Operating instruction manual

netFIELD App OPC UA Client

Hilscher Gesellschaft für Systemautomation mbH
www.hilscher.com

DOC210305OI02EN | Revision 2 | English | 2021-10 | Released | Public

Table of contents

1 About his document

1.1 Description of the contents

This document describes the **netFIELD App OPC UA Client** from Hilscher.

1.2 List of revisions

Index	Date	Author	Revision
1	04-27-2021	MKE	Document created
2	10-26-2021	MKE	Document updated to container version 1.1. Support of OPC UA Server Authentication with client certificates in section *Servers* [▸ page 20] added. Chapter *Deploying the container* [▸ page 9] updated.

Table 1: List of revisions

1.3 Conventions in this document

The term *Edge Device* in this document refers to all devices and virtual machines that are running on the *netFIELD Operating System* (*netFIELD OS*) and thus includes the *netFIELD OS Datacenter* on virtualization platforms.

Notes, operation instructions and results of operation steps are marked as follows:

Notes

Important:
<important note>

Note:
<simple note>

Folgen
<note, where to find further information>

Operation instructions

1. <operational step>
➢ <instruction>
➢ <instruction>

2. <operational step>
➢ <instruction>
➢ <instruction>

Results

⇨ <intermediate result>

⇨ <final result>

2 Brief description

Key features

The **netFIELD App OPC UA Client** enables your Edge Device to collect data from OPC UA servers and to publish it to the local MQTT message bus (via MQTT Broker like *mosquitto*) and/or to the cloud (e.g. to the netFIELD Platform via the *netFIELD App Platform Connector*).

The client can handle multiple OPC UA server sessions and is capable of browsing OPC Address Spaces for quick and easy selection of the data points/nodes to be monitored. It also allows you to define individual sample rates and MQTT publication settings, and provides the monitored data as interoperable and easy-to-process JSON encoded messages.

Like all netFIELD application containers, the OPC UA Client runs in the **IoT Edge Docker** of your netFIELD Edge Device (or netFIELD OS Datacenter) and can be deployed (and configured) from the netFIELD Portal. Information on this can be found in section *Deploying the container* [▸ page 9].

Requirements

- netFIELD Edge Device with netFIELD Operating System (netFIELD OS) or netFIELD OS Datacenter (for virtual machines).
- OPC UA Server(s) in a reachable network (or on the netFIELD Edge Device itself) to which the app can connect in order to monitor OPC UA nodes.
- MQTT Broker in a reachable network (or on the netFIELD Edge Device itself) to which the app can publish the data of the monitored nodes.
- If your use case requires transferring MQTT messages to the netFIELD Cloud: *netFIELD App Platform Connector* (on the netFIELD Edge Device itself or in a reachable network).
- If your use case requires the transfer of the MQTT messages to any other cloud platform, you need the corresponding platform connector app (on the netFIELD Edge Device/Datacenter itself or in a reachable network), e.g.:
  - *netFIELD App Azure Connector*
  - *netFIELD App AWS Connector*
  - *netFIELD App Google Connector*

Limitations

- OPC UA write operations are not supported by the app.
- Connecting to OPC UA Servers via HTTPS is not supported by the app.

Use case example

In the use case depicted below, the netFIELD App OPC UA Client is deployed in the **IoT Edge Docker** of the **netFIELD OS** of an **OnPremise device**. It monitors multiple nodes of various OPC UA Servers located in an OT network connected to one of the Ethernet interfaces of the OnPremise device.

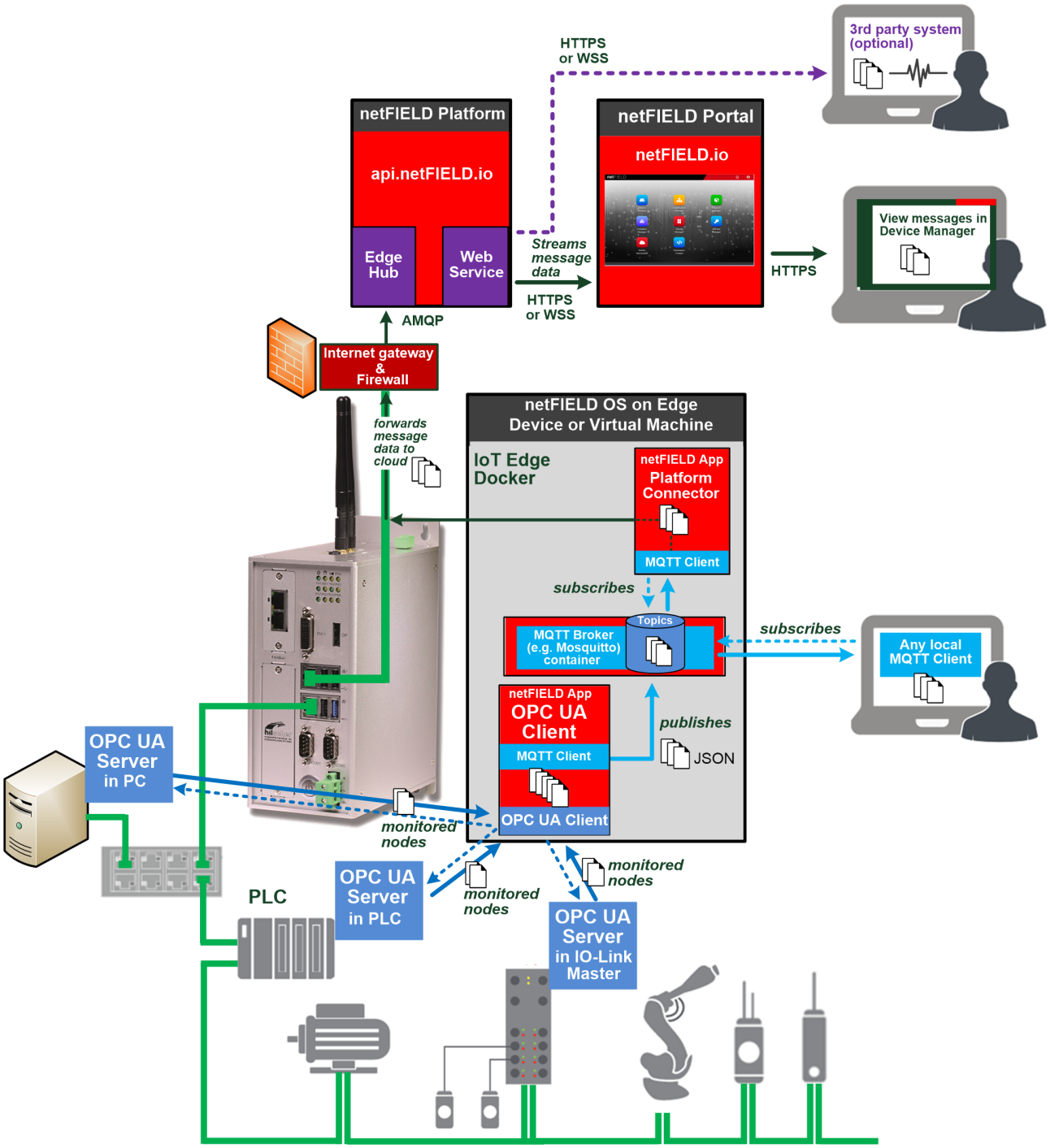

Figure 1: netFIELD App OPC UA Client data flow example

The app converts the monitored OPC UA data into MQTT topics and publishes them to the MQTT broker (i.e. mosquitto), which, in this case, also runs in the Docker on the same device. (Note that the MQTT Broker could also be running on a different device connected to the OnPremise device via Ethernet.)
The broker publishes the JSON-encoded topics to other MQTT Clients that have subscribed to them – in this case to the MQTT Client that is part of the netFIELD App Platform Connector, which sends the data to the netFIELD Platform via AMQP. (Note that the subscribing MQTT Clients can run on the same device and/or on other devices connected to the OnPremise device.)
At the netFIELD Platform in the cloud, the data is made available for third party applications via `netfield.io` API. The mere JSON messages can also be displayed in a plug-in dashboard in the Portal's **Device Manager**.

3 Quick start

This chapter provides an overview of the steps you must perform to get your *netFIELD App OPC UA Client* started.

#	Step		For details see
1	Deploy container on your netFIELD Edge Device respectively netFIELD OS Datacenter. **Note**: You might also need the *mosquitto* container on your netFIELD Edge Device (i.e. if you do not intend to use another MQTT Broker). If you intend to transfer MQTT topic messages to the netFIELD cloud, you also need the netFIELD App Platform Connector.		Section *Deploying the container* [▸ page 9]
2	Login to the **Local Device Manager** of the netFIELD Edge Device respectively netFIELD OS Datacenter.		Section *Remote Control* in the *netFIELD Portal* manual, DOC190701OIxxEN)
3	**If applicable**: If you do not intend to use the default standard MQTT settings (e.g. if your MQTT Broker is on another machine), adapt the MQTT Settings of the app according to your use case.		Section *MQTT Client Settings* [▸ page 30]
4	Specify the OPC UA Server(s) whose data you want to monitor.		Section *Servers* [▸ page 20]
5	Specify the nodes to be published as MQTT topics and specify the MQTT parameters:		Section *Selected Nodes* [▸ page 23]
	5.1	Select OPC UA Server.	
	5.2	Browse **Address Space** of selected server.	
	5.3	Select node.	
	5.4	Click **Start Monitoring.**	
	5.5	Specify MQTT parameters for topic in **Start Monitoring** dialog.	
	5.6	Click **OK** in **Start Monitoring** dialog to start publishing the monitored node as MQTT Topic to the MQTT Broker.	
6	Subscribe to the MQTT topic(s):		
	6.1	Copy the MQTT Topic name string of the previously created topic to your clipboard.	Section *Monitored Items* [▸ page 26]
	6.2	Open the MQTT Client that is intended to subscribe to the monitored node/topic (e.g. the *netFIELD App Platform Connector*)	Section *Use case example: Monitor node data in netFIELD Portal* [▸ page 36]
	6.3	Add the MQTT Topic name string to the subscription list of your subscribing MQTT Client.	
7	**If applicable**: Monitor MQTT JSON messages in the cloud (i.e. if you have subscribed to the MQTT topics with your *netFIELD App Platform Connector*):		Section *Use case example: Monitor node data in netFIELD Portal* [▸ page 36]
	7.1	In the Device Manager of the netFIELD Portal, open DEVICE NAVIGATION and go to CONTAINERS > netFIELD App Platform Connector	
	7.2	Open **Topics** tab and click on a topic to which you have subscribed.	

Table 2: Quick start overview netFIELD App OPC UA Client

4 Deploying the container

4.1 Step-by-step instructions

This section describes how to deploy (“install”) the *netFIELD App OPC UA Client* on your Edge Device (like e.g. netFIELD OnPremise).

Note:
This manual deployment is necessary only if the container is not deployed via *Deployment Manifest* or via *Fleet Management* (see the corresponding chapters in the operating instruction manual *netFIELD Portal*, DOC190701OIxxEN).
Check the **Installed Containers** tab (Device Manager > [your device] > DEVICE NAVIGATION > Containers > Installed Containers) to see if the container has already been deployed and is running on the device.

1. Select netFIELD App container in Portal.
   - ➢ Login to the netFIELD Portal and select your device in the Portal’s **Device Manager**.
   - ➢ Open the **Available Containers** tab in the DEVICE NAVIGATION (DEVICE NAVIGATION > Containers > Available Containers).

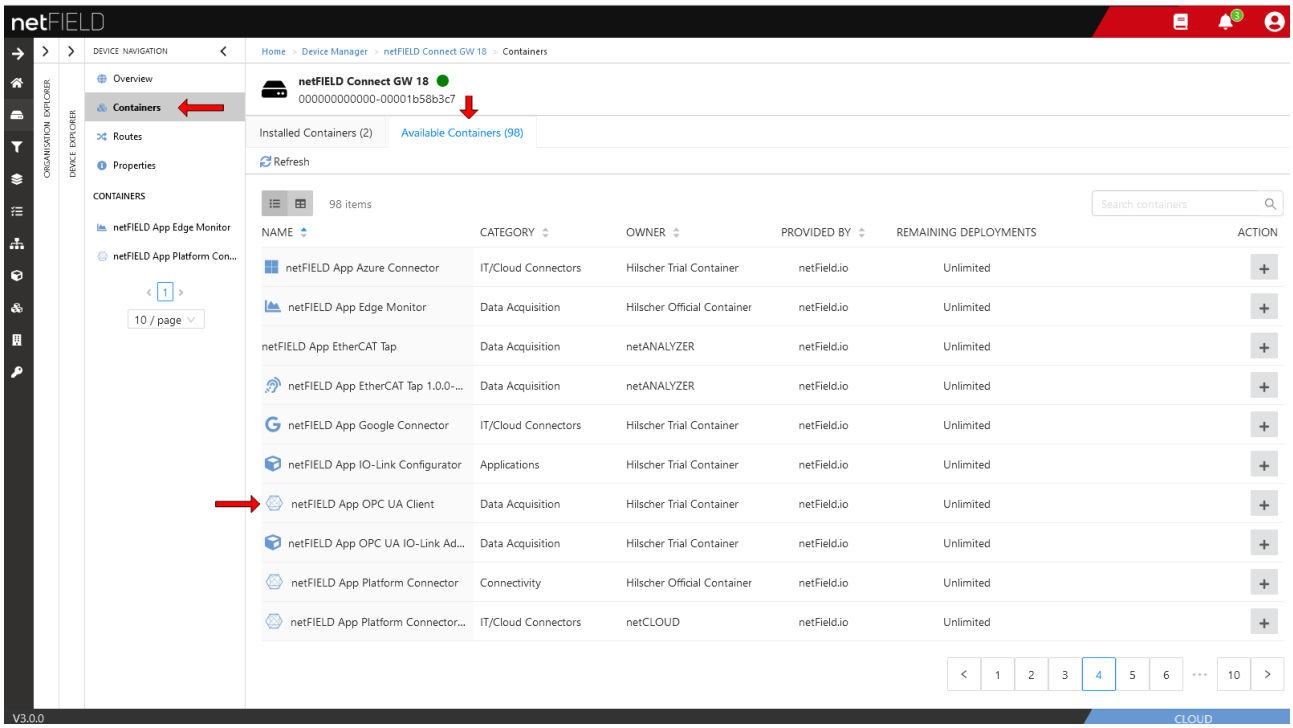

Figure 2: Available Containers tab

   - ➢ Scroll through the list and look for the **netFIELD App OPC UA Client** container.
   - ➢ Click on the **netFIELD App OPC UA Client** entry or on the **+** button.

