IIoT Quick Start Guide

Pepperl+Fuchs IO-Link Master, ICE2 and ICE3 MQTT, and OPC UA

Manual

Your automation, our passion.

PEPPERL+FUCHS

With regard to the supply of products, the current issue of the following document is applicable: The General Terms of Delivery for Products and Services of the Electrical Industry, published by the Central Association of the Electrical Industry (Zentralverband Elektrotechnik und Elektroindustrie (ZVEI) e.V.) in its most recent version as well as the supplementary clause: "Expanded reservation of proprietorship"

Worldwide

Pepperl+Fuchs Group
Lilienthalstr. 200
68307 Mannheim
Germany
Phone: +49 621 776 - 0
E-mail: info@de.pepperl-fuchs.com

North American Headquarters

Pepperl+Fuchs Inc.
1600 Enterprise Parkway
Twinsburg, Ohio 44087
USA
Phone: +1 330 425-3555
E-mail: sales@us.pepperl-fuchs.com

Asia Headquarters

Pepperl+Fuchs Pte. Ltd.
P+F Building
18 Ayer Rajah Crescent
Singapore 139942
Phone: +65 6779-9091
E-mail: sales@sg.pepperl-fuchs.com
https://www.pepperl-fuchs.com

1 Introduction

Pepperl+Fuchs has developed an innovative way to share IIoT and Industry 4.0 data from its sensors and other sensor or actuator systems through communication solutions including its IO-Link master, the ICE2 and ICE3 devices, or the ICRL Ethernet switch series. The ICE2 and ICE3 solutions embed IIoT protocols such as OPC UA and MQTT along with various REST APIs to enable automatic data sharing with or without PLC communication. The IIoT protocols described below can communicate simultaneously along with the industrial Ethernet protocols such as PROFINET, Ethernet/IP, or ModbusTCP.

Pepperl+Fuchs' ICE2 and ICE3 have the ability to communicate process-critical data with the PLC for real-time control and send the non-time-critical data via MQTT or OPC UA for diagnostics, predictive maintenance, trending, and reporting. This ability is called MultiLink^{TM}.

The graphic below illustrates how the protocols run concurrently and send process and sensor health data to multiple software systems. The software systems can be a cloud-enabled dashboard, on-premise SCADA or MES system, or a centralized Unified Name Space (UNS) software system.

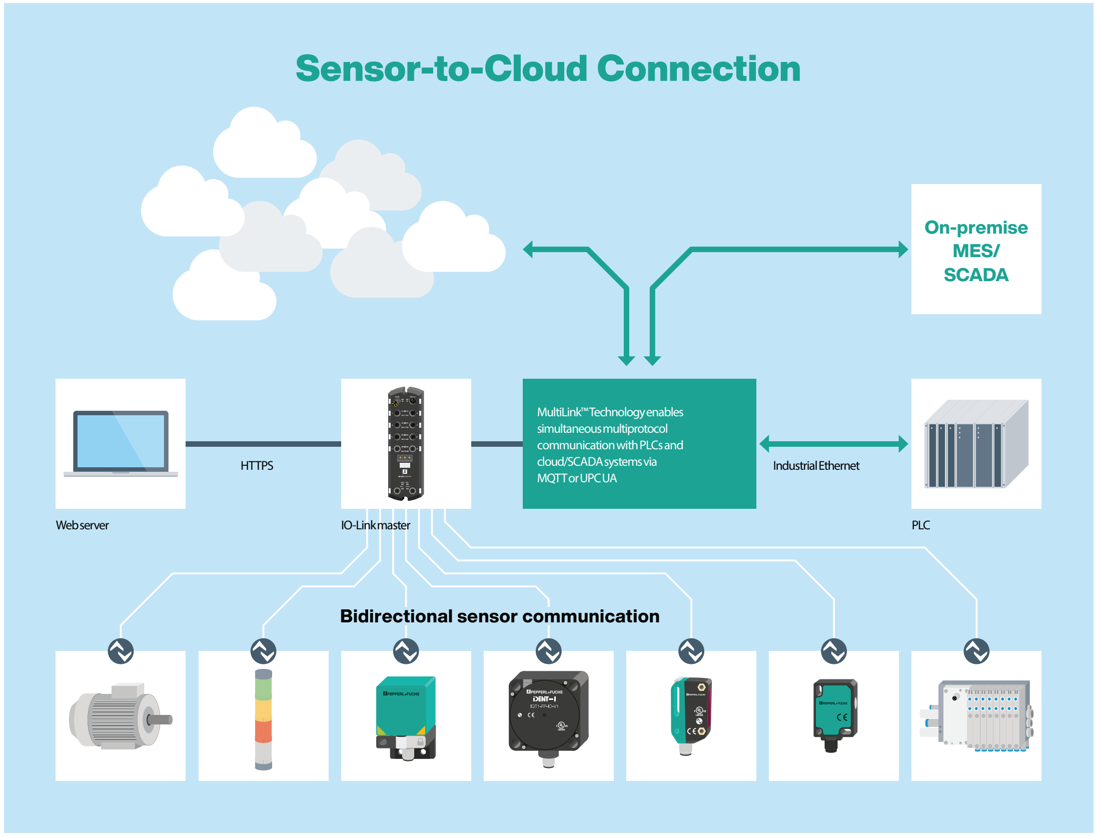

Sensor-to-Cloud Connection
On-premise MES/ SCADA
HTTPS
MultiLink™ Technology enables simultaneous multiprotocol communication with PLCs and cloud/SCADA systems via MQTT or UPC UA
Industrial Ethernet
Web server
IO-Link master
PLC
Bidirectional sensor communication

Figure 1.1

The differences and capabilities of OPC UA and MQTT are described in the following table:

	OPC UA	MQTT
Communication Model	Client/Server Publish/Subscribe	Publish/Subscribe
Communication Method	Request/Response Publish/Subscribe (based on MQTT/AMQP)	Publish/Subscribe
Participants	Identified (Handshake during connection)	Unknown (Broker has no info about participants)
Relationship	n:1 (Client/Server) n:m (Pub/Sub)	n:m
Data Update	Cyclic and change-driven (for events and alarms)	Change-driven
Data Overhead	High	Low
Standardization	High	Low
Metadata	Yes	No
Security	Integrated (Optional)	Separate (On higher network level)
Driving Industry	Automation	IT

Table 1.1

Depending on the project application needs and its SCADA or MES control software capabilities, the customer can decide which protocol will work best. Flexibility and innovation are the driving principles behind Pepperl+Fuchs' Sensorik 4.0 strategy, which gives the customer control of their sensing and data-sharing requirements for automation applications.

This guide will provide the necessary understanding to enable the MQTT and OPC UA protocols within the embedded webserver on the ICE2 and ICE3 sensor edge gateway and its associated connection software called PortVision DX.

Additionally, Pepperl+Fuchs' IIoT Starter Kits contain everything that you need to integrate IO-Link technology into your system. The following diagrams illustrate the suggested connections and layout of the products included in the kits, making setup quick and easy.

2 MQTT Protocol Primer

The ICE2 and ICE3 MQTT package provides a way to publish various data to an MQTT broker. MQTT is a simple publish-subscribe messaging protocol that is becoming popular for use within Industrial Internet of Things (IIoT) applications:

- http://mqtt.org
- wikipedia.org/wiki/MQTT

What makes MQTT different from industrial communication protocols?

It is a **lightweight** protocol that enables communication between devices and on-premise software and cloud-based solutions. The devices do not need a lot of processing power, making them small and inexpensive. These devices generate and send a limited amount of data and can be operated on low-quality networks. While industrial communication protocols tend to offer fast, real-time data transfer, MQTT does not.

The MQTT standard does not define any format for the published messages, but JSON has been almost universally adopted by the MQTT implementation in the IIoT area, so JSON has been implemented by Pepperl+Fuchs for the ICE2 and ICE3 IO-Link master MQTT implementation.

Currently, most data is published "on-change" with a minimum interval of two seconds between updates. One exception to this publishing rule is that the ICE2 and ICE3 "uptime" message is sent at a fixed ten-second interval.

How does the MQTT network architecture operate with the Pepperl+Fuchs ICE2 and ICE3?

MQTT's network architecture involves clients and a broker instead. Here, the client establishes a connection to the broker and transmits its internal data. The broker acts as a central clearing house for data and can be in the cloud or on a private server. It uses topics to organize the data sent by the client. For instance, ICE2 and ICE3 IO-Link masters have between eight to sixteen IO ports. Any data for a sensor connected to port 1 has "port 1" in its topic name. One of the topics Pepperl+Fuchs IO-Link masters publish, or send, is the IO-Link sensor's process data. When the IO-Link master publishes updates on a topic, the broker updates that information.

One of the most important operations the broker performs is it allows clients to subscribe to this data. An application that needs to use data from the distance sensor (think Pepperl+Fuchs' R200 optical sensor) connected to port 1 on the IO-Link master will simply subscribe to the "process data from port 1" topic. Data will then flow from the sensor to the customer's SCADA, MES, or Cloud dashboard application.

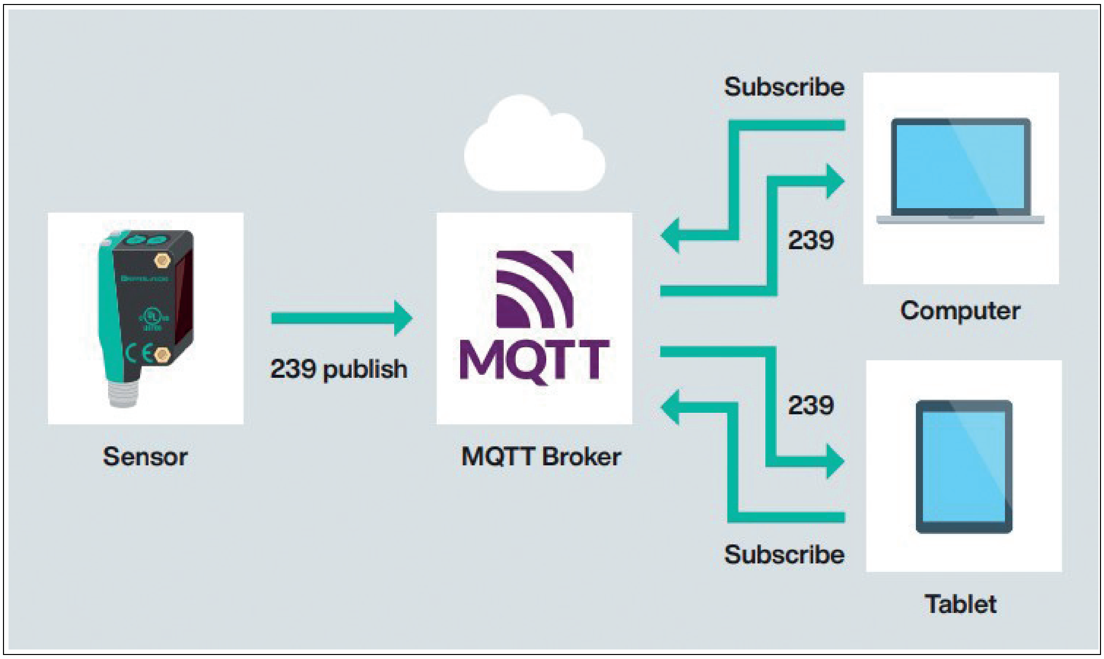
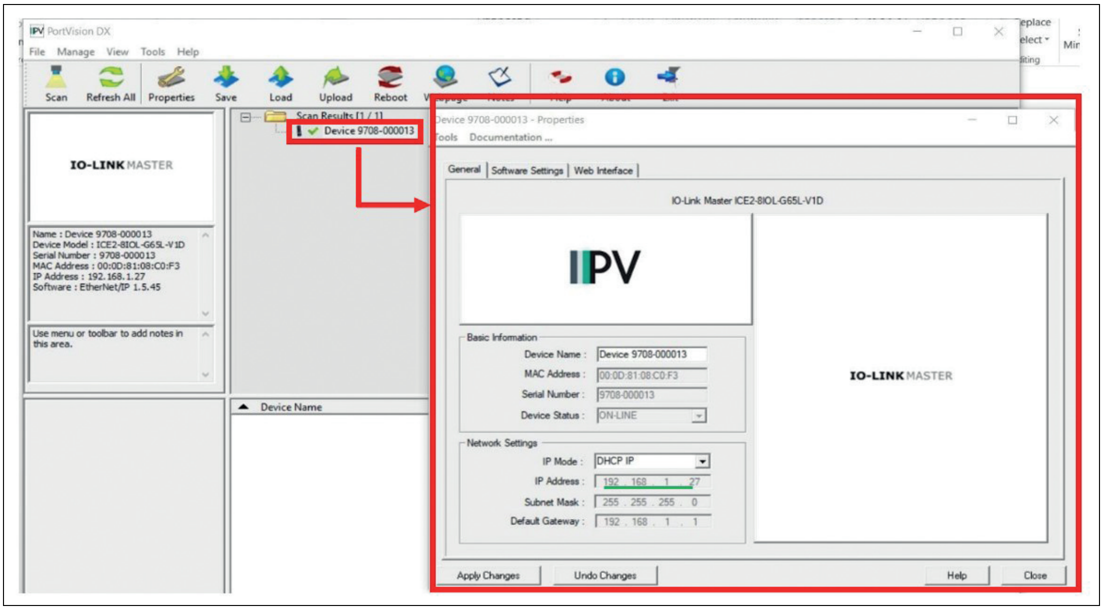


Figure 2.1 An IO-Link distance measurement sensor published the "target distance" to the broker. User application programs subscribe to this data (topic) and receive updates when they are available.

3 Configuring ICE2 / ICE3 Network Settings

Finding ICE2 / ICE3 in PortVision DX

1. Start PortVision DX and click **Scan** to scan the network for IO-Link masters.
2. Your master will populate under **Scan Results**. Once it has populated, double-click the device name to open the device properties window.



Name : Device 9708-000013
Device Model : ICE2-8IOL-G65L-V1D
Serial Number : 9708-000013
MAC Address : 00:0D:81:08:C0:F3
IP Address : 192.168.1.27
Software : EtherNet/IP 1.5.45

Use menu or toolbar to add notes in this area.

