optimize!
softing

User Manual

dataFEED OPC Suite



Version: V5.38.0

© Softing Industrial Automation GmbH

Disclaimer of liability

The information contained in these instructions corresponds to the technical status at the time of printing of it and is passed on with the best of our knowledge. The information in these instructions is in no event a basis for warranty claims or contractual agreements concerning the described products, and may especially not be deemed as warranty concerning the quality and durability pursuant to Sec. 443 German Civil Code. We reserve the right to make any alterations or improvements to these instructions without prior notice. The actual design of products may deviate from the information contained in the instructions if technical alterations and product improvements so require.

It may not, in part or in its entirety, be reproduced, copied, or transferred into electronic media.

Softing Industrial Automation GmbH

Richard-Reitzner-Allee 6
85540 Haar / Germany
http://industrial.softing.com

+ 49 89 4 56 56-0
+ 49 89 4 56 56-488
info.automation@softing.com
support.automation@softing.com

The latest version of this manual is available in the Softing download area at:
http://industrial.softing.com/en/downloads.html

Table of Contents

Chapter 1 Welcome to Softing dataFEED OPC Suite 9

Chapter 2 dataFEED OPC Suite Quickstart 12

2.1 OPC Classic Tunnel 13
2.2 OPC UA Server 14
2.3 OPC Bridge 15
2.4 OPC Optimizer 16
2.5 OPC Concentrator 17
2.6 OPC Filter 18
2.7 OPC Client2Client 19
2.8 OPC Server2Server 20
2.9 OPC2File 21
2.10 OPC2Database 22
2.11 OPC Classic Store And Forward 23

Chapter 3 dataFEED OPC Suite Introduction 25

3.1 OPC Classic Tunnel 26
3.2 OPC UA 28
3.3 OPC Bridge 29
3.4 OPC Optimizer 30
3.5 OPC Concentrator 32
3.6 OPC Filter 33
3.7 OPC Client2Client 35
3.8 OPC Server2Server 38
3.9 OPC2File 40
3.10 OPC2Database 42
3.11 OPC Classic Store And Forward 44

Chapter 4 Installation 47

4.1 Operating Systems 47
4.2 Installation 48
4.3 Uninstallation 49
4.4 DCOM Configuration 49
4.4.1 System Wide Security Parameters 51
4.4.2 Application Specific Security Parameters 57
4.4.3 Activating Guest Access 60
4.4.4 Configuring the Windows Firewall 61

Chapter 5 Licensing 65

5.1 License Model 65
5.2 License Manager 69
5.3 Demonstration Mode 71

Chapter 6 Operation 73

6.1 Start Menu ... 73
6.2 Tray Application ... 75
6.3 Web Pages ... 81
6.3.1 Information ... 83
6.3.2 Diagnostics ... 84
6.3.2.1 Process Values ... 85
6.3.2.2 Clients ... 88
6.3.2.3 Configuration ... 89
6.3.2.4 Servers ... 90
6.3.2.5 Cyclic Requests ... 92
6.3.2.6 Device Connections ... 94
6.3.2.7 OPC UA Server Information ... 98
6.3.3 Configuration ... 99
6.3.3.1 Trace ... 100
6.3.3.2 Password ... 101
6.4 Building Own Web Pages ... 102
6.5 Demonstration Mode ... 103
6.6 Importing Multiprotocol OPC Server Configurations ... 103
6.6.1 Import - Configuration File Selection ... 104
6.6.2 Import - PLC Connections Selection ... 105
6.6.3 Import - PLC Symbols Selection ... 107
6.6.4 Import - Results ... 108
6.7 Identity Settings ... 110
6.7.1 dataFEED OPC Suite V5.38 ... 110
6.7.2 Changes Between V2.x/V3x and V4.x ... 112
6.8 System Items ... 114
6.9 OPC Tunnel Discovery Service ... 123
6.10 OPC UA Local Discovery Service ... 123

Chapter 7 Configuration ... 124

7.1 Configuration Concept ... 125
7.1.1 Data Sources ... 127
7.1.2 Data Processing ... 129
7.1.3 Data Destinations ... 130
7.2 dataFEED OPC Suite Configurator ... 131
7.2.1 General ... 132
7.2.2 Configuration Handling ... 135
7.2.3 Backup & Restore ... 140
7.2.4 Import Multiprotocol OPC Server Configuration ... 144
7.2.5 Local Application Access ... 144
7.2.6 Tree Handling ... 146
7.2.6.1 General ... 146
7.2.6.2 Local Namespace ... 149
7.2.6.3 Tree Node States and Icons ... 153
7.2.6.4 Online Mode ... 156
7.2.6.5 Offline Mode ... 157
7.2.6.6 Checked States ... 157
7.3 dataFEED OPC Suite Configuration ... 159
7.3.1 Data Sources ... 161

© Softing Industrial Automation GmbH

7.3.1.1 Siemens Devices ... 163
Configuration ... 164
Siemens Device Wizard ... 168
7.3.1.2 Siemens SINUMERIK CNC Devices ... 192
Configuration ... 193
SINUMERIK Device Wizard ... 202
7.3.1.3 Rockwell Devices ... 227
Configuration ... 227
Rockwell Device Wizard ... 232
7.3.1.4 Modbus Devices ... 243
Configuration ... 243
Modbus Device Wizard ... 248
7.3.1.5 B&R Devices ... 262
Configuration ... 263
B&R Device Wizard ... 266
7.3.1.6 Mitsubishi Devices ... 277
Configuration ... 277
Mitsubishi Device Wizard ... 282
7.3.1.7 OPC UA Server ... 291
Configuration ... 292
Server Connection Test ... 300
Advanced Settings ... 302
Global Settings ... 317
UA Server Connection Wizard ... 321
Appendix ... 341
7.3.1.8 OPC Server ... 342
Configuration ... 343
Advanced Settings ... 347
Connection test ... 350
Filter Item Properties ... 351
Filter File ... 356
OPC Server Wizard ... 358
Import Wizard ... 370
Export Wizard ... 374
7.3.1.9 OPC Classic Tunnel Server ... 377
Configuration ... 378
Advanced Settings ... 381
Connection test ... 384
Filter Item Properties ... 385
Filter File ... 390
Tunnel Server Wizard ... 392
Import Wizard ... 403
Export Wizard ... 407
7.3.1.10 Database Read ... 410
Configuration ... 411
Database Wizard ... 420
Database Action Wizard ... 428
Configuration Files ... 441
Limitations ... 452
7.3.1.11 MQTT Broker ... 453
Configuration ... 454
MQTT Broker Connection Wizard ... 457
Import Wizard ... 472

Export Wizard ... 475
Define JSON and XML payload format ... 477
Connect to Azure IoT Hub via SSL and WSS ... 484
Connect to AWS IoT Device via SSL and WSS ... 496
7.3.1.12 MQTT Publisher ... 506
Configuration ... 507
MQTT Publisher Connection Wizard ... 516
7.3.1.13 File Read ... 524
Configuration ... 525
File Read Action Wizard ... 528
7.3.1.14 Local Items ... 561
Configuration ... 562
Local Items Root ... 566
Advanced Settings ... 571
Local Items File ... 573
Import Wizard ... 574
Export Wizard ... 578
7.3.1.15 Appendix ... 580
S7 Symbol File Import Wizard ... 581
Import Wizard ... 590
Export Wizard ... 594
Global Settings Wizard ... 597
Item Syntax ... 602
Siemens S7-2 Advanced Settings ... 631
PLC Test Connection ... 635
Plc Connection Item Properties ... 636
TSAP ... 639
TSAP for S7 ... 640
IP Address ... 642
Suffixes ... 643
7.3.2 Data Processing ... 647
7.3.2.1 OPC Classic Store And Forward ... 648
Configuration ... 650
Store and Forward Item Settings ... 655
Store and Forward Wizard ... 656
File Format ... 662
Import Wizard ... 674
Export Wizard ... 682
7.3.2.2 Exchange ... 684
Configuration ... 686
Exchange Action Properties ... 693
Advance Exchange Action Editor ... 695
File Format ... 697
Import Wizard ... 704
Export Wizard ... 712
7.3.2.3 Preprocessing Engine ... 714
Configuration ... 716
Preprocessing Action Wizard ... 721
Data Values Placeholder Wizard ... 766
Data Values Placeholder ... 770
Configuration Files ... 774
7.3.3 Data Destinations ... 785
7.3.3.1 OPC UA Client ... 786

© Softing Industrial Automation GmbH

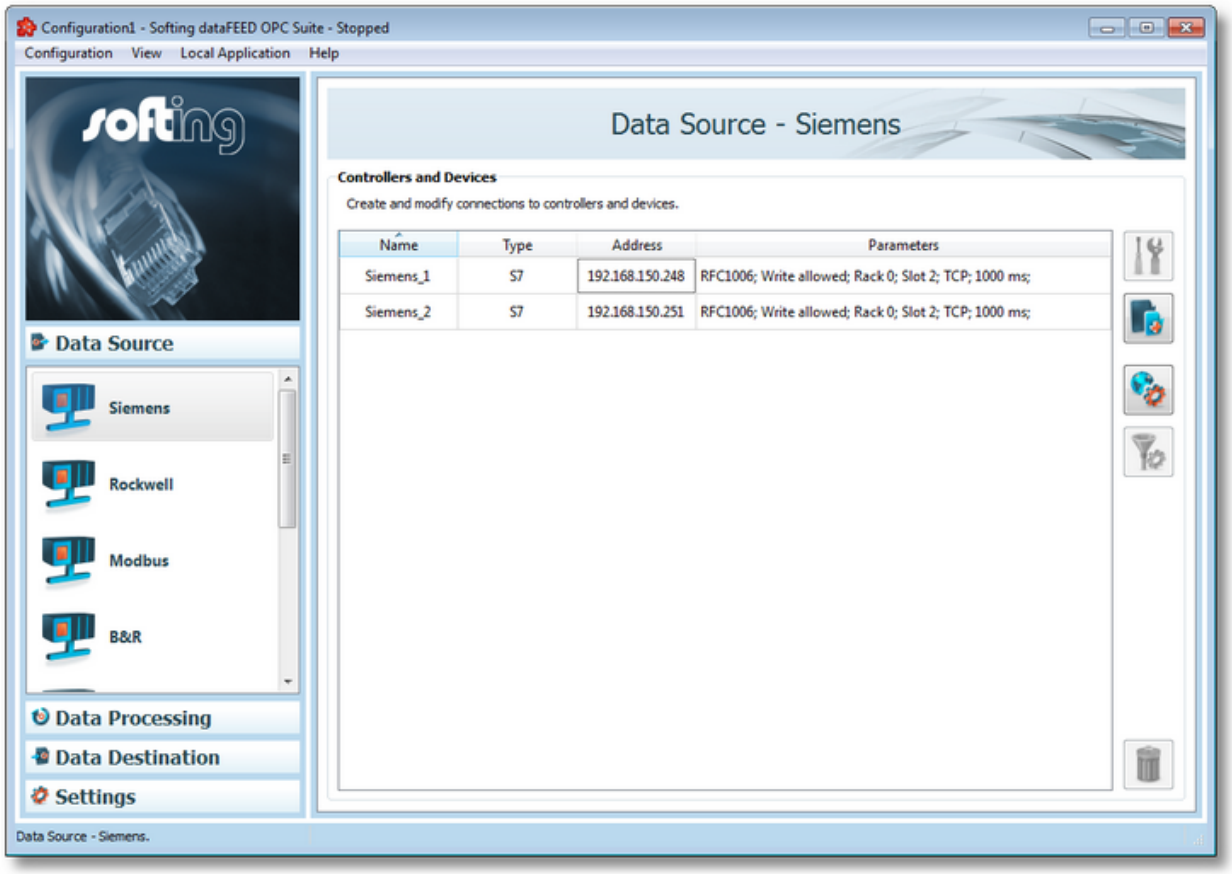
Configuration ... 788
7.3.3.2 OPC Client ... 867
Configuration ... 868
7.3.3.3 OPC Classic Tunnel Client ... 870
Configuration ... 871
7.3.3.4 MQTT Broker ... 876
Configuration ... 877
MQTT Connection Wizard ... 881
Connect to Azure IoT Hub via SSL and WSS ... 907
Connect to AWS IoT Device via SSL ... 921
7.3.3.5 Database ... 933
Configuration ... 935
Database Properties ... 943
Database Wizard ... 945
Database Action Wizard ... 952
Table Column Wizard ... 976
Data Value Placeholder Wizard ... 984
Data Values Placeholder ... 990
Configuration Files ... 996
Multiple Configuration Files ... 1011
7.3.3.6 REST Server ... 1014
Configuration ... 1015
Connection Wizard ... 1022
Action Wizard ... 1032
Data Value Placeholder Wizard ... 1053
Bosch PPM Example ... 1059
Parameter Properties ... 1064
Data Values Placeholder ... 1065
Configuration Files ... 1068
7.3.3.7 NoSQL Database ... 1080
Configuration ... 1081
Connection Wizard ... 1088
Action Wizard ... 1099
Data Value Placeholder Wizard ... 1122
Supported Databases ... 1128
Parameter Properties ... 1136
Line Protocol Composer ... 1137
Tag Field Parameter Properties ... 1140
Data Values Placeholder ... 1145
Configuration Files ... 1148
7.3.3.8 File ... 1161
Configuration ... 1162
File Wizard ... 1167
File Action Wizard ... 1175
Data Value Placeholder Wizard ... 1191
Data Values Placeholder ... 1198
Configuration Files ... 1203
7.3.3.9 Web Service ... 1217
Configuration ... 1217
Samples ... 1237
7.3.4 Settings ... 1267
7.3.4.1 Diagnostic ... 1268
Configuration ... 1268

7.3.4.2 Web Server ... 1271
Configuration ... 1271
7.3.4.3 Advanced ... 1273
Configuration ... 1273
7.3.4.4 Licenses ... 1286
Configuration ... 1286
7.3.5 General ... 1292
7.3.5.1 Global Initial and Alias Values ... 1293
7.3.5.2 Trigger Wizard ... 1296
Welcome ... 1296
Trigger Type ... 1297
Trigger Type (Item, Timer, Variable) ... 1298
Trigger Type (Timer, Item, Variable) ... 1300
Trigger Item ... 1301
Timer Selection ... 1304
Timer Identity ... 1305
Timer Type ... 1306
Schedule Timer Settings ... 1308
Interval Timer Settings ... 1309
7.3.5.3 Item Trigger Wizard ... 1310
Welcome ... 1310
Trigger Item ... 1311
7.3.5.4 Timer Trigger Wizard ... 1314
Welcome ... 1314
Timer Selection ... 1315
Timer Identity ... 1317
Timer Type ... 1318
Schedule Timer Settings ... 1320
Interval Timer Settings ... 1321
7.3.5.5 Variable Trigger ... 1322
Variable Trigger Wizard ... 1323
Condition Expression Composer Wizard ... 1335

© Softing Industrial Automation GmbH

1 Welcome to Softing dataFEED OPC Suite

Welcome to Softing dataFEED OPC Suite V5.38.



▼ Quickstart Softing dataFEED OPC Suite

- Avoiding DCOM @ dataFEED OPC Suite OPC Tunnel.
- Convert OPC Data @ dataFEED OPC Suite OPC Bridge.
- Optimize OPC Communication Traffic @ dataFEED OPC Suite OPC Optimizer.
- Consolidate and Harmonize OPC Name Spaces @ dataFEED OPC Suite OPC Concentrator.
- Filter OPC Data @ dataFEED OPC Suite OPC Filter.
- Exchange Data Between OPC Servers @ dataFEED OPC Suite OPC Server2Server.
- Exchange Data Between OPC Clients @ dataFEED OPC Suite OPC Client2Client.

- Archive OPC Data @ dataFEED OPC Suite OPC2File.
- Archive OPC Data @ dataFEED OPC Suite OPC2Database.
- Protection Against Data Loss @ dataFEED OPC Suite OPC Store and Forward.

▾ Introduction Softing dataFEED OPC Suite

- Avoiding DCOM @ dataFEED OPC Suite OPC Tunnel.
- Converting OPC Data dataFEED OPC Suite OPC Bridge.
- Optimize OPC Communication Traffic @ dataFEED OPC Suite OPC Optimizer.
- Consolidate and Harmonize OPC Name Spaces @ dataFEED OPC Suite OPC Concentrator.
- Filter OPC Data @ dataFEED OPC Suite OPC Filter.
- Exchange Data Between OPC Servers @ dataFEED OPC Suite OPC Server2Server.
- Exchange Data Between OPC Clients dataFEED OPC Suite OPC Client2Client.
- Archive OPC Data dataFEED OPC Suite OPC OPC2File.
- Archive OPC Data @ dataFEED OPC Suite OPC OPC2Database.
- Protection Against Data Loss @ dataFEED OPC Suite OPC Store And Forward.

▾ Operation of Softing dataFEED OPC Suite

- Install Softing dataFEED OPC Suite @ dataFEED OPC Suite Installation.
- Install and manage licenses for Softing dataFEED OPC Suite @ dataFEED OPC Suite Licensing.
- Operate Softing dataFEED OPC Suite @ dataFEED OPC Suite Operation.

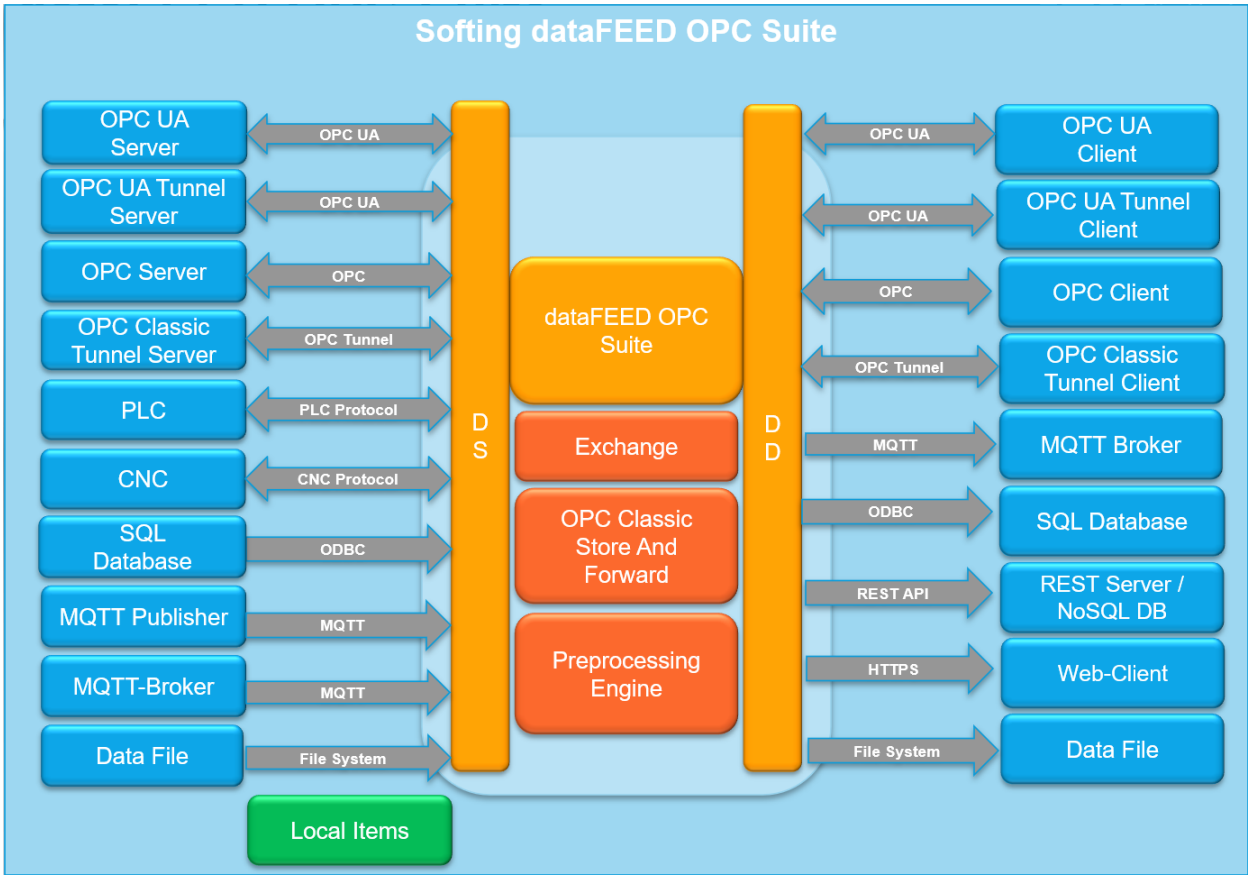
▾ Configuration of Softing dataFEED OPC Suite

© Softing Industrial Automation GmbH

- Learn the configuration concept of Softing dataFEED OPC Suite @ dataFEED OPC Suite Configuration Concepts.
- Operate the Softing dataFEED OPC Suite configurator @ dataFEED OPC Suite Configurator.
- Configure local dataFEED OPC Suite applications @ dataFEED OPC Suite Configuration.

Softing dataFEED OPC Suite V5.38
© 2024 Softing Industrial Automation GmbH

2 dataFEED OPC Suite Quickstart



▼ Softing dataFEED OPC Suite Quickstart

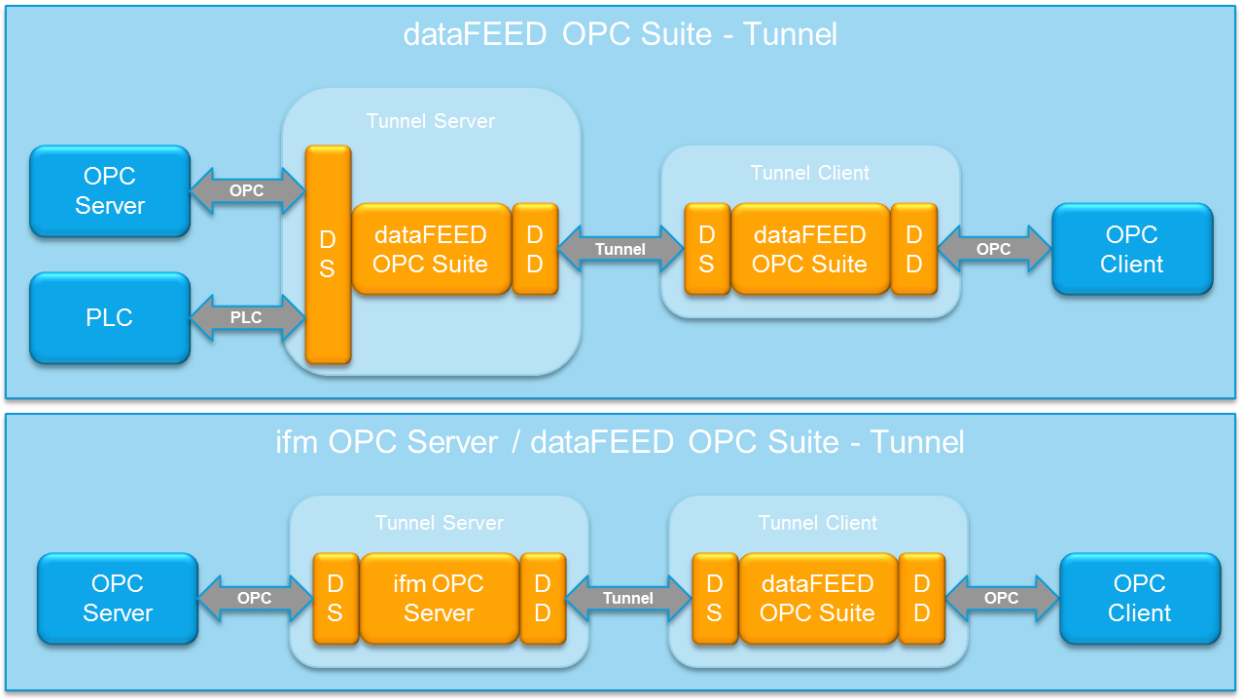
- Avoiding DCOM @ dataFEED OPC Suite OPC Tunnel.
- Convert OPC data @ dataFEED OPC Suite OPC Bridge.
- Optimize OPC communication traffic @ dataFEED OPC Suite OPC Optimizer.
- Consolidate and harmonize OPC namespaces @ dataFEED OPC Suite OPC Concentrator.
- Filter OPC data @ dataFEED OPC Suite OPC Filter.
- Exchange data between OPC servers @ dataFEED OPC Suite OPC Server2Server.
- Exchange data between OPC clients @ dataFEED OPC Suite OPC Client2Client.
- Archive OPC data to a file @ dataFEED OPC Suite OPC2File.
- Archive OPC data to a database @ dataFEED OPC Suite OPC2Database.

© Softing Industrial Automation GmbH

- Protection against data loss @ dataFEED OPC Suite OPC Store And Forward.
- Configuration of OPC UA @ dataFEED OPC Suite OPC UA Client configuration

Softing dataFEED OPC Suite V5.38
© 2024 Softing Industrial Automation GmbH

2.1 OPC Classic Tunnel



The Softing OPC Classic Tunnel provides you with a secure and easy possibility of cross-network OPC communication. The Softing OPC Classic Tunnel eliminates the configuration problems that were frequently encountered previously with the DCOM security settings!

You can find a more detailed description of the OPC Classic Tunnel functionality under Introduction - OPC Classic Tunnel.

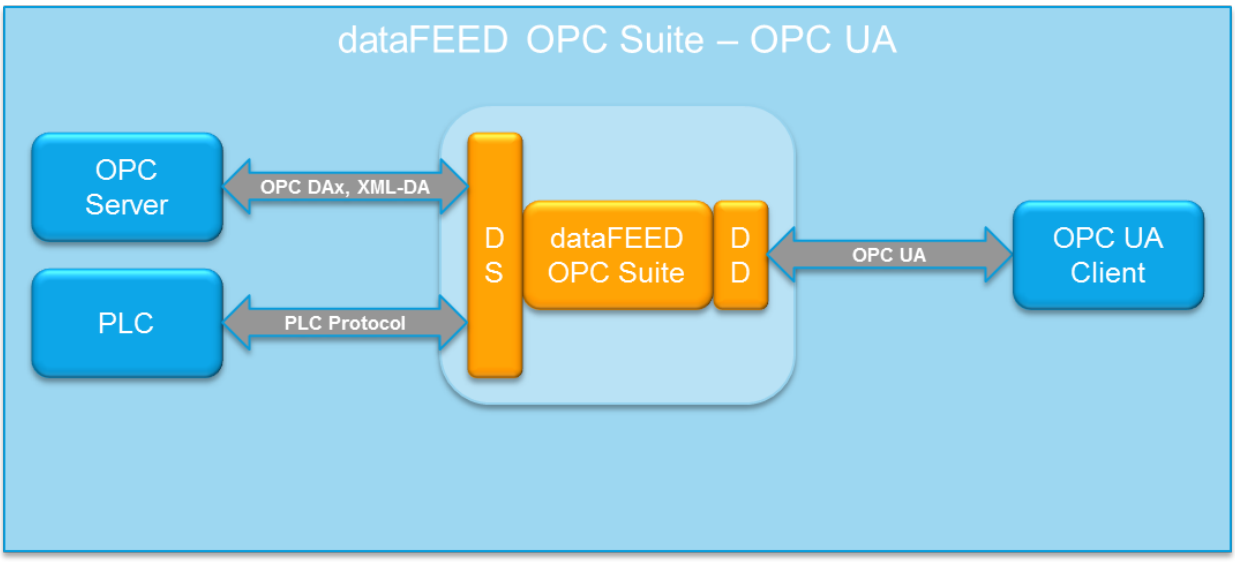
▼ To configure the OPC Classic Tunnel Server, the following configuration steps should be executed:

- Add one or more server connections to the corresponding OPC Server(s) as data source(s) @ Configuration: Data Source - OPC Server.
- Enable and configure the OPC Tunnel client data destination @ Configuration: Data Destination - OPC Classic Tunnel Client.

▼ To configure the OPC Classic Tunnel Client, the following configuration steps should be executed:

- Add a single server connection to the OPC Tunnel server as a data source @ Configuration: Data Source - OPC Classic Tunnel Server.
- Enable and configure the OPC client data destination @ Configuration: Data Destination - OPC Client.

2.2 OPC UA Server



The configuration of the OPC UA functionality is described in the chapter OPC UA Client - Configuration.

The following structure is displayed in OPC UA Clients, connected with dataFEED OPC Suite .

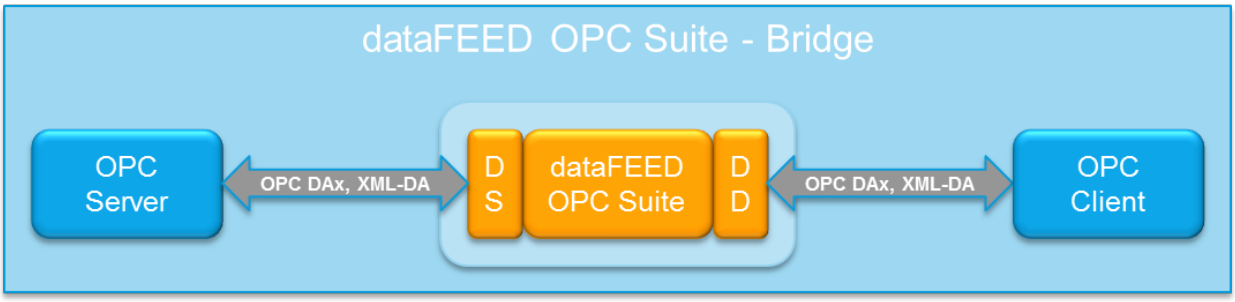
The namespace is displayed as a tree structure in the OPC UA Client. Under "OPC and Tunnel", the items, which are imported in the data source, can be browsed. Under "Server", you can find OPC UA specific diagnostic information.

© Softing Industrial Automation GmbH

The configuration of the OPC UA communication includes the following steps:

- Add one or more server connections as data source(s) @ Configuration: Data Source - OPC Server.
- Enable and configure the OPC UA client data destination @ Configuration: Data Destination - OPC UA Client.

2.3 OPC Bridge



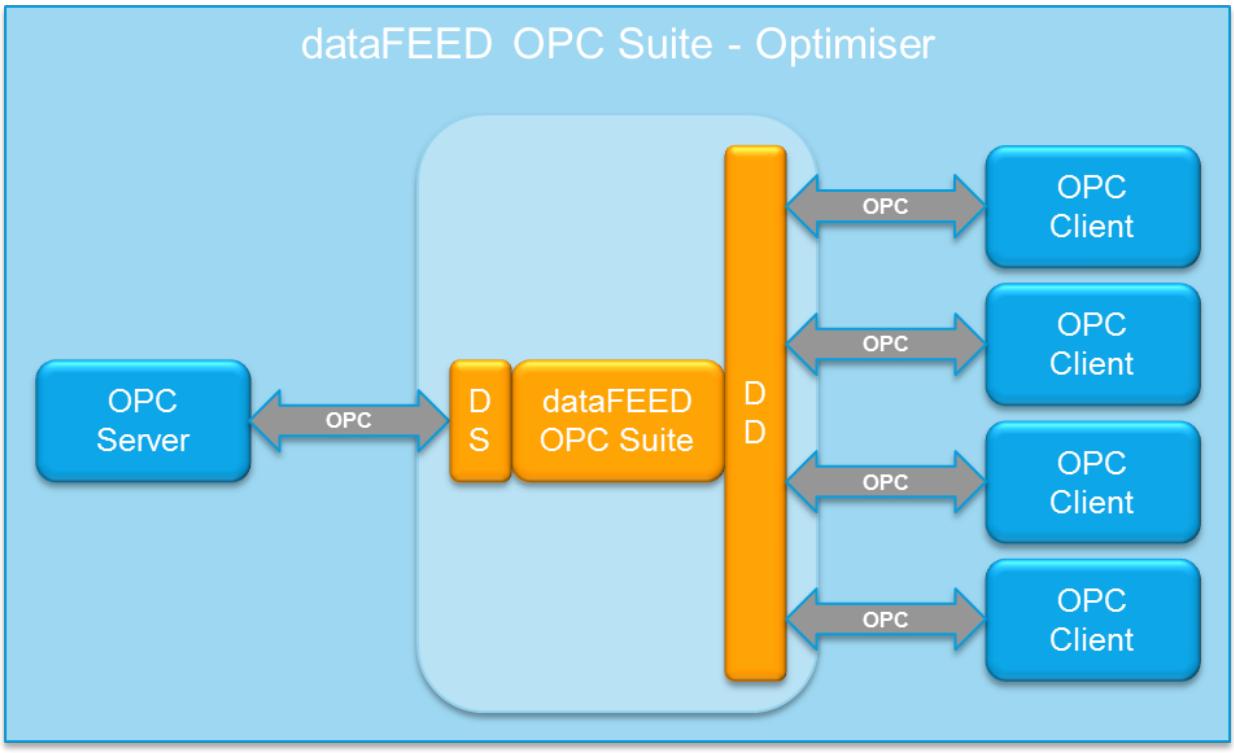
The Softing OPC Bridge gives one or more OPC Data Access and/or OPC XML-DA client applications quick and easy access to any type of OPC data access server. Vice versa, it allows any OPC data access server to be addressed by one or more data OPC access clients and/or OPC XML-DA client applications.

A more detailed description about the Softing OPC Bridge functionality can be found under Introduction - OPC Bridge.

▼ In order to configure the OPC Bridge, the following configuration steps should be executed:

- Add one or more server connections to the corresponding OPC Server(s) as data source(s) @ Configuration: Data Source - OPC Server.
- Enable and configure the OPC client data destination @ Configuration: Data Destination - OPC Client.

2.4 OPC Optimizer



The Softing OPC Optimizer allows multiple OPC clients to access an OPC server in such a way that workload on the server is minimized. According to the definition and functionality of Data Access, every OPC client communicating with the DA server creates groups and items in the DA server. The management of many groups simultaneously as well as the processing of multiple read requests to the often identical OPC items leads to a high communication load in the OPC server.

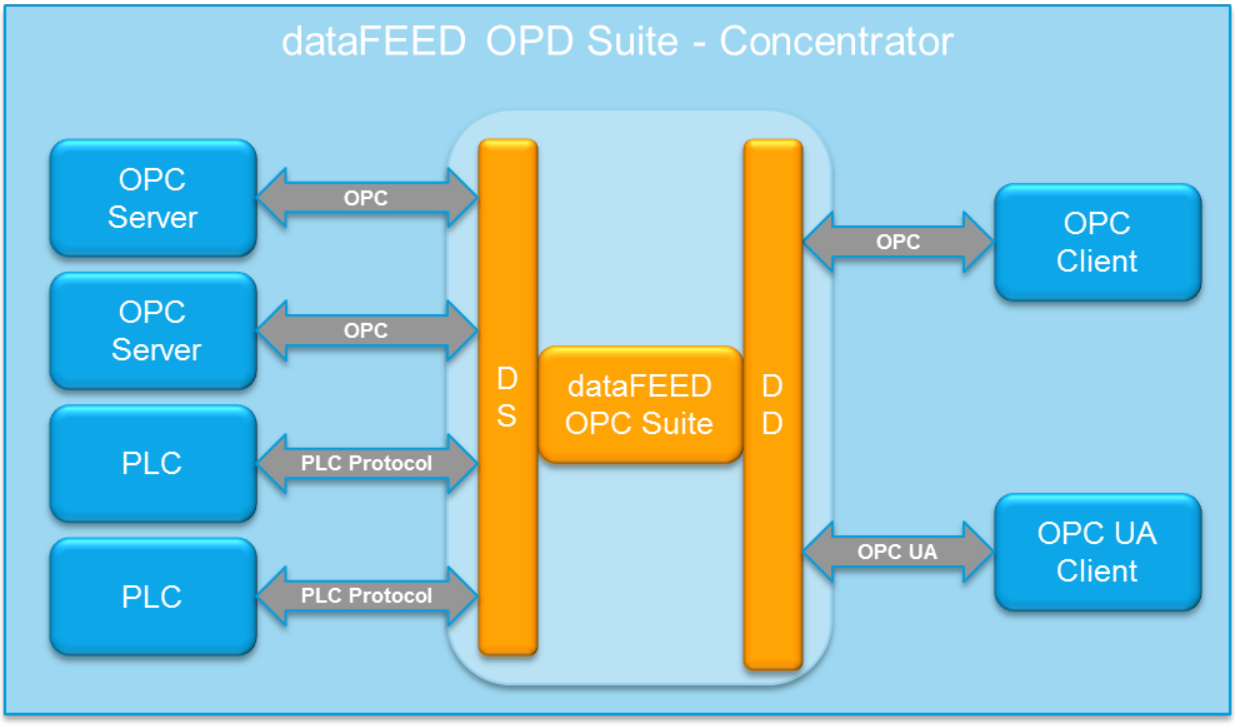
A more detailed description about the OPC Optimizer functionality can be found under Introduction - OPC Optimizer.

▼ In order to configure the OPC Optimizer, the following configuration steps should be executed:

- Add a server connection to the corresponding OPC Server as a data source @ Configuration: Data Source - OPC Server.
- Enable and configure the OPC client data destination @ Configuration: Data Destination - OPC Client.

© Softing Industrial Automation GmbH

2.5 OPC Concentrator



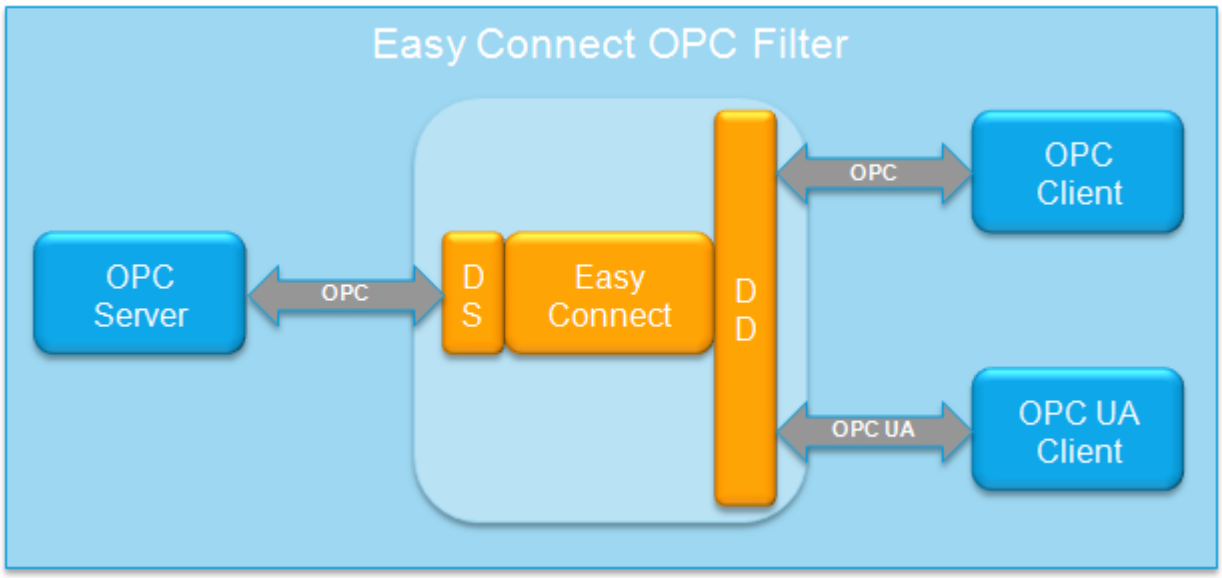
The Softing OPC Concentrator combines multiple OPC servers in one by consolidating the namespaces and OPC items of multiple OPC servers in one OPC server. This saves the project engineer valuable work and time when setting up client applications and facilitates the management of large systems. Multiple OPC clients can access multiple OPC servers via one OPC Concentrator

A more detailed description about the OPC Concentrator functionality can be found under Introduction - OPC Concentrator.

▾ In order to configure the OPC Concentrator, the following configuration steps should be executed:

- Add server connections to the corresponding OPC Servers as data sources @ Configuration: Data Source - OPC Server.
- Enable and configure the OPC client data destination @ Configuration: Data Destination - OPC Client.

2.6 OPC Filter



The Softing OPC Filter allows customizing the items that are provided by an OPC server. It makes no difference whether communication with the OPC server is via DCOM, XML-DA or the OPC Tunnel.

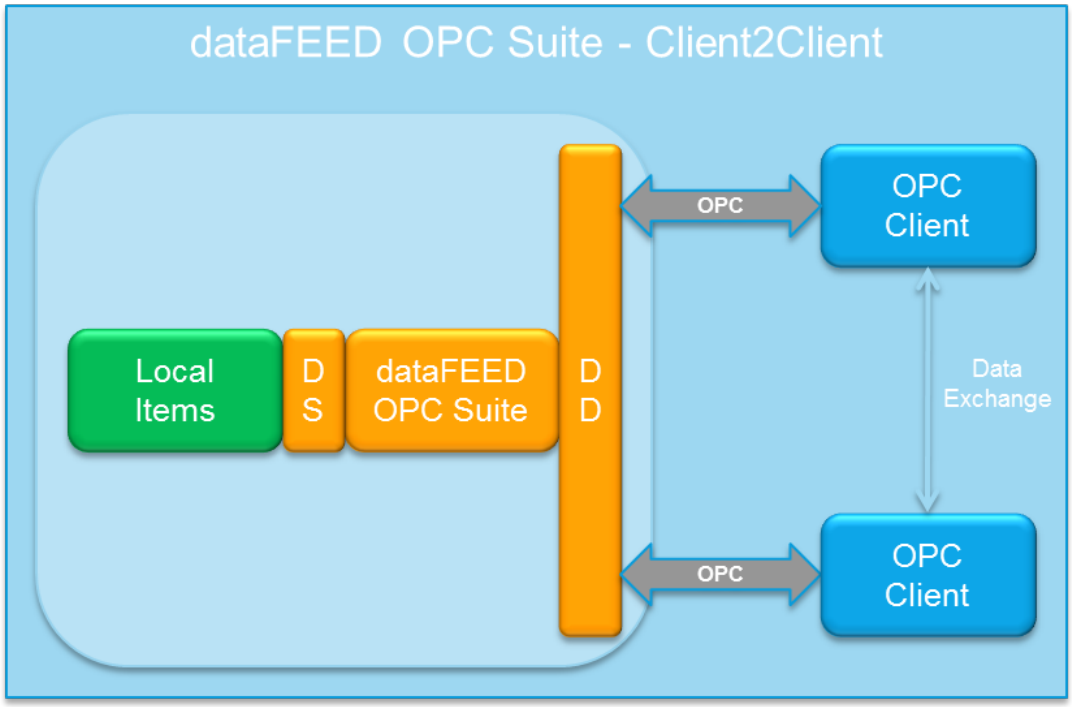
A more detailed description about the OPC Concentrator functionality can be found under Introduction - OPC Filter.

▼ In order to configure the OPC Filter, the following configuration steps should be executed:

- Add a server connection to the corresponding OPC or OPC Tunnel server as a data source @ Configuration: Data Source - OPC Server or Configuration: Data Source - Tunnel Server.
- Activate the OPC Filter in the corresponding server connection wizard @ OPC Server Wizard - Filter or OPC Tunnel Wizard - Filter.
- Define the filtered data items and specify their properties @ OPC Server Wizard - Filter Items or OPC Tunnel Wizard - Filter Items.
- Enable and configure the OPC client data destination @ Configuration: Data Destination - OPC Client.

© Softing Industrial Automation GmbH

2.7 OPC Client2Client



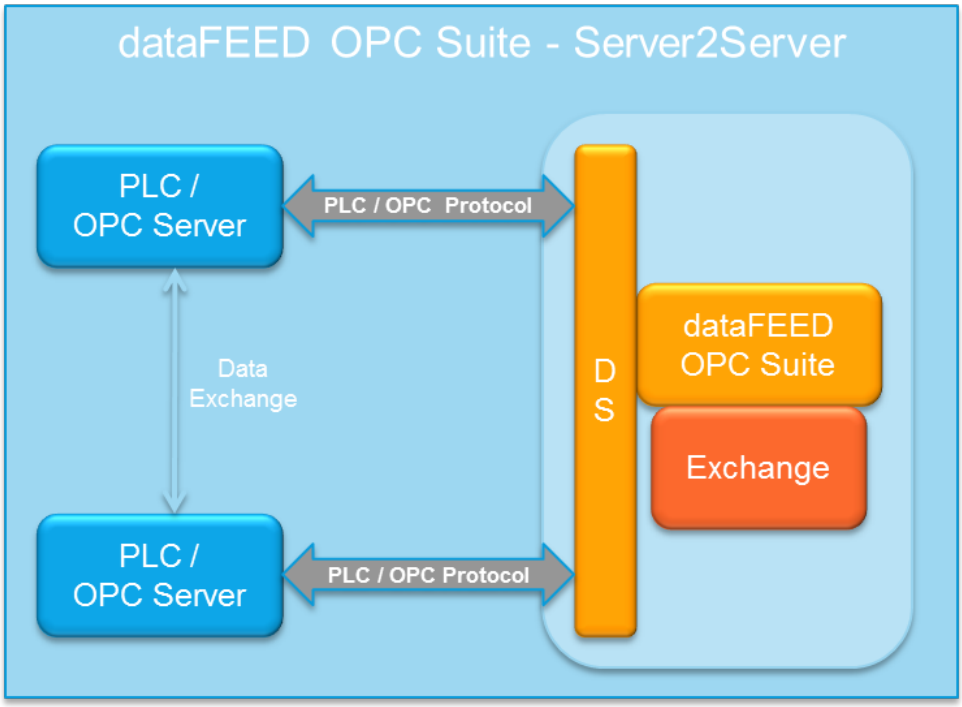
Softing OPC Client2Client allows an OPC client to transmit data to one or more other OPC clients. OPC Client2Client thereby acts as an intelligent data memory which can receive configurable OPC items from an OPC client and pass them on to another OPC client. The OPC namespace of OPC Client2Client can be statically defined in the dataFEED OPC Suite Configurator by defined corresponding Local Items or extended dynamically by the OPC clients by creating items.

A more detailed description about the OPC Client2Client functionality can be found at Introduction - OPC Client2Client.

▼ In order to configure the OPC Client2Client, the following configuration steps should be executed:

- Define the corresponding static OPC Client2Client namespace by defining corresponding local items as a data source @ Configuration: Data Source - Local Items.
- Enable and configure the OPC client data destination @ Configuration: Data Destination - OPC Client.

2.8 OPC Server2Server



dataFEED OPC Suite OPC Server2Server allows two OPC Servers to exchange data between each other. Both OPC Servers are acting as a data source, introducing data items to the namespace of the local dataFEED OPC Suite application. The Exchange module of Softing dataFEED OPC Suite is responsible for actually copying data between the corresponding data items from one OPC server to the other, therefore realizing the OPC Server2Server functionality.

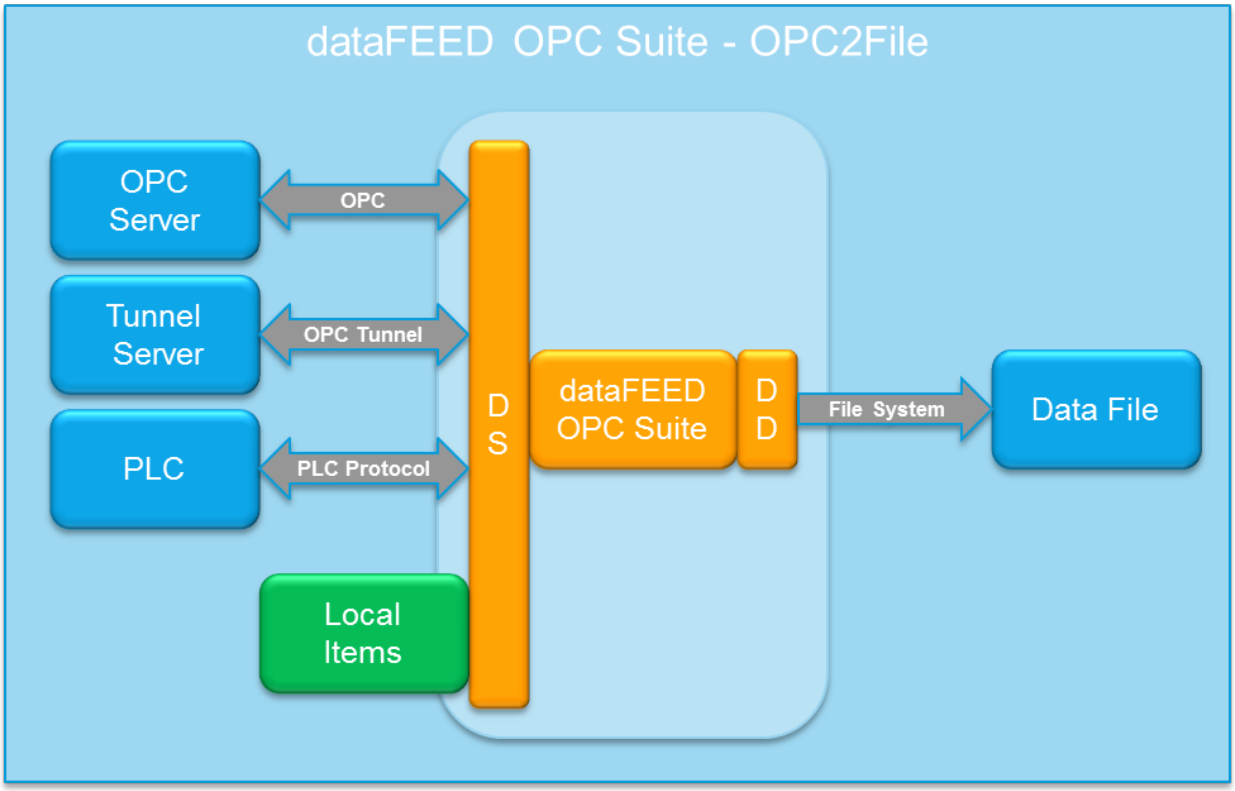
A more detailed description about the OPC Server2Server functionality can be found at Introduction - OPC Server2Server.

▼ In order to configure OPC Server2Server, the following configuration steps should be executed:

- Add a server connection to the first OPC server as a first data source @ Configuration: Data Source - OPC Server.
- Add a server connection to the second OPC server as a second data source @ Configuration: Data Source - OPC Server.
- Enable the OPC Exchange functionality and define the corresponding Exchange actions @ Configuration: Data Processing - Exchange.

© Softing Industrial Automation GmbH

2.9 OPC2File



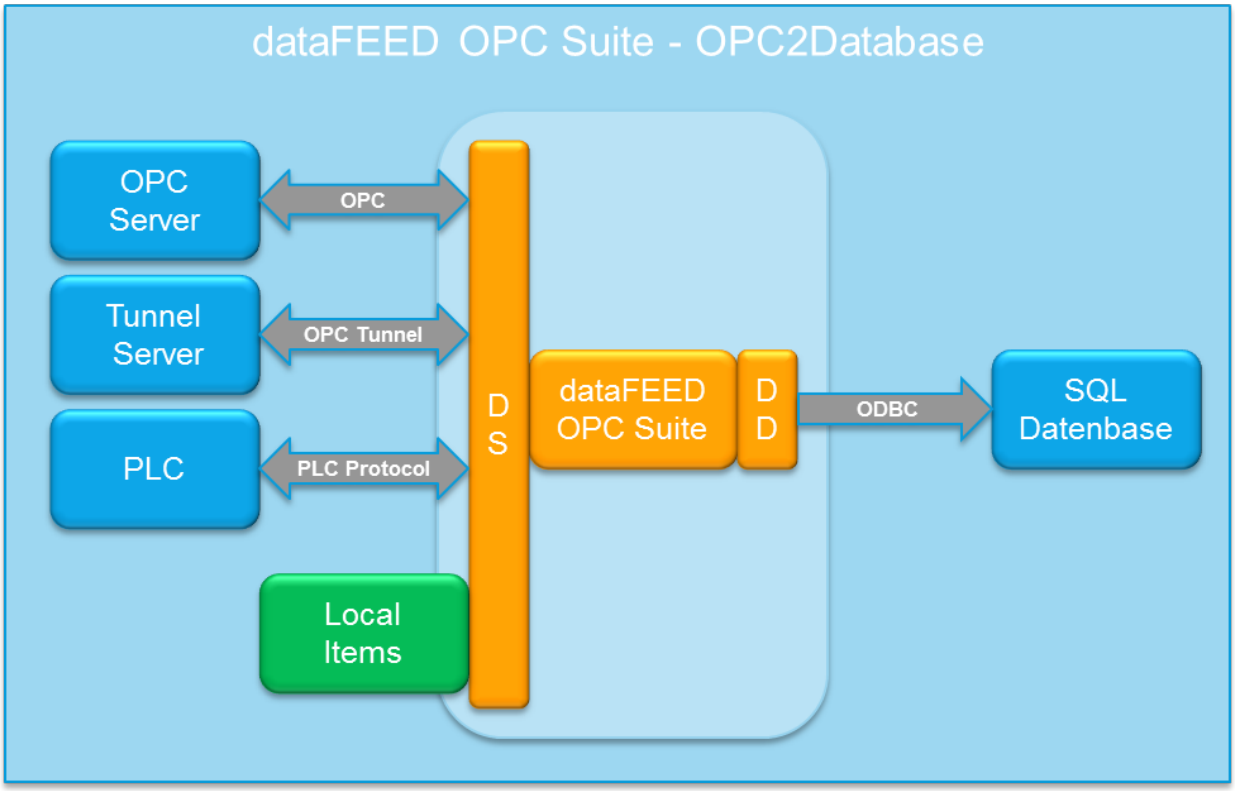
Softing OPC2File stores data from any OPC Data Access, OPC Tunnel or OPC XML-DA server in corresponding text files. Different types of text files like *.txt, *.xml or *.html are supported. The data handled by Softing dataFEED OPC Suite are written to file by executing simple file text actions.

A more detailed description about the OPC2File functionality can be found at Introduction - OPC2File.

▼ In order to configure the OPC2File, the following configuration steps should be executed:

- Add one or more server connections to the corresponding OPC or OPC Tunnel server(s) as data source(s) @ Configuration: Data Source - OPC Server or Configuration: Data Source - Tunnel Server.
- Enable the OPC2File functionality @ Configuration: Data Destination - File.
- Add one or more output data files as data destinations @ File Definition Wizard
- Define the file actions for actually writing the data to the data files @ File Action Wizard.

2.10 OPC2Database



Softing OPC2Database stores data from any OPC Data Access, OPC Tunnel or OPC XML-DA server in SQL databases like Microsoft SQL Server, Oracle, MySQL, IBM DB2 and other ODBC-enabled databases. The data handled by Softing dataFEED OPC Suite are written to the database by executing simple database SQL actions.

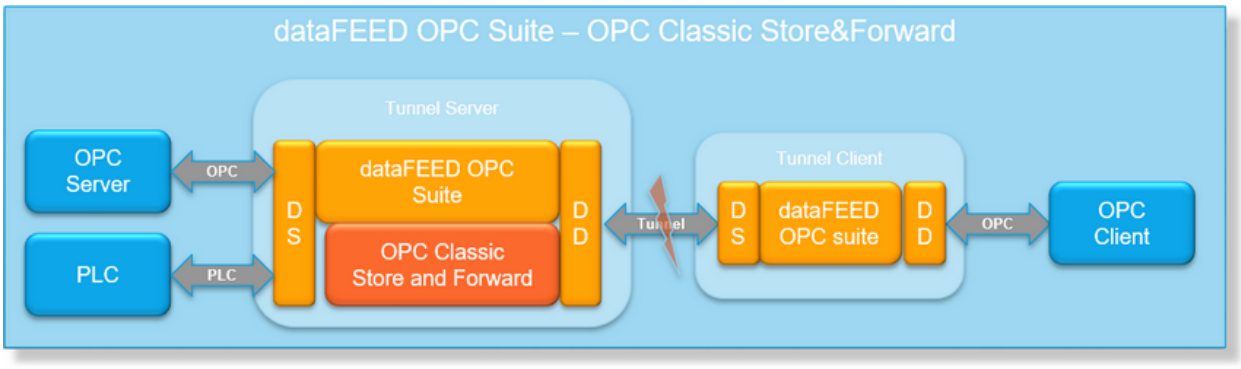
A more detailed description about the OPC2Database functionality can be found at Introduction - OPC2Database.

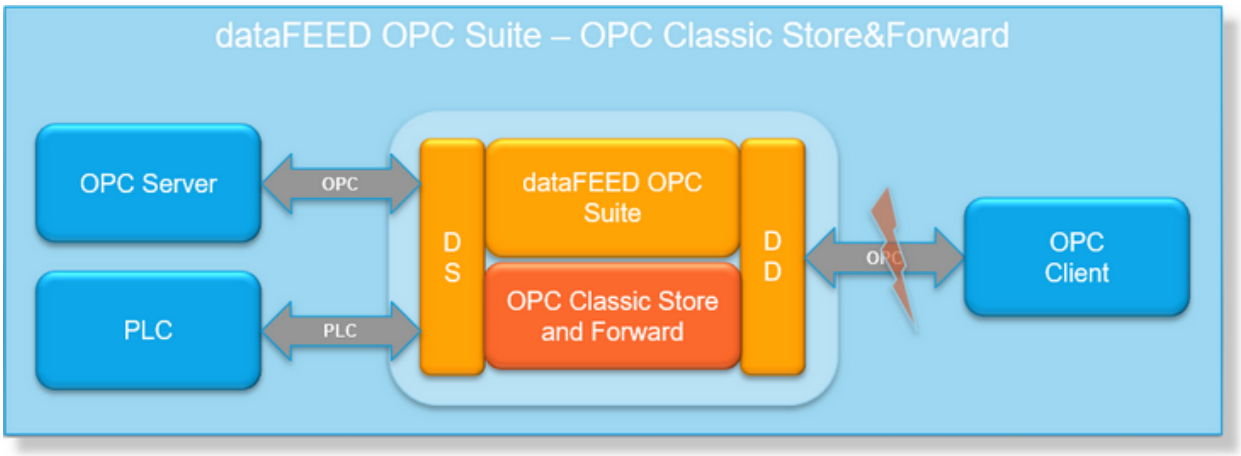
▼ In order to configure the OPC2Database, the following configuration steps should be executed:

- Add one or more server connections to the corresponding OPC or OPC Tunnel server(s) as data source(s) @ Configuration: Data Source - OPC Server or Configuration: Data Source - Tunnel Server.
- Enable the OPC2Database functionality @ Configuration: Data Destination - Database.
- Add one or more databases resp. ODBC data source connections as data destinations @ Database Definition Wizard
- Define the database actions for actually writing the data to the databases @ Database Action Wizard.

© Softing Industrial Automation GmbH

2.11 OPC Classic Store And Forward





dataFEED OPC Suite *OPC Classic Store And Forward* provides a configurable connection monitoring for one connection to an OPC or OPC Classic Tunnel client. If the local dataFEED OPC Suite detects a connection interruption, the corresponding data values are stored in a local data storage and forwarded to the corresponding client when the connection has been reestablished.

A more detailed description of the OPC Classic Store And Forward functionality can be found under Introduction - OPC Classic Store And Forward.

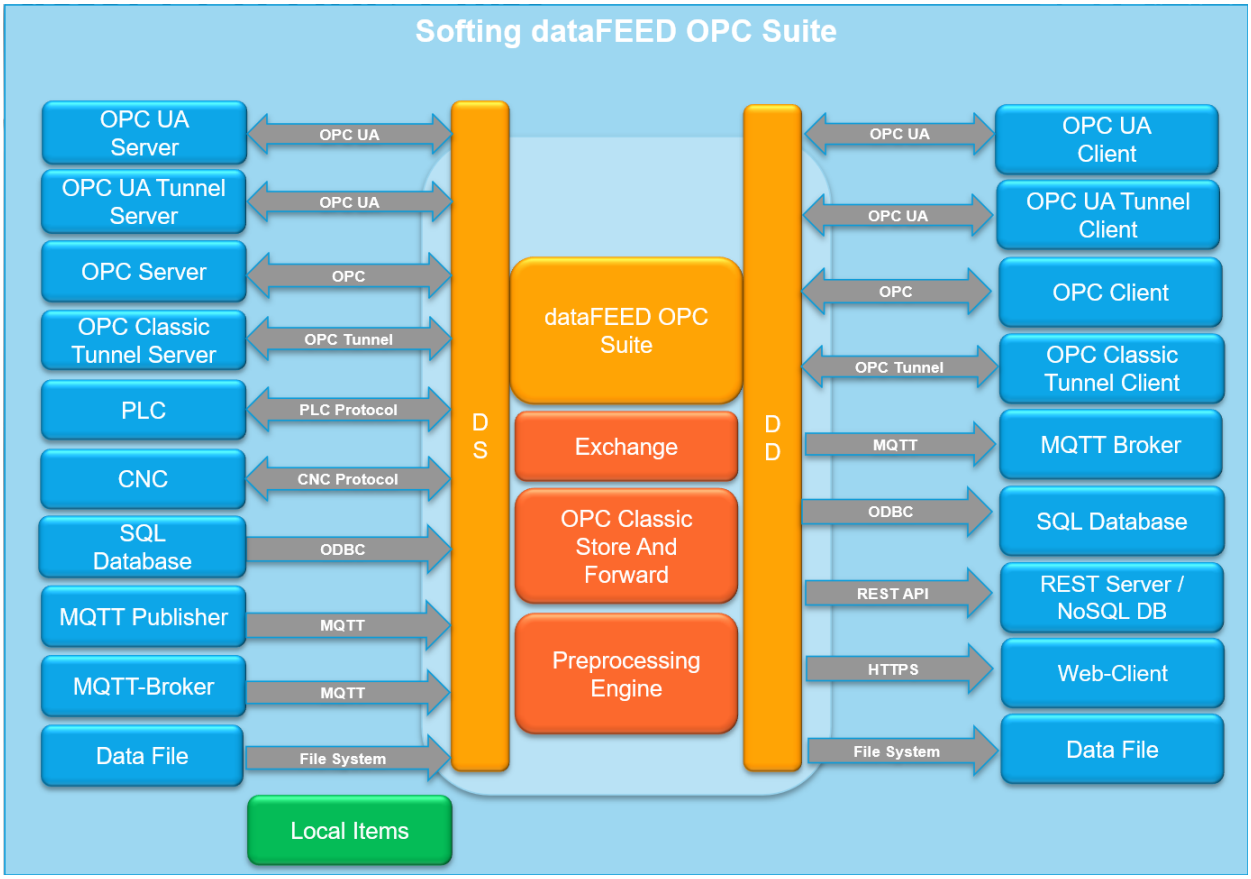
▼ To configure the OPC Classic Store And Forward, the following configuration steps should be executed:

- Add one or more server connections to the corresponding OPC or OPC Classic Tunnel server(s) as data source(s) @ Configuration: Data Source - OPC Server or Configuration: Data Source - OPC Classic Tunnel Server.

- Enable the OPC Classic Store And Forward functionality @ Configuration: Data Processing - Classic Store And Forward.
- Configure the local data storage and the data forwarding @ OPC Classic Store And Forward Wizard
- Define the data items to be monitored and specify their properties @ OPC Classic Store And Forward Wizard - Storage Items.

© Softing Industrial Automation GmbH

3 dataFEED OPC Suite Introduction

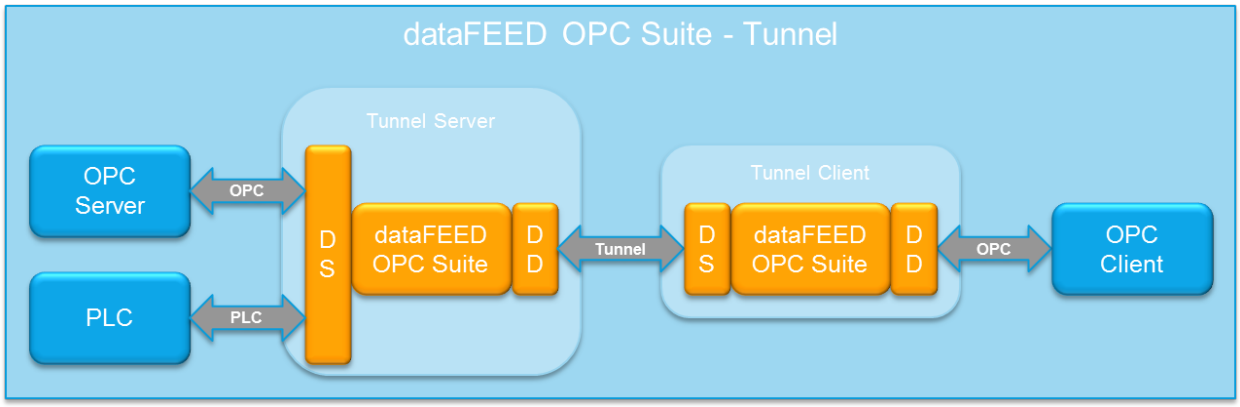


▼ Softing dataFEED OPC Suite Introduction

- Avoiding DCOM @ dataFEED OPC Suite OPC Tunnel.
- Convert OPC data @ dataFEED OPC Suite OPC Bridge.
- Optimize OPC communication traffic @ dataFEED OPC Suite OPC Optimizer.
- Consolidate and harmonize OPC namespaces @ dataFEED OPC Suite OPC Concentrator.
- Filter OPC data @ dataFEED OPC Suite OPC Filter.
- Exchange data between OPC servers @ dataFEED OPC Suite OPC Server2Server.
- Exchange data between OPC clients @ dataFEED OPC Suite OPC Client2Client.
- Archive OPC data to a file @ dataFEED OPC Suite OPC2File.
- Archive OPC data to a database @ dataFEED OPC Suite OPC2Database.

- Protection against data loss @ dataFEED OPC Suite OPC Store And Forward.
- OPC UA communication @ dataFEED OPC Suite OPC UA

3.1 OPC Classic Tunnel



Avoiding DCOM

The Softing OPC Classic Tunnel provides you with a secure and easy possibility of cross-network OPC communication. The Softing OPC Classic Tunnel eliminates the configuration problems that were frequently encountered previously with the DCOM security settings! You save yourself time and unnecessary questions to the manufacturer. The Softing OPC Classic Tunnel takes care of all the obstacles you usually run into when networking OPC clients and servers that are located on different computer platforms, in different network domains, or in completely separate networks. The high-performance data exchange is stable and reliable.

- Cross-platform OPC communication across firewall boundaries.
- Commissioning of cross-network OPC communication in only a few minutes.
- Fast detection of connection interruptions by monitoring to an accuracy of one second.
- OPC Tunnel automatically reestablishes the OPC connections after an interruption of communication.
- Unlike OPC via DCOM, the computers do not need to be opened to unauthorized access from the outside.

Quick Commissioning with Softing OPC Classic Tunnel

Designed as a “DCOM bypass,” the Softing OPC Classic Tunnel allows reliable, high-performance communication between OPC components on networked computers. For this purpose, the OPC Classic Tunnel is installed on both the OPC client computer and the OPC server computer. The OPC Classic Tunnel installations on the client side and the server side communicate via a TCP/IP connection that can optionally be encrypted. In this way, the data that

© Softing Industrial Automation GmbH

are exchanged between client and server applications are “tunneled” via TCP/IP; DCOM is avoided completely, and the time-consuming and nerve-racking configuration of the DCOM security settings is eliminated once and for all.

Configuring the OPC Classic Tunnel

To configure an OPC Tunnel two independent Softing dataFEED OPC Suite configurations are needed: One to represent the OPC Classic Tunnel server and one to represent the OPC Tunnel client. Typically these two configurations are executed on two different computers and communicate over the network OPC Classic Tunnel protocol.

Both the OPC Classic Tunnel server and the associated OPC Classic Tunnel client must be configured to use the same TCP/IP port number. The port numbers can be configured within the data destination OPC Classic Tunnel client configuration page on the OPC Classic Tunnel server, and within the corresponding data source OPC Classic Tunnel server connection wizard on the OPC Classic Tunnel client.

Attention: Please make sure that the configured TCP/IP port number is not used by other local dataFEED OPC Suite applications on the server and client computers and that your network infrastructure (like firewalls or switches) is prepared to forward TCP/IP packets using the configured port number.

Note: The OPC Classic Tunnel discovery service on the OPC Classic Tunnel client side will find the corresponding OPC Classic Tunnel server only if the OPC Classic Tunnel server is actually running.

- To configure the OPC Classic Tunnel Server, the following configuration steps should be executed:

  - Add one or more server connections to the corresponding OPC server(s) as data source(s) @ Configuration: Data Source - OPC Server.
  - Enable and configure the OPC Tunnel client data destination @ Configuration: Data Destination - OPC Classic Tunnel Client.

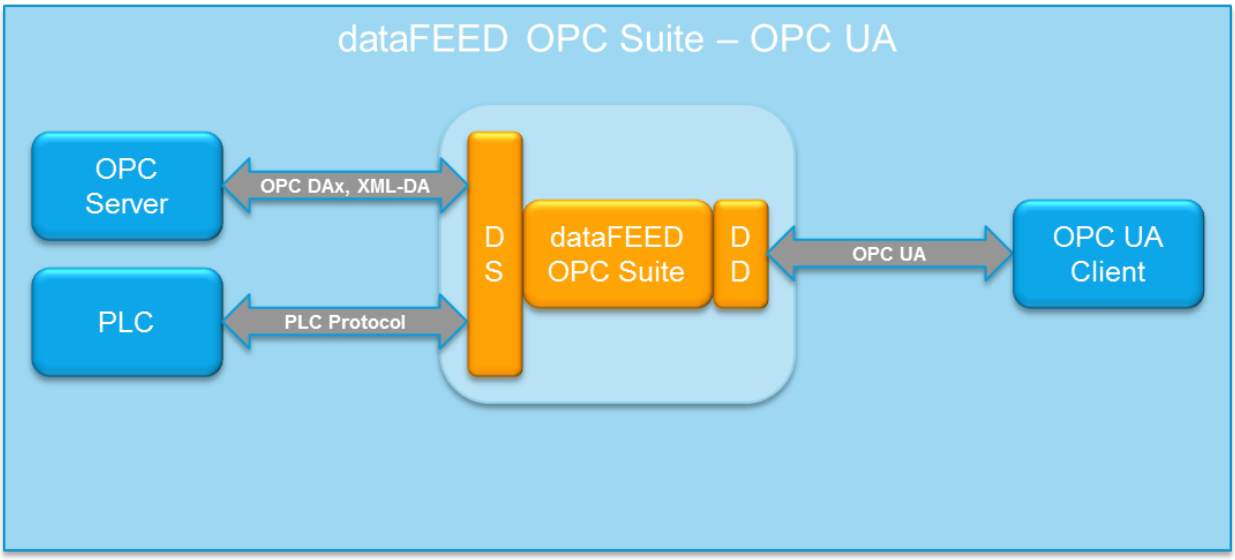
- To configure the OPC Classic Tunnel Client, the following configuration steps should be executed:

  - Add a single server connection to the OPC Classic Tunnel server as a data source @ Configuration: Data Source - OPC Classic Tunnel Server.
  - Enable and configure the OPC client data destination @ Configuration: Data Destination - OPC Client.

- Enabling additional functionalities for the OPC Classic Tunnel:

  - Filter the data items provided by the corresponding data sources @ Introduction - OPC Filter.
  - Protect against data loss @ Introduction - OPC Classic Store And Forward.

3.2 OPC UA



Adding an OPC UA Client as data destination offers the possibility to build up a communication between dataFEED OPC Suite and an OPC UA Client in a fast and easy way.

Thereby you can include OPC UA Clients, which are not running on Windows PCs, into your communication network.

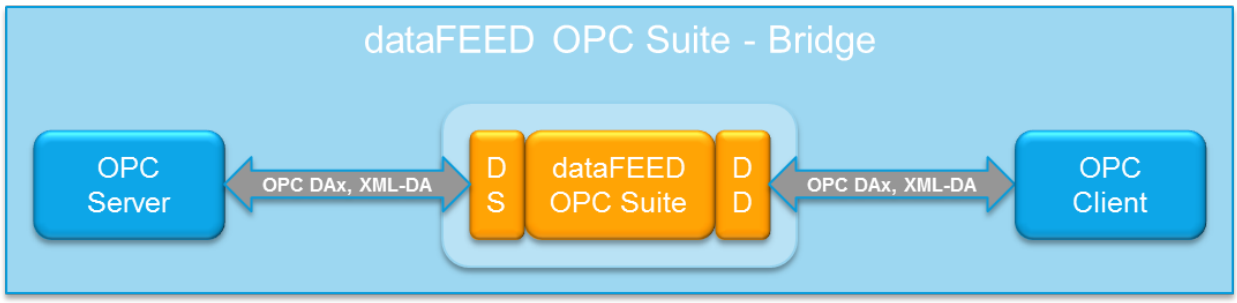
You can choose between different security mechanisms, like

- exchange of certificates
- authentication by username and password
- encryption of data with different security algorithms.

The configuration is explained in detail under Data destination - OPC UA Client - Configuration

© Softing Industrial Automation GmbH

3.3 OPC Bridge



The Softing OPC Bridge gives one or more OPC Data Access and/or OPC XML-DA client applications quick and easy access to any type of OPC data access server. Vice versa, it allows any OPC Data Access server to be addressed by one or more OPC Data Access clients and/or OPC XML-DA client applications.

Which OPC Data Access version (DA 1.0, DA 2.05 or DA 3) the OPC client or OPC server has implemented is irrelevant. The OPC Bridge automatically recognizes the implemented OPC specifications and converts them in such a way that even those OPC components can communicate which would not be compatible without the OPC Bridge. The OPC Bridge thus acts as a software gateway, "bridging" the different implementations.

From the point of view of an OPC client application, access to the configured OPC server through the OPC Bridge is fully transparent, which means that the OPC Bridge provides the namespace of the configured OPC server to a client without any changes.

Bridge Between Data Access Implementations

There are only minor differences between the OPC DA 1.0, DA 2.05 and DA 3 versions. Nevertheless, the more recent versions contain extensions that are not supported by previous implementations. As no compliance test tool exists for the OPC DA 1.0 Specification, no compliance verification can be provided for OPC DA 1.0 servers. The OPC Bridge turns a non-compliant OPC server into a compliant OPC server!

Bridge Between DA and XML DA Implementations

Substantial differences exist between the DCOM based Data Access Specifications and the web services and XML based XML DA Specification. XML DA servers can run on all operating systems that support web services. The XML DA client and server can communicate over the Internet across firewall boundaries. The OPC Bridge connects the DCOM world with the web services world. The OPC Bridge always needs to be installed in a PC supporting DCOM.

Configuring the OPC Bridge

To configure the OPC Bridge you mainly need to configure the data source connections to your corresponding OPC servers providing the OPC data you want to convert or bridge. For the OPC Bridge an arbitrary amount of data sources and OPC servers, respectively, can be used.

© Softing Industrial Automation GmbH

As a second configuration step, the OPC client data destination functionality of Softing dataFEED OPC Suite needs to be enabled and configured. Any necessary data or OPC specification conversion is then automatically done by the local dataFEED OPC Suite application during run time.

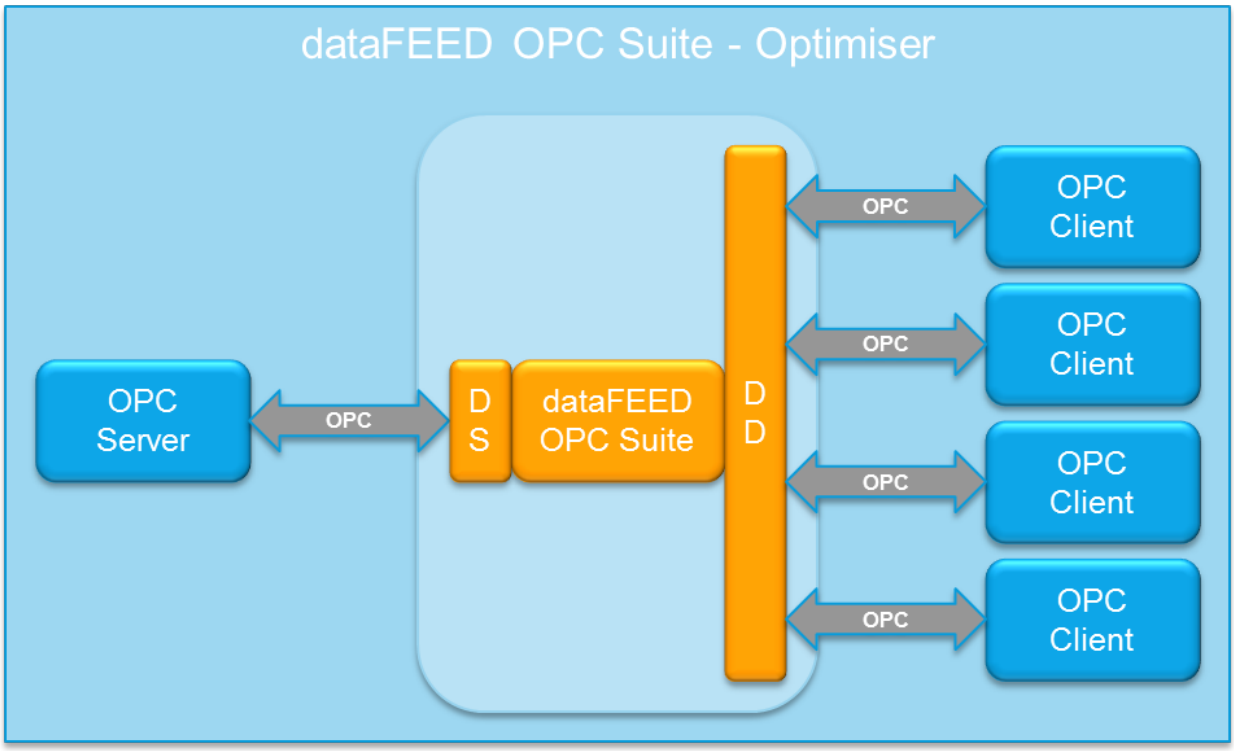
Note: The OPC client data destination functionality is enabled and already completely configured by default.

- In order to configure the OPC Bridge, the following configuration steps should be executed:

  - Add one or more server connections to the corresponding OPC server(s) as data source(s) @ Configuration: Data Source - OPC Server.
  - Enable and configure the OPC client data destination @ Configuration: Data Destination - OPC Client.

Additionally the data items provided by the different data sources can also be filtered using the OPC Filter as described in Introduction: OPC Filter.

3.4 OPC Optimizer



The Softing OPC Optimizer allows multiple OPC clients to access an OPC server in such a way that workload on the server is minimized. According to the definition and functionality of Data Access, every OPC client communicating with the OPC DA server creates groups and items in the DA server. The management of many groups simultaneously as well as the

© Softing Industrial Automation GmbH

processing of multiple read requests to the often identical OPC items leads to a high communication load in the OPC server.

As the name indicates, the OPC Optimizer optimizes read and write access to the items of an OPC server. Read access to identical OPC items with different update rates are combined and data acquisition from the peripherals by the OPC server is bundled using intelligent algorithms.

The configuration settings offer additional possibilities for optimization. You can specify, for example,

- that the OPC server should only read the data from the device again after the data have reached a certain age;
- that the update cycles are determined independently of the OPC clients.

Configuring the OPC Optimizer

To configure the OPC Optimizer you mainly need to configure the data source connection to your corresponding OPC server providing the OPC data you want to optimize. Generally, for the OPC Optimizer an arbitrary amount of data sources and OPC servers, respectively, can be used.

As a second configuration step, the OPC client data destination functionality of Softing dataFEED OPC Suite needs to be enabled and configured.

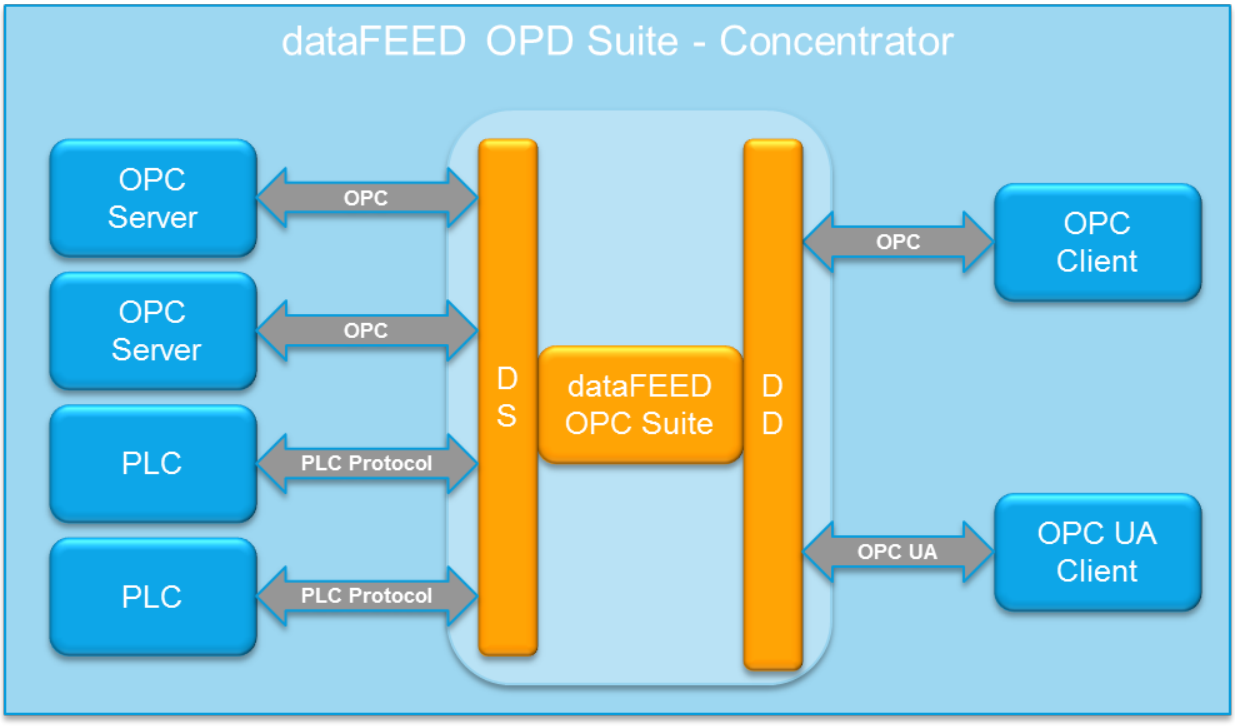
Note: The OPC client data destination functionality is enabled and already completely configured by default.

▾ In order to configure the OPC Optimizer, the following configuration steps should be executed:

- Add a server connection to the corresponding OPC server as a data source @ Configuration: Data Source - OPC Server.
- Enable and configure the OPC client data destination @ Configuration: Data Destination - OPC Client.

Additionally the data items provided by the different data sources can also be filtered using the OPC Filter as described in Introduction: OPC Filter.

3.5 OPC Concentrator



The dataFEED OPC Suite OPC Concentrator combines multiple OPC servers in one by consolidating the namespaces and OPC items of multiple OPC servers in one OPC server. This saves the project engineer valuable work and time when setting up client applications and facilitates the management of large systems.

Multiple OPC clients can access multiple OPC servers via one OPC Concentrator.

Harmonizing Data Access

Using the OPC Concentrator, several OPC Clients can access several OPC servers with different namespaces. Depending on the implementation of the vendor, OPC servers can provide a flat or a hierarchical namespace. In addition, the structure of an OPC server's namespace and the semantics to address variables in an OPC server significantly differ from vendor to vendor. OPC Concentrator helps to unify and harmonize access to the data in an OPC server.

Cascading of OPC Servers

Data of several OPC servers is bundled and concentrated into one OPC server by using the OPC Concentrator. Data of several OPC Concentrators DLLs or several OPC Concentrator applications on several computers can be concentrated into another OPC Concentrator. That way, complex systems with a large number of OPC servers can be optimally structured. Running the OPC Concentrators on different computers, which are interconnected via DCOM, can also improve the performance of the overall system.

© Softing Industrial Automation GmbH

Configuring the OPC Concentrator

To configure the OPC Concentrator you mainly need to configure the data source connections to your corresponding OPC servers providing the OPC data you want to concentrate. Each server connection needs to be assigned a unique connection name in order to distinguish the different data items within the namespace of the local dataFEED OPC Suite application.

As a second configuration step, the OPC client data destination functionality of Softing dataFEED OPC Suite needs to be enabled and configured.

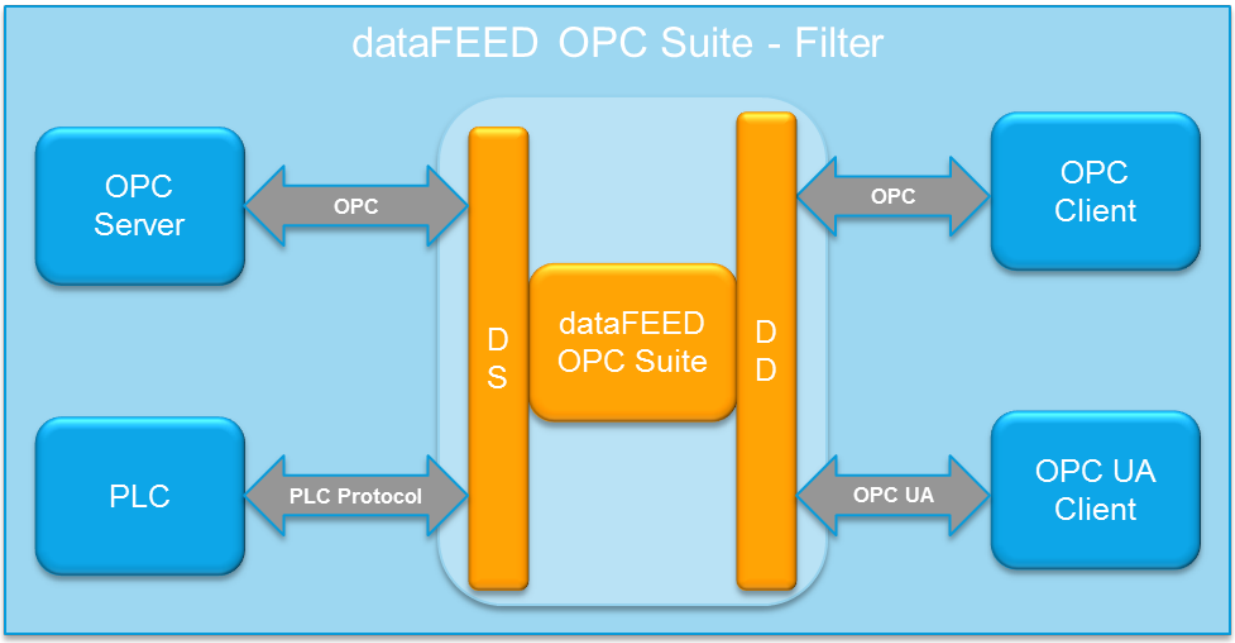
Note: The OPC client data destination functionality is enabled and already completely configured by default.

▼ In order to configure the OPC Concentrator, the following configuration steps should be executed:

- Add server connections to the corresponding OPC servers as data sources @ Configuration: Data Source - OPC Server.
- Enable and configure the OPC client data destination @ Configuration: Data Destination - OPC Client.

Additionally the data items provided by the different data sources can also be filtered using the OPC Filter as described in Introduction: OPC Filter.

3.6 OPC Filter



The Softing OPC Filter allows customizing the items that are provided by a specific data source – a corresponding OPC or OPC Tunnel server. With the Softing OPC Filter the following functionalities can be implemented:

- Hide data items.
- Change the data type of data items.
- Restrict the access rights for data items.
- Change the item ID of a data item as desired.

It makes no difference whether the corresponding data source is an OPC server communicating via OPC, OPC XML-DA or the OPC Tunnel protocol.

Limiting the Items

With the OPC Filter you can limit the items provided by an OPC server to those you need, and control the access rights. This enhances system security as incorrect access, whether accidental or intentional, is no longer possible.

Standardizing the OPC Namespace

You can use the OPC Filter to standardize the structure of the items in your system, independently of the OPC servers you are using. In this way you can, for example, replace a server without any influence on the OPC clients.

Configuring the OPC Filter

To configure the OPC Filter you mainly need to configure the data source connections to your corresponding OPC or OPC Tunnel servers providing the OPC data you want to filter. For the OPC Filter an arbitrary amount of data sources, OPC servers and OPC Tunnel servers, respectively, can be used.

The OPC Filter functionality needs to be activated separately for each server connection within the corresponding Server Connection Wizard. If the corresponding OPC or OPC Tunnel server is running, the dataFEED OPC Suite Configurator reads the available data items and presents them for filter item selection in the last page of the Server Connection Wizard. By checking or unchecking the corresponding items you can include / exclude (filter) the corresponding items from the namespace of the local dataFEED OPC Suite application.

If the corresponding OPC or OPC Tunnel server is not running, you can manually add data items to the local namespace using the buttons on the right side of the wizard page and the context sensitive menu in the namespace tree control. Additionally you can import a predefined list of filter items from an external filter file.

As a second configuration step, the OPC client data destination functionality of Softing dataFEED OPC Suite needs to be enabled and configured.

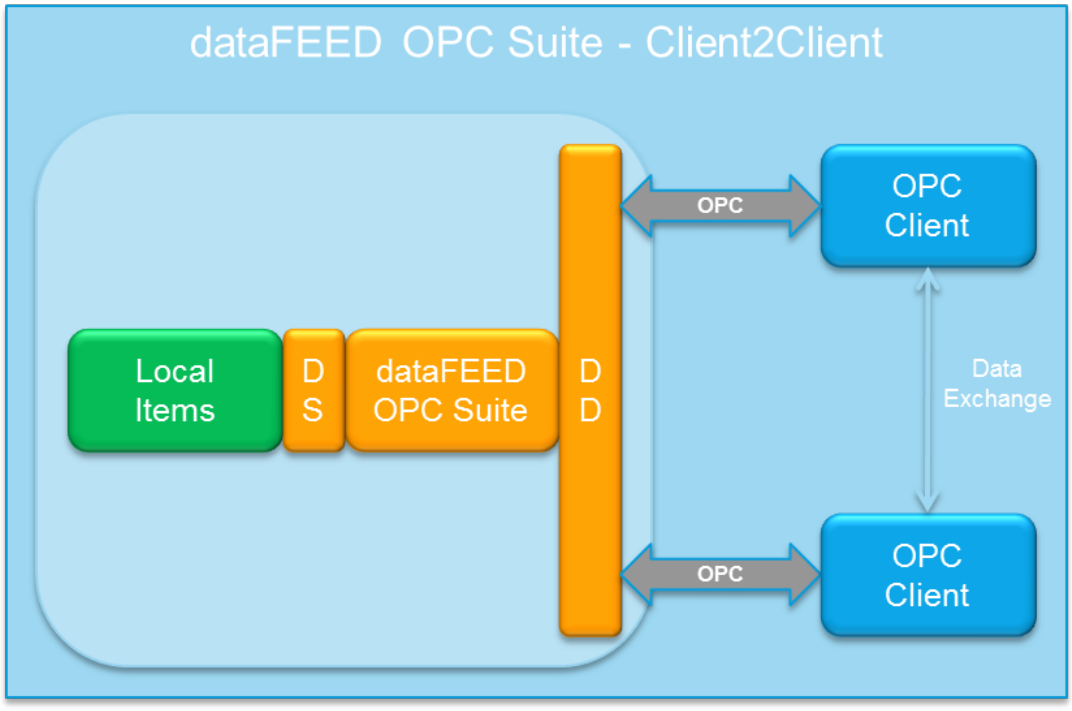
© Softing Industrial Automation GmbH

Note: The OPC client data destination functionality is enabled and already completely configured by default.

In order to configure the OPC Filter, the following configuration steps should be executed:

- Add a server connection to the corresponding OPC or OPC Tunnel server as a data source @ Configuration: Data Source - OPC Server or Configuration: Data Source - Tunnel Server.
- Activate the OPC Filter in the corresponding server connection wizard @ OPC Server Wizard - Filter or OPC Tunnel Wizard - Filter.
- Define the filtered data items and specify their properties @ OPC Server Wizard - Filter Items or OPC Tunnel Wizard - Filter Items.
- Enable and configure the OPC client data destination @ Configuration: Data Destination - OPC Client.

3.7 OPC Client2Client

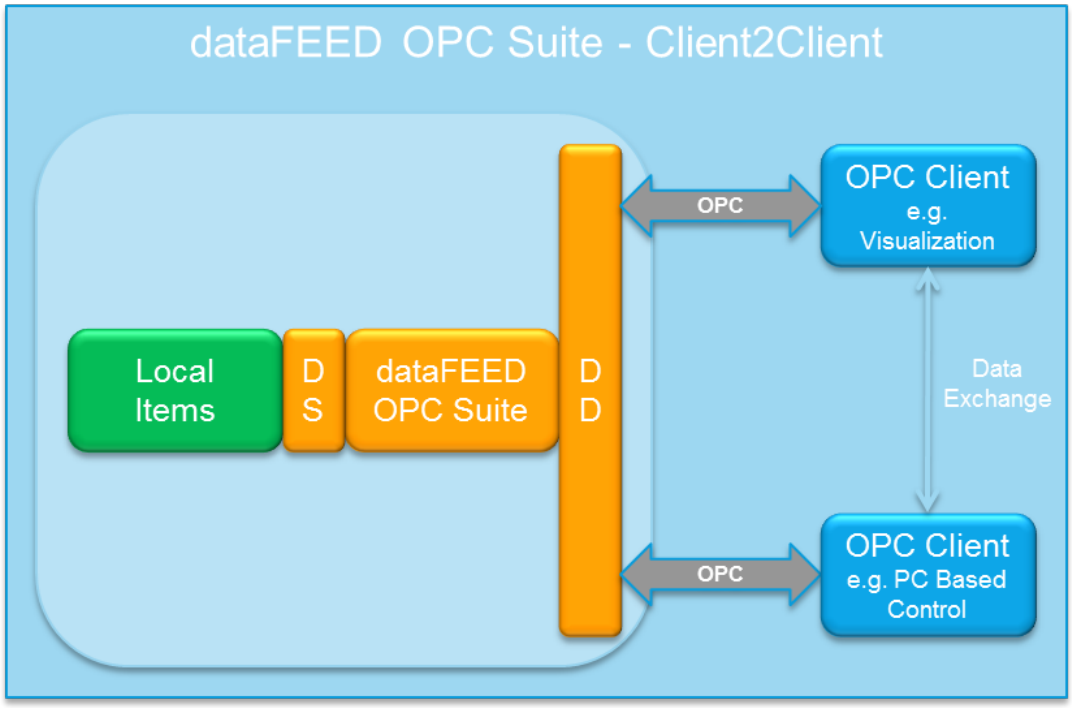


Softing OPC Client2Client allows an OPC client to transmit data to one or more other OPC clients. OPC Client2Client thereby acts as an intelligent data memory which can receive configurable OPC items from an OPC client and pass them on to another OPC client. In this way, OPC Client2Client allows supervisory control systems, SCADA systems or any applications that do not have an OPC server interface to exchange data directly. OPC Client2Client is also suitable for testing OPC clients by configuring the tool as an OPC server with any desired server name, namespace and OPC items. You can use any OPC client, e.g.

the Softing OPC Demo Client, to write values to OPC Client2Client, either manually or script controlled. This allows you to simulate real-world conditions and specific fault situations in the test lab.

OPC Client2Client as ”Gateway” Between OPC Client Applications

OPC Client2Client can be started from any OPC client applications and is available to any OPC client applications as soon as it has been started as a service. The first time an OPC client application accesses OPC Client2Client, the default namespace of OPC Client2Client is provided, for example, Node1 with Tag1 and Tag2, Node2 with Tag1 and Tag2.



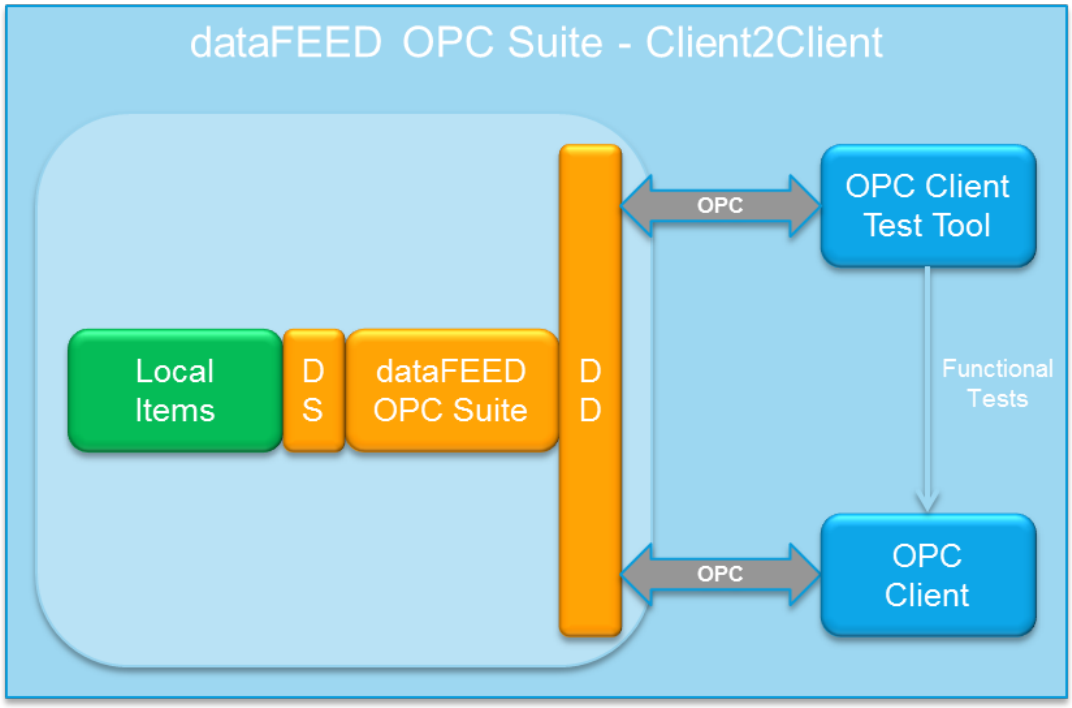
The OPC namespace of OPC Client2Client can be statically defined in the dataFEED OPC Suite Configurator by defining the corresponding Local Items, or extended dynamically by the OPC clients by creating items. The items of the namespace may be of any data type covered by the OPC Specifications. Arrays of fixed and variable length are also supported for these data types. All values of OPC Client2Client are shown to be of ”bad” quality until they have been written to by an OPC client for the first time. The Local Items system item allows monitoring communication between OPC Client2Client and the OPC client(s).

OPC Client2Client as OPC Server Simulation for Testing OPC Client Applications

OPC Client2Client also allows one of the OPC clients to function logically as a server. For this purpose, OPC Client2Client can be configured with a defined server name, Program ID and Class ID, a defined namespace can be created and OPC Client2Client items can be written to, e.g. under script control, by an OPC client application. OPC Client2Client also allows writing the quality and timestamp of each item. This allows simulating real-world conditions for testing OPC

© Softing Industrial Automation GmbH

clients without devices as well as provoking error states, conducting load tests, and much more.



If the client acting as the server does not update the data any longer, the client being tested is informed of this fact.

Configuring OPC Client2Client

To configure the OPC Client2Client you mainly need to create a corresponding data namespace within the local dataFEED OPC Suite application. Data items from this namespace act as data 'gateway' items between the two OPC clients – each OPC client can read and write data using the corresponding OPC mechanisms.

The OPC Client2Client namespace is configured by defining so-called Local Items. Because these local items extend the local namespace, the corresponding local items configuration section can be found within the Data Source group of the dataFEED OPC Suite Configurator.

Local items can be manually added to the local namespace using the buttons on the right side of the local items configuration page and the context sensitive menu in the namespace tree control on the local items configuration page. Additionally you can import a predefined list of local items from an external local items definition file.

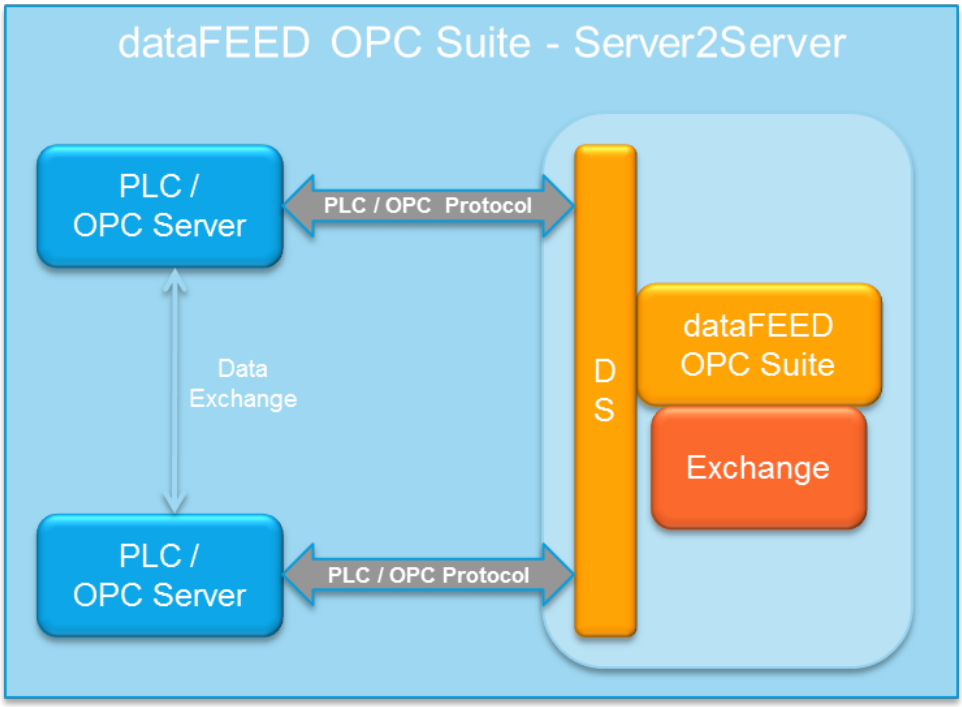
As a second configuration step, the OPC client data destination functionality of Softing dataFEED OPC Suite needs to be enabled and configured.

Note: The OPC client data destination functionality is enabled and already completely configured by default.

▼ In order to configure the OPC Client2Client, the following configuration steps should be executed:

- Define the corresponding static OPC Client2Client namespace by defining corresponding local items as a data source @ Configuration: Data Source - Local Items.
- Enable and configure the OPC client data destination @ Configuration: Data Destination - OPC Client.

3.8 OPC Server2Server



dataFEED OPC Suite OPC Server2Server allows two OPC Servers to exchange data between each other. Both OPC Servers act as a data source, introducing data items to the namespace of the local dataFEED OPC Suite application. The Exchange module of Softing dataFEED OPC Suite is responsible for actually copying data between the corresponding data items from one OPC server to the other – thereby providing the OPC Server2Server functionality.

The data copy mechanism within the Exchange module of Softing dataFEED OPC Suite is internally organized using so-called Exchange actions. Besides the corresponding source and destination data items, the following parameters needs to be specified for each Exchange action:

- An update rate with which the corresponding Exchange action should be executed.
- An execution condition specifying under what conditions the corresponding Exchange action should be executed.

© Softing Industrial Automation GmbH

- The different data item values which should be transferred from the source to the destination data item. (Possible values to be transferred are: The data value itself, its quality or its time stamp.)

Configuring OPC Server2Server

To configure the OPC Server2Server functionality you mainly need to configure the data source connections to the corresponding OPC servers. Each OPC server adds its data items to the namespace of the local dataFEED OPC Suite instance.

As a second configuration step, the corresponding Exchange actions – which are responsible for the actual data exchange between two data items – need to be created and configured. This can either be done by using the corresponding buttons and the context sensitive menu on the source and destination tree controls of the Exchange configuration page or by just dragging and dropping the corresponding source and destination items from the tree controls to the Exchange action list control.

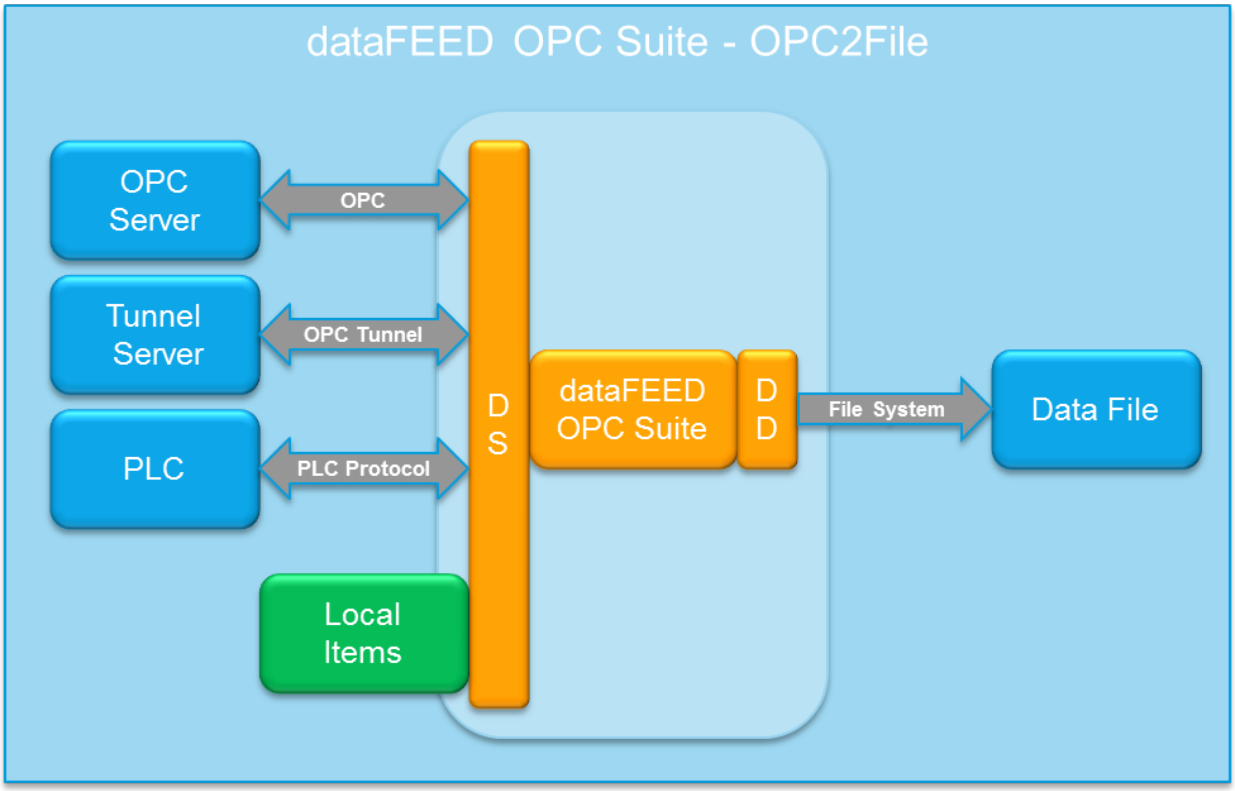
Note: The dataFEED OPC Suite Exchange functionality is not necessarily restricted to data items provided by OPC servers – it can be applied to any data item from the namespace of the corresponding local dataFEED OPC Suite application.

▼ In order to configure OPC Server2Server, the following configuration steps should be executed:

- Add a server connection to the first OPC server as a first data source @ Configuration: Data Source - OPC Server.
- Add a server connection to the second OPC server as a second data source @ Configuration: Data Source - OPC Server.
- Enable the OPC Exchange functionality and define the corresponding Exchange actions @ Configuration: Data Processing - Exchange.

Additionally the data items provided by the different data sources can also be filtered using the OPC Filter as described in Introduction: OPC Filter.

3.9 OPC2File



Softing OPC2File stores data from any OPC Data Access, OPC Tunnel or OPC XML-DA server in the corresponding text files. Different types of text files like *.txt, *.xml or *.html are supported. The data handled by Softing dataFEED OPC Suite are written to file by executing simple file text actions.

- Archiving OPC or OPC Tunnel data in files.
- Simple configuration by defining simple text actions.
- Conditional execution of these file actions is possible.
- No programming is necessary!

The data writing mechanism within the OPC2File module of Softing <%OEMPRODUCT% is internally organized using so-called file file actions. Each file action can write multiple data items at once. These data items are defined by placeholders which are replaced by the corresponding values during the run time of the local dataFEED OPC Suite application. Besides the placeholders, the following parameters need to be specified for each file action:

- An update rate with which the corresponding file action should be executed.
- An execution condition specifying under what conditions the corresponding text action should be executed.
- The corresponding data item value which should actually be written to the file. (Possible values to be written are: the data value itself, the quality or the time stamp of the corresponding data item.)

© Softing Industrial Automation GmbH

Configuring OPC2File

To configure the OPC2File functionality you mainly need to configure the data source connections to the corresponding OPC or OPC Tunnel servers providing the data items that should be written to the file.

As a second configuration step, the corresponding file and file actions – which are responsible for writing the data – need to be created and configured.

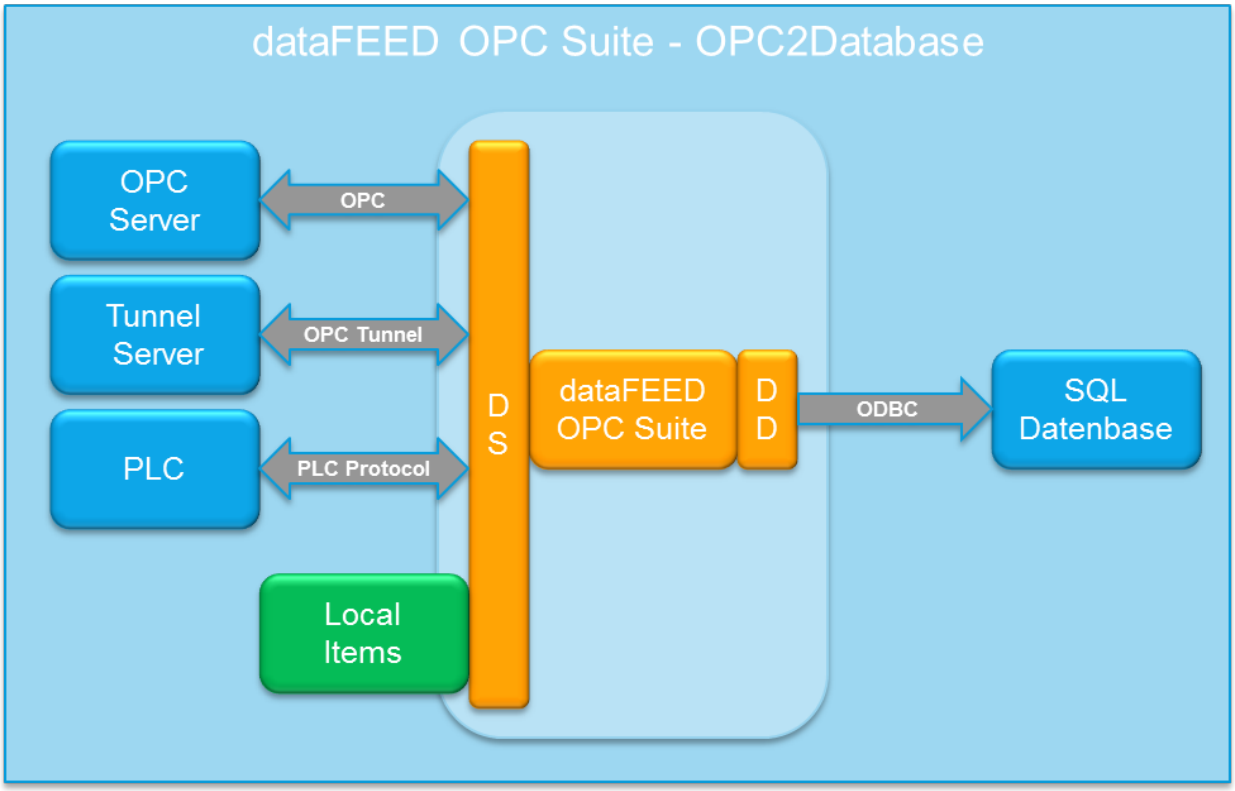
Note: The dataFEED OPC Suite OPC2File functionality is not necessarily restricted to data items provided by OPC or OPC Tunnel servers – it can be applied to any data item from the namespace of the corresponding local dataFEED OPC Suite application.

▼ In order to configure the OPC2File, the following configuration steps should be executed:

- Add one or more server connections to the corresponding OPC or OPC Tunnel server(s) as data source(s) @ Configuration: Data Source - OPC Server or Configuration: Data Source - Tunnel Server.
- Enable the OPC2File functionality @ Configuration: Data Destination - File.
- Add one or more output data files as data destinations @ File Wizard
- Define the file actions for actually writing the data to the data files @ File Action Wizard.

Additionally the data items provided by the different data sources can also be filtered using the OPC Filter as described in Introduction: OPC Filter.

3.10 OPC2Database



Softing OPC2DataBase stores data from any OPC Data Access, OPC Tunnel or OPC XML-DA server in SQL databases like Microsoft SQL Server, Oracle, MySQL, IBM DB2 and other ODBC-enabled databases. The data handled by Softing dataFEED OPC Suite are written to the database by executing simple database SQL actions.

- Archiving OPC or OPC Tunnel data in databases.
- Simple configuration by defining simple SQL actions.
- Conditional execution of these SQL actions is possible.
- No programming is necessary!

The data writing mechanism within the OPC2DataBase module of Softing <%OEMPRODUCT% is internally organized using so-called SQL database actions. Each SQL database action can write multiple data items at once. These data items are defined by placeholders which are replaced by the corresponding values during the runtime of the local dataFEED OPC Suite application. Besides the placeholders, the following parameters need to be specified for each SQL database action:

- An update rate with which the corresponding SQL database action should be executed.
- An execution condition specifying under what conditions the corresponding SQL database action should be executed.

© Softing Industrial Automation GmbH

- The corresponding data item value which should actually be written to the database. (Possible values to be written are: the data value itself, the quality or the time stamp of the corresponding data item.)

The following types of SQL statements are supported:

- INSERT INTO SQL statements.
- Stored Procedures.
- General SQL statements.

Configuring OPC2Database

To configure the OPC2Database functionality you mainly need to configure the data source connections to the corresponding OPC or OPC Tunnel servers providing the corresponding data items which should be written into the database.

As a second configuration step, the corresponding databases and database actions – which are responsible for writing the data – need to be created and configured.

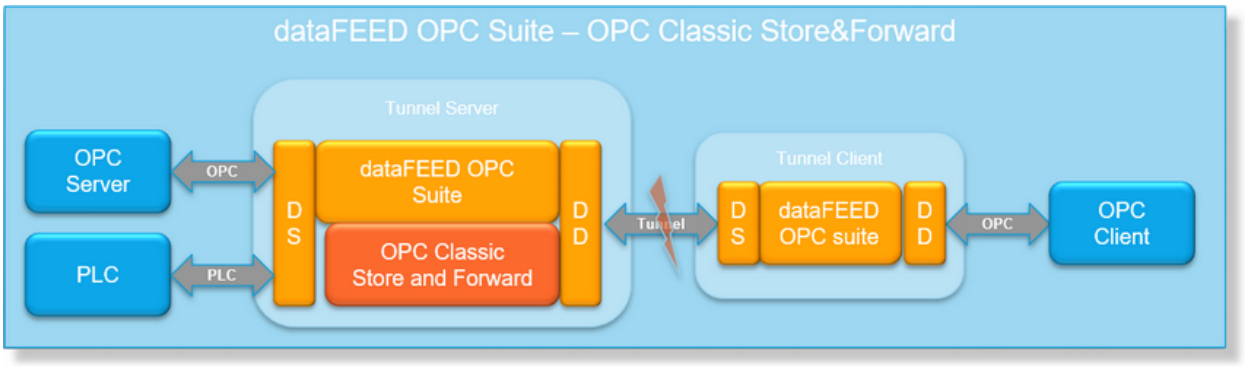
Note: The dataFEED OPC Suite OPC2Database functionality is not necessarily restricted to data items provided by OPC or OPC Tunnel servers – it can be applied to any data item from the namespace of the corresponding local dataFEED OPC Suite application.

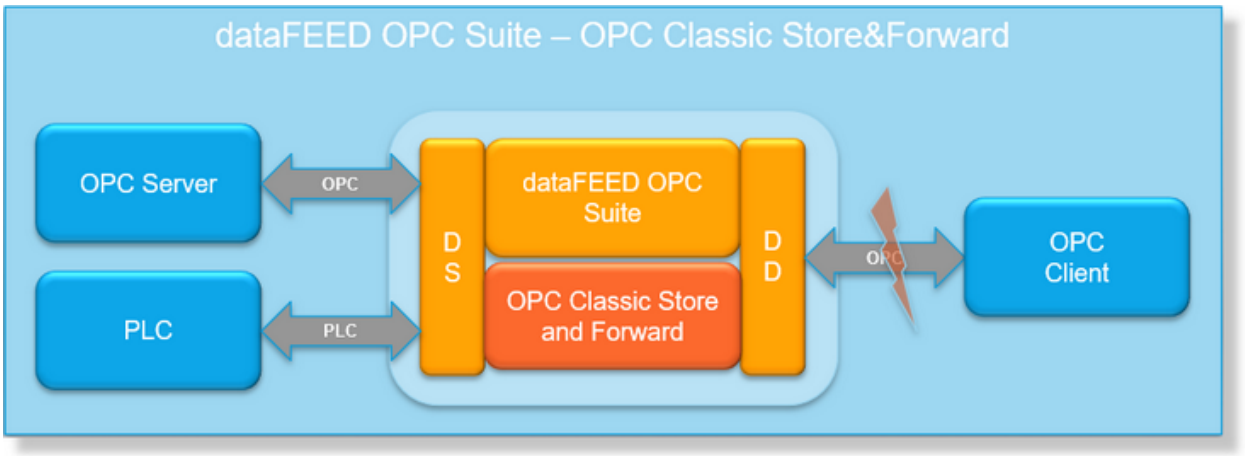
▾ In order to configure the OPC2Database, the following configuration steps should be executed:

- Add one or more server connections to the corresponding OPC or OPC Tunnel server(s) as data source(s) @ Configuration: Data Source - OPC Server or Configuration: Data Source - Tunnel Server.
- Enable the OPC2Database functionality @ Configuration: Data Destination - Database.
- Add one or more databases / ODBC data source connections as data destinations @ Database Definition Wizard
- Define the database actions for actually writing the data to the databases @ Database Action Wizard.

Additionally the data items provided by the different data sources can also be filtered using the OPC Filter as described in Introduction: OPC Filter.

3.11 OPC Classic Store And Forward





dataFEED OPC Suite OPC Classic Store And Forward provides a configurable connection monitoring for one connection to an OPC or OPC Classic Tunnel client. If the local dataFEED OPC Suite detects a connection interruption, the corresponding data values are stored in a local data storage and forwarded to the corresponding client when the connection has been reestablished.

Attention: When activating OPC Classic Store And Forward for a connection to an OPC or OPC Classic Tunnel client some restrictions and constraints apply to this monitoring client connection. However, these restrictions do not interfere with any other configured functionalities of the corresponding local dataFEED OPC Suite application.

Each local dataFEED OPC Suite application can monitor exactly one client connection. The monitoring client is either identified by the client name or it is the first client connected to the corresponding local dataFEED OPC Suite application.

If an OPC or OPC Classic Tunnel client is identified as the monitoring client, it can only use data items which are defined as monitoring data items in the local dataFEED OPC Suite application. If the monitoring client tries to access non-monitoring data items, it will always receive data values with the OPC quality *BAD - Configuration Error.* Otherwise it would not be possible to keep the timeline in the forwarding phase both between monitoring data items (historic data) and non-monitoring data items (actual data).

If OPC Classic Store and Forward is configured to identify the monitoring client as the first connected client, all other client connections will be blocked from the local dataFEED OPC Suite

© Softing Industrial Automation GmbH

application. Otherwise, in the case of a connection interruption to the real monitoring client, any other connected client could become the monitoring client and receive the stored data.

Note: It is generally not recommended to configure OPC Classic Store And Forward to identify the monitoring client as the first connected client, because if more than one client wants to connect to the local dataFEED OPC Suite application, it is generally not predictable, which client will win and become the monitoring client. However, some OPC clients do not support the configuration of the client name, so using the first connected client as the monitoring client could be the only possibility to use OPC Classic Store And Forward.

If an OPC or OPC Classic Tunnel client is identified as the monitoring client, it cannot directly read or write data items anymore - the local dataFEED OPC Suite application will answer any such attempt with an corresponding error message. Otherwise, it would not be possible to keep the timeline between the different monitoring items during the forwarding phase.

Note: After an interrupted connection to the monitoring client is reestablished, a complete snapshot of all monitoring items is initially send to the monitoring client. However this could result in some duplicate data changes (one received before and one after the connection interruption) within the monitoring client.

The actual data storage mechanism for a connection interruption to the monitoring client is organized internally using so-called Store And Forward actions. These OPC Classic Store and Forward actions are implicitly defined for each data item selected as a monitoring data item. The following parameters need to be specified for each OPC Classic Store and Forward action:

- An update rate with which the storage action should be executed.
- An execution condition specifying under what conditions the storage action should be executed.

Attention: The update rate for the storage actions (the cycle time with which the data files will be written to the local data storage during a connection interruption) and the general update rate used by the OPC client during normal operations can be configured to have different values for some special situations.

However, in most of the use cases you should ensure that both update rates have the same value in order to obtain the same number of data changes both in the normal operation phase and in the storage phase when the connection is interrupted.

Attention: The update rate for the forwarding phase can be additionally configured in the Store And Forward Wizard. If this forwarding update rate is not smaller (meaning faster) than the storage action update rate ***AND*** the normal update rate specified by the OPC client, the data forwarding will never catch up with the present time. (Even worse, the stored data will grow forever (or until the maximum storage size is reached) because the data forwarding would then be slower than the actual data acquisition.)

Configuring OPC Classic Store and Forward

To configure the OPC Store And Forward functionality you mainly need to configure the data source connections to the corresponding OPC or OPC Classic Tunnel servers providing the data items you want to monitor.

As a second configuration step, Store And Forward needs to be activated and configured in the OPC Classic Store And Forward Wizard.

In the last step, the data items to be monitored are specified. If the corresponding OPC or OPC Classic Tunnel server is running, the dataFEED OPC Suite Configurator reads the available data items and presents them for selection of the monitoring items on the last page of the OPC Classic Store and Forward Wizard or on the main configuration page of OPC Classic Store and Forward in the dataFEED OPC Suite Configurator. By checking or unchecking the individual items you can include them in, or exclude them from, the set of monitoring items for OPC Classic Store and Forward.

If the corresponding OPC or OPC Classic Tunnel server is not running, you can manually add data items to the local namespace by using the buttons on the right side of the wizard page or the context sensitive menu in the namespace tree control. You can also import a predefined list of monitoring items from an external Store And Forward definition file.

Note: The dataFEED OPC Suite OPC Classic Store And Forward functionality is not necessarily restricted to data items provided by OPC or OPC Classic Tunnel servers – it can be applied to any data item from the namespace of the corresponding local dataFEED OPC Suite application.

- To configure the OPC Classic Store And Forward, the following configuration steps should be executed:

  - Add one or more server connections to the corresponding OPC or OPC Tunnel server(s) as data source(s) @ Configuration: Data Source - OPC Server or Configuration: Data Source - OPC Classic Tunnel Server.
  - Enable the OPC Store And Forward functionality @ Configuration: Data Processing - OPC Classic Store And Forward.
  - Configure the local data storage and the data forwarding @ OPC Classic Store And Forward Wizard
  - Define the monitoring data items and specify their properties @ OPC Classic Store And Forward Wizard - Storage Items.

In addition, the data items provided by the different data sources can also be filtered using the OPC Filter as described in Introduction: OPC Filter.

© Softing Industrial Automation GmbH

4 Installation

This section of the dataFEED OPC Suite Online Help describes procedures and configurations for installing dataFEED OPC Suite.

- Supported operating systems @ Supported Operating Systems.
- Installation procedure of Softing dataFEED OPC Suite @ Installation of Softing dataFEED OPC Suite.
- Uninstallation procedure of Softing dataFEED OPC Suite @ Uninstallation of Softing dataFEED OPC Suite.
- Configure DCOM in order to enable OPC communication over the network @ DCOM Configuration.

Softing dataFEED OPC Suite V5.38
© 2024 Softing Industrial Automation GmbH

4.1 Operating Systems

Softing dataFEED OPC Suite supports the following Microsoft Windows operating systems:

- Microsoft Windows 7 (Service Pack 1 or later)
- Microsoft Windows 8.1
- Microsoft Windows Embedded 8.1 Industry
- Microsoft Windows 10
- Microsoft Windows 10 IoT Enterprise
- Microsoft Windows 11
- Microsoft Windows 11 IoT Enterprise
- Microsoft Windows Server 2008 R2 (SP1)
- Microsoft Windows Server 2012 R2
- Microsoft Windows Server 2016
- Microsoft Windows Server 2019
- Microsoft Windows Server 2022

Note: It is strongly recommended to install the latest service packs and security updates available from Microsoft for the corresponding operating system before installing Softing dataFEED OPC Suite.

Installation Prerequisites

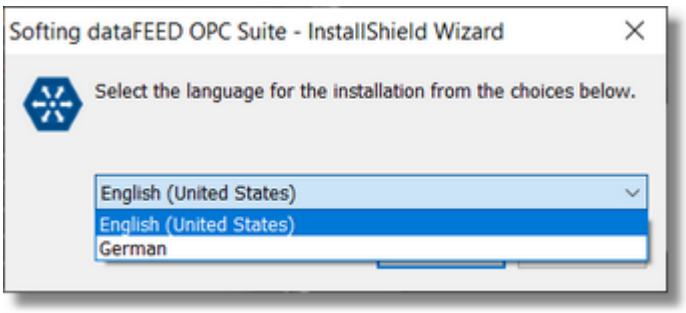
The Softing dataFEED OPC Suite requires as minimum hardware:

- Processor: Intel Pentium 2 Core (1.8 GHz)
- RAM: 4 GB
- Hard Disk Space: 2 GB.

Note: The Final need Processor and RAM depends on the used scenario.

4.2 Installation

The Softing dataFEED OPC Suite installation is done by a corresponding installation wizard. The first step in the installation wizard is to select the corresponding installation language. The chosen language will also be the initial language for the dataFEED OPC Suite Configurator, however the language can be changed at any time within the Configurator user interface:



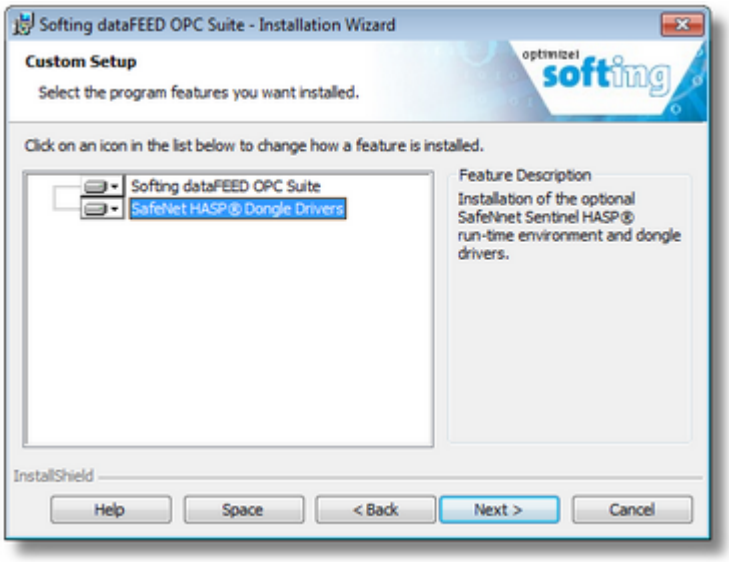
In addition to the dataFEED OPC Suite application the installer also installs the following additional applications:

- Softing License Manager.
- Softing OPC Demo Server
- Softing OPC Demo Client
- Softing OPC UA Demo Client
- OPC UA Configuration Tool
- OPC foundation LDS

Note: If you want to use the USB hardware license keys (dongles) you also need to install the SafeNet dongle driver separately. This can be done either directly by selecting the SafeNet

© Softing Industrial Automation GmbH

dongle driver in the dataFEED OPC Suite setup or by executing the corresponding link in the Softing License Manager start menu - *Softing / License Manager / SafeNet Dongle Driver Installation.*



4.3 Uninstallation

Uninstalling of Softing dataFEED OPC Suite can be done from *Windows Control Panel* in the *Programs and Functions* folder.

Note: All existing configurations of Softing dataFEED OPC Suite and the associated data files will just be unregistered but not deleted from your PC. After reinstalling dataFEED OPC Suite, they will be registered and should be available again.

Note: The License Manager, the OPC Demo Client and the OPC Demo Server will only be uninstalled if no other Softing product using those tools is installed on the PC.

Note: Even if the License Manager is uninstalled, the corresponding licenses will remain on the PC and will be available again when the License Manager is reinstalled.

4.4 DCOM Configuration

This chapter of the dataFEED OPC Suite Online Help provides a short overview of the necessary DCOM settings to allow OPC communication between networked computers, even in different domains.

What is DCOM?

DCOM, the Distributed Component Object Model of Microsoft, is an object oriented RPC system that enables remote procedure calls. It was defined by Microsoft to enable the communication of

COM based applications over a network. OPC uses the DCOM protocol for computer-to-computer communication.

Under OPC DA, the "dcomcnfg" service program is used for computer-to-computer communication between OPC clients and OPC servers.

This service program will look and behave differently depending on the operating system used (XP / Vista / 7 / 2003 Server / 2008 Server). This document refers to Windows XP.

DCOM permits only authenticated access between computers. We recommend registering the computers in the same domain and specifying a user group (e.g., "OPC Users") for OPC communication on all computers.

This chapter will provide recommendations for suitable DCOM settings.

Attention: The DCOM configuration "dcomcnfg" reaches deep into the Windows operating system. Therefore, any wrong adjustment can result in an unstable operating system.

These DCOM settings require lowering the security of the system. The security settings to be selected in order to allow the communication between distributed computers via DCOM / OPC (e.g. authentication "None"; access authorization "Everyone") result in giving all users on the network unlimited access to all data and services provided by the current computer.

In particular the opening of port #135 creates a critical security hole – this allows applications to interfere with Windows components through "remote procedure calls" via the network without any restrictions

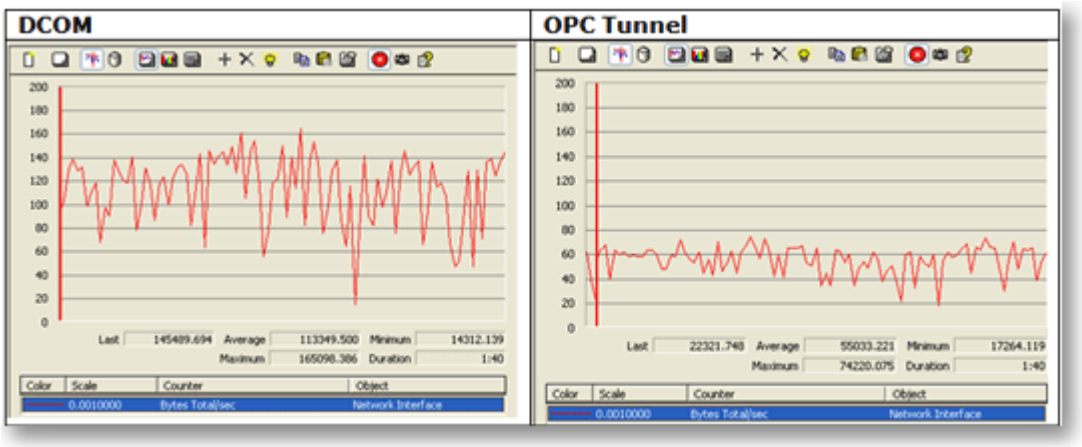
Actually this results in an open gateway for many computer viruses!

By-passing of DCOM communication by using the Softing OPC Tunnel

In order to avoid these security issues you should use the Softing OPC Tunnel protocol instead of pure OPC communication. By using the OPC Tunnel you will benefit from the following features:

- Data transfer without security holes in your system.
- No time consuming trial-and-error approach to find the appropriate DCOM settings.
- Data communication beyond firewalls.
- Immediate detection of communication breaks (with DCOM this can take several minutes).
- Increased performance due to dramatically reduced bandwidth requirements.

© Softing Industrial Automation GmbH



The above graphs present the data volumes (scale in kB) transferred between an OPC client and an OPC server when using "standard OPC" via DCOM (left) and when using the Softing OPC Tunnel (right). When communicating via the OPC Tunnel on average only 50% of the bandwidth is required, compared to the DCOM-based data transfer. (Source: OPC Training Institute).

DCOM Settings

▼ If you want to use DCOM / OPC communication, the following sections describe the necessary configuration steps:

- DCOM Configuration: System Wide Security Parameters
- DCOM Configuration: Application Specific Security Parameters
- DCOM Configuration: Activating Guest Access
- DCOM Configuration: Configuring the Windows Firewall

Note: In certain circumstances, "dcomcnfg" may issue DCOM configuration warnings. You will be asked whether you want to automatically correct discrepancies in the DCOM configuration. This is not necessary for the settings described here.

4.4.1 System Wide Security Parameters

The following preferences should be set on the client computer and the server computer.

Start "dcomcnfg"

- Log in with administrator rights

- Open the Windows start menu
- Select "Run"
- Enter "dcomcnfg"
- Click "OK"

Image 1: "Run" dialog in Microsoft Windows:

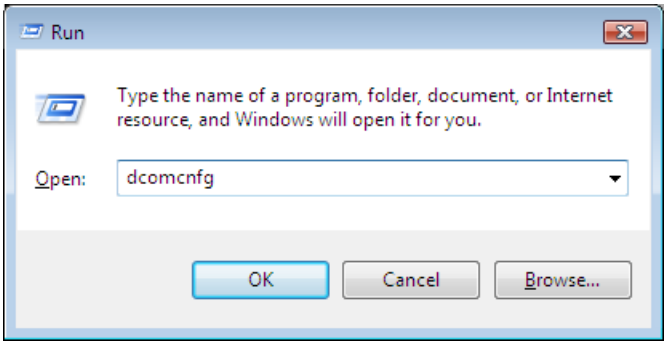
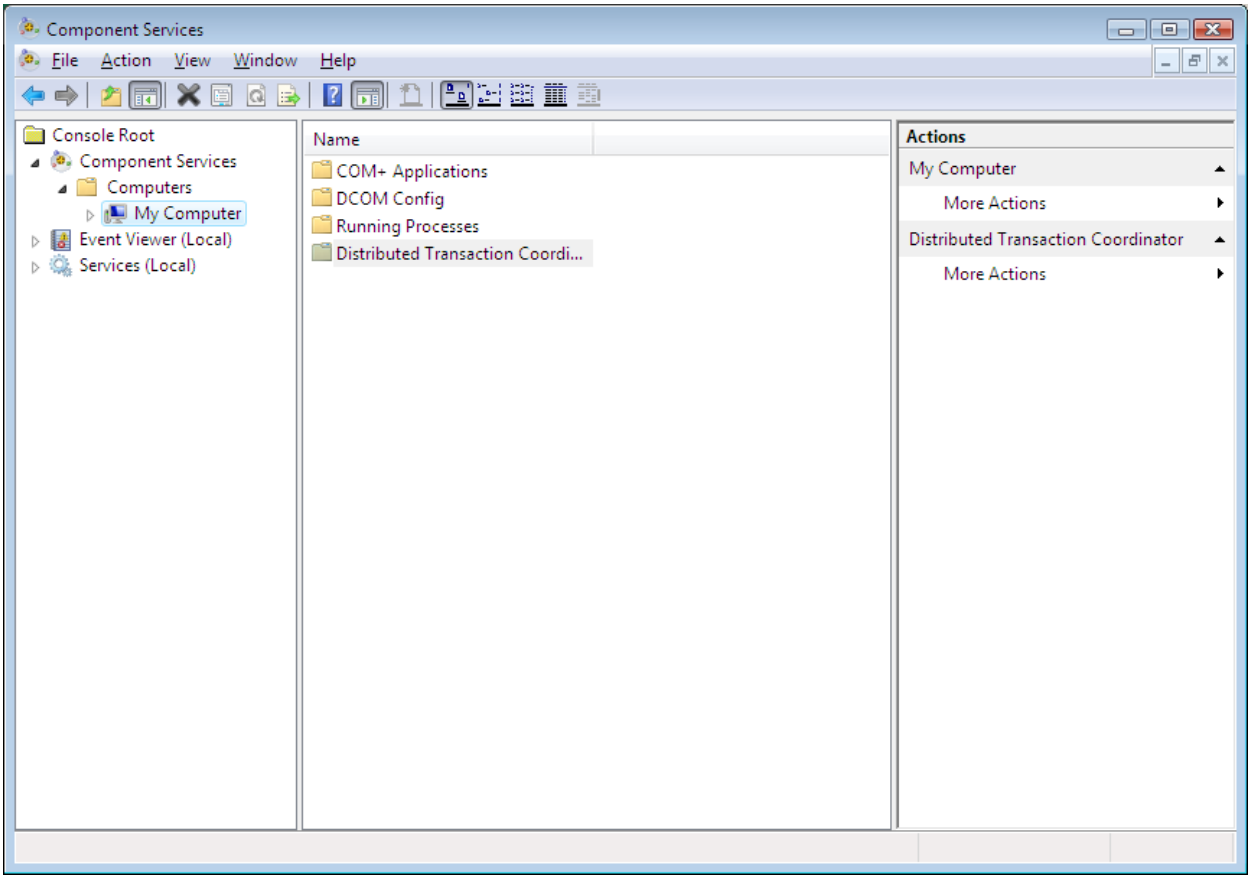


Image 2: "dcomcnfg" start page:



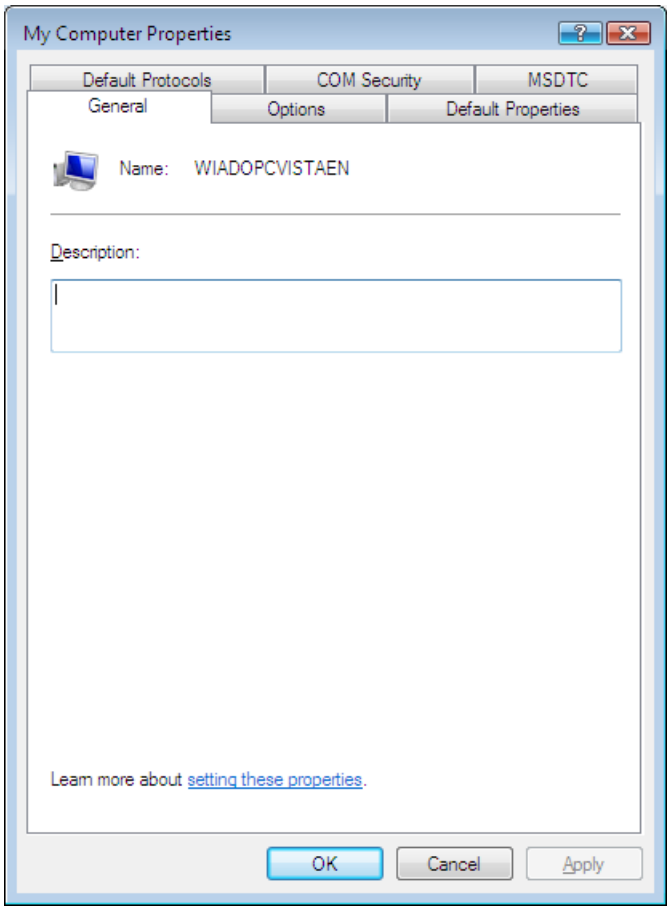
Open the "My Computer Properties" dialog

- Select "Console Root / Component Services / Computers / My Computer" in the tree view

© Softing Industrial Automation GmbH

- Open the context menu (right mouse button) -> Properties

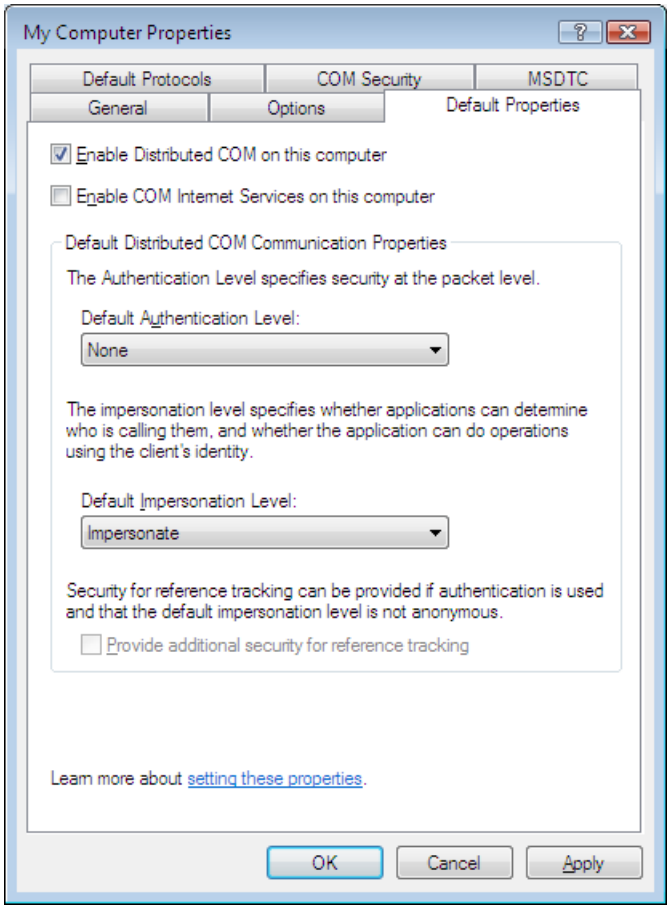
Image 3: "My Computer Properties" start page:



Set "Default Properties"

- Select the "Default Properties" tab
- Enable Distributed COM on this computer

Image 4: "Default Properties" tab:



Note: Default Authentication Level indicates when authentication should be carried out (never, when connecting, for every packet, etc.).

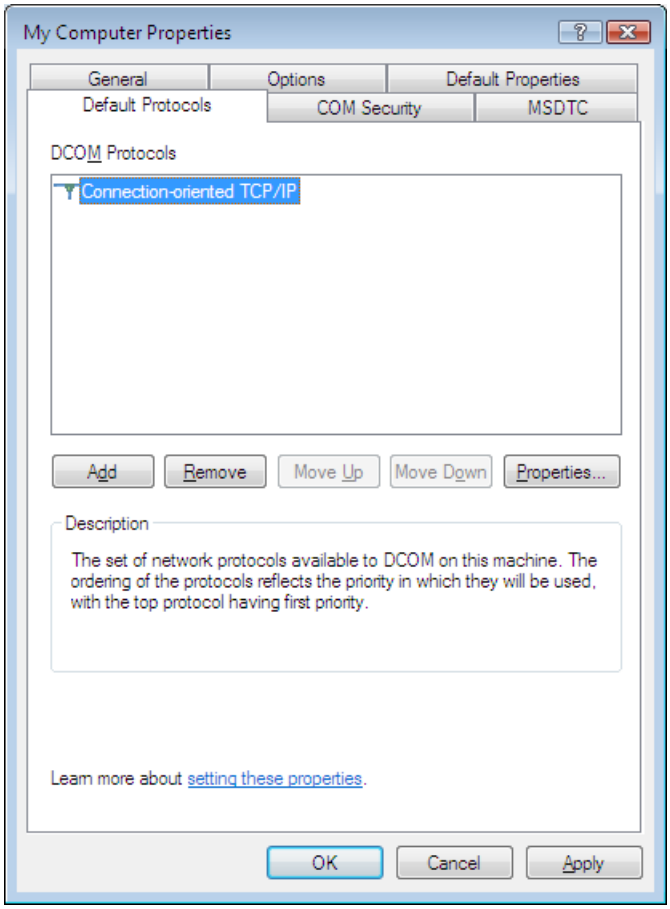
Note: Default Impersonation Level indicates whether applications can determine who is calling them and carry out operations using the client's identity.

Set "Default Protocols"

- Select the "Default Protocols" tab
- Check that "Connection-oriented TCP/IP" is in the first position

Image 5: "Default Protocols" tab:

© Softing Industrial Automation GmbH

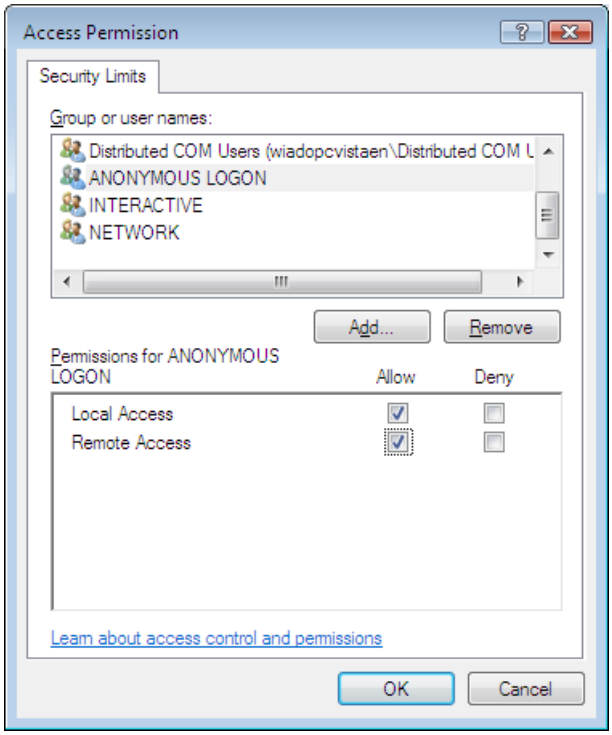


Note: Windows checks the protocols in the order in which they are displayed here. If "Connection-oriented TCP/IP" is not in the first position, there will be delays. If the time it takes to establish a connection is longer than the time allocated for the server, the server will not be able to start. A high number of DCOM protocols will lead to unnecessary delays if the connection is broken. Unnecessary entries should therefore be deleted.

Set "COM Security"

- Select the "COM Security" tab
- Click the "Access Permission: Edit Limits" button
- Add "ANONYMOUS LOGON", "Everyone", "INTERACTIVE", "NETWORK" and "SYSTEM" as user names (see note)
- Allow "Local Access" and "Remote Access" for these user names
- Click "OK"

Image 6 : Setting "Access Permission":



If a desired group or user name does not appear in the selection window, it can be added as follows:

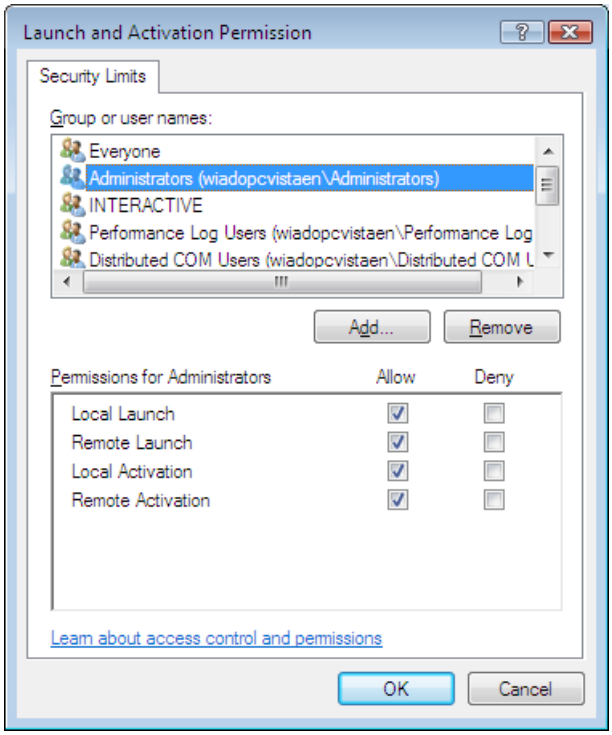
- Click "Add"
- Click "Advanced"
- Click "Find Now"
- Mark "Group or user names" (multiple selections are possible by holding down the "Ctrl" key)
- Click "OK"

Note: The "Everyone" user group contains all users (local and domain). You can also create permissions for a group (e.g., "OPC Users") and add all OPC users to this group. This group must then exist on the server computer and client computer. Windows does not allow empty passwords for DCOM communication.

- Launch and Activation Permissions: Click "Edit Limits"
- Add "Everyone", "INTERACTIVE", "NETWORK" and "SYSTEM" as user names
- Allow "Local Launch", "Remote Launch", "Local Activation" and "Remote Activation" for these users and the administrator
- Click "OK"

Image 7: Setting "Launch Permission":

© Softing Industrial Automation GmbH



- Access Permission: Click "Edit Default"
- Add "Everyone", "INTERACTIVE", "NETWORK" and "SYSTEM" as user names
- Allow "Local Access" and "Remote Access" for these users and the administrator
- Click "OK"

- Launch and Activation Permissions: Click "Edit Default"
- Add "Everyone", "INTERACTIVE", "NETWORK" and "SYSTEM" as user names
- Allow "Local Launch", "Remote Launch", "Local Activation" and "Remote Activation" for these users and the administrator
- Click "OK"

To finish, click "OK" in the "My Computer Properties" dialog to save all these changes.

4.4.2 Application Specific Security Parameters

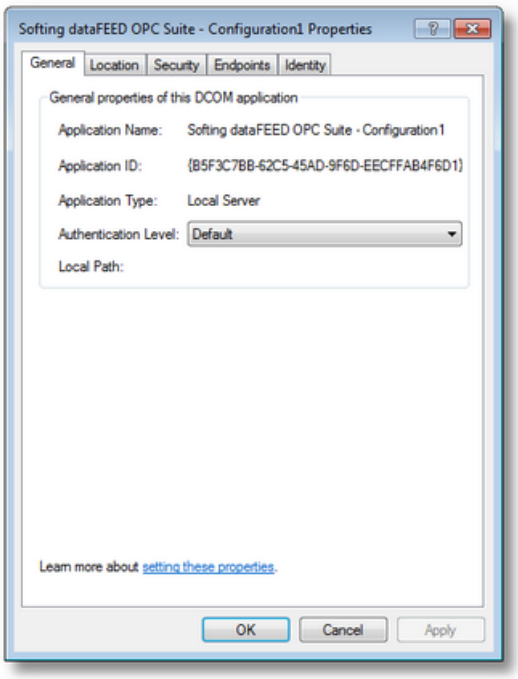
Carry out the following steps for all configured local dataFEED OPC Suite applications and for OPC enumeration service *OPCenum.exe* on your local computer.

Open the "Properties" dialog for OPCEnum and for the corresponding local dataFEED OPC Suite application

- Start "DCOM Config" (see section DCOM settings for system-wide security parameters)

- Select "Console Root / Component Services / Computers / My Computer / DCOM Config" in the tree view
- Select DCOM server (right side)
- Open context menu (right mouse button) -> Properties

Image 8 : Application-specific start page for DCOM configuration:

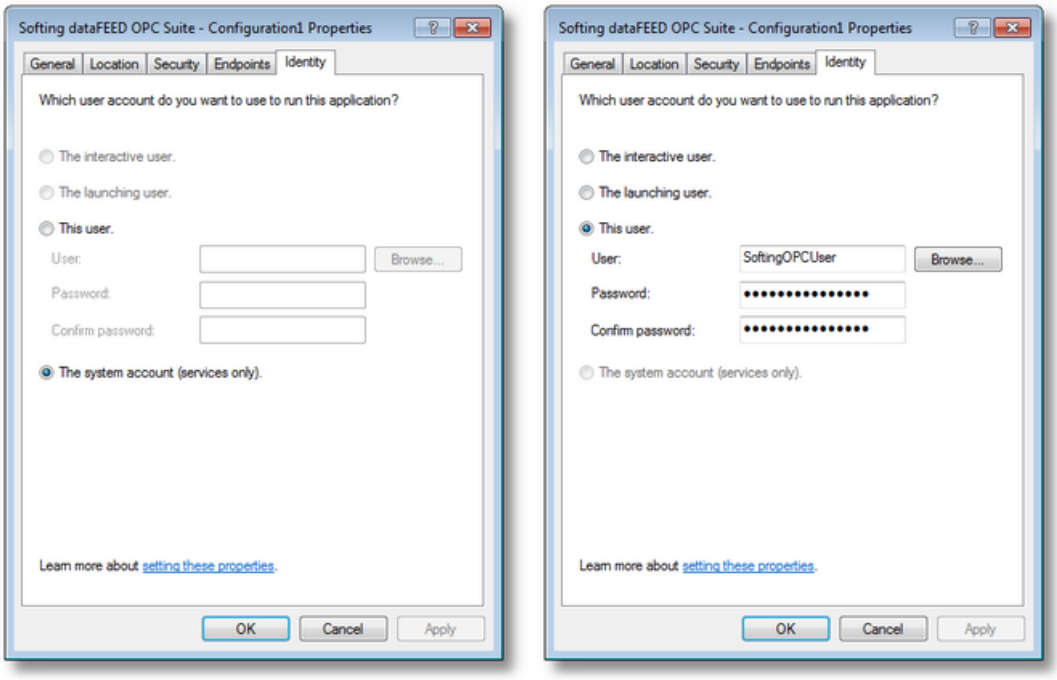


Identity

- Select the "Identity" tab

Image 9 : Setting the identity:

© Softing Industrial Automation GmbH



Note: If a local dataFEED OPC Suite application is running as a "service", the "interactive user" and "launching user" lines will be grayed out.

Note: If a local dataFEED OPC Suite application is currently configured to be executed as an application, it will be started with the SoftingOPCUser user account. You should not change this setting manually. (If you change this user, make sure that it is within the Administrator group and available both on the client and on the server side.)

Note: If a local dataFEED OPC Suite application need to be connected to a DCS server that's running on a specific user then you will need to select the "this user" option and to specify the "User" and "Password" for this specific DCS server otherwise the data subscription functionality might fail.

Set security preferences

- Select the "Security" tab
- Launch and Activation Permissions: Select "Use Default"
- Access Permissions: Select "Use Default"
- Configuration Permissions: Select "Customize" and click "Edit"
- Add "Everyone", "INTERACTIVE", "NETWORK" and "SYSTEM" as user names
- Allow "Full Control" and "Read" for these users and the administrator
- Click "OK"

Note: In order to enable the "dcomcnfg" settings, the corresponding OPC servers and OPC clients (resp. local dataFEED OPC Suite applications) must be restarted.

4.4.3 Activating Guest Access

With the standard Windows XP installation users authenticate as guests on remote computers. This means that a DCOM client will not be able to connect to a server until guest access has been activated and the Guest user has sufficient permissions to access the server.

Note: This section is only relevant to the Microsoft Windows XP operating system.

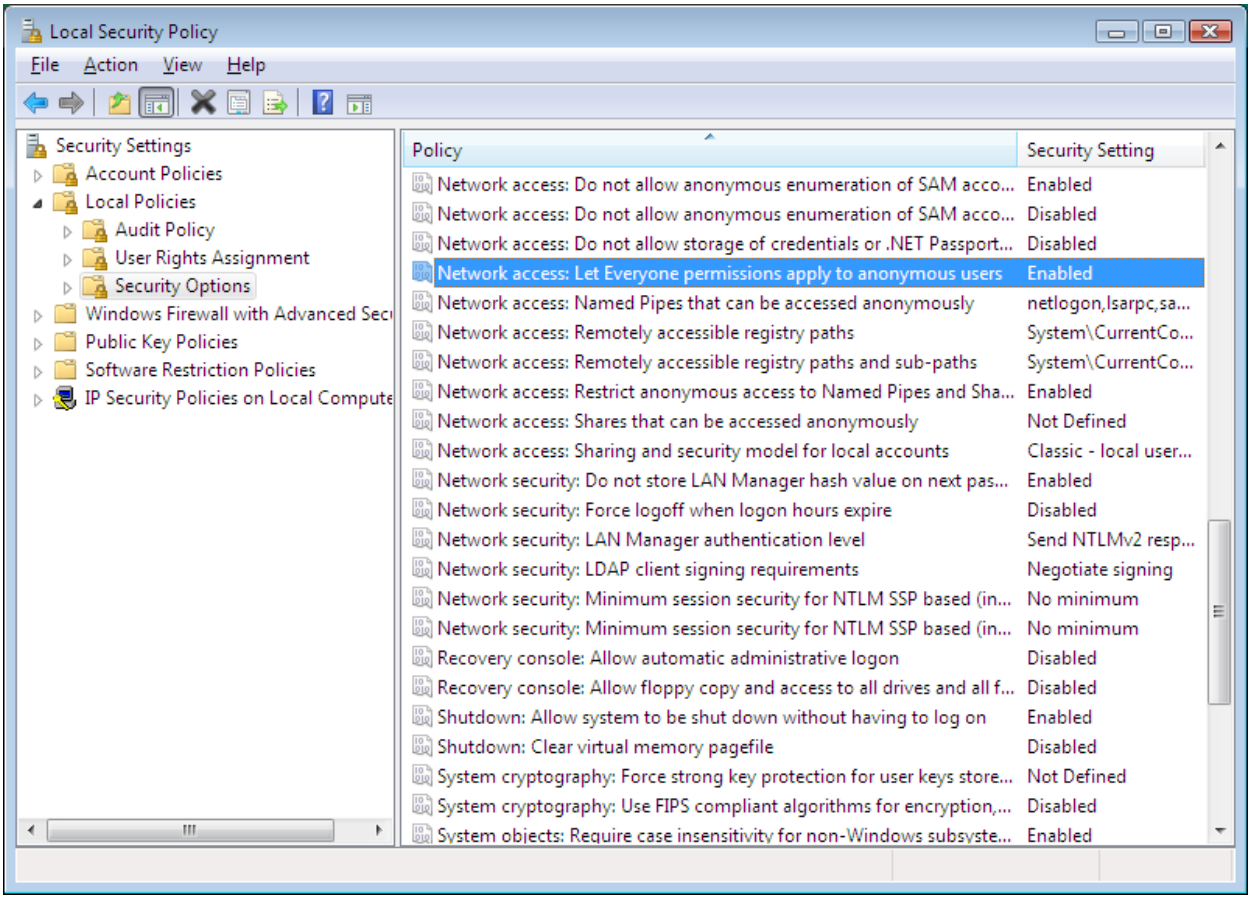
This section is only relevant to Windows XP. With the standard XP installation, users are authenticated on remote computers as guests. This means that a DCOM client will not be able to connect to a server until guest access has been activated and the guest has sufficient permissions to access the server.

The following preferences should be set on the client computer and the server computer.

- Open the Windows start menu
- Select "Settings / Control Panel"
- Select "Administrative Tools"
- Select "Local Security Policy"
- Select "Local Policies / Security Options" in the tree view
- Double-click "Network access: Sharing and security model for local accounts"
- Select "Classic - local users authenticate as themselves"
- Double-click "Network access: Let Everyone permissions apply to anonymous users"
- Select "Enabled"

Image 10: Establishing "Local Security Settings":

© Softing Industrial Automation GmbH



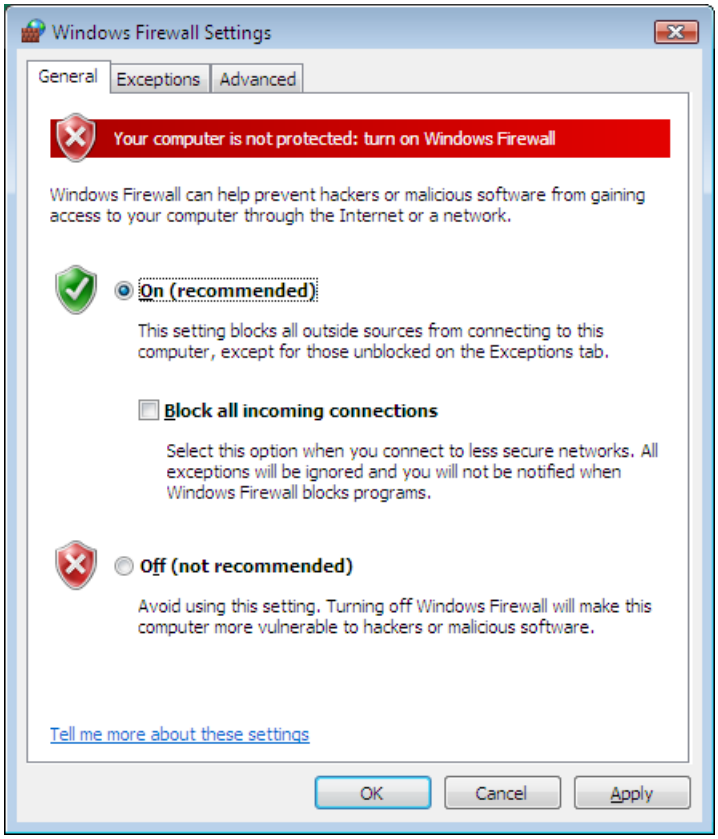
4.4.4 Configuring the Windows Firewall

This section refers to the firewall in Windows XP SP2. However, these settings generally apply to all other firewalls as well. The configuration should be carried out on the client computer and the server computer.

Start "Windows Firewall" configurator

- Open the Windows start menu
- Select "Settings / Control Panel"
- Select "Windows Firewall"

Image 11 : The Windows Firewall:



Specify "Exceptions"

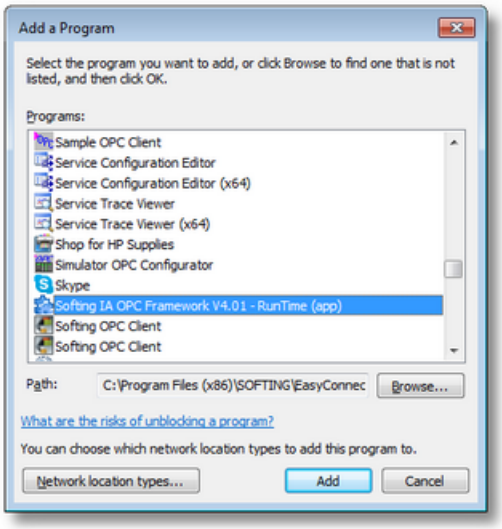
- Select the "Exceptions" tab
- Click "Add Program"
- Add the following programs:
    - All OPC clients / OPC servers
    - Microsoft Management Console
    - "OPCEnum.exe" (Windows/System32)
    - Microsoft File and Printer sharing

Note: Not all applications are shown in the "Add a Program" dialog. Other applications can be found using the "Browse" button.

Note: Only *.exe files are included in the exceptions list. For an in-process server, please select the application which calls the server.

Image 12 : Specifying exceptions:

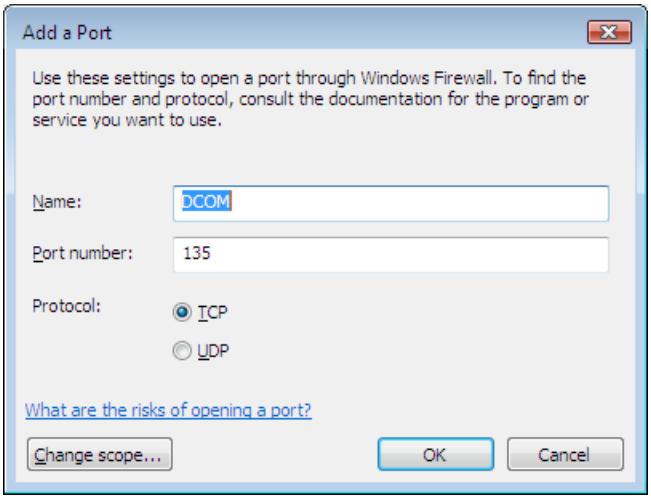
© Softing Industrial Automation GmbH



Open port

- Select the "Exceptions" tab
- Click "Add Port"
- Set the following preferences:
  - Name: "DCOM"
  - Port number: "135"
  - Select "TCP" radio button
- Click "OK"

Image 13 : Opening a port for DCOM:



Attention: Opening port 135 creates a critical security hole. This allows applications to carry out remote procedure calls over the network and thus influence Windows components. This creates a gateway for many computer viruses. Please ensure that port 135 remains closed between your company network and the Internet

© Softing Industrial Automation GmbH

5 Licensing

This section of the dataFEED OPC Suite Online Help describes the licensing of Softing dataFEED OPC Suite.

- Features and licenses of Softing dataFEED OPC Suite @ License Model of Softing dataFEED OPC Suite.
- Installation and activation of licenses @ Softing License Manager.
- Use and restrictions of the demonstration mode @ Demonstration Mode

5.1 License Model

Softing OPC products and their scalable features are licensed and managed following a uniform concept.

License Types

The following general types of licenses are available:

- Node locked licenses with a corresponding software key.
- Portable licenses with a hardware dongle for the USB port.

Note: If you want to use the USB hardware license keys (dongles) you also need to install the SafeNet dongle driver separately. This can be done either directly by selecting the SafeNet dongle driver in the dataFEED OPC Suite setup or by executing the corresponding link in the Softing License Manager start menu - *Softing / License Manager / SafeNet Dongle Driver Installation.*

Softing dataFEED OPC Suite V5.38 enables/disables functionalities based on licensed features. All supported features (available starting with version 4.00) are enabled by the following licenses:

dataFEED OPC Organizer Classic License, enables the following functionalities:

- OPC Bridge.
- OPC Server Concentrator
- OPC Optimizer
- Interface for the communication with different clients - OPC DA (V1.00, V2.05 and V3.00) and XML-DA
- Interface for the communication with different server - OPC DA (V1.00, V2.05 and V3.00) and XML-DA
- OPC Filtering
- Store and Forward add-on

Note: The number of possible server connection based on DA/AE client interface is limited to 1 DA (DA, tunnel or XML-DA) and 1 AE sever connection – the first configured connection will be establish and all others will be rejected at the initialization time.

dataFEED OPC Server Extension License, enables the following functionalities:

- Server connections (DA/AE, XML-DA or tunnel – this required also as OPC Tunnel as additional license) to external OPC Servers as data source – the following number of server connection can be activated as option into license (Please note that the corresponding licensed server connections are counted for all types of server connections together and not separately).

The following number of server connections should be available:

- 50 server connections

dataFEED OPC Server Extension Extended License, enables the following functionalities:

- Extends the maximum allowed a server connection with 50 additional server connections.

Attention: The extension with 50 server connections functionality introduced by **dataFEED OPC Server Extension Extended** license is dependent about the existence of **dataFEED OPC Server Extension** license - otherwise it will be ignored by dataFEED OPC Suite.

Attention: Both licenses (**dataFEED OPC Server Extension** and **dataFEED OPC Server Extension Extended**) are enabled by the dataFEED OPC Suite only if the **dataFEED OPC Organizer Classic** license is activated on the machine.

Attention: The overall number of actually used license items is computed at runtime over all local dataFEED OPC Suite applications currently running on the corresponding computer. The Configuration Tool, however, shows only the license items used by the actual configuration.

dataFEED OPC Tunnel License, enables the following functionalities:

- All features enabled by OPC Organizer Classic license
- OPC Tunnel DA server interfaces
- OPC Tunnel DA client interfaces (the number of tunnel server connections is limited to 1 by Base DA license or to a specific number by Server Extension license)

dataFEED OPC Organizer UA License, enables the following functionalities:

- All features enabled by OPC Organizer Classic license
- OPC UA Server interfaces

© Softing Industrial Automation GmbH

dataFEED OPC UA Tunnel License, enables the following functionalities:

- All features enabled by OPC Organizer Classic license
- OPC UA Tunnel Server and Client

- **Note:** The maximum number of allowed OPC Tunnel connections as data sources is limited to 50.

dataFEED Web Service License, enables the following functionalities:

- All features enabled by OPC Organizer Classic license
- OPC Web Service

dataFEED OPC Data Handler License, enables the following functionalities:

- All features enabled by OPC Organizer Classic license
- OPC to File add-on
- OPC to Database add-ons (SQL or NoSQL databases)
- OPC from Database add-on
- Server to Server add-on (Exchange)
- Client to Client – Storage local items (internal address space)

Note: The maximum number of items used for all running instances/configurations into add-ons is limited to 10.000 collector item.

Note: The number of server connections is limited to 1 by the **dataFEED OPC Organizer Classic** license, otherwise the **dataFEED OPC Server Extension** license is needed.

dataFEED OPC Server Siemens License, enables the following functionalities:

- All features enabled by OPC Organizer Classic license
- Siemens family PLC connections (unlimited number of device connections)

dataFEED OPC Server Siemens SINUMERIK CNC License, enables the following functionalities:

- All features enabled by OPC Organizer Classic license
- Siemens SINUMERIK CNC family PLC connections (unlimited number of device connections)

dataFEED OPC Server Rockwell License, enables the following functionalities:

- All features enabled by OPC Organizer Classic license
- Rockwell family PLC connections (unlimited number of device connections)

dataFEED OPC Server Mitsubishi License, enables the following functionalities:

- All features enabled by OPC Organizer Classic license
- Mitsubishi family PLC connections (unlimited number of device connections)

dataFEED OPC Server Modbus License, enables the following functionalities:

- All features enabled by OPC Organizer Classic license
- Modbus family PLC connections (unlimited number of device connections)

dataFEED OPC Server B&R License, enables the following functionalities:

- All features enabled by OPC Organizer Classic license
- B&R family PLC connections (unlimited number of device connections)

dataFEED OPC Suite License – Enables all available functionalities from Softing dataFEED OPC Suite product (except **OPC UA Store&Forward, Preprocessing Engine** and **OPC UA Tunnel (Client and Server)** functionalities).

dataFEED OPC Suite Extended License – Enables all available functionalities from Softing dataFEED OPC Suite product (including **OPC UA Server Store&Forward**, **Preprocessing Engine** and **OPC UA Tunnel (Client and Server)** functionalities).

Note: Maximum number of server connections allowed by this licenses (**dataFEED OPC Suite** and **dataFEED OPC Suite Extended**) is 100.

Installation and Activation of Licenses

All licenses can be installed and activated with the external Softing License Manager.

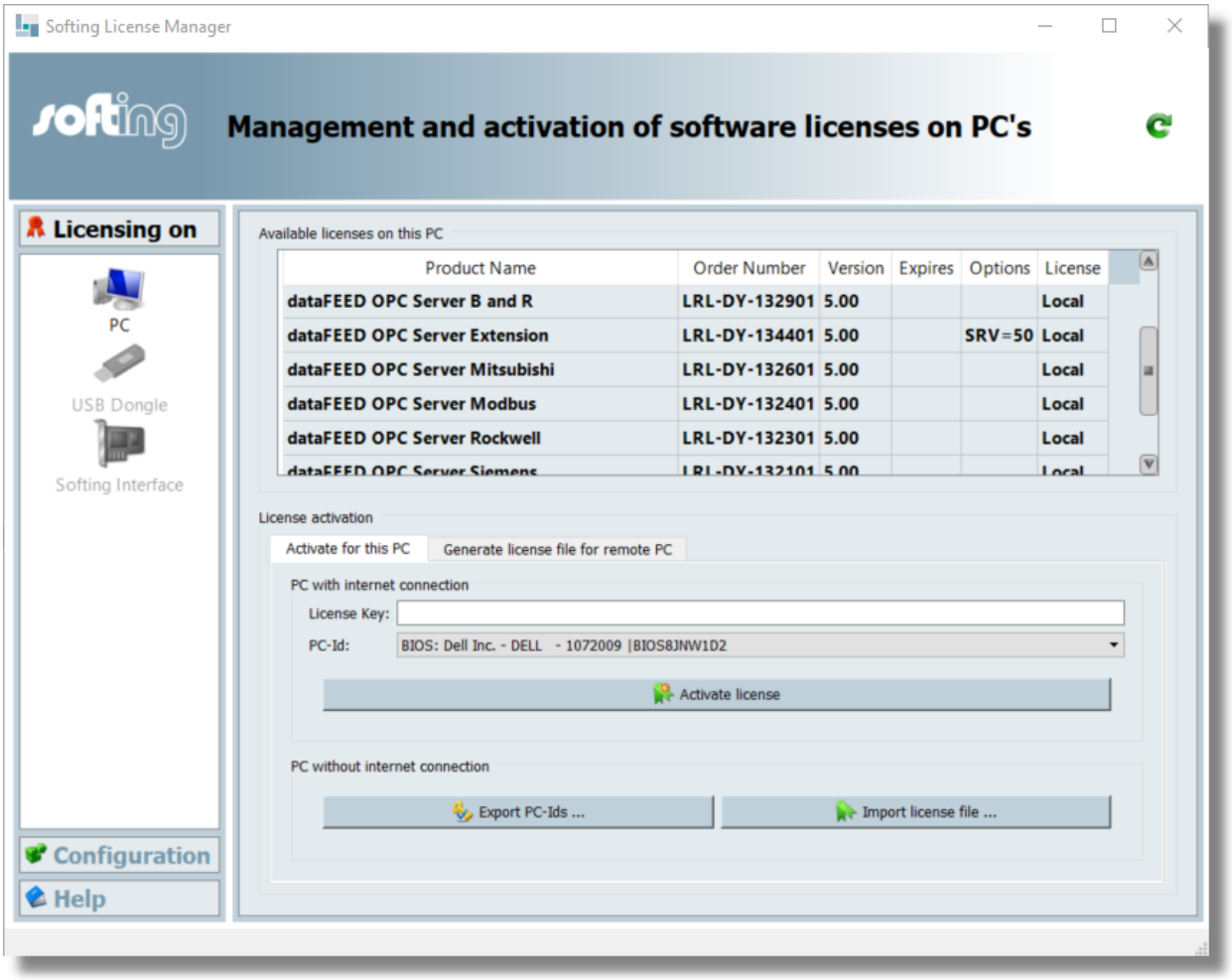
If there are no licenses installed, Softing dataFEED OPC Suite is executed in a time/function limited Demonstration Mode.

Attention: If the actual licenses are changed during the runtime of the local dataFEED OPC Suite application, the local dataFEED OPC Suite application will be stopped and the configurator and the tray application will show the yellow 'Demonstration Mode Expired' icon for the corresponding configuration.

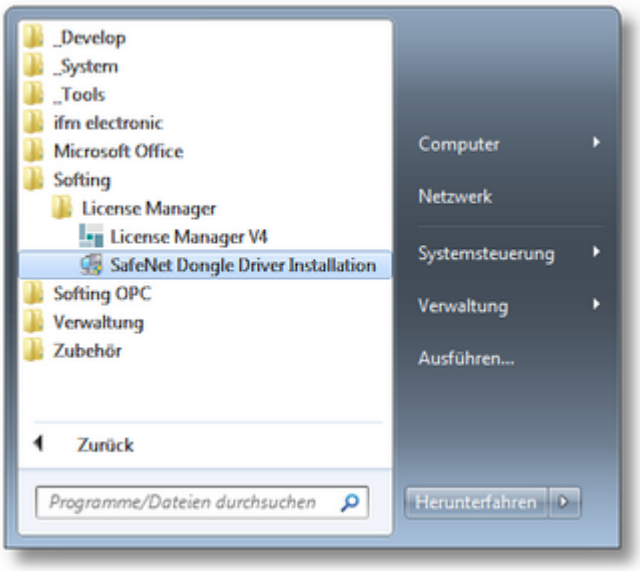
© Softing Industrial Automation GmbH

5.2 License Manager

The Softing License Manager can be used for installing and activating the licenses for the different features of Softing dataFEED OPC Suite.



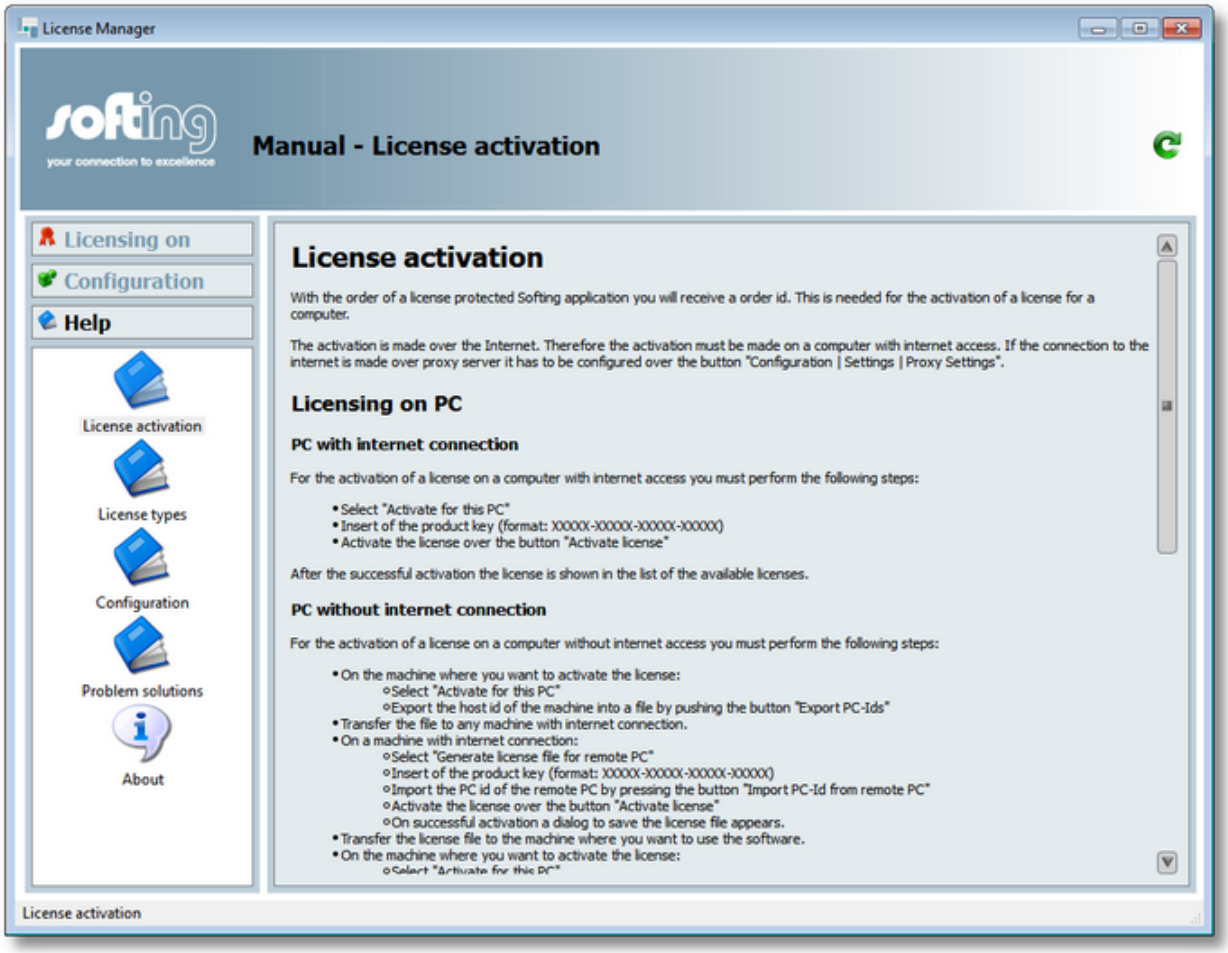
To start the License Manager select *Softing / License Manager / License Manager V4* in the Windows start menu:



Note: If you want to use the USB hardware license keys (dongles) you may need to install the SafeNet Dongle Driver separately.

A detailed description of the licensing process can be found in the online help for the Softing License Manager, in the corresponding help section:

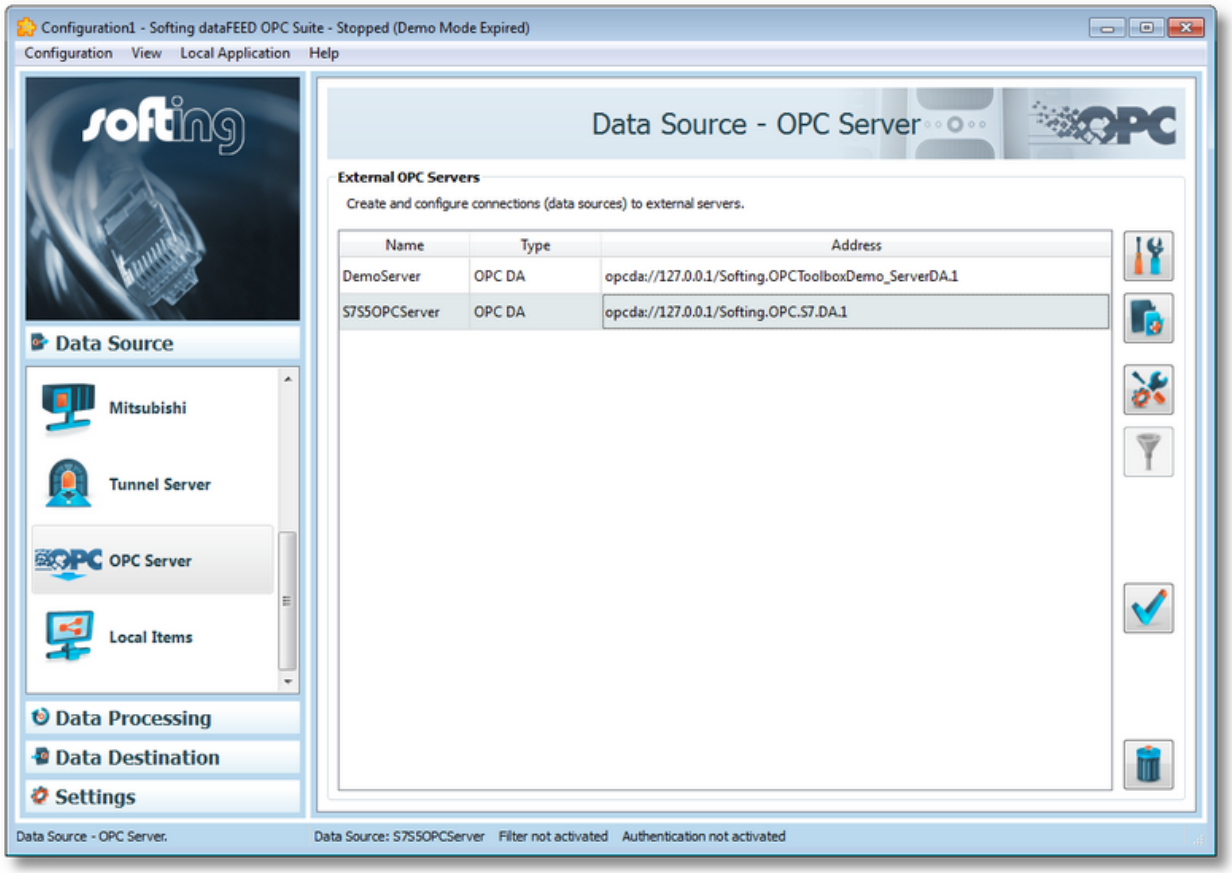
© Softing Industrial Automation GmbH



5.3 Demonstration Mode

If no license is installed on the local computer Softing dataFEED OPC Suite runs for 72 hours in a limited demonstration mode. All dataFEED OPC Suite features are completely enabled

When the demonstration period ends after 72 hours, the local dataFEED OPC Suite application stops all functionalities. The user is notified by a warning message from the dataFEED OPC Suite tray application, and both the configurator and the tray application show the corresponding yellow icon.



Restarting the local dataFEED OPC Suite application will start a new 72-hours demonstration period.

Note: The demonstration mode is only available if no licenses are installed on the corresponding computer. Installing at least one license key will disable the demonstration mode completely.

© Softing Industrial Automation GmbH

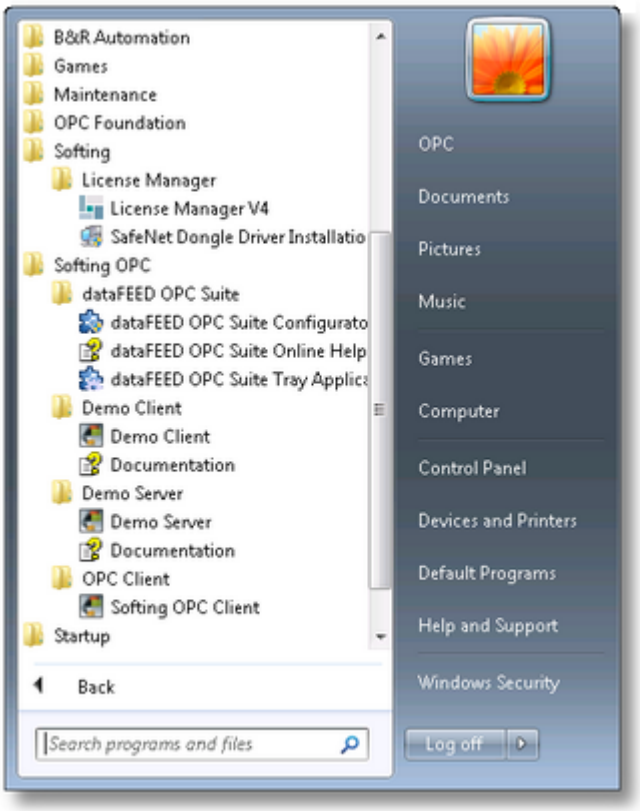
6 Operation

This section of the dataFEED OPC Suite Online Help describes some general operating procedures for Softing dataFEED OPC Suite.

- Explain the different start menu entries of dataFEED OPC Suite @ Start Menu.
- Show how to use the dataFEED OPC Suite Tray Application @ Tray Application.
- Show how to use the different web pages of the integrated web server @ Web Pages.
- Give an overview of the dataFEED OPC Suite demonstration mode @ Demonstration Mode.
- List the different OPC and Tunnel default identity settings @ Identity Settings.
- Explain the contents of the different system items of dataFEED OPC Suite @ System Items.
- Show how OPC Tunnel clients can find Tunnel servers in the network @ OPC Tunnel Discovery Service.

6.1 Start Menu

After installing Softing dataFEED OPC Suite the following entries are created in the Windows start menu:



Start Menu Shortcut	Description
Softing OPC / dataFEED OPC Suite / dataFEED OPC Suite Configurator	Starts the dataFEED OPC Suite Configurator.
Softing OPC / dataFEED OPC Suite / dataFEED OPC Suite Online Help	Opens this help file.
Softing OPC / dataFEED OPC Suite / dataFEED OPC Suite Tray Application	Starts the dataFEED OPC Suite tray application.
Softing OPC / Demo Client / Demo Client	Starts the Softing OPC Demonstration Client.
Softing OPC / Demo Client / Documentation	Opens the online help for the Softing OPC Demonstration Client.