The deployment dialog screen opens:

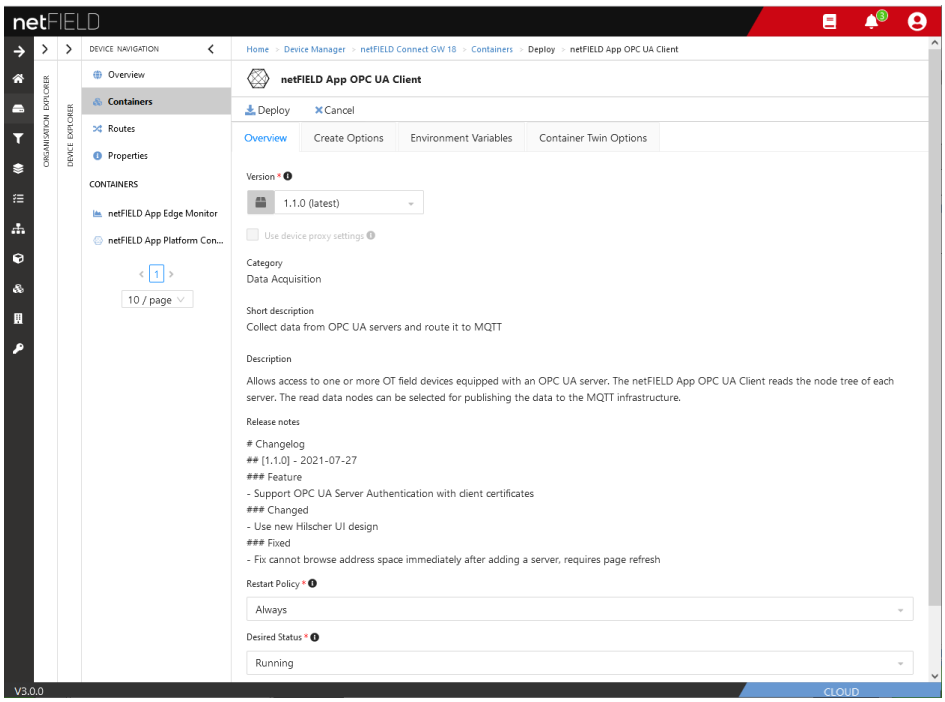

Figure 3: Deploy container

- In the **Overview** tab of the netFIELD App OPC UA Client deployment dialog, keep the preset default settings. (This ensures that the latest version of the container will be deployed and that it will be automatically started on the device after deployment.)

Important:

We strongly recommend you to keep also the default settings in the **Create Options** and **Container Twin Options** tabs.
If necessary, these configuration settings can be changed later (i.e. after having deployed the container). The settings are described in section *Changing default Container settings in netFIELD Portal (experts only)* [page 14]. Note that only expert users should change the Create Options.

2. Enter personal encryption key (optional).
   - Open the **Environment Variable** tab.

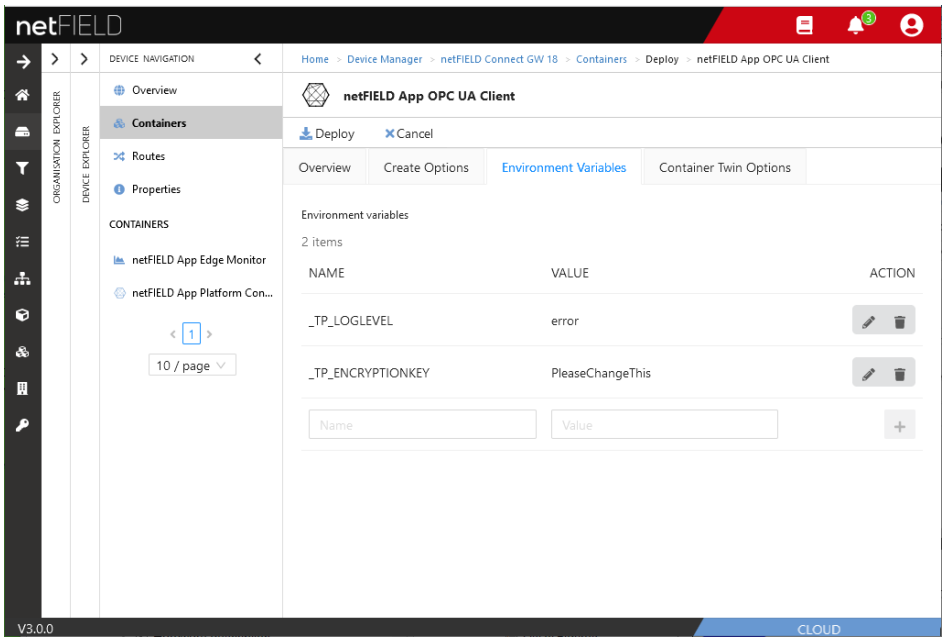

Figure 4: Encryption key in Environment Variables

Note:
The encryption key is used by the OPC UA Client app to generate a security hash tag for server access credentials when you add a new OPC UA Server (that requires a certificate or user name and password) to your local configuration.
This ensures that the credentials in the configuration of your OPC UA Client instance become “portable”; i.e. that they can be used by other instances of the OPC UA Client app (e.g. running on other netFIELD Edge Devices or Datacenters) when this configuration is exported and imported accordingly. You can export/import (= download/upload) your configuration in the *Configuration Manager* [page 33].
Note that every other instance of the OPC UA Client that shall use an imported configuration must have the same encryption key in its environment variables.
Note also that you can use the given `PleaseChangeThis` encryption key as default key if you do not want to define your own “personal” key.

- To customize the encryption key, click the **Edit item** button next to the `_TP_ENCRYPTIONKEY` variable.

➢ Enter your new key into the **VALUE** field. Note that the key must consist of exactly 16 characters and that it is case sensitive.

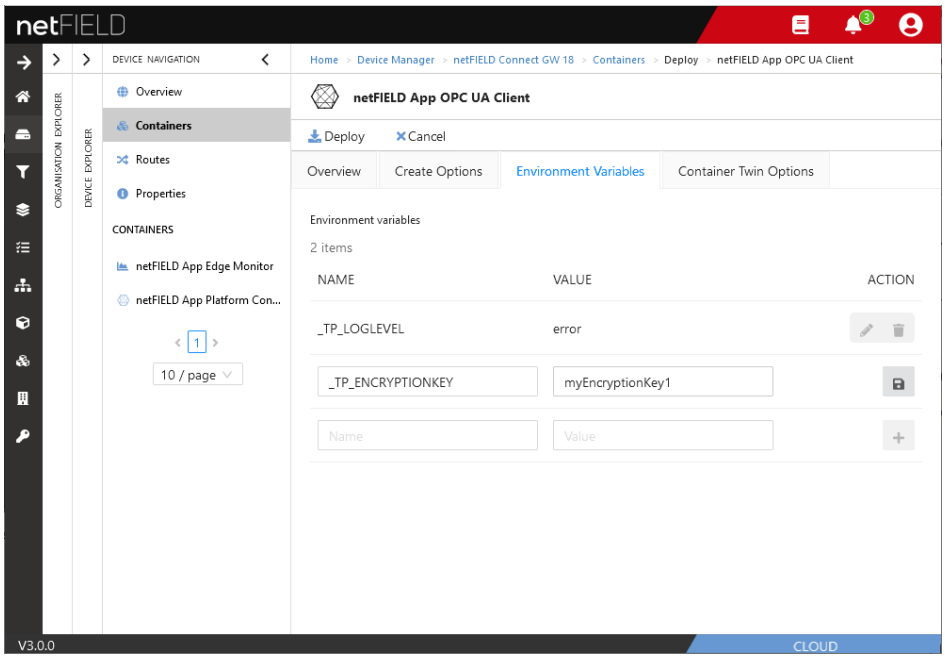

Figure 5: Change encryption key in Environment Variables

➢ Click **Save item** button.

3. Deploy the netFIELD App container image.

➢ Click **Deploy** button.

➪ The container image is downloaded from the cloud to the device, and automatically started on the device. This may take a few minutes. After its deployment, the container will from now on be listed in the **Installed Containers** tab of your device. Note that it might take a few minutes until the container STATUS changes to “Connected” (green dot).

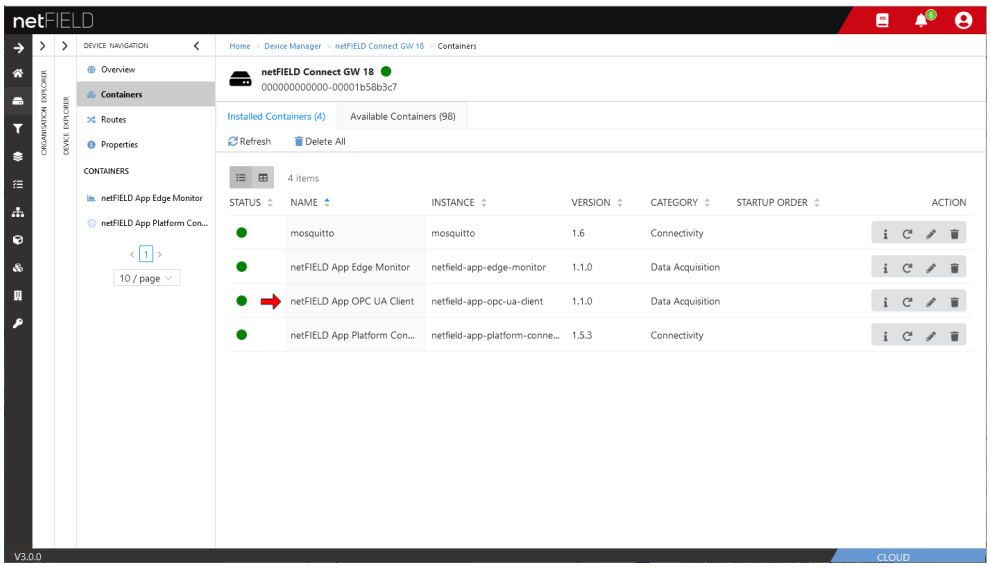

Figure 6: OPC UA Client in Installed Containers tab

Note:

Note that the OPC-UA Client requires an MQTT Broker. We recommend you to deploy the *Mosquitto* MQTT Broker on your Edge Device. *Mosqitto* is also available for deployment in the **Available Containers** tab.
If you want to transfer the topics to the netFIELD Platform, you must also deploy the *netFIELD App Platform Connector*, which is also available for deployment in the **Available Containers** tab.

4.2 Changing default Container settings in netFIELD Portal (experts only)

4.2.1 Overview

If you do not want to use the default container configuration settings, you can change them before deployment, or even retroactively after deployment (i.e. if the container has already been installed on the device) by using the **Update** button. After the new settings have been saved, they are automatically transferred to the container on the device.

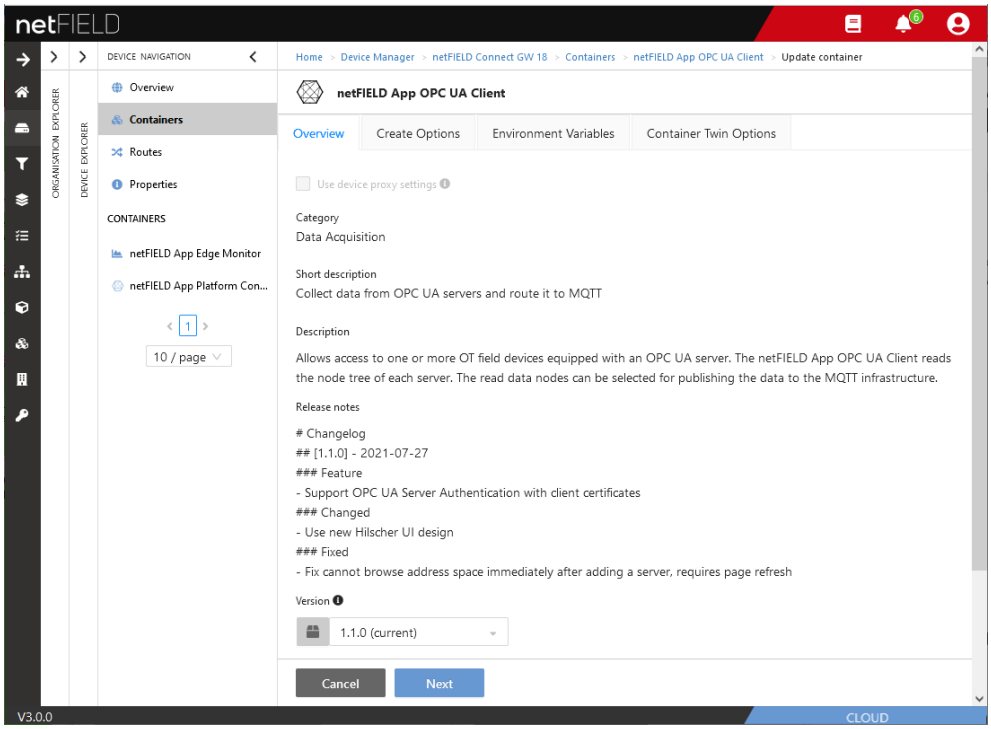

Figure 7: Overview tab

Elements	Description
Use device proxy settings	If your device uses an HTTP or HTTPS proxy server in its local IT network you can choose here that the container shall use the same proxy server settings that are stored for the device in the portal. **Note**: This option is not applicable for the OPC UA Client app because it does not establish any HTTP, HTTPS or FTP connections.
Category	Informative texts about the container. Cannot be changed here.
Short description	
Description	
Release notes	
Version	If other versions of the container are available, you can select a different version (e.g. an older version) by clicking in the field. The container on the device will be updated accordingly.

Elements	Description	
Restart Policy	In the drop-down list, you can determine under which conditions the container on the device is to be restarted by the system.	
	Always	The container is always restarted, no matter why it was deactivated or crashed.
	Never	The container is never restarted.
	On-Failed	The container is restarted when it crashes, but not if it has been "properly" deactivated.
	On-Unhealthy	The container is restarted when it crashes or is diagnosed as "unhealthy" by the system.
Desired Status	In the drop-down list, you can determine the operating state in which the container shall be by default. Note that you can also stop a currently running container by selecting the **Stopped** option, respectively start a currently stopped container by selecting the **Running** option.	
	Stopped	The system does not start the container until it receives the command to do so (locally on the device using CLI or by updating the configuration in the portal).
	Running	The system starts the container immediately (i.e. as soon as the image has been completely downloaded to the device).
Startup Order	Optional parameter: Here you can determine the order (in relation to other containers) in which the container should be started on the device when first deployed. The order is declared with integers, where a container given a startup value of 0 is started first and then higher numbers follow.	
Cancel	Closes the tab without saving changes.	
Next	Opens the next tab. **Note**: You can save the changes with the **Save** button in the last tab (Container Twin Options) The new settings will then be transferred to the container on the device.	

