



LoRaWAN Driver (Milesight) for Tridium Niagara 4 Technical Guide

Date 14/08/2024

Revision 1.2

TYRRELL PRODUCTS LTD

INTRODUCTION	3
LICENSING & SOFTWARE MAINTENANCE	8
DRIVER INSTALLATION	10
MILESIGHT SETUP	11
MILESIGHT APPLICATIONS	12
LORAWAN SENSORS	14
NIAGARA WEB SERVICE	17
LORAWAN DRIVER	18
NETWORK SERVER	19
DEVICE MANAGER	20
POINT DISCOVERY	22
SUPPORTED DEVICES	23
LORAWAN DEVICE PAYLOAD DE-CODING	25
NIAGARA GENERIC JSON	30
REVISION HISTORY	31

INTRODUCTION

The LoRaWAN Driver can be used to provide a fast and simple interface to a LoRaWAN based system of devices. The driver can be used from any Niagara Station (Web Sup / JACE / 3rd Party Controller) to a Mulesight LoRaWAN Gateway (such as UG65). This includes Mulesight Gateways on a local a LAN or via a remote cellular connection.

The driver is designed to simplify the integration of LoRaWAN networks into Niagara without having to manually setup and decode JSON payloads. The driver will support 'known' devices and also support new / unknown devices.

The connection to the LoRaWAN Gateway is bidirectional over a HTTP interface.

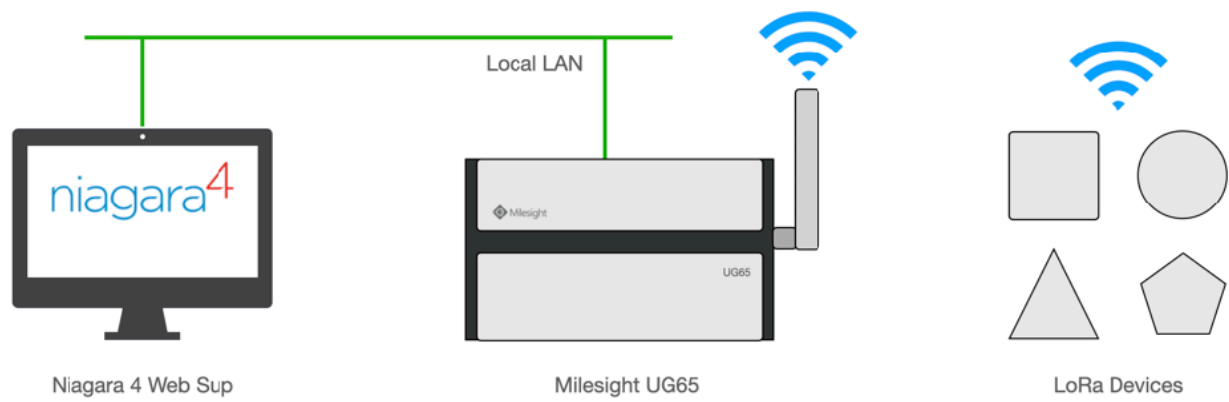
The Driver is compatible with all brands of Niagara 4 (Tridium / Centraline / Distech / Honeywell / JCI / Trend etc).

TYRRELL PRODUCTS LTD

The LoRaWAN Driver can be used in several different scenarios on different Tridium Niagara based platforms via a Milesight Gateway:

- Web Supervisor
- JACE8000 / JACE9000
- Any 3rd Party Controller (IoT Controller / Lynxspring / iSMA MAC36 etc).

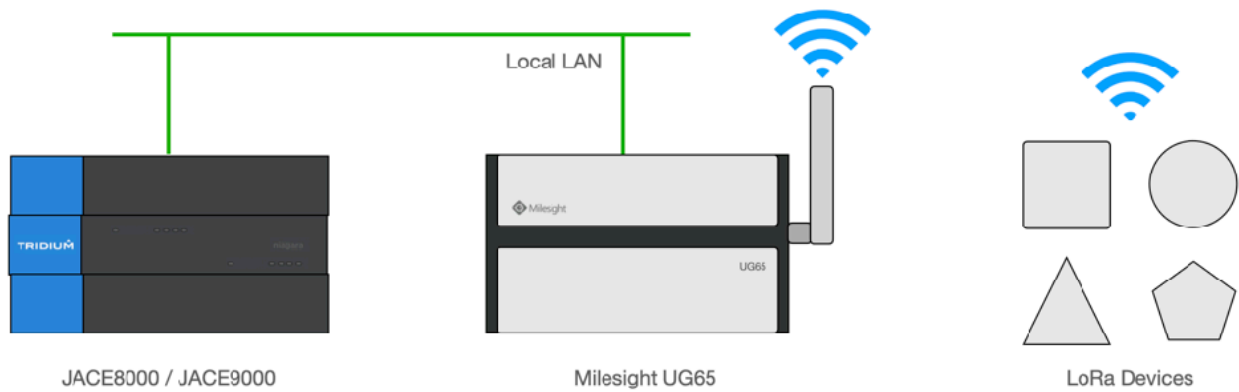
Example 1
Web Supervisor - Milesight UG65 Gateway (Local LAN)



The Web Supervisor integrates directly to the local Milesight UG65 Gateway.

All configured LoRaWAN devices will be discoverable in the Niagara Station and their enclosed points will also be discoverable.