© Softing Industrial Automation GmbH

Softing OPC / Demo Server / Demo Server	Starts the Softing OPC Demonstration Server.
Softing OPC / Demo Server / Documentation	Opens the online help for the Softing OPC Demonstration Server.
Softing OPC / OPC Client / Softing OPC Client	Starts the Softing OPC Demonstration UA Client.
Softing / License Manager / License Manager V4	Starts the Softing License Manager
Softing / License Manager / SafeNet Dongle Driver Installation	Installs the SafeNet dongle drivers

6.2 Tray Application

The dataFEED OPC Suite Tray Application gives you an easy way to start, stop and configure the local dataFEED OPC Suite applications and their corresponding configurations.

dataFEED OPC Suite Tray Application

The menu of the dataFEED OPC Suite Tray Application is shown when you right-click on the icon of the tray application in the notification area of the Windows task bar.



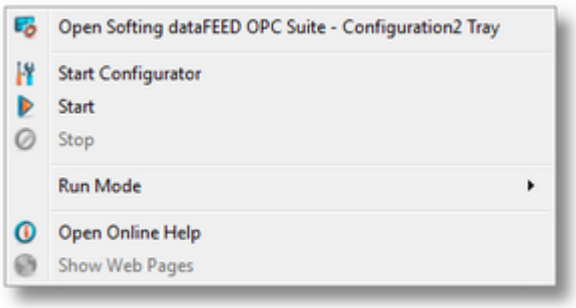
The color of the icon of the dataFEED OPC Suite Tray Application represents the current running state of the local dataFEED OPC Suite application:

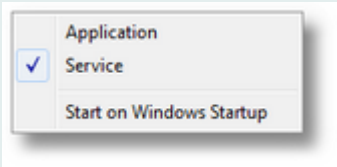
Icon Color	State	Description
Red	Stopped	The local dataFEED OPC Suite application is not running.

		
Green 	Running (or demonstration mode)	The local dataFEED OPC Suite application is running with all the required licenses, or it is running in the 72 hours limited demonstration mode. **Note:** The green icon can also mean that the corresponding local dataFEED OPC Suite application is running in the demonstration mode.
Yellow 	Stopped (demonstration mode expired or licenses changed)	The local dataFEED OPC Suite application has been stopped because the demo mode expired or because licenses have been changed during the run time of the local dataFEED OPC Suite application. **Note:** The yellow icon (demonstration mode expired) is also shown if the current licenses have been changed during the run time of the local dataFEED OPC Suite application.
Green/Yellow 	Running (or demonstration mode) & initializing UA Server functionality.	The local dataFEED OPC Suite application is initialing the UA Server functionality (the initialization of the address is a time consuming task depending of the size of address space and the speed of the connection to the devices/servers). The DA/AE functionality is ready and client connections are allowed at this time. **Note:** The green/yellow icon can also mean that the corresponding local dataFEED OPC Suite application is running in the demonstration mode.
Green/Red 	Running (or demonstration mode) without UA Server functionality.	The local dataFEED OPC Suite application is running without UA Server functionality. The UA Server was not started due the reason that it can not be initialized (certificates not match, access rights are not allowed or others). The DA/AE functionality is ready and client connections are allowed at this time. **Note:** The green/red icon can also mean that the corresponding local dataFEED OPC Suite application is running in the demonstration mode.

The organization of the dataFEED OPC Suite Tray Application menu depends on the number of configured local dataFEED OPC Suite applications. If only one configuration is present, the menu has the following format:

© Softing Industrial Automation GmbH



Menu Entry	Description
Open Softing dataFEED OPC Suite - <Configuration> Tray	Opens the tray application dialog for the local dataFEED OPC Suite application. (See Tray Application Dialog for a detailed description.)
Start Configurator	Starts the Softing dataFEED OPC Suite Configurator for the local dataFEED OPC Suite application.
Start	Starts (or restarts) the local dataFEED OPC Suite application.
Stop	Stops the local dataFEED OPC Suite application.
Run Mode	Opens a submenu where the run mode of the local dataFEED OPC Suite application can be configured: 
Run Mode / Application	The local dataFEED OPC Suite application is executed as normal Windows application.
Run Mode / Service	The local dataFEED OPC Suite application is registered in the Windows Service Manager and executed as a Windows service.

Run Mode / Start on Windows Startup	Automatically starts the local dataFEED OPC Suite application when Windows (resp. the corresponding computer) is starting up.
Open Online Help	Opens this help file.
Show Web Pages	Shows the web pages of the local dataFEED OPC Suite application in the default web browser.
Exit Tray Application	Exits the tray application completely. It can be restarted from the Windows Start menu by selecting *Softing / dataFEED OPC Suite / dataFEED OPC Suite Tray Application*

Note: Starting, stopping and changing the Run Mode of a local dataFEED OPC Suite application requires administrator rights.

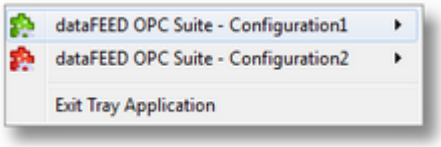
On Windows XP or on Windows Vista and Windows 7 with disabled Windows User Access Control (UAC), the corresponding menu entries will not be enabled if the current user does not have administrator rights.

On Windows Vista, XP, 7 Server 2003, Server 2008 and Windows 8 with activated Windows User Access Control, the corresponding Windows Elevation Dialog is shown asking the user for administrator credentials or for an acknowledgment of the planned operation.

Note: The Run Mode can only be changed if the corresponding local dataFEED OPC Suite application is not running.

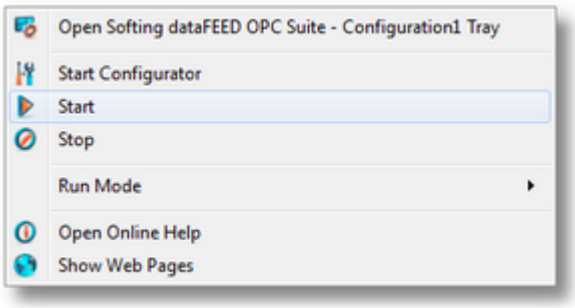
Note: The web pages can only the shown if the corresponding local dataFEED OPC Suite application is currently running.

If there is more than one configured local dataFEED OPC Suite application on the current computer, the dataFEED OPC Suite Tray Application menu shows an entry for each configured local dataFEED OPC Suite application. The icons again indicate the current running states of the individual local dataFEED OPC Suite applications:



The different functionalities for the corresponding local dataFEED OPC Suite applications are now provided in a submenu:

© Softing Industrial Automation GmbH

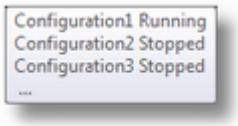


The color of the icon of the dataFEED OPC Suite Tray Application is now computed as follows:

Icon Color	Description
Red 	At least one of the currently configured local dataFEED OPC Suite applications is not running.
Green 	All currently configured local dataFEED OPC Suite applications are running with all the required licenses, or they are currently running in the limited demonstration mode.
Yellow 	At least one of the currently configured local dataFEED OPC Suite applications has been stopped because a demo mode expired or because licenses have been changed during the run time of the associated local dataFEED OPC Suite application.
Green/Yellow 	At least one of the currently configured local dataFEED OPC Suite applications is running and initializing its UA Server functionality.
Green/Red 	At least one of the currently configured local dataFEED OPC Suite applications is running without its UA Server functionality..

Configuration Status

The dataFEED OPC Suite Tray Application will display a tool tip message containing the names and states of up to three configurations (first three configurations, sorted in alphabetical order):

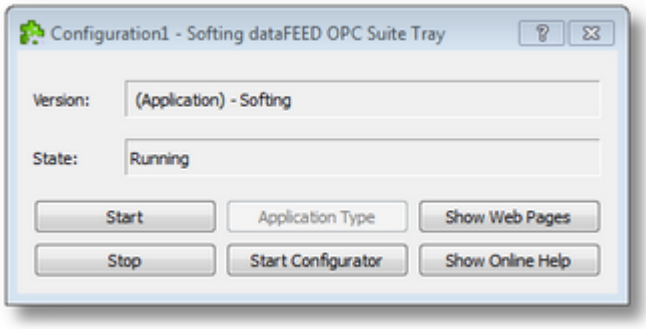


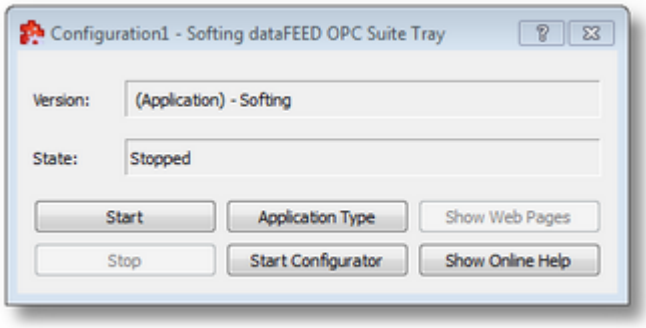
dataFEED OPC Suite Tray Application Dialog

The dataFEED OPC Suite Tray Application Dialog displays the configuration name in the title bar and contains information about the local dataFEED OPC Suite application:

- Complete application version (to be used in support inquiries).
- The current run mode and the OEM version.
- The current state.

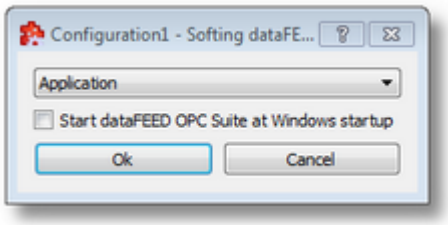
It also features buttons to access the most important functions described above in the dataFEED OPC Suite Tray Application section.





The *Application Type* button opens another dialog that allows selecting the run mode of the local dataFEED OPC Suite application. Here, you can also choose whether the application should start on Windows startup. The button is only available if the corresponding local dataFEED OPC Suite application is not running.

© Softing Industrial Automation GmbH



Note: Starting, stopping and changing the Run Mode of a local dataFEED OPC Suite application requires administrator rights.

On Windows XP or on Windows Vista and Windows 7 with disabled Windows User Access Control (UAC), the corresponding menu entries will not be enabled if the current user does not have administrator rights.

On Windows Vista and Windows 7 with activated Windows User Access Control, the corresponding Windows Elevation Dialog is shown asking the user for administrator credentials or for an acknowledgment of the planned operation.

Note: The Run Mode can only be changed if the corresponding local dataFEED OPC Suite application is not running.

Note: The web pages can only the shown if the corresponding local dataFEED OPC Suite application is currently running.

6.3 Web Pages

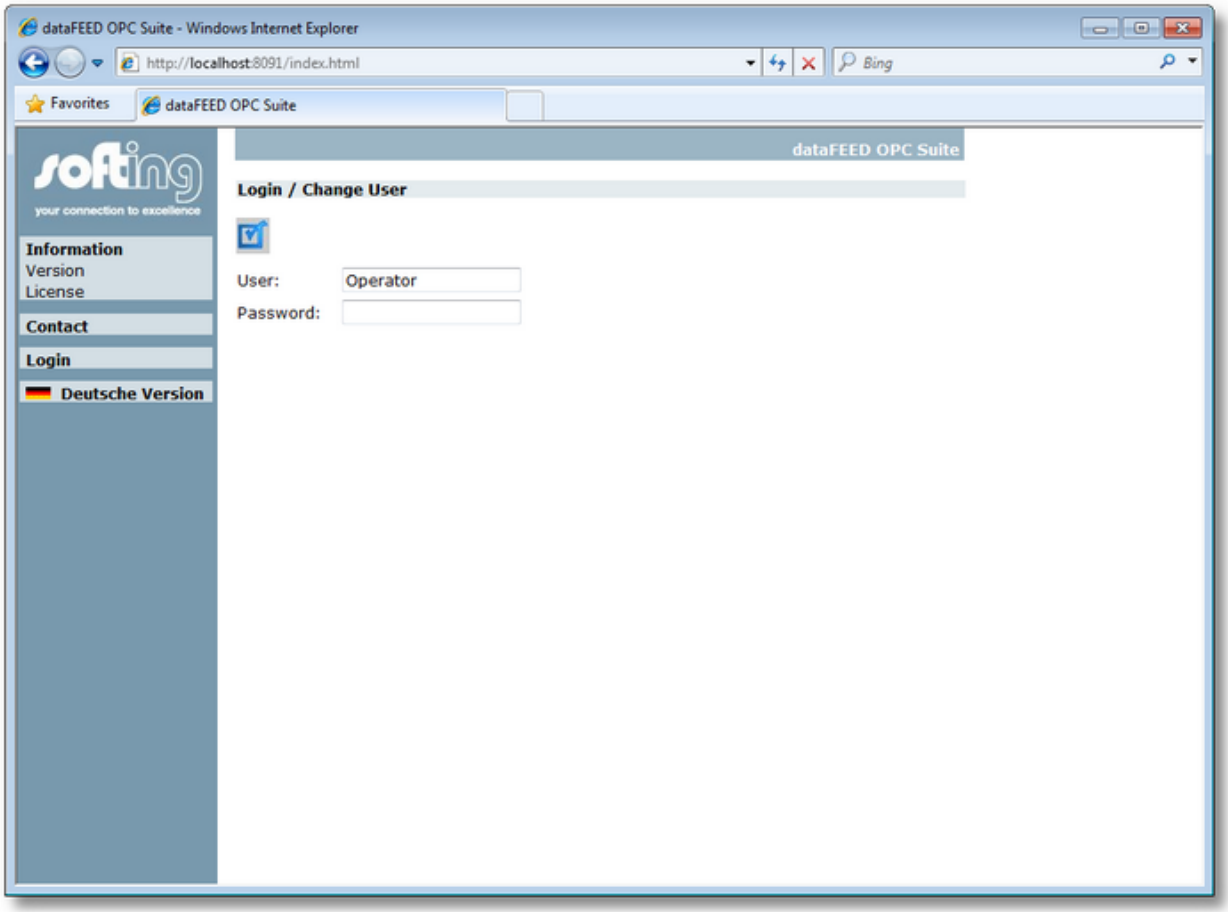
The web pages of the local dataFEED OPC Suite application provide run time information about the actual licenses and license items, version information, diagnostics information and contact data for requesting technical support for Softing dataFEED OPC Suite.

Attention: When the 72-hours demonstration mode expires, the local dataFEED OPC Suite application is generally stopped. As a result, the corresponding web pages will not be available anymore, either.

The available web pages depend on the provided user credentials. The following default passwords are defined for each local dataFEED OPC Suite application and can be changed under Configuration: Settings - Web Server.

User	Password
Operator	op
Administrator	ad

The user can be changed either on the *Login* or *Change User* web page:



Press the button to apply the changes.

Depending on the provided user credentials, the following web pages are available:

User	Available Web Pages
<none>	• Version and license information. • Contact data and login web page. See Information Web Pages for details.
Operator	Additionally: • Diagnostics information (process values, servers, clients, etc.). • Current configuration and trace outputs. See Diagnostics Web Pages for details.
Administrator	Additionally: • Possibility to change the operator and administrator passwords.

© Softing Industrial Automation GmbH

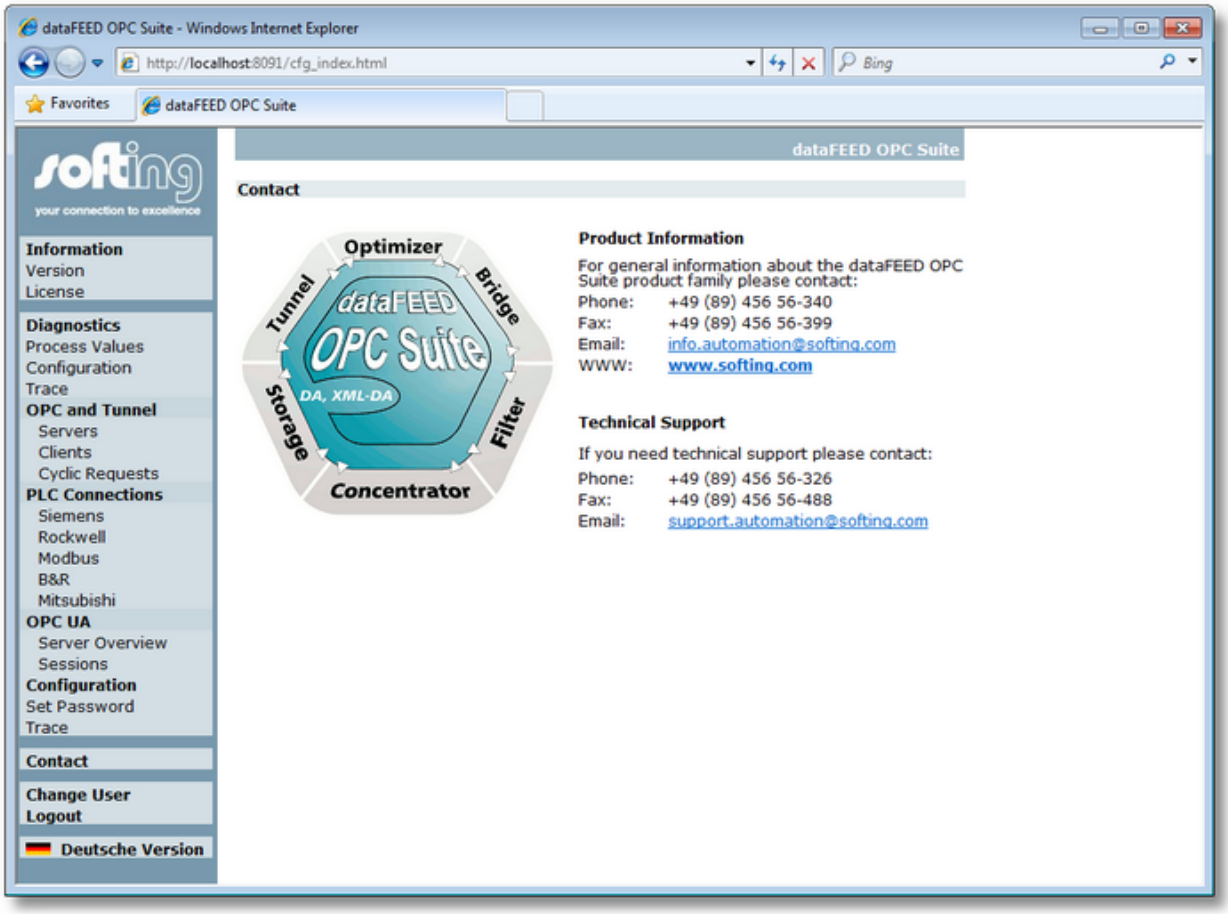
	• Possibility to change the trace output settings.
	Note: If the operator or administrator password or the trace output settings are changed using the web pages, the corresponding information in the Configurator is not updated automatically. In order to avoid inconsistencies, the corresponding configuration must be manually reloaded in the Configurator.
	See Configuration Web Pages for details.

6.3.1 Information

The following information web pages are available also without providing operator or administrator user credentials.

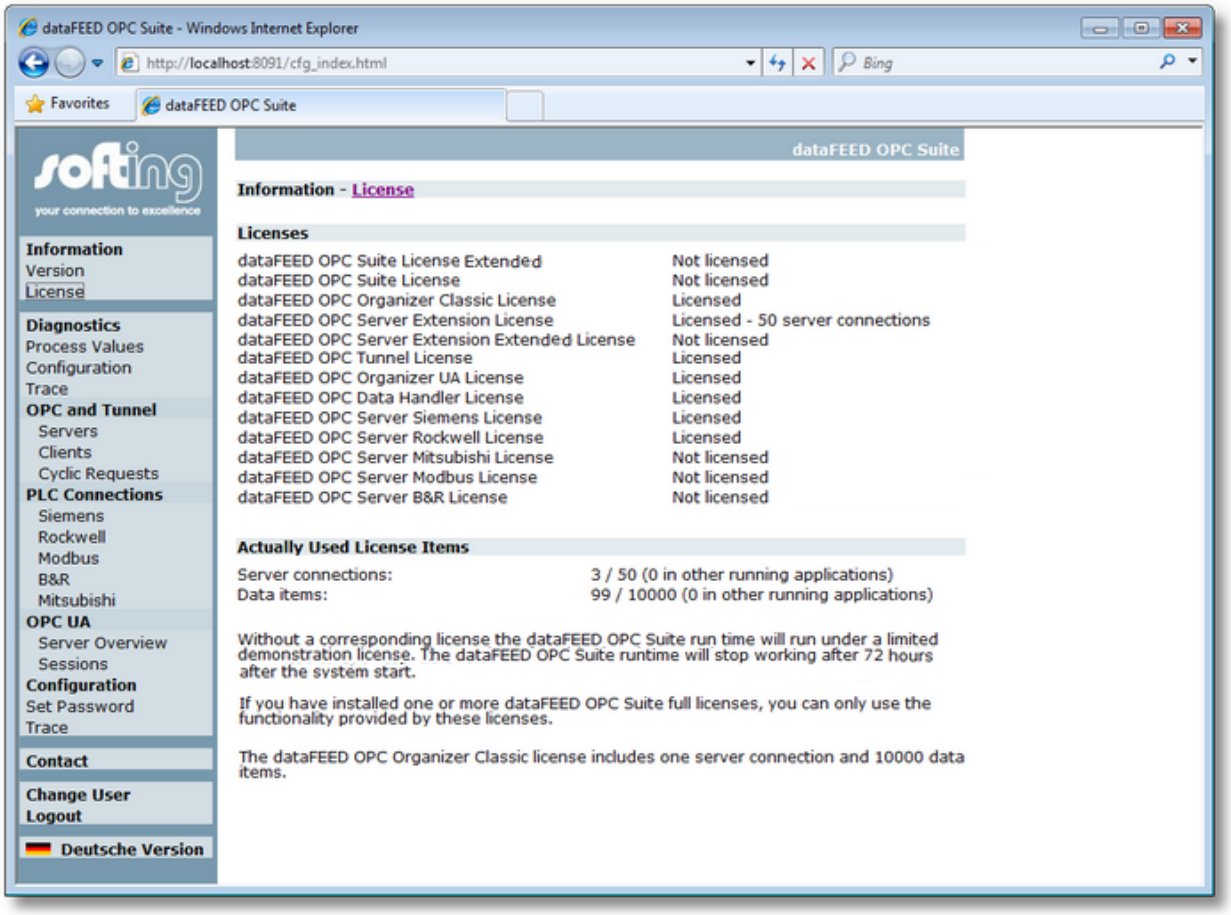
Contact Web Page

The *Contact* web page of dataFEED OPC Suite provides general contact data for Softing dataFEED OPC Suite.



License Web Page

Information regarding the installed licenses is displayed on the *License* page. This page also shows the license items currently used for all running local dataFEED OPC Suite applications.



Attention: License items are always counted for all the currently running local dataFEED OPC Suite applications. However, the dataFEED OPC Suite Configurator only shows the license items used for the current configuration.

6.3.2 Diagnostics

- The following diagnostics web pages are available for the local dataFEED OPC Suite applications:

- Read and write data item values @ Diagnostics Web Pages: Process Values.
- Display the currently connected OPC and Tunnel clients @ Diagnostics Web Pages: Clients.

© Softing Industrial Automation GmbH

- Review the current configuration for the local dataFEED OPC Suite application @ Diagnostics Web Pages: Configuration.
- Display the current states of the OPC and Tunnel server connections @ Diagnostics Web Pages: Servers.
- Display the currently active cyclic requests @ Diagnostics Web Pages: Cyclic Requests.
- Display the OPC UA Server Information @ OPC UA Server Information

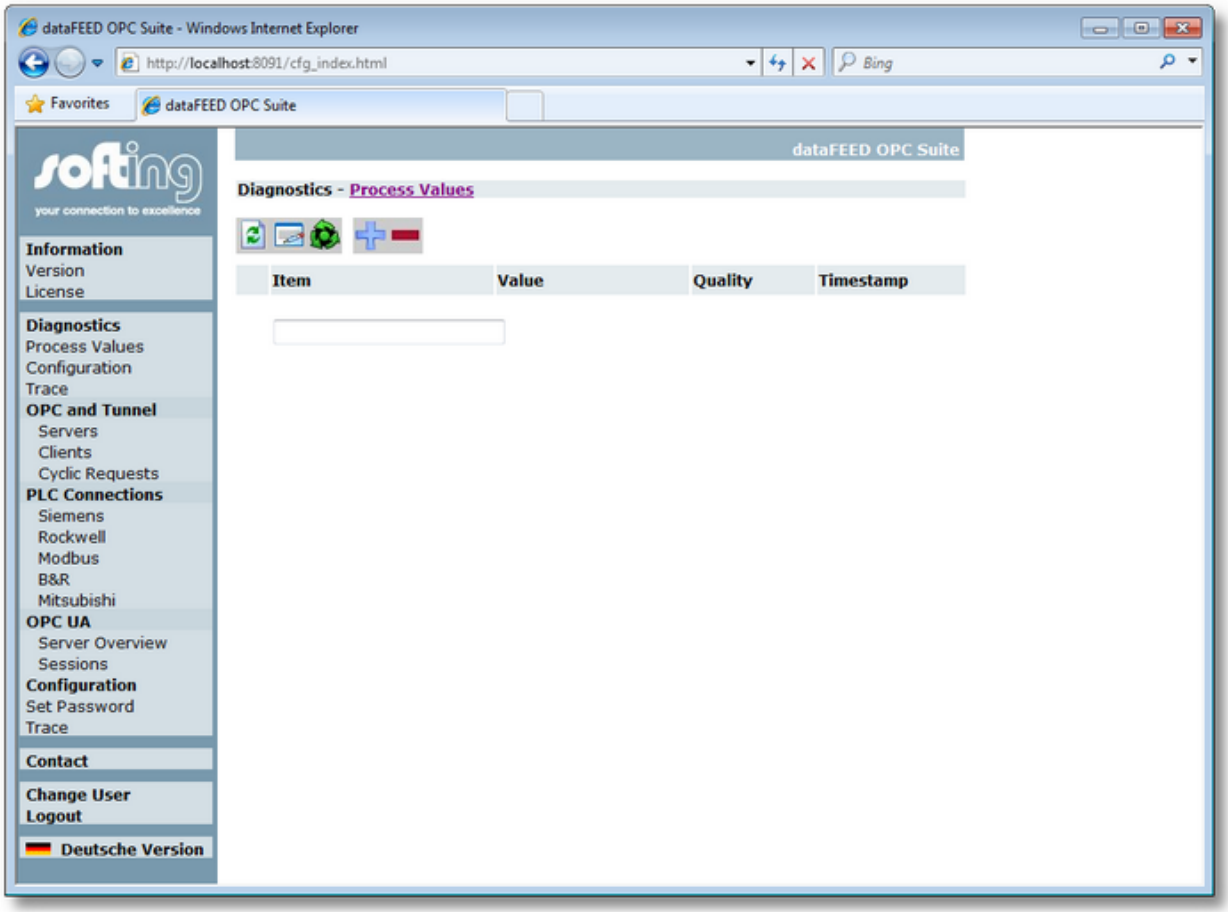
Attention: To access these web pages, you must be logged in either as operator or as administrator.

6.3.2.1 Process Values

The web interface of Softing dataFEED OPC Suite allows reading and writing real time data item values of the local dataFEED OPC Suite application.

Note: To access this web page, you must be logged in either as operator or as administrator.

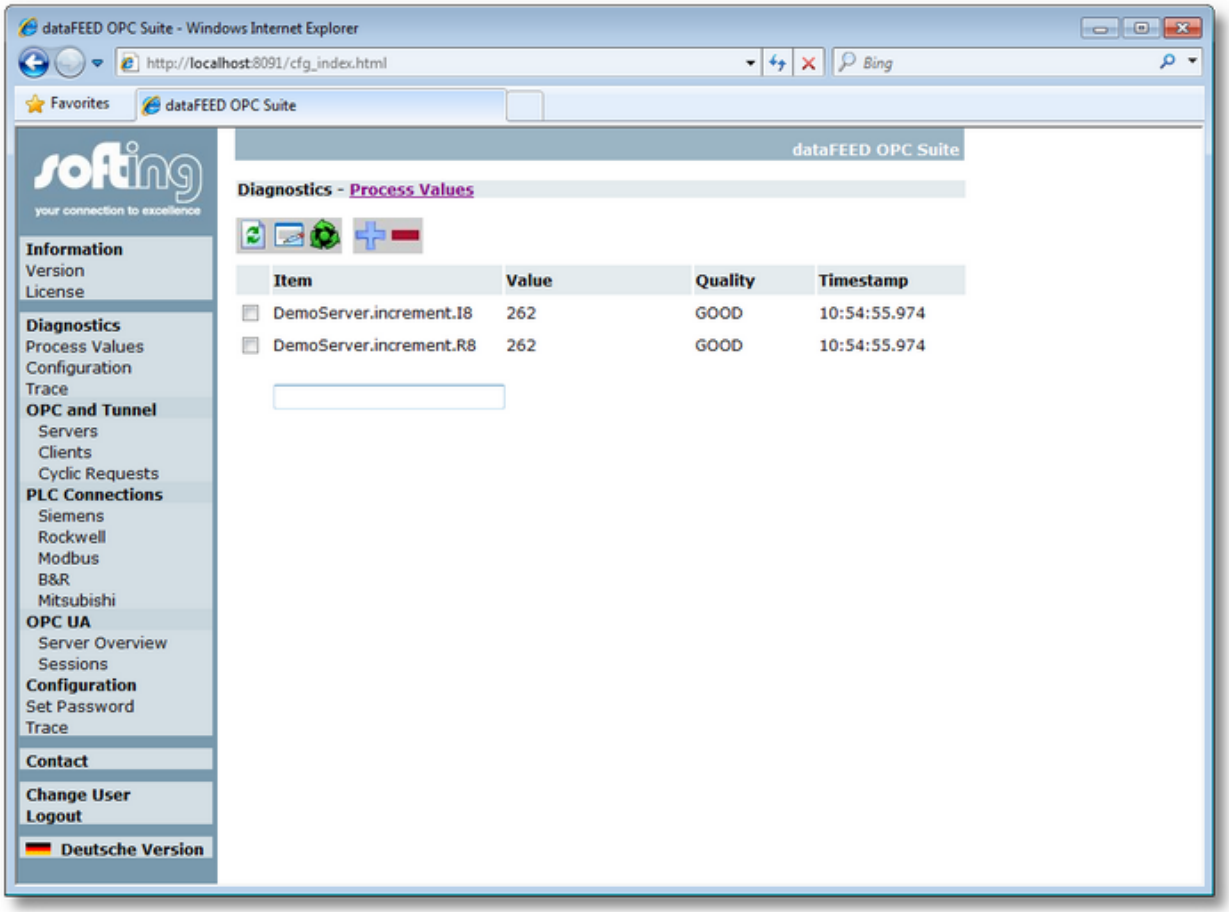
Select *Diagnostics* | *Process Values* to open the start up page for monitoring the process values - at the beginning the data item list is empty:



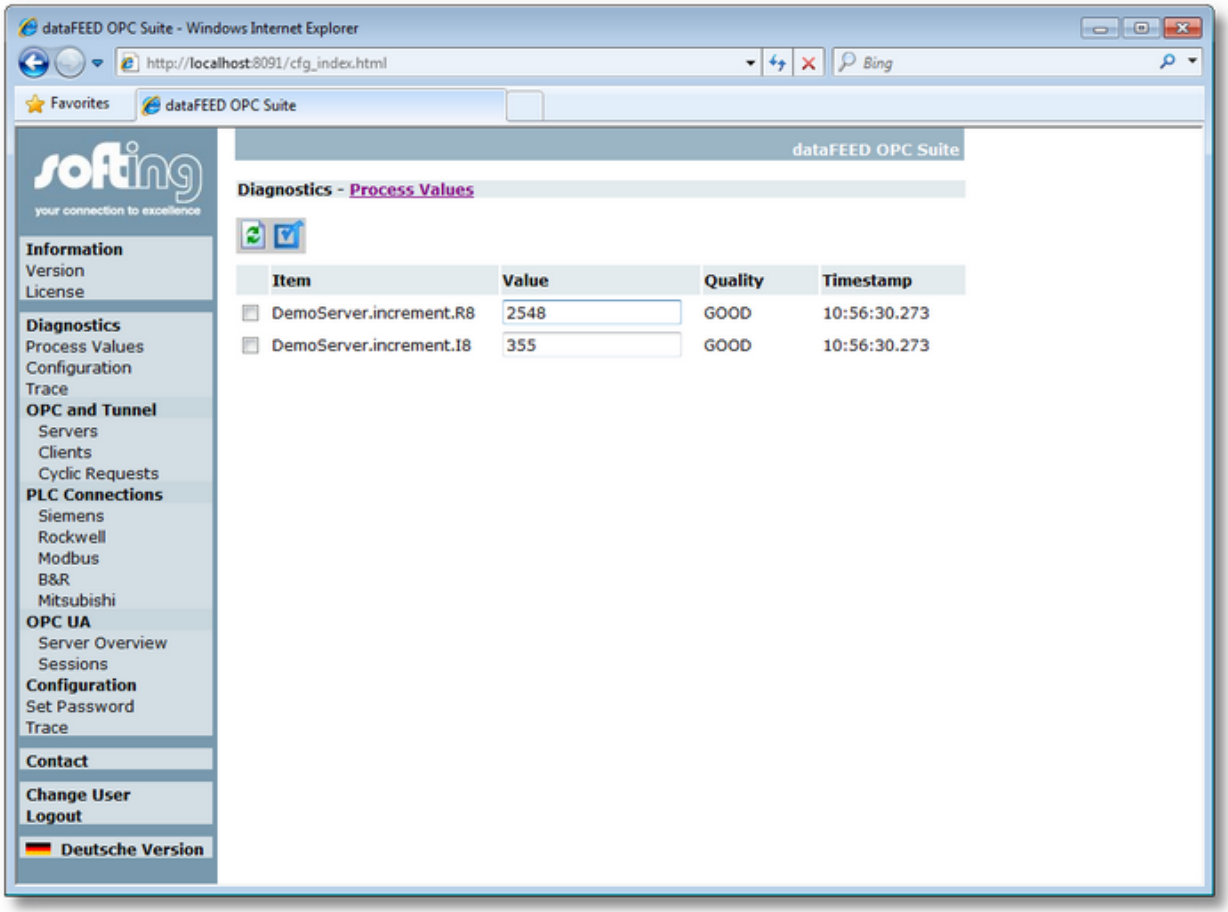
To add an item to the monitoring list, enter the complete item path in the edit control at the end of the list and press the button. Repeat this step for all items you want to add to the item list. Items can be deleted by selecting the corresponding item line in the list and pressing the button.

You can always manually refresh (manually read the values) by pressing the button. By pressing the button you can advise the web browser to automatically update the data values every five seconds. This automatic refresh can be stopped again by pressing the button.

© Softing Industrial Automation GmbH



Data item values can also be written by clicking the button which opens the corresponding write values page. To modify a data value, enter the new value in the corresponding edit control, select the associated item line on the left and press the button.



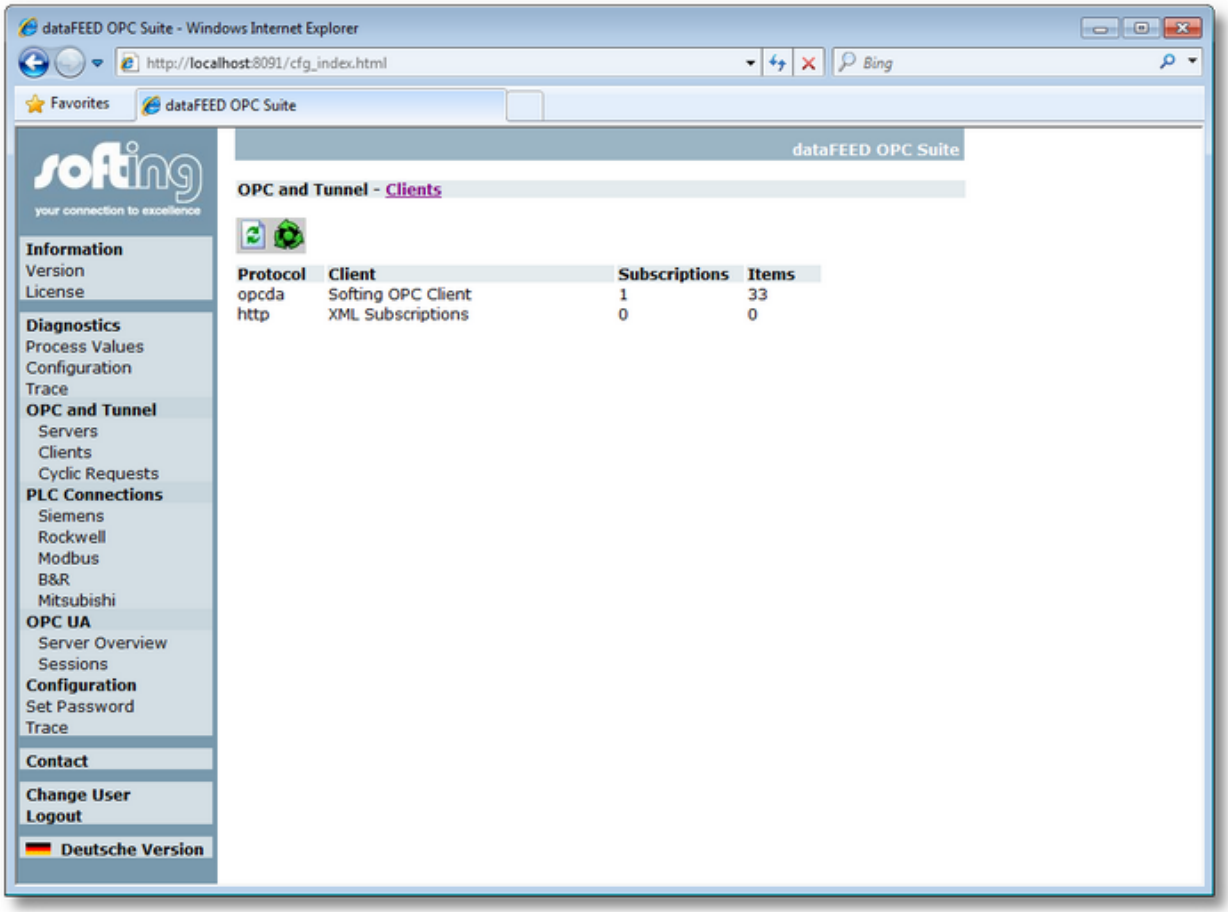
6.3.2.2 Clients

This web page displays the OPC and Tunnel clients currently connected to the local dataFEED OPC Suite application.

Note: To access this web page, you must be logged in either as operator or as administrator.

Select *Diagnostics* | *Clients* to open the page for displaying the currently connected clients.

© Softing Industrial Automation GmbH



Each client connection is shown in a separate line. To manually refresh the displayed data, press the button. By pressing the you can advise your web browser to automatically update the client data every five seconds. This automatic refresh can be stopped again by pressing the button.

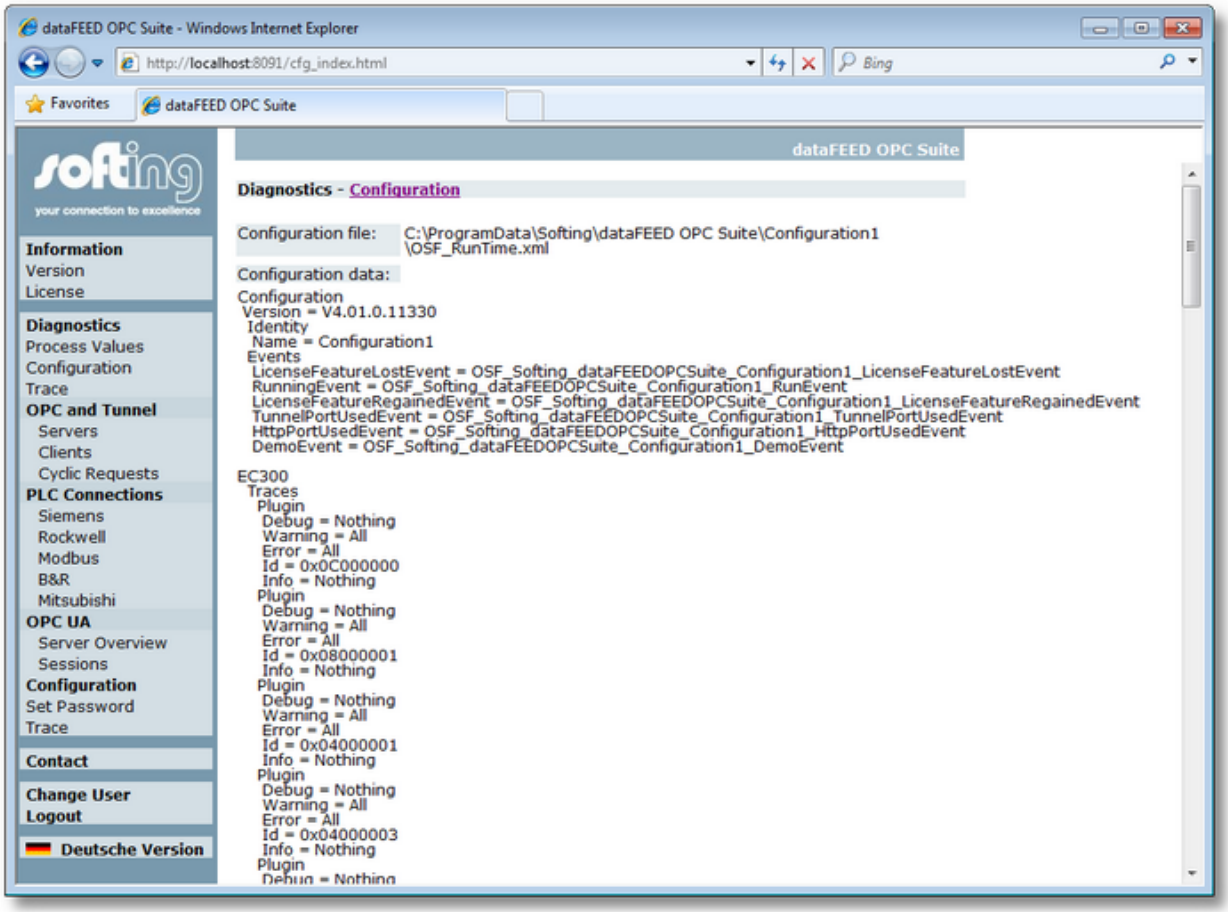
The *XML Subscriptions* line always contains the number of subscriptions and items for all the OPC XML-DA clients that are currently connected to the local dataFEED OPC Suite application. According to the OPC XML-DA Specification, an active subscription cannot be assigned to a specific client connection. Below the *XML Subscriptions* line, any active XML-DA client connection would be displayed separately, together with its IP address and IP port number.

6.3.2.3 Configuration

The parameters of the local dataFEED OPC Suite application can be displayed on this web page.

Note: To access this web page, you must be logged in either as operator or as administrator.

Select *Diagnostics* | *Configuration* to open the page for displaying the current configuration.



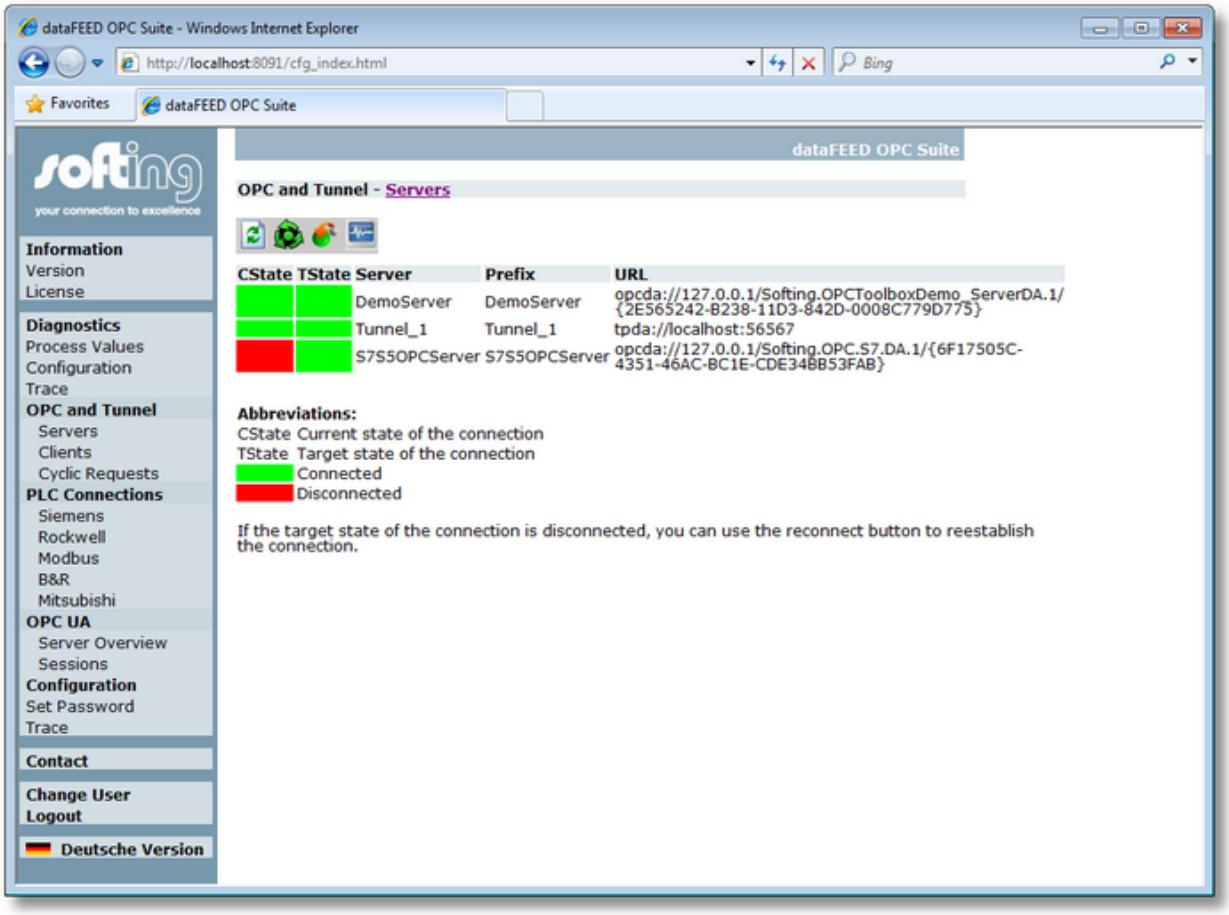
6.3.2.4 Servers

On this web page the current OPC and Tunnel server connections of the local dataFEED OPC Suite application are displayed.

Note: To access this web page, you must be logged in either as operator or as administrator.

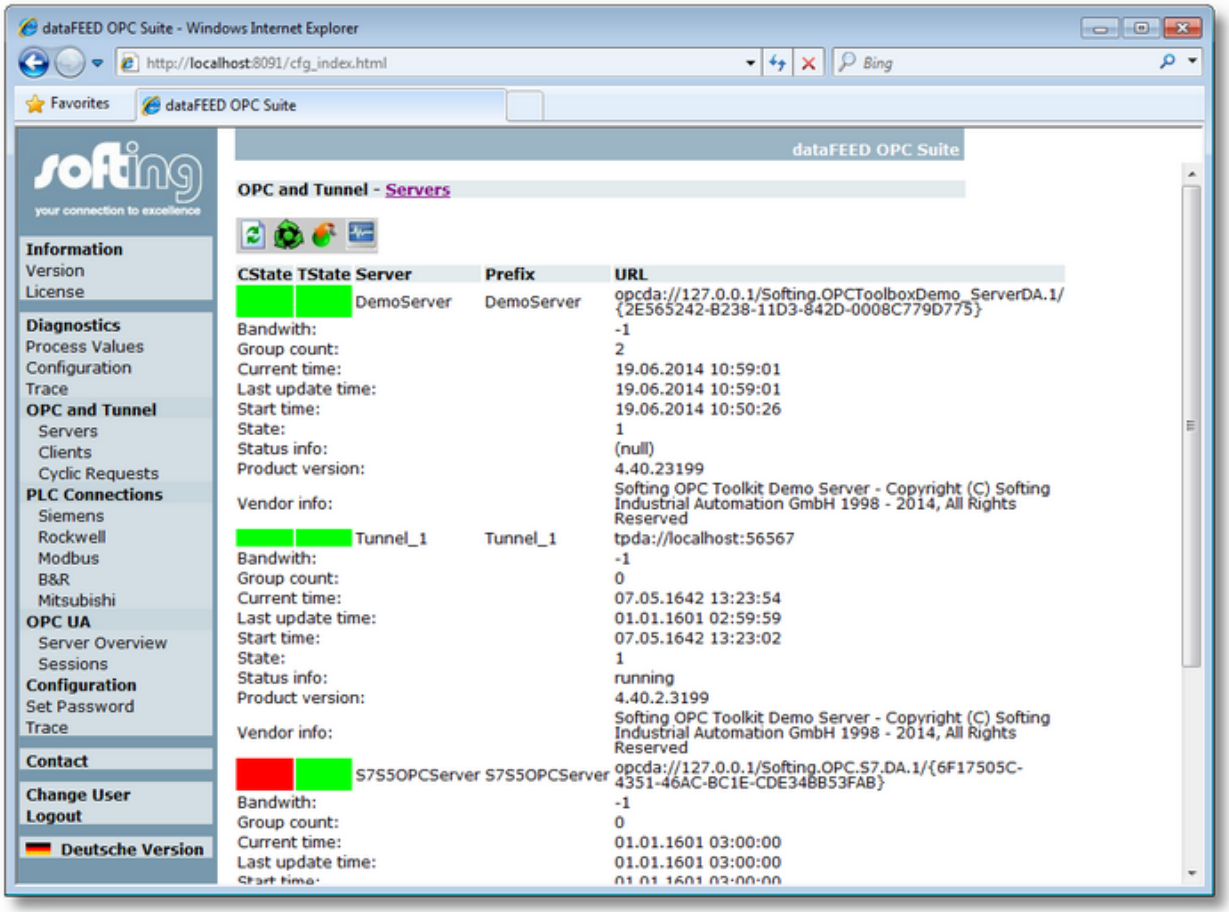
Select *Diagnostics* | *OPC and Tunnel* | *Servers* to open the page for displaying information about the current server connections.

© Softing Industrial Automation GmbH



The first two columns indicate the current state of the connection and the desired target state. When the local dataFEED OPC Suite application receives a shutdown request from an OPC server, it sets the target state to *Not connected*. Communication with the server can be reestablished by manually opening a connection to the corresponding server by pressing the button.

Details on each server connection can be displayed by pressing the button:



To manually refresh the displayed data, press the button. By pressing the you can advise your web browser to automatically update the server data every five seconds. This automatic refresh can be stopped again by pressing the button.

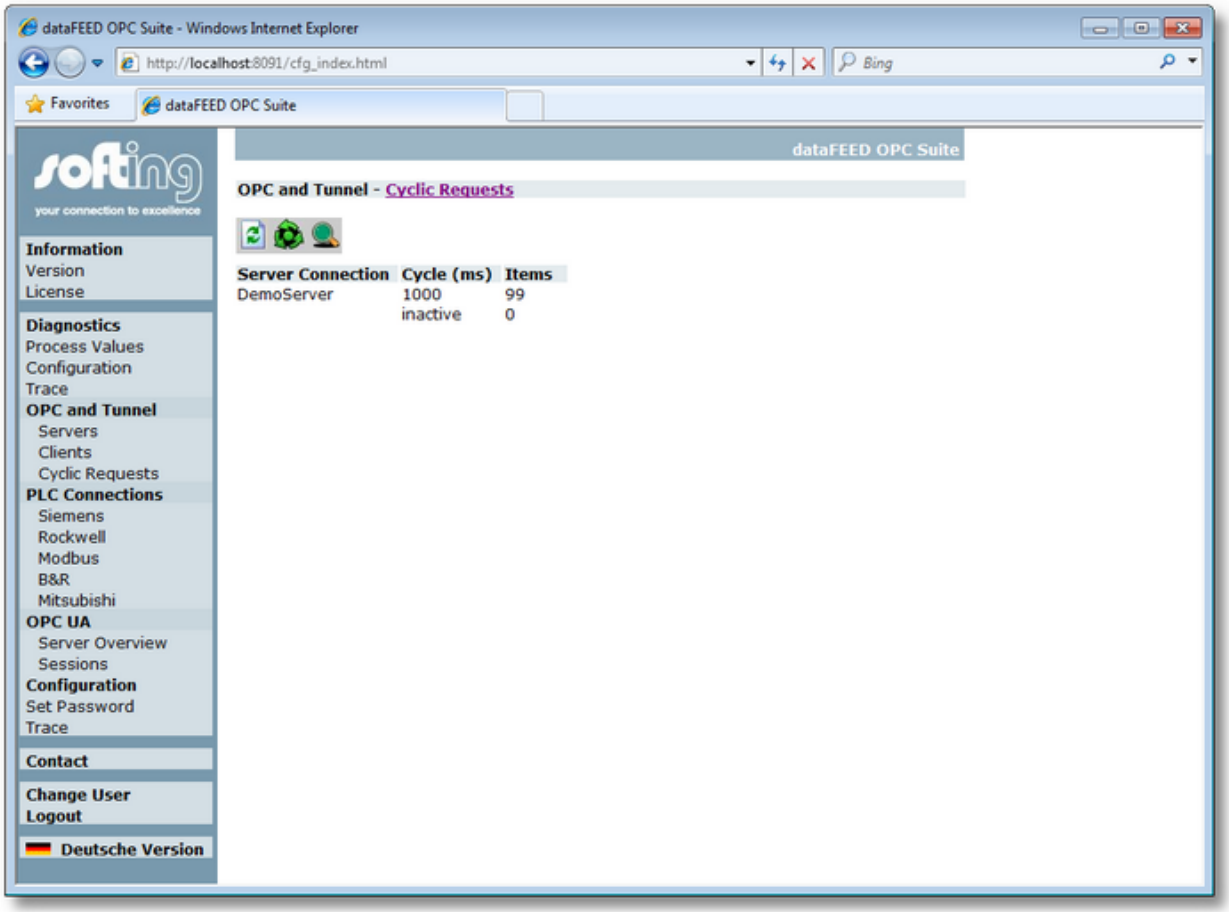
6.3.2.5 Cyclic Requests

On this web page the currently configured cyclic requests and some associated additional information are displayed for the local dataFEED OPC Suite application.

Note: To access this web page, you must be logged in either as operator or as administrator.

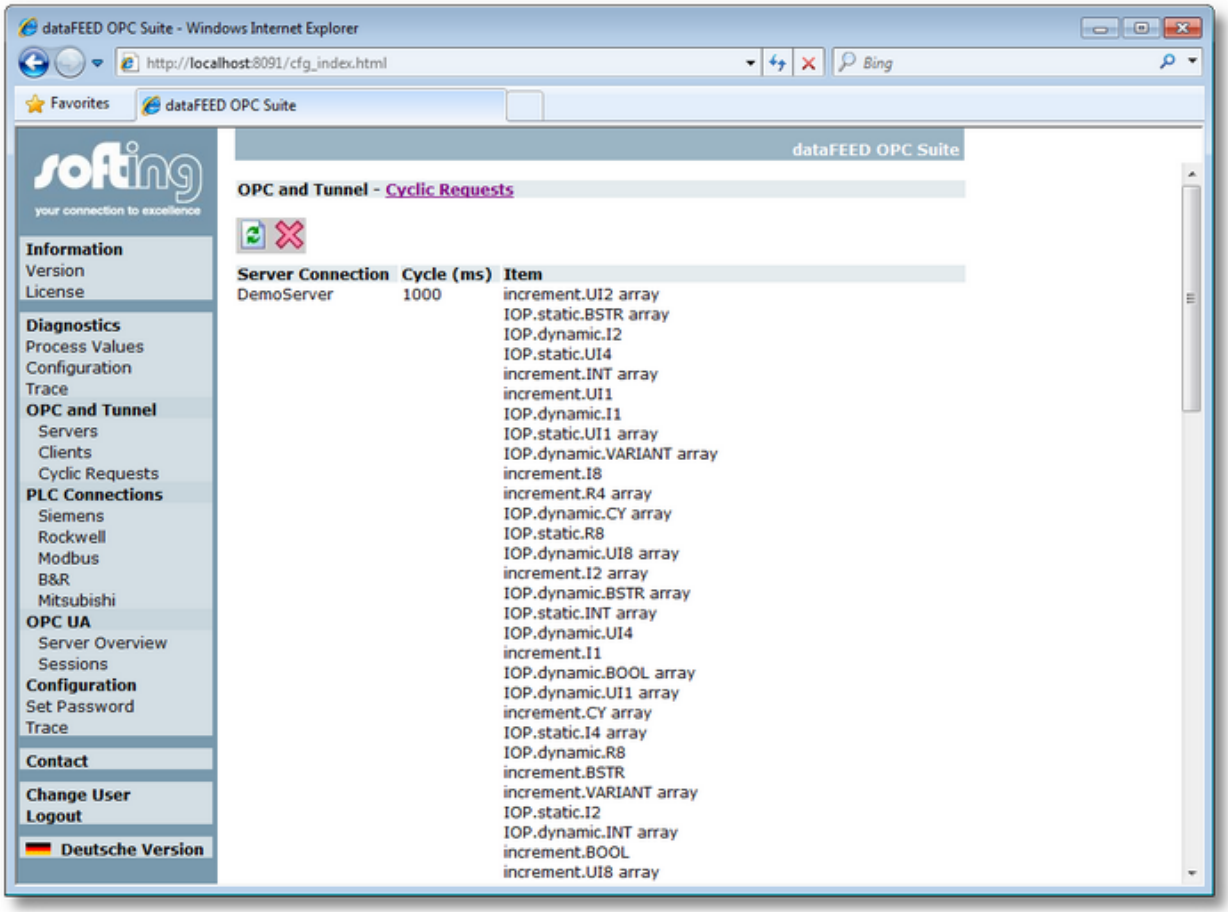
Select *Diagnostics* | *OPC and Tunnel* | *Cyclic Requests* to open the page for displaying information about the currently configured cyclic request.

© Softing Industrial Automation GmbH



Each active item in an active group, which an OPC client creates in the OPC server of the local dataFEED OPC Suite application, creates an entry in the Cyclic Requests page. Here you can see the server items which are automatically monitored for value changes.

Details on each server connection can be displayed by pressing the button:



If objects are shown in red on these two pages, they could not be created in the OPC server.

To manually refresh the displayed data, press the button. By pressing the you can advise your web browser to automatically update the server data every five seconds. This automatic refresh can be stopped again by pressing the button.

6.3.2.6 Device Connections

On this web page the currently configured device connections and some associated additional information are displayed for the local dataFEED OPC Suite application.

Note: To access this web page, you must be logged in either as operator or as administrator.

In the left side navigator there are different menu entries, corresponding to the different device families the local dataFEED OPC Suite application can access:

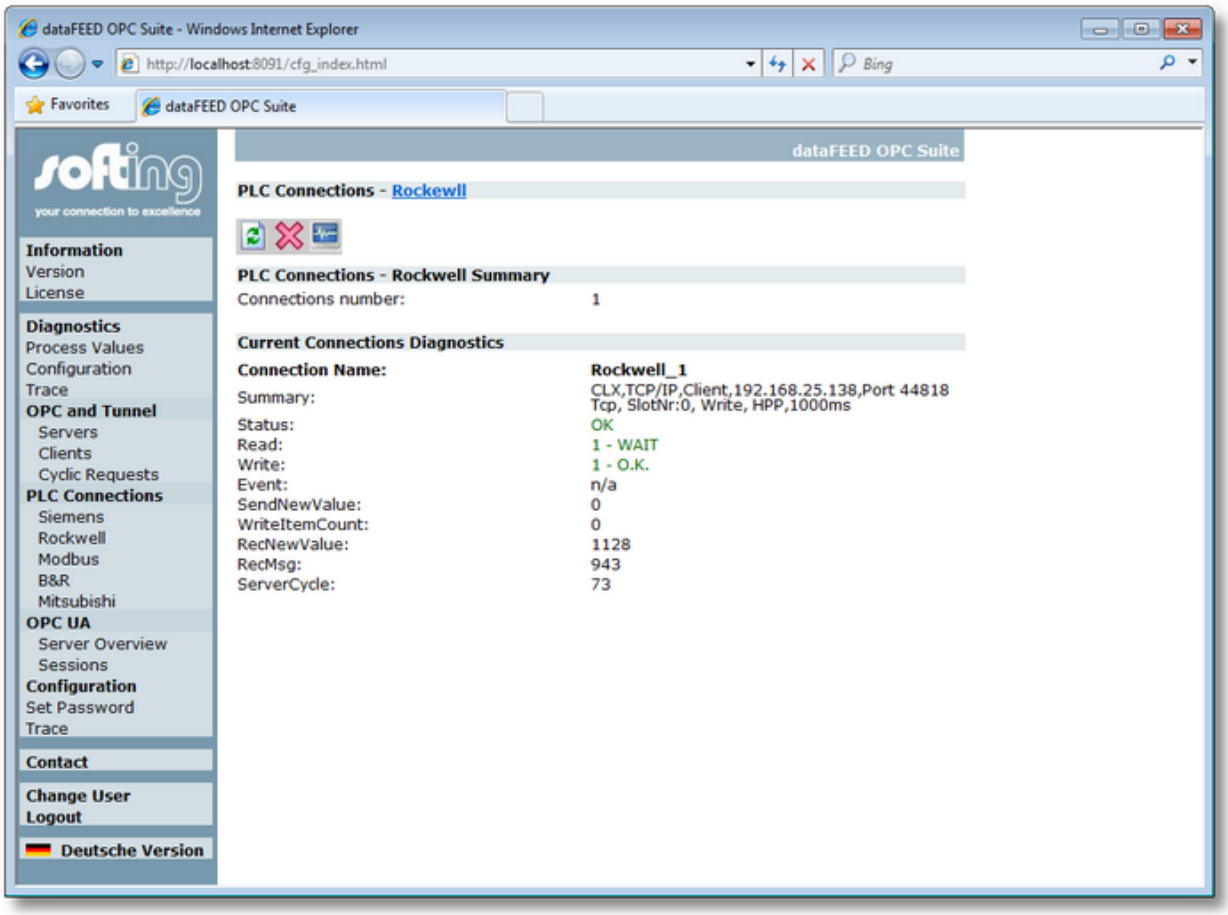
- **Siemens**
- **Rockwell**
- **Modbus**
- **B&R**

© Softing Industrial Automation GmbH

- **Mitsubishi**

Select a device family for displaying information about the current configured device connections of the corresponding type.

For the **Siemens, Rockwell, Modbus and Mitsubishi** families the information which is presented is quite similar.



You can always manually refresh (manually read the values) by pressing the button. By pressing the button you can advise the web browser to automatically update the data values every five seconds. This automatic refresh can be stopped again by pressing the button.

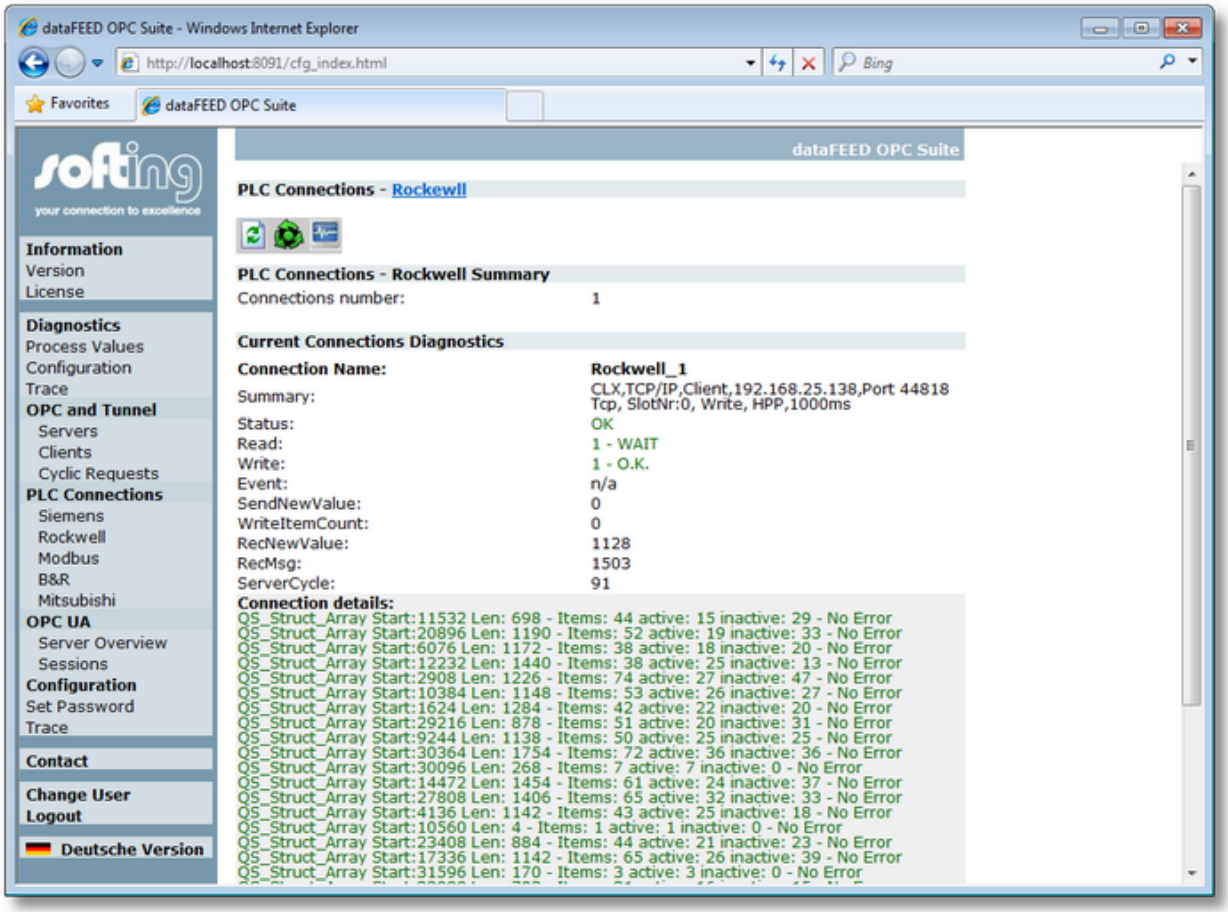
The table below provides a short description of the individual parameters which can be monitored for each device connection:

Attribute	Description
Connection Name	Name of the device connection./text>
Summary	Short summary of the basic configuration options for the current device connection.

Status	Overall operational status for the current device connection.
Read	
Write	
Event	
SendNewValue	The number of values up to now that were written by the OPC client to the OPC server. If write optimization is set to “Zusammenhängende Blöcke immer zusammenfassen” (always combine related blocks) and the write access speed of the OPC client is very fast, the OPC server optimizes and only writes the last word written to the controller. This means that there may be a difference between the number of items sent to the server by the client and the values that were passed on from the server to the hardware (PLC). SendNewValue corresponds to the logger entry “Gesendete Daten” (Show Send Values).
WriteItemCount	Specifies the number of values that were written up to now by the OPC server to the hardware (PLC). WriteItemCount corresponds to the logger entry “Senden” (Show Send).
RecNewValue	Specifies the number of values which were transmitted up to now by the OPC server to the OPC client. This always occurs when the value of a process variable changes. The messages to the client are not faster than the “Group Update Rate” provided by the client. When a value changes faster than specified in the “Group Update Rate,” the client is not informed about the intermediate value. RecNewValue corresponds to the logger entry “Empfange Daten” (Show Rec Values).
RecMsg	Specifies the number of frames which the OPC server has received up to now from the hardware. The read interval specified during connection parameterization shows how frequently the OPC server reads from the hardware. If the optimized protocol (High Performance Protocol) is used, the number of actually transferred frames may differ since this protocol can combine several inquiries in one frame. “RecMsg” corresponds to the logger entry “Empfangen” (Show Receive).
ServerCycle	The number of passes is indicated as to how often the server reads the registered frames of this connection from the PLC. After all frames are read and the set read access interval has expired, this number increments itself upward. The values can also be queried with the special item “ServerCycle” via the OPC interface.

Detailed information for each device connection can be displayed by pressing the  button. In this mode, the web browser will update the displayed data every five seconds

© Softing Industrial Automation GmbH



The individual data areas which are requested from the devices (PLC) are listed in one ore more subsequent line, under the *Connection details* label.

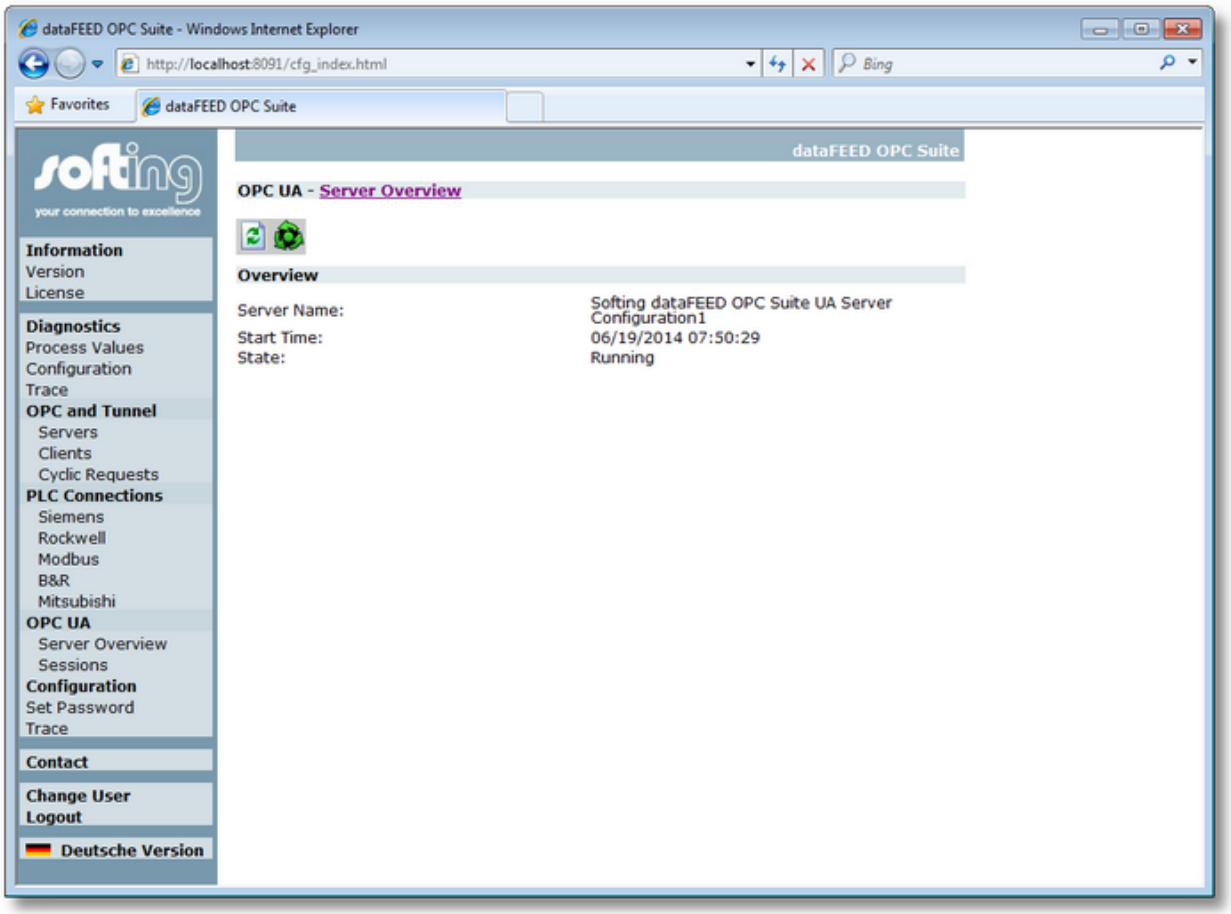
Attribute	Description
ID	Specifies which area of the hardware (PLC) the server is accessing (e.g., data block, flag, input, output, timer, etc.).
DB	The data block number is indicated.
Start	Specifies the start address of the area starting at which the area is accessed.
Length	Specifies the length of the area which will be read with this inquiry. The unity of the area is considered. The actual length (in bytes) results from the orientation of the data area.
Elements	Specifies how many items are registered for this access path and how many of these are switched active and inactive.
PLC error	Shows the error number and the error text which the PLC sent back to the inquiry.

6.3.2.7 OPC UA Server Information

Here you can find diagnostic information for each created OPC UA connection.

In the server overview you get information about:

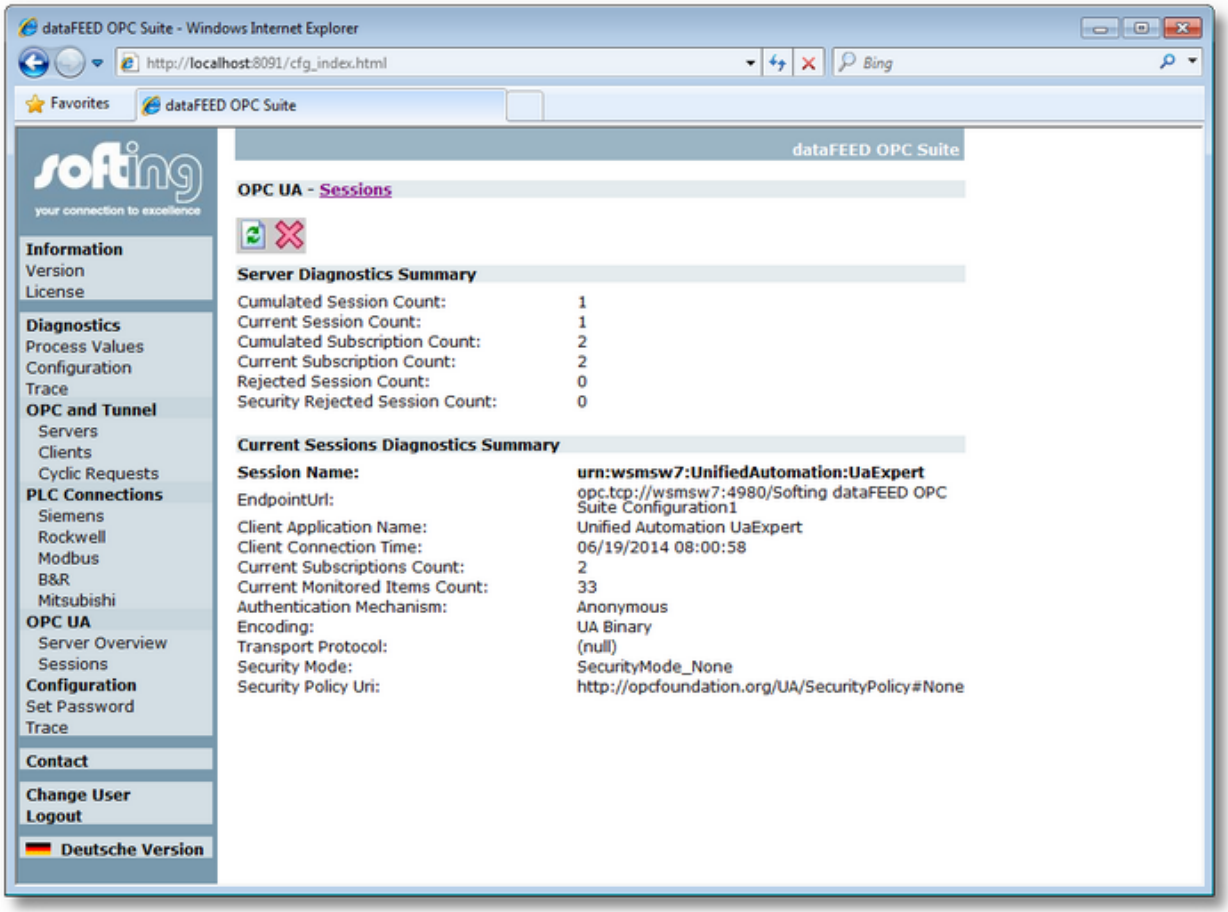
- server name
- start time
- state



Under "sessions" you get information about:

- diagnostic values of the server
- each running connection to an OPC UA Client
- security settings of the single connections

© Softing Industrial Automation GmbH



6.3.3 Configuration

- The following configuration web pages are available for the local dataFEED OPC Suite applications:

  - Configure the trace output configuration @ Configuration Web Pages: Trace.
  - Change the operator and administrator passwords @ Configuration Web Pages: Password.

Attention: To access these web pages, you must be logged in as administrator.

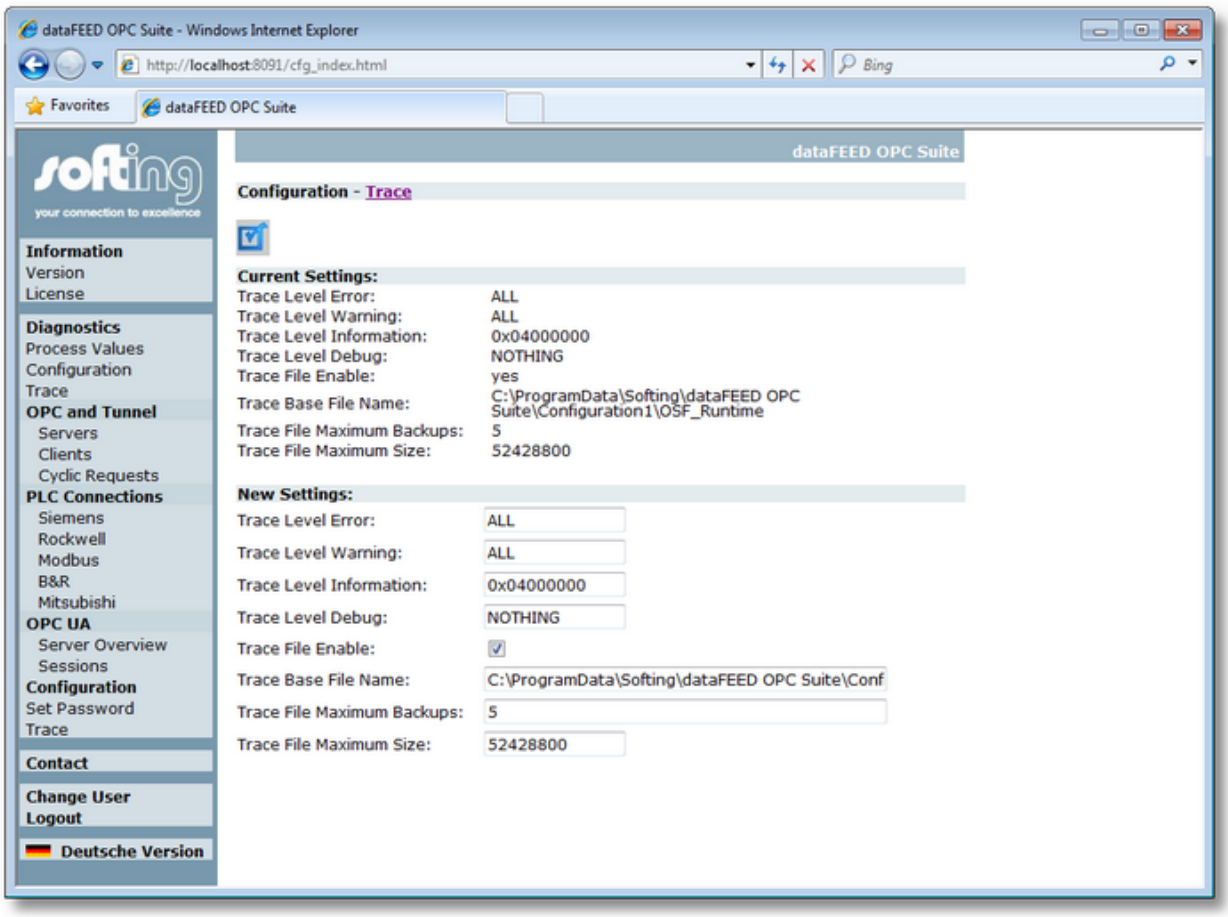
Softing dataFEED OPC Suite V5.38
© 2024 Softing Industrial Automation GmbH

6.3.3.1 Trace

On this web page the currently configured trace settings of the local dataFEED OPC Suite application are displayed and can be modified.

Note: To access this web page, you must be logged in as administrator.

Select *Configuration* | *Trace* to open the page for displaying and modifying the trace settings of the local dataFEED OPC Suite application.



Press the button to apply the changes. The different trace settings are described in detail in Configuration: Settings - Diagnostics.

Attention: If the trace configuration is changed using the web pages, the corresponding information loaded in the dataFEED OPC Suite Configurator is not updated automatically. To avoid inconsistencies, the corresponding configuration must be manually reloaded in the dataFEED OPC Suite Configurator.

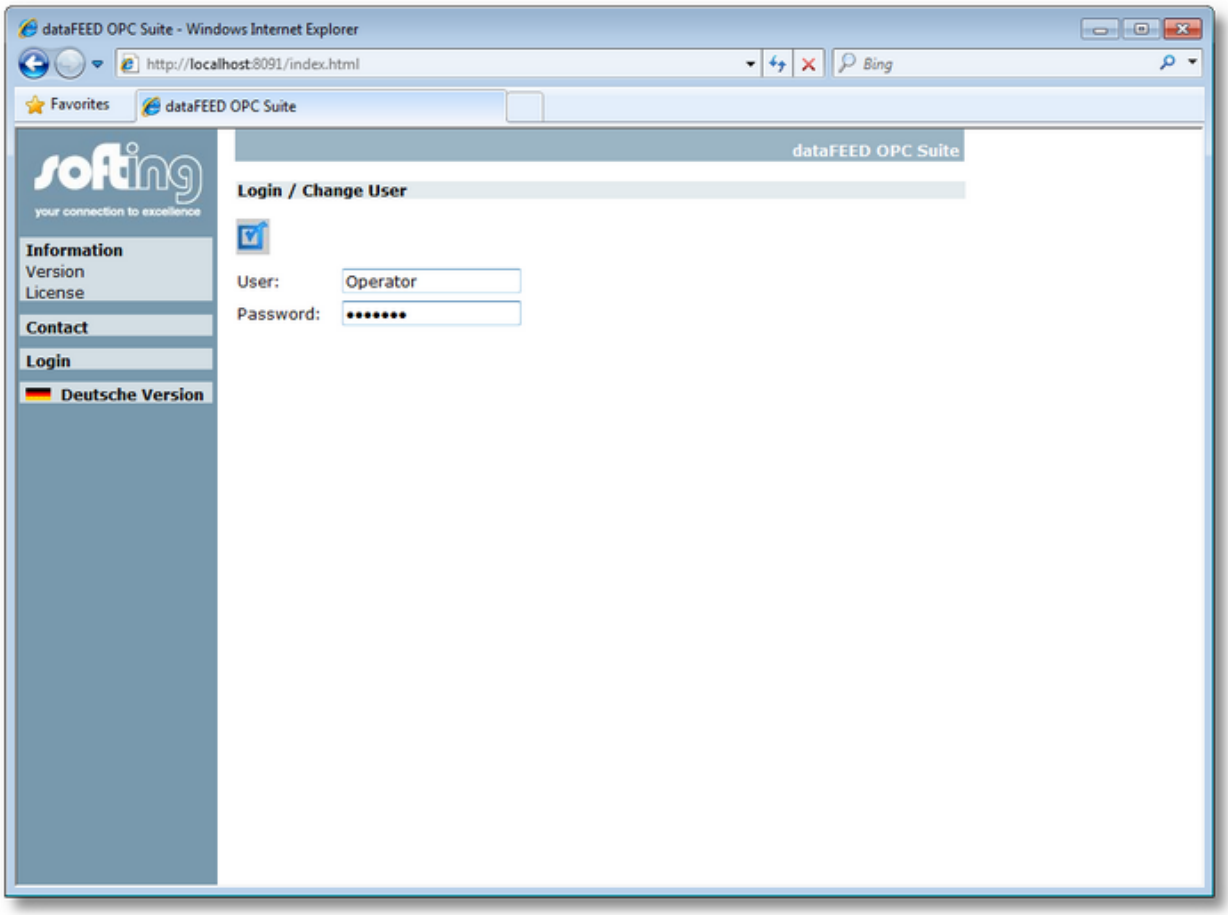
© Softing Industrial Automation GmbH

6.3.3.2 Password

On this web page you can change the passwords of the operator and administrator users for the web access of the local dataFEED OPC Suite application.

Note: To access this web page, you must be logged in as administrator.

Select *Configuration* | *Set Password* to open the page for modifying the operator and administrator passwords of the local dataFEED OPC Suite application.



Press the button to apply the changes. These passwords can also be changed in dataFEED OPC Suite as described in detail in Configuration: Settings - Web Server.

Attention: If the passwords are changed using the web pages, the corresponding information loaded in the dataFEED OPC Suite Configurator is not updated automatically. To avoid inconsistencies, the corresponding configuration must be manually reloaded in the dataFEED OPC Suite Configurator.

6.4 Building Own Web Pages

Softing dataFEED OPC Suite features an integrated web server which allows you to provide your own web pages for displaying and setting values. The architecture and performance of the web server have been designed for small-scale visualizations.

You will find the original Softing dataFEED OPC Suite web pages in the directory *<WindowsProgramFolder>\Softing\dataFEED OPC Suite\www\Softing*. Beyond the *.\www\Softing* folder you can store your own web pages. The integrated web server will first browse the *www* and then the *www\Softing* folder for the corresponding HTML files.

In the corresponding *www* folder you will find an example of a visualization page (*sampleSite.html*). To display this sample visualization page, just type *http://localhost:<PortNumber>/sampleSite.html* into the address field of your web browser (where *<PortNumber>* is the current TCP/IP port number for the corresponding configuration - typically *8091*).

Note: The page uses ActiveX controls, which may only function properly if you have customized the security settings of your web browser accordingly.

Displaying Item Values and Item Qualities

The web server allows you to integrate wildcard variables in the HTML code. These variables will then be replaced by the current values when you refresh the HTML page.

The wildcards generally use the following syntax:

```
<%#wildcard#%>
```

The following wildcard types are supported:

Wildcard	Actual Value
`ITEMVALUE##ItemID`	The actual data value of the data item with the given item ID.
`ITEMQUALITY##ItemID`	The actual quality of the data item with the given item ID as a string.
`ITEMQUALITY##ItemID##color`	The actual quality of the data item with the given item ID as a color code. A GOOD quality is shown in black, all other qualities in gray.

Example: The `<%#ITEMVALUE##_System.General.Heartbeat#%>` wildcard will be replaced by the value of the *_System.General.Heartbeat* item.

Writing Item Values

You can also use the web server to set the values of data items. This is done by transmitting the data of an HTML form from the client to the server.

© Softing Industrial Automation GmbH

The form must contain an element representing the action to be executed.

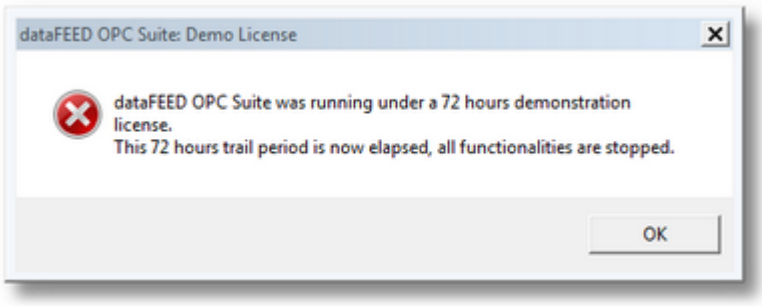
When writing individual values, you have to use the command `CMD##WRITEITEM`. The values to be written are reported to the web server via input elements named `ITEMVALUE##ItemID`.

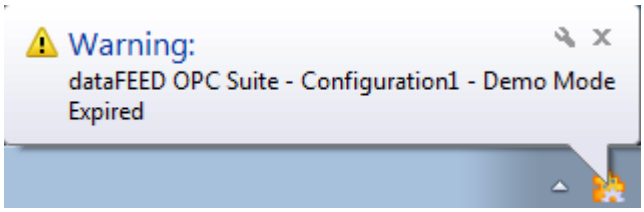
When writing multiple values on one page, you have to use the command `CMD##WRITEITEM##CHECK`. Here, too, the values to be written are reported to the web server via input elements named `ITEMVALUE##ItemID`. However, only those values will be written for which a `CHECK##ItemID` has been set.

6.5 Demonstration Mode

If no license is installed on the local computer Softing dataFEED OPC Suite runs for 72 hours in a limited demonstration mode. All dataFEED OPC Suite features are completely enabled.

When the demonstration period ends after 72 hours, the local dataFEED OPC Suite application stops all functionalities. The user is notified by a corresponding warning message from the dataFEED OPC Suite Tray Application, and both the Configurator and the Tray Application show the corresponding yellow icon.





Restarting the local dataFEED OPC Suite application will start a new 72-hours demonstration period.

Note: The demonstration mode is only available if no licenses are installed on the corresponding computer. Installing at least one license key will disable the demonstration mode completely.

6.6 Importing Multiprotocol OPC Server Configurations

dataFEED OPC Suite provides an easy way to import already configured PLC connections from the Softing Multiprotocol OPC Server.

The import will have as a result the automatically addition to the current dataFEED OPC Suite configuration of the selected PLC connections identified in the specified Multiprotocol OPC Server configuration, together with the connections settings and the available symbols, if specified so.

Note: The import will NOT create a new configuration, but merge the items selected for the import into the currently opened dataFEED OPC Suite configuration.

Note: The configuration may need additional user adjustments for the PLC specific settings.

You can start the dataFEED OPC Suite Import Multiprotocol OPC Server Configuration wizard by selecting *Configuration / Import Multiprotocol OPC Server Configuration* in the dataFEED OPC Suite Configurator main menu.

Attention: If no PLC families are selected for usage in the *Feature Selection Dialog*, the *Import Multiprotocol OPC Server Configuration* will not be present in the *Configuration* menu.

▼ This chapter of the dataFEED OPC Suite Online Help describes the usage of the Multiprotocol OPC Server Configuration Import Wizard:

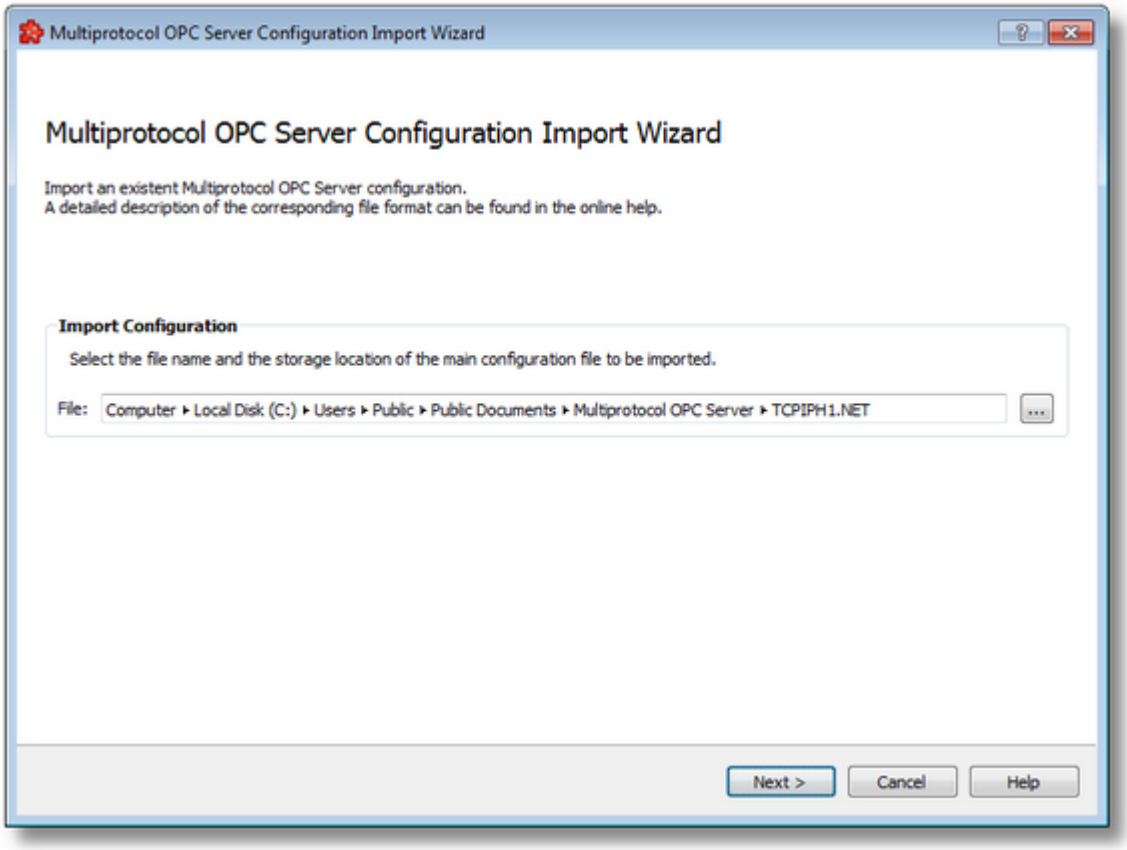
- Select the file to be imported @ Import Multiprotocol OPC Server Configuration - Configuration File Selection
- Select the PLC connections to import @ Import Multiprotocol OPC Server Configuration - PLC Connections Selection
- Select the PLC connections to import symbols from @ Import Multiprotocol OPC Server Configuration - PLC Symbols Selection
- Examine the import results @ Import Multiprotocol OPC Server Configuration - Results

Softing dataFEED OPC Suite V5.38
© 2024 Softing Industrial Automation GmbH

6.6.1 Import - Configuration File Selection

The first page of the Multiprotocol OPC Server Import wizard allows you to choose a Multiprotocol OPC Server configuration to be imported. The configuration is chosen by specifying its main file, usually named *TCPIPH1.NET.*

The file can be selected by manually typing its name in the *File* input box or by clicking the browse button located to the right of the input box (a standard file selection dialog will open).



The file path and name should be valid (they should point to an existing file).

When you press the *Next* button, the file path and name are verified. If the file identified by the specified file path and name does not exist, an error message is displayed.

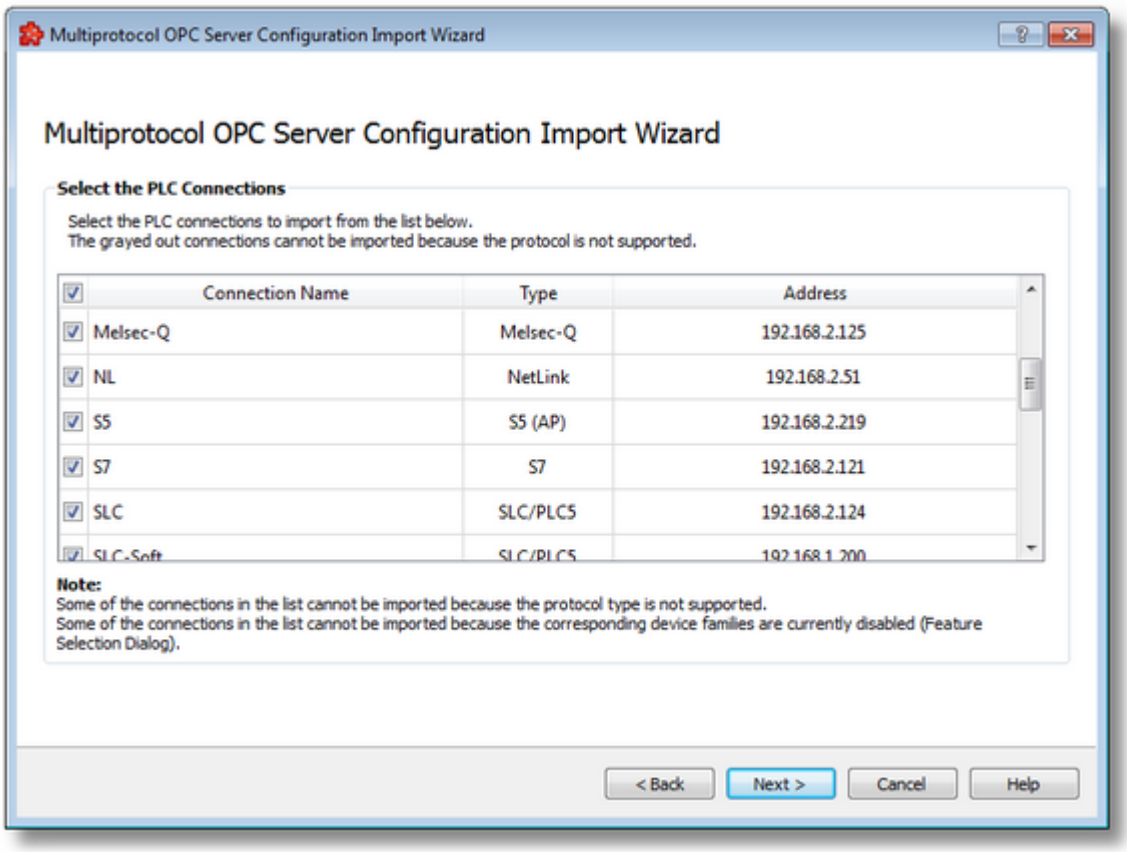
Multiprotocol OPC Server Configuration Import Wizard - PLC Connections Selection >>>

6.6.2 Import - PLC Connections Selection

The second page in the Multiprotocol OPC Server Import wizard allows you to choose one ore more PLC connections to be imported, from the list with the connections identified in the specified Multiprotocol OPC Server configuration.

A list with all available PLC connections is presented and you must select the desired connections within it.

For user comfort, a select/unselect all check-box is offered in the header of the list.

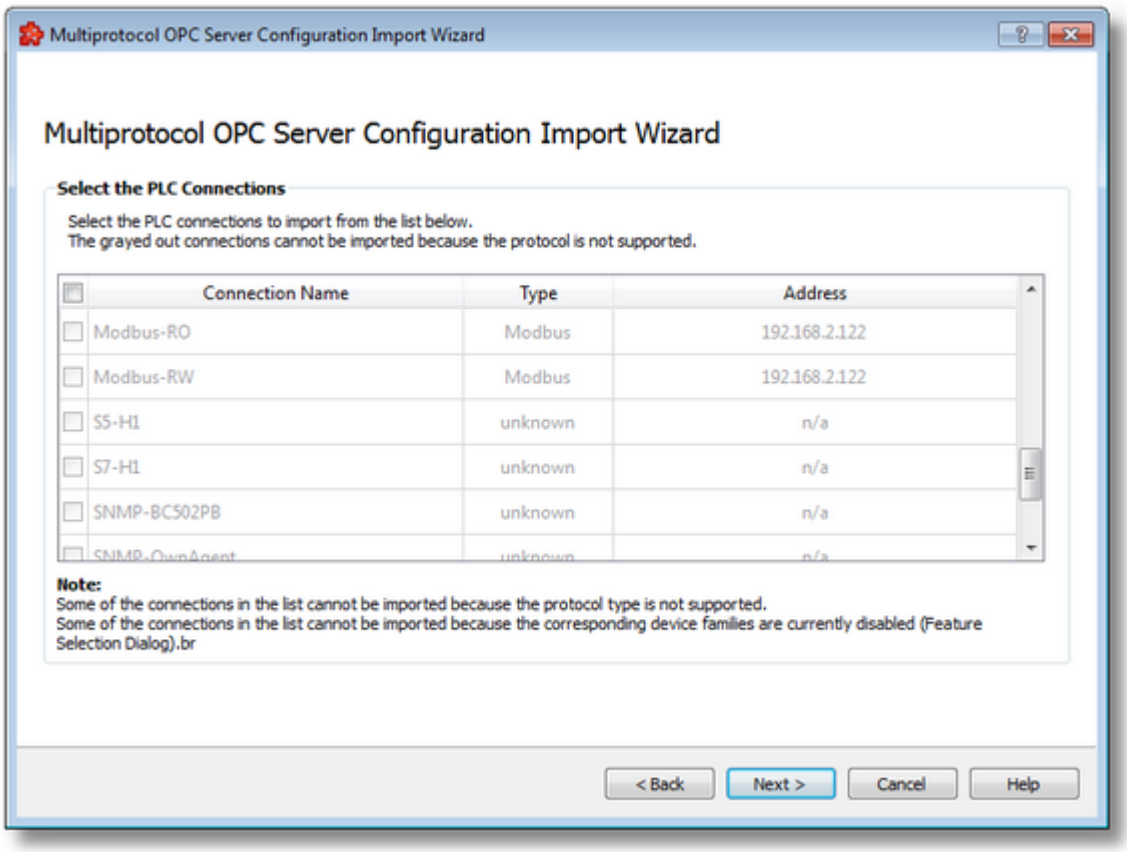


Note: The current configuration may already have defined connections with the names selected for the import from the Multiprotocol OPC Server configuration. In this case, the user is notified with a dialog box and asked for confirmation of some adjustments to the connection names to be imported. In order to ensure the uniqueness of the connection names, the newly imported connections names which already exist will be added with the *_imported* suffix (and a numeric ID, if necessary).

Note: At least one PLC connection must be selected in order to continue the wizard.

Some of the connections might not be selectable, as shown in the screenshot below:

© Softing Industrial Automation GmbH



A listed PLC connection can be non-selectable and also marked in grayed-out text for one of the following reasons:

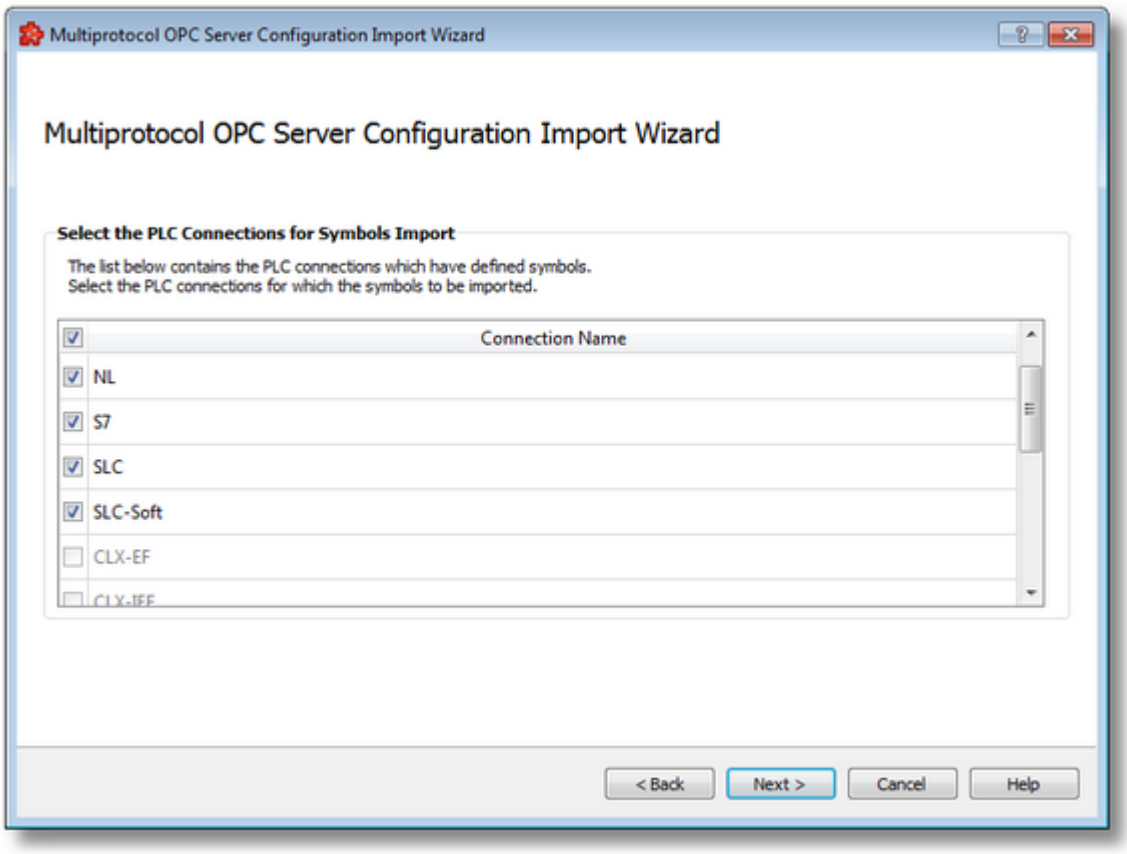
1. The protocol is no longer supported in the current version of the dataFEED OPC Suite. Such a connection can be easily identified by the *unknown* text displayed in *Type* column.
2. The family to which the protocol of the PLC connection belongs to is currently disabled in the *Feature Selection Dialog.*

<<< Multiprotocol OPC Server Configuration Import Wizard - Configuration Selection
Multiprotocol OPC Server Configuration Import Wizard - PLC Symbols Selection >>>

6.6.3 Import - PLC Symbols Selection

The third page in the Multiprotocol OPC Server Import wizard allows you to choose whether the symbols are to be also imported for the selected PLC connections in the previous page.

A list is shown. It contains those PLC connections selected for import which also have symbols defined. You can select one ore more connections within this list.



Note: The PLC connections which were selected in the previous page but don't have defined symbols are displayed as grayed-out and non-selectable.

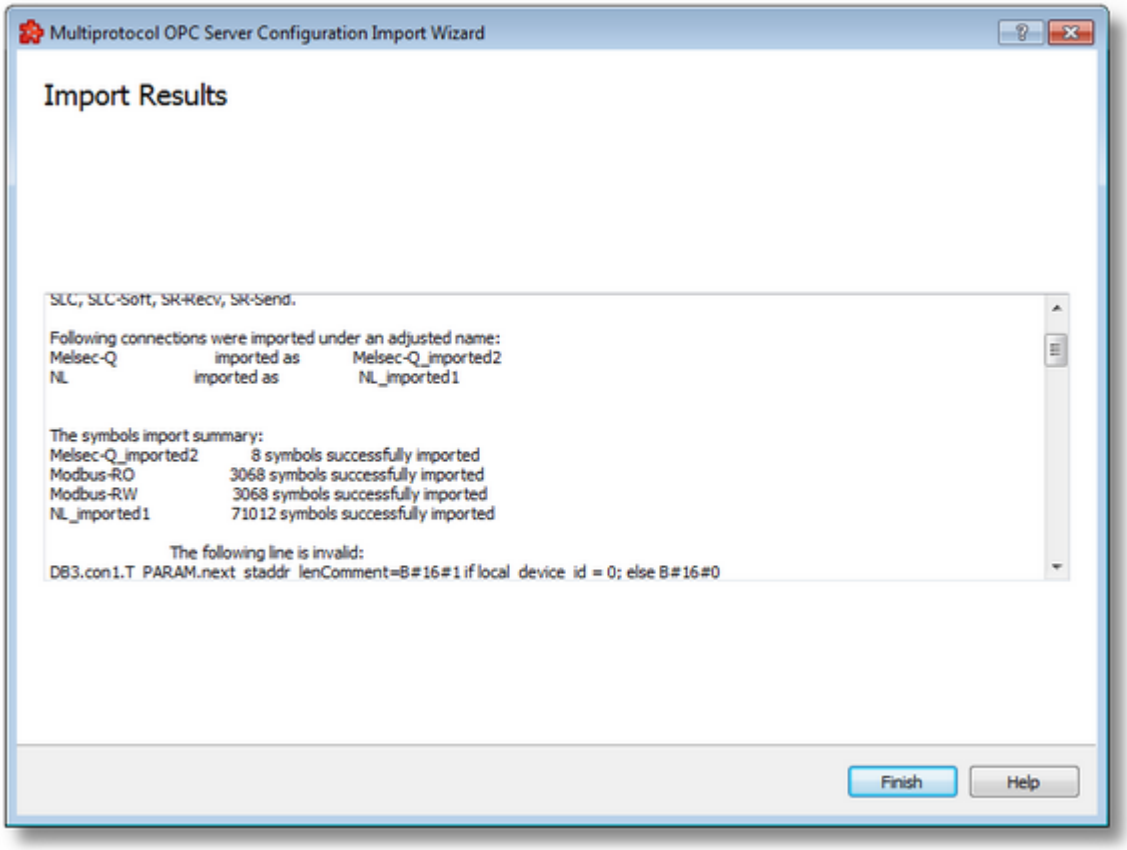
Note: The wizard can be continued even with nothing selected in this page. In this case, a notification message box will be displayed.

<<< Multiprotocol OPC Server Configuration Import Wizard - PLC Connections Selection
Multiprotocol OPC Server Configuration Import Wizard -Results >>>

6.6.4 Import - Results

The fourth page of the Multiprotocol OPC Server Import wizard displays relevant information for examining the import result.

© Softing Industrial Automation GmbH



This page is the final page of the Multiprotocol OPC Server Import wizard. A back button is not provided as the import action cannot be undone. The import can be successful or it can fail (totally or partially).

The information presented in the import results covers the following:

- The list of the PLC connections selected for import.
- The list of the successfully imported PLC connections, preserving also the original connection name.
- The list of the PLC connections imported under an adjusted name for ensuring unique names for the connections.
- The number of items imported for each PLC connection, for which this option was selected.
- Error found while parsing the symbol file.

After a successful import, the import wizard is closed. The new PLC connections are merged with their imported items into the existing address space of the current configuration.

<<< Multiprotocol OPC Server Configuration Import Wizard - PLC Symbols Selection

6.7 Identity Settings

This chapter of the dataFEED OPC Suite Online Help describes the following topics:

- Display the default identity settings for dataFEED OPC Suite V5.38 @ Identity Settings in Softing dataFEED OPC Suite V5.38.
- Display the differences between V2.x/3.x and V4.x identity settings @ Changes Between V2.x/3.x and V4.x Configurations.

6.7.1 dataFEED OPC Suite V5.38

Softing dataFEED OPC Suite V5.38 uses the following default identity settings. When a new configuration is created, the dataFEED OPC Suite Configurator always creates a new set of unique identity values.

Attention: All of the following identity values must be unique over all configurations running on one computer. If they are manually changed, the corresponding uniqueness must also be ensured manually.

Data Destination - OPC Client

OPC Data Access Clients

The identity settings for OPC Data Access (DA) clients can be modified in the dataFEED OPC Suite Configurator as described in detail under Configuration: Data Destination - OPC Client.

Attribute	Default Value
OPC DA server name	*Softing dataFEED OPC Suite - <ConfigurationName>*
OPC DA server class ID	*{DA771DCC-3C2C-4687-81F2-FF79C0355D20}*
OPC DA server program ID	*Softing.OPC.DF.<ConfigurationName>.DA*

The <*ConfigurationName*> attribute is replaced with the real name of the configuration.

OPC Alarms and Events Clients

The identity settings for OPC Alarms and Events (AE) clients can be modified in the dataFEED OPC Suite Configurator as described in detail under Configuration: Data Destination - OPC Client.

© Softing Industrial Automation GmbH

Attribute	Default Value
OPC AE server name	*Softing dataFEED OPC Suite - <ConfigurationName>*
OPC AE server class ID	*{A6455EB9-4719-4D6C-BD1D-D9577053E170}*
OPC AE server program ID	*Softing.OPC.DF.<ConfigurationName>.AE*

The <*ConfigurationName*> attribute is replaced with the real name of the configuration.

OPC XML Data Access Clients

The identity settings for OPC XML Data Access (XML-DA) clients can be modified in the dataFEED OPC Suite Configurator as described in detail under Configuration: Data Destination - OPC Client.

Attribute	Default Value
TCP/IP port number	*8091*

Note: The default XML-DA port number is incremented by one for each new configuration.

Note: This TCP/IP port number is also used as the port number for the integrated web server.

Data Destination - Tunnel Client

The identity settings for OPC Tunnel clients can be modified in the dataFEED OPC Suite Configurator as described in detail under Configuration: Data Destination - Tunnel Client.

Attribute	Default Value
TCP/IP port number	*56766*

Note: The default Tunnel port number is incremented by one for each new configuration.

Data Destination - OPC UA Client

The identity settings for OPC UA Clients can be modified in the dataFEED OPC Suite Configurator. Find more information on Configuration - Data Destination - OPC UA Client

Attribute	Default Value

Server Name	Softing dataFEED OPC Suite OPC UA Server <ConfigurationName>
Server Endpoint	*Acceptable formats for Server Endpoints are:* *opc.tcp://<HostName>:<Port>/<Suffix> or opc.tcp://<IP Address>:<Port>/<Suffix>*

Note: The default OPC UA port number is incremented by one for each new configuration, starting with port number 4980.

Windows Service

If the local dataFEED OPC Suite application is configured to be executed as a Windows service, the following settings are used to register the corresponding server in the Windows Service Manager. The identity settings can be modified in the dataFEED OPC Suite Configurator as described in detail under Configuration: Settings - Advanced.

Attribute	**OEM specific values**
Windows service name	*SofDF<ConfigurationName>*
Windows class name	*SofDF<ConfigurationName>*
NT service description	*Softing dataFEED OPC Suite - <ConfigurationName>*

The <*ConfigurationName*> attribute is replaced with the real name of the configuration.

Add here some information about the UA Server identities...

6.7.2 Changes Between V2.x/V3x and V4.x

This chapter of the Softing dataFEED OPC Suite Online Help describes the differences in the default identity settings between V2.x/V3.x configurations and V4.x dataFEED OPC Suite configurations.

Note: All the different default identity settings have been changed with version V4.x.

Data Destination - OPC Client

OPC Data Access Clients

© Softing Industrial Automation GmbH

The identity settings for OPC Data Access (DA) clients can be modified in the dataFEED OPC Suite Configurator as described in detail under Configuration: Data Destination - OPC Client.

Attribute	Default Value V2.x/V3.x	Default Value V4.x
OPC DA server name	*Softing Easy Connect - <ConfigurationName>*	*Softing dataFEED OPC Suite - <ConfigurationName>*
OPC DA server class ID	*{A76F65FF-A34F-43dd-9595-C3351B490F37}*	*{DA771DCC-3C2C-4687-81F2-FF79C0355D20}*
OPC DA server program ID	*Softing.OPC.EC.<ConfigurationName>.DA*	*Softing.OPC.DF.<ConfigurationName>.DA*

The <*ConfigurationName*> attribute is replaced with the real name of the configuration.

OPC Alarms and Events Clients

The identity settings for OPC Alarms and Events (AE) clients can be modified in the dataFEED OPC Suite Configurator as described in detail under Configuration: Data Destination - OPC Client.

Attribute	Default Value V2.x/V3.x	Default Value V4.x
OPC AE server name	*Softing Easy Connect <ConfigurationName>*	*Softing dataFEED OPC Suite - <ConfigurationName>*
OPC AE server class ID	*{26BBB9F2-4792-4475-B268-BB0F9BA86C5E}*	*{A6455EB9-4719-4D6C-BD1D-D9577053E170}*
OPC AE server program ID	*Softing.OPC.EC.<ConfigurationName>.AE*	*Softing.OPC.DF.<ConfigurationName>.AE*

The <*ConfigurationName*> attribute is replaced with the real name of the configuration.

Data Destination - OPC UA Client

The identity settings for OPC UA clients can be modified in the dataFEED OPC Suite Configurator as described in detail under Configuration: Data Destination - OPC UA Client.

Attribute	Default Value V2.x/V3.x	Default Value V4.x

OPC UA Server Name	*Softing Easy Connect OPC UA Server <ConfigurationName>*	*Softing dataFEED OPC Suite UA Server <ConfigurationName>*
OPC UA Server URL	*urn:<HostName>/Softing/OpcUA/<ConfigurationName>*	*urn: <HostName>/Softing/dataFEEDOPC Suite/OpcUA/<ConfigurationName>*

The <*ConfigurationName*> attribute is replaced with the real name of the configuration.

Windows Service

If the local dataFEED OPC Suite application is configured to be executed as a Windows service, the following settings are used to register the corresponding server in the Windows Service Manager. The identity settings can be modified in the dataFEED OPC Suite Configurator as described in detail under Configuration: Settings - Advanced.

Attribute	**Default Value V2.x/V3.x**	**Default Value V4.x**
Windows service name	*Softing dataFEED OPC Suite <Configuration Name>*	*SofDF<ConfigurationName>*
Windows class name	*OCTsvc <Configuration Name>*	*SofDF<ConfigurationName>*
Windows service description	*OCTsvc <Configuration Name>*	*Softing dataFEED OPC Suite - <ConfigurationName>*

The <*ConfigurationName*> attribute is replaced with the real name of the configuration.

6.8 System Items

The local dataFEED OPC Suite application provides a set of system items providing information about the internal states of some dataFEED OPC Suite functionalities.

The provision of the different system items can be enabled or disabled in the dataFEED OPC Suite Configurator and is described in detail in Configuration: Settings - Advanced.

General System Items

Item ID	**Data Type**	**Access Rights**	**Description**

© Softing Industrial Automation GmbH

_System.General .Heartbeat	UI4	read	With this item, a client can determine via the OPC notification mechanism whether the corresponding local dataFEED OPC Suite application is still running. The value of this item is incremented by one after each read access. It is initialized with 0 when the local dataFEED OPC Suite application is started.
_System.General .SessionID	UI4	read	From this value, a client can determine that the local dataFEED OPC Suite application has been restarted. When the local dataFEED OPC Suite application is started, the bytes 6,5,4 and 3 of the current *FILETIME* are assigned to this value once (counting is from 0 to 7 with 7 as the most significant byte. This value is thus incremented by one every 1.7 seconds).
_System.General .LocalTime	DATE	read	This item provides the local current date-time using the operating system time.

Data Sources System Items

For each defined data-source connection a set of special items is predefined in the internal address space. Based on data-source type a specific group of system items is created. All connections are grouped under a node-based on its data-source types.

- **Siemens, Rockwell, Modbus or Mitsubishi data-source type:**

Item ID	Data Type	Access Rights	Description
_System.Data_Source.<DataSource>.<Connection Name>.Connected	BOOL	read	The item specifies the connection establishment between dataFEED OPC Suite and the corresponding configured PLC. **Note:** The connection is established only when a connected OPC Client is requesting its first read/subscribe operation on the corresponding connection. Without performing read/subscribe operations no connection to the PLC is initialized.
_System.Data_Source.<DataSource>.<Connection Name>.Status	UI4	read	The item returns the connection status between dataFEED OPC Suite and corresponding configured PLC. The following values are allowed to be returned: **Connection** | **Description**

Status	
0	Connection successfully established.
1	Bad connection parameters.
2	Maximum number of connections reached.
3	Connection does not exist.
4	Function not implemented.
5	Invalid connection handle.
6	No data available yet.
7	Waiting for acknowledge for last sent data.
8	Internal error.
9	Polling a non-existing job.
10	Driver not opened or invalid driver.
11	Network destination station overloaded.
12	Successfully received blocked data.
13	Invalid adapter or adapter does not exist.
14	Connection job already running.
15	Function is not supported by corresponding PLC.
16	Temporary not enough resources. Try again later.
17	No memory available.
18	Industrial Ethernet Signature not received or invalid.
19	Data error.
20	Logical protocol error.
21	Timeout.
22	Connection refused by PLC.

© Softing Industrial Automation GmbH

			<table><tr><td>23</td><td>Remote station or gateway sent a device specific error.</td></tr><tr><td>24</td><td>Connection closed forever. It will not be re-established automatically,</td></tr><tr><td>25</td><td>Forbidden IP Port.</td></tr><tr><td>26</td><td>Connection broken peer station.</td></tr><tr><td>27</td><td>Connection established after timeout.</td></tr></table>
_System.Data_Source.<DataSource>.<ConnectionName>.ServerCycle	UI4	read	The special item ServerCycle shows the number of complete passes of the server for a connection.
_System.Data_Source.<DataSource>.<ConnectionName>.MaxInterval	I4	read	The special item MaxInterval is used to access the measured Maximum Update Interval (in milliseconds) of all items of a corresponding connection. This value refers to the last complete polling cycle.
_System.Data_Source.<DataSource>.<ConnectionName>.ItemCount	UI4	read	The special ItemCount item is used to access the number of active items of the corresponding connection. This value is also indicated in the status window under the heading “Items.”
_System.Data_Source.<DataSource>.<ConnectionName>.ErrorCount	UI4	read	The special ErrorCount item is used to access the number of all active, erroneous items of the applicable Access Path. If the connection is in a “bad” state, all items have errors (i.e., the ItemCount item corresponds to the ErrorCount item).
_System.Data_Source.<DataSource>.<ConnectionName>.IpAddress	STRING	read	The special item IpAddress is used to access the target IP address of the applicable Access Path.
_System.Data_Source.<DataSource>.<ConnectionName>.PollingInterval	UI4	read	The special item Poll Interval is used to access the current poll interval. The current poll interval is indicated in milliseconds. A negative value indicates that polling was disabled. In this case, the last entered positive value remains on the display.

<table>
<tr><td></td><td></td><td></td><td>Note: This system item is not applicable for Siemens Send/Receive connections.</td></tr>
<tr><td>_System.Data_Source.<DataSource>.<ConnectionName>.PLCStatus</td><td>UI4</td><td>read</td><td>The special Item PLCStatus is used to indicate the status of an S7 CPU. With S7 connections and HPP switched on, it can be registered.
The item is read every 5 seconds from the PLC. If something is written to this item, it is read immediately.
Note: This is available only for Siemens S7 connections.
The following values are allowed to be returned:
<table>
<tr><th>PLC Status</th><th>Description</th></tr>
<tr><td>Unknown</td><td>PLC does not provide this information or it can not be read.</td></tr>
<tr><td>Start</td><td>Starting.</td></tr>
<tr><td>Run</td><td>Running.</td></tr>
<tr><td>Stop</td><td>Stopped.</td></tr>
</table></td></tr>
<tr><td>_System.Data_Source.<DataSource>.<ConnectionName>.BrowsingReady</td><td>BOOL</td><td>read</td><td>The special Item BrowsingReady is used to indicate when the browse operations are available of dataFEED OPC Suite. Basically this is done after the initialization of all Rockwell defined connections.
Note: This is available for Rockwell CompactLogix / ControlLogix connections.</td></tr>
</table>

- **B&R data-source type:**

Item ID	Data Type	Access Rights	Description
_System.Data_Source.<DataSource>.<ConnectionName>.Connected	BOOL	read	The item specifies the connection establishment between dataFEED OPC Suite and corresponding

© Softing Industrial Automation GmbH

			configured B&R device.
_System.Data_Source.<DataSource>.<ConnectionName>.DeviceIPAddress	STRING	read	The item specifies the IP address of the target station (IP address of the PLC) or the DNS name of the target station is entered.
_System.Data_Source.<DataSource>.<ConnectionName>.PVIIPAddress	STRING	read	The item specifies the IP address of configured PVI Manager.
_System.Data_Source.<DataSource>.<ConnectionName>.ItemCount	UI4	read	The special ItemCount item is used to access the number of active items of the corresponding connection. This value is also indicated in the status window under the heading “Items.”
_System.Data_Source.<DataSource>.<ConnectionName>.ErrorCount	UI4	read	The special ErrorCount item is used to access the number of all active, erroneous items of the applicable Access Path. If the connection is in a “bad” state, all items have errors (i.e., the ItemCount item corresponds to the ErrorCount item).

- **OPCServer or Tunnel Server data-source type:**

Item ID	Data Type	Access Rights	Description
_System.Data_Source.<DataSource>.<ConnectionName>.Connected	BOOL	read	The Boolean value of this system item shows if there is currently a connection to the corresponding server or not.
_System.Data_Source.<DataSource>.<ConnectionName>.ProvideItems	BOOL	read	This item returns the Provide Items attribute of the corresponding server connection as a string.
_System.Data_Source.<DataSource>.<ConnectionName>.URL	STRING	read	This item returns the associated connection URL of the corresponding server connection.
_System.Data_Source.<DataSource>.<ConnectionName>..ItemPrefix	STRING	read	This item returns the configured item prefix (i.e. connection name) of the corresponding server connection. This item is only provided if an item prefix / connection name has been configured.

- **Local Items data-source type:**

Item ID	Data Type	Access Rights	Description

© Softing Industrial Automation GmbH

_System.Data_Source.<DataSource>.<ConnectionName>.LocalItemsWatchdog	UI4	read and write	A client that acts logically as the OPC server has to cyclically write to the LocalItemsWatchdog. Within the time interval specified in the configuration under Local Items - Advanced Settings, the local dataFEED OPC Suite application checks whether the watchdog has been written to. If it has not been written to, the qualities of all items are set to BAD (Communication Failure). Watchdog monitoring can be enabled in the Local Items - Advanced Settings with the Activate Local Items Watchdog configuration attribute. Without having this configuration attribute set, the watchdog is disabled.

Store and Forward System Items

Item ID	Data Type	Access Rights	Description
_System.Data_Processing.StoreAndForward.FileLoadPercentage	R8	read	An OPC or Tunnel client can

			use this value in order to determine how many of the pre-configured storable data changes have already been stored in the local data storage. It represents the percentage of the stored data changes with a value range from 0 to 100.
_System.Data_Processing.StoreAndForward.OverWritingStarted	BOOL	read	An OPC or Tunnel client can use this value to determine if the maximum number of stored transactions has been reached and if the overwriting of older values has been started.
_System.Data_Processing.StoreAndForward.Stored Transactions	I4	read	An OPC client can use this value in order to determine how many transactions are currently stored in the local storage file.
_System.Data_Processing.StoreAndForward.ExceededMaxSize	BOOL	read	Returns true when the Max. size specified for the local storage file has been exceeded (but the Max. data changes limit has not been reached). The warning is

© Softing Industrial Automation GmbH

			generated but the Store And Forward plug-in continues to write data values to the file while the Max. data changes limit is not reached.

The Store And Forward system items are only enabled if the Store And Forward functionality is enabled for the corresponding local dataFEED OPC Suite application.

6.9 OPC Tunnel Discovery Service

The OPC Tunnel Discovery Service is part of the Softing dataFEED OPC Suite suite. The purpose of this service is to keep track of all local dataFEED OPC Suite applications currently running on the computer and to provide this information to all dataFEED OPC Suite Configurators in the network for configuring a corresponding OPC Tunnel server connection.

The communication between the Configurator and the discovery service is done over TCP/UDP. The Discovery Service reports all running local dataFEED OPC Suite applications to the computer the request originally came from. No response is sent if there are no running local dataFEED OPC Suite applications.

Note: For configurations running on the local machine the IP address 127.0.0.1 is reported.

Attention: Please make sure that no other application interferes with TCP/UDP port ***53842*** on the machine where the OPC Tunnel Discovery Service is running or the service may not receive the corresponding discovery requests.

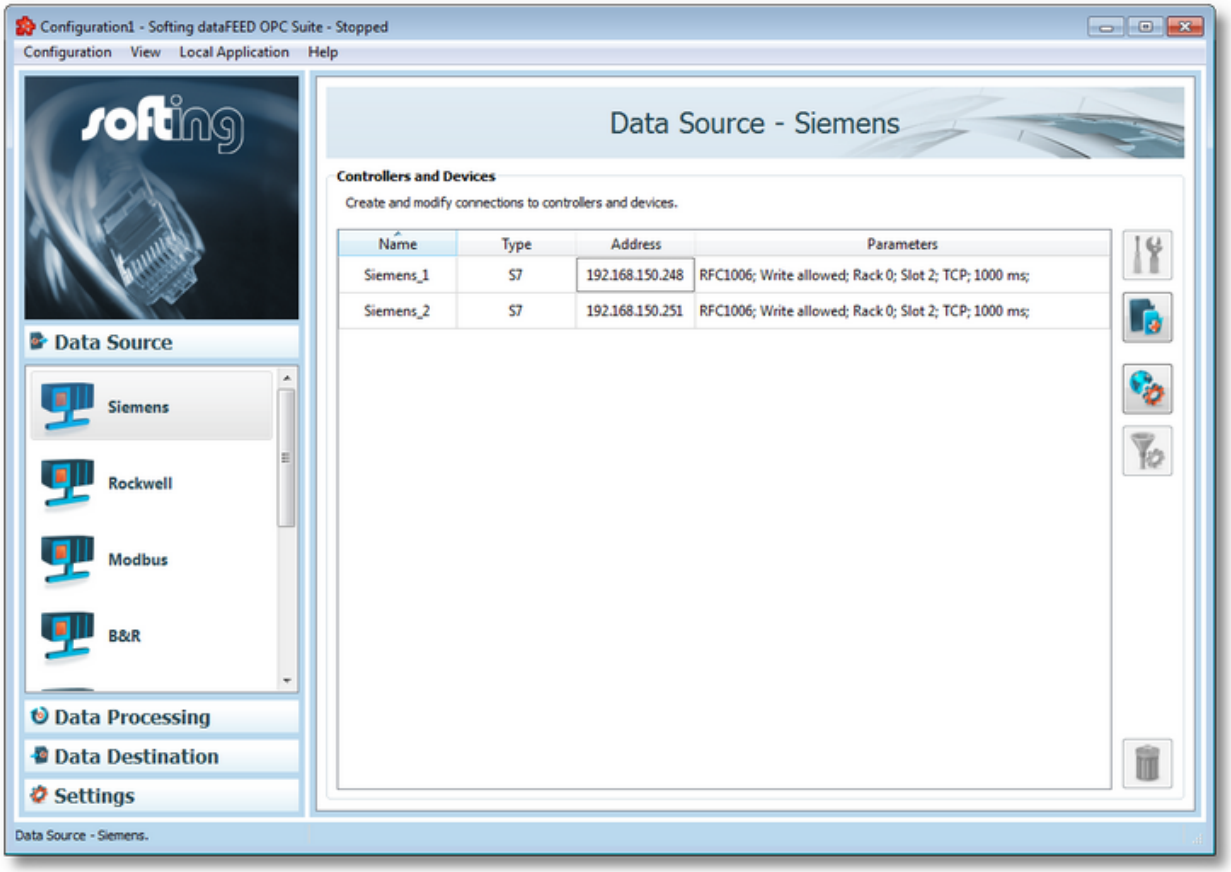
Attention: Please make sure that no other application interferes with TCP/UDP port ***53843*** on the machine where the dataFEED OPC Suite Configurator is running or the Configurator may not receive the corresponding discovery responses.

6.10 OPC UA Local Discovery Service

The OPC UA Local Discovery Service is part of the Softing dataFEED OPC Suite Suite. The purpose of this service is to keep track of all local dataFEED OPC Suite applications currently running on the computer and to provide this information to all OPC UA Clients in the network for the configuration of an OPC UA communication.

Each *dataFEED OPC Suite* configuration, which communicates via OPC UA, has to register at the Local Discovery Service. OPC UA Clients can get access to the corresponding *dataFEED OPC Suite* configuration via the Local Discovery Service. Therefore the previously configured security settings have to be considered.

7 Configuration

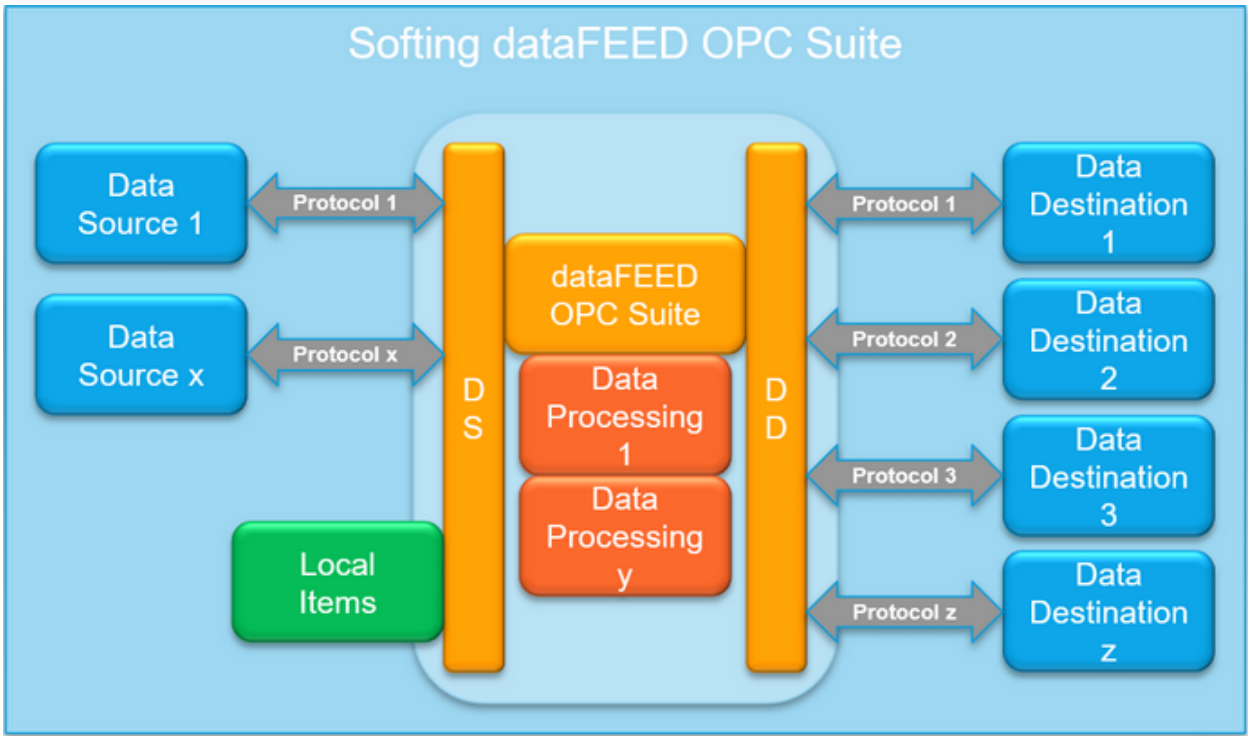


- Within this chapter of the online help the configuration of Softing dataFEED OPC Suite is described:

- Learn about the general configuration concept of Softing dataFEED OPC Suite @ Softing Configuration Concepts.
- See the general handling of the dataFEED OPC Suite configurator @ dataFEED OPC Suite Configurator.
- Find descriptions about the settings of the different function groups of Softing dataFEED OPC Suite @ dataFEED OPC Suite Configuration.

Softing dataFEED OPC Suite V5.38
© 2024 Softing Industrial Automation GmbH

7.1 Configuration Concept



The configuration concept of Softing dataFEED OPC Suite is mainly based on so called data sources and data destinations.

The different data source functionalities are responsible for building up the local namespace of the corresponding local dataFEED OPC Suite applications. This local namespace is organized in the form of an item tree with nodes, tags and properties as elements.

The different data destination functionalities use already existing data items from the local namespace – they do not introduce any data items to the local namespace.

Note: *Data Source* and *Data Destination* do ***not*** define a data flow direction. These terms only serve to indicate who is responsible for defining data items and who uses these already defined data items. In fact, the actual data flow is normally bidirectional, i.e. an OPC client would be a typical data destination using data items from the local namespace – however it generally can read from and write to these data items.

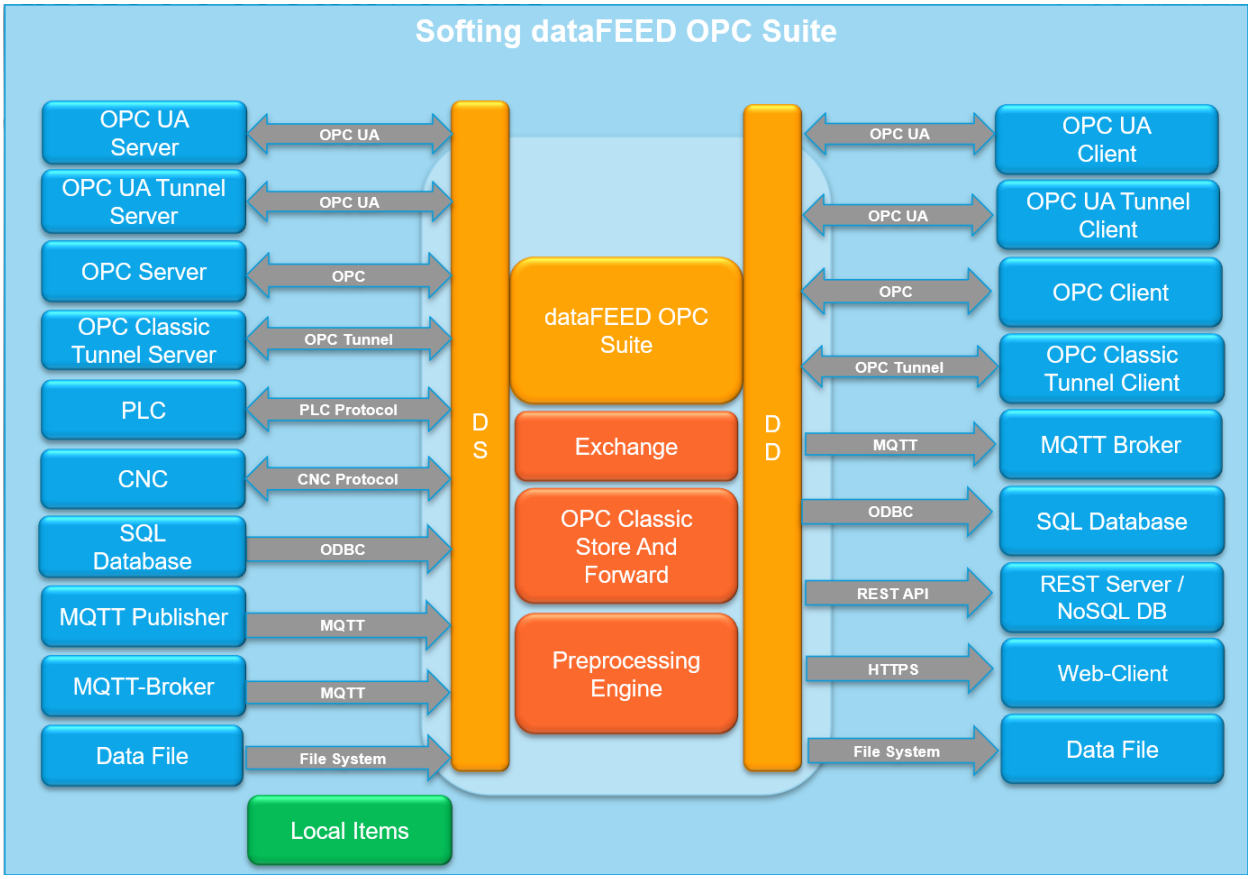
Data source entities are typically external entities like OPC or OPC Tunnel servers or external hardware devices. However, internal modules of the local dataFEED OPC Suite application, like Local Items, can also act as a data source entity. See dataFEED OPC Suite Data Sources for details.

Data destination entities are typically external entities like OPC or OPC Tunnel clients, SQL databases or data files. See dataFEED OPC Suite Data Destinations for details.

In addition, there are so called data processing functionalities which – like data destination functionalities – just use already defined data items from the local namespace and do some computations with the corresponding data values. Data processing entities are typically internal

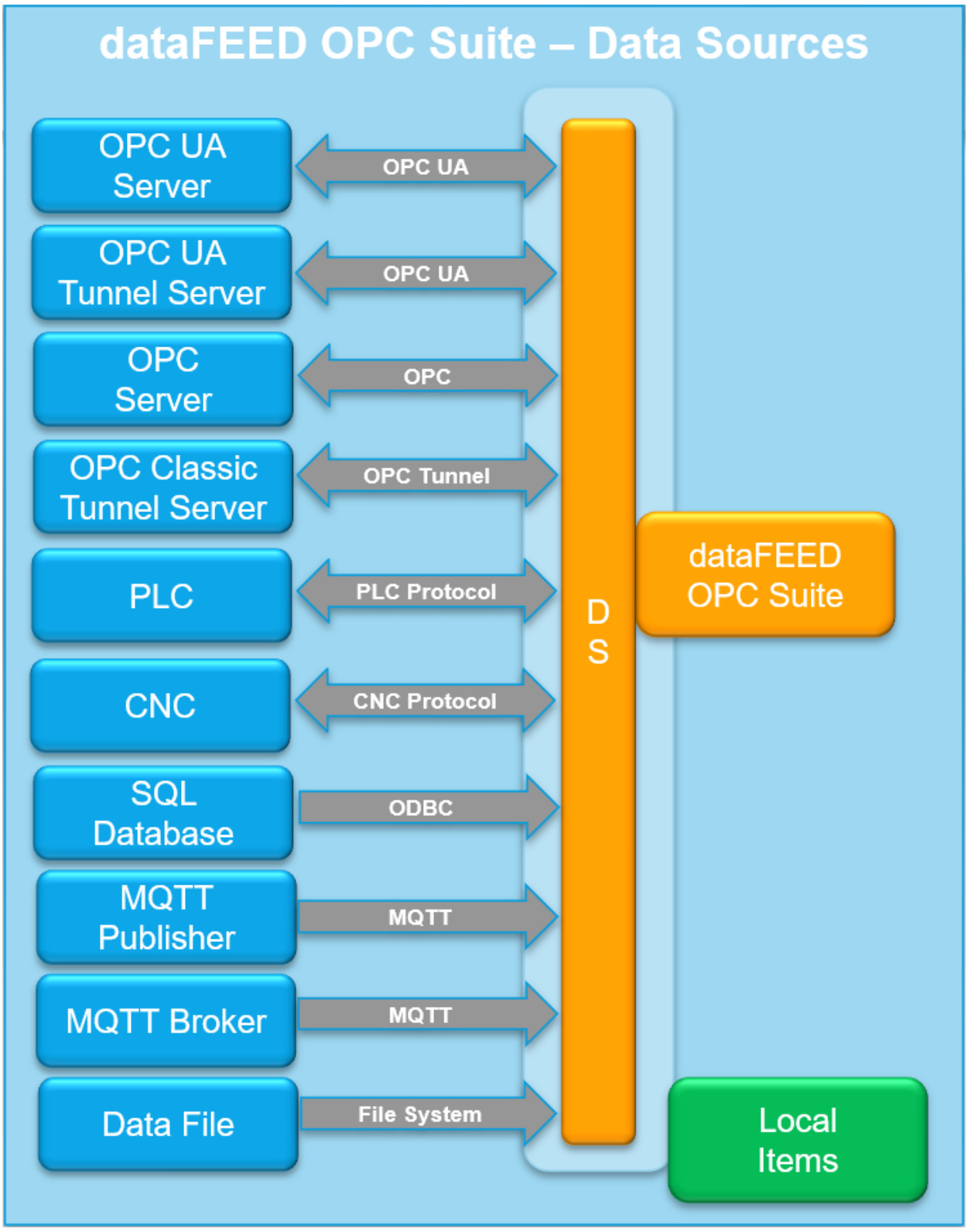
modules of the local dataFEED OPC Suite application like Exchange or Store And Forward. See dataFEED OPC Suite Data Processing for details.

Version V5.38 of Softing dataFEED OPC Suite supports the following data sources, data processing and data destination units:



Softing dataFEED OPC Suite V5.38
© 2024 Softing Industrial Automation GmbH

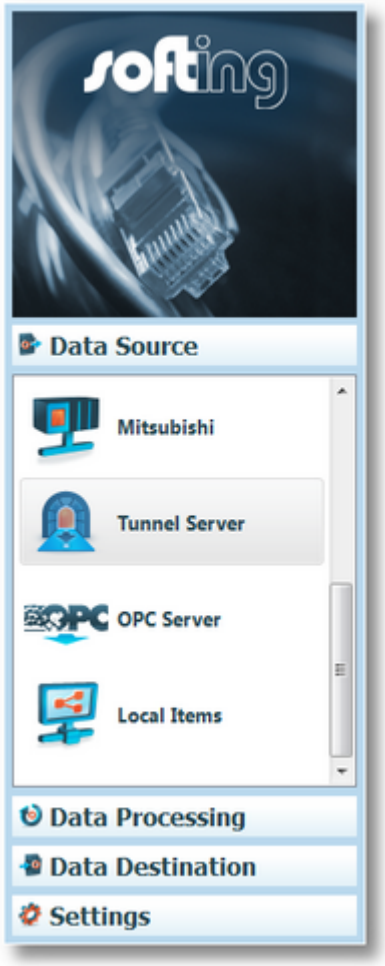
7.1.1 Data Sources



The different data source functionalities are responsible for building up the local namespace of the corresponding local dataFEED OPC Suite applications. This local namespace is organized in the form of an item tree with nodes, tags and properties as elements.

Note: *Data Source* and *Data Destination* do ***not*** define a data flow direction. These terms only serve to indicate who is responsible for defining data items and who uses these already defined data items. In fact, the actual data flow is normally bidirectional, i.e. an OPC client would be a typical data destination using data items from the local namespace – however it generally can read from and write to these data items.

Data source entities are typically external entities like OPC or OPC Tunnel servers or external hardware devices. However, internal modules of the local dataFEED OPC Suite application, like Local Items, can also act as a data source entity.

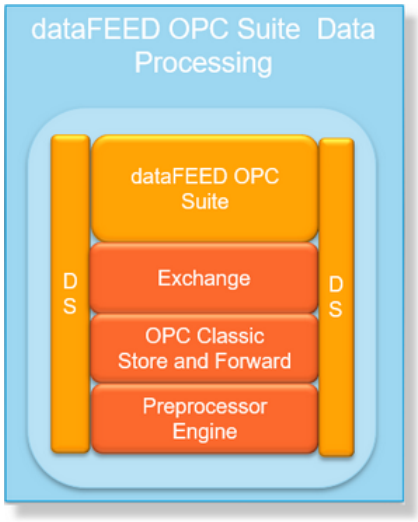


▼ Softing dataFEED OPC Suite - Data Sources

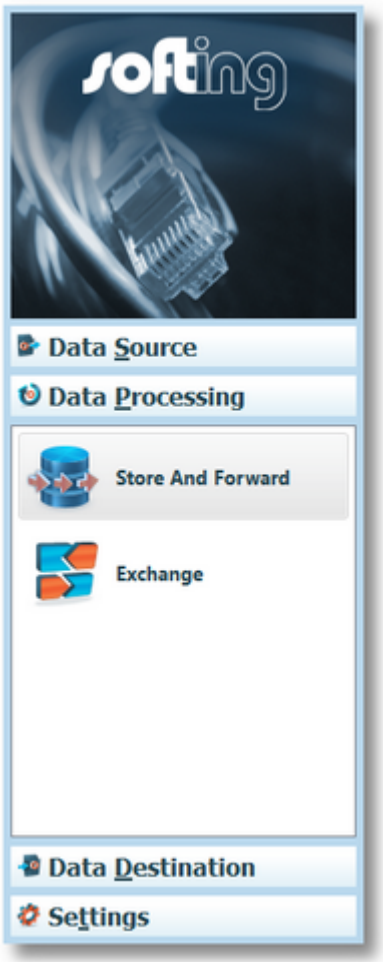
- Add connections to external OPC Tunnel servers @ Configuration: Data Source - Tunnel Server
- Add connections to external OPC servers @ Configuration: Data Source - OPC Server
- Exchange data between OPC clients @ Configuration: Data Source - Local Items

© Softing Industrial Automation GmbH

7.1.2 Data Processing



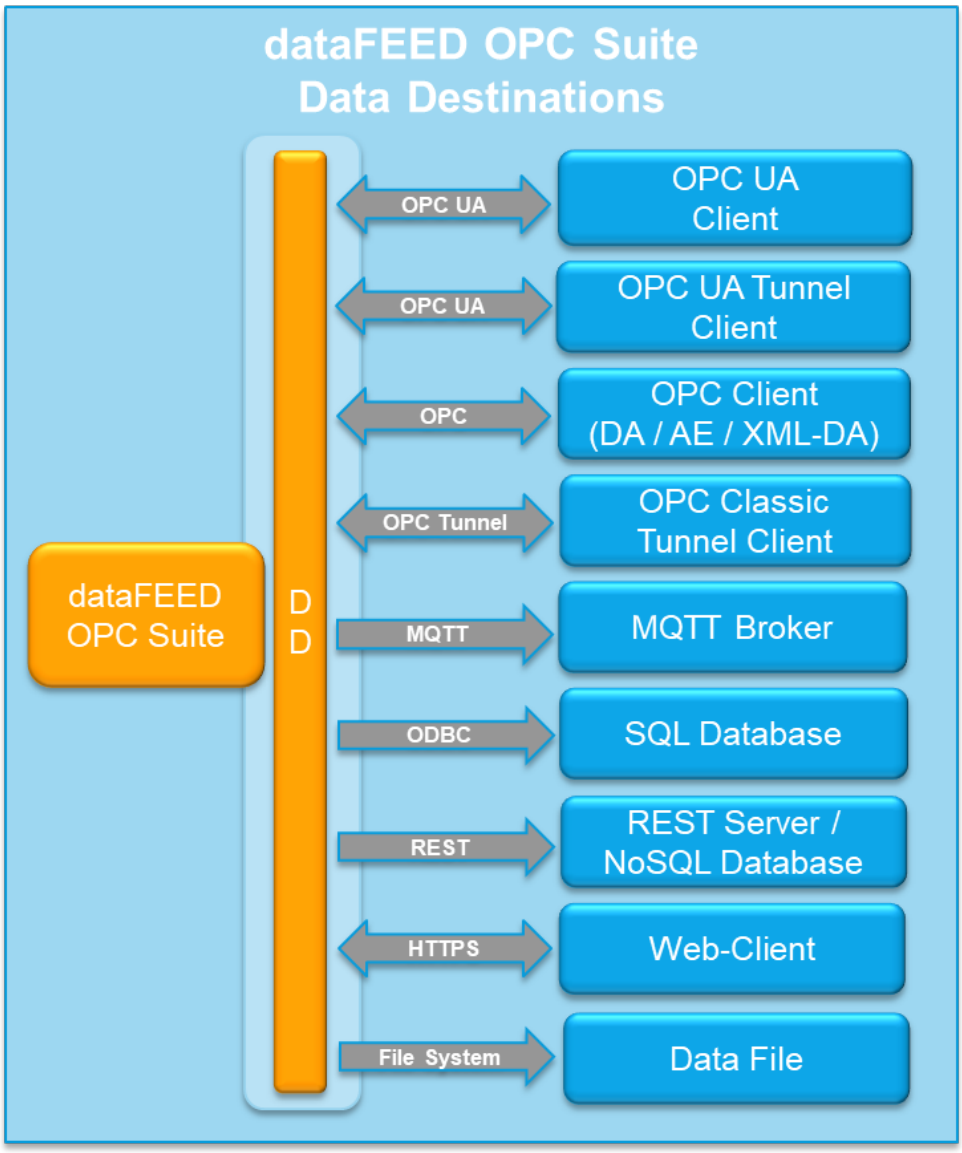
Additionally there are so called data processing functionalities which – like data destination functionalities – just use already defined data items from the local namespace and do some computations with the corresponding data values. Data processing entities are typically internal modules of the local dataFEED OPC Suite application like Exchange or Store And Forward.



▼ Softing dataFEED OPC Suite - Data Processing

- Protection against data loss @ Configuration: Data Processing - Store And Forward
- Exchange data between OPC servers @ Configuration: Data Processing - Exchange

7.1.3 Data Destinations



The different data destination functionalities use already existing data items from the local namespace – they do not introduce any data items to the local namespace.

Note: *Data Source* and *Data Destination* do ***not*** define a data flow direction. These terms only serve to indicate who is responsible for defining data items and who uses these already defined data items. In fact, the actual data flow is normally bidirectional, i.e. an OPC client would be a typical data destination using data items from the local namespace – however it generally can read from and write to these data items.

Data destination entities are typically external entities like OPC or OPC Tunnel clients, SQL databases or data files.

© Softing Industrial Automation GmbH



▼ Softing dataFEED OPC Suite - Data Destinations

- Access of external OPC Tunnel clients to the local namespace @ Configuration: Data Destination - Tunnel Client
- Access of external OPC clients to the local namespace @ Configuration: Data Destination - OPC Client
- Archive data to a database @ Configuration: Data Destination - Database
- Archive data to a file @ Configuration: Data Destination - File

7.2 dataFEED OPC Suite Configurator

▼ This section of the dataFEED OPC Suite Online Help describes the general handling of the dataFEED OPC Suite Configurator.

- How to use the Softing dataFEED OPC Suite Configurator @ Configurator - General.
- Manage dataFEED OPC Suite configurations @ Configuration Handling.
- Backup and restore configurations @ Backup & Restore.
- Start and stop the local dataFEED OPC Suite application @ Local Application Access.
- Manage data items within the tree controls of the dataFEED OPC Suite Configurator @ Tree Handling.

© Softing Industrial Automation GmbH

7.2.1 General

The Softing dataFEED OPC Suite Configurator consists of following elements:

- Main Window
- Menu Bar
- Navigator
- Configuration Section
- Status Bar

Main Window

The title bar of the dataFEED OPC Suite Configurator shows the Softing dataFEED OPC Suite icon.

This icon can have four different colors, reflecting the current local dataFEED OPC Suite application running state:

-  Icon Neutral : No configuration loaded.
-  Icon Red: The local dataFEED OPC Suite application for the loaded configuration is not running.
-  Icon Green: The local dataFEED OPC Suite application for the loaded configuration is running.
-  Icon Green/Red: The local dataFEED OPC Suite application for the loaded configuration is running without UA Server functionality .
-  Icon Green/Yellow: The local dataFEED OPC Suite application for the loaded configuration is running but the UA Server is in "initializing" state .
-  Icon Yellow: The local dataFEED OPC Suite application demonstration period for the loaded configuration has expired.

The main window icon is also used in the Windows taskbar to indicate the state of the currently loaded configuration.

The following text is be displayed in the dataFEED OPC Suite Configurator title bar:

<ConfigurationName> - Softing dataFEED OPC Suite - <OnlineState>

If no configuration is loaded then the text displayed in the title bar is:

Softing dataFEED OPC Suite

Online State	Description

© Softing Industrial Automation GmbH

Stopped	The local dataFEED OPC Suite application for the loaded configuration is not running.
Running	The local dataFEED OPC Suite application for the loaded configuration is running.
Demo Mode Expired	The local dataFEED OPC Suite application demonstration period for the loaded configuration has expired.

These states correspond to the respective application icon states described above.

If the currently loaded configuration has been modified and not yet saved, an asterisk (*) is added at the end of its name in the title bar text.

Menu Bar

The dataFEED OPC Suite Configurator main window shows a menu bar. The menu entries displayed in the menu bar and the corresponding submenus depend on whether the Configurator is configured to handle multiple or single dataFEED OPC Suite configurations.

For some menu entries, specific icons – typically used within common Windows applications – are displayed next to the corresponding menu entries (actions).

Also, for the menu entries corresponding to common Windows actions, straightforward shortcuts are displayed and supported (e.g. New, Save, Copy, Paste etc).

All the menu entries (main menu bar, menu actions) display and implement unambiguous accelerators. The corresponding shortcut letters are highlighted when pressing the ALT modifier key.

The main menu contains the following entries:

- Configuration
- View
- Local Application
- Help

The following section provides a short description of each of the above menu entries.

Configuration

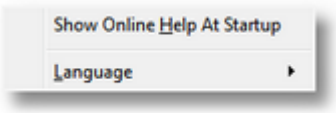
The *Configuration* menu entry contains actions for managing the configurations.

The Configuration Handling page offers a detailed description of this menu.

View

The *View* menu entry allows quick-configuring the following:

- have the dataFEED OPC Suite Configurator show the present Online Help at startup.
- select the display language for the entire dataFEED OPC Suite Configurator user interface.



An additional submenu appears when the *Language* menu entry is selected. There you can choose one of the currently supported languages (e.g. *English*, *German*, etc). A check mark is displayed next to the active language.

At the very first start of the dataFEED OPC Suite Configurator the display language is the same as the one used during the installation process.

Note: The chosen language is applied immediately without the need to restart the application.

Local Application

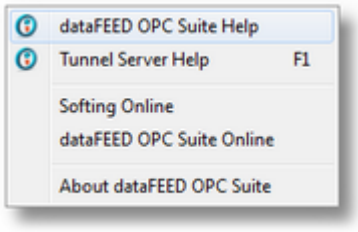
The *Local Application* menu entry contains actions which can be performed on the local dataFEED OPC Suite application instance for the current configuration.

The Local Application Access page offers a detailed description of the available actions.

Help

The Help menu entry allows quick access to this help file, which can also be accessed at any time by pressing the *F1* key (the F1 key usually opens the help page associated with the active page). In addition, this menu also contains useful links to the Softing dataFEED OPC Suite website.

The corresponding menu actions are presented below:



Action	Description
dataFEED OPC Suite Help	Opens this help file.

© Softing Industrial Automation GmbH

<*ConfigurationSection*> Help F1	Opens this help file at the current <*ConfigurationSection*> page. This entry is only available if a configuration is currently open in the Configurator.
Softing Online	Opens the Softing web page in the default web browser.
dataFEED OPC Suite Online	Opens a web page with useful information related to dataFEED OPC Suite.
About dataFEED OPC Suite	Shows the *About* dialog where the current application version number can be found.

Note: The <*ConfigurationSection*> *Help* submenu entry is only available if a configuration is currently open. The *F1* quick-access key is associated with this entry. If this submenu entry is not provided, the quick-access key will open the dataFEED OPC Suite Help instead.

Navigator

The navigator control on the left side of the Configurator main window displays the different configuration groups for the currently loaded Softing dataFEED OPC Suite configuration. The navigator can have up to four sections providing group related configuration options.

The Configuration Concepts chapter comprises more details on all the configuration groups.

Configuration Section

The main part of the main window represents the configuration page for the selected navigator add-on. This section is the most used one when configuring various settings for different configurations.

Status Bar

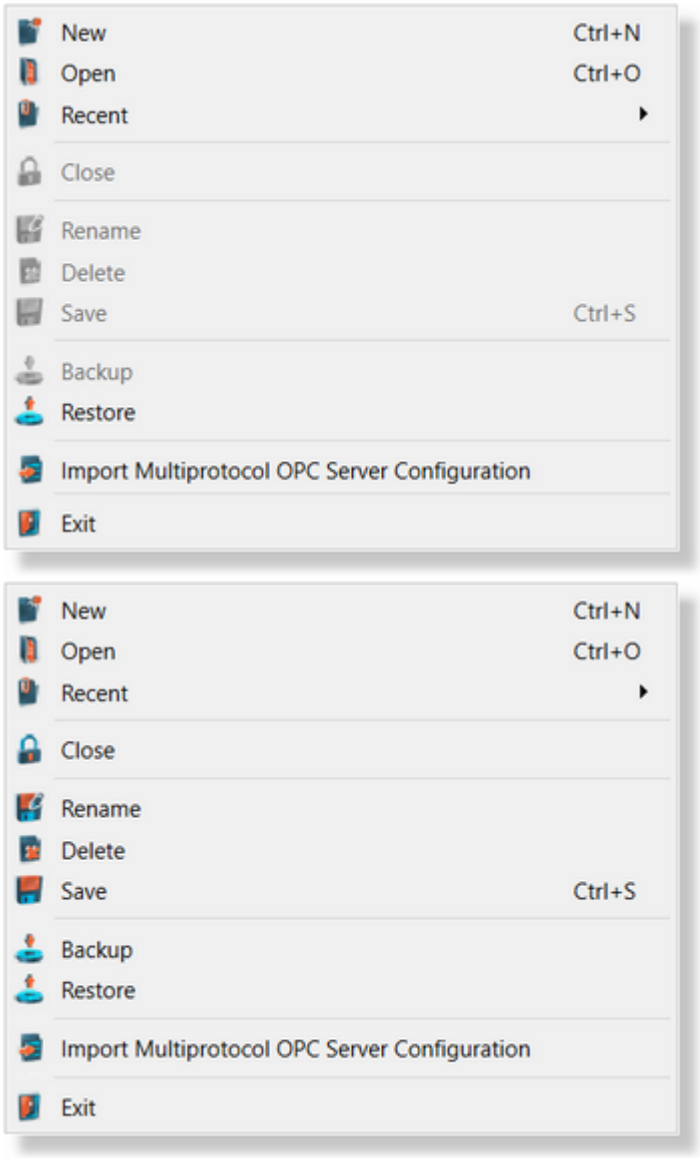
The main window includes a status bar where relevant information is displayed. The status bar is divided into two sections, one for each part of the main window: the navigator section (left side) and the configuration section (center and right side). The navigator status bar provides information related to the currently selected configuration section. The configuration status bar offers different information depending on the selected controls and items in the configuration section.

7.2.2 Configuration Handling

The *Configuration* main menu entry provides actions for handling Softing dataFEED OPC Suite configurations.

Through this main menu the following actions can be accomplished: creating, opening, closing or saving configurations, renaming or deleting configurations, backing up and restoring configurations, importing older configurations or exiting the application.

Various entries in the *Configuration* main menu may also be disabled depending on whether a configuration is loaded in the dataFEED OPC Suite Configurator. An example of both menus can be found below (left-side: no configuration is loaded; right-side: configuration loaded):

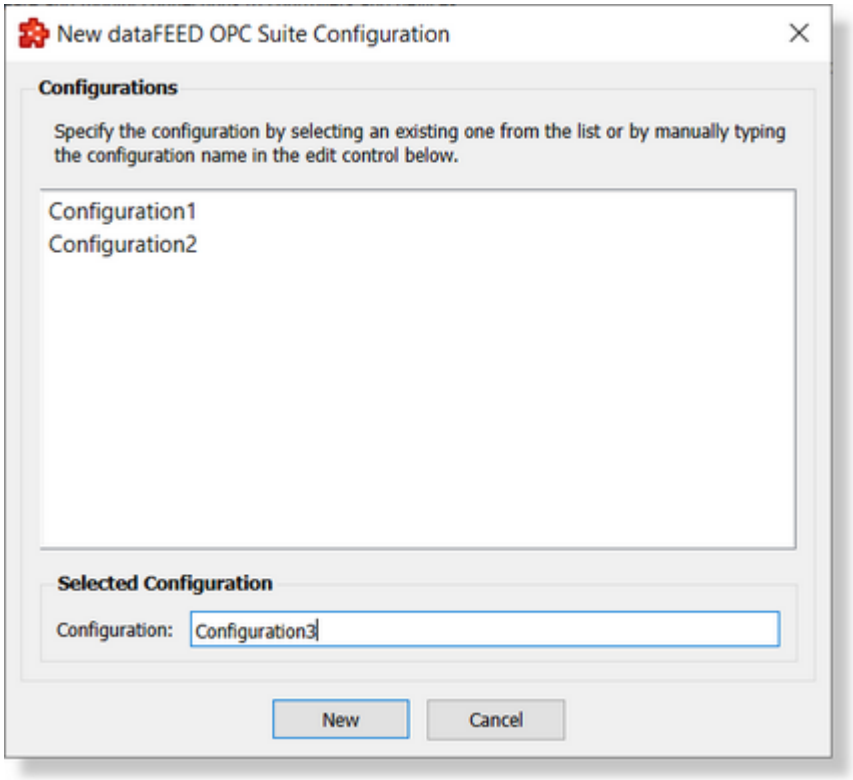


The following section describes each of the above menu entries.

New

The *New* menu entry allows creating new configurations. When this entry is selected the *New dataFEED OPC Suite Configuration* dialog is displayed:

© Softing Industrial Automation GmbH

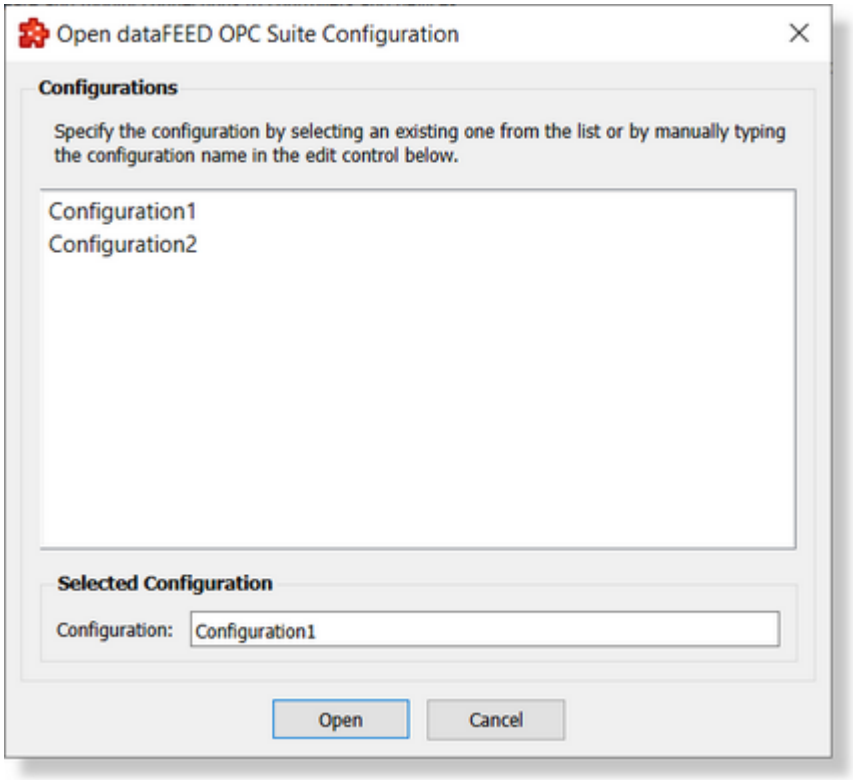


The window provides a list of already existing configurations and allows overwriting one of them. In this case a confirmation dialog will appear asking whether to overwrite the selected configuration or not. The configuration name can be typed (or modified) manually in the input box provided under *Selected Configuration*. A configuration name which is not already in use is suggested by default.

Attention: The configuration name must not contain the following characters: blank characters (e.g. the space character), slash / (also known as solidus), backslash \ (also known as reverse solidus), colon :, asterisk *, quotation marks ' and " (first sign also known as apostrophe), angular brackets < and > (also known as less-than-sign and greater-than-sign), vertical line | (also known as binary or sign).

Open

The *Open* menu entry allows loading existing configurations. When this entry is selected, the *Open dataFEED OPC Suite Configuration* dialog is displayed:



The name of the configuration to be opened can either be selected from the list in the *Configurations* area or typed in the *Configuration* input box under *Selected Configuration*.

Recent

The *Recent* menu entry allows quick access to the last used configurations. A maximum number of 10 configurations is displayed in this list. These configurations are sorted by the time the configuration was loaded, with the most recently loaded configuration at the top.

This menu entry can be used as a convenient way to open a recently used configuration by just clicking the corresponding entry in the list, without the need to browse through all the existing configurations.

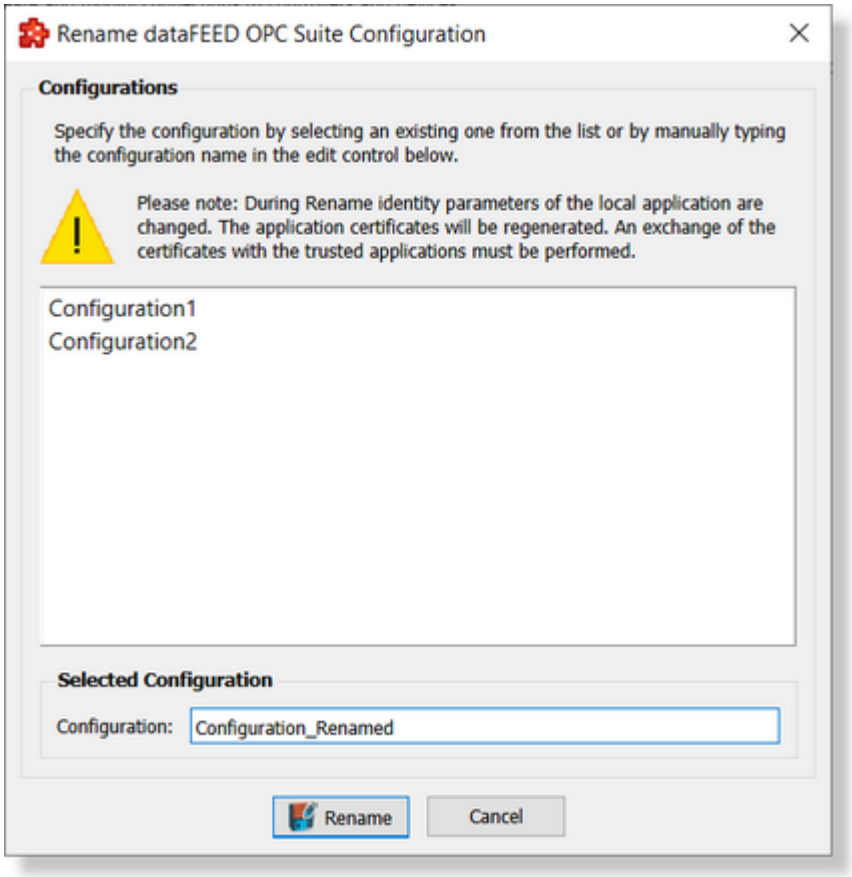
Close

The *Close* menu entry closes the currently loaded configuration. If the loaded configuration has been modified and not saved, a message box will appear asking whether to save the changes.

Rename

The *Rename* menu entry allows changing the name of the currently loaded configuration. When this entry is selected, the *Rename dataFEED OPC Suite Configuration* dialog is displayed:

© Softing Industrial Automation GmbH



The dialog provides a list of already existing configurations, but does not allow overwriting any of them. The current configuration name is automatically populated in the Configuration input box under *Selected Configuration* and can be manually modified.

Attention: The configuration name must not contain the following characters: blank characters (e.g. the space character), slash / (also known as solidus), backslash \ (also known as reverse solidus), colon :, asterisk *, quotation marks ' and " (first sign also known as apostrophe), angular brackets < and > (also known as less-than-sign and greater-than-sign), vertical line | (also known as binary or sign).

Attention: After a configuration is renamed, all the application certificates are regenerated due the reason that the configuration name is part of the mandatory identities on the own certificate. Due this reason, the trust relation with the other application should be redone.

Delete

The *Delete* menu entry allows deleting the currently loaded configuration.

Attention: The delete operation will fail if the local dataFEED OPC Suite application is running for the currently loaded configuration. An error message will be displayed informing you that the delete operation failed.

Attention: The *Delete* action always has to be confirmed by the user. Once deleted a configuration cannot be recovered any longer (it will not be disabled / unloaded / archived / etc., but physically deleted from the disk).

Save

The *Save* menu entry allows saving all the changes made to the currently loaded configuration.

Attention: If the local dataFEED OPC Suite application for the current configuration is running during the save operation, it must be restarted in order to apply the recent changes.

Backup

The *Backup* menu entry allows performing a complete export of the currently loaded configuration into a single backup file.

The Backup & Restore page offers a detailed description of this feature.

Restore

The *Restore* menu entry allows importing a previously exported configuration from a backup file.

The Backup & Restore chapter provides more details about this feature.

Import Multiprotocol OPC Server Configuration

The *Import Multiprotocol OPC Server Configuration* menu entry allows importing configurations created with Softing Multiprotocol OPC Server.

The OSF_CFG_Import_Multiprotocol_OPC_Server page offers a detailed description of this feature.

Exit

The *Exit* menu entry allows quitting the Softing dataFEED OPC Suite Configurator. Should there be any unsaved changes in the currently loaded configuration, a dialog box will appear asking whether to commit the changes.

7.2.3 Backup & Restore

The *Backup* and *Restore* features allow exporting and importing different configurations in order to migrate the server and all of its settings from one machine to another.

© Softing Industrial Automation GmbH

Backup

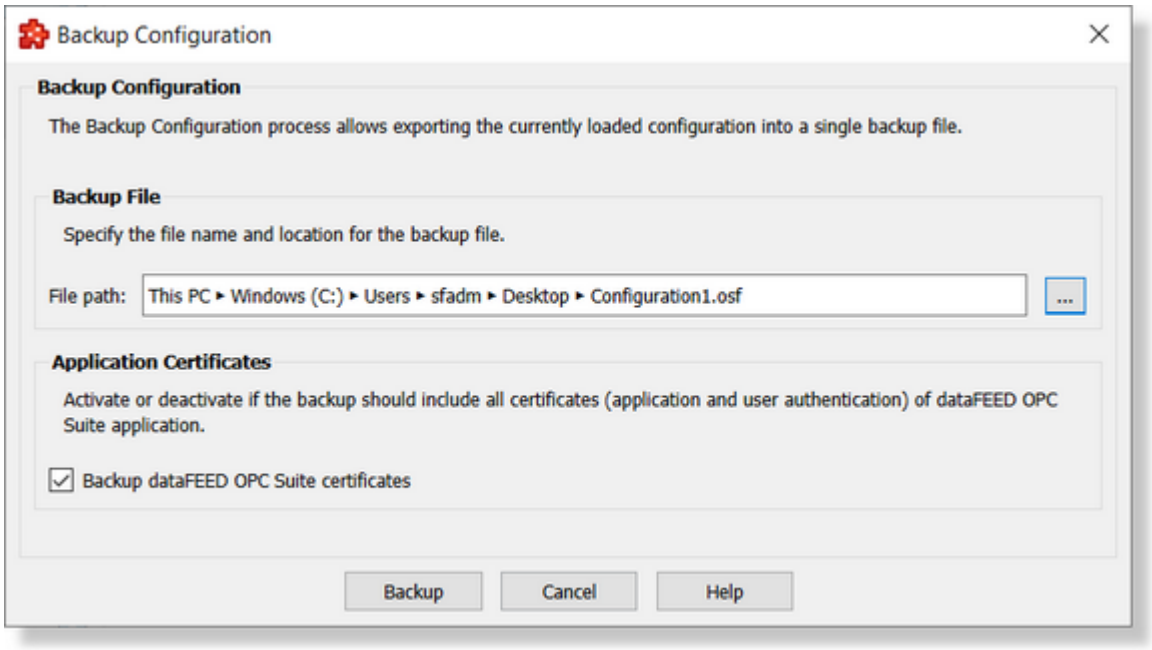
The *Backup* menu functionality performs a complete export of the currently loaded configuration into a single backup file.

The backup process can be started by clicking the corresponding entry on the *Configuration* main menu. The action is only available if a configuration is currently loaded.

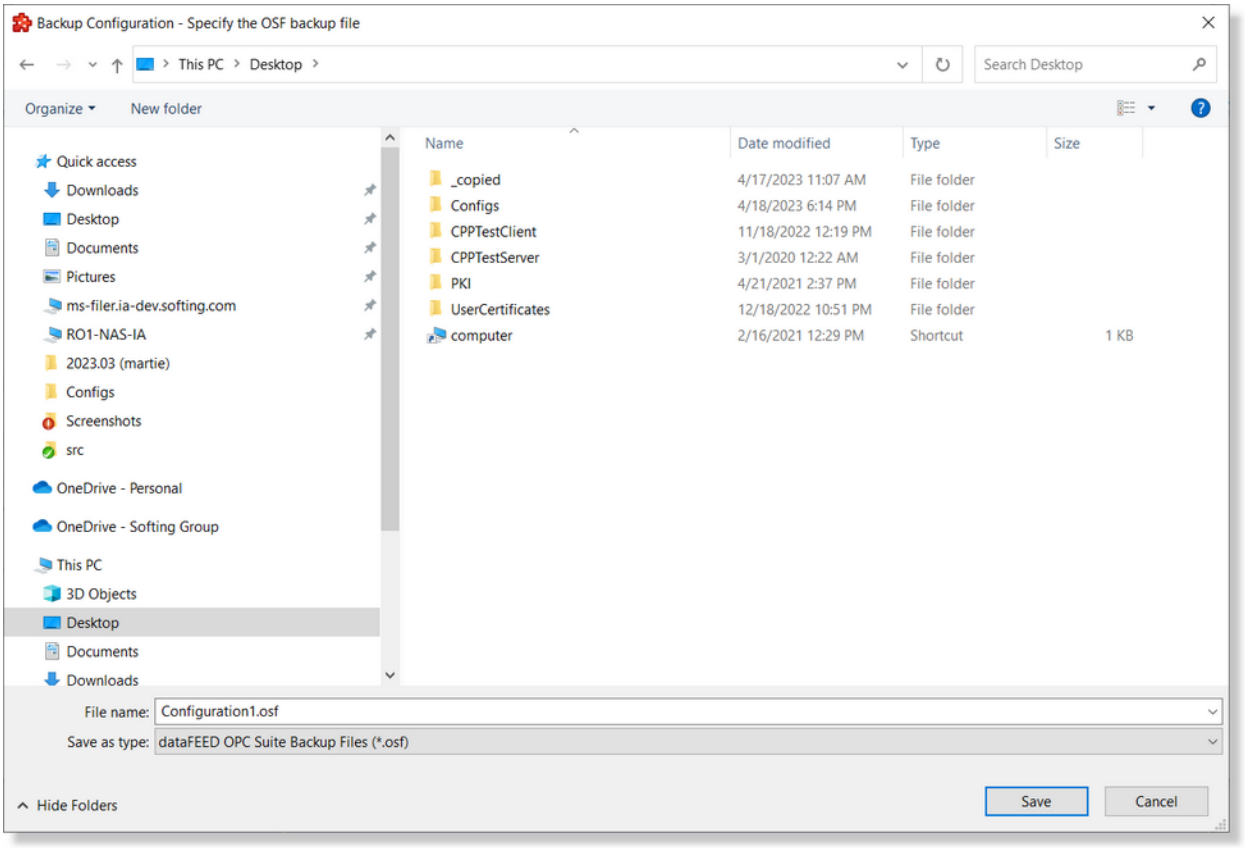
The *Backup Configuration* dialog allows the user to specify:

- The *File path*, initially displaying a path to the location last used by the user with the Configurator tool, and the current name of the configuration instance as the file name.
- The option to backup the dataFEED OPC Suite application and user authentication certificates.

The *Browse* button allows the user to specify the file path and the file name of the backup file.



A file save dialog appears when the *Browse* button is clicked, where the user can specify the location and the name of the backup file:



After specifying the custom file path and the file name in the file save dialog, when the *Save* button is clicked, the user is returned to the *Backup Configuration* dialog.

The user can activate or deactivate the option to include the dataFEED OPC Suite application certificate files in the configuration backup file.

The backup process can be completed by clicking on the *Backup* button.

The resulting backup file will be a binary file containing all the data relevant to the configuration.

Restore

The *Restore* feature allows importing a previously exported configuration from a backup file.

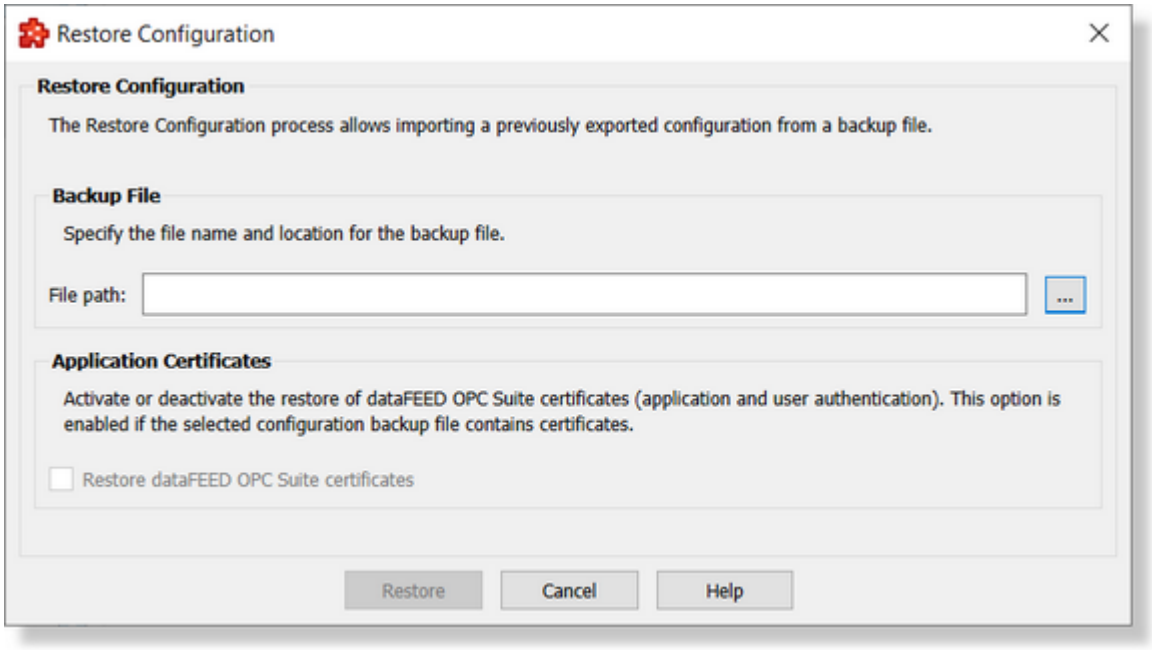
The restore process can be started by clicking the corresponding entry on the *Configuration* main menu.

The *Restore Configuration* dialog window allows the user to specify:

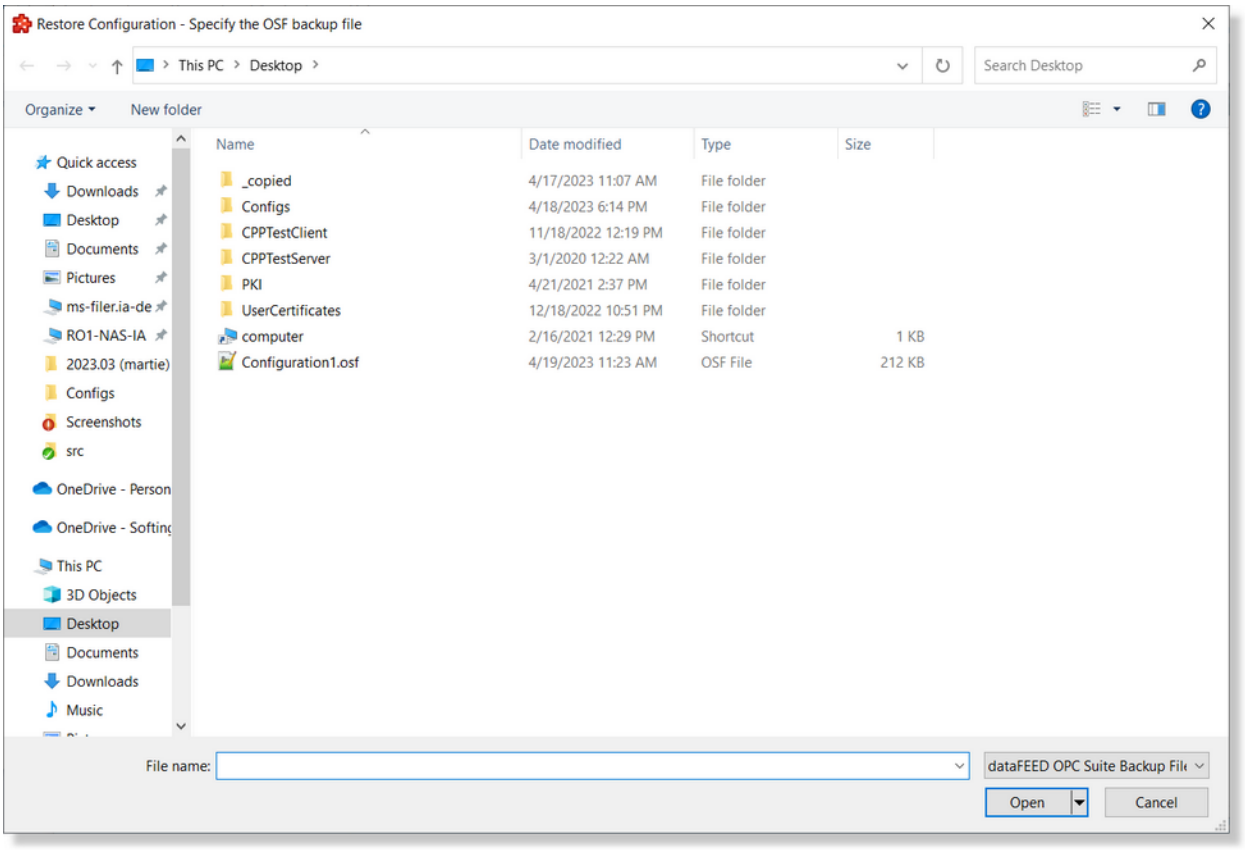
- The *File path*.
- The option to restore the dataFEED OPC Suite application certificates.

With the *Browse* button the user is directed to a file open dialog, where the backup file can be selected.

© Softing Industrial Automation GmbH

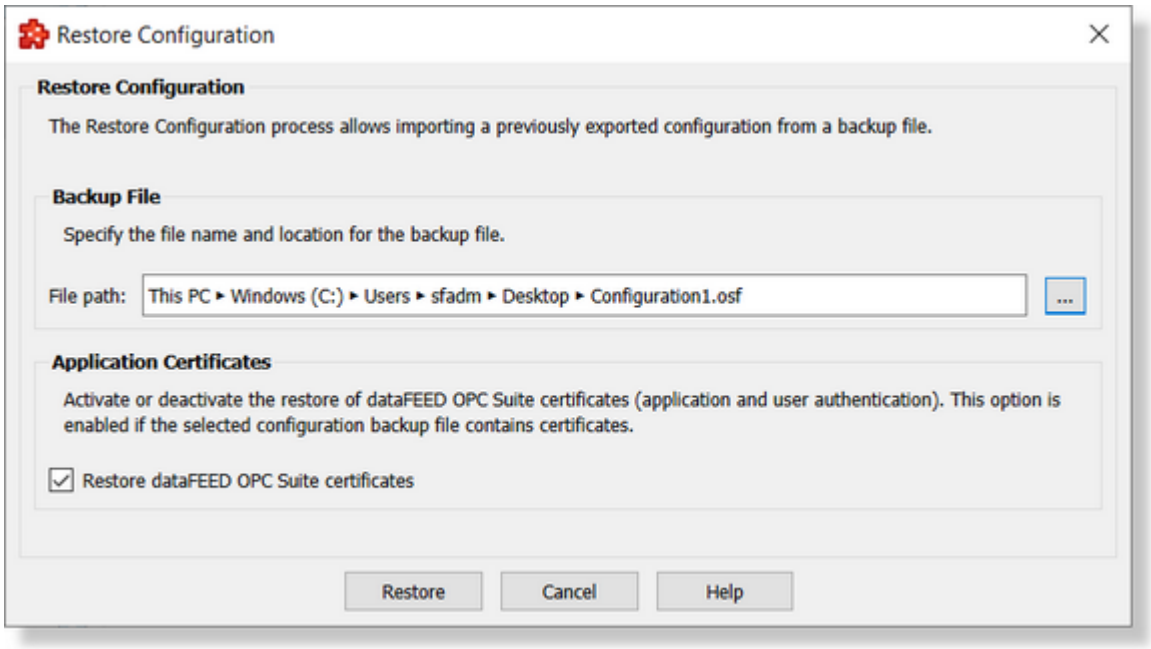


A file open dialog appears when the *Browse* button is clicked, where the user can select the backup file:



After selecting the backup file in the file open dialog, when the *Open* button is clicked, the user is returned to the *Restore Configuration* dialog.

