

Integration Objects'

**Seamless & Secure IT-OT-IIoT Integration
Platform**

Smart IoT Highway

Version 2.4.3

**APPLICATION CONFIGURATION
GUIDE**

Integration Objects' Smart IoT Highway Application Configuration Guide Version 2.4.3

Published April 2026.

Copyright © 2018 - 2026 Integration Objects. All rights reserved.

No part of this document may be reproduced, stored in a retrieval system, translated, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of Integration Objects.

Windows®, Windows NT® and .NET are registered trademarks of Microsoft Corporation.

SIOTH® is a registered trademark of Integration Objects.

TABLE OF CONTENTS

PREFACE	17
INTRODUCTION	19
SIOTH Features	20
1. Data Exchange with OT-IT Systems	20
1.1. Connectors	20
1.2. Data Flow	21
2. Cyber Security	22
2.1. Architecture Level	22
2.2. Communications and Data Flow Level	23
2.3. System Level	24
2.3.1. Authentication	24
2.3.2. Role-based Access	25
2.3.3. Abnormal System Behaviors Detection	25
2.3.4. High Availability	26
3. Data Model	26
4. Job Engine	27
5. Unified Control Interface	28
6. SIOTH® UHMI SCADA	28
6.1. UHMI SCADA Designer	29
6.2. UHMI SCADA Runtime	30
7. Scalability	30
8. Flexible User Management	30

APPLICATION CONFIGURATION	32
1. User Interface Overview	32
2. Nodes.....	33
2.1. Add New Node.....	34
2.2. Edit Node.....	37
2.3. Download Log Event Viewer	37
2.4. Delete Node	37
3. Devices	38
3.1. Add New Device	39
3.1.1. Allen Bradley	41
3.1.2. AMQP-091	43
3.1.3. Azure Event Hub.....	45
3.1.4. BACnet.....	47
3.1.5. Data Stores	48
3.1.5.1. InfluxDB	49
3.1.5.2. Kafka	50
3.1.5.3. MongoDB.....	52
3.1.5.4. MS Access.....	53
3.1.5.5. MySQL.....	54
3.1.5.6. ODBC	55
3.1.5.7. Oracle	57
3.1.5.8. PI.....	58
3.1.5.9. PI AF	59
3.1.5.10. PostgreSQL	60
3.1.5.11. SQL Server	61
3.1.6. DNP3.....	63
3.1.7. FTP Server	69

3.1.8.	HARTIP	71
3.1.9.	IEC 60870-5-104.....	73
3.1.10.	J1939.....	78
3.1.11.	Modbus.....	79
3.1.12.	MQTT	83
3.1.13.	OPC AE	88
3.1.14.	OPC DA	89
3.1.15.	OPC HDA.....	92
3.1.16.	OPC UA	94
3.1.17.	S7.....	97
3.1.18.	SMS Server.....	101
3.1.19.	SMTP Server	103
3.1.20.	SNMP.....	104
3.2.	Edit Device	107
3.3.	Delete Device.....	107
3.4.	Create Multiple Devices.....	108
4.	Projects.....	108
4.1.	Add New Project.....	109
4.2.	Open Project	111
4.3.	Edit Project.....	112
4.4.	Export Project	112
4.5.	Import Project.....	112
4.6.	Delete Project	112
4.7.	Reload Project	113
5.	Data Flows.....	113
5.1.	Manage Data Flows.....	114

5.1.1.	Add New Data Flow	114
5.1.2.	Edit Data Flow.....	115
5.1.3.	Duplicate Data Flow	115
5.1.4.	Start/Stop Data Flows.....	116
5.1.5.	Delete Data Flow	117
5.2.	Configure Data Flow	118
5.2.1.	Identification Step.....	121
5.2.2.	Offline Deployment.....	125
5.2.3.	Payload Transformation Step	127
5.2.4.	Tag Configuration Step.....	132
5.2.5.	Tag Mapping Step	135
5.2.6.	Protocols.....	135
5.2.6.1.	Allen Bradley.....	135
5.2.6.2.	AMQP-091	137
5.2.6.3.	BACnet	143
5.2.6.4.	DNP3.....	144
5.2.6.5.	FTP	145
5.2.6.6.	HARTIP	146
5.2.6.7.	IEC 60870-5-104.....	147
5.2.6.8.	J1939.....	148
5.2.6.9.	Modbus.....	149
5.2.6.10.	MQTT.....	152
5.2.6.11.	OPC AE.....	165
5.2.6.12.	OPC DA.....	169
5.2.6.13.	OPC HDA	172
5.2.6.14.	OPC UA.....	178
5.2.6.15.	REST	195
5.2.6.16.	S7	201

5.2.7.	Data Stores	203
5.2.7.1.	CSV File	203
5.2.7.2.	InfluxDB	208
5.2.7.3.	Kafka	210
5.2.7.4.	MongoDB.....	213
5.2.7.5.	MS Access.....	215
5.2.7.6.	MYSQL.....	222
5.2.7.7.	ODBC	229
5.2.7.8.	Oracle	235
5.2.7.9.	PI.....	241
5.2.7.10.	PI AF.....	248
5.2.7.11.	PostgreSQL	254
5.2.7.12.	SQL Server	260
5.2.8.	Brokers	268
5.2.8.1.	OPC UA Server	268
5.2.8.2.	MQTT Broker.....	287
5.2.8.3.	Azure Event Hub.....	291
5.2.9.	Network Watcher	295
5.2.9.1.	PING	295
5.2.9.2.	Health	298
5.2.9.3.	SNMP	300
5.2.9.4.	SNMP Trap.....	304
5.2.10.	Routers	306
5.2.10.1.	Bridge Server	306
5.2.10.2.	Bridge Client.....	310
5.2.10.3.	Router	315

TABLE OF FIGURES

FIGURE 1: SIOTH® PLATFORM OVERVIEW.....	19
FIGURE 2: SIOTH® HOME PAGE	32
FIGURE 3: APPLICATION CONFIGURATION HOME PAGE.....	33
FIGURE 4: NODES EXPLORER	34
FIGURE 5: ADD NEW NODE CONFIGURATION VIEW.....	35
FIGURE 6: DEVICES EXPLORER	38
FIGURE 7: ADD NEW DEVICE CONFIGURATION VIEW.....	39
FIGURE 8: ALLEN BRADLEY DEVICE CONFIGURATION VIEW	41
FIGURE 9: AMQP-091 DEVICE CONFIGURATION VIEW.....	43
FIGURE 10: AZURE EVENT HUB DEVICE CONFIGURATION VIEW	46
FIGURE 11: BACNET DEVICE CONFIGURATION VIEW	47
FIGURE 12: DATA STORES DEVICE CONFIGURATION VIEW	49
FIGURE 13: INFLUXDB DEVICE CONFIGURATION VIEW	49
FIGURE 14: KAFKA DEVICE CONFIGURATION VIEW	50
FIGURE 15: MONGODB DEVICE CONFIGURATION VIEW	52
FIGURE 16: MS ACCESS DEVICE CONFIGURATION VIEW	53
FIGURE 17: MYSQL DEVICE CONFIGURATION VIEW	54
FIGURE 18: ODBC DEVICE CONFIGURATION VIEW	55
FIGURE 19: ORACLE DEVICE CONFIGURATION VIEW	57
FIGURE 20: PI DEVICE CONFIGURATION VIEW.....	58
FIGURE 21: PI AF DEVICE CONFIGURATION VIEW	59
FIGURE 22: POSTGRES SQL DEVICE CONFIGURATION VIEW	60
FIGURE 23: SQL SERVER DEVICE CONFIGURATION VIEW	61

FIGURE 24: DNP3 DEVICE CONFIGURATION VIEW.....	63
FIGURE 25: FTP SERVER DEVICE CONFIGURATION VIEW	69
FIGURE 26: HARTIP DEVICE CONFIGURATION VIEW	71
FIGURE 27: IEC 60870-5-104 DEVICE CONFIGURATION VIEW	73
FIGURE 28: J1939 DEVICE CONFIGURATION VIEW.....	78
FIGURE 29: MODBUS DEVICE CONFIGURATION VIEW	79
FIGURE 30: MQTT DEVICE CONFIGURATION VIEW	83
FIGURE 31: OPC AE DEVICE CONFIGURATION VIEW	88
FIGURE 32: OPC DA DEVICE CONFIGURATION VIEW	89
FIGURE 33: OPC HDA DEVICE CONFIGURATION VIEW	92
FIGURE 34: OPC UA DEVICE CONFIGURATION VIEW	94
FIGURE 35: S7 DEVICE CONFIGURATION VIEW	97
FIGURE 36: SMS SERVER DEVICE CONFIGURATION VIEW - TWILIO	101
FIGURE 37: SMS SERVER DEVICE CONFIGURATION VIEW - SMSEAGLE.....	101
FIGURE 38: SMTP SERVER DEVICE CONFIGURATION VIEW.....	103
FIGURE 39: SNMP DEVICE CONFIGURATION VIEW.....	104
FIGURE 40: PROJECTS EXPLORER	109
FIGURE 41: ADD NEW PROJECT CONFIGURATION VIEW.....	110
FIGURE 42: DATA FLOWS EXPLORER	113
FIGURE 43: ADD NEW DATA FLOW CONFIGURATION VIEW.....	114
FIGURE 44: DATA FLOW DUPLICATION VIEW	116
FIGURE 45: START/STOP DATA FLOWS.....	117
FIGURE 46: DATA FLOW EDITOR.....	118
FIGURE 47: BLOCKS TOOLBOX	119
FIGURE 48: SHAPES TOOLBOX.....	119
FIGURE 49: CONFIGURE DATA FLOW - DRAG AND DROP CONNECTORS.....	120
FIGURE 50: CONFIGURE DATA FLOW - BLOCKS CONFIGURATION.....	120
FIGURE 51: CONNECTOR IDENTIFICATION CONFIGURATION VIEW.....	121

FIGURE 52: CONNECTOR IDENTIFICATION CONFIGURATION VIEW - OFFLINE OPTION	126
FIGURE 53: CONNECTOR IDENTIFICATION CONFIGURATION VIEW - DOWNLOAD CONFIGURATION	126
FIGURE 54: CONNECTOR TOPICS CONFIGURATION VIEW - VARIABLE CONFIGURATION.....	127
FIGURE 55: CONNECTOR IDENTIFICATION CONFIGURATION VIEW - PAYLOAD TRANSFORMATION CONFIGURATION	128
FIGURE 56: CONNECTOR PAYLOAD TRANSFORMATION CONFIGURATION VIEW.....	130
FIGURE 57: CONNECTOR TABLE CONFIGURATION VIEW.....	131
FIGURE 58: PAYLOAD TRANSFORMATION RESULT	131
FIGURE 59: CONNECTOR AS SOURCE - TAG CONFIGURATION VIEW.....	132
FIGURE 60: CONNECTOR AS DESTINATION - TAG CONFIGURATION VIEW.....	133
FIGURE 61: CONNECTOR AS DESTINATION - MAPPED VS. UNMAPPED SOURCE CONNECTOR	134
FIGURE 62: CONNECTOR - NEW SUBSCRIPTION CONFIGURATION VIEW	134
FIGURE 63: CONNECTOR AS DESTINATION - TAG MAPPING.....	135
FIGURE 64: ALLEN BRADLEY SOURCE/DESTINATION CONNECTOR - SUBSCRIPTION CONFIGURATION VIEW	136
FIGURE 65: AMQP-091 CONNECTOR AS SOURCE - CONFIGURATION VIEW.....	137
FIGURE 66: AMQP-091 CONNECTOR AS DESTINATION - CONFIGURATION VIEW.....	137
FIGURE 67: BACNET CONNECTOR - SUBSCRIPTION CONFIGURATION VIEW	143
FIGURE 68: DNP3 CONNECTOR AS SOURCE - SUBSCRIPTION CONFIGURATION VIEW.....	144
FIGURE 69: DNP3 CONNECTOR AS DESTINATION - TAG CONFIGURATION VIEW	145
FIGURE 70: FTP CONNECTOR - SUBSCRIPTION CONFIGURATION VIEW	145
FIGURE 71: HARTIP CONNECTOR - SUBSCRIPTION CONFIGURATION VIEW	146
FIGURE 72: IEC 60870-5-104 CONNECTOR - SUBSCRIPTION CONFIGURATION VIEW	147
FIGURE 73: J1939 CONNECTOR - SUBSCRIPTION CONFIGURATION VIEW.....	148
FIGURE 74: MODBUS CONNECTOR - SUBSCRIPTION CONFIGURATION VIEW.....	150
FIGURE 75: MQTT STANDARD CONNECTOR AS SOURCE - TOPICS CONFIGURATION VIEW	153
FIGURE 76: MQTT SPARKPLUG CONNECTOR AS SOURCE - TOPICS CONFIGURATION VIEW	158
FIGURE 77: MQTT SPARKPLUG CONNECTOR AS SOURCE - TAGS CONFIGURATION VIEW.....	158
FIGURE 78: MQTT STANDARD CONNECTOR AS DESTINATION - TOPICS CONFIGURATION VIEW.....	159

FIGURE 79: MQTT SPARKPLUG CONNECTOR AS DESTINATION - TOPICS CONFIGURATION VIEW	163
FIGURE 80: OPC AE CONNECTOR AS SOURCE - SUBSCRIPTION CONFIGURATION VIEW	166
FIGURE 81: OPC AE CONNECTOR AS SOURCE - SUBSCRIPTION CONFIGURATION VIEW	167
FIGURE 82: OPC DA CONNECTOR AS SOURCE / DESTINATION - SUBSCRIPTION CONFIGURATION VIEW.....	170
FIGURE 83: OPC HDA CONNECTOR AS SOURCE - SUBSCRIPTION CONFIGURATION VIEW	173
FIGURE 84: OPC HDA CONNECTOR AS DESTINATION - SUBSCRIPTION CONFIGURATION VIEW.....	177
FIGURE 85: OPC UA CONNECTOR - REDUNDANCY CONFIGURATION VIEW.....	179
FIGURE 86: OPC UA CONNECTOR AS SOURCE - TAG CONFIGURATION VIEW.....	181
FIGURE 87: OPC UA CONNECTOR AS SOURCE - SUBSCRIPTION CONFIGURATION VIEW FOR DA & AC.....	182
FIGURE 88: OPC UA CONNECTOR AS SOURCE - SUBSCRIPTION CONFIGURATION VIEW FOR HA	184
FIGURE 89: OPC UA CONNECTOR AS SOURCE - SUBSCRIPTION CONFIGURATION VIEW FOR HE.....	190
FIGURE 90: OPC UA CONNECTOR AS DESTINATION - TAG CONFIGURATION VIEW	193
FIGURE 91: REST CONNECTOR - INFORMATION AND SPECIFICATION VIEW	195
FIGURE 92: REST CONNECTOR AS SOURCE - FIELDS CONFIGURATION VIEW	200
FIGURE 93: REST CONNECTOR AS DESTINATION - FIELD MAPPING CONFIGURATION VIEW.....	201
FIGURE 94: S7 CONNECTOR AS SOURCE / DESTINATION - SUBSCRIPTION CONFIGURATION VIEW.....	202
FIGURE 95: CSV CONNECTOR AS SOURCE – CSV CONFIGURATION VIEW	204
FIGURE 96: CSV CONNECTOR AS DESTINATION - CONFIGURATION VIEW	206
FIGURE 97: INFLUXDB CONNECTOR AS DESTINATION - MEASUREMENT CONFIGURATION VIEW	209
FIGURE 98: KAFKA CONNECTOR AS SOURCE - TAG CONFIGURATION VIEW.....	211
FIGURE 99: KAFKA CONNECTOR AS DESTINATION - TAG CONFIGURATION VIEW	212
FIGURE 100: MONGODB CONNECTOR AS DESTINATION - CREATE NEW COLLECTION VIEW	214
FIGURE 101: MONGODB CONNECTOR AS DESTINATION – COLLECTION CONFIGURATION VIEW	214
FIGURE 102: MONGODB CONNECTOR AS DESTINATION - READ SETTINGS CONFIGURATION VIEW.....	215
FIGURE 103: MS ACCESS CONNECTOR AS SOURCE – TABLE CONFIGURATION VIEW	216
FIGURE 104: MS ACCESS CONNECTOR AS DESTINATION – CREATE NEW TABLE CONFIGURATION VIEW.....	221
FIGURE 105: MS ACCESS CONNECTOR AS DESTINATION – TABLE CONFIGURATION AND FIELD MAPPING CONFIGURATION VIEW	222

FIGURE 106: MYSQL CONNECTOR AS SOURCE - TABLE CONFIGURATION VIEW.....	223
FIGURE 107: MYSQL CONNECTOR AS DESTINATION - CREATE NEW TABLE CONFIGURATION VIEW.....	228
FIGURE 108: MS ACCESS CONNECTOR AS DESTINATION – TABLE CONFIGURATION AND FIELD MAPPING CONFIGURATION VIEW	228
FIGURE 109: ODBC CONNECTOR AS SOURCE - TABLE CONFIGURATION VIEW.....	229
FIGURE 110: ODBC CONNECTOR AS DESTINATION - CREATE NEW TABLE CONFIGURATION VIEW.....	234
FIGURE 111: ODBC CONNECTOR AS DESTINATION - TABLE CONFIGURATION AND FIELD MAPPING CONFIGURATION VIEW	234
FIGURE 112: ORACLE CONNECTOR AS SOURCE - TABLE CONFIGURATION VIEW.....	235
FIGURE 113: MS ACCESS CONNECTOR AS DESTINATION – CREATE NEW TABLE CONFIGURATION VIEW.....	240
FIGURE 114: ORACLE CONNECTOR AS DESTINATION – TABLE CONFIGURATION AND FIELD MAPPING CONFIGURATION VIEW	241
FIGURE 115: PI CONNECTOR AS SOURCE - TAG CONFIGURATION VIEW	242
FIGURE 116: PI CONNECTOR AS SOURCE - CURRENT VALUES READ MODE - SUBSCRIPTION CONFIGURATION VIEW	243
FIGURE 117: PI CONNECTOR AS SOURCE - ARCHIVED VALUES READ MODE - SUBSCRIPTION CONFIGURATION VIEW	244
FIGURE 118: PI CONNECTOR AS DESTINATION - SUBSCRIPTION CONFIGURATION VIEW.....	246
FIGURE 119: PI CONNECTOR AS DESTINATION - TAG CONFIGURATION VIEW.....	247
FIGURE 120: PI CONNECTOR AS DESTINATION - PAYLOAD TRANSFORMATION	248
FIGURE 121: PI AF CONNECTOR AS SOURCE - TAG CONFIGURATION VIEW	249
FIGURE 122: PI AF CONNECTOR AS SOURCE - CURRENT VALUES READ MODE - SUBSCRIPTION CONFIGURATION VIEW	250
FIGURE 123: PI AF CONNECTOR AS SOURCE - ARCHIVED VALUES READ MODE - SUBSCRIPTION CONFIGURATION VIEW	250
FIGURE 124: PI AF CONNECTOR AS DESTINATION - SUBSCRIPTION CONFIGURATION VIEW	252
FIGURE 125: PI AF CONNECTOR AS DESTINATION - TAG CONFIGURATION VIEW	253
FIGURE 126: POSTGRESQL CONNECTOR AS SOURCE - TABLE CONFIGURATION VIEW.....	254

FIGURE 127: POSTGRES SQL CONNECTOR AS DESTINATION – CREATE NEW TABLE CONFIGURATION VIEW.....	259
FIGURE 128: POSTGRES SQL CONNECTOR AS DESTINATION – TABLE CONFIGURATION AND FIELD MAPPING CONFIGURATION VIEW	260
FIGURE 129: SQL SERVER CONNECTOR AS SOURCE - TABLE CONFIGURATION VIEW.....	261
FIGURE 130: SQL SERVER CONNECTOR AS DESTINATION – CREATE NEW TABLE CONFIGURATION VIEW.....	266
FIGURE 131: SQL SERVER CONNECTOR AS DESTINATION – TABLE CONFIGURATION AND FIELD MAPPING CONFIGURATION VIEW	267
FIGURE 132: SQL SERVER CONNECTOR AS DESTINATION - PAYLOAD TRANSFORMATION.....	268
FIGURE 133: OPC UA SERVER BROKER - UA SERVER CONFIGURATION VIEW	269
FIGURE 134: OPC UA SERVER BROKER - TAGS CONFIGURATION VIEW	277
FIGURE 135: OPC UA SERVER BROKER - TAGS CSV TEMPLATE	277
FIGURE 136: OPC UA SERVER BROKER - FIELDS MAPPING VIEW.....	282
FIGURE 137: OPC UA SERVER BROKER - HISTORIAN CONFIGURATION VIEW	283
FIGURE 138: MQTT BROKER CONFIGURATION VIEW	287
FIGURE 139: AZURE EVENT HUB CONNECTOR AS DESTINATION – CONFIGURATION VIEW	292
FIGURE 140: PING CONNECTOR AS SOURCE - TAG CONFIGURATION VIEW	296
FIGURE 141: PING CONNECTOR AS SOURCE - NEW TAG CONFIGURATION VIEW	297
FIGURE 142: HEALTH CONNECTOR AS SOURCE - KPIs CONFIGURATION VIEW	299
FIGURE 143: HEALTH CONNECTOR AS SOURCE - CUSTOM KPIs CONFIGURATION VIEW	300
FIGURE 144: SNMP CONNECTOR AS SOURCE - SUBSCRIPTION CONFIGURATION VIEW	301
FIGURE 145: SNMP CONNECTOR AS DESTINATION - TAG CONFIGURATION VIEW.....	303
FIGURE 146: SNMP TRAP CONNECTOR AS SOURCE - TAGS CONFIGURATION VIEW	305
FIGURE 147: BRIDGE SERVER CONNECTOR CONFIGURATION VIEW	307
FIGURE 148: BRIDGE CLIENT CONNECTOR IDENTIFICATION CONFIGURATION VIEW	311
FIGURE 149: BRIDGE CLIENT CONNECTOR PORT MAPPING CONFIGURATION VIEW	314
FIGURE 150: ROUTER CONNECTOR IDENTIFICATION CONFIGURATION VIEW	315
FIGURE 151: ROUTER CONNECTOR PORT MAPPING CONFIGURATION VIEW	318

LIST OF TABLES

TABLE 1: NODE CONFIGURATION PARAMETERS	36
TABLE 2: SUPPORTED DEVICES	40
TABLE 3: ALLEN BRADLEY DEVICE CONFIGURATION PARAMETERS	43
TABLE 4: AMQP-091 DEVICE CONFIGURATION PARAMETERS	45
TABLE 5: AZURE EVENT HUB DEVICE CONFIGURATION PARAMETERS	47
TABLE 6: BACNET DEVICE CONFIGURATION PARAMETERS	48
TABLE 7: INFLUXDB DEVICE CONFIGURATION PARAMETERS	50
TABLE 8: KAFKA DEVICE CONFIGURATION PARAMETERS	52
TABLE 9: MONGODB DEVICE CONFIGURATION PARAMETERS	53
TABLE 10: MS ACCESS DEVICE CONFIGURATION PARAMETERS	54
TABLE 11: MYSQL DEVICE CONFIGURATION PARAMETERS	55
TABLE 12: ODBC DEVICE CONFIGURATION PARAMETERS	56
TABLE 13: ORACLE DEVICE CONFIGURATION PARAMETERS	57
TABLE 14: PI DEVICE CONFIGURATION PARAMETERS	58
TABLE 15: PI AF DEVICE CONFIGURATION PARAMETERS	59
TABLE 16: POSTGRESQL DEVICE CONFIGURATION PARAMETERS	61
TABLE 17: SQL SERVER DEVICE CONFIGURATION PARAMETERS	62
TABLE 18: DNP3 DEVICE CONFIGURATION PARAMETERS	68
TABLE 19: FTP DEVICE CONFIGURATION PARAMETERS	71
TABLE 20: HARTIP DEVICE CONFIGURATION PARAMETERS	73
TABLE 21: IEC 60870-5-104 DEVICE CONFIGURATION PARAMETERS	77
TABLE 22: J1939 DEVICE CONFIGURATION PARAMETERS	78
TABLE 23: MODBUS DEVICE CONFIGURATION PARAMETERS	82

TABLE 24: MQTT DEVICE CONFIGURATION PARAMETERS	87
TABLE 25: OPC AE DEVICE CONFIGURATION PARAMETERS	89
TABLE 26: OPC DA DEVICE CONFIGURATION PARAMETERS	92
TABLE 27: OPC HDA DEVICE CONFIGURATION PARAMETERS	93
TABLE 28: OPC UA DEVICE CONFIGURATION PARAMETERS.....	97
TABLE 29: S7 DEVICE CONFIGURATION PARAMETERS.....	100
TABLE 30: S7 RACK AND SLOT PARAMETERS	100
TABLE 31: SMS SERVER DEVICE CONFIGURATION PARAMETERS	102
TABLE 32: SMTP SERVER DEVICE CONFIGURATION PARAMETERS.....	104
TABLE 33: SNMP DEVICE CONFIGURATION PARAMETERS	107
TABLE 34: PROJECT CONFIGURATION PARAMETERS.....	110
TABLE 35: PROJECT HOMEPAGE VIEW	111
TABLE 36: ADD NEW DATA FLOW CONFIGURATION PARAMETERS.....	115
TABLE 37: CONNECTOR IDENTIFICATION CONFIGURATION PARAMETERS	125
TABLE 38: CONNECTOR - NEW SUBSCRIPTION CONFIGURATION PARAMETERS.....	134
TABLE 39: ALLEN BRADLEY SOURCE/DESTINATION CONNECTOR - SUBSCRIPTION CONFIGURATION PARAMETERS	136
TABLE 40: AMQP-091 CONNECTOR - CONFIGURATION PARAMETERS	142
TABLE 41: BACNET CONNECTOR - SUBSCRIPTION CONFIGURATION PARAMETERS	143
TABLE 42: DNP3 CONNECTOR AS SOURCE - SUBSCRIPTION CONFIGURATION PARAMETERS.....	144
TABLE 43: FTP CONNECTOR - SUBSCRIPTION CONFIGURATION PARAMETERS	146
TABLE 44: HARTIP CONNECTOR - SUBSCRIPTION CONFIGURATION PARAMETERS	147
TABLE 45: IEC 60870-5-104 CONNECTOR - SUBSCRIPTION CONFIGURATION PARAMETERS	148
TABLE 46: J1939 CONNECTOR - SUBSCRIPTION CONFIGURATION PARAMETERS.....	149
TABLE 47: MODBUS CONNECTOR ADDITIONAL CONFIGURATION PARAMETERS	149
TABLE 48: MODBUS CONNECTOR - SUBSCRIPTION CONFIGURATION PARAMETERS	151
TABLE 49: MODBUS CONNECTOR - SUBSCRIPTION TEMPLATE FIELDS	152
TABLE 50: MQTT STANDARD CONNECTOR AS SOURCE - TOPICS CONFIGURATION PARAMETERS	157

TABLE 51: MQTT STANDARD CONNECTOR AS DESTINATION - TOPICS CONFIGURATION PARAMETERS.....	162
TABLE 52: MQTT SPARKPLUG CONNECTOR AS DESTINATION - TOPICS CONFIGURATION PARAMETERS.....	165
TABLE 53: OPC AE CONNECTOR AS SOURCE - SUBSCRIPTION CONFIGURATION PARAMETERS	169
TABLE 54: OPC DA CONNECTOR - ADDITIONAL CONFIGURATION PARAMETERS.....	170
TABLE 55: OPC DA CONNECTOR - SUBSCRIPTION CONFIGURATION PARAMETERS	172
TABLE 56: OPC HDA CONNECTOR AS SOURCE - SUBSCRIPTION CONFIGURATION PARAMETERS	177
TABLE 57: OPC HDA CONNECTOR AS DESTINATION - SUBSCRIPTION CONFIGURATION PARAMETERS.....	178
TABLE 58: OPC UA CONNECTOR - ADDITIONAL CONFIGURATION PARAMETERS.....	180
TABLE 59: OPC UA CONNECTOR AS SOURCE - TAG CONFIGURATION PARAMETERS.....	182
TABLE 60: OPC UA CONNECTOR AS SOURCE - SUBSCRIPTION CONFIGURATION PARAMETERS FOR DA & AC.	183
TABLE 61: OPC UA CONNECTOR AS SOURCE - SUBSCRIPTION CONFIGURATION PARAMETERS FOR HA	186
TABLE 62: OPC UA HISTORICAL AGGREGATE FUNCTIONS TABLE.....	189
TABLE 63: OPC UA CONNECTOR AS SOURCE - SUBSCRIPTION CONFIGURATION PARAMETERS FOR HE	192
TABLE 64: OPC UA CONNECTOR AS DESTINATION - TAG CONFIGURATION PARAMETERS	194
TABLE 65: REST CONNECTOR - INFORMATION AND SPECIFICATION PARAMETERS	200
TABLE 66: S7 CONNECTOR - SUBSCRIPTION CONFIGURATION PARAMETERS.....	203
TABLE 67: CSV CONNECTOR AS SOURCE - CONFIGURATION PARAMETERS.....	206
TABLE 68: CSV CONNECTOR AS DESTINATION - CONFIGURATION PARAMETERS	208
TABLE 69: INFLUXDB CONNECTOR AS DESTINATION - MEASUREMENT CONFIGURATION PARAMETERS.....	210
TABLE 70: KAFKA CONNECTOR AS SOURCE - TAG CONFIGURATION PARAMETERS	212
TABLE 71: KAFKA CONNECTOR AS DESTINATION - TAG CONFIGURATION PARAMETERS.....	213
TABLE 72: MONGODB CONNECTOR AS DESTINATION - READ SETTINGS CONFIGURATION PARAMETERS	215
TABLE 73: MS ACCESS CONNECTOR AS SOURCE – TABLE CONFIGURATION PARAMETERS.....	216
TABLE 74: MYSQL CONNECTOR AS SOURCE - TABLE CONFIGURATION PARAMETERS	227
TABLE 75: ODBC CONNECTOR AS SOURCE – TABLE CONFIGURATION PARAMETERS.....	229
TABLE 76: ORACLE CONNECTOR AS SOURCE – TABLE CONFIGURATION PARAMETERS.....	236
TABLE 77: PI CONNECTOR ADDITIONAL CONFIGURATION PARAMETERS.....	242
TABLE 78: PI CONNECTOR AS SOURCE - SUBSCRIPTION CONFIGURATION PARAMETERS.....	245

TABLE 79: PI CONNECTOR AS DESTINATION - SUBSCRIPTION CONFIGURATION PARAMETERS	246
TABLE 80: PI AF CONNECTOR AS SOURCE - SUBSCRIPTION CONFIGURATION PARAMETERS.....	252
TABLE 81: PI AF CONNECTOR AS DESTINATION - SUBSCRIPTION CONFIGURATION PARAMETERS	253
TABLE 82: POSTGRESQL CONNECTOR AS SOURCE - TABLE CONFIGURATION PARAMETERS	255
TABLE 83: SQL SERVER ADDITIONAL CONFIGURATION PARAMETERS	261
TABLE 84: SQL SERVER CONNECTOR AS SOURCE - TABLE CONFIGURATION PARAMETERS	262
TABLE 85: OPC UA SERVER BROKER - UA SERVER CONFIGURATION PARAMETERS	276
TABLE 86: OPC UA SERVER BROKER - TAGS TEMPLATE PARAMETERS.....	280
TABLE 87: OPC UA SERVER BROKER - TAGS PARAMETERS.....	281
TABLE 88: OPC UA SERVER BROKER - HISTORIAN CONFIGURATION PARAMETERS.....	286
TABLE 89: MQTT BROKER CONFIGURATION PARAMETERS.....	291
TABLE 90: AZURE EVENT HUB CONNECTOR AS DESTINATION – CONFIGURATION PARAMETERS.....	295
TABLE 91: PING CONNECTOR AS SOURCE - NEW TAG CONFIGURATION PARAMETERS.....	298
TABLE 92: HEALTH CONNECTOR AS SOURCE - KPIS CONFIGURATION PARAMETERS.....	300
TABLE 93: SNMP CONNECTOR AS SOURCE - SUBSCRIPTION CONFIGURATION PARAMETERS	302
TABLE 94: SNMP CONNECTOR AS DESTINATION - TAG CONFIGURATION PARAMETERS.....	304
TABLE 95: SNMP TRAP CONNECTOR AS SOURCE - TAGS CONFIGURATION PARAMETERS.....	305
TABLE 96: SNMP TRAP CONNECTOR AS SOURCE - ADD MULTIPLE DEVICES CONFIGURATION VIEW	306
TABLE 97: BRIDGE SERVER CONNECTOR CONFIGURATION PARAMETERS.....	310
TABLE 98: BRIDGE CLIENT CONNECTOR IDENTIFICATION CONFIGURATION PARAMETERS.....	314
TABLE 99: BRIDGE CLIENT CONNECTOR PORT MAPPING CONFIGURATION PARAMETERS.....	314
TABLE 100: ROUTER CONNECTOR IDENTIFICATION CONFIGURATION PARAMETERS	317
TABLE 101: ROUTER CONNECTOR PORT MAPPING CONFIGURATION PARAMETERS	318

PREFACE

About This User Guide

This guide:

- Present Integration Objects' Smart IoT Highway.
- Describe the functions provided by Integration Objects' Smart IoT Highway.
- Explain each step of the configuration process.

Target Audience

This document is intended for users, application engineers, and IT/OT integrators who are responsible for configuring Integration Objects' Smart IoT Highway, with a particular focus on devices, connectors, and data flows.

It is assumed that users have basic knowledge of the communication protocols referenced in this user guide, including Allen-Bradley, AMQP, BACnet, DNP, HART-IP, IEC, Modbus, MQTT, OPC (Classic), OPC UA, S7, and SNMP.

Familiarity with common data storage technologies such as MS SQL Server, Oracle, MS Access, MySQL, PostgreSQL, CSV files, and ODBC is also expected.

Document Conventions

Convention	Description
Bold	Bolded text indicates user interface elements, such as buttons, menu items, and dialog names.
(!) Note	Information to be noted

Customer Support Services

Phone	Email
Americas: +1 713 609 9208 Europe-Africa-Middle East +216 71 195 360	Support: customerservice@integrationobjects.com Sales: sales@integrationobjects.com Online: www.integrationobjects.com

INTRODUCTION

Smart IoT Highway (SIOTH®) is an advanced IT-OT integration platform designed to facilitate secure data exchange and transformation. It establishes secure end-to-end pipelines to collect and store data from edge IoT devices and various other sources. SIOTH® enables organizations of all sizes to easily connect applications, systems, and services in a managed, scalable, and secure environment. This comprehensive integration solution allows for seamless connectivity between IT and OT, enabling the conversion of industrial data into actionable intelligence and valuable insights.

The SIOTH® platform operates on robust functional architecture, as illustrated in the figure below:

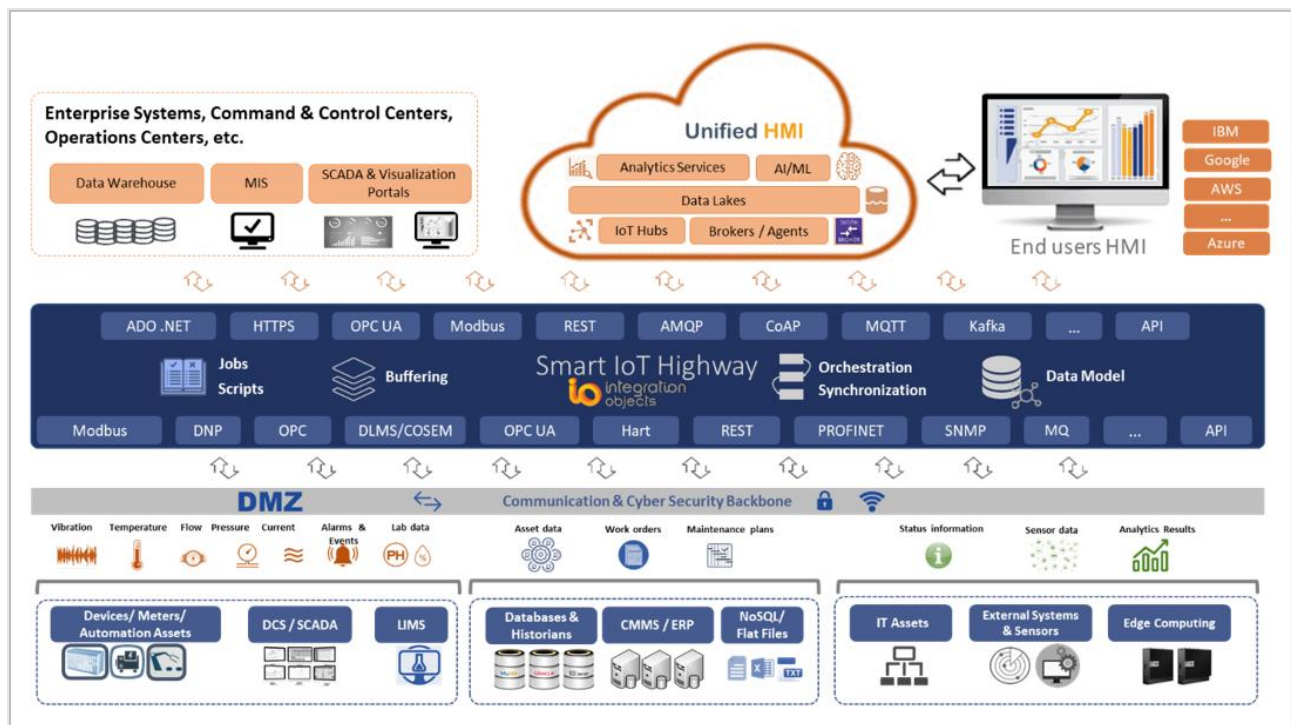


Figure 1: SIOTH® Platform Overview

SIOTH Features

1. Data Exchange with OT-IT Systems

SIOTH enables seamless connectivity and reliable data exchange across industrial (OT) and enterprise (IT) environments. The platform supports a wide range of data sources, protocols, and systems, providing a unified integration layer with the following capabilities:

- **Industrial Communication Protocols**

Supports protocols such as Modbus TCP, Modbus Serial, OPC Classic (DA, HDA, A&E), OPC UA, DNP3, HART and S7.

- **IoT Protocols**

Support protocols including MQTT, REST, AMQP, SNMP, and FTP.

- **Database Integration**

Connect with SQL-based databases such as Microsoft SQL Server, Oracle, MySQL and PostgreSQL.

- **Historians and Other Systems**

Integrate with data historians such as PI and InfluxDB, as well as streaming and messaging platforms such as Apache Kafka.

In addition, SIOTH supports extended connectivity through APIs and plug-ins, enabling flexible integration within diverse enterprise environments.

1.1. Connectors

SIOTH connectors are responsible for retrieving, transferring and delivering data. There are two types of connectors:

- **Source Connectors**

Communicate with data sources such as field devices, automation systems, or applications to collect data and push it into the SIOTH platform.

- **Destination Connectors**

Retrieve data from the SIOTH platform and deliver it to destination systems.

SIOTH connectors provide the following key features:

- **Automatic Reconnection**

Connectors continuously monitor communication to their respective systems. In the event of a connection loss, they automatically initiate a reconnection process without requiring user intervention.

- **Store and Forward**

When devices experience connectivity or network interruptions, connectors temporarily store collected data from source devices locally. Once connectivity is restored, the buffered data is automatically forwarded to the destination device or system.

1.2. Data Flow

SIOTH Data Flow defines logical links between source and destination connectors, representing end-to-end data exchange paths between sources and destinations.

This approach eliminates the need to deploy and interconnect multiple standalone applications across disparate systems and try to figure out how to get them to connect. SIOTH provides a unified mechanism that guarantees effortless connectors deployment - whether locally behind a firewall, or in the cloud - while ensuring reliable data collection and delivery.

Once data is received by SIOTH®, it can be routed simultaneously to multiple processing pipelines and destination nodes. This approach streamlines integration, simplifies architecture, and ensures scalable and resilient data exchange.

2. Cyber Security

SIOTH is secure by design, developed in compliance with **ISA/IEC 62443** standard. It provides robust cybersecurity features across multiple layers while addressing the core principles of **CIA Triad**:

- **Confidentiality**

Ensures that sensitive information is accessible only to authorized users and systems through authentication, authorization, and encryption mechanisms.

- **Integrity**

Protects data from being altered or tampered during transmission or storage using validation, digital signatures, and secure communication protocols.

- **Availability**

Guarantees that data and services remain consistently accessible to authorized users, even in the presence of failures or cyberattacks, through redundancy, failover, and recovery mechanisms.

2.1. Architecture Level

- **Support of DMZ architecture (Single or multiple DMZs)**

The primary SIOTH security feature lies in its architecture design. **SIOTH** provides several deployment options that allow users to design their system integration architecture based on a **defense-in-depth** approach while supporting network segmentation constraints.

By supporting DMZ architectures, SIOTH® enables users to:

- Deploy SIOTH® platform within a DMZ layer.
- Eliminate direct communication between the enterprise and the control network layers.
- Control and restrict flows and requests directed toward the control network layer.

This approach removes security bad practices while ensuring **data availability** for business applications and in a secure manner.

- **Firewall-friendly Configuration:**

All SIOTH communication ports are configurable, making the platform fully adaptable to existing firewall and network security policies.

2.2. Communications and Data Flow Level

- **HTTP and HTTPS Communication Protocols:**

By default, SIOTH platform uses the **HTTP protocol** for communication.

To enhance security and protect data transmission, users can switch the platform communication protocol to HTTPS, which leverages **SSL/TLS** data encryption.

This ensures secure communication between clients and servers, preventing unauthorized access or data interception.

- **Data Confidentiality and Integrity:**

By enabling the data encryption feature, SIOTH connectors encrypt and authenticate each packet. The encrypted packet can only be generated and interpreted by the connectors, ensuring:

- **Confidentiality:** protection against eavesdropping or espionage.
- **Integrity:** protection against data corruption or forgery.

- **User Authentication:**

SIOTH connectors can operate using specific user accounts configured with the least privilege principle, minimizing the risk of unauthorized access.

- **Protocol Break:**

SIOTH allows the use of different protocols for source and destination communications. This approach ensures that if a vulnerability is discovered in one protocol, it does not propagate through the entire data flow, strengthening resilience.

- **Support of Security Features:**

SIOTH supports advanced security modes, security policies, and authentication mechanisms specific to certain protocols, such as OPC UA, ensuring compliance with industry-standard security practices.

- **Ports Unification:**

All SIOTH connectors within the same platform can communicate seamlessly using a unified port. This provides significant advantages when deploying multiple connectors across diverse networks, as it:

- Eliminates the need to manage multiple ports.
- Simplifies firewall configuration.
- Streamlines communication while reducing complexity.

2.3. System Level

2.3.1. Authentication

- **User Authentication:**

Access to SIOTH platform is protected by **username** and **password** credentials.

(!) Note

- **Default username:** Administrator
- **Default Password:** Pa\$\$word123

- **Multi-Factor Authentication (MFA):**

SIOTH supports multi-factor authentication, requiring users to verify their identity using two or more factors before access is granted.

This significantly strengthens security by reducing the risk of compromised credentials.

MFA can be enabled both for new accounts and existing users seeking to enhance their account security.

- **User authorization via the OAuth 2.0:**

The platform supports OAuth 2.0 protocol for secure and standardized user authorization, ensuring safe access delegation and compliance with modern authentication practices.

2.3.2.Role-based Access

- **Role-based Dashboards:**

SIOTH provides dashboards that are tailored according to user roles, ensuring that each user has access only to information and tools relevant to their responsibilities.

- **Configurable User Profiles and Access Rights:**

Administrators can configure user profiles, privileges, and access rights through a dedicated admin environment. This allows fine-grained control over what each user can see and do in the system.

- **Traceability of Operations and User Actions:**

All user activities and system operations are continuously logged and traceable, ensuring complete visibility into every action performed on the platform. This guarantees effective auditing, accountability, and compliance with cybersecurity standards, as all user interactions through the interface are fully tracked and recorded.

2.3.3.Abnormal System Behaviors Detection

SIOTH connectors continuously monitor and report systems health status and metrics to the platform, including:

- CPU and memory consumption.
- Data flow and communications status.

- General health check indicators.

These metrics are evaluated against configurable rules within SIOTH® Job Engine.

When anomalies are evaluated, such as communication interruptions, suspected intrusion, or performance degradation, the system automatically generates alerts and notifications for users and administrators.

2.3.4.High Availability

SIOTH ensures business continuity through:

- Cluster-based deployments.
- Redundancy options such as **active-active** and **active-passive** configurations.

This architecture provides resilience, fault tolerance, and continuous availability of critical services.

3. Data Model

SIOTH® Data Model provides a user-friendly, meaningful, unified, and hierarchical organization of data. It enables users to organize and interact with data collected from various sources in a structured and consistent way.

- **Hierarchical Organization:**

Data elements are organized in a hierarchical model, making it easier to navigate, interpret, and manage relationships among data.

- **Object-Oriented Approach:**

The models are object-oriented, built upon data element definitions called classes.

- **Classes:** define attributes that describe the properties of the data elements and methods to be called using them.
- **Instances:** define copies of the classes that reference live data set as constant values, coming from SIOTH connectors or using expressions.

- **Integration with SIOTH® Connectors:**

Instances of defined classes can directly reference data retrieved by source connectors and make them available within the platform, including the SIOTH Job Engine rules and workflows.

- **Real-Time and Historical Data**

The model supports:

- **Real-time access** to the most recent attribute values.
- **Historical storage** of data, which can later be leveraged in the SIOTH Job Engine for analytics, monitoring, and rule execution.

4. Job Engine

SIOTH job Engine is a highly customizable and configurable component designed for **orchestration and synchronization**. It combines the power of both a **rule engine** and a workflow engine, enabling advanced automation and event-driven processing within the platform.

- **Rule Engine:**

Detects and processes complex events in real-time by applying user defined conditions and logic.

- **Workflow Engine:**

Executes predefined sequences of tasks or processes - whether configured to run sequentially or parallelly – ensuring efficient coordination and automation of operations.

- **Key Capabilities:**

- Execution of rules, workflows and scripts.
- Complex event detection, enabling proactive monitoring and alerts.
- Data transformation and aggregation for harmonizing information collected from diverse sources.

- Integration with data retrieved through SIOTH Data Flow from IoT devices, external data sources, and connected assets.

This flexible engine allows organizations to automate responses, enrich data, and implement intelligent event-driven architectures tailored to their operational needs.

5. Unified Control Interface

SIOTH® provides an all-in-one visualization system designed to simplify configuration and monitoring tasks across the platform. It delivers an intuitive, web-based graphical interface that allows users to easily interact with both IT and OT environments.

Key Features:

- **Environment Configuration:**

Manage and configure essential SIOTH® components, including:

- Nodes management
- Devices management
- Connectors and data flows
- Classes and instances
- Rules and workflows

- **Dashboard and SCADA Design:**

Create and configure web-based dashboards and SCADA displays to visualize real-time and historical data.

6. SIOTH® UHMI SCADA

SIOTH® UHMI SCADA enables users to build web-based industrial applications tailored to their operational needs. It merges traditional SCADA systems and dashboards into a single, unified interface that is web-deployed and accessible across phones, tablets, desktops, and multi-screen control room setups.

With **SIOTH UHMI SCADA**, users can:

- Monitor trends, devices health, and assets status in real-time.
- Visualize and interact with data through intuitive, customizable dashboards.
- Adapt visualization and control interfaces seamlessly to different environments.

SIOTH UHMI SCADA platform is composed of two environments:

- **UHMI SCADA Designer:**

This is the design environment, equivalent to an engineering workstation, where users can create and configure dashboards and SCADA displays.

- **UHMI SCADA Runtime:**

This is the runtime environment, where end-users perform their daily monitoring and operational tasks using the published dashboards and.

6.1. UHMI SCADA Designer

UHMI SCADA Designer is a graphical editor environment that allows users to create, configure, and manage all aspects of their web-based applications and user interface components. Key capabilities include:

- **Project creation:** set up new projects and define project parameters.
- **Layout configuration:** customize application layouts, including headers, footers, and navigation menus.
- **Data connectivity:** connect to multiple data sources and configure data streams and variables.
- **Dashboard and SCADA design:** design interactive dashboards and SCADA displays with customizable widgets and visualizations.
- **User management:** manage project users, roles, and access rights.
- **Publishing:** deploy and publish the completed applications to **UHMI SCADA Runtime** for end-user access.

6.2. UHMI SCADA Runtime

UHMI SCADA Runtime is a web-based runtime environment where all published applications are deployed. It provides end-users with:

- Access to published dashboards and SCADA displays.
- Real-time interaction with data sources and variables.
- A secure, role-based interface to ensure proper access control and operational security.

7. Scalability

SIOTH is a scalable platform designed to support multi-site integration and mission-critical applications. Its modular architecture allows deployment across a distributed cluster of nodes (physical or virtual) to provide:

- **Distributed Data Exchange:** efficiently distribute data collection processing, and transfer tasks among multiple nodes.
- **Redundancy and Quick Recovery:** duplicate jobs and services to ensure fast recovery in the event of a node or service failure.
- **Horizontal and Vertical Scaling:** scale locally within a single site or globally by interconnecting multiple sites.

This scalable architecture ensures that operations grow and adapt to evolving business requirements without compromising performance or reliability.

8. Flexible User Management

SIOTH platform provides flexible and secure user management, allowing administrator to configure user accounts, roles, and access privileges according to business needs. Key features include:

- **Identity management:** leverages OpenID Connect for centralized and standardized identity management.
- **Directory Integration:** Seamless integration with Microsoft Active Directory enables synchronization of users, groups and roles.

- **Modern Authentication:** Supports OAuth 2.0 protocol for secure and flexible authentication workflows.

This flexible approach ensures secure, scalable, and manageable access across multiple projects and operational environments.

APPLICATION CONFIGURATION

1. User Interface Overview

The **Application Configuration** module enables you to define and manage data integration flows according to specific business application requirements. This module can be accessed directly from the SIOTH home page by selecting the **Application Configuration** shortcut.

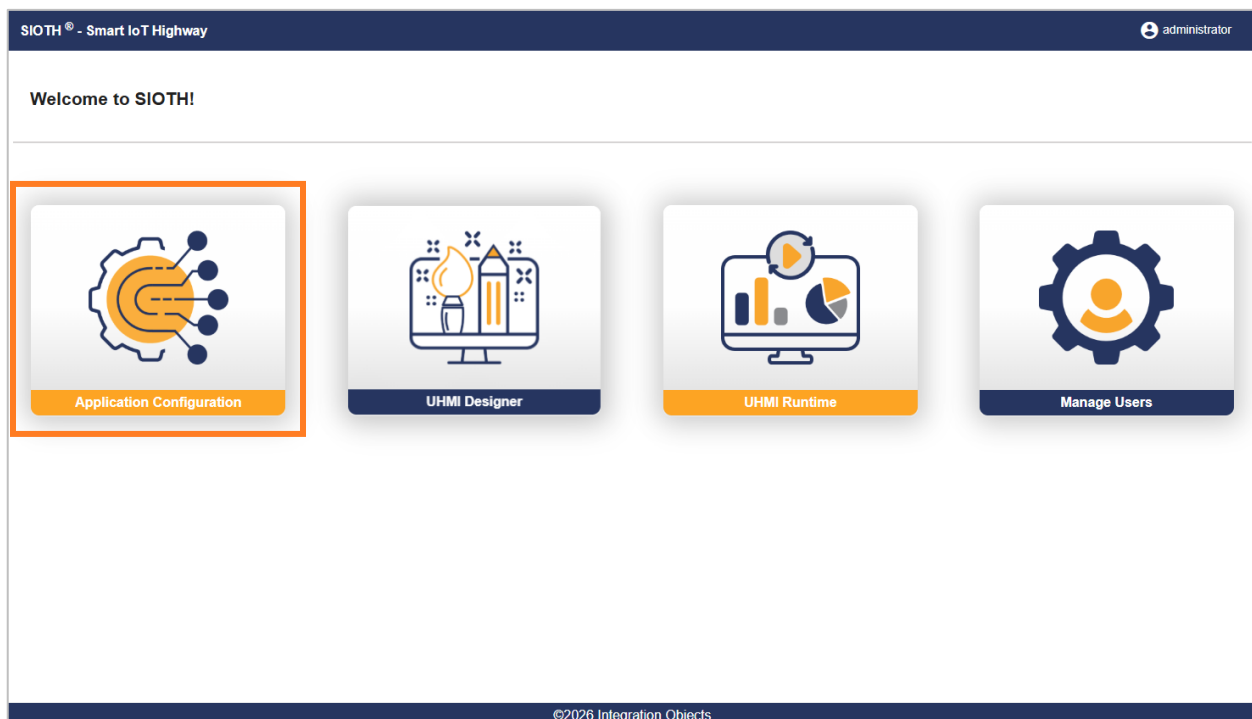


Figure 2: SIOTH® Home Page

Upon selection, users are redirected to the **Application Configuration** home page.

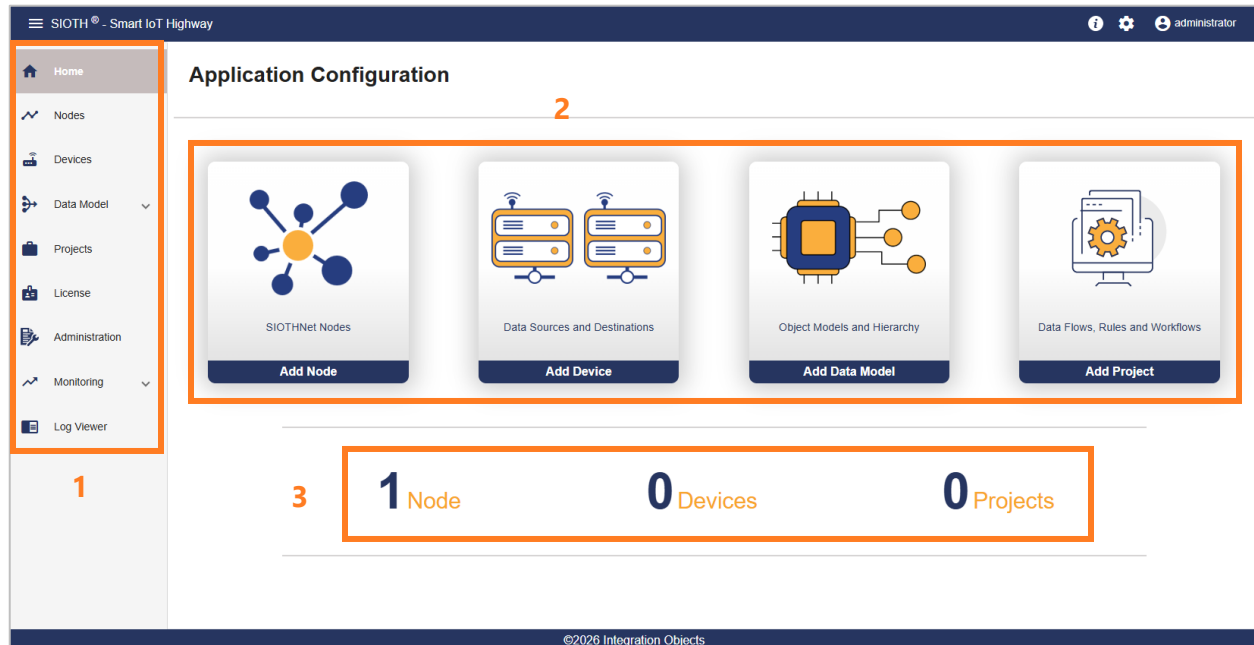


Figure 3: Application Configuration Home Page

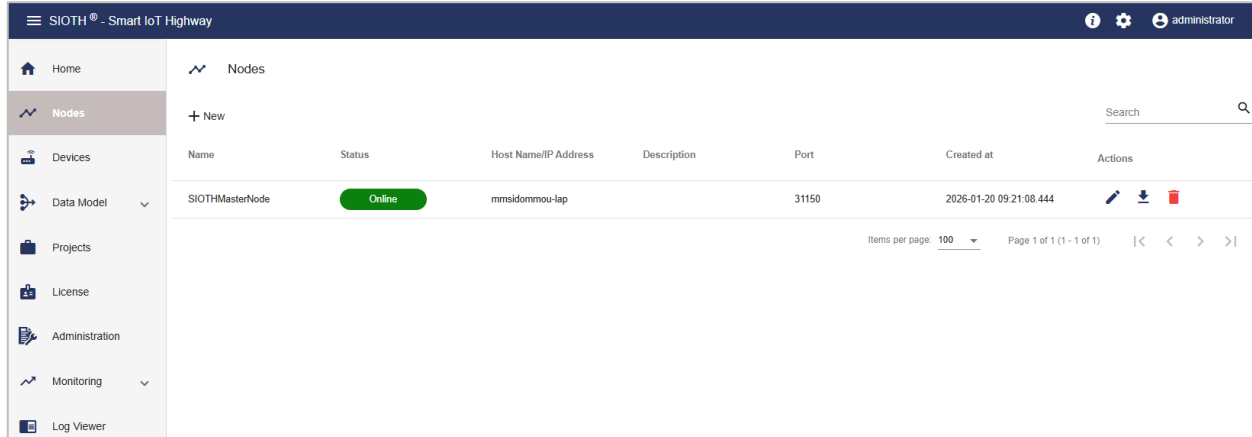
The Application Configuration page is organized into three main sections:

- **Left-Side Navigation Menu (1):** Provides access to configuration and monitoring elements related to the SIOTH application environment. Available sections include Nodes, Devices, Data Model, Projects, License, Administration, Monitoring, and Log Viewer.
- **Quick Access Cards (2):** Provides shortcuts for quickly creating new SIOTH Nodes, Devices, Data Model classes and instances and Projects.
- **Configuration Status (3):** Displays a summarized view of the number of nodes, devices, and projects currently configured within the SIOTH Application Configuration module.

2. Nodes

A **Node** in the SIOTH platform represents any machine on which SIOTH components are deployed. Nodes are responsible for receiving, creating, storing, and transmitting data across distributed network flows. They can be deployed either locally or remotely, depending on system architecture and operational requirements.

Click **Nodes** from the left sidebar menu. An explorer is displayed, listing all the added nodes, each one presented on a separate line.



Name	Status	Host Name/IP Address	Description	Port	Created at	Actions
SIOTHMasterNode	Online	mmsidommmou-lap		31150	2026-01-20 09:21:08.444	[Edit] [Download] [Delete]




Figure 4: Nodes Explorer

(!) Note

Nodes must be defined before starting the project configuration, particularly when SIOTH® components are deployed across multiple machines.

The primary node, named **SIOTHMasterNode**, is automatically created during installation. This node represents the local machine on which the SIOTH middleware is installed.

The following actions are available for each node:

-  **Edit:** Enables you to update the parameters related to the node.
-  **Download:** Allows you to download the log event viewer for the selected node.
-  **Delete:** Allows you to delete a node from the nodes list.

2.1. Add New Node

Click **New** in the Nodes Explorer section. A pop-up will be displayed, showing the parameters related to the new node.

Node

Name *
Node_1

Host Name/IP Address *

Description

Max 500 characters 0/500

☐ Offline

Installation Folder *
C:\Program Files\Integration Objects\Integration Objects' Smart IoT Highway

Port *
31150

Advanced Configuration ▼

CA Certificate Path *
C:\ProgramData\Integration Objects\CertificateStores\Integration Objects' Smart

Client Certificate Path *
C:\ProgramData\Integration Objects\CertificateStores\Integration Objects' Smart

Client Certificate Password *

Cancel Save

Figure 5: Add New Node Configuration View

Parameter	Description	Default Value
<i>Name</i>	Unique identifier for the node. It is generated automatically but can be updated to a user-friendly value.	Node_1
<i>Host Name/IP Address</i>	Host name or IP address of the machine on which the node will be deployed.	


Description	Optional field for providing additional information about the node.	
Offline	Indicates whether the node deployment operates in offline mode or not.	Unchecked
Installation Folder	Installation directory of the SIOTH platform on the node..	Smart IoT Highway installation folder
Port	TCP port used for communication with the node.	31150
Advanced Configuration		
CA certificate Path	Path to the Certificate Authority (CA) certificate used to establish secure communications.	Smart IoT Highway MQTT Master Broker folder
Client Certificate Path	Path to the client certificate used to establish mutual authentication between systems.	Smart IoT Highway MQTT Master Broker folder
Client Certificate Password	Password protecting the client certificate.	

Table 1: Node Configuration Parameters

Click **Save** to save the node configuration and add it to the list.

2.2. Edit Node

To update a node configuration, follow these steps:


1. Open the **Nodes** page by clicking **Nodes** from the left side bar menu.
2. Locate the node you want to edit.
3. Click the **Edit** icon  in the **Actions** column of the nodes' explorer. A pop-up is open, displaying the current configuration of the node.
4. Edit the description and click **Save** to submit the changes.

(!) Note

For an existing node, the **Description** field is the only parameter that can be modified.


2.3. Download Log Event Viewer

To download the log event viewer of a node, follow these steps:

1. Open the **Nodes** page by clicking **Nodes** from the left side bar menu.
2. Locate the required node.
3. Click the **Download** icon  in the **Actions** column of the Nodes explorer.

2.4. Delete Node

To delete a node, follow these steps:

1. Open the **Nodes** page by clicking **Nodes** from the left side bar menu.
2. Locate the node you want to delete.
3. Click the **Delete** icon  in the **Actions** column of the Nodes explorer.

A confirmation dialog is displayed.

- Click **Yes** to permanently delete the node.
- Click **No** to abort the operation and retain the node.

(!) Note

The master node, **SIOTHMasterNode**, is permanent and cannot be deleted.

3. Devices

Devices in the SIOTH platform represent IT or OT systems, applications, equipment, or databases from which data can be retrieved or to which data can be delivered.

Click **Devices** from the left sidebar menu. An explorer is displayed, listing all the devices, each one presented on a separate line.

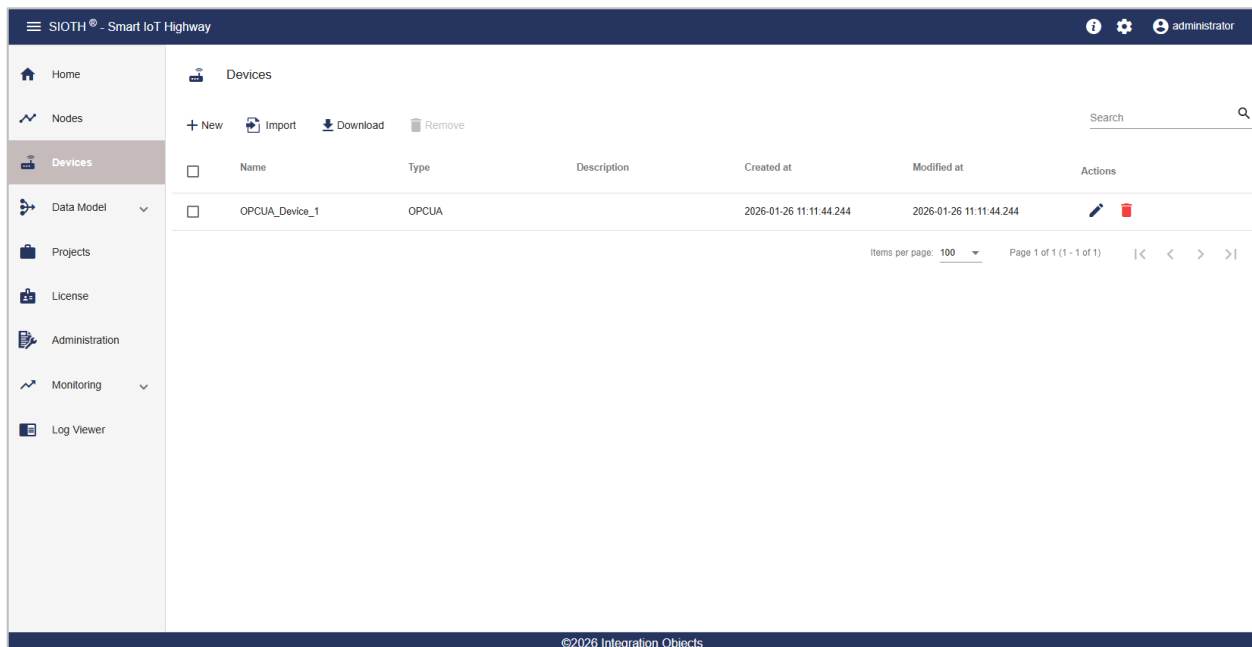




Figure 6: Devices Explorer

For each device, the following actions are available:

 **Edit:** Allows modification of the device's configuration parameters.

 **Delete:** Deletes the device from the list of configured devices.

3.1. Add New Device

Click **New** in the **Devices** Explorer section. A pop-up will be displayed, showing the parameters related to the new device.



The image shows a configuration window for adding a new device. It features a progress bar at the top with two steps: '1 Identification' (highlighted) and '2 Configuration'. Below the progress bar, there are three input fields: 'Type *' with a dropdown menu showing 'OPC', 'Name *' with the text 'OPC_Device_1', and 'Description' which is a text area. At the bottom left, it says 'Max 500 characters' and '0/500'. At the bottom right, there are two buttons: 'Cancel' and 'Next'.

Figure 7: Add New Device Configuration View

The New Device Configuration process is organized into two steps:

1. Device Selection:

Select the device type, assign a name, and optionally provide a meaningful description. Once completed, click **Next**.

2. Connection Configuration:

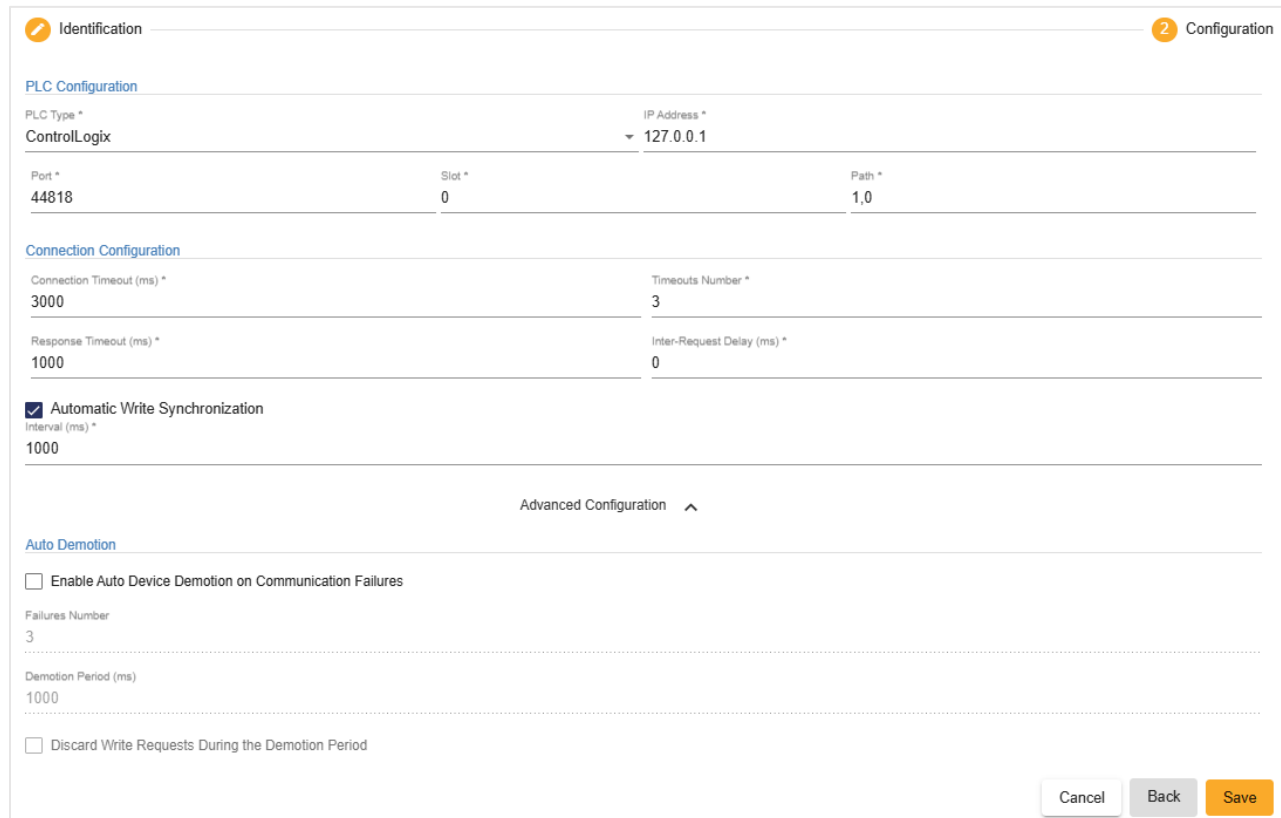
Specify the connection parameters required for the device. This required parameters vary depending on the selected device type. Detailed instructions for configuring each device type are provided in the following subsections.

Device	Description
Allen Bradley	Devices supporting Allen Bradley protocol.
AMQP-091	Brokers supported AMQP-091.
Azure Event Hub	Microsoft Azure Event Hub service for large-scale data ingestion.
BACnet	Devices supporting BACnet protocol.
Data Stores	SQL like databases, historians, No-SQL databases, PI, PI AF, Apache Kafka, InfluxDB.
DNP3	RTUs and PLCs that support the DNP3 protocol.
FTP	FTP servers
HartIP	Devices supporting HartIP protocol.
IEC 60870-5-104	Devices supporting IEC 60870-5-104 protocol.
J1939	Devices supporting J1939 protocol.
Modbus	Modbus devices (TCP and Serial).
MQTT	MQTT Brokers (Standard and Sparkplug).
OPC AE	OPC Classic AE Servers.
OPC DA	OPC Classic DA Servers.
OPC HDA	OPC Classic HDA Servers.
OPC UA	OPC UA Servers.
S7	RTUs and PLCs supporting S7 protocol.
SMS Server	SMS servers are used for sending text notifications.
SMTP Server	SMTP servers are used for sending email notifications.
SNMP	SNMP-enabled devices such as servers, workstations, network routers, switches, firewalls, UPS (Uninterrupted Power Supply) systems, PLCs, and other network devices.

Table 2: Supported Devices

3.1.1. Allen Bradley

Select **Allen Bradley** from the **Type** drop-down list, then click **Next**.



The screenshot shows the 'Configuration' tab of the Allen Bradley device configuration interface. It is divided into three main sections: PLC Configuration, Connection Configuration, and Auto Demotion.

PLC Configuration:

- PLC Type *: ControlLogix (dropdown menu)
- IP Address *: 127.0.0.1
- Port *: 44818
- Slot *: 0
- Path *: 1,0

Connection Configuration:

- Connection Timeout (ms) *: 3000
- Timeouts Number *: 3
- Response Timeout (ms) *: 1000
- Inter-Request Delay (ms) *: 0

Automatic Write Synchronization:

- ☒ Automatic Write Synchronization
- Interval (ms) *: 1000

Advanced Configuration: (collapsed section)

Auto Demotion:

- ☐ Enable Auto Device Demotion on Communication Failures
- Failures Number: 3
- Demotion Period (ms): 1000
- ☐ Discard Write Requests During the Demotion Period

Buttons at the bottom right: Cancel, Back, Save.

