460ETCQT-NNA1
Protocol Gateway

Product User Guide

Firmware Version 8.8.37

Trademarks

CompactLogix, ControlLogix, & PLC-5 are registered trademarks of Rockwell Automation, Inc. EtherNet/IP is a trademark of the ODVA. MicroLogix, RSLogix 500, and SLC are trademarks of Rockwell Automation, Inc. Microsoft, Windows, and Internet Explorer are registered trademarks of Microsoft Corporation. BACnet® is a registered trademark of American Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE). All other trademarks and registered trademarks are the property of their holders.

Limited Warranty

Real Time Automation, Inc. warrants that this product is free from defects and functions properly.

EXCEPT AS SPECIFICALLY SET FORTH ABOVE, REAL TIME AUTOMATION, INC. DISCLAIMS ALL OTHER WARRANTIES, BOTH EXPRESSED AND IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR APPLICATION. THIS LIMITED WARRANTY GIVES YOU SPECIFIC LEGAL RIGHTS. YOU MAY ALSO HAVE OTHER RIGHTS, WHICH VARY FROM STATE TO STATE.

The examples and diagrams in this manual are included solely for illustrative purposes. Because of the many variables and requirements associated with any particular application, Real Time Automation, Inc. cannot assume responsibility or liability for actual use based on the examples and diagrams. Except as specifically set forth above, Real Time Automation and its distributors and dealers will in no event be liable for any damages whatsoever, either direct or indirect, including but not limited to loss of business profits, income, or use of data. Some states do not allow exclusion or limitation of incidental or consequential damages; therefore, the limitations set forth in this agreement may not apply to you.

No patent liability is assumed by Real Time Automation with respect to use of information, circuits, equipment, or software described in this manual.

Government End-Users

If this software is acquired by or on behalf of a unit or agency of the United States Government, this provision applies: The software (a) was developed at private expense, is existing computer software, and was not developed with government funds; (b) is a trade secret of Real Time Automation, Inc. for all purposes of the Freedom of Information Act; (c) is “restricted computer software” submitted with restricted rights in accordance with subparagraphs (a) through (d) of the Commercial “Computer Software-Restricted Rights” clause at 52.227-19 and its successors; (d) in all respects is proprietary data belonging solely to Real Time Automation, Inc.; (e) is unpublished and all rights are reserved under copyright laws of the United States. For units of the Department of Defense (DoD), this software is licensed only with “Restricted Rights”: as that term is defined in the DoD Supplement of the Federal Acquisition Regulation 52.227-7013 (c) (1) (ii), rights in Technical Data and Computer Software and its successors, and: Use, duplication, or disclosures is subject to restrictions as set forth in subdivision (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at 52.227-7013. If this software was acquired under GSA schedule, the U.S. Government has agreed to refrain from changing or removing any insignia or lettering from the Software or documentation that is provided or from producing copies of the manual or media. Real Time Automation, Inc.

Revision History

Version	Date	Notes
8.4.5	11/18/2019	Features Added 1. Released OPC UA Server (US) Protocol 2. Ability to now Import/Export Template Files with out an FTP session Bug Fixes 3. Updated Profinet Server (PS) on N34 hardware Platform 4. Updated Wi-Fi software
8.6.0	2/28/20	Bug Fixes 1. Omron Plc Communication fixes for EtherNet/IP 2. Profinet GSDML Substitute values fix
8.7.4	9/1/20	Features Added: 1. BMS, BM, DFM, DS, DM, TCP, USB, PBS have been ported to the latest base software 2. TCP,BMS,BM now Available on N2E and N2EW hardware Platform 3. New ASCII Mode Available on TCP/A/USB/WI protocols 4. User Guides updated with more examples Bug Fixes: 1. Improved Data Mapping and String Mapping performance 2. Improved functionality/performance on EC,ETC,ES,MC,MS,BS,BC, A,,WI,PS protocols
8.7.22	4/6/21	Features Added: 1. Support for RSLogix Versions 32 + with unsigned data type support 2. ETC now support Long integer files (L files) for MicroLogix PLCS that support them 3. SC now supports data block (DB) access
8.7.53	4/28/21	Features Added: 5. Added support for the NNBU hardware platform 6. Improved RFIDeas scanner support 7. Updated MM and MRS to use Modbus RTU Client and Modbus RTU Server terminology

Version	Date	Notes
8.9.22	2/5/24	Features Added: 1. Added priority-based reads for client protocols 2. Added improved diagnostic timers for client protocols 3. Reduced minimum delay between messages to zero ms on client protocols 4. Added support for USB serial connections 5. Added support for multiple connections on EtherNet/IP Adapter 6. Added 100ms and 1000ms heartbeat values for diagnostic use 7. Added configurable data size to EtherNet/IP adapter and DeviceNet Slave 8. Added support for TTL communications on N34, NNA1, NNA4, N2E, and N2EW hardware 9. Added support for JSON payloads to MQTT 10. Added Network Bitmap Status to ASCII, USB, and TCP protocols Bug Fixes: 11. Fixed COV Subscription Issues on BACnet MS/TP 12. Fixed timing issues affecting gateway performance 13. Fixed a bug where the Run Idle Header on the output instance for EtherNet/IP Scanner was not checked by default
8.9.29	4/1/24	Features Added: 14. Added ability to do raw HEX byte copy when receiving data over ASCII, TCP, or USB. Bug Fixes: 15. Fixed bug where function code 15 did not work on MM/MC. 16. Fixed bug relating to writing zeros on start up on BS. 17. Fixed bug where MQTT client did not appear in display data page when MQTT was paired with BACnet
8.9.37	7/30/24	Bug Fixes: 18. EIP IO Communication fixes 19. Timing fixes 20. USB Fixes a. Inactivity Timeout b. Inactivity Timeout Logging c. Port Restart Logging d. Webpage fixes 21. ProfiNet Timing Fix 22. EIP PanelView Fixes a. Support for Explicit Messaging

Overview

The 460ETCQT-NNA1 gateway Easily and securely connect Allen-Bradley PLCs to an MQTT broker . By following this guide, you will be able to configure the 460ETCQT-NNA1 gateway.

Number of ASCII devices is dependent on the Hardware and Product number of the 460 gateway.

For further customization and advanced use, please reference the appendices located online at: http://www.rtautomation.com/product/460-gateway-support/.

If at any time you need further assistance, do not hesitate to call Real Time Automation support. Support Hours are Monday-Friday 8am-5pm CST

Toll free: 1-800-249-1612
Email: support@rtautomation.com

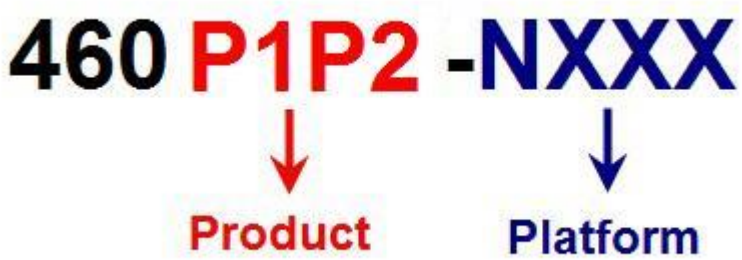
Hardware Platforms

The 460 Product Line supports a number of different hardware platforms. There are differences in how they are powered, what serial settings are supported, and some diagnostic features supported (such as LEDs). For these sections, be sure to identify the hardware platform you are using.

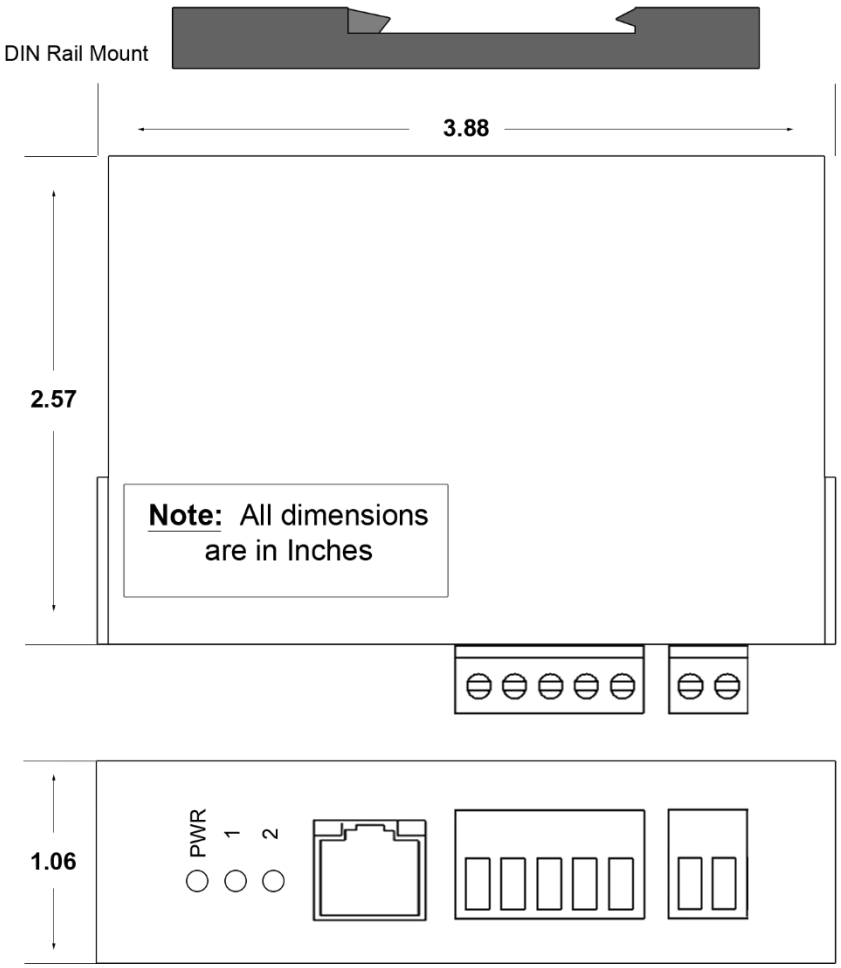
To find which hardware platform you are using:

1) Look on the front or back label of the unit for the part number.
2) On the webpage inside the gateway, navigate to the dropdown menu under **Other** and select **Utilities**. Click the **Listing of Revisions** button. The full part number is displayed here.

Once you have the full part number, the platform will be the number following the "-N":

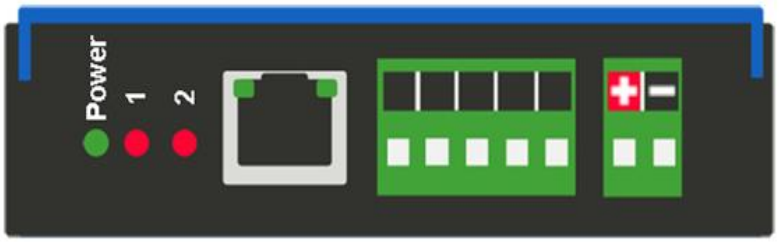


Hardware – NNA1



Powering the Gateway

1) Connect a 12-24 VDC power source to the gateway, Red Wire = (+) Black Wire = (-).
   a) The unit draws 175mA @ 12 V.

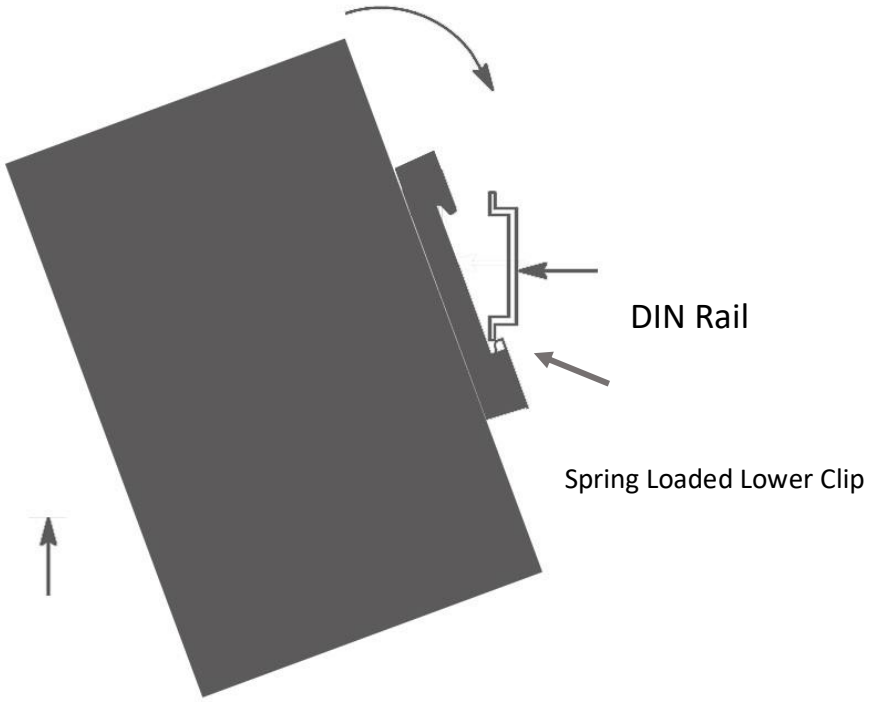


Mounting with a DIN Rail

Installing

Follow these steps to install your interface converter.

1) Mount your DIN Rail.
2) Hook the bottom mounting flange under the DIN Rail.
3) While pressing the 460ETCQT-NNA1 against the rail, press up to engage the spring loaded lower clip and rotate the unit parallel to the DIN Rail.
4) Release upward pressure.



Removing

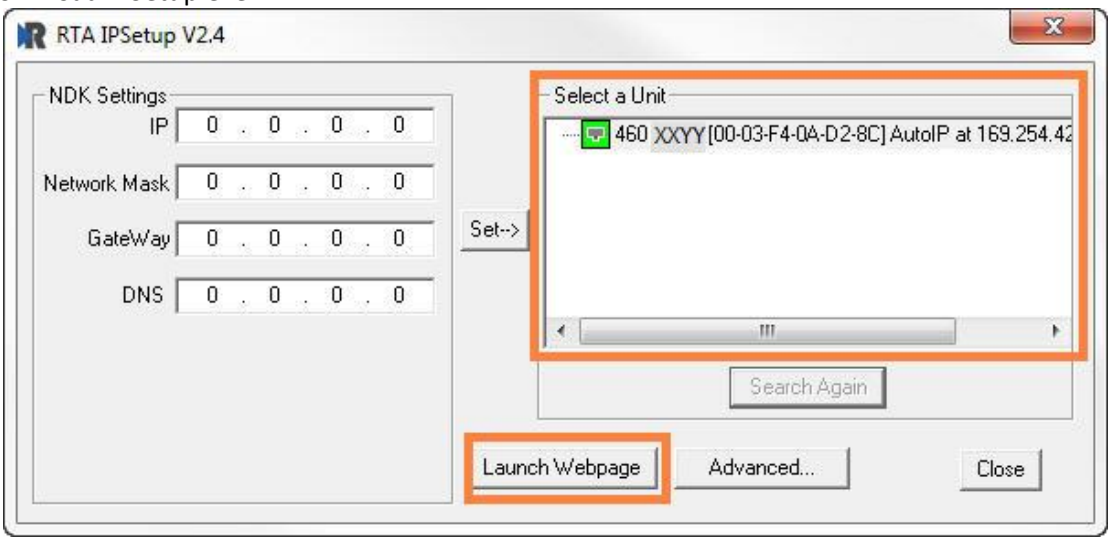
Follow these steps to remove your interface converter.

1) Press up on unit to engage the spring loaded lower clip.
2) Swing top of the unit away from DIN Rail.

Accessing the Main Page

The following steps will help you access the browser based configuration of the gateway. By default, DHCP is enabled. If the gateway fails to obtain an IP address over DHCP it will Auto IP with 169.254.X.Y. For more information on your Operating system network setting refer to the Accessing Browser Configuration document from our support web site.

1) Scan the QR code on the back of the unit or navigate to www.rtautomation.com/460-gateway-support and download IPSetup.exe.



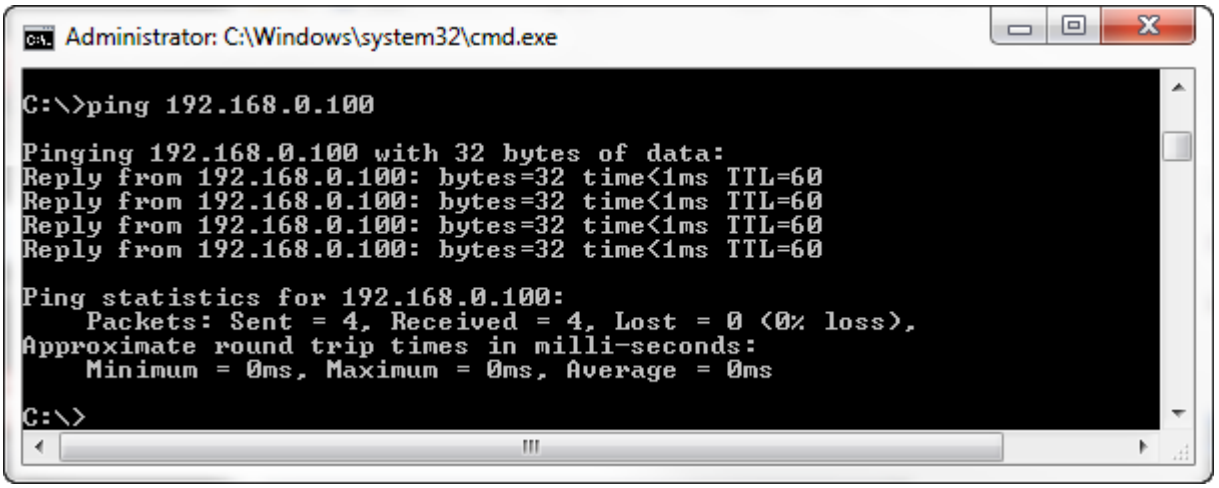
2) Run the IPSetup.exe program.
3) Find unit under "Select a Unit".
   a. Change Gateway's IP address to match that of your PC if DHCP has failed.
      i. You will know DHCP has failed if the gateway's IP address is AutoIP at 169.254.X.Y.
      ii. If successful, it will say DHCP'd at ex: 192.168.0.100 or however your DCHP Client is set up.
   b. If you do not see the gateway in this tool, then your PC is most likely set up as a static IP.
      i. Change your PC's network settings to be DHCP. If DHCP fails, then it will change to be on the 169.254.x.y network.
      ii. Relaunch the IP Setup tool to see if gateway can be discovered now.
4) Click **Launch Webpage**. The Main page should appear.

Default setting is set to DHCP. If DHCP fails, default IP Address is 169.254.x.y

Error: Main Page Does Not Launch

If the Main Page does not launch, please verify the following:

1) Check that the PC is set for a valid IP Address
   a. Open a MS-DOS Command Prompt
   b. Type "ipconfig" and press enter
   c. Note the PC's IP Address, Subnet, and Default Gateway
2) The gateway must be on the same Network/Subnet as the PC whether it's setup for DHCP or Static. Once you have both devices on the same network, you should be able to ping the gateway using a MS-DOS Command Prompt.



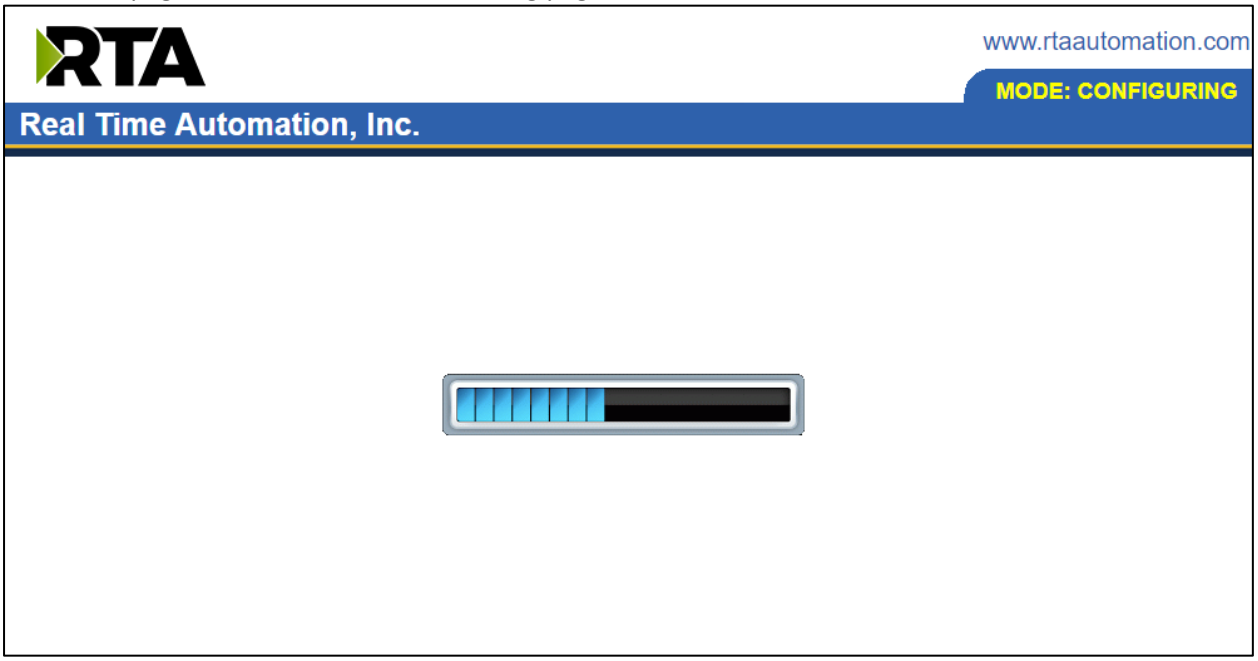
The Screenshot above shows a gateway that is currently set to a static IP Address of 192.168.0.100. If you are able to successfully ping your gateway, open a browser and try to view the main page of the gateway by entering the IP Address of the gateway as the URL.

Committing Changes to the Settings

All changes made to the settings of the gateway in Configuration Mode will not take effect until the gateway is restarted via the webpage. Changes will not be stored if the gateway's power is removed prior to a reboot.

NOTE: The gateway does not need to be restarted after every change. Multiple changes can be made before a restart, but they will not be committed until the gateway is restarted.

When all desired changes have been made, press the **Restart Now** button.
The webpage will redirect to our rebooting page shown below:



The reboot can take up to 20 seconds.
If the IP address has not been modified, the gateway will automatically redirect to the main page.
If the IP address was modified, a message will appear at the top of the page to instruct the user to manually open a new webpage at that new IP.

Main Page

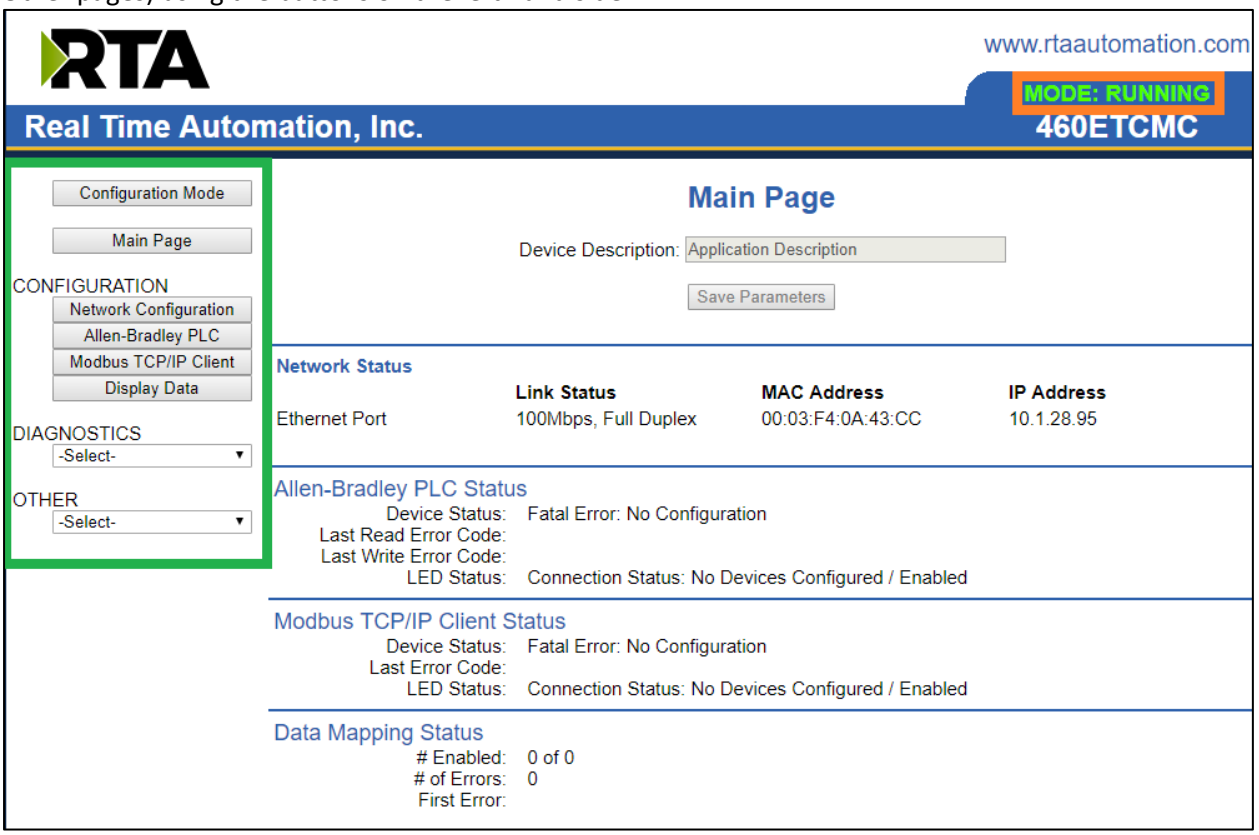
The main page is where important information about your gateway and its connections are displayed.
Mode (orange box below):
Running Mode:

- Protocol communications are enabled
- Configuration cannot be changed during Running Mode. If changes are needed, click the **Configuration Mode** button shown in the green box below

Configuring Mode:

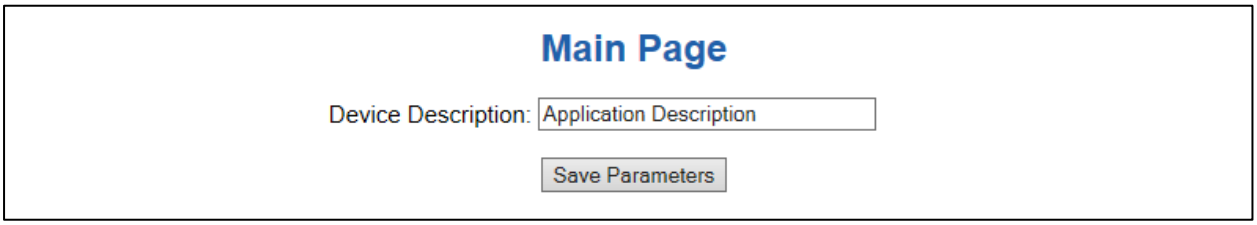
- Protocol communication is stopped and no data is transmitted
- Configuration is allowed

Navigation (green box below):
You can easily switch between modes and navigate between pages (Configuration, Diagnostics, and Other pages) using the buttons on the left hand side.



Device Configuration

The device configuration area is where you assign the device description parameter. Changes can only be made when the gateway is in Configuration Mode.



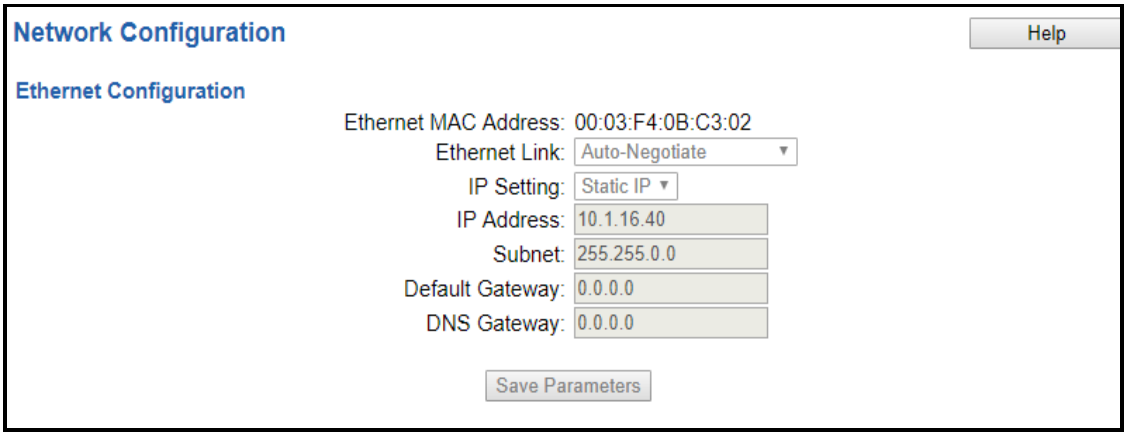
Once you are done configuring the Description, click the **Save Parameters** button.

Network Configuration

The network configuration area is where you assign the IP address and other network parameters. Changes can only be made when the gateway is in Configuration Mode.

Once you are done configuring the Network Settings, click the **Save Parameters** button.

If you are changing the IP Address of the gateway, the change will not take effect until the unit has been rebooted. After reboot, you must enter the new IP Address into the URL.



It is recommended to leave the DNS Gateway set to 0.0.0.0 and the Ethernet Link as Auto-Negotiate. If configuring the gateway to use E-mail, the DNS Gateway must be set.

Allen-Bradley PLC Configuration

Click the **Allen-Bradley PLC** button to access the configuration page.

1) Select which **Network Interface** to use for this Allen-Bradley PLC connection. If using single port hardware, the Network Interface will default to Ethernet port only.

1) **Delay Between Messages**: Enter the length of time to delay between read and write scan line requests (ms).

2) **Response Timeout**: Enter the amount of time the gateway should wait before a timeout is issued for a read/write request (ms).

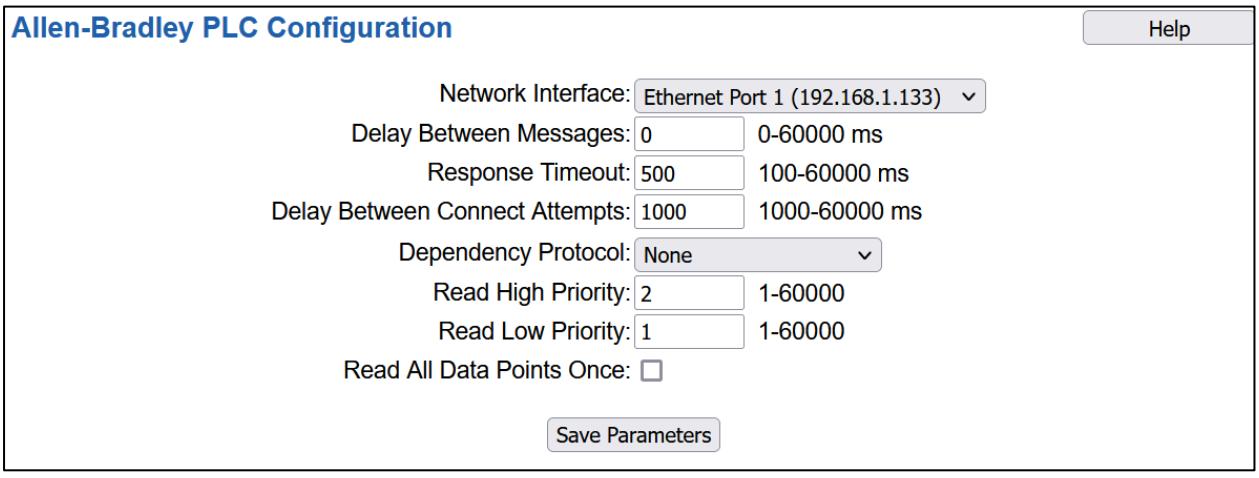
3) **Delay Between Connect Attempts**: Enter the amount of time the gateway should wait between attempts to connect to the PLC.

4) **Dependency Protocol**: If enabled, the Allen-Bradley PLC communication will stop if communication to the selected protocol is lost.

5) **Read High Priority**: Configures the number of high priority requests to process before switching to low priority requests. This number should be higher than the Read Low Priority.

6) **Read Low Priority**: Enter the number of low priority requests to process before switching to high priority requests. This number should be lower than the Read High Priority.

7) **Read All Data Points Once**: If Enabled, the gateway will read all configured data points once on startup regardless of priority, then begin processing requests based on priority after all points have been read once.



External PLC Configuration

The bottom area of the Allen-Bradley PLC Configuration page lets you configure up to five PLCs.

There are three ways to configure this protocol:

1) Auto-Configure Group by Device (Default)

2) Auto-Configure Group by Data Type

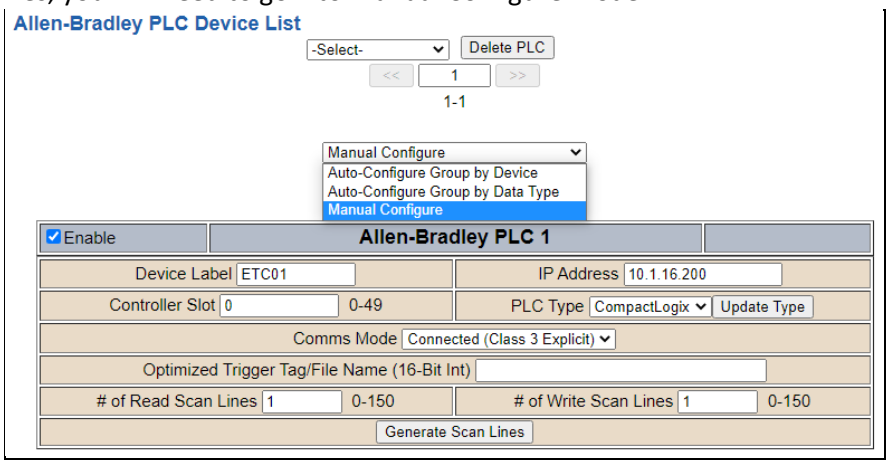
3) Manual Mode

NOTE: You may go back and forth between modes, but when reverting from Manual Mode to either of the two Auto-Configure Modes, all changes made in Manual Mode will be discarded.

