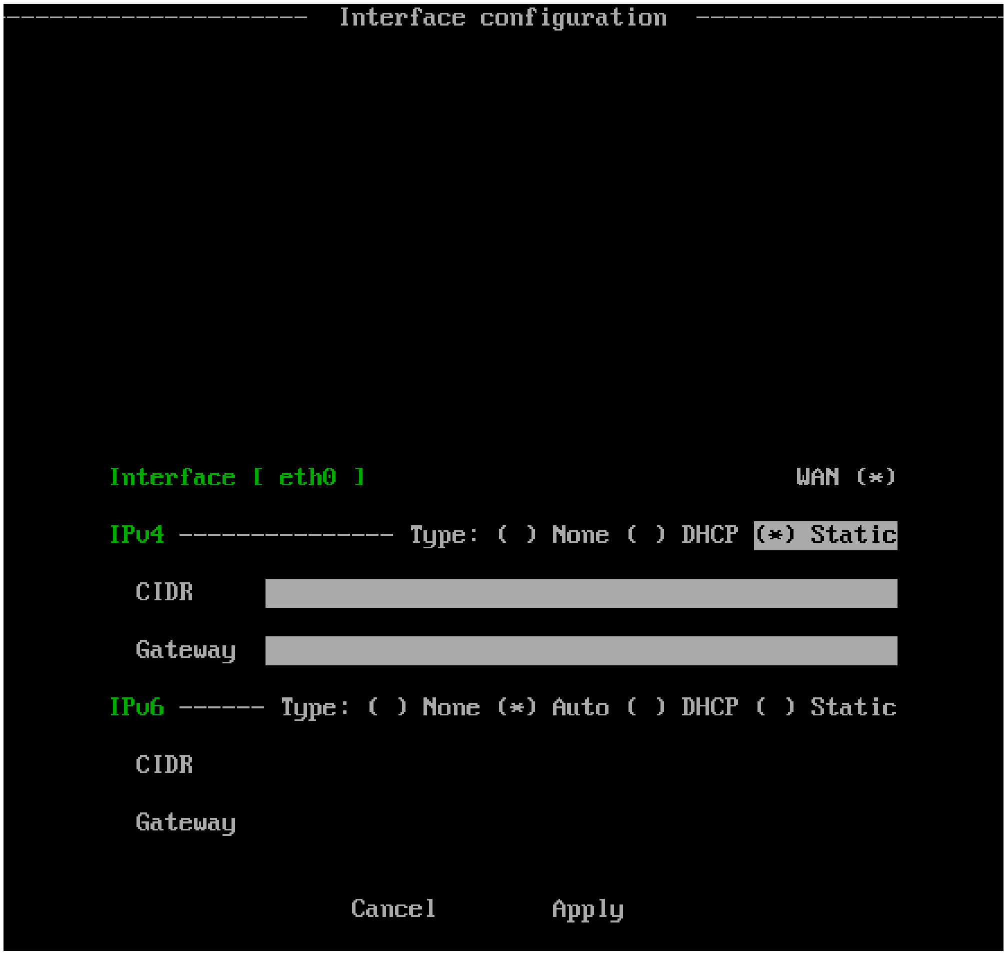
Select the interface configuration Type and enter details, if required.



```
Select interface you would like to configure.

                 (*) eth0

         Cancel         Next
```

In the following example, a static IP address can be configured by supplying:

- **CIDR** - This Classless Inter-Domain Routing (CIDR) notation identifes a block of IP addresses, similar to network subnetting. This example of CIDR notation (`192.168.0.15/24)` means that the first 24 bits of the IP address are used for network routing.
- **Gateway** - The default gateway address on a network can be found using the `ipconfig` command.



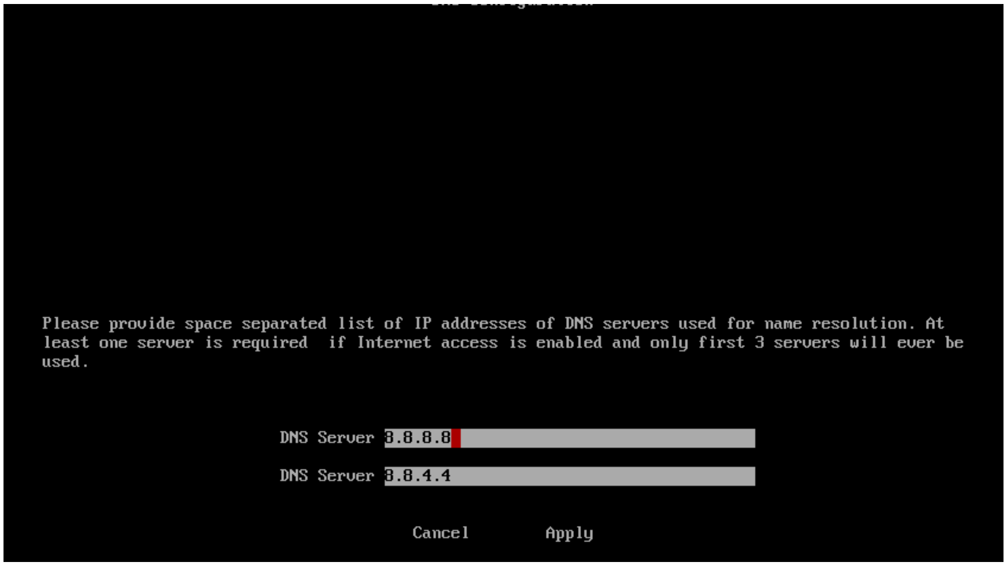
```
----------------------  Interface configuration  ----------------------

   Interface [ eth0 ]                                      WAN (*)
   IPv4 --------------- Type: ( ) None ( ) DHCP (*) Static
     CIDR
     Gateway
   IPv6 ------ Type: ( ) None (*) Auto ( ) DHCP ( ) Static
     CIDR
     Gateway

                      Cancel        Apply
```

Configure DNS

Use this dialog to enter space-separated IP addresses of the DNS servers.

- One server is required.

- If internet access is enabled, only the first three servers will be used.



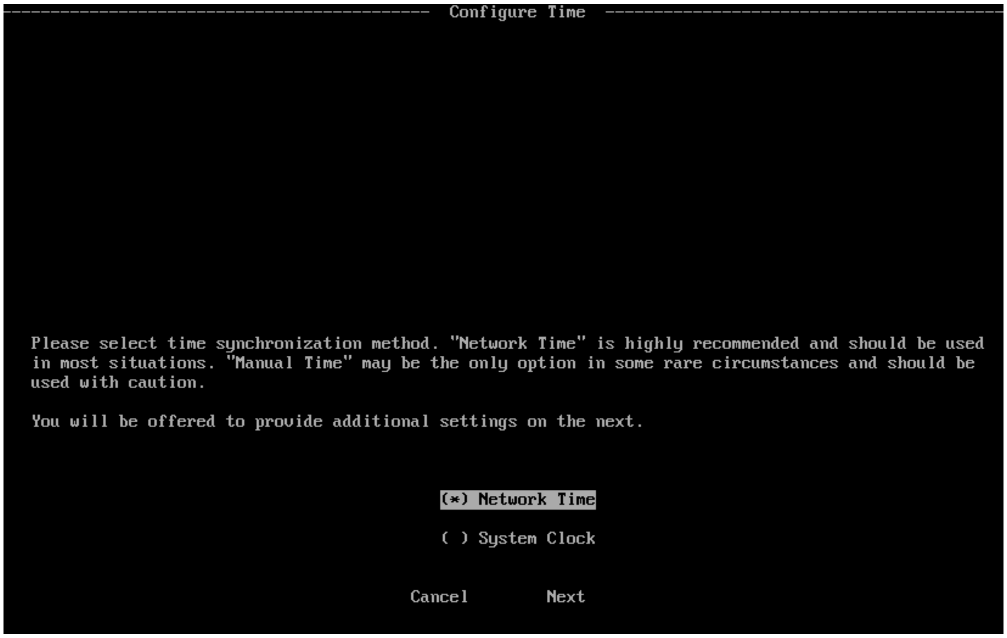
Please provide space separated list of IP addresses of DNS servers used for name resolution. At least one server is required if Internet access is enabled and only first 3 servers will ever be used.

DNS Server 8.8.8.8

DNS Server 8.8.4.4

Configure Time

Network Time is the recommended configuration. Only in rare instances should the System Clock be used for time synchronization.



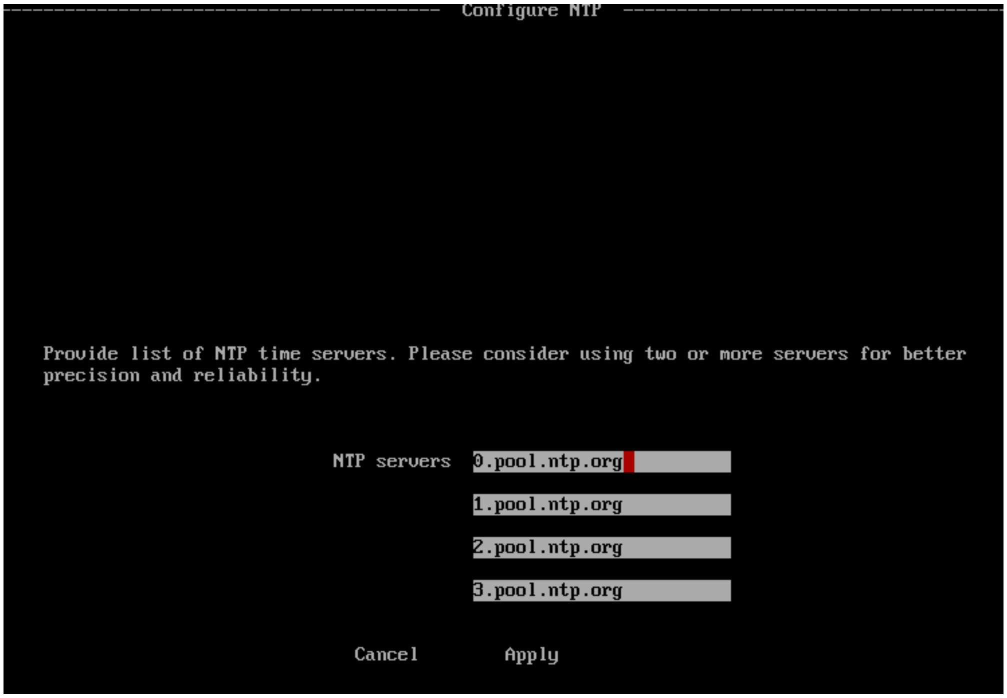
Please select time synchronization method. "Network Time" is highly recommended and should be used in most situations. "Manual Time" may be the only option in some rare circumstances and should be used with caution.

You will be offered to provide additional settings on the next.

(*) Network Time

( ) System Clock

Enter NTP (Network Time Protocol) servers and select **Apply**.
Two or more NTP servers is recommended.

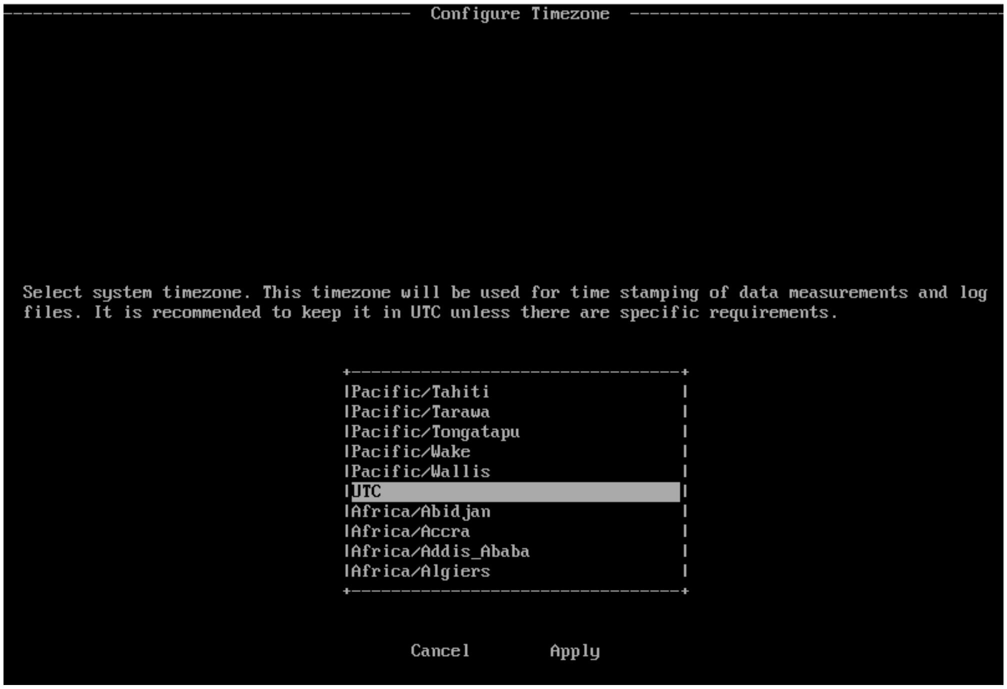


```
Configure NTP

Provide list of NTP time servers. Please consider using two or more servers for better
precision and reliability.

NTP servers  0.pool.ntp.org
             1.pool.ntp.org
             2.pool.ntp.org
             3.pool.ntp.org

    Cancel        Apply
```

Set Timezone

The timezone setting is significant because it is used for collected data and log timestamps.

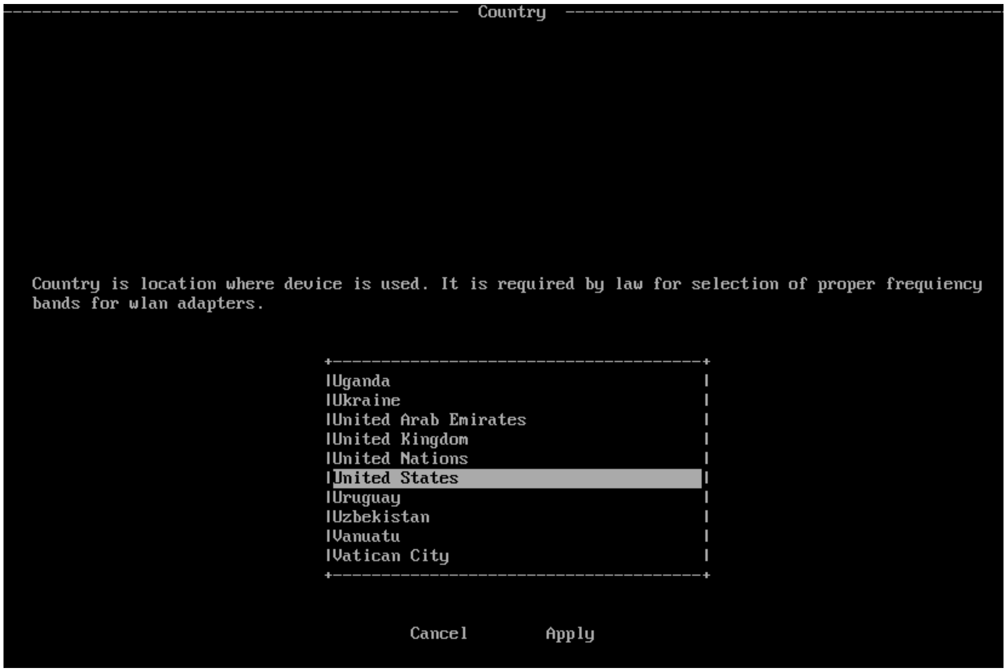
1. Use the arrow keys to scroll and select the timezone.
2. **Tab** to **Apply** and press **Enter**.



```
Configure Timezone

Select system timezone. This timezone will be used for time stamping of data measurements and log
files. It is recommended to keep it in UTC unless there are specific requirements.

+---------------------------------+
|Pacific/Tahiti                   |
|Pacific/Tarawa                   |
|Pacific/Tongatapu                |
|Pacific/Wake                     |
|Pacific/Wallis                   |
|UTC                              |
|Africa/Abidjan                   |
|Africa/Accra                     |
|Africa/Addis_Ababa               |
|Africa/Algiers                   |
+---------------------------------+

    Cancel        Apply
```

Change Country

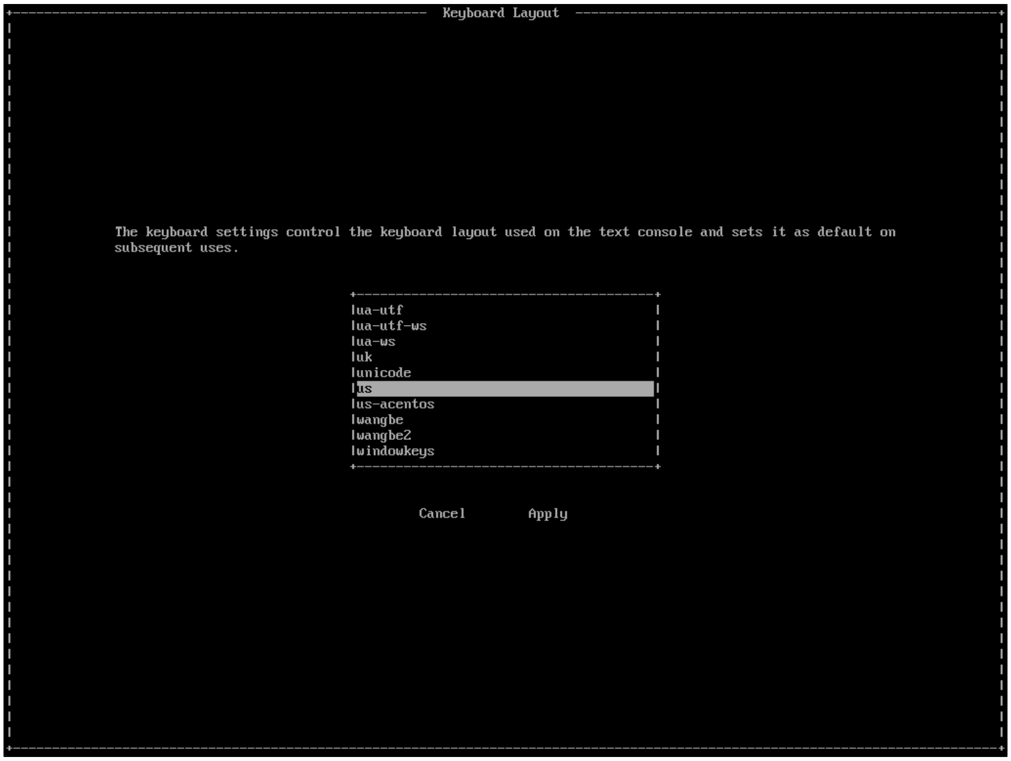
Country is the physical location of the device. It is required by law for selection of proper frequency bands for WLAN adapters.



```
Country is location where device is used. It is required by law for selection of proper frequiency
bands for wlan adapters.
```

Change Keyboard Layout

Use this menu to change the keyboard layout to match the keyboard layout of your country.



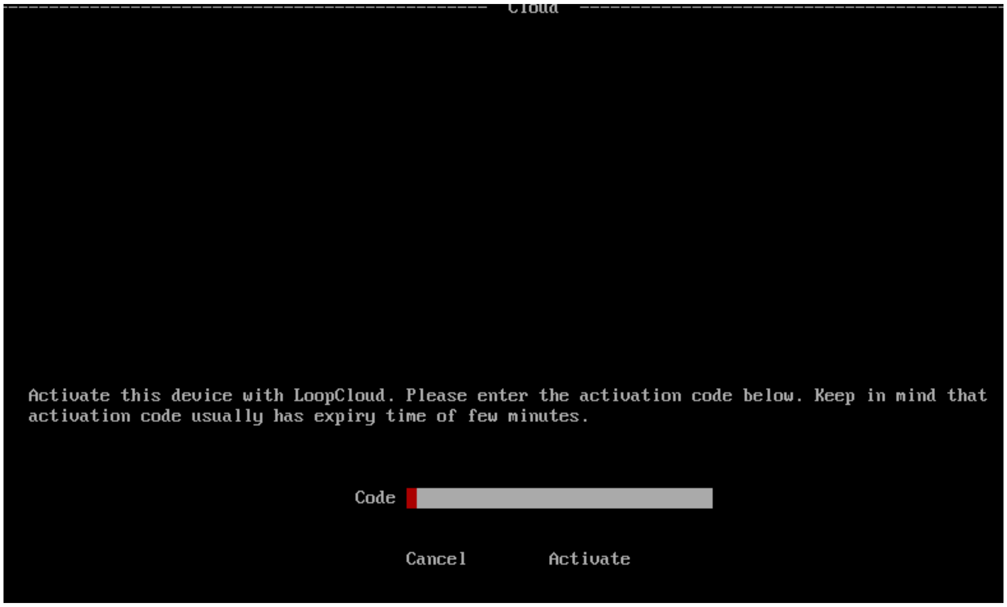
```
The keyboard settings control the keyboard layout used on the text console and sets it as default on
subsequent uses.
```

Manage Cloud Activation

Use this option to activate the LoopEdge device with LoopCloud. Keep in mind that the activation code usually has an expiration time of a few minutes. For prerequisites and instructions for getting an Activation Code, see Configure LoopCloud Connectivity.

Typically, this TUI option is used only when someone at another location, accessing the browser interface, provides an activation code to the technician who is physically located with the device.

- Enter the activation code and select **Activate**.



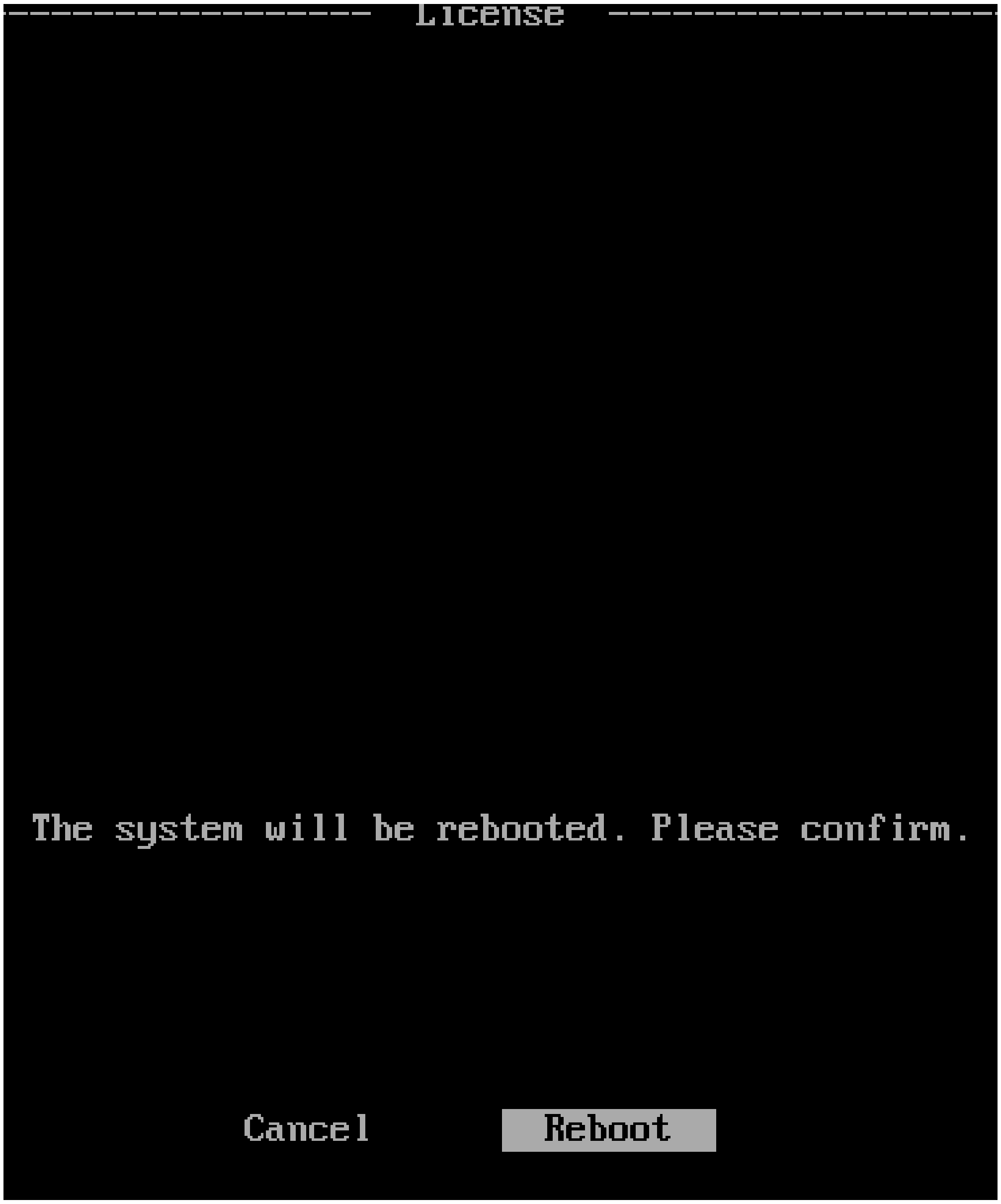
Cloud

Activate this device with LoopCloud. Please enter the activation code below. Keep in mind that activation code usually has expiry time of few minutes.

Reboot

If you decide not to reboot, use the arrow keys to select **Cancel** and then press **Enter**.

The operating system will reboot and return to the initial screen where you can once again press **F2** to log in and choose TUI Configuration Options.

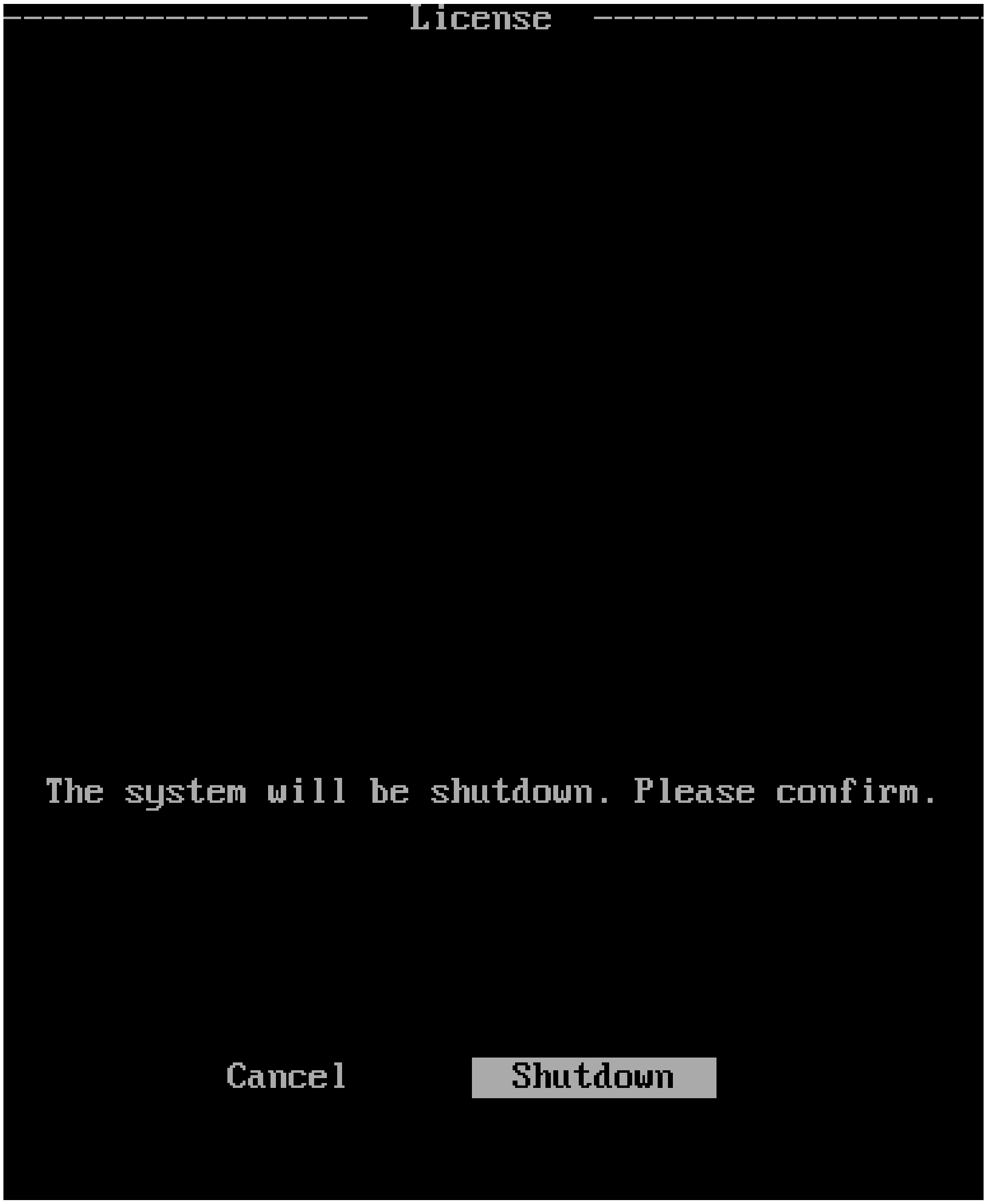


The system will be rebooted. Please confirm.

Shutdown

If you decide not to shut down the operating system, use the arrow keys to select **Cancel** and then press **Enter**.

When you shut down Voice of the Machine Edge, the TUI interface closes altogether.



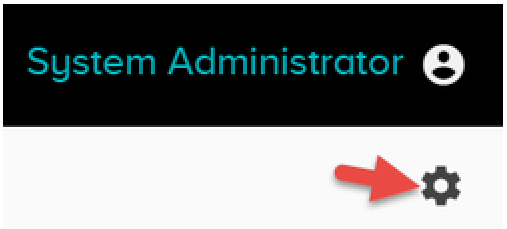
```
License

The system will be shutdown. Please confirm.
```

Related articles

- How to Activate an Offline License Key
- How to Use Terminal User Interface with Voice of the Machine Edge
- Voice of the Machine Edge Starter Kit Guide
- How to Update Sensor Rate
- How to Add a New/Blank Sensor

How to Navigate Voice of the Machine Dashboard

The dashboard provides statistics about the gateway. It includes the **CPU Utilization, Memory Usage, Network In, Network Out.** The dashboard has chart toggles to customize chart layouts and can be modified to what the user prefers.



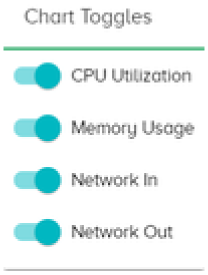
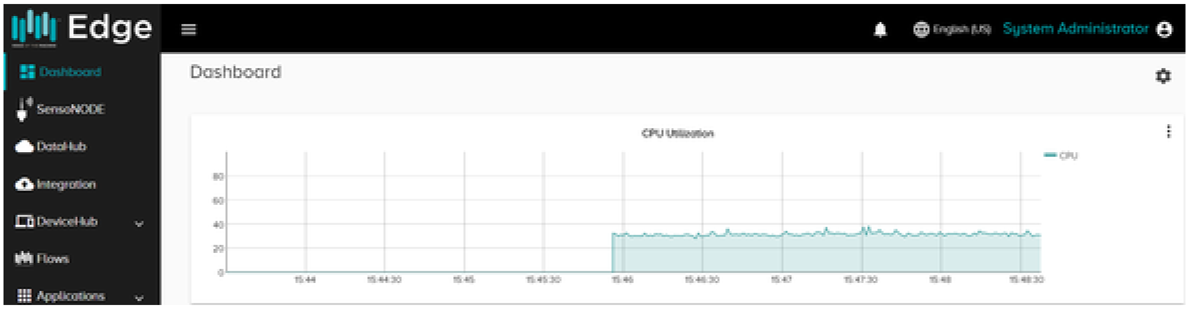


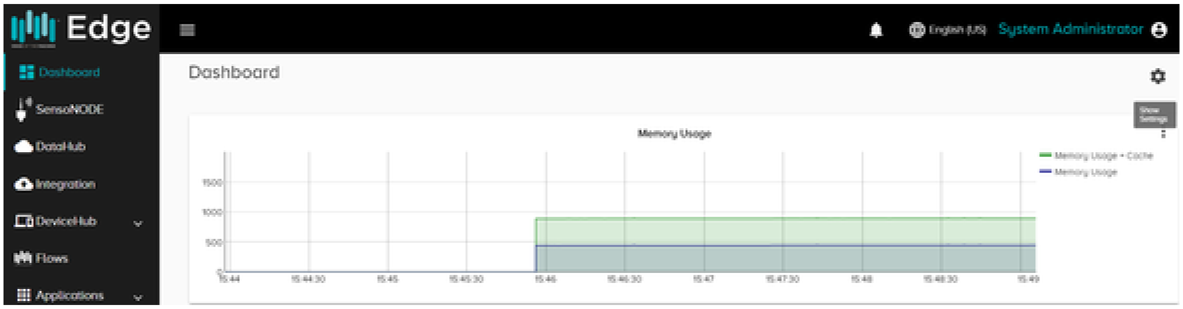
Chart Toggles

- CPU Utilization
- Memory Usage
- Network In
- Network Out

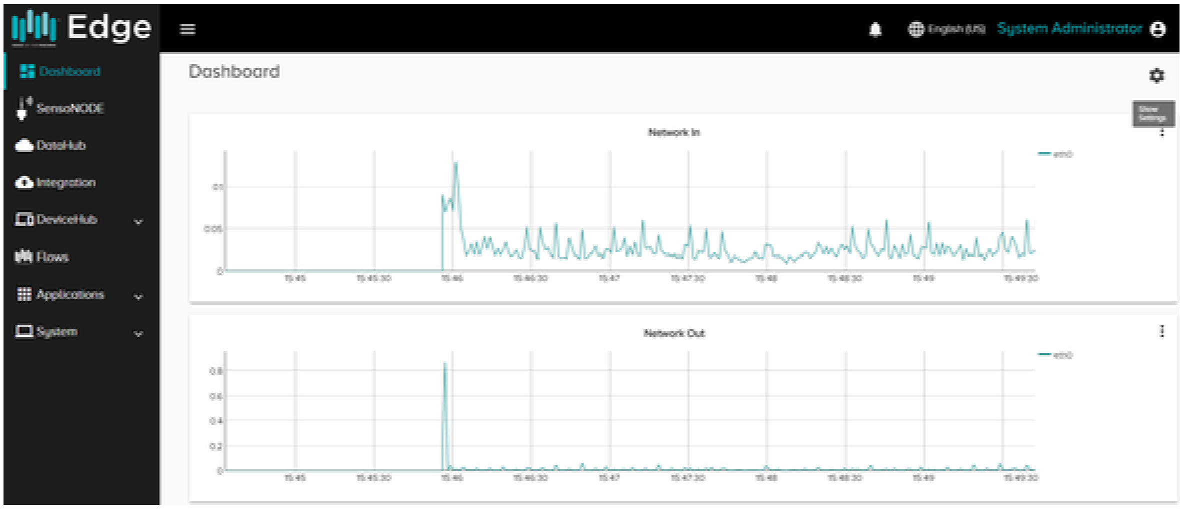
- **CPU Utilization-** it refers to the computer's usage of processing resources or the amount of work handled by the CPU. The y-axis represents the percentage of the CPU utilization and the x-axis is the time



- **Memory Usage-** This graph shows the details on system memory usage. There are two graphs for memory usage, one which gives the memory and cache usage and the other is the actual memory usage. The y-axis of the graph represents the time memory used in megabytes and the x-axis is the time:



- **Network graphs-** These graphs provide the value of the network statistics. The Network in-graph provides the download speed, whereas the network out-graph provides the upload speed. The y-axis represents the Mb and the y-axis is the time. The co-ordinate is the actual speed in Mbps.



Initially the device is provided with a 15 day trial license.

Contact Parker for a longer term license.

Related articles

- Voice of the Machine Edge
- How to Activate an Offline License Key
- How to Complete a SensoNODE Gold Site Survey (Video)
- How to Use Terminal User Interface with Voice of the Machine Edge
- How to Read Sensor Data String Outputs

How to Connect SensoNODE to Gateway

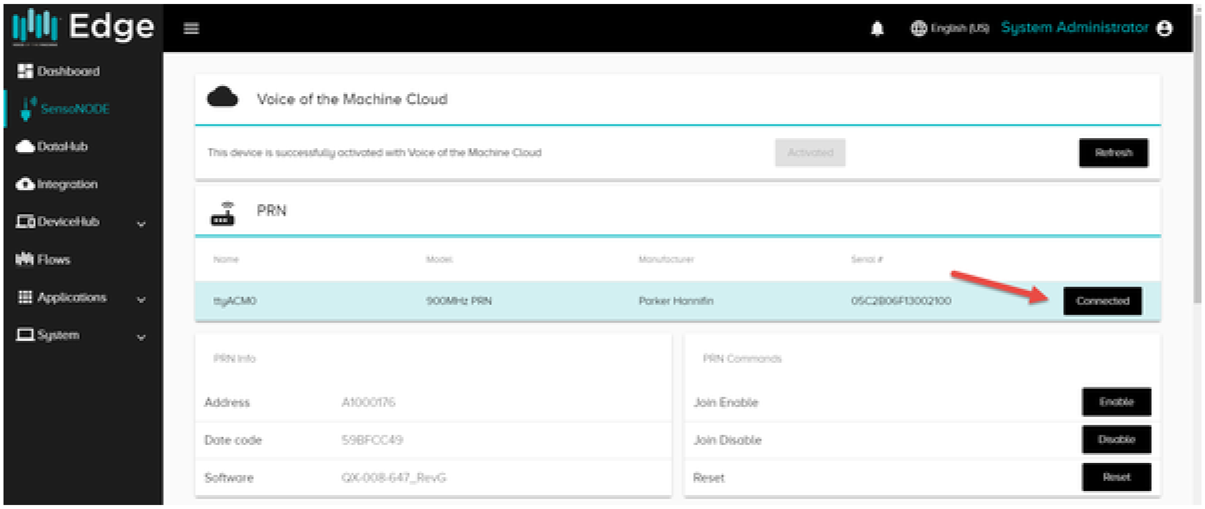
IMPORTANT
Do **not** click the black button on the cap or the gold pin inside the sensor. This will delete the connection and reset the sensor. If this happens refer to How to Add a New/Blank Sensor

Equipment Needed

- **Screwdriver Torx T15** (Screws on Gold Cap)
- **Battery CR123A** (Included with each sensor)
  - Can be bought at any local hardware store

Connect PRN to the Gateway

- Click **Connect** to the right, to connect a PRN and it will display the Sensor information along with some basic information about the PRN.



This device is successfully activated with Voice of the Machine Cloud

Name	Model	Manufacturer	Serial #
ttyACM0	900MHz PRN	Parker Hannifin	05C2B06F13002100

PRN Info

Address	A1000176
Date code	59BFCC49
Software	QX-008-647_RevG

PRN Commands

Join Enable

Join Disable

Reset

- The user will see a list of the PRN connected to the gateway. The PRN details includes
  - **Name**: the port to which it is connected to
  - **Model**: The model type of the PRN
  - **Manufacturer**: The manufacturer name of the PRN
  - **Serial #**: The serial number of the PRN

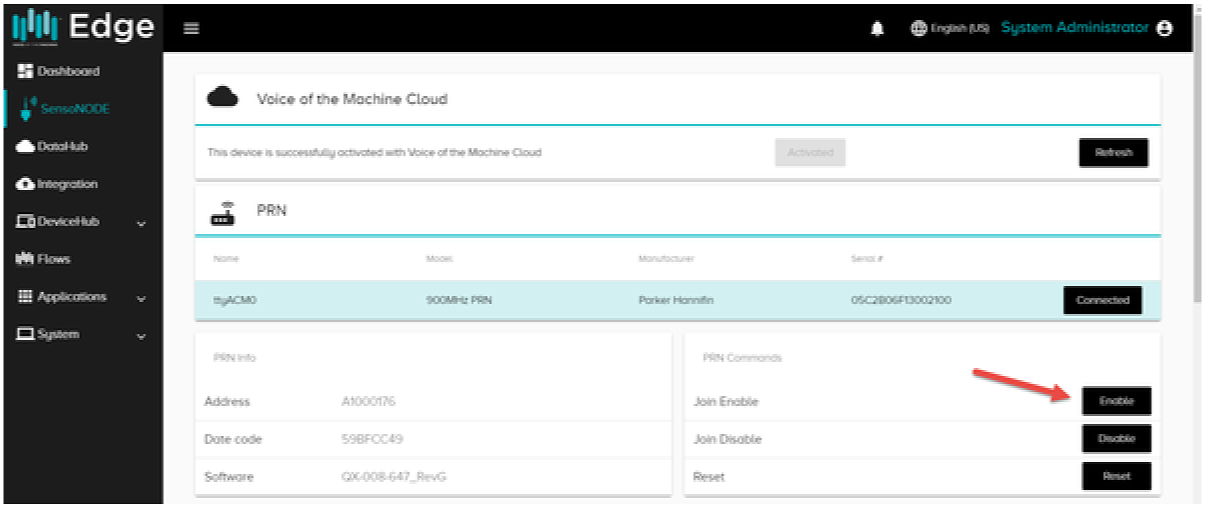
- The most frequently used commands can also be executed from here
  - **Join Enable**- This command allows new sensor to join the PRN
  - **Join Disable**- This command does not allow a new sensor to join the PRN. It is default at power up
  - **Reset**- This command resets the data stored by the PRN. The Join Status is turned off if Enabled initially.

Do **not** click **Reset** it will delete all sensors without a confirmation. If it happens just rejoin all the sensors with the steps below.

Connecting Sensor to the PRN

Once logged into the Voice of the Machine Edge system, click on the **SensoNODE** tab in the left-navigation menu.

1. Click **Enable** (within the PRN Commands)



Voice of the Machine Cloud

This device is successfully activated with Voice of the Machine Cloud — Activated — Refresh

PRN

Name	Model	Manufacturer	Serial #	
ttyACM0	900MHz PRN	Parker Hannifin	05C2B06F13002100	Connected

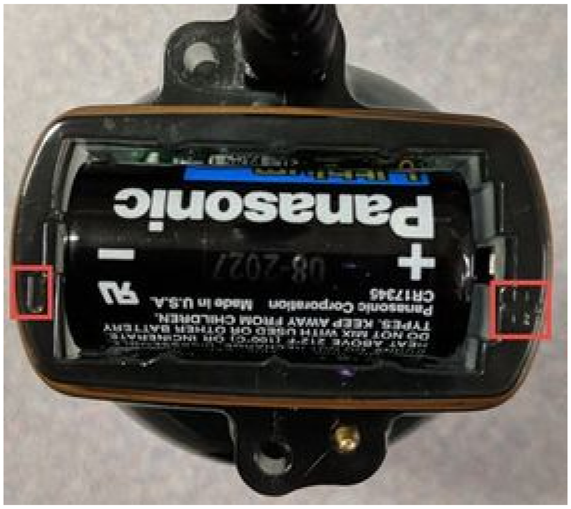
PRN Info

Address	A1000176
Date code	59BFCC49
Software	QX-008-647_RevG

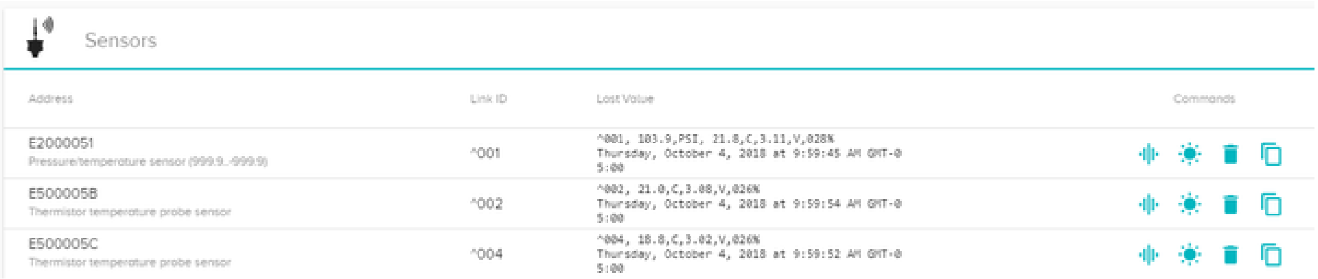
PRN Commands

Join Enable	Enable
Join Disable	Disable
Reset	Reset

2. Insert the battery into the desired SensoNODE Gold sensor (+ and - and labeled on the sensor)



3. Sensor serial number (located on sticker on the back) will display in the Sensors menu (at bottom of screen)

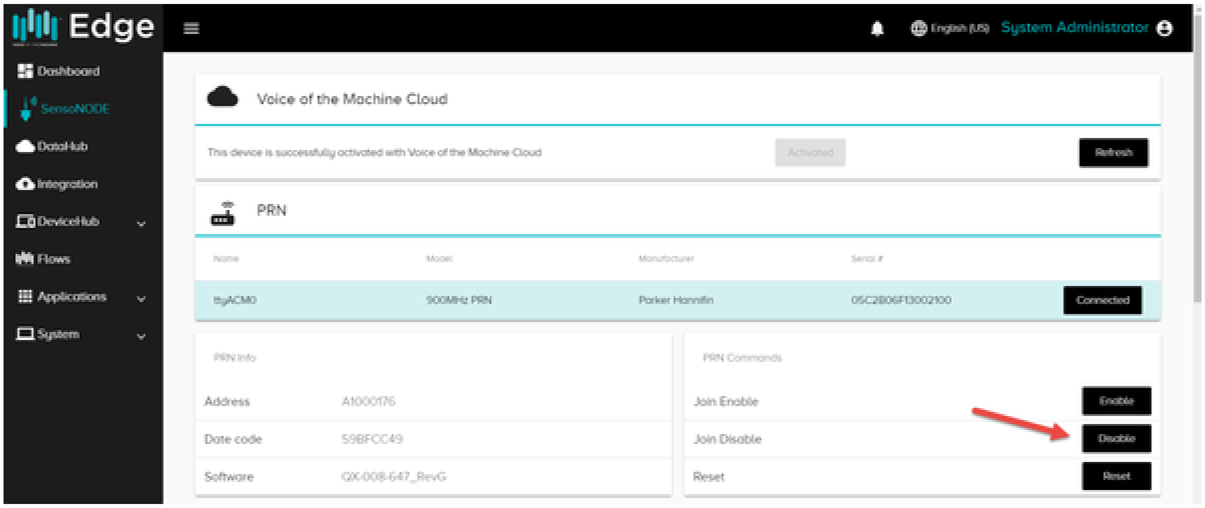


Sensors

Address	Link ID	Last Value	Commands
E2000051 Pressure/temperature sensor (999.9..-999.9)	^001	^001, 103.9,PSI, 21.8,C,3.11,V,028% Thursday, October 4, 2018 at 9:59:45 AM GMT-0 5:00	
E500005B Thermistor temperature probe sensor	^002	^002, 21.0,C,3.08,V,026% Thursday, October 4, 2018 at 9:59:54 AM GMT-0 5:00	
E500005C Thermistor temperature probe sensor	^004	^004, 18.8,C,3.02,V,026% Thursday, October 4, 2018 at 9:59:52 AM GMT-0 5:00	

4. Repeat steps 2-3 with all remaining sensors.

5. Click **Disable** (within the PRN Commands)



Voice of the Machine Cloud

This device is successfully activated with Voice of the Machine Cloud — Activated — Refresh

PRN

Name	Model	Manufacturer	Serial #	
ttyACM0	900MHz PRN	Parker Hannifin	05C2B06F13002100	Connected

PRN Info

Address	A1000176
Date code	59BFCC49
Software	QX-008-647_RevG

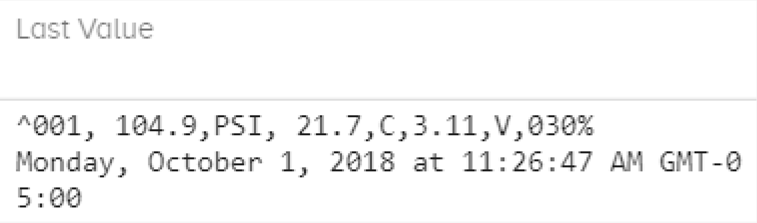
PRN Commands

Join Enable	Enable
Join Disable	Disable
Reset	Reset

6. Sensor is now joined to the Voice of the Machine Edge system

SensoNODE sensors

- The user will see the list of sensors connected to the PRN under the Sensor Inventory list
- Last Value displays data in the following format:

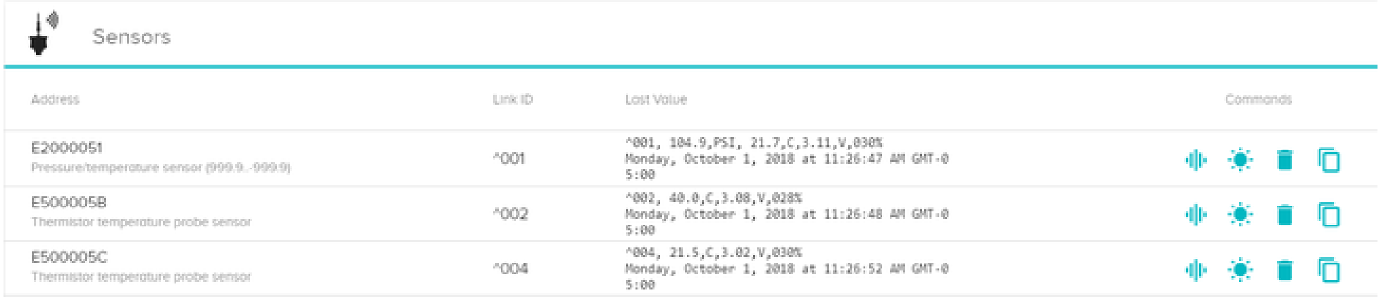


Last Value

```
^001, 104.9,PSI, 21.7,C,3.11,V,030%
Monday, October 1, 2018 at 11:26:47 AM GMT-0
5:00
```

- Link ID (^001)
- Measure Data (104.9)
- Unit (PSI)
- Additional Measured Data (21.7)
- Additional Unit (C)
- Battery Level (3.11)
- Battery Units (V)
- Signal Strength (030%)
- Date and Time (Monday, October 1, 2018 at 11:26:47 AM GMT-05:00)

- Various commands related to the sensor can be access from the sidebar next to the sensor. Each icon shows what command it is when the user hoovers over the icon
- The various sensor level commands include:
  - **Update Sensor Rate** - Changing the rate will affect battery life and impact sensor data. Be caution when changing.
  - **Sensor Identity**- Sensor will flash LED. Flash pattern: Blue-Green-Blue. Supported in sensor firmware RevA and greater
  - **Delete node**- Delete the specified node from the link table
  - **Copy**- Copies the Sensor identifying info into clipboard (Used in Flows)



Sensors

Address	Link ID	Last Value	Commands
E2000051 Pressure/temperature sensor (999.9..-999.9)	^001	^001, 104.9,PSI, 21.7,C,3.11,V,030% Monday, October 1, 2018 at 11:26:47 AM GMT-0 5:00	
E500005B Thermistor temperature probe sensor	^002	^002, 40.0,C,3.08,V,028% Monday, October 1, 2018 at 11:26:48 AM GMT-0 5:00	
E500005C Thermistor temperature probe sensor	^004	^004, 21.5,C,3.02,V,030% Monday, October 1, 2018 at 11:26:52 AM GMT-0 5:00	

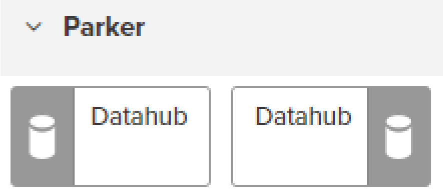
Related articles

How to Send Data to the Cloud

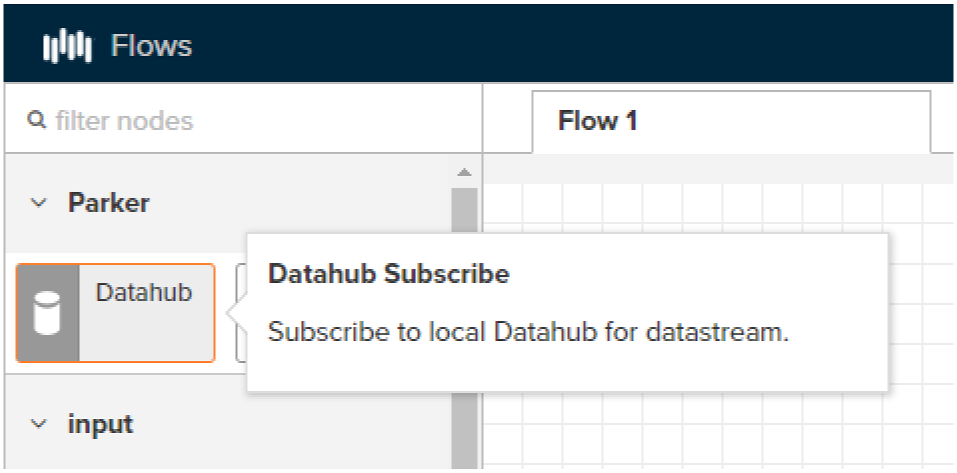
SensoNode data can be sent to any cloud platform using the flows. Raw data itself can be sent to any cloud platform or to any end application using the flows function. The user can easily get the data from the sensor by using the information obtained on the SensoNODE page.