IO-Link Master ICE2-8IOL-G65L-V1D

Device Name : Device 9708-000013
MAC Address : 00:0D:81:08:C0:F3
Serial Number : 9708-000013
Device Status : ON-LINE

IP Mode : DHCP IP
IP Address : 192 . 168 . 1 . 27
Subnet Mask : 255 . 255 . 255 . 0
Default Gateway : 192 . 168 . 1 . 1

Figure 3.1 Device Properties window

Network setup: DHCP

1. In the Network Settings section of the Device Properties window, the **IP Mode** must be set to "**dhcp**." If yours is not, click the drop-down menu to change it.Then click the **Apply Changes** button.
2. Now that the IO-Link master is running DHCP IP mode, your router will assign it an IP address. Again, click **Scan** in PortVision DX to scan for the IO-Link master. Note its IP address, start a web browser, and open a page at that IP address. In the example, the IP address is 192.168.1.27. Alternatively, you can use the Webpage button in PortVision DX.

Network setup: Static IP

1. In the Network Settings section of the Device Properties window, the **IP Mode** must be set to "**static.**" If yours is not, click the drop-down menu to change it. Then click the **Apply Changes** button.

2. Now enter the static IP address, Subnet mask, and Gateway Address, and then click the **Apply Changes** button.
3. You will also need to enter the DNS1 and DNS2 on the web interface if your MQTT broker is using a server name (URL).
4. Click **Scan** in PortVision DX to scan for the IO-Link master. Note its IP address, start a web browser, and open a page at that IP address. In the example, the IP address is 192.168.1.27.

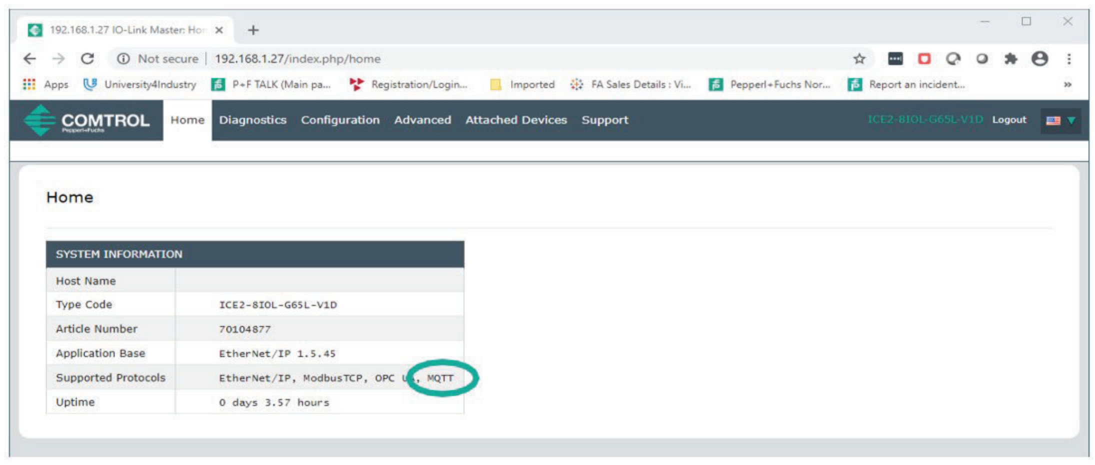
   Alternatively, you can use the Webpage button in PortVision DX.

3.1 Configuring MQTT Protocol

To enable the MQTT publish and subscribe capability within Pepperl+Fuchs' ICE2 and ICE3 IO-Link masters, see the following configuration instructions.

3.2 MQTT Settings

You must also make sure that your IO-Link master has the right firmware to support MQTT. If your home screen lists MQTT as one of the supported protocols, you are ready to configure MQTT. If not, you will need to update a few internal files. Contact your Pepperl+Fuchs Sales support to obtain the latest firmware version if the firmware does not show MQTT capability.



Home

SYSTEM INFORMATION	
Host Name	
Type Code	ICE2-8IOL-G65L-V1D
Article Number	70104877
Application Base	EtherNet/IP 1.5.45
Supported Protocols	EtherNet/IP, ModbusTCP, OPC UA, MQTT
Uptime	0 days 3.57 hours

Figure 3.2 MQTT as supported protocol

With MQTT supported, we will now configure MQTT and "tell" the IO-Link master everything it needs to know about the broker. In the web browser, click the **Configuration** menu, then select **MQTT**.

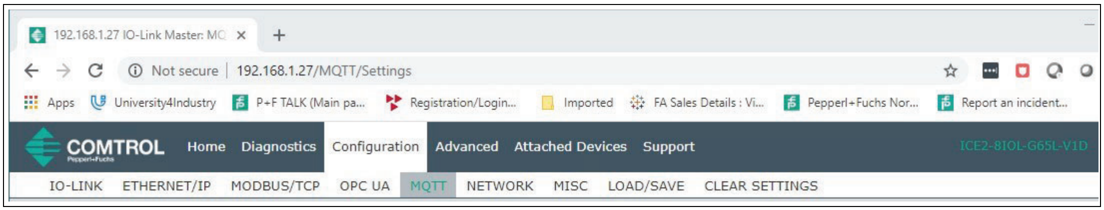
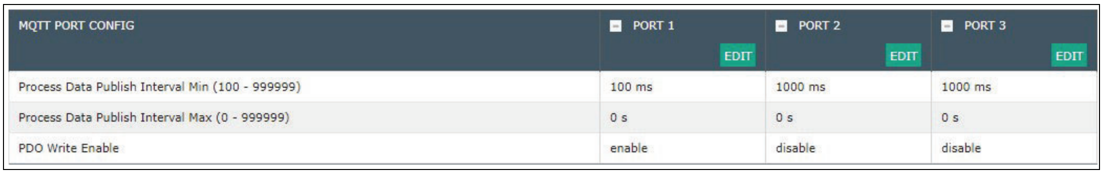


Figure 3.3 Configuring MQTT

3.3 Port-Specific MQTT Settings



MQTT PORT CONFIG	▬ PORT 1 EDIT	▬ PORT 2 EDIT	▬ PORT 3 EDIT
Process Data Publish Interval Min (100 - 999999)	100 ms	1000 ms	1000 ms
Process Data Publish Interval Max (0 - 999999)	0 s	0 s	0 s
PDO Write Enable	enable	disable	disable

Figure 3.4 MQTT settings

- **Process Data Publish Interval Min (100**–999999)specifies the rate the IO-Link master will publish port data when the data is changing faster than this rate. If you set this parameter to 100 ms, the IO-Link master will publish the process data every 100 ms, even though the data is changing at a faster rate.
- **Process Data Publish Interval Max (0**–999999) specifies the maximum time between two updates (i.e., a heartbeat). A value of 0 s indicates that updates are only published based on the process data changes. Any other number forces an update after this **Max Interval**, even if the process data has not changed.
- **PDO Write Enable** specifies whether writing output data is allowed at this point. In applications where MQTT is only used to read port data, PDO Write Enable "disable" is the right setting. If a PLC is used to write RFID tag data and a client application is only interested in logging the result of the read activity, this is the right setting. If the sensor configurations are managed via MQTT while the PLC is only using the sensor data to run logic and control, PDO Write Enable "enable" is likely the correct setting.

3.4 MQTT Client Configuration Data

Global Configuration Settings

Name	Type	Default	Description
MqttClientEnable	Enum	Disable	enable/disable the MQTT client (not implemented)
MqttServerName	String		MQTT server host-name or IPv4 address
MqttServerPort	Int	1883	MQTT server port (065535)
MqttUseTls	Boolean	False	Use SSL/TLS encryption
MqttUsername	String		Username sent to server for authentication
MqttPassword	String		Password sent to server for authentication
MqttClientId	String		Client ID sent to server when connecting
MqttTopicBase	String		Path prefix used for all publish messages
MqttPubRetain	Boolean	False	Request that broker retain published messages

Name	Type	Default	Description
MqttIsduWriteEnable	String	Disable	Enable ISDU write

Table 3.1

Per-Port Configuration Settings

Name	Type	Default	Description
MqttPdoWriteEnable	String	Disable	Enable PDO write

ISDU Read/Write

Since MQTT lacks intrinsic support for request/response semantics, ISDU read/write requests and responses are handled via a pair of topics:

- MqttTopicBase/port/n/isdu/request/client_transaction_id and
- MqttTopicBase/port/n/isdu/response/client_transaction_id

Requests for ISDU read/write are published by other clients to the "request" topic shown above. The client_transaction_id is an arbitrary string chosen by the requesting client and should be chosen to be unique. After the ISDU operation is completed, the IOLM will publish the response to the corresponding "response" topic (with the same client_transaction_id as the request).

ISDU Request Payload

Name	Type	Description
Op	String	Required—must be "read" or "write"
Index	Integer	Required
Subindex	Integer	Optional—defaults to 0 if not provided

Fields specific to read requests:

Name	Type	Description
Format	String	Optional—if present, it determines the format of the returned read data in the response. Should be one of the following: "str", "raw", or "unit." If not provided, read data will be in all formats.

Fields specific to write requests (exactly one of uint, raw, or str must be present):

Name	Type	Description
Raw	String	Whitespace-delimited hex-byte data string
Uint	Integer	Integer data value (requires len field)

Name	Type	Description
Str	String	UTF-8 data string (len field is optional)
Len	Integer	Required for uint data; optional for str data. Controls number of data bytes written.

In a write request with str data and a len field, the string will be NULL-padded to the requested length before being written to the device.

ISDU Response Payload

The ISDU response payload is a JSON object with the fields described below:

Name	Type	Description
Op	String	"Op" value from request
Index	Integer	"Index" value from request
Subindex	Integer	"Subindex" value from request
Status	String	String "OK" if the request was successful, otherwise an error message

Fields specific to read response (one or more of raw, str, uint may be present)

Name	Type	Description
Raw	String	Whitespace-delimited hex-byte data string
Uint	Integer	Unsigned integer value
Str	String	UFT-8 string data
Len	Integer	Number of bytes read

PDO Write

PDO values may be written by publishing to MQTTTopicBase/port/n/pdowr. The payload may contain PDO data fields in one of two formats: raw or uint. Data in raw format must match the PDO length exactly. Data in uint format is supported only for PDO lengths of no more than four bytes. It may also contain a Boolean **valid** flag.

Implementation Notes

The Pepperl+Fuchs IO-Link master MQTT package is implemented using the Eclipse Paho C client (www.eclipse.org/paho/clients/c) libraryversion 1.3.0 (github.com/eclipse/paho.mqtt.c/tree/v1.3.0).

This concludes the MQTT portion on the Pepperl+Fuchs IO-Link master. The next section will provide connectivity guidance on OPC UA server and client information.

4 OPC UA Protocol Primer for Pepperl+Fuchs' ICE2/3 IO-Link Master

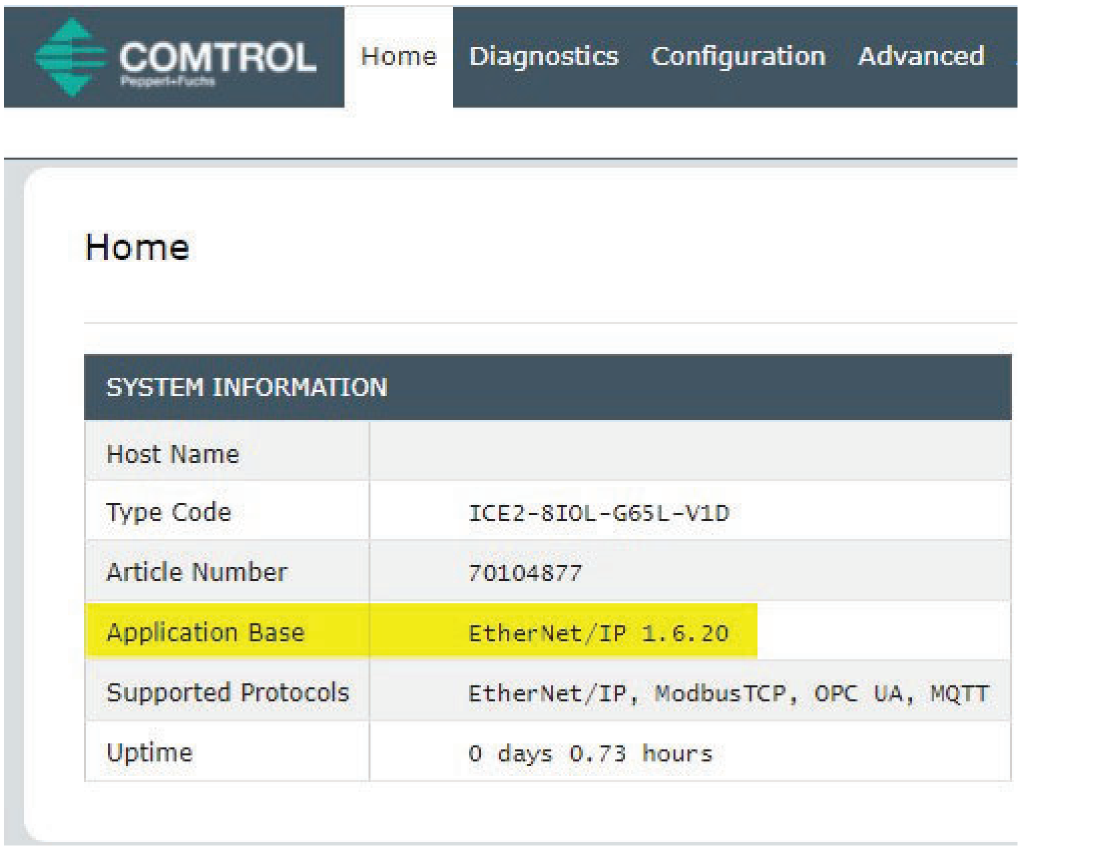
How to Enable OPC UA on IO-Link Master

Ensure Compatibility

1. Check the IO-Link master to confirm it has the following application base images installed:
   - **EIP v1.5.0.1 or higher**
   - **PNIO v1.5.0.1 or higher**
2. Check your software version using one of the following methods:

Web Page Method

1. Ensure you are connected to the same network as your IOLM.
2. Type in the IP address of your IOLM.
3. The software version will display on the Home page of the IP address.



