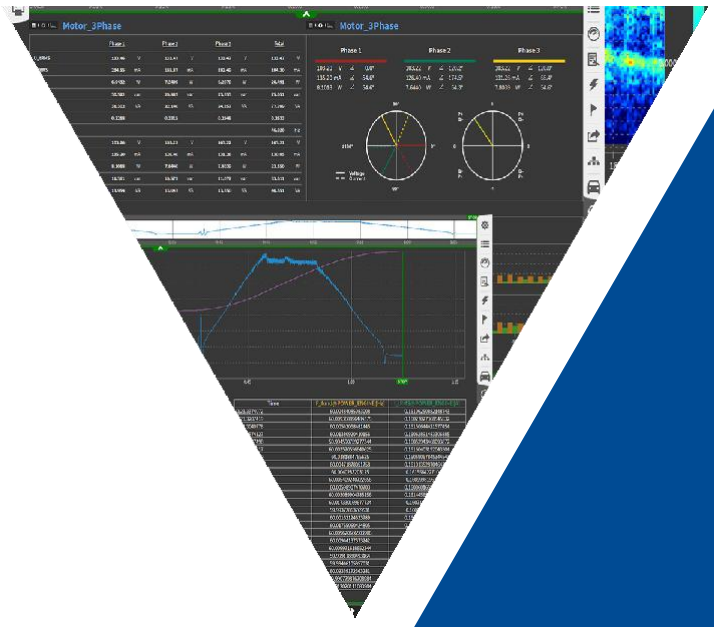
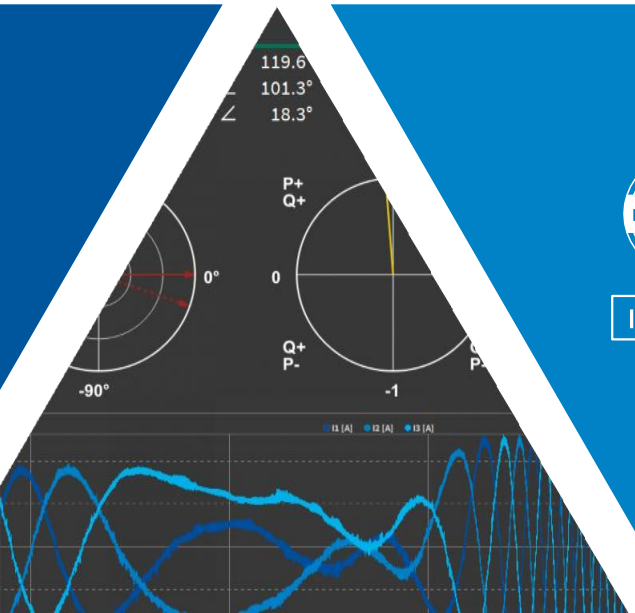
DEWETRON



OPC UA

SOFTWARE MANUAL



EN ISO 14001

ISO 9001



Copyright © DEWETRON GmbH

This document contains information which is protected by copyright. All rights are reserved. Reproduction, adaptation, or translation without prior written permission is prohibited, except as allowed under the copyright laws.

All trademarks and registered trademarks are acknowledged to be the property of their owners.

CONTENTS

1 Preface **1**

2 Document History **3**

3 OPC UA Introduction **5**
3.1 OPC UA Communication Interface 5
3.1.1 Overview 5
3.1.2 Historical Context and Evolution 5
3.1.3 DEWETRON Implementation 6

4 OPC UA Server **7**
4.1 Configuring the OPC UA Server in OXYGEN 7
4.2 Channel Parameters 8

5 OPC UA Client **9**
5.1 OPC UA Client Overview 9
5.2 Setting up an OPC UA Client 9
5.2.1 Connection handling 10
5.2.2 Security, Certificates and authentication 11
5.2.3 Channel Setup 12
5.2.4 Buffering and Update Behavior 13
5.2.5 General Usage Notes 13
5.2.6 Performance Considerations 14

CHAPTER

ONE

PREFACE

The information contained in this document is subject to change without notice.