The user can activate or deactivate the option to restore the dataFEED OPC Suite application certificate files from the configuration backup file. This option is enabled if the selected configuration backup file contains application and user authentication certificates.



The restore process can be completed by clicking on the *Restore* button.

If the selected file is a valid backup file, the restore procedure will start and the resulting configuration will be opened on completion.

7.2.4 Import Multiprotocol OPC Server Configuration

The *Import Multiprotocol OPC Server Configuration* menu item from the *Configuration* main menu allows importing configurations created with Softing Multiprotocol OPC Server into the current configuration.

See the Import Multiprotocol OPC Server Configuration chapter for more details about this feature.

7.2.5 Local Application Access

The dataFEED OPC Suite Configurator interacts with the local dataFEED OPC Suite application for monitoring the running state of the local dataFEED OPC Suite application corresponding to the current configuration and for performing basic control operations on it.

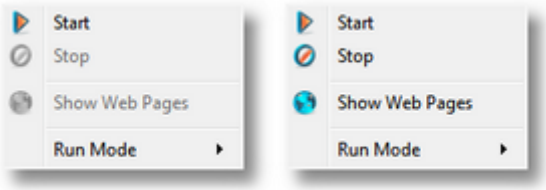
The Configurator always displays the running state of the local dataFEED OPC Suite application corresponding to the current configuration. This is done by updating the following:

- Configurator application title bar icon
- Configurator application title bar text

© Softing Industrial Automation GmbH

A detailed description of the running states and associated icons can be found in the Main Window section of the General page.

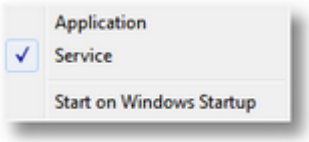
The *Local Application* main menu entry provides various actions to access the local dataFEED OPC Suite application corresponding to the currently loaded configuration. The available entries on the *Local Application* menu differ depending on whether the local dataFEED OPC Suite application is running (right side) or not (left side):



The available actions are described below:

Action	Description
Start	Starts/restarts the local dataFEED OPC Suite application for the currently loaded configuration.
Stop	Stops the local dataFEED OPC Suite application for the currently loaded configuration.
Show Web Pages	Opens the web pages of the currently loaded and running configuration. For more details about the Web Pages functionality see the Web Pages chapter in the *Operations* section of this help file.
Run Mode	Allows configuring how the local dataFEED OPC Suite application will run and whether it should start with Windows or not.

Run Mode menu item allows changing the run mode settings for the local dataFEED OPC Suite application. When the local dataFEED OPC Suite application is running all the following menu items are disabled. A description of each *Run Mode* menu item is provided in the table below:



Action	Description
Application	If selected the local dataFEED OPC Suite application is registered to run as an application for the currently loaded configuration.
Service	If selected the local dataFEED OPC Suite application is registered to run as a Windows service for the currently loaded configuration.

Start on Windows Startup	If selected the local dataFEED OPC Suite application will be started automatically on Windows startup. If the local dataFEED OPC Suite application run mode is set to *Application* and *Start on Windows Startup* is enabled, the application will not start until a user logs on. If the run mode is set to Service it may start before a user logs on.

Note: Application and Service run modes are mutually exclusive; only one of the entries can be selected at a time.

The local dataFEED OPC Suite application can also be accessed through the *Tray Application* menu as described under Tray Application in the *Operations* section of this help file.

7.2.6 Tree Handling

- This chapter covers the tree handling through the entire dataFEED OPC Suite Configurator application:

- Obtain general information about the namespace tree structure @ Tree Handling - General
- Access a detailed description of the namespace tree @ Tree Handling - Local Namespace
- Observe how the node state and icon change @ Tree Handling - Tree Node States and Icons
- Read about the tree view when the server connections are established @ Tree Handling - Online Mode
- Inspect the tree view when the server connections are down @ Tree Handling - Offline Mode
- Get details on where the tree items can be checked or unchecked inside the Store And Forward or filter pages @ Tree Handling - Checked States

7.2.6.1 General

The dataFEED OPC Suite tree refers to the OPC namespace tree structure visible from multiple pages of the Configurator: Filter Items, Configuration: Data Processing - Exchange, Configuration: Data Processing - Store And Forward, the Data Item page of the *Data Value Placeholder Wizard* and others. A similar tree is also provided by the local dataFEED OPC Suite application for the clients connected to it. The tree is referred to as "similar" because there are some exceptions in the local dataFEED OPC Suite application regarding the items under a specific server having the Item Prefix value set to: EmptyDA, EmptyAE or EmptyLI. These exceptions will be further discussed.

© Softing Industrial Automation GmbH

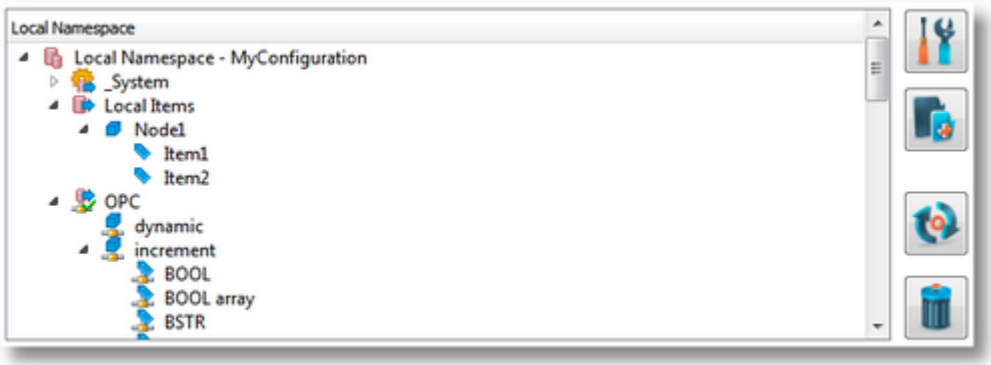
Namespace Structure

The namespace structure from the local dataFEED OPC Suite application is the same as the one on the Configurator page. It is structured as follows:

- The root element is named *Local Namespace - <ConfigurationName>* where *<ConfigurationName>* represents the name of the dataFEED OPC Suite configuration. It has an associated tool tip containing *Local Namespace Softing dataFEED OPC Suite*.
- The next level in the tree is composed of the *_System* node, the *Local Items* root node and the server connection's *Data Source Name* node for each server defined on the OPC Server and Tunnel Server pages. All of them have tool tips providing information about the associated connections.
- The next levels are specific to each type of parent tree node. For example the *_System* node has the following children: the *General* node (containing system tags related to OPC Server functionality:*Heartbeat, SessionID and LocalTime*), the *Data_Source* node (containing further information about the PLC connections, OPC and Tunnel Server statuses), the *Data_Processing* node (containing indicators for the *Store And Forward* status). *The Data_Source* node is only displayed if the *Activate connection items* checkbox is enabled in the *Global Application Settings* group box on the Configuration: Settings - Advanced page. Likewise the *Data_Processing* node is only displayed if the *Activate Store And Forward* checkbox is enabled on the Data Processing - Store And Forward page. When the *Local Items* root node is expanded the entire namespace of *Local Items* is displayed. This node is always shown and may have different names. For more information see Configuration: Data Source - Local Items. If other nodes representing the defined server connections are expanded, they will display the corresponding namespace of the server connection.
- The Tags and Properties items are at the lowest levels of the tree. Tool tips containing useful information on these items (Data Item, Data Type, Data Access or Server Path) are displayed.

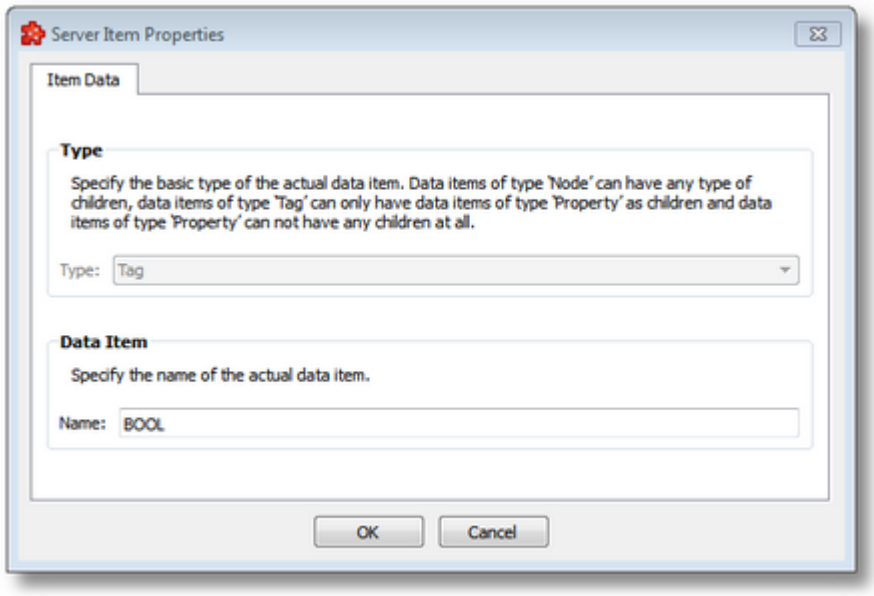
Note: The _System node is only displayed if the *Activate System Items* checkbox is enabled in the *Global Application Settings* group box on the Configuration: Settings - Advanced page.

The following image shows an example of an expanded tree from the Data Item page:



The buttons on the right side of the tree list vary depending on the page where the tree is located. The buttons allow performing the following actions:

- Change the properties of the tree item selected in the local namespace by using the *Properties*  button, pressing the keyboard shortcut CTRL+P or selecting the *Properties* action from the context sensitive menu (where available). The properties dialog is available depending on the selected item:
  - *Local Namespace* root node - the *Properties* action is disabled for this item.
  - A server connection node - the *Properties* actions is not available for this item either.
  - An item belonging to a static server connection (a Filter Item)
  - An item belonging to a dynamic server connection (not a Filter Item). The dialog displayed is similar to that of a static item, but only the item name can be modified:



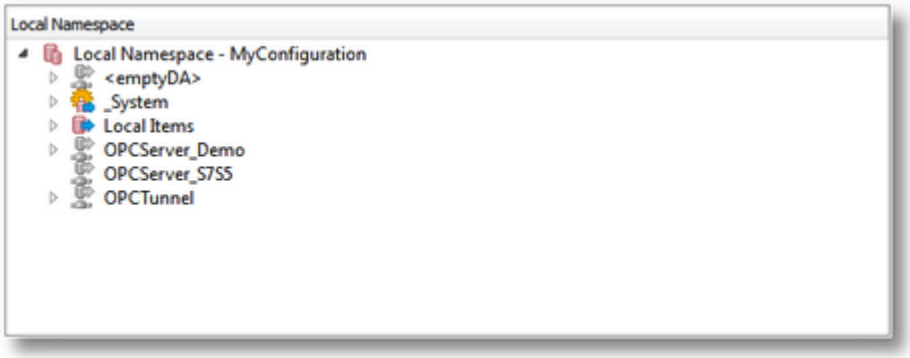
  - A pre-defined Local Item.
  - A System Item - the Properties action is disabled for this item.

- Add a new item to the local namespace by using the *Add*  button, pressing the keyboard shortcut CTRL+A or selecting the *Add* action from the context sensitive menu (where available).
- Refresh the local tree namespace by using the *Refresh*  button, pressing the keyboard shortcut CTRL+R or selecting the *Refresh* action from the context sensitive menu (where available). The entire namespace is recreated by browsing all defined server connections. If non-referred user-defined items exist in the configuration then the respective items are deleted from the namespace.
- Delete an item by using the *Delete*  button, pressing the DEL / DELETE key or selecting the *Delete* action from the context sensitive menu (where available). If an item is used in other add-ons it cannot be deleted.

© Softing Industrial Automation GmbH

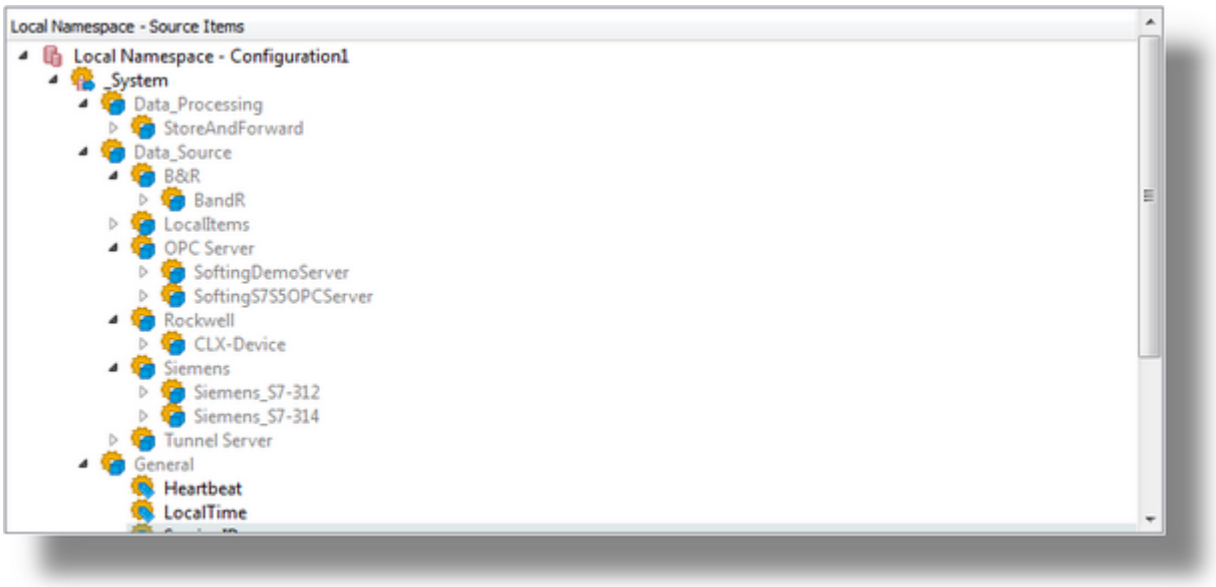
7.2.6.2 Local Namespace

The *Local Namespace* refers to the internal namespace of <%OEMPRODUCT% which is composed of the system items, the local items and the server connection specific items. Each of these separate groups will have its own node at the second level of the dataFEED OPC Suite namespace tree, right below the *Local Namespace* root node. The image below illustrates a simple structure of the tree namespace:



The _System Root Node

The *_System* node is located at the first level of the *Local Namespace* tree. It holds a sub-tree structure containing information about dataFEED OPC Suite *General* OPC items, defined *Data Sources* and configured *Data Processing* functionalities. This node is displayed by default as the *Activate System Items* and the *Activate Connection Items* checkboxes are initially enabled on the Configuration: Settings - Advanced page of the Configurator. An image of the expanded *_System* node is shown below:



These items do not provide a context menu and the buttons on the right are disabled. *Read* is the only allowed Data Access right for each of them**.** A detailed description of the _System items can be found on the dataFEED OPC Suite System Items page.

The Local Items Root Node

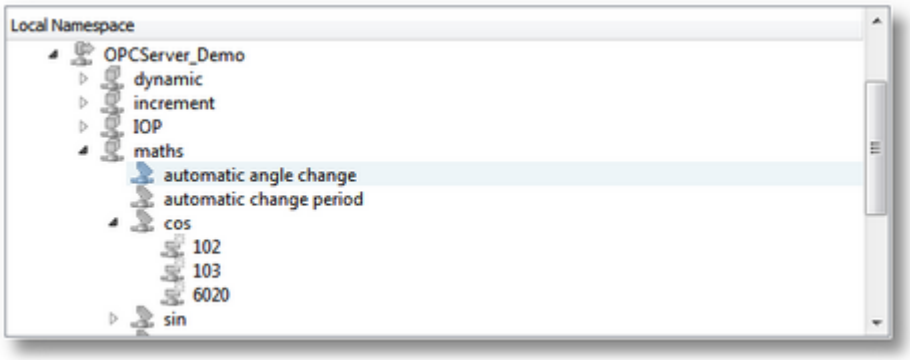
The *Local Items* root node is located at the first level of the *Local Namespace* tree, having the *_System* node as a sibling. It holds a sub-tree structure of the dataFEED OPC Suite internal storage items. A detailed description of their functionality can be found on the Configuration: Data Source - Local Items page. The Local Items node sub-tree is composed of nodes and tags having any data type supported by the server. The defined *Local Items* are saved in the *LocalItems.txt* file in the *LocalItems* folder under the respective configuration folder. Two specific context menus are available for this sub-tree, which will be described in the Context Menus section below. The following picture represents a simple *Local Items* tree:



The context sensitive menu also contains the actions *Copy* and *Paste*. The *Paste* action will only be active if something was previously copied. The paste operation will only be available for the Local Items node if the clipboard contains data copied from the same place. If the clipboard data does not belong to the Local Items node an error message will be displayed.

The Server Connection Root Node

The server connection specific nodes are located at the first level of the Local Namespace tree, having the *_System* and *Local Items* nodes as siblings. A node of this type holds an external server's namespace tree structure. This namespace can be configured either dynamically or statically. A detailed description of their functionality can be found in the Data Source - Tunnel Server chapter and the Data Source - OPC Server chapter. An example namespace of a Softing dataFEED OPC Suite Demo Server is displayed below:



© Softing Industrial Automation GmbH

The Server Connection root node name will always be the Item Prefix of that server connection. For the above example the root node name is *OPC_1* and it contains:

- Nodes: *dynamic, increment, maths*
- Tags: *automatic angle change, cos, sin*
- Properties: *102, 103, 6020*

The context sensitive menu also contains the actions *Copy*, *Paste* and *Connect and Refresh*. The *Paste* action will only be active if data was previously copied to the clipboard. The paste operation will only be available for a Server Connection node if the clipboard contains data copied from the same place. If the clipboard data does not belong to the Server Connection node an error message will be displayed.

The *Connect and Refresh* action will browse the server namespace and retrieve the latest structure. The *Local Namespace* tree will be updated accordingly. After the refresh operation, the corresponding sub-tree collapses and the internally cached data is destroyed.

Note: Any manually added items, nodes or properties will be automatically removed after a Connect and Refresh operation!

The Tag

The Tag nodes represent the items of a server namespace and can be located at the last level of the *Local Namespace* tree. These nodes represent the leaves of the Local Namespace tree and contain values and access rights. A tag may have certain properties which are represented as child nodes (this particular case prevents the tag nodes to be the leaves of the namespace tree). The picture below shows the tags of the *IOP.dynamic* node of the Demo Server:



The Property

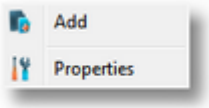
The property nodes are the leaves of the Local Namespace tree, as they are the child nodes of the tag items.The following picture shows the properties of the child nodes under the *IOP.dynamic* node of the Demo Server:



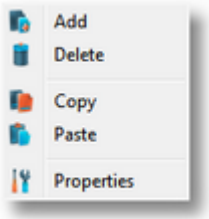
Tree Specific Context Sensitive Menus

The context menus vary depending on the currently selected tree item over which the context menu is opened. Another factor that affects the available actions on a context sensitive menu is the Windows clipboard, which if empty will cause the *Paste* action to be disabled. Taking into account the above information, the following section describes the different types of context menus:

- There are no context menus for the *Local Namespace* root node, for the *Alarms and Events* server connections and for the _System items nodes.
- The *Local Items* root node context menu has two entries, one for adding a new element to the Local Items tree and one that will open the *Properties* dialog for renaming the Local Items root node:



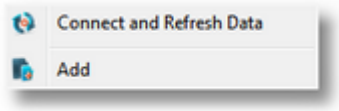
- The context sensitive menu that is displayed for other elements of the Local Items namespace (e.g. nodes or tags) initially contains four actions. The fifth action, *Paste*, is only displayed if something was previously copied to the clipboard. The *Delete* action will remove the selected element and all of its children after the delete action is confirmed. The *Copy* action will copy the element and its entire sub-tree to the clipboard while the *Properties* action will display the Local Items properties dialog, which allows changing the *Name*, the *Data Type* or the *Comment* attribute of the selected element:



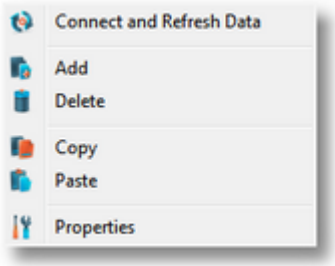
Note: The Type element cannot be changed. A node cannot be turned into a tag, or vice-versa.

© Softing Industrial Automation GmbH

- The Server Connections root node context menu has the following two entries: *Connect and Refresh Data* which executes a browse operation on the server connection, and *Add* which allows adding a new element to the namespace of the selected server connection. After the refresh operation, the corresponding server connection sub-tree collapses and the internally cached data is destroyed.



- The context menu displayed over items belonging to a server connection namespace (nodes, tags or properties) has five actions by default. The first two actions are described above (*Connect and Refresh Data*, *Add*). A sixth action is *Paste*, which is only visible if something has been previously copied to the clipboard. The *Delete* action will remove the selected element and all of its children after the delete action is confirmed. The *Copy* action will copy the element and its entire sub-tree to the clipboard while the *Properties* action will display the properties dialog, which provides different contents for static and dynamic server connections.



Note: The *Copy* action is disabled for items of the Property type. This is because a property cannot be copied separately, but only along with the parent node.

Certain rules apply to the *Copy*/*Paste* actions:

- It is not possible to copy data from the *Local Items* and paste it under a server connection. Although the *Paste* action will be enabled an error message will be displayed as the data does not have the required format. The same applies vice-versa: data copied from server connection nodes cannot be pasted under a Local Items node.
- It is not possible to copy from a dynamic server connection and paste the copied contents under a static server connection. The *Paste* context menu action will not be disabled but an error message will be shown if the rule is not obeyed.
- It is possible to copy data from a static server connection and paste it under a dynamic server connection.

7.2.6.3 Tree Node States and Icons

This page explains how the state of a node and its attached icon change depending on various factors. These factors only influence the server connections, and not the *_System* and *Local Items* nodes. There are two types of server connections:

- *Dynamic* server connections are server connections where the namespace is retrieved from the server side every time a browse operation is executed. All items are dynamically retrieved from the external server and are then completely available in the Local Namespace. The tool tip over a dynamic server connection indicates that the connection is dynamic by adding *Filter not activated* after the connection string.
- *Static* server connections are server connections where a filter containing the desired namespace is created and saved on the dataFEED OPC Suite side, and which is displayed when a browse operation is executed. Only filtered items from the external server are then available in the *Local Namespace*. The tool tip over a server connection indicates that the connection is static by adding *Filter activated* after the connection string.

Both the Dynamic and the Static server connections can have different states and icons for a node based on whether the current connection is online or offline.

Red Exclamation Mark State

Items for which no corresponding item exists in the connected server appear with a *red exclamation mark*. This state only applies to the tags and properties of an external server connection. The role of this item state is to notify that the respective item is not consistent as it differs from the one on the server side, or that the server side item does not exist anymore. The red exclamation mark is used to invalidate items from the dataFEED OPC Suite namespace in accordance with the items found in the server namespaces. An image with invalidated items is shown below:



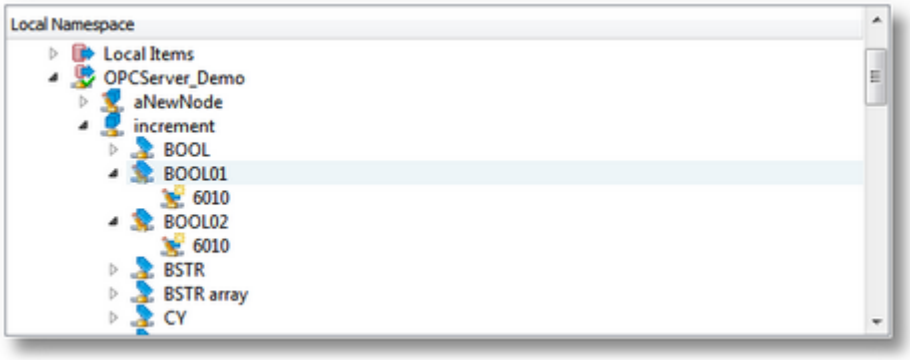
Note: Dynamic item creation may be supported on branches or on entire namespaces of OPC/Tunnel Servers, therefore the items created in the corresponding branches of the respective servers will not be invalidated. In fact the servers supporting dynamic item creation will update their internal namespaces, allocating the items created by the client. These dynamic items will appear as normal items if the connection is refreshed (from this client or from another) but they are not saved in persistent storage (and are therefore discarded when the server is stopped or restarted).

Modified State

Items which have been modified are marked with a small overlay pencil icon. This modified state applies to tags and properties. The state can only be applied to the tags and properties of an

© Softing Industrial Automation GmbH

external server connection. The role of this item state is to notify that the respective item differs from the associated server-side item. An image with modified items is shown below:



Icon Types

There are different types of icons for nodes, tags, properties, Local Items, _System nodes and server connection nodes. These icon types also depend on the state of the connection where the item is defined: online, offline, modified or invalid. The following table describes the classification of the icon types:

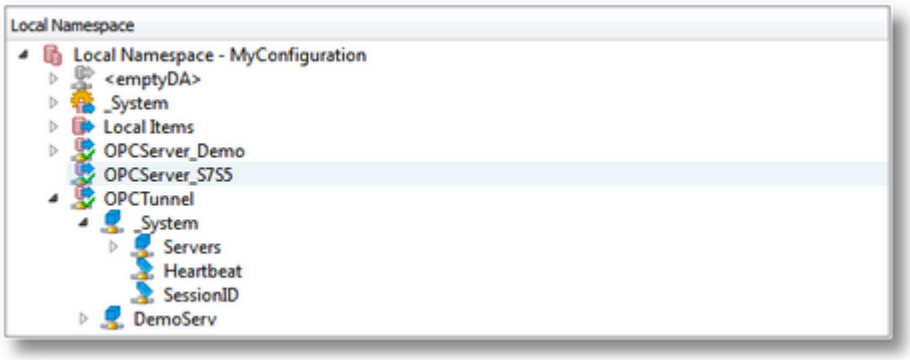
Icon	Description	Comments
	Local Namespace root node icon	This icon represents the root node of the dataFEED OPC Suite tree and it never changes.
	_System root node icon	This icon represents the *_System* root node and it never changes. This node is not always displayed.
	_System items node icon	General icon for a node below the _System node.
	_System items tag icon	General icon for a tag below the _System node.
	Local Items root node icon	This icon represents the Local Items root node and it never changes. This node is always displayed even if no items are defined in the Local Items namespace.
	Local Items node icon	General icon for a node below the Local Items node.
	Local Items tag icon	General icon for a tag below the Local Items node.
	Online server connection root node icon	This icon represents the root node of an online server connection (the connection type is irrelevant).
	Online server connection node icon	General icon for a node below an online server connection node.
	Online server connection tag icon	General icon for a tag below an online server connection node.

	Online server connection property icon	General icon for a property below an online server connection node.
	Online server connection invalidated tag icon	General icon for an invalid tag below an online server connection node.
	Online server connection invalidated property icon	General icon for an invalid property below an online server connection node.
	Offline server connection root node icon	This icon represents the root node of an offline server connection (the connection type is irrelevant).
	Offline server connection node icon	General icon for a node below an offline server connection node.
	Offline server connection tag icon	General icon for a tag below an offline server connection node.
	Offline server connection property icon	General icon for a property below an offline server connection node.
	Offline server connection invalidated tag icon	General icon for an invalid tag below an offline server connection node.
	Offline server connection invalidated property icon	General icon for an invalid property below an offline server connection node.
	Offline server connection tag icon	General icon for a tag below an offline server connection node.
	Offline server connection node icon	General icon for a node below an offline server connection node.
	Interrupted server connection root node icon	This icon represents an interrupted server connection.

7.2.6.4 Online Mode

The online mode is one of the states of the *Local Namespace* tree when the server connection is established and the server namespace is correctly retrieved. It can be easily identified by the colored item icons. Additionally the root node for the corresponding online server connection is marked with a green check mark:  . The image below shows various icons representing the online state of the server connection:

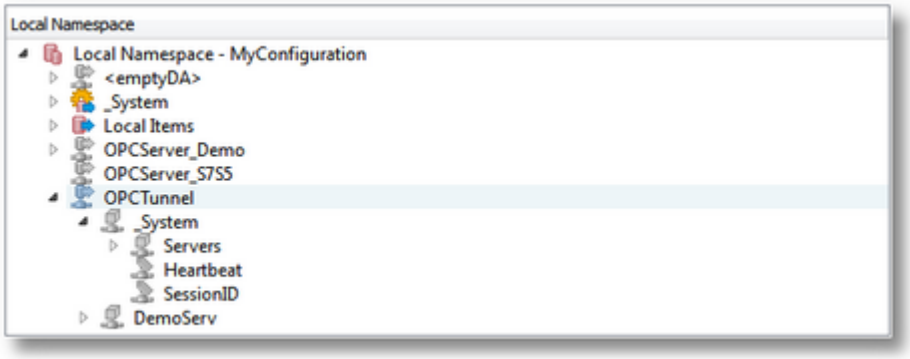
© Softing Industrial Automation GmbH



Note: The online state is only applicable for external server connections as the *_System* node and the *Local Items* node only have an online state (these items cannot be offline).

7.2.6.5 Offline Mode

The offline mode is one of the states of the *Local Namespace* tree when the server connection is disconnected. It can be easily identified by the grayed item icons. Additionally the root node for the corresponding server connection is either marked with a red x or grayed out. The image below shows various icons representing the offline state of the server connection:

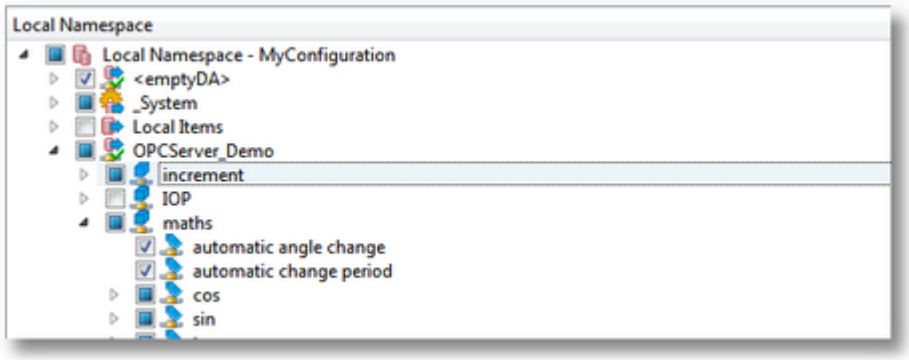


Note: The online state is only applicable for external server connections as the *_System* node and the *Local Items* node only have an online state (these items cannot be offline).

Note: In offline mode the server namespace might be incomplete.

7.2.6.6 Checked States

The Checked States page describes the complete *Local Namespace* tree from the Configuration: Data Processing - Store And Forward page, from the Store And Forward Wizard - Storage Items wizard page and a partial server namespace displayed on the OPC Server Wizard - Filter Items wizard page. The namespace from the *Store And Forward* add-on is similar to the one in *Local Namespace* described so far. As an extra it has a checkbox to the left of each tree item, which allows marking the respective item for monitoring. The checkboxes next to each item can be seen in the image below:

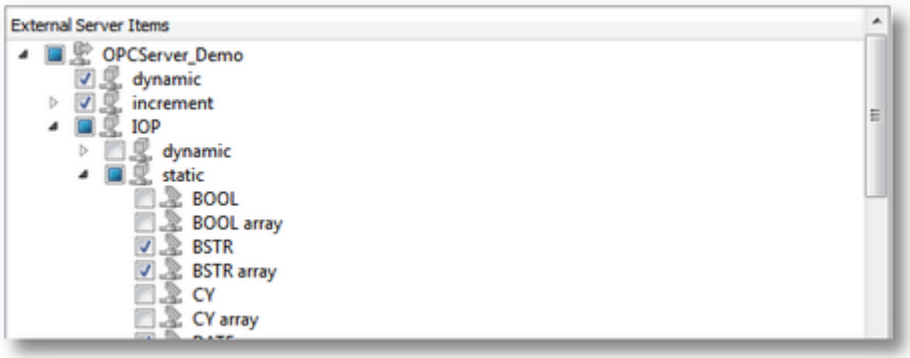


An element of the tree can be:

- marked (e.g. <emptyDA>, automatic angle change, automatic change period)
- unmarked (e.g. IOP, Local Items)
- partially marked, meaning the respective node has both marked and unmarked child nodes (e.g. _System, maths, cos, sin, OPCServer_Demo)
- disabled

Attention: All other states discussed so far: *online, offline, modified* and *invalid* still apply without further restrictions.

The OPC Server Wizard - Filter Items wizard page displays a similar tree which follows the same rules as the *Store And Forward* tree. The only difference is that the previously mentioned page displays only a partial sub-tree of the *Local Namespace*: the corresponding server connection namespace part:

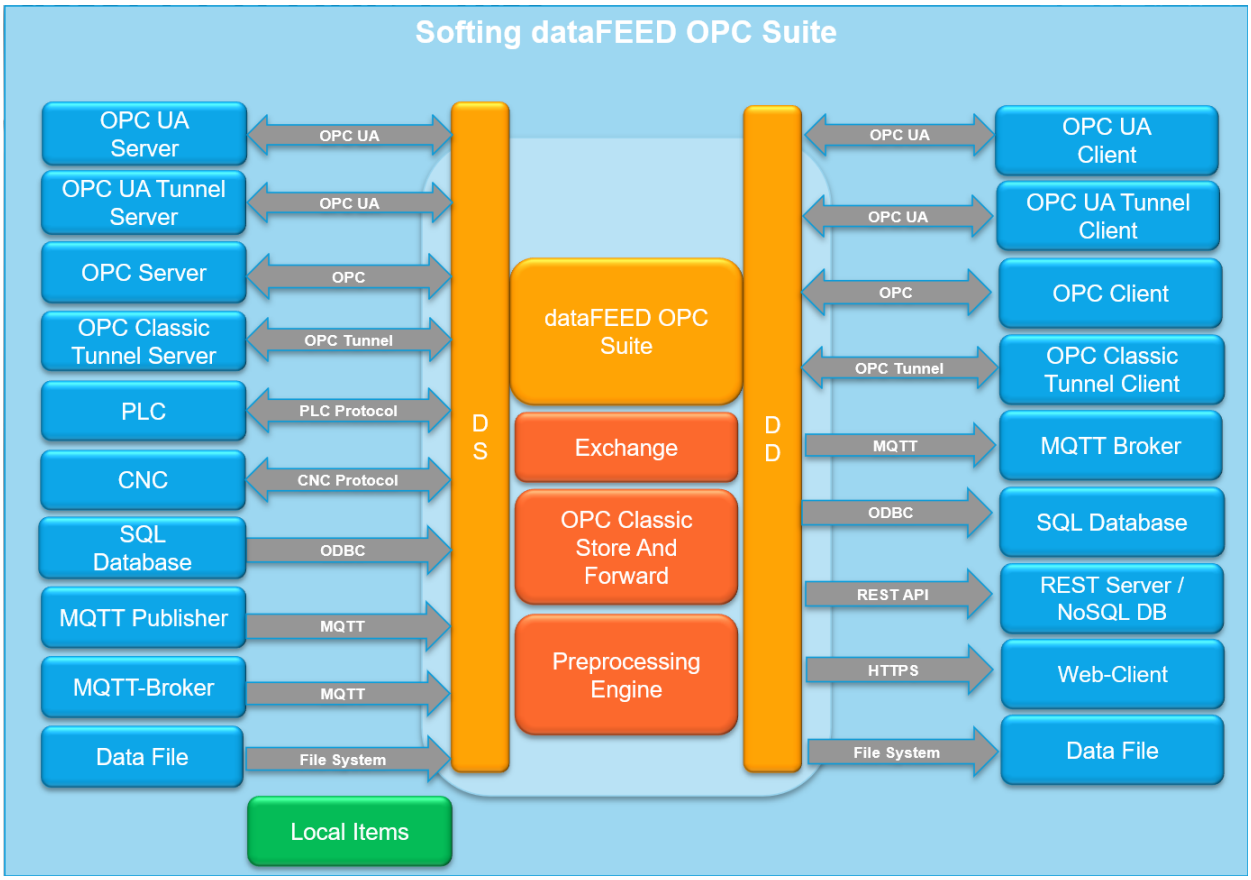


More information about the buttons on the right side in the above image can be found on the OPC Server Wizard - Filter Items wizard page.

© Softing Industrial Automation GmbH

7.3 dataFEED OPC Suite Configuration

In this section of the online help the user interface and the different configuration settings of the Softing dataFEED OPC Suite Configurator are described.



▾ Softing dataFEED OPC Suite - Data Sources

- Add connections to Siemens Devices @ Configuration: Data Source - Siemens Devices
- Add connections to Siemens SINUMERIK CNC Devices @ Configuration: Data Source - Siemens SINUMERIK CNC
- Add connections to Rockwell Devices @ Configuration: Data Source - Rockwell Devices
- Add connections to Modbus Devices @ Configuration: Data Source - Modbus Devices
- Add connections to B&R Devices @ Configuration: Data Source - B&R Devices
- Add connections to Mitsubishi Devices @ Configuration: Data Source - Mitsubishi Devices
- Add connections to OPC UA Servers @ Configuration: Data Source - OPC UA Server

- Add connections to external OPC servers @ Configuration: Data Source - OPC Server
- Add connections to external OPC Tunnel servers @ Configuration: Data Source - OPC Classic Tunnel Server
- Add connections to MQTT Publishers @ Configuration: Data Source - MQTT Publisher
- Add connections to MQTT Brokers @ Configuration: Data Source - MQTT Broker
- Reading fata From Files @ Configuration: Data Source - File Read
- Exchange data between OPC Clients @ Configuration: Data Source - Local Items

▾ Softing dataFEED OPC Suite - Data Processing

- Protection against data loss @ Configuration: Data Processing - Store And Forward
- Exchange data between OPC servers @ Configuration: Data Processing - Exchange
- Processing input data @ Configuration: Data Processing - Preprocessing Engine

▾ Softing dataFEED OPC Suite - Data Destinations

- Communicate with OPC UA Clients @ Configuration: Data Destination - OPC UA Client
- Access of external OPC Clients to the local namespace @ Configuration: Data Destination - OPC Client
- Access of external OPC Tunnel clients to the local namespace @ Configuration: Data Destination - OPC Classic Tunnel Client
- Output access to MOTT Brokers @ Configuration: Data Destination - MQTT Broker
- Communicate with Web/REST Clients @ Configuration: Data Destination - Web Service
- Archive data to SQL Databases @ Configuration: Data Destination - Database
- Archive data to REST Servers @ Configuration: Data Destination - REST Server
- Archive data to NoSQL Databases @ Configuration: Data Destination - NoSQL Database
- Archive data to a System File @ Configuration: Data Destination - File

© Softing Industrial Automation GmbH

▾ Softing dataFEED OPC Suite - Settings

- Configure diagnostics trace outputs to file @ Configuration: Settings - Diagnostics
- Enable the integrated web server @ Configuration: Settings - Web Server
- Advanced configuration settings @ Configuration: Settings - Advanced
- Discover the currently installed licenses and the license items used @ Configuration: Settings - Licenses

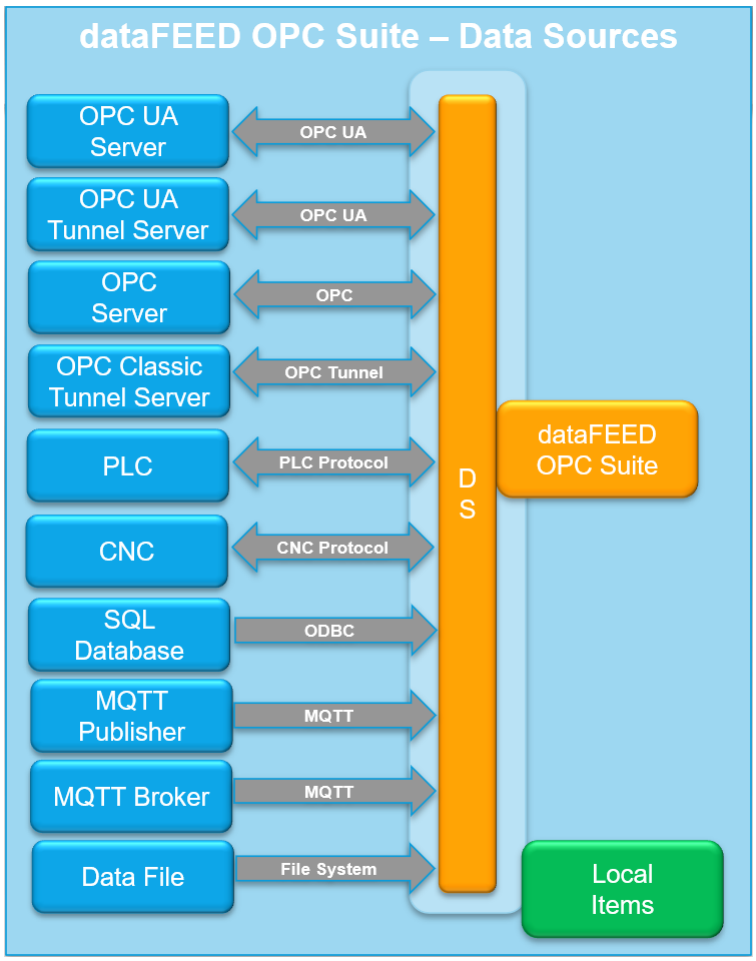
▾ Softing dataFEED OPC Suite - General

- Maintain global initial and alias values for actions of the local application @ Configuration: Global Initial and Alias Values

Softing dataFEED OPC Suite V5.38
© 2024 Softing Industrial Automation GmbH

7.3.1 Data Sources

This section of the Softing dataFEED OPC Suite Online Help describes the different dataFEED OPC Suite data sources.



▼ Softing dataFEED OPC Suite - Data Sources

- Add connections to Siemens Devices @ Configuration: Data Source - Siemens Devices
- Add connections to Siemens SINUMERIK CNC Devices @ Configuration: Data Source - Siemens SINUMERIK CNC
- Add connections to Rockwell Devices @ Configuration: Data Source - Rockwell Devices
- Add connections to Modbus Devices @ Configuration: Data Source - Modbus Devices
- Add connections to B&R Devices @ Configuration: Data Source - B&R Devices
- Add connections to Mitsubishi Devices @ Configuration: Data Source - Mitsubishi Devices
- Add connections to OPC UA Servers and UA Tunnel Servers @ Configuration: Data Source - OPC UA Server

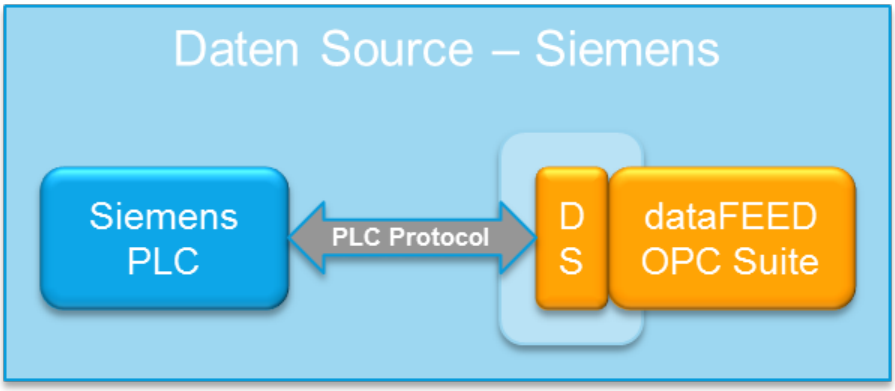
© Softing Industrial Automation GmbH

- Add connections to external OPC servers @ Configuration: Data Source - OPC Server
- Add connections to external OPC Tunnel servers @ Configuration: Data Source - Tunnel Server
- Add connections to MQTT Brokers @ Configuration: Data Source - MQTT Broker
- Add connections to MQTT Publishers @ Configuration: Data Source - MQTT Publisher
- Add connections to Data Files @ Configuration: Data Source - File Read
- Exchange data between OPC clients @ Configuration: Data Source - Local Items

Softing dataFEED OPC Suite V5.38
© 2024 Softing Industrial Automation GmbH

7.3.1.1 Siemens Devices

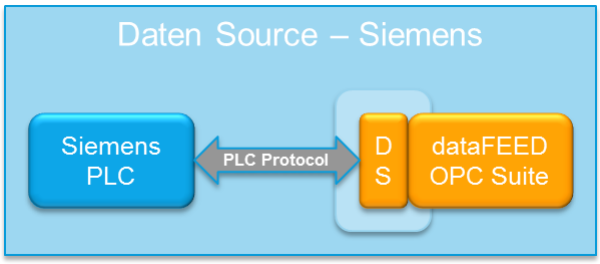
This section of the Softing dataFEED OPC Suite Online Help describes how to configure Siemens devices within Softing dataFEED OPC Suite.



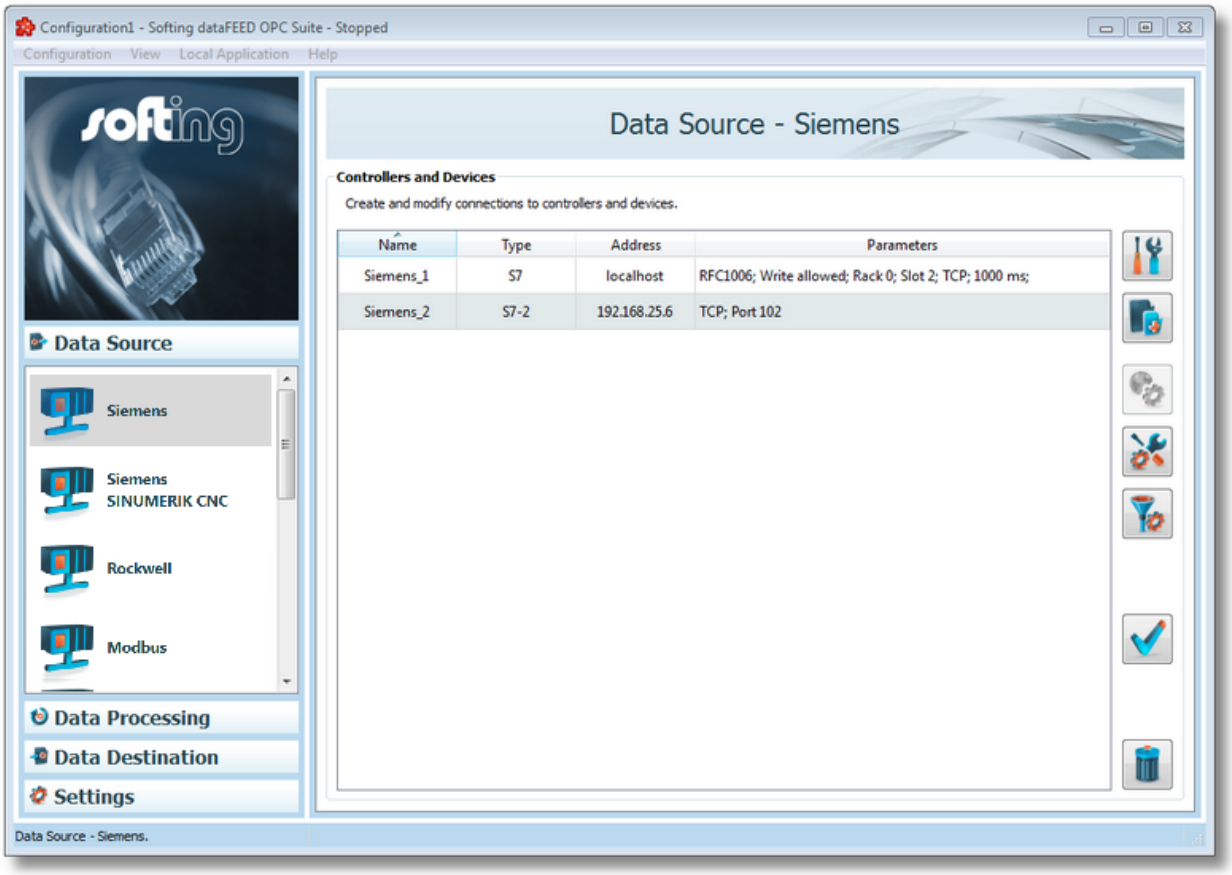
Controllers and devices data source makes it possible to exchange data between the field devices (PLCs) of a wide variety of manufacturers and OPC clients via Ethernet TCP/IP, RFC1006.

A detailed description on how to configure access to PLC devices functionality can be found in the Configuration section.

7.3.1.1.1 Configuration



Select Siemens Devices from the Data Source group in the navigator on the left to start configuring the connection to an external Siemens device. The Data Source - Siemens Device main user interface is displayed. The main element of this page is the External Devices list. Each entry shown in the list of the configuration section represents a connection to an external device.



Note: The list is only shown and editable if a configuration has been loaded.

Each Device connection has a unique name within the configuration, which is given in the *Name* column.

The name of the device will be shown in the namespace tree in the Softing dataFEED OPC Suite Configurator and in the local dataFEED OPC Suite application, and its entire namespace will be under this name.

© Softing Industrial Automation GmbH

Note: Device name must be unique string within a dataFEED OPC Suite configuration. Also the name could not contain address space delimiters, multi-byte characters, empty spaces.

The device type (second column) can be one of: S7, S7-2, S5, Send/Receive.

The third column is addressing the actual device address which can be IPv4, Host-name or IPv6.

Parameters column contains various attributes of the connection like: PLC header type, write allowed, ports, communication protocol.

Operation

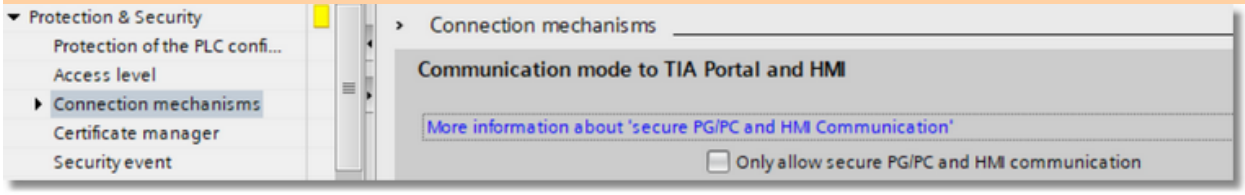
From the Siemens Devices main user interface certain operations can be performed over the selected connection by using the buttons on the right, by accessing the context sensitive menu or by pressing the keyboard shortcuts. The following actions are available:

- Change the properties of the selected data source by using the *Properties* button  , pressing the keyboard shortcut CTRL+P or selecting *Properties* from the context menu.
- Add a new connection by using the *Add* button  , pressing the keyboard shortcut CTRL+A or selecting *Add* from the context menu.
- Change global settings for the selected data source by using the Global settings button  .
- Change advanced settings for the selected data source by using the *Advanced settings* button  . (please note that advanced settings are connection specific while the global settings applies to all Siemens connections). This is available only for Siemens S7-2 connections (connections to Siemens S7 1200 and 1500 PLCs).
- Delete the selected data source by using the *Delete* button  , pressing the DEL / DELETE key on the keyboard or selecting *Delete* from the context menu.
- Address space definition button  , by pressing it the Siemens Device Wizard will be opened in it's last page. For more details see Address Space Definition page.
- Use the test connection button  to check the availability of the specified PLC.

Attention: Starting with the TIA version 17, Siemens added a new encryption layer to the communication protocol.
The following CPU families are affected:
- S7-1500 controller family as of firmware version V2.9
- S7-1200 controller family as of firmware version V4.5

To be able to communicate with the device the "Only allow secure PG/PC and HMI connection"

check box needs to be unchecked in the TIA project device settings.
If only secure communication is detected the test connection responds with "TLSError" error code.

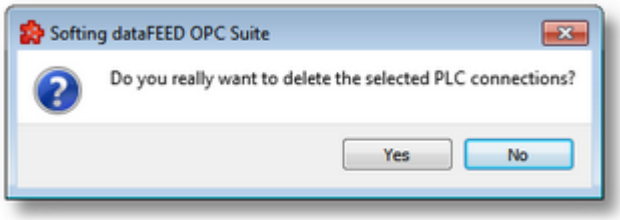


Note: If no controller / device connection is selected, the *Add a new data source* button is the only one available.

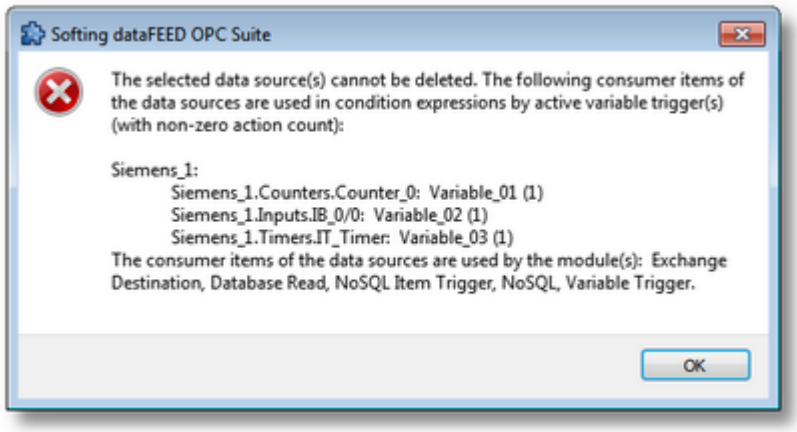
The *Delete* button  will remove the selected device connections from the list.

The user can select more than one server connection with the help of the CTRL or SHIFT keys.

A dialog will then prompt the user to confirm the deletion.

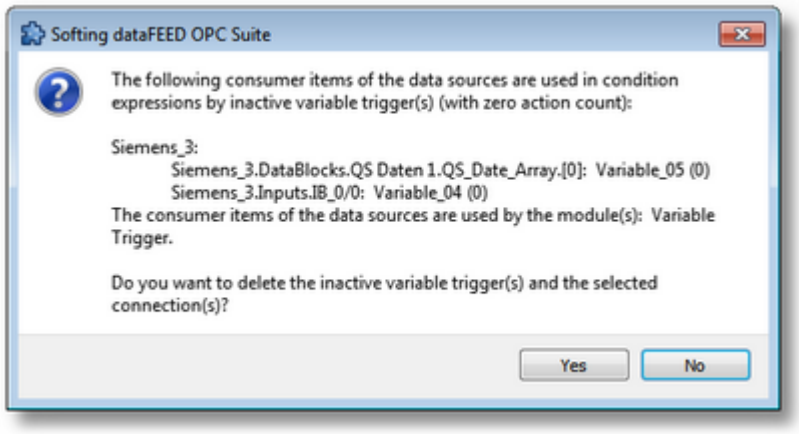


A dialog will inform the user that the data sources cannot be deleted, if consumer items and/or system items of the selected data sources are used by active variable triggers or other modules.



A dialog will prompt the user to confirm deletion of the data sources and inactive variable triggers, if consumer items and/or system items of the selected data sources are used only by inactive variable triggers.

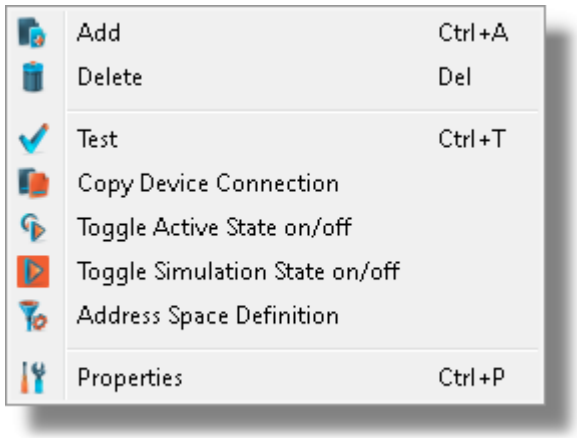
© Softing Industrial Automation GmbH



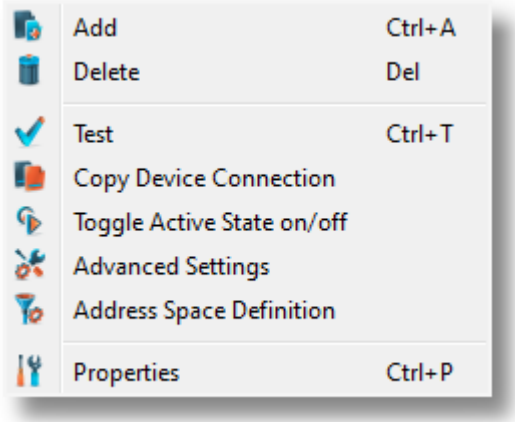
Attention: A device connection cannot be deleted if at least one data item defined in its namespace is used in a Processing or Data Destination action.

All these actions are also provided in the context menu of the External device list. Which actions are available depends on the current selection in the list.

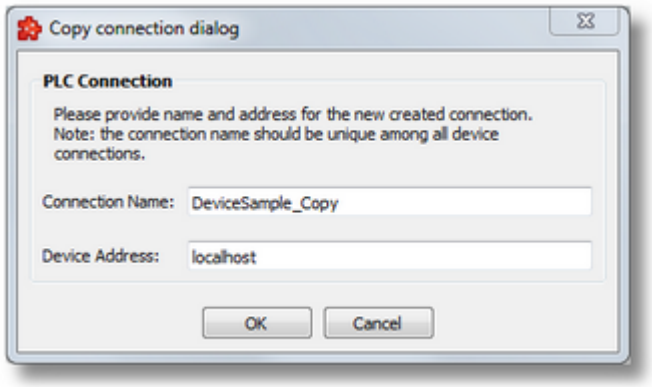
An example of the context menu can be seen below:



For Siemens S7-2 connection Simulation option is missing but instead Advanced Settings is available, like in the picture below:



- Copy device connection  . This action is available only in the context menu and it leads to a dialog where the user may provide a name and device address. See the screen shot below:



The proposed name is the original connection's name post fixed with "_Copy" and the default address will be "localhost".

- Toggle Activate state  . This action will enable / disable a devices connection. A disabled connection will be shown as grayed out in the device connection list and it will be ignored by runtime.

- Toggle Simulation state  . This action will mark a connection as simulation (or remove the simulation mode). Simulation connections are shown in red in main device list. Please notice that S7-2 connections do not support the simulation mode so this option will be invisible for those connections.

Note: The connection name should be unique among all device connections names.

Note: The device address should have a valid host name or an IP address in format IPv4 or IPv6.

7.3.1.1.2 Siemens Device Wizard

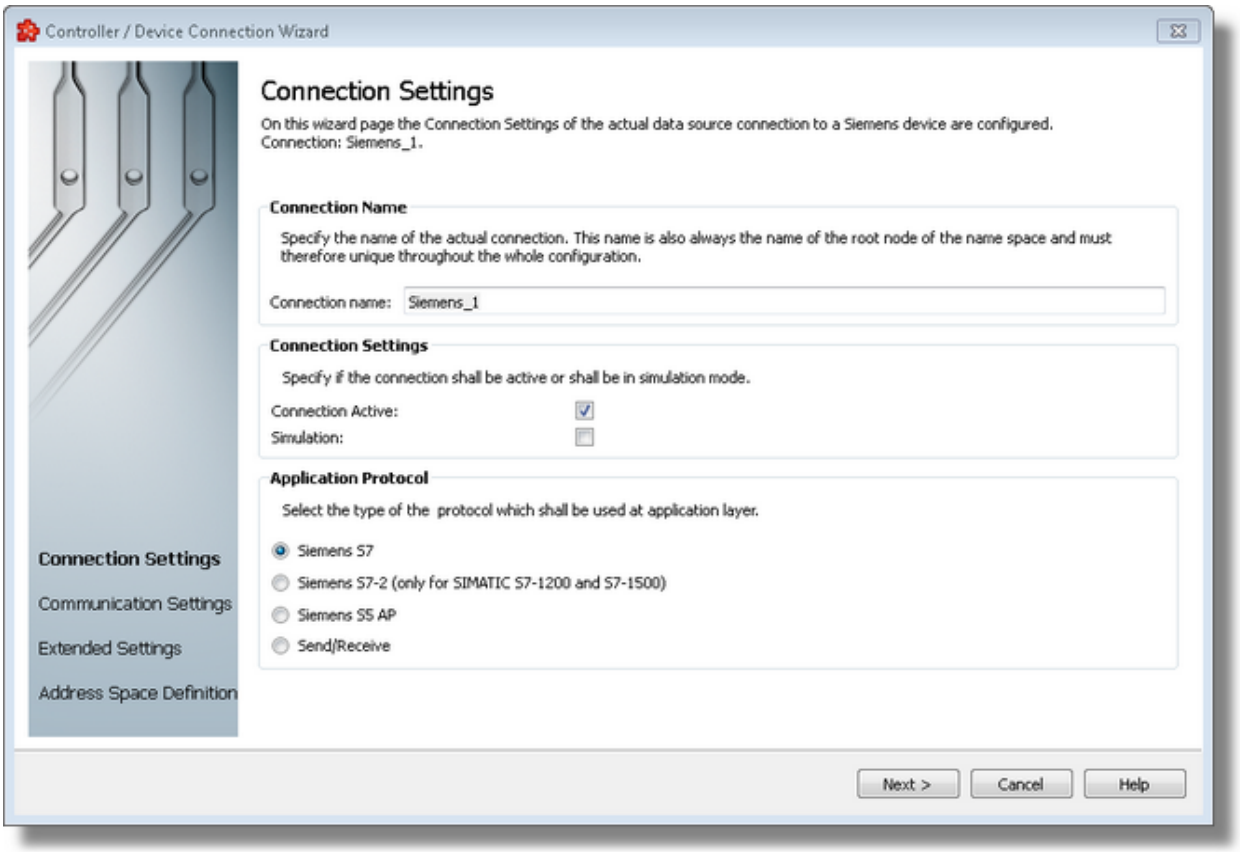
With the Siemens Device Wizard you can create and modify a data source connection to an external PLC device:

- Specify connection settings @ Siemens Device Wizard - Connection Settings
- Specify communication parameters @ Siemens Device Wizard - Communication Settings
- Extended settings @ Siemens Device Wizard - Extended Settings

© Softing Industrial Automation GmbH

- Extended & Filter settings @ Siemens Device Wizard - Extended & Filter Settings
- Address space definition @ Siemens Device Wizard - Address Space Definition

On this page the connection name that will uniquely identify the data source and some generic connection settings can be provided.



Connection Name

An unique identifier for the current connection. In case the provided name is already used by other connection, the user use is not allowed to go into the next wizard page.

Connection active

This flag specifies if the current connection will be enabled or not.

Simulation

This flag specifies if the current connection will be simulated by the server or it will represent a real device connection.

Note: Simulation option is not available for Siemens S7-2 connections. The check box will be disabled and not checked when S7-2 protocol type is selected.

Application protocol

Currently supported protocols are:

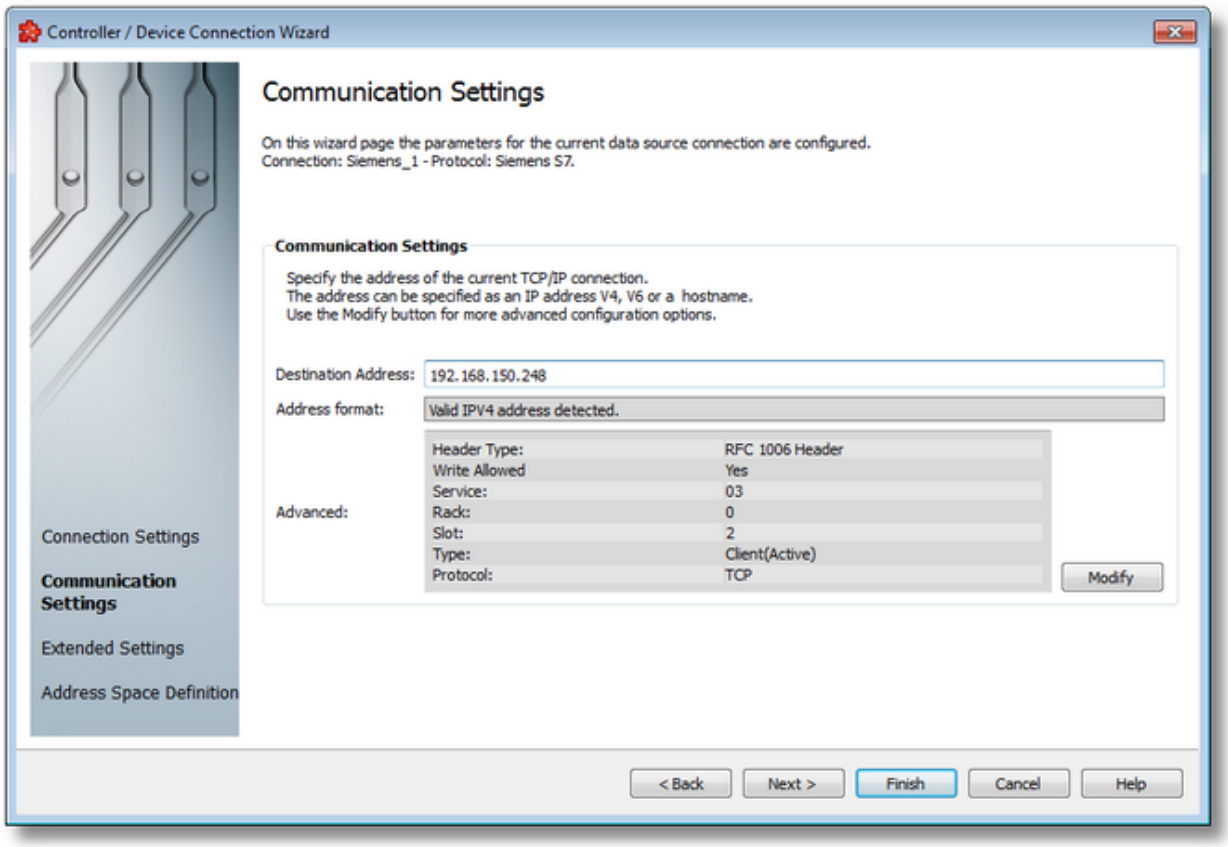
- Siemens S7
- Siemens S7-2 (only for Simatic S7-1200 and S7-1500) Siemens Device Wizard - Communication Settings S7-2 >>>
- Siemens S5
- Send/Receive

Note: The "NetLink" protocol was removed starting with version 5.38. On existing configurations created based on older versions using "NetLink" protocol, the protocol is available but recommended to be switch to Siemens S7.

Siemens Device Wizard - Communication Settings >>>

On this wizard page various parameters for the current TCP/IP connection are configured.

© Softing Industrial Automation GmbH



Destination Address: IP address or name

This is where the IP address of the target station (IP address of the PLC) or the DNS name of the target station is entered. If the computer was configured for DNS and if a DNS server is available on the network, the symbolic name of the target station can also be entered.

For how an IP address is set up, see IP address.

Address Format

Read-Only field; automatically detects the format of destination address.

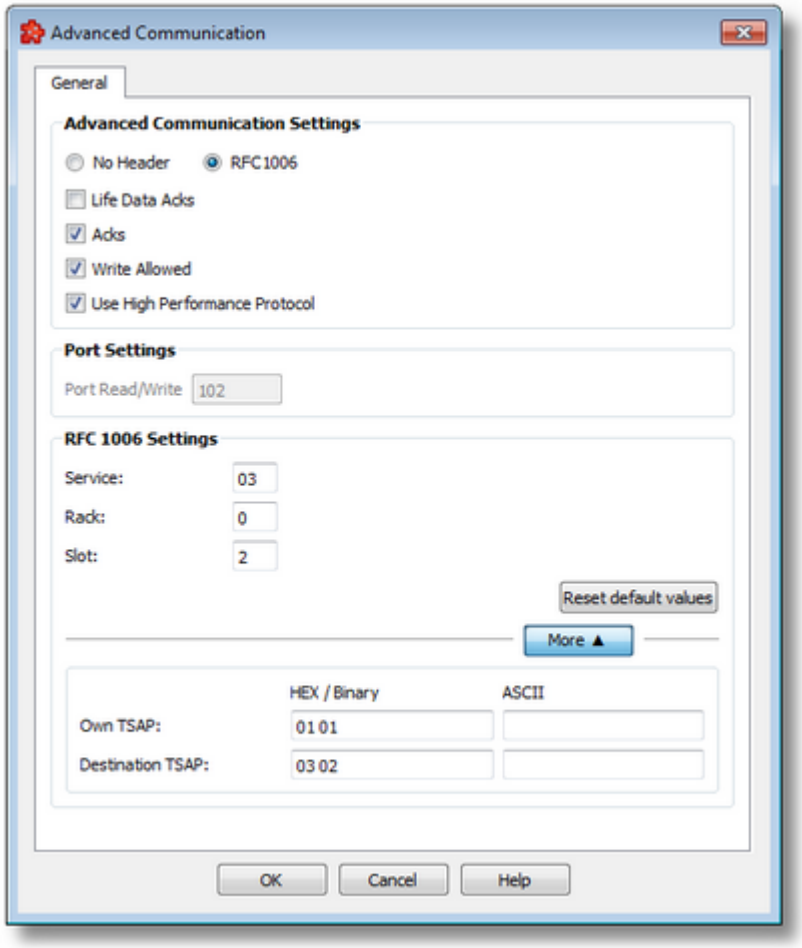
Advanced:

Read-Only field; this is a summary for TCP/IP Communication advanced settings. These settings can be changed using Modify button that lead to following dialogs:

- Advanced Communication Settings S7
- Advanced Communication Settings S5
- Advanced Communication Settings Send/Receive

<<< Siemens Device Wizard - Connection Settings Siemens Device Wizard - Extended Settings >>>

In this dialog specific S7 advanced communication parameters can be adjusted.



No Header

"No header" is used for connections to stations which only support the pure TCP/IP protocol. In this case, the application program is responsible for monitoring on both sides.

RFC1006

When RFC1006 is activated, H1 frames are packed into a TCP/IP frame for transportation.

© Softing Industrial Automation GmbH

Acks

When you select "Acks", connection monitoring of the socket libraries is enabled. If this option is disabled, the connection is considered as concluded after the KeepAliveTime but the communication partner is not told. We recommend enabling this option.

Life Data Acks

When you select "Life Data Acks", user data telegrams are transmitted with no data content in order to keep a connection alive which is not used cyclically. Both communication partners must support this functionality. With communication to S7 PLCs, we recommend switching this option off.

Write to the PLC permitted (Write Allowed)

Here you specify whether write access to the PLC is to be possible via this connection and a write connection is to be established to the hardware. You can always disable write accessing via a connection if you only want to use read accesses or if you want to be sure that no values on the PLC are to be changed. When write accessing the PLC via a Write (Write activeConnection) or a Send/Receive (Send Direct Connection for Send/Receive), certain points of the PLC are written with values (each write access is preceded by a read access).

Use High Performance Protocol

When the optimized protocol is activated, accesses to variables which are NOT in the same operand area (e.g., DB5.W3 + MW1 + T2, etc.) are converted internally on the server to a single field access to the target device. This reduces the number of data packets that are transported via the network. The share of user data is greater, the data packets are better utilized and the network load is reduced. Thus the server works particularly quickly in this mode.

Port for read accesses, port for write accesses

Port numbers are addresses which are used within the transport layer to address applications. Port numbers are required for TCP connections. The port is parallel to the TSAPs for RFC1006 and H1 connections. The port number is a 16-bit number from 1 to 65535. Keep the following in mind when you set up the connection.

Note: The port must be identical on both sides of the connection before you can set up the connection.

Read, write and event ports should be addressed differently. Port numbers 1 to 1023 should not be used since these are already being used by well-known services (so-called well-known ports).

RFC 1006 Settings

Service:
Contains identifiers for devices provided with S7 resources.

01: PG or PC

02: OS (operator or monitoring device)

03: Other (e.g., OPC server, Simatic S7 PLC)

Rack:
The number of the rack in which the CPU resides, a number from 0 to 7.

Slot:
Slot number in which the CPU resides, a number from 0 to 31.

For S7-300 the slot 2 is used always.

Local TSAP, remote TSAP

TSAPs (Transport Service Access Points) are addresses which are used within the transport layer to address applications. TSAPs are required for RFC1006 connections. To set up a connection, both the local TSAP and the TSAP of the communication partner must be specified. The following must be adhered to.

Attention: The TSAPs must coincide crosswise before the connection can be established!

The local TSAP of the one system must correspond to the remote TSAP of the other system.

More

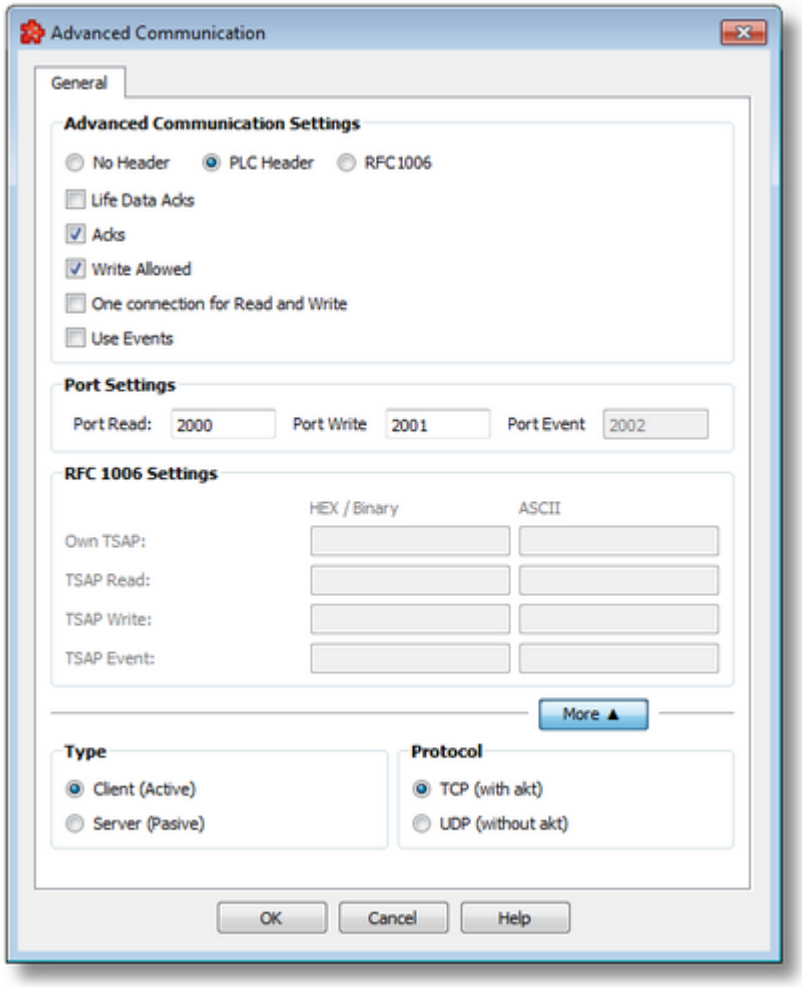
The button "More" opens entry fields for the TSAPs.

For further information on the TSAPs

TSAPs for S7 connections

In this dialog specific S5 advanced communication parameters can be adjusted.

© Softing Industrial Automation GmbH



No Header

"No header" is used for connections to stations which only support the pure TCP/IP protocol. In this case, the application program is responsible for monitoring on both sides.

PLC Header

"PLC header" is used for communication to the INAT components (S5-TCP/IP, echolink, echocollect)

RFC1006

When RFC1006 is activated, H1 frames are packed into a TCP/IP frame for transportation.

Life Data Acks

When you select "Life Data Acks", user data telegrams are transmitted with no data content in order to keep a connection alive which is not used cyclically. Both communication partners must support this functionality. With communication to S7 PLCs, we recommend switching this option off.

Acks

When you select "Acks", connection monitoring of the socket libraries is enabled. If this option is disabled, the connection is considered as concluded after the KeepAliveTime but the communication partner is not told. We recommend enabling this option.

Write to the PLC permitted (Write Allowed)

Here you specify whether write access to the PLC is to be possible via this connection and a write connection is to be established to the hardware. You can always disable write accessing via a connection if you only want to use read accesses or if you want to be sure that no values on the PLC are to be changed. When write accessing the PLC via a Write (Write activeConnection) or a Send/Receive (Send Direct Connection for Send/Receive), certain points of the PLC are written with values (each write access is preceded by a read access).

A connection for Read and Write access

With the S7 and Send/Receive protocols, read and write accesses via a connection are possible and activated as the default setting. Only a port for TCP/IP or only a TSAP for RFC1006 must be specified for the read and write access. The S5 protocol is an exception: Read and write accesses via one connection are NOT possible. One connection is required for read accesses and another connection is required for write accesses. A port number (or TSAP) is required for read accesses and a port number (or TSAP) is required for write accesses. More recent INAT S5-TCP/IP cards, echolink and echocollect permit read and write accesses via one connection.

Use Events

Events from the PLC are only possible with S5 connections.

In addition to cyclic read (fetch active) and write (write active) accesses, the server also supports events from the PLC. A channel is reserved which is used to deliver the spontaneous data from the PLC. The event channel is its own connection. Accordingly, a separate “port for events” (for TCP/IP connections) or a separate (remote) “TSAP for events” (for RFC1006 connections) is defined.

The port numbers/TSAPs for read, write accesses and events may never be the same!

Port for Read, Port for Write, Port for Read/Write

Port numbers are addresses which are used within the transport layer to address applications. Port numbers are required for TCP connections. The port is parallel to the TSAPs for RFC1006

© Softing Industrial Automation GmbH

and H1 connections. The port number is a 16-bit number from 1 to 65535. Keep the following in mind when you set up the connection.

Note: The port must be identical on both sides of the connection before you can set up the connection.

Read, write and event ports should be addressed differently. Port numbers 1 to 1023 should not be used since these are already being used by well-known services (so-called well-known ports).

RFC 1006 Settings

Local TSAP, remote TSAP

TSAPs (Transport Service Access Points) are addresses which are used within the transport layer to address applications. TSAPs are required for RFC1006 connections. To set up a connection, both the local TSAP and the TSAP of the communication partner must be specified. The following must be adhered to.

The local TSAP of the one system must correspond to the remote TSAP of the other system.

Rules for the entry of the TSAPs for S5 connections

· TSAPs have a minimum length of 2 bytes and a maximum length of 8 bytes for TCP/IP.

· TSAPs can be entered either as hex or as ASCII characters.

· The TSAPs for S5 connections have no special meaning. They are selected at random.

For further information on the TSAPs

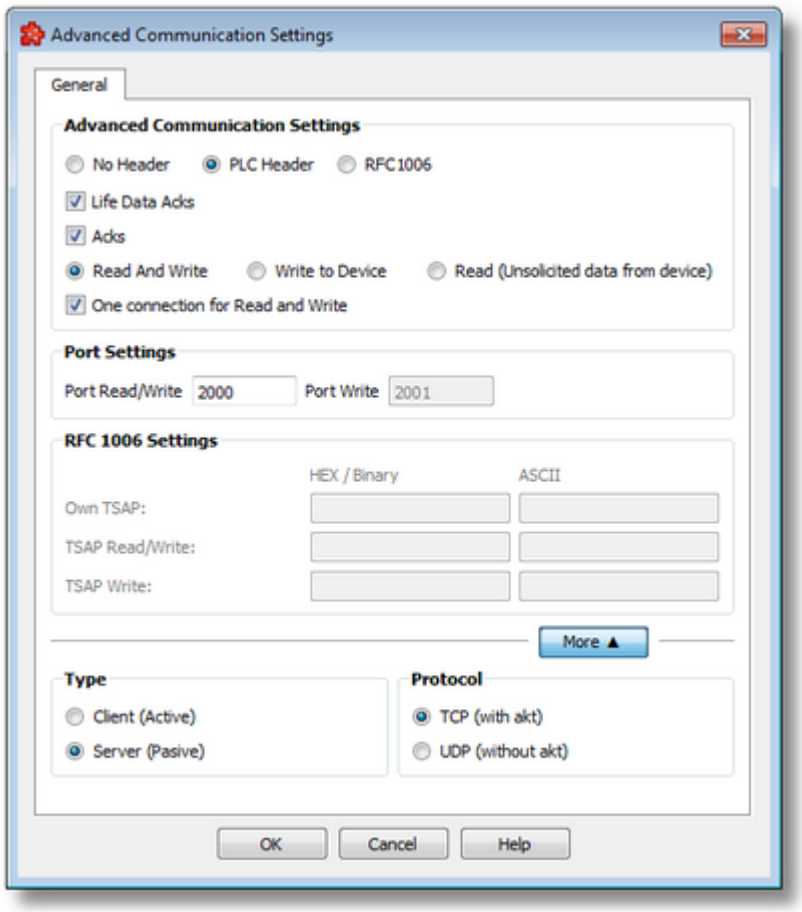
Type active (client) / passive (server)

Here you specify whether the own station actively initiates the connection setup or passively expects the connection setup by the other station. You may not enter the same value on both sides of the connection. Usually, the PC with OPC server is parameterized as active. This is also used here as the default value.

TCP/UDP protocol

The TCP is a secure protocol with checksum and confirmation. UDP is not secure and is handled via datagram services. The TCP protocol should be selected for normal data transmission.

In this dialog specific Send/Receive advanced communication parameters can be adjusted.



No Header

"No header" is used for connections to stations which only support the pure TCP/IP protocol. In this case, the application program is responsible for monitoring on both sides. Keep attention to have a fix length of frames when using no header.

PLC Header

"PLC header" is used for communication to the INAT components (S5-TCP/IP, echolink, echocollect)

RFC1006

When RFC1006 is activated, H1 frames are packed into a TCP/IP frame for transportation.

© Softing Industrial Automation GmbH

Acks

When you select "Acks", connection monitoring of the socket libraries is enabled, acknowledgement frames are sent. If this option is disabled, the connection is considered as concluded after the KeepAliveTime but the communication partner is not told. We recommend enabling this option.

Life Data Acks

When you select "Life Data Acks", user data telegrams are transmitted with no data content in order to keep a connection alive which is not used cyclically. Both communication partners must support this functionality. With communication to S7 PLCs, we recommend switching this option off.

Write to the PLC permitted (Write Allowed)

Here you specify whether write access to the PLC is to be possible via this connection and a write connection is to be established to the hardware. You can always disable write accessing via a connection if you only want to use read accesses or if you want to be sure that no values on the PLC are to be changed. When write accessing the PLC via a Write (Write activeConnection) or a Send/Receive (Send Direct Connection for Send/Receive), certain points of the PLC are written with values (each write access is preceded by a read access).

Read and Write

Data could be written into the device (Write), and unsolicited data can be sent from the device (Read). The read and write tags have nothing in common.

Write to Device

Data could only be written into the device (Write).

Read (Unsolicited data from device)

Only unsolicited data can be sent from the device (Read)

One connection for Read and Write access

When read and write accesses is enabled, it is possible to establish a duplex connection to the device for read and write. With the option disabled, there are two connections established. It depends of the capability of the device, whether one or two connections should be selected.

Port for Read, Port for Write, Port for Read/Write

Port numbers are addresses which are used within the transport layer to address applications. Port numbers are required for TCP connections. The port is parallel to the TSAPs for RFC1006 and H1 connections. The port number is a 16-bit number from 1 to 65535. Keep the following in mind when you set up the connection.

Note: The port must be identical on both sides of the connection before you can set up the connection.

Read, write and event ports should be addressed differently. Port numbers 1 to 1023 should not be used since these are already being used by well-known services (so-called well-known ports).

RFC 1006 Settings:

Local TSAP, remote TSAP

TSAPs (Transport Service Access Points) are addresses which are used within the transport layer to address applications. TSAPs are required for RFC1006 connections. To set up a connection, both the local TSAP and the TSAP of the communication partner must be specified. The following must be adhered to.

The local TSAP of the one system must correspond to the remote TSAP of the other system.

For further information on the TSAPs

Type active (client) / passive (server)

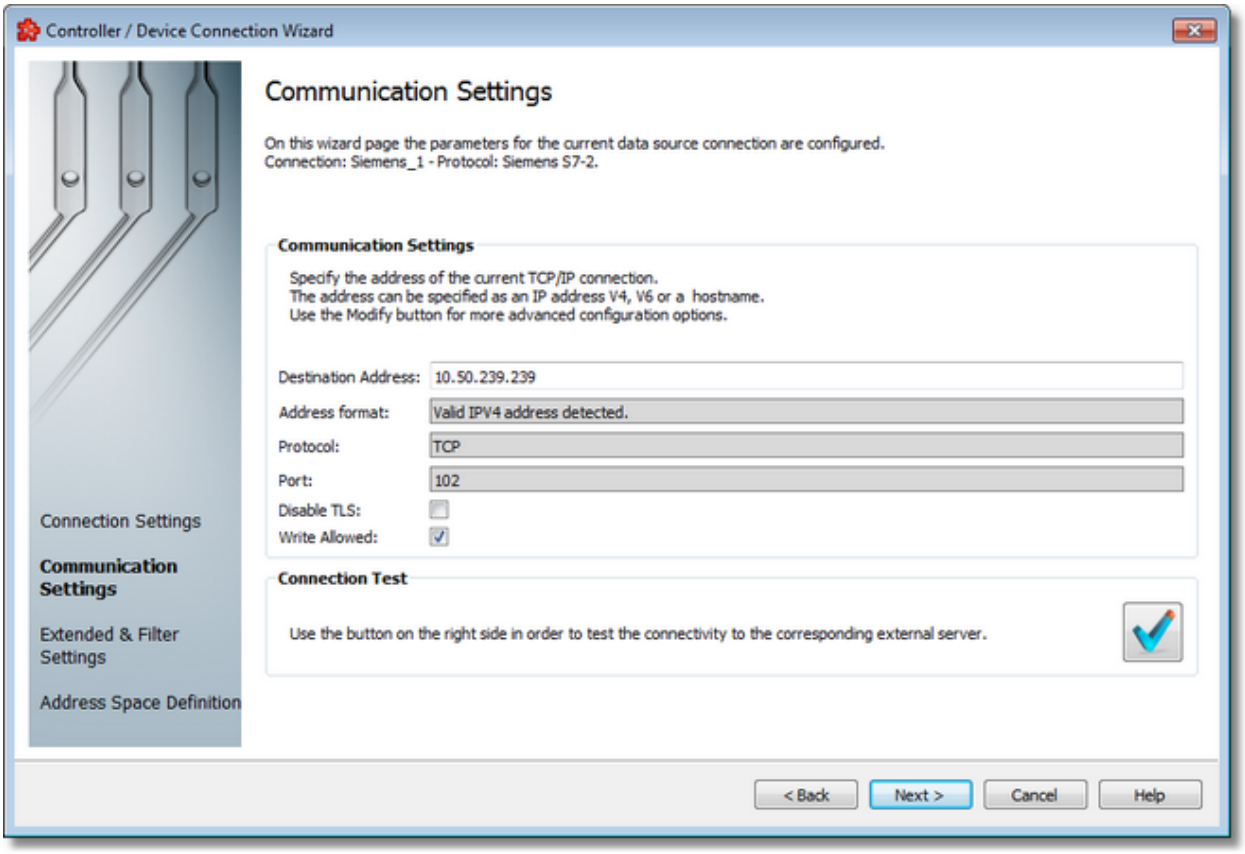
Here you specify whether the own station actively initiates the connection setup or passively expects the connection setup by the other station. You may not enter the same value on both sides of the connection. Usually, the PC with OPC server is parameterized as active. This is also used here as the default value.

TCP/UDP protocol

The TCP is a secure protocol with checksum and confirmation. UDP is not secure and is handled via datagram services. The TCP protocol should be selected for normal data transmission.

On this wizard page various parameters for the current TCP/IP connection are configured and connection test.

© Softing Industrial Automation GmbH



Destination Address: IP address or name

This is where the IP address of the target station (IP address of the PLC) or the DNS name of the target station is entered. If the computer was configured for DNS and if a DNS server is available on the network, the symbolic name of the target station can also be entered.

For how an IP address is set up, see IP address.

Address Format

Read-Only field; automatically detects the format of destination address.

Protocol

Read-Only field; always TCP since no other protocols are supported.

Port

Read-Only field; always 102.

Disable TLS

Disable TLS support for S7-2 connections. In checked state the TLS protocol communication will be disabled.

If it is not checked by default it will try to connect with TLS and if it is not supported by the PLC, it will fall back to normal communication.

Write Allowed

If checked, write operations are allowed to PLC based on the items read/write protection settings.

In unchecked state all PLC items are Read-Only.

Connection test

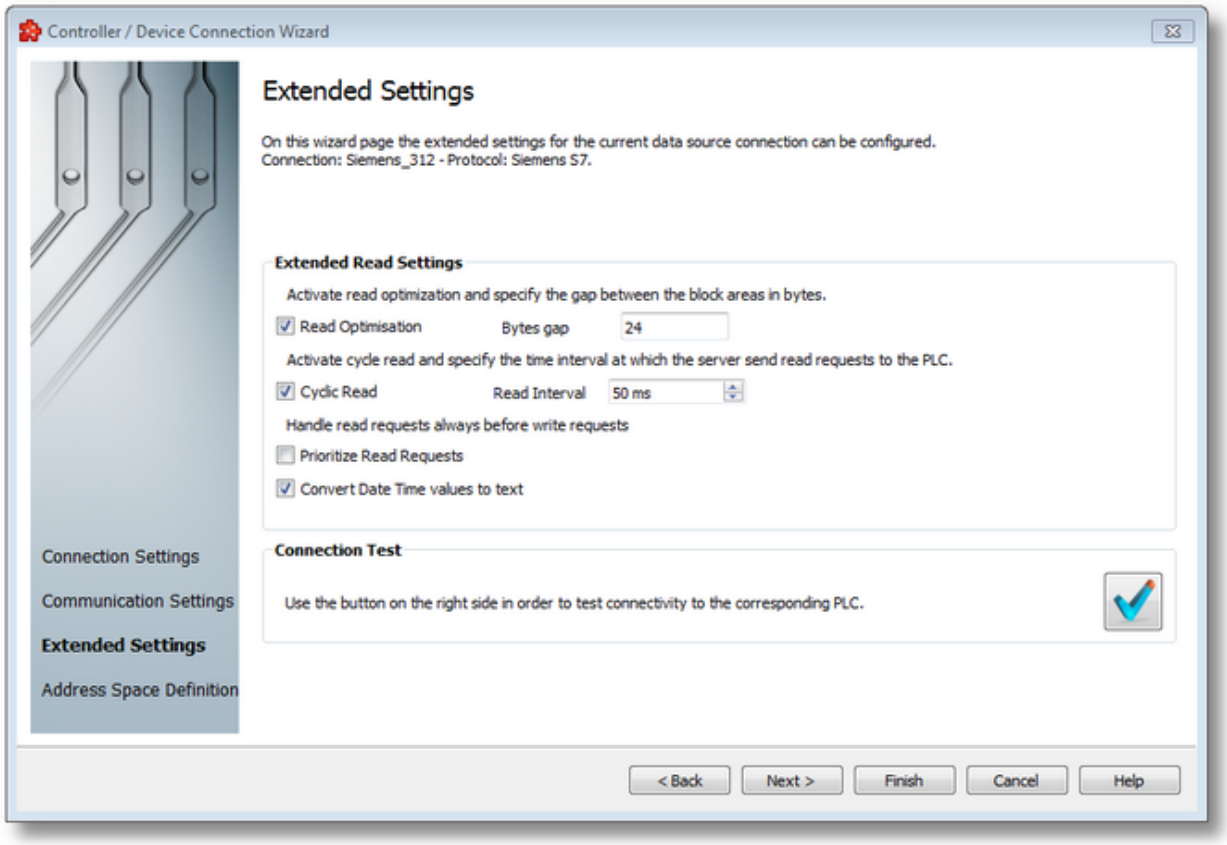
Convenience button  provided to check it the current settings can be used to connect at device.

The same functionality is provided in main configuration overview.

<<< Siemens Device Wizard - Connection Settings Siemens Device Wizard - Filter Settings >>>

On this wizard page the extended settings for the current data source connection can be configured.

© Softing Industrial Automation GmbH



Read optimization

Reading from the controller can be optimized here. If several variables are requested which are located in the same operand area (e.g., in a data block or in the flag area), the complete area is read from the first variable to the last variable without read optimization. Gaps between the individual block areas are also read no matter how large these gaps are. The maximum size of the gaps (in bytes) between the requested individual items within a data block without a new block being created is entered. If the gap equals the specified value, a new block is created.

Enable cyclic read access and read interval

Cyclic read access is activated as the standard. The server reads the values of the variables from the hardware via a fetch active connection at a certain time interval. It can be specified in Read interval [in ms] the time intervals at which the server is to send a read request to the PLC. The setting “0” means that it should be read as soon as possible. If "Cyclic read" is selected, read access is performed once when the client requests and no longer cyclically. However, this is only necessary in special cases and requires a precise knowledge of the communication.

Prioritize Read Requests

This option enables the application to change the order or requests in its queue in such way that the Read requests are executed before Write requests, if there are any in the queue.

For example, if there are currently two Write requests to be executed and a Read request is received, then the latter will be executed first, and only after that the Write requests.

Convert Date Time values to text

This option is on by default and it is a per connection setting. It specified that all date and time values coming from a PLC are forwarded to dataFEED in text format. If the option is unchecked, three date and time types will be converted to date time values instead of text.

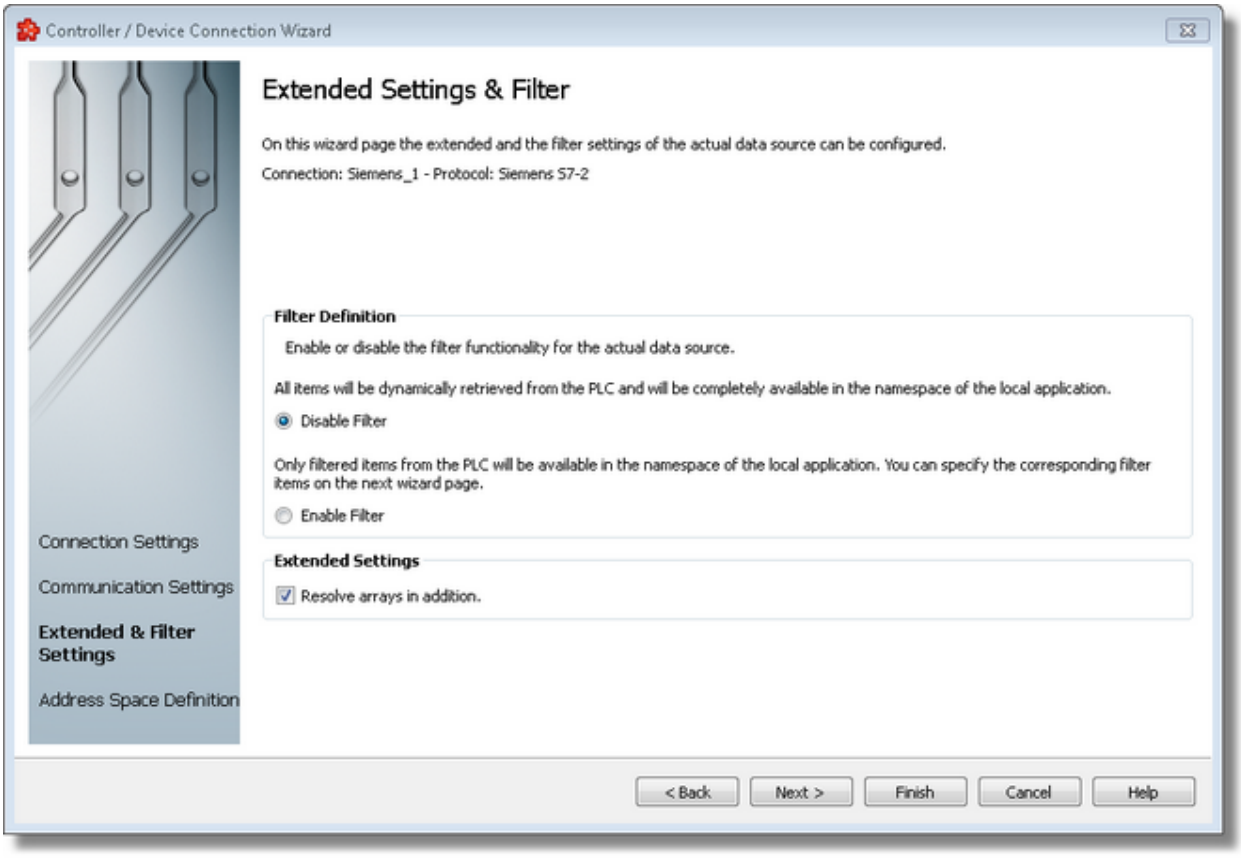
These three types are DT (Date and Time), DU (Date and Time (SQL)) and D (Date) (see Suffixes section for more information).

Connection Test

Use the test connection button  to check the availability of the specified PLC.

<<< Siemens Device Wizard - Communication Settings Siemens Device Wizard - Address Space Definition >>>

On this wizard page you can activate or deactivate the filter functionality and set the extended settings for the selected data source.



© Softing Industrial Automation GmbH

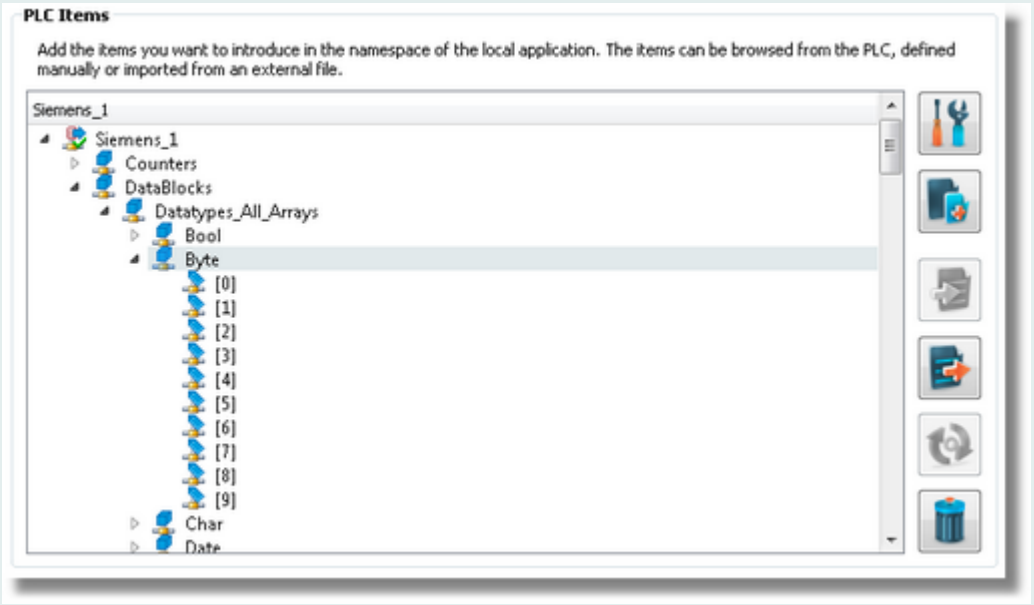
Filter Definition

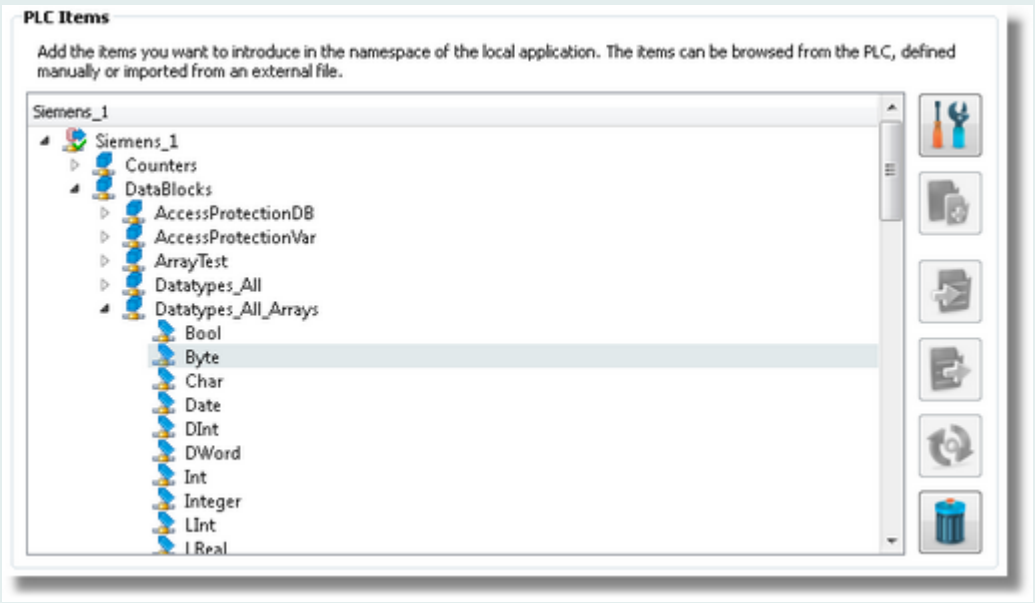
For the *Filter Definition* there are two possible configuration settings:

Filter Definition	Description
Deactivate Filter	All items will be dynamically retrieved from the device and will be completely available in the namespace of the local dataFEED OPC Suite application.
Activate Filter	Only filtered items from the device will be available in the namespace of the local dataFEED OPC Suite application. You can specify the filter items on the next wizard page.

Extended Settings

For the extended settings the following options are available for being configured:

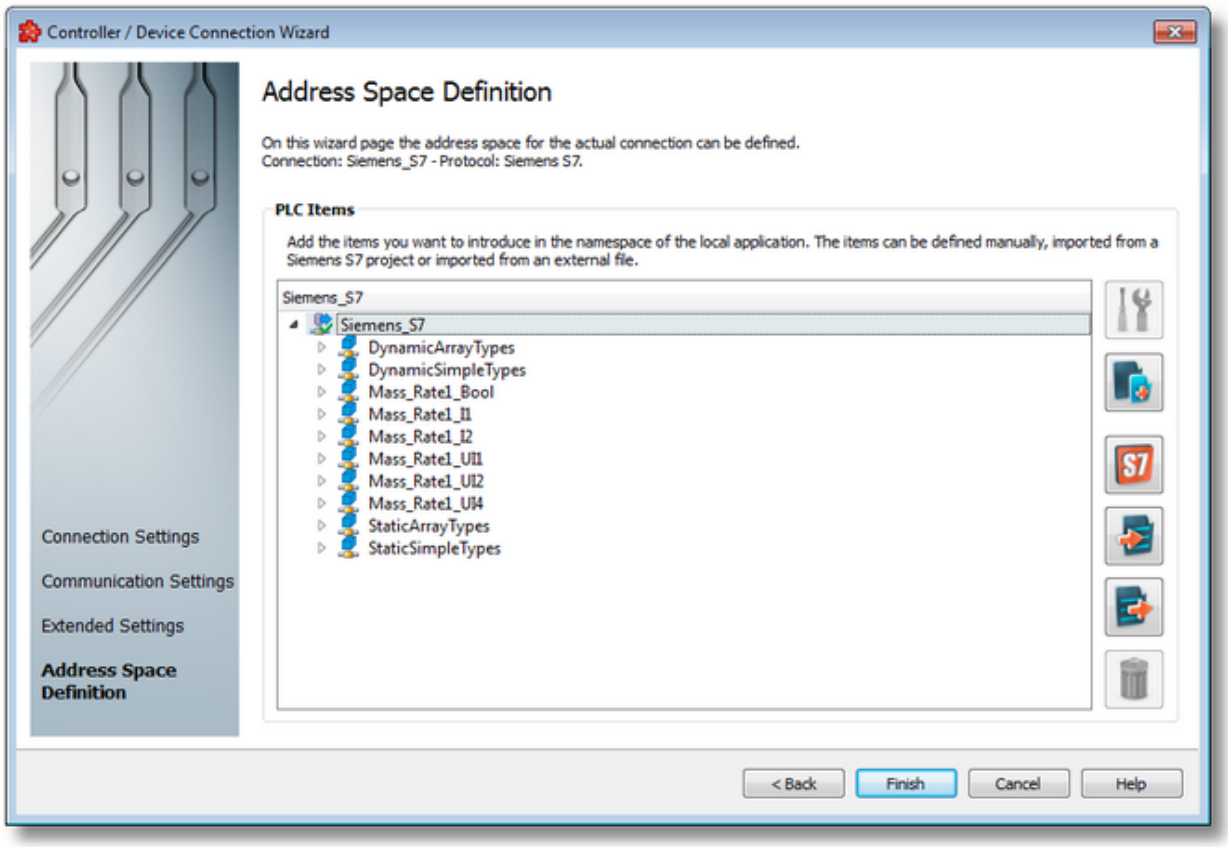
Extended settings	Description
Resolve arrays in addition	Enabling this option will create an item child for each elements of a variable which has an array data type. Sample: • Enabled: 

	• Disabled: 

<<< Siemens Device Wizard - Communication Settings Siemens Device Wizard - Address Space Definition >>>

On this wizard page the data items for the actual PLC connection can be defined. The address space can be defined by manually adding items, by importing the address space from an external file or by importing a device symbol file (for connections of type S7 and Modbus).

© Softing Industrial Automation GmbH



PLC Connection Items

An editable tree view where the user can change the address space structure or content by adding, deleting or modifying items.

There are three categories of items that may be used in the tree:

- **root node** (having the name of the current connection). This node is read only, the user has just the possibility to add or delete children of this.
- **nodes:** these elements are associated with branches in tree. They can have children but they are typeless and do not carry any syntax.

Note: The items of type Node represent intermediate nodes in the tree. They are used for hierarchy purposes.

- **tags:** these elements are associated with leaves in tree.

Note: A tag cannot have any children, so Add is not available for tags.

Operation

Depending on the current selection in the *Plc Connection Items* tree, the following operations are possible using the buttons on the right, the context menu or the corresponding shortcuts:

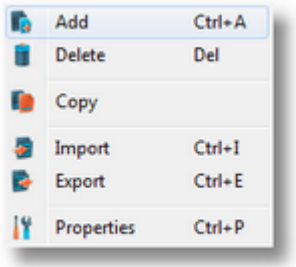
- Properties of the selected item (Ctrl+P) 
- Add a new item (Ctrl+A) 
- Import from file (Ctrl+I) 
- Export to file (Ctrl+E) 
- Delete data items (Del) 
- Import S7 symbol file 

Attention: Importing S7 symbol files is not available for Siemens S7-2 connections (direct connections to Siemens S7-1200 or S7-1500 PLCs).

Two more operations are supported, which are only accessible from the context menu:

- Copy data item(s) 
- Paste to data item(s) 

An example of the context menu available on *Plc Connection Items* tree is shown below:



Delete data items

Triggering the *Delete data items*  action with the corresponding button on the right wizard side or *Delete* from the context menu will remove the currently selected node from the *Data Items* namespace tree.

© Softing Industrial Automation GmbH

Attention: The root node of the namespace tree can not be deleted.

Attention: A node which is currently used in a data processing or data destination action can not be deleted.

Copy

The *Copy*  action is provided in the context menu of the *Plc Connection Items* tree. Triggering this action will copy the selected node to the local clipboard. All the properties of the node are copied. Also, if the node contains children, the entire branch is copied.

Note: The root node of the namespace tree does not provide a *Copy* action.

Paste

The *Paste*  action is provided in the context menu of the *Plc Connection Items* tree. This action will add the node currently stored in the local clipboard as a a child of the currently selected node. Whether the action is available in the context menu depends on the type of the currently selected node and the type of the node in the clipboard.

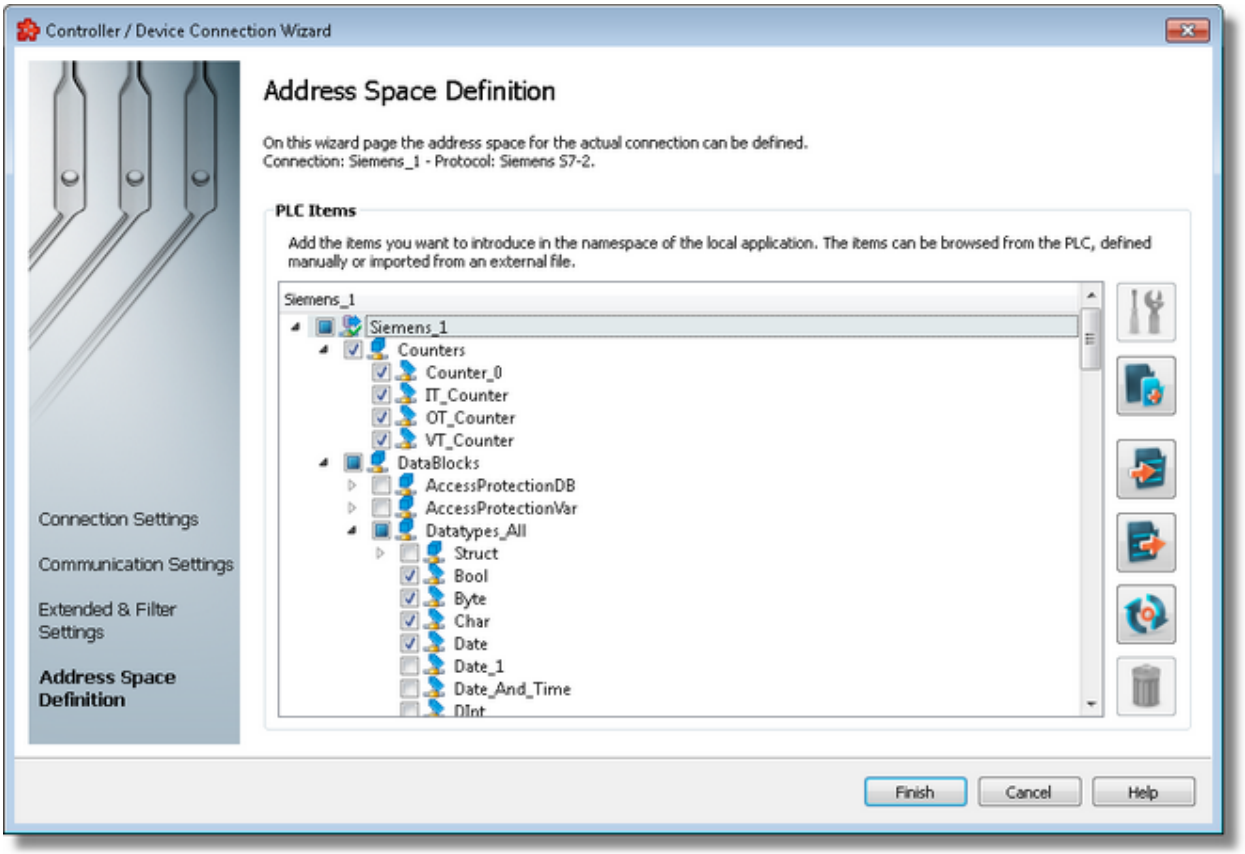
Note: The *Paste* action is only available if a node was previously copied to the clipboard.

If the *Paste* action is performed on a node which already has a child of the same name as the node in the clipboard, the pasted node is renamed automatically by adding a numerical suffix to the original name.

```
E.g. NodeA, NodeA01, NodeA02, ...
```

<<< Siemens Device Wizard - Extended Settings

On this wizard page the filter data items for the S7-2 Siemens connection can be defined. Select the data items you want to introduce into the namespace of the local dataFEED OPC Suite application from the corresponding tree control (external server must be running) or import the filter definition from an external file.



If the PLC device is connected, the namespace is shown in the tree under the root node named after the actual connection. The tree is expandable and each item can be selected to be included in the filter file or not. The selection is downward recursive, which means that when you include a node all subsequent nodes, tags and properties in that particular branch of the namespace tree will automatically be included.

If the connection to the device failed, the namespace tree will only contain the server root node. However, it is possible at any time to try reconnecting to the external server and refreshing the data by using the *Refresh data items* button .

Operation

Depending on the current selection in the *External Server Items* tree, the following operations are possible using the buttons on the right, the context sensitive menu or the corresponding shortcuts:

- Properties of the selected item (Ctrl+P)
- Add a new item (Ctrl+A)
- Import from file (Ctrl+I)
- Export to file (Ctrl+E)
- Refresh data items (Ctrl+R)

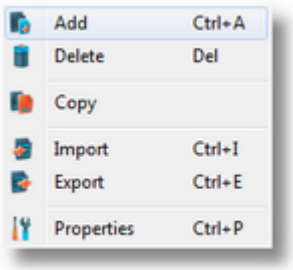
© Softing Industrial Automation GmbH

- Delete data items (Del)

Two more operations are supported, which are only accessible from the contextual menu:

- Copy data item(s)
- Paste to data item(s)

An example of the contextual menu available on *Data Items* tree nodes is shown below:



Delete data items

Triggering the *Delete data items* action with the corresponding button on the right wizard side or *Delete* from the context menu will remove the currently selected node from the *Data Items* namespace tree.

Attention: The root node of the namespace tree can not be deleted.

Attention: A node which is currently used in a data processing or data destination action can not be deleted.

Copy

The *Copy* action is provided in the context menu of the *External Server Items* tree. Triggering this action will copy the selected node to the local clipboard. All the properties of the node are copied. Also, if the node contains children, the entire branch is copied.

Note: The root node of the namespace tree does not provide a *Copy* action.

Paste

The *Paste* action is provided in the context menu of the *External Server Items* tree. This action will add the node currently stored in the local clipboard as a a child of the currently selected node. Whether the action is available in the context menu depends on the type of the currently selected node and the type of the node in the clipboard.

Note: The *Paste* action is only available if a node was previously copied to the clipboard.

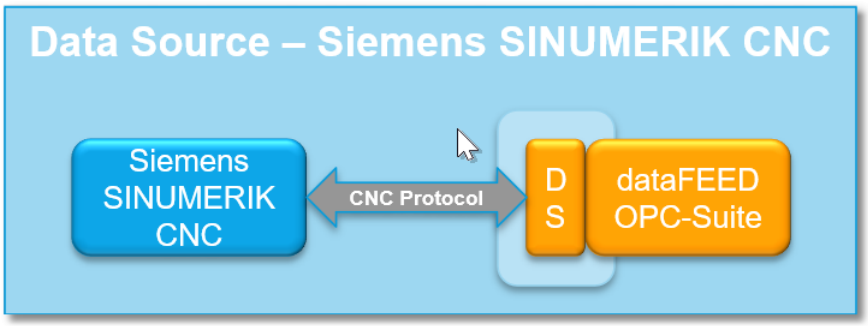
If the *Paste* action is performed on a node which already has a child of the same name as the node in the clipboard, the pasted node is renamed automatically by adding a numerical suffix to the original name.

```
E.g. NodeA, NodeA01, NodeA02, ...
```

<<< Siemens Device Wizard - Filter Settings

7.3.1.2 Siemens SINUMERIK CNC Devices

This section of the Softing dataFEED OPC Suite Online Help describes how to configure Siemens SINUMEMRIK CNC devices within Softing dataFEED OPC Suite.

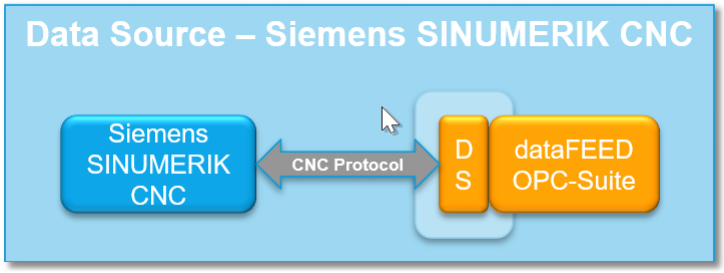


Controllers and devices data source makes it possible to exchange data between the field devices (Siemens SINUMERIK CNCs) of a wide variety of manufacturers and OPC clients via Ethernet TCP/IP.

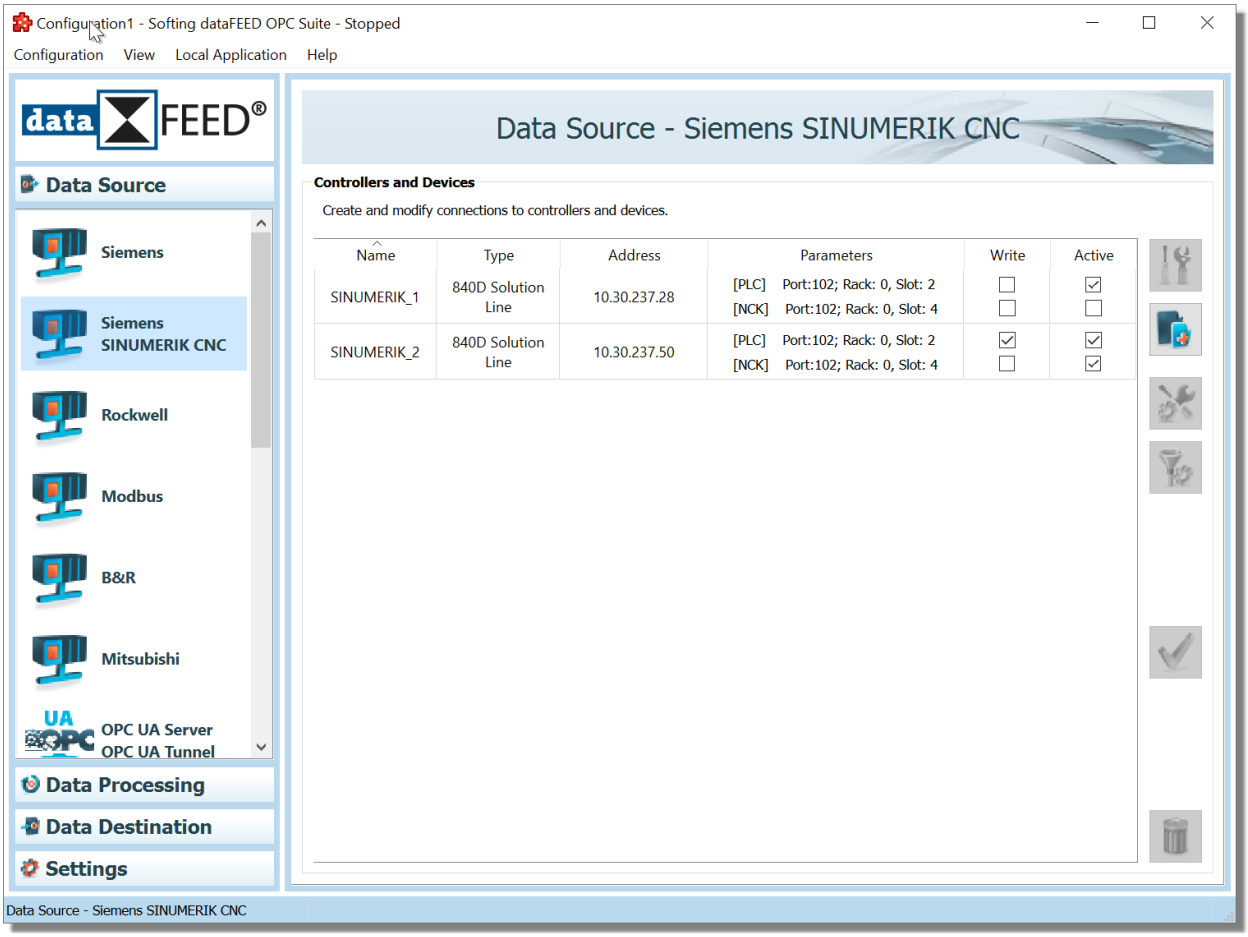
A detailed description on how to configure access to Siemens SINUMERIK CNC devices functionality can be found in the Configuration section.

© Softing Industrial Automation GmbH

7.3.1.2.1 Configuration



Select Siemens SINUMERIK CNC devices from the Data Source group in the navigator on the left to start configuring the connection to an external Siemens SINUMERIK CNC device. The Data Source - Siemens SINUMERIK CNC Device main user interface is displayed. The main element of this page is the External Devices list. Each entry shown in the list of the configuration section represents a connection to an external device.



Note: The Siemens SINUMERIK CNC devices list is visible and editable only when a configuration has been loaded.

Defining a Siemens SINUMERIK CNC connection requests to define its connection name, which will be visible on the device list table in "*Name*" column. The connection name must be unique over the entire configuration (no other data source connection could use the same name as connection name identifier).

Attention: A Siemens SINUMERIK CNC connection could enable parallel connections to PLC and NCK device components. Disabling both device components is disabling the entire connection. For each of the device component will be created on the address space tree a connection root node using the Connection Name post fixed with a corresponding suffix ("***-PLC***" for PLC device component and "***-NCK***" for NCK device component).

Defining a device connection using a connection name (default or not) will generate on the address space, directly under the root node, 2 different device component connections - acting and working as individual connections.

Sample:

Defining a Siemens SINUMERIK CNC connection using **SINUMERIK_1** as connection name, will create into the address space under the root node, 2 separate device component connections:

- **SINUMERIK_1-PLC** - the connection to the PLC device component.
- **SINUMERIK_1-NCK** - the connection to the NCK device component.

The PLC/NCK device component could be individually activated/deactivated.

Attention: The dataFEED OPC Suite configuration tool address space shows the device components (PLC and NCK) even if these components are disabled (not activated). At dataFEED OPC Suite Runtime only the enabled (active) device component are visible (a disabled connection device component being completely ignored at initialization time).

The "**-PLC**"/"**-NCK**" suffixes are not configurable - modifying them is not allowed.

Note: Device (Connection) name must be unique within a dataFEED OPC Suite configuration. Also, the name could not contain address space delimiters, multi-byte characters or empty spaces.

The supported Siemens SINUMERIK CNC device type is:

- Siemens SINUMERIK CNC 840d Solution Line.

The device type is visible on device type column (second column - ***Type***).

On the third column (***Address***) is addressed the actual device IP Address which can be IPv4 or IPv6 address or HostName (with or without DHCP address).

On the forth column (***Parameters***) are visible information about configured communication parameters:

- port (default port 102 - not configurable).
- rack and slot (used for defining the communication TSAP parameter).

The access rights can be defined for each individual device component (PLC and NCK). By default all the connections are read only - this can be modified by enabling the write access on

© Softing Industrial Automation GmbH

fifth column (***Write***). Enabling the write access per device component allows write access on writable nodes.

Attention: Enabling the write access on a device component (PLC or NCK) is enabling the write service only for nodes with write access right, read only nodes still not possible being written.

The device component (PLC and NCK) can be activated/deactivated from the sixth column (***Active***). Activating/Deactivating the PLC or NCK component has no impact on the address space shown by the dataFEED OPC Suite Configuration Tool, still at Runtime, only the activated connections are visible on the address space.

Note: Disabling a device component (PLC or NCK) is allowed only if items from its address space are not used on other functionality.

Operations

From the Siemens SINUMERIK CNC devices main user interface certain operations can be performed over the selected connection by using the buttons on the right, by accessing the context sensitive menu or by pressing the keyboard shortcuts. The following actions are available:

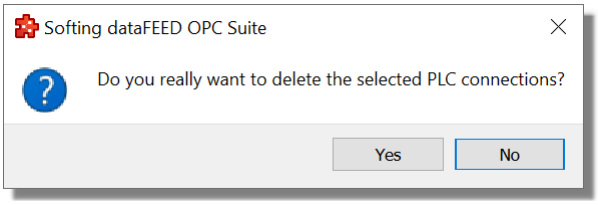
- Change the properties of the selected data source by using the Properties button  , pressing the keyboard shortcut CTRL+P or selecting Properties from the context menu.
- Add a new connection by using the *Add* button  , pressing the keyboard shortcut CTRL+A or selecting *Add* from the context menu.
- Change advanced settings for the selected data source by using the *Advanced settings* button  . (please note that advanced settings are connection specific while the global settings applies to all Siemens SINUMERIK CNC connections).
- Delete the selected data source by using the *Delete* button  , pressing the DEL/DELETE key on the keyboard or selecting *Delete* from the context menu.
- Address space definition button  , by pressing it the Siemens SINUMERIK CNC device Wizard will be opened in it's last page. For more details see Address Space Definition page.
- Use the test connection button  to check the availability of the specified CNC.

Note: If no device connection is selected, the *Add a new data source* button is the only one available.

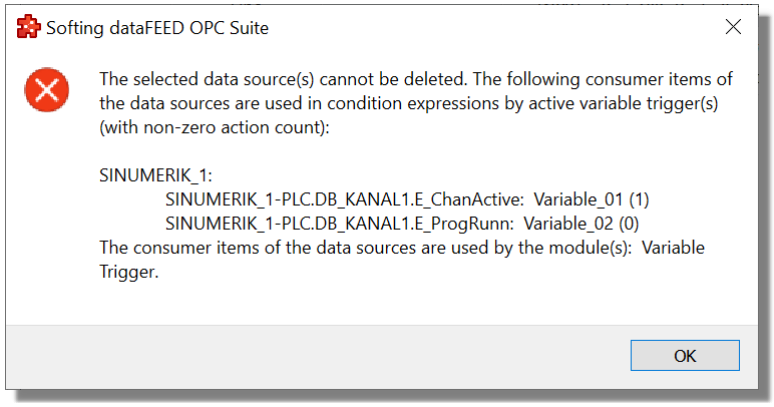
The ***Delete*** button  will remove the selected device connections from the list.

The user can select more than one server connection with the help of the CTRL or SHIFT keys.

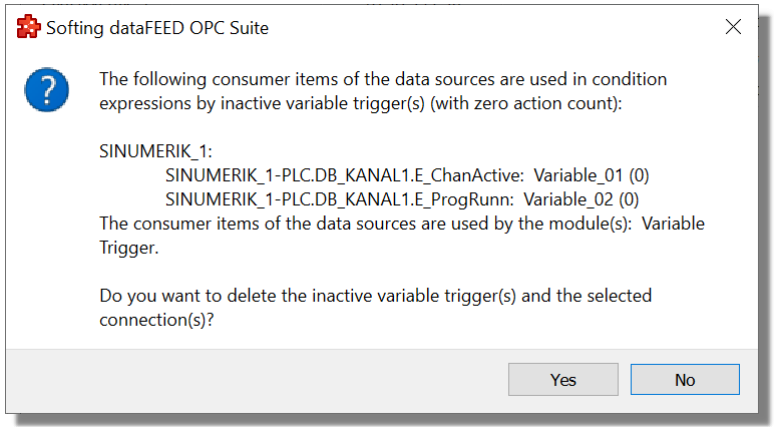
A dialog will then prompt the user to confirm the deletion.



A dialog will inform the user that the data sources cannot be deleted, if consumer items and/or system items of the selected data sources are used by active variable triggers or other modules.



A dialog will prompt the user to confirm deletion of the data sources and inactive variable triggers, if consumer items and/or system items of the selected data sources are used only by inactive variable triggers.

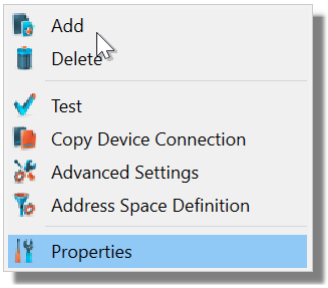


Attention: A device connection cannot be deleted if at least one data item defined in its namespace is used in a Processing or Data Destination action.

All these actions are also provided in the context menu of the External device list. Which actions are available depends on the current selection in the list.

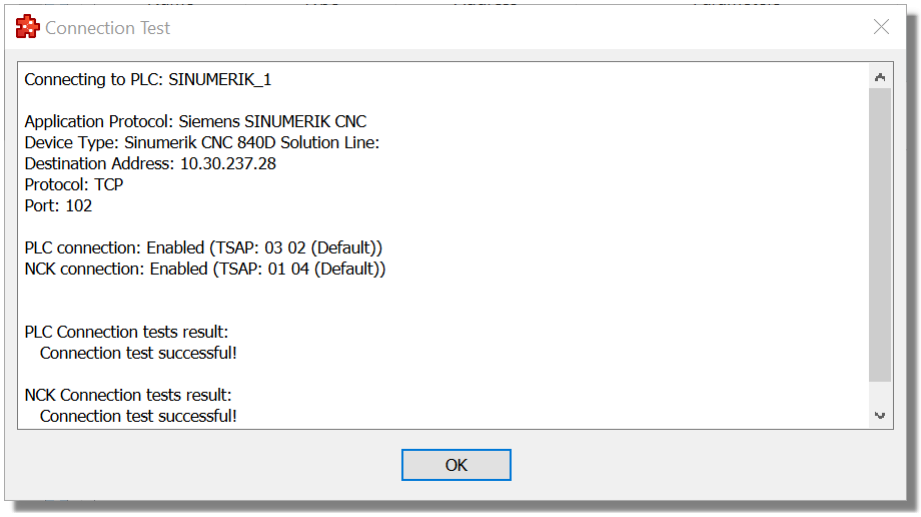
An example of the context menu can be seen below:

© Softing Industrial Automation GmbH



- ***Test*** device connection  . This action is available from context menu or corresponding table's button and leads to a dialog where the connection to the CNC is tested.

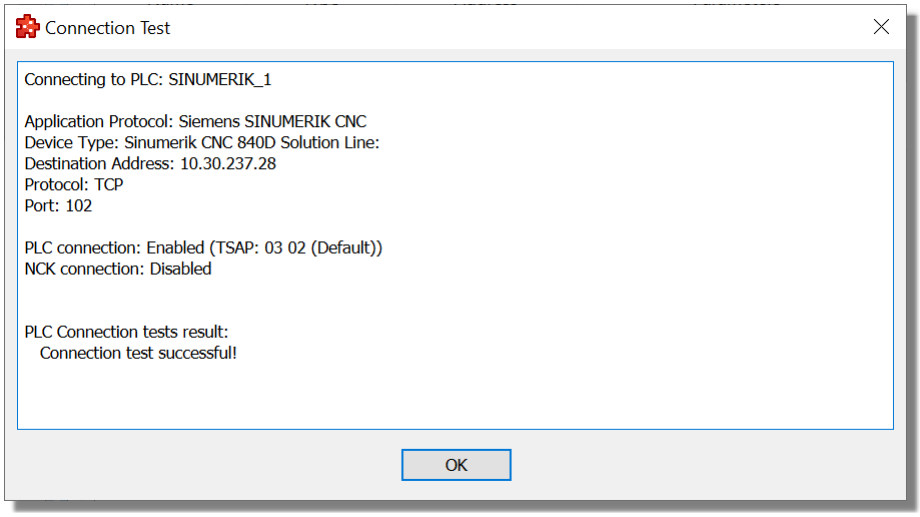
Attention: Only the activated device components (PLC and/or NCK) are tested.



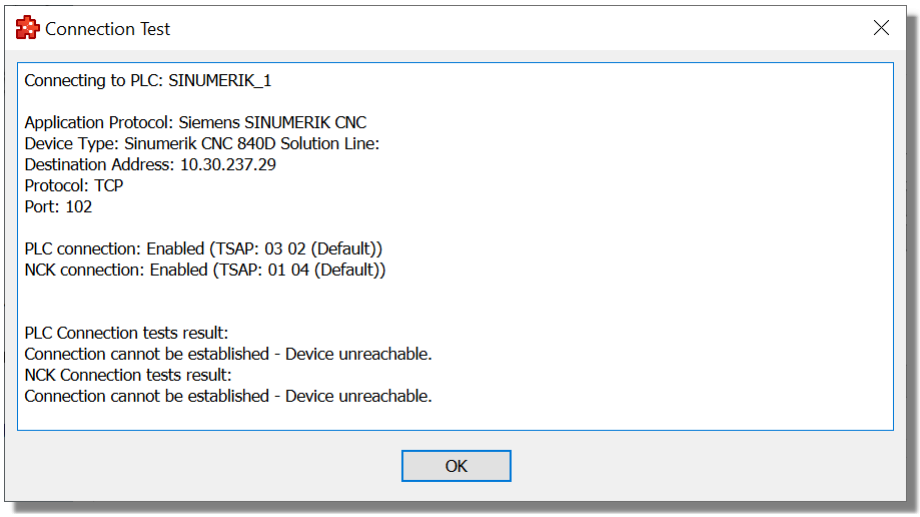
On the Connection Test dialog are visible multiple information about current Siemens SINUMERIK CNC connection:

- Device Type - currently being supported only SINUMERIK CNC 840d Solution Line.
- Device IP Address
- Protocol - TCP being the only one supported.
- Port - communication port (102).
- PLC device component - Status (Enable/Disable) and its corresponding TSAP parameter.
- NCK device component - Status (Enable/Disable) and its corresponding TSAP parameter.

When one of the device component (PLC or NCK) is disabled - the connection to the corresponding device component is not tested.



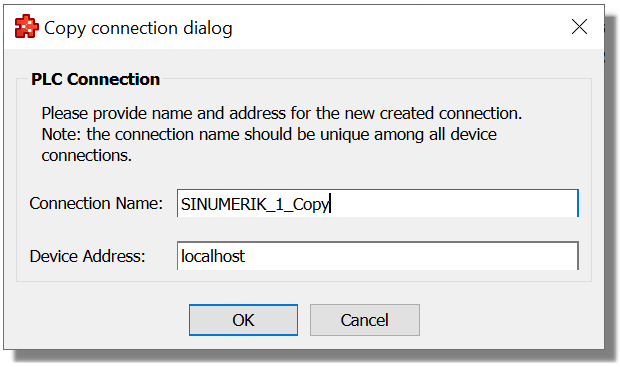
A failure on testing the SINUMERIK CNC connection is signaled with additional details about the failure reason where possible.



Attention: While testing the connection a real connection is tried being establish to the Siemens SINUMERIK CNC device (only active device component). This operation could affect the communication to the device on other products when the CNC is not supporting multiple connections or all the available slots are used by other application. Also, the test connection result could be affected by the available slots on the SINUMERIK CNC device.

- Copy device connection  . This action is available only in the context menu and it leads to a dialog where the user may provide a name and device address. See the screen shot below:

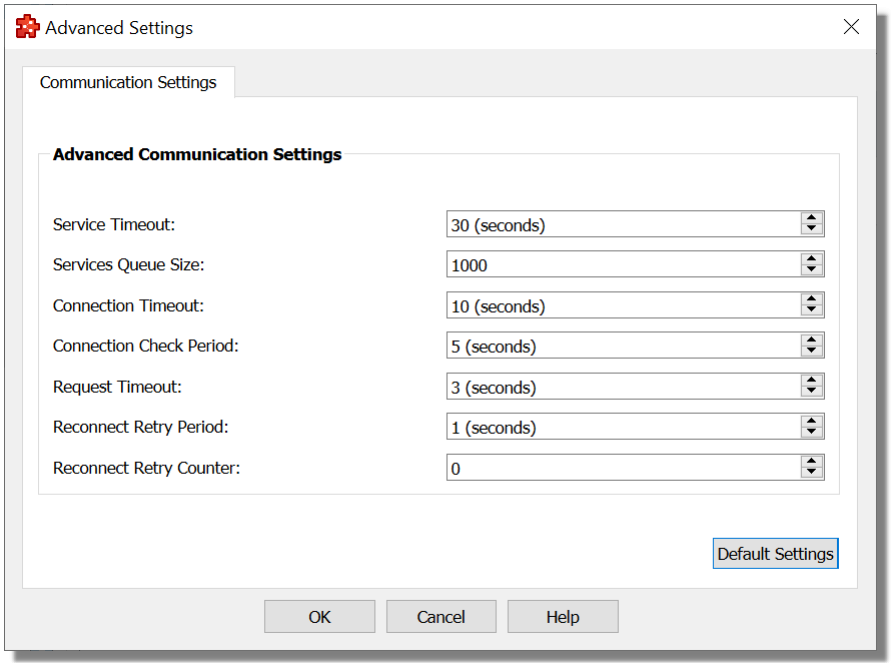
© Softing Industrial Automation GmbH



The proposed name is the original connection's name post fixed with "_Copy" and the default address will be "localhost".

Copying a Siemens SINUMERIK CNC connection is copying all the connections parameters and its defined address space.

- Advanced Settings  of selected Siemens SINUMERIK CNC connection. The advanced communication settings can be configured by opening the specific dialog.



Parameter Name	Description	Default value and value range (if applicable)
Service Timeout	Maximum time in seconds allowed for a service being completely executed. One service might be spitted into multiple protocol requests based on number of operations which should be processed in a single call - therefore it	Default value: 30 seconds (configurable at second range) Range: 1 to 600 seconds

	is advisable to have a larger value for this parameter.	**Restriction**: The value should be at least equal (≥) to the double of Request Timeout value.
Services Queue Size	Maximum number of concurrent pending services for execution. Any additional service call made outside of the services queue is automatically rejected.	Default value: 1000 Range: 100 to 5000
Connection Timeout	Time in seconds the connection API waits for connection to be made. Also, it is used as watchdog operation for Siemens SINUMERIK CNC connections. In case there is no traffic on the network (beside the keep alive mechanism) for an interval longer than the Connection Timeout value the connection will be automatically disconnect into the communication module. This should normally not happen as the application triggered keep alive mechanism is active.	Default value: 10 seconds (configurable at second range). Range: 1 to 600 seconds **Restriction:** The value should be at least equal (≥) to the value of Connection Check Period multiplied with the maximum between 1 and Reconnect Retry Counter value.
Connection Check Period	Time in seconds used for monitoring the PLC connection and its availability on the network. The keep alive mechanism is handled sending a message for each Connection Check Period.	Default value: 5 seconds (configurable at seconds range). Range: 1 to 600 seconds **Restriction:** The value should be at least equal (≥) to the Request Timeout value. **Restriction:** The value should be at most equal (≤) to the Connection Timeout value divided with maximum between 1 and the Reconnect Retry Counter value.
Request Timeout	Time in seconds used by the protocol communication module to complete a request at the device layer. This is the maximum time allowed for the CNC to execute and confirm execution of a	Default value: 3 seconds (configurable at second range). Range: 1 - 300 seconds

© Softing Industrial Automation GmbH

	request. It should be adjustable based on network latency and PLC performance.	**Restriction:** The value should be at most equal (≤) to the minimum between the Connection Check Period value and Service Timeout value divided by 2.
Reconnect Retry Period	Delay between consecutive reconnecting attempts to the CNC.	Default value: 1 second (configurable at second range). Range: 0 to 60 seconds
Reconnect Retry Counter	The number of connection attempts before to mark a connection as not accessible.	Default value: 0 (Referring to the default retries counter. In this case the communication module decides itself when and how many times a reconnection attempt is tried). Range: 0 to 60 **Restriction:** The value should be at most equal to the Connection Timeout value divided with the Connection Check Period value.

Note: All restrictions are automatically applied by the dataFEED OPC Suite Configurator. Values outside of these restrictions are not permitted.

$$1000 <= \text{RequestTimeout} <= \text{ConnectionCheckPeriod} <= \frac{\text{ConnectionTimeout}}{\text{ReconnectRetryCounter}} \quad \text{(only if value > 0)}$$

$$1000 <= \text{RequestTimeout} <= \frac{\text{ServiceTimeout}}{2}$$

The default settings can be restored by using the **[Default Settings]** button.

- Address Space  of selected Siemens SINUMERIK CNC connection. The address space of selected Siemens SINUMERIK CNC connection can be used and modified.

The address space of both PLC and NCK components are editable:

- PLC component - address space definition.
- NCK component - address space definition.

Attention: Changing on the address space definition is visible at Runtime only after an application restart.

Attention: The address space is visible for both device components (PLC and/or NCK) even if the corresponding component is not activated.

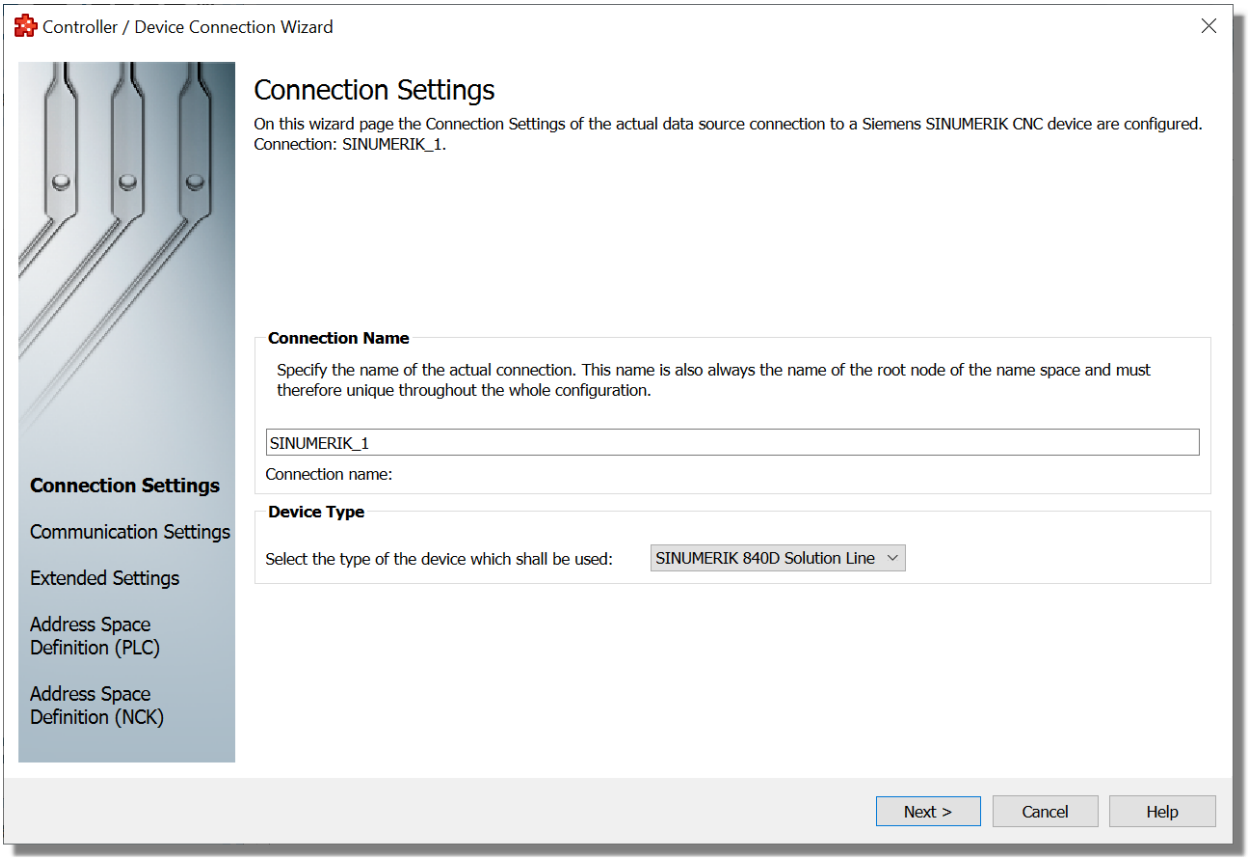
7.3.1.2.2 SINUMERIK Device Wizard

With the Siemens SINUMERIK CNC Device Wizard you can create and modify a data source connection to an external SINUMERIK CNC device:

- Specify connection settings @ Siemens SINUMERIK CNC Device Wizard - Connection Settings
- Specify communication settings @ Siemens SINUMERIK CNC Device Wizard - Communication Settings
- Extended settings @ Siemens SINUMERIK CNC Device Wizard - Extended Settings
- Address Space Definition of PLC device component @ Siemens SINUMERIK CNC Device Wizard - Address Space Definition (PLC)
- Address Space Definition of NCK device component @ Siemens SINUMERIK CNC Device Wizard - Address Space Definition (NCK)

On this wizard page the connection settings (Connection Name and Device Type) can be configured.

© Softing Industrial Automation GmbH



Connection Name

An unique identifier for the current connection. In case the provided name is already used by other connection (for all available data sources), the user is not allowed to go into the next wizard page.

Based on activated device components (PLC and NCK), on the address space, directly under the root not will be created device component connection root nodes - this nodes will be created by adding the corresponding suffix ("***-PLC***" or "***-NCK***") to the connection name.

Defining a Siemens SINUMERIK CNC connection using **SINUMERIK_1** as connection name, will create into the address space under the root node, 2 separate device component connection:

- **SINUMERIK_1-PLC** - the connection to the PLC device component.
- **SINUMERIK_1-NCK** - the connection to the NCK device component.

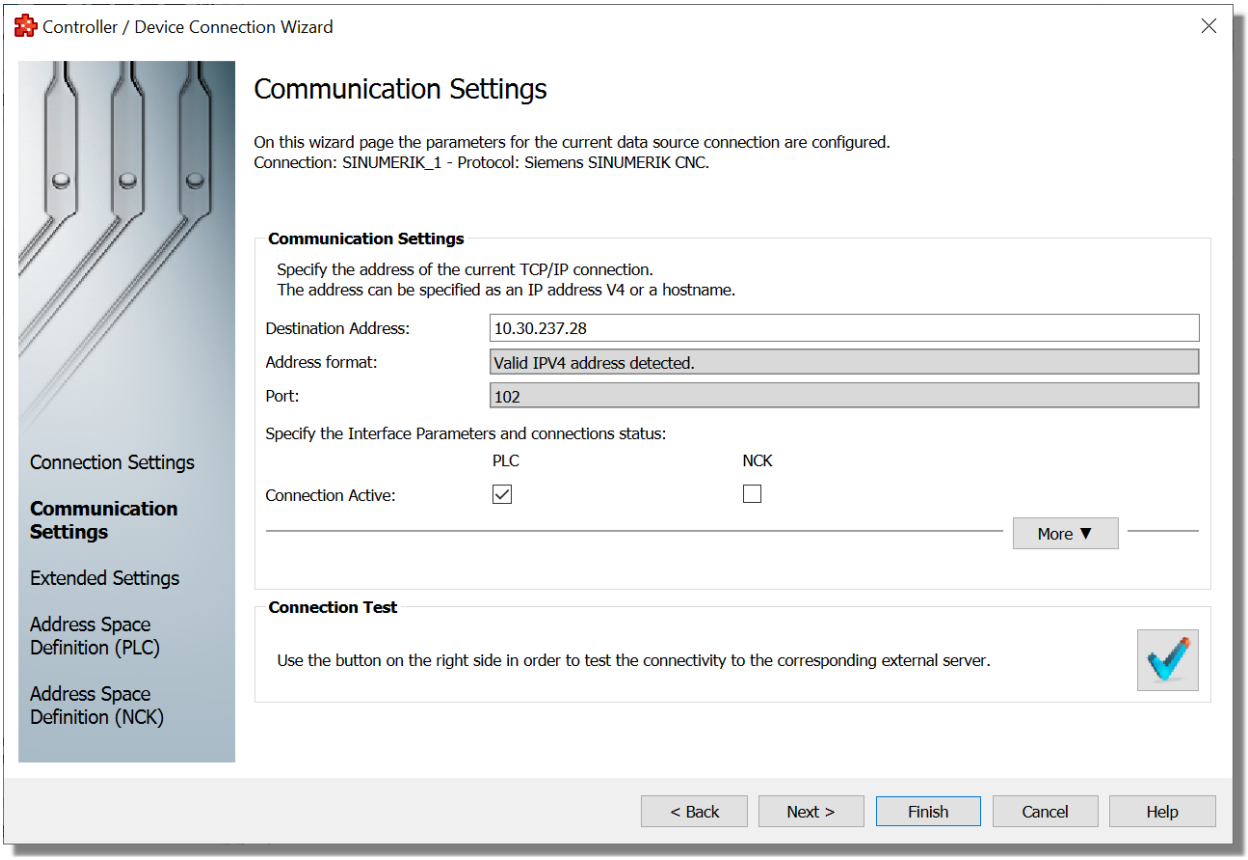
Note: Due the way how the address space is created for both device components it is mandatory that not only the connection name is unique - also the device component connection name should be unique too (applied on all defined data sources).

Device Type

Specifies the supported Siemens SINUMERIK CNC device types. Currently only the ***Siemens SINUMERIK 840D Solution Line*** is supported.

Siemens SINUMERIK CNC Device Wizard - Communication Settings >>>

On this wizard page various parameters for the current TCP/IP connection are configured. Other communication settings can be configure too.



The communication settings are configurable:

- **Destination Address: IP address or Hostname**

The IP Address of the target station (IP address of the PLC) or the DNS name of the target station is entered. If the computer was configured for DNS and if a DNS server is available on the network, the symbolic name of the target station can also be entered.

Additional information related to IP Address set up, available under IP address settings.

- **Address Format**

Read-Only field; automatically detects the format of destination address.

- **Port**

Read-Only field; always 102.

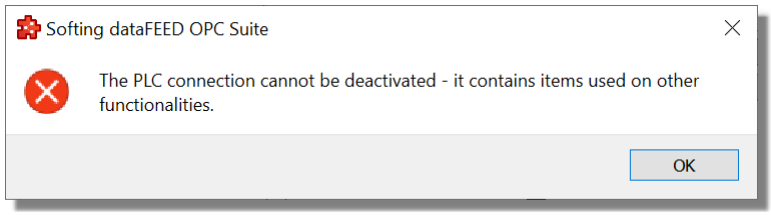
- **Connection Active**

© Softing Industrial Automation GmbH

The device components (PLC and NCK) could be activate/deactivated individually. Each of the supported device component can be activated or deactivated.

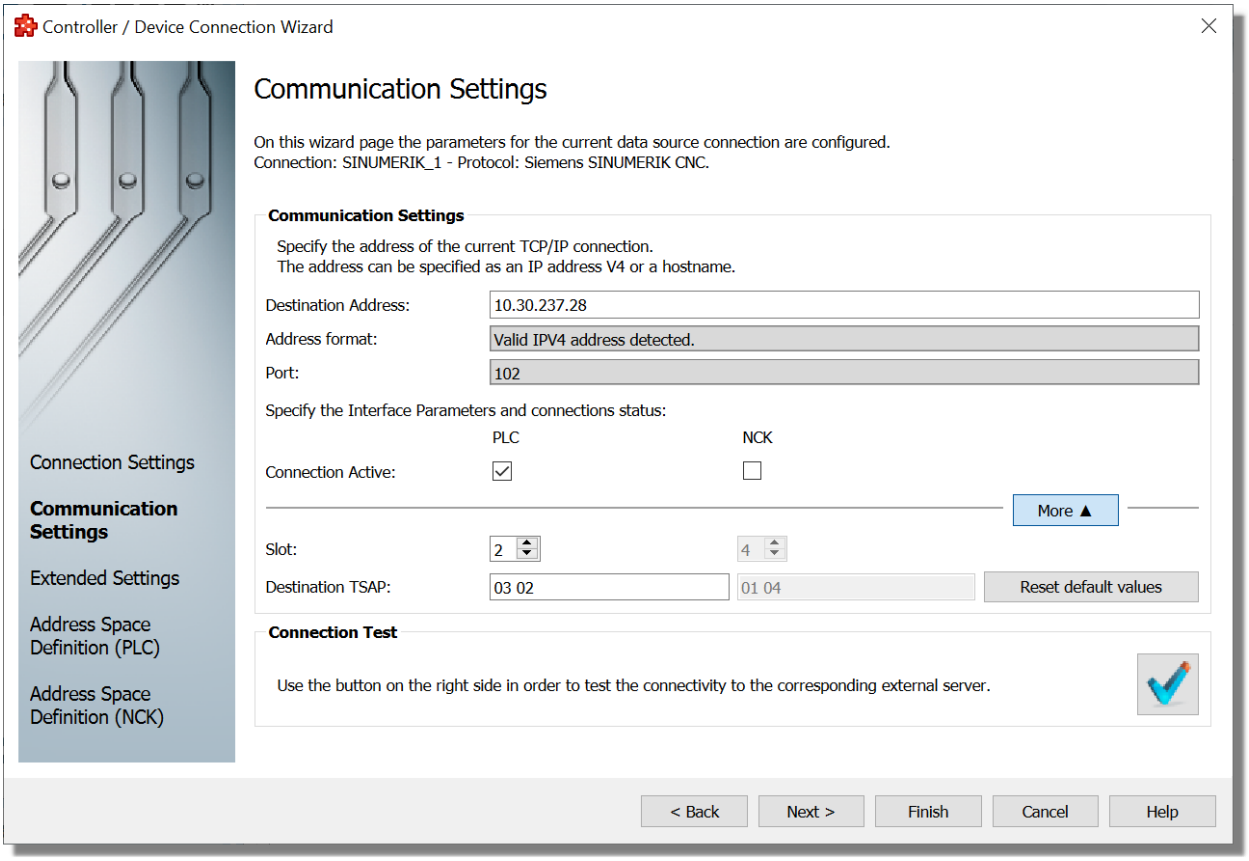
Attention: Deactivating a device component (PLC or NCK) is not removing the corresponding address space from the complete address space visible on Configuration Tool. Still, a disabled device connection component is completely ignore at Runtime initialization (a connected data destination consumer will not see at browse time a disabled connection to a specific device component).

A device component cannot be deactivated when items from its corresponding address space are used into processing and destination functionality as placeholders or triggers. The detection of used connection (device component (PLC or NCK) connection which cannot be deactivated) is detected while trying to deactivate the component - this will be signaled on an error message.



The advanced communication settings for each available device component are possible being set by pressing the "**More**" 

button. Immediately after, communication settings are available on the Communication Settings wizard page.



For each activated component device connection it could be configure the corresponding TSAP settings. The TSAP can be change by modifying the ***Slot*** (which is part of defined TSAP) or by editing the ***Destination TSAP***.

Slot:

Slot number in which the CPU resides, a number between 0 and 31. For PLC component (using S7-300 PLC family the second slot is always used.

Local TSAP, remote TSAP:

TSAPs (Transport Service Access Points) are addresses which are used within the transport layer to address applications. TSAPs are required for RFC1006 connections. To set up a connection, both the local TSAP and the TSAP of the communication partner must be specified. The following must be adhered to.

Attention: The TSAPs must coincide crosswise before the connection can be established!

The local TSAP of the one system must correspond to the remote TSAP of the other system.

Pressing the "***Reset default values***" button is resetting the TSAP settings to the default values:

- PLC device component: 03 02.
- NCK device component: 01 04.

© Softing Industrial Automation GmbH

Note: The TSAP settings are configurable only for activated device component. Disabling a device component (PLC or NCK) will remove the possibility to configure its corresponding TSAP.

Connection test

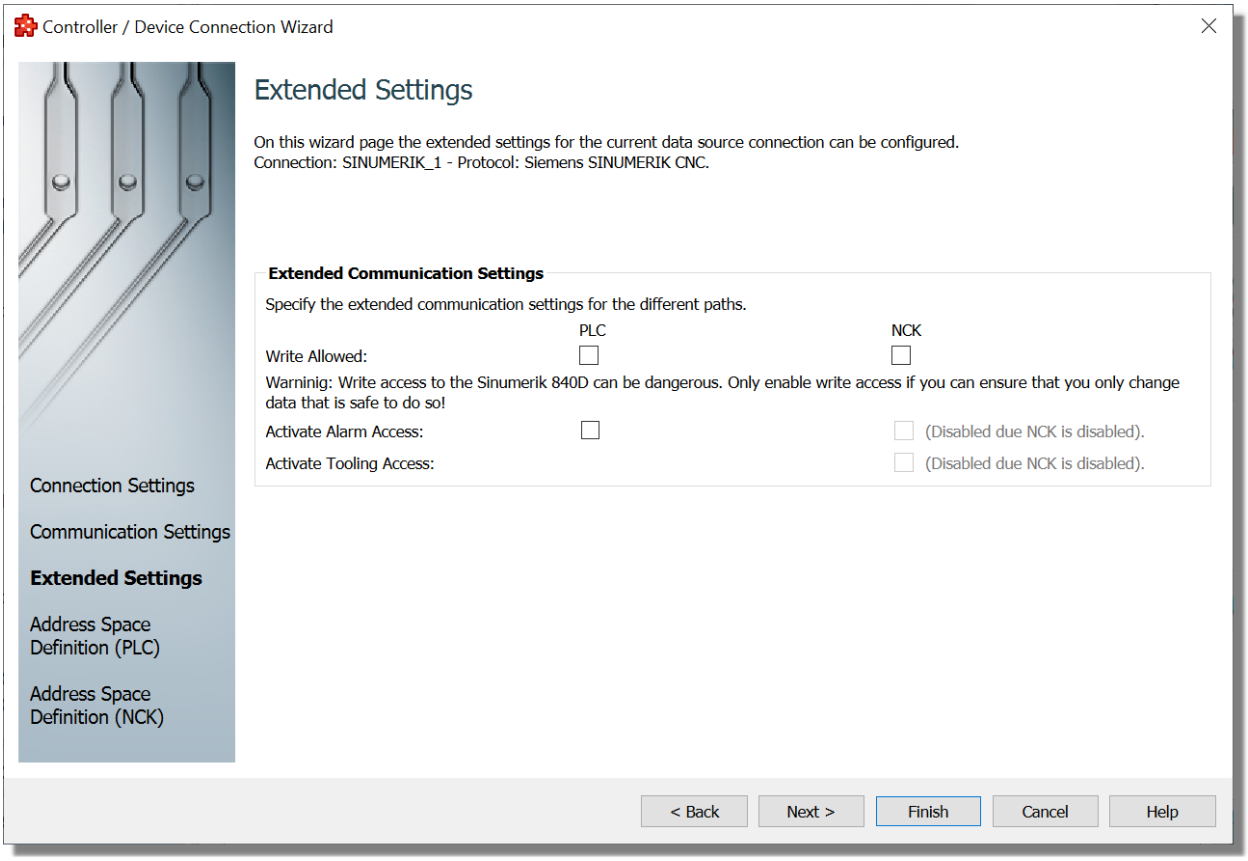
Convenience button  provided to check if the current settings can be used to connect at device.

The same functionality is provided in main configuration overview test connection.

Note: Only the active device components connections are tested.

<<< Siemens SINUMERIK CNC Device Wizard - Connection Settings Siemens SINUMERIK CNC Device Wizard - Extended Settings >>>

On this wizard page the extended settings for the current data source connection can be configured. The settings can be individually applied for each available device component connection (PLC and NCK).



Write Allowed

Enables the write access rights on a CNC connection (PLC or NCK device component). By default the write access rights is disabled on both PLC and NCK components.

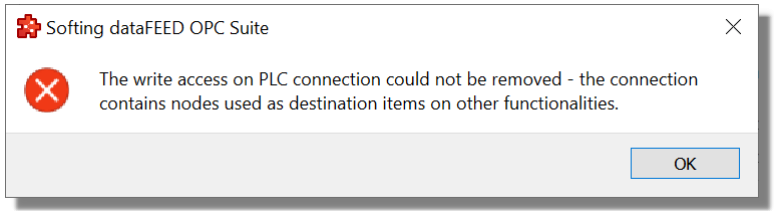
Enabling the write access rights on a defined connection, it allowing the write access on the address space definition on items which are not read only.

Attention: Activating the Write Allowed option is not automatically update the write access rights on the defined address space - this option is only allowing the write access right on items which has the write access rights, the read only items are not automatically updated.

Attention: Activating the Write Allowed option on NCK component can be dangerous for the Siemens SINUMERIK CNC device - writing value of items which are not intended being written could have a major impact on the CNC functionality.

The option can be enabled on individual components even if the connection is disabled.

Note: Trying the disable the write allowed option on a connection which uses some items from corresponding connection address space as destination items is not allowed by the dataFEED OPC Suite Configuration Tool. This is signaled with a proper error message.



Updating the write access rights has an immediate response on the address space - the corresponding address space being modified by updating the items access rights.

By default the write allowed option is disabled.

Activate Alarms Access

The alarms nodes of specific device component (PLC or NCK) connection could be activated or deactivated. Activating the alarms access is allowing access to the CNC alarms handled as exposed nodes or variables on the address space.

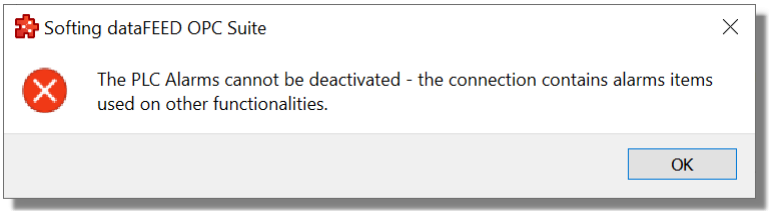
Attention: The CNC alarms are handled based on the exposed address space as OPC UA/DA variables not as OPC UA/DA alarms. Basically, the variables can be used on OPC Subscriptions or IO operations - not acting as real OPC Alarms.

Immediately after the option is enabled the result is visible on the exposed address space - the alarms nodes being visible.

© Softing Industrial Automation GmbH

Note: The alarms nodes are obtained directly from the CNC device - on offline connections those nodes cannot be obtain, as result the nodes are not visible on address space for offline connections.

Disabling the alarms is not allowed in scenarios when alarm nodes are using on other functionality as placeholder or triggers. This is signaled with a proper error message.



Note: Disabling a device component (PLC or NCK) is also disabling the possibility to activate the alarms on corresponding device component (PLC or NCK).

By default the Alarms are deactivated for both PLC and NCK connection components.

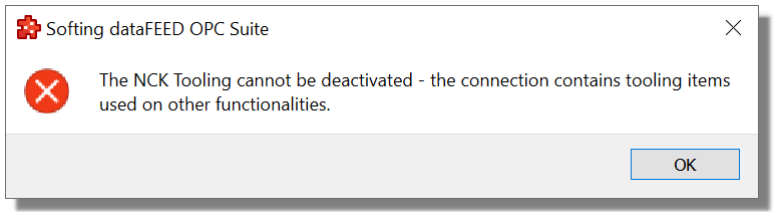
Activate Tooling Access

The Tooling functionality can be enabled on NCK device component only.

Enabling the tooling access, is automatically updating the address space for corresponding NCK device component by adding the specific ***Tooling*** nodes into the exposed address space.

Note: The Tooling nodes are obtained directly from the CNC device - on offline connections those nodes cannot be obtain, as result the nodes are not visible on address space for offline connections.

Disabling the tooling is not allowed in scenarios when tooling nodes are using on other functionality as placeholder or triggers. This is signaled with a proper error message.



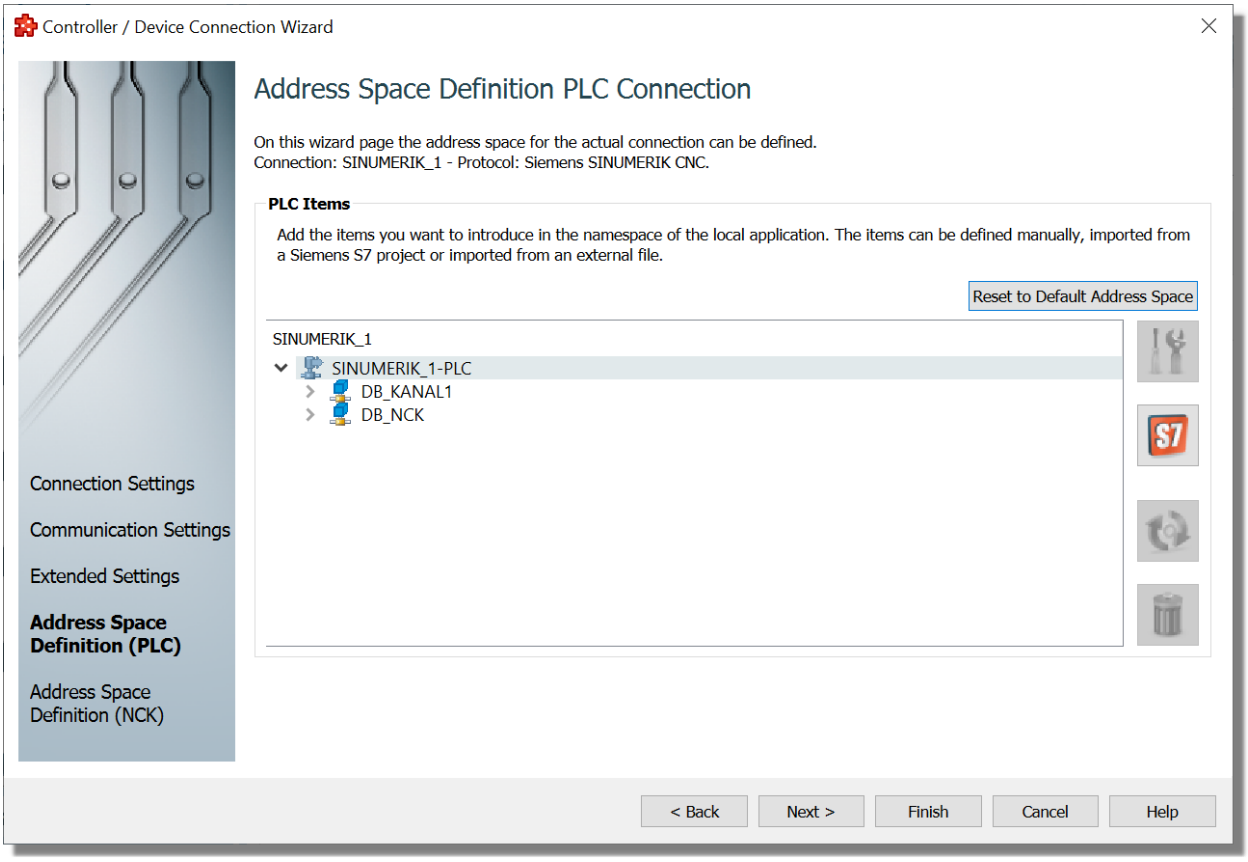
Attention: The tooling nodes are initialized after connection initialization with information from the connected CNC (NCK device component). A set of information is read from device and used to tooling initialization. Due this reason, the tooling nodes requires some additional time to be initialized. Immediately after the connection initialization the Tooling nodes are not accessible - being accessible only after its initialization. The time needed to initialize them depends on the

availability of CNC, communication capabilities and CNC response. IO and subscribe operations being not allowed until complete initialization.

<<< Siemens SINUMERIK CNC Device Wizard - Communication Settings
Siemens SINUMERIK CNC Device Wizard - Address Space Definition (PLC) >>>

On this wizard page, the address space of the PLC device component is visible and can be manipulated. The address space can be updated by resetting it to the default address space, by importing the project address space (*.s7p or *.sdfi formats) or by removing some item from the current address space.

Note: Address Space manipulation by adding or modifying existing nodes is not allowed.



PLC Connection Items

An editable tree view where the user can change the address space structure or content by resetting to default address space, importing PLC project files or deleting items.

There are three categories of items that may be used in the tree:

- **root node** (having the name of the current connection). This node is read only, the user has just the possibility to add or delete children of this.

© Softing Industrial Automation GmbH

- **nodes:** these elements are associated with branches in tree. They can have children but they are typeless and do not carry any syntax.

Note: The items of type Node represent intermediate nodes in the tree. They are used for hierarchy purposes.

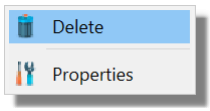
- **tags**: these elements are associated with leaves in tree.

Operation

Depending on the current selection in the *PLC Connection Items* tree, the following operations are possible using the buttons on the right, the context menu or the corresponding shortcuts:

- Display Properties of the selected item (Ctrl+P) 
- Import S7 symbol file 
- Refresh data items (Ctrl+R) 
- Delete data items (Del) 

Display Properties and Delete selected item operations are available also on context menu activated on right click on selected item/node. An example of the context menu available on *PLC Connection Items* tree is shown below:

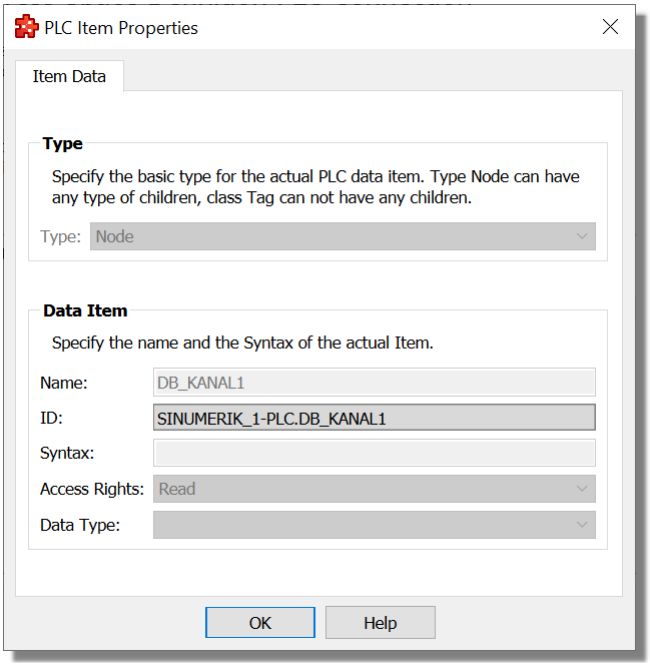


Display Properties of Selected Items/Nodes from PLC Address Space

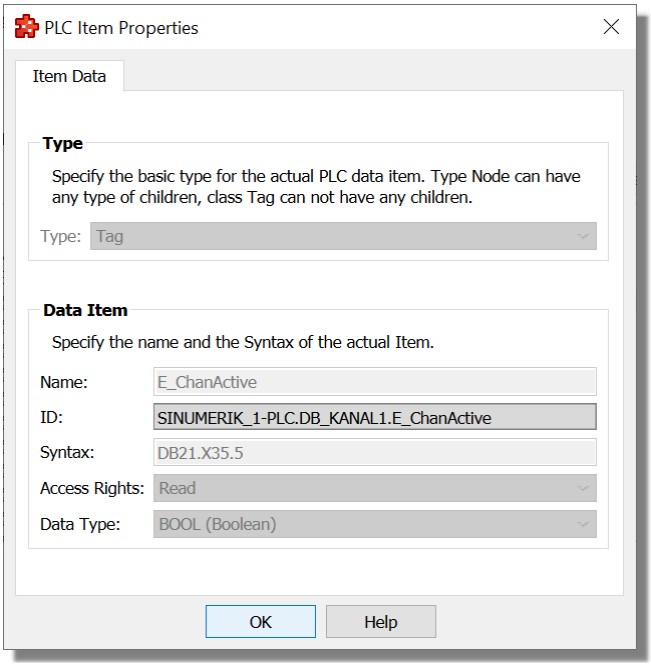
The display property operation is triggered by selecting the node to be modified and then pressing the *Properties* button  , using the shortcut or selecting the *Properties* action from the context menu.

Base on item type, the properties of a variable from address space could be different:

- Node Properties:



- Item Properties:



Note: The properties of a selected node/item from PLC address space are read only (could not be modified).

The available configuration options in this dialog are described in the table below:

Attribute	Description
Type	The type of the data item.

© Softing Industrial Automation GmbH

	The possible types are: *Node* and *Tag*.
Name	The OPC item name identifying the corresponding data item inside the dataFEED OPC Suite address space.
Id	The full OPC item path for the data item inside the dataFEED OPC Suite address space.
Syntax	The PLC syntax for the data item. The PLC syntax is dependent of the protocol for which the current PLC Connection is configured. For more information please click here.
Access rights	The access rights for the data item. The possible values are: *Read*, *Write*, *Read and Write*.
Data type	The data type of the data item. The data type for the item is extracted out of a valid syntax provided by the user.

Note: Items of type *Node* may have child items of any existing type.

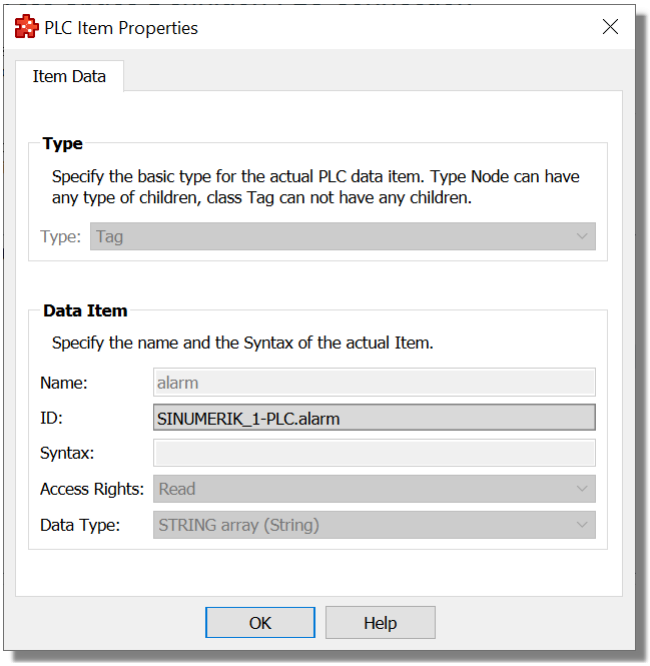
An item from the PLC connection address space may be of any data type covered by the OPC Specifications. The following table describes the available data types mapped to the corresponding VARIANT types from the OPC Specification.

Data type	VARIANT type	Description
I1	*VT_I1*	8 bit signed integer.
UI1	*VT_UI1*	8 bit unsigned integer.
I2	*VT_I2*	16 bit signed integer.
UI2	*VT_UI2*	16 bit unsigned integer.
I4	*VT_I4*	32 bit signed integer.
UI4	*VT_UI4*	32 bit unsigned integer.
I8	*VT_I8*	64 bit signed integer.
UI8	*VT_UI8*	64 bit unsigned integer.
R4	*VT_R4*	32 bit floating point.
R8	*VT_R8*	64 bit floating point.
BOOL	*VT_BOOL*	Boolean.

STRING	*VT_BSTR*	String.
DATETIME	*VT_DATE*	Date and time.
DECIMAL	*VT_CY*	Decimal.
ANY	*VT_VARIANT*	Any data type.

Note: Arrays of fixed and variable length are also supported for these data types.

Activating the PLC alarms is creating a set of nodes into the address space definition of current configuration - those nodes are received directly from the PLC connection and has no Siemens S7 Syntax.



Note: The alarms nodes are not stored in any configuration files - being exclusively receive online from the PLC device component connection. Offline configuration of Alarm Nodes is not possible.

Import S7 Symbol File

The Softing dataFEED OPC Suite Configurator allows the import of S7 PLC connection items from an external STEP 7 symbol file or from a TIA symbol file generated with the Softing dataFEED Exporter.

Note: Make sure the Softing dataFEED Exporter application is installed on that PC where you have installed your **Siemens Totally Integrated Automation Portal.**

© Softing Industrial Automation GmbH

The import action can be triggered by using the *Import S7 symbol file* button  .

Attention: The import from S7 symbol file action applies on the entire address space of the current Siemens SINUMERIK CNC - PLC connection. Any existing items will be automatically removed from the address space prior to performing the import (exception being the items which are used on other functionality as placeholder or trigger).

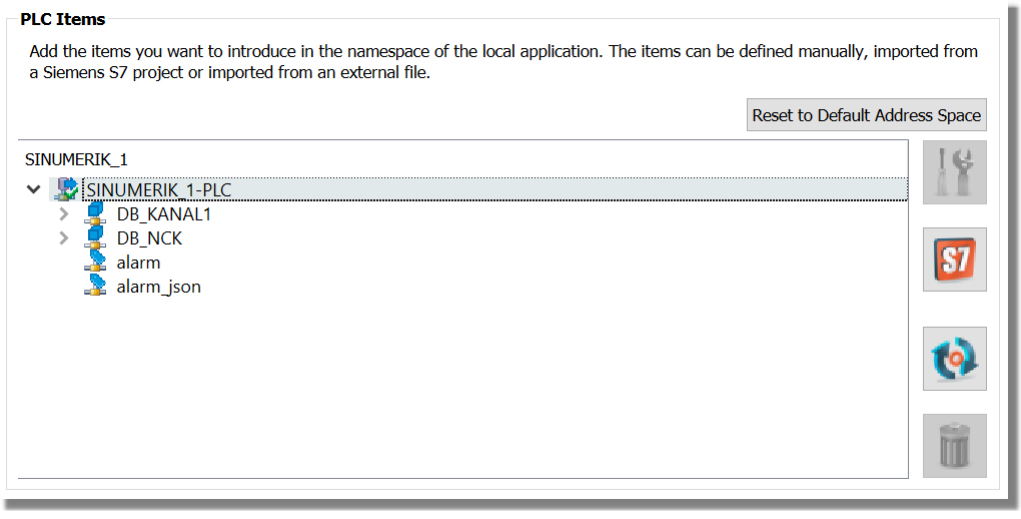
When this import action is triggered, the *Symbol File Import Wizard* is started.

- The Symbol File Import Wizard allows importing different data items:

  - Select the file to be imported @ S7 Project Import Wizard - File Selection
  - Select the PLC Program to use for import @ S7 Project Import Wizard - PLC Program Selection
  - Select the Device to use for import @ S7 Project Import Wizard - Device Selection
  - Select the PLC Blocks to import @ S7 Project Import Wizard - PLC Blocks Selection
  - Examine the import results @ S7 Import Wizard - Import Results

Refresh Address Space (PLC Connection)

The alarms nodes and items (when Alarms are enabled on PLC device component) are dynamically obtain from an online connection. Those nodes can be manipulated (deletion of nodes and items only), also it could be the situation that the alarms available on PLC connection were changed - the alarm nodes could be automatically updated (from current online PLC connection) by pressing the Refresh Address Space  button.



Attention: The refresh address space functionality is applicable only for nodes and items which are retrieve online - on PLC device component being applicable only on alarm nodes. When Alarms are disabled, the Refresh Address Space option is disabled too.

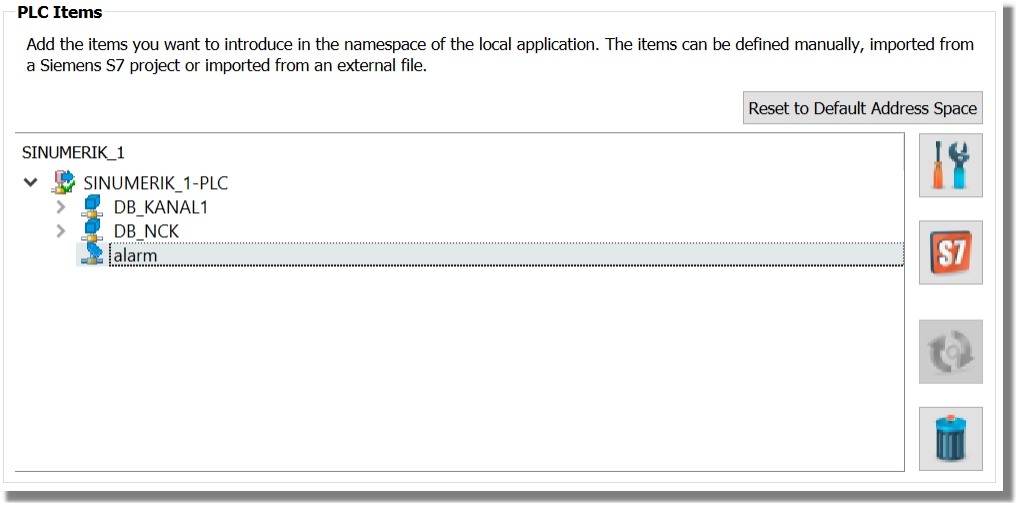
Delete PLC Data Items

Triggering the *Delete data items*  action with the corresponding button on the right wizard side or *Delete* from the context menu will remove the currently selected node from the *Data Items* address space tree.

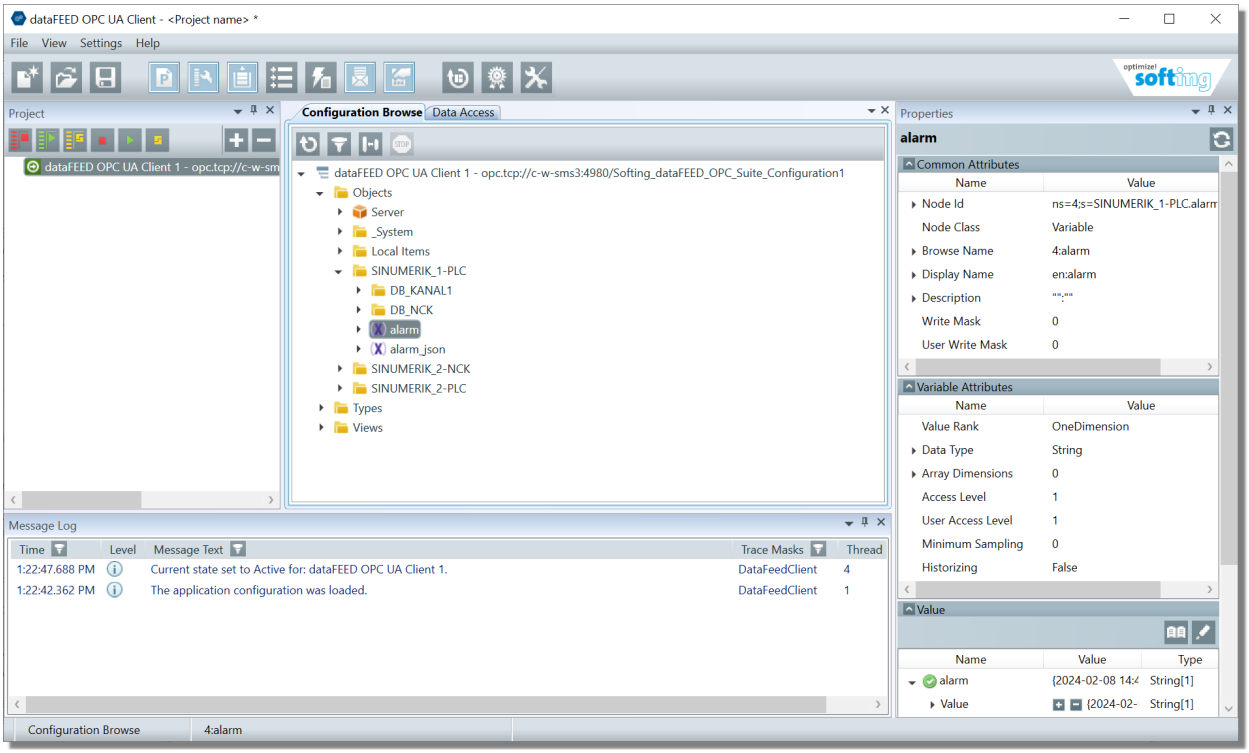
Attention: The root node of the PLC connection address space tree can not be deleted.

NOTE: A node or item which is currently used in a data processing or data destination action can not be deleted.

Attention: Deletion of alarms nodes is not a permanent operation - also this is not visible at dataFEED OPC Suite Runtime - Those nodes (when Alarms are activated) are always retrieved from an online connection being update at browse time. Connecting an OPC Client to the dataFEED OPC Suite application will always see the entire set of Alarm nodes.



© Softing Industrial Automation GmbH



Reset to Default Address Space

The address space of PLC connection can be reset to the default address space by pressing the "Reset to Default Address Space" button. The current connection address space is replaced with the default one.

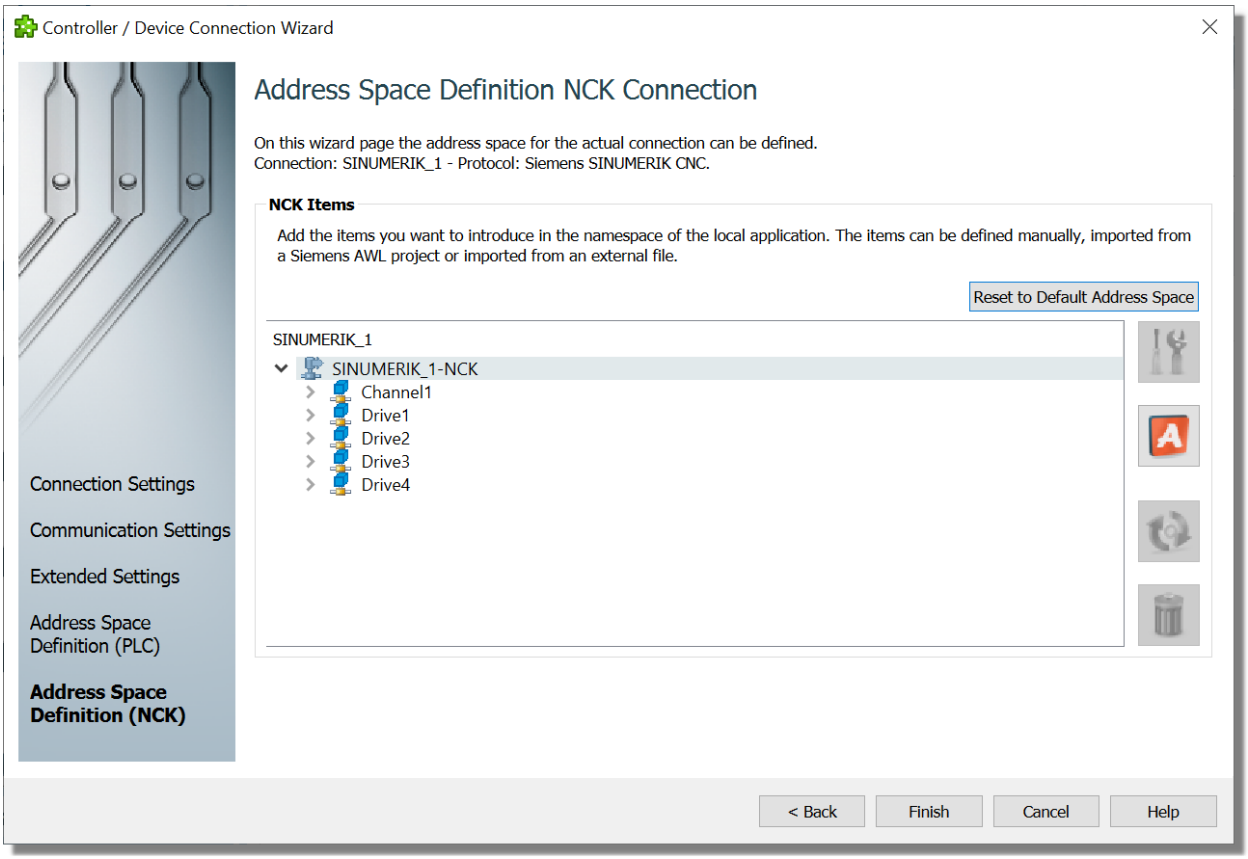
Note: The ***Reset to Default Address Space*** operation can be executed only when the current address space is not providing items used in other functionality as placeholders or triggers - in this scenario an error message is prompted out.

