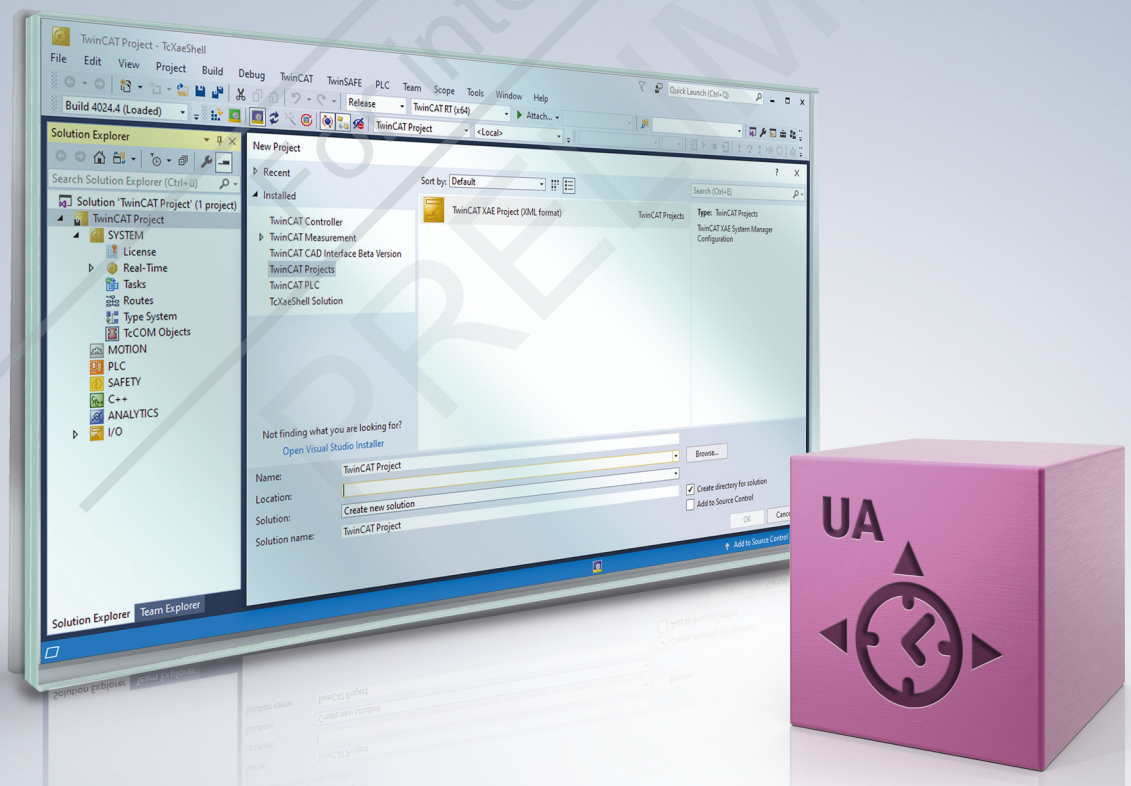
BECKHOFF New Automation Technology

Manual | EN

TF6105

TwinCAT 3 | OPC UA Pub/Sub



2024-07-05 | Version: 1.0.0

Table of contents

1 Foreword

1.1 Notes on the documentation

This description is intended exclusively for trained specialists in control and automation technology who are familiar with the applicable national standards.
For installation and commissioning of the components, it is absolutely necessary to observe the documentation and the following notes and explanations.
The qualified personnel is obliged to always use the currently valid documentation.

The responsible staff must ensure that the application or use of the products described satisfies all requirements for safety, including all the relevant laws, regulations, guidelines, and standards.

Disclaimer

The documentation has been prepared with care. The products described are, however, constantly under development.
We reserve the right to revise and change the documentation at any time and without notice.
No claims to modify products that have already been supplied may be made on the basis of the data, diagrams, and descriptions in this documentation.

Trademarks

Beckhoff®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® and XPlanar® are registered and licensed trademarks of Beckhoff Automation GmbH.
If third parties make use of designations or trademarks used in this publication for their own purposes, this could infringe upon the rights of the owners of the said designations.

Patents

The EtherCAT Technology is covered, including but not limited to the following patent applications and patents:
EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702
and similar applications and registrations in several other countries.

EtherCAT®

EtherCAT® is registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany

Copyright

1.2 For your safety

Safety regulations

Read the following explanations for your safety.
Always observe and follow product-specific safety instructions, which you may find at the appropriate places in this document.

Exclusion of liability

All the components are supplied in particular hardware and software configurations which are appropriate for the application. Modifications to hardware or software configurations other than those described in the documentation are not permitted, and nullify the liability of Beckhoff Automation GmbH & Co. KG.

Personnel qualification

This description is only intended for trained specialists in control, automation, and drive technology who are familiar with the applicable national standards.

Signal words

The signal words used in the documentation are classified below. In order to prevent injury and damage to persons and property, read and follow the safety and warning notices.

Personal injury warnings

⚠ DANGER

Hazard with high risk of death or serious injury.

⚠ WARNING

Hazard with medium risk of death or serious injury.

⚠ CAUTION

There is a low-risk hazard that could result in medium or minor injury.

Warning of damage to property or environment

NOTICE

The environment, equipment, or data may be damaged.

Information on handling the product

This information includes, for example:
recommendations for action, assistance or further information on the product.

1.3 Notes on information security

The products of Beckhoff Automation GmbH & Co. KG (Beckhoff), insofar as they can be accessed online, are equipped with security functions that support the secure operation of plants, systems, machines and networks. Despite the security functions, the creation, implementation and constant updating of a holistic security concept for the operation are necessary to protect the respective plant, system, machine and networks against cyber threats. The products sold by Beckhoff are only part of the overall security concept. The customer is responsible for preventing unauthorized access by third parties to its equipment, systems, machines and networks. The latter should be connected to the corporate network or the Internet only if appropriate protective measures have been set up.

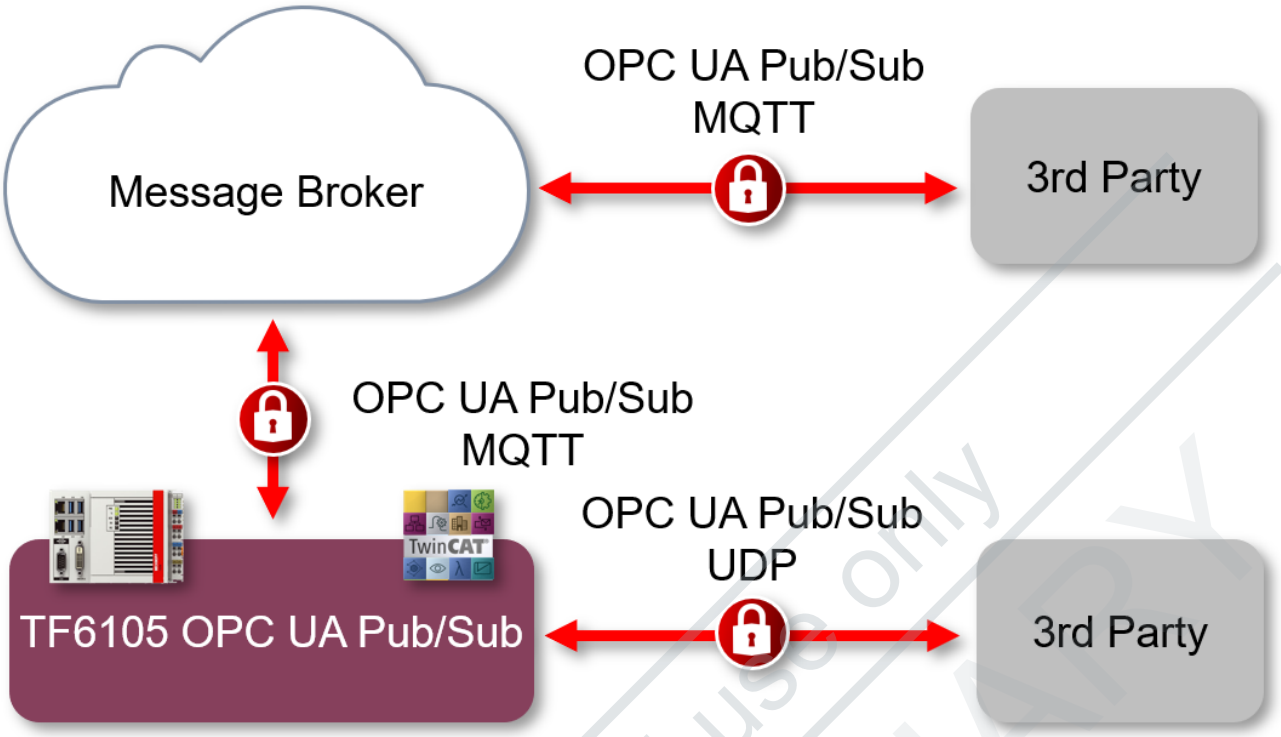
In addition, the recommendations from Beckhoff regarding appropriate protective measures should be observed. Further information regarding information security and industrial security can be found in our https://www.beckhoff.com/secguide.

Beckhoff products and solutions undergo continuous further development. This also applies to security functions. In light of this continuous further development, Beckhoff expressly recommends that the products are kept up to date at all times and that updates are installed for the products once they have been made available. Using outdated or unsupported product versions can increase the risk of cyber threats.

To stay informed about information security for Beckhoff products, subscribe to the RSS feed at https://www.beckhoff.com/secinfo.

2 Overview

The product TF6105 TC3 OPC UA Pub/Sub includes connectivity via different transport protocols as specified by the Pub/Sub extension to the OPC UA specification (OPC10000-14). A variety of functionalities from the specification has been implemented and is available for usage.



This product includes:

- a real-time driver to enable OPC UA Pub/Sub communication via UDP unicast or multicast
- a real-time driver to enable OPC UA Pub/Sub communication via MQTT
- an extension for the TwinCAT System Manager to configure OPC UA Pub/Sub
- an extension for the TwinCAT System Manager to scan remote OPC UA server devices
- an optional integration into the TwinCAT OPC UA Server (TF6100 TC3 OPC UA) to use the configuration interface based on the standardized OPC UA Pub/Sub information model.

3 Installation

3.1 System requirements

Technical data	Description
Operating system	Windows 7/10, Windows Embedded Standard 7
Target platform	PC architecture (x86, x64)
Minimum TwinCAT version	TwinCAT 3.1 Build 4026
Required TwinCAT setup level	TwinCAT 3 XAE, XAR
Required TwinCAT license	TF6105 TC3 OPC UA Pub/Sub
Hardware requirements	This product provides support for different transport protocols. If you want to use UDP, a network interface card which is compatible with the TwinCAT Realtime-Ethernet Adapter driver is required, see below. If you want to use MQTT, no special network interface card is required.

Realtime-Ethernet Adapter

For UDP communication, this product is based on the TwinCAT Realtime-Ethernet Adapter driver. To check whether your network interface card is suitable for this driver, please perform the following steps. Please note that the installation of this driver is not required if you only want to use OPC UA Pub/Sub via MQTT.

1. Create an OPC UA RT Device by right clicking on **Devices** and selecting **Add New Item.**
2. Expand the category OPC UA and select **Real-Time OPC UA Device.**
3. Now select the adapter and find the appropriate Ethernet interface by selecting **Compatible Devices**.
4. Try to install the TwinCAT RT-Ethernet Adapter driver for your network interface card.

„For demo use only“

If your network interface card is listed under the “for demo use only” category, the TwinCAT RT-Ethernet Adapter driver has been installed but might not provide realtime capabilities. However, you may still use it for evaluation and testing.

3.2 Installation

If you are using TwinCAT 3.1 Build 4026 (and higher) on the Microsoft Windows operating system, you can install this function via the TwinCAT Package Manager, see Installation documentation. Normally you install the function via the corresponding workload. However, you can also install the packages contained in the workload individually. This documentation briefly describes the installation process via the workload.

Command line program TcPkg

You can use the TcPkg Command Line Interface (CLI) to display the available workloads on the system:

```
tcpkg list -t workload
```

You can use the following command to install the workload of this function.

```
tcpkg install TF610x.OpcUaClientPubSub.XAE
tcpkg install TF610x.OpcUaClientPubSub.XAR
```

TwinCAT Package Manager UI

You can use the User Interface (UI) to display all available workloads and install them if required.

To do this, follow the corresponding instructions in the interface.

Installation of the corresponding package or workload can be either done via TcPkg CLI or TcPkg UI.

Unprepared TwinCAT restart can cause data loss

The installation of this function may result in a TwinCAT restart.

Make sure that no critical TwinCAT applications are running on the system or shut them down in an orderly manner first.

3.3 Licensing

The TwinCAT 3 function can be activated as a full version or as a 7-day test version. Both license types can be activated via the TwinCAT 3 development environment (XAE).

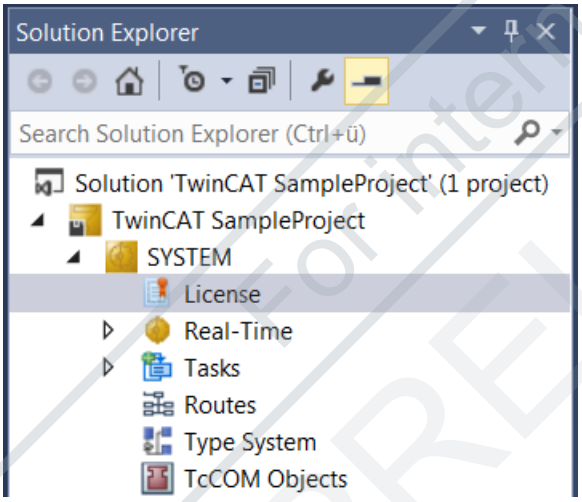
Licensing the full version of a TwinCAT 3 Function

A description of the procedure to license a full version can be found in the Beckhoff Information System in the documentation "TwinCAT 3 Licensing".

Licensing the 7-day test version of a TwinCAT 3 Function

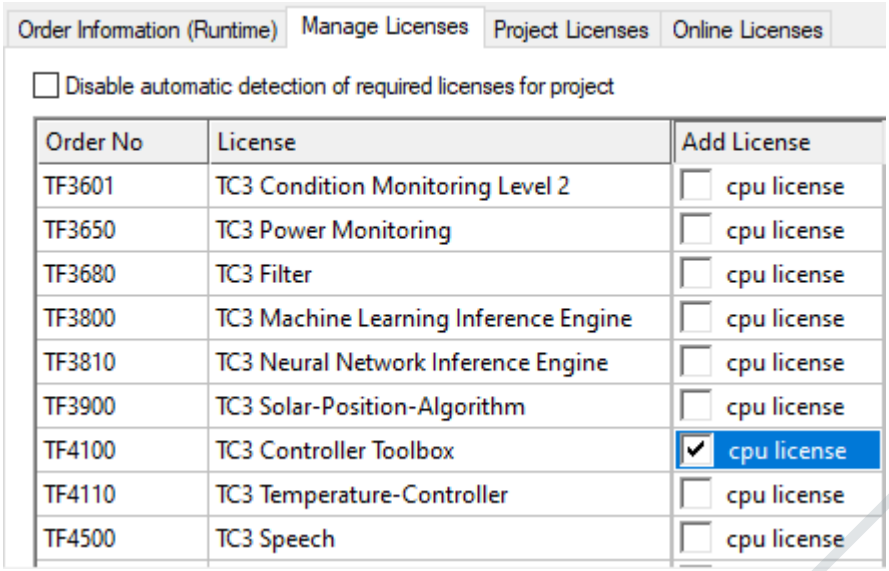
A 7-day test version cannot be enabled for a TwinCAT 3 license dongle.

1. Start the TwinCAT 3 development environment (XAE).
2. Open an existing TwinCAT 3 project or create a new project.
3. If you want to activate the license for a remote device, set the desired target system. To do this, select the target system from the **Choose Target System** drop-down list in the toolbar.
   ⇨ The licensing settings always refer to the selected target system. When the project is activated on the target system, the corresponding TwinCAT 3 licenses are automatically copied to this system.
4. In the **Solution Explorer**, double-click **License** in the **SYSTEM** subtree.

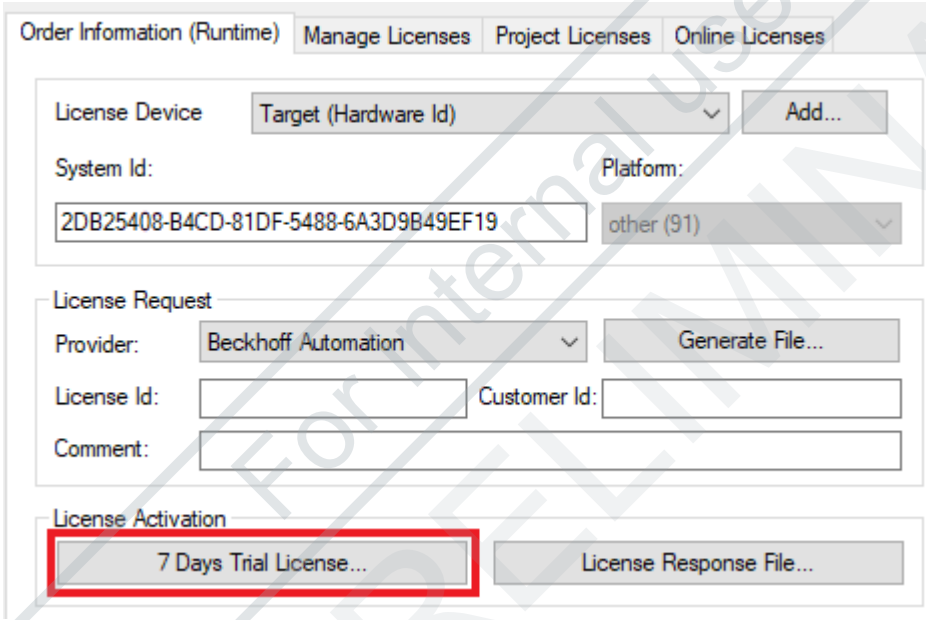


⇨ The TwinCAT 3 license manager opens.

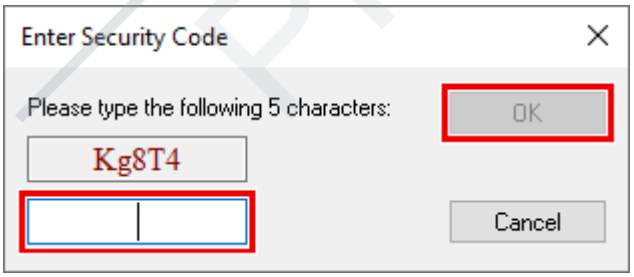
5. Open the **Manage Licenses** tab. In the **Add License** column, check the check box for the license you want to add to your project (e.g. "TF4100 TC3 Controller Toolbox").



6. Open the **Order Information (Runtime)** tab.
   ⇨ In the tabular overview of licenses, the previously selected license is displayed with the status “missing”**.**
7. Click **7-Day Trial License...** to activate the 7-day trial license.



   ⇨ A dialog box opens, prompting you to enter the security code displayed in the dialog.



8. Enter the code exactly as it is displayed and confirm the entry.
9. Confirm the subsequent dialog, which indicates the successful activation.
   ⇨ In the tabular overview of licenses, the license status now indicates the expiry date of the license.

10. Restart the TwinCAT system.

⇨ The 7-day trial version is enabled.

4 Technical introduction

4.1 Quick Start

The following chapter guides you through the first time setup of an OPC UA Pub/Sub configuration in TwinCAT. Please make sure that you have followed the installation [▶ 9] instructions before moving on and that your system meets the system requirements [▶ 9]. The following quick start tutorials are available:

Tutorial	Description
UDP [▶ 13]	Demonstrates how to set up a PLC project, OPC UA RT device and a publisher that sends a data set containing one variable via UDP. The variable is linked to a variable from the PLC project via the process image.
MQTT [▶ 17]	Demonstrates how to set up a PLC project, OPC UA RT device and a publisher that sends a data set containing one variable via MQTT. The variable is linked to a variable from the PLC project via the TwinCAT Target Browser.

Offline configuration exchange

OPC UA Pub/Sub defines a file format for offline configuration exchange that is based on the UA Binary format (OPC UA specification, chapter 5.2). If you want to set up an OPC UA Pub/Sub configuration between two different devices, we strongly recommend using this common exchange file. The chapter Configuration import/export [▶ 35] includes more information about this topic.

4.1.1 UDP

This quick start tutorial demonstrates how to set up a PLC project, OPC UA RT device and a publisher that sends a data set containing one variable via UDP. The variable is linked to a variable from the PLC project via the process image.

Preparing the PLC project

Prepare a PLC project that defines one output variable with data type INT, e.g.:

```
PROGRAM MAIN
VAR
  nCounter AT%Q* : INT;
END_VAR

nCounter := nCounter + 1;
```

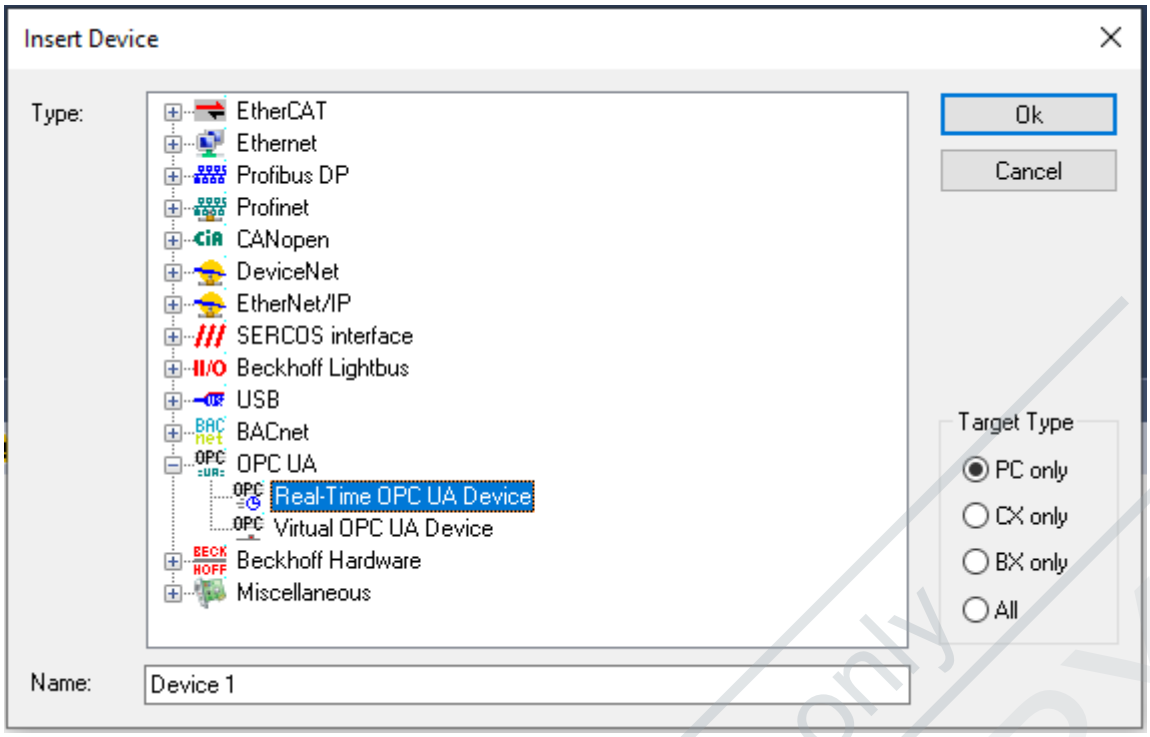
Compile the PLC project so that the PLC process image is created.