DEWETRON GmbH (DEWETRON) shall not be liable for any errors contained in this document. DEWETRON MAKES NO WARRANTIES OF ANY KIND ABOUT THIS DOCUMENT, WHETHER EXPRESS OR IMPLIED. DEWETRON SPECIFICALLY DISCLAIMS THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. DEWETRON shall not be liable for any direct, indirect, special, incidental, or consequential damages, whether based on contract, tort, or any other legal theory, in connection with the furnishing of this document or the use of the information in this document.

Technical Support

Please contact your local authorized DEWETRON representative first for any support and service questions.

For Europe and Asia, please contact:

DEWETRON GmbH
Parkring 4, 8074 Grambach
AUSTRIA
Tel.:+43 316 3070-555
Fax: +43 316 3070-90
Email: support@dewetron.com
Web: http://www.dewetron.com
The telephone hotline is available Monday to Friday between 08:00 and 17:00 CET (GMT +1:00)

For America, please contact:

DEWETRON, Inc.
2850 South County Trail
East Greenwich, RI 02818
USA
Tel.: +1 401 398 7963
Fax: +1 401 284 3755
Email: us.support@dewetron.com
Web: http://www.dewetron.com

The telephone hotline is available Monday to Friday between 08:00 and 17:00 GST (GMT -5:00)

Restricted Rights Legend: Use Austrian law for duplication or disclosure.

DEWETRON GmbH
Parkring 4
8074 Grambach
AUSTRIA

Please find further information at https://www.dewetron.com.

CHAPTER

TWO

DOCUMENT HISTORY

Table 2.1: Document History

Date	Author	SW Version	Change	Doc. Rev.
21.05.2026	MF	OXYGEN 8.1	Initial release	1.0
16.11.2026	MF	OXYGEN 8.3	Minor updates and corrections applied	1.1

Authors: M. Fuchs (MF)

CHAPTER

THREE

OPC UA INTRODUCTION

3.1 OPC UA Communication Interface

The Open Platform Communications Unified Architecture (OPC UA) is a standardized, platform-independent communication standard that enables data exchange between devices, machines and software applications. It was published by the OPC Foundation in 2008 with the first official version. OPC UA supports modern communication protocols, high security standards, platform independence, scalability and flexibility. Another key factor is manufacturer independence, which enables OPC UA to facilitate communication between different manufacturers. This is crucial for networked production and industrial automation, thus the industry 4.0 standard.

3.1.1 Overview

Modern industrial environments rely on the integration of heterogeneous systems, often comprising both legacy and contemporary control units, sensors, and machines. Bridging the gap between incompatible communication technologies is critical for effective monitoring and control. OPC (Open Platform Communications) provides a unified language that enables seamless data exchange between devices, automation systems, databases, and software solutions, regardless of the underlying hardware or protocol.

3.1.2 Historical Context and Evolution

- **OPC Classic (1996):** Originally developed to standardize multi-vendor device connectivity. While it established a secure framework, it was built on Windows COM/DCOM technology and was strictly limited to Windows-based environments.
- **OPC UA (2006):** To address platform dependency and modern security requirements, the OPC Foundation introduced Unified Architecture (UA). This evolution is platform-independent, offers enhanced reliability, and facilitates secure industrial communication across all operating systems.

3.1.3 DEWETRON Implementation

Starting with OXYGEN 8.1, DEWETRON high-accuracy measurement systems natively support the OPC UA standard. This allows for flexible network integration in two primary modes:

- **OPC UA Server:** Publishing high-accuracy measurement data to the network.
- **OPC UA Client:** Receiving data from external devices and third-party systems.