Table 3: Elements in Overview tab

4.2.2 Create Options

The Create Options in stringified JSON format contain the initial configuration parameters of the container.

Important:
Do not change these parameters unless you are an expert user. Wrong settings may lead to malfunctions of the application container.

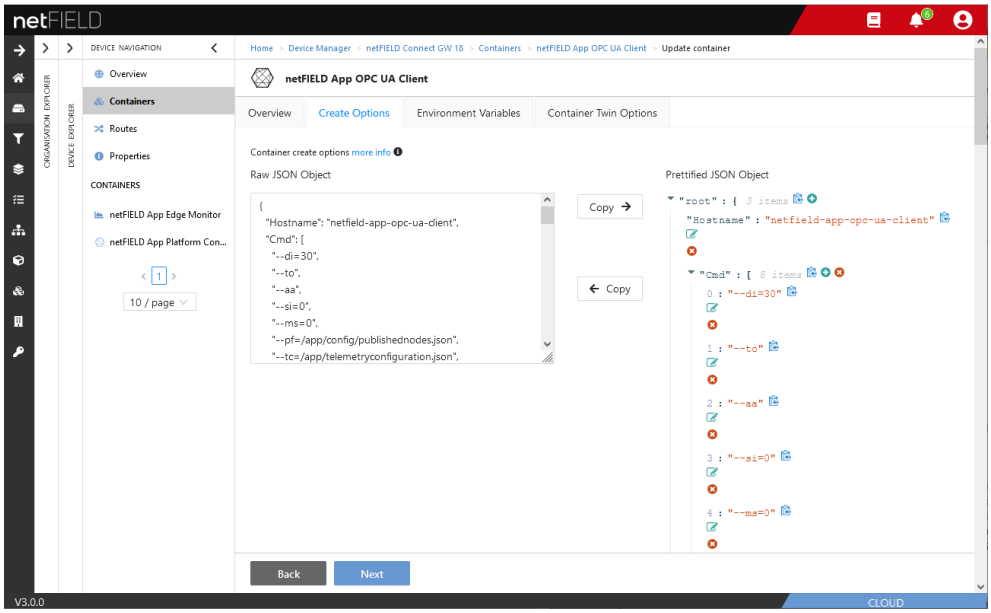

Figure 8: Container Create Options

You can edit the JSON object of the “Create Options” directly in the **Raw JSON Object** field on the left side, or you can edit it as **Prettified JSON Object** in the field on the right side of the screen. The content of the **RAW JSON Object** field will be transferred to the container on the device after saving; however, the **Prettified JSON Object** field allows you more convenient editing of the object items. If you are using the **Prettified JSON Object** field for editing, you can take over the edited object items into the **Raw JSON Object** field by clicking the ← **copy** button.

Element	Description
Copy →	Copies the content of the **Raw JSON Object** field into the **Prettified JSON Object** field (for more convenient editing).
Copy ←	Copies the content of the **Prettified JSON Object** field into the **Raw JSON Object** field (for transferring it to the container on the device after saving).
	Copies the object item to your clipboard.
	Adds a new object item.
	Deletes the object item.
	Opens a box that allows you to edit a value/parameter of an object item. Click below the box to take over the new value/parameter. Click below the box to cancel.
Back	Opens the previous tab.

Element	Description
Next	Opens the next tab. **Note**: You can save the changes with the **Save** button in the last tab (Container Twin Options). The new settings will then be transferred to the container on the device.

Table 4: Operating elements for Container Create Options update

4.2.3 Environment Variables

With the environment variables, you can customize the log level and the encryption key of the app.

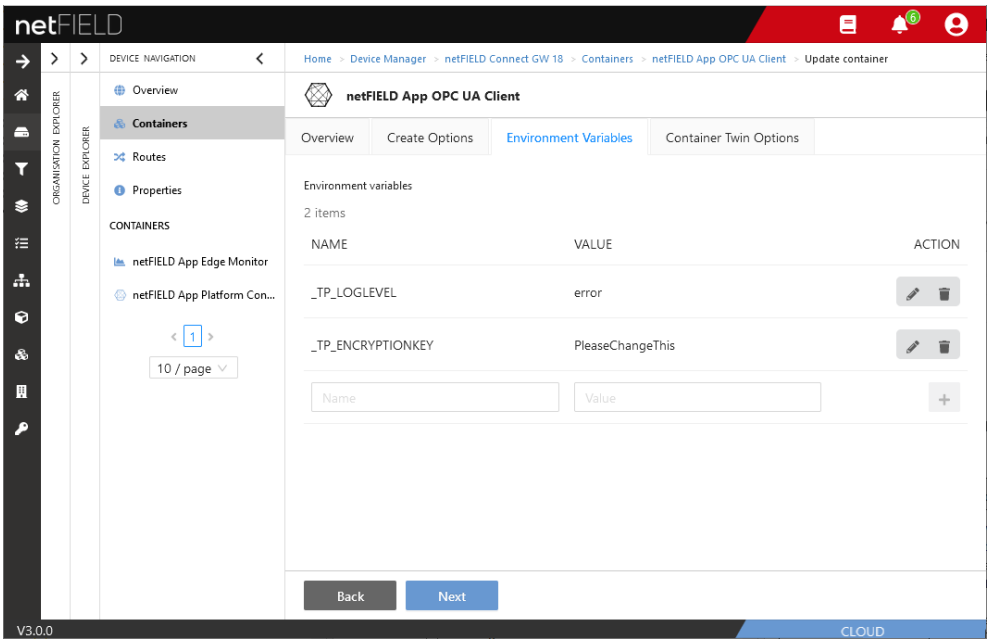

Figure 9: Environment Variables

Variable	Description
_TP_LOGLEVEL	Defines what kind of messages issued by container will be logged by the netFIELD OS. The preset `error` level means that all messages belonging to the “severity level” **Error** will be logged. **Important**: Do not change this parameter unless you are an expert user. Changing to a lower level can lead to the issuing of too many messages, which can cause log overflow.
_TP_ENCRYPTIONKEY	The encryption key is used by the OPC UA Client app to generate a security hash tag for server access credentials when you add a new OPC UA Server (that requires a certificate or user name and password) to your local configuration. This ensures that the credentials in the configuration of your OPC UA Client instance become “portable”; i.e. that they can be used by other instances of the OPC UA Client app (e.g. running on other netFIELD Edge Devices or Datacenters) when the configuration is exported and imported accordingly. Note that every other instance of the OPC UA Client that shall use an imported configuration must have the same encryption key in its environment variables. Note also that you can use the given `PleaseChangeThis` encryption key as default key if you do not want to define your own “personal” key.

Table 5: Environment Variables

Element	Description
	Select this button to change name or value of the variable. Use the button to save your changes for the time being.
	Deletes the variable.
	Adds a new variable. First fill-in the **NAME** and **VALUE** fields, then click button to add the variable.
Back	Opens the previous tab.
Next	Opens the next tab. **Note**: You can save the changes with the **Save** button in the last tab (Container Twin Options) The new settings will then be transferred to the container on the device.

Table 6: Operating elements for Environment Variables

4.2.4 Container Twin Options

Container Twin Options are not applicable for this container.
However, if you have made changes to the container configuration settings, you can save the changes here in this tab. The changed settings are then automatically transferred to the container on the device.

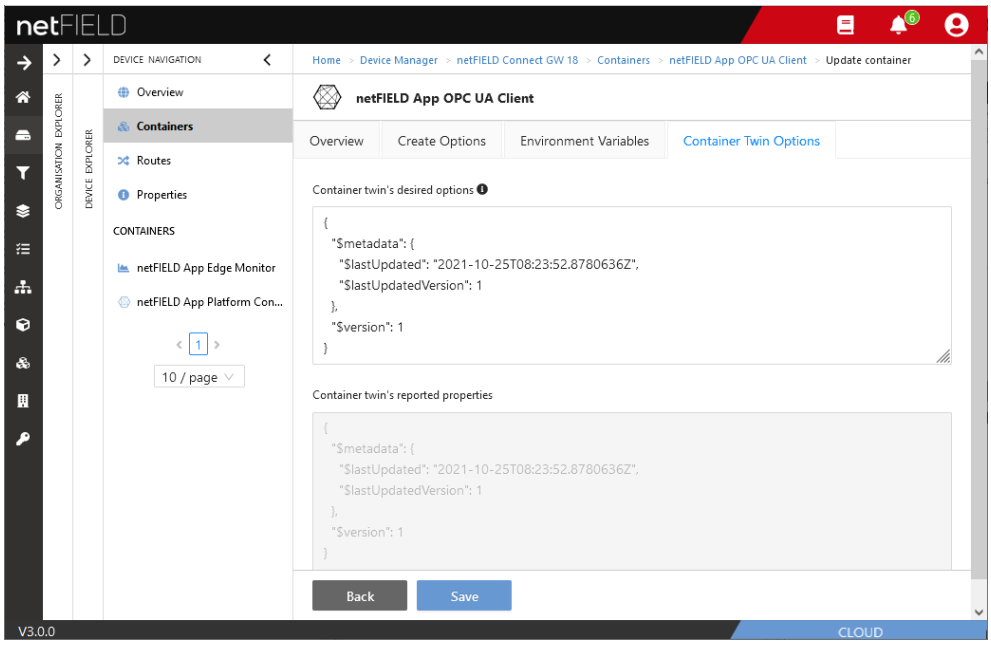

Figure 10: Container Twin Properties

Element	Description
Back	Opens the previous tab.
Save	Saves your changes. The changed parameters are then automatically transferred to the container on the device. **Note**: The application of the changed parameters requires a restart of the container on the device.

Table 7: Operating elements for Container Twin Options

5 Configuring the OPC UA Client in Local Device Manager

5.1 Overview

The OPC UA Client app container provides a configuration GUI in the Local Device Manager of the netFIELD OS. This configuration GUI is automatically plugged-in when the container is deployed. After having established a connection to the Local Device Manager (e.g. by Remote Control from the netFIELD Portal, see section *Remote Control* in the *netFIELD Portal* manual, DOC190701OIxxEN), the configuration GUI can be selected in the navigation panel (1) of the Local Device Manager.

Note:

Note that it might take a few minutes after deployment before the **netFIELD App OPC UA Client** entry becomes visible in the navigation panel. You may also have to reload the web page in your browser by pressing **F5** on your keyboard.

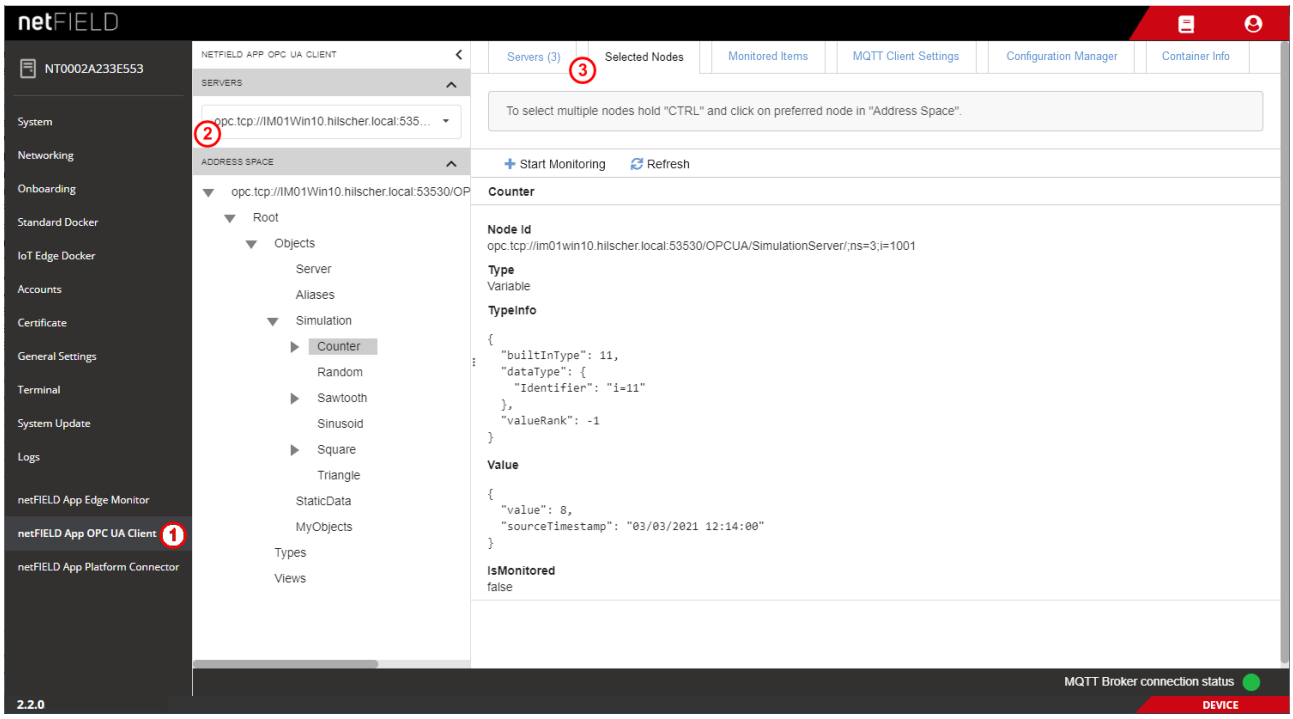

Figure 11: Configuration GUI plugin in local Device Manager

In the “Browse area” (2) on the left, you can select an OPC UA Server in the **SERVERS** drop-down list for browsing its data nodes, which are then displayed in a tree hierarchy in the **ADDRESS SPACE**. Use the little arrow head icons to expand or collapse items.

The tabs in the header of the screen (3) allow you to navigate through the configuration and management options of the OPC UA Client app.

5.2 Servers

In the **Servers** tab, you can add and manage the OPC UA servers to which you want to connect with your OPC UA Client.