Configuring a publisher

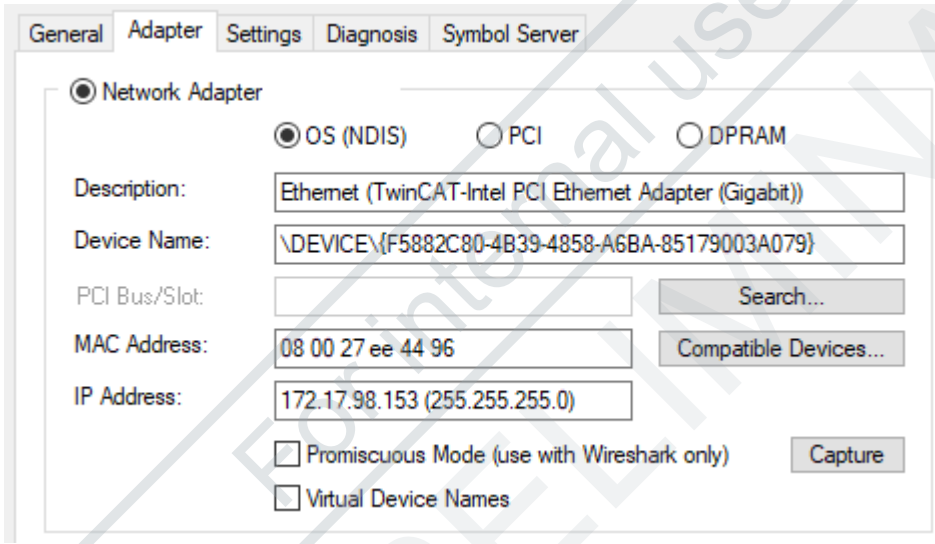
Please perform the following configuration steps to set up an OPC UA Publisher device:

1. Make sure that TwinCAT is in config mode.
2. In Solution Explorer, expand the I/O node and select the child item Devices.
3. Right-click Devices and select **Add new element...**

4. Expand the category **OPC UA**, select the **Real-Time OPC UA Device** and click on **Ok**.

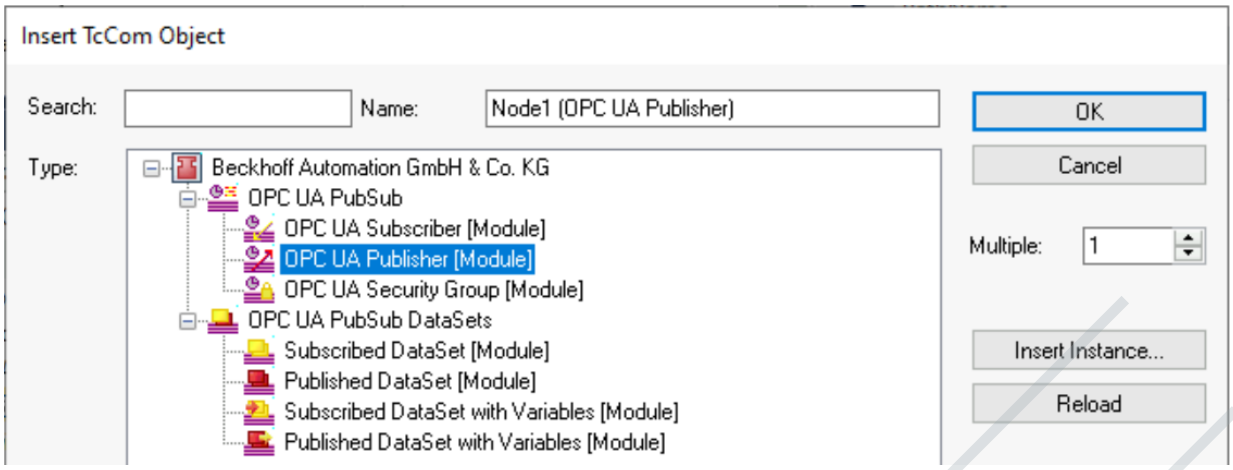


5. Double-click the added device and navigate to the tab **Adapter**.

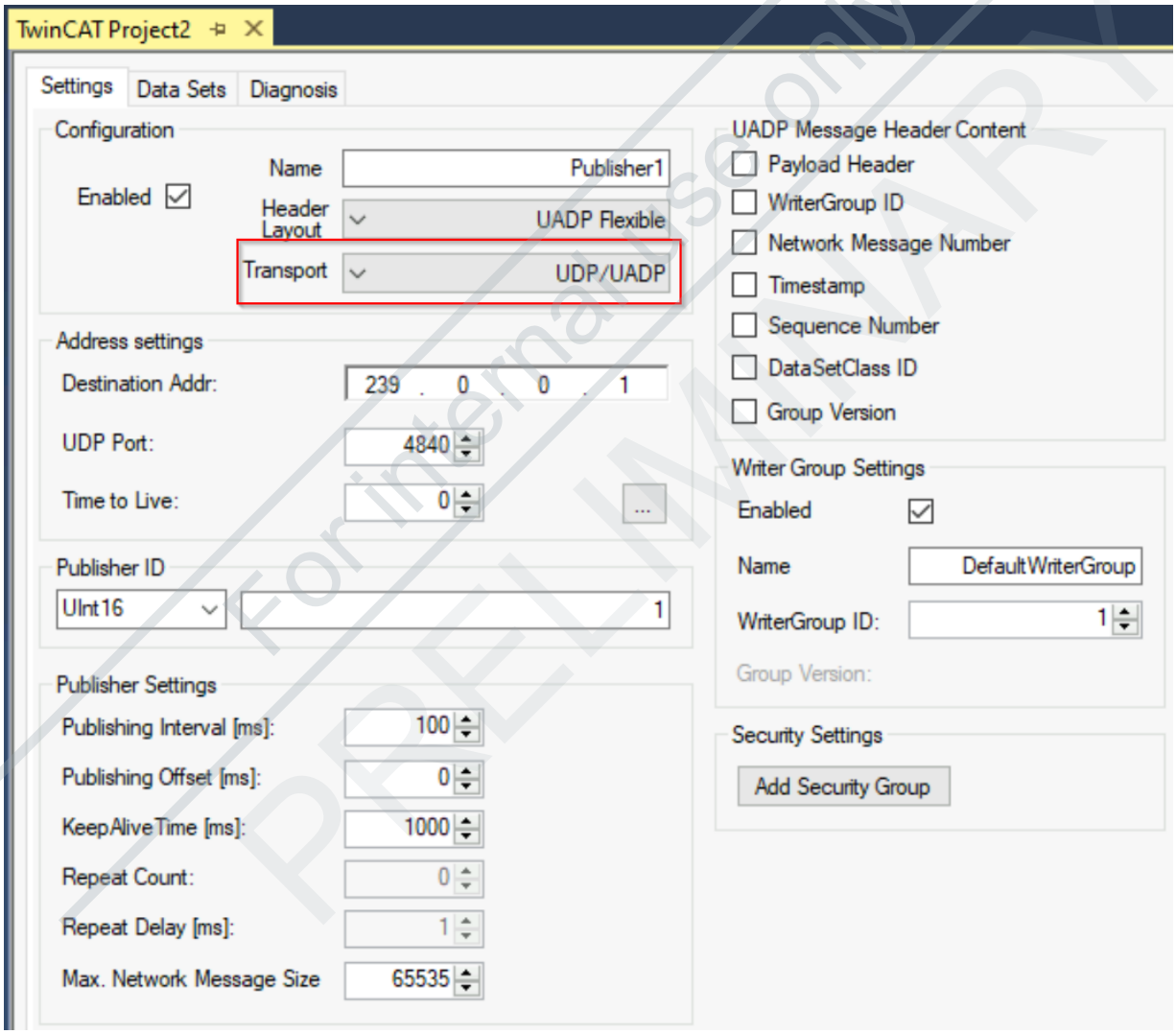


6. Bind the device to one of your Network Interface Cards. Make sure that you have installed the TwinCAT-Intel PCI Ethernet Adapter driver on that NIC.
7. Right-click the device and select **Add new element...**

8. Select the entry **OPC UA Publisher (Module)** and click on **Ok**.

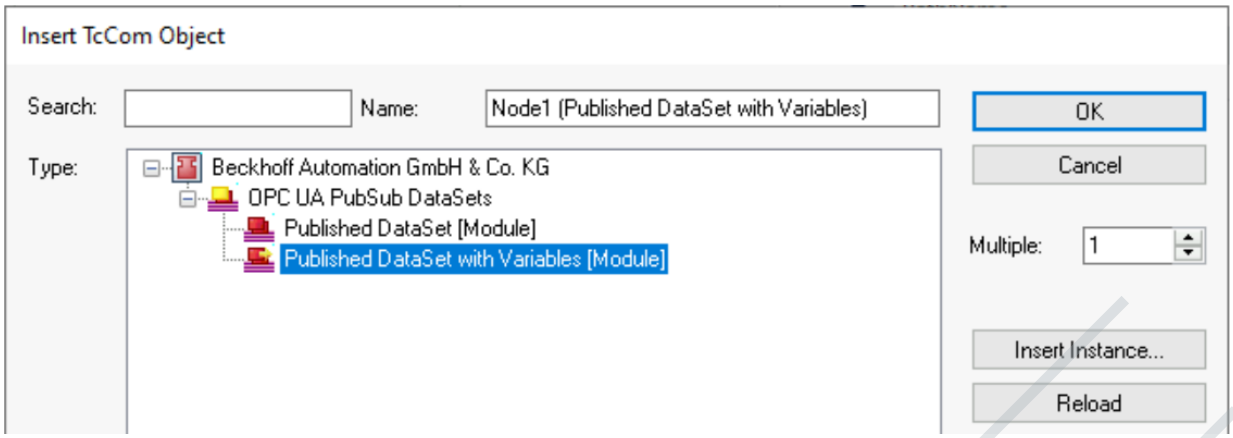


9. To configure the added **Publisher** node, simply double click it in the tree view. There are many configuration options on the Publisher, e.g. the transport to use (UDP, MQTT), the data format (binary, JSON), optional header fields and so on. We want to leave these settings on their default values for the time being. Only make sure that the transport **UDP/UADP** is selected.

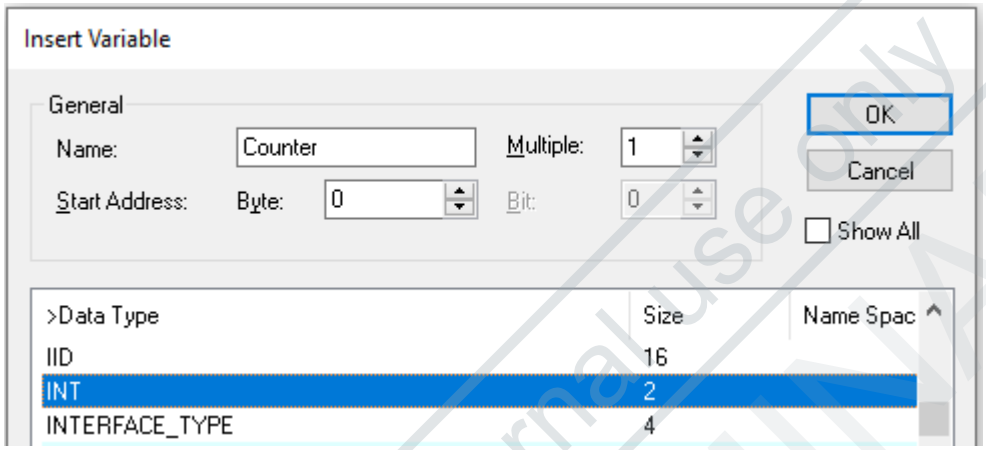


10. In the next step you want to configure a so-called Dataset in order to define variables that should be send out by the Publisher. Right-click the **Publisher** node and select **Add new element....**

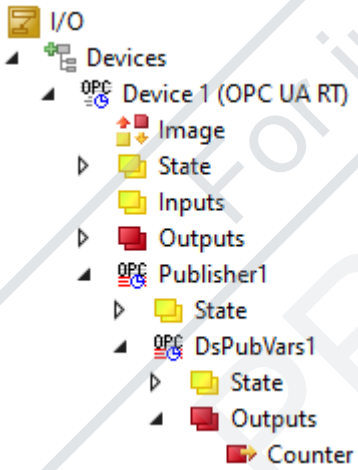
11. Select the entry **Published DataSet with Variables (Module)** and click on **Ok.**



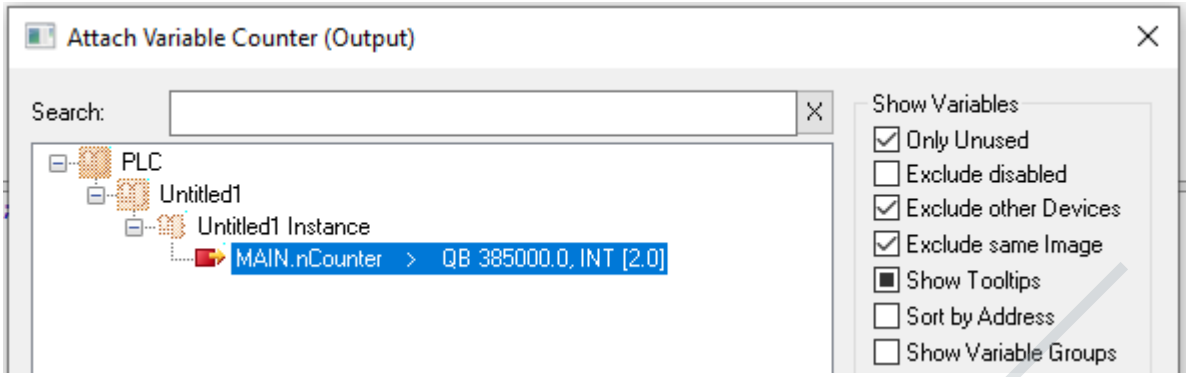
12. On the added data set, right-click the **Outputs** node and select **Add new element...**
13. Set a name for the process variable, e.g. **Counter** and select **INT** as the Datatype. Click on **Ok** to add the variable to the process image of the Dataset.



⇨ Your configuration should now look as follows:



14. Right-click the added variable and select **Change link...**. Select the PLC variable that you have created earlier and click on **Ok**.



Activate the project

Activate the TwinCAT project by clicking the **Activate** button on the TwinCAT XAE toolbar. Make sure that you have selected the correct target device (we assume that you are using your local device) and that you are using the correct Real-Time settings so that you can activate the TwinCAT Runtime on your device. Please consult the regular TwinCAT documentation for more information about how to activate a TwinCAT configuration and switch TwinCAT into Run mode.

In this quick start tutorial, we have used UDP as the transport protocol. You can now connect either an OPC UA Pub/Sub subscriber to the data or also have a look at the data using Wireshark. If you want to connect a subscriber, we highly recommend to make use of the Configuration import/export [▶ 35].

4.1.2 MQTT

This quick start tutorial demonstrates how to set up a PLC project, OPC UA RT device and a publisher that sends a data set containing one variable via MQTT. The variable is linked to a variable from the PLC project via the process image.

Preparing the PLC project

Prepare a PLC project that defines one variable with data type INT, e.g.:

```
PROGRAM MAIN
VAR
  nCounter : INT;
END_VAR

nCounter := nCounter + 1;
```

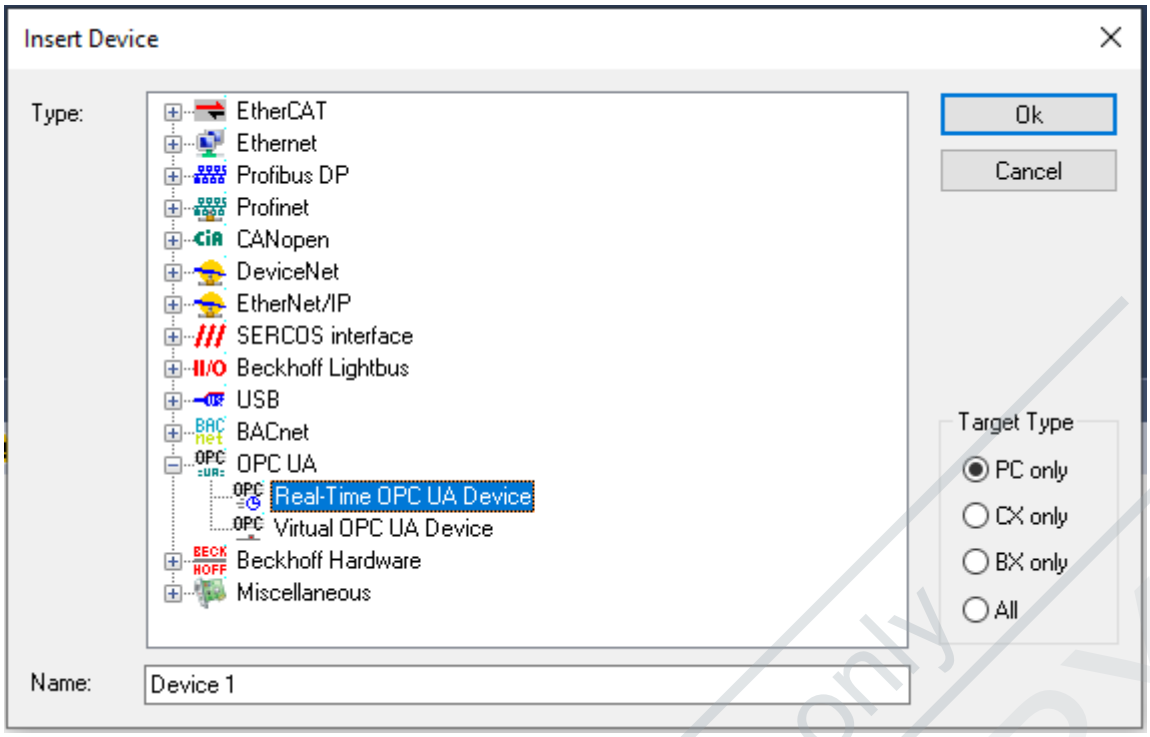
Please note that you do not need to set any input/output compiler statement because we will not use the process image to link the variable. Activate this configuration and start the PLC program.

Configuring a publisher

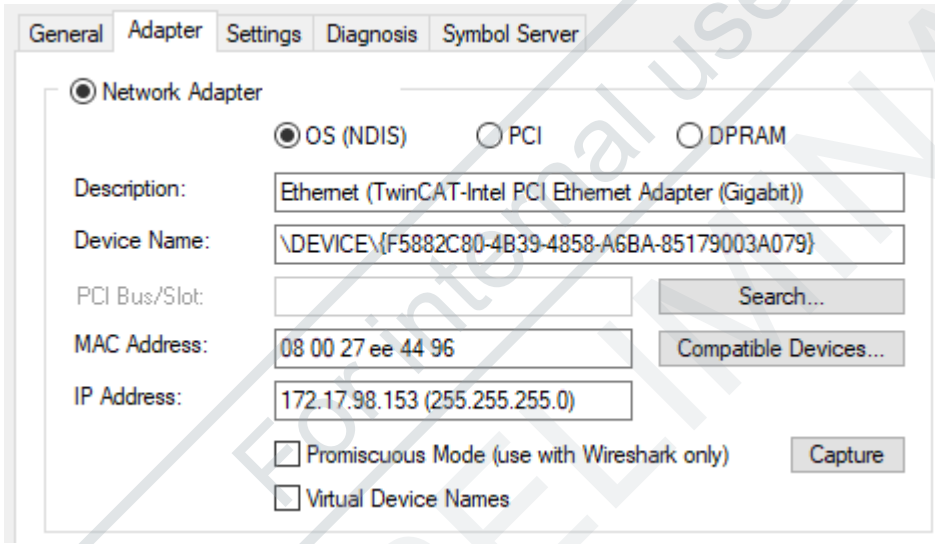
Please perform the following configuration steps to set up an OPC UA Publisher device:

1. Make sure that TwinCAT is in config mode.
2. In Solution Explorer, expand the I/O node and select the child item **Devices**.
3. Right-click **Devices** and select **Add new element...**

4. Expand the category **OPC UA**, select the **Real-Time OPC UA Device** and click on **Ok**.

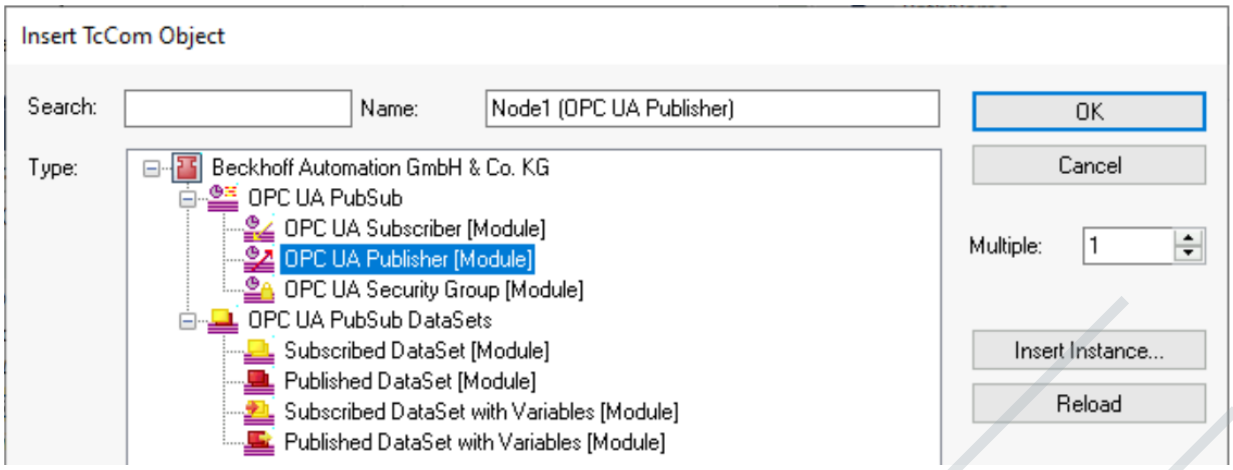


5. Double-click the added device and navigate to the tab **Adapter**.

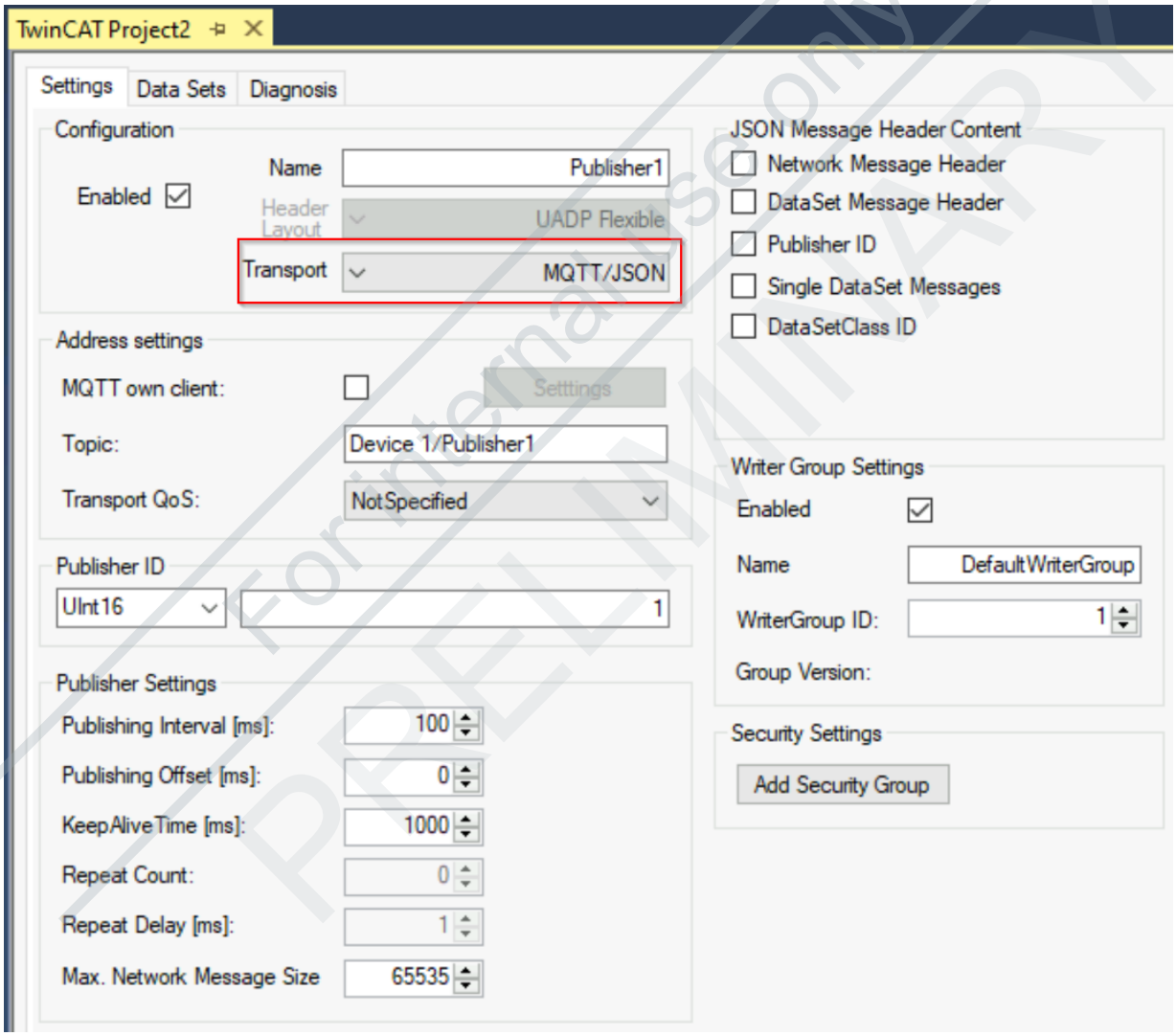


6. Bind the device to one of your Network Interface Cards. Make sure that you have installed the TwinCAT-Intel PCI Ethernet Adapter driver on that NIC.
7. Right-click the device and select **Add new element...**

8. Select the entry **OPC UA Publisher (Module)** and click on **Ok**.

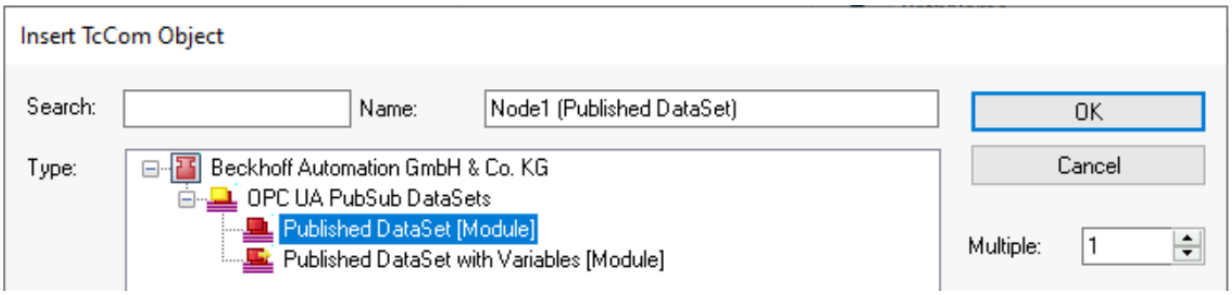


9. To configure the added **Publisher** node, simply double click it in the tree view. There are many configuration options on the Publisher, e.g. the transport to use (UDP, MQTT), the data format (binary, JSON), optional header fields and so on. We want to leave these settings on their default values for the time being and only change the transport setting to **MQTT/JSON**.