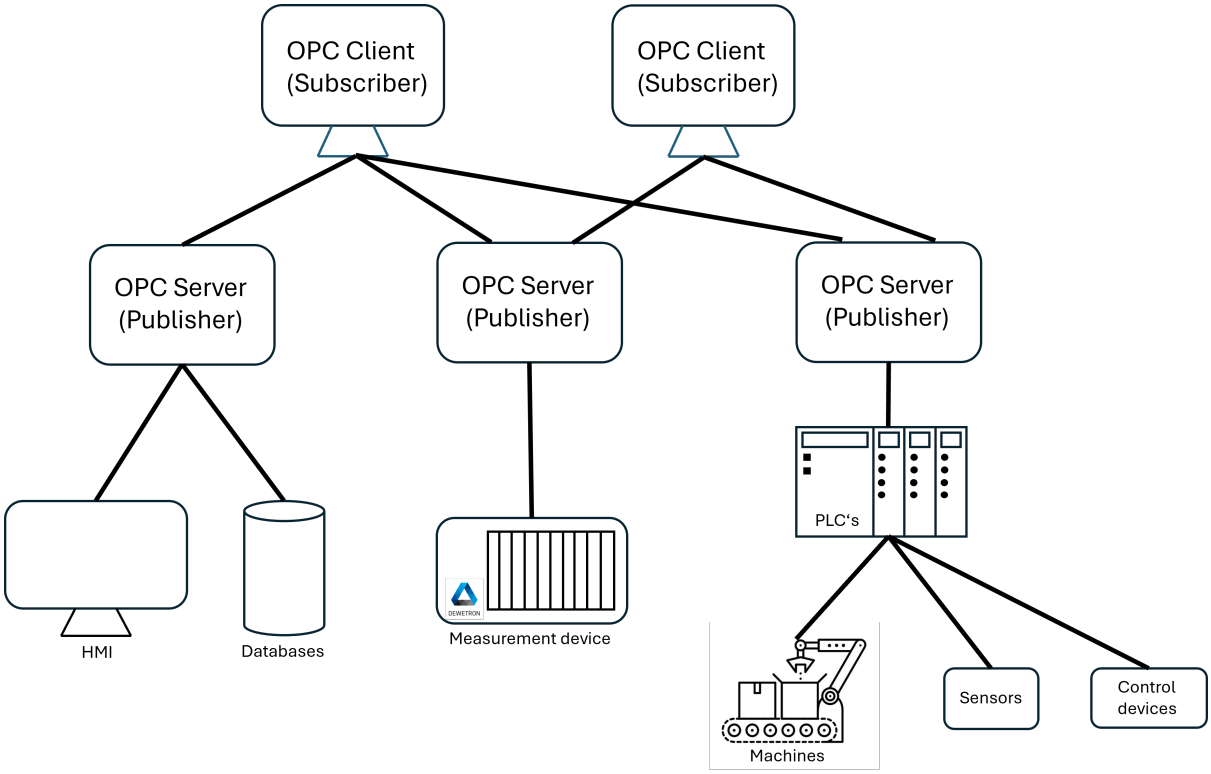


Fig. 3.1: OPC UA schematic

CHAPTER

FOUR

OPC UA SERVER

4.1 Configuring the OPC UA Server in OXYGEN

OXYGEN (v8.1 and higher) includes an integrated OPC UA Server for standardized data exchange with external OPC UA Clients.

Setup steps: (See Fig. 4.1)

- Navigate to Measurement Settings > Servers
- Click + to create a new OPC UA Server instance.
- Configure the OPC UA Server parameters (see below)
- Select the channels to be exposed to the OPC UA Clients.
- Click the Play button to start the server

Configuration Parameters:

- **Port:** Set the communication port (default: 4840). Adjust this to meet your IT or firewall requirements.
- **Update Rate (Global Limit):** Defines the maximum frequency for data processing to manage system load and bandwidth. This rate acts as a global cap and dictates the upper limit for both the Sampling and Publishing Intervals.
  - **Sampling Interval:** Frequency at which the server polls internal data.
  - **Publishing Interval:** Frequency at which the server sends data packets to the client.

Attention

Shorter intervals enable faster monitoring, while longer intervals minimize the measurement device's CPU load.

- **Statistic Mode:**

  Determines how data is processed for high-frequency channels within the publishing interval:
  - *Last Value:* Returns the most recent value (default).
  - *Minimum / Maximum:* Returns the lowest or highest value.
  - *Average:* Returns the arithmetic mean.
  - *RMS:* Returns the Root Mean Square value.

- **Channel Selection:** Enable or disable specific measurement channels. Only selected channels are visible in the OPC UA Server's address space.