Figure 8: Allen Bradley Device Configuration View

Parameter	Description	Default Value
PLC Type	Type and model of the Allen Bradley PLC. Supported types are: <ul style="list-style-type: none"> ControlLogix Plc5 Slc500 Micro800 MicroLogix 	ControlLogix
IP Address	IP address of the target device.	127.0.0.1

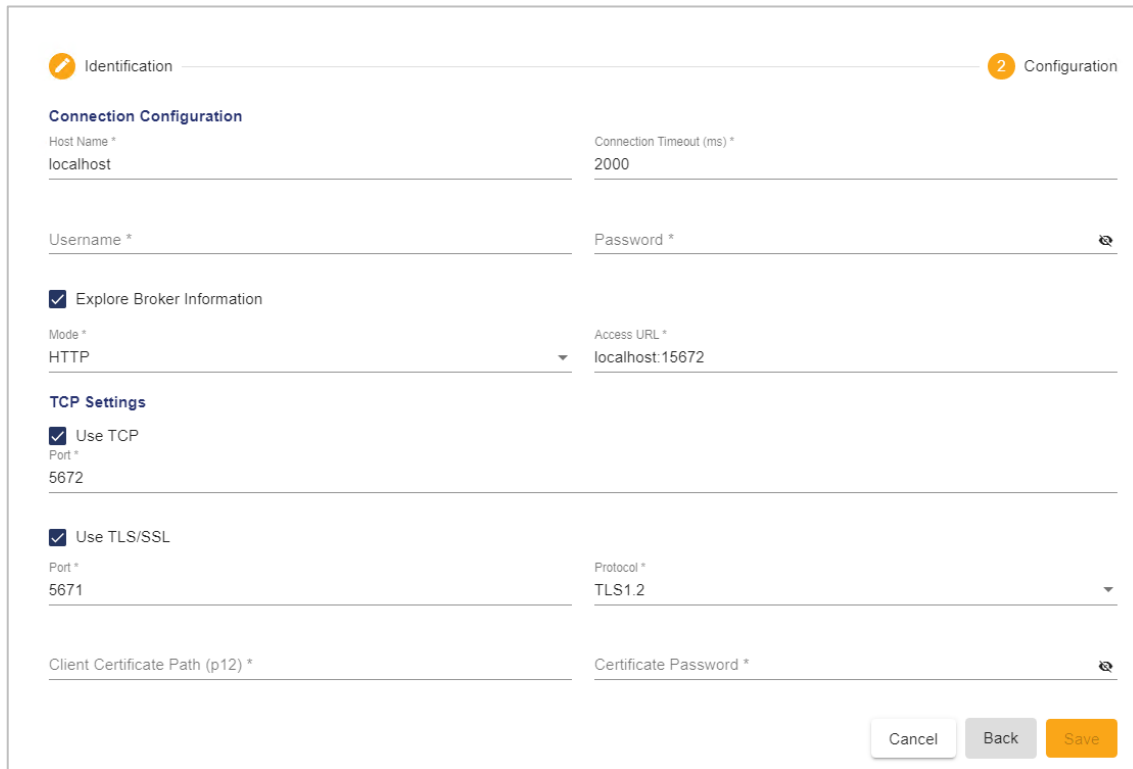
Port	TCP port used to connect to the PLC.	44818
Slot	<p>Physical location within the PLC chassis where modules are inserted. Module can be:</p> <ul style="list-style-type: none"> • Input: read signals from sensors. • Output: control actuators and devices. • Communication: network connectivity. • Special functions: specific tasks such as motion control. <p>Slot number identifies the module's location in the PLC.</p>	0
Path	Addressing scheme used to access specific data within the PLC's memory or the I/O modules.	1,0
Connection Configuration		
Connection Timeout (ms)	Maximum time to wait for a response from the device.	3000
Timeouts Number	Number of retries if no valid response is received.	3
Response Timeout (ms)	Time the device waits after sending a read request.	1000
Inter-Request Delay (ms)	Time interval between successive requests sent by the master to the device.	0
Automatic Write Synchronization	Enhances performance for multiple write operations.	Enabled
Interval (ms)	Interval for automatic writing synchronization.	1000
Advanced Configuration		
Enable Auto Device	Temporarily demotes the device after a configured number of communication failures.	Unchecked

<i>Demotion on Communication Failures</i>		
<i>Failures Number</i>	Number of successive failures before the device is demoted.	3
<i>Demotion Period (ms)</i>	Time during which no read requests are sent to the device after demotion.	1000
<i>Discard Write Requests During the Demotion Period</i>	When enabled, write requests are not sent during the demotion period.	Unchecked

Table 3: Allen Bradley Device Configuration Parameters

3.1.2. AMQP-091

Select **AMQP-091** from the **Type** drop-down list, then click **Next**.



The screenshot shows the 'Configuration' tab of the AMQP-091 device configuration interface. It is divided into two main sections: 'Connection Configuration' and 'TCP Settings'.

Connection Configuration:

- Host Name ***: localhost
- Connection Timeout (ms) ***: 2000
- Username ***: (empty)
- Password ***: (empty, with a password icon)
- ☒ **Explore Broker Information**
- Mode ***: HTTP (dropdown menu)
- Access URL ***: localhost:15672

TCP Settings:

- ☒ **Use TCP**
- Port ***: 5672
- ☒ **Use TLS/SSL**
- Port ***: 5671
- Protocol ***: TLS1.2 (dropdown menu)
- Client Certificate Path (p12) ***: (empty)
- Certificate Password ***: (empty, with a password icon)

At the bottom right, there are three buttons: 'Cancel', 'Back', and 'Save'.

Figure 9: AMQP-091 Device Configuration View

Parameter	Description	Default Value
Connection Configuration		
Host Name	Hostname or IP address of the AMQP server.	localhost
Connection Timeout (ms)	Maximum time to wait when establishing a connection to the AMQP server.	2000
Username	Username for authenticating with the AMQP server.	
Password	Password for authenticating with the AMQP server.	
Broker Information		
Explore Broker Information	When enabled, it provides access to RabbitMQ Broker parameters.	Checked
Mode	Connection mode based on RabbitMQ configuration. Supported modes are: <ul style="list-style-type: none"> • HTTP. • HTTPS. 	HTTP
Access URL	URL of the RabbitMQ Management UI (Management Console).	Localhost:15672
TCP Settings		
Use TCP	Enables TCP connection to the broker.	Checked
Port	AMQP Broker port for message reception. Common defaults: 5672 for AMQP-091 over TCP, or 5671 for TLS/SSL connections.	5672

<i>TLS/SSL Settings</i>		
<i>Use TLS/SSL</i>	Enables secure communication using TLS/SSL.	Unchecked
<i>Port</i>	AMQP Broker port for TLS/SSL connections.	5671
<i>Protocol</i>	Supported TLS/SSL protocols: <ul style="list-style-type: none"> • TLS 1.0 • TLS 1.1 • TLS 1.2 • SSL 3 	TLS1.2
<i>Client Certificate Path (p12)</i>	Full path to the client digital certificate, including its name and extension, in p12 format used for authentication.	
<i>Certificate Password</i>	Password protecting the client certificate.	

Table 4: AMQP-091 Device Configuration Parameters
(!) Note

Communication over TLS will fail if the certificate file is not in p12 format.

3.1.3. Azure Event Hub

Select **Azure Event Hub** from the **Type** drop-down list, then click **Next**.



Figure 10: Azure Event Hub Device Configuration View

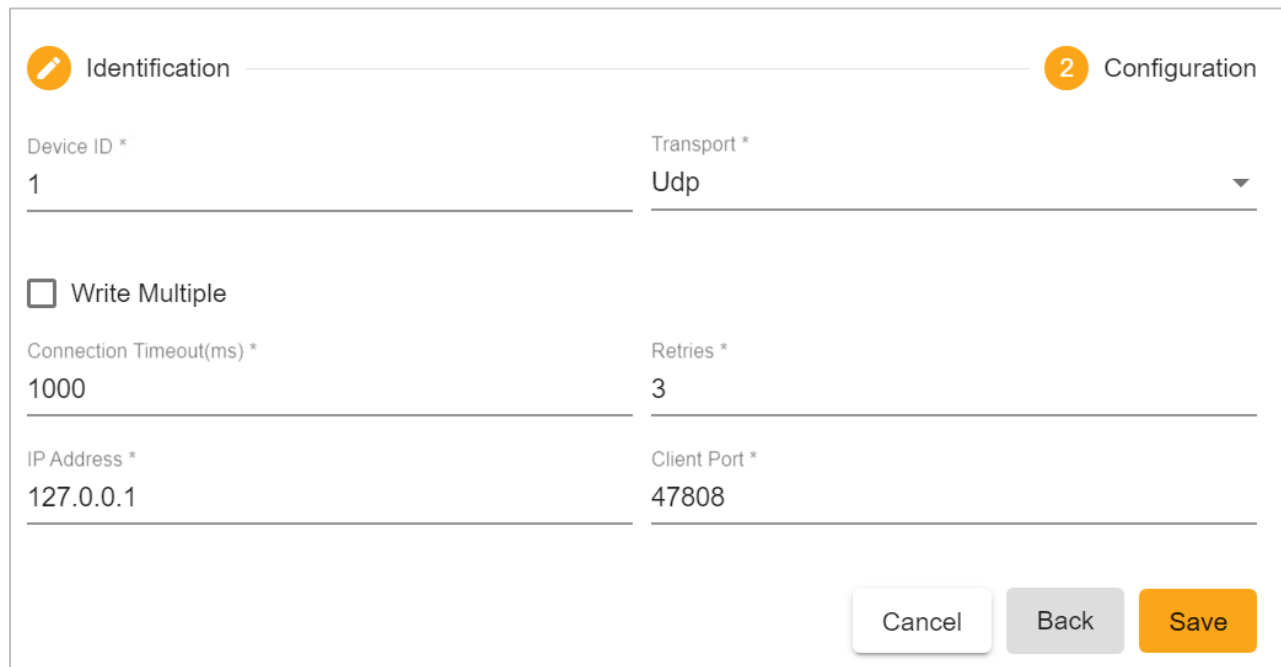
Parameter	Description	Default Value
Host Name	Hostname or IP address of the Event Hub server.	
Event Hub NameSpace	Namespace of the Event Hub to connect to.	
Connection String	Required connection string for AMQPS and KAFKA protocols.	
Shared Access Signature for HTTPS	Defines the type of SAS (Shared Access Signature) to use: <ul style="list-style-type: none"> SAS Policy: Standard SAS authentication policy. SAS Token: Direct token for access. 	SAS Policy
Policy	Policy associated with the selected SAS policy.	
Key	Key associated with the selected SAS policy.	

Token	Authentication token used when SAS Token is selected as the policy.	
--------------	---	--

Table 5: Azure Event Hub Device Configuration Parameters

3.1.4. BACnet

Select **BACnet** from the **Type** drop-down list, then click **Next**.



The screenshot shows the BACnet Device Configuration View. It has two tabs: 'Identification' (active) and 'Configuration'. Under 'Identification', there are fields for 'Device ID *' (value: 1), 'Transport *' (value: Udp), 'Write Multiple' (checkbox), 'Connection Timeout(ms) *' (value: 1000), 'Retries *' (value: 3), 'IP Address *' (value: 127.0.0.1), and 'Client Port *' (value: 47808). At the bottom right are 'Cancel', 'Back', and 'Save' buttons.

Figure 11: BACnet Device Configuration View

Parameter	Description	Default Value
Device ID	Unique identifier of the device. Must be unique within the BACnet network to avoid conflicts.	1
Transport	Select the communication protocol used to connect to the device: <ul style="list-style-type: none"> UDP MSTP 	UDP

	<ul style="list-style-type: none"> Ipv6 	
Write Multiple	Allows writing multiple points in a single request.	Unchecked
Connection Timeout (ms)	Time (in milliseconds) to wait between retransmissions of an Application Layer Protocol Data Unit (APDU) requiring acknowledgement when no acknowledgement has been received. Only applicable if the number of retries is greater than zero.	1000
Retries	Maximum number of times an APDU is retransmitted. <ul style="list-style-type: none"> 0: no retries. >0: enables APDU Timeout attribute. 	3
IP Address	IP address of the target BACnet module.	127.0.0.1
Client Port	Port used to communicate with the device.	47808

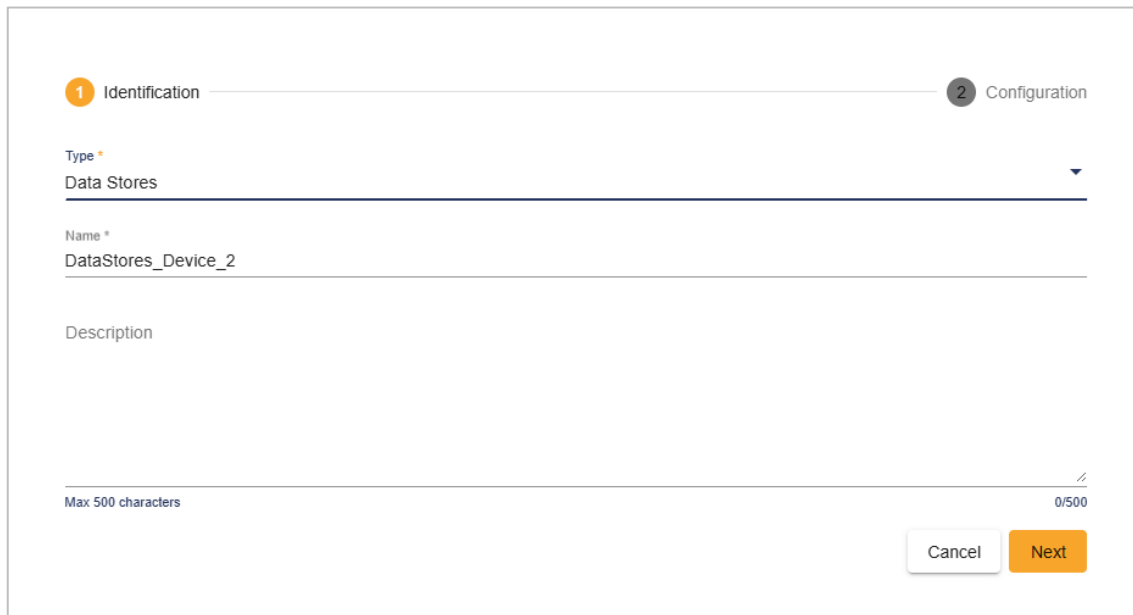
Table 6: BACnet Device Configuration Parameters

3.1.5. Data Stores

SIOTH supports integration with a broad range of data store types, including:

- **Relational Databases:** SQL Server, Oracle, MS Access, MySQL, PostgreSQL.
- **ODBC-Compatible Sources:** ODBC connections.
- **Time-Series and NoSQL Databases:** InfluxDB, MongoDB, Redis DB.
- **Industrial Data Stores:** PI, PI AF.
- **Streaming Platforms:** Kafka.

Select **Data Stores** from the **Type** drop-down list, then click **Next**. The configuration screen will appear, where you must provide the required connection parameters:



The form is titled "Data Stores Device Configuration View" and is divided into two sections: "1 Identification" and "2 Configuration".

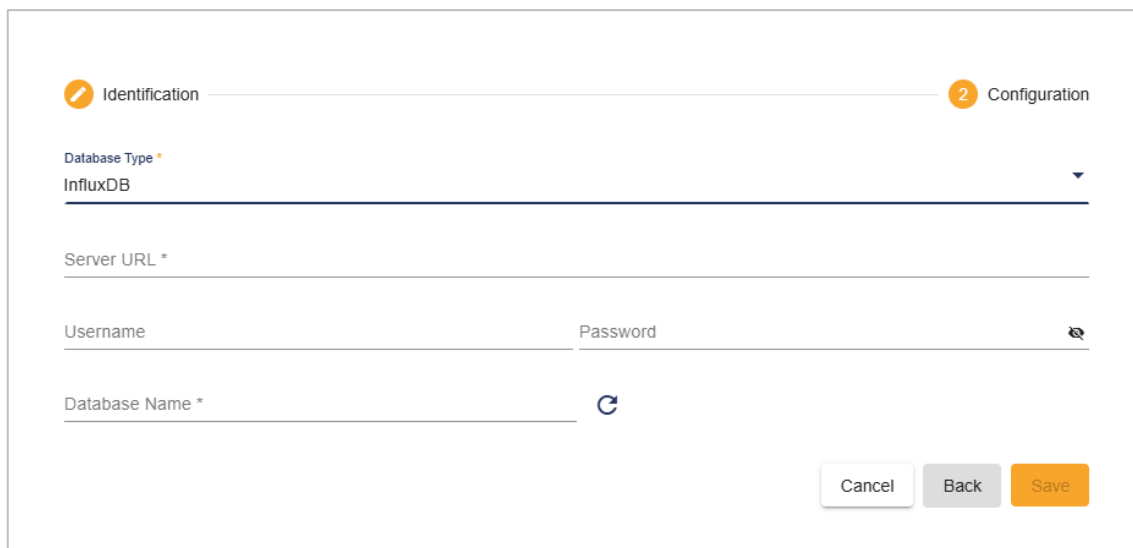
1 Identification

- Type ***: A dropdown menu with "Data Stores" selected.
- Name ***: A text input field containing "DataStores_Device_2".
- Description**: A text input field.
- Max 500 characters**: A label indicating the character limit.
- 0/500**: A character count indicator.
- Buttons**: "Cancel" and "Next" buttons.

Figure 12: Data Stores Device Configuration View

3.1.5.1. InfluxDB

Select **InfluxDB** from the **Database Type** drop-down list.



The form is titled "InfluxDB Device Configuration View" and is divided into two sections: "1 Identification" and "2 Configuration".

1 Identification

- Database Type ***: A dropdown menu with "InfluxDB" selected.
- Server URL ***: A text input field.
- Username**: A text input field.
- Password**: A text input field with a password icon.
- Database Name ***: A text input field with a refresh icon.
- Buttons**: "Cancel", "Back", and "Save" buttons.

Figure 13: InfluxDB Device Configuration View

Parameter	Description	Default Value
Server URL	URL of the InfluxDB server. The URL must start with http:// and may include a domain name or IP address. The port is optional; if not specified, it defaults to 8086 .	
Username	Username for authenticating with the InfluxDB server.	
Password	Password associated with the specified username.	
Database Name	Name of the target InfluxDB database. Use the refresh icon to browse and select from available databases.	

Table 7: InfluxDB Device Configuration Parameters

3.1.5.2. Kafka

Select **Kafka** from the **Database Type** drop-down list.

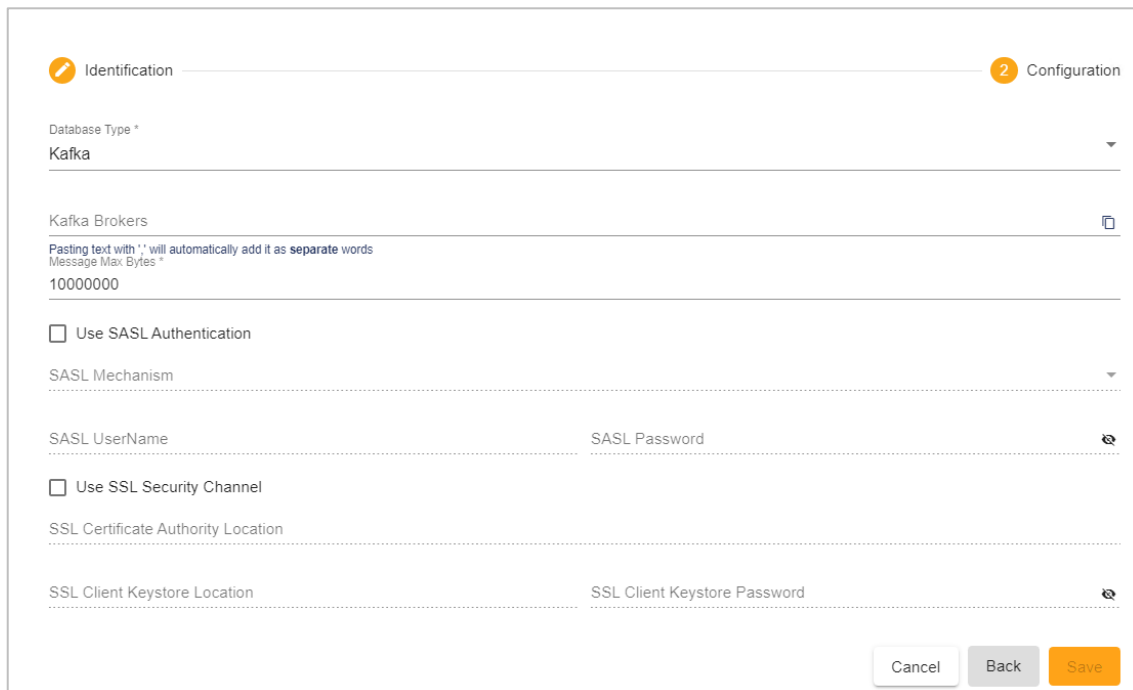


Figure 14: Kafka Device Configuration View

Parameter	Description	Default Value
<i>Kafka Brokers</i>	A Kafka broker receives messages from producers and stores them on disk keyed by unique offset. Brokers also allow consumers to fetch messages by topic, partition, and offset. Enter the broker and click + to add it.	
<i>Message Max Bytes</i>	Maximum message batch size accepted by the broker.	10000000
SASL Authentication		
<i>Use SASL Authentication</i>	Check this option to enable SASL authentication with Kafka.	Unchecked
<i>SASL Mechanism</i>	SASL mechanism to use. Options are: <ul style="list-style-type: none"> • Plain • ScreamSha256 • ScreamSha512 	
<i>SASL Username</i>	Username used for SASL authentication.	
<i>SASL Password</i>	Password associated with the SASL username and used for SASL authentication.	
SSL Security Channel		
<i>Use SSL Security Channel</i>	Check this option to enable SSL security.	Unchecked

SSL Certificate Authority Location	Path to the SSL Certificate Authority file.	
SSL Client Keystore Location	Path to the SSL Client Keystore file.	
SSL Client Keystore Password	Password for the SSL Client Keystore.	

Table 8: Kafka Device Configuration Parameters

3.1.5.3. MongoDB

Select **MongoDB** from the **Database Type** drop-down list.

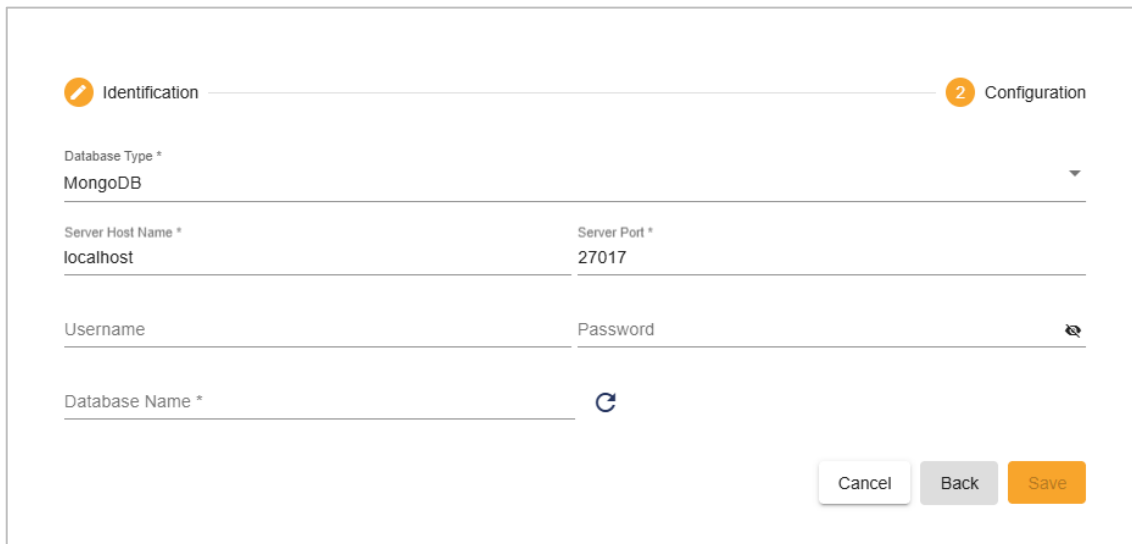


Figure 15: MongoDB Device Configuration View


Parameter	Description	Default Value
Server Host Name	IP address or host name of the MongoDB server.	localhost
Server Port	Port number used by the MongoDB service.	27017
Username	Username required to authenticate with the MongoDB server.	
Password	Password associated with the specified username for authentication.	
Database Name	Name of the database to connect to. You can use the refresh icon  to browse the available databases. Name of the database to connect to. Use the refresh icon to browse available databases.	

Table 9: MongoDB Device Configuration Parameters

3.1.5.4. MS Access

Select **MS Access** from the **Database Type** drop-down list.



The screenshot shows the 'MS Access' configuration view. At the top, there are two tabs: 'Identification' (active) and 'Configuration' (indicated by a '2' in a circle). Below the tabs, the 'Database Type' is set to 'MSAccess'. The 'Response Timeout (ms) *' is set to '30000'. The 'File Path *' field is empty. There is a checkbox for 'Database Password' which is currently unchecked. Below this, the 'Database Password' field is shown with a masked password (dots) and a refresh icon. At the bottom right, there are three buttons: 'Cancel', 'Back', and 'Save'.

Figure 16: MS Access Device Configuration View

Parameter	Description	Default Value
Response Timeout (ms)	Maximum time, in milliseconds, to wait for a response from the database before timing out.	30000
File Path	Full path to the Microsoft Access database file.	
Database Password	Password for the Microsoft Access database, required if the file is password-protected.	

Table 10: MS Access Device Configuration Parameters

3.1.5.5. MySQL

Select **MySQL** from the **Database Type** drop-down list.

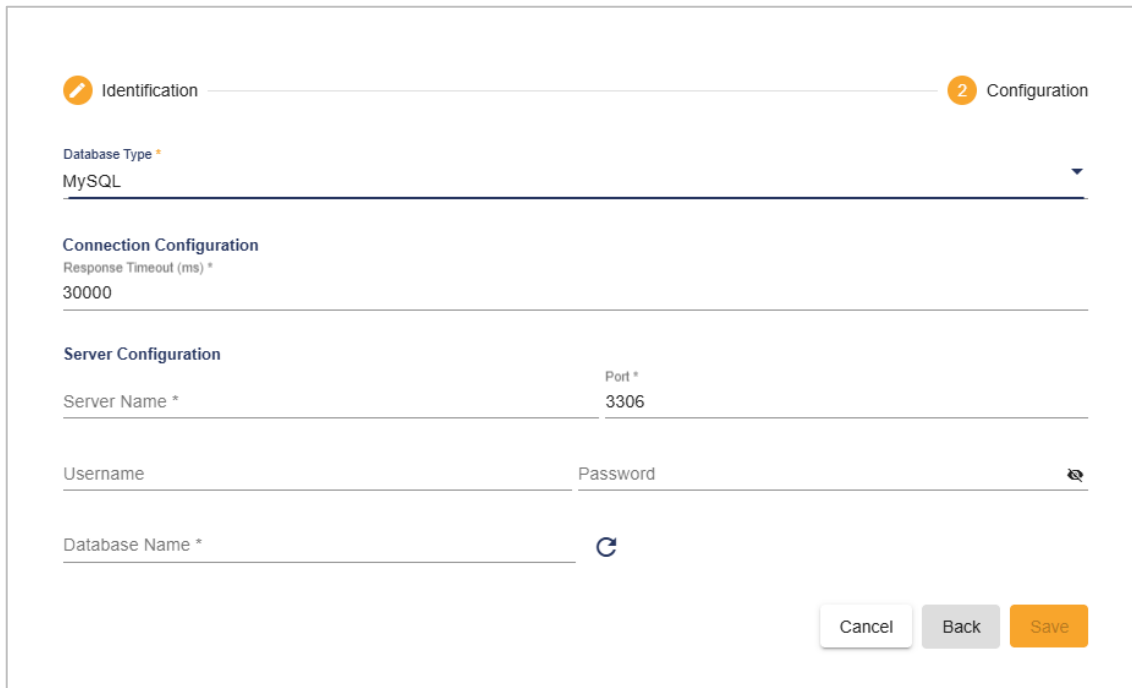


Figure 17: MySQL Device Configuration View


Parameter	Description	Default Value
Response Timeout (ms)	Maximum time, in milliseconds, to wait for a response from the database before timing out.	30000
Server Name	Name or IP address of the MySQL server instance.	
Port	Port used by MySQL clients, connectors, and utilities.	3306
Username	Username used to connect to the MySQL database.	
Password	Password associated with the username for authentication.	
Database Name	Name of the MySQL database. Use the refresh icon  to browse and select from available databases.	

Table 11: MySQL Device Configuration Parameters

3.1.5.6. ODBC

Select **ODBC** from the **Database Type** drop-down list.



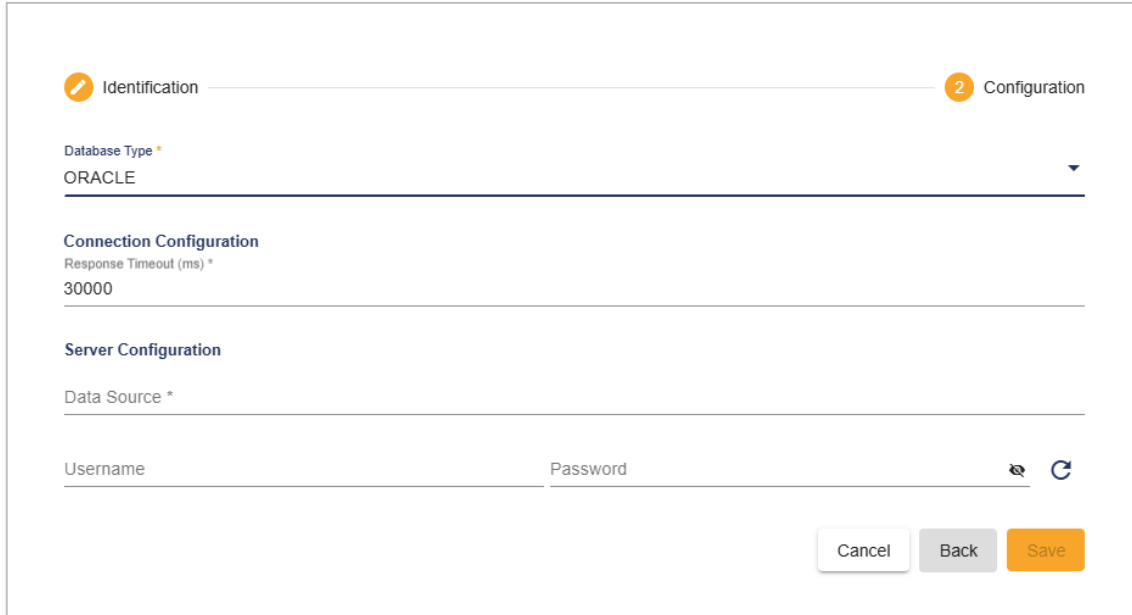
Figure 18: ODBC Device Configuration View

Parameter	Description	Default Value
<i>Connection String</i>	<p>Connection string used to connect to the ODBC data source.</p> <p>Examples: DBC SQL Server connection string:</p> <ul style="list-style-type: none"> Without credentials: Driver={SQL Server};Server=.\sqlexpress; Database=master;Trusted_Connection=True; With credentials: Driver={SQL Server};Server=.\SQLEXPRESS;Database=Test;UID= <username>;PWD= <password>; 	
<i>Response Timeout (ms)</i>	Maximum time, in milliseconds, to wait for a response from the database before timing out.	30000

Table 12: ODBC Device Configuration Parameters

3.1.5.7. Oracle

Select **Oracle** from the **Database Type** drop-down list.



The screenshot shows the Oracle Device Configuration View. At the top, there are two tabs: 'Identification' (active) and 'Configuration'. Below the tabs, the 'Database Type' is set to 'ORACLE'. Under 'Connection Configuration', the 'Response Timeout (ms) *' is set to '30000'. Under 'Server Configuration', the 'Data Source *' is empty. At the bottom, there are fields for 'Username' and 'Password', with a 'Cancel' button, a 'Back' button, and a 'Save' button.

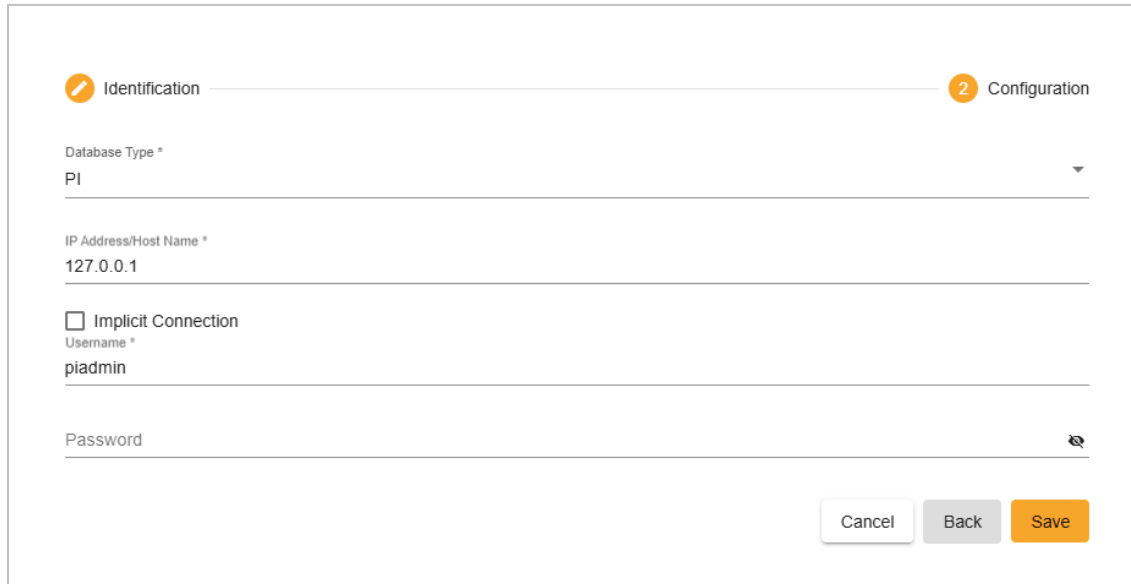
Figure 19: Oracle Device Configuration View

Parameter	Description	Default Value
Response Timeout (ms)	Maximum time, in milliseconds, to wait for a response from the database before timing out.	30000
Data Source	The Oracle data source name specifying the target database to connect to.	
Username	Oracle user account used for authentication.	
Password	Password associated with the Oracle user account.	

Table 13: Oracle Device Configuration Parameters

3.1.5.8. PI

Select **PI** from the **Database Type** drop-down list.



The screenshot shows a configuration window with two tabs: 'Identification' (active) and 'Configuration'. The 'Identification' tab contains the following fields:

- Database Type ***: A drop-down menu with 'PI' selected.
- IP Address/Host Name ***: A text field containing '127.0.0.1'.
- Implicit Connection**: An unchecked checkbox.
- Username ***: A text field containing 'piadmin'.
- Password**: A text field with a toggle icon on the right.

At the bottom right, there are three buttons: 'Cancel', 'Back', and 'Save'.

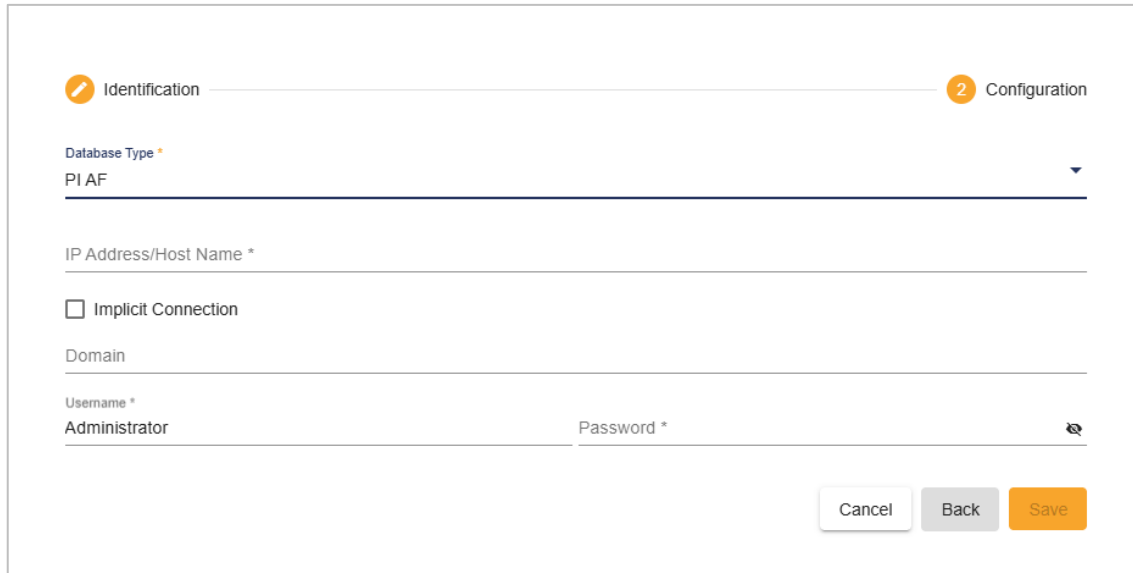
Figure 20: PI Device Configuration View

Parameter	Description	Default Value
<i>IP Address / Host Name</i>	P address or host name of the PI server.	127.0.0.1
<i>Implicit Connection</i>	Determines whether the connector connects implicitly or explicitly. Check to enable implicit connection.	Unchecked
<i>Username</i>	User account used to connect to the PI server when Implicit Connection is disabled.	piadmin
<i>Password</i>	Password associated with the username, required when Implicit Connection is disabled.	

Table 14: PI Device Configuration Parameters

3.1.5.9. PI AF

Select **PI AF** from the **Database Type** drop-down list.



The screenshot shows a configuration window for PI AF. At the top, there are two tabs: 'Identification' (active) and 'Configuration'. Below the tabs, the 'Database Type' is set to 'PI AF'. There is a text field for 'IP Address/Host Name *'. Below that is a checkbox for 'Implicit Connection'. Underneath the checkbox is a text field for 'Domain'. At the bottom, there are two text fields: 'Username *' (containing 'Administrator') and 'Password *' (with a password icon). At the bottom right, there are three buttons: 'Cancel', 'Back', and 'Save'.

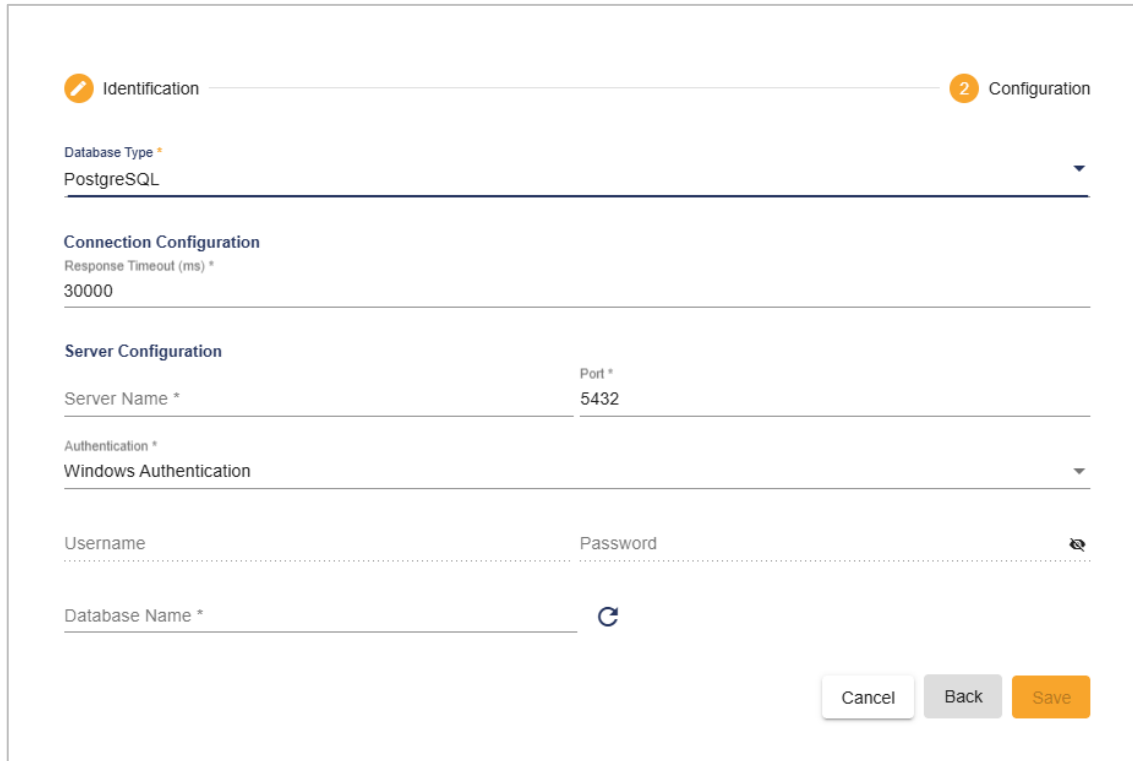
Figure 21: PI AF Device Configuration View

Parameter	Description	Default Value
IP Address / Host Name	IP address or host name of the PI AF server.	
Implicit Connection	Determines whether the connector connects implicitly or explicitly. Check to enable implicit connection.	Unchecked
Domain	Domain used for authentication against the PI AF server when Implicit Connection is disabled.	
Username	User account used to connect to the PI AF server when Implicit Connection is disabled.	Administrator
Password	Password associated with the username, required when Implicit Connection is disabled.	

Table 15: PI AF Device Configuration Parameters

3.1.5.10. PostgreSQL

Select **PostgreSQL** from the **Database Type** drop-down list.



The image shows a web-based configuration interface for PostgreSQL. At the top, there are two tabs: 'Identification' (active) and 'Configuration'. Below the tabs, the 'Database Type' is set to 'PostgreSQL'. Under 'Connection Configuration', the 'Response Timeout (ms)' is set to '30000'. The 'Server Configuration' section includes 'Server Name', 'Port' (set to '5432'), 'Authentication' (set to 'Windows Authentication'), 'Username', 'Password', and 'Database Name'. There are 'Cancel', 'Back', and 'Save' buttons at the bottom right.

Figure 22: PostgreSQL Device Configuration View

Parameter	Description	Default Value
Response Timeout (ms)	Maximum time, in milliseconds, to wait for a response from the database before timing out.	30000
Server Name	Name or IP address of the PostgreSQL server.	
Port	Port number on which the PostgreSQL server is listening.	5432
Authentication	Mode used to authenticate with the PostgreSQL server. Options:	Windows Authentication


	<ul style="list-style-type: none"> • Windows Authentication: Uses Windows identity. • Standard Authentication: Requires username and password. 	
<i>Username</i>	Username for connecting to the PostgreSQL server (required only for Standard Authentication).	
<i>Password</i>	Password associated with the username (required only for Standard Authentication).	
<i>Database Name</i>	Name of the PostgreSQL database to connect to. You can use the refresh icon  to browse available databases.	

Table 16: PostgreSQL Device Configuration Parameters

3.1.5.11. SQL Server

Select **SQL Server** from the **Database Type** drop-down list.

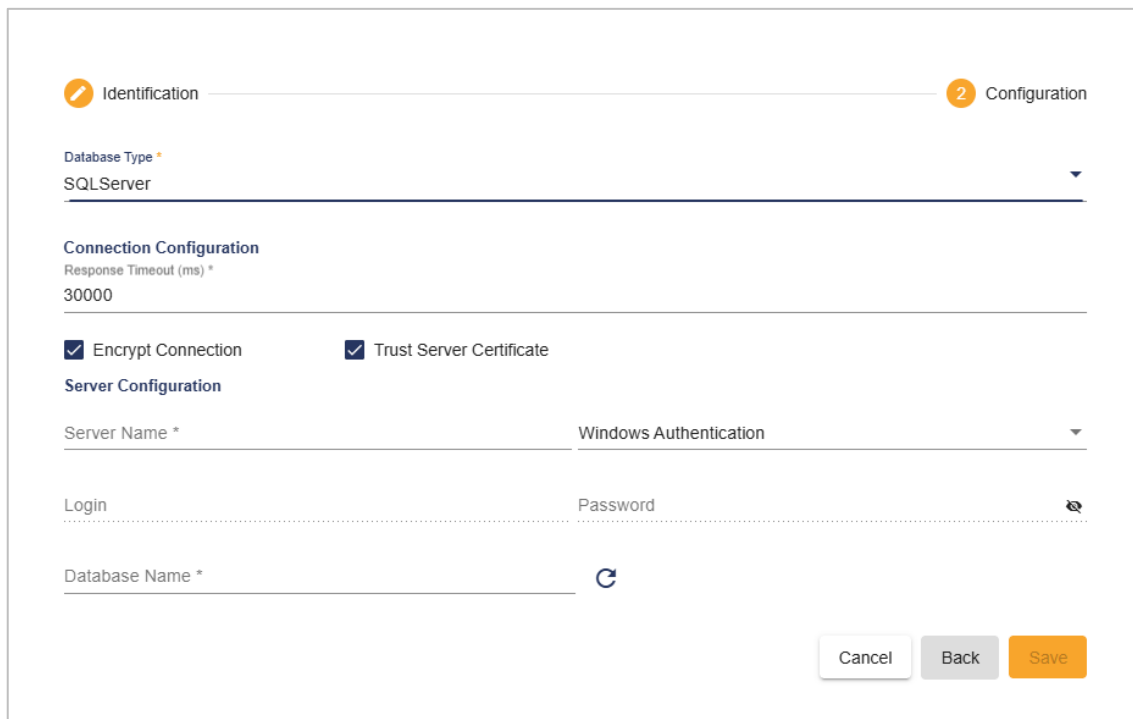


Figure 23: SQL Server Device Configuration View


Parameter	Description	Default Value
<i>Response Timeout (ms)</i>	Maximum time, in milliseconds, to wait for a response from the database before timing out.	30000
<i>Encrypt Connection</i>	Enables encryption for the SQL Server connection.	Checked
<i>Trust Server Certificate</i>	Accepts the server's SSL certificate without validation, useful for self-signed certificates.	Checked
<i>Server Name</i>	Specifies the SQL Server instance name: <ul style="list-style-type: none"> Default instance: use the machine name. Named instance: use <computer_name>\<instance_name> (e.g., DBSRVR\SQLEXPRESS). 	
<i>Authentication</i>	Mode used to connect to the SQL Server. Options: <ul style="list-style-type: none"> Windows Authentication: Uses Windows credentials. SQL Server Authentication: Requires login and password. 	Windows Authentication
<i>Login</i>	Login used to connect to the SQL Server instance (required only for SQL Server Authentication).	
<i>Password</i>	Password associated with the login (required only for SQL Server Authentication).	
<i>Database Name</i>	Name of the target SQL Server database. Use the refresh icon  to browse and select from available databases.	

Table 17: SQL Server Device Configuration Parameters

3.1.6. DNP3

Select **DNP3** from the **Type** drop-down list, then click **Next**.

1 Identification

2 Configuration

Channels Configuration

Channel Type *
TCP Channel

Master DNP Address *
1

Slave DNP Address *
2

Connection Configuration

Channel Name *
Channel_1772611137255

Min Retry Delay (ms) *
1000

Max Retry Delay (ms) *
60000

Reconnection Delay *
0

Connection Retries Number *
0

Response Timeout (ms) *
10000

Keep Alive Timeout (ms) *
60000

Address Configuration

IP Address *
127.0.0.1

Port Number *
20000

Advanced Configuration

Master Configuration

Task Response Timeout (ms) *
5000

Task Retry Period (ms) *
5000

Task Start Timeout (ms) *
10000

Time Synchronization Mode *
Default

Control Qualifier Mode *
Allow Two Byte

☒ Perform Integrity Scan on Startup

Class Configuration

Class	Scan Rate (ms)	Unsolicited Event
Class1	5000	<input checked="" type="checkbox"/>
Class2	5000	<input checked="" type="checkbox"/>
Class3	5000	<input checked="" type="checkbox"/>

Integrity Poll Scan Rate (ms)
3600000

☐ Integrity on Event Overflow IIN

☒ Disable Unsolicited Messages During Startup

Figure 24: DNP3 Device Configuration View

(!) Note

To use the DNP3 Protocol, you need to download and install **OpenSSL for Windows 1.1.1v x64**.

Installer name: Win64OpenSSL-1_1_1v

Parameter	Description	Default Value
Channels Configuration		
Channel Type	Type of the DNP communication channel: <ul style="list-style-type: none"> TCP Channel. Serial Channel. TLS Channel. 	TCP Channel
Master DNP Address	Unique master address for communication with slave devices. Range: 0 - 65519.	1
Slave DNP Address	Unique slave address. Range: 0 - 65519.	2
Connection Configuration		
Channel Name	Name assigned to the channel.	Channel_xx
Min Retry Delay (ms)	Minimum retry interval after failure.	1000
Max Retry Delay (ms)	Maximum retry interval after failure.	60000
Reconnection Delay	Time before attempting to reconnect.	0
Baud	Serial Channel only. Communication speed in bits per second.	9600

<i>Data Bits</i>	Serial Channel only. Number of data bits per character.	8
<i>Stop Bits</i>	Serial Channel only. Number of stop bits used.	1
<i>Parity</i>	Serial Channel only. Available options are: <ul style="list-style-type: none"> • None. • Even. • Odd. 	None
<i>Flow Control</i>	Serial Channel only. Available options are: <ul style="list-style-type: none"> • None. • Hardware. • XONXOFF. 	None
<i>Peer Certificate File Path</i>	TLS Channel only. Path to the peer certificate file.	
<i>Local Certificate File path</i>	TLS Channel only. Path to the local certificate file.	
<i>Private Key File Path</i>	TLS Channel only. Path to the private key file.	
<i>Max Verify Path</i>	TLS Channel only. Maximum verification path depth.	

Connection Retries Number	Number of link-layer retries if link-layer confirmation is enabled.	0
Response Timeout (ms)	Maximum time to retransmit an unsolicited response without confirmation from the master.	10000
Keep Alive Timeout (ms)	Time without a message before the link-layer sends a keep-alive request.	60000
Address Configuration		
IP Address	IP address of the DNP3 device.	
Port Number	TCP port used for Ethernet communication.	20000
Advanced Configuration		
Master Configuration		
Task Response Timeout (ms)	Timeout for task response.	5000
Task Retry Period (ms)	Interval between retry attempts for a task.	5000
Task Start Timeout (ms)	Timeout for starting a task.	10000

<i>Time Synchronization Mode</i>	<p>Determines how the outstation requests time synchronization from the master. Available options are:</p> <ul style="list-style-type: none"> • Default. • UTC. • Current Host Time. 	Default
<i>Control Qualifier Mode</i>	<p>Qualifier mode for master requests. Available options are:</p> <ul style="list-style-type: none"> • Allow One Byte • Allow Two Byte 	Allow Two Byte
<i>Perform Integrity Scan on Startup</i>	<p>Run integrity polls automatically when the DNP connector restarts.</p>	Checked
<i>Class Configuration</i>		
<i>Class</i>	<p>Event data classification:</p> <ul style="list-style-type: none"> • Class 1 (Critical Events): Alarms, critical system changes, high-priority events that require immediate attention. • Class 2 (Important Events): Operational changes, medium-priority events that are important but not urgent. 	

	<ul style="list-style-type: none"> • Class 3 (Low-Priority Events): Non-critical, low-priority events that are retrieved less frequently. 	
<i>Scan Rate (ms)</i>	Interval between event polls to detect event data changes.	5000
<i>Unsolicited Event</i>	Allow the outstation to send unsolicited messages for Class 1-3 data without requiring a polling request from the master station.	Checked
<i>Integrity Poll Scan Rate (ms)</i>	Interval for full data retrieval polls from the DNP slave device.	3600000
<i>Integrity on Event Overflow IIN</i>	Triggers integrity polls if event buffer overflow is indicated by the DNP server.	Unchecked
<i>Disable Unsolicited Messages During Startup</i>	Disables unsolicited messages during startup if enabled.	Checked

Table 18: DNP3 Device Configuration Parameters

3.1.7. FTP Server

Select **FTP** from the **Type** drop-down list, then click **Next**.

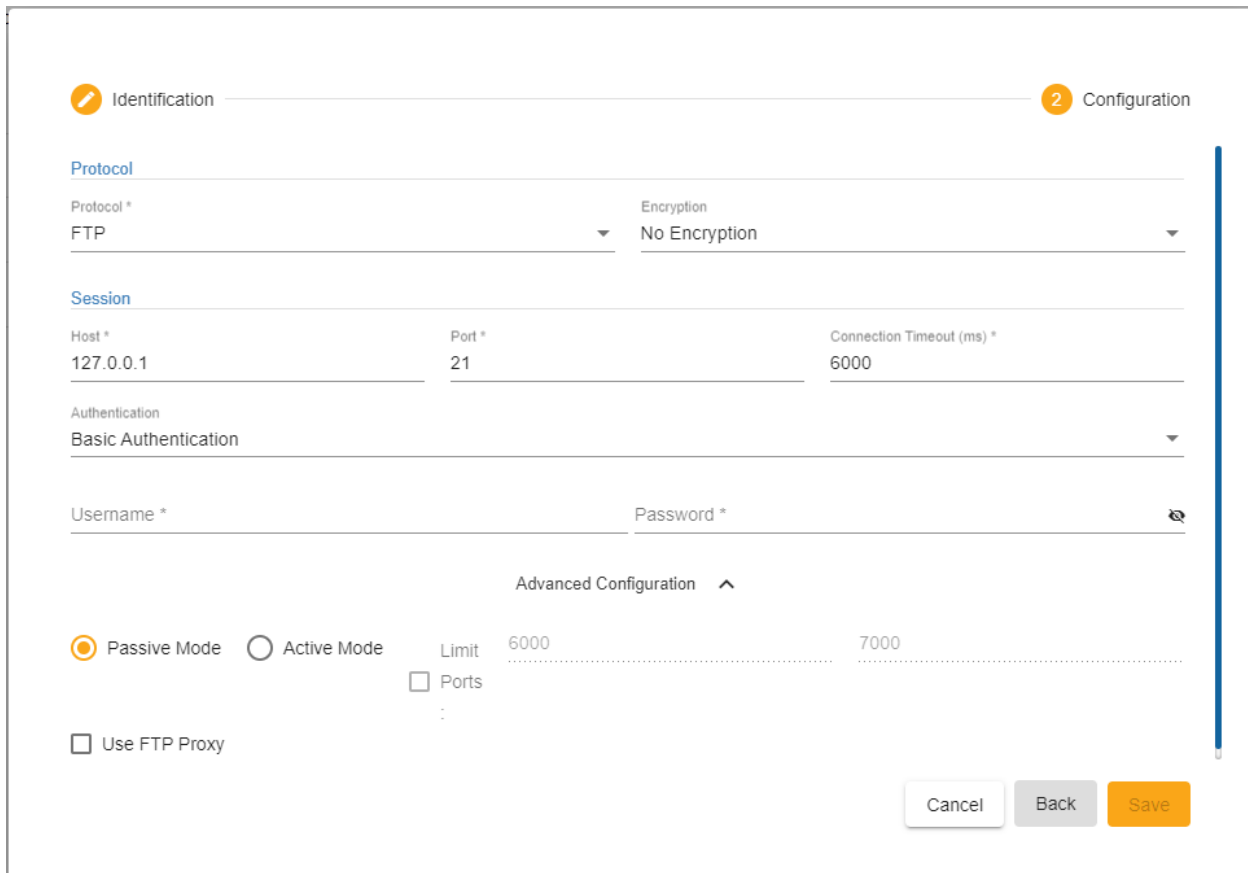


Figure 25: FTP Server Device Configuration View

Parameter	Description	Default Value
Protocol	Choose the protocol for the file transfer: <ul style="list-style-type: none"> FTP (File Transfer Protocol) SFTP (SSH File Transfer Protocol) 	FTP
Encryption	Define the level of Encryption to be used: <ul style="list-style-type: none"> No Encryption 	No Encryption

	<ul style="list-style-type: none"> • TLS/SSL Implicit Encryption • TLS/SSL Explicit Encryption 	
CA Certificate	Path to the Certificate Authority file for SSL/TLS.	
SSL Protocol	Supported SSL/TLS versions: <ul style="list-style-type: none"> • TLS • TLS 1.1 • TLS 1.2 	TLS
Host	IP address of the FTP Server.	127.0.0.1
Port	FTP typically uses port 21 for control commands and port 20 for data transfer. Additional ports may be required for active mode FTP.	21
Connection Timeout (ms)	Time in milliseconds the server waits before closing the current request and initiating a new one if it fails.	6000
Authentication	Select authentication type: <ul style="list-style-type: none"> • Basic Authentication: provide username and password. • Anonymous: access without authentication. 	Basic Authentication
Username	Username used to connect to the FTP server.	
Password	Password associated with the username and used to connect to the FTP server.	
Advanced Configuration		
Passive Mode	The server establishes data connection and provides the client with an IP and port.	Checked

Active Mode	The client specifies a port, and the server connects back to establish the data channel.	Unchecked
Limit Ports	Define a specific range of ports for data connections in Active Mode.	6000 - 7000
Use FTP Proxy	Enable FTP proxy with Passive Mode.	Unchecked
Proxy Host	Hostname of the FTP proxy server.	
Proxy Port	Port of the FTP proxy server.	8080
Proxy Username	Proxy authentication username.	
Proxy Password	Proxy authentication password.	

Table 19: FTP Device Configuration Parameters

3.1.8. HARTIP

Select **HARTIP** from the **Type** drop-down list, then click **Next**.

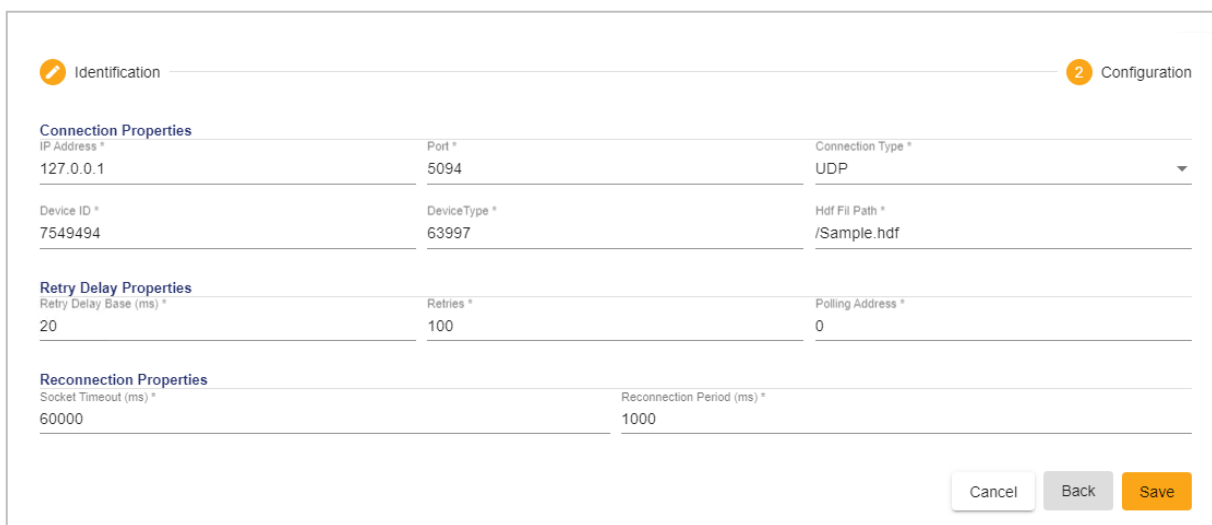


Figure 26: HARTIP Device Configuration View

Parameter	Description	Default Value
Connection Properties		
IP Address	IP address of the target module	127.0.0.1
Port	Port used to initiate a session. Subsequent traffic uses a different port selected by the server.	5094
Connection Type	Select the connection type: <ul style="list-style-type: none"> UDP: Fast, connectionless, no error checking, used for streaming. TCP: Reliable, connection-based, ensures data order and accuracy, used for web and file transfers. 	UDP
Device ID	Unique identifier assigned to each HART-enabled device on the network.	7549494
Device Type	Specifies the category or class of the HART device. (e.g., pressure transmitter, temperature sensor, flow meter).	63997
HDF File Path	Path to the HART Device File (HDF), which contains device capabilities, parameters, and communication details.	/Sample.hdf
Retry Delay Properties		
Retry Delay Base (ms)	Time in milliseconds to wait after a failed execution before retrying.	20
Retries	Number of times the master will resend the query if no valid response is received.	100

Polling Address	HART addressing (single byte, also known as "short address").	0
Reconnection Properties		
Socket Timeout (ms)	Maximum time in milliseconds a socket connection will wait for a response before timing out.	60000
Reconnection Period (ms)	Interval in milliseconds between attempts to reconnect to a device or server after a failure.	1000

Table 20: HARTIP Device Configuration Parameters

3.1.9. IEC 60870-5-104

Select **IEC 60870-5-104** from the **Type** drop-down list, then click **Next**.

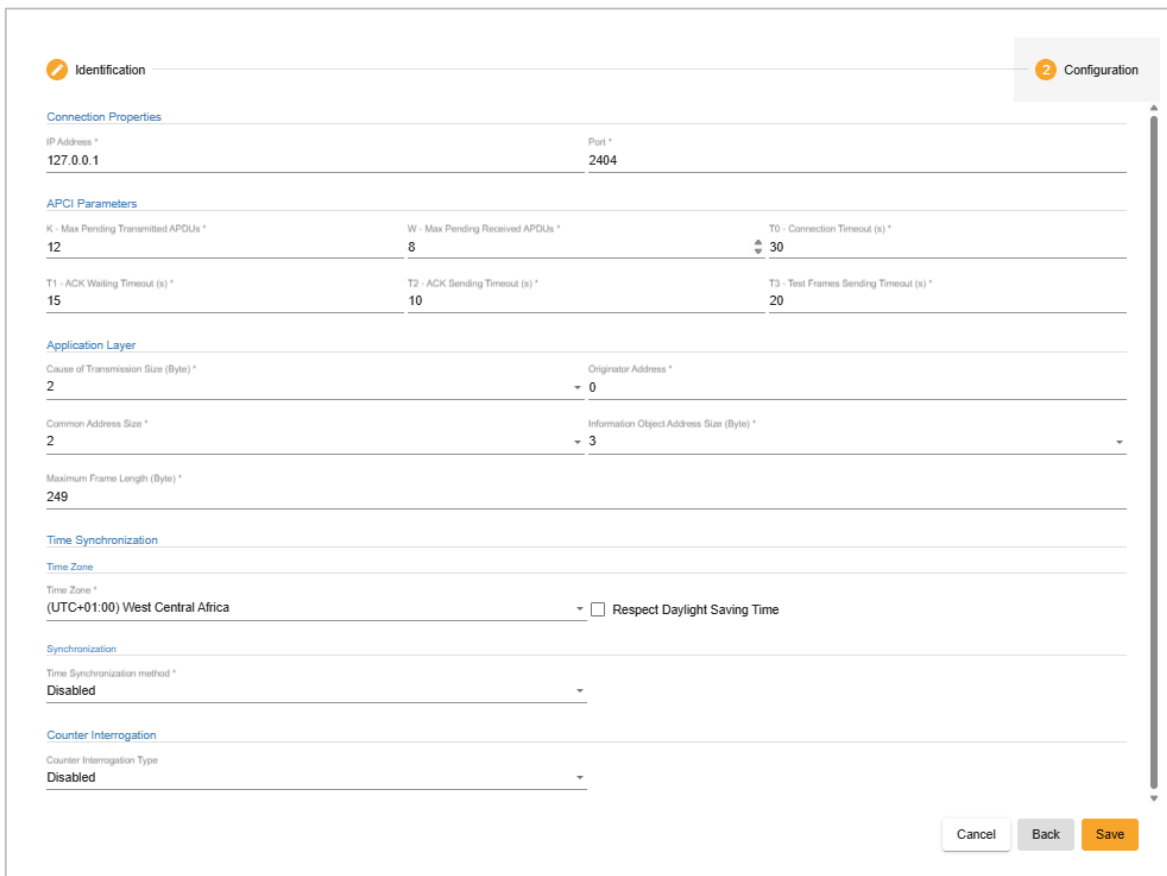


Figure 27: IEC 60870-5-104 Device Configuration View

Parameter	Description	Default Value
<i>Connection Properties</i>		
<i>IP Address</i>	IP address of the device.	127.0.0.1
<i>Port</i>	Port used for message delivery.	2404
<i>APCI Parameters</i>		
<i>K - Max Pending Transmitted APDUs</i>	Maximum number of unconfirmed APDUs the slave outstation can transmit before requiring acknowledgment from the SCADA master.	12
<i>W - Max Pending Received APDUs</i>	Maximum number of APDUs the slave outstation can receive before sending confirmation of receipt.	8
<i>T0 - Connection Timeout (s)</i>	Timeout for establishing a connection.	30
<i>T1 - ACK Waiting Timeout (s)</i>	Maximum time the slave waits for an APDU to be acknowledged by the SCADA master.	15

<i>T2 - ACK Sending Timeout (s)</i>	Maximum idle time after receiving an APDU before sending a response acknowledgment.	10
<i>T3 - Test Frames Sending Timeout (s)</i>	Interval at which the slave sends a test frame to check channel availability during inactivity.	20
<i>Application Layer</i>		
<i>Cause of Transmission Size (byte)</i>	Number of octets in an ASDU COT field. Available options are: <ul style="list-style-type: none"> One Octet: originator address not included. Two Octets: originator address included. 	2
<i>Originator Address</i>	Second byte of the COT field, used by dual-mode devices to route responses to the correct master. Range: 0–254.	0
<i>Common Address Size</i>	Number of octets allowed in a device's common address. Available options are: <ul style="list-style-type: none"> One Octet. Two Octets. 	2
<i>Information Object Address Size (byte)</i>	Number of allowed octets in an information object address (IOA). Available options are: <ul style="list-style-type: none"> One Octet. Two Octets. 	3

	<ul style="list-style-type: none"> Three Octets. 	
Maximum Frame Length (byte)	Maximum supported frame length.	249
Time Synchronization		
Time Zone		
Time Zone	Time zone used for synchronization with the device.	(UTC+01:00) West Central Africa
Respect Daylight Saving Time	Adjust device time ± 1 hour during daylight saving transitions.	Unchecked
Synchronization		
Time Synchronization Method	Method for synchronization: <ul style="list-style-type: none"> Disabled: No synchronization. Absolute: Synchronizes to a fixed time. Interval: Synchronizes on startup and periodically using Sync Interval. 	Unchecked
Sync Absolute	Absolute time value used when Absolute method is selected.	12:00:00 AM

<i>Sync Interval (min)</i>	Interval in minutes between synchronizations when Interval method is selected.	30
<i>Counter Interrogation</i>		
<i>Counter Interrogation Type</i>	<p>Defines how the master requests the current counter value from the slave. Three types are available:</p> <ul style="list-style-type: none"> • Disabled: No counter interrogation. • Initial: applied when the master device requests the current value of a counter from the slave device for the first time or after a reset. • Cyclic: involves periodic requests by the master to retrieve updated counter values from the slave at regular intervals. It ensures that the master continuously monitors the counter's state over time. 	Unchecked
<i>Counter Interrogation Period (min)</i>	Interval in minutes for cyclic counter interrogation, (used when type = Cyclic).	30

Table 21: IEC 60870-5-104 Device Configuration Parameters

3.1.10. J1939

Select **J1939** from the **Type** drop-down list, then click **Next**.



The screenshot shows a configuration window with two tabs: 'Identification' (active) and 'Configuration'. Under 'Identification', there are two fields: 'Connection Timeout (ms) *' with a value of '1000' and 'Can Interface *' with a value of 'J1939'. At the bottom right, there are three buttons: 'Cancel', 'Back', and 'Save'.

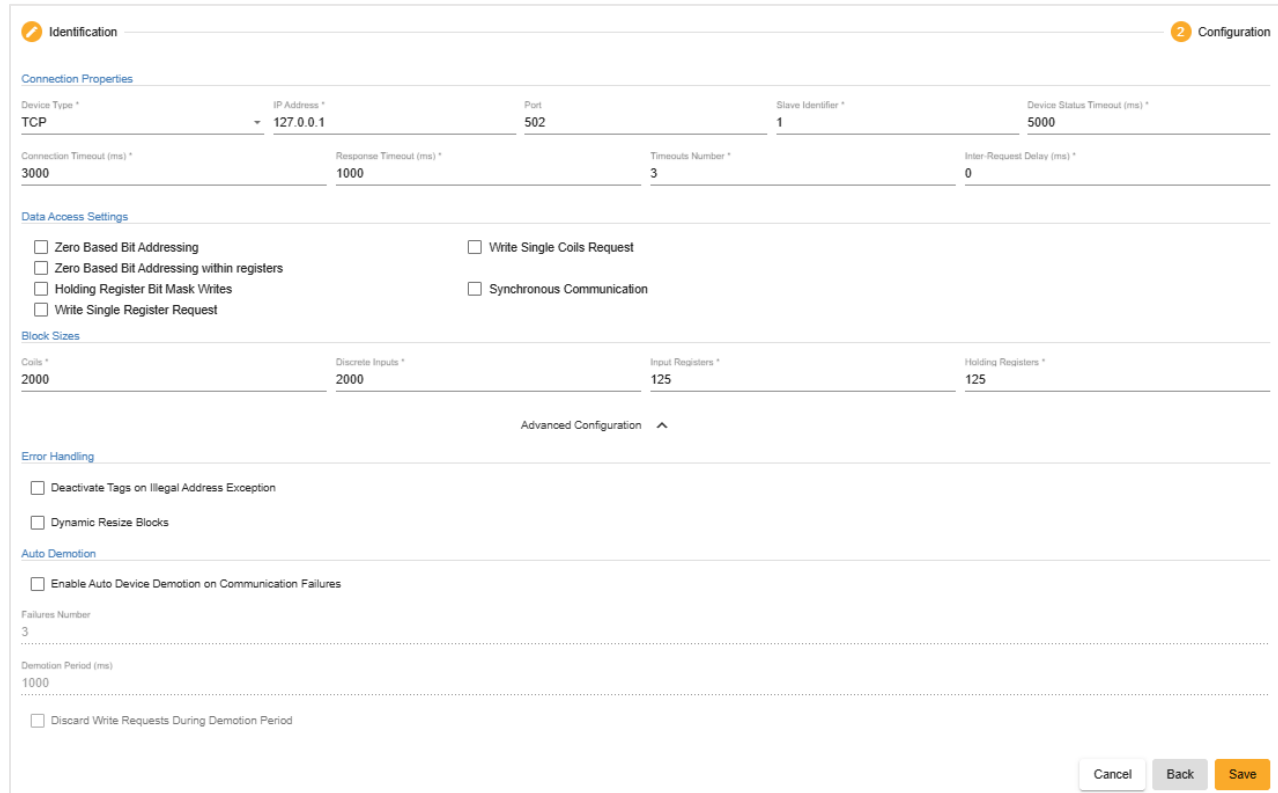
Figure 28: J1939 Device Configuration View

Parameter	Description	Default Value
Connection Timeout (ms)	Maximum time to wait for a response from the device.	1000
Can Interface	The CANbus interface uses an asynchronous transmission scheme controlled by start and stop bits at the beginning and end of each character.	J1939

Table 22: J1939 Device Configuration Parameters

3.1.11. Modbus

Select **Modbus** from the **Type** drop-down list, then click **Next**.



The screenshot shows the 'Configuration' tab of the Modbus device configuration interface. It includes sections for Connection Properties, Data Access Settings, Block Sizes, Error Handling, and Auto Demotion. The 'Device Type' is set to 'TCP', and the 'IP Address' is '127.0.0.1'. The 'Port' is '502', and the 'Slave Identifier' is '1'. The 'Device Status Timeout (ms)' is '5000'. The 'Connection Timeout (ms)' is '3000', 'Response Timeout (ms)' is '1000', 'Timeouts Number' is '3', and 'Inter-Request Delay (ms)' is '0'. The 'Data Access Settings' section has checkboxes for 'Zero Based Bit Addressing', 'Zero Based Bit Addressing within registers', 'Holding Register Bit Mask Writes', 'Write Single Register Request', 'Write Single Coils Request', and 'Synchronous Communication'. The 'Block Sizes' section has 'Coils' set to '2000', 'Discrete Inputs' set to '2000', 'Input Registers' set to '125', and 'Holding Registers' set to '125'. The 'Error Handling' section has checkboxes for 'Deactivate Tags on Illegal Address Exception' and 'Dynamic Resize Blocks'. The 'Auto Demotion' section has a checkbox for 'Enable Auto Device Demotion on Communication Failures'. The 'Failures Number' is '3', and the 'Demotion Period (ms)' is '1000'. There is a checkbox for 'Discard Write Requests During Demotion Period'. At the bottom right, there are 'Cancel', 'Back', and 'Save' buttons.