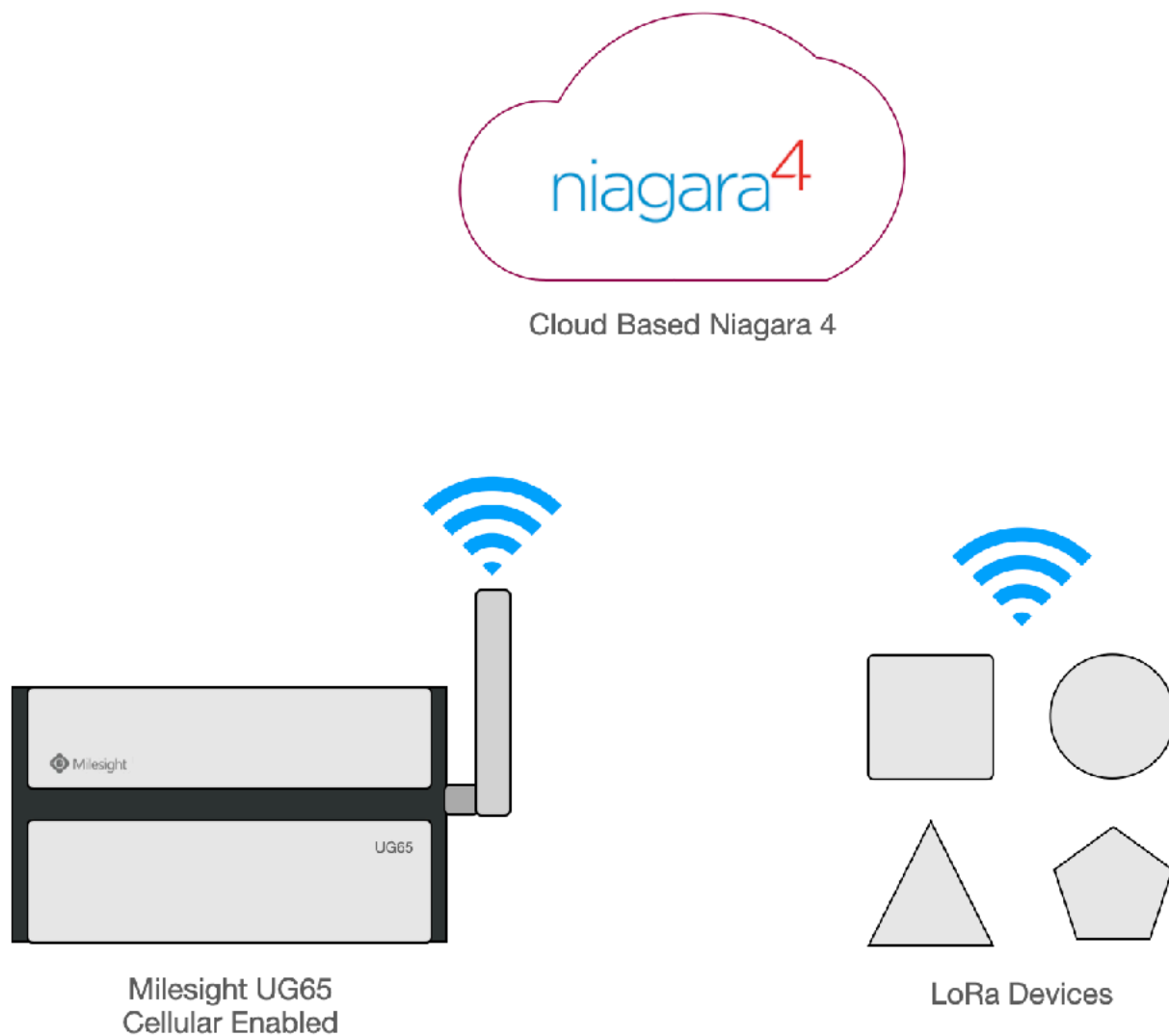
**Example 2:
Niagara Enabled Controller - Milesight UG65 Gateway (Local LAN)**



The JACE8000 / 3rd Party Controller integrates directly to the local Milesight UG65 Gateway.

All configured LoRaWAN devices will be discoverable in the Niagara Station and their enclosed points will also be discoverable.

**Example 3:
Cloud Based Niagara 4 - Remote Milesight UG65 Gateway (Cellular)**



The Web Supervisor integrates remotely to the remote Milesight UG65 Gateway (with integrated SIM card).

All configured LoRaWAN devices will be discoverable in the Niagara Station and their enclosed points will also be discoverable.

LICENSING & SOFTWARE MAINTENANCE

The LoRaWAN driver is licensed based on points. Each LoRaWAN point will also consume one Global Capacity Point License.

You will need to provide your Niagara 4 Host ID as part of your purchase. If you are expanding your system in the future you will need to ensure that your LoRaWAN Driver has been expanded to cover the number of new points being added.

Once the license has been generated you can re-import your niagara license files from the Platform > License Manager providing you have an internet connection, alternatively you can be emailed a copy of the new license files.

The LoRaWAN Driver includes a software maintenance feature. Every new purchase of the driver will support the current release of Niagara 4 and the next release of Niagara 4, any subsequent upgrades will require a software maintenance license to be purchased.

As an example the current release of Niagara 4 is N4.13, a new driver purchase will cover you for N4.13 and a future upgrade to N4.14. Any further upgrades, for example to N4.15 or above, will require a software maintenance license to be updated. The software maintenance license would then cover you for the now current release of Niagara 4 (as an example N4.15). You can upgrade from any previous release with a single software maintenance purchase.

Ensure the target Host License Manager is up to date with a Tyrrell.license and Tyrrell.certifcate containing the required license features.

Any questions or queries in relation to this item should be sent to sales@tyrrellproducts.com

TYRRELL PRODUCTS LTD

LoRaWAN Driver License Packs:

Product Code	Description
LoRaWAN 0025	LoRaWAN Driver 25 Point License Pack 1x LoRaWAN Network 25x LoRaWAN Points
LoRaWAN 0050	LoRaWAN Driver 50 Point License Pack 1x LoRaWAN Network 50x LoRaWAN Points
LoRaWAN 0100	LoRaWAN Driver 100 Point License Pack 1x LoRaWAN Network 100x LoRaWAN Points
LoRaWAN 0250	LoRaWAN Driver 250 Point License Pack 1x LoRaWAN Network 250x LoRaWAN Points
LoRaWAN 0500	LoRaWAN Driver 500 Point License Pack 1x LoRaWAN Network 500x LoRaWAN Points

LoRaWAN Driver Upgrade Packs:

LoRa0025-UPG	25x LoRaWAN Points
LoRa0050-UPG	50x LoRaWAN Points
LoRa0100-UPG	100x LoRaWAN Points
LoRa0250-UPG	250x LoRaWAN Points
LoRa0500-UPG	500x LoRaWAN Points

DRIVER INSTALLATION

The LoRaWAN Driver supports Niagara 4.10 and above.

NOTE:

If your installation is running an older version of the Niagara software then it must be upgraded to meet the above requirements to run this service.

Any future updates to the LoRaWAN Service will be available for the current release and previous Niagara 4 release. All other releases will become legacy and unsupported.

Niagara 4 Installation:

You will need the version specific JAR files for your Niagara 4 installation. These can be downloaded from the Customer Portal or alternatively contact support.

To install the Service copy the below JARS to c:\niagara\niagara 4.x.xx\modules

- ▶ LoRaWAN-rt.jar
- ▶ LoRaWAN-wb.jar

Once the files have been put into the correct directory close your workbench, and relaunch. Any running Stations on the local machine will have to be re-started to make use of the LoRaWAN Driver.

The LoRaWAN Driver is now ready to use in a local station or to commission / update a JACE. To install the driver on a JACE use the Commissioning Wizard on the platform of the target device.

