



**Operating instruction manual  
netFIELD App EtherCAT Tap**

**Hilscher Gesellschaft für Systemautomation mbH  
[www.hilscher.com](http://www.hilscher.com)**

DOC201102OI04EN | Revision 4 | English | 2023-02 | Released | Public

# Table of contents

<b>1</b>	<b>Introduction .....</b>	<b>3</b>
1.1	About this document .....	3
1.1.1	Description of the contents .....	3
1.1.2	List of revisions .....	3
1.1.3	Conventions in this document.....	3
1.2	Brief description .....	4
1.3	General Requirements .....	5
1.3.1	MQTT broker.....	5
1.3.2	License.....	5
1.3.3	Memory .....	6
<b>2</b>	<b>Use cases .....</b>	<b>7</b>
2.1	Overview .....	7
2.2	Connecting EtherCAT network via netMIRROR (hardware TAP).....	8
2.3	Connecting EtherCAT network as software TAP .....	12
<b>3</b>	<b>Start parameters of the container .....</b>	<b>15</b>
<b>4</b>	<b>Configuration web pages of the EtherCAT Tap .....</b>	<b>16</b>
4.1	Overview .....	16
4.2	“Status” tab.....	17
4.3	“About” tab .....	18
4.4	“Configuration Backup” tab .....	19
4.5	“Publisher Management” tab.....	21
4.6	MQTT Client Settings.....	24
4.7	License Server Settings .....	27
4.8	EtherCAT configuration.....	29
4.8.1	OT Interface settings.....	29
4.8.2	EtherCAT Configuration node.....	31
4.8.3	How to configure the EtherCAT Tap with ENI file .....	38
<b>5</b>	<b>Good to know.....</b>	<b>45</b>
5.1	MQTT message format .....	45
5.2	Known restrictions of the EtherCAT Tap.....	47
5.3	Mapping of TwinCAT data types to EtherCAT Tap-supported data types .....	48
5.4	Using SSL/TLS encryption (optional).....	49
5.5	Installing CodeMeter and activating container license .....	50
<b>6</b>	<b>Legal notes .....</b>	<b>56</b>
	<b>List of Figures .....</b>	<b>60</b>
	<b>List of Tables.....</b>	<b>62</b>
	<b>Contacts.....</b>	<b>63</b>

# 1 Introduction

## 1.1 About this document

### 1.1.1 Description of the contents

This document describes the netFIELD App **EtherCAT Tap** from Hilscher.

### 1.1.2 List of revisions

Index	Date	Revision
4	2023-02-27	Document completely revised

Table 1: List of revisions

### 1.1.3 Conventions in this document

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

#### Notes



#### Important:

<important note>



#### Note:

<simple note>



<note, where to find further information>

#### Operation instructions

1. <operational step>

➤ <instruction>

➤ <instruction>

2. <operational step>

➤ <instruction>

➤ <instruction>

#### Results

↻ <intermediate result>

⇒ <final result>

## 1.2 Brief description

**EtherCAT Tap** is an application container that monitors EtherCAT traffic and publishes selected machinery process data via MQTT. It allows you to “tap into” the process data and to run data analytics applications without interfering neither with your existing network setup nor with your EtherCAT data exchange. Because there is no need to change the configuration of your PLC or machinery, it can thus be easily used in “brown-field” plants/automation networks.

The EtherCAT Tap features a web-based configuration GUI that can be easily accessed via HTTP(S) browser connection without having to install any additional software tools.

It allows you to conveniently build your EtherCAT configuration (including process data semantics and variables) by simply importing an ENI (EtherCAT Network Information) file from TwinCAT. This configuration GUI is described in chapter *Configuration web pages of the EtherCAT Tap* [► page 16].

## 1.3 General Requirements

### 1.3.1 MQTT broker

The EtherCAT Tap container publishes the acquired data via MQTT and thus requires an MQTT broker for operation. The MQTT broker can be running on the same host or on a different host machine in your local IT network.

### 1.3.2 License

#### CodeMeter licensing technology

The netFIELD App EtherCAT Tap is protected by the *CodeMeter* licensing technology from *Wibu-Systems*. In order to use the app properly, you need a *CodeMeter* license server (*CodeMeter User Runtime*) and the EtherCAT Tap license key, which you must download and activate in the license server.

For purchasing the license key, please contact your Hilscher sales representative.

If you are not already a user of the *CodeMeter* ecosystem and have not implemented a *CodeMeter* license server yet, you can download the *CodeMeter User Runtime* from *Wibu-Systems* under <https://www.wibu.com/us/support/user/downloads-user-software.html> and install it on a local server/machine/PC that can be reached by the EtherCAT Tap via TCP/IP connection. Exemplary instructions on how to install the license server and import and activate the EtherCAT Tap license key are provided in section *Installing CodeMeter and activating container license* [▶ page 50].

Note that for you as an end-user, the *CodeMeter* technology is free-of-charge, you only have to pay for the EtherCAT Tap license key from Hilscher.

#### Floating license

The EtherCAT Tap license is a “floating” license, which means that it is not bound to an individual instance of the app running on a particular device. Whichever instance of the app “grabs” the license from the *CodeMeter* server first, can use it until the instance itself closes the connection to the server again (either by the application container being shut down altogether or by deleting the corresponding license server endpoint on the **License Server Settings** page of the app). Other instances of the EtherCAT Tap (e.g. a EtherCAT Tap container running on another device) can then “take over” the “freed” license by simply connecting to the corresponding license server.

### Maintenance period

Currently, the netFIELD App EtherCAT Tap license is a “one-time payment license”, meaning that you can use the container license without time restriction, once you have paid the license fee.

The maintenance period, in which you are entitled to deploy software updates of the container, is one year. After this period, you can still use your last deployed container version indefinitely; however, for using higher container versions, you will have to acquire a new or updated license.

### License-free demo mode

Without a license, you can still deploy, configure and operate the EtherCAT Tap for testing purposes; however, all acquired and published process data values will be invalidated (set to null), except for the first two variables of your **Active Publishers** list (see section “*Publisher Management*” *tab* [▶ page 21]), which will be available with “correct” values for 30 minutes. After 30 minutes, the first two variables will also be set to null.

## 1.3.3 Memory

The RAM needed by the EtherCAT Tap depends on the amount of EtherCAT data that you intend to capture and process in your application. We recommend you to provide at least **1 GB of free RAM** on your Edge Device (ideally 2 GB or more). In use cases with a stable amount of little data processing, less than 1 GB RAM might be sufficient.

## 2 Use cases

### 2.1 Overview

The netX SoC is a multi-protocol communication controller from Hilscher, which is available as single chip, embedded module (like e.g. comX) or cifX PC card.

You can use a netX-based Industrial Ethernet interface together with the netANALYZER driver as “capture interface” for your EtherCAT Tap container.

The netANALYZER driver is included in the netFIELD Operating System (netFIELD OS), but will also be available as separate driver in the near future.

Being equipped with a netX-based cifX interface and the netFIELD OS (which features the required netANALYZER driver and a container engine), the netFIELD OnPremise edge gateway (NIOT-E-TIJCX-GB-RE/NFLD) is currently the ideal ready-to-use host device for your EtherCAT Tap.

Note that it is best practice to place your host device (respectively the “mirror” device [like the netMIRROR] if you are using one) between the Master and the first Slave device of the EtherCAT network, because this is usually the position where all process data frames of all devices will be present for capturing.

## 2.2 Connecting EtherCAT network via netMIRROR (hardware TAP)

You can use the Hilscher **netMIRROR** (NMR-TFE-RE, part no. 7340.100) device as a hardware-based “TAP” (Test Access Point) in connection with your netFIELD OnPremise device. The netMIRROR device “mirrors” the Ethernet data traffic with virtually no delay (~1 ns) and without affecting the OT network. It can be placed easily between your EtherCAT Master and the first Slave device without further PLC “engineering”. In addition, by using such a hardware mirror, you avoid the risk of traffic disruption if the OnPremise device is turned off or configured, or if a software update is performed on that device.



For technical information about the **netMIRROR** device, see user manual *netMIRROR – NMR-TFE-RE*, DOC161104UMxxEN.

The requirements for this use case are:

- netMIRROR device
- Your host device is equipped with a netX-based Industrial Ethernet interface (e.g. *cifX* in OnPremise edge gateway)
- netFIELD Operating System (“netFIELD OS”, includes the required netANALYZER driver)  
or  
3<sup>rd</sup> party operating system with netANALYZER driver

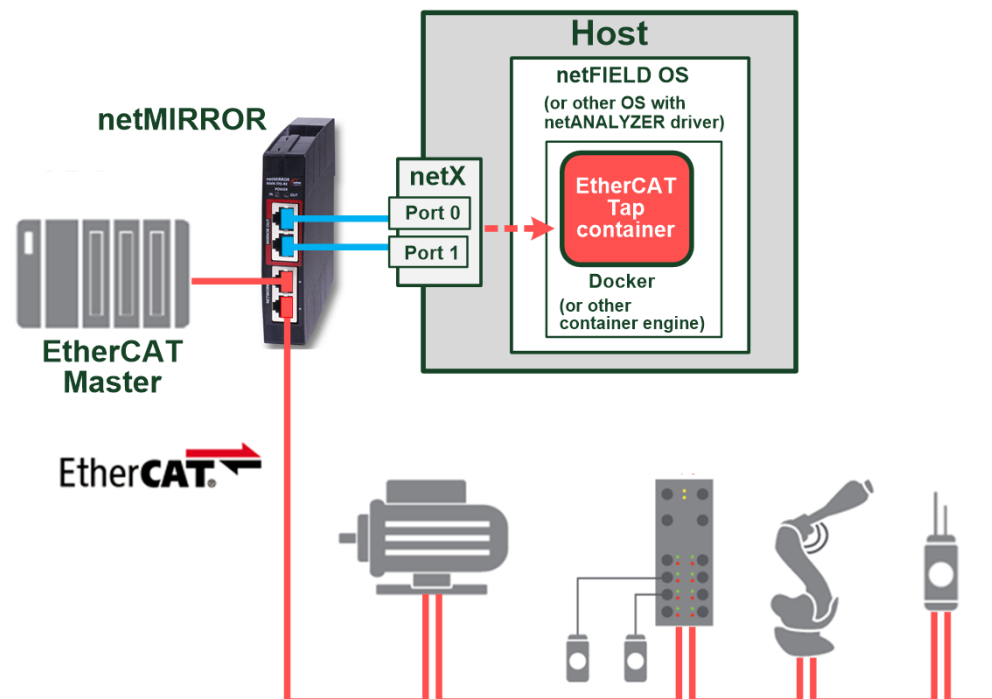


Figure 1: netX-based interface with netMIRROR

- Connect the **NETWORK** port **B** of the netMIRROR to the EtherCAT Master.
- Connect the **NETWORK** port **A** of the netMIRROR to the first EtherCAT Slave device.
- Connect both **MIRROR OUT** ports of the netMIRROR to the ports of the netX-based interface.



You can use “patch” or “crossover” cables. Note that you must set the **Crossover Mode** for the ports on the **OT Interface Settings** page accordingly (see section *OT Interface settings* [▶ page 29]). Note also that the signals at MIRROR OUT **Port A** are already crossed. Therefore you must configure the OT Interface Settings as shown in the figures below:

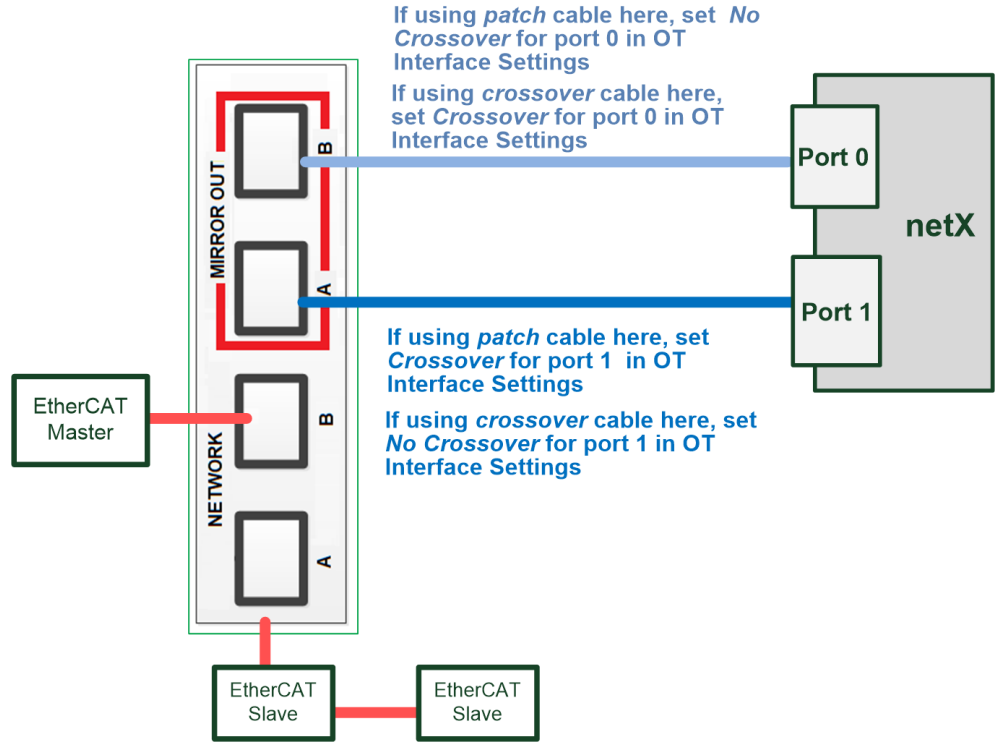


Figure 2: Connecting netMIRROR ports to netX ports: B to 0 and A to 1

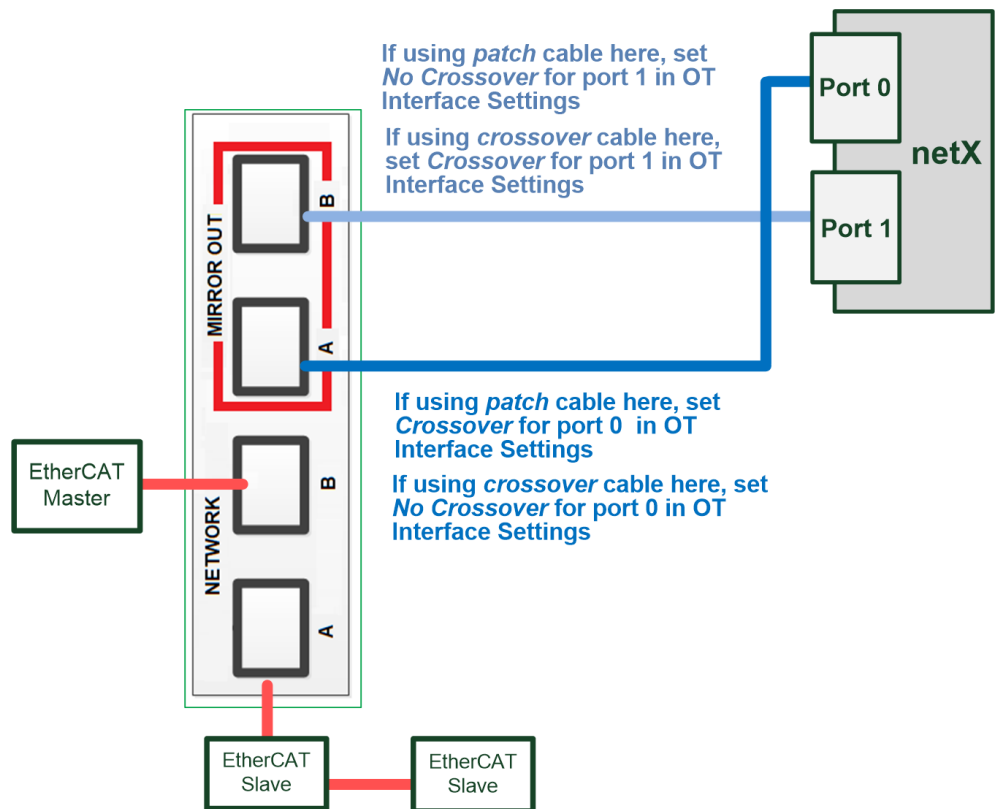


Figure 3: Connecting netMIRROR ports to netX ports: B to 1 and A to 0

- Set the **Link Speed** of the netX-based interface to 100 Mbit/s fixed.

This can be done in the configuration GUI of the EtherCAT Tap in the Local Device Manager (see section *OT Interface settings* [▶ page 29]). In the **Status** tab of the EtherCAT Tap configuration GUI, you can check whether the **Link states at port 0 and 1** of the netX-based interface are in state “up” (which should be the case if you have connected the netMIRROR correctly). See section “*Status*” tab [▶ page 17] for details.

**Note:**

When using a Hardware TAP, we strongly recommend you to keep the **Direction detection** option of the EtherCAT Tap on its default setting, which is `Automatic direction detection`.

This is because the netMIRROR is affected by the dynamic RX/TX channel negotiation (auto-mdix) that takes place after starting the EtherCAT bus communication. The fixed frame direction setting of the EtherCAT Tap might become obsolete if the EtherCAT bus is restarted again (with a different RX/TX channel assignment).

The **Direction detection** option can be found in the configuration GUI of the EtherCAT Tap under: **netFIELD App EtherCAT Tap > EtherCAT Configuration > General > Direction detection**.

(Select the  **EtherCAT Configuration** item in the navigation tree to display the **Direction detection** option.)

---

Note that the netX-based interface allows only “passive” data capturing. The EtherCAT Tap therefore also needs a “standard” LAN interface on the host for its TCP/IP, HTTP and MQTT communication. This LAN interface is necessary for deploying the container and for accessing its configuration web GUI. If your MQTT broker and your *CodeMeter* license server are running outside of your host, this interface will also allow the container to publish its MQTT data and to access the *CodeMeter* license server:

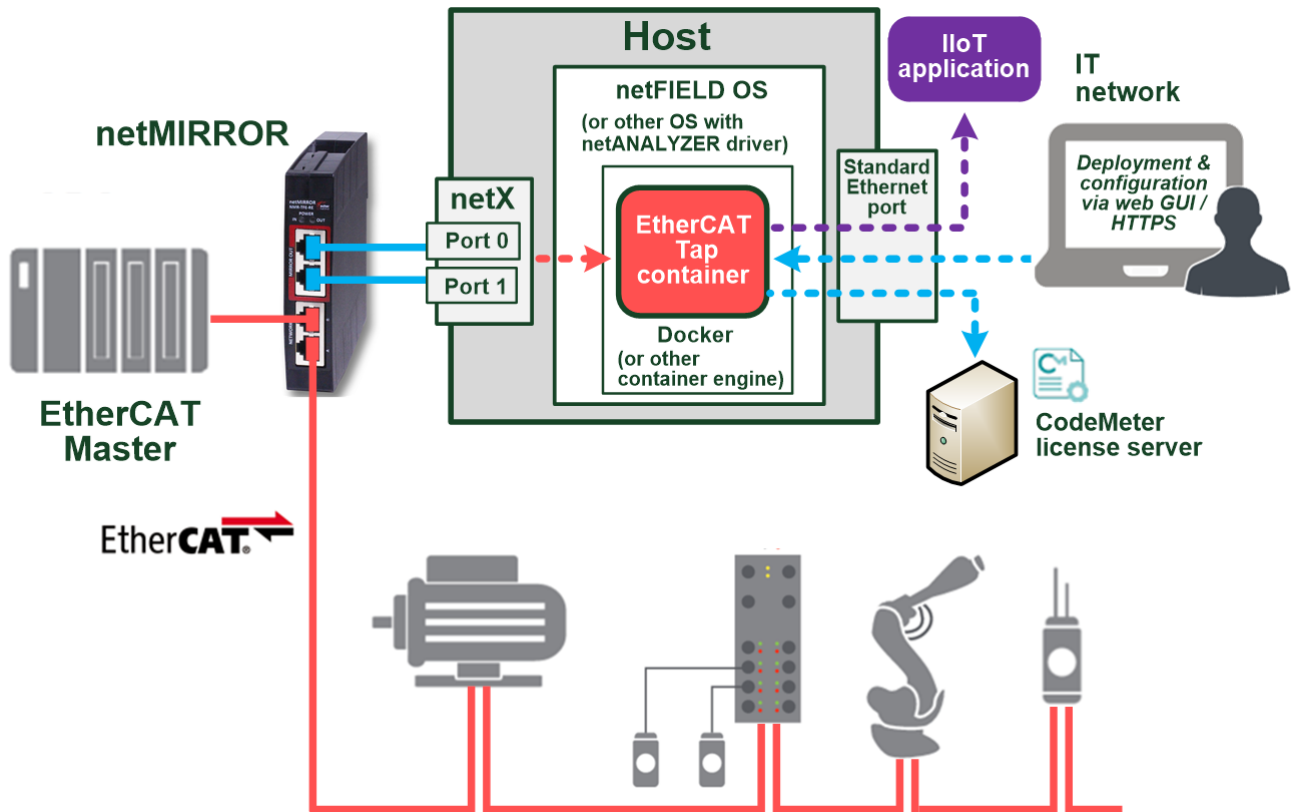


Figure 4: netX-based interface with netMIRROR and LAN connection

## 2.3 Connecting EtherCAT network as software TAP

You can connect the netX-based Industrial Ethernet interface of your host device directly to the EtherCAT Master, without interposed "hardware mirroring". This setup is also called a "software TAP".