Steps to collect data from the SensoNode

1. Select the SensoNode tab which gives information about the PRN and the SensoNODE sensors.
2. Under Sensors the serial number is displayed, the same number can be found on the sensor. (Write the serial number down)
3. Open **Flows**
4. Under the Parker section the user will find two nodes Datahub Publish and Datahub Subscribe



5. Select the **Datahub Subscribe** node to collect the data from the particular sensor.

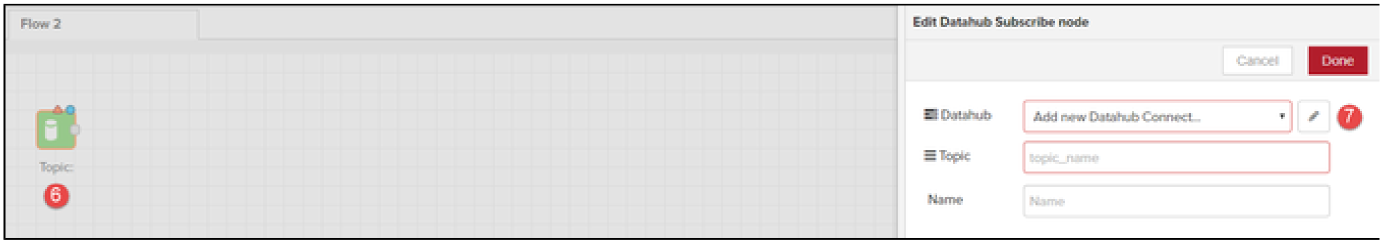


Datahub Subscribe

Subscribe to local Datahub for datastream.

6. Drag the Datahub onto the Flow space

7. Click the pencil icon to the right of DataHub



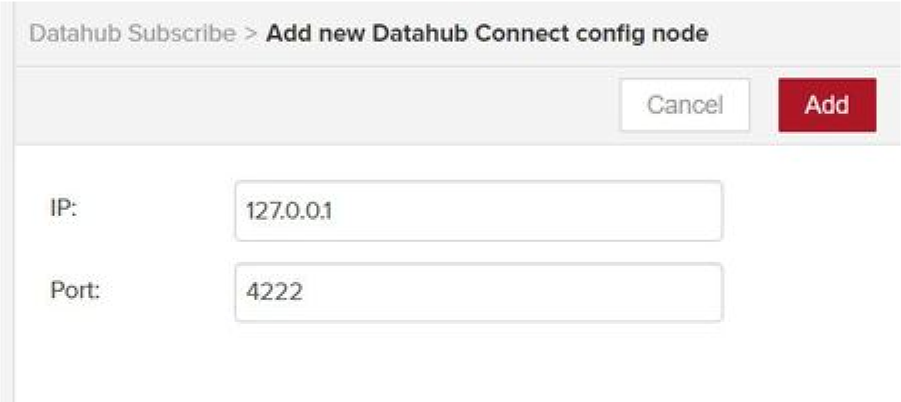
Edit Datahub Subscribe node

Datahub: Add new Datahub Connect...

Topic: topic_name

Name: Name

8. Click **Add** to add the predefined server (Default IP and Port should be correct)

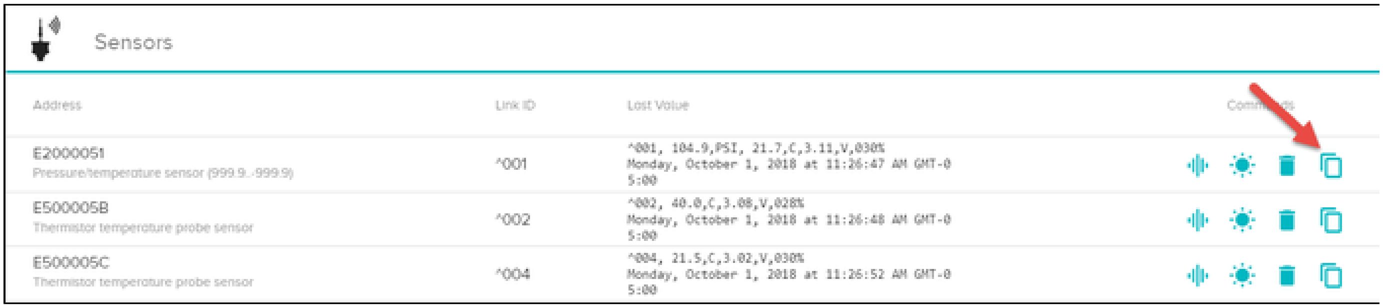


Datahub Subscribe > **Add new Datahub Connect config node**

IP: 127.0.0.1

Port: 4222

9. Go to SensoNODE and scroll down to Sensors click the right-most ⧉ icon to copy.



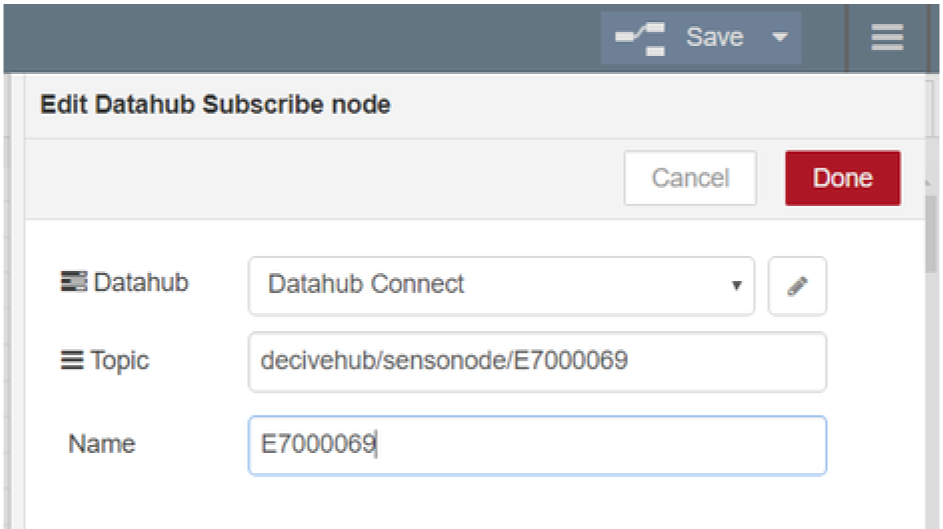
Sensors

Address	Link ID	Last Value	Commands
E2000051 Pressure/temperature sensor (999.9..-999.9)	^001	^001, 104.9,PSI, 21.7,C,3.11,V,030% Monday, October 1, 2018 at 11:26:47 AM GMT-0 5:00	
E500005B Thermistor temperature probe sensor	^002	^002, 40.0,C,3.08,V,028% Monday, October 1, 2018 at 11:26:48 AM GMT-0 5:00	
E500005C Thermistor temperature probe sensor	^004	^004, 21.5,C,3.02,V,030% Monday, October 1, 2018 at 11:26:52 AM GMT-0 5:00	

If SensoNODE is not working correctly the user is able to enter the topic name to "*devicehub/sensonode/*Serial Number**"

Topic consists of the serial number which changes for each sensor the remaining part stays the same **devicehub/senonode/**

10. Click **Done**



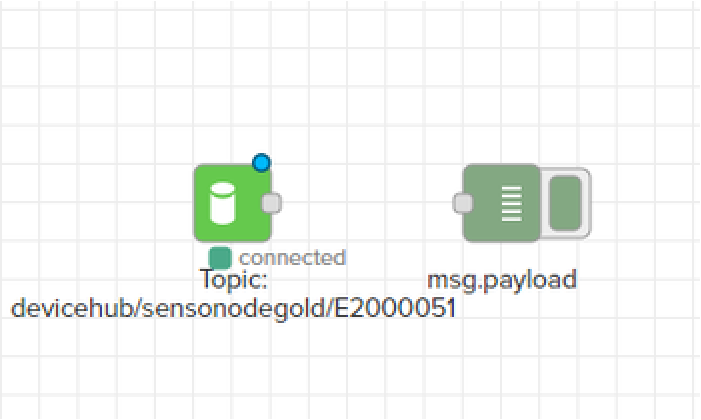
Edit Datahub Subscribe node

Datahub: Datahub Connect

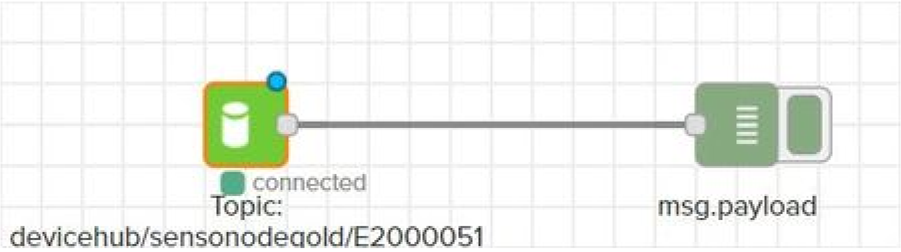
Topic: decivehub/sensonode/E7000069

Name: E7000069

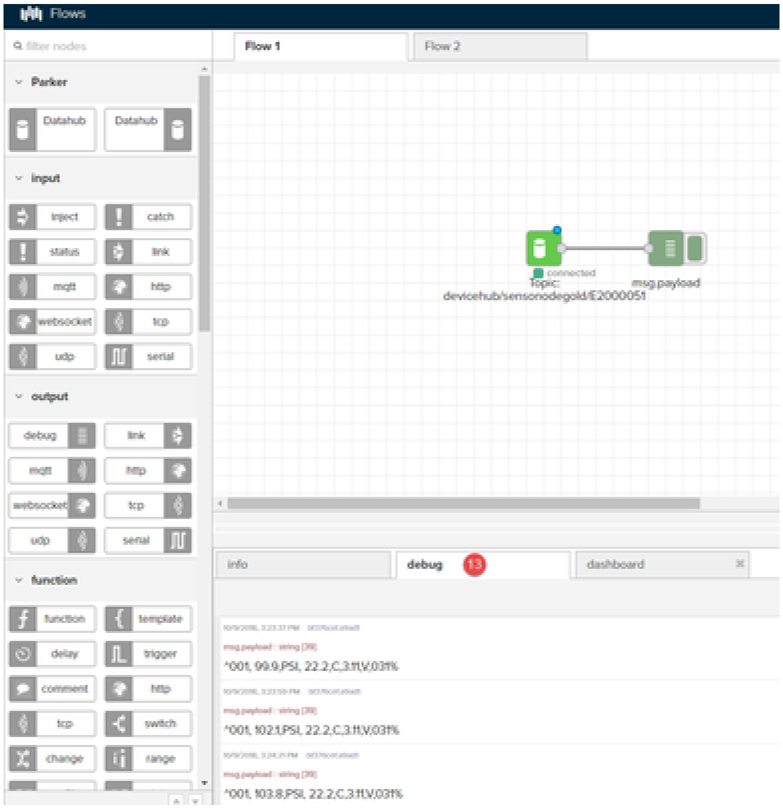
11. Drag a Debug Node from the outputs into the Flows space.



12. Connect the SensoNODE Datahub to it. The DataHub node can be connected to mqtt,http dashboard etc to perform analytics on to the sensonode data.



13. Click Debug on the lower panel to find the data output.



```
10/9/2018, 3:23:37 PM  [illegible]
msg.payload : string [39]
^001, 99.9,PSI, 22.2,C,3.11,V,03f%
10/9/2018, 3:23:59 PM  [illegible]
msg.payload : string [39]
^001, 102.1,PSI, 22.2,C,3.11,V,03f%
10/9/2018, 3:24:21 PM  [illegible]
msg.payload : string [39]
^001, 103.8,PSI, 22.2,C,3.11,V,03f%
```

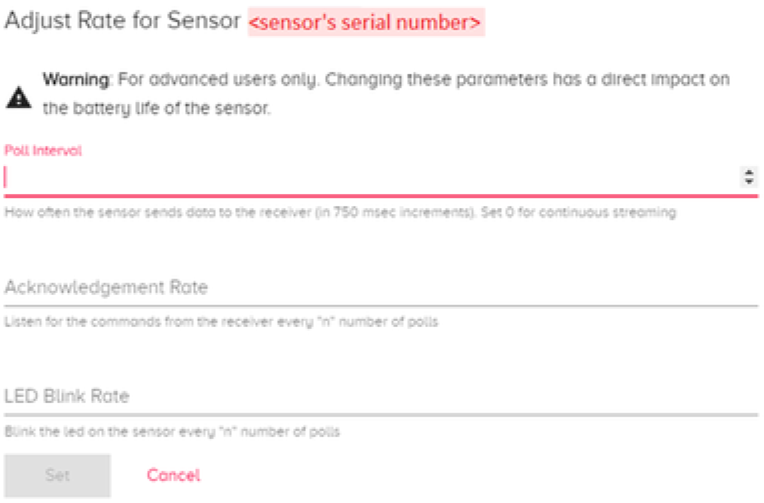
How to Update Sensor Rate

IMPORTANT

For advanced users only. Changing these parameters has a direct impact on the battery life of the sensor.

STEPS:

1. Click on **SensoNODE** on the left navigation panel
2. Navigate to **Sensors** within the main page
3. Under the **Commands** column identify the **Update Sensor Rate** icon
4. The following menu will display with 3 options



Adjust Rate for Sensor <sensor's serial number>

Warning: For advanced users only. Changing these parameters has a direct impact on the battery life of the sensor.

Poll Interval

How often the sensor sends data to the receiver (in 750 msec increments). Set 0 for continuous streaming

Acknowledgement Rate

Listen for the commands from the receiver every "n" number of polls

LED Blink Rate

Blink the led on the sensor every "n" number of polls

Default Settings (Only Integers are accepted)

Poll: 24   **Poll Interval:** How often sensor sends data to gateway. Acknowledgement Rate and LED are dependent on Poll Interval. Do not input values less than 1.

Ack: 15   **Acknowledgement Rate:** How often gateway sends commands to sensor (ex. Updating Sensor Rate and NODE Identify)

LED: 1   **LED Blink Rate:** How often LED on sensor blinks

EXAMPLE:

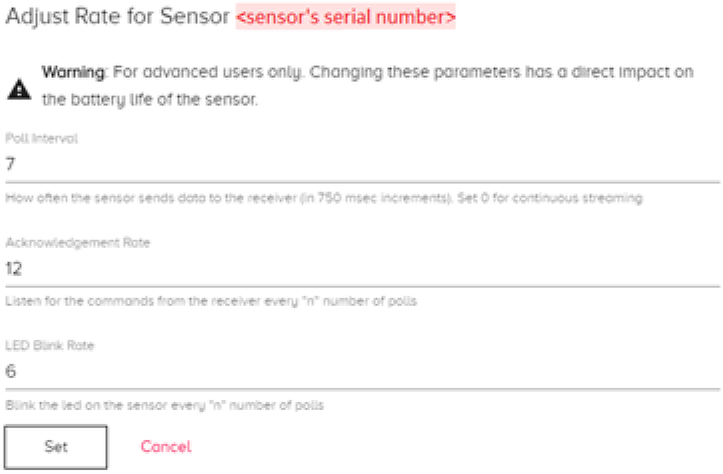
User wants to send data from sensor to gateway every **5 seconds**, acknowledge commands from gateway-to-sensor every **60 seconds**, and blink LED every **30 seconds**.

- Determine Poll Interval
  - 5 seconds / 0.75 = **6.66667** (0.75 is 750 milliseconds - this is constant for all Poll Intervals)
- Determine Acknowledgement Rate
  - 60 seconds / 5 seconds = **12** (5 seconds is determined by the Poll Interval)
- Determine LED Blink Rate
  - 30 seconds / 5 seconds = **6** (5 seconds is determined by the Poll Interval)
- Input determined values into *Adjust Rate for Sensor* menu

Poll: 7 (Need to round to nearest Integer)

Ack: 12

LED: 6



Adjust Rate for Sensor <sensor's serial number>

Warning: For advanced users only. Changing these parameters has a direct impact on the battery life of the sensor.

Poll Interval: 7

How often the sensor sends data to the receiver (in 750 msec increments). Set 0 for continuous streaming

Acknowledgement Rate: 12

Listen for the commands from the receiver every "n" number of polls

LED Blink Rate: 6

Blink the led on the sensor every "n" number of polls

5. Click **Set**

6. *Note*: Command has been queued and will send upon next acknowledge interval

- How to Activate an Offline License Key
- How to Complete a SensoNODE Gold Site Survey (Video)
- How to Use Terminal User Interface with Voice of the Machine Edge
- Voice of the Machine Edge Starter Kit Guide
- Voice of the Machine Mobile App

SensoNODE Gold LED Status Colors

SensoNODE Sensors

LED Status - Start Up Sequence

Red - Green - Blue - Sensor start up (If this repeats over and over, the battery is too low to start up the radio)

LED Status - Long Fading Flash

Red - Sensor is in Discovery Mode. Looking to connect with a receiver. (Sensor will attempt to connect every 3 seconds. If it is unable to join after 200 attempts it will go into a deep sleep state)

LED Status - Single Brief Flash

> **Signal Rate**
> The LED blink rate is configured in the Voice of the Machine Edge SensoNODE page. The default LED blink rate is set to 20 seconds.

The LED gives a different color depending on the data measured. It is broken into 4 quartiles

Green - Sensor Measurement - Sensor reading between 0% - 25% of span

Yellow - Sensor Measurement - Sensor reading between 25% - 50% of span

Orange - Sensor Measurement - Sensor reading between 50% - 75% of span

Red - Sensor Measurement - Sensor reading between 75% - 100% of span

Examples:

Humidity sensor measures 0%-100%

Green - 0% - 25%

Yellow - 25% - 50%

Orange - 50% - 75%

Red - 75% - 100%

Temperature sensor measures (-40°F) - 230°F

Green: (-40°F) - 27.5°F

Yellow: 27.5°F - 95°F

Orange: 95°F - 162.5°F

Red : 162.5°F - 230°F

Vibration sensor measures X-axis peak acceleration 0 milli-g - 532+milli-g

Green: 0 milli-g - 176 milli-g

Yellow: 177 milli-g - 354 milli-g

Orange: 355 milli-g - 531 milli-g

Red : 532 milli-g - Above

PRN Receivers

- Supplies continuous power to sensors
- Used with IEC/UL 508 Class 2 power supply
    - The chargers below have been tested and found to work. Most of them are available through Amazon and elsewhere.

    - mfgr: LG model MCS-01WR (Google Nexus phones)
    - mfgr: Motorola model SPN5504A (Droid phones), and SPN5505A (Motorola phones)
    - mfgr: Samsung model ETA0U80JBE (Samsung phones)
    - mfgr: Samsung model EP-TA10JWE (Samsung galaxy tablets)
    - mfgr: Asus model AD83531 (google nexus tablets)
- USB adapters have also been tested and found to work with the non-compliant USB chargers. PortaPow USB adapters universal version only has been found to work. The smart charge or Apple versions did not work.

LED Status - Start Up Sequence

Red - Green - Blue - Sensor start up

LED Status - Working as Repeater

Blue: Every 2 seconds

LED Status - Working as a PRN (Every 2 Seconds)

Green: It has received a transmission from one or more linked Sensor Nodes

Blue: It has not received a transmission from Sensor Nodes since previous LED blink.

Red - **Green** (Continuously): Cannot initialize the wireless communications stack and initialize the radio.

How to Read Sensor Data String Outputs

Low Pressure Sensor (Vacuum Sensor)

- Serial Number Prefix “E0”
- Measured Phenomena - Pressure/temperature sensor, 99.99 to -99.99 PSI gauge pressure range.
- Data string example: “^008, 12.04,PSI, 2.3,C,2.99,V,015%”
- Data string description:

“^link address, primary measurement (pressure), units (PSI), secondary measurement (ambient temperature), units (°C), battery voltage, units (V), RSSI%”

Medium Pressure Sensor (Pneumatic Sensor)

- Serial Number Prefix “E2”
- Measured Phenomena - Pressure/temperature sensor, 999.9 to -999.9 PSI gauge pressure range.
- Data string example: “^003, 82.4,PSI, 2.3,C,2.99,V,015%”
- Data string description:

“^link address, primary measurement (pressure), units (PSI), secondary measurement (ambient temperature), units (°C), battery voltage, units (V), RSSI%”

High Pressure Sensor (Hydraulic Sensor)

- Serial Number Prefix “E4”
- Measured Phenomena - Pressure/temperature sensor, 9999 to -9999 PSI gauge pressure range.
- Data string example: “^005, 824,PSI, 2.3,C,2.99,V,015%”
- Data string description:

“^link address, primary measurement (pressure), units (PSI), secondary measurement (ambient temperature), units (°C), battery voltage, units (V), RSSI%”

Temperature Sensor (Probe)

- Serial Number Prefix “E5”
- Measured Phenomena – Fluid Temperature
- Data string example: “^004,-11.0,C,2.96,V,030%”
- Data string description:

“^link address, primary measurement (temperature), units (°C), battery voltage, units (V), RSSI%”

Temperature Sensor (Foot)

- Serial Number Prefix “E6”
- Measured Phenomena – Surface Temperature, Ambient Temperature
- Data string example: “^008,-11.0,C, 84.3,C,2.96,V,030%”
- Data string description:

“^link address, primary measurement (temperature), units (°C), secondary measurement (temperature), units (°C), battery voltage, units (V), RSSI%”

Humidity Sensor (Pneumatic)

- Serial Number Prefix “E7”
- Measured Phenomena - Humidity/temperature sensor
- Data string example: “^003, 32.4,%, 21.8,C,3.08,V,032%”
- Data string description:

“^link address, primary measurement (relative humidity), units (percent %), secondary measurement (temperature), units (°C), battery voltage, units (V), RSSI%”

Current Sensor

- Serial Number Prefix “E9”
- Measured Phenomena - Single phase current (average, peak) sensor
- Data string example: “^005, 42.0,%, 59.4,%,3.06,V,025%”
- Data string description:

“^link address, average measurement, units (percent % of full scale amperage), peak measurement, units (percent % of full scale amperage), battery voltage, units (V), RSSI%”

Analog Transmitter

- Serial Number Prefix “ED”
- Measured Phenomena – 4mA to 20mA Sensor Transmitter
- Data string example: “^008,100.00,%,2.99,V,028%”
- Data string description:

“^link address, primary measurement, units (percent % of mapped native measurement), battery voltage, units (V), RSSI%”

Flow Sensor (Pneumatic)

- Serial Number Prefix “EF”
- Measured Phenomena - Gas flow sensor,
- Data string example: “^006,4095,,4095,,4095,,3.02,V,031%”

(The flow sensor is a special case, it does not return calibrated readings with units – just **raw** 12 bit ADC values for use with calibration/flow calculation formulas)

- Data String Description:

“^link address, differential pressure (raw count value), absolute pressure (raw count value), temperature (raw count value), battery voltage, units (V), RSSI%”

Vibration Sensor (Accelerometer)

- Serial Number Prefix “F1”
- Measured Phenomena – Acceleration or Velocity with temperature output
- Data string example (sensor outputs three data strings):

1st (X) data string: “^008, 1256,Xpk, 2504,Xrms,2.99,V,028%”

2nd (Y) data string: “^008, 3456,Ypk, 4510,Yrms, -11,C,028%”

3rd (Z) data string: “^008, 3210,Zpk, 5202,Zrms,16,g,A,028%”

- Data string description:

1st data string: “^link address, X peak measurement, Xpk, X RMS measurement, Xrms, battery voltage, units (V), RSSI%”

2nd data string: “^link address, Y peak measurement, Ypk, Y RMS measurement, Yrms, secondary measurement, units (°C), RSSI%”

3rd data string: “^link address, Z peak measurement, Zpk, Z RMS measurement, Zrms, acceleration range (2, 4, 8, or 16 in g), units (mg or 100*mm/sec), acceleration/velocity mode (A or V), RSSI%”

*Note that each acceleration data item is 5 digits long and will be padded out with leading spaces as needed.

Flexible Displacement Transmitter (Strain or Position)

- Serial Number Prefix “F2”
- Measured Phenomena –Strain Sensor Transmitter with temperature output.
- Data string example (device transmits two data strings):

1st (Channel A) data string: “^006,A,101.20,%, 45.2,C,3.02,V,031%”

2nd (Channel B) data string: “^006,B, 21.40,%, 45.2,C,3.02,V,031%”

- Data string description:

1st data string: “^link address, channel A, percent strain, %, temperature, units (°C), battery voltage, units (V), RSSI%”

2nd data string: “^link address, channel B, percent strain, %, temperature, units (°C), battery voltage, units (V), RSSI%”

Related articles

- SensoNODE Gold LED Status Colors
- How to Read Sensor Data String Outputs
- How to Add a New/Blank Sensor

What is DataHub in Voice of the Machine Edge

- DataHub enables *northbound* cloud connectivity. Use DataHub to send edge-level data securely.
- DataHub enables *local* connections to the cloud using the MQTT protocol. DataHub buffers the data and if the connection drops out, no data is ever lost. DataHub enables Voice of the Machine Edge to run offline, but can push all offline data to the cloud once an internet connection is established. This helps in situations where network connectivity is not always reliable.
- The capacity of the data is limited by the hardware footprint and the data is discarded on a first-in first-out basis. To visualize these connections: Use Voice of the Machine Edge Flows to Visualize Data Flows.

How to Use Integration

Connectors to several third-party cloud services can be configured in Voice of the Machine Edge I. This enables efficient edge processing, with data published to the service provider.
Prerequisite
Knowledge of the third-party cloud service provider is required in order to configure Voice of the Machine Edge connectors.

Cloud Service Providers

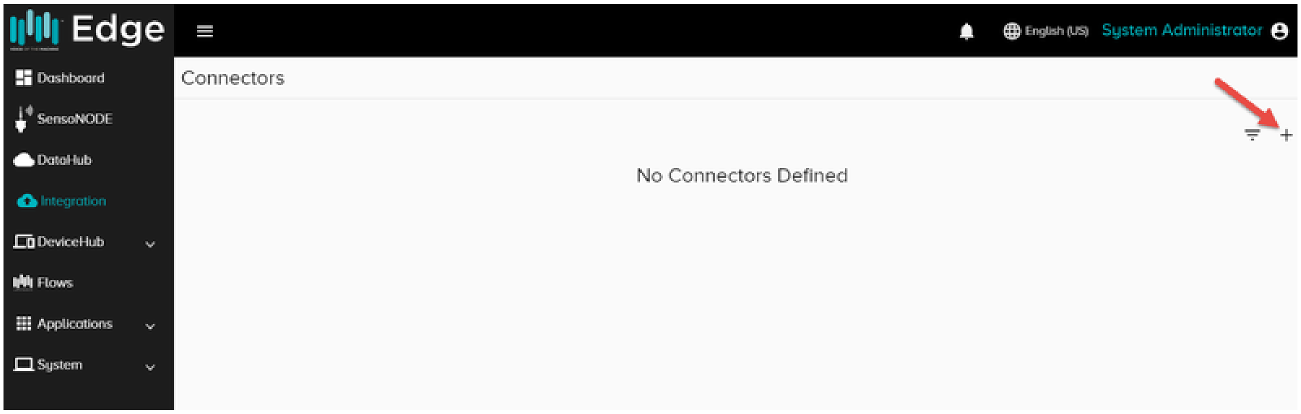
Voice of the Machine Edge lets you add connectors to enable integration with the following cloud service providers. Click the links to view each vendor's documentation

- **Microsoft Azure IoT Hub**: Connect over SSL with this cloud service to integrate with other Azure services.
- **Google Cloud IoT Core**: Integrate over SSL with Google data analytics services.
- **IBM Watson MQTT**: Send data over SSL to the Watson IoT platform, which serves as the MQTT broker, publishing messages to subscribed clients.
- **Generic MQTT**: Connect with Cloud using a generic MQTT broker (unsecure or over SSL).

Add a Third-Party Connector

To add a third-party connector:

1. In the navigation panel, go to **Integration**.



Connectors

No Connectors Defined

2. Click the **+** at the top right of the window.

3. Select the provider.

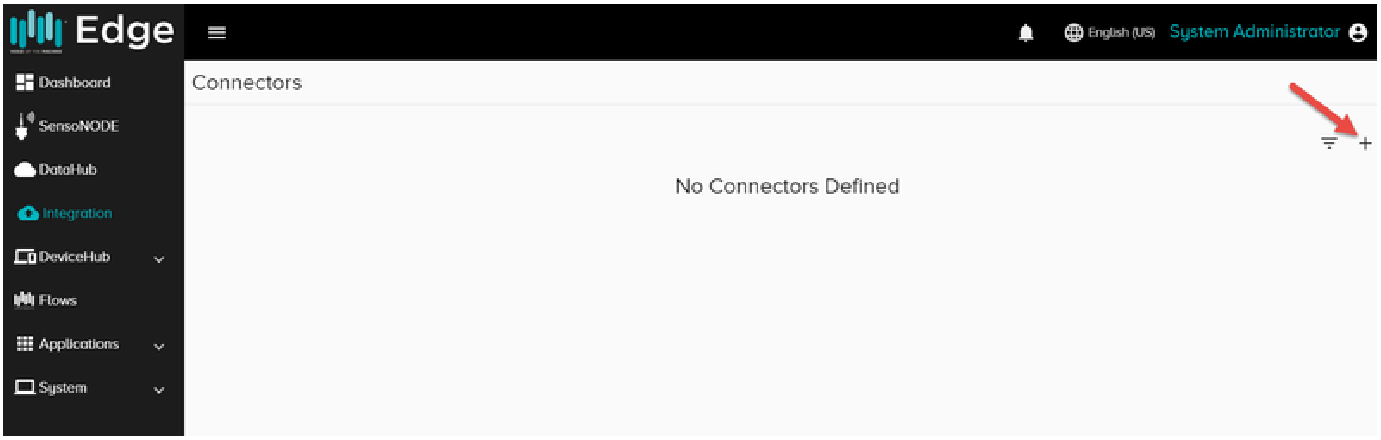
4. Complete the Connector form, as shown in the following example.
   Note: Each connector will require parameters specific to the cloud service provider. Check the provider's vendor documentation for details.

Manage Third-Party Connectors

The following options enable third-party cloud connector management:

- Edit
- Remove
- Enable/Disable

To access any of these options, click the **Actions** icon next to the listed connector.



Connectors

No Connectors Defined

How to Use Flows to Visualize Data Flows

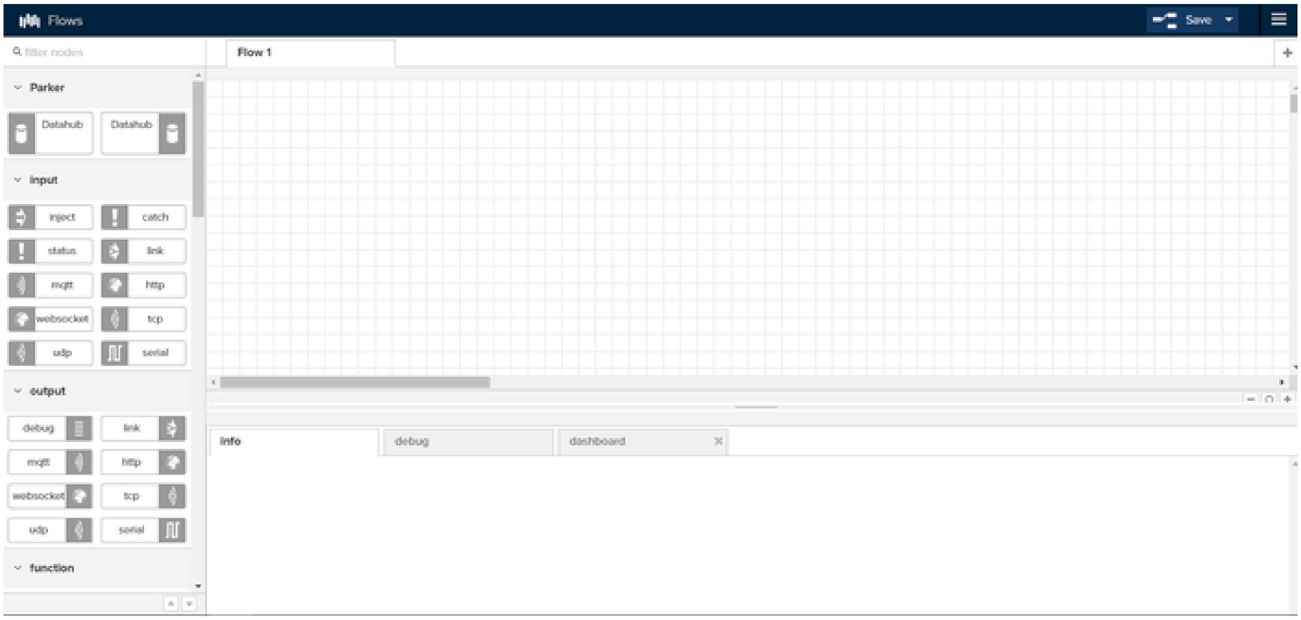
Use this drag-and-drop flow editor to connect devices. Voice of the Machine Edge Flows enables you to test and troubleshoot device connectivity. Refer to the following procedures to learn more about Voice of the Machine Edge Flows.

Navigate Voice of the Machine Edge Flows

Before taking the initial steps to create a device data flow, orient yourself to the Flow features.

To become familiar with flow configurations:

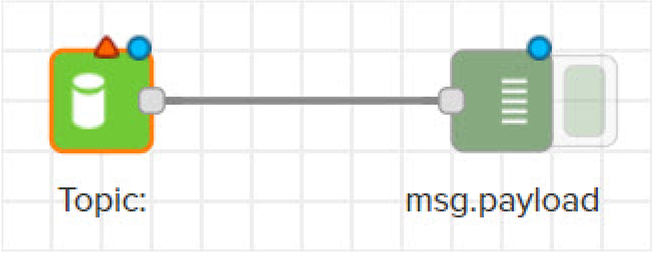
1. In the navigation panel, select **Flows**.
2. In the Flows window, click **Go to Flows**.
   You will use this blank canvas to wire together devices, simply by dragging and dropping nodes from the palette onto the canvas and then connecting them.



   - Scroll through the palette, especially to view the DataHub Publish and DataHub Subscribe nodes.
   - Hover your mouse over a node to read the tooltip description.

Create a Simple Flow

At a minimum, a simple flow consists of an input (Inject node), an output (Debug node), and a wire between them.



Test your Voice of the Machine Edge configurations by creating a flow in Voice of the Machine Edge Flows. The steps listed here enable you to connect to a local device at the edge. For this example, a DataHub node will subscribe to the local message broker with a topic for a local Modbus device, which has been configured in DeviceHub.

Prerequisites

- Create a device in DeviceHub and add a device tag so that a topic can be generated. See DeviceHub Add, Monitor, or Delete Device and DeviceHub Add a Tag.
- Check DeviceHub to verify that the device is connected.

Draw the Flow

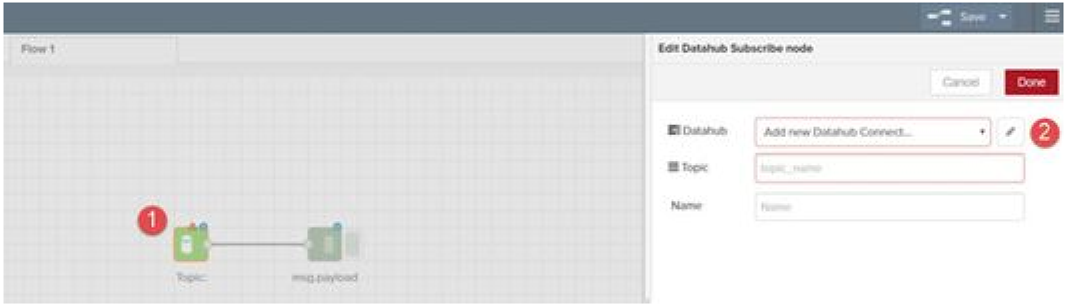
To create a flow to test your DeviceHub configuration:

1. Go to Flows.
2. Search or scroll through the palette of nodes to locate the Litmus Automation **DataHub Subscribe** and **DataHub Publish** nodes. Hover your mouse over the DataHub nodes to identify the Subscribe node.
3. Drag the **DataHub Subscribe** node onto the canvas.
4. Scroll up through the palette to the output nodes and drag the **Debug** node onto the canvas.
5. Connect the two nodes by drawing a line between them.

Configure the Flow

To configure the flow's subscribe node:

1. In the canvas, double-click the **DataHub Subscribe** node to view the parameters.



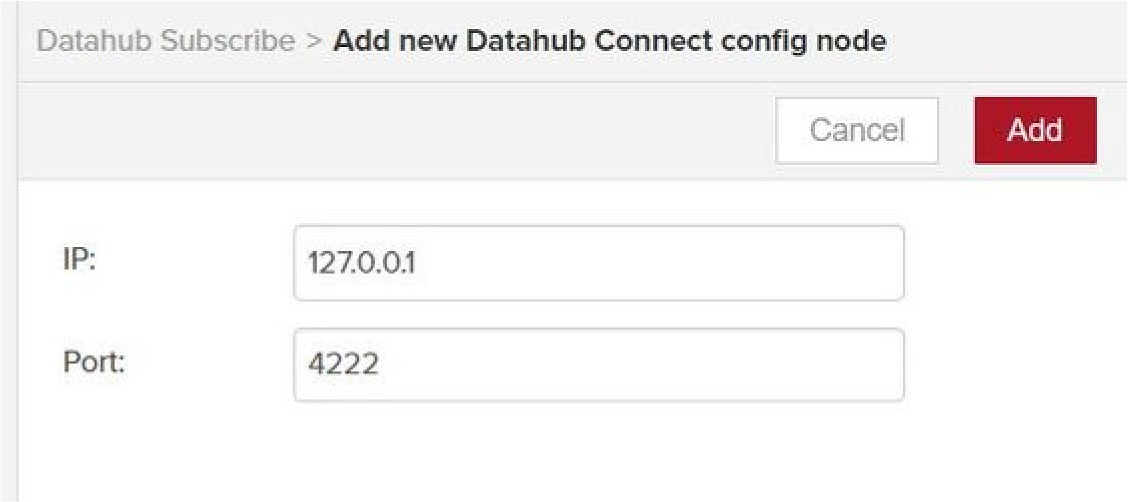
Edit Datahub Subscribe node

Datahub: Add new Datahub Connect...

Topic: topic_name

Name: Name

2. While in edit mode, click the pencil icon next to the DataHub field to view its connection parameters and click **Add**.
   a. The IP address and port fields are populated automatically. Do *not* modify these default values. Simply click **Add**.



Datahub Subscribe > Add new Datahub Connect config node

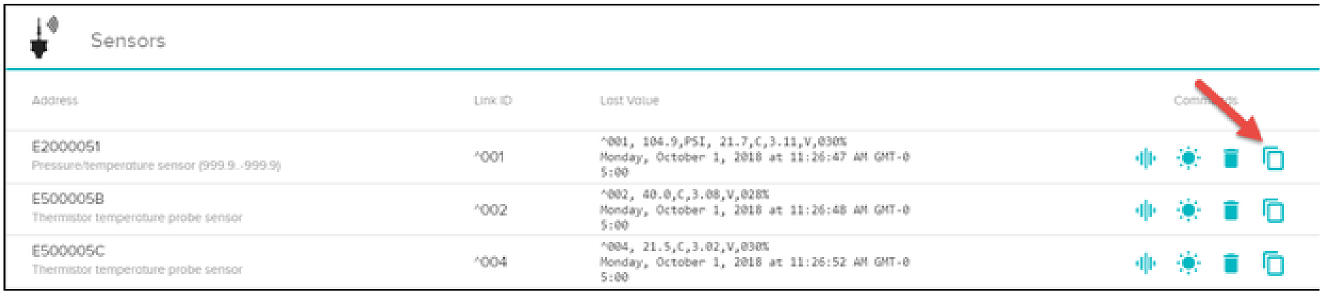
IP: 127.0.0.1

Port: 4222

3. Connect a DataHub

a. If you are connecting **SensoNODE** DataHub

i. Go to SensoNODE and scroll down to Sensors click the right-most ⧉ icon to copy.

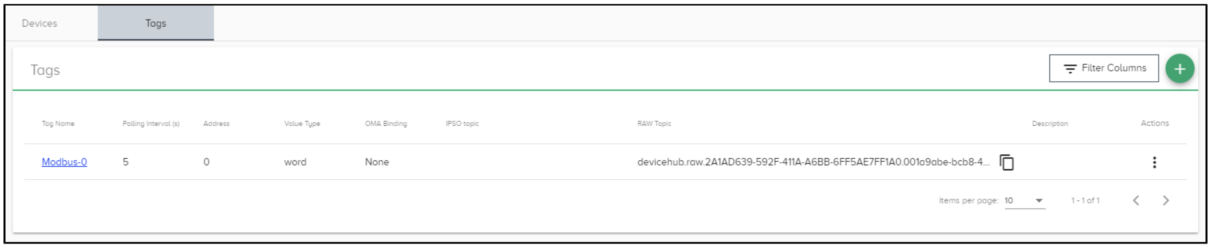


Sensors

Address	Link ID	Last Value	Commands
E2000051 Pressure/temperature sensor (999.9..-999.9)	^001	^001, 104.9,PSI, 21.7,C,3.11,V,030% Monday, October 1, 2018 at 11:26:47 AM GMT-0 5:00	
E500005B Thermistor temperature probe sensor	^002	^002, 40.0,C,3.08,V,028% Monday, October 1, 2018 at 11:26:48 AM GMT-0 5:00	
E500005C Thermistor temperature probe sensor	^004	^004, 21.5,C,3.02,V,030% Monday, October 1, 2018 at 11:26:52 AM GMT-0 5:00	

b. If you are connecting to **DeviceHub** Datahub

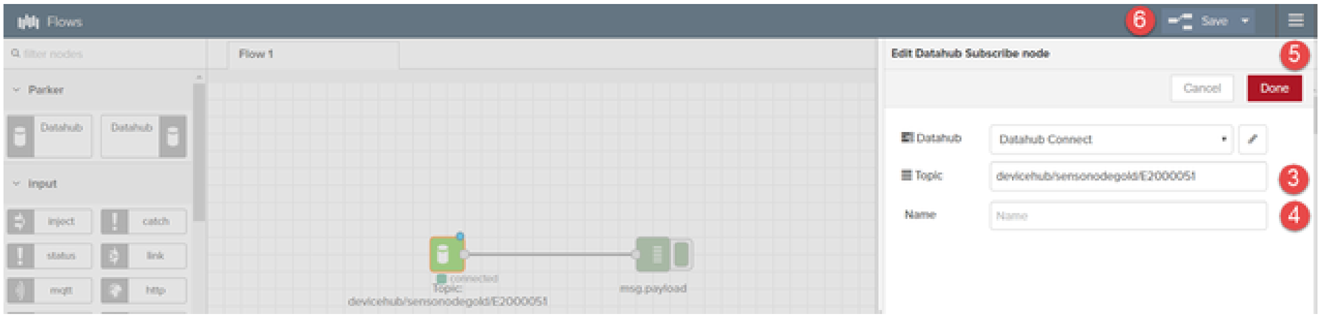
ii. Go to **DeviceHub > Tags** tab and click the right-most ⧉ icon for the tag to copy the **Raw Topic** from the device tag.



Tag Name	Polling Interval (s)	Address	Value Type	OMA Binding	IPSO topic	RAW Topic	Description	Actions
Modbus-0	5	0	word	None		devicehub.raw.2A1AD639-592F-411A-A6BB-6FF5AE7FF1A0.001a9abe-bcb8-4... ⧉		⋮

Items per page: 10 1 - 1 of 1

c. Once the topic is copied to the clipboard, you can paste it into the Topic of the flow's DataHub Subscribe node, as shown



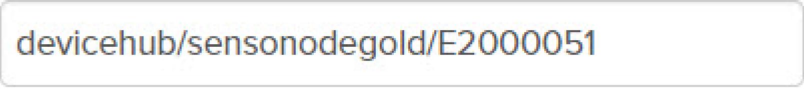
Topic: devicehub/sensonodegold/E2000051

Datahub: Datahub Connect

Topic: devicehub/sensonodegold/E2000051

Name: Name

If unable to copy the topic from the SensoNODE. Enter the format below. Replace E2000051 with the your sensor serial number.



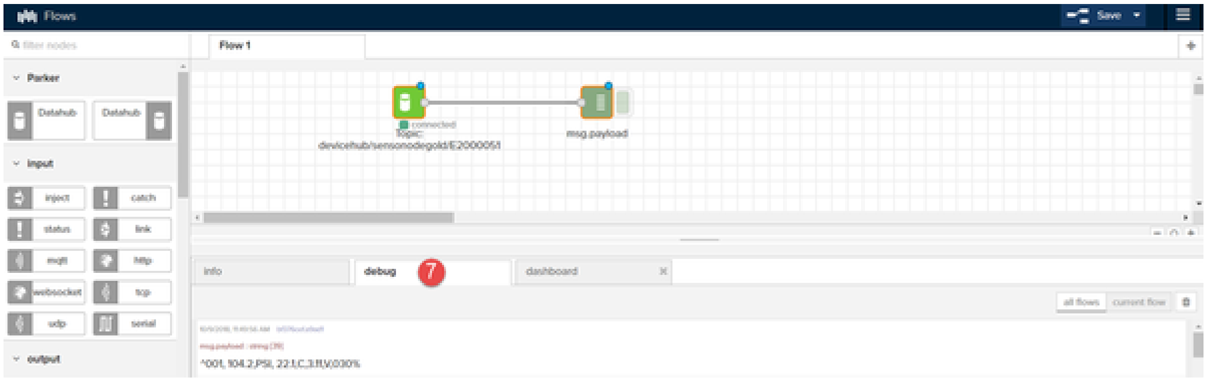
devicehub/sensonodegold/E2000051

4. Enter a **Name** for the **DeviceHub Subscribe** node.

5. Click **Done**.

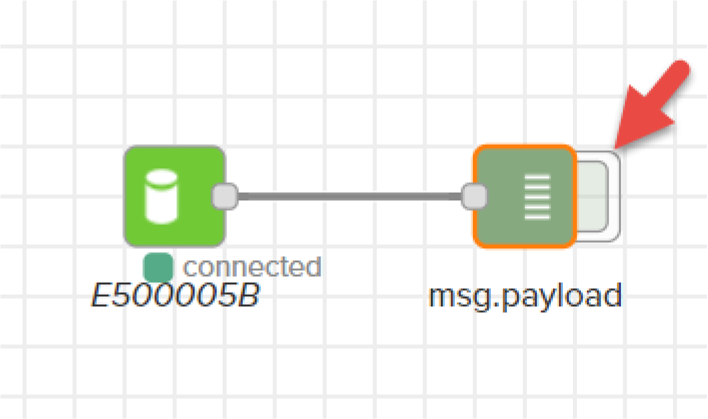
6. Click **Save**.This last *Save* step can be easily overlooked, but it is essential to successfully connect to the device and to retrieve data from the device.