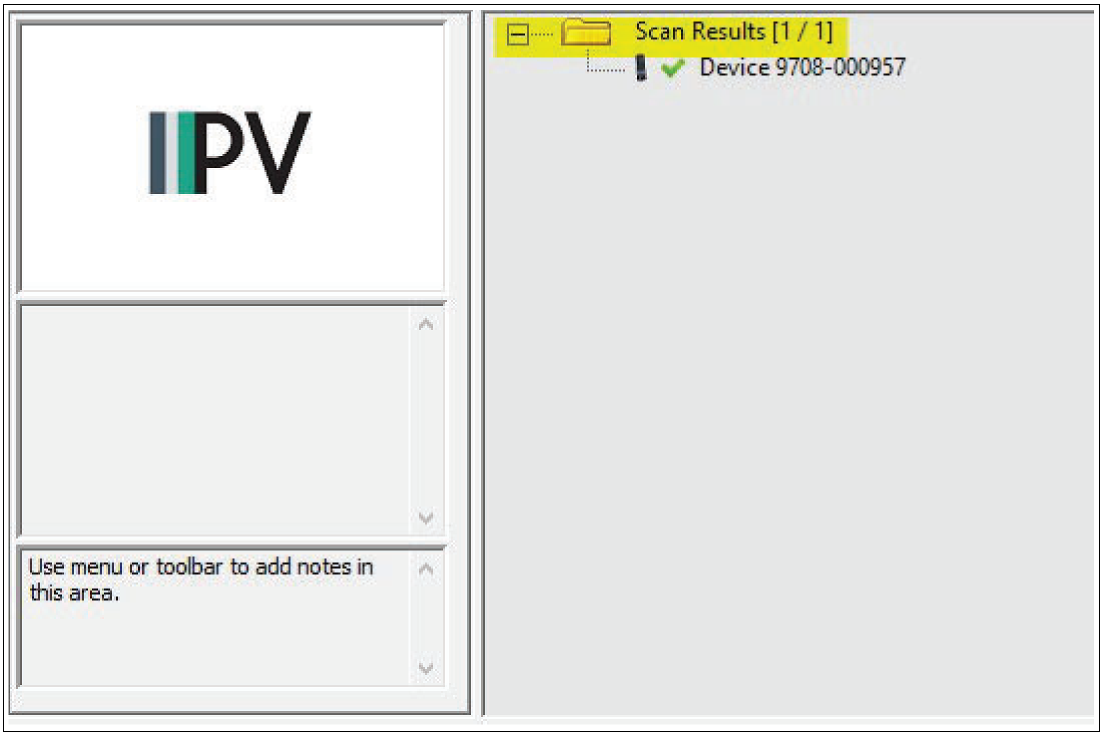
Home

SYSTEM INFORMATION	
Host Name	
Type Code	ICE2-8IOL-G65L-V1D
Article Number	70104877
Application Base	EtherNet/IP 1.6.20
Supported Protocols	EtherNet/IP, ModbusTCP, OPC UA, MQTT
Uptime	0 days 0.73 hours

Figure 4.1 Software version information

PortVision DX Method

1. To check your software on PortVision DX, launch the application, left-click **Scan** once the IOLM is connected to the network, check the box labeled **IO-Link Master / ICE2 / ICE3**, and left-click **Scan**. Highlight the folder of the connected device. In the example below, the folder is named **Scan Results [1/1]**.

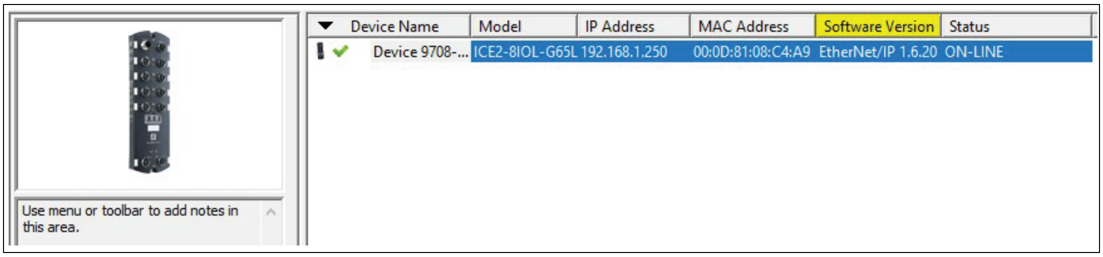


Scan Results [1 / 1]
Device 9708-000957

Use menu or toolbar to add notes in this area.

Figure 4.2 Scan results

2. The IO-Link master will appear on the bottom section of PortVision DX. Navigate to **Software Version** and confirm its EIP / PNIO is v1.5.0.1 or higher. Keep note of the IP address, as it will be used in the next step.



Device Name	Model	IP Address	MAC Address	Software Version	Status
Device 9708-...	ICE2-8IOL-G65L	192.168.1.250	00:0D:81:08:C4:A9	EtherNet/IP 1.6.20	ON-LINE

Use menu or toolbar to add notes in this area.

Figure 4.3 Software version information

Tip

Not the correct version?

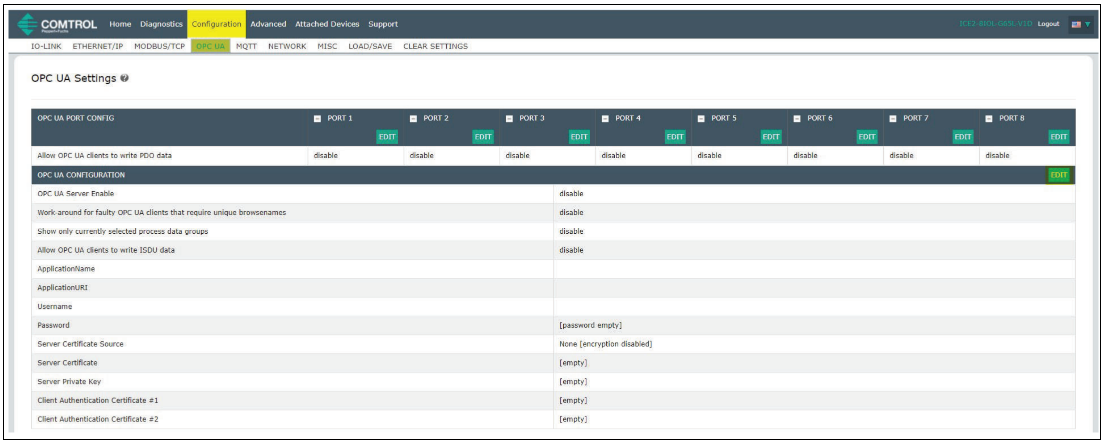
Depending on when the IO-Link master was manufactured, you may need to send the IO-Link master to Pepperl+Fuchs for image updates, if below v1.5.0.1

If you do NOT have PortVision DX, install it using this link:

Install PortVision DX

Configuring

1. After confirming your software version is correct, open a web browser and enter the IO-Link master IP address.
2. Once loaded, click on **Configuration**, the **OPC UA**.
3. Click **Edit** for the **OPC UA Configuration** tab on the right side.



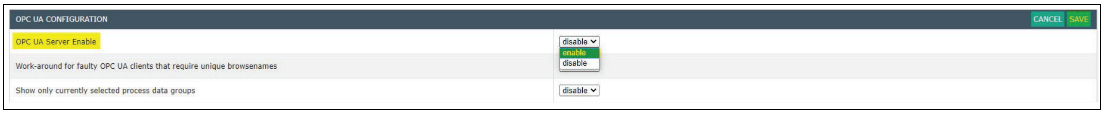
OPC UA Settings

OPC UA PORT CONFIG	PORT 1 EDIT	PORT 2 EDIT	PORT 3 EDIT	PORT 4 EDIT	PORT 5 EDIT	PORT 6 EDIT	PORT 7 EDIT	PORT 8 EDIT
Allow OPC UA clients to write PDO data	disable	disable	disable	disable	disable	disable	disable	disable

OPC UA CONFIGURATION	EDIT
OPC UA Server Enable	disable
Work-around for faulty OPC UA clients that require unique browsenames	disable
Show only currently selected process data groups	disable
Allow OPC UA clients to write ISDU data	disable
ApplicationName	
ApplicationURI	
Username	
Password	[password empty]
Server Certificate Source	None [encryption disabled]
Server Certificate	[empty]
Server Private Key	[empty]
Client Authentication Certificate #1	[empty]
Client Authentication Certificate #2	[empty]

Figure 4.4 OPC UA settings

4. Open the drop-down menu and click **enable** for **OPC UA Server Enable**.

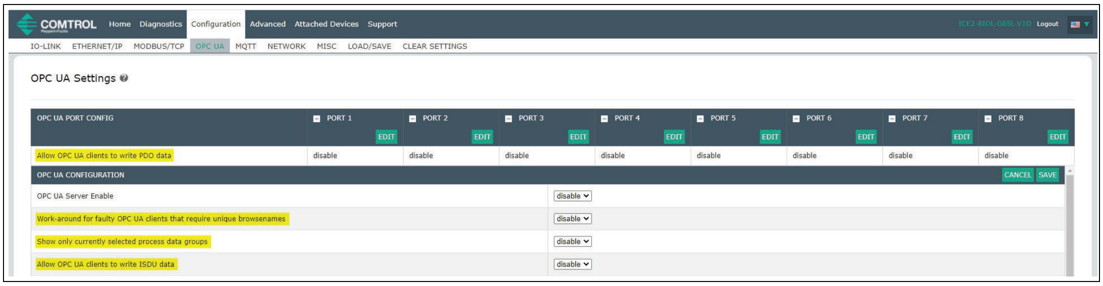


OPC UA CONFIGURATION	
OPC UA Server Enable	disable
Work-around for faulty OPC UA clients that require unique browsenames	
Show only currently selected process data groups	disable

Figure 4.5 Enabling OPC UA server

Additional Parameter Settings

Depending on the software version, there will be different parameter settings for OPC UA. Below are the available options for OPC UA configuration:



OPC UA Settings

OPC UA PORT CONFIG	PORT 1	PORT 2	PORT 3	PORT 4	PORT 5	PORT 6	PORT 7	PORT 8
Allow OPC UA clients to write PDO data	disable	disable	disable	disable	disable	disable	disable	disable

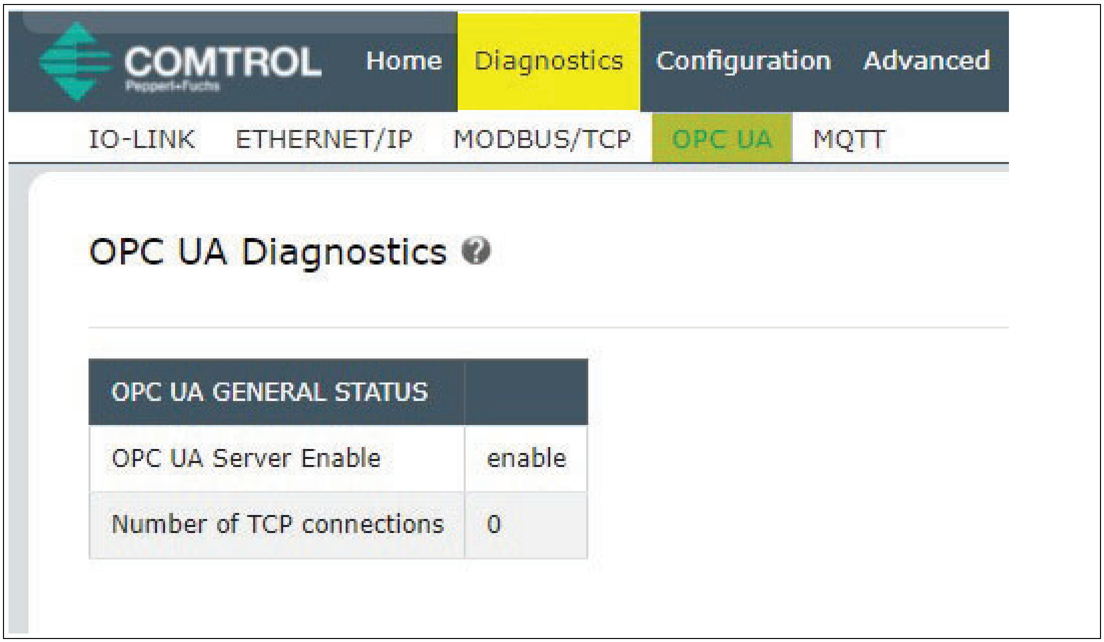
OPC UA CONFIGURATION	
OPC UA Server Enable	disable
Work-around for faulty OPC UA clients that require unique browsenames	disable
Show only currently selected process data groups	disable
Allow OPC UA clients to write ISDU data	disable

Figure 4.6 OPC UA configuration options

The following table provides information for OPC UA Configuration Settings:

Options	OPC UA Configuration Descriptions
OPC UA PORT CONFIGURATION	
Allow OPC UA clients to write PDO data (Default = disable)	Determines whether OPC UA clients are allowed to write PDO data to the IO-Link devices
OPC UA CONFIGURATION	
Allow OPC UA clients to write PDO data (Default = disable)	This option controls whether or not the OPC UA server runs on the IO-Link master.
Workaround for faulty OPC UA clients that require unique browsenames (Default = disable)	Enables an alternative set of browsenames where each node's browsename is unique. Normally only browsepaths are required to be unique.
Show only currently selected process data groups (Default = disable)	Some slaves have multiple possible layouts for PDI/PDO data, and the user selects the active layout when configuring the slave device. If this option is set to "enable", then only the currently active PDI/PDO layout will be present in the OPC UA object tree. If this option is "disable" (the default) the OPC UA object tree will contain all possible PDI/PDO layouts, and the OPC UA client must select the correct one.
Allow OPC UA clients to write ISDU data (Default = disable)	Determines whether OPC UA clients are allowed to write ISDU data to the IO-Link devices.

To check the number of TCP connections, click on **Diagnostics | OPC UA**.