MILESIGHT SETUP

The Milesight Gateway will need to have a specific configuration applied:

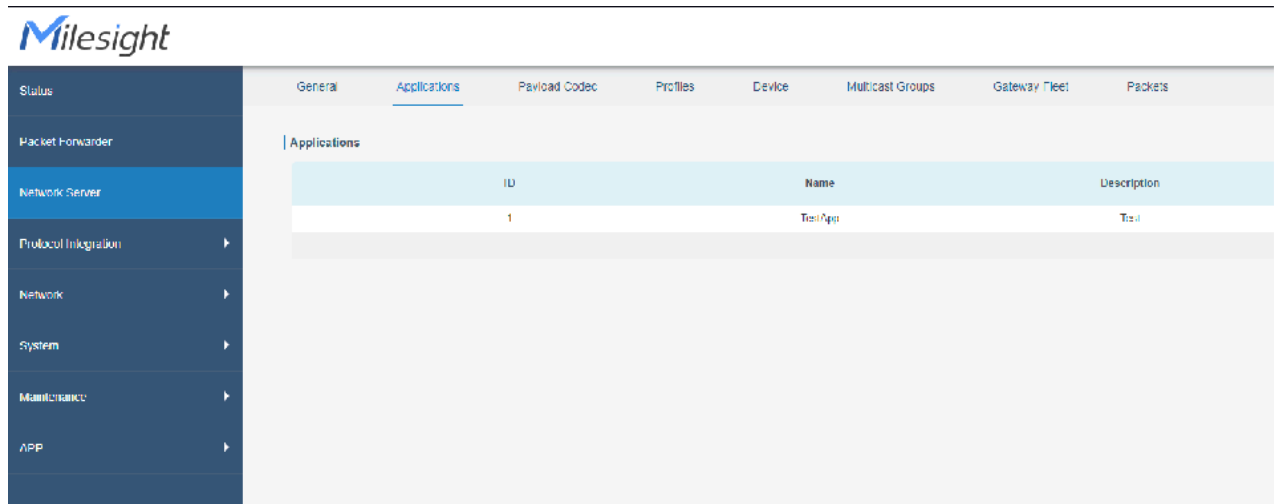
- Create An Application(s)
- Assign LoRaWAN Devices to the Application(s)

The integration between Niagara 4 and Milesight gateway will support HTTP connections only. This is a current limitation of the Milesight Gateways.

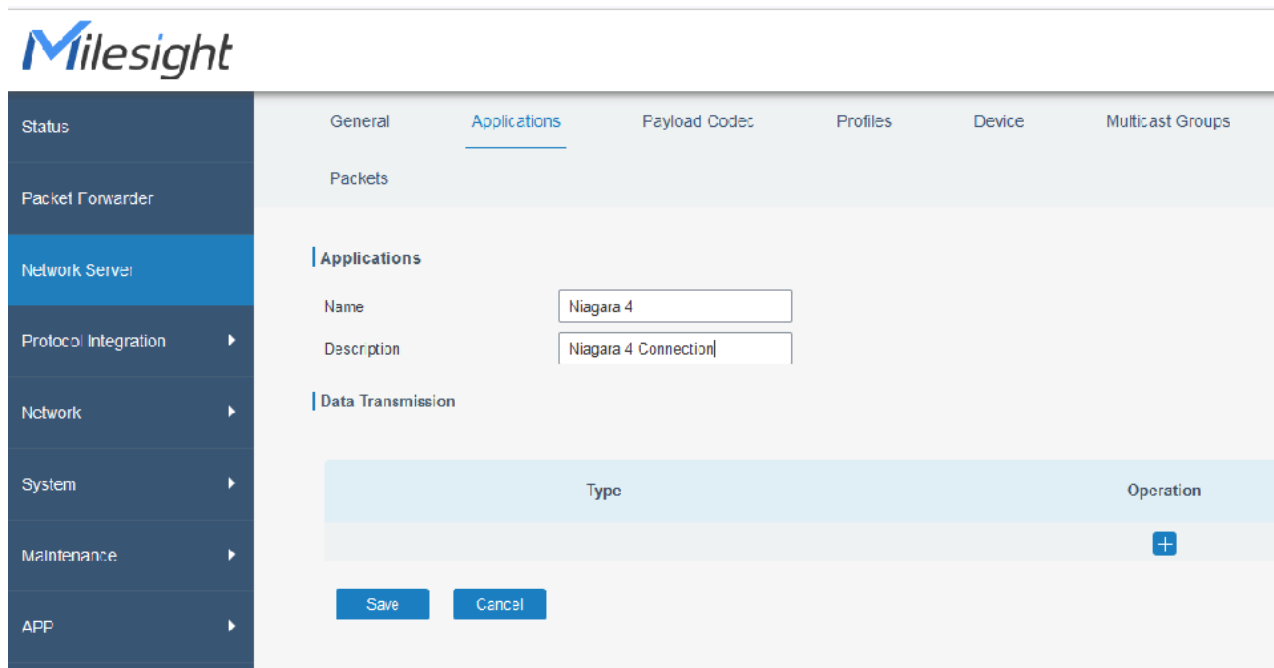
MILESIGHT APPLICATIONS

Login to your Milesight Gateway (the default details are printed on the rear of the unit).

Navigate to **Network Server > Applications**



Create a new **Application** and save it.



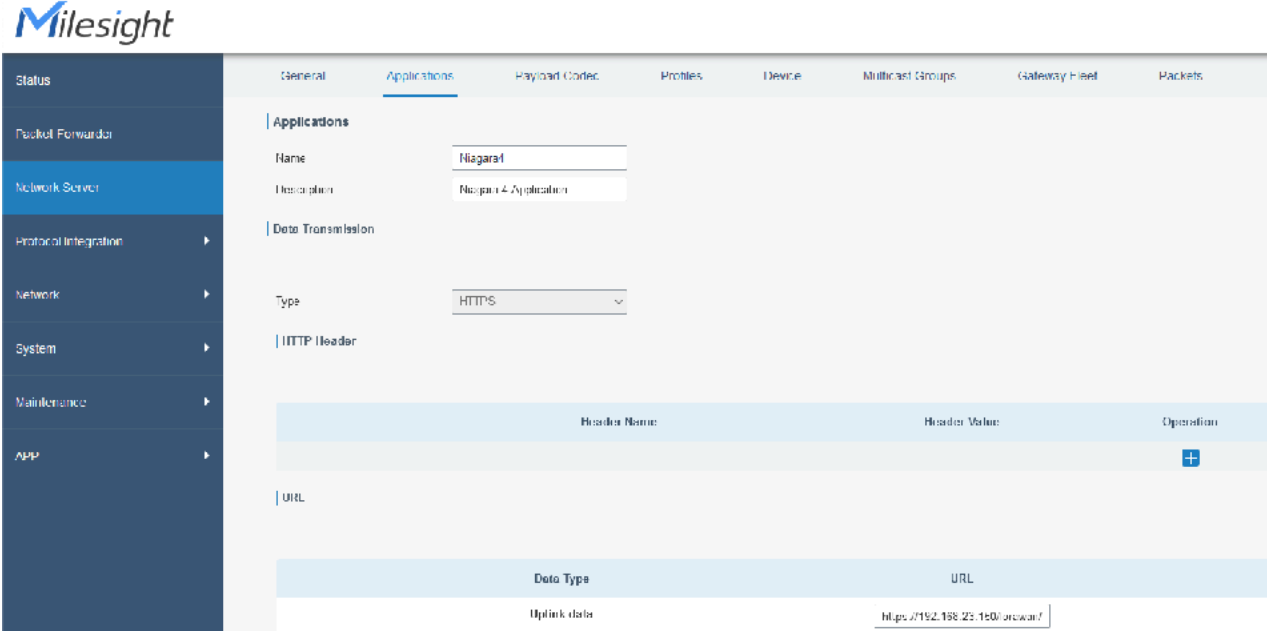
TYRRELL PRODUCTS LTD

Re-edit the Application and a new **Operation - HTTP**

Edit the **Uplink Data** field to be `http://IPADDRESS/lorawan/noAuthListener/uplink`.