Figure 29: Modbus Device Configuration View

Parameter	Description	Default Value
Connection Properties		
Device Type	<p>Defines the communication interface used for Modbus connections. Available options are:</p> <ul style="list-style-type: none"> TCP: Modbus TCP/IP, based on Modbus RTU over Ethernet. SERIAL: Communication over RS-485, RS-232, or similar serial interfaces. 	TCP

<i>Slave Identifier</i>	One-byte slave identifier included in each message. Valid range: 1–255. Identifier 0 is reserved for broadcast to all slaves.	1
<i>Connection Timeout (ms)</i>	Time interval before the server is considered unresponsive.	3000
<i>Response Timeout (ms)</i>	Time the device waits for a response after sending a read request.	1000
<i>Timeouts Number</i>	Maximum number of allowed timeouts when the server does not respond.	3
<i>Device Type = TCP</i>		
<i>IP Address</i>	IP address of the Modbus TCP device.	127.0.0.1
<i>Port</i>	TCP port used for Modbus communication.	502
<i>Device Status Timeout (ms)</i>	Interval at which the connector checks the device communication status.	5000
<i>Inter-Request Delay (ms)</i>	Delay between successive requests sent by the master to the slave.	0
<i>Device Type = SERIAL</i>		
<i>COM Port</i>	Serial COM port number.	1
<i>Transmission Mode</i>	Defines how message bytes are encoded and decoded on the serial line. Supported modes are ASCII and RTU.	RTU
<i>Baud Rate</i>	Speed of the serial communication line.	9600
<i>Data Bit</i>	Number of data bits in each transmitted character.	8

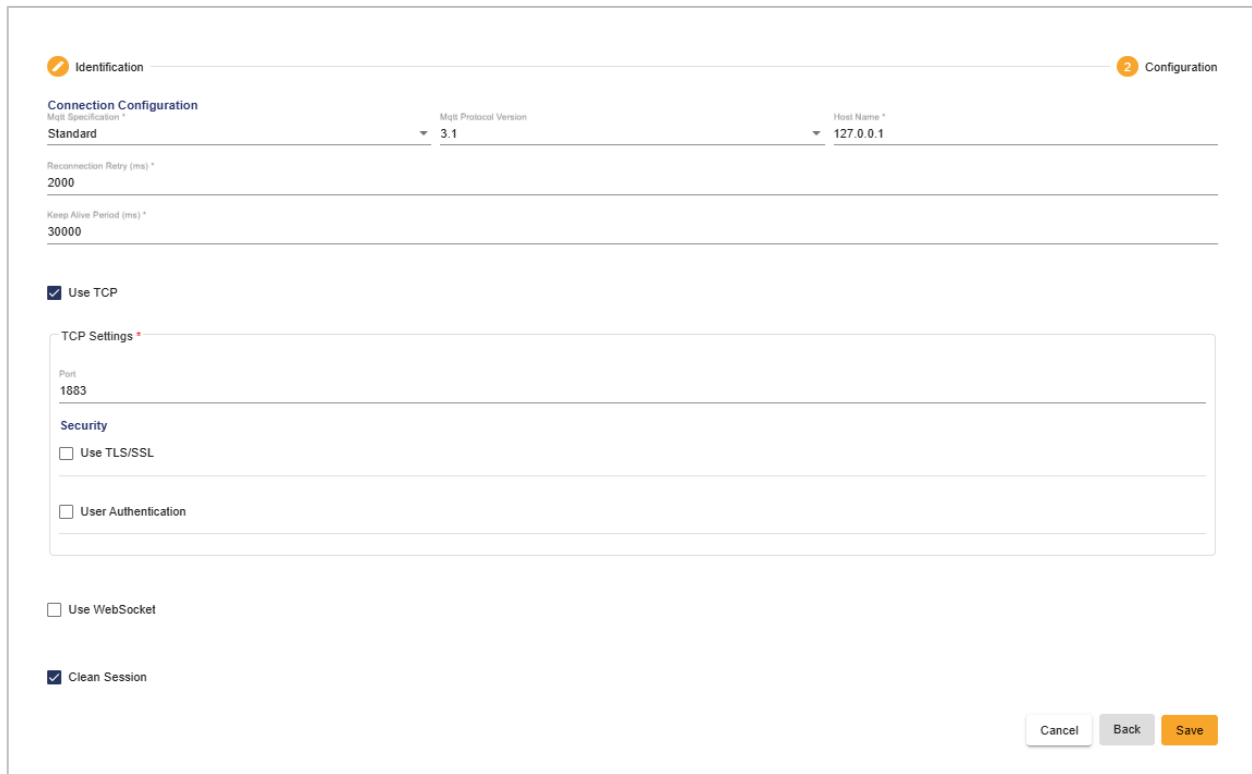
Parity	Parity bit configuration. Options: Odd, Even, None.	Even
Stop Bits	Number of stop bits transmitted between two bytes. Options: 1 or 2.	1
Data Access Settings		
Zero based bit addressing	Enables zero-based addressing when the device uses an address convention starting at one.	Unchecked
Zero based bit addressing within registers	Allows individual bits within a register to be addressed using zero-based or one-based indexing.	Unchecked
Holding register bit mask writes	Updates only the targeted bit when writing to a holding register, without affecting other bits.	Unchecked
Write multiple registers request	Enables writing a contiguous block of holding registers in a single request.	Unchecked
Write multiple coils request	Enables writing a contiguous block of coils in a single request.	Unchecked
Synchronous communication	Ensures requests and responses are processed sequentially.	Unchecked
Block Sizes		
Coils	Output coils block size per Modbus request. Range: 1-2000 bits.	2000
Discrete Inputs	Input coils block size per Modbus request. Range: 1-2000 bits.	2000

<i>Input Registers</i>	Input register block size per Modbus request. Range: 1-125 registers.	125
<i>Holding Registers</i>	Holding register block size per Modbus request. Range: 1-125 registers.	125
<i>Advanced Configuration</i>		
<i>Deactivate tags on illegal address exception</i>	Stops polling a data block when an illegal address exception occurs.	Unchecked
<i>Dynamic Resize Blocks</i>	Automatically resizes blocks when an illegal data address is detected.	Unchecked
<i>Enable auto device demotion on communication failures</i>	Temporarily demotes the device when communication failures reach a configured limit.	Unchecked
<i>Failure number</i>	Number of consecutive failures required to trigger device demotion.	3
<i>Demotion Period (ms)</i>	Duration during which no read requests are sent to the device.	1000
<i>Discard write requests during the demotion period</i>	Prevents the write requests from being sent while the device is demoted.	Unchecked

Table 23: Modbus Device Configuration Parameters

3.1.12. MQTT

Select **MQTT** from the **Type** drop-down list, then click **Next**.



The screenshot shows the 'MQTT Device Configuration View' interface. It has two tabs: 'Identification' (active) and 'Configuration'. Under 'Identification', there's a 'Connection Configuration' section with a 'Mqtt Specification' dropdown set to 'Standard', a 'Mqtt Protocol Version' dropdown set to '3.1', and a 'Host Name' field set to '127.0.0.1'. Below these are 'Reconnection Retry (ms)' set to '2000' and 'Keep Alive Period (ms)' set to '30000'. There's a checkbox for 'Use TCP' which is checked. Below it is a 'TCP Settings' section with a 'Port' field set to '1883'. Under 'Security', there are checkboxes for 'Use TLS/SSL' and 'User Authentication', both of which are unchecked. At the bottom, there's a checkbox for 'Use WebSocket' (unchecked) and a checkbox for 'Clean Session' (checked). At the bottom right, there are 'Cancel', 'Back', and 'Save' buttons.

Figure 30: MQTT Device Configuration View

Parameter	Description	Default Value
MQTT Specification	<p>Defines the MQTT usage mode:</p> <ul style="list-style-type: none"> Standard: Standard MQTT communication. Sparkplug: Enables stateful messaging using device lifecycle management (e.g., birth and last will & testament messages) to ensure data validity and integrity. 	Standard

<i>MQTT Protocol Version</i>	Supported MQTT protocol version: <ul style="list-style-type: none">• Standard: 3.1, 3.1.1• Sparkplug: 3.1.1	3.1
<i>Host Name</i>	Hostname or IP address of the MQTT broker.	127.0.0.1
<i>Reconnection Retry (ms)</i>	Delay (in milliseconds) before retrying a connection when the connector loses communication with the broker.	2000
<i>Keep Alive Period (ms)</i>	Interval (in milliseconds) for keep-alive messages used to verify the client-broker connection.	30000
<i>MQTT over TCP</i>		
<i>Use TCP</i>	Enables MQTT communication over TCP.	Checked
<i>Port</i>	MQTT broker port (1883 for standard TCP, 8883 for TLS).	1883
<i>MQTT over TCP - TLS/SSL</i>		
<i>Use TLS/SSL</i>	Enables secure communication with the MQTT broker using TLS/SSL.	Unchecked
<i>Protocol</i>	Supported TLS/SSL versions: <ul style="list-style-type: none">• TLS 1.0• TLS 1.1• TLS 1.2	

CA Certificate Path	Path to the Certificate Authority (CA) certificate.	
Client Certificate Path	Path to the client digital certificate.	
Certificate Password	Password associated with the client certificate.	
User Authentication (TCP)		
Use Authentication	Enables authentication with the MQTT broker using a username and password.	Unchecked
Username	Username used for authentication.	
Password	Password associated with the username.	
MQTT over WebSocket		
Use WebSocket	Enables MQTT communication over WebSocket.	Unchecked
WebSocket Port	Broker port for WebSocket connections (commonly 9001).	9001
Path	Path used to connect to the MQTT broker over WebSocket.	Path
MQTT over WebSocket - TLS/SSL		

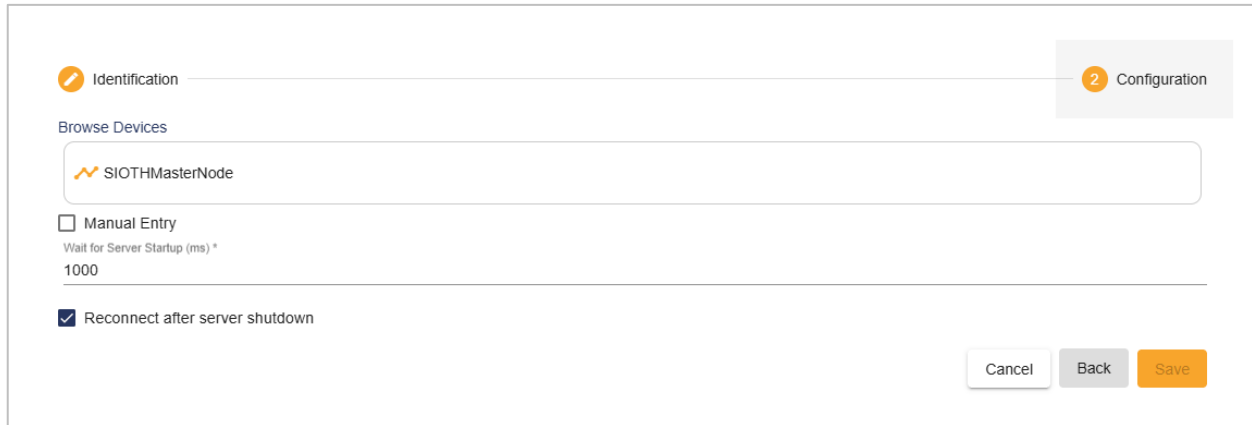
<i>Use TLS/SSL</i>	Enables secure communication with the MQTT broker using TLS/SSL.	Unchecked
<i>Protocol</i>	Supported TLS/SSL versions: <ul style="list-style-type: none"> • TLS 1.0 • TLS 1.1 • TLS 1.2 	
<i>CA Certificate Path</i>	Path to the Certificate Authority (CA) certificate.	
<i>Client Certificate Path</i>	Path to the client digital certificate.	
<i>Certificate Password</i>	Password associated with the client certificate.	
<i>User Authentication</i>		
<i>Use Authentication</i>	Enables authentication with the MQTT broker using a username and password.	Unchecked
<i>Username</i>	Username used by the MQTT connector for authentication.	
<i>Password</i>	Password associated with the username.	
<i>AWS Signature version 4</i>		

<i>AWS Signature version 4</i>	Enables AWS Signature Version 4 authentication for MQTT over WebSocket.	Unchecked
<i>Access Key</i>	Identity Access Management (IAM) or AWS user long-term credential.	
<i>Secret Key</i>	Secret key used to sign requests sent to AWS services.	
<i>AWS Region</i>	AWS region hosting the service.	us-east-1
<i>Service Name</i>	Name of the AWS service.	
<i>Clean Session</i>		
<i>Clean Session</i>	<p>Defines how session state is handled across connections:</p> <ul style="list-style-type: none"> • Checked: Non-persistent session. the broker does not store subscriptions or undelivered messages. • Unchecked: Persistent session. the broker stores subscriptions and undelivered messages. 	Unchecked

Table 24: MQTT Device Configuration Parameters

3.1.13. OPC AE

Select **OPC AE** from the **Type** drop-down list, then click **Next**.



The screenshot shows the 'OPC AE Device Configuration View' interface. It features a progress bar at the top with two steps: '1 Identification' (active) and '2 Configuration'. Below the progress bar, there is a 'Browse Devices' section with a search bar containing 'SIOTHMasterNode'. A checkbox labeled 'Manual Entry' is present, with a sub-label 'Wait for Server Startup (ms) *' and a value of '1000'. A checkbox labeled 'Reconnect after server shutdown' is checked. At the bottom right, there are three buttons: 'Cancel', 'Back', and 'Save'.

Figure 31: OPC AE Device Configuration View

Parameter	Description	Default Value
Browse Devices	Select a node to start browsing for available OPC AE servers. Once detected, choose the desired Prog ID of the OPC AE server to connect to.	
Manual Entry	<p>Enables manual configuration of the parameters required to connect to the OPC AE server. When enabled, the following fields must be specified:</p> <ul style="list-style-type: none"> • Browse from: The node from which browsing will be initiated. Default value: SIOTHMasterNode. • IP Address: IP address of the machine hosting the OPC Classic AE Server. Default value: 127.0.0.1. • OPC Server Name: The Program Identifier (Prog ID) of the OPC Classic AE Server. 	

	Example: IntegrationObjects.AdvancedSimulator.1.	
Wait for Server Startup (ms)	Time in milliseconds the SIOTH OPC Connector waits for the OPC server to start before attempting a connection.	1000
Reconnect after Server Shutdown	If enabled, the connector automatically reconnects when the server issues a shutdown request.	Checked

Table 25: OPC AE Device Configuration Parameters

3.1.14. OPC DA

Select **OPC DA** from the **Type** drop-down list, then click **Next**.



Figure 32: OPC DA Device Configuration View

Parameter	Description	Default Value
Browse Devices	Select a node to start browsing for available OPC DA servers. Once detected, choose the desired Prog ID of the OPC DA server to connect to.	

Manual Entry	<p>Enables manual configuration of the parameters required to connect to the OPC DA server. When enabled, the following fields must be specified:</p> <ul style="list-style-type: none"> • Browse from: The node from which browsing will be initiated. Default value: SIOTHMasterNode. • IP Address: IP address of the machine hosting the OPC Classic DA Server. Default value: 127.0.0.1. • OPC Server Name: The Program Identifier (Prog ID) of the OPC Classic DA Server. Example: IntegrationObjects.AdvancedSimulator.1. 	
Server Separator	Character or string used to separate elements in an address or tag name. This may vary depending on the OPC server software.	.
Wait for Server Startup (ms)	Time in milliseconds the SIOTH OPC Connector waits for the OPC server to start before attempting a connection.	1000
Reconnect after Server Shutdown	If enabled, the connector automatically reconnects when the server issues a shutdown request.	Checked
Redundancy Configuration		
Configure Redundancy	Enables redundancy by configuring additional OPC DA servers.	Unchecked
Failover Mode	Defines the redundancy mode:	Cold

	<ul style="list-style-type: none"> • Cold: Secondary server is activated only if the primary fails. • Hot: Secondary server runs in parallel with the primary. 	
Redundant Device	Select redundant OPC DA device(s) from the list.	
Group ID	Defines the OPC DA device group ID. Devices in the same group (e.g., 1) are considered together. The connector switches to the next group only when all devices in the current group are down.	1
Monitor Server State	Tracks server responsiveness and switches to a redundant server if the primary fails.	Checked
Attempts Number	Number of reconnect attempts before switching to the secondary server. Only available when Monitor Server State is enabled.	3
Delay (ms)	Time in milliseconds between reconnect attempts or before switching to the secondary server after a failure. Only available when Monitor Server State is enabled.	5000
Monitor Watchdog Tag	Enables switchover monitoring using a watchdog tag.	Unchecked
Tag Name	Item ID of the watchdog tag. The tag typically updates continuously at a defined interval to confirm communication health with the OPC DA server. Only available when Monitor Watchdog Tag is enabled.	

Check Period (ms)	Frequency in milliseconds at which the watchdog tag is checked for value updates. Only available when Monitor Watchdog Tag is enabled.	10000
Specific Value	Enables monitoring switchover logic against a specific tag value. When enabled, the value to monitor must be provided. Only available when Monitor Watchdog Tag is enabled.	Unchecked
Enable Publishing from Both Servers	Allows both primary and secondary servers to publish data simultaneously. Available only when Failover Mode is set to Hot.	Unchecked

Table 26: OPC DA Device Configuration Parameters

3.1.15. OPC HDA

Select **OPC HDA** from the **Type** drop-down list, then click **Next**.

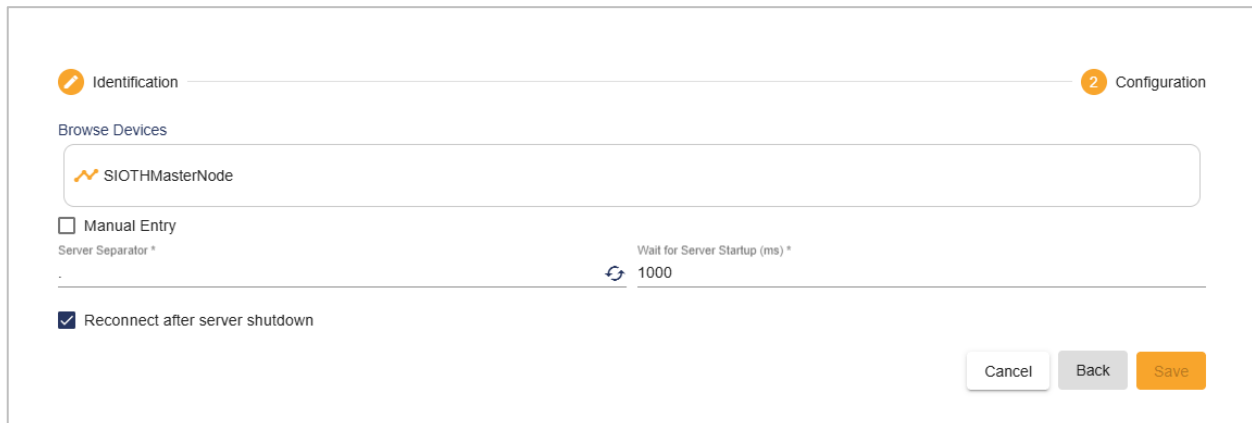


Figure 33: OPC HDA Device Configuration View

Parameter	Description	Default Value
Browse Devices	Select a node to start browsing for available OPC HDA servers. Once detected, choose the desired Prog ID of the OPC HDA server to connect to.	
Manual Entry	<p>Enables manual configuration of the parameters required to connect to the OPC HDA server. When enabled, the following fields must be specified:</p> <ul style="list-style-type: none"> • Browse from: The node from which browsing will be initiated. Default value: SIOTHMasterNode. • IP Address: IP address of the machine hosting the OPC Classic HDA Server. Default value: 127.0.0.1. • OPC Server Name: The Program Identifier (Prog ID) of the OPC Classic HDA Server. Example: IntegrationObjects.AdvancedSimulator.1. 	
Server Separator	Character or string used to separate elements in an address or tag name. This may vary depending on the OPC server software.	.
Wait for Server Startup (ms)	Time in milliseconds the SIOTH OPC Connector waits for the OPC server to start before attempting a connection.	1000
Reconnect after Server Shutdown	If enabled, the connector automatically reconnects when the server issues a shutdown request.	Checked

Table 27: OPC HDA Device Configuration Parameters

3.1.16. OPC UA

Select **OPC UA** from the **Type** drop-down list, then click **Next**.

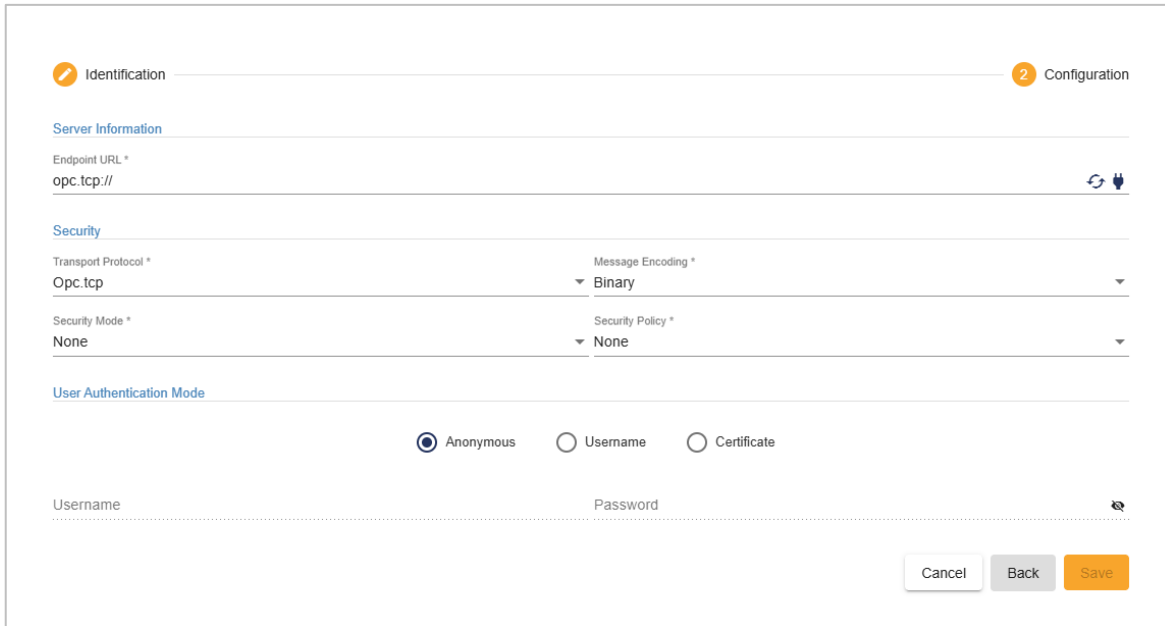




Figure 34: OPC UA Device Configuration View

Parameter	Description	Default Value
Server Information		
Endpoint URL	<p>Specifies the network address used by clients to connect to the OPC UA server. OPC UA primarily uses the opc.tcp scheme, in addition to standard http and https URLs.</p> <p>Example:</p> <p>opc.tcp://server-pc:62640/IntegrationObjects/ServerSimulator</p> <p>Options:</p>	

	<ul style="list-style-type: none"> • Browse Local Servers  : Lists OPC UA servers available on the local machine. • Test Connection  : Verifies connectivity to the specified endpoint. 	
Security		
Transport Protocol	<p>Defines the transport layer protocol used to communicate with the OPC UA server. Available options are:</p> <ul style="list-style-type: none"> • Opc.tcp • Https 	Opc.tcp
Message Encoding	<p>Specifies how data types and structures are serialized during communication. Available options are:</p> <ul style="list-style-type: none"> • Binary • Xml 	Binary
Security Mode	<p>Defines the level of security applied to the communication channel. Available options are:</p> <ul style="list-style-type: none"> • None: No security is applied. • Sign: Messages are digitally signed using the Application Instance Certificate's private key. • Sign_Encrypt: Messages are signed and encrypted using the Application Instance Certificate. 	None

<i>Security Policy</i>	<p>Specifies the cryptographic algorithms used for signing and encryption. Available options are:</p> <ul style="list-style-type: none"> • None: No cryptographic protection is applied. • Basic256: Uses 256-bit encryption. • Basic256Sha256: Uses SHA-256 for digital signatures with 256-bit encryption, providing stronger security than Basic256. • Aes128Sha256RsaOaep: Uses AES-128 for encryption, SHA-256 for signatures, and RSA-OAEP for secure key exchange. • Aes256-Sha256-RsaPss: Uses AES-256 for encryption, SHA-256 for signatures, and RSA-PSS for enhanced cryptographic security. 	None
<i>User Authentication Mode</i>		
<i>Anonymous</i>	Allows connection without providing user credentials.	Enabled
<i>Username</i>	<p>Authenticates the user using a username and password. When enabled, the following fields become mandatory:</p> <ul style="list-style-type: none"> • Username: Username used for authentication. • Password: Password associated with the username. 	Disabled

<i>Certificate</i>	<p>Authenticates the user using an X.509 certificate. When enabled, the following fields become mandatory:</p> <ul style="list-style-type: none"> • Certificate (.pfx): Path to the certificate file. • Certification Password: Password associated with the certificate. 	Disabled
---------------------------	---	----------

Table 28: OPC UA Device Configuration Parameters

3.1.17. S7

Select **S7** from the **Type** drop-down list, then click **Next**.



The screenshot shows the 'S7 Device Configuration View' with two tabs: 'Identification' and 'Configuration'. The 'Identification' tab is active, showing the following fields:

- IP Address *: 127.0.0.1
- Port *: 102
- Connection Timeout (ms) *: 2000
- Max Clients Number *: 1
- Network Adapter
- Type *: S7-300
- Rack *: 0
- Slot *: 2
- PDU *: 240
- ☒ Enable Redundancy
- Failover Mode *: Hot
- Redundant Device *
- Monitoring Type *: Monitor Server State
- Attempts Number *: 3
- Delay (ms) *: 10000

At the bottom right, there are three buttons: 'Cancel', 'Back', and 'Save'.

Figure 35: S7 Device Configuration View

Parameter	Description	Default Value
<i>IP Address</i>	IP address of the CPU or external Ethernet communication card.	127.0.0.1
<i>Port</i>	TCP port used to establish a connection with the PLC.	102
<i>Connection Timeout (ms)</i>	Maximum time to wait for a response from the device before the connection attempt is aborted.	2000
<i>Max Clients Number</i>	Maximum number of simultaneous client connections supported by the Siemens S7 CPU.	1
<i>Network Adaptor</i>	Local network interface used by the client to establish the connection.	
<i>Type</i>	Siemens S7 device type. Supported options are: <ul style="list-style-type: none"> • S7-300 • S7-400 • S7-1200 • S7-1500 	S7-300
<i>Rack</i>	Rack (mounting rail) number of the S7 system modules.	0
<i>Slot</i>	Slot number indicating the physical position of the CPU or I/O module. When using expansion racks, this value may represent a combination of rack and slot numbers.	2
<i>PDU</i>	Size of the Protocol Data Unit (PDU) used to exchange commands, requests, and responses between the client and the PLC.	240

Redundancy		
Enable Redundancy	Specifies whether redundancy is enabled for the device.	Unchecked
Failover Mode	Redundancy operating mode. Available options are: <ul style="list-style-type: none"> • Cold: Secondary server is activated only if the primary fails. • Hot: Secondary server runs in parallel with the primary. 	Hot
Redundant Device	Secondary S7 device selected from the list of configured devices.	
Monitoring Type	Method used to monitor the availability of the primary device. Options are: <ul style="list-style-type: none"> • Monitor Server State • Monitor Watchdog Tag 	Monitor Server State
Monitoring Type = Monitor Server State		
Attempts Number	Number of connection attempts to the primary device before triggering failover to the secondary device.	3
Delay (ms)	Time delay between consecutive connection attempts or before switching to the secondary device after a failure is detected.	10000
Monitoring Type = Monitor Watchdog Tag		
Tag Name	Name of the Watchdog Tag used to monitor the operational state of the primary device.	
Address	Address of the Watchdog Tag in the PLC.	

Type	Data type of the Watchdog Tag. Supported types include: Int, String, Real, DInt, DTL, DWord, Byte, Char, LReal, LWord, USInt, SInt, Word, Date, WChar, BOOL, LINT, UINT, UDINT, ULINT, S5TIME, TIME, LTIME, TIME_OF_DAY, LTIME_OF_DAY, DATE_AND_TIME, LDT, WSTRING, TIMER, COUNTER.	
Timeout (ms)	Maximum time to wait for an update from the Watchdog Tag to confirm proper operation.	
Update Rate (ms)	Frequency at which the Watchdog Tag is read from the PLC.	

Table 29: S7 Device Configuration Parameters

Below are the special cases for S7 Rack and Slot values in table format:

	Rack	Slot	Possible Values
S7-300	0	2	Always 0 for Rack and 2 for Slot.
S7-400	Not Fixed	Not Fixed	It depends on the hardware configuration.
S7-1200	0	0	Can be 0 or 1.
S7-1500	0	0	Can be 0 or 1

Table 30: S7 Rack and Slot Parameters

3.1.18. SMS Server

Select **SMS server** from the **Type** drop-down list, then click **Next**.

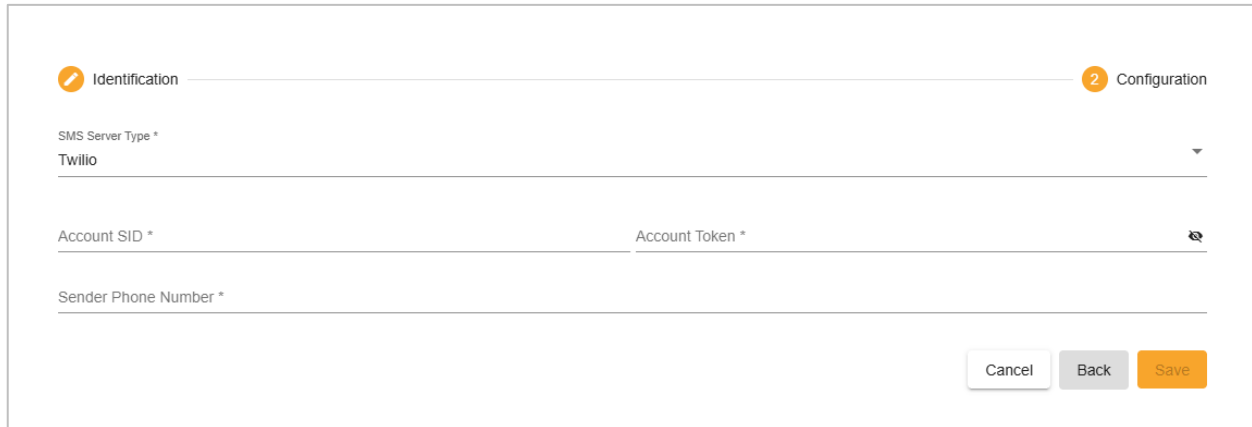


Figure 36: SMS Server Device Configuration View - Twilio

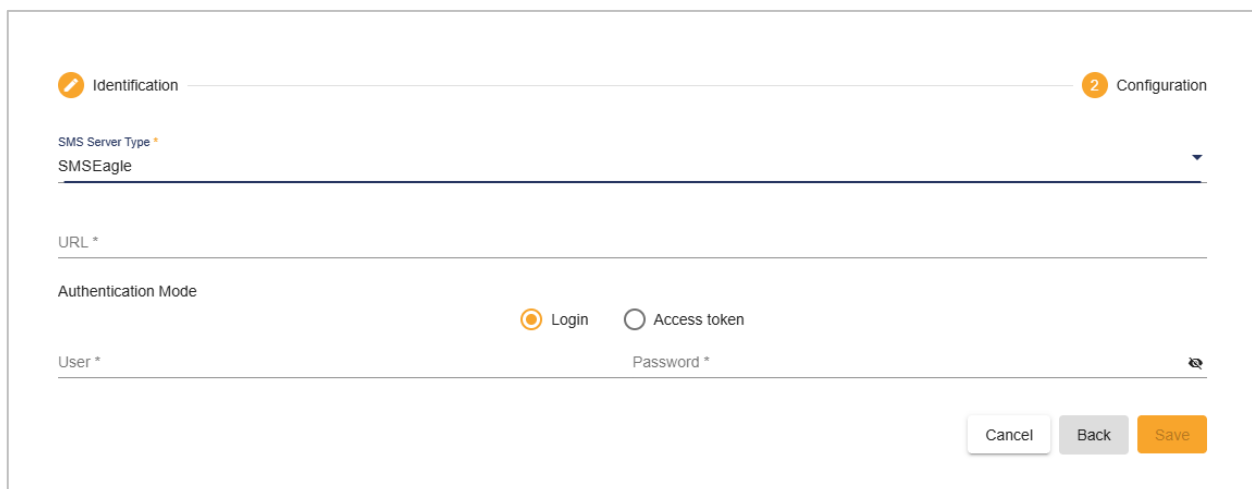


Figure 37: SMS Server Device Configuration View - SMSEagle

Parameter	Description	Default Value
<i>SMS Server Type</i>	<p>Specifies the SMS service provider to be used.</p> <p>Available options are:</p> <ul style="list-style-type: none"> Twilio: Cloud-based SMS service for sending messages via REST API. 	Twilio

	<ul style="list-style-type: none"> • SMSEagle: Hardware SMS gateway providing an HTTP API for message delivery. 	
Twilio SMS Server		
Account SID	Unique identifier used to access Twilio resources. This value is available in the Account Info section of the Twilio console.	
Account Token	Authentication token associated with the Account SID, used as a password. Available in the Account Info section of the Twilio console.	
Sender Phone Number	Phone number used as the originator of outgoing SMS messages. This number is provided by Twilio and listed in the Twilio console.	
SMSEagle Server		
URL	Base URL of the SMSEagle HTTP API used to send SMS messages.	
Login	Authentication method for API v1 using a username and password.	Selected
User	Username used to authenticate with the SMSEagle API.	
Password	Password associated with the configured username.	
Access Token	Alternative authentication method to username/password, using an API access token.	
Token	Token value used when Access Token authentication is selected.	

Table 31: SMS Server Device Configuration Parameters

(!) Note

To send SMS notifications using **Twilio**, you must first create an account at <https://www.twilio.com/>. Once registration is complete, Twilio will provide the required connection parameters: **Account SID**, **Account Token** and **Sender Phone Number**.

To send SMS notifications using **SMSEagle**, access to an SMSEagle device and its HTTP API is required. The connection setup requires configuration parameters such as **Base URL**, **Authentication method** and corresponding **authentication credentials**.

These parameters must be configured in the **SMS Server** device settings within **SIOTH** to enable SMS notification delivery.

3.1.19. SMTP Server

Select **SMTP** from the **Type** drop-down list, then click **Next**.

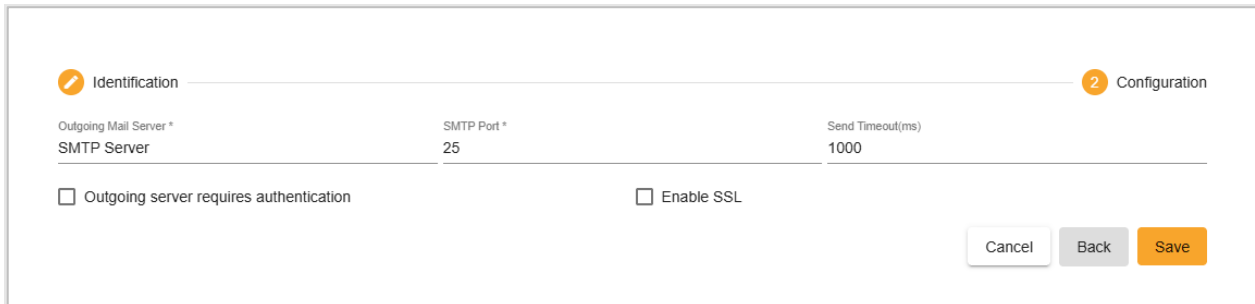


Figure 38: SMTP Server Device Configuration View

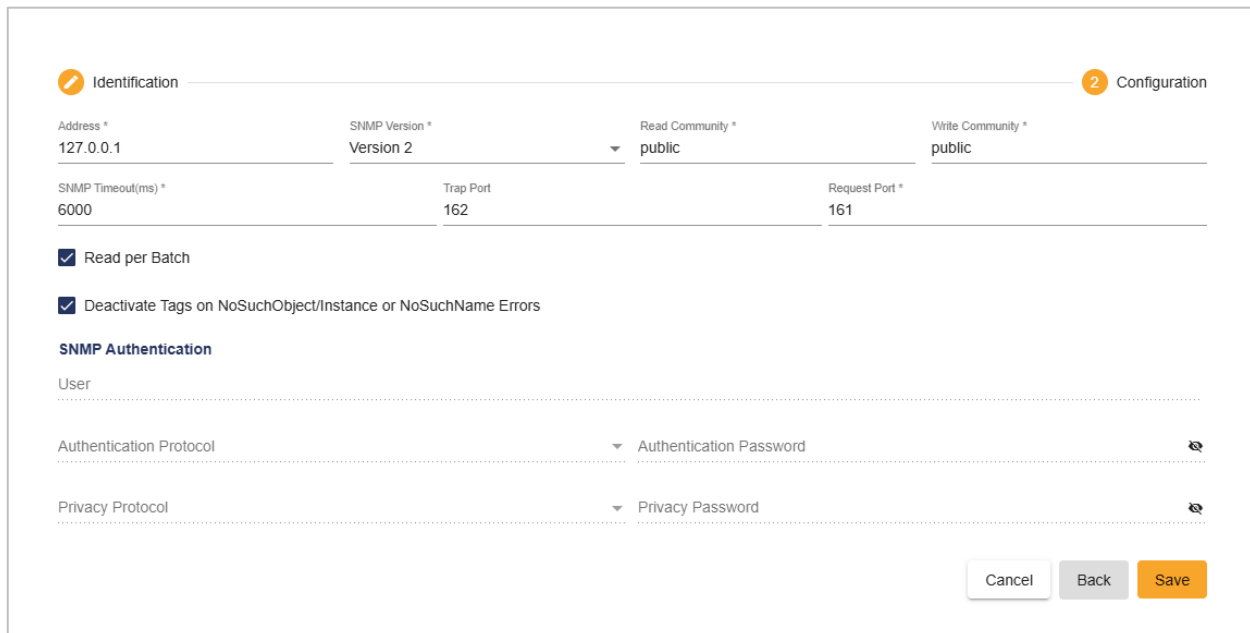
Parameter	Description	Default Value
Outgoing Mail Server	Specifies the SMTP server responsible for handling all outgoing email messages.	SMTP Server
SMTP Port	Defines the port number used by the SMTP server for email transmission.	25

<i>Send Timeout (ms)</i>	Specifies the maximum time, in milliseconds, the server waits before closing a failed request and retrying.	1000
<i>Outgoing Server Required Authentication</i>	Enables authentication for the outgoing SMTP server.	Unchecked
<i>Username</i>	Username used to authenticate with the SMTP server.	
<i>Password</i>	Password associated with the specified username.	
<i>Enable SSL</i>	Specifies whether SSL encryption is used when connecting to the SMTP server.	Unchecked

Table 32: SMTP Server Device Configuration Parameters

3.1.20. SNMP

Select **SNMP** from the **Type** drop-down list, then click **Next**.



The screenshot shows the 'Configuration' tab of the SNMP configuration interface. It includes fields for Address (127.0.0.1), SNMP Version (Version 2), Read Community (public), and Write Community (public). Below these are fields for SNMP Timeout (6000), Trap Port (162), and Request Port (161). There are two checked checkboxes: 'Read per Batch' and 'Deactivate Tags on NoSuchObject/Instance or NoSuchName Errors'. The 'SNMP Authentication' section contains fields for User, Authentication Protocol, Authentication Password, Privacy Protocol, and Privacy Password. At the bottom right are 'Cancel', 'Back', and 'Save' buttons.

Figure 39: SNMP Device Configuration View

Parameter	Description	Default Value
<i>Address</i>	Specifies the IP address of the SNMP-enabled device.	127.0.0.1
<i>SNMP Version</i>	Specifies the SNMP protocol version supported by the device. Available options are: <ul style="list-style-type: none"> Version 1 Version 2 Version 3 	Version 2
<i>Read Community</i>	Defines the community string that provides read-only access to retrieve information from the SNMP device.	public
<i>Write Community</i>	Defines the community string that provides read/write access, allowing modifications to the device configuration.	public
<i>SNMP Timeout (ms)</i>	Specifies the time, in milliseconds, to wait before marking the SNMP device as unresponsive.	6000
<i>Trap Port</i>	Specifies the SNMP port used to receive traps (unsolicited messages) sent by SNMP agents.	162
<i>Request Port</i>	Specifies the port used for sending SNMP requests and receiving responses between the connector and the agent.	161
<i>Read per Batch</i>	Enables batching of multiple SNMP read requests to reduce communication overhead and improve performance.	Checked

Deactivate Tags on NoSuchObject/ Instance or NoSuchName Errors	When enabled, automatically deactivates tags that return NoSuchObject , NoSuchInstance , or NoSuchName errors during polling.	Checked
SNMP Authentication		
User	Specifies the username used for SNMP v3 authentication to identify the entity interacting with the SNMP agent.	
Authentication Protocol	Specifies the authentication protocol. Available options are: <ul style="list-style-type: none"> • MD5 • SHA 	
Authentication Password	Specifies the authentication password required when MD5 or SHA is selected.	
Privacy Protocol	Specifies the encryption protocol. Available options are: <ul style="list-style-type: none"> • DES • AES128 • AES192 • AES256 	

Privacy Password	Specifies the encryption password required when privacy is enabled on the SNMP agent.	
-------------------------	---	--


Table 33: SNMP Device Configuration Parameters

(!) Note

Authentication verifies the identity of entities accessing the SNMP agent. Privacy enables encryption of SNMP v3 messages, ensuring data confidentiality during communication. Privacy protocols provide significantly stronger security compared to the community string-based mechanism used in SNMP v1 and SNMP v2.

3.2. Edit Device

To edit a device, follow these steps:


1. Open the **Devices** page by clicking **Devices** from the left sidebar menu.
2. Locate the device you want to edit.
3. Click the **Edit** icon  in the **Actions** column of the devices' explorer. A pop-up is open, displaying the current configuration of the device.
4. Edit the description and click **Save** to submit the changes.

(!) Note

Once a device has been created, its **name** and **type** cannot be modified.

3.3. Delete Device

To delete a device, follow these steps:



1. Open the **Devices** page by clicking **Devices** from the left sidebar menu.
2. Locate the device you want to delete.
3. Click the **Delete** icon  in the **Actions** column of the devices' explorer.

A confirmation dialog is displayed.

- Click **Yes** to permanently delete the device.
- Click **No** to abort the operation and retain the device.

3.4. Create Multiple Devices

To create multiple devices of the same type, follow these steps:

1. Open the **Devices** page by clicking **Devices** from the left sidebar menu.
2. Click the **Download** icon .
3. Select the target device type by browsing through the displayed drop-down list.
4. Edit the downloaded CSV file with the devices' parameters.
5. Click the **Import** icon .
6. Browse and select the file to import.
7. Click **Open** to load your devices.

4. Projects

In **SIOTH**, each project groups all configuration elements required to build and operate a solution, including data flows, rules, workflows, variables, and events.

The **Projects Explorer** provides a consolidated view of all created projects. It allows to browse existing projects and manage them.

Click **Projects** from the left sidebar menu. An explorer is displayed, listing all the added projects, each one presented on a separate line.

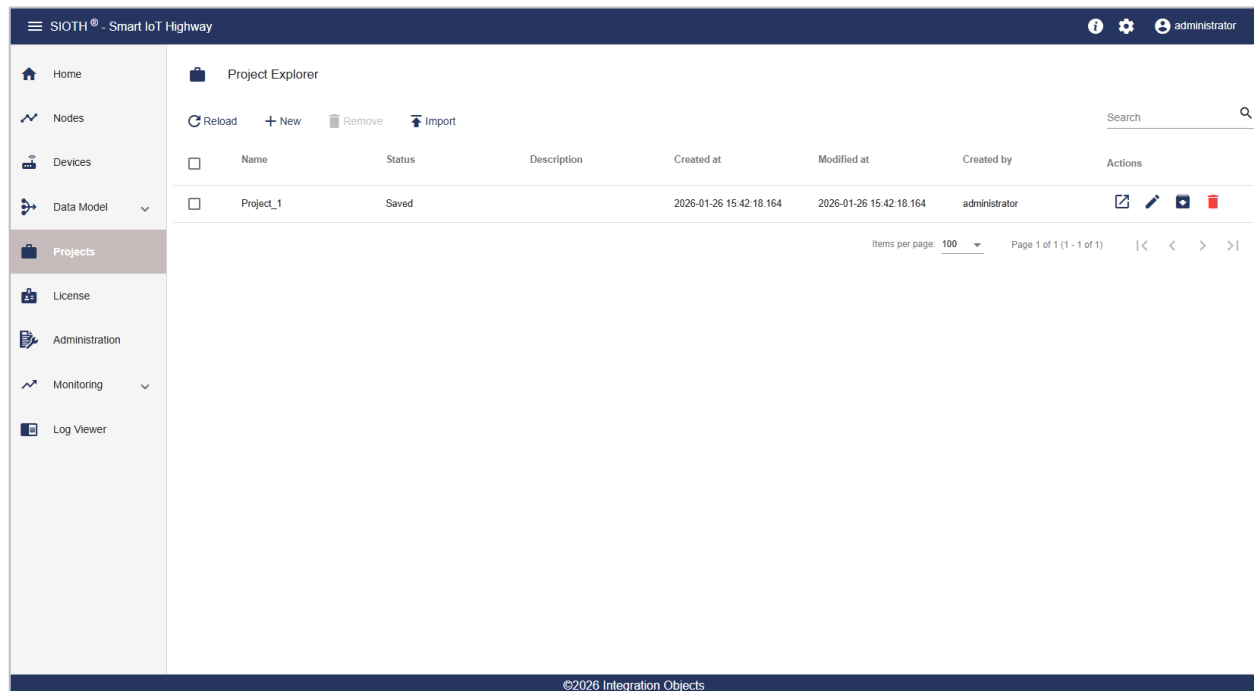






Figure 40: Projects Explorer

For each project, the following actions are available:

-  **Open Project Homepage:** Opens the project workspace, allowing access to and management of project modules such as Data Flows, Rules, Workflows, Variables and Events.
-  **Edit:** Allows modification of the project configuration parameters.
-  **Export:** Exports the selected project configuration for backup, transfer, or reuse purposes.
-  **Delete:** Deletes the project from the projects list.

4.1. Add New Project

Click the **Add Project** card on the Application Configuration home page or click **Projects** in the left-sidebar menu and then click **New**. A pop-up will be displayed, showing the parameters related to the new project.

New Project

Name *
Project_1

Description

Max 500 characters 0/500

☒ Auto Reload

Cancel Save

Figure 41: Add New Project Configuration View


Parameter	Description	Default Value
<i>Name</i>	Unique identifier for the project. The name is generated automatically but can be updated to a user-friendly value.	Project_1
<i>Description</i>	Optional field used to describe the project.	
<i>Auto Reload</i>	Automatically starts project rules and workflows after a machine restart.	Checked

Table 34: Project Configuration Parameters

Click **Save** to apply the changes and add the project to the list of projects.

4.2. Open Project

To open a project, follow these steps:

1. Open the **Projects** page by clicking **Projects** from the left sidebar menu.
2. Locate the project you want to open.
3. Click the **Open** icon  in the **Actions** column of the Projects' explorer.

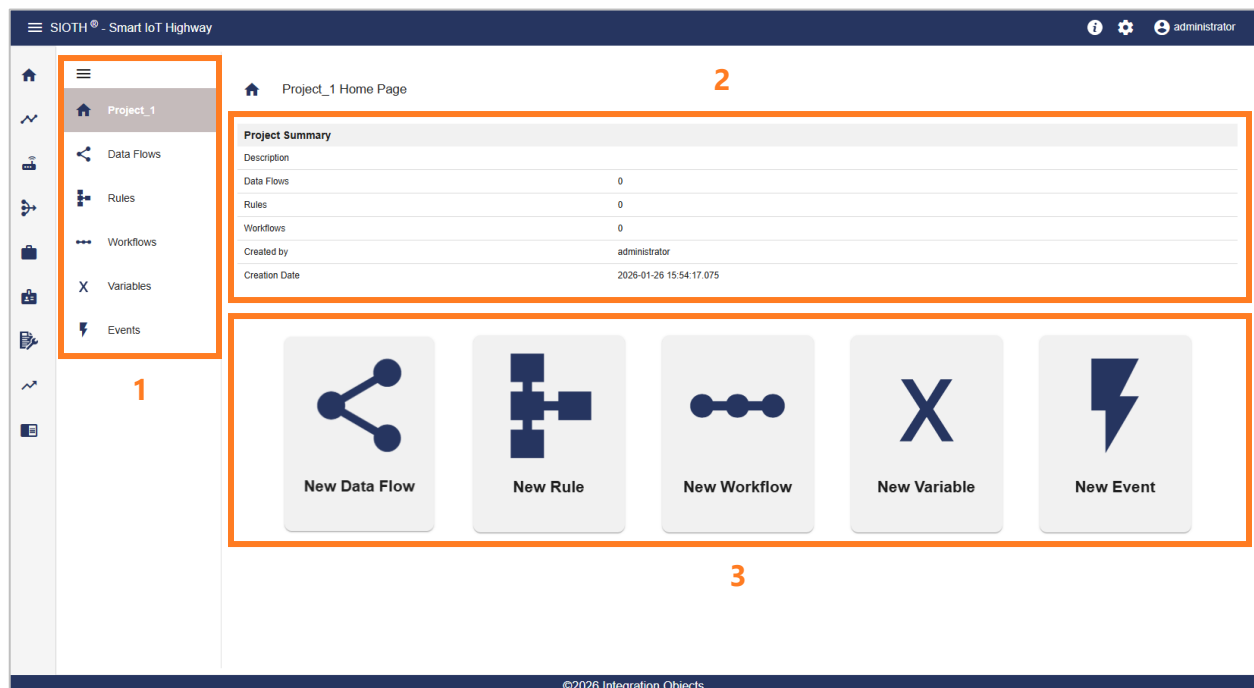



Table 35: Project Homepage View

The project homepage is divided into three main sections:

- **Menu (1):** Provides access to Data Flows, Rules, Workflows, Variables, and Events.
- **Project Summary (2):** Displays a summary of the project, including the number of configured Data Flows, Rules, and Workflows, as well as the project creator and creation date.
- **Shortcuts (3):** Provides quick access for creating new Data Flows, Rules, Workflows, Variables, and Events.


4.3. Edit Project

To update the project configuration, follow these steps:

1. Open the **Projects** page by clicking **Projects** from the left sidebar menu.
2. Locate the project you want to edit.
3. Click the **Edit** icon  in the **Actions** column of the projects' explorer. A pop-up is open, displaying the current configuration of the project.
4. Edit the project parameters and click **Save** to submit the changes.


4.4. Export Project

To export the project configuration, follow these steps:

1. Open the **Projects** page by clicking **Projects** from the left sidebar menu.
2. Locate the project you want to export.
3. Click the **Export** icon  in the **Actions** column of the project' explorer.


4.5. Import Project

To export the project configuration, follow these steps:

1. Open the **Projects** page by clicking **Projects** from the left sidebar menu.
2. Click the **Import** button  **Import** in the **top menu** of the project' explorer.
3. Select a .zip file to be imported.

4.6. Delete Project

To delete a project, follow these steps:


1. Open the **Projects** page by clicking **Projects** from the left sidebar menu.
2. Locate the project you want to delete.
3. Click the **Delete** icon  in the **Actions** column of the projects' explorer.

A confirmation dialog is displayed.

- Click **Yes** to permanently delete the project.
- Click **No** to abort the operation and retain the project.

4.7. Reload Project

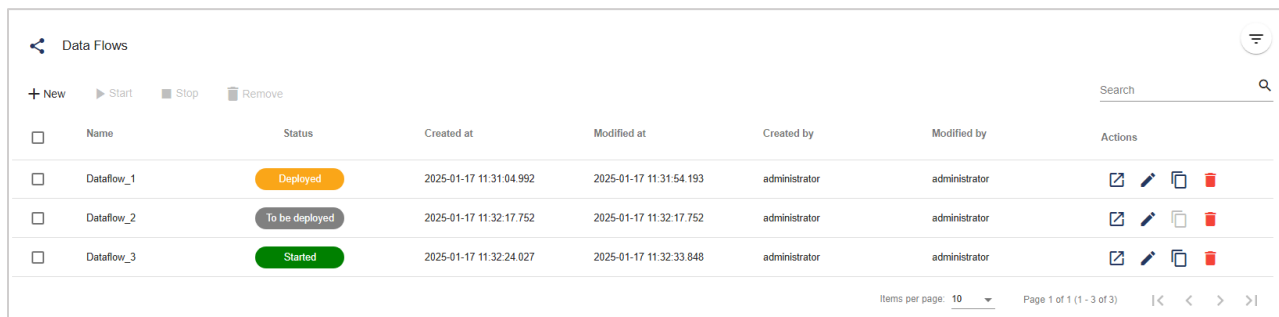
To reload the projects list, follow these steps:

1. Open the **Projects** page by clicking **Projects** from the left sidebar menu.
2. Click the **Reload** button  in the **top menu** of the project' explorer.

5. Data Flows

The **Data Flows** module in the SIOTH platform allows to configure and visually represent the data flow between sources and destinations using an intuitive graphical editor. This module simplifies the design, configuration, and management of data transfer logic across connected devices.

To access the data flows page, click **Projects** from the left sidebar menu and then click **Data Flows**. An explorer is displayed, listing all the added data flows, each one presented on a separate line.












	Name	Status	Created at	Modified at	Created by	Modified by	Actions
<input type="checkbox"/>	Dataflow_1	Deployed	2025-01-17 11:31:04.992	2025-01-17 11:31:54.193	administrator	administrator	  
<input type="checkbox"/>	Dataflow_2	To be deployed	2025-01-17 11:32:17.752	2025-01-17 11:32:17.752	administrator	administrator	  
<input type="checkbox"/>	Dataflow_3	Started	2025-01-17 11:32:24.027	2025-01-17 11:32:33.848	administrator	administrator	  

Figure 42: Data Flows Explorer

The following actions are available for each data flow:



Open Data Flow Editor: Opens the graphical editor to view and modify the data flow configuration.



Edit: Allows modification of the data flow parameters.



Duplicate: Creates one or more copies of the selected data flow, including all configured connectors.



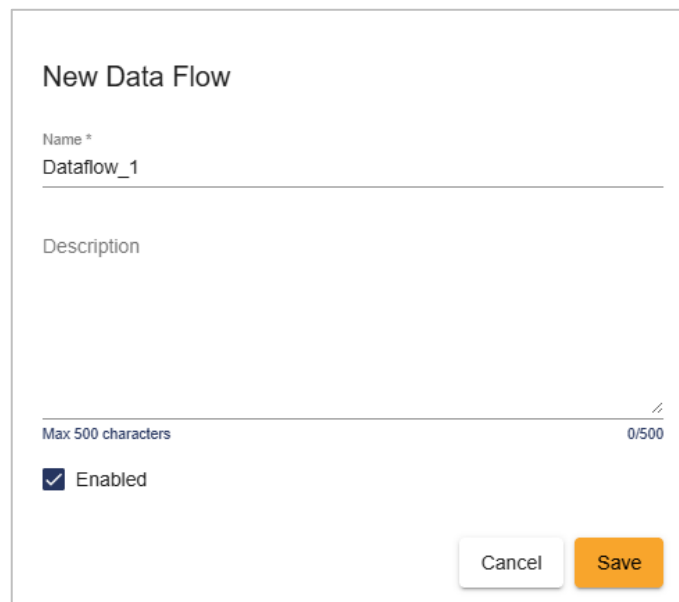
Delete: Deletes the data flow from the list.

5.1. Manage Data Flows

From the **Data Flows** explorer, you can manage the data transfer between the devices that you created by configuring connectors in the data flow editor, editing, duplicating, or deleting created data flows.

5.1.1. Add New Data Flow

Click the **New Data Flow** card on the project home page or click **Data Flows** in the project left-sidebar menu and then click **New**. A pop-up will be displayed, showing the parameters related to the new data flow.



The form titled "New Data Flow" contains the following fields and controls:

- Name ***: A text input field with the value "Dataflow_1".
- Description**: A large text area for entering a description.
- Character Count**: A label "Max 500 characters" and a character count "0/500" with a reset icon.
- Enabled**: A checkbox that is checked, labeled "Enabled".
- Buttons**: "Cancel" and "Save" buttons at the bottom right.

Figure 43: Add New Data Flow Configuration View

Parameter	Description	Default Value
-----------	-------------	---------------


Name	Unique name of the data flow.	Dataflow_<Num>
Description	Optional text describing the purpose of the data flow.	
Enabled	Determines whether the data flow is active. Only enabled data flows can be started using the Start/Stop feature.	Checked

Table 36: Add New Data Flow Configuration Parameters

Click **Save** to submit the changes and add the data flow to the list of created data flows.

5.1.2. Edit Data Flow


To update the data flow configuration, follow these steps:

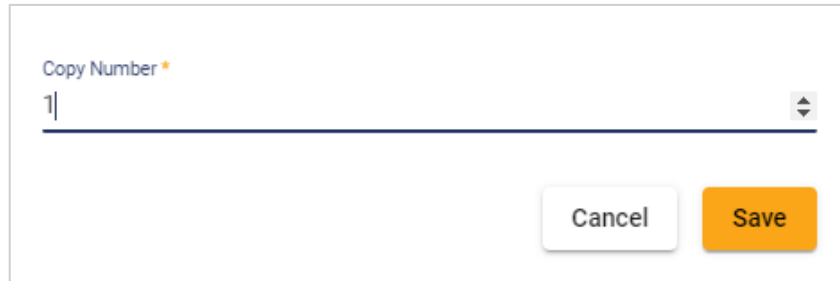
1. Open the **Projects** page by clicking **Projects** from the left sidebar menu.
2. Locate and open the project containing the data flow you want to edit.
3. Open the Data Flows page by clicking **Data Flows** from the project left sidebar menu.
4. Locate the data flow you want to edit.
5. Click the **Edit** icon  in the **Actions** column of the data flows' explorer. A pop-up will appear, displaying the current configuration of the data flow.
6. Edit the data flow parameters and click **Save** to submit the changes.

5.1.3. Duplicate Data Flow

To duplicate a data flow, follow these steps:

1. Open the **Projects** page by clicking **Projects** from the left sidebar menu.
2. Locate and open the project containing the data flow you want to duplicate.
3. Open the Data Flows page by clicking **Data Flows** from the project left sidebar menu.
4. Locate the data flow you want to duplicate.

- Click the **Duplicate** icon  in the **Actions** column of the data flows' explorer. A pop-up will appear, where you must enter the number of copies you want to create.



A pop-up window titled 'Data Flow Duplication View'. It contains a text input field labeled 'Copy Number *' with the value '1' entered. Below the input field are two buttons: 'Cancel' and 'Save'.

Figure 44: Data Flow Duplication View

- Specify the number of copies and click **Save** to confirm.

(!) Note



The data flow duplication feature is available only when the original data flow is deployed.

(!) Note

Duplicated data flows are assigned randomly generated names, which can be modified later as needed.

5.1.4. Start/Stop Data Flows

To start or stop data flows, follow these steps:

- Open the **Projects** page by clicking **Projects** from the left sidebar menu.
- Locate and open the project containing the data flows you want to start/stop.
- Open the Data Flows page by clicking **Data Flows** from the project left sidebar menu.
- Locate and select the data flows you want to start/stop.
- From the data flows explorer top menu, click the **Start**  or **Stop**  icon.

Data Flows							
<div> <div>+ New</div> <div>▶ Start</div> <div>■ Stop</div> <div>🗑 Remove</div> </div> <div>Search</div>							
	Name	Status	Created at	Modified at	Created by	Modified by	Actions
<input checked="" type="checkbox"/>	MQTTBroker	Started	2022-07-19 20:17:13.727	2022-07-19 20:18:02.335	administrator	administrator	
<input type="checkbox"/>	Dataflow_1	Deployed	2022-08-09 11:27:31.645	2022-08-23 12:18:48.393	administrator	administrator	
<input checked="" type="checkbox"/>	Dataflow_2	Stopped	2022-08-15 17:18:41.203	2022-08-15 17:34:29.218	MAG	MAG	
<input type="checkbox"/>	Overall_test_1908	Deployed	2022-08-19 11:53:13.289	2022-08-23 09:42:40.698	administrator	administrator	
<input type="checkbox"/>	Dataflow_3	To be deployed	2022-08-22 12:08:55.739	2022-08-22 12:08:55.739	administrator	administrator	
<div>Items per page: 10</div> <div>Page 1 of 1 (1 - 5 of 5)</div> <div> <div><</div> <div>></div> </div>							

Figure 45: Start/Stop Data Flows


(!) Note

The **Start** and **Stop** buttons are enabled only when at least one data flow is selected and the selected data flows meet the required conditions:

- **Start** is enabled when all the selected data flows are deployed and stopped.
- **Stop** is enabled when all the selected data flows are deployed and currently running.

5.1.5. Delete Data Flow

To delete a data flow, follow these steps:


- Open the **Projects** page by clicking **Projects** from the left sidebar menu.
- Locate and open the project containing the data flow you want to delete.
- Open the Data Flows page by clicking **Data Flows** from the project left sidebar menu.
- Locate the data flow you want to delete.
- Click the **Delete** icon  in the **Actions** column of the data flows' explorer.

A confirmation dialog is displayed.

- Click **Yes** to permanently delete the data flow.
- Click **No** to abort the operation and retain the data flow.

5.2. Configure Data Flow

To open the data flow editor, follow these steps:

- Open the **Projects** page by clicking **Projects** from the left sidebar menu.
- Locate and open the project containing the data flow you want to open.
- Open the Data Flows page by clicking **Data Flows** from the project left sidebar menu.
- Locate the data flow you want to open.
- Click the **Open** icon  in the **Actions** column of the data flows' explorer.

The **Data Flow Editor** page is displayed, presenting the data flow graphical editor.

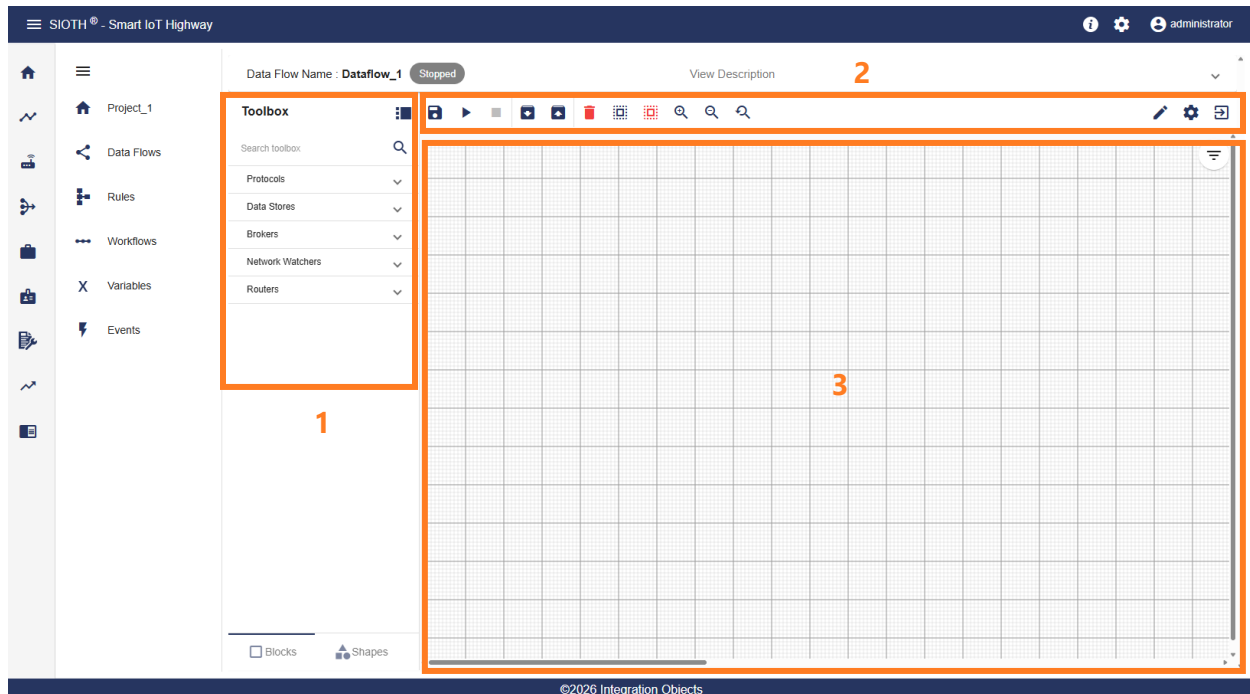


Figure 46: Data Flow Editor

The data flow editor is divided into three main areas:

- **Toolbox (1):** Provides components that can be dragged and dropped into the workspace:
 - **Blocks:** Represent SIOTH connectors components. Available blocks categories include Protocols, Data Stores, Brokers, Network Watchers, and Routers.

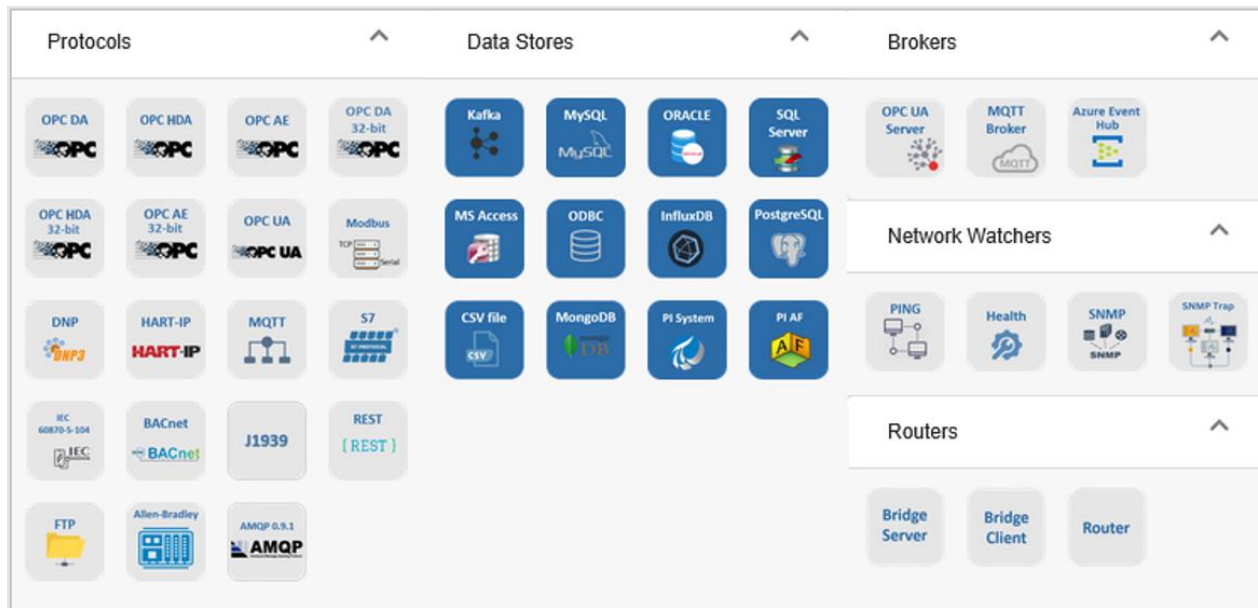


Figure 47: Blocks Toolbox

- **Shapes:** Includes drawing tools such as lines, geometric shapes, and image insertion tools.

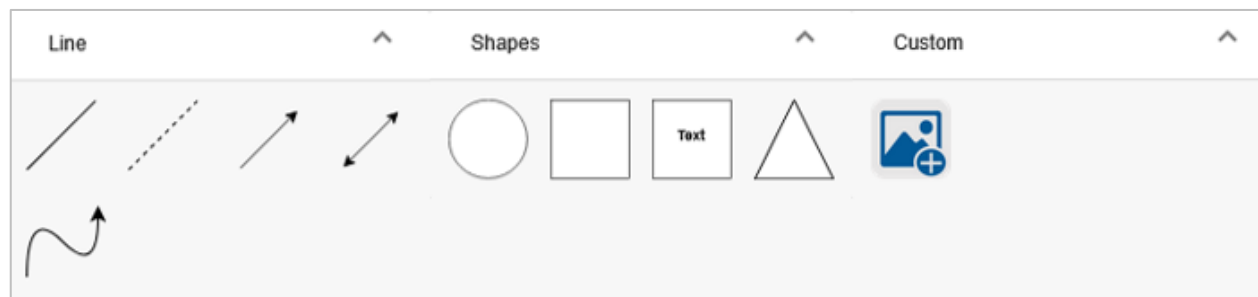


Figure 48: Shapes Toolbox

- **Editor Toolbar (2):** Provides access to core actions, including saving, starting, or stopping a data flow. It also allows to import or export data flows, delete, select or deselect all the elements, zoom in and out, reset the zoom level, configure grid setting and exit the editor to return to the data flow list.
- **Workspace (3):** The main design area where blocks are placed, connected, and configured to define data flows.

To begin configuring a data flow, drag and drop the required blocks from the **Toolbox** into the **Workspace**.

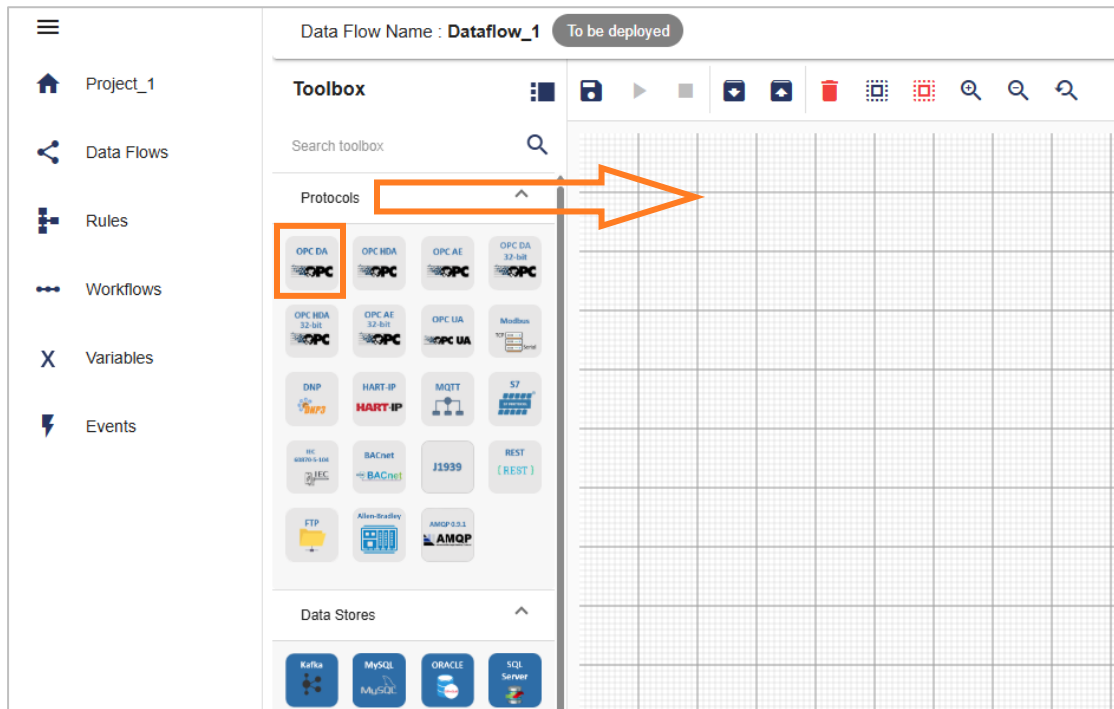



Figure 49: Configure Data Flow - Drag and Drop Connectors

To configure a connector, select the block in the workspace and click the **Settings** icon .