1) To add additional PLCs, click the -Select- dropdown under Allen-Bradley PLC Device List and select **Add Generic PLC** option.
   a) To remove a device, navigate to the server to delete using the **<<** and **>>** buttons and click the **Delete PLC** button.
   b) To create a new PLC with the same parameters already configured from another PLC, click the -Select- dropdown and select the **Add from PLC X** option (where X represents the PLC you wish to copy parameters from). Once created, you can make any additional changes needed to that new PLC.

   NOTE: Auto-Configure Modes can ONLY be used in PLC 1.

2) To edit scan lines, you will need to go into Manual Configure Mode.



External PLC Configuration: Auto-Configure

While in either of the two Auto-Configure modes, the number of scan lines and the actual scan lines themselves cannot be edited. Auto-Configure Mode looks at the other protocol and then configures the scan lines within the PLC to match. The PLC Tag/File Names and Data Types will be defined after the other protocol is configured.

If the PLC is a CompactLogix, ControlLogix or FlexLogix, the data will be configured according to the following rules:

1) Any 8 Bit Signed/Unsigned data will be mapped as **Sint**.
2) Any 16 Bit Signed/Unsigned data will be mapped as **Int**.
3) Any 32 Bit Signed/Unsigned data will be mapped as **Dint**.
4) Any 32 Bit Float and 64 Bit Float data will be mapped as **Real**.
5) Any Coils or 1 Bit Binary Packs will be mapped as **Bool (1 Bit**).
6) Any Coils or 8/16/32 Bit Binary Packs will be mapped as **Bit Array (32 bit)**.
7) Any String Data Types will be mapped as **String**.

If the PLC is a MicroLogix, SLC or PLC5E, the data will be configured according to the following rules:

1) Any 8 Bit Signed/Unsigned and 16 Bit Signed/Unsigned data will be mapped as **Int**.
2) Any 32 Bit Signed/Unsigned, 32 Bit Float, and 64 Bit Float data will be mapped as **Real**.
3) Any Coils or 1/8/16/32 Bit Binary Packs will be mapped as **Bit Array (16 bit)**.
4) Any String Data Types will be mapped as **String**.

Regardless of PLC type, the following is also true:

1) The read or write direction depends on whether it is configured as a read or write on the other protocol.
2) If the other protocol exceeds the number of Sint, Int, Dint, Real, Bool, Bit Array, or String data types the Allen-Bradley PLC supports (see limits on webpage), then nothing will be mapped. You will see the number of scan lines remain at 0 and the main page will display the following error:

   a) To fix this error, simply decrease the amount of data you configured on the other protocol so that the max number of Tag/File Name is not exceeded or call customer support to increase the limits.

Auto-Configure Group by Device vs. Auto-Configure Group by Data Type

There are two different methods for Auto-Configure: Group by Device or Group by Data Type.

There are a couple of rules to keep in mind when using Auto-Configure Mode:

1) If the other protocol inside the gateway is a server, slave, or adapter protocol, then there are no differences between the Auto-Configure modes.

Group by Device (Default Method)

Group by Device goes through the other protocol on the gateway and auto-configures the data groups on the Allen-Bradley PLC for all the data points on the other protocol's first device. After it finishes with the first device, it will auto-configure all the points for the second device (if one is configured), and so on.

The data in this method is not optimized- there could potentially be a lot of wasted/unused data space, but it will be organized more logically from the master/client's point of view.

Group by Data Type

Group by Data Type goes through the other protocol on the gateway and auto-configures the data groups on the Allen-Bradley PLC for all the data points within the other protocol.

Another way to view this option is to say that the data points allocated are packed together so there is very little wasted data space. The data is packed or optimized.

***Example**:* *Protocol A is a master/client protocol that has 2 devices with the same setup:*

Device_1 has 1 integer scan line, 1 float scan line, 1 integer scan line- each for 1 point of data
Device_2 has 1 integer scan line, 1 float scan line, 1 integer scan line- each for 1 point of data

Protocol B is a server/slave/adapter protocol that can be mapped as follows:

Group by Device - Protocol B will have 4 scan lines that will look like the following: Scan Line 1 and 2 will represent Device_1 and Scan Line 3 and 4 will represent Device_2.

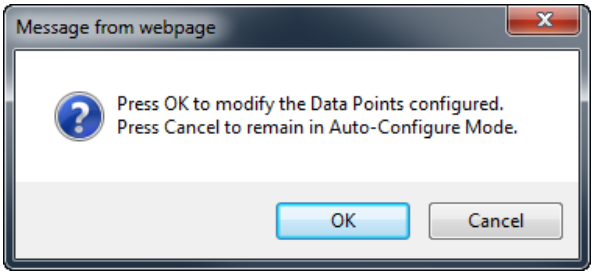
Scan Line 1 => Type Integer, length of 2
Scan Line 2 => Type Float, length of 1
Scan Line 3 => Type Integer, length of 2
Scan Line 4 => Type Float, length of 1

Group by Data Type - Protocol B will have 2 scan lines that will look like the following: Like data types from Device_1 and Device_2 will be combined.

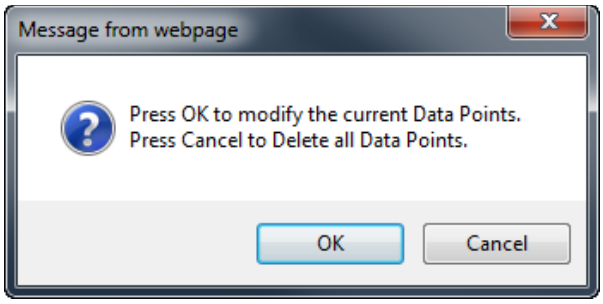
Scan Line 1 => Type Integer, length of 4
Scan Line 2 => Type Float, length of 2

External PLC Configuration: Manual Configure Mode

1) To transition from either of the two Auto-Configure modes to Manual Configure Mode, click the dropdown in the middle of the Allen-Bradley Configuration page and select Manual Configure.
   a) When prompted, click **OK** to confirm mode change or **Cancel** to remain in Auto-Configure Mode.



2) Once OK is clicked, there are two options for how to proceed.



3) To keep the scan lines that are already configured, press **OK**.

   a) You would want this option if you are adding additional scan lines or you want to modify the scan line(s) that already exist.

4) To delete the scan lines that are already there and start over, press **Cancel**.

5) To add additional PLCs, click the -Select- dropdown under Allen-Bradley PLC Device List and select **Add Generic PLC** option.

   a) To remove a device, navigate to the server to delete using the **<<** and **>>** buttons and click the **Delete PLC** button.
   b) To create a new PLC with the same parameters already configured from another PLC, click the -Select- dropdown and select the **Add from PLC X** option (where X represents the PLC you wish to copy parameters from). Once created, you can make any additional changes needed to that new PLC.

6) The **Enable** check box should be selected for the device.
7) Enter a **Device Label** to identify the device within the gateway.

8) Enter the **IP Address** of the PLC, the **Controller Slot** (Integrated Ethernet, use Slot 0), and select the **PLC Type**. The Controller Slot is the slot where the controller is located, not the Ethernet card being used. These three parameters must match the PLC you are communicating to.

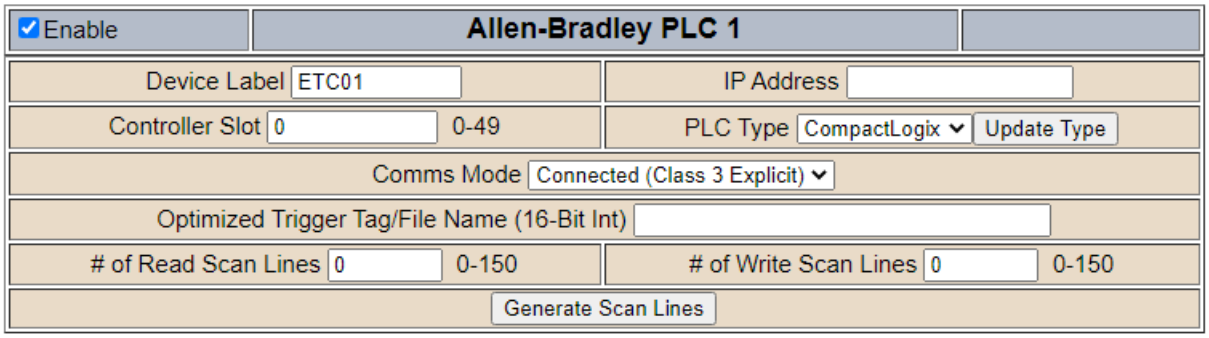
   NOTE: ControlLogix Rev 32 is ONLY supported with CompactLogix 5380 and above and ControlLogix 5580 and above.
   NOTE: When using GuardLogix PLCs select ControlLogix as the PLC Type. When connecting to GuardLogix PLCs the RTA Gateway will be unable to read or write safety tags.

9) Select the **Comms Mode**. **Unconnected (UCMM)** messaging relies on shared resources to transfer data to the PLC. This could result in message timeouts if there are a lot of devices fighting for these shared buffers. If you don't want the RTA gateway to constantly keep the connection open to the PLC but only maintain a connection when there is data needed to be transferred, then Unconnected (UCMM) will work best if you are only writing to the PLC. **Connected (Class 3 Explicit)** messaging relies on reserved resources to transfer data to/from the PLC**. Connected (Class 3 Explicit)** messaging is recommended if you are reading and writing and always want to keep that connection open to the PLC.

10) Enter an **Optimized Trigger Tag/File Name** to enable the triggering optimization that is available. The Optimized Trigger forces the 460ETC gateway to read ONLY the Optimized Trigger Tag until a value has a change of state. Please reference the Optimized Trigger Guide in the section below.

11) Enter the "# of Read Scan Lines" and "# of Write Scan Lines".

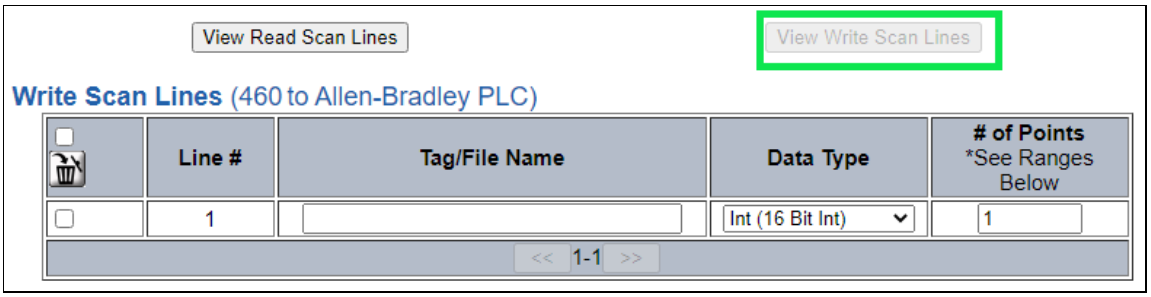
12) Click **Generate Scan Lines** to have the read and write scan lines auto generated for you. If you need to manually configure the read and write scan lines you can do so after they have been generated.

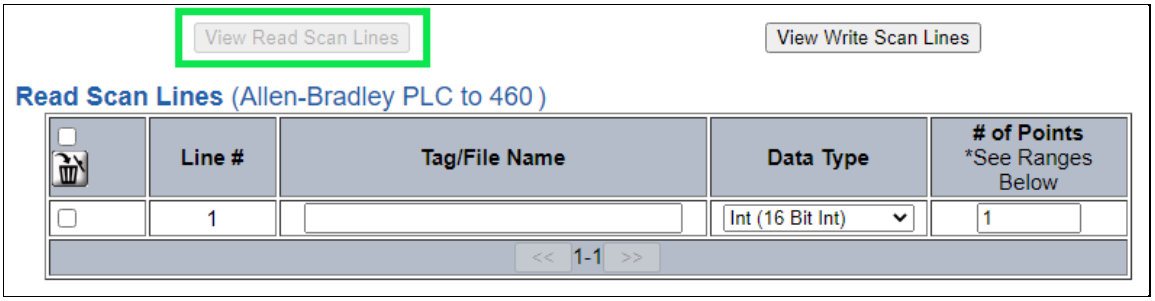


Configuring Read and Write Scan Lines

Follow these steps to manually configure Read and Write Scan Lines.

1) Click the **View Read Scan Lines** or **View Write Scan Lines** button.





2) Enter the **Tag/File Name** that is set up within the PLC. If you are trying to access a tag that is defined in the Program Scope, please see the Access Program Scope Tag section below.

NOTE: If you are **ONLY** using Write Scan Lines, then the RTA gateway will not connect to the PLC until we receive valid data from the source. It's recommended you use **Unconnected** messaging so when the RTA gateway sends data to the PLC, we only send it once and close the connection until a Change of State. Leaving it at Connected messaging, once we receive data, the RTA gateway will constantly be writing to the PLC to maintain that connection.

a) If you wish to start from a point other than the base, add [#] to the end of the Tag/File Name to specify which point is the starting point.

i) **Example:** A tag called "ReadTag" has dimension of 100 in the PLC. By default, we will start at point 0 of that array. Therefore, "ReadTag" and "ReadTag[0]" refer to the same point. To start from a different point, such as array index 27, enter in "ReadTag[27]" as the Tag/File Name in the gateway's scan line. This means the gateway will go to "ReadTag" and start at array index 27.

ii) If you wish to access a specific bit from any data type, you **must** use the Mapping Page's Set Bit math function. *You may **not** use ReadTag/0.0 to access bits.*

3) Select the **Data Type** of the Tag/File.

4) Enter the **# of Points** you want to move from the PLC Tag/File to the gateway. See the *Scan Line Data Limit* section at the bottom of the page for the given max values.

   a) If using a CompactLogix or ControlLogix, below are the scan line data limits.

Scan Line Data Limit

Data Type	Length Range
Bool	1
Bit Array	100
Sint	400
Int	200
Dint	100
Real	100
String	1

   b) If using a ControlLogix Rev 32 (CompactLogix 5380 or ControlLogix 5580), below are the scan line limits.

Scan Line Data Limit

Data Type	Length Range
Bool	1
Bit Array	100
Sint	400
USint	400
Int	200
UInt	200
Dint	100
UDint	100
Real	100
String	1

   c) If using Micrologix PLC, below are the scan line limits.

Scan Line Data Limit

Data Type	Length Range
Bit Array	100
Int	100
Real	50
String	1
Long	50

d) If using a SLC 5/05 or PLC5E, below are the scan line limits.

Scan Line Data Limit

Data Type	Length Range
Bit Array	100
Int	100
Real	50
String	1

5) When configuring read scanlines there is an optional priority configuration. There are three priority selections available, how often each priority is read is configurable in the Allen-Bradley PLC Configuration section using the Read High Priority and Read Low Priority Configurations.

a) High: Read the scanline based on the Read High Priority configuration.

b) Low: Read the scanline based on the Read Low Priority configuration.

c) Once: Read the scanline once on gateway startup or upon a new connection and never again during normal operation.

Read Scan Lines (Allen-Bradley PLC to 460ETCMC)

Line #	Priority	Tag/File Name	Data Type	# of Points *See Ranges Below
1	High	ETC01_N2G0_BIT1	Bool (1 Bit)	1

<< 1-1 >>

High
Low
Once

Save Parameters

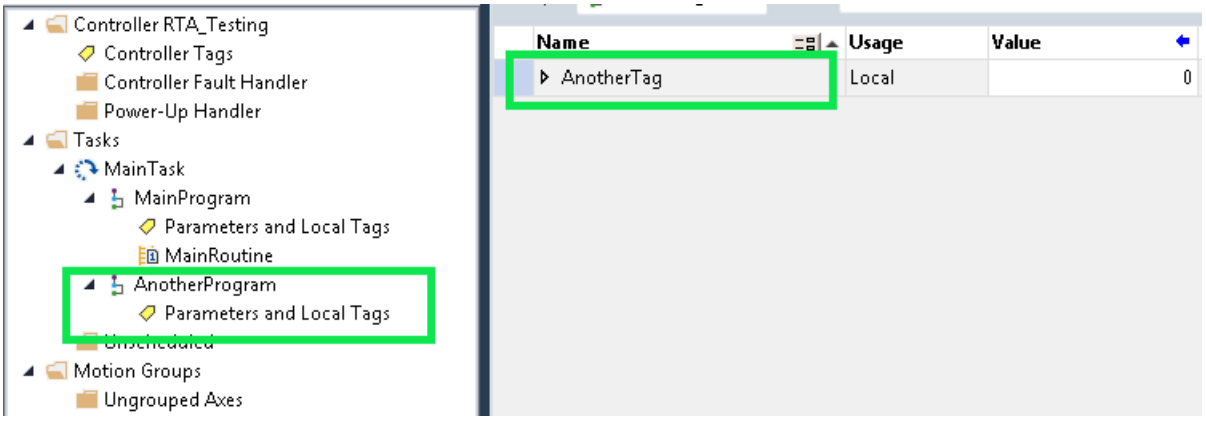
6) Click the **Save Parameters** button.

7) Repeat for the other direction if needed.

Access Program Scope Tags

There are two different types of tags in the PLC: Controller Scope tags and Program Scope tags. With Controller Scope tags, these tag names can be entered into the gateway without any additional syntax. If you are using a tag that is defined within Program Scope, then the tag name inside of the RTA gateway needs additional syntax for it to successfully communicate.

Example: “AnotherTag” is created in the Program Scope called “AnotherProgram”.



To access this Program Scope tag within the RTA 460, you must use the following syntax:

Tag Name = “PROGRAM:ProgramName.TagName” where Program Name = Scope name & TagName = Actual Tag Name, Data type will vary.

	Line #	Tag/File Name	Data Type	# of Points *See Ranges Below
☐	1	PROGRAM:Anotherprogram.Anothertag	Dint (32 Bit Int)	1
<< 1-1 >>				

Optimized Trigger Guide

The Optimized Trigger forces the 460ETC gateway to read ONLY the Optimized Trigger Tag until the trigger value has a change of state. Once there is a change of state then it will mark **ALL** ETC Read Scan Lines "Invalid", then will execute a read for all ETC Read Scan Lines until **ALL** read data is valid. Once all Read Scan Lines have been read and marked valid, it will set the ETC Handshake **#** to the value of ETC Optimized Trigger. You will be able to utilize the ETC Handshake **#** to map over to any of the Technology Triggers and/or as a Handshake Reference.

Note: **#** represents the Allen-Bradley PLC **#** on the Allen-Bradley configuration page of the gateway, if you only have 1 PLC configured your **#** is 1

If you have a timeout and we are not able to read a particular Read Scan Line, then you will stay in a loop of trying to make sure all data is valid before setting the Handshake value equal to Trigger value.

How does this work?

1) Read ETC Optimized Trigger tag until Change of State.
    a. Value 0 = Enabled but Not valid value
    b. Value 65535 = Disabled
2) Map the ETC Optimized Trigger (Source) over to ETC Trigger # (Dest).
3) If ETC Trigger # value changes states, mark all ETC Read Scan Lines "Invalid".
4) Read all ETC Read Scan Lines until ALL source read data is valid.
5) ETC Handshake # value is set equal to ETC Trigger 0 value.
6) Map ETC Handshake # to protocol 2 Technology Trigger (A/USB/TCP/WI) and/or reference data point.

How do you set this up?

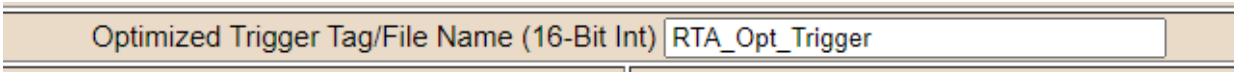
There are 2 options below to synchronize all data when sending data over to protocol 2.

Option 1: Sends data every trigger no matter if it's new or not

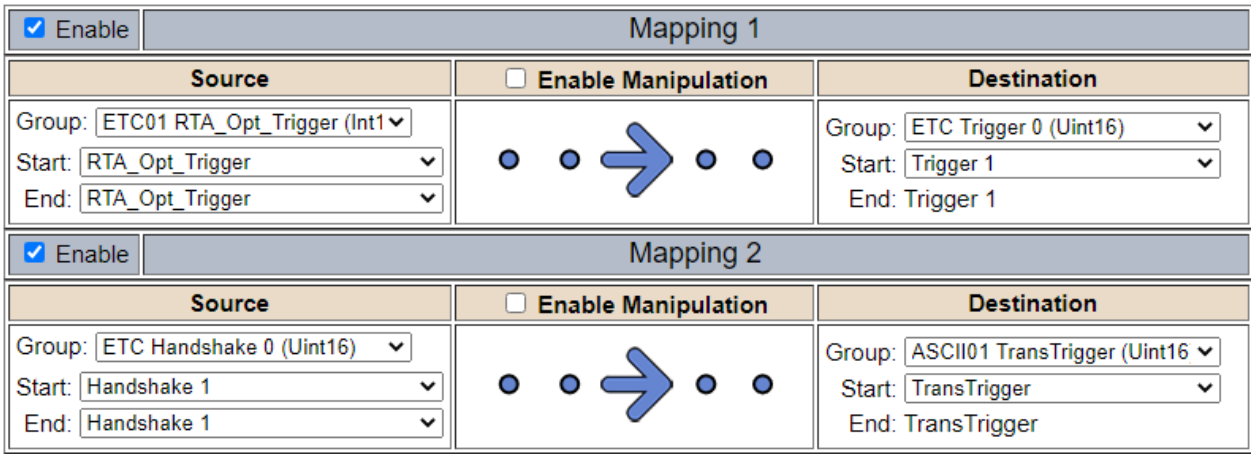
We'll be using an 460ETCA for this example, this will utilize the ETC Optimization Trigger and the Technology Trigger (A/USB/TCP/WI) for ASCII (A).

1) Configure all your Read Scan Lines your looking to send over to your ASCII device.
2) Within the ETC configuration, setup a PLC tag that you can identify as your Optimization Trigger.

Optimized Trigger tag can be unique to your PLC program



3) In the Data Mapping page, manually add 2 additional mappings identical to the example below.

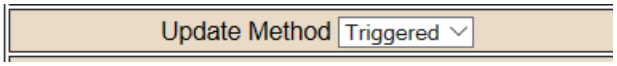


4) Update all your Read Scan Line PLC tags with data.
5) Nothing should have updated in your ASCII device.
6) Update the RTA_Opt_Trigger PLC tag to 1.

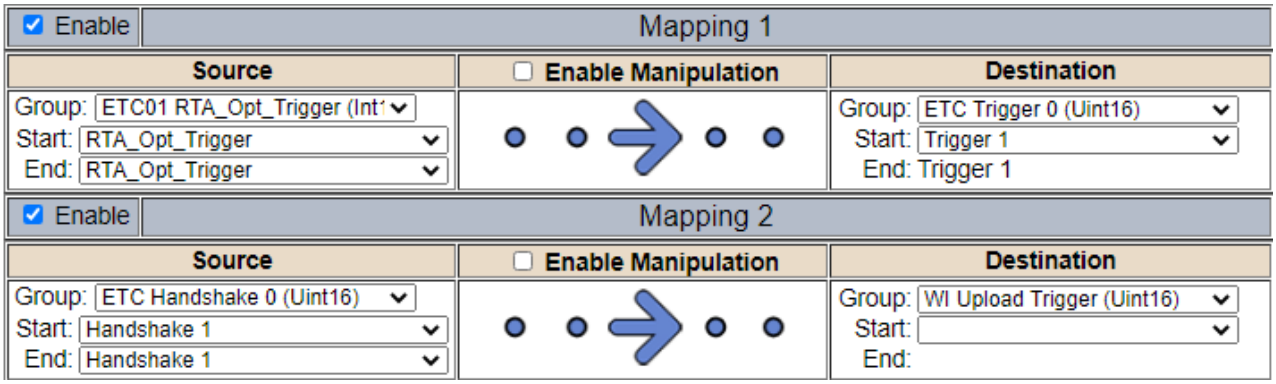
7) Now your ASCII device will be updated with the data.
8) Increment the RTA_Opt_Trigger PLC tag
9) Your ASCII device will get update again, regardless if data is new or not.

If your product is a Web Interface e.g. 460ETCWI acting only as a Client:

1) On the WI configuration page change the Update Method to be Triggered.



2) In the Data Mapping page, manually add 2 additional mappings identical to the example below.



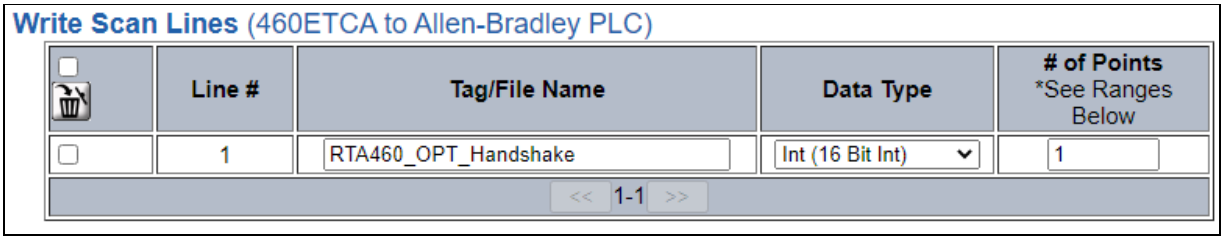
Option 2: Sends data ONLY on Change of State

We'll be using an 460ETCA for this example, this will utilize the ETC Optimization Trigger and a Write Scan Line for a handshake so that the PLC knows the triggering functionality is working.

If using the WI (Web Interface 460ETCWI) then use the WI Upload Trigger in your destination mapping

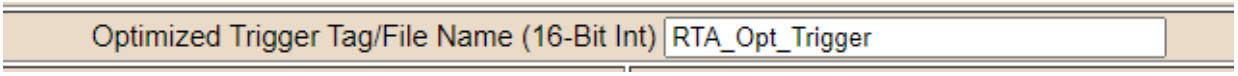
1) Configure all your Read Scan Lines your looking to send over to your ASCII device.
2) Configure a Write Scan line that updates the PLC with the Handshake from the RTAgateway.

Handshake tag can be unique to your PLC program

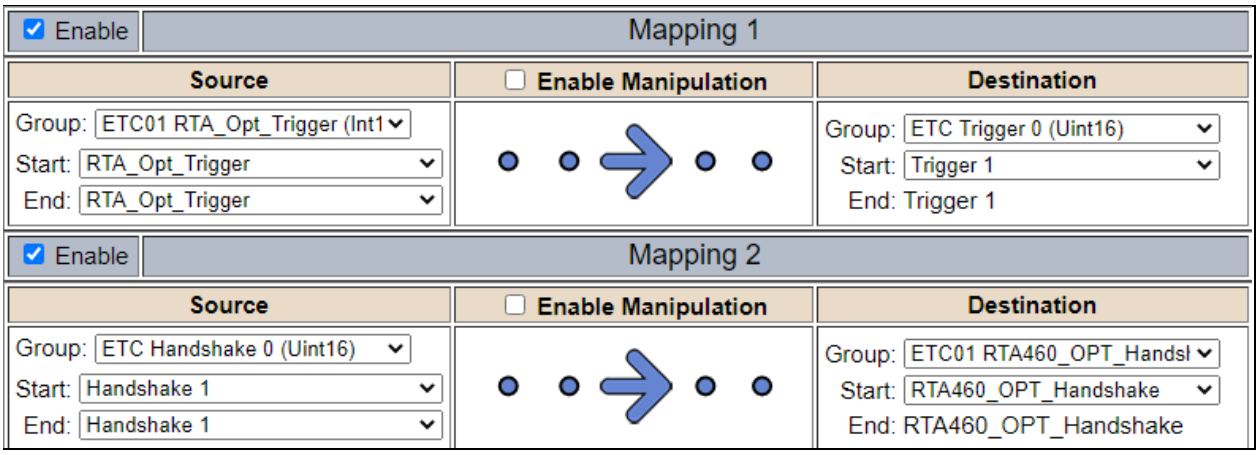


3) Within the ETC configuration, setup a PLC tag that you can identify as your Optimization Trigger.

Optimized Trigger tag can be unique to your PLC program



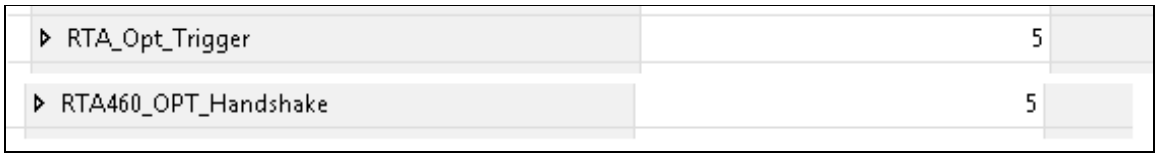
4) In the Data Mapping page, manually add 2 additional mappings identical to the example below.



5) Update all your Read Scan Line PLC tags with data.
6) Nothing should have updated in your ASCII device.
7) Update the RTA_Opt_Trigger PLC tag to 1.

8) Now your ASCII device will be updated with the data.
9) Increment the RTA_Opt_Trigger PLC tag
10) The ASCII device should NOT be updated because the data is not new.
11) Update your Read Scan Line tag with new data.

12) Increment the RTA_Opt_Trigger PLC tag again
13) Now your ASCII device will be updated with the new data.
14) In a working application the Handshake tag in your PLC should match the Optimization Trigger tag.



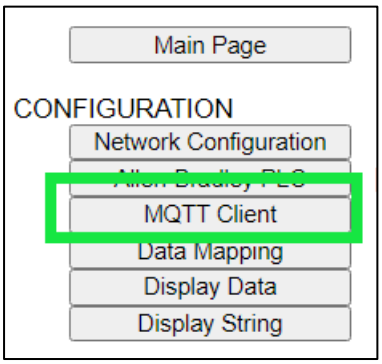
MQTT Client Configuration

You can configure up to three MQTT connections.

1. Configure up to three MQTT broker devices.
2. Configure up to one Amazon Web Services (AWS) IoT Core connection.
3. Configure up to one Microsoft Azure connection.

NOTE: A single AWS OR Azure connection is supported and can co-exist with up to two additional MQTT broker devices.

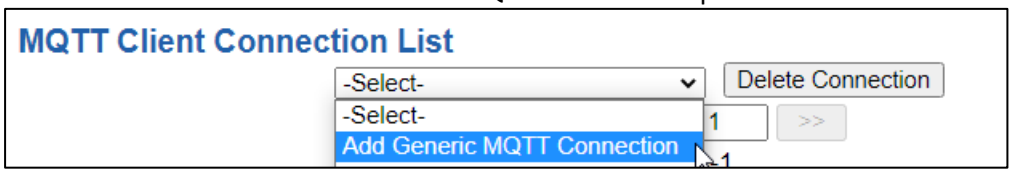
Click the **MQTT** button to continue configuration.



MQTT Devices Configuration

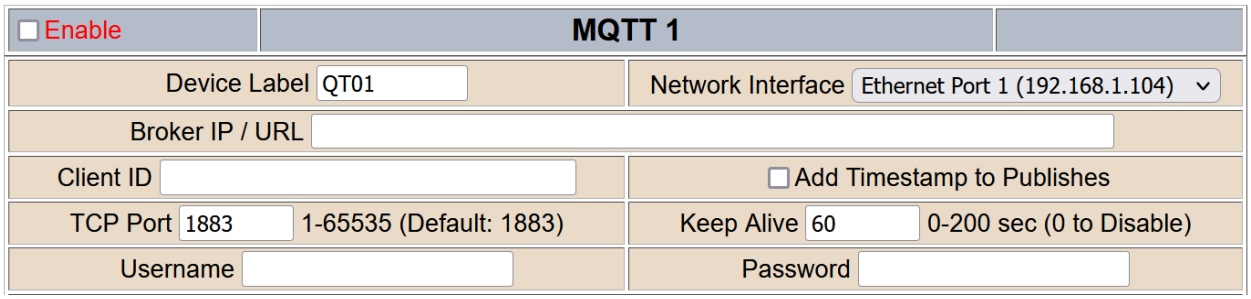
You can configure up to three MQTT devices.

1) To add an MQTT device, or additional MQTT devices, click the -Select- dropdown menu under MQTT Client Connection List and select **Add Generic MQTT Connection** option.

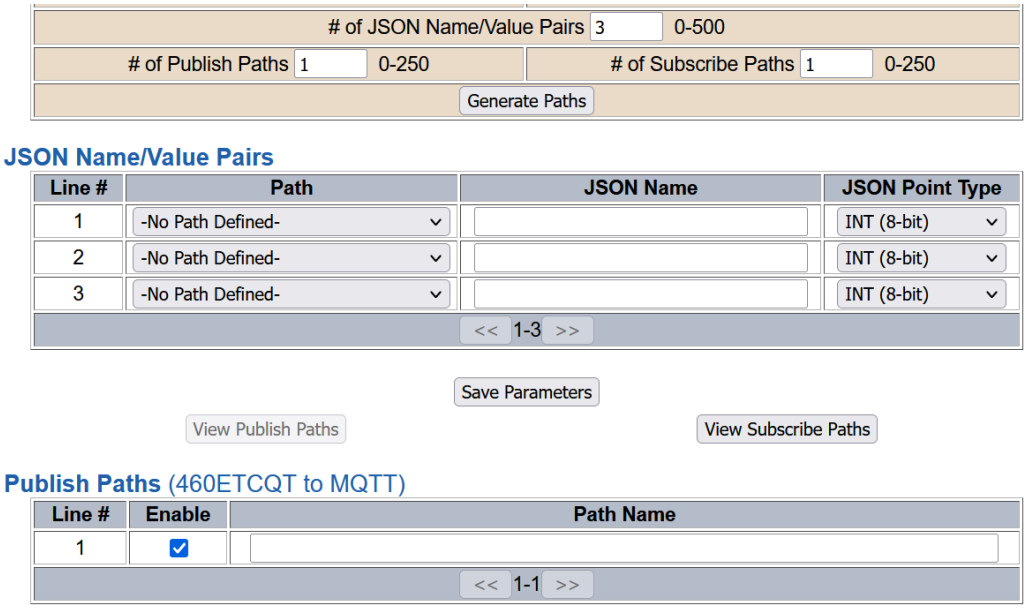


   a) To remove a device, navigate to the MQTT device and click the **Delete Connection** button.
   b) To create a new MQTT device with the same parameters already configured from another MQTT device, click the -Select- dropdown menu and select the **Add from MQTT X** option (where X represents the MQTT device you wish to copy parameters from).
   c) Once created, you can make any additional changes needed to that new MQTT device.
2) The **Enable** check box should be selected for the device to enable communications.
3) Enter in a **Device Label** to identify the device in the gateways.