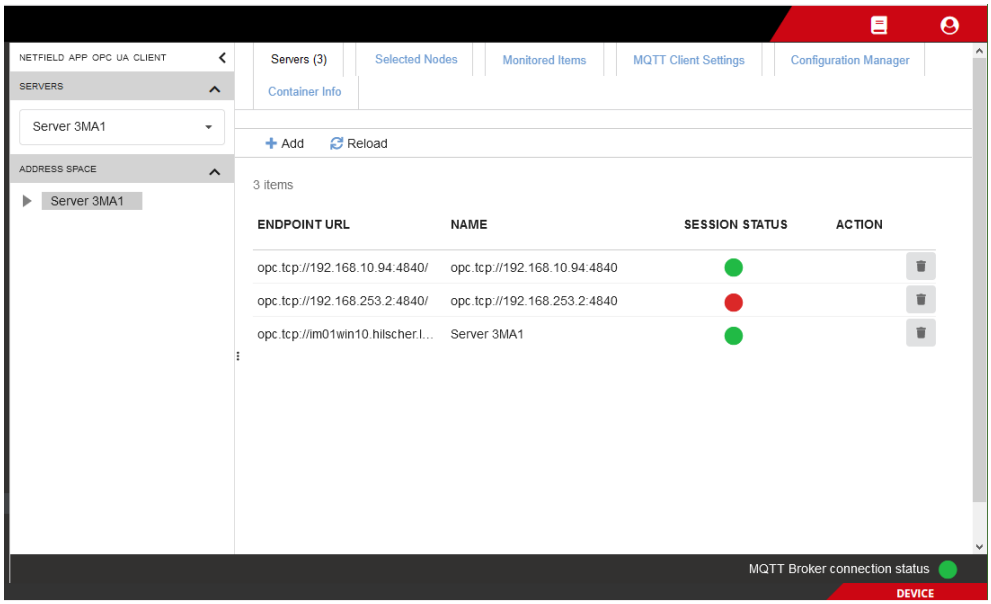

Figure 12: Servers tab

Element	Description	
Add	Click here to add an OPC UA Server to which you want to connect with your Client. **Note**: The Client automatically tries connects to the server after having added the server to the list.	
Reload	Refreshes the display of the Session Status.	
ENDPOINT URL	URL of the OPC UA Server.	
NAME	Display name of the server (defined by user).	
SESSION STATE	Shows the state of the server-client connection.	
	Connecting	
	Disconnected	
	Connected	
ACTION		Closes the connection to the OPC UA server and deletes it from the list.

Table 8: Elements in Servers tab

If a SESSION STATE is green (i.e. “connected”), you can click on the server entry to open the **Selected Nodes** tab where you can select the nodes of the server for monitoring.

Adding an OPC UA Server (i.e. connecting to a server)

➢ Click + **Add** button.

➪ The **Add Server** dialog opens:

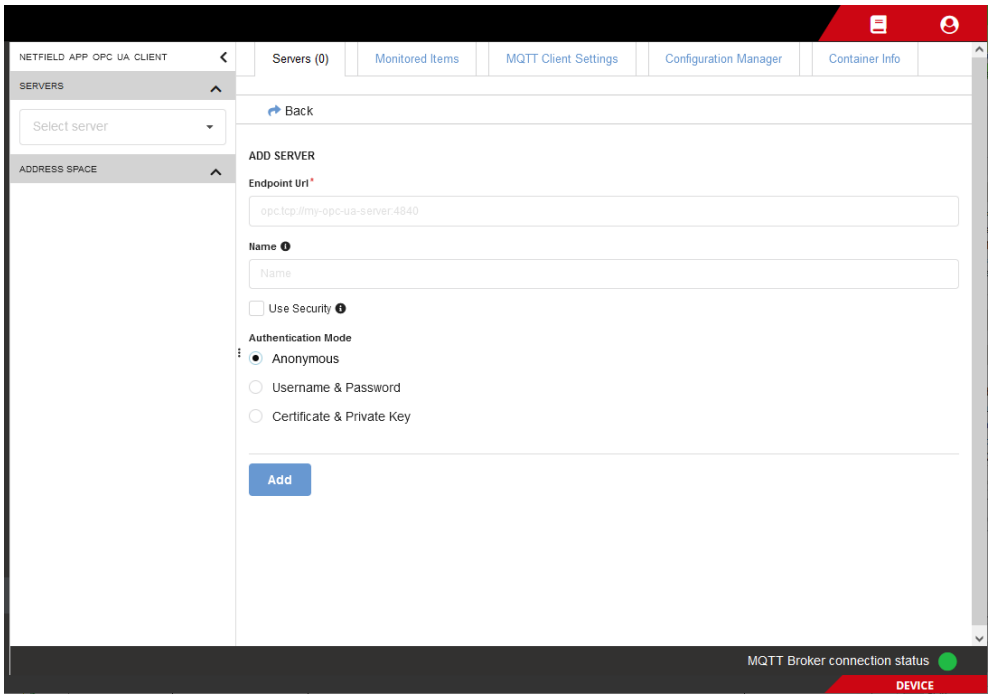

Figure 13: Add server dialog

Element	Description
Back	Click here to exit the dialog without saving.
Endpoint URL	Enter here the endpoint URL of the OPC UA Server. Use the following syntax: `opc.tcp://<server name or IP address>:<port number>`
Name	Enter here a display name for the server (optional). If left empty, the Endpoint URL will be used as display name.
Use Security	Select this option if you want the client to automatically connect to the most secure endpoint that is available on the server. **Note**: By default, the client uses anonymous user authentication. However, the client also supports user authentication by username and password or certificate.

Element	Description	
Authentication Mode	Select here the authentication method for the server. **Note**: When saved, the credentials that you enter here will be encrypted using the encryption key defined in the environment variables of the container (see section *Environment Variables* [▸ page 17]). This ensures that the credentials in the configuration of your OPC UA Client instance become “portable”; i.e. that they can be used by other instances of the OPC UA Client app (e.g. running on other netFIELD Edge Devices or Datacenters) when the configuration is exported and imported accordingly.	
	Anonymous	Select this option if the server does not require any credentials.
	Username & Password	Select this option if the server requires username and password. This opens the **Username** and **Password** fields, in which you can enter the corresponding credentials.
	Certificate & Private Key	Select this option if the server uses a certificate in PEM format. This opens the **Certificate** and **PrivateKey** fields, into which you can paste the credentials (i.e. certificate and private key strings). To do so, open the corresponding certificate file in a text editor, copy the contained text string to your clipboard and paste it into the **Certificate** field. Take care to copy and paste the entire string (including the BEGIN CERTIFICATE and END CERTIFICATE tags). Do the same with the private key file.
Add	Click here to add the OPC UA server to your servers list. The client automatically tries to connect to the server after having added it to the list.	

Table 9: Elements in Add Server dialog

5.3 Selected Nodes

The **Selected Nodes** tab allows you to browse the nodes in the **ADDRESS SPACE** of a connected OPC UA server to select nodes for monitoring. The tab opens automatically when you click on a server in the **Servers** tab.

Note:
Note that the SESSION STATUS of the server must be in state “connected” (which is indicated by a green dot) in order to browse its ADDRESS SPACE.

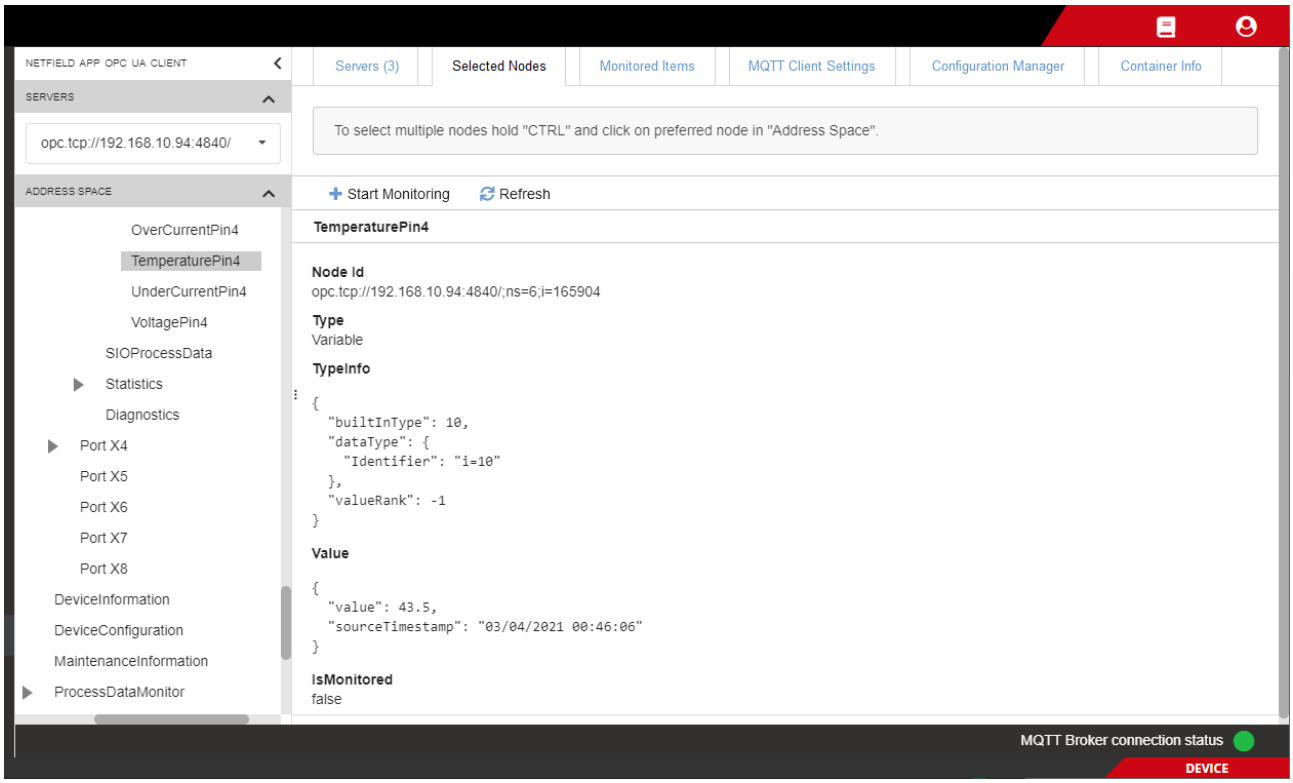

Figure 14: Selected Nodes

The main area displays the parameters and the value of the selected node in JSON format.

Element	Description
Start Monitoring	Click this button if you want to monitor the selected node. (This button is only visible if you have selected a node that can be monitored.)
Stop Monitoring	Click this button if you want to stop monitoring the selected node. (This button is only visible if you have selected a node that is being monitored.) **Note**: As an alternative, you can use the **Stop Monitoring** button for this node in the **Monitored Items** tab.
Refresh	Refreshes the Selected Nodes list.
Start Monitoring Selected Nodes	Click this button if you want to monitor all currently selected nodes. (This button is only visible if you have selected multiple nodes by using the CTRL key on your keyboard.)
Remove from Selected	Click here to unselect a node. (This button is only visible if you have selected multiple nodes by using CTRL on your keyboard.)

Table 10: Elements in Selected Nodes tab

Start Monitoring

➢ Select a node and click **Start Monitoring** button.

The **Start Monitoring** dialog opens:

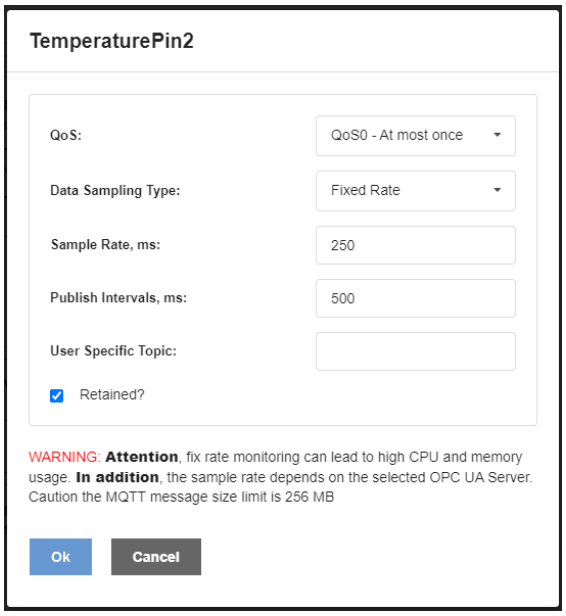

Figure 15: Start monitoring item dialog

Element	Description	
QoS	In the drop-down list, select the MQTT Quality of Service. **Note**: This parameter defines the QoS by which the OPC UA Client app publishes the selected node to the MQTT Broker. For performance reasons, we recommend you to use `QoS0` whenever possible. Using `QoS1` or `QoS2` will increase performance requirements of the app.	
Data Sampling Type	In the drop-down list, select the data sampling type:	
	On Change	Publish data only if the node's value has changed.
	Fixed Rate	Publish data at a fixed rate. If you select this option, the **Publish Intervals** field appears.
Sample Rate, ms	Specify here the sample rate (in milliseconds) at which the OPC UA Client shall poll the OPC UA Server for fresh values. Note that this value depends mainly on the capabilities of the server.	
Publish Intervals, ms	Specify here the interval (in milliseconds) at which the OPC UA Client app publishes the sampled node value(s) to the MQTT broker. **Note**: If more than one node value sample has been acquired in this time-span, the message will contain an array of node values. If you e.g. define a sample rate of `250` ms and a publish interval of `1000` ms, each message will contain four node values. Keep in mind that messages should not grow endlessly in size. Therefore, configure this parameter according to your use case and in a way that provides a good trade-off between the publish interval and the number of samples coming from the node in that period.	
User Specific Topic	Enter here the MQTT Topic name string under which the messages for this node will published to the broker. If left empty, the app uses the default string (see section *MQTT message format* [▸ page 40]). Note that you are free to define the topic hierarchy and name according to your individual needs. You are only restricted not to use the `#` and `+` characters and not to use `$` as the very first character.	
Retained?	"Retained" flag of MQTT message.	
OK	Click here to start monitoring this node. This node/topic will be added to the list in the **Monitored Items** tab.	
Cancel	Click here to leave the dialog without saving the data.	