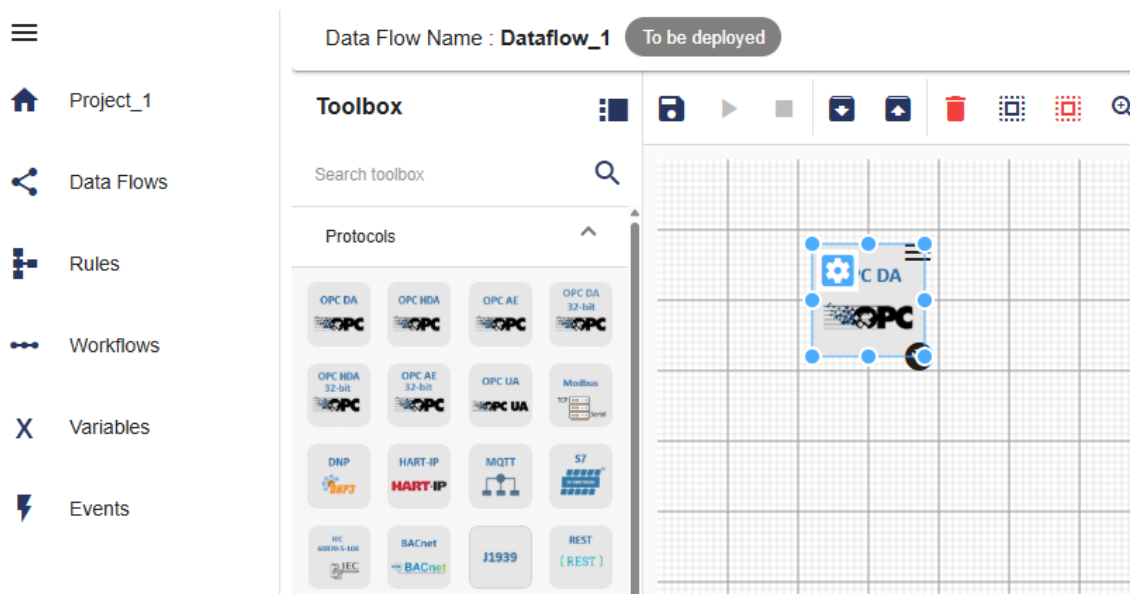


Figure 50: Configure Data Flow - Blocks Configuration

A configuration window opens, allowing you to define connector parameters. Click **Save** to store the configuration.

Once both **source** and **destination** connectors are configured, click **Save and Deploy** from the toolbar. This saves the data flow and deploys the connectors to their assigned nodes.

To start data exchange, click **Start** from the editor toolbar, or return to the **Data Flows** list, select the data flow, and click **Start**.

Source and destination connectors share most of the same configuration parameters. Additional settings depends on the selected block.

5.2.1. Identification Step

When opening the connector settings, the configuration process starts with the **Identification** step.

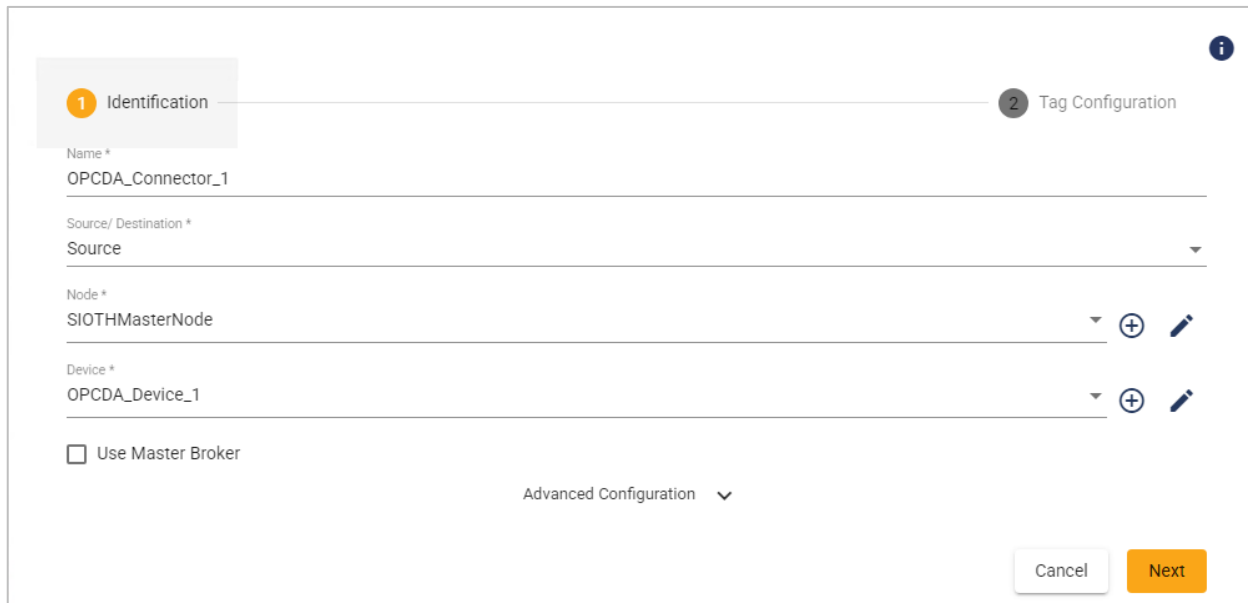


Figure 51: Connector Identification Configuration View

All connectors share the same core configuration parameters. However, some connectors include additional parameters specific to their respective protocols.

Parameter	Description	Default Value
Name	Defines the name of the connector.	<ConnectorType>_Connector_X
Source / Destination	<p>Specifies the connector flow type:</p> <ul style="list-style-type: none"> • Source: Subscribes to data from external data sources and publishes it to SIOTH. • Destination: Subscribes to data from SIOTH and publishes it to external data sources. 	Source
Node	Specifies the node (machine) hosting the connector.	SIOTHMasterNode
Device	Specifies the device associated with the connector.	
Use Master Broker	Enables the use of the Master Broker for data transfer.	Unchecked
Advanced Configuration		
Data Encryption	Secures transmitted data across cloud and system environments.	Checked
Offline	Allows downloading the connector package for manual or offline deployment.	Unchecked
Allow Incoming Write Requests	Allows write requests from Destination connectors to Source connectors.	Checked

<i>Service Configuration</i>		
<i>Service Configuration</i>	Enables configuration and adjustment of service parameters.	Checked
<i>Logon Type</i>	Specifies the account used to run the service: <ul style="list-style-type: none"> Local System Specific Account 	Local System
<i>Username</i>	Username of the specific account when Logon Type is set to <i>Specific Account</i> .	
<i>Password</i>	Password of the specific account when Logon Type is set to <i>Specific Account</i> .	
<i>Startup Type</i>	Defines how and when the service starts: <ul style="list-style-type: none"> Automatic Automatic (Delayed Start) Manual Disabled 	Automatic
<i>Data Transfer Mode</i>		
<i>Topic</i>	When the connector is set as Source , it transfers all tags to a single topic.	Checked
<i>Per Tag</i>	When the connector is set as Source , transfers each tag individually.	Unchecked

Data Publishing Port	Network port used by the Source connector to publish data to the destination.	
Data Request Port	Network port used to receive or request data from the source.	
Client ID	Unique identifier for the connector when using the SIOTH Master Broker.	
Store and Forward		
Store and Forward	Buffers data locally when communication fails and forwards it once the connection is restored. Automatically enabled for Source connectors and disabled for Destination connectors.	Checked
Forward Type	<p>Defines how buffered data is retrieved when the connector is set as Destination:</p> <ul style="list-style-type: none"> • FIFO: Data is retrieved in the order received. • Latest Only: Only the most recent data is retrieved. 	
Batch Size (Rows)	Number of rows processed per data transfer batch.	1000
Maximum Store Size (MB)	Maximum local storage size used for buffering data.	5120
Log Settings		

<i>Auto Append</i>	Automatically appends new log entries to existing log files.	Checked
<i>Log Level</i>	Defines the log severity level: <ul style="list-style-type: none"> Information Debug 	Information
<i>Buffer Size (MB)</i>	Amount of memory allocated for temporary data buffering.	100
<i>Maximum Files</i>	Maximum number of log files retained.	10
<i>Log File Max size (MB)</i>	Maximum size allowed per log file.	10
<i>Auto Save Timeout (s)</i>	Time interval (in seconds) before data is automatically saved.	5

Table 37: Connector Identification Configuration Parameters

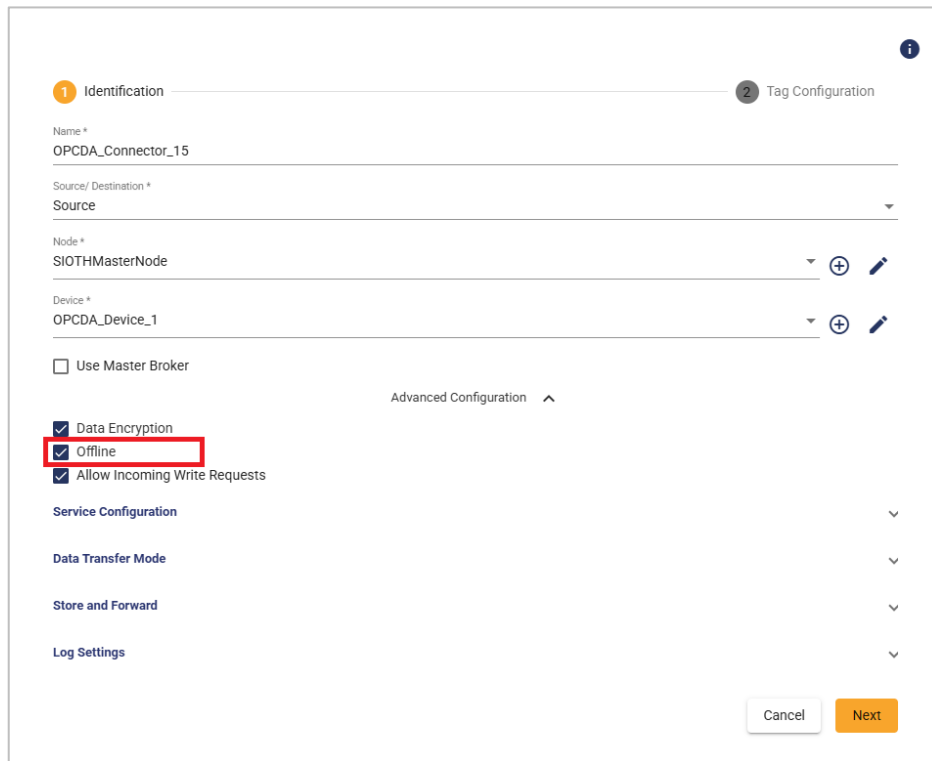
Once the connector identification step is completed, click **Next** to proceed to the next configuration page.

5.2.2. Offline Deployment

When network access to the target machine is unavailable, you can use the **Offline** (Manual) deployment option. This approach allows you to download the connector configuration package and deploy it manually on a remote node, without requiring an active data flow deployment.


To deploy your connector in offline mode, follow the steps below:

- 1- During the connector configuration, enable the **Offline** option in the **Advanced Configuration** section.



The image shows a configuration window with two tabs: '1 Identification' and '2 Tag Configuration'. The 'Identification' tab is active. It contains fields for 'Name *' (OPCDA_Connector_15), 'Source/ Destination *' (Source), 'Node *' (SIOTMasterNode), and 'Device *' (OPCDA_Device_1). There are expandable sections for 'Advanced Configuration', 'Service Configuration', 'Data Transfer Mode', 'Store and Forward', and 'Log Settings'. In the 'Advanced Configuration' section, three checkboxes are visible: 'Data Encryption' (checked), 'Offline' (checked and highlighted with a red box), and 'Allow Incoming Write Requests' (checked). At the bottom right, there are 'Cancel' and 'Next' buttons.

Figure 52: Connector Identification Configuration View - Offline Option

- 2- Click **Save** to store the connector configuration.
- 3- Reopen the connector configuration view to access the saved settings.
- 4- Click the **Download Configuration** icon  next to the **Offline** option in the **Advanced Configuration** section of the **Identification** tab. A JSON file containing the connector configuration will be downloaded.



This image is a zoomed-in view of the 'Advanced Configuration' section from Figure 52. It shows the 'Offline' checkbox, which is checked and has a red box around it. Next to the 'Offline' checkbox is a yellow download icon (a square with a downward arrow).

Figure 53: Connector Identification Configuration View - Download Configuration

- 5- Copy the downloaded JSON configuration file and place it into the **Agent Configuration** folder of the remote node installation directory.

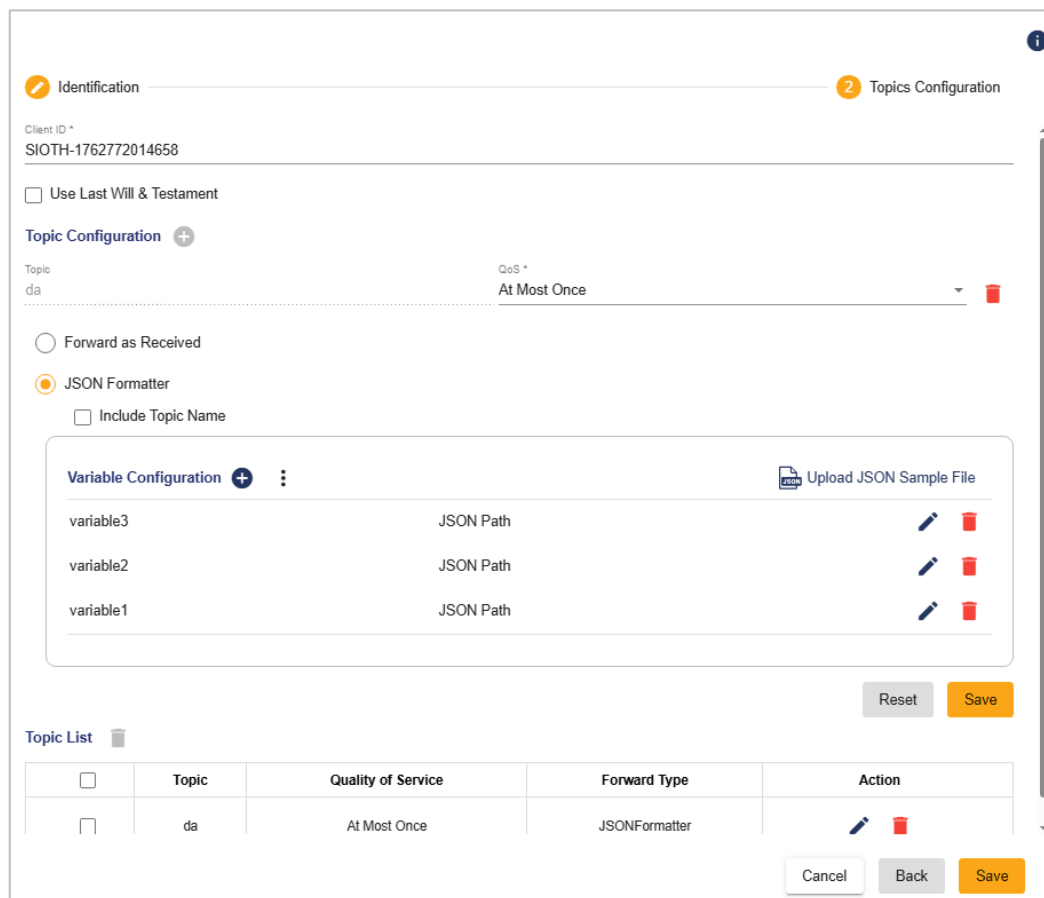
Path: *..\Program Files\Integration Objects\Integration Objects' Smart IoT Highway\Connectors*

5.2.3. Payload Transformation Step

For **non-tag-based** source connectors such as **MQTT Standard** configured with **Variables**, the Payload Transformation step becomes available when the destination connector is either **MS SQL Server** or **PI System**.

To enable and configure payload transformation, follow the steps below:

- 1- Configure the non-tag-based source connector (such as MQTT Standard) using Variables Configuration.



Identification Topics Configuration

Client ID *

SIOTH-1762772014658

☐ Use Last Will & Testament

Topic Configuration +

Topic

da

QoS *

At Most Once

☐ Forward as Received

☒ JSON Formatter

☐ Include Topic Name

Variable Configuration +

Upload JSON Sample File

Variable	JSON Path	Action
variable3	JSON Path	
variable2	JSON Path	
variable1	JSON Path	

Reset Save

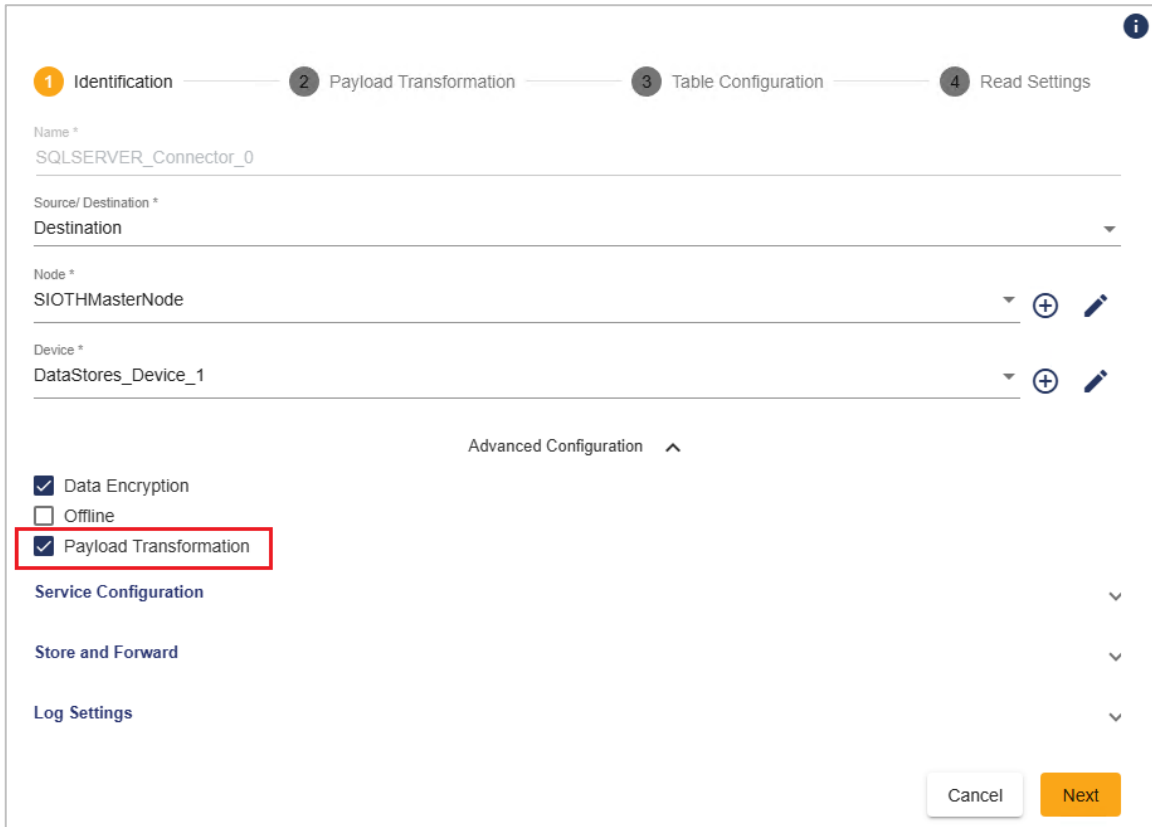
Topic List

	Topic	Quality of Service	Forward Type	Action
<input type="checkbox"/>	da	At Most Once	JSONFormatter	

Cancel Back Save

Figure 54: Connector Topics Configuration View - Variable Configuration

- 2- Navigate to the **Advanced Configuration** section in destination connector (such as MS SQL Server) during the Identification step and enable the **Payload Transformation** option.



The screenshot shows the 'Connector Identification Configuration View' with four steps: 1 Identification, 2 Payload Transformation, 3 Table Configuration, and 4 Read Settings. The 'Identification' step is active. Fields include: Name * (SQLSERVER_Connector_0), Source/ Destination * (Destination), Node * (SIOTMasterNode), and Device * (DataStores_Device_1). The 'Advanced Configuration' section is expanded, showing checkboxes for Data Encryption (checked), Offline (unchecked), and Payload Transformation (checked and highlighted with a red box). Below are expandable sections for Service Configuration, Store and Forward, and Log Settings. At the bottom are 'Cancel' and 'Next' buttons.

Figure 55: Connector Identification Configuration View - Payload Transformation Configuration

- 3- Proceed to the **Payload Transformation** step and configure the following sections:

Step Details:

Input Schema Selection:

- Select the source connector from the **Input Schema** list.
- Use the **Preview Schema** button to verify the available source variables.

Repeating Fields Configuration:

- Add repeating fields in the **Output Schema** section to define the target payload structure. These fields represent the payload elements that will be mapped to SQL table columns in the next step.

- Specify the number of repetitions for each field based on the number of variables or tags.
- Confirm the configuration to display the target and source fields in the **Field Transformation** table.

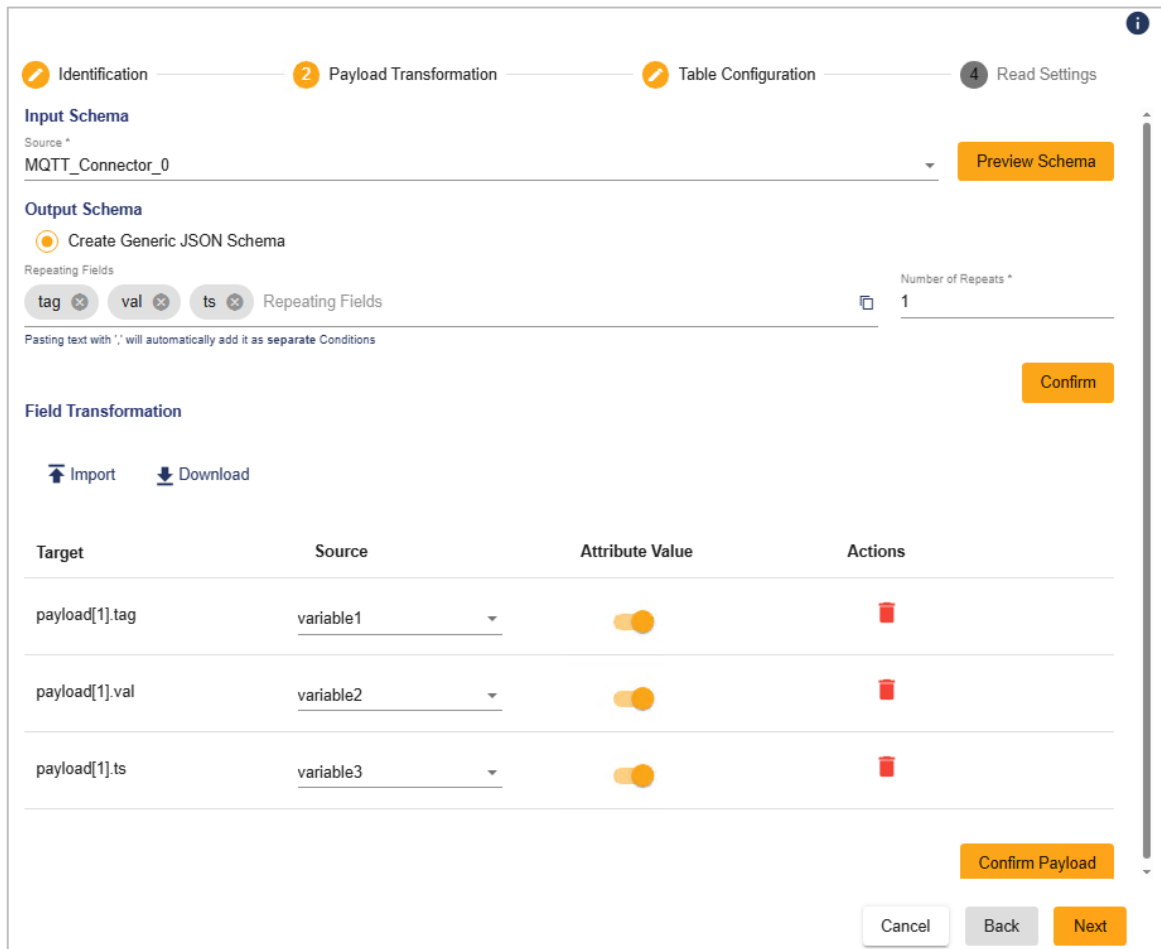
Field Transformation:

This section displays fields using the format: `targetPayload[i].repeatingField`.

For each field, configure the following options:

- **Source:** Select the corresponding source field from the dropdown list generated from the input schema.
- **Attribute Value:** Enabled by default to publish the field value. Disable this option to publish the field name instead.
- **Delete:** Delete the field from the transformation list.

- 4- Confirm the payload transformation configuration.

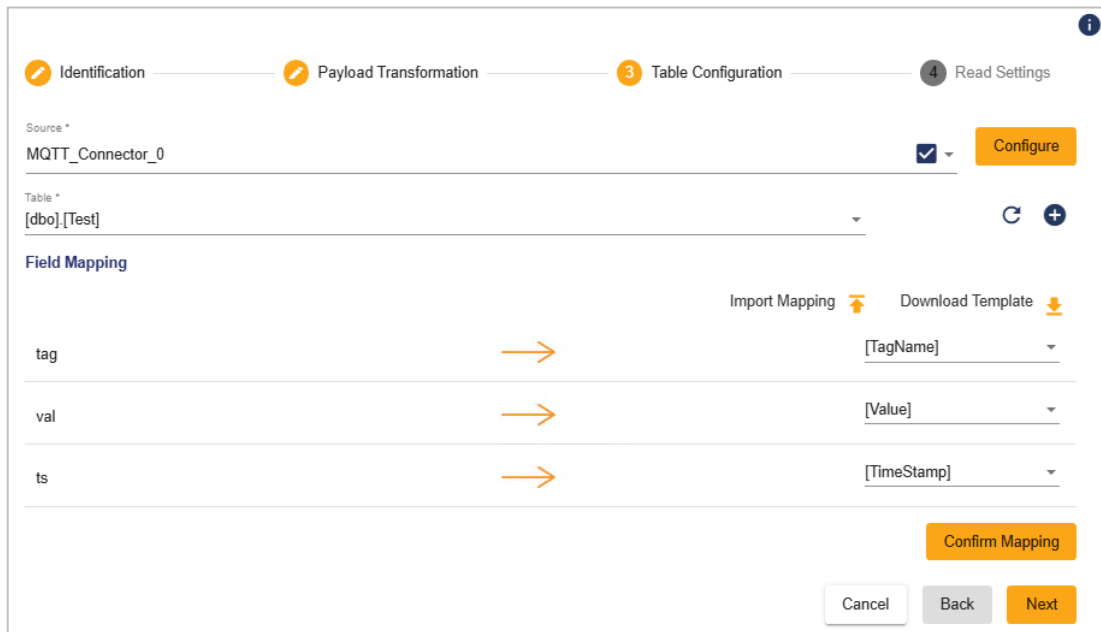


The screenshot shows the 'Payload Transformation' step in the configuration wizard. It includes sections for 'Input Schema' (MQTT_Connector_0), 'Output Schema' (Create Generic JSON Schema), 'Repeating Fields' (tag, val, ts), and a 'Field Transformation' table. The table maps payload fields to source variables and includes toggle switches and delete icons. Navigation buttons like 'Next' and 'Confirm Payload' are at the bottom.

Target	Source	Attribute Value	Actions
payload[1].tag	variable1	<input checked="" type="checkbox"/>	
payload[1].val	variable2	<input checked="" type="checkbox"/>	
payload[1].ts	variable3	<input checked="" type="checkbox"/>	

Figure 56: Connector Payload Transformation Configuration View

- 5- Click **Next** to proceed to the **Table Configuration** step:
 - Select the source connector and the target table.
 - The mapping view displays repeating fields on the left and SQL table columns on the right. Map each repeating field to its corresponding SQL column.
 - Confirm the mapping to save the configuration.



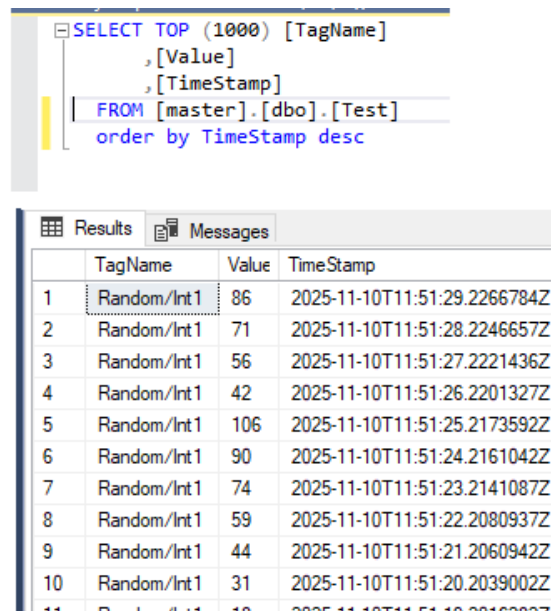
The screenshot shows the 'Table Configuration' step in a four-step process. The 'Source' is 'MQTT_Connector_0' and the 'Table' is '[dbo].[Test]'. The 'Field Mapping' section shows three fields: 'tag' mapped to '[TagName]', 'val' mapped to '[Value]', and 'ts' mapped to '[TimeStamp]'. There are buttons for 'Import Mapping', 'Download Template', 'Confirm Mapping', 'Cancel', 'Back', and 'Next'.

Figure 57: Connector Table Configuration View

6- Click **Save** to store the connector configuration.

7- Click **Save and Deploy**, then click **Start** to launch the data flow.

As shown below, the **MQTT payload data** will be automatically stored in the SQL table in a **tag-based format**.



The top part shows a SQL query: `SELECT TOP (1000) [TagName], [Value], [TimeStamp] FROM [master].[dbo].[Test] order by TimeStamp desc`. The bottom part shows a table with 10 rows of data.

	TagName	Value	TimeStamp
1	Random/Int1	86	2025-11-10T11:51:29.2266784Z
2	Random/Int1	71	2025-11-10T11:51:28.2246657Z
3	Random/Int1	56	2025-11-10T11:51:27.2221436Z
4	Random/Int1	42	2025-11-10T11:51:26.2201327Z
5	Random/Int1	106	2025-11-10T11:51:25.2173592Z
6	Random/Int1	90	2025-11-10T11:51:24.2161042Z
7	Random/Int1	74	2025-11-10T11:51:23.2141087Z
8	Random/Int1	59	2025-11-10T11:51:22.2080937Z
9	Random/Int1	44	2025-11-10T11:51:21.2060942Z
10	Random/Int1	31	2025-11-10T11:51:20.2039002Z

Figure 58: Payload Transformation Result

5.2.4. Tag Configuration Step

After completing the **Source Connector Identification**, the next step is **Tag Configuration**. SIOTH connectors allow to configure one or multiple subscriptions, depending on the connector type and the selected data source.

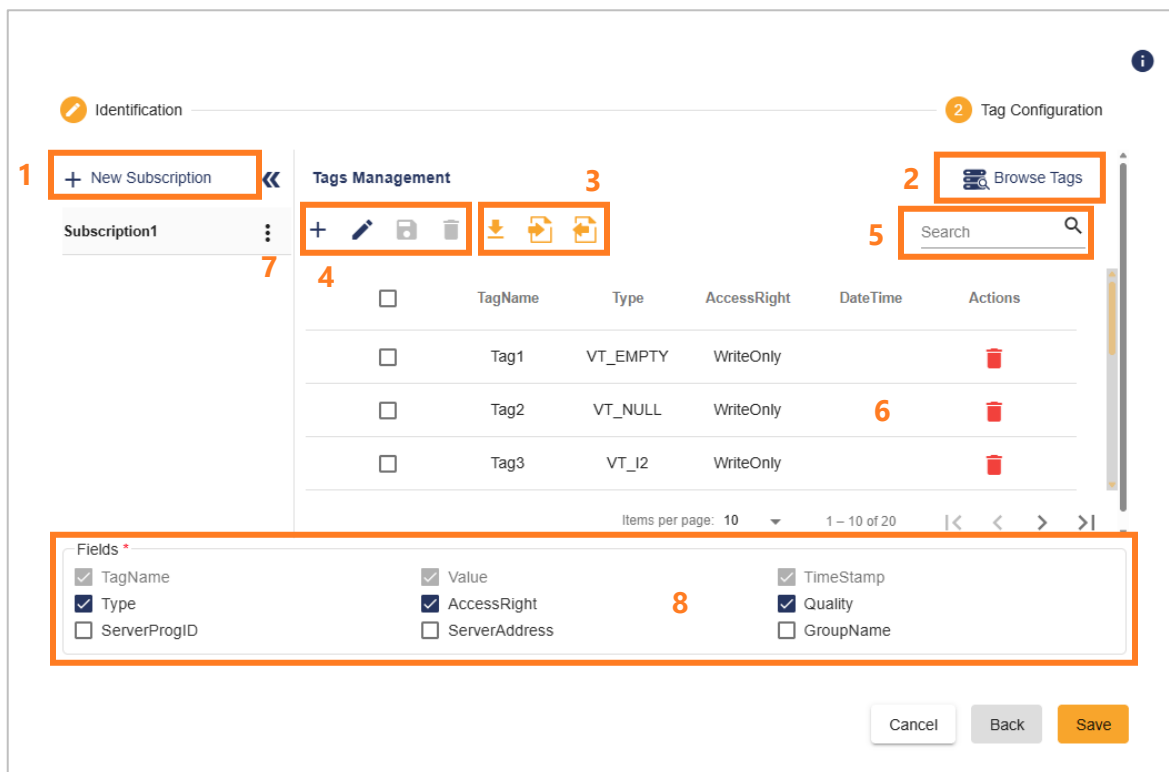


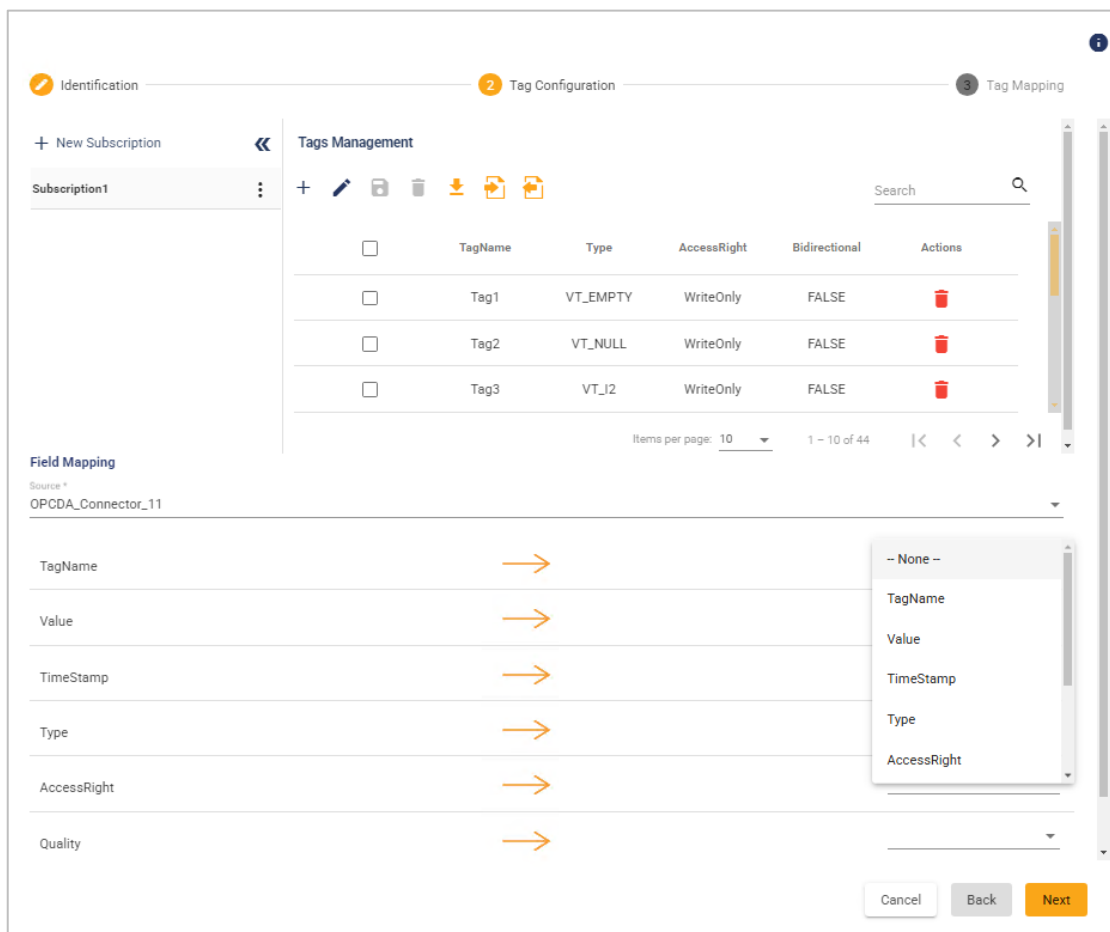
Figure 59: Connector as Source - Tag Configuration View

The following actions are available in the Tag Configuration view:

- **New Subscription (1):** Create a new subscription.
- **Browse Tags (2):** Browse the connected device to select tags for import.
- **Download Template (3):** Download a tags template for reference.
- **Import (3):** Import tags using a CSV file.
- **Export (3):** Export the configured tags for verification or backup.

- **Add, Edit, Save** and **Remove** (4): Manage tags by adding, editing, saving, or removing entries.
- **Search (5)**: Search for a specific tag using the search field.
- **Preview Imported tags (6)**: Preview the list of imported tags before applying them.
- **Edit, Delete subscription (7)**: Modify or delete an existing subscription.
- **Fields (8)**: Configure the fields associated with all tags.

For **Destination connectors**, the **Fields** option is replaced by **Field Mapping**. This feature allows you to map fields from the source connector to the corresponding fields of the destination connector.



The screenshot displays the 'Tag Configuration' view, which is divided into two main sections: 'Tags Management' and 'Field Mapping'.

Tags Management: This section contains a table with the following columns: TagName, Type, AccessRight, Bidirectional, and Actions. The table lists three tags: Tag1 (VT_EMPTY, WriteOnly, FALSE), Tag2 (VT_NULL, WriteOnly, FALSE), and Tag3 (VT_I2, WriteOnly, FALSE). Each tag has a checkbox in the first column and a red trash icon in the Actions column. A search bar is located to the right of the table. Below the table, there is a pagination control showing 'Items per page: 10' and '1 - 10 of 44'.

Field Mapping: This section is for mapping fields from the source connector 'OPCDA_Connector_11' to the destination connector. It lists several fields: TagName, Value, TimeStamp, Type, AccessRight, and Quality. Each field has a right-pointing arrow next to it. A dropdown menu is open, showing the following options: -- None --, TagName, Value, TimeStamp, Type, and AccessRight. At the bottom of the field mapping section, there are three buttons: 'Cancel', 'Back', and 'Next'.

Figure 60: Connector as Destination - Tag Configuration View

Once the field mapping is completed, click **Confirm Mapping** to validate it. After a source connector has been successfully mapped, a checkmark icon appears next to its name.

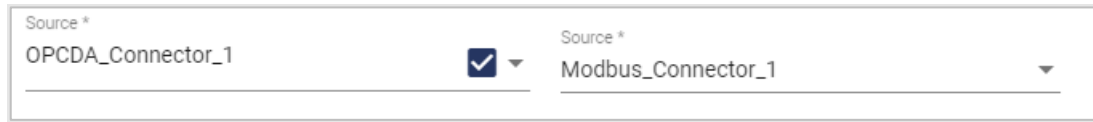


Figure 61: Connector as Destination - Mapped vs. Unmapped Source Connector

To add a subscription, click the **New Subscription** button from the left section in the **Tag Configuration** page. A new window will open where you can configure the subscription settings.

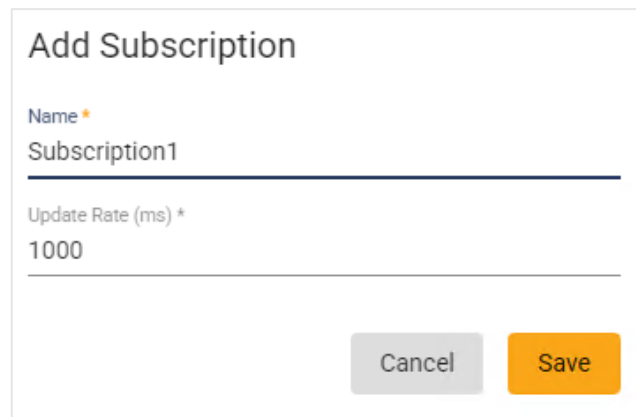




Figure 62: Connector - New Subscription Configuration View

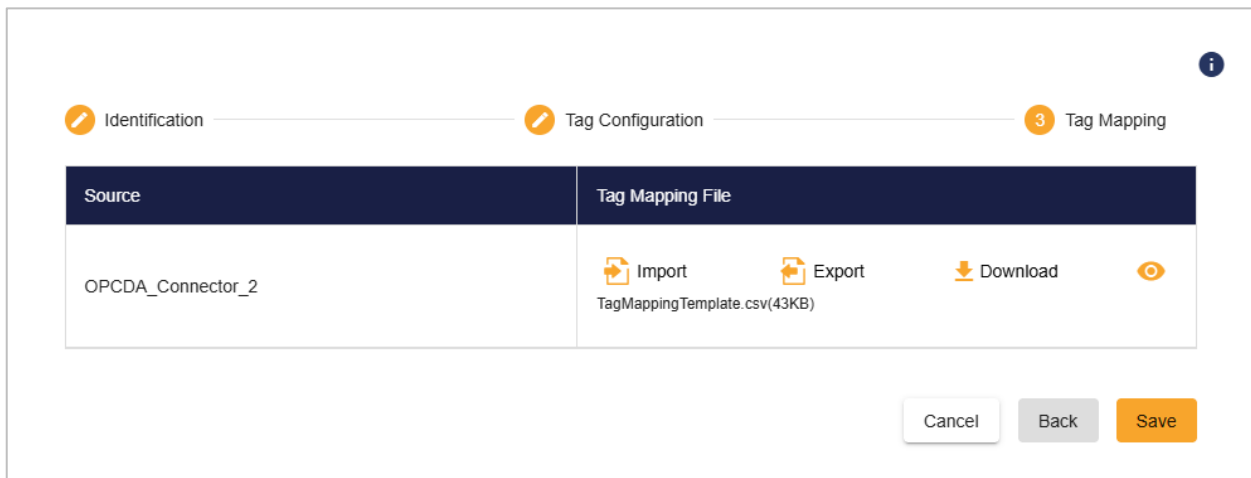
Parameter	Description	Default Value
Name	Defines the name of the subscription.	Subscription1
Update Rate (ms)	Specifies the frequency of data read requests. In OnDataChange mode, this value defines the maximum rate at which data change notifications are sent to the client callbacks.	1000





Table 38: Connector - New Subscription Configuration Parameters

5.2.5. Tag Mapping Step

For **Destination connectors**, tag mapping between the source and destination connectors is mandatory to ensure correct data transfer. You can define the tag mapping manually or import it using a **CSV file** by clicking the **Import** icon . This approach is especially useful when configuring large numbers of tags.

Additionally, you can download a **Tags Mapping Template** by clicking the **Download Tags Template** icon . This template serves as a reference for creating a valid mapping file and helps ensure that the required format and structure are respected.



Source	Tag Mapping File
OPCDA_Connector_2	 Import  Export  Download 

Cancel Back Save

Figure 63: Connector as Destination - Tag Mapping

5.2.6. Protocols

Protocol connectors enable communication between **SIOTH** and external devices or systems using standard industrial and communication protocols.

5.2.6.1. Allen Bradley

Click **New Subscription** from the left section in the **Tag Configuration** page to add a subscription to an Allen Bradley connector. A new window will open where you can configure the subscription settings.

Add Subscription

Name *
Subscription1

Device *
AllenBradley_Device_1

Update Rate (ms) *
1000

Auto Sync Rate (ms) *
500

Cancel Save

Figure 64: Allen Bradley Source/Destination Connector - Subscription Configuration View

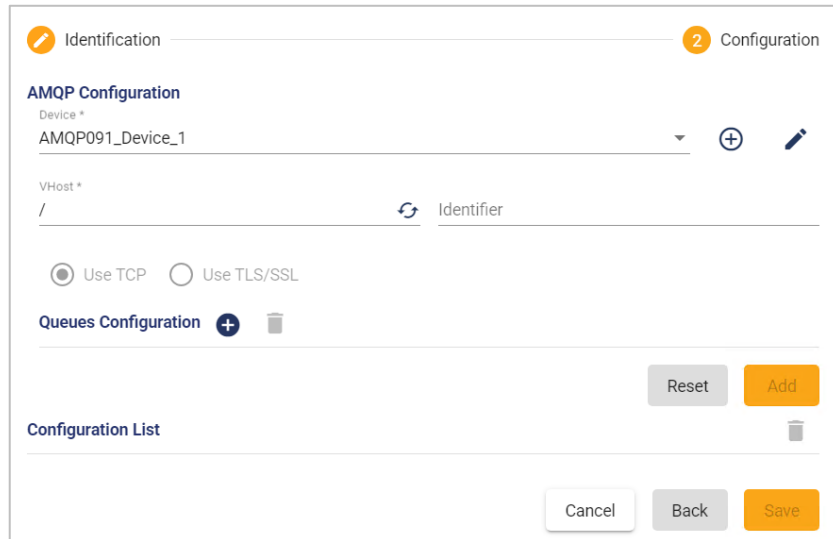
Parameter	Description	Default Value
Name	Specifies the name of the subscription.	Subscription1
Device	Identifies the Allen Bradley device from which data is read or to which data is written.	
Update Rate (ms)	Defines the interval at which data read requests are sent to the device. This parameter is configurable when the connector is configured as a Source .	1000
Auto Sync Rate (ms)	Specifies how often the locally cached data is automatically synchronized with the PLC to ensure consistency with the device values. This parameter is configurable when the connector is configured as a Source .	500

Table 39: Allen Bradley Source/Destination Connector - Subscription Configuration Parameters

5.2.6.2. AMQP-091

Click **Next** to access the **AMQP-091** specific configuration. The parameters displayed vary depending on whether the connector is configured as a **Source** or a **Destination**.

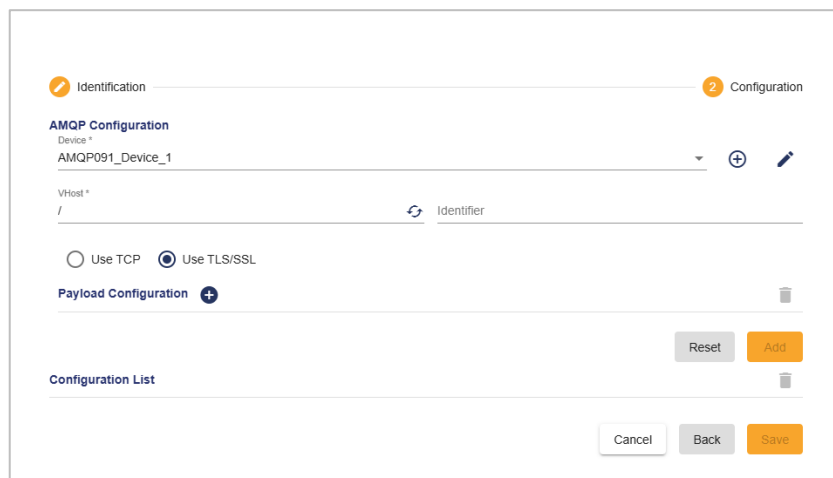
AMQP-091 Connector as Source:



The configuration view for AMQP-091 as a Source connector. It features a top navigation bar with 'Identification' (active) and 'Configuration' (2). The main section is titled 'AMQP Configuration' and includes a 'Device *' dropdown set to 'AMQP091_Device_1'. Below this is a 'VHost *' field with a slash '/' and an 'Identifier' field with a refresh icon. There are two radio buttons: 'Use TCP' (selected) and 'Use TLS/SSL'. A 'Queues Configuration' section has a plus icon and a trash icon. At the bottom right are 'Reset' and 'Add' buttons. A 'Configuration List' section at the bottom has a trash icon and 'Cancel', 'Back', and 'Save' buttons.

Figure 65: AMQP-091 Connector as Source - Configuration View

AMQP-091 Connector as Destination:



The configuration view for AMQP-091 as a Destination connector. It features a top navigation bar with 'Identification' (active) and 'Configuration' (2). The main section is titled 'AMQP Configuration' and includes a 'Device *' dropdown set to 'AMQP091_Device_1'. Below this is a 'VHost *' field with a slash '/' and an 'Identifier' field with a refresh icon. There are two radio buttons: 'Use TCP' and 'Use TLS/SSL' (selected). A 'Payload Configuration' section has a plus icon and a trash icon. At the bottom right are 'Reset' and 'Add' buttons. A 'Configuration List' section at the bottom has a trash icon and 'Cancel', 'Back', and 'Save' buttons.

Figure 66: AMQP-091 Connector as Destination - Configuration View

Parameter	Description	Default Value
AMQP Configuration		
Device	Refers to the AMQP 0.91 device from which messages are received or to which messages are published.	
VHost	Specifies the Virtual Host (VHost) name. The VHost defines a logical grouping of exchanges, queues, and permissions within the AMQP broker.	/
Identifier	A unique identifier used to distinguish clients, connections, or resources (such as queues or exchanges) within the VHost, supporting routing, tracking, and connection management.	
Use TCP	Enables communication with the AMQP broker using the TCP protocol. Availability depends on the selected device configuration.	
Use TLS/SSL	Enables encrypted communication between the client and the AMQP broker using TLS/SSL, ensuring secure message transmission. Availability depends on the selected device configuration.	
Queues Configuration - (Connector as Source)		
Name	A unique name identifying the queue from which messages are consumed.	
Type	Defines the queue type.	Classic
Content Encoding	Filters incoming messages based on their content encoding.	None

Acknowledge Mode	<p>Defines how message acknowledgments are handled:</p> <ul style="list-style-type: none"> • Auto Acknowledge: Messages are automatically acknowledged upon receipt. • Positive Acknowledgment: Messages are acknowledged manually after successful processing. • Nack with Requeue True: Messages are negatively acknowledged and requeued for reprocessing. • Nack with Requeue False: Messages are negatively acknowledged and discarded or routed to a dead-letter queue. 	Auto Acknowledge
Filters		
Use Filters	Enables filtering of received messages based on predefined criteria.	Unchecked
Delivery Mode	<p>Defines message persistence:</p> <ul style="list-style-type: none"> • Persistent: Messages are written to disk to prevent data loss. • Transient: Messages are stored in memory only. 	Persistent
Type	Specifies the message content type (e.g., text, binary, JSON).	
Message ID	A unique identifier assigned to each message for tracking and deduplication.	

Correlation ID	Used to associate related messages, typically in request/response communication patterns.	
User ID	Identifies the user associated with the message, supporting authentication and authorization.	
App ID	Identifies the application that generated the message.	
Headers Configuration	<p>Allows defining custom message headers. Each header includes:</p> <ul style="list-style-type: none"> • Key: The name of the header, used to identify the additional message metadata. • Value: The content associated with the key, which can be of various types. • Type: The type of the header value, which can be a string, number, or boolean. 	
Queues Configuration - (Connector as Destination)		
Source	Specifies the source connector from which data is received.	
Publish Mode	<p>Defines how messages are published:</p> <ul style="list-style-type: none"> • Basic: Messages are published individually. • Per Batch: Messages are published in batches to improve throughput. 	Basic
Exchange Configuration		
Exchange	Defines the AMQP exchange responsible for routing messages to queues.	

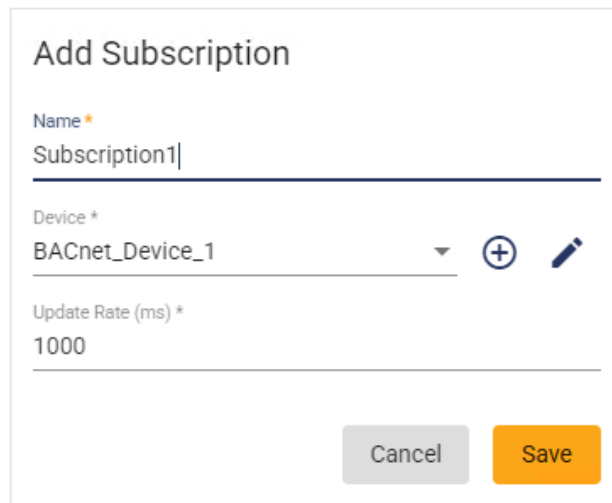
<i>Routing Keys</i>	Specifies routing keys used by the exchange to route messages to the appropriate queues.	
<i>Message Properties</i>		
<i>Delivery Mode</i>	Defines message persistence (Persistent or Transient).	Persistent
<i>Type</i>	Specifies the message content type (e.g., text, binary, JSON).	
<i>Content Type</i>	Indicates the payload content type.	application/Json
<i>Content Encoding</i>	Filters messages based on content encoding. Supported values include None , Gzip , and Zip .	None
<i>Message ID</i>	A unique identifier assigned to each message.	
<i>Correlation ID</i>	Associates related messages within a transaction or communication flow.	
<i>Expiration (ms)</i>	Specifies the message time-to-live (TTL) in milliseconds. After expiration, the message is discarded or routed to a dead-letter queue.	
<i>Timestamp</i>	Defines when the timestamp is applied to the message: <ul style="list-style-type: none"> • Publish Timestamp: Applies the publish time. • None: No timestamp applied. 	Publish Timestamp
<i>User ID</i>	Identifies the user associated with the message.	
<i>App ID</i>	Identifies the application that generated the message.	
<i>Headers Configuration</i>	Allows defining custom message headers. Each header includes:	

	<ul style="list-style-type: none"> • Key: The name of the header, used to identify the additional message metadata. • Value: The content associated with the key, which can be of various types. • Type: The type of the header value, which can be a string, number, or boolean. 	
Data Filtering	Enables filtering of received data using configurable templates that can be downloaded, imported, or exported.	Unchecked
Fields Aliasing	Allows renaming payload fields for improved clarity or compatibility.	Unchecked
Data Aliasing	Enables defining alternate names for data elements using aliasing templates that can be downloaded, imported, or exported.	Unchecked
Split Payload	Allows large payloads to be split into smaller parts for optimized processing.	Unchecked
JSON Formatter	Ensures JSON payloads are properly structured. Supports variable-based configuration or manual definition using the Payload Builder.	Unchecked
Configuration List		
Configuration List	Provides access to existing AMQP configurations, allowing users to view, edit, or delete configurations.	

Table 40: AMQP-091 Connector - Configuration Parameters

5.2.6.3. BACnet

Click **New Subscription** from the left section in the **Tag Configuration** page to add a subscription to a BACnet connector. A new window will open where you can configure the subscription settings.



The image shows a 'Add Subscription' dialog box. It contains three input fields: 'Name' with the value 'Subscription1', 'Device' with a dropdown menu showing 'BACnet_Device_1' and a plus icon, and 'Update Rate (ms)' with the value '1000'. At the bottom, there are 'Cancel' and 'Save' buttons.

Figure 67: BACnet Connector - Subscription Configuration View

Parameter	Description	Default Value
Name	Name of the subscription.	Subscription1
Device	Refers to the BACnet device from which data will be read or to which data will be written.	
Update Rate (ms)	Defines the frequency of data read requests in milliseconds. In On Data Changes mode, this value sets the maximum rate at which data change notifications are sent to the client's callback. Lower values result in more frequent updates, while higher values reduce communication overhead.	1000

Table 41: BACnet Connector - Subscription Configuration Parameters

(!) Note

When the BACnet Connector is configured as **Destination**, the **Device** selection is performed during the **Identification** step, rather than in the subscription configuration.

5.2.6.4. DNP3

Click **New Subscription** from the left panel of the **Tag Configuration** page to add a subscription to a DNP3 connector configured as **Source**. A configuration window will open, allowing you to define the subscription parameters.



Figure 68: DNP3 Connector as Source - Subscription Configuration View

Parameter	Description	Default Value
<i>DNP Device</i>	Identifies the DNP3 device from which data will be read or to which data will be written.	

Table 42: DNP3 Connector as Source - Subscription Configuration Parameters

When the connector is configured as **Destination** and you reach the **Tag Configuration** page, select the **Source Connector** and configure the corresponding **field mapping**.

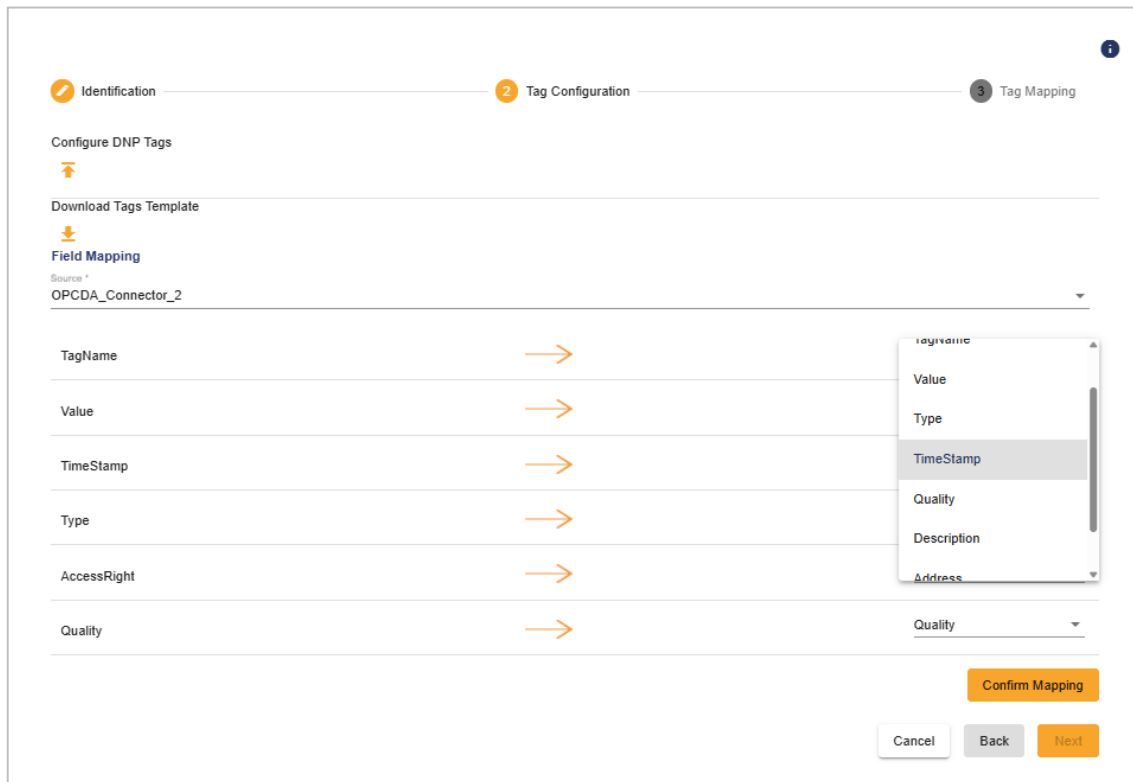


Figure 69: DNP3 Connector as Destination - Tag Configuration View

5.2.6.5. FTP

Click **New Subscription** from the left panel of the **Tag Configuration** page to add a subscription to an FTP connector. A configuration window will open, allowing you to define the subscription parameters.

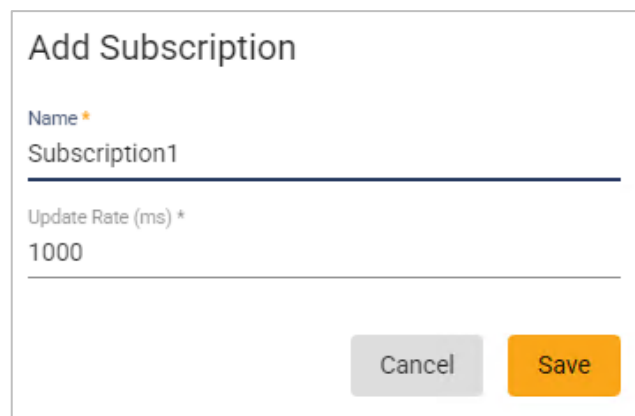


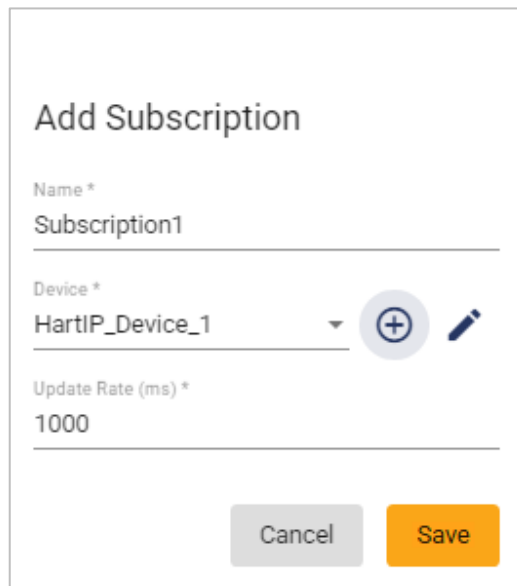
Figure 70: FTP Connector - Subscription Configuration View

Parameter	Description	Default Value
Name	Name assigned to the subscription.	Subscription1
Update Rate (ms)	Frequency of data read requests in milliseconds. In On Data Changes mode, this defines the maximum rate at which data change notifications are sent to the client's callback. Lower values increase update frequency, while higher values reduce communication overhead.	1000

Table 43: FTP Connector - Subscription Configuration Parameters

5.2.6.6. HARTIP

Click **New Subscription** from the left panel of the **Tag Configuration** page to add a subscription to a HARTIP connector. A configuration window will open, allowing you to define the subscription parameters.



The image shows a configuration window titled "Add Subscription". It contains three input fields: "Name *" with the value "Subscription1", "Device *" with a dropdown menu showing "HartIP_Device_1" and a plus icon, and "Update Rate (ms) *" with the value "1000". At the bottom, there are two buttons: "Cancel" and "Save".

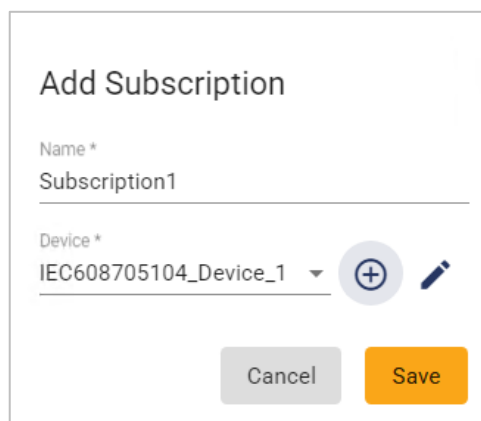
Figure 71: HARTIP Connector - Subscription Configuration View

Parameter	Description	Default Value
Name	Name assigned to the subscription.	Subscription1
Device	Specifies the HARTIP device from which data will be read or to which data will be written.	
Update Rate (ms)	Frequency of data read and write requests in milliseconds. In On Data Changes mode, this defines the maximum rate at which data change notifications are sent to the client's callback. Lower values increase update frequency, while higher values reduce communication overhead.	

Table 44: HARTIP Connector - Subscription Configuration Parameters

5.2.6.7. IEC 60870-5-104

Click **New Subscription** from the left panel of the **Tag Configuration** page to add a subscription to an IEC connector. A configuration window will open, allowing you to define the subscription parameters.



The image shows a configuration window titled "Add Subscription". It contains two input fields: "Name *" with the value "Subscription1" and "Device *" with a dropdown menu showing "IEC608705104_Device_1". To the right of the dropdown is a circular button with a plus sign and a pencil icon. At the bottom are two buttons: "Cancel" and "Save".

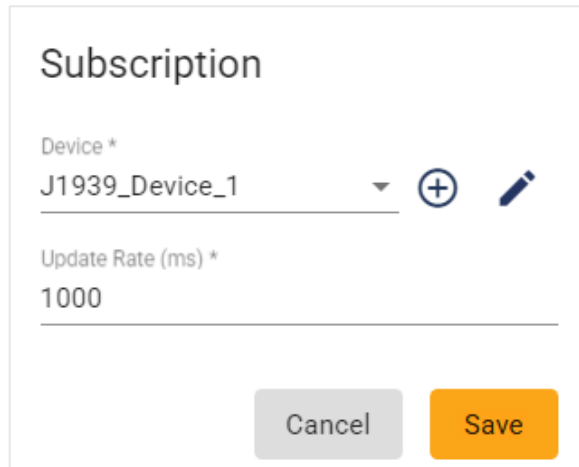
Figure 72: IEC 60870-5-104 Connector - Subscription Configuration View

Parameter	Description	Default Value
Name	Specifies the name of the subscription.	Subscription1
Device	Refers to the IEC 60870-5-104 device from which data will be read or to which data will be written.	

Table 45: IEC 60870-5-104 Connector - Subscription Configuration Parameters

5.2.6.8. J1939

Click **New Subscription** from the left panel of the **Tag Configuration** page to add a subscription to an J1939 connector. A configuration window will open, allowing you to define the subscription parameters.



The image shows a 'Subscription' configuration window. It has a title 'Subscription'. Below the title, there are two input fields. The first is labeled 'Device *' and contains the text 'J1939_Device_1'. To the right of this field are two icons: a blue circle with a white plus sign and a blue pencil icon. The second input field is labeled 'Update Rate (ms) *' and contains the text '1000'. At the bottom of the window, there are two buttons: a grey 'Cancel' button and an orange 'Save' button.

Figure 73: J1939 Connector - Subscription Configuration View

Parameter	Description	Default Value
Device	Specifies the J1939 device from which data will be read or to which data will be written.	
Update Rate (ms)	Frequency of data read requests in milliseconds. In On Data Changes mode, this defines the maximum rate	

	at which data change notifications are sent to the client's callback. Lower values increase update frequency, while higher values reduce communication overhead.	
--	--	--

Table 46: J1939 Connector - Subscription Configuration Parameters

5.2.6.9. Modbus

The **Modbus Connector** requires the same core parameters as other connectors. In addition, it supports configuration of parameters specific to Modbus functionality.

Parameter	Description	Default Value
Advanced Configuration		
Allow Incoming Write Requests	Enables write requests from the Destination connector to the Source. This option is only available when the connector is configured as Source .	Checked

Table 47: Modbus Connector Additional Configuration Parameters

Click **New Subscription** in the left panel of the **Tag Configuration** page to add a subscription to a Modbus connector. A new window will open where you can configure the subscription settings.

Add Subscription

Name *
Subscription1

Device *
Modbus_Device_1

Update Rate (ms) *
1000

Publish Mode *
Synchronous

Cancel Save

Figure 74: Modbus Connector - Subscription Configuration View

Parameter	Description	Default Value
Name	Specifies the name of the subscription.	Subscription1
Device	Refers to the Modbus device from which data will be read or to which data will be written.	
Update Rate (ms)	<p>Defines the frequency of data read requests. In On Data Change mode, this determines the maximum rate at which data change notifications are sent to the client's callback.</p> <p>This parameter is configurable when the connector is configured as a Source.</p>	1000

<i>Publish Mode</i>	<p>Specifies the data publishing method supported by Modbus devices:</p> <ul style="list-style-type: none"> • On Data Change: Data is transmitted only when a value changes within the specified update rate. • Synchronous: Uses synchronous polling to collect data from the Modbus server. <p>This parameter is configurable when the connector is configured as a Source.</p>	Synchronous
----------------------------	--	-------------

Table 48: Modbus Connector - Subscription Configuration Parameters

(!) Note:

For **Source Modbus Connectors**, you can configure multiple subscriptions, enabling simultaneous communication with multiple devices or data points.

(!) Note:

The connector natively supports **decimal register address types**. If a register address is provided in **hexadecimal format**, it must first be converted to decimal. After conversion, the connector will interpret the address correctly as a decimal and recognize the corresponding register.

When downloading a template to configure multiple subscriptions for a Modbus Source Connector, the following fields are available for configuration:

Field	Description
<i>Device</i>	Specifies the Modbus device associated with the subscription.
<i>Subscription_Name</i>	Defines the name of the subscription.
<i>Update_Rate</i>	Frequency (in milliseconds) at which data read requests are sent.

<i>Publish_Mode</i>	Data publishing method. Options include On Data Change or Synchronous .
<i>TagName</i>	The name assigned to the tag for identification purposes.
<i>Address_Type</i>	Type of register address (e.g., decimal). Hexadecimal addresses must be converted to decimal.
<i>AccessRight</i>	Defines whether the tag is readable, writable, or both.
<i>Type</i>	Specifies the data type of the tag (e.g., integer, float).
<i>Address</i>	Register address of the tag in the device.
<i>Bit_Range</i>	Defines the bit range when working with bit-level data.
<i>String_Byte_Swap</i>	Option to swap bytes within a string value.
<i>Dword_Swap</i>	Option to swap double word values.
<i>Word_Swap</i>	Option to swap word values.
<i>Byte_Swap</i>	Option to swap byte order within the data.
<i>Bit_Order_Swap</i>	Option to reverse the order of bits.

Table 49: Modbus Connector - Subscription Template Fields

5.2.6.10. MQTT

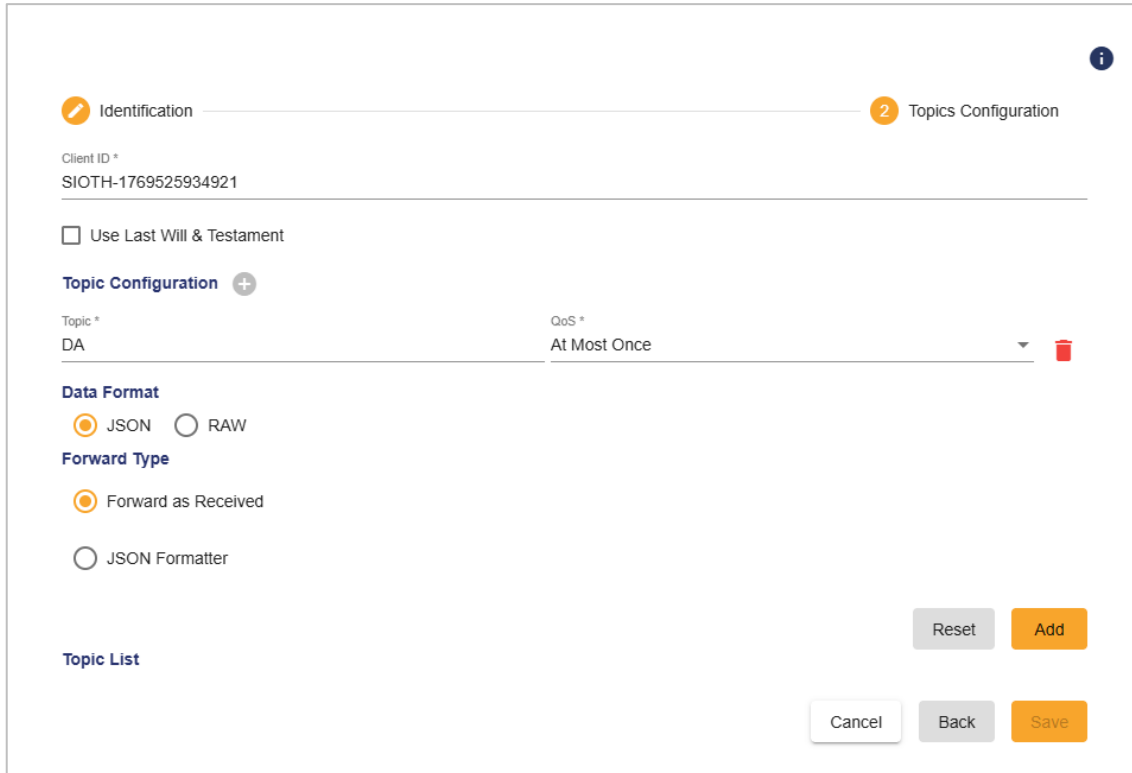
The **MQTT Connector** requires the same core parameters to be configured as other connectors. Once the connector identification step is completed and validated, you are redirected to the **Topics Configuration** page, where the topics used for data exchange are defined.

The topics configuration process depends on the selected MQTT specification:

- **Standard MQTT:** Topics are defined manually, allowing flexible and custom topic structures.

- **Sparkplug MQTT:** Topics follow the **Sparkplug B specification**, which standardizes message formats and topic hierarchies to ensure interoperability with Sparkplug-compliant devices and brokers.