Immediately after the operation is executed - the address space of current PLC connection is reset to the default address space. This operation become permanent only after saving the configuration is performed. Also, closing the current Address Space Definition - PLC wizard page is loosing all the changes.

<<< Siemens SINUMERIK CNC Device Wizard - Extended Settings Siemens SINUMERIK CNC Device Wizard - Address Space Definition (NCK) >>>

On this wizard page, the address space of the NCK device component is visible and can be manipulated. The address space can be updated by resetting it to the default address space, by importing the project address space (*.awl format) or by removing some item from the current address space (option available only for Alarms and Tooling nodes - when those functionality are enabled).

Note: Address Space manipulation by adding or modifying existing nodes is not allowed. Also, removing of standard nodes is not allowed - this operation being allowed exclusively for Alarms and Tooling nodes.



NCK Connection Items

An editable tree view where the user can change the address space structure or content by resetting to default address space, importing NCK project files or deleting items (only on Alarms and Tooling nodes being available).

There are three categories of items that may be used in the tree:

- **root node** (having the name of the current connection). This node is read only, the user has just the possibility to add or delete children of this.
- **nodes:** these elements are associated with branches in tree. They can have children but they are typeless and do not carry any syntax.

Note: The items of type Node represent intermediate nodes in the tree. They are used for hierarchy purposes.

- **tags**: these elements are associated with leaves in tree.

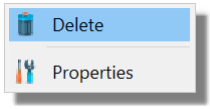
© Softing Industrial Automation GmbH

Operation

Depending on the current selection in the *NCK Connection Items* tree, the following operations are possible using the buttons on the right, the context menu or the corresponding shortcuts:

- Display Properties of the selected item (Ctrl+P) 
- Import AWL symbol file 
- Refresh data items (Ctrl+R) 
- Delete data items (Del) 

Display Properties and Delete selected item operations are available also on context menu activated on right click on selected item/node. An example of the context menu available on *NCK Connection Items* tree is shown below:

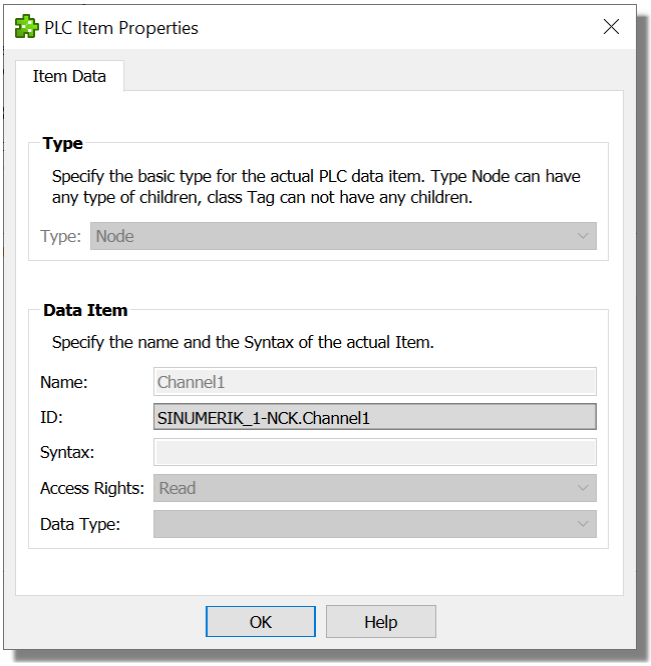


Display Properties of Selected Items/Nodes from NCK Address Space

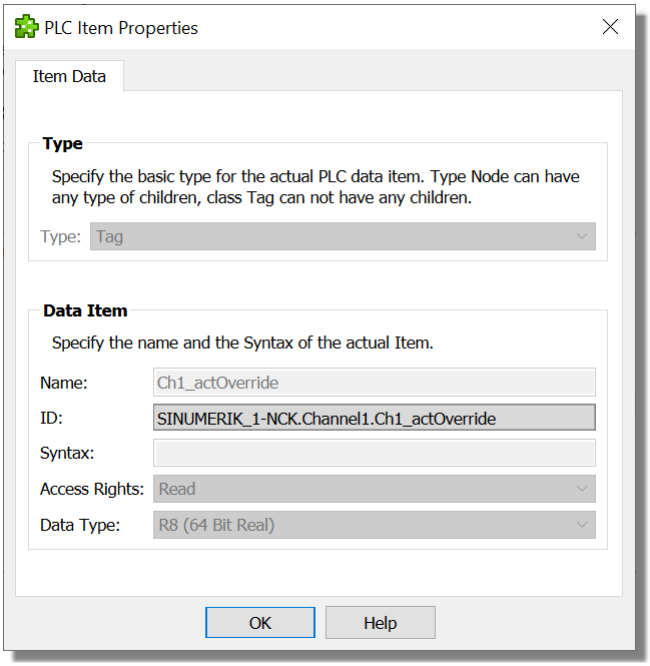
The display property operation is triggered by selecting the node to be modified and then pressing the *Properties* button  , using the shortcut or selecting the *Properties* action from the context menu.

Base on item type, the properties of a variable from address space could be different:

- Node Properties:



- Item Properties:



Note: The properties of a selected node/item from NCK address space are read only (could not be modified).

The available configuration options in this dialog are described in the table below:

Attribute	Description
Type	The type of the data item.

© Softing Industrial Automation GmbH

	The possible types are: *Node* and *Tag*.
Name	The OPC item name identifying the corresponding data item inside the dataFEED OPC Suite address space.
Id	The full OPC item path for the data item inside the dataFEED OPC Suite address space.
Syntax	The NCK nodes have no syntax (this field being reserved for PLC connections).
Access rights	The access rights for the data item. The possible values are: *Read*, *Write*, *Read and Write*.
Data type	The data type of the data item. The data type for the item is extracted out of a valid syntax provided by the user.

Note: Items of type *Node* may have child items of any existing type.

An item from the NCK connection address space may be of any data type covered by the OPC Specifications. The following table describes the available data types mapped to the corresponding VARIANT types from the OPC Specification.

Data type	VARIANT type	Description
I1	*VT_I1*	8 bit signed integer.
UI1	*VT_UI1*	8 bit unsigned integer.
I2	*VT_I2*	16 bit signed integer.
UI2	*VT_UI2*	16 bit unsigned integer.
I4	*VT_I4*	32 bit signed integer.
UI4	*VT_UI4*	32 bit unsigned integer.
I8	*VT_I8*	64 bit signed integer.
UI8	*VT_UI8*	64 bit unsigned integer.
R4	*VT_R4*	32 bit floating point.
R8	*VT_R8*	64 bit floating point.
BOOL	*VT_BOOL*	Boolean.
STRING	*VT_BSTR*	String.

DATETIME	*VT_DATE*	Date and time.
DECIMAL	*VT_CY*	Decimal.
ANY	*VT_VARIANT*	Any data type.

Note: Arrays of fixed and variable length are also supported for these data types.

Note: The alarms and tooling nodes are not stored in any configuration files - being exclusively receive online from the NCK device component connection. Offline configuration of Alarm and Tooling Nodes is not possible.

Import AWL Symbol File

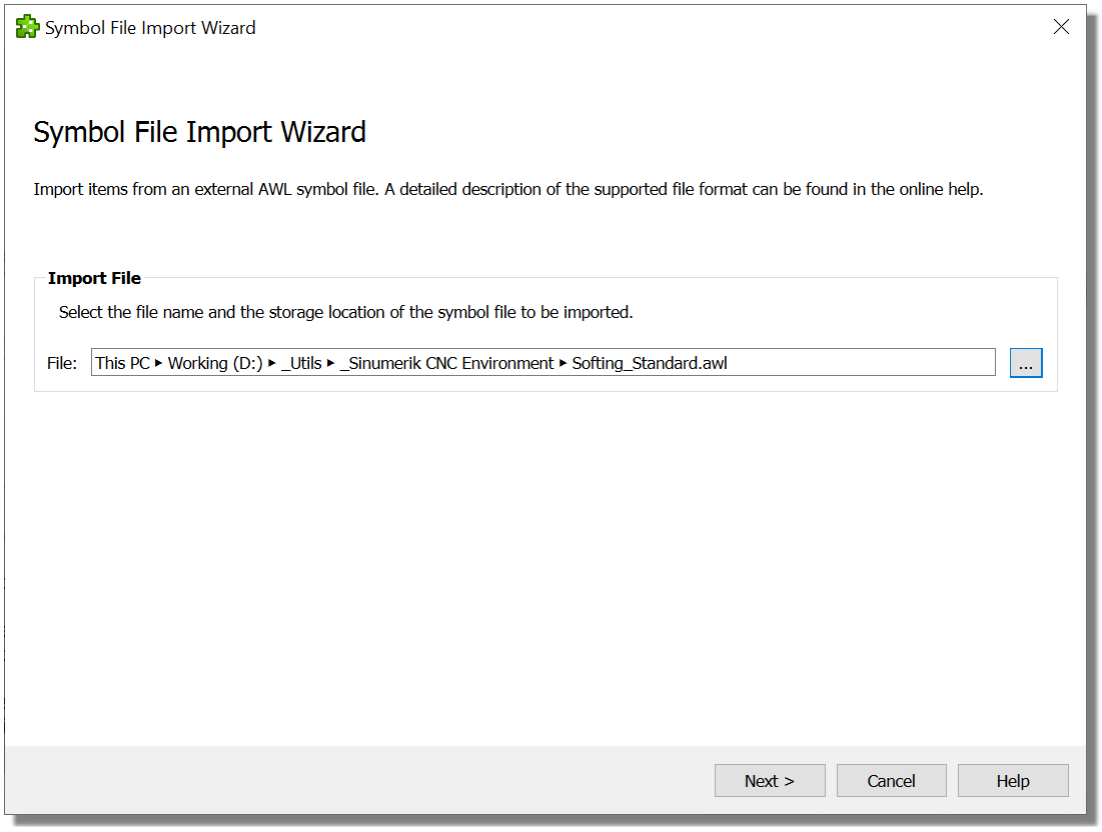
The Softing dataFEED OPC Suite Configurator allows the import of AWL NCK connection items from an external AWL symbol file.

The import action can be triggered by using the *Import AWL symbol file* button  .

Attention: The import from AWL symbol file action applies on the entire address space of the current Siemens SINUMERIK CNC - NCK connection. Any existing items will be automatically removed from the address space prior to performing the import (exception being the items which are used on other functionality as placeholder or trigger).

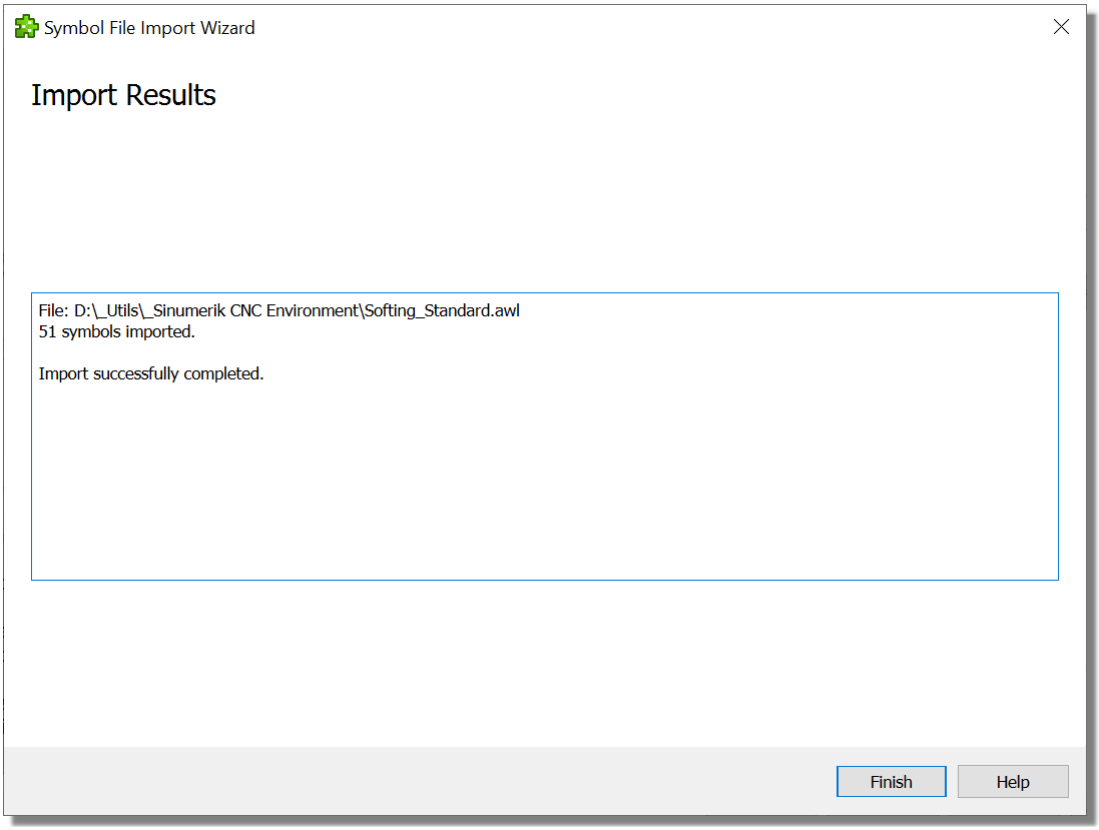
When this import action is triggered, the *Symbol File Import Wizard* is started.

© Softing Industrial Automation GmbH



On this step, it should be provided an *.awl file. After file selection, the Next button become available. Pressing Next button, the file is *.awl project file loaded, validated and prepared for address space extraction.

Immediately after, the address space is updated with the imported nodes - a report on imported nodes being provided.



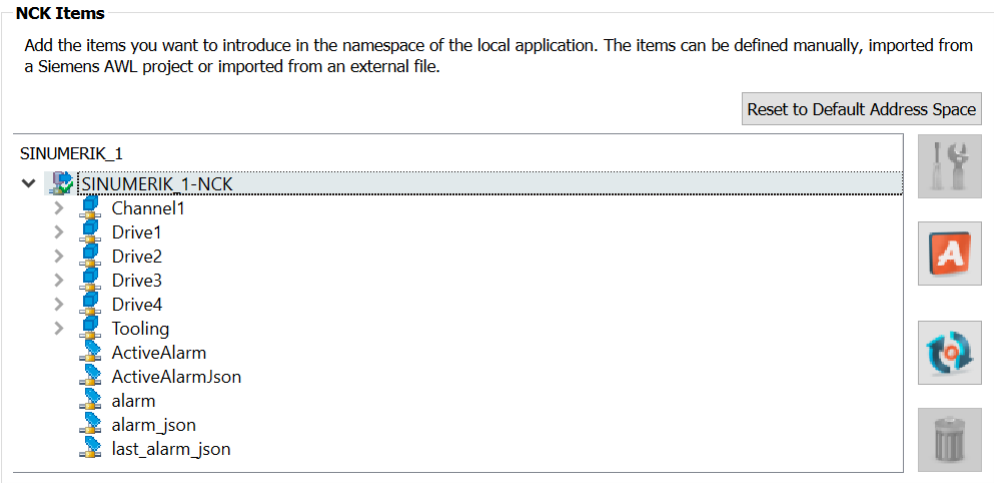
The imported address space is immediately available on the exposed NCK device component address space.

Attention: The Tooling and Alarm nodes (when activated) are not affected by importing the *.awl project file. Only the standard nodes are overwritten. The alarms and tooling nodes being exclusively obtain online from the NCK connection.

Refresh Address Space (NCK Connection)

The alarms and tooling nodes (when Alarms and/or Tooling are enabled on NCK device component) are dynamically obtain from an online connection. Those nodes can be manipulated (deletion of nodes and items only), also it could be the situation that the alarms or tooling nodes available on NCK connection were changed - the alarm and tooling nodes could be automatically updated (from current online NCK connection) by pressing the Refresh Address Space button.

© Softing Industrial Automation GmbH



Attention: The refresh address space functionality is applicable only for nodes and items which are retrieve online - on NCK device component being applicable only on alarm and tooling nodes. When Alarms and/or Tooling are disabled, the Refresh Address Space option is disabled too.

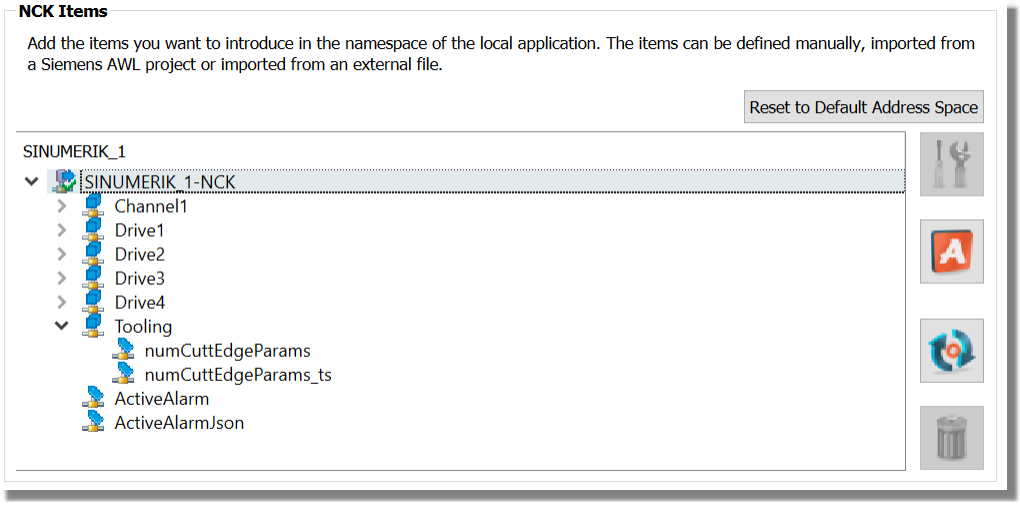
Delete NCK Data Items

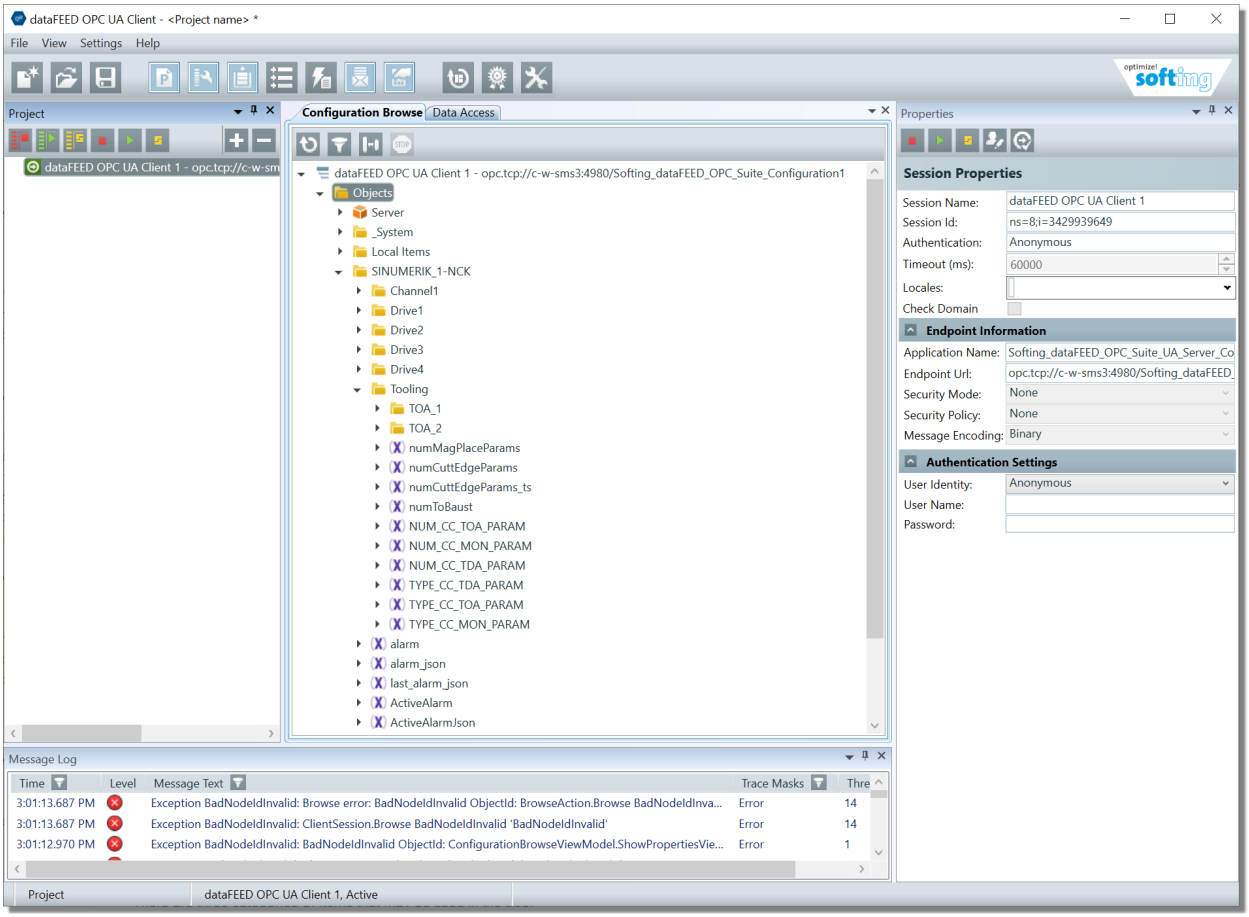
Triggering the *Delete NCK data items*  action with the corresponding button on the right wizard side or *Delete* from the context menu will remove the currently selected node from the *Data Items* address space tree.

Attention: The root node of the NCK connection address space tree can not be deleted.

NOTE: A node or item which is currently used in a data processing or data destination action can not be deleted.

Attention: Deletion of alarms or tooling nodes is not a permanent operation - also this is not visible at dataFEED OPC Suite Runtime - Those nodes (when Alarms and/or Tooling are activated) are always retrieved from an online connection being update at browse time. Connecting an OPC Client to the dataFEED OPC Suite application will always see the entire set of Alarm and/or Tooling nodes.





Reset to Default Address Space

The address space of NCK connection can be reset to the default address space by pressing the "Reset to Default Address Space" button. The current connection address space is replaced with the default one.

© Softing Industrial Automation GmbH

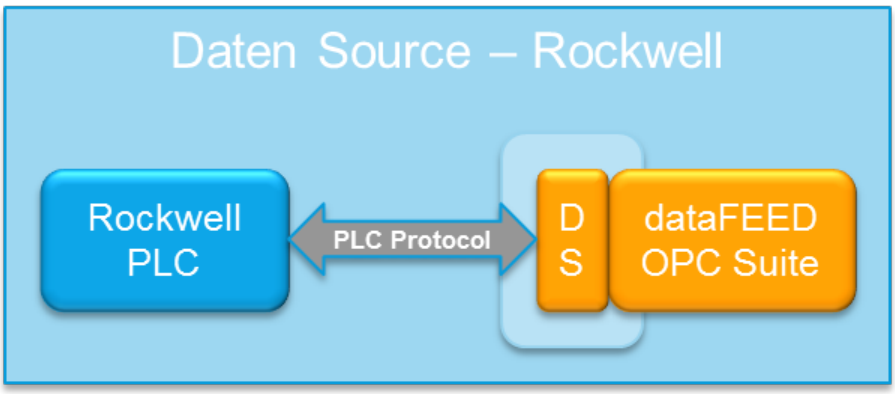
Note: The ***Reset to Default Address Space*** operation can be executed only when the current address space is not providing items used in other functionality as placeholders or triggers - in this scenario an error message is prompted out.

Immediately after the operation is executed - the address space of current NCK connection is reset to the default address space. This operation become permanent only after saving the configuration is performed. Also, closing the current Address Space Definition - NCK wizard page is loosing all the changes.

<<< Siemens SINUMERIK CNC Device Wizard - Address Space Definition (PLC)

7.3.1.3 Rockwell Devices

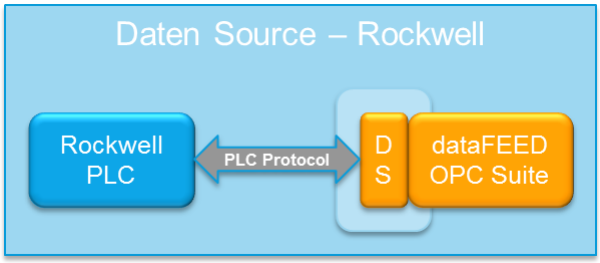
This section of the Softing dataFEED OPC Suite Online Help describes how to configure Rockwell devices within Softing dataFEED OPC Suite.



Controllers and devices data source makes it possible to exchange data between the field devices (PLCs) of a wide variety of manufacturers and OPC clients via Ethernet TCP/IP.

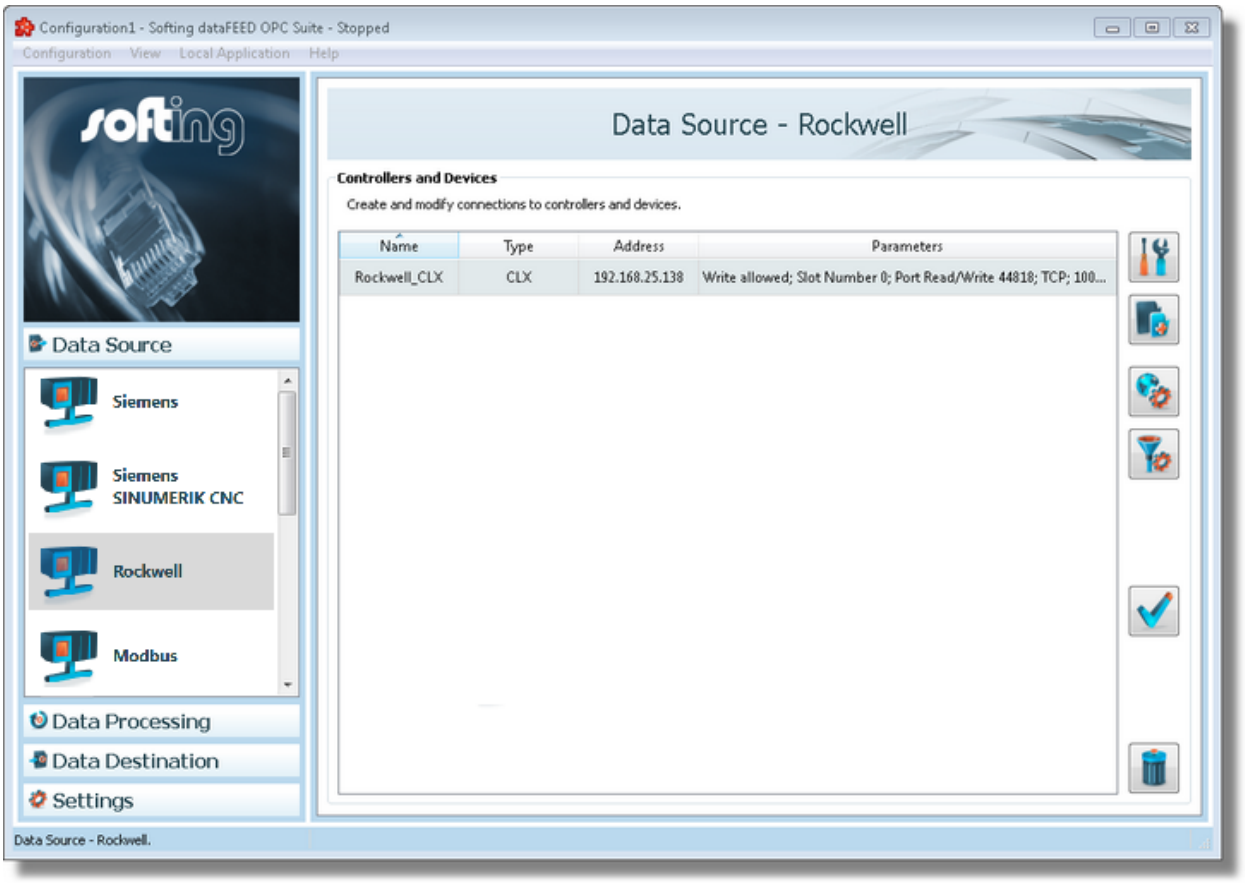
A detailed description of how to configure access to PLC devices functionality can be found in the Configuration section.

7.3.1.3.1 Configuration



To start configuring the connection to an external Rockwell device select Rockwell Devices from the Data Source group in the navigator on the left. The Data Source - Rockwell Device main

user interface is displayed. The main element of this page is the External Devices list. Each entry shown in the list of the configuration section represents a connection to an external device.



Note: The list is only shown and editable if a configuration has been loaded.

Each Device connection has a unique name within the configuration, which is given in the *Name* column.

The name of the device will be shown in the namespace tree in the Softing dataFEED OPC Suite Configurator and in the local dataFEED OPC Suite application, and its entire namespace will be under this name.

Note: Device name must be unique string within a dataFEED OPC Suite configuration. Also the name could not contain address space delimiters, multi-byte characters, empty spaces.

The device type (second column) can be one of: CLX or SLC/PLC5

Third column: address represents the actual device address which can be IPv4, hostname or IPv6.

Parameters column contains various attributes of the connection like: plc header type, write allowed, ports, communication protocol.

© Softing Industrial Automation GmbH

Operation

From the Siemens Devices main user interface certain operations can be performed over the selected connection by using the buttons on the right, by accessing the context sensitive menu or by pressing the keyboard shortcuts. The following actions are available:

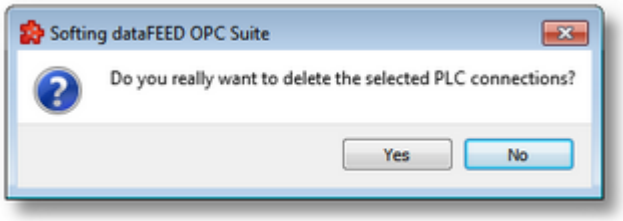
- Change the properties of the selected data source by using the *Properties* button  , pressing the keyboard shortcut CTRL+P or selecting *Properties* from the context menu.
- Add a new connection by using the *Add* button  , pressing the keyboard shortcut CTRL+A or selecting *Add* from the context menu.
- Change global settings for the selected data source by using the Global settings button  .
- Delete the selected data source by using the *Delete* button  , pressing the DEL / DELETE key on the keyboard or selecting *Delete* from the context menu.
- Address space definition button  , by pressing it the Siemens Device Wizard will be opened in it's last page. For more details see Address Space Definition page.
- Use the test connection button  to check the availability of the specified PLC.

Note: If no controller / device connection is selected, the *Add a new data source* button is the only one available.

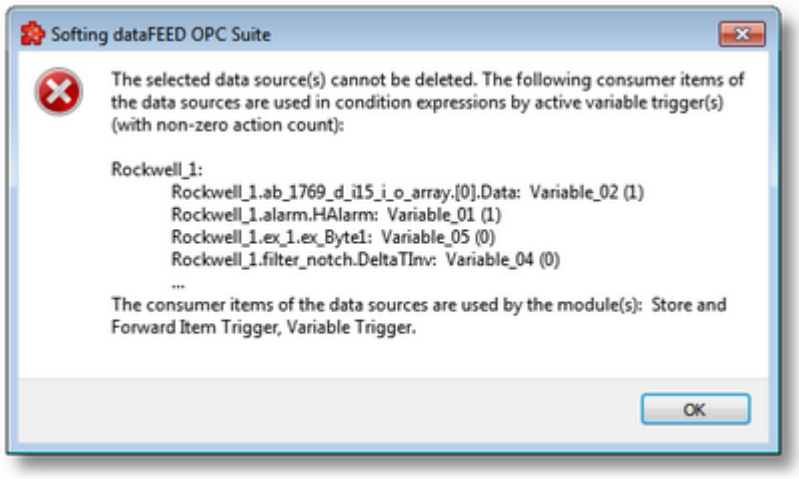
The *Delete* button  will remove the selected device connections from the list.

The user can select more than one server connection with the help of the CTRL or SHIFT keys.

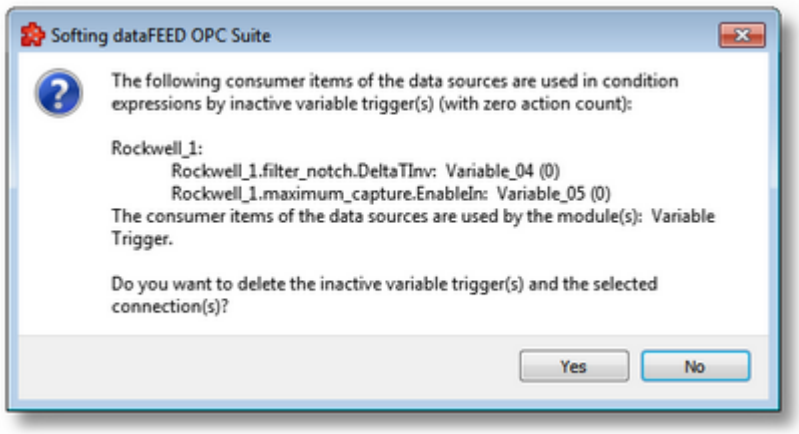
A dialog will then prompt the user to confirm the deletion.



A dialog will inform the user that the data sources cannot be deleted, if consumer items and/or system items of the selected data sources are used by active variable triggers or other modules.



A dialog will prompt the user to confirm deletion of the data sources and inactive variable triggers, if consumer items and/or system items of the selected data sources are used only by inactive variable triggers.

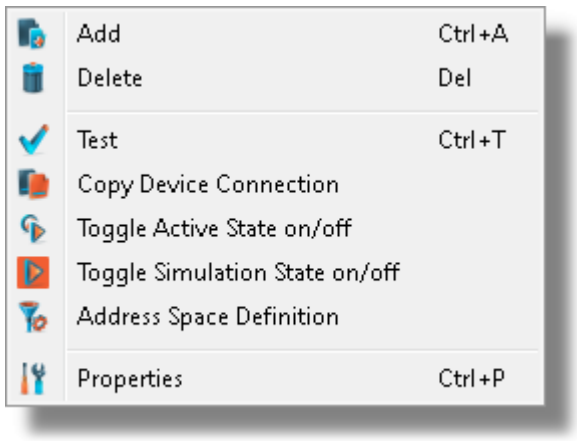


Attention: A device connection cannot be deleted if at least one data item defined in its namespace is used in a Processing or Data Destination action.

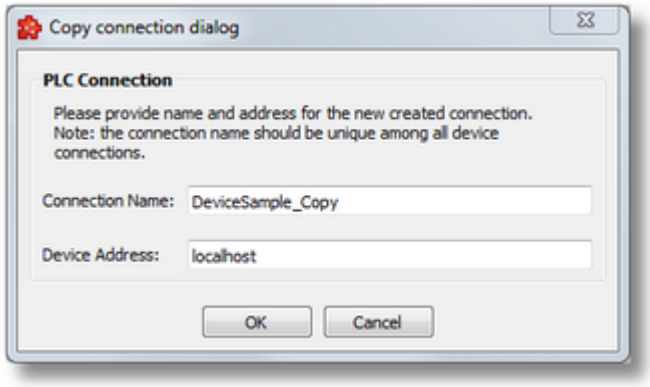
All these actions are also provided in the context menu of the External device list. Which actions are available depends on the current selection in the list.

An example of the context menu can be seen below:

© Softing Industrial Automation GmbH



- Copy device connection  . This action is available only in the context menu and it leads to a dialog where the user may provide a name and device address. See the screen shot below:



The proposed name is the original connection's name post fixed with "_Copy" and the default address will be "localhost".

- Toggle Activate state  . This action will enable / disable a devices connection. A disabled connection will be shown as grayed out in the device connection list and it will be ignored by runtime.

- Toggle Simulation state  . This action will mark a connection as simulation (or remove the simulation mode). Simulation connections are shown in red in main device list.

Note: The connection name should be unique among all device connections names.

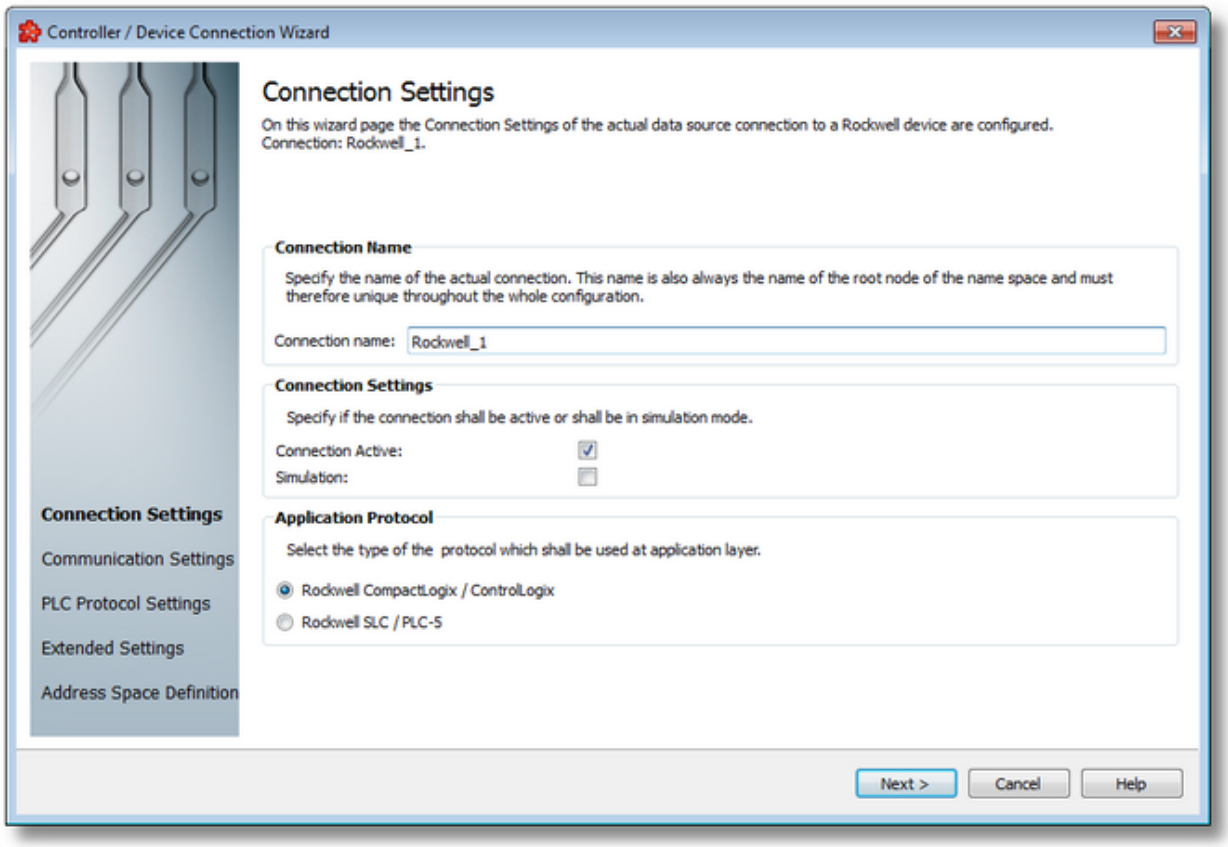
Note: The device address should a valid host name or an IP address in format IPv4 or IPv6.

7.3.1.3.2 Rockwell Device Wizard

With the Rockwell Device Wizard you can create and modify a data source connection to an external PLC device:

- Specify connection settings @ Rockwell Device Wizard - Connection Settings
- Specify communication parameters @ Rockwell Device Wizard - Communication Settings
- Protocol settings @ Rockwell Device Wizard - Plc Protocol Settings
- Extended settings @ Rockwell Device Wizard - Extended Settings
- Address space definition @ Rockwell Device Wizard - Address Space Definition

On this page the connection name that will uniquely identify the data source and some generic connection settings can be provided.



Connection Name

© Softing Industrial Automation GmbH

An unique identifier for the current connection. In case the provided name is already used by other connection, the user use is not allowed to go into the next wizard page.

Connection active

This flag specifies if the current connection will be enabled or not.

Simulation

This flag specifies if the current connection will be simulated by the server or it will represent a real device connection.

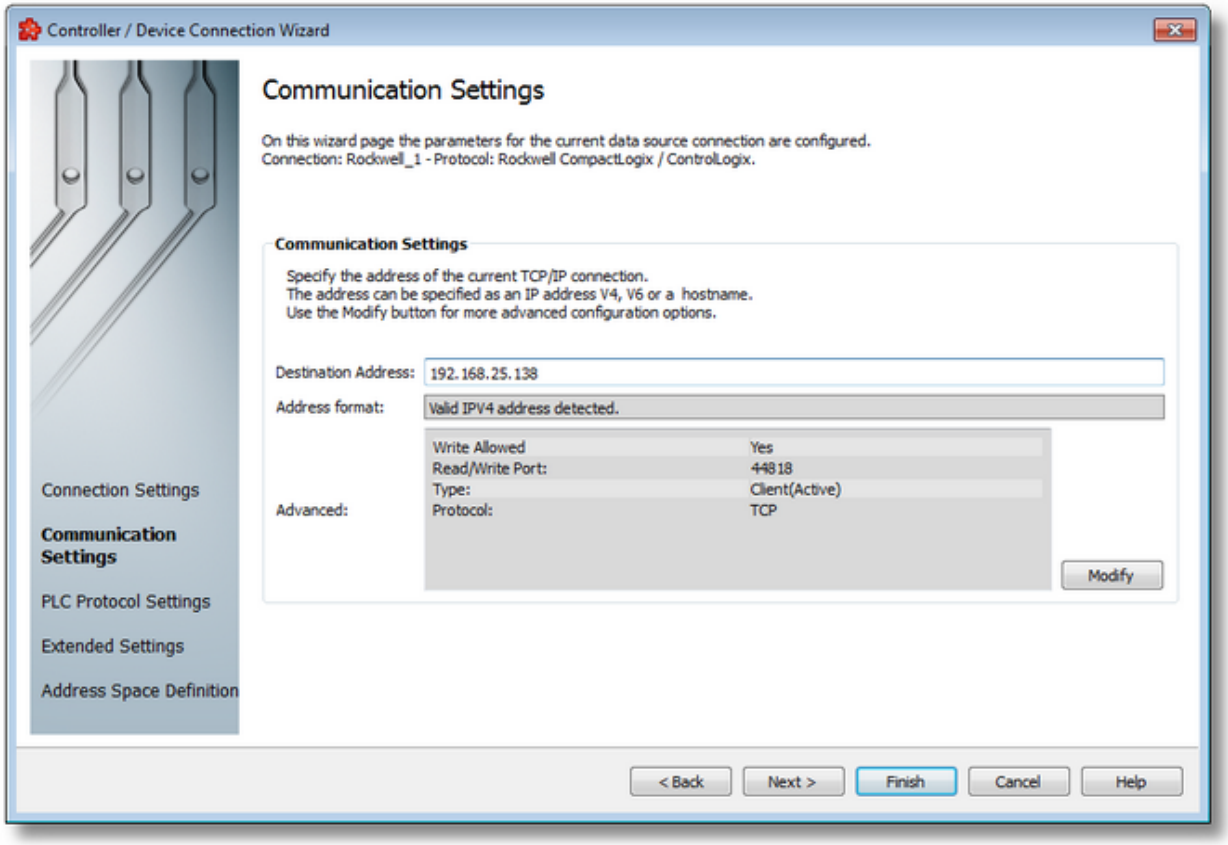
Application protocol

Currently supported protocols are:

- CompactLogix / ControlLogix
- SLC / PLC-5

Rockwell Device Wizard - Communication Settings >>>

On this wizard page various parameters for the current TCP/IP connection are configured.



Destination Address: IP address or name

This is where the IP address of the target station (IP address of the PLC) or the DNS name of the target station is entered. If the computer was configured for DNS and if a DNS server is available on the network, the symbolic name of the target station can also be entered.

For how an IP address is set up, see IP address.

Address Format

Read-Only field; automatically detects the format of destination address.

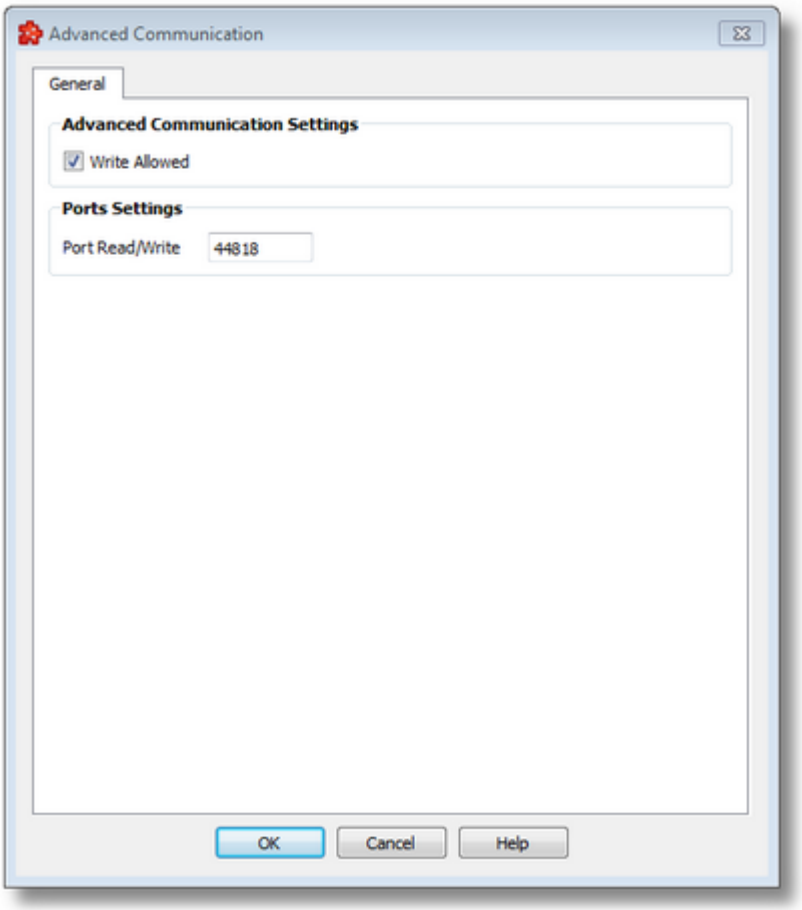
Advanced:

Read-Only field; this is a summary for TCP/IP Communication advanced settings. These settings can be changed using Modify button that lead to Advanced Communication Settings

<<< Rockwell Device Wizard - Connection Settings Rockwell Device Wizard - Plc Protocol Settings >>>

© Softing Industrial Automation GmbH

In this dialog specific Rockwell communication parameters can be adjusted.



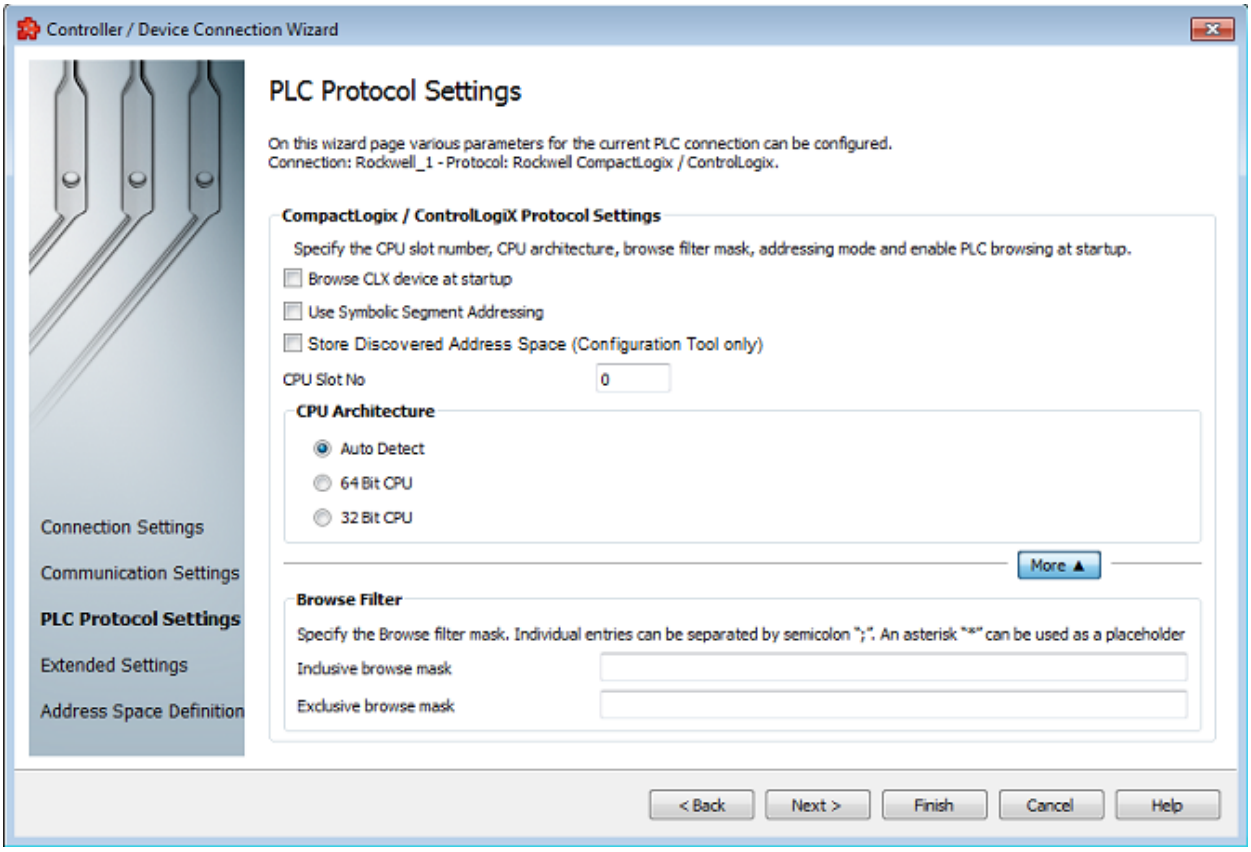
Write to the PLC permitted (Write Allowed)

Here you specify whether write access to the PLC is to be possible via this connection and a write connection is to be established to the hardware. You can always disable write accessing via a connection if you only want to use read accesses or if you want to be sure that no values on the PLC are to be changed. When write accessing the PLC via a Write (Write activeConnection) or a Send/Receive (Send Direct Connection for Send/Receive), certain points of the PLC are written with values (each write access is preceded by a read access).

Port for read accesses, port for write accesses

Port numbers are addresses which are used within the transport layer to address applications. Port numbers are required for TCP connections. The port is parallel to the TSAPs for RFC1006 and H1 connections. The port number is a 16-bit number from 1 to 65535. Keep the following in mind when you set up the connection.

In this page PLC parameters for RockWell CompactLogix / ControlLogix devices can be specified.



Browse CLX device at Startup

This option is enabling the complete browse operation on Rockwell CLX at start-up time - otherwise this operation is performed on first browse required by the connected client on specified connection.

Attention: Browse operations on Rockwell CLX devices could be a time consuming operation.

Use Symbolic Segment Addressing

Using Symbolic Segment (Tag Name) Addressing as addressing mode to access CLX Tags.

Stored Discovered Address Space (Configuration Tool only)

Storing the discovered address space is serializing the current address space of selected configuration (at saving time). Reloading configuration, the connection address space is loading from alias file (going online and browsing the address space being not necessary).

© Softing Industrial Automation GmbH

Note: The stored address space is not applicable at Application Runtime - being used by the Configuration tool only.

Connect and Refresh operation is updating the address space with the current information from the device.

CPU slot

The slot of the ControlLogiX CPU is specified here.

CPU Architecture

Here you can choose, whether you are connected to a 32 Bit CPU (default setting) or a 64 Bit CPU.

Browse filter masks

Some PLC programs contain many variables. These are created by PLC program code generators and are used internally. Not all tags available on the PLC are used by SCADA systems. Usually only a few of these variables are involved.

The start of the OPC server and the access of OPC clients to indicate the browsing list can take a very long time if there are many variables. This is particularly true when the Ethernet module of the PLC is not integrated in the CPU.

Setting the browser filter:

The CLX browser filters are entered in the input fields Inclusive browser mask and Exclusive browser mask. A combination is possible. The inclusive screen is evaluated first followed by the exclusive.

Inclusive browser mask:

Only the specified values are shown and available for OPC browsing.

Exclusive browser mask:

The specified variables are not shown.

If a mask is blank, it will be ignored.

A mask is a text which consists of one or more entries.

An entry without a dot “.” refers to the variables in “controller tags.”

Example: Variable1.

An asterisk “*” can be used at the end of the entry as the placeholder for any character string.

Example: Var*

This determines all variables which begin with the text “Var.” These would also be: Variable1, variable2, VAR3, variance.

If “program tags” are to be addressed, an entry contains a dot (“.”). The name of the subprogram is specified in front of the dot. The asterisk “*” can be used as a placeholder at the end. The variable name is specified after the dot. The asterisk “*” can also be used behind that.

Example: Program:MainProgram.Var*

Example: Program:A*.*

A program name without a dot is ignored.

Individual entries are separated by a semicolon (“;”).

Upper case and lower case letters are ignored.

Blanks are not allowed.

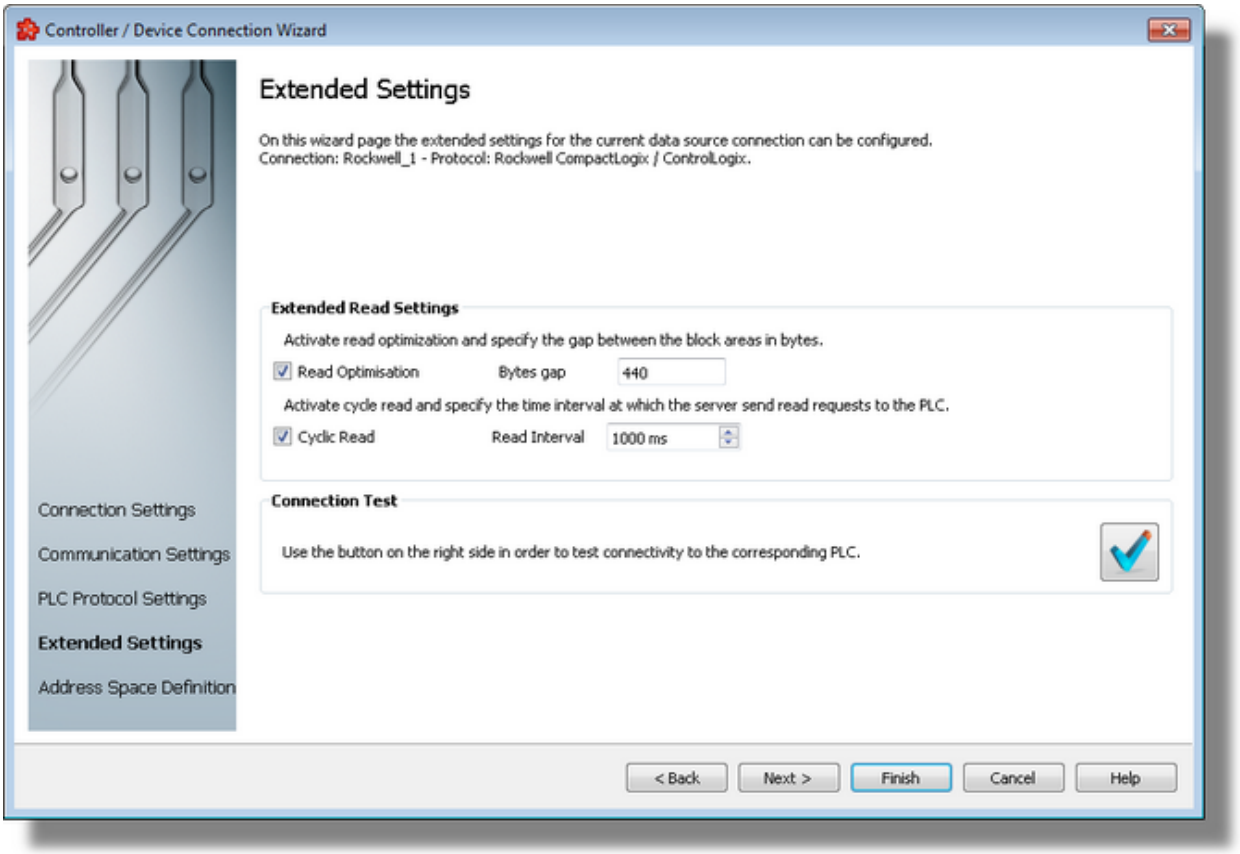
Variable names may not contain a dot (“.”), a semicolon (“;”) or an asterisk (“*”). If such characters are included anyway, a search cannot be made for them.

The maximum number of characters for a filter is 300 characters.

<<< Rockwell Device Wizard - Communication Settings Rockwell Device Wizard - Communication Settings >>>

On this wizard page the extended settings for the current data source connection can be configured.

© Softing Industrial Automation GmbH



Read optimization

Reading from the controller can be optimized here. If several variables are requested which are located in the same operand area (e.g., in a data block or in the flag area), the complete area is read from the first variable to the last variable without read optimization. Gaps between the individual block areas are also read no matter how large these gaps are. The maximum size of the gaps (in bytes) between the requested individual items within a data block without a new block being created is entered. If the gap equals the specified value, a new block is created.

Enable cyclic read access and read interval

Cyclic read access is activated as the standard. The server reads the values of the variables from the hardware via a fetch active connection at a certain time interval. It can be specified in Read interval [in ms] the time intervals at which the server is to send a read request to the PLC. The setting “0” means that it should be read as soon as possible. If "Cyclic read" is selected, read access is performed once when the client requests and no longer cyclically. However, this is only necessary in special cases and requires a precise knowledge of the communication.

Continue to Address Space definition

Optionally by checking this flag, the user can define the alias file for the current connection now.

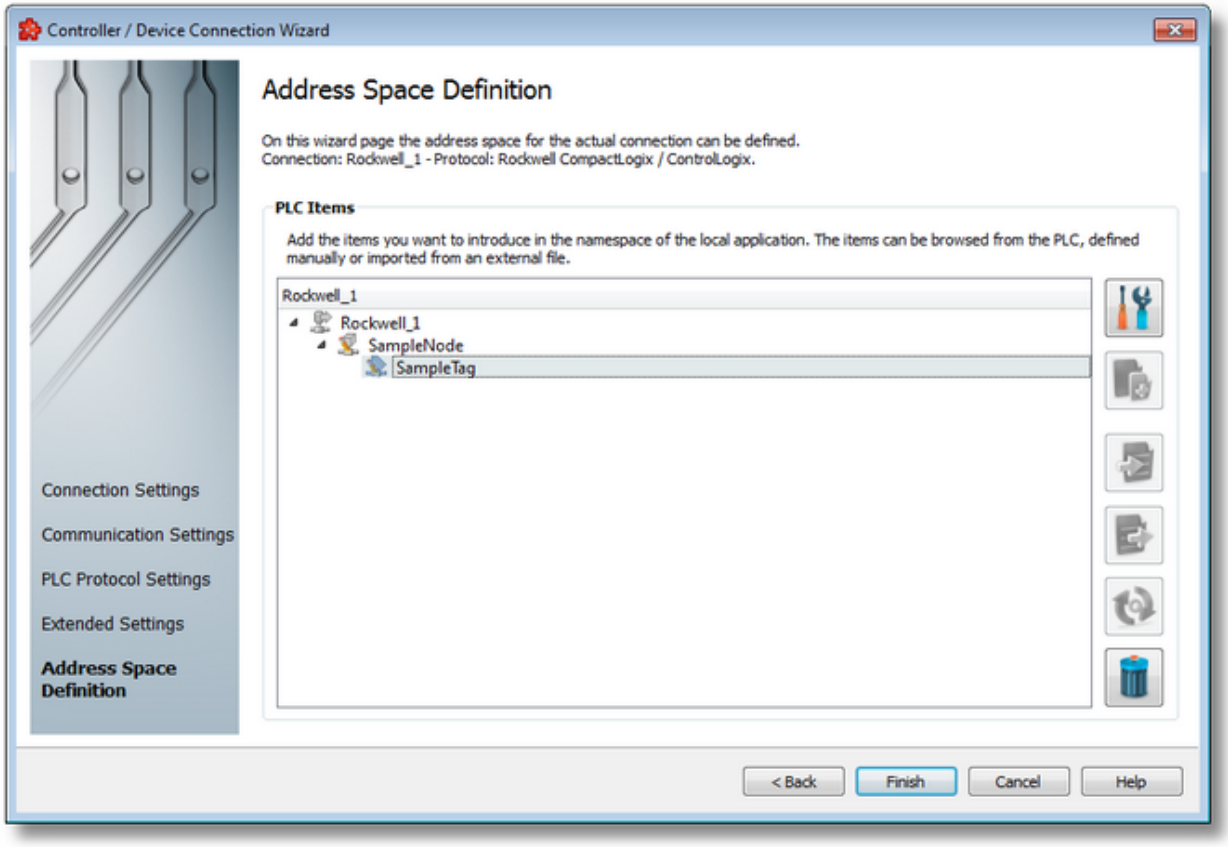
Although this step may be postponed to a later time.

Connection test.

Use the connection test button  to check the availability of the specified PLC.

<<< Rockwell Device Wizard - Plc Protocol Settings Rockwell Device Wizard - Address Space Definition >>>

On this wizard page the data items for the actual PLC connection can be defined. The address space can be defined by manually adding items, by importing the address space from an external file or by importing a device symbol file (for connections of tyoe S7 and Modbus).



PLC Connection Items

An editable tree view where the user can change the address space structure or content by adding, deleting or modifying items.

There are 3 categories of items that may be used in the tree:

© Softing Industrial Automation GmbH

- **root node** (having the name of the current connection). This node is read only, the user has just the possibility to add or delete child's of this.
- **nodes:** these elements are associated with branches in tree. They can have children but they are typeless and do not carry any syntax.

Note: The items of type Node represent intermediate nodes in the tree, they are used for hierarchy purposes.

- **tags:** these elements are associated with leaves in tree.

Note: A tag cannot have any children, so Add is not available for tags.

Operation

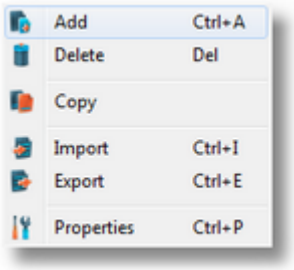
Depending on the current selection in the *Plc Connection Items* tree, the following operations are possible using the buttons on the right, the context sensitive menu or the corresponding shortcuts:

- Properties of the selected item (Ctrl+P) 
- Add a new item (Ctrl+A) 
- Import from file (Ctrl+I) 
- Export to file (Ctrl+E) 
- Delete data items (Del) 
- Refresh data items (Ctrl+R) 

Two more operations are supported, which are only accessible from the contextual menu:

- Copy data item(s) 
- Paste to data item(s) 

An example of the contextual menu available on *Plc Connection Items* tree is shown below:



Delete data items

Triggering the *Delete data items*  action with the corresponding button on the right wizard side or *Delete* from the context menu will remove the currently selected node from the *Data Items* namespace tree.

Attention: The root node of the namespace tree can not be deleted.

Attention: A node which is currently used in a data processing or data destination action can not be deleted.

Copy

The *Copy*  action is provided in the context menu of the *Plc Connection Items* tree. Triggering this action will copy the selected node to the local clipboard. All the properties of the node are copied. Also, if the node contains children, the entire branch is copied.

Note: The root node of the namespace tree does not provide a *Copy* action.

Paste

The *Paste*  action is provided in the context menu of the *Plc Connection Items* tree. This action will add the node currently stored in the local clipboard as a a child of the currently selected node. Whether the action is available in the context menu depends on the type of the currently selected node and the type of the node in the clipboard.

Note: The *Paste* action is only available if a node was previously copied to the clipboard.

If the *Paste* action is performed on a node which already has a child of the same name as the node in the clipboard, the pasted node is renamed automatically by adding a numerical suffix to the original name.

```
E.g. NodeA, NodeA01, NodeA02, ...
```


© Softing Industrial Automation GmbH

Refresh data items

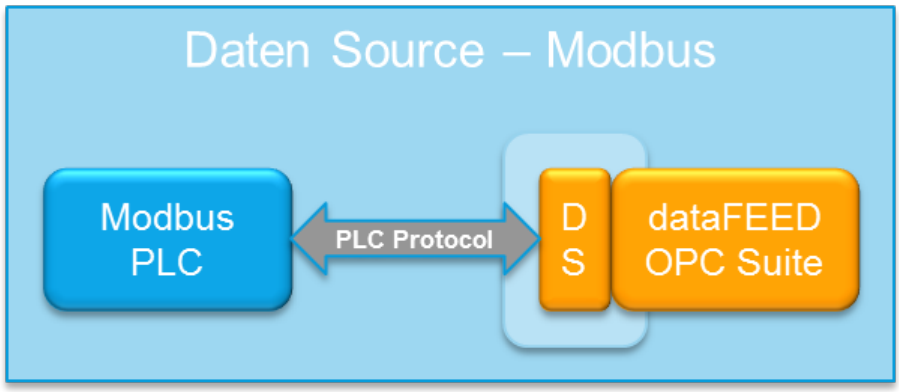
Triggering the Refresh *data items*  action with the corresponding button on the right wizard will perform a device query to retrieve the actual items.

Note: This operation may be time very consuming, depending on device configuration and connection speed.

<<< Rockwell Device Wizard - Extended Settings

7.3.1.4 Modbus Devices

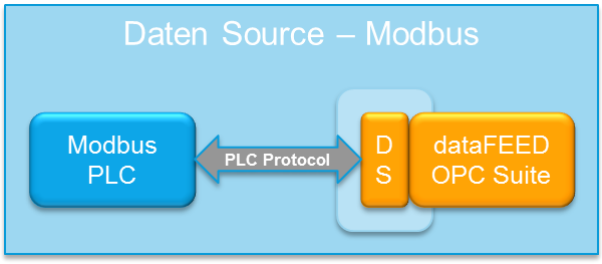
This section of the Softing dataFEED OPC Suite Online Help describes how to configure Modbus devices within Softing dataFEED OPC Suite.



Controllers and devices data source makes it possible to exchange data between the field devices (PLCs) of a wide variety of manufacturers and OPC clients via Ethernet TCP/IP.

A detailed description of how to configure access to PLC devices functionality can be found in the Configuration section.

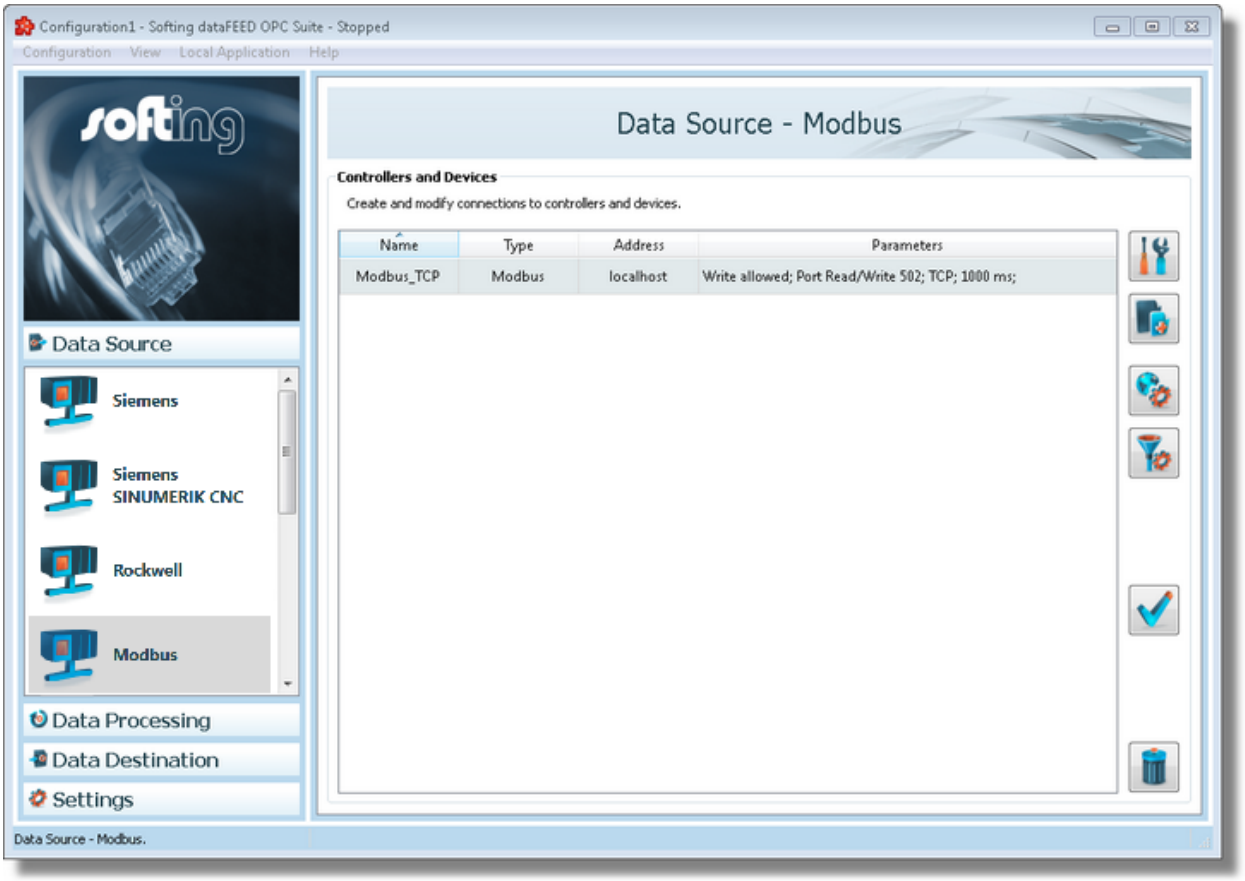
7.3.1.4.1 Configuration



To start configuring the connection to an external Modbus device select Modbus Devices from the Data Source group in the navigator on the left.

The Data Source - Modbus Device main user interface is displayed. The main element of this page is the External Devices list.

Each entry shown in the list of the configuration section represents a connection to an external device.



Note: The list is only shown and editable if a configuration has been loaded.

Each Device connection has a unique name within the configuration, which is given in the *Name* column.

The name of the device will be shown in the namespace tree in the Softing dataFEED OPC Suite Configurator and in the local dataFEED OPC Suite application, and its entire namespace will be under this name.

Note: Device name must be unique string within a dataFEED OPC Suiteconfiguration. Also the name could not contain address space delimiters, multi-byte characters, empty spaces.

The device type (second column) can be Modbus.

Third column: address represents the actual device address which can be IPv4, hostname or IPv6.

Parameters column contains various attributes of the connection like: plc header type, write allowed, ports, communication protocol.

© Softing Industrial Automation GmbH

Operation

From the Siemens Devices main user interface certain operations can be performed over the selected connection by using the buttons on the right, by accessing the context sensitive menu or by pressing the keyboard shortcuts. The following actions are available:

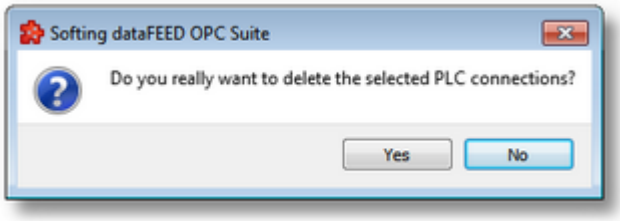
- Change the properties of the selected data source by using the *Properties* button  , pressing the keyboard shortcut CTRL+P or selecting *Properties* from the context menu.
- Add a new connection by using the *Add* button  , pressing the keyboard shortcut CTRL+A or selecting *Add* from the context menu.
- Change global settings for the selected data source by using the Global settings button  .
- Delete the selected data source by using the *Delete* button  , pressing the DEL / DELETE key on the keyboard or selecting *Delete* from the context menu.
- Address space definition button  , by pressing it the Siemens Device Wizard will be opened in it's last page. For more details see Address Space Definition page.
- Use the test connection button  to check the availability of the specified PLC.

Note: If no controller / device connection is selected, the *Add a new data source* button is the only one available.

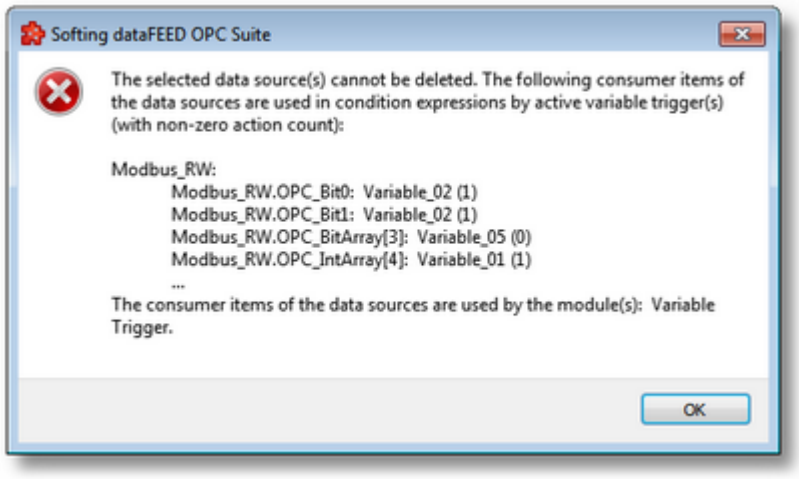
The *Delete* button  will remove the selected device connections from the list.

The user can select more than one server connection with the help of the CTRL or SHIFT keys.

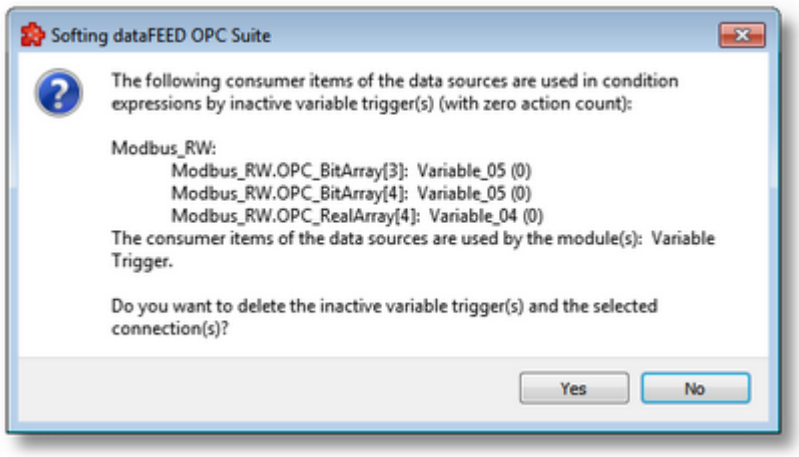
A dialog will then prompt the user to confirm the deletion.



A dialog will inform the user that the data sources cannot be deleted, if consumer items and/or system items of the selected data sources are used by active variable triggers or other modules.



A dialog will prompt the user to confirm deletion of the data sources and inactive variable triggers, if consumer items and/or system items of the selected data sources are used only by inactive variable triggers.

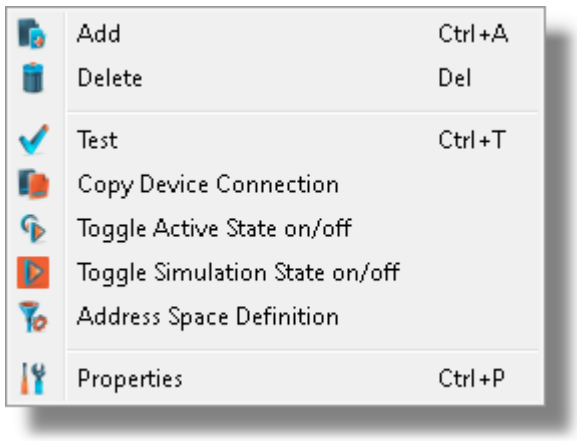


Attention: A device connection cannot be deleted if at least one data item defined in its namespace is used in a Processing or Data Destination action.

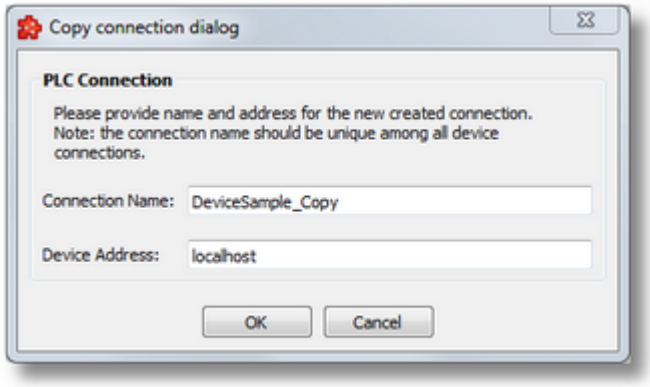
All these actions are also provided in the context menu of the External device list. Which actions are available depends on the current selection in the list.

An example of the context menu can be seen below:

© Softing Industrial Automation GmbH



- Copy device connection  . This action is available only in the context menu and it leads to a dialog where the user may provide a name and device address. See the screen shot below:



The proposed name is the original connection's name post fixed with "_Copy" and the default address will be "localhost".

- Toggle Activate state  . This action will enable / disable a devices connection. A disabled connection will be shown as grayed out in the device connection list and it will be ignored by runtime.

- Toggle Simulation state  . This action will mark a connection as simulation (or remove the simulation mode). Simulation connections are shown in red in main device list.

Note: The connection name should be unique among all device connections names.

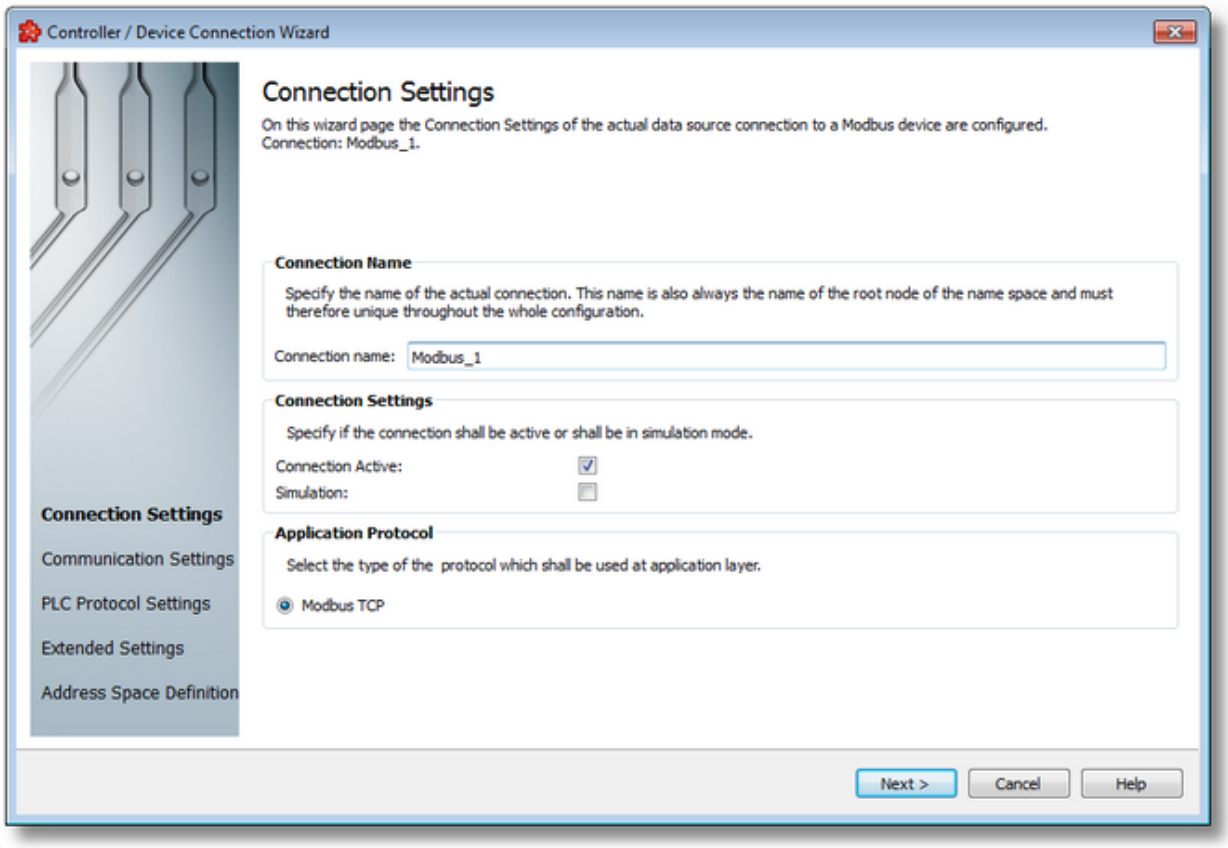
Note: The device address should have a valid host name or an IP address in format IPv4 or IPv6.

7.3.1.4.2 Modbus Device Wizard

With the Modbus Device Wizard you can create and modify a data source connection to an external PLC device:

- Specify connection settings @ Modbus Device Wizard - Connection Settings
- Specify communication parameters @ Modbus Device Wizard - Communication Settings
- Protocol settings @ Modbus Device Wizard - Plc Protocol Settings
- Extended settings @ Modbus Device Wizard - Extended Settings
- Address space definition @ Modbus Device Wizard - Address Space Definition

On this page the connection name that will uniquely identify the data source and some generic connection settings can be provided.



Connection Name

© Softing Industrial Automation GmbH

An unique identifier for the current connection. In case the provided name is already used by other connection, the user use is not allowed to go into the next wizard page.

Connection active

This flag specifies if the current connection will be enabled or not.

Simulation

This flag specifies if the current connection will be simulated by the server or it will represent a real device connection.

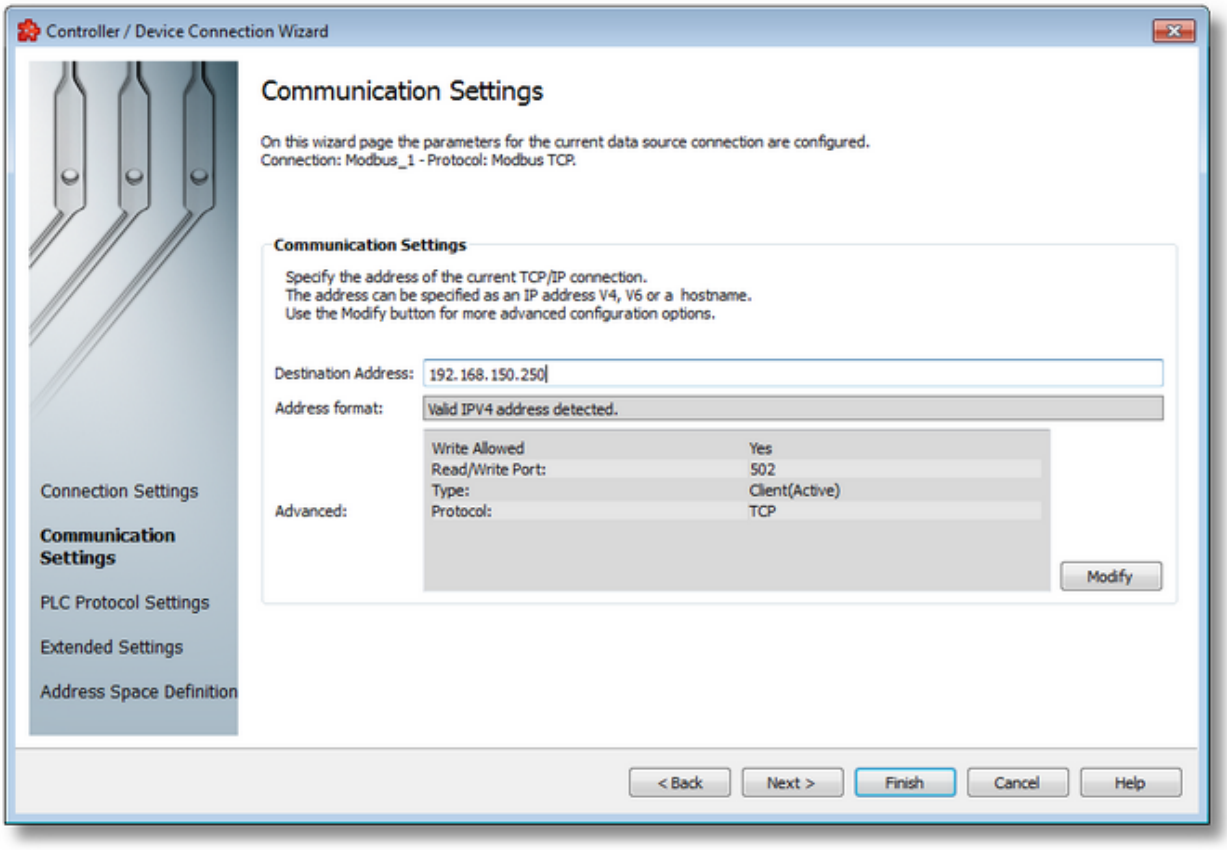
Application protocol

Currently supported protocol is:

- Modbus TCP

Modbus Device Wizard - Communication Settings >>>

On this wizard page various parameters for the current TCP/IP connection are configured.



Destination Address: IP address or name

This is where the IP address of the target station (IP address of the PLC) or the DNS name of the target station is entered. If the computer was configured for DNS and if a DNS server is available on the network, the symbolic name of the target station can also be entered.

For how an IP address is set up, see IP address.

Address Format

Read-Only field; automatically detects the format of destination address.

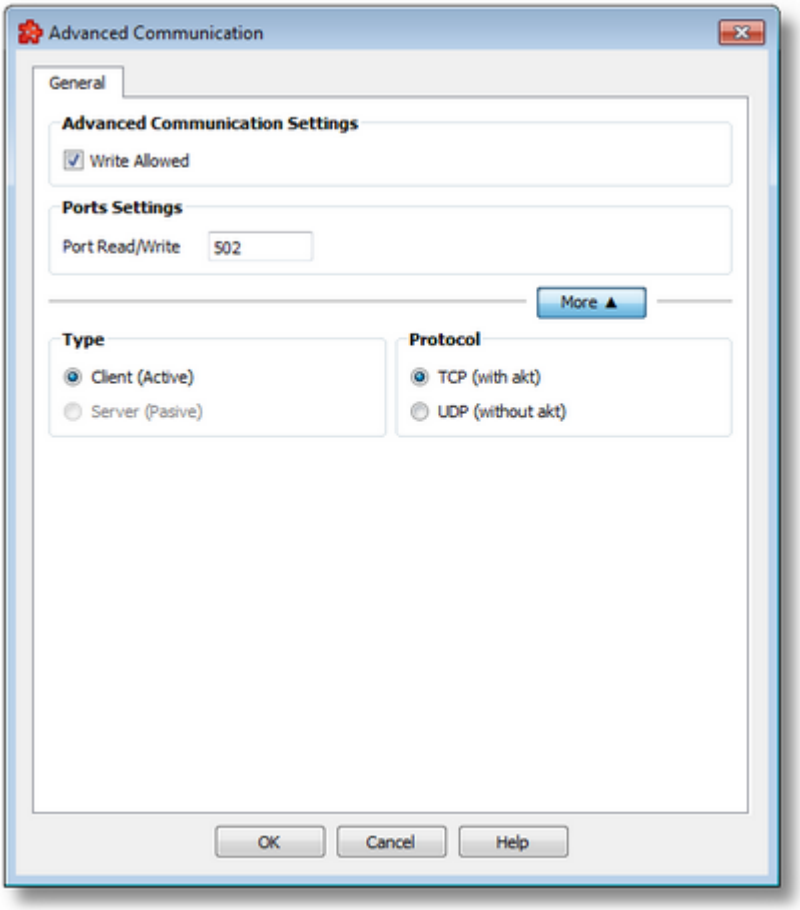
Advanced:

Read-Only field; this is a summary for TCP/IP Communication advanced settings. These settings can be changed using Modify button that lead to Advanced Communication Settings

<<< Modbus Device Wizard - Connection Settings Modbus Device Wizard - Plc Protocol Settings >>>

In this dialog specific Modbus communication parameters can be adjusted.

© Softing Industrial Automation GmbH



Write to the PLC permitted (Write Allowed)

Here you specify whether write access to the PLC is to be possible via this connection and a write connection is to be established to the hardware. You can always disable write accessing via a connection if you only want to use read accesses or if you want to be sure that no values on the PLC are to be changed. When write accessing the PLC via a Write (Write activeConnection) or a Send/Receive (Send Direct Connection for Send/Receive), certain points of the PLC are written with values (each write access is preceded by a read access).

Port for read accesses, port for write accesses

Port numbers are addresses which are used within the transport layer to address applications. Port numbers are required for TCP connections. The port is parallel to the TSAPs for RFC1006 and H1 connections. The port number is a 16-bit number from 1 to 65535. Keep the following in mind when you set up the connection.

Type active (client) / passive (server)

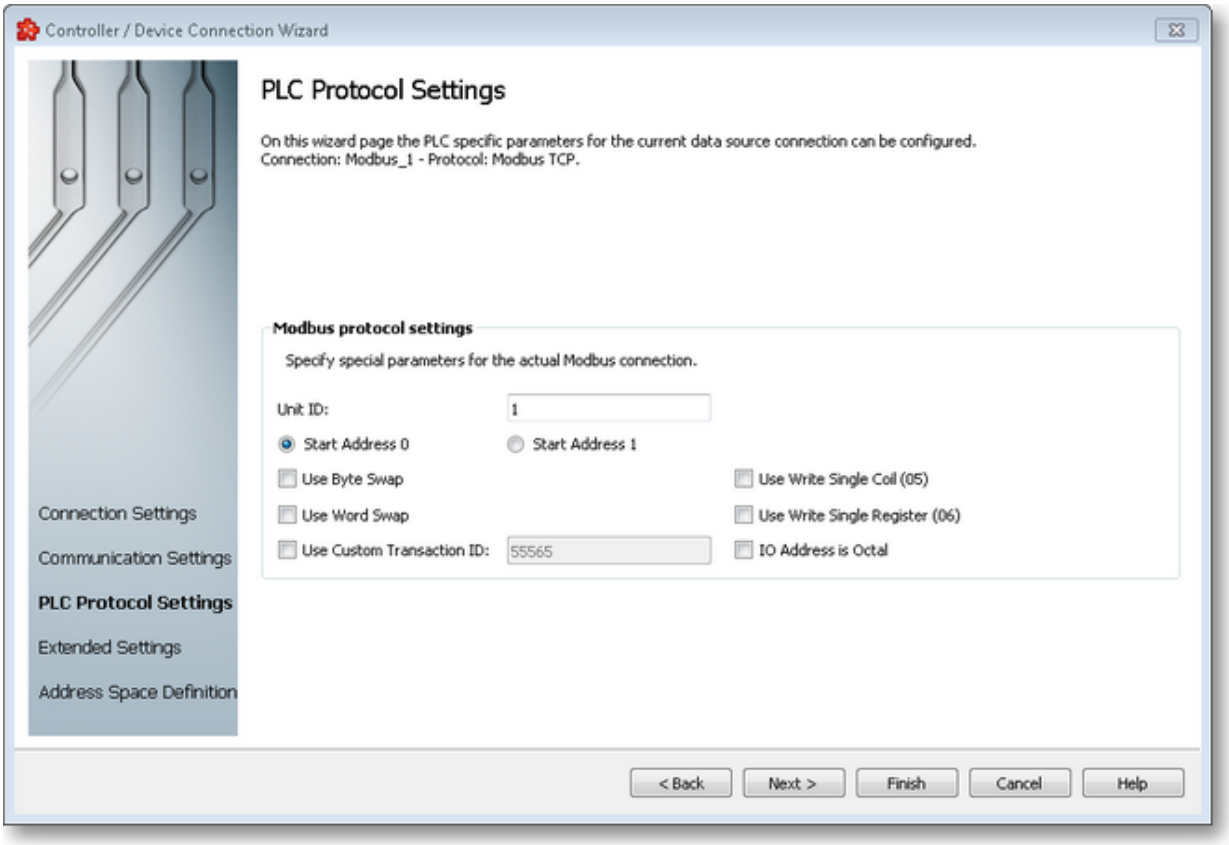
Here you specify whether the own station actively initiates the connection setup or passively expects the connection setup by the other station. You may not enter the same value on both

sides of the connection. Usually, the PC with OPC server is parameterized as active. This is also used here as the default value.

TCP/UDP protocol

The TCP is a secure protocol with checksum and confirmation. UDP is not secure and is handled via datagram services. The TCP protocol should be selected for normal data transmission.

In this page special parameters of a connection to Modbus controllers can be set.



Unit ID

- The Unit ID that is set is transmitted in the Modbus frame.

Start address 0

- The start address is transmitted as specified in the frame.

Start address 1

- Some Modbus devices expect a start address in the frame which is 1 less than the entered address. The first address is 1.

© Softing Industrial Automation GmbH

Use Byte Swap

- Switch bytes within a word

Use Word Swap

- Reverse words within a double word (32 bits)

Use Custom Transaction ID

- Enabling this checkbox, generates a random and unique Transaction ID value for the configured devices with the same destination address. The Transaction ID is a two byte value, used for the unique identification of each request (default value 55565).

Use Write Single Coil (05)

- The Modbus opcode Write Single Coil 05 is used for write accesses instead of Write Multiple Coils 15.

Use Write Single Register (06)

- The Modbus opcode Write Single Register 06 is used for write accesses instead of Write Multiple Register 16.

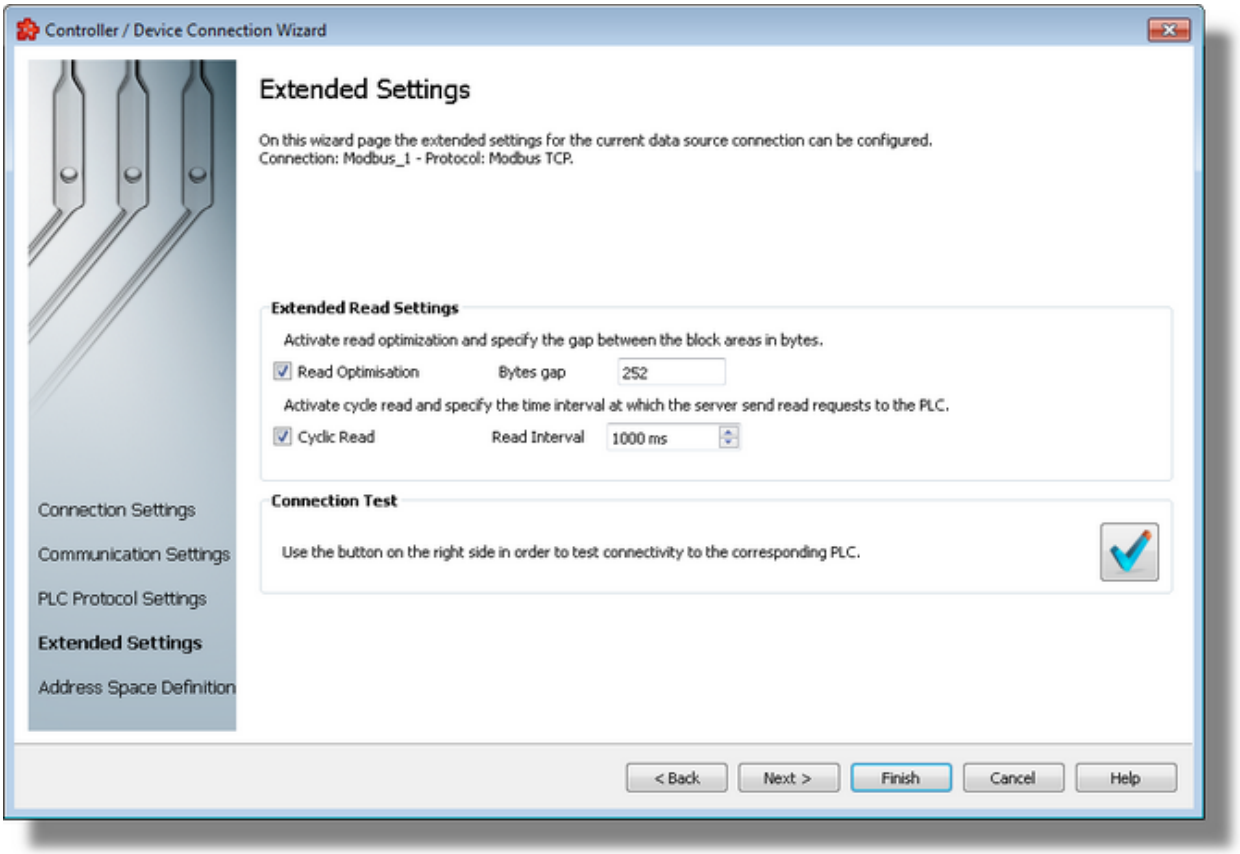
IO Address is Octal

- The start address of certain devices is octal. This means that the numbers 8 and 9 are invalid characters. This applies to the inputs and outputs. In addition, J is specified for inputs and P is specified for outputs in the item syntax.

Note: The item Unit ID specified via the item syntax is always enabled (e.g., Modbus_1.Id2.R2). For more information, go to the Modbus Item Syntax description. When specified, the item Unit ID from the item syntax will override the Unit ID value specified on the PLC Protocol Settings wizard page (e.g., using Item ID Modbus_1.Id2.R2 for the client subscribed item will use Unit ID 2, while using Item ID Modbus_1.R2 for the client subscribed item will use the Unit ID value specified in the configuration wizard).

<<< Modbus Device Wizard - Communication Settings Modbus Device Wizard - Extended settings >>>

On this wizard page the extended settings for the current data source connection can be configured.



Read optimization

Reading from the controller can be optimized here. If several variables are requested which are located in the same operand area (e.g., in a data block or in the flag area), the complete area is read from the first variable to the last variable without read optimization. Gaps between the individual block areas are also read no matter how large these gaps are. The maximum size of the gaps (in bytes) between the requested individual items within a data block without a new block being created is entered. If the gap equals the specified value, a new block is created.

Enable cyclic read access and read interval

Cyclic read access is activated as the standard. The server reads the values of the variables from the hardware via a fetch active connection at a certain time interval. It can be specified in Read interval [in ms] the time intervals at which the server is to send a read request to the PLC. The setting “0” means that it should be read as soon as possible. If "Cyclic read" is selected, read access is performed once when the client requests and no longer cyclically. However, this is only necessary in special cases and requires a precise knowledge of the communication.

Continue to Address Space definition

Optionally by checking this flag, the user can define the alias file for the current connection now. Although this step may be postponed to a later time.

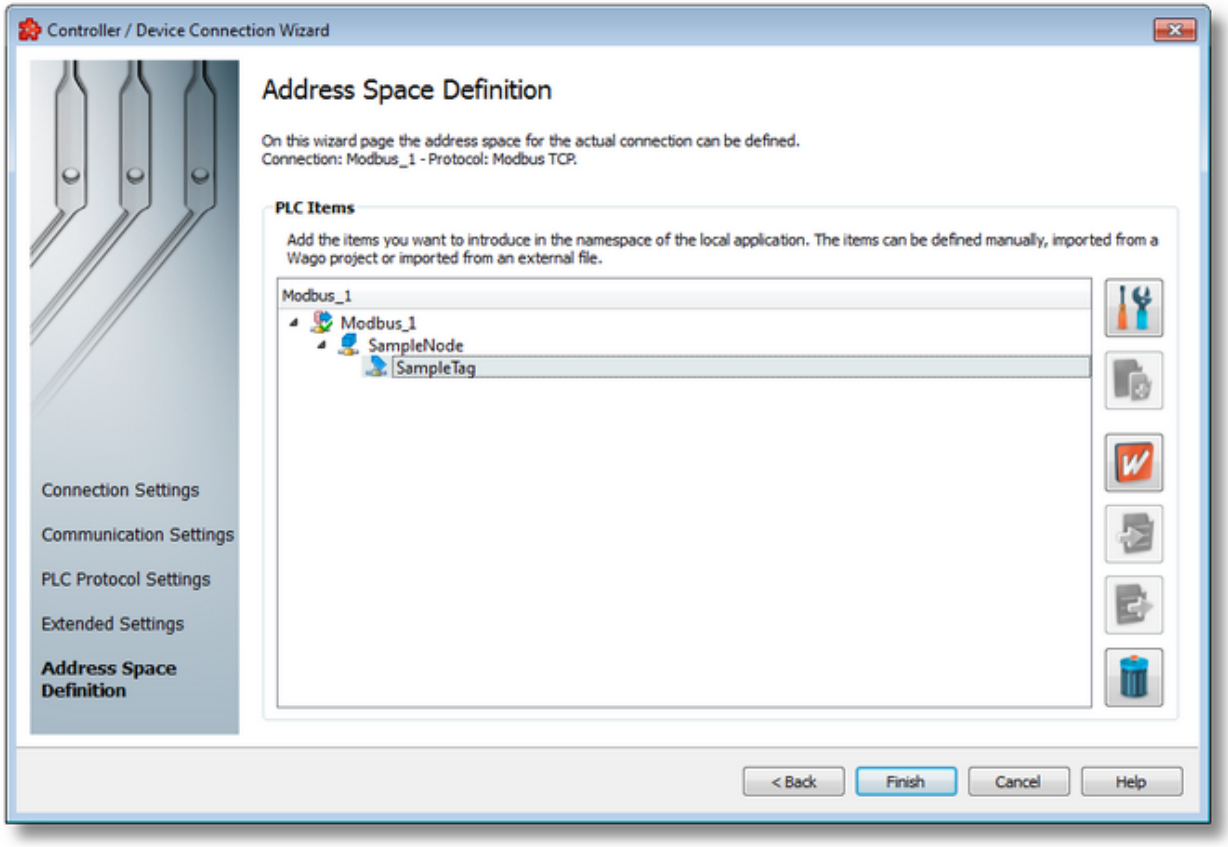
© Softing Industrial Automation GmbH

Connection Test

Use the test connection button  to check the availability of the specified PLC.

<<< Modbus Device Wizard - Plc Protocol Settings Modbus Device Wizard - Address Space Definition >>>

On this wizard page the data items for the actual PLC connection can be defined. The address space can be defined by manually adding items, by importing the address space from an external file or by importing a device symbol file (for connections of tyoe S7 and Modbus).



PLC Connection Items

An editable tree view where the user can change the address space structure or content by adding, deleting or modifying items.

There are 3 categories of items that may be used in the tree:

- **root node** (having the name of the current connection). This node is read only, the user has just the possibility to add or delete child's of this.

- **nodes:** these elements are associated with branches in tree. They can have children but they are typeless and do not carry any syntax.

Note: The items of type Node represent intermediate nodes in the tree, they are used for hierarchy purposes.

- **tags:** these elements are associated with leaves in tree.

Note: A tag cannot have any children, so Add is not available for tags.

Operation

Depending on the current selection in the *Plc Connection Items* tree, the following operations are possible using the buttons on the right, the context sensitive menu or the corresponding shortcuts:

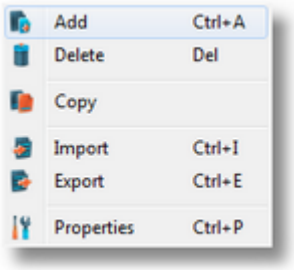
- Properties of the selected item (Ctrl+P) 
- Add a new item (Ctrl+A) 
- Import from file (Ctrl+I) 
- Export to file (Ctrl+E) 
- Delete data items (Del) 
- Import Wago symbol file 

Two more operations are supported, which are only accessible from the contextual menu:

- Copy data item(s) 
- Paste to data item(s) 

An example of the contextual menu available on *Plc Connection Items* tree is shown below:

© Softing Industrial Automation GmbH



Delete data items

Triggering the *Delete data items*  action with the corresponding button on the right wizard side or *Delete* from the context menu will remove the currently selected node from the *Data Items* namespace tree.

Attention: The root node of the namespace tree can not be deleted.

Attention: A node which is currently used in a data processing or data destination action can not be deleted.

Copy

The *Copy*  action is provided in the context menu of the *Plc Connection Items* tree. Triggering this action will copy the selected node to the local clipboard. All the properties of the node are copied. Also, if the node contains children, the entire branch is copied.

Note: The root node of the namespace tree does not provide a *Copy* action.

Paste

The *Paste*  action is provided in the context menu of the *Plc Connection Items* tree. This action will add the node currently stored in the local clipboard as a a child of the currently selected node. Whether the action is available in the context menu depends on the type of the currently selected node and the type of the node in the clipboard.

Note: The *Paste* action is only available if a node was previously copied to the clipboard.

If the *Paste* action is performed on a node which already has a child of the same name as the node in the clipboard, the pasted node is renamed automatically by adding a numerical suffix to the original name.

```
E.g. NodeA, NodeA01, NodeA02, ...
```

<<< Modbus Device Wizard - Extended Settings

The Softing Configurator allows the import of Modbus PLC connection items from an external Wago symbol file .

The import action can be triggered by using the *Import Wago symbol file* button  .

Attention: The import from Wago symbol file action applies on the entire address space of the current Modbus PLC connection. Any existing items will be automatically removed from the address space prior to performing the import.

When this import action is triggered the *Symbol File Import Wizard* is started.

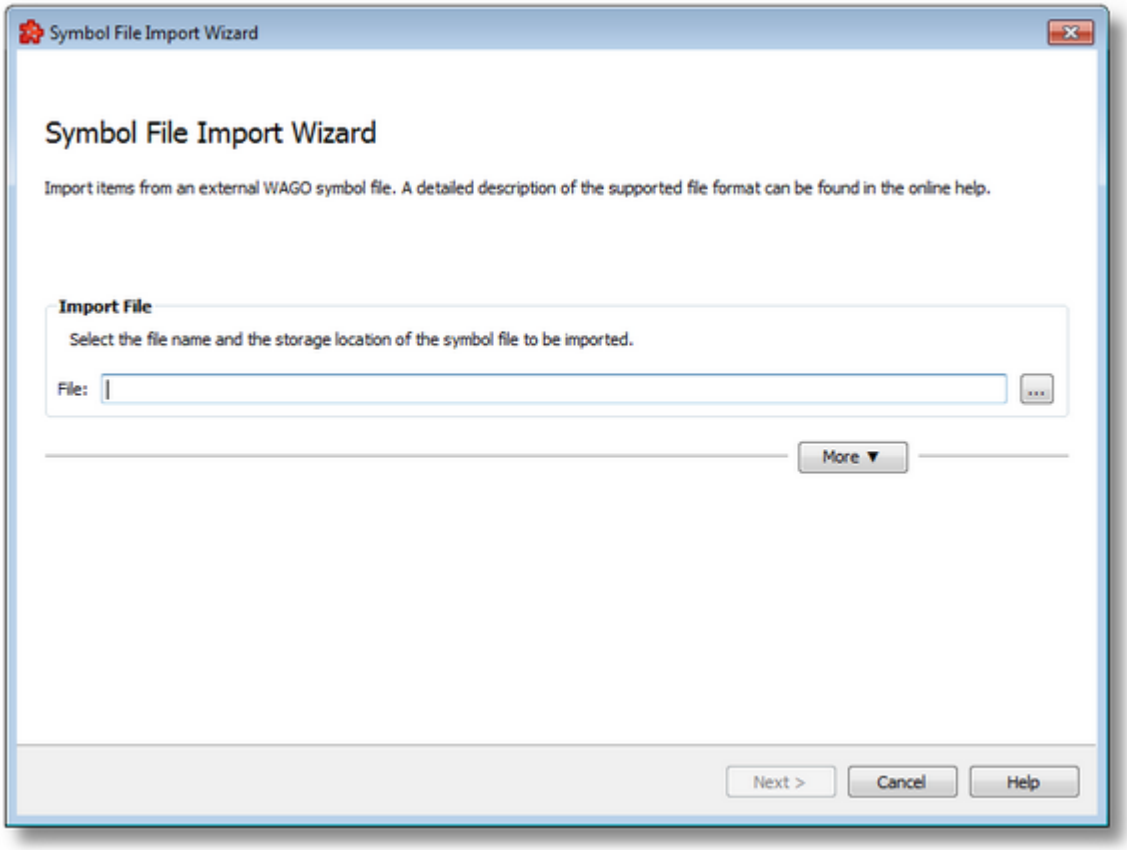
▾ The Symbol File Import Wizard allows importing different data items:

- Select the file to be imported @ Modbus Import Wizard - Import File Selection
- Examine the import results @ Modbus Import Wizard - Import Results

The Symbol File Import wizard page allows you to choose a file to be imported.

The file can be selected by manually typing its name in the *File* input box or by clicking the browse button located to the right of the input box (a standard file selection dialog will open).

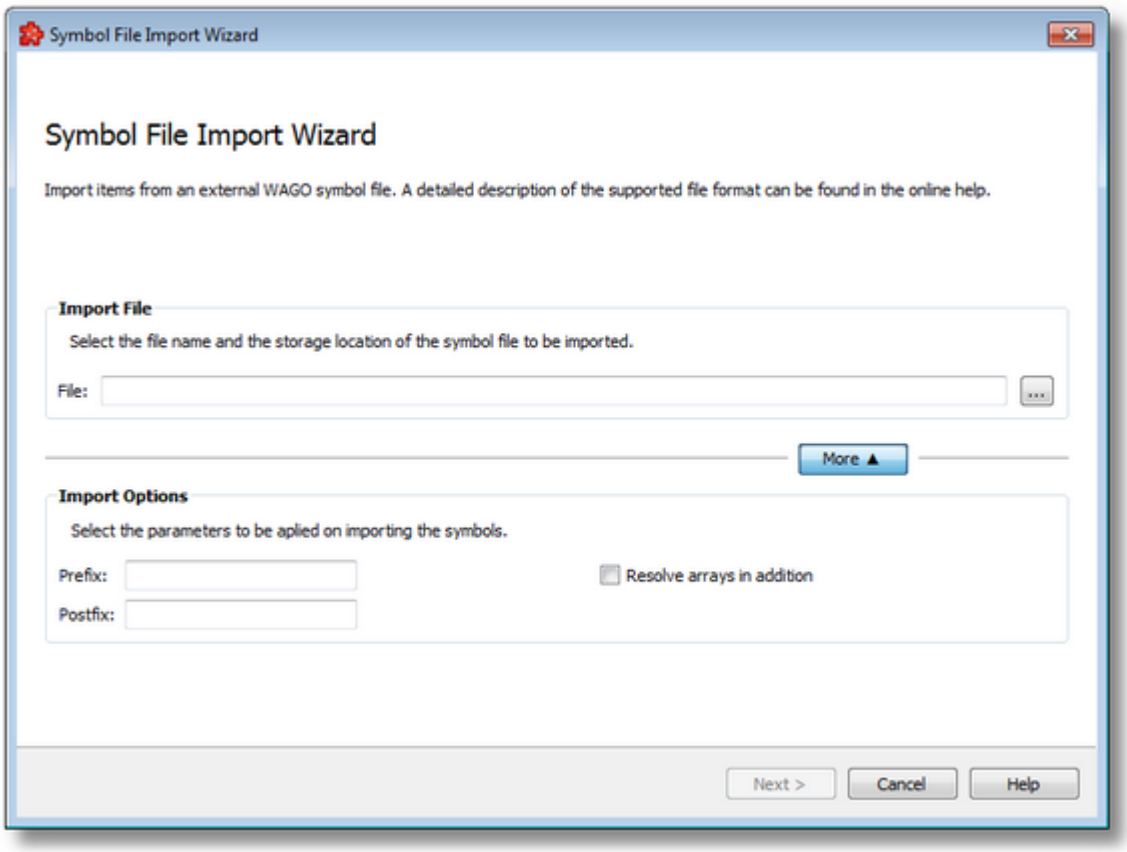
© Softing Industrial Automation GmbH



The file path and name should be valid (they should point to an existing file).

When you press the *Next* button the file path and name are verified. If the file identified by the specified file path and name does not exist an error message is displayed. The actual import begins if the selected file is valid. If the import takes longer to complete a waiting dialog is displayed. When the import operation is finished the *Import Results* page is displayed.

The import action can be parameterized with some advanced options. For this, additional controls are presented to the user by pressing the *More* button, as shown in the picture below:



The available additional options in this wizard page are described in the table below:

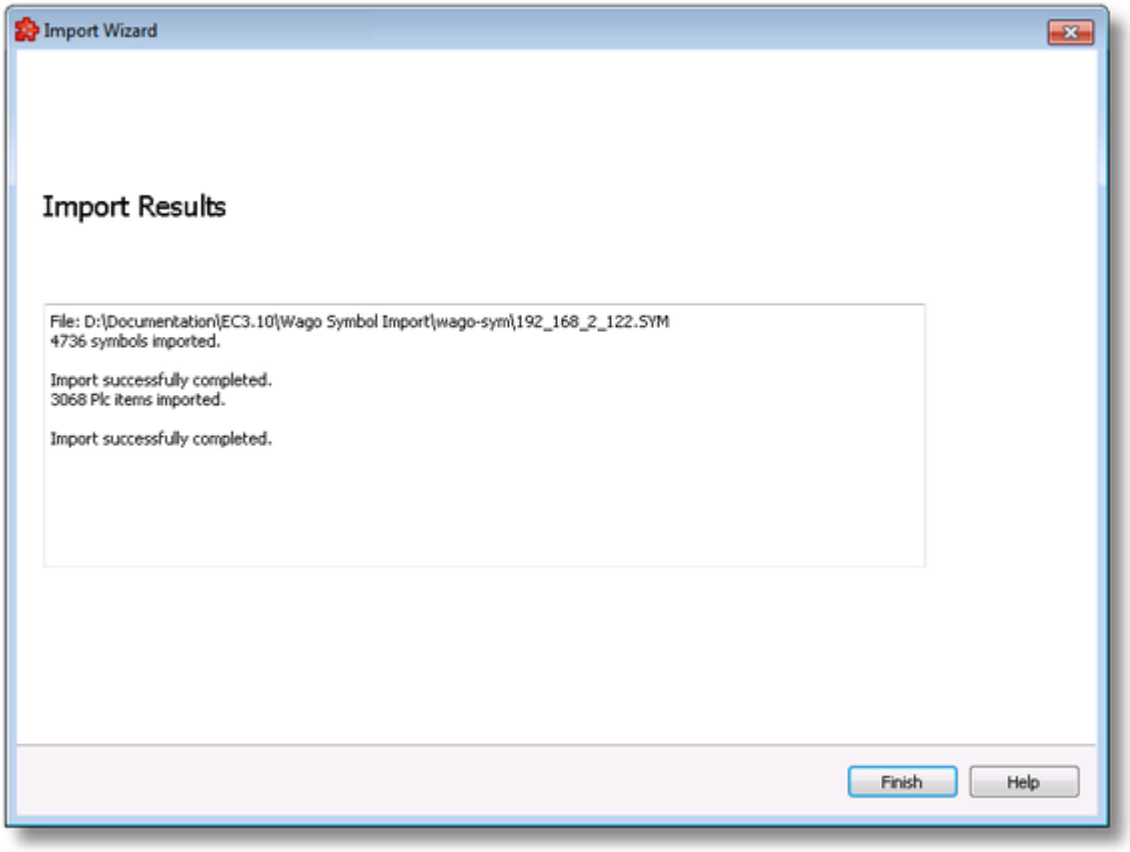
Attribute	Description
Prefix	A prefix to be added to each data item which is created as result of the import.
Postfix	A postfix to be added to each data item which is created as result of the import.
Resolve arrays in addition	Every single element of an array is given its own icon name. Default value is switched off.

Note: The usage of the hierarchical separator character in the *Prefix* and *Postfix* fields is allowed and can be used to change/adapt the hierarchy of the imported data items.

Attention: The usage of *Resolve arrays in addition* option may greatly enlarge the size of the resulting address space and this takes more time during program start.

The Import Results wizard page displays the import results.

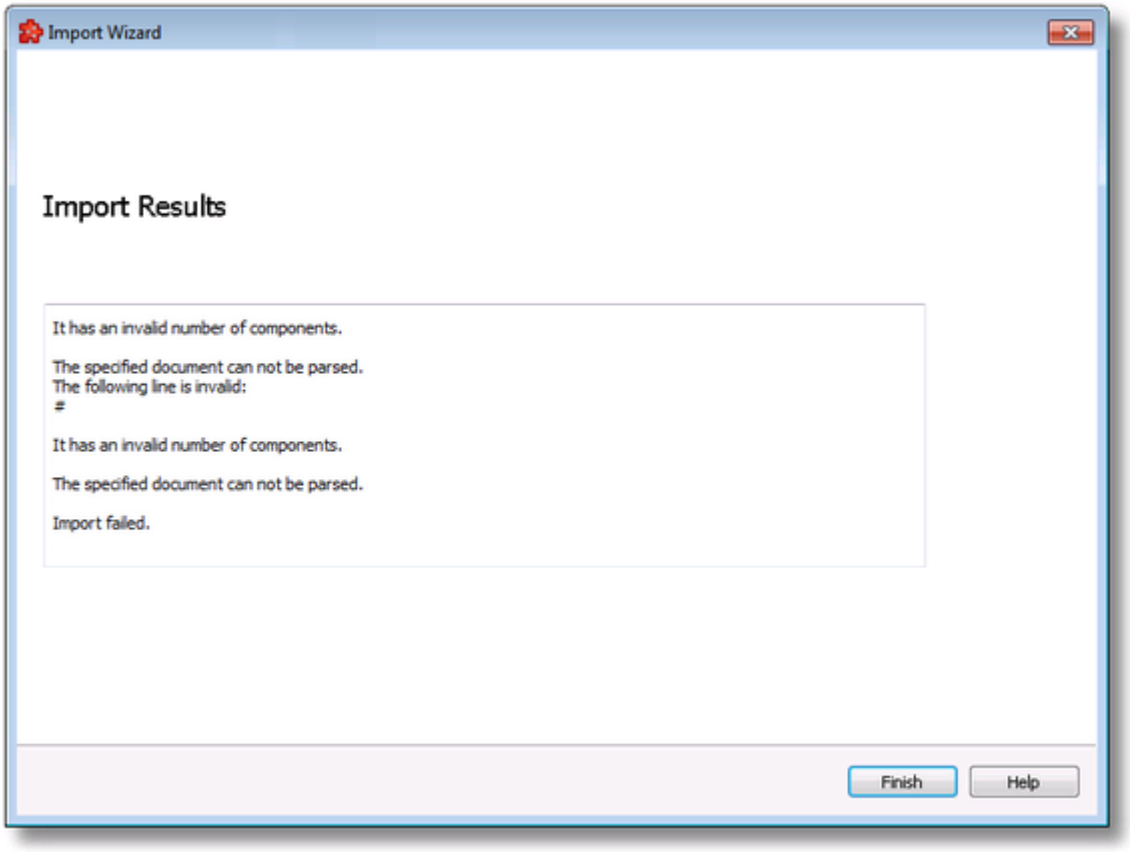
© Softing Industrial Automation GmbH



This page is the final page of the *Import Wago Symbol File Wizard*. A back button is not provided as the import action cannot be undone. The import can be successful or it can fail:

1. After a successful import the results page contains the following information:
   - The absolute path of the imported file.
   - The number items imported. This number includes only the items of type Tag, the items of type Node are not counted.
   - The confirmation message that the import was successful.
2. After a failed import the results page contains the following information:
   - The invalid line where the import failed.
   - The reason why the line is considered invalid.
   - The confirmation message that the import failed.

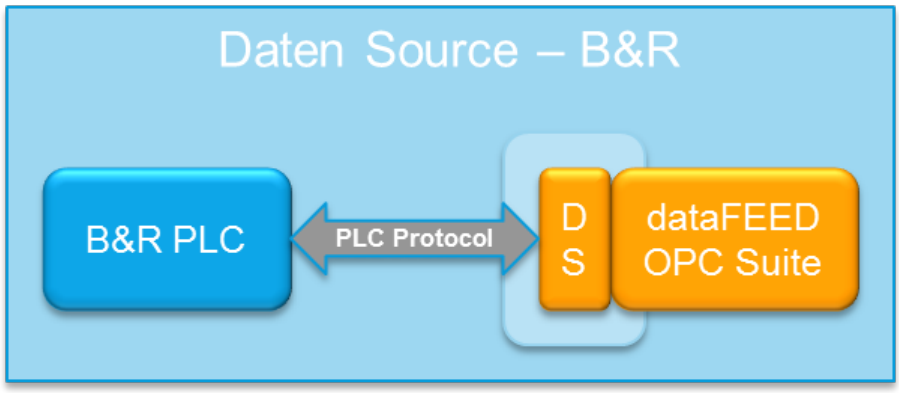
An *Import Results* example for a failed import operation can be seen below:



When the import wizard is closed after a successful import, the new items are added to the current PLC connection tree and all the previous data items are removed.

7.3.1.5 B&R Devices

This section of the Softing dataFEED OPC Suite Online Help describes how to configure B&R devices within Softing dataFEED OPC Suite.

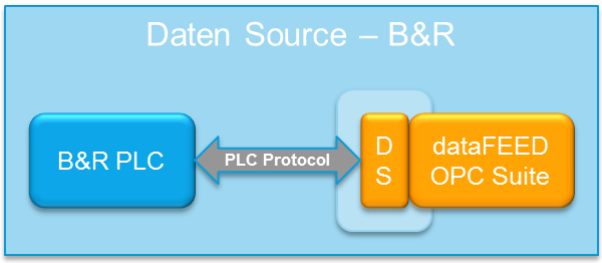


Controllers and devices data source makes it possible to exchange data between the field devices (PLCs) of a wide variety of manufacturers and OPC clients via Ethernet TCP/IP.

© Softing Industrial Automation GmbH

A detailed description of how to configure access to PLC devices functionality can be found in the Configuration section.

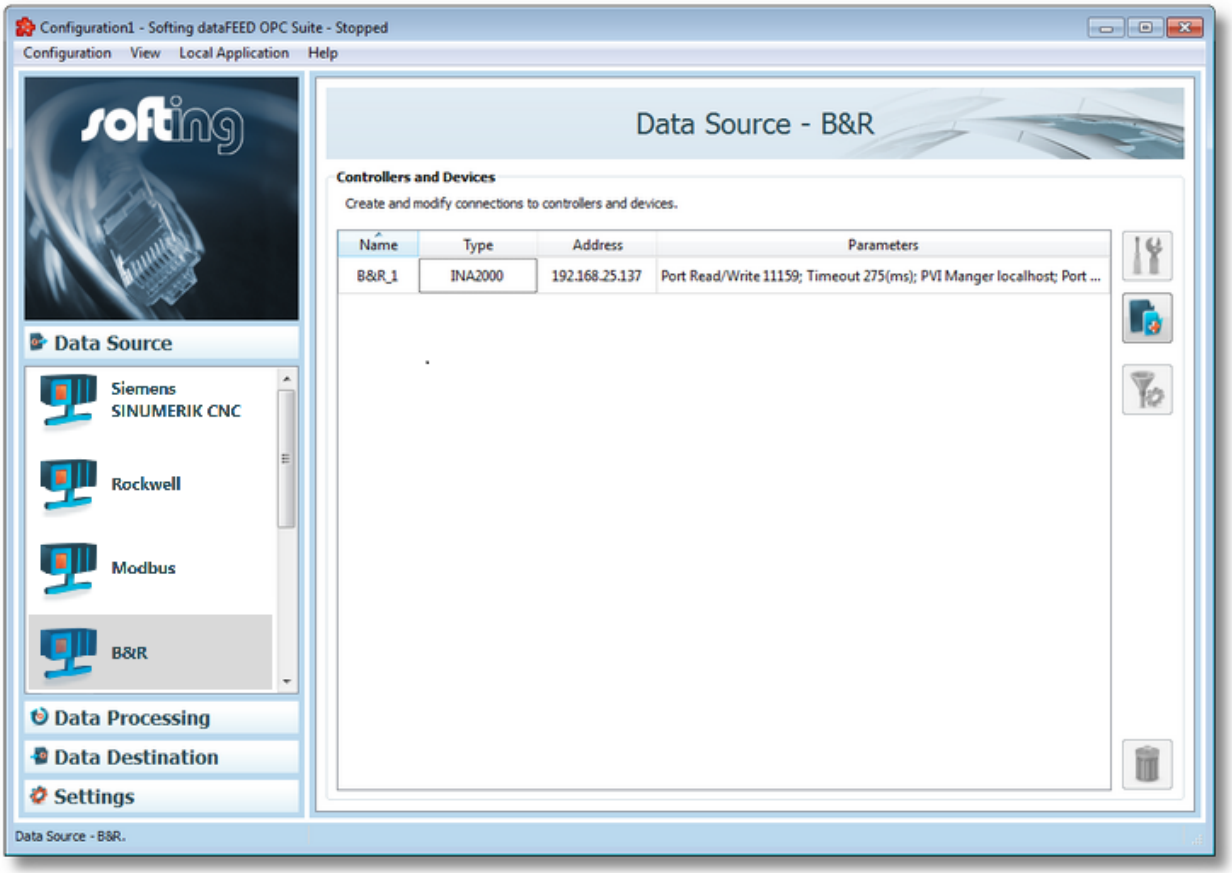
7.3.1.5.1 Configuration



To start configuring the connection to an external B&R device select B&R Devices from the Data Source group in the navigator on the left.

The Data Source - B&R Device main user interface is displayed. The main element of this page is the External Devices list.

Each entry shown in the list of the configuration section represents a connection to an external device.



Note: The list is only shown and editable if a configuration has been loaded.

Each Device connection has a unique name within the configuration, which is given in the *Name* column.

The name of the device will be shown in the namespace tree in the Softing dataFEED OPC Suite Configurator and in the local dataFEED OPC Suite application, and its entire namespace will be under this name.

Note: Device name must be unique string within a dataFEED OPC Suite configuration. Also the name could not contain address space delimiters, multi-byte characters, empty spaces.

The device type (second column) can be B&R.

Third column: address represents the actual device address which can be IPv4, hostname or IPv6.

Parameters column contains various attributes of the connection like: plc header type, write allowed, ports, communication protocol.

Operation

From the Siemens Devices main user interface certain operations can be performed over the selected connection by using the buttons on the right, by accessing the context sensitive menu or by pressing the keyboard shortcuts. The following actions are available:

- Change the properties of the selected data source by using the *Properties* button  , pressing the keyboard shortcut CTRL+P or selecting *Properties* from the context menu.
- Add a new connection by using the *Add* button  , pressing the keyboard shortcut CTRL+A or selecting *Add* from the context menu.
- Delete the selected data source by using the *Delete* button  , pressing the DEL / DELETE key on the keyboard or selecting *Delete* from the context menu.
- Address space definition button  , by pressing it the Siemens Device Wizard will be opened in it's last page. For more details see Address Space Definition page.

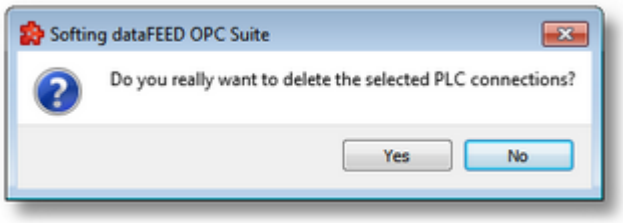
Note: If no controller / device connection is selected, the *Add a new data source* button is the only one available.

The *Delete* button  will remove the selected device connections from the list.

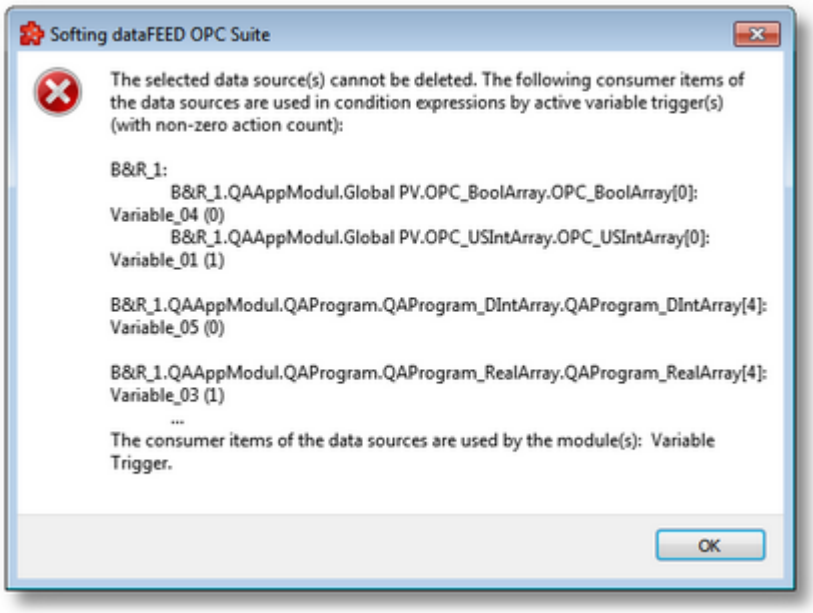
The user can select more than one server connection with the help of the CTRL or SHIFT keys.

A dialog will then prompt the user to confirm the deletion.

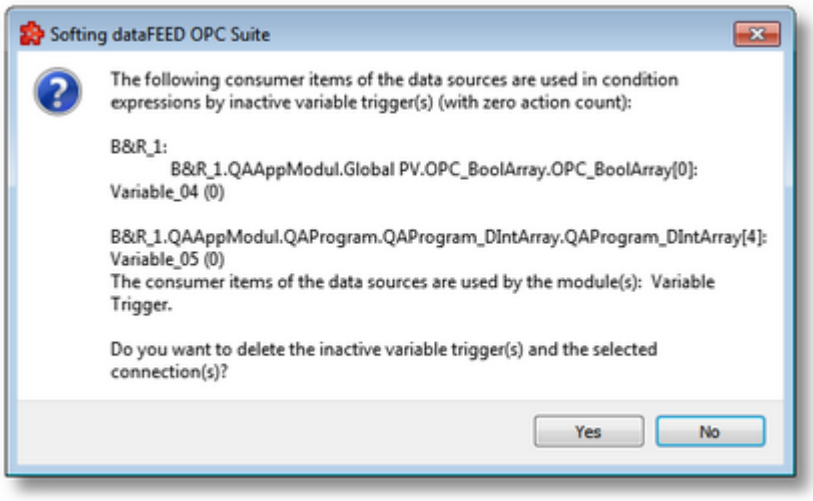
© Softing Industrial Automation GmbH



A dialog will inform the user that the data sources cannot be deleted, if consumer items and/or system items of the selected data sources are used by active variable triggers or other modules.



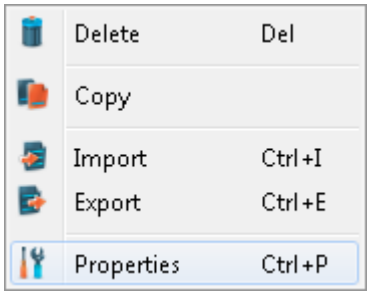
A dialog will prompt the user to confirm deletion of the data sources and inactive variable triggers, if consumer items and/or system items of the selected data sources are used only by inactive variable triggers.



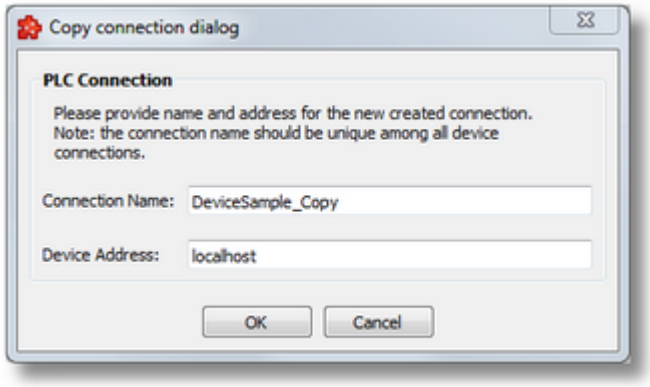
Attention: A device connection cannot be deleted if at least one data item defined in its namespace is used in a Processing or Data Destination action.

All these actions are also provided in the context menu of the External device list. Which actions are available depends on the current selection in the list.

An example of the context menu can be seen below:



- Copy device connection  . This action is available only in the context menu and it leads to a dialog where the user may provide a name and device address. See the screen shot below:



The proposed name is the original connection's

3 name post fixed with "_Copy" and the default address will be "localhost".

- Toggle Activate state  . This action will enable / disable a devices connection. A disabled connection will be shown as grayed out in the device connection list and it will be ignored by runtime.

Note: The connection name should be unique among all device connections names.

Note: The device address should have a valid host name or an IP address in format IPv4 or IPv6.

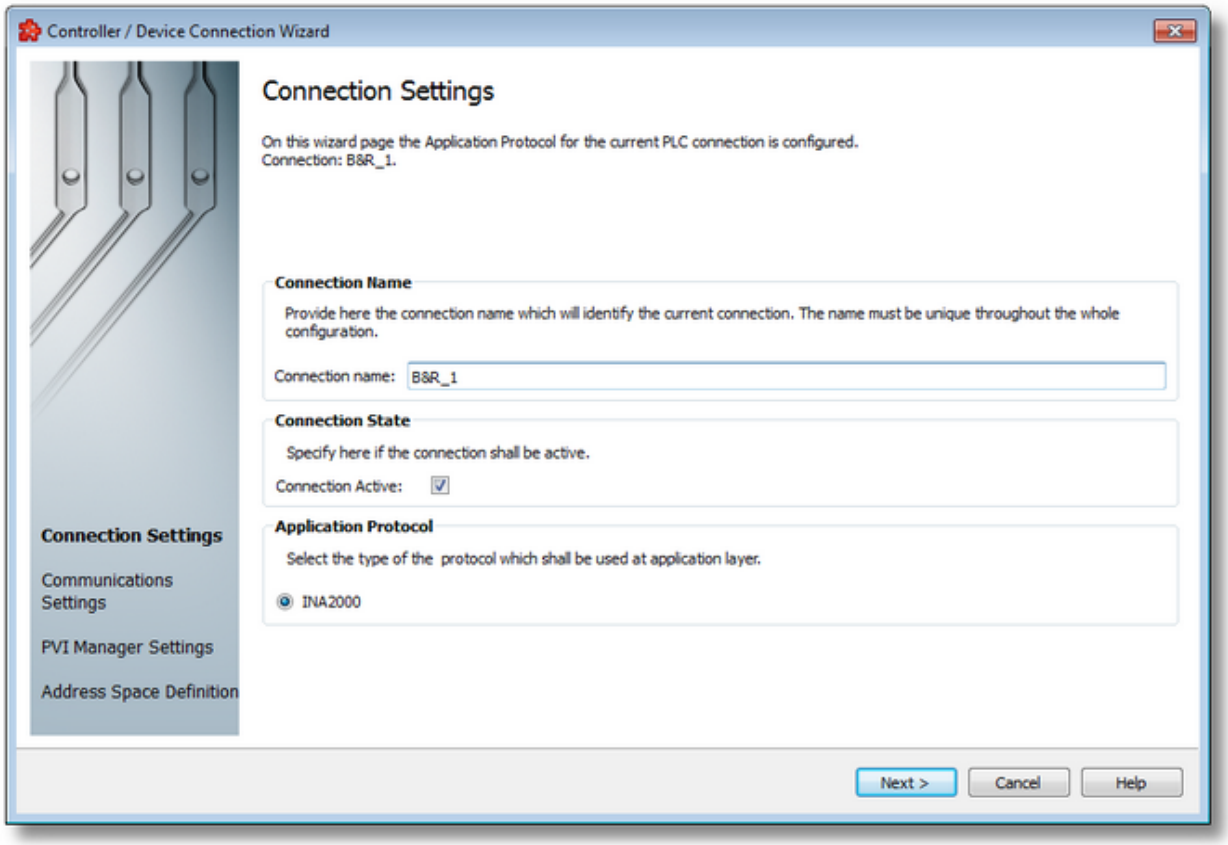
7.3.1.5.2 B&R Device Wizard

- With the B&R Device Wizard you can create and modify a data source connection to an external PLC device:

© Softing Industrial Automation GmbH

- Specify connection settings @ B&R Device Wizard - Connection Settings
- Specify communication parameters @ B&R Device Wizard - Communication Settings
- Extended settings @ B&R Device Wizard - PVI Manager Settings
- Filter settings @ B&R Device Wizard - Filter Settings
- Address space definition @ B&R Device Wizard - Address Space Definition

On this page the connection name that will uniquely identify the data source and some generic connection settings can be provided.



Connection Name

An unique identifier for the current connection. In case the provided name is already used by other connection, the user use is not allowed to go into the next wizard page.

Connection active

This flag specifies if the current connection will be enabled or not.

Simulation

This flag specifies if the current connection will be simulated by the server or it will represent a real device connection.

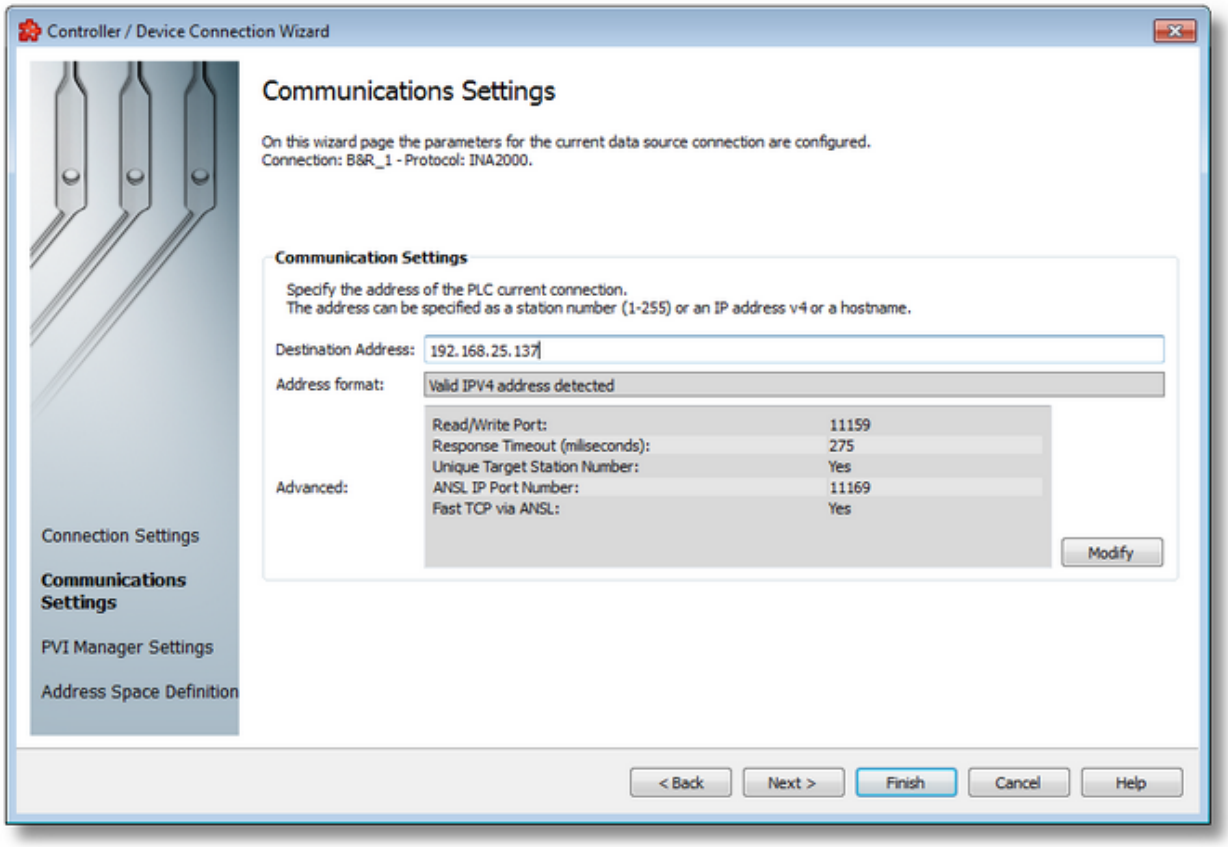
Application protocol

Currently supported protocol is

- INA2000

B&R Device Wizard - Communication Settings >>>

On this wizard page various parameters for the current TCP/IP connection are configured.



Destination Address: IP address / hostname or station number

This is where the IP address of the target station (IP address of the PLC) or the DNS name of the target station is entered.

© Softing Industrial Automation GmbH

The device can be identified by a station number (1 -255).

Address Format

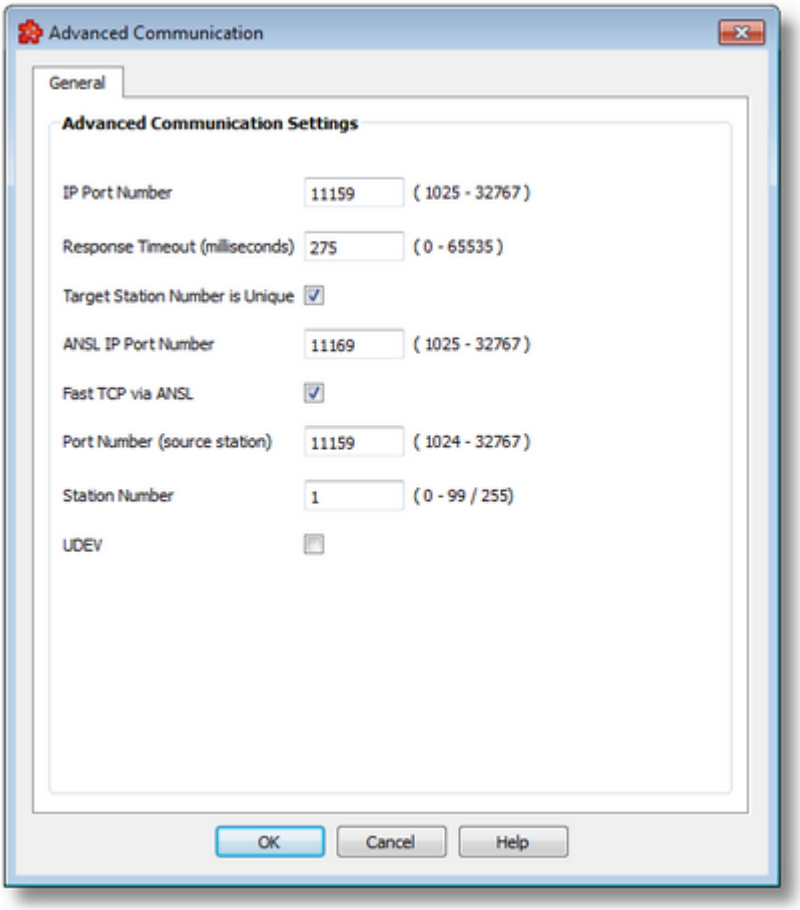
Read-Only field; automatically detects the format of destination address.

Advanced:

Read-Only field; this is a summary for Communication advanced settings. These settings can be changed using Modify button that lead to Advanced Communication Settings

<<< B&R Device Wizard - Connection Settings B&R Device Wizard - PVI Manager Settings >>>

In this dialog specific B&R communication parameters can be adjusted.



IP port number of the target station.

If the value is specified in Hexadecimal, then "0x" must be used to introduce the value (e.g. "/LOPO=0x2b97").The port number must match the setting on the target station (PLC).

Response timeout in ms.

If a response to a sent frame is not received with this time, then the frame is transferred again. After the third retry, the connection is deemed canceled.

The time in which an interrupted connection is detected is a result of ResponseTimeout * 4.

Target Station Number is unique.

Enables or disables the check to see if the target station number (parameter /DA) is unique.

ANSL IP port number.

ANSL IP port number of the target station (peer station) for fast TCP download/upload.

Fast TCP via ANSL

Turn on/off fast TCP download/upload via ANSL.

Port number of the source station (separate station)

If the value is specified in Hexadecimal, then "0x" must be used to introduce the value (e.g. "/LOPO=0x2b97").

A different default port number then only needs to be set if it is not unique within the local computer.

Station number of the source station (own station).

The station number must be unique within the INA2000 network. The highest station number is 99 for SG3 and SGC with AR Version < 2.00, or 255 for SG4 and SGC with AR Version >= 2.00.

UDEV

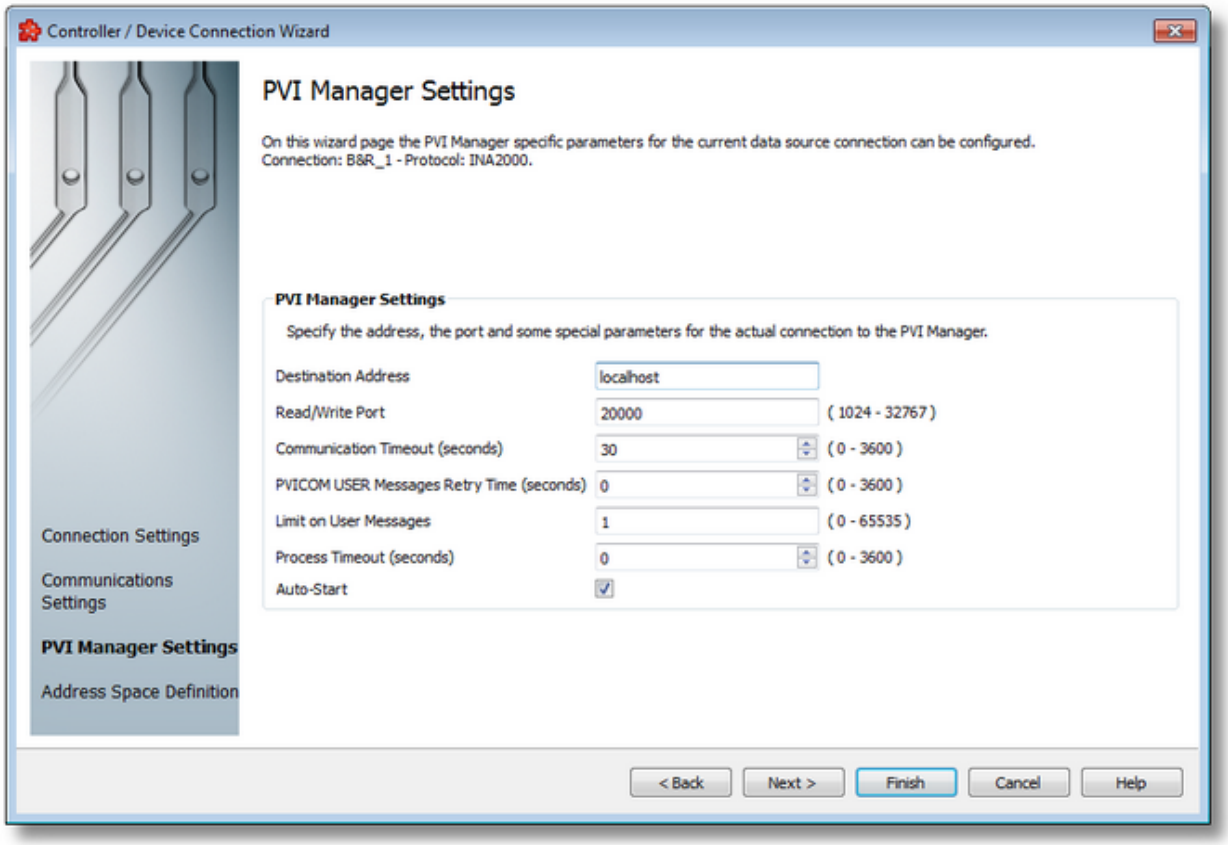
This parameter determines how the parameters /LOPO and /SA function.

With "/UDEV=0" (default) all process objects with the type "device object" are grouped together as one service object and transferred over an INA2000 connection. If the process objects have different values for /LOPO and /SA, the object created first always wins. This setting corresponds to the behavior of PVI Versions <= 3.0.0.

© Softing Industrial Automation GmbH

With "/UDEV=1", process objects with different values for /LOPO and /SA are transferred using separate service objects. INA2000 communication therefore also takes place via separate INA2000 connections. This setting allows multiple INA2000 connections to be established to one PLC with different port numbers or station numbers (source station).

On this wizard page the extended settings for the current data source connection can be configured.



Destination address.

If the IP initialization parameter is specified, then the Remote communication type is used for PVICOM communication. Either the TCP/IP address (e.g. "IP=10.43.50.62") or the host name ("IP=SystemPC1") of the server (PVI Manager) is specified with this parameter. If the host name is given, then the IP address assigned to the name is determined by the system.

Read/Write port

TCP/IP port number. Default: "PN=20000".

This parameter is only required in conjunction with the IP parameter. Value: 1024 - 32767.

Communication timeout.

Communication timeout for the client/server (application/PVI Manager) communication in seconds.

The timeout also defines the time for the life sign monitor. An additional period is needed to detect a disruption in the client/server connection. To determine the total time where a communication disruption would be detected (at the latest), the timeout value specified here has to be doubled. If the timeout is 5, then an interruption in the connection can be detected in as early as 5 seconds and as late as 10 seconds. Connection interrupts are not recognized as such by the PVICOM interface if they occur within the timeout period. Data transfer simply continues once the connection is reestablished. Often used values are 3 - 10 seconds for local communication and 5 - 30 seconds for remote communication.

Value: 1 - 3600, = 0 ... Default value of 30 s used. The value specified in the function argument "Timeout" serves as a presetting for this parameter.

Retry time for PVICOM user messages is seconds.

The default setting ("RetryTime=0") should normally be used for this parameter.

Value: 1 - 3600, <= 0 ... no repetition. The value specified in the function argument "RetryTime" serves as a presetting for this parameter.

Limit on user messages.

Default: "LM=1".

PVICOM applications basically do not need to change the default setting of this parameter. However, if an application is able to respond quickly to PVICOM user messages, then a higher limit value ("LM=100", "LM=500") can speed up user message execution.

Value: 1 - 65535, <= 0 -> No limiting.

Process timeout in seconds.

Default: "PT=0".

This timeout can be used to monitor the time between requests and responses. This parameter is only intended for PVICOM applications which carry out particular tasks. Standard application do not need to be monitored and should use the default setting (no timeout monitoring).

Value: 1 - 3600, <=0 -> No timeout monitoring.

Auto-Start (only for the local communication type).

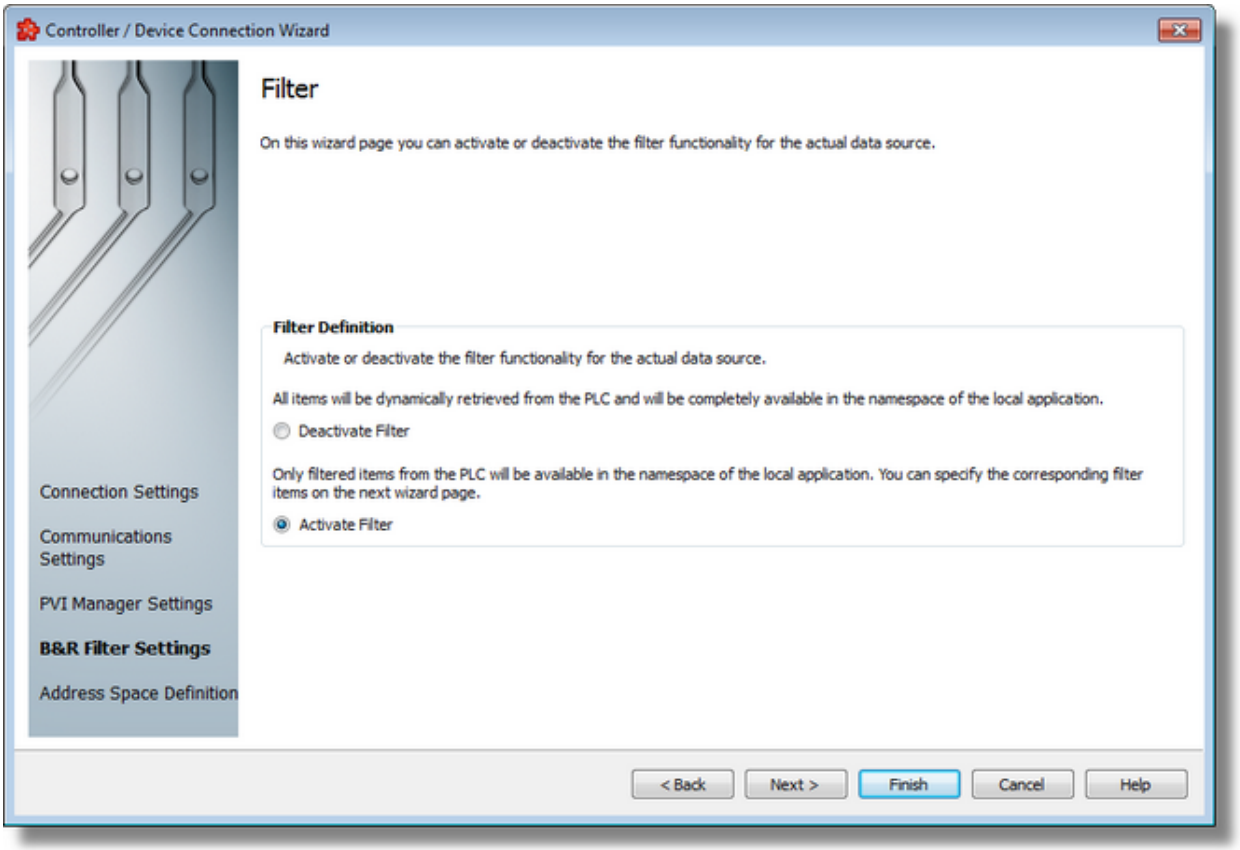
1 ... PVI Manager is started (default).

© Softing Industrial Automation GmbH

0 ... PVI Manager is not started automatically and must be started already when the function is called.

<<< B&R Device Wizard - Communication Settings B&R Device Wizard - Filter Settings >>>

On this wizard page you can activate or deactivate the filter functionality for the selected data source.



Filter Definition

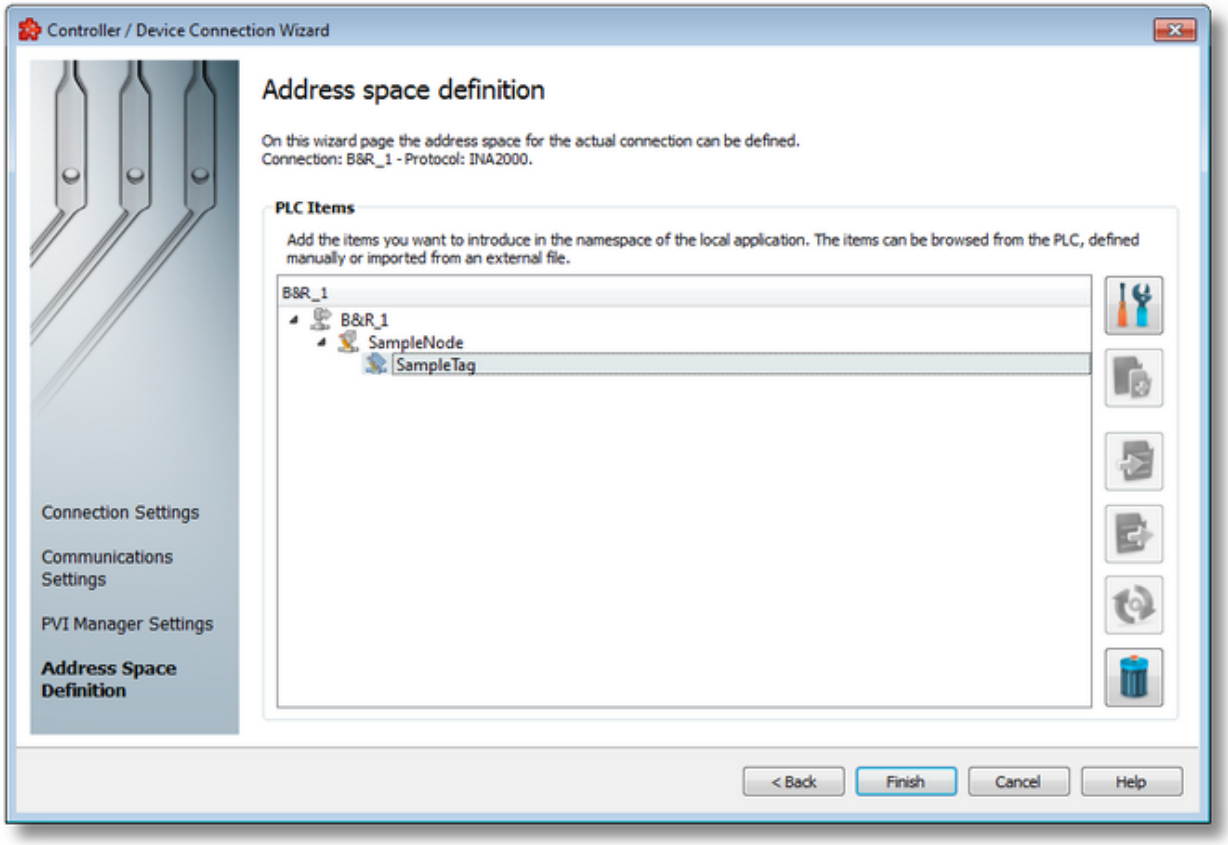
For the *Filter Definition* there are two possible configuration settings:

Filter Definition	Description
Deactivate Filter	All items will be dynamically retrieved from the device and will be completely available in the namespace of the local dataFEED OPC Suite application.

Activate Filter	Only filtered items from the device will be available in the namespace of the local dataFEED OPC Suite application. You can specify the filter items on the next wizard page.

<<< B&R Device Wizard - PVI Manager Settings B&R Device Wizard - Address Space Definition >>>

On this wizard page the data items for the actual connection can be defined. The address space can be defined by manually adding items, by importing the address space from an external file.



PLC Connection Items

An editable tree view where the user can change the address space structure or content by adding, deleting or modifying items.

There are 3 categories of items that may be used in the tree:

- **root node** (having the name of the current connection). This node is read only, the user has just the possibility to add or delete child's of this.
- **nodes:** these elements are associated with branches in tree. They can have children but they are typeless and do not carry any syntax.

© Softing Industrial Automation GmbH

Note: The items of type Node represent intermediate nodes in the tree, they are used for hierarchy purposes.

- **tags:** these elements are associated with leaves in tree.

Note: A tag cannot have any children, so Add is not available for tags.

Operation

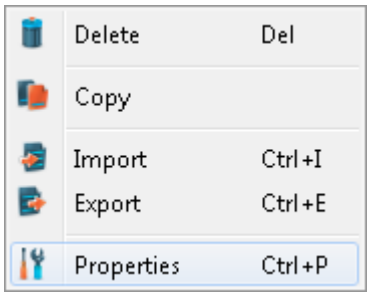
Depending on the current selection in the *Plc Connection Items* tree, the following operations are possible using the buttons on the right, the context sensitive menu or the corresponding shortcuts:

- Properties of the selected item (Ctrl+P) 
- Import from file (Ctrl+I) 
- Export to file (Ctrl+E) 
- Delete data items (Del) 
- Refresh data items (Ctrl+R) 

Two more operations are supported, which are only accessible from the contextual menu:

- Copy data item(s) 
- Paste to data item(s) 

An example of the contextual menu available on *Plc Connection Items* tree is shown below:



Delete data items

Triggering the *Delete data items*  action with the corresponding button on the right wizard side or *Delete* from the context menu will remove the currently selected node from the *Data Items* namespace tree.

Attention: The root node of the namespace tree can not be deleted.

Attention: A node which is currently used in a data processing or data destination action can not be deleted.

Copy

The *Copy*  action is provided in the context menu of the *Plc Connection Items* tree. Triggering this action will copy the selected node to the local clipboard. All the properties of the node are copied. Also, if the node contains children, the entire branch is copied.

Note: The root node of the namespace tree does not provide a *Copy* action.

Paste

The *Paste*  action is provided in the context menu of the *Plc Connection Items* tree. This action will add the node currently stored in the local clipboard as a a child of the currently selected node. Whether the action is available in the context menu depends on the type of the currently selected node and the type of the node in the clipboard.

Note: The *Paste* action is only available if a node was previously copied to the clipboard.

If the *Paste* action is performed on a node which already has a child of the same name as the node in the clipboard, the pasted node is renamed automatically by adding a numerical suffix to the original name.

```
E.g. NodeA, NodeA01, NodeA02, ...
```

Refresh data items

Triggering the Refresh *data items*  action with the corresponding button on the right wizard will perform a device query to retrieve the actual items.

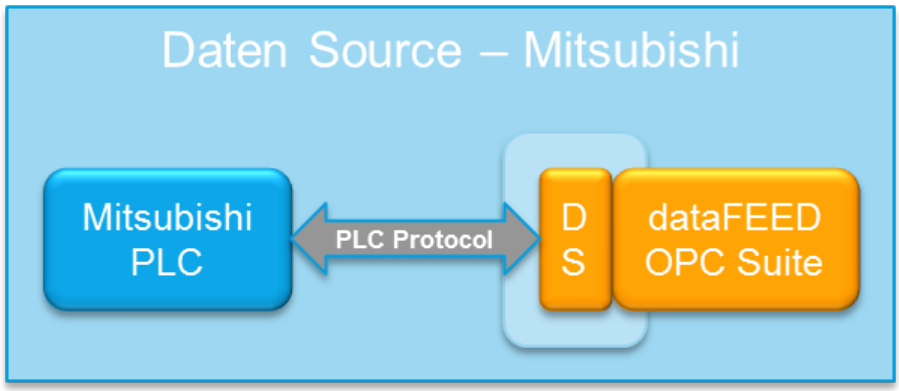
Note: This operation may be very time consuming, depending on device configuration and connection speed.

© Softing Industrial Automation GmbH

<<< B&R Device Wizard - Filter Settings

7.3.1.6 Mitsubishi Devices

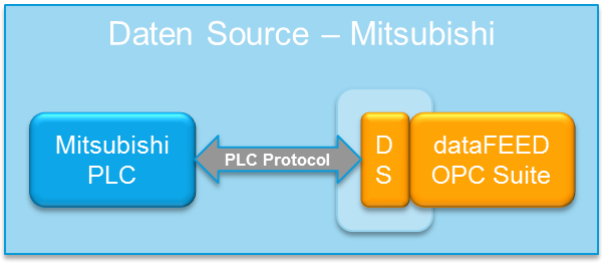
This section of the Softing dataFEED OPC Suite Online Help describes how to configure Mitsubishi devices within Softing dataFEED OPC Suite.



Controllers and devices data source makes it possible to exchange data between the field devices (PLCs) of a wide variety of manufacturers and OPC clients via Ethernet TCP/IP.

A detailed description of how to configure access to PLC devices functionality can be found in the Configuration section.

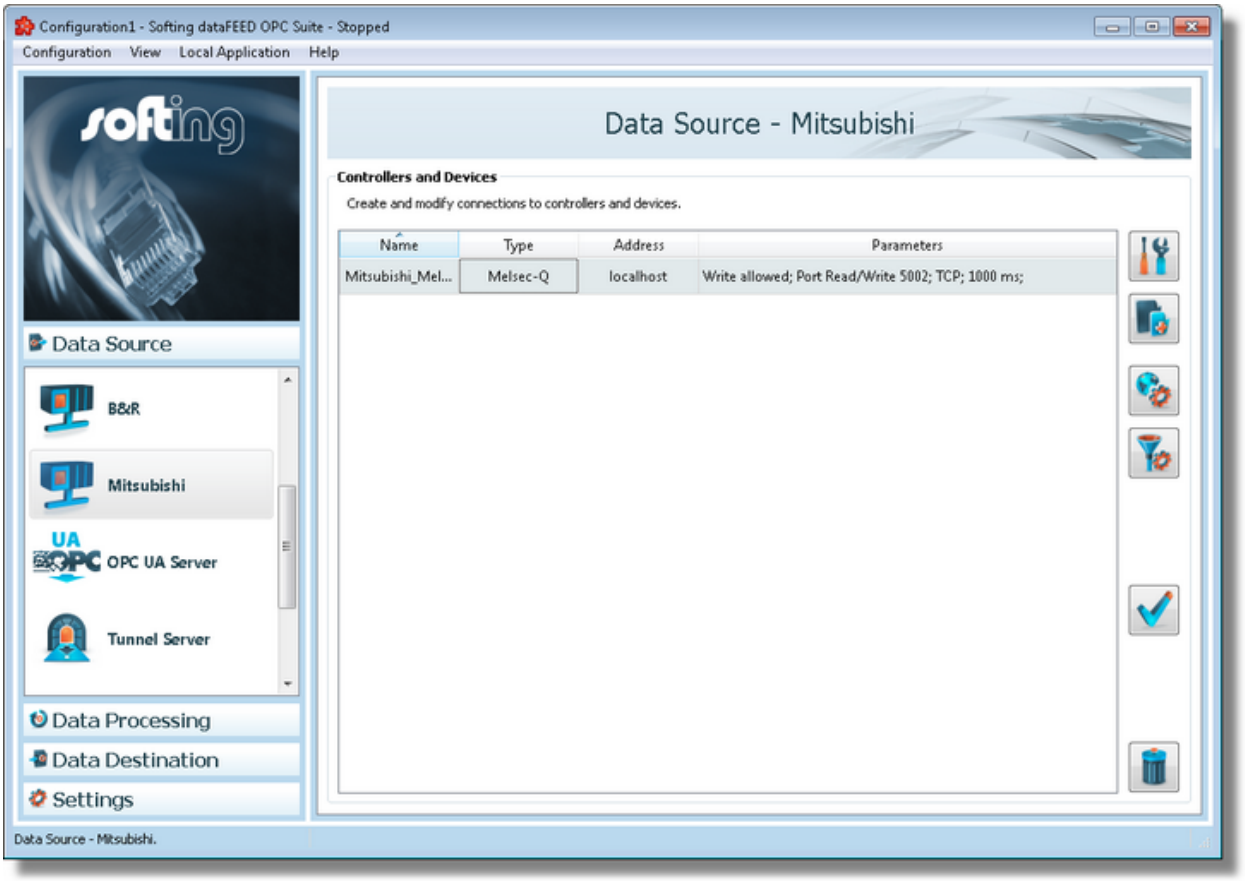
7.3.1.6.1 Configuration



To start configuring the connection to an external Mitsubishi device select Mitsubishi Devices from the Data Source group in the navigator on the left.

The Data Source - Mitsubishi Device main user interface is displayed. The main element of this page is the External Devices list.

Each entry shown in the list of the configuration section represents a connection to an external device.



Note: The list is only shown and editable if a configuration has been loaded.

Each Device connection has a unique name within the configuration, which is given in the *Name* column.

The name of the device will be shown in the namespace tree in the Softing dataFEED OPC Suite Configurator and in the local dataFEED OPC Suite application, and its entire namespace will be under this name.

Note: Device name must be unique string within a dataFEED OPC Suite configuration. Also the name could not contain address space delimiters, multi-byte characters, empty spaces.

The device type (second column) can be Mitsubishi.

Third column: address represents the actual device address which can be IPv4, hostname or IPv6.

Parameters column contains various attributes of the connection like: plc header type, write allowed, ports, communication protocol.

Operation

© Softing Industrial Automation GmbH

From the Mitsubishi Devices main user interface certain operations can be performed over the selected connection by using the buttons on the right, by accessing the context sensitive menu or by pressing the keyboard shortcuts. The following actions are available:

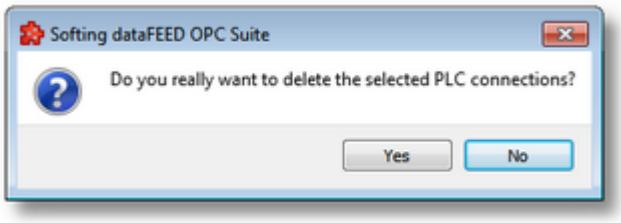
- Change the properties of the selected data source by using the *Properties* button  , pressing the keyboard shortcut CTRL+P or selecting *Properties* from the context menu.
- Add a new connection by using the *Add* button  , pressing the keyboard shortcut CTRL+A or selecting *Add* from the context menu.
- Change global settings for the selected data source by using the Global settings button  .
- Delete the selected data source by using the *Delete* button  , pressing the DEL / DELETE key on the keyboard or selecting *Delete* from the context menu.
- Address space definition button  , by pressing it the Mitsubishi Device Wizard will be opened in it's last page. For more details see Address Space Definition page.
- Use the test connection button  to check the availability of the specified PLC.

Note: If no controller / device connection is selected, the *Add a new data source* button is the only one available.

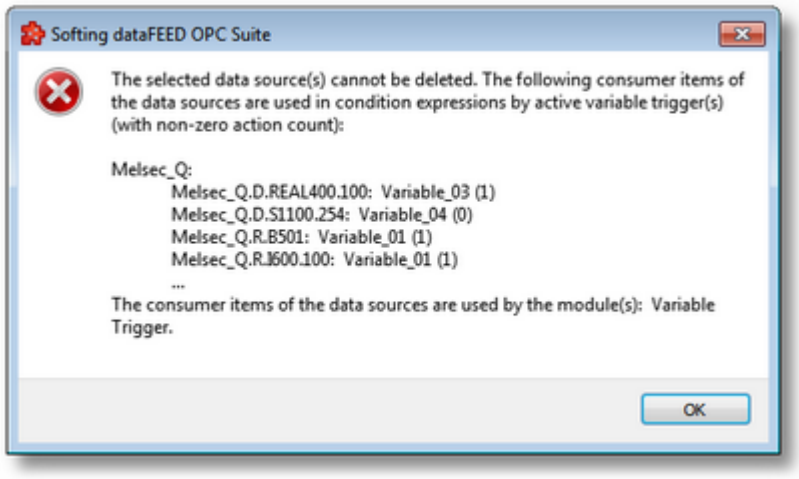
The *Delete* button  will remove the selected device connections from the list.

The user can select more than one server connection with the help of the CTRL or SHIFT keys.

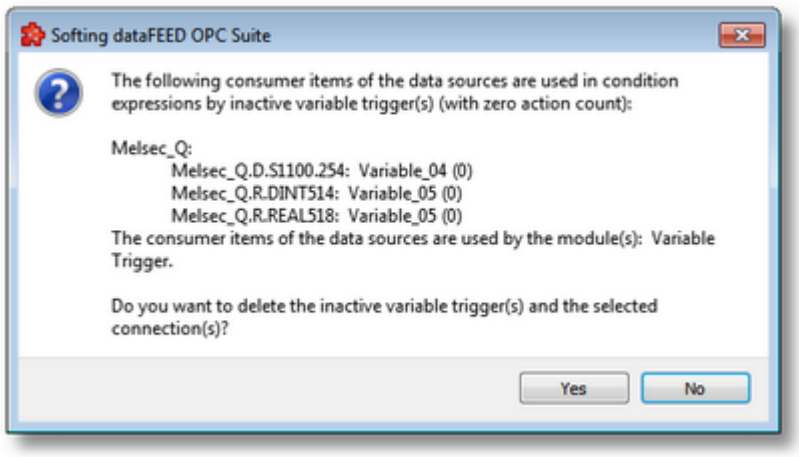
A dialog will then prompt the user to confirm the deletion.



A dialog will inform the user that the data sources cannot be deleted, if consumer items and/or system items of the selected data sources are used by active variable triggers or other modules.



A dialog will prompt the user to confirm deletion of the data sources and inactive variable triggers, if consumer items and/or system items of the selected data sources are used only by inactive variable triggers.

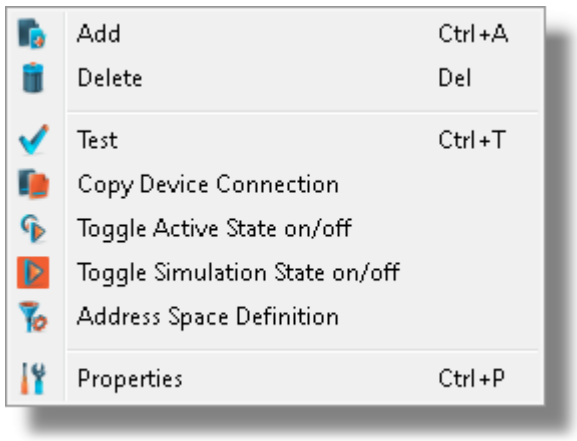


Attention: A device connection cannot be deleted if at least one data item defined in its namespace is used in a Processing or Data Destination action.

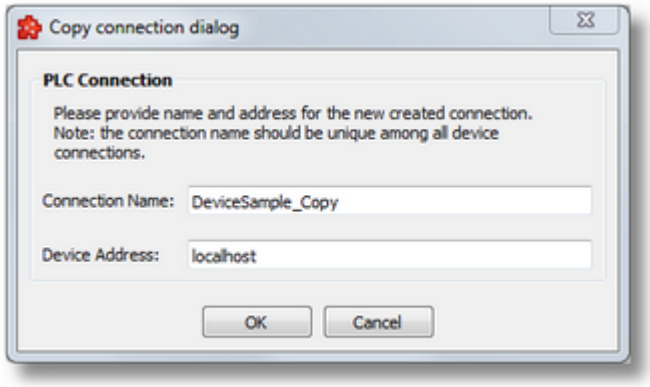
All these actions are also provided in the context menu of the External device list. Which actions are available depends on the current selection in the list.

An example of the context menu can be seen below:

© Softing Industrial Automation GmbH



- Copy device connection  . This action is available only in the context menu and it leads to a dialog where the user may provide a name and device address. See the screen shot below:



The proposed name is the original connection's name post fixed with "_Copy" and the default address will be "localhost".

- Toggle Activate state  . This action will enable / disable a devices connection. A disabled connection will be shown as grayed out in the device connection list and it will be ignored by runtime.

- Toggle Simulation state  . This action will mark a connection as simulation (or remove the simulation mode). Simulation connections are shown in red in main device list.

Note: The connection name should be unique among all device connections names.

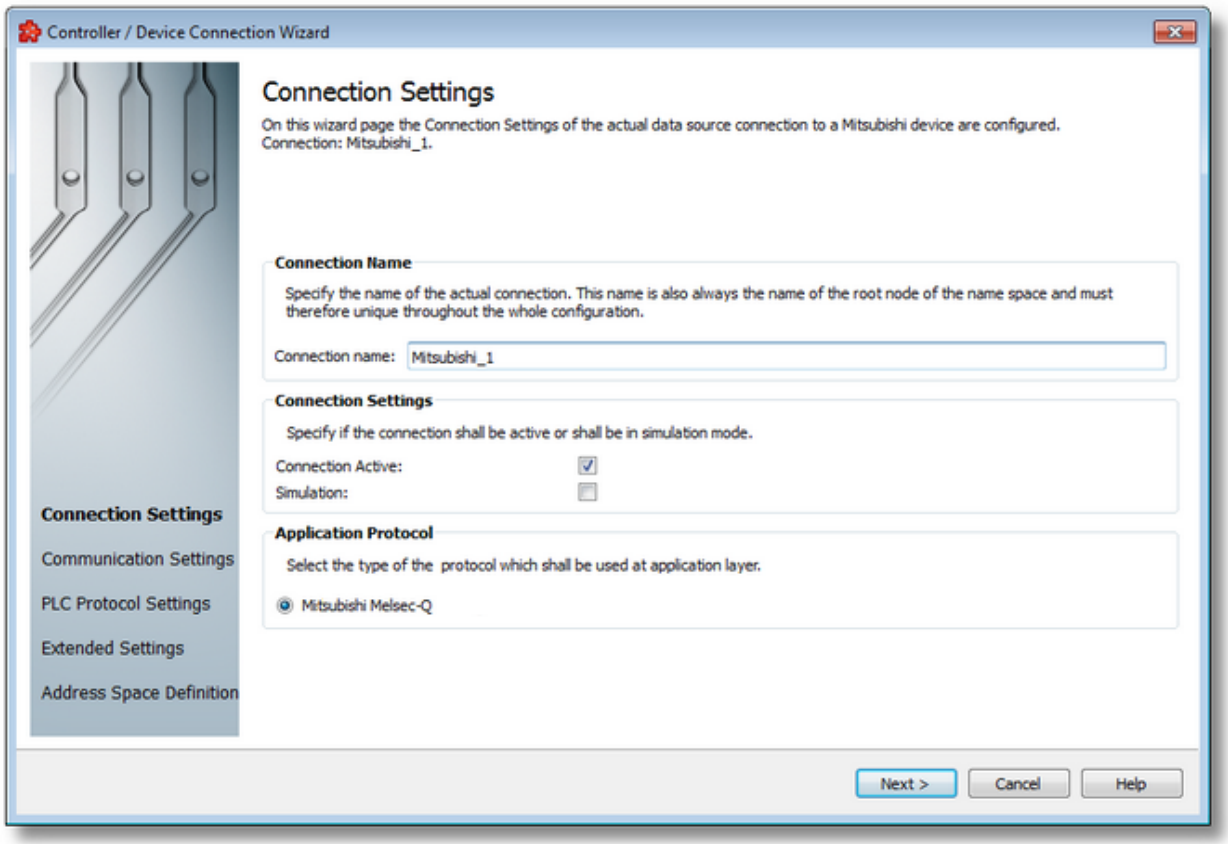
Note: The device address should have a valid host name or an IP address in format IPv4 or IPv6.

7.3.1.6.2 Mitsubishi Device Wizard

With the Mitsubishi Device Wizard you can create and modify a data source connection to an external PLC device:

- Specify connection settings @ Mitsubishi Device Wizard - Connection Settings
- Specify communication parameters @ Mitsubishi Device Wizard - Communication Settings
- Protocol settings @ Mitsubishi Device Wizard - Plc Protocol Settings
- Extended settings @ Mitsubishi Device Wizard - Extended Settings
- Address space definition @ Mitsubishi Device Wizard - Address Space Definition

On this page the connection name that will uniquely identify the data source and some generic connection settings can be provided.



© Softing Industrial Automation GmbH

Connection Name

An unique identifier for the current connection. In case the provided name is already used by other connection, the user use is not allowed to go into the next wizard page.

Connection active

This flag specifies if the current connection will be enabled or not.

Simulation

This flag specifies if the current connection will be simulated by the server or it will represent a real device connection.

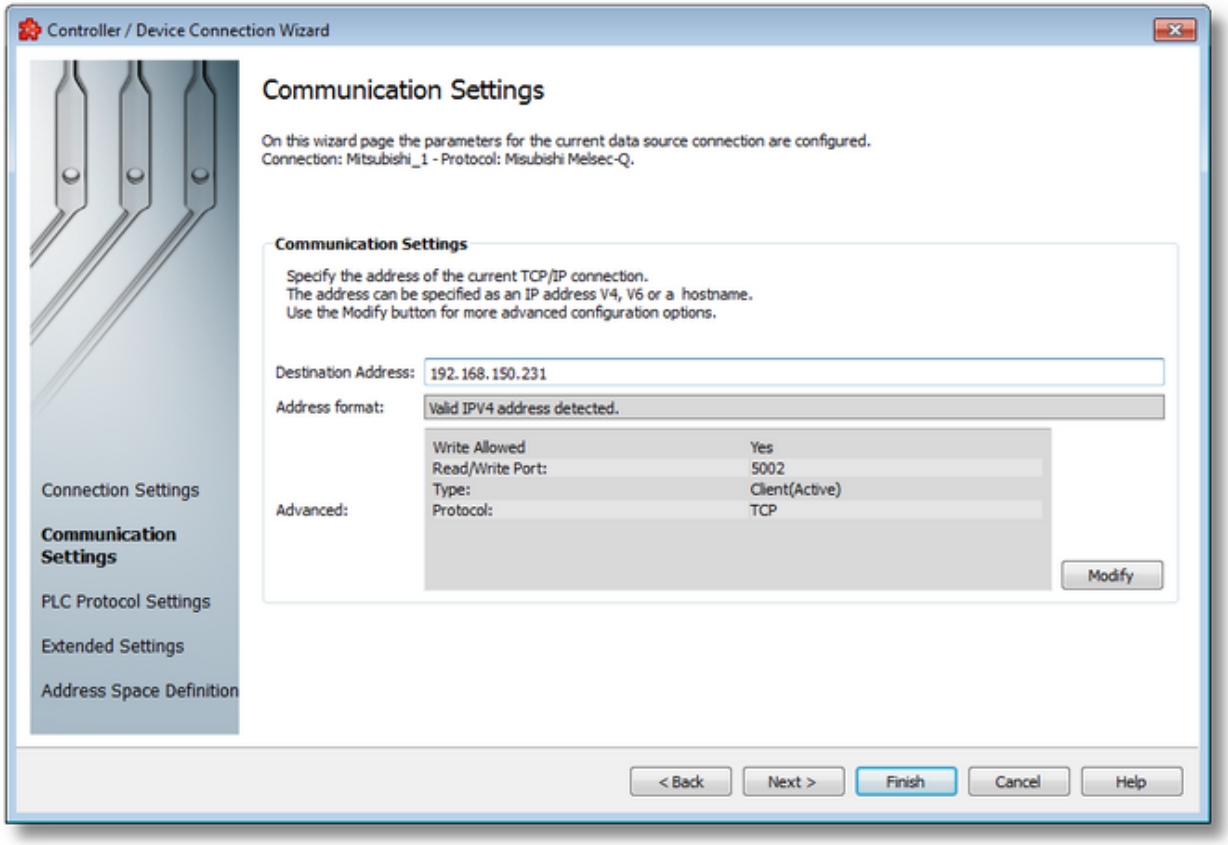
Application protocol

Currently supported protocol is

- Mitsubishi Melsec-Q

Mitsubishi Device Wizard - Communication Settings >>>

On this wizard page various parameters for the current TCP/IP connection are configured.



Destination Address: IP address or name

This is where the IP address of the target station (IP address of the PLC) or the DNS name of the target station is entered. If the computer was configured for DNS and if a DNS server is available on the network, the symbolic name of the target station can also be entered.

For how an IP address is set up, see IP address.

Address Format

Read-Only field; automatically detects the for of destination address.

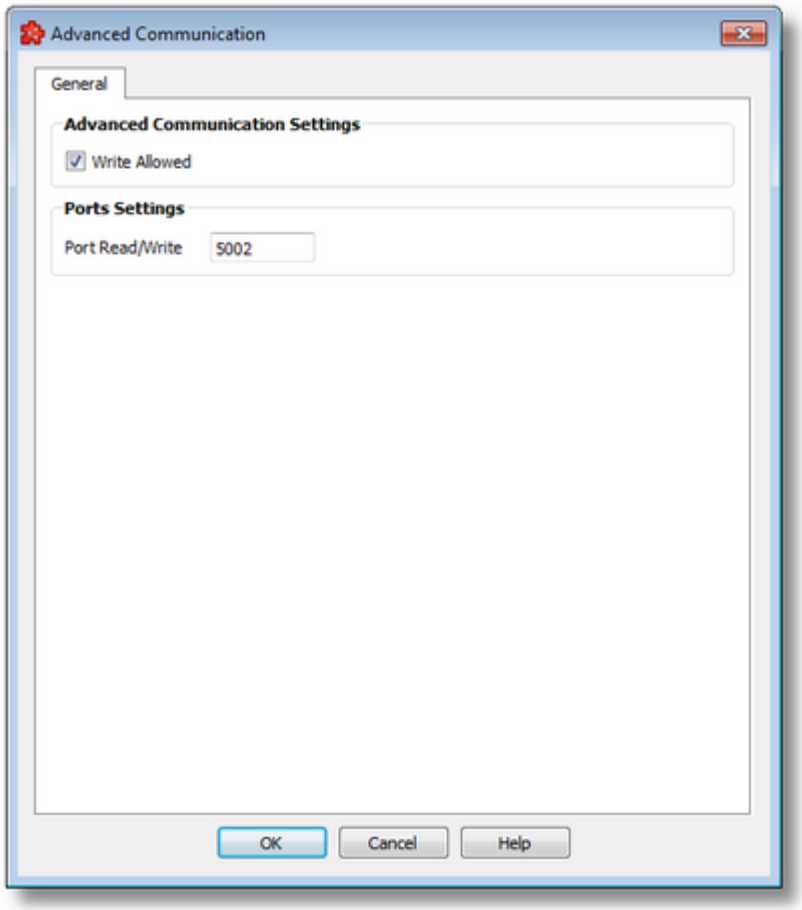
Advanced:

Read-Only field; this is a summary for TCP/IP Communication advanced settings. These settings can be changed using Modify button that lead to Advanced Communication Settings

<<< Mitsubishi Device Wizard - Connection Settings Mitsubishi Device Wizard - Plc Protocol Settings >>>

© Softing Industrial Automation GmbH

In this dialog specific Mitsubishi communication parameters can be adjusted.