### Note:

Note the disadvantages of the software TAP. It may disrupt traffic on the EtherCAT network in the following events:

- If the host device is turned off or configured
- If the EtherCAT Tap container is stopped or restarted
- If the EtherCAT network settings are changed

The requirements for this use case are:

- Your host device is equipped with a netX-based Industrial Ethernet interface (e.g. OnPremise edge gateway with *cifX*)
- netFIELD Operating System ("netFIELD OS", includes the required netANALYZER driver)  
or  
3<sup>rd</sup> party operating system with netANALYZER driver

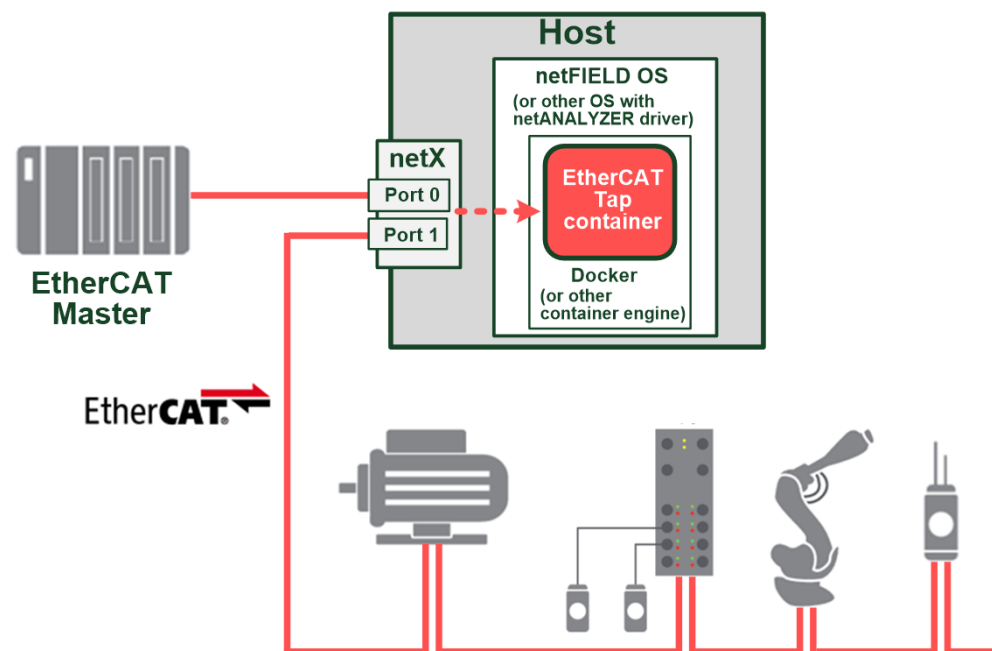


Figure 5: Connection without hardware mirror

- Connect the EtherCAT Master to port **0** of the netX-based interface. Connect the other port of the netX-based interface to the first EtherCAT Slave device.
- Make sure that the **Link Speed** of the netX-based interface is set to *Automatic*, which is the default setting. This can be done in the configuration web GUI of the EtherCAT Tap (see section *OT Interface settings* [▶ page 29]). In the **Status** tab of the EtherCAT Tap configuration GUI, you can check whether the **Link states at port 0 and 1** of the netX-based interface are in state "up" (which should be the case if you have connected the netX-based interface correctly). See section "*Status*" tab [▶ page 17] for details.

Note that the netX-based interface allows only “passive” data capturing. The EtherCAT Tap therefore also needs a “standard” LAN interface on the host for its TCP/IP, HTTP and MQTT communication. This LAN interface is necessary for deploying the container and for accessing its configuration web GUI. If your MQTT broker and your CodeMeter license server are running outside of your host, this interface will also allow the container to publish its MQTT data and to access the CodeMeter license server:

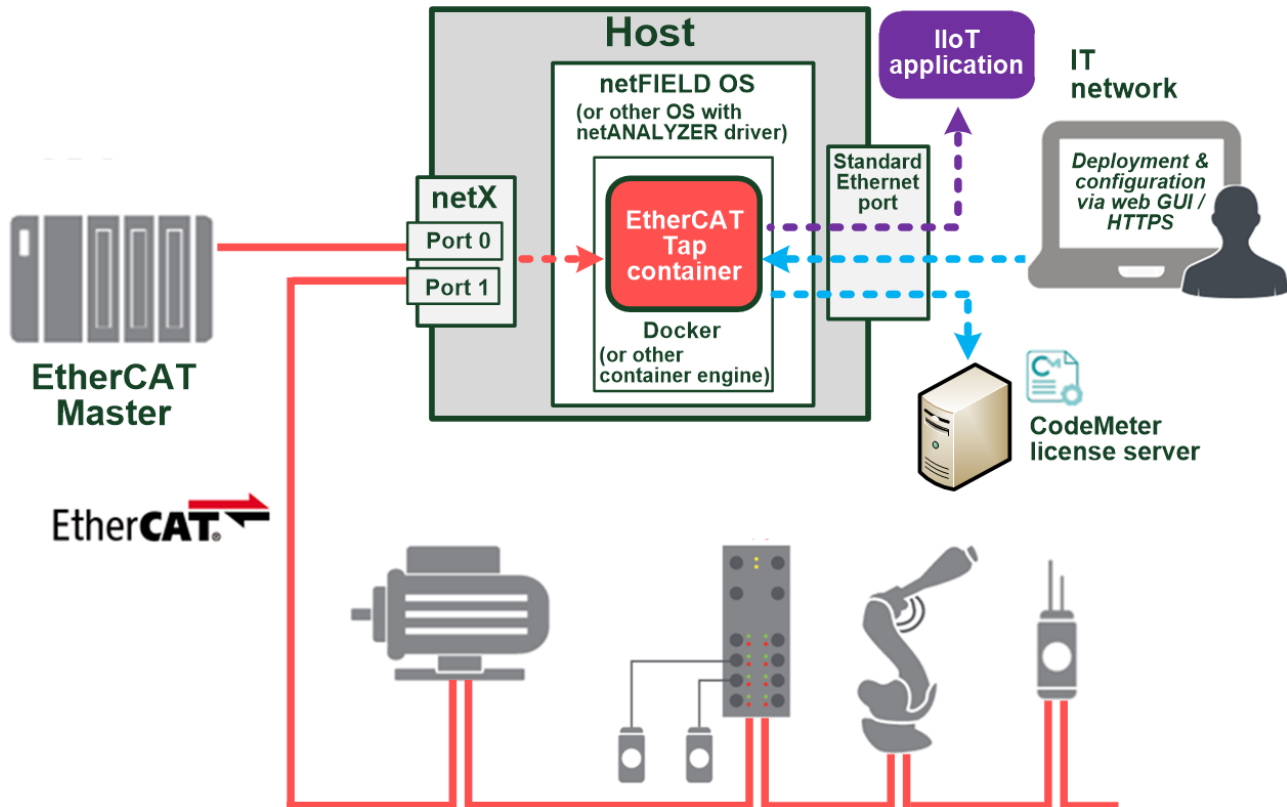


Figure 6: netX and LAN without netMIRROR

### Automatic frame direction detection versus fixed frame direction setting

When using a Software TAP, we recommend you to keep the **Direction detection** option of the EtherCAT Tap on its default setting, which is `Automatic direction detection`.

The **Direction detection** option can be found in the configuration GUI of the EtherCAT Tap under: **netFIELD App EtherCAT Tap > EtherCAT Configuration > General > Direction detection**. (Select the **EtherCAT Configuration** item in the navigation tree to display the **Direction detection** option.)

However, in some rare cases, the EtherCAT Tap might not be able to detect the direction of some frames properly. This is indicated by the “unable to detect” message for the **Detected direction at port 0 / 1** parameter in the **Status** tab of the EtherCAT Tap configuration GUI (see section “*Status*” tab [▶ page 17]). Missing data points on the MQTT broker might also be an indicator of failing frame direction detection.

If direction detection fails, you can set a `Fixed frame direction` depending on how you have connected the ports of the netX-based interface to your EtherCAT network.

If you have connected port 0 to the Slaves, select the Fixed frame direction (Port 0: Input, Port 1: Output) option:

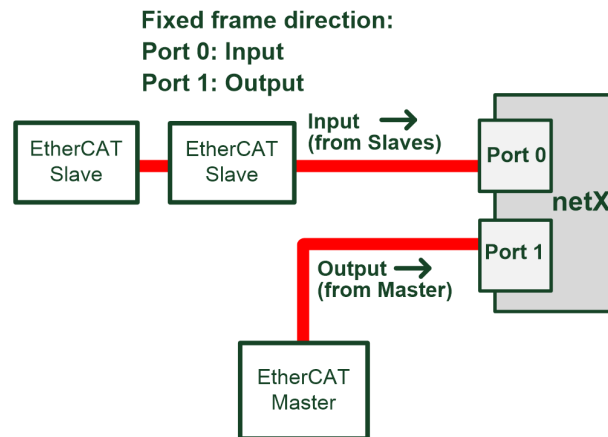


Figure 7: Input at port 0

If you have connected port 0 to the Master, select the Fixed frame direction (Port 0: Output, Port 1: Input) option:

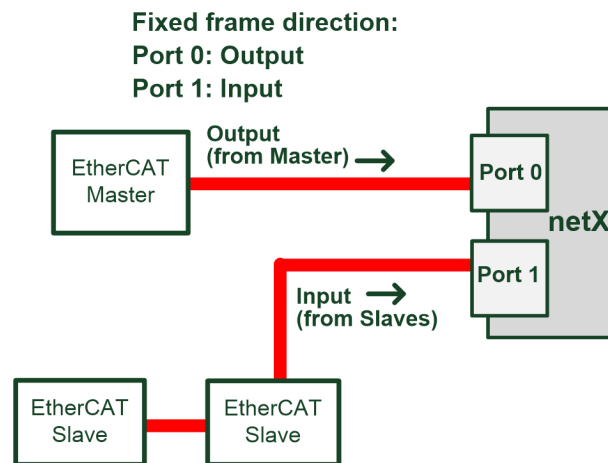


Figure 8: Output at port 0

### 3 Start parameters of the container

The start-up parameters and yaml files for the different use cases are provided in the Hilscher Knowledge base.

- Go to the [netFIELD App EtherCAT Tap](https://kb.hilscher.com/x/LCiJBw) page (https://kb.hilscher.com/x/LCiJBw).
- On the *netFIELD App EtherCAT Tap* page, click on the link under **Current release**.
- On the current release page, you will find the start-up parameters and links to the yaml files under the **Release Notes**.

## 4 Configuration web pages of the EtherCAT Tap

### 4.1 Overview

The EtherCAT Tap container provides a configuration GUI that can be accessed via port 80 of the container. Note that you have to make sure that this port can be accessed from the outside via web browser. We recommend you to use a reverse proxy or API gateway with HTTPS and authentication support for this. For testing purposes, you can temporarily expose the port via Docker port expose.

**Note:**

If the container is deployed on a host with a netFIELD Operating System (netFIELD OS), the configuration web GUI is automatically plugged-into the **Local Device Manager** of the netFIELD OS, where it can be accessed via the **netFIELD App EtherCAT Tap** entry in the navigation panel (side bar) of the Local Device Manager.

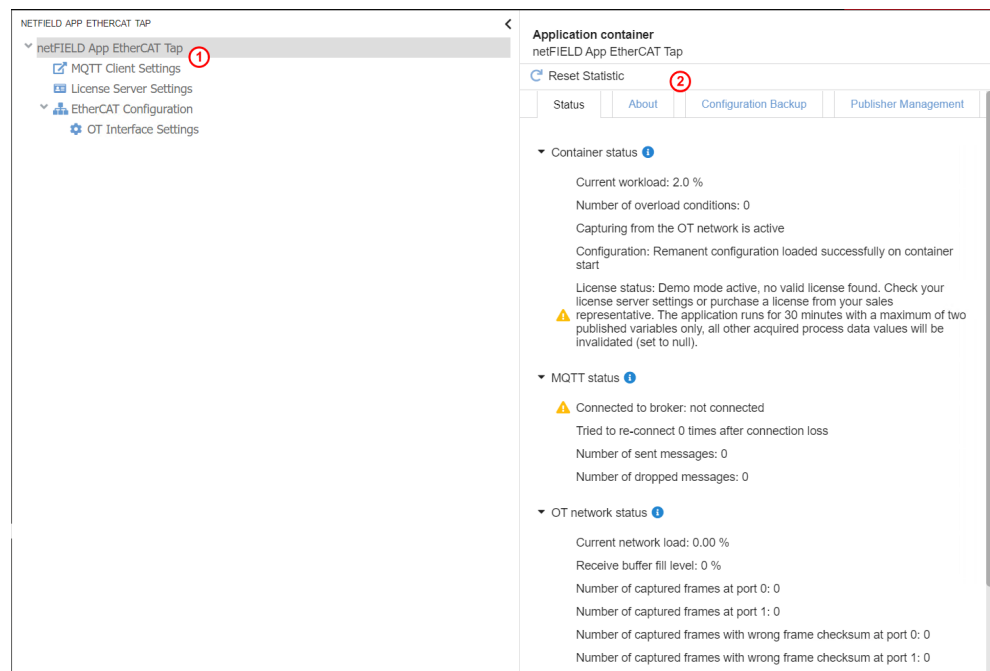


Figure 9: Configuration GUI of EtherCAT Tap

You can navigate through the configuration options by selecting an item in the navigation tree (1). Further parameters/configuration options can be selected in the tabs in the editor window (2) on the right side of the screen.




## 4.2 “Status” tab

The **Status** tab shows information about the operating state of the EtherCAT Tap, of the MQTT connection and of the EtherCAT network traffic.

To open the **Status** tab, first select the **netFIELD App EtherCAT Tap** root element in the navigation tree, then the **Status** tab in the Editor window.

Hovering over an entry will display extra information in a tooltip.

To reset the statistics data to zero (such as e.g. frame counters), click  **Reset Statistics** button in the header.

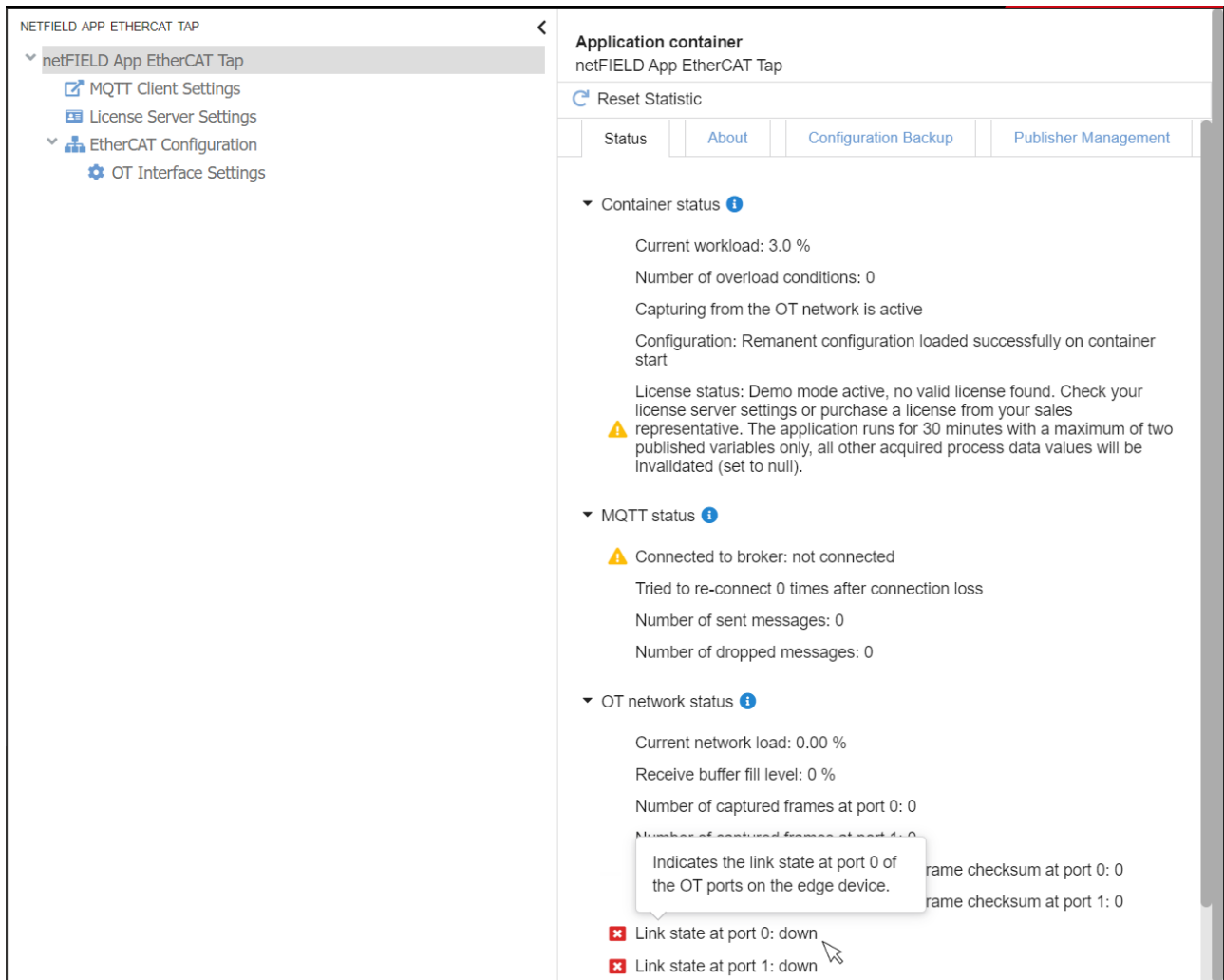


Figure 10: Status tab



**Note:**

Under **OT network status**, check whether the **Link state at port 0/1** is in state “up”. If a port link is in state down”, check whether you have connected the ports of the capture interface according to the instructions provided in the *Use cases* [▶ page 7] chapter.