MQTT Standard Connector as Source:



The screenshot shows the 'Topics Configuration' view of the MQTT Standard Connector configuration. It features a progress bar at the top with 'Identification' (1) and 'Topics Configuration' (2). The 'Client ID' is set to 'SIOTH-1769525934921'. There is a checkbox for 'Use Last Will & Testament'. The 'Topic Configuration' section includes a '+' icon, a 'Topic' field with 'DA', and a 'QoS' dropdown menu set to 'At Most Once'. The 'Data Format' section has radio buttons for 'JSON' (selected) and 'RAW'. The 'Forward Type' section has radio buttons for 'Forward as Received' (selected) and 'JSON Formatter'. At the bottom, there is a 'Topic List' label and buttons for 'Reset', 'Add', 'Cancel', 'Back', and 'Save'.

Figure 75: MQTT Standard Connector as Source - Topics Configuration View

Parameter	Description	Default Value
<i>Client ID</i>	A unique identifier that distinguishes each MQTT client connecting to the broker.	SIOTH-XXXXXXXXXXXXXX
<i>Last Will & Testament</i>		

<i>Use Last Will & Testament</i>	Enables the Last Will and Testament (LWT) feature, allowing the broker to notify other clients when the connector disconnects unexpectedly.	Unchecked
<i>Topic</i>	Topic on which the last will message is published.	StoreForward
<i>Message</i>	Message content published when the client disconnects ungracefully.	Offline
<i>QOS</i>	<p>Quality of Service level used to deliver the last will message:</p> <ul style="list-style-type: none"> • At Most Once: Best-effort delivery; message loss may occur. This level is suitable, for example, for ambient sensor data where occasional message loss is acceptable. • At Least Once: Guaranteed delivery, but duplicates may occur. • Exactly One: Guaranteed single delivery, suitable for critical use cases, such as billing systems, where lost or duplicated messages could lead to inconsistencies. 	At Most Once
<i>Last-Will Retain</i>	<p>Determines whether the broker retains the last published message on the topic or not:</p> <ul style="list-style-type: none"> • Checked: The retained message is stored and sent to new subscribers immediately. • Unchecked: Messages are discarded if no subscribers are present. 	Unchecked
<i>Use State</i>	Specifies whether the connector publishes its operational state to the MQTT broker or not:	Unchecked

	<ul style="list-style-type: none"> • Checked: Publishes <i>online</i> when running and <i>offline</i> when stopped. • Unchecked: No state messages are published. 	
Topic Configuration		
Topic	UTF-8 string defining the logical channel for message distribution. Topics consist of one or more hierarchical levels separated by forward slashes (/).	
QoS	<p>Defines the reliability level for message delivery:</p> <ul style="list-style-type: none"> • At Most Once: Best-effort delivery; message loss may occur. • At Least Once: Guaranteed delivery, but duplicates may occur. • Exactly One: Guaranteed single delivery, suitable for critical use cases. 	At Most Once
Data Format		
JSON	Defines the payload format used by the connector as JSON format.	Checked
RAW	Defines the payload format used by the connector as RAW format.	Unchecked
Forward Type		
Forward as Received	When enabled, messages are forwarded to the destination exactly as received from the broker, without transformation or reformatting.	Checked
JSON Formatter	Enables formatting and structuring MQTT message payloads into valid JSON objects before sending them	Unchecked

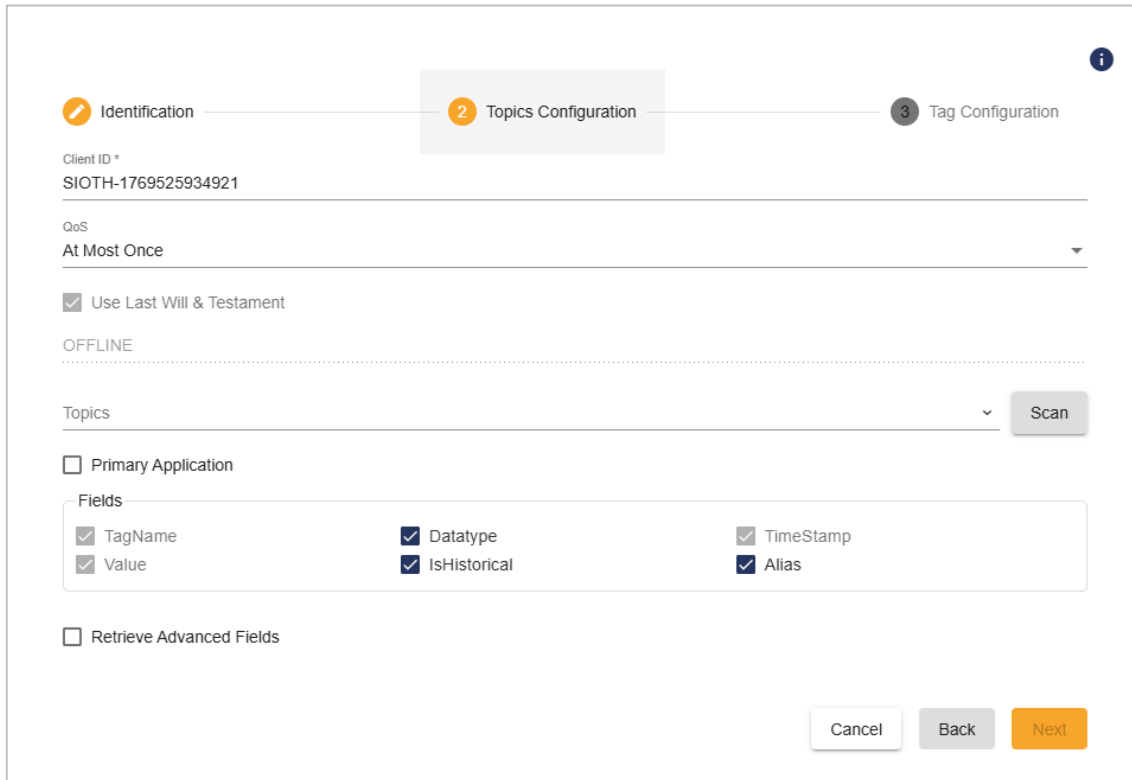
	<p>to the destination. The JSON Formatter includes the following parameters:</p> <ul style="list-style-type: none"> • Include Topic Name: Includes the topic name in the formatted JSON payload (Default: Unchecked). • Variable Configuration: <ul style="list-style-type: none"> - Upload JSON Sample File: Upload a sample JSON file to define JSON paths. - Add Variable: Define variable Name, Value Type, JSON Path, Type and other specific parameters related to the selected Value Type. - Download Template: Download a template including JSON Path, Arithmetic Expression, Datetime Formatting, and Additional Field. - Import Template: Import a predefined variable configuration template. - Export Template: Export the configured JSON mapping for reuse. 	
<i>Text Formatter</i>	<p>Enables formatting and structuring MQTT message payloads into valid Text objects before sending them to the destination. The Text Formatter includes the following parameters:</p> <ul style="list-style-type: none"> • Include Topic Name: Includes the topic name in the formatted RAW payload (Default: Unchecked). 	<i>Text Formatter</i>

	<ul style="list-style-type: none"> • Variable Configuration: <ul style="list-style-type: none"> - Add Variable: Define variable Name, Value Type, Type and other specific parameters related to the selected Value Type. - Download Template: Download a template including Arithmetic Expression, Datetime Formatting, Additional Field, Key Extraction and Regex Extraction. - Import Template: Import a predefined variable configuration template. - Export Template: Export the configured RAW mapping for reuse. 	
Topic List		
Topic List	Displays all configured topics for the connector. Users can view, edit, or delete existing topics.	

Table 50: MQTT Standard Connector as Source - Topics Configuration Parameters

MQTT Sparkplug Connector as Source

For the **MQTT Sparkplug** specification, topics can be discovered automatically by clicking the **Scan** button. When scanning starts, the connector establishes a connection with the MQTT broker and begins retrieving available Sparkplug topics. As topics are detected, they are progressively displayed in the topics list. You may click **Stop** at any time to halt the scanning process. Once the scan is complete, select the required topics by checking them in the displayed list.



Identification — **2 Topics Configuration** — 3 Tag Configuration

Client ID *
SIOTH-1769525934921

QoS
At Most Once

☒ Use Last Will & Testament

OFFLINE

Topics Scan

☐ Primary Application

Fields

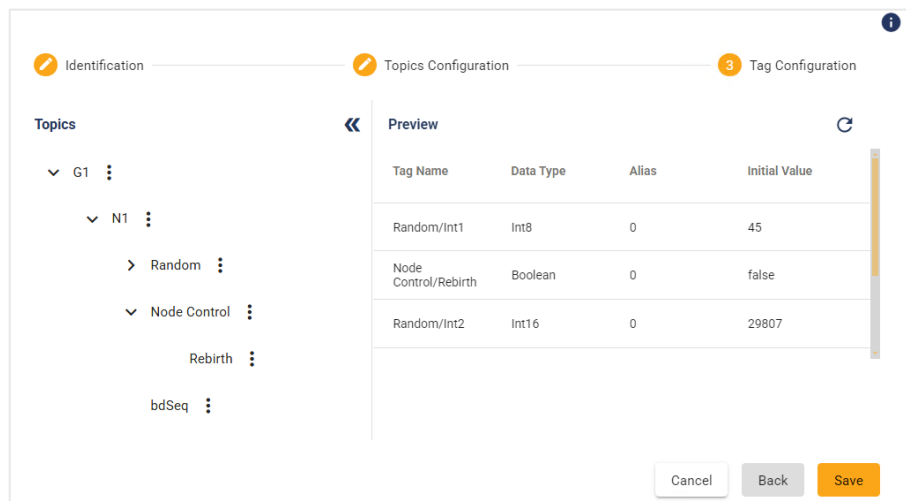
<input checked="" type="checkbox"/> TagName	<input checked="" type="checkbox"/> Datatype	<input checked="" type="checkbox"/> TimeStamp
<input checked="" type="checkbox"/> Value	<input checked="" type="checkbox"/> IsHistorical	<input checked="" type="checkbox"/> Alias

☐ Retrieve Advanced Fields

Cancel Back Next

Figure 76: MQTT Sparkplug Connector as Source - Topics Configuration View

Click **Next**. The **Tag Configuration** page will be displayed, showing the list of browsed tags configured during the previous step.



Identification — Topics Configuration — **3 Tag Configuration**

Topics

- ▼ G1
 - ▼ N1
 - > Random
 - ▼ Node Control
 - Rebirth
 - bdSeq

Preview

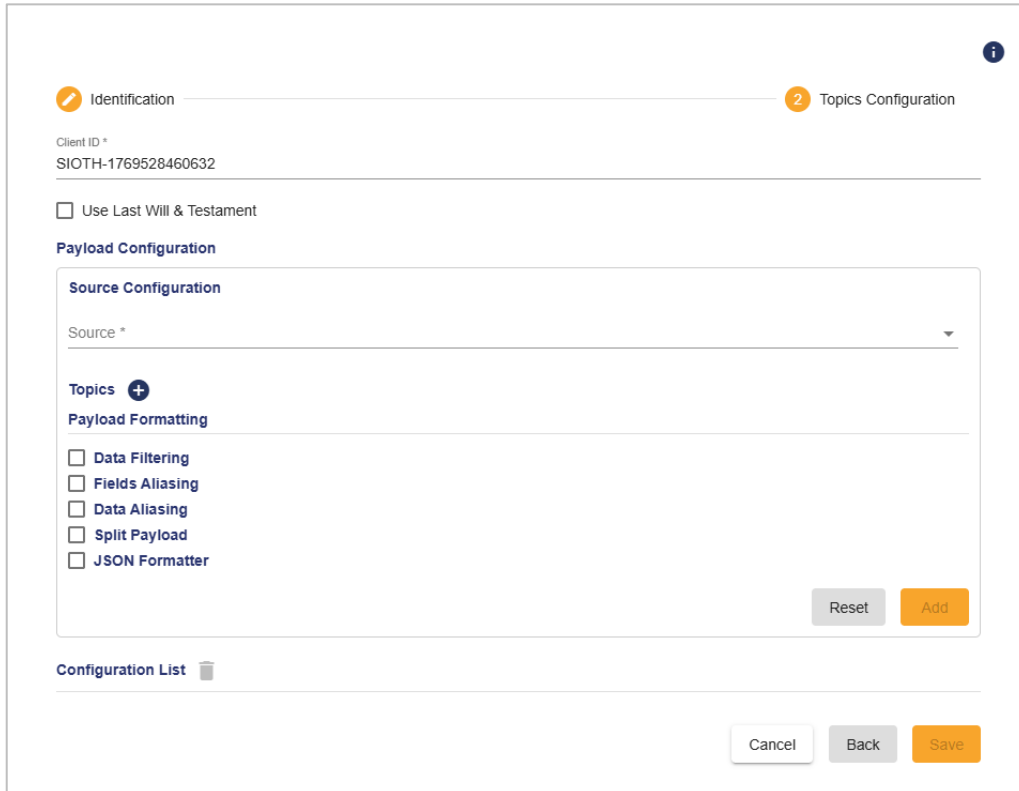
Tag Name	Data Type	Alias	Initial Value
Random/Int1	Int8	0	45
Node Control/Rebirth	Boolean	0	false
Random/Int2	Int16	0	29807

Cancel Back Save

Figure 77: MQTT Sparkplug Connector as Source - Tags Configuration View

Click **Save** to commit the changes.

MQTT Standard Connector as Destination:



The screenshot shows the 'Topics Configuration' view for an MQTT Standard Connector. It features a progress bar at the top with 'Identification' (1) and 'Topics Configuration' (2). The 'Client ID' is set to 'SIOTH-1769528460632'. There is a checkbox for 'Use Last Will & Testament'. The 'Payload Configuration' section includes a 'Source Configuration' dropdown, a 'Topics' section with a plus icon, and a 'Payload Formatting' section with checkboxes for 'Data Filtering', 'Fields Aliasing', 'Data Aliasing', 'Split Payload', and 'JSON Formatter'. At the bottom right of the configuration area are 'Reset' and 'Add' buttons. At the very bottom are 'Cancel', 'Back', and 'Save' buttons. A 'Configuration List' icon is also visible.

Figure 78: MQTT Standard Connector as Destination - Topics Configuration View

Parameter	Description	Default Value
<i>Client ID</i>	Unique identifier that distinguishes each MQTT client connecting to the broker.	SIOTH-XXXXXXXXXXXXXX
<i>Last Will & Testament</i>		

<i>Use Last Will & Testament</i>	Enables the Last Will and Testament (LWT) feature, allowing the broker to notify other clients when the connector disconnects unexpectedly.	Unchecked
<i>Topic</i>	Specifies the topic on which the last will message is published.	StoreForward
<i>Message</i>	Defines the message content published if the client disconnects ungracefully.	Offline
<i>QoS</i>	<p>Defines the reliability level for message delivery:</p> <ul style="list-style-type: none"> • At Most Once: Best-effort delivery; message loss may occur. • At Least Once: Guaranteed delivery, but duplicates may occur. • Exactly One: Guaranteed single delivery, suitable for critical use cases. 	At Most Once
<i>Use Last-Will Retain</i>	<p>Determines whether the broker retains the last published message on the topic:</p> <ul style="list-style-type: none"> • Checked: The retained message is stored and delivered to new subscribers immediately. • Unchecked: The message is discarded if no subscribers are present at publish time. 	Unchecked
<i>Payload Configuration</i>		
<i>Source Configuration</i>		

Source	Specifies the source connector from which data is received.	
Topics		
Topic	UTF-8 string defining the logical channel for message distribution. Topics may include multiple hierarchical levels separated by forward slashes (/).	
QoS	<p>Defines the reliability level for message delivery between the client and the broker:</p> <ul style="list-style-type: none"> • At Most Once: Best-effort delivery; message loss may occur. • At Least Once: Guaranteed delivery with possible duplicates. • Exactly Once: Guaranteed single delivery. 	At Most Once
Use Last-Will Retain	<p>Determines whether the broker retains the last published message on the topic:</p> <ul style="list-style-type: none"> • Checked: The retained message is stored and delivered to new subscribers immediately. • Unchecked: The message is discarded if no subscribers are present at publish time. 	Unchecked
Payload Formatting		

<i>Data Filtering</i>	Enables filtering of received data using templates. Users can download, import, or export filter templates.	Unchecked
<i>Fields Aliasing</i>	Enables renaming of fields or attributes within the payload for clarity or compatibility. Displays existing field-to-alias mappings.	Unchecked
<i>Data Aliasing</i>	Enables defining alternate names or references for data elements in the payload. Users can download, import, or export aliasing templates.	Unchecked
<i>Split Payload</i>	Allows large payloads to be divided into smaller segments for more efficient processing or transmission.	Unchecked
<i>JSON Formatter</i>	Ensures JSON payloads are properly structured and formatted. Users can configure variables in JSON format or use the Payload Builder to manually define a JSON payload.	Unchecked
<i>Configuration List</i>	Displays all configured topics for the connector, allowing users to view, edit, or delete existing configurations.	

Table 51: MQTT Standard Connector as Destination - Topics Configuration Parameters

MQTT Sparkplug Connector as Destination

1 Identification

Client ID *

SIOTH-1769529661665

QoS

At Most Once

☒ Use Last Will & Testament

OFFLINE

SparkPlug Version *

spBv1.0

Node ID *

Source *

2 Topics Configuration

Group ID *

Host ID

Cancel Back Save

Figure 79: MQTT Sparkplug Connector as Destination - Topics Configuration View

Parameter	Description	Default Value
Client ID	A unique identifier that distinguishes each MQTT client connecting to the broker.	SIOTH-XXXXXXXXXXXXXX
QoS	<p>Defines the reliability level for message delivery between the client and the broker:</p> <ul style="list-style-type: none"> At Most Once: Best-effort delivery; message loss may occur. At Least Once: Guaranteed delivery with possible duplicates. 	At Most Once

	<ul style="list-style-type: none"> • Exactly Once: Guaranteed single delivery. 	
<i>Use Last-Will Retain</i>	<p>Determines whether the broker retains the last published message on the topic:</p> <ul style="list-style-type: none"> • Checked: The retained message is stored and delivered to new subscribers immediately. • Unchecked: The message is discarded if no subscribers are present at publish time. 	Checked
<i>Sparkplug Version</i>	Specifies the Sparkplug protocol version used for communication between MQTT clients and the broker.	spBv1.0
<i>Group ID</i>	Identifies the Sparkplug group to which the node belongs, enabling logical organization of MQTT messages within a group context.	
<i>Node ID</i>	A unique identifier representing the Sparkplug node within the specified group. The node publishes its metrics under this identifier.	
<i>Host ID</i>	Identifies the host system or device running the Sparkplug node, allowing association of published data with its originating host.	
<i>Source</i>	Specifies the source connector from which data is received.	
<i>Device Name</i>	Defines the name of the Sparkplug device used when publishing metrics.	
<i>Retain</i>	Determines whether the broker retains the last published message for a topic:	Unchecked

	<ul style="list-style-type: none"> • Checked: The broker stores the last message and delivers it immediately to new subscribers. • Unchecked: Messages are discarded if no subscribers are present at the time of publication. 	
Field Mapping	Enables mapping between source fields and Sparkplug metrics, defining how incoming data is translated into Sparkplug-compliant structures.	
Metrics	Defines the individual data points (metrics) published by the Sparkplug device, including metric names, data types, and values.	

Table 52: MQTT Sparkplug Connector as Destination - Topics Configuration Parameters

5.2.6.11. OPC AE

The **OPC AE Connector** must be configured according to the OPC server architecture to ensure correct operation:

- **64-bit Server:** Configure the OPC AE Connector for 64-bit environments to ensure compatibility and optimal performance.
- **32-bit Server:** Use the 32-bit version of the OPC AE Connector, as the 64-bit version is not compatible with 32-bit servers.

The **OPC AE Connector** is available **only as a Source**.

The **Subscription Configuration** section allows you to manage subscriptions and related tags. It provides several options for field selection, enabling flexible configuration and management of alarm and event data.

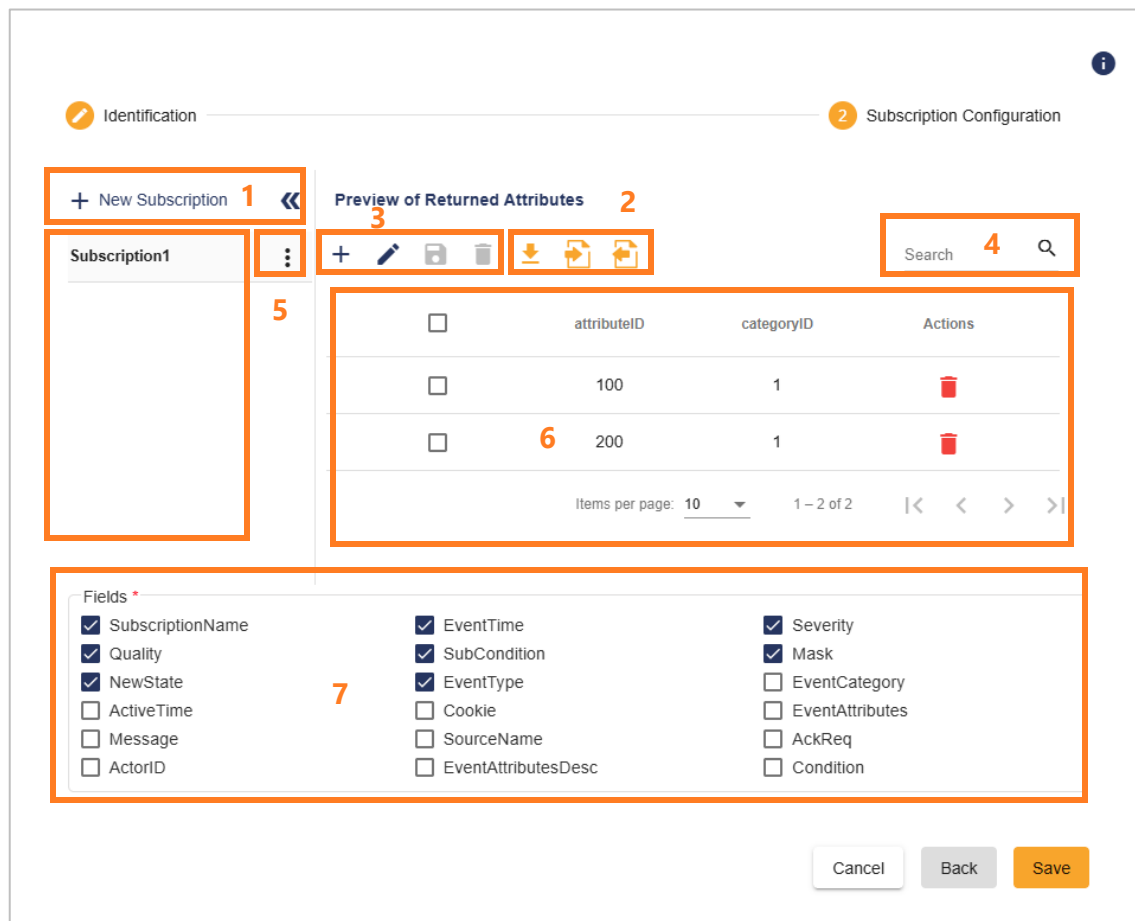
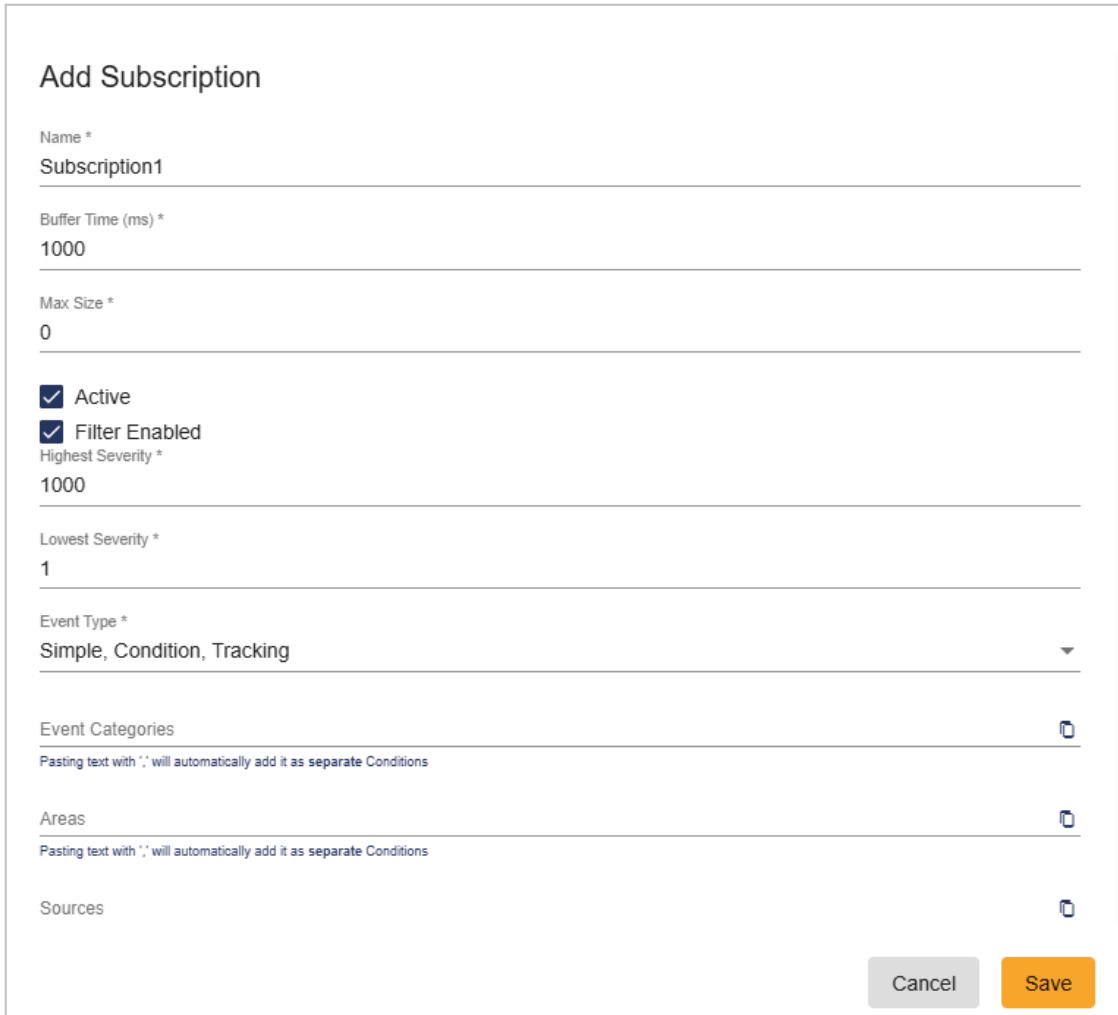


Figure 80: OPC AE Connector as Source - Subscription Configuration View

The following actions are available in the Subscription Configuration view:

- **New Subscription (1):** Create a new subscription.
- **Download Template (2):** Download a tag template for reference.
- **Import (2):** Import tags using a CSV file.
- **Export (2):** Export configured tags for verification or backup purposes.
- **Add, Edit, Save, and Delete (3):** Manage tags by adding, editing, saving, or removing entries.
- **Search (4):** Search for a specific tag using the search field.
- **Edit, Delete Subscription (5):** Modify or delete an existing subscription.
- **Preview Imported Tags (6):** Preview imported tags before applying them.
- **Fields (7):** Configure fields associated with all tags.

Click **New Subscription** from the left panel of the **Subscription Configuration** page to add a subscription to an OPC AE connector. A configuration window will open, allowing you to define the subscription parameters.



Add Subscription

Name *
Subscription1

Buffer Time (ms) *
1000

Max Size *
0

☒ Active
☒ Filter Enabled
Highest Severity *
1000

Lowest Severity *
1

Event Type *
Simple, Condition, Tracking

Event Categories

Pasting text with ',' will automatically add it as separate Conditions

Areas

Pasting text with ',' will automatically add it as separate Conditions

Sources

Cancel Save

Figure 81: OPC AE Connector as Source - Subscription Configuration View

Parameter	Description	Default Value
Name	Specifies the name of the subscription.	Subscription1
Buffer Time (ms)	Defines how frequently (in milliseconds) the server sends event notifications to the client.	1000

Max Size	Specifies the maximum number of events sent in a single callback. A value of 0 indicates no limit. When a value greater than 0 is configured, the server may invoke the callback more frequently than the defined Buffer Time during high event rates, limiting the number of events per callback to the specified value.	0
Active	Indicates whether the subscription is active: <ul style="list-style-type: none"> TRUE (Checked): The subscription is created as active. FALSE (Unchecked): The subscription is created as inactive. 	Checked
Filter		
Filter Enabled	Enables filtering of alarm and event data. When enabled, only events matching the defined criteria are received.	Unchecked
Highest Severity	Defines the upper bound of the severity range. A value of 1000 represents the highest severity. Events with a severity greater than or equal to this value are received.	1000
Lowest Severity	Defines the lower bound of the severity range. A value of 1 represents the lowest severity. Events with a severity less than or equal to this value are received.	1
Event Type	Specifies the types of events to monitor. Multiple options can be selected: <ul style="list-style-type: none"> Simple Condition 	Simple, Condition, Tracking

	<ul style="list-style-type: none"> Tracking 	
Event Categories	Defines the event categories supported by the OPC AE server.	
Areas	Logical groupings used to organize alarms and events, such as by physical location or system type (e.g., Boiler Room, Pump Station).	
Sources	Specifies the origin of alarms or events, typically representing devices or equipment (e.g., Pump1, Boiler1).	

Table 53: OPC AE Connector as Source - Subscription Configuration Parameters

5.2.6.12. OPC DA

The **OPC DA Connector** must be configured according to the OPC server architecture to ensure proper operation:

- **64-bit Server:** Configure the OPC DA Connector for 64-bit environments to ensure compatibility and optimal performance.
- **32-bit Server:** Use the 32-bit version of the OPC DA Connector, as the 64-bit version is not compatible with 32-bit OPC DA servers.

The **OPC DA Connector** requires the same core parameters as other connectors and provides additional configuration options specific to OPC DA functionality.

Parameter	Description	Default Value
Advanced Configuration		
Allow Incoming Write Requests	Enables write requests from Destination connectors to be forwarded to the Source connector.	Checked
Enable Sending Write Requests to Source	Enables write requests to be sent to Source connectors.	Checked

Table 54: OPC DA Connector - Additional Configuration Parameters

Click **New Subscription** from the left section in the **Tag Configuration** page to add a subscription to an OPC DA connector. A configuration window will open, allowing you to define the subscription parameters.

Source Connector

Add Subscription
Name *
Subscription1
Update Rate (ms) *
1000
Dead Band (%) *
0
Time Bias (min) *
0
Read Mode *
OnDataChange
Cancel Save

Destination Connector

Add Subscription
Name *
Subscription1
Update Rate (ms) *
1000
Dead Band (%) *
0
Time Bias (min) *
0
Write Mode *
Synchronous
Cancel Save

Figure 82: OPC DA Connector as Source / Destination - Subscription Configuration View

Parameter	Description	Default Value
Name	Specifies the name of the subscription.	Subscription1
Update Rate (ms)	Defines the frequency of data read requests. In OnDataChange mode, this value represents the maximum rate at which data change notifications are sent to the client callback.	1000
Dead Band (%)	Specifies the percentage range within which a tag value can change without being reported. Applying a deadband reduces insignificant data updates and conserves system resources.	0
Time Bias (min)	Adjusts for time zone differences between the device collecting the data and the client receiving it.	0
When the OPC DA Connector is set as Source		
Read Mode	<p>Defines how data is read from the OPC DA server:</p> <ul style="list-style-type: none"> • OnDataChange: Data is transmitted only when a value change occurs within the specified update rate. • Synchronous: Data is collected using a polling mechanism. • Asynchronous: Data is collected using an asynchronous polling mechanism. 	OnDataChange
When the OPC DA Connector is set as Destination		
Write Mode	Defines how data is written to the OPC DA server:	Synchronous

	<ul style="list-style-type: none"> • Synchronous: The client waits for confirmation before proceeding with the next write, ensuring reliability. • Asynchronous: Write requests are sent without waiting for confirmation, improving performance. • SynchronousIO2: Optimized synchronous write mode for OPC DA 2.0-compliant servers. • AsynchronousIO3: Enhanced asynchronous write mode for OPC DA 3.0-compliant servers, allowing concurrent writes. • SynchronousIO2VQT: Like SynchronousIO2, including Value, Quality, and Timestamp (VQT) information. • AsynchronousIO3VQT: Like AsynchronousIO3, including Value, Quality, and Timestamp (VQT) information. 	
--	--	--

Table 55: OPC DA Connector - Subscription Configuration Parameters

5.2.6.13. OPC HDA

The **OPC HDA Connector** must be configured according to the OPC server architecture to ensure correct operation and optimal performance:

- **64-bit Server:** Configure the OPC HDA Connector for 64-bit environments to ensure compatibility and optimal performance.
- **32-bit Server:** Use the 32-bit version of the OPC HDA Connector, as the 64-bit version is not compatible with 32-bit systems.

Click **New Subscription** from the left section in the **Tag Configuration** page to add a subscription to an OPC HDA connector. A new window will open where you can configure the subscription settings.

Add Subscription

Name *
Subscription1

Read Mode *
Async Read Raw

Start Time *
1/28/2026, 14:49:08

End Time
☒ 1/28/2026, 14:59:08

Values Number *
0

☐ Bounds

☐ Restart from the Latest Executed Time

Loop Period (ms) *
10000

Data Interval (ms) *
10000

Cancel Save

Figure 83: OPC HDA Connector as Source - Subscription Configuration View

Parameter	Description	Default Value
Name	Specifies the name of the subscription.	Subscription1
Read Mode	<p>Defines the data retrieval mode supported by OPC HDA:</p> <ul style="list-style-type: none"> Sync Read Raw: Retrieves raw historical data synchronously and waits for the server response. 	Async Read Raw

	<ul style="list-style-type: none"> • Sync Read Processed: Retrieves processed (aggregated) historical data synchronously. • Async Read Raw: Retrieves raw historical data asynchronously; the client is notified when data is available. • Async Read Processed: Retrieves processed historical data asynchronously. • Async Advise Raw: Subscribes to raw historical data updates and receives notifications when new data is available. • Async Advise Processed: Subscribes to processed historical data updates and receives notifications when new data is available. 	
Start Time	Defines the start of the time range for historical data retrieval.	Current time – 10 minutes
End Time	Defines the end of the time range for historical data retrieval.	Current time
Values Number	Specifies the maximum number of values to be returned per item within the defined time range. A value of 0 indicates no limit.	0
Bounds	When enabled, bounding values are included in the results.	Unchecked
Restart from the Last Executed Time	Resumes historical data processing from the point where the previous execution stopped.	Unchecked

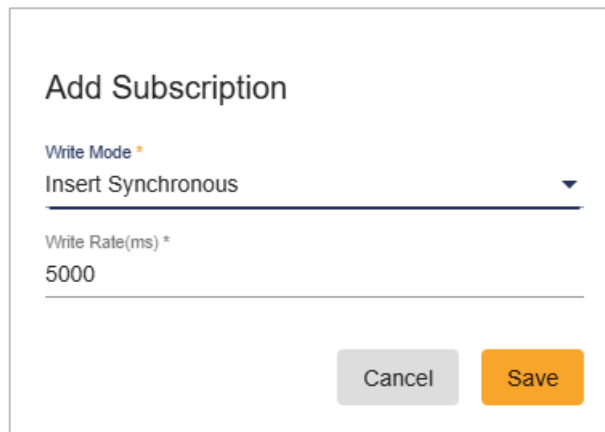
Loop Period (ms)	Defines the execution interval of the data retrieval loop.	10000
Data Interval (ms)	Specifies how frequently new data is transmitted.	10000
Update Interval (ms)	Defines the frequency at which updated data is sent.	10000
Resample Interval (ms)	Specifies the interval between returned data values.	10000
Interval Number	Defines the number of resample intervals between consecutive data updates.	1
Aggregate	<p>Specifies the aggregation functions applied when using processed read modes (Sync Read Processed, Async Read Processed, and Async Advise Processed). Supported functions include:</p> <ul style="list-style-type: none"> • Interpolative: Estimates values at specified timestamps using interpolation. • Total: Computes the sum of all values within the time range. • Average: Calculates the arithmetic mean of values within the time range. • Time Average: Computes the weighted average based on time intervals. • Count: Counts the number of valid data points within the time range. 	Interpolative

	<ul style="list-style-type: none"> • Stdev: Calculates the standard deviation to measure variability. • Minimum Actual Time: Returns the timestamp of the minimum value. • Minimum: Retrieves the smallest recorded value. • Maximum Actual Time: Returns the timestamp of the maximum value. • Maximum: Retrieves the highest recorded value. • Start: Returns the first recorded value in the time range. • End: Returns the last recorded value in the time range . • Delta: Computes the difference between the first and last values. • Regslope: Determines the slope of the best-fit regression line. • Regconst: Returns the y-intercept of the regression line. • Regdev: Measures deviation from the regression line. • Variance: Calculates the spread of data values from the average. • Range: Computes the difference between maximum and minimum values. 	
--	---	--

	<ul style="list-style-type: none"> • Duration Good: Measures total duration where data quality was good. • Duration Bad: Measures total duration where data quality was bad. • Percent Good: Percentage of time with good data quality. • Percent Bad: Percentage of time with bad data quality. • Worst Quality: Identifies the worst data quality status within the given range. • Annotation: Retrieves comments or metadata associated with data points. 	
--	--	--

Table 56: OPC HDA Connector as Source - Subscription Configuration Parameters

When configured as a **Destination**, click the **New Subscription** button from the left section in the **Tag Configuration** page to add a subscription to an OPC HDA connector configured as **Destination**. A new window will open where you can configure the subscription settings.



The image shows a dialog box titled "Add Subscription". It contains a "Write Mode" dropdown menu with "Insert Synchronous" selected. Below this is a "Write Rate(ms) *" input field with the value "5000". At the bottom right are "Cancel" and "Save" buttons.

Figure 84: OPC HDA Connector as Destination - Subscription Configuration View

Parameter	Description	Default Value
Write Mode	<p>Defines how historical data is written to the OPC HDA server:</p> <ul style="list-style-type: none"> • Insert Synchronous: Inserts historical data synchronously, ensuring immediate persistence. • Replace Synchronous: Replaces existing historical data synchronously. • Insert Replace Synchronous: Inserts new data or replaces existing data synchronously based on conditions. • Insert Asynchronous: Inserts historical data asynchronously, allowing other operations to continue. • Replace Asynchronous: Replaces existing historical data asynchronously. • Insert Replace Asynchronous: Inserts or replaces data asynchronously while allowing continued server operation. 	Insert Synchronous
Write Rate (ms)	Specifies the interval, in milliseconds, at which historical data is written to the server.	5000

Table 57: OPC HDA Connector as Destination - Subscription Configuration Parameters

5.2.6.14. OPC UA

The **OPC UA Connector** requires the configuration of the same core parameters as other connectors and provides additional settings specific to OPC UA communication and functionality.

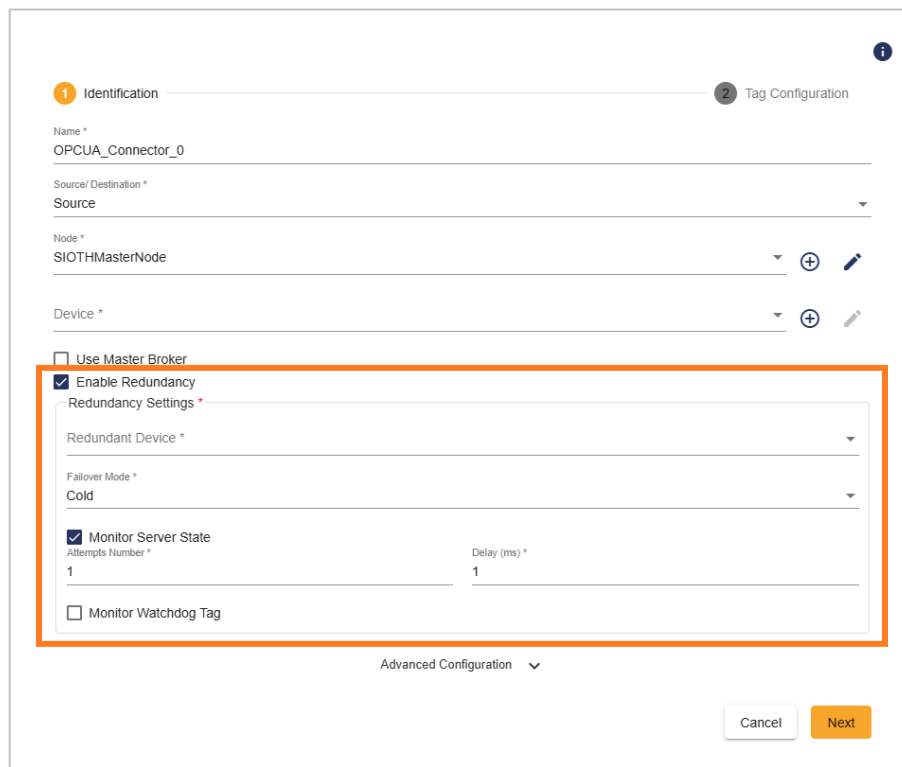


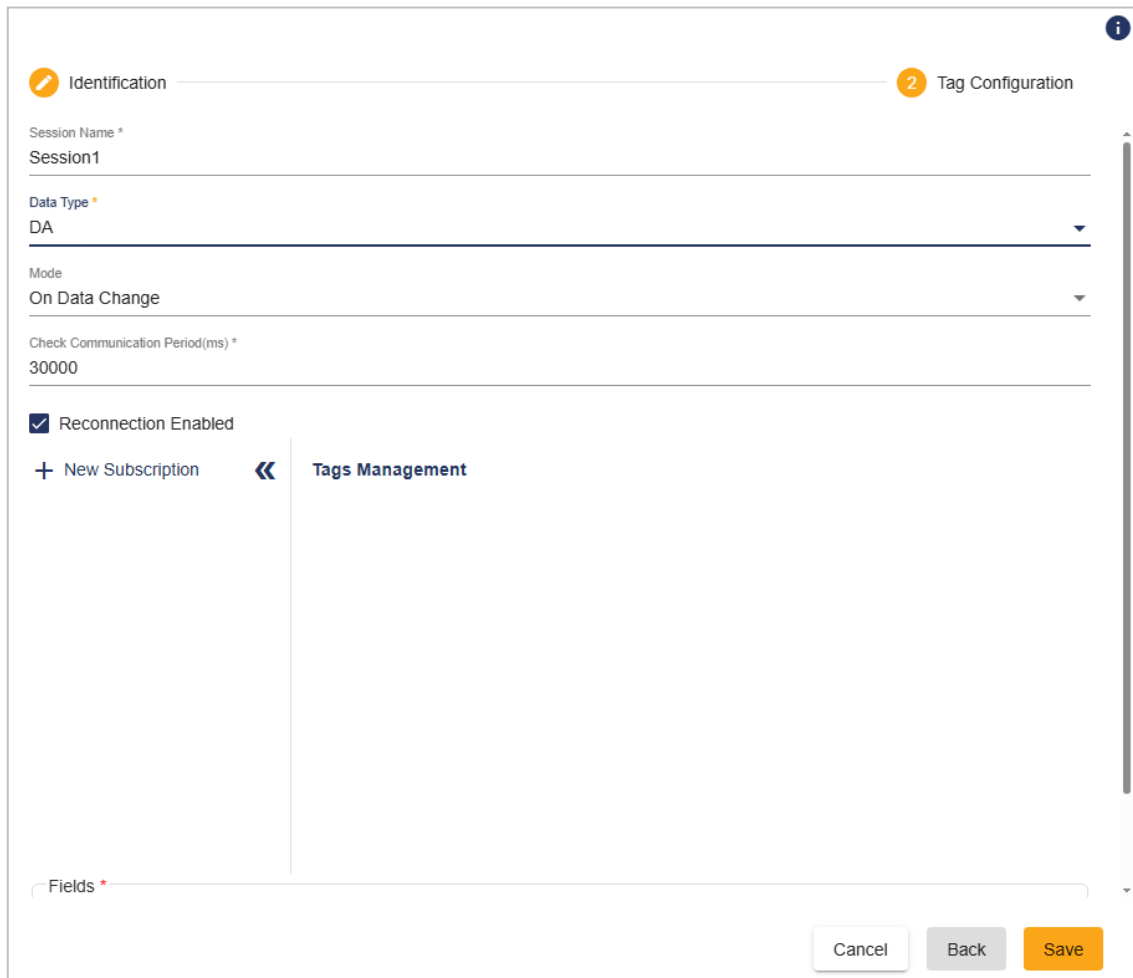
Figure 85: OPC UA Connector - Redundancy Configuration View

Parameter	Description	Default Value
Redundancy Settings		
Enable Redundancy	Enables redundancy by allowing configuration of an additional OPC UA server.	Unchecked
Redundant Device	Selects the redundant OPC UA device from the available list.	
Failure Mode	Defines the redundancy operation mode: <ul style="list-style-type: none"> Cold: The secondary server is activated only if the primary server fails. Hot: The secondary server runs in parallel with the primary server. 	Cold

<i>Failure Mode = Cold</i>		
<i>Monitor Server State</i>	Continuously monitors server availability and switches to the redundant server if the primary server becomes unavailable.	Checked
<i>Attempts Number</i>	Specifies the number of reconnection attempts before switching to the redundant server. Reconnection attempts occur based on the configured delay.	1
<i>Delay (ms)</i>	Time interval (in milliseconds) between reconnection attempts or before switching to the redundant server after a failure is detected.	1
<i>Monitor Watchdog Tag</i>	Enables redundancy monitoring using a watchdog tag value.	Unchecked
<i>Tag Name</i>	Specifies the Item ID of the watchdog tag. The watchdog tag typically updates continuously at a defined interval.	
<i>Timeout (ms)</i>	Maximum time (in milliseconds) to wait for a watchdog tag update before triggering a switchover.	10000
<i>Failure Mode = Hot</i>		
<i>Check Service Level</i>	Enables validation of the OPC UA server's service level to ensure it meets operational requirements.	Checked

Table 58: OPC UA Connector - Additional Configuration Parameters

The **Tag Configuration** section allows you to perform actions related to subscription and tag management. Various options are available for **Field selection**, providing flexibility in how data fields are configured and managed.



The screenshot displays the 'Tag Configuration' view of the OPC UA Connector as Source. The interface is divided into two main sections: 'Identification' and 'Tag Configuration'. The 'Identification' section contains the following fields:

- Session Name ***: Session1
- Data Type ***: DA
- Mode**: On Data Change
- Check Communication Period(ms) ***: 30000
- Reconnection Enabled**: ☒

The 'Tag Configuration' section is currently empty, showing a '+ New Subscription' button and a 'Tags Management' tab. At the bottom right, there are 'Cancel', 'Back', and 'Save' buttons.

Figure 86: OPC UA Connector as Source - Tag Configuration View

Parameter	Description	Default Value
<i>Session Name</i>	Unique identifier for the session established between the OPC UA client and server.	Session1
<i>Data Type</i>	Specifies the type of OPC UA data exchange: <ul style="list-style-type: none"> DA: Data Access HA: Historical Access HE: Historical Events AC: Alarms and Conditions 	DA

Mode	<p>Defines the data acquisition mode:</p> <ul style="list-style-type: none"> • On Data Change (DA & AC): Updates are received only when values change. • Synchronous (HA & HE): Data is retrieved even if no change occurs. 	On Data Change
Check Communication Period (ms)	Interval at which the connector verifies the communication status with the OPC UA server.	30000
Reconnection Enabled	Enables automatic reconnection attempts if the connection is interrupted.	Checked

Table 59: OPC UA Connector as Source - Tag Configuration Parameters

Click **New Subscription** from the left panel of the Tag Configuration page to add a subscription to an OPC UA connector configured as **Source**. The available configuration parameters depend on the selected data type.

- **DA & AC:**

Add Subscription

Name *	Publishing Interval(ms) *
Subscription1	1000
Keep Alive Count *	Life Time Count *
10	1000
Max Notifications per Publish *	Priority *
0	255

Cancel
Save

Figure 87: OPC UA Connector as Source - Subscription Configuration View for DA & AC

Parameter	Description	Default Value
<i>Name</i>	Specifies the subscription name.	Subscription1
<i>Publishing Interval (ms)</i>	Rate at which the server sends notifications to the client.	1000
<i>Keep Alive Count</i>	Number of publishing cycles without notifications before a keep-alive message is sent.	10
<i>Life Time Count</i>	Number of missed publishing cycles before the subscription is deleted by the server.	1000
<i>Max Notifications Per Publish</i>	Maximum number of notifications per published response. A value of 0 indicates no limit.	0
<i>Priority</i>	Subscription priority used by the server to schedule notification delivery.	255

Table 60: OPC UA Connector as Source - Subscription Configuration Parameters for DA & AC

- **HA:**

Add Subscription

Name *

Subscription1

Read Mode *

Read Raw

☒ Absolute Time
 ☐ Relative Time

Start Time *

1/28/2026, 09:30:19

End Time

Max Returned Values *

0

☐ Restart from the Last Executed Time

Waiting Time (ms) *

10000

Resample Interval (ms) *

900000

☐ Read Modified ☒ Include Bounds

Cancel Save

Figure 88: OPC UA Connector as Source - Subscription Configuration View for HA

Parameter	Description	Default Value
<i>Name</i>	Specifies the name of the subscription.	Subscription1
<i>Read Mode</i>	Defines how historical data is retrieved: <ul style="list-style-type: none"> • Read Raw: Retrieves raw historical values directly from the server. • Read Processed: Retrieves processed or aggregated historical values. 	Read Raw

<i>Absolute Time</i>	Uses a specific date and time (for example, 2025-11-04T10:00:00Z) as the reference point for querying historical data.	Checked
<i>Relative Time</i>	Uses a time offset relative to the current time (for example, last 1 hour) instead of a fixed timestamp.	Unchecked
<i>Start Time</i>	Specifies the start of the time range from which historical data is retrieved.	Current Time
<i>End Time</i>	Specifies the end of the time range for the historical data query.	Unchecked
<i>Max Returned Values</i>	Limits the maximum number of historical data points returned by the server in a single response. A value of 0 indicates no limit.	0
<i>Restart from the Last Executed Time</i>	When enabled, resumes data retrieval from the point where the previous execution stopped.	Unchecked
<i>Waiting Time (ms)</i>	Time (in milliseconds) the client waits for a server response before retrying or stopping the request.	10000
<i>Resample Interval (ms)</i>	Defines the interval (in milliseconds) between returned data points. The server interpolates values to match this interval.	900000
<i>Aggregate</i>	Specifies the aggregation function applied to raw historical data before it is returned. This setting is available only when the Read Mode is set to Read Processed . Examples of aggregation functions include Average, Min, Max, Count , and others. Refer	Interpolative

	to the OPC UA Historical Aggregate Functions Table below for a complete list of available functions.	
Processing Interval (ms)	Defines the processing interval used for aggregation. Available only when Read Mode is set to Read Processed .	
Read Modified	When enabled, retrieves records of historical data modifications instead of only original values. Available only when Read Mode is set to Read Raw .	Unchecked
Include Bounds	Includes boundary values (data points exactly at Start Time and End Time) in the results. Available only when Read Mode is set to Read Raw .	Checked
Clean Processed Data	Removes intermediate or duplicate processed values, returning only finalized aggregated results. Available only when Read Mode is set to Read Processed .	Unchecked

Table 61: OPC UA Connector as Source - Subscription Configuration Parameters for HA

Function	Description
Annotation Count <i>Error! Not a valid link.</i>	Returns the total number of annotations within the specified interval.
Average	Calculates the mean value of the data across the specified interval.
Count	Returns the total number of raw data values recorded during the interval.
Delta	Computes the difference between the starting value and the ending value of the interval.

<i>Delta Bounds</i>	Computes the difference between the start bound and end bound values using Simple Bounding Values.
<i>Duration Bad</i>	Calculates the total amount of time within the interval during which the data quality was marked as Bad.
<i>Duration Good</i>	Calculates the total amount of time within the interval during which the data quality was marked as Good.
<i>Duration In State Non Zero</i>	Determines the total time a Boolean or numeric value remained in a non-zero state using Simple Bounding Values.
<i>Duration In State Zero</i>	Determines the total time a Boolean or numeric value remained in a zero-state using Simple Bounding Values.
<i>End</i>	Returns the data value at the end of the interval.
<i>End Bound</i>	Returns the value at the end of the interval based on Simple Bounding Values.
<i>Interpolative</i>	Calculates the value at the start of each interval by interpolating between the surrounding data points.
<i>Maximum</i>	Returns the highest raw value within the interval, timestamped at the start of the interval.
<i>Maximum 2</i>	Returns the maximum value within the interval, including Simple Bounding Values.
<i>Maximum Actual Time</i>	Returns the highest value in the interval along with the actual timestamp at which it occurred.
<i>Maximum Actual Time 2</i>	Returns the maximum value and its actual timestamp, including Simple Bounding Values.

<i>Minimum</i>	Returns the lowest raw value within the interval, timestamped at the start of the interval.
<i>Minimum 2</i>	Returns the minimum value within the interval, including Simple Bounding Values.
<i>Minimum Actual Time</i>	Returns the lowest value in the interval along with the actual timestamp at which it occurred.
<i>Minimum Actual Time 2</i>	Returns the minimum value and its actual timestamp, including Simple Bounding Values.
<i>Number Of Transitions</i>	Returns the number of transitions between zero and non-zero states for a Boolean or numeric value within the interval.
<i>Percent Bad</i>	Calculates the percentage (0–100) of data in the interval that has a Bad status code.
<i>Percent Good</i>	Calculates the percentage (0–100) of data in the interval that has a Good status code.
<i>Range</i>	Calculates the difference between the minimum and maximum values within the interval.
<i>Range2</i>	Calculates the difference between the Minimum2 and Maximum2 values within the interval.
<i>Standard Deviation Population</i>	Computes the population standard deviation (n) for the interval, including Simple Bounding Values.
<i>Standard Deviation Sample</i>	Computes the sample standard deviation (n-1) for the interval.
<i>Start</i>	Returns the data value at the beginning of the interval.

<i>Start Bound</i>	Returns the value at the beginning of the interval based on Simple Bounding Values.
<i>Time Average</i>	Calculates the time-weighted average over the interval using Interpolated Bounding Values.
<i>Time Average 2</i>	Calculates the time-weighted average over the interval using Simple Bounding Values.
<i>Total</i>	Calculates the total (time integral) of the data over the interval using Interpolated Bounding Values.
<i>Total 2</i>	Calculates the total (time integral) of the data over the interval using Simple Bounding Values.
<i>Variance Population</i>	Computes the variance for the interval based on the population standard deviation, including Simple Bounding Values.
<i>Variance Sample</i>	Computes the variance for the interval based on the sample standard deviation.
<i>Worst Quality</i>	Returns the worst data quality StatusCode observed within the interval.
<i>Worst Quality 2</i>	Returns the worst data quality StatusCode observed within the interval, including Simple Bounding Values.

Table 62: OPC UA Historical Aggregate Functions Table

- **HE:**

Add Subscription

Name *	Subscription1	Publishing Interval(ms) *	1000
Keep Alive Count *	10	Life Time Count *	1000
Max Notifications per Publish *	0	Priority *	255
Read Mode *	Read Row Loop		
Start Time *	1/28/2020, 10:03:25	<input type="checkbox"/> End Time	
Max Returned Values *	0		
<input type="checkbox"/> Restart from the Last Executed Time			
Waiting Time (ms) *	10000	Resample Interval (ms) *	900000

Filters

Simple Events

EventId
EventType
SourceName
Time
Severity
Simple Events

Pasting text with ";" will automatically add it as **separate** Conditions

Condition Type

Pasting text with ";" will automatically add it as **separate** Conditions

Audit Event Type

Pasting text with ";" will automatically add it as **separate** Conditions

Custom Attributes

Simple Events

AckedState
AckNote
AckTime
Simple Events

Pasting text with ";" will automatically add it as **separate** Conditions

☐ Alias Mapping
☐ Concatenation

Cancel
Save

Figure 89: OPC UA Connector as Source - Subscription Configuration View for HE

Parameter	Description	Default Value
Name	Specifies the name of the subscription.	Subscription1
Publishing Interval (ms)	Defines the rate at which notifications are sent to the client.	1000

<i>Keep Alive Count</i>	Number of publishing cycles without notifications before a keep-alive message is sent.	10
<i>Life Time Count</i>	Number of cycles without a publish request before the subscription is deleted by the server.	1000
<i>Max Notifications per Publish</i>	Maximum number of notifications per published response. A value of 0 indicates no limit.	0
<i>Priority</i>	Determines the subscription's priority when multiple subscriptions are publishing notifications. Equal priorities are handled using round-robin scheduling.	255
<i>Read Mode</i>	Defines the data reading behavior.	Read Row loop
<i>Start Time</i>	Specifies the beginning of the data retrieval time range.	Current Time
<i>End Time</i>	Specifies the end of the data retrieval time range.	Unchecked
<i>Max Returned Values</i>	Limits the maximum number of returned values per request.	0
<i>Restart from the Last Executed Time</i>	Resumes reading from the last successfully executed time.	Unchecked
<i>Waiting Time (ms)</i>	Timeout period the client waits for a server response.	10000
<i>Resample Interval (ms)</i>	Interval used to interpolate returned data values.	900000
<i>Filters</i>		

<i>Simple Events</i>	Defines the standard event fields to retrieve.	EventId, EventType, SourceName, Time, Severity
<i>Condition Type</i>	Filters events representing conditions whose state can change over time.	
<i>Audit Event Type</i>	Filters audit events related to system or user actions affecting the server.	
<i>Custom Attributes</i>		
<i>Simple Events</i>	Defines vendor-specific custom event attributes to retrieve.	AckedState, AckNote, AckTime
<i>Alias Mapping</i>	Enables mapping event fields to aliases.	Unchecked
<i>Concatenation</i>	Concatenates multiple event field values into a single output.	Unchecked

Table 63: OPC UA Connector as Source - Subscription Configuration Parameters for HE

When configured as a **Destination**, the OPC UA Connector displays a dedicated set of parameters in the **Tag Configuration** view. These parameters are designed to support field mapping and the management of data received from one or more source connectors.

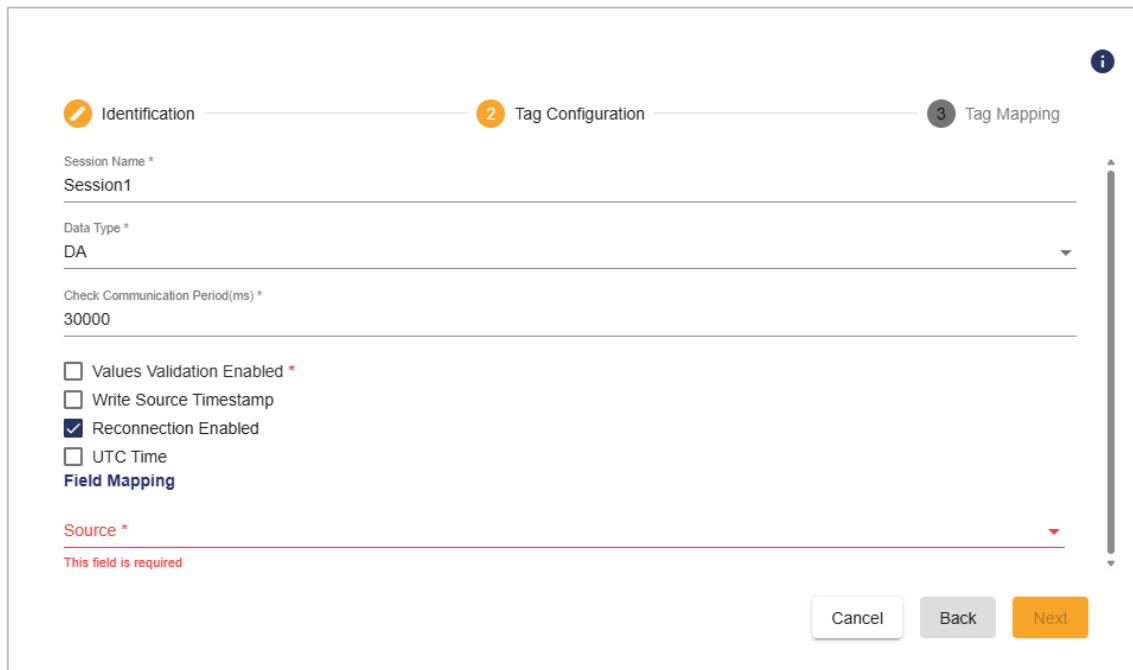


Figure 90: OPC UA Connector as Destination - Tag Configuration View

Parameter	Description	Default Value
Session Name	Specifies a unique identifier for the session established between the OPC UA client and the OPC UA server.	Session1
Data Type	Defines the type of data exchanged between the OPC UA Server and the OPC UA Connector. The supported type is DA (Data Access) .	DA
Mode	Specifies the communication mode used by the connector. Available options: <ul style="list-style-type: none"> Synchronous Asynchronous 	Synchronous

Check Communication Period (ms)	Defines the interval, in milliseconds, at which the client verifies the communication status and connection health with the OPC UA server.	30000
Values Validation Enabled	Enables data validation to ensure integrity and correctness before reading from or writing data to the OPC UA server.	Unchecked
Condition	Defines the condition that determines when data is written to the OPC UA server.	Timestamp Newer Than
Only Send Data Newer Than (ms)	Specifies a time threshold in milliseconds. Data older than this value is not transmitted to the OPC UA server.	1800000
Write Source Timestamp	When enabled, writes the original source timestamp together with the data value to the OPC UA server.	Unchecked
Reconnection Enabled	Enables automatic reconnection attempts if communication with the OPC UA server is lost or interrupted.	Checked
UTC Time	Enables the use of UTC timestamps for data exchange instead of local system time.	Unchecked
Fields Mapping	Defines how fields from source connectors are mapped to corresponding fields in the destination OPC UA connector.	
Source	Displays and allows selection of the source connector whose fields will be mapped to the destination OPC UA connector.	

Table 64: OPC UA Connector as Destination - Tag Configuration Parameters

5.2.6.15.REST

The **REST Connector** enables SIOTH to connect to any REST-compliant web service. It uses the same core identification and validation steps as other connectors.

Once the connector identification is completed and validated, you are redirected to the **Information and Specification** page, where the REST API parameters are defined.

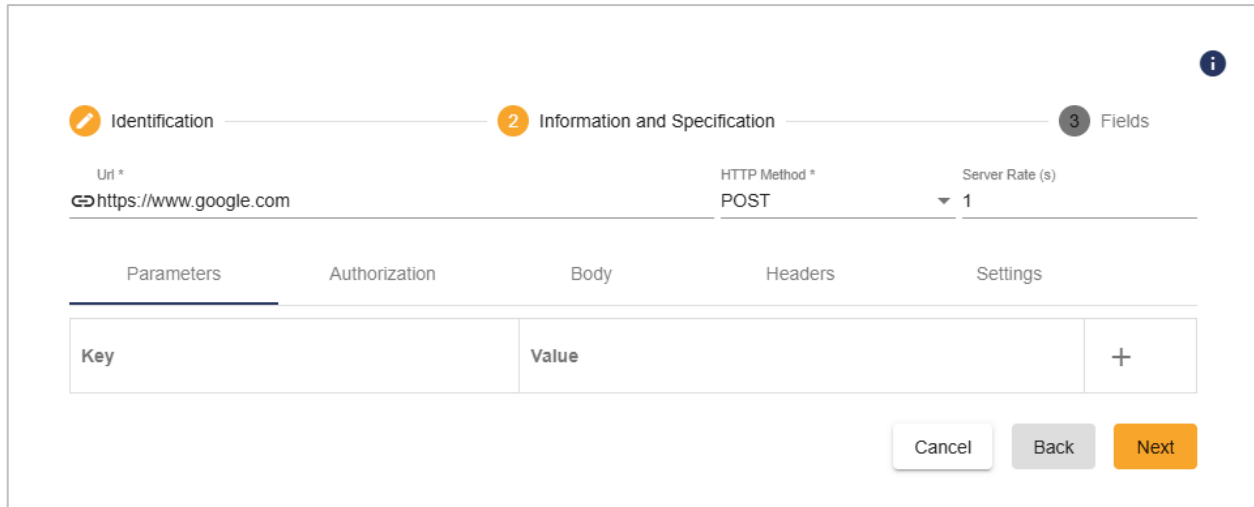


Figure 91: REST Connector - Information and Specification View

Parameter	Description	Default Value
<i>URL</i>	URL of the web service to connect to. The URL must follow the format: https://server_IP_or_HOSTNAME where <i>server_IP_or_HOSTNAME</i> is the IP address or hostname of the server hosting the web service.	https://www.google.com
<i>HTTP Method</i>	HTTP method used to perform the request. Available options: <ul style="list-style-type: none">• POST• GET	POST

	<ul style="list-style-type: none"> • PATCH • OPTIONS • PUT • DELETE 	
Server Rate (s)	Delay, in seconds, between consecutive data read requests. This parameter is available only when the connector is configured as a Source .	1
Parameters	Defines variable components of the REST request (such as query parameters or path parameters) used to specify exchanged data.	
Authorization	<p>Defines the authentication method used to connect to the REST service. Available options include:</p> <ul style="list-style-type: none"> • None: No authentication is required, or authentication is handled through a custom header. • Basic: Username and password-based authentication. When selected, credential encryption option is available. • API Key: Token-based authentication. <ul style="list-style-type: none"> ○ Add to: Defines where to include the key (Header or Query). ○ Key: Name of the key field. ○ Value: API key or token value used for authentication. • Bearer Token: Authorization using a short-lived access token. 	None

	<ul style="list-style-type: none"> • OAuth 2.0: Secure authorization using third-party tokens. <ul style="list-style-type: none"> ○ Access Token URL: Endpoint where the client retrieves an access token. ○ Client ID: Public identifier for the client application. ○ Client Secret: Secret key used to authenticate the client with the authorization server. ○ Scopes: Access permissions requested by the client. • DigestAuth: Challenge-response authentication using hashed credentials (username and password required). • HawkAuth: Lightweight authentication with integrity and replay protection. <ul style="list-style-type: none"> ○ Algorithm: Hashing algorithm that is used (e.g., SHA256). ○ Hawk Auth ID: Identifier for the Hawk credentials. ○ Hawk Auth Key: Shared secret key used to generate message authentication codes. • AWS Signature Version 4: Authentication for AWS IoT Core. <ul style="list-style-type: none"> ○ Add to: Whether to include the signature in the Header or Query. 	
--	---	--

	<ul style="list-style-type: none"> ○ Access Key: AWS access key ID used for authentication. ○ Secret Key: Secret key associated with the AWS access key. ○ AWS Region: Region where the AWS service is hosted. ○ Service Name: Name of the AWS service being accessed. • X.509 Client Certificate: TLS certificate-based authentication. <ul style="list-style-type: none"> ○ Protocol: Communication protocol (TLS 1.0, TLS 1.1, TLS 1.2 or SSL 3). ○ CA Certificate Path: Path to the Certificate Authority (CA) certificate used to verify the server. ○ PFX Key Path: Path to the client's .pfx (PKCS#12) certificate file containing the private key. ○ Password: Password protecting the .pfx file, if applicable. 	
Body	<p>Defines the format of the HTTP request body. Supported options include:</p> <ul style="list-style-type: none"> • Type: <ul style="list-style-type: none"> ○ Raw: Enter a properly formatted JSON body. ○ FormData: Define key-value pairs. • Data Type: JSON 	

<i>Headers</i>	<p>Defines custom HTTP headers used to provide metadata for the API request and response, such as authorization details, content type, caching behavior, and cookies. Proper header configuration is required to ensure successful API communication.</p>	
<i>Settings</i>	<p>Allows configuration of proxy settings. Available options include:</p> <ul style="list-style-type: none"> • Use Proxy Authentication: Enables authentication for the proxy server. When enabled, a username and password must be provided. • Use the System Proxy: Uses the proxy configuration defined at the operating system level. • Respect HTTP_PROXY, HTTPS_PROXY, NO_PROXY environment variables: Applies proxy settings defined through standard environment variables. • Add a custom proxy configuration: Allows manual definition of proxy settings: <ul style="list-style-type: none"> ○ HTTP / HTTPS: Select the protocol used by the proxy. ○ Proxy Custom Server: Hostname or IP address of the proxy server. ○ Proxy Custom Server Port: Port used to connect to the proxy server. 	

	<ul style="list-style-type: none"> Proxy Auth: Username and password required for authenticated proxy access. 	
--	--	--

Table 65: REST Connector - Information and Specification Parameters

Click **Next** when the **Information and Specification** parameters are configured. The next configuration page depends on the type of connector.

REST Connector as Source: The **Fields** Configuration view allows to define the fields to be extracted from the REST response, including any fields that require a specific datetime format.

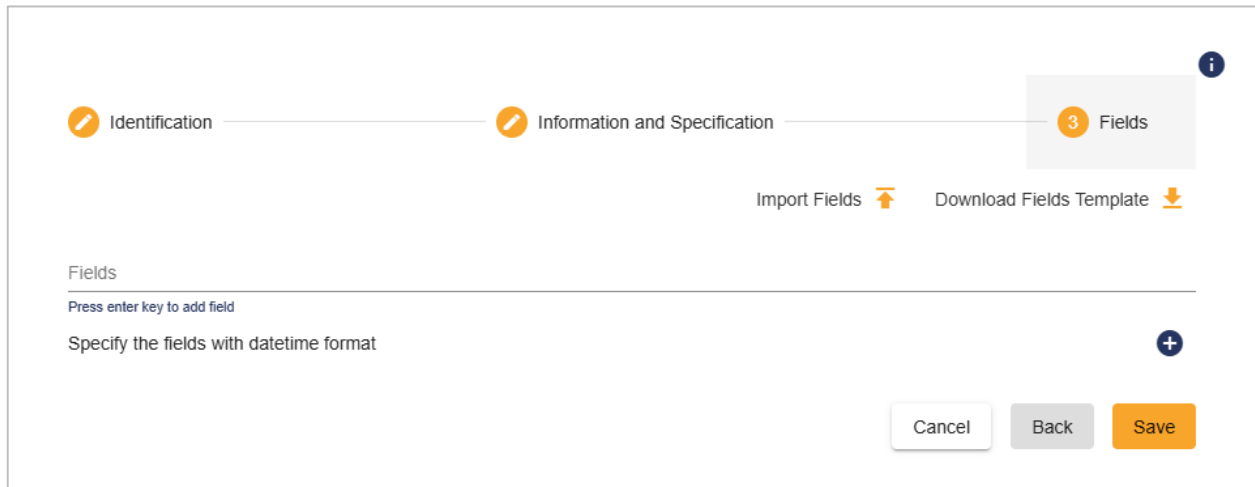
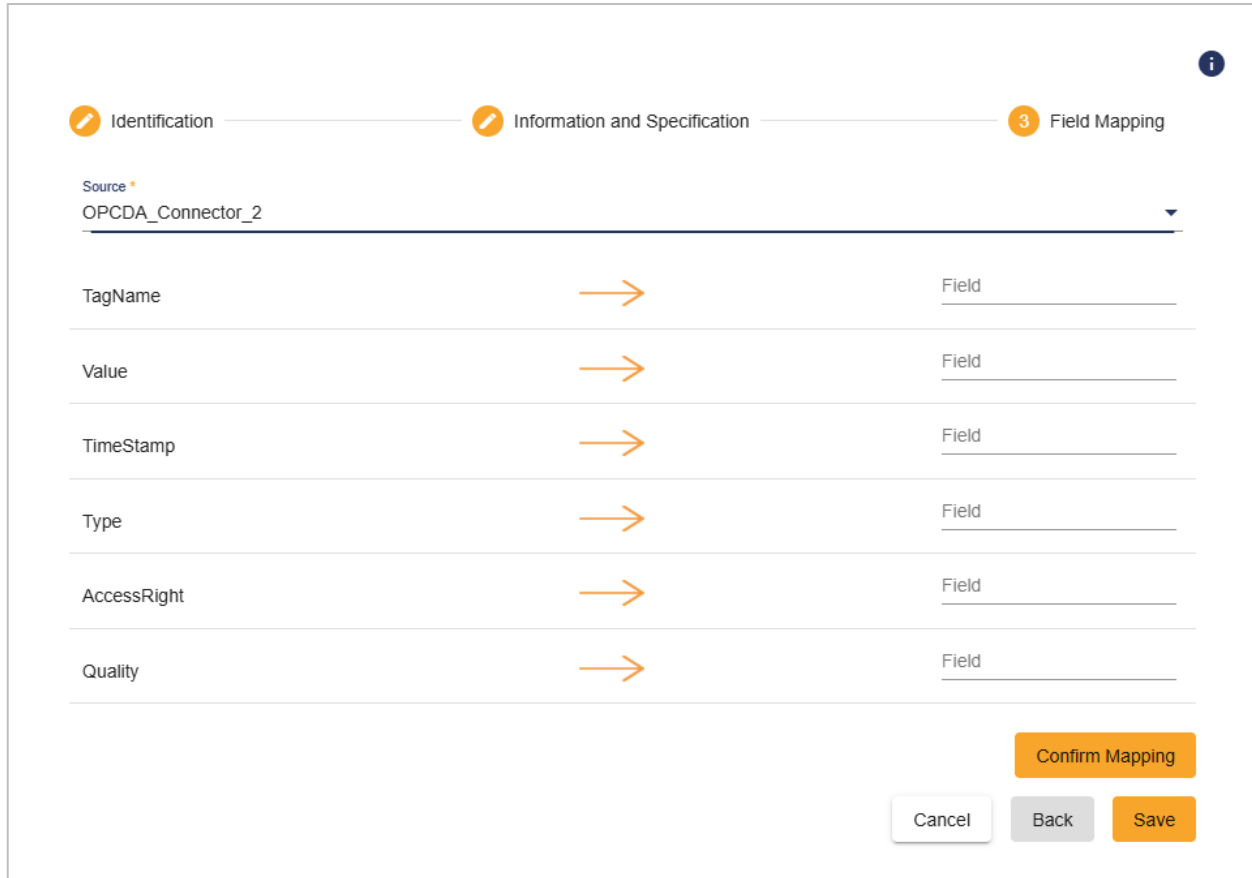


Figure 92: REST Connector as Source - Fields Configuration View

REST Connector as Destination: The **Field Mapping** Configuration view allows mapping source fields to destination fields to ensure correct data transfer.