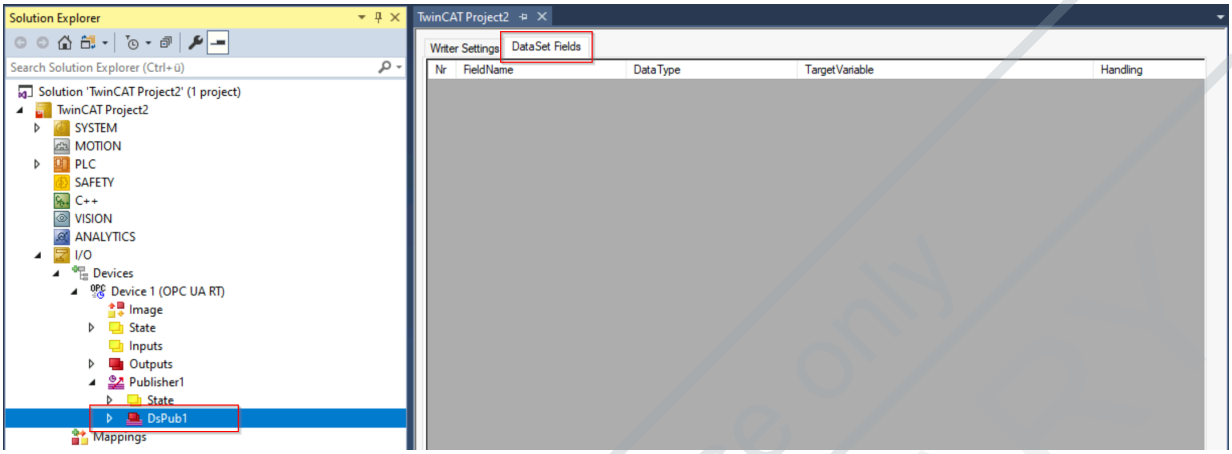


10. In the next step you want to configure a so-called Dataset in order to define variables that should be send out by the Publisher. Right-click the **Publisher** node and select **Add new element....**

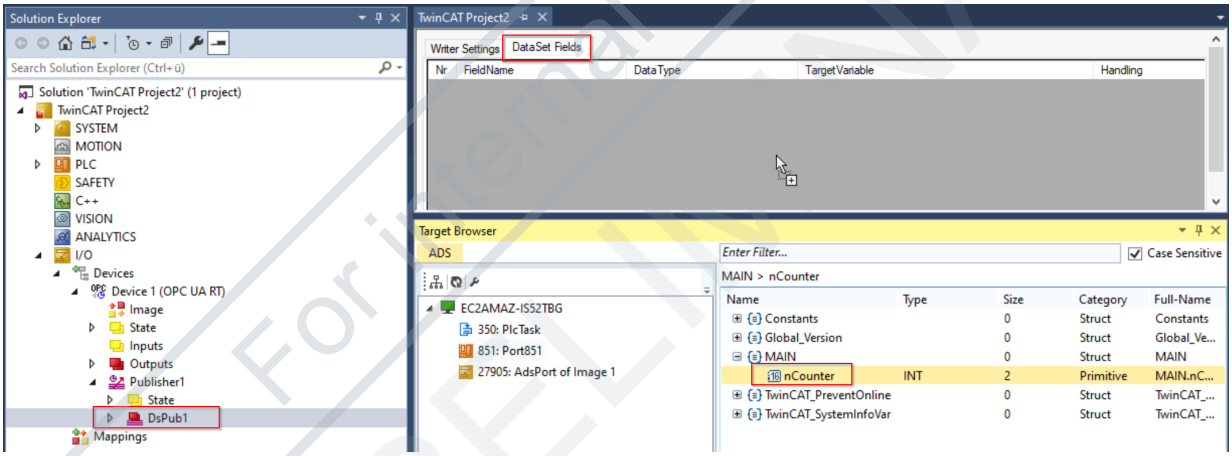
11. Select the entry **Published DataSet (Module)** and click on **Ok**.



12. Double-click on the data set and open the **Data Set fields** tab.



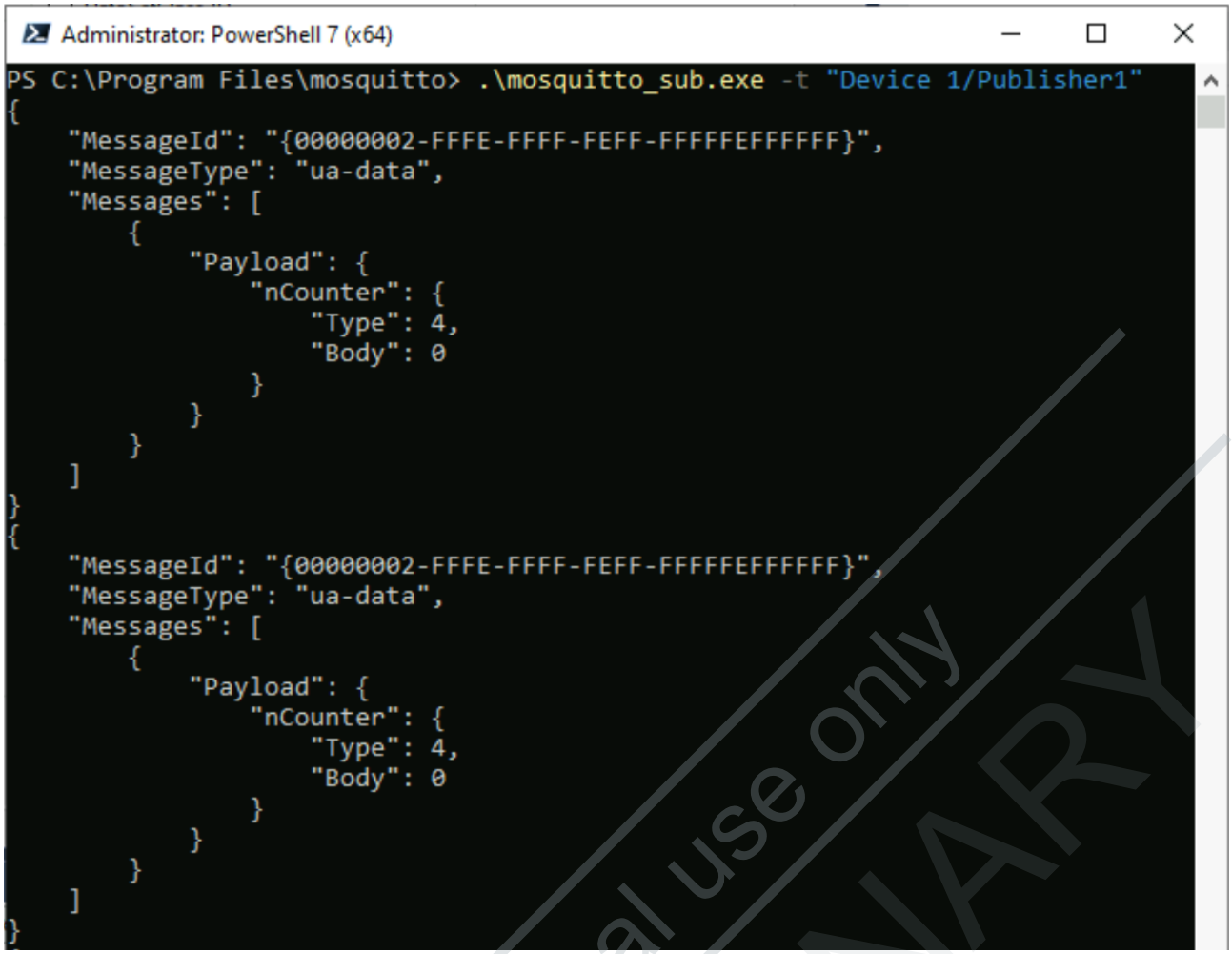
13. Open the TwinCAT Target Browser, navigate to the MAIN program of your PLC Project and drag&drop the variable nCounter to the list of data set fields.



Activating the project

Activate the TwinCAT project by clicking the **Activate** button on the TwinCAT XAE toolbar. Make sure that you have selected the correct target device (we assume that you are using your local device) and that you are using the correct Real-Time settings so that you can activate the TwinCAT Runtime on your device. Please consult the regular TwinCAT documentation for more information about how to activate a TwinCAT configuration and switch TwinCAT into Run mode.

In this quick start tutorial, we have used MQTT as the transport protocol. You can now use any MQTT client that has access to the message broker to subscribe to the published data set. The following screenshot shows the command line tool **mosquitto_sub.exe** that has been subscribed to the data. This tool is part of the Mosquitto software application.



4.2 Supported features

OPC UA Pub/Sub comes with a large set of features. This product may not support every feature right from the beginning, more features and functionalities will be added over time as product updates. The following table shows a list of all supported functionalities at the moment.

Header layouts

Name	Supported
Flexible	Yes
UADP Dynamic	Yes
UADP PeriodicFixed	Yes
JSON DataSetMessage	Yes
JSON Minimal	Yes
JSON NetworkMessage	Yes

UDP transport

Feature	Supported
Publisher	Yes
Subscriber	Yes
UADP chunking	Yes
IP fragmentation	Yes
DataKey-Frames	Yes
Delta-Frames	Yes
KeepAlive	Yes
Message timeout handling	Yes
UADP raw encoding	Yes
UADP variant encoding	Yes
Pub/Sub security message signing	Not supported yet
Pub/Sub security message encryption	Not supported yet
Support for primitive data types (INT, REAL, ...)	Yes
Support for arrays of primitive data types (INT, REAL, ...)	Yes
Support for data structures	Yes
Support for arrays of data structures	Yes
Support for Unions	Not supported yet
Support for Variants	Not supported yet
Support for local data sets	Yes
Support for pre-configured (global) data sets	Yes
Support for unicast/multicast	Yes
Support for configuration of UDP port	Yes
Support for event messages	Not supported yet
Support for field-specific data in data sets (Delta-Deadband, Deadband-Type, Substitute Values)	Not supported yet
Support for Ethernet VLAN transport	Not supported yet
Support for MetaData services	Not supported yet

MQTT transport

Feature	Supported
Publisher	Yes
Subscriber	Yes
Key-Frames	Yes
Delta-Frames UADP	Yes
Delta-Frames JSON	Not supported yet
KeepAlive	Not supported yet
Support for MQTT version 3.1.1	Yes
Support for MQTT version 5.0	Yes
Support for MQTT / TLS versions 1.2 and 1.3	Yes
JSON encoding (reversible)	Yes
JSON encoding (non-reversible)	Yes
UADP encoding	Yes
Support for event messages	Not supported yet
Support for primitive data types (INT, REAL, ...)	Yes
Support for arrays of primitive data types (INT, REAL, ...)	Yes
Support for data structures	Yes
Support for arrays of data structures	Yes
Support for Unions	Not supported yet
Support for Variants	Not supported yet
Support for local data sets	Yes
Support for global data sets	Yes
Support for MetaData services	Not supported yet

Supported message brokers

Our TwinCAT MQTT protocol driver supports MQTT versions 3.1.1 and 5.0. Therefore, all message brokers are supported that also support these versions. We have successfully tested communication with Mosquitto, HiveMQ and AWS IoT Core.

OPC UA Pub/Sub Security

Feature	Supported
Pre-Shared-Key (PSK)	Not supported yet
Local SKS	Not supported yet
Remote SKS	Not supported yet

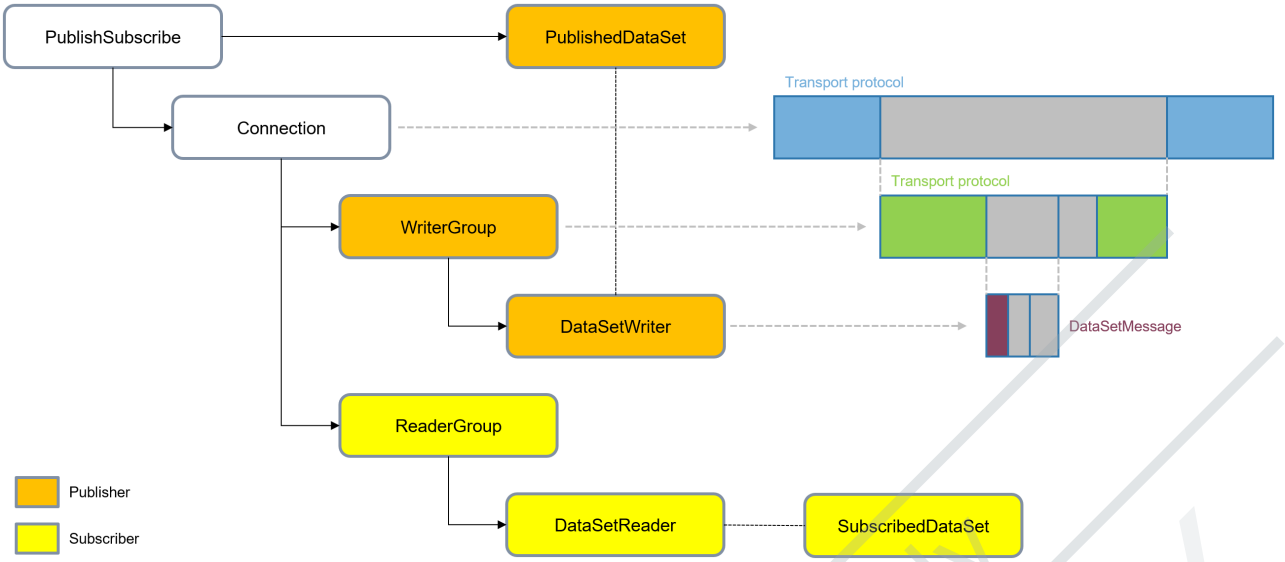
Configuration handling

Feature	Supported
Offline configuration exchange (UA-Binary)	Yes
Online configuration exchange (read)	Not supported yet
Online configuration exchange (read)	Not supported yet

4.3 Protocol overview

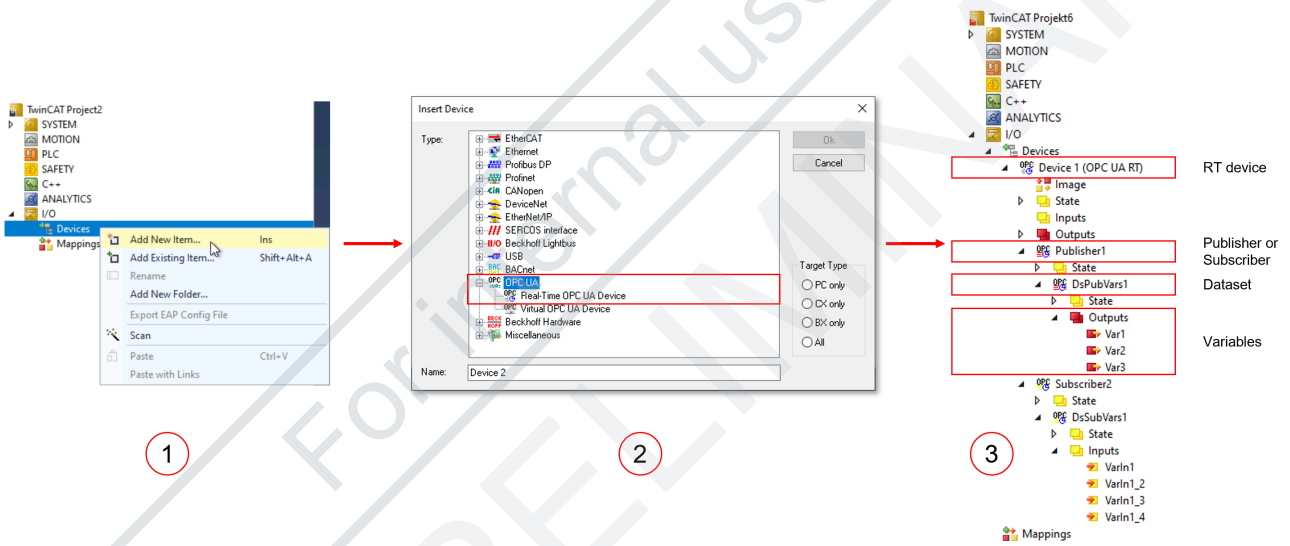
OPC UA Pub/Sub defines different configuration parameters for the various components. They define the behavior of Publisher and Subscriber. Configuration of these parameters can be performed through the OPC UA Information Model for PubSub or through vendor-specific mechanisms, which is, in case of the product TF6105 TC3 OPC UA Pub/Sub, the TwinCAT XAE environment.

The following diagram depicts the different configuration components and their relation to each other. The WriterGroup, DataSetWriter and PublishedDataSet components define the publisher whereas the ReaderGroup, DataSetReader and SubscribedDataSet define the subscriber.



For more detailed information about the individual components, we recommend to consult the OPC UA specification, part 14 (Pub/Sub).

When using the TwinCAT XAE environment to set up the configuration, the engineering workflow can be visualized as follows:



Step	Description
1	A new device is added to the TwinCAT I/O configuration.
2	From the list of devices, you can then select the “Real-Time OPC UA Device”, which will be the entry point for all OPC UA Pub/Sub settings.
3	Publishers, subscribers, datasets and variables are added as well as their corresponding parameters.

The following chapters provide an overview of the different components and link to more detailed documentation articles.

Device

The OPC UA RT (Realtime) Device is the entry point of the OPC UA Pub/Sub configuration and will be linked to a network interface card based on the TwinCAT Realtime-Ethernet Adapter driver. The device may contain one or more publishers and/or subscribers as well as (global) data sets.

Publisher/Subscriber

A Publisher or Subscriber node defines whether the connected data sets are either send (published) or received (subscribed). Each Publisher/Subscriber [▶ 25] node contains address information for the transport protocol [▶ 31] that should be used.

Data set

Data sets can be specified in two different scopes. You can either define a data set locally on a Publisher or Subscriber or globally on the device. Global data sets can be shared by multiple Publishers or Subscribers. There are two different types of data sets which define the way how variables are created and handled:

- Data set with variables: this data set type can be configured with variables that appear in the process image of the data set. The variables can then be linked to other process image variables.
- Data set without variables: this data set type can be configured with variables via the TwinCAT Target Browser. The variables are then linked internally automatically and do not appear on the process image.

For more information, please visit our documentation article about data sets [▶ 27].

Variables

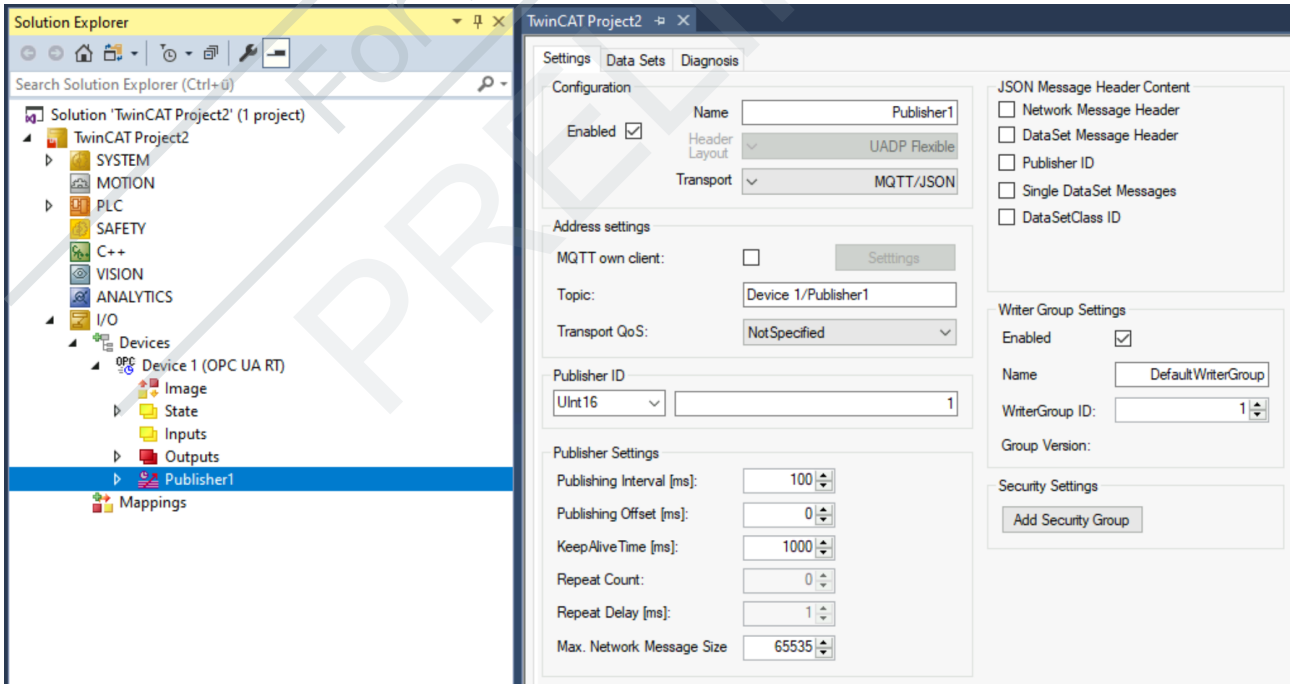
Variables are added to the data set as so-called “data set fields”. They can either be linked to other process image variables or are connected directly to other variables via the TwinCAT Target Browser. See our documentation article about data sets [▶ 27] for more information.