## 4.3 “About” tab

The **About** tab shows general information about the container.

To open the **About** tab, first select the **netFIELD App EtherCAT Tap** root element in the navigation tree, then the **About** tab in the Editor window.

The screenshot shows the configuration interface for the 'netFIELD App EtherCAT Tap'. The left sidebar contains a navigation tree with the following items: 'netFIELD App EtherCAT Tap' (selected), 'MQTT Client Settings', 'License Server Settings', and 'EtherCAT Configuration'. The main content area is titled 'Application container' and shows details for 'netFIELD App EtherCAT Tap'. The details are organized into sections: 'Name' (netFIELD App EtherCAT Tap [Container ID: netfield-app-ethercat-tap]), 'Version' (1.2.0-rc1), 'API Version' (1), 'Description' (The netFIELD App EtherCAT Tap allows extracting machinery process data simply by monitoring the existing network traffic. There is no need to change PLC or machine configuration. It works with any EtherCAT network, independent of PLC or device vendor.), 'Dependencies' (MQTT broker), 'Vendor' (Hilscher Gesellschaft für Systemautomation mbH), 'Licenses' (HILSCHER netIoT Source Code LICENSE AGREEMENT), and 'Disclaimer' (See <https://netfield.io/termsOfUse>).

Figure 11: About tab

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

Table 2: About tab

## 4.4 “Configuration Backup” tab

In the **Configuration Backup** tab, you can save the EtherCAT Tap configuration settings to your device via your web browser’s download function. You can also restore a formerly saved configuration by uploading the configuration file via web browser.

To open the **Configuration Backup** tab, first select the **netFIELD App EtherCAT Tap** root element in the navigation tree, then the **Configuration Backup** tab in the Editor window.

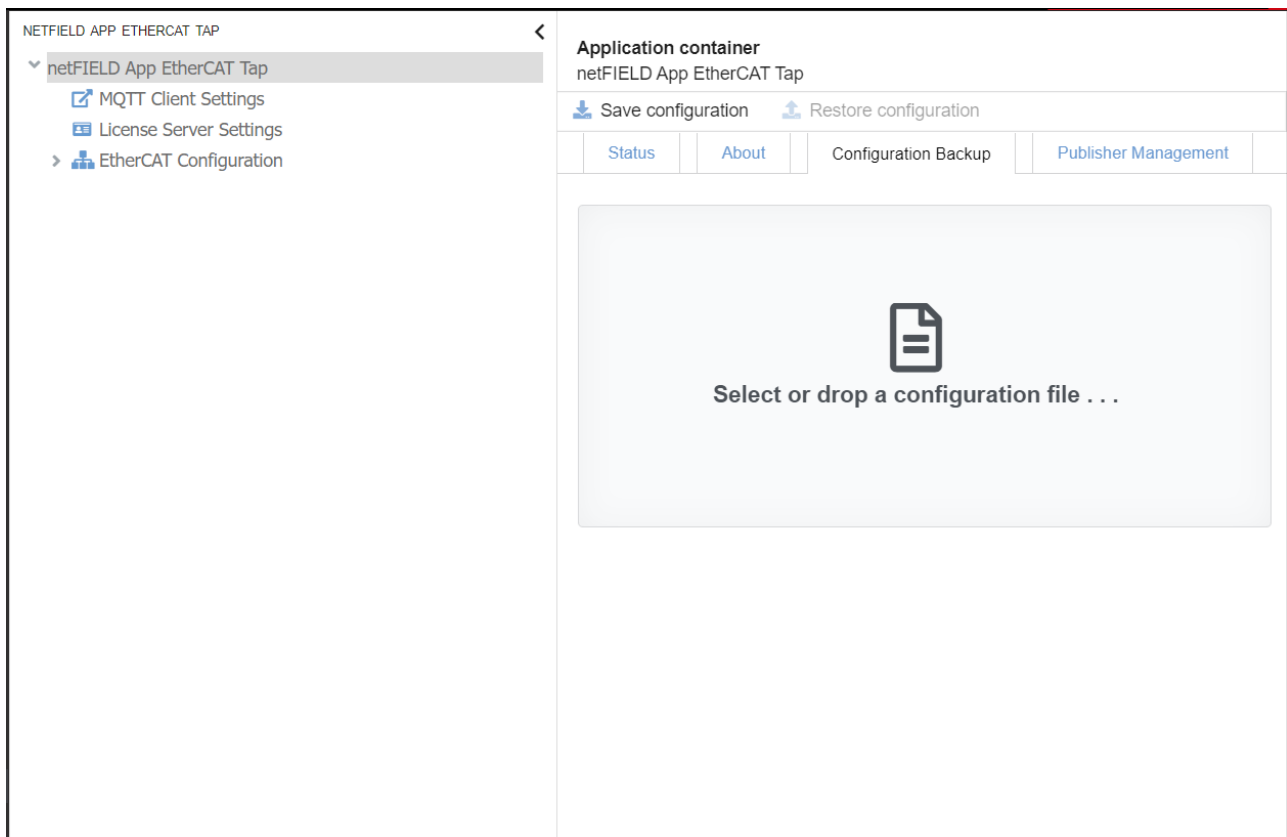


Figure 12: Configuration Backup

### Save configuration

- To save your current configuration, click **Save configuration** button.
- The configuration settings are saved to your device as ZIP file. (The download path depends on the settings of your web browser.)  
The name of the ZIP file is made up by Hardware ID, Container ID and date/time of the download.

### Restore configuration

To restore a formerly saved configuration, 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 **Restore configuration** button is enabled, and you can now “load” the configuration by clicking the button.



**Important:**

The **Restore configuration** function will overwrite the current configuration settings. We recommend you to save your current configuration before using this function.

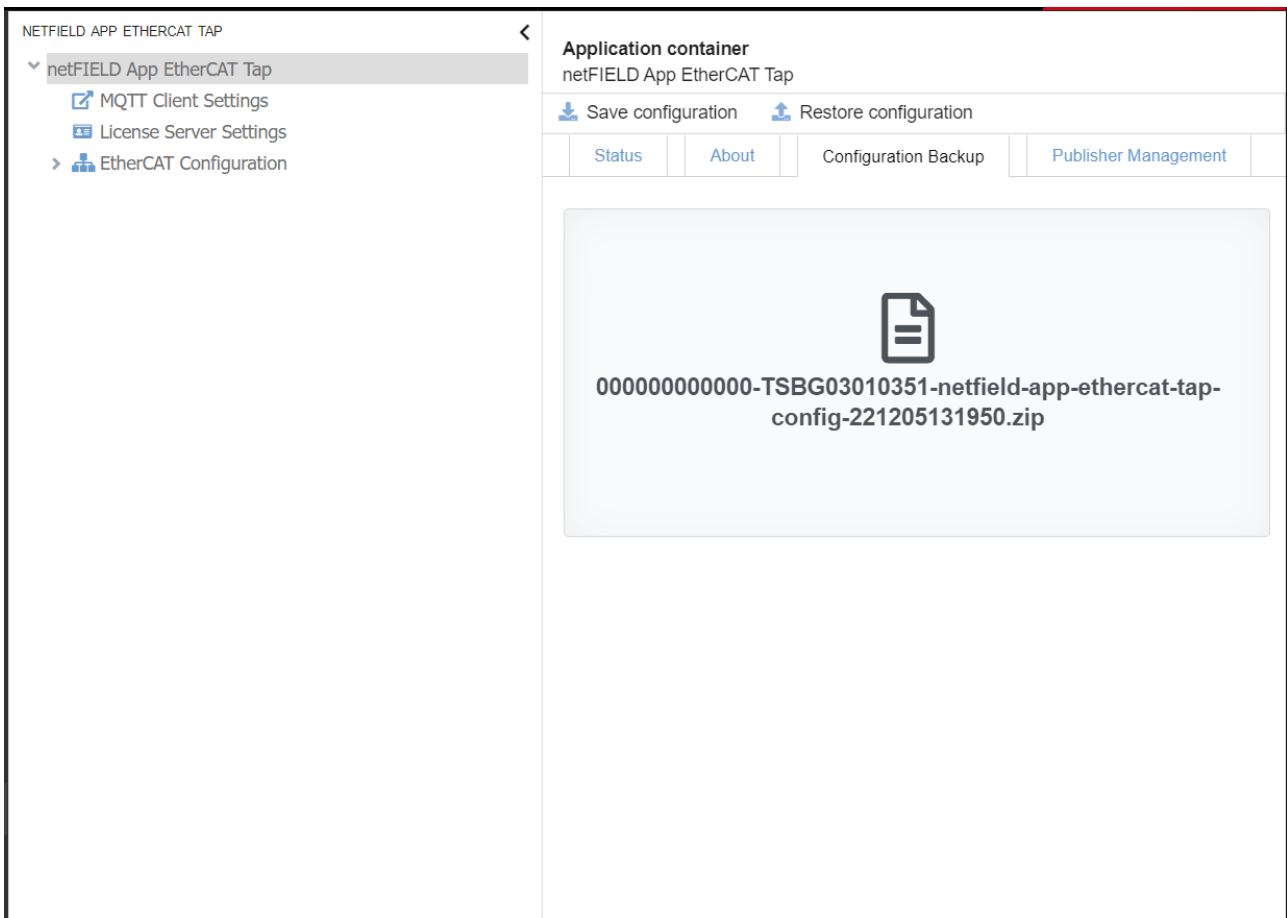


Figure 13: Restore configuration

## 4.5 “Publisher Management” tab

The **Publisher Management** tab shows the variables that the EtherCAT Tap container publishes to the MQTT Broker.

Note that you cannot define here which variables are to be published. For publishing a variable, you have to select it in the **EtherCAT Configuration** tree and then choose the **Publish data** option for it in the EtherCAT variable editor.

You can, however, delete a variable here if you do not want to publish it any longer.

To open the **Publisher Management** tab, first select the **netFIELD App EtherCAT Tap** root element in the navigation tree, then the **Publisher Management** tab in the Editor window.

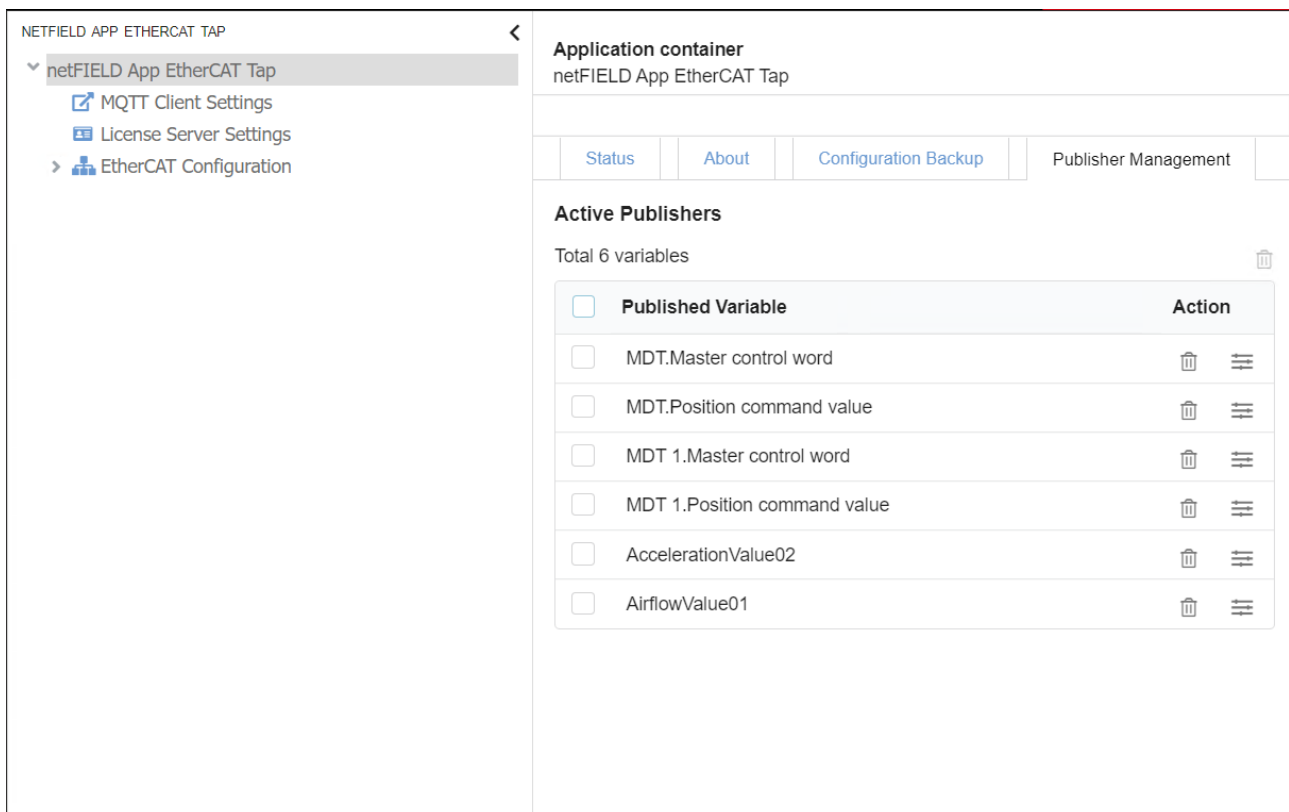


Figure 14: Publisher Management






Element	Description
Active Publishers	Shows the number of variables that are being published.
Published Variable	Name of the variable that is being published. You can select one or multiple variables by clicking the checkboxes, and then delete the selected variables at once by clicking the  button. <b>Note:</b> The order of the published variables in the list is determined by the chronological order in which they were selected for publishing in the EtherCAT Variable editor.
Action	 Deletes the variable
	 Shows details of the variable (see table below)
	If the list contains more than ten entries, you can scroll here to display the next ten items

Table 3: Elements of Publisher Management tab



**Note:**


The maximum allowed size of a JSON message is 100 kB. When this size is reached, the message is sent immediately, disregarding the MQTT publishing interval configured for this variable (i.e. even if the set time interval has not yet been reached).

- Click the  button to open a window showing details of the variable like its MQTT Topic string and other configuration parameters:

**Variable**

○ EtherCAT Master (EtherCAT) / +FRC Air Flow (EP3174-0002) / Inputs / AirflowValue01

---

**Topic:** netfield/000000000000-TSBG03010351/netfield-app-ethercat-tap/EtherCAT Master (EtherCAT)/FRC Air Flow (EP3174-0002)/Inputs/Airflow.Value01/638 

▼ General

Data type	Signed 16 Bit Integer
Bit length	16
Byte order	Not swapped
Physical address offset	4482
Physical bit offset	0
Logical address	Not used

▼ Standardization


Standardization type	Keep original value and do not standardize
----------------------	--

▼ Publisher data

Quality of Service	QoS0
Data sampling	take data directly as acquired from data source
Publish interval	2 s
Retained	false

Ok

Figure 15: Parameters of EtherCAT variable

Element	Description
○	The first line shows the path to the variable (for unique identification) in the EtherCAT network.
<b>Topic</b>	Shows the MQTT topic string under which the variable is published. By default, the EtherCAT Tap uses the following string: <code>netFIELD/&lt;gateway prefix&gt;/&lt;container id&gt;/&lt;master name&gt;/&lt;slave name&gt;/&lt;direction name&gt;/&lt;variable name&gt;/&lt;node id&gt;</code>  If you do not want to use the default MQTT Topic string, you can define your own string when you select the variable in the EtherCAT Configuration tree for publishing ( <b>EtherCAT Variable</b> > <b>Publish data</b> option > <b>Topic</b> field). Note that you are free to define the topic hierarchy and name according to your individual needs there. You are only restricted not to use the # and + characters and not to use \$ as very first character.
	Click this button to copy the topic to your clipboard, e.g. for subscribing to it in your MQTT client.
<b>General</b>	Shows data formats of the variable.
<b>Standardization</b>	Shows standardization settings for the variable.

Element	Description
<b>Publisher data</b>	Shows the MQTT publication settings for the variable.
<b>Ok</b>	Click this button to close the window.

*Table 4: Elements Variable's details window*

## 4.6 MQTT Client Settings

On the **MQTT Client Settings** page, you can customize the MQTT client settings of the EtherCAT Tap container.



**Note:**

If you use the EtherCAT Tap container on a host with a netFIELD Operating System (netFIELD OS), this page features the **Use general settings** option, allowing you to use the standard MQTT client settings of the netFIELD OS.

These standard MQTT client settings can be viewed (and changed) in the Local Device Manager of the netFIELD OS under **General Settings > Default MQTT Client Settings**.

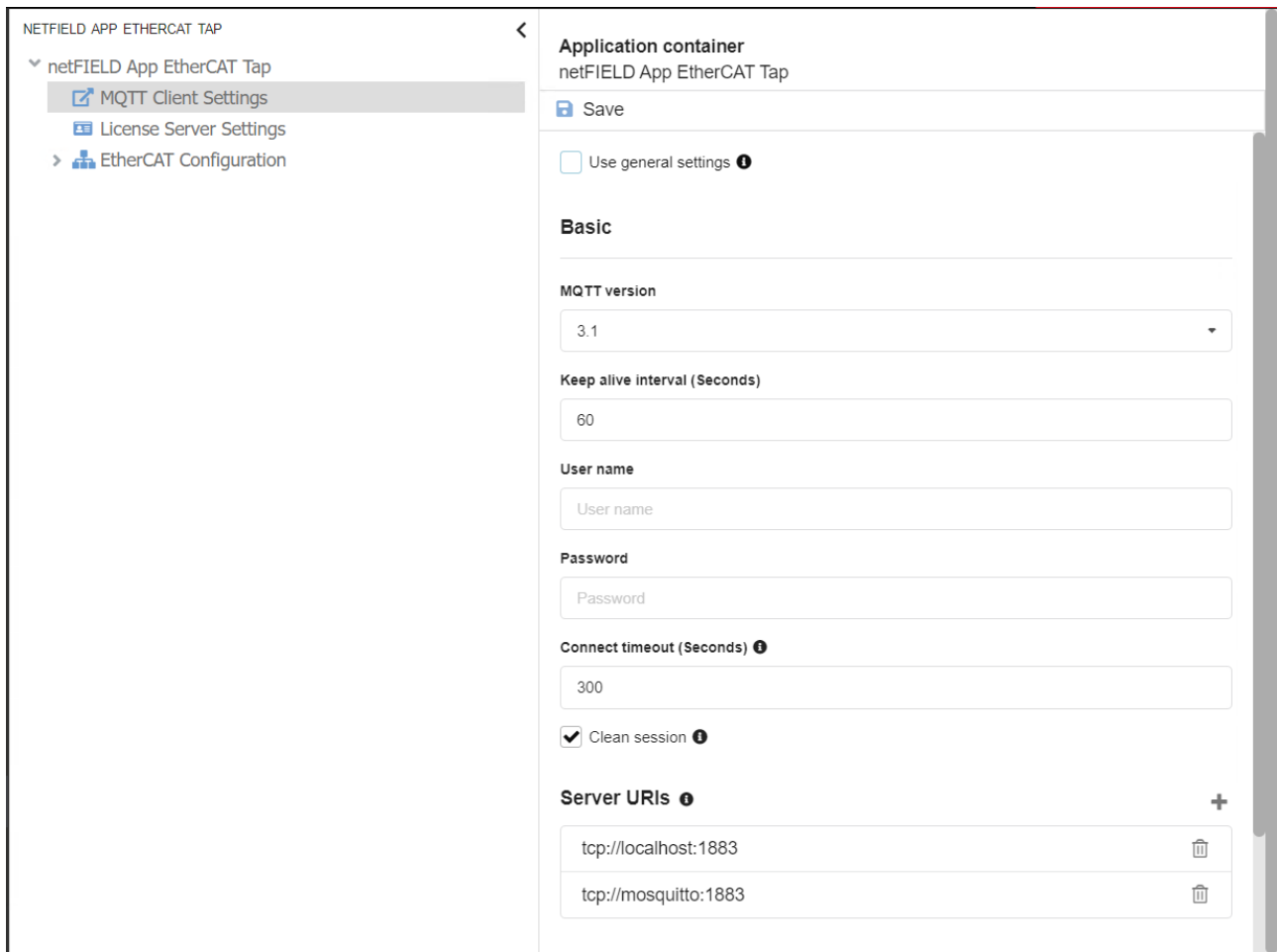





Figure 16: MQTT Client settings



Element		Description	
Basic	MQTT version	MQTT version to be used (depending on the MQTT broker).	
	Keep alive interval (Seconds)	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).	
	Password	Password for authentication at the broker (if implemented and required by the broker).	
	Connect timeout (Seconds)	Defines the maximum length of time in seconds that is allowed for completing the connection process.	
	Clean session	If <b>Clean session</b> 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 <b>Clean session</b> is unchecked, the broker creates a persistent session for the client.	
	Server URIs	Server URI of the MQTT broker. <b>Note:</b> 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.	
		Click this button to add a new server URI.	
		Click this button to delete an existing server URI.	
Last Will and Testament	Use last will and testament options	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.) This opens the following fields for configuring the LWT parameters:	
		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”

Element		Description	
SSL / TLS	Use SSL / TLS	Select this option if you want to use SSL/TLS encryption for creating a secure connection to the MQTT broker. <b>Note:</b> This option is for expert users only! If the MQTT broker and the EtherCAT Tap are running on the same device, a secure SSL/TLS connection is not necessary (the overhead of the secure connection can thus be avoided). Selecting this option opens the following fields for configuring the SSL / TLS parameters:	
		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, EtherCAT Tap will also accept invalid certificates from the broker (not recommended).