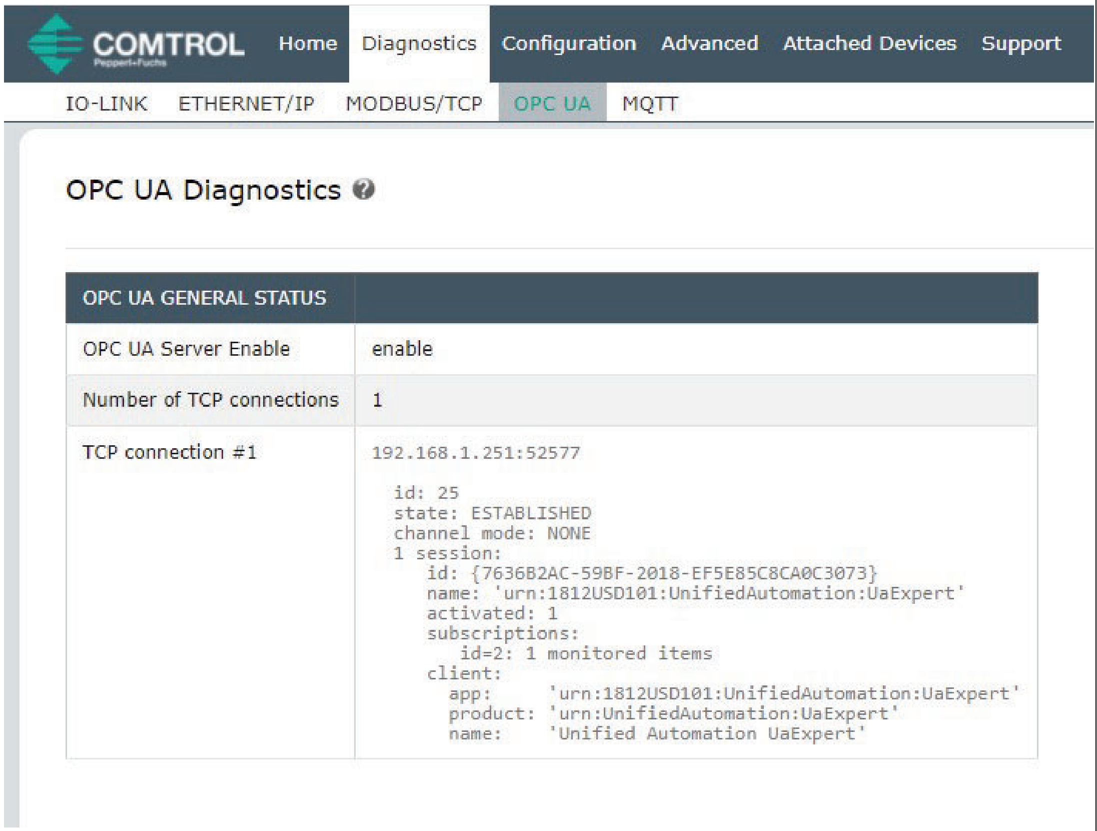


OPC UA Diagnostics

OPC UA GENERAL STATUS	
OPC UA Server Enable	enable
Number of TCP connections	0

Figure 4.7 Diagnostics window

If you have a connection (after completion of next section), this page will display:



OPC UA Diagnostics

OPC UA GENERAL STATUS	
OPC UA Server Enable	enable
Number of TCP connections	1
TCP connection #1	192.168.1.251:52577

```
192.168.1.251:52577

  id: 25
  state: ESTABLISHED
  channel mode: NONE
  1 session:
     id: {7636B2AC-59BF-2018-EF5E85C8CA0C3073}
     name: 'urn:1812USD101:UnifiedAutomation:UaExpert'
     activated: 1
     subscriptions:
        id=2: 1 monitored items
     client:
       app:     'urn:1812USD101:UnifiedAutomation:UaExpert'
       product: 'urn:UnifiedAutomation:UaExpert'
       name:    'Unified Automation UaExpert'
```

Figure 4.8 Diagnostics window with TCP connection displayed

5 Client and Server: OPC UA

Before you connect to the client, review the following summary of how our IO-Link master communicates with your OPC UA client of choice.

Overview

The IO-Link master acts as the OPC UA server. The OPC UA client can be your choice of either HMI, SCADA, or a PC application. Once the client is ready for the server, you will be required to enter an Endpoint to access the server, such as typing in the IP address of the server. Your client connects to the OPC UA server's endpoint to communicate. Pepperl+Fuchs does not use a Discovery Server between the client and server.

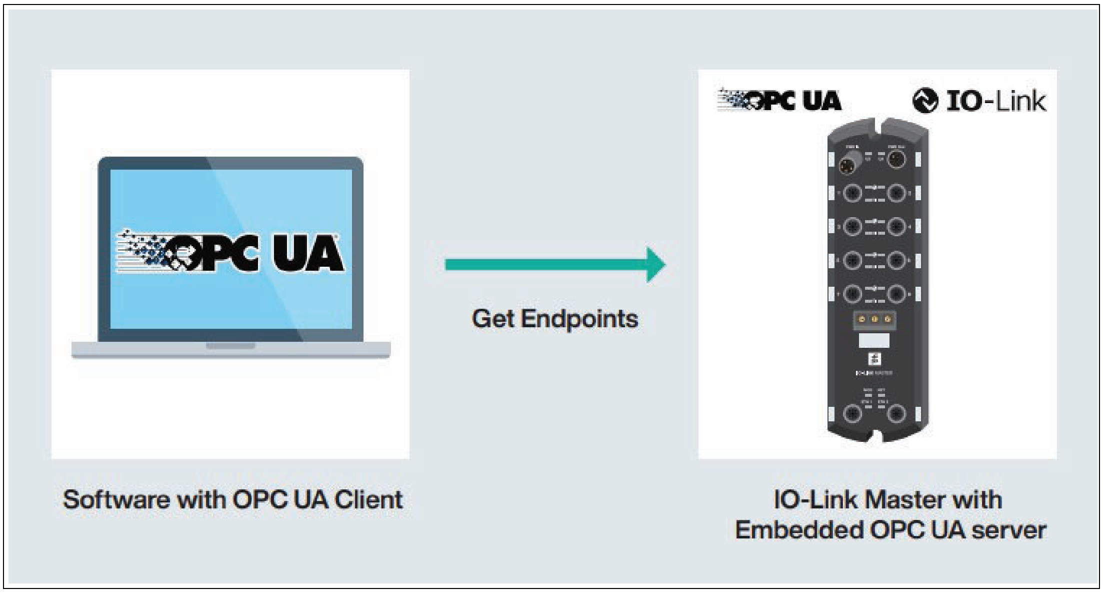


Figure 5.1 OPC UA server connection

The design behind the connection isn't important at this time; however, it helps explain how your IO-Link master is working behind the scenes.

In order to start viewing tags and data from your IO-Link master, you'll need to choose your OPC UA client. For this document, UaExpert will be the example program.

6 Using an OPC UA Client to Connect to the IOLM Server

The material presented is solely for educational purposes to show the connection(s) between the OPC UA client, OPC UA server, and sensors/actuators.

UaExpert will be the example application for the OPC UA client—it's free to download and quick to use. To download UaExpert, use this link:

https://www.unified-automation.com/downloads/undefined

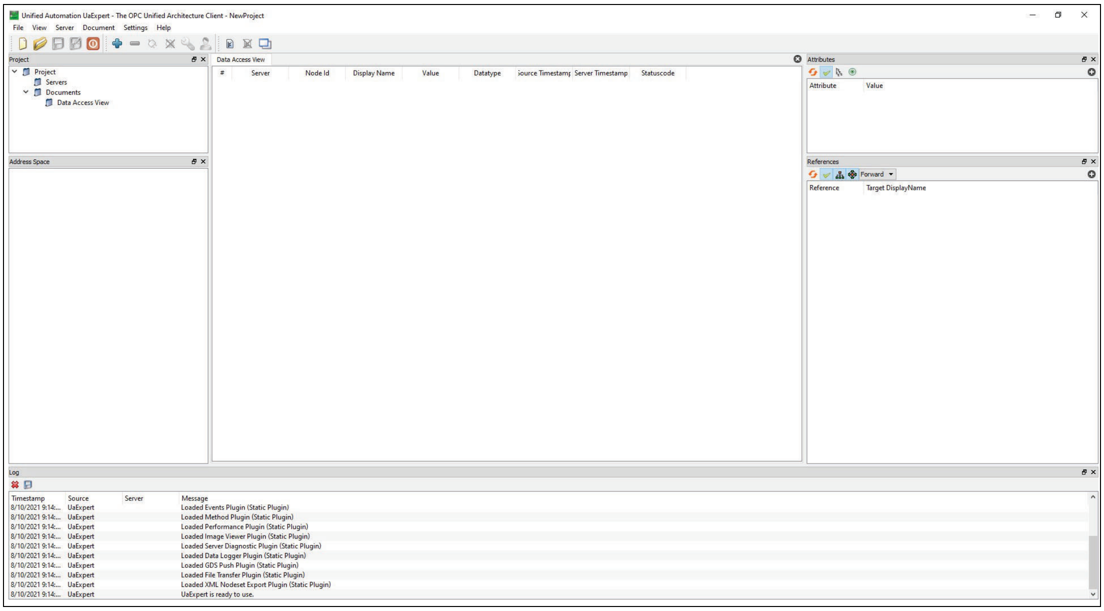
You will need to create an account before downloading and you will be prompted to enter additional information when opening the application for the first time.

There are two method that can be used to connect to the server:

- **Method One** is for general OPC UA client connections;
- **Method Two** is designed specifically for UaExpert.

Method One: Connecting with OPC Clients

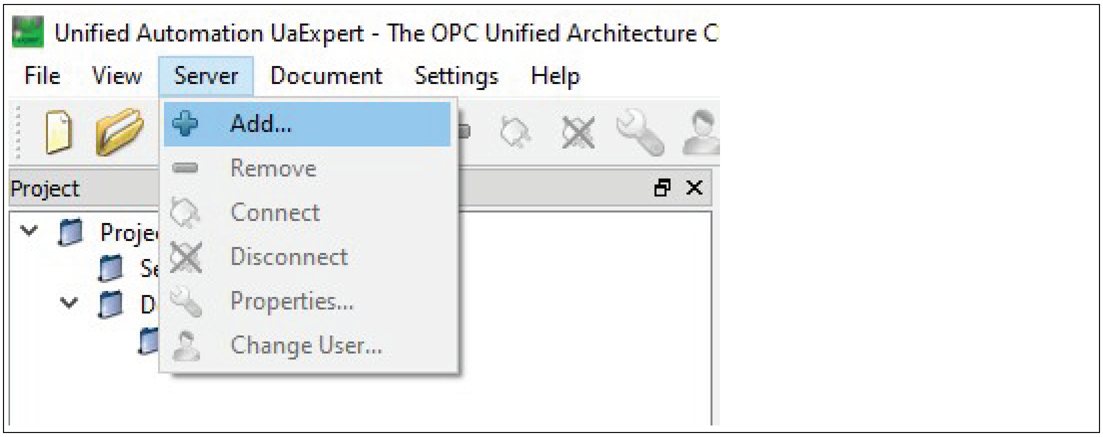
1. Once you have UaExpert (or your OPC UA client of choice) downloaded, your starting client will appear like the image below. You may have to drag some windows around to get the same exact appearance.



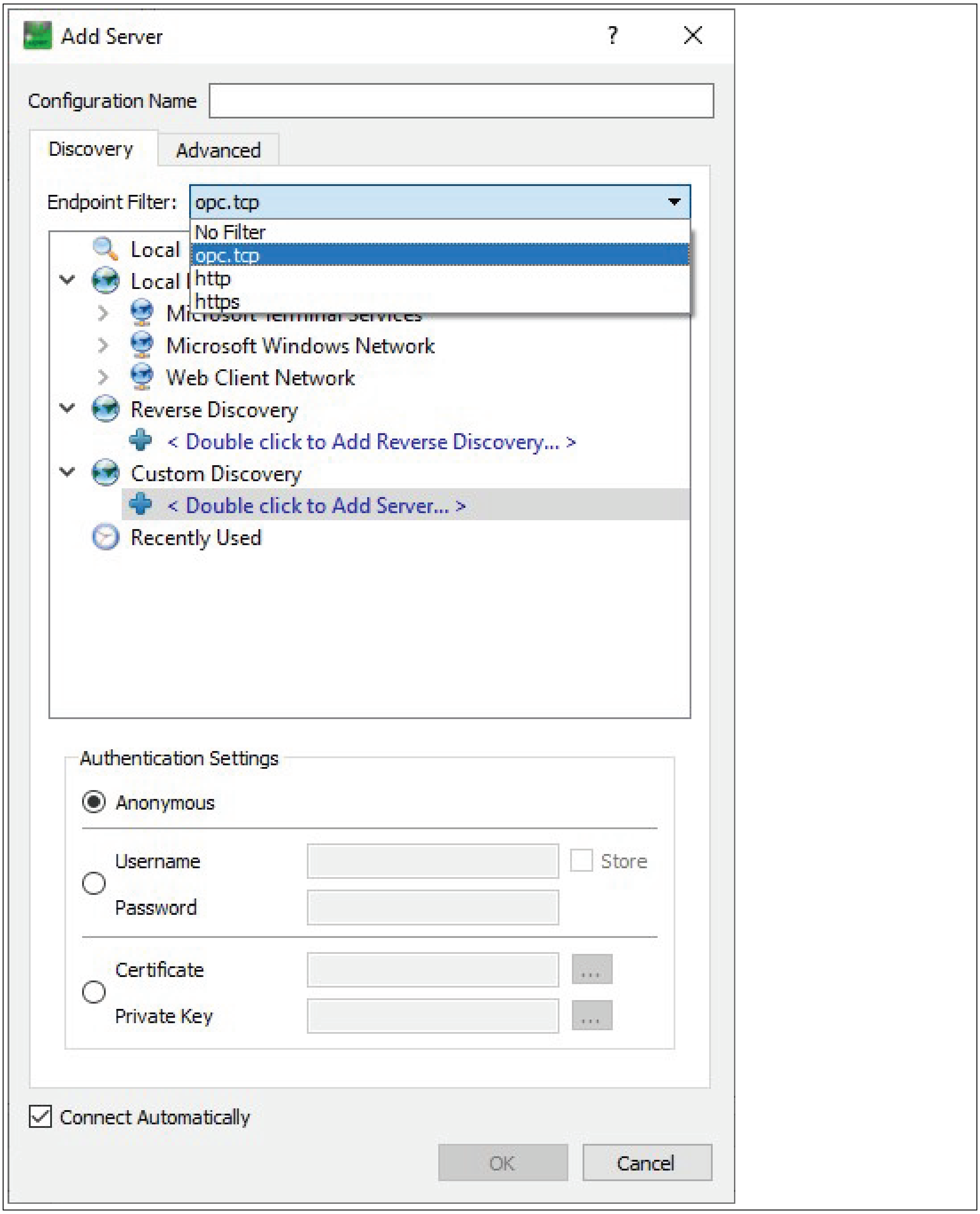
```
Timestamp          Source      Server      Message
8/10/2021 9:14:... UaExpert                Loaded Events Plugin (Static Plugin)
8/10/2021 9:14:... UaExpert                Loaded Method Plugin (Static Plugin)
8/10/2021 9:14:... UaExpert                Loaded Performance Plugin (Static Plugin)
8/10/2021 9:14:... UaExpert                Loaded Image Viewer Plugin (Static Plugin)
8/10/2021 9:14:... UaExpert                Loaded Server Diagnostic Plugin (Static Plugin)
8/10/2021 9:14:... UaExpert                Loaded Data Logger Plugin (Static Plugin)
8/10/2021 9:14:... UaExpert                Loaded GDS Push Plugin (Static Plugin)
8/10/2021 9:14:... UaExpert                Loaded File Transfer Plugin (Static Plugin)
8/10/2021 9:14:... UaExpert                Loaded XML Nodeset Export Plugin (Static Plugin)
8/10/2021 9:14:... UaExpert                UaExpert is ready to use.
```

Figure 6.1 OPC UA client dashboard

2. Next, find your "**add server**" option. In UA Expert, that will be located in the top left under **Server**. Select **Add**.