4) Select which **Network Interface** to use for MQTT device connection. Option only available on the N2E hardware platform.
5) Enter the unique MQTT **broker IP address** or **URL**, if this value does not match, the gateway will timeout.
6) Enter **TCP Port** for the MQTT broker to open a connection on. If this value doesn't match, the gateway will not open a connection.
7) **Keep Alive:** Enter in the amount of time that the gateway should attempt to ping the broker to keep the MQTT connection alive, 0 disables this feature.
8) Enter a **Client ID** to be used when connecting to the Broker.
9) Check the **Add Timestamp to Publishes** checkbox to add a timestamp to the published JSON payloads.
10) **Username and Password:** Enter if authentication to the MQTT broker is necessary.



Configuring Subscribe and Publish Topics



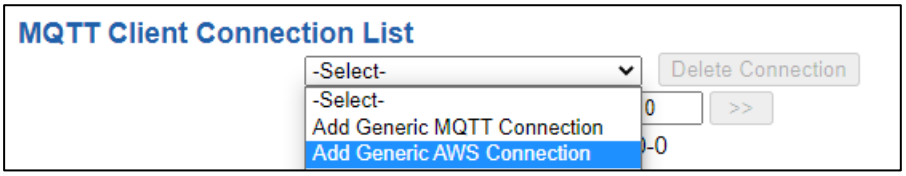
11) Enter in "**# of Subscribe Paths**" and/or "**# of Publish Paths**".
12) Enter in "**# of JSON Name/Value Pairs**"
13) Click the **Generate Paths** button to have the lines generated for you.
14) **# of Subscribe Paths:** Enter in the number of topics to subscribe from the broker. Once the topics are subscribed to, the MQTT broker will publish the messages to the gateway.

15) **# of Publish Paths:** Enter the number of topics to publish to the broker from the mating protocol.
16) **# of JSON Name/Value Pairs:** Enter the number of JSON name/value pairs to configure, each name/value pair can be associated with a single publish or subscribe path.
17) Select the publish or subscribe path each name/value pair should be associated with.
18) Select the **Point Type** of the name/value pairs
19) Click **Save Parameters** button when complete.

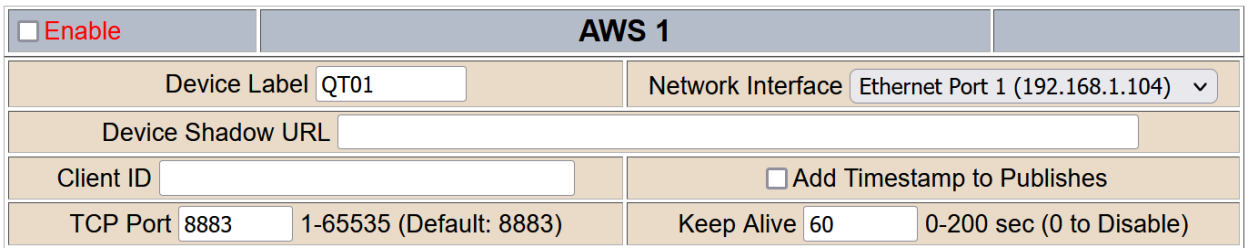
Amazon Web Services (AWS) Configuration

You can only configure one AWS IoT Core connection with your RTA product.

1) To add an AWS connection, click the -Select- dropdown menu under MQTT Client Connection List and select **Add Generic AWS Connection** option.
    a. To remove a device, navigate to the AWS device to delete and click the **Delete Connection** button.



2) The **Enable** check box should be selected for the device.
3) Enter in a **Device Label** to identify the device within the gateways mapping.
4) Select which **Network Interface** to use for AWS IoT Core connection. Option only available on the N2E hardware.
5) **Device Shadow URL:** Enter in the URL path for the AWS MQTT broker.
    a. AWS IoT console will provide you a device shadow URL such as: https://aabb11224e9ex-ats.iot.us-east-2.amazonaws.com/things/RTA_Testing/shadow?name=RTA_Ninja
    b. Within the RTA gateway configuration only enter in "aabb11224e9ex-ats.iot.us-east-2.amazonaws.com" portion of the URL, everything else is ignored.
6) Enter the **TCP Port** for the MQTT broker to open a connection on. If this value doesn't match, the gateway will not open a connection.
7) **Keep Alive:** Enter in the amount of time that the gateway should attempt to ping the broker to keep the MQTT connection alive, 0 disables this feature.
8) Enter a **Client ID** to be used when connecting to the Broker.
9) Check the **Add Timestamp to Publishes** checkbox to add a timestamp to the payload.



Additional AWS Requirements

There are five items that are required to establish an AWS IoT Core connection.

1) The Device Shadow URL
2) Within AWS create a certificate for your RTA gateway. Once AWS has generated a certificate, you'll be given a private key and certificate to download.
3) A root certificate to authenticate the connection. Specifically, the **Starfield Class 2 Certification Authority Root Certificate** is required. This can be obtained from:
    - Starfield techs certificate repository: https://certs.starfieldtech.com/repository
    - or a direct download link here: https://certs.starfieldtech.com/repository/sf-class2-root.crt
4) The private key, certificate, and root certificate will need to be FTP'd into the RTA gateway's Flash File System.
5) The time configuration will need to be set to the current date and time to establish a connection. This can be done in one of two ways:
    - The date and time can be set manually; however, this will need to be set again any time the gateway experiences a power loss or reboot.
    - The date and time can be set automatically by utilizing an NTP server.
    - Both options can be configured by navigating to Other > Time Configuration on the gateway's webpage.

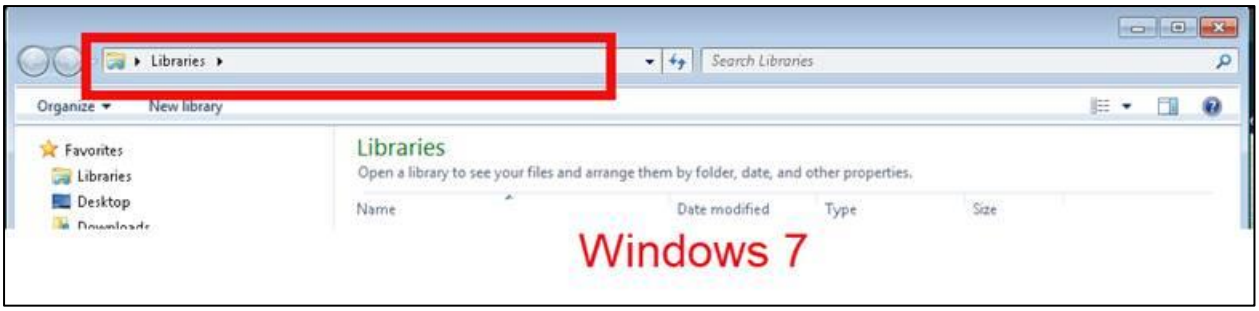
How to FTP files into the RTA gateway

1) Save off the private key and certificate files to your desktop, keep these files in a secured location.
2) Within your Windows Task bar, right click and open a new Windows/File Explorer folder or go into your start menu and type File Explore.

1.  Windows 10

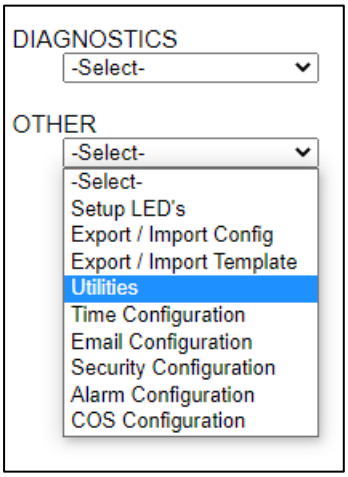
2.  Windows 7

3) You should now have a window that looks like the image below.

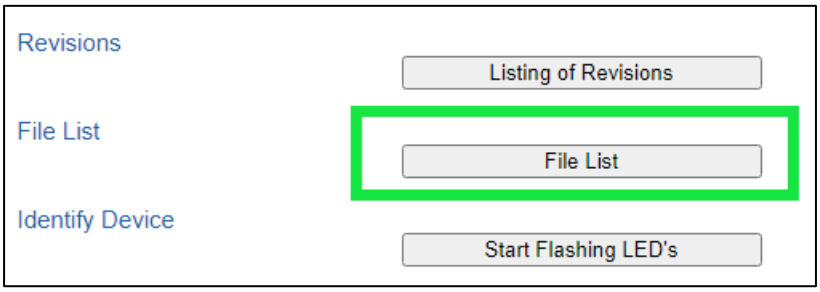


4) In the address bar (within the red box shown above) type **ftp://xxx.xxx.xxx.xxx** (IP Address of RTA gateway).
   a. You will then see a pop-up window, Username: **ffs** Password: **rtarocks**
   b. Paste the certificate and private key into this ftp session, close out the session by exiting out.

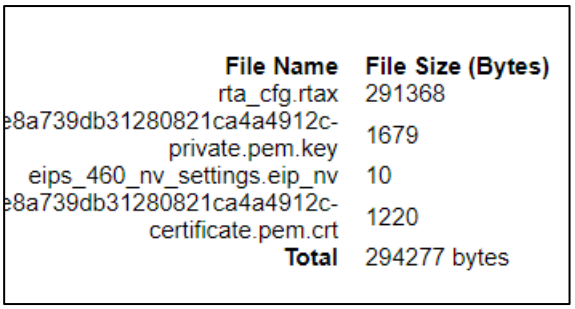
5) Navigate to the RTA gateway and on the left-hand side, click the OTHER -Select- dropdown and select Utilities.



6) Once on the Utilities page click the File List button.



7) Verify that your certification and private key files appear on this page.



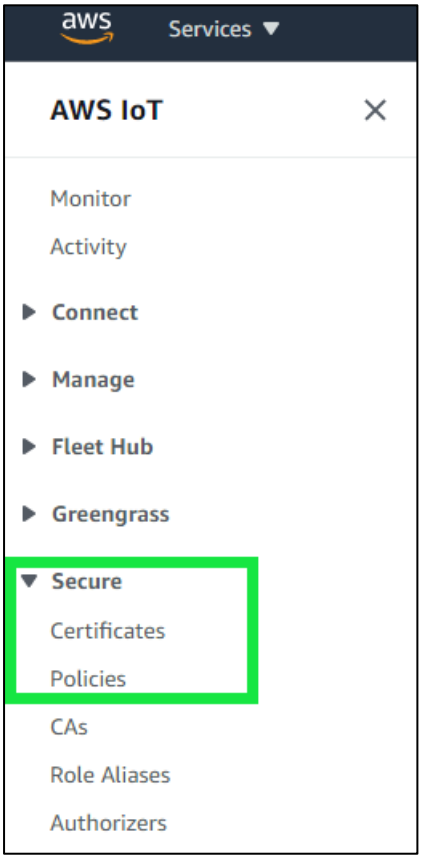
File Name	File Size (Bytes)
rta_cfg.rtax	291368
e8a739db31280821ca4a4912c-private.pem.key	1679
eips_460_nv_settings.eip_nv	10
e8a739db31280821ca4a4912c-certificate.pem.crt	1220
Total	294277 bytes

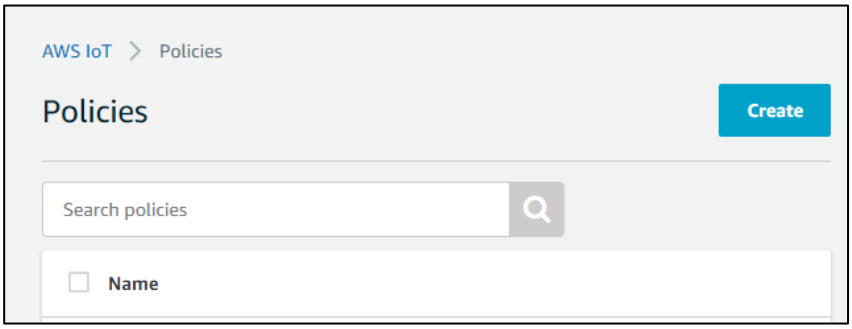
AWS IoT Core Service Setup

Within your AWS account you'll need to navigate to the IoT Core service page where you'll setup a "Thing" and "Policies".

Before you can register your RTA gateway as a "thing," we need to setup a "policy" for it. This policy will be assigned to our "thing" during the registration process and will grant it the permissions needed to access the MQTT topics that we will use to publish and subscribe messages. From the left-hand menu, select "Secure", and then the submenu of "Policies".

Click the button "Create".



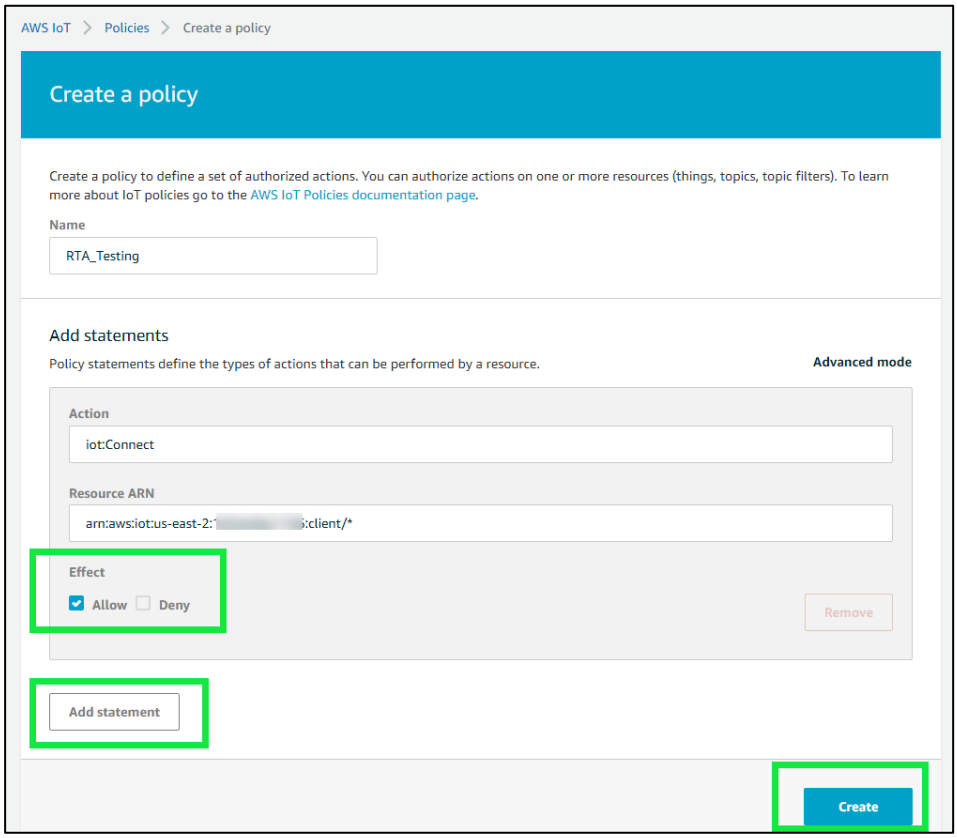


From the policy creation page, you add the statements that will dictate what connected devices are allowed to do. Assign a unique name to your policy and add four statements with the information listed below. Notice that when you type in the action, the field labeled "Resource ARN" will be automatically populated.
Check "Allow" under the "Effect" field and replace the last portion of each Resource ARN that reads, "replaceWithA", with an asterisk (*). When finished, you should have the following statements:

Action	Resource ARN
iot : Connect	arn:aws:iot:(your region):(your account #):client/*
iot : Publish	arn:aws:iot:(your region):(your account #):topic/*
iot : Receive	arn:aws:iot:(your region):(your account #):topic/*
iot : Subscribe	arn:aws:iot:(your region):(your account #):topicfilter/*

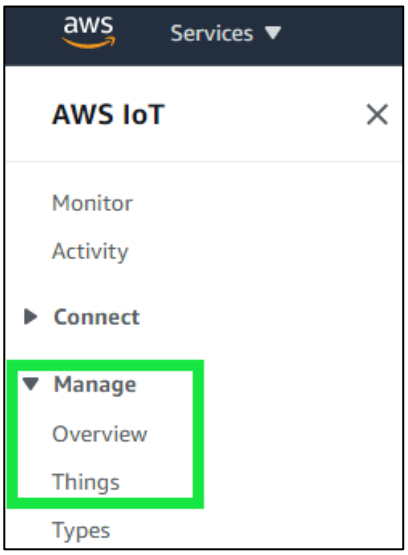
Click the **Add statement** button to create the Publish, Receive and Subscribe statements. Once completed click the **Create** button. Please note that in a production environment, you will want to be ***a lot*** more selective with your policy creation (e.g., don't use an asterisk at the end of a Resource ARN).

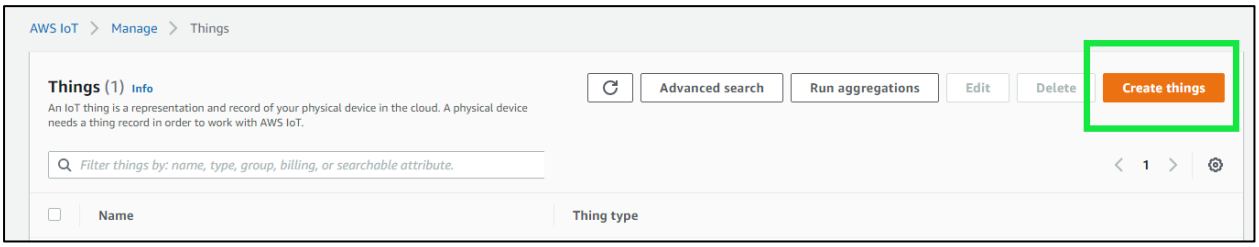
When they have been entered, click "Create". Now it's time to register our "thing".



AWS IoT Core Service Things Configuration

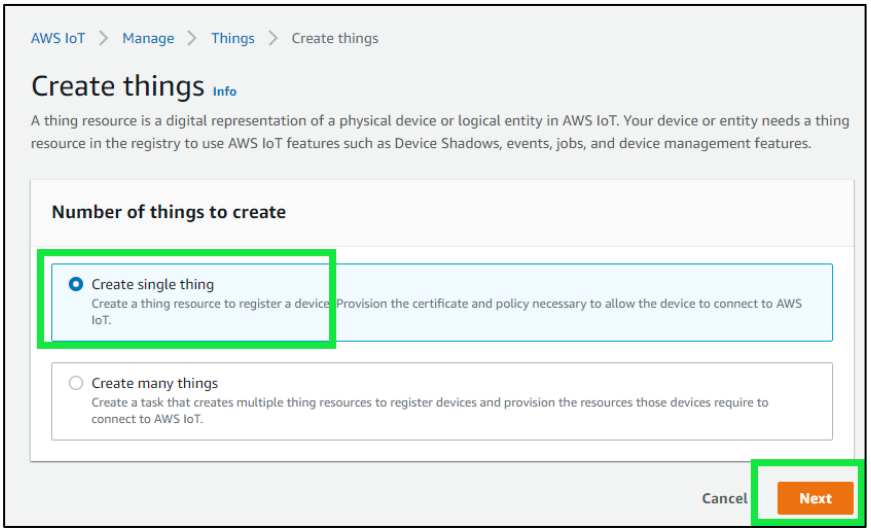
Back at the main menu on the left pane, click on the “Manage” menu option, and then the “Things” submenu.



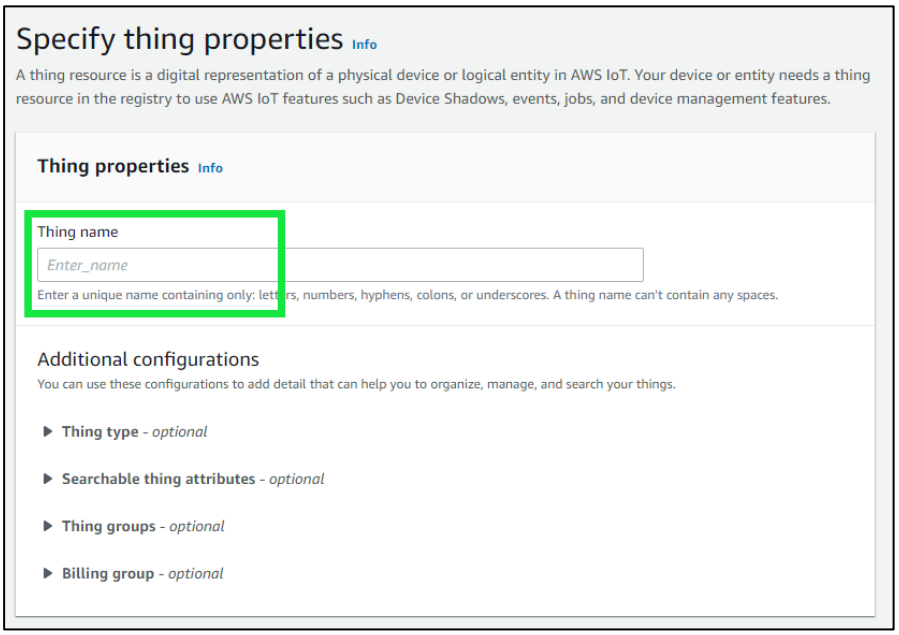


This will take us to a window that allows you to register a single “thing,” or multiple “things.” Click on the button labeled, “Create things.”

A new window will open with a number of things to create, chose “Create single thing” and click the Next button. If you have multiple RTA gateways, then you’ll need to select “Create many things”.

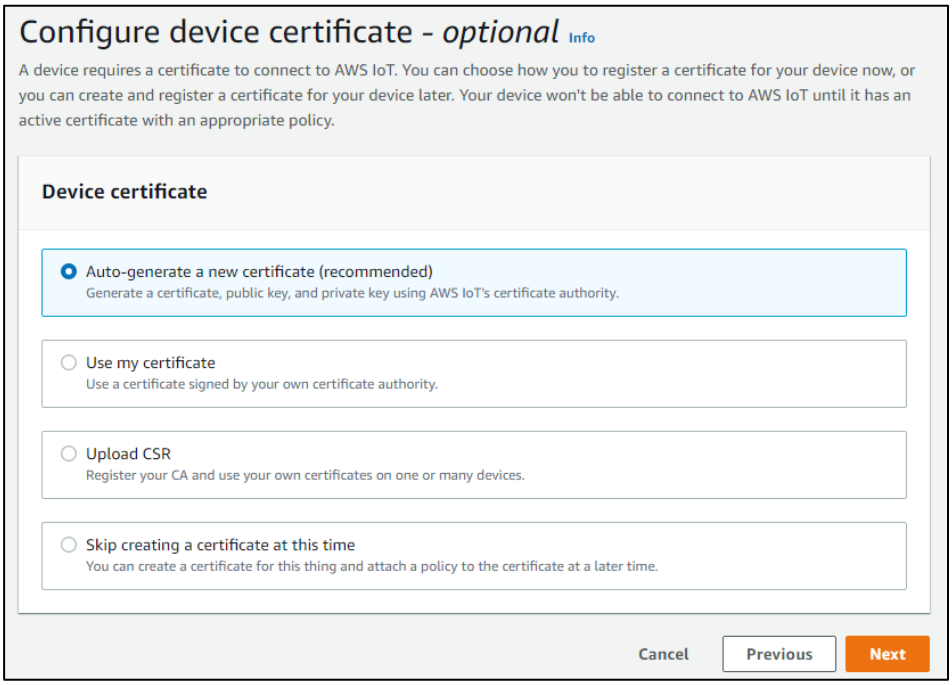


The next setting will be the “Specify thing properties”, here you will give your “Thing” a unique name and click the Next button at the bottom.



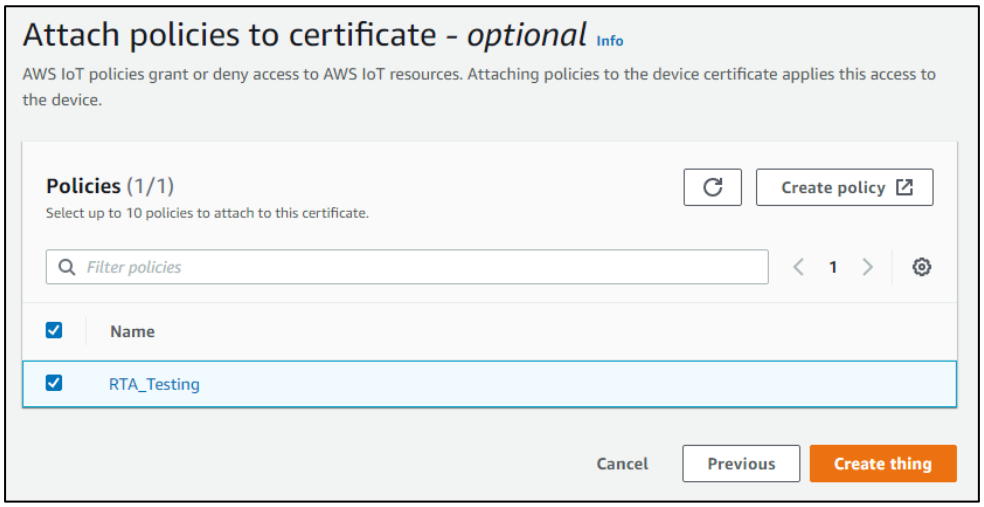
Certificate setup

Here you associate your “Thing” with the certificate that will be used to authenticate it with the AWS IoT Core service. Auto-generate is fine, click the Next button.

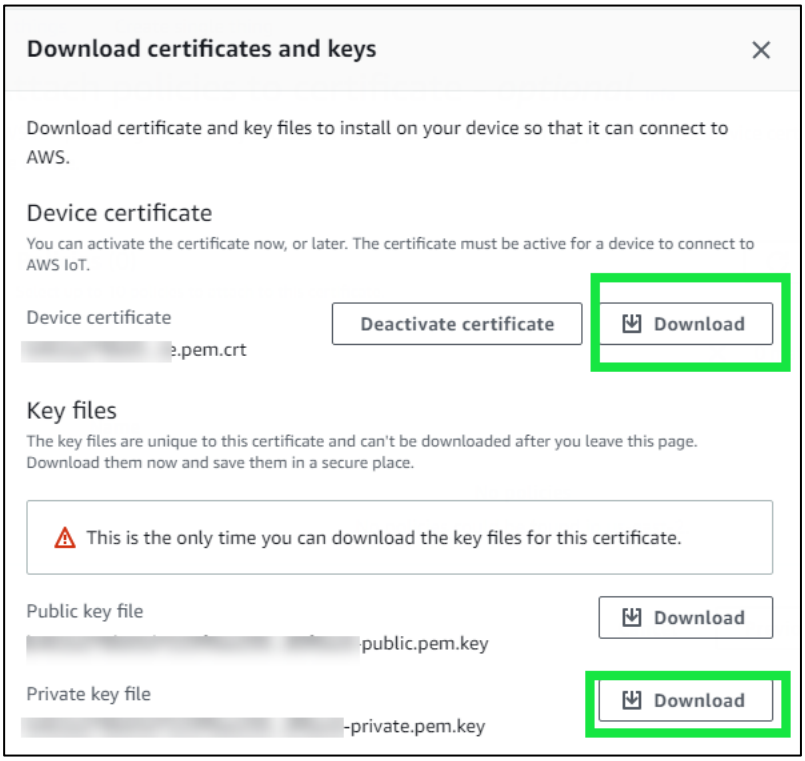

Configure device certificate - optional Info
A device requires a certificate to connect to AWS IoT. You can choose how you to register a certificate for your device now, or you can create and register a certificate for your device later. Your device won't be able to connect to AWS IoT until it has an active certificate with an appropriate policy.
Device certificate
Auto-generate a new certificate (recommended)
Generate a certificate, public key, and private key using AWS IoT's certificate authority.
Use my certificate
Use a certificate signed by your own certificate authority.
Upload CSR
Register your CA and use your own certificates on one or many devices.
Skip creating a certificate at this time
You can create a certificate for this thing and attach a policy to the certificate at a later time.
Cancel
Previous
Next

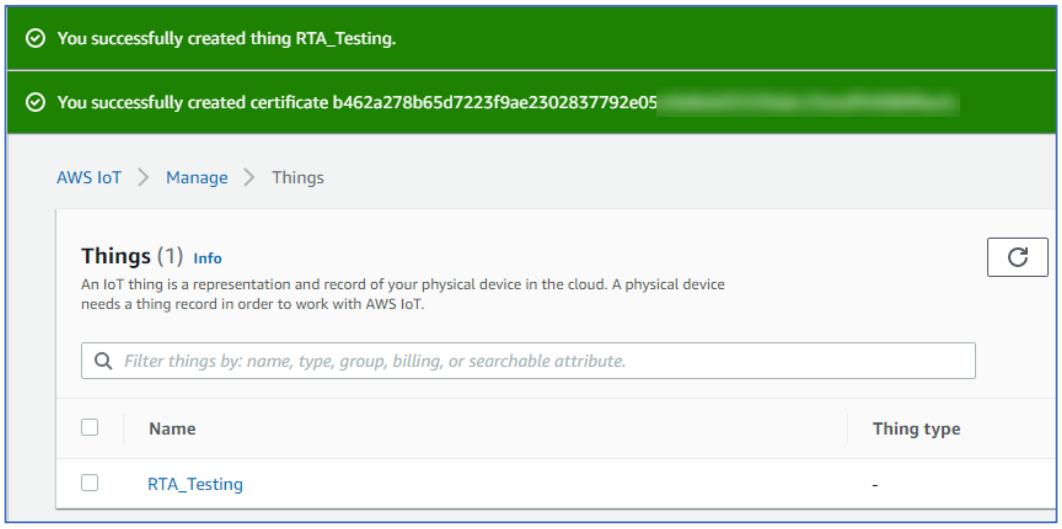
Attach policies to certificate

Next you'll see the policy you created previously, select the policy and click "Create thing" a pop up will appear to "Download certificates and keys".

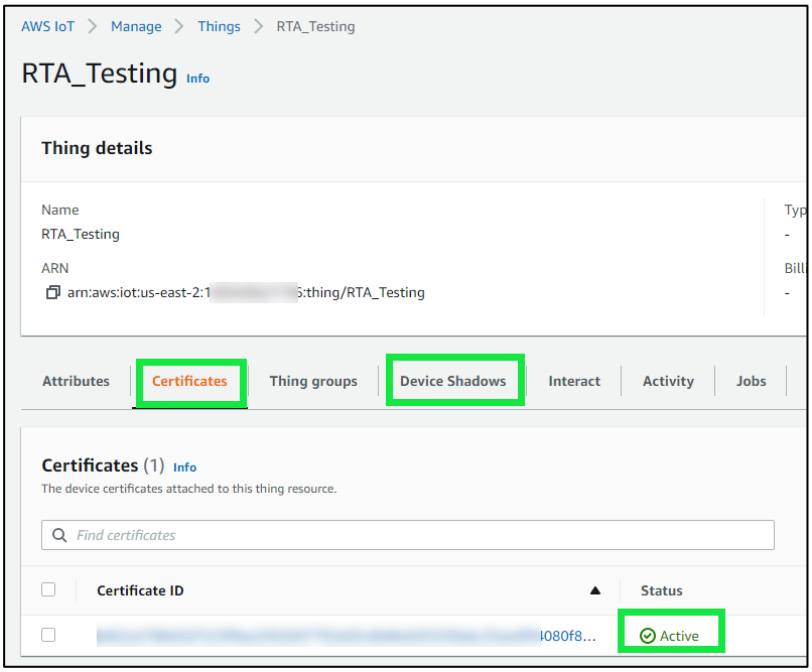


Download the certificate and the private key. Once downloaded, navigate back to this user guide section "how to FTP files into the RTA gateway" to load the certificate and private key into the gateway.

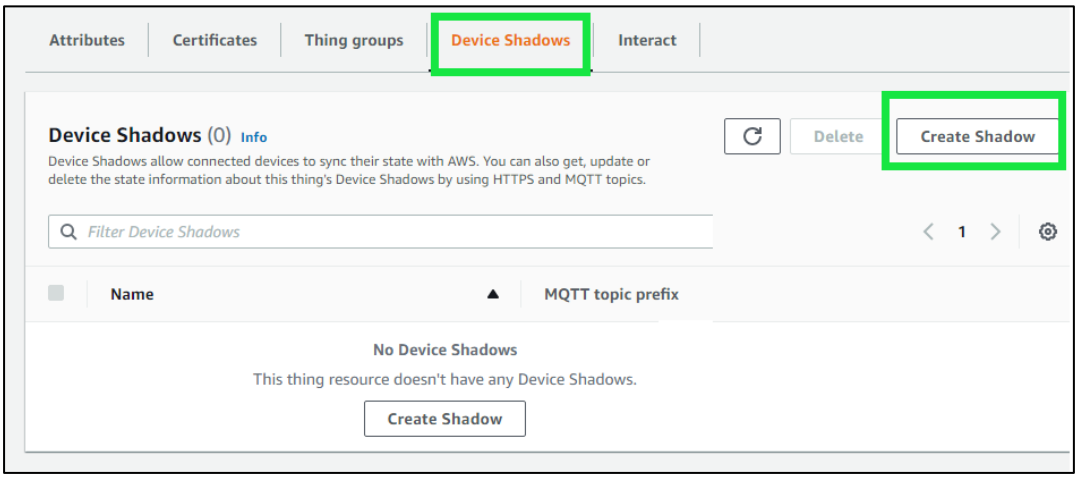




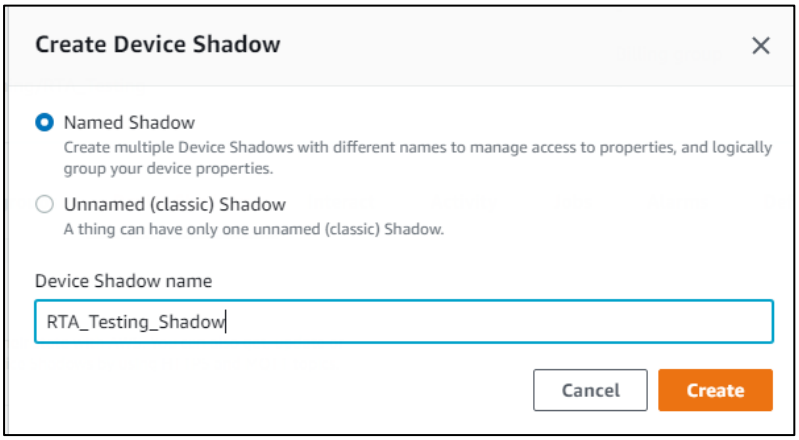
Once you have successfully downloaded the files you will be redirected to the Things page. Within the Things page, click on the thing name you setup, in this example it would be RTA_Testing. From this page, you can view if the certificate is active and create a Device Shadow URL.