Where *IPADDRESS* is the address of your Niagara 4 Station. Example:

`http://192.168.23.10/lorawan/noAuthListener/uplink`



The screenshot shows the Milesight web interface. On the left is a dark blue sidebar with navigation options: Status, Packet Forwarder, Network Server (highlighted), Protocol Integration, Network, System, Maintenance, and APP. The main content area has tabs for General, Applications (selected), Payload Codec, Profiles, Device, Multicast Groups, Gateway Fleet, and Packets. Under the Applications tab, the configuration for an application named 'Niagara4' is shown. The Name field is 'Niagara4' and the Description is 'Niagara 4 Application'. Under Data Transmission, the Type is set to 'HTTPS'. Under the HTTP Header section, there is a table with columns for Header Name, Header Value, and Operation. A plus sign (+) is visible in the Operation column. Under the URL section, there is a table with columns for Data Type and URL. The Data Type is 'Uplink data' and the URL is 'http://192.168.23.10/lorawan/'.

Save both the Operation and Application, then return to the main menu.

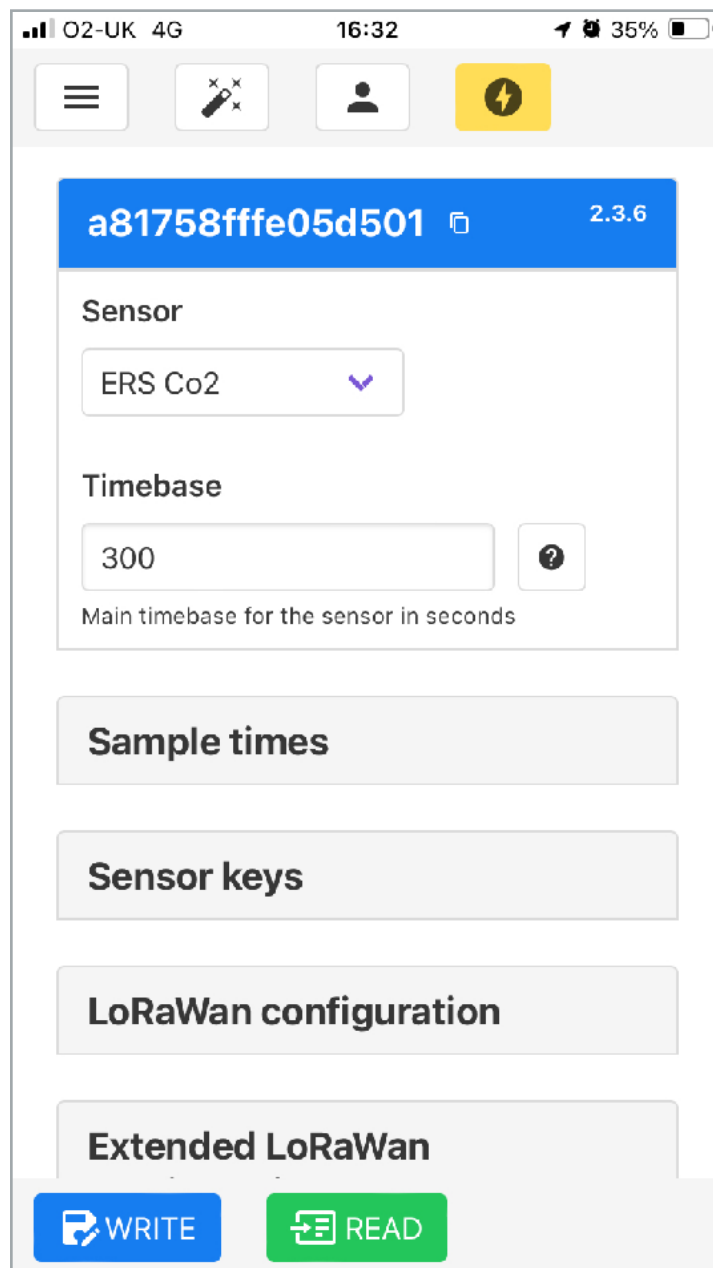
LORAWAN SENSORS

Before configuring the Niagara Station you will need to ensure that your Milesight Gateway is paired with at least one LoRaWAN Device.

You may require a dedicate Mobile Phone App to configure the sensor, the guide will demonstrate an Elsys sensor.

Download the Sensor App onto your Mobile Device.

Note: Your mobile device must be fitted with a NFC chip to work.



Make note of the Sensor UID (**a81758fffe05d501**)

TYRRELL PRODUCTS LTD

In the Milesight Gateway navigate to **Network Server > Device**

Add a new Device

Device Name	Elsys 3in1
Description	Technicals Desk
Device EUI	a81758ffe05d501
Device-Profile	ClassA-OTAA ▾
Application	Niagara4 ▾
Payload Codec	None ▾
fPort	1
Frame-counter Validation	<input type="checkbox"/>
Application Key	328d1a19bd244d3b129ba1514fd
Device Address	
Network Session Key	
Application Session Key	
Uplink Frame-counter	0
Downlink Frame-counter	0

Setting	Description
Device Name	Unique Name of the Device
Description	User Friendly Description
Device EUI	From the Sensors App
Device Profile	From Spec Sheet Of Device
Application	Your Custom Application in the previous step
Payload Codec	None By Default Can be changed for new / unknown devices (see later section)
App Key	From the Sensors App Milesight Default is: 5572404c696e6b4c6f52613230313823



A

TYRRELL PRODUCTS LTD

Once you have added a device ensure it is active and sending data before proceeding.

Once working the **Last Seen** and **Activated** fields will update.

The screenshot displays a web application interface for managing devices. At the top right, there is a user profile icon labeled 'admin' and a refresh icon. Below this is a navigation menu with tabs: 'General', 'Applications', 'Payload Codec', 'Profiles', 'Device' (selected), 'Multicast Groups', 'Gateway Fleet', and 'Packets'. A help icon (?) is also present. The main content area is titled 'Device' and contains three buttons: 'Add', 'Bulk Import', and 'Delete All'. A search bar is located on the right side of the main content area. Below the buttons and search bar is a table with the following columns: 'Device Name', 'Device ID', 'Device Profile', 'Application', 'Last Seen', 'Activated', and 'Operation'. The table contains one row with the following data: 'Hoya 3m1', 'A817E8F7FF0E1AD1', 'Class A D/A', 'Nagios4', '28 seconds ago', a green checkmark, and a blue edit/delete icon. At the bottom left of the table, it says 'Showing 1 of 1 items'.