The screenshot shows the 'Field Mapping' configuration view for a REST Connector as Destination. The interface includes a progress bar at the top with three steps: 'Identification', 'Information and Specification', and '3 Field Mapping'. Below the progress bar, the 'Source' is set to 'OPCDA_Connector_2'. The main area contains a table with six rows, each representing a source field mapped to a destination 'Field'. The source fields are 'TagName', 'Value', 'TimeStamp', 'Type', 'AccessRight', and 'Quality'. Each row has an orange arrow pointing from the source field to the destination 'Field'. At the bottom right, there are four buttons: 'Confirm Mapping' (orange), 'Cancel' (white), 'Back' (grey), and 'Save' (orange).

Source Field	Destination Field
TagName	Field
Value	Field
TimeStamp	Field
Type	Field
AccessRight	Field
Quality	Field

Figure 93: REST Connector as Destination - Field Mapping Configuration View

5.2.6.16.S7

Click **New Subscription** from the left section in the **Tag Configuration** page to add a subscription to an S7 connector. A new window will open where you can configure the subscription settings.

Source Connector

Add Subscription

Name *
Subscription1|

Device *
S7_Device_1 ▼ ⊕ ✎

Update Rate (ms) *
1000

PublishMode *
Synchronous ▼

Cancel
Save

Destination Connector

Add Subscription

Name *
Subscription1|

Update Rate (ms) *
1000

PublishMode *
Synchronous ▼

Cancel
Save

Figure 94: S7 Connector as Source / Destination - Subscription Configuration View

Parameter	Description	Default Value
<i>Name</i>	Specifies the name of the subscription.	Subscription1
<i>Device</i>	Specifies the S7 device from which data is read or to which data is written. When the connector is configured as a Destination , the Device field is configured during the Identification step.	
<i>Update Rate (ms)</i>	Defines the frequency, in milliseconds, of data read requests. When On Data Change mode is enabled, this value represents the maximum rate at which data change notifications are sent to the client callback.	1000

<i>Publish Mode</i>	<p>Defines the data publishing method supported by S7 devices:</p> <ul style="list-style-type: none"> • On Data Change: Data is transmitted only when a value changes within the specified update rate. • Synchronous: Data is collected using a synchronous polling mechanism. 	Synchronous
----------------------------	---	-------------

Table 66: S7 Connector - Subscription Configuration Parameters

5.2.7. Data Stores

5.2.7.1. CSV File

Click **Next** to access the **CSV Configuration** page, where parameters specific to CSV file handling can be defined.

The available configuration options vary depending on whether the connector is configured as a **Source** or a **Destination**.

CSV Connector as Source:

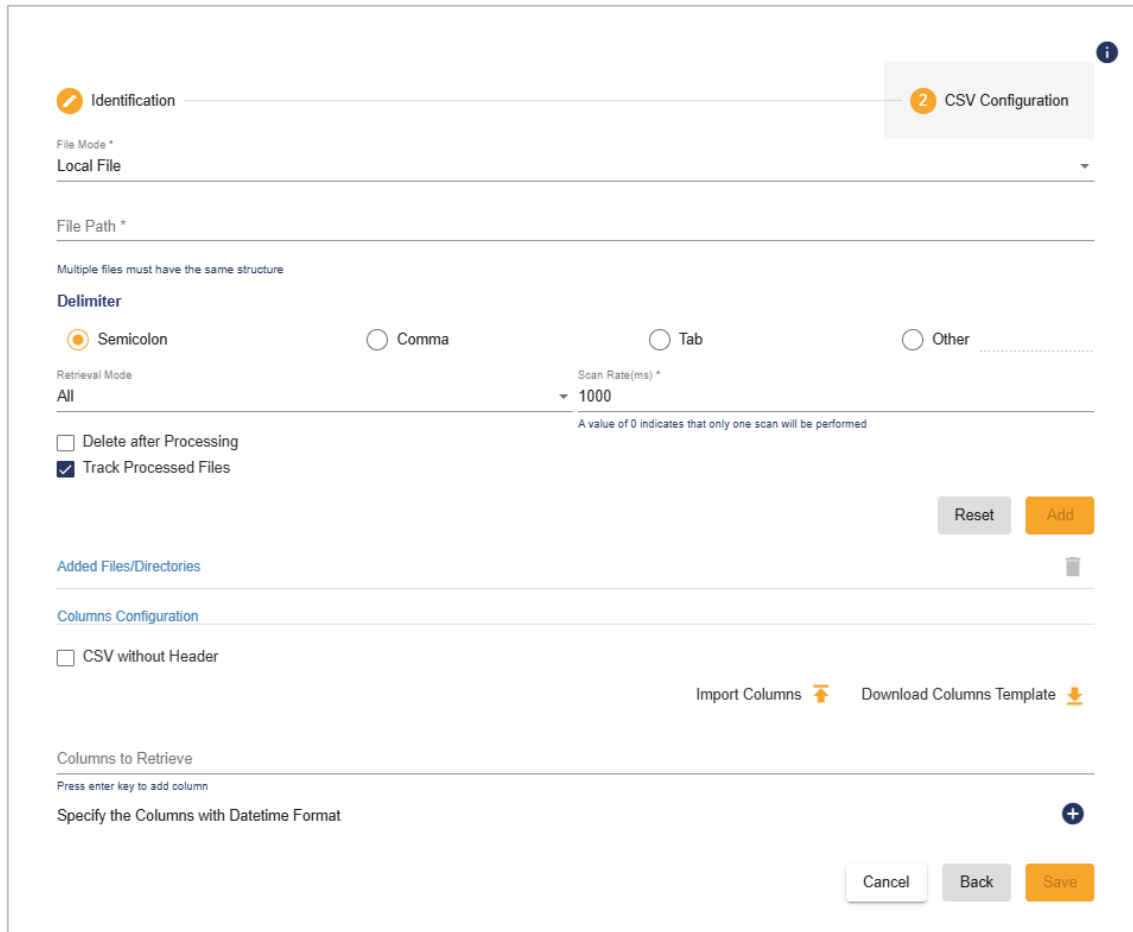


Figure 95: CSV Connector as Source – CSV Configuration View

Parameter	Description	Default Value
<i>File Mode</i>	<p>Specifies how CSV files are accessed:</p> <ul style="list-style-type: none"> • Local File: Reads files directly from a folder on the local system. • FTP File: Retrieves CSV files from an FTP server using the configured device and file path. 	Local File

Device	Specifies the FTP device used for file transfer when File Mode is set to FTP .	
File Path	Path to a single CSV file or a folder containing multiple CSV files to be processed. All files within a folder must share the same structure (headers, delimiters, etc.).	
Delimiter	Character(s) used to separate columns in the CSV file. Supported options include semicolon (;), comma (,), tab (\t), or a custom delimiter.	Semicolon
Retrieval Mode	Specifies which rows from the CSV file are published: <ul style="list-style-type: none"> • All: Publishes every row from the CSV file. • Last Value: Publishes only the last value from each processed file, determined by First Row, Last Row, or a Referenced Date Field, using a specified Date Format. • Last Value by Item: Publishes the most recent value for each item, using both the Referenced Date Field and Referenced Tag Field to identify the correct row, with a specified Date Format. 	All
Scan Rate (ms)	Interval, in milliseconds, at which the CSV file(s) are reprocessed. A value of 0 indicates that the file is read only once.	1000
Delete after Processing	When enabled, deletes the CSV file after it has been successfully processed.	Unchecked
Track Processed Files	Applies only when processing files from a folder: <ul style="list-style-type: none"> • Enabled: Stores processed file names to prevent reprocessing after a connector restart. 	Enabled

	<ul style="list-style-type: none"> • Disabled: Reprocesses all files after restart, which may result in duplicate data. 	
CSV without Header	If enabled, the first row is treated as data; otherwise, the first row is treated as a header.	Unchecked
Columns to Retrieve	Specifies which columns to retrieve, either by column name (if headers exist) or by column index.	
Specify the Columns with Datetime Format	Defines columns that contain datetime values and specifies their expected formats for accurate parsing. Multiple datetime columns with different formats can be configured.	

Table 67: CSV Connector as Source - Configuration Parameters

CSV Connector as Destination:

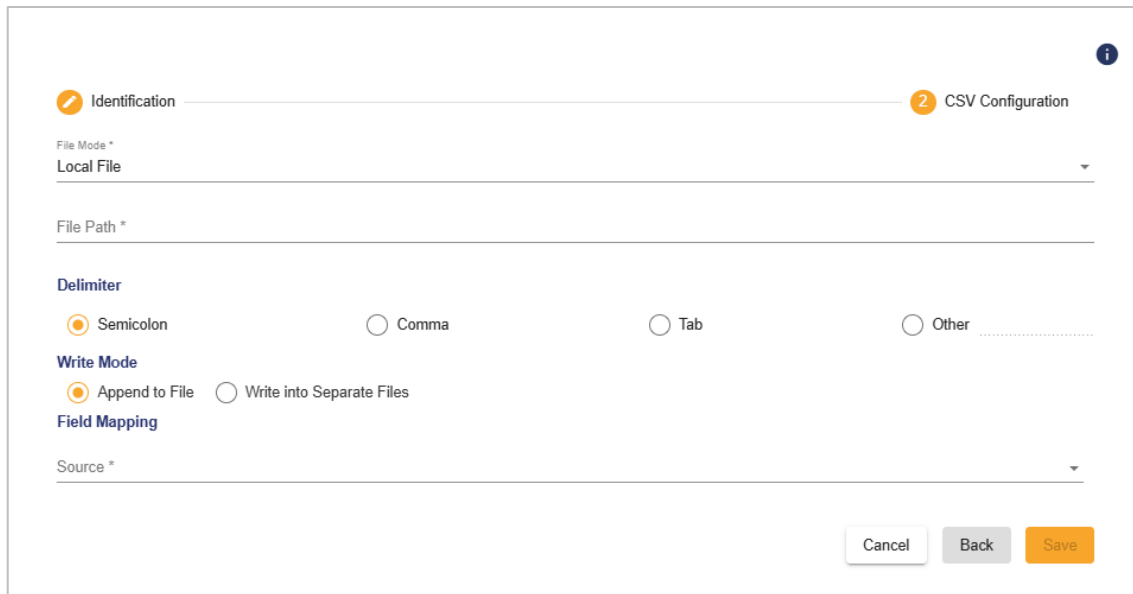


Figure 96: CSV Connector as Destination - Configuration View

Parameter	Description	Default Value
File Mode	<p>Specifies how CSV files are accessed:</p> <ul style="list-style-type: none"> • Local File: Writes files directly to a folder on the local system. • FTP File: Writes CSV files to an FTP server using the configured device and file path. 	Local File
Device	Specifies the FTP device used for file transfer when File Mode is set to FTP .	
File Path	Path to a single CSV file or a destination folder. All generated CSV files must share the same structure (headers, delimiters, etc.).	
Delimiter	Character(s) used to separate columns in the CSV file. Supported options include semicolon (;), comma (,), tab (\t), or a custom delimiter.	Semicolon
Write Mode	<p>Determines how data is written to CSV files:</p> <ul style="list-style-type: none"> • Append to File: Appends new rows to an existing CSV file. • Write into Separate Files: Creates separate files per source field. In this mode, each source field must be mapped to a corresponding destination field. When this mode is selected, the following parameters must be configured: <ul style="list-style-type: none"> ○ Write Header: Determines whether to include column headers in each file or not. ○ Max Size (Mb): Specifies the maximum file size before creating a new file. 	Append to File

	<ul style="list-style-type: none"> ○ Periodicity (ms): Defines the time interval for writing data. 	
Field Mapping	Defines the mapping between source fields and CSV columns, ensuring correct data placement within the CSV structure.	

Table 68: CSV Connector as Destination - Configuration Parameters

5.2.7.2. InfluxDB

Click **Next** to proceed to the **Measurement Configuration** page, where you can define parameters related to Influx measurements.

(!) Note

The InfluxDB Connector is available **only as a Destination**.

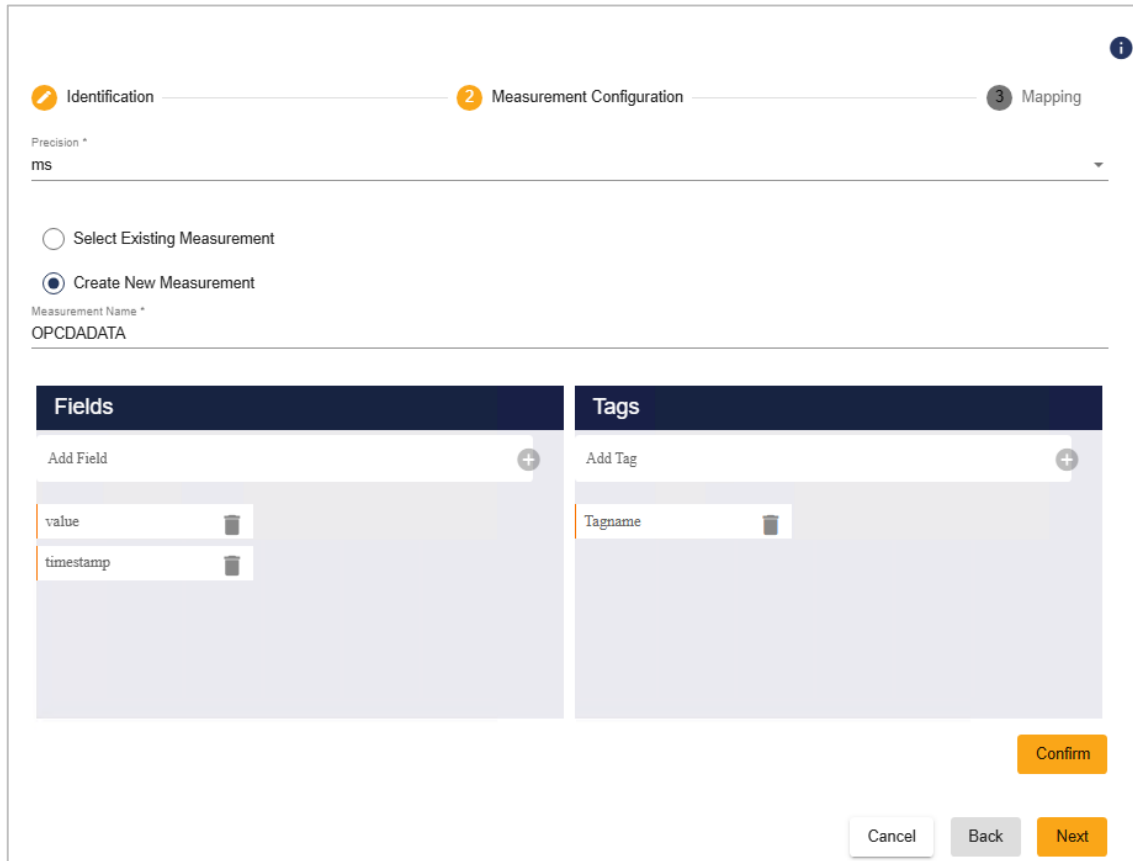


Figure 97: InfluxDB Connector as Destination - Measurement Configuration View

Parameter	Description	Default Value
<i>Precision</i>	<p>Defines the timestamp precision for data points written to InfluxDB. Options include:</p> <ul style="list-style-type: none"> ns: Nanoseconds us: Microseconds ms: Milliseconds s: Seconds 	ms
<i>Select Existing Measurement</i>	Enables selection of an existing measurement from the InfluxDB database.	Checked

Create New Measurement	Creates a new measurement in InfluxDB using the provided measurement name.	
Measurement Name	<ul style="list-style-type: none"> If Select Existing Measurement is enabled, choose a measurement from the drop-down list.If Create New Measurement is enabled, specify the name of the new measurement. 	
Fields	Defines the actual data values to be stored in InfluxDB.	
Tags	Labels used to describe or categorize data.	

Table 69: InfluxDB Connector as Destination - Measurement Configuration Parameters

Click **Next** to proceed to the **Mapping** page. On this page, you can map tags and fields from the source connectors to the corresponding measurement fields in InfluxDB.

5.2.7.3. Kafka

Click **Next** to proceed to the **Tag Configuration** page. The available parameters may vary depending on whether the connector is configured as a **Source** or a **Destination**.

Kafka Connector as Source:



Figure 98: Kafka Connector as Source - Tag Configuration View

Parameter	Description	Default Value
Group ID	Unique identifier assigned to a group of Kafka consumers. Consumers with the same Group ID share message consumption from topics, enabling load balancing.	Group_123
Auto Offset Reset	Determines consumer behavior when no initial offset is found or the current offset is invalid. Options: <ul style="list-style-type: none"> Earliest: Resets to the earliest available message. Error: Throws an exception if no valid offset is present. Latest: Resets to the latest available message. 	Latest

<i>Fetch Message Max Byte</i>	Maximum size (in bytes) of a message the consumer can fetch in a single request. By default, Kafka allows large messages, typically up to 50 MB.	52428800
<i>Session Timeout (ms)</i>	Timeout used to detect client failures in Kafka's group management. Heartbeats are sent periodically; if none are received within this interval, the broker triggers a rebalance.	10000
<i>Kafka Topics</i>	Lists the Kafka topics used by the broker for message exchange.	

Table 70: Kafka Connector as Source - Tag Configuration Parameters

Kafka Connector as Destination:

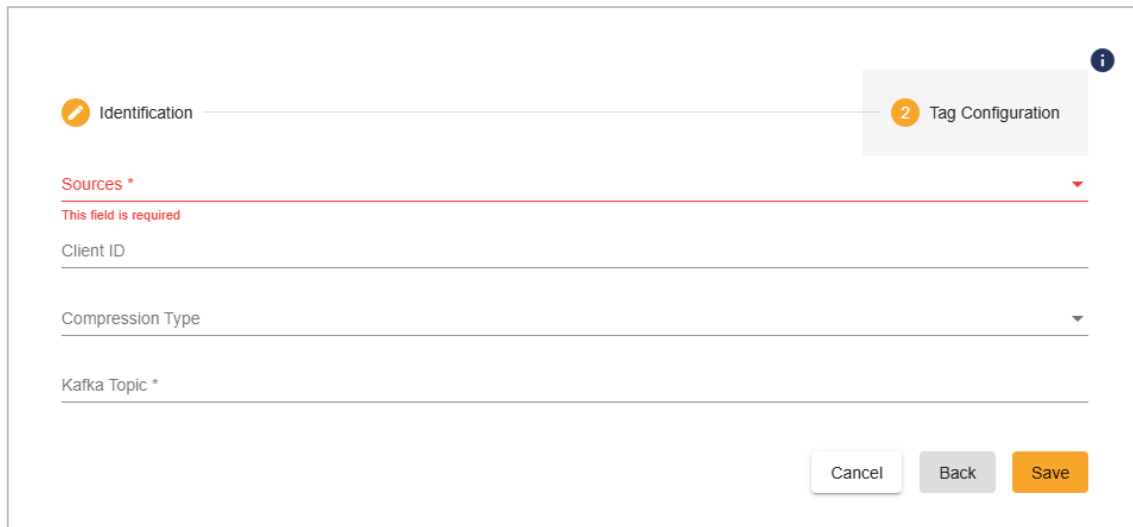


Figure 99: Kafka Connector as Destination - Tag Configuration View

Parameter	Description	Default Value
<i>Sources</i>	Specifies the source connectors from which data will be received and published to the Kafka broker.	

<i>Client ID</i>	Unique identifier for the Kafka producer client, used for request identification, monitoring, and logging.	
<i>Compression Type</i>	Compression algorithm used when sending messages to the Kafka broker. Options: <ul style="list-style-type: none"> • None: No compression. • Gzip: High compression, higher CPU usage. • Snappy: Fast compression/decompression, moderate ratio. • Lz4: Very fast compression/decompression, suitable for low-latency systems. • Zstd: Balanced compression and performance. 	None
<i>Kafka Topic</i>	Kafka topic to which the connector will publish data.	

Table 71: Kafka Connector as Destination - Tag Configuration Parameters

5.2.7.4. MongoDB

MongoDB is a cross-platform, document-oriented database classified as NoSQL system. It stores data in flexible, JSON-like documents with optional schemas.

Click **Next** to proceed to the **Collection Configuration** page, where you can define parameters related to data mapping.

(!) Note

The MongoDB Connector is available **only as a Destination**.

On this page, you will need to select the **Source Connector** from which data will be received and stored in MongoDB, as well as the table (collection) that will hold the data. You can either select an existing collection or create a new one directly from the configuration interface.



Create New Collection

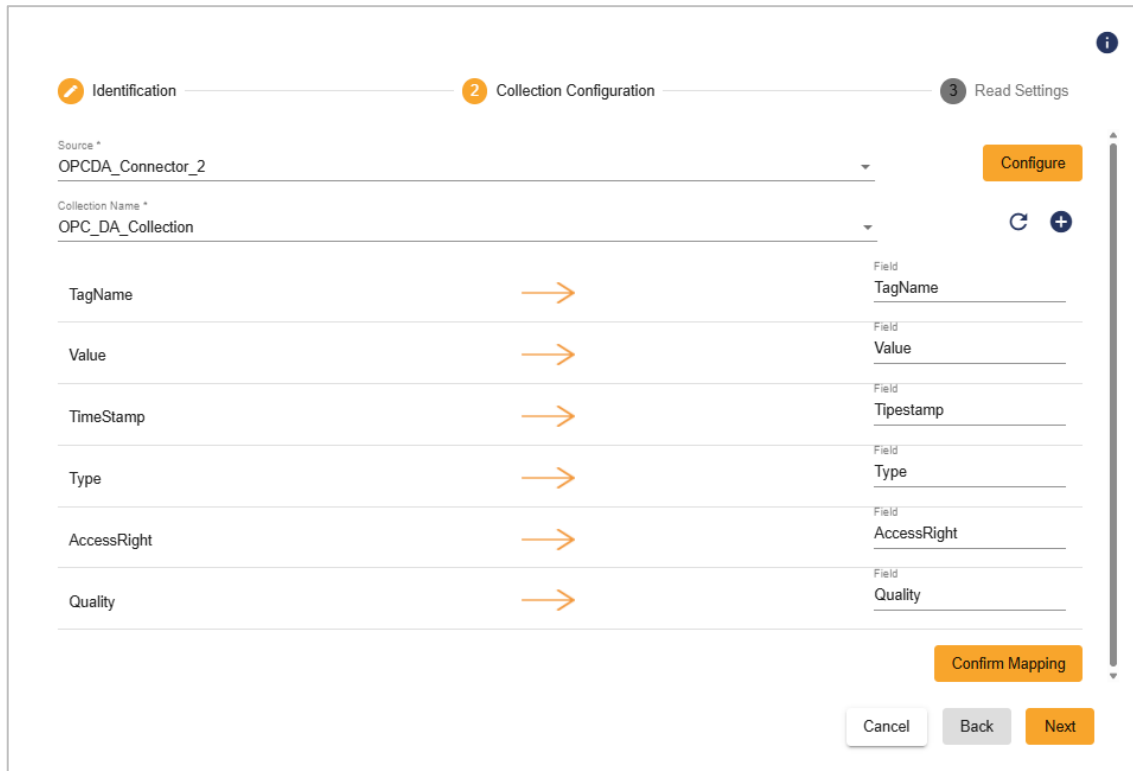
Collection name *

OPC_DA_Collection

Cancel Save

Figure 100: MongoDB Connector as Destination - Create New Collection View

After selecting or creating the collection, you will be required to map the **source connector tags/fields** to the corresponding **MongoDB fields** to ensure correct data alignment and storage structure.



Identification 2 Collection Configuration 3 Read Settings

Source *
OPCDA_Connector_2

Collection Name *
OPC_DA_Collection

Configure

Tag Name → Field
TagName

Value → Field
Value

TimeStamp → Field
Tipestamp

Type → Field
Type

AccessRight → Field
AccessRight

Quality → Field
Quality

Confirm Mapping

Cancel Back Next

Figure 101: MongoDB Connector as Destination – Collection Configuration View

Click the **Confirm Mapping** button to validate the configuration, and then click **Next** to proceed to the next step, where you can configure the **Read Settings**, such as the timeout for read requests.

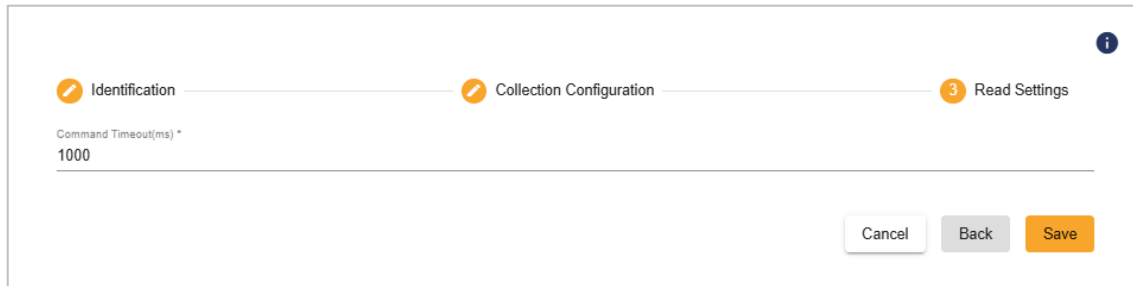


Figure 102: MongoDB Connector as Destination - Read Settings Configuration View

Parameter	Description	Default Value
<i>Command Timeout (ms)</i>	Specifies the maximum duration (in milliseconds) that the connector will wait for a response to a read command before timing out.	1000

Table 72: MongoDB Connector as Destination - Read Settings Configuration Parameters

5.2.7.5. MS Access

Click **Next** to proceed to the **Table Configuration** page, where you can define parameters related to data mapping.

(!) Note

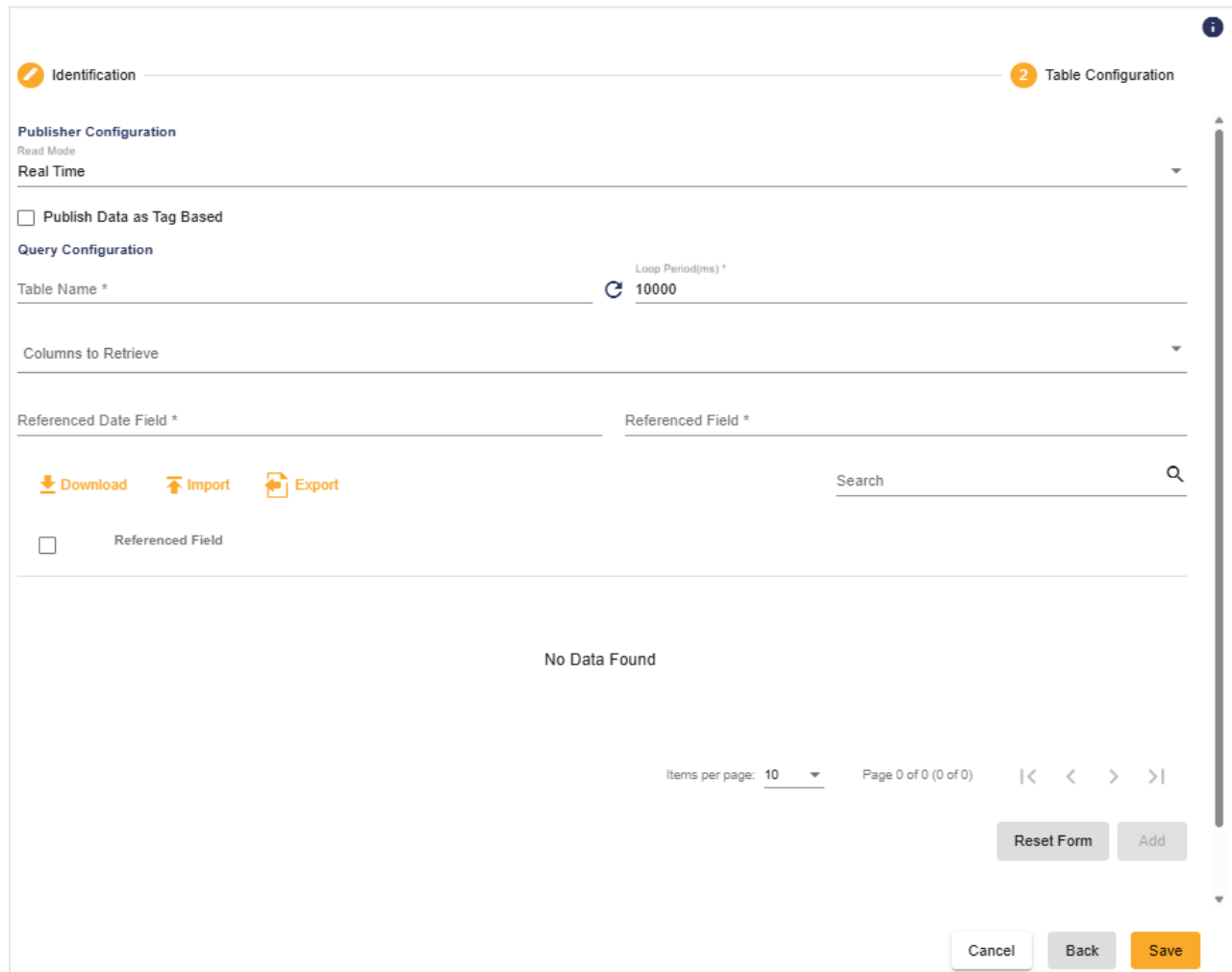
Before using the connector, ensure that the **Microsoft Access Database Engine** is installed on your system. This component is required to enable interaction with **Microsoft Access** and **Excel** files.

- [Microsoft Access Database Engine 2016 Redistributable \(64-bit\)](#)
- [Microsoft Access Database Engine 2016 Redistributable \(32-bit\)](#)

Choose the 32-bit or 64-bit version according to your operating system and the version of the application you are running.

Configuration options vary depending on whether the connector is set as a **Source** or a **Destination**.

MS Access Connector as Source:



The screenshot displays the 'Table Configuration' view for the MS Access Connector as Source. The interface includes a top navigation bar with 'Identification' and 'Table Configuration' tabs. The 'Table Configuration' tab is selected. Below the tabs, there are several configuration sections: 'Publisher Configuration' with a 'Read Mode' dropdown set to 'Real Time'; a checkbox for 'Publish Data as Tag Based'; 'Query Configuration' with a 'Table Name' field and a 'Loop Period(ms)' field set to '10000'; and a table for 'Referenced Fields'. The table is currently empty, showing 'No Data Found'. At the bottom, there are buttons for 'Reset Form', 'Add', 'Cancel', 'Back', and 'Save'.

Figure 103: MS Access Connector as Source – Table Configuration View

Parameter	Description	Default Value
<i>Publisher Configuration</i>		
<i>Read Mode</i>	Specifies how data is retrieved. Available options include:	Real Time

	<ul style="list-style-type: none"> • Real Time: Retrieves the most recent values as they are updated. • Historian: Retrieves historical data from the database. 	
Publish Data as Tag Based	When enabled, data is published using tags instead of raw fields.	Unchecked
Query Configuration		
Default Query Configuration		
Table Name	Specifies the name of the source table from which data will be retrieved.	
Loop Period (ms)	Defines the refresh interval in milliseconds for retrieving data.	10000
Columns to Retrieve	Select the specific columns to fetch from the source table.	
Referenced Date Field	Field used as the reference for date or timestamp.	
Referenced Field	Field used as the reference for tag-based or key-based retrieval.	
Referenced Field Table	After adding a referenced field, the data is displayed in the table. The user must select at least one item from the table to add the query. Alternatively, the user can add items using a template	

	<ul style="list-style-type: none"> - Download Template: Download a template including an example of Referenced Field. - Import Template: Import a predefined Referenced Field configuration template. - Export Template: Export the configured items for reuse. 	
Reset Form	Reset the table configuration by clicking the Reset Form button.	
Publish Data as Tag Based Configuration		
Write Mode	<p>The write mode is available only when "Publish Data as Tag Based" is enabled. All write operations are executed from the OPC UA Server destination.</p> <p>The are two write mode options:</p> <ul style="list-style-type: none"> • Insert: used to insert a new row into the table. • Update: used to update an existing row in the table. 	Insert
Query configuration		
Default Type	Select the default type for the referenced field table items.	String
Referenced Type Field	Field used as the reference for data type.	

Type Mapping	<p>The type mapping button is available only when "Referenced Type Field" is added.</p> <p>Its purpose is to map database data types with OPC UA Server data types.</p>	
Field Aliasing		
Field to map	Defines the tag-based fields (TagName, Value, Timestamp, and Quality) to be mapped to the table fields.	
Field name	Select the table field corresponding to the chosen " Field to map ".	
Historian Read Mode Query Configuration		
Start Time	Defines the start of the time range for historical data retrieval.	Current time – 10 minutes
End Time	Defines the end of the time range for historical data retrieval.	Unchecked Current time
Data Interval(ms)	Specifies how frequently new data is transmitted.	60000
Real time Offset(ms)	<p>Specifies a time offset applied when the calculated End Time exceeds the current system time (real time).</p> <p>When this condition is met, the End Time is adjusted using the formula:</p>	2000

	Adjusted End Time = Real Time – Real Time Offset The Real Time Offset must be lower than the Loop Period .	
Restart from Last Execution Time	Resumes historical data processing from the point where the previous execution stopped.	Unchecked
Include Bounds	When enabled, bounding values are included in the results.	Unchecked
Queries		
Queries	Displays the list of configured queries. Queries in the table can be edited or removed.	

Table 73: MS Access Connector as Source – Table Configuration Parameters

Once the **Query Configuration** parameters are set, you can select the tags to be read from the **Referenced Field**. Additionally, you can use the **Download**, **Export**, and **Import** options to manage and configure the list of tags associated with the **Referenced Field**.

Click **Add** and then **Save** your configuration.







MS Access Connector as Destination:



Select the **Source Connector** from which data will be received and stored in the database, along with the table that will store the data. You can choose an existing table or create a new one directly from the configuration interface.

Create New Table

Table Name *
OPC_DA_Table

+

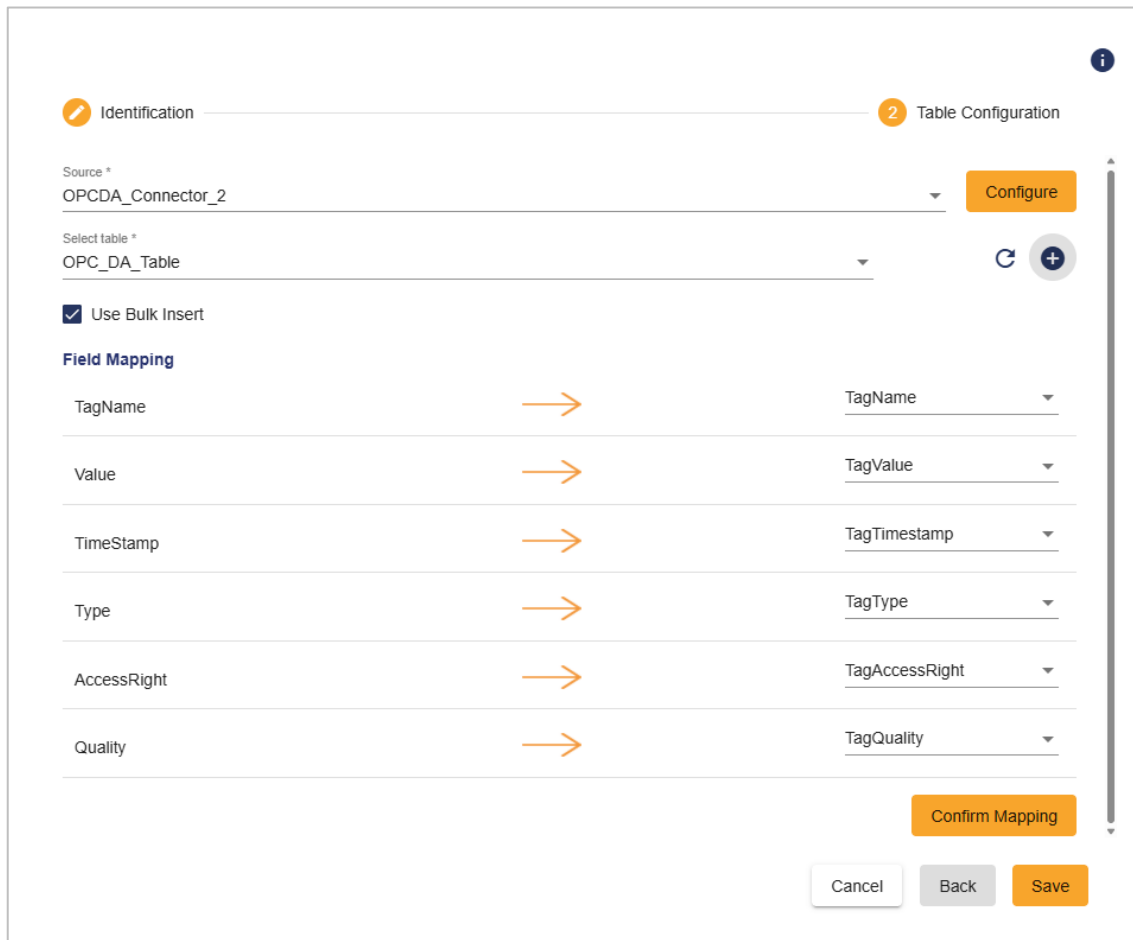
Column Name *	Data Type *		
TagName	Short Text	▼	
Column Name *	Data Type *		
TagValue	Short Text	▼	
Column Name *	Data Type *		
TagTimestamp	Date/Time	▼	
Column Name *	Data Type *		
TagType	Short Text	▼	
Column Name *	Data Type *		
TagAccessRight	Short Text	▼	
Column Name *	Data Type *		
TagQuality	Short Text	▼	

Upload Table Schema (csv) 
Download Template 

Cancel Save

Figure 104: MS Access Connector as Destination – Create New Table Configuration View

After selecting or creating the table, map the **Source Connector** tags/fields to the corresponding table columns to ensure proper data alignment and storage structure.



The screenshot displays the 'Table Configuration' step (labeled '2') of the MS Access Connector configuration. The 'Identification' step (labeled '1') is also visible. The 'Source' is set to 'OPCDA_Connector_2' and the 'Select table' is 'OPCDA_Table'. A 'Configure' button is next to the source. The 'Use Bulk Insert' checkbox is checked. The 'Field Mapping' section shows a table with columns for source fields and their corresponding destination fields in the target table. The source fields are TagName, Value, TimeStamp, Type, AccessRight, and Quality. The destination fields are TagName, TagValue, TagTimestamp, TagType, TagAccessRight, and TagQuality. Arrows indicate the mapping from source to destination. A 'Confirm Mapping' button is at the bottom right, along with 'Cancel', 'Back', and 'Save' buttons.

Source Field	Destination Field
TagName	TagName
Value	TagValue
TimeStamp	TagTimestamp
Type	TagType
AccessRight	TagAccessRight
Quality	TagQuality

Figure 105: MS Access Connector as Destination – Table Configuration and Field Mapping Configuration View

Once the mapping is complete, click **Confirm Mapping** to validate the configuration and then **Save** your configuration.

5.2.7.6. MYSQL

Click **Next** to proceed to the **Table Configuration** page, where you can define parameters related to data mapping.

Configuration options vary depending on whether the connector is set as a **Source** or a **Destination**.

MySQL Connector as Source:

1 Identification
2 Table Configuration

Publisher Configuration
Read Mode
Real Time

☐ Publish Data as Tag Based

Query Configuration

Table Name *

Loop Period(ms) *
10000

Columns to Retrieve

Referenced Date Field *
Referenced Field *

Download
Import
Export

Search

☐ Referenced Field

No Data Found

Items per page: 10
Page 0 of 0 (0 of 0)

Reset Form
Add

Cancel
Back
Save

Figure 106: MySQL Connector as Source - Table Configuration View

Parameter	Description	Default Value
<i>Publisher Configuration</i>		
<i>Read Mode</i>	<p>Specifies how data is retrieved. Available options include:</p> <ul style="list-style-type: none"> Real Time: Retrieves the most recent values as they are updated. 	Real Time

	<ul style="list-style-type: none"> • Historian: Retrieves historical data from the database. 	
<i>Publish Data as Tag Based</i>	When enabled, data is published using tags instead of raw fields.	Unchecked
<i>Query Configuration</i>		
<i>Default Query Configuration</i>		
<i>Table Name</i>	Specifies the name of the source table from which data will be retrieved.	
<i>Loop Period (ms)</i>	Defines the refresh interval in milliseconds for retrieving data.	10000
<i>Columns to Retrieve</i>	Select the specific columns to fetch from the source table.	
<i>Referenced Date Field</i>	Field used as the reference for date or timestamp.	
<i>Referenced Field</i>	Field used as the reference for tag-based or key-based retrieval.	
<i>Referenced Field Table</i>	After adding a referenced field, the data is displayed in the table. The user must select at least one item from the table to add the query. Alternatively, the user can add items using a template	

	<ul style="list-style-type: none"> - Download Template: Download a template including an example of Referenced Field. - Import Template: Import a predefined Referenced Field configuration template. - Export Template: Export the configured items for reuse. 	
Reset Form	Reset the table configuration by clicking the Reset Form button.	
Publish Data as Tag Based Configuration		
Write Mode	<p>The write mode is available only when "Publish Data as Tag Based" is enabled. All write operations are executed from the OPC UA Server destination.</p> <p>The are two write mode options:</p> <ul style="list-style-type: none"> • Insert: used to insert a new row into the table. • Update: used to update an existing row in the table. 	Insert
Query configuration		
Default Type	Select the default type for the referenced field table items.	String
Referenced Type Field	Field used as the reference for data type.	

Type Mapping	<p>The type mapping button is available only when "Referenced Type Field" is added.</p> <p>Its purpose is to map database data types with OPC UA Server data types.</p>	
Field Aliasing		
Field to map	Defines the tag-based fields (TagName, Value, Timestamp, and Quality) to be mapped to the table fields.	
Field name	Select the table field corresponding to the chosen " Field to map ".	
Historian Read Mode Query Configuration		
Start Time	Defines the start of the time range for historical data retrieval.	Current time – 10 minutes
End Time	Defines the end of the time range for historical data retrieval.	Unchecked Current time
Data Interval(ms)	Specifies how frequently new data is transmitted.	60000
Real time Offset(ms)	<p>Specifies a time offset applied when the calculated End Time exceeds the current system time (real time).</p> <p>When this condition is met, the End Time is adjusted using the formula:</p>	2000

	Adjusted End Time = Real Time – Real Time Offset The Real Time Offset must be lower than the Loop Period .	
<i>Restart from Last Execution Time</i>	Resumes historical data processing from the point where the previous execution stopped.	Unchecked
<i>Include Bounds</i>	When enabled, bounding values are included in the results.	Unchecked
<i>Queries</i>		
<i>Queries</i>	Displays the list of configured queries. Queries in the table can be edited or removed.	

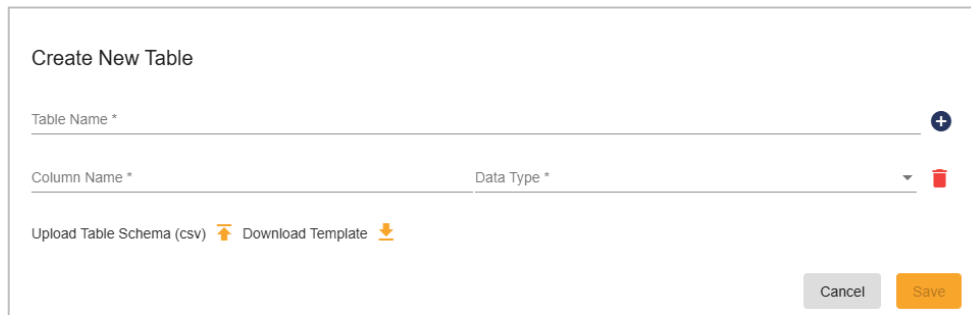
Table 74: MySQL Connector as Source - Table Configuration Parameters

Once the **Query Configuration** parameters are set, you can select the tags to be read from the **Referenced Field**. Additionally, you can use the **Download**, **Export**, and **Import** options to manage and configure the list of tags associated with the **Referenced Field**.

Click **Add** and then **Save** your configuration.

MySQL Connector as Destination:

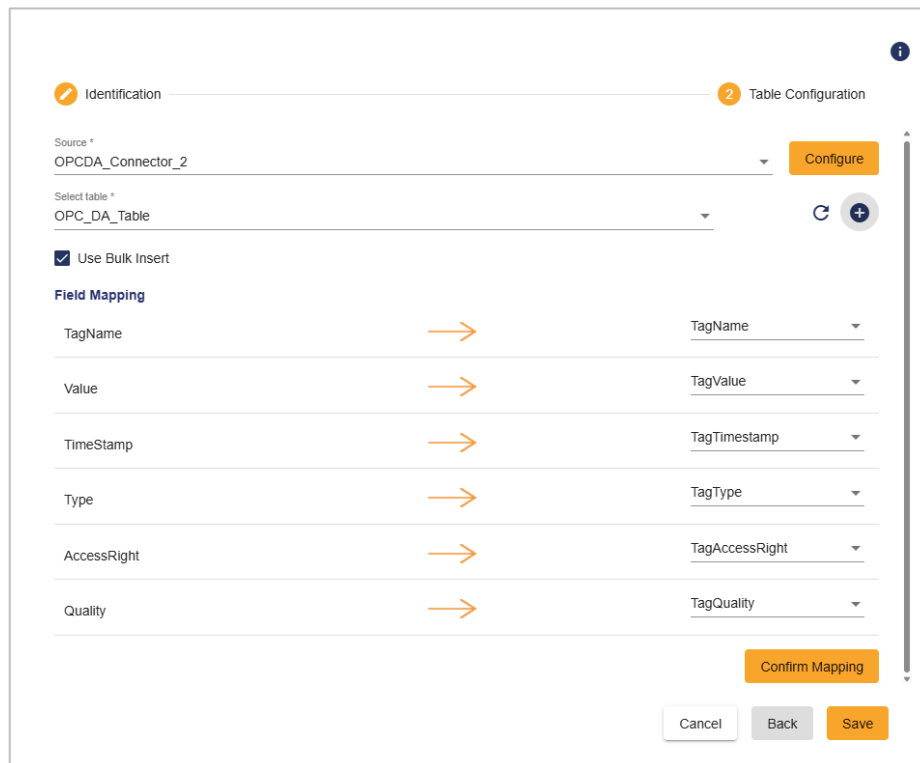
Select the **Source Connector** from which data will be received and stored in the database, along with the table that will store the data. You can choose an existing table or create a new one directly from the configuration interface.



The 'Create New Table' configuration view includes a title bar, a 'Table Name' input field with a plus icon, a 'Column Name' and 'Data Type' input fields with a dropdown and a trash icon, and an 'Upload Table Schema (csv)' button with a download icon. At the bottom are 'Cancel' and 'Save' buttons.

Figure 107: MySQL Connector as Destination - Create New Table Configuration View

After selecting or creating the table, map the **Source Connector** tags/fields to the corresponding table columns to ensure proper data alignment and storage structure.



The 'Table Configuration' view shows the 'Source' as 'OPCDA_Connector_2' and the 'Select table' as 'OPC_DA_Table'. It includes a 'Use Bulk Insert' checkbox and a 'Field Mapping' section. The mapping section lists source fields (TagName, Value, TimeStamp, Type, AccessRight, Quality) and their corresponding destination fields (TagName, TagValue, TagTimestamp, TagType, TagAccessRight, TagQuality). A 'Confirm Mapping' button is at the bottom right, along with 'Cancel', 'Back', and 'Save' buttons.

Figure 108: MS Access Connector as Destination – Table Configuration and Field Mapping Configuration View

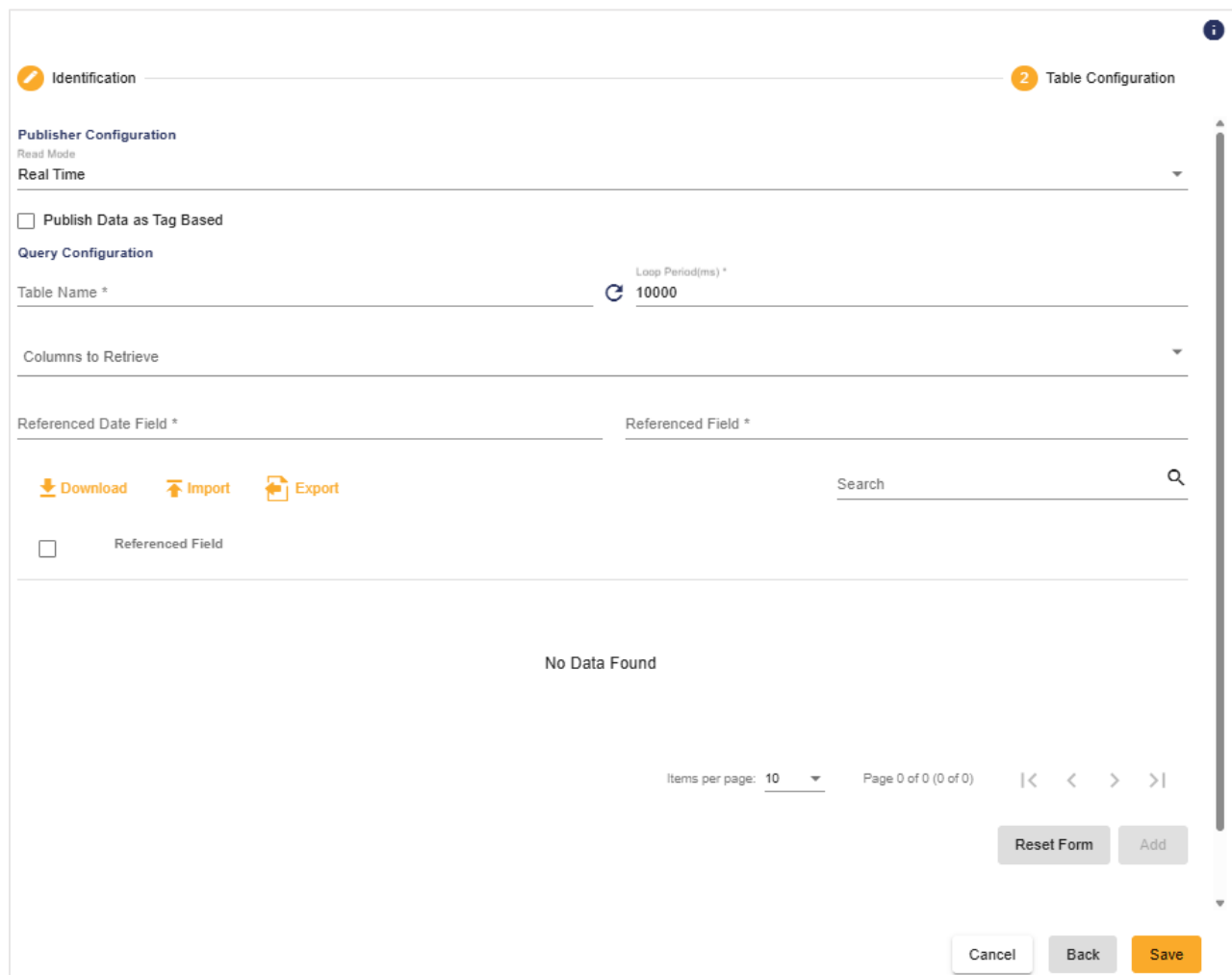
Once the mapping is complete, click **Confirm Mapping** to validate the configuration and then **Save** your configuration.

5.2.7.7. ODBC

Click **Next** to proceed to the **Table Configuration** page, where you can define parameters related to data mapping.

Configuration options vary depending on whether the connector is set as a **Source** or a **Destination**.

ODBC Connector as Source:



The screenshot displays the 'Table Configuration' view for an ODBC Connector as Source. The interface includes a top navigation bar with 'Identification' and 'Table Configuration' tabs. The 'Table Configuration' section contains several configuration options: 'Publisher Configuration' (Read Mode: Real Time), 'Query Configuration' (Table Name, Loop Period: 10000), and a table for 'Referenced Field'. The table is currently empty, showing 'No Data Found'. At the bottom, there are buttons for 'Cancel', 'Back', 'Save', 'Reset Form', and 'Add'.

Figure 109: ODBC Connector as Source - Table Configuration View

Parameter	Description	Default Value
<i>Publisher Configuration</i>		

Read Mode	<p>Specifies how data is retrieved. Available options include:</p> <ul style="list-style-type: none"> • Real Time: Retrieves the most recent values as they are updated. • Historian: Retrieves historical data from the database. 	Real Time
Publish Data as Tag Based	When enabled, data is published using tags instead of raw fields.	Unchecked
Query Configuration		
Default Query Configuration		
Table Name	Specifies the name of the source table from which data will be retrieved.	
Loop Period (ms)	Defines the refresh interval in milliseconds for retrieving data.	10000
Columns to Retrieve	Select the specific columns to fetch from the source table.	
Referenced Date Field	Field used as the reference for date or timestamp.	
Referenced Field	Field used as the reference for tag-based or key-based retrieval.	
Referenced Field Table	<p>After adding a referenced field, the data is displayed in the table.</p> <p>The user must select at least one item from the table to add the query.</p>	

	<p>Alternatively, the user can add items using a template</p> <ul style="list-style-type: none"> - Download Template: Download a template including an example of Referenced Field. - Import Template: Import a predefined Referenced Field configuration template. - Export Template: Export the configured items for reuse. 	
Reset Form	Reset the table configuration by clicking the Reset Form button.	
Publish Data as Tag Based Configuration		
Write Mode	<p>The write mode is available only when “Publish Data as Tag Based” is enabled. All write operations are executed from the OPC UA Server destination.</p> <p>The are two write mode options:</p> <ul style="list-style-type: none"> • Insert: used to insert a new row into the table. • Update: used to update an existing row in the table. 	Insert
Query configuration		
Default Type	Select the default type for the referenced field table items.	String

Referenced Type Field	Field used as the reference for data type.	
Type Mapping	<p>The type mapping button is available only when "Referenced Type Field" is added.</p> <p>Its purpose is to map database data types with OPC UA Server data types.</p>	
Field Aliasing		
Field to map	Defines the tag-based fields (TagName, Value, Timestamp, and Quality) to be mapped to the table fields.	
Field name	Select the table field corresponding to the chosen "Field to map" .	
Historian Read Mode Query Configuration		
Start Time	Defines the start of the time range for historical data retrieval.	Current time – 10 minutes
End Time	Defines the end of the time range for historical data retrieval.	Unchecked Current time
Data Interval(ms)	Specifies how frequently new data is transmitted.	60000
Real time Offset(ms)	Specifies a time offset applied when the calculated End Time exceeds the current system time (real time).	2000

	<p>When this condition is met, the End Time is adjusted using the formula:</p> <p>Adjusted End Time = Real Time – Real Time Offset</p> <p>The Real Time Offset must be lower than the Loop Period.</p>	
Restart from Last Execution Time	Resumes historical data processing from the point where the previous execution stopped.	Unchecked
Include Bounds	When enabled, bounding values are included in the results.	Unchecked
Queries		
Queries	<p>Displays the list of configured queries.</p> <p>Queries in the table can be edited or removed.</p>	

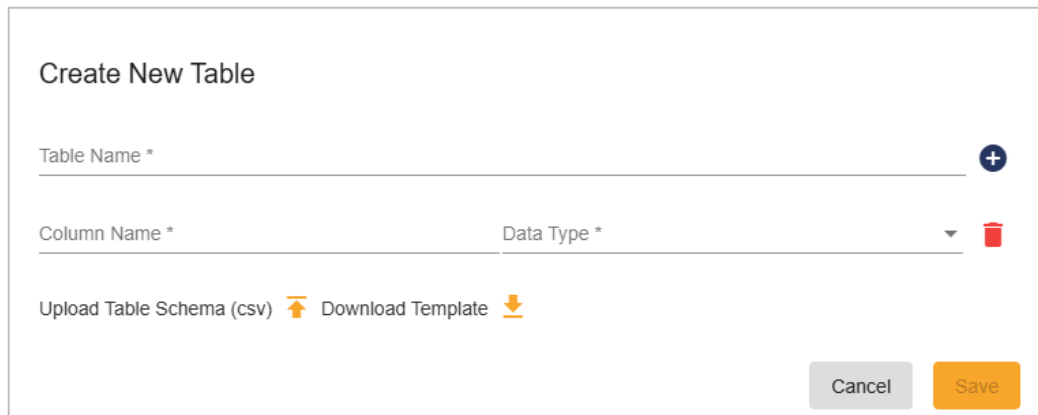
Table 75: ODBC Connector as Source – Table Configuration Parameters

Once the **Query Configuration** parameters are set, you can select the tags to be read from the **Referenced Field**. Additionally, you can use the **Download**, **Export**, and **Import** options to manage and configure the list of tags associated with the **Referenced Field**.

Click **Add** and then **Save** your configuration.

ODBC Connector as Destination:

Select the **Source Connector** from which data will be received and stored in the database, along with the table that will store the data. You can choose an existing table or create a new one directly from the configuration interface.



Create New Table

Table Name *

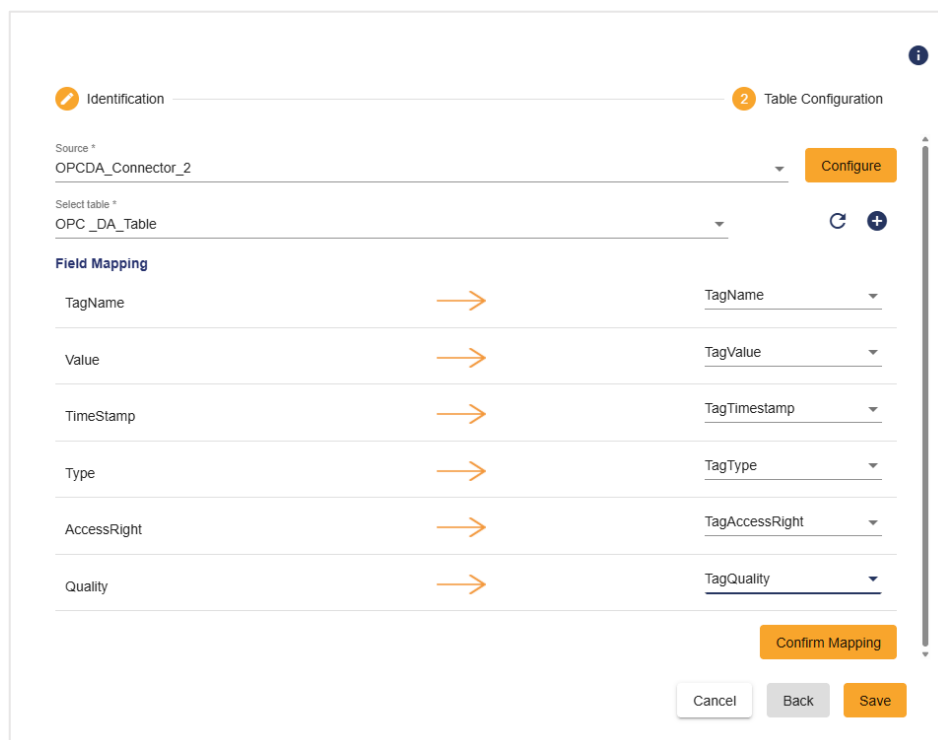
Column Name * Data Type *

Upload Table Schema (csv) Download Template

Cancel Save

Figure 110: ODBC Connector as Destination - Create New Table Configuration View

After selecting or creating the table, map the **Source Connector** tags/fields to the corresponding table columns to ensure proper data alignment and storage structure.



Identification **2 Table Configuration**

Source *
OPCDA_Connector_2 Configure

Select table *
OPC_DA_Table Refresh Add

Field Mapping

TagName	→	TagName
Value	→	TagValue
TimeStamp	→	TagTimestamp
Type	→	TagType
AccessRight	→	TagAccessRight
Quality	→	TagQuality

Confirm Mapping

Cancel Back Save

Figure 111: ODBC Connector as Destination - Table Configuration and Field Mapping Configuration View

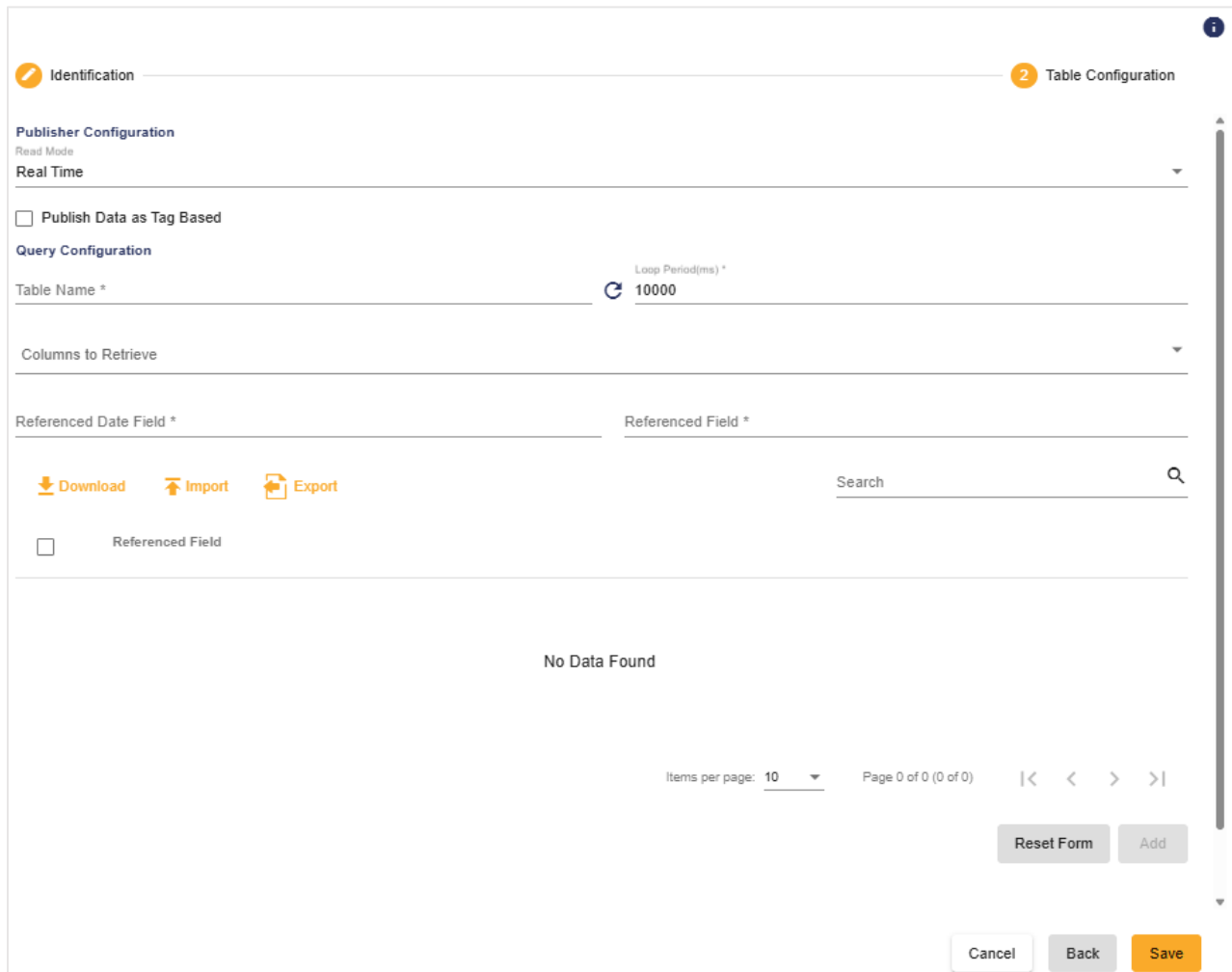
Once the mapping is complete, click **Confirm Mapping** to validate the configuration and then **Save** your configuration.

5.2.7.8. Oracle

Click **Next** to proceed to the **Table Configuration** page, where you can define parameters related to data mapping.

Configuration options vary depending on whether the connector is set as a **Source** or a **Destination**.

Oracle Connector as Source:



The screenshot displays the 'Table Configuration' page for the Oracle Connector as Source. The page is divided into two main sections: 'Publisher Configuration' and 'Query Configuration'. The 'Publisher Configuration' section includes a 'Read Mode' dropdown set to 'Real Time' and a checkbox for 'Publish Data as Tag Based'. The 'Query Configuration' section includes a 'Table Name' field, a 'Loop Period(ms)' field set to '10000', a 'Columns to Retrieve' dropdown, and two fields for 'Referenced Date Field' and 'Referenced Field'. Below these fields are buttons for 'Download', 'Import', and 'Export', along with a 'Search' field. At the bottom of the page, there is a 'Referenced Field' checkbox, a 'No Data Found' message, a pagination bar showing 'Items per page: 10' and 'Page 0 of 0 (0 of 0)', and buttons for 'Reset Form', 'Add', 'Cancel', 'Back', and 'Save'.

Figure 112: Oracle Connector as Source - Table Configuration View

Parameter	Description	Default Value
<i>Publisher Configuration</i>		
<i>Read Mode</i>	<p>Specifies how data is retrieved. Available options include:</p> <ul style="list-style-type: none"> • Real Time: Retrieves the most recent values as they are updated. • Historian: Retrieves historical data from the database. 	Real Time
<i>Publish Data as Tag Based</i>	When enabled, data is published using tags instead of raw fields.	Unchecked
<i>Query Configuration</i>		
<i>Default Query Configuration</i>		
<i>Table Name</i>	Specifies the name of the source table from which data will be retrieved.	
<i>Loop Period (ms)</i>	Defines the refresh interval in milliseconds for retrieving data.	10000
<i>Columns to Retrieve</i>	Select the specific columns to fetch from the source table.	
<i>Referenced Date Field</i>	Field used as the reference for date or timestamp.	
<i>Referenced Field</i>	Field used as the reference for tag-based or key-based retrieval.	

Referenced Field Table	<p>After adding a referenced field, the data is displayed in the table. The user must select at least one item from the table to add the query. Alternatively, the user can add items using a template</p> <ul style="list-style-type: none"> - Download Template: Download a template including an example of Referenced Field. - Import Template: Import a predefined Referenced Field configuration template. - Export Template: Export the configured items for reuse. 	
Reset Form	Reset the table configuration by clicking the Reset Form button.	
Publish Data as Tag Based Configuration		
Write Mode	<p>The write mode is available only when "Publish Data as Tag Based" is enabled. All write operations are executed from the OPC UA Server destination.</p> <p>The are two write mode options:</p> <ul style="list-style-type: none"> • Insert: used to insert a new row into the table. • Update: used to update an existing row in the table. 	Insert

Query configuration		
Default Type	Select the default type for the referenced field table items.	String
Referenced Type Field	Field used as the reference for data type.	
Type Mapping	<p>The type mapping button is available only when "Referenced Type Field" is added.</p> <p>Its purpose is to map database data types with OPC UA Server data types.</p>	
Field Aliasing		
Field to map	Defines the tag-based fields (TagName, Value, Timestamp, and Quality) to be mapped to the table fields.	
Field name	Select the table field corresponding to the chosen "Field to map" .	
Historian Read Mode Query Configuration		
Start Time	Defines the start of the time range for historical data retrieval.	Current time – 10 minutes
End Time	Defines the end of the time range for historical data retrieval.	Unchecked Current time
Data Interval(ms)	Specifies how frequently new data is transmitted.	60000

Real time Offset(ms)	<p>Specifies a time offset applied when the calculated End Time exceeds the current system time (real time).</p> <p>When this condition is met, the End Time is adjusted using the formula:</p> <p>Adjusted End Time = Real Time – Real Time Offset</p> <p>The Real Time Offset must be lower than the Loop Period.</p>	2000
Restart from Last Execution Time	Resumes historical data processing from the point where the previous execution stopped.	Unchecked
Include Bounds	When enabled, bounding values are included in the results.	Unchecked
Queries		
Queries	<p>Displays the list of configured queries.</p> <p>Queries in the table can be edited or removed.</p>	

Table 76: Oracle Connector as Source – Table Configuration Parameters

Once the **Query Configuration** parameters are set, you can select the tags to be read from the **Referenced Field**. Additionally, you can use the **Download**, **Export**, and **Import** options to manage and configure the list of tags associated with the **Referenced Field**.

Click **Add** and then **Save** your configuration.

Oracle Connector as Destination:

Select the **Source Connector** from which data will be received and stored in the database, along with the table that will store the data. You can choose an existing table or create a new one directly from the configuration interface.