Along with the certificate and private key, your RTA product will need the Device Shadow URL. Click the Device Shadows tab and click “Create Shadow”.

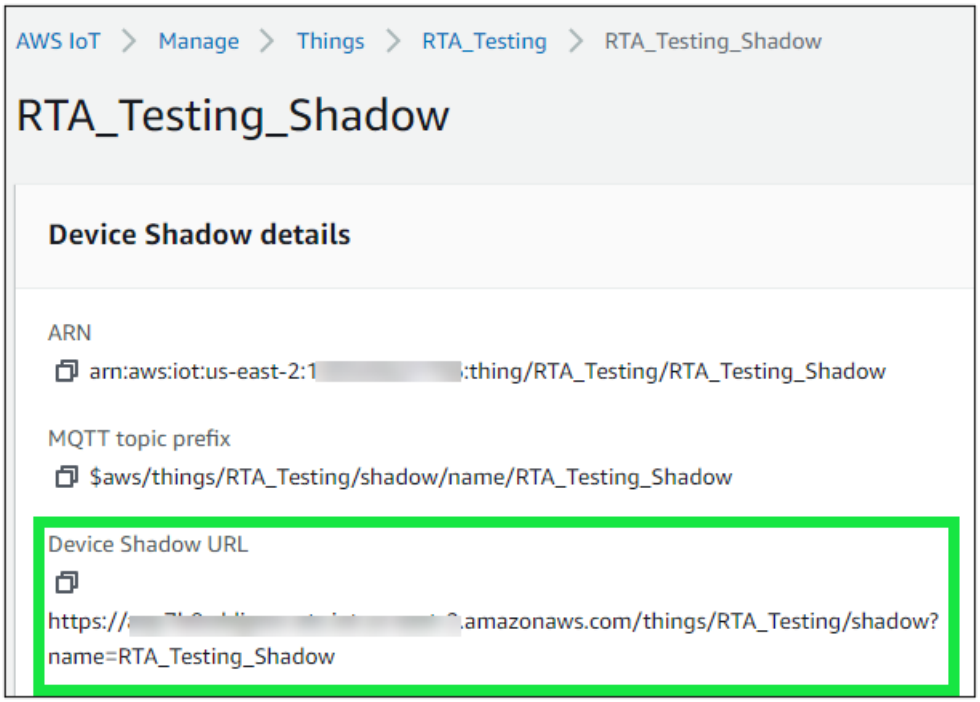


Enter in a Device Shadow name and click the Create button.

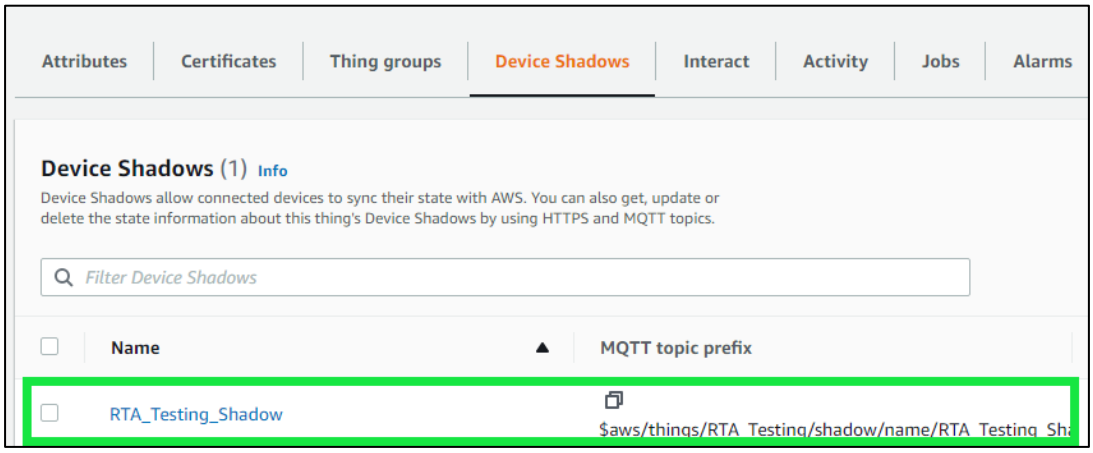


You will be redirected to the Things page where you’ll see your new Device Shadow created.

Click the Device Shadow name, in this case our example “RTA_Testing_Shadow” and it will display the details. You only want to copy the Device Shadow URL. Ignore the “https://” and everything after the “.com”

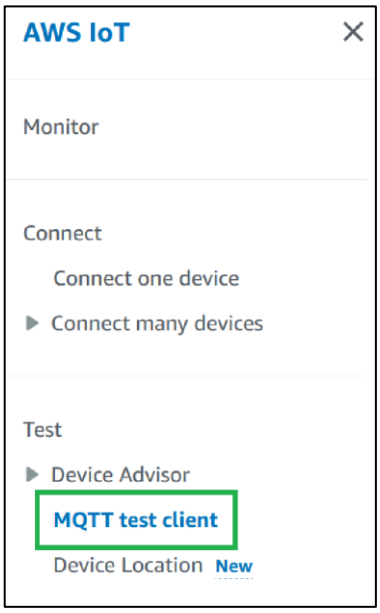


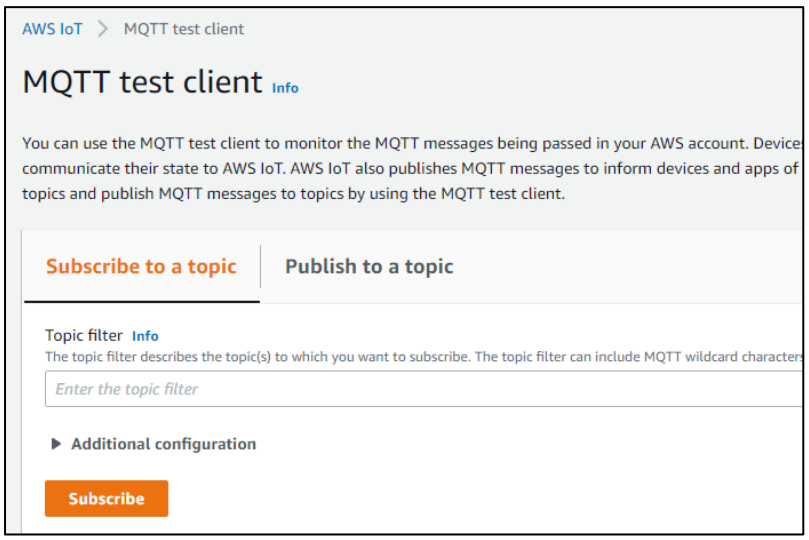
Within the RTA gateway configuration Device Shadow URL, enter in "aabb11224e9ex-ats.iot.us-east-2.amazonaws.com," everything else is ignored.



Testing AWS Communication

Once you have the AWS IoT Core service configured, you can use their “MQTT test client” feature to Publish a topic to the RTA gateway and view published data.



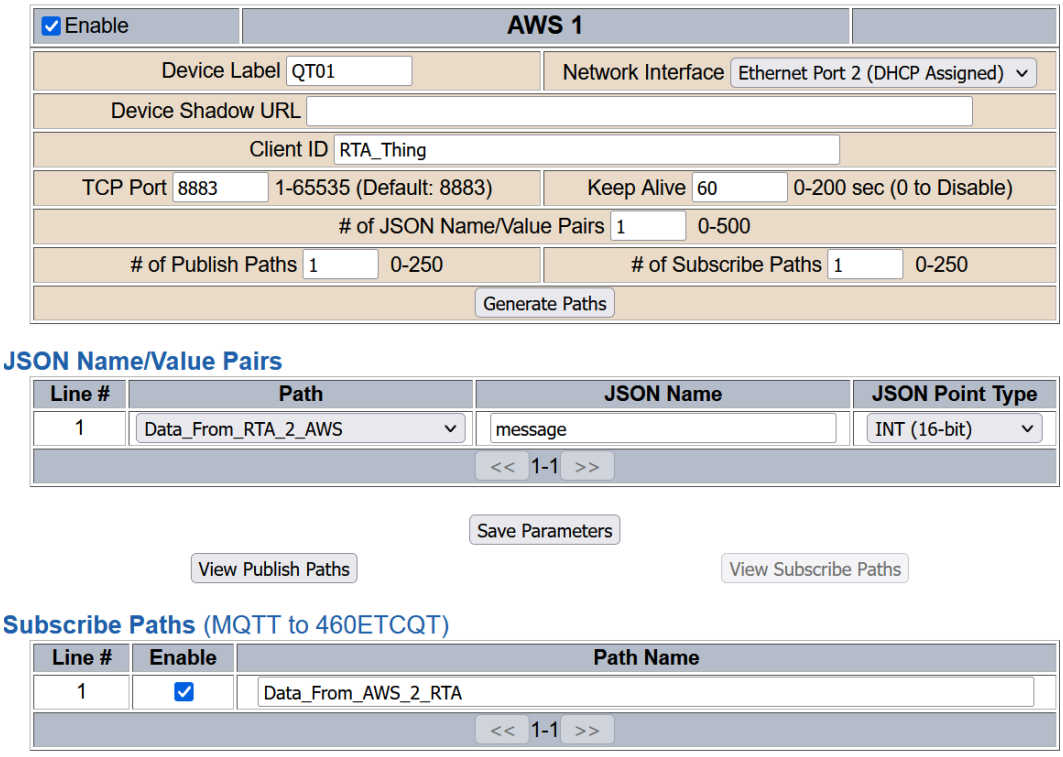


Note: This is assumed the certificate, private key and Device Shadow URL have already been configured in AWS, the two files have been FTP’d into the RTA gateway, and the Device Shadow URL is configured.

Using the AWS MQTT test client, you can Subscribe to a topic (data from the RTA), and you can Publish to a topic (data to the RTA).

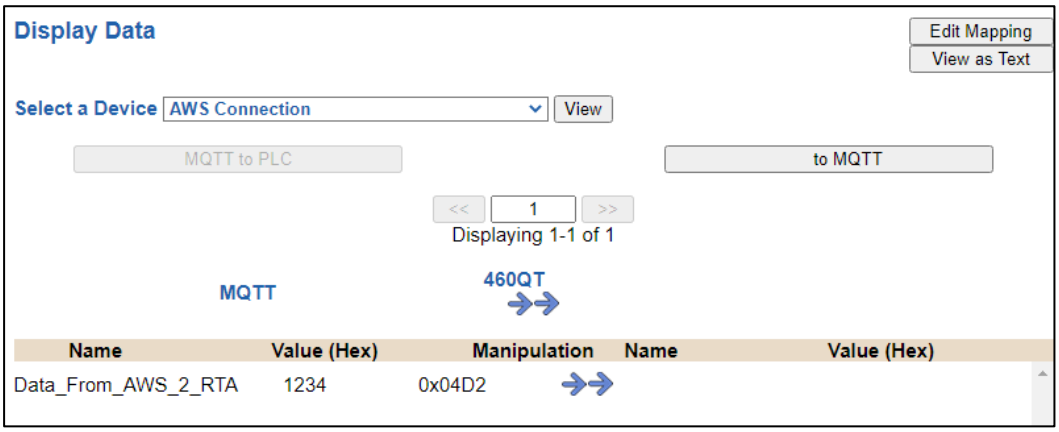
Send data from AWS to RTA gateway (Subscribe Topic)

Below is how the RTA AWS IoT Core Service is setup to Subscribe data from AWS to the RTA.



Within AWS, click the “Publish to a topic” tab. Enter in the topic name that is defined in the RTA gateway “Subscribe Topics” configuration. In the Message payload, after the “:” enter in your value, if using a string be sure your data is in “ “ . For example, “message”: 1234 or “message”: “Hello World.” Once you have your data, click the Publish button.

Navigate to the RTA Display data and refresh the web page. You will see your data being updated.



Send data from RTA gateway to AWS (Publish Topics)

This example shows a PLC writing data to the RTA gateway and presenting that data to the Publish topic.

Enable	AWS 1	
Device Label QT01	Network Interface Ethernet Port 2 (DHCP Assigned)	
Device Shadow URL		
Client ID RTA_Thing		
TCP Port 8883 1-65535 (Default: 8883)	Keep Alive 60 0-200 sec (0 to Disable)	
# of JSON Name/Value Pairs 1 0-500		
# of Publish Paths 1 0-250	# of Subscribe Paths 1 0-250	
Generate Paths		

JSON Name/Value Pairs

Line #	Path	JSON Name	JSON Point Type
1	Data_From_RTA_2_AWS	Data	INT (16-bit)
<< 1-1 >>			

Save Parameters

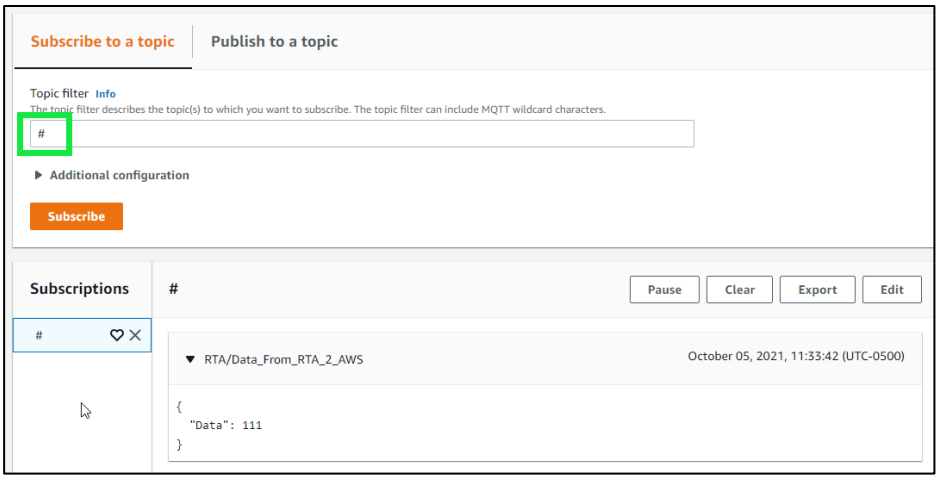
View Publish Paths View Subscribe Paths

Publish Paths (460ETCQT to MQTT)

Line #	Enable	Path Name
1	☑	Data_From_RTA_2_AWS
<< 1-1 >>		

PLC			460ETCQT → →	MQTT		
Name	Value (Hex)		Manipulation	Name	Value (Hex)	
PLC_Data_2_AWS	111	0x006F	→ →	QT01 Data_From_RTA_2_AWS	111	0x00

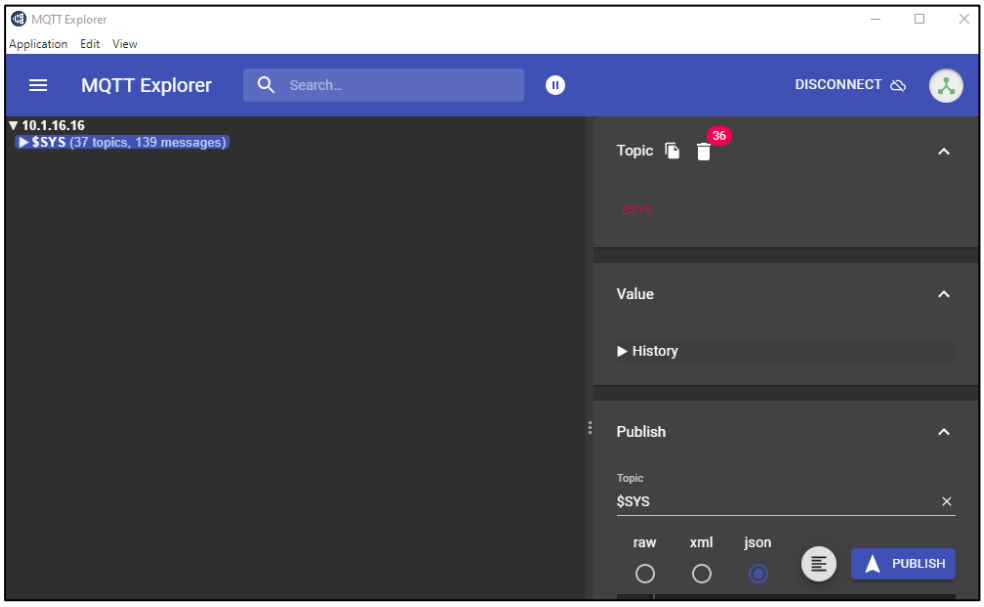
In the topic filter, use a wildcard character of "#" (subscribe to all topics), and click the Subscribe button. You'll see the subscription once the new Publish data comes in. The "RTA" is the Client ID that is configured in the MQTT device configuration of the RTA. The "Data_From_RTA_2_AWS" is the Publish Topic name configured in the AWS device configuration page of the RTA.


Subscribe to a topic
Publish to a topic
Topic filter Info
The topic filter describes the topic(s) to which you want to subscribe. The topic filter can include MQTT wildcard characters.
#
Additional configuration
Subscribe
Subscriptions
#
Pause
Clear
Export
Edit
#
RTA/Data_From_RTA_2_AWS
October 05, 2021, 11:33:42 (UTC-0500)
{
"Data": 111
}

Testing Your MQTT Connections with MQTT Explorer

For this test example, we are going to be using MQTT Explorer (https://mqtt-explorer.com/) which can be downloaded for free. This tool can monitor MQTT client/broker relationships.

Once you launch the MQTT Explorer tool, setup a new connection. The host will be IP of your MQTT broker.



Next, you can configure the RTA gateway. The broker IP address listed below is set up to be the IP of the machine where the broker is installed.

Enable	MQTT 1	
Device Label QT01	Network Interface Ethernet Port 1 (192.168.1.26)	
Broker IP / URL 10.1.16.16		
Client ID RTA_Thing		
TCP Port 1883 1-65535 (Default: 1883)	Keep Alive 60 0-200 sec (0 to Disable)	
Username	Password	
0-256 characters	# of JSON Name/Value Pairs 2 0-500	
# of Publish Paths 1 0-250	# of Subscribe Paths 1 0-250	
Generate Paths		

Send data from RTA gateway to MQTT Explorer (Publish Topic)

JSON Name/Value Pairs

Line #	Path	JSON Name	JSON Point Type
1	PLC_Data_2_Explorer	Data	INT (8-bit)
	<< 1-1 >>		

Save Parameters

View Publish Paths | View Subscribe Paths

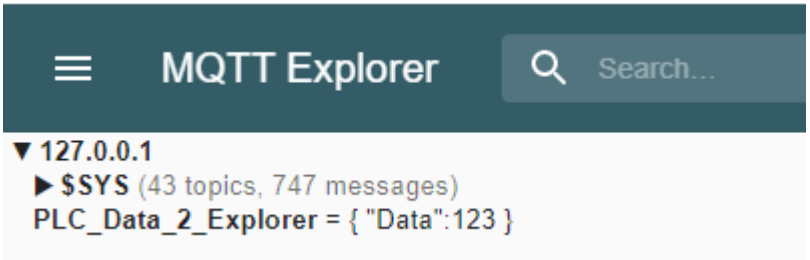
Subscribe Paths (MQTT to 460ETCQT)

Line #	Enable	Path Name
1	☑	PLC_Data_2_Explorer
	<< 1-1 >>	

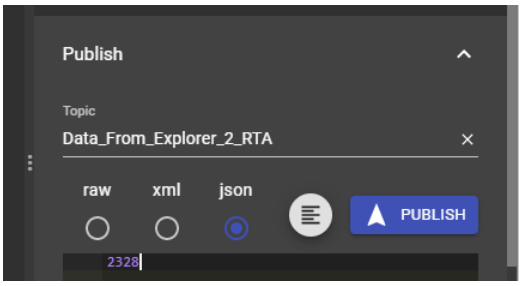
PLC			460ETCQT →	MQTT	
Name	**Value (Hex)**	**Manipulation**		**Name**	**Value (Hex)**
PLC_Data_2_Explorer	123	0x007B	→	PLC_Data_2_Explorer Data	123

This example shows a PLC writing data to the RTA gateway and presenting that data to the MQTT publish topic.

Write a value in “my PLC” and MQTT Explorer will subscribe to that topic.



Send data from MQTT Explorer to RTA gateway (Subscribe Topic)



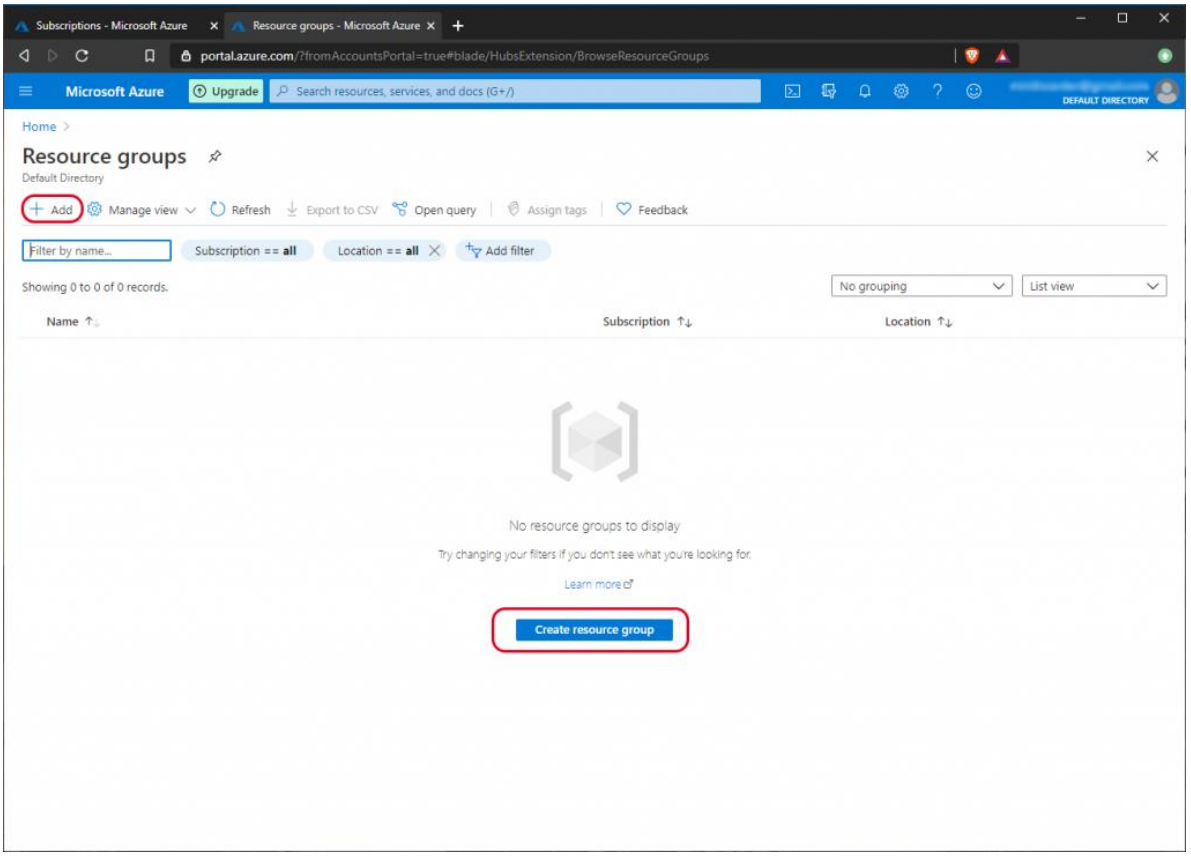
The RTA gateway has a topic name of Data_From_Explorer_2_RTA that MQTT Explorer is going to be publishing to. Enter in the topic name to publish, enter in the value (our example is 2328), then click Publish.

PLC			460ETCQT	MQTT	
Name	**Value (Hex)**		**Manipulation**	**Name**	**Value (Hex)**
ETC01_G2N0_INT	2328	0x0918		QT01 Data_From_Explorer_2_RTA	2328

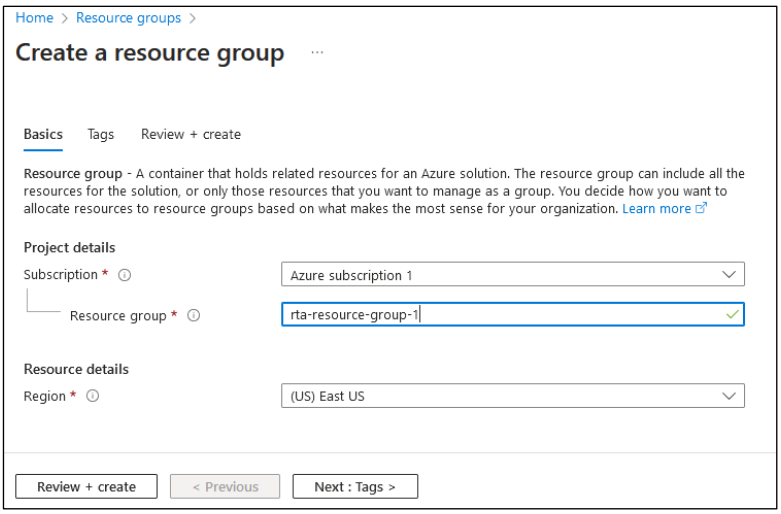
Microsoft Azure Service Setup

Please note this section outlines the bare minimum configuration to get the RTA gateway connected to Azure and is not necessarily a recommended configuration in a production environment.

1. Create a Microsoft Azure subscription if you do not have one already
2. Create a resource group
    a. In the search bar at the top type in “Resource Groups” and select the resource groups under services to navigate to the resource groups page.
    b. Hit Add in the top left corner, there may also be a create resource group button in the middle if no resource groups exist.

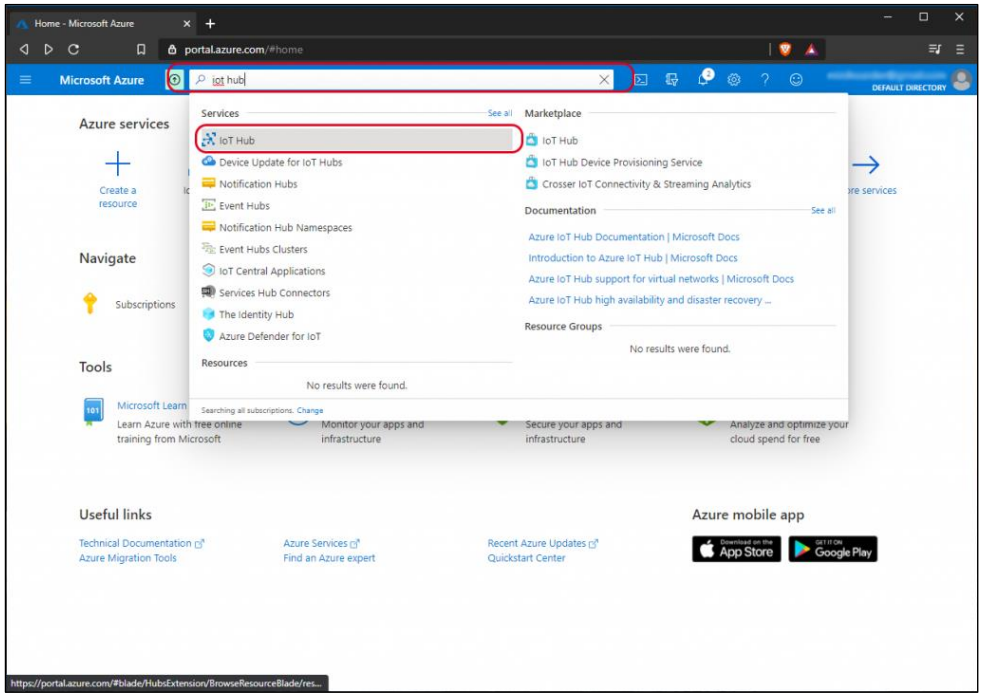


    c. In the create a resource group window give the resource group a name, the available Azure subscription should be selected by default.



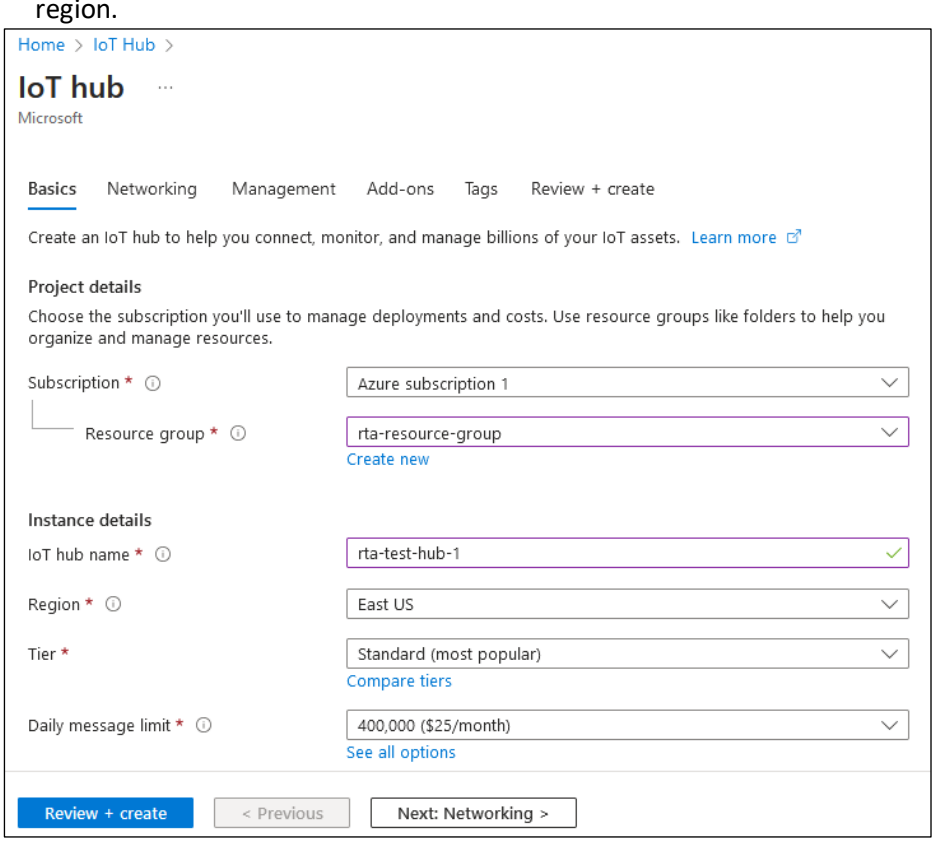
d. Hit Review + Create in the bottom left corner.
e. In the review window, hit create in the bottom left corner to create the resource group.
f. You should be re-directed back to the resource groups window where you can see the newly created resource group is listed.

3. Create an IoT Hub
   a. In the search bar at the top type in “IoT Hub” and select the IoT under services to navigate to the IoT Hub page.



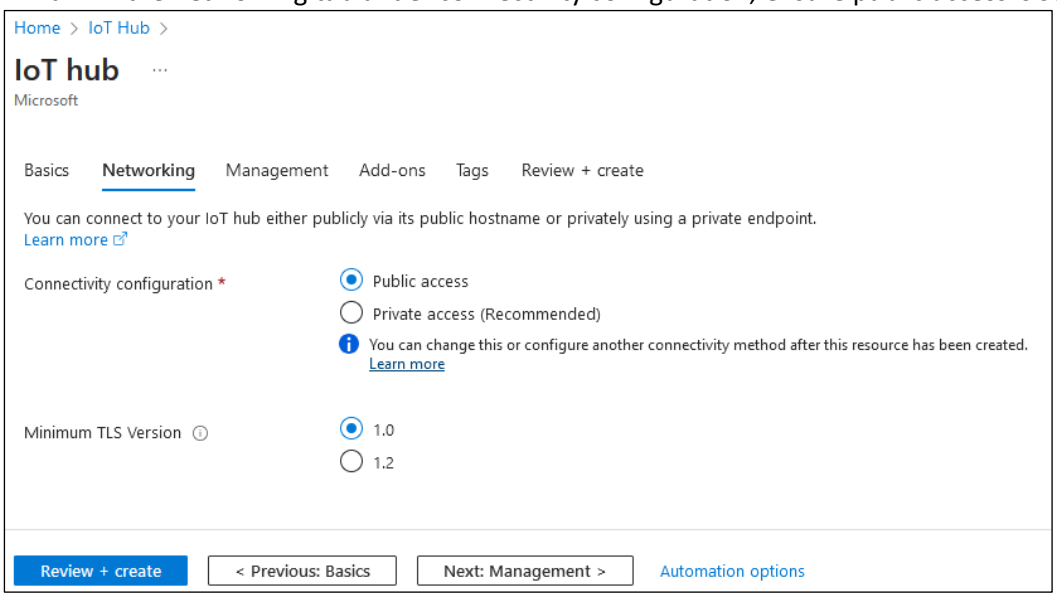
   a. Hit Add in the top left corner, there may also be a create IoT Hub button in the middle if no IoT Hubs exist.

b. Select the resource group created previously, enter a name for the hub, and select a region.



c. Hit Next at the bottom of the page to proceed to the networking tab.