Device Name	Device ID	Device Profile	Application	Last Seen	Activated	Operation
Hoya 3m1	A817E8F7FF0E1AD1	Class A D/A	Nagios4	28 seconds ago	✓	 

NIAGARA WEB SERVICE

Connect to your Niagara 4 Station.

Navigate to **Station > Services > Web Service**.

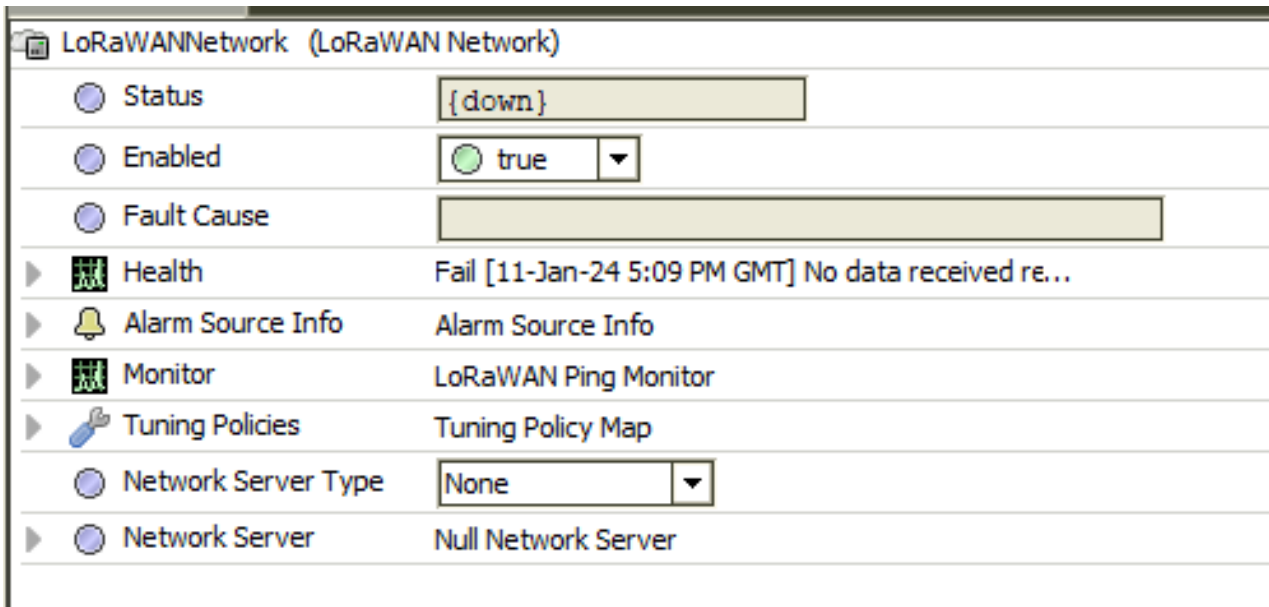
Set the **HTTP** option to **TRUE** and Ensure **HTTPS ONLY** is set to **FALSE**.

WebService (Web Service)	
Status	{ok}
Fault Cause	
Enabled	<input checked="" type="radio"/> true ▼
▶ Http Port	80 tcp
Http Enabled	<input checked="" type="radio"/> true ▼
▶ Https Port	443 tcp
Https Enabled	<input checked="" type="radio"/> true ▼
Https Only	<input type="radio"/> false ▼

LORAWAN DRIVER

Navigate to Station > Config > Drivers and add a new LoRaWAN Driver.

Navigate to the AX Property Sheet view of the Network.



Change the **Network Server** Type to **Milesight UG65**

NETWORK SERVER

Expand the **Network Server** Section

Property Sheet

📁 LoRaWANNetwork (LoRaWAN Network)

⊙ Status

⊙ Enabled true ▼

⊙ Fault Cause

▶ 🏠 Health Fail [11-Jan-24 5:09 PM GMT] No data received re...

▶ 🔔 Alarm Source Info Alarm Source Info

▶ 🏠 Monitor LoRaWAN Ping Monitor

▶ 🔧 Tuning Policies Tuning Policy Map

⊙ Network Server Type ▼

▼ ⊙ Network Server Milesight UG65 Network Server

⊙ Url

⊙ Username

⊙ Password

Note: The user account **MUST** be the admin account.

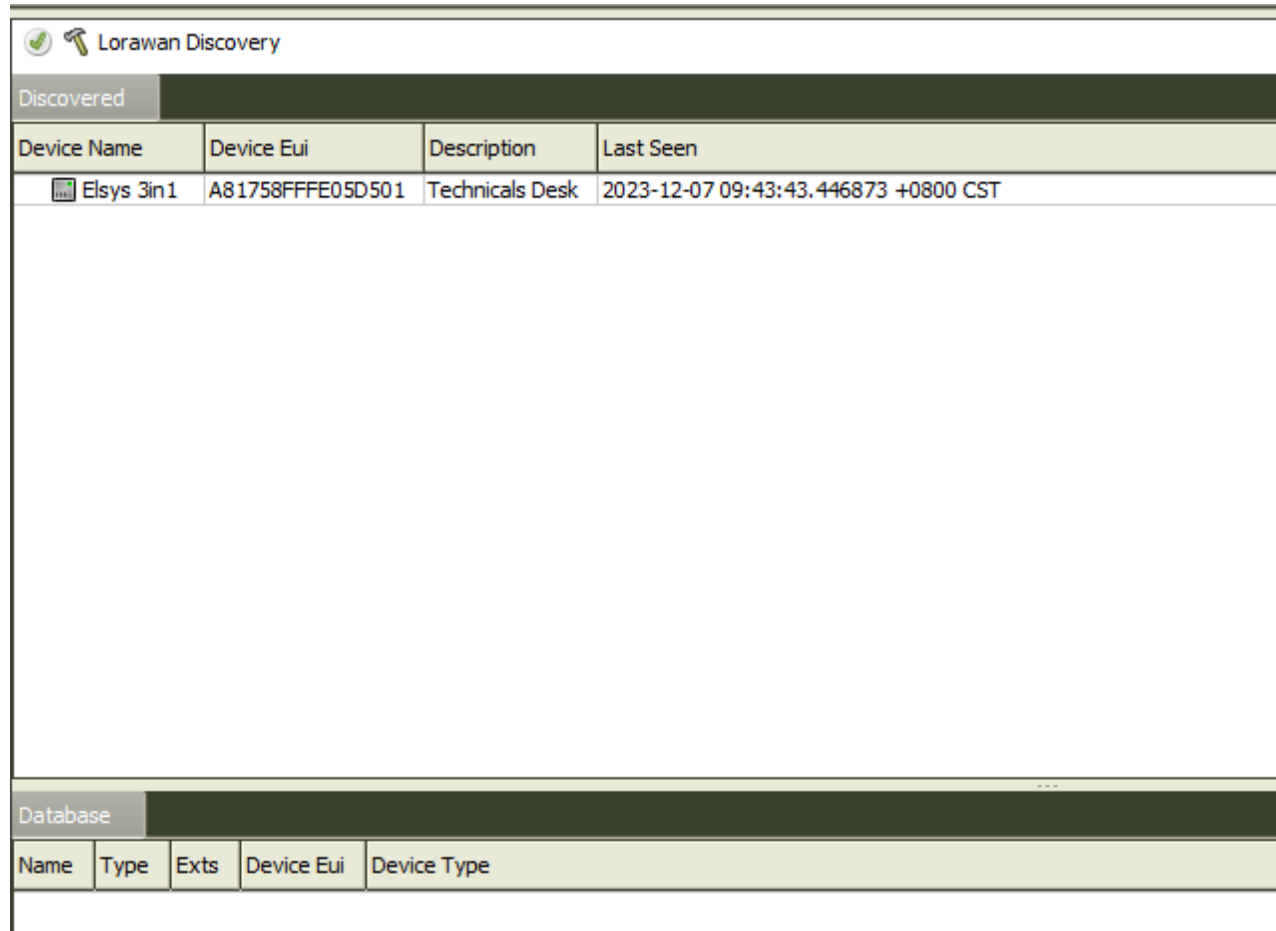
Setting	Description
URL	IP of the Milesight Gateway http://192.168.23.150
Username	admin
Password	Admin account password