7. At the bottom of the Flow canvas, click the **Debug** tab to view the messages received from the DeviceHub device.



```
msg.payload : string [39]
^001, 104.2,PSI, 22.1,C,3.11,V,030%
```

Turn off the flow by clicking to the right of the msg.payload(Debug Node).



Related articles

- Use Cases for Flow
- How to Use Flows Node Type
- How to Use Flow Options
- How to Use Flows Dashboard
- How to Use Flows to Visualize Data Flows

How to Use Flows Dashboard

Identify Dashboard Elements

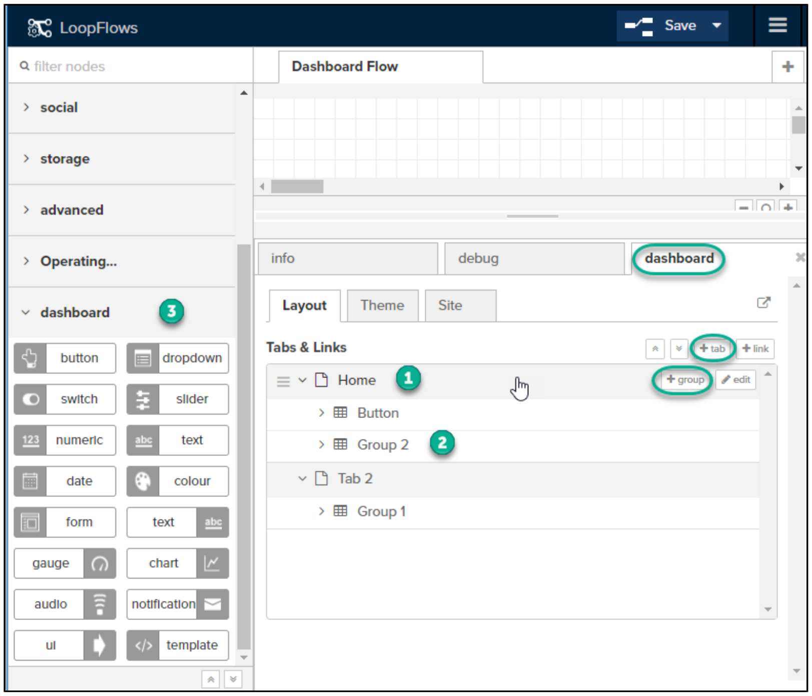
> These procedures for creating flow dashboards assume that you already have worked with flows.

Flows offers easy drag-and-drop dashboard construction, but before you get started, it is useful to understand the elements used to create a dashboard.

	Dashboard Element	Description
1	**Tab**	A dashboard can have multiple tabs, which enable navigation to different pages.
2	**Group**	Use a group as a building block to organize display nodes on a page. In a web UI context, this grouping often is referred to as a portlet.
3	**Nodes**	Drag Dashboard Nodes onto the Canvas and designate the *tab* and *group* to which the nodes will belong.

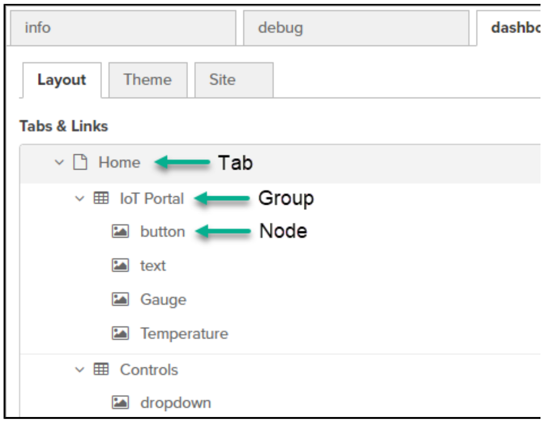
For an overview of these dashboard elements, go to the upper right actions icon ☰ and select **View > Dashboard**. Then, click the **Layout** tab to view the dashboard configuration below the canvas.

Example of Dashboard Layout View



> The **+ tab**, **+ link**, **+ group**, and **edit** tabs in the dashboard layout view are visible only when you mouse over the tab/group row.

Example of Basic Flow Dashboard Eleme

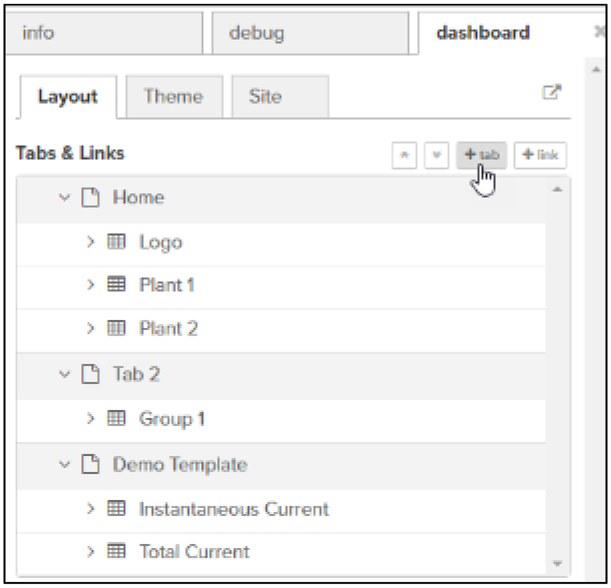


Add or Edit a Tab

The easiest way to add or edit a dashboard tab is via the sidebar dashboard tab.

Add a Tab

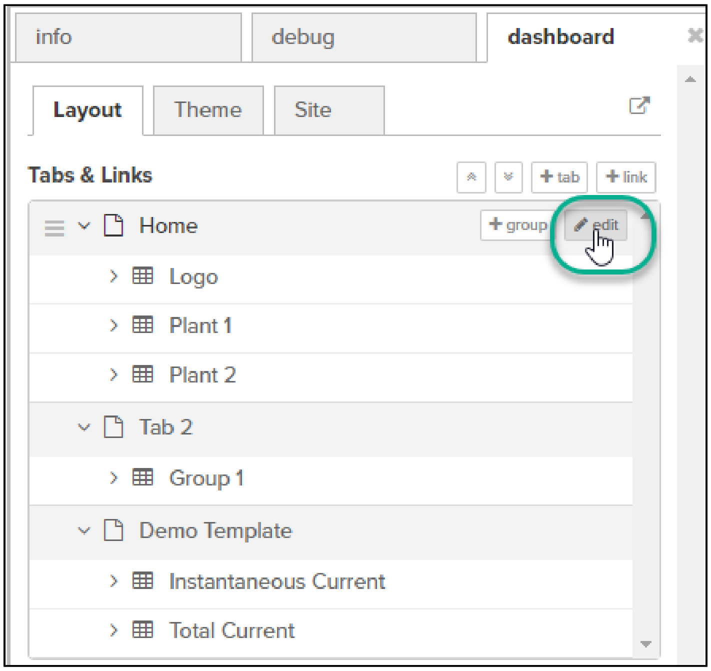
1. In the upper right actions ☰ list, select **View > Dashboard**. Then, click **Layout**.
2. Simply click **+ tab** and a new tab will be created.



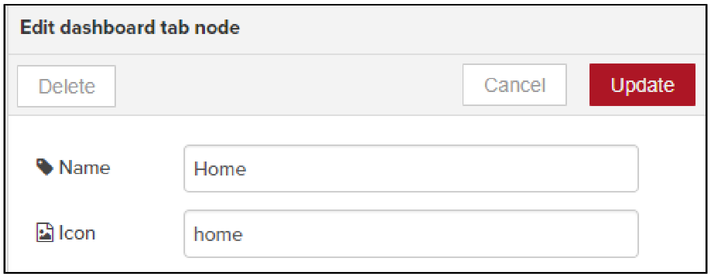
3. In the Flows main canvas view, click **Save**.

Rename a Tab

1. In the dashboard layout view, mouse over the tab row and click **edit**.



2. Modify the name and/or icon and click **Update**.



Name: Home

Icon: home

3. In the Flows main canvas view, click **Save**.

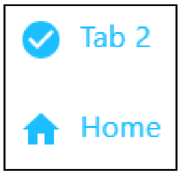
Change a Tab Icon

Tab icons can be modified by selecting alternate icons from these sources:

- Material Design Icons
- Font Awesome Icons

To use a different tab icon:

1. Mouse over the tab row and click **edit**.
2. Enter the name of the icon (found in the sources listed above), such as `check_circle` or `home`.
3. Click **Update**.
4. Be sure to click **Save**.



Add or Edit a Group

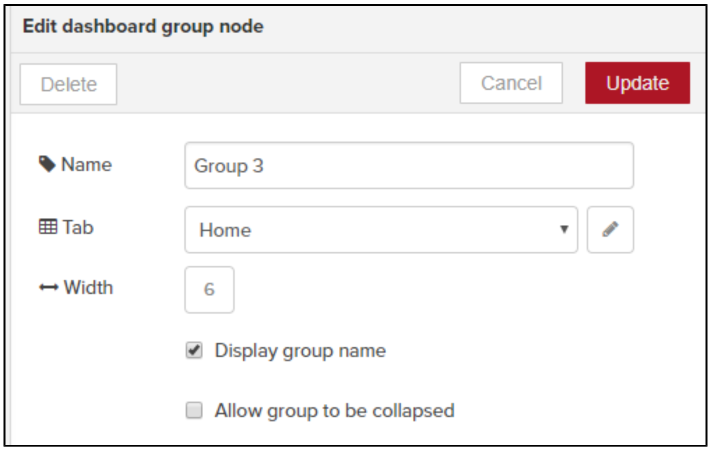
The easiest way to add or edit a dashboard group is via the sidebar dashboard view.

Add a Group

1. In the upper right actions ☰ list, select **View > Dashboard**. Then, click **Layout**.
2. Select a tab and click **+ group**. A new group will be created.
3. In the Flows main canvas view, click **Save**.

Edit a Group

1. In the dashboard layout view, mouse over the group row and click **edit**.
2. Modify the Name and/or the Tab to which the group should belong.
3. Click **Update**.
4. In the Flows main canvas view, click **Save**.



Edit dashboard group node

Name: Group 3

Tab: Home

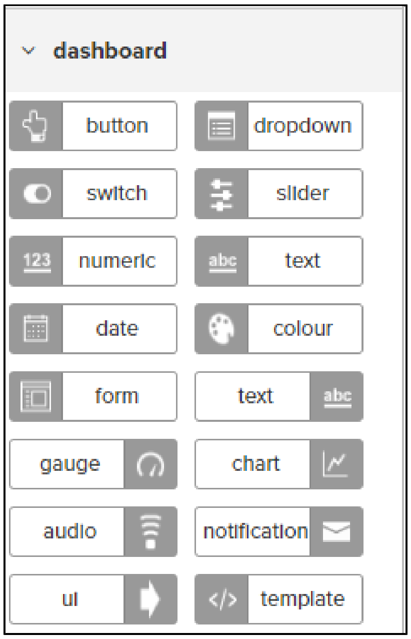
Width: 6

☑ Display group name

□ Allow group to be collapsed

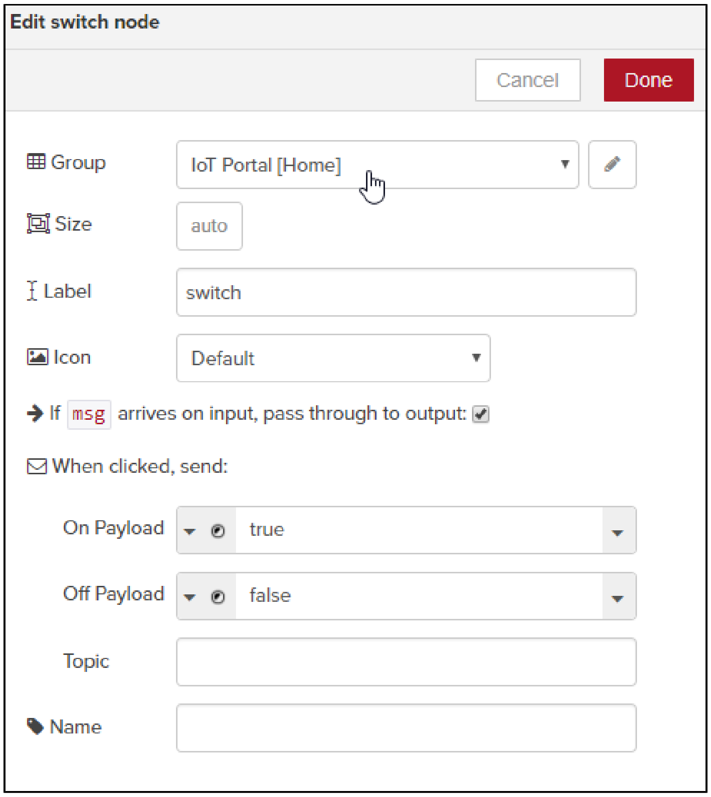
Drag Dashboard Nodes onto the Canvas

The Flows dashboard palette includes the following nodes, which when dragged onto a canvas, become widgets in a dashboard. These nodes enable you to build a dashboard with common elements, such as buttons, dropdown lists, and charts. When you combine these nodes with data collection flows, you can monitor sensor data in a single-pane-of-glass view.



Configure a Dashboard Node

When you add a dashboard node to the canvas, you must double-click it to configure its settings. Most of the dashboard nodes must be assigned to a dashboard Group, as shown in the following example.



Edit switch node

Cancel | Done

Group: IoT Portal [Home]

Size: auto

Label: switch

Icon: Default

If msg arrives on input, pass through to output: ☑

When clicked, send:

On Payload: true

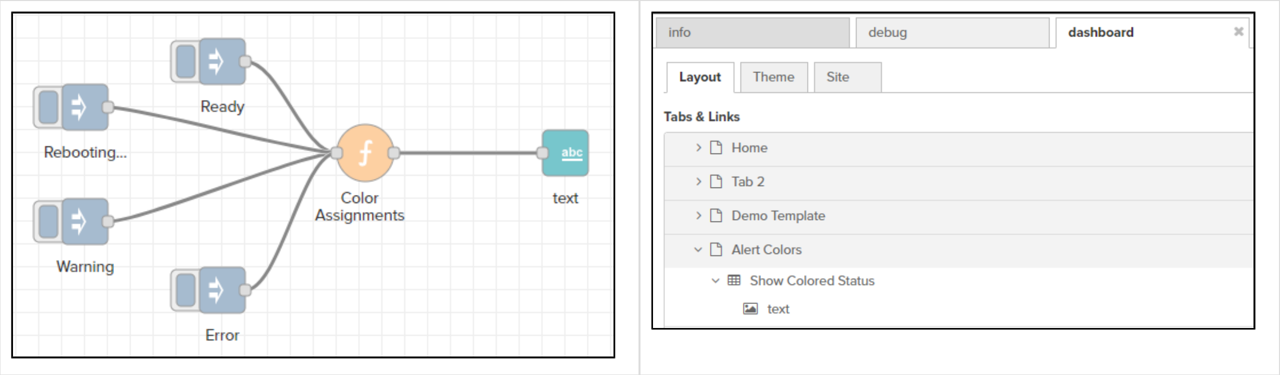
Off Payload: false

Topic:

Name:

Customize Message Text Colors

There may be times when you want to assign colors to displayed messages, particularly when monitoring device status. The following example illustrates how to configure a flow and dashboard to change the message color based on the device status: Ready, Rebooting, Warning, or Error.



For your convenience, the following code can be copied and imported into Flows so that you can explore the configuration.

1. Copy the following code block to the clipboard.
2. In Flows, import from the clipboard.

```
[{"id":"7b889445.29cbfc","type":"ui_text","z":"21b9c2dc.7bf46e","group":"f7e07b80.a8b2c8","order":0,"width":"0","height":"0","name":"","label":"","format":"<font color= {{msg.color}} > {{msg.payload}} </font>","layout":"col-center","x":640,"y":220,"wires":[]},{"id":"4b8181b0.ec8ab","type":"inject","z":"21b9c2dc.7bf46e","name":"","topic":"","payload":"Ready","payloadType":"str","repeat":"","crontab":"","once":false,"x":340,"y":140,"wires":[["435cd506.e398bc"]]},{"id":"435cd506.e398bc","type":"function","z":"21b9c2dc.7bf46e","name":"Color Assignments","func":"switch (msg.payload) {\n\ncase \"Ready\" : \n    msg.color = \"lime\";\n    break;\n    \ncase \"Warning\" : \n    msg.color = \"orange\";\n    break;\n    \ncase \"Rebooting...\" : \n    msg.color = \"blue\";\n    break;\n    \ndefault : \n    msg.color = \"red\";\n    break;\n}\n\nreturn msg;","outputs":1,"noerr":0,"x":465,"y":220,"wires":[["7b889445.29cbfc"]]},{"id":"f3fc7605.077428","type":"inject","z":"21b9c2dc.7bf46e","name":"","topic":"","payload":"Warning","payloadType":"str","repeat":"","crontab":"","once":false,"x":220,"y":280,"wires":[["435cd506.e398bc"]]},{"id":"37d6f0de.eb7e9","type":"inject","z":"21b9c2dc.7bf46e","name":"","topic":"","payload":"Rebooting...","payloadType":"str","repeat":"","crontab":"","once":false,"x":220,"y":180,"wires":[["435cd506.e398bc"]]},{"id":"29b7a81c.d345c8","type":"inject","z":"21b9c2dc.7bf46e","name":"","topic":"","payload":"Error","payloadType":"str","repeat":"","crontab":"","once":false,"x":340,"y":340,"wires":[["435cd506.e398bc"]]},{"id":"f7e07b80.a8b2c8","type":"ui_group","z":"","name":"Show Colored Status","tab":"55213901.3e2a88","disp":false,"width":"6","collapse":false},{"id":"55213901.3e2a88","type":"ui_tab","z":"","name":"Alert Colors","icon":"dashboard"}]
```

Nodes		Steps to configure the nodes and dashboard
Inject	timestamp	This node can be found in the **Input** list. This node is used primarily to demonstrate this functionality. In a production environment, device messages would need to be parsed to extract status values. 1. Drag an Inject node onto the canvas for every status message to be displayed. 2. Set the Payload to text and enter the status, such as Warning. 3. Click **Done**, then click **Save**.

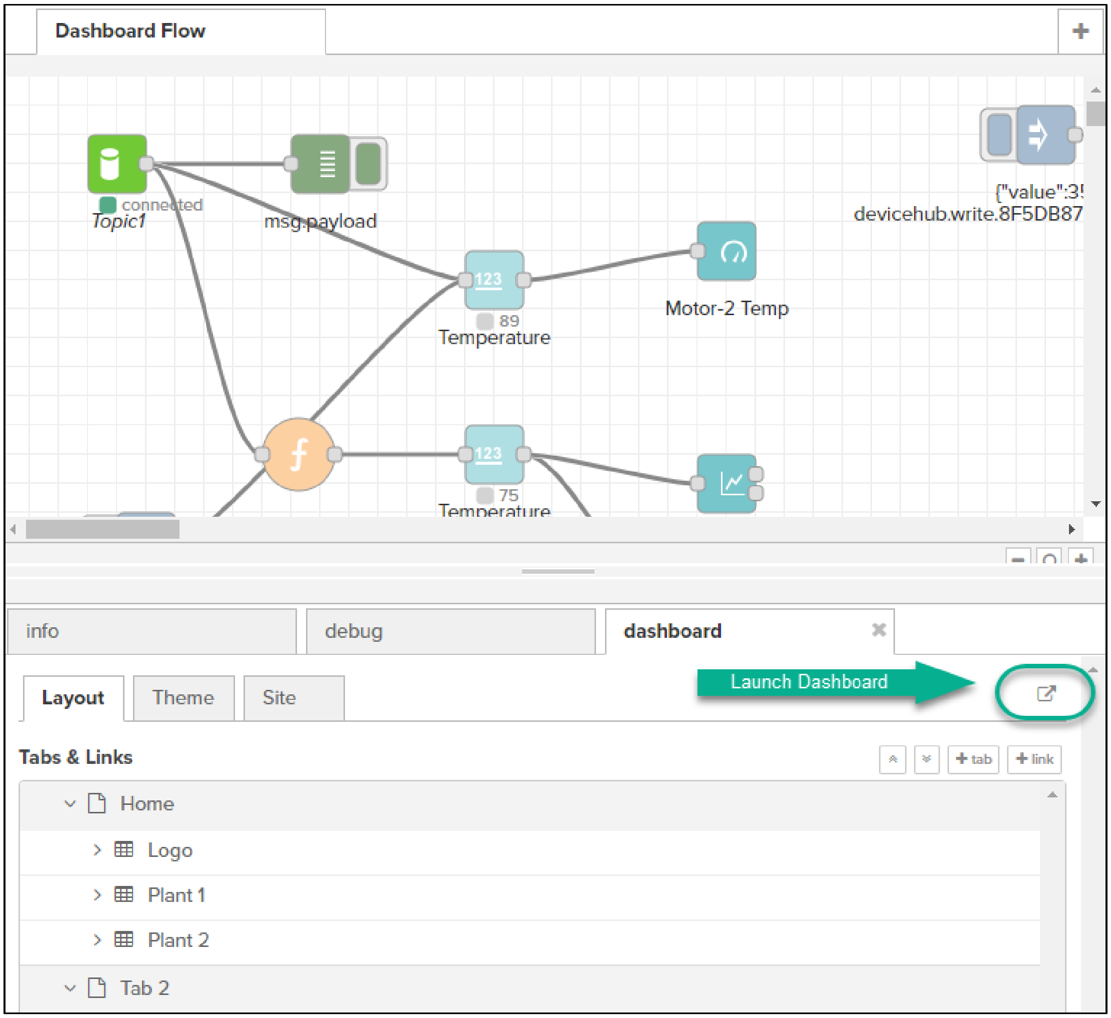
Function	ƒ	This node can be found in the **Function** list. Use this node to assign a color to each status message.
		1. Drag a Function node onto the canvas and wire it to the Inject node.
		2. Enter code for the function. For this status message use case, the following code was used.
		`switch (msg.payload) {`
		`case "Ready" :`
		`msg.color = "lime";`
		`break;`
		`case "Warning" :`
		`msg.color = "orange";`
		`break;`
		`case "Rebooting..." :`
		`msg.color = "blue";`
		`break;`
		`default :`
		`msg.color = "red";`
		`break;`
		`}`
		`return msg;`

Text	abc text	This ui_text node can be found in the **Dashboard** list. Note that there are two text nodes in the Dashboard list. This is not the text input field. 1. Drag the ui_text node onto the canvas and wire it to the Function node. 2. In the **Value Format**, enter: `<font color= {{msg.color}} > {{msg.payload}} </font>`

Launch the Flow Dashboard

Once the dashboard elements have been configured, you can review the results in the dashboard.

In the **dashboard** tab, click the launch icon at the upper right of the sidebar dashboard view, displayed below the canvas.



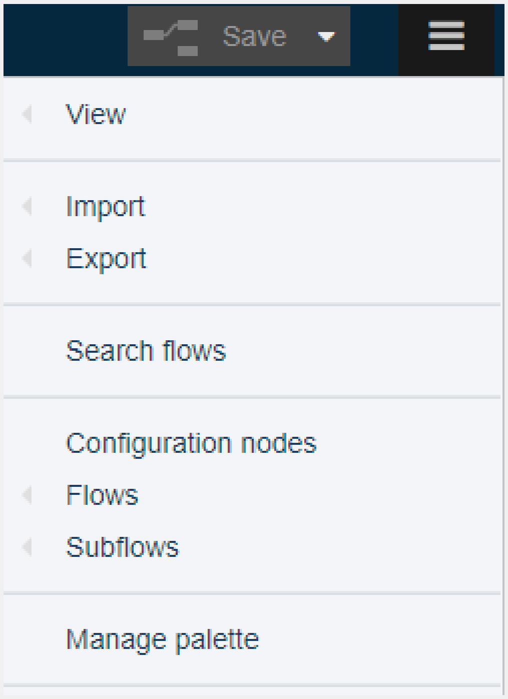
The URL for the dashboard will have the format for your IP address: `https://nnn.nnn.nnn.nnn/flows/api/ui/#/0`

Related articles

- Use Cases for Flow
- How to Use Flows Node Type
- How to Use Flow Options
- How to Use Flows Dashboard
- How to Use Flows to Visualize Data Flows

How to Use Flow Options

Voice of the Machine Edge Flows options can be accessed by clicking the **list** icon (**)** at the top right of the Flows window.

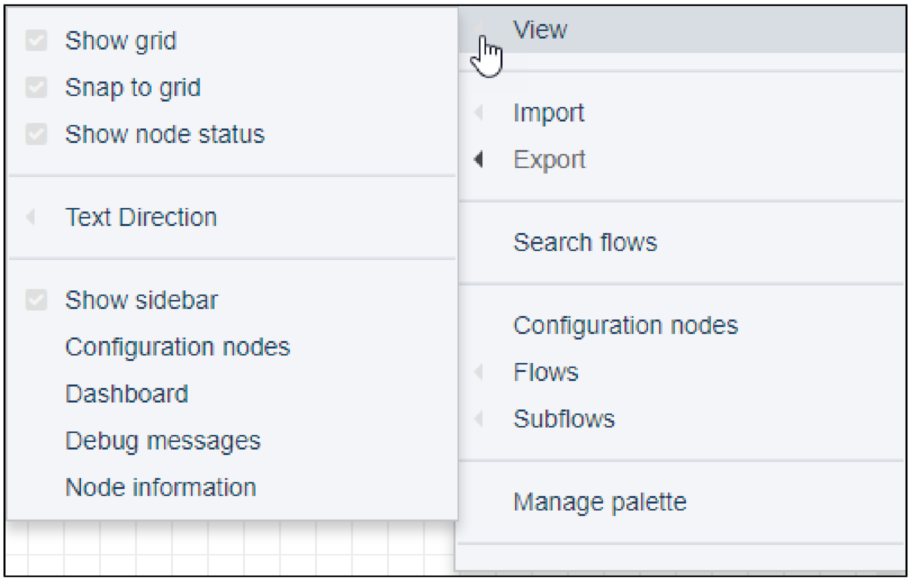


Flows options

- Customize the Flows View
- Export and Import Flows
- Search for Flows
- Manage Configuration Nodes
- Add, Rename, Delete Flows
- Create Subflows
- Manage the Nodes Palette
- Select Flow Nodes
- Save Flow Options

Customize the Flows View

Several options enable customization of the flows view.



Click the **list** icon (**)** at the top right of the Flows window.

The following options can be toggled on and off via **> View**:

Show grid: Grid displays by default. Toggle the option to remove the grid lines.

Snap to grid: Snap to grid is enabled by default. Toggle the option to disable/enable alignment of nodes with grid lines.

Show node status: The publish and subscribe nodes, for example, display status such as *connected* or *waiting for message*. Toggle the option to disable/enable the status.

Show sidebar: Actually, the sidebar in the Flows window is *beneath* the canvas. Best practices recommend keeping this message display area enabled. The sidebar includes the tabs:

- **Info**: Description of a selected node and its functionality. This tab is always enabled.
- **Debug**: Messages that can be used to verify a flow's logic. This tab is always enabled.
- **Dashboard**: User-configured dashboard of the flow logic. Enable this tab with **> View > Dashboard**.
- **Config**: Nodes that are used/unused in flows. Enable this tab with **> Configuration nodes**.

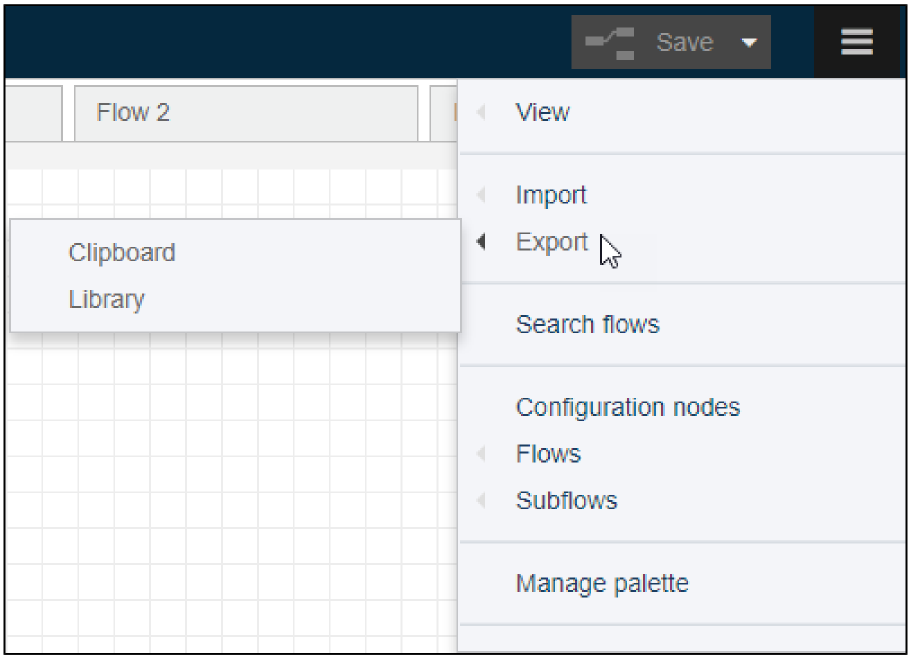
Text Direction: Configure the way text will appear in node fields, when configuring or editing a node.

- **Default**: Left-to-right text.
- **Left-to-righ**t: As you type, the text will be appended to the left-justified text string.
- **Right-to-left**: As you type, the text will be appended to the right-justified text string.
- **Contextual**: Text direction is determined by the type and shape of the node.

Export and Import Flows

Exporting and Importing enables ease-of-deployment of flow logic for a variety of devices and use cases.

Click the **list** icon (**)** at the top right of the Flows window.



Export Nodes

The **Export** option offers two choices: **Clipboard** and **Library**.

As the name implies, the clipboard option temporarily saves node data to the clipboard. To save node data for a longer term, use the Export to Library option.

Note: When exporting, passwords are *not* exposed.

Export to Clipboard

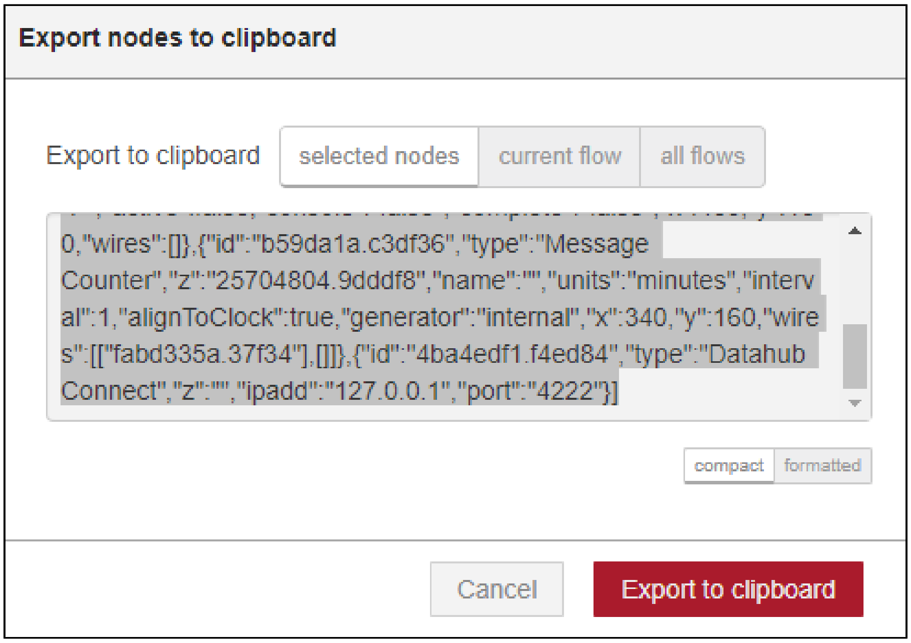
Prerequisite

- Before exporting nodes, select the nodes and connecting wires within a flow tab. See Select Flow Nodes.
- Even though you only select nodes in the current flow, the dialog lists an option to export *all flows*.

When exporting nodes to the clipboard, choose from these three options and click **Export to clipboard**:

- **current flow** - Copy all nodes in the current flow, not just the selected nodes, to the clipboard.
- **all flows** - Use caution when selecting this option because when you import the clipboard contents, all of the flows will be duplicated.
- **selected nodes** - Select nodes within a flow (see Select Flow Nodes) and then select **> Export > Clipboard**.
  The JSON schema can be viewed in **compact** or **formatted** mode.

When you export to the clipboard, as with any copy/paste operation, the nodes now are available to paste into a flow.

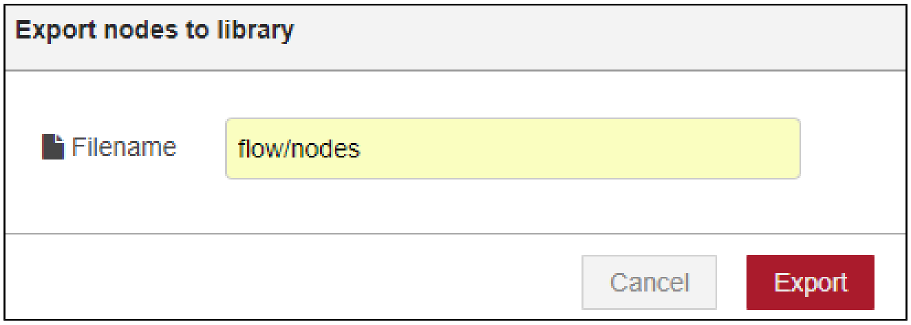


```
0,"wires":[]},{"id":"b59da1a.c3df36","type":"Message
Counter","z":"25704804.9dddf8","name":"","units":"minutes","interv
al":1,"alignToClock":true,"generator":"internal","x":340,"y":160,"wire
s":[["fabd335a.37f34"],[]]},{"id":"4ba4edf1.f4ed84","type":"Datahub
Connect","z":"","ipadd":"127.0.0.1","port":"4222"}]
```

Export to Library

Prerequisite
Before exporting nodes, select the nodes and connecting wires within a flow tab. See Select Flow Nodes.

When you export nodes to a library, you are prompted to enter a path and filename for the library. This location is created dynamically and does *n ot* need to be a pre-existing directory or file.



Filename: flow/nodes

If you re-use the path/filename, the previous export will be overwritten.

Import Nodes

The **Import** operation relies on a previous export. Just as with export, the import operation lists two options: **Clipboard** and **Library**.

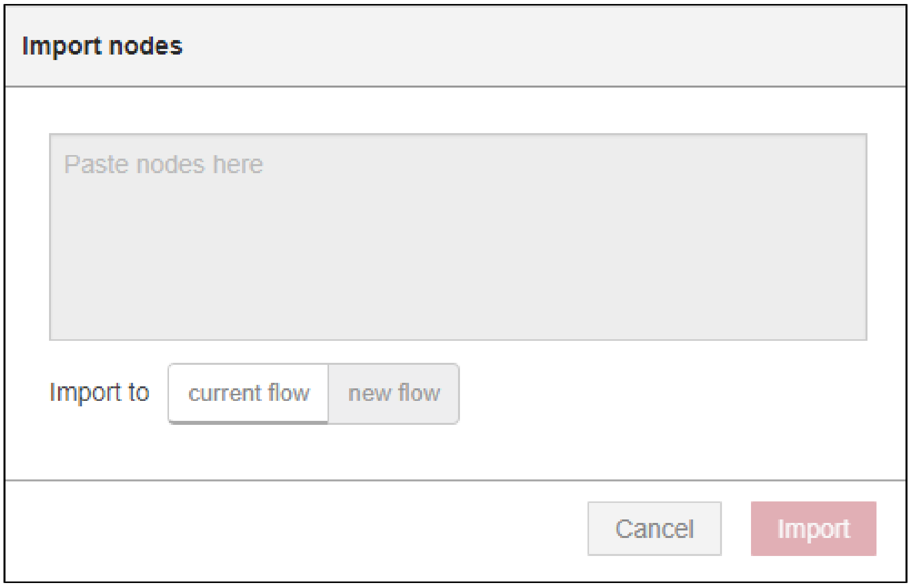
Import from Clipboard

The clipboard option provides access to data temporarily saved to the clipboard. For data saved for a longer term, use the Import from Library option.

1. When importing nodes from the clipboard, choose from these options:

- **current flow** - Import the nodes that were exported to the clipboard into the currently selected tab. If you selected **current flow** when you exported the nodes, the import will create a new flow tab.
- **new flow** - Create a new tab that will contain the imported nodes.

2. Paste the contents of the clipboard (CTRL-V) into the dialog and click **Import**.



Be sure to **Save** the flow after importing from the clipboard.

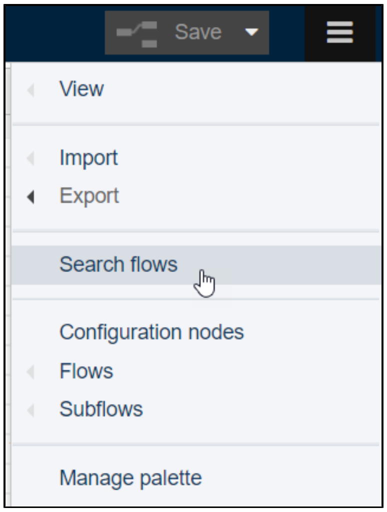
Import from Library

This option simply imports the nodes that were exported to a specific library (path/filename).
Be sure to **Save** the flow after importing from a library.

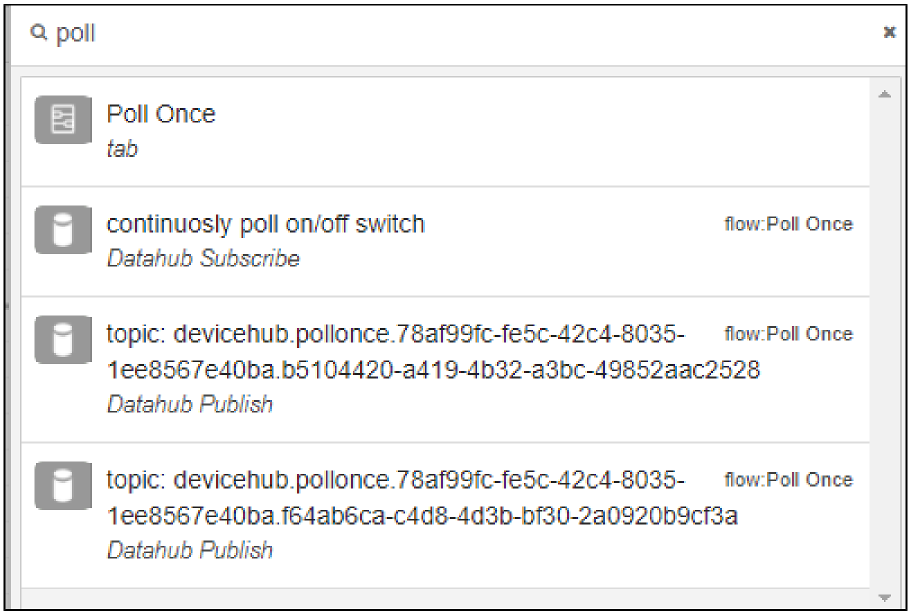
Search for Flows

When searching for a flow, the search engine supports partial search.

1. To search for specific flows in the Flows window, go to **> Search flows**.



2. Enter search text.
   The search results include not only flow tabs, but also nodes that contain the search text, as illustrated in the following screen capture.



poll

Poll Once
tab

continuosly poll on/off switch — flow:Poll Once
Datahub Subscribe

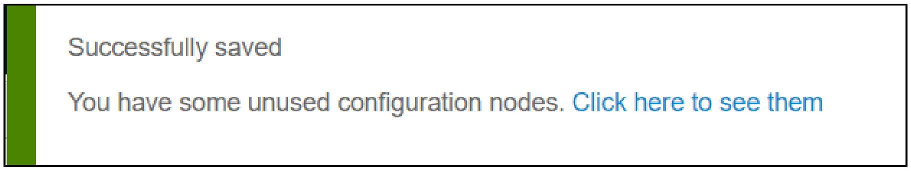
topic: devicehub.pollonce.78af99fc-fe5c-42c4-8035-1ee8567e40ba.b5104420-a419-4b32-a3bc-49852aac2528 — flow:Poll Once
Datahub Publish

topic: devicehub.pollonce.78af99fc-fe5c-42c4-8035-1ee8567e40ba.f64ab6ca-c4d8-4d3b-bf30-2a0920b9cf3a — flow:Poll Once
Datahub Publish

Manage Configuration Nodes

When you select Configuration Nodes, a **config** tab opens beneath the canvas to display used and unused nodes. When you delete a node from a flow, it may not be deleted from the backend.

- Server-based nodes that require network/connectivity settings, such as MQTT nodes, will be found here if they have been deleted from a flow. Delete these types of unused nodes from the backend because they may continue to try connecting to a port and degrade performance.
- Not all deleted nodes will appear in the config window.
- When you save a flow or the entire workspace, a pop-up message notifies you if there are unused nodes.

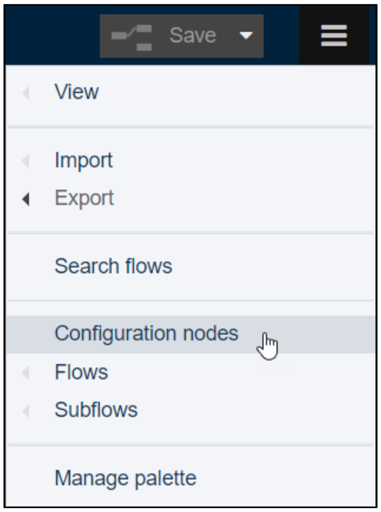


Successfully saved

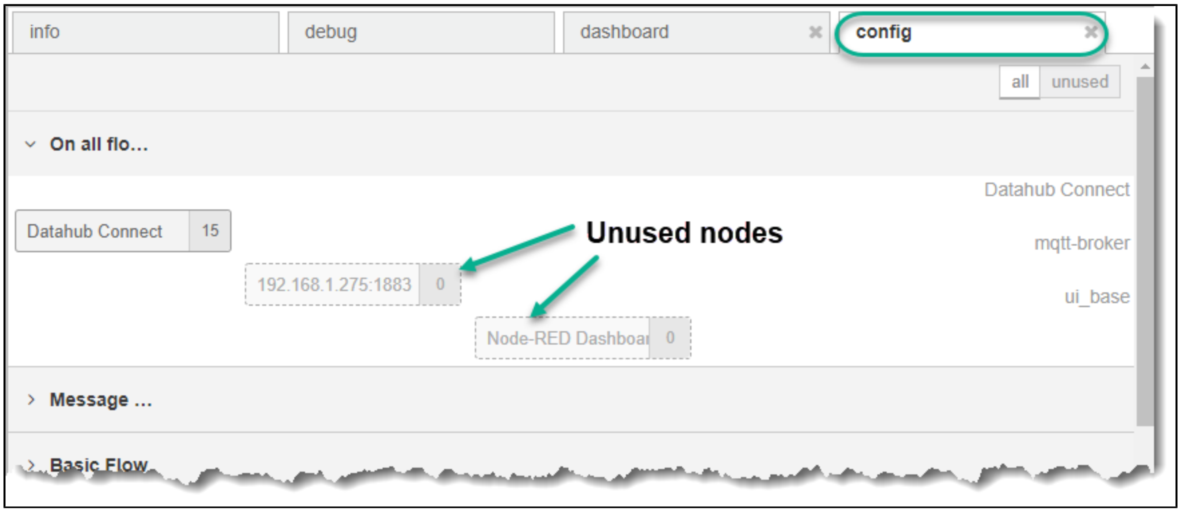
You have some unused configuration nodes. Click here to see them

To identify and remove unused nodes:

1. Click the **list** icon (**)** at the top right of the Flows window.
2. Select **Configuration nodes** from the **Flows** list.



A **config** tab will open beneath the flows canvas.

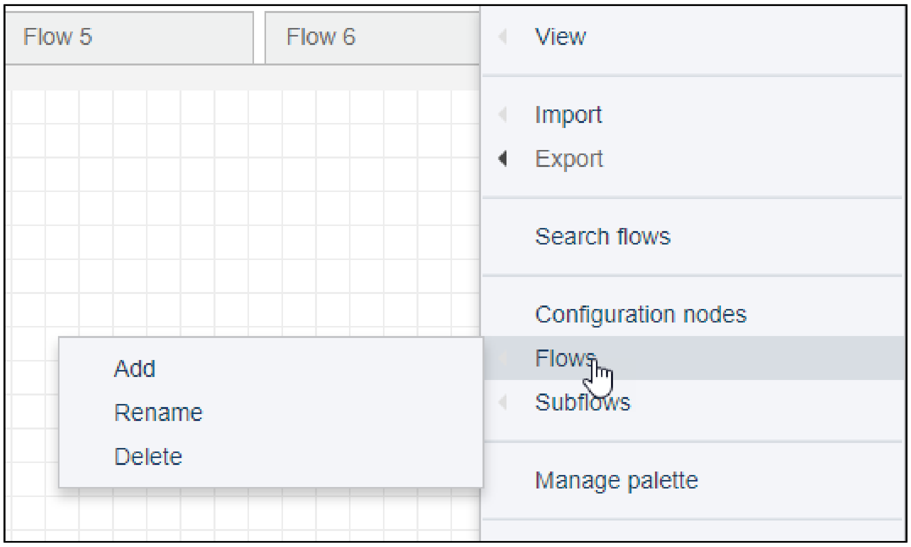


3. Identify unused nodes by the **dotted-line border** and a **0** for the number of instances.
4. Double-click an unused node to launch its editing dialog and then click **Delete**.

Add, Rename, Delete Flows

Click the **list** icon (**)** at the top right of the Flows window.

From the **Flows** list, the following options are available:



- **Add** - This operation provides an alternative to using the **+** to create a new flow canvas. To add a flow, select **Flows > Add**.
- **Rename** - Flows, by default, are given a generic name, such as Flow 5, which is displayed in the tab. Use the rename option to give a canvas a meaningful name. Select a flow tab and then from the list, select **Flows > Rename**.
- **Delete** - When you select a tab, that flow becomes the current flow. To delete a flow, select the tab and then from the list, select **Flows > Delete**.

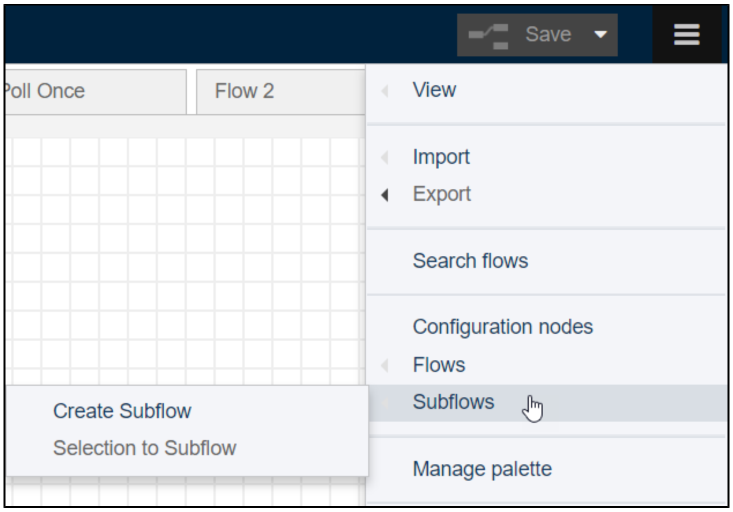
Be sure to click **Save** to commit the Add, Rename, and Delete actions.

Create Subflows

Subflows serve as subroutines to place complex logic into code blocks, which then can be called from the main flow. As with any structured programming language, this modular approach simplifies troubleshooting, modifications, and basic understanding.

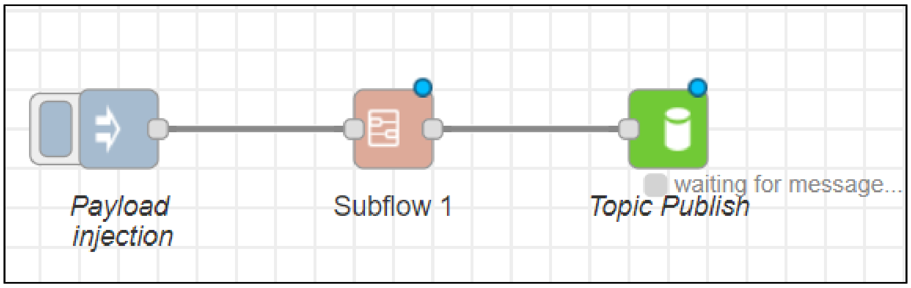
To create a subflow:

1. In the flows window, select **> Subflows > Create Subflow**.



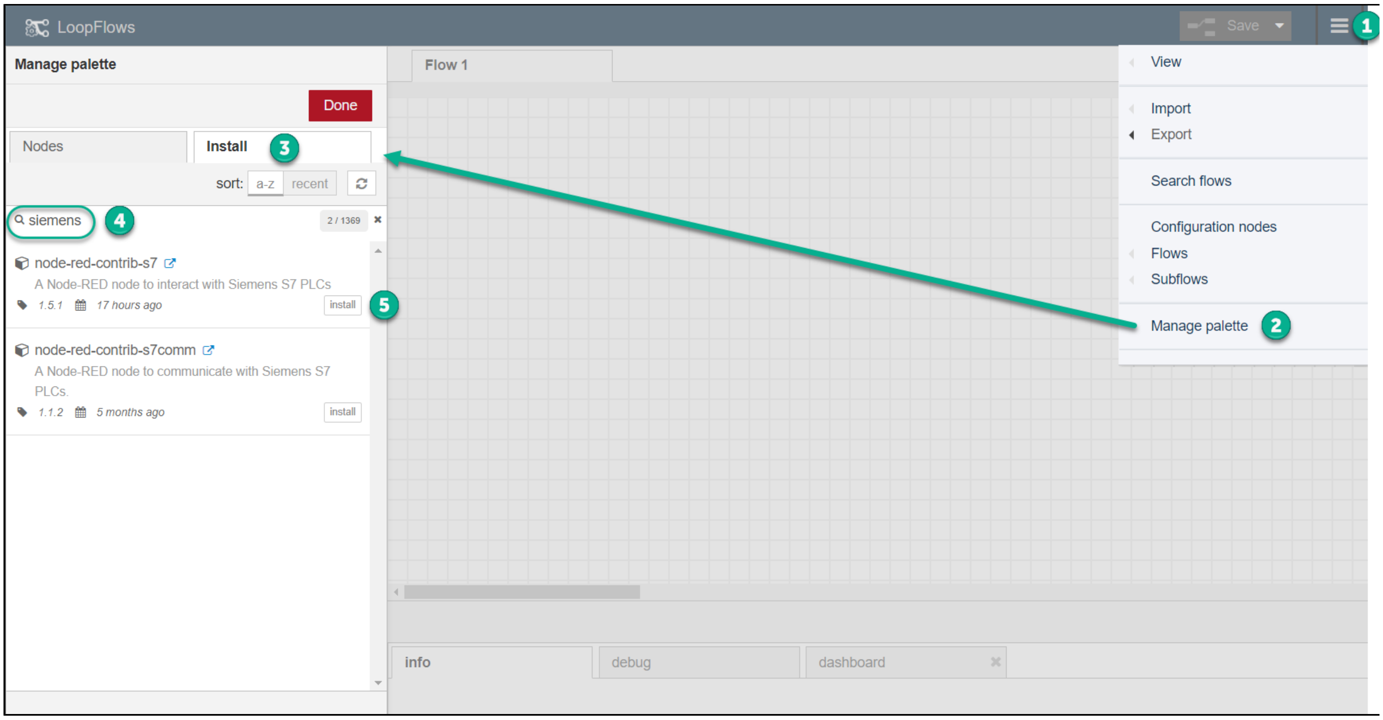
2. Drag and connect nodes in the canvas.
3. Alternatively, while creating the main flow, select the logic that you want to place into a separate code block and then select **> Subflows > Selection to Subflow**.