3. This will be your endpoint to connect to the server. When working with OPC via TCP, choose the **opc.tpc** option.



Add Server

Configuration Name

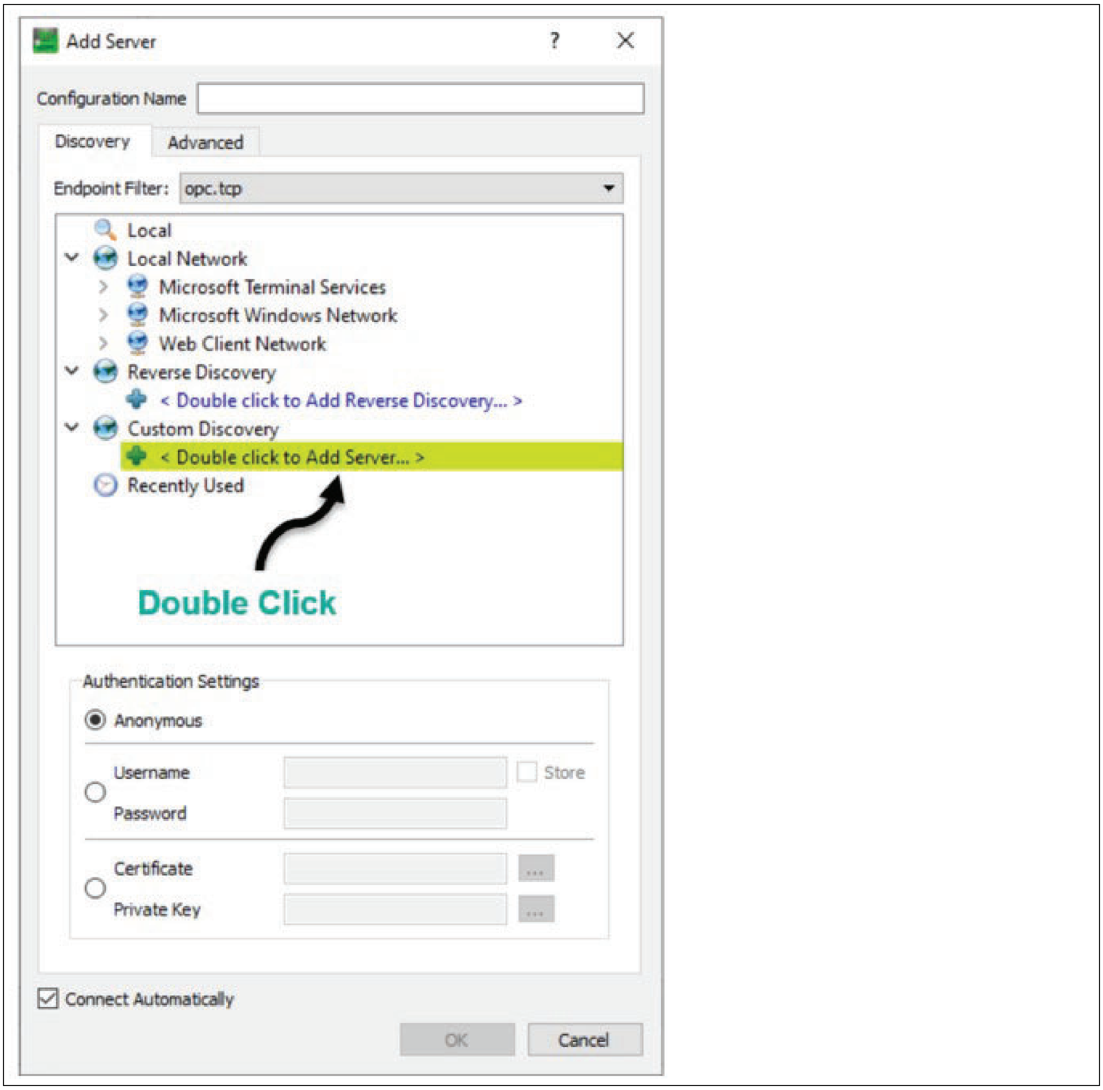
Endpoint Filter: opc.tcp

Authentication Settings

● Anonymous

☑ Connect Automatically

4. Discover the server through the UA Client. Double-click on **<Double click to Add Server...>** under the **Custom Discovery** tab.



Add Server

Configuration Name

Discovery | Advanced

Endpoint Filter: opc.tcp

- Local
- Local Network
  - Microsoft Terminal Services
  - Microsoft Windows Network
  - Web Client Network
- Reverse Discovery
  - < Double click to Add Reverse Discovery... >
- Custom Discovery
  - < Double click to Add Server... >
- Recently Used

Double Click

Authentication Settings

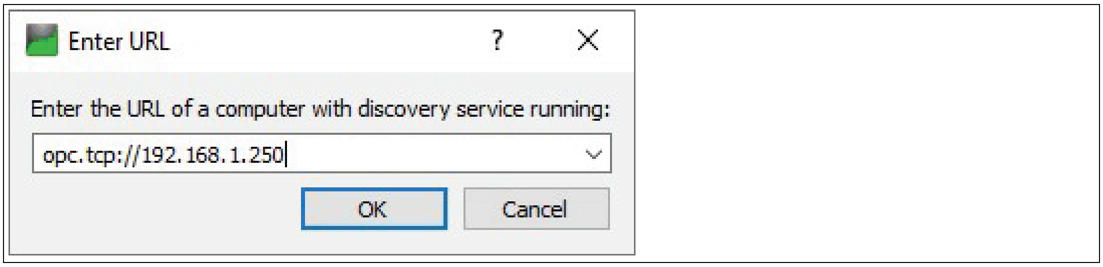
● Anonymous

○ Username / Password □ Store

○ Certificate / Private Key

☑ Connect Automatically

5. Enter the URL of your IO-Link master and click **OK**.

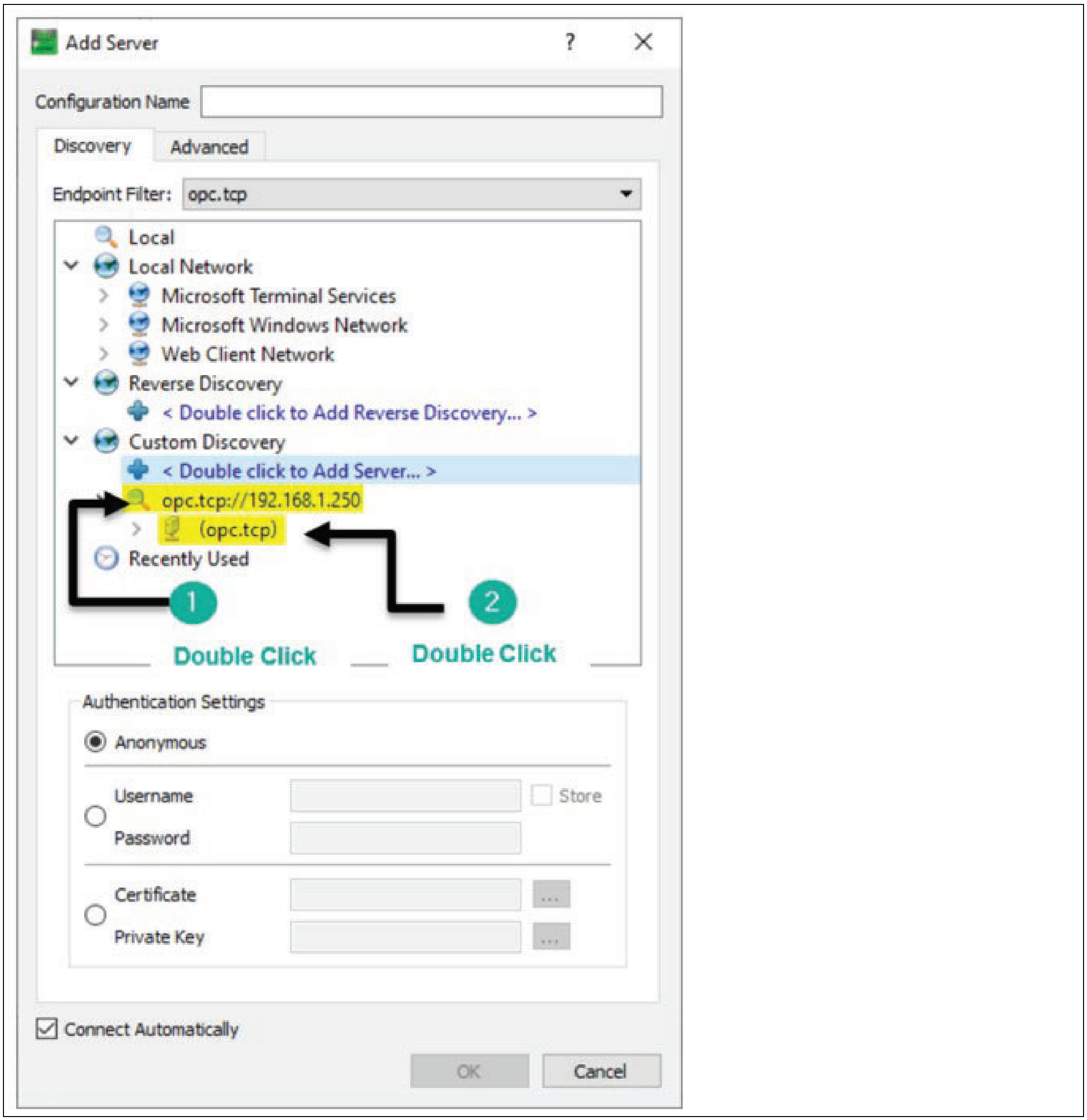


Enter URL

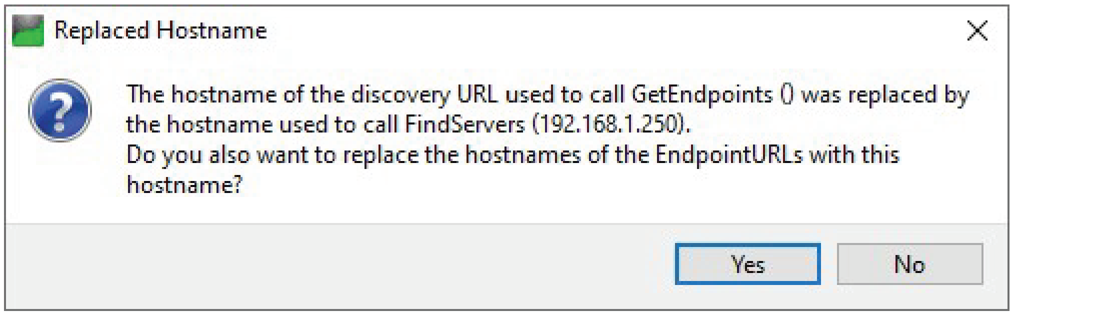
Enter the URL of a computer with discovery service running:

opc.tcp://192.168.1.250

6. Double-click the IP address you just added. Then double-click the endpoint name. Since the master has not yet been named within the OPCUA Configuration tab via the Web Server, it is simply named (opc.tcp).