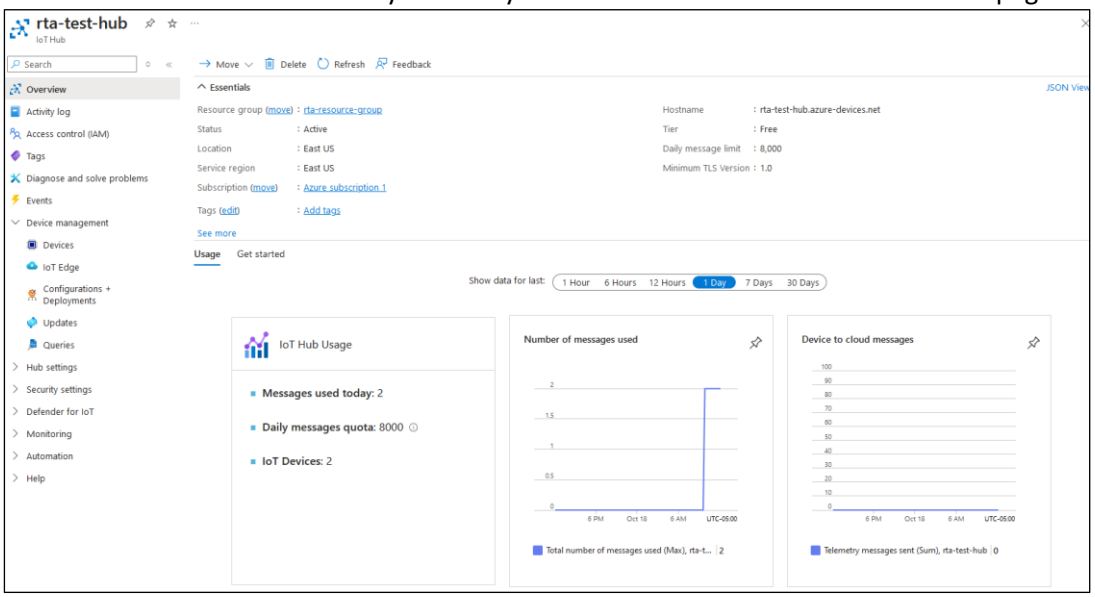
d. In the networking tab under connectivity configuration, ensure public access is selected.



e. Hit review and create in the bottom left corner of the page.

f. On the review and create page, hit create in the bottom left corner of the page.

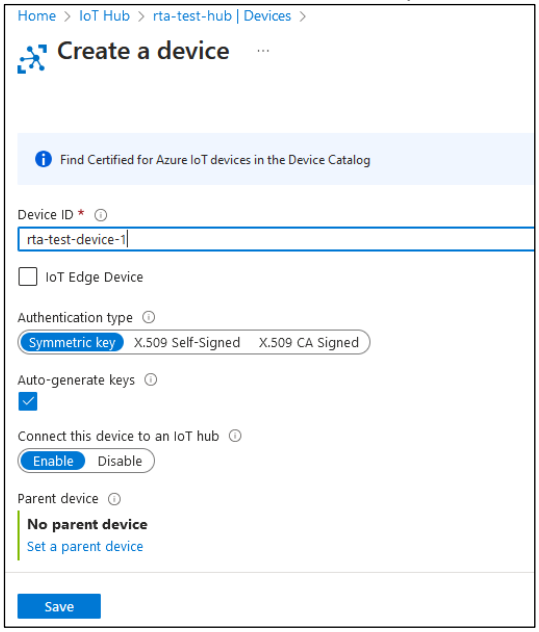
g. One the deployment is completed click on “Go to Resource” to be re-directed to the newly created IoT Hubs overview page. If go to resource is not an option, navigate to the IoT hub and select your newly created IoT hub to access the overview page.



h. In the left panel of the IoT Hub overview select Devices under the Device Management section.

i. Click Add Device button in the top left corner of the page to create a new IoT device in this hub.

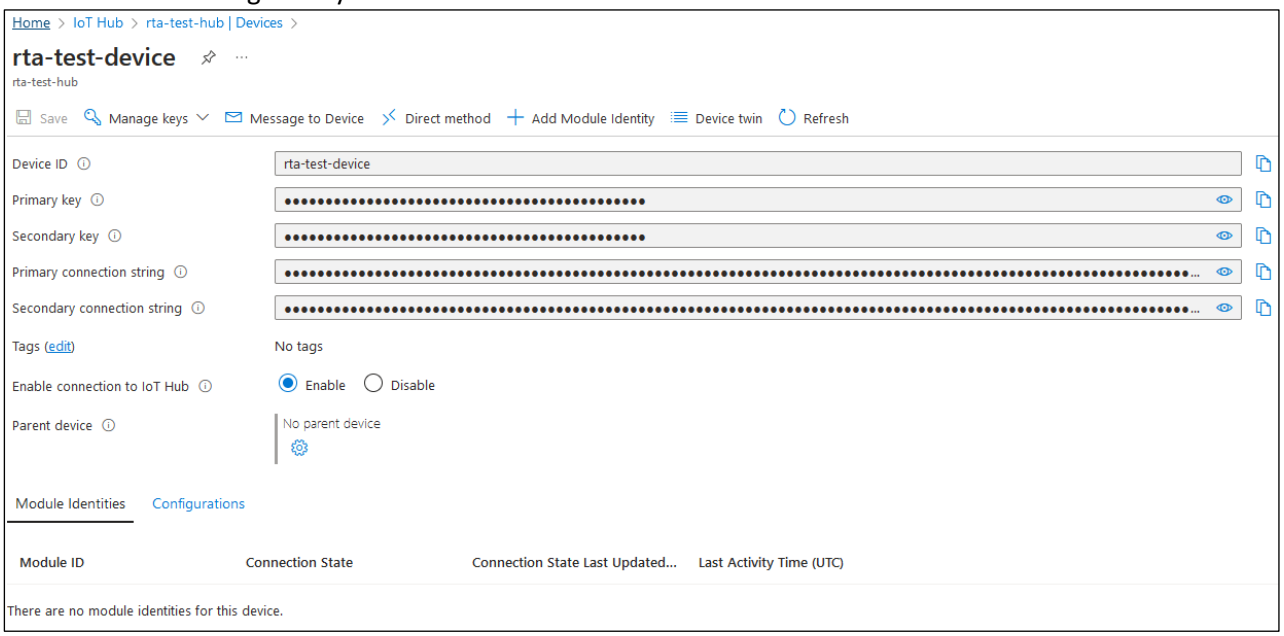
j. In the create a device window enter a name for the device in the Device ID section and ensure the “Connect his device to an IoT hub” option is enabled.



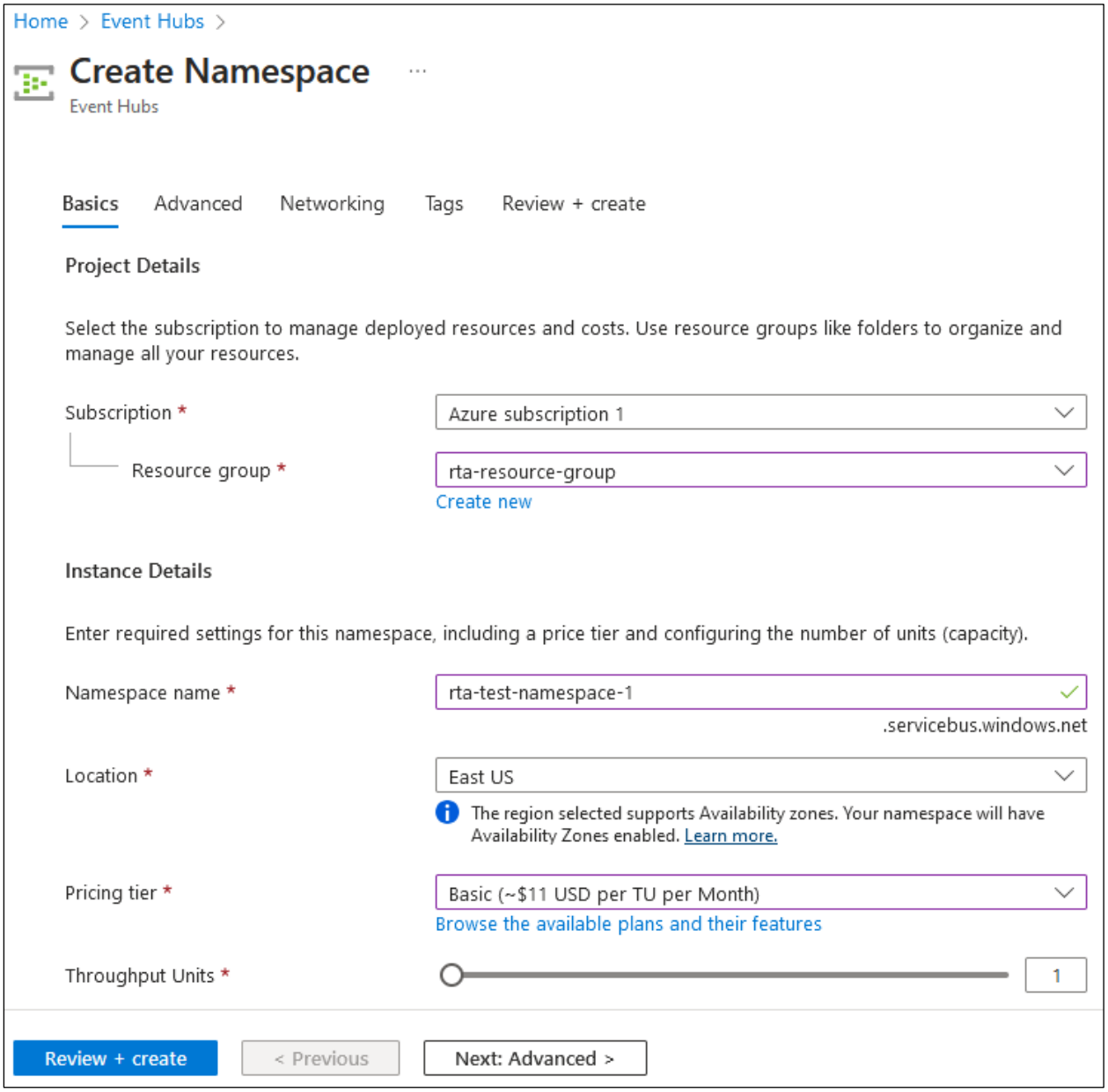
k. Hit Save in the bottom left corner of the window to create the device and be re-directed to the device list in the IoT Hubs overview window.

l. Click on the newly created device in the device list to view the devices configuration.

m. Copy the Primary connection string for the device, this will be used when connecting the RTA gateway to Azure.

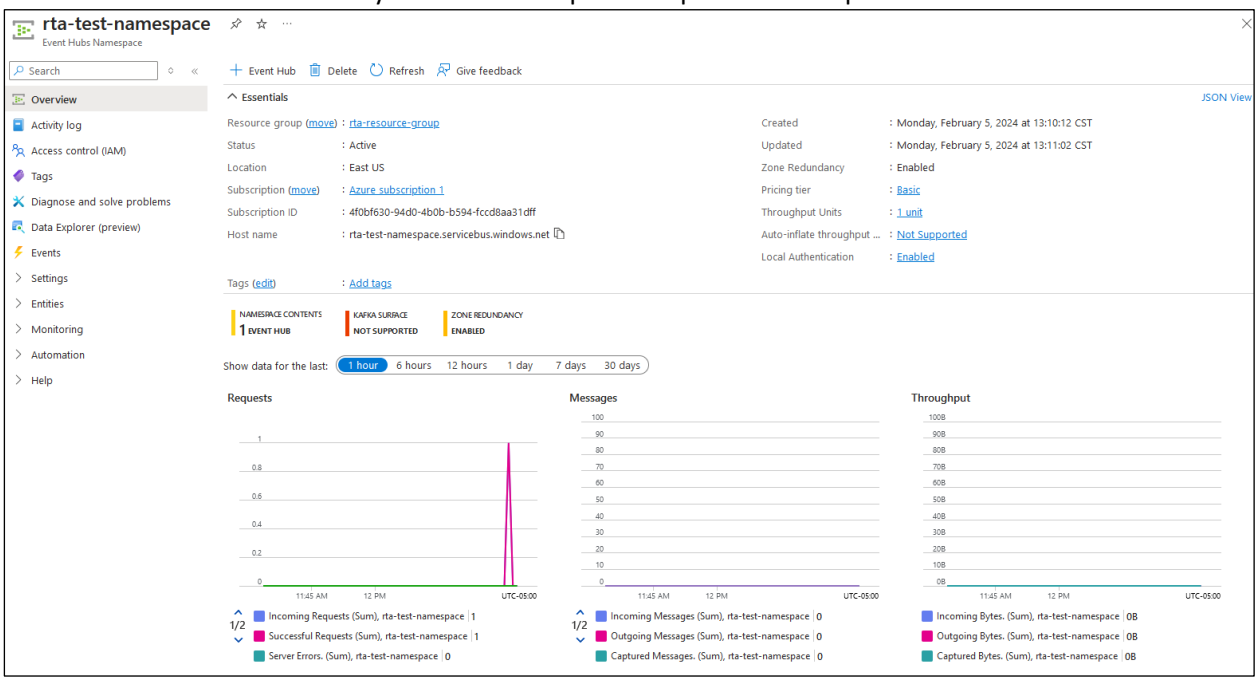


4. Create an Event Hub
    a. In the search bar at the top type in “Event Hubs” and select the Event Hubs under services to navigate to the Event Hubs page.
    b. Hit Add in the top left corner, there may also be a create Event Hub button in the middle if no Event Hubs Namespaces exist.
    c. In the Create Namespace window, select your previously created resource group, enter a name for the namespace, and select a pricing tier based on your needs.

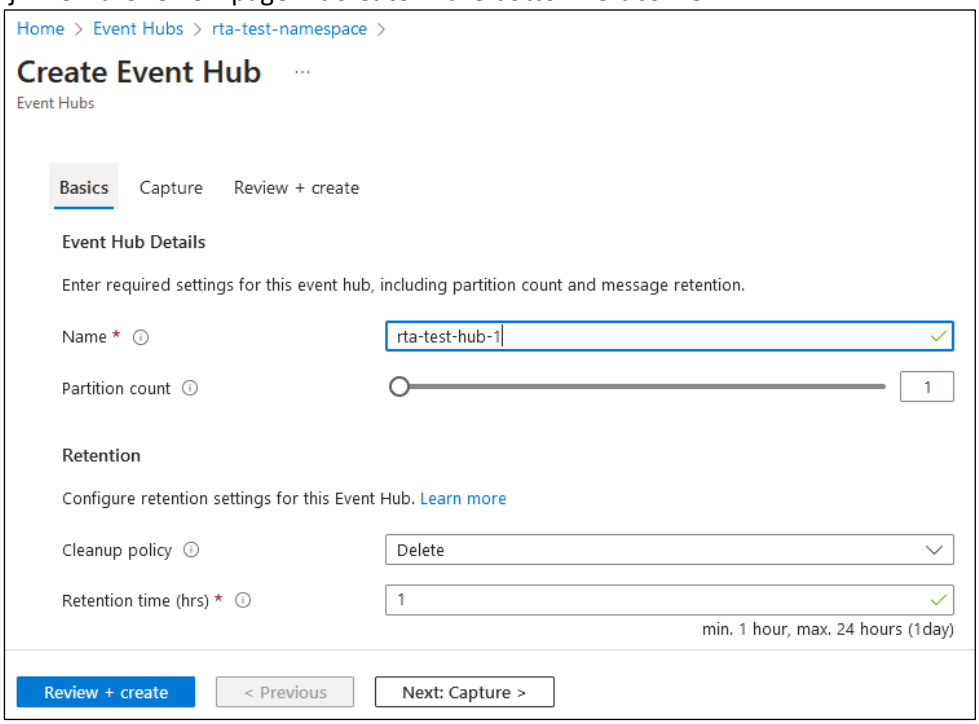


    d. Hit Review + Create in the bottom left corner of the page.
    e. Hit Create in the bottom left corner of the page to create the namespace.

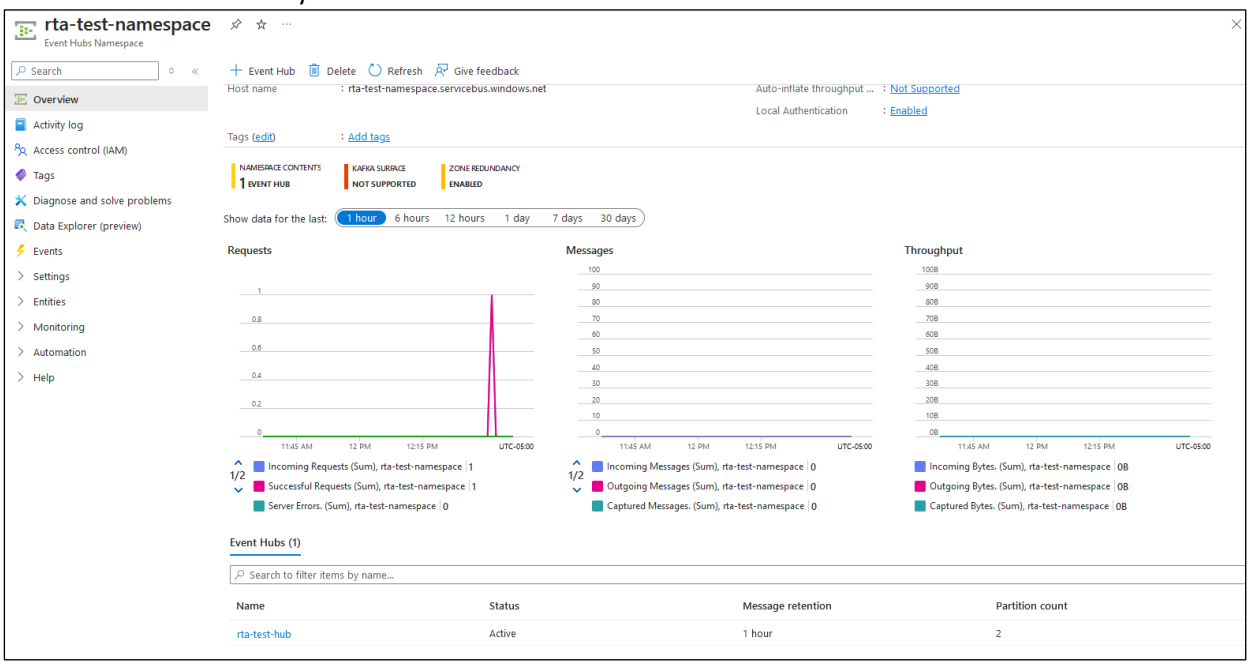
f. Once the namespace has finished initializing click on go to resource to navigate to the namespaces main panel. If go to resource in not an option navigate to Event Hubs and select the newly created namespace to open the main panel.



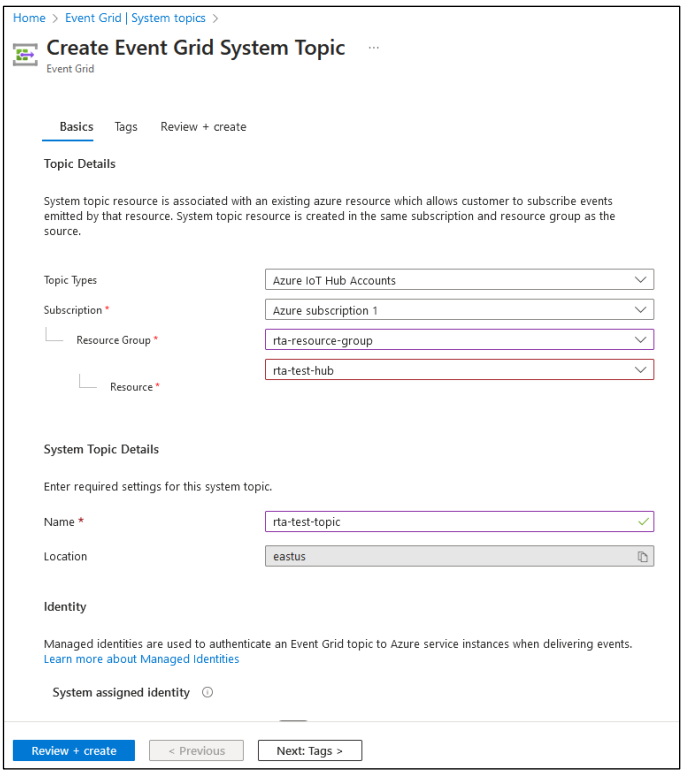
g. Click Add Event Hub in the top left corner.
h. Enter in a name for the event hub
i. Click review + create in the bottom left corner.
j. On the review page hit create in the bottom left corner.



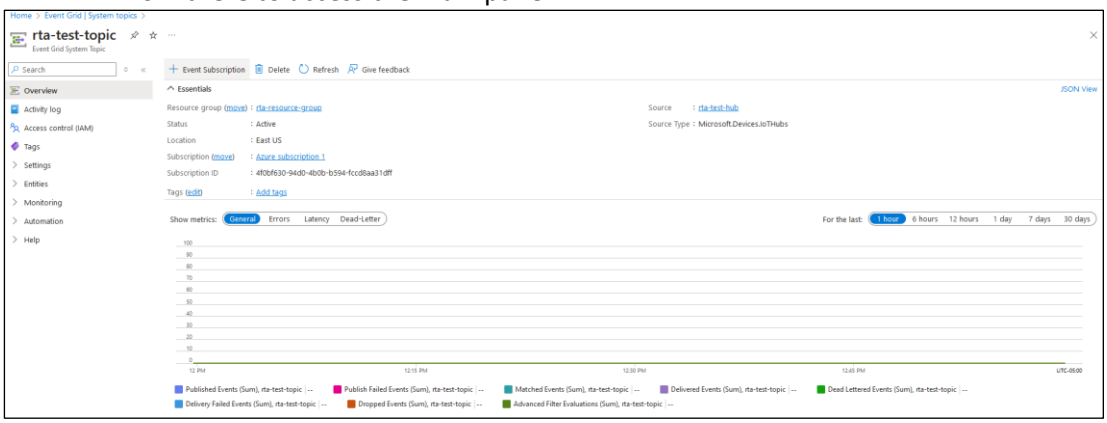
k. You should be returned to the namespace overview, scroll down and you should see the newly created event hub at the bottom of the overview section.



5. Create an Event Grid System Topic
   a. In the search bar at the top type in “Event Grid System Topics” and select the event grid system topics option under services to navigate to the even grid system topics page..
   b. Click create in the top left corner to create a new event grid system topic.
   c. In the Create Even Grid System Topic window enter set the Topic Types to “Azure IoT Hub Accounts, Select your azure subscription, select your resource group, and enter a name for the topic

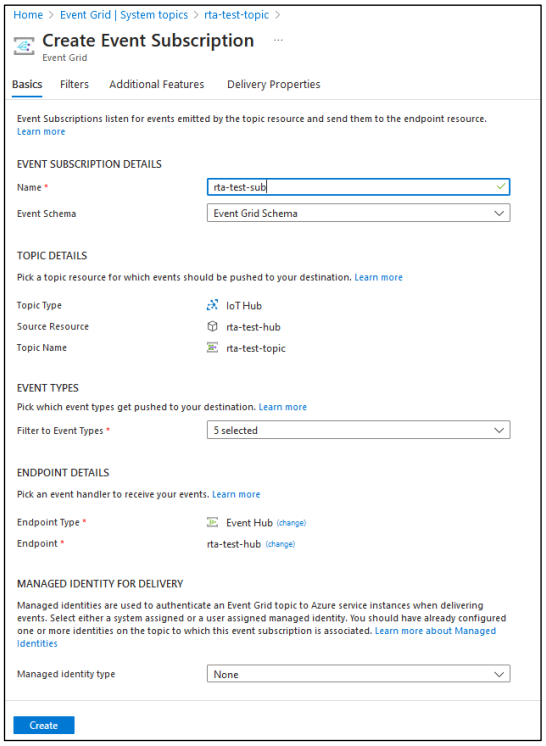


d. Click Review + create in the bottom left corner.
e. On the review + create page click create the create the Event Grid System Topic.
f. Once the event grid system topic is initialized click go to resource to navigate to the Event Grid System Topic Panel. If go to resource is not an option navigate back to the even grid system topic section and select the newly created event grid system topic from there to access the main panel.



g. In the your new event grid system topics main panel hit add event subscription in the top left corner.
h. In the create Event Subscription window:
    i. Enter a name for the subscription

ii. Under event types, ensure that Device Created, Device Deleted, Device Connect, Device Disconnect, and Device Telemetry are all selected.
iii. Under Endpoint Select Event Hub as the endpoint type.
iv. Under endpoint details click “Configure an Endpoint”
v. In the newly opened Select Event Hub window, select your azure subscription, resource group, namespace, and event hub and click confirm selection.



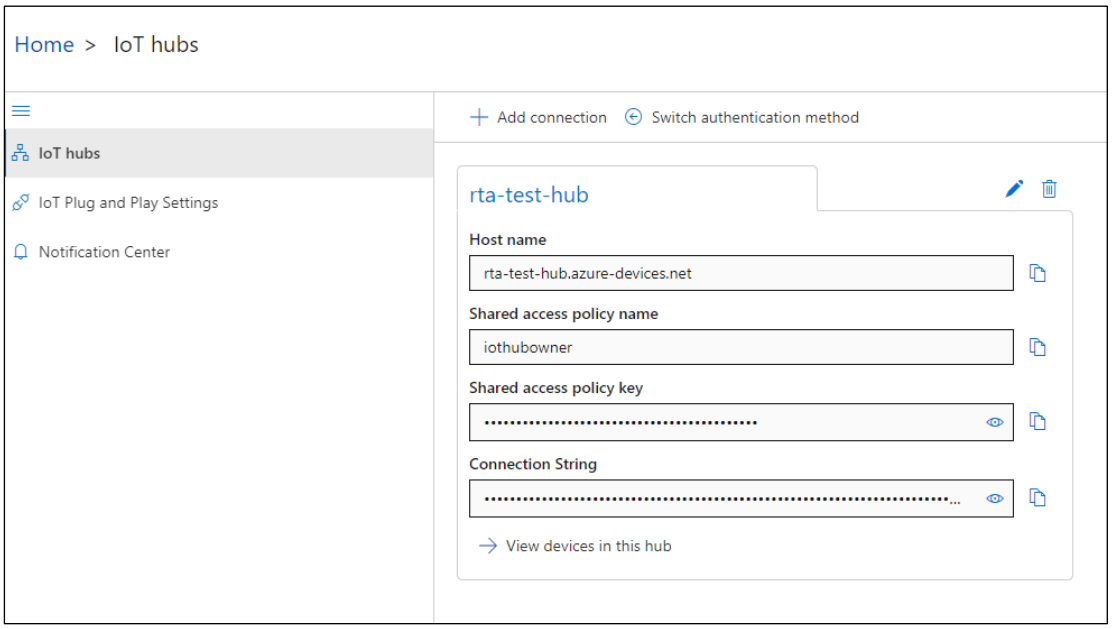
i. Click create in the lower left corner of the webpage.

Testing Microsoft Azure Communication

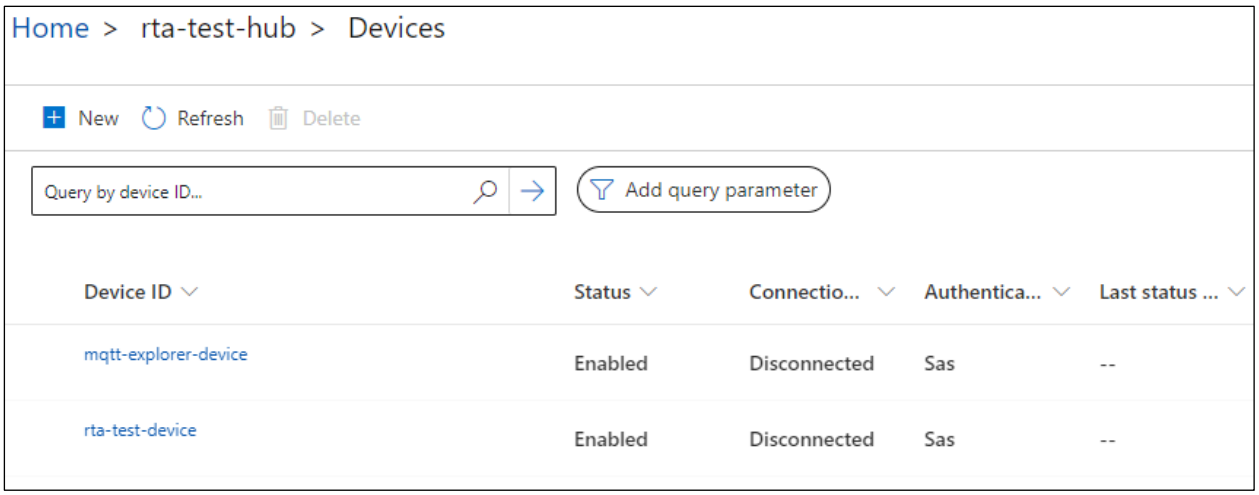
Once you have the Azure configured, you can use their Azure IoT Explorer utility to test the connection is working.

To set up Azure IoT Explorer:

1. open the utility and select the IoT Hubs tab on the left
2. Click the Add connection button
3. In the Add connection String window that appears enter the connection string for your IoT Hub.
    a. This can be obtained by navigating to the IoT Hubs overview window in azure, selecting Shared access policies from the left pane, and opening the iothubowner policy.
4. Click save.
5. Your hub should now be listed in the IoT hubs section as shown below.



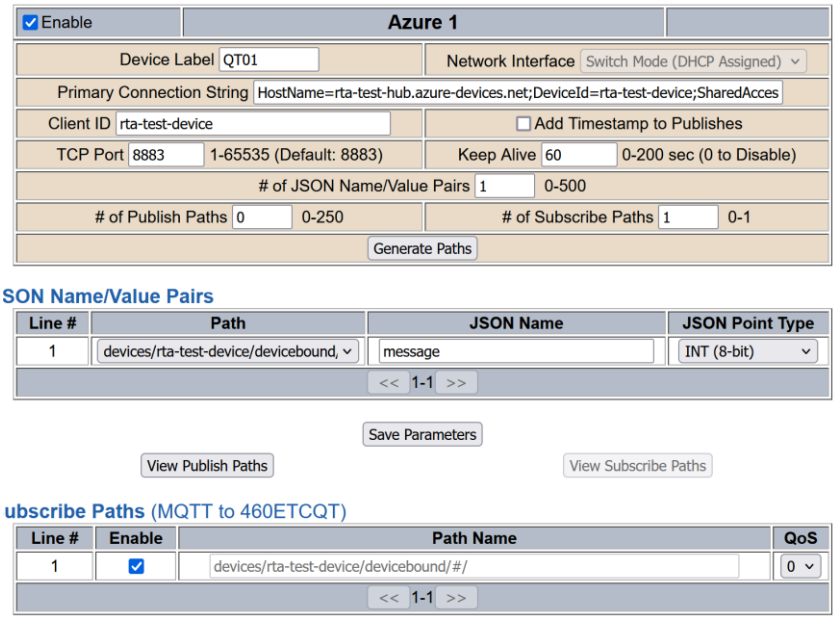
6. You can then click on the hub name to view devices available in the hub and see their current status.



7. Selecting a device within the hub will provide device specific information such as its Primary Connection string. It also provides useful tools for seeing data being from the device to Azure as well as send test messages from Azure to the device.

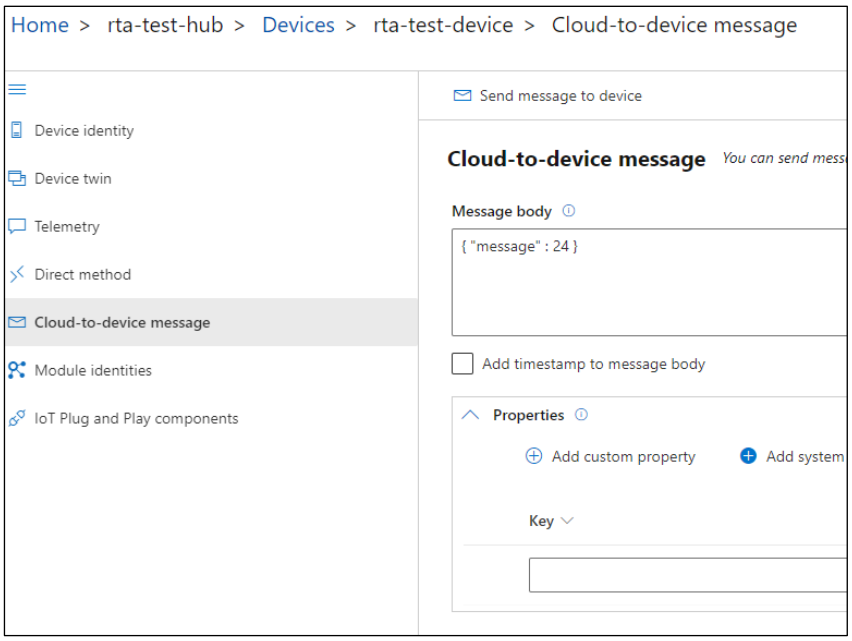
Send data from Microsoft Azure to RTA gateway

Below is how the RTA Microsoft Azure Service is set up to receive data from Microsoft Azure.

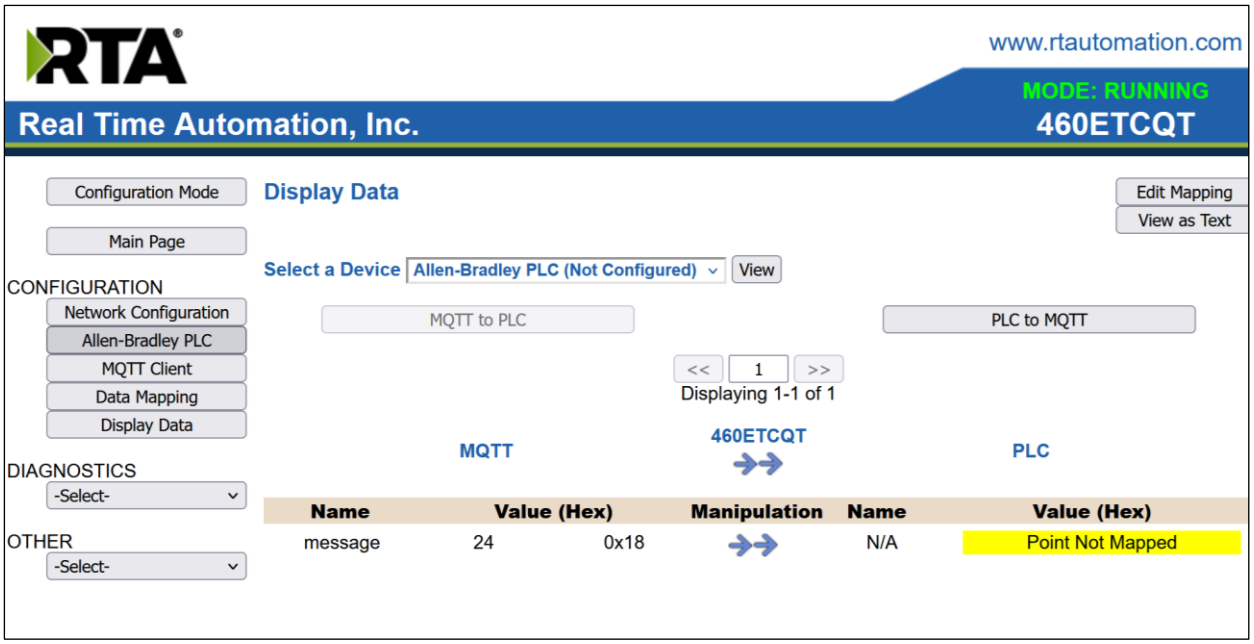


In Azure IoT Explorer, navigate to the device you wish to receive data from Azure on. In the device configuration select the Cloud-to-device message tab on the left.