Once a subflow is created, it becomes available in the node palette and can be used as a node in a flow.



Manage the Nodes Palette

In addition to the standard set of nodes shown in the palette, a variety of other nodes specific to a PLC or application can be included in the palette.
Simply search the nodes database to find the desired nodes. This illustration shows the steps to add nodes to the palette, as described in the subsequent procedure.



To install additional nodes in the Flows palette:

1. Click the **list** icon (**)** at the top right of the Flows window..
2. Select **Manage palette**.
3. In the Manage palette dialog, select the **Install** tab.
4. Enter a search string. In the above example, a search for *siemens* nodes is shown.
5. Select the desired nodes in the search results, click **Install** and then **Done**.

To find the newly added nodes:

1. In the Manage palette dialog, select the **Nodes** tab.
2. In the *filter nodes* box, enter a search string to search for the node you need.

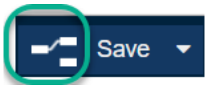
Select Flow Nodes

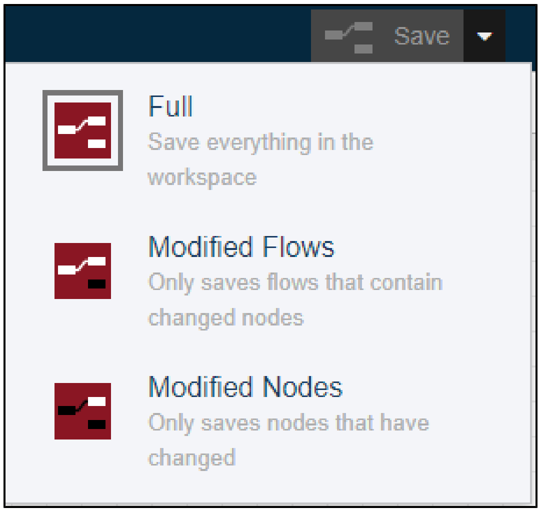
When developing flow logic, certain operations lend themselves to selecting and copying/pasting the node logic.

- To select nodes and their connectors, simply click and drag within a flow tab.
- The selected nodes will display a red outline.

Save Flow Options

When saving a flow, select one of the following options from the list.

Note that the Save icon indicates the type of save that is currently active for the Save button: 


Save
Full
Save everything in the workspace
Modified Flows
Only saves flows that contain changed nodes
Modified Nodes
Only saves nodes that have changed

- **Full** - Flow logic from all tabbed canvases in the workspace will be saved to the Voice of the Machine Edge database.
  Note: If many flows containing complex logic have been created, a full save may not be your best save option.
- **Modified Flows** - Save the flow logic from only the flows that have been modified.
- **Modified Nodes** - This is the fastest option for individual node changes, as it only saves the updated nodes.

To cancel revisions to a flow, either use CTRL-Z or close the browser page.

Use Cases for Flow

Refer to the following procedures to get an idea of how flows can be used, for example, to read from and write to device registers.

- Flow to Write to and Read a File
- Flow to Write to Tags
- Flow to Save Data to a USB Device
- Flow to Subscribe to All Topics
- Flow to Poll a Tag Once for an Event-Driven Poll
- Flow to Save Multiple Register Values
- Flow to Use Voice of the Machine Edge as an FTP Client

Flow to Write to and Read a File

Use these procedures to create a log file on the Voice of the Machine Edge device, populate it, and then read its contents. These instructions provide an example of how to use flows to write to and read data from files.

1. Create a Log File
2. Write to a File
    a. Write a Message to the File
3. Read a File
4. Video of a Flow to Write to and Read a File

Tips:

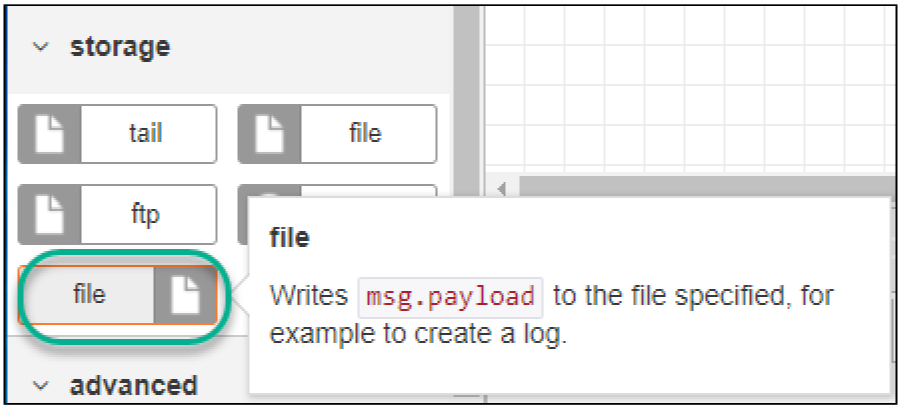
- Use the **filter nodes** search function in the node palette to find specific nodes, such as a **file** node.
- As you modify flow configurations, be sure to save the flows. Otherwise, the nodes will exist only in the flow editor and the revised flows will not be executable.
- Filenames used in node configurations are case sensitive.

Create a Log File

A file must first be created before data can be written to it.

To create a file in a flow:

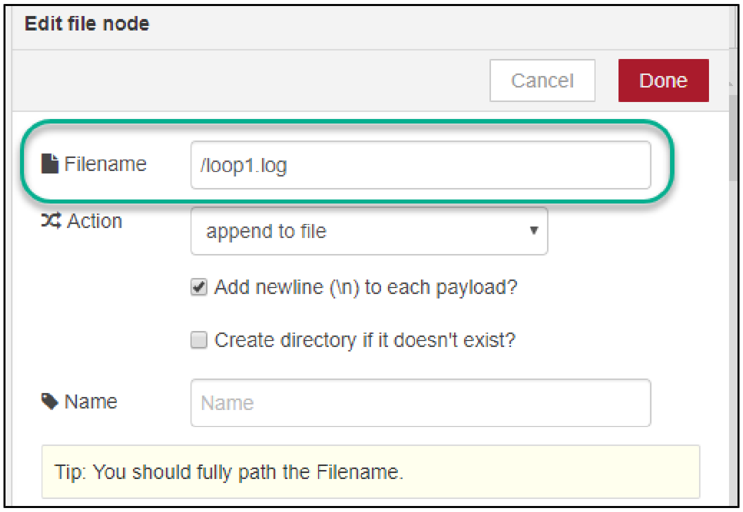
1. In the navigation panel, select **Flows**.
2. In the Flows window, click **Go to Flows**.
3. Click the **+** at the upper right of the canvas to start a new flow.
4. Drag a **file (write)** node onto the canvas.
   This node can be found under **storage** in the palette. Note that there are two file nodes: a **file in** node to read and a **file** node to write.



file

Writes `msg.payload` to the file specified, for example to create a log.

5. Double-click the **file** node and enter a **filename prefaced with a forward slash (/)**.
   The forward slash is required in the filename. For example: `/loop1.log`
   The file will be created on the Voice of the Machine Edge device. *If a subdirectory already exists*, you can supply the full path to the file. For example: `/logfiles/loop1.log`



Edit file node

Cancel | Done

Filename: /loop1.log

Action: append to file

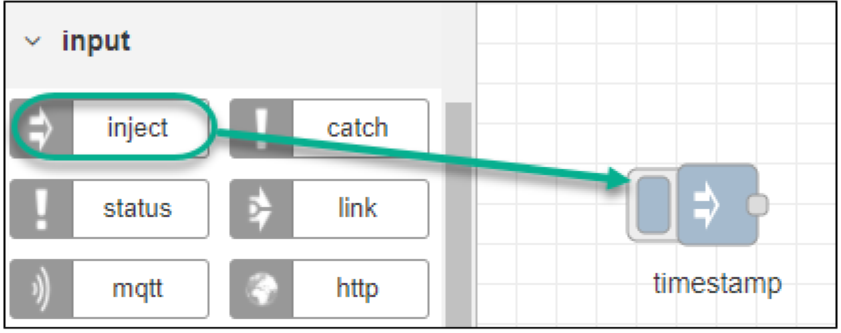
☑ Add newline (\n) to each payload?

☐ Create directory if it doesn't exist?

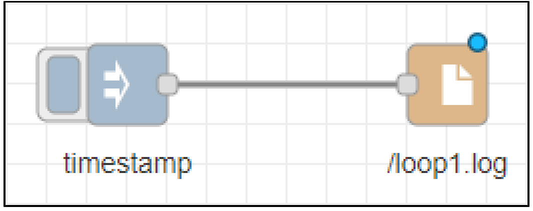
Name: Name

Tip: You should fully path the Filename.

6. Select an Action of **append to file** and click **Done**.
7. Drag an **inject** node onto the canvas.
   This node can be found under **input** in the palette. Note that it defaults to **timestamp**. This node serves as a *trigger* to create the file. Other payload options are available, but they will be covered later in advanced flow examples.



8. Wire the **inject** node to the **file** node.



9. Click **Save**, at the top right of the canvas, to save the flow.
   If there is an issue with the save action, an error message will be displayed in the debug tab area below the canvas. If the file creation is successful, no messages will be shown in the debug tab.
10. Click the button to the left of the **inject** (timestamp) node to inject the timestamp, which creates the file locally. In this example, the `/loop1.log` file is saved.
    A common error in this scenario is the omission of the forward slash in the filename. The forward slash is required.
11. Go to Read a File for the instructions to verify that the file was created.

Write to a File

In the steps listed in Create a Log File, the **inject** node was used to trigger the creation of the file. The following steps illustrate a basic example of writing to a file.

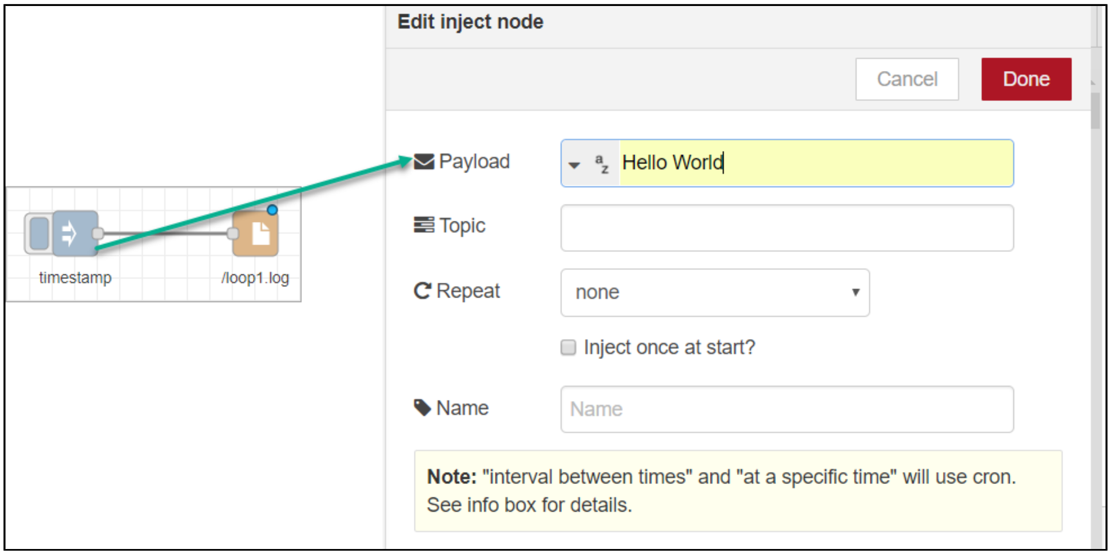
To continue writing to the file that was created in Create a Log File:

1. Click the button to the left of the **inject** (timestamp) node to inject the timestamp, which inserts the timestamp into the file.
2. Go to the steps in Read a File to verify that the write was successful.
3. Continue to click the **inject** button to write additional timestamps to the file.

Write a Message to the File

To illustrate yet another way to populate a file from a flow, try these steps to inject a string into the flow:

1. In your write flow, double-click the **inject** mode to change the payload to **string**, enter text, and click **Done**. In this example, we have entered: `Hello World`.



Edit inject node

Payload: Hello World

Topic:

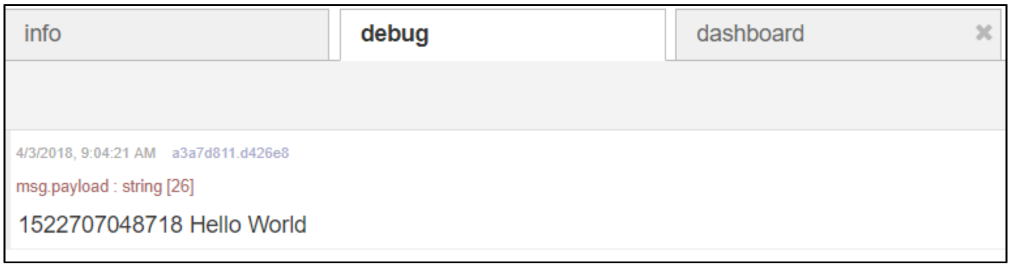
Repeat: none

□ Inject once at start?

Name: Name

Note: "interval between times" and "at a specific time" will use cron. See info box for details.

2. Click **Save** at the top right of the canvas.
3. Click the button to the left of the **inject** node to inject the message (payload) into the flow.
4. Go to Read a File for the instructions to verify that the message was written to the file. In this example, the debug area will display the timestamp from the previous steps and the inserted message.

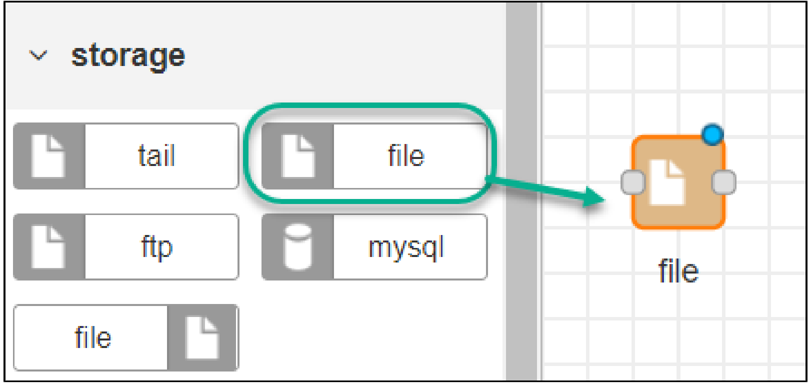


```
4/3/2018, 9:04:21 AM   a3a7d811.d426e8
msg.payload : string [26]
1522707048718 Hello World
```

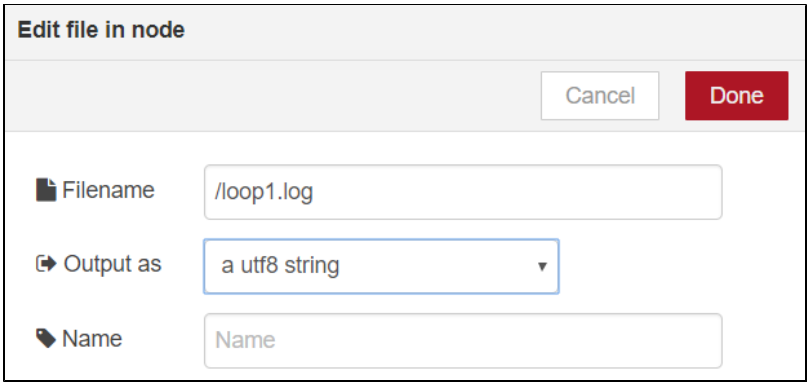
Read a File

To read the file that was created in Create a Log File and written to in Write to a File, take these steps:

1. Drag the **file in** node onto the canvas.



2. Double-click the node, enter the filename, and click **Done**.
   The output will be in the standard utf8 string format.
   The forward slash is required in the filename. For example: `/loop1.log`

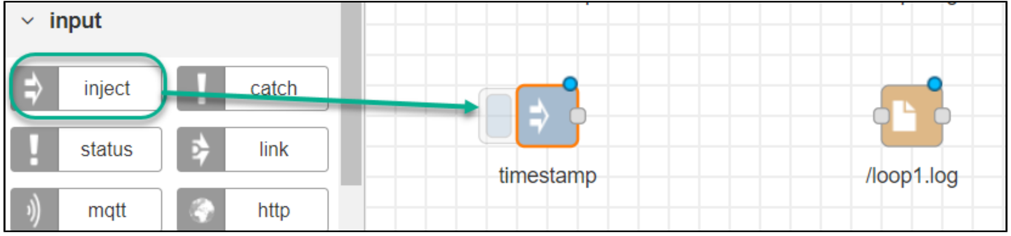


Filename /loop1.log

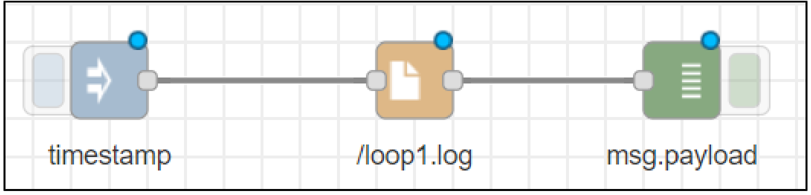
Output as a utf8 string

Name Name

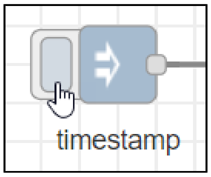
3. Drag an **inject** node onto the canvas, to the left of the **file in** node.



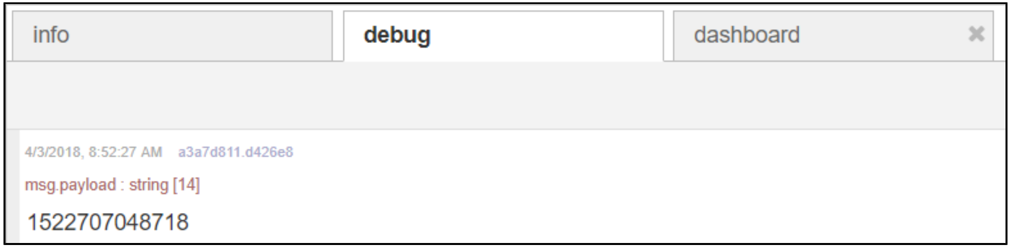
4. Drag a **debug** node onto the canvas, to the right of the **file in** node.
5. Connect the **inject** node and the **debug** node to the **file in** node.



6. Click **Save** at the top right of the canvas.
7. At the bottom of the canvas, click the **debug** tab.
8. Click the button to the left of the **inject** (timestamp) node to *trigger* the flow.



The debug area shows the timestamp (payload) that has been written to the `/loop1.log` file.



```
4/3/2018, 8:52:27 AM   a3a7d811.d426e8
msg.payload : string [14]
1522707048718
```

Flow to Write to Tags

Under certain circumstances, it may be necessary to control a device by updating values in registers (tags), for example, turning a switch on or off.

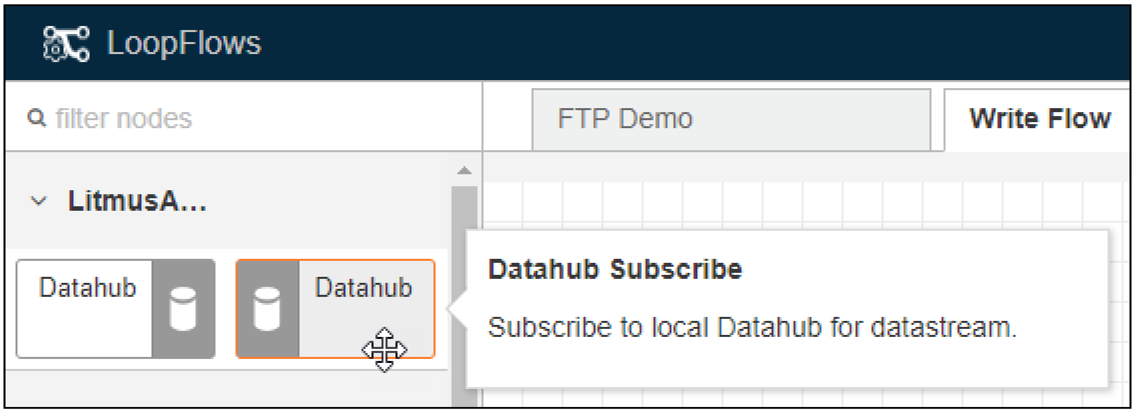
- Some device registers are read-only. For example, to write to a tag, select a Holding register tag, *not* an Input register tag. See DeviceHub Add a Tag.
- PLCs have a dedicated set of read-only registers, as well as a set of registers that can be updated. Refer to the device vendor's documentation for register-specific details.

The following steps provide an example of how to create a basic Voice of the Machine Edge Flow to write to a device register/tag.

1. Subscribe to DeviceHub
2. Publish to DataHub
3. View the Flow Results
4. Troubleshoot the Write to Tag Flow
5. View a Video of a Flow that Writes to a Tag

Subscribe to DeviceHub

1. In the Voice of the Machine Edge navigation panel, click **Flows**.
2. Click the **+** icon at the top right of the window to create a new flow with a blank canvas.
3. Drag the **DataHub Subscribe** node onto the canvas.

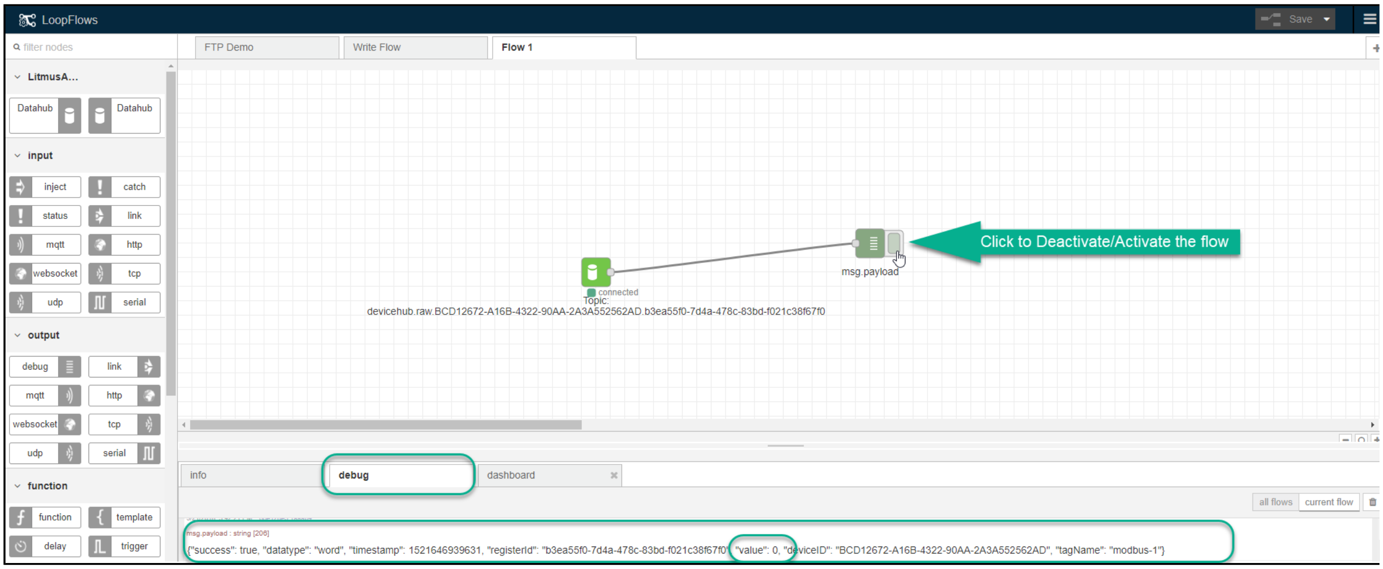


Datahub Subscribe

Subscribe to local Datahub for datastream.

4. Double-click the node to edit it. Use the procedure in Create a Flow for a Voice of the Machine Edge Device to:
   a. Update DataHub Connect.
   b. Copy a topic from a DeviceHub tag.
   c. Paste the topic into the DataHub Subscribe node.
5. Click **Done** to save the node.
6. Drag the **debug** node onto the canvas.
7. Connect the two nodes.
8. Click **Save** to save and activate the flow.

At this point, the DataHub node should be sending data to the Debug node. Go to the **debug** tab below the canvas to view the results.
Note: If you are not seeing messages in the debug area, check that the tag's polling interval is greater than zero.

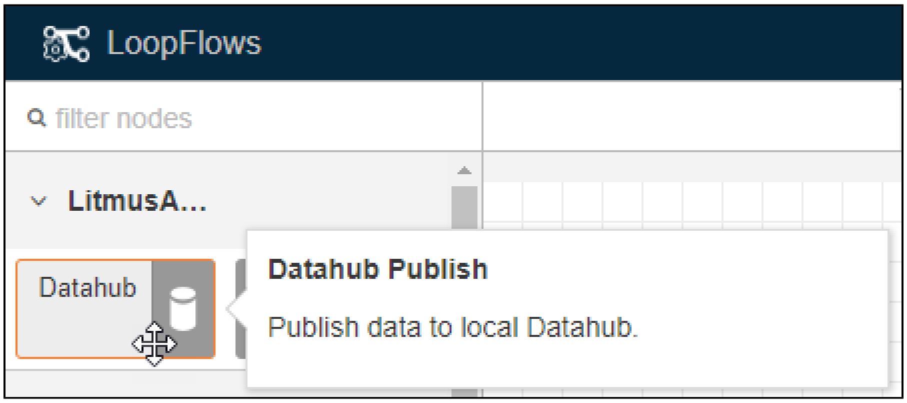


Topic:
devicehub.raw.BCD12672-A16B-4322-90AA-2A3A552562AD.b3ea55f0-7d4a-478c-83bd-f021c38f67f0

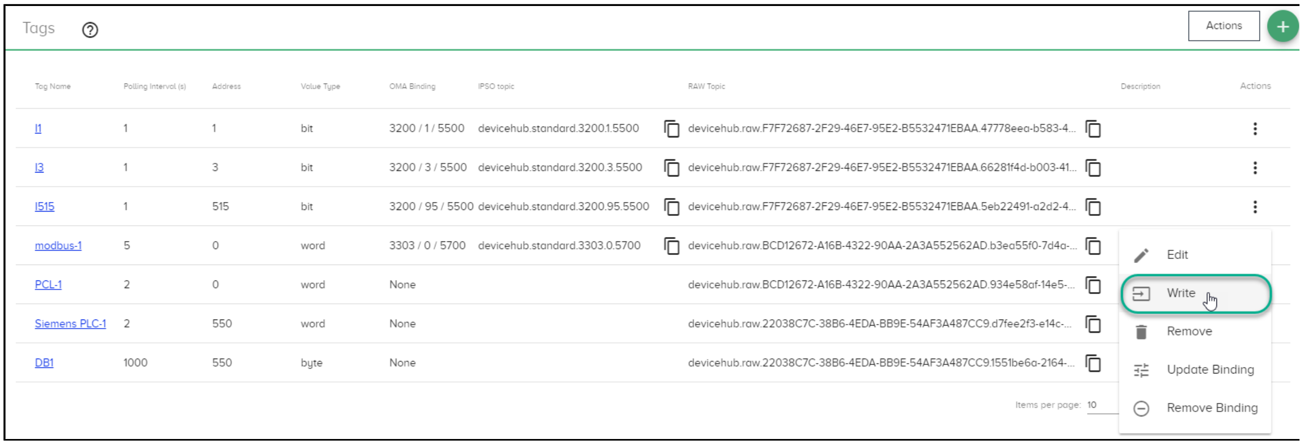
```
msg.payload : string [206]
{"success": true, "datatype": "word", "timestamp": 1521646939631, "registerId": "b3ea55f0-7d4a-478c-83bd-f021c38f67f0", "value": 0, "deviceID": "BCD12672-A16B-4322-90AA-2A3A552562AD", "tagName": "modbus-1"}
```

Publish to DataHub

1. Drag the **DataHub Publish** node onto the canvas.



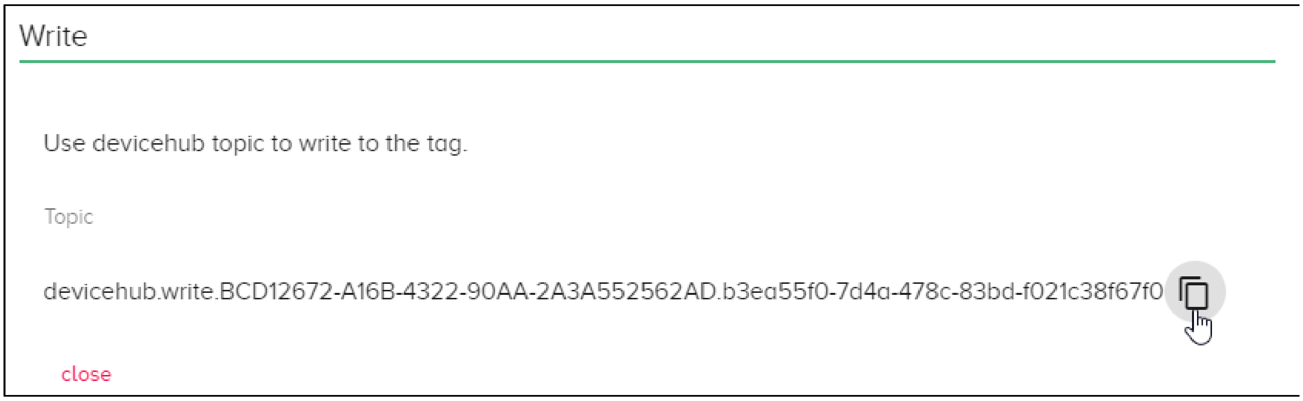
2. Go to **DeviceHub > Tags** and locate the same tag used in Subscribe to DataHub.
3. In the Actions list for the tag, select **Write**.



Tag Name	Polling Interval (s)	Address	Value Type	OMA Binding	IPSO topic	RAW Topic	Description	Actions
I1	1	1	bit	3200 / 1 / 5500	devicehub.standard.3200.1.5500	devicehub.raw.F7F72687-2F29-46E7-95E2-B5532471EBAA.47778eea-b583-4...		⋮
I3	1	3	bit	3200 / 3 / 5500	devicehub.standard.3200.3.5500	devicehub.raw.F7F72687-2F29-46E7-95E2-B5532471EBAA.66281f4d-b003-41...		⋮
I515	1	515	bit	3200 / 95 / 5500	devicehub.standard.3200.95.5500	devicehub.raw.F7F72687-2F29-46E7-95E2-B5532471EBAA.5eb22491-a2d2-4...		⋮
modbus-1	5	0	word	3303 / 0 / 5700	devicehub.standard.3303.0.5700	devicehub.raw.BCD12672-A16B-4322-90AA-2A3A552562AD.b3ea55f0-7d4a-...		
PCL-1	2	0	word	None		devicehub.raw.BCD12672-A16B-4322-90AA-2A3A552562AD.934e58af-14e5-...		
Siemens PLC-1	2	550	word	None		devicehub.raw.22038C7C-38B6-4EDA-BB9E-54AF3A487CC9.d7fee2f3-e14c-...		
DB1	1000	550	byte	None		devicehub.raw.22038C7C-38B6-4EDA-BB9E-54AF3A487CC9.1551be6a-2164-...		

Items per page: 10

4. In the Write pop-up, click the ⧉ icon to copy the *write topic*.



Write

Use devicehub topic to write to the tag.

Topic

devicehub.write.BCD12672-A16B-4322-90AA-2A3A552562AD.b3ea55f0-7d4a-478c-83bd-f021c38f67f0

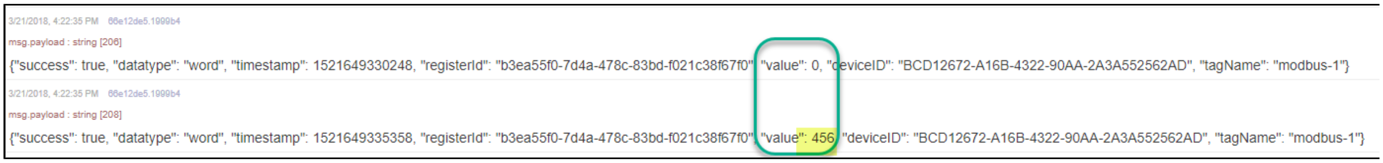
5. Back in the canvas, double-click the **DataHub Publish** node, paste the write topic into the node, and click **Done**.
6. Drag an **inject** (input) node onto the canvas.
7. Connect the Inject and DataHub Publish nodes.
8. Double-click the **inject** node and select **String** for the Payload.
9. Enter the text into the Payload field, following the format shown below, and click **Done**.

```
{"value":456}
```

Note: Any string value can be entered, as long as you adhere to the JSON format (shown above) for the payload text.

10. Click **Save** to save and activate the flow.

At this point, the value should be changing. Go to the **debug** tab below the canvas to view the results.

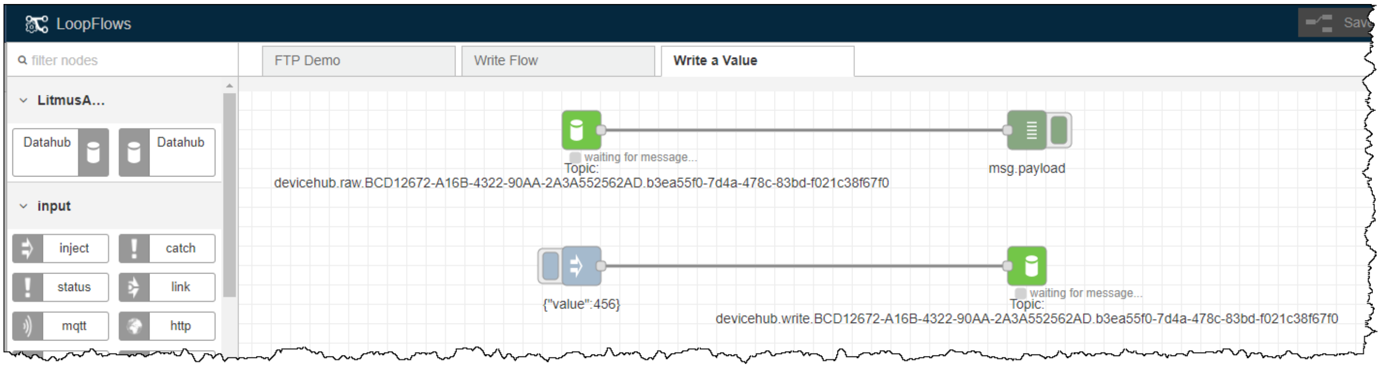


```
3/21/2018, 4:22:35 PM   66e12de5.1999b4
msg.payload : string [206]
{"success": true, "datatype": "word", "timestamp": 1521649330248, "registerId": "b3ea55f0-7d4a-478c-83bd-f021c38f67f0", "value": 0, "deviceID": "BCD12672-A16B-4322-90AA-2A3A552562AD", "tagName": "modbus-1"}
3/21/2018, 4:22:35 PM   66e12de5.1999b4
msg.payload : string [208]
{"success": true, "datatype": "word", "timestamp": 1521649335358, "registerId": "b3ea55f0-7d4a-478c-83bd-f021c38f67f0", "value": 456, "deviceID": "BCD12672-A16B-4322-90AA-2A3A552562AD", "tagName": "modbus-1"}
```

View the Flow Results

This flow now contains two connections, as shown in the following illustration.

- The first connection shows the *subscribe* action.
- The second connection shows the *publish* to the device tag.



Troubleshoot the Write to Tag Flow

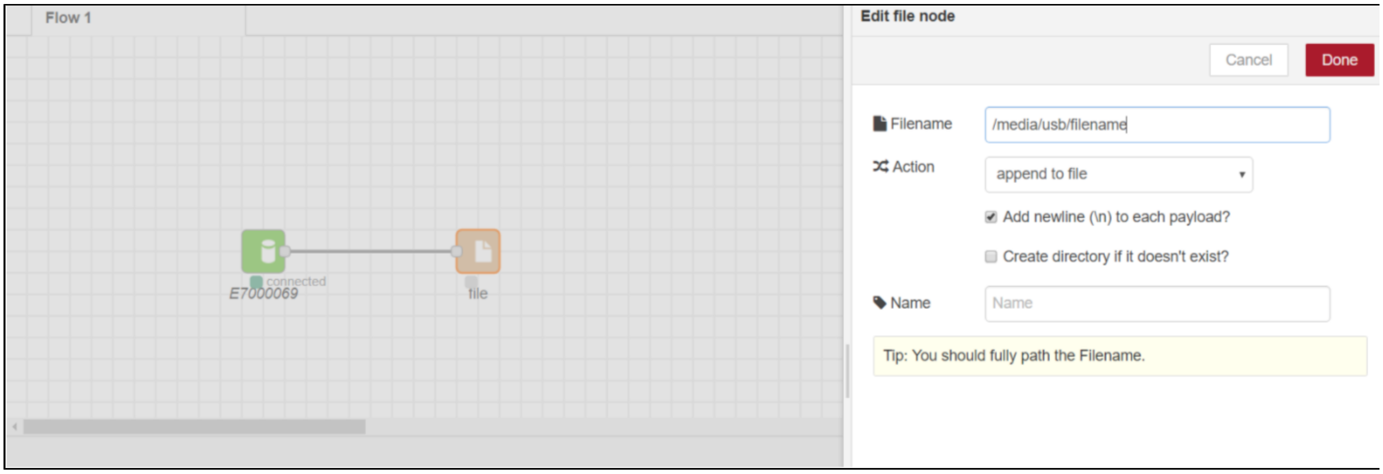
If you do not see the value changing, check the following:

- **Have you saved recent flow changes?** Click **Save** above the canvas.
- **Is the device connected?** Go to **DeviceHub > Devices** to check.
- **Have you chosen a valid register that allows writes?** Make sure you are not trying to write to an Input register. Go to **DeviceHub > Tags**, edit the tag, and verify the register name.
- **Is the Debug node activated?** Toggle the switch on the Debug node.

Flow to Save Data to a USB Device

Flows can be used for various storage operations. A common use case involves saving data to a flash drive connected to the Voice of the Machine Edge device. For this scenario, select the *file node* and define the path to the USB drive. **Example**: `/media/usb/filename` where the filename is in the format, `/<pathname>/<filename>.<ext>`

In the following example, data from a local topic defined in DataHub is being saved to a USB drive using a file node.



Edit file node

Filename: /media/usb/filename

Action: append to file

☑ Add newline (\n) to each payload?

□ Create directory if it doesn't exist?

Name: Name

Tip: You should fully path the Filename.

Flow to Subscribe to All Topics

Use this simple flow to test DataHub Subscribe nodes. This example uses an MQTT wildcard (**>**) to subscribe to all topics.

Flow to Poll a Tag Once for an Event-Driven Poll

Often there is no need to regularly poll a register. Instead, a common use case requires data to be collected only when an event, such as a change of state (on/off), triggers a poll.

Example Use Case 1: It likely is not necessary to pay attention to a temperature sensor if the device's motor is *not* running. In this example, the temperature sensor only needs to be monitored if there's a significant change or if another condition warrants monitoring. In cases such as this, a *poll once* flow should be configured.

Example Use Case 2: To minimize cloud IoT storage fees and bandwidth costs, monitor critical processes and capture the data *every second*. Then, publish collected values only *once per minute*. If no problems arise, there is no need to send data every second. However, if an issue occurs, then more granular data is required for analysis, so this event will trigger the *poll once* poll to publish all the once-per-second data collected for the last minute.

Example Use Case 3: A motor for a pump only needs to be polled when the power switch is turned on. In this case, you would want to continuously poll the switch and when its state changes, execute a *poll once* trigger to poll the motor control.

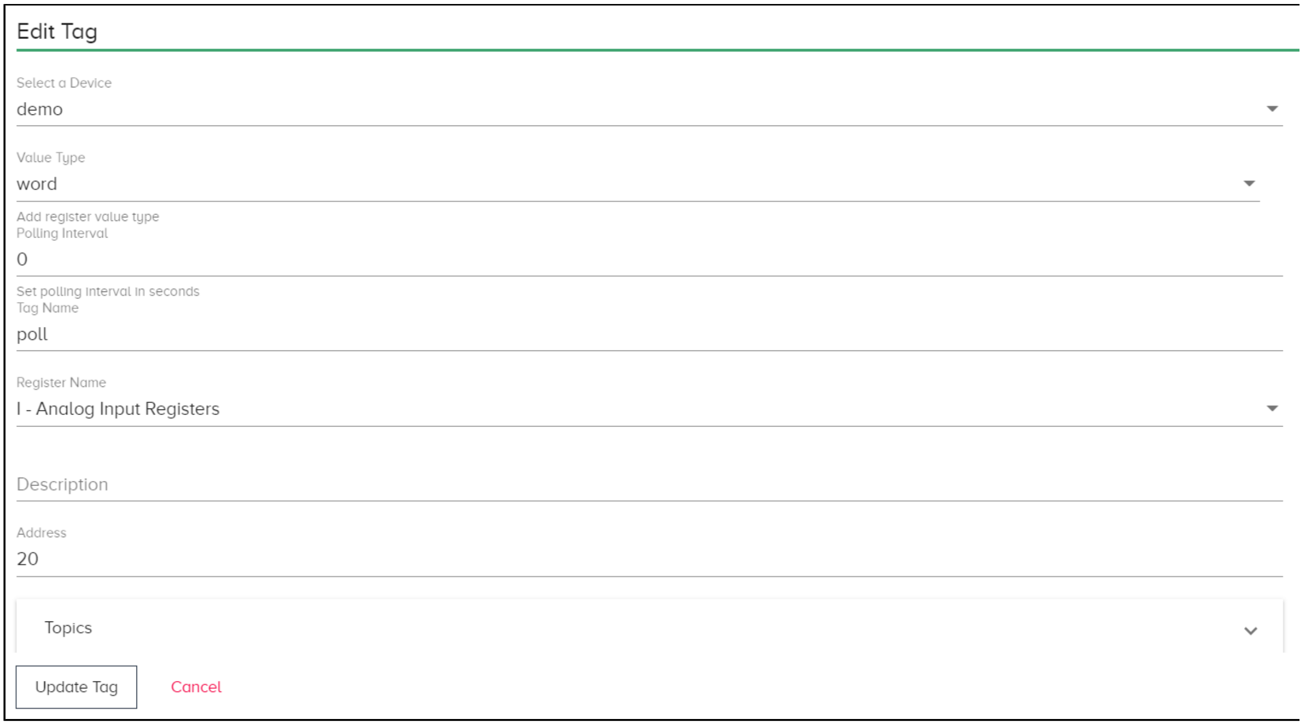
The *poll once* device topic for a DeviceHub tag is designed to be used to poll a device register only when triggered by an event. Take the following steps to configure a device register to poll once:

1. Create Device Tags
2. Create a Poll-Once Flow

Create Device Tags

Before configuring a flow, device tags must be created to specify registers that will be polled.

1. In the Voice of the Machine Edge navigation pane, select **DeviceHub > Tags**.
2. Click the ⊕ icon at the top right of the Tags window.
3. Configure a tag, similar to the following example.
   Poll once tags must have the polling interval set to zero.



Edit Tag

Select a Device
demo

Value Type
word
Add register value type

Polling Interval
0
Set polling interval in seconds

Tag Name
poll

Register Name
I - Analog Input Registers

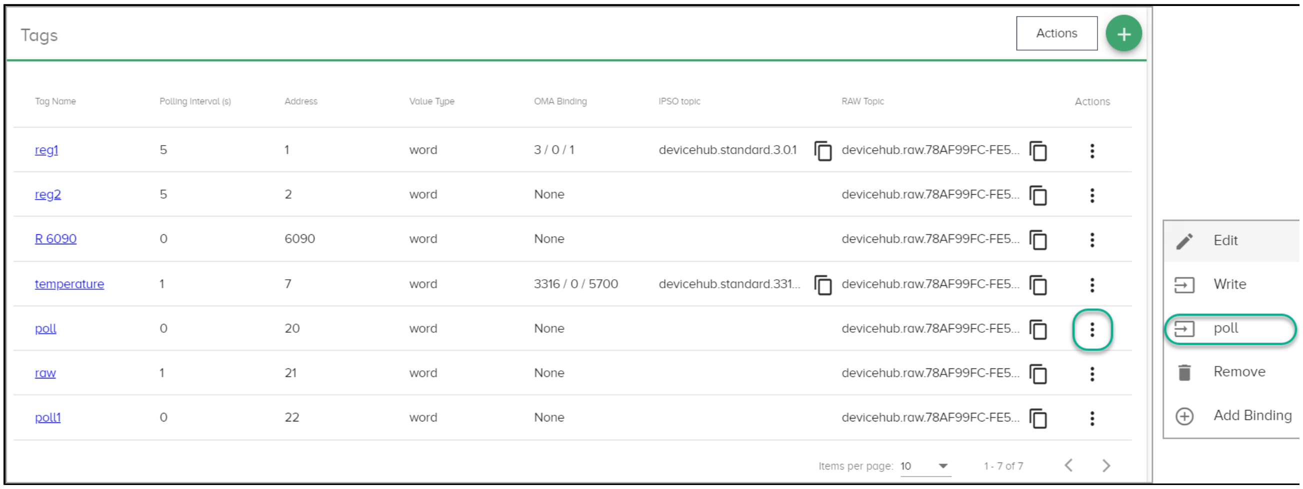
Description

Address
20

Topics

Update Tag   Cancel

4. For this example, three tags are created:
    a. One of these tags will have a polling interval set to 1 because this will be used to continuously poll a device topic, waiting for a state change.
    b. The other tags will be used as *poll once* triggers that respond to events (state changes).
5. In the **DeviceHub > Tags** list, find the tag with the polling interval of zero and select **poll** from the Actions list.

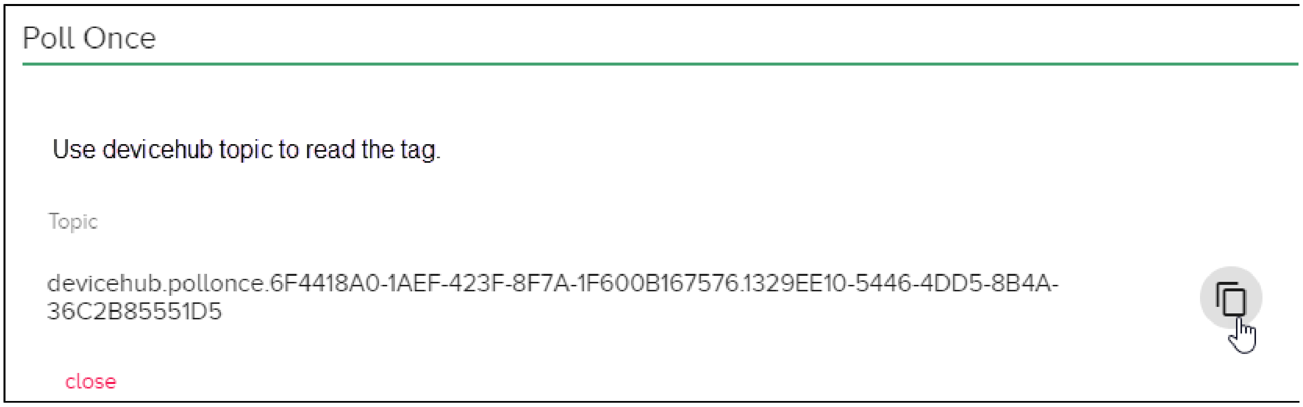


Tags   Actions   +

Tag Name	Polling Interval (s)	Address	Value Type	OMA Binding	IPSO topic	RAW Topic	Actions
reg1	5	1	word	3 / 0 / 1	devicehub.standard.3.0.1	devicehub.raw.78AF99FC-FE5...	⋮
reg2	5	2	word	None		devicehub.raw.78AF99FC-FE5...	⋮
R 6090	0	6090	word	None		devicehub.raw.78AF99FC-FE5...	⋮
temperature	1	7	word	3316 / 0 / 5700	devicehub.standard.331...	devicehub.raw.78AF99FC-FE5...	⋮
poll	0	20	word	None		devicehub.raw.78AF99FC-FE5...	⋮
raw	1	21	word	None		devicehub.raw.78AF99FC-FE5...	⋮
poll1	0	22	word	None		devicehub.raw.78AF99FC-FE5...	⋮

Items per page: 10   1 - 7 of 7

6. In the Poll Once dialog, copy the **Topic**. When you create a flow, you will paste this topic into the flow's DataHub Publish node.