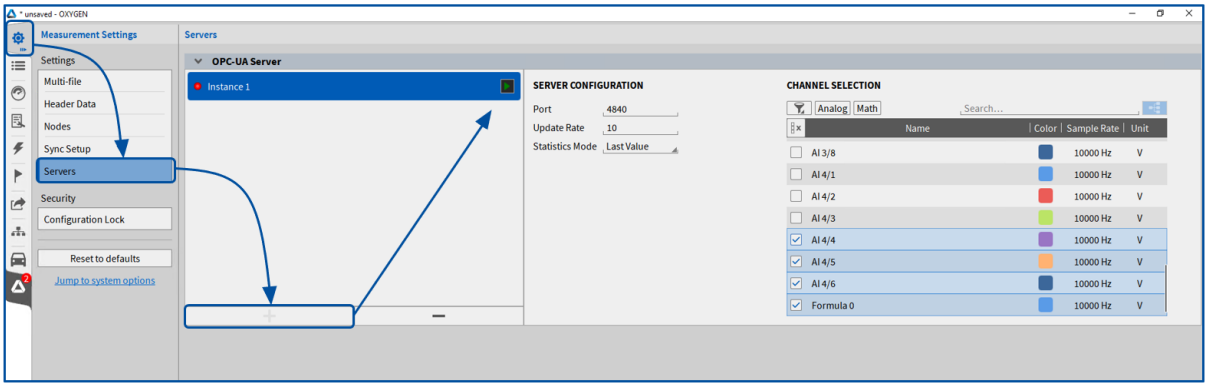


Fig. 4.1: Activate and set OPC UA Server in OXYGEN.

4.2 Channel Parameters

For every channel activated on the OXYGEN OPC UA Server, the following parameters are available to the OPC UA Client:

- *Samplerate:* The actual acquisition rate of the channel within OXYGEN (independent of the OPC UA publishing or sampling intervals).
- *Usable:* the channel status in OXYGEN: 1 (Active) or 0 (Inactive/Deactivated).
- *ScaleFactor:* The scaling factor applied to the channel in OXYGEN.
- *ScaleOffset:* The scaling offset applied to the channel in OXYGEN.
- *Measurement:* The actual measurement value of the channel.

CHAPTER

FIVE

OPC UA CLIENT

5.1 OPC UA Client Overview

The OXYGEN OPC UA Client enables data acquisition and monitoring by connecting to external OPC UA servers. It ensures reliable, industry-standard communication and supports a wide range of applications, from basic connectivity to high-speed data updates.

5.2 Setting up an OPC UA Client

To use the OPC UA Client in OXYGEN (v8.1 and higher), follow these steps:

- **Add Plugin:** Open the OXYGEN channel list, click the + button (See ① in Fig. 5.1), and select *OPC UA Client* from the Data Input section.
- **Select or Add Endpoint:** Choose a previously saved configuration from the Configured Endpoints list, or set up a new target server connection in the *Add Endpoint* section (See ④ in Fig. 5.1).
- **Configure Parameters:**
  - **Name:** Enter a unique, user-defined profile name for this specific connection (e.g., the name of the target machine or server location).
  - **URL:** Enter the full Endpoint URL of the target OPC UA Server (e.g., **opc.tcp://127.0.0.1:4840**).

    Note

    A single OPC UA Server can provide multiple endpoints with different security profiles or communication ports.

  - **Time Source:** Define how data is timestamped:
    - *Internal Time:* Uses the local OXYGEN time upon data reception.
    - *Reported Time:* Uses the OPC UA Server's timestamp. Ensure the OPC UA Server is synchronized and provides timestamps in UTC to avoid data inconsistency.
  - **Publishing Interval:** Set the cycle rate in [ms] for data subscriptions. The minimum/maximum limits are defined by the connected OPC UA Server.
- **Verify Connection:** Click *Check Status* (See ③ in Fig. 5.1). The system will display "Server reachable" (green) or "Server not reachable" (red).

- **Save & Add:** Click the *Store Config* button (see ② in Fig. 5.1) to save this connection profile to the *Configured Endpoints* list. Click *Add* (See ⑤ in Fig. 5.1) to finalize and load the channel configuration.

Tip

You can export the Configured Endpoints as a .csv file to replicate settings across different projects or systems.