4.4 Publisher/Subscriber

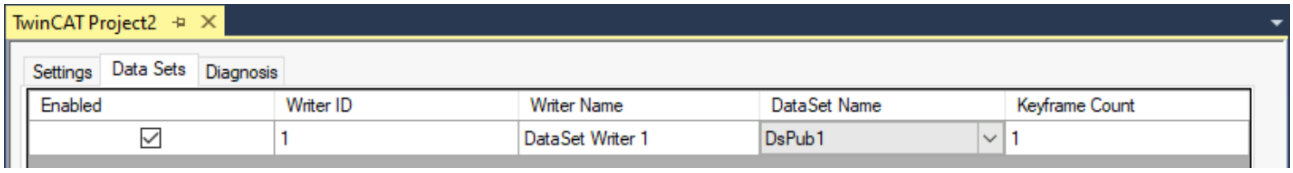
OPC UA Pub/Sub is based on the publisher/subscriber communication pattern. TF6105 TC3 OPC UA Pub/Sub includes different configuration mechanisms to set up a publisher or subscriber including their corresponding data sets [▶ 27]. We also recommend to consult our protocol overview [▶ 23] chapter for an overview of the different configuration components.

Publisher

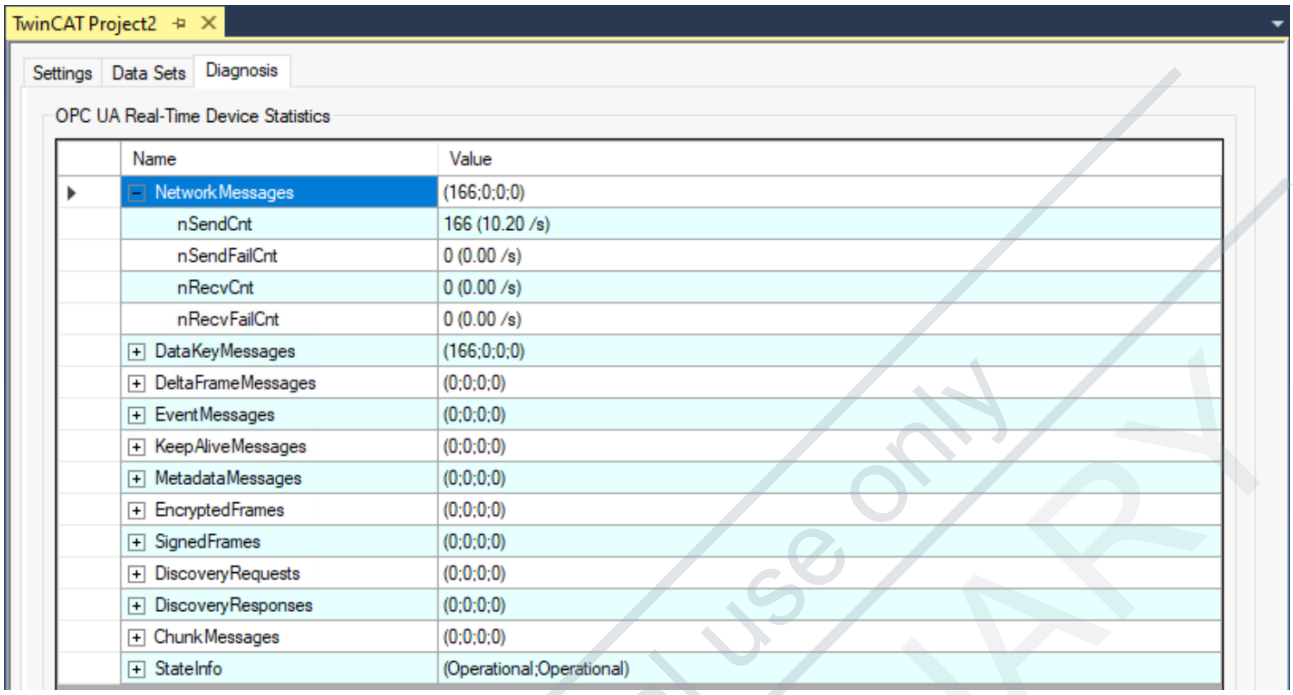
The publisher can be used to set up communication for sending data. It is added directly to the OPC UA RT Device. The settings page includes different configuration parameters to define the transport protocol [▶ 31] as well as additional settings that further define the data communication.



The **Data Sets** tab contains information about all linked local or global data sets.

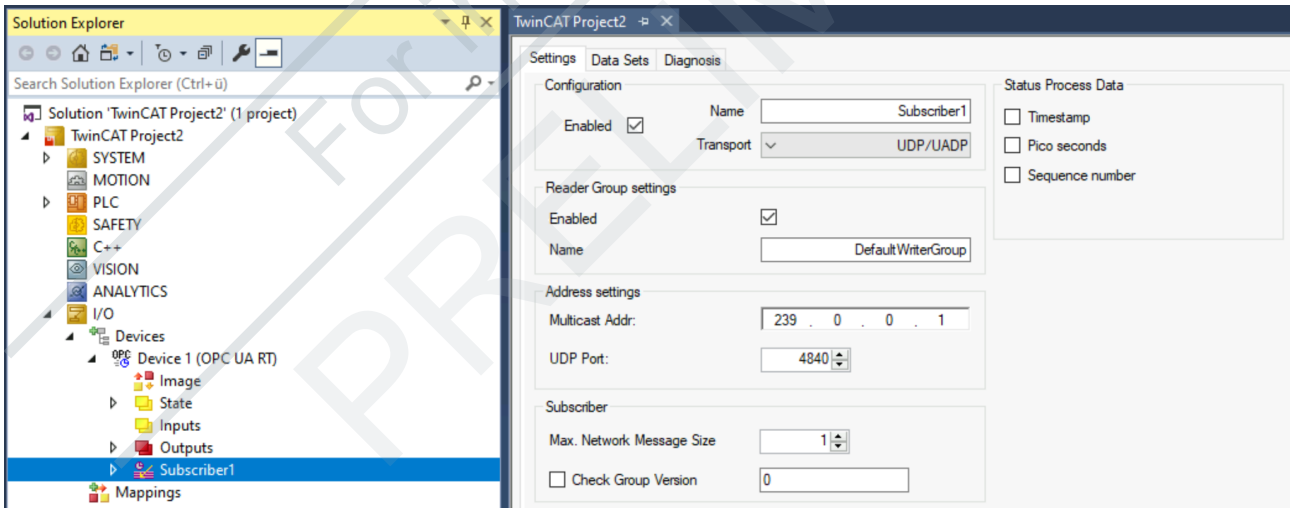


The **Diagnosis** tab contains diagnostics information about the data communication.



Subscriber

The publisher can be used to set up communication for receiving data. It is added directly to the OPC UA RT Device. The settings page includes different configuration parameters to define the transport protocol [▶ 31] as well as additional settings that further define the data communication.



Similarly to the publisher, the **Data Sets** tab contains information about all linked local or global data sets and the **Diagnosis** tab contains diagnostics information about the data communication.

4.5 Data sets

Data sets can be added to a publisher and subscriber and define a list of name and value pairs that represent a list of variable values. Such variables are then represented by so-called data set fields. Meta data can be added to a data set, which provides additional information about the structure and content of a data set.

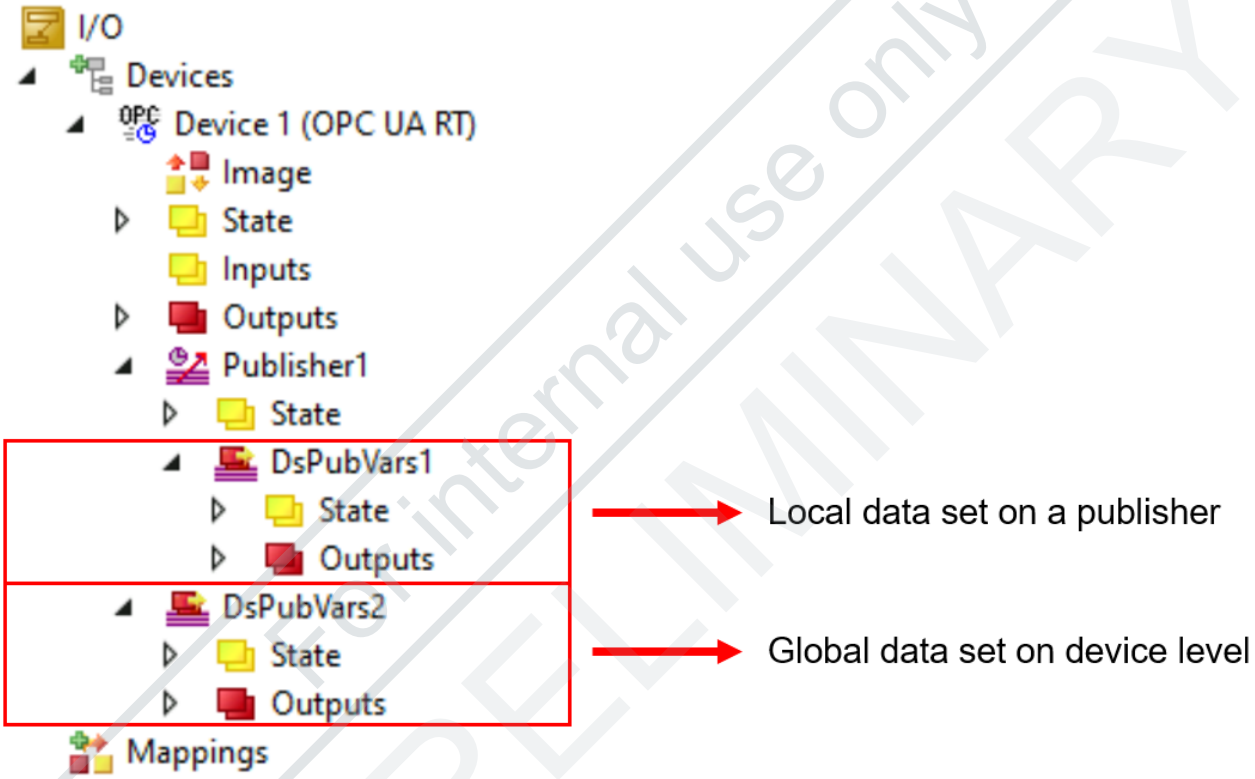
TF6105 TC3 OPC UA Pub/Sub provides two different types of data sets, which define how variables are configured and handled.

- Data set with variables
- Data set without variables

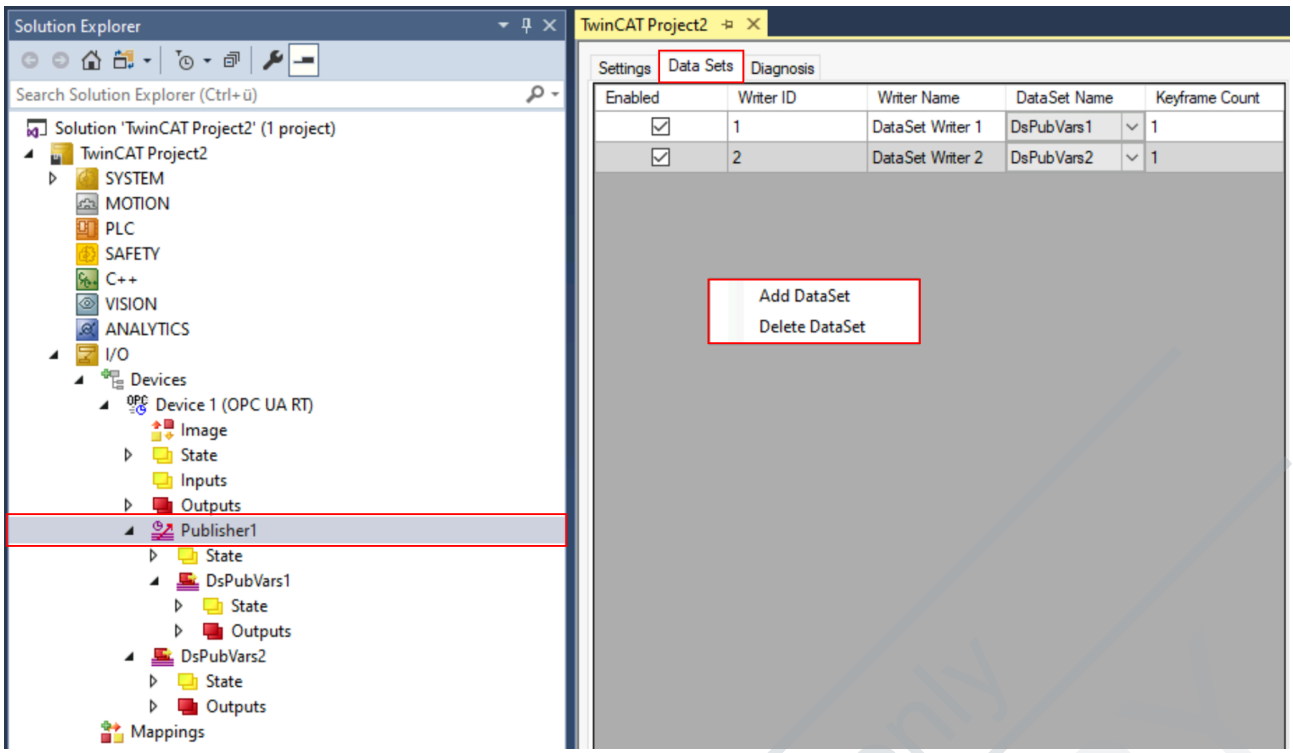
These data set types are described in more detail below.

Scope

Data sets can be configured in two different scopes: locally or globally. Local data sets are configured exclusively for a single publisher or subscriber whereas global data sets are added on device level and can be shared by publishers or subscriber, e.g. if you want to publish the same data points on two different publishers.



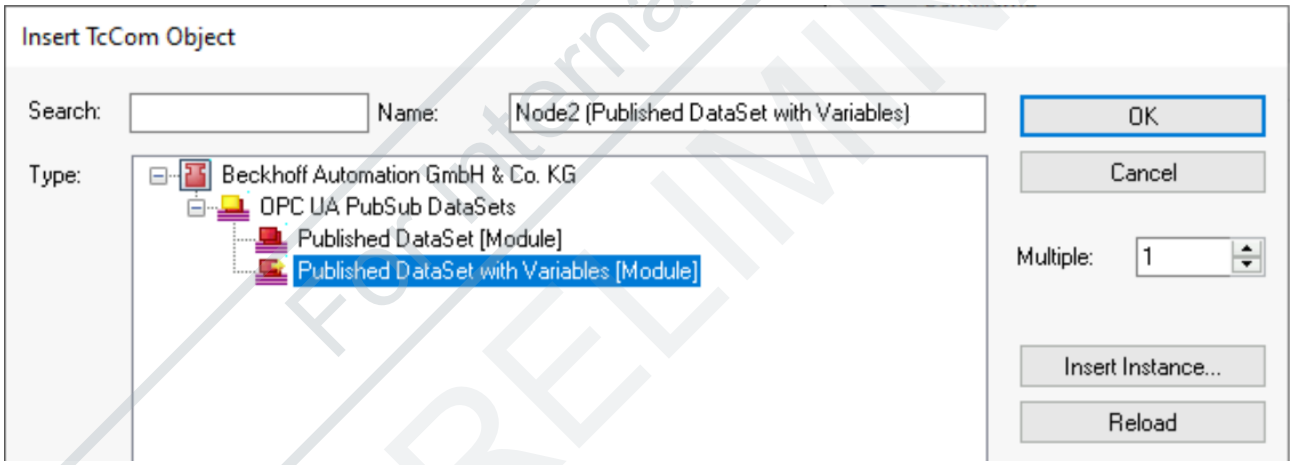
Global data sets can be added to a publisher/subscriber via the **Data Sets** tab:



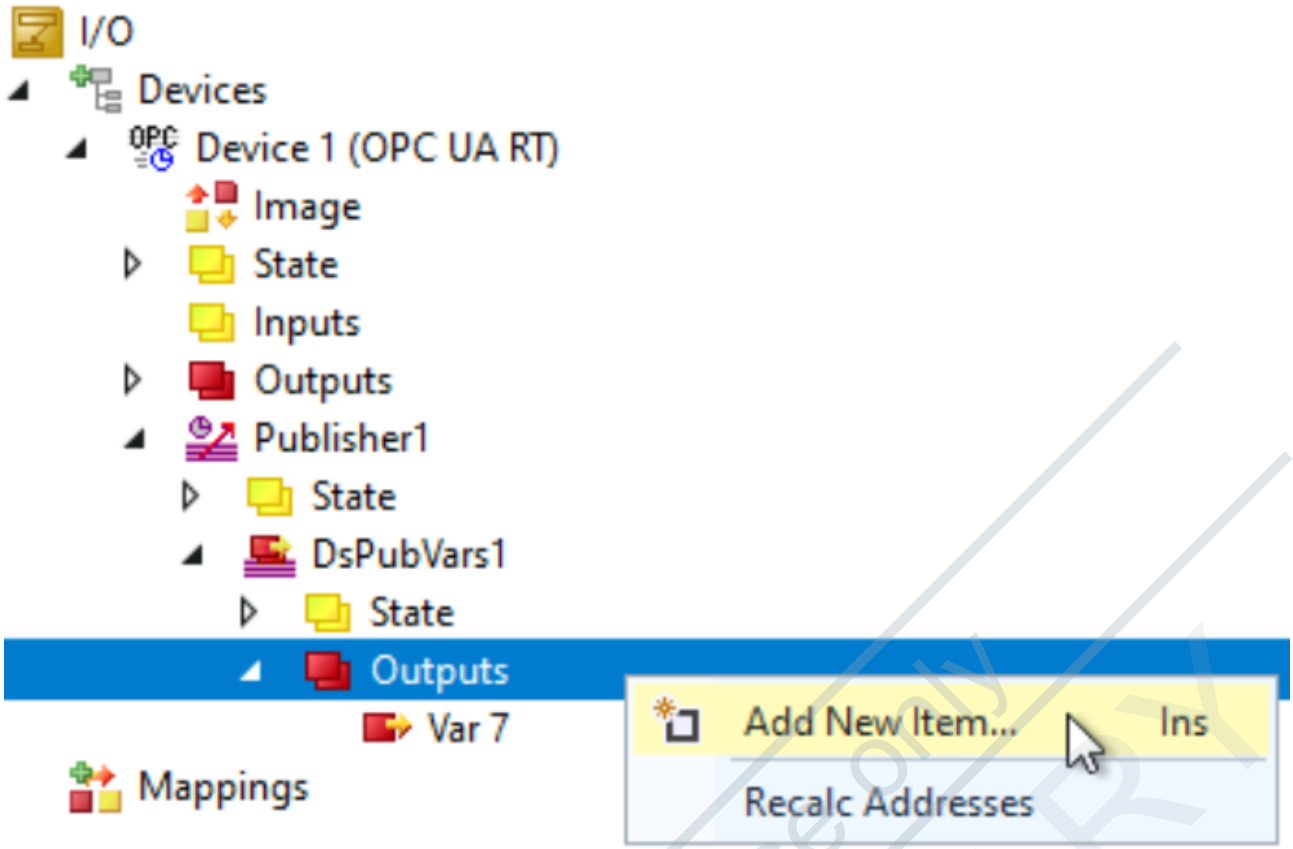
In the example above the publisher has been configured with a local data set called “DsPubVars1” and a global set called “DsPubVars2”.

Data set with variables

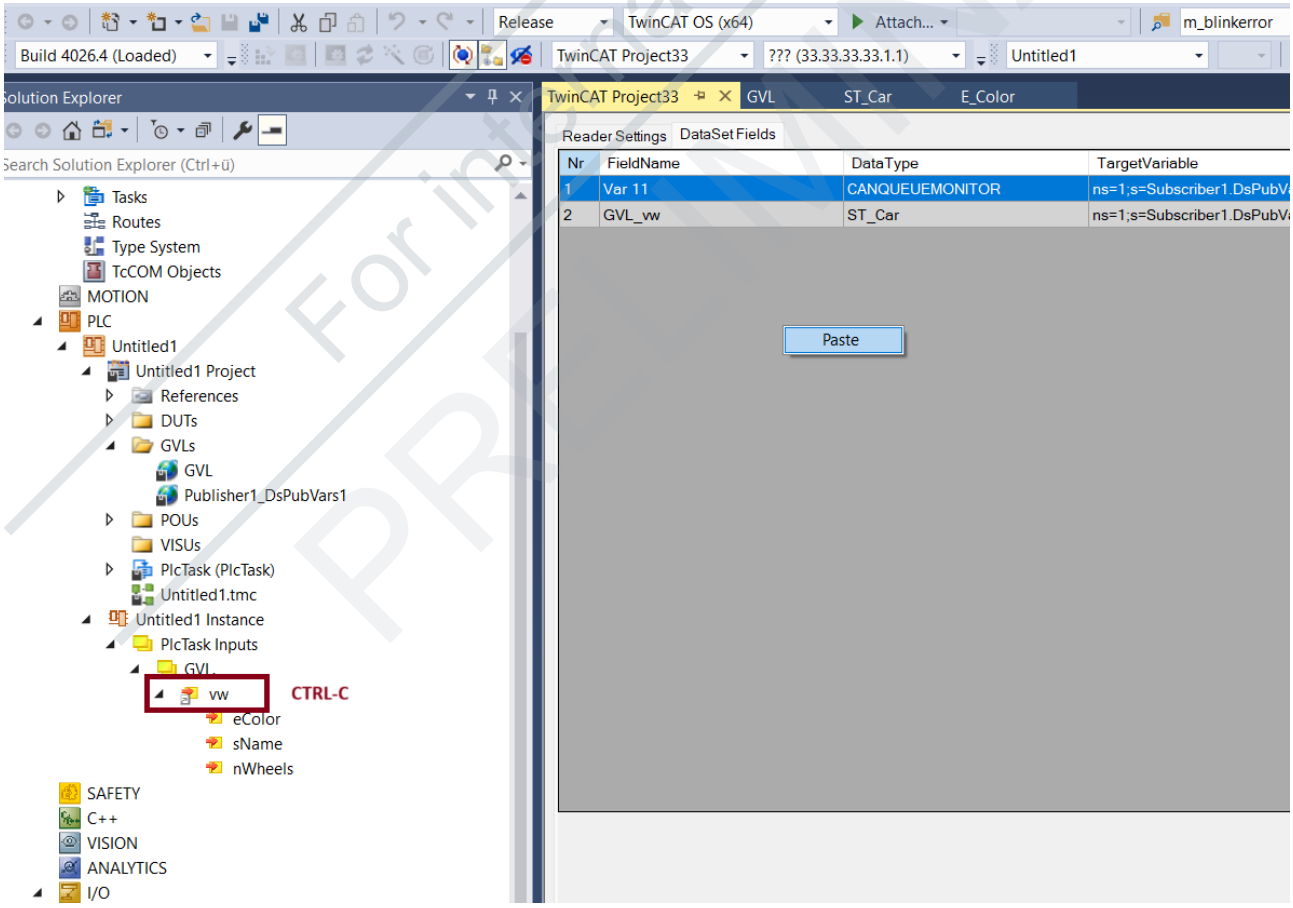
Data sets with variables configure all variables as part of the process image. These data sets can be added either globally or locally, e.g. on a publisher:



The variables can then be added directly on the process image of the data set:



As an alternative, you can also use Copy&Paste on a variable from the PLC process image to add it to the data set, as shown in the following screenshot. Simply use STRG+C to copy the variable to the clipboard and then paste it to the target data set via the context menu.