Poll Once

Use devicehub topic to read the tag.

Topic

devicehub.pollonce.6F4418A0-1AEF-423F-8F7A-1F600B167576.1329EE10-5446-4DD5-8B4A-36C2B85551D5

close

Create a Poll-Once Flow

Several flows can be created to illustrate the logic for a *poll once* scenario. In this example, create a flow that polls the device topic once based on the following logic:

1. Continuously poll a register (for example, a switch for a pump's motor), waiting for a state change.
2. When the state changes (for example, the motor is turned on), trigger a “poll once” that subscribes to the message (for example, the temperature of the motor).
3. Verify that the subscribe nodes are receiving the expected values.

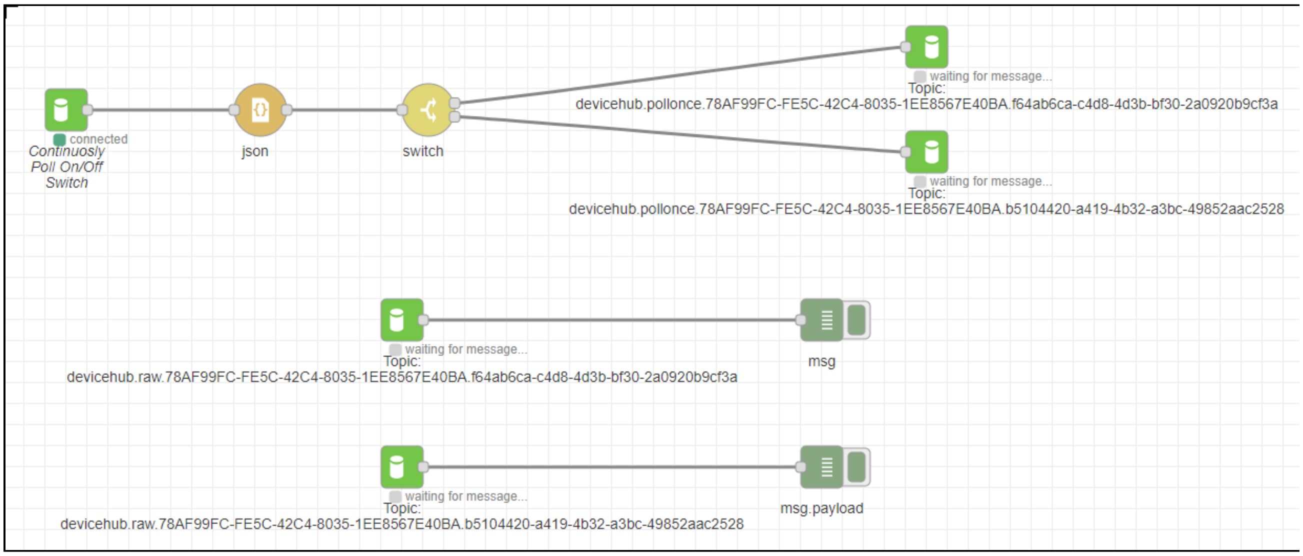
To create a new flow:

1. In the navigation pane, select **Flows**.
2. Click **Go to Flows**.
3. Drag the following nodes onto the canvas:
   a. **DataHub Subscribe**: Paste the **raw** topic for the device here to subscribe to the local DataHub data stream.
   b. **json**: Use this node to parse the `msg.payload` to convert a JSON string to/from a javascript object.
   c. **DataHub Publish**: Send message payload to DataHub. <u>Paste the poll once device topic into this node</u>. The poll once topic has the following format: `devicehub.pollonce.`**{deviceID}.{registerID}**.
   d. **switch**: Route messages based on the value detected. Use a switch for case logic such as:

```
If tag1 > 25, then start polling tag2
or
If tag1 > 25, then start polling tag1 every 0.1 seconds
```

Example Flow

In the following example, two DataHub Publish nodes (configured with the poll once topic) receive the event message that triggers the device poll.



Topic: devicehub.pollonce.78AF99FC-FE5C-42C4-8035-1EE8567E40BA.f64ab6ca-c4d8-4d3b-bf30-2a0920b9cf3a

Topic: devicehub.pollonce.78AF99FC-FE5C-42C4-8035-1EE8567E40BA.b5104420-a419-4b32-a3bc-49852aac2528

Topic: devicehub.raw.78AF99FC-FE5C-42C4-8035-1EE8567E40BA.f64ab6ca-c4d8-4d3b-bf30-2a0920b9cf3a

Topic: devicehub.raw.78AF99FC-FE5C-42C4-8035-1EE8567E40BA.b5104420-a419-4b32-a3bc-49852aac2528

Flow to Save Multiple Register Values

Use Case for Saving Register Values from Multiple PLCs

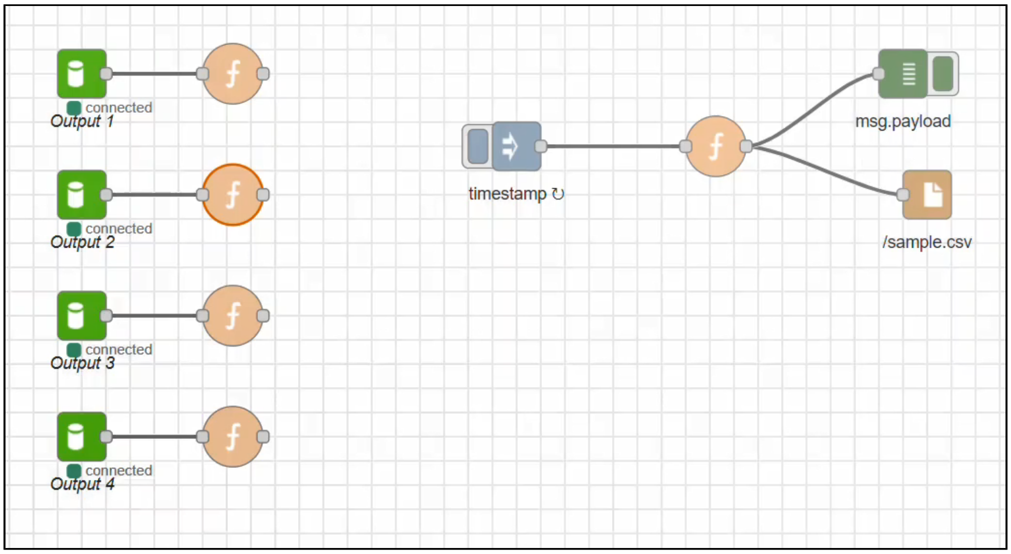
- Collect raw data from various PLC registers.
- Aggregate the data at regular intervals.
- Save the aggregated data for use by other applications.

Note: Even though data is collected at regular intervals, collection from each register does not happen simultaneously. In order to aggregate multiple values, separate processing blocks are needed to save values to global variables, which then can be accessed by all flows, functions, and nodes.

Example Flow for Saving Multiple Values

Goal: Collect data from different types of inputs and use multiple nodes in a flow to save values in a database or simply save values to a CSV file.

The following example saves register values to a comma-separated-values file. A similar method could be used to save values in a MySQL database.

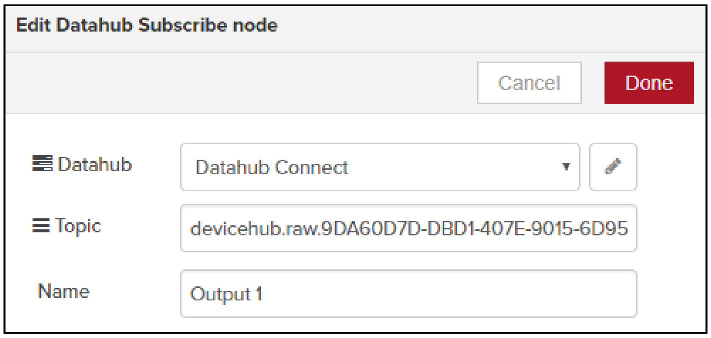


Steps to Create a Flow to Save Values from Multiple Inputs

The procedure shown here is a high-level description of the steps that need to be taken. Use these steps as a basis for configuring a flow specific to your requirements.

To save values from multiple registers to a CSV file:

1. Interrogate a PLC register for a value.
   a. Use a **Datahub Subscribe** node.
   b. Configure the node with a DeviceHub raw topic.
   c. This will be Output 1. The collected value will be the first global value to be saved, as described in Step 2.



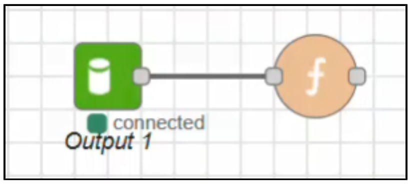
Edit Datahub Subscribe node

Datahub: Datahub Connect

Topic: devicehub.raw.9DA60D7D-DBD1-407E-9015-6D95

Name: Output 1

2. Wire a **Function** node to the **Datahub Subscribe** node.



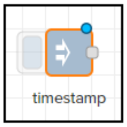
3. Edit the **Function** node and add the following lines to save the value to a Global Variable (output1), which can be accessed by other flows, nodes, and functions:

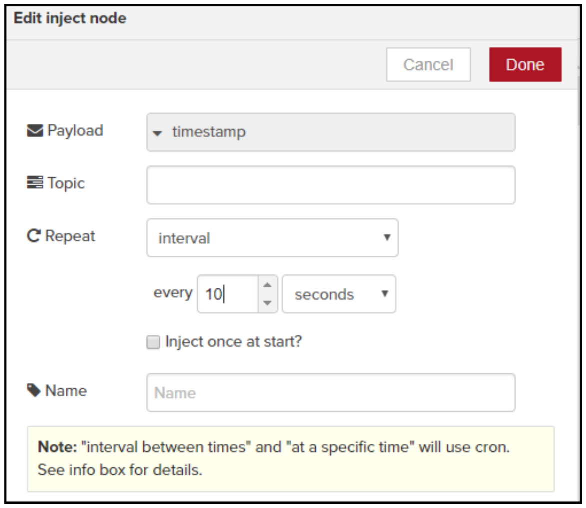
```
var temp = (JSON.parse(msg.payload)).value;
global.set("output1",temp);
return msg;
```

4. Repeat steps 1 through 3 for other PLC registers, resulting in **Datahub Subscribe** nodes: Output 2 through Output 4 (for this example). Note that in some cases, you may have a need to set a value to Null, as shown in the following example.

```
var temp = (JSON.parse(msg.payload)).value;
global.set("output4",null);
return msg;
```

5. Use an **Inject** node to inject a timestamp every 10 seconds to read the global variable values.





Edit inject node

Payload: timestamp

Topic:

Repeat: interval

every 10 seconds

□ Inject once at start?

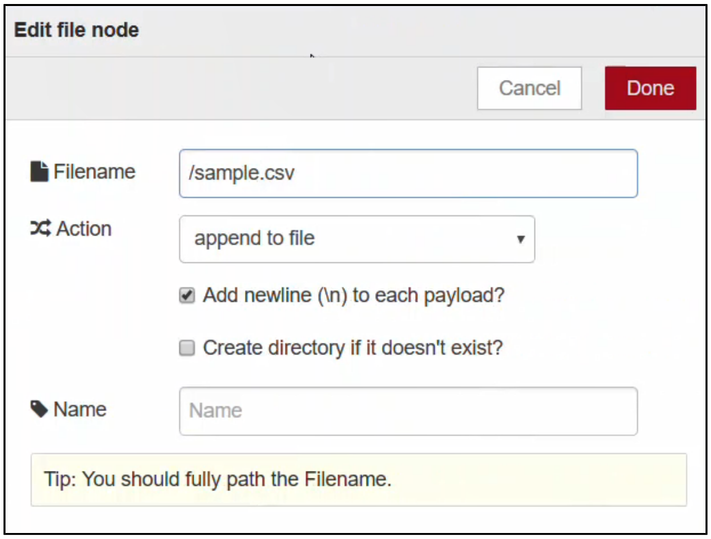
Name: Name

Note: "interval between times" and "at a specific time" will use cron. See info box for details.

6. Wire a **Function** node to the **Inject** node.
7. Edit the **Function** node to set up the format for the CSV file.

```
//create CSV
var temp = global.get("output1") +"," + global.get("output2") +","+global.get("output3") +","+global.
get("output4");
msg.payload = temp;
return msg;
```

8. Wire a **File** node to the **Function** node and edit it, as shown below.
Note: Be sure to **Save** the flow.


Edit file node

Filename: /sample.csv

Action: append to file

☑ Add newline (\n) to each payload?

□ Create directory if it doesn't exist?

Name: Name

Tip: You should fully path the Filename.

Configure the FTP Server Service

Once the CSV file has been populated, export the file via the Voice of the Machine Edge FTP Server Service. See FTP Server.

Flow to Use Voice of the Machine Edge as an FTP Client

he steps provided on this page illustrate how to access an FTP server, where Voice of the Machine Edge is the client.

This example lists the steps to configure a flow to get a file and then read the contents of the file:

1. Create a Flow to Transfer a File to Voice of the Machine Edge
2. Create a Flow to Read File Contents
3. Trigger the Flows
4. View a Video of an FTP Get Flow
5. View a Video of a File Read Flow
6. Troubleshoot Flows

Tips:

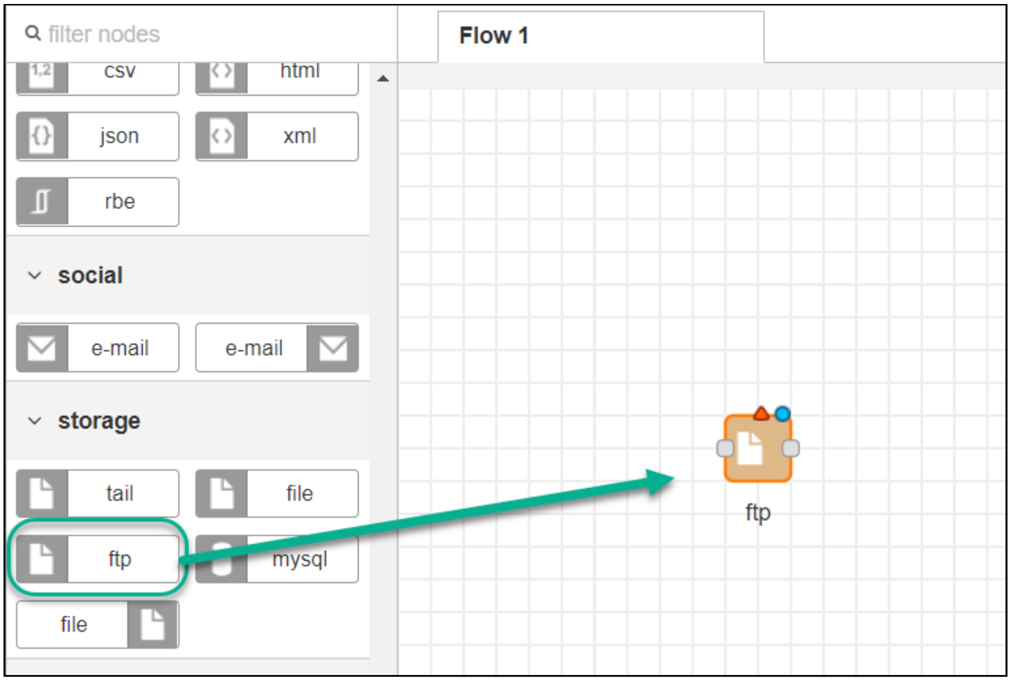
- Use the **filter nodes** function in the node palette to find specific nodes, such as a **file** node.
- As you modify flow configurations, be sure to save the flows. Otherwise, the nodes will exist only in the flow editor and the flows will not be executable.
- Filenames used in node configurations are case sensitive.

Create a Flow to Transfer a File to Voice of the Machine Edge

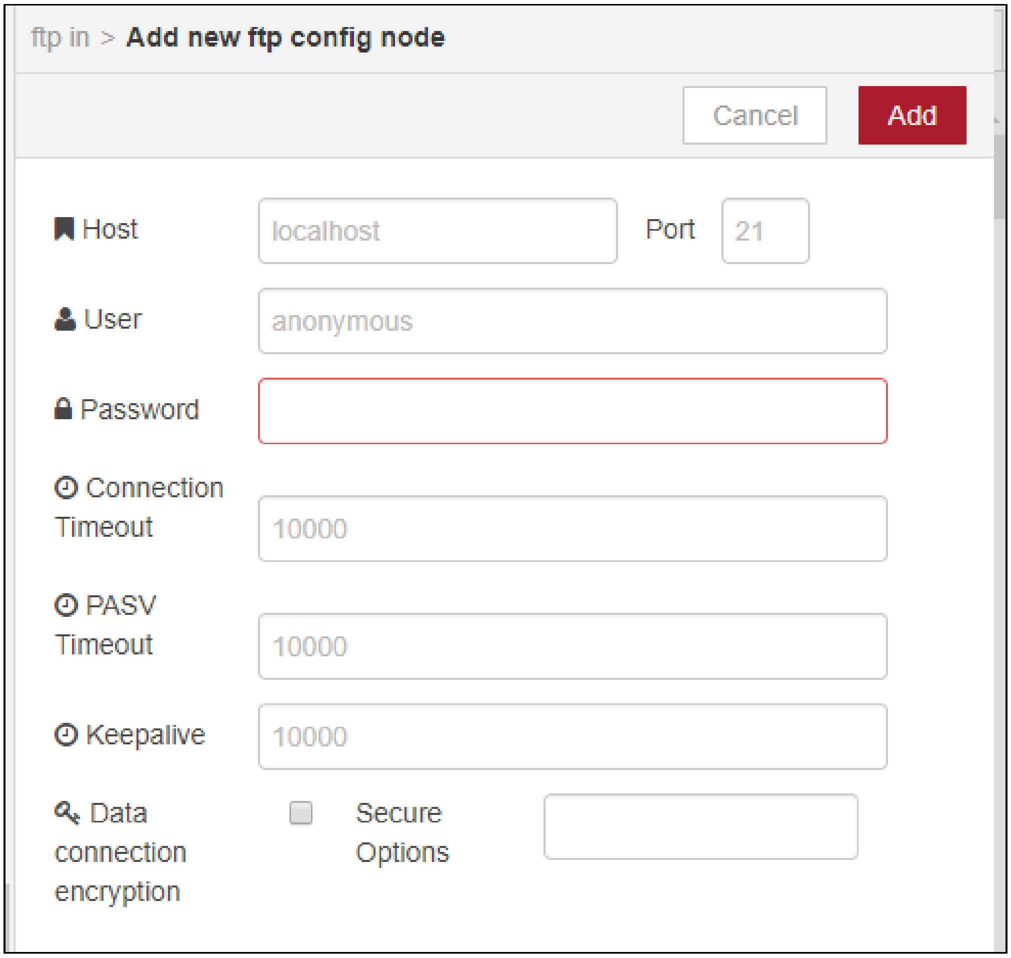
When files need to be transferred to the Voice of the Machine Edge device, do *not* use a USB drive because the file format and write permissions may cause issues. Instead, use FTP, a more secure and reliable method.

Use these steps to create an *FTP Get* flow to transfer a file to the Voice of the Machine Edge device:
In these steps, Voice of the Machine Edge is the ftp client. If you want to configure Voice of the Machine Edge to be an ftp server, see the instructions in FTP Server.

1. On a system with an ftp server, create a file that contains some text. For this example, we are using `HelloWorld.txt`, which contains the text: `Hello World! Let's connect`.
2. On the client, in Voice of the Machine Edge Flows, click **Go to Flows**.
3. Drag the **ftp** node (under **storage** in the palette) onto the canvas.
   When you use the ftp node, Voice of the Machine Edge is always the client. Use an ftp node to interact with an ftp server for the standard ftp client actions: List, Get, Put, Delete.



4. Double-click the **ftp** node to launch the editor where you will configure the FTP server.
5. In the edit dialog, click the pencil ✎ to add a new FTP server.
6. Enter the configuration parameters and click **Add**.
   Enter the FTP server details to establish a connection with Voice of the Machine Edge.
   - **Host**: IP address of the FTP server.
   - **Username**: The FTP username.
   - **Password**: The password for the FTP user.
   - **Port**: If it is a Voice of the Machine Edge FTP server, use port 2121.



ftp in > **Add new ftp config node**

Host: localhost Port: 21

User: anonymous

Password:

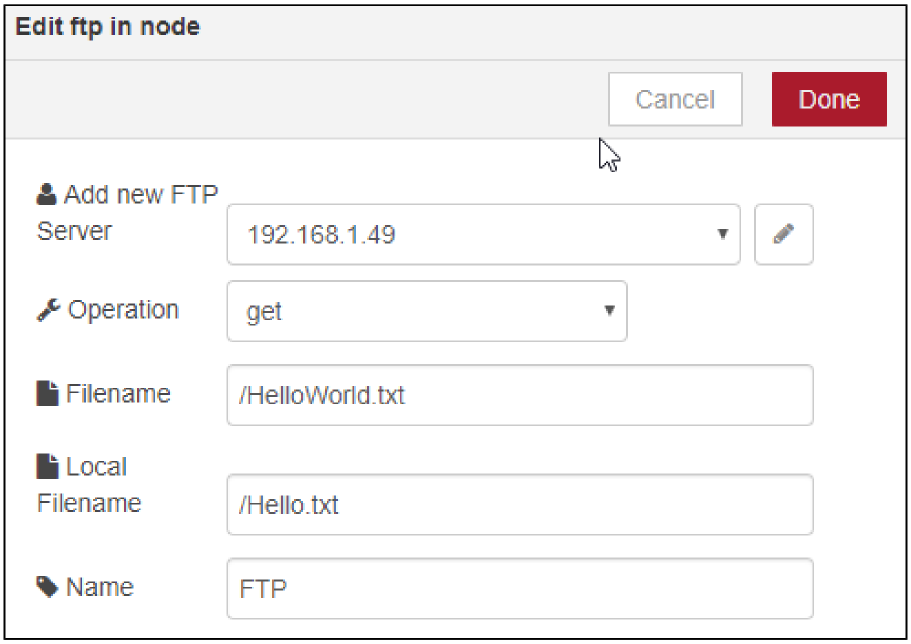
Connection Timeout: 10000

PASV Timeout: 10000

Keepalive: 10000

Data connection encryption: □ Secure Options:

7. Back in the **Edit ftp in node** dialog, configure the following and then click **Done**.
   a. **Operation**: get
   b. **Filename**: The name of the source file on the remote machine that you are transferring/getting. For this example, enter: `/Hello World.txt`
   c. **Local Filename**: The destination filename on the Voice of the Machine Edge client. For this example, enter: `/Hello.txt`
      Note: Be sure to include the forward slash (/) in the filename.
   d. **Name**: (optional) Enter a name for the FTP node in the flow.



Edit ftp in node

Add new FTP Server: 192.168.1.49

Operation: get

Filename: /HelloWorld.txt

Local Filename: /Hello.txt

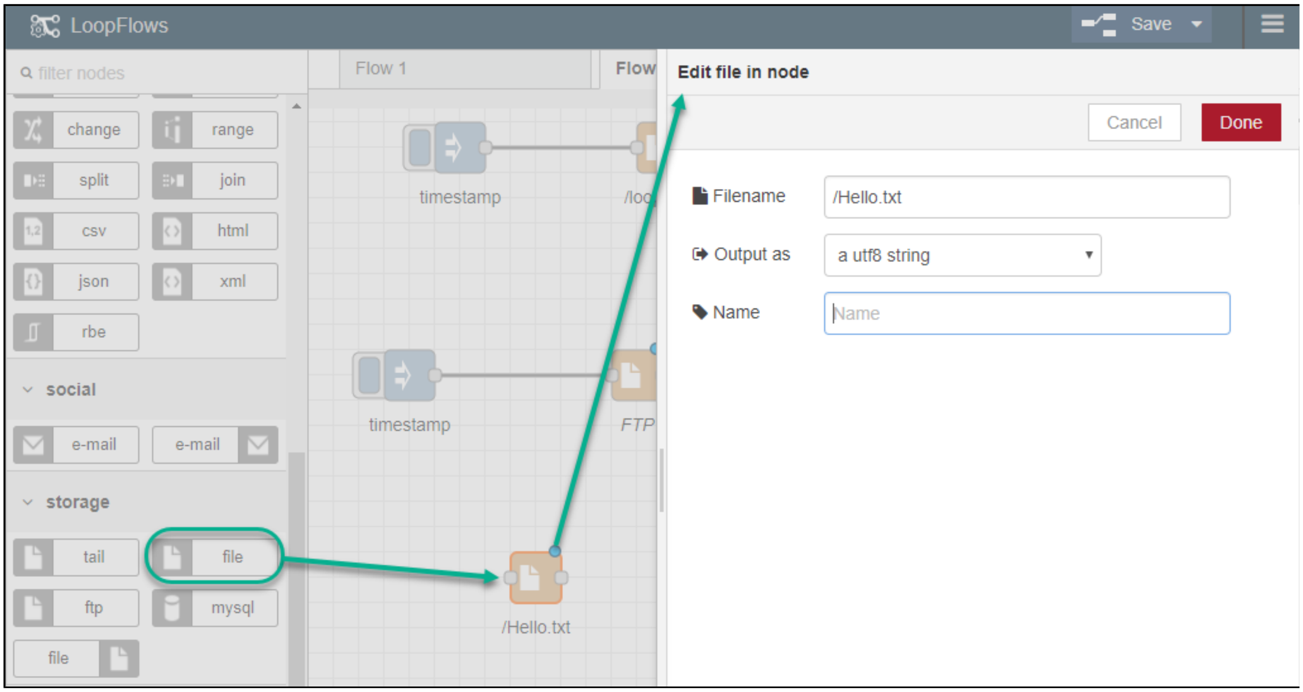
Name: FTP

8. Drag a **debug** node onto the canvas, to the right of the **ftp in** node, and connect it to the **ftp in** node.
9. Drag an **inject** node onto the canvas, to the left of the **ftp in** node, and connect it to the **ftp in** node.
10. Click **Save** to save the flow you just created.
11. Go to Trigger the Flows to trigger the *FTP Get* operation. The status can be viewed in the debug tab, beneath the canvas.

Create a Flow to Read File Contents

In this scenario, the **file in** node is used to read the contents of the file that was just transferred to the device with the FTP flow.

1. Drag the **file in** node (under **storage** in the palette) onto the canvas and modify its parameters.
   a. Enter the filename. For the purpose of this example, enter the file that was just transferred to the Voice of the Machine Edge client: `/Hello.txt`
   b. Output as **a utf8 string**.



Edit file in node

Cancel Done

Filename: /Hello.txt

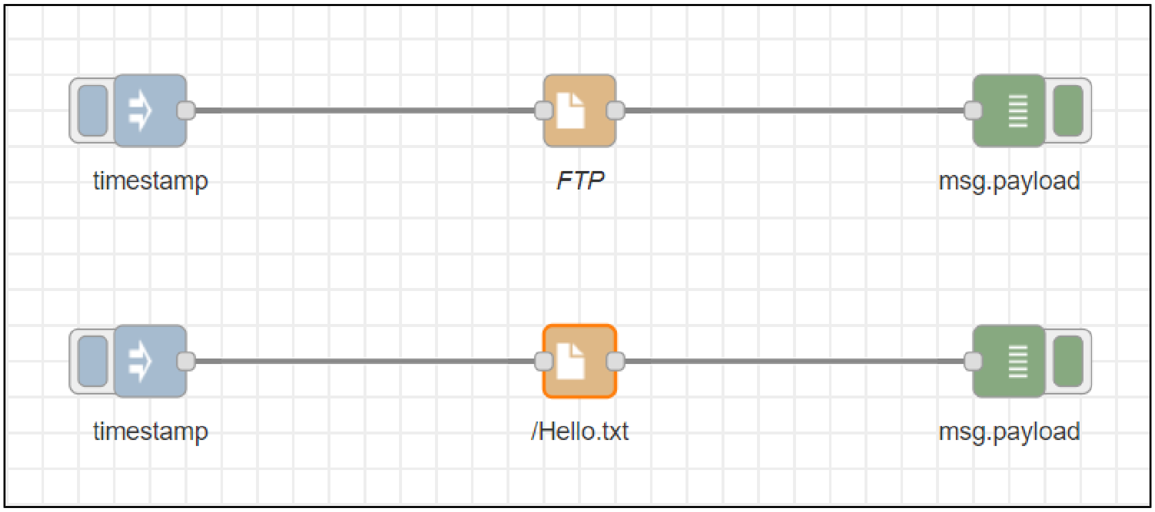
Output as: a utf8 string

Name: Name

2. Drag an **inject** node onto the canvas, to the left of the **file in** node, and connect it to the **file in** node. This serves as the input node in the flow.
3. Drag a **debug** node onto the canvas, to the right of the **file in** node, and connect it to the **file in** node. This serves as the output node.
4. Click **Save** to save the flow you just created.
5. Go to Trigger the Flows to trigger the *File Read* operation. The status can be viewed in the debug tab, beneath the canvas.

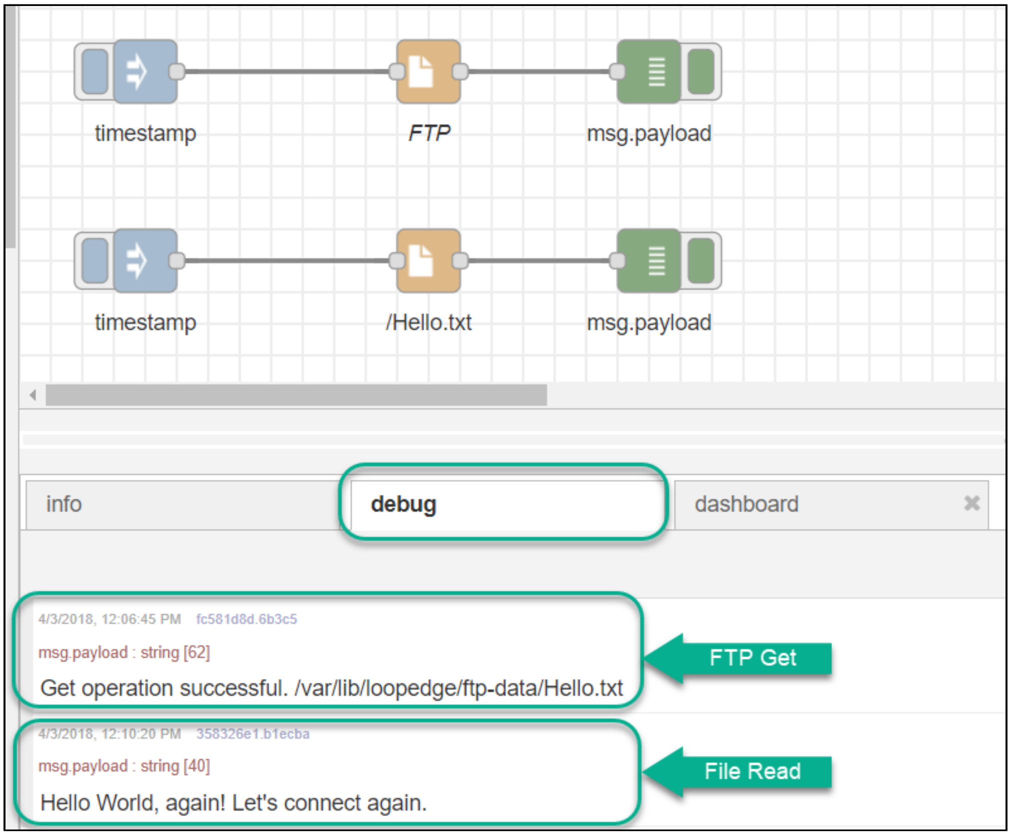
Trigger the Flows

The flows you created in the previous procedures should look something like this:



To inject/activate the flows:

1. First, click the button on the **inject** (timestamp) node of the **ftp** flow.
   The timestamp simply serves as a trigger for an event. In this case, it is doing the *ftp get*.
2. Then, click the button on the **inject** (timestamp) node of the **file in** flow, which reads the contents of the file.
3. Beneath the canvas, select the **debug** tab to view the results of the triggered flows.



```
4/3/2018, 12:06:45 PM   fc581d8d.6b3c5
msg.payload : string [62]
Get operation successful. /var/lib/loopedge/ftp-data/Hello.txt

4/3/2018, 12:10:20 PM   358326e1.b1ecba
msg.payload : string [40]
Hello World, again! Let's connect again.
```

Troubleshoot Flows

If you do not see the expected results, check the following:

- **Have you saved recent flow changes?** Click **Save** above the canvas at the right.
- **Is the Debug node activated?** Toggle the switch on the Debug node.

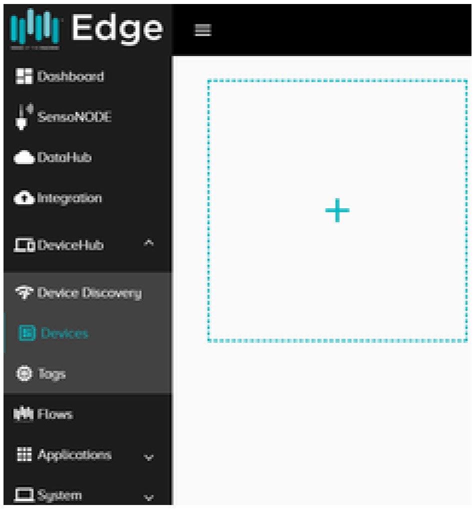
What is DeviceHub

DeviceHub provides the mechanism for connecting to physical devices (PLCs, sensors, etc.) to collect data from them.

- DeviceHub needs to be configured for *southbound* connections, including connections to PLCs.
- DeviceHub collects data from physical devices and publishes it to an internal message broker. The data can be sent securely to the cloud via DataHub, or it can be managed locally and sent to any desired destination, or both.
- DeviceHub's main purpose is to collect data from PLCs (Programmable Logic Controller), classify the data by adding OMA tagging, and publish it to a Message Broker subject to further distribution.
- PLC-level statistics and input/output data can be obtained at the edge using DeviceHub.
- There are 35+ device drivers included in DeviceHub. DeviceHub supports many PLC manufacturers, such as Siemens, Allen-Bradley, Mitsubishi, Omron, etc.

What Can I Configure in DeviceHub?

When you expand **DeviceHub** in the navigation panel, several features enable configuration of a device from which Voice of the Machine Edge will collect data: **Device Discovery**, **Devices**, **Tags**, and **Flows**.



How to Add, Monitor, or Delete a Device

For the purpose of this Getting Started Guide, a procedure to add a Modbus Device in DeviceHub serves as an example of a configuration that is readily available to users.

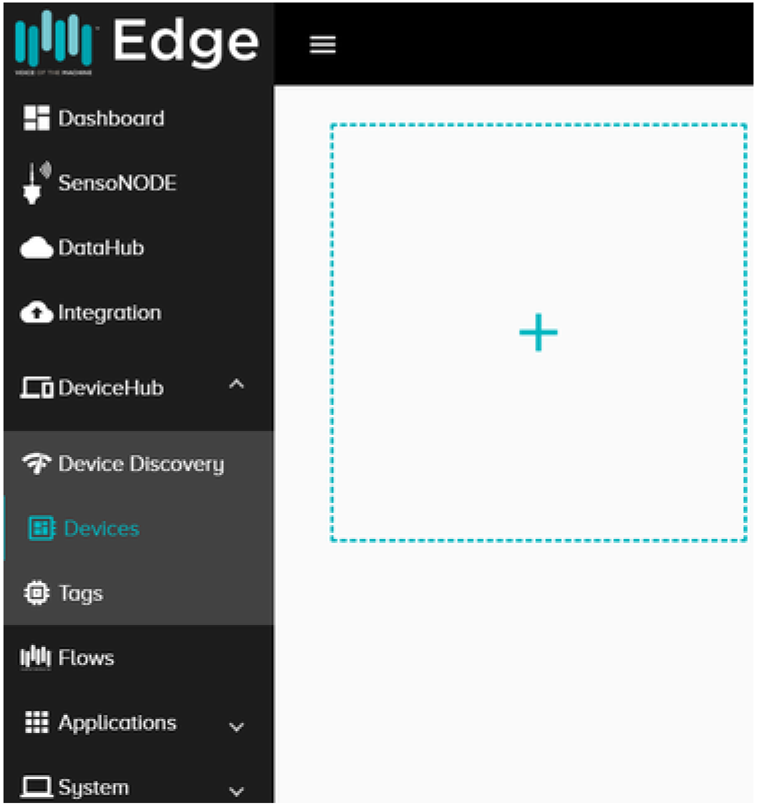
Modbus is a widely used serial communication open protocol for connecting industrial electronic devices. It was designed to be used in programmable logic controllers (PLCs). It uses a simple master-slave configuration to connect the serial ports of two devices. This protocol typically is used to transmit data from sensors and control devices to data-gathering systems such as Voice of the Machine Edge.
Prerequisite
For the example shown on this page, download and install a trial version of the Modbus Slave Simulator to simulate a slave device with which Voice of the Machine Edge will connect.

Add a Device

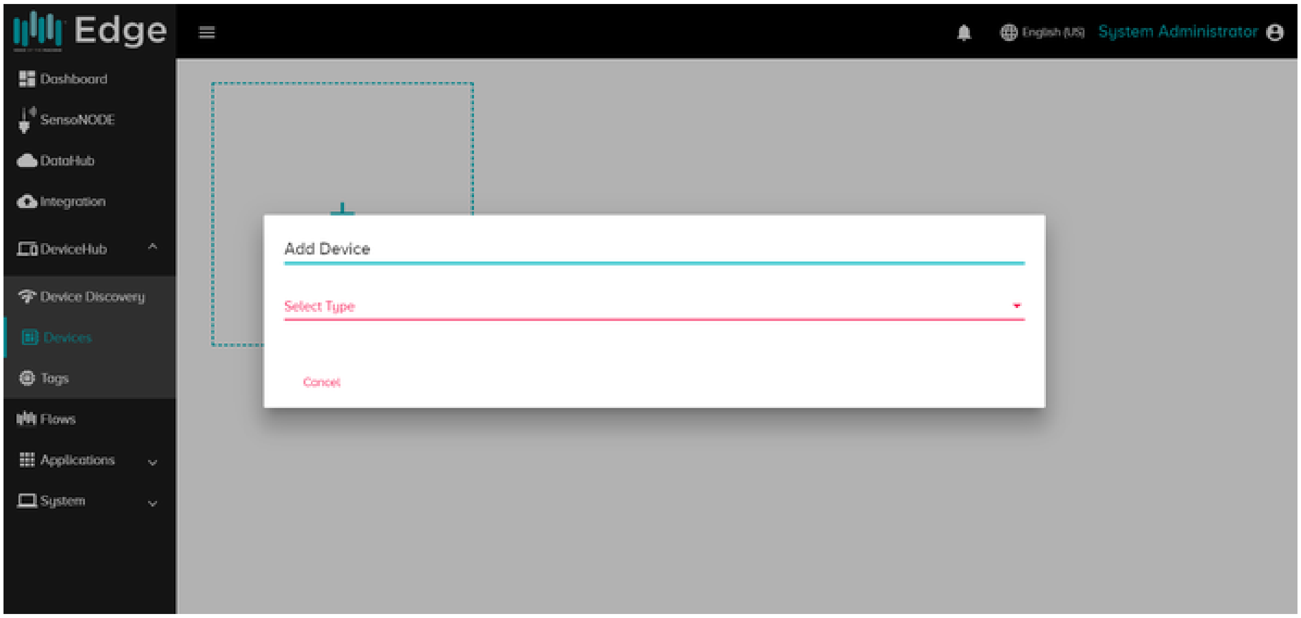
A Modbus device can work on Modbus RTU, which works on a serial interface, and also Modbus TCP, which works on Ethernet IP. More details about the protocol can be found here.

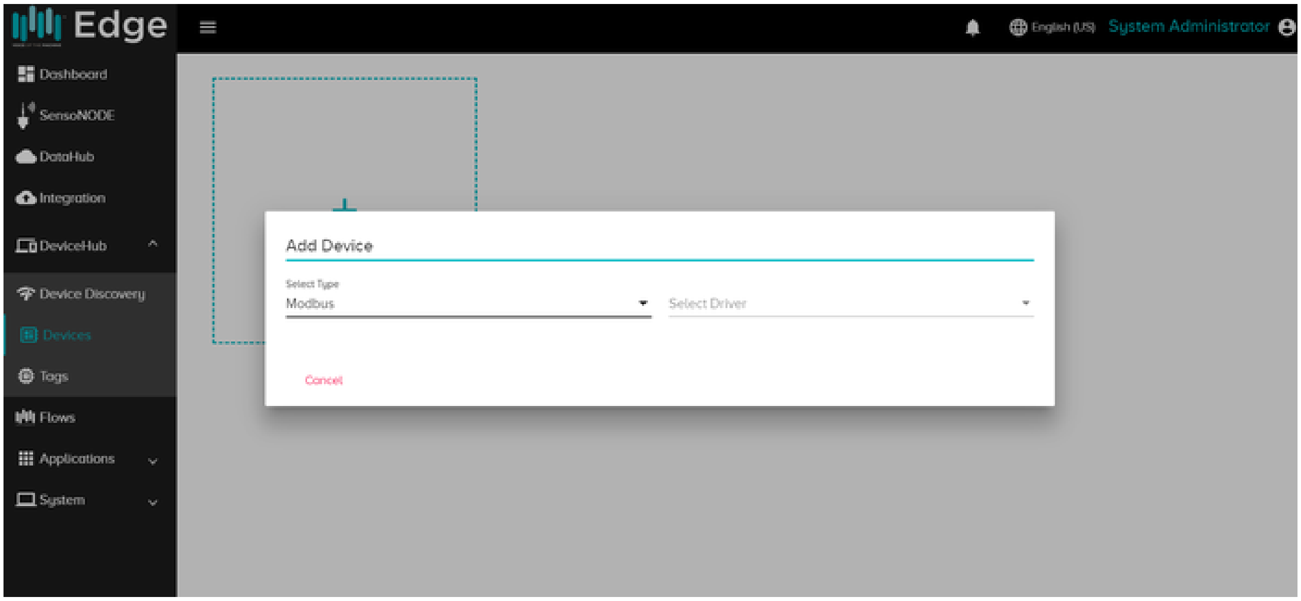
To add a Modbus device to DeviceHub:

1. In the Voice of the Machine Edge navigation panel, click **DeviceHub**.



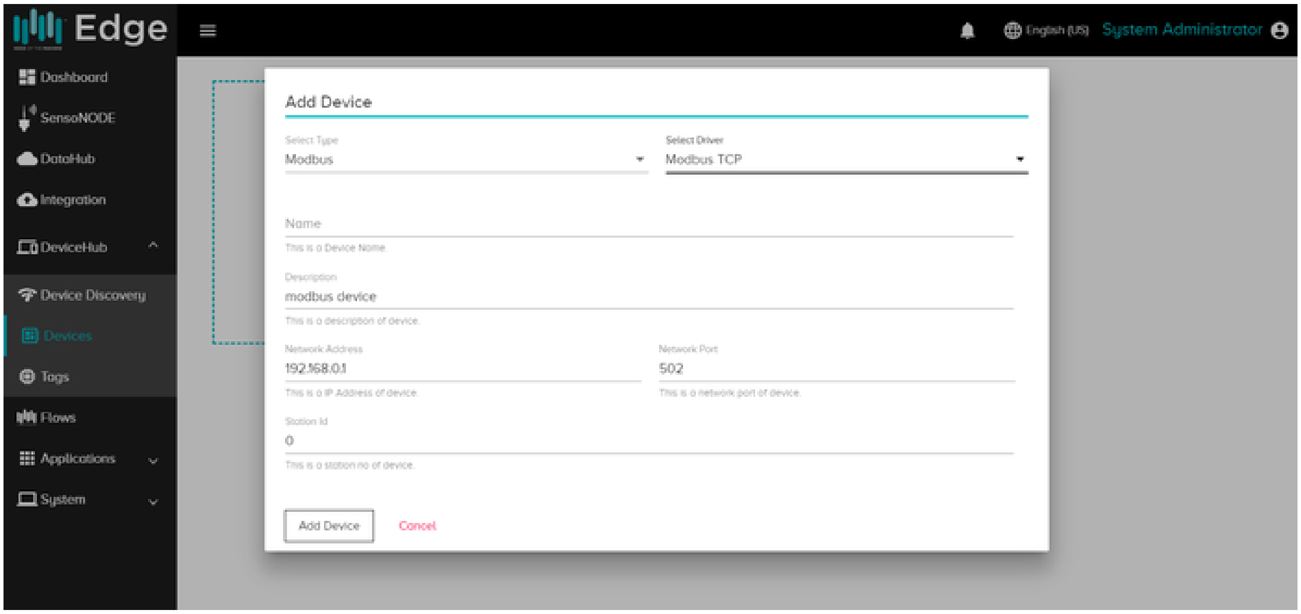
2. Click the **Devices** tab.
3. Click the ➕ in the empty box.
4. In the Add Device dialog, select the device type associated with the device. For this case, select **Modbus**.
   Other device types such as Allen-Bradley, Mitsubishi, Omron, and GE Fanuc may be listed.





Select Type
Modbus

5. Enter a **Device Name** and add an optional **Description**.
6. Enter **Station ID**, **Network Port**, and **Network Address**. This is the IP address of the Modbus device.



Add Device

Select Type: Modbus

Select Driver: Modbus TCP

Name

This is a Device Name.

Description: modbus device

This is a description of device.

Network Address: 192.168.0.1

This is a IP Address of device.

Network Port: 502

This is a network port of device.

Station Id: 0

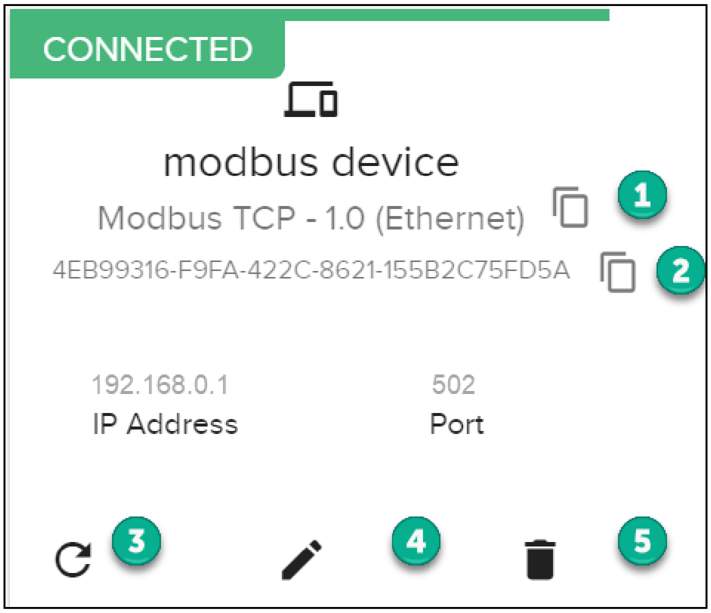
This is a station no of device.

7. Click **Add Device** to create your device in DeviceHub.

Monitor or Delete a DeviceHub Device

When a device is added, a number of icons appear in the device card, as illustrated below, to enable device management.

See also, Troubleshoot Device Connectivity Status.



CONNECTED

modbus device

Modbus TCP - 1.0 (Ethernet)

4EB99316-F9FA-422C-8621-155B2C75FD5A

192.168.0.1
IP Address

502
Port

1. **Copy the device driver name to the clipboard**. This could be used when uploading a large number of tags.
2. **Copy the device ID**. This also could be used when bulk-loading tags.
3. **Refresh the status of the device** when you change a parameter of the PLC.
4. **Edit the device parameters**.
5. **Delete the device**. When deleting a device, you will be shown a warning, *Deleting this device is irreversible*, and you will be prompted to explicitly enter the name of the device. Note that you must type the case-sensitive name exactly as it appears.

After deleting a DeviceHub device, it is recommended that you re-boot the Voice of the Machine Edge device (gateway) before adding another DeviceHub device. It takes some time before the device is actually deleted in the backend.

How to use Device Discovery

How Device Discovery Works

A device discovery service scans the network using the Ethernet interface and returns a list of found devices.

- Discovery is disabled on all interfaces, by default.
- Discovery probes every address in the network at one-second intervals. Therefore, it may take some time to scan an entire network.
- The scan never stops and after 10 minutes, if a device has not been seen on the network, it is removed from the list.

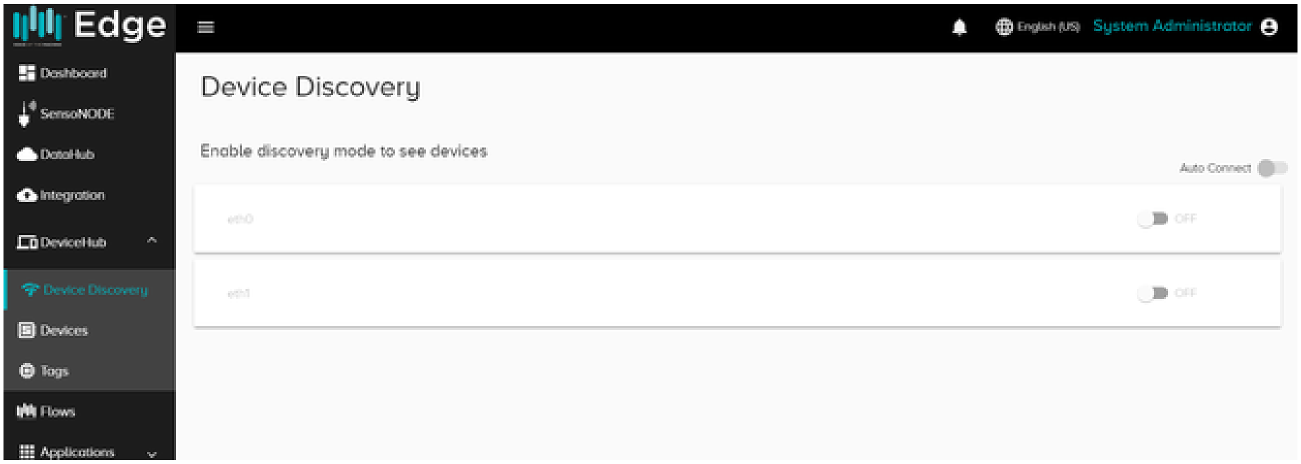
Restrictions Enforced
Due to the constant network activity generated by Device Discovery, certain restrictions are enforced to prevent:

- **Security alerts**: Because discovery regularly scans the network, security systems (such as Trojan detection systems) might flag this activity. To avoid security alerts, network scanning is limited to one-second intervals, with devices dropped after 10 minutes.
- **Missing devices**: Discovery is supported only on subnet /24 networks, to limit the number of nodes that need to be discovered. On networks larger than /24, discovery would likely take longer than 10 minutes to find all the network nodes, which would cause devices to be dropped from the list.