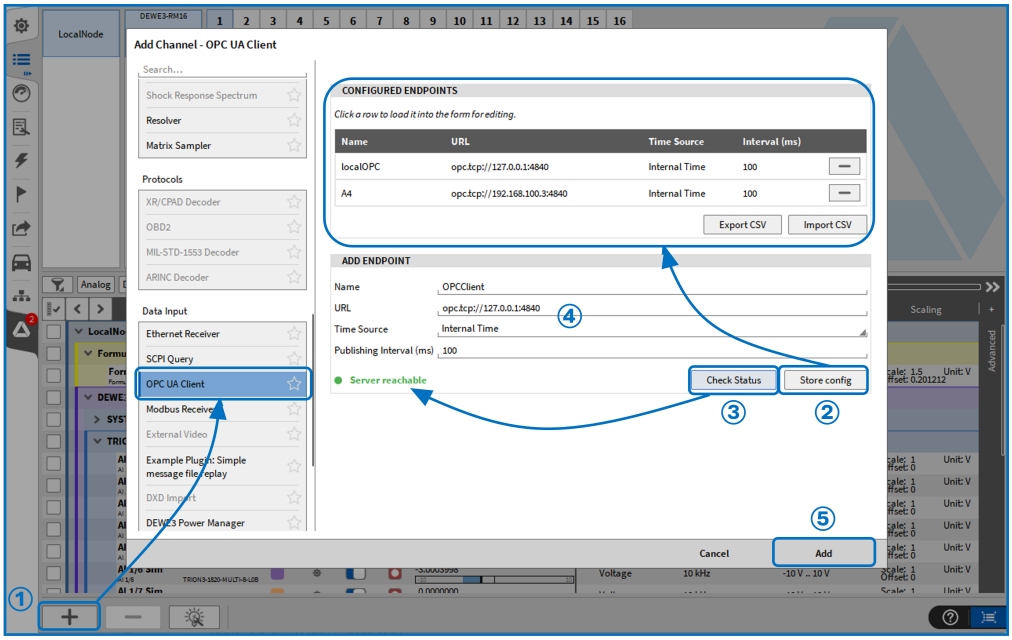


Fig. 5.1: Activate an OPC UA Client

Once added, the OPC UA Client will appear as a new section in the OXYGEN channel list (See Fig. 5.2).

5.2.1 Connection handling

To modify an existing OPC UA Client, click the gear icon (See ① in Fig. 5.2) in the OXYGEN channel list. You can update the Name, URL, Time Source, and Publishing Interval at any time.

Connection States:

The client provides real-time status feedback for troubleshooting:

- **Server not reachable:** Indicates a network-level issue (e.g., wrong IP or firewall).
- **Not connected:** The OPC UA Server is physically reachable, but the session could not be established.
- **Connected:** A session is fully established and active.

Automatic Reconnection:

- **Configuration Changes:** Modifying critical parameters (e.g., URL or security settings) automatically triggers a reconnection.
- **Connection Loss:** If the connection is interrupted, the client attempts to reconnect automatically.

- **Restoration:** Once reconnected, all subscriptions and monitored items are restored automatically.

Attention

For reliable restoration, the server should maintain stable namespace indices and node identifiers.

5.2.2 Security, Certificates and authentication

Security settings (See ③ in Fig. 5.2) follow the standard OPC UA protocol. The client automatically queries the server for available endpoints and security policies. If a previously selected policy is no longer supported by the server, it will be flagged in the interface.

Certificate Handling:

- **Automatic Generation:** Simplifies setup; recommended for most scenarios.
- **Manual Provisioning:** Used for environments with strict security or PKI requirements. Manual certificates must be explicitly enabled.

Authentication:

Two authentication modes are available (See ④ in Fig. 5.2):

- **Anonymous:** No credentials required.
- **Username/Password:** Required if the server restricts access.

Attention

For security reasons, passwords are not stored. They must be re-entered after restarting the application.

Data Acquisition:

If the server requires authentication, a successful login is mandatory before data acquisition can begin.