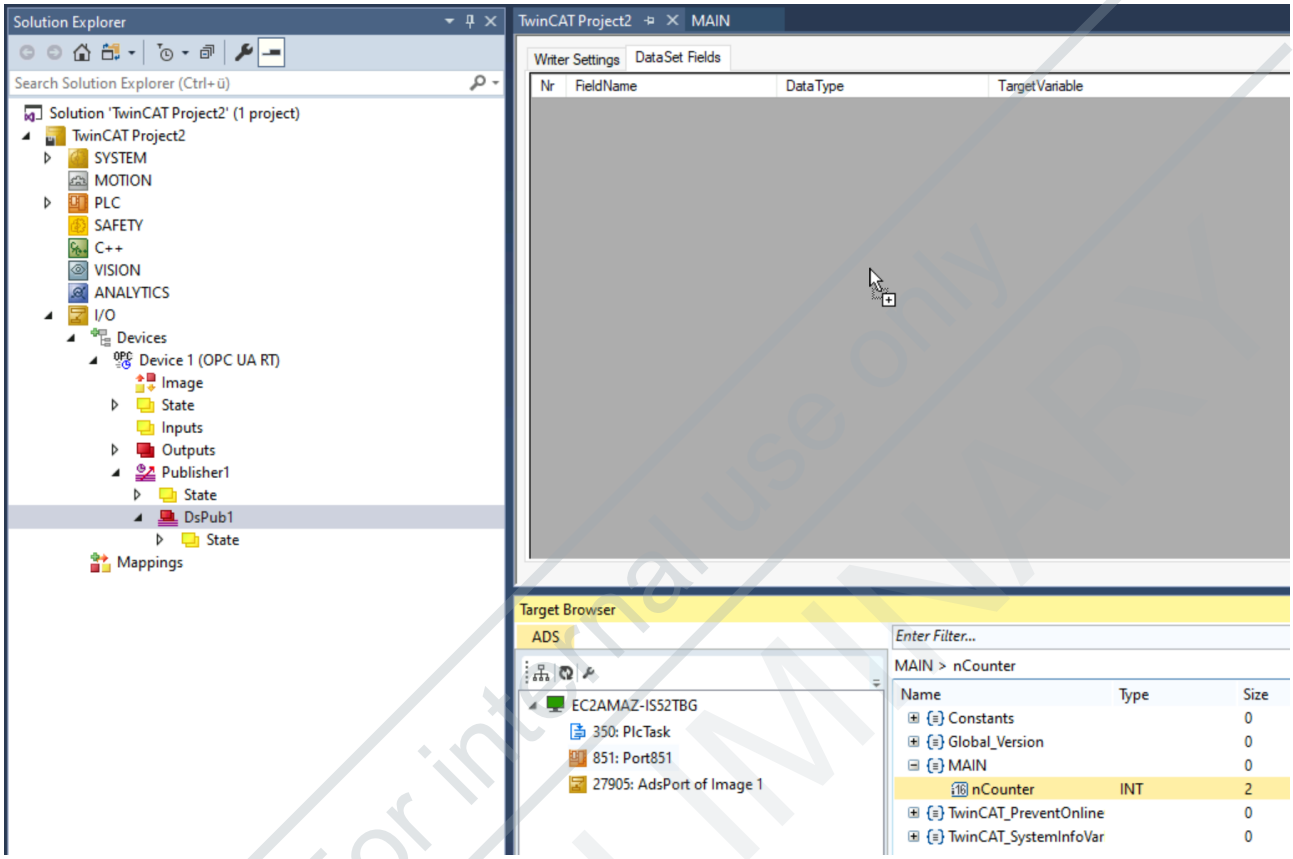


Data set without variables

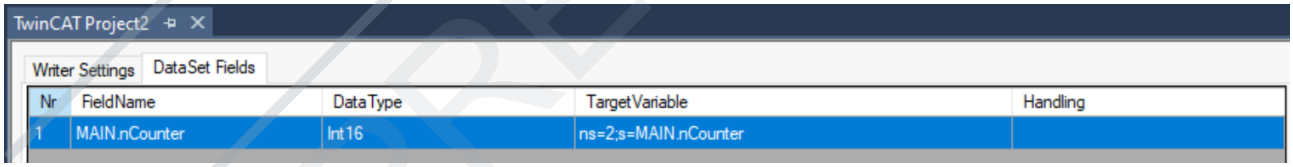
Data sets without variables do not configure the variables as part of the process image. Instead, variables can be added to the data set either via the TwinCAT Target Browser or directly from a PLC process image. The variables are then automatically linked with the corresponding symbol.

> **Changes to the symbol**
>
> Please note that, if the linked symbol changes, the variables need to be removed from the data set and added again via the TwinCAT Target Browser because the symbolic information is retrieved as part of the Drag&Drop process.

To add a variable to this kind of data set, please open the TwinCAT Target Browser and drag a variable to the **Data Set Fields** tab of the data set, as shown in the following screenshot.

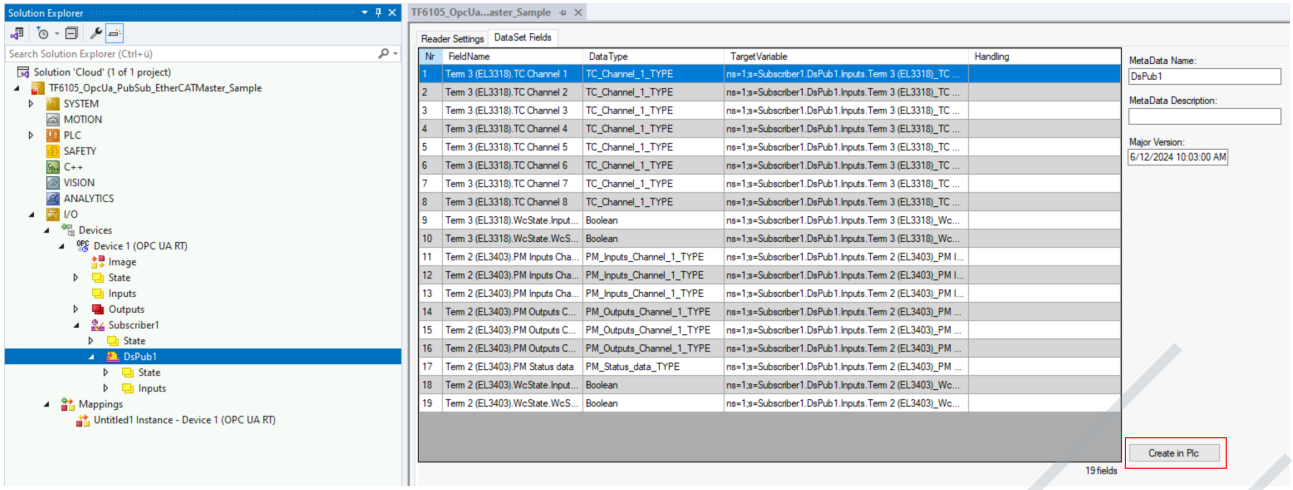


The field configuration of the data set now includes the variable and it is published as soon as the configuration is activated.

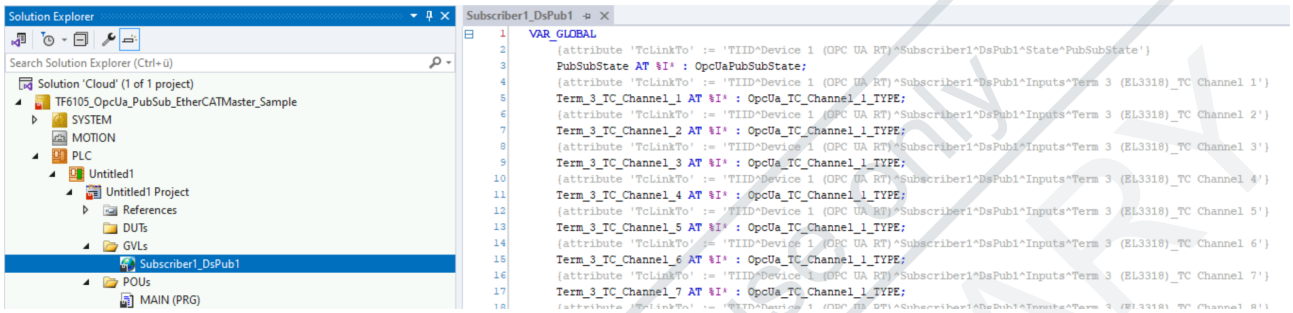


Code generation

Subscriber data sets include an automatic code generation to reduce the engineering time required to create and link matching PLC variables. The code generation is accessible via the corresponding button on the subscriber data set.



This functionality generates a global variable list that contains all the variables from the subscriber data set with matching data types and the attribute “TcLinkTo” that automatically links them to their counterparts.



4.6 Transport protocols

The OPC UA Pub/Sub extension defines different transport protocols over which data can be transmitted. Those transport protocols include:

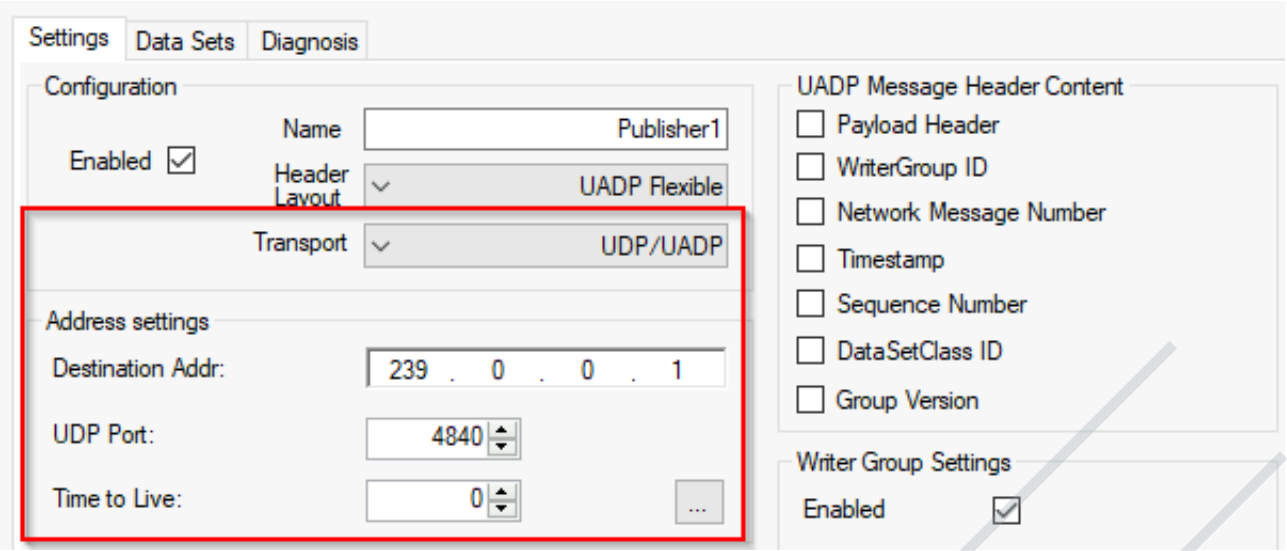
- UDP for fast machine-to-machine communication
- MQTT for machine-to-cloud communication

Each protocol has its own advantages and disadvantages and it depends on the application project which transport is the right one to use. Although MQTT is considered an "IoT" protocol and is often used together with cloud services, it is of course not restricted to cloud-related use cases because MQTT connectivity can also be set up within a private network.

The product TF6105 TC3 OPC UA Pub/Sub supports both transport protocols. The following chapters describe the individual configuration parameters for each transport protocol.

4.6.1 UDP

To configure a Publisher or Subscriber for UDP transport, simply open the Publisher/Subscriber configuration page and select the UDP transport. UDP supports the (binary) UADP encoding and different header layouts [▶ 34]. Different address settings are available to configure connection details, e.g. the destination address, UDP port and time-to-live.

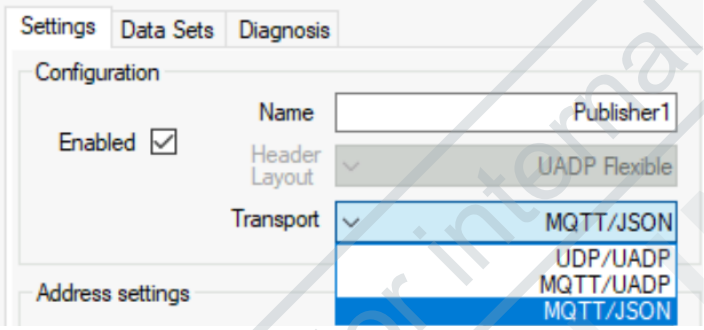


4.6.2 MQTT

To configure a Publisher or Subscriber to use MQTT transport, simply open the Publisher/Subscriber configuration page and select the MQTT transport. MQTT supports the (binary) UADP encoding as well as JSON and may be configured with a specified header layout [▶ 34].

Encoding

There are two encodings available for selection: MQTT/UADP and MQTT/JSON. The first data format uses the binary UADP encoding to encode messages and the last one uses a JSON format.



Depending on the selected header layout, the MQTT/JSON encoding may look as follows:

```
{
  "MessageId": "{0CF7F27A-56FF-1F85-CFC2-12C886365C1D}",
  "MessageType": "ua-data",
  "Messages": [
    {
      "DataSetWriterId": 1,
      "Payload": {
        "nCounter": {
          "Type": 4,
          "Body": 18
        },
        "bToogle": {
          "Type": 1,
          "Body": false
        }
      }
    },
    {
      "DataSetWriterId": 2,
      "Payload": {
        "myNewVariable": {
          "Type": 4,
          "Body": 42
        }
      }
```

```
    }
  ]
}
```

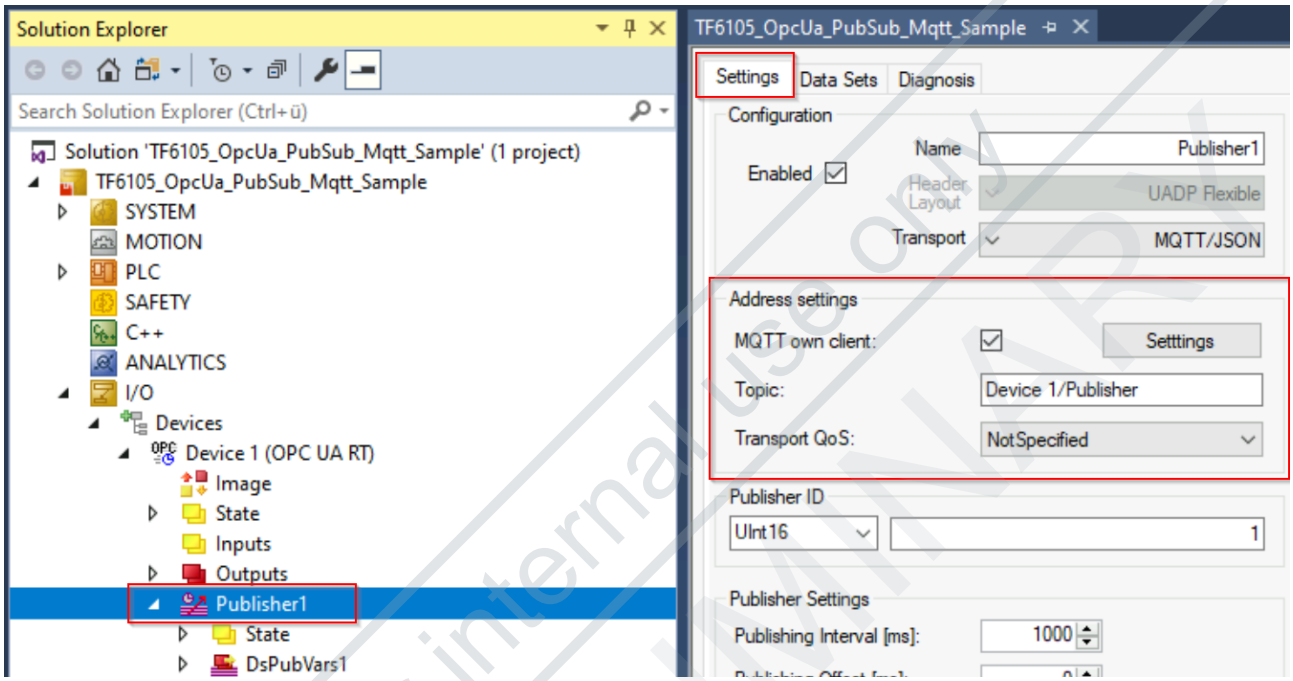
The JSON format is standardized within the OPC UA Pub/Sub specification. For a full description of the JSON scheme and each individual field, please consult the specification.

Connection parameters

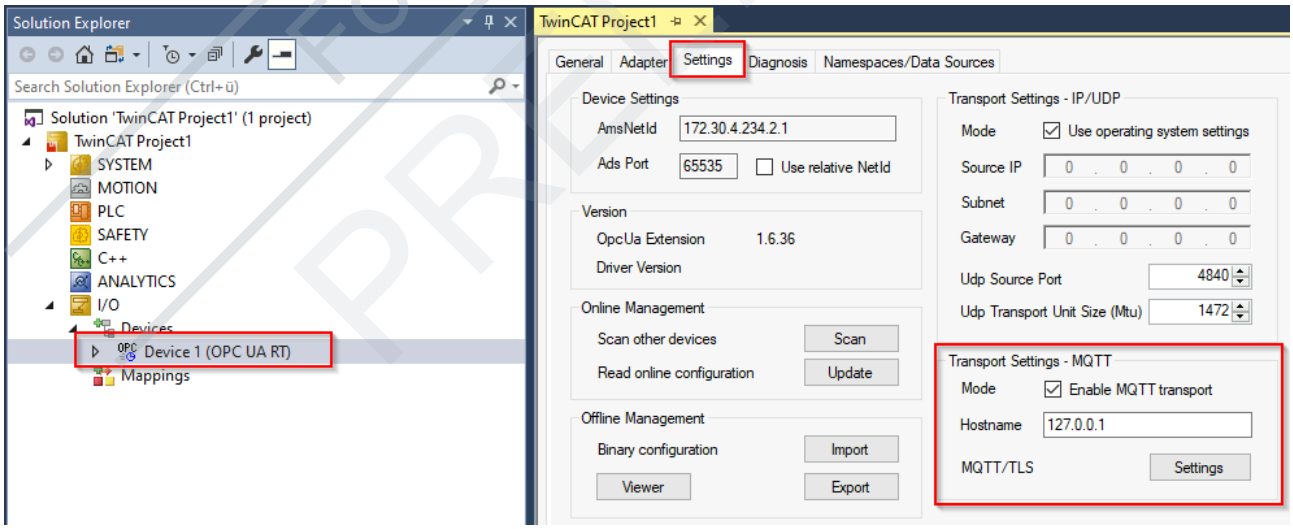
Whenever using MQTT as the transport protocol, you also need to define connection parameters to the MQTT Message Broker. These parameters can be specified, depending on the use case, in two different locations:

- you can either configure individual MQTT connections on each Publisher or Subscriber
- or you can configure one “global” MQTT connection on the OPC UA RT device settings. In this case, all Publishers and Subscribers share one connection to the MQTT Message Broker.

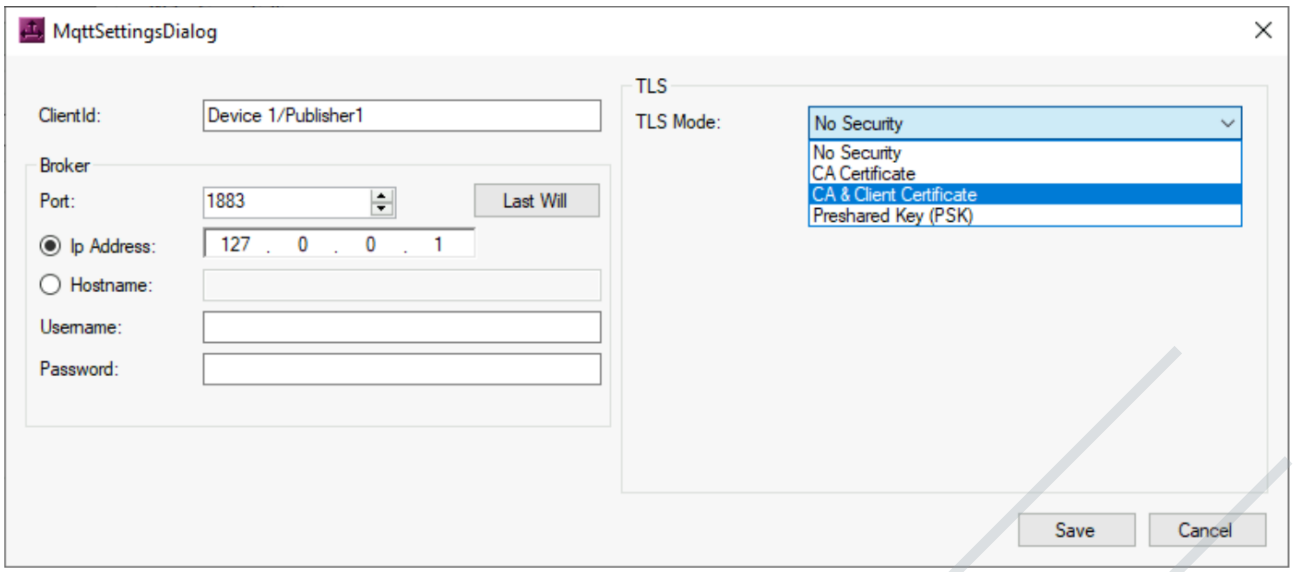
The following screenshot shows the MQTT connection parameters on a Publisher:



The following screenshot shows where to set up the global MQTT connection on the OPC UA RT device:



In both cases, the dialog that allows to enter the connection parameters is the same and looks as follows:



Topics

Data on the MQTT Message Broker is organized in so-called “topics”. From an OPC UA Pub/Sub point-of-view, you can specify topics on two layers:

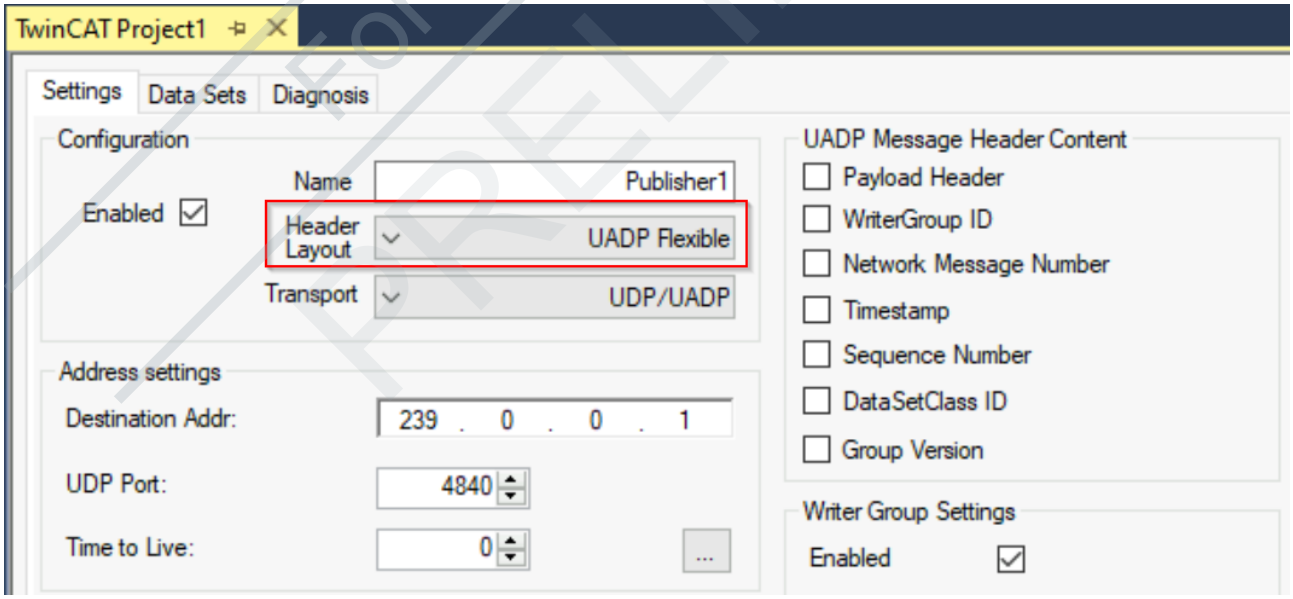
- The topic on a Publisher is configured on the Publisher settings.
- The topic on a Subscriber is configured individually for each DataSet.