TYRRELL PRODUCTS LTD


DEVICE MANAGER

Navigate to the **LoRaWAN Driver > Device Manager** and press **Discover**.

The driver will then discover all available devices from the Milesight Gateway



The screenshot displays the 'Lorawan Discovery' interface. At the top, there is a 'Discovered' section with a dark header. Below this is a table with the following data:

Device Name	Device Eui	Description	Last Seen
 Elsys 3in1	A81758FFFE05D501	Technicals Desk	2023-12-07 09:43:43.446873 +0800 CST

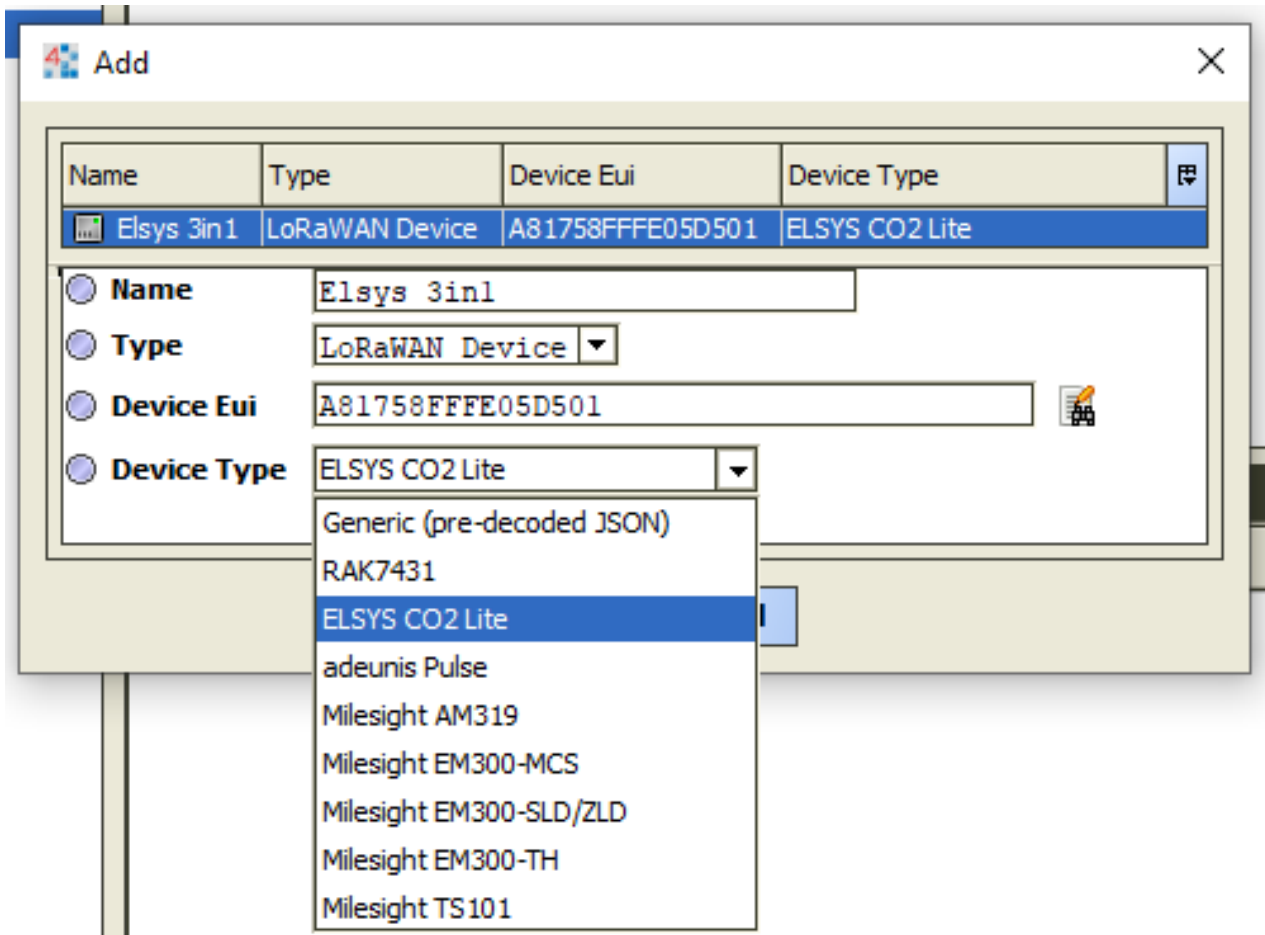
Below the discovered devices is a 'Database' section with a dark header. Below this is a table with the following columns:

Name	Type	Exts	Device Eui	Device Type
------	------	------	------------	-------------

Add the required device(s) to the Station database.

TYRRELL PRODUCTS LTD

When adding a device the **Add Window** will have a **Device Type** option. The driver has a pre-configured library of devices that will automatically 'work'. **New / Unknown** devices can still be added but need to be set with a **Generic** profile and the JSON payload decode. Refer to the section **Supported Devices** for more information.



Once the device has been added navigate to the **Points** container of the device.