Table 11: Elements in Start monitoring item dialog

Selecting multiple nodes

You can select multiple nodes at once by pressing the **CTRL** key on your keyboard while clicking on a node. For each node, a tab containing its parameters and values in JSON format is opened in the main window:

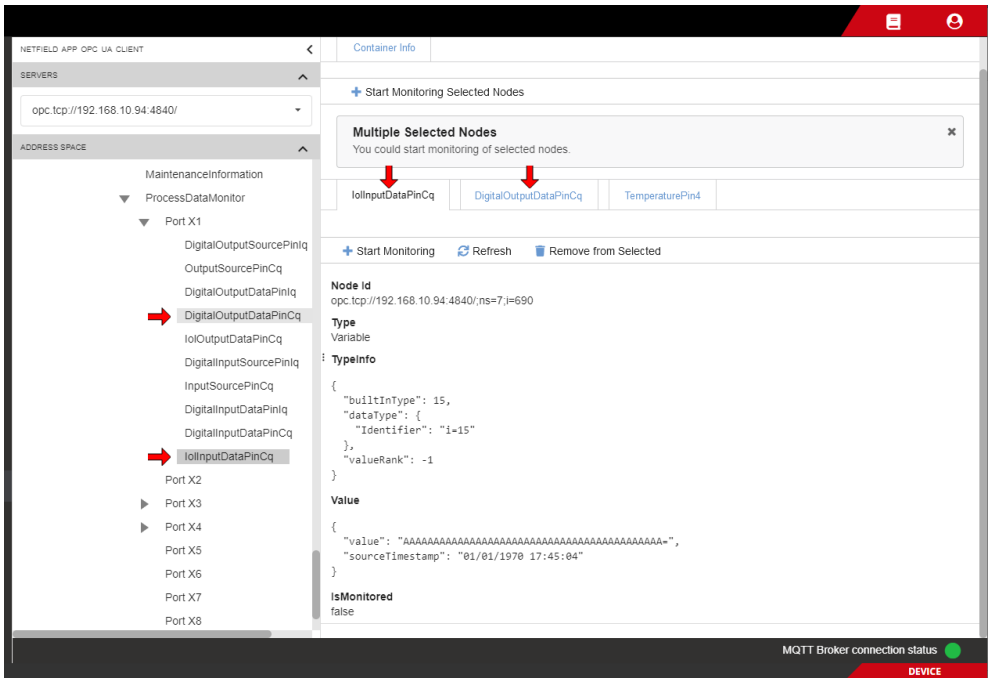

Figure 16: Selection of multiple nodes

5.4 Monitored Items

The **Monitored Items** tab allows you to manage your published nodes. You can click on an item/node in the list to display its parameters and current value in JSON format. You can also delete items here (and thus stop them from being monitored and published) or add new monitored items here (if you know its OPC UA node ID).

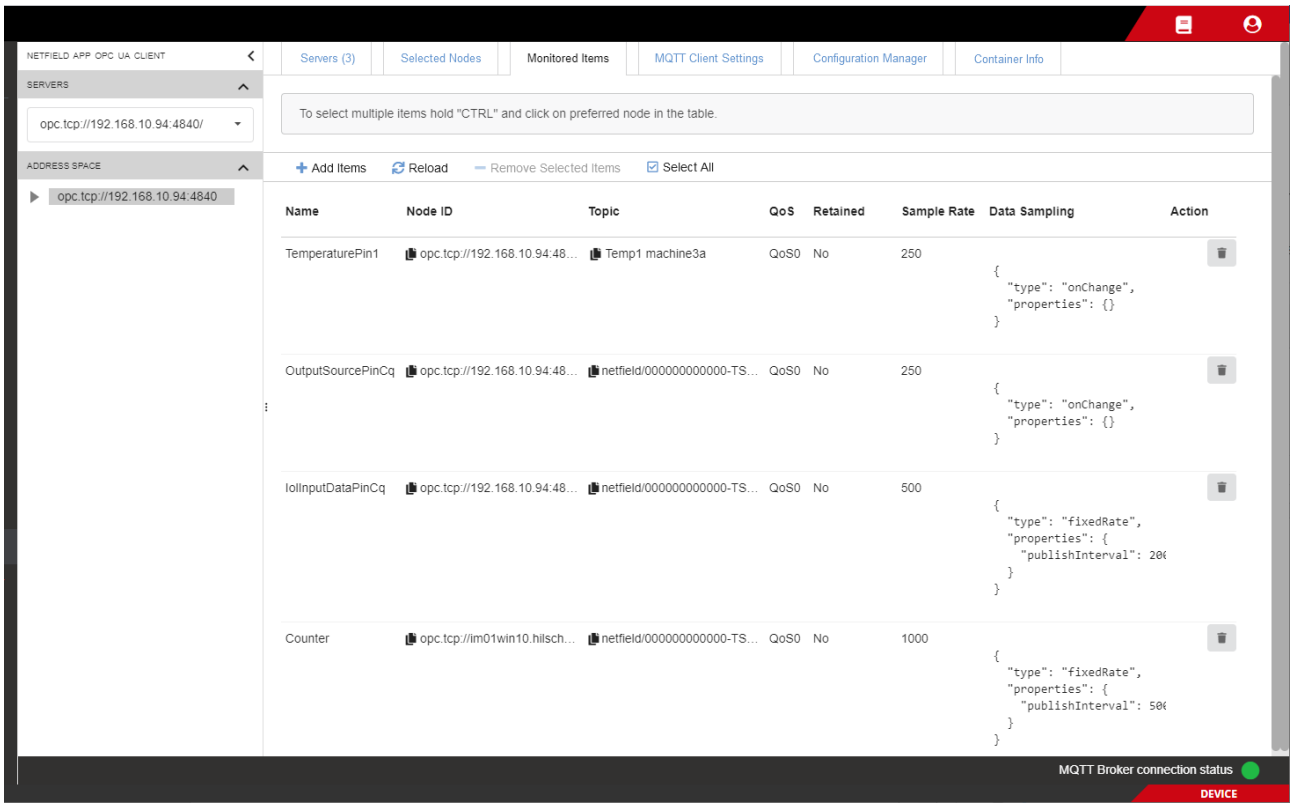

Figure 17: Monitored items

Note:

Note that you cannot change or edit the parameters of the monitored items here. The MQTT publication parameters can only be set when you select a node for monitoring in the **Selected Nodes** tab. If you want to change the MQTT settings of a node retroactively, you first have to delete the node here and then add the node again with the new MQTT parameters in the **Selected Nodes** tab (see section *Selected Nodes* [▸ page 23]).

Element	Description
Add Items	Click here to add a new item to be monitored. A dialog opens in which you can enter the server and the node ID of the item that you want to monitor. Note that it is more convenient to select your item by browsing the ADDRESS SPACE of a selected server in the object tree on the left (see section *Selected Nodes* [▸ page 23]).
Reload	Refreshes the monitored items list.
Remove Selected Items	Click here to delete a selected monitored item. The app stops monitoring and publishing the node.
Select All	Selects all items in the list at once.
Unselect All	Unselects all items in the list at once.

Element	Description	
Name	Name of the OPC UA node as defined by the server.	
Node ID	ID of the monitored node (consisting of the server URI and the OPC UA node ID). You can copy the Node ID to your clipboard by clicking the button. To see the full and unabridged Node ID, hover with your mouse over the entry.	
Topic	MQTT Topic name string under which the messages for this monitored node are published to the broker. If you do not want to use the default string (see section *MQTT message format* [page 40]), you can define your own topic name string in the **User Specific Topic** field when you select a node for monitoring (see section *Selected Nodes* [page 23]). You can copy the string to your clipboard by clicking the button (in order to subscribe to this topic from another MQTT client, e.g. from the *netFIELD App Platform Connector*). To see the full and unabridged string, hover with your mouse over the entry.	
QoS	MQTT “Quality of Service” by which the OPC UA Client app publishes the node to the MQTT broker (as defined by the user).	
Retained	“Retained” flag of MQTT message (as defined by the user).	
Sample Rate	Shows the sample rate in milliseconds at which the OPC UA Client polls the OPC UA Server for fresh values (as defined by the user).	
Data Sampling	Shows the Data Sampling Type (as defined by the user).	
	On Change	The data is only published if the node’s value has changed.
	Fixed Rate	The data is published at a fixed rate. In this case, the Publish Interval in milliseconds is also displayed.
Action		Deletes the item. The app stops monitoring and publishing the node.

Table 12: Elements in Servers tab

➢ Click on an item/node in the list to open a sheet displaying its parameters and current value:

Figure 18: Data of monitored item

- To update the value, click **Refresh** button. (This function triggers a read request from the server and displays the current value regardless of the configured sample rate.)
- To close the sheet, click anywhere outside the sheet.

Selecting items

- To select one or multiple items in the list (e.g. to remove respectively delete it/them), press **CTRL** on your keyboard while clicking on the item(s).

Adding an item

The **Add Items** dialog allows you to add one or several nodes for monitoring. Unlike the **Selected Nodes** tab, the **Add Items** dialog does not allow you to browse the address space of an OPC UA Server, you have to know and manually enter the node ID here.

- Click **Add Items** button.
- The **Add Items** dialog opens.
- Enter the IDs of the nodes that you want to monitor and set the MQTT publishing parameters.

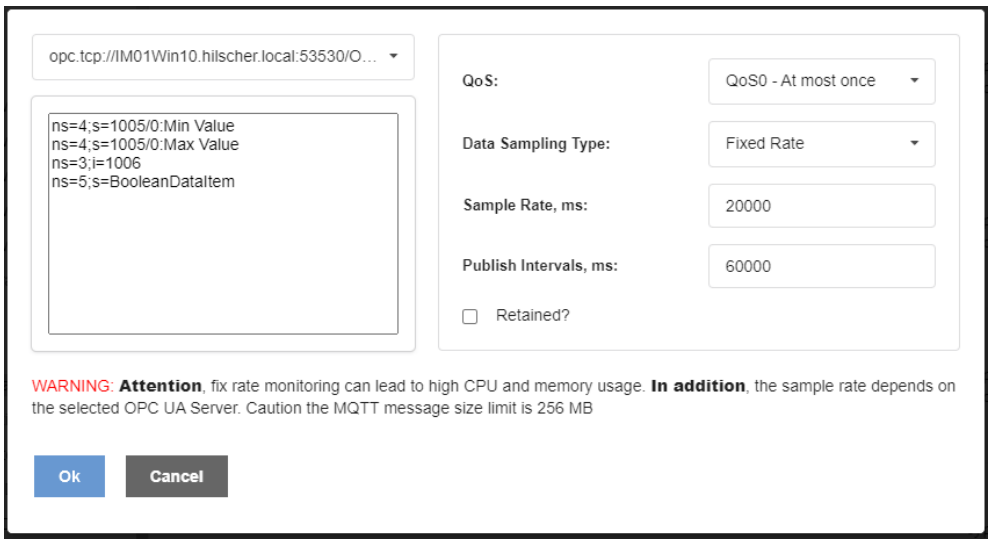

Figure 19: Add item dialog

Element	Description
Server list	In the drop-down list in the upper left corner, select the OPC UA Server whose nodes you want to monitor.
Node IDs field	Use the empty field below the Servers list to enter the IDs of the nodes that you want to monitor. If you want to add multiple nodes here, distinguish the nodes by using separate lines for each entry (press **Return** on your keyboard). **Note**: Each node will become a separate entry in the **Monitored Items** list. Note also that the MQTT parameters that you set in the right area of the dialog will be used for all nodes that you enter here.
QoS	In the drop-down list, select the MQTT Quality of Service. **Note**: This parameter defines the QoS by which the OPC UA Client app publishes the selected node to the MQTT broker. For performance reasons, we recommend you to use `QoS0` whenever possible. Using `QoS1` or `QoS2` will increase performance requirements of the application container

Element	Description	
Data Sampling Type	In the drop-down list, select the data sampling type:	
	On Change	Publish data only if the node's value has changed.
	Fixed Rate	Publish data at a fixed rate. If you select this option, the **Publish Intervals** field appears.
Sample Rate, ms	Specify here the sample rate (in milliseconds) at which the OPC UA Client polls the OPC UA Server for fresh values. Note that this value depends mainly on the capabilities of the server.	
Publish Intervals, ms	Specify here the interval (in milliseconds) at which the OPC UA Client app publishes the selected node(s) to the MQTT broker. **Note**: If more than one node value sample has been acquired in this time-span, the message will contain an array of node values. If you e.g. define a sample rate of `250` ms and a publish interval of `1000` ms, each message will contain four node values. Keep in mind that messages should not grow endlessly in size. Therefore, configure this parameter according to your use case and in a way that provides a good trade-off between the publish interval and the number of samples coming from the node in that time.	
Retained?	"Retained" flag of MQTT message.	
OK	Click here to add the node(s) for monitoring.	
Cancel	Click here to leave the dialog without saving the data.	

Table 13: Elements in Add Item dialog

After clicking the **OK** button, the app checks the nodes for availability on the OPC UA server. Invalid node IDs will be marked by a red cross, valid ones by a green check mark. The valid ones are added to the **Monitored Items** list.

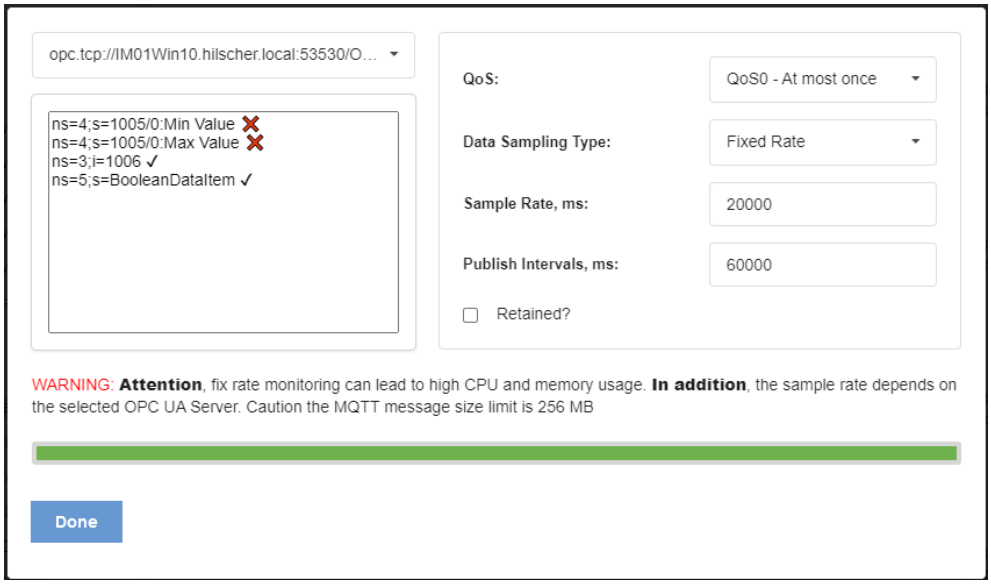

Figure 20: Add items done

➢ Click **Done** to close the dialog and go back to the **Monitored Items** list.

5.5 MQTT Client Settings

In the **MQTT Client Settings** tab, you can customize the MQTT client settings of the OPC UA Client app. By default, the app uses the standard MQTT client settings of the netFIELD OS, which can be viewed (and changed) in the Local Device Manager under **General Settings** > **Default MQTT Client Settings**.