4.7 Header layouts

Header content and message layouts were designed to be flexible and to support different use cases by enabling or disabling individual fields within the headers. While this flexibility makes it possible to support many different use cases with Pub/Sub, the number of possible header field combinations also increases the effort needed for the implementation and verification. On the other hand, within a given application domain or for different use cases some configurations might be more appropriate than others.

Custom header layout configurations (also called the “Flexible” header layout) can be used but they should be limited to applications that do not fall into the use cases for the layouts.

The header layout can be configured on a publisher or subscriber, as shown in the following screenshot.



The following header layouts are supported. The header layout names are derived from the OPC UA specification, part 14, annex A.

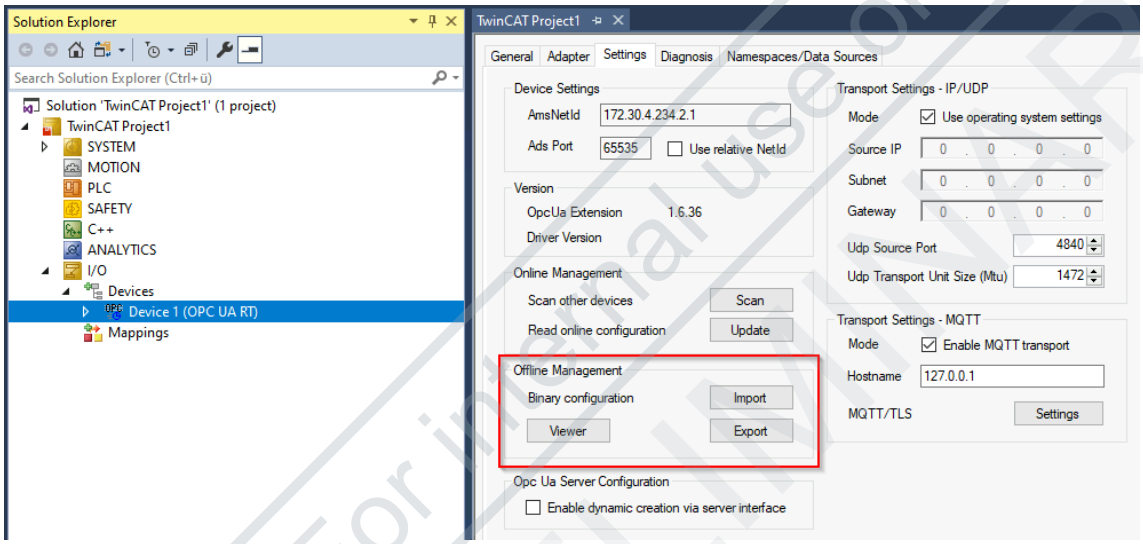
Name	Supported
Flexible	Yes
UADP Dynamic	Yes
UADP PeriodicFixed	Yes
JSON DataSetMessage	Yes
JSON Minimal	Yes
JSON NetworkMessage	Yes

4.8 Configuration import/export

Exchanging configurations can be a very important step during the engineering phase of a project, especially if a Pub/Sub configuration needs to be exchanged between different devices and/or vendors. OPC UA Pub/Sub includes a standardized, file-based format to exchange configurations known as the “UA-Binary import/export” or simply “Configuration import/export”.. The workflow is usually as follows:

- Set up your publishers on device A.
- Export the configuration file.
- Import the configuration file on the other device(s) to set up a corresponding subscriber.

A configuration dialog on the OPC UA RT device in TwinCAT XAE allows to access this import/export functionality.



Changes to the configuration

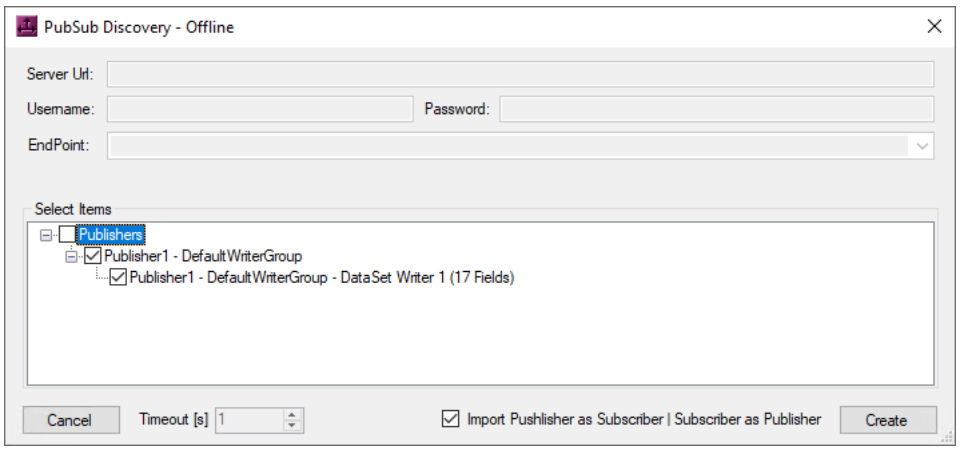
Please note that every change to the configuration also requires to create a new configuration file.

Export

When exporting a configuration file, a dialog is opened that allows to select the destination that the file should be saved to. You can then take the created file and share it with the corresponding party.

Import

When importing a configuration file, a dialog is opened that allows to select the publishers/subscribers that should be imported from the file. The default behavior is that a publisher is imported as a subscriber and vice versa.



MQTT connection settings

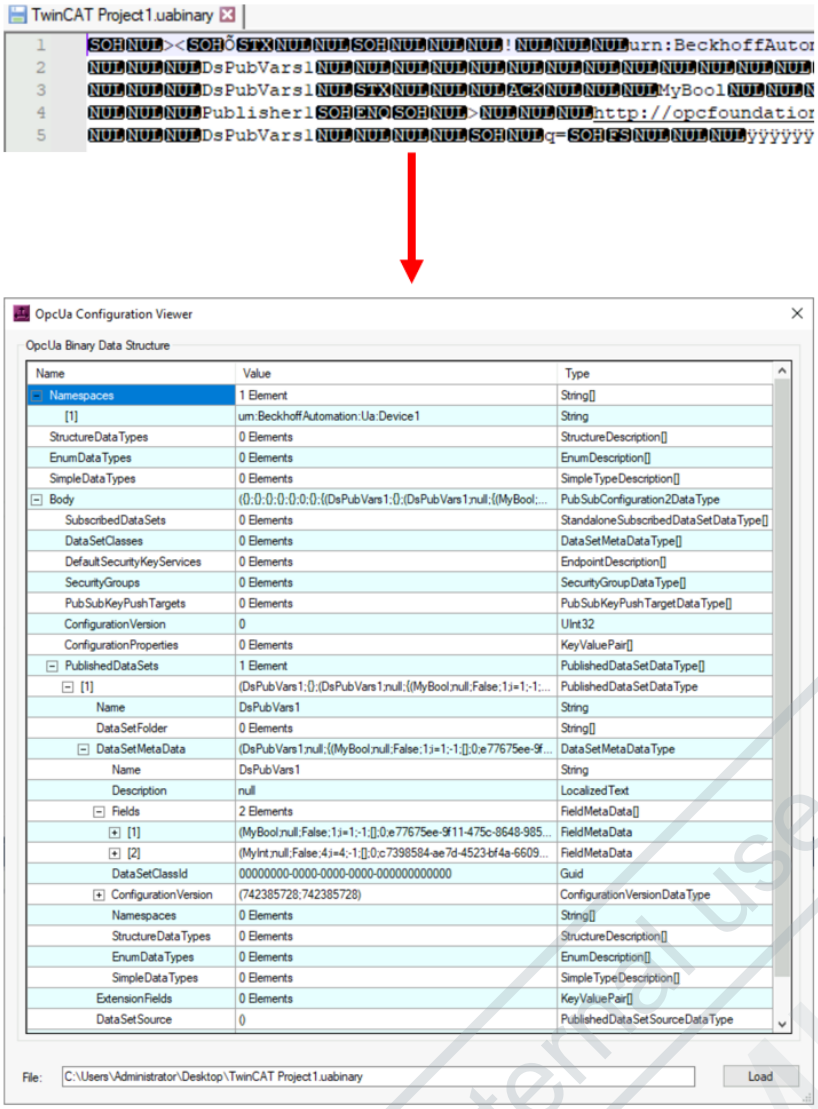
When importing a configuration file that contains MQTT connection settings, the connection details are imported as follows:

If the MQTT connection settings are the same for all configuration items, the settings are imported and added on the OPC UA RT Device.

If the MQTT connection settings are different on some configuration items, the settings are imported on the publisher/subscriber nodes.

Configuration viewer

The same settings area also includes a so-called “configuration viewer”, which allows to preview the contents of the binary Pub/Sub configuration file.

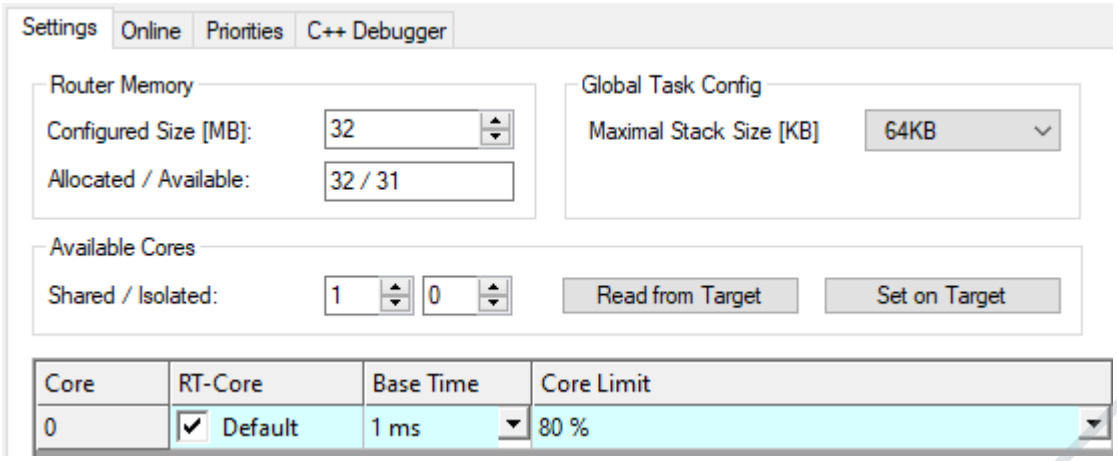


4.9 In-depth

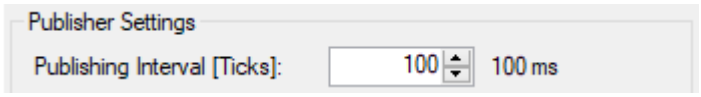
4.9.1 Publishing Interval

The PublishingInterval is configured as part of a Publisher configuration. It specifies the time intervall in which frames should be published and correlates with the BaseTime that has been configured in the TwinCAT Real-Time settings.

Example: the following screenshot shows a system which has a BaseTime of 1ms configured in its TwinCAT Real-Time settings.



When configuring a Publisher, the PublishingInterval now correlates to this BaseTime. The following screenshot shows a configured PublishingInterval of 100ms (100 ticks * 1ms BaseTime).

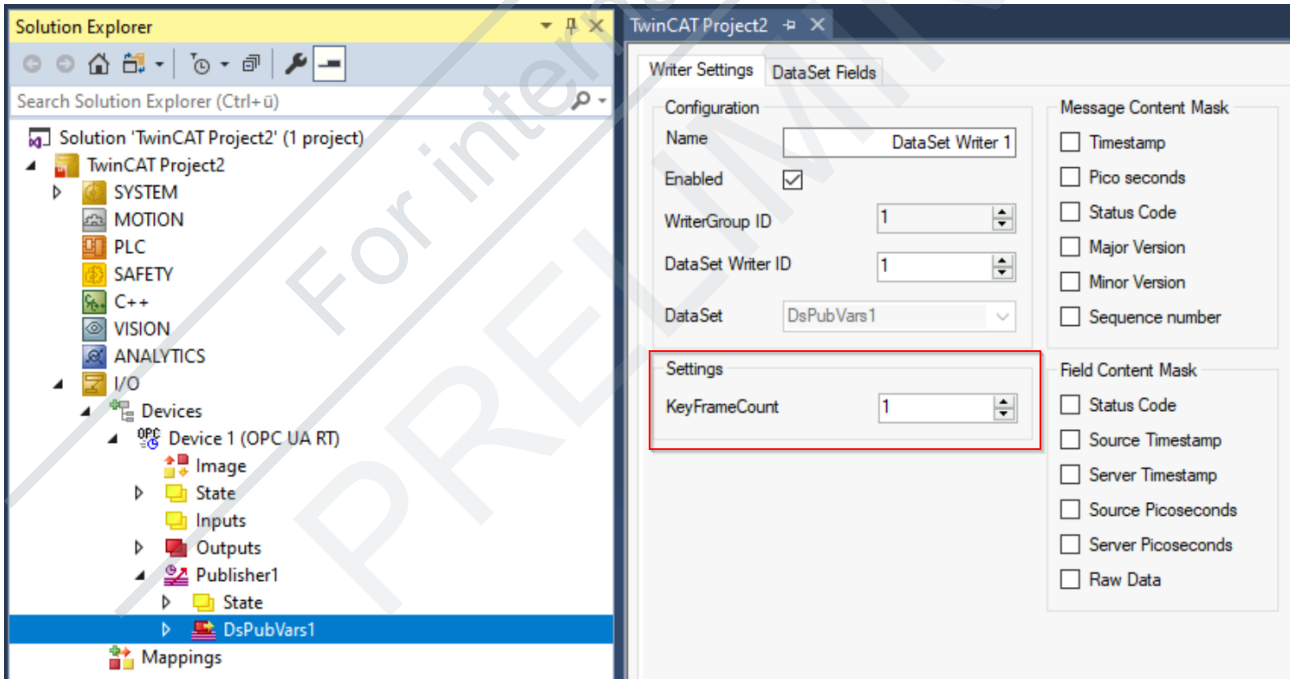


4.9.2 KeyFrames, DeltaFrames, KeepAlive

This documentation article explains how KeyFrames, DeltaFrames and KeepAlive correlate with each other when configured on a publisher or publisher data set.

KeyFrames and DeltaFrames

KeyFrames always contain the full data set whereas DeltaFrames only include incremental updates (fields that have changed). When configuring a Publisher data set, the KeyFrameCount specifies how often a KeyFrame should be published related to the publishing interval.



When set to "1", a KeyFrame is published in every Publisher cycle. When set to a value greater than "1", a KeyFrame is published every n cycles, otherwise a DeltaFrame is published. Please note that Delta-Frames are only published if there is a data change, which means that there may be times in which no Frames are published at all.

KeepAlive

The KeepAlive time, which is configured on a publisher, is used to send out frames if there are no data changes (and therefore delta frames) that are send out for a longer period of time. This ensures that any subscribers still know that the publisher is online. Please note that, when configuring the KeepAlive on a subscriber, that this also correlates with the MessageReceiveTimeout.

4.9.3 MessageReceiveTimeout

The MessageReceiveTimeout is configured as part of a Subscriber configuration. It specifies the time after which an incoming message for a subscriber DataSet is to be expected - otherwise a timeout occurs and the PubSubState process variable goes into an error. The MessageReceiveTimeout may be reset by the KeepAlive [▶ 38].

5 Samples

Sample code and configurations for this product can be obtained from the corresponding repository on GitHub: https://github.com/Beckhoff/TF6105_Samples. There you have the option to clone the repository or download a ZIP file containing the sample. The following samples are currently available:

Sample	Description
UDP [▶ 40]	Demonstrates how to set up a sample publisher and subscriber based on OPC UA Pub/Sub UDP.
MQTT [▶ 41]	Demonstrates how to set up a sample publisher and subscriber based on OPC UA Pub/Sub MQTT.
Publish data from EtherCAT Master	Explains how to set up an EtherCAT Master ADS Symbol Server and configure an OPC UA Publisher to publish its data points. This sample is not available as a download but all required steps will be explained in this documentation article.

We also recommend to check out our quick start tutorials [▶ 13], which provide a guided walkthrough of various configuration use cases.

5.1 UDP

This sample demonstrates how to set up a publisher and subscriber based on OPC UA Pub/Sub UDP. The sample solution contains a TwinCAT project that includes the following PLC configuration:

- a PLC project that generates random data and links the data with data sets [▶ 27] on the OPC UA RT Device's publishers.
- a PLC project that stores the data received by subscribers on the OPC UA RT Device.

The I/O configuration includes an OPC UA RT Device with two publishers and two subscribers:

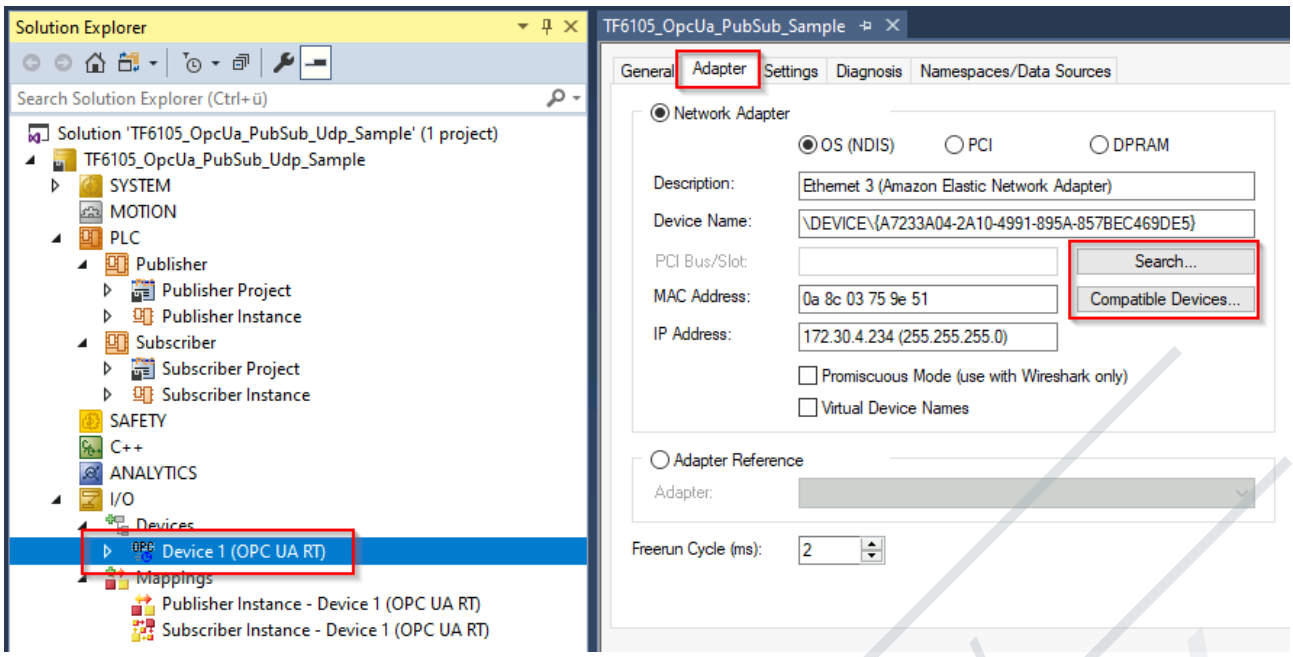
- a publisher that is configured to include a data set with variables. The variables are linked via the process image to the output variables of the first PLC project.
- a publisher that is configured to include a data set without variables. The variables are linked internally via their ADS symbol names to variables in the first PLC project.
- a subscriber that is configured to subscribe to the data published by the first publisher. The input variables are linked with variables from the second PLC project.
- a subscriber that is configured to subscribe to the data published by the second publisher. The input variables are linked with variables from the second PLC project.

Creation of the subscribers

The subscribers have been created automatically based on the Configuration import/export [▶ 35]. We highly recommend using this exchange format wherever possible.

How to start the sample

Before you activate the sample project, the OPC UA RT device needs to be set up for a network interface card that has the TwinCAT Realtime-Ethernet Adapter driver installed. This can be done via the OPC UA RT device settings dialog.



Please also make sure that your system meets the system requirements [▶ 9].

5.2 MQTT

This sample demonstrates how to set up a publisher and subscriber based on OPC UA Pub/Sub MQTT. The sample solution contains a TwinCAT project that includes the following PLC configuration:

- a PLC project that generates random data and links the data with data sets [▶ 27] on the OPC UA RT Device's publishers.
- a PLC project that stores the data received by subscribers on the OPC UA RT Device.

The I/O configuration includes an OPC UA RT Device with two publishers and two subscribers:

- a publisher that is configured to include a data set with variables. The variables are linked via the process image to the output variables of the first PLC project.
- a publisher that is configured to include a data set without variables. The variables are linked internally via their ADS symbol names to variables in the first PLC project.
- a subscriber that is configured to subscribe to the data published by the first publisher. The input variables are linked with variables from the second PLC project.
- a subscriber that is configured to subscribe to the data published by the second publisher. The input variables are linked with variables from the second PLC project.

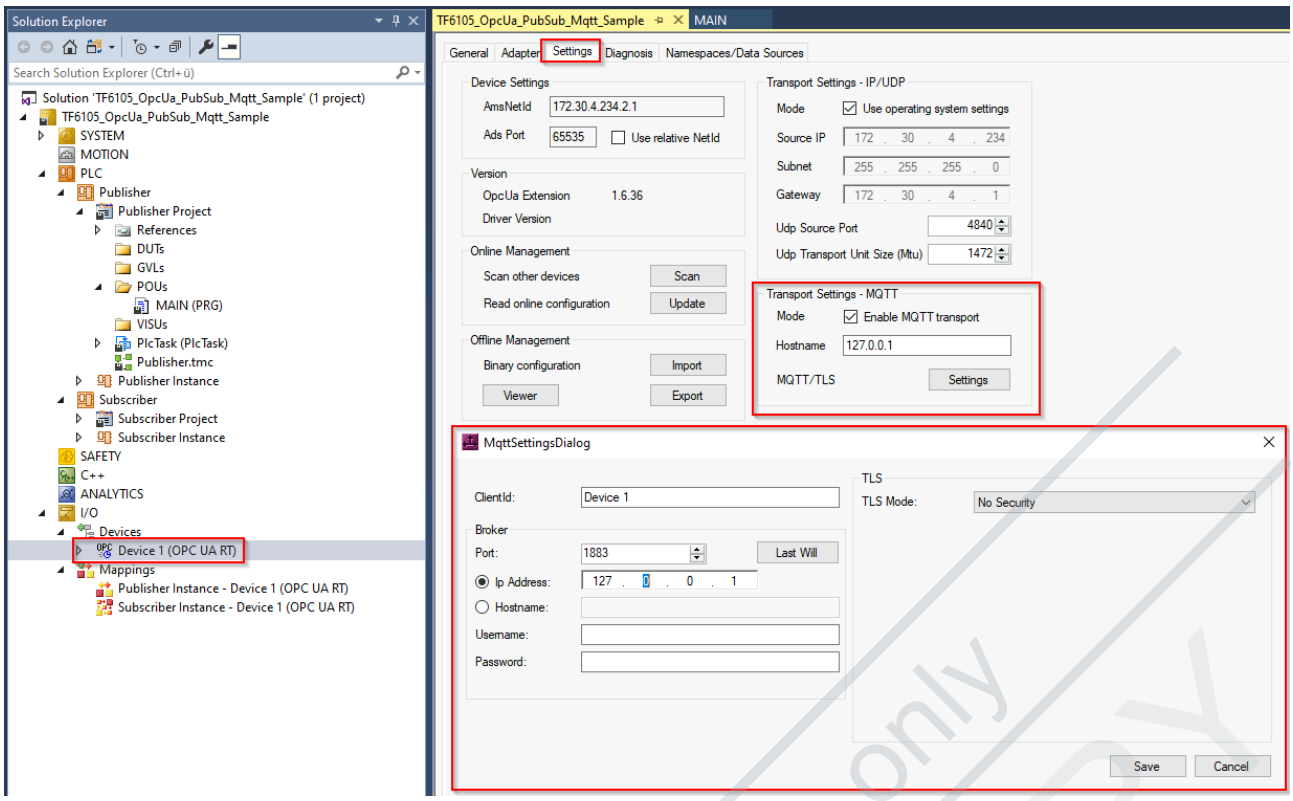
Creation of the subscribers

The subscribers have been created automatically based on the Configuration import/export [▶ 35]. We highly recommend using this exchange format wherever possible.