TYRRELL PRODUCTS LTD

POINT DISCOVERY

In the Point Manager Window press the Discover button and all available points will be presented.

Lorawan Discovery					
Discovered					
Point Name	Display Name	Point Type	Json Field	Device Facets	Point Facets
temperature	Temperature	control:NumericPoint	temperature	precision=1,units=°C	precision=1,units=°C
humidity	Humidity	control:NumericPoint	humidity	precision=0,units=%	precision=0,units=%
co2	CO2	control:NumericPoint	co2	precision=0,units=ppm	precision=0,units=ppm
vdd	Battery	control:NumericPoint	vdd	precision=0,units=mV	precision=0,units=mV

Add the required Points to the Station Database.

Database					
Name	Type	Out	Enabled	Device Facets	Json Field
temperature	Numeric Point	0.0 °C {stale}	true	precision=1,units=°C	temperature
humidity	Numeric Point	0 % {stale}	true	precision=0,units=%	humidity
co2	Numeric Point	0 ppm {stale}	true	precision=0,units=ppm	co2
vdd	Numeric Point	0 mV {stale}	true	precision=0,units=mV	vdd

The Points will all remain STALE until the LoRaWAN sensor next reports to the Gateway. Check the device configuration and either way for the next publish cycle or change the devices config to report more frequently during setup.

Database					
Name	Type	Out	Enabled	Device Facets	Json Field
temperature	Numeric Point	22.9 °C {ok}	true	precision=1,units=°C	temperature
humidity	Numeric Point	39 % {ok}	true	precision=0,units=%	humidity
co2	Numeric Point	1669 ppm {ok}	true	precision=0,units=ppm	co2
vdd	Numeric Point	3627 mV {ok}	true	precision=0,units=mV	vdd

The points will then only update their values when the LoRaWAN sensor reports to the LoRaWAN Gateway (for example every 5 mins).

SUPPORTED DEVICES

The LoRaWAN driver automatically supports the following device types:

- Adonis Pulse Counter
2x Configurable Pulse Counter Inputs
- B Meters RFM-LR1
For pre-equipped single jet water meters
- ElSys CO2 Lite (3 in 1)
Temp / Hum / CO₂
- Milesight AM102L
Temp / Hum
- Milesight AM103L
Temp / Hum / CO₂
- Milesight AM307
7 in 1 Sensor
Temp / Hum / Motion / Light / TVOC / Barometric Press / CO₂
- Milesight AM308
9 in 1 Sensor
Temp / Hum / Motion / Light / TVOC / Pressure / CO₂ / PM2.5 / PM10
- Milesight AM319
11 in 1 Sensor
Temp / Hum / Motion / Light / TVOC / Pressure / CO₂ / PM2.5 / PM10 / (HCHO)² / (O₃)²
- Milesight EM300 DI
Temp / Hum / DI or Pulse Counter
- Milesight EM300 MCS
Temp / Hum / Magnetic Switch
- Milesight EM300 SLD/ZLD
Temp / Hum / Leak Detection
- Milesight EM300 TH
Temp / Hum
- Milesight TS101
Insertion Temp
- Milesight TS201
Flying Lead Temperature Probe
- Milesight TS301
1x Connector for PT100 Sensor OR Magnetic Switch

TYRRELL PRODUCTS LTD

- Milesight TS302
2x Connector for PT100 Sensor OR Magnetic Switch
- Milesight UC300
IO Controller
4* DI / 2* DO / 2* 4-20mA / 2* 0-10v / 2* PT100 Sensor
*RS485 NOT CURRENTLY SUPPORTED!
- Milesight VS350
Passage People Counter
- Milesight WT101* (*In Development*)
Smart Radiator Thermostat

Other devices are supported in one of two ways:

- Use the 'Generic' profile and decode the incoming JSON
- For Sensor/Device Types that are not listed in the LoRaWAN Niagara driver, please ask Tyrrell Products Ltd about adding your LoRaWAN Device type to the driver's drop-down list.

This will greatly simplify the process of adding devices in the future.

TYRRELL PRODUCTS LTD

LORAWAN DEVICE PAYLOAD DE-CODING

New / unknown LoRaWAN device types can still be supported by using the “Generic (pre-decoded JSON)” Device Type drop-down list option in combination with some modifications to the device manufacturer supplied JavaScript “Payload Decoder Function”.

This example will cover a EM300-TH.

You will need the manufacturers decoder script.

As an example:

https://github.com/Milesight-IoT/SensorDecoders/blob/main/EM_Series/EM300_Series/EM300-TH/EM300-TH_Chirpstack.js#L1

TYRRELL PRODUCTS LTD

```
/**
 * Payload Decoder for Milesight Network Server
 *
 * Copyright 2023 Milesight IoT
 *
 * @product EM300-TH
 */
function Decode(fPort, bytes) {
    return milesight(bytes);
}

function milesight(bytes) {
    var decoded = {};

    for (var i = 0; i < bytes.length; ) {
        var channel_id = bytes[i++];
        var channel_type = bytes[i++];

        // BATTERY
        if (channel_id === 0x01 && channel_type === 0x75) {
            decoded.battery = bytes[i];
            i += 1;
        }
        // TEMPERATURE
        else if (channel_id === 0x03 && channel_type === 0x67) {
            // °C
            decoded.temperature = readInt16LE(bytes.slice(i, i + 2)) / 10;
            i += 2;

            // °F
            // decoded.temperature = readInt16LE(bytes.slice(i, i + 2)) / 10 * 1.8 + 32;
            // i += 2;
        }
        // HUMIDITY
        else if (channel_id === 0x04 && channel_type === 0x68) {
            decoded.humidity = bytes[i] / 2;
            i += 1;
        }
        // TEMPERATURE & HUMIDITY HISTROY
        else if (channel_id === 0x20 && channel_type === 0xce) {
            var point = {};
            point.timestamp = readUInt32LE(bytes.slice(i, i + 4));
            point.temperature = readInt16LE(bytes.slice(i + 4, i + 6)) / 10;
            point.humidity = bytes[i + 6] / 2;

            decoded.history = decoded.history || [];
            decoded.history.push(point);
            i += 8;
        } else {
            break;
        }
    }
}
```

return decoded;

More JSON That has been chopped for the sake of convenience.