Enable Device Discovery

To enable device discovery:

1. In the navigation panel, expand DeviceHub and select **Device Discovery**.
2. For the Ethernet interface, toggle the switch to **ON**.



Device Discovery

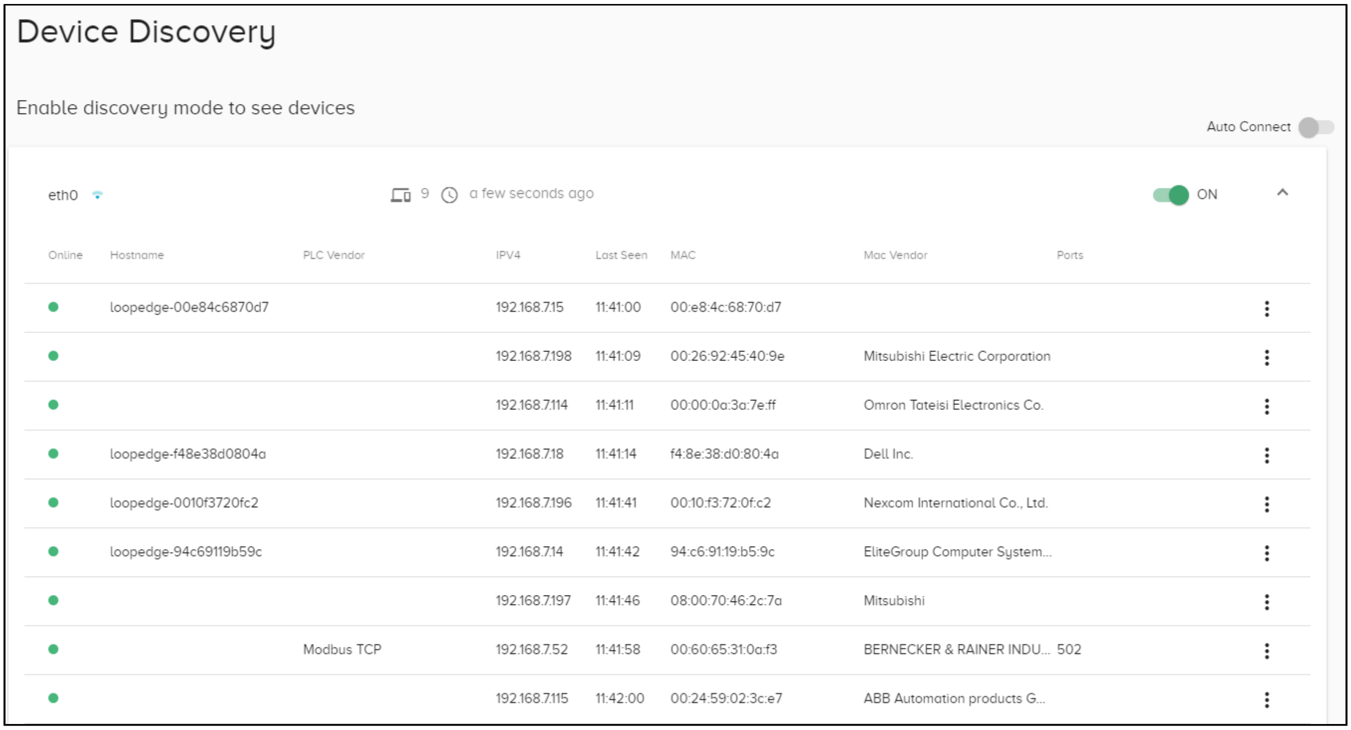
Enable discovery mode to see devices

Auto Connect

eth0 OFF

eth1 OFF

The populated list includes a *Last Seen* column and also indicates the last time Device Discovery scanned the network.



Device Discovery

Enable discovery mode to see devices

eth0 9 a few seconds ago ON

Online	Hostname	PLC Vendor	IPV4	Last Seen	MAC	Mac Vendor	Ports
●	loopedge-00e84c6870d7		192.168.7.15	11:41:00	00:e8:4c:68:70:d7		
●			192.168.7.198	11:41:09	00:26:92:45:40:9e	Mitsubishi Electric Corporation	
●			192.168.7.114	11:41:11	00:00:0a:3a:7e:ff	Omron Tateisi Electronics Co.	
●	loopedge-f48e38d0804a		192.168.7.18	11:41:14	f4:8e:38:d0:80:4a	Dell Inc.	
●	loopedge-0010f3720fc2		192.168.7.196	11:41:41	00:10:f3:72:0f:c2	Nexcom International Co., Ltd.	
●	loopedge-94c69119b59c		192.168.7.14	11:41:42	94:c6:91:19:b5:9c	EliteGroup Computer System...	
●			192.168.7.197	11:41:46	08:00:70:46:2c:7a	Mitsubishi	
●		Modbus TCP	192.168.7.52	11:41:58	00:60:65:31:0a:f3	BERNECKER & RAINER INDU...	502
●			192.168.7.115	11:42:00	00:24:59:02:3c:e7	ABB Automation products G...	

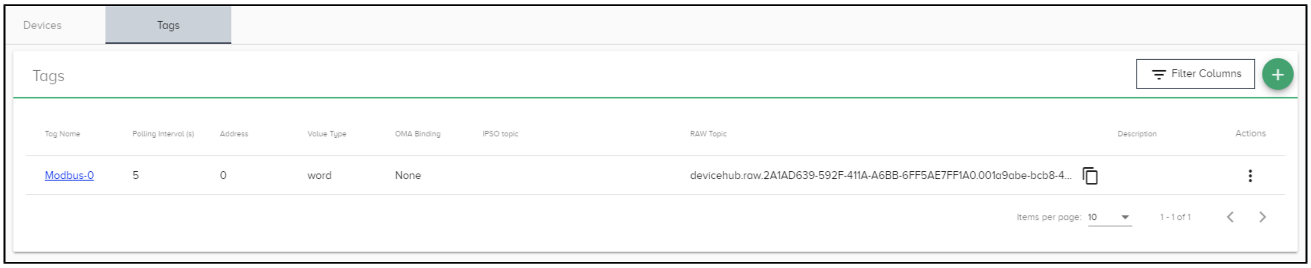
How to Add a Tag in DeviceHub

Once the PLC (Programmable Logic Controller) is connected to the Voice of the Machine Edge IPC (Industrial PC) device using DeviceHub, you can define the PLC registers that DeviceHub will interrogate. Every PLC has memory registers that store the sensor and system values, such as the timer counter. The list of registers includes the analog input, the analog output, the digital input, and the digital output.
See also, DeviceHub Data Publishing Formats.

DeviceHub Tag Overview

A DeviceHub **Tag** enables Voice of the Machine Edge to collect data from a specific memory location (register) in a PLC. The DeviceHub Tag generates a Raw Topic, used to publish data to a message broker for further distribution. The data to all topics is collected on a local NATS topic and it then can be published to another NATS topic that is connected to the cloud connector. You also can directly connect the data to the cloud connector.

- A DeviceHub Tag generates a *Raw Topic*, essential for NATS publishing and subscribing.
- The Raw Topic will be used to configure a DataHub Cloud Connector and Topics to connect to Voice of the Machine Cloud.
- In lieu of Voice of the Machine Cloud, collected data can be viewed at the edge using Voice of the Machine Edge Flows. See Use Flows to Visualize Data Flows.



Tag Name	Polling Interval (s)	Address	Value Type	OMA Binding	IPSO topic	RAW Topic	Description	Actions
Modbus-0	5	0	word	None		devicehub.raw.2A1AD639-592F-411A-A6BB-6FF5AE7FF1A0.001a9abe-bcb8-4...		⋮

Items per page: 10 1 - 1 of 1

The above example shows the tag attributes to be configured.

- **Tag Name** - User-defined name
- **Polling Interval** - Elapsed time in seconds to poll the PLC register
- **Address** - Register in the PLC to be interrogated
- **Value Type** - Data type of the value stored in the register
- **Raw Topic** - Local message broker topic
- **IPSO Topic** - After OMA binding is applied, an IPSO topic is created to send data in the OMA format (See DeviceHub OMA Binding)

Add a Tag

In DeviceHub, you can add a tag for a single PLC register or use the bulk upload feature to upload details listed in a comma-separated values (.csv) file. This procedure lists the steps for a single register.
Prerequisite
A DeviceHub device must exist in Voice of the Machine Edge. See DeviceHub Add, Monitor, or Delete Device.

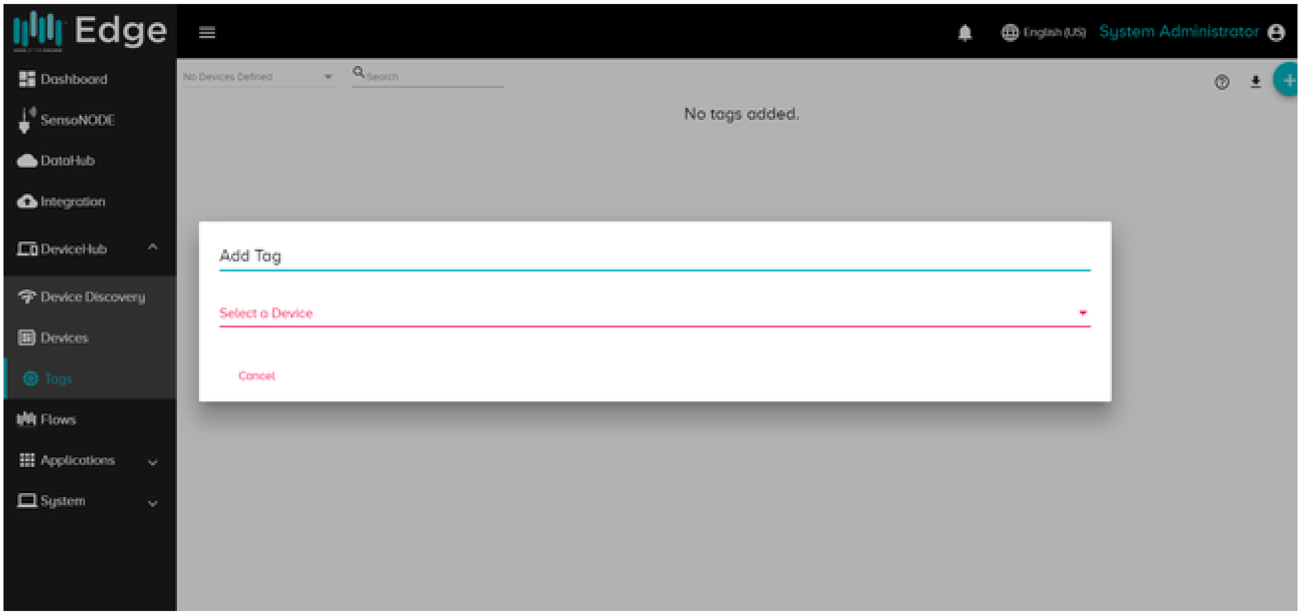
- Certain device drivers, such as Allen-Bradley CompactLogix, will *not* have a list of register names in the Add Tag form. These cases often are referred to as *Free Tags*.
- When a list of registers is not provided, the case-sensitive register name must be entered exactly as it is defined in the PLC.
- In these cases, the device will not display a *Connected* status until a tag with a correct register name is created. See Troubleshoot Device Connectivity Status.

To add a tag for a single register:

After a PLC device is connected to the Voice of the Machine Edge IPC using DeviceHub, create tags to define the device registers from which data will be collected.

Note: These steps show an *example* of a Modbus configuration. Each device driver will have its own specific fields to be configured.

1. In the navigation panel, select **DeviceHub**.
2. Click the **Tags** tab.
3. Click the + icon at the right of the Tags list.
4. Select **Add One**.
5. Select the Device from the drop-down list. Only devices created in DeviceHub will be listed.
   The fields displayed in this Add Tag dialog depend on the device's driver type.



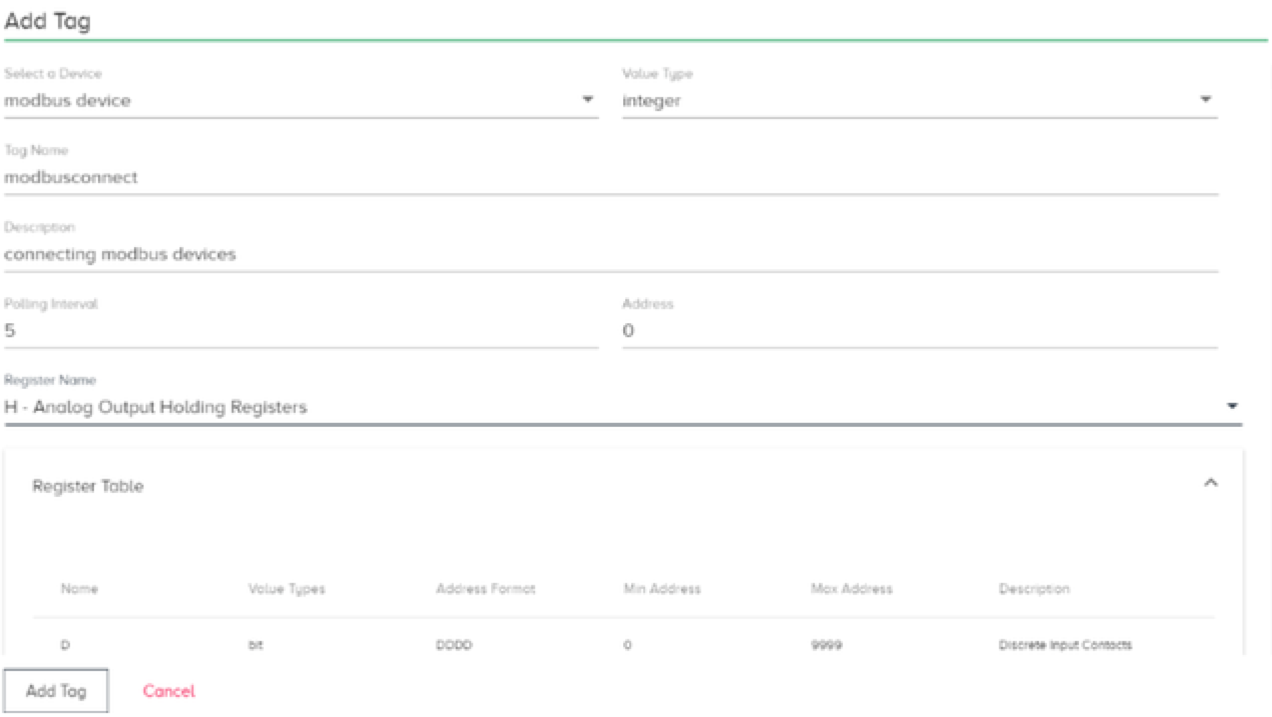
No tags added.

Add Tag

Select a Device

Cancel

6. Select the **Value Type:** string, integer, float, or bit.
7. Enter a **Tag Name** and **Polling Interval** (in seconds).



Add Tag

Select a Device: modbus device

Value Type: integer

Tag Name: modbusconnect

Description: connecting modbus devices

Polling Interval: 5

Address: 0

Register Name: H - Analog Output Holding Registers

Register Table

Name	Value Types	Address Format	Min Address	Max Address	Description
D	bit	DDDD	0	9999	Discrete Input Contacts

8. Select/enter a **Register Name** (Scroll to reference the register table listed at the bottom of the form) and click **Add Tag**.

- Certain device drivers, such as Allen-Bradley CompactLogix, will *not* have a list of register names in the Add Tag form. These cases often are referred to as *Free Tags*.
- When a list of registers is not provided, the case-sensitive register name must be entered exactly as it is defined in the PLC.
- In these cases, the device will not display a *Connected* status until a tag with a correct register name is created.

The data obtained on the local message topic (raw topic) can be published to another local message topic, which is connected to the DataHub cloud connector The user can also directly connect the data to the cloud connector.

OMA tagging
Use DeviceHub Tags to bind with IPSO objects to send collected data in the OMA format. See DeviceHub OMA Binding and DeviceHub Data Publishing Formats.

How to use OMA Binding in DeviceHub

- OMA Binding
- IPSO Object Reference Guide

What is OMA Binding?

When OMA (Open Mobile Alliance) binding is applied to a DeviceHub tag, the collected data will be formatted according to the OMA standard and it will have the structure required for further processing. Once this format is applied, the resulting IPSO (Internet Protocol for Smart Objects) topic ensures that all the collected values will be in the proper format so that the data can be saved in a database and used for analysis in other applications.

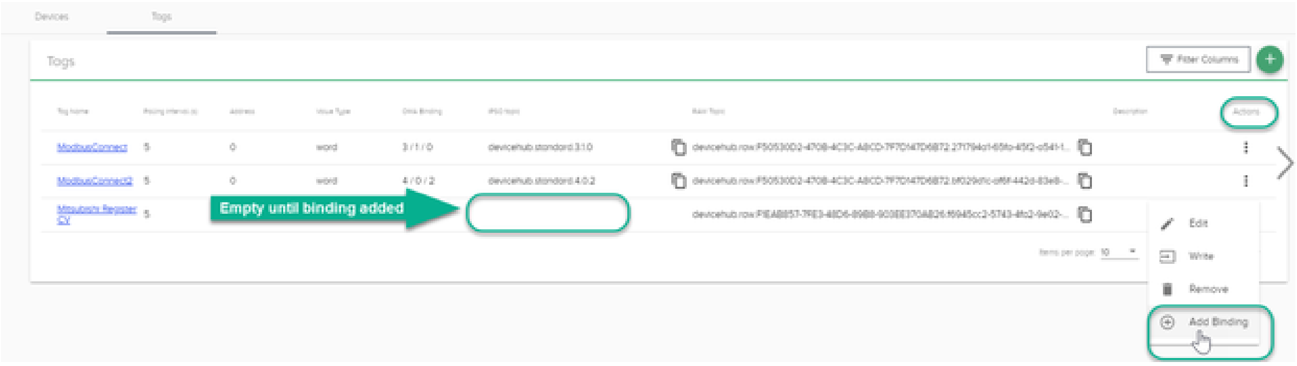
Object Binding in Voice of the Machine Edge

In Voice of the Machine Edge, use a **DeviceHub Tag** to bind to a specific object-instance-resource.
Prerequisites
This procedure assumes that you have created **DeviceHub Devices** and **Tags**.

- DeviceHub Add, Monitor, or Delete Device
- DeviceHub Add a Tag

To apply OMA binding to an object:

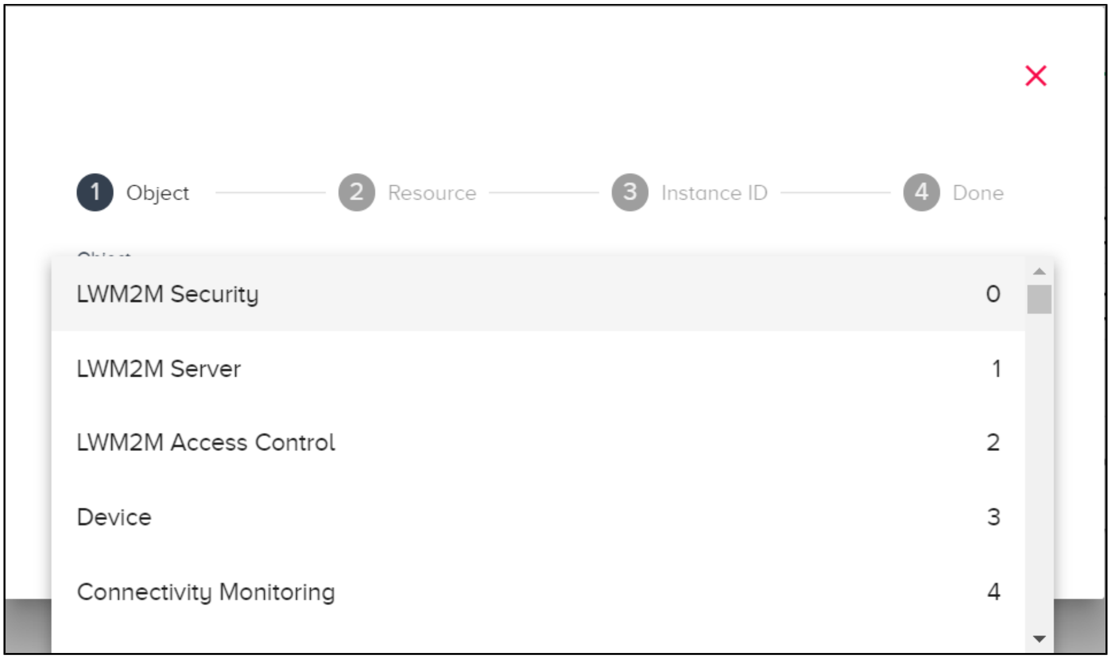
1. In the navigation pane, go to **DeviceHub** and click the **Tags** tab.
2. In the **Actions** list to the right of the tag, select **Add Binding**.

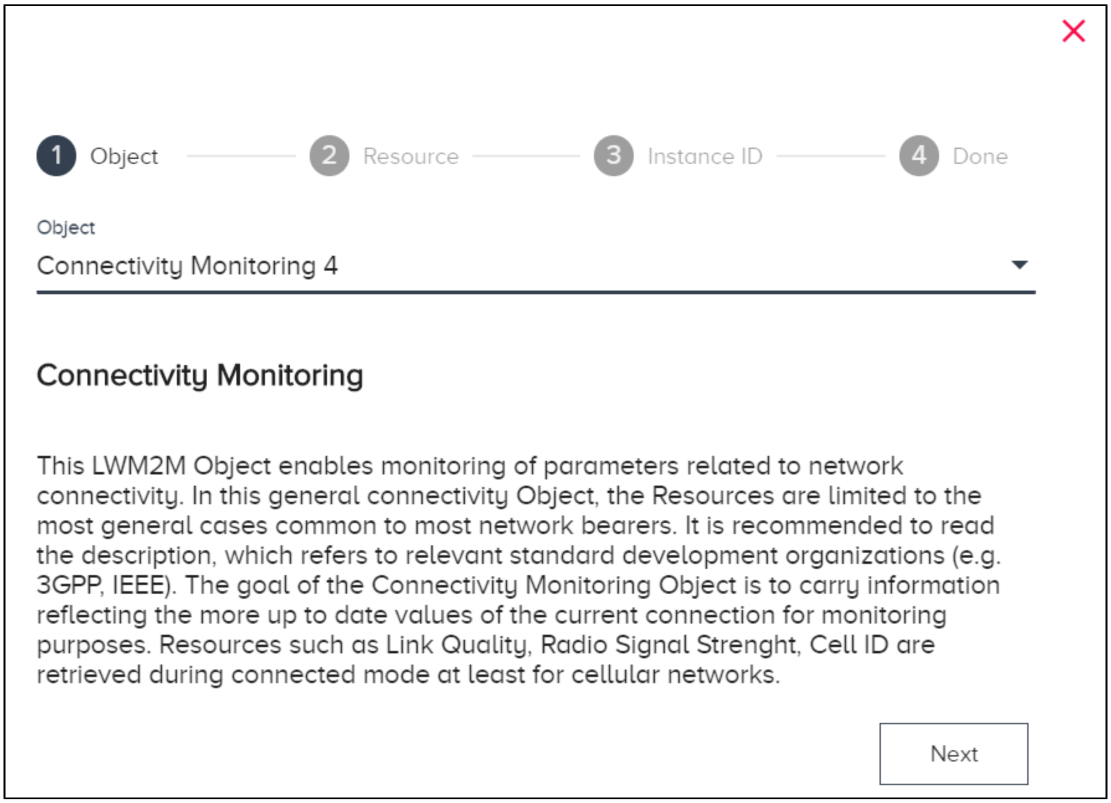


Tag Name	Polling Interval (s)	Address	Value Type	OMA Binding	IPSO topic	Raw Topic	Description	Actions
ModbusConnect	5	0	word	3 / 1 / 0	devicehub.standard.3.1.0	devicehub.raw.F50530D2-4708-4C3C-A8CD-7F7D147D6872.27179a1-65fa-45f2-a541-1...		⋮
ModbusConnect2	5	0	word	4 / 0 / 2	devicehub.standard.4.0.2	devicehub.raw.F50530D2-4708-4C3C-A8CD-7F7D147D6872.bf029dfc-af6f-442d-83e8-...		⋮
Mitsubishi Register CV	5					devicehub.raw.F1EAB857-7FE3-48D6-89B8-9038E370AB26.f6945cc2-5743-4fa2-9e02-...		

Empty until binding added

3. Select the **Object** from the drop-down list and click **Next.**
   0 - LWM2M Security
   1 - LWM2M Server
   2 - LWM2M Access Control
   3 - Device
   4 - Connectivity Monitoring
   Scroll for more objects.



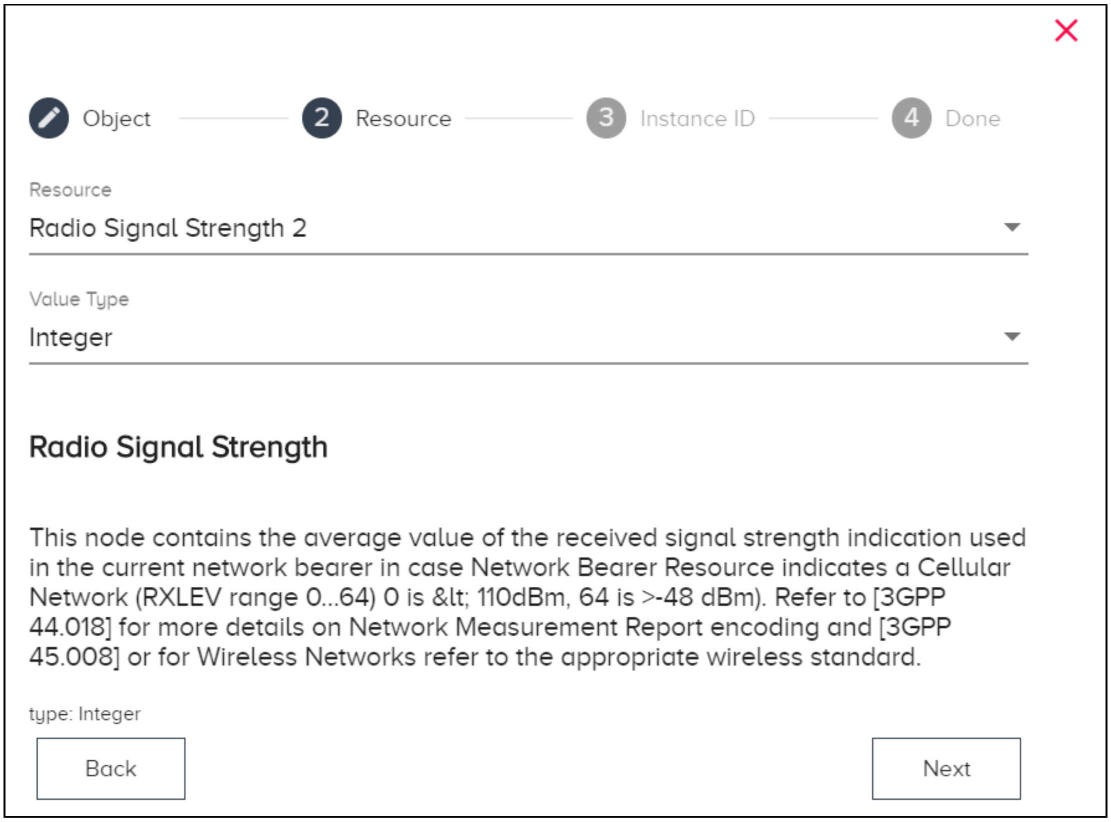


Object
Connectivity Monitoring 4

Connectivity Monitoring

This LWM2M Object enables monitoring of parameters related to network connectivity. In this general connectivity Object, the Resources are limited to the most general cases common to most network bearers. It is recommended to read the description, which refers to relevant standard development organizations (e.g. 3GPP, IEEE). The goal of the Connectivity Monitoring Object is to carry information reflecting the more up to date values of the current connection for monitoring purposes. Resources such as Link Quality, Radio Signal Strenght, Cell ID are retrieved during connected mode at least for cellular networks.

4. Select a **Resource** and **Value Type** and click **Next**.
   Note that the value type will default to a value that is relevant for the resource: String, Integer, Float, Boolean, or Opaque.
   The opaque data type is a sequence of binary octets. An example of this data type is in the Temperature IPSO object, where the *Reset Min and Max Measured Values* resource (3303/0/5605) is an opaque data type.



Resource
Radio Signal Strength 2

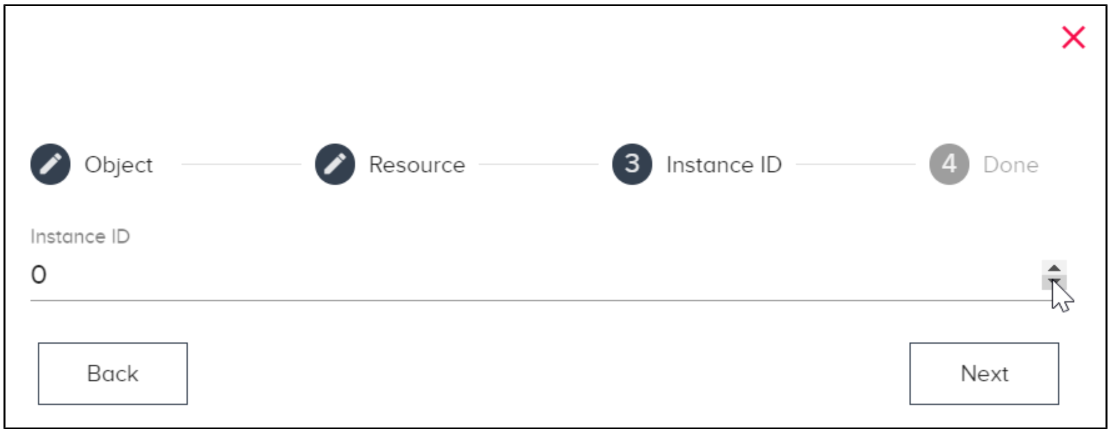
Value Type
Integer

Radio Signal Strength

This node contains the average value of the received signal strength indication used in the current network bearer in case Network Bearer Resource indicates a Cellular Network (RXLEV range 0...64) 0 is &lt; 110dBm, 64 is >-48 dBm). Refer to [3GPP 44.018] for more details on Network Measurement Report encoding and [3GPP 45.008] or for Wireless Networks refer to the appropriate wireless standard.

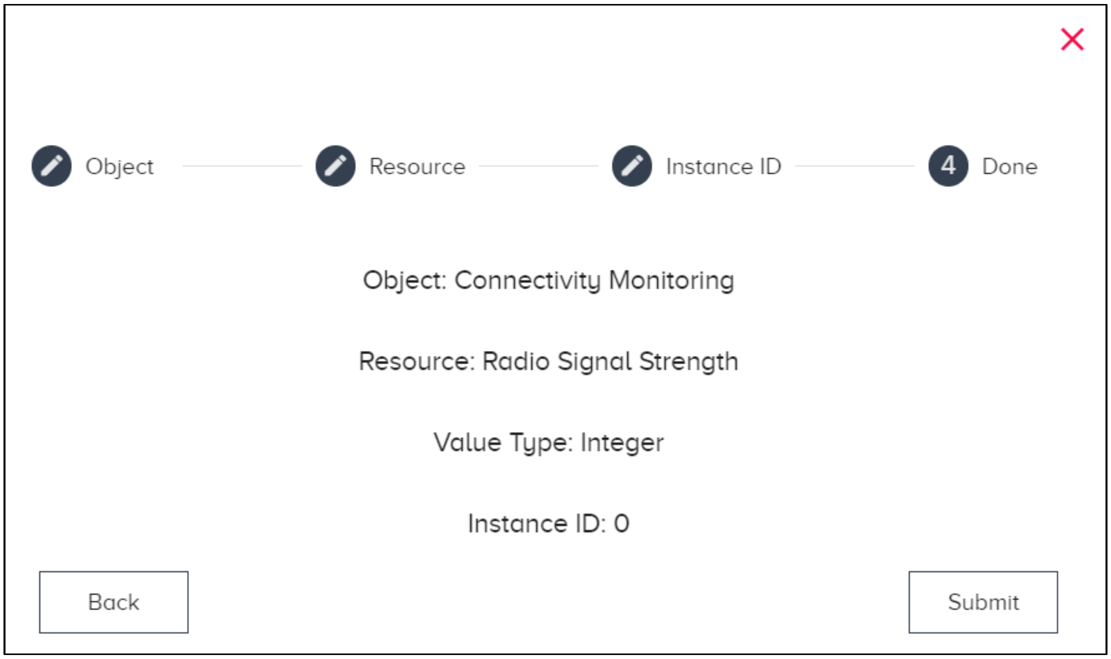
type: Integer

5. Select a numeric **Instance ID** and click **Next**.



Instance ID
0

6. Review the configuration and if it is correct, click **Submit**. Otherwise, click **Back** to re-configure.



Object: Connectivity Monitoring

Resource: Radio Signal Strength

Value Type: Integer

Instance ID: 0

Results

- In this example, the OMA Binding is represented as:**4/0/2**

An IPSO Topic is created and displayed in the Tags list. The topic name identifies the format. For this example the IPSO topic would be:

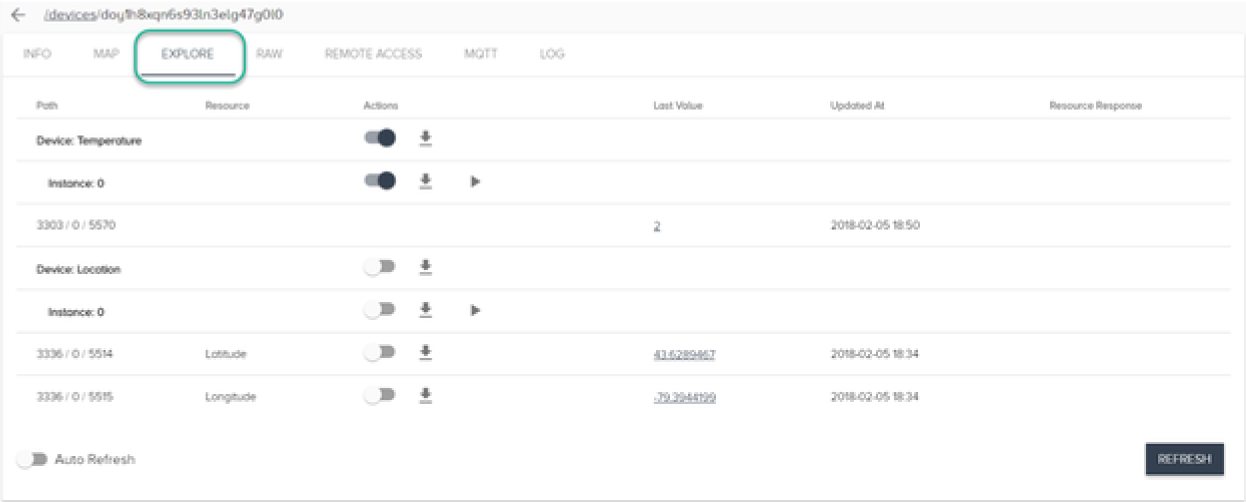
```
devicehub.standard.4.0.2
```

View a Device in Voice of the Machine Cloud After OMA Binding

After OMA binding has been applied, data can be viewed in Voice of the Machine Cloud in the device's Explore tab. Prior to OMA binding, view the data in the device's Raw tab.

To view a device in Voice of the Machine Cloud after OMA binding:

1. In Voice of the Machine Cloud, after selecting a Project, navigate to **Device Mgmt > Devices**.
2. Click the **Voice of the Machine Edge** tab.
3. Click the **Device ID** for the relevant device.
4. In the Device window, click the **Explore** tab.



Path	Resource	Actions	Last Value	Updated At	Resource Response
Device: Temperature					
Instance: 0					
3303 / 0 / 5570			2	2018-02-05 18:50	
Device: Location					
Instance: 0					
3336 / 0 / 5514	Latitude		43.6289467	2018-02-05 18:34	
3336 / 0 / 5515	Longitude		-79.3944199	2018-02-05 18:34	

IPSO Object Reference Guide

The Internet Protocol for Smart Objects (IPSO) defines an object model, based on the Open Mobile Alliance (OMA) standard for the Internet of

Things (IoT), to achieve interoperability for data transmitted between devices and application software. A common set of object definitions enables software to interact with any device.

Use these sections as an overview of IPSO objects.

- What are IPSO Objects?
- Smart Object Standards
  - Example of an Actuation Object
  - Example of a Temperature Sensor Object
- Objects Defined by the OMA Standard
  - LWM2M Objects
  - IPSO Objects (See also, OMA Data Objects in Detail)

- DeviceHub OMA Binding - Bind objects in Voice of the Machine Edge to format the collected data according to the OMA standard so that it will have the structure required for further processing.

What are IPSO Objects?

Simply put, IPSO objects based on the OMA standard represent anything that can be measured.

IoT devices contain one or more of these components from which measurements can be collected:

- **Sensor**: A device sensor detects a condition/state and then converts one form of energy into another. When converted to electrical energy, sensor values can be read and the device can be managed.
- **Actuator**: An actuator takes electrical energy and converts it into an action, such as turning a device on or off.

Smart Object Standards

The smart object model has the following characteristics:

- Standard protocol-independent format.
- Objects define anything that provides data about a device. For example, a light sensor is one object while digital input is another object—each with its own unique object ID.
- An object represents a single measurement value (such as temperature), a control point (such as an on/off switch), or an actuator (such as a dimmer).
- Each object can have multiple resources, such as an on/off switch or the time a device was turned on. These resources can be accessed, read, and managed.
- Each object is defined with an **Object ID** and each of the object's resources has a **Resource ID** that defines an object's observable property, such as a temperature or humidity sensor, or an on/off switch or dimmer. An **Instance ID** is used to identify a specific object, when there are multiple occurrences of an object. Object Instances serve as containers for resources (observable properties or targets for actuation).

The URI (Uniform Resource Identifier) for an object follows this format:

```
Object_ID/Instance_ID/Resource_ID
```

Example of an Actuation Object

An IPSO object for remote actuation, such as turning a device on/off or dimming a light, can have a variety of resources.

- Actuation Object ID: 3306
- Resource ID for the on/off switch: 5850
- Resource ID for the dimmer: 5851
- Resource ID for the time the device has been turned on: 5852
- Resource ID for the state of multi-level output (such as a pilot wire): 5853
- Resource ID for the device's application type (such as motion closure): 5750
- Instance ID: 0 (for a single occurrence of the object)

The JSON schema for an Actuation object:

```
{ "timestamp": 0, "values": [ { "objectId": 3306, "instanceId": 0, "resourceId": 5850, "datatype":
"Boolean", "value": 1 } ] }
```

Example of a Temperature Sensor Object

- Temperature Sensor Object ID: 3303
- Resource ID for Sensor Value: 5700

- Resource ID for Minimum Measured Value: 5601
- Resource ID for Maximum Measured Value: 5602
- Resource ID for Minimum Value Possible for Sensor: 5603
- Resource ID for Maximum Value Possible for Sensor: 5604
- Resource ID for Resetting the Min/Max Values to Current Value: 5605
- Instance ID: 0 (for a single occurrence of the object)

JSON schema for a Temperature Sensor object:

```
{ "timestamp": 0, "values": [ { "objectId": 3303, "instanceId": 0, "resourceId": 5700, "datatype": "Float",
"value": 12.12 } ] }
```

Objects Defined by the OMA Standard

Additional information can be found on the Open Mobile Alliancewebsite.

LWM2M Objects

Object	Object ID	Minimum JSON (Resource for LWM2M Object Definitions: OMA LWM2M Objects)
LWM2M Security	0	{ "timestamp": 0, "values": [{ "objectId": 0, "instanceId": 0, "resourceId": 0, "datatype": "String", "value": ID in the format of coaps://host:port }, { "objectId": 0, "instanceId": 0, "resourceId": 1, "datatype": "Boolean", "value": 1 }, { "objectId": 0, "instanceId": 0, "resourceId": 2, "datatype": "Integer", "value": 0-3 }, { "objectId": 0, "instanceId": 0, "resourceId": 3, "datatype": "Opaque", "value": "" }, { "objectId": 0, "instanceId": 0, "resourceId": 4, "datatype": "Opaque", "value": "" }, { "objectId": 0, "instanceId": 0, "resourceId": 5, "datatype": "Opaque", "value": "" }, { "objectId": 0, "instanceId": 0, "resourceId": 6, "datatype": "Integer", "value": 0-255 }, { "objectId": 0, "instanceId": 0, "resourceId": 7, "datatype": "Opaque", "value": "" }, { "objectId": 0, "instanceId": 0, "resourceId": 8, "datatype": "Opaque", "value": "" }, { "objectId": 0, "instanceId": 0, "resourceId": 9, "datatype": "Integer", "value": "" }, { "objectId": 0, "instanceId": 0, "resourceId": 11, "datatype": "Integer", "value": "" }] }
LWM2M Server	1	{ "timestamp": 0, "values": [{ "objectId": 1, "instanceId": 0, "resourceId": 0, "datatype": "Integer", "value": 1 }, { "objectId": 1, "instanceId": 0, "resourceId": 1, "datatype": "Integer", "value": 1 }, { "objectId": 1, "instanceId": 0, "resourceId": 6, "datatype": "Boolean", "value": 1 }, { "objectId": 1, "instanceId": 0, "resourceId": 7, "datatype": "String", "value": possible values of Resource are listed in 5.2.1.1 }, { "objectId": 1, "instanceId": 0, "resourceId": 8, "datatype": "", "value": },] }
Access Control	2	{ "timestamp": 0, "values": [{ "objectId": 2, "instanceId": 0, "resourceId": 0, "datatype": "Integer", "value": 1-65534 }, { "objectId": 2, "instanceId": 0, "resourceId": 1, "datatype": "Integer", "value": 0-65535 }, { "objectId": 2, "instanceId": 0, "resourceId": 3, "datatype": "Integer", "value": 0-65535 }] }
Device	3	{ "timestamp": 0, "values": [{ "objectId": 3, "instanceId": 0, "resourceId": 4, "datatype": "", "value": "" }, { "objectId": 3, "instanceId": 0, "resourceId": 11, "datatype": "Integer", "value": 0 }, { "objectId": 3, "instanceId": 0, "resourceId": 16, "datatype": "String", "value": possible values of Resource are combination of \"U\" or \"UQ\" and \"S\" or \"SQ\" }] }
Connectivity Monitoring	4	{ "timestamp": 0, "values": [{ "objectId": 4, "instanceId": 0, "resourceId": 0, "datatype": "Integer", "value": network bearer used for the current LWM2M communication session }, { "objectId": 4, "instanceId": 0, "resourceId": 1, "datatype": "Integer", "value": value from the network bearer list }, { "objectId": 4, "instanceId": 0, "resourceId": 2, "datatype": "Integer", "value": average value of the received signal strength }, { "objectId": 4, "instanceId": 0, "resourceId": 4, "datatype": "String", "value": IP addresses assigned to the connectivity interface. (e.g. IPv4, IPv6) },] }

Firmware	5	{ "timestamp": 0, "values": [{ "objectId": 5, "instanceId": 0, "resourceId": 0, "datatype": "Opaque", "value": "" }, { "objectId": 5, "instanceId": 0, "resourceId": 1, "datatype": "String", "value": URI from where the device can download the firmware package by an alternative mechanism }, { "objectId": 5, "instanceId": 0, "resourceId": 2, "datatype": "", "value": "" }, { "objectId": 5, "instanceId": 0, "resourceId": 3, "datatype": "Integer", "value": "1" }, { "objectId": 5, "instanceId": 0, "resourceId": 5, "datatype": "Integer", "value": "0" }] }
Location	6	{ "timestamp": 0, "values": [{ "objectId": 6, "instanceId": 0, "resourceId": 0, "datatype": "String", "value": decimal notation of latitude }, { "objectId": 6, "instanceId": 0, "resourceId": 0, "datatype": "String", "value": decimal notation of longitude}, { "objectId": 6, "instanceId": 0, "resourceId": 5, "datatype": "time", "value": timestamp of measurement}] }
Connectivity Statistics	7	{ "timestamp": 0, "values": [{ "objectId": 7, "instanceId": 0, "resourceId": 6, "datatype": "", "value": "" }] }
Lock and Wipe	8	{ "timestamp": 0, "values": [{ "objectId": 8, "instanceId": 0, "resourceId": 0, "datatype": "Integer", "value": 0 }, { "objectId": 8, "instanceId": 0, "resourceId": 1, "datatype": "String", "value": specify one or several targets for the lock operation }] }
Software Management	9	{ "timestamp": 0, "values": [{ "objectId": 9, "instanceId": 0, "resourceId": 0, "datatype": "String", "value": name of the software package }, { "objectId": 9, "instanceId": 0, "resourceId": 1, "datatype": "String", "value": version of the software package }, { "objectId": 9, "instanceId": 0, "resourceId": 2, "datatype": "Opaque", "value": software package }, { "objectId": 9, "instanceId": 0, "resourceId": 3, "datatype": "String", "value": URI for software package download }, { "objectId": 9, "instanceId": 0, "resourceId": 4, "datatype": "", "value": "" }, { "objectId": 9, "instanceId": 0, "resourceId": 6, "datatype": "", "value": "" }, { "objectId": 9, "instanceId": 0, "resourceId": 7, "datatype": "Integer", "value": 1-5 }, { "objectId": 9, "instanceId": 0, "resourceId": 9, "datatype": "Integer", "value": 0-10 }, { "objectId": 9, "instanceId": 0, "resourceId": 10, "datatype": "", "value": "" }, { "objectId": 9, "instanceId": 0, "resourceId": 11, "datatype": "", "value": "" }, { "objectId": 9, "instanceId": 0, "resourceId": 12, "datatype": "Boolean", "value": 1 }] }
Cellular Connectivity	10	
APN Connection Profile	11	
WLAN Connectivity	12	
Bearer Selection	13	
Software Component	14	
DevCapMgmt	15	
Portfolio	16	
Communications Characteristics	17	
Non-Access Stratum (NAS) Configuration	18	
LWM2M AppData	19	

IPSO Objects (See also,OMA Data Objects in Detail)

Object	Object ID	MInimum JSON
Digital	3200	{ "timestamp": 0, "values": [{ "objectId": 3200, "instanceId": 0, "resourceId": 5500, "datatype": "Boolean", "value": 1 }] }
Digital Output	3201	{ "timestamp": 0, "values": [{ "objectId": 3201, "instanceId": 0, "resourceId": 5550, "datatype": "Boolean", "value": 1 }] }
Analogue Input	3202	{ "timestamp": 0, "values": [{ "objectId": 3202, "instanceId": 0, "resourceId": 5600, "datatype": "Float", "value": 10.2 }] }
Analogue Output	3203	{ "timestamp": 0, "values": [{ "objectId": 3203, "instanceId": 0, "resourceId": 5650, "datatype": "Float", "value": 10.2 }] }
Generic Sensor	3300	{ "timestamp": 0, "values": [{ "objectId": 3300, "instanceId": 0, "resourceId": 5700, "datatype": "Float", "value": 10.2 }] }
Illuminance Sensor	3301	{ "timestamp": 0, "values": [{ "objectId": 3301, "instanceId": 0, "resourceId": 5700, "datatype": "Float", "value": 10.2 }] }
Presence Sensor	3302	{ "timestamp": 0, "values": [{ "objectId": 3302, "instanceId": 0, "resourceId": 5500, "datatype": "Boolean", "value": 0 }] }
Temperature Sensor	3303	{ "timestamp": 0, "values": [{ "objectId": 3303, "instanceId": 0, "resourceId": 5700, "datatype": "Float", "value": 12.12 }] }
Humidity Sensor	3304	{ "timestamp": 0, "values": [{ "objectId": 3304, "instanceId": 0, "resourceId": 5700, "datatype": "Float", "value": 12.12 }] }
Power Measurement	3305	{ "timestamp": 0, "values": [{ "objectId": 3305, "instanceId": 0, "resourceId": 5800, "datatype": "Float", "value": 12.12 }] }
Actuation	3306	{ "timestamp": 0, "values": [{ "objectId": 3306, "instanceId": 0, "resourceId": 5850, "datatype": "Boolean", "value": 1 }] }
Set Point	3308	{ "timestamp": 0, "values": [{ "objectId": 3308, "instanceId": 0, "resourceId": 5900, "datatype": "float", "value": 11.3 }] }
Load Control	3310	{ "timestamp": 0, "values": [{ "objectId": 3310, "instanceId": 0, "resourceId": 5823, "datatype": "String", "value": "Event identifier" }, { "objectId": 3310, "instanceId": 0, "resourceId": 5824, "datatype": "String", "value": "Start Time" }, { "objectId": 3310, "instanceId": 0, "resourceId": 5825, "datatype": "String", "value": "Duration in minimum" }] }
Light Control	3311	{ "timestamp": 0, "values": [{ "objectId": 3311, "instanceId": 0, "resourceId": 5850, "datatype": "Boolean", "value": 1 }] }
Power Control	3312	{ "timestamp": 0, "values": [{ "objectId": 3312, "instanceId": 0, "resourceId": 5850, "datatype": "Boolean", "value": 1 }] }
Accelerometer	3313	{ "timestamp": 0, "values": [{ "objectId": 3313, "instanceId": 0, "resourceId": 5702, "datatype": "Float", "value": 11.2 }, { "objectId": 3313, "instanceId": 0, "resourceId": 5703, "datatype": "Float", "value": 11.2 }, { "objectId": 3313, "instanceId": 0, "resourceId": 5704, "datatype": "Float", "value": 11.2 }] }