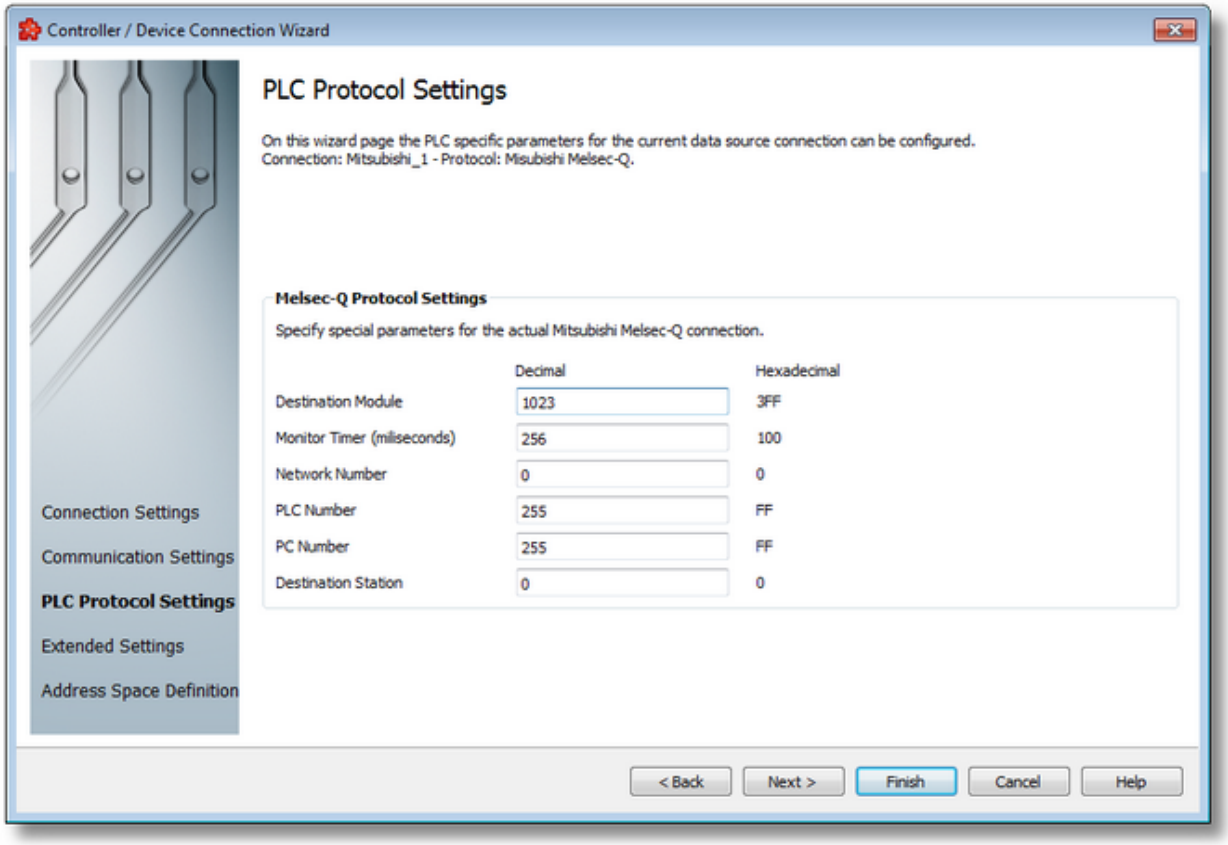
Write to the PLC permitted (Write Allowed)

Here you specify whether write access to the PLC is to be possible via this connection and a write connection is to be established to the hardware. You can always disable write accessing via a connection if you only want to use read accesses or if you want to be sure that no values on the PLC are to be changed. When write accessing the PLC via a Write (Write activeConnection) or a Send/Receive (Send Direct Connection for Send/Receive), certain points of the PLC are written with values (each write access is preceded by a read access).

Port for read accesses, port for write accesses

Port numbers are addresses which are used within the transport layer to address applications. Port numbers are required for TCP connections. The port is parallel to the TSAPs for RFC1006 and H1 connections. The port number is a 16-bit number from 1 to 65535. Keep the following in mind when you set up the connection.

On this page special parameters of a connection to Mitsubishi Melsec-Q controllers can be set.



Destination Module

Network address

Monitor Timer

Time monitoring during which the answer should have been received.

Network number

Network number in the hardware configuration of the programming software.

PLC Number

An additional intermediate PLC number.

PC Number

© Softing Industrial Automation GmbH

Is only applicable when rights management is set up. Access of the PC can be permitted or prohibited with this.

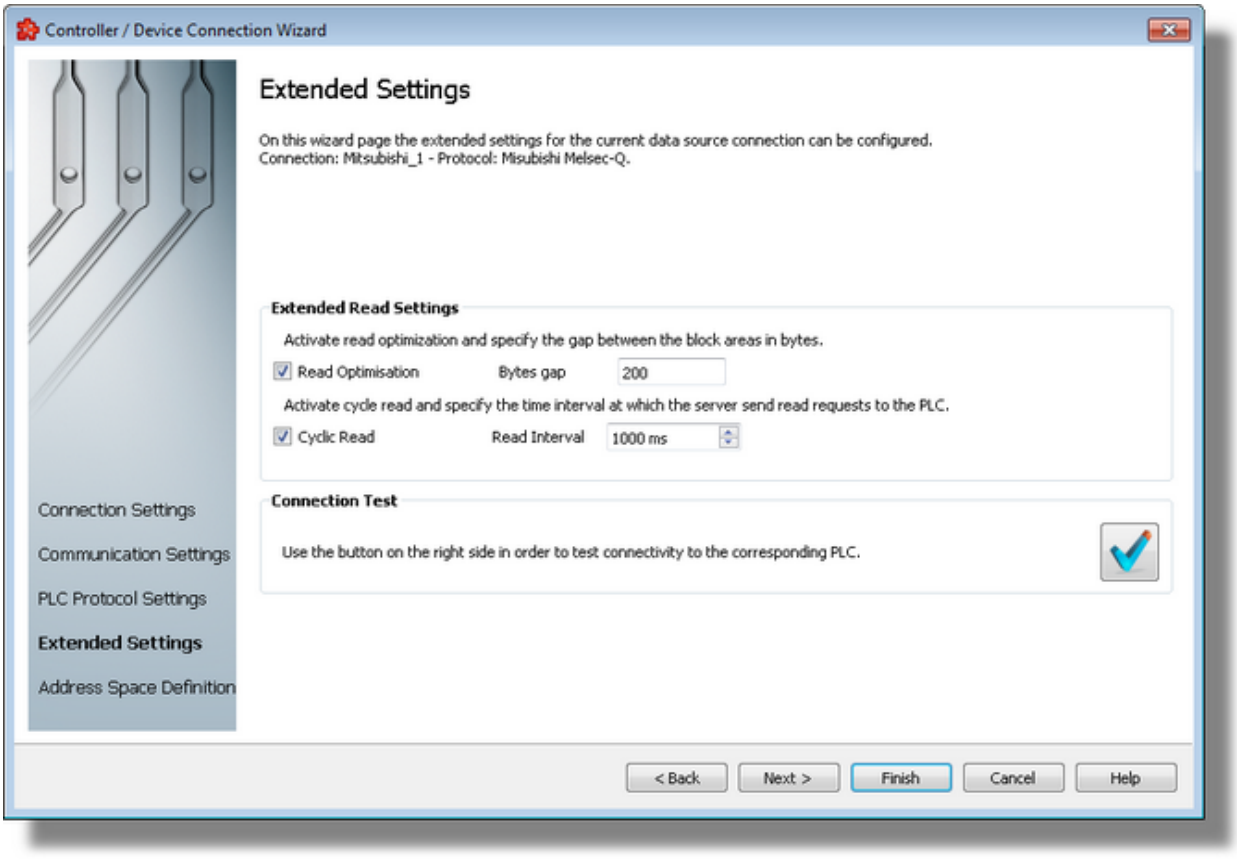
Destination Station

The module address of an additional module in a PLC.

The effects of these values depend on the hardware that is being used.

<<< Mitsubishi Device Wizard - Communication Settings Mitsubishi Device Wizard - Extended Settings >>>

On this wizard page the extended settings for the current data source connection can be configured.



Read optimization

Reading from the controller can be optimized here. If several variables are requested which are located in the same operand area (e.g., in a data block or in the flag area), the complete area is read from the first variable to the last variable without read optimization. Gaps between the individual block areas are also read no matter how large these gaps are. The maximum size of

the gaps (in bytes) between the requested individual items within a data block without a new block being created is entered. If the gap equals the specified value, a new block is created.

Enable cyclic read access and read interval

Cyclic read access is activated as the standard. The server reads the values of the variables from the hardware via a fetch active connection at a certain time interval. It can be specified in Read interval [in ms] the time intervals at which the server is to send a read request to the PLC. The setting “0” means that it should be read as soon as possible. If "Cyclic read" is selected, read access is performed once when the client requests and no longer cyclically. However, this is only necessary in special cases and requires a precise knowledge of the communication.

Continue to Address Space definition

Optionally by checking this flag, the user can define the alias file for the current connection now.

Although this step may be postponed to a later time.

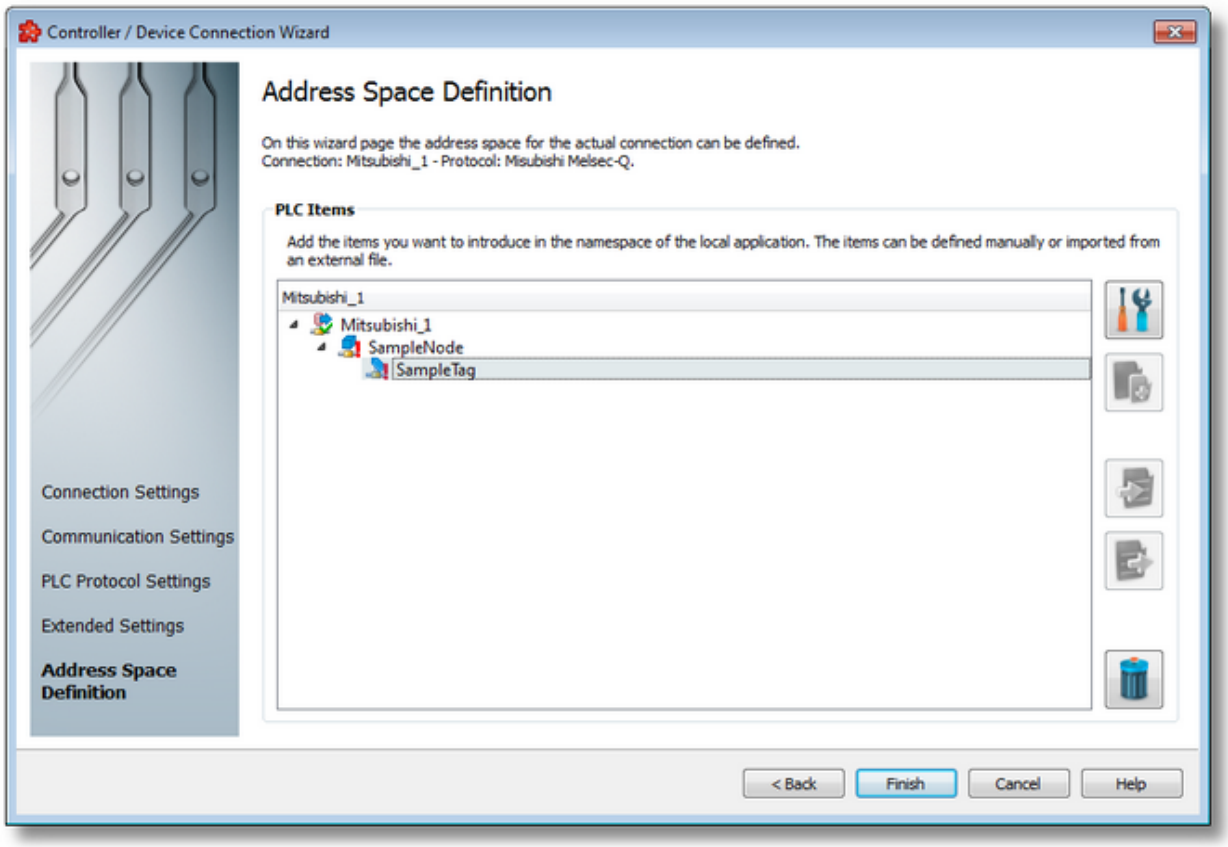
Connection Test

Use the test connection button  to check the availability of the specified PLC.

<<< Mitsubishi Device Wizard - Plc Protocol Settings Mitsubishi Device Wizard - Address Space Definition >>>

On this wizard page the data items for the actual PLC connection can be defined. The address space can be defined by manually adding items, by importing the address space from an external file or by importing a device symbol file (for connections of type S7 and Mitsubishi).

© Softing Industrial Automation GmbH



PLC Connection Items

An editable tree view where the user can change the address space structure or content by adding, deleting or modifying items.

There are 3 categories of items that may be used in the tree:

- **Root node** (having the name of the current connection). This node is read only, the user has just the possibility to add or delete children of this.
- **Nodes:** these elements are associated with branches in tree. They can have children but they are typeless and do not carry any syntax.

Note: The items of type Node represent intermediate nodes in the tree, they are used for hierarchy purposes.

- **Tags:** these elements are associated with leaves in tree.

Note: A tag cannot have any children, so Add is not available for tags.

Operation

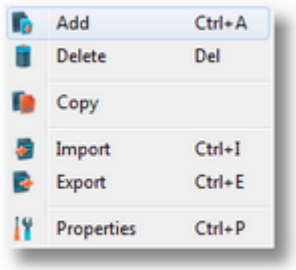
Depending on the current selection in the *Plc Connection Items* tree, the following operations are possible using the buttons on the right, the context sensitive menu or the corresponding shortcuts:

- Properties of the selected item (Ctrl+P) 
- Add a new item (Ctrl+A) 
- Import from file (Ctrl+I) 
- Export to file (Ctrl+E) 
- Delete data items (Del) 

Two more operations are supported, which are only accessible from the contextual menu:

- Copy data item(s) 
- Paste to data item(s) 

An example of the contextual menu available on *Plc Connection Items* tree is shown below:



Delete data items

Triggering the *Delete data items*  action with the corresponding button on the right wizard side or *Delete* from the context menu will remove the currently selected node from the *Data Items* namespace tree.

Attention: The root node of the namespace tree can not be deleted.

Attention: A node which is currently used in a data processing or data destination action can not be deleted.

© Softing Industrial Automation GmbH

Copy

The *Copy* action is provided in the context menu of the *Plc Connection Items* tree. Triggering this action will copy the selected node to the local clipboard. All the properties of the node are copied. Also, if the node contains children, the entire branch is copied.

Note: The root node of the namespace tree does not provide a *Copy* action.

Paste

The *Paste* action is provided in the context menu of the *Plc Connection Items* tree. This action will add the node currently stored in the local clipboard as a a child of the currently selected node. Whether the action is available in the context menu depends on the type of the currently selected node and the type of the node in the clipboard.

Note: The *Paste* action is only available if a node was previously copied to the clipboard.

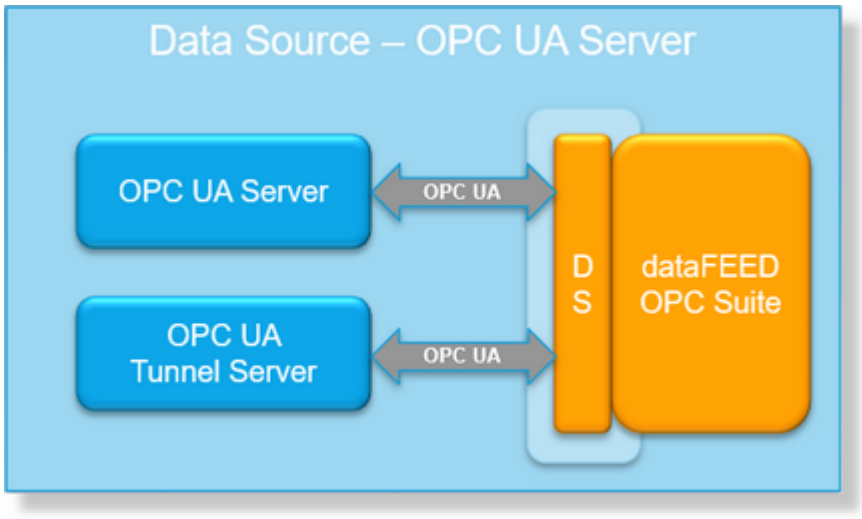
If the *Paste* action is performed on a node which already has a child of the same name as the node in the clipboard, the pasted node is renamed automatically by adding a numerical suffix to the original name.

```
E.g. NodeA, NodeA01, NodeA02, ...
```

<<< Mitsubishi Device Wizard - Extended Settings

7.3.1.7 OPC UA Server

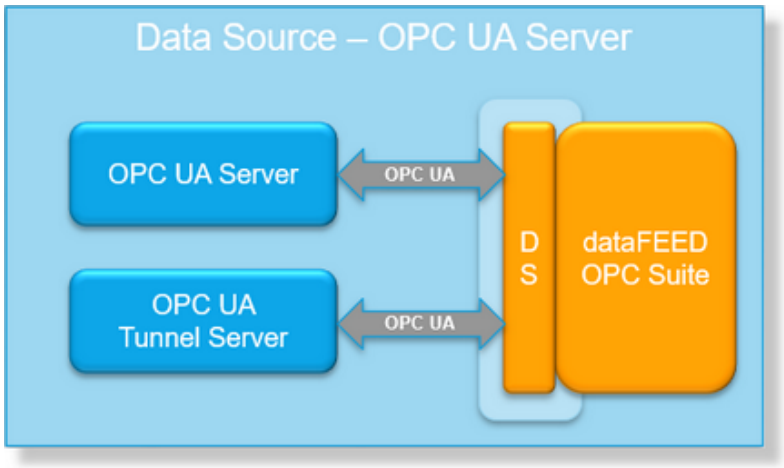
This section of the Softing dataFEED OPC Suite Online Help describes how to configure OPC UA Server or OPC UA Tunnel Server connections within Softing dataFEED OPC Suite.



OPC UA Server data source makes it possible to exchange data between the field devices (PLCs) of a wide variety of manufacturers and OPC UA clients.

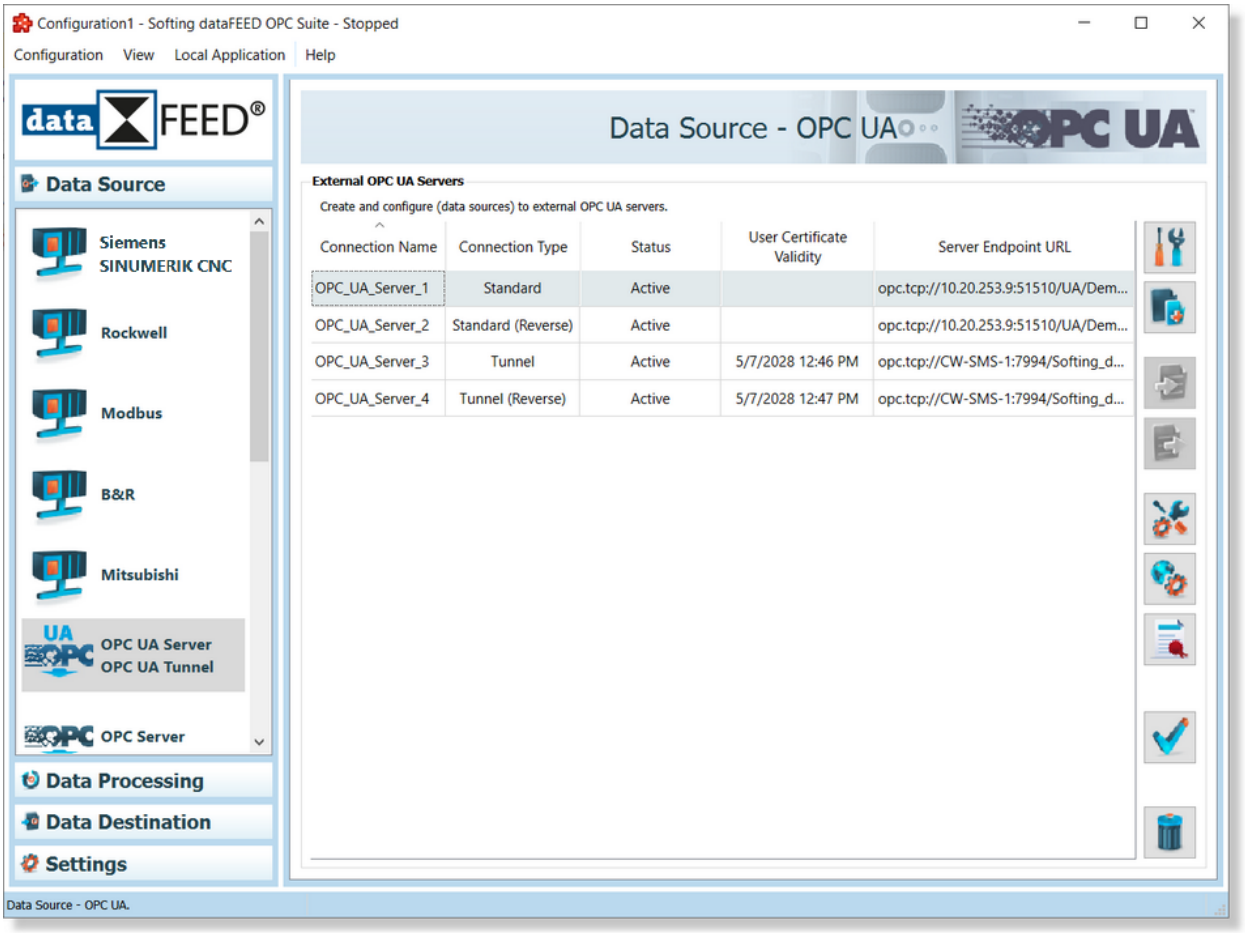
A detailed description on how to configure access to OPC UA Server or OPC UA Tunnel Server connections can be found in the Configuration section.

7.3.1.7.1 Configuration



To start configuring the connection to an external OPC UA Server (or OPC UA Tunnel Server) select OPC UA Server from the Data Source group in the navigator on the left. The Data Source - OPC UA Server main user interface is displayed. The main element of this page is the External OPC UA Server (including OPC UA Tunnel Servers) list. Each entry shown in the list of the configuration section represents a connection to an external UA Server.

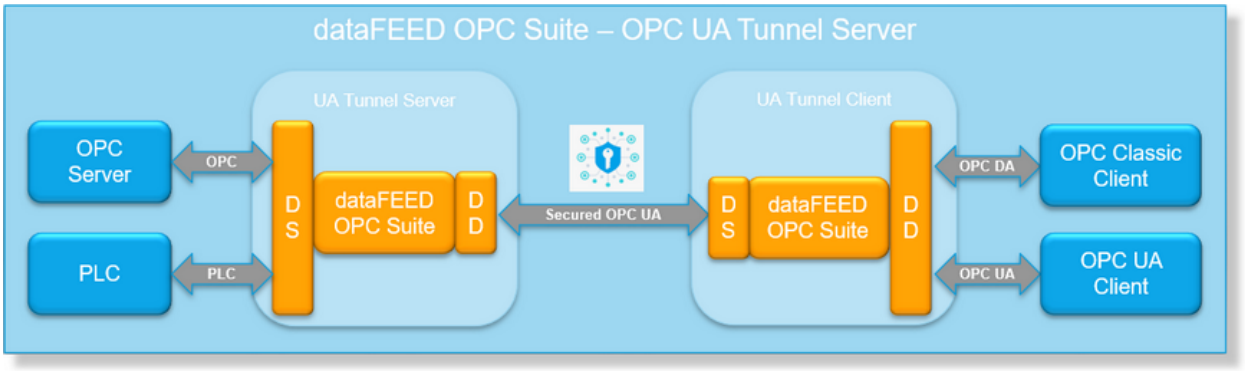
© Softing Industrial Automation GmbH



Note: The list is only shown and editable if a configuration has been loaded.

On Data Source - OPC UA Server / OPC UA Tunnel it is allowed to define connection to OPC UA Server (Standard) or OPC UA Tunnel Server (this is intended to be used for communication between 2 instances of dataFEED OPC Suite based on OPC UA protocol with a high degree of security).

The OPC UA Tunnel solution is using the OPC UA protocol to transport and transfer data between multiple instances of dataFEED OPC Suite. The communication is secured using **Sign and Encrypt** as security mode and **Aes256_sha256_RsaPss** as Security Policy; the user authentication being possible based on **Certificate Authentication** (these options are not configurable being the only one supported).



On the Configurator main user interface, there are visible the configured OPC UA Server (including OPC UA Tunnel) connection - being accessible the following information:

Attribute	Description
Connection Name	An unique name used for distinguish the connection, The name of the device will be shown in the Address Space tree in the Softing dataFEED OPC Suite Configurator and in the local dataFEED OPC Suite application, and its entire Address Space will be under this name. **Note:** Device name must be a unique string within a dataFEED OPC Suite configuration. In addition the name could not contain address space delimiters, multi-byte characters or empty spaces.
Connection Type	Specifies the type of the connection - the following options being available: • **Standard**: Forward connection to an external OPC UA Server based on OPC UA protocol). • **Standard (Reverse)**: Reverse connection to an external OPC UA Server based on OPC UA protocol. • **Tunnel**: Forward connection to a Softing dataFEED OPC Suite Tunnel Server based on secured OPC UA protocol. • **Tunnel (Reverse)**: Reverse connection to a Softing dataFEED OPC Suite Tunnel Server based on secured OPC UA protocol.
Status	Specifies the status of the connection - the following options being available: • **Active**: The connection to the OPC Server (OPC UA Standard or Tunnel) is completely defined and the connection is active (the dataFEED OPC Suite Runtime will try to connect to the server at initialization time).

© Softing Industrial Automation GmbH

	• **Not Active**: The connection to the (OPC Server Standard or Tunnel) is completely defined and the connection is not active (the dataFEED OPC Suite Runtime will ignore the server at initialization time). • **Server Input Missing**: The connection is not completely defined - information about the server is not complete or the trust relation with the server is not completed (the dataFEED OPC Suite Runtime will ignore the server at initialization time). **Note: Server Input Missing** option is available only for not completely defined OPC UA Tunnel connection.
User Certificate Validity	Validity of certificate used for User Authentication (if authentication based on user certificate is configured). For other types of authentication (Anonymous or credentials based user authentication) - the information is missing.
Server Endpoint URL	Specifies then Endpoint URL of the configured OPC UA Server (including Tunnel).

Operation allowed on UA Server Connections (including OPC UA Tunnel):

From the OPC UA Server main user interface certain operations can be performed over the selected connection by using the buttons on the right, by accessing the context sensitive menu or by pressing the keyboard shortcuts. The following actions are available:

- Change the properties of the selected data source by using the *Properties* button  , pressing the keyboard shortcut CTRL+P or selecting *Properties* from the context menu.
- Add a new connection by using the *Add* button  , pressing the keyboard shortcut CTRL+A or selecting *Add* from the context menu.
- Import OPC UA Tunnel Connection settings by using the *Import* button  , pressing the keyboard shortcut CTRL+I or selecting *Import* from the context menu (this option is available exclusively for OPC UA Tunnel connections).
- Export OPC UA Tunnel Connection settings by using the *Export* button  , pressing the keyboard shortcut CTRL+E or selecting the *Export* from the context menu (this option is available exclusively for OPC UA Tunnel connections),
- Change advanced settings for the selected data source by using the *Advanced settings* button  .

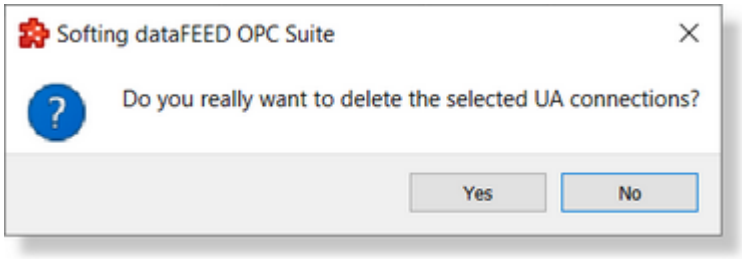
- Change global settings for the selected data source by using the *Global settings* button  .
- Manage the application PKI Store - Certificates handling by using the *Manage application PKI Store button*  .
- Use test connection button  to check the availability of specified server or the correctness of the authentication and security settings.
- Delete the selected data source by using the *Delete* button  , pressing the DEL / DELETE key on the keyboard or selecting *Delete* from the context menu.

Note: If no connection is selected, the *Add a new data source* button and Global settings are the only options enabled.

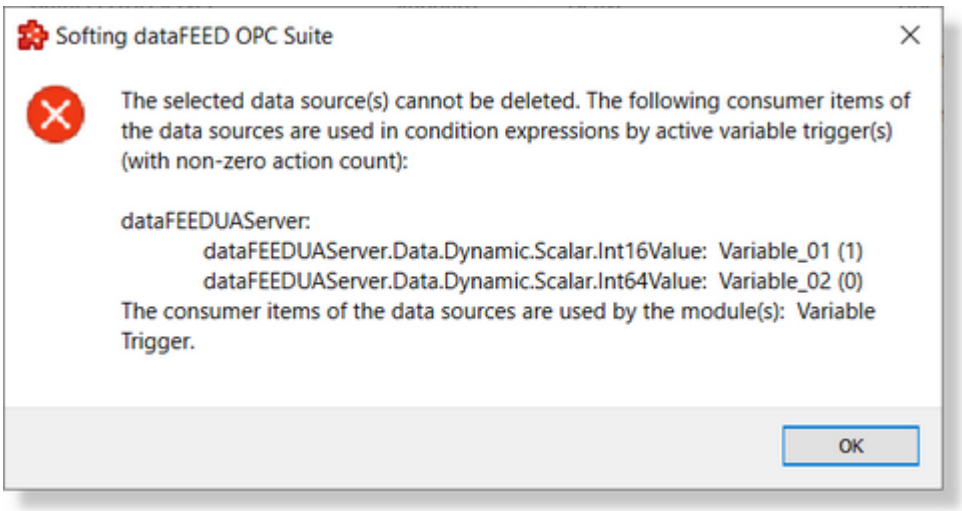
The *Delete* button  will remove the selected device connections from the list.

The user can select more than one server connection with the help of the CTRL or SHIFT keys.

A dialog will then prompt the user to confirm the deletion.

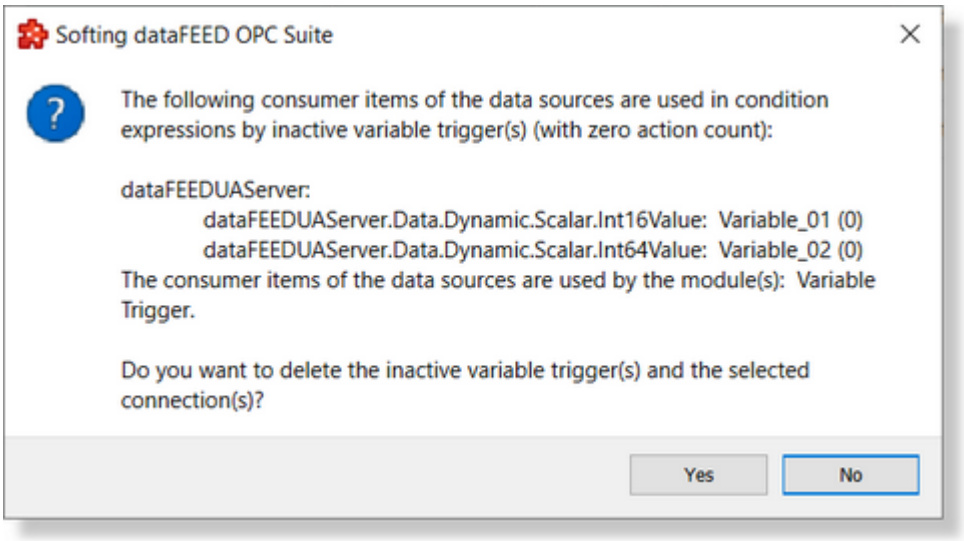


A dialog will inform the user that the data sources cannot be deleted, if consumer items and/or system items of the selected data sources are used by active variable triggers or other modules.



© Softing Industrial Automation GmbH

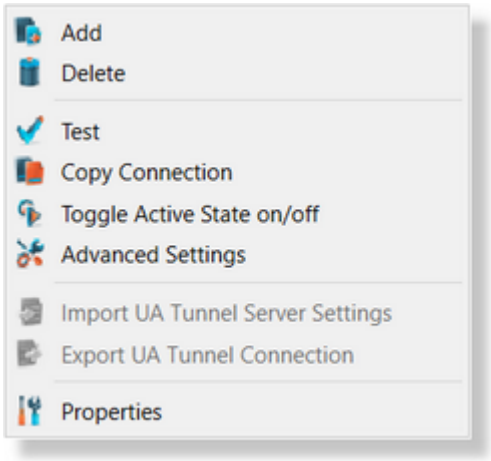
A dialog will prompt the user to confirm deletion of the data sources and inactive variable triggers, if consumer items and/or system items of the selected data sources are used only by inactive variable triggers.



Attention: A device connection cannot be deleted if at least one data item defined in its namespace is used in a Processing or Data Destination action.

These actions are also provided in the context menu of the External device list. Which actions are available depends on the current selection in the list.

An example of the context menu can be seen below:

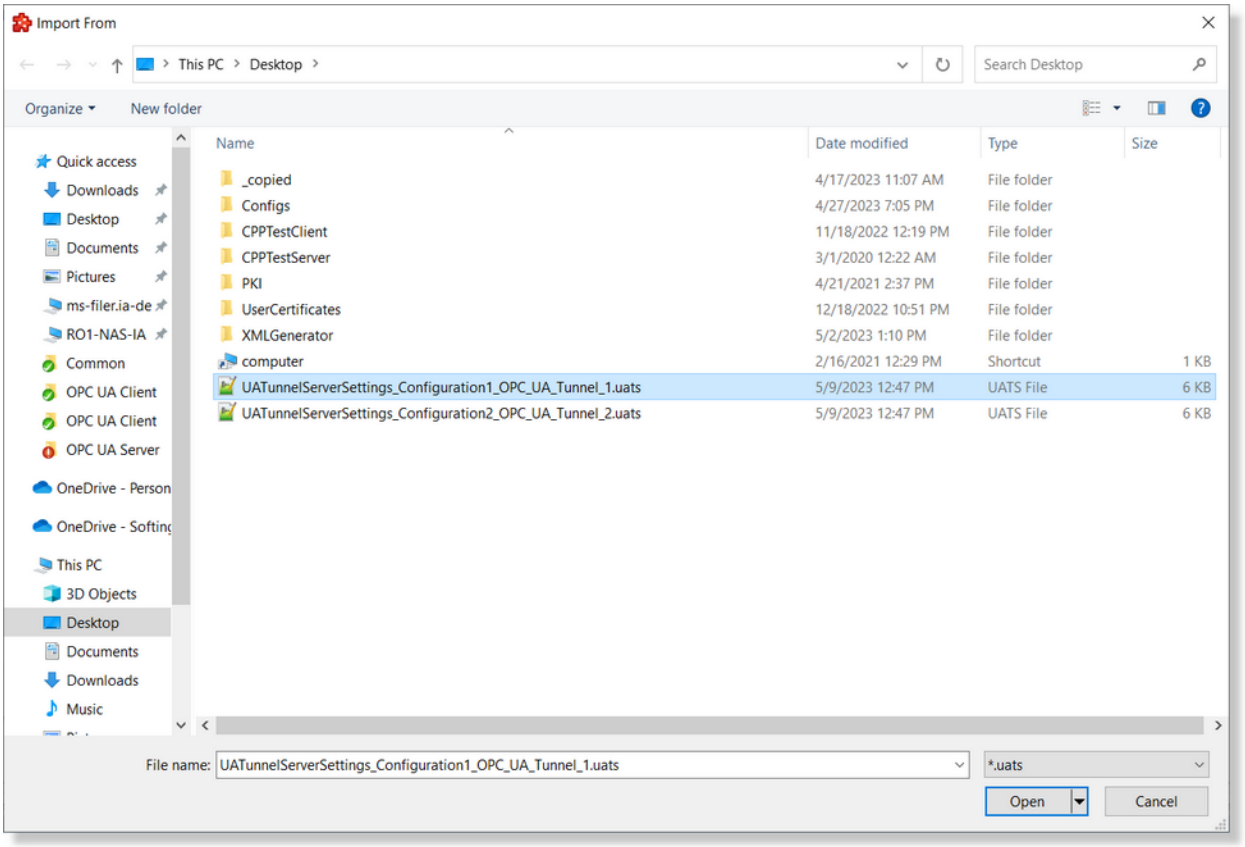


Note: Importing UA Tunnel Server settings and Exporting UA Tunnel Connection options are available only for a selected UA Tunnel connection, This option is not available for Standard OPC UA Server connections.

- Import UA Tunnel Settings  .

The OPC UA Tunnel connections are using as OPC UA as communication protocol, secured by **Sign&Encrypt** with security policy **Aes256_sha256_RsaPss** and authorized by authentication based on **User Certificate**. To complete the connection definition is mandatory to exchange settings and certificates between OPC UA Tunnel client and server (this operation should be executed on both sides).

Importing OPC UA Tunnel server setting is possible by importing the *.uatc file (generated by the data destination OPC UA - Tunnel Endpoint). This file is open, parsed, imported and overrides the connection settings of current connection.



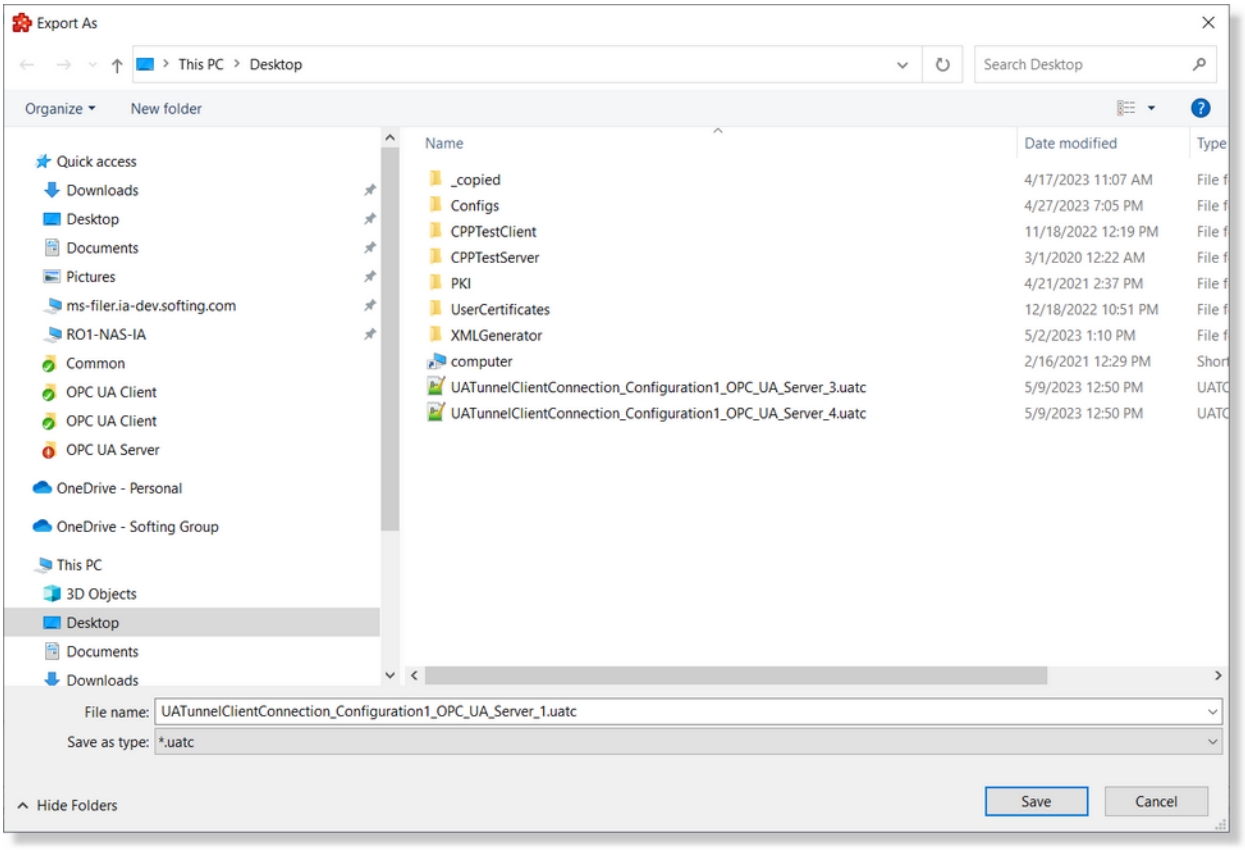
The imported file contains information about connection settings and also the Server Application certificates and the User Authentication certificate (those files are copied automatically on proper PKI location and linked to actual connection).

Note: The previous certificates (Server Application and User Authentication certificates) are still stored until **save** operation is performed.

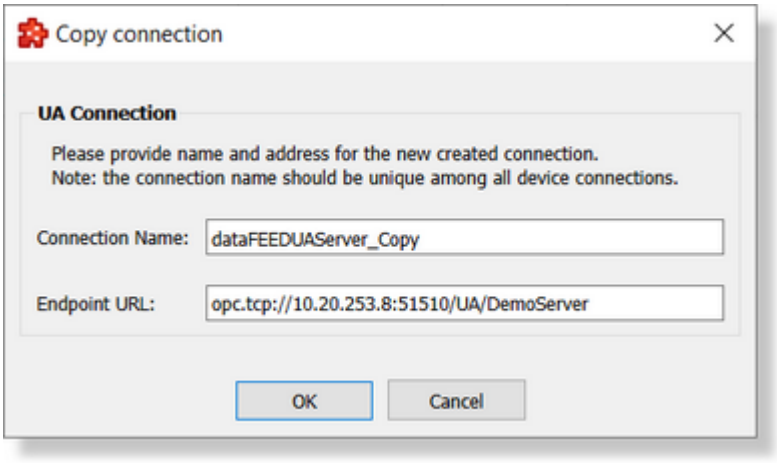
- Export UA Tunnel Connection 

The configured OPC UA Tunnel connections can be exported having as purpose to be imported on data destination OPC UA - Tunnel Endpoint. This operation is mandatory in order to establish the trust relation between OPC UA Tunnel sever and client. This operation is needed in order to have the possibility to export the connection settings and Client Application Certificate.

© Softing Industrial Automation GmbH



- Copy device connection  . This action is available only in the context menu and it leads to a dialog where the user may provide a name and device address. See the screen shot below:



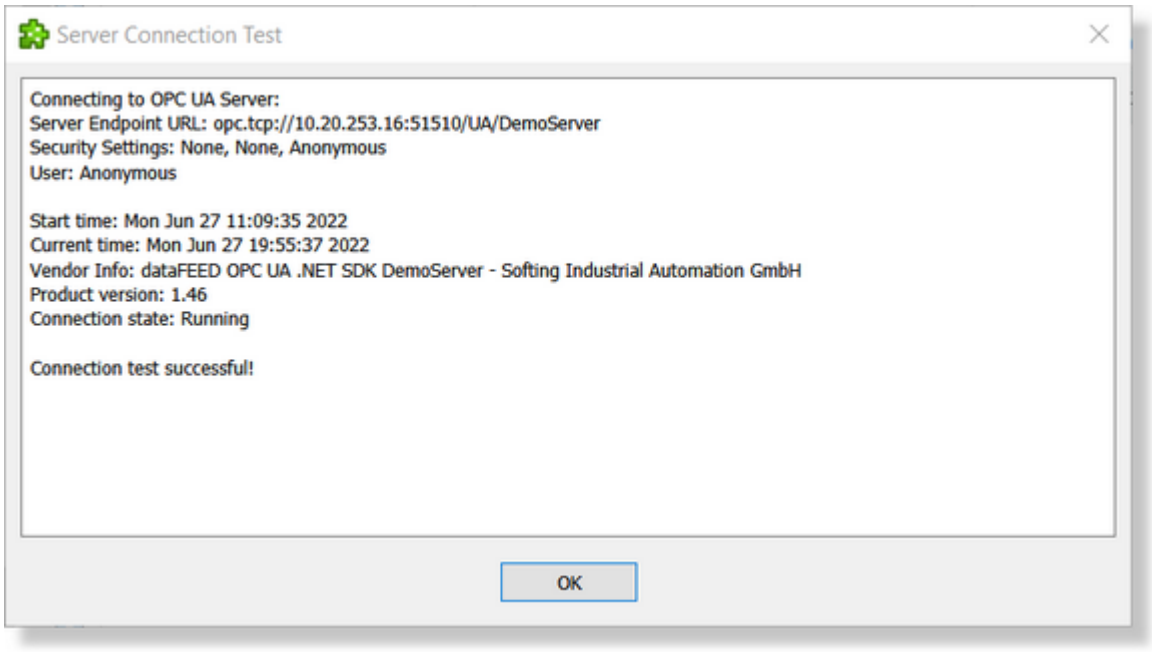
The proposed name is the original connection's name post fixed with "_Copy". The endpoint URL is copied from the source connection.

- Toggle Activate state  . This action will enable / disable a device's connection. A disabled connection will be shown as grayed out in the device connection list and it will be ignored by runtime.

Note: The connection name should be unique among all device connections names.

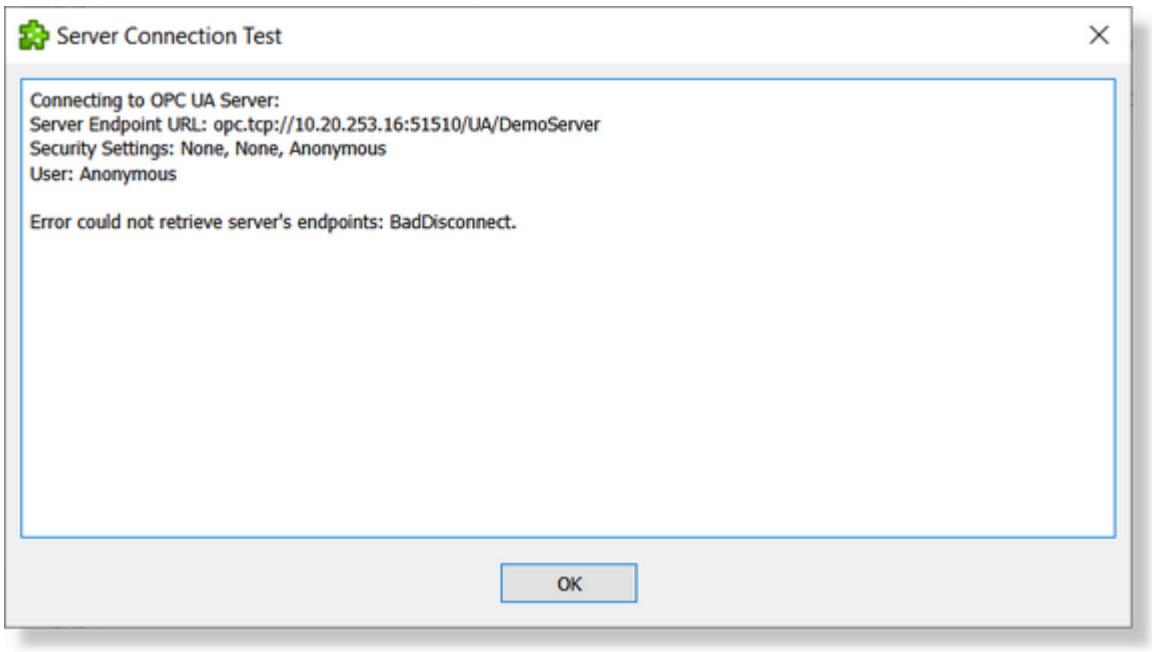
7.3.1.7.2 Server Connection Test

The *Test Connection* button  checks the OPC UA Server connection. The dataFEED OPC Suite Configurator tries to connect to the external OPC UA server. A dialog is shown with the test results. If the connection was successful, information about the server is also displayed:



If no connection could be established, the following dialog box is displayed:

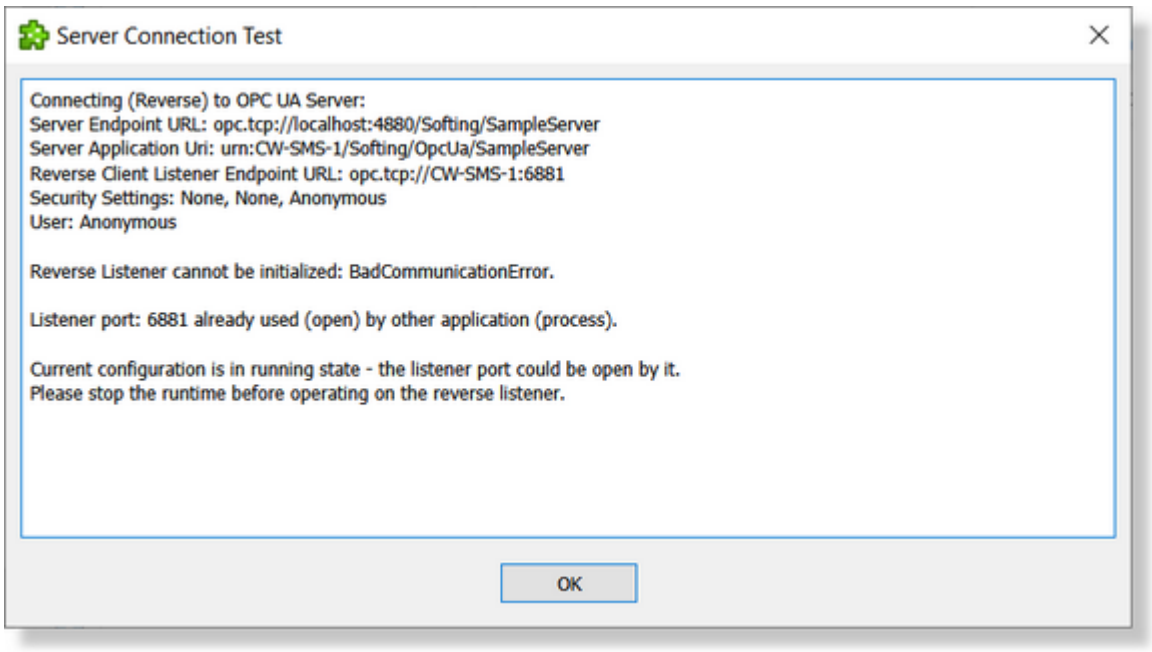
© Softing Industrial Automation GmbH



Note: The test connection dialog cannot be canceled. In some cases it may take up to Network Timeout parameter (default 60 seconds).

In case of a reverse connection used, the connection tests is realized based on starting the Reverse Listener and waiting for server to connect. In case when the server is not connected in time (Network Timeout interval) the connection tests is failing.

Attention: For reverse connections, it is mandatory to be able to start the Reverse Listener. This operation requires to open the port of specified on Client Endpoint Url on the specified network interface - in case when this port is open by other application or other dataFEED Configuration or Runtime of the current Configuration then the test connection will fail. Please ensure the corresponding port can be open by the Reverse Listener.

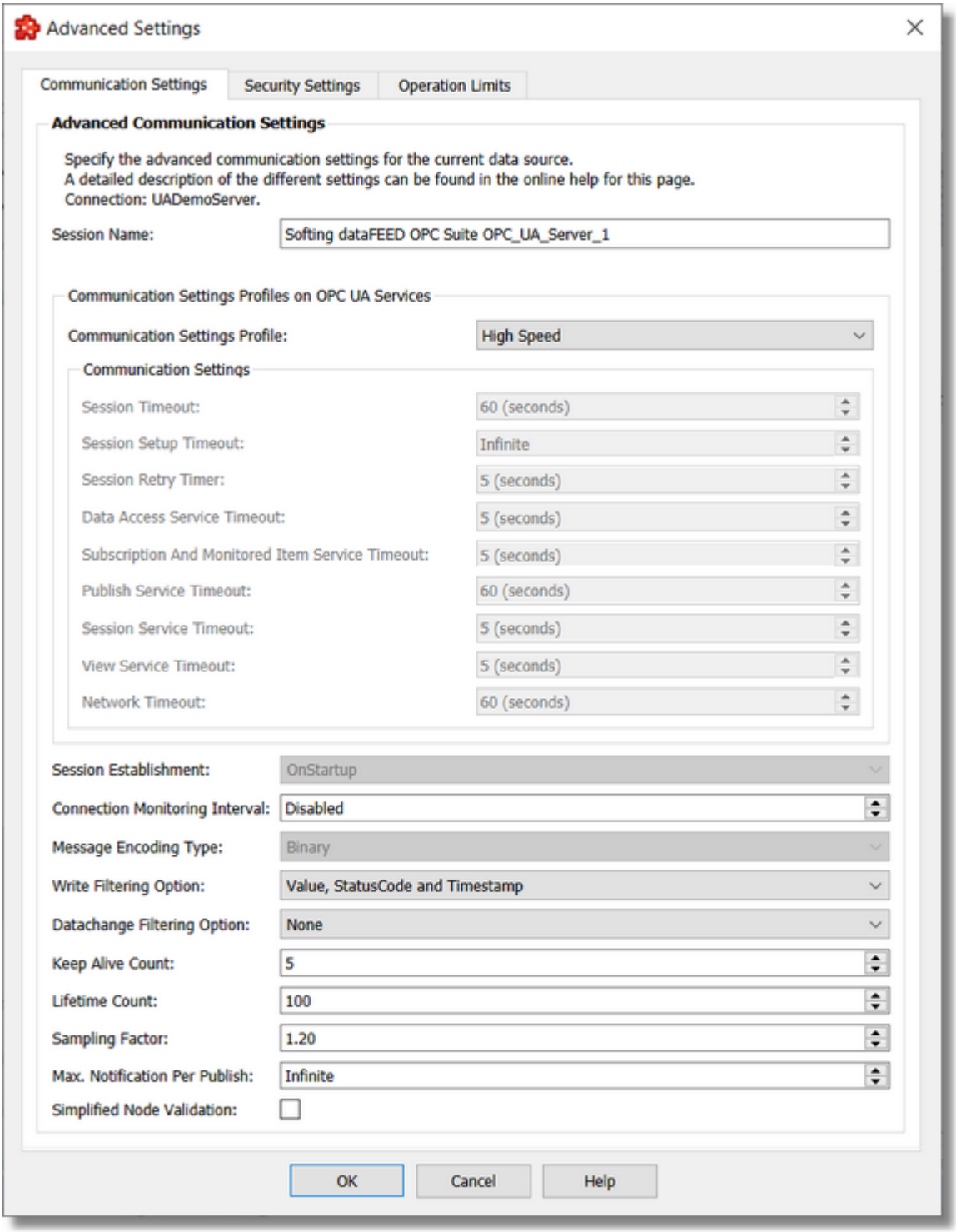


7.3.1.7.3 Advanced Settings

Advanced settings dialog contain a set of parameters needed in order to create a connection to an external OPC UA server. All parameters are connection specific - individually defined for each configured connection (a set of default values being applied in case that the values are not defined).

Advanced Communication Settings:

© Softing Industrial Automation GmbH



Parameter Name	Description	Default value and value range (when applicable).
Session Name	Name that identifies the OPC-UA client session associated with the OPC-UA Server source.	The default value for session name is auto-generated based on the following pattern: [Softing

		dataFEED Suite <ConfigurationName>]
Session Timeout	This parameter specifies the maximum of time which will the connection be active without any activity.	Default value: 60 seconds. Range: 1 to 3600 seconds. In addition it is possible to deactivate the timer by setting value to 0 (Disabled).
Session Retry Timer	This parameter specifies the time after which a re-establishment is performed.	Default value: 5 seconds. Range: 1 to 3600 seconds.
Session Setup Timeout	This parameter specifies the maximum amount of time allowed for application to connect to the external server.	Default value: 30 seconds. Range: 1 to 60 seconds. In addition it is possible to deactivate the timer by setting value to 0 (Infinite).
Session Establishment	Specifies the connection modality: - OnStartup: when the application starts - OnDemand: on the first service call that involves a target server	OnStartup is currently supported only.
Connection Monitoring Interval	Specifies the interval in milliseconds for verifying the health of the connection to the server.	Default value: 0 (disabled). Range: 100 to max(UInt32)
Network Timeout	Specifies the time-out in milliseconds for establishing a secure channel.	Default value: 60 seconds. Range: UInt32.
View Service Timeout	Specifies the time-out in milliseconds for view services.	Default value: 5 seconds. Range: UInt32.
Data Access Service Timeout	Specifies the time-out in milliseconds for data access services.	Default value: 5 seconds. Range: UInt32.
Subscription And Monitored	Specifies the time-out in milliseconds for subscription and monitored item services.	Default value: 5 seconds. Range: UInt32.

© Softing Industrial Automation GmbH

Item Service Timeout		
Publish Service Timeout	Specifies the time-out in milliseconds for the publish service.	Default value: 60 seconds. Range: UInt32.
Session Service Timeout	Specifies the time-out in milliseconds for the session services.	Default value: 5 seconds. Range: UInt32.
Message Encoding Type	Specify how the content of the message is encoded. The sender of the message will encode the data in this format and the receiver will decode the information in order to recreate the original message. OPC UA supports 2 message formats: - Binary – Encodes the date into a serialized byte array. It is designed for fast encoding and decoding. - XML – Uses XML encoded messages. It is slower, but the messages can be easily interpreted. This can be used only for HTTP or HTTPs protocols.	Binary is currently supported only.
Write Filtering Option	Specifies the allowed attributes being changed on a write operation. Basically, on write operations is is allowed to change the following attributes: value, status code and source timestamp. A set of OPC UA servers are not accepting writing all those attributes, rejecting the write operation. With this setting, the user can specify which of the attributes are sent required being written, all the other ones being ignored.	Default value: Value, Status Code and Timestamp Possible values: • Value, Status Code and Timestamp • Value and Status Code • Value and Timestamp • Only value.
Datachange Filtering Option	Specifies the conditions under which a data change notification should be reported. The following attributes could generates datachange notifications: value, status code and source timestamp.	Default value: Value, Status Code and Timestamp.

	The allowed possibilities are: • Value, Status Code and Timestamp - Report a notification if either StatusCode, Value or the Source Timestamp change. • Value and Status code - Report a notification if either the Status Code or the value change. • Status Code - Report a notification ONLY if the Status Code associated with the value changes.	
Keep Alive Count	Sets the threshold for the Subscription's keep-alive counter. Whenever the keep-alive counter reaches the provided threshold, the server publishes a keep-alive notification message for this Subscription. The server uses a keep-alive counter to inform the client that the Subscription is still alive and to provoke activity from the client in order to prevent the Subscription from being deleted. The keep-alive counter is incremented by one whenever the publishing interval expires and is reset to zero when the server publishes any notification message (including the keep-alive message) for the Subscription.	The default value for this attribute is 5 seconds.
Lifetime Count	Sets the threshold for the Subscription's life timer counter. Whenever the life time counter reaches the provided threshold, the server deletes the Subscription. The server uses a life time counter to keep track of client activity for the Subscription. The life time counter of a Subscription is incremented by one whenever the publishing interval expires and it is reset to zero on every Subscription-related client activity (e.g. when the server receives a publish request). When the life time counter reaches the threshold provided by this	Default value for this attribute is 100 cycles.

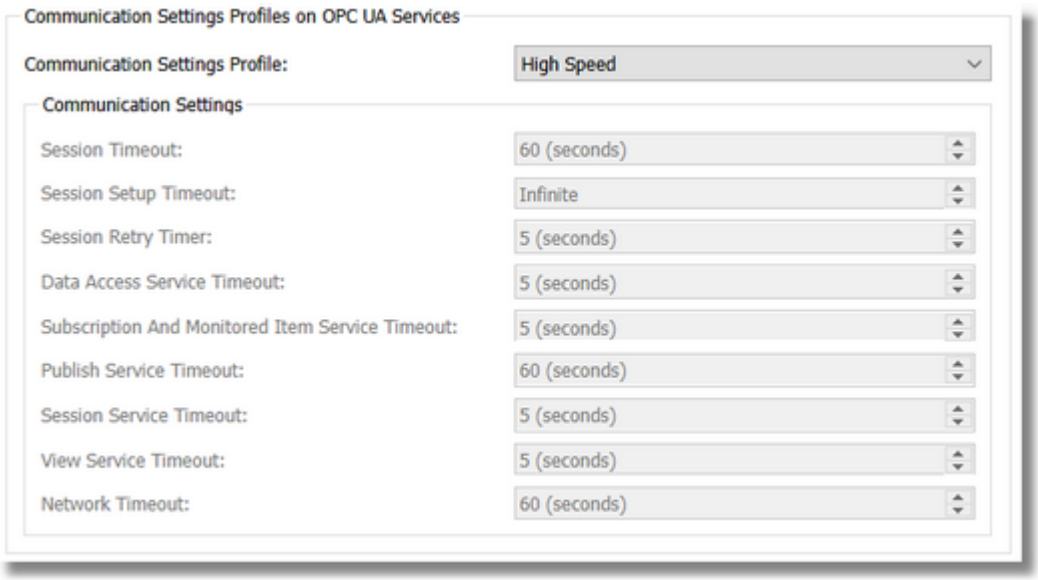
© Softing Industrial Automation GmbH

	method the server deletes the Subscription. In order to ensure regular client activity in case the Subscription has no notifications to publish, the server publishes keep-alive messages when a given number of publishing intervals have elapsed without notifications being available at the server. Therefore, the life time count depends on the maximum keep-alive count of the Subscription. The life time count should always be set to at least three times the maximum keep-alive count.	
Sampling Factor	Specifies the ration between the publish interval on connected UA Client and corresponding UA Server. Basically the subscription to the connected OPC UA Client is not the same with the required publish interval by the connected client - being adjusted with this ratio.	The default value for this attribute is: "1.2".
Max. Notification per Publish	Sets the maximum number of notifications that a client wishes to receive per publish response. If the client does not want to impose restrictions on the message size of the publish response, this parameter should be set to 0 (infinite).	Default value for this attribute is "0" (infinite).
Simplified Node Validation	Disables the browse operation over validate node calls. A validate node operation is reading a set of attributes and determine the node type based on a browse operation. This attribute can disable the browse operation (part of validate node) and always specifying the node type as variable.	Default value for this attribute is disabled.

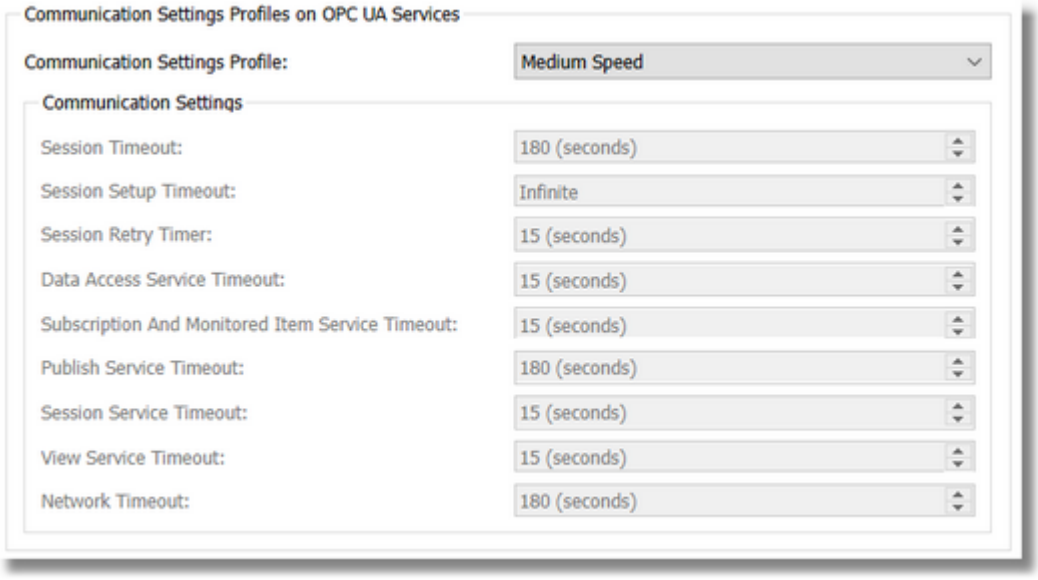
The communication settings are grouped into a set of communication settings profiles. Based on selected communication profile, the communication settings profiles are automatically adjusted with the proper values.

The following communication profiles are available:

- **High Speed:**

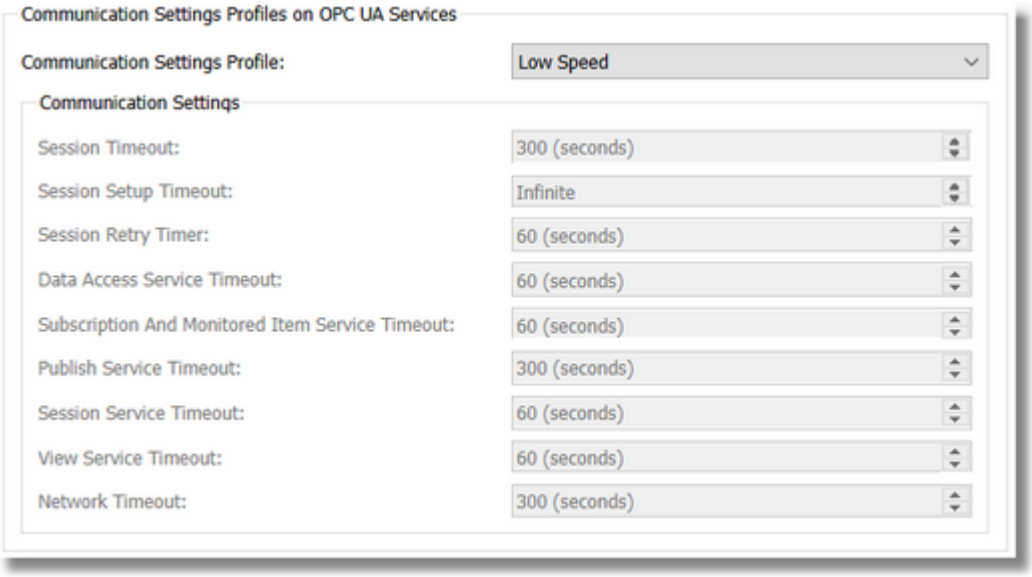


- **Medium Speed:**

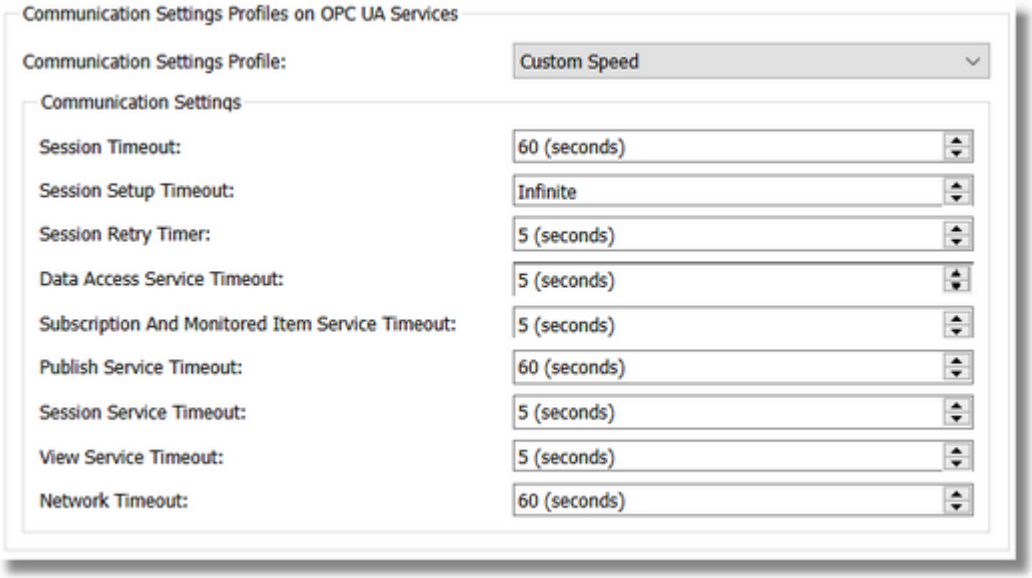


- **Low Speed:**

© Softing Industrial Automation GmbH



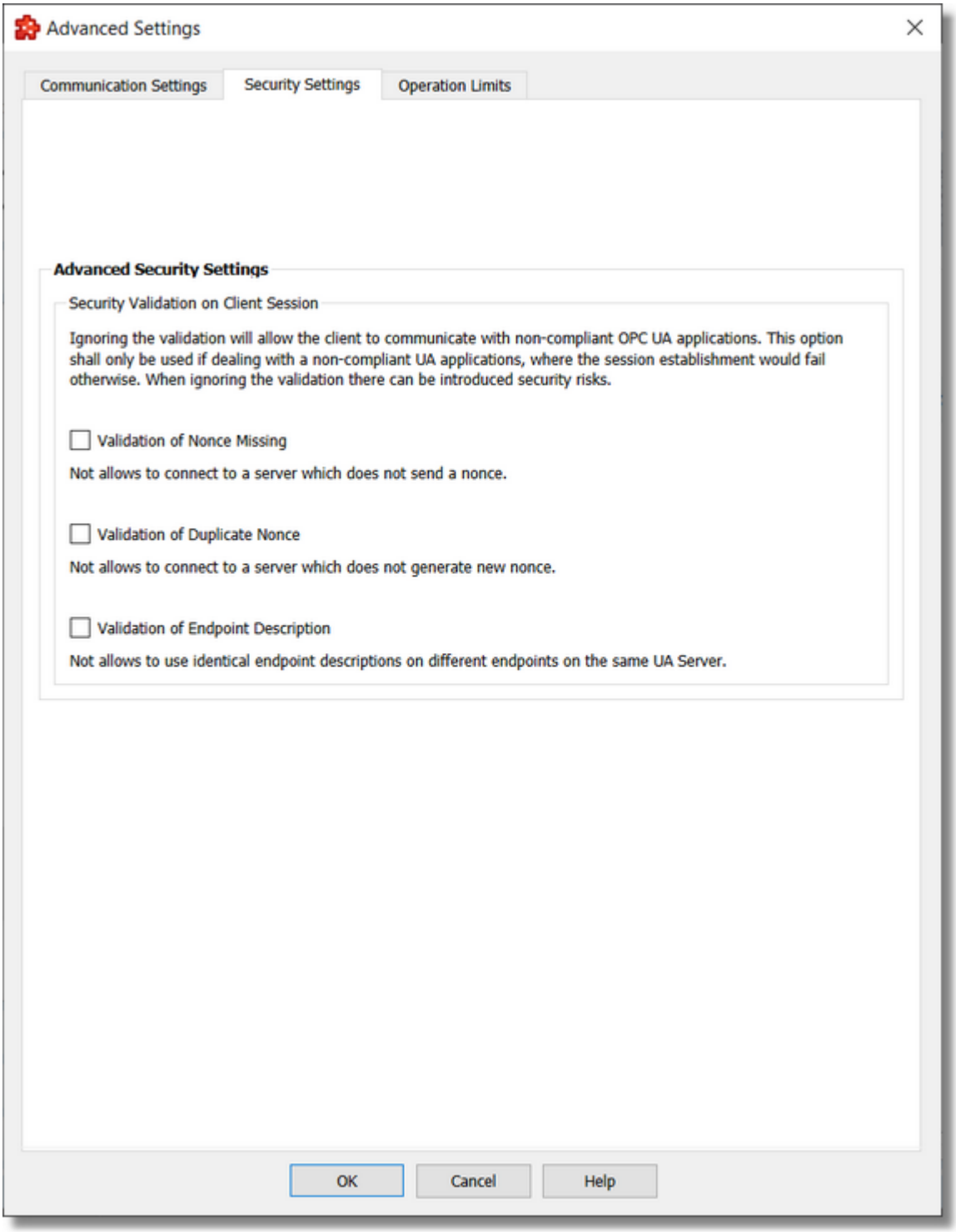
- **Custom Defined:**



On this profile, it is allowed to configure the communication settings parameters and to defined a custom specific ones.

Attention: It is very important to be careful on selecting the proper values of communication settings. These are affecting the communication with the corresponding OPC UA Server as data source.

Advanced Security Settings:



Parameter Name	Description
Validation of Nonce Missing	Not allows to connect to a server which does not send a nonce.

© Softing Industrial Automation GmbH

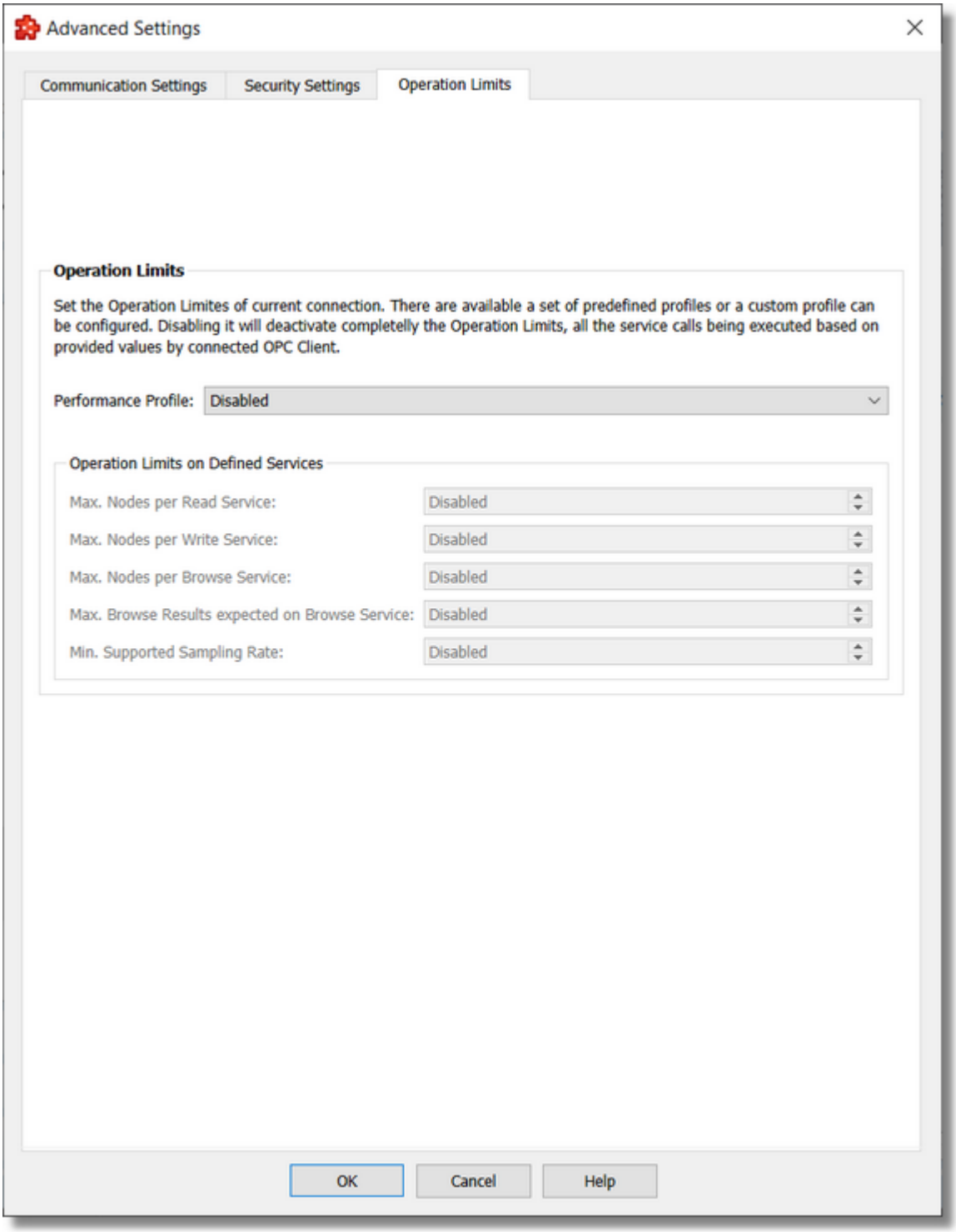
	Client session specific. Client sessions will use the application setting by default. If only specific sessions shall disable this validation, disable it at the specific session. This nonce is a random number (at least 32 byte), which has to be generated on every CreateSession and ActivateSession response by the server. The nonce is part of the encrypted part for user validation and validated at server side. When allowing empty or repeated nonce, then the messages are vulnerable for replay attacks. **Note:** Ignoring this validation is a security risk and contradicts to the OPC UA specification. This option shall only be used when dealing with a non compliant server, where the session establishment would fail otherwise. **Note:** Using this option will ignore a missing nonce for user token security, missing nonce for message security cannot be ignored. Problems with missing nonce in message security shall be solved by using MessageSecurityMode None.
Validation of Duplicate Nonce	Not allows to connect to a server, which does not generate new nonce. Client session specific. Client sessions will use the application setting by default. If only specific sessions shall disable this validation, disable it at the specific session. The effects are similar to sending empty (missing) nonce.
Validation of Endpoint Description	Not allows to ignore different Endpoint Descriptions Client session specific. Client sessions will use the application setting by default. If only specific sessions shall disable this validation, disable it at the specific session. A server returns endpoint descriptions both in the GetEndpoints response as in the CreateSession response. Both endpoint descriptions shall be identical and a client shall validate this. The reason for validation: The GetEndpoints is invoked on a non-secure communication and the response may be altered by man-in-the-middle. By altering the response, either security policies can be removed, endpoint URLs can be changed, this may cause connections to a wrong server, using a non-secure policy. **Attention:** The validation is applicable exclusively on secured connections. **Note:** Ignoring the validation will allow the client to communicate with non-compliant OPC UA applications. This option shall only be used if dealing with a non-compliant UA applications, where the session establishment would fail otherwise. When ignoring the validation you introduce security risks.

Advanced Operation Limits:

These settings are used in order to establish a communication and operate a set of services based on a connection with an OPC UA Server which used Operation Limits.

Note: Communication with OPC UA Servers which uses Operation Limits is possible based on a set of predefined profiles or a customized profile.

© Softing Industrial Automation GmbH



The following Operation Limits are configurable:

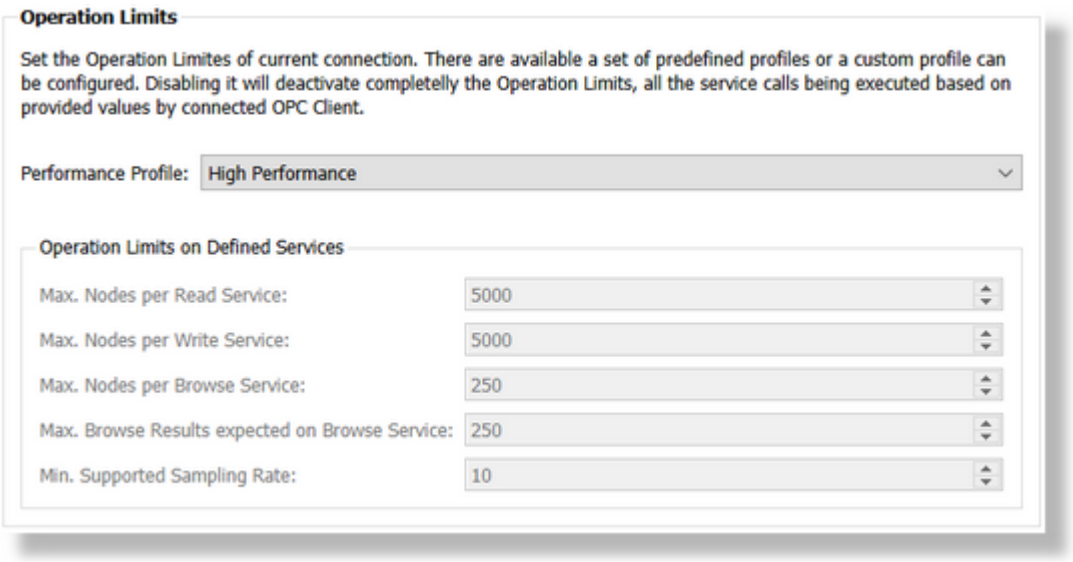
Parameter Name	Description
Maximum Nodes Per Read Service	The value indicates the maximum size of the NodesToRead array when a Client calls the Read Service.

	When a Read Service is requested by the connected Client on a UA Server Connection using a number of requests which is bigger than the configured value, then the transaction is spitted into multiple sub-transactions with maximum value of read requests which is individually executed as individual Read Service.
Maximum Nodes Per Write Service	The value indicates the maximum size of the NodesToWrite array when a Client calls the Write Service. When a Write Service is requested by the connected Client on a UA Server Connection using a number of requests which is bigger than the configured value, then the transaction is spitted into multiple sub-transactions with maximum value of write requests which is individually executed as individual Write Service.
Maximum Nodes Per Browse Service.	The value indicates the maximum size of the NodesToBrowse array when calling the Browse Service or the ContinuationPoints array when a Client calls the BrowseNext Service. When a Browse Service is requested by the connected Client on a UA Server Connection using a number of requests or continuation points which is bigger than the configured value, then the transaction is spitted into multiple sub-transactions with maximum value of browse requests which is individually executed as individual Browse Service.
Maximum Browse Results expected on Browse Service	The value indicates the maximum number of Browse Results requested being received on a Browse Service. When the number of Browse Results is bigger than then requested value, then the Browse Operation is providing as results only the first configured value of results and a continuation point which will be used for receiving the remaining results using a BrowseNext Service.
Minimum Supported Sampling Rate	The value indicates the minimum allowed Sampling Rate on monitoring items. If the requested value is bigger than the configured value, then it will be revised to the configured value.

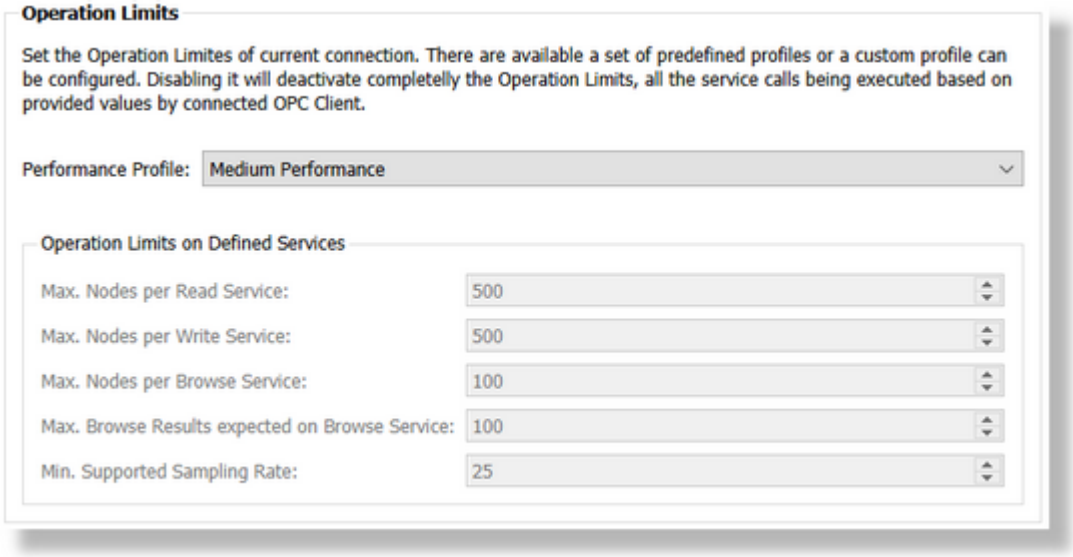
There is allowed to used a set of predefined profiles or to define a customized profile for complete Operation Limit:

- High Performance Profile:

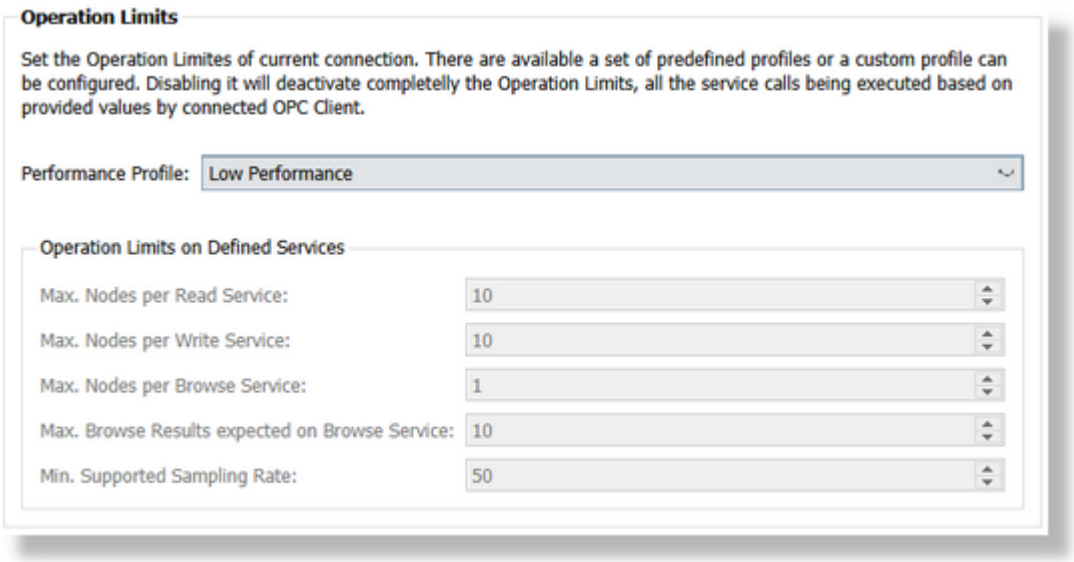
© Softing Industrial Automation GmbH



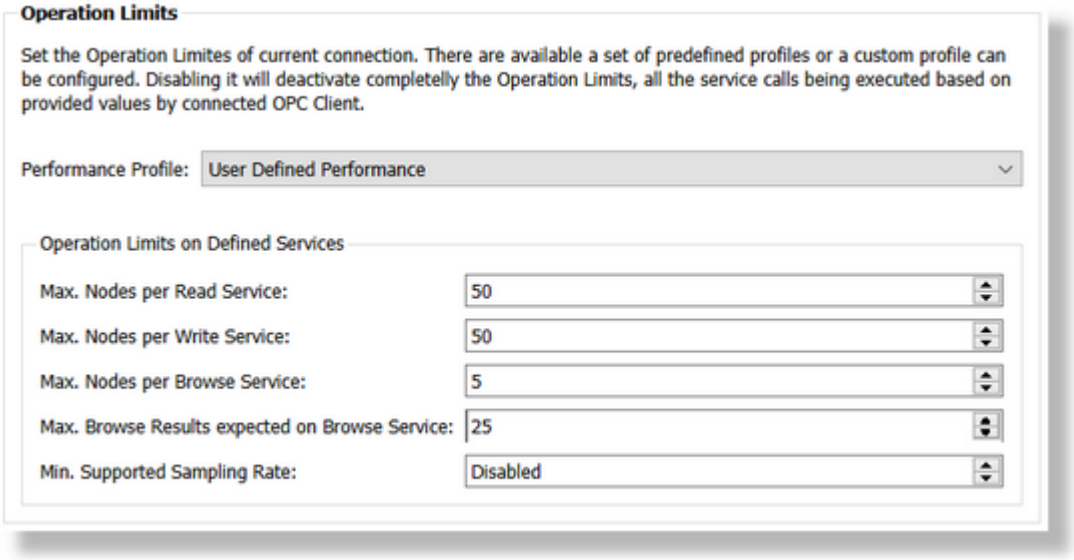
- Medium Performance Profile:



- Low Performance Profile:



- User Defined Performance Profile:



Note: Disabling an operation limit will disable its appliance - basically the number of requests from a transaction operated by a service call will used the same number of requests coming from original requests.

A browse operation requested by a connect OPC UA / DA Client is on a specific node is operated by dataFEED OPC Suite based on following operations:

- Browse the specified node - this operation is receiving all the children of specified node.
- For each received children with node type set to variables is it tried being read a set of 4 properties: Data Type, Access Rights, Value Rank an Array Dimension. This operation is by default executed on a single read operation for the entire set of read requests - basically only one read transaction is created for all requests.
- For each received children it is tries to determine of this node has or not children (this operation is required in order to create the address space or classical (DA, XML-DA or Tunneling) OPC Server) - this operation is executed based on a browse service - basically it

© Softing Industrial Automation GmbH

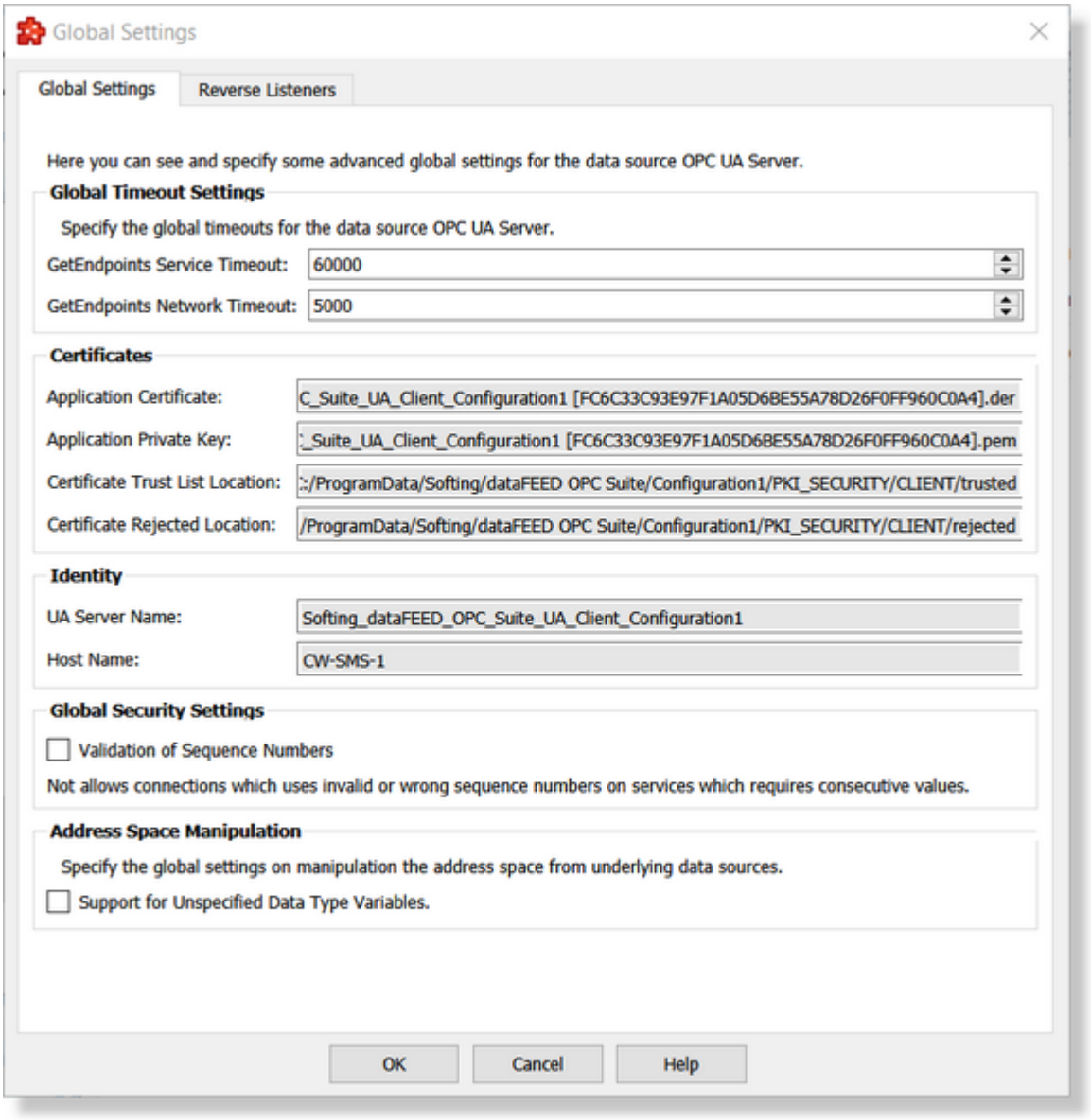
is created a transaction with a set of browse requests which is executed on a single operation.

Attention: Due the reason that a browse operation is executed on connected OPC UA Server based on a set of other operations (browse on specified node, reading of 4 properties on each received result and re-browsing each children) all the operation limits on this operation should be specified on OPC UA Servers which uses operational limits.

7.3.1.7.4 Global Settings

Global Settings

Global settings dialog contains general settings applied to all OPC UA Server (including OPC UA Tunnel) connections. The user is allowed to modify only timeout related settings. The read only information is presented because it may be relevant for dataFeed administrators.



1. Global Timeout Settings

- **GetEndpoints Service Timeout** - Specifies the time-out in milliseconds for establishing a secure channel in the GetEndpointsFromServer service call. Default value is 5000 (ms). Value range: UInt32.
- **GetEndpoints Network Timeout** - Specifies the time-out in milliseconds for the GetEndpointsFromServer service. Default value is 60000 (ms). Value range: UInt32.

2. Identity Settings

- **UA Server Name** - Specifies the server name
- **Host Name** - Local host name

Attention: The local host name should contain exclusively ASCII characters. Usage of special/language related characters (sample: Ä,Ö, Ü, ? , ä, ö, ü, ß) are recommended being not used as host name by the operation system (Microsoft Windows 7 or later) and not allowed being used inside the application certificates by the OPC UA Specifications.

3. Certificates Settings

- **ApplicationCertificate** - Absolute file path to application certificate (der format)
- **ApplicationPrivateKey** - Absolute file path to application private key (pem format)
- **CertificateTrustListLocation** - Local folder containing the trusted certificates
- **RejectedCertificateLocation** - Local folder containing the rejected certificates

4. Global Security Settings

- **Validation of Sequence Numbers**

  Not allows to ignore the sequence number implementation (see OPC Foundation Errata 1.04.3, General communication settings).

  The validation of sequence numbers can be disabled on complete communication, both for OPC UA server and client. It also needs to be disabled at Application.

  The new OPC UA specification solves the loss of messages due to a roll-over of sequence numbers (at 4,294,966,271). The issue was solved by defining a concrete start of sequence numbers (1023) including after roll-over. Previous specifications allowed all starting sequence numbers less than 1,024, so the new definition does not contradict the older specifications. For compatibility reasons sequence numbers less than 1,023 and greater than 4,294,966,271 shall be still accepted.

Note: Ignoring the mantis issue will allow the client or server to communicate with non-compliant OPC UA applications. This option shall only be used if dealing with a non-compliant UA applications, where the session establishment would fail otherwise. When ignoring the implementation for the issue, messages can be lost without recognizing.

© Softing Industrial Automation GmbH

5. Address Space Manipulation

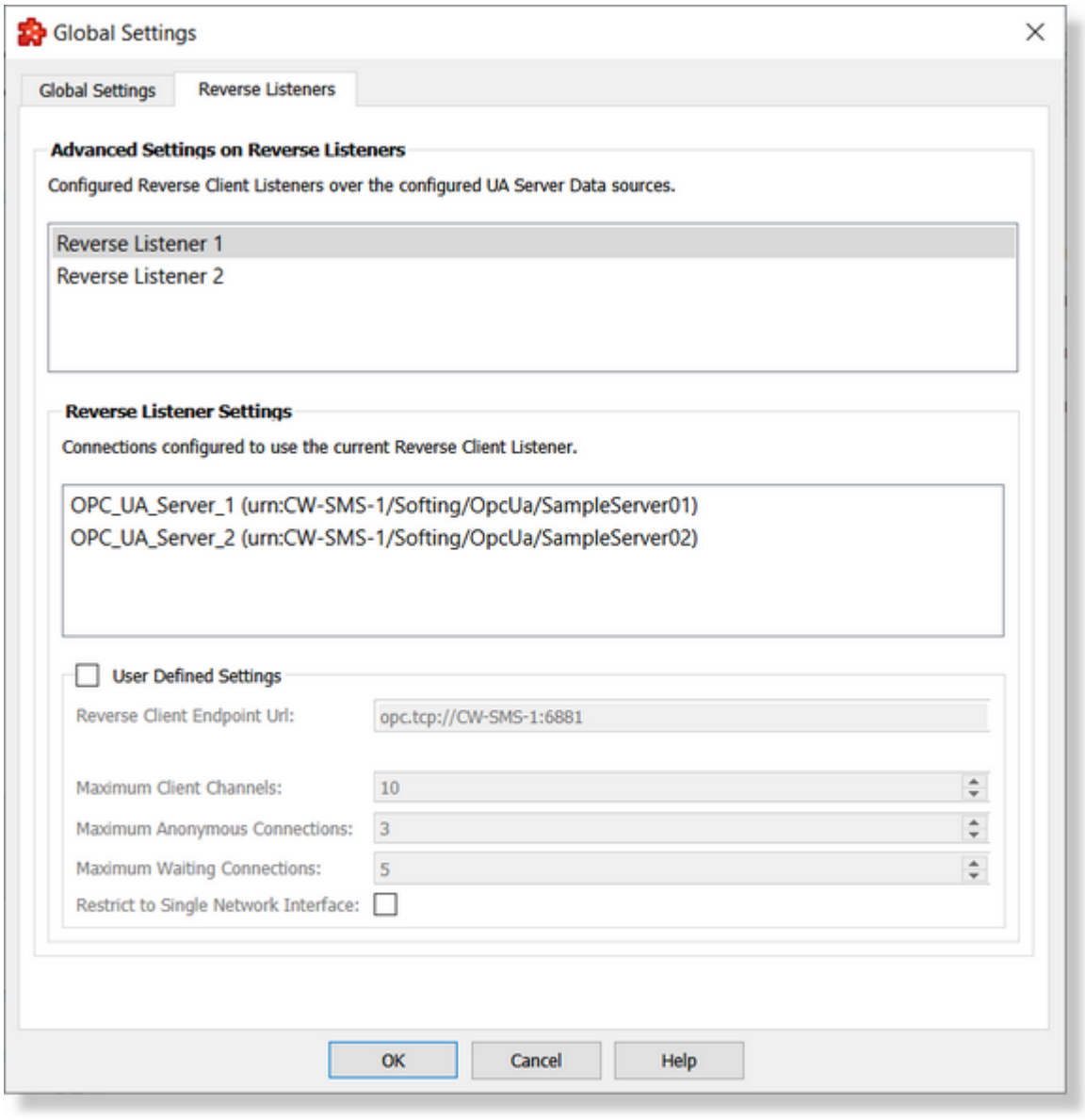
- **Support for unspecified datatype variables**

  Allows to use connections to OPC UA Servers which contains variables with unspecified data type (not specified or usage of DataType Null). Enabling the option will convert the data type of those variable to abstract, being handled as variants.

Note: Validation of items will failed for variables with undefined data type when the option is disabled - the dataFEED OPC Suite application completely ignoring those variables.

Reverse Listeners

The global settings on reverse listeners (if exists at least on reverse connection defined) is specifying the advanced settings of each defined **Reverse Listener**.



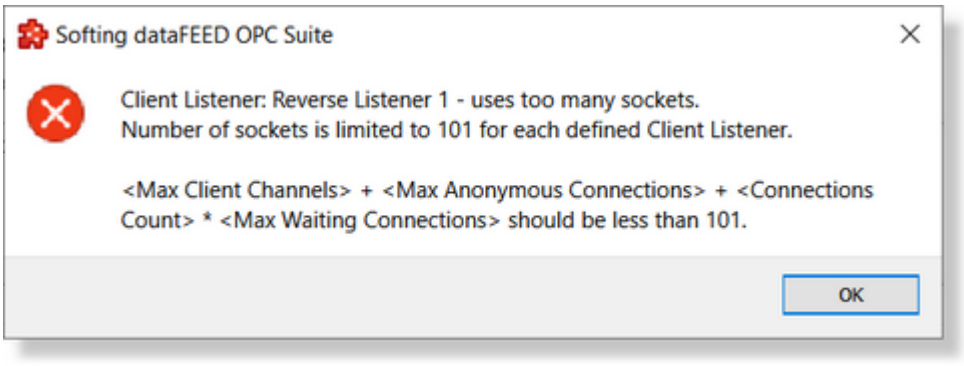
The following settings are configurable for a defined reverse listener:

- Maximum Client Channels: The listener reserves this number of sockets for connections using this reverse listener. This number of channels is shared among all reverse connecting servers. If the number is exceeded, then new connections from servers will fail, thus also he client cannot establish more connections (e.g. as Sessions) via this reverse listener.
- Maximum Anonymous Connections: Once a server opens a physical connection (socket) to the client, the connection will be "anonymous" until the server sends the reverse hello. By sending the reverse hello, the server identified itself and the connection is no longer anonymous. Identified connections will be added to the "waiting connections" from this server until a client session will use it. The number of anonymous channels is shared for all incoming reverse connections from all reverse connecting servers to this listener.

  If this number is exceeded, then the oldest anonymous connection to this listener is closed. A good guide value is the number of expected reverse connecting servers + 1.
- Maximum Waiting Connections: The waiting connections are connections which are mapped to a server (the "reverse hello" was sent), but not yet used by a client session. The used number of waiting connections of one server endpoint does not affect the number of waiting connections of any other server endpoint.
- Restriction to Single Network interface: By default the reverse listener will listen on all network interfaces for incoming connections. If listening is restricted to a single IP address (IPv4 or IPv6) a server can connect only to the IP address in the URL. If listening on a specific interface the client URL shall contain an IP address (not a name to be resolved).

The settings are pre-configured the with the optimized values, still the user has the possibility to use its own set of user defined values.

Attention: The maximum number of sockets which can be used (reserved and open) by a listener is limited to 101. In case when the user defined settings is exceeded this limit, the user is notified with an error message signaling the problem.



Note: A reverse listener can be shared by multiple reverse connections. In this scenario, defining a set of user defined settings for a reverse listener will be reflected on all connections which shares it.

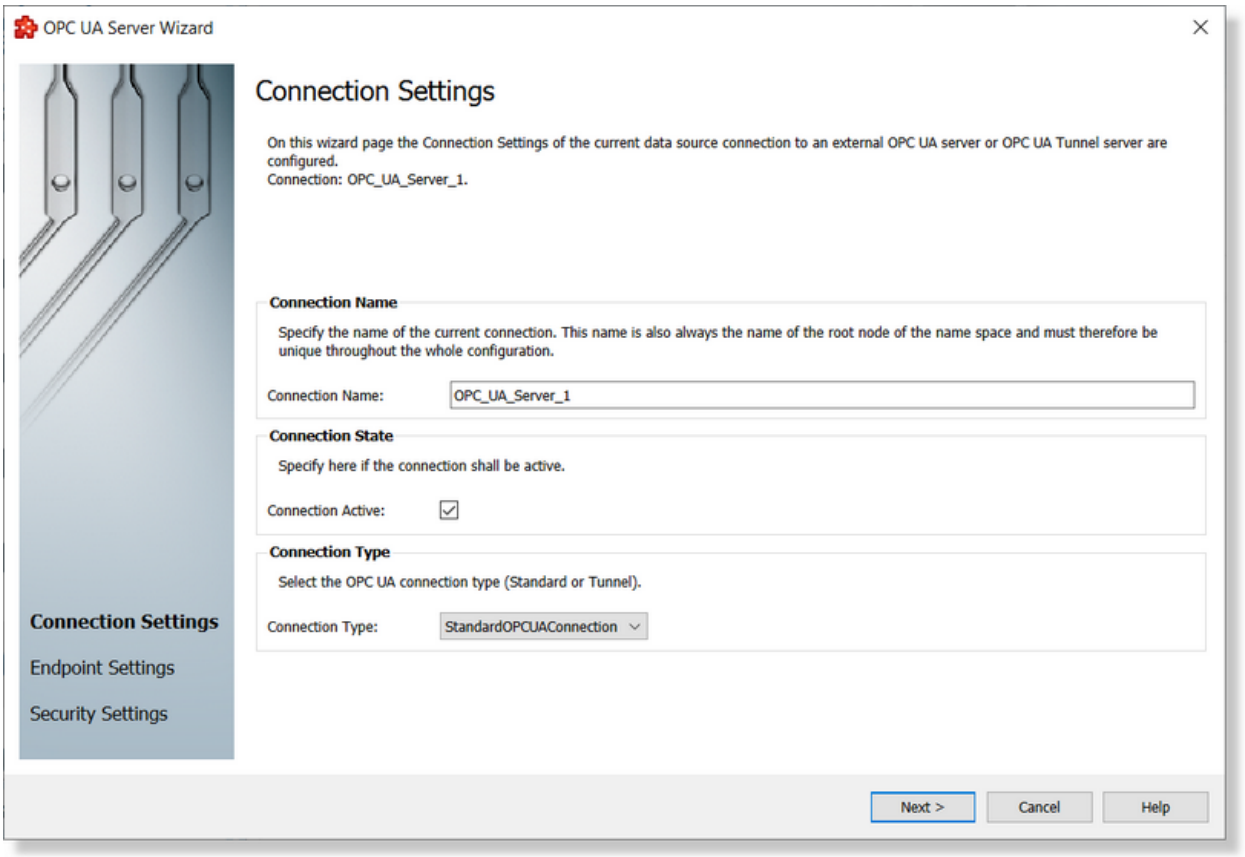
© Softing Industrial Automation GmbH

7.3.1.7.5 UA Server Connection Wizard

With the UA Server Connection Wizard you can create and modify a data source connection to an external OPC UA Server:

- Specify connection settings @ UA Server Connection Wizard - Connection Settings
- Specify UA Server Endpoint URL @ UA Server Connection Wizard - Endpoint Settings (available only Standard OPC UA connections)
- Specify connection's security settings @ UA Server Connection Wizard - Security Settings (available only Standard OPC UA connections)
- Exchanging the connection settings between OPC UA Tunnel Server and OPC UA Tunnel Client @ UA Server connection Wizard - OPC UA Tunnel Settings Exchange (available only for OPC UA Tunnel connections)
- Summary of configured connections settings @ UA Server connection Wizard - Connection Settings Summary (available only for OPC UA Tunnel connections)

On this page the connection name that will uniquely identify the data source and state of the connection is specified.



Connection Name

An unique identifier for the current connection. In case the provided name is already used by other connection, the user use is not allowed to go into the next wizard page.

Connection State

This flag specifies if the current connection will be enabled or not. A disabled connection will be ignored by dataFEED OPC Suite Runtime but the connection settings will be persisted so it can be enabled again at a later time.

Connection Type

The connection type reflects the connection type:

- Standard OPC UA connection (acting as standard OPC UA client connection based on OPC UA protocol)
- OPC UA Tunnel Connection (acting as OPC UA Tunnel client connection to other instance of dataFEED OPC Suite product which acts as OPC UA Tunnel server).

Based on the connection type the wizard follows the Standard OPC UA Connection or OPC UA Tunnel connection definition wizard pages.

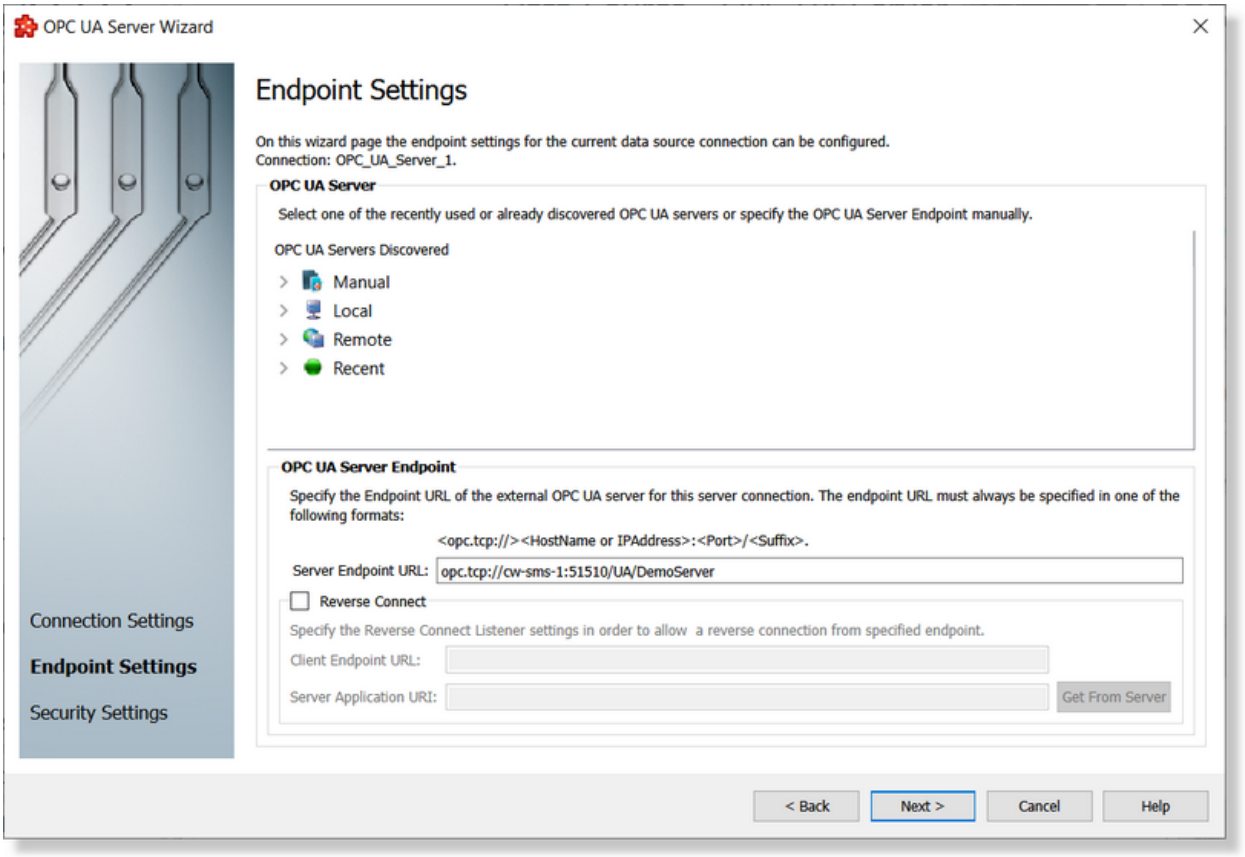
Note: Definition of an OPC UA Tunnel connection follows a diferent set of steps than a Standard OPC UA Connection.

(Standard OPC UA Connection) UA Server Connection Wizard - Endpoint Settings Page >>>

(OPC UA Tunnel Connection) UA Server Connection Wizard - OPC UA Tunnel Settings Exchange >>>

On this wizard page the external OPC UA Server endpoint URL is configured.

© Softing Industrial Automation GmbH



OPC UA Server panel

This panel contains a server discovery widget. It supports the user to find a server locally or on the network and allows to specify an Endpoint that was used previously.

It contains 4 basic entries:

- **Manual**: By expanding this node the user can add a server address or a discovery server. The configurator will try to obtain the available endpoints for these entries.
- The manually added entries will be saved for the next session. They can also be removed from the list or can be edited using a popup menu.
- **Local**: Under this node all servers listed in LDS (local discovery server) are shown. Expanding each of them will retrieve server's available endpoints.
- **Remote**: Similar to local node. But in this case on the second level the network domains are shown, on the third level workstations, then available servers and lastly the endpoints.
- **Recent**: Once a connection wizards completes, the selected endpoint will be stored in Recent group to be available for further configurations. Please note that only the latest 10 endpoints are stored.

The user has the option to remove them using the popup menu.

Note: Once a server endpoint is clicked, it is automatically added to the Endpoint URL edit box.

OPC UA Server Endpoint panel

This panel contains Endpoint URL edit box beside some explanations about accepted formats. Only the URL starting with opc.tpc:// are admitted.

The host can be specified either by HostName or Ip address.

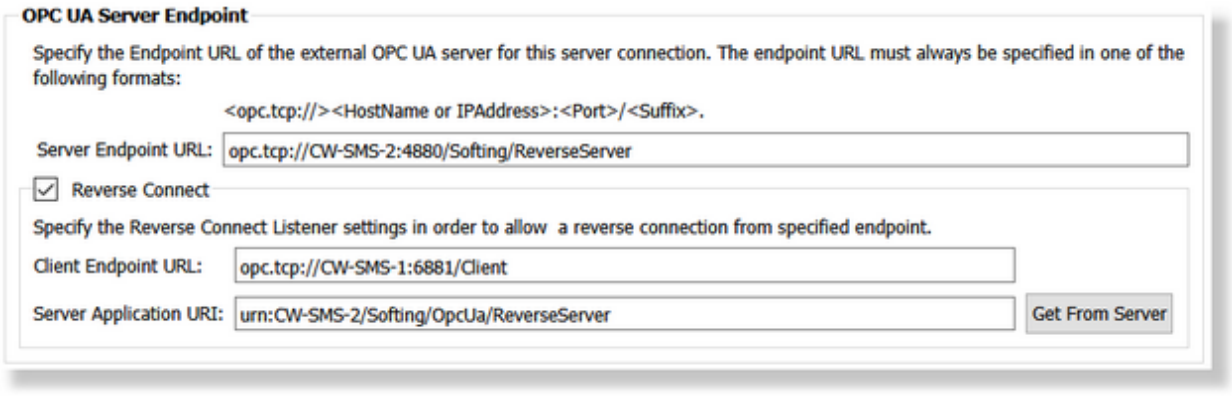
The XML message encoding type is not supported. Therefore the associated endpoints are shown in the discovery panel as disabled.

Reverse Connect support

The application is supporting **Reverse Connectivity**. This functionality is used specially in the context of existence of a firewall between the dataFEED OPC Suite acting as a Opc UA Client application and the Opc Ua Server which blocks the possibility to initiate the connection from the client side.

Enabling the **Reverse Connect** functionality, it changing the behavior of the connection manager. On a **Forward Mode**, the connection is established by the client application, which opens the secure communication channel directly. On **Reverse Mode**, the Opc Client UA is not the one who initiates the connection and opens the unsecured channel (this being not possible due the reason that the firewall on server side is blocking it), the Opc UA Client is just opening a **Reverse Listener,** which listen on a dedicated port for Opc UA Servers which tries themselves to initiate the communication (only the unsecured channel being open on server request) to the Opc UA Client. When the **Reverse Listener,** identifies a server which tries to initiate the communication, it validates the server (based on Server Application Uri, Server Endpoint Url and its certificate) and opens the secured communication channel.

The functionality can be enabled by enabling the **Reverse Connect** support:



The following identities are mandatory being provided:

- ***Client Endpoint Url***: the endpoint Url of the **Reverse Listener** should be provided by the user. By default it auto-generated based on machine host-name and an auto-incremented port. The Client Endpoint Url will be used by the Reverse Listener to listen for connection attempts from the OPC UA Servers.

© Softing Industrial Automation GmbH

The Client Endpoint Url could use the current Hostname or IP Address.

Note: Only binary encoding message is supported, the XML encoding being not supported.

- ***Server Application Uri:*** the application Uri of the data source Opc UA Server. This is use to uniquely identify the Opc UA Server. This information can be obtained online from the Opc UA Server by pressing "**Get From Server**" button.

This operation will start the Reverse Listener, waits for a server (based on the Server Endpoint Url previously defined) to try to connect (to open the unsecured channel) and collect its Server Application Uri. This operation should be executed inside a timeout of 60 seconds - in case when the server not tried to connect inside the timeout interval, then the operation will fail.

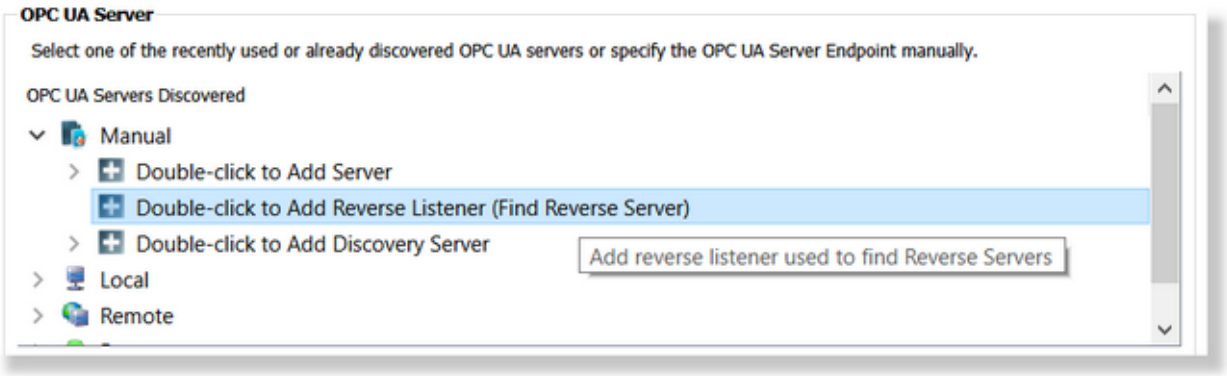
Attention: The reverse listener can be uniquely initialized by an application, in case when the Runtime was already started using the same listener (on the same port) then reverse listener can not be initialized (the port is in already in use) and the operation will fail.

Attention: The port used by the reverse listener (based on the Client Endpoint Url) should not be used on the same interface by other application, otherwise the Reverse Listener cannot be initialized and the operation will fail.

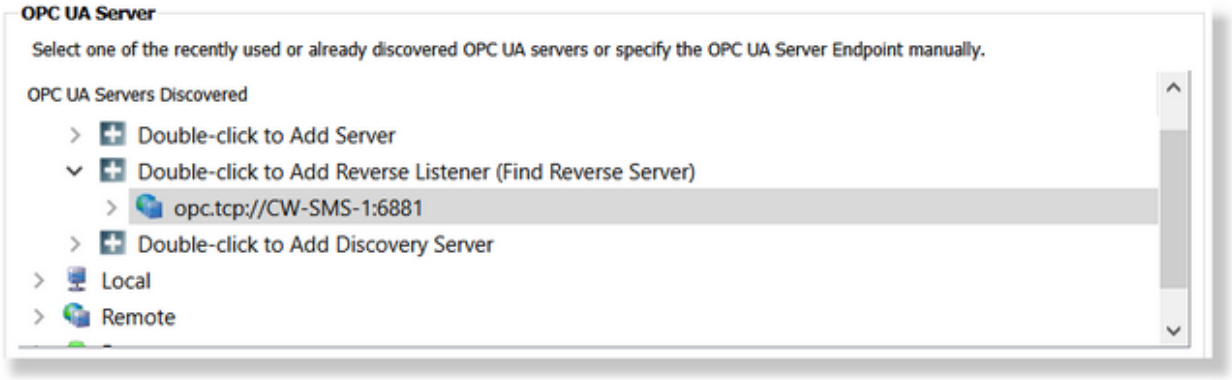
The application supports multiple connections on reverse mode, handled using one or multiple listeners. Basically, a configured Reverse Listener (being generated based on the Client Endpoint Url) can be assigned to multiple reverse connections. Also, it allowed to define multiple Reverse Listeners on the same configuration. A listener, can be uniquely identified based the pair of Host Name/IP Address and Port used on the Client Endpoint Url definition.

The advanced settings of a configured reverse listener can be modified using the Global Settings dialog available on main UA Server data page.

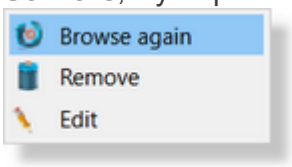
The Opc UA Servers, supporting Reverse Connect, can be discovered online by specifying the **Client Endpoint Url** on **Manual Opc UA Server Discovery (Find Reverse Server)**:



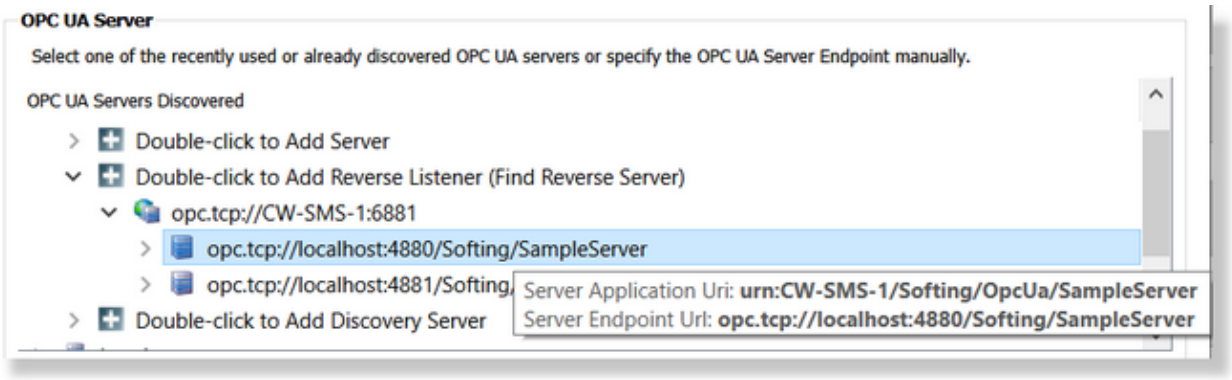
This operation requires to provide the Client Endpoint Url on "**Double-click to Add Reverse Listener (Find Reverse Server)**" option. Immediately after, the Reverse Listener will be created (not activated and started yet) and become available for **Find/Browse** operations:



The newly added **Reverse Listener** can be activated and started for discovering **Reverse Servers**, by expanding it or using the **Browse again** functionality from the context menu

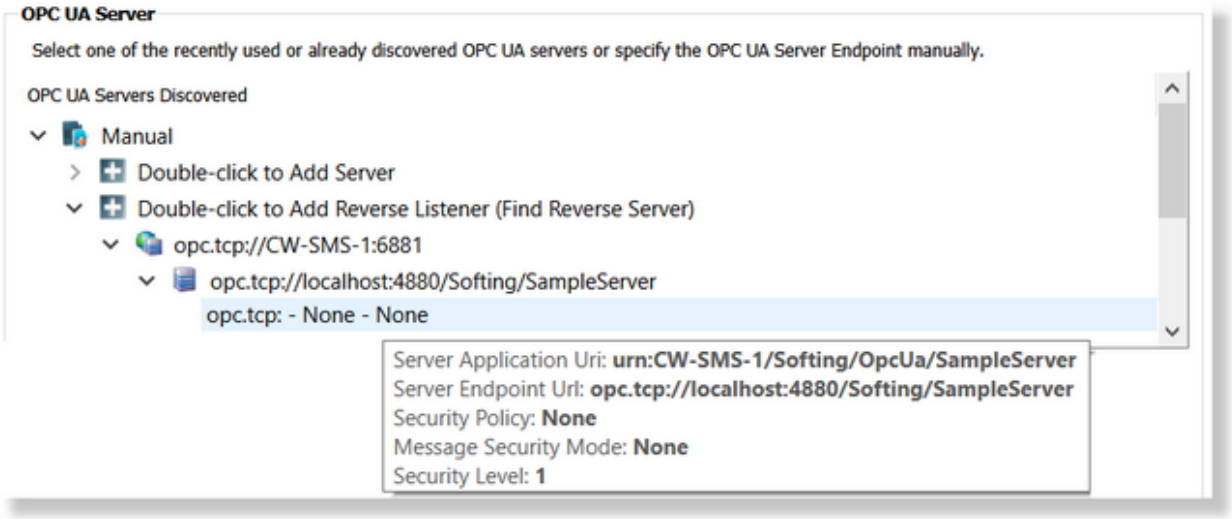


(activated on right click). Immediately after, the **Reverse Listener** is activated and starts listening for Opc UA Server(s) which tries to initiates the connection to the listener. Those Opc UA Server(s) are identified and made available for usage (selection). The listener will listen only for 60 seconds on servers trying to connect, if no connections are established, the timeout will expire and the operation will fail - in this case no Opc UA Server being found and available for selection.



Expanding or browsing a selected server will discover all its endpoints, making them available for selection and usage as Opc UA Server data sources.

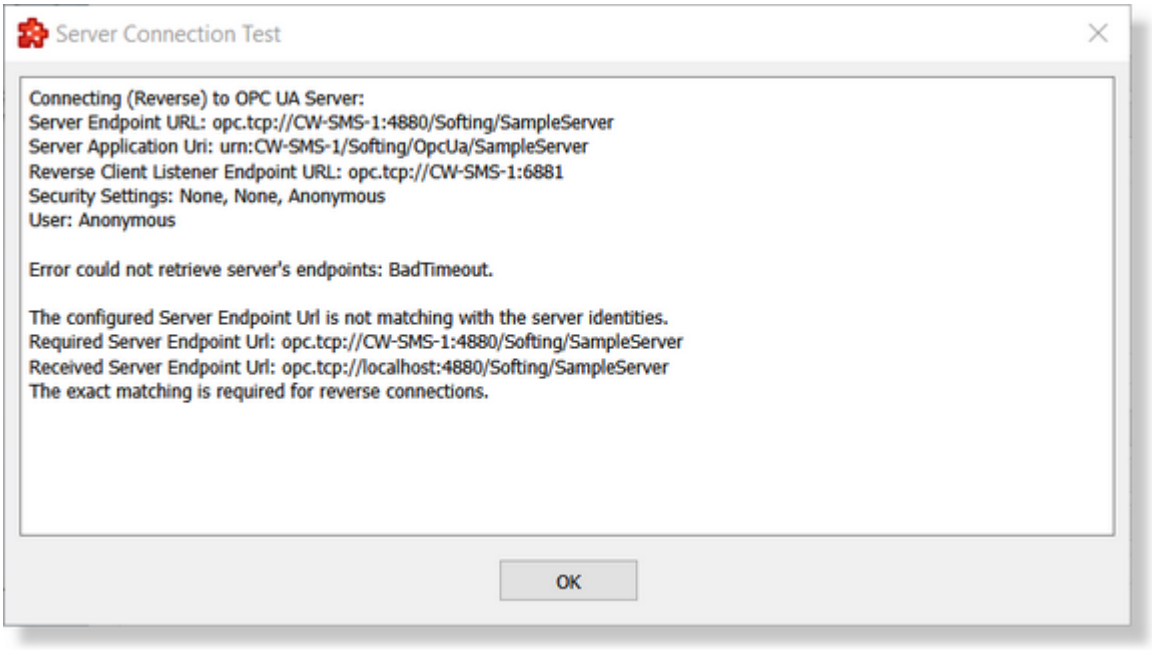
© Softing Industrial Automation GmbH



An endpoint can be selected, immediately after, it will be used as data source... all the necessary identities being obtained online from the server. All the server identities (Server Endpoint Url, Client Endpoint Url and Server Application Uri) are pre-configured with the values from the selected server.

Attention: Finding Reverse Servers is operated based on starting a reverse listener - in case when the reverse listener cannot be started the operation will always fail. A common reason for not starting a reverse listener is: the specified port is used by an external application, by other dataFEED OPC Suite configuration in running state or by the Runtime of the same configuration (in case when the configuration is started).

In the context of Reverse Connectivity, it is very important to use as Server Endpoint URL exactly the one defined on the Opc UA Server as data source. Switching the IP Address with the Hostname (operation allowed for a forward connection) is not allowed on reverse connections. When this is happening, the Reverse Listener will reject a server connection attempt from the OPC UA Server side. This can be detected based on "Test Connection" operation - the results specifying it:

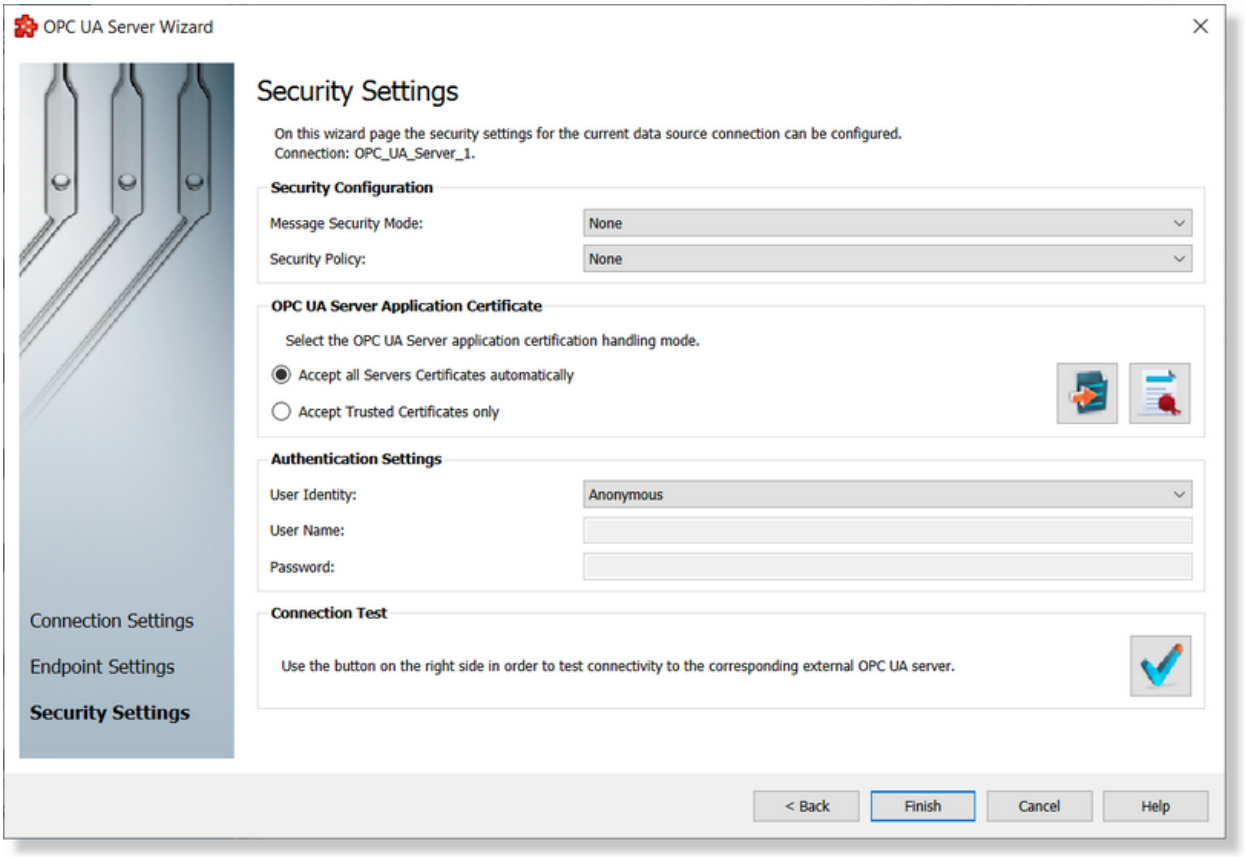


Note: The next button is not available as long as the Server Endpoint URL edit is empty. In case of a reverse connection, the Client Endpoint Url and Server Application Uri are also mandatory being provided.

<<< UA Server Connection Wizard - Connection Settings UA Server Connection Wizard - Security Settings >>>

On this wizard page various parameters regarding secure communication with UA Server are configured. Beside that a test connection functionality is provided.

© Softing Industrial Automation GmbH



Security Configuration

Security Configuration settings consists in Message Security Mode and Security Policy settings.

Message Security Mode

Specifies whether signing or encryption is used and when it is used if messages are exchanged. Possible options are:

- *None* – no security is applied. Messages can be read over the network. This is the most insecure mode, but also the fastest one.
- *Sign* – messages are sign using a certificate. The message body is not encrypted, but data integrity is assured.
- *Sign & Encrypt* – messages are signed and encrypted. All of them are private and their integrity is assured.

Security Policy

Specifies the encryption algorithm applied for messages. Possible options are:

- *None* – no encryption is used
- *Basic128RSA15* – 128 bit encryption is used.
- *Basic256* – 256 bit encryption is used.

OPC UA Server Application Certificate

This section allows to select the certificate handling mode:

- *Accept all servers certificates automatically.* Communication between dataFEED OPC Suite and external OPC UA servers is possible even when the target server is not trusted.
- *Accept trusted certificates only.* In order to trust a server certificate the user should access Server certificate retrieval by pressing the  button.
- A general overview and handling for certificate store can be found inside Certificate Management by pressing the  button.

Authentication settings

Specifies the credentials which must be passed to the server for authentication. Possible options are:

- *Anonymous* – no user credentials are provided for authentication.
- *User Name* – a user name and a password is provided for authentication – these must be specified.
- *Certificate* - the user must provide a certificate to prove its identity.

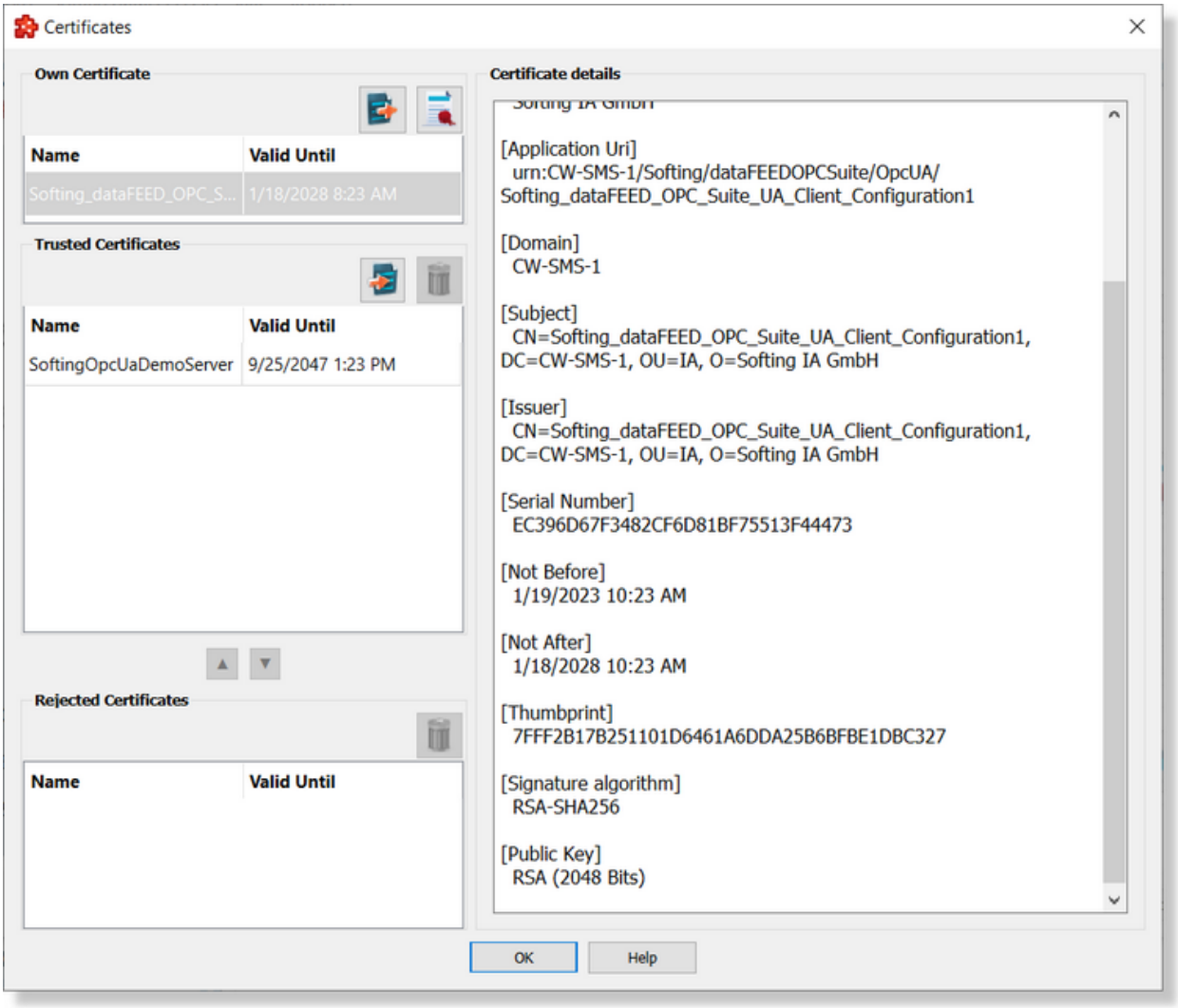
Connection test.

Use the connection test button  to check the availability of the server specified or the correctness of the authentication and security settings.

<<< UA Server Connection Wizard - Endpoint Settings Page

This dialog facilitates certificates handling and details view.

© Softing Industrial Automation GmbH



Left column contains Own certificate, Trusted certificates and Rejected certificates lists while the right column is populated with the details of the selected certificate.

Own Certificate.

This list contains only one certificate and offers the possibility to be exported.

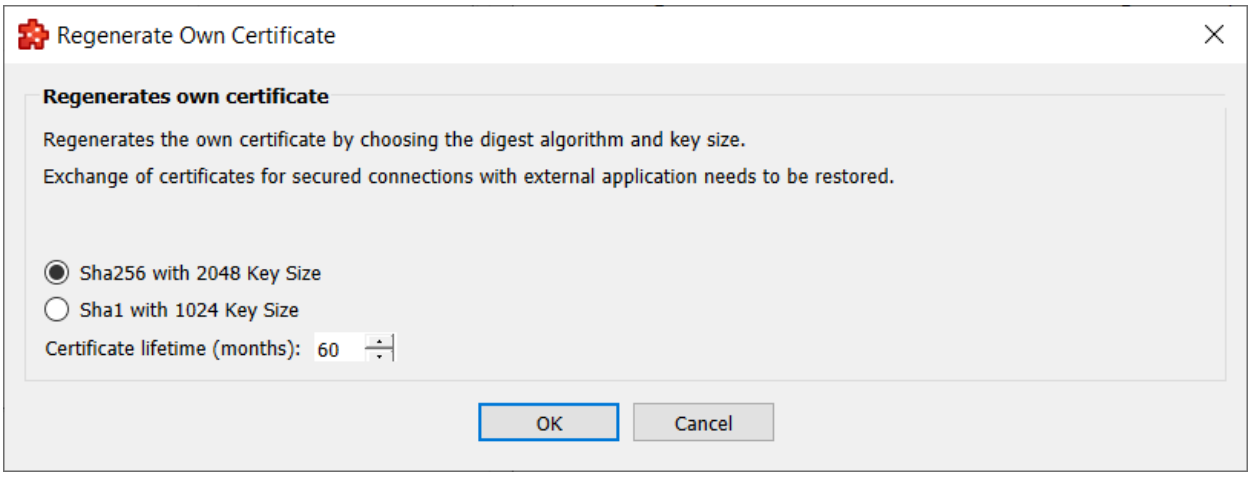
By using the export button  the user is prompted to select a destination folder where to save dataFEED OPC Suite UA Client application certificate. This functionality is particularly useful when the Message security mode is other than "none".

The ***Regenerate own certificate***  button allows you to restore your own certificate based on the selected check-sum algorithm and key size for encryption. Regardless of the settings of the previous certificate, the check-sum algorithm SHA256 and the key size 2048 are preset for encryption.

It is also possible to set the certificate lifetime in months (default value is 60 months).

Note:
By executing this operation, the application's own certificate is regenerated and all secured

connections to the application should be configured again. In this way, the trust relationship between the application and the connected servers and clients is renewed.



Trusted Certificates.

This group contains the list of the currently trusted certificates. The validity date can be easily checked on the "Valid Until" column. Certificates from this list can be dragged and dropped in the rejected certificate list.

By using the import button  the user is prompted to select a server application certificate (.der format) that will be added to the trusted folder. This operation basically can substitute Receive server certificate operation.

One certificate can be deleted using  button.

Rejected Certificates.

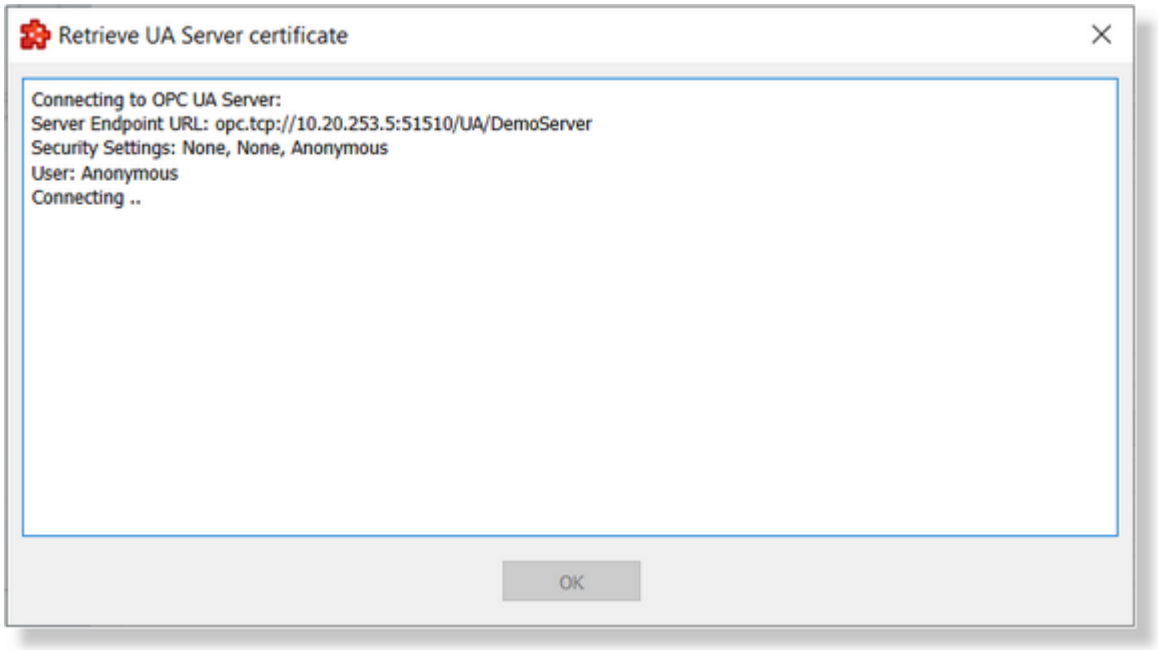
Contains the list of the rejected certificates. Each certificate can be dragged into trusted or deleted.
The up and down arrow buttons can be used to easily move certificates from trusted to rejected locations and vice-versa.

Certificate details.

In this area shows some details of the selected certificate like: location (on the disk), Application Name, Application Uri, Organization, Subject, Issuer, Validity dates and so on.
The details area is read-only.

The retrieve server certificate button  obtains external server's certificate and stores it into the trusted folder. While dataFEED OPC Suite Configurator tries to connect to the external OPC UA server a waiting dialog is shown:

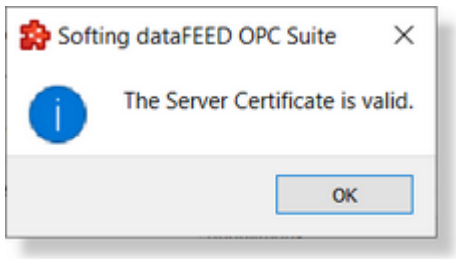
© Softing Industrial Automation GmbH



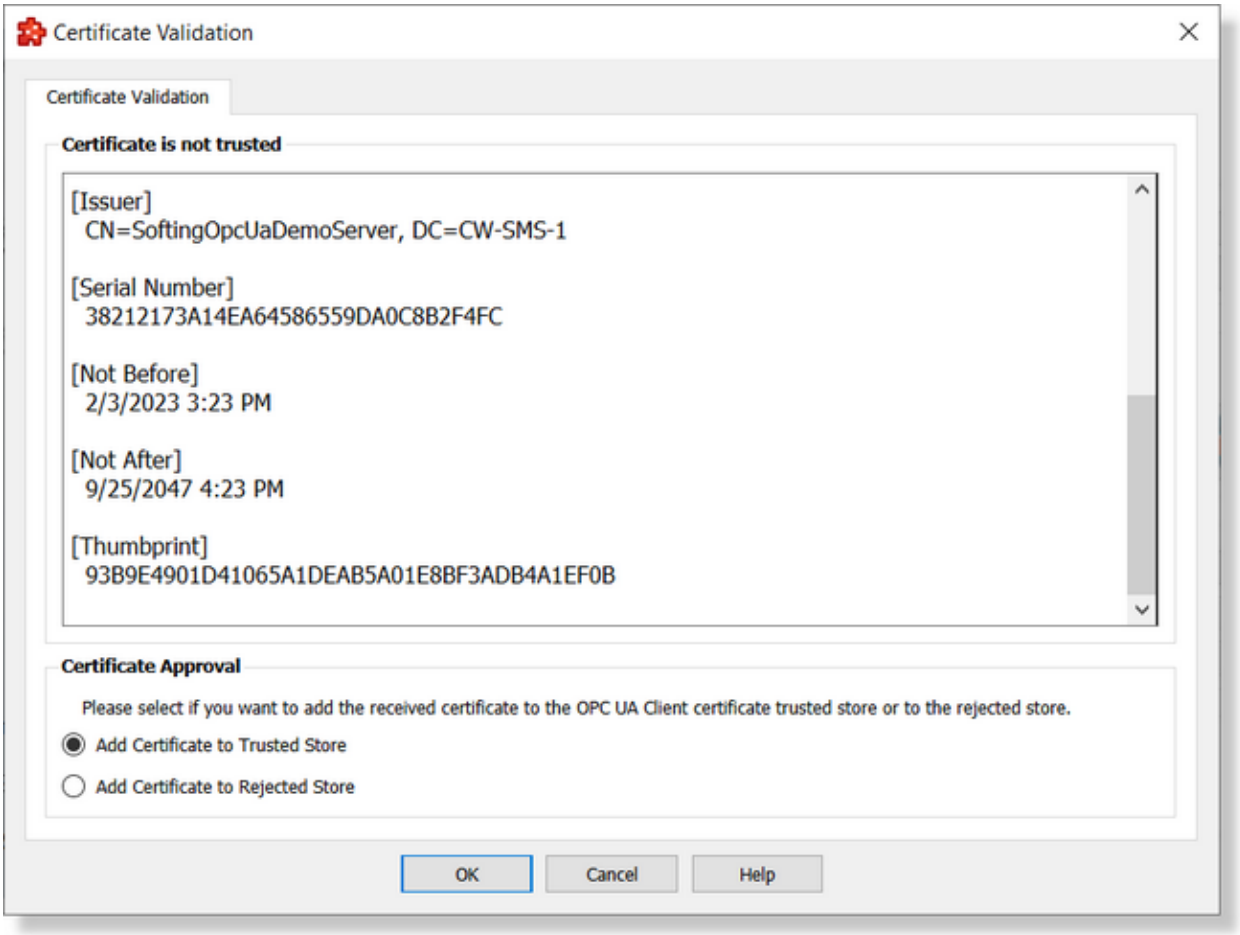
Both a successful and a failed connection attempt are displayed in a message window.

A Certificate validation is also displayed allowing the user to inspect the incoming certificate and trust it.

If the server's certificate is already trusted and valid, the following box is displayed:



The Certificate validation dialog allows the user to briefly investigate a certificate by checking details like Issuer details, expiration date etc



Certificate view

Certificate view offers information regarding Subject, Issuer, Serial Number, Validity dates and Thumbprint.

Subject and Issuer are composite fields. They can contain information about Domain Component, Email, Common Name, Organization Unit, Organization, Locality, State and Country.

Certificate Approval

A user can trust this incoming certificate or reject the certificate.

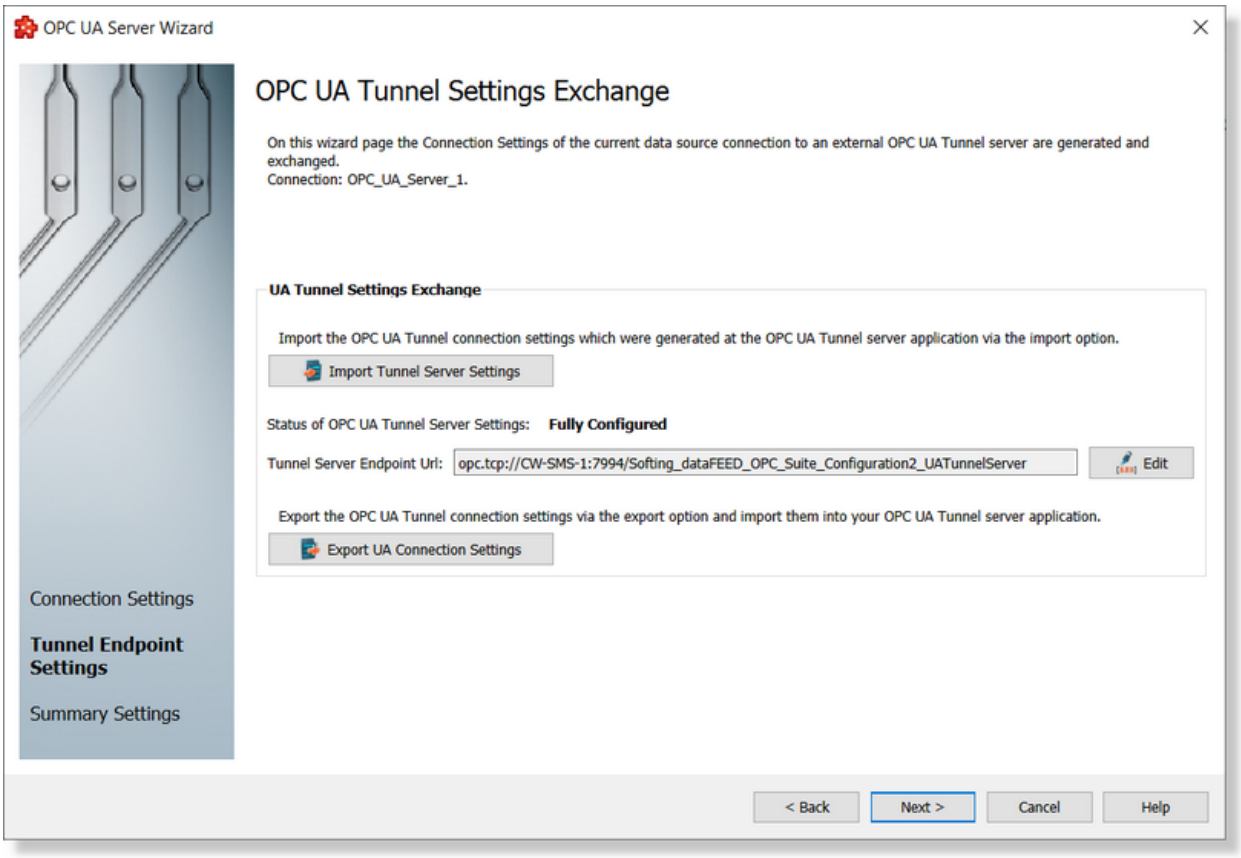
Note: adding a certificate to the rejected store does not mean the connection to server is prohibited (do not confound rejected with revocation list). Rejected store is a folder where certificates are kept to be investigated by an administrator at a later time, then moved to trusted, revocation or ignored.

On this wizard page it is specifying the connection parameters for an OPC UA Tunnel connection to other instance of dataFEED OPC Suite acting as OPC UA Tunnel server.

Also, the trust relation between OPC UA Tunnel client and server is established by importing the Tunnel Endpoint settings and connection parameters (user authentication certificate) from dataFEED OPC Suite Data Destination OPC UA Tunnel Server and exporting the currently

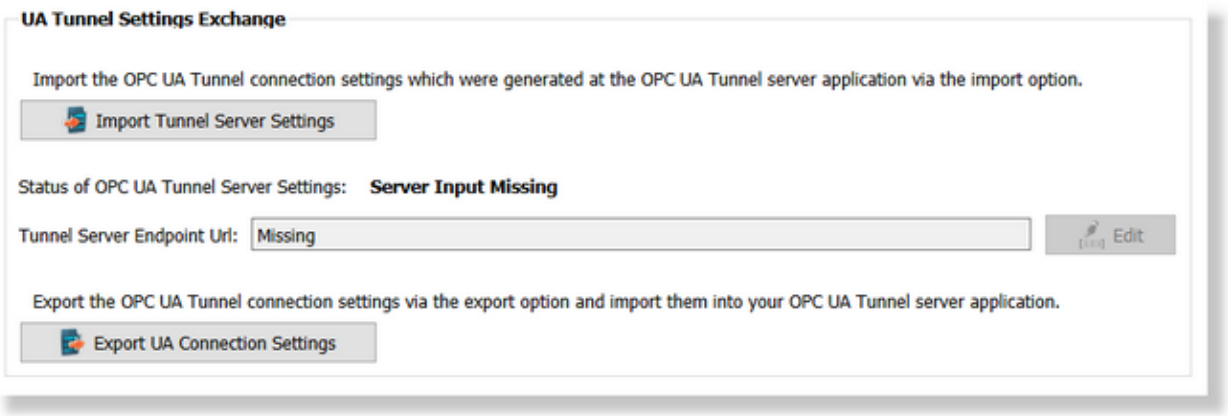
© Softing Industrial Automation GmbH

defined OPC UA Tunnel Connection settings which later will be imported by the OPC UA Tunnel Server.



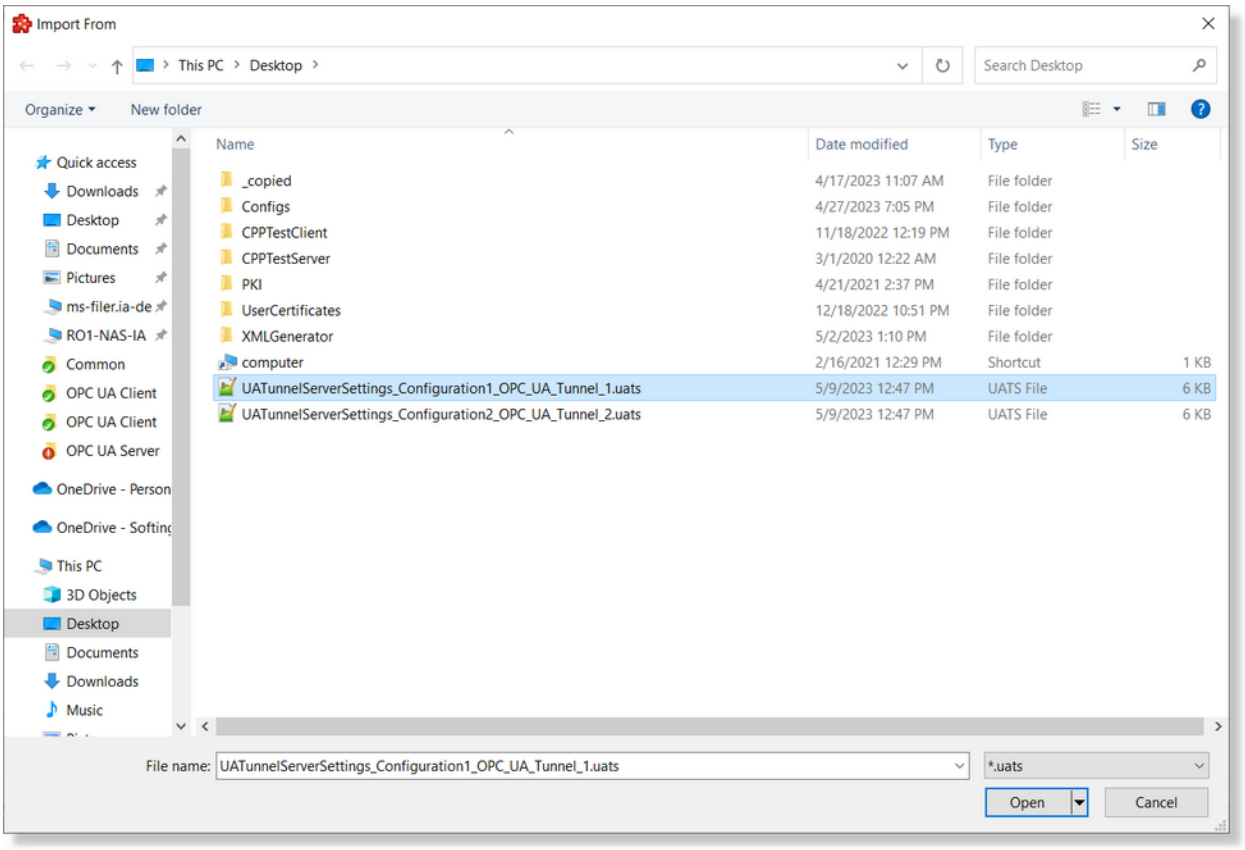
Attention: The import operation in mandatory for a newly defined connection. On this step, it is imported the connection parameters and it is established the truste relation with the OPC UA Tunnel server.

Before importing a the *.uats file (containing the settings from the OPC UA Tunnel Server), the **Status of OPC UA Tunnel Server Settings** is "**Server Input Missing**", this signaling that the connection is not completely defined.



Attention: A connection which is not completely defined, the OPC UA Tunnel Settings being not imported yet, is not allowed to be finalized (finalizing wizard is not allowed).

It is mandatory to import the OPC UA Tunnel server settings by pressing the  *Import Tunnel Server Settings* button. Immediately after, it is allowed to select the file which will be imported.

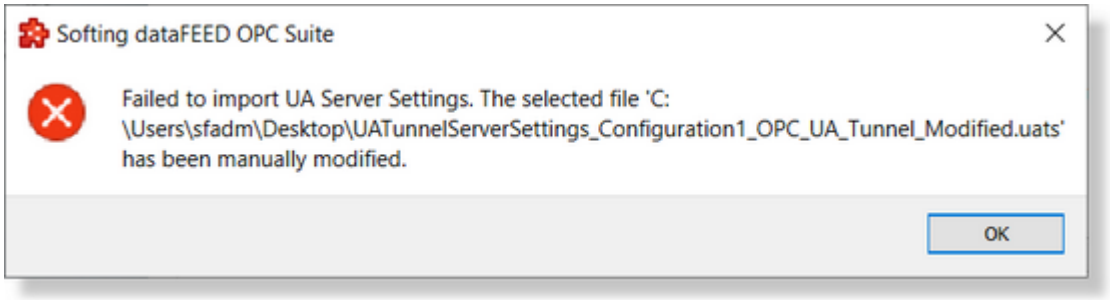


Note: The OPC UA Tunnel server settings files are obtained by exporting them from the OPC UA Tunnel Server configuration - Data Destination OPC UA - OPC UA Tunnel Endpoint.

This file contains information about the OPC UA Tunnel settings (identities of the OPC UA Tunnel Endpoint and certificates own by the OPC UA Tunnel Endpoint (Application and User Authentication certificates)

Attention: A manually edited file (modified with an external tool), is rejected by the import operation.

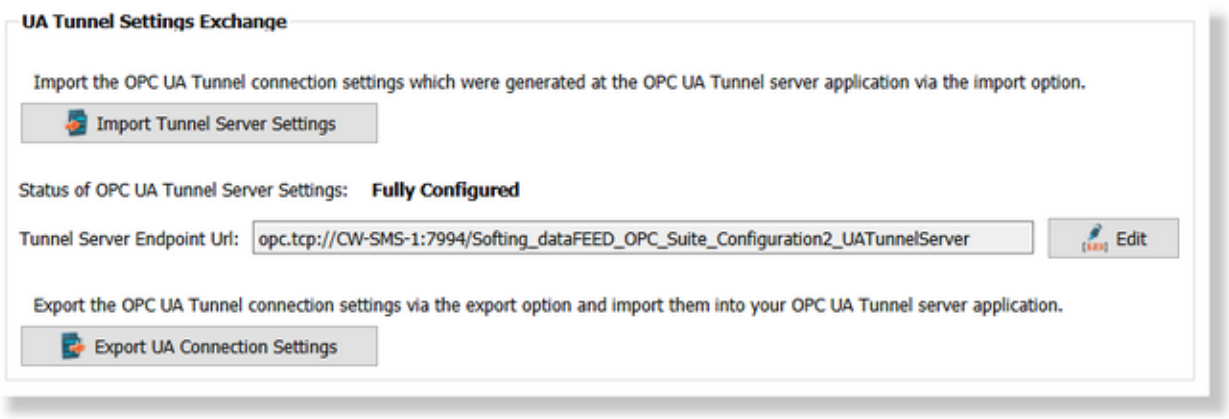
© Softing Industrial Automation GmbH



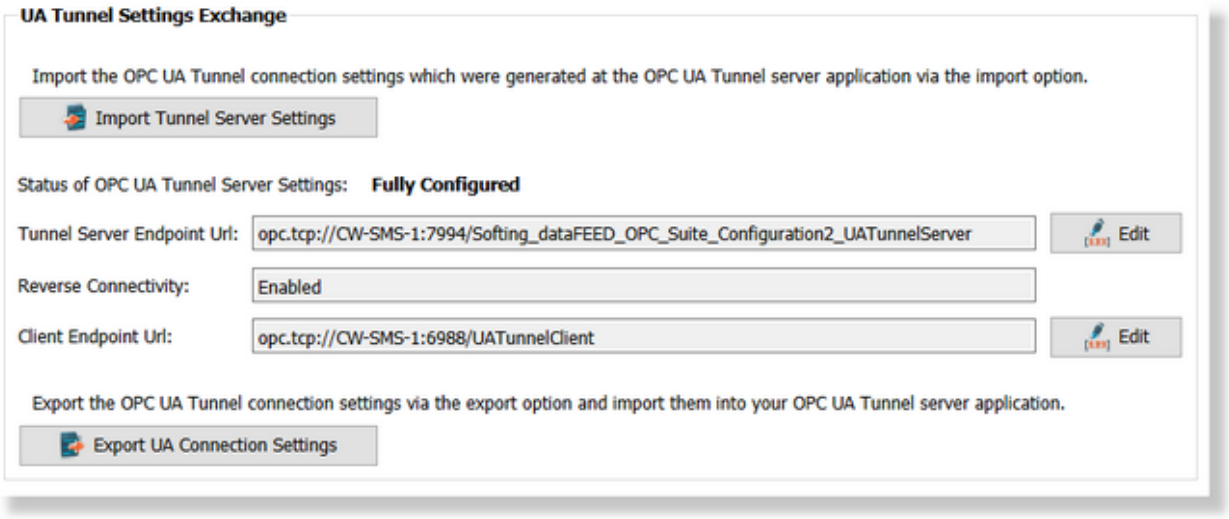
Immediately after the import operation is executed, the current connection settings are overwritten with the imported settings, the connection is validation and the connection definition is allowed to continue.

At this step, the current OPC UA Tunnel client imported the OPC UA Tunnel connection settings and the OPC UA Tunnel server become trusted for it.

Attention: At this step, the imported OPC UA Tunnel sever is trusted by OPC UA Tunnel client, still the opposite relation is not already done (the OPC UA Tunnel client is not already trusted by OPC UA Tunnel server). This operation requires to export the current connection settings and to import them into the OPC UA Tunnel Endpoint.



In case of a reverse connection, the imported settings are also reflecting the Client Endpoint URL settings needed for enabling the Reverse Connectivity.



The *Tunnel Server Endpoint URL* and *Client Endpoint URL* can be edited by pressing the **Edit** button(s). This allows the Endpoint URL to be modified (usually this operation is needed in order to replace a HostName with the corresponding IP Address when the DHCP Server is not available).

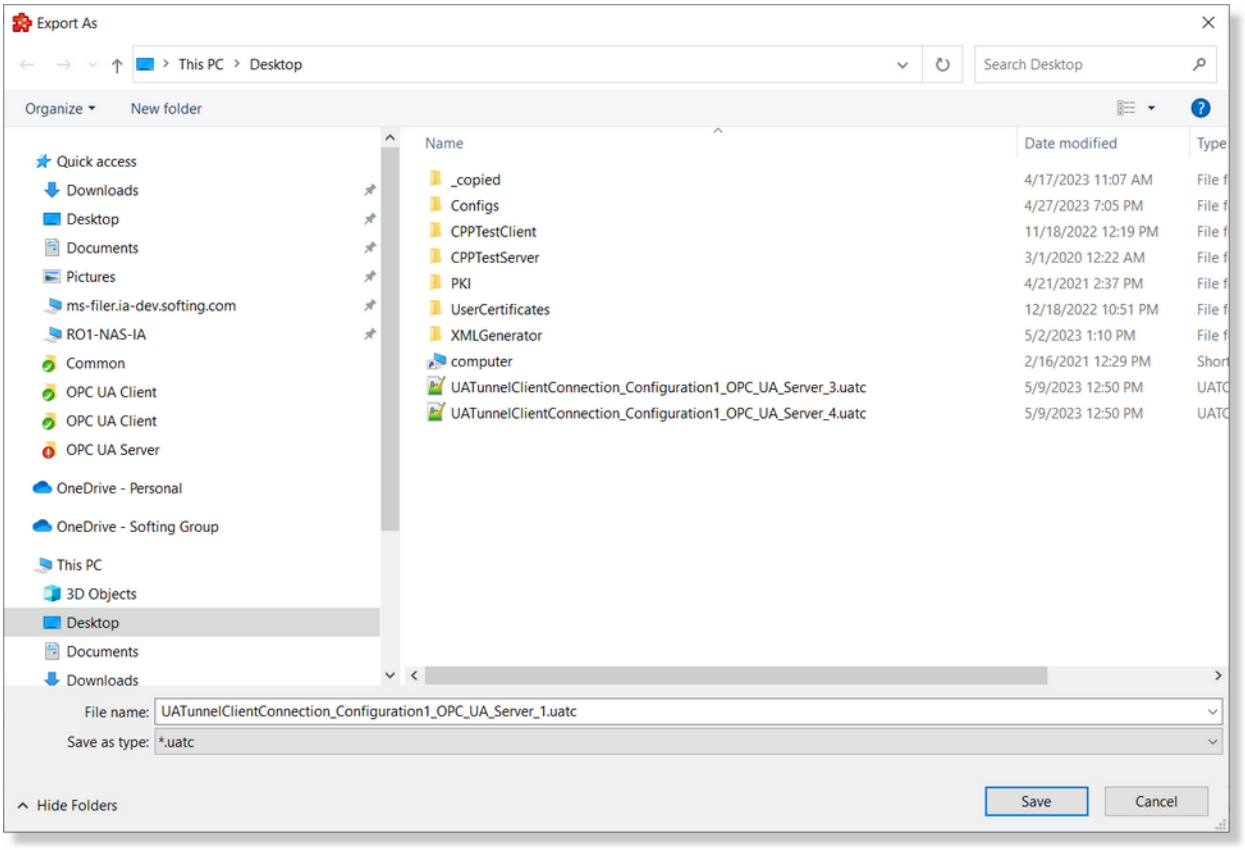
Attention: The edited Endpoint URL should match with the Endpoint URL of the server which is intended being connected.

After the OPC UA Tunnel server settings are imported the *Status of OPC UA Tunnel Server Settings* is changed to **Fully Configured**. Since this moment the connection is completely defined on the OPC UA Tunnel client.

The OPC UA Tunnel Connection settings and the corresponding Application Certificate should be exported in order to be later imported on the OPC UA Tunnel server side. This operation is mandatory to finalize the trust relation between OPC UA Tunnel client (data source) and OPC UA Tunnel server (data destination).

The export operation can be perform by pressing the  *Export UA Connection Settings* button. This will allow to select the location and the file which will be exported.

© Softing Industrial Automation GmbH

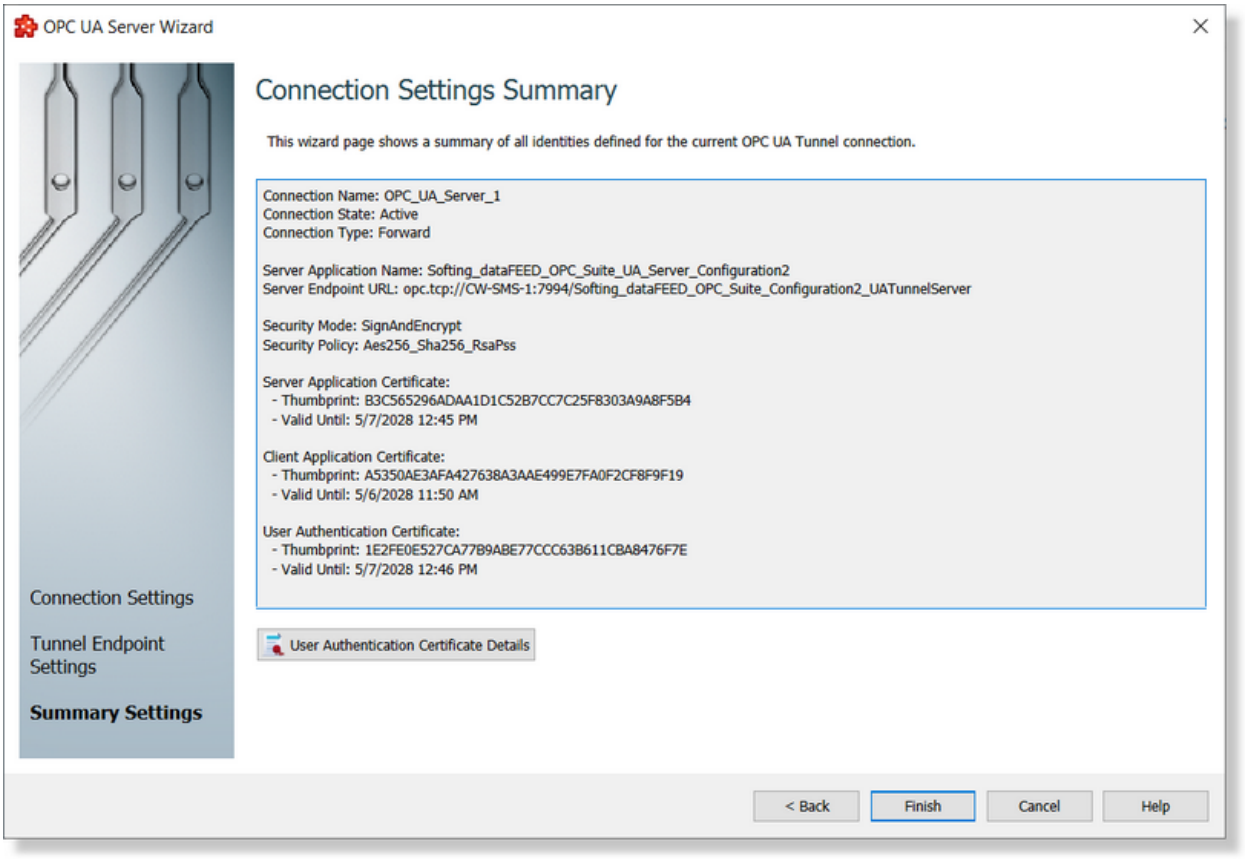


Finalizing the export operation, allows the user to import this file on the OPC UA Tunnel server in order to complete the connection definition.

Note: The import/export operation can be executed on local hard drive but also over the network share drive (if access to it is allowed for the current user).

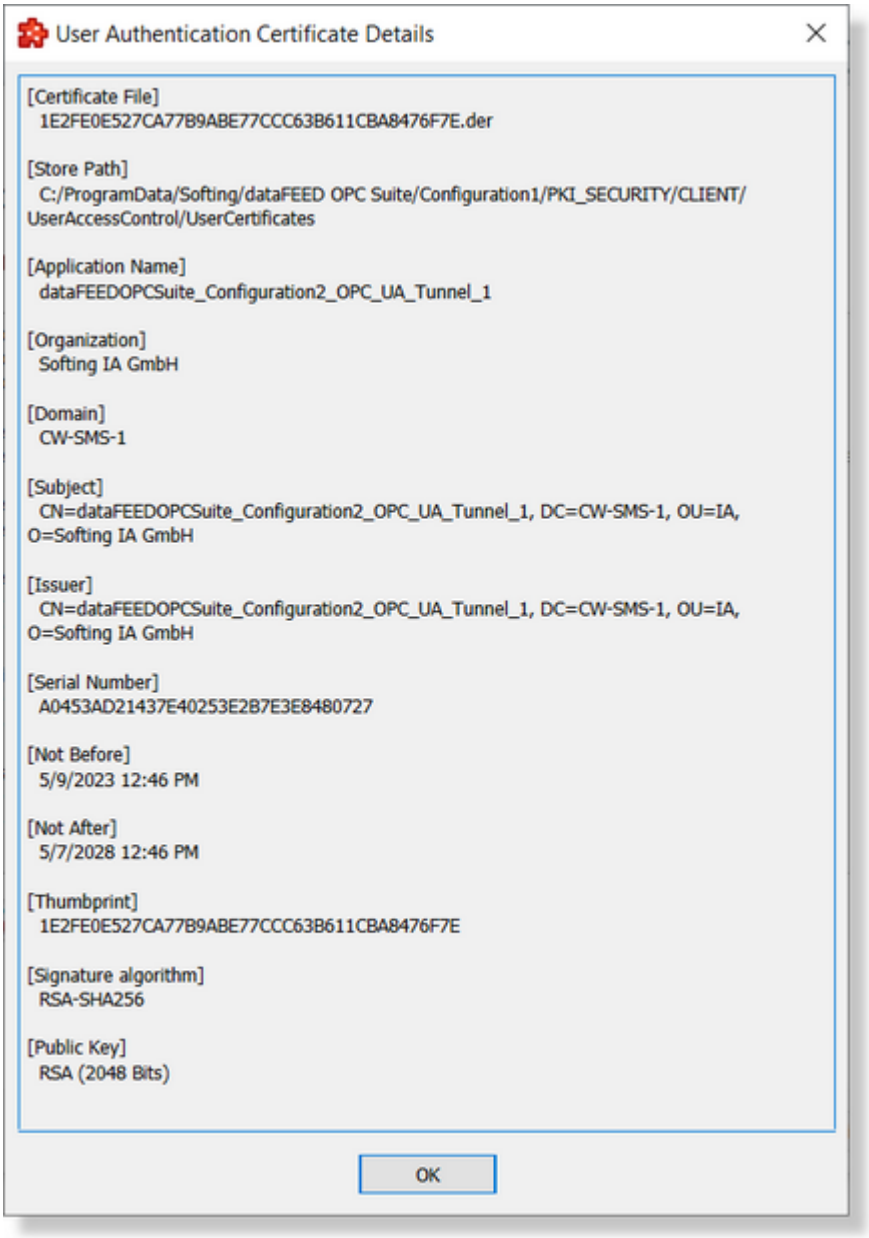
<<< UA Server Connection Wizard - Connection Settings UA Server Connection Wizard - Connection Settings Summary >>>

On this wizard page, it is shown a summary of the configured connection settings (this includes the imported settings) for a specified OPC UA Tunnel connection. The detailed settings are available on the page on read only format.



Also, it is allowed to visualize the User Authentication Certificates details by pressing the  "User Authentication Certificate Details" button. This operation opens a dialog containing all the details of actual user authentication certificate.

© Softing Industrial Automation GmbH



Pressing "**Finish**" button is finalizing the definition of an OPC UA Tunnel connection and the connection became visible on the Data Source OPC UA Server / OPC UA Tunnel - External OPC Ua Servers table. Canceling the wizard is discarding all the defined parameters of actual connection.

<<< UA Server Connection Wizard - OPC UA Tunnel Settings Exchange

7.3.1.7.6 Appendix

The exposed address space for each data source connection is modeled into the aggregated server under a dedicated node created based on connection name (or prefix).

The dataFEED OPC Suite application is supported only variables based on native (simple) data-types. Variables created based on complex data types on the connected server being completely ignored by the application - being not available on the active data destinations.

Support for Abstract Variables

The variable with abstract data types (variable on connected server with a ***"HasTypeDefinition"*** reference to "***BaseDataVariableType***") are supported by the application, but a set of limitations or conversions are applied:

- The **scalars** are fully supported, exposed over:
  - OPC UA interface (UA Server) - no changes applied than the variable from the connected server.
  - Classical OPC interface (DA/XML-DA and Tunnel Servers) - items are created based on unknown data type (not variant) - the only limitation being that over the write service the user should specify the data type of the value which is intend to be written.
  - All the other data destination are supporting this variables - the only limitation coming from the destinations which are not supporting variable data types (values which are dynamically changing their data type).
  - Data processors (Exchange and Classical Store&Forward) - both functionality required a constant data type - while the values are not changing dynamically its data type - the functionality is working as expected; in case that the values are dynamically change the data type then a conversion to the "initial" data type (the data type received at first data-change) is tried being operated - in case of failure the entire action (exchange or store) is invalidated and not executed.
- The **arrays** are supported too, but a set of limitations and conversions are applied:
  - arrays containing elements with not initialized values (data type being not set or set to null) are not supported - a received value in this format will be ignored and a "StatusCode_BadNotSupported" status code will be delivered to the corresponding data destination.
  - arrays containing elements with different data types are tried being converted to a common data type - if the conversion is possible then it is delivered to the data destination the converted variant to the common data type, otherwise "StatusCode_BadNotSupported" status code will be delivered.

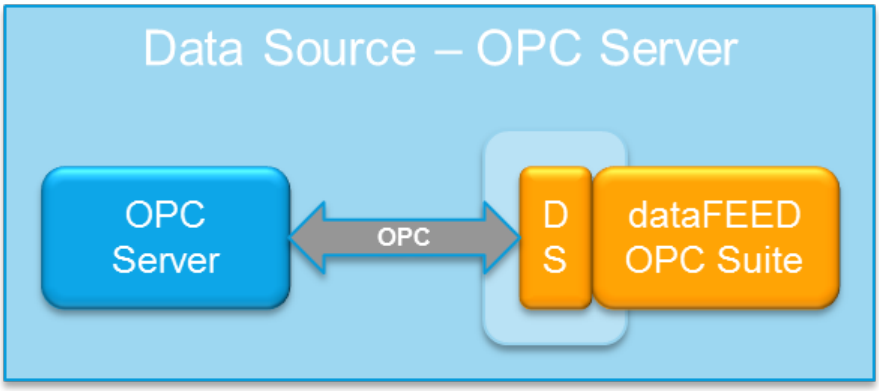
    Attention: it is possible to lose some data or the precision can be affected by the applied conversion.

  - usage of arrays containing elements with different data types is not possible on Data Processor functionality (Exchange and Classical Store&Forward).
  - writing arrays of variants required to specify the data-type of each element - these can be different and dynamically changed between consecutive operations.
  - writing arrays of variants base on a specified index range (basically being performed a partially write operation) is not supported by the application.

7.3.1.8 OPC Server

This section of the Softing dataFEED OPC Suite Online Help describes how to configure the OPC server data source functionality of Softing dataFEED OPC Suite.

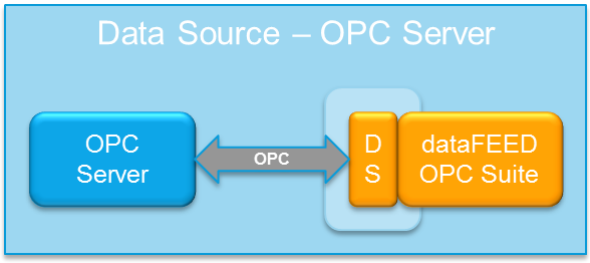
© Softing Industrial Automation GmbH



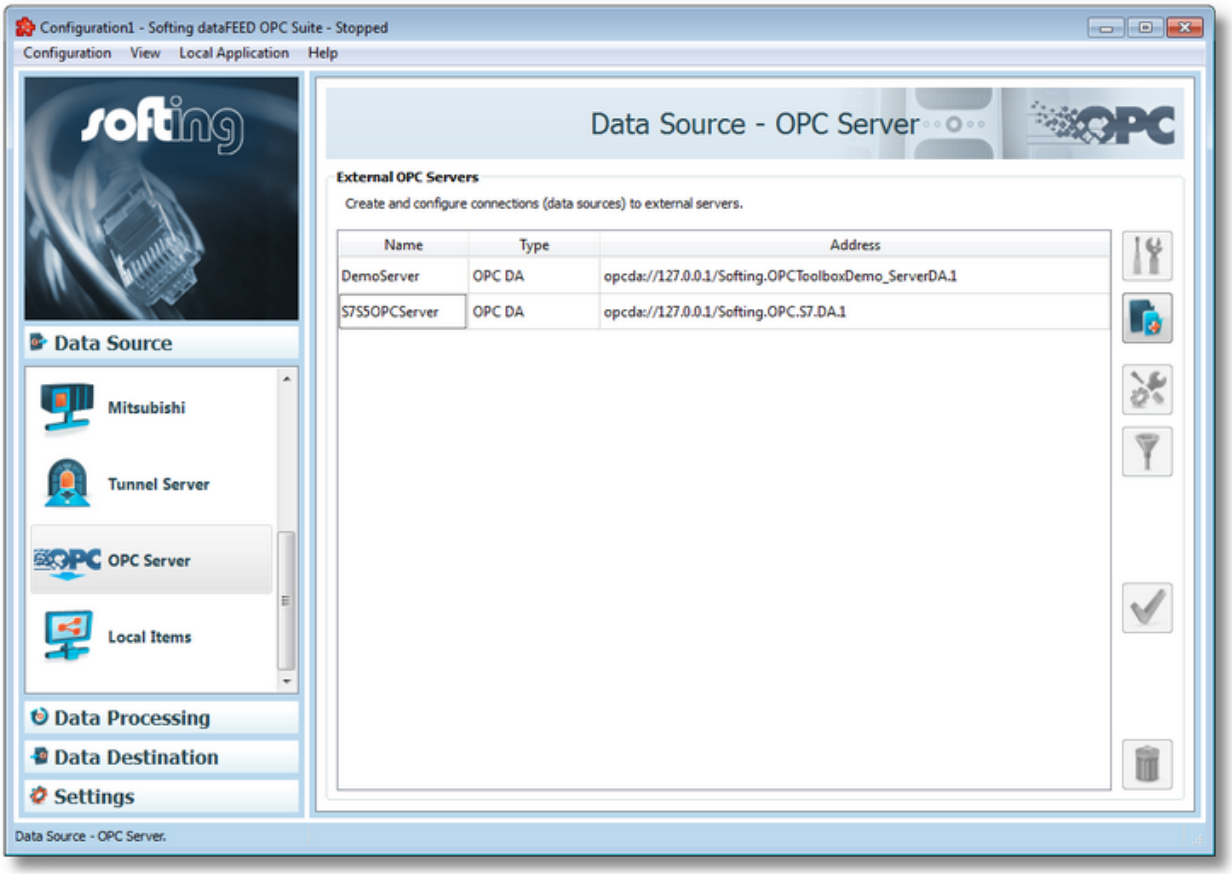
Softing dataFEED OPC Suite can access and capture data from any OPC DA or OPC XML-DA server. Data can be then processed, saved, analyzed or transferred by using the *Data Processing* and/or *Data Destination* functionalities of dataFEED OPC Suite. The way dataFEED OPC Suite handles this is by opening individual OPC server connections to the external OPC servers. The namespace elements of the servers to be addressed are integrated into the namespace of the local dataFEED OPC Suite application.

To see the necessary steps to configure an OPC server connection, start at the Configuration: Data Source - OPC Server page.

7.3.1.8.1 Configuration



To start configuring the connection to an external OPC Server select *OPC Server* from the *Data Source* group in the navigator on the left. The Data Source - OPC Server main user interface is displayed. The main element of this page is the *External OPC Servers* list. Each entry shown in the list of the configuration section represents a connection to an external OPC server:



Note: The list is only shown and editable if a configuration has been loaded.

Each OPC server connection has a unique name within the configuration, which is given in the *Name* column.

The name of the server will be shown in the namespace tree in the Softing dataFEED OPC Suite Configurator and in the local dataFEED OPC Suite application, and its entire namespace will be under this name.

Note: The name can also be an empty string, but it has to be unique. If the name of the OPC server is empty, the dataFEED OPC Suite Configurator tree will expose the OPC server's namespace under a node named <*empty*>, but the local dataFEED OPC Suite application will expose it to other clients directly in the root.

The type of each OPC server is displayed in the second column, and the server *Address* in the third. The type of each server can be either OPC DA Server or OPC AE Server.

Attention: For correct OPC DA/AE functionality (DCOM based protocols) on PCs running Windows 10, please ensure that the 1803 OS version is used (build 17134.48).

Operation

© Softing Industrial Automation GmbH

From the *OPC Server* main user interface certain operations can be performed over the selected OPC server by using the buttons on the right, by accessing the context sensitive menu or by pressing the keyboard shortcuts. The following actions are available:

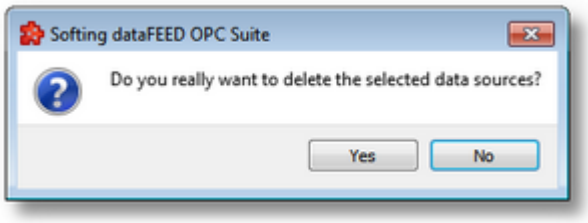
- Change the properties of the selected data source by using the *Properties* button  , pressing the keyboard shortcut CTRL+P or selecting *Properties* from the context menu.
- Add a new data source by using the *Add* button  , pressing the keyboard shortcut CTRL+A or selecting *Add* from the context menu.
- Change advanced settings for the selected data source by using the *Advanced settings* button  or selecting *Advanced* from the context menu.
- Filter data items for the selected data source by using the Filter button  , pressing the keyboard shortcut CTRL+F or selecting Filter from the context menu.
- Perform a connection test for the selected data source by using the *Connection test* button  , pressing the keyboard shortcut CTRL+T or selecting *Test* from the context menu.
- Delete the selected data source by using the *Delete* button  , pressing the DEL / DELETE key on the keyboard or selecting *Delete* from the context menu.

Note: If no OPC server connection is selected, the *Add a new data source* button is the only one available.

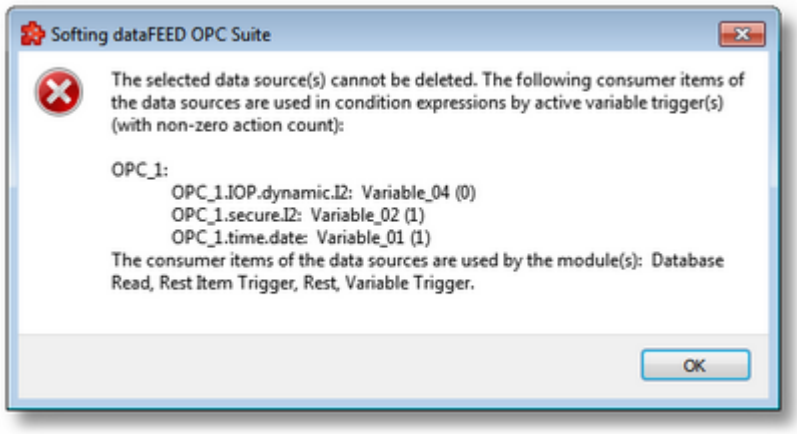
The *Delete* button  will remove the selected server connections from the list.

The user can select more than one server connection with the help of the CTRL or SHIFT keys.

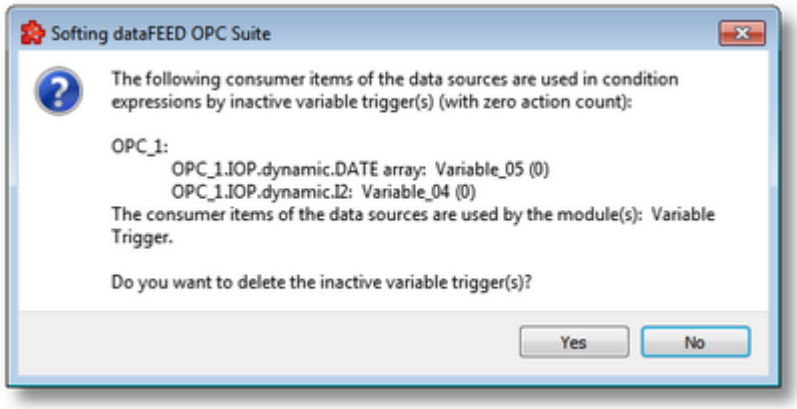
A dialog will then prompt the user to confirm the deletion.



A dialog will inform the user that the data sources cannot be deleted, if consumer items and/or system items of the selected data sources are used by active variable triggers or other modules.



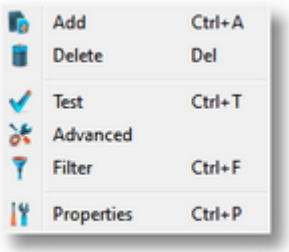
A dialog will prompt the user to confirm deletion of the inactive variable triggers, if consumer items and/or system items of the selected data sources are used only by inactive variable triggers. A following delete command will remove the selected server connections from the list.