Table 5: MQTT Client Settings

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



**Note:**

After changing the settings – e.g. by defining a new MQTT server (broker) URI – you can check the state of the new MQTT connection on the **Status** tab (see section “*Status*” tab [▶ page 17]). For information on the structure of MQTT messages, see section *MQTT message format* [▶ page 45].

## 4.7 License Server Settings

On the **License Server Settings** page, you must specify the IP address or host name (“endpoint”) of your local license server.



### Note:

You can specify more than one server endpoint here if multiple local license servers are available, e.g. as backup servers.

The app will try to retrieve a valid license from the first server endpoint in the list. If the first server is not available or if there is no valid license on this server, the app will automatically try the next one, and so forth.

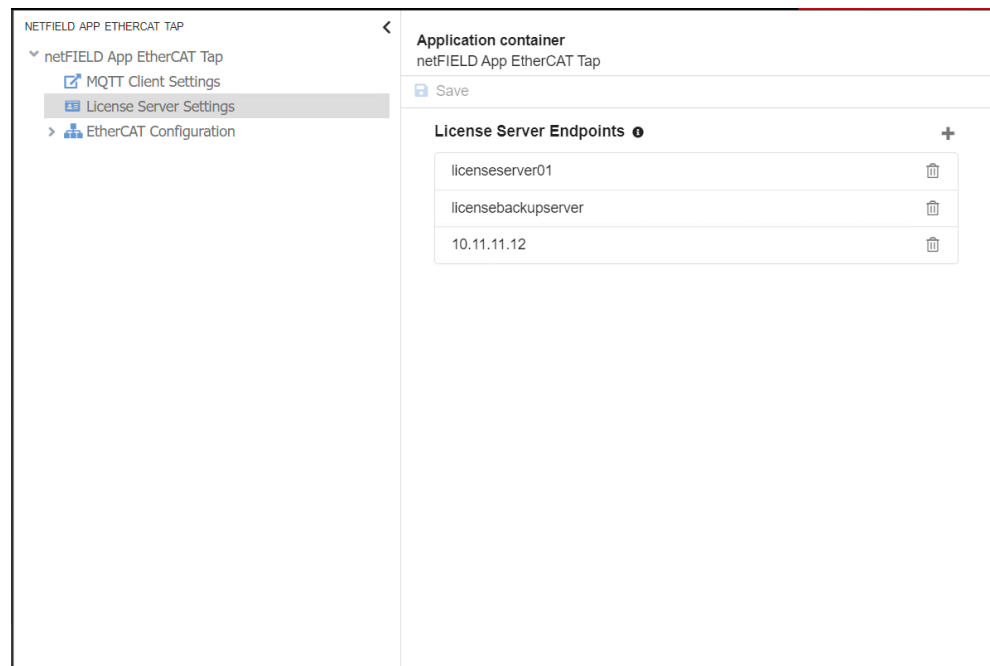


Figure 17: License Server Settings

1. Specify license server.
  - To specify a license server, click **+** button.
  - In the **License Server Endpoints** field, enter the IP address or the hostname of the machine on which your license server is running.
  - Click **Save** button.

- After the **Succeeded to save license server settings!** message appears, the EtherCAT Tap immediately tries to connect to the server and retrieve its license.

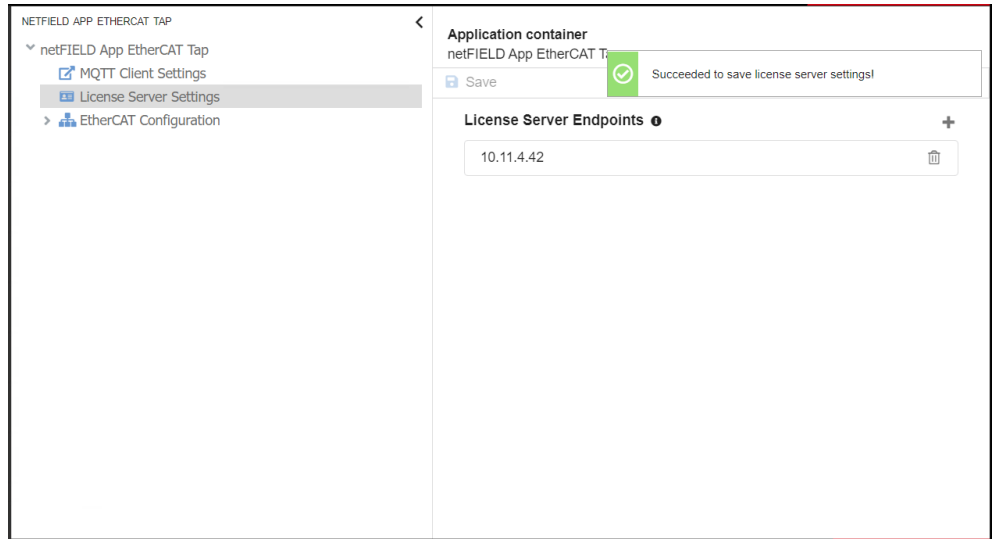


Figure 18: License Server Settings saved

2. Check the license state on the **Status** page.
  - To open the **Status** tab, first select the **netFIELD App EtherCAT Tap** root element in the navigation tree, then the **Status** tab in the Editor window.
  - Under **Container status**, check the license state:

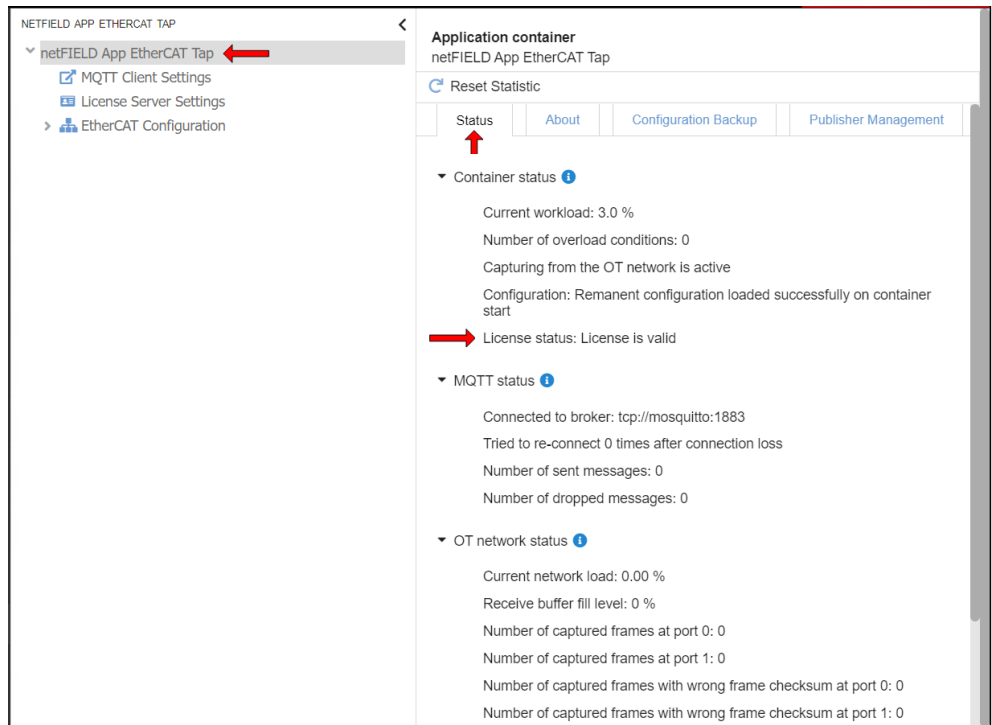


Figure 19: Check license status

## 4.8 EtherCAT configuration

### 4.8.1 OT Interface settings

On the **OT Interface settings** page, you can configure the capture interface that you are using to connect your device (respectively your “tap”) to the physical EtherCAT network.

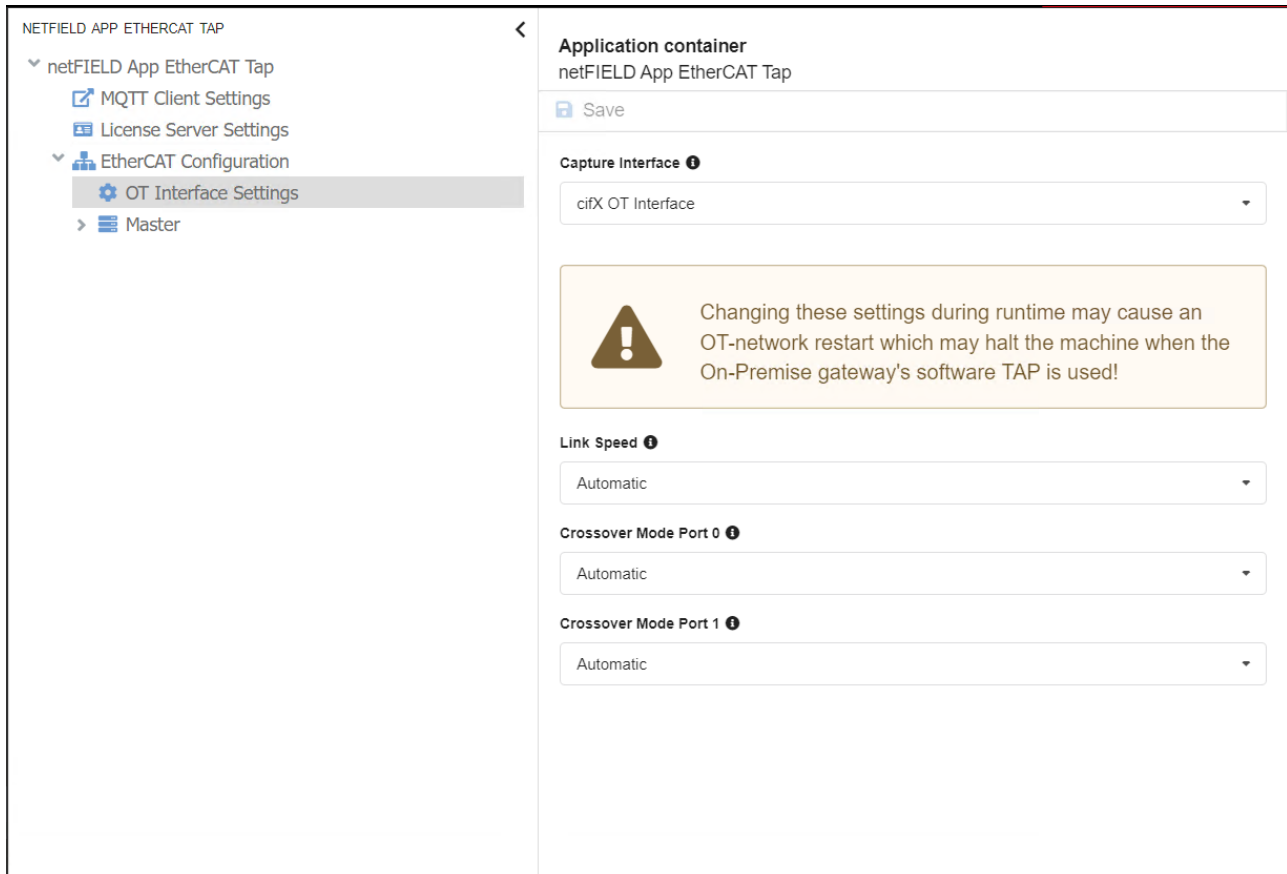


Figure 20: OT Interface Settings


Parameter/Element		Description
 <b>Save</b>		Click this button to save your new settings.
Capture Interface	cifX OT Interface	Preset to cifX OT interface. Cannot be changed. <b>Note:</b> cifX is a netX-based Industrial Ethernet interface card, which is e.g. included in the netFIELD OnPremise edge gateway.
Link Speed	Automatic	Automatic synchronization. You must use this setting if you have connected the netX-based interface <i>directly</i> to the EtherCAT network; i.e. if you are using a “software tap”, not a “hardware tap” (see section <i>Connecting EtherCAT network as software TAP</i> [▶ page 12]).  <i>Automatic</i> is also the recommended default setting if you are using a “hardware tap” like the netMIRROR. Note, however, that when using automatic link speed negotiation in combination with a netMIRROR TAP, the EtherCAT Tap (due to a little time delay caused by the negotiation process) might miss some of the first frames (and thus some of the configuration data) that are exchanged on the bus during the EtherCAT startup phase.
	10 MBit/s fixed	Use this setting if you are using a “hardware TAP” like e.g. netMIRROR (see section <i>Connecting EtherCAT network via netMIRROR (hardware TAP)</i> [▶ page 8]) and the EtherCAT network operates at 10 Mbit/s. <b>Note:</b> Rarely used.
	100 MBit/s fixed	Use this setting if you are using a “hardware TAP” like e.g. netMIRROR and the EtherCAT network operates at 100 Mbit/s.
Crossover Mode Port 0	Automatic	Use this setting if you have connected the netX-based interface <i>directly</i> to the EtherCAT network (i.e. if you are using a “software tap”, not a “hardware tap”) and have set the <b>Link Speed</b> to <i>Automatic</i> . (See section <i>Connecting EtherCAT network as software TAP</i> [▶ page 12]).
	No Crossover	Use this setting if you are using a “hardware tap” like e.g. netMIRROR and: <ul style="list-style-type: none"> <li>• If port 0 is connected via “patch” cable to the MIRROR OUT port B</li> <li>• If port 0 is connected via “crossover” cable to the MIRROR OUT port A</li> </ul> (For clarification see also section <i>Connecting EtherCAT network via netMIRROR (hardware TAP)</i> [▶ page 8])
	Crossover	Use this setting if you are using a “hardware tap” like e.g. netMIRROR and: <ul style="list-style-type: none"> <li>• If port 0 is connected via “crossover” cable to the MIRROR OUT port B</li> <li>• If port 0 is connected via “patch” cable to the MIRROR OUT port A</li> </ul> (For clarification see also section <i>Connecting EtherCAT network via netMIRROR (hardware TAP)</i> [▶ page 8])
Crossover Mode Port 1	Automatic	Use this setting if you have connected the netX-based interface <i>directly</i> to the EtherCAT network (i.e. if you are using a “software tap”, not a “hardware tap”) and have set the <b>Link Speed</b> to <i>Automatic</i> . (See section <i>Connecting EtherCAT network as software TAP</i> [▶ page 12]).
	No Crossover	Use this setting if you are using a “hardware tap” like e.g. netMIRROR and: <ul style="list-style-type: none"> <li>• If port 1 is connected via “patch” cable to the MIRROR OUT port B</li> <li>• If port 1 is connected via “crossover” cable to the MIRROR OUT port A</li> </ul> (For clarification see section <i>Connecting EtherCAT network via netMIRROR (hardware TAP)</i> [▶ page 8])
	Crossover	Use this setting if you are using a “hardware tap” like e.g. netMIRROR and: <ul style="list-style-type: none"> <li>• If port 1 is connected via “crossover” cable to the MIRROR OUT port B</li> <li>• If port 1 is connected via “patch” cable to the MIRROR OUT port A</li> </ul> (For clarification see section <i>Connecting EtherCAT network via netMIRROR (hardware TAP)</i> [▶ page 8])

Table 6: Parameters OT Interface Settings

## 4.8.2 EtherCAT Configuration node

In order to capture process data and re-publish it via MQTT, the EtherCAT Tap must get “acquainted” with the participants, data points and certain parameters of your actual EtherCAT network.

There are two ways to provide the necessary information for the Tap:

- Automatically by importing the ENI (EtherCAT Network Information) file from TwinCAT (recommended).
- Manually by “rebuilding” the structure of your network by adding the Master and the Slaves to the **EtherCAT Configuration** tree, and then configure their parameters and variables “by hand”.