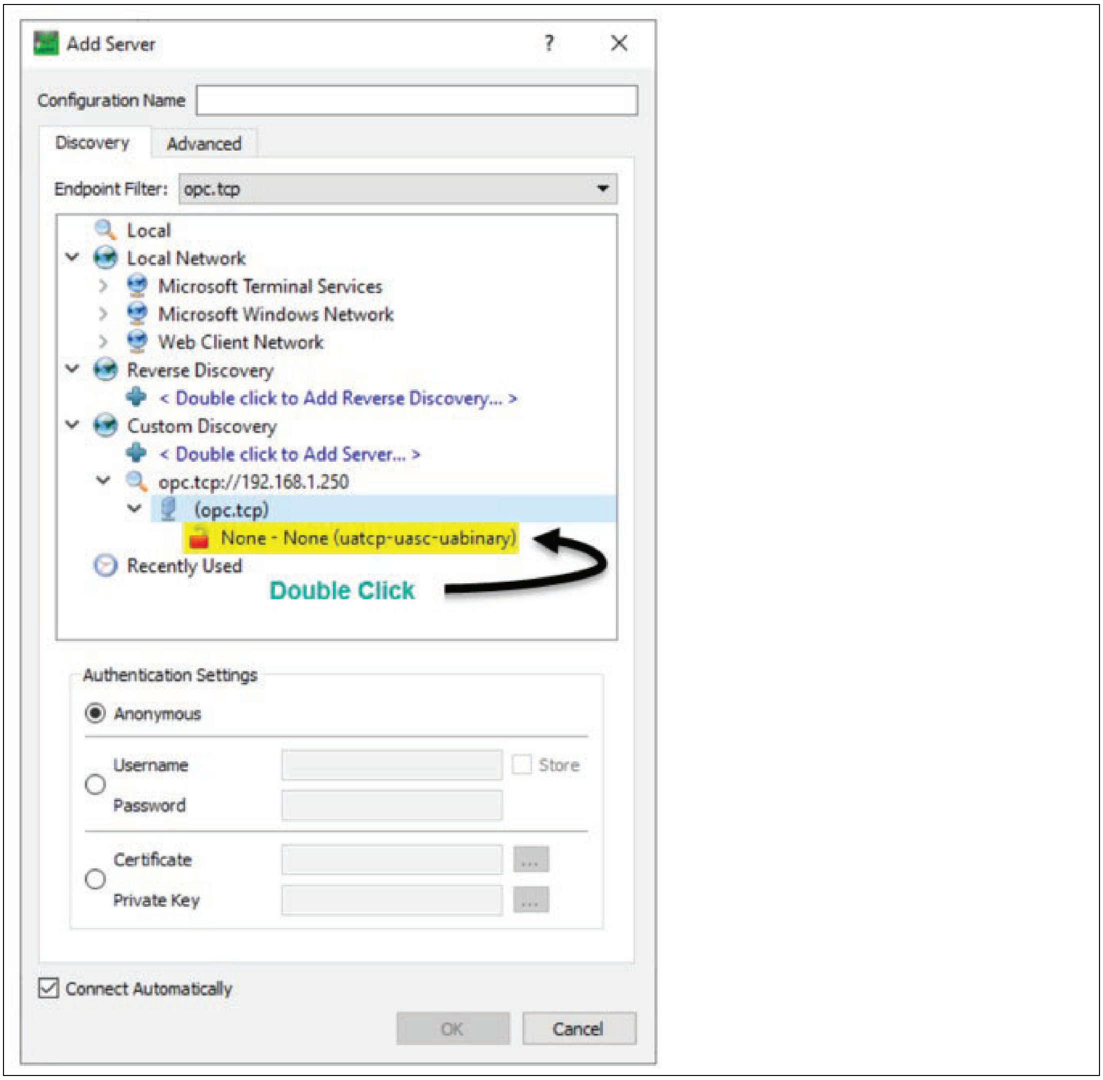
7. Replace the hostname of the connected endpoint with the IP address of the master. Click **Yes** to continue.



Replaced Hostname

The hostname of the discovery URL used to call GetEndpoints () was replaced by the hostname used to call FindServers (192.168.1.250).
Do you also want to replace the hostnames of the EndpointURLs with this hostname?

8. Double-click the lock symbol to connect.



Add Server

Configuration Name

Discovery | Advanced

Endpoint Filter: opc.tcp

- Local
- Local Network
  - Microsoft Terminal Services
  - Microsoft Windows Network
  - Web Client Network
- Reverse Discovery
  - < Double click to Add Reverse Discovery... >
- Custom Discovery
  - < Double click to Add Server... >
  - opc.tcp://192.168.1.250
    - (opc.tcp)
      - None - None (uatcp-uasc-uabinary)
- Recently Used

Double Click

Authentication Settings

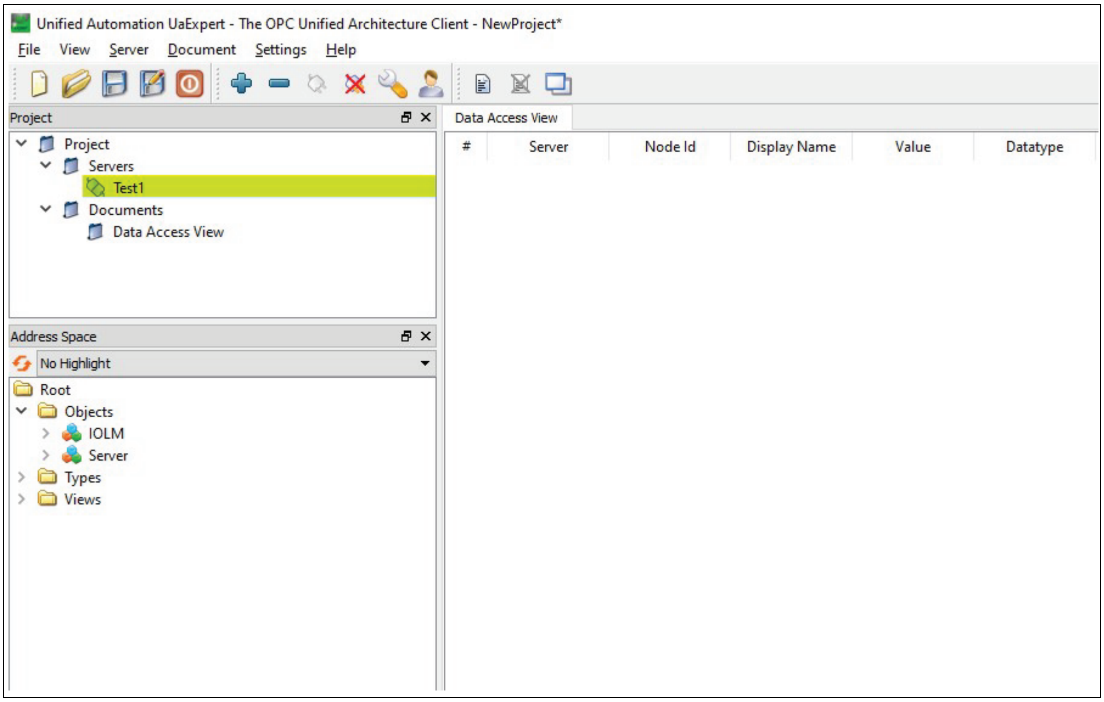
● Anonymous

○ Username / Password □ Store

○ Certificate / Private Key

☑ Connect Automatically

9. The server is now connected! A blank name for the connection is acceptable, but may be renamed by right-clicking the connection icon, going into **Properties**, and changing the **Configuration Name** field. In the example below, the connection was named "**Test1**."


Unified Automation UaExpert - The OPC Unified Architecture Client - NewProject*
File View Server Document Settings Help
Project
Project
Servers
Test1
Documents
Data Access View
Address Space
No Highlight
Root
Objects
IOLM
Server
Types
Views
Data Access View
# Server Node Id Display Name Value Datatype

Additional Steps:

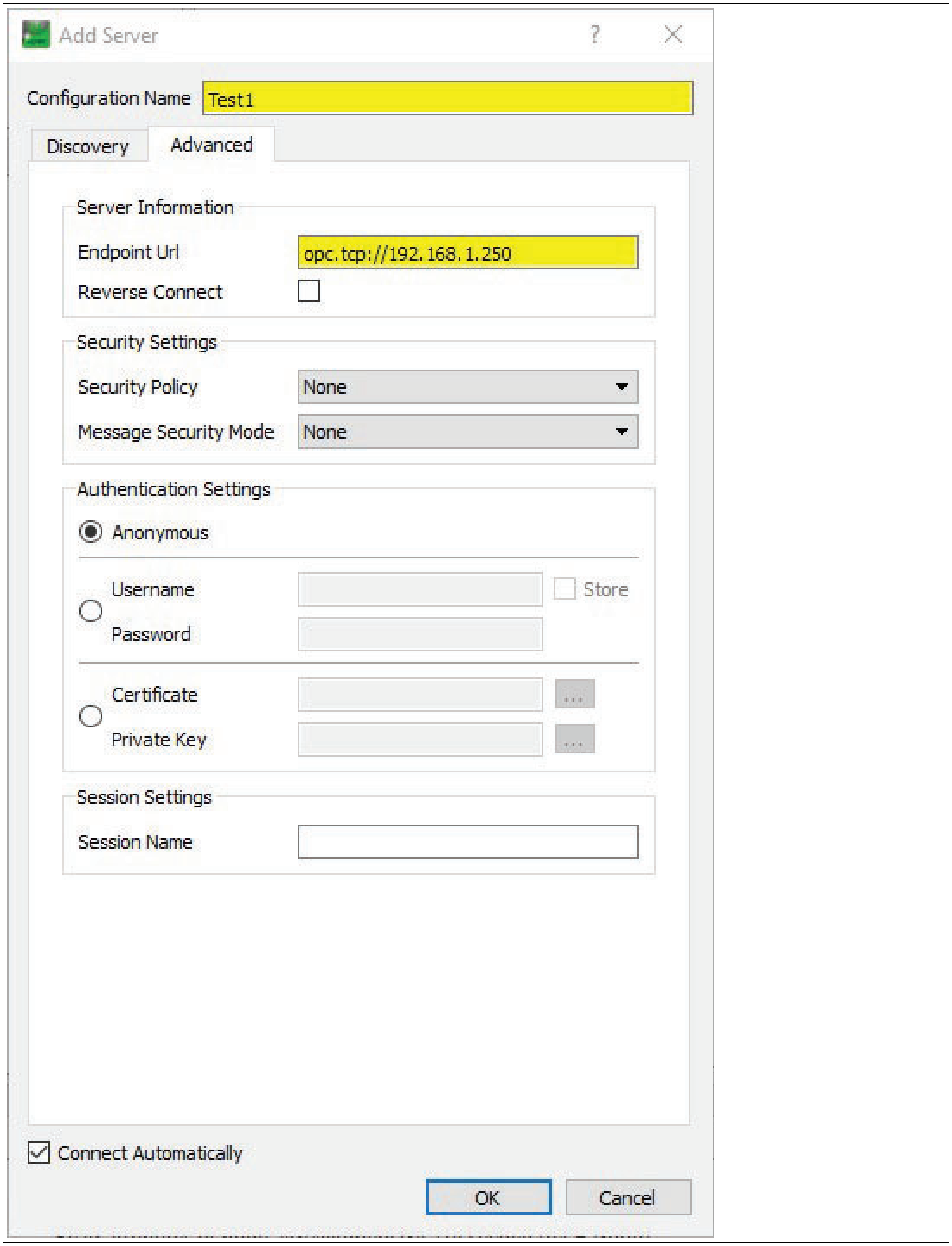
1. If your server and client are not connected yet, please continue to the next page
2. If you have successfully connected, see "Viewing Tags and Data on the Client" on page 26 to view tags inside your OPC UA client.
3. If you want to find a specific tag, see "How to Read Tag Values: Practical Application" on page 30 .

Method Two: Connecting with UaExpert

If the server did not connect, try another strategy, such as inputting the Endpoint URL directly for the server. In this case, opc.tcp is needed in front of the IO-Link master's IP address. You can find this on the **Advanced** tab for "**Add Server**." This **will** directly connect you to the OPC UA server for your IO-Link master. A configuration name is needed to connect using this method.

Note

You may have to type "port:4840" after the IP address.



Configuration Name: Test1

Server Information

- Endpoint Url: opc.tcp://192.168.1.250
- Reverse Connect: □

Security Settings

- Security Policy: None
- Message Security Mode: None

Authentication Settings

- ● Anonymous
- ○ Username / Password — □ Store
- ○ Certificate / Private Key

Session Settings

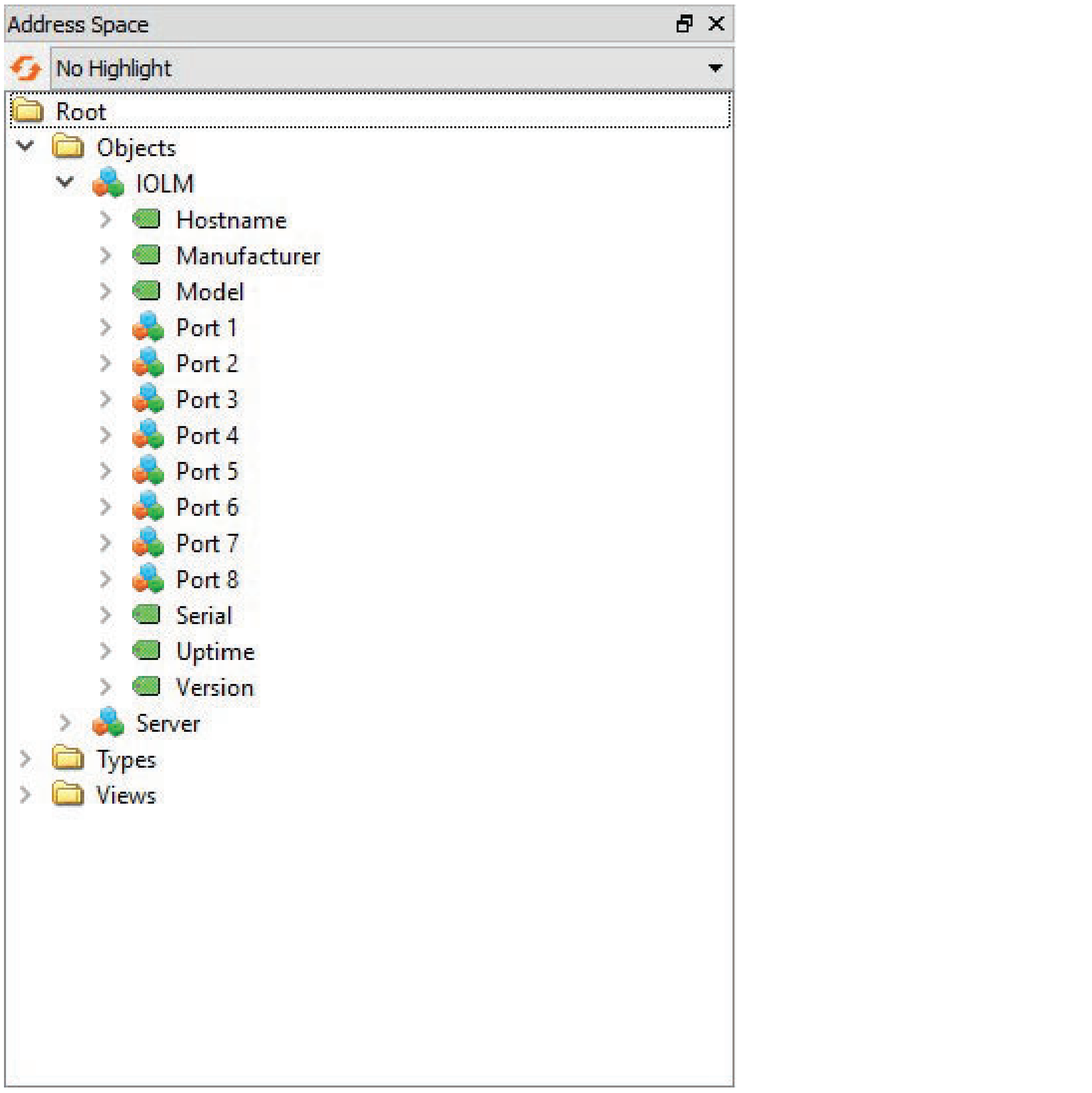
- Session Name:

☑ Connect Automatically

Viewing Tags and Data on the Client

After successfully connecting your client and server, you can now start viewing your tags and data in the client, using the following steps:

1. Look for **IOLM** in the Address Space on the left side. The Address Space will provide all the IDs, Cycle Times, ISDU, POI information, and additional tags. Use the drop-down icon to view an expanded look as shown below:

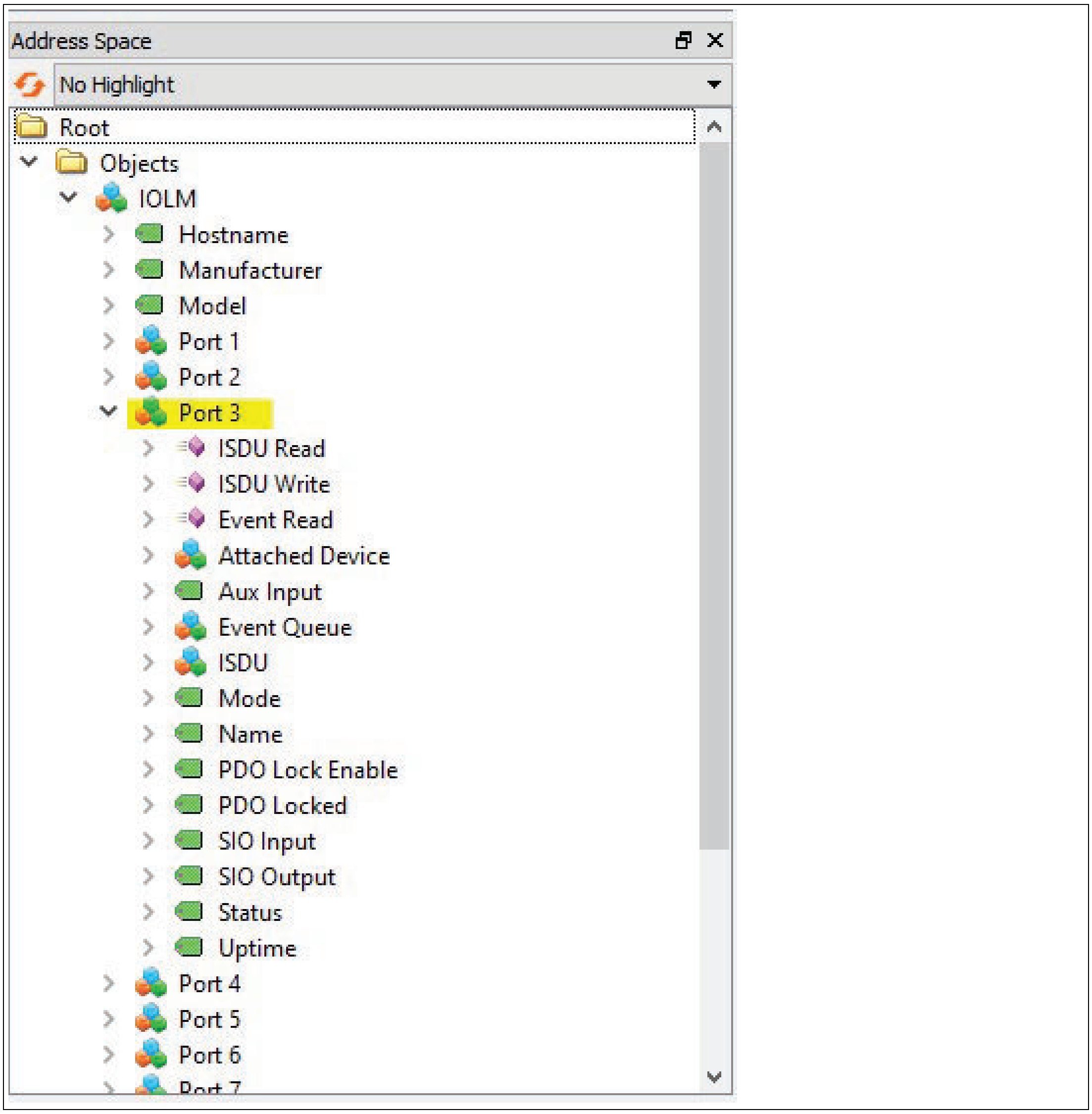


Address Space

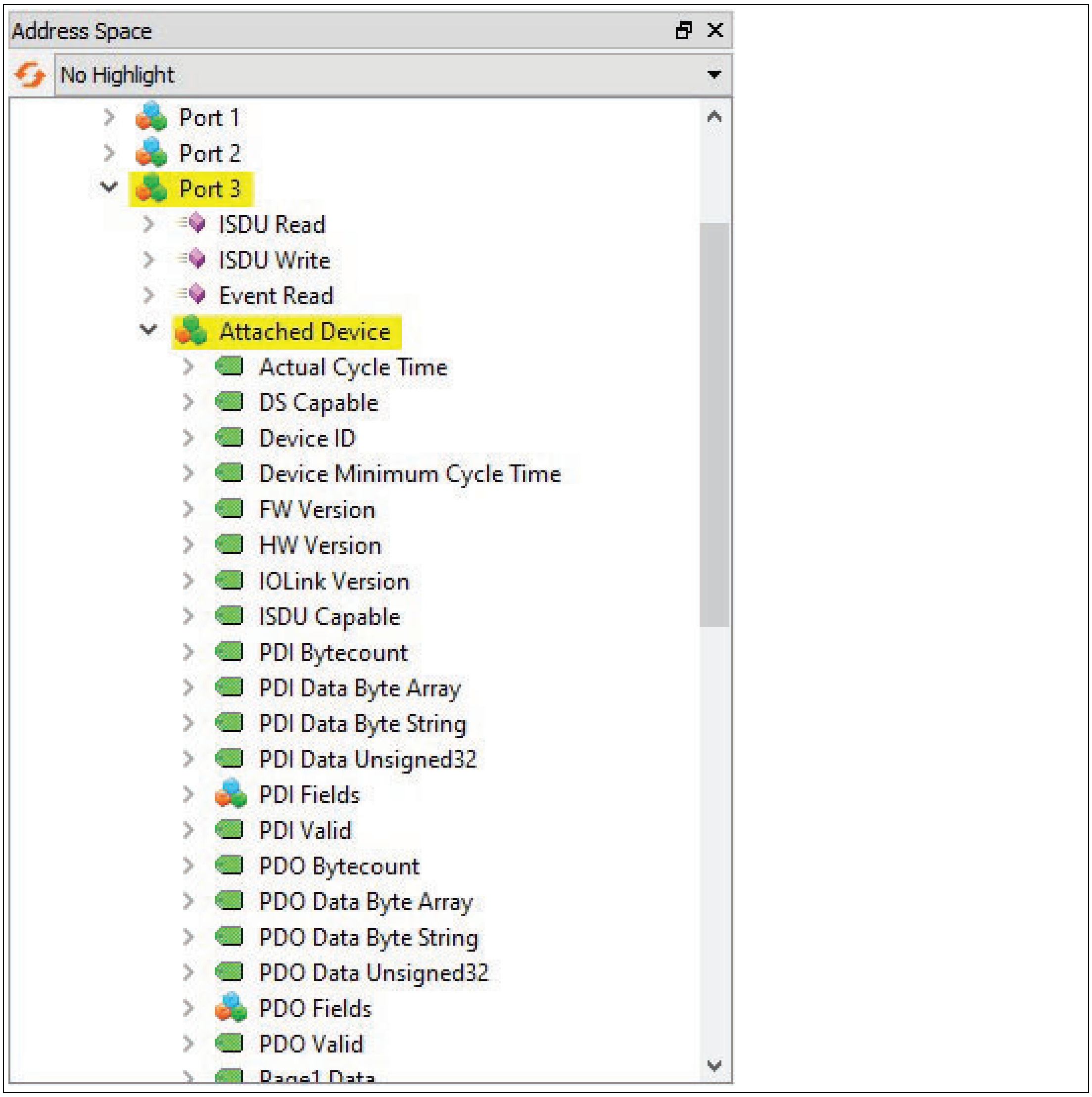
No Highlight

```
Root
⌄ Objects
    ⌄ IOLM
        › Hostname
        › Manufacturer
        › Model
        › Port 1
        › Port 2
        › Port 3
        › Port 4
        › Port 5
        › Port 6
        › Port 7
        › Port 8
        › Serial
        › Uptime
        › Version
    › Server
› Types
› Views
```

2. Expand one of your **Ports** that a sensor is attached to. In this example, a Pepperl+Fuchs Diffuse mode photoelectric sensor (OBD1400-R200-2EP-IO-V1) is attached to Port 3. To view data, the IODD files for the device in use MUST be uploaded to the web server.

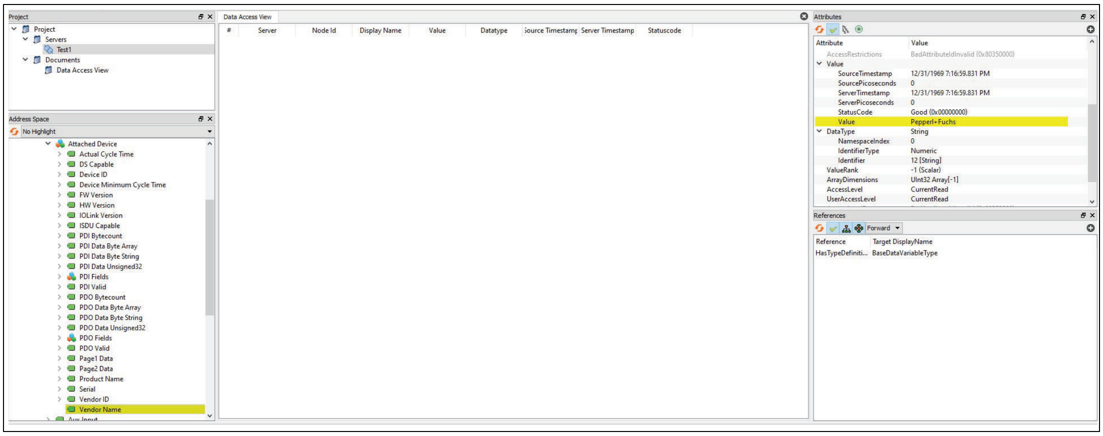


3. Expand **Attached Device**. This will be the destination for all your tags you want to view on your OPC UA client for that specific port.



- Port 1
- Port 2
- Port 3
  - ISDU Read
  - ISDU Write
  - Event Read
  - Attached Device
    - Actual Cycle Time
    - DS Capable
    - Device ID
    - Device Minimum Cycle Time
    - FW Version
    - HW Version
    - IOLink Version
    - ISDU Capable
    - PDI Bytecount
    - PDI Data Byte Array
    - PDI Data Byte String
    - PDI Data Unsigned32
    - PDI Fields
    - PDI Valid
    - PDO Bytecount
    - PDO Data Byte Array
    - PDO Data Byte String
    - PDO Data Unsigned32
    - PDO Fields
    - PDO Valid
    - Page1 Data

4. Now, you can explore the different tags and view any data you wish to select. To test if the IO-Link master is properly working with its respective sensor(s), click on **Vendor Name** tag (left side), then look over on the **Attributes** (right side). Under **Value**, the Vendor Name of our device (Pepperl+Fuchs) for the port confirms that the client, server, and port are all communicating—you can now freely view the data for any tag.



- Attached Device
  - Actual Cycle Time
  - DS Capable
  - Device ID
  - Device Minimum Cycle Time
  - FW Version
  - HW Version
  - IOLink Version
  - ISDU Capable
  - PDI Bytecount
  - PDI Data Byte Array
  - PDI Data Byte String
  - PDI Data Unsigned32
  - PDI Fields
  - PDI Valid
  - PDO Bytecount
  - PDO Data Byte Array
  - PDO Data Byte String
  - PDO Data Unsigned32
  - PDO Fields
  - PDO Valid
  - Page1 Data
  - Page2 Data
  - Product Name
  - Serial
  - Vendor ID
  - Vendor Name

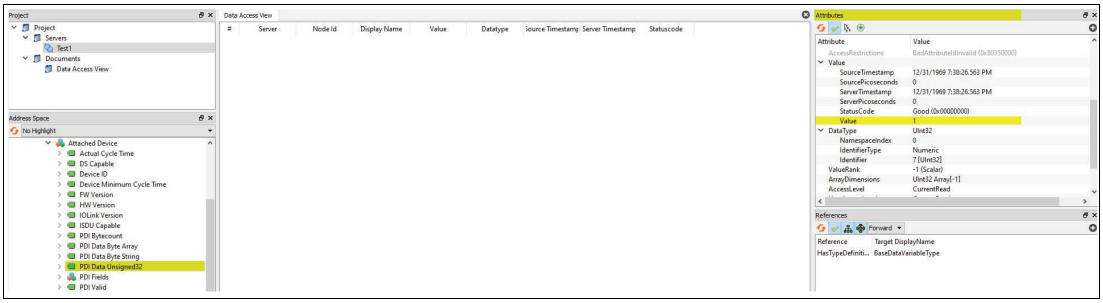
#	Server	Node Id	Display Name	Value	Datatype	Source Timestamp	Server Timestamp	Statuscode

Attribute	Value
AccessRestrictions	BadAttributeIdInvalid (0x80350000)
Value	
SourceTimestamp	12/31/1969 7:16:59.831 PM
SourcePicoseconds	0
ServerTimestamp	12/31/1969 7:16:59.831 PM
ServerPicoseconds	0
StatusCode	Good (0x00000000)
Value	Pepperl+Fuchs
DataType	String
NamespaceIndex	0
IdentifierType	Numeric
Identifier	12 [String]
ValueRank	-1 (Scalar)
ArrayDimensions	UInt32 Array[-1]
AccessLevel	CurrentRead
UserAccessLevel	CurrentRead

Reference	Target DisplayName
HasTypeDefiniti...	BaseDataVariableType

How to Read Tag Values: Practical Application

1. For the Pepperl+Fuchs OBD1400-R200-2EP-IO-V1 Photoelectric sensor, let's find out the value it is reading. Choose the tag PDI Data Unsigned32. The attributes will now change. Under **Values**, you can see the static value is 1. This means the sensor is producing an output, which you can confirm by checking the output status indicator on the sensor. If the static value is 0, this means the sensor is not producing an output.