Magnetometer	3314	`{ "timestamp": 0, "values": [ { "objectId": 3314, "instanceId": 0, "resourceId": 5702, "datatype": "Float", "value": 11.2 } ] }`
Barometer	3315	`{ "timestamp": 0, "values": [ { "objectId": 3315, "instanceId": 0, "resourceId": 5700, "datatype": "Float", "value": 11.2 } ] }`
Voltage	3316	`{ "timestamp": 0, "values": [ { "objectId": 3316, "instanceId": 0, "resourceId": 5700, "datatype": "Float", "value": 11.2 } ] }`
Current	3317	`{ "timestamp": 0, "values": [ { "objectId": 3317, "instanceId": 0, "resourceId": 5700, "datatype": "Float", "value": 11.2 } ] }`
Frequency	3318	`{ "timestamp": 0, "values": [ { "objectId": 3318, "instanceId": 0, "resourceId": 5700, "datatype": "Float", "value": 11.2 } ] }`
Depth	3319	`{ "timestamp": 0, "values": [ { "objectId": 3319, "instanceId": 0, "resourceId": 5700, "datatype": "Float", "value": 11.2 } ] }`
Percentage	3320	`{ "timestamp": 0, "values": [ { "objectId": 3320, "instanceId": 0, "resourceId": 5700, "datatype": "Float", "value": 11.2 } ] }`
Altitude	3321	`{ "timestamp": 0, "values": [ { "objectId": 3321, "instanceId": 0, "resourceId": 5700, "datatype": "Float", "value": 11.2 } ] }`
Load	3322	`{ "timestamp": 0, "values": [ { "objectId": 3322, "instanceId": 0, "resourceId": 5700, "datatype": "Float", "value": 11.2 } ] }`
Pressure	3323	`{ "timestamp": 0, "values": [ { "objectId": 3323, "instanceId": 0, "resourceId": 5700, "datatype": "Float", "value": 11.2 } ] }`
Loudness	3324	`{ "timestamp": 0, "values": [ { "objectId": 3324, "instanceId": 0, "resourceId": 5700, "datatype": "Float", "value": 11.2 } ] }`
Concentration	3325	`{ "timestamp": 0, "values": [ { "objectId": 3325, "instanceId": 0, "resourceId": 5700, "datatype": "Float", "value": 11.2 } ] }`
Acidity	3326	`{ "timestamp": 0, "values": [ { "objectId": 3326, "instanceId": 0, "resourceId": 5700, "datatype": "Float", "value": 11.2 } ] }`
Conductivity	3327	`{ "timestamp": 0, "values": [ { "objectId": 3327, "instanceId": 0, "resourceId": 5700, "datatype": "Float", "value": 11.2 } ] }`
Power	3328	`{ "timestamp": 0, "values": [ { "objectId": 3328, "instanceId": 0, "resourceId": 5700, "datatype": "Float", "value": 11.2 } ] }`
Power Factor	3329	`{ "timestamp": 0, "values": [ { "objectId": 3329, "instanceId": 0, "resourceId": 5700, "datatype": "Float", "value": 11.2 } ] }`
Distance	3330	`{ "timestamp": 0, "values": [ { "objectId": 3330, "instanceId": 0, "resourceId": 5700, "datatype": "Float", "value": 11.2 } ] }`
Energy	3331	`{ "timestamp": 0, "values": [ { "objectId": 3331, "instanceId": 0, "resourceId": 5805, "datatype": "Float", "value": 11.2 } ] }`
Direction	3332	`{ "timestamp": 0, "values": [ { "objectId": 3332, "instanceId": 0, "resourceId": 5705, "datatype": "Float", "value": 11.2 } ] }`

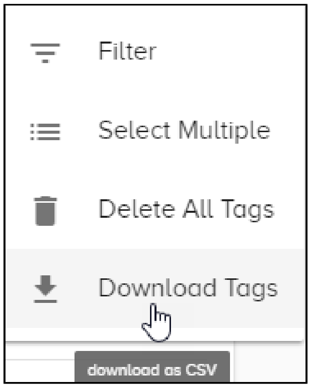
Time	3333	{ "timestamp": 0, "values": [{ "objectId": 3333, "instanceId": 0, "resourceId": 5506, "datatype": "Float", "value": 11.2 }] }
Gyrometer	3334	{ "timestamp": 0, "values": [{ "objectId": 3334, "instanceId": 0, "resourceId": 5702, "datatype": "Float", "value": 12.3 }, { "objectId": 3334, "instanceId": 0, "resourceId": 5703, "datatype": "Float", "value": 123.4 }, { "objectId": 3334, "instanceId": 0, "resourceId": 5704, "datatype": "Float", "value": 45.2 }] }
Color	3335	{ "timestamp": 0, "values": [{ "objectId": 3335, "instanceId": 0, "resourceId": 5706, "datatype": "String", "value": "RED" }] }
GPS Lodation	3336	{ "timestamp": 0, "values": [{ "objectId": 3336, "instanceId": 0, "resourceId": 5514, "datatype": "String", "value": "Latitude" }, { "objectId": 3336, "instanceId": 0, "resourceId": 5515, "datatype": "String", "value": "Longitude" }] }
Positioner	3337	{ "timestamp": 0, "values": [{ "objectId": 3337, "instanceId": 0, "resourceId": 5536, "datatype": "Float", "value": 12.3 }, { "objectId": 3336, "instanceId": 0, "resourceId": 5537, "datatype": "Float", "value": 2.234 }] }
Buzzer	3338	{ "timestamp": 0, "values": [{ "objectId": 3338, "instanceId": 0, "resourceId": 5850, "datatype": "Boolean", "value": 1 }] }
Audio Clip	3339	{ "timestamp": 0, "values": [{ "objectId": 3339, "instanceId": 0, "resourceId": 5522, "datatype": "Opaque", "value": "clip" }] }
Timer	3340	{ "timestamp": 0, "values": [{ "objectId": 3340, "instanceId": 0, "resourceId": 5521, "datatype": "Float", "value": 10.23 }] }
Addressable Text Display	3341	{ "timestamp": 0, "values": [{ "objectId": 3341, "instanceId": 0, "resourceId": 5527, "datatype": "String", "value": "Text" }] }
On/Off Switch	3342	{ "timestamp": 0, "values": [{ "objectId": 3342, "instanceId": 0, "resourceId": 5500, "datatype": "Boolean", "value": 0 }] }
Lever Control	3343	{ "timestamp": 0, "values": [{ "objectId": 3343, "instanceId": 0, "resourceId": 5851, "datatype": "Float", "value": 0.23 }] }
Up/Down Control	3344	{ "timestamp": 0, "values": [{ "objectId": 3344, "instanceId": 0, "resourceId": 5532, "datatype": "Boolean", "value": 0 }] }
Multiple Axis Joystick	3345	{ "timestamp": 0, "values": [{ "objectId": 3345, "instanceId": 0, "resourceId": 5500, "datatype": "Boolean", "value": 0 }] }
Rate	3346	{ "timestamp": 0, "values": [{ "objectId": 3346, "instanceId": 0, "resourceId": 5700, "datatype": "Float", "value": 0.12 }] }
Push Button	3347	{ "timestamp": 0, "values": [{ "objectId": 3347, "instanceId": 0, "resourceId": 5500, "datatype": "Boolean", "value": 0 }] }
Multistate Selector	3348	{ "timestamp": 0, "values": [{ "objectId": 3348, "instanceId": 0, "resourceId": 5547, "datatype": "Integer", "value": 12 }] }

How to Manage Tags in DeviceHub

A large number of tags can be especially challenging to manage. To facilitate actions on tags, the following options are available:

- Filter Tags
- Delete All Tags
- Remove or Edit a Single Tag
- Select Multiple Tags to Remove
- Download Tags

- See also, DeviceHub Bulk Upload Tags and DeviceHub Add a Tag.

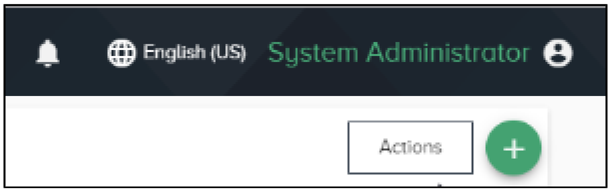


Filter Tags

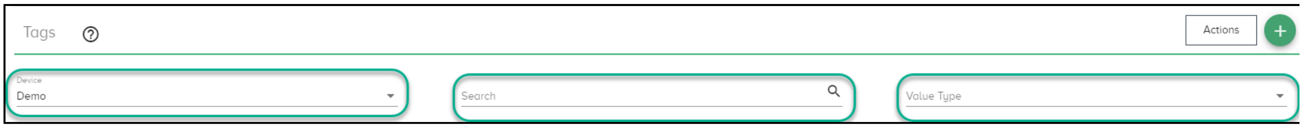
With hundreds to thousands of tags for devices, a filter is required to isolate a single tag or a group of tags.

To filter a list of tags:

1. In the Voice of the Machine Edge navigation panel, go to **DeviceHub > Tags**.
2. Click **Actions** at the top right of the window.



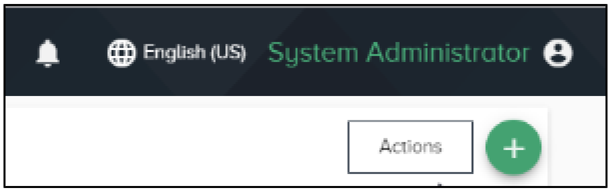
3. Select **Filter** from the Actions list to display the filter selectors and search box.



Delete All Tags

To delete <u>all</u> tags:

1. In the Voice of the Machine Edge navigation panel, go to **DeviceHub > Tags**.
2. Click **Actions** at the top right of the window.

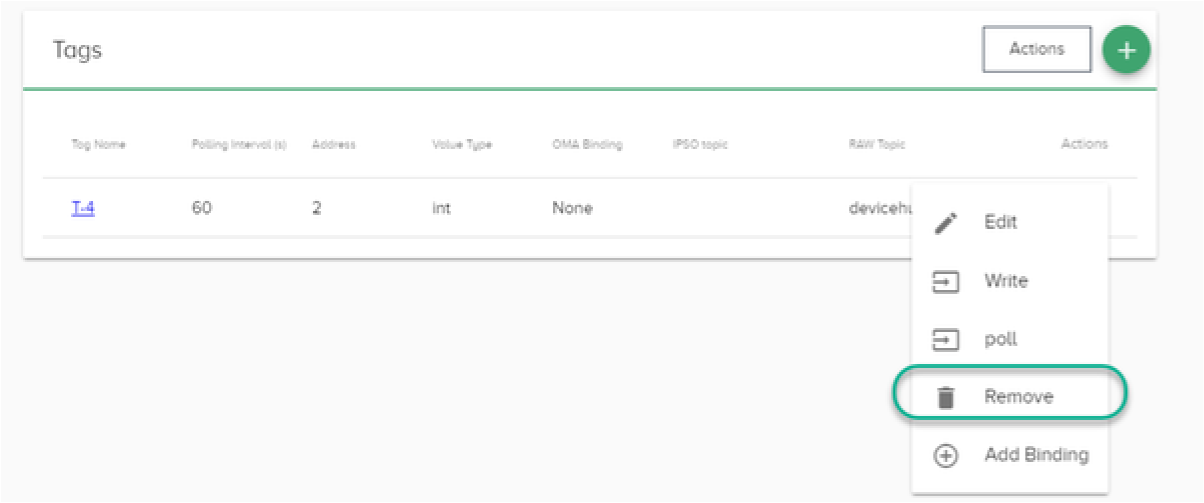


3. Select **Delete All Tags** from the Actions list.
   A pop-up window asks you to confirm this action.

Remove or Edit a Single Tag

To delete a single tag:

1. In the Voice of the Machine Edge navigation panel, go to **DeviceHub > Tags**.
2. Select **Remove** from the tag's Action list.
   You will be prompted to confirm the tag's deletion.

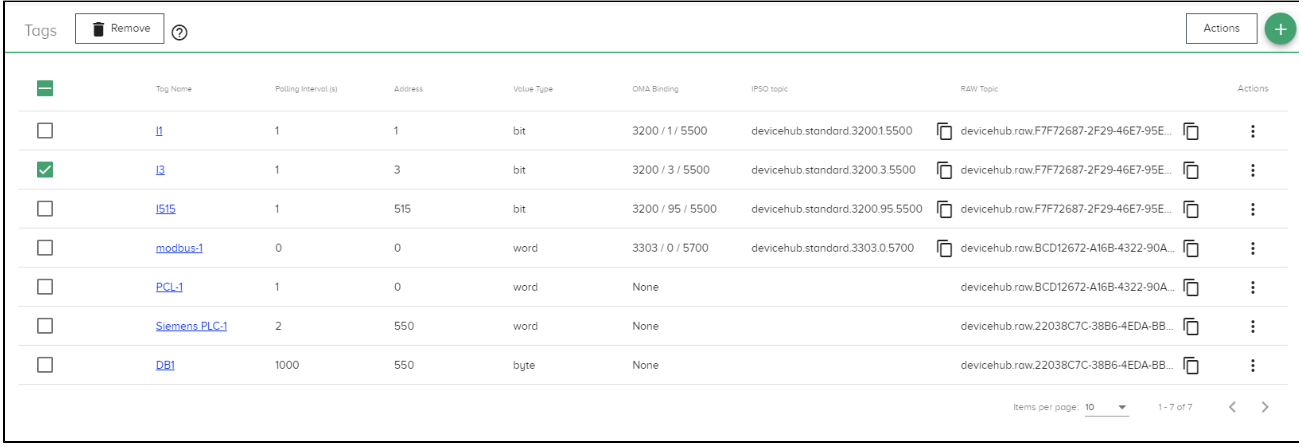


Tag Name	Polling Interval (s)	Address	Value Type	OMA Binding	IPSO topic	RAW Topic	Actions
T-4	60	2	int	None		devicehu	

Select Multiple Tags to Remove

To select specific tags for deletion:

1. In the Voice of the Machine Edge navigation panel, go to **DeviceHub > Tags**.
2. Click **Actions** at the top right of the window.
3. Select **Select Multiple** from the list to display the tags prefaced with checkboxes.
4. Check the tags to be removed. This action enables the **Remove** button at the top of the Tags list.



▬	Tag Name	Polling Interval (s)	Address	Value Type	OMA Binding	IPSO topic	RAW Topic	Actions
□	I1	1	1	bit	3200 / 1 / 5500	devicehub.standard.32001.5500	devicehub.raw.F7F72687-2F29-46E7-95E...	⋮
☑	I3	1	3	bit	3200 / 3 / 5500	devicehub.standard.3200.3.5500	devicehub.raw.F7F72687-2F29-46E7-95E...	⋮
□	I515	1	515	bit	3200 / 95 / 5500	devicehub.standard.3200.95.5500	devicehub.raw.F7F72687-2F29-46E7-95E...	⋮
□	modbus-1	0	0	word	3303 / 0 / 5700	devicehub.standard.3303.0.5700	devicehub.raw.BCD12672-A16B-4322-90A...	⋮
□	PCL-1	1	0	word	None		devicehub.raw.BCD12672-A16B-4322-90A...	⋮
□	Siemens PLC-1	2	550	word	None		devicehub.raw.22038C7C-38B6-4EDA-BB...	⋮
□	DB1	1000	550	byte	None		devicehub.raw.22038C7C-38B6-4EDA-BB...	⋮

Items per page: 10 1 - 7 of 7

5. Click **Remove** at the top of the Tags list.

A pop-up window prompts you to confirm this action.

Download Tags

To download tags to a CSV file:

1. In the Voice of the Machine Edge navigation panel, go to **DeviceHub > Tags**.
2. Click **Actions** at the top right of the window.
3. Select **Download Tags**.

A CSV file is created and downloaded. The filename has the following date-specific format: `tag-YYYY-MM-DD.csv`.

How to Bulk Upload Tags in DeviceHub

Typically, in a large environment, data needs to be collected from hundreds of device registers. In this case, adding tags one by one is a daunting task. Voice of the Machine Edge DeviceHub makes that task easier with a bulk load option, which creates tags from an uploaded comma-separated-values (CSV) file. This attribute values in the CSV file can be derived from DeviceHub Devices and from register definitions supplied by PLC vendors.

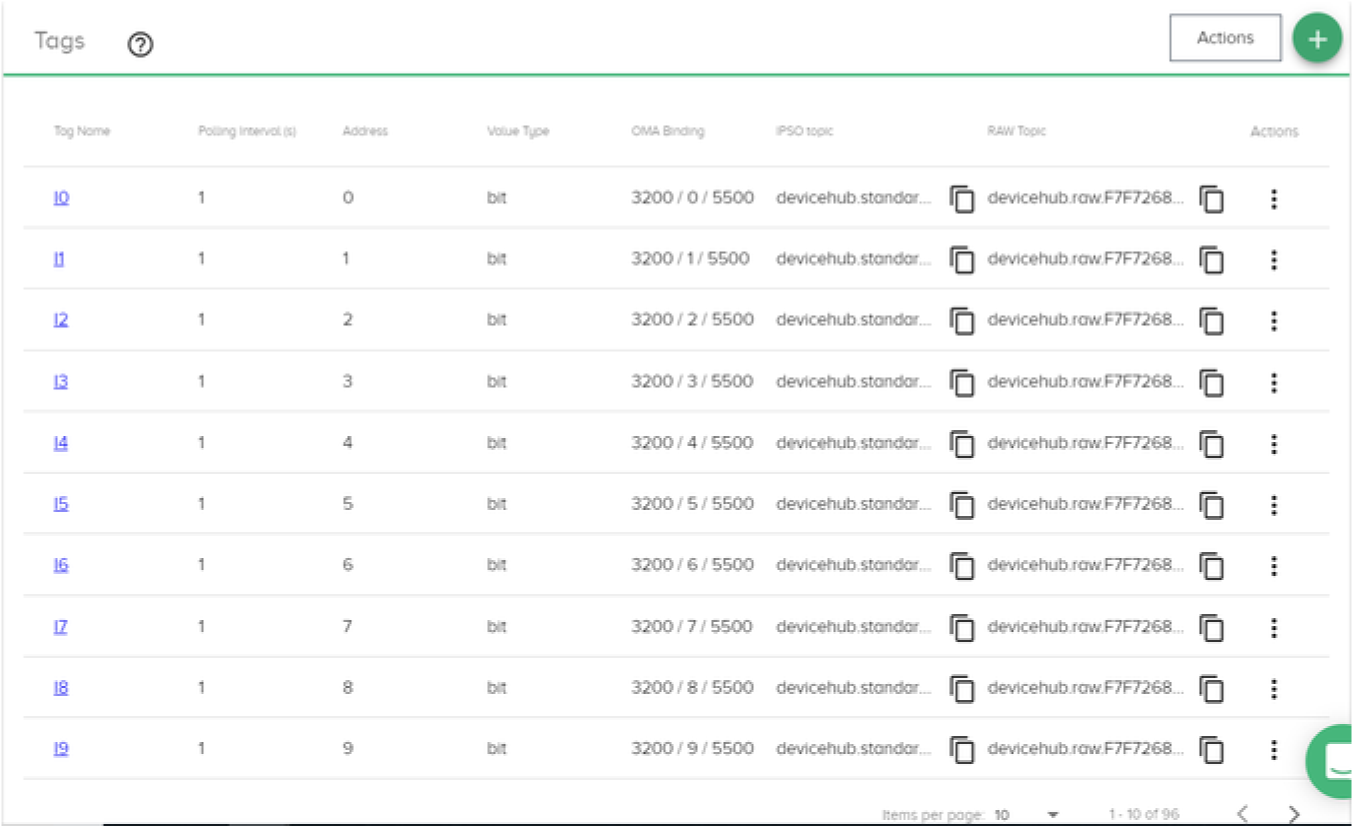
Use the following information to complete the bulk-loading task:

For additional tag-management tasks, see DeviceHub Manage Tags.

Overview of DeviceHub Tag Bulk Loading

Prerequisite
A DeviceHub device must be configured for the specific device and driver, for example, Allen-Bradley Micrologix. See DeviceHub Add, Monitor, or Delete Device.

The goal of this bulk-loading operation is to add a large set of tags/registers to Voice of the Machine Edge, as illustrated in the following DeviceHub Tags list. To add just a single tag, see DeviceHub Add a Tag.



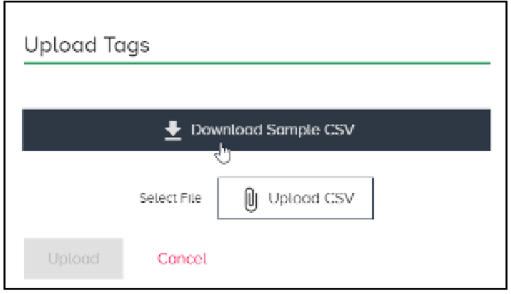
Tag Name	Polling Interval (s)	Address	Value Type	OMA Binding	IPSO topic	RAW Topic	Actions
I0	1	0	bit	3200 / 0 / 5500	devicehub.standar...	devicehub.raw.F7F7268...	⋮
I1	1	1	bit	3200 / 1 / 5500	devicehub.standar...	devicehub.raw.F7F7268...	⋮
I2	1	2	bit	3200 / 2 / 5500	devicehub.standar...	devicehub.raw.F7F7268...	⋮
I3	1	3	bit	3200 / 3 / 5500	devicehub.standar...	devicehub.raw.F7F7268...	⋮
I4	1	4	bit	3200 / 4 / 5500	devicehub.standar...	devicehub.raw.F7F7268...	⋮
I5	1	5	bit	3200 / 5 / 5500	devicehub.standar...	devicehub.raw.F7F7268...	⋮
I6	1	6	bit	3200 / 6 / 5500	devicehub.standar...	devicehub.raw.F7F7268...	⋮
I7	1	7	bit	3200 / 7 / 5500	devicehub.standar...	devicehub.raw.F7F7268...	⋮
I8	1	8	bit	3200 / 8 / 5500	devicehub.standar...	devicehub.raw.F7F7268...	⋮
I9	1	9	bit	3200 / 9 / 5500	devicehub.standar...	devicehub.raw.F7F7268...	⋮

Items per page: 10 1 - 10 of 96

Download the Sample CSV File Template

Voice of the Machine Edge provides a template of the required fields.

1. Go to **DeviceHub > Tags**.
2. Click the ⊕ at the right of the window.
3. Select **Upload CSV** from the list.
4. Click **Download Sample CSV**.



A `tag-template.csv` file downloads.

5. Populate the CSV File.

Populate the CSV File

A row/record for each tag must be populated with attribute values, as described below. Refer to the PLC's register definitions for details.

1. Using the downloaded template, add a row for each tag to be added.
2. Keep the header row as defined in the template:

deviceId	deviceName	driverName	registerName	tagName	description	valueType	address	dbNumber	pollinginterval	omaObjectId	omaInstanceId	omaResourceId	omaValueType

Backward compatibility for previous CSV versions
If you are using a CSV file from a previous Voice of the Machine Edge version:

- Check the downloaded CSV template to verify that all columns are present in your CSV file. Additional columns may have been added or columns may have been renamed.
- The order of the attribute columns in the latest CSV template may be different from previous versions. The columns can be in any order.

Example of a Tag CSV File

deviceId	deviceName	driverName	registerName	tagName	description	valueType	address	dbNumber	pollinginterval	omaObjectI
0b55610e-d501-4b68-832c-ca392aa20faa	Siemens-S7-D1	siemens_snap7	C5:0.PRE	D1000		bit	10	1	1	3303
	Siemens-S7-D1	siemens_snap7	C5:0.PRE	D1001		bit	10	2	2	3303

Definition of Tag Attributes

- See CSV Formatting Rules.
- For OMA object and resource details, refer to the device vendor documentation, as well as this reference page: IPSO Object Reference Guide.

Tag Attribute Rules

- Either `deviceId` or `deviceName` can be omitted, but not both. One of these fields is required to find the corresponding device.
  - If a device identified by the `deviceId` is *not* found, the entire tag record is rejected.
  - If a device identified by the `deviceName` is *not* found, this is *not* an error and the tag record is added.
- Attribute values are required unless noted as optional.
  - If an optional value is null, it is ignored. This allows for the loading of register definitions for multiple device types from a single file.
  - If the value is null and the device driver defines it as mandatory, the tag will not be imported.
  - If an optional value is present in the CSV file, but it is not supported by the device driver, it is treated as an error and the tag will not be imported.
- Attribute columns in the CSV can be in any order.

Attribute Name	Data Type	Sample Value	Description	Notes

`deviceId`	string	`"0b55610e-d501-4b68-832c-ca392aa20faa"`	ID of the device (optional)	Click the pages ⧉ icon next to the very long device ID in the DeviceHub device card to copy the device ID. If a device identified by the `deviceId` is *not* found, `deviceName` is used.
`deviceName`	string	`"Siemens-S7-D01773"`	Name of the device (used if deviceId is not provided)	Copy the user-defined name for the device from the DeviceHub device card. If a device identified by the `deviceName` is *not* found, this is *not* an error and the tag record is added.
`driverName`	string	`"siemens_snap7"`	Driver name is for validating optional attributes against *registration.json*. This type should directly correspond to the name of the driver package.	Click the pages ⧉ icon next to the driver name in the Device card to copy the driver name. Using this mandatory driver name, along with the device name, enables a user to replicate the device with the same tags over multiple Voice of the Machine Edge systems.
`registerName`	string	`"C5:0.PRE"`	PLC register name or full PLC register address	This name is case sensitive.
`tagName`	string	"D1000"	Tag name that will be displayed in DeviceHub	Often this is a combination of the register and address
`description`	string		Any text (optional)	An *optional* user-supplied description for each tag
`omaObjectId`	integer	`3303`	OMA Object ID (optional)	The numeric ID for the OMA object. Example for a temperature sensor object: 3303 All four OMA values either must be present or all four must be empty. When OMA values are present, OMA binding is applied to the tag.
`omaInstanceId`	integer	`0`	OMA Instance ID (optional)	All four OMA values either must be present or all four must be empty. When OMA values are present, OMA binding is applied to the tag.
`omaResourceId`	integer	`5602`	OMA Resource ID (optional)	Example for the maximum measured value of a temperature sensor object: 5602 All four OMA values either must be present or all four must be empty. When OMA values are present, OMA binding is applied to the tag.
`omaValueType`	string, one of: • String • Integer • Float • Boolean • Opaque • Time • Objlink	`"Boolean"`	OMA value type (optional)	All four OMA values either must be present or all four must be empty. When OMA values are present, OMA binding is applied to the tag.
`valueType`	string	`"bit"`	Type of the register value	Examples: bit, word

`address`	integer	`10`	Register address	Refer to the PLC vendor's guide for a list of valid addresses.
`dbNumber`	integer	`1`	Siemens dbNumber (optional)	This is a device-specific attribute. Refer to the PLC vendor's guide for a valid list.

CSV Formatting and Escaping Rules

Because CSV format is not standardized, several different methods could be used for escaping special characters. And, depending on locality, a different value separator may be used. For example, in European countries, a semi-colon is widely used as a separator. The Voice of the Machine Edge CSV specification adheres to the rules used by Microsoft Excel, as listed below.

- Field separators can be comma (,), semicolon (;), or tab (\t).
- Fields can have white spaces around them. This space is removed during processing.
- Fields can be escaped with double quotes (") or single quotes ('). Leading or trailing white spaces inside the quotes are preserved.
- Any separator characters within a quoted field is treated as part of the field value.
- Any quotation character in a field value must be escaped with a backslash (\ ) character.
- Fields are parsed based on their data type: **string** values are read "as is" and **numeric** values are parsed with a numeric parser.
- Empty fields are treated as null values.

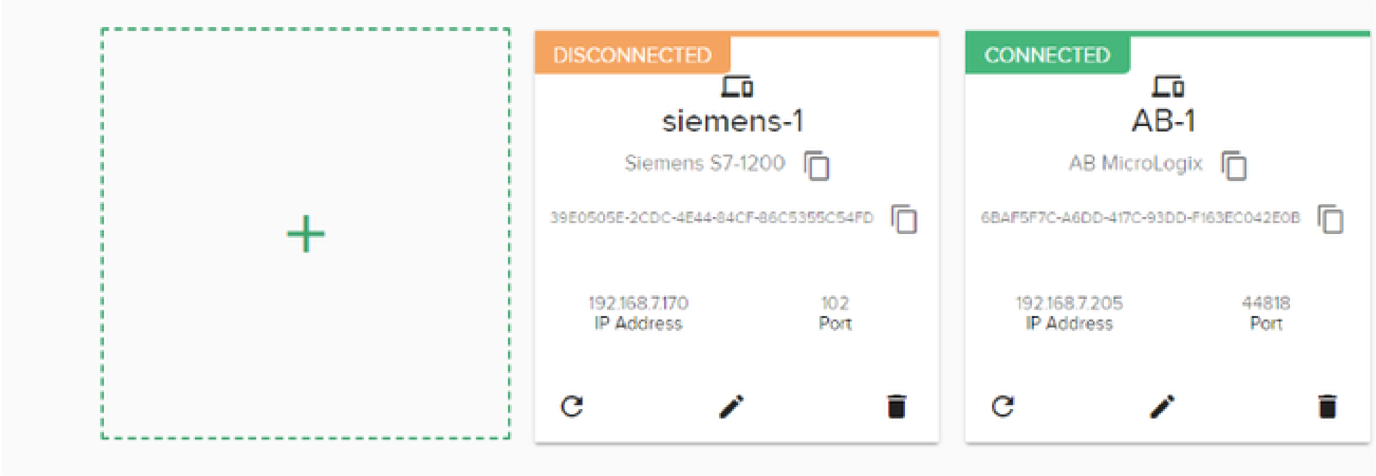
Upload the CSV File into DeviceHub

1. Go to **DeviceHub > Tags**.
2. Click the ⊕ at the upper right of the Tags window.
3. Select **Upload CSV**.
4. Select the file and click **Upload**.

Upon a successful upload, the tags will appear in the Tags window. If the details in the CSV file do not match the expected configuration, error messages will indicate the issues.

How to Troubleshoot DeviceHub Connectivity Status

At times, a device may display as *Disconnected*, as in the following example.



DISCONNECTED
siemens-1
Siemens S7-1200
39E0505E-2CDC-4E44-84CF-86C5355C54FD
192.168.7.170
IP Address
102
Port

CONNECTED
AB-1
AB MicroLogix
6BAF5F7C-A6DD-417C-93DD-F163EC042E0B
192.168.7.205
IP Address
44818
Port

Why is the Device Disconnected?

Certain conditions cause a PLC to be disconnected in Voice of the Machine Edge.

Condition	**Reasons PLC Appears as Disconnected**	**Steps to Connect the Device**
Free Tag: Explicit register name required	• Certain device drivers, such as Allen-Bradley CompactLogix, will *not* have a list of register names in the DeviceHub Add Tag form. • When a list of registers is not provided, the case-sensitive register name must be entered exactly as it is defined in the PLC. • In these cases, the device will not display a *Conn ected* status until a tag with a correct register name is created.	1. See DeviceHub Add a Tag and DeviceHub Manage Tags to add a tag with a case-sensitive register name.
HID Device in Wait State	• Some HID (Human Interface Devices), such as barcode scanners, are always disconnected and only show a connected status when they send data to Voice of the Machine Edge.	1. This is a temporary state and no intervention is required. The device will connect when there is data to receive.
Incorrect PLC Configuration	• If configuration changes have been made to a PLC, the best way to ensure accurate Voice of the Machine Edge status is to delete and recreate the DeviceHub Device.	1. Delete the DeviceHub Device. See DeviceHub Add, Monitor, or Delete Device. 2. Create a new DeviceHub Device with the correct PLC configuration.
Incorrect Tag Information	• To prevent the PLC from going into fail safe mode or from crashing the process when a tag has incorrect information, Voice of the Machine Edge disconnects communication with the PLC. • For example, a tag may contain an incorrect address.	1. Monitor all the tags in Flows to identify the bad tag. 2. Find the tag with a success flag set to *false*. 3. Delete the tag from DeviceHub. See DeviceHub Manage Tags. 4. If there is no data on the flows, restart the PLC. If a node in a flow indicates "waiting for messages" and nothing is displayed in the debug console, this means that Voice of the Machine Edge has stopped communicating with the PLC, so there is no data.

How to Navigate System Panel in Voice of the Machine Edge

In the navigation panel, click **System** to access the list of system features

System Feature	Find more information here...
Info	System Info
Certificates	Certificates
Network	System Network
Device Management	Device Management
Users	User Management
FTP	FTP Server
License	License Management
Status	Status
Backup/Restore	Backup/Restore

How to Interact with Systems Info

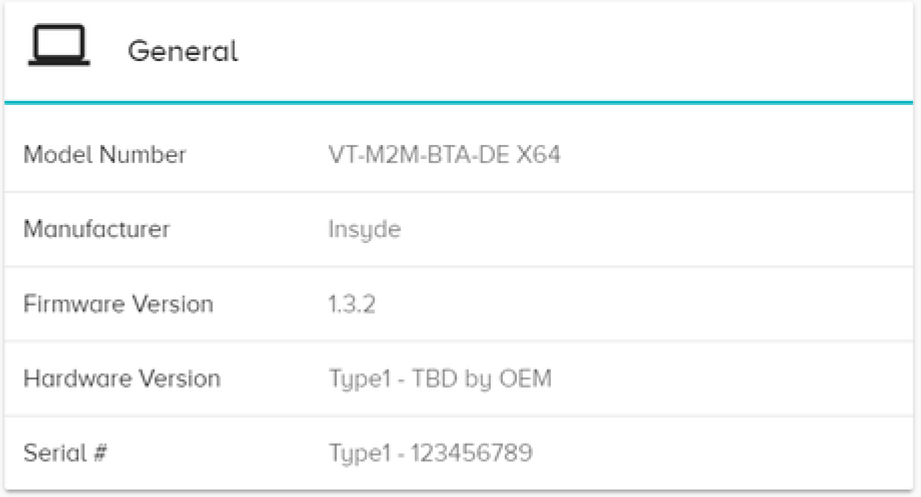
Info

The Info tab of the system as the name suggests gives the details about the device. What is the general information, what is the CPU statistics, the RAM and the memory size. It also gives the detail about the serial and the network interfaces on the device.

- General
- CPU
- Disk
- Memory
- Network Interfaces
- Serial Interfaces

General

The General tab provides general information about the gateway, This includes **Model Number, Manufacturer, Firmware Version, Serial #.**

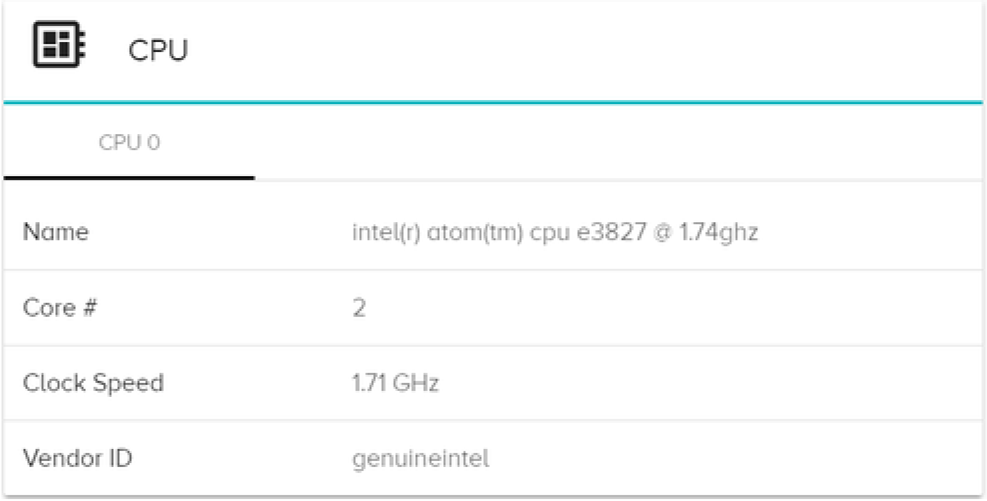


General

Model Number	VT-M2M-BTA-DE X64
Manufacturer	Insyde
Firmware Version	1.3.2
Hardware Version	Type1 - TBD by OEM
Serial #	Type1 - 123456789

CPU

Th CPU tab gives the information about the CPU which includes, **Name, Number of Cores (Core#),Clock Speed, Vendor ID.**

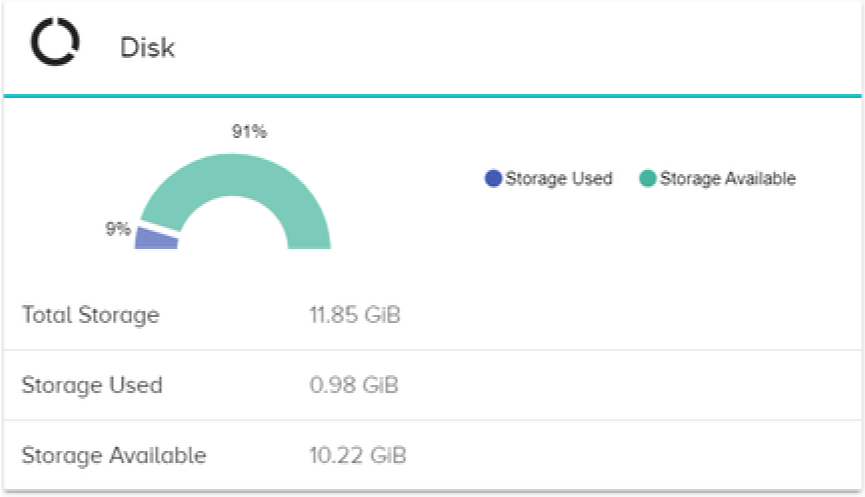


CPU

Name	intel(r) atom(tm) cpu e3827 @ 1.74ghz
Core #	2
Clock Speed	1.71 GHz
Vendor ID	genuineintel

Disk

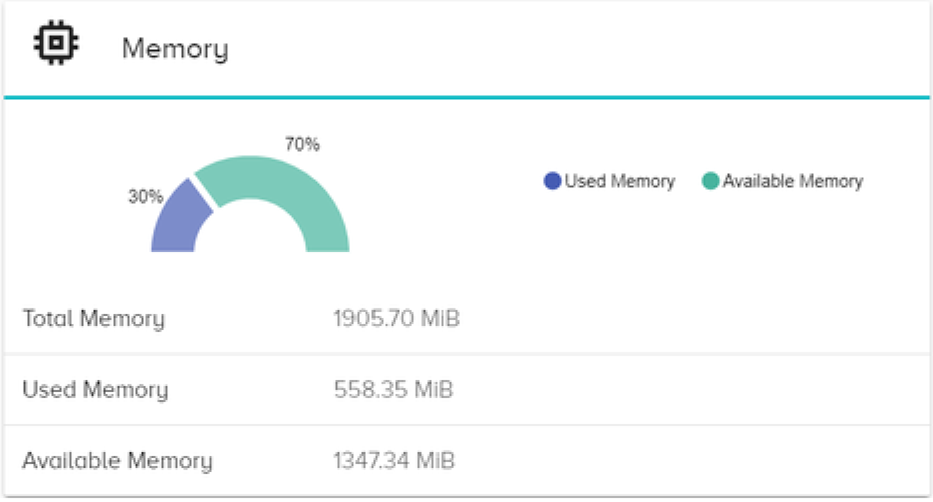
The Disk tab gives us the information about the Disk size of the system.



Total Storage	11.85 GiB
Storage Used	0.98 GiB
Storage Available	10.22 GiB

Memory

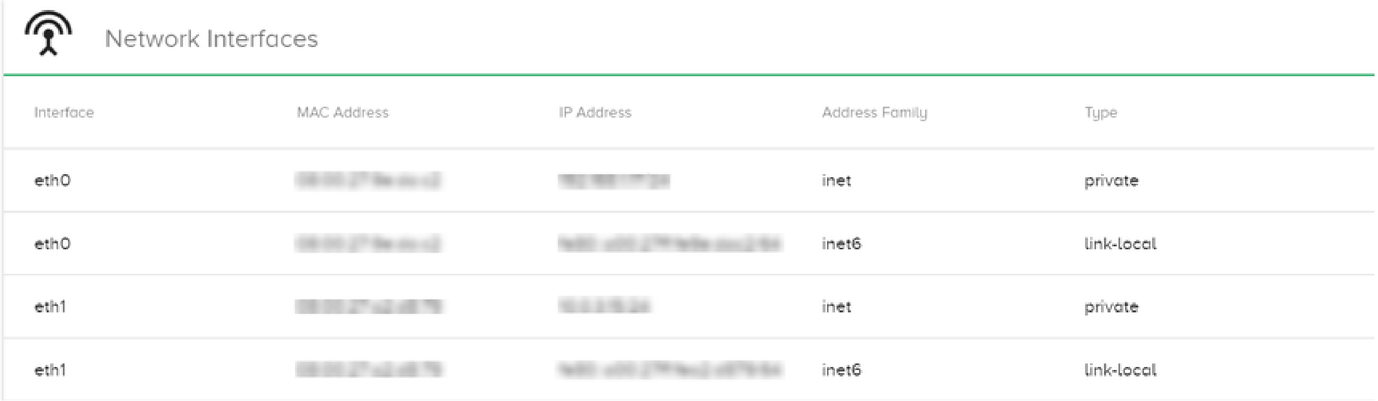
The Memory tab is the details about the memory on the system.



Total Memory	1905.70 MiB
Used Memory	558.35 MiB
Available Memory	1347.34 MiB

Network Interfaces

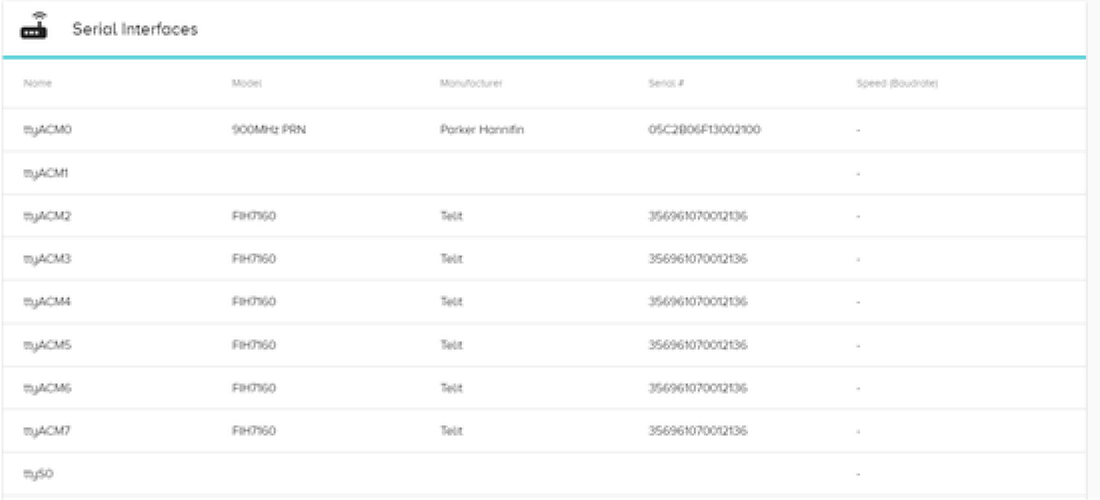
The Network tab gives the information about the network interfaces such as Ethernet, WAN etc. that the device supports.



Interface	MAC Address	IP Address	Address Family	Type
eth0	[redacted]	[redacted]	inet	private
eth0	[redacted]	[redacted]	inet6	link-local
eth1	[redacted]	[redacted]	inet	private
eth1	[redacted]	[redacted]	inet6	link-local

Serial Interfaces

The serial tab gives the information about the serial interfaces present on the hardware.



Serial Interfaces

Name	Model	Manufacturer	Serial #	Speed (Baudrate)
ttyACM0	900MHz PRN	Parker Hannifin	05C2B06F13002100	-
ttyACM1				-
ttyACM2	FIH7160	Telit	356961070012136	-
ttyACM3	FIH7160	Telit	356961070012136	-
ttyACM4	FIH7160	Telit	356961070012136	-
ttyACM5	FIH7160	Telit	356961070012136	-
ttyACM6	FIH7160	Telit	356961070012136	-
ttyACM7	FIH7160	Telit	356961070012136	-
ttyS0				-

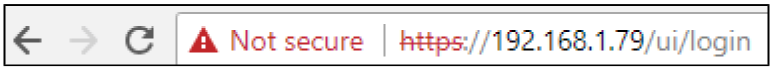
Related Articles

- How to Activate an Offline License Key
- How to Use Terminal User Interface with Voice of the Machine Edge
- Voice of the Machine Edge Starter Kit Guide
- How to Update Sensor Rate
- How to Add a New/Blank Sensor

How to Use Certificates

The Certificates form is used to store certificates locally on the Voice of the Machine Edge device. The user can install or uninstall certificates from this form.

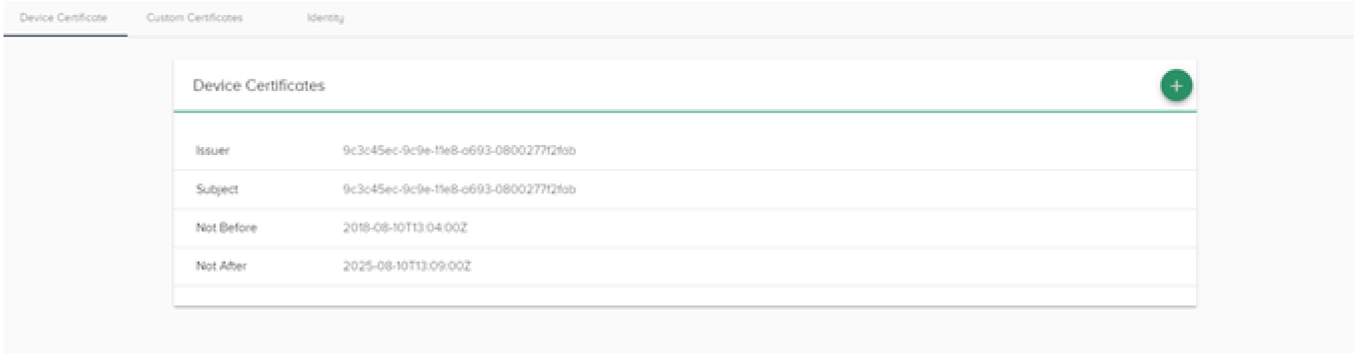
Voice of the Machine Edge includes a default SSL certificate and therefore all communication is secure by default. A certificate is required only if you want to give the device a DNS name and remove the browser's security warning.



Not secure | ~~https~~://192.168.1.79/ui/login

To add a device certificate:

1. In the navigation panel, expand System and click **Certificates**.
2. Select the **Device Certificate** tab.
3. Click ⊕.



Device Certificates

Issuer	9c3c45ec-9c9e-11e8-a693-0800277f2fab
Subject	9c3c45ec-9c9e-11e8-a693-0800277f2fab
Not Before	2018-08-10T13:04:00Z
Not After	2025-08-10T13:09:00Z

When adding a certificate, the following parameters are required:

- **SSL Certificate**– After submitting a certificate signing request (CSR) to a certificate authority, you will receive both a certificate and a key. Enter the certificate received from the certificate authority.
- **CA Chain**– Certificate Authority (CA) Chain – This is the chain of SSL certificates from the root to the end-user certificate, including intermediate certificates.
- **Private Key**– After submitting a certificate signing request (CSR) to a certificate authority, you will receive both a certificate and a key. Enter the key received from the certificate authority.

Related articles

- How to Activate an Offline License Key
- How to Use Terminal User Interface with Voice of the Machine Edge
- Voice of the Machine Edge Starter Kit Guide
- How to Update Sensor Rate
- How to Add a New/Blank Sensor

How to use Networks

A Voice of the Machine device will have two Ethernet interfaces:

- eth0 always will be connected to the internet and located in the DMZ (demilitarized zone). The DMZ is a separate network that has restricted access to a company's internal network.
- eth1 will be connected to a private network, with an operational DHCP server on this network.

Modify Network Info

The following network settings can be modified in Voice of the Machine Edge:

- **Friendly Name**: Enter or revise the descriptive name and click **Change** to apply the change.
- **Country**: Modify the text and click **Change** to apply the change.
- **Timezone**: Select from the drop-down list.
- **Fallback DNS Resolvers**: Click **Add** to add an IPv4 address (n.n.n.n). Listed DNS resolvers can be deleted by clicking the trash can icon.
- **NTP Servers**:
    - Expand the NTP Servers list.
    - Enter the **DNS Name** of an NTP server.
    - Click **Add** to add a Network Time Protocol server to the pool of NTP servers.

- Listed NTP servers can be deleted by clicking the trash can icon.

Modify Network Interface Settings

View and modify the Voice of the Machine Edge network gateway configuration.

- Modifications to these network settings are required only when you want to change to a static address.
- The eth0 interface typically is all that is required.
- The eth0 interface must be used for first-time bootup for DNS resolution.
- The WAN checkbox must be enabled in order to be able to configure the network gateway address.

1. In the navigation panel, expand **System**.
2. Select **Network**.
3. Select an interface at the left and modify any of the following settings:
    - **WAN**- Enable or disable the WAN connection.
    - **IPv4 Type**- The IPv4 address is assigned by DHCP, but it can be changed to Static.
    - **IPv6 Type**- The IPv6 address type defaults to Auto, but it can be changed to DHCP or Static.

When changing a Type to Static, a **CIDR** (Classless Inter-Domain Routing) address and a network **Gateway** address must be entered. The gateway address depends on how the router is configured.

Example of an IPv4 CIDR:

```
193.168.1.66/63
```

Click **Update** to apply the changes.

After changing the network configuration, the device needs to go through a power cycle and restart.

Related articles

- How to Activate an Offline License Key
- How to Use Terminal User Interface with Voice of the Machine Edge
- Voice of the Machine Edge Starter Kit Guide
- How to Update Sensor Rate
- How to Add a New/Blank Sensor

How to Use Device Management

Device Management

Use Device Management features to manage Voice of the Machine Edge functions integrated with Voice of the Machine Cloud.