In the Cloud-to-device message tab, enter the message to be sent to the RTA in the message body section and click Send message to device at the top of the page. An example payload for the configuration above can be seen in the image below.

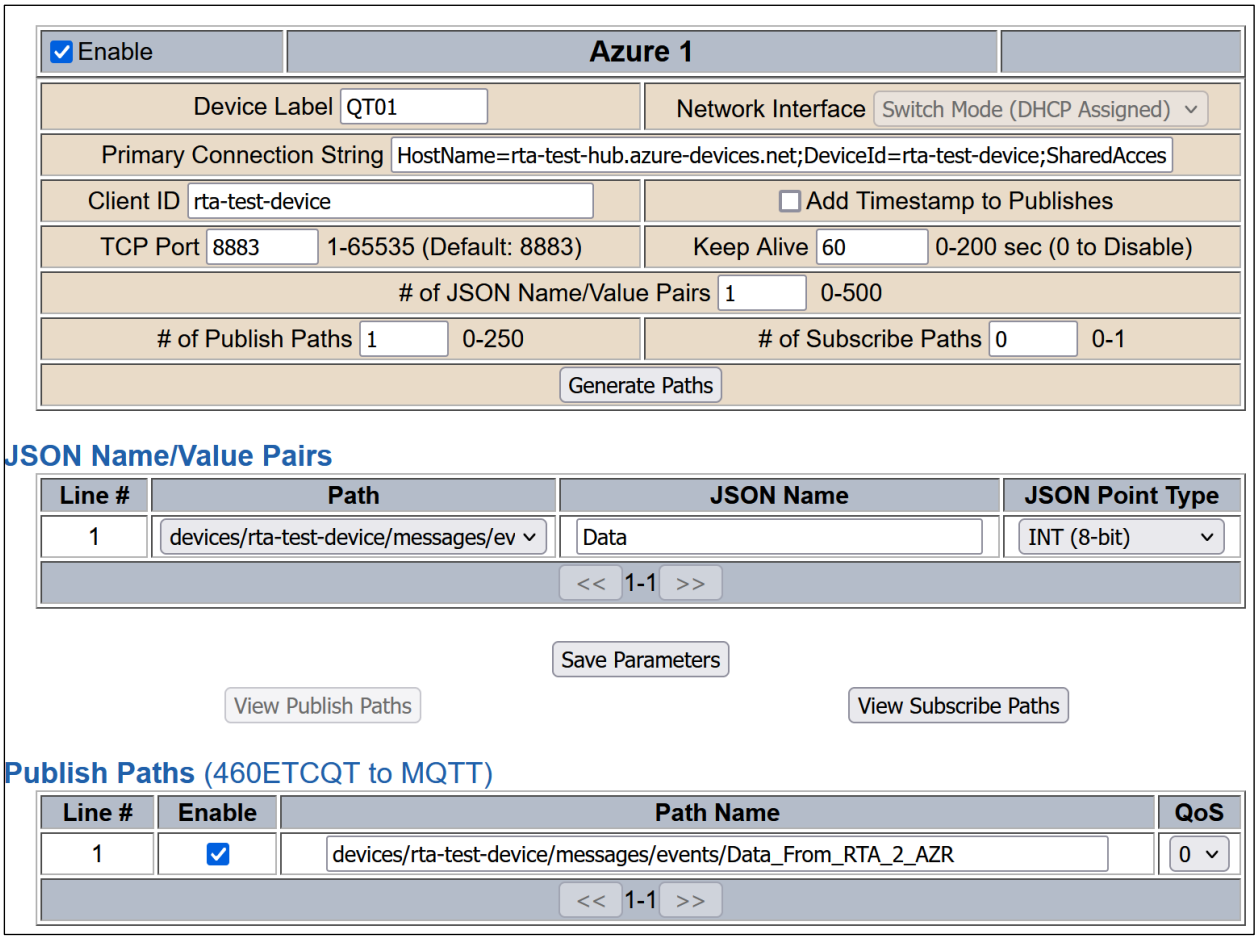


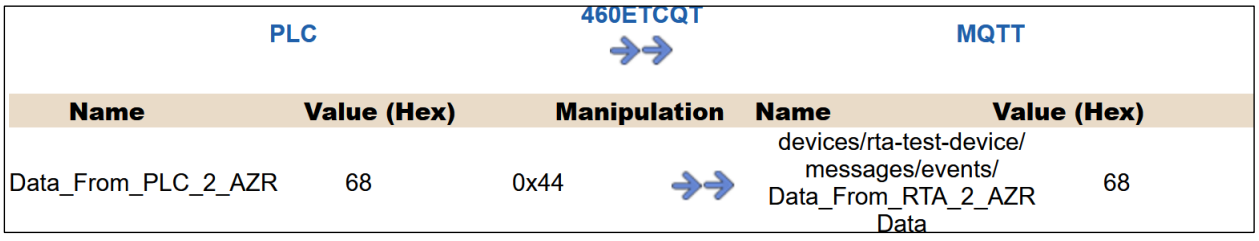
Navigate to the RTA display data page, view data from MQTT, refresh the page, and the value should be visible as shown below.



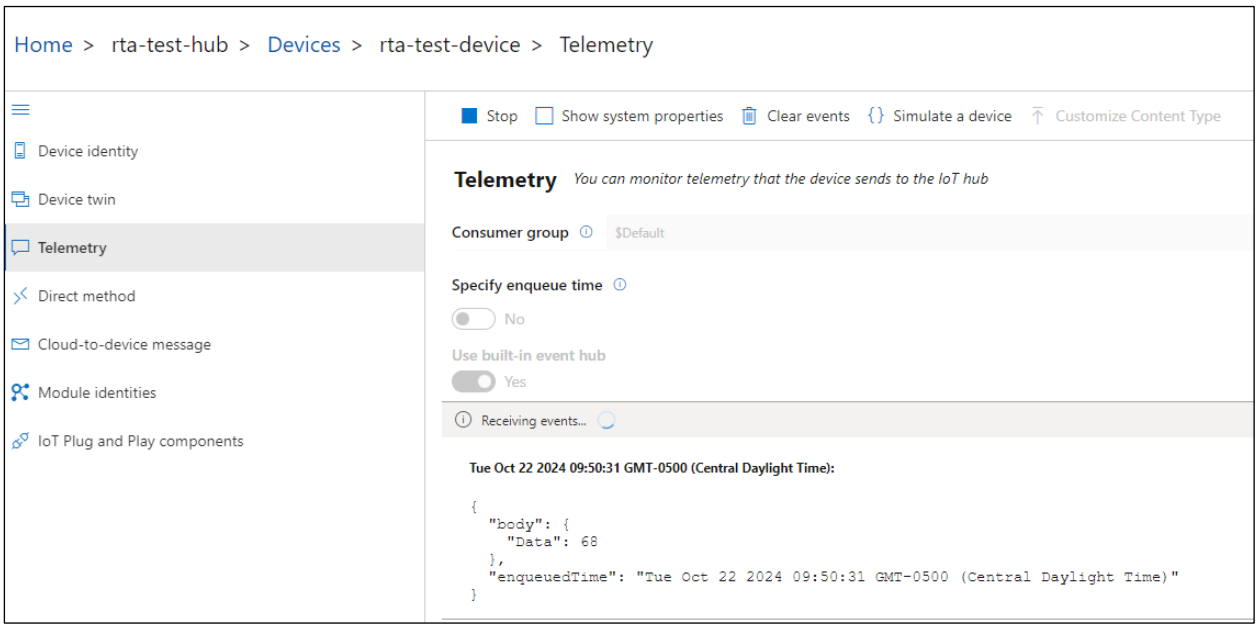
Send data from RTA gateway to Microsoft Azure (Publish Topics)

This example shows a PLC writing data to the RTA gateway and presenting that data to the Publish topic on Microsoft Azure





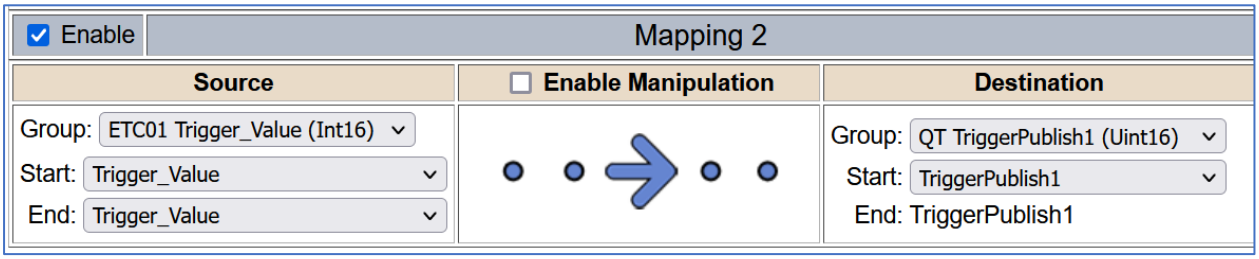
In Azure IoT Explorer, navigate to the device you would like to publish data to and select the telemetry tab on the left. In the telemetry window click Start in the top left corner. Any messages published to the selected device will now be shown in the telemetry window.



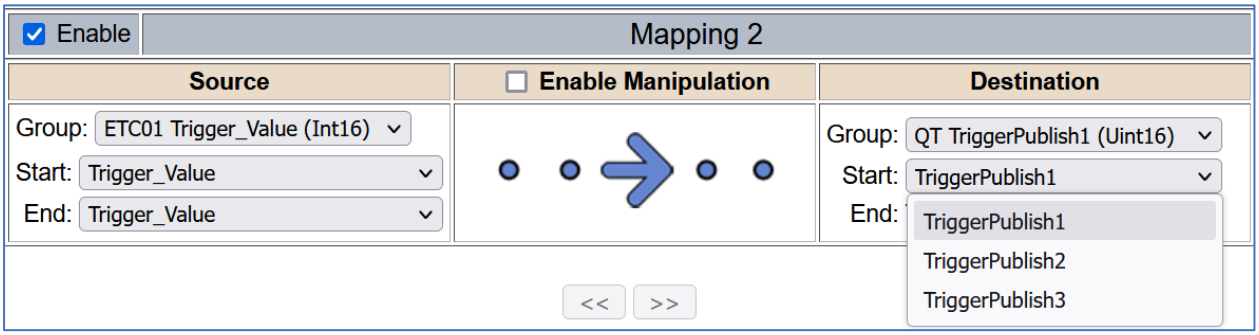
QT Publish Trigger

By default, the RTA gateway will publish to the broker based on change of state. This means a new publish to the broker will occur any time a value changes. In an application where the client is being charged per publish, such as with AWS, this is not ideal. In this situation the QT Publish trigger can be configured. When the QT Publish Trigger is configured a publish will only occur when the trigger value is incremented, allowing the user to control when data is published.

In the example below, it is shown how the trigger would be configured for MQTT device 1. A trigger value is being mapped from the mating protocol into the QT TriggerPublish1 destination. With this configuration all topics configured on MQTT device 1 will be published when the trigger value is incremented.



To trigger for MQTT device 2 or 3, TriggerPublish2 or TriggerPublish3 should be selected under the "Start" dropdown in the destination configuration as shown below.



Mapping - Transferring Data Between Devices

There are 5 ways to move data from one protocol to the other. You can combine any of the following options to customize your gateway as needed.

Option 1 – Data Auto-Configure Mappings: The gateway will automatically take the data type (excluding strings) from one protocol and look for the same data type defined in the other protocol. If there isn't a matching data type, the gateway will map the data to the largest available data type. See Data Auto-Configure section for more details.

Option 2 – String Auto-Configure: The gateway will automatically take the string data type from one protocol and map it into the other. See String Auto-Configure section for more details.

Option 3 – Manual Configure Mappings: If you don't want to use the Auto-Configure Mappings function, you must use the manual mapping feature to configure translations.

Option 4 – Manipulation/Scaling: You can customize your data by using math operations, scaling, or bit manipulation. See Data Mapping-Explanation section for more details.

Option 5 – Move Diagnostic Information: You can manually move diagnostic information from the gateway to either protocol. Diagnostic information is not mapped in Auto-Configure Mappings Mode. See Diagnostic Info section for more details.

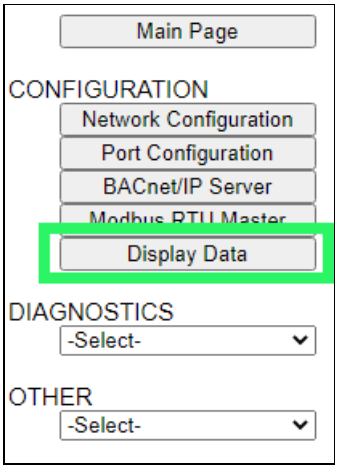
Going from Manual Mapping to Auto-Mapping will delete ALL mappings and manipulations configured.

Display Mapping and Values

The Display Data and Display String pages are where you can view the actual data for each mapping that is set up.

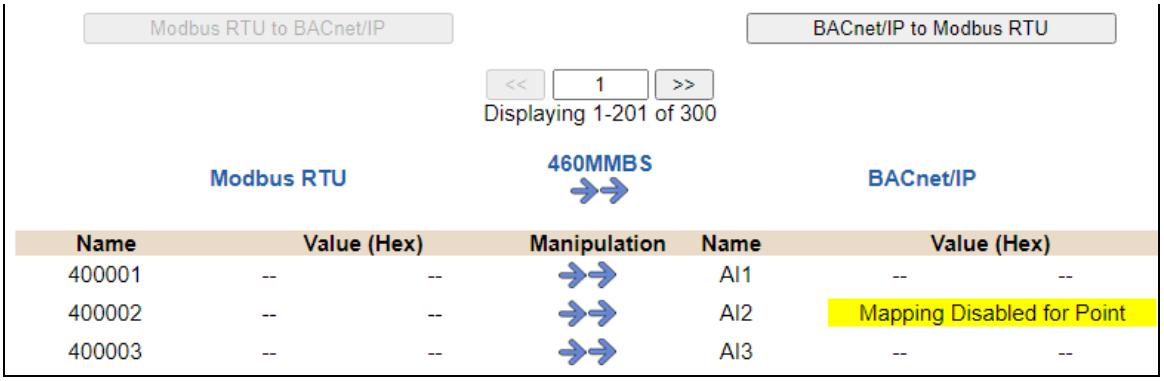
Display Data

Click the **Display Data** button to view how the data is mapped and what the values of each mapping are.



Here you will see how each data point (excluding strings) is mapped. To view, select the device from the dropdown menu and click **View** to generate the information regarding that device. Then select either the ***Protocol 1 to Protocol 2*** or ***Protocol 2 to Protocol 1*** button, correlating to the direction you wish to see the data.

This page is very useful when verifying that all data is mapped somehow from one protocol to another. If a data point is not mapped, it will display on this page in a yellow highlighted box. The Display Data page will display up to 200 mappings per page, simply navigate to the next page for the additional mapping to display.



In the above example, we see the following:

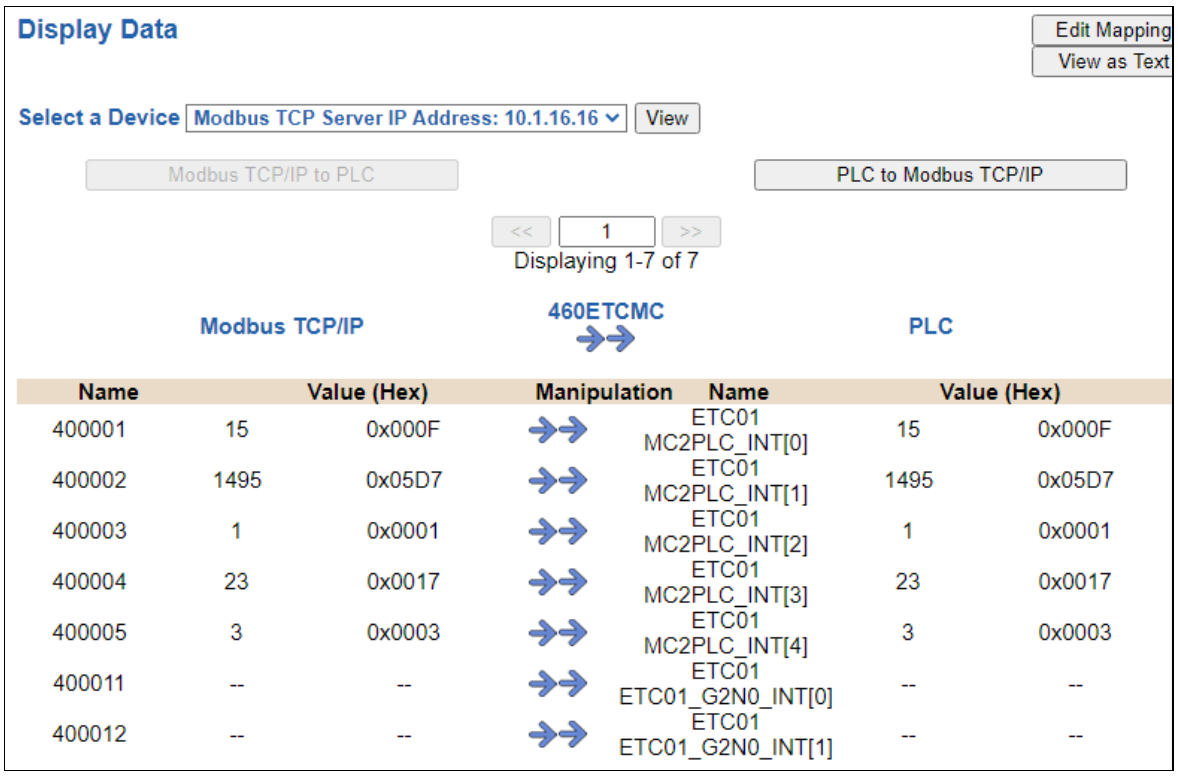
- Modbus register 400001 from Slave 1 is being mapped to AI1 on BACnet
- Nothing is being moved from Modbus register 400002 to AI2 on BACnet because the mapping is disabled
- Modbus register 400003 from Slave 1 is being mapped to AI3 on BACnet

NOTE: If a data point is mapped twice, only the first instance of it will show here. EX: If Modbus 400001 & 400040 from Slave 1 are both mapped to AI1, only 400001 will show as being mapped to AI1.

If there are values of "- - "on this page, it indicates that the source has not yet been validated and no data is being sent to the destination.

The example below reflects the Modbus to PLC flow of data. The Modbus (left side) is the source and the PLC (right side) is the destination.

- The 460 gateway has received valid responses from Modbus registers 400001- 400005 and therefore can pass the data on to the PLC tag called MC2PLC_INT.
- The 460 gateway has NOT received valid responses from Modbus register 400011 & 400012. As a result, the data cannot be passed to the PLC tag ETC01_GN0_INT2 and indicates so by using "- - "in the value column of the table.

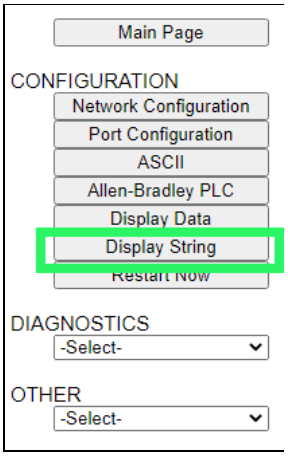


To view the actual data mappings, click the **Edit Mapping** button. For more details, see the Data Mapping-Explanation section.

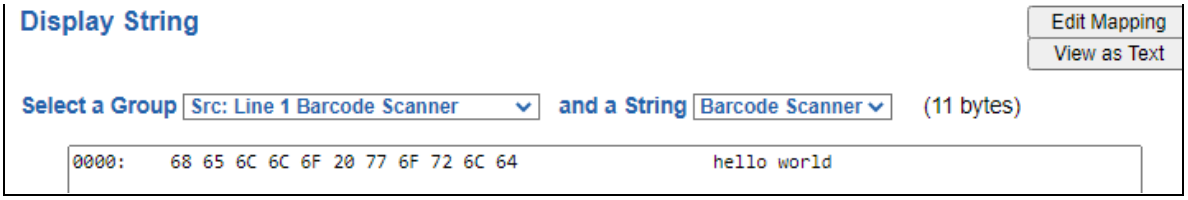
To view the data mappings purely as text, click the **View as Text** button. For more details, see the View Data Mapping as Text section.

Display String

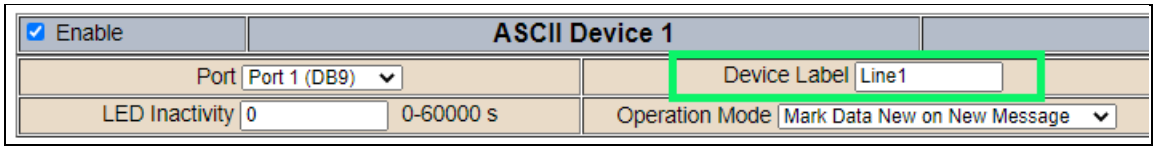
Click the **Display String** button to view what the values of each Parsing and/or Concatenating strings are, you can also click on the Edit Mapping to view the mapping of each string.

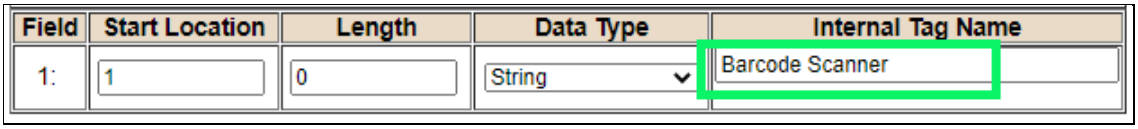


To view the source or destination groups from a string, click the dropdown menu to generate the information regarding that device. The string data will be displayed in both Hex and ASCII (only the ASCII data is sent). The example below shows data that is coming from the source device. A group will be displayed for each Parsing/Concatenating String field that is configured.



In the Group drop down, “Line1” is defined on the ASCII Device configuration page and “Barcode Scanner” is defined in the ASCII Parsing configuration.

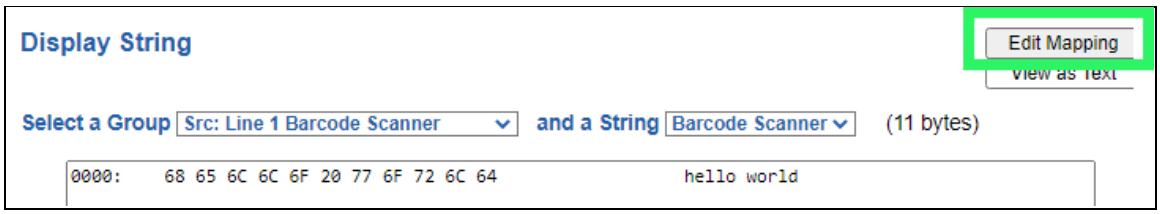




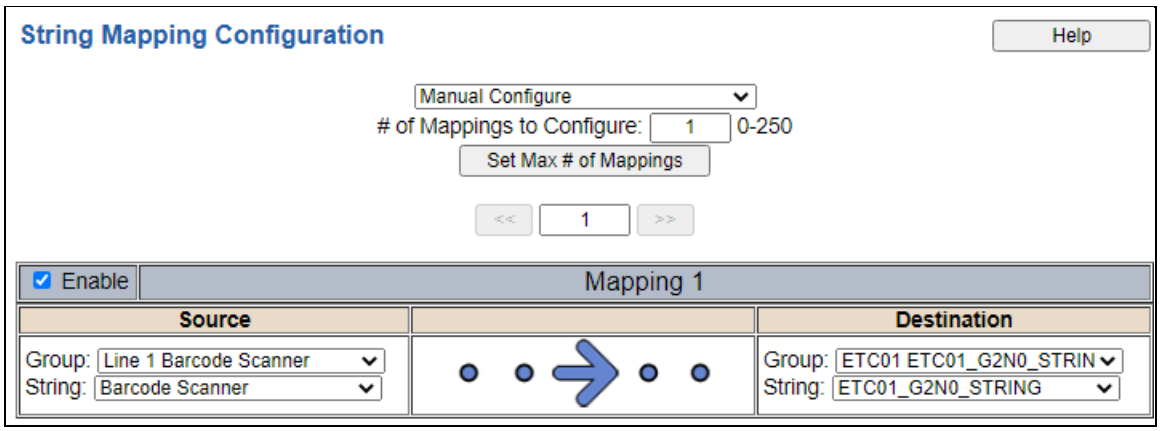
If there are values of "Data Not Valid "on this page, it indicates that the source has not been validated yet and no data is being sent to the destination.

NOTE: You can view the whole string data by clicking on **Diagnostics Info** drop down and navigating to ASCII Diagnostics page. You will also have to select the port you want to view in the dropdown below ASCII.

To view the string mappings, click the **Edit Mapping** button. For more details see the **String Mapping-Explanation** section.



NOTE: Only String data types can be mapped to another String data type.



To view the string mappings purely as text, click the **View as Text** button. For more details see the **View String Mapping** as Text section.

Display String use case

Sending a message of “RTA,Support,Rocks” from an ASCII device to the RTA unit. The ASCII Parsing Configuration would look like my example below. There are more detailed examples of what all the fields represent in the ASCII Parsing section.

ASCII Device 1 (Line1)

Max Number of Fields: 3 1-50 | Min Number of Fields: 1 1-50

Parsing Delimiter: , 44 0x2c

Update Fields

Field	Start Location	Length	Data Type	Internal Tag Name
1:	1	0	String	Header 1
2:	1	0	String	Header 2
3:	1	0	String	Header 3

The message is broken up into 3 “Groups” or Parsing fields.

Display String — Edit Mapping / View as Text

Select a Group: Src: Line1 Header 1 and a String: Header 1 (3 bytes)

```
0000:    52 54 41                                          RTA
```

Display String — Edit Mapping / View as Text

Select a Group: Src: Line1 Header 2 and a String: Header 2 (7 bytes)

```
0000:    53 75 70 70 6F 72 74                              Support
```

Display String — Edit Mapping / View as Text

Select a Group: Src: Line1 Header 3 and a String: Header 3 (5 bytes)

```
0000:    52 6F 63 6B 73                                    Rocks
```

To view the Entire message, click on the Diagnostic drop down, select Diagnostics Info. Select ASCII, click view, select your Port. Whole data will be in the Last Message Sent Diagnostic box.

Diagnostics

ASCII View

Port 1 (DB9) View

Last Message Sent (17 bytes)

```
0000:    52 54 41 2C 53 75 70 70 6F 72 74 2C 52 6F 63 6B   RTA,Support,Rock
0016:    73                                                s
```

Data and String Mapping – Auto-Configure

The Auto-Configure function looks at both protocols and will map the data between the two protocols as best as it can so that all data is mapped. Inputs of like data types will map to outputs of the other protocols like data types first. If a matching data type cannot be found, then the largest available data type will be used. Only when there is no other option is data truncated and mapped into a smaller data type.

If the Auto-Configure function does not map the data as you want or you want to add/modify the mappings, you may do so by going into Manual Configure mode.

The following are examples of the Auto-Configure function.

1) This example shows a common valid setup.

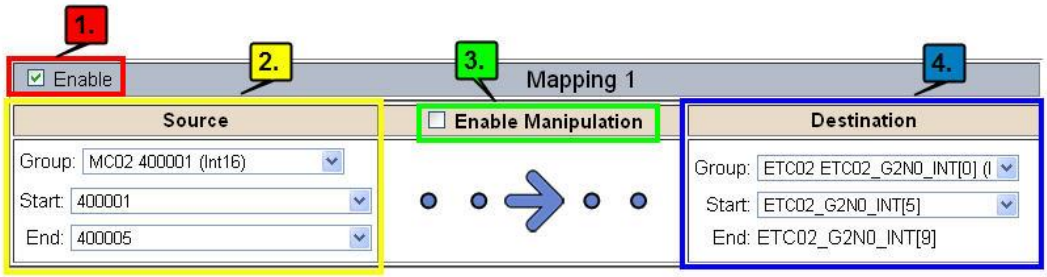
   a. Both Source values were able to be mapped to a corresponding Destination value.

2) This example shows how Auto-Configure will make its best guess.

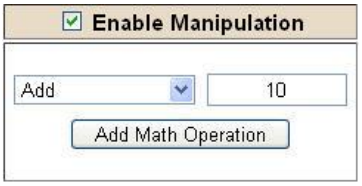
   a. The 32-bit Float from the Source location could not find a matching Destination data-type. After all other like data types were mapped, the only data type available was the 2nd 32-bit Uint data type. Auto-Configure was completed even though the data in the Float will be truncated.

Data Mapping – Explanation

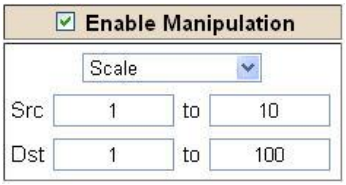
Below are the different parts that can be modified to make up a data mapping.



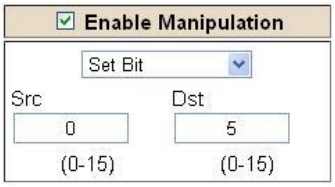
1) Enable (red box above): Check to enable mapping. If not checked, this mapping is skipped.
2) Source Field (yellow box above):
   a) Group - Select the data group you set up in the protocol config to use for this mapping.
   b) Start - This is the starting point for this mapping.
   c) End - This is the final point to be included for this mapping.
3) Manipulation Area (green box above):
   a) Enable the Data Manipulation. This can be enabled for any mapping.
   b) Click **Add Math Operation** for each operation needed. Up to 3 are allowed unless you are using the Scale, Set Bit, or Invert Bit functions. If using Scale, Set Bit, or Invert Bit, then only 1 operation is allowed.
   c) Select the Operation(s) to perform.
      i) Math Operations are performed in the order they are selected.
      ii) If more than one point is selected on the source, the Math Operations will be performed on every point.
   d) Enter the value(s) for the operation.



Example of Add (similar for Subtract, Multiple, Divide, and MOD). This will add a value of 10 to the source field before it is written to the destination field.



Example of Scale. This will scale the source values from 1-10 into 1-100 for the destination.



Example of Set Bit (similar to Invert Bit). This will take the value of the 0th source bit and copy it into the value of the 5th destination bit.

4) Destination Field (blue box above):
   a) Group - Select the data group you set up in the protocol config to use for this mapping.
   b) Start - This is the starting point for where the data is being stored.
   c) End - The End point is derived from the length of the source and cannot be modified.

Data Mapping – Adding Diagnostic Information

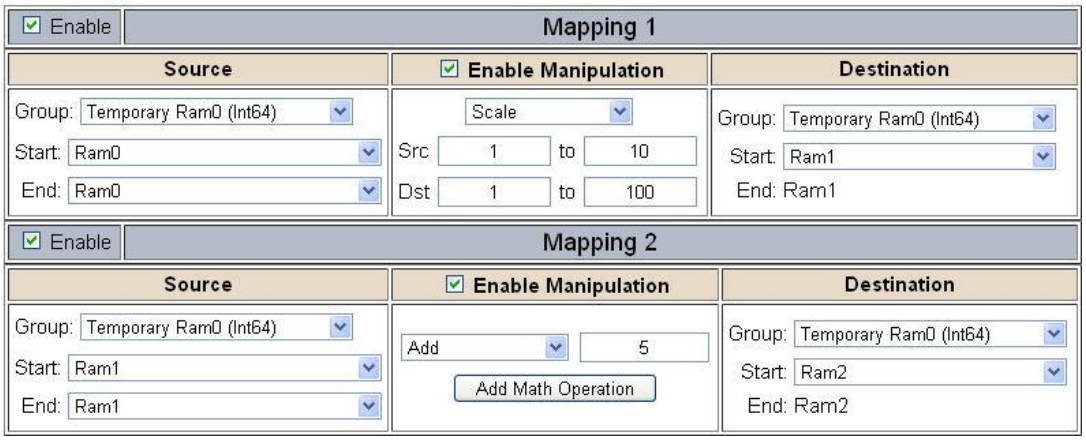
Data Mapping offers 5 different types of information in addition to any scan lines specified for each protocol.

IMPORTANT NOTE: Only add Diagnostic Information **AFTER** both sides of the gateway have been configured. If changes to either protocol are made after diagnostic information has been added to the mapping table, it is necessary to verify all mappings. Remapping may be necessary.

1) Temporary Ram (Int64)

a) This offers five levels of 64bit Integer space to assist in multiple stages of math operations. For example, you may wish to scale and then add 5. You can set up a single translation to scale with the destination as the temporary ram. Then another translation to add 5 with the source as the temporary ram.

b) The gateway will automatically convert the Source to fit the Destination, so there is no need for Int 8, 16, 32 since the 64 may be used for any case.



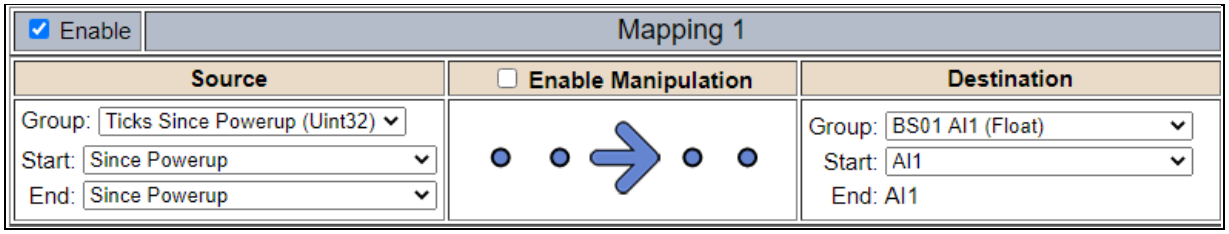
In this example, Ram0 is scaled into Ram1. Ram1 is then increased by 5 and stored into Ram2. Ram0 and Ram2 could be considered a source or destination group.