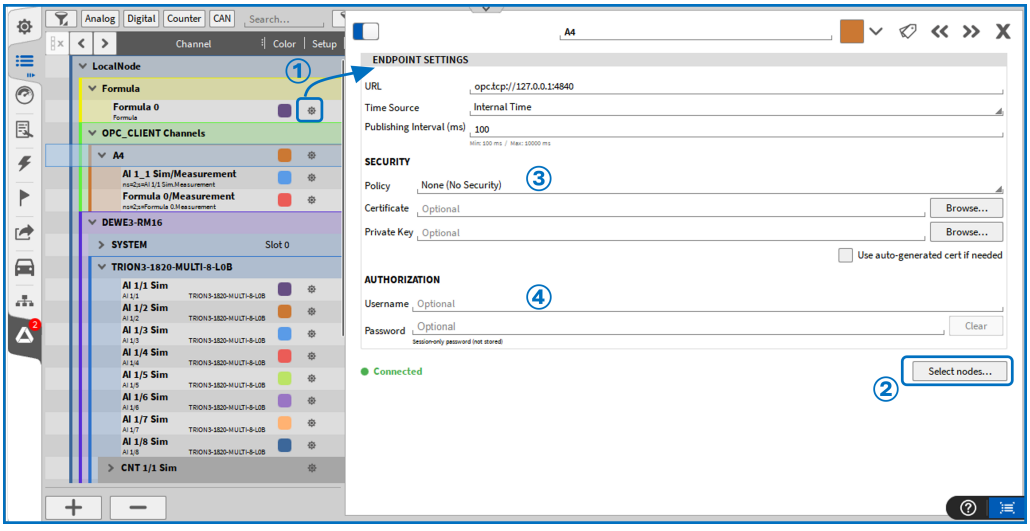


Fig. 5.2: OPC UA Client settings

5.2.3 Channel Setup

After configuring the server, click *Select Nodes* (See ② in Fig. 5.2) to choose the channels for subscription. Data acquisition is managed via the standard OPC UA subscription mechanism.

Node Selection: All available nodes are displayed in a tree view (See Fig. 5.3). The structure and available objects depend on the specific OPC UA Server implementation. For details on parameters provided by an OXYGEN OPC UA Server, refer to the *Channel Parameters* section in the Server chapter.

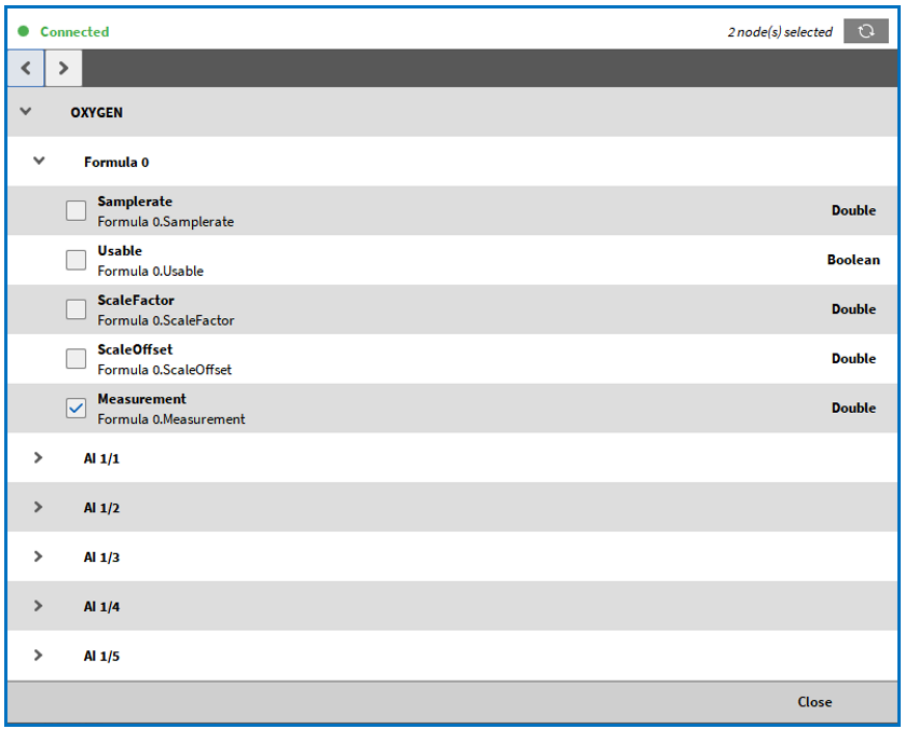


Fig. 5.3: Node selection for OPC UA Client

Channel Configuration:

Once selected, channels appear in the OXYGEN channel list under the respective OPC UA Client section. Click the gear icon (See ① in Fig. 5.4) of a channel to edit its properties:

- *Sampling Interval [ms]:* Defines how often the server samples the data (See ② in Fig. 5.4).
  - Example: A Sampling Interval of 20 ms combined with a Publishing Interval of 100 ms means the server sends a packet every 100 ms containing 5 sampled values for that channel.
- *Scaling:* Individual scaling can be applied to each channel in the scaling section (See ③ in Fig. 5.4).