The structure of your network and the data points (that the Tap knows of) is represented in the navigation tree under the **EtherCAT Configuration** element:

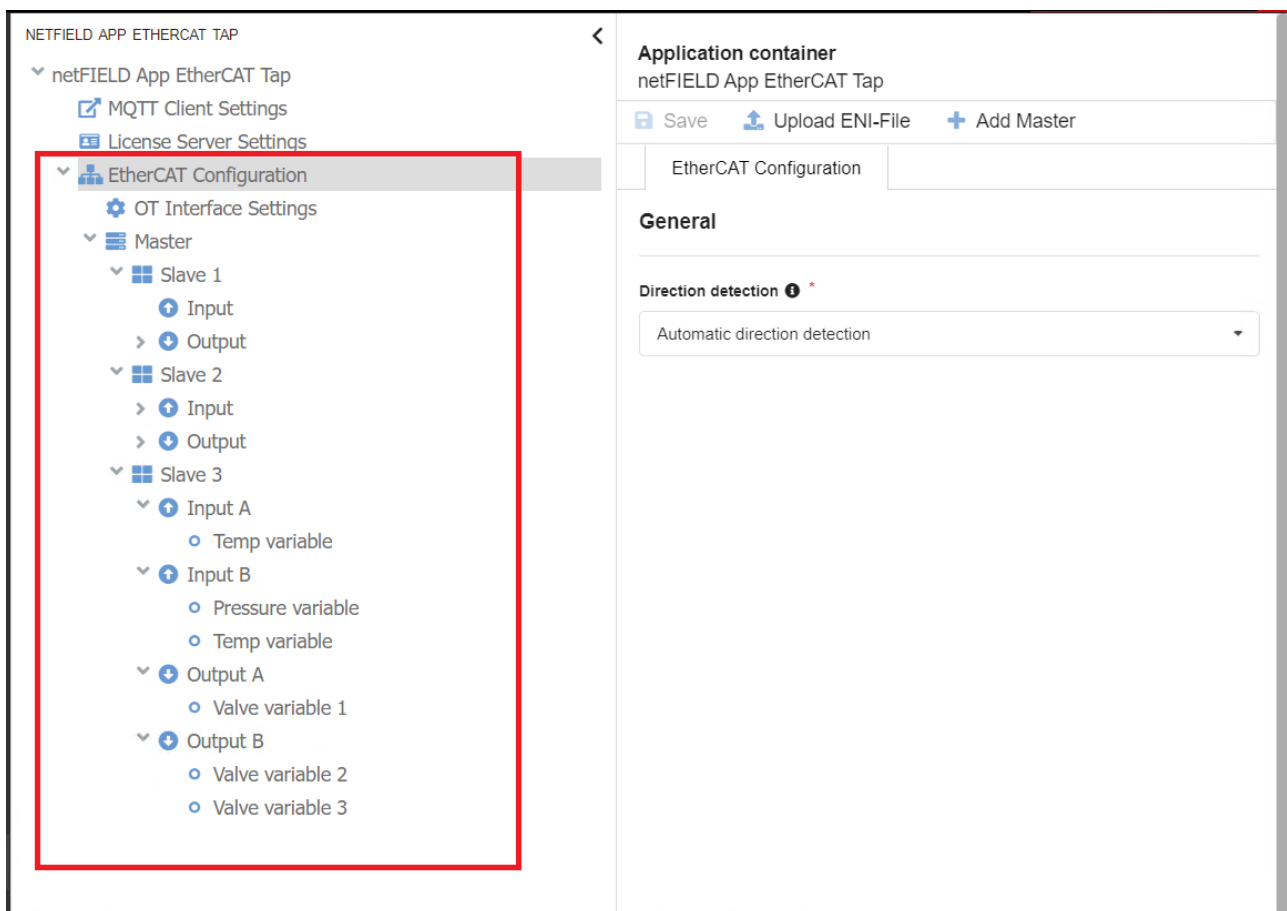


Figure 21: EtherCAT Configuration tree

An arrow symbol **>** in front of an element in the EtherCAT navigation tree indicates that the element contains further subordinate elements, like e.g. inputs/outputs of a Slave.

Clicking on an element will display the subordinate elements in the navigation tree.

Selecting an element will also display its parameters in the Editor window on the right side of the screen, where they can be changed if necessary:

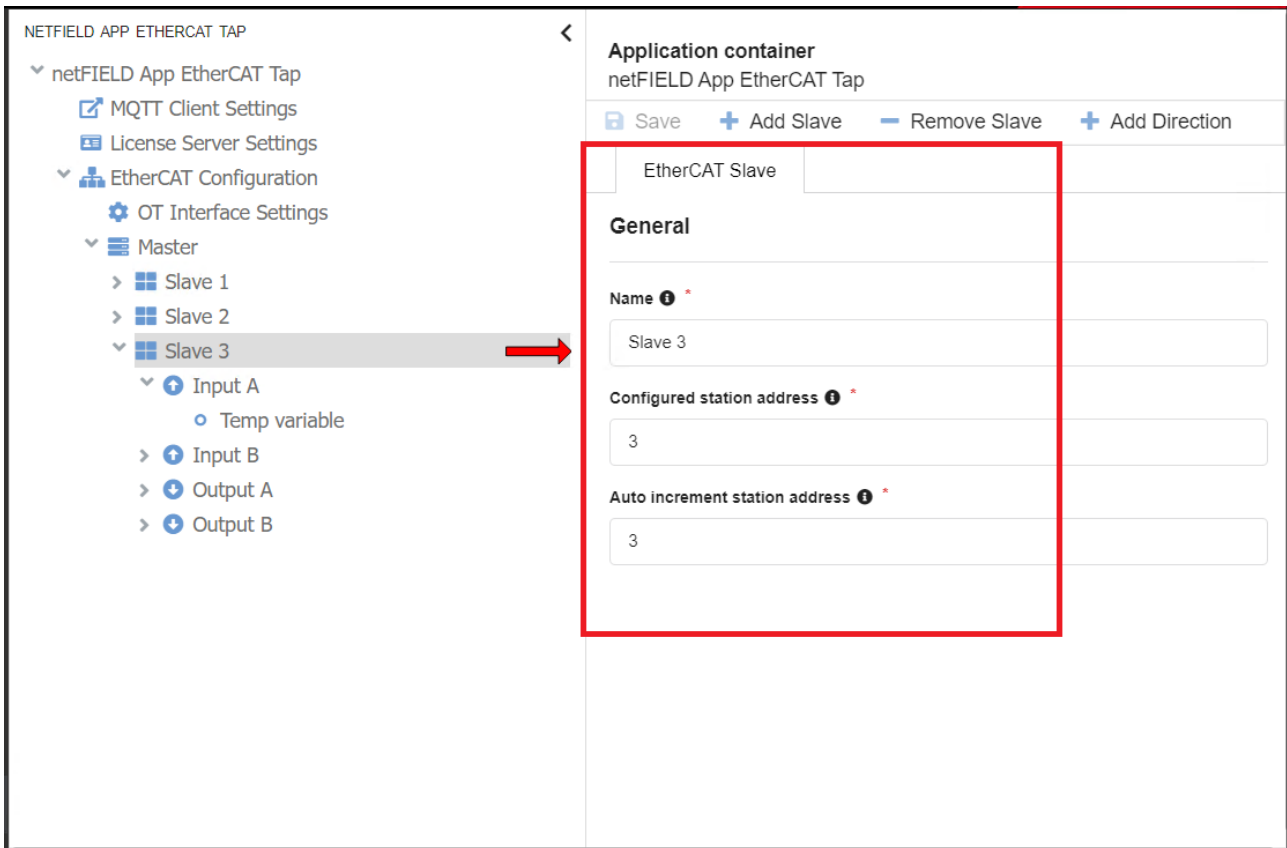


Figure 22: Parameters of selected EtherCAT element in Editor window



The options in the menu bar on top of the Editor window are context sensitive and depend on the element that you have selected in the EtherCAT navigation tree:

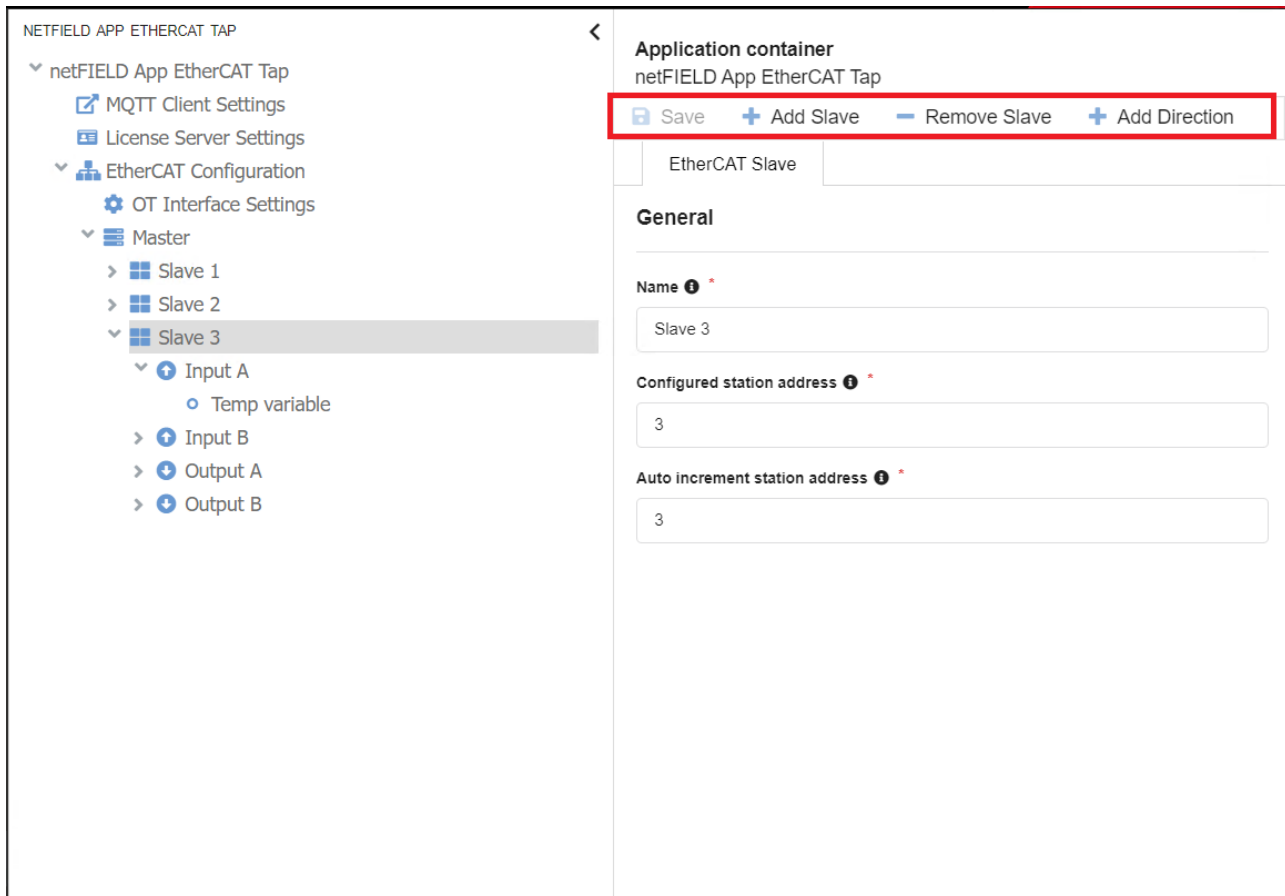















Figure 23: Menu bar

In the menu bar, you can save changes that you have made to the parameters of the selected element, or you can delete the selected element altogether. You can also add further elements on the same level (e.g. another Slave next to a selected Slave) or new elements on the subordinate level (e.g. a variable below a selected input/output).

The following table describes the options and parameters of the elements belonging to your EtherCAT Configuration. Click on an element in the EtherCAT Configuration tree to see the parameters and menu options:

Icon	Element	Parameter	Options in menu
	EtherCAT	Direction detection	Select here either auto-detection mode or set the frame direction manually.
		Automatic direction detection	The EtherCAT Tap tries to detect the frame direction automatically. This is the recommended default setting
		Fixed frame direction (Port 0: Input, Port 1: Output)	Select this option if you are using a "Software TAP" and you have connected the Slaves to port 0
		Fixed frame direction (Port 0: Output, Port 1: Input)	Select this option if you are using a "Software TAP" and you have connected the Master to port 0
	Master	Name	Name of the EtherCAT Master. Can be freely defined by the user, but must not be empty (mandatory field).
	Slave	Name	Name of the Slave device. Can be freely defined by the user, but must not be empty (mandatory field).
		Configured station address	Station address according to the EtherCAT configuration.
		Auto increment station address	Auto-incremented station address according to the EtherCAT configuration.
 	Direction (Output/ Input)	Name	Name of the EtherCAT data item. Can be freely defined by the user, but must not be empty (mandatory field).
		Direction	IO direction of the data item. Select <b>Output</b> for the data that the Slave receives from the Master in order to forward it to an actuator. An Output is symbolized by the  icon. Select <b>Input</b> for the data that the Slave receives from a sensor in order to forward it to the Master. An Input is symbolized by the  icon.

Icon	Element	Parameter			Options in menu	
	Variable	Name	Name of the variable. Can be freely defined by the user, but must not be empty (mandatory field). This is the name of the EtherCAT Variable that will be published via MQTT. It will also be added to the EtherCAT Variable list in the Publisher Management tab (see section “ <i>Publisher Management</i> ” tab [▶ page 21]). <b>Note:</b> Not to be confused with the MQTT <b>Topic</b> name (see <b>Topic</b> field under <b>Publish data</b> ).			<b>Save:</b> Saves changed parameters of selected variable <b>+ Add Variable:</b> Adds new variable on same level <b>- Remove Variable:</b> Removes selected variable <b>+ Clone Variable:</b> Adds a copy of the selected variable on same level, which you can use as a “template” for editing a new variable
		Data type	Supported types:	Bit length:	Description/ Value range:	
		Boolean	1	'0': FALSE '1': TRUE		
		Signed 8 Bit Integer	8	-128 ... 127		
		Signed 16 Bit Integer	16	-32768 ... 32767		
Signed 32 Bit Integer	32	$-2^{31} \dots +2^{31}-1$				
Signed 64 Bit Integer	64	$-2^{63} \dots +2^{63}-1$				
Unsigned 8 Bit Integer	8	0 ... 255				
Unsigned 16 Bit Integer	16	0 ... 65535				
Unsigned 32 Bit Integer	32	0 ... $+2^{32}-1$				
Unsigned 64 Bit Integer	64	0 ... $+2^{64}-1$				
32 Bit Floating Point	32	Single precision floating point range				
64 Bit Floating Point	64	Double precision floating point range				
Visible String	8*n	UTF-8 / ASCII encoded string				
Octet String	8*n	String containing a sequence of hex encoded octets in format 01ABF7				
Bit length	Number of bits of the variable according to the selected data type. Corresponds by default to the maximum length allowed by the data type. You may specify a lower value. If this bit length is shorter than the above specified data type, the upper bits will be padded. For signed data types, a sign extension will be done.					
Byte order	Specify here if the byte order in a data word shall be swapped byte-wise in regard to the native network order: Swapped Not swapped					

Icon	Element	Parameter		Options in menu	
	Variable	Physical address offset	Physical address of the EtherCAT variable according to the configuration of the machine.		<p> <b>Save</b>: Saves changed parameters of selected variable</p> <p> <b>Add Variable</b>: Adds new variable on same level</p> <p> <b>Remove Variable</b>: Removes selected variable</p> <p> <b>Clone Variable</b>: Adds a copy of the selected variable on same level, which you can use as a "template" for editing a new variable</p>
		Physical bit offset	Number of bits the variable is shifted at the physical address offset. Must be a value between 0 and 7.		
		Enable Logical Address	Select this option if "virtual" logical addresses are used. Selecting this option enables the following fields:		
		Logical address	Enter here the logical byte address of the variable.		
		Logical bit offset	Enter here the logical offset of the variable in bits, relative to the logical byte address.		
		Standardization type	<p>Allows scaling of analog values. You can either <b>Keep original value and do not standardize</b> or select the <b>Scale linearly by factor and offset</b> option.</p> <p>Selecting the <b>Scale linearly...</b> option enables the <b>Factor</b> and <b>Offset</b> fields. The standardization value will be computed according to the formula: <math>raw\ value * factor + offset</math></p> <p><b>Note:</b> Standardization is supported only for integer data types.</p>		



Icon	Element	Parameter	Options in menu
	Variable	Publish data Select this option to publish the variable via MQTT. It will also be added to the <b>EtherCAT Variable</b> list in the <b>Publisher Management</b> tab (see section “ <i>Publisher Management</i> ” tab [▶ page 21]). You can configure the following parameters for publishing the variable:	 <b>Save</b> : Saves changed parameters of selected variable + <b>Add Variable</b> : Adds new variable on same level - <b>Remove Variable</b> : Removes selected variable + <b>Clone Variable</b> : Adds a copy of the selected variable on same level, which you can use as a “template” for editing a new variable
		Topic Enter here the MQTT Topic name string under which the variable will published. If left empty, the Tap uses the default string (see section <i>MQTT message format</i> [▶ page 45]). 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.	
		Quality of service Select the MQTT QoS from the drop-down list. <b>Note</b> : For performance reasons it is recommended to use QoS0 whenever possible. Using QoS1 or QoS2 will increase performance requirements of the application container.	
		Publish interval Define the publish interval in - Milliseconds - Seconds - Minutes - Hours <b>Note</b> : MQTT messages will be published approximately at this interval (may jitter by a few milliseconds). Messages will be sent only when new data values are available. If more than one process data value has been acquired in this time-span, the message will contain an array of process data values. If a message size of 100 kByte is exceeded, the message will be sent even if the interval was not reached. Keep in mind that messages should not grow endlessly in size. Therefore, choose this value 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 data source in that time.	
		Data sampling Define the conditions for the sampling: <ul style="list-style-type: none"> <li>• Only take data if the data source value has changed</li> <li>• Take data directly as acquired from data source:</li> </ul> Note that this option means OT network cycle accurate data sampling and may result in very high amount of data samples in a short time. <ul style="list-style-type: none"> <li>• Sample the acquired data from data source by a fixed sample rate:</li> </ul> Opens the <b>Sample rate in milliseconds</b> field in which you can specify the sample rate (between 1 and 1000 ms)	
Retained “Retained” flag of MQTT message			

Table 7: Elements EtherCAT Configuration

### 4.8.3 How to configure the EtherCAT Tap with ENI file

This section provides step-by-step instructions for creating an EtherCAT Configuration by using an ENI file and for defining a variable to be published via MQTT.

#### Requirements

- The host device on which the EtherCAT Tap is running is powered and connected to your EtherCAT network (see section *Use cases* [▶ page 7]).
- You have exported your EtherCAT engineering project as ENI file from TwinCAT and you have access to the file (please refer to the TwinCAT documentation for information on how to export an ENI file).

#### Step-by-step instructions

1. Set **Direction detection** of the EtherCAT Tap to *Automatic*.
  - In the navigation tree of the EtherCAT Tap, select **EtherCAT Configuration**.

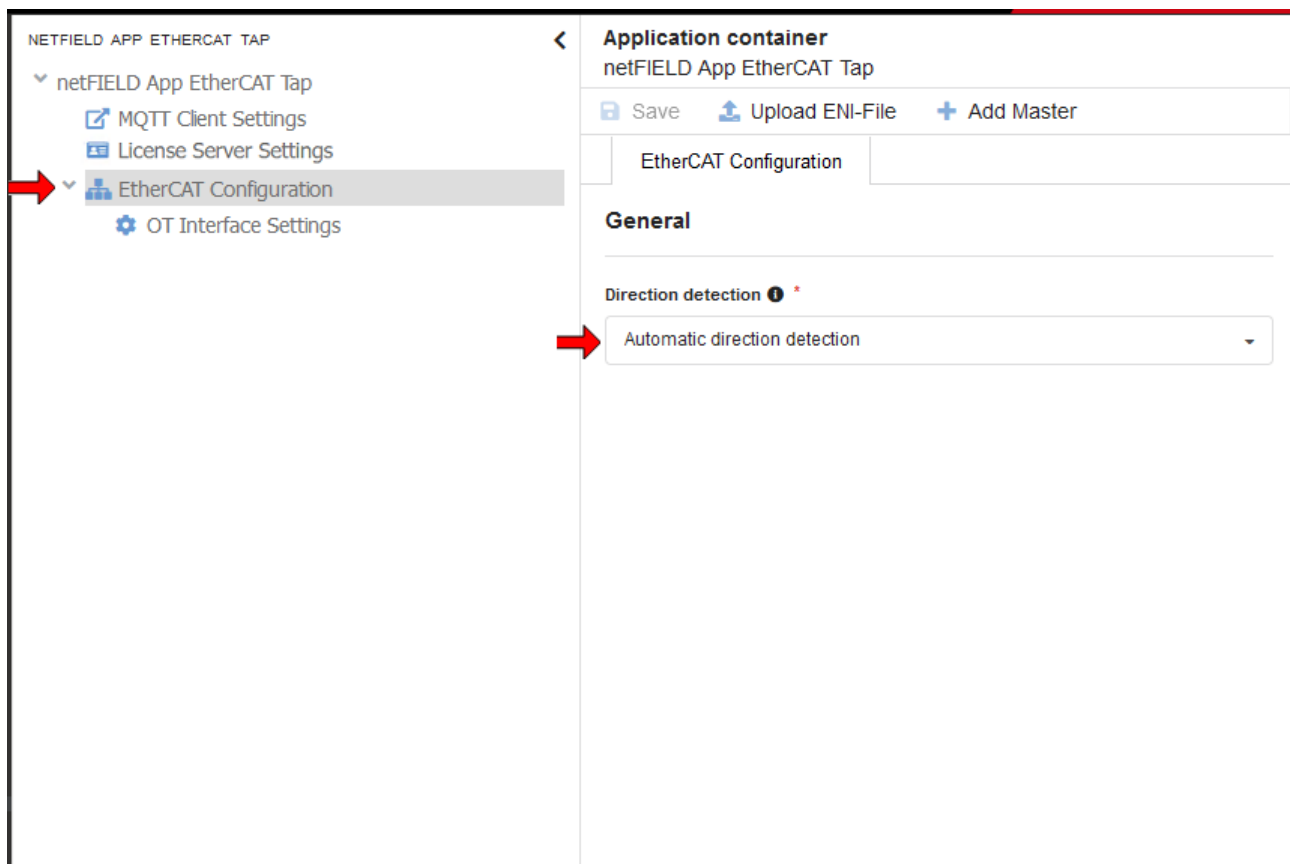


Figure 24: Set direction detection

- In the **Direction detection** dropdown-list, select *Automatic direction detection* (if not set on *Automatic* already), then click **Save** button.