2) Temporary Ram (Double)

a) This is like the Temporary Ram (Int 64), except manipulations will be conducted against the 64bit floating point to allow for large data.

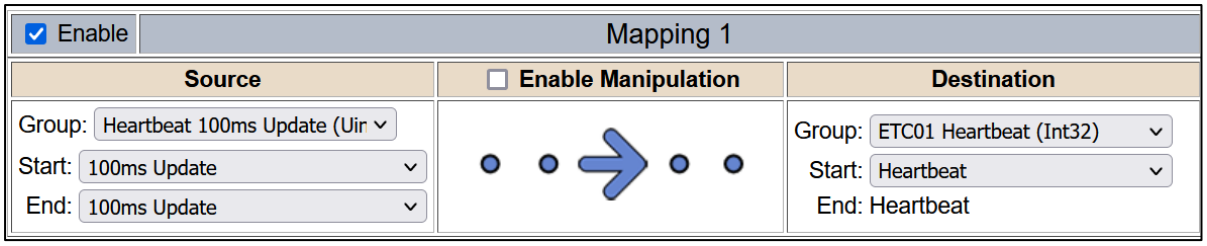
3) Ticks Per Second

a) The gateway operates at 200 ticks per second. This equates to one tick every 5ms. Thus, mapping this to a destination will give easy confirmation of data flow without involving one of the two protocols. If data stops on the destination end, then the RTA is offline.



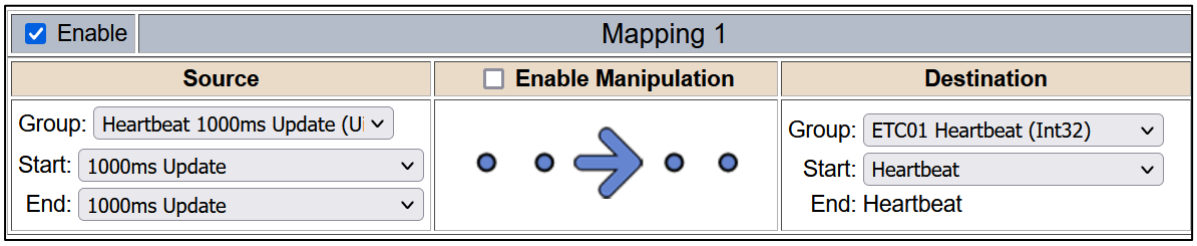
4) **Heartbeat 100ms Update**

a) The Heartbeat 100ms Update variable can be used as a heartbeat that updates once every 100ms. The variable starts at 0 on gateway startup and increments by 1 every 100ms. This can be mapped into a destination on one of the available protocols to monitor the gateways connection status. If the value stops updating every 100ms the gateway is offline.



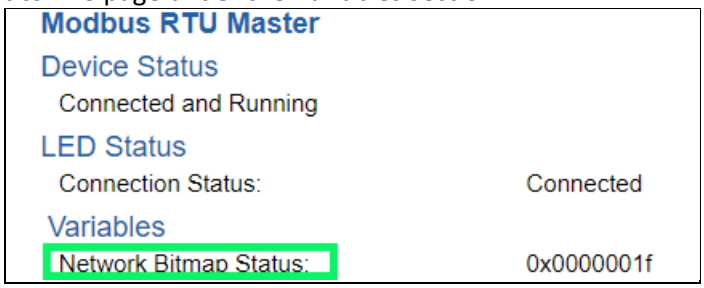
5) **Heartbeat 1000ms Update**

a) The Heartbeat 1000ms Update variable can be used as a heartbeat that updates once every 1000ms. The variable starts at 0 on gateway startup and increments by 1 every 1000ms. This can be mapped into a destination on one of the available protocols to monitor the gateways connection status. If the value stops updating every 1000ms the gateway is offline.



6) **XY_NetBmpStat**

a) If a protocol is a Client/Master, there is a Network Bitmap Status that is provided on the Diagnostics Info page under the Variables section.



b) Since a Client/Master may be trying to communicate with multiple devices on the network, it may be beneficial to know if a Server/Slave device is down. By using this Network Bitmap Status, you can expose the connection statuses of individual devices. **Values shown are in HEX.**

i) 0x00000002 shows that only device 2 is connected

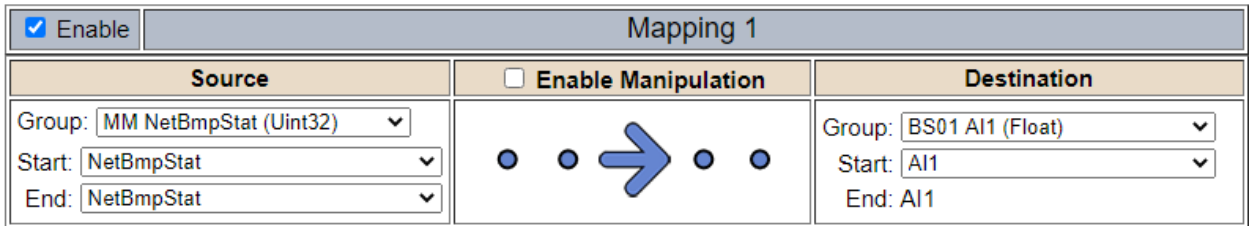
ii) 0x00000003 shows that only devices 1 and 2 are connected

iii) 0x0000001f shows that all 5 devices are connected (shown in image above)

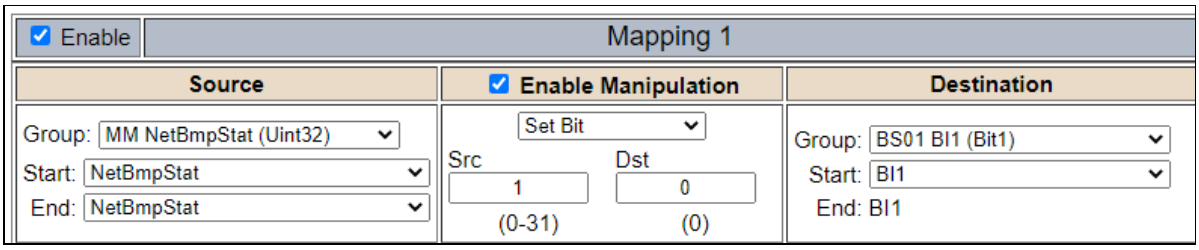
c) There are multiple ways to map the NetBmpStat.

Option 1: Map the whole 32bit value to a destination. Example below shows the NetBmpStat is going to an Analog BACnet object. Using a connection of 5 Modbus Slave devices AI1 will show a value of 31.0000. Open a calculator with programmer mode and type in 31, this will represent bits 0 – 4 are on. This mean all 5 devices are connected and running.

If using an AB PLC with a Tag defined as a Dint, then expand the tag within your RSlogix software to expose the bit level and define each bit as a description such as device1, device2, etc.



Option 2: You can extract individual bits from the NetBmpStat by using the Set Bit Manipulation and map those to a destination. You'll need a mapping for each device you want to monitor. Example below shows Modbus device 2 (out of 5) is being monitor to a BACnet Binary Object. You can define the object in the BACnet Name configuration.



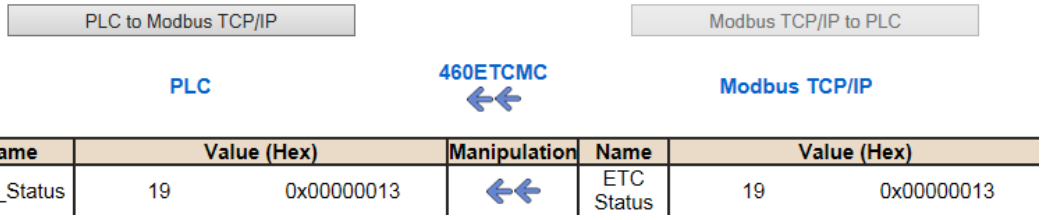
7) **Status_XY**

a) There are two Statuses provided, one for each protocol. This gives access to the overall status of that Protocol. Each Bit has its own meaning as follows:

Common Status: 0x000000FF (bit 0-7) 1st byte

Hex:	Bit Position:	Decimal:	Explanation:
0x00	0	0	if we are a Slave/Server
0x01	0	1	if we are a Master/Client
0x02	1	2	connected (0 not connected)
0x04	2	4	first time scan
0x08	3	8	idle (usually added to connected)
0x10	4	16	running (usually added to connected)
0x20	5	32	bit not used
0x40	6	64	recoverable fault
0x80	7	128	nonrecoverable fault

For this example, the ETC Status is mapped to a PLC tag called PLC_Status



Name	Value (Hex)		Manipulation	Name	Value (Hex)	
PLC_Status	19	0x00000013		ETC Status	19	0x00000013

Example: ETC Status is 0x00000013 (19 decimal), here is the break down

Hex	Bit	Decimal	Explanation
0x01	0(on)	1	if we are a Master/Client
0x02	1(on)	2	connected (0 not connected)
0x10	4(on)	16	running (usually added to connected)
Total:	0x13	19	

External Faults: 0x0000FF00 (bit 8-15) 2nd byte

Hex:	Bit Position:	Decimal:	Explanation:
0x00	8	0	local control
0x01	8	256	remotely idle
0x02	9	512	remotely faulted
0x04	10	1,024	idle due to dependency
0x08	11	2,048	faulted due to dependency

Recoverable Faults: 0x00FF0000 (bit 16-23) 3rd byte

Hex:	Bit Position:	Decimal:	Explanation:
0x01	16	65,536	recoverable fault - timed out
0x02	17	131,072	recoverable fault - Slave err

Non-Recoverable Faults 0xFF000000 (bit 24-31) 4th byte

Hex:	Bit Position:	Decimal:	Explanation:
0x01	24	16,777,216	nonrecoverable fault – task fatal err
0x02	25	33,554,432	nonrecoverable fault – config missing
0x04	26	67,108,864	nonrecoverable fault – bad hardware port
0x08	27	134,217,728	nonrecoverable fault – config err
0x10	28	268,435,456	Configuration Mode
0x20	29	536,870,912	No Ethernet Cable Plugged In

For this example, the MC Status is mapped to a PLC tag called MC_Status

PLC to Modbus TCP/IP | Modbus TCP/IP to PLC

PLC — 460ETCMC — Modbus TCP/IP

Name	Value (Hex)		Manipulation	Name	Value (Hex)	
MC_Status	65601	0x00010041	←	MC Status	65601	0x00010041

Example: MC Status is 0x00010041 (65601 decimal), here is the break down, we know that bytes 1 and 3 are being used, so here is the break down,

Common Status:

Hex:	Bit:	Decimal:	Explanation:
0x01	0(on)	1	if we are a Master/Client
0x40	6(on)	64	recoverable fault

Recoverable Faults:

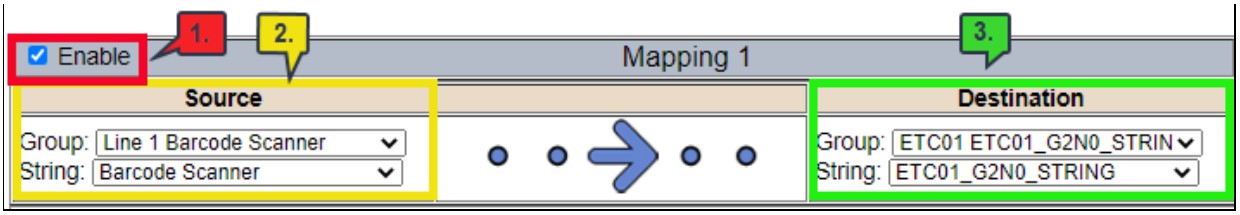
Hex:	Bit:	Decimal:	Explanation:
0x01	16	65,536	recoverable fault – timed

Total:	0x010041	65,601

String Mapping – Explanation

Below are the different parts that can be modified to make up a string mapping.

String data types can only be mapped to other string data types. There is no manipulation that can be done on the string.

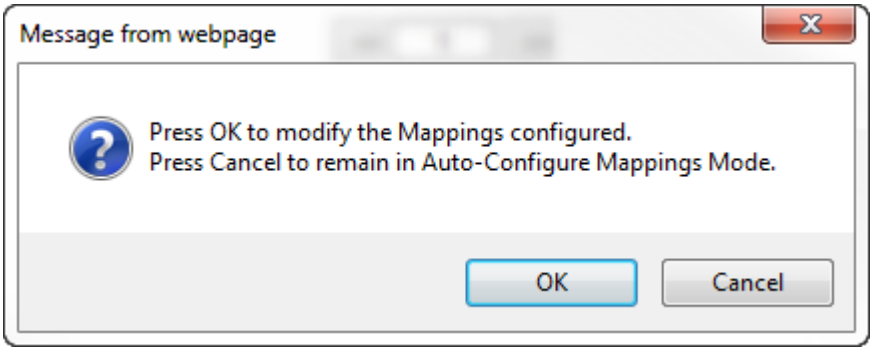


1) Enable (red box above): Check to enable mapping. If not checked, this mapping is skipped.
2) Source Field (yellow box above):
   a) Group - Select the string data group you set up in the protocol config to use for this mapping.
   b) String - This is the string used for this mapping.
3) Destination Field (green box above):
   a) Group - Select the string data group you set up in the protocol config to use for this mapping.
   b) String - This is the string where the data is being stored.

Mapping – Auto-Configure Mode to Manual Configure Mode

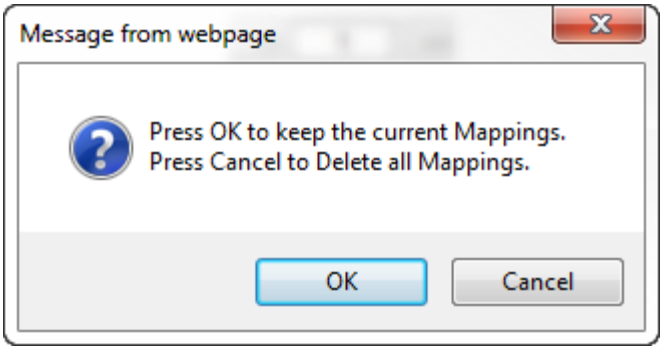
To transition from Auto-Configure Mapping Mode to Manual Configure Mode, click the dropdown at the top of the Mapping Configuration page and select Manual Configure.

After you click this button, you will be prompted to confirm if this is really what you want to do.



Click **OK** to proceed to Manual Configure Mode or click **Cancel** to remain in Auto-Configure Mappings Mode.

Once OK is clicked, there are 2 options on how to proceed from here.



1) To keep the mappings that are already configured press **OK**.

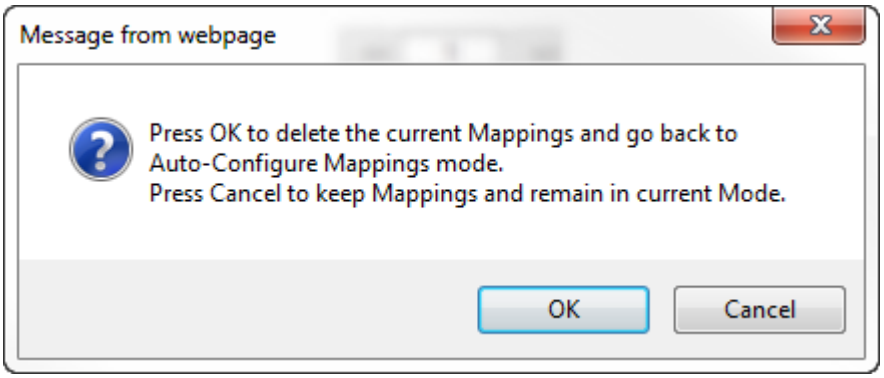
    a) You would want this option if you are adding additional mappings or you want to modify the mapping(s) that already exist.

2) To delete the mappings that are already there and start over press **Cancel**.

To modify the number of mappings, enter a number in the text field next to **# of Mappings to Configure** and click the **Set Max # of Mappings** button. You can always add more mappings if needed.

Mapping – Manual Configure Mode to Auto-Configure Mode

To transition from Manual Configure Mode to Auto-Configure Mapping Mode, click the dropdown menu at the top of the Mapping Configuration page and select Auto-Configure Mappings.



Click **OK** to proceed to delete all current mappings and go back to Auto-Configure Mappings Mode. Click **Cancel** to keep all mappings and remain in Manual Configure Mode.

NOTE: Once you revert to Auto-Configure Mapping Mode there is no way to recover the mappings you lost. Any mappings you previously have added will be deleted as well.

View as Text

Data Mapping

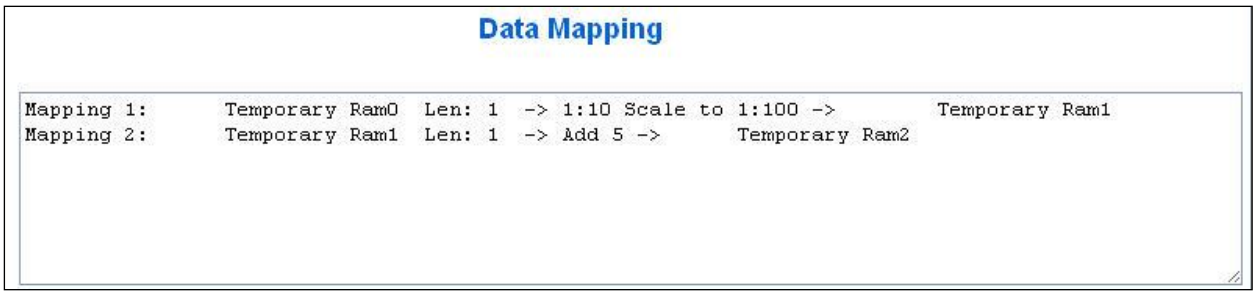
The View as Text page displays the point to point mapping(s) you set up in the Data Mapping section. This will also display any manipulation(s) that are configured.

Each line on this page will read as follows:

> **Mapping** *number*: *source point* **Len:** *Number of points mapped* **->** *manipulation (if blank then no manipulation)* **->** *destination point*

If you are looking for a specific point to see if it is mapped, you can do a find in this text box for your point in question. Example: you defined 20 Registers starting at register 1 and want to see if 400011 is mapped. If it is not in this text box, then it is not mapped, and no data will be transferred.

This is the text display for the example shown under the *Data Mapping- Adding Diagnostic Information* section.



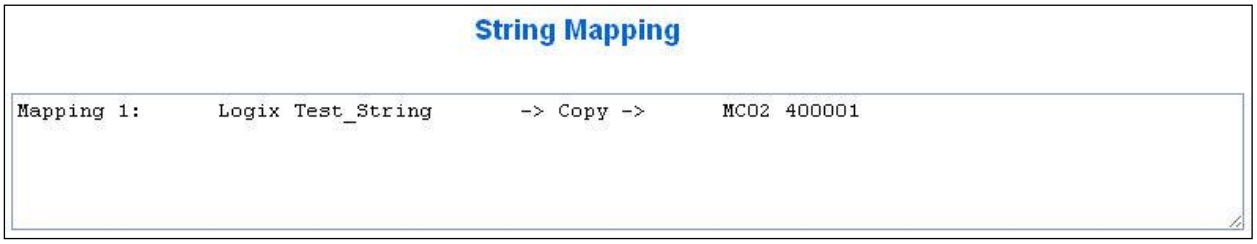
String Mapping

The View as Text page displays the string mapping(s) you set up in the String Mapping section.

Each line on this page will read as follows:

> **Mapping** *number*: *source point* **-> Copy ->** *destination point*

If you are looking for a specific point to see if it is mapped, you can do a find in this text box for your point in question. Example: you defined 20 String Tags in the PLC and want to see if "Test_String" in the Logix PLC is mapped. If it is not in this text box, then it is not mapped, and no data will be transferred.



Base Triggering – Data Validiation Triggering

With Base Triggering, you will be marking data as “Invalid” and force RTA Master/Controller/Client protocols to read all the read data points sources until ALL source protocols data is valid. You will be able to utilize the Handshake to map over to Technology Trigger and/or back over to your source protocol for reference.

How does this work?

1) Map the Triggering Variable (Source) over to Trigger **#** (Dest).
2) If Trigger **#** value changes states mark all Trigger **#** protocols read data as “Invalid”.
3) Read all source read data points until ALL source read data is valid.
4) Handshake **#** value is set equal to Trigger **#** value.
5) Map Handshake **#** to reference data point.
   Note: **#** is an internal reference to the Server/Slave number you are settings up. **ex**. RTA Server/Slave products can only be Trigger 1 and Handshake 1 since we are only 1 device. If RTA is a Master/Client, then you can have a Trigger# for each server/slave connected too.

How do you set this up?

In this example I’m using a 460MCBS. My Building Automation System wants to verify that all data read from Modbus TCP/IP Server is valid.

1) Add an extra Analog Output for your Trigger. This tells the RTA to mark all data invalid.

Write Data Groups (BACnet/IP to 460MCBS)

Data Group	Object Type	Starting Object	# of Objects
1	Analog Output (32 Bit Float)	1	21
2	Binary Output	1	0
3	CharacterString Value	51	0

   a) You can define AI21 as your validation name in the Setup BACnet Names Configuration.

Setup BACnet Names, Units, and COV

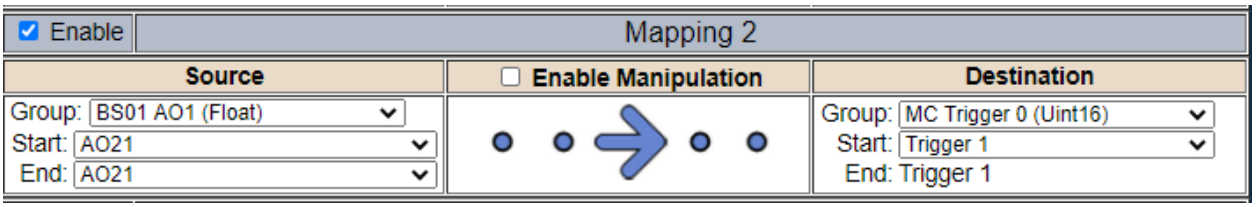
21	G01	Data Validation Trigger	Other	no-units	1.000000

2) Add another Analog Input as reference for when data has been validated. When you write from AO21 to validate data, the RTA will reply to AI40 saying “validation complete”.

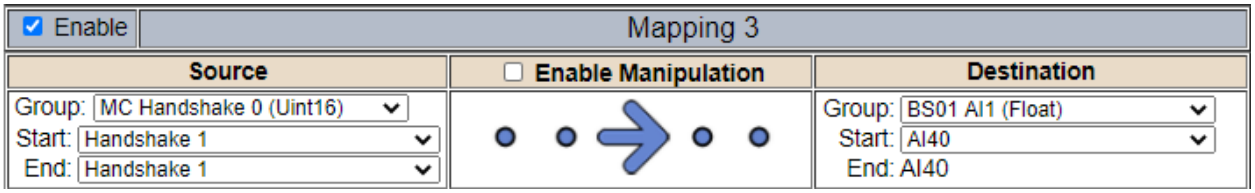
Data Group	Object Type	Starting Object	# of Objects
1	Analog Input (32 Bit Float)	1	40
2	Binary Input	1	0
3	CharacterString Value	1	0

40	G01	Data Validation Result	Other	no-units	1.000000

3) Within the Data Mapping page manually add 2 additional mappings.
4) The first mapping is going to be the Data Validation Triggering. AO21 will write to the RTA, MC Trigger 1 will mark data invalid.



5) The second mapping, the MC Handshake will increment that all data is validated and write to AI21 “all data is validated”. The value of AI40 and AO21 should be the same.



Security Configuration

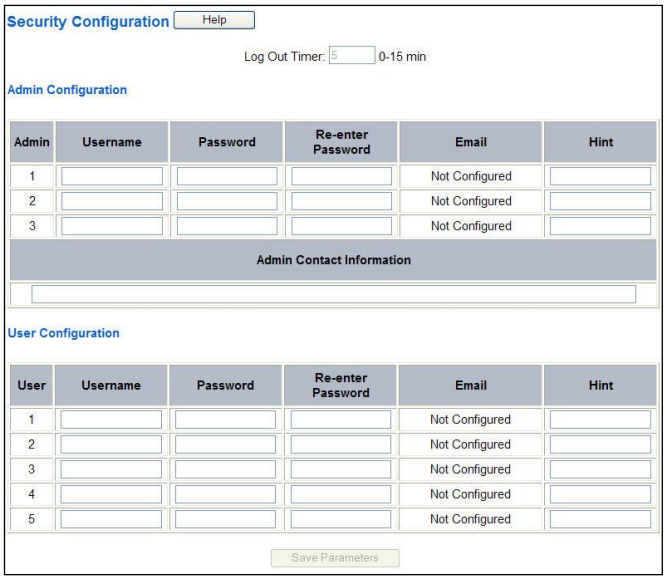
To setup security on the 460 gateway, navigate to **Other->Security Configuration**. You can configure Security for 3 administrators, 5 users, and 1 guest.

THIS IS NOT A TOTAL SECURITY FEATURE

The security feature offers a way to password protect access to diagnostics and configuration on the network. The security feature does not protect against "Air Gap" threats. If the gateway can be physically accessed, security can be reset. All security can be disabled if physical contact can be made. From the login page, click the Reset Password button twice. You will be forced to do a hard reboot (power down) on the gateway within 15 minutes of clicking the button. This process should be used in the event a password is forgotten.

Note: Only Admins have configuration access to all web pages.

1) Log Out Timer: The system will automatically log inactive users off after this period of time. **NOTE**: A time of 0 means that the user will not be automatically logged off. Instead, they must manually click the **Logout** button.
2) Username: Enter a username, max of 32 characters.
3) Password: Enter a password for the username, max of 32 characters, case sensitive.
    a. Re-enter the Password
4) E-mail: In case the password was forgotten, a user can have their password e-mailed to them if e-mail was configured.
5) Hint: A helpful reminder of what the password is.

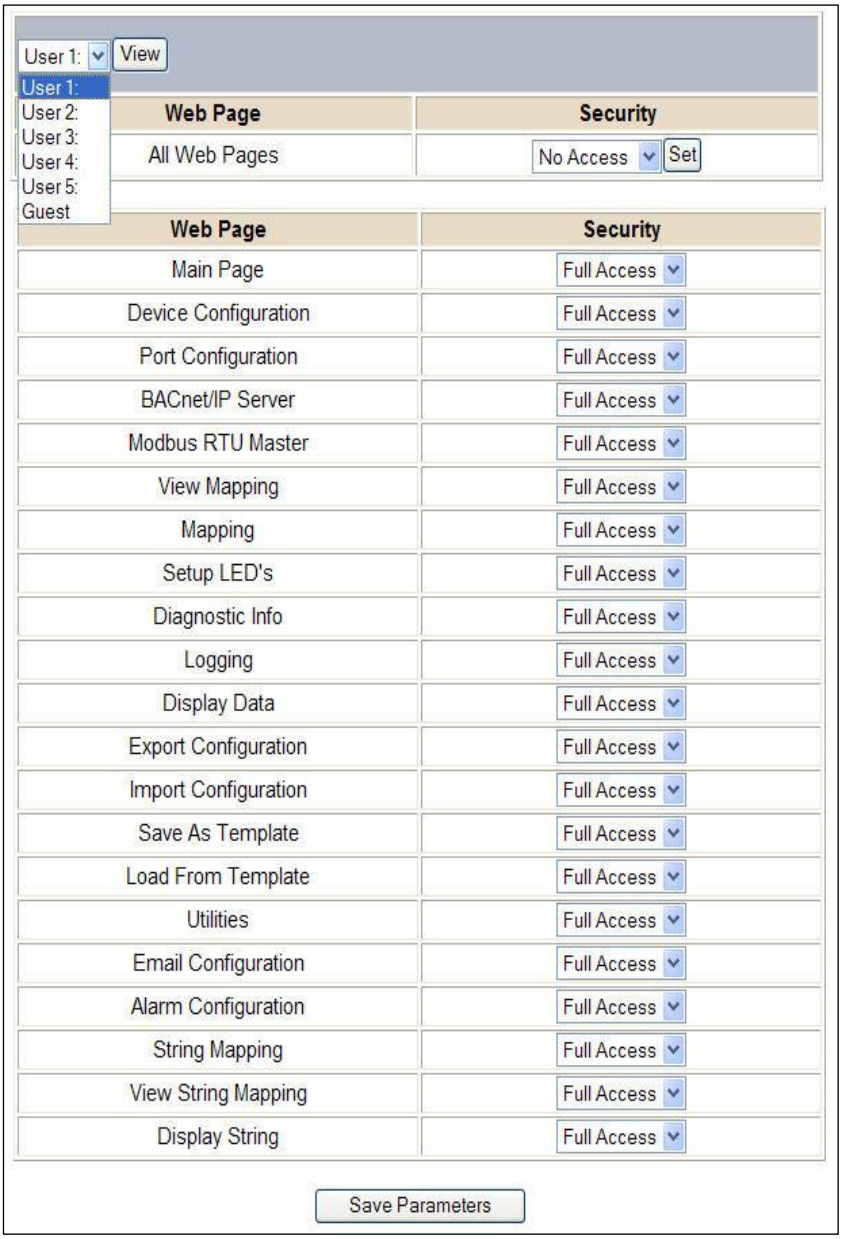


Security Configuration-Security Levels

Each webpage in the gateway can have a separate security level associated with it for each user.

Security Levels:

1) **Full Access**: Capability to view and configure a web page.
2) **View Access**: Capability to view a web page, but cannot configure parameters.
3) **No Access**: No capability of viewing the web page and page will be removed from Navigation.



Security - Log In

Username: Name of the user to login.

Password: Password of the user to login.

Log In: If login is successful, the user will be redirected to the Main Page.

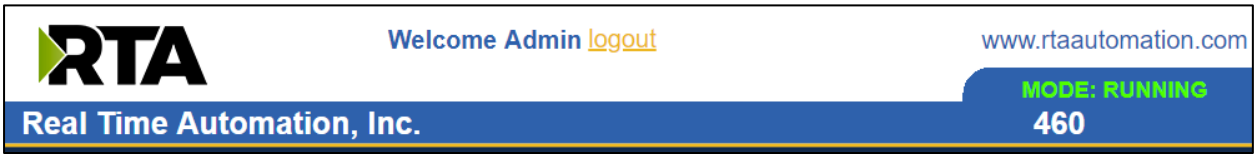
Send Password to Email: Sends the specified User's Password to the email configured for that user.

Display Hint: Displays the hint specified for the User if one was set up.

Reset Password: This is used to reset security settings. Confirm reset password must be selected to confirm this action. Once confirmed, there is a 15 minute window to do a hard reset of the gateway by physically removing and restoring power from the gateway. Once power is restored, you may navigate to the IP address of the gateway as normal.

Security - Log Out

Once a user is done with a session they may click **logout** at the top of any page. The user may also be logged out for inactivity based off of the Log Out Timer specified during the configuration.



Closing the browser is not sufficient to log out.

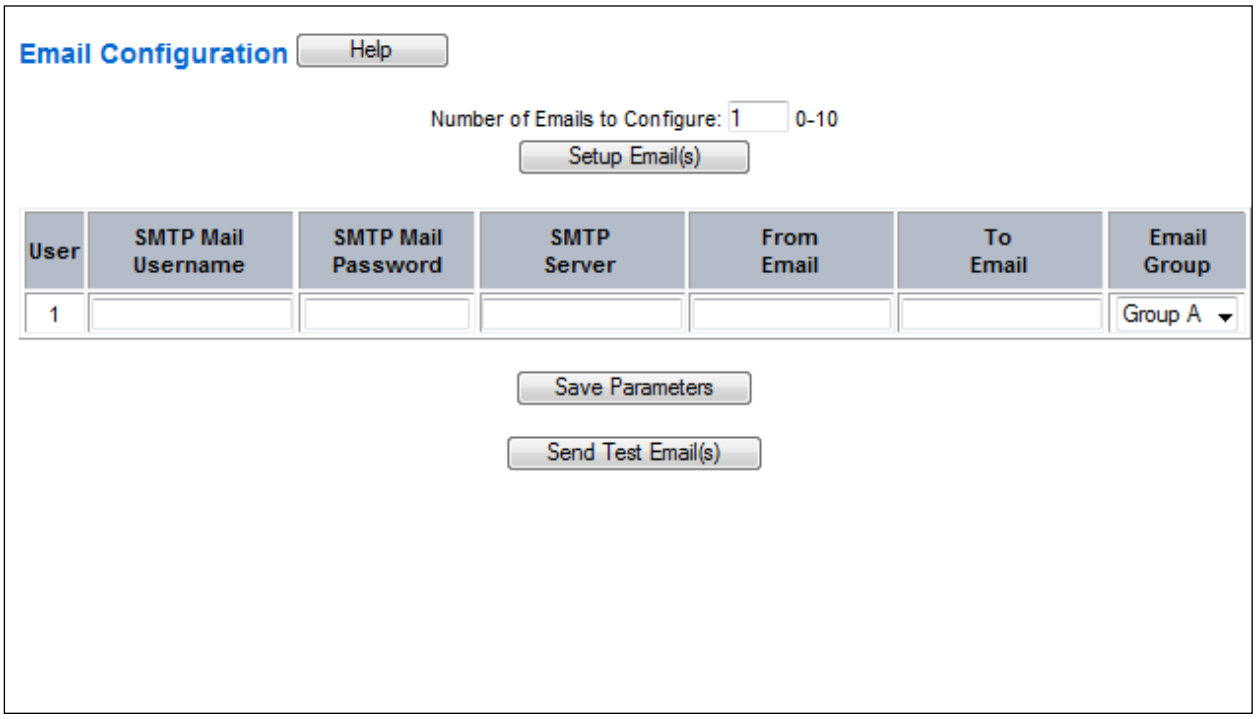
Email Configuration

To setup e-mails on the 460 gateway, navigate to **Other->Email Configuration**.

You can configure up to 10 email addresses.