How to start the sample

Please make sure that your system meets the system requirements [▶ 9].

Before you activate the TwinCAT project, please make sure that you have an MQTT Message Broker running locally on the same system. The message broker should listen on port 1883/tcp without any encryption or user authentication. Please note that this is only for demonstration purposes! Please adapt this configuration to your environment. The MQTT settings can be found on the OPC UA RT device.



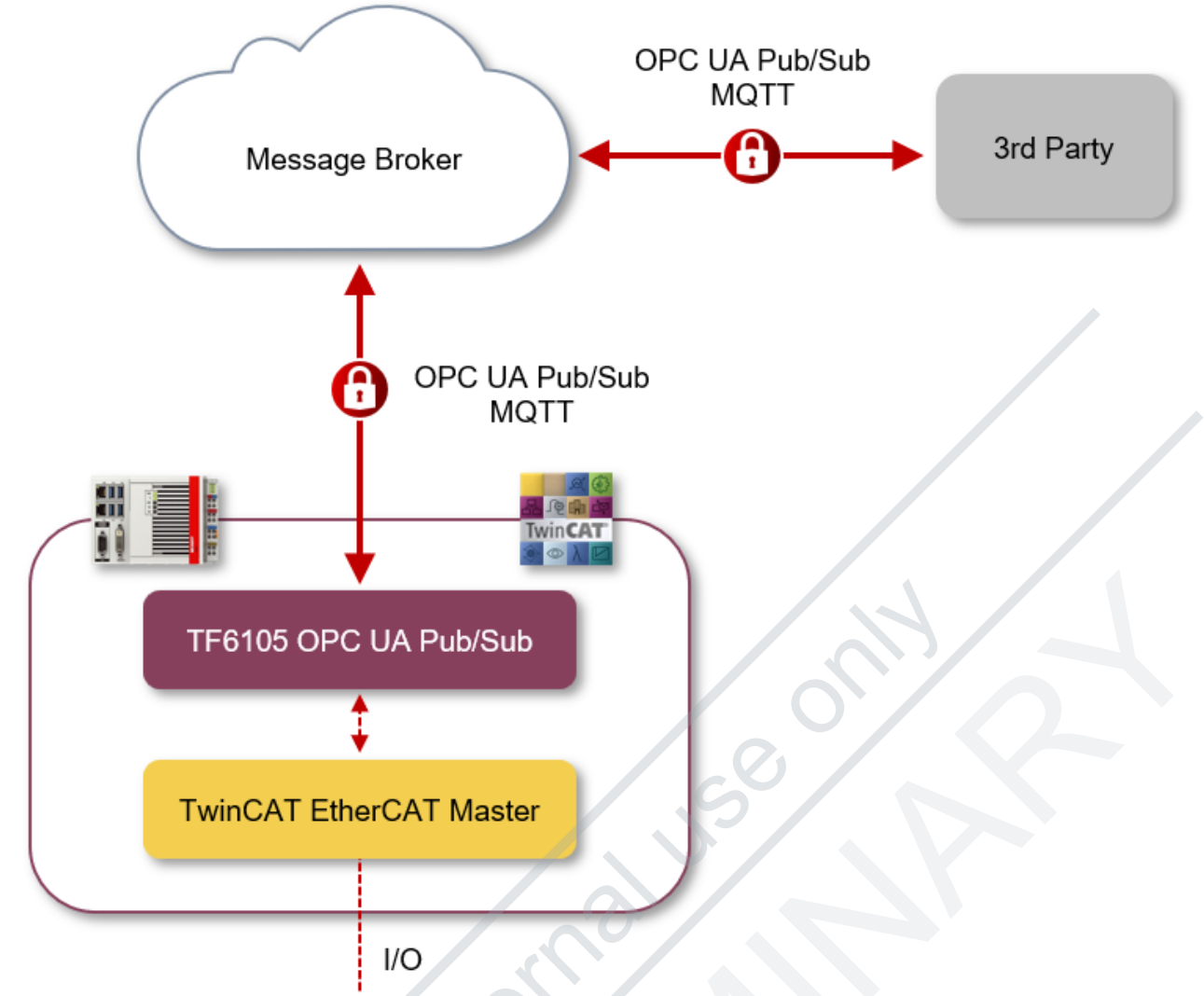
You can then activate the TwinCAT project and login to the PLC projects to debug the PLC runtime. You should see that the subscriber PLC project receives data (which is published by the other PLC project).

5.3 EtherCAT Master

You can easily publish data points from an EtherCAT Master image. An intermediate PLC project is not required but may be added in case you need data conversion or transformation. The following documentation describes all necessary steps to set this use case up. These steps consist of:

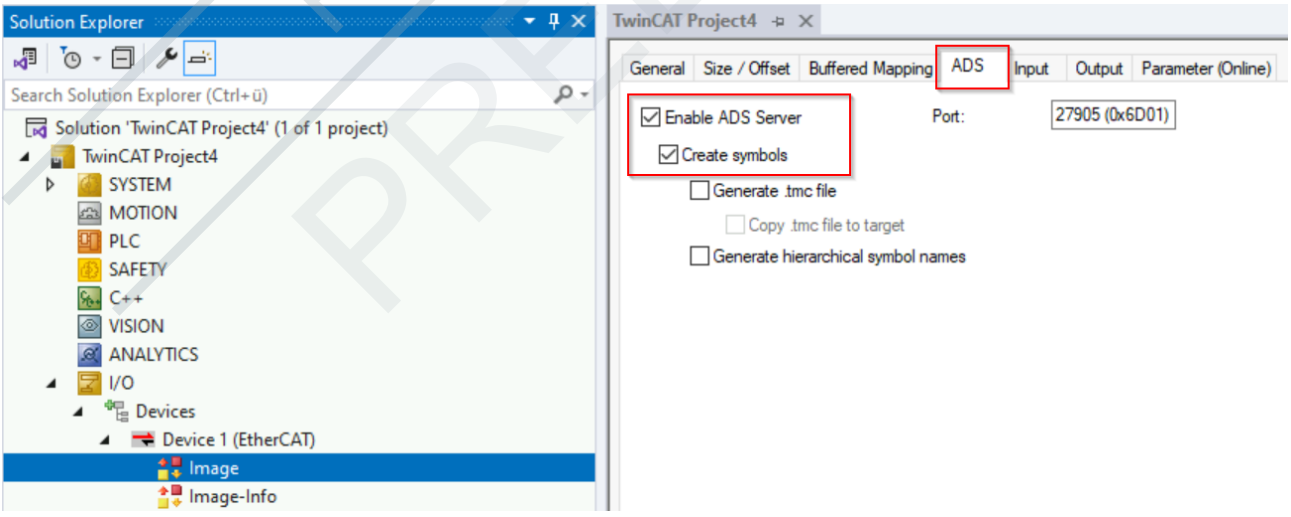
1. Activate the ADS Symbol Server on the EtherCAT Master
2. Set up an OPC UA RT Device
3. Configure a publisher with a data set
4. Add data points from the EtherCAT Master to the publisher data set
5. [optional] Set up a TwinCAT subscriber
6. [optional] Set up a generic MQTT client subscriber

This sample demonstrates this setup based on a use case in which I/O data points from the EtherCAT Master should be published to a message broker in the Cloud. A 3rd-party application should then connect to the same broker and subscribe to the data. This sample may be adapted to different variations of this use case as required by the application.



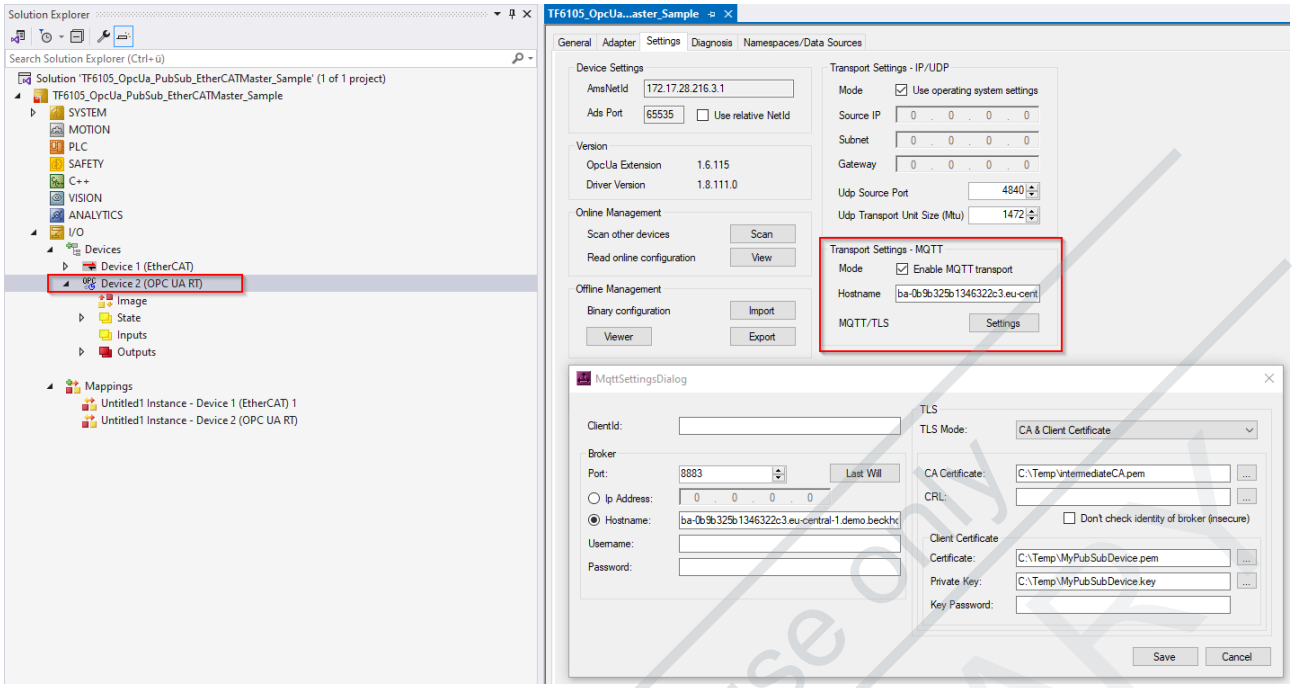
Activate the ADS Symbol Server on the EtherCAT Master

The TwinCAT EtherCAT Master offers the functionalities of an ADS Symbol Server. All terminals and variables on the EtherCAT Master are then made available via ADS. ADS Clients can connect to the ADS Symbol Server in order to browse its namespace and read/write data. You can activate this feature on the ADS settings page of the EtherCAT Master Image.



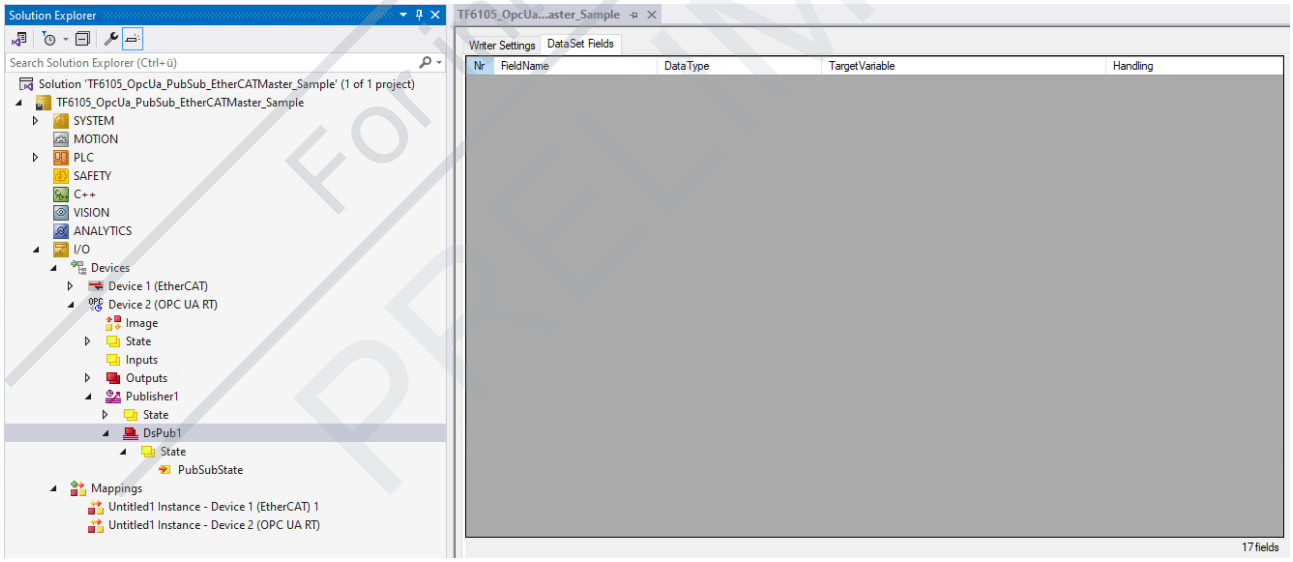
Set up an OPC UA RT Device

Set up an OPC UA RT Device as described in our Quick Start [▶ 13] tutorials. In this sample, the OPC UA RT Device will be configured for a secure MQTT connection to a remote message broker that runs in the Cloud but you can also run this sample locally. Please enter the MQTT connection details according to your environment.



Configure a publisher with a data set

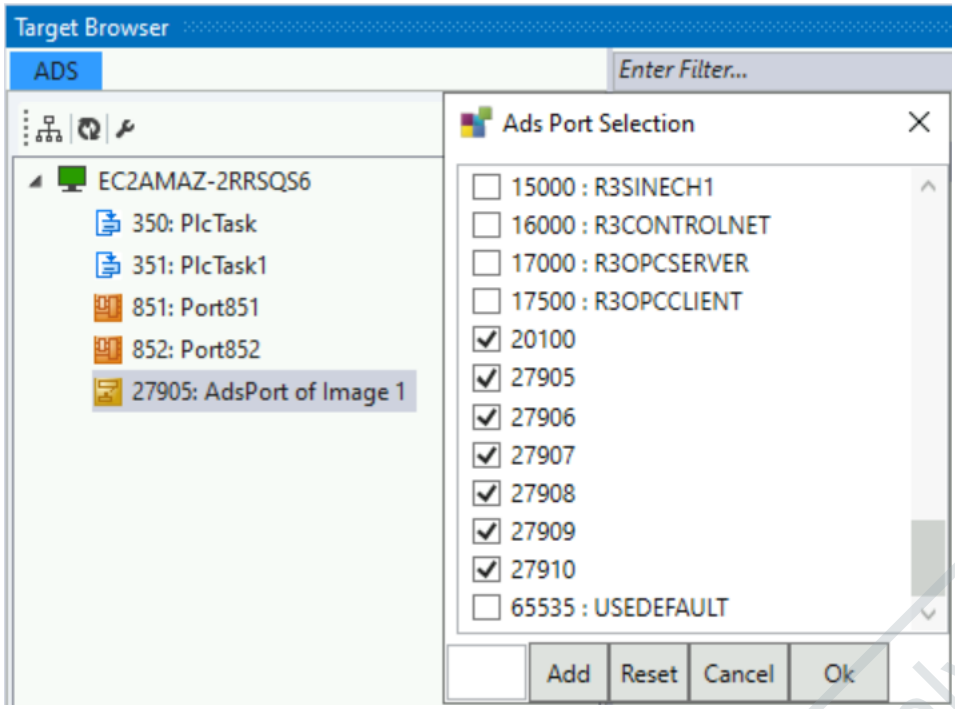
Next, you want to create a publisher that includes a data set containing variables that are linked to the data points of the EtherCAT Master Image. Right-click the OPC UA RT Device and select **Add New Item** and then **OPC UA Publisher**. Then right-click the added publisher and select **Add New Item** and then **Published DataSet**. Note that the data set does not contain any fields yet, as shown in the following screenshot.



You can now add data points from the EtherCAT Master by using the TwinCAT Target Browser.

Add data points from the EtherCAT Master to the publisher data set

Open the TwinCAT Target Browser and navigate to the ADS Symbol Server of the EtherCAT Master. If this ADS Server is not shown yet, you can add its port via the ADS Port Selection dialog, as shown in the following screenshot.

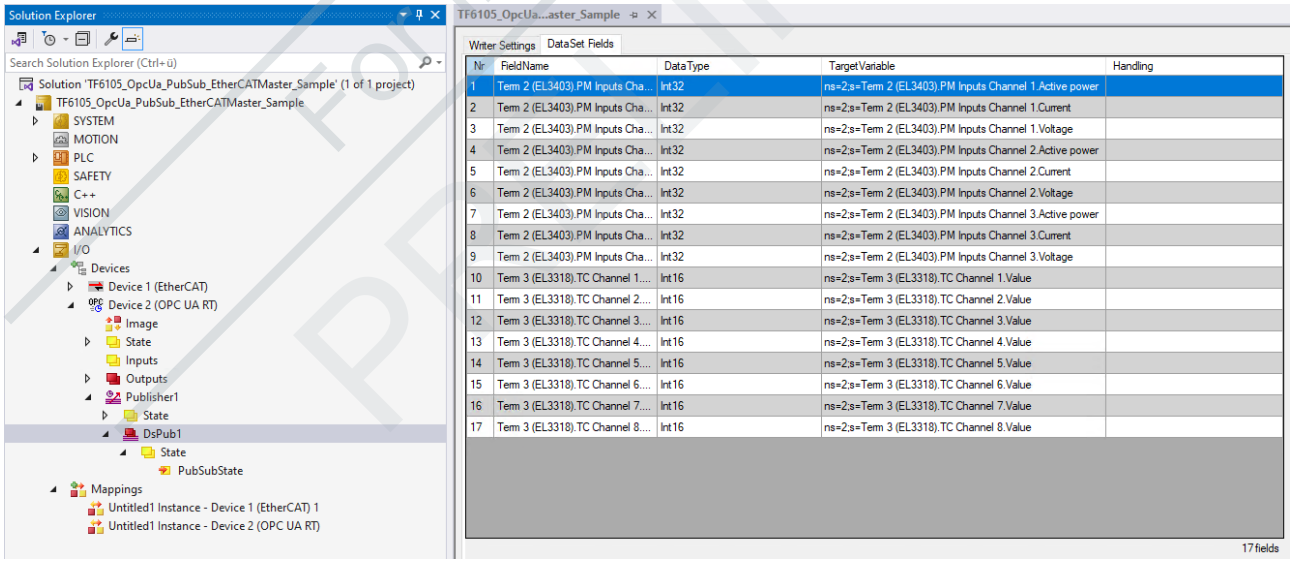


The ADS port of the EtherCAT Master Image can be found on the settings page in step 1.

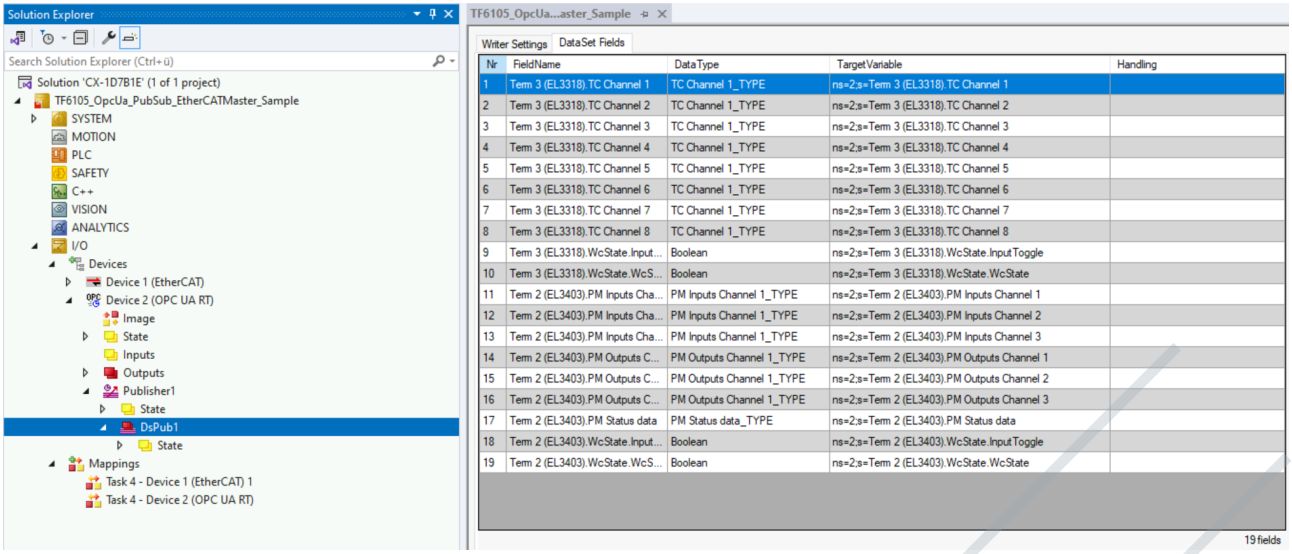
You can now navigate through the namespace of the EtherCAT Master Image and select the data points that you want to publish. When adding data points you have four options:

1. Select every data point individually and use Drag&Drop to add it to the data set.
2. Select the whole process image and use Drag&Drop to add it to the data set. In this case, every terminal on the image will be added as a structured data type to the data set.
3. Select the whole process image and use SHIFT + Drag&Drop to add it to the data set. In this case, only the variables on each terminal will be added to the data set (as simple data types). This option may be required if the OPC UA Pub/Sub subscriber does not support structured types.
4. Select the terminals and use Drag&Drop to add them to the data set.

The following screenshot shows the result of a Drag&Drop operation as described in option 1 for two terminals (EL3403 and EL3318).