2. Set **OT Interface Settings** of the EtherCAT Tap to *Automatic*.
  - In the navigation tree of the EtherCAT Tap, select **OT Interface Settings**.

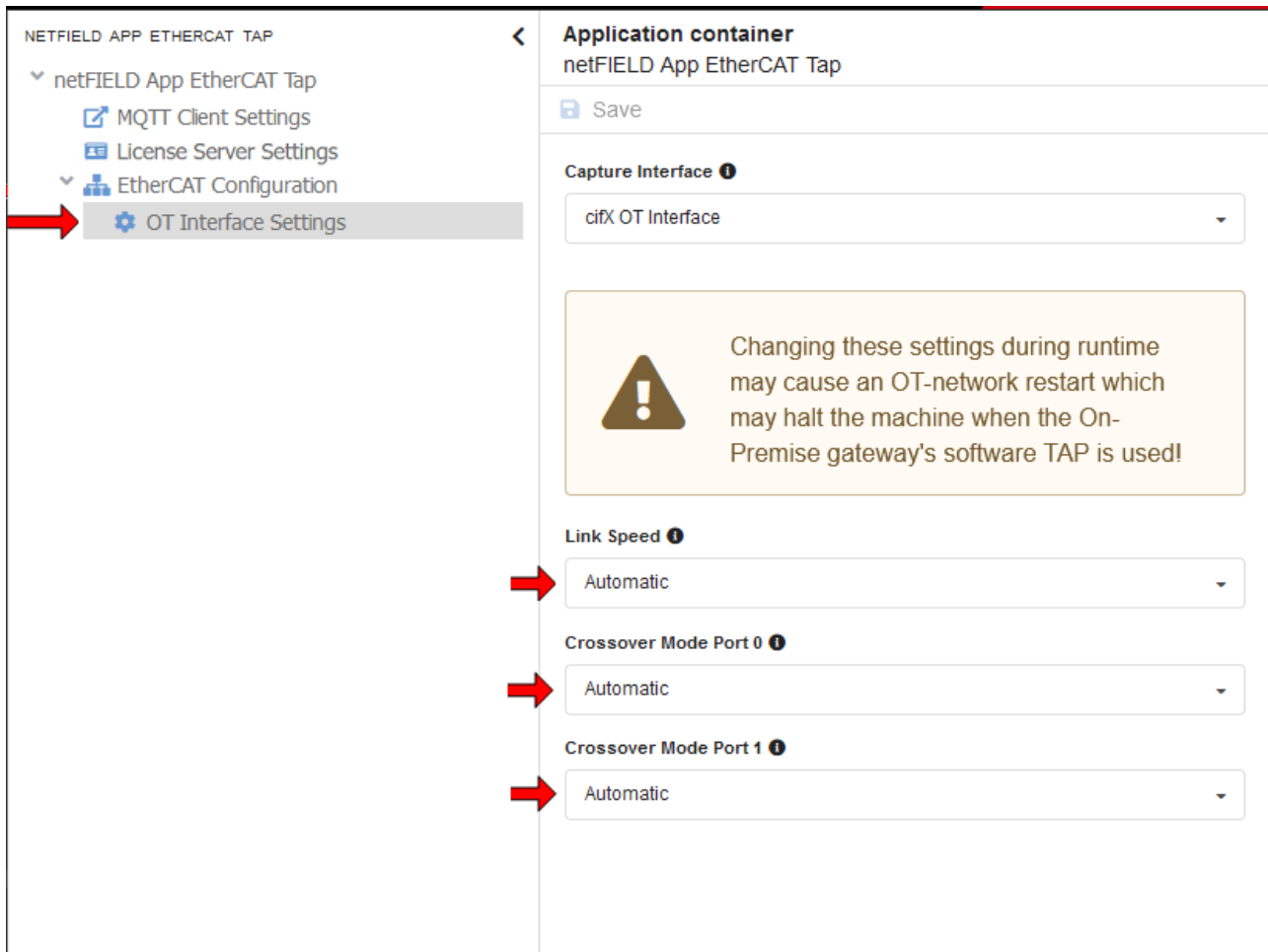



Figure 25: Set port speed

- In the **Link Speed** and **Crossover Mode Port 0/1** dropdown-lists, select *Automatic* (if not set on *Automatic* already), then click **Save** button.

3. Upload ENI file to “build” your EtherCAT Tap configuration.
  - In the navigation tree of the EtherCAT Tap, select **EtherCAT Configuration**, then click  **Upload ENI-File** button.

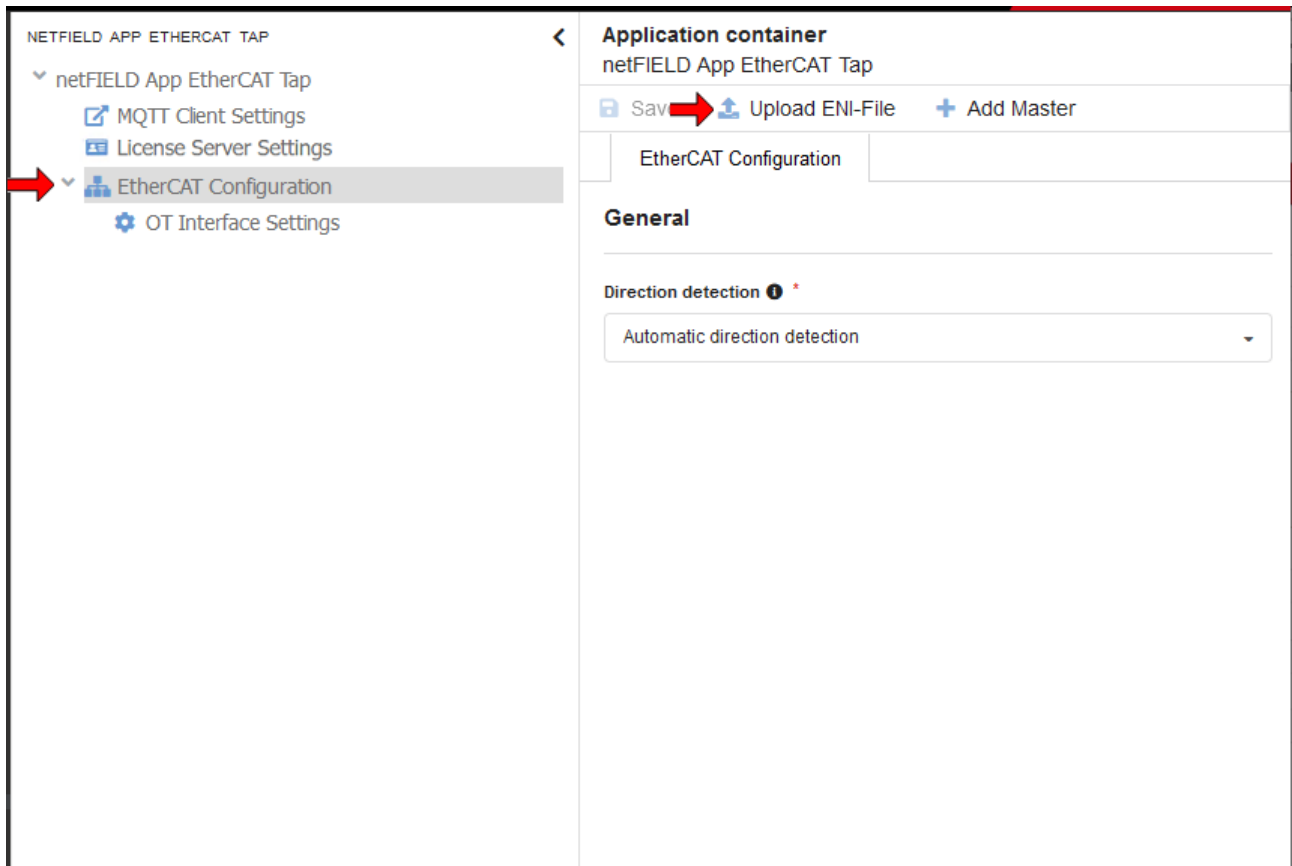


Figure 26: Upload ENI-File

- The **Upload ENI file** dialog opens:

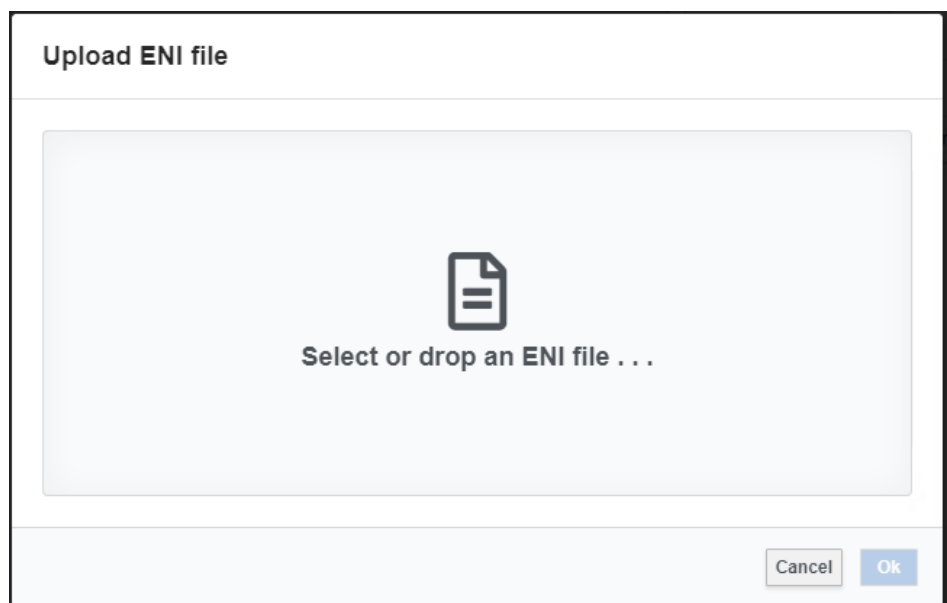


Figure 27: Upload ENI file dialog

- Drag and drop your ENI file in your browser onto the grey area of the dialog window (as an alternative, click into the middle of the dialog area to open the file selection dialog of your browser).



- Click **Ok** button to start the upload.
- The EtherCAT network configuration contained in the ENI file is imported into the EtherCAT Tap. After finishing, the import wizard shows a summary of the import process. If there are errors or warnings, you can manually check the corresponding data items later in the configuration tree.

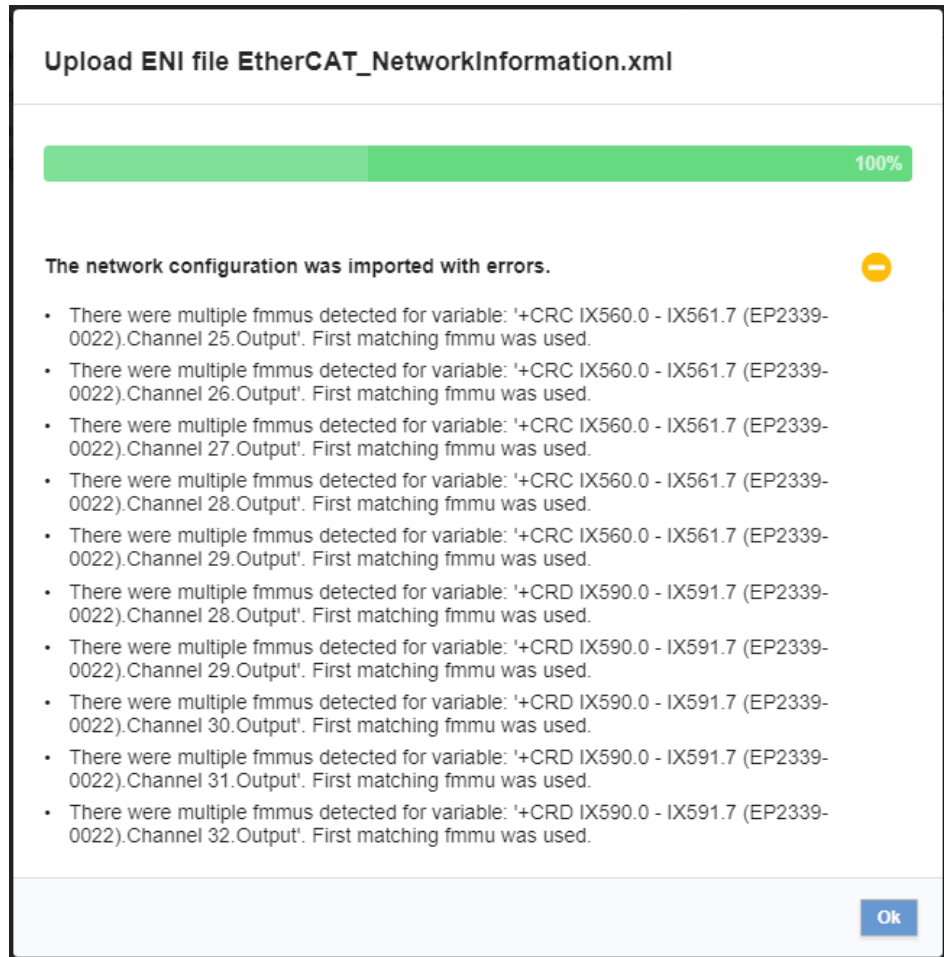


Figure 28: Upload and import finished

- Scroll down to the bottom of the wizard, then click **Ok** button to close the wizard.

➤ The **EtherCAT Configuration** tree now contains the imported network information from the ENI file.

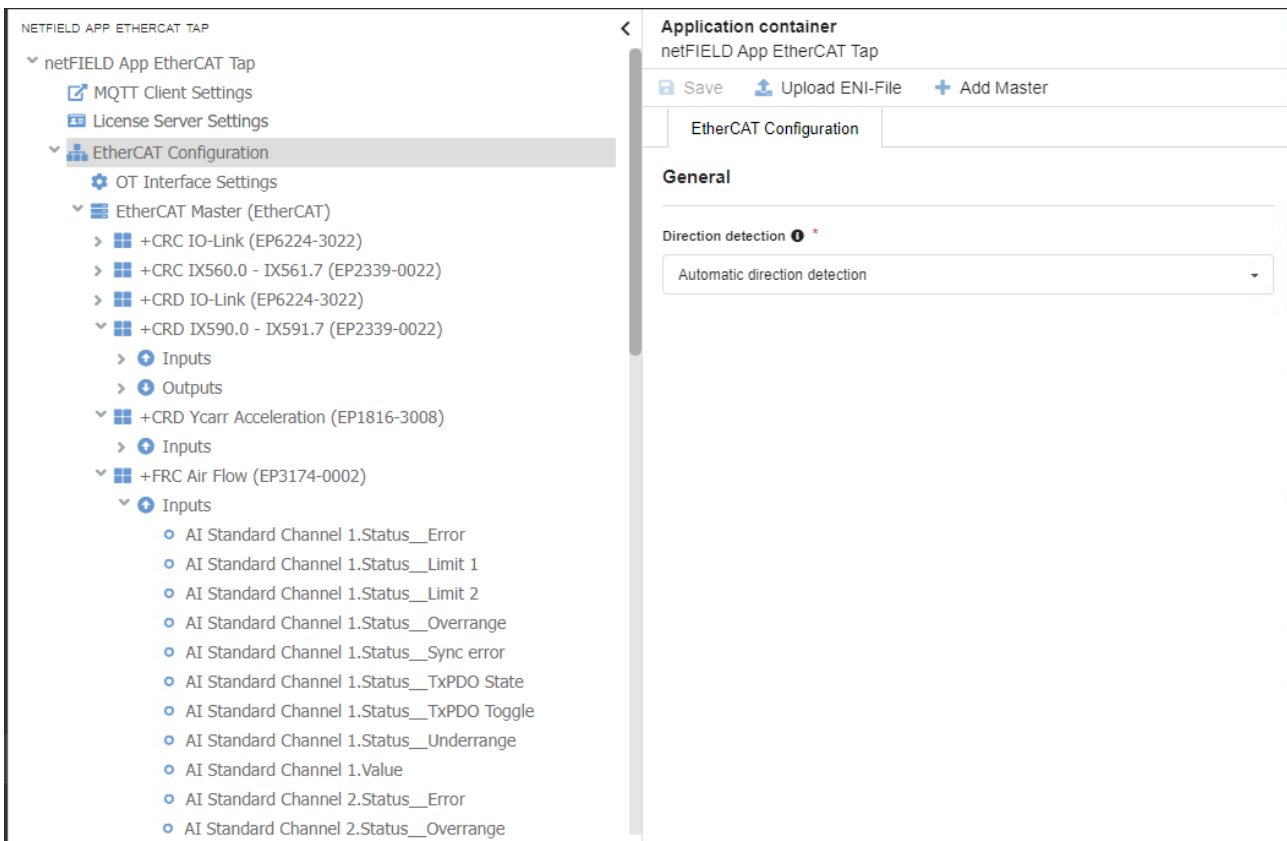


Figure 29: EtherCAT configuration

4. Select Variables for publishing.

- In the **EtherCAT Configuration** tree, select the variable that you want to publish.