Attention: An OPC server connection cannot be deleted if at least one data item defined in its namespace is used in a Processing or Data Destination action.

All these actions are also provided in the context menu of the External OPC Server list. Which actions are available depends on the current selection in the list.

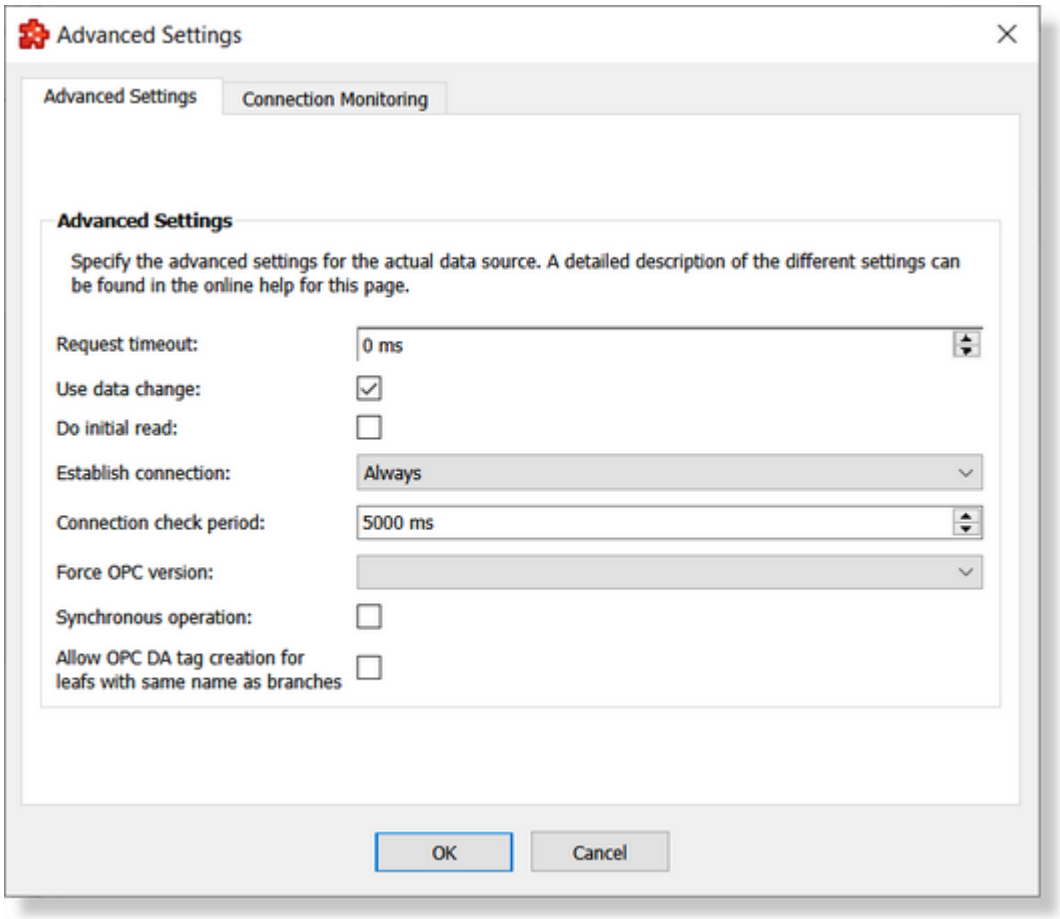
An example of the context menu can be seen below:

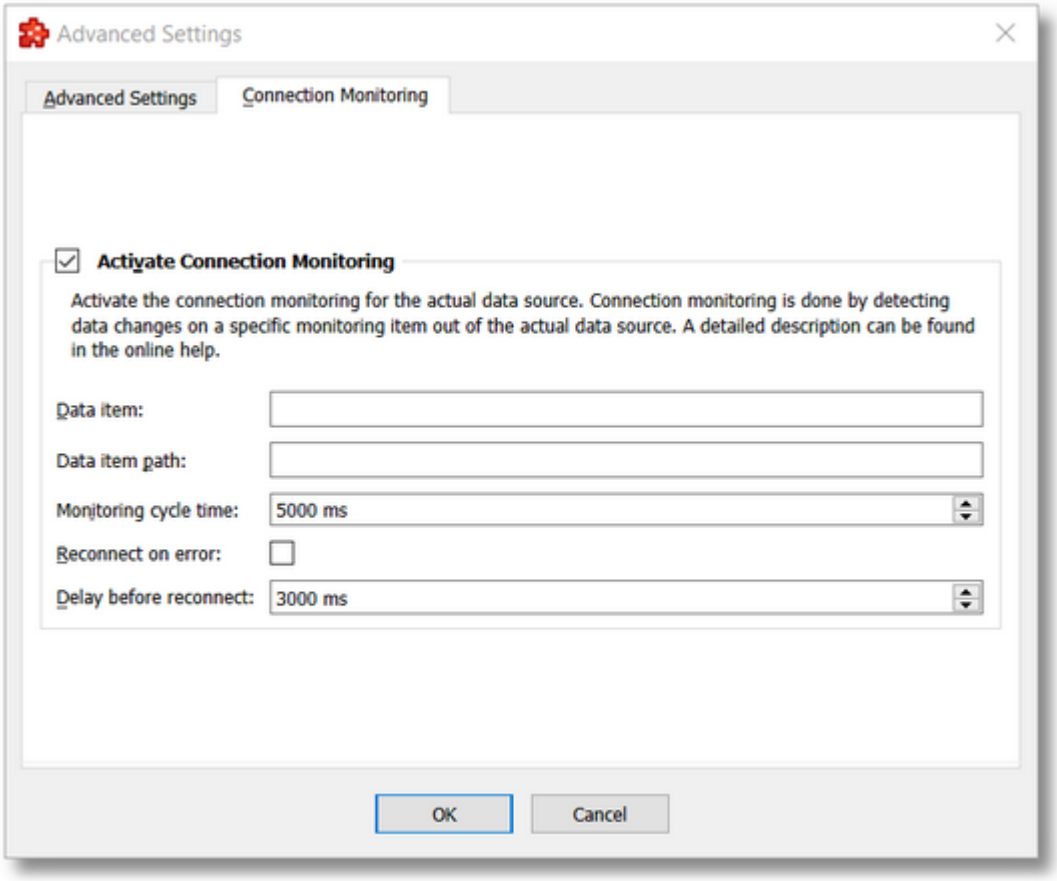


© Softing Industrial Automation GmbH

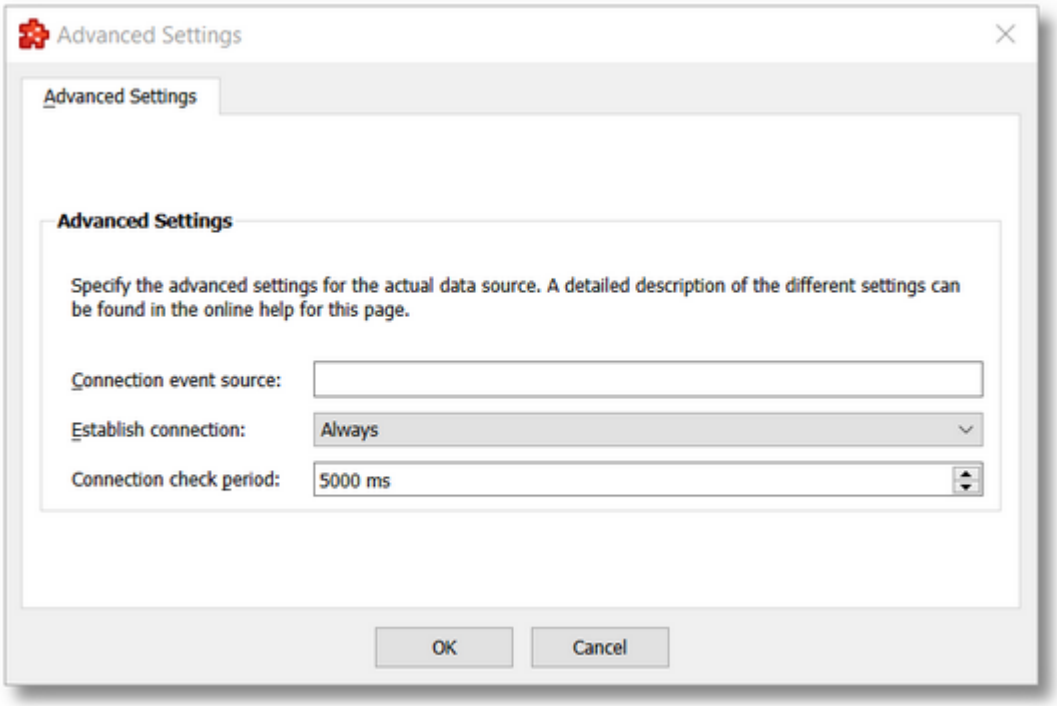
7.3.1.8.2 Advanced Settings

The *Advanced Settings* button shows a dialog that allows the user to modify some advanced settings for the selected server connection. Depending on the external OPC server connection type, different configuration options are available. For an *OPC DA* and *XML-DA* server connection the configuration options are organized in two different tabs, *Advanced Settings* and *Connection Monitoring*, as shown below:





For an *OPC AE* server connection the *Advanced Settings* dialog looks like this:



© Softing Industrial Automation GmbH

Note: For *OPC AE* server connections, there is no *Connection Monitoring* tab in the *Advanced Settings* dialog.

The following tables depict all the configuration options, each with a brief description.

Configuration options for *Advanced Settings*:

Attribute	Description
Request timeout	The maximum time in milliseconds that an I/O request to the Tunnel DA server may take. After this time, the request is canceled in the local dataFEED OPC Suite application.
Use data change	Checked if the OPC notification mechanism should be used for an OPC DA server connection. If unchecked, the values should be updated by synchronously reading the item cache.
Do initial read	Checked if an initial read should be performed on subscription to an OPC DA server.
Connection Event Source	The name of the Event Source to be used for signaling of connection failures at the OPC AE Server.
Establish connection	Specifies when the connection to the OPC server should be established. *Always* means the connection to the OPC server should be established at all times. *On Demand* means the connection will be established only as needed.
Connection check period	The time interval in milliseconds at which the connection to the OPC server is to be regularly checked. If no connection to the server exists, all items of this server receive the "communication failure" quality.
Force OPC version	This option allows selecting a special OPC specification interface to be used for communication with the OPC DA server. Possible values are *DA V1* or *DA V2*.
Activate sync operations	Checked if synchronous operations shall be used.
Allow OPC DA tag creation for leafs with same name as branches	Checked if it is is allowed to create OPC DA tags for leafs that have the same name as existing branches in the address space.

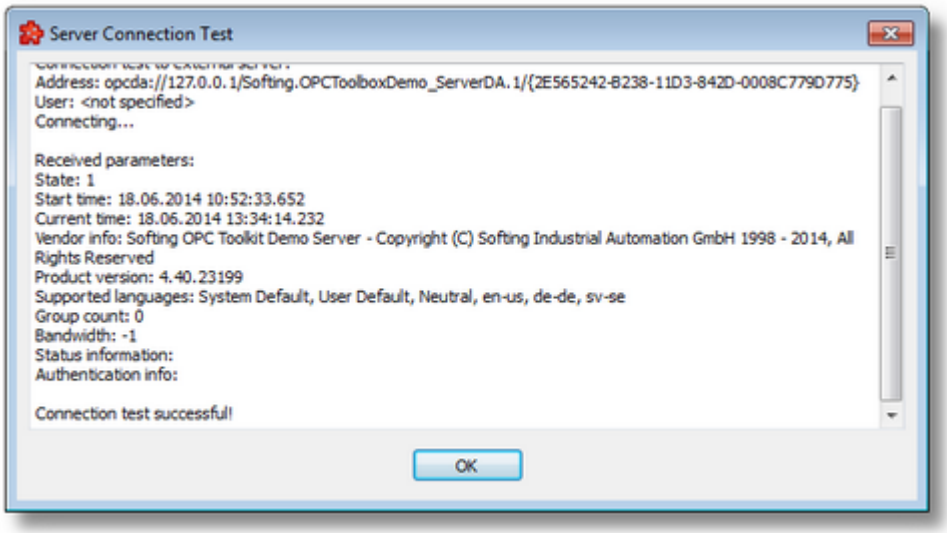
Configuration options for *Connection Monitoring*:

Attribute	Description

Activate Connection Monitoring	By checking this group box, the connection monitoring is enabled and the other configuration options become editable.
Data item	The data item which shall be used for the heartbeat monitoring. This data item will be monitored for all data changes. If an empty string is specified here the heartbeat connection monitoring will be turned off.
Data item path	The data item path (relevant for XML-DA only) corresponding to the data item which shall be used for the heartbeat monitoring.
Monitoring cycle time	Specifies the time interval in milliseconds used by the connection monitoring. The heartbeat data item has to change at least once within this time interval.
Reconnect on error	Checked if the server should restart if the heartbeat item reports there is no connection to the server.
Delay before reconnect	The maximum delay in milliseconds which is allowed for the heartbeat item.

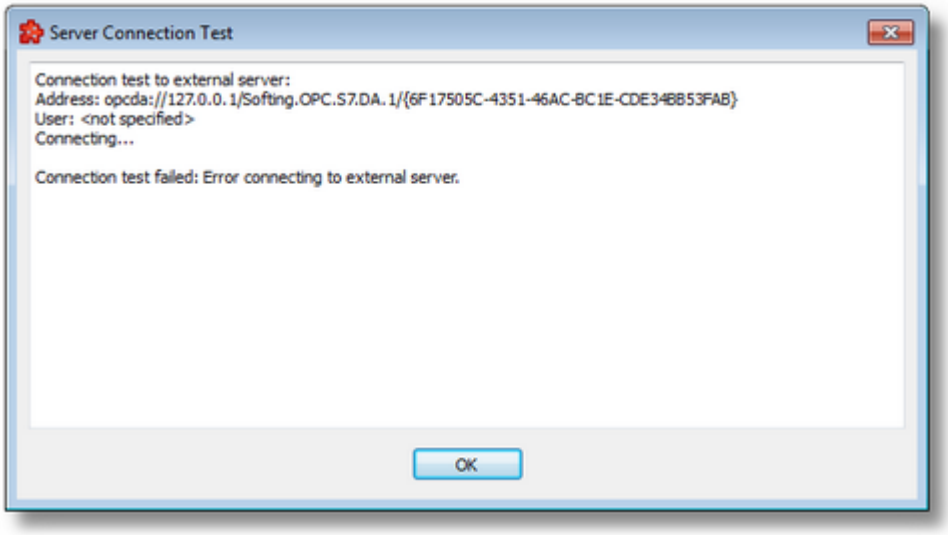
7.3.1.8.3 Connection test

The *Test Connection* button  checks the OPC server connection. The dataFEED OPC Suite Configurator tries to connect to the external OPC server. A dialog is shown with the test results. If the connection was successful, various information about the server is also displayed:



If no connection could be established, the following dialog box is displayed:

© Softing Industrial Automation GmbH

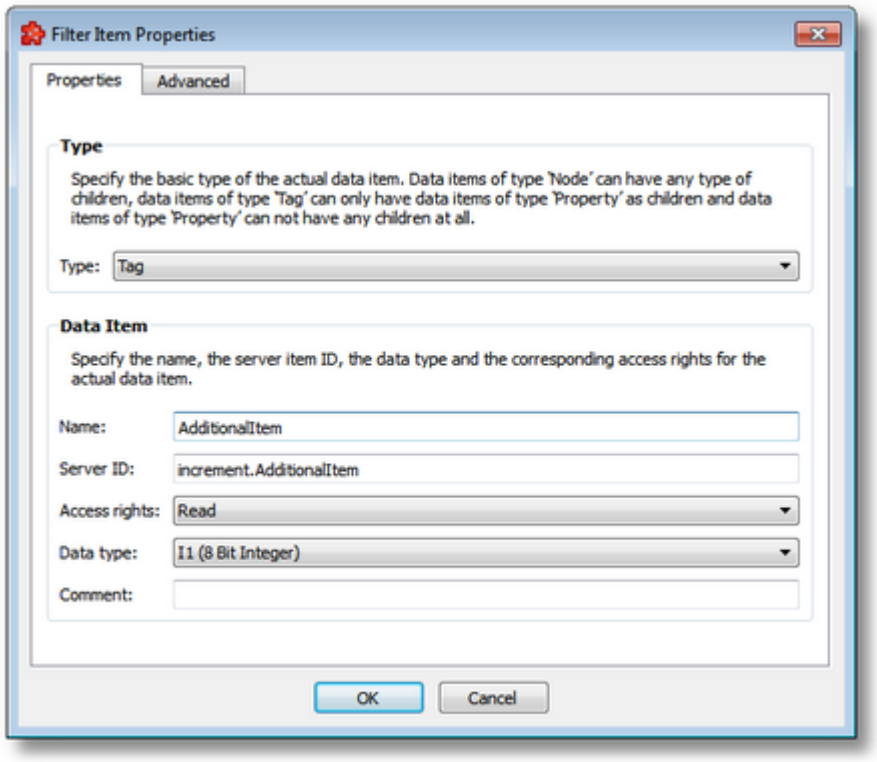


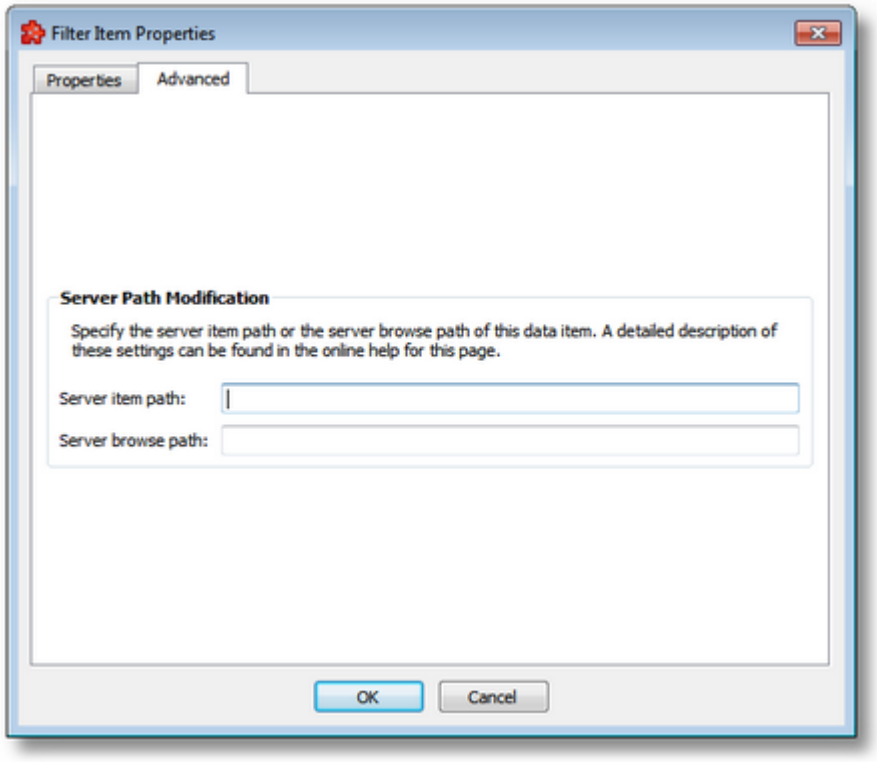
7.3.1.8.4 Filter Item Properties

The properties of the *Filter Item* can be set when adding Filter Items and can be modified when editing previously added Filter Items.

Add a New Filter Item

The add operation is triggered by selecting the parent node and then pressing the *Add a new item* button  , using the shortcut or selecting the *Add* action from the context menu. When adding a new Filter Item the following two-tabbed dialog is displayed:





Note: When you select items of type *Node* for which no *Access Rights* parameter has been set (empty value), the only configurable parameters are the *Name* and the *Access Rights*.

The available configuration options in this dialog are described in the table below:

© Softing Industrial Automation GmbH

Attribute	Description
Type	The type of the data item. The possible types are: *Node*, *Tag* and *Property*.
Name	The OPC item name identifying the corresponding data item inside the dataFEED OPC Suite namespace.
Server ID	The full OPC item path for the data item inside the dataFEED OPC Suite namespace.
Access rights	The access rights for the data item. The possible values are: *Read*, *Write*, *Read and Write*.
Data type	The data type of the data item. The possible values are described in the table below.
Comment	An optional comment/note can be attached to the current data item by using this field.
Server item path	The OPC path for the data item on the external OPC server.
Server browse path	The OPC browse path for the data item, describing the individual browse steps for the data item in the namespace of the external OPC server.

Note: Items of type *Node* may have child items of any existing type.

Note: Items of type *Tag* may have child items of type *Property*.

Note: Items of type *Property* cannot have any children. Therefore, Add is not available for properties.

The OPC filter data items of the namespace may be of any data type covered by the OPC Specifications. The following table describes the available data types mapped to the corresponding VARIANT types from the OPC Specification.

Data type	VARIANT type	Description
I1	*VT_I1*	8 bit signed integer.
UI1	*VT_UI1*	8 bit unsigned integer.
I2	*VT_I2*	16 bit signed integer.
UI2	*VT_UI2*	16 bit unsigned integer.

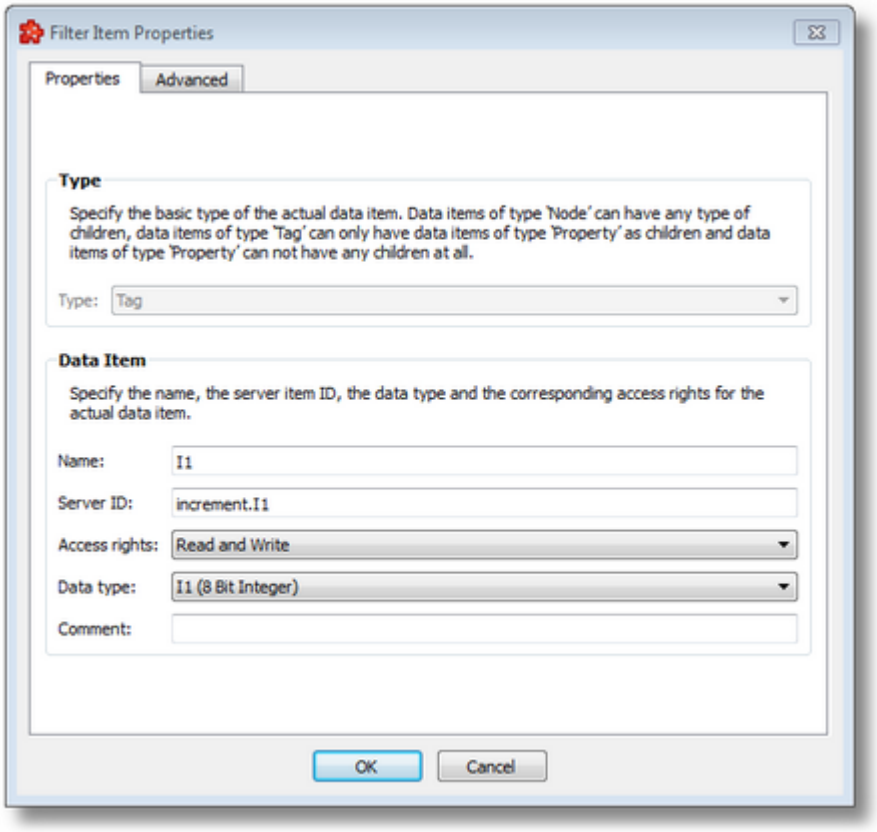
I4	*VT_I4*	32 bit signed integer.
UI4	*VT_UI4*	32 bit unsigned integer.
I8	*VT_I8*	64 bit signed integer.
UI8	*VT_UI8*	64 bit unsigned integer.
R4	*VT_R4*	32 bit floating point.
R8	*VT_R8*	64 bit floating point.
BOOL	*VT_BOOL*	Boolean.
STRING	*VT_BSTR*	String.
DATETIME	*VT_DATE*	Date and time.
DECIMAL	*VT_CY*	Decimal.
ANY	*VT_VARIANT*	Any data type.

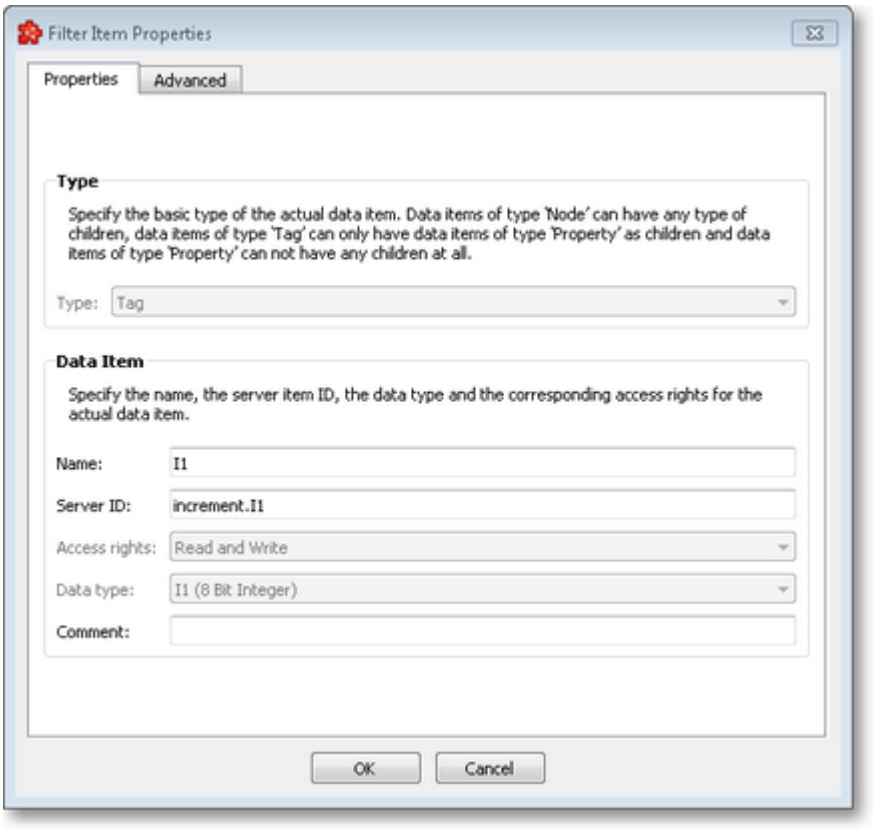
Note: Arrays of fixed and variable length are also supported for these data types.

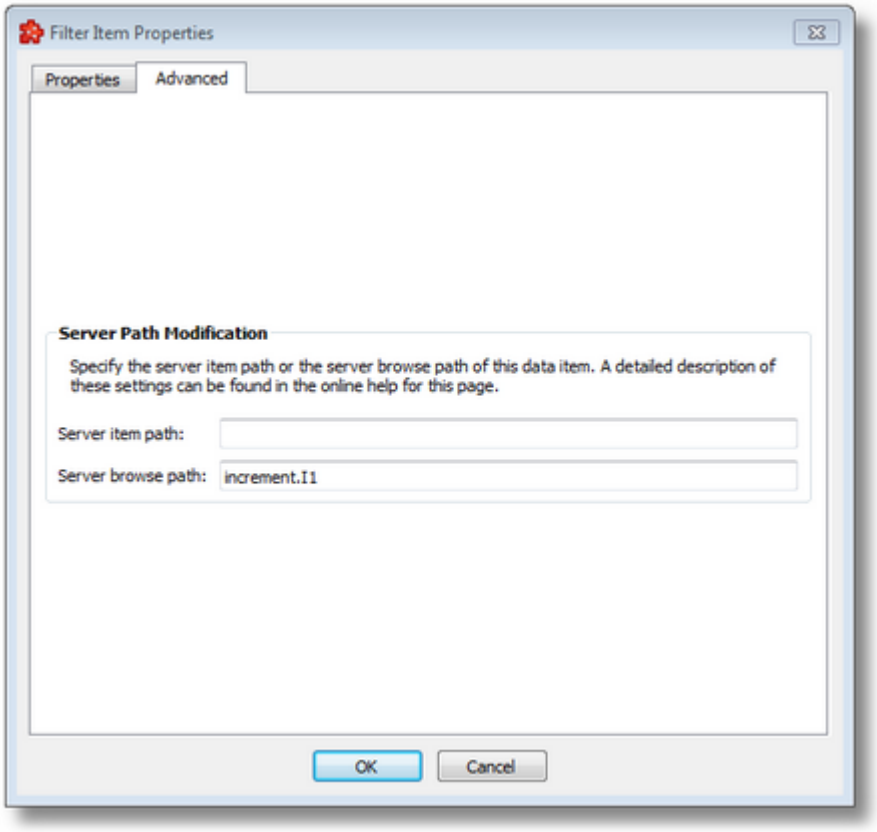
Modify a Filter Item

The modify operation is triggered by selecting the node to be modified and then pressing the *Properties* button  , using the shortcut or selecting the *Properties* action from the context menu. When modifying an existing Filter Item the following two-tabbed dialog is displayed:

© Softing Industrial Automation GmbH


Filter Item Properties
Properties
Advanced
Type
Specify the basic type of the actual data item. Data items of type 'Node' can have any type of children, data items of type 'Tag' can only have data items of type 'Property' as children and data items of type 'Property' can not have any children at all.
Type: Tag
Data Item
Specify the name, the server item ID, the data type and the corresponding access rights for the actual data item.
Name: I1
Server ID: increment.I1
Access rights: Read and Write
Data type: I1 (8 Bit Integer)
Comment:
OK
Cancel


Filter Item Properties
Properties
Advanced
Type
Specify the basic type of the actual data item. Data items of type 'Node' can have any type of children, data items of type 'Tag' can only have data items of type 'Property' as children and data items of type 'Property' can not have any children at all.
Type: Tag
Data Item
Specify the name, the server item ID, the data type and the corresponding access rights for the actual data item.
Name: I1
Server ID: increment.I1
Access rights: Read and Write
Data type: I1 (8 Bit Integer)
Comment:
OK
Cancel



The configuration options are the same as in the above described *Add a New Filter Item* dialog.

Note: The *Type* of the data item cannot be changed anymore.

Note: The *Access rights* and *Data type* parameters of a Filter item are disabled and cannot be changed if the item is being used by other consumer add-ons.

Note: The *Modify* action is also triggered by double-clicking on tree items without children.

7.3.1.8.5 Filter File

Each of the OPC server connections configured to use a filter file will produce a file containing the definition of the filter data items when the configuration is saved.

The files are stored in the *Filter* subfolder of the configuration folder. The filter file is named after the server connection name. The filter files are plain text files, encoded in UTF-8 (thus providing full UNICODE compatibility) and they have a line-oriented structure. The elements of a line are tab-separated. For each filter data item a line is added to the file.

The following line types can be found in the OPC server connection filter file:

- Filter item line, defining a filter data item as described in the table below.

© Softing Industrial Automation GmbH

- Empty line.
- Comment line, starting with #; such a line is ignored during evaluation.

A filter item line has the following format:

```
<TABSofting server ID<TAB>Softing item ID<TAB>Server item ID<TAB>Server item path<TAB
```

The different elements of the line are described in the following table:

Element	Description
Softing server ID	Not used. Should be always an empty string.
dataFEED OPC Suite item ID	The OPC item name identifying the corresponding data item inside the dataFEED OPC Suite namespace.
Server item ID	The OPC item ID for the data item in the external OPC server namespace.
Server item path	The OPC path for the data item on the external OPC server.
Server browse path	The OPC browse path for the data item, describing the individual browse steps for the data item in the namespace of the external OPC server.
Data type	The data type of the data item. The possible values are described in the Filter Item Properties page.
Access rights	The access rights for the data item. The possible values are: r, w, rw corresponding to read, write, read and write, respectively.
Object type	The type of the data item. The possible types are: Node, Tag and Property.
Comment	An optional comment/note attached to the current data item.

Example:

```
#<TAB>Softing server ID<TAB>Softing item ID<TAB>Server item ID<TAB>Server item path<T
#
<TAB><TAB>maths.automatic angle change<TAB>maths.automatic angle change<TAB><TAB>math
<TAB><TAB>maths.automatic change period<TAB>maths.automatic change period<TAB><TAB>ma
<TAB><TAB>maths.cos<TAB>maths.cos<TAB><TAB>maths.cos<TAB>R8<TAB>r<TAB>Tag<TAB>Cosinus
<TAB><TAB>maths.cos|102<TAB>maths.cos#102<TAB><TAB>maths.cos|102<TAB>R8<TAB>rw<TAB>Pr
<TAB><TAB>maths.cos|103<TAB>maths.cos#103<TAB><TAB>maths.cos|103<TAB>R8<TAB>rw<TAB>Pr
<TAB><TAB>maths.cos|6020<TAB>maths.cos#6020<TAB><TAB>maths.cos|6020<TAB>UI2<TAB>rw<TA
<TAB><TAB>maths.sin<TAB>maths.sin<TAB><TAB>maths.sin<TAB>R8<TAB>r<TAB>Tag<TAB>Sinus d
<TAB><TAB>maths.sin|102<TAB>maths.sin#102<TAB><TAB>maths.sin|102<TAB>R8<TAB>r<TAB>Pro
<TAB><TAB>maths.sin|103<TAB>maths.sin#103<TAB><TAB>maths.sin|103<TAB>R8<TAB>r<TAB>Pro
```

```
<TAB><TAB>maths.tan<TAB>maths.tan<TAB><TAB>maths.tan<TAB>R8<TAB>r<TAB>Tag<TAB>
<TAB><TAB>maths.tan|6020<TAB>maths.tan#6020<TAB><TAB>maths.tan|6020<TAB>UI2<TAB>rw<TA
```

7.3.1.8.6 OPC Server Wizard

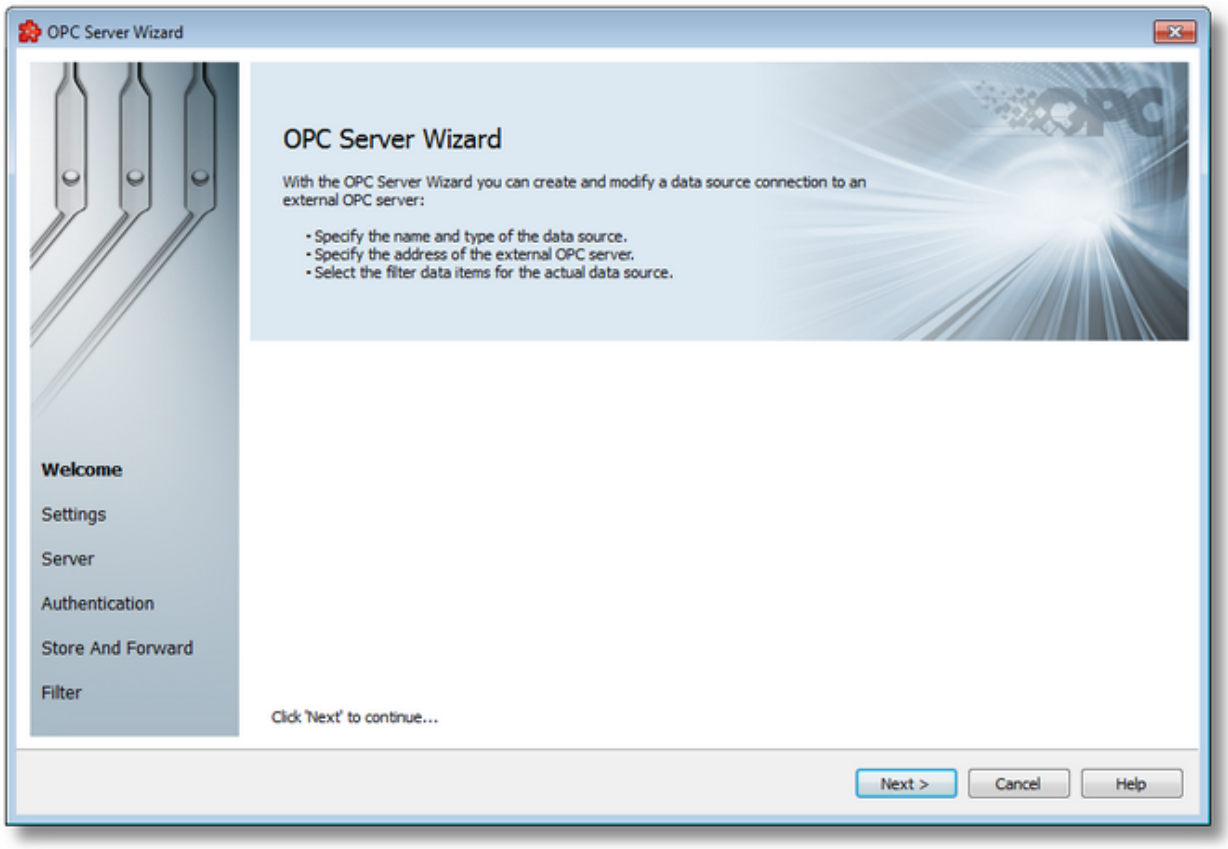
▼ With the OPC Server Wizard you can create and modify a data source connection to an external OPC server:

- Obtain general information about the OPC Server Wizard @ OPC Server Wizard - Welcome
- Specify the name and the type of the OPC Server connection @ OPC Server Wizard - Settings
- Select the address of the external OPC server @ OPC Server Wizard - Server
- Specify the user name and the password for the external OPC server @ OPC Server Wizard - Authentication
- Decide if the connection should be a Store And Forward connection @ OPC Server Wizard - Store And Forward
- Activate or deactivate the filter functionality for the connection @ OPC Server Wizard - Filter
- Select the corresponding filter data items @ OPC Server Wizard - Filter Items

With the OPC Server Wizard you can create and modify a data source connection to an external OPC server:

- Specify the name and type of the external OPC server connection.
- Specify the address of the external OPC server.
- Select the filter data items for the selected data source.

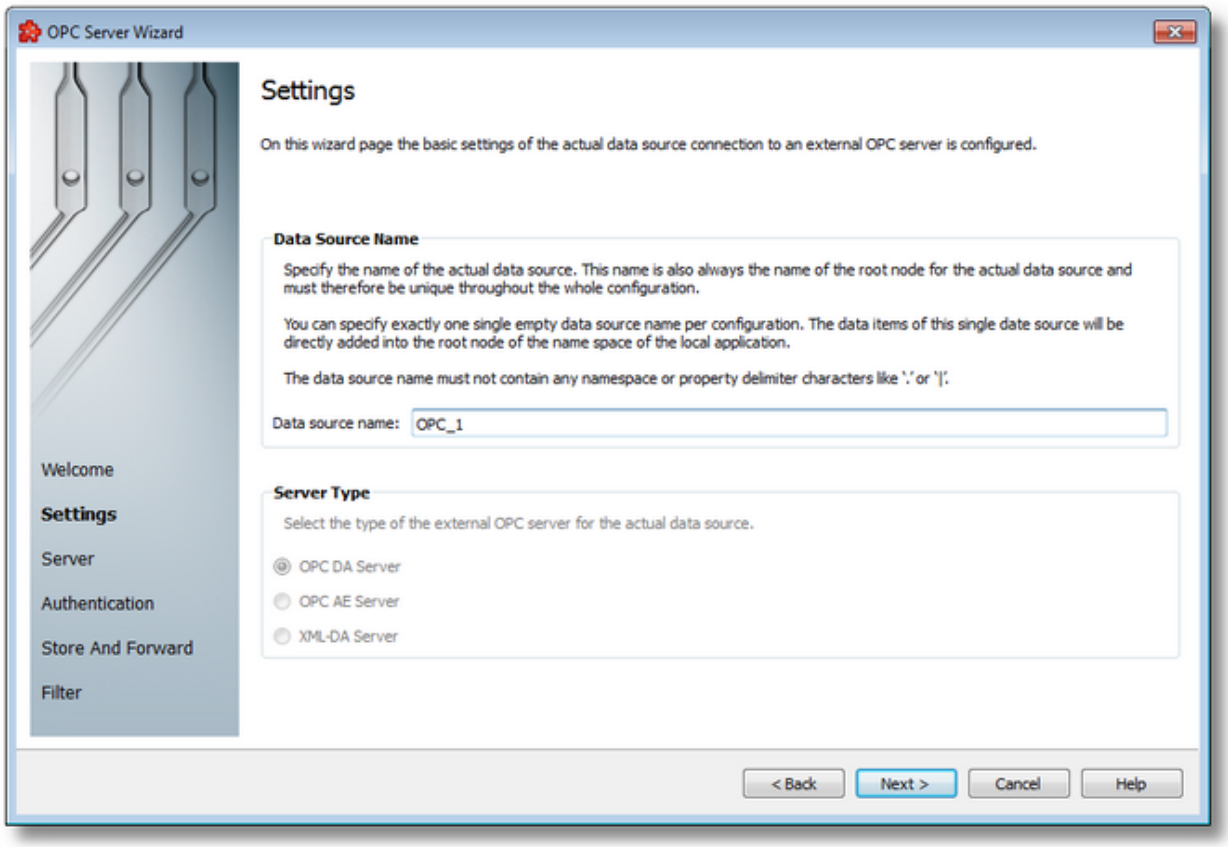
© Softing Industrial Automation GmbH



Press the *Next* button to proceed with the wizard or abort the operation with the *Cancel* button.

OPC Server Wizard - Settings >>>

On this wizard page the basic settings of the data source connection to an external OPC server are configured:



Data Source Name

In the *Data Source Name* field you can specify the name of the data source. This name is also always used as the name of the root node for the specified data source. Therefore, it has to be unique throughout the whole configuration.

Attention: The data source name must be unique throughout the whole configuration and must not contain any namespace or property delimiter characters like '.' or '|'.

Server Type

In the *Server Type* field you can select the type of the external OPC server for the specified data source. The following OPC Server types are possible:

- *OPC DA Server*: OPC Data Access Server.
- *OPC AE Server*: OPC Alarms and Events Server.
- *XML-DA Server*: OPC XML Data Access Server.

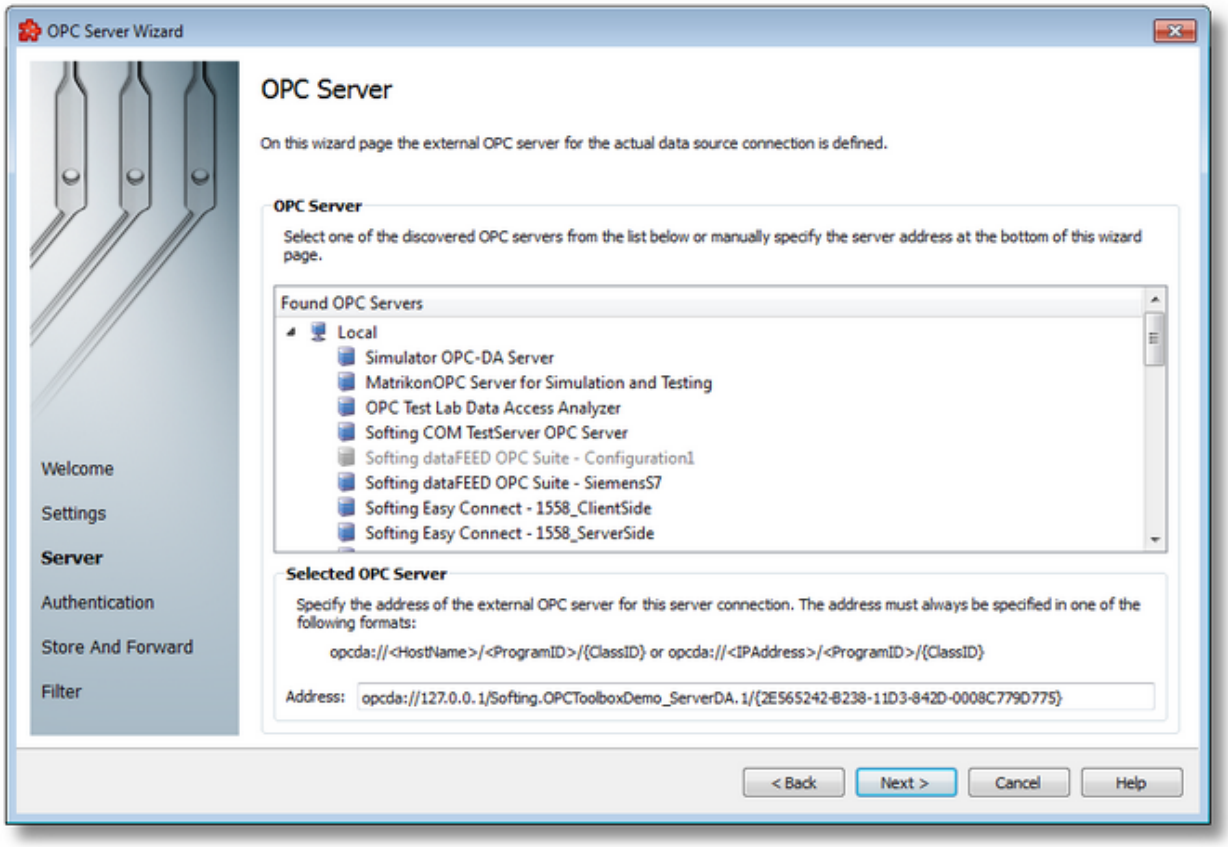
You can specify one empty data source name for each server type per configuration. The data items of this single data source will be directly added to the root node of the namespace of the local application.

© Softing Industrial Automation GmbH

Note: If the data source name is left empty, the root item for this data source connection will be shown in the configurator namespace tree as *<EmptyDA>* or as *<EmptyAE>* depending on the selected OPC server type. However, in the real namespace of the local dataFEED OPC Suite application the corresponding data items will be added directly under the root node of the namespace of the local dataFEED OPC Suite application.

<<< OPC Server Wizard - Welcome OPC Server Wizard - Server >>>

On this wizard page the external OPC server is defined for the selected data source connection:



OPC Server

Select one of the discovered OPC servers from the *Found OPC Server*s list or manually specify the corresponding server address in the *Address* input field at the bottom of this wizard page. The OPC server discovery supports only OPC DA and OPC AE servers, however. For OPC XML-DA servers the address always needs to be entered manually in the corresponding field.

Selected OPC Server

Specify the address of the external OPC server manually if necessary for this OPC server connection. The address must always be specified in one of the address formats described below in the OPC Server Address Format section.

OPC Server Address Format

The address format of a connection to an external OPC server depends on the type of the server.

1. OPC DA and OPC AE servers

```
<ConnectionType>://<HostName>/<ProgramID>/{ClassID}
```

or

```
<ConnectionType>://<IPAddress>/<ProgramID>/{ClassID}
```

The *<ConnectionType>* placeholder is a pre-defined string, the corresponding values are shown in the table below:

<ConnectionType>	**Description**
opcda	Connection to an external OPC Data Access server.
opcae	Connection to an external OPC Alarms & Events server.

The *<ProgramID>* and *<ClassID>* placeholders are the corresponding OPC / DCOM identities that are used for registering the external OPC server in the PC. The *<HostName>* and *<IPAddress>* placeholders specify the host name and the IP address of the PC on which the external OPC server is running.

Note: To connect to an OPC server running on the same PC, the host name *localhost* or the IP address *127.0.0.1* should be used as host name or IP address.

Examples:

```
opcda:///Softing.OPCToolboxCSmpl.Namespace.DA/{3E0289BA-34CA-445b-9EC1-B59F3F482683}
opcda://172.17.3.157/Softing.OPC.S7.DA/{6F17505C-4351-46AC-BC1E-CDE34BB53FAA}
opcda://WRT3-2KPRO/Softing.OPCToolboxDemo_ServerDA/{2e565242-b238-11d3-842d-0008c779d
opcae:///Softing.OPCToolboxDemo_ServerAE.1/{2E565243-B238-11D3-842D-0008C779D775}
```

2. OPC XML-DA servers

```
http://<HostName>:<PortNumber>[/<AddressSuffix>]
```

or

```
http://<IPAddress>:<PortNumber>[/<AddressSuffix>]
```


© Softing Industrial Automation GmbH

The <*HostName*> and <*IPAddress*> placeholders specify the host name and the IP address of the PC on which the external OPC server is running. The <*AddressSuffix*> placeholder is optional and depends on the configuration of the external OPC server.

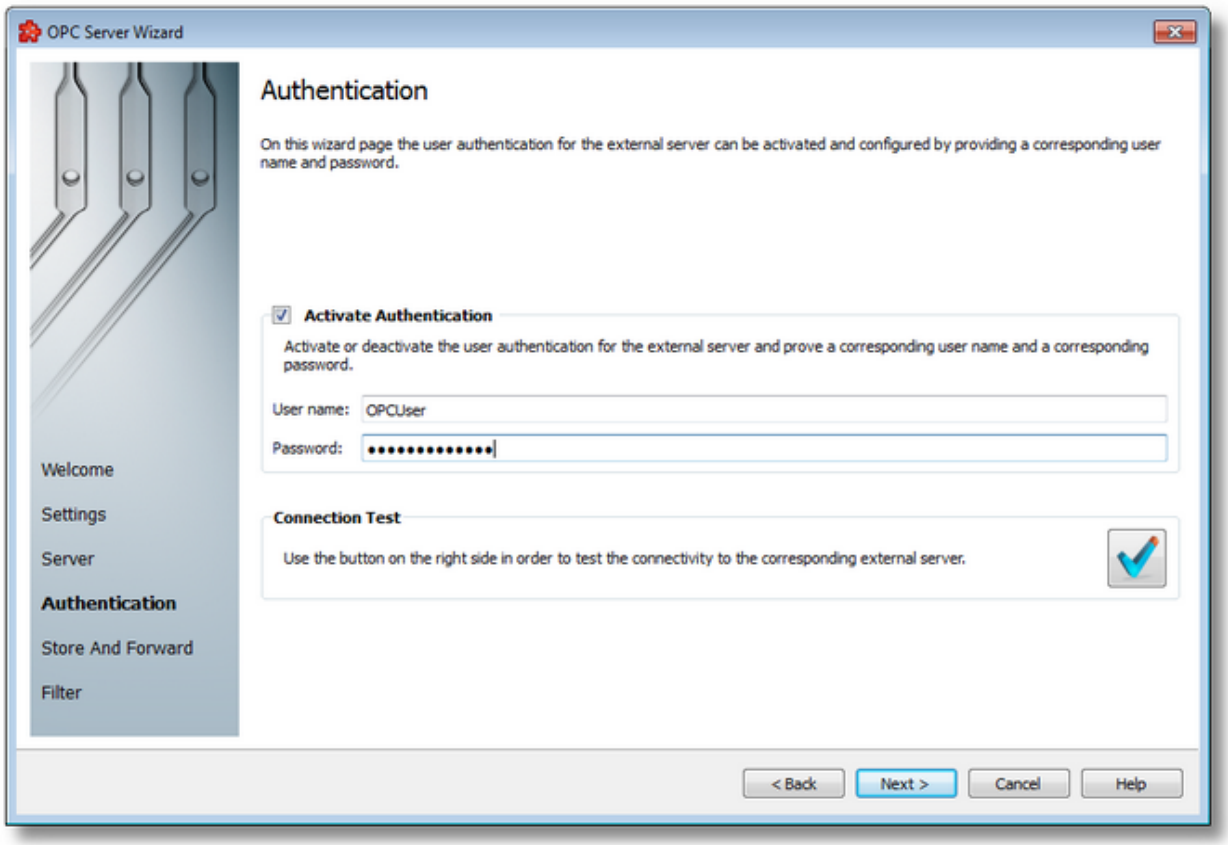
Note: To connect to an OPC server running on the same PC, the host name *localhost* or the IP address *127.0.0.1* should be used as host name or IP address.

Examples:

```
http://localhost:8080/DA
http://172.0.0.1:8079/OPC/DA
http://172.17.3.155:8090
```

<<< OPC Server Wizard - Settings OPC Server Wizard - Authentication >>>

On this wizard page the user authentication for the external OPC server can be activated and configured by providing a corresponding user name and password.



Activate Authentication

By checking or unchecking the Activate Authentication checkbox the user authentication for the external OPC server can be activated or deactivated and a corresponding user name and password can be specified.

Connection Test

This page also offers the possibility to test the connection to the current external OPC server. For the actual test the specified user name and password are used.

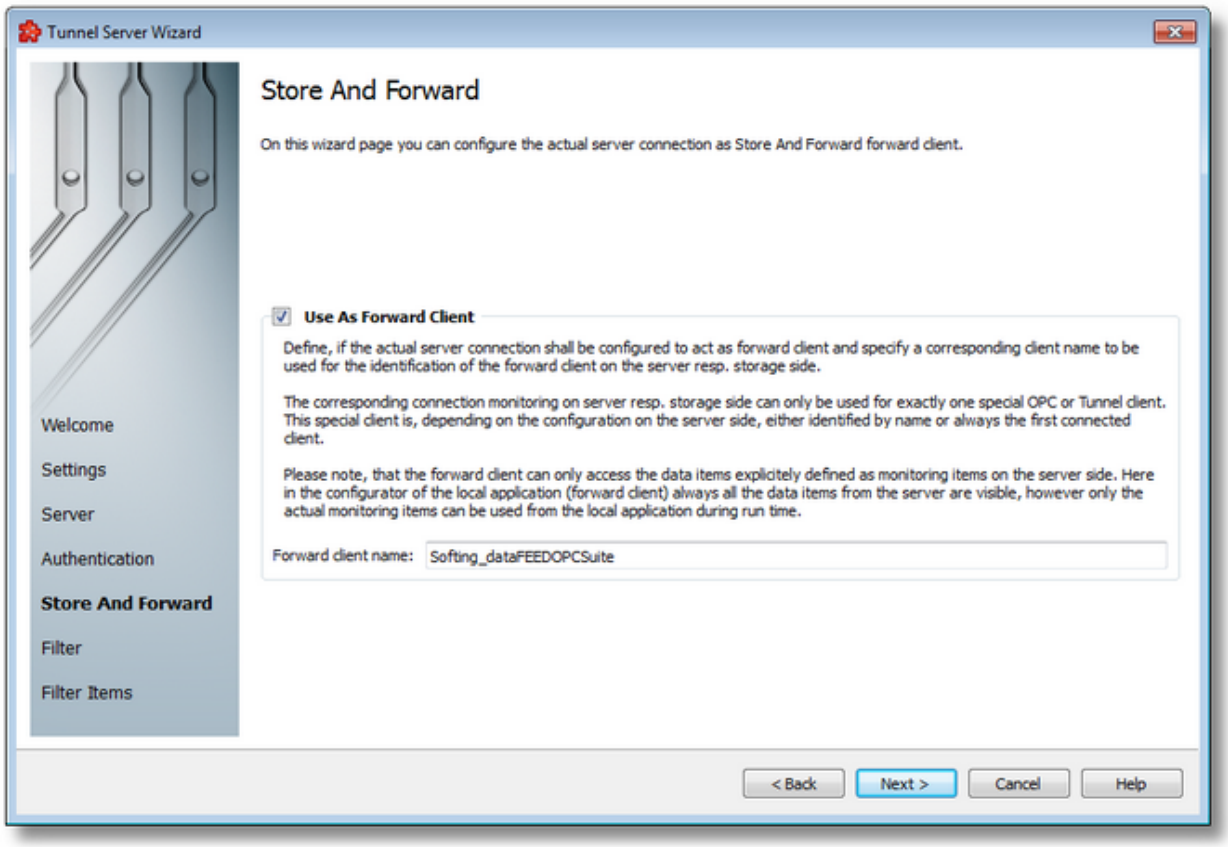
To start the connection test, use the *Test* button  in the *Connection Test* section. A new dialog will appear on screen showing the connection test results. The *Connection Test* dialog is described on the Connection Test page of the OPC Server chapter under Data Sources.

Note: An OPC AE server connection cannot be used as a Store And Forward connection and does not provide the possibility to define filter items. So for OPC AE server connections the *Next* button is replaced by a *Finish* button.

<<< OPC Server Wizard - Server OPC Server Wizard - Store And Forward >>>

On this wizard page you can configure the selected server connection as Store And Forward client:

© Softing Industrial Automation GmbH



By checking the *Use As Forward Client* checkbox, the OPC server connection is configured to act as a forward client.

In the *Forward client name* input field you can specify the name of the forward client which the corresponding OPC server (storage side) will use to identify the client as the forward client.

The corresponding connection monitoring on the server/storage side can only be used for a single, special OPC or Tunnel client. Depending on the configuration on the server side this special client is either identified by the client name or as the first connected client.

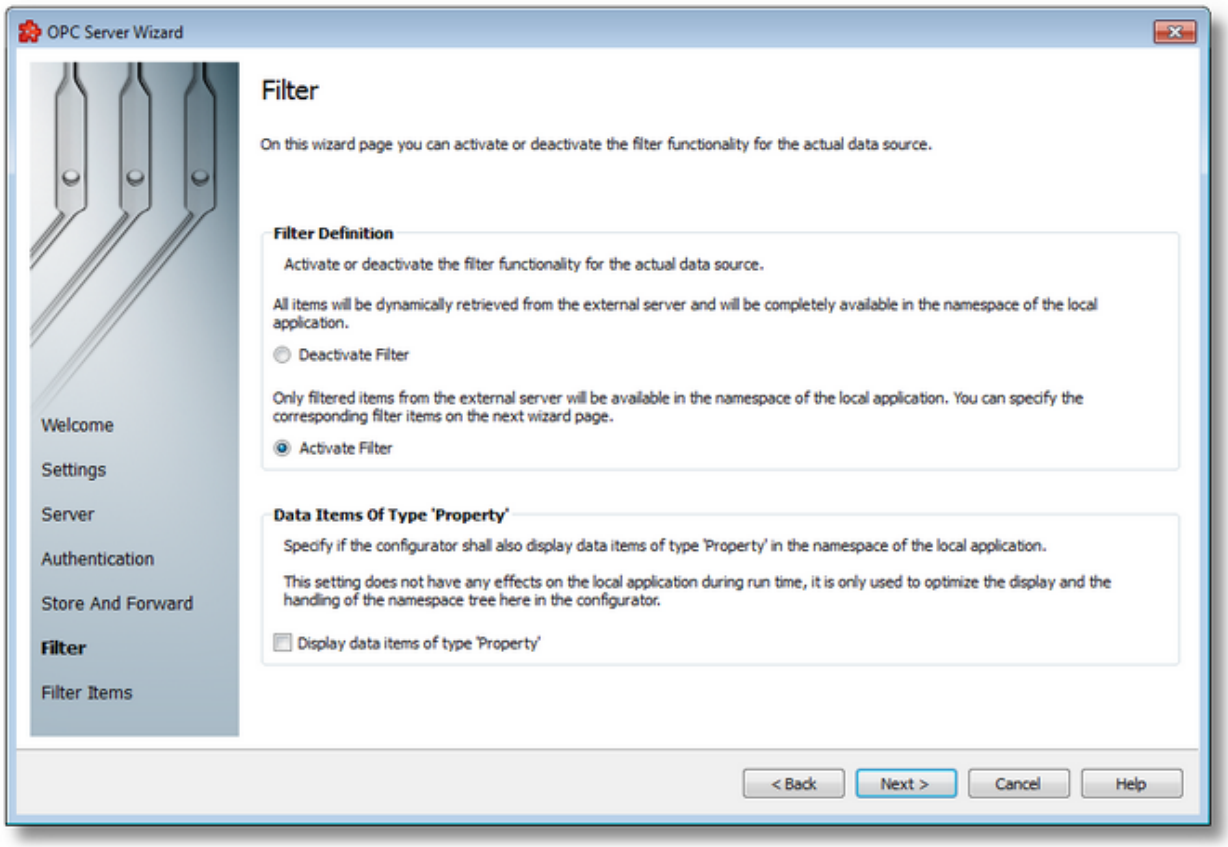
Please note that the forward client can only access the data items explicitly defined as monitoring items on the server side. Here in the configurator of the local dataFEED OPC Suite application (forward client) all the data items from the server are always visible, but only the actual monitoring items can be used from the local dataFEED OPC Suite application during run time.

Attention: The forward client can only access the data items explicitly defined as monitoring items on the server side. If the forward client tries to access data items which are not defined as monitoring items it will always receive the corresponding value with the OPC quality *BAD (Configuration Error)*.

For more details on how to configure Store And Forward see the Configuration: Data Processing - Store And Forward page in the Store And Forward chapter under Data Processing.

<<< OPC Server Wizard - Authentication OPC Server Wizard - Filter >>>

On this wizard page you can activate or deactivate the filter functionality for the selected data source.



Filter Definition

For the *Filter Definition* there are two possible configuration settings:

Filter Definition	Description
Deactivate Filter	All items will be dynamically retrieved from the external server and will be completely available in the namespace of the local dataFEED OPC Suite application.
Activate Filter	Only filtered items from the external server will be available in the namespace of the local dataFEED OPC Suite application. You can specify the filter items on the next wizard page.

Another option of this wizard page allows enabling or disabling the display of data items of the *Property* type in the namespace of the local dataFEED OPC Suite application.

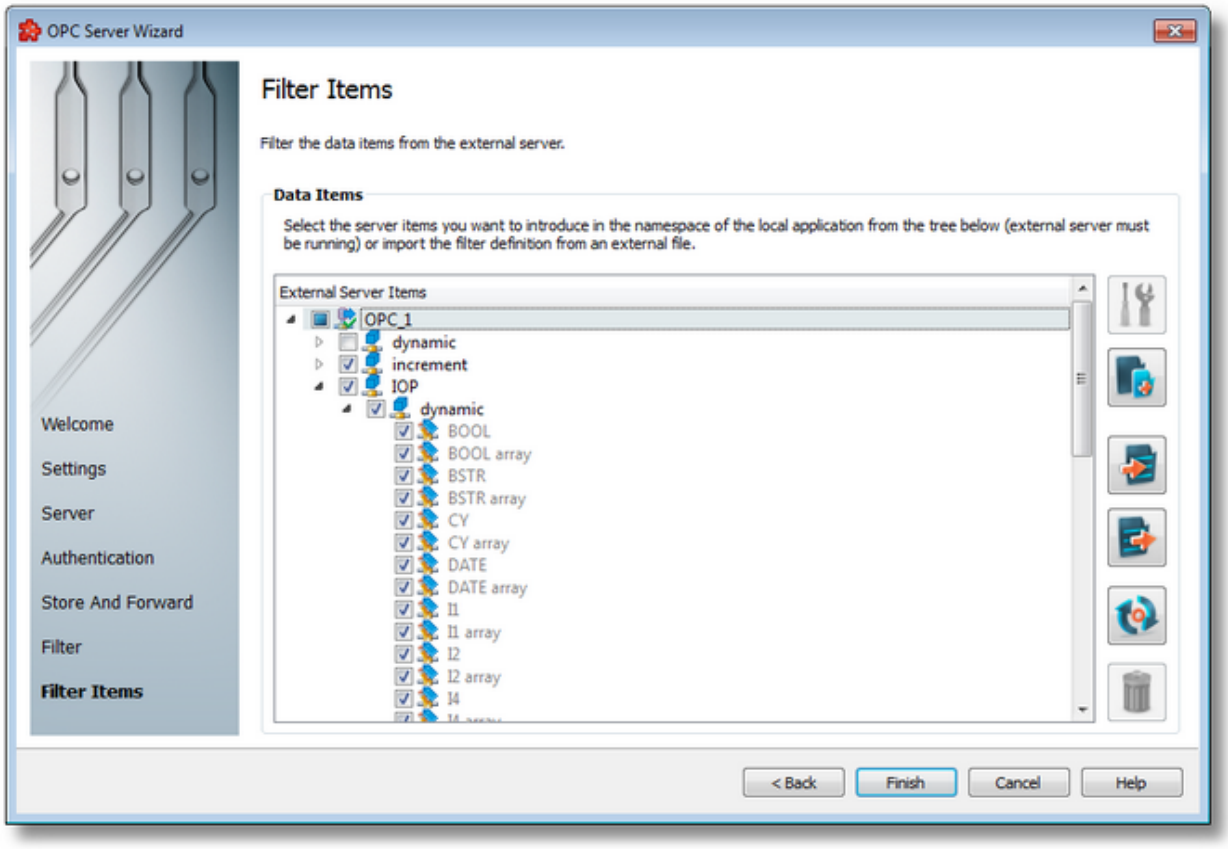
© Softing Industrial Automation GmbH

Attention: This setting does not have any effect on the local dataFEED OPC Suite application during run time, it is only used to optimize the display and the handling of the namespace tree in the dataFEED OPC Suite Configurator.

Depending on the *Filter Definition* selection the *Next* button will be available to continue the OPC Server Wizard with the *Filter Items* page (when the filter functionality was activated) or the *Finish* button will be available allowing you to finish the OPC Server Wizard and save the changes.

<<< OPC Server Wizard - Store And Forward OPC Server Wizard - Filter Items >>>

On this wizard page the filter data items for the connection to the actual external OPC server can be defined. Select the server data items you want to introduce into the namespace of the local dataFEED OPC Suite application from the corresponding tree control (external server must be running) or import the filter definition from an external file.



If the external OPC server is running, the namespace of the external OPC server is shown in the tree under the root node named after the actual connection. The tree is expandable and each item can be selected to be included in the filter file or not. The selection is downward recursive, which means that when you include a node all subsequent nodes, tags and properties in that particular branch of the namespace tree will automatically be included.

If the connection to the OPC server failed, the namespace tree will only contain the server root node. However, it is possible at any time to try reconnecting to the external server and refreshing the data by using the *Refresh data items* button  .

Operation

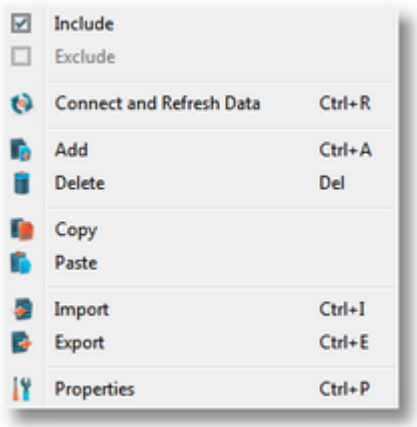
Depending on the current selection in the *External Server Items* tree, the following operations are possible using the buttons on the right, the context sensitive menu or the corresponding shortcuts:

- Properties of the selected item (Ctrl+P) 
- Add a new item (Ctrl+A) 
- Import from file (Ctrl+I) 
- Export to file (Ctrl+E) 
- Refresh data items (Ctrl+R) 
- Delete data items (Del) 

Two more operations are supported, which are only accessible from the contextual menu:

- Copy data item(s)
- Paste to data item(s)

An example of the contextual menu available on *Data Items* tree nodes is shown below:

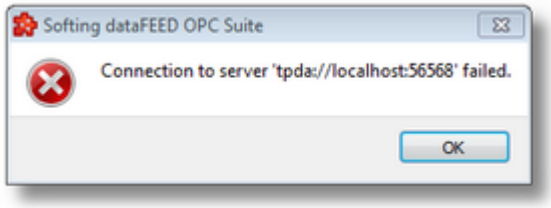


© Softing Industrial Automation GmbH

Refresh data items

The *Refresh data items* (Ctrl+R) operation (or the corresponding *Connect and Refresh Data* context menu action) will try to reconnect to the OPC server to refresh the data items tree.

If the action succeeds the namespace tree will be refreshed in order to reflect possible changes to the external OPC server namespace. If the connection to the remote OPC server fails, an error message box will be displayed:



Delete data items

Triggering the *Delete data items* action with the corresponding button on the right wizard side or *Delete* from the context menu will remove the currently selected node from the *Data Items* namespace tree.

Attention: The root node of the namespace tree can not be deleted.

Attention: A node which is currently used in a data processing or data destination action can not be deleted.

Copy

The *Copy* action is provided in the context menu of the *External Server Items* tree. Triggering this action will copy the selected node to the local clipboard. All the properties of the node are copied. Also, if the node contains children, the entire branch is copied.

Note: The root node of the namespace tree does not provide a *Copy* action.

Paste

The *Paste* action is provided in the context menu of the *External Server Items* tree. This action will add the node currently stored in the local clipboard as a a child of the currently

selected node. Whether the action is available in the context menu depends on the type of the currently selected node and the type of the node in the clipboard.

Note: The *Paste* action is only available if a node was previously copied to the clipboard.

If the *Paste* action is performed on a node which already has a child of the same name as the node in the clipboard, the pasted node is renamed automatically by adding a numerical suffix to the original name.

```
E.g. NodeA, NodeA01, NodeA02, ...
```

<<< OPC Server Wizard - Filter

7.3.1.8.7 Import Wizard

The Softing Configurator allows the import of OPC filter items from an external file.

The external file should be a plain text tab-separated file with the specific OPC filter file format.

The import action can be triggered by using the *Import from file* button  , by pressing the keyboard shortcut CTRL+I or by selecting the *Import* action in the context sensitive menu.

Note: The import action is available for the root of the OPC server connection namespace and for items of type *Node* from the namespace.

Attention: The import action is not available for items of type *Tag* and *Property*.

Note: Name conflicts between the old and new data items are solved by renaming the newly imported items. They are renamed by appending a two-digit number at the end of the name of the first duplicated item.

When the import action is triggered the *Filter Import Wizard* is started.

▼ The Filter Import Wizard allows importing different data items:

- Select the file to be imported @ Import Wizard - Import File Selection
- Examine the import results @ Import Wizard - Import Results

Each of the OPC server connections configured to use a filter file will produce a file containing the definition of the filter data items when the configuration is saved.

© Softing Industrial Automation GmbH

The files are stored in the *Filter* subfolder of the configuration folder. The filter file is named after the server connection name. The filter files are plain text files, encoded in UTF-8 (thus providing full UNICODE compatibility) and they have a line-oriented structure. The elements of a line are tab-separated. For each filter data item a line is added to the file.

The following line types can be found in the OPC server connection filter file:

- Filter item line, defining a filter data item as described in the table below.
- Empty line.
- Comment line, starting with #; such a line is ignored during evaluation.

A filter item line has the following format:

```
<TABSofting server ID<TAB>Softing item ID<TAB>Server item ID<TAB>Server item path<TAB
```

The different elements of the line are described in the following table:

Element	Description
Softing server ID	Not used. Should be always an empty string.
dataFEED OPC Suite item ID	The OPC item name identifying the corresponding data item inside the dataFEED OPC Suite namespace.
Server item ID	The OPC item ID for the data item in the external OPC server namespace.
Server item path	The OPC path for the data item on the external OPC server.
Server browse path	The OPC browse path for the data item, describing the individual browse steps for the data item in the namespace of the external OPC server.
Data type	The data type of the data item. The possible values are described in the Filter Item Properties page.
Access rights	The access rights for the data item.The possible values are: r, w, rw corresponding to read, write, read and write, respectively.
Object type	The type of the data item. The possible types are: Node, Tag and Property.
Comment	An optional comment/note attached to the current data item.

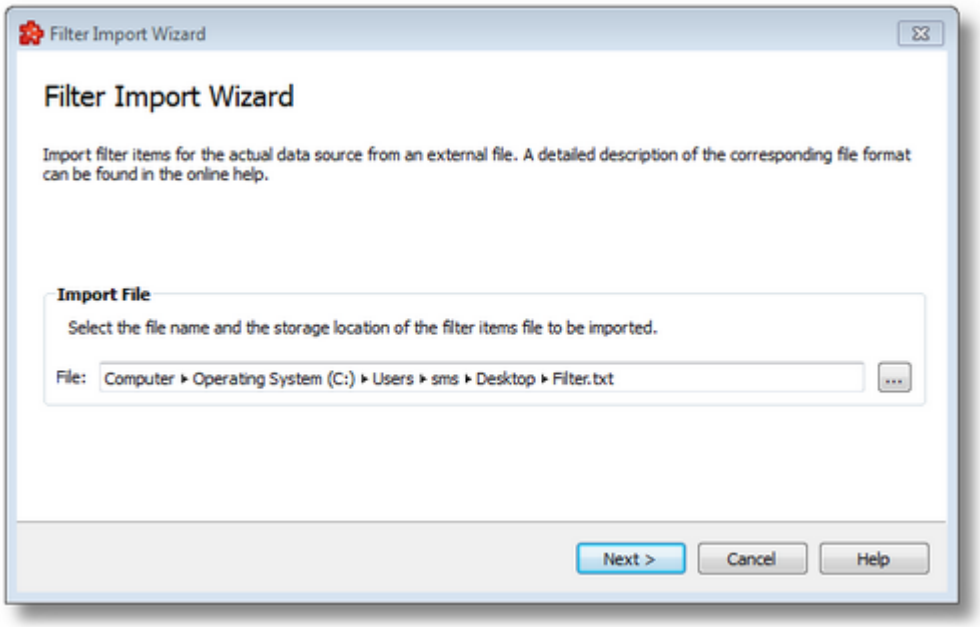
Example:

```
#<TAB>Softing server ID<TAB>Softing item ID<TAB>Server item ID<TAB>Server item path<T
#
<TAB><TAB>maths.automatic angle change<TAB>maths.automatic angle change<TAB><TAB>math
<TAB><TAB>maths.automatic change period<TAB>maths.automatic change period<TAB><TAB>ma
```

```
<TAB><TAB>maths.cos<TAB>maths.cos<TAB><TAB>maths.cos<TAB>R8<TAB>r<TAB>Tag<TAB>Cosinus
<TAB><TAB>maths.cos|102<TAB>maths.cos#102<TAB><TAB>maths.cos|102<TAB>R8<TAB>rw<TAB>Pr
<TAB><TAB>maths.cos|103<TAB>maths.cos#103<TAB><TAB>maths.cos|103<TAB>R8<TAB>rw<TAB>Pr
<TAB><TAB>maths.cos|6020<TAB>maths.cos#6020<TAB><TAB>maths.cos|6020<TAB>UI2<TAB>rw<TA
<TAB><TAB>maths.sin<TAB>maths.sin<TAB><TAB>maths.sin<TAB>R8<TAB>r<TAB>Tag<TAB>Sinus d
<TAB><TAB>maths.sin|102<TAB>maths.sin#102<TAB><TAB>maths.sin|102<TAB>R8<TAB>r<TAB>Pro
<TAB><TAB>maths.sin|103<TAB>maths.sin#103<TAB><TAB>maths.sin|103<TAB>R8<TAB>r<TAB>Pro
<TAB><TAB>maths.tan<TAB>maths.tan<TAB><TAB>maths.tan<TAB>R8<TAB>r<TAB>Tag<TAB>
<TAB><TAB>maths.tan|6020<TAB>maths.tan#6020<TAB><TAB>maths.tan|6020<TAB>UI2<TAB>rw<TA
```

The Import File Selection wizard page allows you to choose a file to be imported.

The file can be selected by manually typing its name in the *File* input box or by clicking the browse button located to the right of the input box (a standard file selection dialog will open).



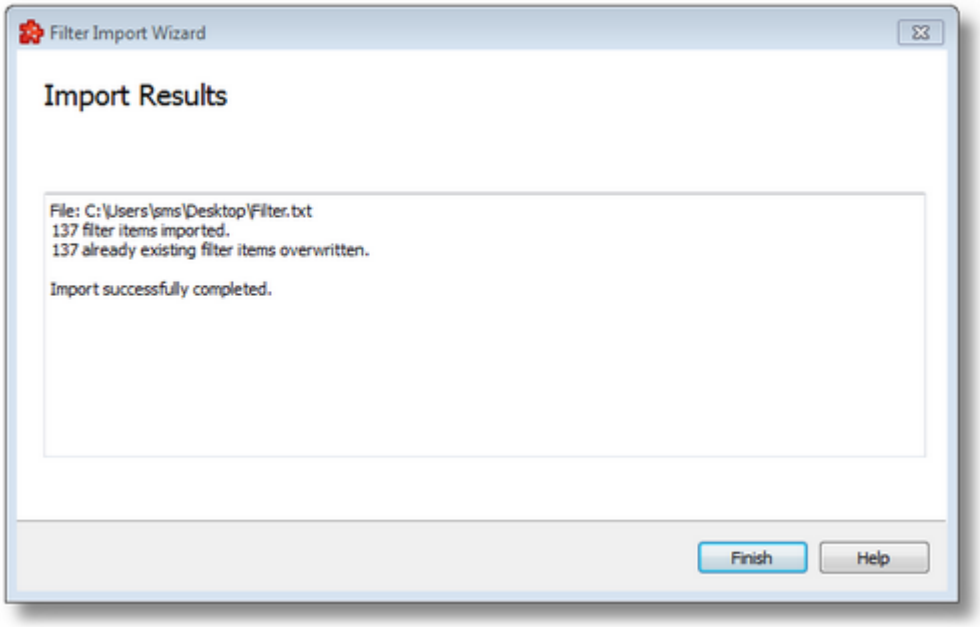
The file path and name should be valid (they should point to an existing file).

When you press the *Next* button the file path and name are verified. If the file identified by the specified file path and name does not exist an error message is displayed. The actual import begins if the selected file is valid. If the import takes longer to complete a waiting dialog is displayed. When the import operation is finished the *Filter Import Results* page is displayed.

Import Wizard - Import Results >>>

The Import Results wizard page displays the import results.

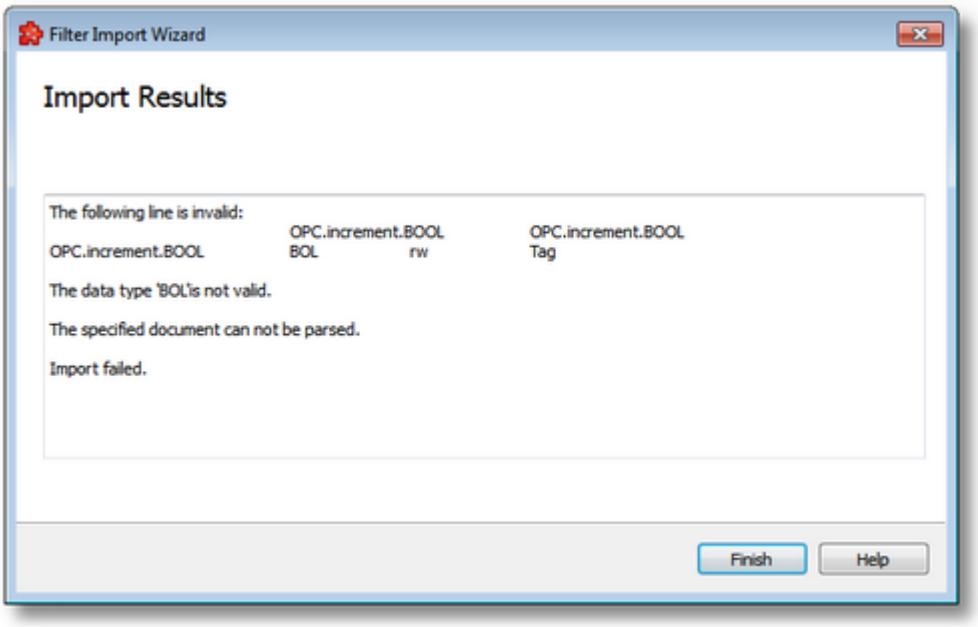
© Softing Industrial Automation GmbH



This page is the final page of the *Filter Import Wizard*. A back button is not provided as the import action cannot be undone. The import can be successful or it can fail:

1. After a successful import the results page contains the following information:
   - The absolute path of the imported file.
   - The number of local items imported. This number includes only the items of type Tag, the items of type Node are not counted.
   - The confirmation message that the import was successful.
2. After a failed import the results page contains the following information:
   - The invalid line where the import failed.
   - The reason why the line is considered invalid.
   - The confirmation message that the import failed.

An *Import Results* example for a failed import operation can be seen below:



When the import wizard is closed after a successful import, the new items are added to the current OPC server connection tree as a branch of the currently selected node item.

<<< Import Wizard - Import File Selection

7.3.1.8.8 Export Wizard

The dataFEED OPC Suite Configurator allows exporting the filter items belonging to an OPC server connection. The entire namespace tree or just a sub-branch of the OPC server connection can be exported.

The resulting external file will be a tab-separated file with the specific OPC filter file format.

The export action can be triggered by pushing the *Export* button  , by pressing the keyboard shortcut CTRL+E or by selecting the *Export* action in the context sensitive menu. The export action is available for the root of the OPC server connection namespace and for any element of type *Node* of the namespace.

Attention: The export action is not available for nodes of type *Tag* or *Property*.

The exported file will contain all the filter items from the branch of the selected node.

Note: The exported file will contain only the selected (checked) data items.

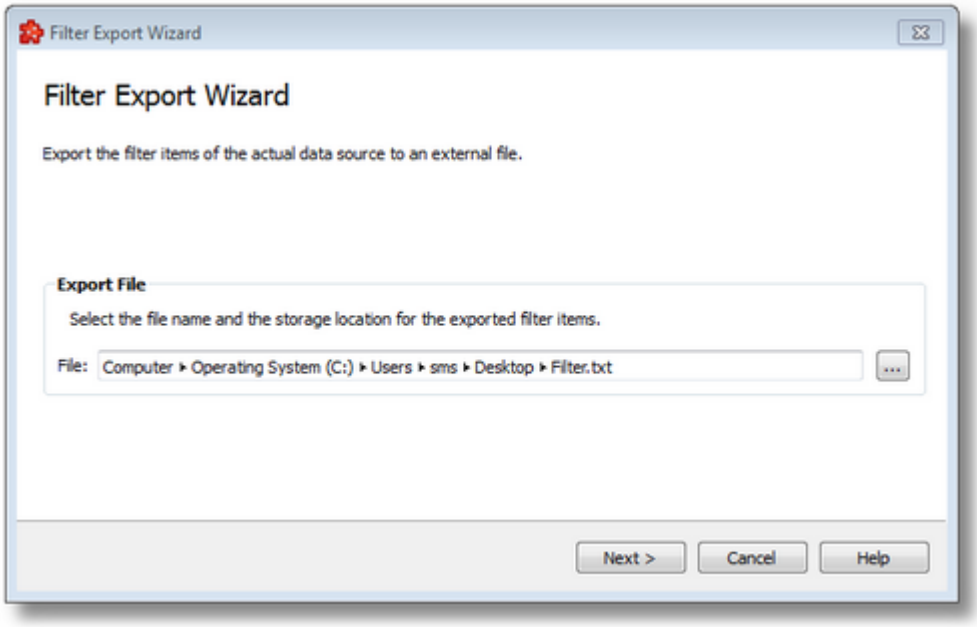
When the export action is triggered the *Filter Export Wizard* is started.

▼ The Filter Export Wizard allows exporting different data items:

© Softing Industrial Automation GmbH

- Select the file the data items are to be exported to @ Export Wizard - Export File Selection
- Examine the export results @ Export Wizard - Export Results

The Export File Selection wizard page allows you to choose a file name and location for the exported file. The file can be selected by manually typing its path and name in the *File* input box or by clicking the browse button located to the right of the input box (a standard file saving dialog will open).

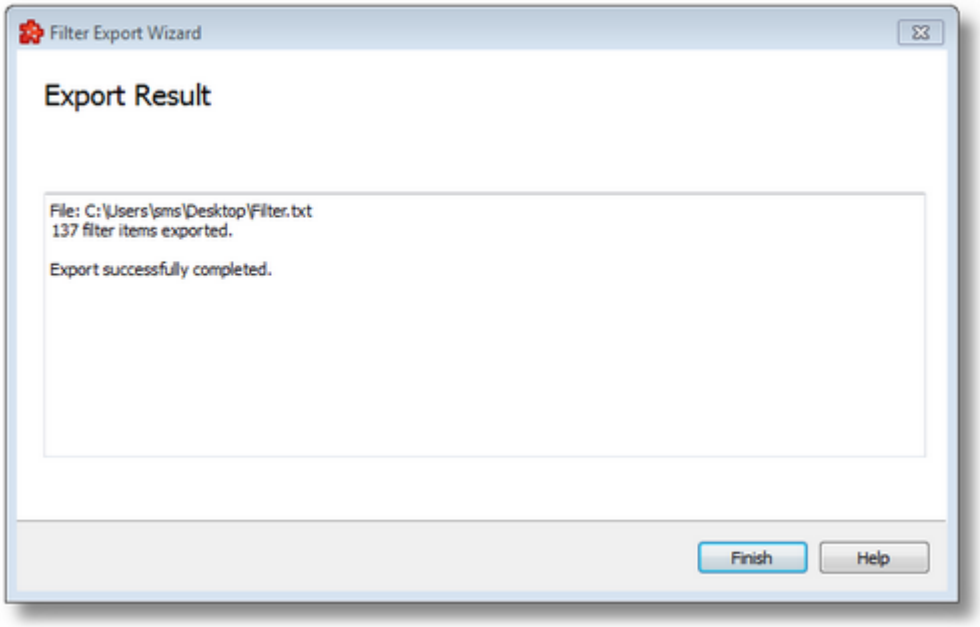


When the *Next* button is pressed the file path is verified and if the path does not exist, an error message is displayed (invalid path). If the specified file path is valid and a file with the specified name already exists, a message box will ask whether to overwrite the existing file or not (select another file). The actual filter export operation begins if the path to the file is valid and either a file with the specified name does not exist or it exists and overwrite has been chosen.

When the export operation is finished the *Export Results* page is displayed.

Export Wizard - Export Results >>>

The Export Results wizard page displays the export results as shown in the image below:



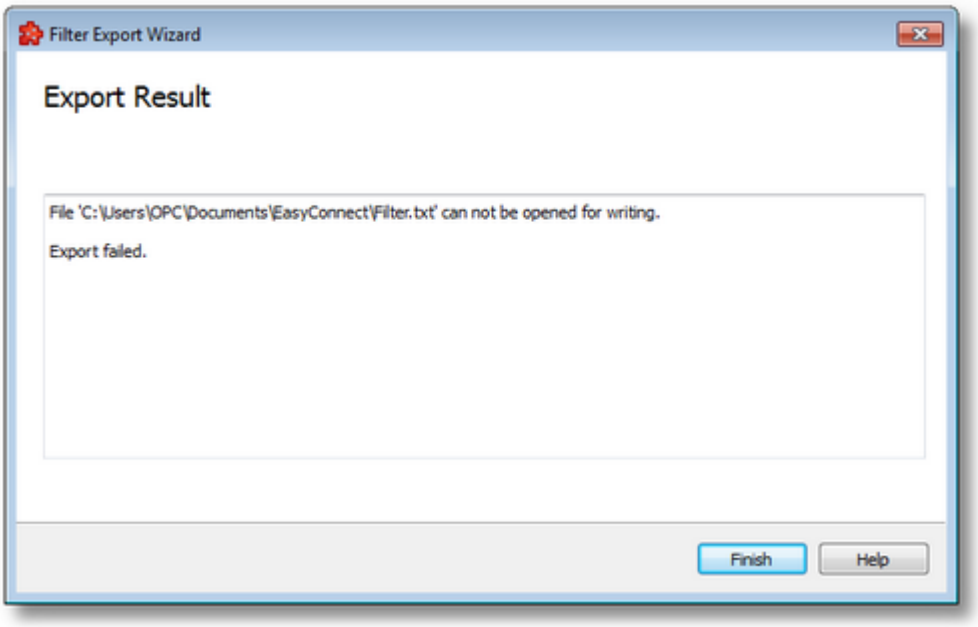
This page is the final page of the Filter Export Wizard. A back button is not provided as the export action cannot be undone. The export can be successful or it can fail:

1. After a successful export the results page contains the following information:
   - The absolute path of the exported file.
   - The number of filter items exported. This number includes only the items of type Tag and Property, the data items of type Node are not counted.
   - The confirmation message that the export was successful.

2. After a failed export the results page contains the following information:
   - The reason why the export failed.
   - The confirmation message that the export failed.

An *Export Results* example for a failed export operation can be seen below:

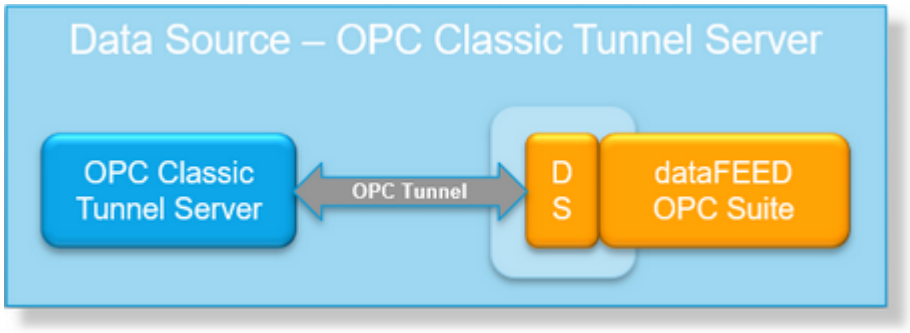
© Softing Industrial Automation GmbH



<<< Export Wizard - Export File Selection

7.3.1.9 OPC Classic Tunnel Server

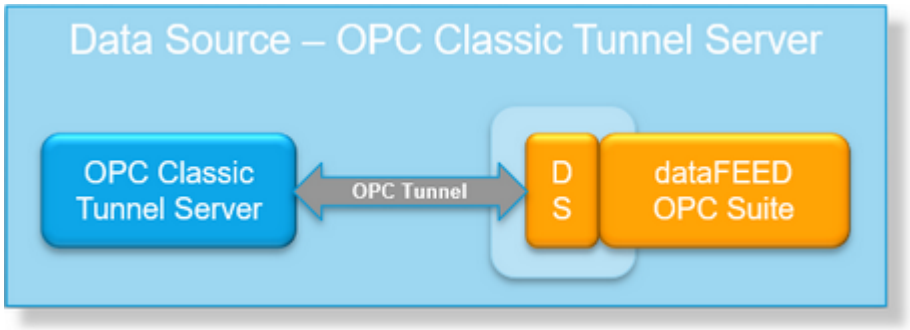
This section of the Softing dataFEED OPC Suite Online Help describes how to configure the Tunnel server data source functionality of Softing dataFEED OPC Suite.



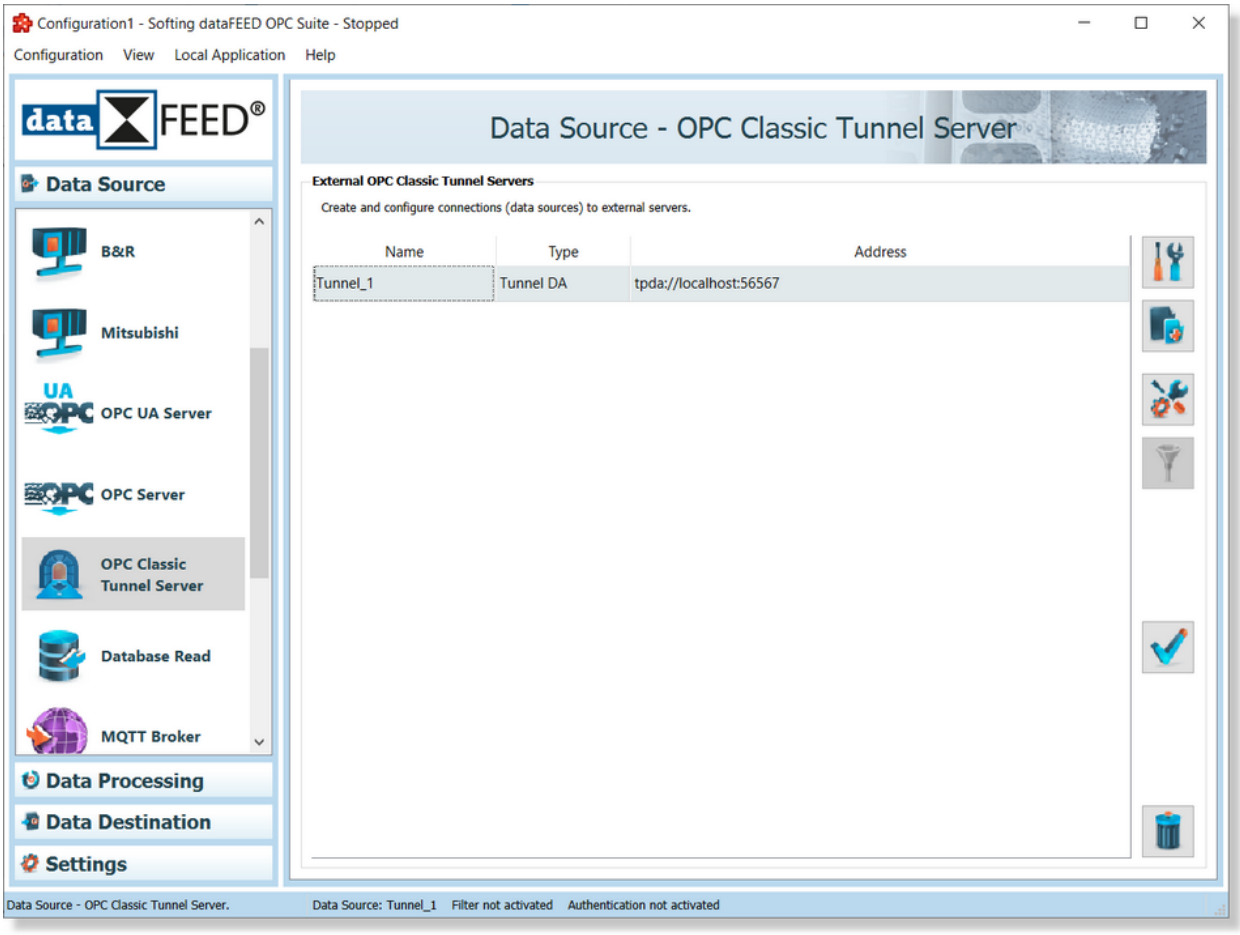
Softing dataFEED OPC Suite can connect to several OPC Classic Tunnel servers, thus avoiding DCOM. The OPC Classic Tunnel provides you with a secure and straightforward cross-network OPC communication. The namespace elements of the servers to be addressed are integrated into the namespace of the Softing dataFEED OPC Suite application.

A general description of the Softing OPC Classic Tunnel can be found under Introduction - OPC Classic Tunnel.

7.3.1.9.1 Configuration



To start configuring the OPC Classic Tunnel Server select *OPC Classic Tunnel Server* from *Data Source* group in the navigator on the left. The OPC Classic Tunnel server main user interface is displayed. The main element of this page is the *External OPC Classic Tunnel Servers* list. Each entry shown in the list of the configuration section represents a connection to an OPC Classic Tunnel server.



Note: The list is only shown and editable if a configuration has been loaded.

Each OPC Classic Tunnel server connection has a unique name within the configuration, which is given in the *Name* column.

© Softing Industrial Automation GmbH

The name of the server will be shown in the namespace tree in the Softing dataFEED OPC Suite Configurator and in the local dataFEED OPC Suite application, and its entire namespace will be under this name.

Note: The name can also be an empty string, but it has to be unique. If the name of the OPC Classic Tunnel server is empty, the dataFEED OPC Suite Configurator tree will expose the Tunnel server's namespace under a node named <*empty*>, but the local dataFEED OPC Suite application will expose it to other clients directly in the root.

The *Type* of each OPC Classic Tunnel server is displayed in the second column and the server *Address* in the third. The type of each server can be either OPC Clasic Tunnel DA Server or OPC Classic Tunnel AE Server.

Operation

From the OPC Classic Tunnel Server main user interface certain operations can be performed over the selected Tunnel server by using the buttons on the right, by accessing the context sensitive menu or by pressing the keyboard shortcuts. The following actions are available:

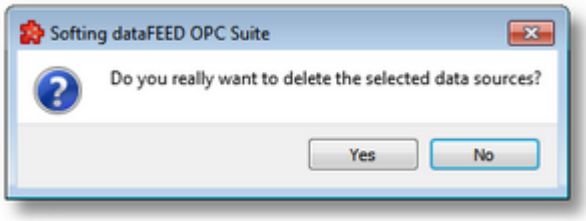
- Change the properties of the selected data source by using the *Properties* button  , pressing the keyboard shortcut CTRL+P or selecting *Properties* from the context menu.
- Add a new data source by using the *Add* button  , pressing the keyboard shortcut CTRL+A or selecting *Add* from the context menu.
- Change advanced settings for the selected data source by using the *Advanced settings* button  or selecting *Advanced* from the context menu.
- Filter data items for the selected data source by using the Filter button  , pressing the keyboard shortcut CTRL+F or selecting Filter from the context menu.
- Perform a connection test for the selected data source by using the *Connection test* button  , pressing the keyboard shortcut CTRL+T or selecting *Test* from the context menu.
- Delete the selected data source by using the *Delete* button  , pressing the DEL / DELETE key on the keyboard or selecting *Delete* from the context menu.

Note: If no Tunnel server connection is selected, the *Add a new data source* button is the only one available.

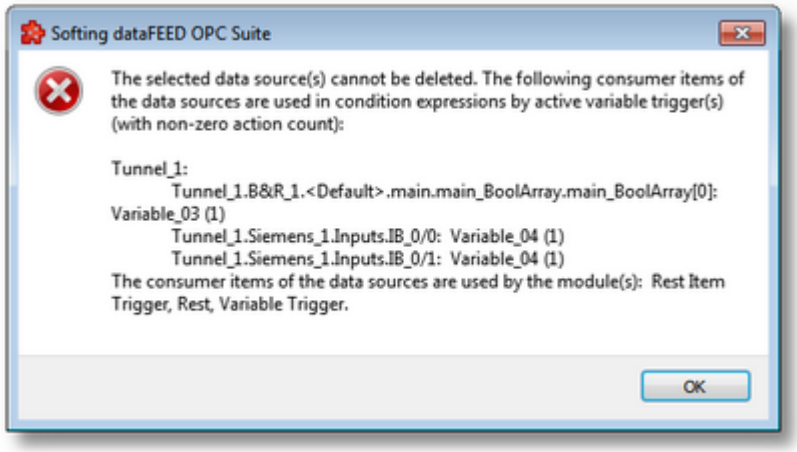
The *Delete* button  will remove the selected server connections from the list.

The user can select more than one server connection with the help of CTRL or SHIFT keys.

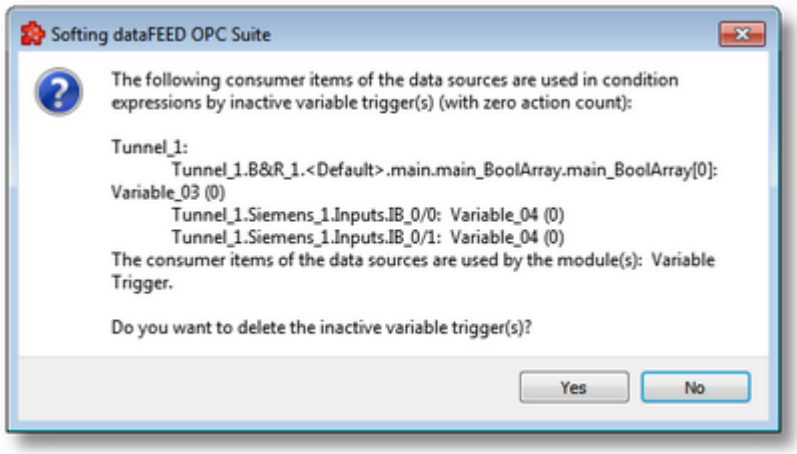
A dialog will then prompt the user to confirm the deletion.



A dialog will inform the user that the data sources cannot be deleted, if consumer items and/or system items of the selected data sources are used by active variable triggers or other modules.



A dialog will prompt the user to confirm deletion of the inactive variable triggers, if consumer items and/or system items of the selected data sources are used only by inactive variable triggers. A following delete command will remove the selected server connections from the list.

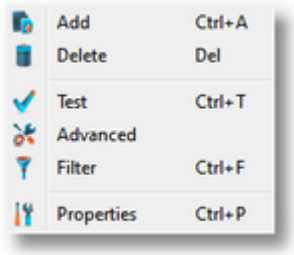


Attention: An OPC Classic Tunnel server connection cannot be deleted if at least one data item defined in its namespace is used in a Processing or Data Destination action.

All these actions are also provided in the context menu of the *External OPC Classic Tunnel Servers* list. Which actions are available depends on the current selection in the list.

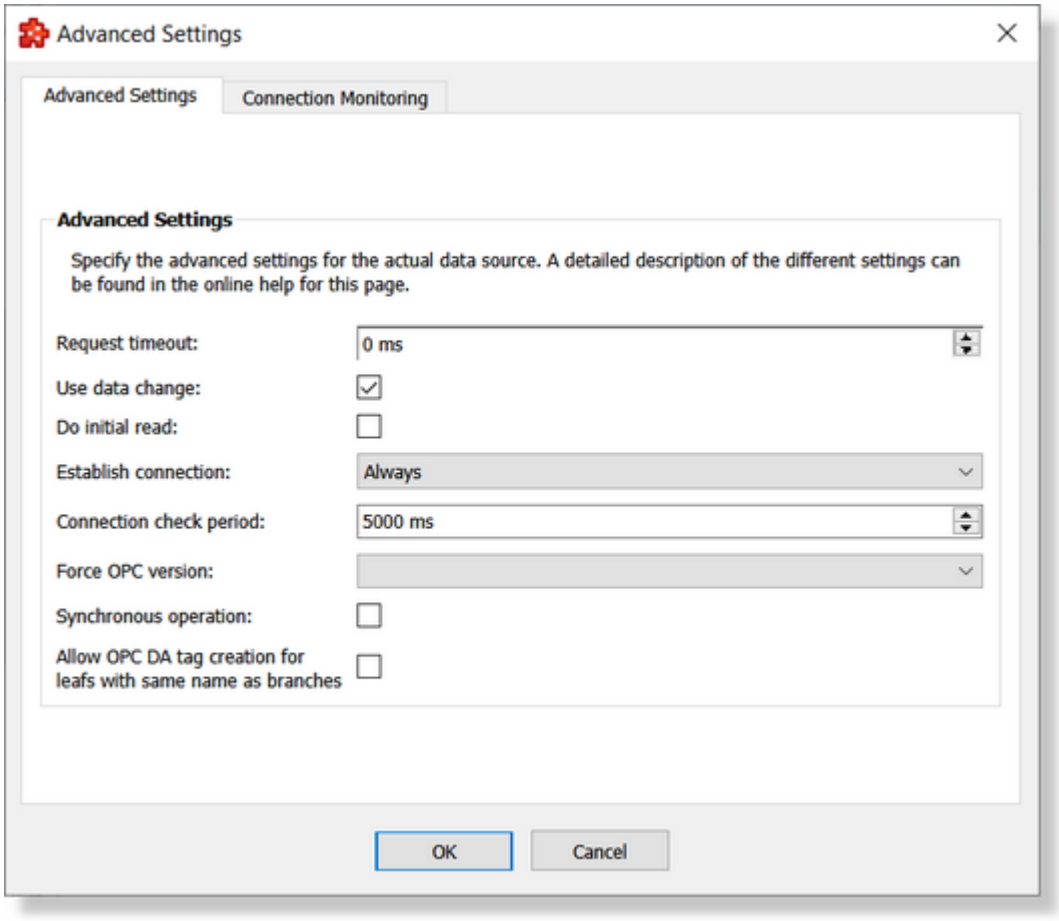
© Softing Industrial Automation GmbH

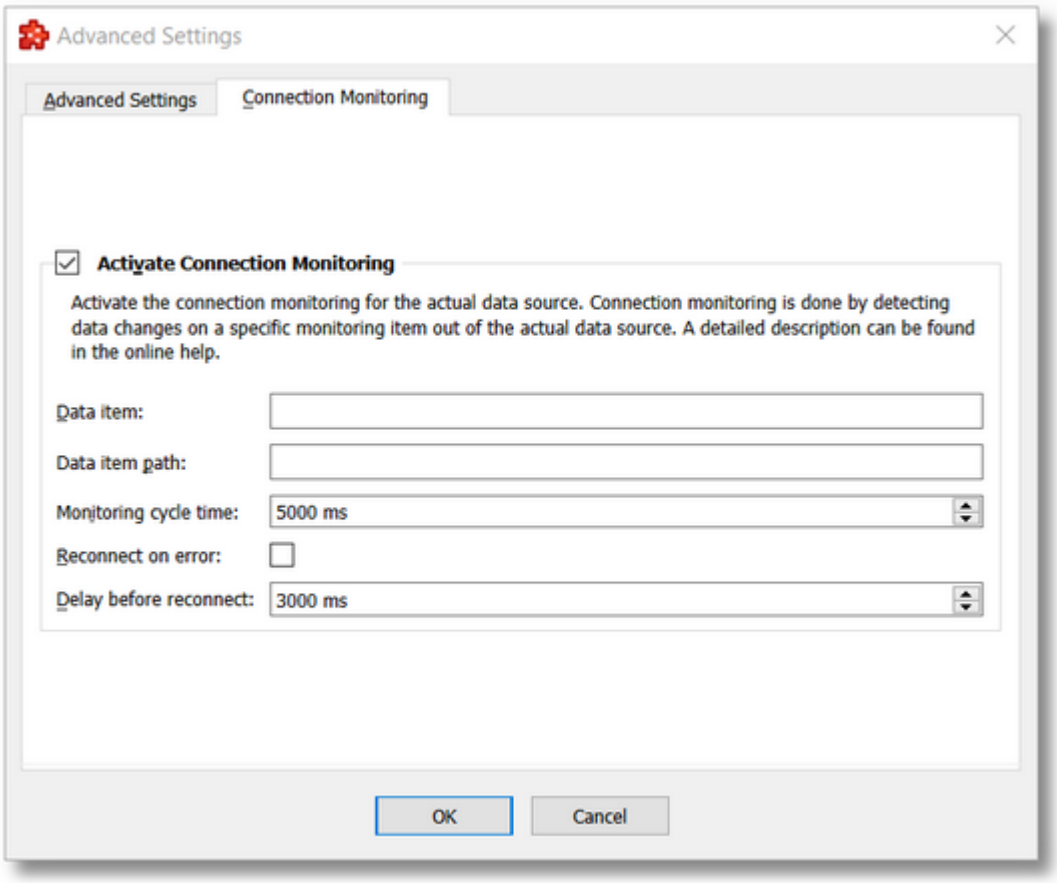
An example of the context menu can be seen below:



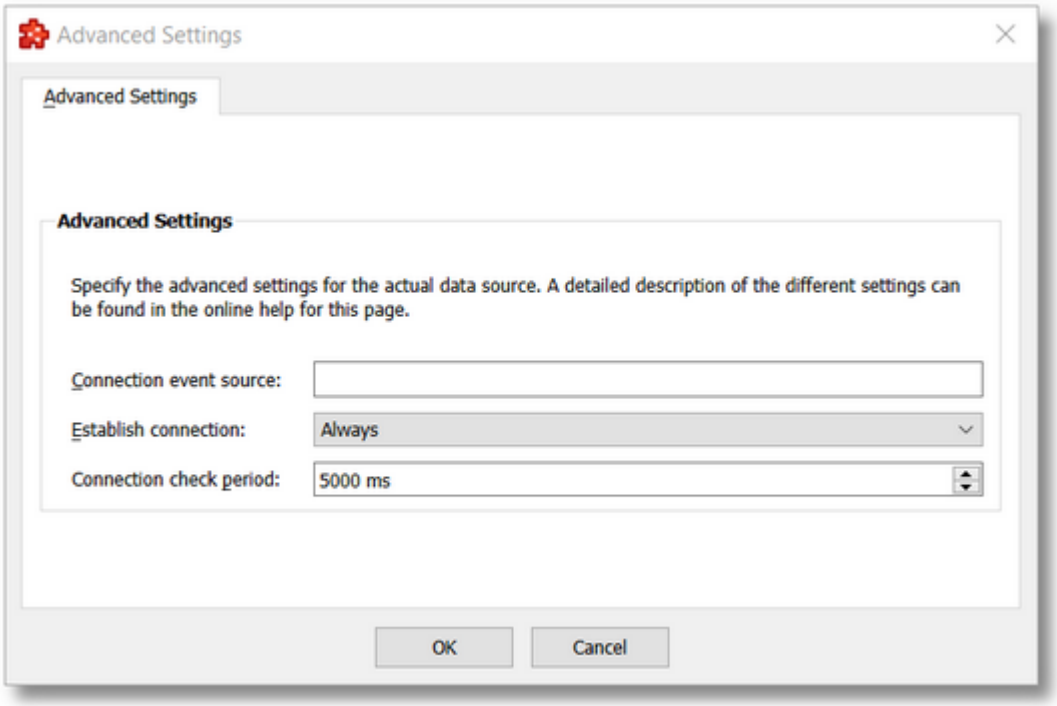
7.3.1.9.2 Advanced Settings

The Advanced Settings button shows a dialog that allows the user modify some default advanced settings. Depending of the OPC Classic Tunnel server connection type, different configuration options are available. For a OPC Classic Tunnel DA server connection the configuration options are organized in two different tabs, *Advanced Settings* and *Connection Monitoring*, as shown below:





For a OPC Classic Tunnel AE server connection, the *Advanced Settings* dialog looks like this:



© Softing Industrial Automation GmbH

Note: For OPC Classic Tunnel AE server connections, there is no *Connection Monitoring* tab in the *Advanced Settings* dialog.

The following tables depict all the configuration options, each with a brief description.

Configuration options for the *Advanced Settings* tab:

Attribute	Description
Request timeout	The maximum time in milliseconds that an I/O request to the OPC Classic Tunnel DA server may take. After this time, the request is canceled in the local dataFEED OPC Suite application.
Use data change	Checked if the OPC notification mechanism should be used for a OPC Classic Tunnel DA server connection. If unchecked, the values should be updated by synchronously reading the item cache.
Do initial read	Checked if an initial read should be performed on subscription to a OPC Classic Tunnel DA server.
Connection Event Source	The name of the Event Source to be used for signaling of connection failures at the OPC Classic Tunnel AE Server.
Establish connection	Specifies when the connection to the OPC Classic Tunnel server should be established. *Always* means the connection to the Tunnel server should be established at all times. *On Demand* means the connection will be established only as needed.
Connection check period	The time interval in milliseconds at which the connection to the Tunnel server is to be regularly checked. If no connection to the server exists, all items of this server receive the "communication failure" quality.
Force OPC version	This option allows selecting a special OPC specification interface to be used for communication with the OPC Classic Tunnel DA server. Possible values are *DA V1* or *DA V2*.
Activate sync operations	Checked if synchronous operations shall be used.
Allow OPC DA tag creation for leafs with same name as branches	Checked if it is is allowed to create OPC DA tags for leafs that have the same name as existing branches in the address space.

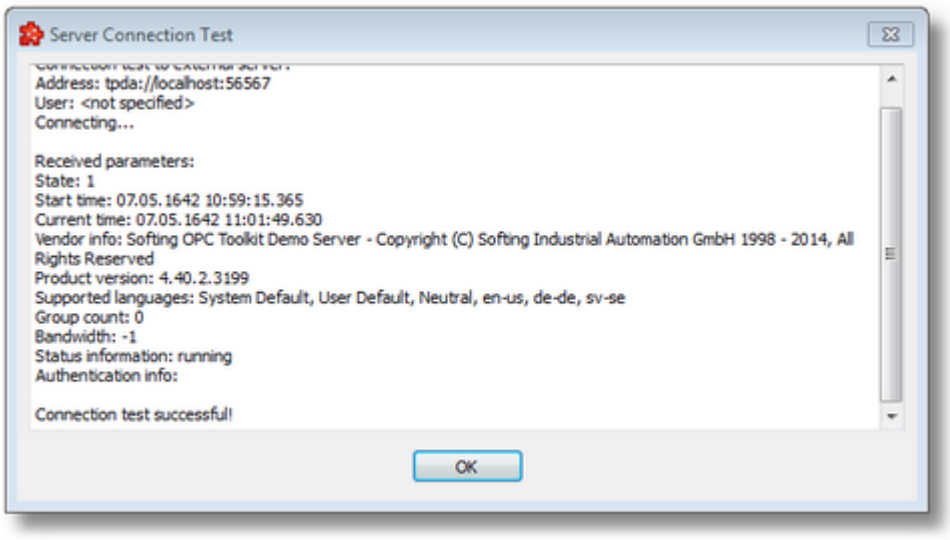
Configuration options from Connection Monitoring tab:

Attribute	Description

Activate Connection Monitoring	By checking this group box, the connection monitoring is enabled and the other configuration options become editable.
Data item	The data item which shall be used for the heartbeat monitoring. This data item will be monitored for all data changes. If an empty string is specified here the heartbeat connection monitoring will be turned off.
Data item path	The data item path (relevant for XML-DA only) corresponding to the data item which shall be used for the heartbeat monitoring.
Monitoring cycle time	Specifies the time interval in milliseconds used by the connection monitoring. The heartbeat data item has to change at least once within this time interval.
Reconnect on error	Checked if the server should restart if the heartbeat item reports there is no connection to the server.
Delay before reconnect	The maximum delay in milliseconds which is allowed for the heartbeat item.

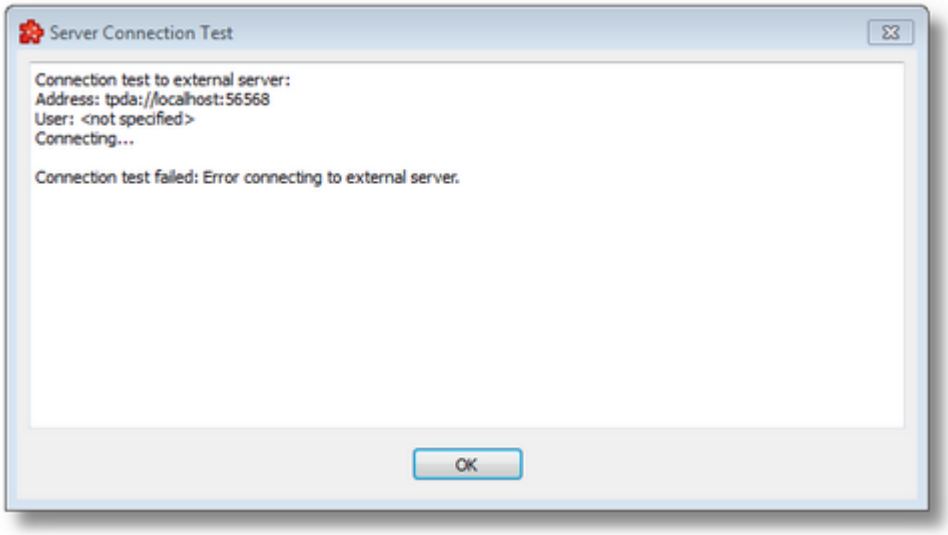
7.3.1.9.3 Connection test

The *Test Connection* button checks the OPC Classic Tunnel server connection. The Configurator tries to connect to the OPC Classic Tunnel server. A status dialog is shown with the test results. If the connection was successful, various information about the server is displayed:



If the connection to the OPC Classic Tunnel Server is not possible an error message indicating the error will be logged in the *Server Connection Test* dialog:

© Softing Industrial Automation GmbH

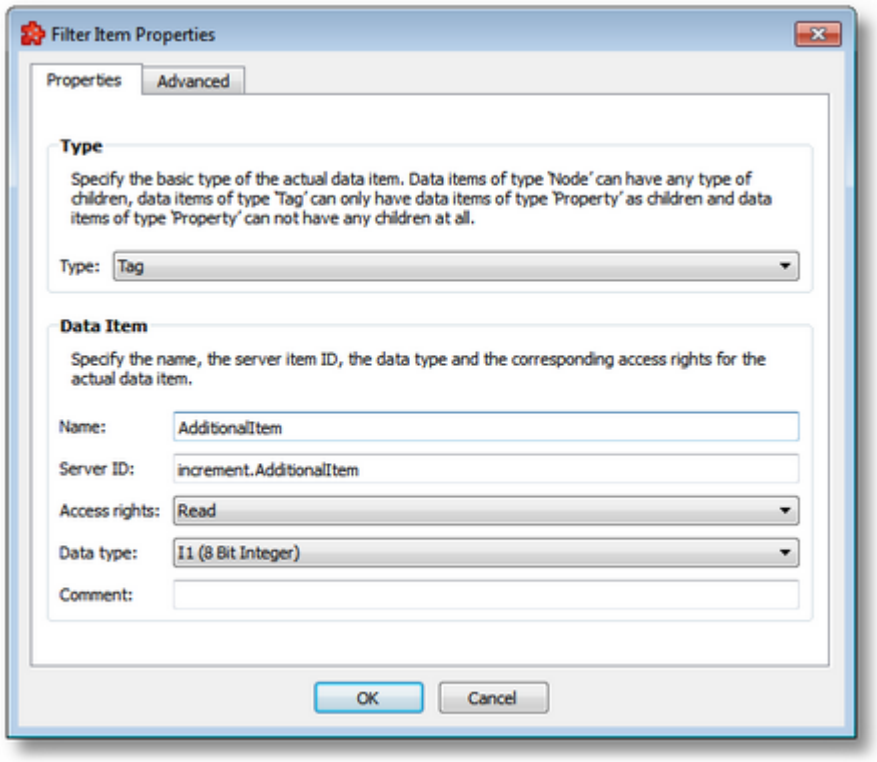


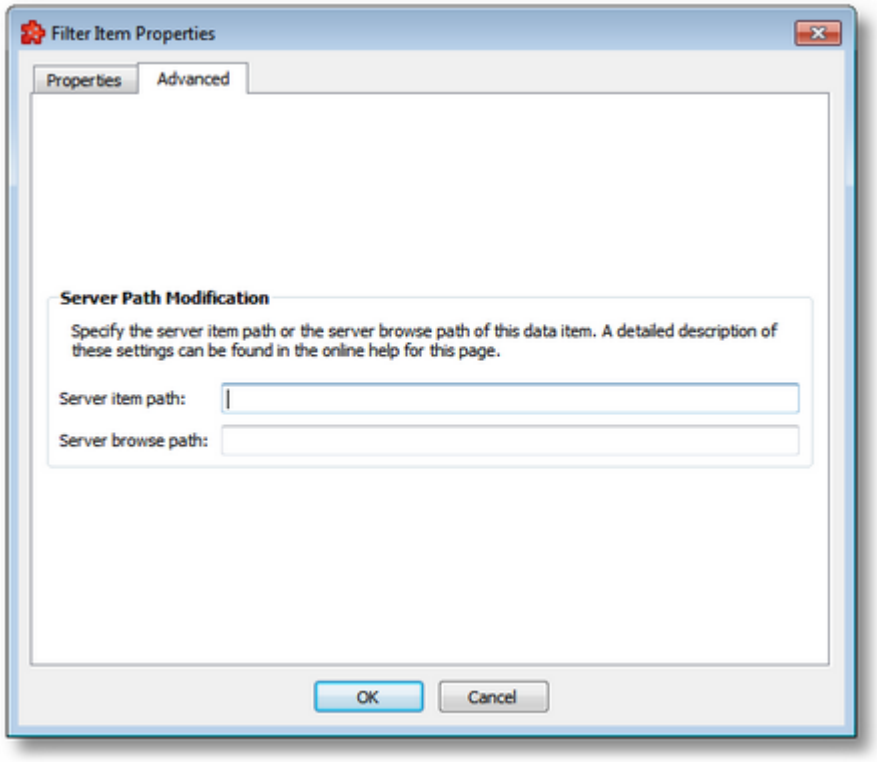
7.3.1.9.4 Filter Item Properties

The properties of the *Filter Item* can be set when adding Filter Items and can be modified when editing previously added Filter Items.

Add a New Filter Item

The add operation is triggered by selecting the parent node and then pressing the *Add a new item* button  , using the shortcut or selecting the *Add* action from the context menu. When adding a new Filter Item the following two-tabbed dialog is displayed:





Note: When you select items of type *Node* for which no *Access Rights* parameter has been set (empty value), the only configurable parameters are the *Name* and the *Access Rights*.

The available configuration options in this dialog are described in the table below:

© Softing Industrial Automation GmbH

Attribute	Description
Type	The type of the data item. The possible types are: *Node*, *Tag* and *Property*.
Name	The Tunnel item name identifying the corresponding data item inside the dataFEED OPC Suite namespace.
Path	The full OPC Classic Tunnel item path for the data item inside the dataFEED OPC Suite namespace.
Access Right	The access rights for the data item. The possible values are: *Read*, *Write*, *Read* and *Write*.
Data Type	The data type of the data item. The possible values are described in the table below.
Comment	An optional comment/note can be attached to the current data item by using this field.
Server path - data item	The OPC path for the data item on the external OPC Classic Tunnel server.
Server path - browsing	The OPC browse path for the data item, describing the individual browse steps for the data item in the namespace of the external OPC Classic Tunnel server.

Note: Items of type *Node* may have child items of any existing type.

Note: Items of type *Tag* may have child items of type *Property*.

Note: Items of type *Property* cannot have any children. Therefore, Add is not available for properties.

The Tunnel filter data items of the namespace may be of any data type covered by the OPC Specifications. The following table describes the available data types mapped to the corresponding VARIANT types from the OPC Specification.

Data type	VARIANT type	Description
I1	*VT_I1*	8 bit signed integer.
UI1	*VT_UI1*	8 bit unsigned integer.
I2	*VT_I2*	16 bit signed integer.
UI2	*VT_UI2*	16 bit unsigned integer.

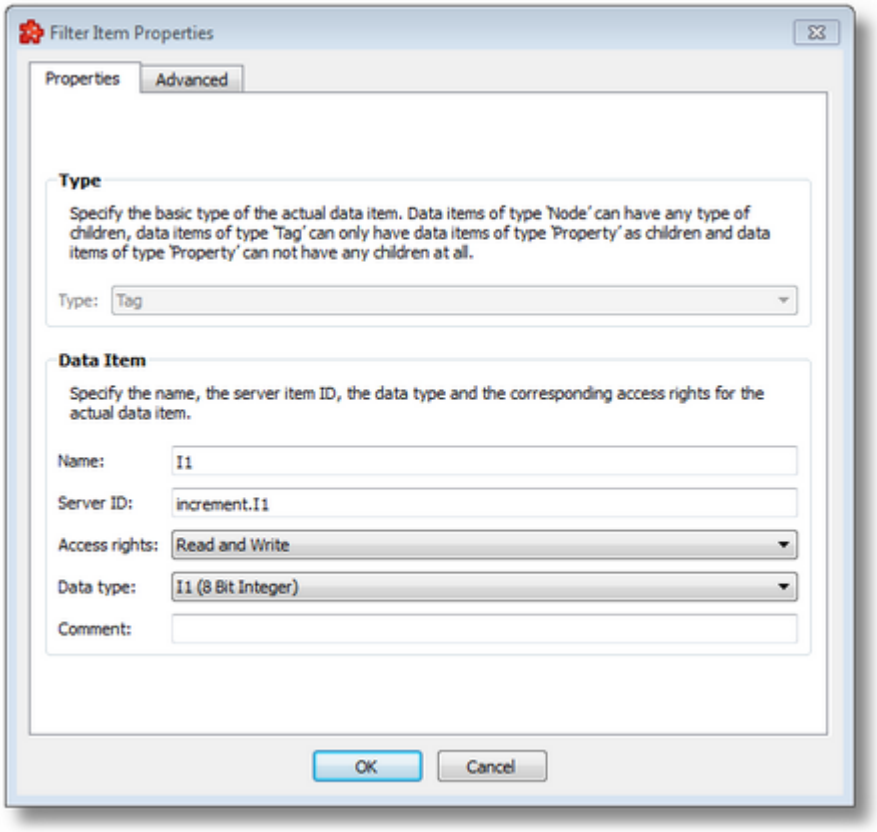
I4	*VT_I4*	32 bit signed integer.
UI4	*VT_UI4*	32 bit unsigned integer.
I8	*VT_I8*	64 bit signed integer.
UI8	*VT_UI8*	64 bit unsigned integer.
R4	*VT_R4*	32 bit floating point.
R8	*VT_R8*	64 bit floating point.
BOOL	*VT_BOOL*	Boolean.
STRING	*VT_BSTR*	String.
DATETIME	*VT_DATE*	Date and time.
DECIMAL	*VT_CY*	Decimal.
ANY	*VT_VARIANT*	Any data type.

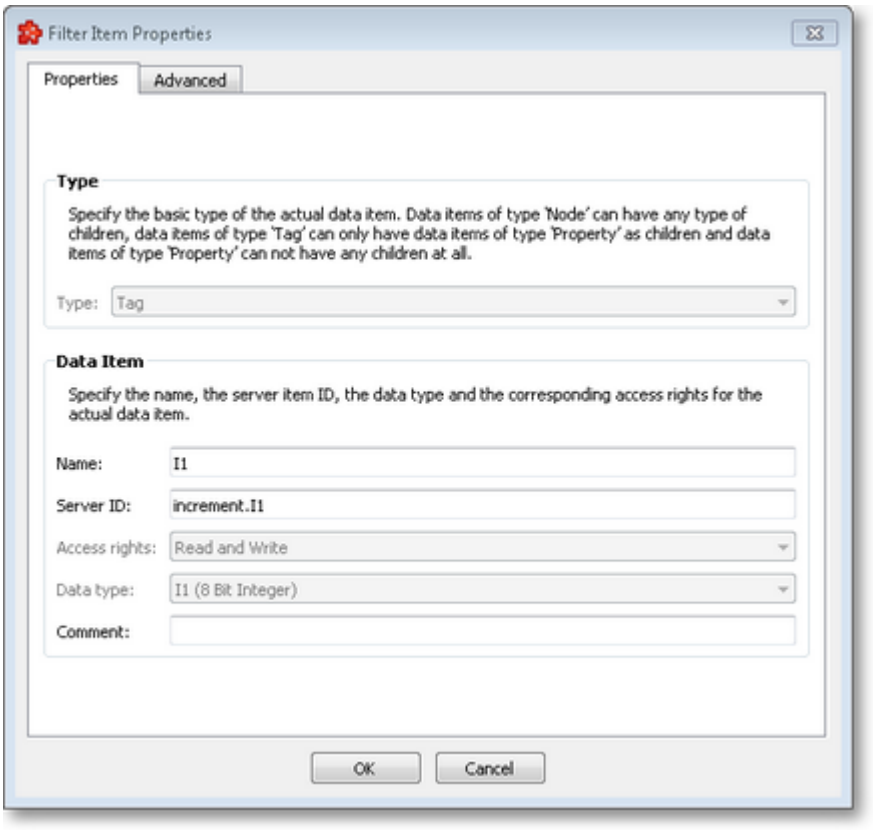
Note: Arrays of fixed and variable length are also supported for these data types.

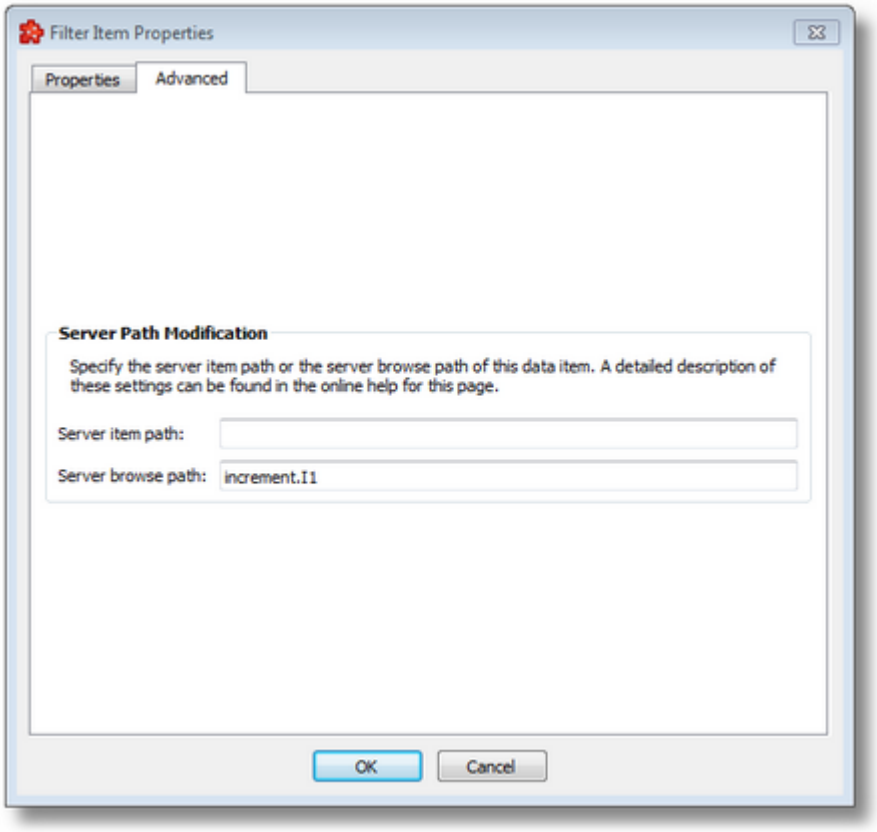
Modify a Filter Item

The modify operation is triggered by selecting the node to be modified and then pressing the *Properties* button  , using the shortcut or selecting the *Properties* action from the context menu. When modifying an existing Filter Item the following two-tabbed dialog is displayed:

© Softing Industrial Automation GmbH


Filter Item Properties
Properties
Advanced
Type
Specify the basic type of the actual data item. Data items of type 'Node' can have any type of children, data items of type 'Tag' can only have data items of type 'Property' as children and data items of type 'Property' can not have any children at all.
Type: Tag
Data Item
Specify the name, the server item ID, the data type and the corresponding access rights for the actual data item.
Name: I1
Server ID: increment.I1
Access rights: Read and Write
Data type: I1 (8 Bit Integer)
Comment:
OK
Cancel


Filter Item Properties
Properties
Advanced
Type
Specify the basic type of the actual data item. Data items of type 'Node' can have any type of children, data items of type 'Tag' can only have data items of type 'Property' as children and data items of type 'Property' can not have any children at all.
Type: Tag
Data Item
Specify the name, the server item ID, the data type and the corresponding access rights for the actual data item.
Name: I1
Server ID: increment.I1
Access rights: Read and Write
Data type: I1 (8 Bit Integer)
Comment:
OK
Cancel



The configuration options are the same as in the above described *Add a New Filter Item* dialog.

Note: The *Type* of the data item cannot be changed anymore.

Note: The *Access rights* and *Data type* parameters of a Filter item are disabled and cannot be changed if the item is being used by other consumer add-ons.

Note: The *Modify* action is also triggered by double-clicking on tree items without children.

7.3.1.9.5 Filter File

Each of the OPC Classic Tunnel server connections configured to use a filter file will produce a file containing the definition of the filter data items when the configuration is saved.

The files are stored in the *Filter* subfolder of the configuration folder. The filter file is named after the server connection name. The filter files are plain text files, encoded in UTF-8 (thus providing full UNICODE compatibility) and they have a line-oriented structure. The elements of a line are tab-separated. For each filter data item a line is added to the file.

The following line types can be found in the Tunnel server connection filter file:

- Filter item line, defining a filter data item as described in the table below.

© Softing Industrial Automation GmbH

- Empty line.
- Comment line, starting with #; such a line is ignored during evaluation.

A filter item line has the following format:

```
<TAB>Softing dataFEED OPC Suite Server ID<TAB>Softing dataFEED OPC Suite Item ID<TAB>
```

The different elements of the line are described in the following table:

Element	Description
Softing dataFEED OPC Suite Server ID	Not used. Should be always empty string.
Softing dataFEED OPC Suite Item ID	The Tunnel item name identifying the corresponding data item inside the dataFEED OPC Suite namespace.
Server item ID	The OPC item ID for the data item in the external OPC Classic Tunnel server namespace.
Server item path	The OPC path for the data item on the external Tunnel server.
Server browse path	The OPC browse path for the data item, describing the individual browse steps for the data item in the namespace of the external OPc Classic Tunnel server.
Data type	The data type of the data item. The possible values are described in the Filter Item Properties page.
Access Rights	The access rights for the data item. The possible values are: *r*, *w*, *rw* corresponding to read, write, read and write, respectively.
Object Type	The type of the data item. The possible types are: *Node*, *Tag* and *Property*.
Comment	An optional comment/note attached to the current data item.

Example:

```
#<TAB>Softing dataFEED OPC Suite server ID<TAB>Softing dataFEED OPC Suite item ID<TAB
#
<TAB><TAB>maths.automatic angle change<TAB>maths.automatic angle change<TAB><TAB>math
<TAB><TAB>maths.automatic change period<TAB>maths.automatic change period<TAB><TAB>ma
<TAB><TAB>maths.cos<TAB>maths.cos<TAB><TAB>maths.cos<TAB>R8<TAB>r<TAB>Tag<TAB>Cosinus
<TAB><TAB>maths.cos|102<TAB>maths.cos#102<TAB><TAB>maths.cos|102<TAB>R8<TAB>rw<TAB>Pr
<TAB><TAB>maths.cos|103<TAB>maths.cos#103<TAB><TAB>maths.cos|103<TAB>R8<TAB>rw<TAB>Pr
<TAB><TAB>maths.cos|6020<TAB>maths.cos#6020<TAB><TAB>maths.cos|6020<TAB>UI2<TAB>rw<TA
<TAB><TAB>maths.sin<TAB>maths.sin<TAB><TAB>maths.sin<TAB>R8<TAB>r<TAB>Tag<TAB>Sinus d
<TAB><TAB>maths.sin|102<TAB>maths.sin#102<TAB><TAB>maths.sin|102<TAB>R8<TAB>r<TAB>Pro
```

```
<TAB><TAB>maths.sin|103<TAB>maths.sin#103<TAB><TAB>maths.sin|103<TAB>R8<TAB>r<TAB>Pro
<TAB><TAB>maths.tan<TAB>maths.tan<TAB><TAB>maths.tan<TAB>R8<TAB>r<TAB>Tag<TAB>
<TAB><TAB>maths.tan|6020<TAB>maths.tan#6020<TAB><TAB>maths.tan|6020<TAB>UI2<TAB>rw<TA
```

7.3.1.9.6 Tunnel Server Wizard

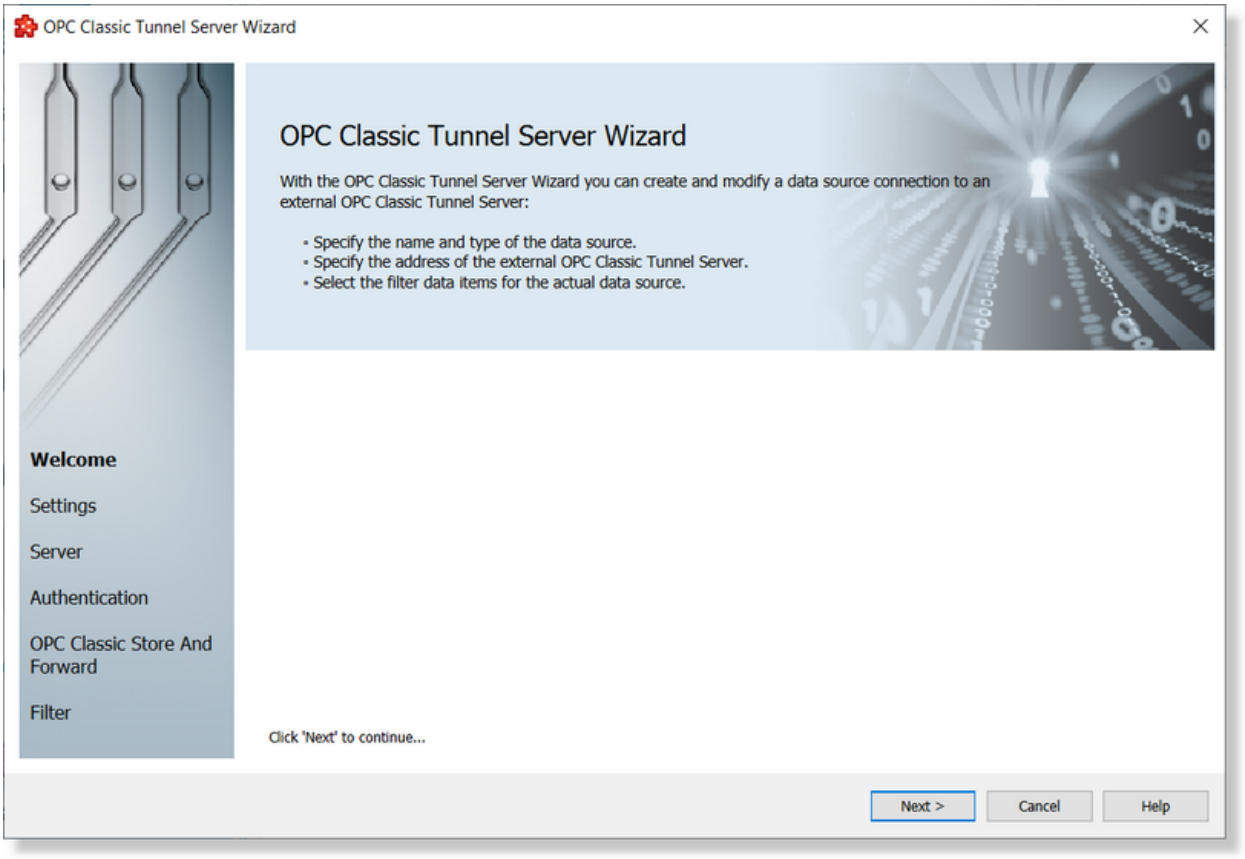
With the OPC Classic Tunnel Server Wizard you can create and modify a data source connection to an external OPC Classic Tunnel server:

- Obtain general information about the Tunnel Server Wizard @ OPC Classic Tunnel Server Wizard - Welcome
- Specify the name and the type of the Tunnel Server @ OPC Classic Tunnel Server Wizard - Settings
- Select the address of the external Tunnel server @ OPC Classic Tunnel Server Wizard - Server
- Specify the user name and the password for the external Tunnel server @ OPC Classic Tunnel Server Wizard - Authentication
- Decide if the connection should be a Store And Forward connection @ OPC Classic Tunnel Server Wizard - Store And Forward
- Activate or deactivate the filter functionality for the connection @ OPC Classic Tunnel Server Wizard - Filter
- Select the corresponding filter data items @ OPC Classic Tunnel Server Wizard - Filter Items

A short overview of the actions needed to define an OPC Classic Tunnel server connection is presented. This page requests no input as it only shows what the wizard allows:

- Specifying the name and type of the external OPC Classic Tunnel server connection.
- Specifying the address of the external OPC Classic Tunnel server.
- Choosing the filter data items for the selected OPC Classic Tunnel server.

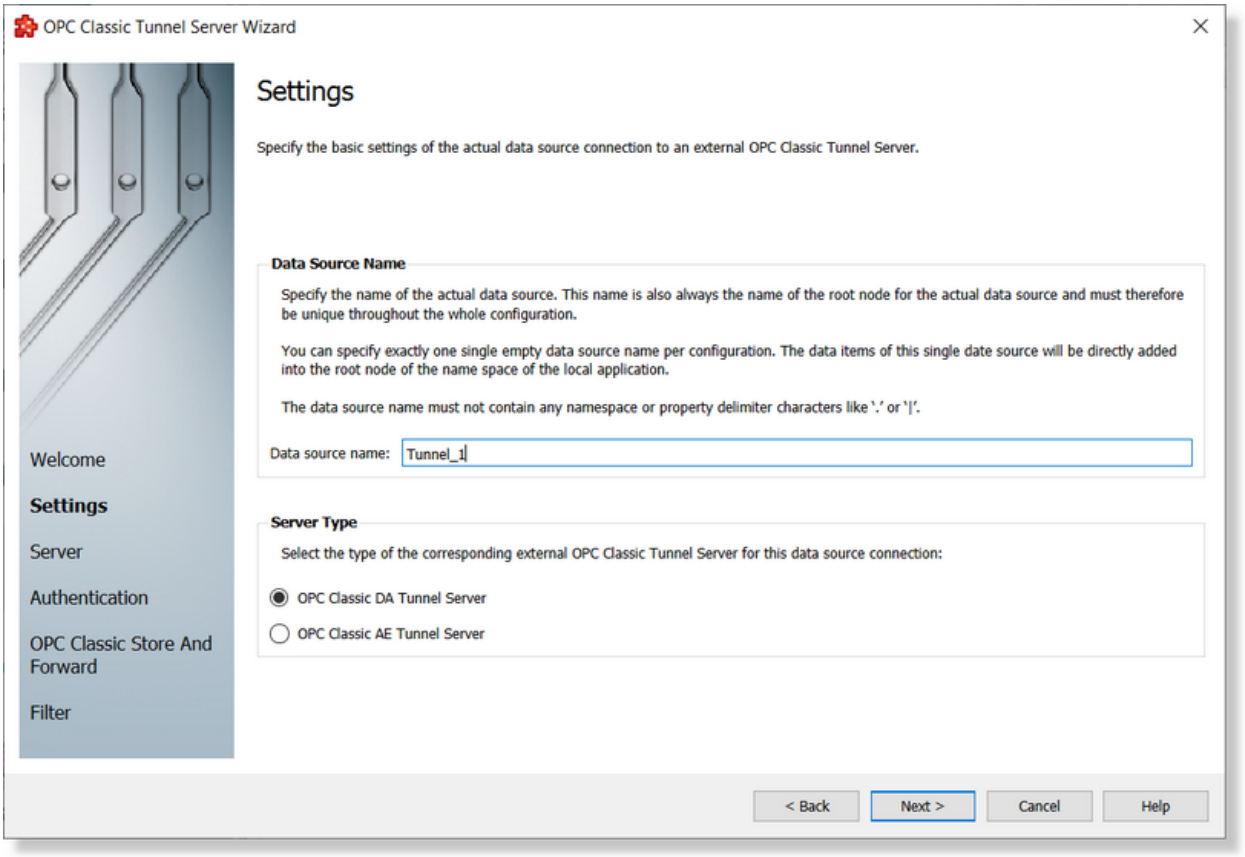
© Softing Industrial Automation GmbH



Press the *Next* button to proceed with the wizard or abort the operation with the *Cancel* button.

Tunnel Server Wizard - Settings >>>

On the Settings wizard page the Tunnel server connection name is defined, which is the server's branch in the global namespace.



Attention: The *Data Source Name* must be unique.

If you try to proceed to the next page and the OPC Classic Tunnel server connection name is a duplicate, an error dialog will ask you to enter a different item prefix. When adding a new Tunnel server connection, you can choose the type of the Tunnel server: *OPC Classic DA Tunnel* or *OPC Classic AE Tunnel* server connection. When modifying an existing OPc Classic Tunnel server connection, the type of the server cannot be changed and is therefore disabled (grayed).

Note: If the data source name is left empty, the root item for this data source connection will be shown in the configurator namespace tree as *<EmptyDA>* or as *<EmptyAE>* depending on the corresponding OPC Classic Tunnel Server type. However, in the real namespace of the local dataFEED OPC Suite application the corresponding data items will be added directly under the root node of the namespace of the local dataFEED OPC Suite application.

Note: It is possible to define two different Tunnel server connections with no data source name, but only if the two server connections have different types: DA and AE, respectively.

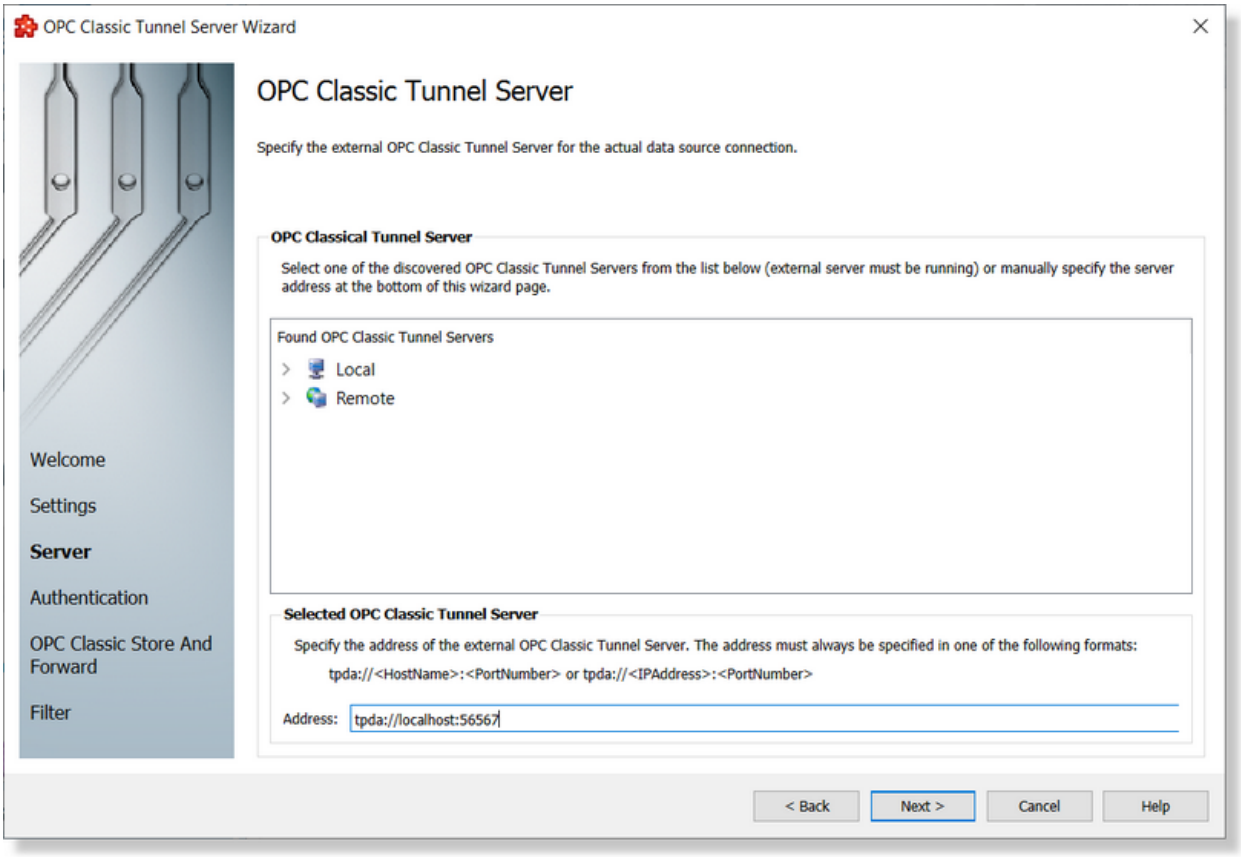
Continue to the next wizard page by pressing the *Next* button, return to the previous one by using the *Back* button, or cancel the wizard by clicking the *Cancel* button.

<<< Tunnel Server Wizard - Welcome Tunnel Server Wizard - Server >>>

© Softing Industrial Automation GmbH

This wizard page allows choosing the external OPC Classic Tunnel server to connect to. There are two ways to specify a Tunnel server. The simplest way is to choose an OPC Classic Tunnel server registered on the local computer or on another computer from the network, using the tree control provided in the page.

The second option is to manually type the server address in the provided *Address* input control at the bottom of the page. The address must use a specific format as described in the OPC Classic Tunnel Server Address Format section below.



OPC Classic Tunnel Server Address Format

To connect to a Tunnel server the connection type, computer name or IP address and the IP port must be specified. The address of an OPC Classic Tunnel server has the following format:

```
<ConnectionType>://[<HostName>|<IPAddress>][:<IPPort>]
```

The *<ConnectionType>* placeholder is a pre-defined string, the corresponding values are shown in the table below:

<ConnectionType>	Description
tpda	OPC Classic Tunnel Data Access

tpae	OPC Classic Tunnel Alarms & Events

Note: To connect to an OPC Classic Tunnel server running on the same PC, the host name *localhost* or the IP address *127.0.0.1* should be used as host name or IP address.

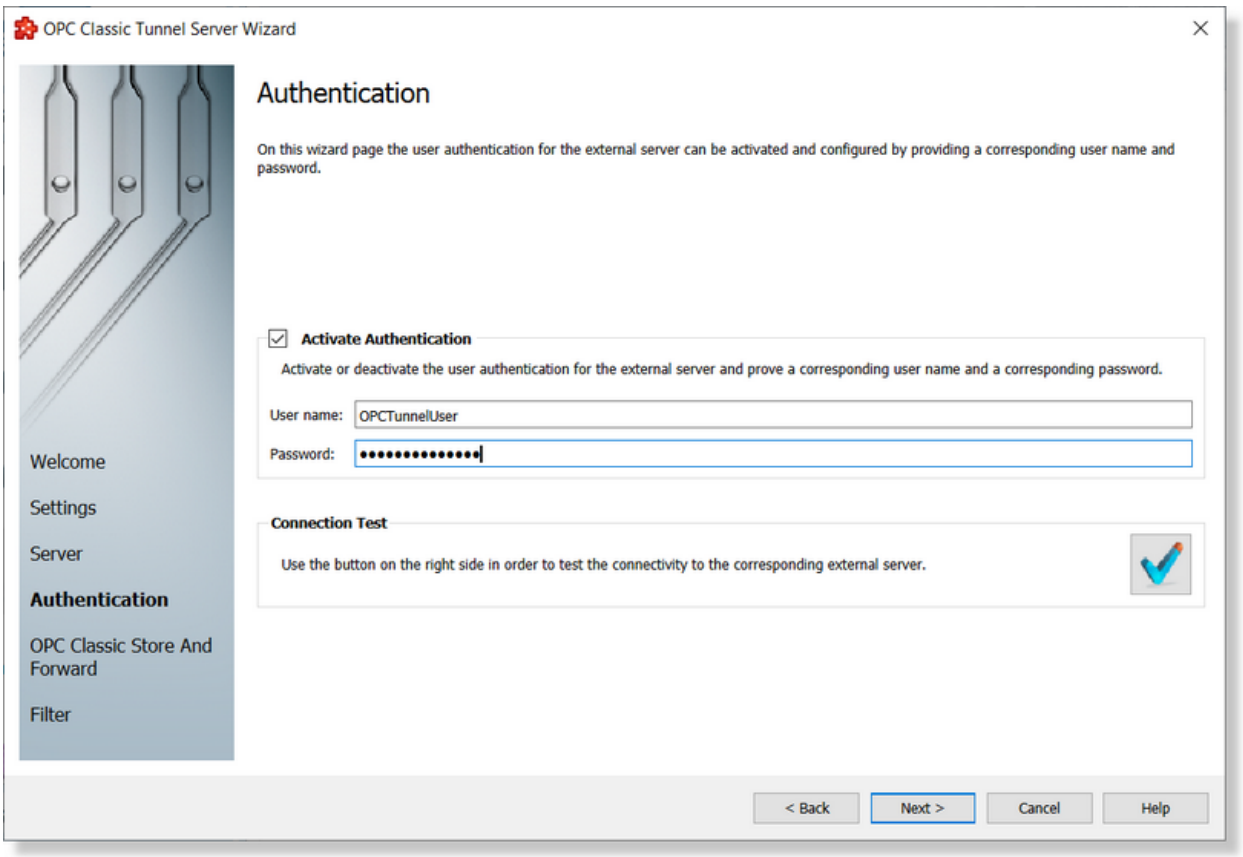
Examples:

```
tpda://localhost:56567
tpda://192.168.150.101:9950
tpae://WKSTA02:56766
```

Continue to the next wizard page by pressing the *Next* button, return to the previous one by using the *Back* button, or discard the changes made so far by clicking the *Cancel* button.

<<< Tunnel Server Wizard - Settings Tunnel Server Wizard - Authentication >>>

This wizard page prompts for credentials to access the selected server if necessary. If the credentials are necessary, the *Activate Authentication* checkbox should be checked to enable the *User name* and *Password* fields for editing.



© Softing Industrial Automation GmbH

Activate Authentication

By checking or unchecking the Activate Authentication checkbox the user authentication for the external Tunnel server can be activated or deactivated and a corresponding user name and password can be specified.

Connection Test

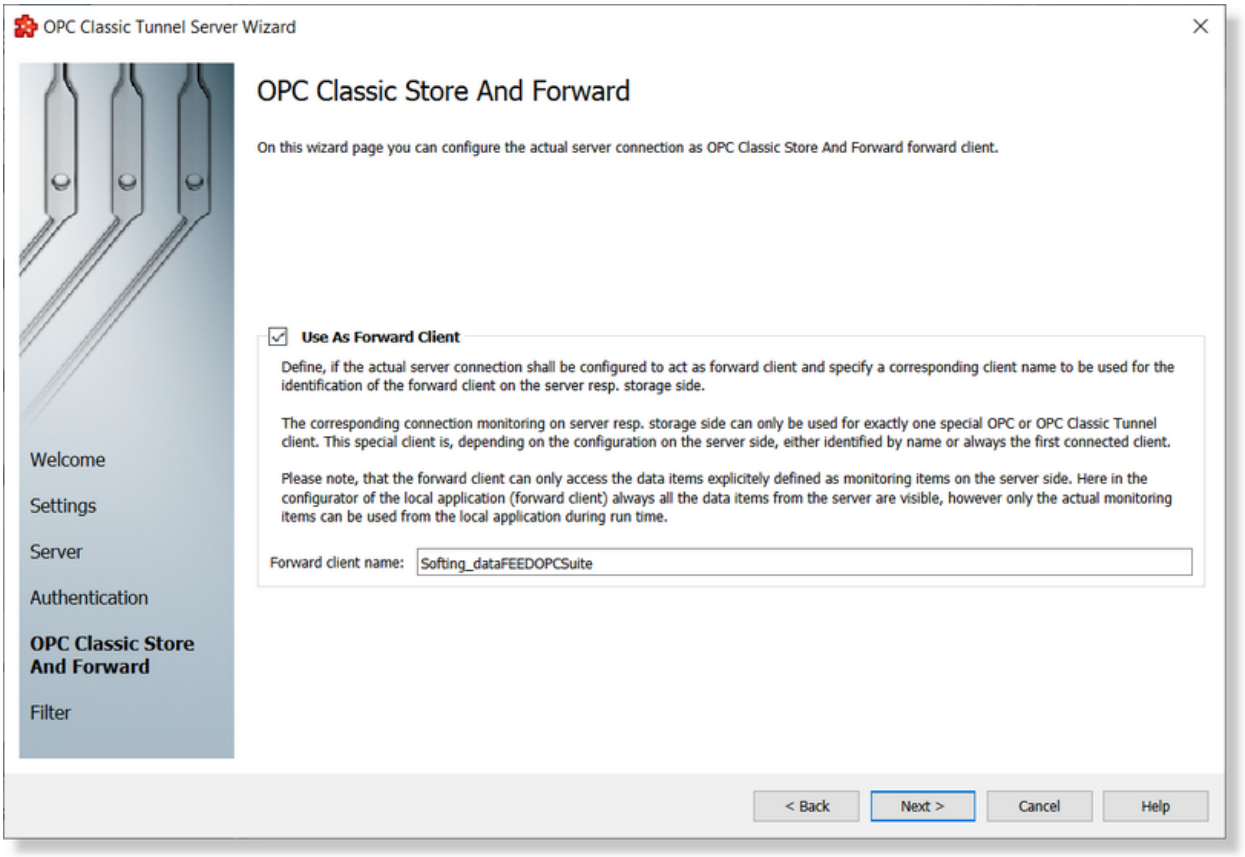
This page also offers the possibility to test the connection to the current external Tunnel server. For the actual test the specified user name and password are used.

To start the connection test, use the *Test* button  in the *Connection Test* section. A new dialog will appear on screen showing the connection test results. The *Connection Test* dialog is described on the Connection Test page of the Tunnel Server chapter under Data Sources.

Note: A Tunnel AE server connection cannot be used as a Store And Forward connection and does not provide the possibility to define filter items. So for Tunnel AE server connections the *Next* button is replaced by a *Finish* button.

<<< Tunnel Server Wizard - Server Tunnel Server Wizard - Store And Forward >>>

On this wizard page you can configure the selected server connection as OPC Classic Store And Forward client:



By checking the *Use As Forward Client* checkbox, the Tunnel server connection is configured to act as a forward client.

In the *Forward client name* input field you can specify the name of the forward client which the corresponding OPC Classic Tunnel server (storage side) will use to identify the client as the forward client.

The corresponding connection monitoring on the server/storage side can only be used for a single, special OPC or OPC Classic Tunnel client. Depending on the configuration on the server side this special client is either identified by the client name or as the first connected client.

Please note that the forward client can only access the data items explicitly defined as monitoring items on the server side. Here in the configurator of the local dataFEED OPC Suite application (forward client) all the data items from the server are always visible, but only the actual monitoring items can be used from the local dataFEED OPC Suite application during run time.

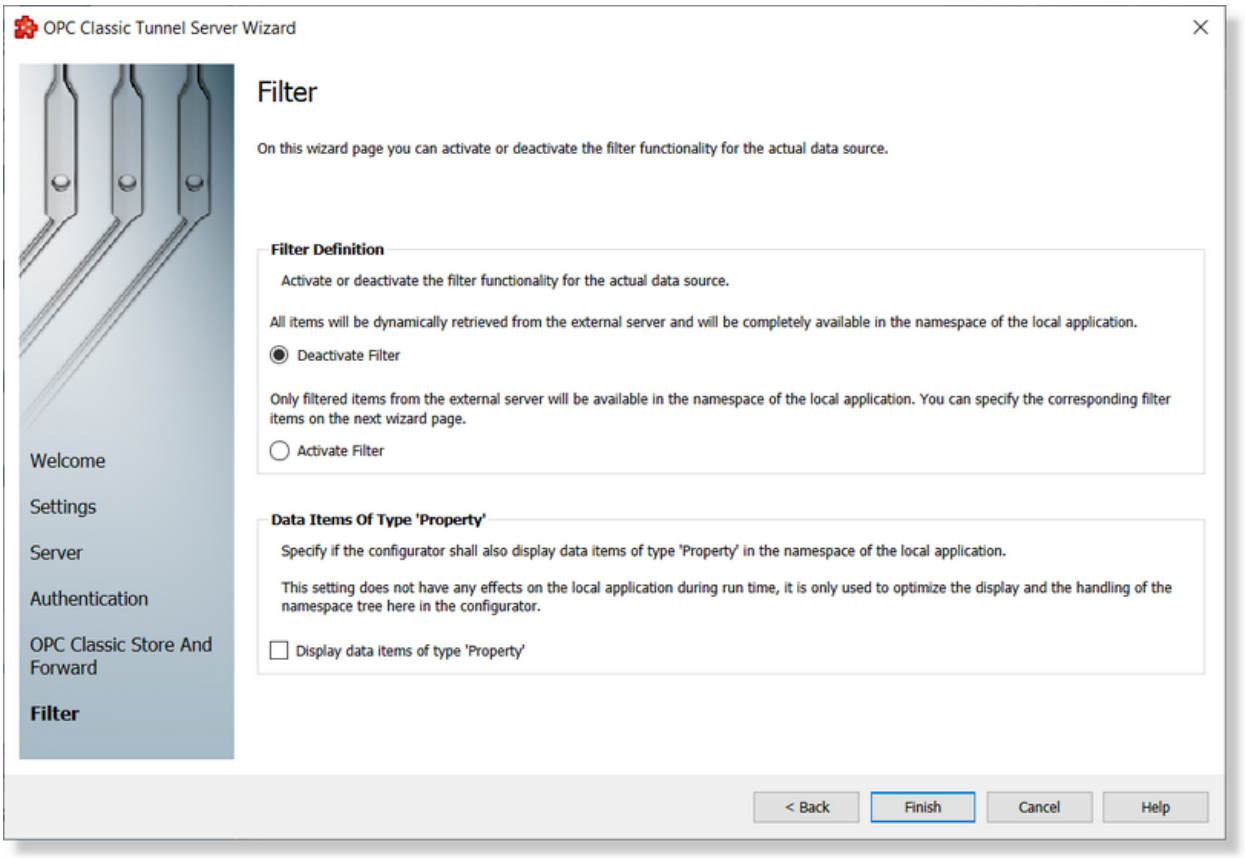
Attention: The forward client can only access the data items explicitly defined as monitoring items on the server side. If the forward client tries to access data items which are not defined as monitoring items it will always receive the corresponding value with the OPC quality *BAD (Configuration Error)*.

For more details on how to configure Store And Forward see the Configuration: Data Processing - OPC Classic Store And Forward page in the OPC Classic Store And Forward chapter under Data Processing.

© Softing Industrial Automation GmbH

<<< Tunnel Server Wizard - Authentication Tunnel Server Wizard - Filter >>>

This wizard page allows the activation or deactivation of the filter functionality for the selected OPC Classic Tunnel server connection.



Filter Definition

For the *Filter Definition* there are two possible configuration settings:

Filter Definition	Description
Deactivate Filter	All items will be dynamically retrieved from the external server and will be completely available in the namespace of the local dataFEED OPC Suite application.
Activate Filter	Only filtered items from the external server will be available in the namespace of the local dataFEED OPC Suite application. You can specify the filter items on the next wizard page.

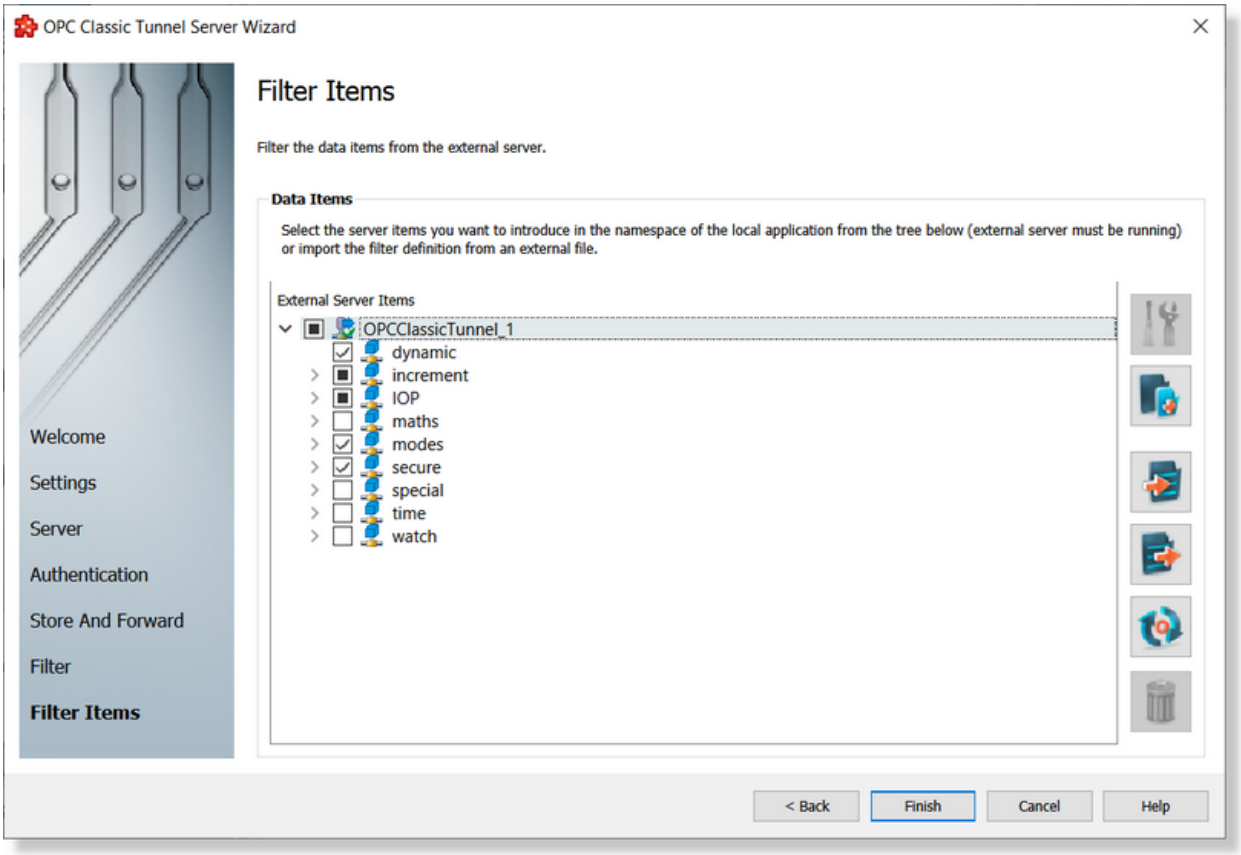
Another option of this wizard page allows enabling or disabling the display of data items of *Property* type in the namespace of the local dataFEED OPC Suite application.

Attention: This setting does not have any effect on the local dataFEED OPC Suite application during run time, it is only used to optimize the display and the handling of the namespace tree in the dataFEED OPC Suite Configurator.

Depending on the *Filter Definition* selection the *Next* button will be available to continue the OPC Classic Tunnel Server Wizard with the *Filter Items* page (when the filter functionality was activated) or the *Finish* button will be available allowing you to finish the OPC Classic Tunnel Server Wizard and save the changes.

<<< Tunnel Server Wizard - Store And Forward Tunnel Server Wizard - Filter Items >>>

On this wizard page the filter data items for the connection to the actual external OPC Classic Tunnel server can be defined. Select the server data items you want to introduce into the namespace of the local dataFEED OPC Suite application from the corresponding tree control (external server must be running) or import the filter definition from an external file.



If the external OPC Classic Tunnel server is running, the namespace of the external OPC Classic Tunnel server is shown in the tree under the root node named after the actual connection. The tree is expandable and each item can be selected to be included in the filter file or not. The selection is downward recursive, which means that when you include a node all

© Softing Industrial Automation GmbH

subsequent nodes, tags and properties in that particular branch of the namespace tree will automatically be included.

If the connection to the OPC Classic Tunnel server failed, the namespace tree will only contain the server root node. However, it is possible at any time to try reconnecting to the external server and refreshing the data by using the *Refresh data items* button  .

Operation

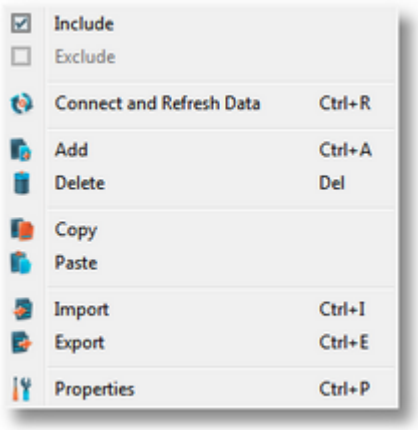
Depending on the current selection in the *External Server Items* tree, the following operations are possible using the buttons on the right, the context sensitive menu or the corresponding shortcuts:

- Properties of the selected item (Ctrl+P) 
- Add a new item (Ctrl+A) 
- Import from file (Ctrl+I) 
- Export to file (Ctrl+E) 
- Refresh data items (Ctrl+R) 
- Delete data items (Del) 

Two more operations are supported, which are only accessible from the contextual menu:

- Copy data item 
- Paste to data item 

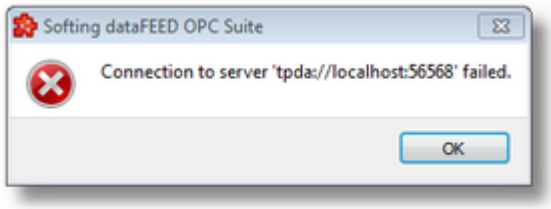
An example of the contextual menu available on *Data Items* tree nodes is shown below:



Refresh data items

Refresh data items (Ctrl+R)  operation (or the corresponding *Connect and Refresh Data* context menu action) will try to reconnect to the Tunnel server to refresh the data items tree.

If the action succeeds the namespace tree will be refreshed in order to reflect possible changes to the remote Tunnel server namespace. If the connection to the remote Tunnel server fails, an error message box will be displayed:



Delete data items

Triggering *Delete data items*  action with the corresponding button on the right wizard side or *Delete* from the context menu will remove the currently selected node from the *Data Items* namespace tree. The action has to be confirmed in the information dialog box which shows up.

Attention: The root node of the namespace tree can not be deleted.

Attention: A node which is used in a Processing or Data Destination action can not be deleted.

Copy

© Softing Industrial Automation GmbH

The *Copy*  action is provided in the context menu of the *External Server Items* tree. Triggering this action will copy the selected node to the local clipboard. All the properties of the node are copied. Also, if the node contains children, the entire branch is copied.

Note: The root node of the namespace tree does not provide a *Copy* action.

Paste

The *Paste*  action is provided in the context menu of the *External Server Items* tree. This action will add the node currently stored in the local clipboard as a a child of the currently selected node. Whether the action is available in the context menu depends on the type of the currently selected node and the type of the node in the clipboard.

Note: The *Paste* action is only available if a node was previously copied to the clipboard.

If the *Paste* action is performed on a node which already has a child of the same name as the node in the clipboard, the pasted node is renamed automatically by adding a numerical suffix to the original name.

```
E.g. NodeA, NodeA01, NodeA02, ...
```

<<< Tunnel Server Wizard - Filter

7.3.1.9.7 Import Wizard

The Softing Configurator allows the import of OPC Classic Tunnel filter items from an external file.

The external file should be a plain text tab-separated file with the specific OPC Classic Tunnel filter file format.

The import action can be triggered by using the *Import from file* button  , by pressing the keyboard shortcut sequence CTRL+I or by selecting the *Import* action in the context sensitive menu.

Note: The import action is available for the root of the OPC Classic Tunnel server connection namespace and for items of type *Node* from the namespace.

Attention: The import action is not available for items of type *Tag* and *Property*.

Note: The names conflicts between the old and new data items are solved by renaming the newly imported items. They are renamed by appending a two-digit number at the end of the name of the first duplicated item.

When the import action is triggered the *Filter Import Wizard* is started.

▾ The Filter Import Wizard allows importing different data items:

- Select the file to be imported @ Import Wizard - Import File Selection
- Examine the import results @ Import Wizard - Import Results

Each of the OPC Classic Tunnel server connections configured to use a filter file will produce a file containing the definition of the filter data items when the configuration is saved. The files are stored in the *Filter* subfolder of the configuration folder. The filter file is named after the server connection name. The filter files are plain text files, encoded in UTF-8 (thus providing full UNICODE compatibility) and they have a line-oriented structure. The elements of a line are tab-separated. For each filter data item a line is added to the file.

The following line types can be found in the Tunnel server connection filter file:

- Filter item line, defining a filter data item as described in the table below.
- Empty line.
- Comment line, starting with #; such a line is ignored during evaluation.

A filter item line has the following format:

```
<TABSofting server ID<TAB>Softing item ID<TAB>Server item ID<TAB>Server item path<TAB
```

The different elements of the line are described in the following table:

Element	Description
Softing server ID	Not used. Should be always an empty string.
dataFEED OPC Suite item ID	The OPC item name identifying the corresponding data item inside the dataFEED OPC Suite namespace.
Server item ID	The OPC item ID for the data item on the external OPC server namespace.
Server item path	The OPC path for the data item in the external OPC server.
Server browse path	The OPC browse path for the data item, describing the individual browse steps for the data item in the namespace of the external OPC server.
Data type	The data type of the data item. The possible values are described in the Filter Item Properties page.

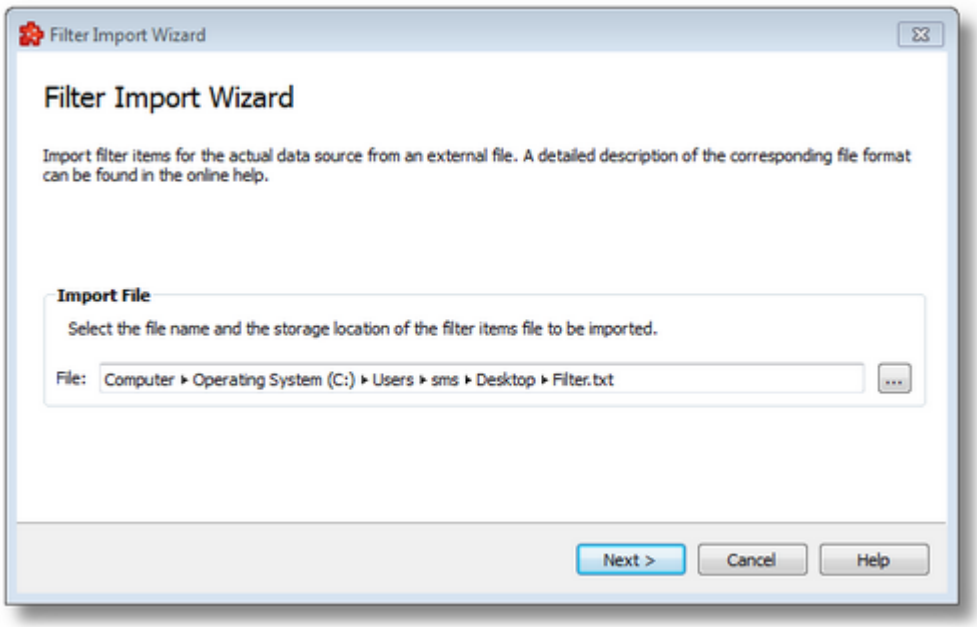
© Softing Industrial Automation GmbH

Access rights	The access rights for the data item. The possible values are: r, w, rw corresponding to read, write, read and write, respectively.
Object type	The type of the data item. The possible types are: Node, Tag and Property.
Comment	An optional comment/note attached to the current data item.

Example:

```
#<TAB>Softing server ID<TAB>Softing item ID<TAB>Server item ID<TAB>Server item path<T
#
<TAB><TAB>maths.automatic angle change<TAB>maths.automatic angle change<TAB><TAB>math
<TAB><TAB>maths.automatic change period<TAB>maths.automatic change period<TAB><TAB>ma
<TAB><TAB>maths.cos<TAB>maths.cos<TAB><TAB>maths.cos<TAB>R8<TAB>r<TAB>Tag<TAB>Cosinus
<TAB><TAB>maths.cos|102<TAB>maths.cos#102<TAB><TAB>maths.cos|102<TAB>R8<TAB>rw<TAB>Pr
<TAB><TAB>maths.cos|103<TAB>maths.cos#103<TAB><TAB>maths.cos|103<TAB>R8<TAB>rw<TAB>Pr
<TAB><TAB>maths.cos|6020<TAB>maths.cos#6020<TAB><TAB>maths.cos|6020<TAB>UI2<TAB>rw<TA
<TAB><TAB>maths.sin<TAB>maths.sin<TAB><TAB>maths.sin<TAB>R8<TAB>r<TAB>Tag<TAB>Sinus d
<TAB><TAB>maths.sin|102<TAB>maths.sin#102<TAB><TAB>maths.sin|102<TAB>R8<TAB>r<TAB>Pro
<TAB><TAB>maths.sin|103<TAB>maths.sin#103<TAB><TAB>maths.sin|103<TAB>R8<TAB>r<TAB>Pro
<TAB><TAB>maths.tan<TAB>maths.tan<TAB><TAB>maths.tan<TAB>R8<TAB>r<TAB>Tag<TAB>
<TAB><TAB>maths.tan|6020<TAB>maths.tan#6020<TAB><TAB>maths.tan|6020<TAB>UI2<TAB>rw<TA
```

The Import File Selection wizard page allows you to choose a file to be imported. The file can be selected by manually typing its name in the *File* input box or by clicking the browse button located to the right of the input box (a standard file opening dialog will open).

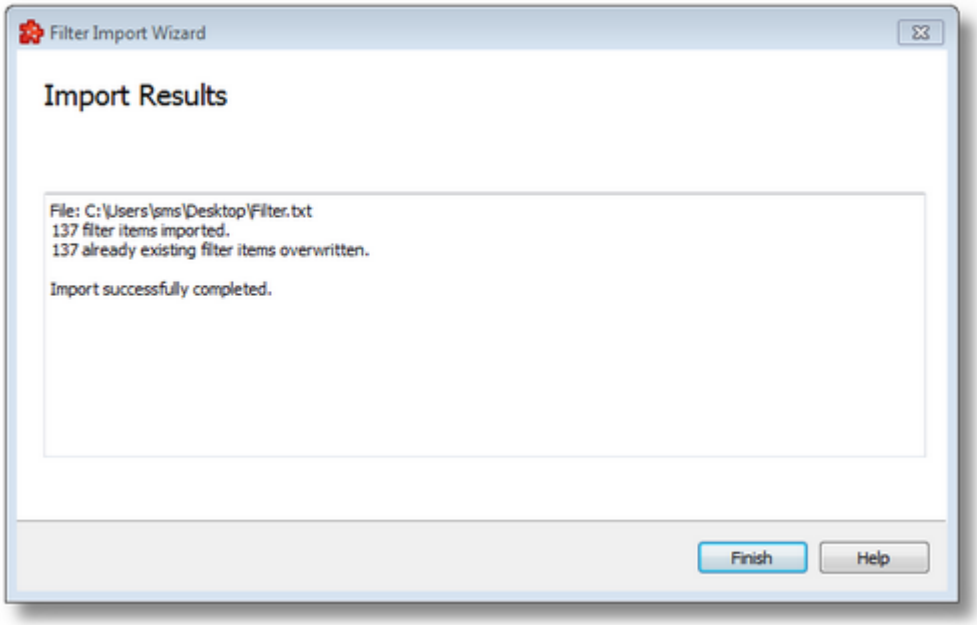


The file path and name should be valid (they should point to an existing file).

When you press the *Next* button the file path and name are verified. If the file identified by the specified file path and name does not exist an error message is displayed. The actual import begins if the selected file is valid. If the import takes longer to complete a waiting dialog is displayed. When the import operation is finished the *Import Results* page is displayed.

Import Wizard - Import Results >>>

The Import Results wizard page displays the import results.



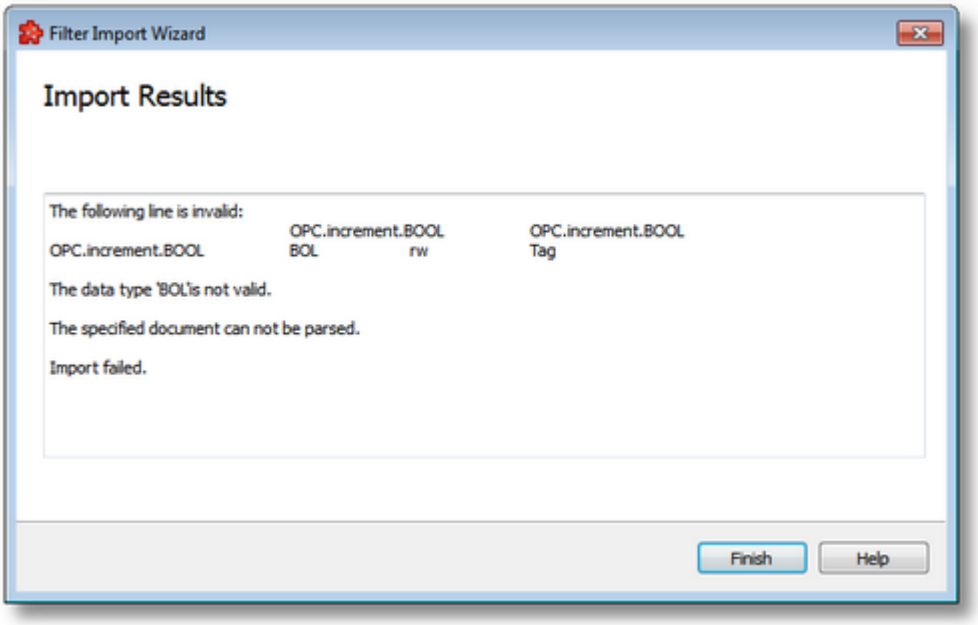
This page is the final page of the *Filter Import Wizard*. A back button is not provided as the import action cannot be undone. The import can be successful or it can fail:

1. After a successful import the results page contains the following information:
   - The absolute path of the imported file.
   - The number of local items imported. This number includes only the items of type Tag, the items of type Node are not counted.
   - The confirmation message that the import was successful.

2. After a failed import the results page contains the following information:
   - The invalid line where the import failed.
   - The reason why the line is considered invalid.
   - The confirmation message that the import failed.

© Softing Industrial Automation GmbH

An *Import Results* example for a failed import operation can be seen below:



When the import wizard is closed after a successful import, the new items are added to the current Tunnel server connection tree as a branch of the currently selected node item.

<<< Import Wizard - Import File Selection

7.3.1.9.8 Export Wizard

The dataFEED OPC Suite Configurator allows exporting the filter items belonging to a OPC Classic Tunnel server connection. The entire namespace tree or just a sub-branch of the OPC Classic Tunnel server connection can be exported.

The resulting external file will be a tab-separated file with the specific OPC Classic Tunnel filter file format.

The export action can be triggered by pushing the *Export* button  , by pressing the keyboard shortcut CTRL+E or by selecting the *Export* action in the context sensitive menu. The export action is available for the root of the Tunnel server connection namespace and for any element of type *Node* of the namespace.

Attention: The export action is not available for nodes of type *Tag* or *Property*.

The exported file will contain all the filter items from the branch of the selected node.

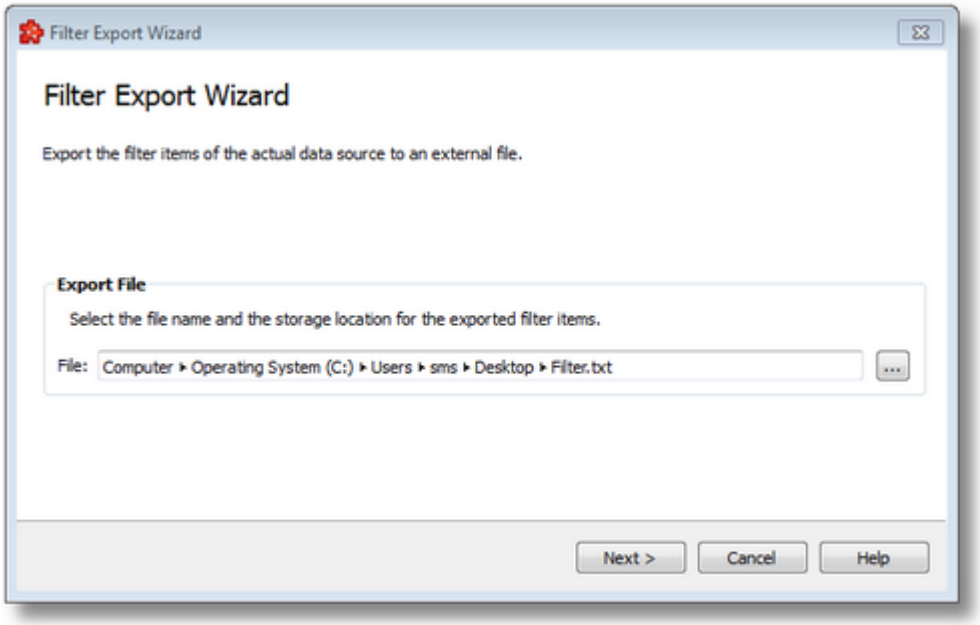
Note: The exported file will contain only the selected (checked) data items.

When the export action is triggered the *Filter Export Wizard* is started.

▼ The Filter Export Wizard allows exporting different data items:

- Select the file the data items are to be exported to @ Export Wizard - Export File Selection
- Examine the export results @ Export Wizard - Export Results

The Export File Selection wizard page allows you to choose a file name and location for the exported file. The file can be selected by manually typing its path and name in the *File* input box or by clicking the browse button located to the right of the input box (a standard file saving dialog will open).



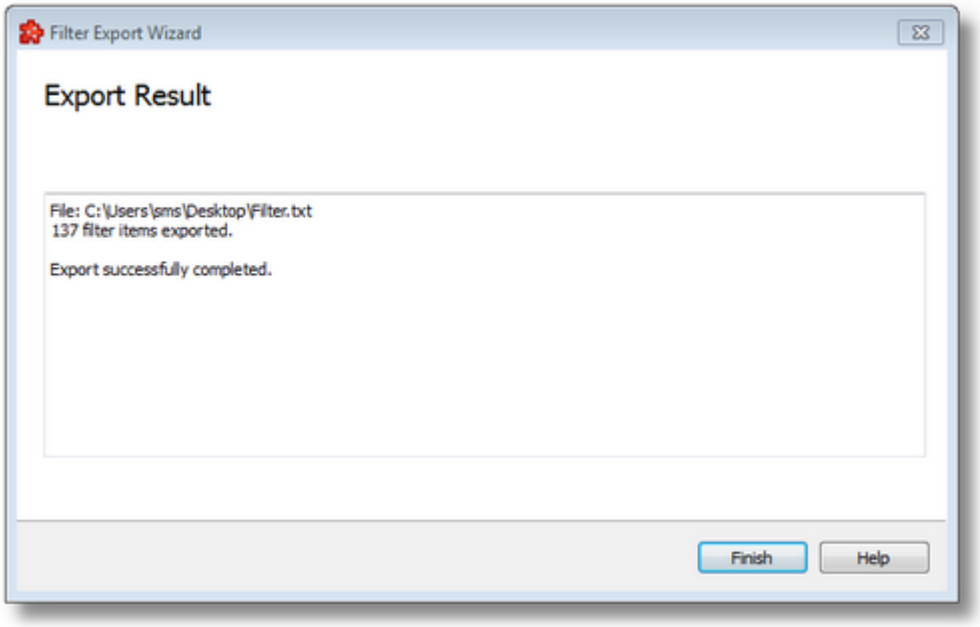
When the *Next* button is pressed the file path is verified and if the path does not exist, an error message is displayed (invalid path). If the specified file path is valid and a file with the specified name already exists, a message box will ask whether to overwrite the existing file or not (select another file). The actual filter export operation begins if the path to the file is valid and either a file with the specified name does not exist or it exists and overwrite has been chosen.

When the export operation is finished the *Export Results* page is displayed.