You will need to copy the whole of the JS to Notepad++ and modify the first part

TYRRELL PRODUCTS LTD

Original Example:

```
/**
 * Payload Decoder for Milesight Network Server
 *
 * Copyright 2023 Milesight IoT
 *
 * @product EM300-TH
 */
function Decode(fPort, bytes) {
    return milesight(bytes);
}

function milesight(bytes) {
    var decoded = {};
}
```

Modified Example

```
/**
 * Payload Decoder for Milesight Network Server
 *
 * Copyright 2023 Milesight IoT
 *
 * @product EM300-TH
 */
function Decode(fPort, bytes) {
    var decoder = {};
    decoder.devEUI = LoRaObject.devEUI;
    decoder.dataJson = milesight(bytes);
    decoder.time = LoRaObject.time;
    return decoder;
}

function milesight(bytes) {
    var decoded = {};
}
```

Information

The entry `decoder.dataJson = milesight(bytes);` must match up with the next section line **function milesight(bytes).**

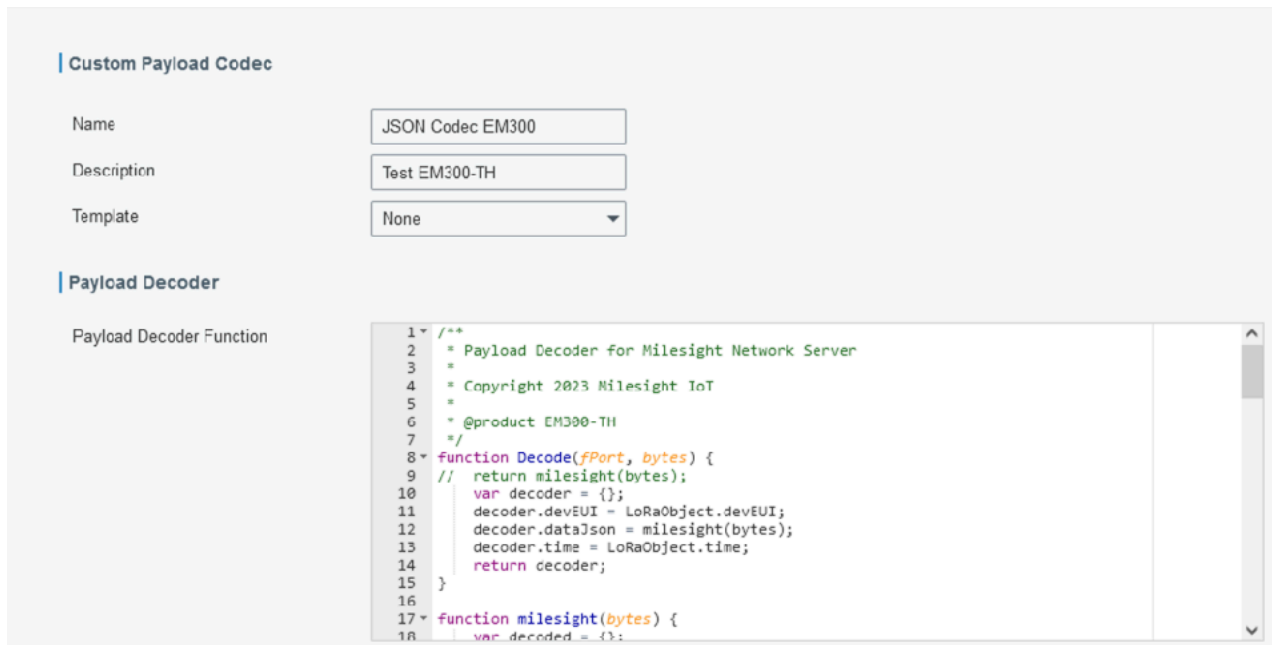
Once modified copy with whole of JS to apse into the milesight gateway.

TYRRELL PRODUCTS LTD

Open the Milesight Web Server and login.

Navigate to **Network Server > Payload Codec**

Add a new **Custom Payload Codec**.



Custom Payload Codec

Name: JSON Codec EM300

Description: Test EM300-TH

Template: None

Payload Decoder

Payload Decoder Function

```
1  /**
2  * Payload Decoder for Milesight Network Server
3  *
4  * Copyright 2023 Milesight IoT
5  *
6  * @product EM300-TH
7  */
8  function Decode(fPort, bytes) {
9  // return milesight(bytes);
10     var decoder = {};
11     decoder.devEUI = LoRaObject.devEUI;
12     decoder.dataJson = milesight(bytes);
13     decoder.time = LoRaObject.time;
14     return decoder;
15 }
16
17 function milesight(bytes) {
18     var decoded = {};
```

Paste the modified JSON into the **Payload Decoder Field**

Save the custom codec.

TYRRELL PRODUCTS LTD

Navigate to **Network Server > Device**

Edit the Device in question

Change the **Payload Codec** from None to **JSON Codec EM300** (or appropriate Codec Name)

MS_EM300 ✕

Device Name	<input type="text" value="MS_EM300"/>
Description	<input type="text" value="TH300"/>
Device EUI	<input type="text" value="24E124136B327698"/>
Device-Profile	<input type="text" value="ClassA-OTAA"/>
Application	<input type="text" value="Niagara4"/>
Payload Codec	<input type="text" value="JSON Codec EM300"/>
fPort	<input type="text" value="1"/>
Frame-counter Validation	<input type="checkbox"/>
Application Key	<input type="text" value="5572404c696e6b4c6f52613230c"/>
Device Address	<input type="text" value="06097d82"/>
Network Session Key	<input type="text" value="f1fb78eab56b278b062ff17edd0d"/>
Application Session Key	<input type="text" value="e1cf7cdb8bc22486270ca8c818e"/>
Uplink Frame-counter	<input type="text" value="29"/>
Downlink Frame-counter	<input type="text" value="2"/>

Save and Apply the change.

TYRRELL PRODUCTS LTD

NIAGARA GENERIC JSON

Return to the Niagara Station and discover the LoRaWAN devices.

Add the target device handset the **Device Type** to **Generic (pre-coded JSON)**

The Point Discovery will **not** work and the points will have to be manually programmed based on their JSON payload names. These can all be obtained from the Payload Decoder.

Name	Type	Facets	Fault Cause	Enabled	Device Facets	Tuning Policy Name	Read Value
Temp	Numeric Point	units=null,precision=1,min=-inf,max=+inf		true		defaultPolicy	24.40 {ok}

<input checked="" type="radio"/> Name	Temp
<input type="radio"/> Type	Cannot edit
<input checked="" type="radio"/> Facets	units=null,precision=1,min=-inf,max=+inf >>
<input checked="" type="radio"/> Fault Cause	<input type="text"/>
<input checked="" type="radio"/> Enabled	<input checked="" type="checkbox"/> true
<input checked="" type="radio"/> Device Facets	>>
<input checked="" type="radio"/> Tuning Policy Name	defaultPolicy
<input checked="" type="radio"/> Read Value	24.40 {ok}
<input checked="" type="radio"/> Write Value	0.00 {ok}
<input checked="" type="radio"/> Json Field	temperature

In the above example the **JSON Field** is **temperature**.

The **Name** and **Facets** can be configured as required based on the point type.

The next time the LoRaWAN device reports to the Gateway the points will update their values.

REVISION HISTORY

REVISION	DESCRIPTION
1.0	Draft Release For Approval
1.1	General Update
1.2	List of Automatically Supported Devices Expanded