1) SMTP Mail Username: The email address that the SMTP server has set up to use.
2) SMTP Mail Password: If authentication is required, enter the SMTP Server's password (Optional).
3) SMTP Server: Enter the Name of the SMTP Server or the IP Address of the Server.
4) From E-mail: Enter the e-mail that will show up as the sender.
5) To E-mail: Enter the e-mail that is to receive the e-mail.
6) E-mail Group: Choose a group for the user. This is used in other web pages.

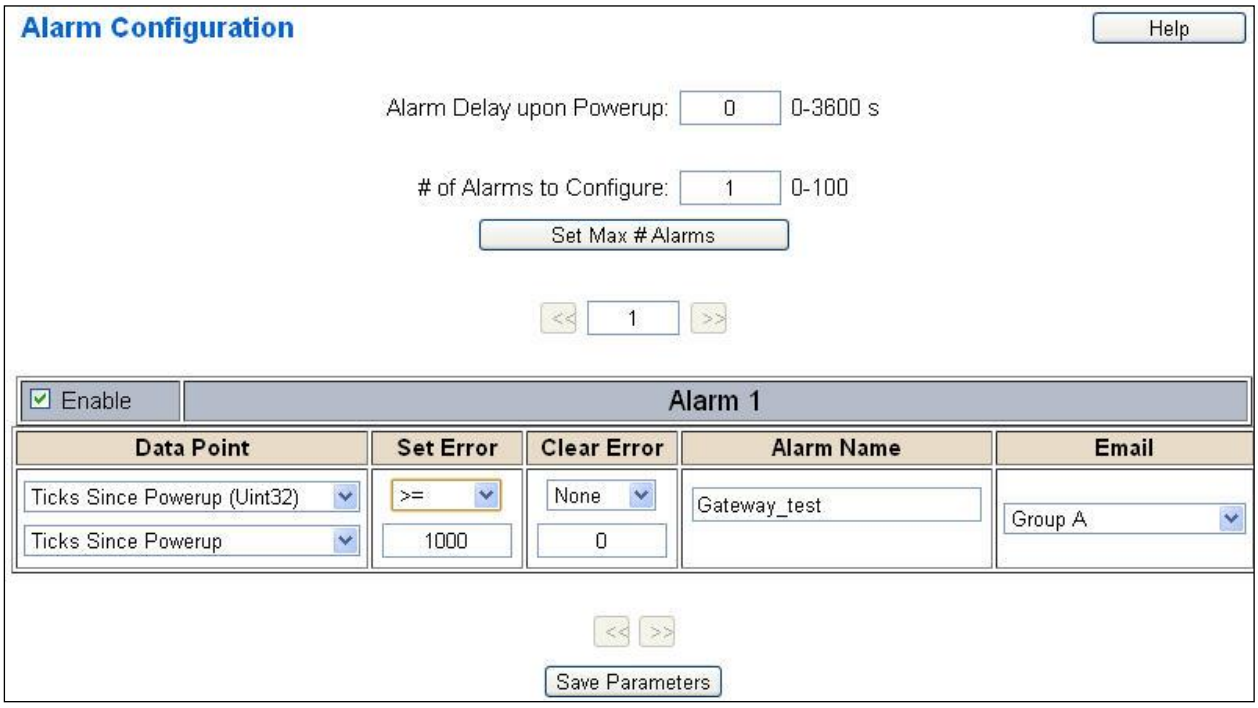
Click the **Save Parameters** button to commit the changes and reboot the gateway.



Alarm Configuration

To setup alarms on the 460 gateway, navigate to **Other->Alarm Configuration**.

1) Alarm Delay upon Powerup: At Powerup, the gateway will have values of '0' stored for all data. This may cause alarms to trigger before these values are updated by the mating protocols. Set this field to provide needed time to update fields before considering values for alarms.



2) Enter the number of alarms to configure and click **Set Max # Alarms** to generate those lines.
3) In the Data Point Section:
    a. Top dropdown: select the Data Group. This dropdown menu will contain all groups that go from the gateway to the network.
    b. Lower dropdown: select the Data Point's Specific Point. This is used to select which point in the group will be monitored for alarms.
4) In the Set Error Section:
    a. Select the Set Error Operation in the top dropdown menu. Available options are <, >, <=, >=, !=, ==, and Change of State (COS). This is the operation that will be used to compare the Data Point value against the Error Value to determine if the alarm needs to be set.
    b. Select the Set Error Value. This value is used as: 'Data Point's Value' 'Operation' 'Value.' Ex: Ticks Since Powerup >= 1000. This will set the alarm after 1000 ticks have elapsed since the unit powered up.

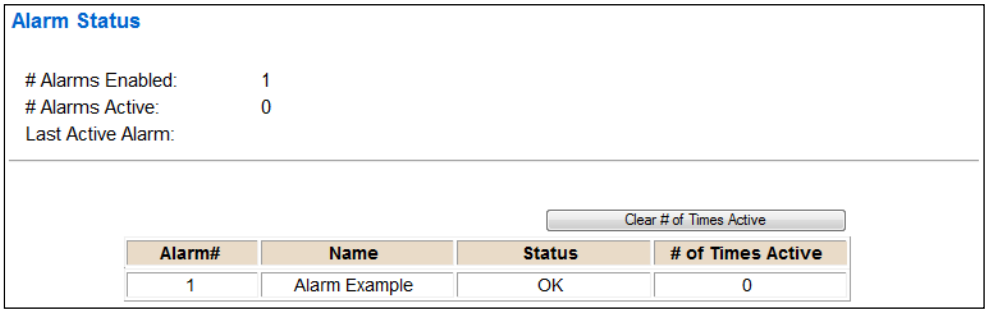
5) In the Clear Error Section:
    a. Select the Clear Error Operation. Available options are <, >, <=, >=, !=, ==, and Change of State (COS). This is the operation that will be used to compare the Data Point value against the Error Value to determine if the alarm needs to be cleared.
    b. Select the Clear Error Value.
    -Ex: Ticks Since Powerup >= 5000. This will clear the alarm after 5000 ticks have elapsed since the unit powered up.
6) Enter an Alarm Name. This will make the alarm unique and will be available in the Alarm Status page as well as in the email generated by the alarm.
7) Select an email to associate this alarm with. When an alarm is set, it sends an email. When an alarm is cleared, it will also send an email.

Click the **Save Parameters** button to commit the changes to memory and reboot the gateway.

Diagnostics – Alarm Status

Alarm Status will only display under the Diagnostic menu tab if at least 1 Alarm is enabled.

1) # Alarms Enabled: This is a count of enabled alarms.
2) # Alarms Active: This is how many alarms are presently active (set).
3) Last Active Alarm: This is the last alarm that the gateway detected.
4) **Clear # of Times Active**: This will reset all alarms '# of Times Active' to 0.
5) Alarm #: The reference number to the given alarm on the alarm setup page.
6) Name: The name of the alarm.
7) Status: The current status of the alarm, either OK or ALARM.
8) # of Times Active: This count represents the number of times this alarm has become active. If an alarm is triggered, this count will increment.

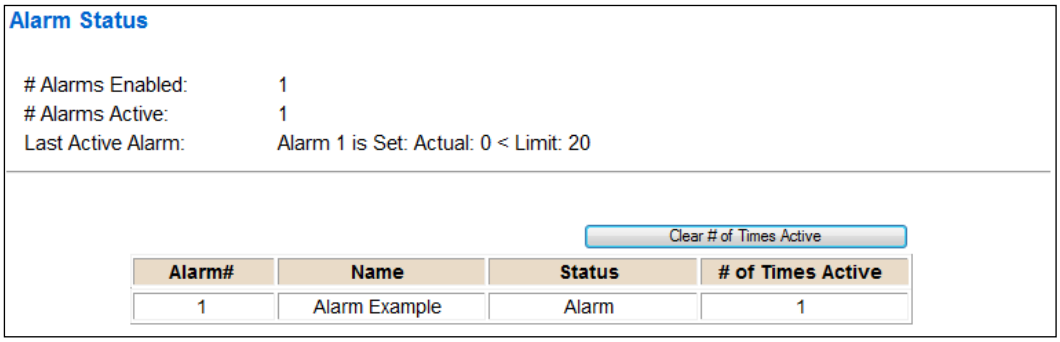


Alarms – Active

While one or more alarms are active, every page will display 'Alarms Active' at the top of the page. This will no longer be displayed if all active alarms have been cleared.

When an alarm is activated, the following will occur:

1) A one-time notification will be sent out to the email associated with the alarm.
2) For duplicate emails to occur, the alarm must be cleared and then become active again.
3) *# Alarms Active* and *# of Times Active* will be incremented.
4) Status of the Individual Alarm will be set to *Alarm*.
5) *Last Active Alarm* field will be populated with details on what triggered the alarm.



Alarms – Clear

When an alarm is cleared, the following will occur:

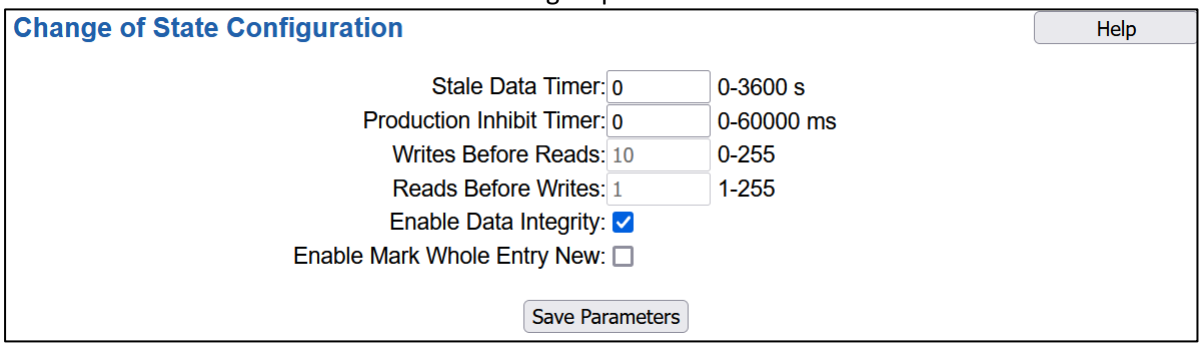
1) A one-time notification will be sent to the email associated with the alarm.
    a. For duplicate emails to occur, the alarm must become active and then be cleared again.
2) Total *# Alarms Active* will decrement. *Last Active Alarm* will not be changed.
3) Status of the Individual Alarm will be reset to *OK*.

Change of State (COS) Configuration

To access the configuration files in the 460 gateway, navigate to dropdown **Other->COS Configuration**. The gateway, by default only writes when data has changed. The gateway also waits to write any data to the destination until the source protocol is successfully connected.

Default values should fit most applications. Change these values with caution as they affect performance.

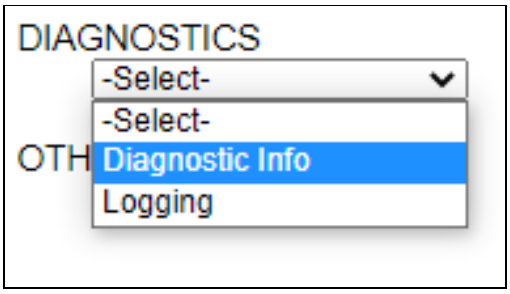
1) **Stale Data Timer**: If the data has not changed within the time allocated in this Stale Data Timer, the data will be marked as stale within the gateway and will force a write request to occur. This timer is to be used to force cyclic updates in the gateway, since data will only be written if it has changed by default. There is a separate timer per data mapping.
   Gateway behavior:
   - If time = 0s => (DEFAULT) The gateway will write out new values on a Change of State basis.
   - If time > 0s => The gateway will write out new values whenever the timer expires to force cyclic updates (write every *x* seconds).
2) **Production Inhibit Timer:** Amount of time after a Change of State write request has occurred before allowing a new Change of State to be written. This is to be used to prevent jitter. Default value is 0ms. This timer takes priority over the Stale Data Timer. There is a separate timer per data mapping. This timer is active only after the first write goes out and the first COS event occurs.
3) **Writes Before Reads**: If multiple writes are queued, execute # of Writes Before Reads before the next read occurs. Default is 10 and should fit most applications.
   Warning: A value of 0 here may starve reads if a lot of writes are queued. This may be useful in applications where a burst of writes may occur and you want to guarantee they all go out before the next set of reads begin.
4) **Reads Before Writes:** If multiple writes are queued, the # of Writes Before Reads will occur before starting the # of Reads Before Writes. Once the # of Reads Before Writes has occurred, the counter for both reads and write will be reset. Default is 1 and should fit most applications.
5) **Enable Data Integrity**: If enabled, do not execute any write requests to the destination until the source data point is connected and communicating. This prevents writes of 0 upon power up.
6) **Enable Mark Whole Entry New**: If Enabled, mark the entire scan line or data group new upon 1 data element within the scan line or data group to be new.



Click the **Save Parameters** button to commit the changes to memory and reboot the gateway.

Diagnostics Info

The Diagnostics page is where you can view both protocols' diagnostics information, # of Data Mappings, # of String Mapping and # Alarm Mappings.



For protocol specific diagnostic information, refer to the next few pages.

Diagnostics Mapping

This section displays the number of mappings that are enabled, Data Mapping and String Mapping will show the # of Errors and First Errors. Alarms will show # active and Last Alarm that was active.

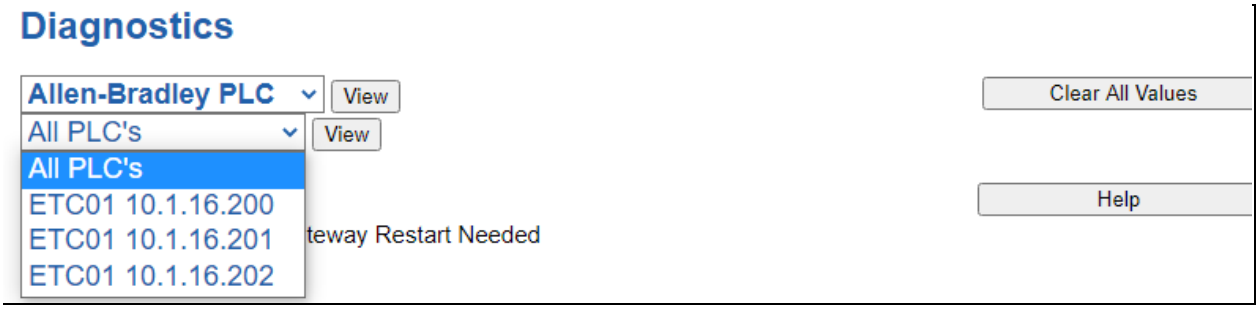
Common Errors:

1) Destination or Source Point does not exist
   a) Solution: Re-map the mapping
2) Source or Destination Pointer too small
   a) There is not enough space on either the Source, or the Destination for the data you want to copy. This is typically seen when the Destination is smaller than the amount of data being transferred to it.
3) Range Discard, Min or Max Value
   a) The actual data value is outside of the defined range
4) Math Error
   a) Operation value cannot be 0
5) Scaling Error
   a) Source Min must be smaller than Source Max
   b) Destination Min must be smaller than Destination Max

Note: you can also view this information on the Main Page.

Diagnostics – Allen-Bradley PLC

Select the **Allen-Bradley PLC** in the dropdown menu on the Diagnostics Page to view a breakdown of the diagnostics and common strings that are displayed on the page. You may also view individual counters by selecting the device in the **All PLC's** dropdown menu and clicking **View**. Additional diagnostic information can be found by clicking the **Help** button.



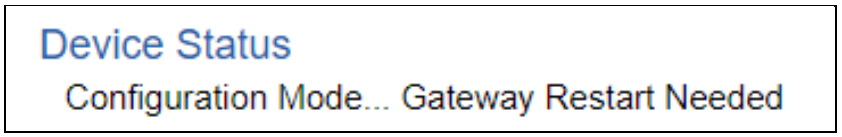
NOTE: This page will auto-refresh every five seconds with the latest data.

Clear All Values - This will only affect displayed values.

1) This will reset all displayed values back to zero and clear the Status Strings.

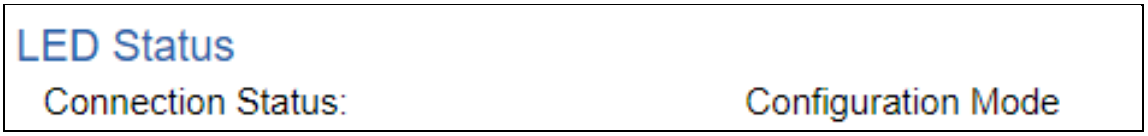
Example: If viewing Allen-Bradley PLC – ETC01 10.1.100.18, this will only clear the values for that specific PLC. This will reduce the *All PLC's* values indirectly, otherwise select All PLCs to clear all devices.

Device Status - This will only display when viewing *All PLCs*.



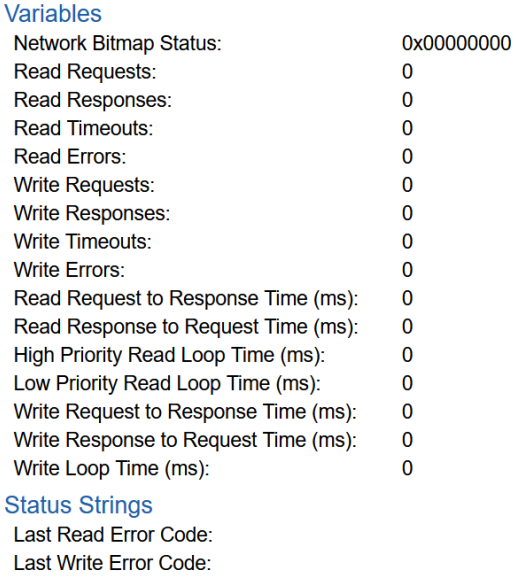
1) Connected – The gateway is connected to all the PLCs that are configured and enabled.
2) Nodes Missing (timed out) – One or more enabled PLCs are missing.
3) Empty Scan List – No PLCs are configured.
4) Dependency Protocol Faulted – The dependent protocol is missing causing the communication to go to inactive.

LED Status - This is the Status for *All PLCs* or the specific PLC selected.



1) Connected (Solid Green) – All the enabled PLC devices are connected and running.
2) Not Connected (Flashing Green) – No PLCs are enabled.
   a) Verify Allen-Bradley PLC settings and ensure that the *Enable* checkbox is checked for the appropriate device(s).
3) Connection Timeout (Flashing Red) – The gateway cannot open a connection to one or more of the enabled PLCs.
   a) Verify the IP, slot and controller type are accurate for the missing connection. Missing connection can be determined from the Network Bitmap Status value.
4) Communication not attempted yet (Flashing Red) – (Specific Server Only) No reads are configured and data needed for writes isn't valid yet.
5) Dependency Error (Flashing Red) – The dependent protocol is missing causing the communication to go to inactive.
   a) The other protocol must be connected.

Variables - These are the values for *All PLCs,* or the specific PLC selected.



1) Network Bitmap Status (hex) – Each bit corresponds to a PLC. If the bit is set, then the PLC is connected, otherwise the bit is 0. Bit 0 (right most) is PLC 1 and Bit 4 is PLC 5.
2) Read Requests – Number of read requests sent from the gateway to the PLC (N2G).
3) Read Responses – Number of valid responses sent from PLC to the gateway (G2N).

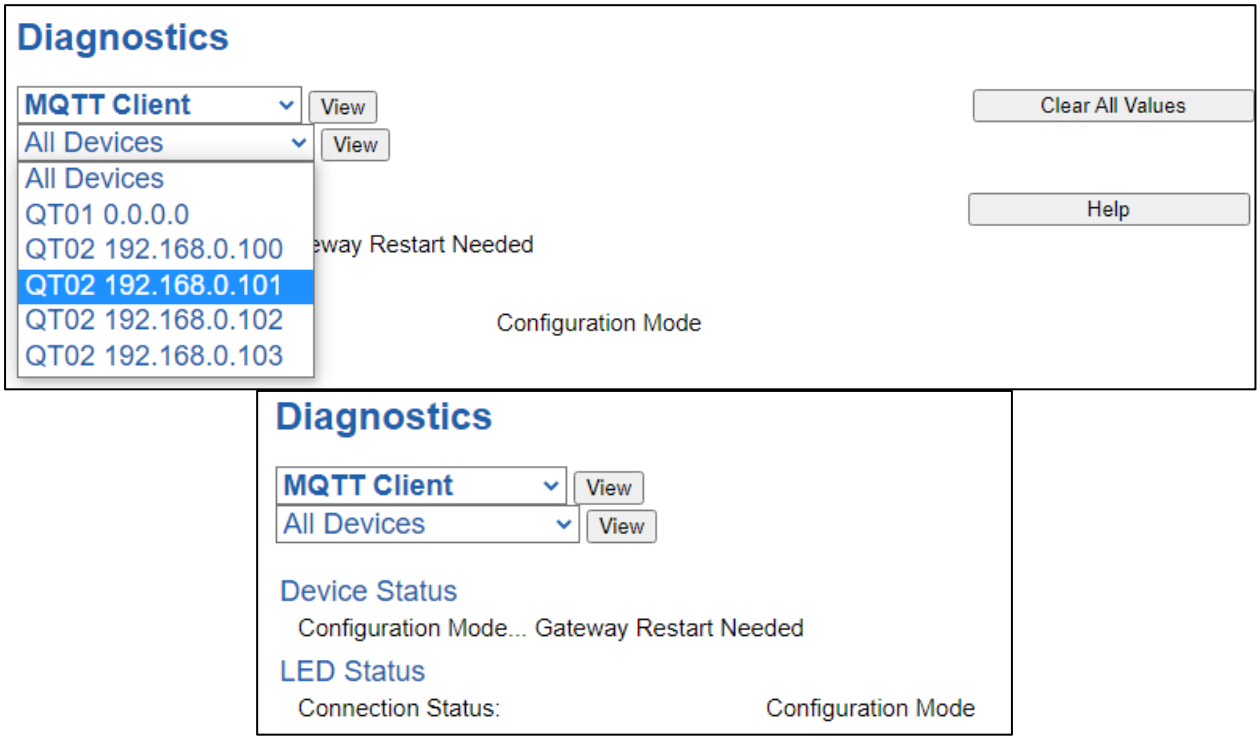
4) Read/Write Timeouts – Since we are TCP based, the gateway will timeout on the read or write and close the TCP connection. This counter will not continually increment. The Network Bitmap Status will reflect the missing PLC.
5) Read Errors – Number of read error responses sent from the PLC to the gateway.
6) Write Requests – Number of write requests sent from the gateway to the PLC (G2N).
7) Write Responses – Number of valid write responses sent from the PLC to the gateway.
8) Write Errors – Number of write error responses sent from the PLC to the gateway.
9) Read Request to Response Time – Number of milliseconds it took the PLC to reply to a request
10) Read Response to Request Time – Number of milliseconds it took the gateway to execute the next request once the previous response has been received.
11) High Priority Read Loop Time – Number of milliseconds it took to execute all high priority read requests.
12) Low Priority Read Loop Time – Number of milliseconds it took to execute all low priority read requests.
13) Write Request to Response Time – Number of milliseconds it took the PLC to reply to a request.
14) Write Response to Request Time – Number of milliseconds it took the gateway to execute the next request once the previous response has been received.
15) Write Loop Time – Number of milliseconds it took to execute all write requests.

Common Error Strings - These are the values for *All PLCs,* or the specific PLC selected.

1) IP: xxx.xxx.xxx.xxx "tagname" (04) Path Segment Error – The tag name is wrong, or the tag is not defined as a controller scope tag.
2) IP: xxx.xxx.xxx.xxx "tagname" (08) Service Not Supported – The IP address or the slot number does not match with the PLC the gateway is setup to communicate with.
3) IP: xxx.xxx.xxx.xxx "tagname" (1E) Embedded Server Error – The tag name that is setup within the gateway doesn't match a tag setup in the PLC.
4) IP: xxx.xxx.xxx.xxx "tagname" (ff,2105) Access beyond end of array – Tried to read/write too much data.
5) IP: xxx.xxx.xxx.xxx "tagname" (ff, 2107) Abbreviated type mismatch – The data type of the tag, on a write, in the gateway doesn't match the tag in the PLC.

Diagnostic – MQTT Client

Select the MQTT Client in the dropdown menu on the Diagnostic page to view the breakdown of the diagnostics and common strings that are display on the page. You may also view the individual MQTT device counters by selecting the device in the All Devices drop down and clicking **View**.



NOTE: This page will auto-refresh every five seconds.

Clear All Values: This will only affect the Variables values.

- This will return all values displayed to zero. Example: If viewing QT02 192.168.0.101, this will only clear the values for that specific device.

Device Status: This will only display when viewing All Servers.
Connected: All Devices configured/enabled are communicating
Not Connected: Fatal Error No Configuration

- No Devices that have been configured are enabled
- No Devices that have been configured and enabled have topics configured

Not Connected: Dependency Protocol is Faulted

- The Dependency Protocol has Faulted

Error: Timeout

- One or more enabled devices are missing
- Verify MQTT broker for correct IP address.

LED Status

Solid Green (Connected):

- The gateway is connected to all the MQTT devices that are configured and enabled

Flashing Green (Not Connected):

- No MQTT devices are configured / enabled. Go to the MQTT Client Device Configuration to configure a device

Flashing Red (Not Connected):

- One or more of the MQTT brokers configured are missing (nodes missing)
- One or more of the MQTT brokers configured do not have topics configured
- The Dependency Protocol has faulted

Flashing Red (Communication not attempted yet):

- No topics are configured and data needed for writes isn't valid yet

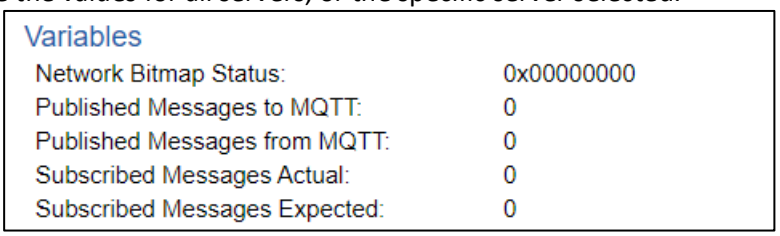
Solid Red (Invalid Configuration):

- No devices are enabled
- One or more of the MQTT devices have a conflicted IP address

Off:

- No Power
- No Ethernet cable connected

Variables: These are the values for all servers, or the specific server selected.



Network Bitmap Status (Displayed in Hex):

- Each bit corresponds to a MQTT device. If the bit is set, the MQTT device is connected, otherwise the bit is 0.
- Bit 0 corresponds to MQTT device 1 and Bit 2 is for MQTT device 3 and so on.

Published Messages to MQTT:

- Number of Write Topics which have been sent to the MQTT broker

Published Messages from MQTT:

- Number of Read Topics which have been sent from the MQTT broker to the gateway

Subscribed Messages Actual:

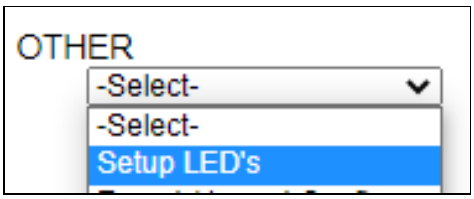
- Number of Successful Subscribed Topics
- This should equal the Subscribed Messages Expected

Subscribed Messages Expected:

- Number of Subscribed Topics that the gateway should have open

LED Configuration

To modify the behavior of the LEDs on the 460 gateway, navigate to **Other->Setup LEDs**.



Each LED may be set to Disabled, Protocol 1, or Protocol 2. If either protocol is a master/client, you may set the LED to represent either all slaves/servers configured in the gateway or a slave/server device.

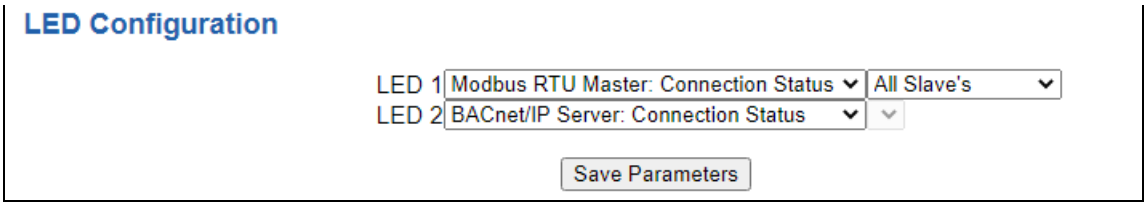
To select a slave/server device:

1) Select the protocol in the left dropdown menu.

2) Click **Save Parameters** to generate the second dropdown menu.

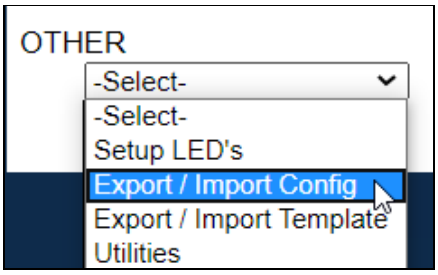
3) Select the individual slave/server in the right dropdown menu.

Click the **Save Parameters** button to commit the changes and reboot the gateway.



Configuration Files

To access the configuration file in the 460 gateway, select the dropdown **Other->Export/Import Config**.



Export Configuration

The Export Configuration allows you to save your configuration file for backup or to be imported into another gateway. This file is named *rta_cfg.rtax* by default.

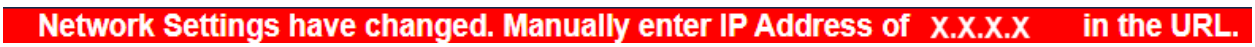
Upon clicking the **Save Configuration to File** button, you will be prompted to select a location to save the file. Different web browsers will yield different looks.

Import Configuration

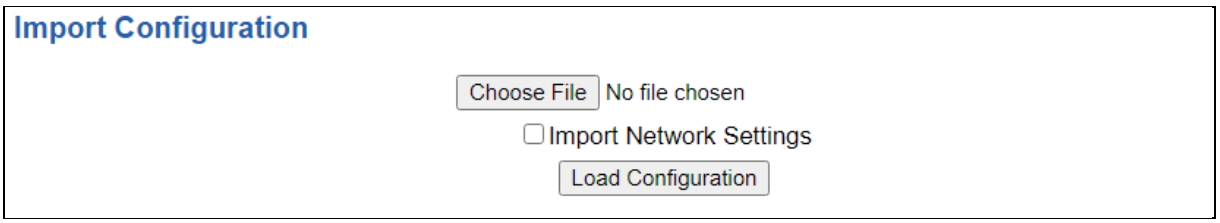
You can import a previously exported configuration file or a configuration file from another device into the 460 gateway, whenever it is in Configuration Mode.

Upon clicking the **Choose File** button, you will be prompted to select a location from which to load the saved file. Once the location is selected, you can choose the **Import Network Settings** checkbox if you want to load the network settings of the configuration file or just load the configuration without the network setting.

If you choose to Import Network Settings, this will override your current gateway's network setting with the settings in the configuration file. After you click on the Load Configuration button, a banner will display your gateway's new IP address.



If the configuration has successfully loaded, the gateway will indicate that it was successful, and a message will appear under the Load Configuration button indicating Restart Needed.



If it encountered an error while trying to load the saved configuration, the gateway will indicate the first error it found and a brief description about it under the Load Configuration button. Contact RTA Support with a screenshot of this error to further troubleshoot.

Save and Replace Configuration Using SD Card

Saving Configuration Using SD Card

This function saves the gateway's configuration automatically to an SD Card each time the gateway is rebooted via the **Restart Now** button on the web page. If this unit should fail in the future, the last configuration stored on the SD card and can be used for a new gateway to get the application back up and running quickly.

This SD Card replaces every configurable field in the gateway, **EXCEPT** for IP Address, Subnet Mask, and Default Gateway.

Replacing Configuration Using SD Card

To replace a configuration in a gateway using the SD Card, a specific sequence of events must be followed for the replacement to happen correctly:

1) Extract SD Card from gateway you wish to copy the configuration from.
2) Power up the gateway you wish to copy the configuration to. DO NOT INSERT SD CARD YET.
3) Navigate to the webpage inside the unit.
4) Navigate to the dropdown **Other->Utilities**.
5) If you are not currently in *Mode: Configuration*, go into Configuration Mode by clicking the **Configuration Mode** button at the top left-hand side of the screen.
6) Press the **Revert to Manufacturing Defaults** button on the Utilities Page. The Configuration will ONLY be replaced by the SD Card if the gateway does not have a configuration already in it.
7) When the unit comes back in *Mode: Running*, insert the SD Card.
8) Do a hard power cycle to the unit by unplugging power. DO NOT RESET POWER VIA WEB PAGES.
    a. It will take an additional 30 seconds for the unit to power up while it is transferring the configuration. During this time, the gateway cannot be accessed via the web page.
9) When the unit comes back up, the configuration should be exactly what was on the SD Card.

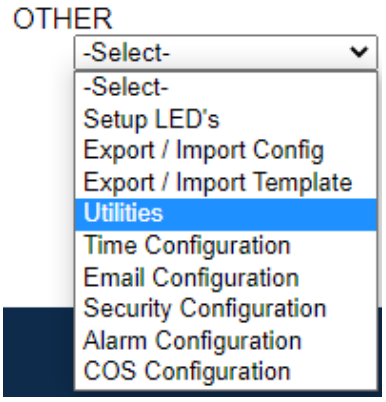
Intelligent Reset Button

If the IP Address of the gateway is forgotten or is unknown, there is an easy way to recover the IP Address using a reset button on the hardware.

1) On the side of the gateway with the SD card slot, there is a small pinhole. Using a paperclip, press the button through this pinhole and hold the button for at least 5 seconds.
2) After 5 seconds, the unit will acknowledge the command and LED 1 and LED 2 will start an alternate Blink Green quickly pattern.
3) Release the button and the gateway will reset to default IP settings (DHCP).

Utilities

To access the Utilities page in the 460 gateway, navigate to **Other->Utilities**. The Utilities screen displays information about the gateway including Operation Time, File System Usage, Memory Usage, and Memory Block Usage.



Here you can also:

- View the full revision of the software.
- View all the files stored in the Flash File System within the gateway.
- Identify your device by clicking the **Start Flashing LEDs** button. By clicking this button, the two diagnostic LEDs will flash red and green. Once you have identified which device you are working with, click the button again to put the LEDs back into running mode.
- Configure the size of the log through the Log Configuration.
- Bring the device back to its last power up settings.
- Bring the device back to its original manufacturing defaults.
- Remove the Configuration File and Flash Files within the gateway.