Export Wizard - Export Results >>>

The Export Results wizard page displays the export results as shown in the image below:

© Softing Industrial Automation GmbH



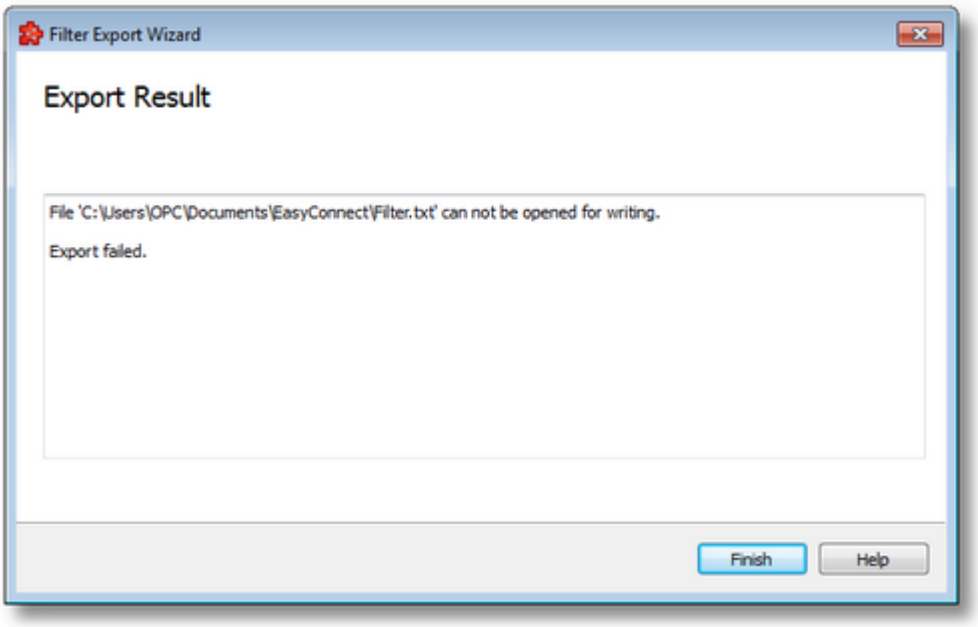
This page is the final page of the Filter Export Wizard. A back button is not provided as the export action cannot be undone. The export can be successful or it can fail:

1. After a successful export the results page contains the following information:
   - The absolute path of the exported file.
   - The number of filter items exported. This number includes only the items of type Tag and Property, the data items of type Node are not counted.
- The confirmation message that the export was successful.

2. After a failed export the results page contains the following information:

- The reason why the export failed.
- The confirmation message that the export failed.

An *Export Results* example for a failed export operation can be seen below:

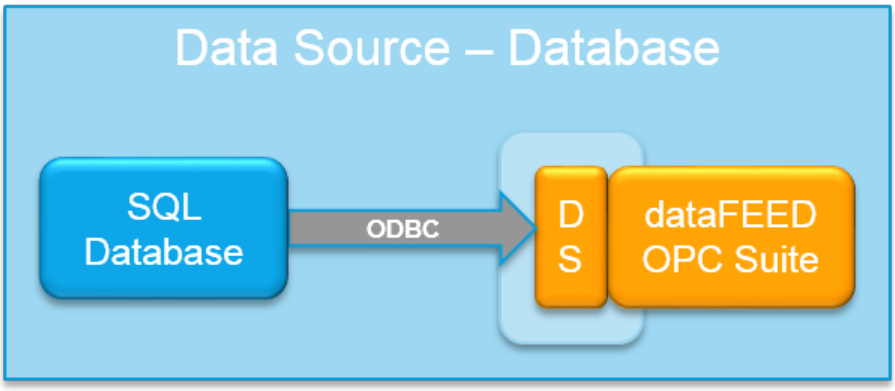


See the Import Wizard chapter for a description of the import operation.

<<< Export Wizard - Export File Selection

7.3.1.10 Database Read

This section of the Softing dataFEED OPC Suite Online Help describes how to configure the database read functionality.



Database Read

Database Read can exchange data between its local address space (data provided by data sources) and databases. It supports any type of database to which an ODBC connection can be established through a Data Service Name (DSN). The data read by Softing dataFEED OPC Suite can be written into items via SQL statements or stored procedures.

© Softing Industrial Automation GmbH

Softing dataFEED OPC Suite allows the following operations to be done:

- Configure the connection to the desired database
- Define SELECT FROM statements as actions to be triggered
- Define Stored Procedure and Function calling actions to be triggered
- Define general SQL statements as actions to be triggered
- Transfer of OPC data from any data source server to a Database server
- Transfer data from Database into writable items.

The Database Read add-on can be activated or deactivated by using the *Activate Database Read* checkbox. When deactivating the Database Read add-on, the configured database connections and actions are not lost.

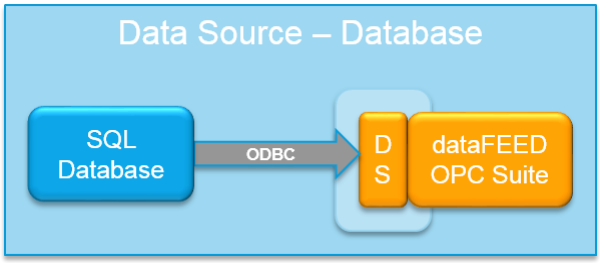
Connecting to a database is done through *Open Database Connectivity* (ODBC) which provides a standard software interface for accessing the most common types of databases like: Microsoft SQL Server, MySQL, Oracle. Thus, dataFEED OPC Suite uses ODBC to execute SQL commands on a database. It accomplishes this by using a driver as a translation layer between dataFEED OPC Suite and database management system (DBMS). Each driver is configured with a DSN that defines the connection to the target database.

The DSN connection string must contain the following:

- Database location: a host name or IP address and optionally a database name and port number
- Logon information, if it is required by the database

A more detailed description can be found on the Configuration: Data Source - Database Read page.

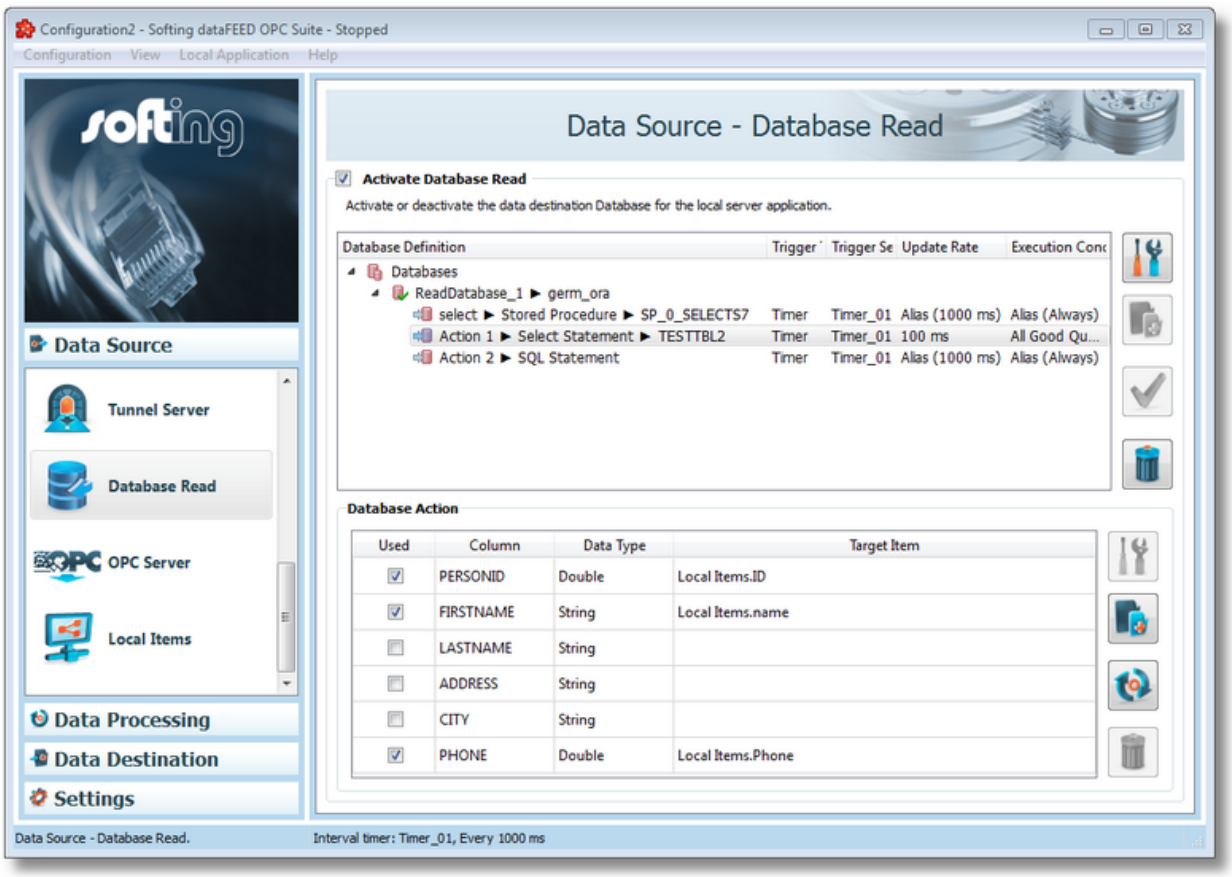
7.3.1.10.1Configuration



The Database Read functionality is configured in the *Data Source - Database Read* main user interface. The main add-on UI page can be reached by choosing the *Database Read* option from the *Data Source* section in the navigation panel on the left. The Database Action section provides different controls depending on what type of database action is selected in the database tree. If a database connection or the root *Databases* node is selected then this section will be disabled. Three types of database actions exist:

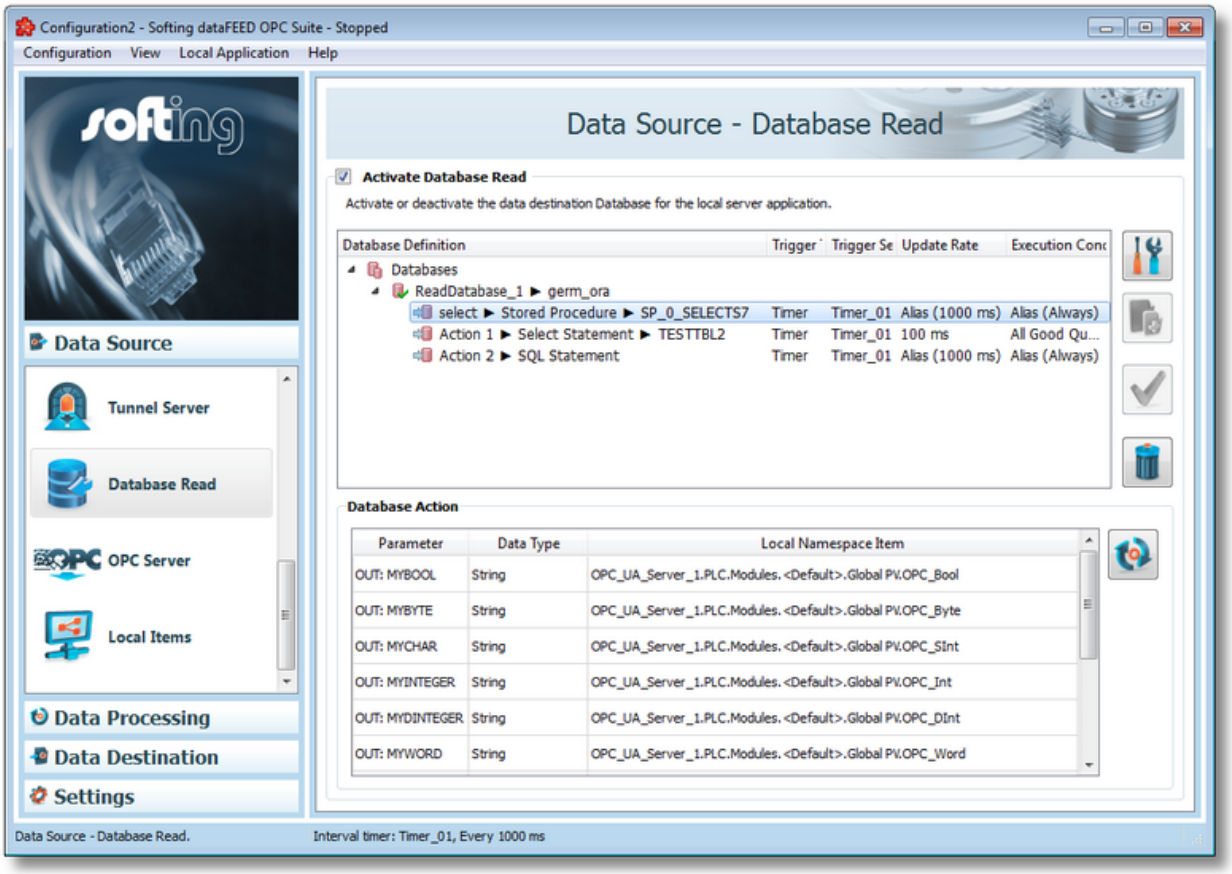
- *SELECT FROM* SQL statement which defines an Select action in a user friendly way.
- *Stored Procedure* is an action editor which allows defining actions which call stored procedures or function in a user friendly way.
- *General SQL Statement* is an action editor which allows typing any SQL query code and assigning items.

The image below illustrates the main user interface of the Database Read add-on (Select database action selected):

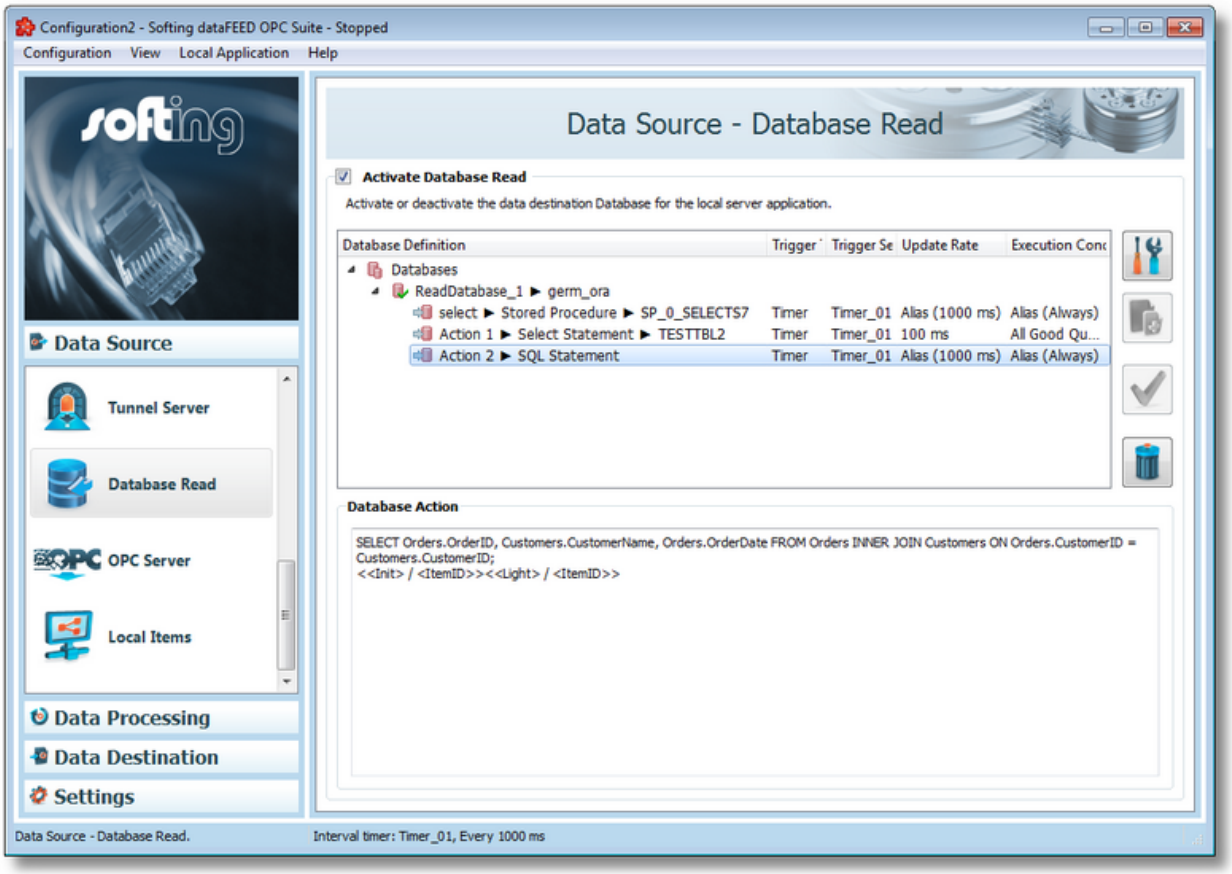


This image represents a Stored Procedure or Function statement database action (Function currently selected):

© Softing Industrial Automation GmbH



The image below represents a general SQL statement database action:



For more details about the Database Read functionalities see the Database Read page.

The entire *Database Read* add-on functionality can be activated or deactivated by using the *Activate Database Read* checkbox.

Note: If database definitions or database actions are defined and the add-on functionality is deactivated, these definitions will not be deleted from the configuration files (task and subscription files). Instead they are marked as deactivated - prefixed with the deactivated mark: *#!* These lines are then ignored by the local dataFEED OPC Suite application.

By default the Database Read functionality is disabled. It needs to be enabled first by selecting the *Activate Database Read* checkbox at the top of the *Data Source - Database Read* main UI page in order to use it.

The *Data Destination - Database Read* main view consists of:

- The *Activate Database Read* checkbox which allows enabling / disabling this add-on.
- A tree displaying the database definitions and database actions related to them.
- An action editor which displays details about or offers support in editing the selected database action.

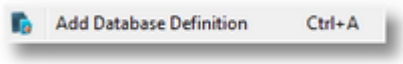
Database definition handling

© Softing Industrial Automation GmbH

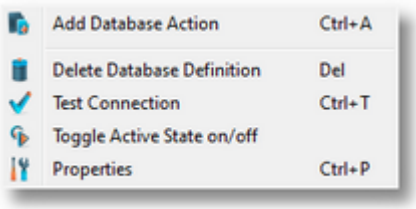
From the Data Source - Database Read main user interface, certain operations can be performed using the buttons on the right, keyboard shortcuts or the corresponding context menu actions:

- Add a new database definition by selecting the *Databases* root element and using the *Add* button  , pressing the keyboard shortcut CTRL+A or selecting the *Add* action from the root element context menu.
- Change the properties of the selected database definition using the *Properties* button  , pressing the keyboard shortcut CTRL+P or selecting the *Properties* action from the context menu.
- Add a new action for the selected database definition using the Add button  , pressing the keyboard shortcut CTRL+A or selecting the Add action from the database definition context menu.
- Change the properties of the selected database action using the *Properties* button  , pressing the keyboard shortcut CTRL+P or selecting the *Properties* action from the context menu.
- Remove the selected database definition / database action using the *Delete* button  , pressing the DELETE / DEL keyboard key or selecting the *Delete* action from the context menu.
- Modify the trigger type for the selected database action by selecting the *Modify Trigger Type* action from the context menu.
- Modify the trigger settings for the selected database action by selecting the *Modify Trigger Settings* action from the context menu. For more information see the Item Trigger Wizard or the Timer Trigger Wizard.

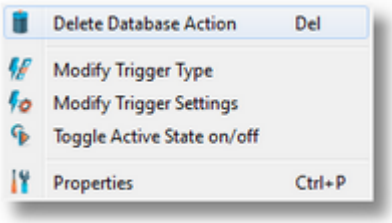
The context menu of the *Databases* root node looks like this:



The *Database Definition* context menu looks like this:



The *Database Action* context menu looks like this:



Each Database connection and action can be activated or deactivated separately using Toggle Active State on/off option. Once deactivated the item is grayed out inside the view.

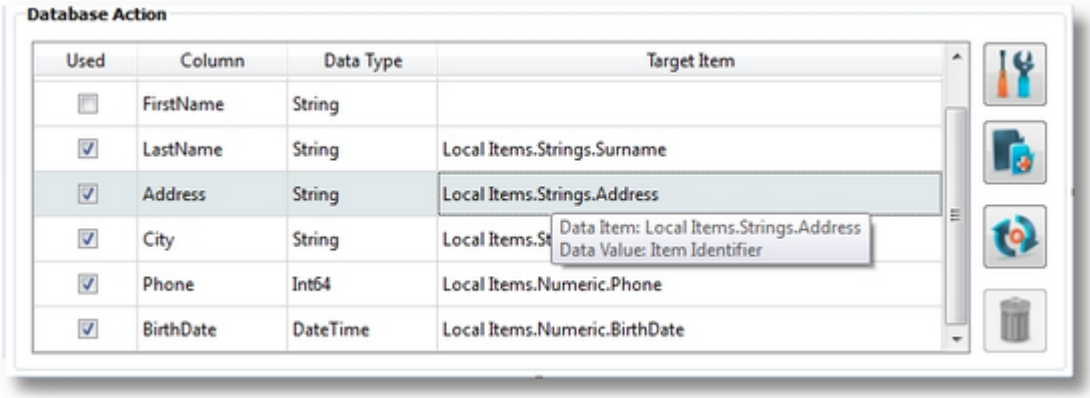
For database action definitions in the Database tree the trigger type, the update rate and the execution condition can be defined. These values can be edited by clicking in the corresponding cell.

Action editor handling for SELECT FROM actions

The SELECT FROM action editor offers a convenient way to edit a *Select From* SQL statement by displaying an interactive table grid containing the table column names as rows with their respective data associations. The action editor table has the following fields:

- *Used* - column which determines whether the respective column will receive any data from the incoming transaction. When you select the checkbox in this cell the Data Value Placeholder Wizard will open. In this wizard you can associate the table column name with a placeholder. When the checkbox is unchecked this association will be lost and the defined placeholder will automatically be deleted.
- *Column* - column name in the selected database definition from which the value will be read.
- *Data Type* - column displaying the data type of the database table column.
- *Data Value* - column in the interactive table which displays the user defined placeholder. If the column *Used* is not checked this column will contain no placeholder reference.

An example of a complex action editor containing selected and unselected rows of both mandatory and non-mandatory types can be seen below:



For more details see the Database Action Wizard - Select Statement page.

© Softing Industrial Automation GmbH

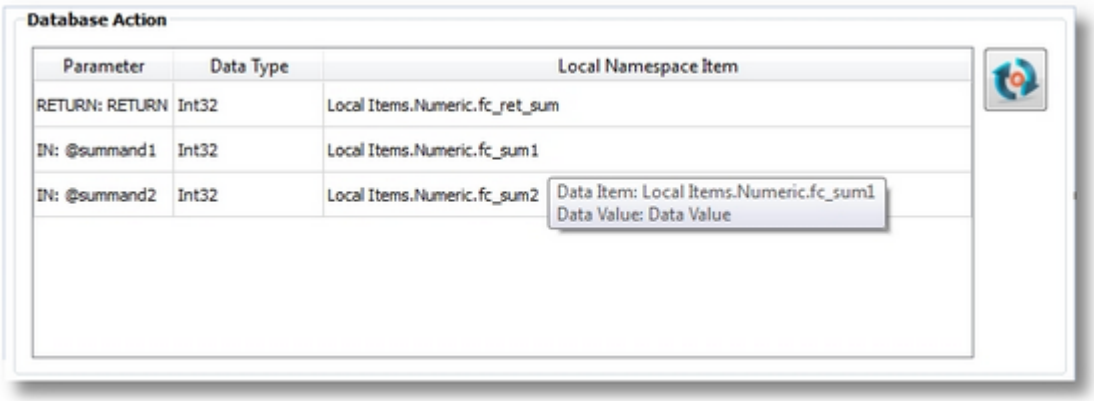
Note: dataFEED OPC Suite displays texts encoded in UTF-8. The Unicode characters are thus fully supported for Database names or table column names.

Attention: Due to Qt-specific limitations, database data types cannot be detected properly. Qt enumerates certain sub-types into the same categories, thus these sub-types will not be treated as those types.
This means that for example the database types SQL_DECIMAL, SQL_NUMERIC, SQL_REAL, SQL_FLOAT, SQL_DOUBLE all fall into the Double type category and the application will treat all of them as Double.

This behavior can usually be seen in the main view.

Action editor for Stored Procedure or Function actions

The *Stored Procedure or Functions Action* editor shows the parameter, data type and local namespace items in a read-only view. No editing is available, like for Select statements.



The table above shows the list of parameters for the selected Stored Procedure or Function action, with their direction before their names, data types and the item names to which they are assigned. The Refresh button on the right side of the table can be used to refresh the parameter meta data from the database.

If the application finds that parameters of the stored procedure or function don't exist anymore, those rows will be marked as invalid, with color red.

For more details see the Database Action Wizard - Stored Procedure page.

Action editor for General SQL Statement actions

The *General SQL Statement* editor does not offer the possibility to edit a general SQL statement. It only shows a read-only view of the SQL action.



As an example, the following statement first runs an initialization statement, then it creates an Insert Into action and executes them on a database. dataFEED OPC Suite replaces the placeholders accordingly:

```
use Test_Database GO INSERT INTO Test_Table (itemId,itemValue)  VALUES (<<R8> / <Item
```

For more details see the Database Action Wizard - SQL Statement page.

Note: Configuration files are encoded in UTF-8, thus providing full support of Unicode characters.

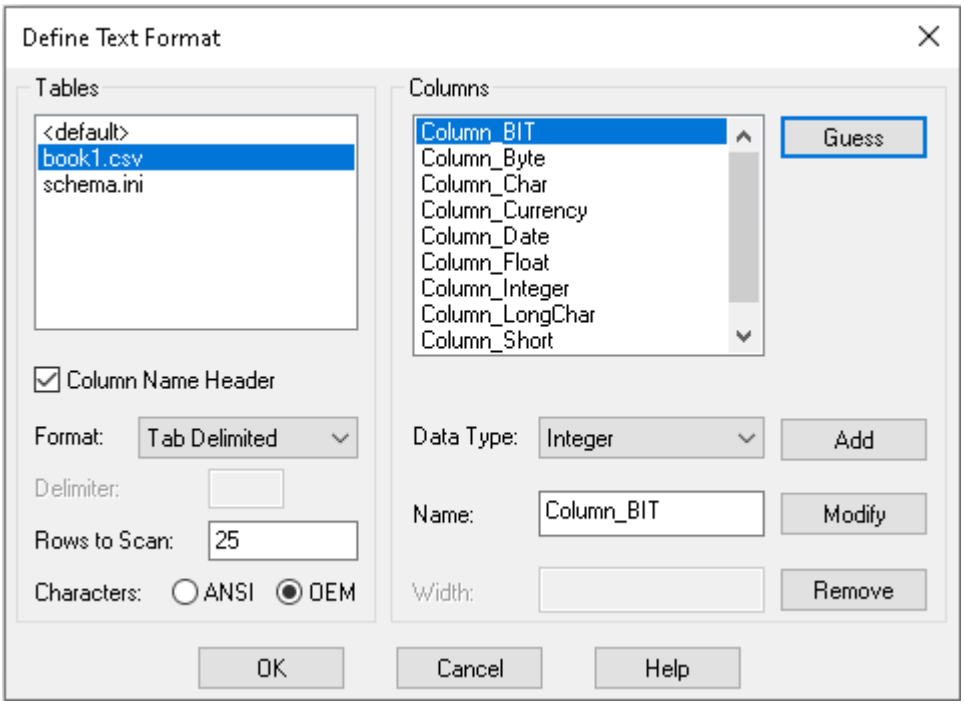
.txt / .csv data Read through Microsoft ODBC Text Driver

Connecting to .txt / .csv files is done through *Open Database Connectivity* (ODBC) 32-bit. To configure the ODBC connection follow the steps:

- From System DSN, create a new Data Source by selecting for example the default Microsoft Text Driver
- Add a Data Source Name and select the location of the source .txt / .csv file
- Press the Options and Define Format buttons to open the Define Text Format dialog
- Continue by defining the source file specific settings: usage of Column Name Header, Format (tab ,comma delimited), data-types of the columns, etc

See below an ODBC definition example to a local .csv file TAB separated, containing columns with all supported Microsoft ODBC Text Driver data types.

© Softing Industrial Automation GmbH



The schema.ini file will contain:

ColNameHeader=True

Format=TabDelimited

MaxScanRows=25

CharacterSet=OEM

Col1=COLUMN_BIT Bit

Col2=COLUMN_BYTE Byte

Col3=COLUMN_CHAR Char Width 255

Col4=COLUMN_CURRENCY Currency

Col5=COLUMN_DATE Date

Col6=COLUMN_FLOAT Float

Col7=COLUMN_INTEGER Integer

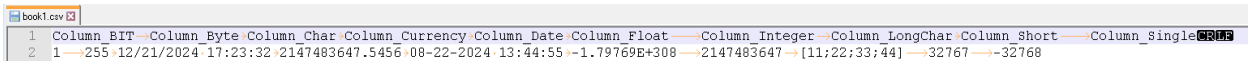
Col8=COLUMN_LONGCHAR LongChar

Col9=COLUMN_SHORT Short

Col10=COLUMN_SINGLE Single

Note: For German OS, if the .csv is comma separated, then the schema.ini file must contain Format=CSV

Example of book1.csv TAB separated file:



Example of values in the .csv file:

Column_BIT	Column_Byte	Column_Char	Column_Currency	Column_Date	Column_Float	Column_Integer	Column_LongChar	Column_Short	Column_Single
1	1	strArray0;strArray1;strArray2	1234567890123.45	08-22-2024 13:44:55	1.7976E+308	-2147483648	[-128;255;127;0]	-32768	-32768
0	255	12/20/2024 17:23:32	2147483647.5456	08-22-2024 13:44:56	-1.7976 9E+308	2147483647	strArray0; strArray1; strArray2; strArray3	32767	32767
1	0	11;22;33;44	-12345678901234.5	08-22-2024 13:44:57	1.7976 9E+308	0	String	1	1

Note: The columns with data-type Char and LongChar can also contain Arrays and DateTime values (the string value must match the configured DateTime format in dataFEED OPC Suite)

For details regarding functional limitations of the DatabaseRead, see the Limitations page.

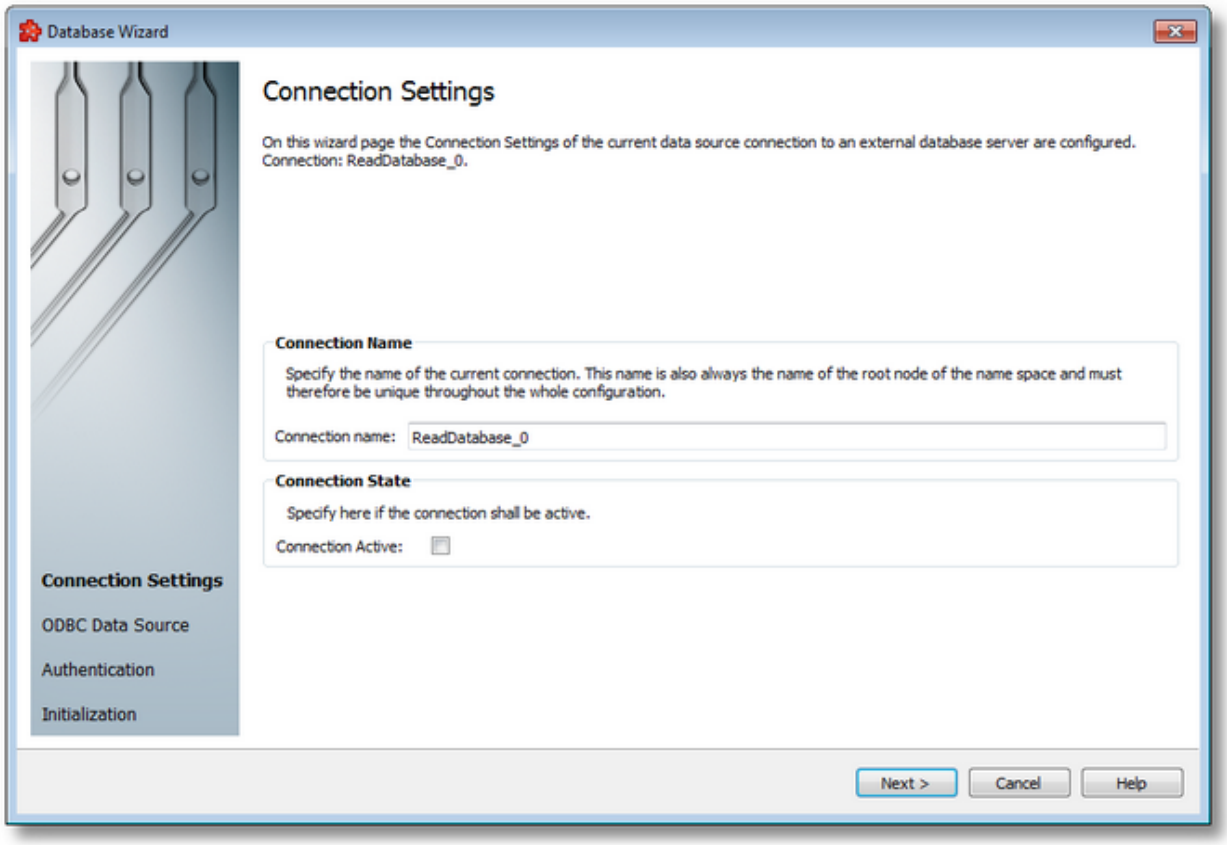
7.3.1.10.2Database Wizard

With the Database Wizard you can create and modify an ODBC connection to an external SQL database server:

- Create a new connection or modify an existing one @ Database Wizard - Connection Settings
- Select the ODBC data source for the database connection @ Database Wizard - ODBC Data Source
- Specify the user name and the password for the database connection @ Database Wizard - Authentication
- Specify an SQL initialization statement for the database server @ Database Wizard - Initialization

© Softing Industrial Automation GmbH

The role of the *Connection Settings* wizard page is to provide a simple way for creating or modifying database connections. It contains two sections: *Connection Name* and *Connection State*:



The first section allows the user to specify the name of the connection. When creating a new connection its name is automatically generated by the system in the format "ReadDatabase_X", where 'X' is the next available connection number. When modifying an existing connection, the user must provide a unique connection name to avoid name collisions with the exiting connection names.

Note: If a connection name is taken, the application will show an error message: "There is already a connection with this name. Please choose another connection name". A unique connection name must be provided in any case to continue.

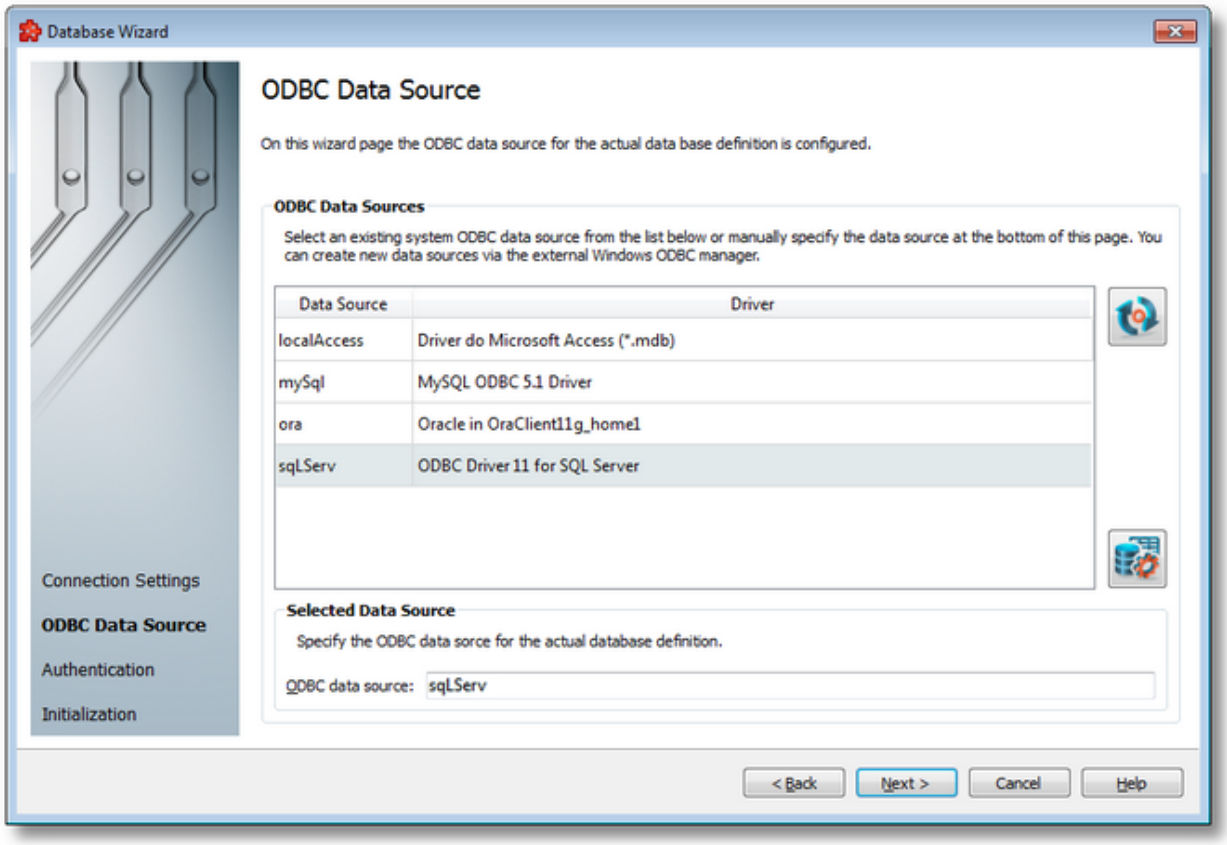
The second section contains a check box for activating or deactivating a connection.

The *Connection Active* flag specifies if the current connection will be enabled or not. In case this is set to false (check box unchecked) DataFeed will not connect to this database at runtime so none of its actions will be executed.

Note: In case a connection is not active, it will be grayed out on the main view along with all the actions associated with it.

Database Wizard - ODBC Data Source >>>

The role of the *ODBC Data Source* wizard page is to provide an intuitive and simple way for configuring the database connections. It contains two sections: *ODBC Data Sources* and *Selected Data Source*:



The first section allows selecting an already defined ODBC connection from the System DSN. If the desired connection does not exist in the list, a new ODBC connection can be defined by using the *Open the Windows ODBC data source manager* button  .

Note: Only the System DSN data sources are displayed in the *ODBC Data Sources* table.

Also note the following limitation: If both 32 bit and 64 bit ODBC drivers are installed in the operating system and there are connections defined in both of them, only the connections defined in the 32 bit driver will be shown in DataFEED. DataFEED has this limitation because it is a 32 bit application and such applications use the 32 bit ODBC driver.

Using the *Reload ODBC data sources* button  or pressing the keyboard shortcut CTRL+R updates the available data source list. After a new ODBC database definition is added to the Windows System DSN, an update needs to be done in order to see the latest addition(s). The ODBC Data Source table contains 2 columns:

© Softing Industrial Automation GmbH

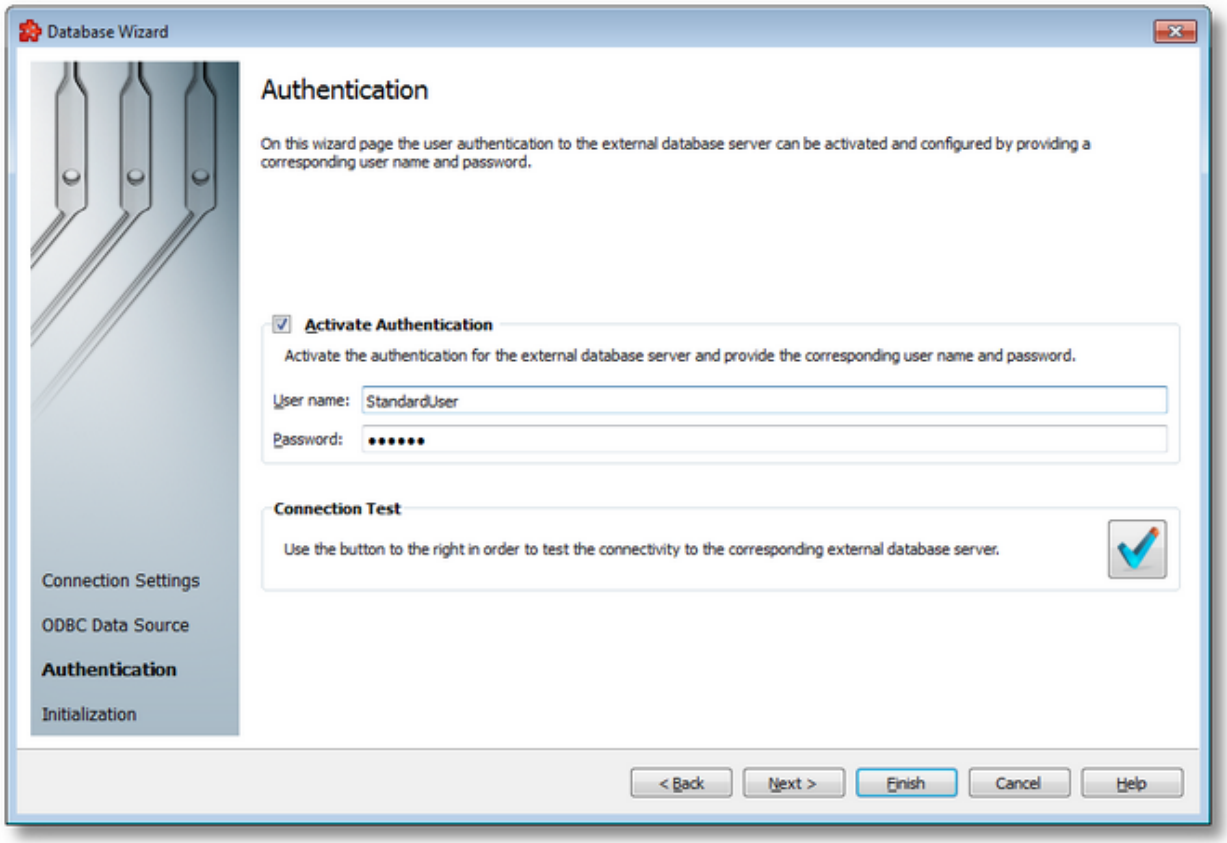
- *Data Source* - contains the ODBC Data Source name. This name will also be used as the database connection name in the *Databases* tree.
- *Driver* - contains the name of ODBC data source driver used for connecting to the database.

The second section contains an editable text box that is automatically populated with the selected DSN name. The name of the data source can also be manually written in case of an offline configuration when the DSN will be created later.

Note: It is mandatory to provide an *ODBC data source* name through which the database connection will be established, therefore it is not possible to continue the wizard without providing one.

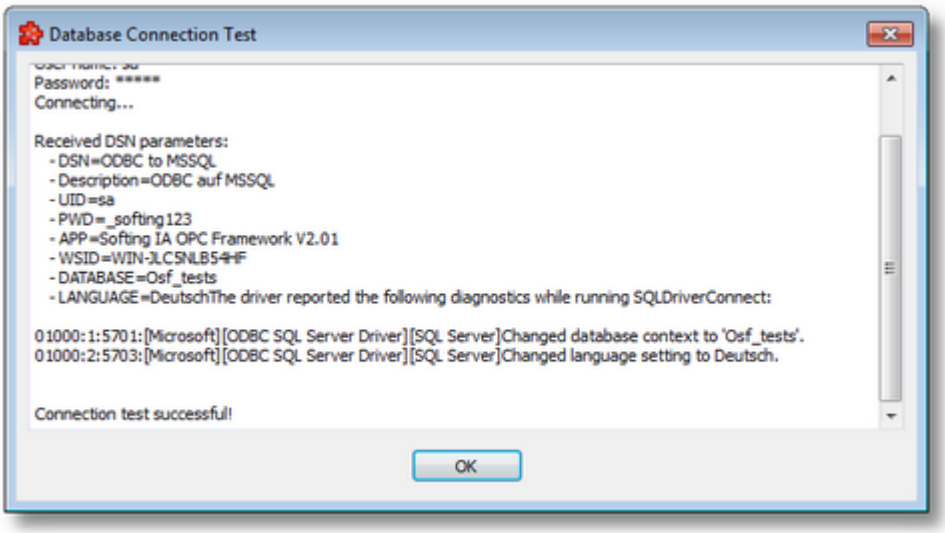
<<< Database Wizard - Connection Settings Database Wizard - Authentication >>>

The next step in configuring a database definition is to provide (if necessary) the authentication information for connecting to the ODBC. This is done in the Authentication page, which is organized as follows:

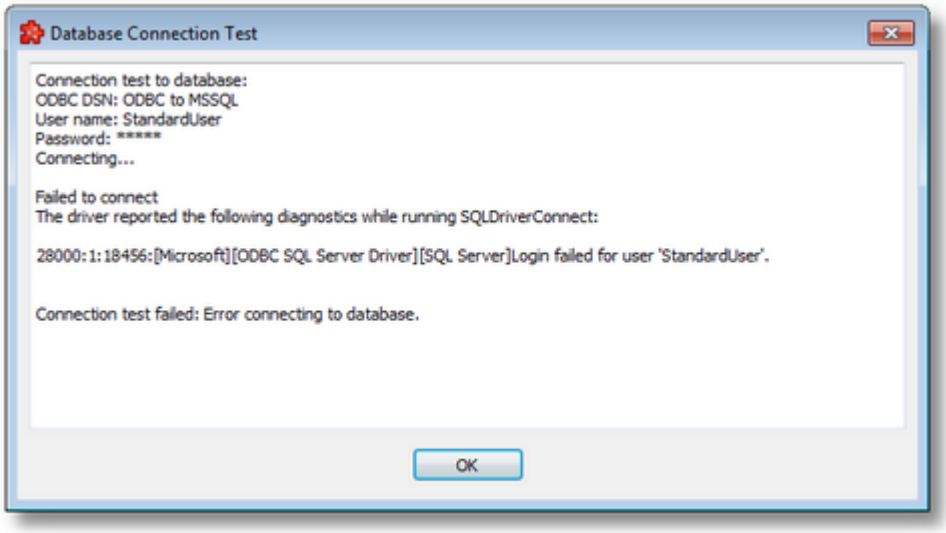


- A short description of this page's functionality is shown at the top of the wizard page.

- The *Activate Authentication* section has an associated checkbox for activating or deactivating this part of the database definition process. When active, the two fields for *User name* and *Password* are enabled and the corresponding data can be filled in. By default the Activate Authentication checkbox is disabled and the *User name* and *Password* fields are empty.
- The last section, *Connection Test*, allows testing the current configuration (including the non-mandatory user name and password). To perform the connection test, click the *Connection test* button  or press the keyboard shortcut CTRL+T. The *Database Connection Test* dialog appears where you can see the connection test results. A successful database connection test result is shown below:



Various pieces of information about the server is also displayed in this window. If the test connection failed, the database connection test result will look like this:



To proceed with the wizard and continue to define the database initialization statement press the *Next* button. To go back to the previous page (to change the DSN) use the *Back* button. Quit the *Database Wizard* by pressing the *Cancel* button.

© Softing Industrial Automation GmbH

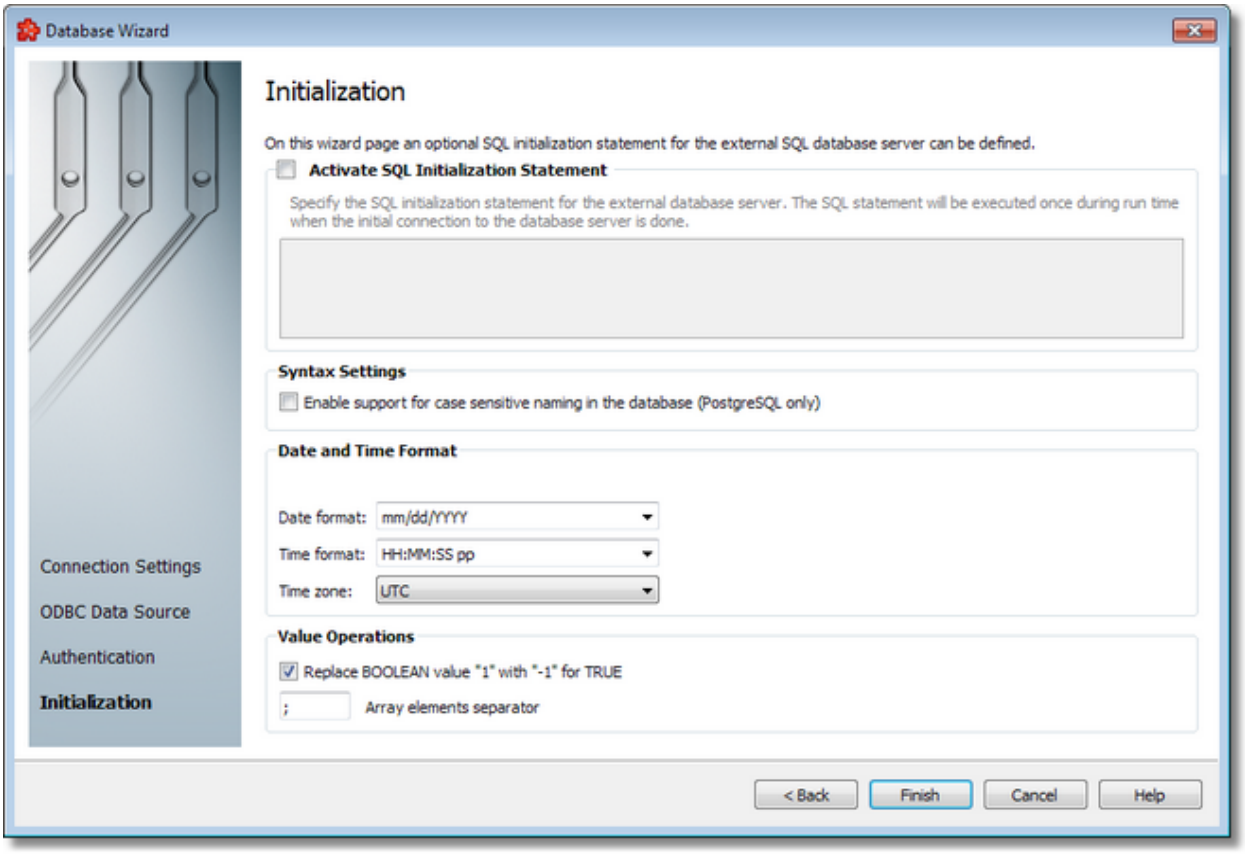
To stop at this point, press the *Finish* button. The next page, containing the database initialization statement and other details, can be skipped and default values will be set and saved in the configuration.

<<< Database Wizard - ODBC Data Source Database Wizard - Initialization >>>

The last page of the database definition wizard is the Initialization page which allows defining an initialization statement for the external SQL database, and other details like date and time format and value operations.

The first section has an associated checkbox, *Activate SQL Initialization Statement*, and a text field. By default this checkbox is not selected, therefore no SQL initialization command will be executed.

The wizard page is shown below:



When the *Activate SQL Initialization Statement* ckeckbox is selected, the free form SQL text editor is enabled where you can define a custom SQL statement.

Note: This SQL initialization statement will be executed each time the local dataFEED OPC Suite application connects to the database, that is when the local dataFEED OPC Suite application starts and also when the local dataFEED OPC Suite application reconnects to the database. It will not be executed when adding new Database Actions in the dataFEED OPC

Suite Configurator. The default database specified in the ODBC manager will be used in the Configurator and tables from that database will be automatically displayed in the Database Action Wizard - Database Table page.

For better **PostgreSQL support** (see also section below) the user needs to add a special command to the Windows ODBC Driver settings or to the initialization script on this wizard page, when using tables and stored procedures in actions:

```
SET SEARCH_PATH to public, custom_schema_1,..., custom_schema_n;
```

for the ODBC driver, and

```
SET SEARCH_PATH = public, custom_schema_1,..., custom_schema_n;
```

for the wizard page.

Setting the schema list only in the ODBC driver is advised.

Note: If the above schema list is given only in the wizard page, its settings will not be applied until the Runtime is started, so the Configurator will not take advantage of it. If the schema list is specified in the ODBC driver settings, then it will be applied as soon as the settings are saved in the driver. Thus, the Configurator can show database content (tables, stored procedures) only from the specified schemas.

Please read the PostgreSQL documentation regarding schemas.

This command specifies to PostgreSQL which schema it will search for the tables and stored procedures in, because dataFEED OPC Suite is not reading or saving this information into the configuration and PostgreSQL requires them by default.

Syntax Settings

This section contains a setting introduced for better support of PostgreSQL case sensitive naming convention, regarding how database tables, their columns and stored procedures are used.

Note: In PostgreSQL table and column names can be specified surrounded with quotation marks ("), which results in them being considered as case sensitive entities. This means that a table created with the name *"ExampleTable"* will not be the same table as another one created without quotation marks, i.e. *ExampleTable*. In fact, the latter one will be created as *exampletable*. The same rule applies for the column names of a table, and stored procedures and function. Also, a table can have column names specified with and without quotation marks when creating it.

When enabled, the Enable support for case sensitive naming in database (PostgreSQL only) option enables the application to be more compliant with this feature in the database, by using the following rules:

- when listing the tables from the database, dataFEED OPC Suite will list both case sensitive and case insensitive tables and their columns

© Softing Industrial Automation GmbH

- as noted above, there may be tables with the same name, but with different-case letters (they will be regarded as different tables)
- when saving the configuration, the dataFEED OPC Suite will save the table and column names surrounded by quotation marks for the Runtime to be able to correctly specify to PostgreSQL which table it is using in an action
- when loading the configuration, the application will remove the quotation marks internally so that the user will not see them on the user interface
- e.g.: SELECT "ID_OPCDAV3","INFO","BIT","BYTE","CHAR","Word","DWORD","Integer","DInteger","REAL","STRING","UI8","I8","R8" FROM "tb_0_OPCDAV3" - this is how a saved command will be executed when the option is enabled, and the table was created containing lower and upper-case letters in its name and its columns
- the option is applied to Select command actions and Stored procedure/Function actions; the user needs to add quotation marks when creating a general SQL command action manually for tables, columns and Stored procedures/Functions

Other rules:

- when the option is disabled (by default) the application may not display all the tables and their columns correctly
- the list of tables may be incomplete
- the list of columns may be incomplete or empty
- the same applies for stored procedures and functions as well, except for their parameter names
- it is not advised to use connections to PostgreSQL with the option disabled
- it is not advised to use other types of database connections with the option enabled (PostgreSQL only)

Time And Date Format

Different time and date formats can be used when reading time and date values from database. The default values are `HH:MM:SS pp` for the *Time format* and `mm/dd/YYYY` for the *Date format*. Other defined values for Time and Date format are:

- *Time format*: `HH:MM:SS pp, HH:MM:SS.FFF pp, HH:MM:SS.FFF, HH:MM:SS`
- *Date format*: `mm/dd/YYYY, dd-mm-YYYY, dd.mm.YYYY, dd/mm/YYYY, YYYY-mm-dd, mm-dd-YYYY`

Values for Time and Date format can also be defined manually but in this case special care must be taken to ensure that the definitions are recognized by the local dataFEED OPC Suite application.

In addition to the default values listed in the format selection combo boxes, the user can add his/her own custom format. Custom Date and Time format strings are accepted as valid, when the format strings contain in their composition the following accepted format substrings and format delimiters. These formats allow the inclusion of specific locale information for the regional settings.

- *Valid Date format strings*: `AAAAAA, aaa, BBBBBBBB, bbb, ccccccccccccccccccc, dd, jjj, mm, UU, ww, xxxxxxxxxx, yy, YYYY`
- *Valid Time format strings*: `II, HH, MM, SS, pp, XXXXXXXX, FFFFFF, FFFFF, FFFF, FFF ,FF, F`
- *Valid Date and Time format delimiters*: `<dot>, <colon>, <slash>, <comma>, <backslash>, <vertical bar>, <semicolon>, <space>, <tab>, <minus>`

Time Zone handles the source localization time: UTC (Universal Time Coordinated) or local time (local time localization). The default time zone is UTC.

Value Operations

The Database Connection may be configured to replace all TRUE BOOLEAN values with "-1" instead of "1" by marking the *Replace BOOLEAN value "1" with "-1" for TRUE* checkbox in the *Value Operations* group box.

Note: The *Replace BOOLEAN value "1" with "-1" for TRUE* checkbox is marked by default for new Database Definitions created with dataFEED OPC Suite v4.30 and later, and unmarked for existing Database Definitions from configurations created with earlier dataFEED OPC Suite versions.

The Database Connection may also be configured to use different array separators by changing **Array elements separator** token.

Note: By default the array separator is ";"

Press *Finish* to complete the wizard or quit the wizard by pressing the *Cancel* button.

<<< Database Wizard - Authentication

7.3.1.10.3Database Action Wizard

With the Database Action Wizard you can create a new Database Action definition or modify an existing one:

- Define the database action type @ Database Action Wizard - Action Type
- Define the SELECT SQL statement to be executed @ Database Action Wizard - Select Statement
- Define the WHERE condition @ Database Action Wizard - Where Condition
- Define the Stored Procedure SQL statement to be executed @ Database Action Wizard - Stored Procedure

© Softing Industrial Automation GmbH

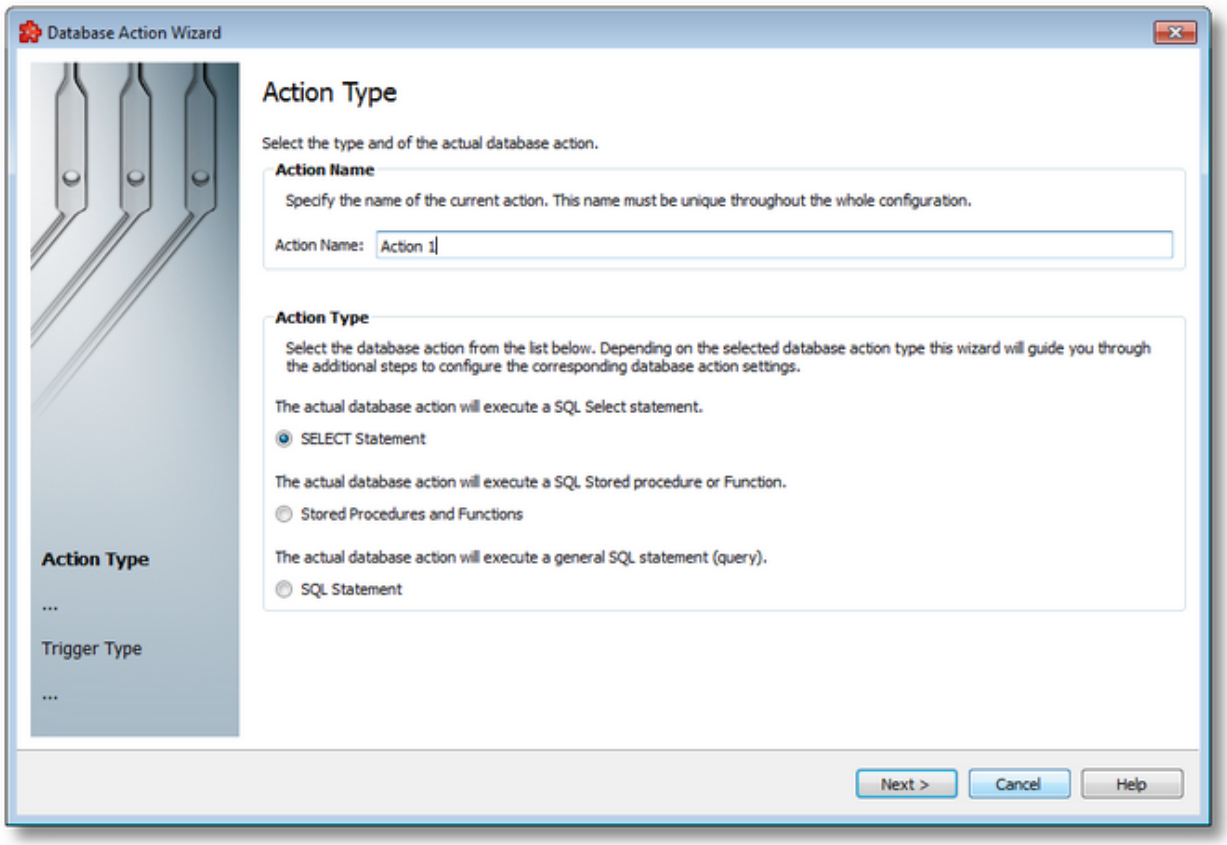
- Define the SQL statement to be executed @ Database Action Wizard - SQL Statement
- Choose the database action trigger type @ Database Action Wizard - Trigger Type

Note: A Database Action Wizard flow will not include all the pages listed above. A specific branch will be followed during the wizard, according to the selections performed in some of the pages.

The Action Type wizard page describes the different actions you can choose when defining a new Database Action.

Here the action name can be defined. Action name must be unique within current configuration actions and it is used to reflect different values in system items (like: total number of executed SQL statements, total number of failed SQL statements, number of open SQL statements etc).

Besides that it contains three radio buttons for the three types of database actions: Select Statement, SQL Stored Procedure or Function Statement and General SQL Statement, plus a short description of each.



Choose the type of action you want to perform on the database:

- *Select Statement* - only *SQL Select* statements can be created if this option is selected.

- *Stored Procedures and Functions* - allows executing the created SQL stored procedures with some parameters.
- *SQL Statement* - allows defining a specific SQL statement by adding specific SQL text in the a free form SQL text editor.

Note: It is possible that some of the defined database actions might appear as different ones after the configuration is reloaded into dataFEED OPC Suite Configurator (for example a General SQL action acting as an insert into statement will appear as an Insert Into statement).

To proceed with the wizard and continue to define the selected type of database action use the *Next* button, to go back to the previous wizard page use the *Back* button, or quit the wizard and discard the changes by pressing the *Cancel* button.

Depending on the selected *Action Type* the next wizard page will be:

- Database Action Wizard - Select Statement - if the selected action is *Select Statement*,
- Database Action Wizard - Stored Procedure - if the selected action is *Stored Procedures and Functions*,
- Database Action Wizard - SQL Statement - if the selected action is *SQL Statement*.

The description of the Database Action Wizard will continue with the wizard pages for defining a *Select Statement* by default.

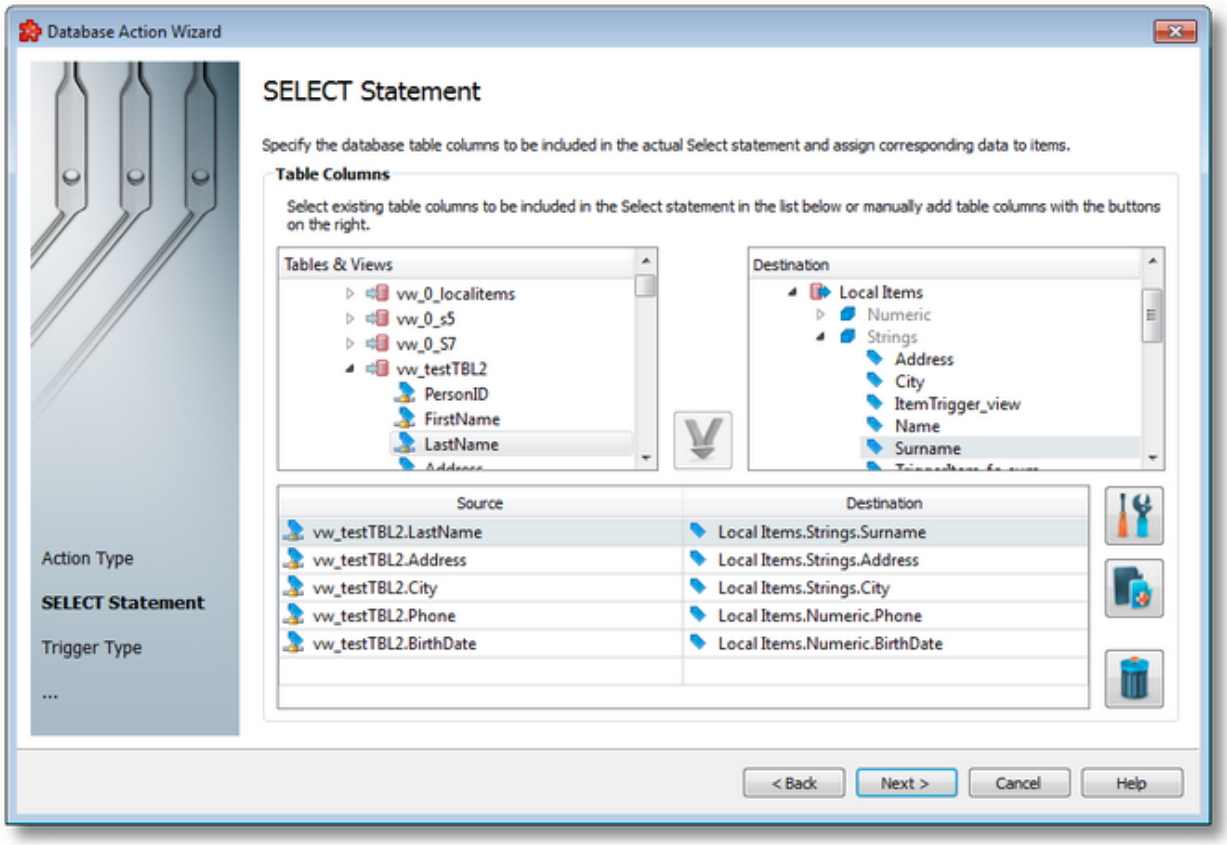
Note: When the connection to a database is not available, configuring and editing a Stored Procedure action is not allowed. The user will be presented a message about this situation.

This is because the application needs additional meta data about stored procedures in order to be able to configure a Stored Procedure action, like for example: not only the name of the stored procedure, but its type also (procedure or function); not only the names of the parameter, but their data types and directions as well.

Database Action Wizard - Select Statement >>>

The Select Statement wizard page allows selecting the desired table/view columns that compose the *Select* SQL statement action. The available tables and views names are retrieved online when expanding rows in the top left panel. Further more on expanding a table or view row, the columns names are provided.

© Softing Industrial Automation GmbH



The top right window provides the items available in current address space.

Finally the bottom table contains the linking between specific columns and address space items. The matching between columns and items can be done by dragging columns and items into bottom table.

At runtime the items will be written with the values read from the database on a routine specified in the trigger type and settings pages.

Please note that the selected items must be writable. One item can be used only once in a select statement action.

The user can delete already created column - item links by selecting the lines and press *Delete* button  . The source or destination items can be modified by dropping other items on top of them.

Only columns belonging to one table or view can be used in one select statement. In order to change the table or view, all the existing links need to be deleted then dropping columns from another table or view is allowed.

Column - items links handling

Besides drag and drop facilities to create new links or modify existing one, the links can be managed with buttons from right side of the table.

- *Add* button  , or pressing the keyboard shortcut CTRL+A is available at any moment and by using it a *Select Action Properties* dialog is shown.
- *Properties* button  , or pressing the keyboard shortcut CTRL+P is available only when a single row is selected in bottom table. When used, it opens the Select Action Properties dialog with current column and item already filled in.
- *Delete* button  , or pressing the DELETE / DEL is available is one or more rows are selected in bottom table. By clicking on it the selected links are deleted.

Select action properties dialog

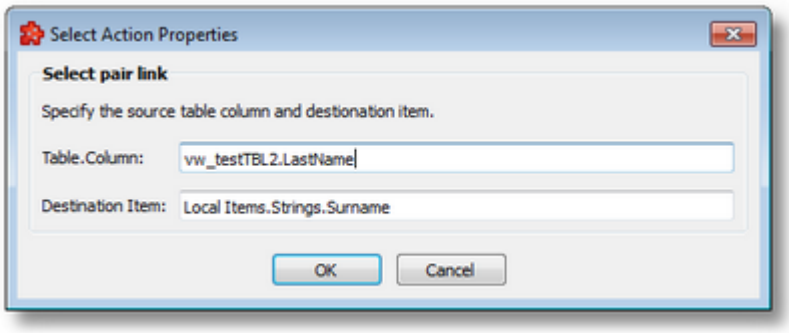


Table column text area should be used to specify the column used in SELECT link in the format: TableName.ColumnName.

On accept, this field needs to meet several criteria like:

- it cannot be blank, table name and column name cannot be blank
- accepted characters are: alphanumeric characters, ampersand, underscore and number sign
- if there are already other links made, the table name must match the used table and the column must not be used in other link

The Destination Item text must specify the fully qualified item name.

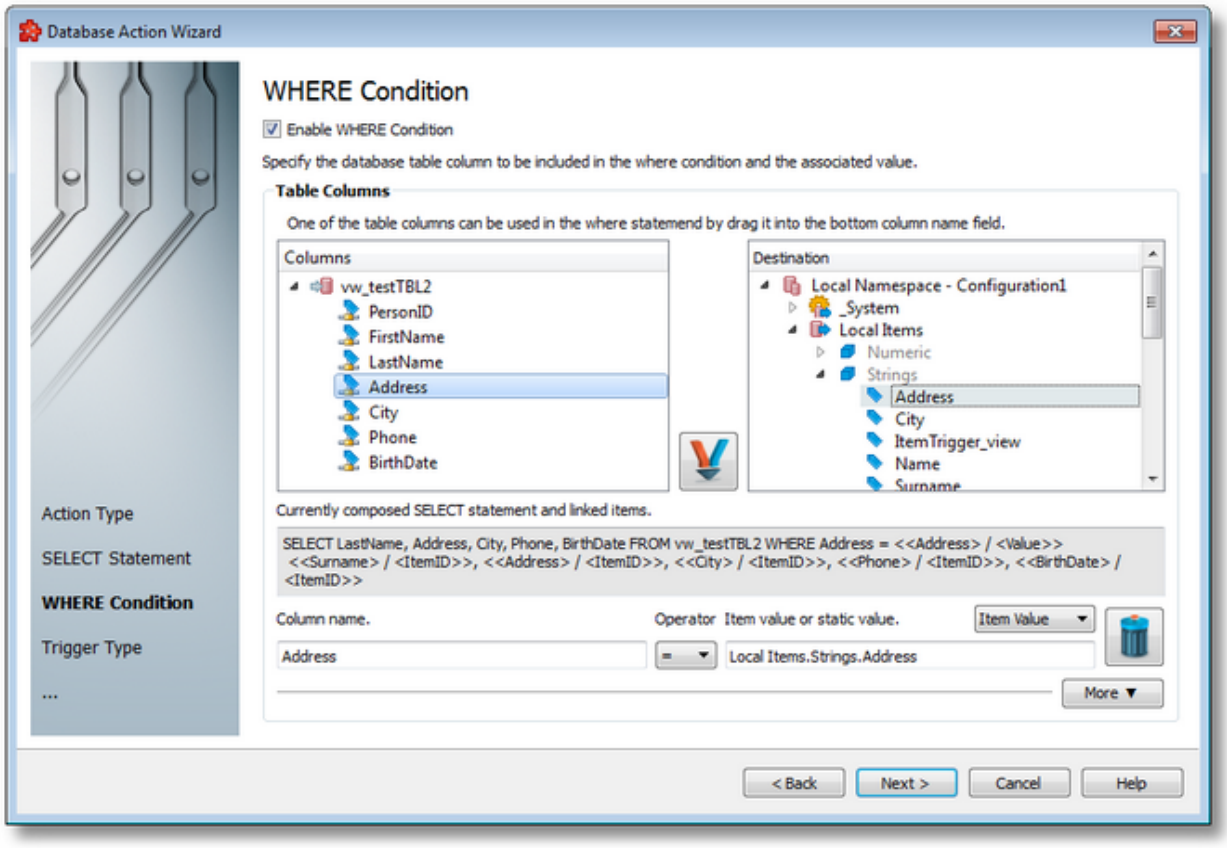
Some validation rules apply to this field also:

- item should exist in address space
- item should not be used in previously created links

<<< Database Action Wizard - Action Type Database Action Wizard - Where Condtion >>>

© Softing Industrial Automation GmbH

The Where Condition wizard page allows composing of a simple condition to be used in the current statement.



The page consists of several elements that can facilitate the creation of the condition.

Table columns tree view contains all the columns of the current table or view. They can be dragged into Column name filed.

Beside this a tree view containing the address space can be used to drag items into Item value field. Also, instead of items, static values can be defined for different data types.

The **Currently composed SELECT statement** panel is a read-only area where the user can review the already associated columns with items. The items are described in place holder manner.

The **Column name** field contains the used column. The user can replace this one by dragging a column name from the Columns tree view. Beside that the column name can be added manually.

Please note that in case manually edited columns do not exist in the table at runtime the action execution will fail.

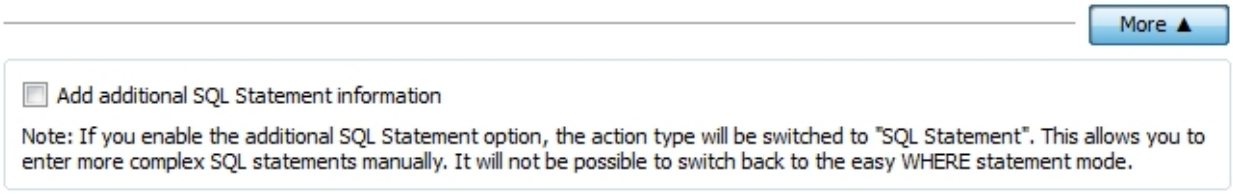
Operator is a drop-down box containing a list of operators that can be used in this simple condition builder. The available operators are: =, <>, <, >, <=, >=

Item value or static value field can contain either an item value (dragged and dropped from address space tree view or manually edited) or a value. The switch between these two cases can be done with the combo box placed right after this label. This combo is important in the validation stage: in case the item value is selected the input is used to validate the specified items.

When specifying a static value with content of type string the user needs to specify the value between ' (single quotes) characters, like this: *'example text',* for databases to be able to interpret the value correctly. Otherwise, the user can enter anything, but the operation might fail.

The user can reset the already created condition by using the *Delete* button  . The column name and item value will be cleared and the operator reset.

Finally the **More** button can be used to expand a hidden panel, which enables the user to convert the Select Statement to an SQL Statement.



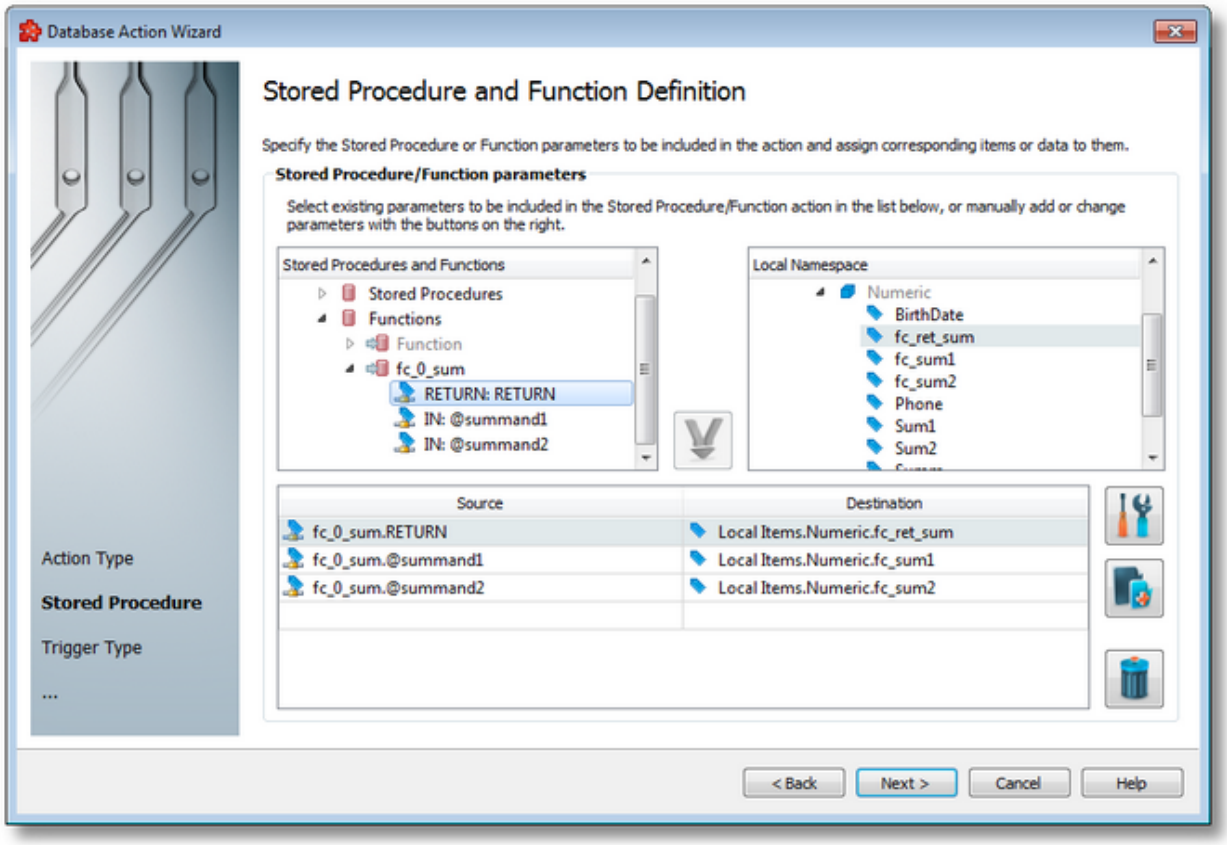
Please note for the using of date and time formats that they will be converted to the ISO 8601 format. The conversion affects address space items of data type DATETIME or STRING. STRING values must have the format YYYY-mm-dd-HH:MM:SS.FFF or YYYY-mm-dd-HH:MM:SS.

<<< Database Action Wizard - Select Statement Type

Database Action Wizard - Trigger >>>

The *Stored Procedure and Function Definition* wizard page allows the selection of the desired stored procedure or function parameters that compose the *Stored Procedure* SQL statement action. The available stored procedure and function names are retrieved online when the wizard page is loaded, along with all parameter names and their types (as in input, output, input/output or return parameters).

© Softing Industrial Automation GmbH



The top right window provides the items available in current address space.

Finally the bottom table contains the linking between specific parameters and address space items. The matching between parameters and items can be done by dragging parameters and items into bottom table, by selecting one of each from the corresponding source views and pressing the down-pointing button in the middle, or by using the *Add* button on the right of the mapping table. An entire stored procedure can be drag and dropped and all its parameters will be added to the mappings table. After that items can be dragged to complete the mappings.

The contents of the *Destination* column in the mappings table can be reordered by using drag and drop.

A static value can be assigned to a parameter, which can be specified on the dialog which pops up if the user presses the *Add* or *Properties* buttons.

Stored procedures without any parameters can be set up, too. They are simply called by the runtime using the assigned trigger. These stored procedures cannot be assigned any items or static values, or edited by the right-side buttons.

The runtime will execute the created stored procedure on an occasion or recurring interval specified in the trigger type and settings pages.

Please note that the access rights of the selected items must match the type of parameters they are mapped to (see above). This means that an input parameter can be assigned a readable or writeable item, an output and a return parameter can only be assigned a writeable item and an input/output parameter a readable or writeable item.

A parameter or an item can be used only once in a stored procedure action. All parameters, except the return type parameter (if available), which is optional, have to be assigned in order for the stored procedure call to be successful in the runtime.

The user can delete already created parameter-item links by selecting the desired lines and pressing the *Delete* button  . The source and destination items can be modified by dropping other items on top of them.

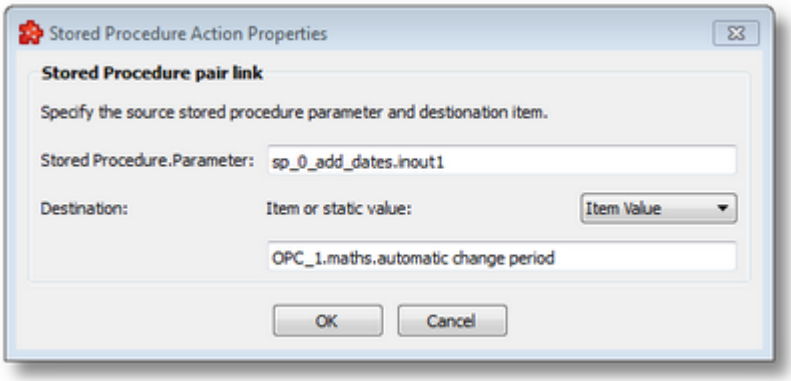
Only parameters belonging to one stored procedure or function can be used in one stored procedure action. In order to change the table or view, all the existing links need to be deleted, after which dropping parameters from another stored procedure or function is allowed. All other stored procedures and functions will be grayed out until a new one is added.

Handling parameter-item links

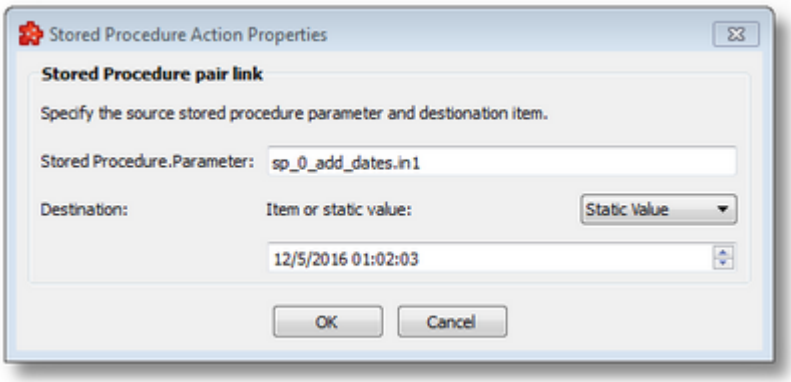
Besides drag and drop facilities to create new links or modify existing one, the links can be managed with buttons from the right side of the table.

- *Add* button  , or pressing the keyboard shortcut CTRL+A key, it is available at any moment and by using it a stored procedure action dialog is shown.
- *Properties* button  , or pressing the keyboard shortcut CTRL+P, it is available only when a single row is selected in the bottom table. When used it opens the *Stored Procedure Action Properties* dialog with the current parameter and item already filled in.
- *Delete* button  , or pressing the DELETE / DEL key, it is available if one or more rows are selected in the bottom table. By clicking it the selected links are deleted.

Stored Procedure action properties dialog (item values and static values)



© Softing Industrial Automation GmbH



Stored Procedure Parameter text area is used to specify the parameter used in the stored procedure link in the format: Stored Procedure.Parameter.

On pressing the *OK* button this field needs to meet several criteria like:

- it cannot be blank, stored procedure name and parameter name cannot be blank
- accepted character are: alphanumeric characters, ampersand, underscore and number sign
- the procedure name must match the already used stored procedure name and the parameter must not be used in other links

The *Destination field* can be an Item or static values (depending on the data type of the parameter). It can be changed by selecting the desired value from the drop down menu. Original values will be lost after closing the dialog by using the *OK* button, and they will be kept if *Cancel* is pressed.
Also, appropriate validation messages are shown before accepting a newly created or modified link.

Possible data types for the static value field are: *Date time*, *Integer*, *Decimal*, *Boolean* and simple *text*.

Some validation rules are applied to Item value field:

- an item name must specify the fully qualified item name
- item must exist in address space; it cannot be dynamically created
- item must not be used in previously created links

Validation rules apply to the Static value field:

- a static value must match the data type of the parameter (this is enforced by changing the field to the corresponding data type field when the parameter name has been validated to exist)
- no static value can be assigned to a parameter whose type is output or return type parameter (static values cannot be assigned values coming from the execution process of a stored procedure)
- the Item Value/Static Value drop down field is disabled in this case above

- if the static value cannot be interpreted according to the type of the parameter, the value will be treated as a text value by default

Notes for non-existing or changed stored procedures/functions or parameters:

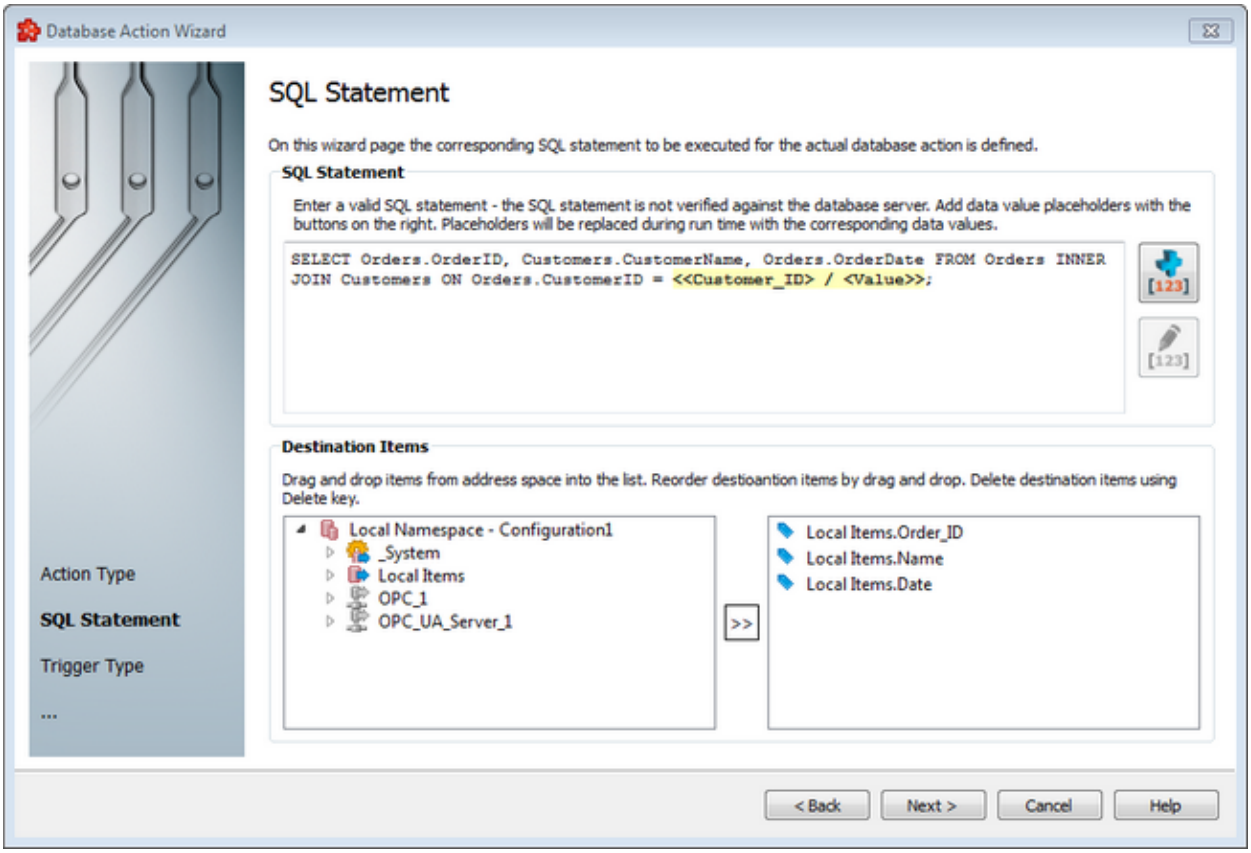
- If a stored procedure or function doesn't exist in the current database, its mappings will be invalidated with color red
- If a stored procedure or function's parameter name has changed or doesn't exist any more in the current database, it will be invalidated with color red
- Invalidated stored procedures or parameter mappings cannot be edited with the *Add* or *Modify* buttons; their meta data from their original database is not known
- The user can assign new stored procedures with their parameters or individual parameters in the mappings table to correct invalidated entries

<<< Database Action Wizard - Action Type Database Action Wizard - Trigger Type >>>

This wizard page allows defining the SQL statement text for the current database action with the help of the action editor. Certain operations are available using the buttons on the right, the context sensitive menu or the keyboard shortcuts:

- Insert a data value placeholder using the *Insert Data Value Placeholder* button  , pressing the keyboard shortcut CTRL+I or selecting the *Insert Data Placeholder* action from the context menu.
- Modify an existing data value placeholder using the *Modify Data Value Placeholder* button  , pressing the keyboard shortcut CTRL+M or selecting the *Edit Data Placeholder* action from the context menu.
- Cut, copy, paste or delete text with or without placeholders by selecting the *Cut*, *Copy*, *Paste* or *Delete* actions from the context menu or using the respective keyboard shortcuts CTRL+X, CTRL+C, CTRL+V and the DELETE / DEL key.

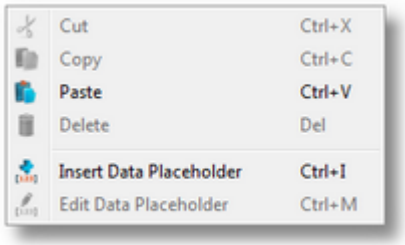
© Softing Industrial Automation GmbH



Free text blocks can be added in the action editor, including special characters. The database-specific action editor does not support new line and tab characters, however. The above example shows a more complex SELECT statement, which cannot be composed with the Select Statement Action type. An Inner Join operation is presented.:

```
SELECT Orders.OrderID, Customers.CustomerName, Orders.OrderDate FROM Orders INNER JOI
ON Orders.CustomerID = Customers.CustomerID;
```

An example of the context sensitive menu is presented below:



Note: The *Cut*, *Copy* and *Paste* actions use the Windows Clipboard for data transfer, thus allowing data to be transferred to/from other applications (e.g. the textual representation of the action editor content can be used in external applications and text from external applications can be used in the action editor).

Attention: The placeholders visible in the action editor are aliases used only for displaying the placeholders. The real placeholder form is visible in the subscription file. Example: *<<R8> / <Value>>* placeholder from the action editor is serialized in the subscription file as: *#%-ItemValue-%increment.R8_%Current-%#*. Do not manually modify the subscription file by inserting placeholders in the action editor template.

Bottom panel **Destination items** allows composing a list of items that will be updated with the values provided by currently composed statement execution.

The items can be dragged from current address space (left tree view) into the list (right list view). Items inside the list can be reordered by drag and drop operations. In order to delete items from the list select the item then use the Delete key.

On the runtime the statement is executed and last record is fetched. The destinations items are mapped to the columns of the retrieved record in order.

In case the number of items exceeds the number of columns, the remaining items are ignored; also if number of columns is greater than number of items, the exceeding columns are not used.

Note: If the statement contain errors or the last record cannot be fetched, destination items will not be updated. If the retrieved data from the database does not match the type of the associated items, the update will fail for that particular item.

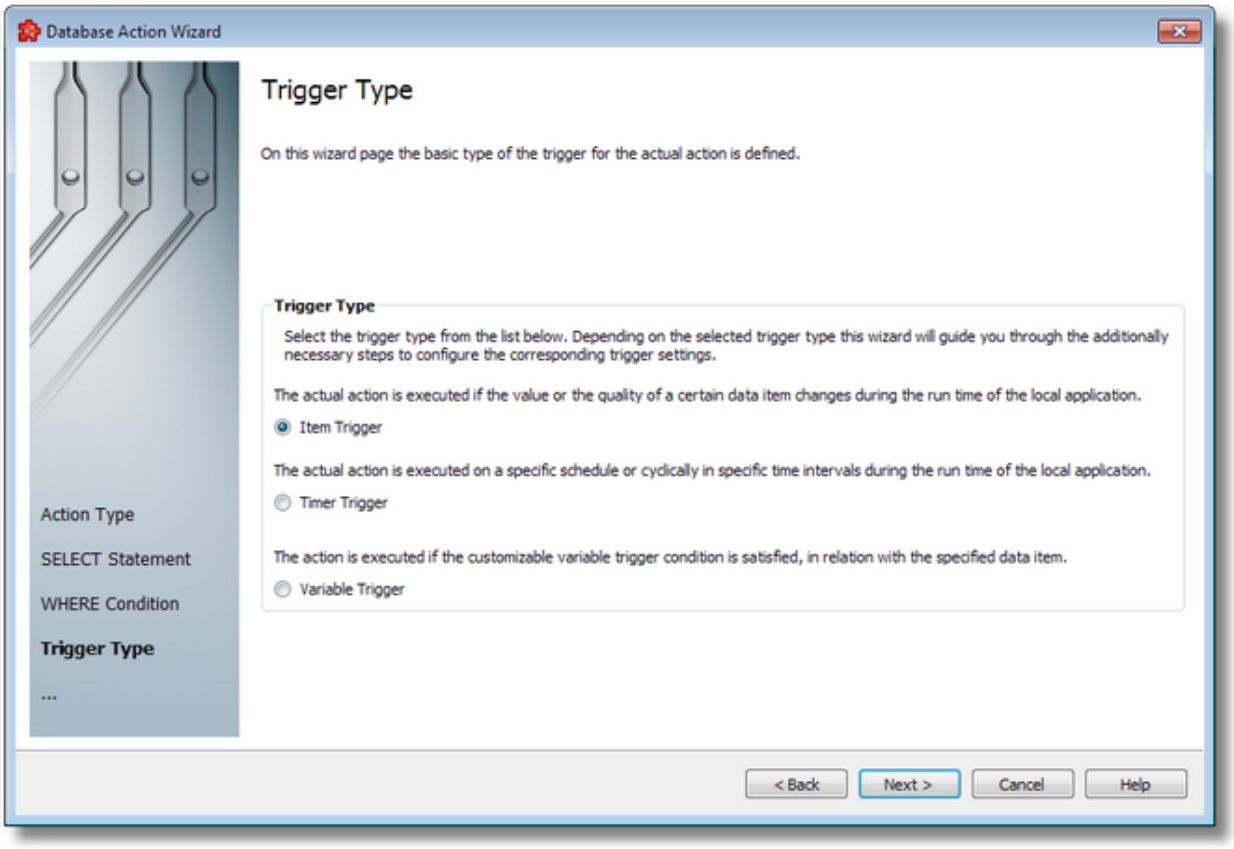
<<< Database Action Wizard - Action Type Database Action Wizard - Trigger Type >>>

The Trigger Type wizard page provides the possibility to select the trigger type for the actual action. The following trigger types are available:

- *Item Trigger* - This trigger is fired when a data change for the selected item occurs. A data change means that the value or the quality of the item changed.
- *Timer Trigger* - This trigger is fired at a specified time or interval.
- *Variable Trigger* - This trigger is fired when a data change for the items in the variable conditions occurs. A data change means that the value or the quality of the item changed.

Attention: The trigger only initiates the database action execution. The action will be subsequently validated, however, and might not be executed after all.

© Softing Industrial Automation GmbH



To proceed with the wizard and continue to define the selected type of trigger use the *Next* button, to go back to the previous wizard page use the *Back* button, or discard the wizard by pressing the *Cancel* button.

Depending on the selected *Trigger Type* the next wizard page will be:

- Database Action Wizard - Trigger Item - if the selected trigger type is *Item Trigger*,
- Database Action Wizard - Timer Selection - if the selected trigger type is *Timer Trigger*,
- Database Action Wizard - Variable Trigger - if the selected trigger type is *Variable Trigger*

<<< Database Action Wizard - Where Condtion

7.3.1.10.4Configuration Files

All the Database Read configuration settings are stored in two separate files:

- Tasks file - contains a general description of the Database definitions
- Subscriptions file - which contains all the descriptions related to the stored data:
    - what data will be stored
    - what is the frequency of storing the data
    - where and when will the data be stored

Note: The path and file name of these configuration files are fixed and not configurable.

The task and subscription files are stored in the *Tasks* folder under the instance path. For task configuration a text file called *Tasks.txt* and for subscription configuration a text file called *Subscriptions.txt* will be created in the *Tasks* configuration folder. Although these files can be manually edited, using the dataFEED OPC Suite Configurator is recommended.

Note: The recommendation is to use only the dataFEED OPC Suite Configurator to edit these files. This ensures that all the definitions will be valid.

If the configuration is manually edited try loading the configuration in the dataFEED OPC Suite Configurator as all the configuration files are automatically validated at load-time and, if there are any mistakes, an error message will be displayed. The dialog will contain an appropriate description and will point to the first invalid line of the local items configuration file.

Both configurations files are plain text files encoded in UTF-8 (thus providing full UNICODE compatibility) and they have line-oriented structure. The elements of a line are tab-separated. The format of one line is the following:

```
Line type<TAB>Line data
```

Task Configuration File

The following line types are defined in the Tasks file:

Line Type	Description
DBR	Defines the DSN parameters used in the ODBC connection: database server address, ODBC connection name and authentication credentials.
INIT	Used to specify an initialization SQL command which is executed only once after the local dataFEED OPC Suite application starts up (database selection etc).
Commented Lines (#)	Any line that starts with the "#" character is considered a comment line which is ignored during evaluation.

The DBR line

The DBR line describes the DSN parameters used in the ODBC connection. The DBR line has the following format:

```
DBR<TAB>DatabaseID<TAB>DSN<TAB>IP Address or Host name<TAB>ODBC connection name<TAB>U
```


© Softing Industrial Automation GmbH

Parameter	Description
DBR	The type of the line, predefined keyword.
DatabaseID	Unique database identifier within the task file (used for matching the INIT line and the statements in the corresponding subscription file).
DSN	The type of the database driver, predefined keyword if ODBC connection.
IP Address or Host name	Specifies the location of the ODBC connection. Generally this parameter will have the *localhost* value.
ODBC Connection name	The name of ODBC connection as it is saved in the ODBC Manager.
User name	Specifies what user name to use for connecting to the DSN connection. This parameter holds the value of the *User Name* field in the Database Wizard - Authentication page.
Password	Specifies what password to use for connecting to the DSN connection. This parameter holds the value of the *Password* field in the Database Wizard - Authentication page. This value is encrypted.
ArraySeparator	The separator for arrays. This can be any string (lenght > 0). The default value is semicolon.
ReplaceBool	Replace Boolean value "1" with "-1" for TRUE.
DateTimeFormat	The format used to convert string (read from database) to DateTime. Default value is mm/dd/YYYY HH:MM:SS pp
TimeFormatStore	Possible values are UTC or Local.
ActiveState	TRUE if connection is enabled, FALSE otherwise.

Examples:

```
DBR   ReadDatabase_1    DSN   localhost   MySqlDB     test_user1  75593C2324398DDB02
DBR   ReadDatabase_2    DSN   localhost   OracleDB    test_user2  B9FB9D1B5D09FE3100
```

The INIT line

The INIT line can be used to specify an initialization SQL command for any of the database connections. This statement will be executed once, immediately after the local dataFEED OPC Suite starts:

```
INIT<TAB>Database ID<TAB>SQL statement
```

Parameter	Description
INIT	The type of the line, predefined keyword.
Database ID	Unique database identifier within the task file (used for matching the DBR line and the statements in the corresponding subscription file).
SQL statement	A database initialization SQL command that is executed once, immediately after the startup of the local dataFEED OPC Suite application.

Example:

```
INIT<TAB>MyDatabase_4<TAB>ALTER SESSION SET NLS_LANG = JAPANESE_JAPAN.JA16EUC;
```

Note: The *Database ID* must be defined in the previous DBR definition line.

Task File - Example:

```
#     Database Read Add-on - configuration
#
DBR   ReadDatabase_1    DSN   localhost   MySqlDB     test_user1  75593C2324398DDB02
#
DBR   ReadDatabase_2    DSN   localhost   SqlServerDB test_user1  75593C2324398DDB02
#
DBR   ReadDatabase_3    DSN   localhost   OracleDB    test_user2  B9FB9D1B5D09FE3100
INIT  ReadDatabase_3    ALTER SESSION SET NLS_LANG = JAPANESE_JAPAN.JA16EUC;
#
DBR   ReadDatabase_4    DSN   localhost   OracleDB2   test_user2  B9FB9D1B5D09FE3100
#
```

Subscription Configuration File

The *Data Source - Database Read* generates multiple line types:

- common lines which are specific to all the consumer add-ons: condition lines, subscription lines and item lines;
- specific lines: modifier lines and copy lines.

Common Lines:

The following line types are common lines and will be used to configure different functionalities:

© Softing Industrial Automation GmbH

Line Type	Description
Subscription	Defines the update rate with which the current action will be executed.
Condition	Defines the execution condition for the current action.
Item	This line defines a trigger which is fired if the item data has changed. The trigger data includes all other items which have changed at the same time (were reported in the same data change callback from the OPC server). An item data change consists of a different value or quality of the respective item compared to the previous one.
Timer	This line defines a trigger which is fired at specific time and date or at specific intervals.
Commented Lines (#)	Any line that starts with the "#" character is considered a comment line which is ignored during evaluation.

The subscription file is divided into two sections:

- The global area:

  The global area extends from the first subscription file line to the first trigger line (e.g: ITEM, TIMER lines).

  This area contains definitions valid for the entire subscription file.

- The local area:

  The local area covers all the actions between two trigger lines in the subscription file.

  The lines in this area are only executed if the parent trigger fires.

Subscription Line

A subscription line has the following format:

```
SUBSCRIPTION<TAB>Subscription ID<TAB>UPDATERATE<TAB>Update rate value<TAB>ACTIVE<TAB>
```

The subscription line contains the following configurable parameters:

Parameter	Description
SUBSCRIPTION	The type of the line.
Subscription ID	It is used to identify the subscription and must be unique.

UPDATERATE	Predefined keyword.
Update rate value	Defines the update-rate with which the current action will be executed.
ACTIVE	Predefined keyword.
Active value	Active value which can be TRUE or FALSE.

The subscription lines always contain the global subscription line as well as all subscriptions which are used at least once in consumer actions. The global subscription is represented by the subscription line with the empty *Subscription ID*. The default update rate value for the global subscription is *1000* milliseconds.

The value of the update rate is saved in milliseconds. This value may not be set to 0 inside the dataFEED OPC Suite Configurator but it can be manually set by editing the subscription file with an external text editor. An update rate value of 0 means *as fast as possible* but this is not recommended from a resource-related point of view.

Setting a subscription to be active or inactive is an advanced feature which cannot be configured from within the dataFEED OPC Suite Configurator. By default the active value for each and every subscription is set to *TRUE* (meaning that all subscriptions are active since creation). If a subscription's active value is set to *FALSE* then the data changes of the items configured to be part of that subscription will be ignored.

Examples:

```
SUBSCRIPTION            UPDATERATE  1000  ACTIVE      TRUE
SUBSCRIPTION      Subscription_300  UPDATERATE  300   ACTIVE      TRUE
```

Condition Line

A condition line defines whether an action below an item trigger will be executed or not (when the item trigger is fired). This can be used in other configuration lines as well to condition the execution of an item action. One example of this is the conditional execution of a SQL statement in the Database Read add-on. A condition line has the following format:

```
CONDITION<TAB>Condition ID<TAB>Condition Statement
```

The condition line contains the following configurable parameters:

Parameter	**Description**
CONDITION	The type of the line, predefined keyword.
Condition ID	Unique condition identifier.
Condition statement	Single condition statement.

© Softing Industrial Automation GmbH

Conditions can be used to react to communication errors. Various actions can be executed depending on the quality of items.

The following condition statements are supported:

Condition	Description
AllGoodQuality	The trigger actions are executed only if all items in the transaction have the quality *Good*.
Always	The trigger actions are executed always, ignoring the quality of the items in the incoming transaction.
ItemQualityChanged	The trigger actions are executed only if the quality of at least one item in the transaction has changed.
OneBadQuality	The trigger actions are executed if exactly one item in the transaction has the quality *Bad*.

The condition lines always contain the global condition line as well as all conditions which are used at least once in consumer actions. The global condition is represented by the condition line with the empty *Condition ID*. The default condition statement value for the global condition is *Always*. As a general rule all the condition lines need to be declared in the global section of the subscription file. If no condition is defined the Always condition is applied as default.

Examples:

```
CONDITION          Always
CONDITION    AllGoodQuality     AllGoodQuality
```

Item Line

An item line is used to add an item to a subscription. The associated actions are executed each time the value or the quality of the specified data items changes during the run time of the local dataFEED OPC Suite application.

An item line has the following format:

```
ITEM<TAB>Subscription Item ID<TAB>Server Item ID<TAB>Subscription ID
```

The item line contains the following configurable parameters:

Parameter	Description
ITEM	The type of the line, predefined keyword.
Subscription Item ID	Unique subscription item identifier (implicitly created by concatenating the Server Item ID and the Subscription ID with the underscore character in between them).

Server Item ID	Item identifier within the local dataFEED OPC Suite application address namespace. This could be a server connection item ID or a local item ID.
Subscription ID	Subscription identifier (defined above, see Subscription Line).

This trigger is fired if the item data or quality has changed (also referred to as data change). The trigger data includes all other items which were included in the data change (items which had their data change reported in the same transaction).

Note: A single line consisting of the *Server Item ID* and *Subscription ID* is added to the subscription file for an item used several times with the same update rate in any consumer add-ons. This optimizes the local dataFEED OPC Suite application throughput.

Example:

```
ITEM  increment.I1_Subscription_300 increment.I1      Subscription_300
```

Note: The subscription item ID can be empty. In this case, however, only the Server Item ID can be used to identify the item.

Timer Line

This trigger is fired at a specified time and date or at specific intervals. The associated actions are executed on a specific schedule or periodically during the run time of the local dataFEED OPC Suite application.

The format of the *Timer Line* is:

```
TIMER<TAB>Timer ID<TAB>INTERVAL<TAB>Rate in milliseconds
```

or

```
TIMER<TAB>Timer ID<TAB>SCHEDULE<TAB>Schedule range<TAB>First execution<TAB>Rate<TAB>L
```

The interval timer contains the following configurable parameters:

Parameter	Description
TIMER	The type of the line, predefined keyword.
Timer ID	Unique timer identifier.
INTERVAL	Denotes an interval timer trigger, predefined keyword.

© Softing Industrial Automation GmbH

Interval value	Defines the interval in milliseconds at which the timer will trigger the associated action(s).

The timer schedule contains the following configurable parameters:

Parameter	**Description**
TIMER	The type of the line, predefined keyword
Timer ID	Unique timer identifier.
SCHEDULE	Denotes a scheduled timer trigger, predefined keyword.
Schedule range	Defines the range of the schedule (see below the table).
First execution	Defines the time the timer should begin firing events (see below the table).
Rate	Defines the rate at which the timer should trigger the associated action(s).
Last execution	Defines the time the timer should stop firing events (see below the table).
List Of Days	Should be used only with the Week and Month schedule. It must contain the days on which the execution is desired. If the list of days is not present, the obtained functionality is the same as by using the day marker in every day of the week/month.

The timer's *Schedule range* may have one of the following values:

- MINUTE
- HOUR
- DAY
- WEEK - List of days: 1d = Monday, 2d = Tuesday, …, 7d = Sunday.
- MONTH - List of days: 1d = First day of month, 2d = Second day of month, …, 30d = Thirtieth day of month, 31d = Thirty-first day of the month.

The *First execution* defines the time the timer should begin firing events.

Example:

```
1d13h10m15s
```

Note: If parts of the execution time are 0, they do not have to be in the string, e.g. 1d0h10m0s = 1d10m

The timer *Rate* defines the event firing rate. An empty timer *Rate* means a single execution.

Last Execution: The last execution time of the schedule. Could be empty.

List of Days: Shall be used only with the Week and Month schedule. It should contain the days on which execution is desired. If the list is not specified the same functionality can be achieved by using the day marker in the first and last execution time.

Note: If the List of Days and the day marker in first and last execution time are used in the same timer trigger the two setting types should match the marked days!

Examples:

```
# Following timer fires events every 250 milliseconds (4 times a second):
TIMER       INTERVAL    250
# Following timer fires events every 20 seconds of a minute starting at second 10 (eg
TIMER       SCHEDULE    MINUTE      10s   20s
# Following timer fires events every 4 hours of a day starting at 6:30, ending at 20:
# (e.g. 6:30, 10:30, 14:30, 18:30).
TIMER       SCHEDULE    DAY   6h30m 4h    20h
# Following timer fires events every 4 hours of a day starting at 6:30, ending at 20:
# Friday (e.g. 6:30, 10:30, 14:30, 18:30 of each day).
TIMER       SCHEDULE    WEEK  6h30m 4h    20h   1d2d3d4d5d
# Following timer fires events every 4 hours starting at 6:30, ending at 22:45 (e.g.
# 18:30, 22:30), on the 1th, 2th, 10th, 20th of every month.
TIMER       SCHEDULE    MONTH 6h30m 4h    22h45m      1d2d10d20d
# Following timer fires events every 1 hours, 3 minutes and 3 seconds starting at 00:
# 23:59:59 (e.g. 00:00:01, 01:03:04, 02:06:07 etc), on the 3rd, 6th, 9th, 12th, 15,th
TIMER Timer_02    SCHEDULE    MONTH 1s    1h3m3s      23h59m59s   3d6d9d12d15d18d21d
```

Specific Lines:

The following line types are Database specific lines and will be used to configure different functionalities:

Line Type	Description
DatabaseRead (SQL2) line	Defines a SQL statement.

DatabaseRead (SQL2) Line

The format of the SQL2 line is the following:

```
SQL2<TAB>Condition ID<TAB>Database ID<TAB>Action Name<TAB>Action Type<TAB>SQL stateme
```


© Softing Industrial Automation GmbH

The SQL line contains the following configurable parameters:

Parameter	Description
SQL2	The type of the line, predefined keyword.
Condition ID	Condition identifier (defined above, see Condition Line).
Database ID	Database definition identifier (defined above, see DB2 Line).
Action Name	Unique name for an action (among all the database read actions)
Action Type	The type of the action. Possible values: 0 for Select Statement; 1 Stored Procedure/Function; 2 SQL Statement
SQL statement	A SQL statement (can be a *Select* statement, a *Stored Procedure* call or any *General SQL* statement).
Item name list	list with items placeholders of type ID and current data value. As in the following format: <START_DELIMITER><ITEM_ID Token>%ItemID_SubscriptionID%<Current Token><END DELIMITER> <START_DELIMITER> = #%- <END DELIMITER> = -%# <ITEM_ID Token> = ItemItemID <Current Token> = Current
Trigger ID	Associated trigger identifier (one of the above defined: Item Trigger, Timer Trigger)

The SQL2 statement parameter defines the SQL instruction which will be executed on the associated database when the trigger fires the event. It may contain plain text and also data placeholders.

An example of SQL2 configuration lines and the appropriate item lines is presented below:

```
ITEM  Local Items.I4_2_ Local Items.I4_2
#
ITEM  Local Items.I4_3_ Local Items.I4_3
#
ITEM  Local Items.ItemTrigger_      Local Items.ItemTrigger
SQL2        ReadDatabase_1    Action 1    0     SELECT ID FROM TABLE_1 WHERE NAME='ab
SQL2        ReadDatabase_1    Action 2    0     SELECT ID FROM TABLE_1 WHERE NAME=#%-
#
ITEM  Local Items.STR_1_      Local Items.STR_1
#
```

```
TIMER Timer_5sec  INTERVAL    1000
SQL2        ReadDatabase_2    Action 3    0     SELECT ID_bar FROM tb_0_bar ##WHERE##
```

The first Action (SQL2) will read the ID column from the TABLE_1 table (only if the NAME column is equal to 'abc' string) and put the value into the `Local Items.I4_2_` ITEM. This will be executed only if an Item is triggered (`Local Items.ItemTrigger_`).

The second Action (SQL2) will read the ID column from the TABLE_1 table. The where clause is using a Placeholder: the Current value of the `Local Items.STR_1_ Item` will be replaced at each execution. This will be executed only if an Item is triggered `(Local Items.ItemTrigger_)`.

The third Action (SQL2) will read the ID_bar column from tb_0_bar table. It does not use a where clause. This will be executed only if the timer trigger `(Timer_5sec)` expires.

If the statements are associated with triggers of type *Item*, the respective statements are evaluated on each item change, but are only executed if all of the following conditions are true:

- The subscription item ID of the statement has to be defined in the configuration.
- A change occurs for the respective item.
- The action condition is true.
- The placeholders in the statement can be replaced by their respective values (the changed data source may invalidate the action).

If the statements are associated with triggers of type *Timer*, the respective statements are evaluated and executed only if all of the following conditions are true:

- The subscription item ID of the statement has to be defined in the configuration.
- The action condition is true.
- The placeholders in the statement can be replaced by their respective values (the changed data source may invalidate the action).

Note: The order of the triggers and actions in the subscription configuration files is not random. A trigger should always be followed by its associated actions. Interleaving items or actions manually can lead to invalid configurations.

7.3.1.10.5Limitations

The functional limitations for Database Read are described below:

1. ODBC Driver limitations: If both 32 bit and 64 bit ODBC drivers are installed in the operating system and there are connections defined in both of them, only the connections defined in the 32 bit driver will be shown in DataFEED. DataFEED has this limitation because it is a 32 bit application and such applications use the 32 bit ODBC driver.

© Softing Industrial Automation GmbH

2. With Oracle Databases the DateTime type items from PLC engines can be used only with TIMESTAMP types.

3. Due to Qt-specific limitations, database data types cannot be detected property. Qt enumerates certain sub-types into the same categories, thus these sub-types will not be treated as those types. This means that for example the database types SQL_DECIMAL, SQL_NUMERIC, SQL_REAL, SQL_FLOAT, SQL_DOUBLE all fall into the Double type category and the application will treat all of them as Double.

4. In MSSQL you can call only scalar-valued functions types, no table-valued function calls are possible.

5. On MySQL the SystemItems: ItemsWrittenFailedLastAction, ItemsWrittenFailedTotal are not updated if overflow happens on item retrieval.

6. If the MSSQL Action contains WAITFOR statement the SQL server documentation remarks that "Cursors cannot be opened on WAITFOR statements." so the System Items can't be updated correctly.

The functional limitations for .txt / .csv data Read and possible conversion failures are described below:

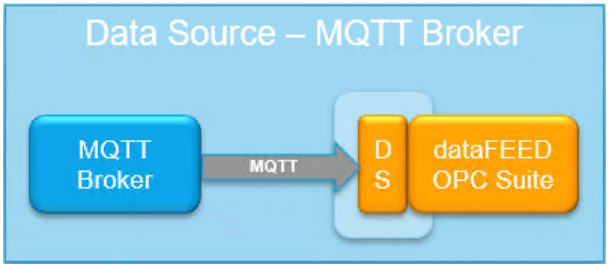
1. The .txt / .csv columns values must have the correct data-type matching the schema.ini file and the dataFEED destination variable types.

2. For writing into DATETIME variable from a column value as a DateTime but with the data-type as CHAR, the column-value must match the "Date format" defined in Database Read Definition Wizard. If the column data-type is DATE, then the column-value must match the ODBC driver required format; example for Microsoft Text Driver, the Date format is: mm-dd-yyyy hh:mm:ss

3. For writing into ARRAY variables, the columns data-types must be CHAR with the correct size set. The index separator must match the "Array elements separator" defined in Database Read Definition Wizard; example1: column value: 1;2;3;4 can be written in ARRAY variable, output value being: [0,3][1,2,3,4]; example2: column value: [1;2;3;4] can only be written in ARRAY variable of type INT

7.3.1.11 MQTT Broker

This section of the Softing dataFEED OPC Suite online help explains how an MQTT Broker is configured as data source by using the functionality provided by Softing dataFEED OPC Suite.

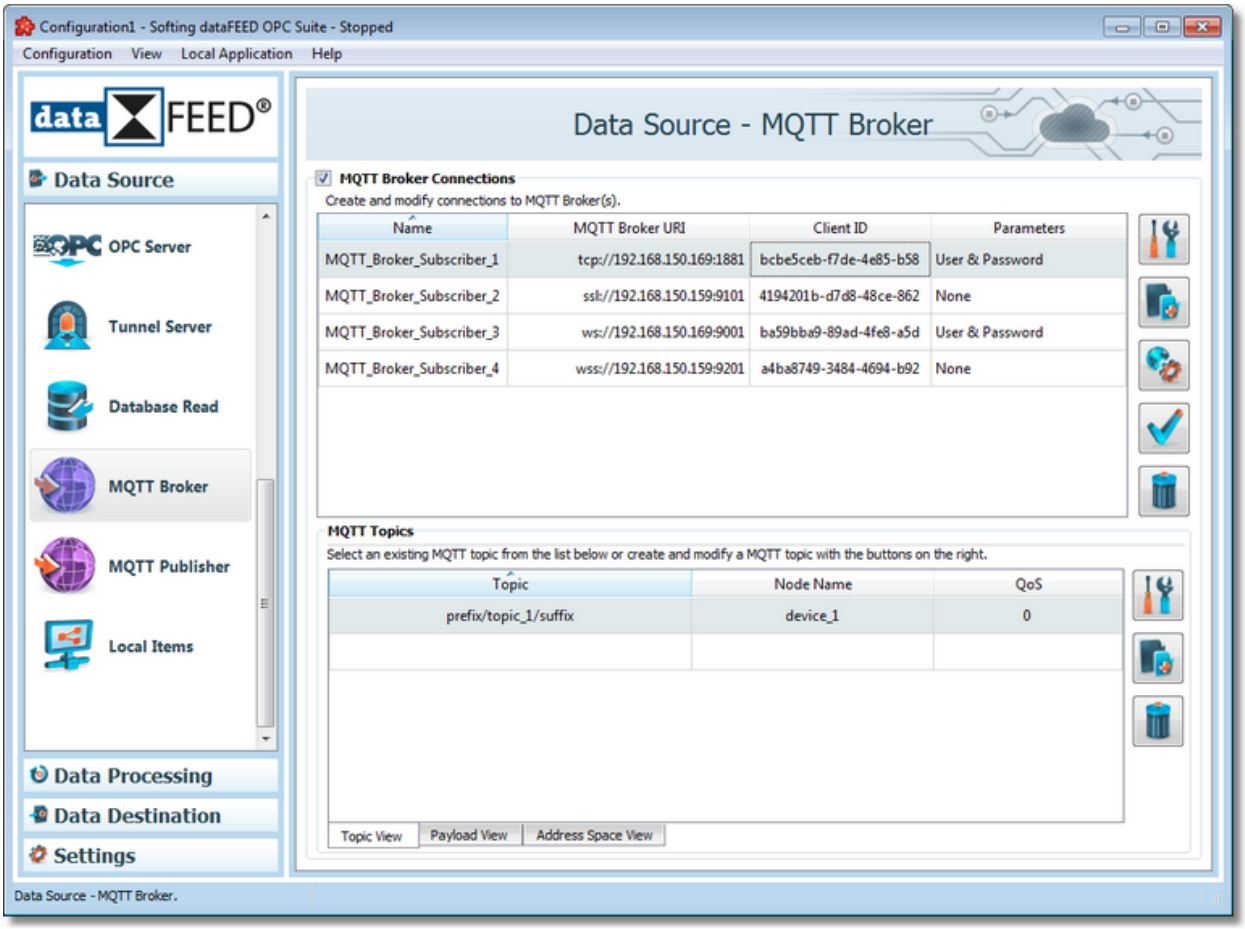


The add-on implemented MQTT Subscriber functionality makes it possible to receive data from a *MQTT Broker*, by subscribing to MQTT Topics from a *MQTT Broker* data source. The received data can be then processed, saved, analyzed or transferred by using the *Data Processing* and/or *Data Destination* functionalities.

A detailed description on how to configure MQTT Subscriber connections to a *MQTT Broker* can be found in the Configuration section.

7.3.1.11.1Configuration

In order to configure MQTT Subscriber connections to MQTT Brokers select *Data Source* in the left window part and then select the MQTT Broker icon. The MQTT Broker connections overview is opened.



Add an MQTT connection

You can establish a connection to an *MQTT Publisher* by using the button *Add*  or the menu command *Add* in the context menu. The MQTT Broker Connection wizard is opened.

Modify an existing MQTT connection

© Softing Industrial Automation GmbH

The configuration of an existing connection to an *MQTT Publisher* can be modified. To do so, select the line in the connection overview that contains the connection to be modified. Then you can use the button *Properties*  or the menu command *Properties* from the context menu to open the MQTT Broker Connection wizard.

Modify the topic configuration of an existing MQTT connection

The address space of an *MQTT* connection is composed of items. You can specify these items within the MQTT Broker Connection wizard.

An existing connection can be modified directly in the Topic Configuration page of the wizard by selecting the connection in the connection overview.

Connection test

Click test connection button  to check the availability of specified MQTT broker or the correctness of the authentication and security settings.

Remove a connection

Delete the selected data source from the list by using the *Delete*  button, pressing the *DEL / DELETE* key on the keyboard or selecting *Delete* from the context menu.

Note: If no connection is selected, the Add a new data source button and Global settings are the only options enabled.

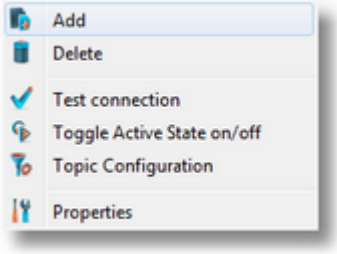
The user can select more than one connection with the help of the *CTRL* or *SHIFT* keys.

A dialog will then prompt the user to confirm the deletion.

Attention: A connection cannot be deleted if at least one data item defined in its namespace is used.

These actions are also provided in the context menu of the MQTT connection list. Which actions are available depends on the current selection in the list.

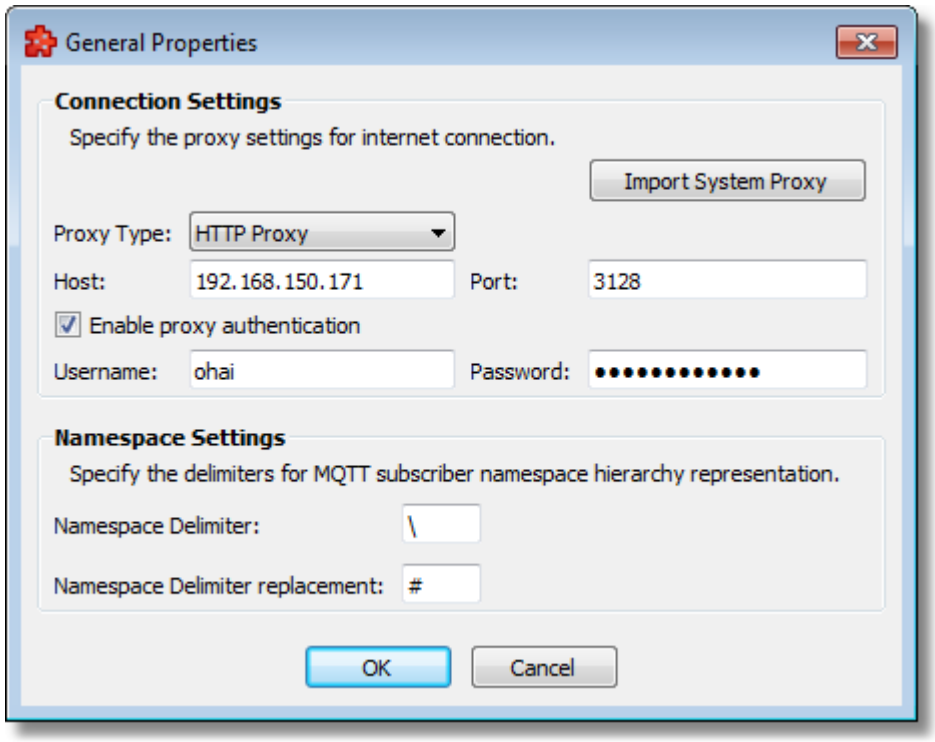
Following an example of a context menu:



Toggle active state on/off

Toggle active state on/off  This action will enable / disable a device's connection. A disabled connection will be shown as grayed out in the device connection list and it will be ignored by runtime.

General Properties



The *Connection Settings* section allows for the customization of proxy settings. The user can either import system proxy settings automatically (Import System Proxy button), or specify manually the host, port and authentication settings.

These settings will be applied to all connections in the plugin, and the following parts of the functionality will be affected: test connection, getting the server certificate and action execution. This means that if the network, on which the dataFEED is configured, is behind a proxy server and dataFEED is not configured with a proxy, like in the image above, the mentioned pieces of functionality will fail. In order for them to work correctly, the user needs to specify the type of proxy, the host name of the server, a port number and user credentials, if necessary.

© Softing Industrial Automation GmbH

The *Namespace Settings* section allows to specify the local delimiters for the hierarchy representation of the MQTT Broker Subscriber namespace. The namespace delimiter is used in the Payload - Item Mapping dialog window, within the expressions of the payload key and item/tag name. If the namespace delimiter is the same as the dataFEED OPC Suite application namespace delimiter, the namespace delimiter is replaced with the namespace delimiter replacement.

Note: The namespace delimiter and the namespace delimiter replacement fields are editable and can contain only one character. The default value of the namespace delimiter is '\'. The default value of the namespace delimiter replacement is '#'.

Note: The namespace delimiter and the namespace delimiter value must be different than the characters allowed for the expression of the payload key and item/tag name: (0-9), (a-z), (A-Z), '_' and '-'. The namespace delimiter replacement must be different than the application supported namespace delimiters '.' and '/', and the application supported property delimiter '|'. The namespace delimiter replacement value must be different than the namespace delimiter value.

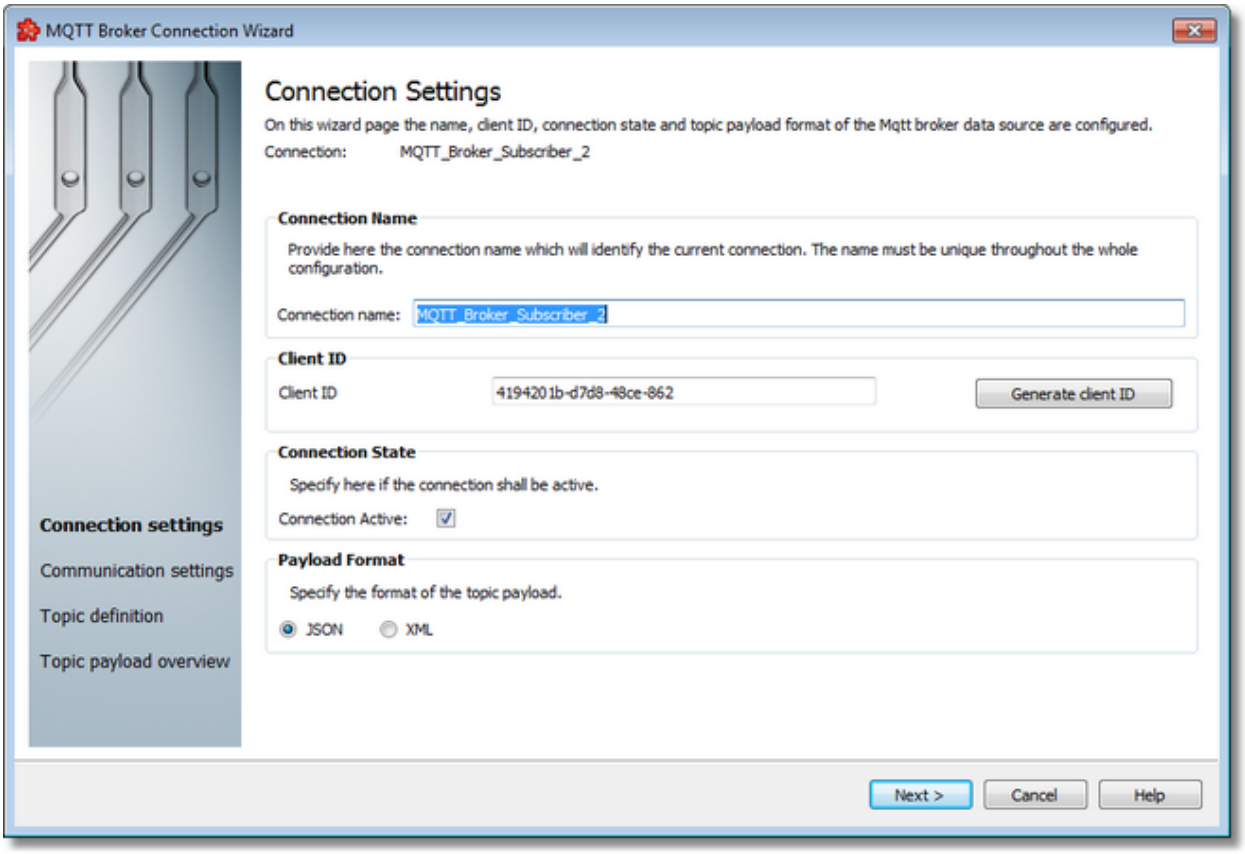
Note: If the namespace delimiter is already used for the hierarchy representation of the MQTT Broker Subscriber namespace, its value cannot be changed in the General Properties dialog, and the user is warned about this. If the namespace delimiter was already replaced in the hierarchy representation of the MQTT Broker Subscriber namespace, the namespace delimiter replacement value cannot be changed in the General Properties dialog, and the user is warned about this.

7.3.1.11.2MQTT Broker Connection Wizard

With the MQTT Broker Connection Wizard you can create and modify a data source connection to an MQTT Broker.

- Specify connection settings @ MQTT Broker Connection Wizard - Connection Settings
- Specify communication settings @ MQTT Broker Connection Wizard - Communication Settings
- Specify address space definition @ MQTT Broker Connection Wizard - Topic Configuration
- Review the topic hierarchies and their payloads @ MQTT Broker Connection Wizard - Topic Payload Overview

This page allows to specify connection settings such as e.g. the unique connection name for identifying the connection.



Connection Name

A unique identifier for the current connection. In case the provided name is already used by other connection, the user is not allowed to proceed to the next wizard page.

Client ID

The client ID must be unique for the respective MQTT Broker. If two MQTT clients are using the identical client ID, the broker is disconnected. The client (as e.g. dataFEED) cannot recognize that the client ID is already used by another client.

The MQTT Broker can be configured in a way that it only accepts connections with specific client IDs.

Connection active

This flag specifies if the current connection will be activated or not.

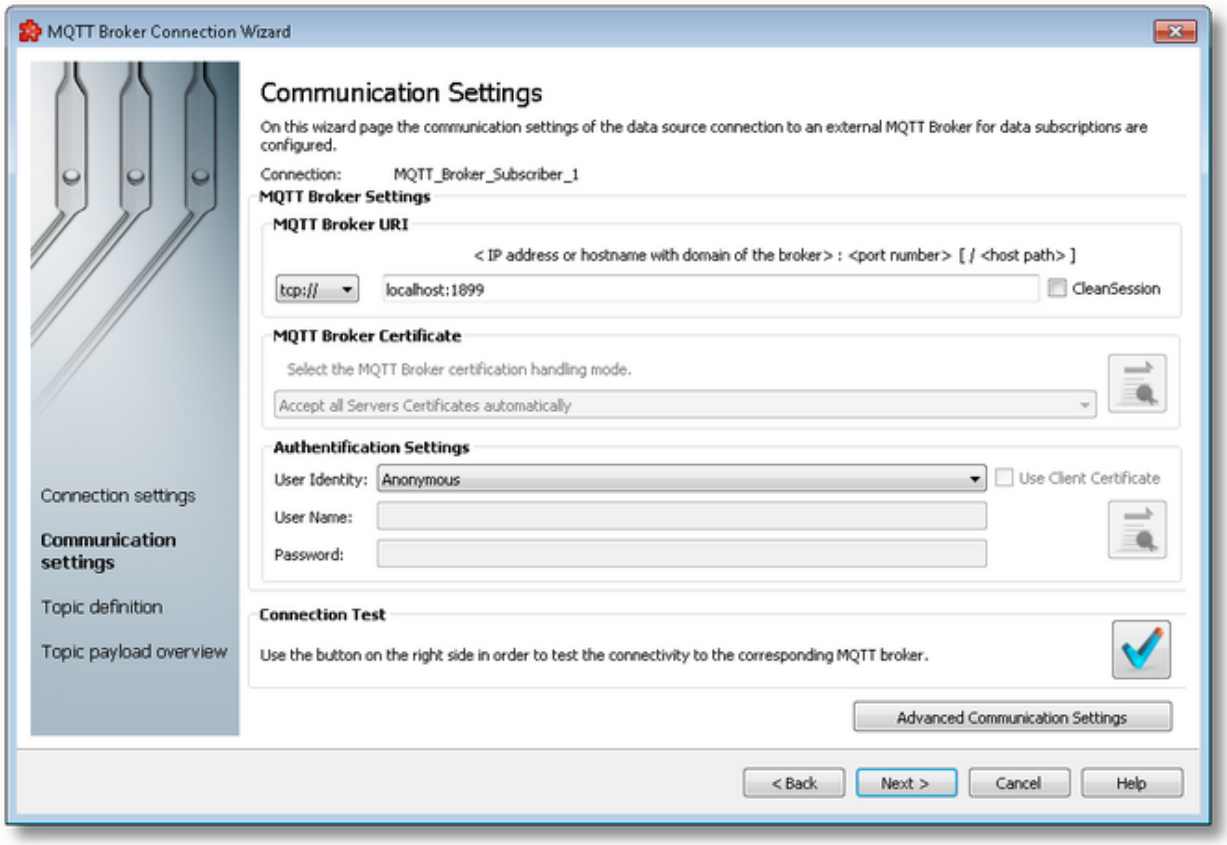
Payload format

This flag specifies the format of the topic payload (JSON or XML).

© Softing Industrial Automation GmbH

Connection Wizard - Comunication Settings >>>

This page allows to parameterize the communication settings and the authentication method to the MQTT Broker.



Broker URI

The Broker URI is composed of the transport protocol, the Full Qualified Domain name (FQDN, consisting of host name and domain name [see wikipedia]) optionally followed by the port number separated by a colon.

For the transport protocol, the following variants could be chosen:

1. tcp:// MQTT uses the plain TCP. MQTT brokers usually offer this service at port 1883.
2. ssl:// MQTT uses SSL or TLS security upon TCP. MQTT brokers usually offer this service at port 8883.
3. ws:// MQTT over WebSockets. MQTT brokers usually offer this service at port 8000.
4. wss:// MQTT over secure WebSockets. MQTT brokers usually offer this service at port 443.

The FQDNS can be replaced by the corresponding IPv4- or IPv6-address.

Note: By default, for *RabbitMQ* broker the *WebSocket* endpoint is available on the */ws* path and for *Secure WebSo*cket endpoint the path is */ws*.
E.g.: ws://localhost:15675/ws and wss://localhost:15675/ws

MQTT Broker Certificate

If the transport protocol ssl:// has been chosen, you may configure here how the identity of the MQTT broker should be verified by the dataFEED MQTT subscriber. You have the choice between the following options:

1. Accept all Server certificates automatically: This option disables verification of the broker identity. The broker just needs to provide some valid certificate.

2. Accept Trusted Certificates only: Connections are only completely established, if the broker proofs its identity with the certificate chain, which has been assigned to the connection. This option is only available, if you have assigned a broker certificate chain for this dataFEED connection with the Certificates dialog.

Authentication settings

Depending on the MQTT Broker configuration dataFEED may need to identify itself as an MQTT client with user name and password or by a client certificate.

In this case you need to select the entry "Username and Password" in "User Identity" Then fill out the input fields for user name and password accordingly. If needed activate the "Use Client Certificate" check box and assign a corresponding client certificate file through the MQTT Client Certificate dialog.

If no identification is required, leave the entry "Anonymous" in "User Identity".

Note: dataFEED OPC Suite does not use a strong encryption to protect the passwords contained in your configuration files. Therefore we recommend, to protect the PC where dataFEED OPC Suite is installed against unauthorized access. In addition you can use a hard disk encryption for protection purposes.

Connection test

Use the button Connection Test  to allow dataFEED OPC Suite establishing a connection to the MQTT Broker with the specified parameters for test purposes.

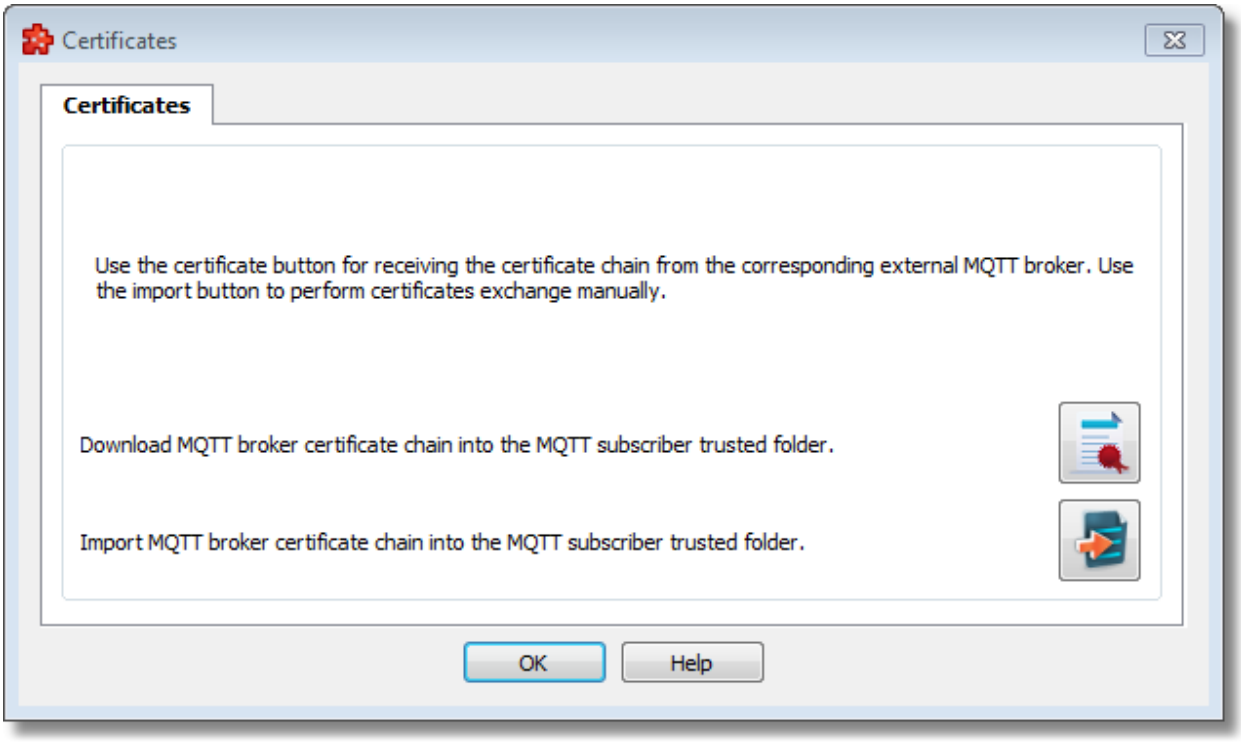
Advanced Communication Settings

© Softing Industrial Automation GmbH

Specific communication protocol settings related to the MQTT Broker connection can be provided through the Advanced Communication Settings dialog.

<<< Connection Wizard - Connection Settings Connection Wizard - Topic Configuration >>>

This dialog provides options to facilitate the secure communication between client and external broker by certificate handling.



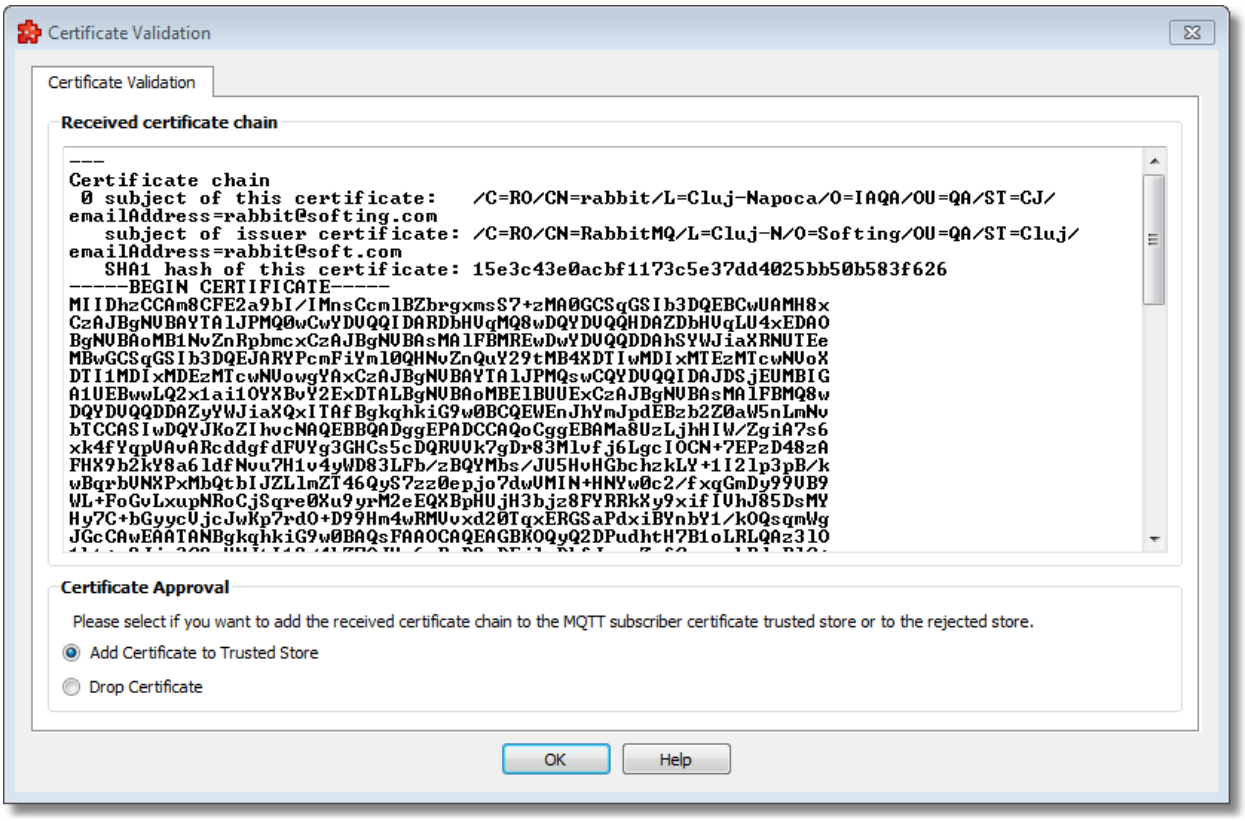
Download MQTT broker Certificate chain into the MQTT subscriber trusted folder

By pressing the receive broker certificate button  the user initiates a server connection and implicitly a certificate exchange. Once the server certificate chain is obtained it is presented to the user into certificate validation to be investigated. If the user accepts the certificate chain it would then added to the trusted folder and assigned to the connection. Otherwise the certificate chain is removed from the file system.

Import MQTT broker Certificate chain into the MQTT subscriber trusted folder

By using the import button  the user is prompted to select a broker certificate chain (*.crt or .pem* format) that will be added to the trusted folder. This operation basically can substitute Receive server certificate operation.

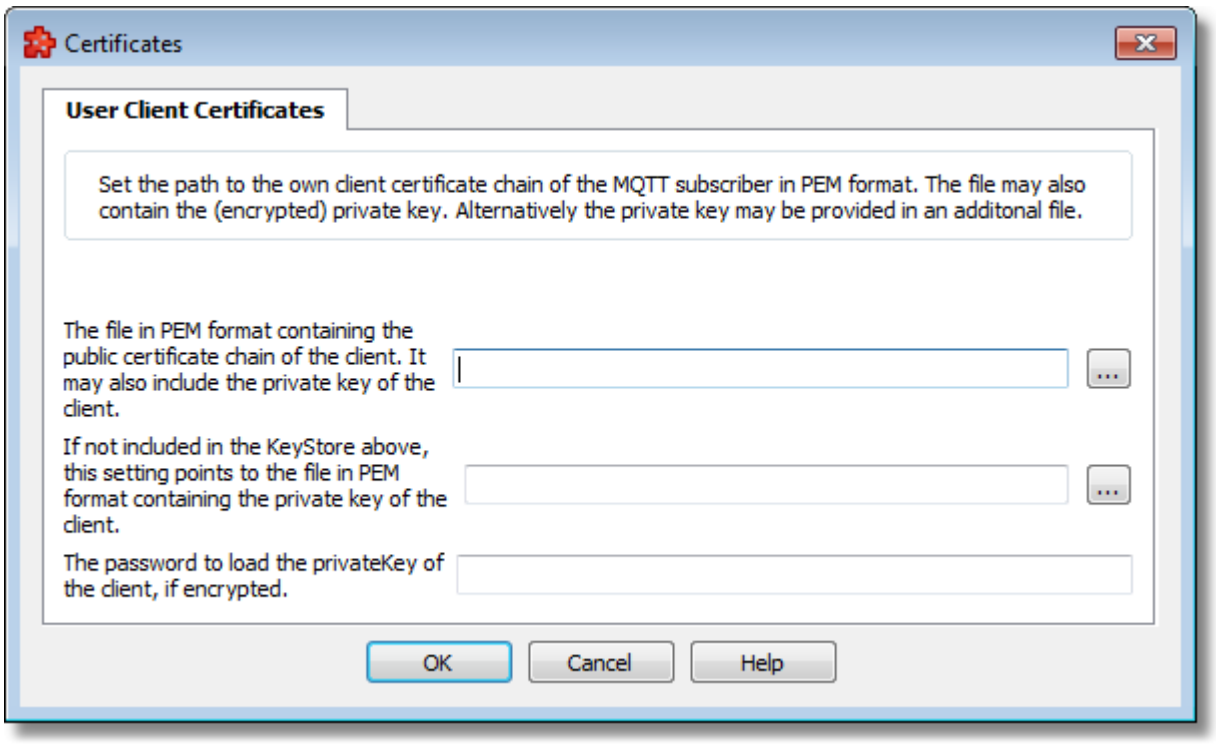
This dialog displays the received certificate chain of the MQTT broker.



The user may accept or discard the received certificate chain. If the user accepts the certificate chain it would then added to the trusted folder and assigned to the connection. Otherwise the certificate chain is removed from the file system.

This dialog allows to select the client certificate, which is intended to identify the client by the MQTT broker.

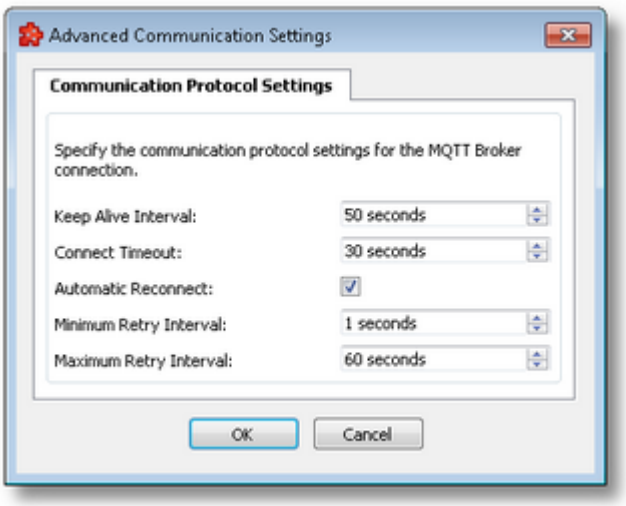
© Softing Industrial Automation GmbH



This dialog provides the following input fields:

1. The file in PEM format containing the public certificate chain of the client. It may also include the private key of the client.
2. If not included in the KeyStore above, this setting points to the file in PEM format containing the private key of the client.
3. The password to load the privateKey of the client, if encrypted.

The dialog allows to set specific communication protocol settings for the MQTT Broker connection.



Keep Alive Interval:

- The maximum time that should pass without communication between the client and the server. Set to 0 if no keep alive processing is needed (default value 50 seconds).

Connect Timeout:

- The time interval in seconds to allow a connect to complete (default value 30 seconds).

Automatic Reconnect:

- Reconnect automatically in the case of a connection being lost (default value enabled).

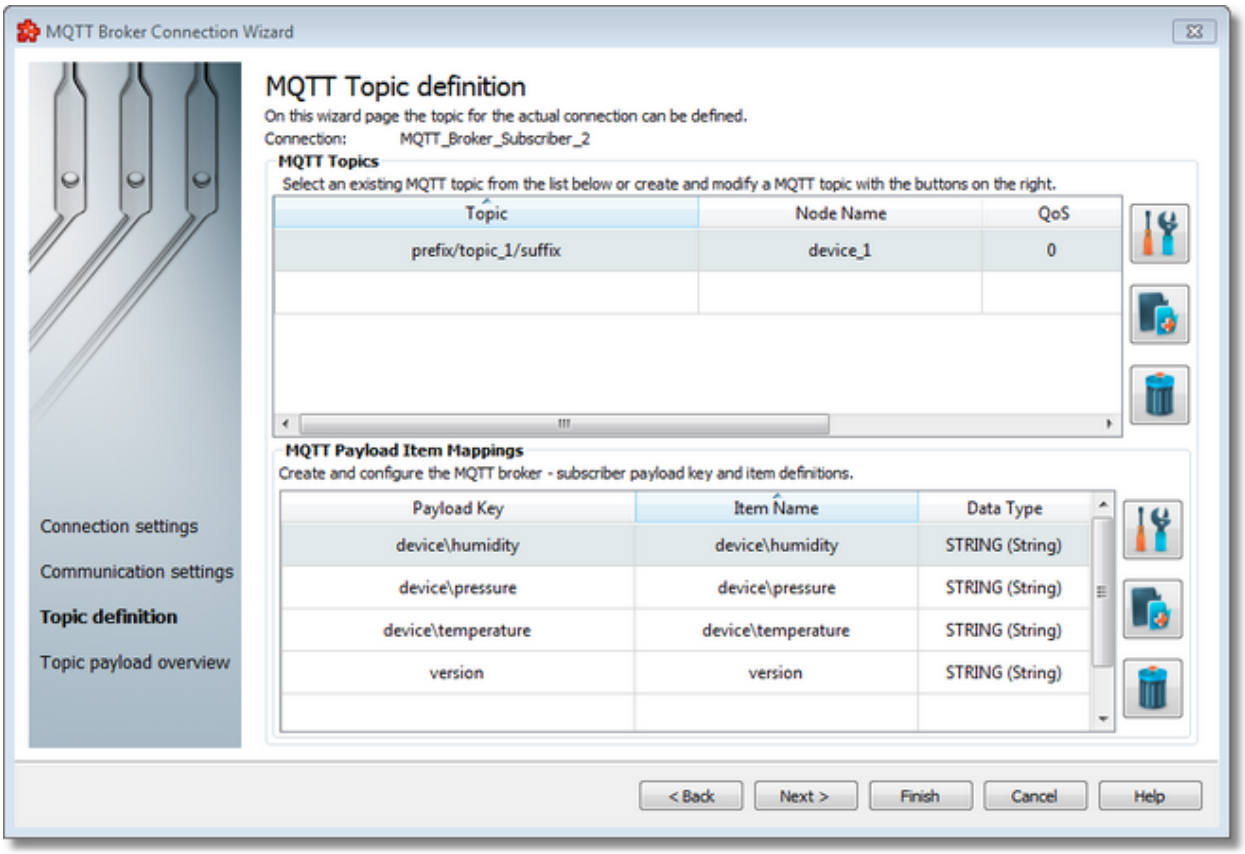
Minimum Retry Interval:

- Minimum retry interval in seconds. Doubled on each failed retry (default value 1 second).

Maximum Retry Interval:

- Maximum retry interval in seconds. The doubling stops here on failed retries (default value 60 seconds).

This page allows to configure the topic definition of the *MQTT Broker* subscriber connection.



MQTT topics and MQTT payload key / item mappings definition

© Softing Industrial Automation GmbH

An editable table view where the user can change the topic structure and the payload keys that are mapped to the internal addresses.

- **Topics:** these elements are associated with branches in tree. The Softing dataFEED OPC Suite will subscribe these topics to an external *MQTT* broker. The user can change the prefix or the suffix of these topics.
- **Payload key / item mappings:** these elements are used for mapping the received message from the *MQTT* broker with the internal address space of the connection.

Operations for topic definition

Depending on the current selection in the topic-tree, the following operations are possible using the buttons on the right, the context menu or the corresponding shortcuts:

- Topic properties (Ctrl+P) 
- Add new topic (Ctrl+A) 
- Delete topic(s) (Del) 

Additional operations are accessible only from the context menu:

- Import 
- Export 

Delete item(s)

Triggering the *Delete topic(s)*  action with the corresponding button on the wizard's right hand side or *Delete* from the context menu in order to remove the currently selected item(s).

Import

Triggering the Import  action on the connection root, with no rows selected in the MQTT topics definition table, will start the Import wizard.

Export

Triggering the Export  action on single or multiple rows selected in the MQTT topics definition table, will start the Export wizard.

Attention: A node that is currently used in a data processing or data destination action can not be deleted.

Operations for payload and item mapping

Depending on the current selection in the topic-tree and the payload mapping row the following operations are possible using the buttons on the right, the context menu or the corresponding shortcuts:

- Payload properties (Ctrl+Shift+P) 
- Add new payload (Ctrl+Shift+A) 
- Delete payload(s) (Shift+Del) 

Additional operations are accessible only from the context menu:

- Import 
- Export 

Delete item(s)

Triggering the *Delete payload(s)*  action with the corresponding button on the right wizard side or *Delete* from the context menu will remove the currently selected payload key and the mapped item.

Import

Triggering the Import  action for the currently selected MQTT Topic, with no rows selected in the MQTT payload key item mapping definition table, will start the Import wizard.

Export

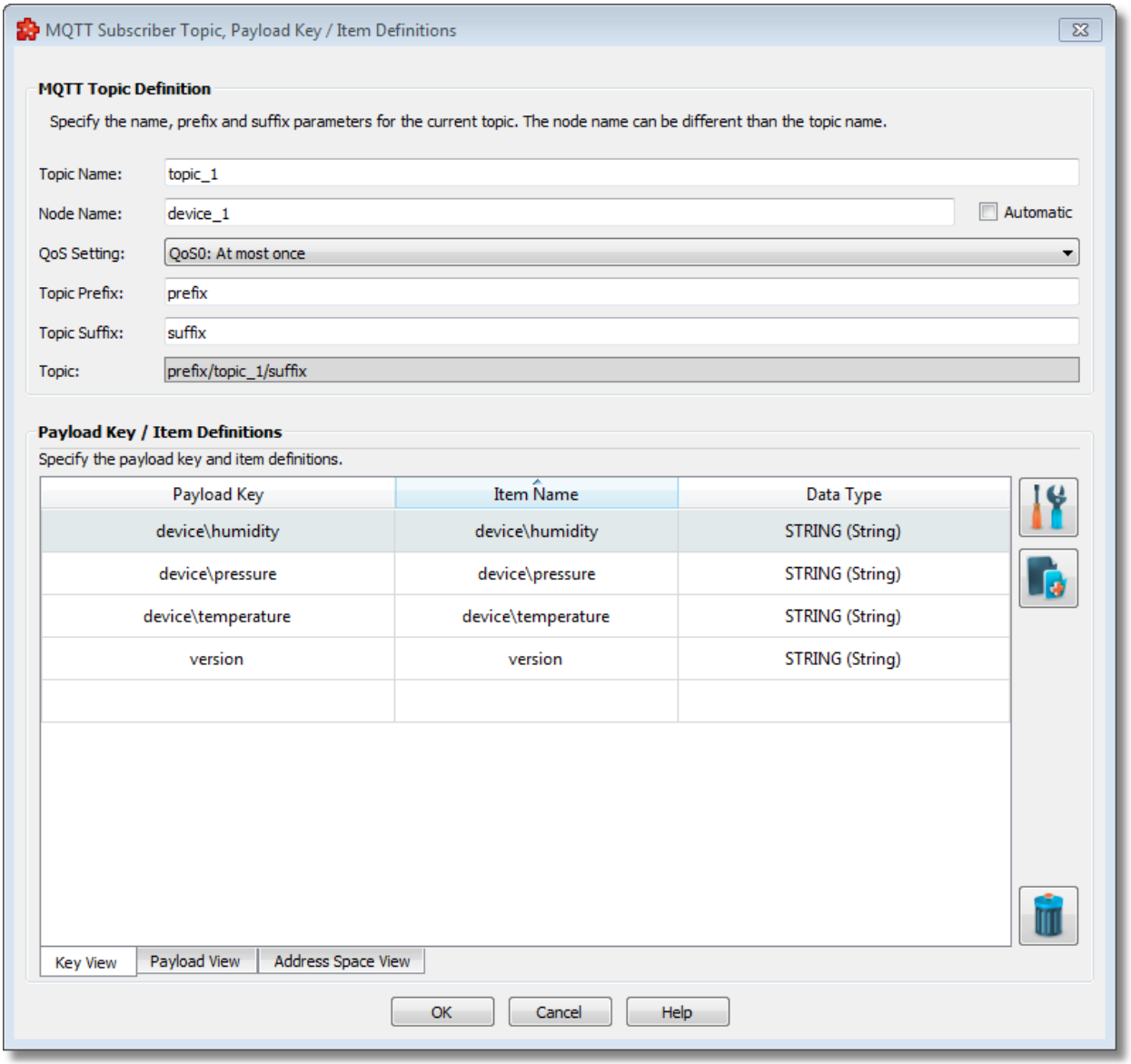
© Softing Industrial Automation GmbH

Triggering the Export  action on single or multiple rows selected in the MQTT payload key item mapping definitions table, will start the Export wizard.

Attention: A tag that is currently used in a data processing or data destination action can not be deleted.

<<< Connection Wizard - Communication Settings Connection Wizard - Topic Payload Overview>>>

The *MQTT Subscriber Topic, Payload Key / Item Definition* section allows to add or modify MQTT topic definitions such as names, prefix and suffix, and to add or modify payload key / item mapping definitions for the *MQTT Broker* subscriber connection.



Topic properties / Add new topic

Topic Name

This edit line allows to specify a new topic name or to modify an existing one.

Node Name

Here you can enter the item name for the current hierarchy level. The user can specify a node name that is different than the topic name.

Note: Characters *Tab* and *Newline* cannot be used for topic names. Characters '.', '|', '/', '+', '#', *Tab* and *Newline* cannot be used for node names. If the topic name requires the usage of a character that would not be allowed in the node name, please specify a node name that is different than the topic name, by disabling the *Automatic* check box and changing the node name manually.

Note: Topic prefixes and suffixes also disallow the usage of the characters '.', '|', '+', '#', *Tab* and *Newline*, but allow to use the character '/'.

Note: The topics and the node names that you are using must be unique for the corresponding connection.

QoS Settings

The QoS setting is used to control the delivery of MQTT messages in case of TCP connection loss. The following settings are possible:

1. QoS0: The MQTT message is published with the safety of the current TCP connection.
2. QoS1: The MQTT messge is repeated in the next TCP connection, if the delivery of the message in the current TCP connection fails.
3. QoS2: The MQTT protocol ensure that a message is delivered exactly once.

Topic Prefix and Suffix

Here you can enter the prefix and suffix of the defined topic. Special characters such as '.', '|', '+', '#', *Tab* and *Newline* must not be used.

Syntax

Here you can see the syntax of the defined topic. This information is read only. The Softing dataFEED OPC Suite will subscribe these topic syntax to an external *MQTT* broker

© Softing Industrial Automation GmbH

Operations for payload and item mapping

Depending on the current selection in the topic-tree and the payload mapping row the following operations are possible using the buttons on the right, the context menu or the corresponding shortcuts:

- Payload properties (Ctrl+P) 
- Add new payload (Ctrl+A) 
- Delete payload(s) (Del) 

Additional operations are accessible only from the context menu:

- Import 
- Export 

Import

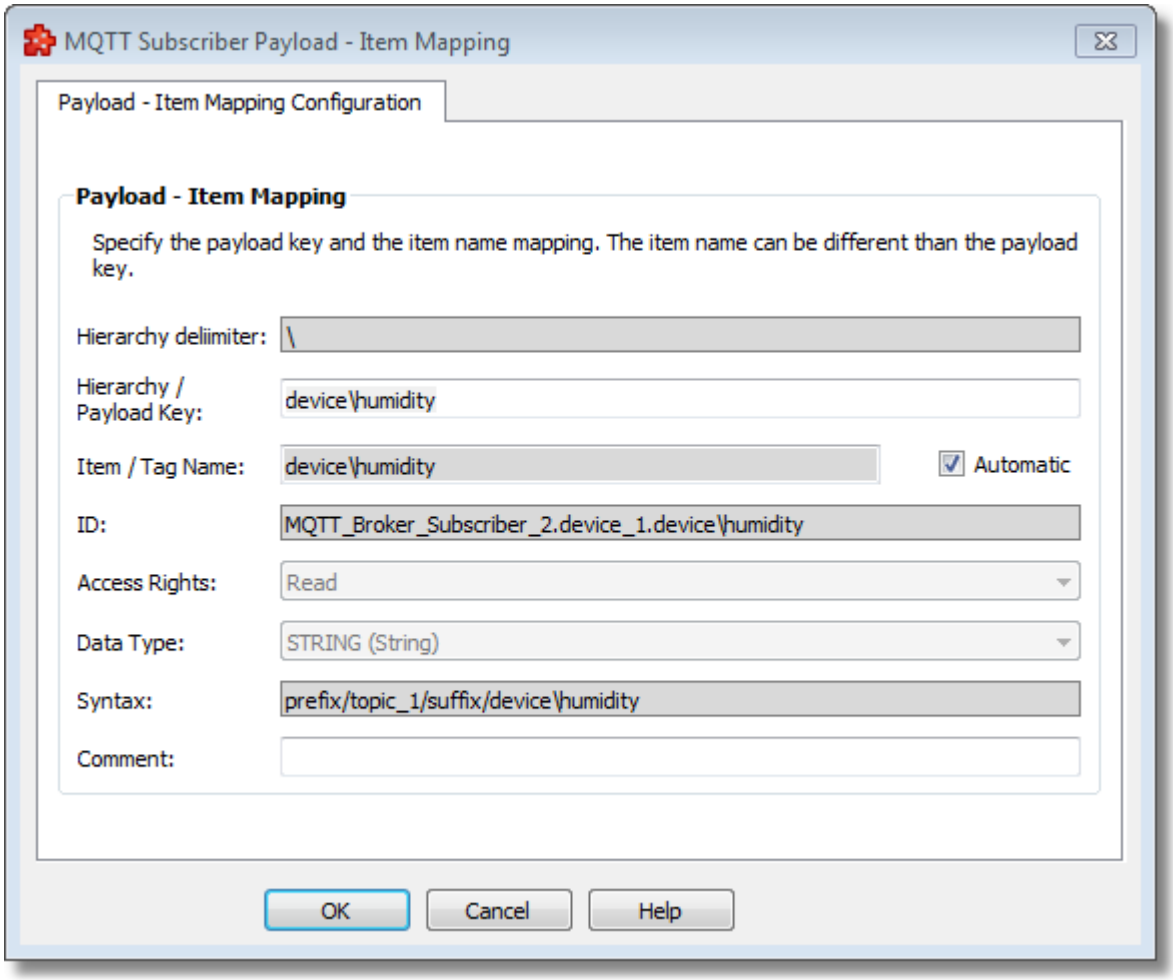
Triggering the Import  action for the currently selected MQTT Topic, with no rows selected in the MQTT payload key item mapping definition table, will start the Import wizard.

Export

Triggering the Export  action on single or multiple rows selected in the MQTT payload key item mapping definitions table, will start the Export wizard.

Payload Key / Item Definition

The *MQTT Subscriber Payload - Item Mapping* section allows to add or modify the payload keys and items / tag names of the *MQTT Broker* subscriber connection and to create the mapping relation between them.



Hierarchy delimiter

This not editable field displays the value of the namespace delimiter that can be used in the expression of the payload key and the item/tag name, for MQTT subscriber namespace hierarchy representation.

Hierarchy / Payload Key

This edit line allows to specify a new payload key or modify an existing one. Special characters such as *Tab* and *Newline* must not be used. The name you are using must be unique for the corresponding parent node.

Item / Tag Name

Here you can enter the item name for the current hierarchy level. The user can specify a item name that is different than the payload key. The name you are using must be unique for the corresponding parent node. The name may only consist of printable ASCII characters. Special characters such as '.', '|', '/', '+', '#', *Tab* and *Newline* must not be used.

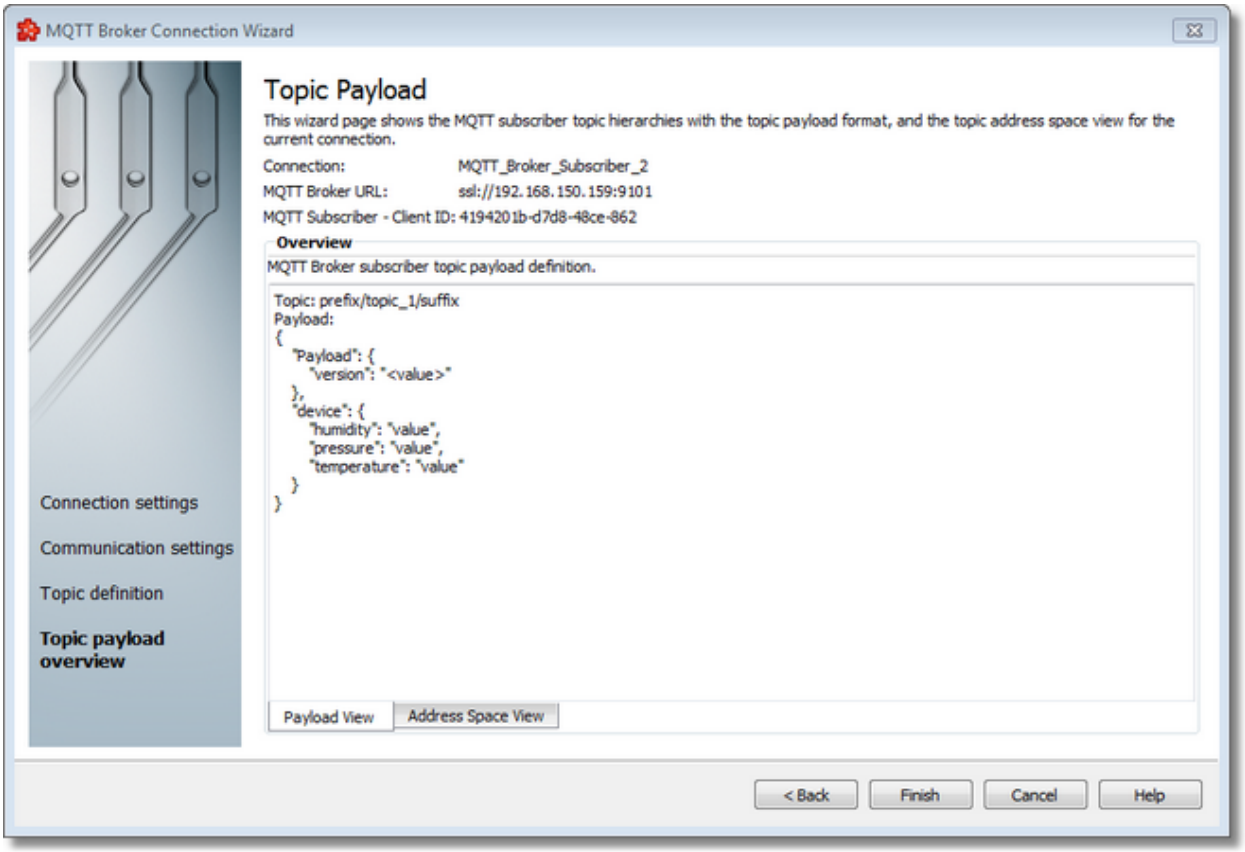
© Softing Industrial Automation GmbH

If the payload key requires the usage of a character that would not be allowed in the item name, please specify an item name that is different than the payload key, by disabling the *Automatic* check box and changing the item name manually.

Comment

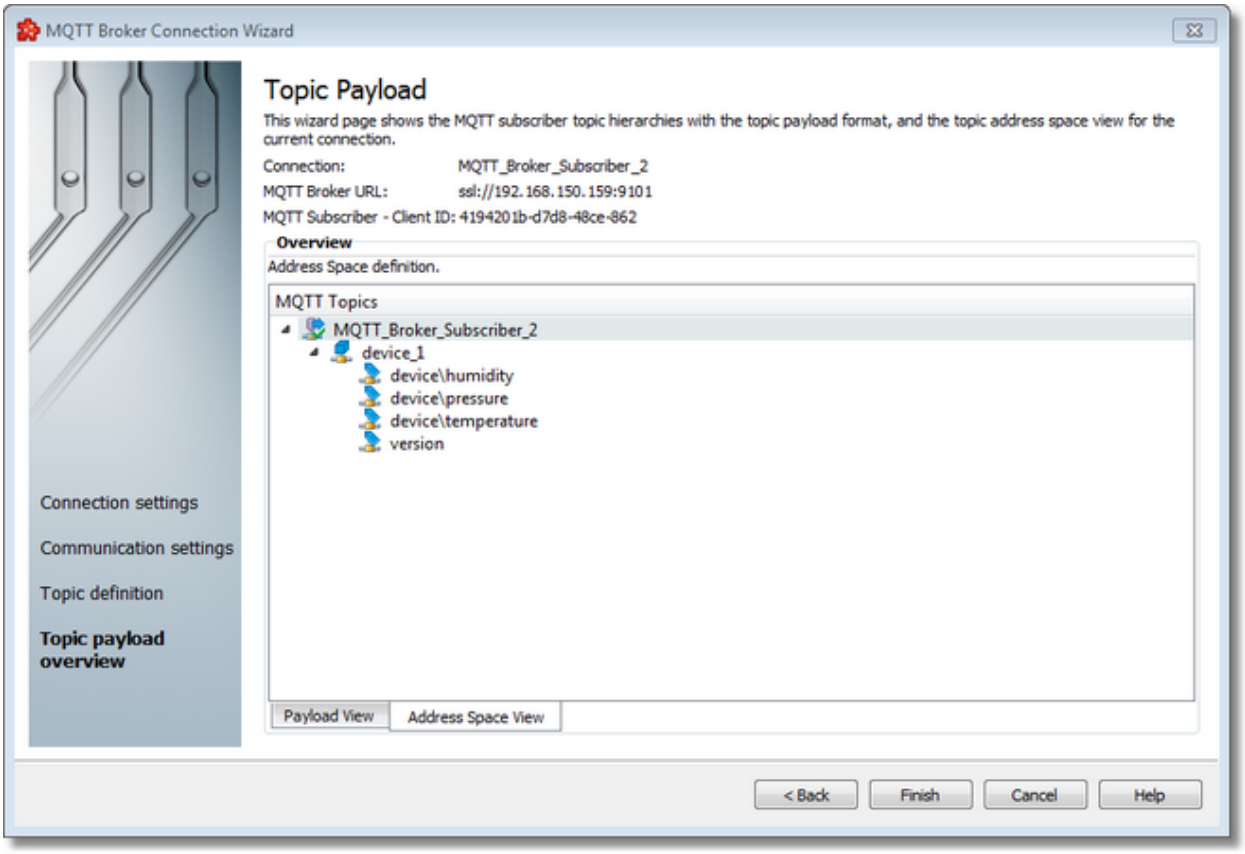
This edit line allows to specify a comment or modify an existing one.

On this wizard page the user is presented with the final form of the content for the current connection, regarding the MQTT Broker subscriber topic hierarchies and the topic payload JSON or XML formats.



This is a read-only view, used just for reviewing the correctness of the content. The user can only copy the content from the topic payload definition pane.

The user can view the address space of the current connection in a separate tab.



<<< Connection Wizard - Topic Definition

7.3.1.11.3Import Wizard

The Softing Configurator allows the import of MQTT Broker Subscriber connection items from an external file.

The import action can be triggered by selecting the *Import* action in the context sensitive menu  .

Note: The import action is available for the connection root item and for items of type *Node* from the namespace.

The import action can be triggered from the context sensitive menu of the:

- MQTT topic definition table, when no rows are selected in the table.

In this case imported files can contain topics and child payload key item mapping definitions. The import operation is performed on the MQTT Broker Subscriber connection root. Files that contain only payload key item mapping definitions with no topic parent, cannot be imported.

- MQTT payload key item mapping definition table, when no rows are selected in the table.

© Softing Industrial Automation GmbH

In this case imported files can contain child payload key item mapping definitions. The import operation is performed on the MQTT topic that contains the child payload key item mappings listed in the current table. Files that contain topics and child payload key item mapping definitions, cannot be imported.

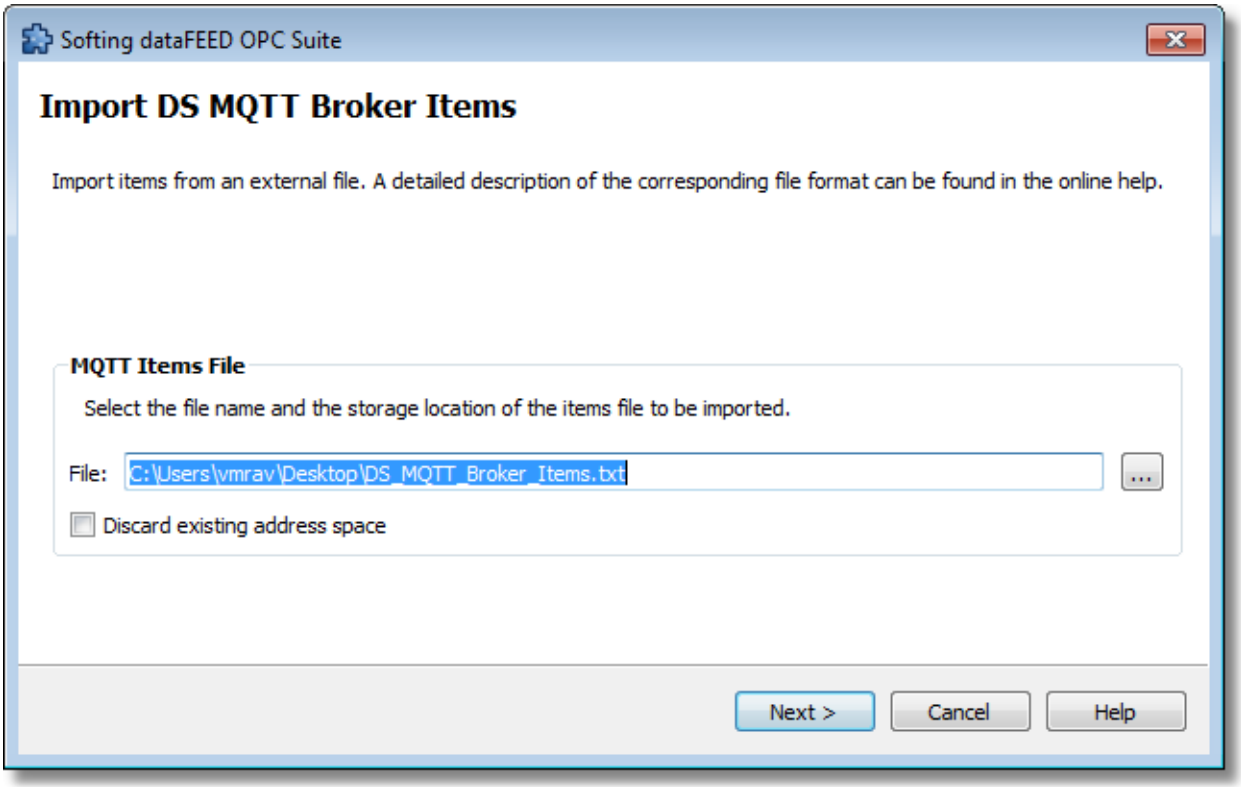
When the import action is triggered, the Data Source MQTT Broker Items *Import Wizard* is started.

▾ The Filter Import Wizard allows importing different data items:

- Select the file to be imported @ Import Wizard - Import File Selection
- Examine the import results @ Import Wizard - Import Results

The Import DS MQTT Broker Items wizard page allows you to choose a file to be imported. Text files with extensions *.txt and *.dat are supported.

The file can be selected by manually typing its name in the *File* input box or by clicking the browse button located to the right of the input box (a standard file selection dialog will open).



The file path and name should be valid (they should point to an existing file).

The already existing address space can be deleted. It is overwritten with the imported address space.

In addition it is allowed to merge the existing address space with the newly imported one. The merging conflicts are resolved by overwriting the existing identifiers with the newly imported ones.

Note: If item identifiers for items that are used as data processing or data destination items cause a conflict, the import operation for these items will not be performed and these items will not be overwritten.

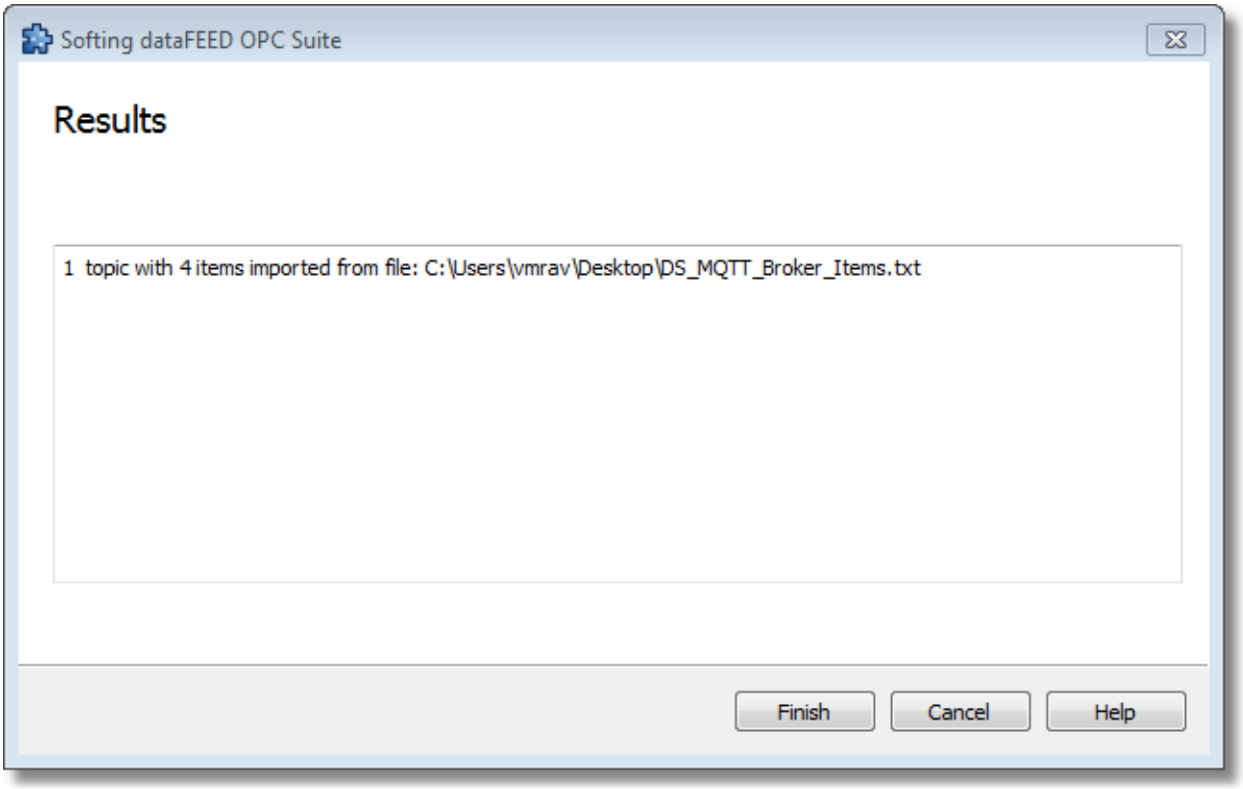
When you press *Next*, the file path and name are verified. If the file identified by the specified file path and name does not exist, an error message is displayed.

The actual import begins if the selected file is valid.

After pressing *Next*, the file is analyzed and the wizard is continued with the import operation itself. When the import operation is finished, the *Import Results* page is displayed.

Import Wizard - Import Results >>>

The Import Results wizard page displays the import results.



This page is the final page of the Filter Import Wizard. A back button is not provided as the import action cannot be undone.

After the import the results page contains the following information:

© Softing Industrial Automation GmbH

- The absolute path of the imported file.
- The number of items imported. This number includes items of type topic and child items (if the imported files contains topic and child payload key item mapping definitions), or topic child items (if the imported file contains payload key item mapping definitions).
- The number of items updated. In this case the items with the same identifiers as contained in the file to be imported are overwritten with the newly imported ones, in case the items are not used as data destination or data processing items.

After the import the results page can also contain warning messages:

- if some items of the existing address space cannot be overwritten, if they are used at least once as data destination or data processing items.
- if errors occurred during the import operation at specific line indexes of the file.
- if in the MQTT topic definitions table, a file is imported containing MQTT payload key item mapping definitions.
- if in the MQTT payload key item mapping definitions table, a file is imported containing MQTT topics and child payload key item mapping definitions.

When the import wizard is closed after a successful import, the new items are added to the current MQTT Broker Subscriber connection's MQTT topic definition table or MQTT payload key item mapping definition table, depending on where the import operation was performed.

<<< Import Wizard - Import File Selection

7.3.1.11.4Export Wizard

The dataFEED OPC Suite Configurator allows exporting the MQTT Broker Subscriber connection items.

The export action can be triggered by selecting the *Export* action in the context sensitive menu  .

The export action can be triggered from the context sensitive menu of the:

- MQTT topic definition table, when a single or multiple MQTT topic selection is made in the table.

  The exported file will contain the definitions of selected the topics and their child payload key item mappings.

- MQTT payload key item mapping definition table, when a single or multiple MQTT payload key item mapping selection is made in the table.

  The exported file will contain the definitions of the selected MQTT payload key item mappings.

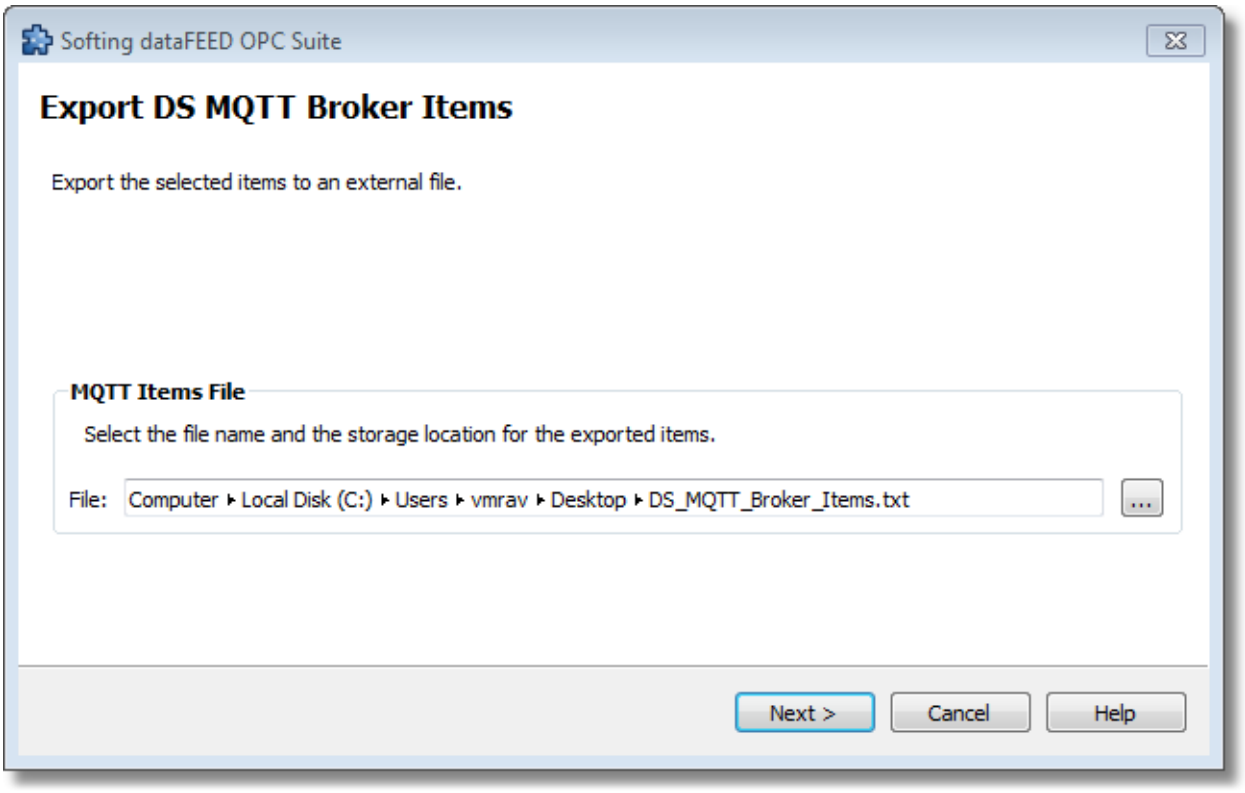
When the export action is triggered the *Filter Export Wizard* is started.

▾ The Filter Export Wizard allows exporting different data items:

- Select the file the data items are to be exported to @ Export Wizard - Export File Selection
- Examine the export results @ Export Wizard - Export Results

The Export File Selection wizard page allows you to choose a file name and location for the exported file. Text files with extensions *.txt and *.dat are supported.

The file can be selected by manually typing its path and name in the File input box or by clicking the browse button located to the right of the input box (a standard file saving dialog will open).



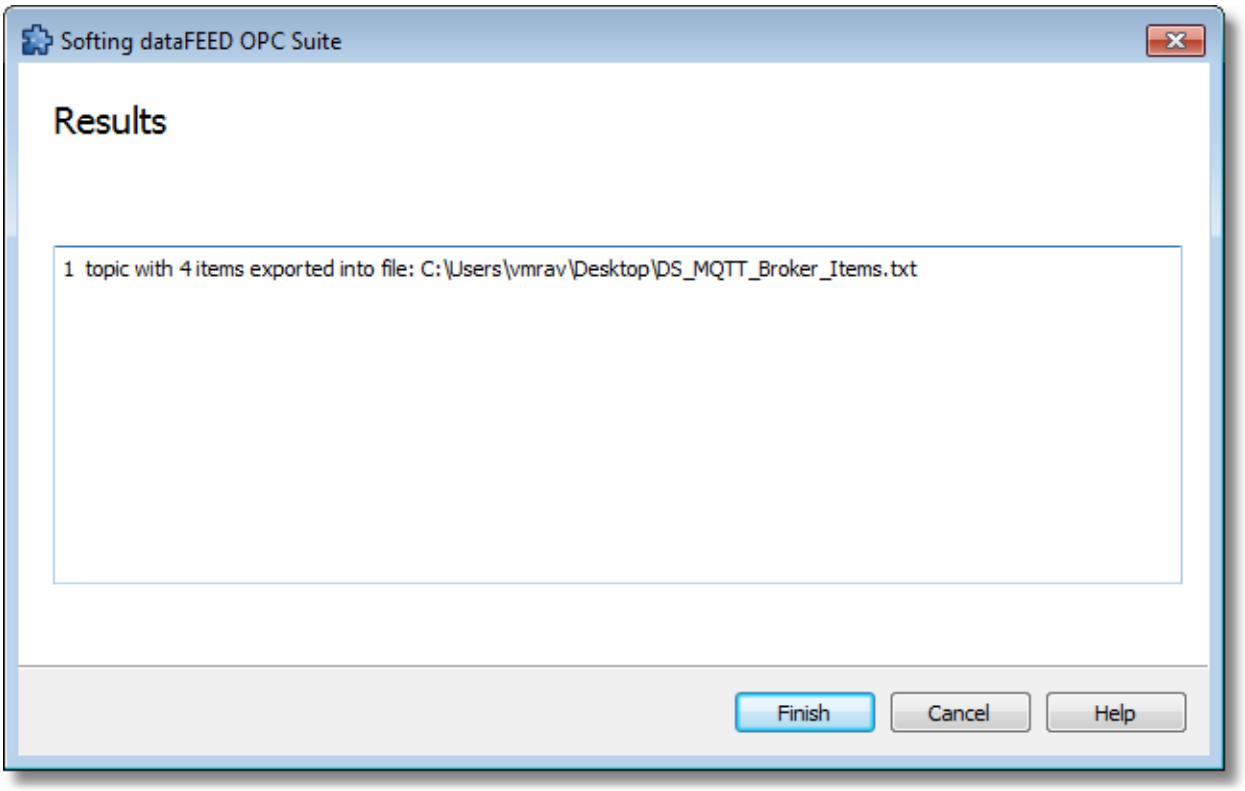
When the *Next* button is pressed the file path is verified and if the path does not exist, an error message is displayed (invalid path). If the specified file path is valid and a file with the specified name already exists, a message box will ask whether to overwrite the existing file or not (select another file). The actual file export operation begins if the path to the file is valid and either a file with the specified name does not exist or it exists and overwrite has been chosen.