Create New Table

Table Name *
OPC_DA_Table

+

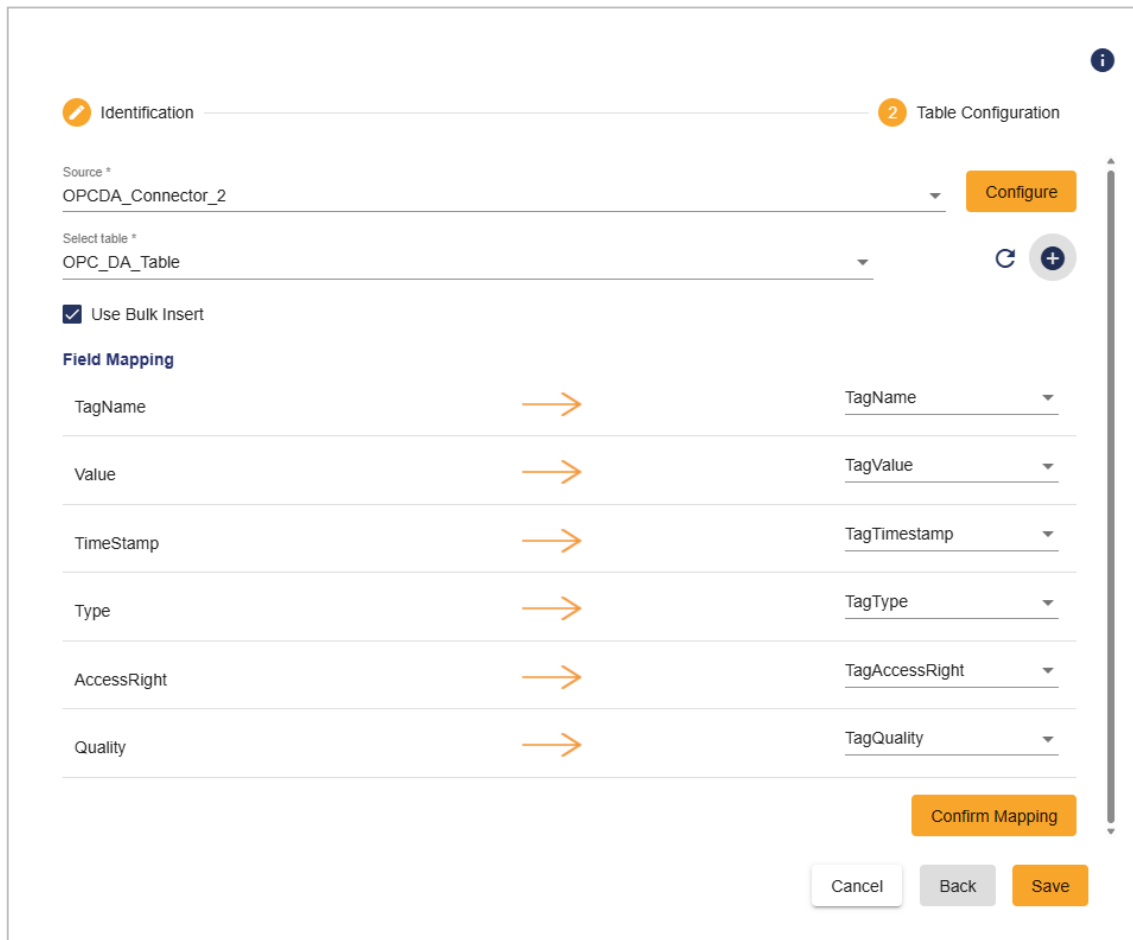
Column Name *	Data Type *		
TagName	Short Text	▼	🗑
Column Name *	Data Type *		
TagValue	Short Text	▼	🗑
Column Name *	Data Type *		
TagTimestamp	Date/Time	▼	🗑
Column Name *	Data Type *		
TagType	Short Text	▼	🗑
Column Name *	Data Type *		
TagAccessRight	Short Text	▼	🗑
Column Name *	Data Type *		
TagQuality	Short Text	▼	🗑

Upload Table Schema (csv) 📁
Download Template 📄

Cancel
Save

Figure 113: MS Access Connector as Destination – Create New Table Configuration View

After selecting or creating the table, map the **Source Connector** tags/fields to the corresponding table columns to ensure proper data alignment and storage structure.



The screenshot shows a configuration window with two tabs: "Identification" (active) and "Table Configuration".

Identification Tab:

- Source ***: OPCDA_Connector_2
- Select table ***: OPC_DA_Table
- ☒ Use Bulk Insert

Field Mapping:

Source Field	Destination Field
TagName	TagName
Value	TagValue
TimeStamp	TagTimestamp
Type	TagType
AccessRight	TagAccessRight
Quality	TagQuality

Buttons: Confirm Mapping, Cancel, Back, Save.

Figure 114: Oracle Connector as Destination – Table Configuration and Field Mapping Configuration View

Once the mapping is complete, click **Confirm Mapping** to validate the configuration and then **Save** your configuration.

5.2.7.9. PI

The **PI Connector** requires the same core parameters to be configured during the **Identification** step. In addition, it provides extra configuration options specific to the PI system.

Parameter	Description	Default Value
Advanced Configuration		
Payload Transformation	Available when the connector is configured as a Destination . Enables the transformation of a field-based payload into a basic tag-based payload.	Unchecked

Table 77: PI Connector Additional Configuration Parameters

Click **Next** to proceed to the **Tag Configuration** page. The available parameters may vary depending on whether the connector is configured as a **Source** or a **Destination**.

PI Connector as Source:

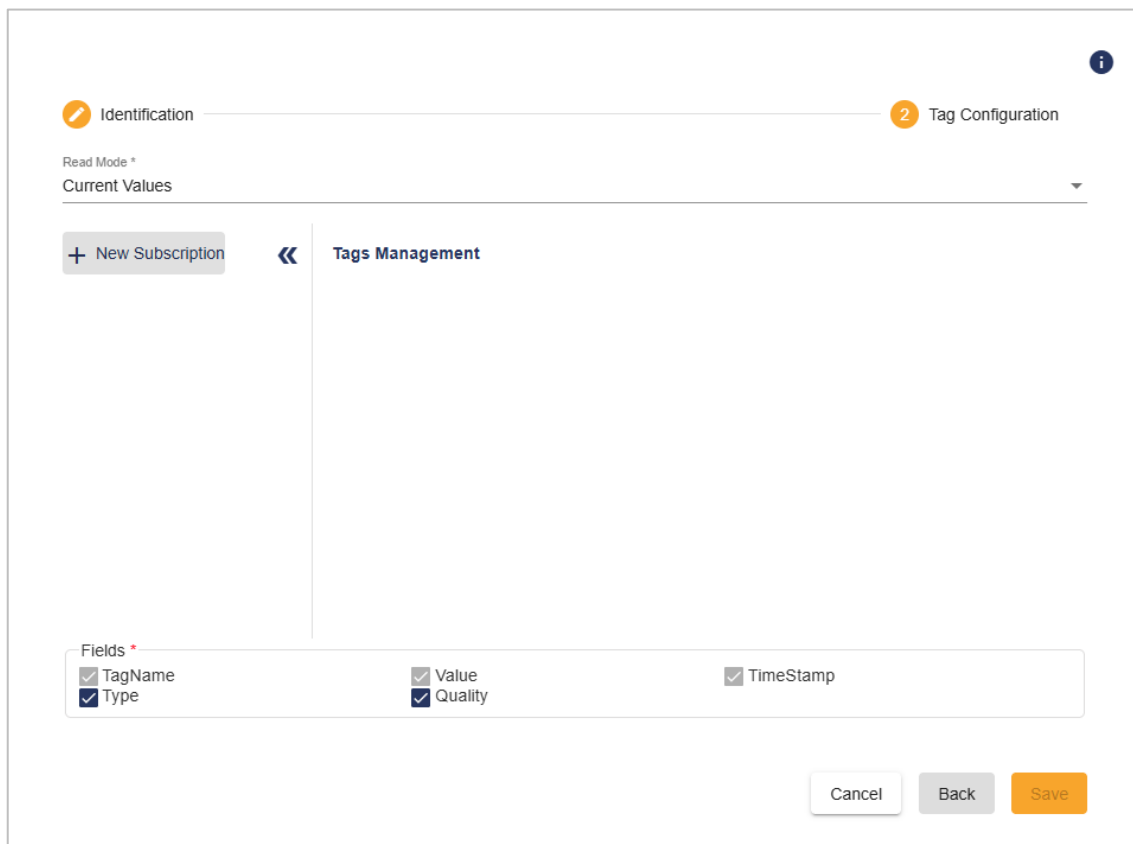
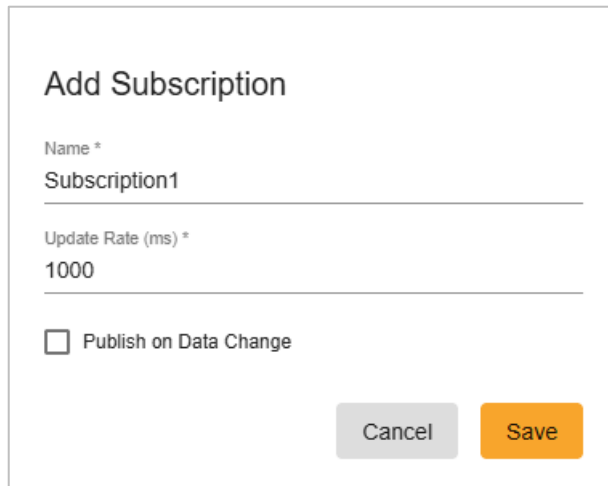


Figure 115: PI Connector as Source - Tag Configuration View

Click **New Subscription** from the left section in the **Tag Configuration** page to add a subscription to a PI connector configured as **Source**. A new window opens, allowing you to configure the subscription settings.

The **Subscription Configuration** view varies based on the selected **Read Mode**:

- **Current Values:** Retrieves the most recent values available at the time of the request.
- **Archived Values:** Retrieves historical (archived) data for a specified time range or period.



The image shows a dialog box titled "Add Subscription". It contains two text input fields. The first field is labeled "Name *" and contains the text "Subscription1". The second field is labeled "Update Rate (ms) *" and contains the text "1000". Below these fields is a checkbox labeled "Publish on Data Change", which is currently unchecked. At the bottom right of the dialog box are two buttons: a grey "Cancel" button and an orange "Save" button.

Figure 116: PI Connector as Source - Current Values Read Mode - Subscription Configuration View

Add Subscription

Name *
Subscription1

Start Time *
1/29/2026, 14:45:15

☐ End Time

Read Period (s) *
10

Sample Interval (s) *
60

Max Returned Values *
0

Bounds *
Inside

☐ Restart from Last Executed Time

Cancel Save

Figure 117: PI Connector as Source - Archived Values Read Mode - Subscription Configuration View

Parameter	Description	Default Value
Name	Specifies the name of the subscription.	Subscription1
Read Mode: Current Values		
Update Rate (ms)	Defines the frequency of data read requests. In On Data Change mode, this value represents the maximum rate at which data change notifications are sent to the client callback. Expressed in milliseconds.	1000
Publish on Data Change	When enabled, data is sent to the PI Client only when a value changes within the defined update rate.	Unchecked
Read Mode: Archived Values		

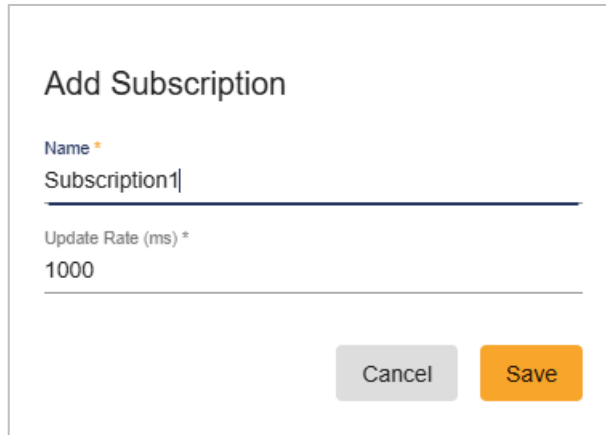
<i>Start Time</i>	Specifies the start of the period from which archived data is read.	Current time
<i>End Time</i>	Specifies the end of the data retrieval period.	Unchecked
<i>Read Period (s)</i>	Defines the time range, in seconds, for retrieving archived data, indicating how far back data is fetched from the PI System.	10
<i>Sample Interval (s)</i>	Defines the sampling frequency, in seconds, within the specified read period. For example, a value of 60 samples data every minute.	60
<i>Max Returned Values</i>	Limits the maximum number of data points returned by the connector to prevent excessive data retrieval.	0
<i>Bounds</i>	Defines how boundary values are handled: <ul style="list-style-type: none"> • Inside: Returns values within the specified range. • Outside: Allows values outside the defined range. 	Inside
<i>Restart from Last Executed Time</i>	When enabled, historical data processing resumes from the last successfully executed timestamp.	Unchecked

Table 78: PI Connector as Source - Subscription Configuration Parameters

After the subscription is added, click **Browse PI Points** to view the available PI points and select the tags to be imported. You can also click **Download Template** to obtain a reference template for tag configuration. Once the template is updated with the required tags, click **Import Tags** to import them into the system.

PI Connector as Destination:

Click **New Subscription** from the left section in the **Tag Configuration** page to add a subscription to a PI connector configured as **Destination**. A new window opens, allowing you to configure the subscription settings.



The image shows a dialog box titled "Add Subscription". It contains two input fields: "Name" with a red asterisk and "Update Rate (ms)" with a red asterisk. The "Name" field contains the text "Subscription1" and the "Update Rate (ms)" field contains the text "1000". At the bottom right of the dialog box are two buttons: "Cancel" (grey) and "Save" (orange).

Figure 118: PI Connector as Destination - Subscription Configuration View

Parameter	Description	Default Value
<i>Name</i>	Specifies the name of the subscription.	Subscription1
<i>Update Rate (ms)</i>	Defines the frequency of data write requests.	1000

Table 79: PI Connector as Destination - Subscription Configuration Parameters

Once the subscription is added, you can use the **Browse PI Points** button to browse available PI points and select the list of tags. You will also be required to **select the Source Connector** and **configure the field mapping** accordingly.

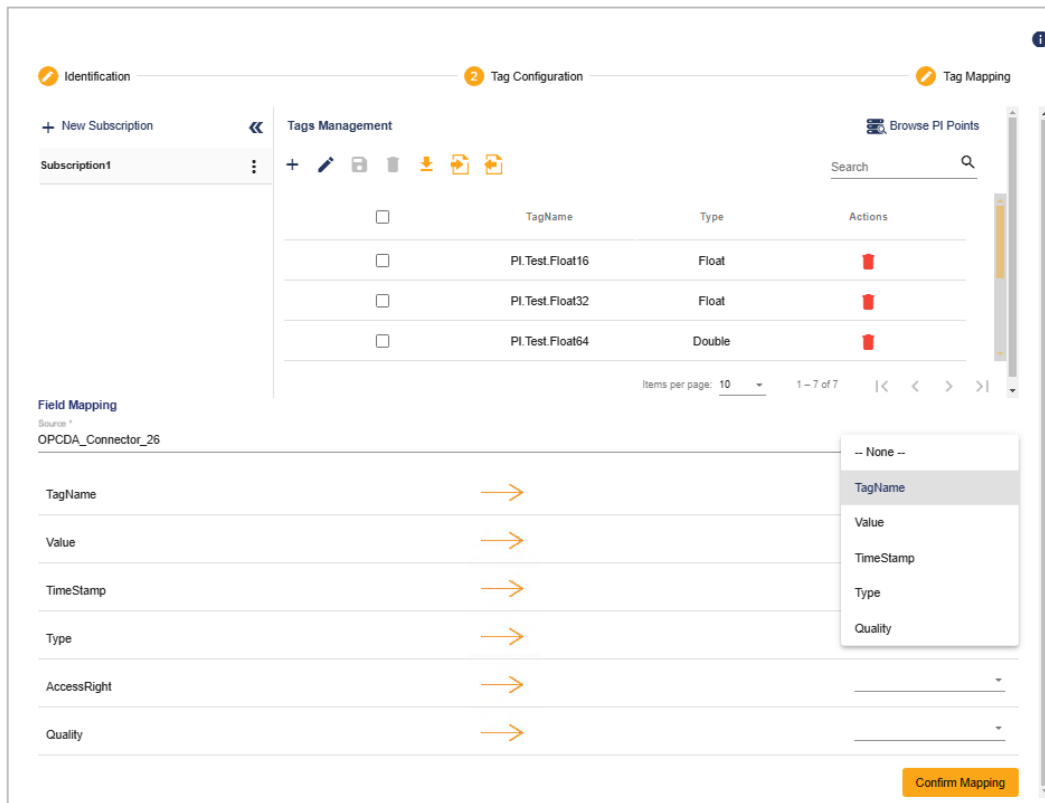


Figure 119: PI Connector as Destination - Tag Configuration View

If the **Payload Transformation** option is enabled, you can define transformation logic to modify the structure or content of the payload before it is written or processed. This feature allows you to adapt payload formats, apply mappings, or inject additional data to meet the requirements of the target system or workflow.

Identification
2 Payload Transformation
Tag Configuration
Tag Mapping

Input Schema

Source *

OPCDA_Connector_26

Preview Schema

Output Schema

Create Generic JSON Schema

Repeating Fields

TN TS VAL Repeating Fields

Number of Repeats *

1

Pasting text with ',' will automatically add it as separate conditions

Confirm

Field Transformation

Import Download

Target	Source	Attribute Value	Actions
payload[1].TN	TagName	<input checked="" type="checkbox"/>	
payload[1].TS	Value	<input checked="" type="checkbox"/>	
payload[1].VAL	Value	<input checked="" type="checkbox"/>	

Confirm Payload

Cancel Back Next

Figure 120: PI Connector as Destination - Payload Transformation

5.2.7.10.PI AF

Click **Next** to proceed to the **Tag Configuration** page. The available parameters may vary depending on whether the connector is configured as a **Source** or a **Destination**.

PI AF Connector as Source:

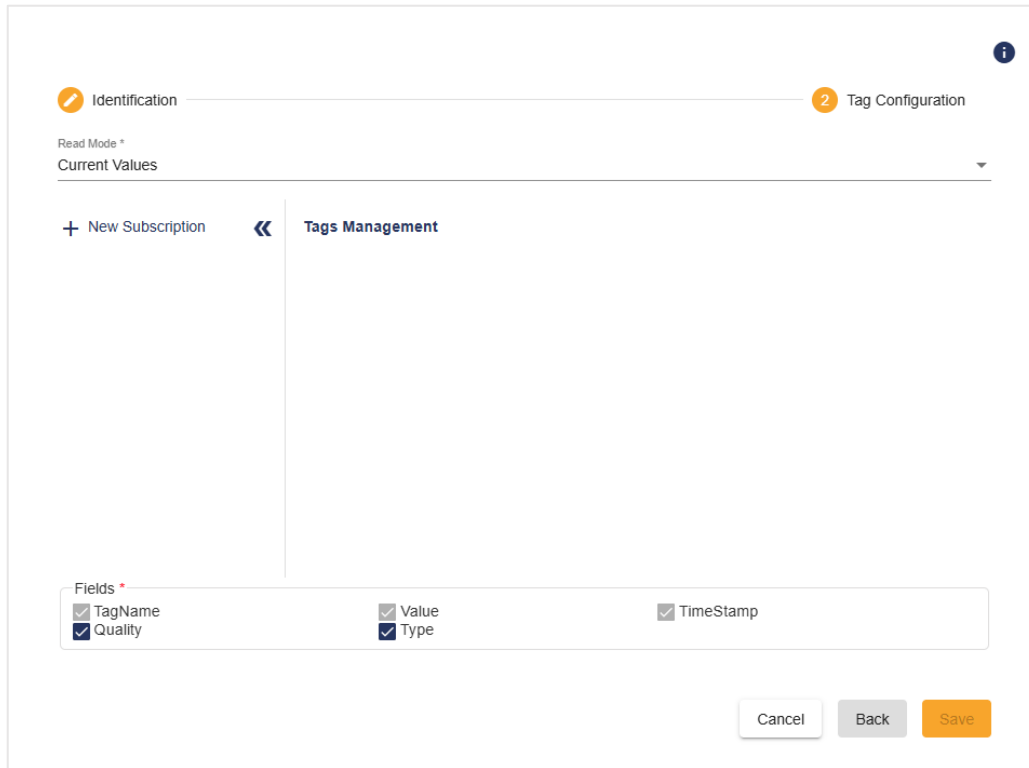


Figure 121: PI AF Connector as Source - Tag Configuration View

Click **New Subscription** from the left section in the **Tag Configuration** page to add a subscription to a PI AF connector configured as **Source**. A new window opens, allowing you to configure the subscription settings.

The **Subscription Configuration** view varies based on the selected **Read Mode**:

- **Current Values:** Retrieves the most recent values available at the time of the request.
- **Archived Values:** Retrieves historical (archived) data for a specified time range or period.

Add Subscription

Name *

Subscription1

Database *

Update Rate (ms) *

1000

☐ Publish on Data Change

Cancel Save

Figure 122: PI AF Connector as Source - Current Values Read Mode - Subscription Configuration View

Add Subscription

Name *

Subscription1

Database *

Start Time *

1/29/2026, 15:03:26

☐ End Time

Read Period (s) *

10

Sample Interval (s) *

60

Max Returned Values *

0

Bounds *

Inside

☐ Restart from Last Executed Time

Cancel Save

Figure 123: PI AF Connector as Source - Archived Values Read Mode - Subscription Configuration View

Parameter	Description	Default Value
Name	Specifies the name of the subscription.	Subscription1
Database	Displays the list of available PI AF databases. Click the Refresh icon to load or update the list.	
Read Mode: Current Values		
Update Rate (ms)	Defines the frequency of data read requests. In On Data Change mode, this value represents the maximum rate at which data change notifications are sent to the client callback. Expressed in milliseconds.	1000
Publish on Data Change	When enabled, data is transmitted to the PI Client only when a value change occurs within the defined update rate.	Unchecked
Read Mode: Archived Values		
Start Time	Specifies the start of the period from which archived data is read.	Current time
End Time	Specifies the end of the data retrieval period.	Unchecked
Read Period (s)	Defines the time range, in seconds, for retrieving archived data from the PI AF System, indicating how far back data is fetched.	10
Sample Interval (s)	Defines the sampling frequency, in seconds, within the specified read period. For example, a value of 60 samples data every minute.	60
Max Returned Values	Limits the maximum number of data points returned by the connector to prevent excessive data retrieval.	0

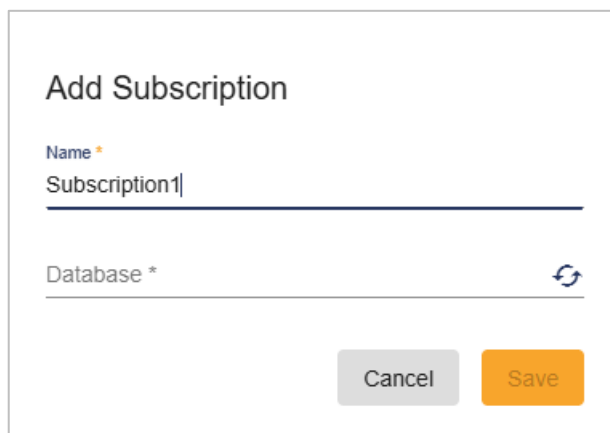
<i>Bounds</i>	<p>Defines how boundary values are handled:</p> <ul style="list-style-type: none"> • Inside: Returns values within the specified range. • Outside: Allows values outside the defined range. 	Inside
<i>Restart from Last Executed Time</i>	When enabled, historical data processing resumes from the last successfully executed timestamp.	Unchecked

Table 80: PI AF Connector as Source - Subscription Configuration Parameters

After the subscription is added, click **Browse PI AF Attributes** to view the PI AF elements tree and select the list of attributes to be imported. You can also click **Download Template** to obtain a reference template for tag configuration. Once the template is updated with the required tags, click **Import Tags** to import them into the system.

PI AF Connector as Destination:

To add a subscription to an PI AF connector configured as **Destination**, click the **New Subscription** button from the left section in the **Tag Configuration** page. A new window will open where you can configure the subscription settings.



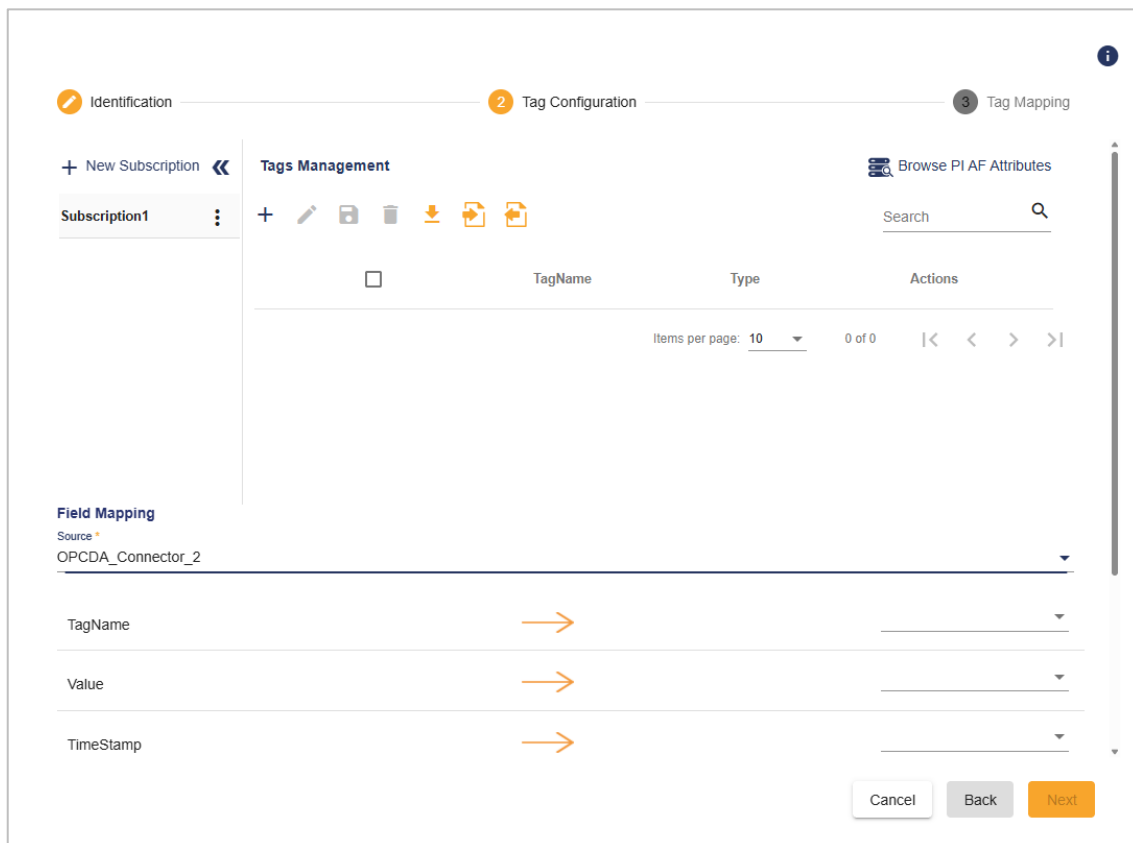
The image shows a dialog box titled "Add Subscription". It contains two input fields: "Name *" with the text "Subscription1" and "Database *" which is currently empty. To the right of the "Database *" field is a circular refresh icon. At the bottom of the dialog are two buttons: "Cancel" (grey) and "Save" (orange).

Figure 124: PI AF Connector as Destination - Subscription Configuration View

Parameter	Description	Default Value
Name	Specifies the name of the subscription.	Subscription1
Database	Displays the list of available PI AF databases. Click the Refresh icon to load or update the list.	

Table 81: PI AF Connector as Destination - Subscription Configuration Parameters

Once the subscription is added, you can use the **Browse PI AF Attributes** button to browse available PI AF elements tree and select the list of tags. You will also be required to **select the Source Connector** and **configure the field mapping** accordingly.



The screenshot displays the 'Tag Configuration' view for a PI AF Connector as Destination. The interface includes a top navigation bar with three tabs: 'Identification', 'Tag Configuration' (selected), and 'Tag Mapping'. Below the navigation bar, there is a 'Tags Management' section with a table for adding tags. The table has columns for 'TagName', 'Type', and 'Actions'. Below the table, there is a 'Field Mapping' section where the source connector is set to 'OPCDA_Connector_2'. Three fields (TagName, Value, TimeStamp) are mapped to the destination fields using orange arrows. At the bottom, there are 'Cancel', 'Back', and 'Next' buttons.

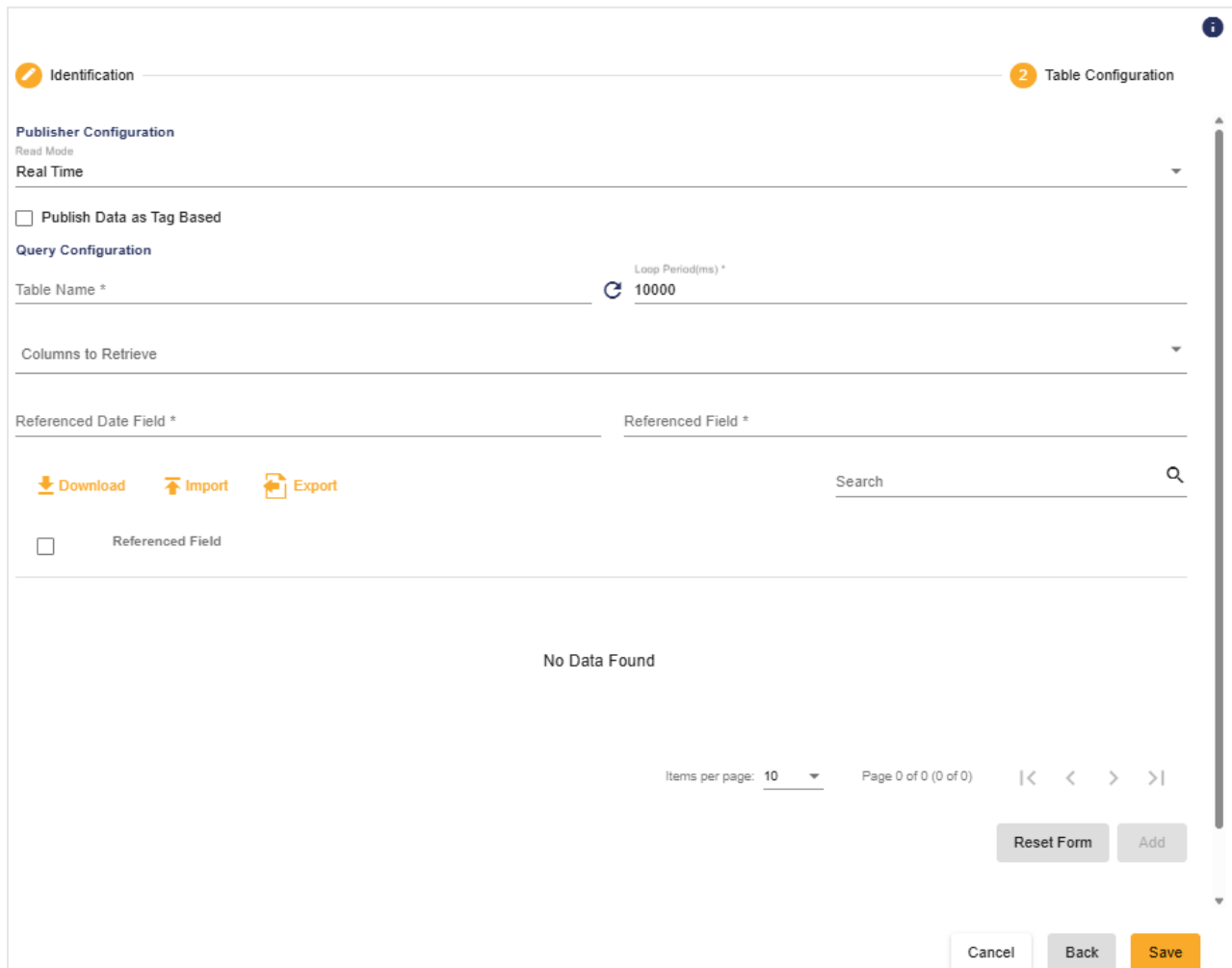
Figure 125: PI AF Connector as Destination - Tag Configuration View

5.2.7.11. PostgreSQL

Click **Next** to proceed to the **Table Configuration** page, where you can define parameters related to data mapping.

Configuration options vary depending on whether the connector is set as a **Source** or a **Destination**.

PostgreSQL Connector as Source:



The screenshot displays the 'Table Configuration' view for a PostgreSQL connector acting as a source. The interface includes a top navigation bar with 'Identification' and 'Table Configuration' tabs. The 'Table Configuration' section is divided into several functional areas:

- Publisher Configuration:** Shows 'Read Mode' set to 'Real Time'.
- Query Configuration:** Includes a 'Table Name' field and a 'Loop Period(ms)' dropdown set to '10000'.
- Referenced Date Field / Referenced Field:** Two empty text input fields for specifying date and field references.
- Actions:** Buttons for 'Download', 'Import', and 'Export' are visible.
- Search:** A search bar with a magnifying glass icon.
- Table Data:** A section labeled 'No Data Found' with a table structure.
- Footer:** Contains pagination controls ('Items per page: 10', 'Page 0 of 0 (0 of 0)'), navigation arrows, and buttons for 'Reset Form', 'Add', 'Cancel', 'Back', and 'Save'.

Figure 126: PostgreSQL Connector as Source - Table Configuration View

Parameter	Description	Default Value
Publisher Configuration		
Read Mode	<p>Specifies how data is retrieved. Available options include:</p> <ul style="list-style-type: none"> • Real Time: Retrieves the most recent values as they are updated. • Historian: Retrieves historical data from the database. 	Real Time
Publish Data as Tag Based	When enabled, data is published using tags instead of raw fields.	Unchecked
Query Configuration		
Default Query Configuration		
Table Name	Specifies the name of the source table from which data will be retrieved.	
Loop Period (ms)	Defines the refresh interval in milliseconds for retrieving data.	10000
Columns to Retrieve	Select the specific columns to fetch from the source table.	
Referenced Date Field	Field used as the reference for date or timestamp.	
Referenced Field	Field used as the reference for tag-based or key-based retrieval.	

Referenced Field Table	<p>After adding a referenced field, the data is displayed in the table. The user must select at least one item from the table to add the query. Alternatively, the user can add items using a template</p> <ul style="list-style-type: none"> - Download Template: Download a template including an example of Referenced Field. - Import Template: Import a predefined Referenced Field configuration template. - Export Template: Export the configured items for reuse. 	
Reset Form	Reset the table configuration by clicking the Reset Form button.	
Publish Data as Tag Based Configuration		
Write Mode	<p>The write mode is available only when "Publish Data as Tag Based" is enabled. All write operations are executed from the OPC UA Server destination.</p> <p>The are two write mode options:</p> <ul style="list-style-type: none"> • Insert: used to insert a new row into the table. • Update: used to update an existing row in the table. 	Insert

Query configuration		
Default Type	Select the default type for the referenced field table items.	String
Referenced Type Field	Field used as the reference for data type.	
Type Mapping	<p>The type mapping button is available only when "Referenced Type Field" is added.</p> <p>Its purpose is to map database data types with OPC UA Server data types.</p>	
Field Aliasing		
Field to map	Defines the tag-based fields (TagName, Value, Timestamp, and Quality) to be mapped to the table fields.	
Field name	Select the table field corresponding to the chosen "Field to map" .	
Historian Read Mode Query Configuration		
Start Time	Defines the start of the time range for historical data retrieval.	Current time – 10 minutes
End Time	Defines the end of the time range for historical data retrieval.	Unchecked Current time
Data Interval(ms)	Specifies how frequently new data is transmitted.	60000

Real time Offset(ms)	<p>Specifies a time offset applied when the calculated End Time exceeds the current system time (real time).</p> <p>When this condition is met, the End Time is adjusted using the formula:</p> <p>Adjusted End Time = Real Time – Real Time Offset</p> <p>The Real Time Offset must be lower than the Loop Period.</p>	2000
Restart from Last Execution Time	Resumes historical data processing from the point where the previous execution stopped.	Unchecked
Include Bounds	When enabled, bounding values are included in the results.	Unchecked
Queries		
Queries	<p>Displays the list of configured queries.</p> <p>Queries in the table can be edited or removed.</p>	

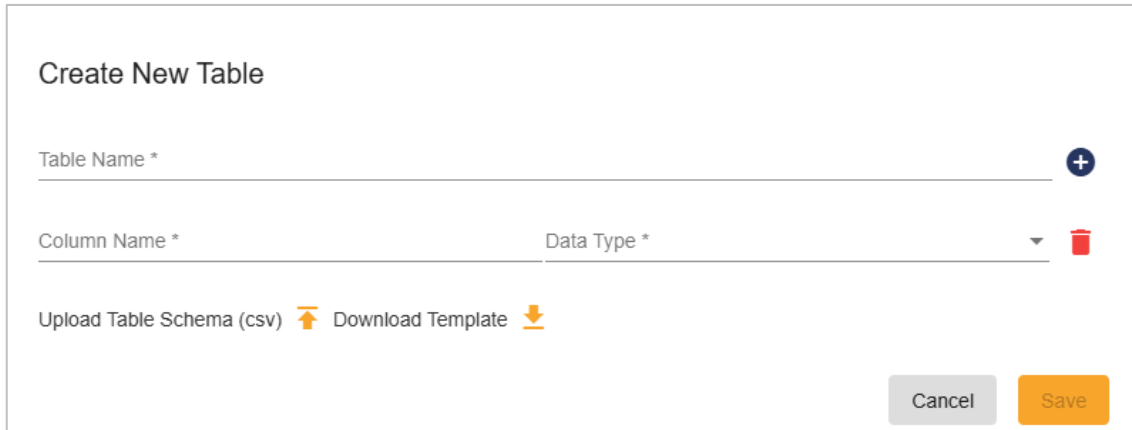
Table 82: PostgreSQL Connector as Source - Table Configuration Parameters

Once the **Query Configuration** parameters are set, you can select the tags to be read from the **Referenced Field**. Additionally, you can use the **Download**, **Export**, and **Import** options to manage and configure the list of tags associated with the **Referenced Field**.

Click **Add** and then **Save** your configuration.

PostgreSQL Connector as Destination:

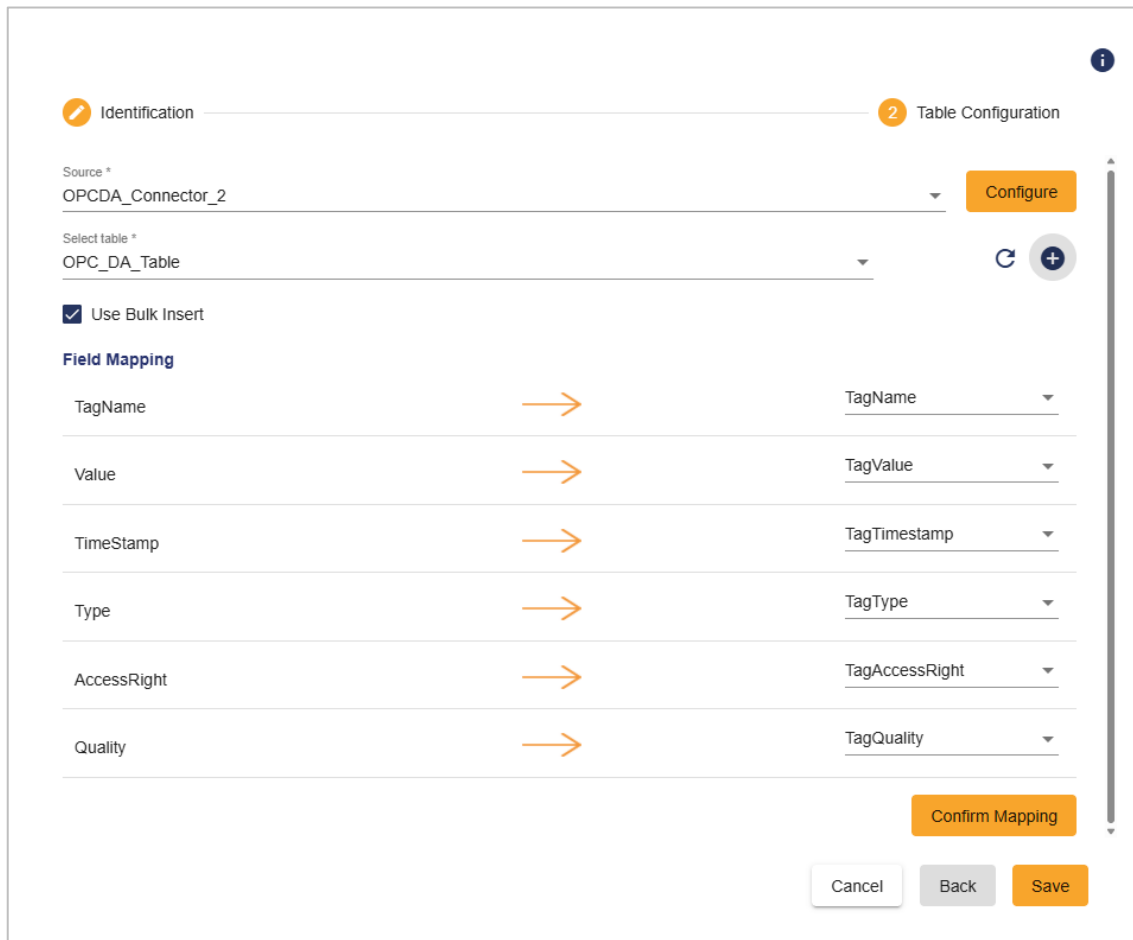
Select the **Source Connector** from which data will be received and stored in the database, along with the table that will store the data. You can choose an existing table or create a new one directly from the configuration interface.



The image shows a 'Create New Table' configuration window. It has a title bar 'Create New Table'. Below the title bar, there is a text input field for 'Table Name *' with a blue plus icon to its right. Below that, there is a table with two columns: 'Column Name *' and 'Data Type *'. The 'Data Type *' column has a dropdown arrow and a red trash icon to its right. At the bottom left, there are two links: 'Upload Table Schema (csv)' with an orange upload icon, and 'Download Template' with an orange download icon. At the bottom right, there are two buttons: 'Cancel' (grey) and 'Save' (orange).

Figure 127: PostgreSQL Connector as Destination – Create New Table Configuration View

After selecting or creating the table, map the **Source Connector** tags/fields to the corresponding table columns to ensure proper data alignment and storage structure.



The screenshot displays the configuration interface for a PostgreSQL Connector, specifically the 'Table Configuration' step (indicated by a '2' in a circle). The interface is divided into two main sections: 'Identification' and 'Table Configuration'.

Identification Section:

- Source ***: A dropdown menu showing 'OPCDA_Connector_2'.
- Select table ***: A dropdown menu showing 'OPC_DA_Table'.
- Use Bulk Insert**: A checkbox that is checked.

Field Mapping Section:

This section contains a table with two columns: 'Source Field' and 'Destination Field'. Each source field is mapped to a corresponding destination field in the target table. The mappings are as follows:

Source Field	Destination Field
TagName	TagName
Value	TagValue
TimeStamp	TagTimestamp
Type	TagType
AccessRight	TagAccessRight
Quality	TagQuality

At the bottom of the interface, there are three buttons: 'Cancel', 'Back', and 'Save'. A 'Confirm Mapping' button is also present, which is highlighted in orange.

Figure 128: PostgreSQL Connector as Destination – Table Configuration and Field Mapping Configuration View

Once the mapping is complete, click **Confirm Mapping** to validate the configuration and then **Save** your configuration.

5.2.7.12. SQL Server

The **SQL Server** requires the same core parameters to be configured during the **Identification** step. In addition, it provides extra configuration options.

Parameter	Description	Default Value
Advanced Configuration		
Payload Transformation	Available when the connector is configured as a Destination . Enables the transformation of a field-based payload into a basic tag-based payload.	Unchecked

Table 83: SQL Server Additional Configuration Parameters

Click **Next** to proceed to the **Table Configuration** page, where you can define parameters related to data mapping.

Configuration options vary depending on whether the connector is set as a **Source** or a **Destination**.

SQL Server Connector as Source:

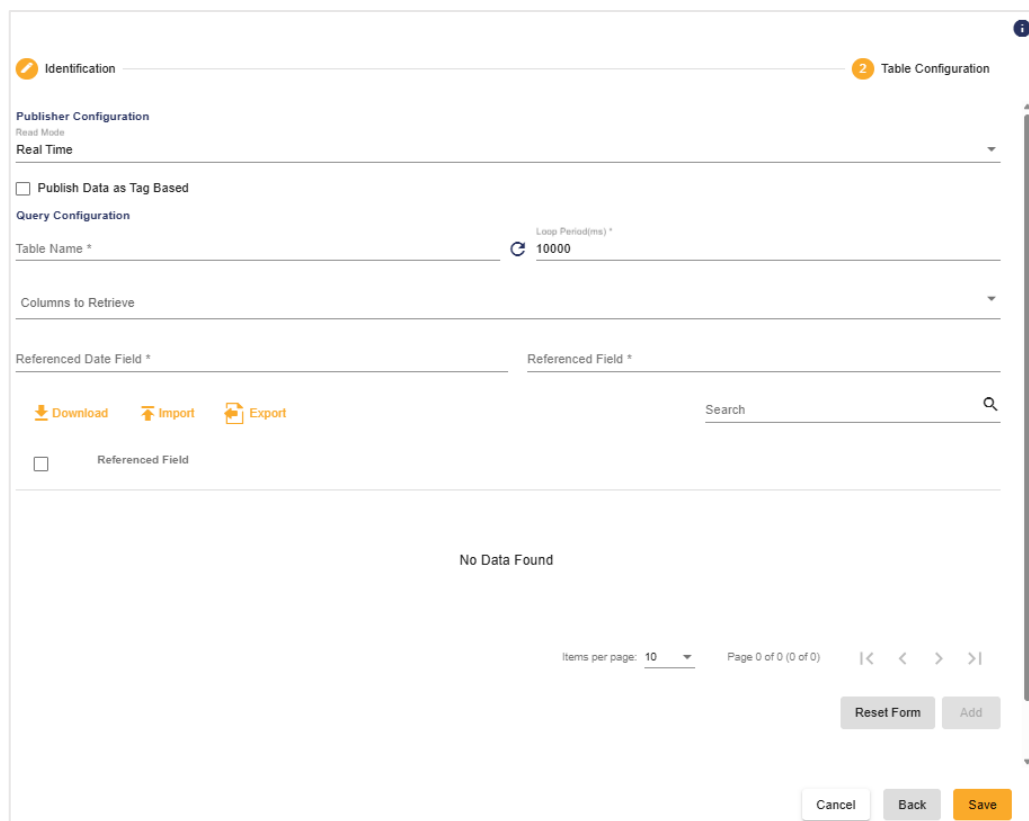


Figure 129: SQL Server Connector as Source - Table Configuration View

Parameter	Description	Default Value
<i>Publisher Configuration</i>		
<i>Read Mode</i>	<p>Specifies how data is retrieved. Available options include:</p> <ul style="list-style-type: none"> • Real Time: Retrieves the most recent values as they are updated. • Historian: Retrieves historical data from the database. 	Real Time
<i>Publish Data as Tag Based</i>	When enabled, data is published using tags instead of raw fields.	Unchecked
<i>Query Configuration</i>		
<i>Default Query Configuration</i>		
<i>Table Name</i>	Specifies the name of the source table from which data will be retrieved.	
<i>Loop Period (ms)</i>	Defines the refresh interval in milliseconds for retrieving data.	10000
<i>Columns to Retrieve</i>	Select the specific columns to fetch from the source table.	
<i>Referenced Date Field</i>	Field used as the reference for date or timestamp.	
<i>Referenced Field</i>	Field used as the reference for tag-based or key-based retrieval.	

Referenced Field Table	<p>After adding a referenced field, the data is displayed in the table. The user must select at least one item from the table to add the query. Alternatively, the user can add items using a template</p> <ul style="list-style-type: none"> - Download Template: Download a template including an example of Referenced Field. - Import Template: Import a predefined Referenced Field configuration template. - Export Template: Export the configured items for reuse. 	
Reset Form	Reset the table configuration by clicking the Reset Form button.	
Publish Data as Tag Based Configuration		
Write Mode	<p>The write mode is available only when "Publish Data as Tag Based" is enabled. All write operations are executed from the OPC UA Server destination.</p> <p>The are two write mode options:</p> <ul style="list-style-type: none"> • Insert: used to insert a new row into the table. • Update: used to update an existing row in the table. 	Insert

Query configuration		
Default Type	Select the default type for the referenced field table items.	String
Referenced Type Field	Field used as the reference for data type.	
Type Mapping	<p>The type mapping button is available only when "Referenced Type Field" is added.</p> <p>Its purpose is to map database data types with OPC UA Server data types.</p>	
Field Aliasing		
Field to map	Defines the tag-based fields (TagName, Value, Timestamp, and Quality) to be mapped to the table fields.	
Field name	Select the table field corresponding to the chosen " Field to map ".	
Historian Read Mode Query Configuration		
Start Time	Defines the start of the time range for historical data retrieval.	Current time – 10 minutes
End Time	Defines the end of the time range for historical data retrieval.	Unchecked Current time
Data Interval(ms)	Specifies how frequently new data is transmitted.	60000

<i>Real time Offset(ms)</i>	<p>Specifies a time offset applied when the calculated End Time exceeds the current system time (real time).</p> <p>When this condition is met, the End Time is adjusted using the formula:</p> <p>Adjusted End Time = Real Time – Real Time Offset</p> <p>The Real Time Offset must be lower than the Loop Period.</p>	2000
<i>Restart from Last Execution Time</i>	Resumes historical data processing from the point where the previous execution stopped.	Unchecked
<i>Include Bounds</i>	When enabled, bounding values are included in the results.	Unchecked
<i>Queries</i>		
<i>Queries</i>	<p>Displays the list of configured queries.</p> <p>Queries in the table can be edited or removed.</p>	

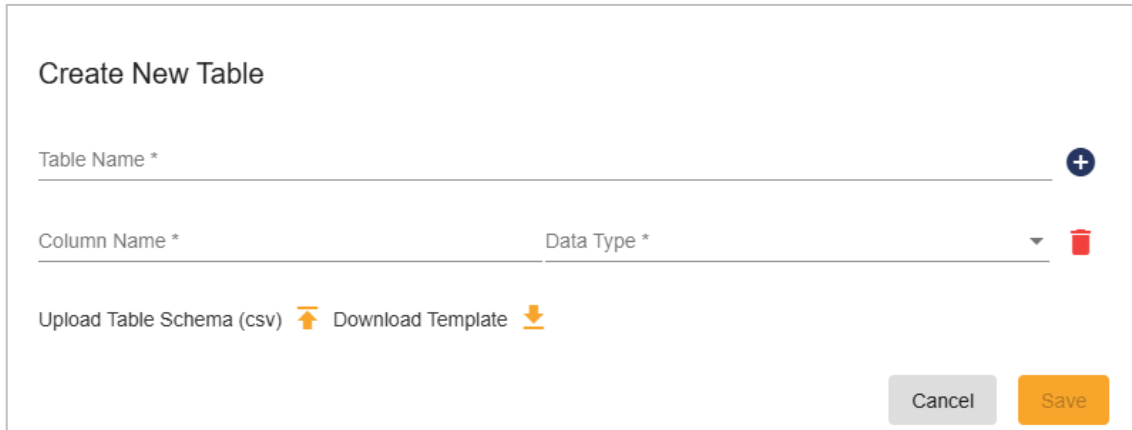
Table 84: SQL Server Connector as Source - Table Configuration Parameters

Once the **Query Configuration** parameters are set, you can select the tags to be read from the **Referenced Field**. Additionally, you can use the **Download**, **Export**, and **Import** options to manage and configure the list of tags associated with the **Referenced Field**.

Click **Add** and then **Save** your configuration.

SQL Server Connector as Destination:

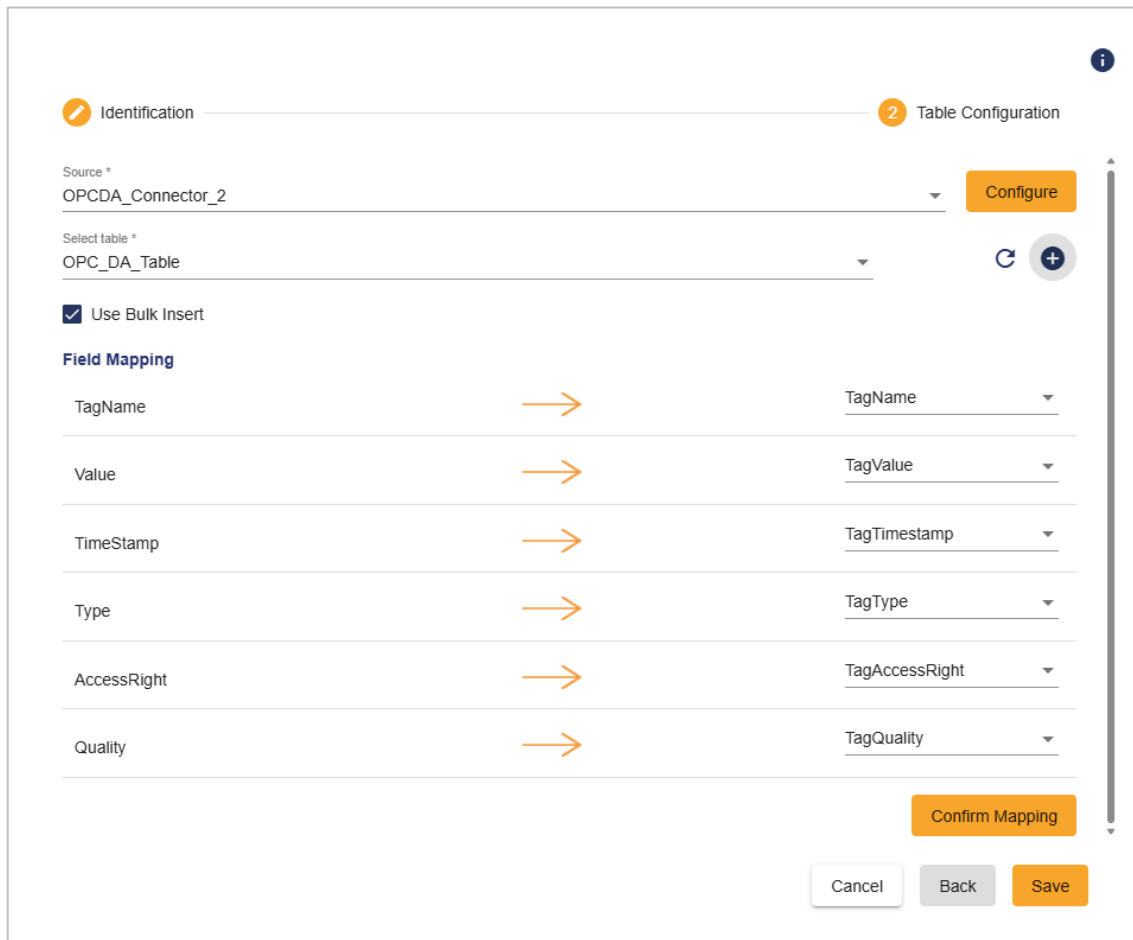
Select the **Source Connector** from which data will be received and stored in the database, along with the table that will store the data. You can choose an existing table or create a new one directly from the configuration interface.



The image shows a 'Create New Table' configuration window. It has a title bar 'Create New Table'. Below it is a text input field for 'Table Name *' with a blue plus icon to its right. Below that is a table with two columns: 'Column Name *' and 'Data Type *'. The 'Data Type *' column has a dropdown arrow and a red trash icon to its right. At the bottom left, there are two links: 'Upload Table Schema (csv)' with an orange upload icon and 'Download Template' with an orange download icon. At the bottom right, there are two buttons: 'Cancel' (grey) and 'Save' (orange).

Figure 130: SQL Server Connector as Destination – Create New Table Configuration View

After selecting or creating the table, map the **Source Connector** tags/fields to the corresponding table columns to ensure proper data alignment and storage structure.



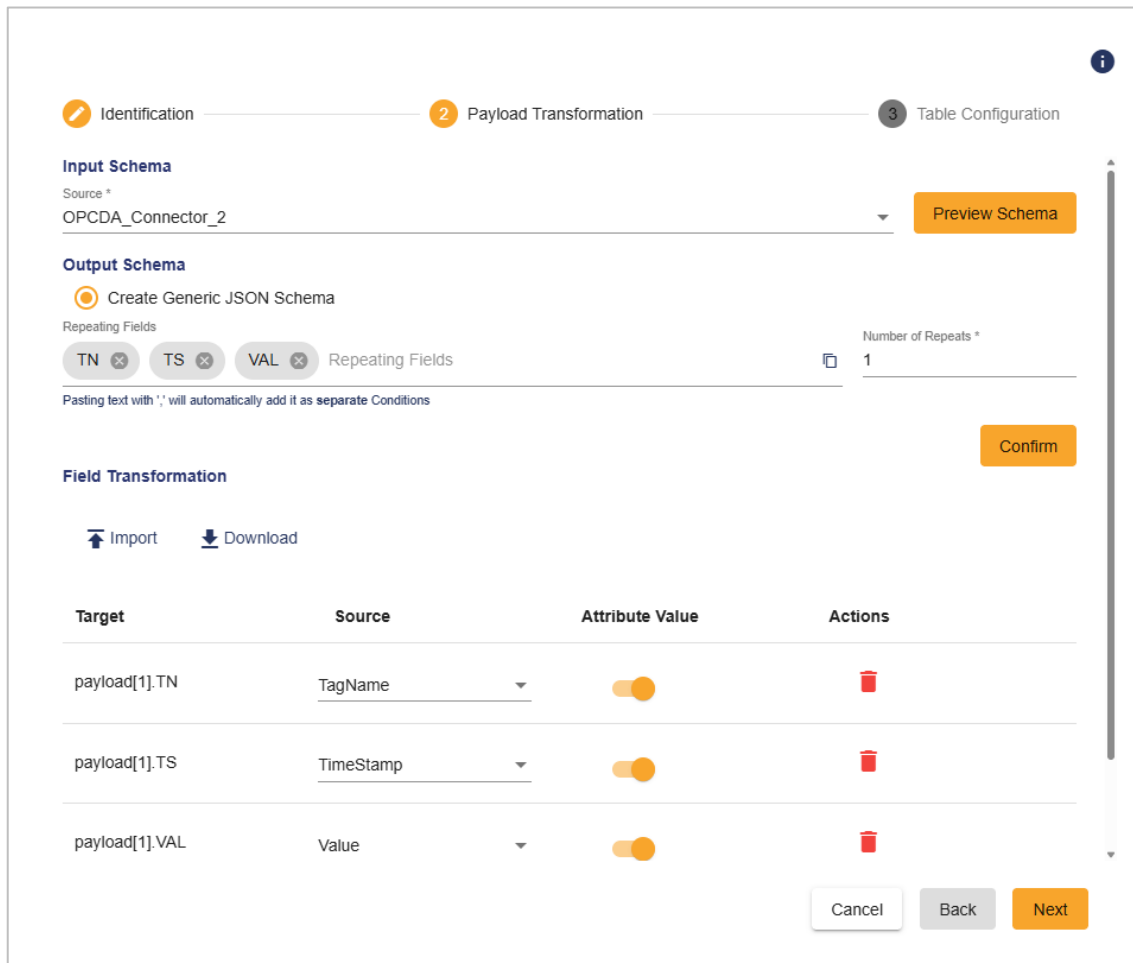
The screenshot displays the 'Table Configuration' step (labeled '2') of the configuration process. It shows the 'Source' as 'OPCDA_Connector_2' and the 'Select table' as 'OPCDA_Table'. A 'Configure' button is next to the source. Below this, the 'Use Bulk Insert' checkbox is checked. The 'Field Mapping' section lists source fields on the left and target fields on the right, connected by orange arrows. The source fields are TagName, Value, TimeStamp, Type, AccessRight, and Quality. The target fields are TagName, TagValue, TagTimestamp, TagType, TagAccessRight, and TagQuality. A 'Confirm Mapping' button is at the bottom right of the mapping section. At the very bottom, there are 'Cancel', 'Back', and 'Save' buttons.

Source Field	Target Field
TagName	TagName
Value	TagValue
TimeStamp	TagTimestamp
Type	TagType
AccessRight	TagAccessRight
Quality	TagQuality

Figure 131: SQL Server Connector as Destination – Table Configuration and Field Mapping Configuration View

Once the mapping is complete, click **Confirm Mapping** to validate the configuration and then **Save** your configuration.

If the **Payload Transformation** option is enabled, you can define transformation logic to modify the structure or content of the payload before it is written or processed. This feature allows you to adapt payload formats, apply mappings, or inject additional data to meet the requirements of the target system or workflow.



Identification — **2** Payload Transformation — 3 Table Configuration

Input Schema

Source *
OPCDA_Connector_2

Output Schema

Create Generic JSON Schema

Repeating Fields

TN TS VAL Repeating Fields

Number of Repeats *
1

Pasting text with ',' will automatically add it as separate Conditions

Field Transformation

Import Download

Target	Source	Attribute Value	Actions
payload[1].TN	TagName	<input checked="" type="checkbox"/>	
payload[1].TS	TimeStamp	<input checked="" type="checkbox"/>	
payload[1].VAL	Value	<input checked="" type="checkbox"/>	

Confirm

Cancel Back Next

Figure 132: SQL Server Connector as Destination - Payload Transformation

5.2.8. Brokers

SIOTH Brokers act as intermediary servers that receive messages from SIOTH publish connectors or third-party applications and route them to the appropriate destination clients. Destination clients can include other SIOTH subscribe connectors or external applications.

5.2.8.1. OPC UA Server

To configure the **OPC UA Server**, follow the four below:

5.2.8.1.1. UA Server Configuration

Specify the communication and security settings for the SIOTH OPC UA Server, including TCP and HTTPS endpoints, security modes, and user authentication parameters.

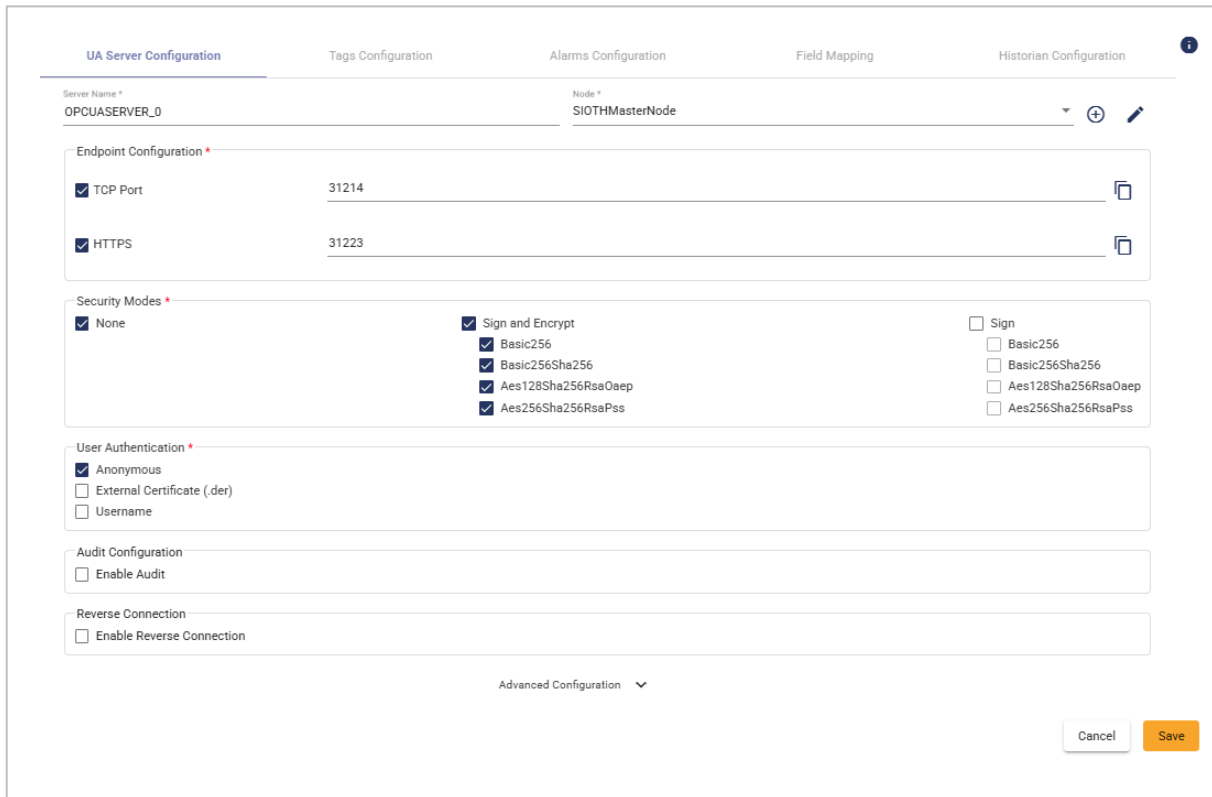


Figure 133: OPC UA Server Broker - UA Server Configuration View

Parameter	Description	Default Value
Server Name	Specifies the name of the OPC UA Server.	OPCUAServer_0
Node	Identifies the SIOTH node where the OPC UA Server will be deployed.	SIOTHMasterNode
Endpoint Configuration		

TCP Port	Defines the port used by the OPC UA Server. Clients append this port to the controller's TCP URL to establish a connection.	Automatically Generated
HTTPS	Defines the port used for HTTPS connections. Clients append this port to the controller's HTTPS URL to connect securely.	Automatically Generated
Security Modes		
None	No cryptographic protection is applied.	Checked
Sign and Encrypt	<p>Defines algorithms for signing and encryption:</p> <ul style="list-style-type: none"> • Basic256: 256-bit encryption • Basic256Sha256: SHA256 signatures with 256-bit encryption • Aes128Sha256RsaOaep: AES 128-bit encryption with SHA256 and RSA-OAEP • Aes256Sha256RsaPss: AES 256-bit encryption with SHA256 and RSA-PSS 	Checked
Sign	<p>Defines algorithms for signing only:</p> <ul style="list-style-type: none"> • Basic256: 256-bit signing • Basic256Sha256: SHA256 signatures with 256-bit encryption • Aes128Sha256RsaOaep: AES 128-bit signing with SHA256 and RSA-OAEP • Aes256Sha256RsaPss: AES 256-bit signing with SHA256 and RSA-PSS 	Unchecked
User Authentication		

<i>Anonymous</i>	No user identity required.	Checked
<i>External Certificate (.der)</i>	Enables authentication with an external certificate. File path must be specified.	Unchecked
<i>Username</i>	Enables username/password authentication. Multiple users can be configured.	Unchecked
<i>Audit Configuration</i>		
<i>Enable Audit</i>	Activates auditing for the OPC UA Server to track and log relevant actions and events.	Unchecked
<i>Reverse Connection</i>		
<i>Enable Reverse Connection</i>	Allows the server to establish a connection to the client instead of the client initiating the connection.	Unchecked
<i>Client Parameters</i>		
<i>Client Endpoint URL</i>	Specifies the endpoint URL that the client will use to connect to the server.	opc.tcp://
<i>Timeout (ms)</i>	Maximum time in milliseconds the client waits for a response from the server.	30000
<i>Max Session Count</i>	Maximum number of concurrent sessions allowed for the client.	0
<i>Server Configuration</i>		
<i>Connection Interval (ms)</i>	Interval in milliseconds between connection attempts or keep-alive messages.	15000
<i>Connection Timeout (ms)</i>	Maximum time in milliseconds the server waits for a client to establish a connection.	30000

<i>Reject Timeout (ms)</i>	Time in milliseconds before rejecting a connection attempt if the server is busy or unresponsive.	60000
<i>Advanced Configuration</i>		
<i>Data Encryption</i>	Secures transmitted data between cloud and system components.	Checked
<i>Offline</i>	Allows downloading the connector package for manual or offline deployment. Refer to the Offline Deployment section for more details about how to deploy a connector in offline mode.	Unchecked
<i>Store and Forward</i>	Enables data backlog when communication fails; automatically resends once reconnected.	Unchecked
<i>MSMQ Configuration</i>	<p>Creates an MSMQ queue for temporary message storage. Configuration includes:</p> <ul style="list-style-type: none"> • IP Address: Machine hosting the MSMQ (default: 127.0.0.1). • Queue Label: Queue Name (default: private\$\DemoRun). • MSMQ Data Encryption: Enables message encryption (default: checked). 	Unchecked
<i>Service Configuration</i>		
<i>Service Configuration</i>	Defines parameters for service setup and behavior.	Checked
<i>Logon Type</i>	<p>Service logon type:</p> <ul style="list-style-type: none"> • Local System • Specific Account 	Local System

Startup Type	Defines service start mode: <ul style="list-style-type: none"> Automatic Automatic (Delayed Start) Manual Disabled 	Automatic
Data Transfer Mode		
Data Request Port	Port used for data request operations.	
Client ID	Unique identifier for each server connecting to the master broker.	
Log Settings		
Auto Append	Automatically appends new entries to existing log files.	Checked
Log Level	Severity of log entries: <ul style="list-style-type: none"> Information Debug 	Information
Buffer size (MB)	Memory allocated for temporary data storage before processing.	100
Maximum Files	Maximum number of retained log files.	10
Log File Max Size (MB)	Maximum allowed log file size.	10
Auto Save Timeout (s)	Time before data is automatically saved.	5
Advanced Server Configuration		

<i>Auto Accept Untrusted Certificates</i>	Automatically accepts untrusted certificates into the Trusted Certificates folder.	Checked
<i>Minimum Certificate Key Size (Bits)</i>	Minimum RSA key size for certificates.	2048
<i>Operation Timeout (ms)</i>	Maximum response time for operation requests.	600000
<i>Max String Length</i>	Maximum allowed string length.	1048576
<i>Max Byte String Length</i>	Maximum allowed byte string length.	1048576
<i>Max Array Length</i>	Maximum number of elements in an array.	65535
<i>Max Session Count</i>	Maximum number of concurrent sessions.	100
<i>Diagnostics Enabled</i>	Collects diagnostic information when enabled.	Unchecked
<i>Max Message Size (Byte)</i>	Maximum message size the server can process.	4194304
<i>Channel Lifetime (ms)</i>	Maximum duration a communication channel remains open.	300000
<i>Min Request Thread Count</i>	Minimum threads available to process incoming requests.	5

<i>Max Queued Request Count</i>	Maximum queued requests when threads are busy.	2000
<i>Max Request Thread Count</i>	Maximum threads for processing requests.	100
<i>Min Session Timeout (ms)</i>	Minimum idle time before session termination.	10000
<i>Max Session Timeout (ms)</i>	Maximum idle time before automatic session termination.	3600000
<i>Max Subscription Lifetime (ms)</i>	Maximum time a subscription can remain active without published requests.	3600000
<i>Min Subscription Lifetime (ms)</i>	Minimum time a subscription must remain active before closure or renewal.	10000
<i>Max Notification Queue Size</i>	Maximum notifications queued before overflow.	100
<i>Max Notifications per Publish</i>	Maximum notifications sent per publish request.	1000
<i>Min Publishing Interval (ms)</i>	Minimum interval between published notifications.	100
<i>Max Subscription Count</i>	Maximum subscriptions per client.	100

Max Event Queue Size	Maximum events in the queue.	10000
Max Publish Request Count	Maximum simultaneous publishing requests.	1000
Default Session Timeout (ms)	Default session timeout if unspecified by client.	600000
Max Request Age (ms)	Maximum age of a pending request before expiration.	600000
Max Publishing Interval (ms)	Maximum time between subscription updates.	3600000
Max Message Queue Size	Maximum queued messages before discard or delay.	100
Max Buffer Size (Byte)	Maximum size of temporary storage buffers.	65535
Reset Configuration	Resets all advanced server configuration parameters to default.	

Table 85: OPC UA Server Broker - UA Server Configuration Parameters

5.2.8.1.2. Tags Configuration

Configure the **OPC UA Server address space** by adding tags from source connectors or importing them from a preconfigured CSV file. The CSV must follow the SIOTH OPC UA Server format.

(!) Note

The SIOTH OPC UA Server automatically imports tags from the linked source connectors.