Device management is used for:

- Activating a Voice of the Machine Edge device for remote access in the cloud.
- Getting updates using LWM2M (Lightweight Machine to Machine).
- Managing the Voice of the Machine Edge device, such as rebooting the device or executing a factory reset.

Status

Once a Voice of the Machine Edge device has been cloud-activated, it can be accessed remotely from anywhere in the world.

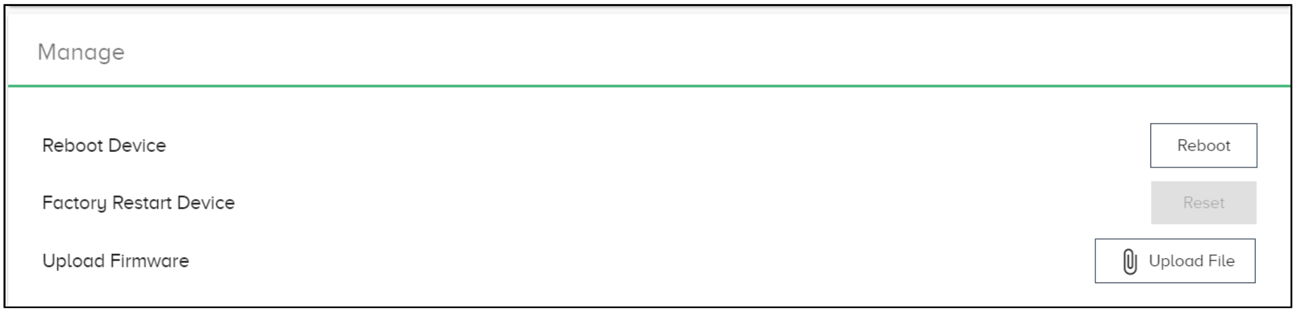
The status of the device's connection includes the last activity, last status change, last message, last status, and status code.

Status Code	Meaning
0	Active
10	Connecting
20	Not Configured

Manage

The following device management functions can be executed on the Voice of the Machine Edge system.

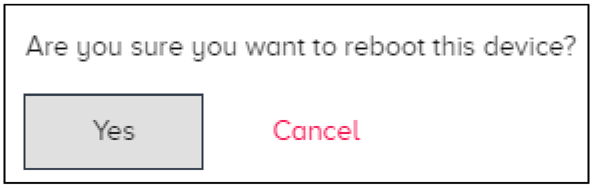
- Reboot
- Revert to factory settings
- Upload firmware



Reboot Voice of the Machine Edge Device

To reboot the device:

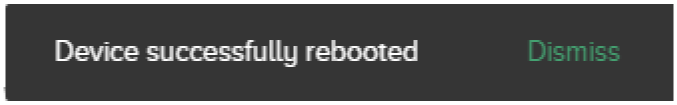
1. In the navigation panel, select: **System > Device Management**.
2. Click **Reboot** and you will be prompted with a confirmation pop-up:


Are you sure you want to reboot this device?

3. A spinning process symbol indicates that the reboot is in progress.

Rebooting Device . . . .

4. A status message at the bottom of the browser window will indicate when the process is complete.


Device successfully rebooted

Factory Restart Device

As the label implies, all Voice of the Machine Edge configurations will be removed and Voice of the Machine Edge will return to its original state, before any configurations were created.

Please contact Parker before using this feature.

Related articles

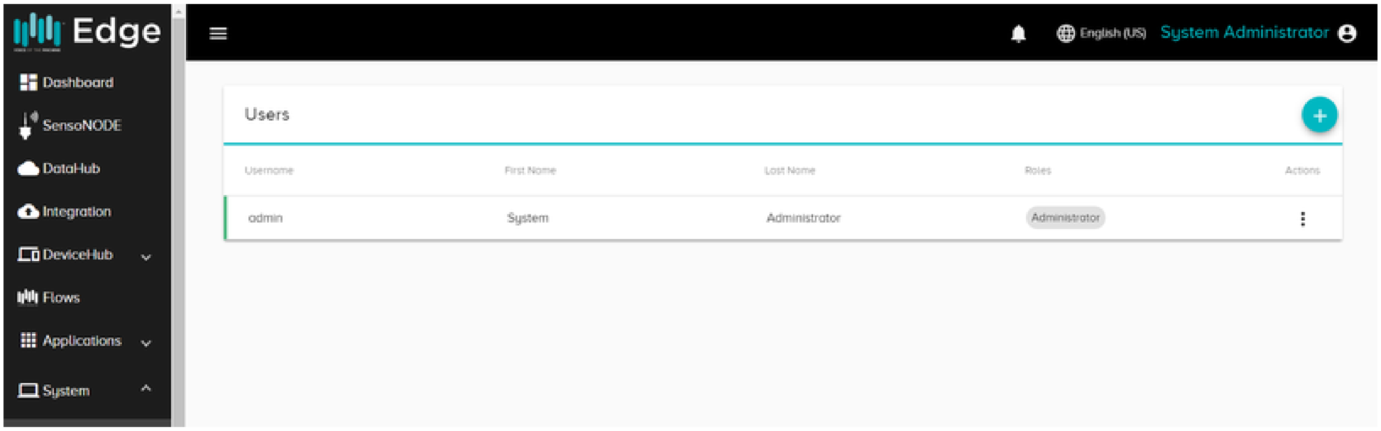
- How to Activate an Offline License Key
- How to Use Terminal User Interface with Voice of the Machine Edge
- Voice of the Machine Edge Starter Kit Guide
- How to Update Sensor Rate
- How to Add a New/Blank Sensor

How to Manage Users

User Management

Voice of the Machine Edge has three levels of user authorization based on the type of user role selected. Each Voice of the Machine Edge feature/component has independent authorization and its own scope of privileges. See User Roles and Privileges.

- Observer
- Developer
- Administrator



Users

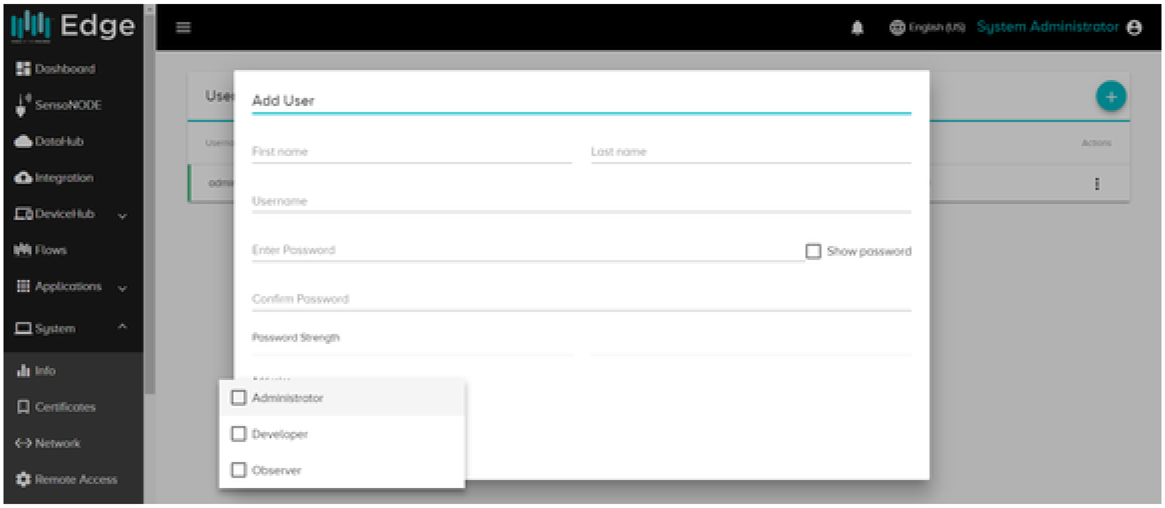
Username	First Name	Last Name	Roles	Actions
admin	System	Administrator	Administrator	⋮

Add a Voice of the Machine Edge User

- Once a user is added, the Administrator must email the credentials to the user.
- In the interest of security, the password that is created when adding a user is a temporary password, which must be changed by the user at the initial login.

To add a Voice of the Machine Edge user:

1. In the navigation panel, select: **System > Users**.
2. Click the ⊕ icon in the upper right corner.
3. In the Add User dialog, enter the name, username, password, and roles and then click **Add User**.



Password Requirements

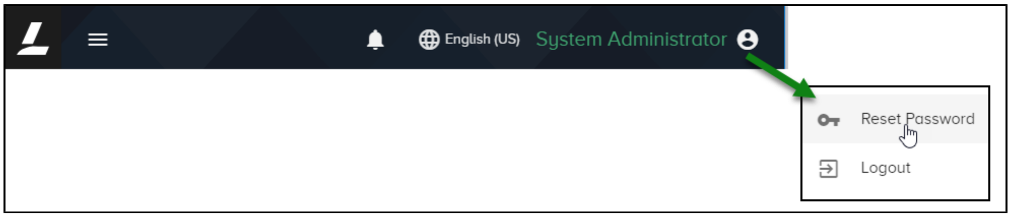
The password requires a minimum of three bars, which indicate that the password complies with the following rules:

- Contains at least **1** capital letter.
- Contains at least **1** number.

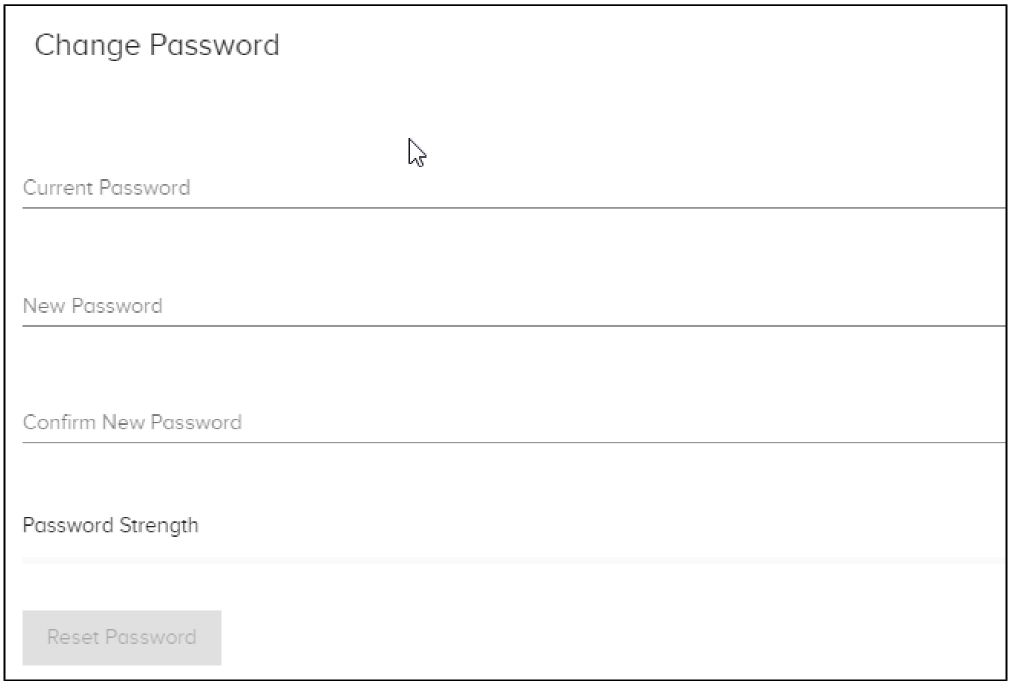
- Should be at least **7** characters long.

Reset User Password

Users can reset their passwords at any time.



Enter a new password according to the Password Requirements and click **Reset Password**.



Edit, Remove, Disable/Enable a User

To change a Voice of the Machine Edge user's credentials or role or to disable/enable a user:

1. In the navigation panel, select: **System > Users**.
2. In the Actions list to the right of the user, select **Edit**,**Remove**, or **Disable/Enable**.
    a. The **Remove** action prompts for confirmation before deleting the user.
    b. **Disable/Enable** is a toggle action.
    c. **Edit** enables changes to all user fields.

User Roles and Privileges

This table lists the various privileges based on the defined user role. Note that a user can be assigned more than one role.

Note: Backup/Restore requires an additional license.

	Role		
Feature/Task	**Observer**	**Developer**	**Administrator**
Authentication			

License Administration			✓
License: view only	✓	✓	✓
User Password Updates	✓	✓	✓
User Profile Updates	✓	✓	✓
User Management • manage users • activate/deactivate license • configure authentication providers			✓
Device Management			
View Device Info	✓	✓	✓
Manage Device • configure interfaces • perform actions, such as reboot and upgrade			✓
DataHub			
View cloud connectors and nodes	✓	✓	✓
Create, update, delete nodes		✓	
Administer DataHub • create/update/delete cloud connectors • subscribe to a raw data socket			✓
Cloud Connectors			
List/view cloud connectors and nodes	✓	✓	✓
Create/update/delete nodes		✓	
Create/update/delete cloud connectors • view protected cloud connector data • subscribe to a raw data socket			✓
DeviceHub			
View drivers, devices, registers, OMA mappings	✓	✓	✓
Create devices, tags, OMA mappings		✓	
Create, update, delete devices			✓
Marketplace			
List marketplaces and view marketplace apps	✓	✓	✓
List applications and view details Start and stop applications		✓	✓
Administer Marketplace • add/remove marketplaces • configure and launch apps from marketplace			✓
Flows			
View flows	✓	✓	✓
Create, modify, and save flows		✓	
Start and stop flows			✓
FTP			

View FTP service status and list users	✓	✓	✓
Administer FTP • start/stop ftp • manage ftp users			✓

Related articles

- How to Activate an Offline License Key
- How to Use Terminal User Interface with Voice of the Machine Edge
- Voice of the Machine Edge Starter Kit Guide
- How to Update Sensor Rate
- How to Add a New/Blank Sensor

How to Use FTP Server with Voice of the Machine Edge

FTP Server

The File Transfer Protocol (FTP) is a standard network protocol used to transfer files between a client and server on a computer network.

- Voice of the Machine Edge supports FTP, which allows devices to securely exchange files with the Voice of the Machine Edge device. Voice of the Machine Edge will be the server in this exchange and the client will be running on another device that is on the same network as the Voice of the Machine Edge device.
- FTP can be used to transfer data into a .csv file or similar formats. Likewise, it also can be used to read a file from the drive and send data to the cloud.

To configure the Voice of the Machine Edge FTP service:

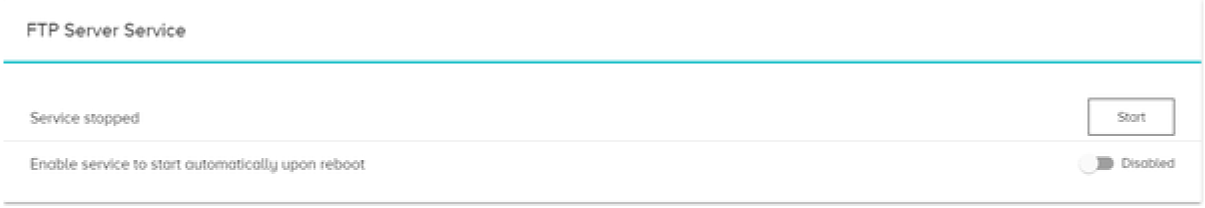
- Add and Manage an FTP User
- Start the FTP Server
- Configure the FTP Client
- FTP a File to Voice of the Machine Edge
- Use Voice of the Machine Edge as an FTP Client

Add and Manage an FTP User

Add one or more users to get the usernames and passwords needed to connect to the FTP server.

To add and manage a user:

1. In the navigation panel, select:**System > FTP Server**.

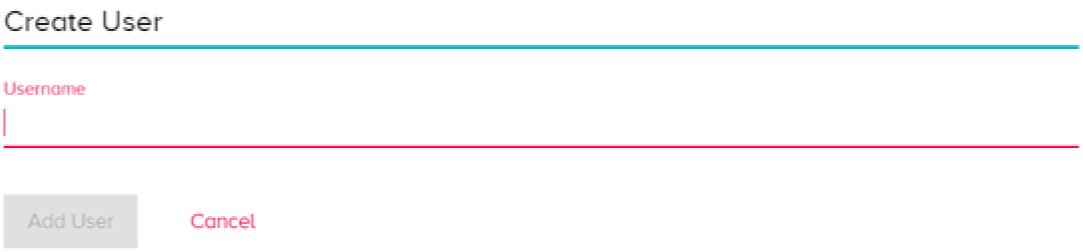


FTP Server Service

Service stopped

Enable service to start automatically upon reboot — Disabled

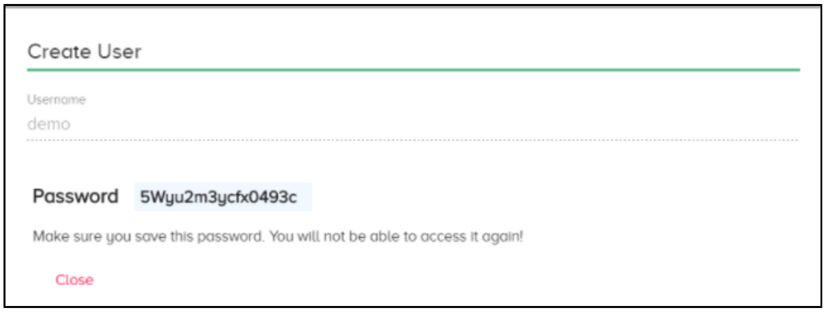
2. In the Users section of the page, click the ⊕ icon to define a username for the FTP server.



Create User

Username

3. In the Create User dialog, enter a **Username** and click **Add User**.
   - Be sure to copy the password that is displayed in the Create User dialog because, for security reasons, it will not be accessible again.
   - The password can be reset later by selecting **Reset Password** from the Actions list next to the username.

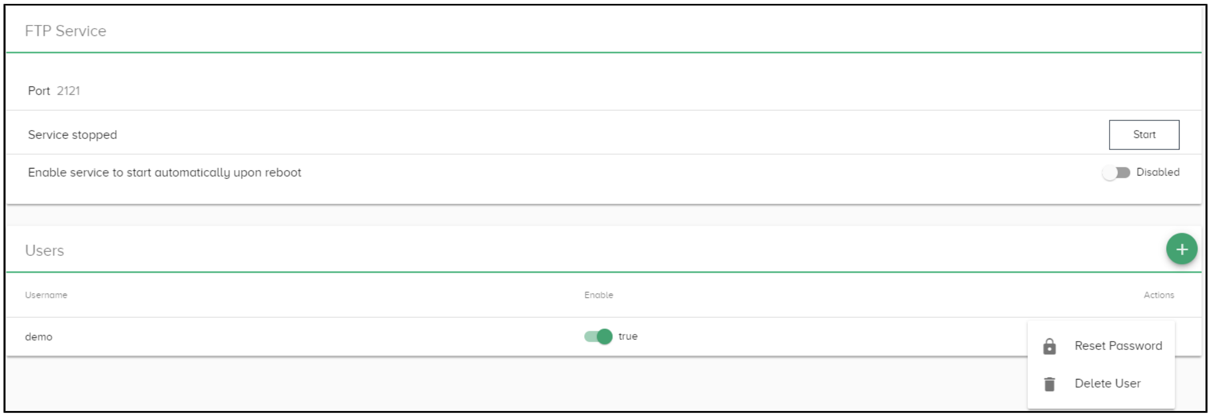


Create User

Username
demo

Password 5Wyu2m3ycfx0493c

Make sure you save this password. You will not be able to access it again!



FTP Service

Port 2121

Service stopped

Enable service to start automatically upon reboot — Disabled

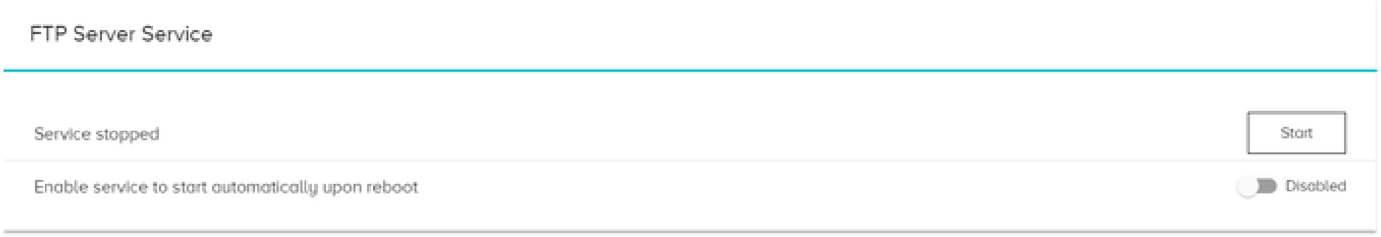
Users

Username	Enable	Actions
demo	true	

4. **Enabled/disable** a user with the toggle switch, next to a user's name.
5. **Reset a user's password** by selecting the option from the Actions list, next to a user's name.
6. **Delete a user** by selecting the option from the Actions list, next to a user's name.

Start the FTP Server

1. In the FTP Service section of the **System > FTP Server** page, click **Start**.
2. Toggle the switch to enable the service to start automatically when Voice of the Machine Edge is rebooted.



FTP Server Service

Service stopped

Enable service to start automatically upon reboot — Disabled

Configure the FTP Client

The FTP client must be running on a device that is on the same network as the Voice of the Machine Edge device.

Use the following details to configure the FTP client, to establish a connection to the Voice of the Machine Edge server.

- **Host**: IP address of the Voice of the Machine Edge device.
- **Username**: The Voice of the Machine Edge FTP username defined in the above Add and Manage an FTP User procedure.
- **Password**: The password that was generated when the Voice of the Machine Edge FTP user was created in the Add and Manage an FTP User procedure.
- **Port**: The default Voice of the Machine Edge FTP port is **21**.

FTP a File to Voice of the Machine Edge

In the FTP client application, enter the credentials listed in Configure the FTP Client and then transfer a file to the Voice of the Machine Edge device.

- The file that is transferred to the Voice of the Machine Edge device is saved in a secure directory.
- The file can be accessed using:`/<filename>`

Use Voice of the Machine Edge as an FTP Client

If you are using Voice of the Machine Edge as a client of FTP, there is no need to configure a Voice of the Machine Edge FTP Server. For an example of how to use Voice of the Machine Edge as an FTP client.

Related articles

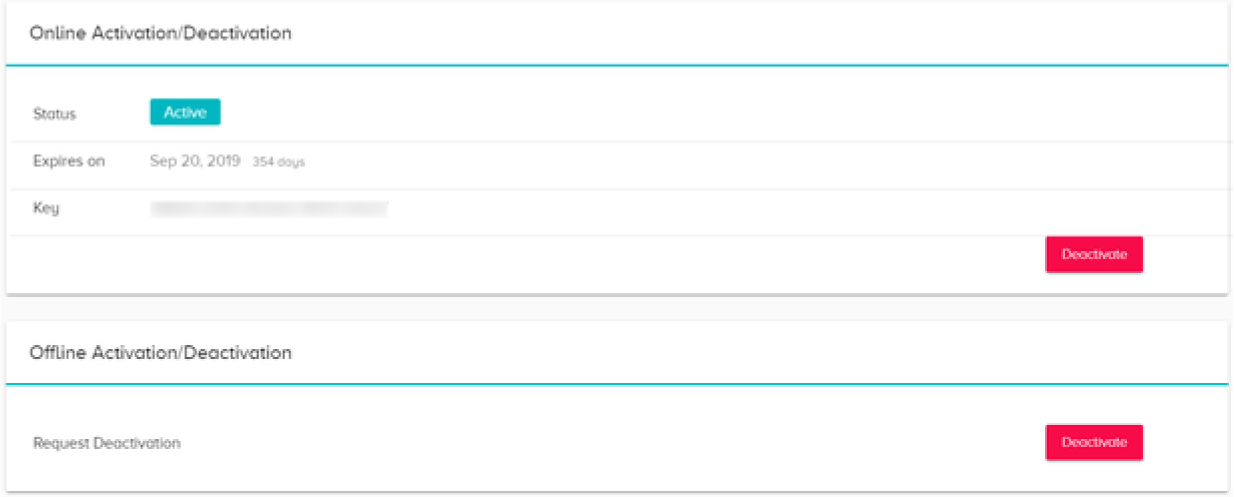
- How to Activate an Offline License Key
- How to Use Terminal User Interface with Voice of the Machine Edge
- Voice of the Machine Edge Starter Kit Guide
- How to Update Sensor Rate
- How to Add a New/Blank Sensor

How to Manage Licenses

License

> **A Voice of the Machine Edge license is initially valid for 15 days for trial purposes. The user is required to contact your Parker representative to obtain an official and full license.**

The key goes to this page and the device is activated. If the device is going offline, or not to be used, then the same license key can be deactivated and installed in the new device. During the trial period, the user may receive a message indicating your license expiry. This may occur every few seconds. The license page also indicates the date until which the license is valid, allowing the user time to obtain the license before expiration. The license can be activated online as well as offline. The same key works if the device is online or offline.



Online Activation/Deactivation

Status Active

Expires on Sep 20, 2019 354 days

Key [redacted]

Offline Activation/Deactivation

Request Deactivation

Related articles

- How to Activate an Offline License Key
- How to Use Terminal User Interface with Voice of the Machine Edge
- Voice of the Machine Edge Starter Kit Guide
- How to Update Sensor Rate
- How to Add a New/Blank Sensor

How to Activate an Offline License Key

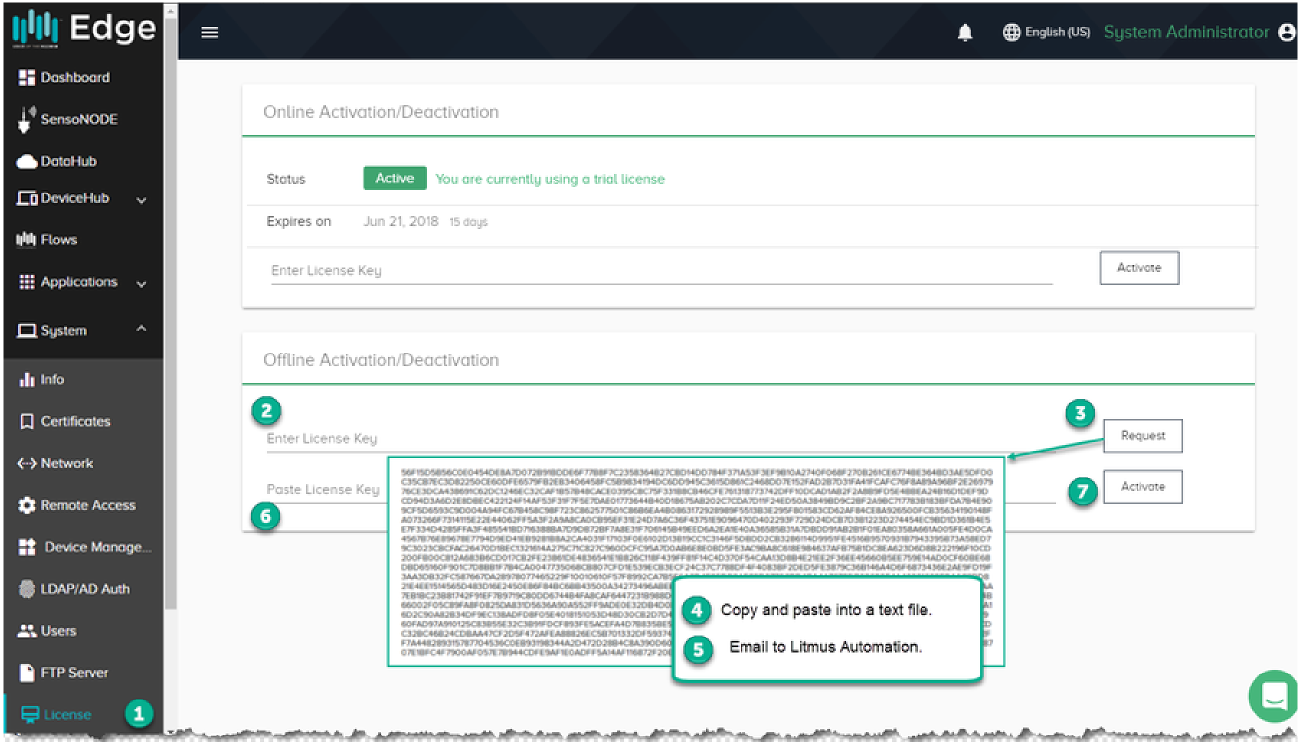
Offline License Activation Overview

- Voice of the Machine Edge does not require Ethernet or an internet connection during installation. The installation works completely offline. However, Ethernet (not internet) is required during first bootup, after the installation is complete.
- Voice of the Machine Edge license activation can be performed without being connected to the internet.

To activate a Voice of the Machine Edge license, without an internet connection:

Parker will send a license key, similar to the following: `86D32-7BADC-D7A97-7944A-2946D`

1. In the navigation panel, select **System > License**.
2. In the Offline Activation/Deactivation section of the window, enter the license key.
3. Click **Request**.
   A pop-up will fill with a large amount of text.



Status: Active — You are currently using a trial license

Expires on: Jun 21, 2018 15 days

4 Copy and paste into a text file.

5 Email to Litmus Automation.

4. Copy and paste the text into a text file.
5. Email the text file to Parker.
6. When you receive a response to the email, paste the license key into the field.
7. Click **Activate**.

Related articles

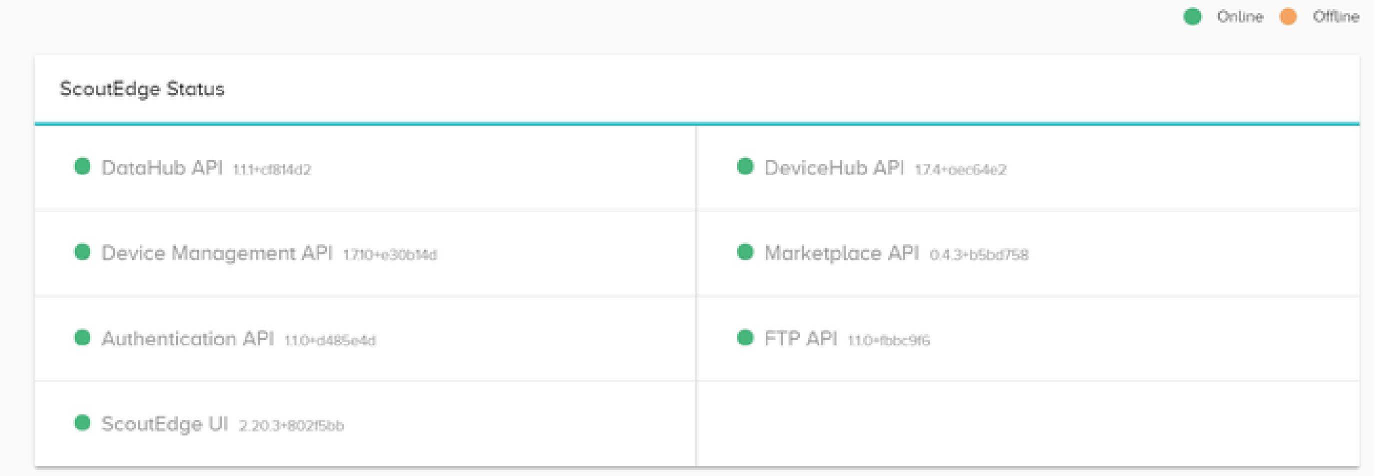
- How to Activate an Offline License Key
- How to Use Terminal User Interface with Voice of the Machine Edge
- Voice of the Machine Edge Starter Kit Guide
- How to Get Started with Voice of the Machine Cloud
- How to Update Sensor Rate

How to View Status

Status

The status page is used primarily by Customer Support when troubleshooting an issue. It shows the status and version of the underlying software components.

To access the Status page, go to the navigation panel and select **System > Status**.



● Online ● Offline

ScoutEdge Status

● DataHub API 1.1.1+cf814d2	● DeviceHub API 1.7.4+aec64e2
● Device Management API 1.7.10+e30b14d	● Marketplace API 0.4.3+b5bd758
● Authentication API 1.1.0+d485e4d	● FTP API 1.1.0+fbbc9f6
● ScoutEdge UI 2.20.3+802f5bb	

Related articles

- How to Activate an Offline License Key
- How to Use Terminal User Interface with Voice of the Machine Edge
- Voice of the Machine Edge Starter Kit Guide
- How to Update Sensor Rate
- How to Add a New/Blank Sensor

How to Backup and Restore

Backup and Restore

As with any system, backup and restore tasks become part of an organization's standard operating procedures. Because a device can have hardware failures, a disaster recovery plan needs to be in place to support device replacement.

The Voice of the Machine Edge Backup/Restore feature supports the following use cases:

- User deleted or overwrote a portion of the Voice of the Machine Edge configuration and wants to return to the last-known good configuration.
- Device hardware fails and the device must been replaced with another one.

Note: Backup/Restore is not intended for replication of configurations.

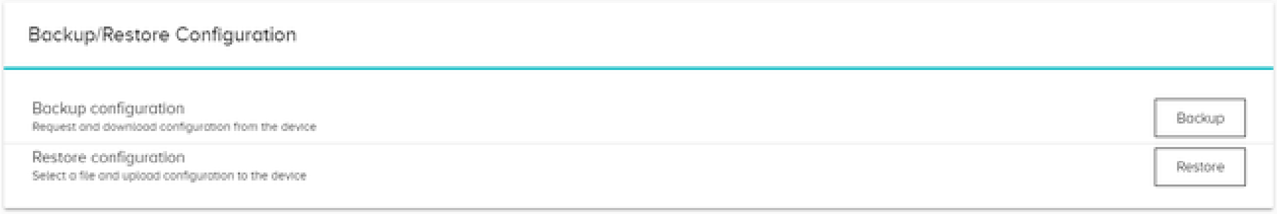
Check Backup/Restore Prerequisites

Prerequisites

- If you have an older Voice of the Machine Edge version, the Backup/Restore feature may not be available.
- The minimum Voice of the Machine Edge version that includes the Backup/Restore feature is: 1.1.0.
- Your organization's standards and requirements dictate the frequency of device backups.

Obtain a Backup/Restore License

This Backup/Restore feature requires a license upgrade, as indicated when accessing the Backup/Restore page: **System > Backup/Restore**



Backup/Restore Configuration

Backup configuration
Request and download configuration from the device

Restore configuration
Select a file and upload configuration to the device

Request a License Key

- Contact your Parker representative to request a license key.
- Once an additional license key has been received, go to **System > License** to enter the license key and activate the license.

What gets backed up?

- configurations
- collected data

Where is the backup stored?

Voice of the Machine Edge configurations and data are saved in a JSON file (`backup.json`), which by default, is downloaded to the computer from which you are accessing Voice of the Machine Edge. On a Windows computer, this file can be found in the `Downloads` folder.

Backup File Characteristics

The lightweight backup file:

- Makes efficient use of storage resources.
- Uses hashing algorithms to secure credentials.

Best Practices

The following recommendations help to ensure seamless data recovery:

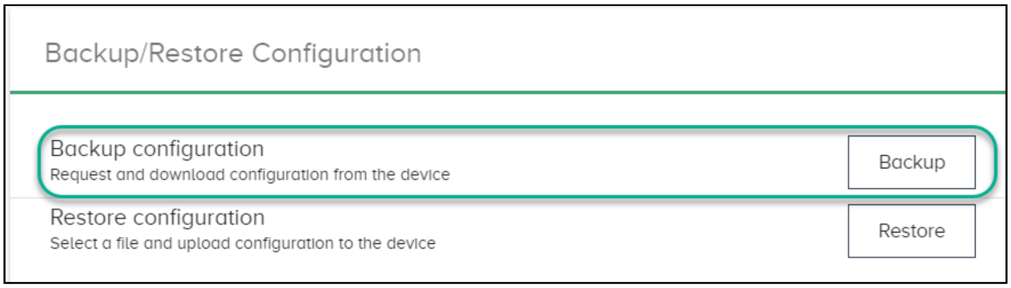
- Rename the `backup.json` file, based on a naming convention that will help identify the date the backup was taken. Example: `06-18-2018-VoiceoftheMachineEdge-backup.json`

- Move the JSON file from the default `Downloads` directory to an operational directory on a server that can be accessed by the operational team.

Steps to back up a Voice of the Machine Edge device

Use these steps to back up a Voice of the Machine Edge device:

1. **System > Backup/Restore**
2. Click **Backup**.



Backup/Restore Configuration

Backup configuration
Request and download configuration from the device

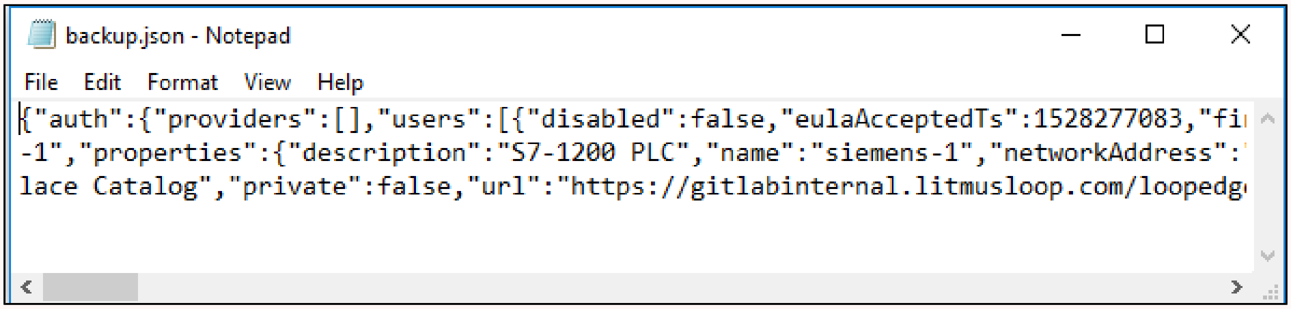
Restore configuration
Select a file and upload configuration to the device

3. A `backup.json` file will be saved to the `Downloads` folder.

How will I know the backup was successful?

The `backup.json`file will open in Notepad.

- This json file contains the Voice of the Machine Edge configurations, as well as, the data.
- Use this file to restore a Voice of the Machine Edge device.



```
{"auth":{"providers":[],"users":[{"disabled":false,"eulaAcceptedTs":1528277083,"fi
-1","properties":{"description":"S7-1200 PLC","name":"siemens-1","networkAddress":
lace Catalog","private":false,"url":"https://gitlabinternal.litmusloop.com/loopedg
```

Modifications to the contents of this file may render it unusable for restoring a Voice of the Machine Edge device.

Restore a Voice of the Machine Edge Device

Prerequisite

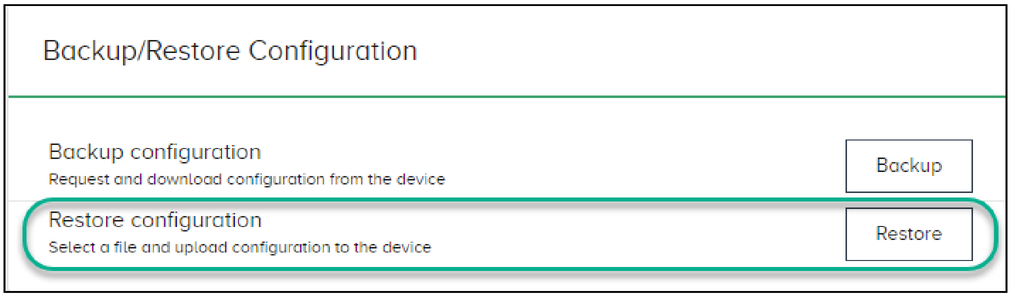
- Regularly back up Voice of the Machine Edge.
- The backup can only be restored to the same device or to a replacement device and it only can be restored to a device with the same license key.

Steps to restore a Voice of the Machine Edge device

The Restore operation overwrites all configurations and data collected since the last backup.

Use these steps to restore a Voice of the Machine Edge device:

1. **System > Backup/Restore**
2. Click **Restore**.



Backup/Restore Configuration

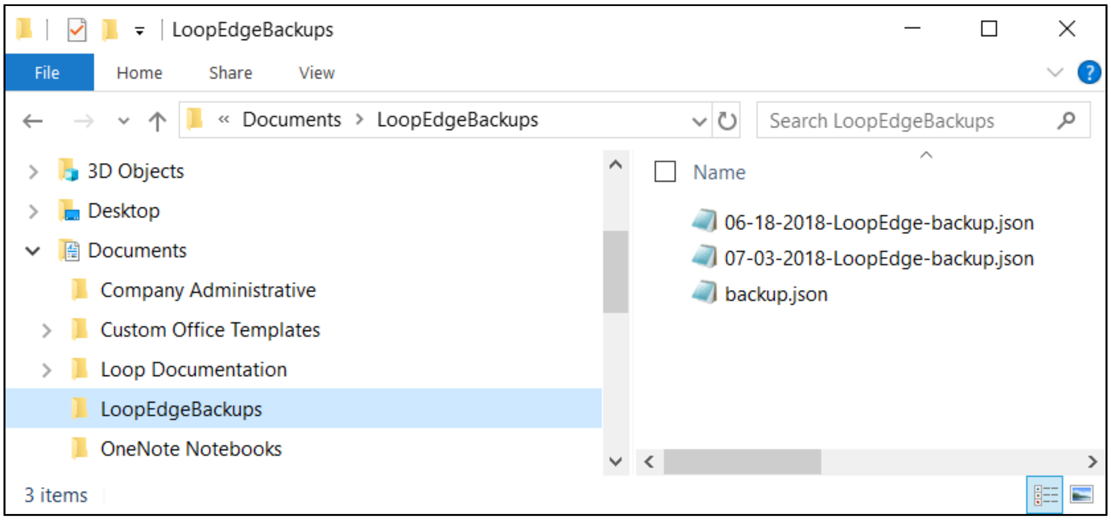
Backup configuration

Request and download configuration from the device

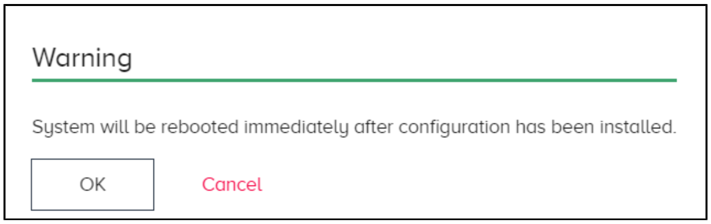
Restore configuration

Select a file and upload configuration to the device

3. In the File Explorer window, select the backup file and click **Open**.



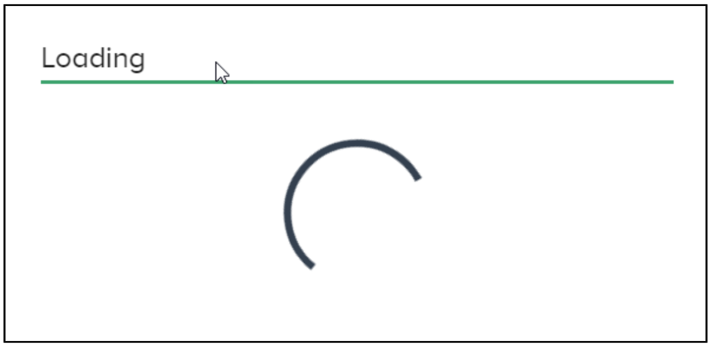
4. A warning message indicates that the system will reboot after the restore has completed.



Warning

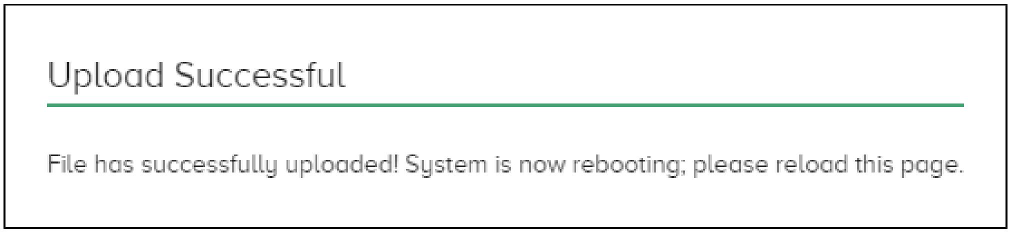
System will be rebooted immediately after configuration has been installed.

5. The Loading icon will spin until the restore is complete.



Once restored, the Voice of the Machine Edge device will restart with the configuration that was backed up.

How will I know the restore was successful?



Upload Successful

File has successfully uploaded! System is now rebooting; please reload this page.

Replace a Damaged Device

Whenever a gateway device needs to be replaced, several steps must be taken to get the new device configured and online, with minimal interruption.

Check Device Replacement Prerequisites

Prerequisites

- Regularly back up Voice of the Machine Edge.
- The replacement device must be on the same network as the device that was backed up. If the backed up device has a static IP address assigned to it, then the replacement device will get that same IP address. For example, a replacement device has an IP address of 10.0.0.120 and the backed up gateway device is on a network with an IP address of 192.168.1.1. After Voice of the Machine Edge restoration, the replacement device will no longer be accessible with the 10.0.0.120 address because it will now assume the 192.168.1.1 address.
- The backup can only be restored to the same device or to a replacement device and it only can be restored to a device with the same license key.

Decommission a Voice of the Machine Edge Device

Before physically removing a device from operations, the following steps need to be taken to logically deprovision it.

- In Voice of the Machine Cloud, set the device status to **Maintenance**.
  This action takes the Voice of the Machine Edge device offline until a new device is deployed.

Migrate to a New Voice of the Machine Edge Device

Once a new device is ready for deployment, the following steps need to be taken to deploy the new device and migrate the data from the old device to the new device.

1. Add the new device to the network.
2. Go to **System > License** to enter the license key and activate the license.
3. Restore the configurations and data from the last backup.
4. Activate the Voice of the Machine Cloud connection by setting the device status to **Online**.
   This action registers the new device with Voice of the Machine Cloud and the corresponding LWM2M and MQTT protocols.

Related articles

- How to Activate an Offline License Key
- How to Use Terminal User Interface with Voice of the Machine Edge
- Voice of the Machine Edge Starter Kit Guide
- How to Update Sensor Rate
- How to Add a New/Blank Sensor

All FAQ about Voice of the Machine Edge?

What is Voice of the Machine Edge?

- Voice of the Machine Edge is an operating system on an Industrial PC (IPC) appliance. This edge-level software seamlessly collects data from IoT devices/systems (such as sensors and PLCs) and integrates the data into the cloud or into on-premise enterprise systems. Extensive knowledge of 35+ device drivers, both legacy and newer systems, enables Voice of the Machine Edge to manage the variety of devices in an industrial environment. Voice of the Machine Edge software provides support for many protocols/buses (RS232, RS485, CAN, Ethernet, Fieldbus, etc.) via an IPC appliance that is connected to the Internet. The raw or processed data can be visualized and analyzed at the edge or sent to LoopCloud, near real time.

How do I access Voice of the Machine Edge?

- Access Voice of the Machine Edge software by connecting via a browser on the same network as the Voice of the Machine Edge device.
- Enter the IP address into the browser URL, for example: https://192.168.1.49
- Refer to How to Get Started with Voice of the Machine Edge

What is the function of DataHub?

- DataHub enables *northbound* cloud connectivity. Use DataHub to send edge-level data securely to LoopCloud.
- DataHub enables *local* connections to the cloud using the MQTT protocol. DataHub buffers the data and if the connection drops out, no data is ever lost. DataHub enables Voice of the Machine Edge to run offline, but can push all offline data to the cloud once an internet connection is established. This helps in situations where network connectivity is not always reliable.
- The capacity of the data is limited by the hardware footprint and the data is discarded on a first-in first-out basis. To visualize these connections: Use Voice of the Machine Edge Flows to Visualize Data Flows.

What is the DeviceHub?

DeviceHub provides the mechanism for connecting to physical devices (PLCs, sensors, etc.) to collect data from them.

- DeviceHub needs to be configured for *southbound* connections, including connections to PLCs.
- DeviceHub collects data from physical devices and publishes it to an internal message broker. The data can be sent securely to the cloud via DataHub, or it can be managed locally and sent to any desired destination, or both.
- DeviceHub's main purpose is to collect data from PLCs (Programmable Logic Controller), classify the data by adding OMA tagging, and publish it to a Message Broker subject for further distribution.
- PLC-level statistics and input/output data can be obtained at the edge using DeviceHub.
- There are 35+ device drivers included in DeviceHub. DeviceHub supports many PLC manufacturers, such as Siemens, Allen-Bradley, Mitsubishi, Omron, etc.

Do I need to configure both DeviceHub and DataHub or are they independent?

- These Voice of the Machine Edge components are independent and need separate configurations. To get the maximum value out of their Voice of the Machine Edge environments, however, most customers require both configurations.
- **DeviceHub** is the component that needs to be configured for *southbound* connections, including connections to PLCs.
- **DataHub** needs to be configured for *northbound* connections. Upon activation from LoopCloud, a DataHub Cloud Connector is configured, by default. This connector must then be enabled to send data to the cloud. Local NATS topics can then be connected to the cloud connector in DataHub.

What is Flows?

- Flows offers flow-based programming with a drag-and-drop flow editor. It can be used for data manipulation, data normalization, and visualization.
- Refer to

How can I push a flow to many devices?

- Flows has an import/export feature to share flows.
- Refer to How to Use Flow Options

Why doesn't the Flows "export to clipboard" option copy the data to the clipboard?

I thought I exported a flow to the clipboard, but when I try to import from the clipboard, the text is not in the clipboard.

- Before exporting nodes, select the nodes and connecting wires within a flow tab.

What is the purpose of a Device Tag in DeviceHub?

- A DeviceHub device tag refers to the PLC tag/register. PLC registers hold PLC runtime information.
- Tags are the actual input and output registers in a device to which sensors are connected. Every PLC has a tag and tags differ from PLC to PLC.
- Examples of tags include: digital input, digital output, analog input, analog output, timers, counters, etc.

Can we program a PLC using Voice of the Machine Edge?

No. The PLC (Programmable Logic Controller) needs to be already programmed. Voice of the Machine Edge reads the PLC data.