If you want to use different settings for your OPC UA Client app – e.g. if you want to use a different broker than the preset `tcp://mosquitto:1883` – you must uncheck the **Use general settings** option and enter your new parameters in the configuration fields that are now displayed:

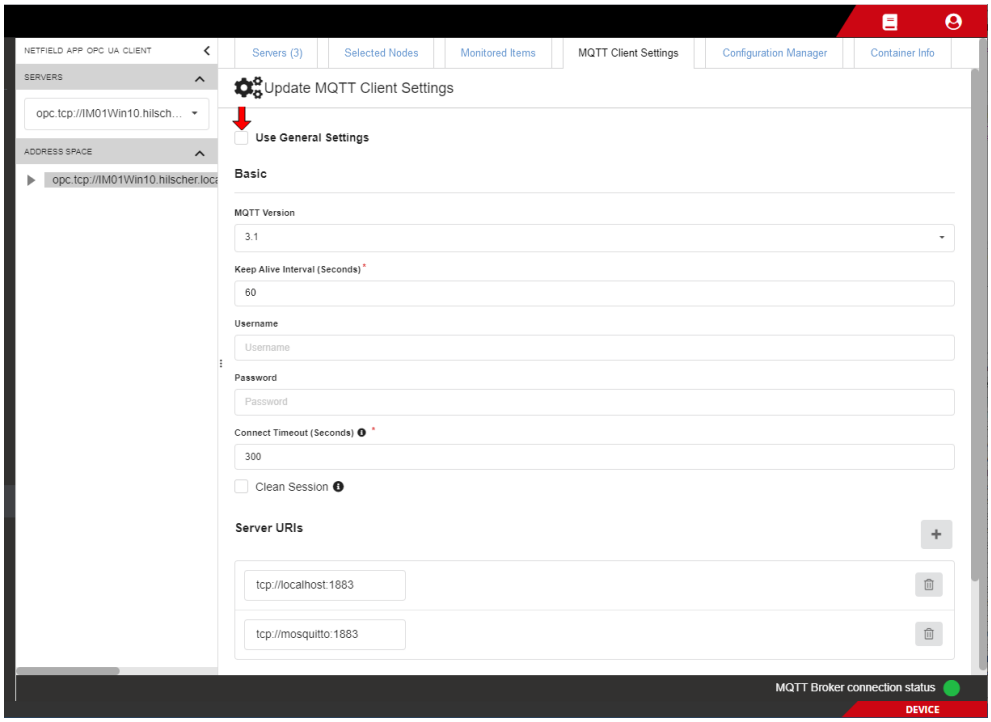

Figure 21: MQTT Client Settings

Note:
Changes to the MQTT Client Settings that you make here for your OPC UA Client app will not affect the standard “global” MQTT Settings of your netFIELD OS in the Local Device Manager under **General Settings** > **Default MQTT Client Settings**.

Element	Description
MQTT version	MQTT version to be used (depending on the MQTT Broker).
Keep alive interval	Defines the maximum length of time in seconds that the broker and client may not communicate with each other.
User name	User name for authentication at the Broker (if implemented and required by the Broker). Note that the Mosquitto Broker deployed from the netFIELD Portal does not require login authentication.
Password	Password for authentication at the Broker (if implemented and required by the Broker). Note that the Mosquitto Broker from the netFIELD Portal does not require login authentication.
Connect timeout	Defines the maximum length of time in seconds that is allowed for completing the connection process.

Element	Description	
Clean session	If **Clean session** is selected, the client does not want a persistent session (meaning that if the client disconnects for any reason, all information and messages that are queued from a previous persistent session are lost. If **Clean session** is unchecked, the broker creates a persistent session for the client.	
Server URIs	Server URI of the MQTT Broker **Note**: When multiple server URIs are specified, the client will try to connect to each server one after the other, starting with the first server in the list. If a server connection was established successfully, only this connection will be used. The client will not open multiple connections to multiple servers simultaneously.	
Last Will and Testament	Select this option if you want to use the “last will and testament” (LWT) feature of MQTT. (I.e. to notify other clients about an unexpected loss of connection to the broker)	
	Topic name	Topic name of LWT message
	Retained	“Retained” flag of LWT message
	Quality of Service	QoS of LWT message
	Message	Message text, e.g. “unexpected loss of connection”
SSL / TLS	Select this option if you want to use SSL/TLS encryption for creating a secure connection to the MQTT Broker. **Note**: This option is for expert users only! In the standard use case, in which the Mosquitto Broker and the OPC UA Client app are running on the same device, a secure SSL/TLS connection is not necessary (because the connection is “internal” and the overhead of the secure connection can thus be avoided). If you want to use SSL/TLS encryption anyway, see section *Using SSL/TLS encryption (optional)* [▸ page 42] for further information.	
	File name and path to private key in PEM format	Enter here the complete path to the private key on the device; e.g.: `/etc/ssl/private/client-key.pem`
	File name and path to certificate chains in PEM format	Enter here the complete path to the certificate chains on the device; e.g.: `/etc/ssl/services/client-cert.pem`
	Override the trusted CA certificates in PEM format	Enter here the complete path to override the trusted CA certificates on the device; e.g.: `/etc/ssl/services/ca-cert.pem`
	Enable verification of the server certificate	If this option is disabled, the OPC UA Client app will also accept invalid certificates from the Broker (not recommended).

Table 14: MQTT Client Settings

➢ Click **Save** button to save your new MQTT Client Settings.

➪ The **Succeeded to save MQTT client settings** message appears.

➢ Check the **MQTT Broker connection status** indicator in the footer to see if the connection to the new server has been successfully established:

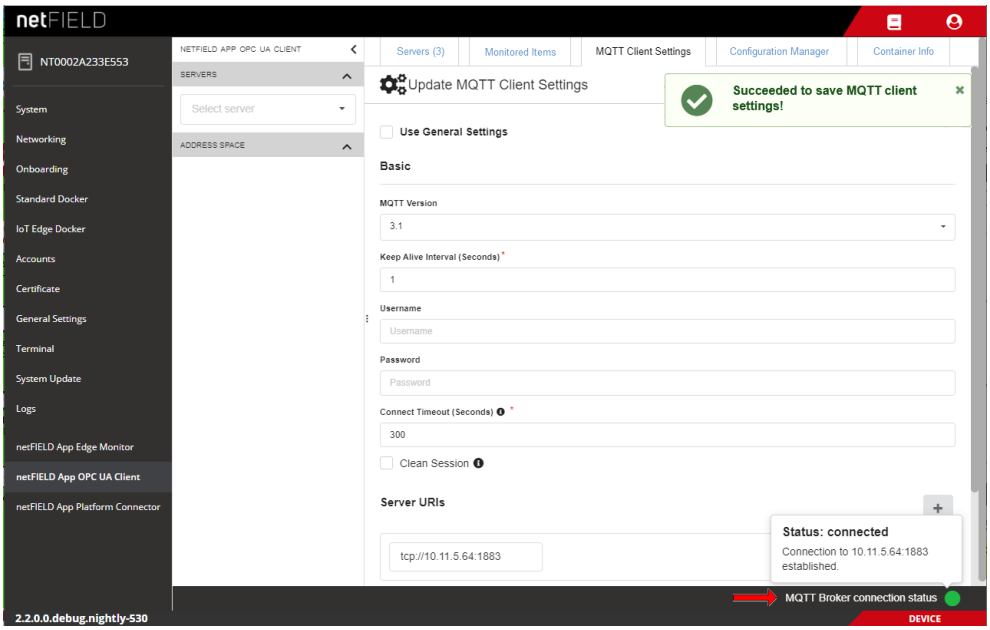

Figure 22: MQTT server connection status indicator in footer

5.6 Configuration Manager

In the **Configuration Manager** tab, you can save the OPC UA Client app configuration settings to your local PC by downloading it. You can also restore a formerly saved configuration by uploading the configuration file. The download/upload function allows you to practically “clone” your configuration and use it in other OPC UA Client instances (e.g. running on other netFIELD Edge Devices or netFIELD OS Datacenters).

Note:
The OPC UA Client app uses an encryption key to generate a security hash tag for server access credentials when you add a new OPC UA Server (that requires a certificate or user name and password) to your local configuration. This ensures that the credentials in the configuration of your OPC UA Client instance become “portable”; i.e. that they can be used by other instances of the OPC UA Client app when the configuration is exported/ downloaded and imported/uploaded to other instances.
Note that every other instance of the OPC UA Client that shall use an imported configuration containing server credentials must have the same encryption key in its environment variables.

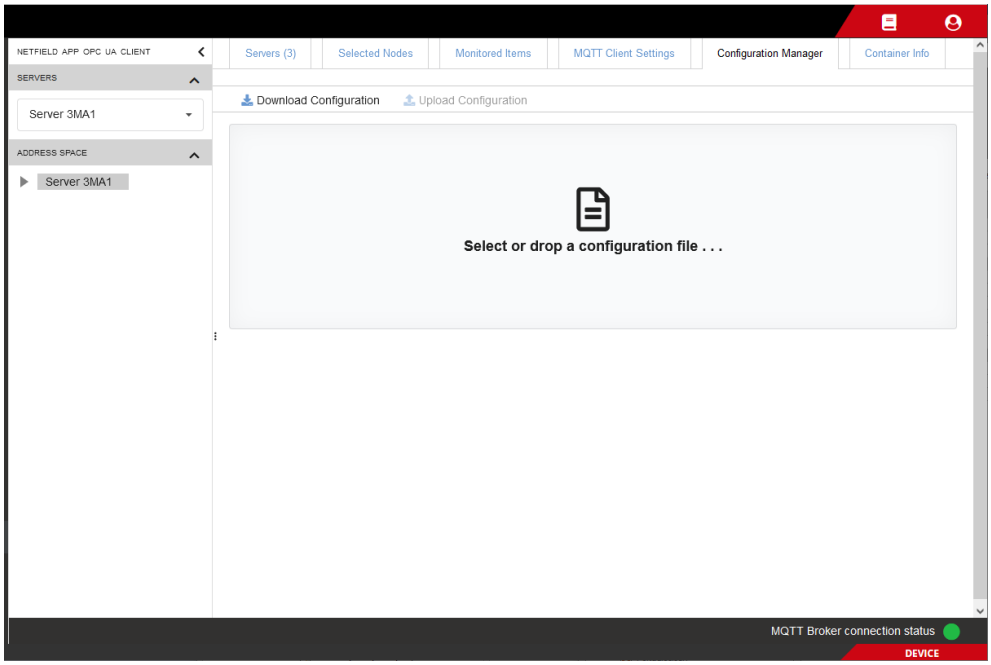

Figure 23: Configuration Backup

Save configuration

➢ To save your current configuration, click **Download Configuration** button.

The configuration settings are saved to your local PC as ZIP file. (The download path depends on the settings of your web browser.)
The name of the ZIP file is made up by the gateway prefix, app name and date/time of the download.

Note:
The “gateway prefix” is by default the hardware ID of the Edge Device on which the OPC UA Client app is deployed.

Restore/import configuration

To restore a formerly saved configuration (or import it into other instances), you must first select the configuration ZIP file by dragging and dropping it from your desktop onto the grey field (as an alternative, you can open the standard Windows file selection dialog by clicking into the grey field).

After having selected the file, the **Upload Configuration** button is enabled, and you can now “load” the configuration by clicking the button.

Important:
The **Upload Configuration** function will overwrite the current configuration settings. We recommend you to save your current configuration before using this function.

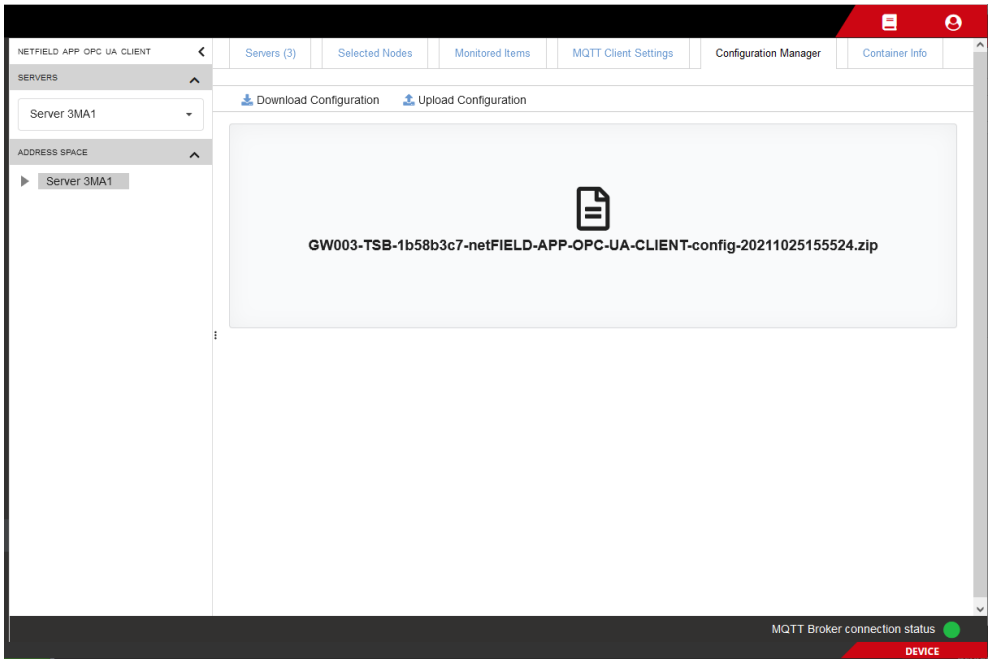

Figure 24: Upload Configuration

5.7 Container Info

The **Container Info** tab shows general information about the container.

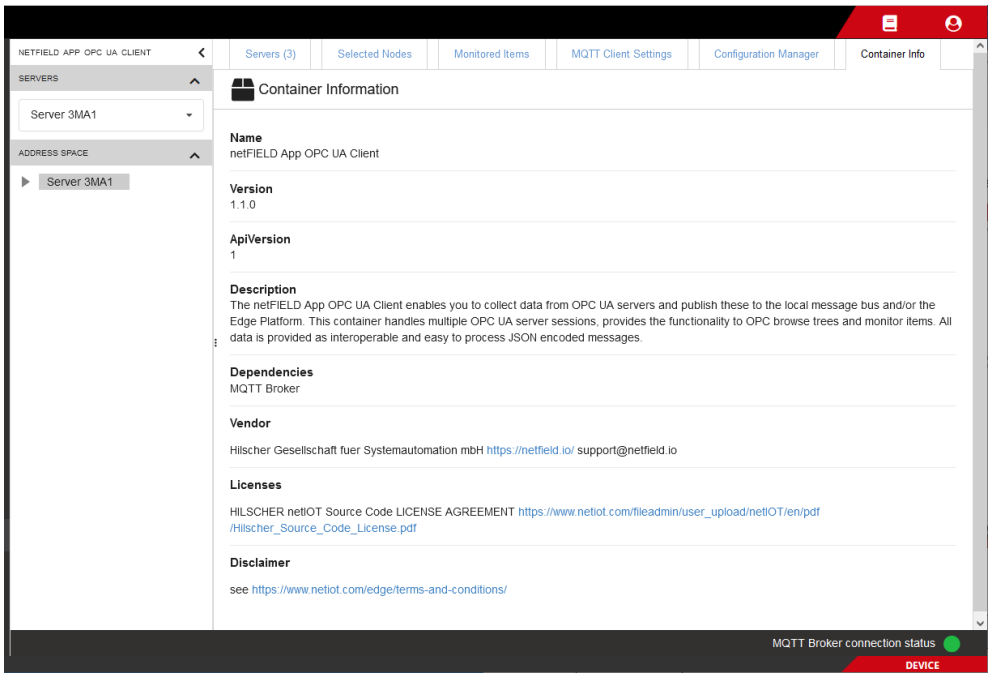

Figure 25: Container Info

Category	Description
Name	Container name
Version	Container software version
API Version	REST API version of the container
Description	Brief description of the function of the container
Dependencies	Other containers or components required for proper operation of the container
Vendor	Vendor of container
Licenses	Name of the software license(s), under which the container was published
Disclaimer	Path/link to the software license(s)

Table 15: Info tab

6 Use case example: Monitor node data in netFIELD Portal

This chapter describes how to subscribe to published nodes/topics with the *netFIELD App Platform Connector* and monitor the data in the netFIELD Portal.