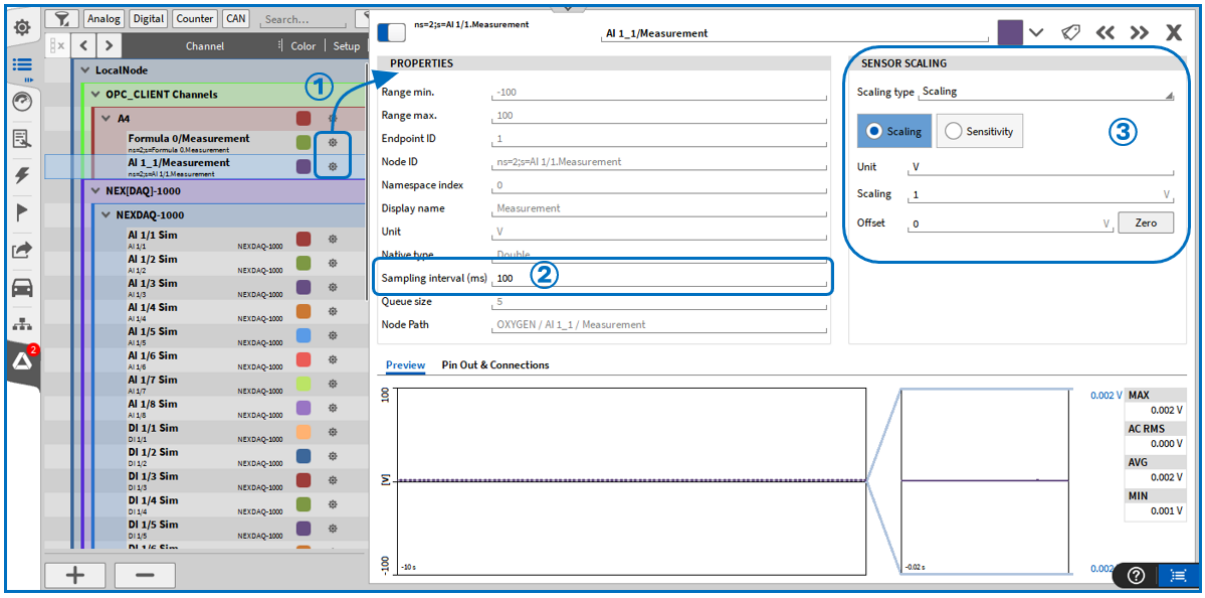


Fig. 5.4: OPC UA Client channel settings

5.2.4 Buffering and Update Behavior

The client utilizes buffering (queueing) for monitored items to prevent data loss during temporary communication delays or high-speed updates. Data is recorded whenever a value or its timestamp changes. This ensures consistent time-series recording even if the measurement value remains constant.

5.2.5 General Usage Notes

To ensure reliable operation, consider the following:

- **Server Implementation:** The client requires a standard-compliant OPC UA service implementation on the server side.
- **Stable Identifiers:** Use stable node identifiers to ensure consistent automatic reconnection.
- **Security:** Align security policies with your specific network deployment requirements.
- **Time Sync:** Verify OPC UA Server time synchronization when using “Reported Time” (OPC UA Server timestamps). When using OPC UA Client-side (“Internal Time”) timestamps, note that data may be delivered in batches (depending on OPC UA queue size and publishing interval). As a result, multiple values may share very similar or identical client timestamps, even if they were sampled at different times on the server. Ensure both server and client system clocks are correctly synchronized. It is recommended not to use manually adjusted or unsynchronized system time, as this may result in incorrect or misleading timestamps, especially when working with the reported time of the OPC UA Server.

5.2.6 Performance Considerations

The OXYGEN OPC UA Client is verified for high-performance scenarios, including millisecond-level sampling. However, actual performance is influenced by:

- Server processing capabilities and system load.
- Network bandwidth and latency.
- The ratio between sampling and publishing intervals.

Recommendation: Always balance sampling and publishing intervals to match your hardware and network infrastructure.