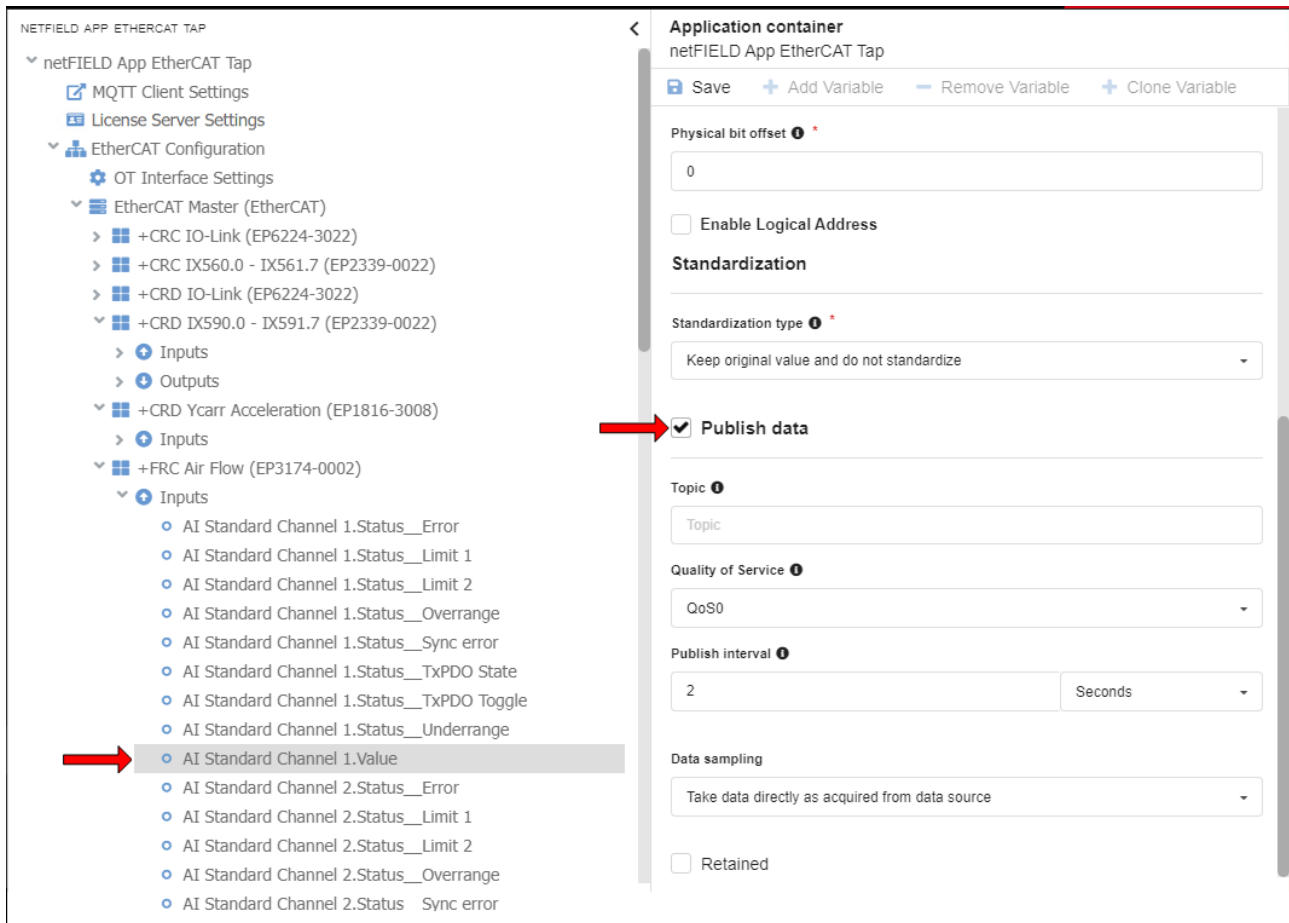


Figure 30: Publish variable

- In the Editor screen of the selected variable, select the **Publish data** option and set the publication parameters, like Topic, MQTT Quality of Service, sampling rate etc.



**Note:**

If you do not enter a Topic here, the EtherCAT Tap will use the following default string:  
`netFIELD/<gateway prefix>/<container id>/<master name>/<slave name>/<direction name>/<variable name>/<node id>`

Note also 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 in the string.

- You may also want to change the name of the variable into something that is more intelligible for the subscribers who will consume the MQTT message, like e.g. "AirflowValue01"

- After having finished setting the publisher parameters, click **Save** in the menu.

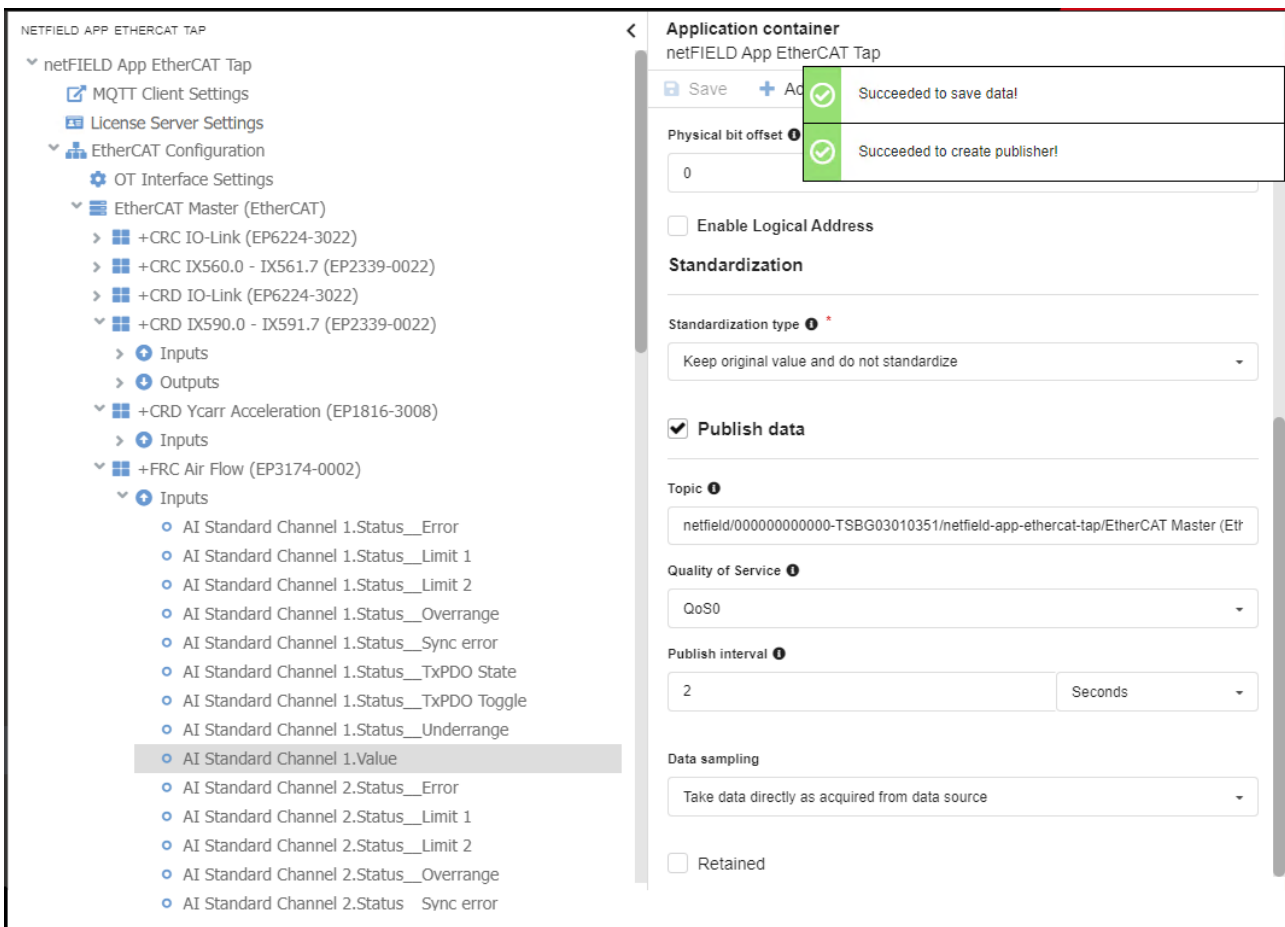


Figure 31: Publisher created

- ⇒ EtherCAT Tap immediately starts to publish the variable to the MQTT Broker according to your settings.



**Note:**

You can keep an overview of your published variables and their parameters in the Publisher Management tab (see section “Publisher Management” tab [▶ page 21]).

## 5 Good to know...

### 5.1 MQTT message format

#### Structure of MQTT Topic (default)



##### Note:

This is the structure of the MQTT Topic string that will be used by default if you do not specify your own **Topic** string when you select a variable in the EtherCAT Configuration tree for publishing (**EtherCAT Variable** > **Publish data** option > **Topic** field).

#### Syntax:

```
netFIELD/<gateway prefix>/<container id>/<master name>/<slave name>/<direction name>/<variable name>/<node id>
```

#### Elements in topic:

Element in topic	Description
<gateway prefix>	If the EtherCAT Tap container is used on a netFIELD OS, the <code>gateway prefix</code> identifies the “source” (= gateway) of the MQTT message according to global MQTT settings of the netFIELD OS. (Local Device Manager > General Settings > Default MQTT Client Settings > Gateway settings > Gateway prefix). <b>Note:</b> By default, the <code>gateway prefix</code> is automatically determined to be identical with the Hardware ID of the netFIELD OS Device respectively of the netFIELD OS Datacenter.
<container id>	Identifier according to <code>CONTAINER_ID</code> environment variable. <b>Note:</b> By default, the Container ID is <code>netfield-app-ethercat-tap</code> , but can be changed by the user. You can change it in the Docker Compose file or – if you are a netFIELD Cloud user – in the netFIELD Portal.
<master name>	Name of the Master
<slave name>	Name of the Slave device
<direction name>	Specifies “inputs” or “outputs”
<variable name>	Name of the variable
<node id>	Unique number assigned to the variable when it was created in the configuration tree. Stays the same until deletion of the element (will not be changed by changing the name of the variable in the configuration GUI, or of any other element which is contained in the topic)

Table 8: Descriptions of elements in topic

#### Example:

```
netfield/000000000000-TSBG03010351/netfield-app-ethercat-tap/
EtherCAT Master (EtherCAT)/.FRC Air Flow (EP3174-0002)/Inputs/
Airflow.Value01/638
```

### Structure of message content

```

{
  "schemaVersion": 1,
  "nodeId": "251",
  "messageType": "processdata",
  "dataType": "octetString",
  "data": [
    {
      "timestamp": "2020-05-28T13:28:55.514890874",
      "value": "0000"
    },
    {
      "timestamp": "2020-05-28T13:28:55.522859992",
      "value": "0100"
    },
    {
      "timestamp": "2020-05-28T13:28:55.530893872",
      "value": "00AB"
    },
    {
      "timestamp": "2020-05-28T13:28:55.538880191",
      "value": "FF00"
    },
    {
      "timestamp": "2020-05-28T13:28:55.546894711",
      "value": "0123"
    }
  ]
}
    
```

Element	Note	
schemaVersion	Identifies the version of the JSON structure of the message (as depicted in the example above). The current structure version is "1". If the structure changes in future, the schemaVersion will be incremented.	
nodeId	Is the same node ID as in the topic.	
messageType	Is always "processdata".	
dataType	Is the data type of the variable as specified in the configuration GUI ("octetString" in this example).	
data	Is the array of one or multiple process data samples contained in this message. The amount may vary according to the sample and publish intervals. Each data element consists of:	
	timestamp	Point in time (in nanosecond resolution and in ISO 8601 format) when the data sample was acquired from the network.
	value	Actual data value according to data type, either as number, bool or string.

Table 9: Notes on message content

## 5.2 Known restrictions of the EtherCAT Tap

This section lists the known restrictions concerning the EtherCAT Tap and the export of the EtherCAT Network Information (ENI) from TwinCAT and its subsequent import in the EtherCAT Tap.

- Word-wise swapped data types are not supported. The data value will be captured and published un-swapped.
- Only slaves containing process data are extracted from the ENI file: Slaves that do not have process data will not be displayed in the EtherCAT Configuration tree.
- Some engineering tools other than TwinCAT do not export the full set of information as specified in the ENI specification, or they do not respect all details of the ENI schema. Thus, importing an ENI file from a tool other than TwinCAT may be prone to errors. Please check the correctness of the configuration after importing such files.
- The hierarchy of the slaves is not taken into account on parsing the ENI file during import. In the EtherCAT Configuration tree, all slaves will be shown on the same level.
- Normalization is only available for variables of type "INT".

## 5.3 Mapping of TwinCAT data types to EtherCAT Tap-supported data types

The following table shows the mapping of data types from the TwinCAT to data types supported by the EtherCAT Tap.

Category	TwinCAT data types	EtherCAT Tap data types
Binary	BOOL	boolean
	BIT	boolean
	BOOLEAN	boolean
Unsigned integers	BYTE	unsigned8
	WORD	unsigned16
	DWORD	unsigned32
	USINT	unsigned8
	UINT	unsigned16
	UINT24	unsigned32
	UDINT	unsigned32
	UINT40	unsigned64
	UINT48	unsigned64
	UINT56	unsigned64
	UINT64	unsigned64
Bit	BIT1	unsigned8
	BIT2	unsigned8
	BIT3	unsigned8
	BIT4	unsigned8
	BIT5	unsigned8
	BIT6	unsigned8
	BIT7	unsigned8
	BIT8	unsigned8
	BIT9	unsigned16
	BIT10	unsigned16
	BIT11	unsigned16
	BIT12	unsigned16
	BIT13	unsigned16
	BIT14	unsigned16
	BIT15	unsigned16
	BIT16	unsigned16
Bit array	BITARR8	unsigned8
	BITARR16	unsigned16
	BITARR32	unsigned32
Integers	SINT	integer8
	INT	integer16
	INT24	integer32
	DINT	integer32
	INT40	integer64
	INT48	integer64
	INT56	integer64
	INT64	integer64
Floating point numbers	REAL	real32
	LREAL	real64



Category	TwinCAT data types	EtherCAT Tap data types
Others	ARRAY_OF_INT	octetString
	ARRAY_OF_SINT	octetString
	ARRAY_OF_DINT	octetString
	ARRAY_OF_UDINT	octetString

Table 10: Mapping of data types from TwinCAT

## 5.4 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 EtherCAT Tap container needs for establishing a secure SSL/TLS connection to the MQTT broker are not managed by the EtherCAT Tap container itself. Instead, they are to be stored on the host device and mapped into the container from the operating system.