Requirements

- You have deployed the *netFIELD App Platform Connector* (with its default container settings) on the Edge Device
- You have deployed the *mosquitto* MQTT Broker container (with its default container settings) on the Edge Device
- You have created monitored items in the netFIELD App OPC UA Client
- You are logged-in to the Local Device Manager
- You have a netFIELD Portal account

Step-by-step instructions

1. Copy Topic to clipboard.
   ➢ In the **Monitored Items** tab of the netFIELD App OPC UA Client, click on the icon in the **Topic** column of the topic to which you want to subscribe.

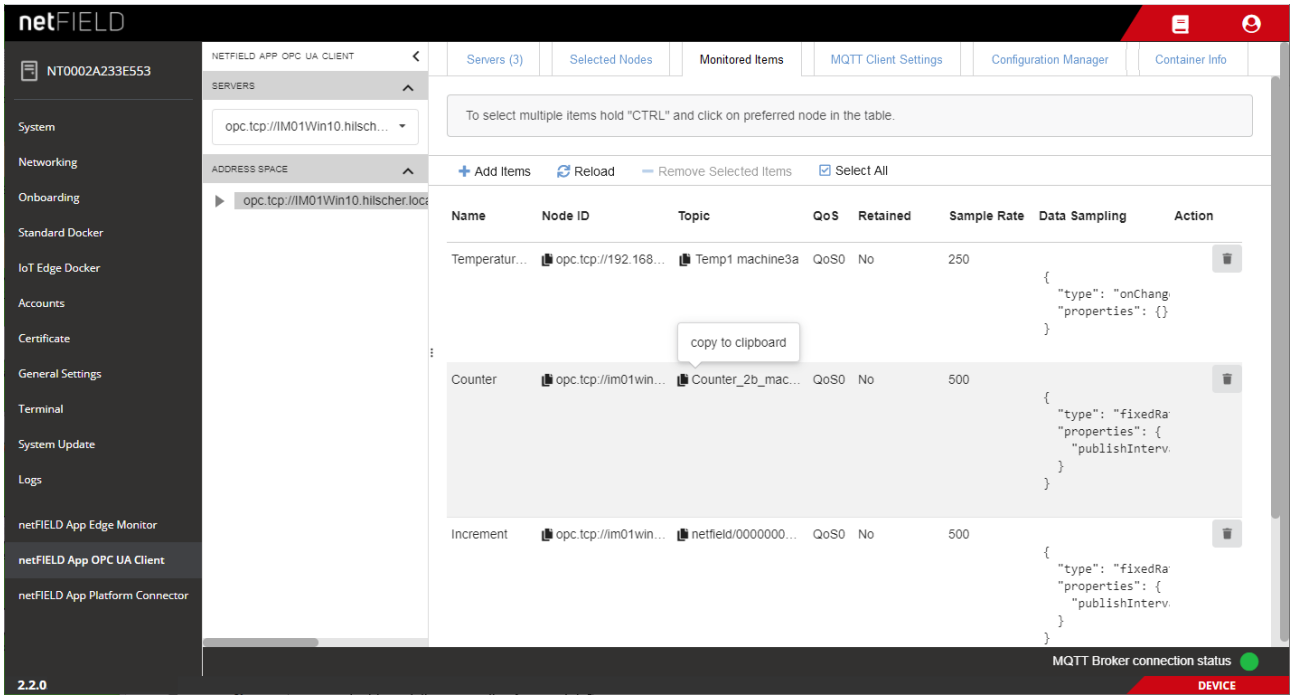

Figure 26: Select Topic for subscription

2. Subscribe to Topic.
   ➢ In the navigation panel of the Local Device Manager, select **netFIELD APP Platform Connector**.

The Platform Connector dashboard opens:

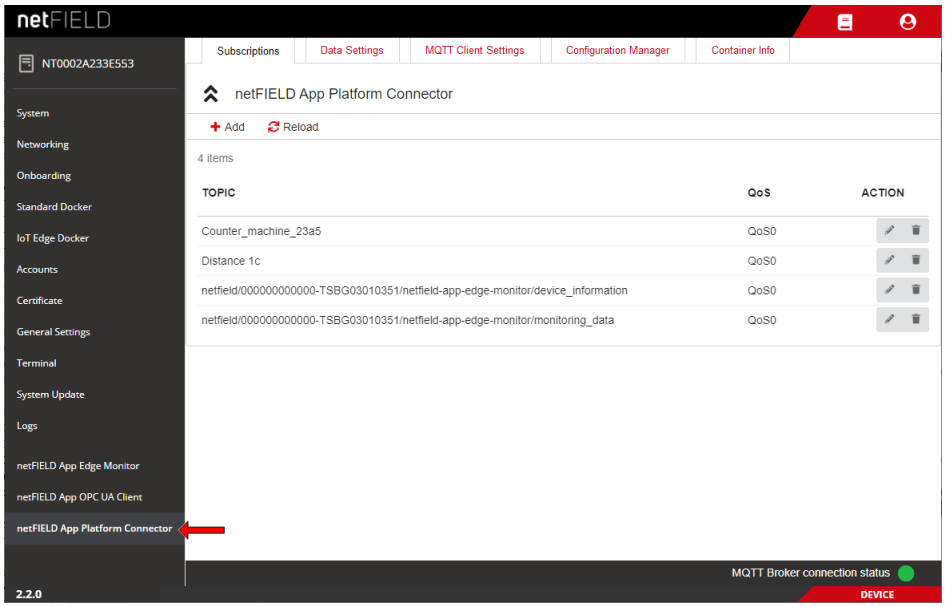

Figure 27: Subscribe to topic in Platform Connector 1

- In the **Subscriptions** tab, click **+ Add** button.
- In the **Topic** field, paste the string of the topic.

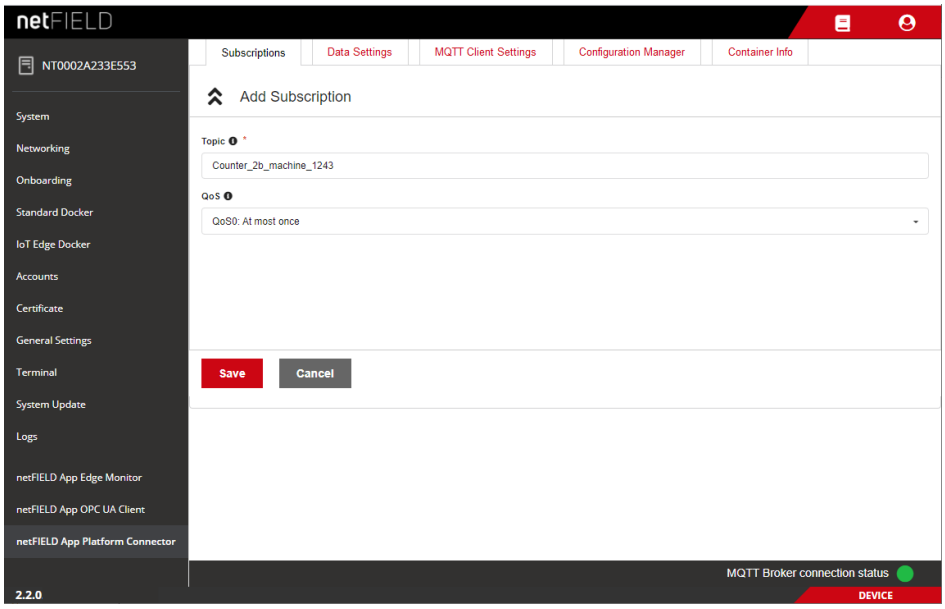

Figure 28: Subscribe to topic in Platform Connector 2

- In the **QoS** drop-down list, select the MQTT Quality of Service (default is QoS0).

Note:

This is the QoS of the message delivery from the MQTT Broker to the subscribing client, i.e. from *mosquitto* to the Platform Connector on the device. If you define here a QoS that is lower than the QoS defined for the topic in the OPC UA Client, the MQTT broker transmits the message with the lower QoS.
Note that the QoS does not relate to the messages that are being sent from the Platform Connector to the netFIELD Cloud.

➢ Click **Save** button to add the topic.

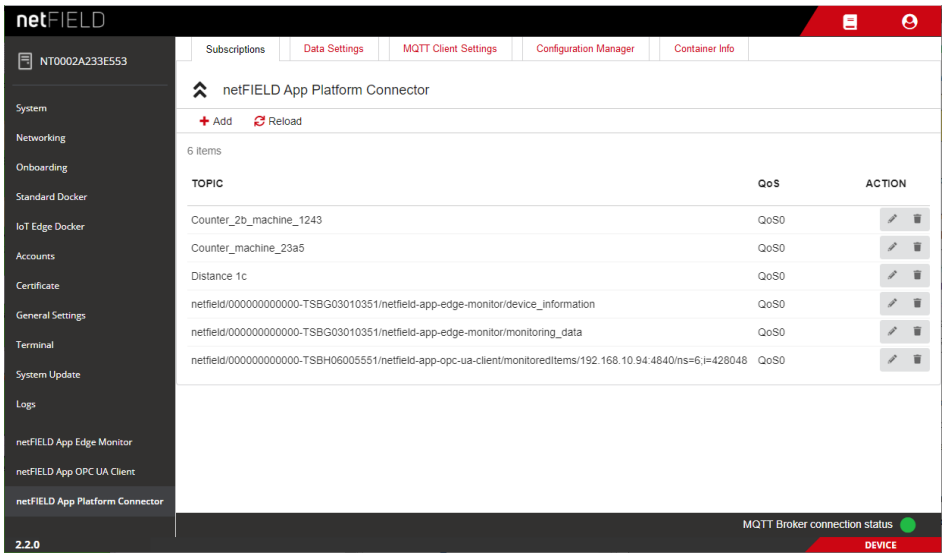

Figure 29: Subscribe to topic in Platform Connector 3

➪ The topic is added to the list and the Platform Connector (i.e. the MQTT client within the Platform Connector) sends a SUBSCRIBE message for this topic to the MQTT Broker. The new topic subscription will also be automatically added to the subscriptions list in the cloud dashboard.

3. Monitor Topic in netFIELD Portal.

➢ In the Portal, open the **Device Manager** app and select your Edge Device.

➢ In the **DEVICE NAVIGATION**, click on **netFIELD App Platform Connector** under **CONTAINERS**.

➪ The dashboard of the Platform Connector app opens.

➢ Open the **Topics** tab.

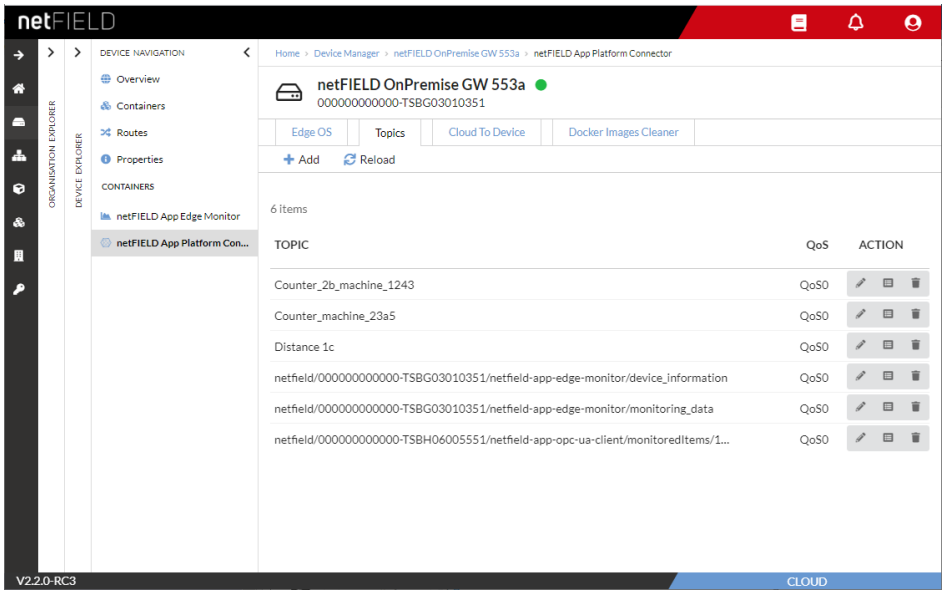

Figure 30: Platform Connector dashboard in Portal

Note:

Note that the **Topics** list in the dashboard in the Portal is identical with the **Topics** list in the Local Device Manager.

➢ To display the data, click on the topic or click the (View Data) button next to the topic.