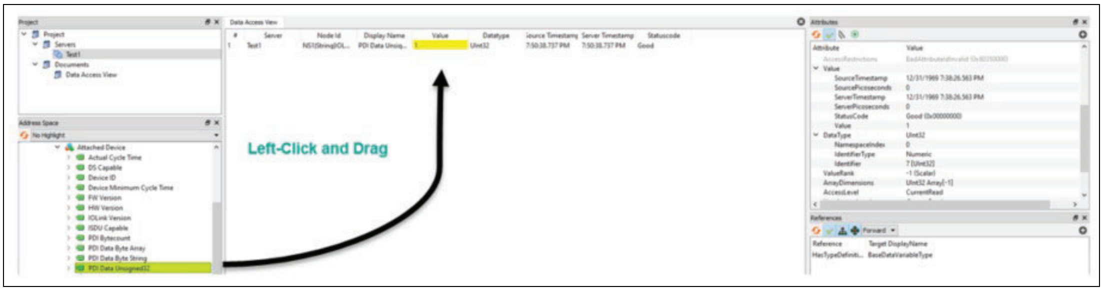


#	Server	Node Id	Display Name	Value	Datatype	Source Timestamp	Server Timestamp	Statuscode

Attribute	Value
AccessRestrictions	BadAttributeIdInvalid (0x80350000)
Value	
SourceTimestamp	12/31/1969 7:38:26.563 PM
SourcePicoseconds	0
ServerTimestamp	12/31/1969 7:38:26.563 PM
ServerPicoseconds	0
StatusCode	Good (0x00000000)
Value	1
DataType	UInt32
NamespaceIndex	0
IdentifierType	Numeric
Identifier	7 [UInt32]
ValueRank	-1 (Scalar)
ArrayDimensions	UInt32 Array[-1]
AccessLevel	CurrentRead

Reference	Target DisplayName
HasTypeDefiniti...	BaseDataVariableType

2. To view the data change in real time, you'll need to use the Data Access View tab (the center panel in the image below). This will give you live data while testing your sensor. To start using the Data Access View tab, pick the tag you want to track data for. In this case, PDI Data Unsigned32 will work—drag it to the Data Access View tab and it will now display in real time. Watch the value change as you move the sensor. The value will correlate to the output light on the sensor.



#	Server	Node Id	Display Name	Value	Datatype	Source Timestamp	Server Timestamp	Statuscode
1	Test1	NS1|String|IOL...	PDI Data Unsig...	1	UInt32	7:50:38.737 PM	7:50:38.737 PM	Good

6.1 IO-Link Master Tags

Tags are "Read-Only" unless noted as Read/Write (RW).

Tag Names: Attached Devices	IO-Link Tag Description
Actual Cycle Time	When the master sends a packet and receives it. This is noted as a period rather than a frequency (ex: 5ms). The Actual Cycle Time is a negotiation between the IO-Link master and the IO-Link sensor / actuator; this time will vary depending on the device and master. It will be the greater of either the master or device min cycle time.
DS Capable	Determines if the attached device is data storage capable; Boolean. Data Storage allows saving and loading of multiple parameters as a single object. This object can be uploaded or downloaded to facilitate sensor cloning or effortless replacement of a bad sensor.
Device ID	A numerical identifier set by the device manufacturer into the sensor hardware that cannot be changed. Useful for basic identification and criteria to allow certain automated options such as automatic data storage download.
Device Min Cycle Time	At the low level hardware IO-Link information exchange, the Device Min Cycle Time is the minimum period of time at which the device can run. It may not necessarily be used, but rather an "Actual Cycle Time" is negotiated during IOL handshake which is acceptable to both master and device.
FW Version	Displayed under IOLM properties, this is the Firmware of the attached device if displayed under "Port X/Attached device." It cannot be changed.
HW Version	Displayed under IOLM properties, this is the Hardware of the attached device if displayed under "Port X/Attached device." It cannot be changed.
IO-Link Version	This is the version of the IO-Link spec that the device conforms to.Version 1.0 or 1.1 is reported.
ISDU Capable	Implementation of the Index Service Data Unit; this tag is true when the device supports ISDUs.
PDI Bytecount	Size of the input process data payload
PDI Data Byte Array	One of several different format options to read PDI; formatted in an array.
PDI Data Byte String	One of several different format options to display PDI; String
PDI Data Unsigned32:	One of several different format options to read PDI; simple data type of UInt32 (4 bytes).

Tag Names: Attached Devices	IO-Link Tag Description
PDI Files	Collection of tags; only available if a valid IODD file has been loaded for that particular device. Offers "smart automatic formatting" of the PDI payload by parsing PDI. You may observe, for example, a raw 32 bit value sorted into a flow rate and a temperature complete with engineering units, and the tags will be automatically labeled as "flow rate" and "temperature."
PDI Valid	This tag is true when the device is sending valid PDI data. The device (sensor) determines if the data is valid.
PDO Bytecount	Size of the output process data payload.
PDO Data Byte Array (RW)	One of several different format options to read PDO; formatted in an array.
PDO Data Unsigned	One of several different format options to read PDO; simple data type with of UInt32 (4 bytes).
PDO Fields	Collection of tags; only available if a valid IODD file has been loaded for that particular device. Offers "smart automatic formatting" of the PDO payload by parsing PDO.
Page 1 Data	ISDU Index 0; Index that tells critical information on the device such as Min Cycle Time, etc.
Page 2 Data	Used for devices that are minimalistic and do not implement ISDU. Used to store parameter data(16 bytes).
Product Name	Often called "model" or "family of devices." This tag is any string of alphanumeric characters. Ex: TD2807, Q4X.
Serial	Numerical Identifier assigned by the device (sensor) manufacturer at build time; unchangeable and unique for each device built.
Vendor ID	Assigned to each vendor in the IO-Link community. The Vendor ID will appear the same for every IO-Link compatible product made by that OEM.
Vendor Name	The common name of the vendor.(Ex: Pepperl-Fuchs)

Tag Names: ISDU	Description
Data (RW)	Data as ByteString. Multiple Bytes.
Data08 (RW)	Data as one Byte.
Data16 (RW)	Data as UInt16; Two Bytes.
Data32 (RW)	Data as UInt32; Four Bytes.
Index (RW)	Index of ISDU to read/write.

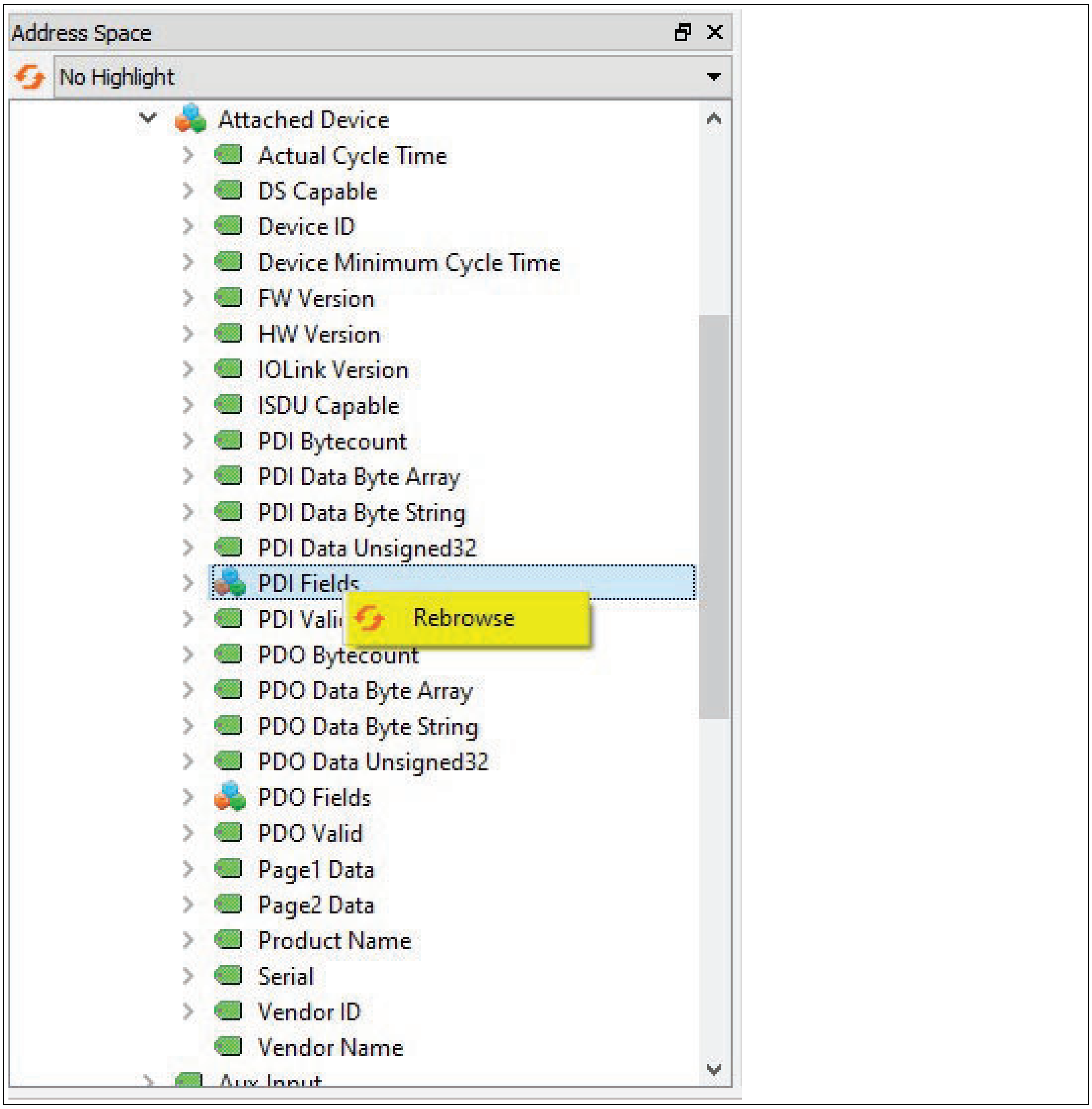
Tag Names: ISDU	Description
Request (RW)	Set to 1 for read ISDU and 2 for write ISDU. Set to 0 to clear RW.
Status (RO)	Indicates the status of the most recently executed request. 1 = Success; 2 = Failed; 0 = cleared state.
SubIndex (RW)	SubIndex of ISDU to read/write.

Tag Names: Port	Description
Aux Inout	Status of an auxiliary input; either Boolean or Binary (single bit). Pin 2 of the IOLM port.
Event Queue	Queue of device and master events. Allows events to be read using simple data types.
Event Read	Method that allows events to be read.
ISDU Read	Method for reading ISDU data.
ISDU Write	Method for writing ISDU data.
Mode	Displays the current status of the port (ex: IO-Link, digital input, digital output, reset, idle).
Name	Name of the port. Ex: IO-Link Port 3.
PDO Lock Enable	Protocol can lock PDO, if true.
PDO Locked	Protocol has PDO locked.
SIO Input	Indicates the Boolean status of pin 4 on a port that is configured to allow simple inputs (standard digital input); this would not be valid while in IOL mode.
SIO Output	Refer to SIO Input, but for simple outputs. Not valid for a port in IOL mode.
Status	Status of the port. Ex: pre-operate, operate, init.
Uptime	Amount of time the port has been actively connected to an IO-Link device.

6.2 Additional Tips for UaExpert

Rebrowsing

When changing sensors, viewing new data, or updating the environment/client/server, you can use the Rebrowse option. To do this, select a node in the Address Space and click "Rebrowse."



6.3 Contact Us

Still have questions?

This handy guide was designed to introduce you to IO-Link technology and to address many of the common questions new users have. If you have any questions or concerns not addressed in this document, a member of our staff will always be happy to help. Feel free to contact us by telephone or email and we will be sure to get back to you as soon as possible.

How to reach us...

Phone

US: +1-330-425-3555

International: +49-621-776-0

Monday–Friday: 6:00 a.m. to 8:00 p.m.

Saturday & Sunday: 10:00 a.m. to 5:00 p.m.

Email

ask@us.pepperl-fuchs.com

Online

https://www.pepperl-fuchs.com

Your automation, our passion.

Explosion Protection

- Intrinsic Safety Barriers
- Signal Conditioners
- FieldConnex® Fieldbus
- Remote I/O Systems
- Electrical Ex Equipment
- Purge and Pressurization
- Industrial HMI
- Mobile Computing and Communications
- HART Interface Solutions
- Surge Protection
- Wireless Solutions
- Level Measurement

Industrial Sensors

- Proximity Sensors
- Photoelectric Sensors
- Industrial Vision
- Ultrasonic Sensors
- Rotary Encoders
- Positioning Systems
- Inclination and Acceleration Sensors
- Fieldbus Modules
- AS-Interface
- Identification Systems
- Displays and Signal Processing
- Connectivity

Pepperl+Fuchs Quality
Download our latest policy here:

www.pepperl-fuchs.com/quality

PEPPERL+FUCHS

www.pepperl-fuchs.com