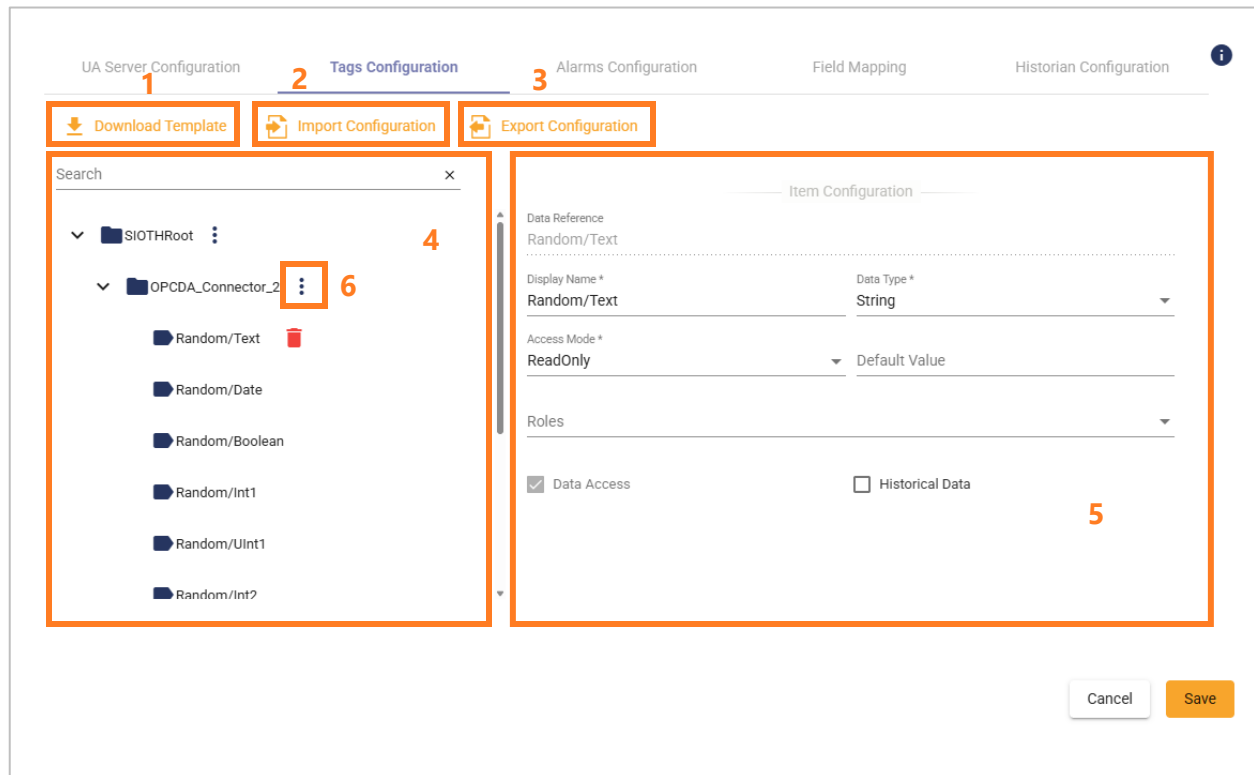


Figure 134: OPC UA Server Broker - Tags Configuration View

The following actions are available:

- **Download Template (1):** Download a CSV template as a reference for tag configuration.

Tags Parameters

	A	B	C	D	E	F	G	H	I	J	K	L
1	SIOTHPoint	SubscriptionName	SourceConnector	IsHDA	AccessRight	Type	DefaultValue	DataRef	CustomModelID	DateTime	TagName	Roles
2	Folder-TagInt1		OPC_Connector	FALSE	ReadOnly	Int32	0	undefined-TagInt1			TagInt1	Operator
3	Folder-TagInt2		OPC_Connector	TRUE	WriteOnly	Int32	0	undefined-TagInt2			TagInt2	Operator
4	Folder-TagInt3		DNP_Connector	TRUE	ReadWrite	Int32	0	undefined-TagInt3			TagInt3	Operator
5	Folder-TagUInt1		OPCUA_Connector	TRUE	ReadOnly	String	0	undefined-TagUInt1			TagUInt1	Operator
6	Folder-TagUInt2		OPC_Connector	TRUE	WriteOnly	String	0	undefined-TagUInt2			TagUInt2	Operator
7	Folder-TagUInt3		Modbus_Connector	FALSE	ReadWrite	String	0	undefined-TagUInt3			TagUInt3	Operator
8	Folder-TagBstr1		OPC_Connector	FALSE	ReadOnly	String		undefined-TagBstr1			TagBstr1	Operator
9	Folder-TagBstr2		Modbus_Connector	TRUE	WriteOnly	String		undefined-TagBstr2			TagBstr2	Operator
10	Folder-TagI1		OPC_Connector	TRUE	ReadWrite	SByte	0	undefined-TagI1			TagI1	Operator
11	Folder-TagUI1		IEC_Connector	FALSE	WriteOnly	Byte	0	undefined-TagUI1			TagUI1	Operator
12	Folder-TagI2		IEC_Connector	TRUE	ReadOnly	Int16	0	undefined-TagI2			TagI2	Operator
13	Folder-TagUI2		IEC_Connector	TRUE	ReadWrite	UInt16	0	undefined-TagUI2			TagUI2	Operator
14	Folder-TagI4		OPCUA_Connector	TRUE	ReadOnly	Int32	0	undefined-TagI4			TagI4	Operator
15	Folder-TagUI4		OPC_Connector	TRUE	ReadWrite	UInt32	0	undefined-TagUI4			TagUI4	Operator
16	Folder-TagR4		OPC_Connector	FALSE	ReadWrite	Float	0	undefined-TagR4			TagR4	Operator
17	Folder-TagR8		DNP_Connector	TRUE	ReadWrite	Float	0	undefined-TagR8			TagR8	Operator
18	Folder-TagCY		DNP_Connector	TRUE	WriteOnly	String	0	undefined-TagCY			TagCY	Operator
19	Folder-TagDATE		OPC_Connector	TRUE	ReadOnly	DateTime		undefined-TagDATE		MM/dd/yyyy HH:mm:ss	TagDATE	Operator
20	Folder-TagBOOL1		OPC_Connector	TRUE	ReadOnly	Boolean		undefined-TagBOOL1			TagBOOL1	Operator
21	Folder-TagBOOL2		OPC_Connector	TRUE	ReadWrite	Boolean		undefined-TagBOOL2			TagBOOL2	Operator

Tags Path

Figure 135: OPC UA Server Broker - Tags CSV Template

Parameter	Description
<i>SIOTHPoint</i>	Tag path within SIOTH.
<i>InstrumentTag</i>	Tag name in the source system.
<i>SubscriptionName</i>	Name of the subscription to the OPC UA Server.
<i>SourceConnector</i>	Data source (OPC DA, Modbus, SNMP, DNP, etc.).
<i>IsHDA</i>	Enable historical data storage (TRUE/FALSE).
<i>AccessRight</i>	Tag access mode: ReadWrite, ReadOnly, WriteOnly.
<i>Type</i>	Data type: String, SByte, Byte, Int16, UInt16, Int32, UInt32, Int64, UInt64, Float, Double, DateTime, Boolean.
<i>DefaultValue</i>	Default value for the tag.
<i>DataRef</i>	Tag data reference.
<i>IsAE</i>	Enable alarms for this tag (TRUE/FALSE).
<i>CustomNodeID</i>	Optional custom Node ID.
<i>TagName</i>	Name of the tag.
<i>CustomAttributes</i>	Names of custom attributes, separated by semicolons (";").
<i>CustomAttributesValues</i>	Lists the values corresponding to each custom attribute, separated by semicolons (";").
<i>CustomAttributesTypes</i>	Lists the types of the custom attributes, separated by semicolons (";").
<i>MaxRange</i>	Maximum allowed value.
<i>MinRange</i>	Minimum allowed value.

<i>HighHigh</i>	Upper critical limit: exceeding this value triggers a HighHigh alarm .
<i>High</i>	Upper warning limit: exceeding this value triggers a High alarm .
<i>LowLow</i>	Lower critical limit; dropping below this value triggers a LowLow alarm .
<i>Low</i>	Lower warning limit; dropping below this value triggers a Low alarm .
<i>OutOfRangeSeverity</i>	Severity level assigned when the value is outside the defined min/max range.
<i>HighHighSeverity</i>	Severity level of the HighHigh alarm .
<i>HighSeverity</i>	Severity level of the High alarm .
<i>LowLowSeverity</i>	Severity level of the LowLow alarm .
<i>LowSeverity</i>	Severity level of the Low alarm .
<i>OutOfRangeMessage</i>	Message text displayed when the value is out of the defined range.
<i>HighHighMessage</i>	Message displayed when a HighHigh alarm is triggered.
<i>HighMessage</i>	Message displayed when a High alarm is triggered.
<i>LowLowMessage</i>	Message displayed when a LowLow alarm is triggered.
<i>LowMessage</i>	Message displayed when a Low alarm is triggered.
<i>isOutOfRangeChecked</i>	Indicates whether OutOfRange condition checking is enabled or not.
<i>isHighHighChecked</i>	Indicates whether HighHigh alarm checking is enabled or not.
<i>isHighChecked</i>	Indicates whether High alarm checking is enabled or not.

<i>isLowLowChecked</i>	Indicates whether LowLow alarm checking is enabled or not.
<i>isLowChecked</i>	Indicates whether Low alarm checking is enabled or not.
<i>OutOfRangeAckMessage</i>	Message displayed when an OutOfRange alarm is acknowledged by the user.
<i>HighHighAckMessage</i>	Acknowledgment message for a HighHigh alarm .
<i>HighAckMessage</i>	Acknowledgment message for a High alarm .
<i>LowLowAckMessage</i>	Acknowledgment message for a LowLow alarm .
<i>LowAckMessage</i>	Acknowledgment message for a Low alarm .
<i>AlarmValueForBooleanTag</i>	Boolean value (TRUE/FALSE) that triggers the alarm for Boolean-type tags.
<i>AlarmSeverityForBooleanTag</i>	Severity level associated with the Boolean alarm condition.
<i>AlarmMessageForBooleanTag</i>	Message displayed when the Boolean alarm condition occurs.
<i>AlarmAckMessageForBooleanTag</i>	Message displayed when the Boolean alarm is acknowledged.

Table 86: OPC UA Server Broker - Tags Template Parameters

- **Import configuration (2):** Import tags from CSV.
- **Export Configuration (3):** Export configured tags to CSV.
- **Server Address Space Viewer (4):** Browse OPC UA address space tree.
- **Item Configuration (5):** View and update selected tag parameters.
- **Three-dot Icon Buttons (6):** Add/remove folders or tags in the address space.

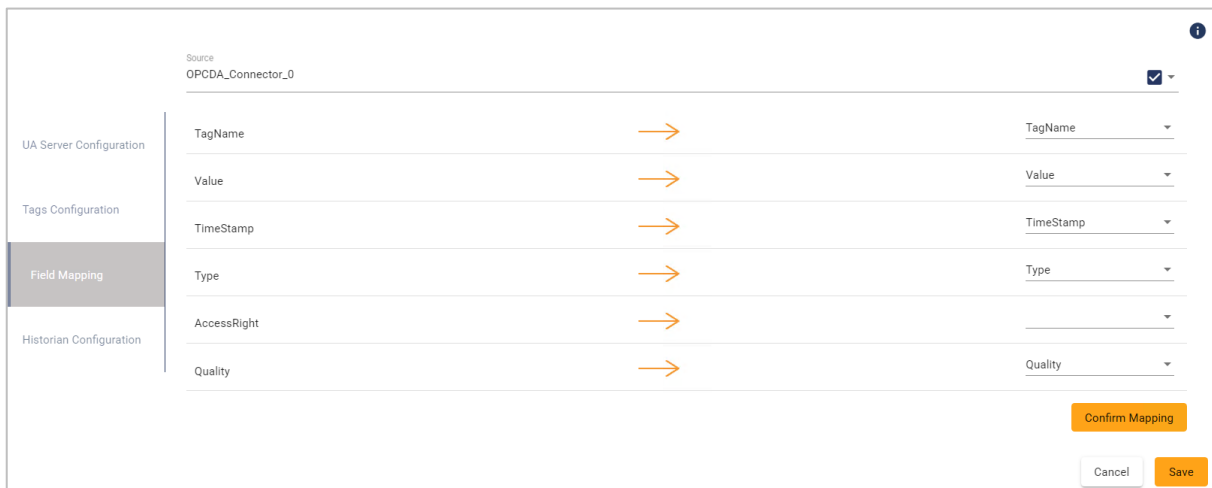
When clicking the tag, the **Item Configuration** section displays all the parameters associated with the tag.

Parameter	Description	Default Value
Item Configuration		
Data Reference	Defines explicit relationship from one node to another.	
Display Name	Tag name displayed in SIOTH.	
Data Type	Data type of the tag.	
Access Mode	ReadOnly, WriteOnly, or ReadWrite.	ReadWrite
Default Value	Default value for the tag.	
Data Access	Enables access to the data when checked.	Checked
Historical Data	Enables access to the historical data when checked.	Unchecked
Alarms & Conditions	Enables alarm and condition configuration.	Unchecked
Alarm Configuration		
Alarm Configuration	Define alarm thresholds to generate alarms such as High High, High, Low, and Low Low.	
Custom Attributes		
Add custom Attributes	Add a custom attribute by specifying its Name , Value , and Type .	
Clear All	Remove all existing custom attributes.	

Table 87: OPC UA Server Broker - Tags Parameters

5.2.8.1.3. Fields Mapping

Map fields from source connectors to the OPC UA Server. Default mappings are provided and can be modified. Click **Confirm Mapping** to validate.



Source	Target	Checkbox
OPCDA_Connector_0		<input checked="" type="checkbox"/>
TagName	TagName	<input type="checkbox"/>
Value	Value	<input type="checkbox"/>
TimeStamp	TimeStamp	<input type="checkbox"/>
Type	Type	<input type="checkbox"/>
AccessRight		<input type="checkbox"/>
Quality	Quality	<input type="checkbox"/>

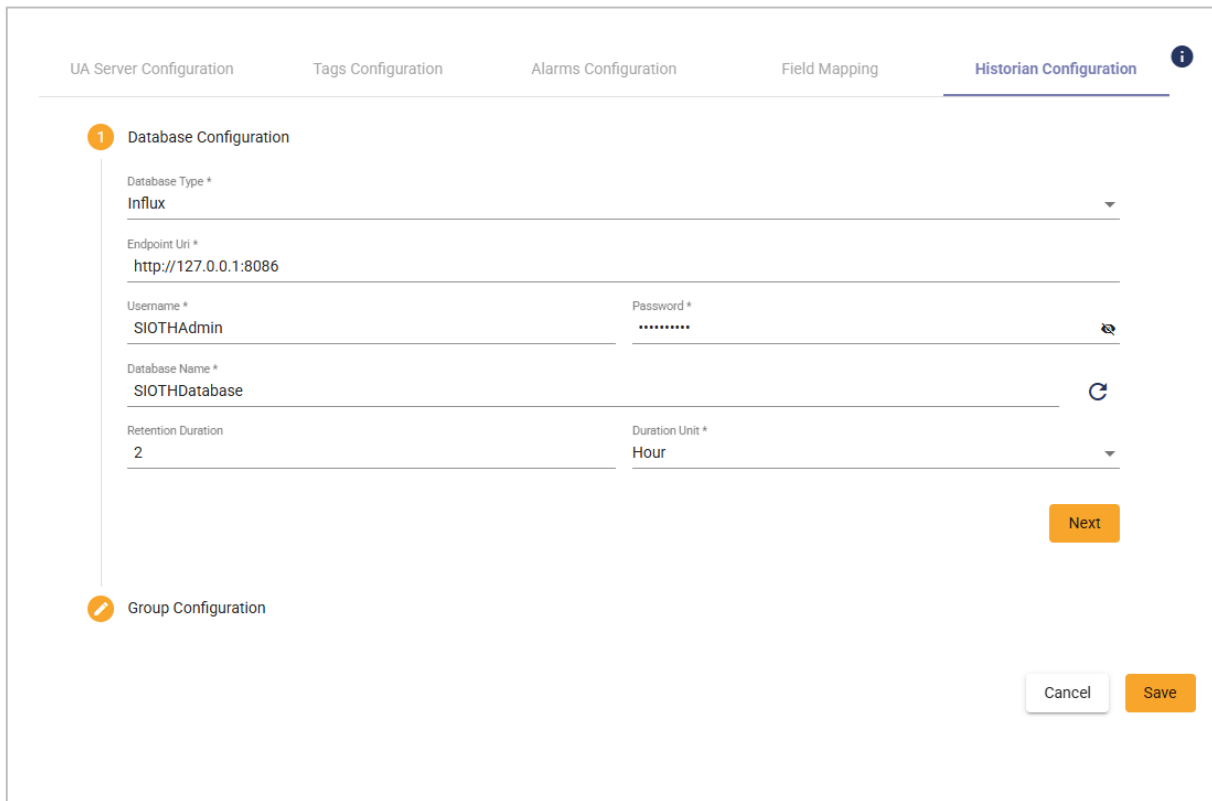
Confirm Mapping

Cancel Save

Figure 136: OPC UA Server Broker - Fields Mapping View

5.2.8.1.4. Historian Configuration

The OPC UA Server includes a built-in historian supporting OPC UA Historical Access (HA). Configure historian settings via the **Historian Configuration** tab.



UA Server Configuration Tags Configuration Alarms Configuration Field Mapping **Historian Configuration** ⓘ

1 Database Configuration

Database Type *
Influx

Endpoint Uri *
http://127.0.0.1:8086

Username * Password *
SIOTHAdmin

Database Name *
SIOTHTDatabase

Retention Duration Duration Unit *
2 Hour

Next

Group Configuration

Cancel Save

Figure 137: OPC UA Server Broker - Historian Configuration View

Parameter	Description	Default Value
Database Type	Specifies the historian database. Options: Influx, SQL Server, MySQL, PostgreSQL.	Influx
Database Type = Influx		
Endpoint URL	URL of Influx database for archiving.	http://127.0.0.1:8086
Username	Username required for the InfluxDB authentication.	SIOTHAdmin
Password	Password associated with the username and required for the InfluxDB authentication.	
Database Name	Name of database in selected system.	SIOTHTDatabase

<i>Retention Duration</i>	How long data will be retained before automatic deletion.	2
<i>Duration Unit</i>	<p>The time unit is used for retention duration.</p> <p>Supported values are:</p> <ul style="list-style-type: none"> • Hour. • Day. • Week. 	Hour
<i>AE Measurement</i>	Names of measurements for alarms/events data.	HistoricalAlarmAndConditionOPCUASERVER_4
<i>HDA Measurement</i>	Names of measurements for historical data.	HistoricalDataOPCUASERVER_4
<i>Database Type = SQL Server</i>		
<i>Server Name</i>	<p>Specifies the SQL Server instance name.</p> <ul style="list-style-type: none"> • Default instance: use the machine name. • Named instance: use the format <computer_name>\<instance_name> (e.g., DBSRVR\SQLEXPRESS). 	
<i>Authentication Mode</i>	<p>Defines the authentication mode to connect to SQL Server. Options:</p> <ul style="list-style-type: none"> • Windows Authentication: User identity is confirmed by Windows. • SQL Server Authentication: Requires login and password. 	Windows Authentication

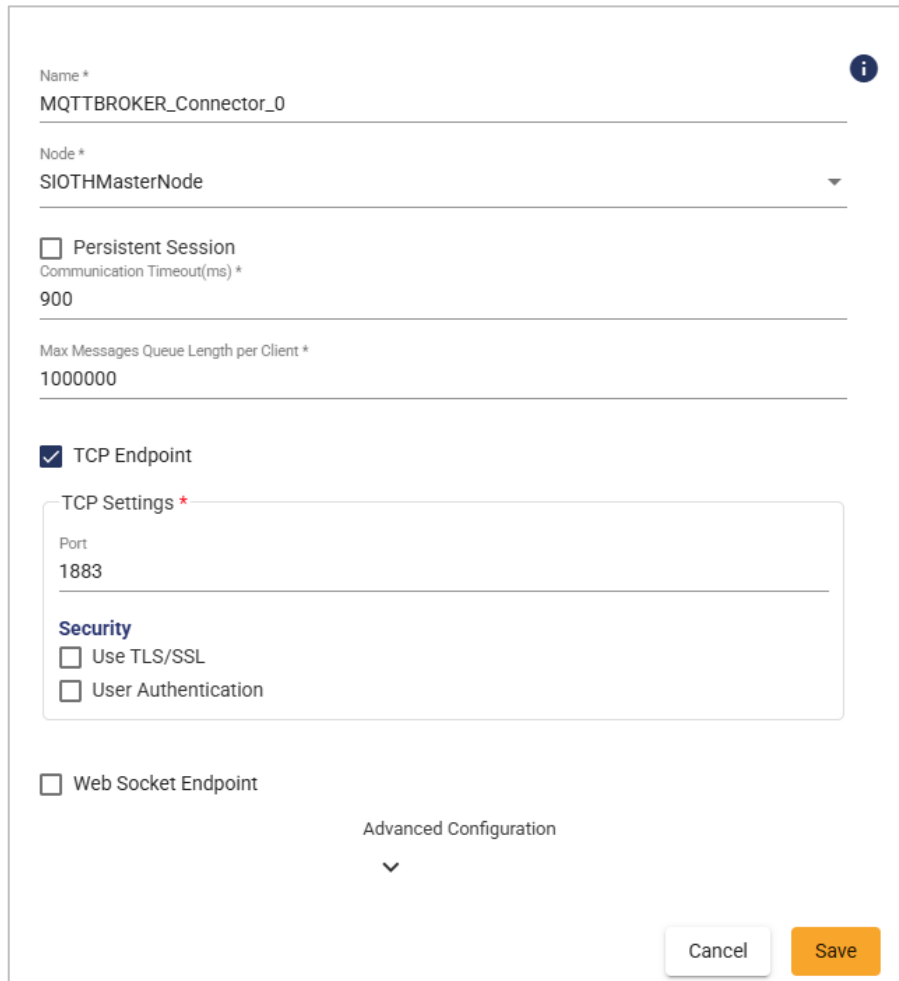
Database Name	Target database name. Click the refresh icon to retrieve and browse available databases.	
AE Table	Name of the table for storing alarm and event (AE) data.	HistoricalAlarmAndConditionOPCUASERVER_4
HDA Table	Name of the table used to store historical data (HDA) from tags.	HistoricalDataOPCUASERVER_4
Use Bulk Insert	Enables inserting multiple rows in a single operation for better performance with large datasets. May increase memory usage.	Checked
Database Type = MySQL		
Server Name	Name of the MySQL server instance.	
Port	Port used by the database client to connect to the server.	3306
Username	Username required for authentication.	
Password	Password associated with the username.	
Database Name	Name of the database on the server.	
AE Table	Name of table used to store alarm and event (AE) data.	HistoricalAlarmAndConditionOPCUASERVER_xx
HDA Table	Name of table used to store HDA data.	HistoricalDataOPCUASERVER_xx

Use Bulk Insert	Enables inserting multiple rows in a single operation for better performance with large datasets. May increase memory usage.	Checked
Database Type = PostgreSQL		
Server Name	Name of the PostgreSQL server instance.	
Port	Port used by the database client to connect to the server.	5432
Authentication Mode	<p>Mode used to connect to the database. Options:</p> <ul style="list-style-type: none"> • Windows Authentication: User identity confirmed by Windows. • Standard Authentication: Requires login and password. 	Windows Authentication
Database Name	Name of the database on the server.	
AE Table	Name of table used to store alarm and event (AE) data.	HistoricalAlarmAndConditionOPCUASERVER_xx
HDA Table	Name of table used to store HDA data.	HistoricalDataOPCUASERVER_xx
Use Bulk Insert	Enables inserting multiple rows in a single operation for better performance with large datasets. May increase memory usage.	Checked

Table 88: OPC UA Server Broker - Historian Configuration Parameters

5.2.8.2. MQTT Broker

The MQTT Broker enables messaging between SIOTH components and external applications. Configure the parameters as shown below.



The screenshot shows the MQTT Broker Configuration View. It includes fields for Name (MQTTBROKER_Connector_0), Node (SIOTHMasterNode), Persistent Session (unchecked), Communication Timeout (900), Max Messages Queue Length per Client (1000000), TCP Endpoint (checked), TCP Settings (Port 1883, Security options: Use TLS/SSL and User Authentication, both unchecked), and Web Socket Endpoint (unchecked). There is an Advanced Configuration dropdown and Cancel/Save buttons at the bottom.

Figure 138: MQTT Broker Configuration View

Parameter	Description	Default Value
Name	The MQTT broker name. It is generated automatically.	
Node	The Node or machine where the MQTT Broker will be deployed.	SIOTHMasterNode

<i>Persistent Session</i>	When checked, it saves all client-relevant information on the broker. The session is identified by the clientId provided when the client connects.	Checked
<i>Communication Timeout (ms)</i>	Specifies the maximum duration a TCP/IP operation will be retried before it is considered failed.	900
<i>Max Messaging Queue Length per Client</i>	Maximum number of MQTT messages allowed per client.	1000000
<i>TCP Endpoint</i>	Enable or disable TCP mode.	checked
<i>TCP Endpoint : Port</i>	Port for MQTT Broker over TCP.	1883
<i>Web Socket Endpoint</i>	Enable or disable WebSocket mode.	Unchecked
<i>WebSocket Endpoint: Port</i>	Port for MQTT Broker over WebSocket.	9001
<i>WebSocket Endpoint: Path</i>	Path used to route MQTT over WebSocket traffic to the broker.	Path
<i>TLS/SSL</i>		
<i>Use TLS/SSL</i>	Enables TLS/SSL secure communication.	
<i>Port</i>	MQTT Broker port for TLS (TCP or WebSocket).	8883 (TCP) 443 (Over WS)

Protocol	Security protocol for MQTT communication. Options: None, TLS 1.1, SSL 1.2	None
CA Certificate Path	Path to the Certification Authority (CA) certificate.	
Server Certificate Path	Path to the broker's digital certificate file.	
Certificate Password	Password for the digital certificate.	
Client Certificate Required	Indicates whether client certificates are required for authentication.	
User Authentication		
Username & Password	Credentials required for clients to connect to the broker when User Authentication is enabled.	
Advanced Configuration		
Data Encryption	Secures transmitted data between cloud and system components.	Checked
Offline	Allows downloading the connector package for manual or offline deployment. Refer to the Offline Deployment section for more details about how to deploy a connector in offline mode.	unchecked
Service Configuration		
Service Configuration	Defines parameters for service setup and behavior.	Checked

Logon Type	Service logon type: <ul style="list-style-type: none"> Local System Specific Account 	Local System
Startup Type	Defines service start mode: <ul style="list-style-type: none"> Automatic Automatic (Delayed Start) Manual Disabled 	Automatic
Data Transfer Mode		
Data Request Port	Port used for data request operations.	
Client ID	Unique identifier for each server connecting to the master broker.	
Log Settings		
Auto Append	Automatically appends new entries to existing log files.	Checked
Log Level	Severity of log entries: <ul style="list-style-type: none"> Information Debug 	Information
Buffer Size (MB)	Memory allocated for temporary data storage before processing.	100
Maximum Files	Maximum number of retained log files.	10
Log File Max size (MB)	Maximum allowed log file size.	10

<i>Auto Save Timeout (s)</i>	Time before data is automatically saved.	5
---	--	---

Table 89: MQTT Broker Configuration Parameters

(!) Note

The SIOTH MQTT Broker automatically publishes tags originating from the connected source connectors.

5.2.8.3. Azure Event Hub

The **Azure Event Hub Connector** enables SIOTH to send real-time data streams to **Azure Event Hubs**, a fully managed, high-throughput data ingestion service in Microsoft Azure. It allows publishing event data collected from various sources into Azure Event Hubs for further processing, analytics, or storage within the Azure ecosystem.

The **Azure Event Hub Connector** requires the same core parameters to be configured during the **Identification** step. The Azure Event Hub Connector is available only as **Destination**.

Click **Next** to proceed to the **Configuration** page.

Identification

2 Configuration

Azure Event Hub Configuration

Device *

AzureEventHub_Device_1

Event Hub Name *

Protocol

AMQPS 1.0

Connection Options

Transport Type

TCP

Connection Idle Timeout (ms) *

20000

Buffer Size (Byte) *

8162

☐ Retry Options

Payload Configuration

Reset

Add

Added Configuration

Cancel

Back

Save

Figure 139: Azure Event Hub Connector as Destination – Configuration View

Parameter	Description	Default Value
Azure Event Hub Configuration		
Device	Defines to the Azure Event Hub device from which data will be read or to which data will be published.	
Event Hub Name	The specific Event Hub within the namespace where the connector will send messages. Each Event Hub has a unique name.	

Protocol	Communication protocol with Event Hub. Options: <ul style="list-style-type: none"> • AMQPS 1.0 • HTTPS • Kafka 	AMQPS 1.0
Connection Options: only for AMQPS 1.0		
Transport Type	Transport method for AMQPS 1.0 connections. Options: <ul style="list-style-type: none"> • TCP (standard) • Web Socket (web-based) 	TCP
Connection Idle Timeout (ms)	Duration to keep a connection open without traffic before closing it as idle.	20000
Buffer Size (Byte)	Buffer size used to temporarily store data for transmission.	8162
Use Proxy	Enables proxy communication for Web Socket connections. Options: <ul style="list-style-type: none"> • Use Proxy (requires proxy credentials and host/port). • Use Default Proxy (uses system proxy with credentials). 	Unchecked
Retry Options: only for AMQPS 1.0		
Retry Options	Enables retry behavior to manage connection or network failures.	Unchecked
Mode	Defines the retry delay calculation mode: <ul style="list-style-type: none"> • Exponential (increasing back-off). 	Exponential

	<ul style="list-style-type: none"> • Fixed (constant delay). 	
Delay (ms)	Base delay between retry attempts (used for exponential mode).	800
Maximum Delay (ms)	Maximum delay between retry attempts.	60000
Maximum Retries	Maximum retry attempts before the operation is marked as failed.	3
Timeout (ms)	Maximum time to wait for completion of a single attempt, including retries.	60000
Payload Configuration		
Source	The source connector from which Azure Event Hub subscribes.	
Message Properties		
Partition Key	Key to determine the partition where the event message will be sent, ensuring message order.	
Partition ID	Specific partition within the Event Hub for the message.	
Content Type	Format of the message payload.	Application/Json
Message ID	Unique identifier for each message.	
Correlation ID	Identifier used to correlate related messages or requests in a workflow.	
Additional Properties Configuration		
Key	Custom property name (unique identifier or label).	

<i>Value</i>	Value associated with the key. Can be string, number, or other type.	
<i>Data Filtering</i>	Enables filtering of received data (Download, Import, Export template).	Unchecked
<i>Fields Aliasing</i>	Allows renaming fields or attributes in the payload for clarity or compatibility.	Unchecked
<i>Data Aliasing</i>	Provides alternate names or references for data elements.	Unchecked
<i>Split Payload</i>	Splits large messages into smaller pieces for easier processing.	Unchecked
<i>JSON Formatter</i>	Ensures JSON payloads are properly structured. Allows variable configuration or manual JSON building.	Unchecked
<i>Added Configuration</i>		
<i>Added Configuration</i>	Allows users to view or verify the added configuration.	

Table 90: Azure Event Hub Connector as Destination – Configuration Parameters

5.2.9. Network Watcher

5.2.9.1. PING

The Ping Connector requires the same core parameters to be configured during the **Identification** step.

(!) Note

The Ping Connector is available **only as a Source**.

Click **Next** to proceed to the **Tag Configuration** page, where you can define parameters related to data mapping.

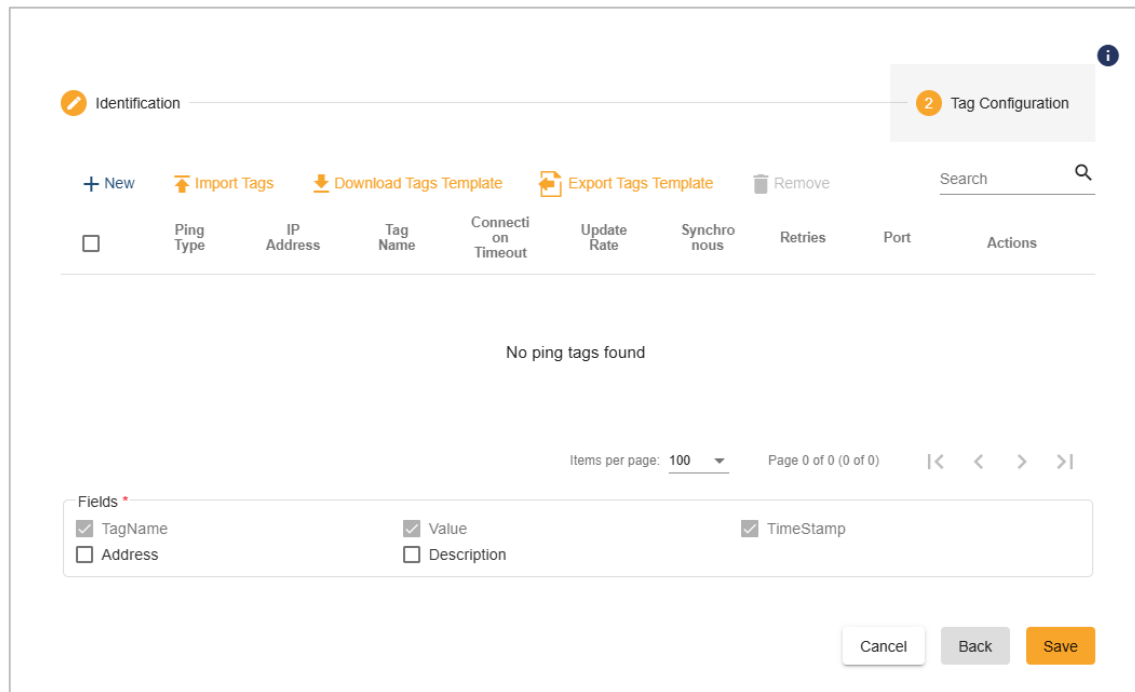


Figure 140: Ping Connector as Source - Tag Configuration View

Click **New** in the top menu to add a new tag. A configuration dialog opens, allowing you to define the parameters for the new tag.

New Tag

Tag Name *
Tag1|

Ping Type *
TCP

IP Address *
127.0.0.1

Update Rate(ms) *
30000

Connection Timeout(ms) *
2000

Retries *
3

Port *
4800

☒ Synchronous

Cancel Save

Figure 141: Ping Connector as Source - New Tag Configuration View

Parameter	Description	Default Value
Tag Name	Specifies the name of the tag.	
Ping Type	<p>Defines the protocol used for the ping request.</p> <p>Available options are:</p> <ul style="list-style-type: none"> • ICMP • TCP • UDP 	TCP
IP Address	Specifies the IP address or hostname of the target machine to be pinged.	127.0.0.1
Update Rate (ms)	Defines the frequency of ping requests expressed in milliseconds.	1000
Connection Timeout (ms)	Specifies the maximum time to wait for a response before the ping request times out.	2000
Retries	Specifies the number of retry attempts if no response is received within the timeout period.	3

Port	Specifies the network port used for the ping connection.	4800
Synchronous	Determines whether the ping request is executed synchronously. When enabled, the round-trip time (RTT) between the source and the target is measured.	Checked

Table 91: Ping Connector as Source - New Tag Configuration Parameters

You can use the **Download Tags Template**, **Export Tags Template**, and **Import Tags** options to manage and configure a list of tags associated using a CSV file format.

Save your configuration using the **Save** button.

5.2.9.2. Health

The Health Connector requires the same core parameters to be configured during the **Identification** step.

(!) Note

The Health Connector is available **only as a Source**.

Click **Next** to proceed to the **KPIs Configuration** page, where you can define parameters related to the counters that are needed to be retrieved.

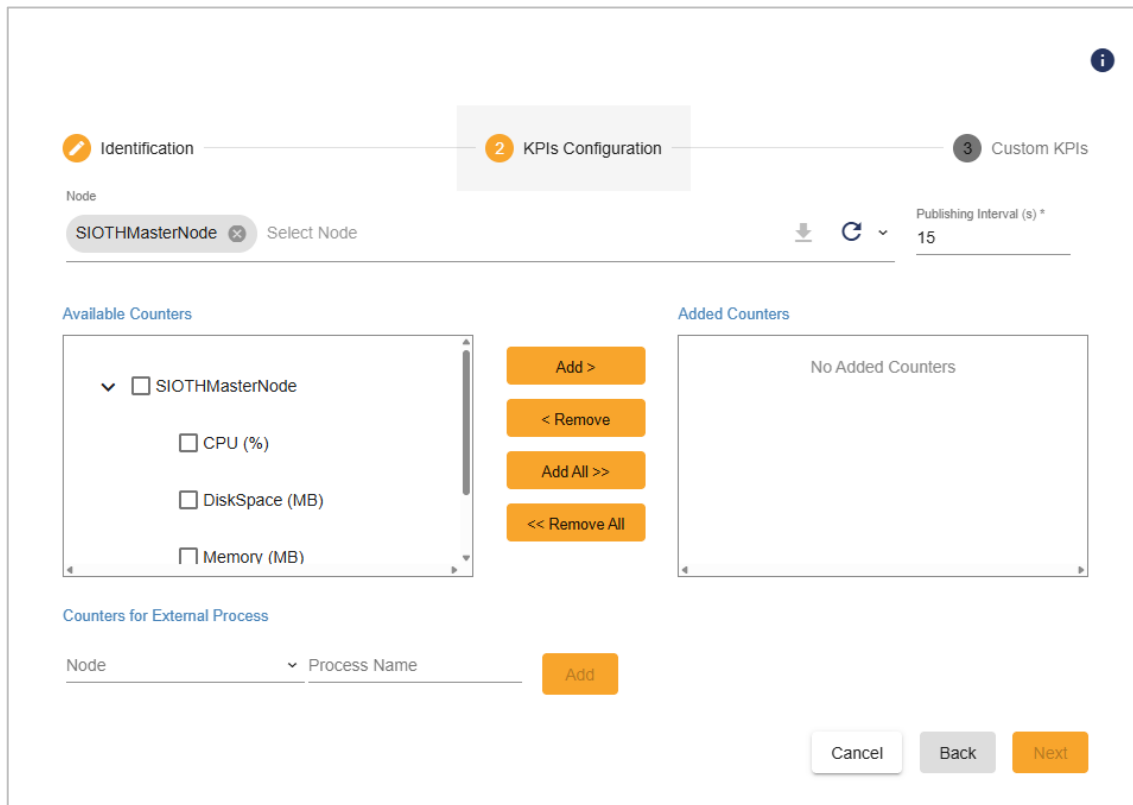


Figure 142: Health Connector as Source - KPIs Configuration View

Parameter	Description	Default Value
Node	Allows selection of the nodes from which system metrics will be retrieved.	SIOTHMasterNode
Publishing Interval (s)	Defines the frequency, in seconds, at which metrics are collected from the selected nodes.	15
Available Counters	Displays the selected nodes along with their available connectors and metrics. From this list, you can select metrics and add them to the Added Counters section.	

Added Counters	Lists the counters configured for retrieval. Counters can be removed by selecting one or more entries and clicking Remove or Remove All .	
Counters for External Process	Allows definition and addition of custom metrics related to external processes. Once added, these processes appear in the Available Counters list.	

Table 92: Health Connector as Source - KPIs Configuration Parameters

Click **Next** to proceed to the **Custom KPIs** page, where you can define and add custom metrics.

Custom KPIs are user-defined expressions derived from available system metrics and previously defined custom KPIs. They can be combined using basic arithmetic operators to create higher-level indicators that more accurately reflect system performance.

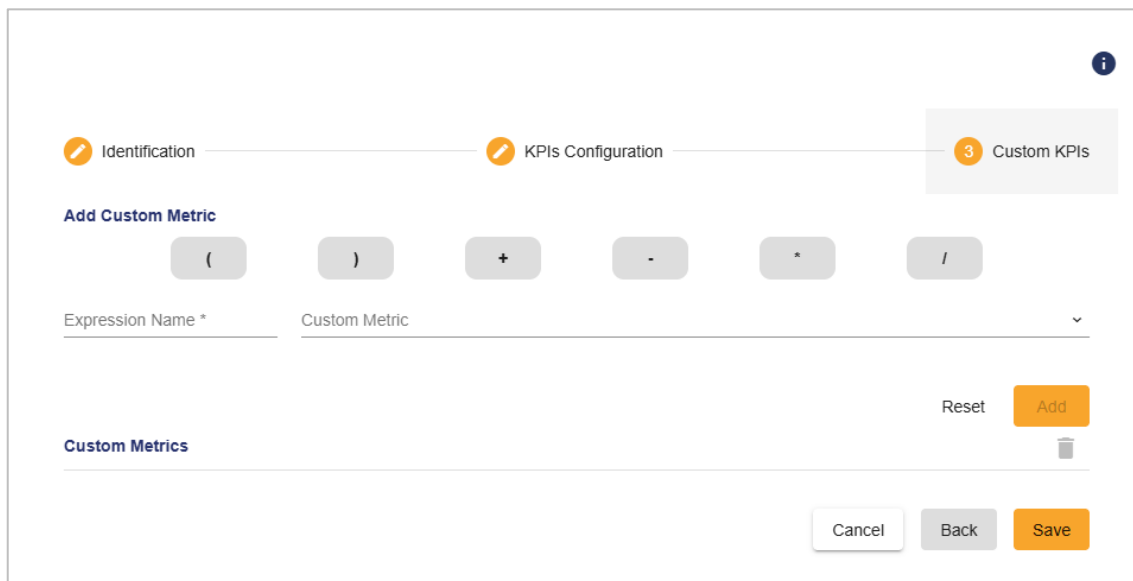


Figure 143: Health Connector as Source - Custom KPIs Configuration View

5.2.9.3. SNMP

Click **Next** to proceed to the **Tag Configuration** page.

The available configuration options vary depending on whether the connector is configured as a **Source** or a **Destination**.

(!) Note:

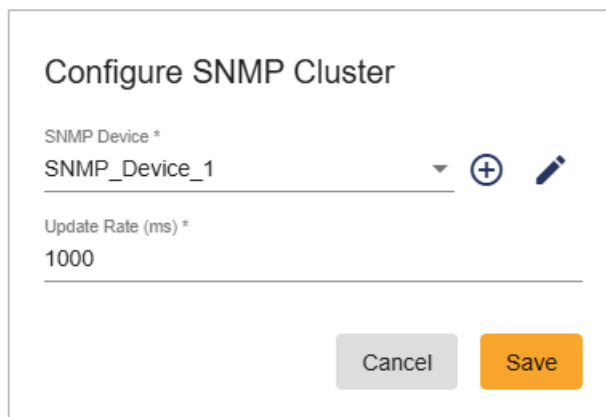
For the SNMP connector to operate properly, the Simple Network Management Protocol (SNMP) feature must be installed and enabled on the target system. Ensure that the following requirements are met:

- The SNMP service is installed and running.
- The SNMP service startup type is set to **Automatic**, allowing it to start with the system.
- A valid SNMP community name (e.g., public) is configured, and the host is authorized to send and receive SNMP packets.

If SNMP is not installed or the service is not running, the connector will not be able to retrieve system metrics.

SNMP Connector as Source:

Click **New Subscription** from the left section in the **Tag Configuration** page to add a subscription to an SNMP connector configured as **Source**. A new window will open where you can configure the subscription settings.



Configure SNMP Cluster

SNMP Device *

SNMP_Device_1

Update Rate (ms) *

1000

Cancel Save

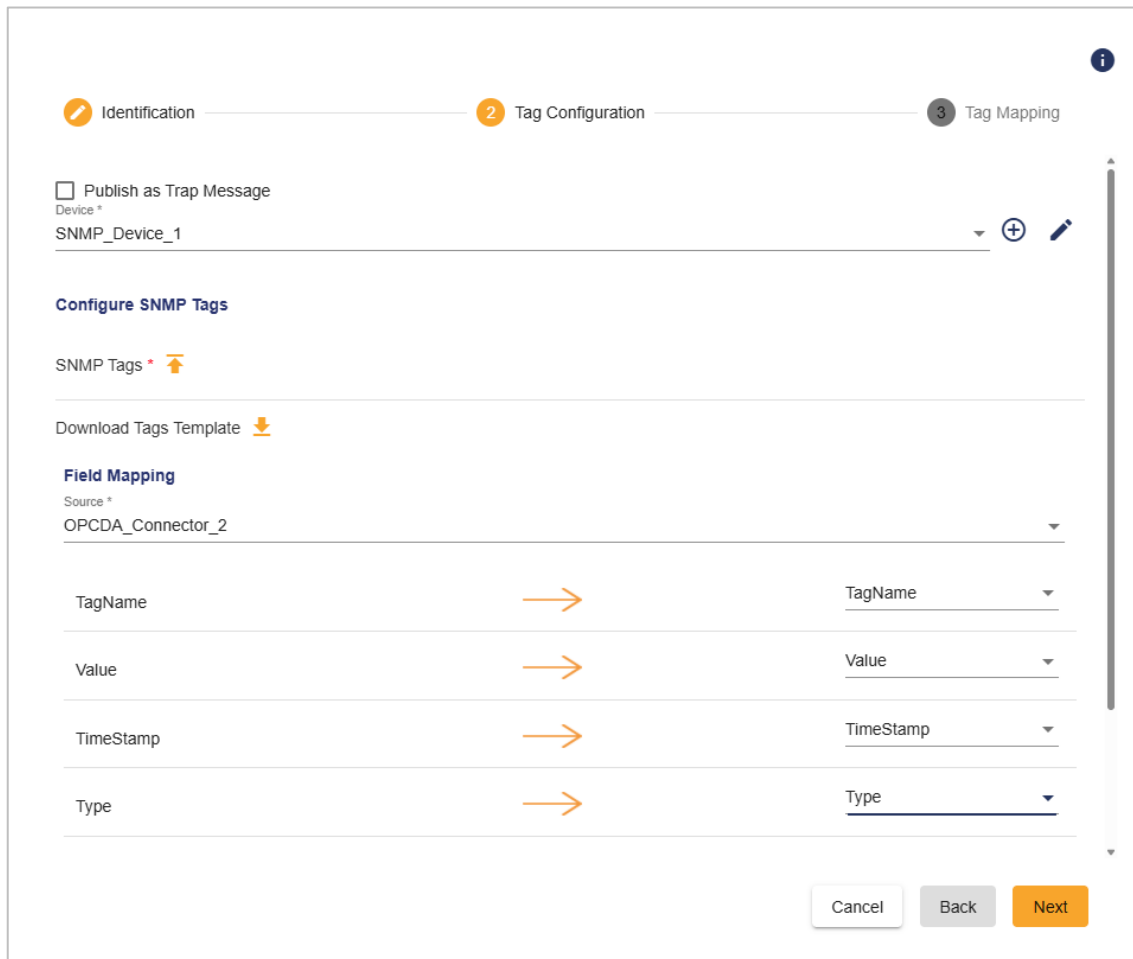
Figure 144: SNMP Connector as Source - Subscription Configuration View

Parameter	Description	Default Value
<i>SNMP device</i>	Specifies the SNMP device from which data will be retrieved.	
<i>Update Rate (ms)</i>	Defines the frequency of data read requests. The value is expressed in milliseconds.	1000

Table 93: SNMP Connector as Source - Subscription Configuration Parameters


SNMP Connector as Destination:

Configure the required parameters on the **Tag Configuration** page to define the mapping between the source connector and the destination SNMP device. This mapping ensures that data from the source connector is correctly transmitted to the target device.



The screenshot shows the 'Tag Configuration' step in a three-step process (Identification, Tag Configuration, Tag Mapping). It includes a 'Publish as Trap Message' checkbox, a 'Device' dropdown menu set to 'SNMP_Device_1', and a 'Configure SNMP Tags' section with a 'Download Tags Template' button. Below this is a 'Field Mapping' table with four rows: 'TagName', 'Value', 'TimeStamp', and 'Type'. Each row has an orange arrow pointing from a source field to a target field. The 'Source' dropdown is set to 'OPCDA_Connector_2'. At the bottom are 'Cancel', 'Back', and 'Next' buttons.

Figure 145: SNMP Connector as Destination - Tag Configuration View

Parameter	Description	Default Value
<i>Publish as Trap Message</i>	When enabled, data is sent as an SNMP Trap message instead of a standard SNMP Set operation.	Unchecked
<i>Device</i>	Specifies the SNMP device to which data will be written.	
<i>Configure SNMP Tags</i>	Allows import a list of SNMP tags from a CSV file. A template file is available via the Download Tags Template icon  .	

Field Mapping	Enables mapping of source fields to SNMP tags, defining how incoming data is written to the destination device.	
----------------------	---	--

Table 94: SNMP Connector as Destination - Tag Configuration Parameters

Click **Confirm Mapping** to validate the configuration and click Next to proceed to the Tag Mapping page.

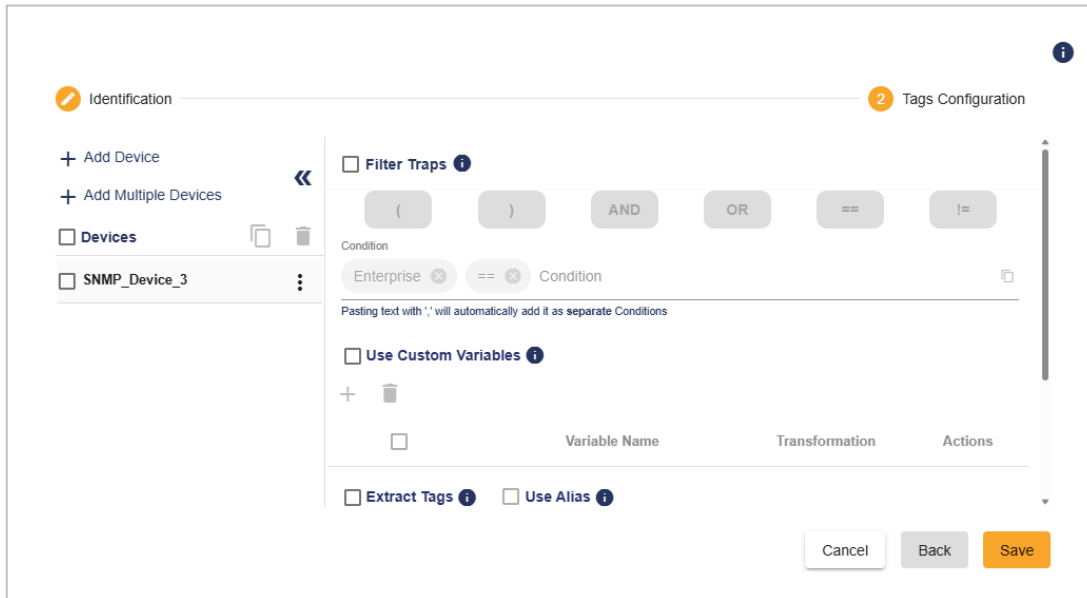
5.2.9.4. SNMP Trap

The SNMP Trap Connector requires the same core parameters to be configured during the **Identification** step.

(!) Note

The SNMP Trap Connector is available **only as a Source**.

Click **Next** to proceed to the **Tags Configuration** page, where you can define the devices and parameters related to SNMP traps.



The screenshot shows the 'Tags Configuration' view for the SNMP Trap Connector. On the left, there's a sidebar with 'Identification' and 'Tags Configuration' tabs. The 'Tags Configuration' tab is active, showing a configuration area with several checkboxes: 'Filter Traps', 'Use Custom Variables', 'Extract Tags', and 'Use Alias'. A condition builder is visible with 'Enterprise' and 'Condition' selected. Below this, there's a table for custom variables with columns for 'Variable Name', 'Transformation', and 'Actions'. At the bottom, there are 'Cancel', 'Back', and 'Save' buttons.

Figure 146: SNMP Trap Connector as Source - Tags Configuration View

Parameter	Description	Default Value
<i>Filter Traps</i>	When enabled, the connector processes only SNMP trap messages that match the defined filter conditions.	Unchecked
<i>Using Custom Variables</i>	Allows the use of custom variables to extract and transform fields from received trap messages for improved data interpretation.	Unchecked
<i>Extract Tags</i>	Converts payload data, variables, and device information contained in SNMP trap messages into structured tags.	Unchecked
<i>Use Alias</i>	When monitoring multiple devices, aliases act as common tag identifiers shared across all configured devices.	Unchecked

Table 95: SNMP Trap Connector as Source - Tags Configuration Parameters

The **Tag Configuration** page allows you to add SNMP devices directly, without navigating back to the **Devices** section. Click **Add Device** to add a new SNMP device. Refer to the **Add New Device** section for detailed information about the SNMP-related parameters.

You can also use the **Add Multiple Devices** option to discover and add devices by scanning a specified range of IP addresses.

Configure Multiple SNMP Devices

Scan by *	Start IP Address *	End IP Address *
Range	192.168.1.1	192.168.1.254
SNMP Version *	Read Community *	Write Community *
Version 2	public	public
SNMP Timeout(ms) *	Trap Port *	Request Port *
6000	162	161

☒ Read per Batch *
 ☒ Deactivate Tags on NoSuchObject/Instance or NoSuchName Errors *

SNMP Authentication

User

Authentication Protocol
 Authentication Password

Privacy Protocol
 Privacy Password

Cancel
 Scan


Table 96: SNMP Trap Connector as Source - Add Multiple Devices Configuration View

5.2.10. Routers

Routers in SIOTH are responsible for directing data between different connectors, nodes, or systems. They provide flexible data routing mechanisms, enabling efficient, reliable, and secure data exchange across distributed architectures.

5.2.10.1. Bridge Server

The **Bridge Server** acts as a centralized communication hub that receives data from source connectors and forwards it to connected clients or routers. It ensures secure, reliable, and high-performance data distribution within multi-node deployments.



Name *

Bridge Server_Connector_0

Node *

SIOTHMasterNode

IP Address *

127.0.0.1

Destination IP Address *

127.0.0.1

Mode *

Reverse Connect

Bridge Client IP Address *

127.0.0.1

Bridge Client Port *

5000

Cancel Save

Figure 147: Bridge Server Connector Configuration View

Parameter	Description	Default Value
Name	Name of the Bridge Server connector instance.	Bridge Server_Connector_X X
Node	Specifies the node or machine where the Bridge Server is deployed.	SIOTHMasterNode
IP Address	IP address of the Bridge Server.	127.0.0.1
Destination IP Address	Target IP address for forwarding data.	127.0.0.1
Mode	Defines the connection mode. Options: <ul style="list-style-type: none"> Reverse Connect: The Bridge Server initiates the TCP connection to the Bridge Client 	Reverse Connect

	<ul style="list-style-type: none"> Normal Connect: The Bridge Client initiates the connection to the Bridge Server 	
Bridge Client IP Address	IP address of the Bridge Client connected to this server.	127.0.0.1
Bridge Client Port	Port used for communication with the Bridge Client.	5000
Connect Retry Interval (ms)	Interval between automatic connection retry attempts.	5000
TCP Connection Control		
Idle Timeout (ms)	Maximum idle time before a TCP connection is closed.	3600000
Keep Alive Timeout (ms)	Interval to send TCP keep-alive messages.	10000
Buffer Size (Kb)	Size of the buffer used for TCP data transfer.	4096
Security		
Subject Name	Certificate subject name for secure communication.	CN=server.siothbridge.io.com
Store Name	Certificate store name.	Root
Store Location	Certificate store location.	LocalMachine
Secure Dedicated Channel Communication	Enables secure communication using dedicated channels.	Checked

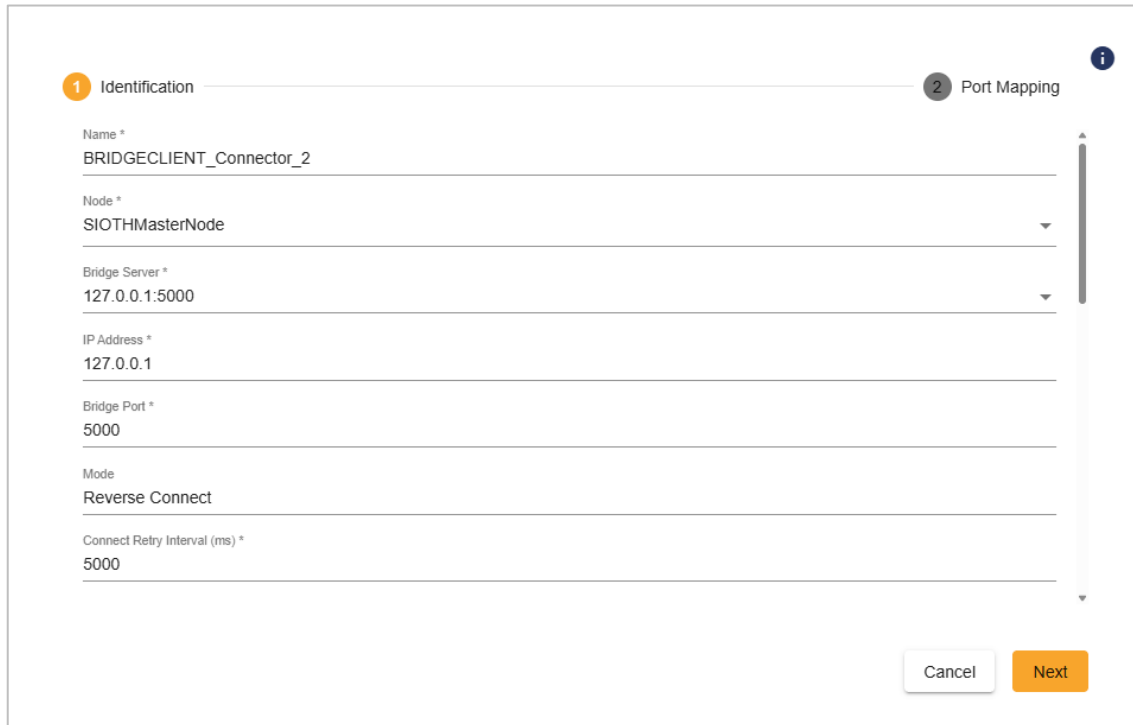
Advanced Configuration		
Offline	Allows downloading the connector package for manual or offline deployment.	Unchecked
Service Configuration		
Service Configuration	Enables configuration and adjustment of service parameters.	Checked
Logon Type	Specifies the account used to run the service: <ul style="list-style-type: none"> Local System Specific Account 	Local System
Username	Username of the specific account when Logon Type is set to <i>Specific Account</i> .	
Password	Password of the specific account when Logon Type is set to <i>Specific Account</i> .	
Startup Type	Defines how and when the service starts: <ul style="list-style-type: none"> Automatic Automatic (Delayed Start) Manual Disabled 	Automatic
Log Settings		
Auto Append	Automatically appends new log entries to existing log files.	Checked
Log Level	Defines the log severity level: <ul style="list-style-type: none"> Information 	Information

	<ul style="list-style-type: none"> • Debug 	
Buffer Size (MB)	Amount of memory allocated for temporary data buffering.	100
Maximum Files	Maximum number of log files retained.	10
Log File Max size (MB)	Maximum size allowed per log file.	10
Auto Save Timeout (s)	Time interval (in seconds) before data is automatically saved.	5

Table 97: Bridge Server Connector Configuration Parameters

5.2.10.2. Bridge Client

The **Bridge Client** connects to a Bridge Server to receive routed data, enabling distributed nodes or systems to access real-time information from other parts of the network.



The image shows a configuration window titled "Bridge Client Connector Identification Configuration View". It has two tabs: "1 Identification" (active) and "2 Port Mapping". The "Identification" tab contains several input fields with labels and asterisks indicating required fields: "Name *" with value "BRIDGECLIENT_Connector_2", "Node *" with value "SIOTHMasterNode", "Bridge Server *" with value "127.0.0.1:5000", "IP Address *" with value "127.0.0.1", "Bridge Port *" with value "5000", "Mode" with value "Reverse Connect", and "Connect Retry Interval (ms) *" with value "5000". There are "Cancel" and "Next" buttons at the bottom right. A vertical scrollbar is on the right side of the form.

Figure 148: Bridge Client Connector Identification Configuration View

Parameter	Description	Default Value
Name	Name of the Bridge Client connector instance.	BRIDGECLIENT_Connector_XX
Node	Specifies the node or machine where the Bridge Client is deployed.	SIOTHMasterNode
Bridge Server	IP address and port of the Bridge Server to connect to.	127.0.0.1:5000
IP Address	IP address of the Bridge Client.	127.0.0.1
Bridge Port	Port of the Bridge Client.	5000
Mode	Defines the connection mode.	Reverse Connect
Connect Retry Interval (ms)	Interval between automatic connection retry attempts.	5000
TCP Connection Control		

Idle Timeout (ms)	Maximum idle time before a TCP connection is closed.	3600000
Keep Alive Timeout (ms)	Interval to send TCP keep-alive messages.	10000
Keep Alive Interval (ms)	Interval between individual keep-alive probes sent to the server.	5000
Buffer Size (Kb)	Size of the buffer used for TCP data transfer.	4096
Security		
Subject Name	Certificate subject name for secure communication.	CN=client.siothbridge.io.com
Store Name	Certificate store name.	Root
Store Location	Certificate store location.	LocalMachine
Server Certificate Dns Name	DNS name of the connected Bridge Server certificate.	server.siothbridge.io.com
Secure Dedicated Channel Communication	Enables secure communication using dedicated channels.	Checked
Advanced Configuration		
Offline	Allows downloading the connector package for manual or offline deployment.	Unchecked
Service Configuration		

Service Configuration	Enables configuration and adjustment of service parameters.	Checked
Logon Type	Specifies the account used to run the service: <ul style="list-style-type: none"> Local System Specific Account 	Local System
Username	Username of the specific account when Logon Type is set to <i>Specific Account</i> .	
Password	Password of the specific account when Logon Type is set to <i>Specific Account</i> .	
Startup Type	Defines how and when the service starts: <ul style="list-style-type: none"> Automatic Automatic (Delayed Start) Manual Disabled 	Automatic
Log Settings		
Auto Append	Automatically appends new log entries to existing log files.	Checked
Log Level	Defines the log severity level: <ul style="list-style-type: none"> Information Debug 	Information
Buffer Size (MB)	Amount of memory allocated for temporary data buffering.	100
Maximum Files	Maximum number of log files retained.	10

Log File Max size (MB)	Maximum size allowed per log file.	10
Auto Save Timeout (s)	Time interval (in seconds) before data is automatically saved.	5

Table 98: Bridge Client Connector Identification Configuration Parameters

Click **Next** to proceed to the **Port Mapping** page. This page allows defining network rules to control which ports and IP addresses the Bridge Client uses to route data.

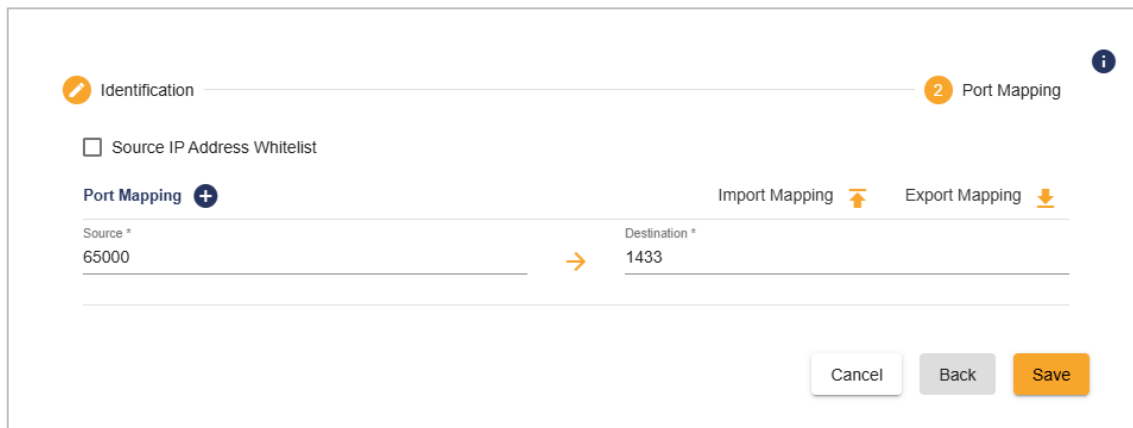


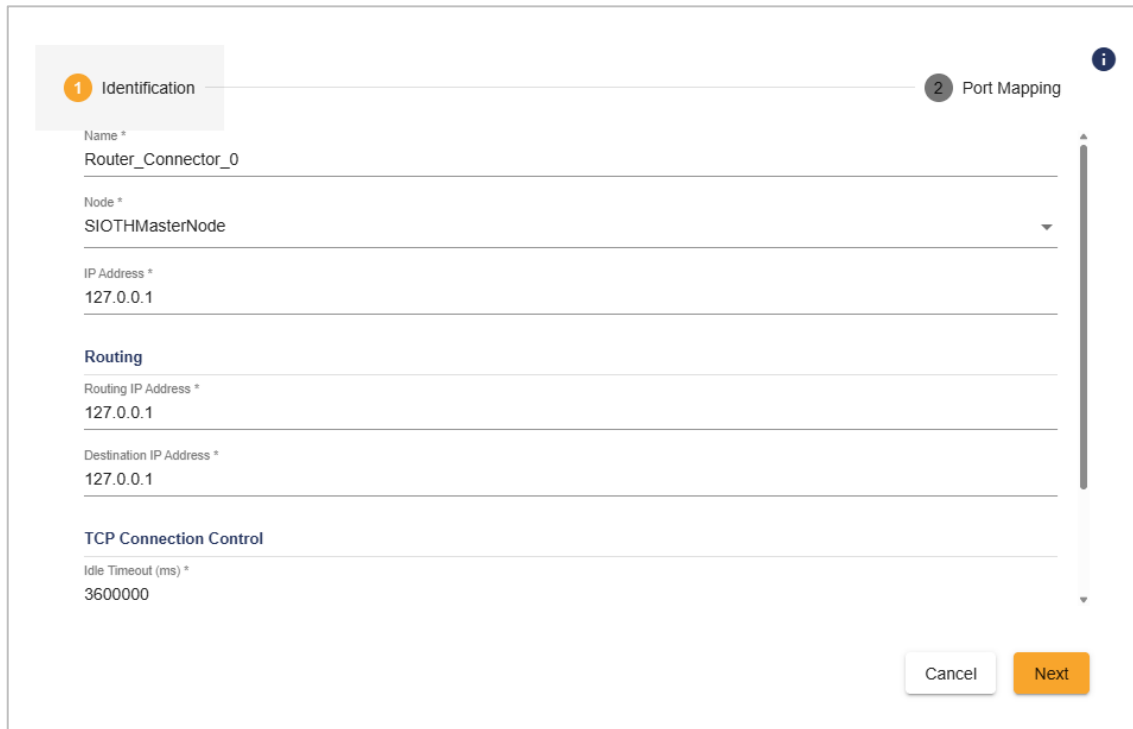
Figure 149: Bridge Client Connector Port Mapping Configuration View

Parameter	Description	Default Value
Source IP Address Whitelist	When enabled, specifies a list of allowed IP addresses that can connect to the Bridge Client.	Unchecked
Port Mapping	Defines the mapping between source and destination ports. Use the Import Mapping and Export Mapping features to configure multiple mappings efficiently.	

Table 99: Bridge Client Connector Port Mapping Configuration Parameters

5.2.10.3.Router

Router Blocks define the logic for routing data between connectors, Bridge Servers, and Bridge Clients. They enable conditional, filtered, or transformed data flows based on configurable rules.



The image shows a configuration window for a Router Connector. It has two tabs: '1 Identification' (active) and '2 Port Mapping'. The 'Identification' tab contains the following fields:

- Name ***: Router_Connector_0
- Node ***: SIOTHMasterNode (selected from a dropdown menu)
- IP Address ***: 127.0.0.1
- Routing** section:
 - Routing IP Address ***: 127.0.0.1
 - Destination IP Address ***: 127.0.0.1
- TCP Connection Control** section:
 - Idle Timeout (ms) ***: 3600000

At the bottom right, there are 'Cancel' and 'Next' buttons. An information icon (i) is located in the top right corner.

Figure 150: Router Connector Identification Configuration View

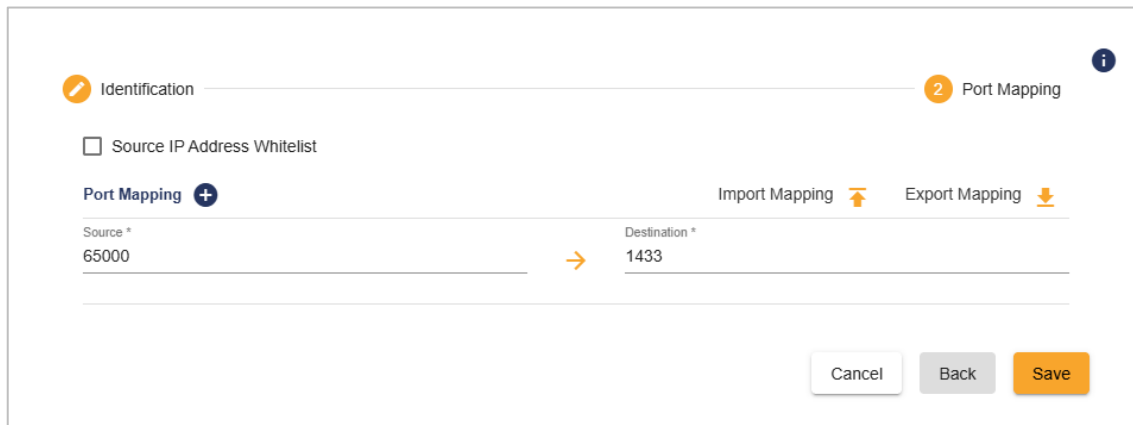
Parameter	Description	Default Value
<i>Name</i>	Name of the Router connector instance.	Router_Connector_X X
<i>Node</i>	Specifies the node or machine where the Router is deployed.	SIOTHMasterNode
<i>IP Address</i>	IP address of the Router.	127.0.0.1
<i>Routing</i>		

<i>Routing IP Address</i>	IP address used for routing data.	127.0.0.1
<i>Destination IP Address</i>	IP address of the destination system or connector.	127.0.0.1
<i>TCP Connection Control</i>		
<i>Idle Timeout (ms)</i>	Maximum idle time before a TCP connection is closed.	3600000
<i>Buffer Size (Kb)</i>	Size of the buffer used for TCP data transfer.	4096
<i>Advanced Configuration</i>		
<i>Offline</i>	Allows downloading the connector package for manual or offline deployment.	Unchecked
<i>Service Configuration</i>		
<i>Service Configuration</i>	Enables configuration and adjustment of service parameters.	Checked
<i>Logon Type</i>	Specifies the account used to run the service: <ul style="list-style-type: none"> Local System Specific Account 	Local System
<i>Username</i>	Username of the specific account when Logon Type is set to <i>Specific Account</i> .	
<i>Password</i>	Password of the specific account when Logon Type is set to <i>Specific Account</i> .	
<i>Startup Type</i>	Defines how and when the service starts:	Automatic

	<ul style="list-style-type: none"> • Automatic • Automatic (Delayed Start) • Manual • Disabled 	
<i>Log Settings</i>		
<i>Auto Append</i>	Automatically appends new log entries to existing log files.	Checked
<i>Log Level</i>	Defines the log severity level: <ul style="list-style-type: none"> • Information • Debug 	Information
<i>Buffer Size (MB)</i>	Amount of memory allocated for temporary data buffering.	100
<i>Maximum Files</i>	Maximum number of log files retained.	10
<i>Log File Max size (MB)</i>	Maximum size allowed per log file.	10
<i>Auto Save Timeout (s)</i>	Time interval (in seconds) before data is automatically saved.	5

Table 100: Router Connector Identification Configuration Parameters

Click **Next** to proceed to the **Port Mapping** page. This page allows defining network rules to control which ports and IP addresses the connector uses to route data.



The screenshot shows the 'Port Mapping' configuration view. At the top, there are two tabs: 'Identification' (active) and 'Port Mapping' (labeled with a '2'). Below the tabs, there is a checkbox for 'Source IP Address Whitelist'. Under the 'Port Mapping' tab, there is a 'Port Mapping' section with a plus icon. Below this, there are two input fields: 'Source *' with the value '65000' and 'Destination *' with the value '1433'. An orange arrow points from the source to the destination. To the right of these fields are two buttons: 'Import Mapping' with an upward arrow and 'Export Mapping' with a downward arrow. At the bottom right, there are three buttons: 'Cancel', 'Back', and 'Save'.

Figure 151: Router Connector Port Mapping Configuration View

Parameter	Description	Default Value
Source IP Address Whitelist	When enabled, specifies a list of allowed IP addresses that can connect to the Router.	Unchecked
Port Mapping	Defines the mapping between source and destination ports. Use the Import Mapping and Export Mapping features to configure multiple mappings efficiently.	

Table 101: Router Connector Port Mapping Configuration Parameters

For additional information on this guide, questions or problems to report, please contact:

Offices

- Americas: +1 713 609 9208
- Europe-Africa-Middle East: +216 71 195 360

Email

- Support Services: customerservice@integrationobjects.com
- Sales: sales@integrationobjects.com

To find out how you can benefit from other Integration Objects' products and services, please visit our website:

Online

- www.integrationobjects.com