➪ The current message data stream is displayed:

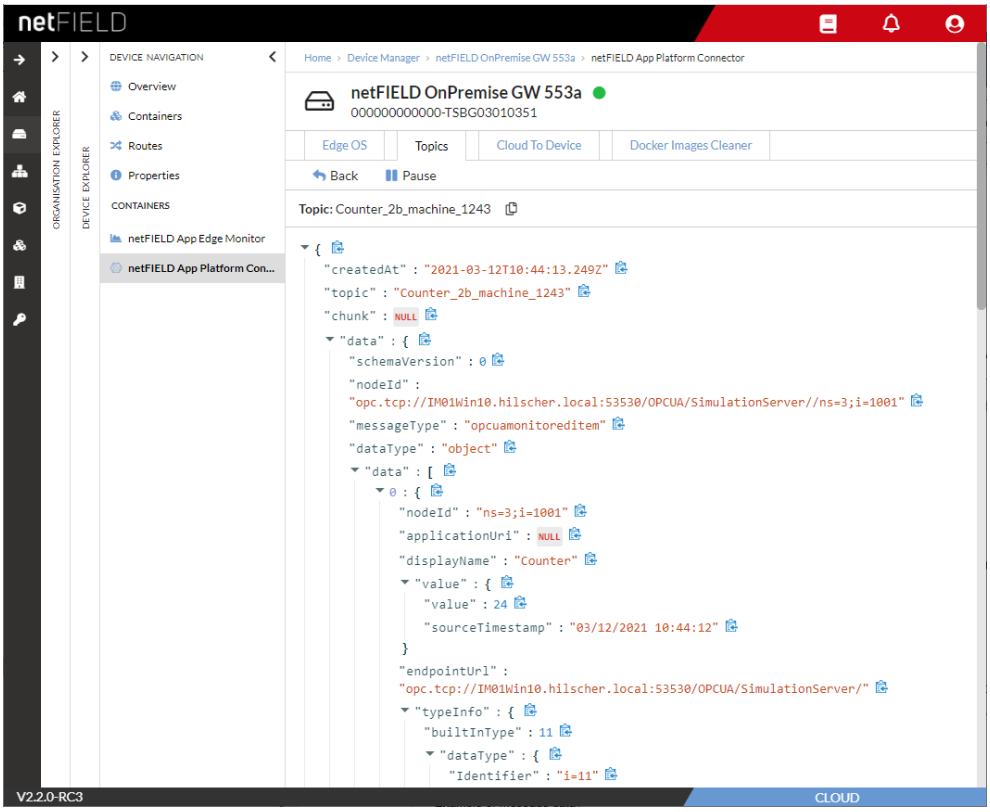

Figure 31: Example of message data in Platform Connector dashboard in Portal

The data stream page features the following control elements for navigation and inspection of the message data:

Control element	Description
Elements in toolbar	
Pause	“Freezes” the currently shown data set
Resume	Resumes to display the “live” data stream
Back	Brings you back to the topic subscriptions list
	Copies the topic to your clipboard
Elements in data set	
	Click on the arrow to display subordinate data elements (expand the structure)
	Click on the arrow to hide subordinate data elements (collapse the structure)
	Click on the icon to copy the data element and subordinate elements to your clipboard
	Indicates that the data element has been copied to the clipboard

Table 16: Control elements on data stream page

7 Good to know...

7.1 MQTT message format

Structure of MQTT Topic name string(default)

Note:
This is the structure of the MQTT Topic name string that will be used by default if you do not specify your own **Topic** string in the **User Specific Topic** field when you select a node for publishing (see section *Selected Nodes* [page 23]).

Default topic name syntax:

```
netfield/<gateway prefix>/netfield-app-opc-ua-client/monitoredItems/
<OPC UA Server URL>/<Node ID>
```

Elements in default topic name:

Element in topic	Description
`netfield`	General prefix for netFIELD application topics.
`<gateway prefix>`	Identifier according to global MQTT settings of the netFIELD OS. (General Settings > Default MQTT Client Settings > Gateway settings > Gateway prefix). **Note**: By default, the gateway prefix is identical with the Hardware ID of the Edge Device.
`netfield-app-opc-ua-client`	Name of the application container.
`monitoredItems`	Monitored items.
`<OPC UA Server URL>`	URL of the server supplying the monitored nodes.
`<Node ID>`	ID of the monitored OPC UA node.

Table 17: Descriptions of elements of topic name

Example:

```
netfield/000000000000-TSBG03010351/netfield-app-opc-ua-client/
monitoredItems/im01win10.hilscher.local:53530/ns=3;i=1001
```

Structure of message content in JSON


```
{
  "createdAt" : "2021-03-12T13:26:55.007Z"
  "topic" : "Saw Counter 1c 42"
  "chunk" : NULL
  "data" : {
    "schemaVersion" : 0
    "nodeId" : "opc.tcp://IM01Win10.hilscher.local:53530/OPCUA/SimulationServer//ns=3;i=1003"
    "messageType" : "opcuamonitoreditem"
    "dataType" : "object"
    "data" : [
      0 : {
        "nodeId" : "ns=3;i=1003"
        "applicationUri" : "urn:IM01Win10.hilscher.local:OPCUA:SimulationServer"
        "displayName" : "Sawtooth"
        "value" : {
          "value" : -2
          "sourceTimestamp" : "2021-03-12T13:26:55Z"
        }
        "endpointUrl" : "opc.tcp://IM01Win10.hilscher.local:53530/OPCUA/SimulationServer/"
        "typeInfo" : {
          "builtInType" : 11
          "dataType" : {
            "Identifier" : "i=11"
          }
          "valueRank" : -1
        }
      }
    ]
  }
}
```

7.2 Using SSL/TLS encryption (optional)

Please note the following if you intend to use SSL/TLS encryption:
The certificates and key files that the MQTT Client in the *netFIELD App OPC UA Client* container needs for establishing a secure SSL/TLS connection to the MQTT Broker are not managed by the *OPC UA Client* app container itself. Instead, they are to be stored on the Edge Device and mapped into the container from the netFIELD OS.
For this mapping, the following standard directories are mapped into the container if you use the default Container Create Options in the netFIELD Portal:
`/etc/ssl/`
`/usr/share/ca-certificates/`

Note:
If you require different directories for your use case, you may change the mapping of these “bind mounts” in the netFIELD Portal before deploying the container (see section *Create Options* [page 16]).

As a user, you can store your required keys and certificates in these directories. By selecting the **SSL / TLS** option on the **MQTT Client Settings** page, you can allow the MQTT Client in the *OPC UA Client* app container to use these files for establishing its secure SSL/TLS connection. Note that these keys and certificates must be stored in PEM format (a specific file format for storing this kind of data) and that you have to specify the full path to the appropriate PEM file in the corresponding fields of the **MQTT Client Settings** page. For example:

File name and path to private key in PEM format:
`/etc/ssl/private/client-key.pem`

File name and path to certificate chains in PEM format:
`/etc/ssl/services/client-cert.pem`

Override the trusted CA certificates in PEM format:
`/etc/ssl/services/ca-cert.pem`

Note also that if you intend to use more than one “secure” MQTT Broker (as listed in the **Server URIs** field), and thus require several different certificates, you have to store them *in one single* PEM file. This is because it is not possible to specify a list of multiple paths to separate PEM files for individual Brokers.

8 Legal notes

Important notes

Utmost care was/is given in the preparation of the documentation at hand consisting of a user's manual, operating manual and any other document type and accompanying texts. However, errors cannot be ruled out. Therefore, we cannot assume any guarantee or legal responsibility for erroneous information or liability of any kind. You are hereby made aware that descriptions found in the user's manual, the accompanying texts and the documentation neither represent a guarantee nor any indication on proper use as stipulated in the agreement or a promised attribute. It cannot be ruled out that the user's manual, the accompanying texts and the documentation do not completely match the described attributes, standards or any other data for the delivered product. A warranty or guarantee with respect to the correctness or accuracy of the information is not assumed.

We reserve the right to modify our products and the specifications for such as well as the corresponding documentation in the form of a user's manual, operating manual and/or any other document types and accompanying texts at any time and without notice without being required to notify of said modification. Changes shall be taken into account in future manuals and do not represent an obligation of any kind, in particular there shall be no right to have delivered documents revised. The manual delivered with the product shall apply.

Under no circumstances shall Hilscher Gesellschaft für Systemautomation mbH be liable for direct, indirect, ancillary or subsequent damage, or for any loss of income, which may arise after use of the information contained herein.

Liability disclaimer

The hardware and/or software was created and tested by Hilscher Gesellschaft für Systemautomation mbH with utmost care and is made available as is. No warranty can be assumed for the performance or flawlessness of the hardware and/or software under all application conditions and scenarios and the work results achieved by the user when using the hardware and/or software. Liability for any damage that may have occurred as a result of using the hardware and/or software or the corresponding documents shall be limited to an event involving willful intent or a grossly negligent violation of a fundamental contractual obligation. However, the right to assert damages due to a violation of a fundamental contractual obligation shall be limited to contract-typical foreseeable damage.

It is hereby expressly agreed upon in particular that any use or utilization of the hardware and/or software in connection with

- Flight control systems in aviation and aerospace;
- Nuclear fission processes in nuclear power plants;
- Medical devices used for life support and
- Vehicle control systems used in passenger transport

shall be excluded. Use of the hardware and/or software in any of the following areas is strictly prohibited:

- For military purposes or in weaponry;
- For designing, engineering, maintaining or operating nuclear systems;
- In flight safety systems, aviation and flight telecommunications systems;
- In life-support systems;
- In systems in which any malfunction in the hardware and/or software may result in physical injuries or fatalities.

You are hereby made aware that the hardware and/or software was not created for use in hazardous environments, which require fail-safe control mechanisms. Use of the hardware and/or software in this kind of environment shall be at your own risk; any liability for damage or loss due to impermissible use shall be excluded.

Warranty

Hilscher Gesellschaft für Systemautomation mbH hereby guarantees that the software shall run without errors in accordance with the requirements listed in the specifications and that there were no defects on the date of acceptance. The warranty period shall be 12 months commencing as of the date of acceptance or purchase (with express declaration or implied, by customer's conclusive behavior, e.g. putting into operation permanently).

The warranty obligation for equipment (hardware) we produce is 36 months, calculated as of the date of delivery ex works. The aforementioned provisions shall not apply if longer warranty periods are mandatory by law pursuant to Section 438 (1.2) BGB, Section 479 (1) BGB and Section 634a (1) BGB [Bürgerliches Gesetzbuch; German Civil Code] If, despite of all due care taken, the delivered product should have a defect, which already existed at the time of the transfer of risk, it shall be at our discretion to either repair the product or to deliver a replacement product, subject to timely notification of defect.

The warranty obligation shall not apply if the notification of defect is not asserted promptly, if the purchaser or third party has tampered with the products, if the defect is the result of natural wear, was caused by unfavorable operating conditions or is due to violations against our operating regulations or against rules of good electrical engineering practice, or if our request to return the defective object is not promptly complied with.

Costs of support, maintenance, customization and product care

Please be advised that any subsequent improvement shall only be free of charge if a defect is found. Any form of technical support, maintenance and customization is not a warranty service, but instead shall be charged extra.

Additional guarantees

Although the hardware and software was developed and tested in-depth with greatest care, Hilscher Gesellschaft für Systemautomation mbH shall not assume any guarantee for the suitability thereof for any purpose that was not confirmed in writing. No guarantee can be granted whereby the hardware and software satisfies your requirements, or the use of the hardware and/or software is uninterruptable or the hardware and/or software is fault-free.

It cannot be guaranteed that patents and/or ownership privileges have not been infringed upon or violated or that the products are free from third-party influence. No additional guarantees or promises shall be made as to whether the product is market current, free from deficiency in title, or can be integrated or is usable for specific purposes, unless such guarantees or promises are required under existing law and cannot be restricted.

Confidentiality

The customer hereby expressly acknowledges that this document contains trade secrets, information protected by copyright and other patent and ownership privileges as well as any related rights of Hilscher Gesellschaft für Systemautomation mbH. The customer agrees to treat as confidential all of the information made available to customer by Hilscher Gesellschaft für Systemautomation mbH and rights, which were disclosed by Hilscher Gesellschaft für Systemautomation mbH and that were made accessible as well as the terms and conditions of this agreement itself.

The parties hereby agree to one another that the information that each party receives from the other party respectively is and shall remain the intellectual property of said other party, unless provided for otherwise in a contractual agreement.

The customer must not allow any third party to become knowledgeable of this expertise and shall only provide knowledge thereof to authorized users as appropriate and necessary. Companies associated with the customer shall not be deemed third parties. The customer must obligate authorized users to confidentiality. The customer should only use the confidential information in connection with the performances specified in this agreement.

The customer must not use this confidential information to his own advantage or for his own purposes or rather to the advantage or for the purpose of a third party, nor must it be used for commercial purposes and this confidential information must only be used to the extent provided for in this agreement or otherwise to the extent as expressly authorized by the disclosing party in written form. The customer has the right, subject to the obligation to confidentiality, to disclose the terms and conditions of this agreement directly to his legal and financial consultants as would be required for the customer's normal business operation.

Export provisions

The delivered product (including technical data) is subject to the legal export and/or import laws as well as any associated regulations of various countries, especially such laws applicable in Germany and in the United States. The products / hardware / software must not be exported into such countries for which export is prohibited under US American export control laws and its supplementary provisions. You hereby agree to strictly follow the regulations and to yourself be responsible for observing them. You are hereby made aware that you may be required to obtain governmental approval to export, reexport or import the product.

List of Figures

List of Tables

Contacts

HEADQUARTERS

Germany

Hilscher Gesellschaft für
Systemautomation mbH
Rheinstrasse 15
65795 Hattersheim
Phone: +49 (0) 6190 9907-0
Fax: +49 (0) 6190 9907-50
E-mail: info@hilscher.com

Support

Phone: +49 (0) 6190 9907-99
E-mail: de.support@hilscher.com

SUBSIDIARIES

China

Hilscher Systemautomation (Shanghai) Co. Ltd.
200010 Shanghai
Phone: +86 (0) 21-6355-5161
E-mail: info@hilscher.cn

Support

Phone: +86 (0) 21-6355-5161
E-mail: cn.support@hilscher.com

France

Hilscher France S.a.r.l.
69800 Saint Priest
Phone: +33 (0) 4 72 37 98 40
E-mail: info@hilscher.fr

Support

Phone: +33 (0) 4 72 37 98 40
E-mail: fr.support@hilscher.com

India

Hilscher India Pvt. Ltd.
Pune, Delhi, Mumbai
Phone: +91 8888 750 777
E-mail: info@hilscher.in

Italy

Hilscher Italia S.r.l.
20090 Vimodrone (MI)
Phone: +39 02 25007068
E-mail: info@hilscher.it

Support

Phone: +39 02 25007068
E-mail: it.support@hilscher.com

Japan

Hilscher Japan KK
Tokyo, 160-0022
Phone: +81 (0) 3-5362-0521
E-mail: info@hilscher.jp

Support

Phone: +81 (0) 3-5362-0521
E-mail: jp.support@hilscher.com

Korea

Hilscher Korea Inc.
Seongnam, Gyeonggi, 463-400
Phone: +82 (0) 31-789-3715
E-mail: info@hilscher.kr

Switzerland

Hilscher Swiss GmbH
4500 Solothurn
Phone: +41 (0) 32 623 6633
E-mail: info@hilscher.ch

Support

Phone: +49 (0) 6190 9907-99
E-mail: ch.support@hilscher.com

USA

Hilscher North America, Inc.
Lisle, IL 60532
Phone: +1 630-505-5301
E-mail: info@hilscher.us

Support

Phone: +1 630-505-5301
E-mail: us.support@hilscher.com