For this mapping, the following standard directories are mapped into the container when you use the netFIELD OS and the Docker Compose files from Hilscher (respectively when you deploy the container via netFIELD Portal using the default Container Create Options):

```
/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 Docker Compose file respectively in the netFIELD Portal before deploying the container.

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 EtherCAT Tap 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.

## 5.5 Installing CodeMeter and activating container license

For users who are new to the *CodeMeter* licensing technology from Wibu, this section describes how to setup a CodeMeter server in your local network and activate your netFIELD App container license key.

The **CodeMeter User Runtime for Windows** is used as an example of a CodeMeter server and the *PROFINET Tap* as an example of a netFIELD App container.

1. Download license server from Wibu-Systems.
  - Go to the **CodeMeter User Software** page <https://www.wibu.com/us/support/user/downloads-user-software.html>
  - Under **CodeMeter User Download**, download the **CodeMeter User Runtime** according to the operating system of the server/machine/PC on which you want to install your license server, e.g. **CodeMeter User Runtime for Windows**:

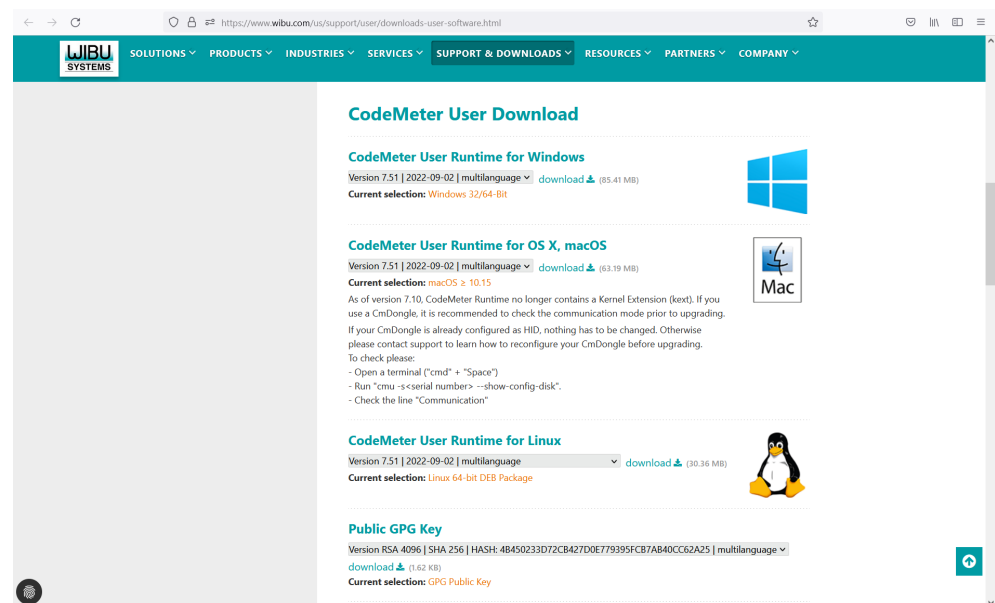


Figure 32: CodeMeter runtime download

2. Install the CodeMeter runtime on your server/machine/PC.
  - Install the license server via the downloaded **CodeMeterRuntime.exe** installer.

- In the **Custom Setup** dialog of the setup wizard, make sure that the **Network Server** option is enabled:

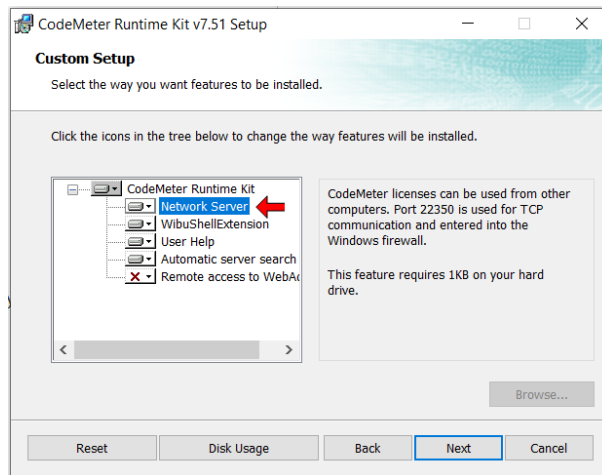


Figure 33: Enable network server option

- After installation, you will find the *CodeMeter Control Center* in the Windows taskbar:

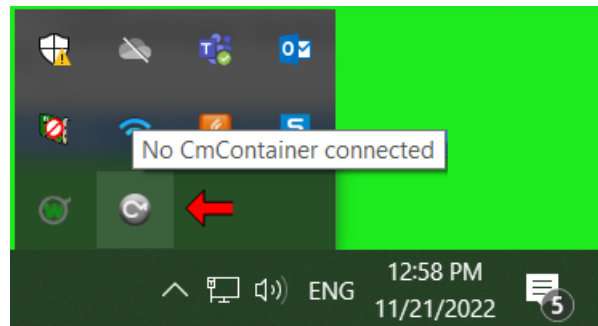


Figure 34: Installed CodeMeter server in taskbar

3. Obtain netFIELD App container license.
  - Contact your local Hilscher sales representative ([www.hilscher.com/company/offices](http://www.hilscher.com/company/offices)) to obtain a license key for your netFIELD App container.
  - After purchasing the license, you will receive a download link for the license key.
4. Activate license key.
  - Open the link using a web browser on the server/machine/PC where the previously installed license server is running.

➤ The link leads you to a Wibu-Systems **Available Licenses** web page:

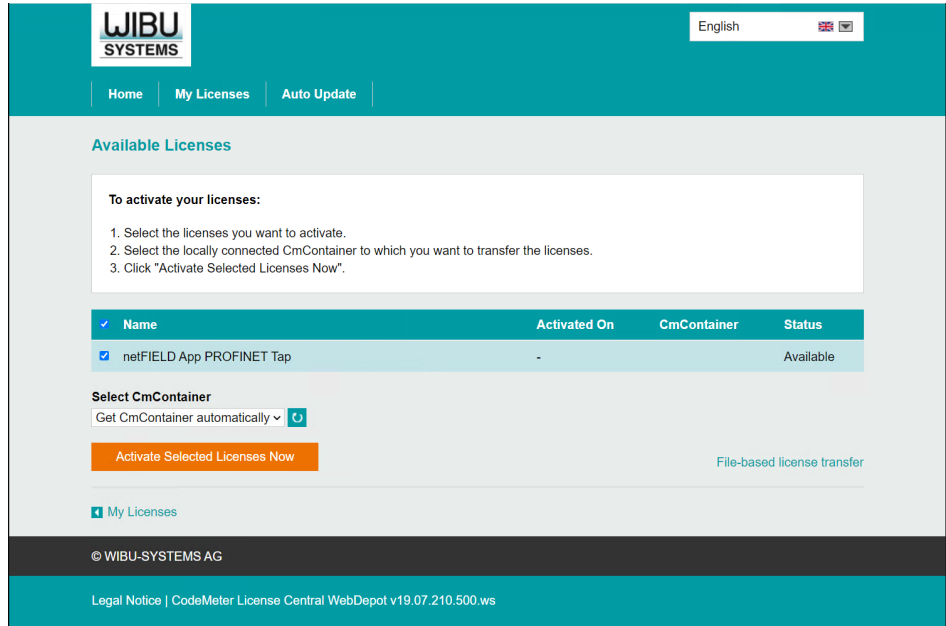


Figure 35: Activate license web page

- Make sure that your netFIELD App container (in this example the **netFIELD App PROFINET Tap** is selected (check-marked), then click **Activate Selected Licenses Now** button.
  - The License key is now automatically downloaded, imported and activated in your local CodeMeter license server.
5. Check activation of license (optional).
- To check your license, open the *CodeMeter Control Center* in the Windows taskbar:

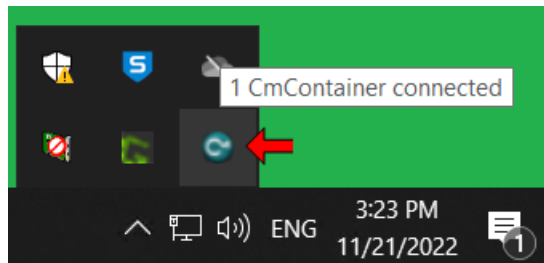


Figure 36: Open CodeMeter Control Center

- In the *CodeMeter Control Center*, you can also open the **WebAdmin** web interface for more details.

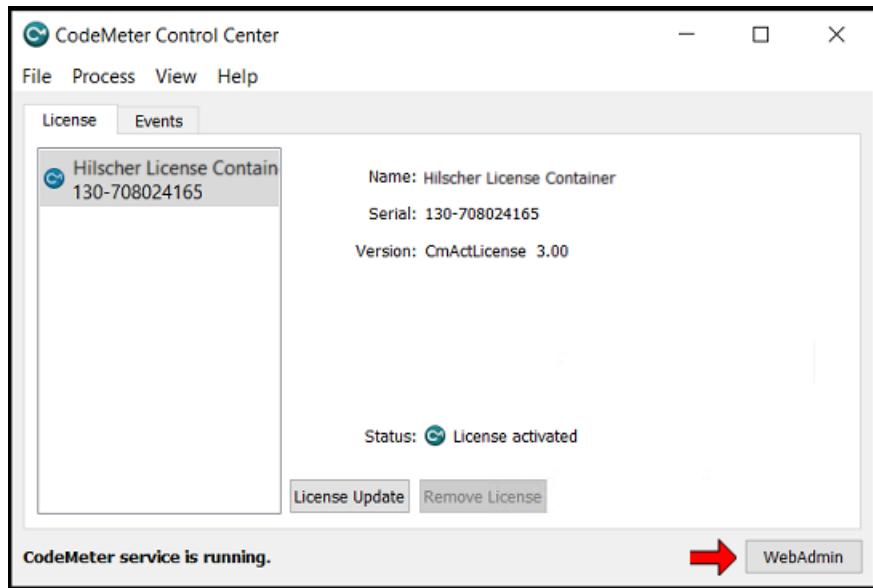


Figure 37: CodeMeter Control Center

- In the **WebAdmin** web interface, open **Container** tab > **Hilscher License Container** > **Licenses** to display information on the license:

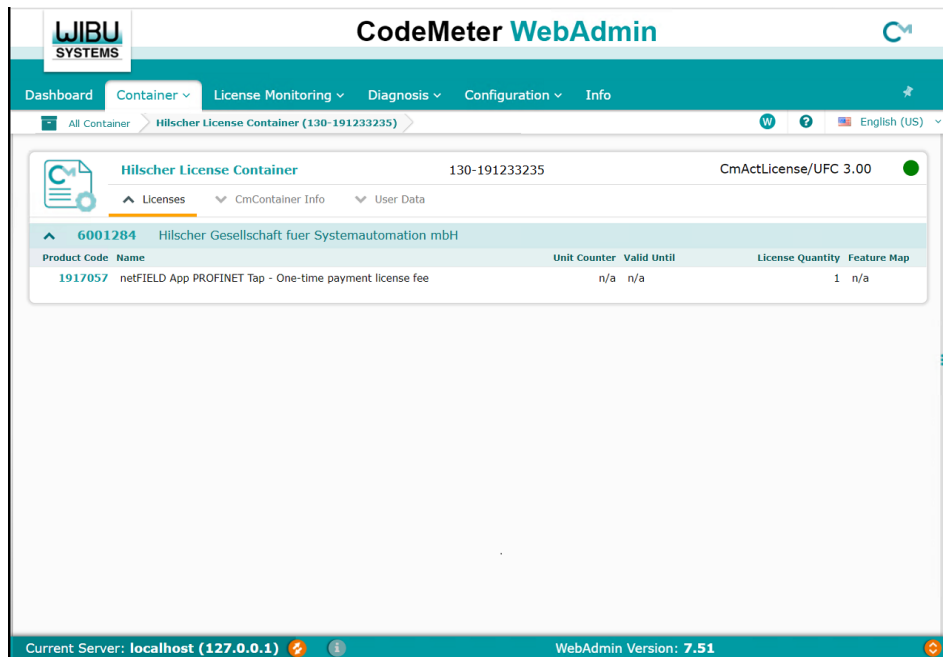


Figure 38: EtherCAT Tap in WebAdmin

- Click on the **Product Code** to open the **Product Item Details** page, where you can find further details, like e.g. the **Maintenance Period**.

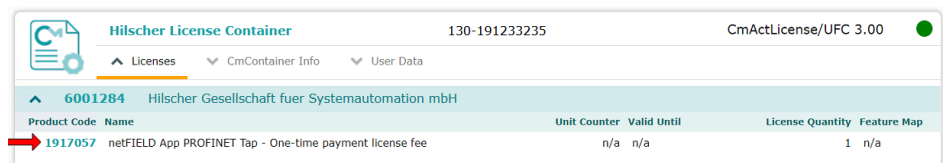
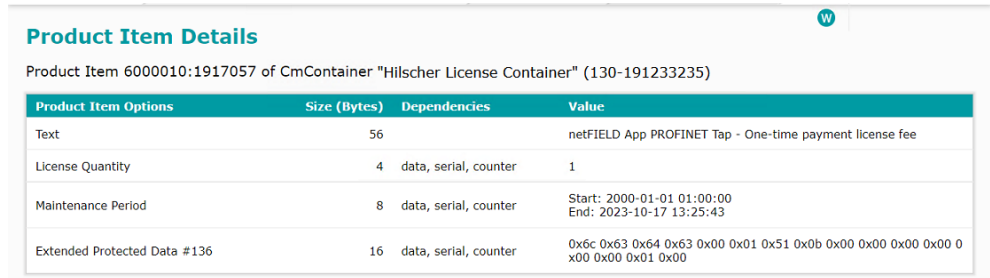


Figure 39: Click on Product Code

➤ The **Maintenance Period** on the **Product Item Details** page tells you for how long you can receive free updates for the app:



Product Item Options	Size (Bytes)	Dependencies	Value
Text	56		netFIELD App PROFINET Tap - One-time payment license fee
License Quantity	4	data, serial, counter	1
Maintenance Period	8	data, serial, counter	Start: 2000-01-01 01:00:00 End: 2023-10-17 13:25:43
Extended Protected Data #136	16	data, serial, counter	0x6c 0x63 0x64 0x63 0x00 0x01 0x51 0x0b 0x00 0x00 0x00 0x00 0x00 0x00 0x01 0x00

Figure 40: Product Item Details

- 6. Configure the CodeMeter license server endpoint in the netFIELD App container.
  - To configure the CodeMeter license server endpoint in your container, open the **License Server Settings** page of the container's configuration web GUI:

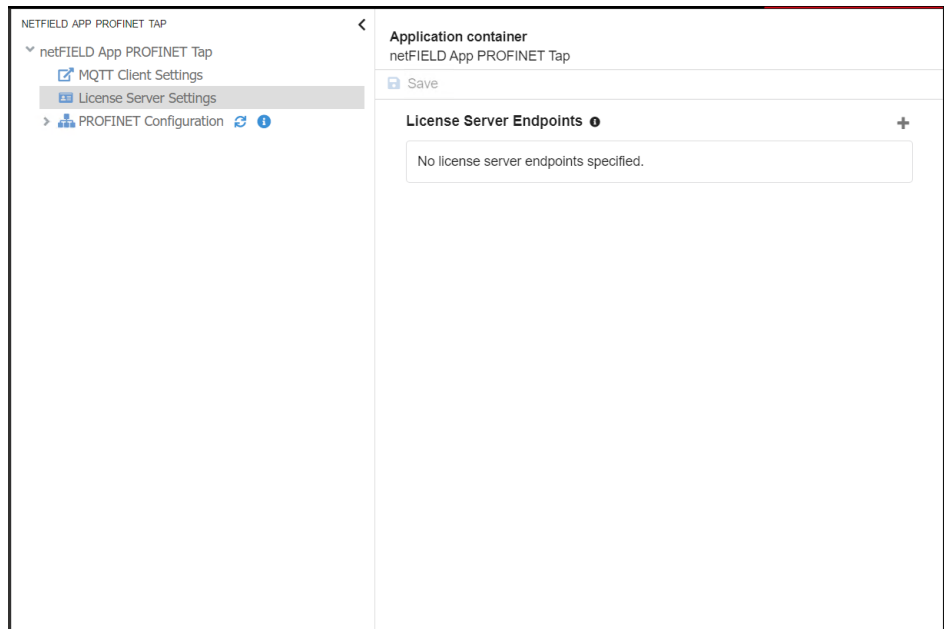


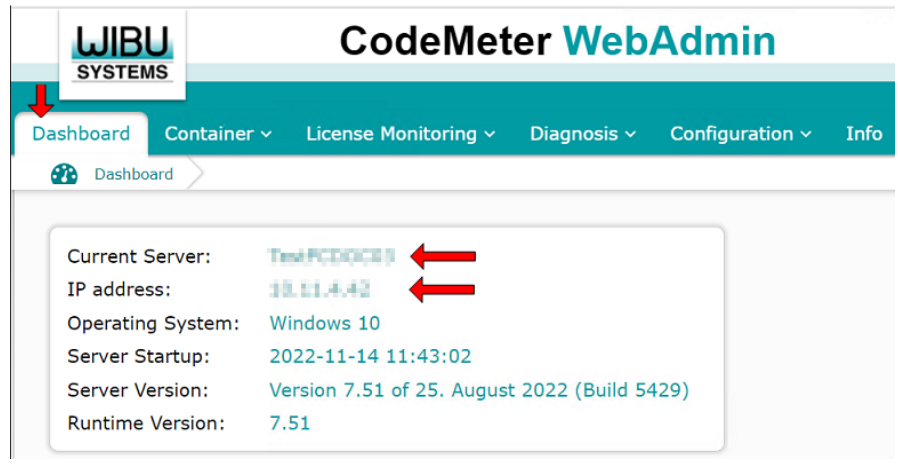
Figure 41: License Server Settings initial screen

- On the **License Server Settings** page, enter the host name or the IP address of the server/machine/PC where the license server is running.



**Note:**

Note that the required host name and IP address are displayed on the **Dashboard** tab of the CodeMeter **WebAdmin** web interface for reference:



- After clicking the **Save** button on the **License Server Settings** page, your netFIELD App container immediately tries to connect to the server and retrieve its license.
7. Check the license state on the **Status** page.
- To check the state, first select the root element in the navigation tree, then open the **Status** tab:

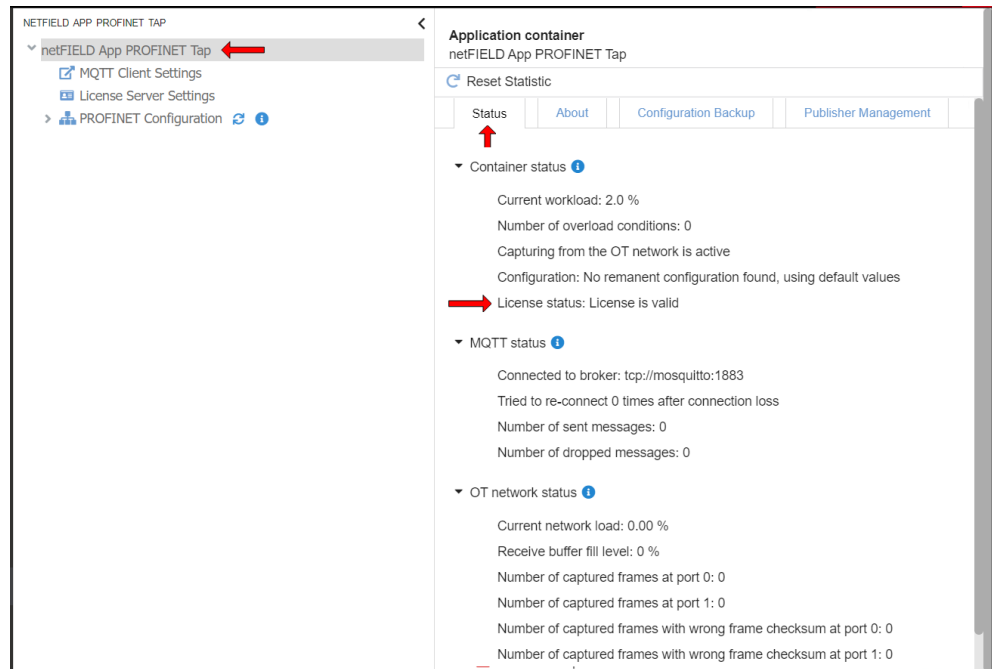


Figure 42: Check license status

## 6 Legal notes

### Terms and conditions

Please read the terms and conditions under <https://www.netfield.io/termsOfUse>.

### Copyright

© Hilscher Gesellschaft für Systemautomation mbH

All rights reserved.

The images, photographs and texts in the accompanying materials (in the form of a user's manual, operator's manual, Statement of Work document and all other document types, support texts, documentation, etc.) are protected by German and international copyright and by international trade and protective provisions. Without the prior written consent, you do not have permission to duplicate them either in full or in part using technical or mechanical methods (print, photocopy or any other method), to edit them using electronic systems or to transfer them. You are not permitted to make changes to copyright notices, markings, trademarks or ownership declarations. Illustrations are provided without taking the patent situation into account. Any company names and product designations provided in this document may be brands or trademarks by the corresponding owner and may be protected under trademark, brand or patent law. Any form of further use shall require the express consent from the relevant owner of the rights.

### 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 uninterrupted 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

Figure 1:	netX-based interface with netMIRROR .....	8
Figure 2:	Connecting netMIRROR ports to netX ports: B to 0 and A to 1.....	9
Figure 3:	Connecting netMIRROR ports to netX ports: B to 1 and A to 0.....	9
Figure 4:	netX-based interface with netMIRROR and LAN connection .....	11
Figure 5:	Connection without hardware mirror.....	12
Figure 6:	netX and LAN without netMIRROR .....	13
Figure 7:	Input at port 0 .....	14
Figure 8:	Output at port 0.....	14
Figure 9:	Configuration GUI of EtherCAT Tap.....	16
Figure 10:	Status tab .....	17
Figure 11:	About tab .....	18
Figure 12:	Configuration Backup .....	19
Figure 13:	Restore configuration .....	20
Figure 14:	Publisher Management.....	21
Figure 15:	Parameters of EtherCAT variable.....	22
Figure 16:	MQTT Client settings.....	24
Figure 17:	License Server Settings .....	27
Figure 18:	License Server Settings saved .....	28
Figure 19:	Check license status.....	28
Figure 20:	OT Interface Settings.....	29
Figure 21:	EtherCAT Configuration tree .....	31
Figure 22:	Parameters of selected EtherCAT element in Editor window.....	32
Figure 23:	Menu bar .....	33
Figure 24:	Set direction detection .....	38
Figure 25:	Set port speed .....	39
Figure 26:	Upload ENI-File .....	40
Figure 27:	Upload ENI file dialog.....	40
Figure 28:	Upload and import finished.....	41
Figure 29:	EtherCAT configuration .....	42
Figure 30:	Publish variable .....	43
Figure 31:	Publisher created.....	44
Figure 32:	CodeMeter runtime download .....	50
Figure 33:	Enable network server option.....	51
Figure 34:	Installed CodeMeter server in taskbar.....	51
Figure 35:	Activate license web page .....	52
Figure 36:	Open CodeMeter Control Center.....	52
Figure 37:	CodeMeter Control Center .....	53
Figure 38:	EtherCAT Tap in WebAdmin .....	53
Figure 39:	Click on Product Code.....	53
Figure 40:	Product Item Details .....	54

---

Figure 41:	License Server Settings initial screen.....	54
Figure 42:	Check license status.....	55

## List of Tables

Table 1:	List of revisions .....	3
Table 2:	About tab .....	18
Table 3:	Elements of Publisher Management tab.....	21
Table 4:	Elements Variable's details window.....	22
Table 5:	MQTT Client Settings .....	25
Table 6:	Parameters OT Interface Settings .....	30
Table 7:	Elements EtherCAT Configuration.....	37
Table 8:	Descriptions of elements in topic .....	45
Table 9:	Notes on message content .....	46
Table 10:	Mapping of data types from TwinCAT .....	48

# Contacts

## HEADQUARTER

### 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](mailto:info@hilscher.com)

### Support

Phone: +49 (0) 6190 9907-990  
E-mail: [hotline@hilscher.com](mailto:hotline@hilscher.com)

## SUBSIDIARIES

### China

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

### Support

Phone: +86 (0) 21-6355-5161  
E-mail: [cn.support@hilscher.com](mailto: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](mailto:info@hilscher.fr)

### Support

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

### India

Hilscher India Pvt. Ltd.  
Pune, Delhi, Mumbai, Bangalore  
Phone: +91 8888 750 777  
E-mail: [info@hilscher.in](mailto:info@hilscher.in)

### Support

Phone: +91 8108884011  
E-mail: [info@hilscher.in](mailto:info@hilscher.in)

### Italy

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

### Support

Phone: +39 02 25007068  
E-mail: [it.support@hilscher.com](mailto:it.support@hilscher.com)

### Japan

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

### Support

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

### Republic of Korea

Hilscher Korea Inc.  
13494, Seongnam, Gyeonggi  
Phone: +82 (0) 31-739-8361  
E-mail: [info@hilscher.kr](mailto:info@hilscher.kr)

### Support

Phone: +82 (0) 31-739-8363  
E-mail: [kr.support@hilscher.com](mailto:kr.support@hilscher.com)

### Austria

Hilscher Austria GmbH  
4020 Linz  
Phone: +43 732 931 675-0  
E-mail: [sales.at@hilscher.com](mailto:sales.at@hilscher.com)

### Support

Phone: +43 732 931 675-0  
E-mail: [at.support@hilscher.com](mailto:at.support@hilscher.com)

### Switzerland

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

### Support

Phone: +41 (0) 32 623 6633  
E-mail: [support.swiss@hilscher.com](mailto:support.swiss@hilscher.com)

### USA

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

### Support

Phone: +1 630-505-5301  
E-mail: [us.support@hilscher.com](mailto:us.support@hilscher.com)