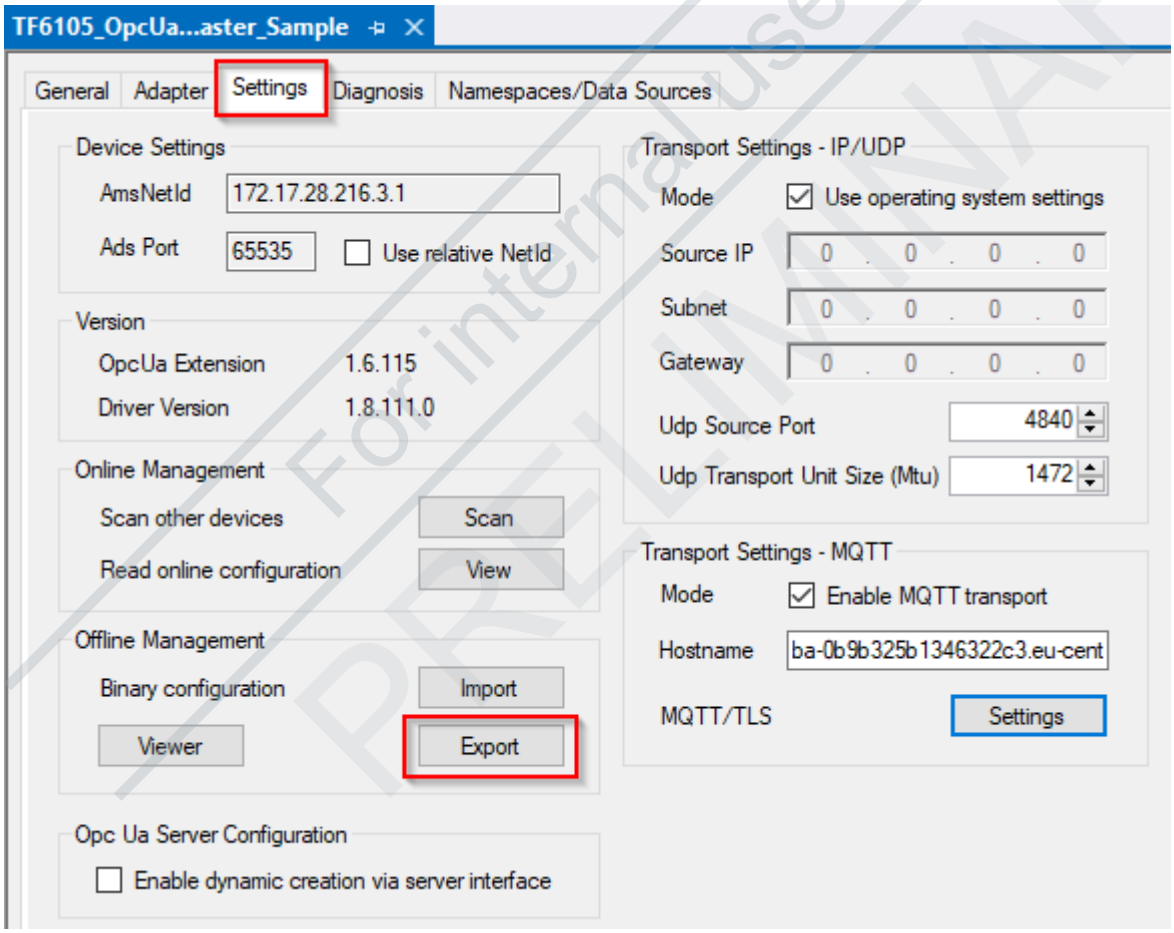
The following screenshot shows the result of a Drag&Drop operation as described in option 4 for two terminals (EL3403 and EL3318).



[optional] Set up a TwinCAT subscriber

You can easily set up a subscriber for the published data points by exporting the configuration [▶ 35] into a OPC UA Binary exchange file. This file can then be imported on the other side to automatically create a subscriber that fits the publisher configuration.

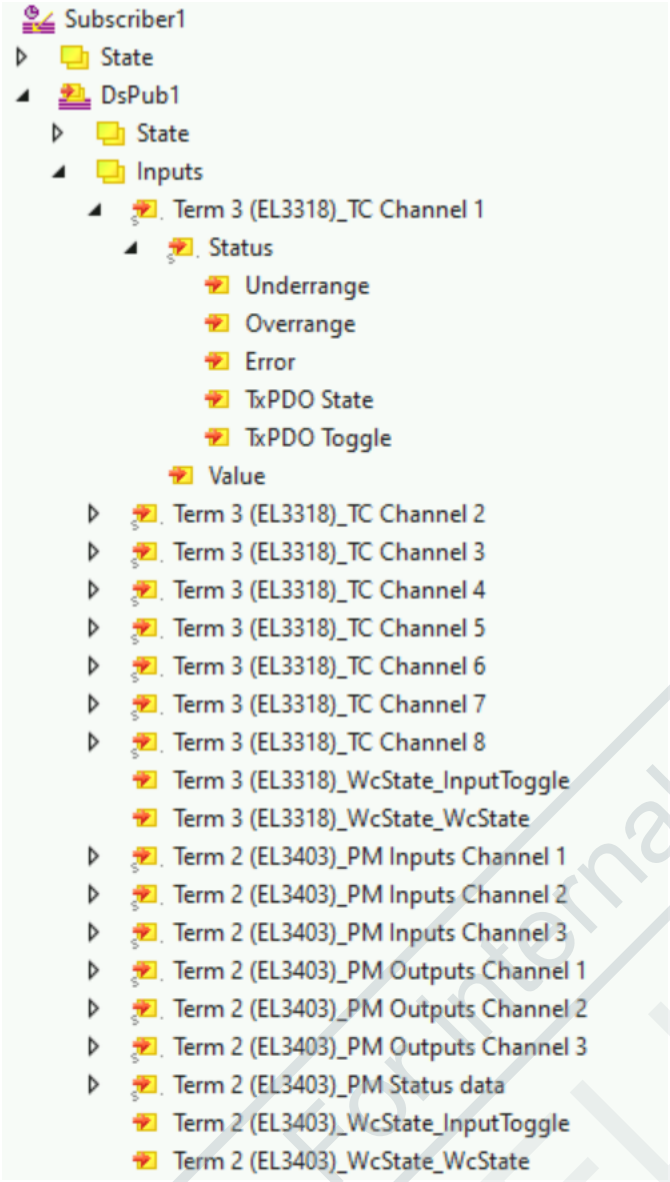
To export the configuration, double-click the OPC UA RT Device and select **Settings**. Click on the button **Export** and save the configuration in a file.



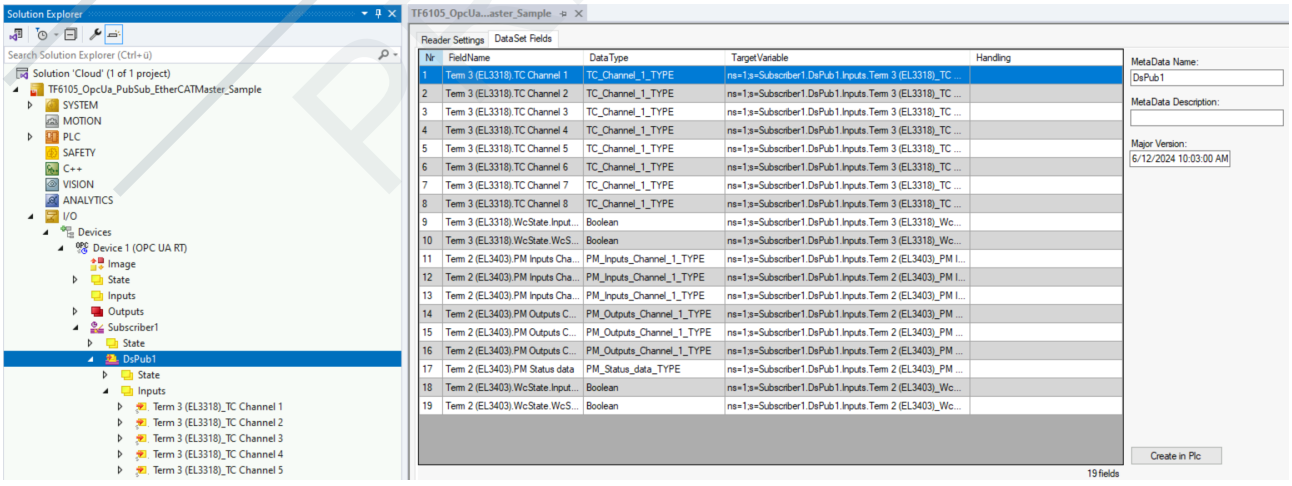
Import this configuration file on the device that should host the subscriber. If the device is also a system that runs TwinCAT, you can import the file to the corresponding TwinCAT project by using the **Import** button on the OPC UA RT Device.

You can also create a subscriber in the same project that also runs the publisher. The controller will therefore act as a publisher AND subscriber.

After importing the configuration file, the subscriber will be automatically configured with a data set that includes the fields from the publisher.



You can then create a mapping with variables e.g. from a PLC application or generate PLC code automatically via the **Create PLC code** button on the data set configuration.



This operation will create a GVL that contains variables that are linked to the process image variables via the attribute “TcLinkTo”, e.g.:

Subscriber1_DsPub1

```
VAR_GLOBAL
    {attribute 'TcLinkTo' := 'TIID^Device 1 (OPC UA RT)^Subscriber1^DsPub1^State^PubSubState'}
    PubSubState AT %I* : OpcUaPubSubState;
    {attribute 'TcLinkTo' := 'TIID^Device 1 (OPC UA RT)^Subscriber1^DsPub1^Inputs^Term 3 (EL3318)_TC Channel 1'}
    Term_3_TC_Channel_1 AT %I* : OpcUa_TC_Channel_1_TYPE;
    {attribute 'TcLinkTo' := 'TIID^Device 1 (OPC UA RT)^Subscriber1^DsPub1^Inputs^Term 3 (EL3318)_TC Channel 2'}
    Term_3_TC_Channel_2 AT %I* : OpcUa_TC_Channel_1_TYPE;
    {attribute 'TcLinkTo' := 'TIID^Device 1 (OPC UA RT)^Subscriber1^DsPub1^Inputs^Term 3 (EL3318)_TC Channel 3'}
    Term_3_TC_Channel_3 AT %I* : OpcUa_TC_Channel_1_TYPE;
    {attribute 'TcLinkTo' := 'TIID^Device 1 (OPC UA RT)^Subscriber1^DsPub1^Inputs^Term 3 (EL3318)_TC Channel 4'}
    Term_3_TC_Channel_4 AT %I* : OpcUa_TC_Channel_1_TYPE;
    {attribute 'TcLinkTo' := 'TIID^Device 1 (OPC UA RT)^Subscriber1^DsPub1^Inputs^Term 3 (EL3318)_TC Channel 5'}
    Term_3_TC_Channel_5 AT %I* : OpcUa_TC_Channel_1_TYPE;
    {attribute 'TcLinkTo' := 'TIID^Device 1 (OPC UA RT)^Subscriber1^DsPub1^Inputs^Term 3 (EL3318)_TC Channel 6'}
    Term_3_TC_Channel_6 AT %I* : OpcUa_TC_Channel_1_TYPE;
```

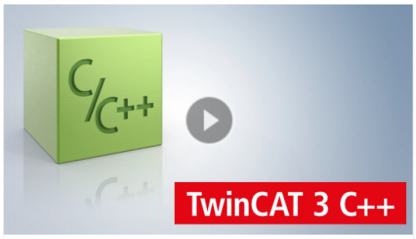
[optional] Set up a generic MQTT client subscriber

You can subscribe any MQTT client to the data that is published to the MQTT message broker. In this example we have used the MQTT/JSON data format which allows an ASCII-based, easy interpretation of the data. The following section shows how the resulting JSON format may look like.

```
{
  "DataSetWriterId": 1,
  "PublisherId": "1",
  "SequenceNumber": 5228,
  "MinorVersion": 771501780,
  "Timestamp": "2024-06-12T10:14:04.734Z",
  "Status": 0,
  "Payload": {
    "Term 3 (EL3318).TC Channel 1": {
      "Status": {
        "Underrange": false,
        "Overrange": false,
        "Error": false,
        "TxPDO State": false,
        "TxPDO Toggle": false
      },
      "Value": 233
    },
    "Term 3 (EL3318).TC Channel 2": {
      "Status": {
        "Underrange": true,
        "Overrange": false,
        "Error": false,
        "TxPDO State": false,
        "TxPDO Toggle": false
      },
      "Value": 13720
    },
...
...
```

6 Tutorials

You can find video tutorials for this product on our website at https://www.beckhoff.com/tutorials. The video tutorials provide a quick introduction to the product and the individual product facets.



14.03.2024 | Multimedia
Tutorial: TwinCAT 3 C++ overview
Get an overview of the C++ integration into TwinCAT.
Learn more →

12.02.2024 | Multimedia
Tutorial: Chart synchronization in TwinCAT Scope
Learn how to dock and synchronize charts in TwinCAT Scope.
Learn more →

12.02.2024 | Multimedia
Tutorial: Getting started with TF6100 TwinCAT 3 OPC UA Server
Learn how to configure the TwinCAT OPC UA Server.
Learn more →

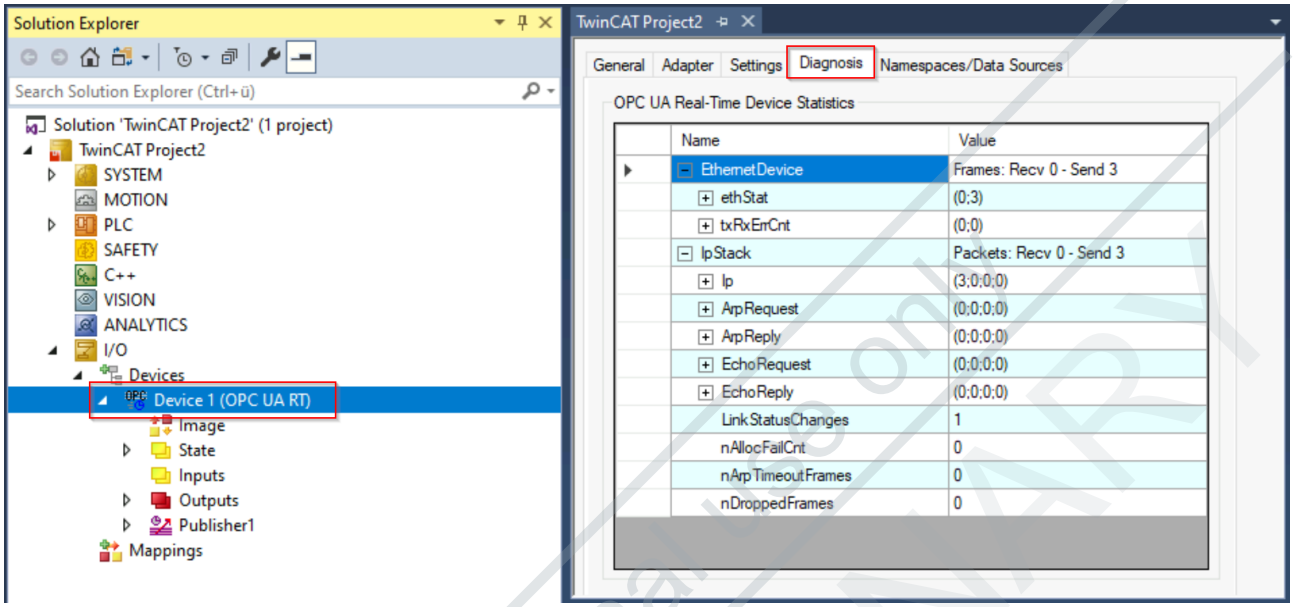
7 Appendix

7.1 Diagnostics

There are different configuration levels on which you can find diagnostics information. This information is available in a separate tab page on the corresponding configuration element.

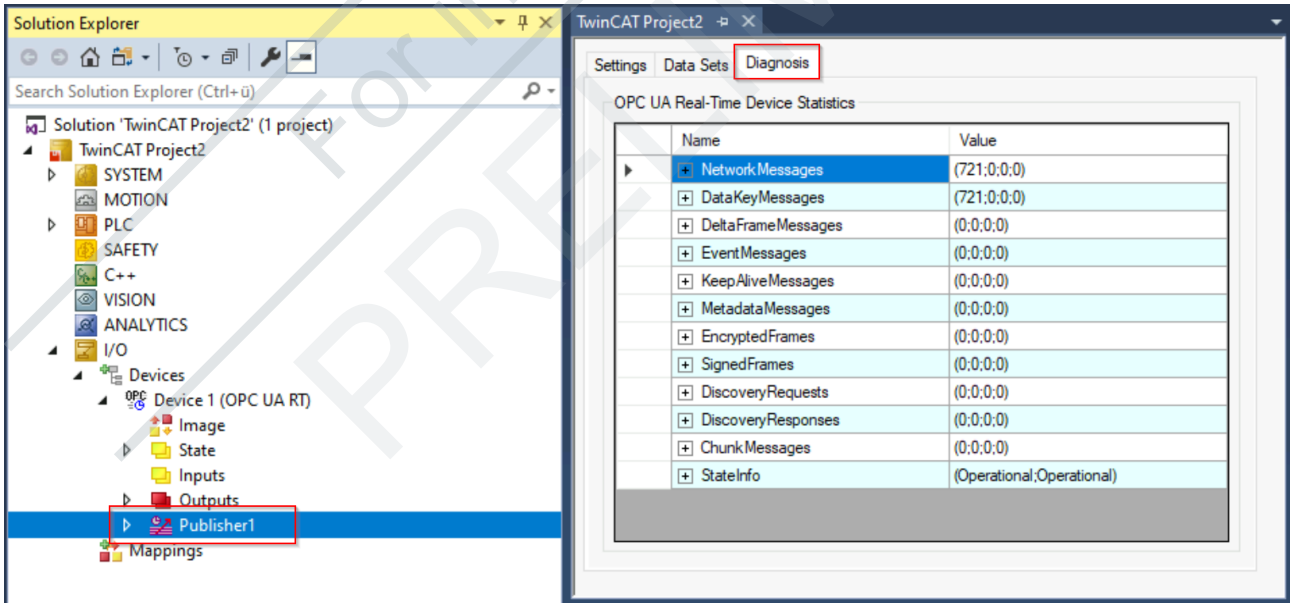
Device

The OPC UA RT device contains the tab **Diagnosis**, which includes different metrics to find out if and how many packets are send or received by the underlying IP stack.



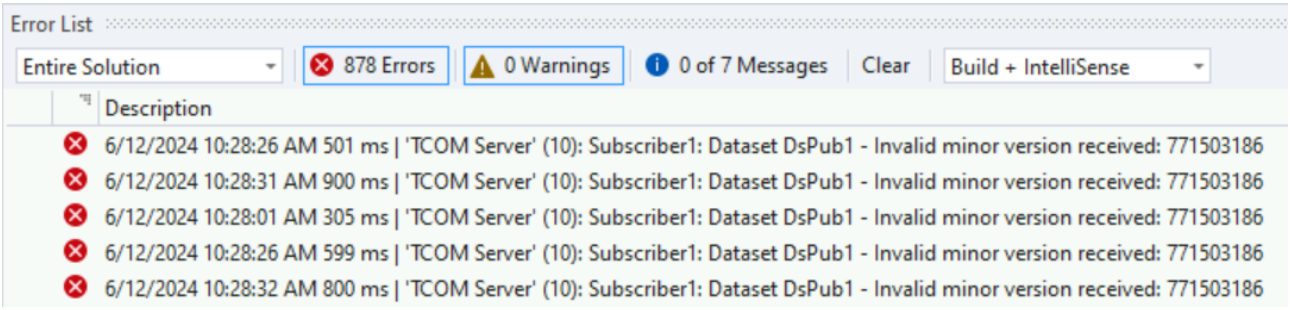
Publisher/Subscriber

Each publisher or subscriber contains a **Diagnosis** tab that includes different metrics for that particular publisher or subscriber.



TwinCAT Error List

The TwinCAT OPC UA Pub/Sub driver also sends logging information to the TwinCAT Error List window, which can be accessed from TwinCAT XAE.



Wireshark

Wireshark is a good software tool for advanced debugging and diagnostics. You can also use Wireshark to diagnose OPC UA Pub/Sub packets.

Example: the following Wireshark trace shows traffic caused by the configured Publisher that sends out packets every 1000ms (timestamp shown in first column) to the UDP Multicast group 224.0.0.1.

```
 16 2.740000000   172.17.98.153        224.0.0.1            OPCUA UADP     52 4840 → 4840 Len=10
 20 3.740000000   172.17.98.153        224.0.0.1            OPCUA UADP     52 4840 → 4840 Len=10
 24 4.740000000   172.17.98.153        224.0.0.1            OPCUA UADP     52 4840 → 4840 Len=10
 27 5.740000000   172.17.98.153        224.0.0.1            OPCUA UADP     52 4840 → 4840 Len=10
 30 6.740000000   172.17.98.153        224.0.0.1            OPCUA UADP     52 4840 → 4840 Len=10
 37 7.740000000   172.17.98.153        224.0.0.1            OPCUA UADP     52 4840 → 4840 Len=10
 46 8.740000000   172.17.98.153        224.0.0.1            OPCUA UADP     52 4840 → 4840 Len=10
 49 9.740000000   172.17.98.153        224.0.0.1            OPCUA UADP     52 4840 → 4840 Len=10
 54 10.740000000  172.17.98.153        224.0.0.1            OPCUA UADP     52 4840 → 4840 Len=10
```

Each packet contains the configured DataSet and its (in our case "counter") variable of data type INT (which equals Int16).

```
UADP Network Message
 > NetworkMessage Header
   Payload Header
 v DataSet Payload
   v DataSetMessage[0] (Data Key Frame Data)
     > DataSetFlags1
     v Data Key Frame
         Field Count: 1
       v DataSetField[0] (Variant Encoding)
         > Variant Encoding Mask: 4
         v Value (Int16)
             18889
```

Wireshark plugin/dissector

You might require the OPC UA Pub/Sub Wireshark dissector if you want to trace and interpret the Pub/Sub messages in a readable manner.

7.2 Troubleshooting

Behavior	Notes
I cannot receive any data on my subscriber and the TwinCAT Error List shows multiple errors that say “Invalid minor version received”.	This error is usually thrown whenever the source data has changed, e.g. when additional fields have been added to a publisher data set. In this case, the version information inside of the OPC UA Pub/Sub paket is incremented and it is therefore required to make the new configuration known to the subscriber. The easiest way to do that is to export the updated publisher configuration into a configuration file [▶ 35] and import it on the subscriber.
A connection to the message broker can not be established and the TwinCAT Error List shows multiple errors that say “Error connecting MQTT client MQTT_ERR_NOT_FOUND”.	Please make sure that you have used the correct hostname or IP address of your MQTT message broker.
A connection to the message broker can not be established and the TwinCAT Error List shows multiple errors that say “Error connecting MQTT client MQTT_ERR_CONN_TIMEDOUT”.	Please make sure that you have used the correct hostname or IP address of your MQTT message broker and that the broker can be reached at that address. Also make sure that you have specified the correct TCP/IP port of your message broker, e.g. 1883 or 8883.
A connection to the message broker can not be established and the TwinCAT Error List shows multiple errors that say “Error connecting MQTT client MQTT_ERR_TLS_VERIFY_FAIL”.	This error message is usually thrown when the Pub/Sub driver does not accept the message broker certificate, which is typically related to using a wrong CA certificate. Please make sure that you have specified the correct CA certificate.
A connection to the message broker can not be established and the TwinCAT Error List shows multiple errors that say “Error connecting MQTT client MQTT_ERR_TLS_HANDSHAKE_FAIL”.	This error message is usually thrown when the message broker does not accept your client certificate. It may also be thrown if you have specified no client certificate at all. Please make sure that you have specified the correct client certificate and that it is trusted by the message broker.
A connection to the message broker can not be established and the TwinCAT Error List shows multiple errors that say “Error connecting MQTT client MQTT_ERR_TLS_KEY_NOTFOUND”.	This error message is usually thrown when the client key cannot be found. Please make sure that the client key exists under the path that you specified in the MQTT connection settings.
A connection to the message broker can not be established and the TwinCAT Error List shows multiple errors that say “Error connecting MQTT client MQTT_ERR_TLS_KEY_INVALID”.	This error message is usually thrown when the client key cannot be opened. Please make sure that you have specified a valid client key file in the MQTT connection settings and that you haven’t mixed the client key with the client certificate.
A connection to the message broker can not be established and the TwinCAT Error List shows multiple errors that say “Error connecting MQTT client MQTT_ERR_TLS_CERT_NOTFOUND”.	This error message is usually thrown when the client certificate cannot be found. Please make sure that the client certificate exists under the path that you specified in the MQTT connection settings.
A connection to the message broker can not be established and the TwinCAT Error List shows multiple errors that say “Error connecting MQTT client MQTT_ERR_TLS_CERT_INVALID”.	This error message is usually thrown when the client certificate cannot be opened. Please make sure that you have specified a valid client certificate file in the MQTT connection settings and that you haven’t mixed the client certificate with the client key.

More Information:
www.beckhoff.com/tf6105

Beckhoff Automation GmbH & Co. KG
Hülshorstweg 20
33415 Verl
Germany
Phone: +49 5246 9630
info@beckhoff.com
www.beckhoff.com