© Softing Industrial Automation GmbH

When the export operation is finished the *Export Results* page is displayed.

Export Wizard - Export Results >>>

The Export Results wizard page displays the export results as shown in the image below:



This page is the final page of the Filter Export Wizard. A back button is not provided as the export action cannot be undone.

After the export the results page contains the following information:

- The absolute path of the exported file.
- The number of items exported. This number includes items of type topic and child items, or topic child items.

<<< Export Wizard - Export File Selection

7.3.1.11.5Define JSON and XML payload format

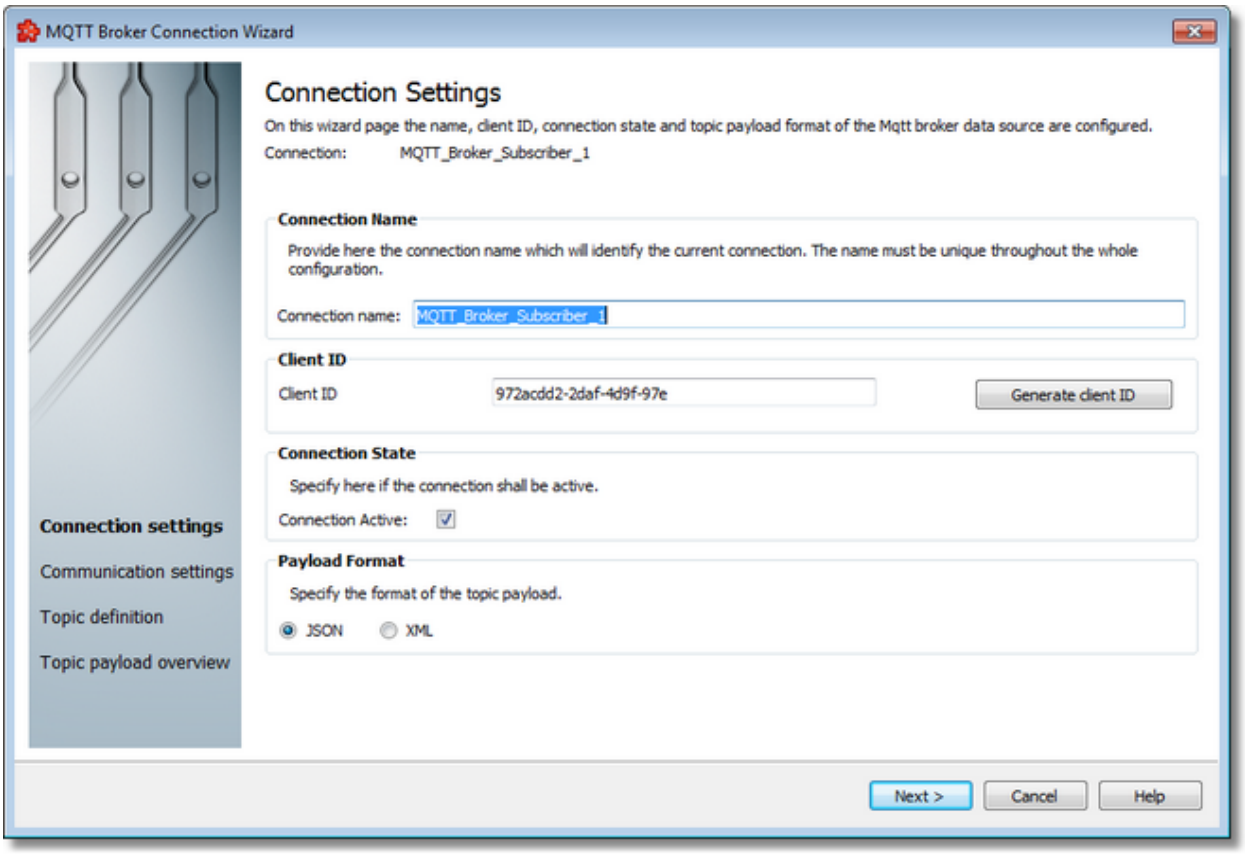
The dataFEED OPC Suite application supports connections to various *MQTT* brokers. This document describes how to configure and receive payloads in *JSON* and *XML* formats.

Download and install MQTT Mosquitto Broker

- Download and install the latest *Mosquitto* broker version form: https://mosquitto.org/download/

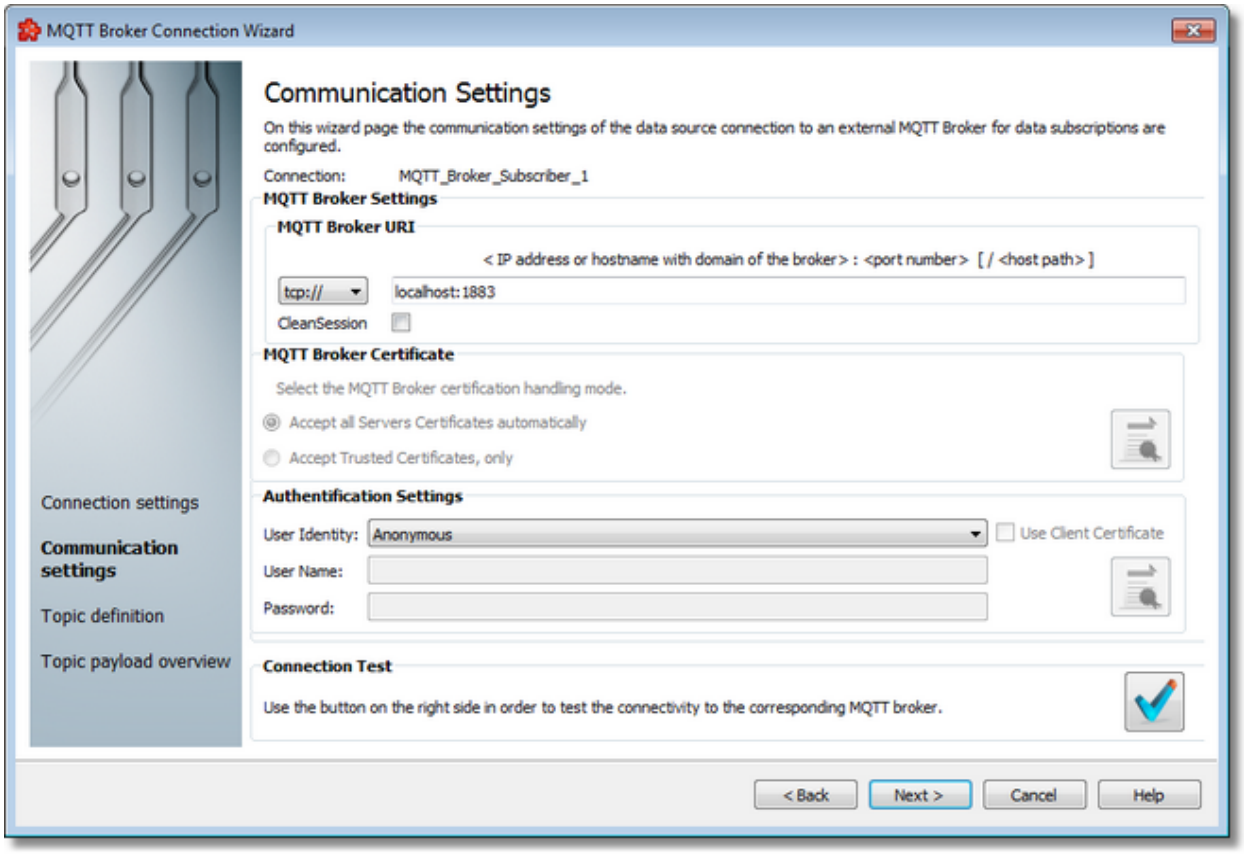
Connect *dataFEED OPC Suite* to MQTT Mosquitto Broker

- Start *dataFEED OPC Suite* Configurator
- Select *Data Source / MQTT Broker*
- Connect to *MQTT Mosquitto* broker using the *TCP* protocol on port 1883 with *JSON* payload format

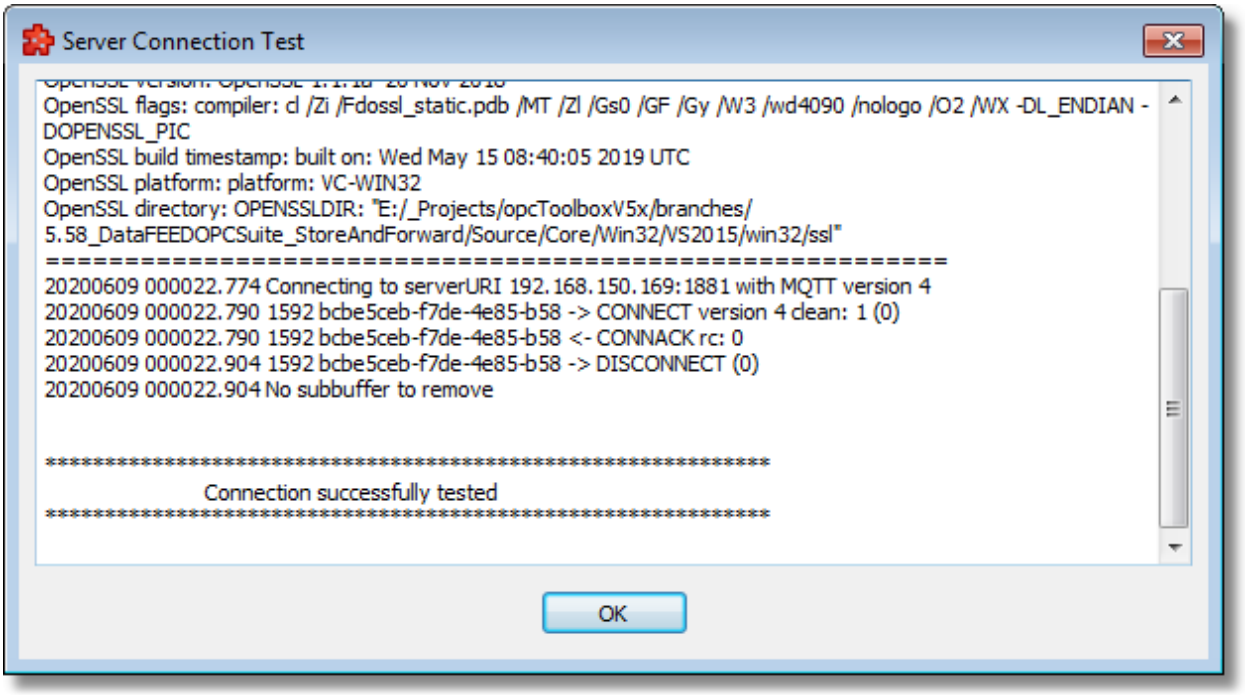


- Press *Next* button

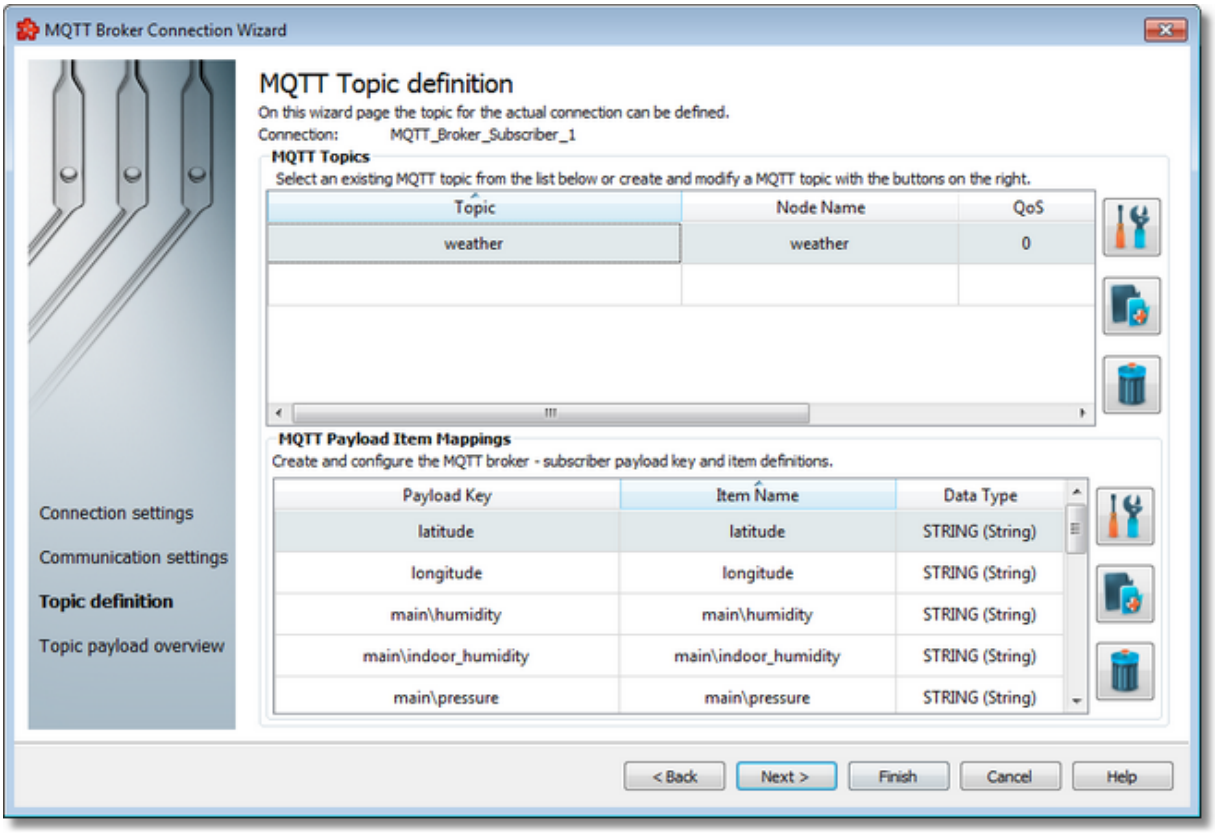
© Softing Industrial Automation GmbH



- Press Connection Test button to verify connection

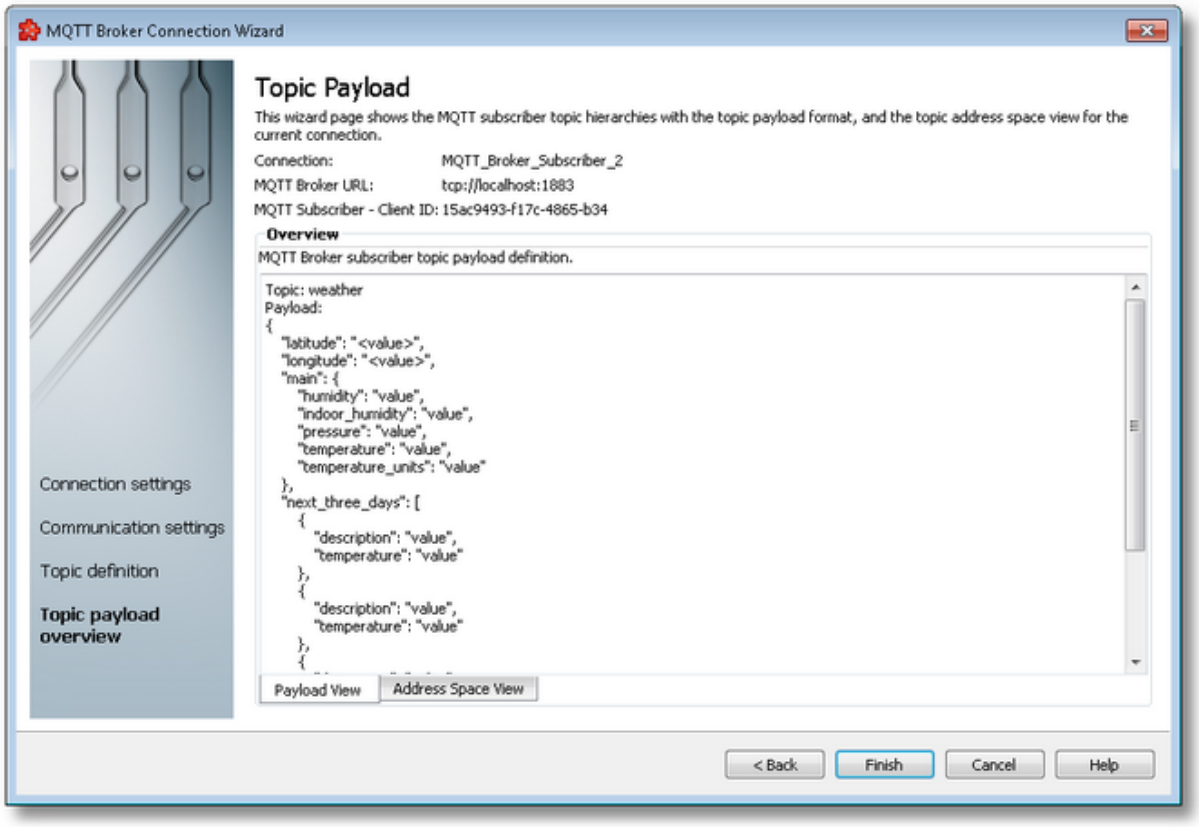


- Define the *MQTT* topic(s). The topics are subscribed to the broker and the payloads that are received in dataFEED OPC Suite will update the items from the address space that are mapped to the current topic via payload keys.



- Check the topic payload overview. The *JSON* published payload should respect the previous defined structure in order to be correctly handled by dataFEED OPC Suite

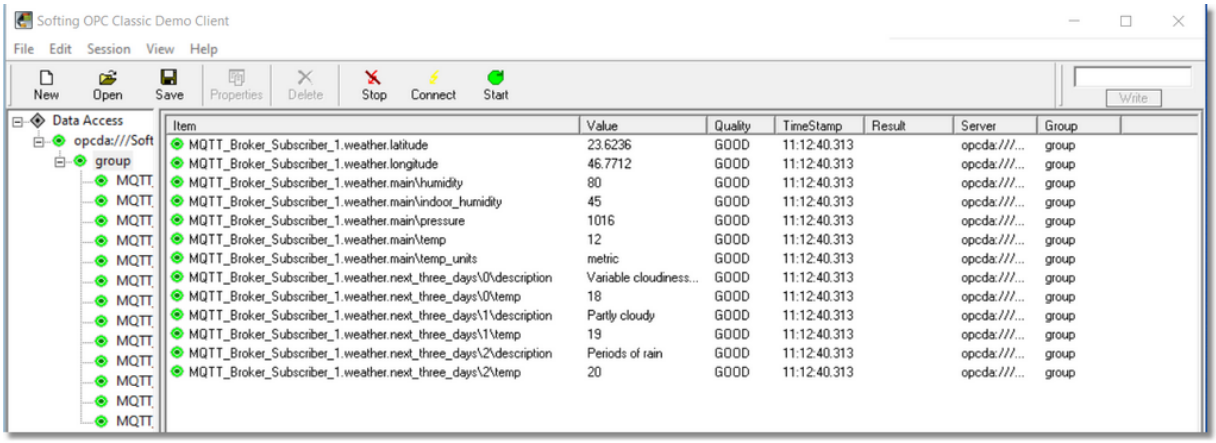
© Softing Industrial Automation GmbH



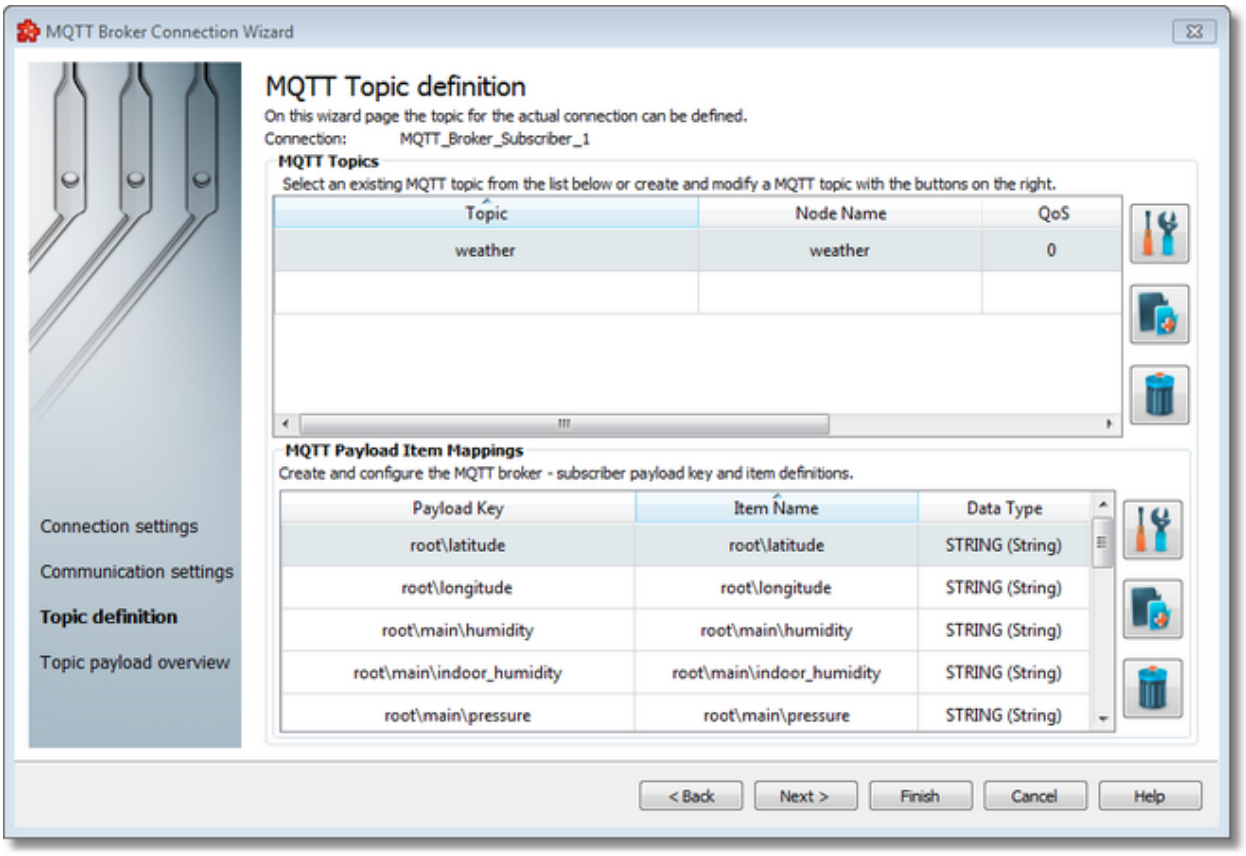
- Send the following payload using an external *MQTT* publisher

```
{
   "longitude":46.7712,
   "latitude":23.6236,
   "main":{
      "temp":12,
      "temp_units":"metric",
      "humidity":80,
      "indoor_humidity":45,
      "pressure":1016
   },
   "next_three_days":[
      {
         "temp":"18",
         "description":"Variable cloudiness with a couple of showers and a thunderstorm
      },
      {
         "temp":"19",
         "description":"Partly cloudy"
      },
      {
         "temp":"20",
         "description":"Periods of rain"
      }
   ]
}
```

- The publish request will be received in dataFEED OPC Suite

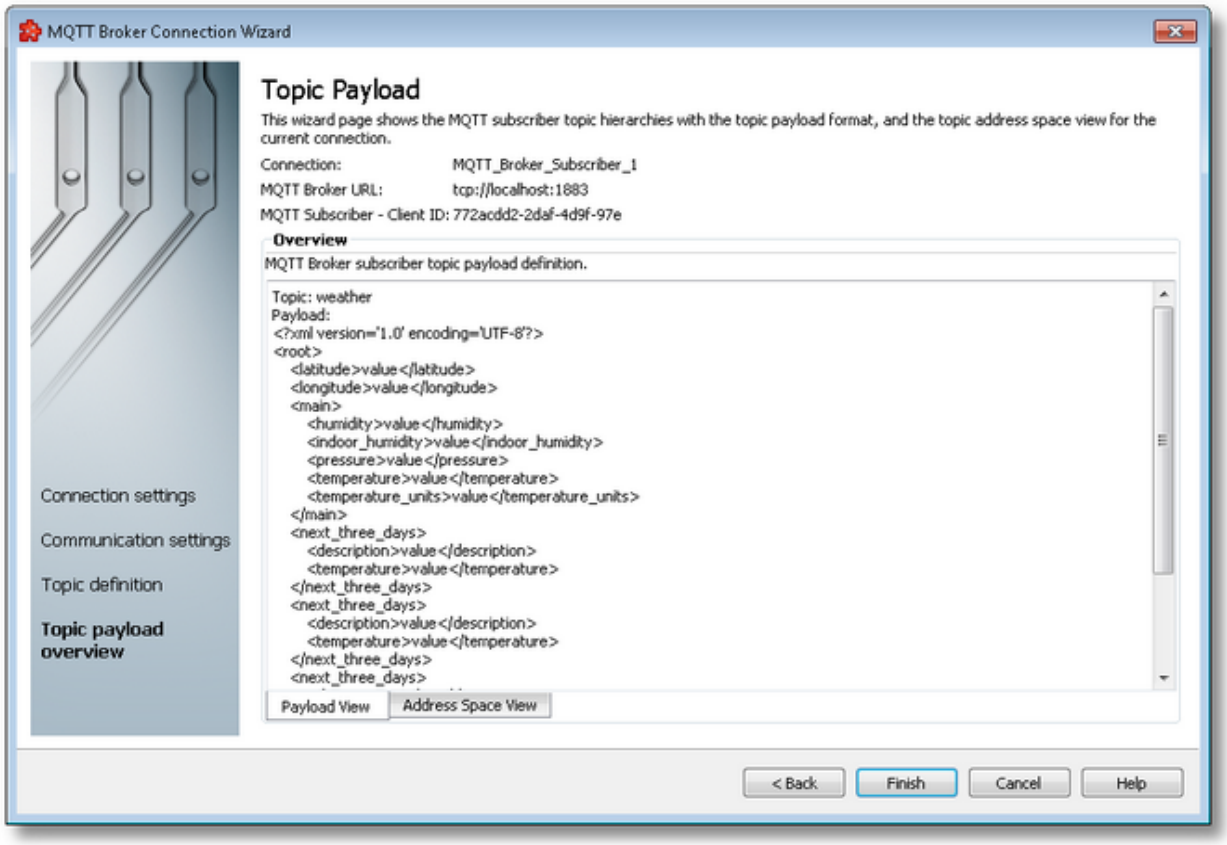


- Change the payload format to XML and modify the topic payload definition



- Check the topic payload overview. The XML published payload should respect the previous defined structure in order to be correctly handled by dataFEED OPC Suite.

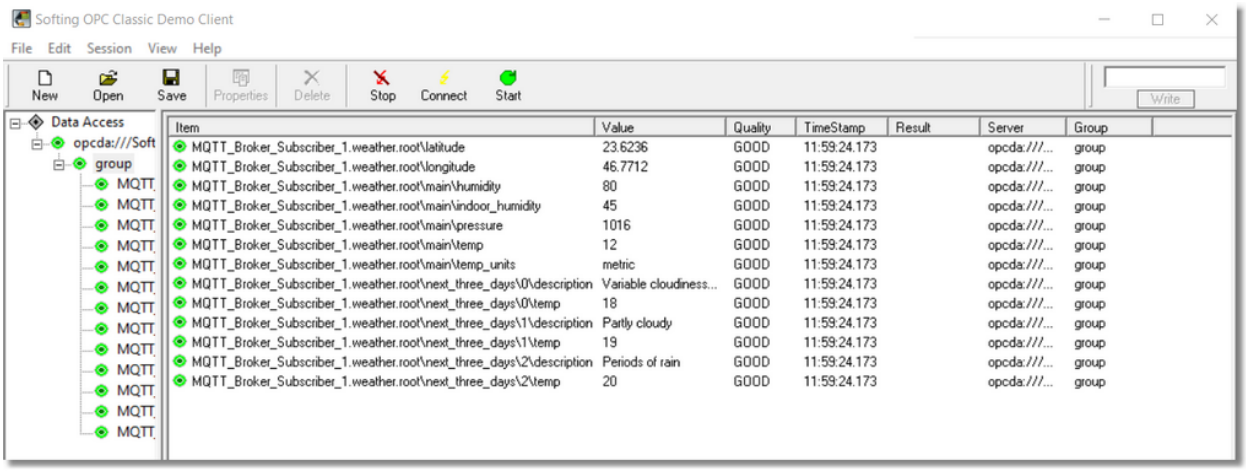
© Softing Industrial Automation GmbH



- Using an external publisher we can send the following xml structure in order to receive the data changes in dataFEED OPC Suite.

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
    <latitude>23.6236</latitude>
    <longitude>46.7712</longitude>
   <main>
          <humidity>80</humidity>
          <indoor_humidity>45</indoor_humidity>
          <pressure>1016</pressure>
          <temp>12</temp>
          <temp_units>metric</temp_units>
   </main>
    <next_three_days>
          <temp>18</temp>
          <description>Variable cloudiness with a couple of showers and a thunderstorm<
   </next_three_days>
   <next_three_days>
          <temp>19</temp>
          <description>Partly cloudy</description>
   </next_three_days>
   <next_three_days>
          <temp>20</temp>
          <description>Periods of rain</description>
   </next_three_days>
</root>
```

- The publish request will be received in dataFEED OPC Suite.



7.3.1.11.6Connect to Azure IoT Hub via SSL and WSS

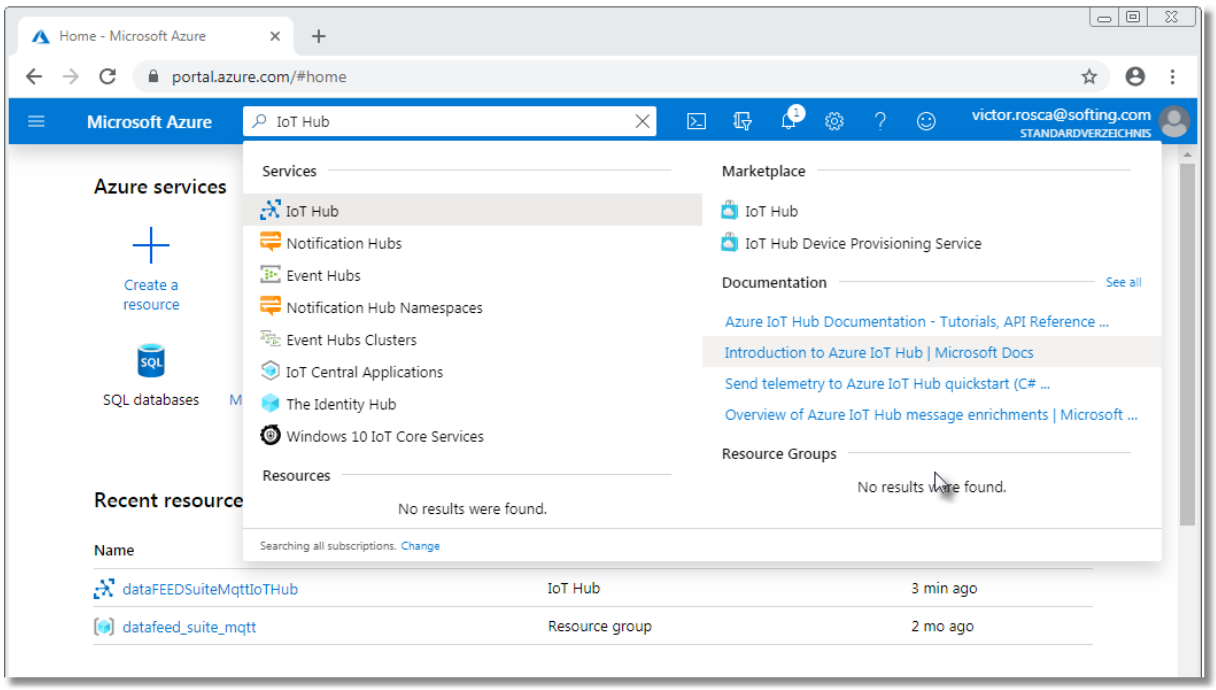
The dataFEED OPC Suite application supports connections with *Azure IoT Hub*. Devices can communicate with *IoT Hub in Azure* using *SSL* and *WSS* protocols, *SSL* via port *8883* and *WSS* via port *443*. In order to connect to *Azure IoT Hub* from *Data Source / MQTT Broker*, please follow the next steps.

Download *Device Explorer*

- Download the latest *Device Explorer* version form: https://github.com/Azure/azure-iot-explorer/releases

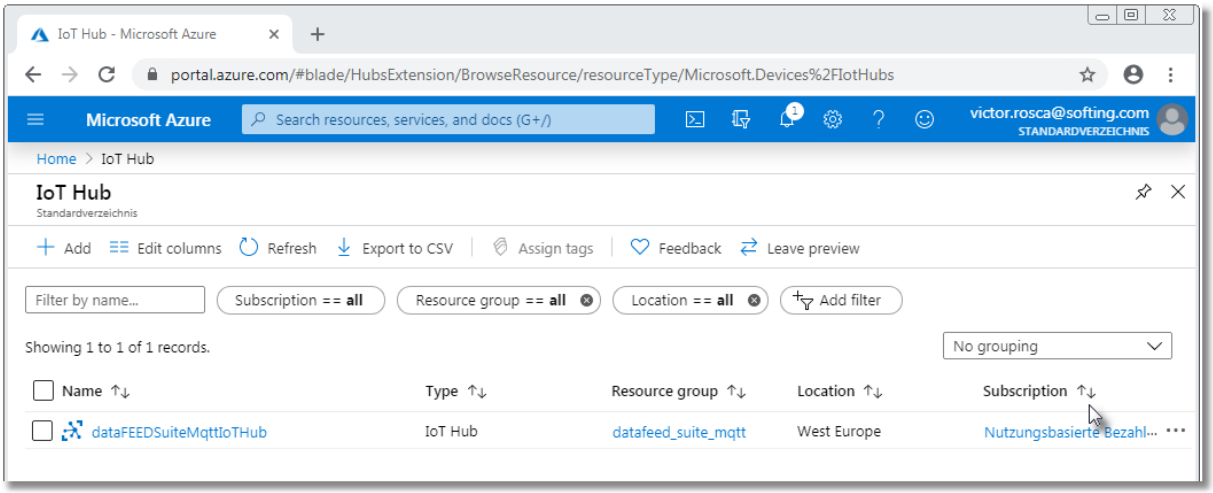
Connect *Device Explorer* to *Azure IoT Hub*

- In *Microsoft Azure Portal* search for *IoT Hub*

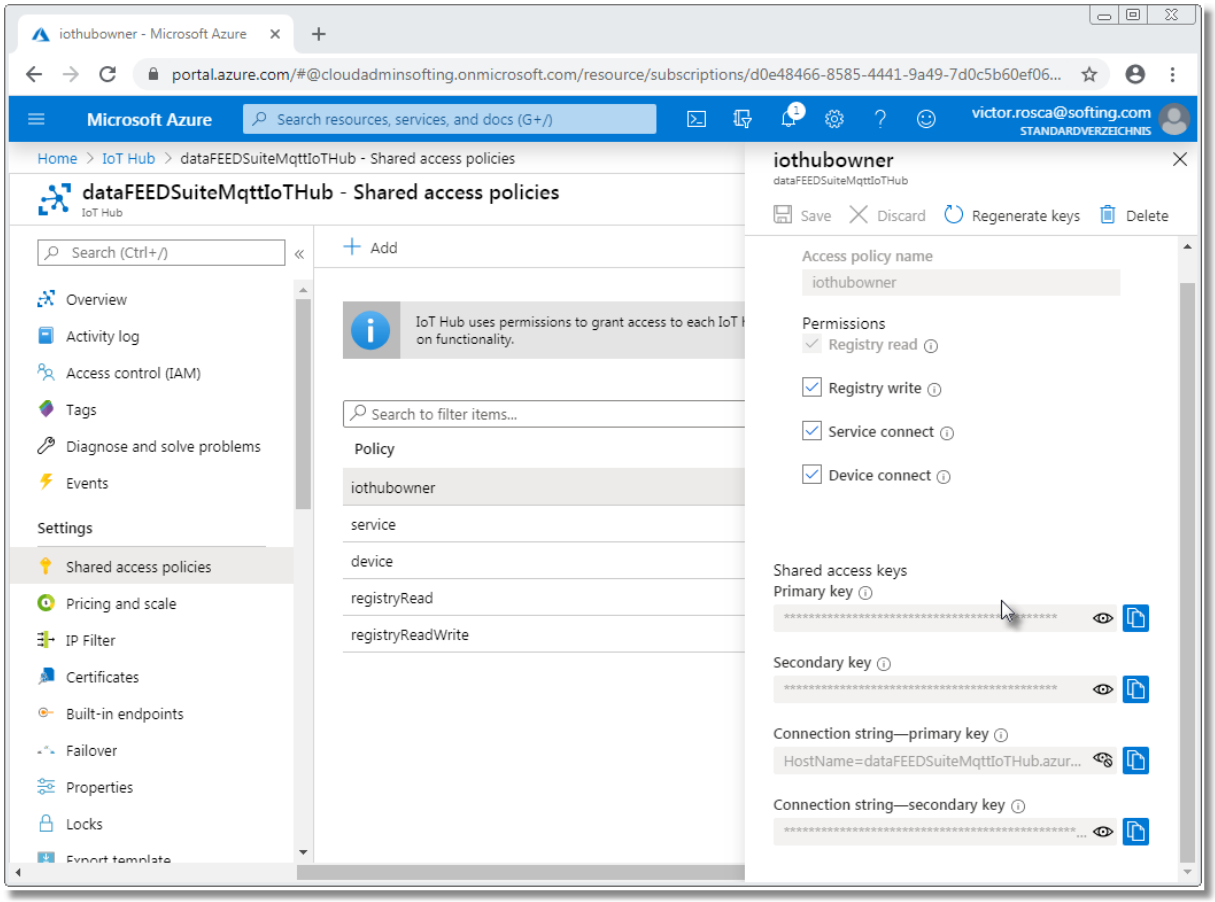


- Select *IoT Hub* to be connected or create a new one

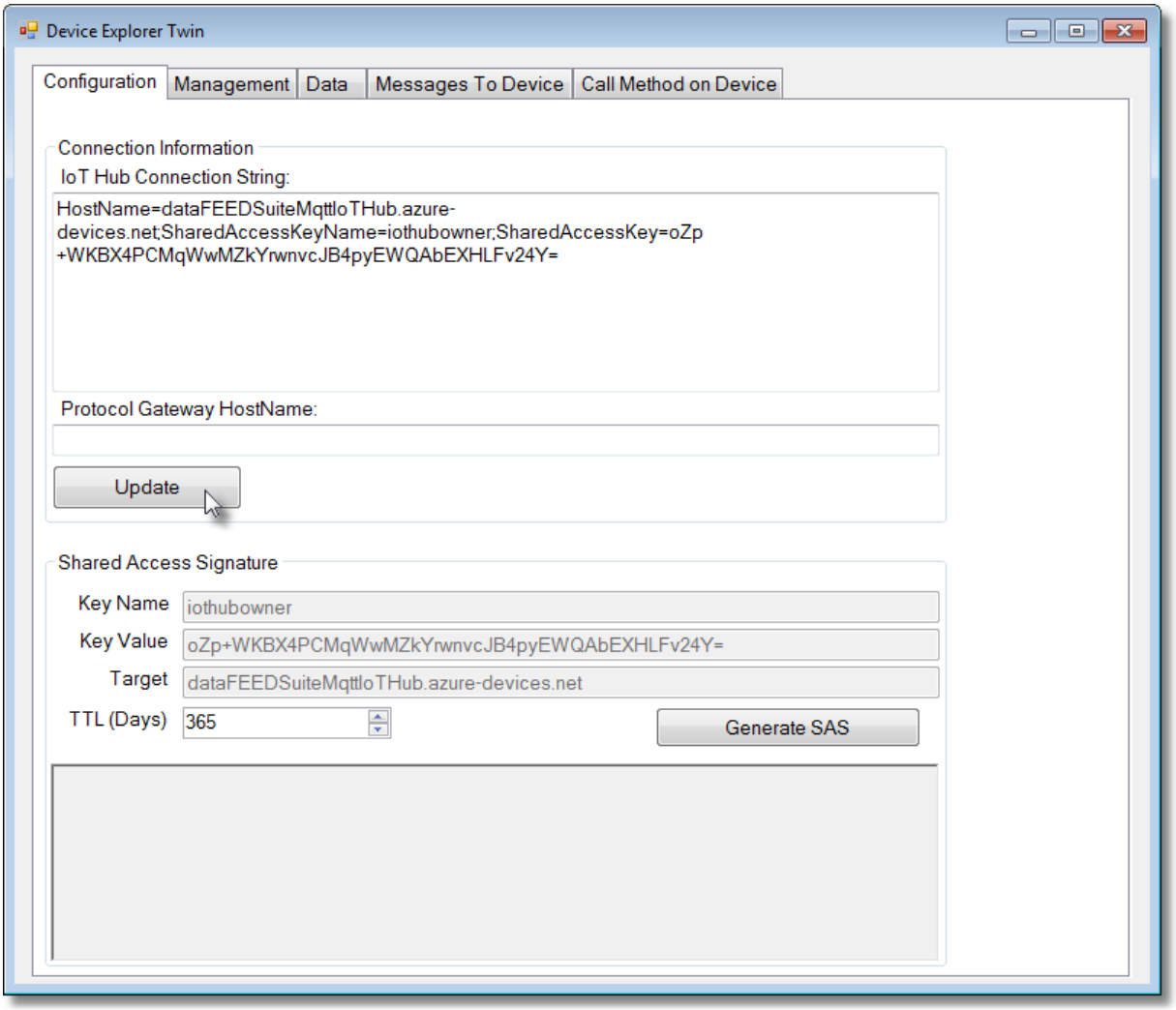
© Softing Industrial Automation GmbH



- Select *Shared access policies* Settings



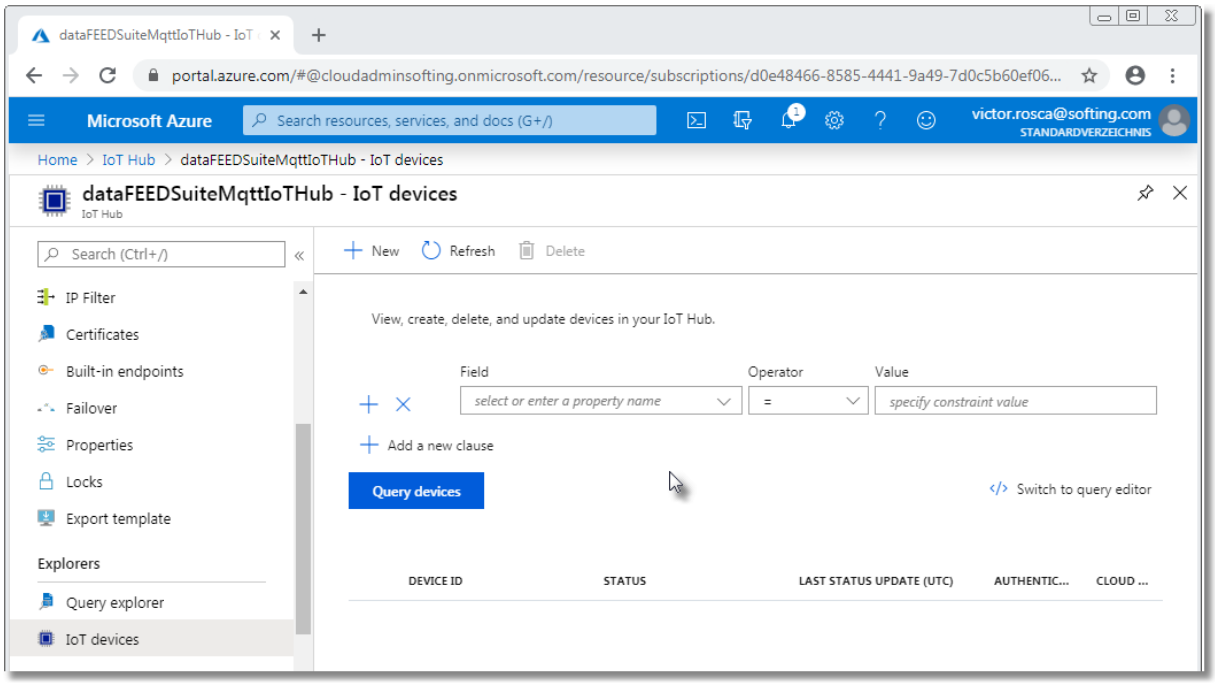
- Select ***iothubowner*** *Policy and* copy *Connection string - primary key*
- In *Device Explorer*, paste connection string to *IoT Hub Connection String* field and press *Update* button



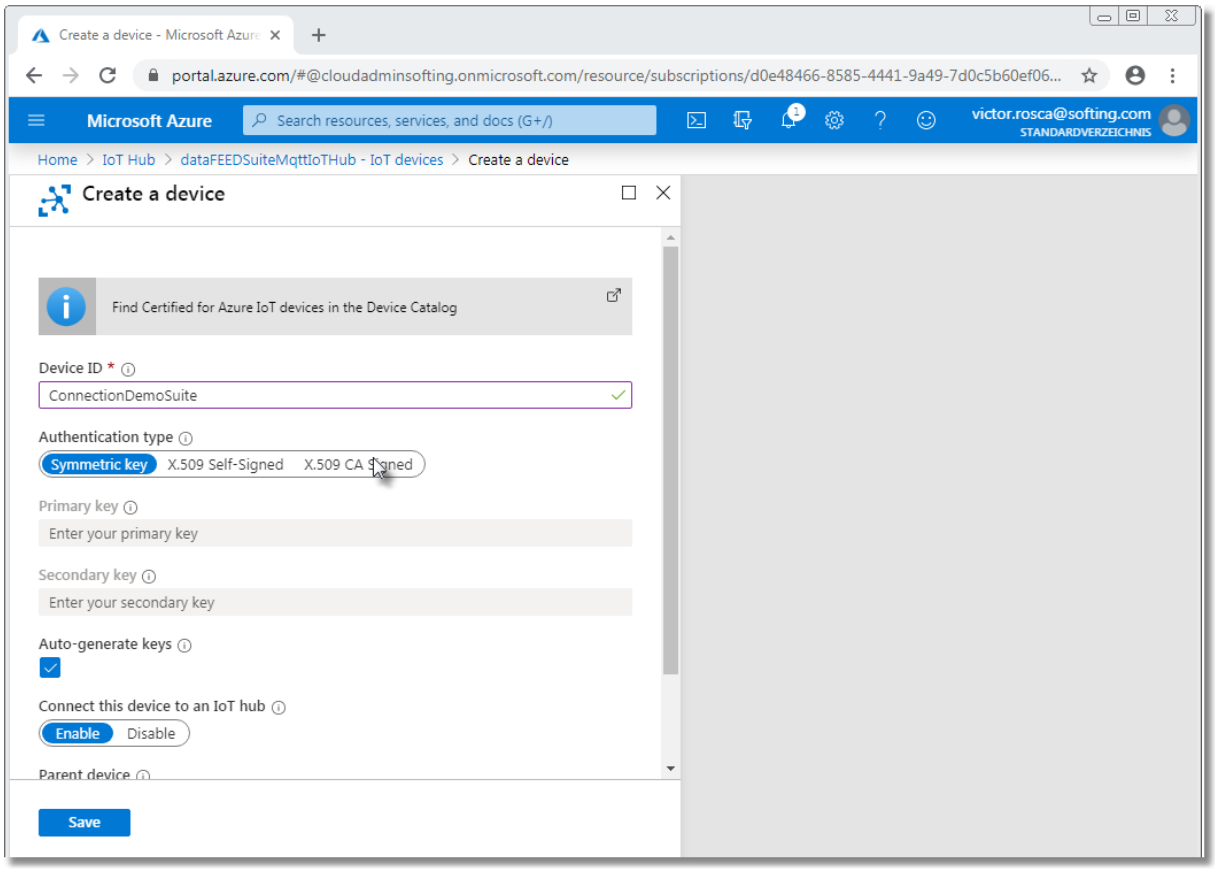
Create MQTT Connection in Azure IoT Hub

- In *Azure Iot Hub* select *IoT devices* and *Add* a new device

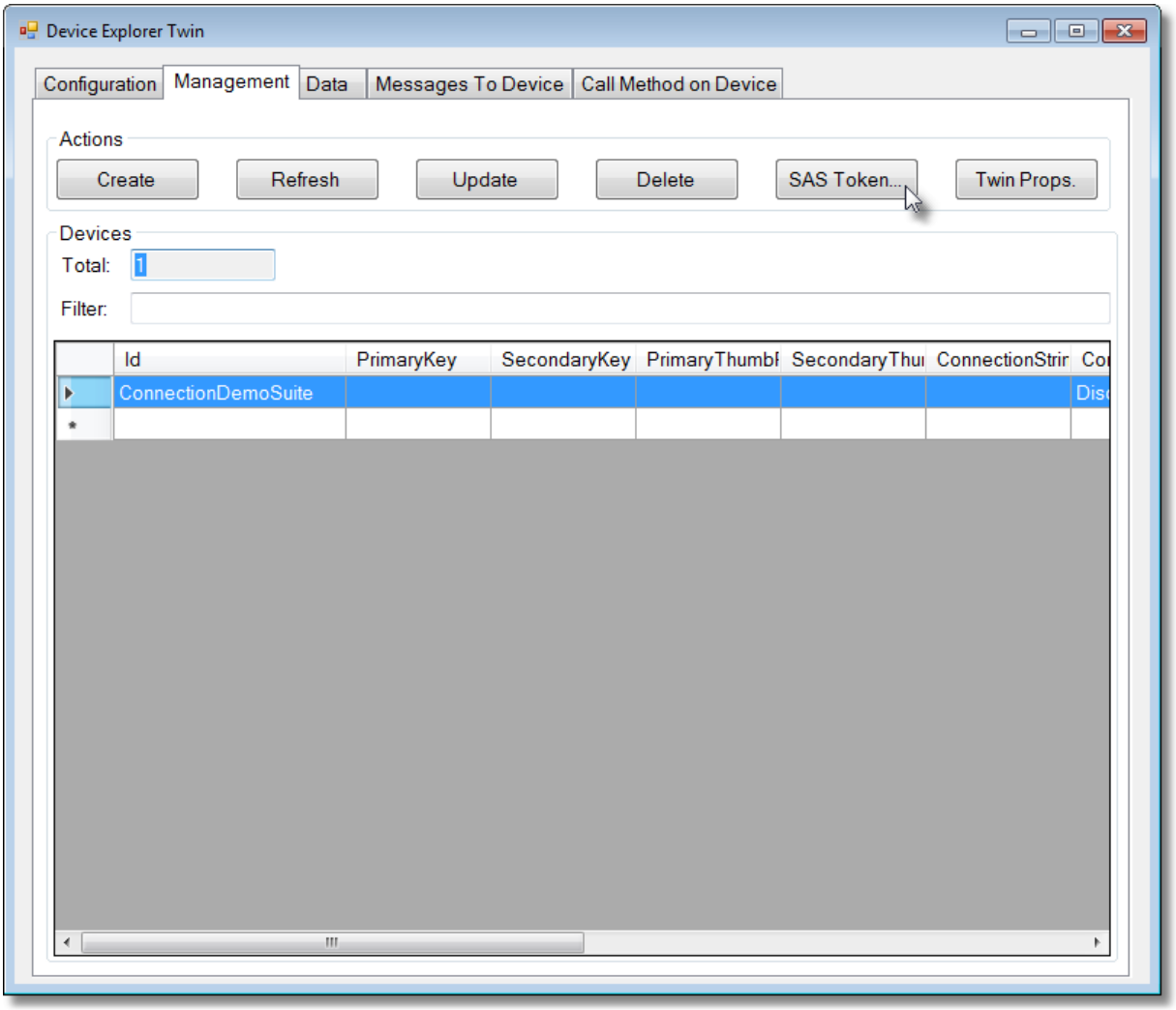
© Softing Industrial Automation GmbH



- Define unique *Device ID* (23 characters maximum)
- Do not modify default settings for rest of the fields and press *Save* button

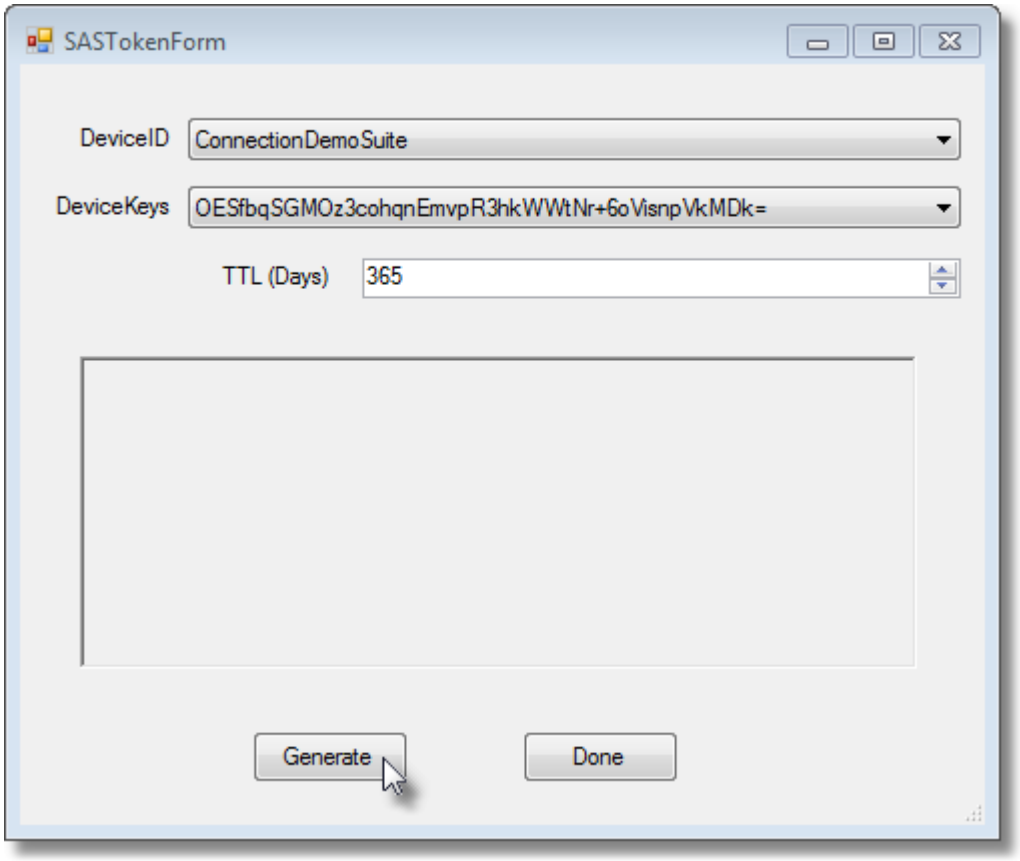


- The added device will appear in *Device Explorer*



- Select *Management* tab
- Select *IoT Hub* device
- Press *SAS Token...* button

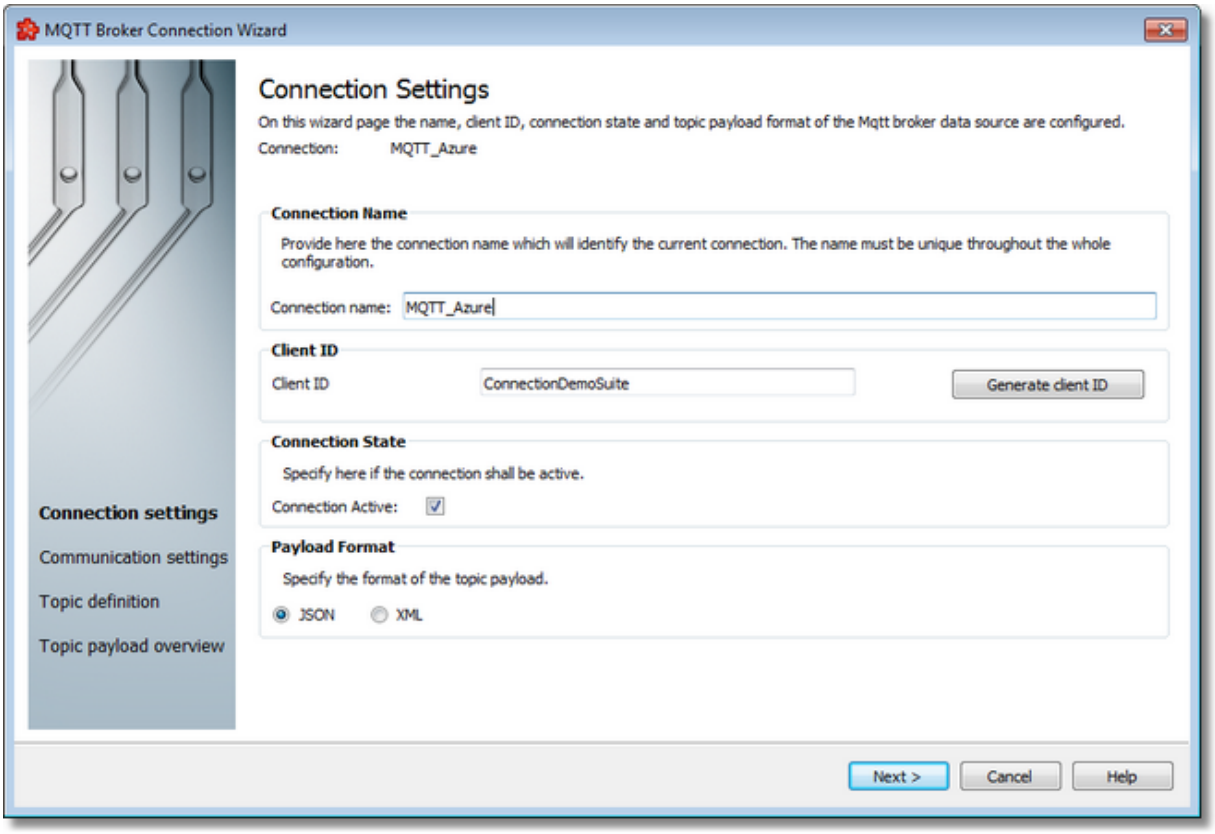
© Softing Industrial Automation GmbH



- Define SAS duration in TTL (Days) (e.g. to 365 days)
- Press Generate button
- Copy generated *SAS token* string for later use

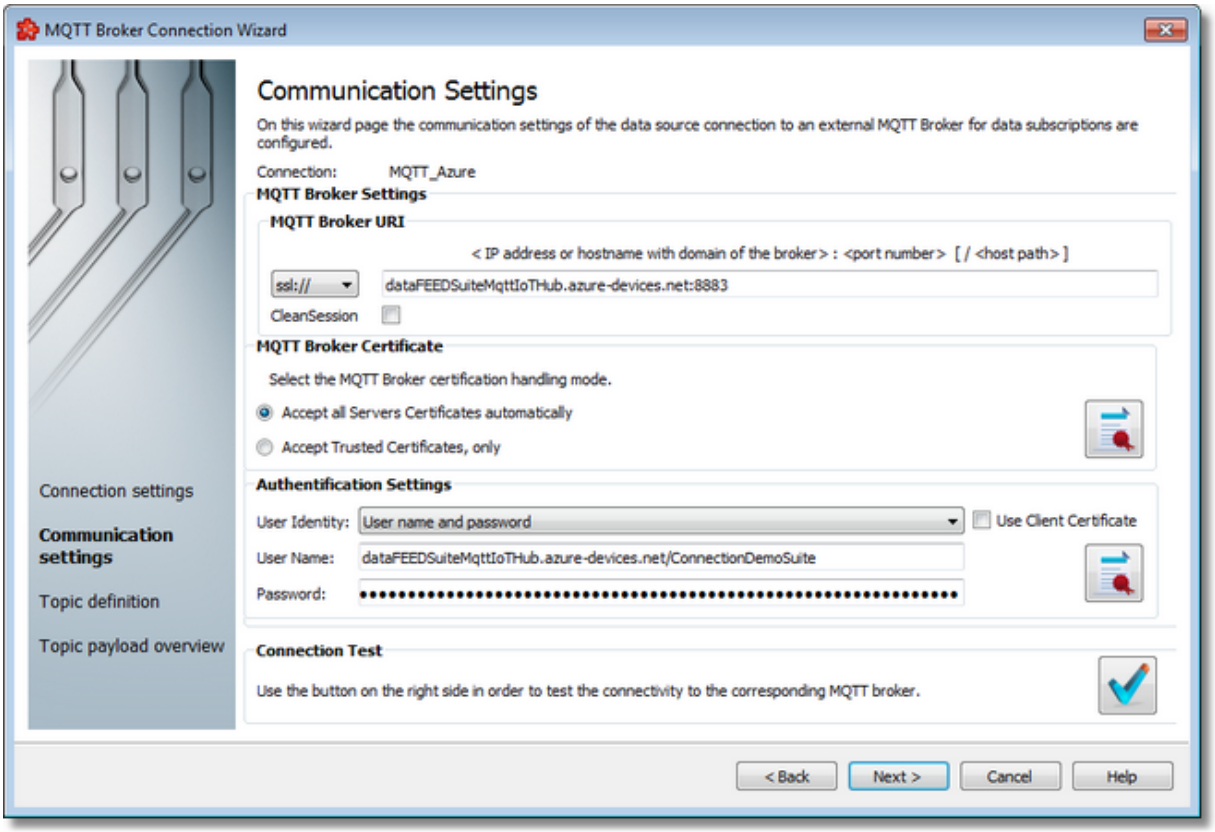
Connect *dataFEED OPC Suite* to *Azure IoT Hub*

- Start *dataFEED OPC Suite* configuration
- Select *Data Source / MQTT Broker*

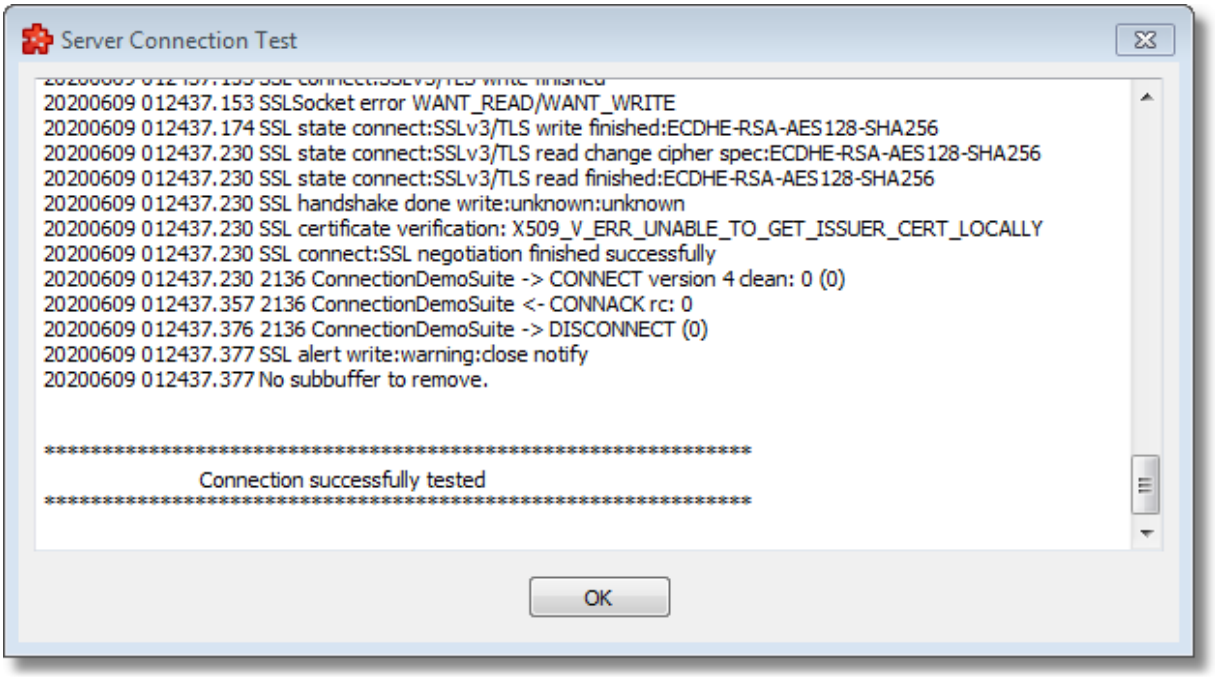


- Enter connection name in *Connection Name* field
- Enter device ID of created *Azure IoT Hub* device in *Client ID* field
- Press *Next* button

© Softing Industrial Automation GmbH



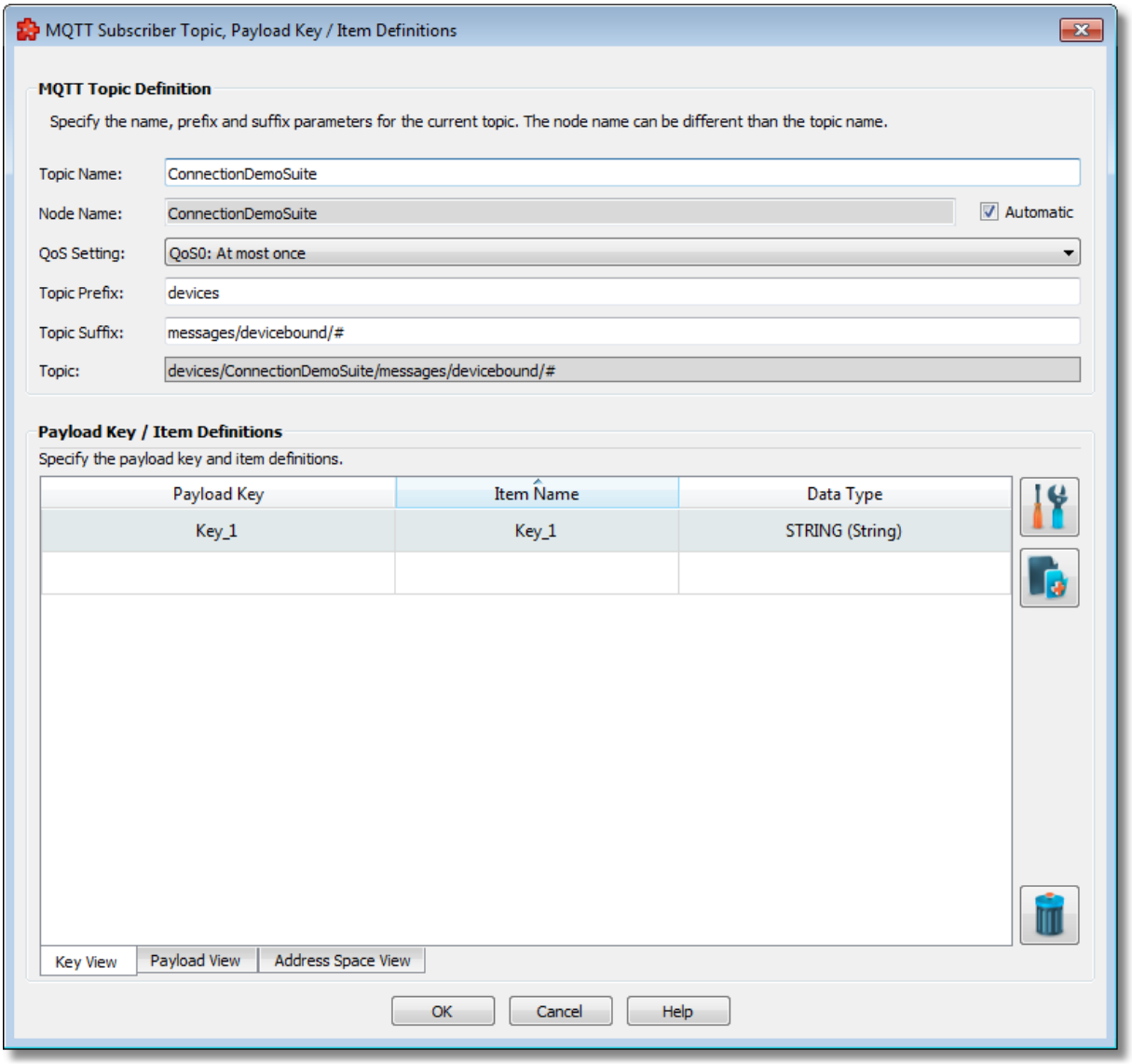
- Set *MQTT Broker URI* to *ssl://<IoT Hub Name>.azure-devices.net:8883*
- Set *User Identity* in *Authentication Settings* to *User name and password*
- Set *User Name* in *Authentication Settings* to *<IoT Hub Name>.azure-devices.net/<DeviceID*
- Set *Password* in *Authentication Settings* to part of generated *SAS Token* after *SharedAccessSignature=string* (e.g. string beginning with *SharedAccessSignature sr=Softing-IoT-Hub-SalesDemo.azure-devices.net*)
- Press Connection Test button to verify connection



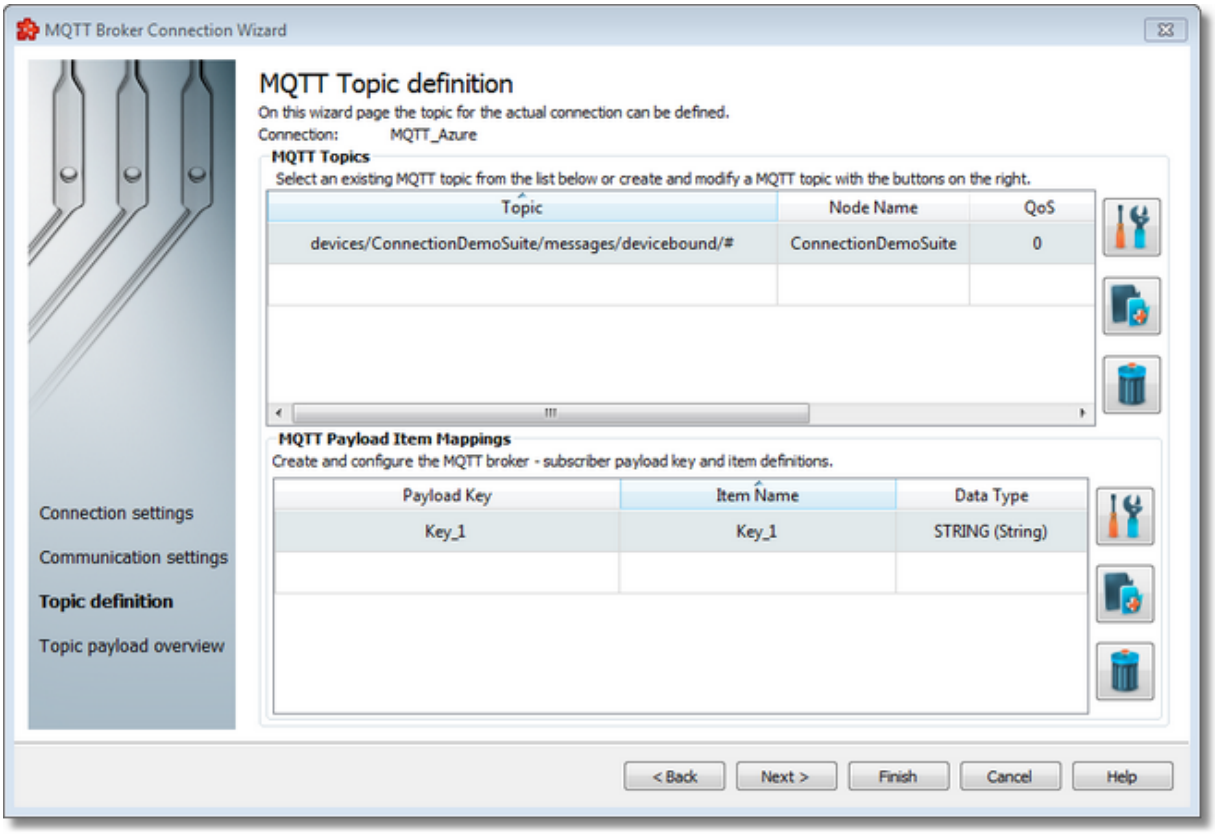
Note: In order to perform a connection via *WSS* the *MQTT Broker URI* should be: ***<IoT Hub>.azure-devices.net:443/$iothub/websocket***
E.g.: *dataFEEDSuiteMqttIoTHub.azure-devices.net:443/$iothub/websocket*

- Press *Next* button
- Add topic(s) and define the payload

© Softing Industrial Automation GmbH



- MQTT Topic definition

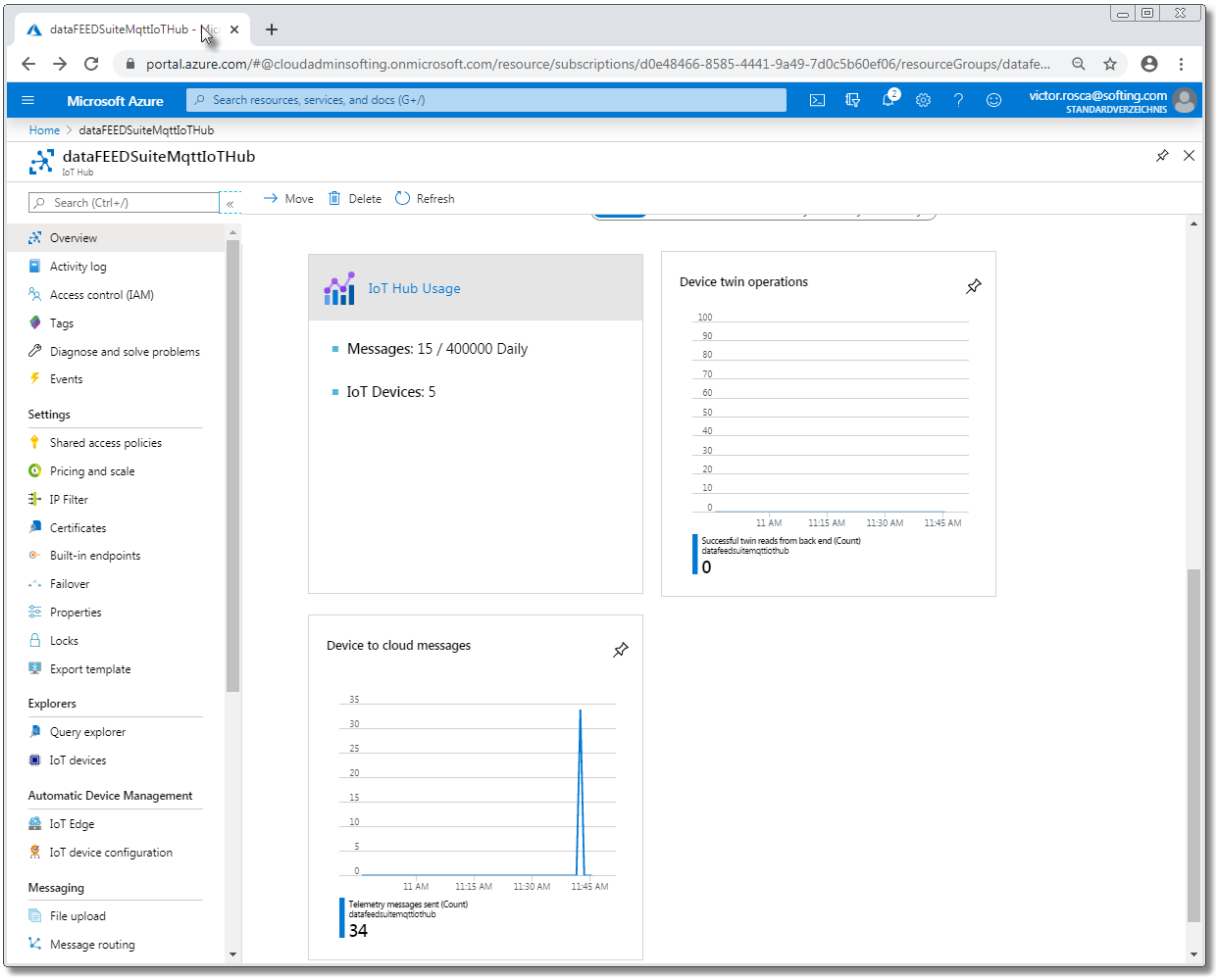


- Press Finish button

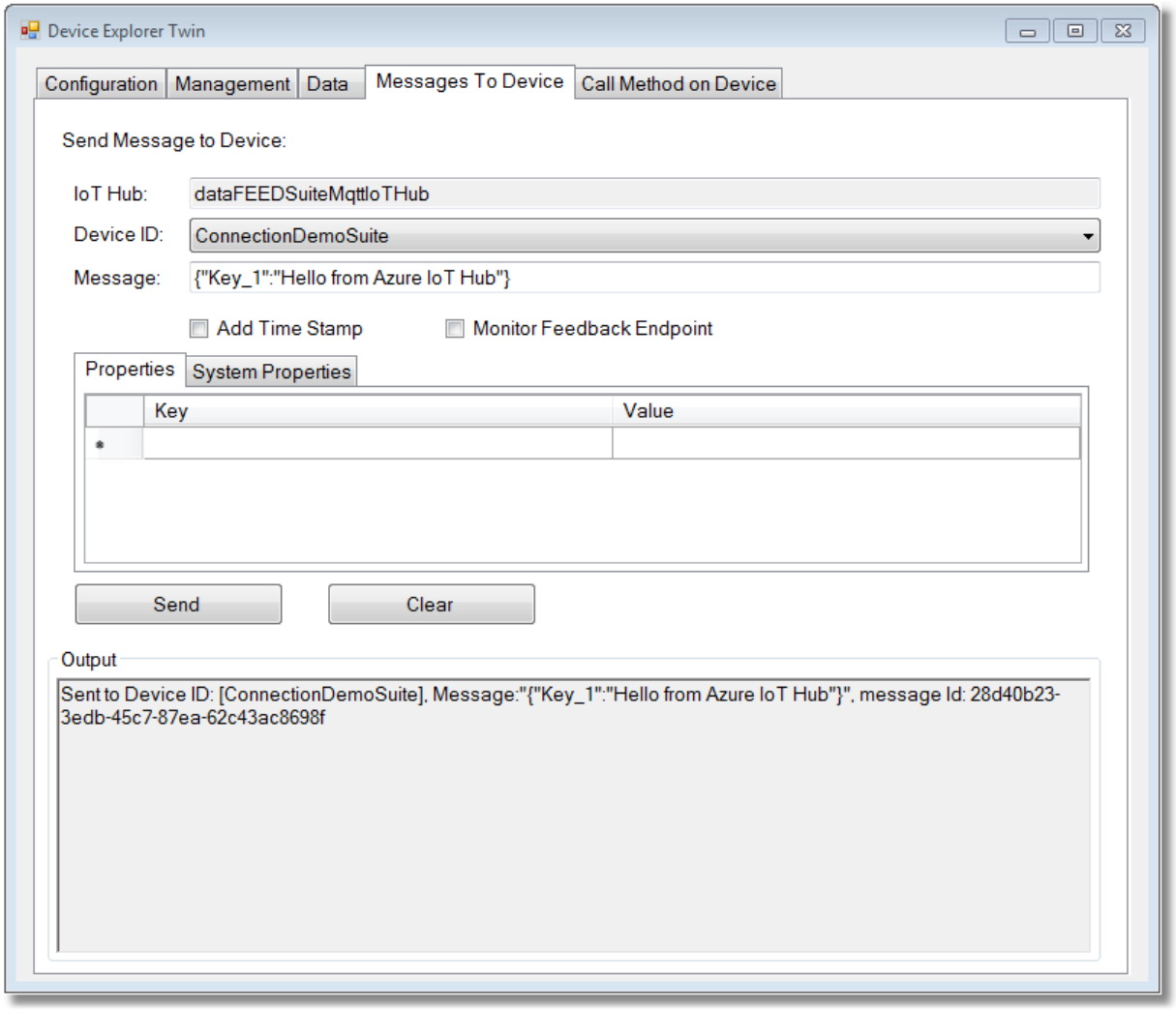
Test MQTT Connection and Data Exchange

- Start dataFEED OPC Suite application
- MQTT messages then are sent to Azure IoT Hub
- Select Overview page

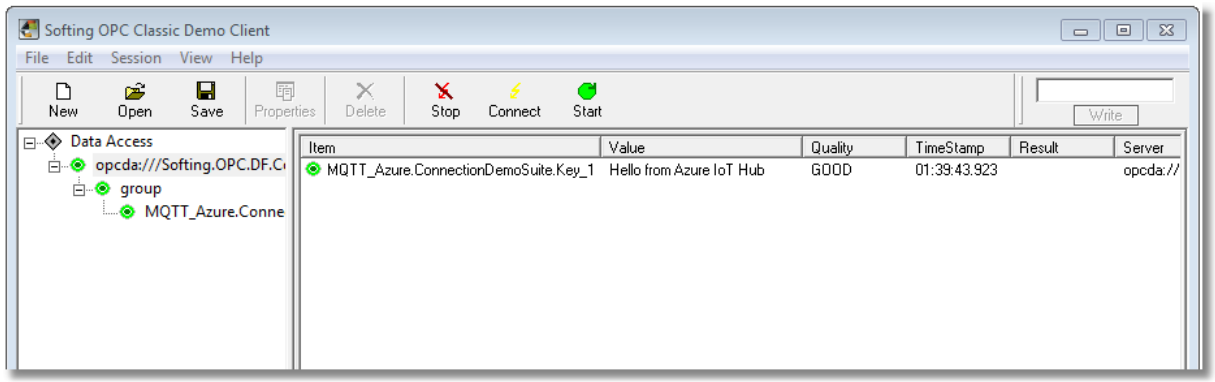
© Softing Industrial Automation GmbH



- In Device Explorer user can monitor the received data after connecting to *IoT Hub*



- The publish request will be received in dataFEED OPC Suite



7.3.1.11.7Connect to AWS IoT Device via SSL and WSS

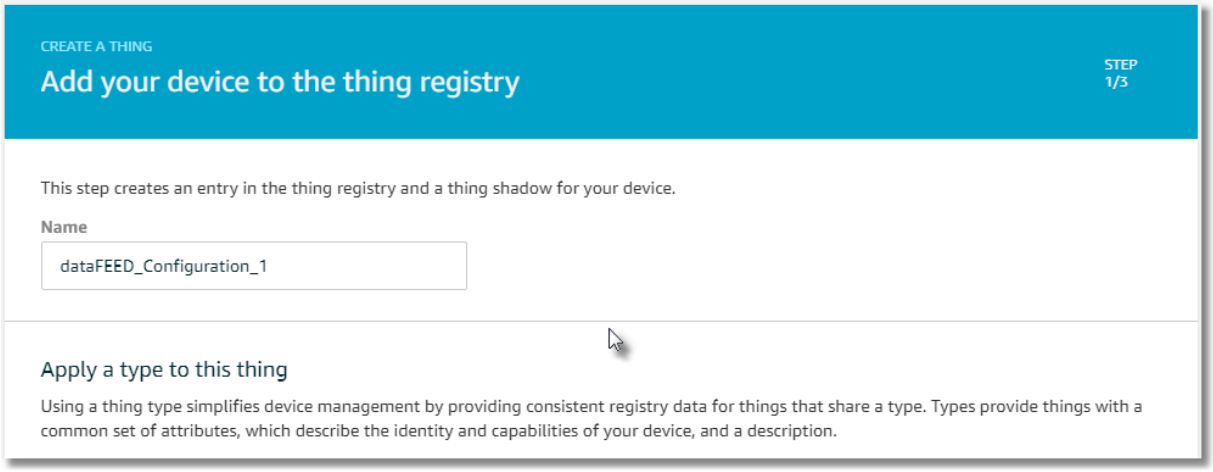
The dataFEED OPC Suite application supports connections with *AWS IoT Device*. This topic describes how to configure dataFEED OPC Suite as *AWS IoT* device (*"Thing"*).

From the *AWS IoT* point of view dataFEED OPC Suite is referred to in this manual as *gateway*. It is based on the *AWS Developer Guide*, from: https://docs.aws.amazon.com/iot/latest/developerguide/what-is-aws-iot.html.

© Softing Industrial Automation GmbH

AWS Preparation Steps

- Register Gateway as Device in the Registry. Follow the instructions given at: https://docs.aws.amazon.com/iot/latest/developerguide/register-device.html.
- In step 4. define unique gateway name



Create and Activate Certificate for Gateway

- Follow the instructions given at https://docs.aws.amazon.com/iot/latest/developerguide/create-device-certificate.html.
- In step 2. download certificate and key files. These files need to be uploaded as *MQTT Client* certificate into the gateway in a later step.

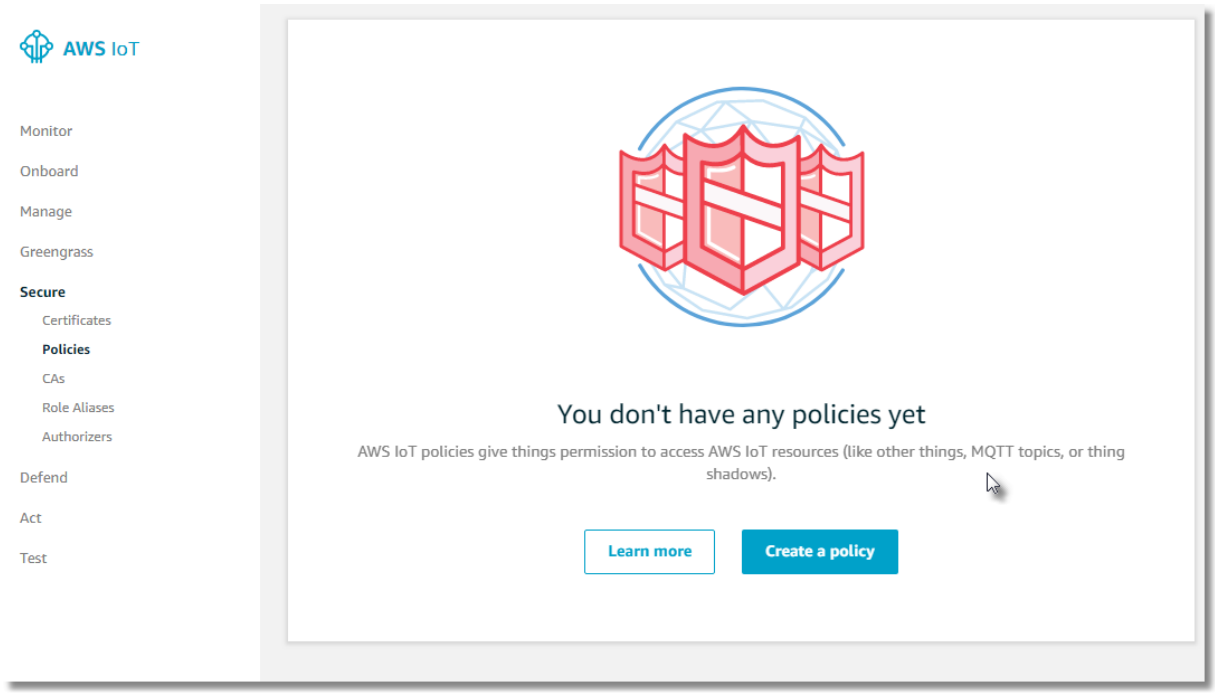
Create AWS IoT Policy for MQTT Client Certificate

- Follow the instructions given at https://docs.aws.amazon.com/iot/latest/developerguide/create-iot-policy.html.
- Define gateway permission for ***iot:Connect*** and ***iot:Publish*** actions.

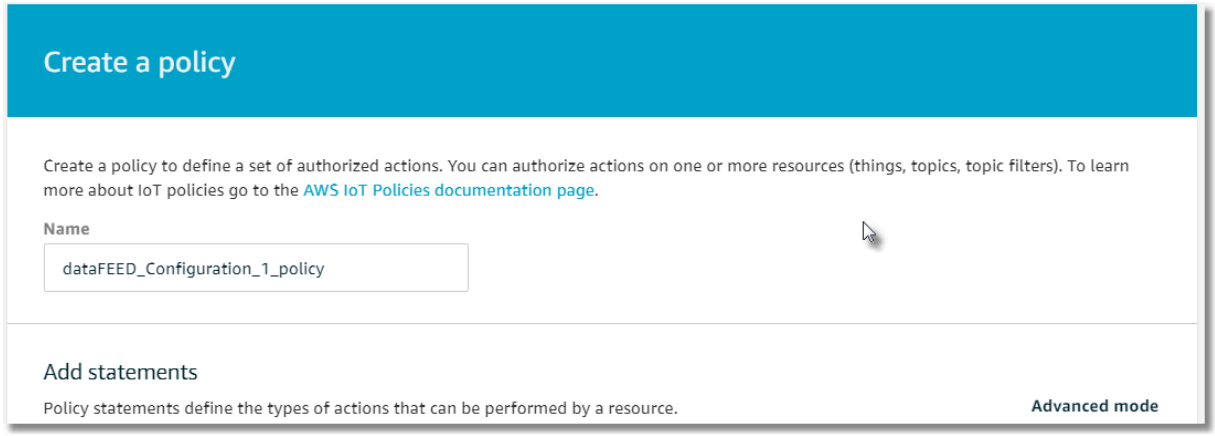
NOTE: For some unknown reasons the attachment of policies as described in step 3. does not work. Instead, it looks like the policy must be created before attaching it to a certificate. However, the certificate dialog does not trigger the creation of a policy.

It is possible to create and attach a policy after the creation of a certificate by following these steps:

- Return to *AWS IoT Console* main screen and navigate to *Secure/Policies*



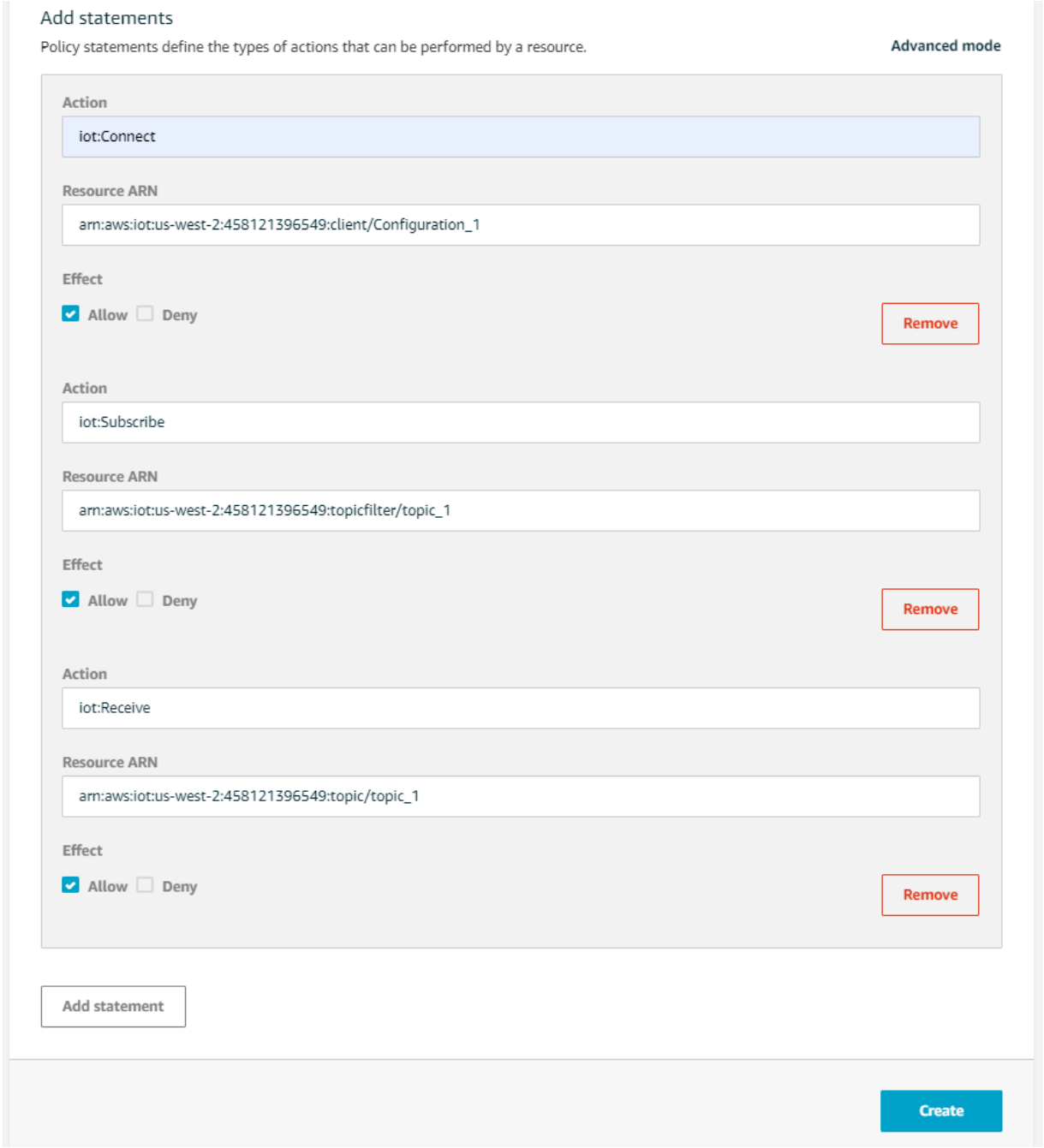
- Press *Create* a policy button
- Define policy name



- Define *gateway* permission for ***iot:Connect*** action
- Press *Add Statement* button
- Define *gateway* permission for ***iot:Subscribe*** action
- Press *Add Statement* button
- Define *gateway* permission for ***iot:Receive*** action
- Define user-specific *ARN* topic. Remember this topic for a later step.

NOTE: The placeholder "*" allows to subscribe to any topic.

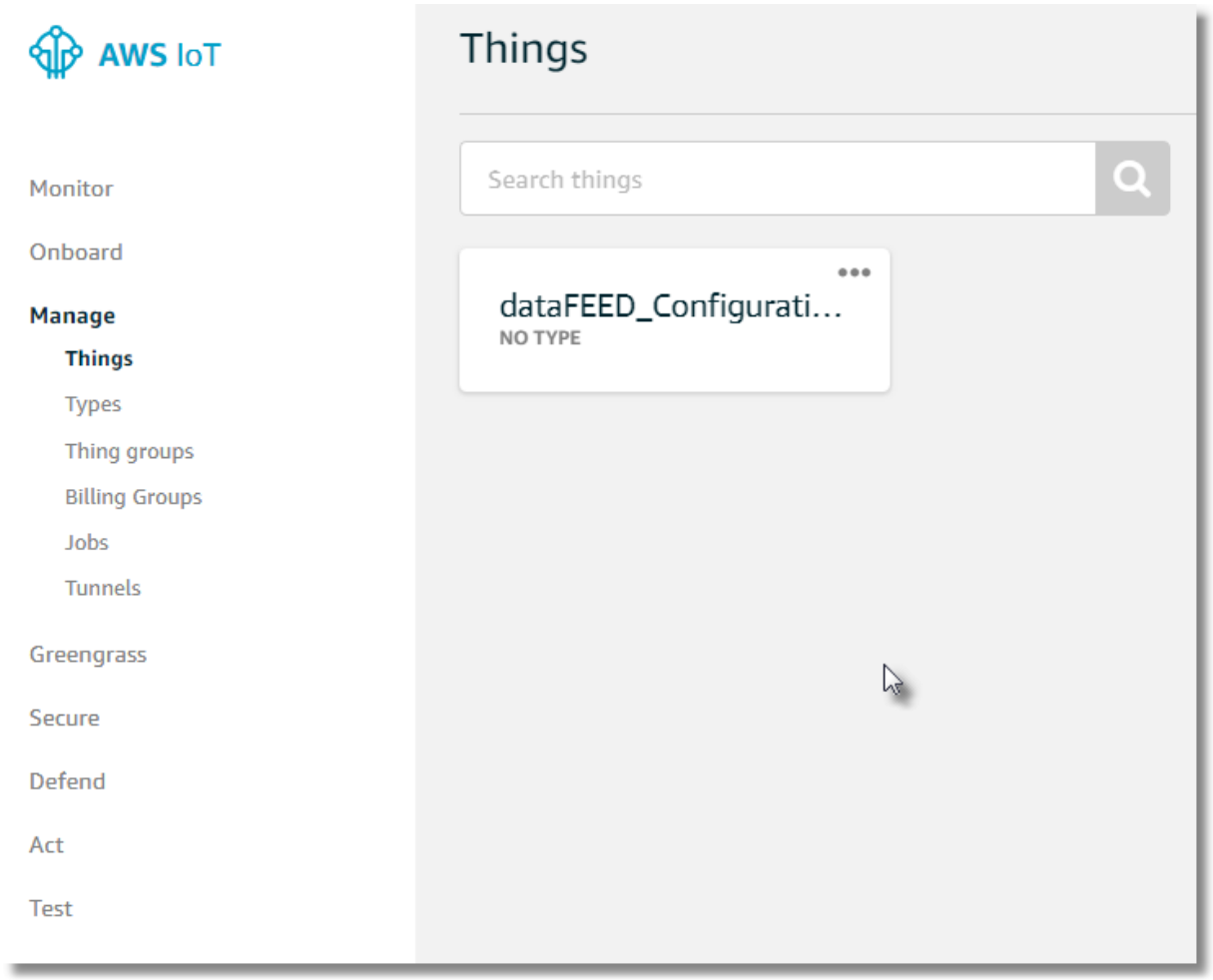
© Softing Industrial Automation GmbH



- Press *Create* button

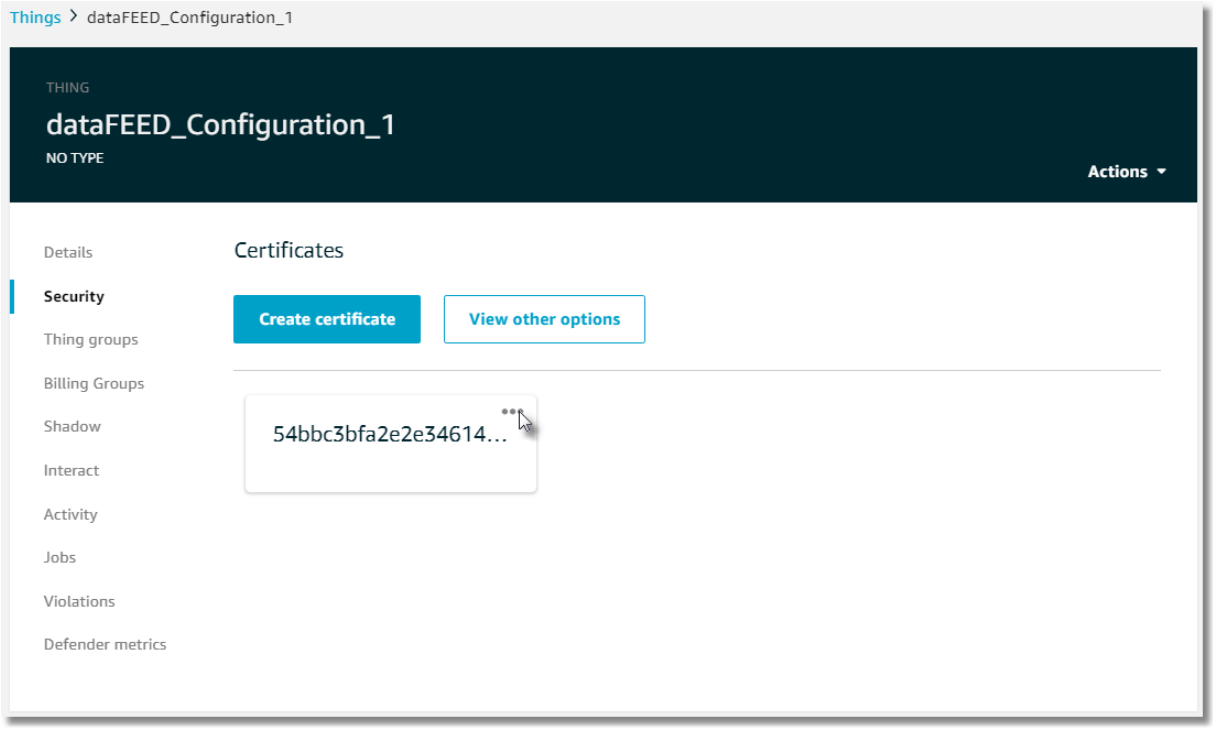
Attach *AWS IoT* Policy to Gateway

- Return to *AWS IoT Console* main screen and navigate to *Manage/Things* page
- Click on device name of gateway

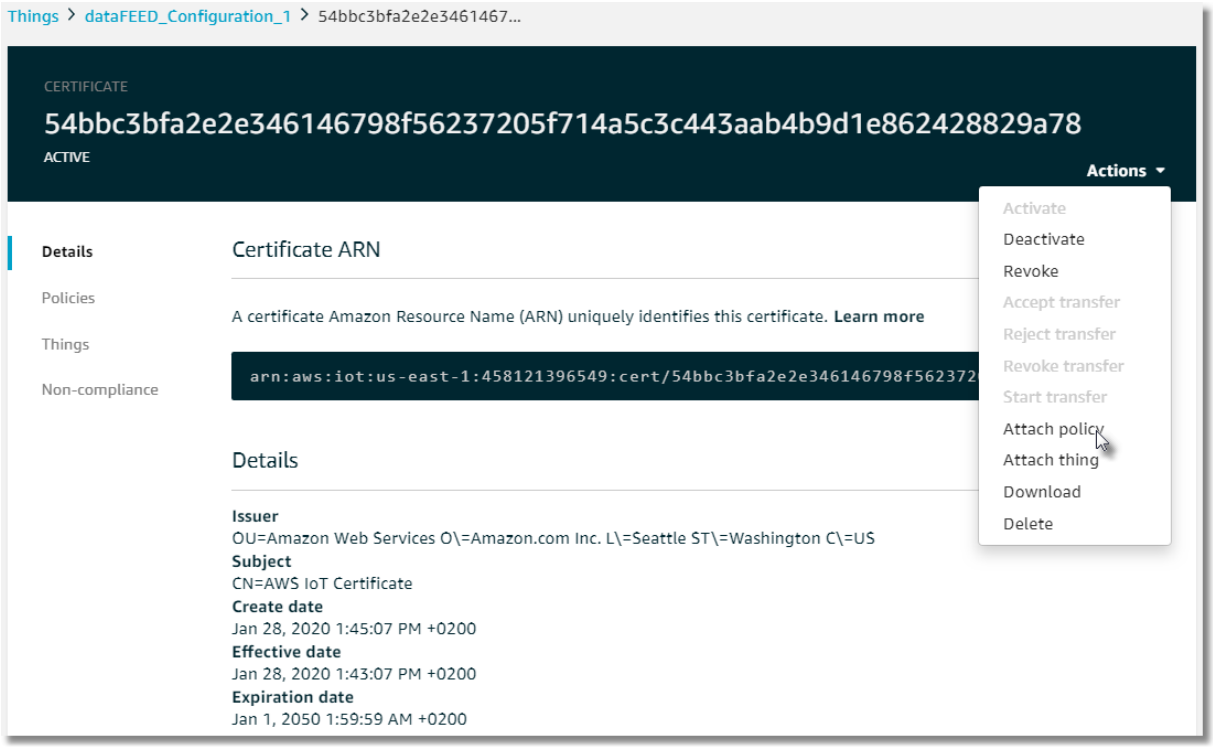


- Click on *Security*
- Click on hash value of created device certificate

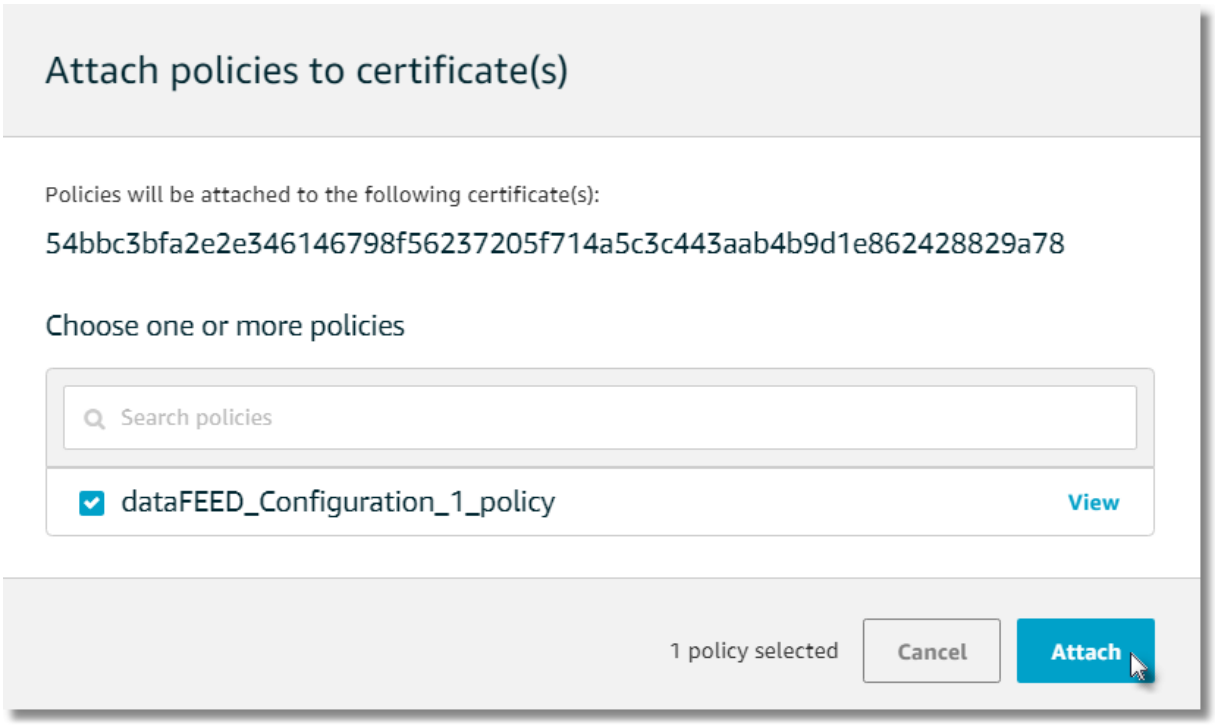
© Softing Industrial Automation GmbH



- Select *Attach policy* in *Actions* menu



- Attach created policy to certificate



Determine AWS MQTT Broker Address

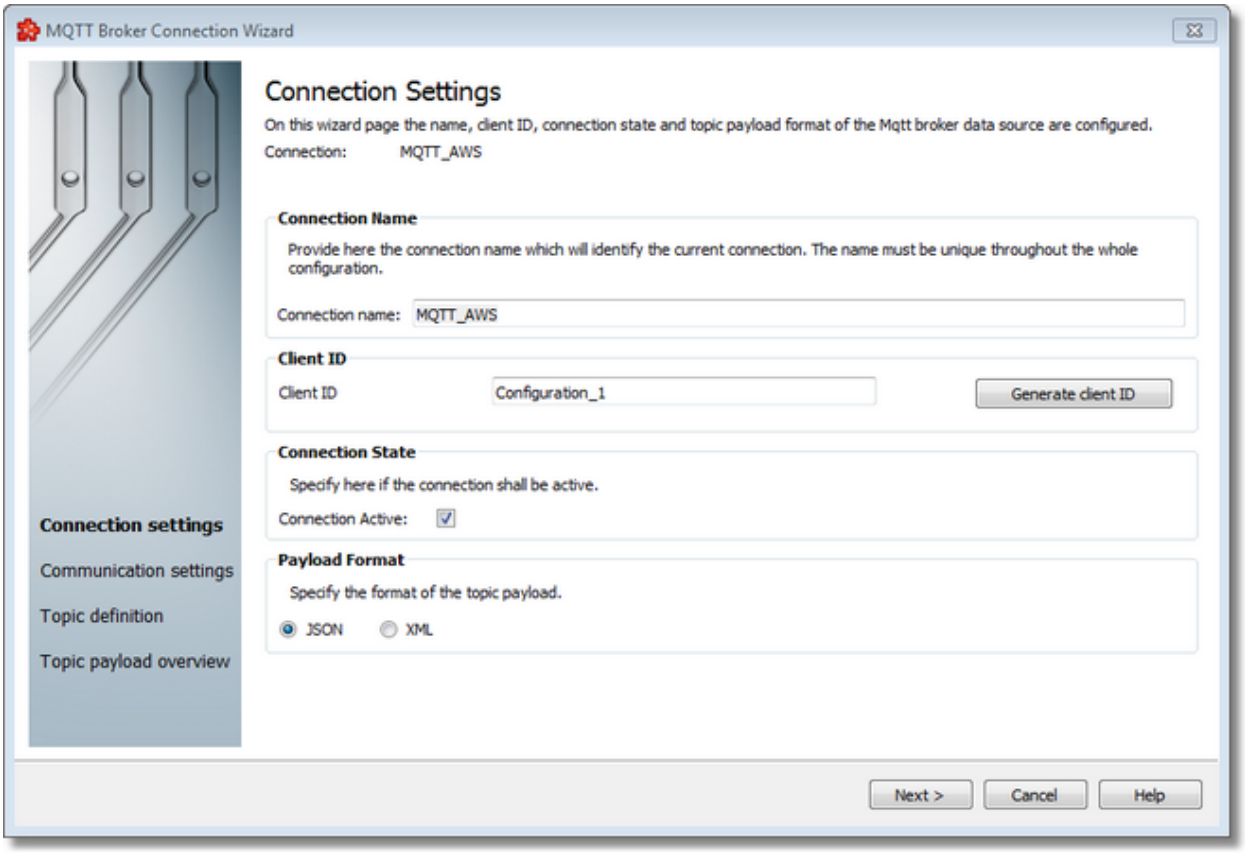
- Return to *AWS IoT Console* main screen and navigate to *Settings*
- *MQTT Broker* address can be found in Custom Endpoint field

NOTE: The address looks similar to "a23px2f5602wdn-ats.iot.us-west-2.amazonaws.com"

Define MQTT Broker Connection

- Start dataFEED OPC Suite Configurator
- Navigate to *Data Source/MQTT Broker* page
- Add a new data destination (*Ctrl+A*)

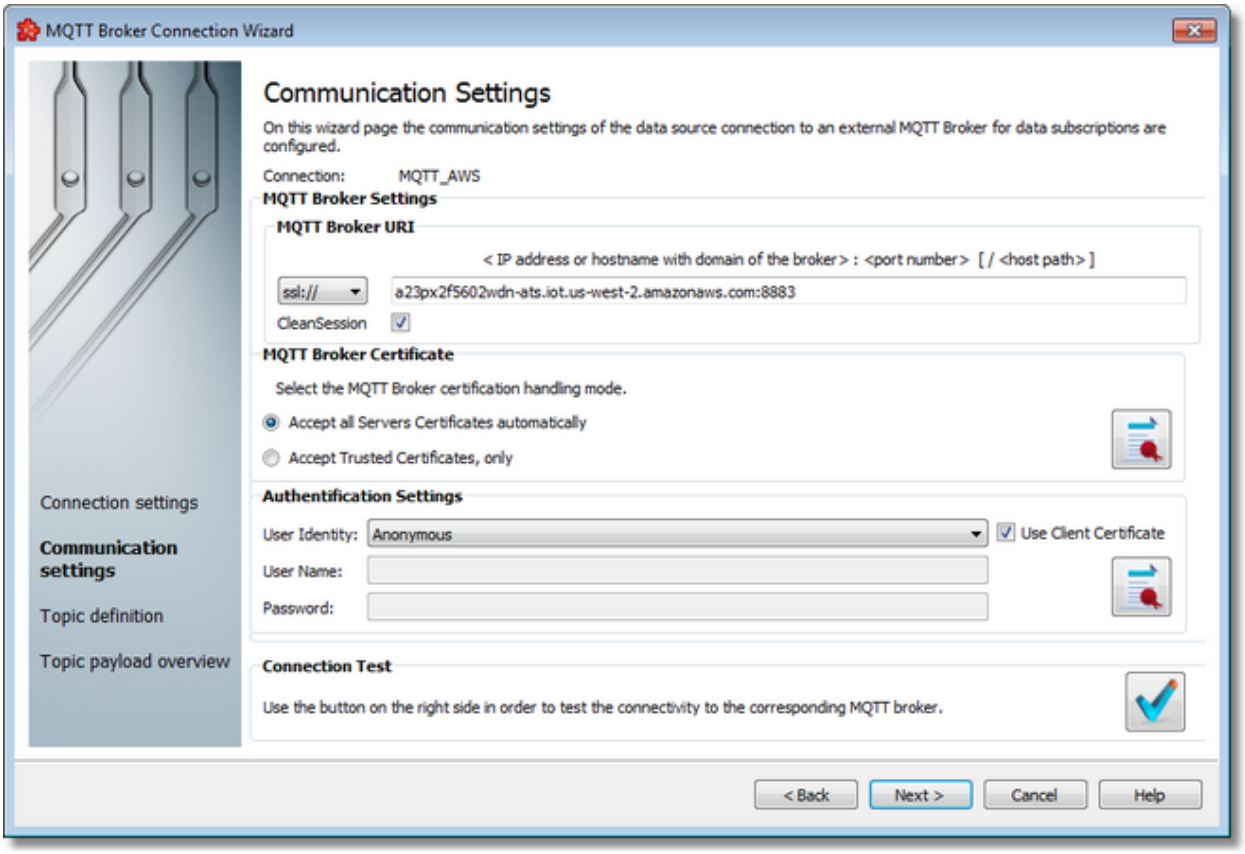
© Softing Industrial Automation GmbH



- Define *MQTT Broker Connection Name* and unique *Client ID*

NOTE: *Device ID* has to be allowed in ***iot:Connect*** policy (see section *AWS Preparation Steps*)

- Navigate to *Communication Settings* page (page 2) of *MQTT Broker Connection Wizard*



- Set *MQTT Broker URI* to *ssl://*
- Enter *AWS MQTT* broker address as prefix in *< IP address or hostname with domain of the broker > [: <port number>]* field
- Enter port number *"8883"* as postfix in *< IP address or hostname with domain of the broker > [: <port number>]* field
- Enable *MQTT Clean Session* flag

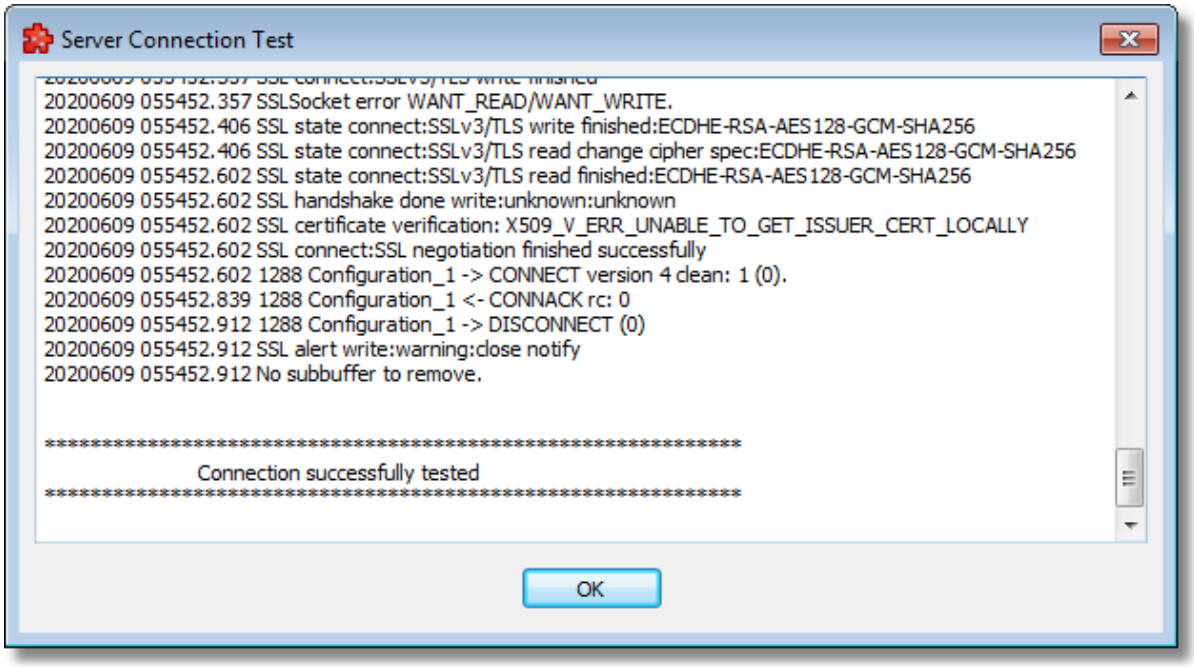
NOTE: *AWS Broker* disconnects, if flag is not set (see https://docs.aws.amazon.com/iot/latest/developerguide/protocols.html)

- Set Authentication Settings/User Identity to Anonymous
- Activate *Use Client Certificate* checkbox
- Select the downloaded certificate "**-certificate.pem"* and private key "**-private.pem.key*"

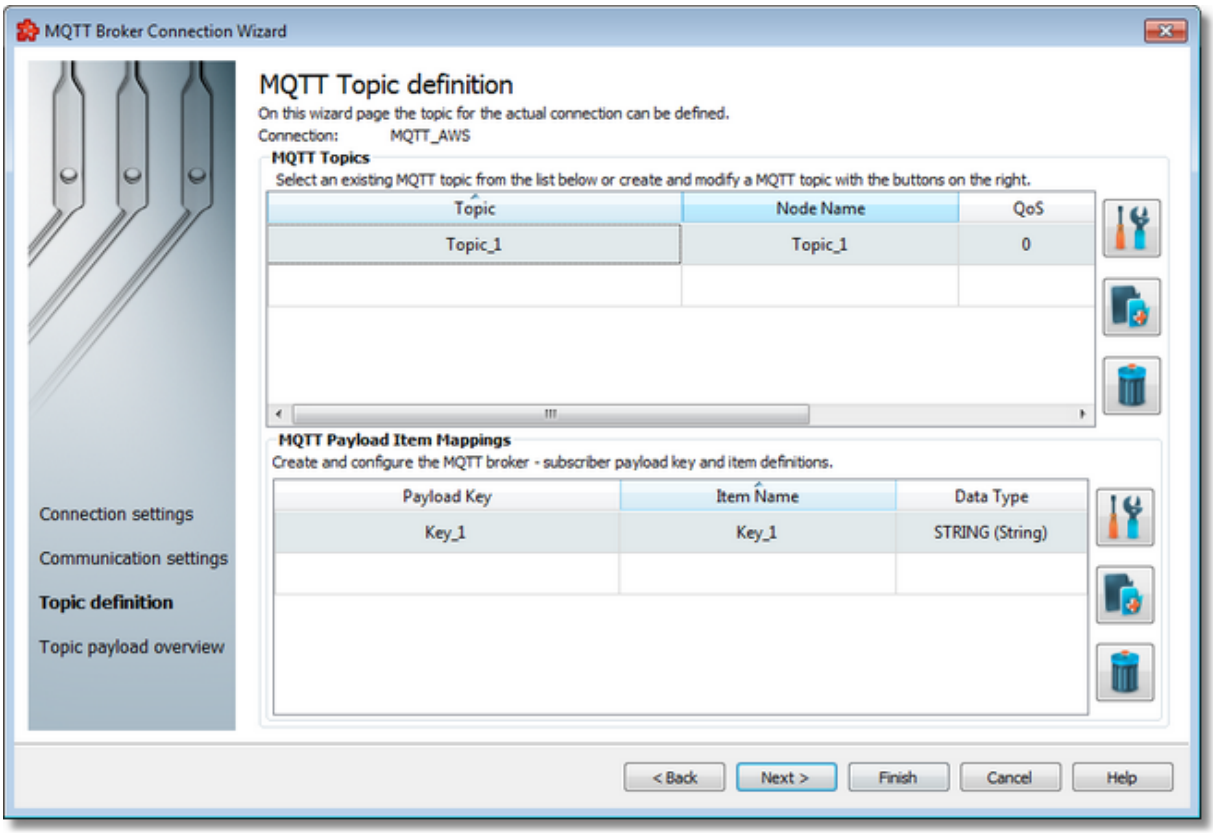
NOTE: Before the private key can be selected the certificate file extension has to be changed to ".pem"

- Press Connection Test button to verify connection

© Softing Industrial Automation GmbH



- Navigate to *MQTT Topic* definition page (page 3) of *MQTT Broker Connection Wizard*



- Add topics

NOTE: Topics have to be allowed in ***iot:Subscribe*** and ***iot:Receive*** policies (see section *AWS Preparation Steps*)

- Press *Save* button

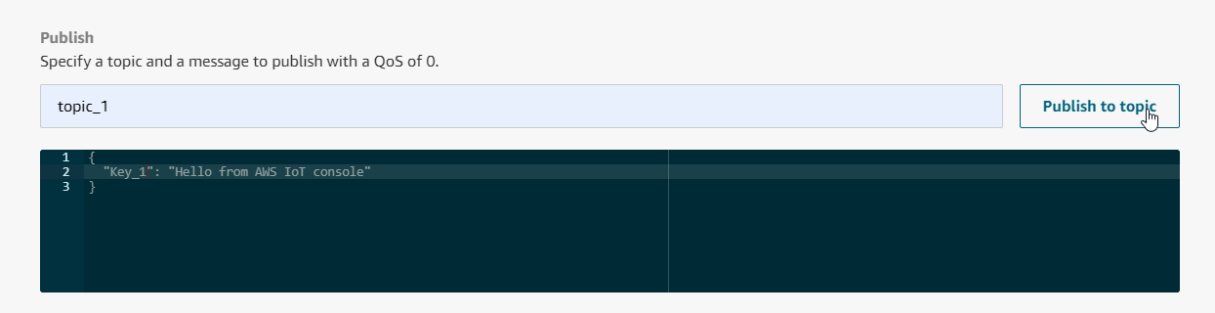
Finish dataFEED OPC Suite Configuration

- dataFEED OPC Suite configuration is finished
- Save configuration and start dataFEED OPC Suite

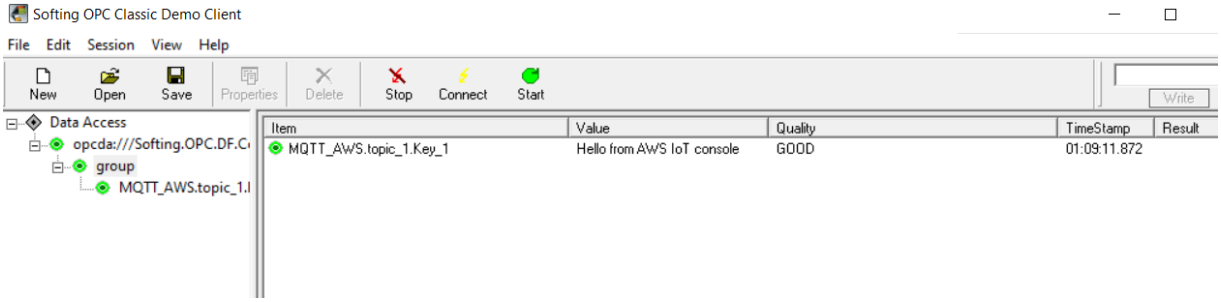
Monitor Values in AWS

- Return to *AWS IoT Console* main screen and navigate to *Test*
- Publish the topic "topic_1" with payload:

```
{
  "Key_1": "Hello from AWS IoT console"
}
```



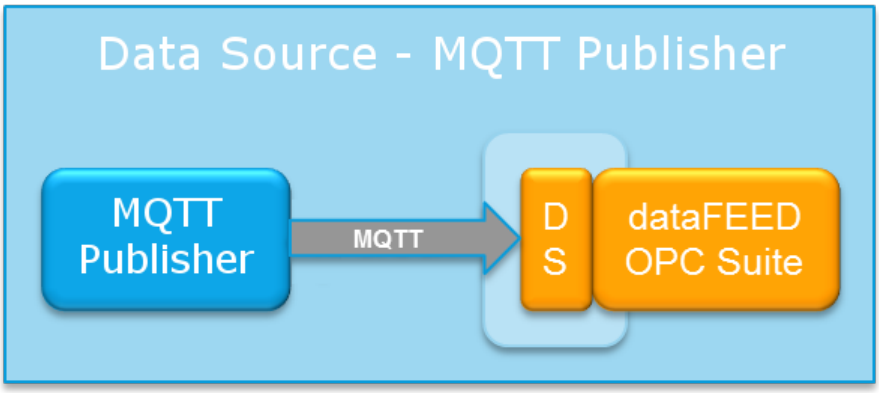
- The publish request will be received in dataFEED OPC Suite



7.3.1.12 MQTT Publisher

This section of the Softing dataFEED OPC Suite Online Help describes how to configure *MQTT Publisher* connections.

© Softing Industrial Automation GmbH

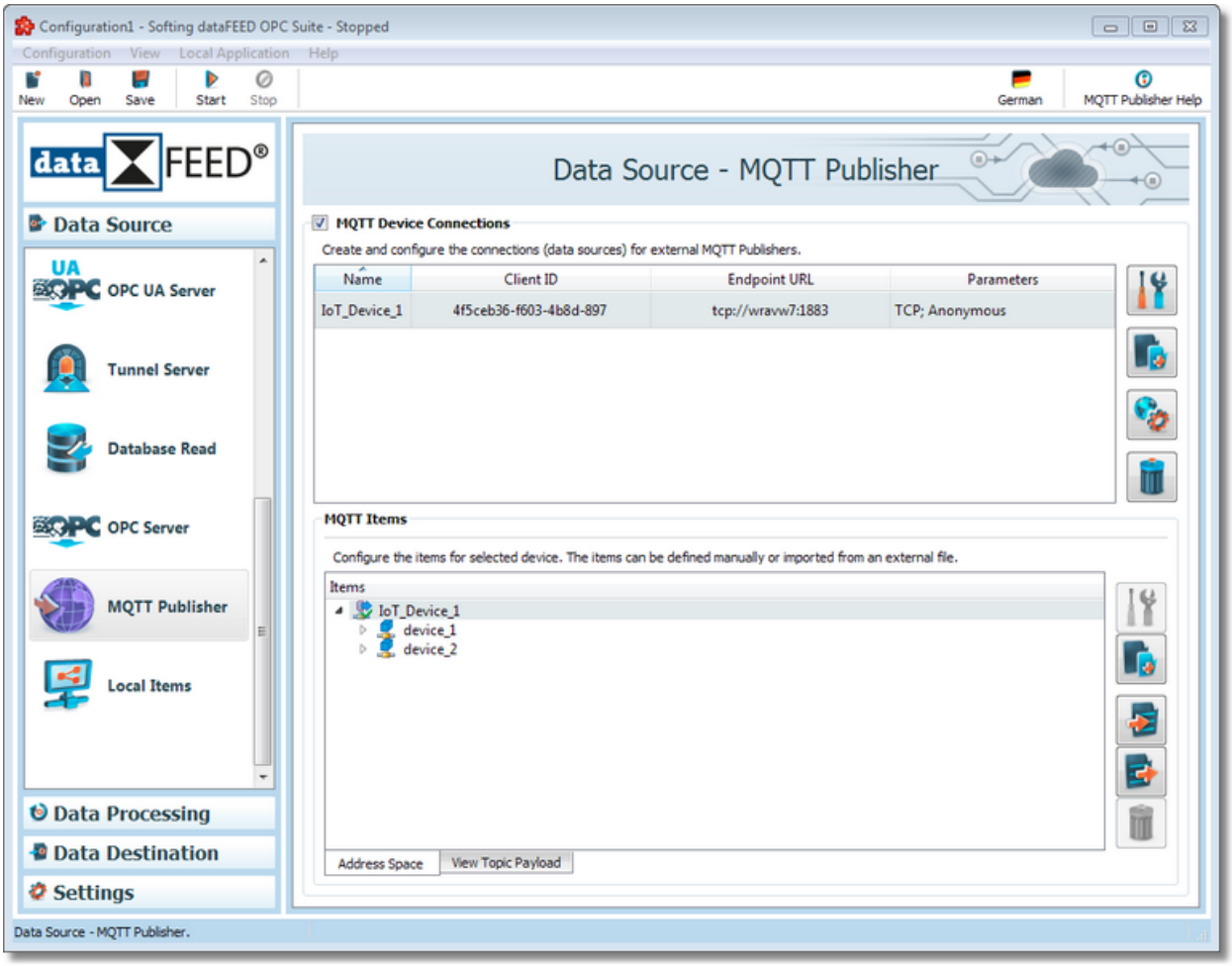


MQTT Publisher data source makes it possible to receive data from a *MQTT Publisher* client. The received data can be then processed, saved, analyzed or transferred by using the *Data Processing* and/or *Data Destination* functionalities.

A detailed description on how to configure access to *MQTT Publisher* connections can be found in the Configuration section.

7.3.1.12.1Configuration

In order to configure connections to *MQTT Publisher* select *Data Source* in the left window part and then select the *MQTT Publisher* icon. The *MQTT* connection overview is opened.



When the *MQTT Publisher* functionality is enabled using the MQTT Device Connections checkbox, the user is provided with the possibility to customize the protocol settings in the *General Properties* section, prior to creating any connections.

Add an MQTT connection

You can establish a connection to an *MQTT Publisher* by using the button *Add*  or the menu command *Add* in the context menu. The MQTT Publisher Connection Wizard is opened.

Modify an existing MQTT connection

The configuration of an existing connection to an *MQTT Publisher* can be modified. To do so, select the line in the connection overview that contains the connection to be modified. Then you can use the button *Properties*  or the menu command *Properties* from the context menu to open the MQTT connection wizard.

Modify the address space of an existing MQTT connection

© Softing Industrial Automation GmbH

The address space of an *MQTT* connection is composed of items. You can specify these items within the MQTT Publisher Connection Wizard.

An existing connection can be modified directly in the Address Space Definition page of the wizard by selecting the connection in the connection overview.

Remove a connection

Delete the selected data source from the list by using the *Delete*  button, pressing the *DEL / DELETE* key on the keyboard or selecting *Delete* from the context menu.

Note: If no connection is selected, the Add a new data source button and Global settings are the only options enabled.

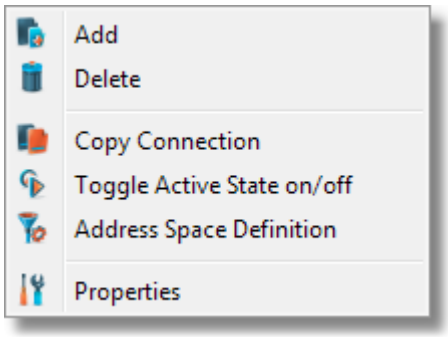
The user can select more than one connection with the help of the *CTRL* or *SHIFT* keys.

A dialog will then prompt the user to confirm the deletion.

Attention: A connection cannot be deleted if at least one data item defined in its namespace is used.

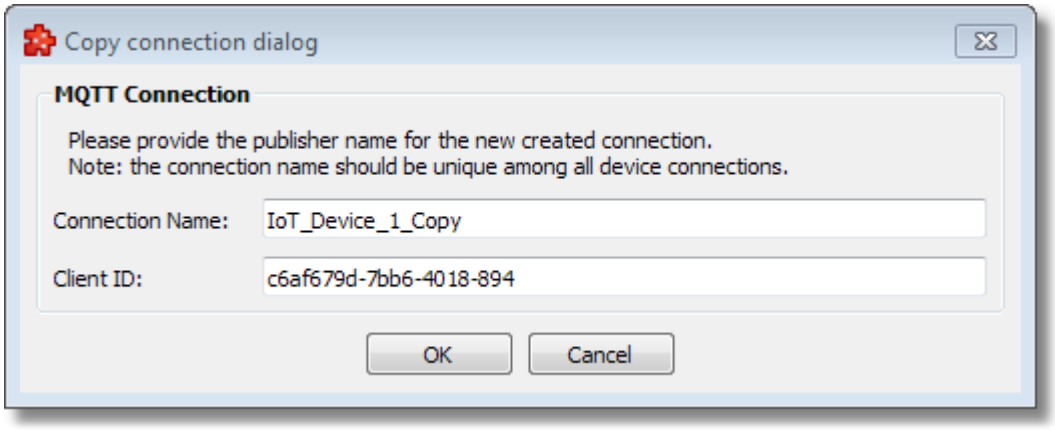
These actions are also provided in the context menu of the MQTT conncetion list. Which actions are available depends on the current selection in the list.

Following an example of a context menu:



Copy Connection

 This function is available only in the context menu and opens a dialog where you can enter *Name* and *Client ID*.



The proposed name is the original connection's name post fixed with "*_Copy*". The endpoint URL is copied from the source connection.

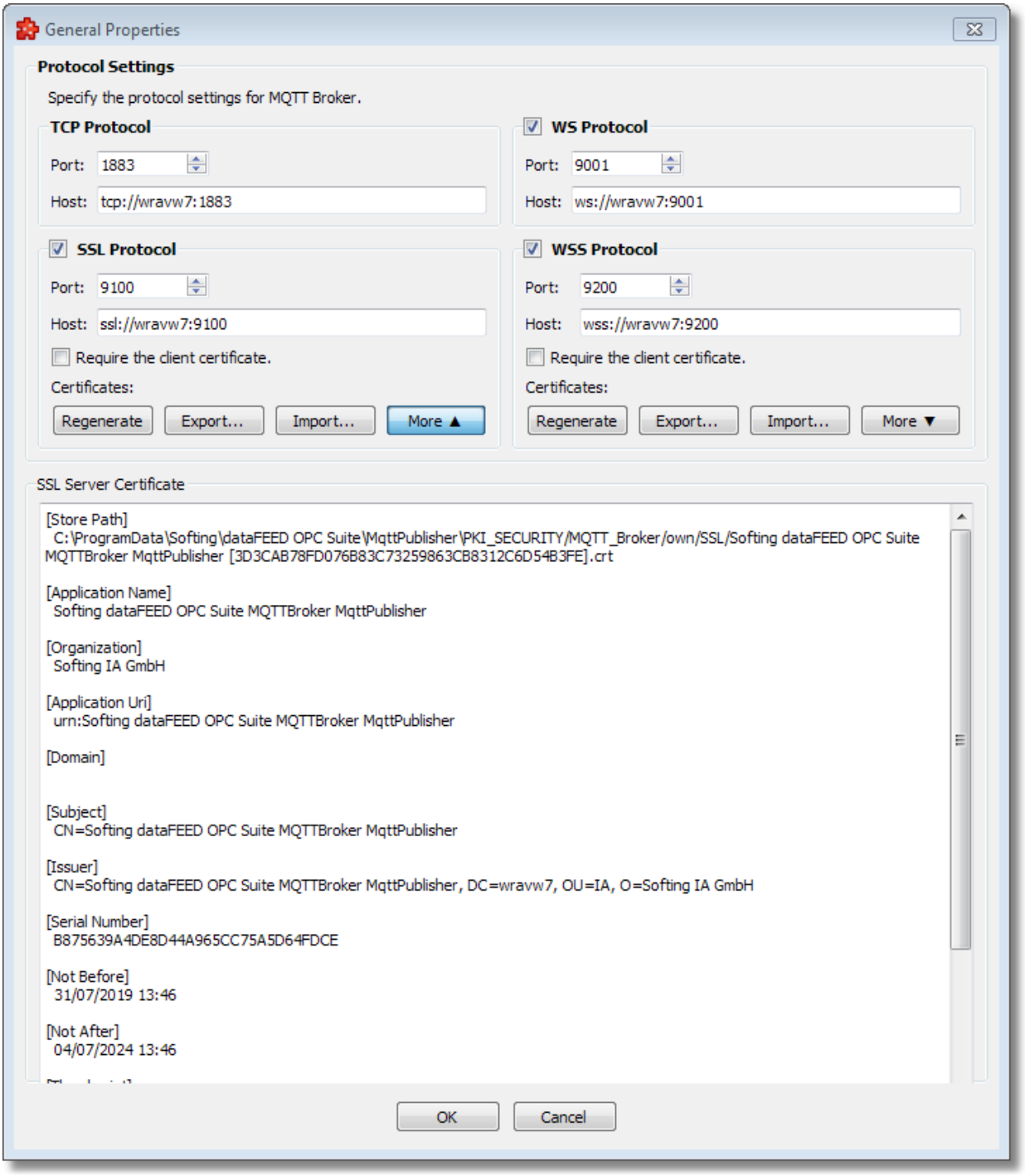
Note: The connection name should be unique among all device connections names. In addition, the *Client ID* should be unique among all defined *MQTT* data sources.

Toggle active state on/off

Toggle active state on/off  This action will enable / disable a device's connection. A disabled connection will be shown as grayed out in the device connection list and it will be ignored by runtime.

General Properties

© Softing Industrial Automation GmbH



The *General Properties* section allows for the customization of protocol settings.

Protocols:
Each section allows customizing port numbers and host URLs, as well as server certificates. Specified port numbers and host URLs will be used in connecting clients which want to send data to dataFEED OPC Suite .

Certificates:

SSL and WSS protocols need certificates to encrypt data communication between dataFEED OPC Suite and connected clients. Thus, certificates for dataFEED OPC Suite (so-called own certificates) are automatically generated when a new configuration is created, an old one is opened, host name is changed from what a configuration was saved with, or when the user presses the *Regenerate* button.

- Regenerate:

Existing certificates for the selected protocol will be deleted and new ones will replace them.

- Export...:

The dataFEED OPC Suite application certificates can be exported with the *Export...* button, which will create several files in the selected directory.
The names of the certificates which are generated are usually in the form of *Softing dataFEED OPC Suite MQTTBroker Configuration1 [0BF952B343C74556C0DBB2E46BA84CE481906408].crt,* which contain the name of the application, the configuration name, a thumbprint enclosed in [] brackets. The extensions may differ: .crt is for certificates (CA and application), .pfx for CA certificate key and .pem for the application certificate key file.

After export the names will have a different form, more easily identifiable: *Configuration1 [SSL_2019-06-06-11_06_22_CA_Certificate].crt, Configuration1 [SSL_2019-06-06-11_06_22_CA_Key].pem, Configuration1 [SSL_2019-06-06-11_06_22_CA_Key].pfx, Configuration1 [SSL_2019-06-06-11_06_22_Server_Certificate].crt and Configuration1 [SSL_2019-06-06-11_06_22_Server_Key].pem.* The meaning of the extensions are the same, but the names contain information like configuration name, protocol type (SSL or WSS) and a timestamp.
The number of files created is more than that of the original certificates, because in addition we create a .pem format of the CA key file, for use with OpenSSL. The PFX file format is used by the OPC UA Certificate Generator tool, which is shipped with dataFEED OPC Suite , to sign application certificates with it. It is converted at export to PEM format to be used with OpenSSL.

- Import...:

User-generated certificate files can be imported from external locations, and can be used for server certificates. They need to be signed by CA certificates and those certificates also need to be provided. See section below.

- More:

The *More* buttons show or hide the contents of the applicaton certificates specific to the protocol section on which they are pressed.

Commands used to create certificates:

Users can create their own certificates for the server side for dataFEED OPC Suite , either by using the OPC UA Certificate Generator tool or OpenSSL. The password for the CA certificate is *Softing1_CA!* and for the application certificate is *Softing1_Server!*.

- for CA:

"C:/Program Files (x86)/Common Files/OPC Foundation/UA/v1.0/Bin/Opc.Ua.CertificateGenerator.exe" -command issue -storePath

© Softing Industrial Automation GmbH

"path_to_store_certificates" -subjectName O="organization"/OU=organization_unit/CN="common_name"/DC="host_name" -ca true -password password -keySize 2048 -hashSize 256 -lifetimeInMonths 60

- for CA signed applicaton (server or client) certificate:

"C:/Program Files (x86)/Common Files/OPC Foundation/UA/v1.0/Bin/Opc.Ua.CertificateGenerator.exe" -command issue -storePath "path_to_store_certificates" -an "application_name" -ikf "path_to_pfx_key_file" -ikp ca_password -keySize 2048 -hashSize 256 -pw application_password -lifetimeInMonths 60

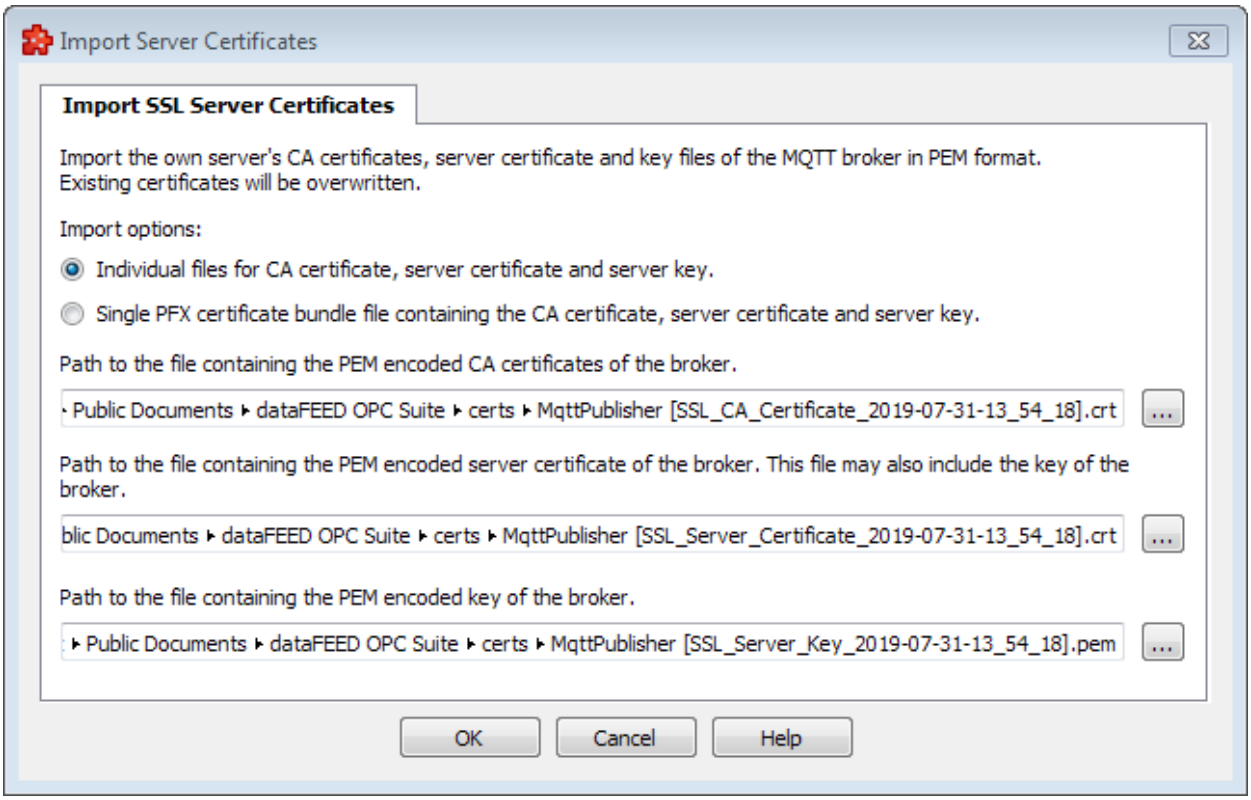
- for converting from DER to PEM:

openssl x509 -inform der -in certificate.cer -out certificate.pem

The values for -storePath, -subjectName, -ikf, -ikp and -password need to be provided by the user.
To create a client certificate the user can export the certificates and use the CA files to sign a newly created certificate with the second command.

If *Require the client certificate* is unchecked, there is no requirement for the client to provide any certificate to the server, authentication is limited to the username and password. If *Require the client certificate* is checked, the client must provide a valid certificate in order to connect successfully. This option by default is unchecked.

Import Server Certificates



Import Server Certificate dialog provides the possibility to import the server certificates (CA certificate, server certificate and server key) used to configure SSL and WSS protocols.

You can import the server certificates via separate files (all in PEM internal format), for this you should provide the following files:

- CA certificate file with extension CRT or PEM;

- server certificate (public server certificate) file with extension CRT or PEM. This file may also include the server key (private server certificate). In this case the next field should be empty.

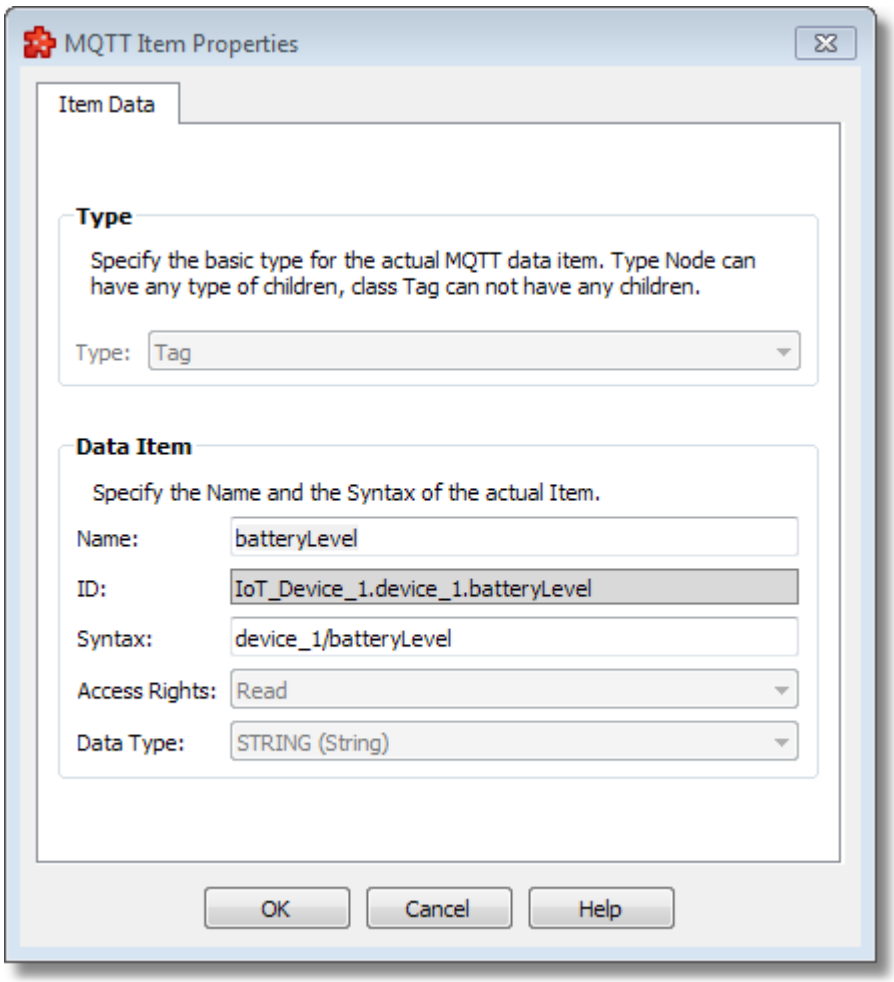
- server key (private server certificate) file with extension CRT, PEM or KEY;

or you can provide a single certificate file that contain the CA certificate, server certificate (public server certificate) and server key (private server certificate) into one single PFX certificate bundle, for this you should provide:

- PFX certificate file in DER encoding;

- the certificate password used to export PFX certificate bundle;

Add or Modify MQTT Item Properties



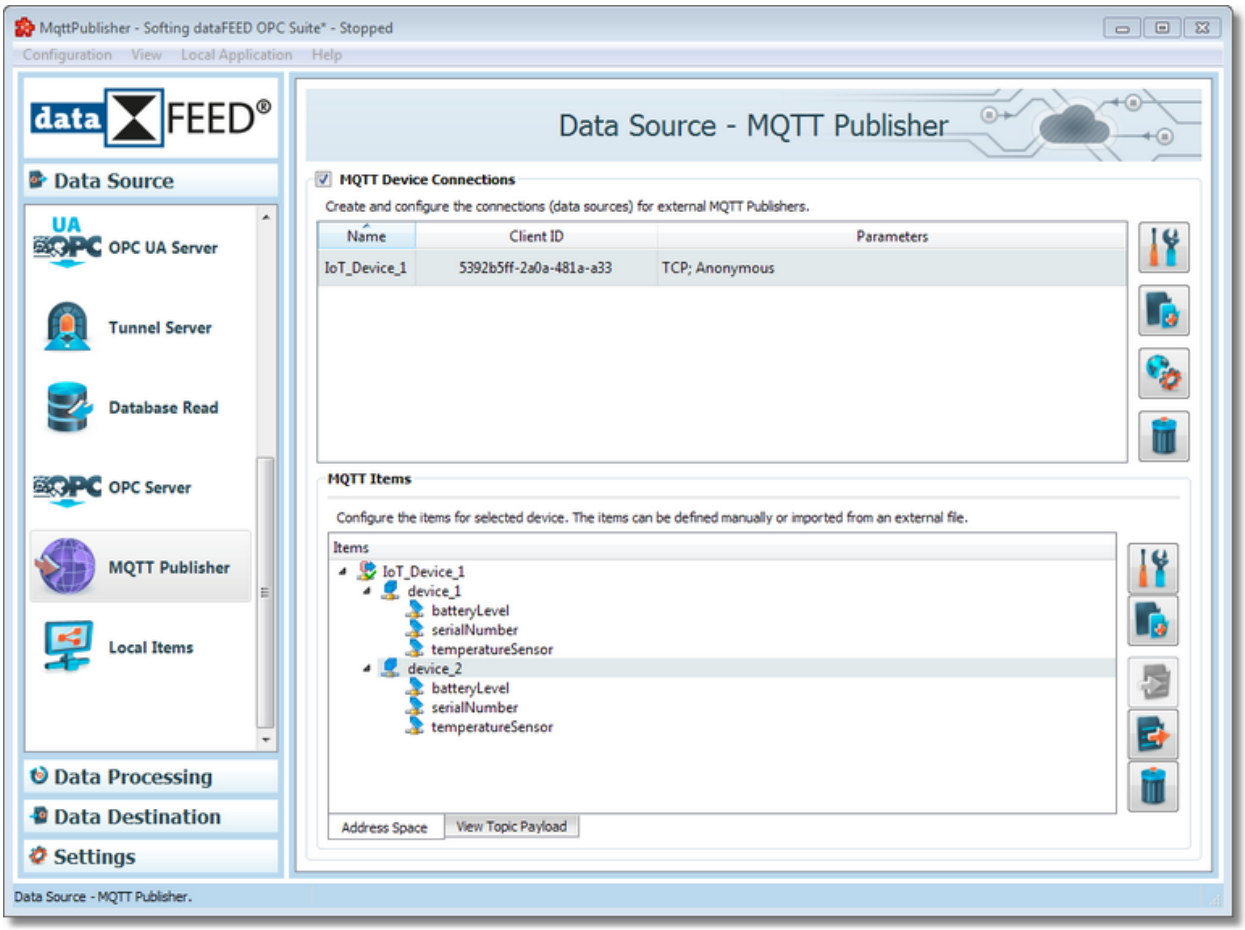
© Softing Industrial Automation GmbH

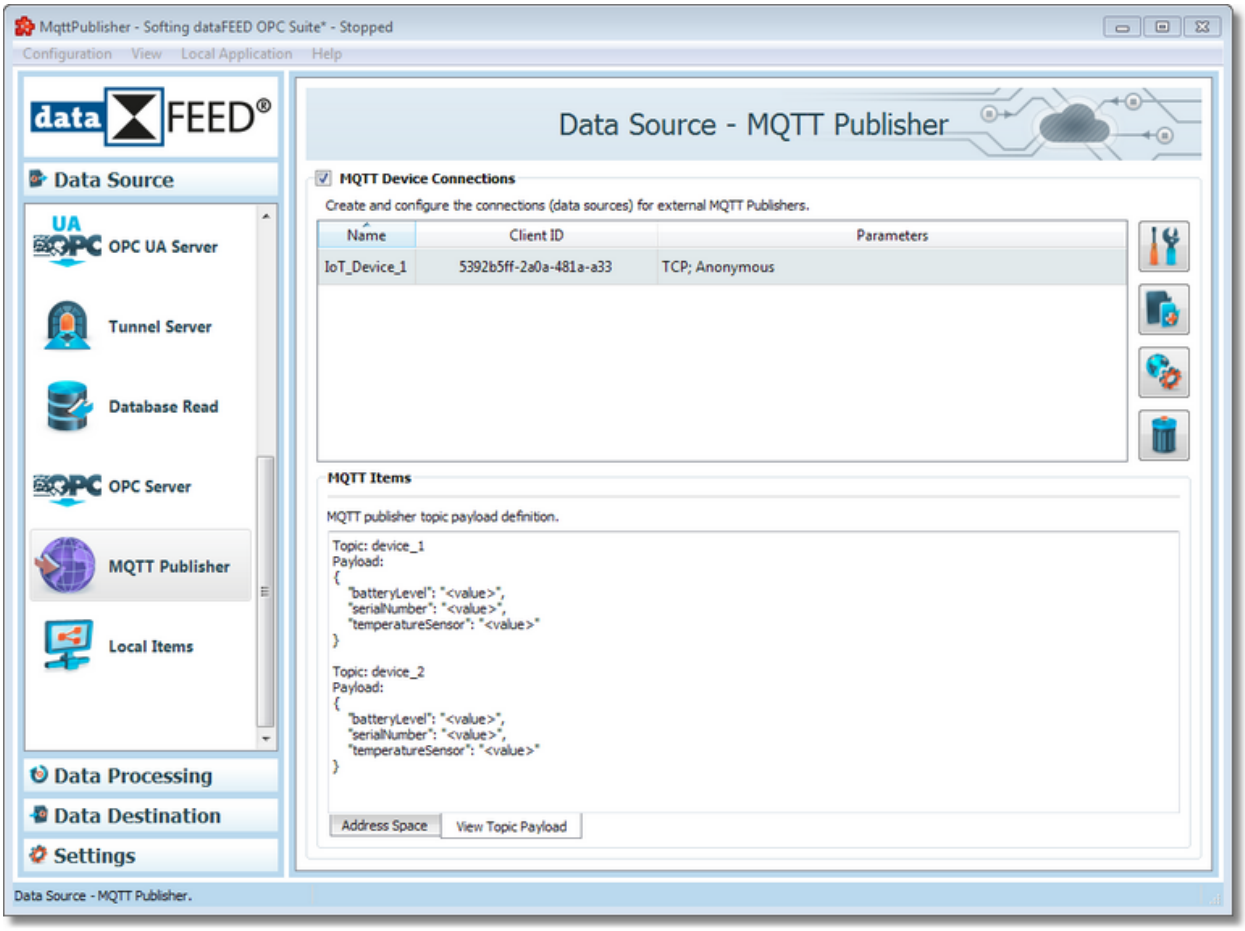
The *MQTT Item Propierties* section allows to add or modify the items from address space of the *MQTT* connection.

Address Space and Topic Payload Overview

When an existing connection is selected, the main view page can also show the address space items, as well as the topic payload definitions for the current connection.

The address space and the topic payload definitions can be seen in the two different tabs in the main screen, below the definition list: Address Space, View Topic Payload.





This lower part of the main view contains tab pages which can also be found in the action wizard and allow easier access for the user to change the address space and to access the topic payload view content, without going through the wizard pages. More information can be found on the specific help pages: Topic Payload Overview.

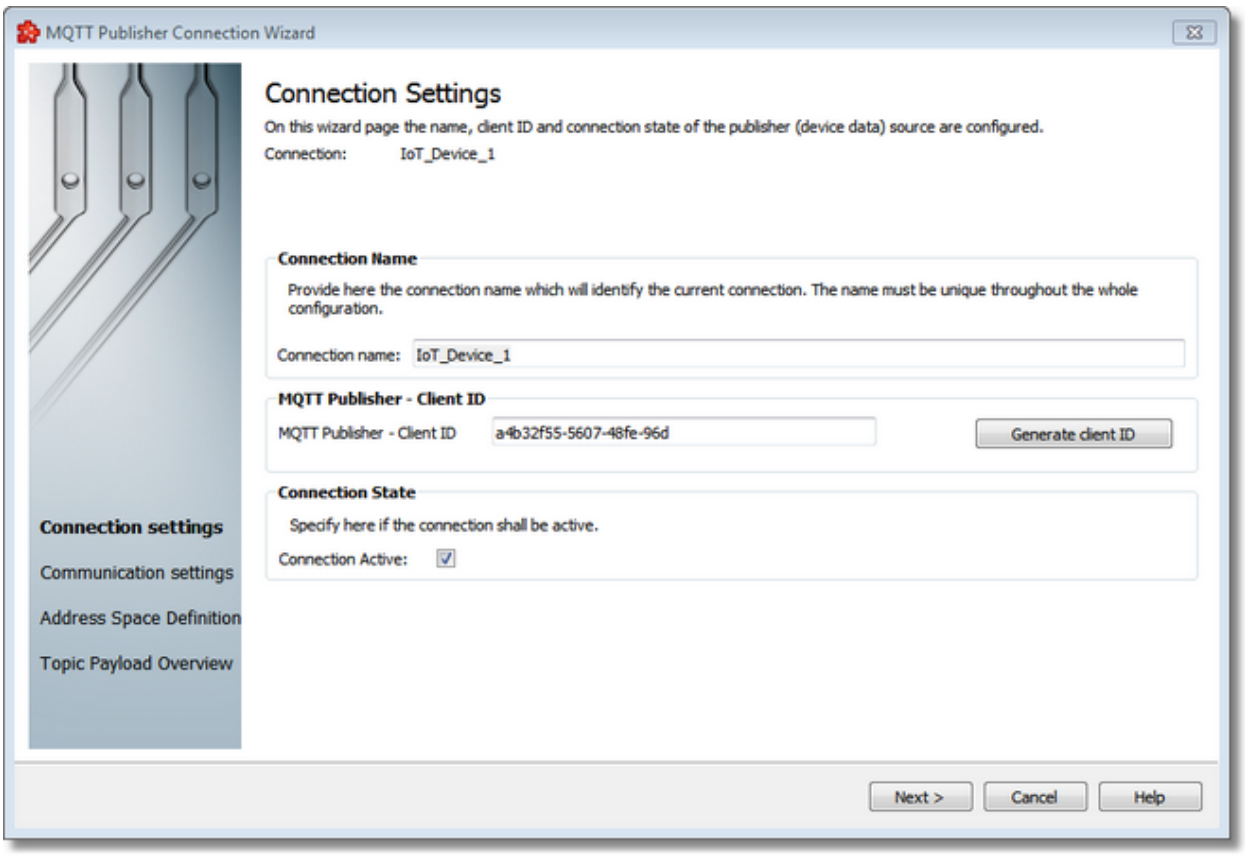
7.3.1.12.2MQTT Publisher Connection Wizard

▾ With the MQTT Publisher Connection Wizard you can create and modify a data source connection to an external MQTT Publisher:

- Specify connection settings @ MQTT Publisher Connection Wizard - Connection Settings
- Specify communication settings @ MQTT Publisher Connection Wizard - Communication Settings
- Specify address space definition @ MQTT Publisher Connection Wizard - Address Space Definition
- Review the topic hierarchies and their payloads @ MQTT Publisher Connection Wizard - Topic Payload Overview

© Softing Industrial Automation GmbH

This page allows to specify connection settings such as e.g. the unique connection name for identifying the connection.



Connection Name

A unique identifier for the current connection. In case the provided name is already used by other connection, the user use is not allowed to proceed to the next wizard page.

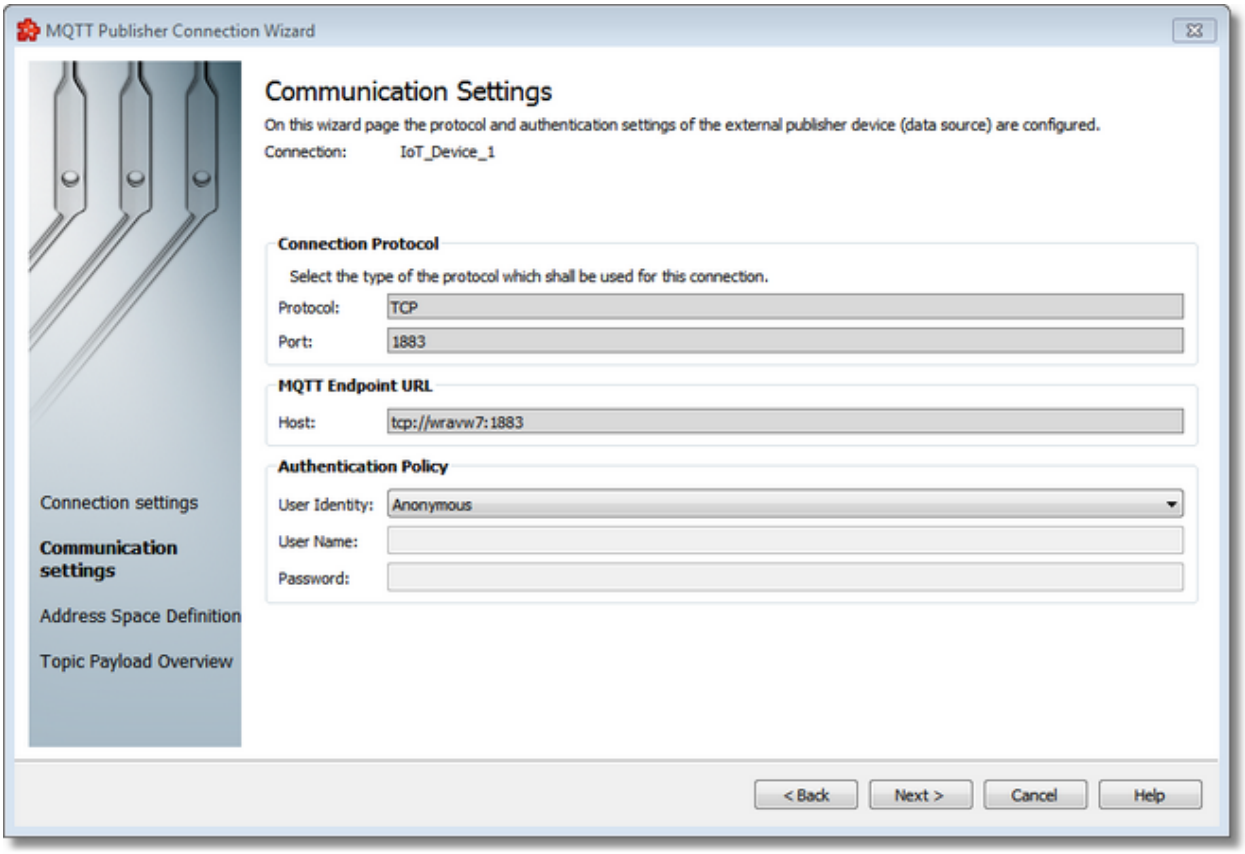
MQTT Publisher - Client ID

The *Client ID* must be unique from all *MQTT* data sources connections for current configuration. The internal broker is configured in a way that it only accepts connections with the specified client ID.

Connection active

This flag specifies if the current connection will be activated or not.

This page allows to parameterize the communication and authentication of the *MQTT Publisher*.



Connection Protocol

For the transport protocol, the following variants could be chosen:

1. *TCP - MQTT* uses the plain *TCP*. The default port is 1883.
2. *SSL - MQTT* uses *SSL* or *TLS* security upon *TCP*. The default port is 9100.
3. *WS - MQTT* over *WebSockets (WS)*. The default port is 9001.
4. *WSS - MQTT* over secure *WebSockets (WSS)*. The default port is 9200.

Note: The default ports can be changed from *General Properties* windows.

Note: By default the protocols SSL, WS and WSS are disabled. You can enable the protocols from *General Properties* windows.

MQTT Endpoint URL

The MQTT Endpoint URL details the host URL that will be used in connecting clients which want to send data to dataFEED OPC Suite .

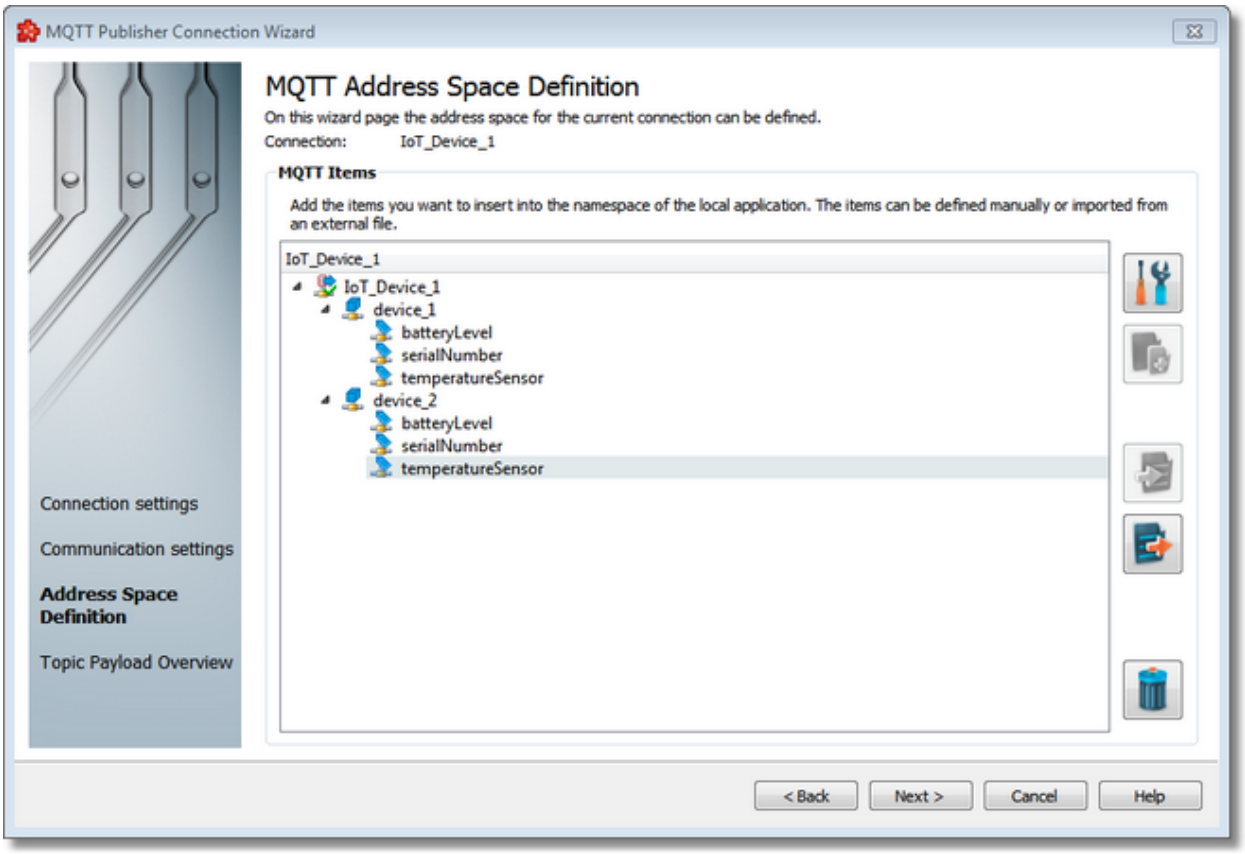
© Softing Industrial Automation GmbH

Authentication Settings

MQTT provides username/password authentication as part of the protocol. In this case you need to select the entry *Username and Password* in *User Identity* then fill out the input fields for user name and password accordingly.

If no identification is required, leave the entry *Anonymous* in *User Identity*. This is the default option.

This page allows to configure the address space of the *MQTT Publisher* connection.



MQTT Address Space Definition

An editable tree view where the user can change the address space structure or its content by adding, deleting or modifying topics.

Note: You cannot use the characters '.', '|', '+', '#', *Tab* and *Newline* for node and item names.

There are three categories of data items that may be used in the tree:

- **Root node** (having the name of the current connection). This node is read only, the user has just the possibility to add or delete children of this node.
- **Nodes:** these elements are associated with branches in tree. They can have child nodes. But no value can be written to the nodes.

Note: The items of type Node represent intermediate nodes in the tree, they are used for hierarchy purposes.

- **Tags:** these elements are associated with leaves in the tree.

Operation

Depending on the current selection in the topic-tree, the following operations are possible using the buttons on the right, the context menu or the corresponding shortcuts:

- Item properties (Ctrl+P) 
- Add new topic (Ctrl+A) 
- Import from file (Ctrl+I) 
- Export to file (Ctrl+E) 
- Delete item(s) (Del) 

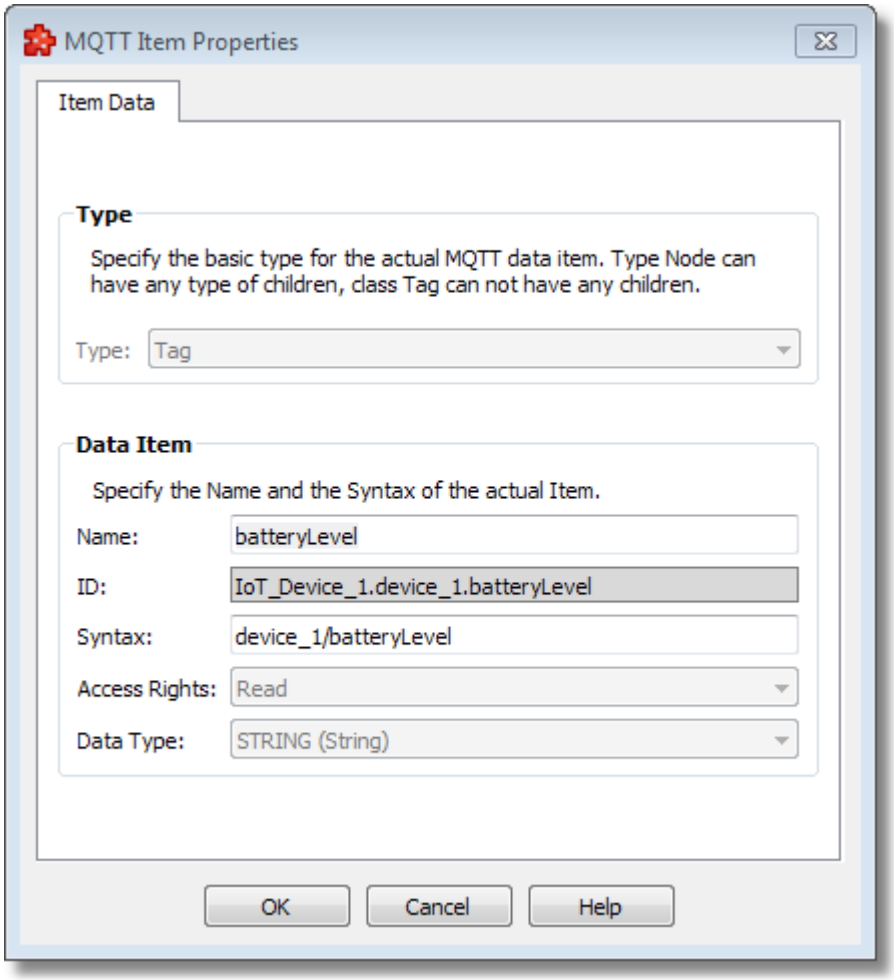
Two more operations are supported that are only accessible from the context menu:

- Copy item(s) 
- Paste item(s) 

Item properties / Add new item

The *MQTT Item Propierties* section allows to add or modify the items from address space of the *MQTT* connection.

© Softing Industrial Automation GmbH



Type

This list box allows to specify the item class. You can choose between *Node* and *Tag*.

Name

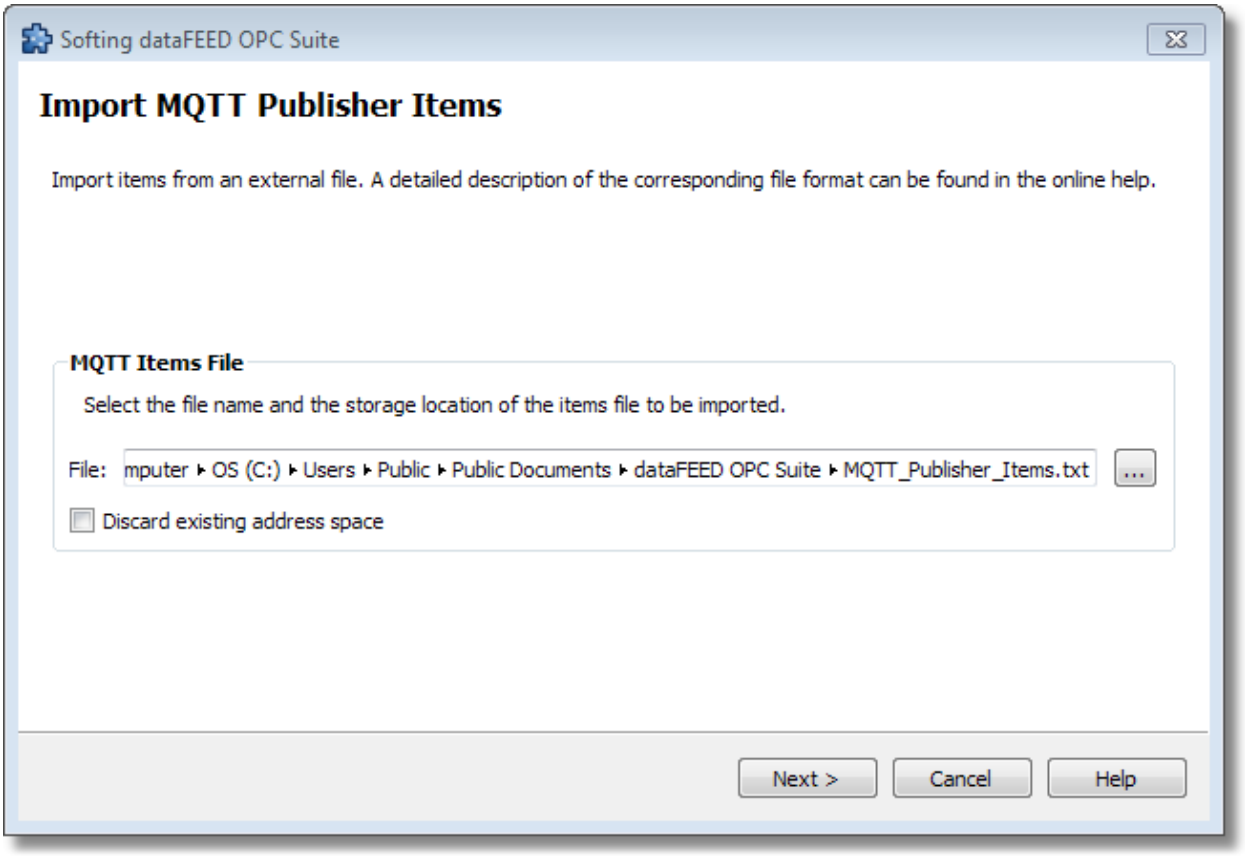
Here you can enter the item name for the current hierarchy level. The name you are using must be unique for the corresponding parent node. The name may only consist of printable ASCII characters. Special characters such as '.', '|', '+', '#', *Tab* and *Newline* must not be used.

Syntax

Here you can enter the item syntax for the current hierarchy level. This syntax will be mapped with the keys from received *MQTT* messages.

Import from file

The *Import MQTT Items* section allows to import the items from an external file.

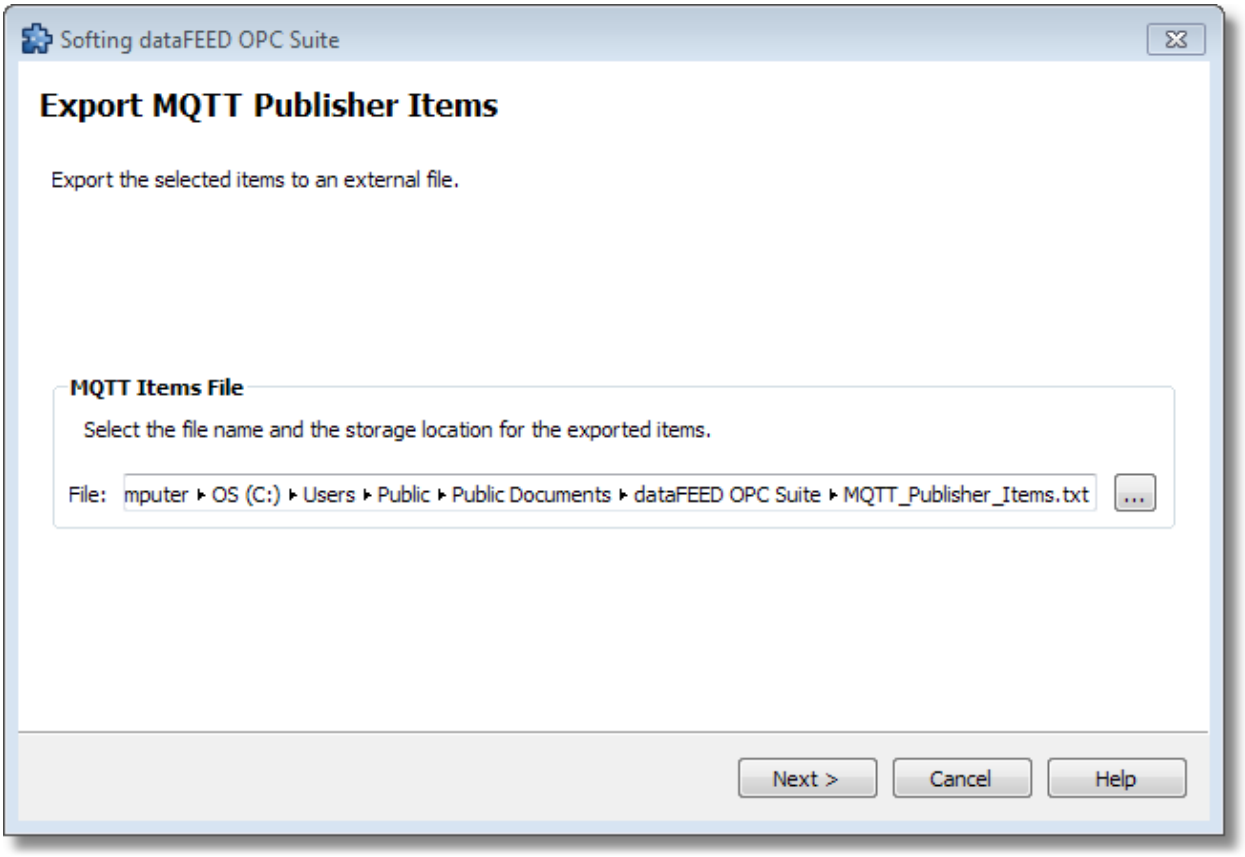


The *Discard existing address space* specifies whether the current address space should be deleted from Configurator and be replaced with the one imported from the given file.

Export to file

The *Export MQTT Items* section allows to export the address space to an external file.

© Softing Industrial Automation GmbH



Delete item(s)

Triggering the *Delete data items*  action with the corresponding button on the right wizard side or *Delete* from the context menu will remove the currently selected item.

Attention: The root node of the namespace tree can not be deleted.

Attention: A node that is currently used in a data processing or data destination action can not be deleted.

Copy

The *Copy*  command is available in the context menu of the item *tree*. Triggering this action will copy the selected topic to the local clipboard. All topic properties are also copied. Even if the topic contains child topics, the complete branch is copied.

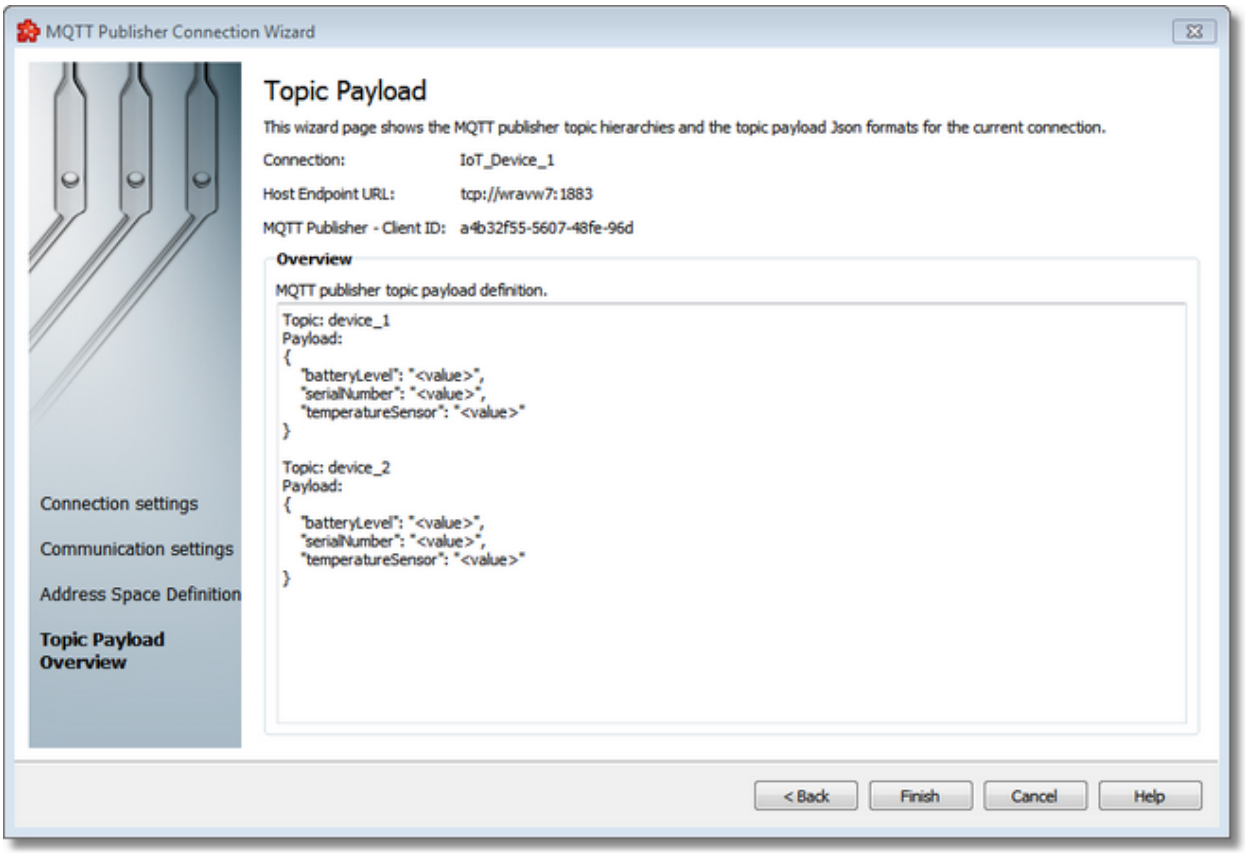
Note: The root node of the namespace tree does not provide a *Copy* action.

Paste

The *Paste* command is available in the context menu of the item tree. This function will add the item currently stored in the local clipboard as a child of the currently selected item. Whether the action is available in the context menu depends on the type of the currently selected node and the type of the node in the clipboard.

Note: The *Paste* action is only available if an item was previously copied to the clipboard.

On this wizard page the user is presented with the final form of the content for the current connection, regarding the MQTT publisher topic hierarchies and the topic payload Json formats.

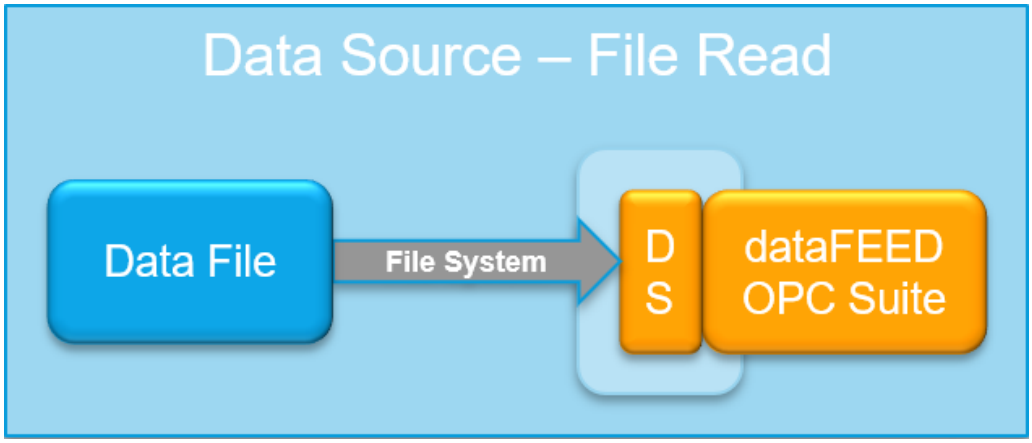


This is a read-only view, used just for reviewing the correctness of the content. The user can only copy the content from the topic payload definition pane.

7.3.1.13 File Read

This section of the Softing dataFEED OPC Suite Online Help describes how to configure *File Read* external data file source.

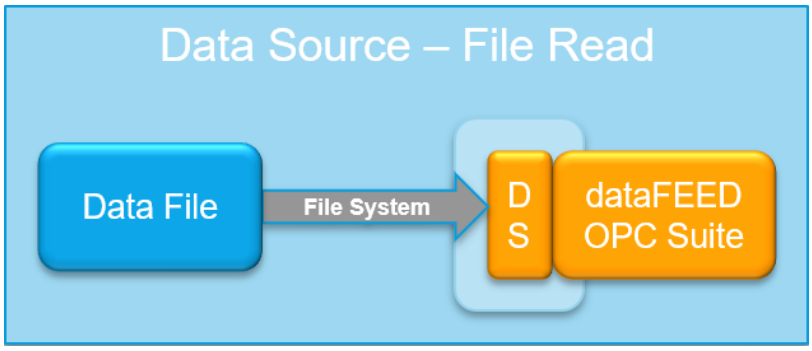
© Softing Industrial Automation GmbH



File Read external data file source allow to read data from external data files. The read data can be then processed, saved, analyzed or transferred by using the *Data Processing* and/or *Data Destination* functionalities.

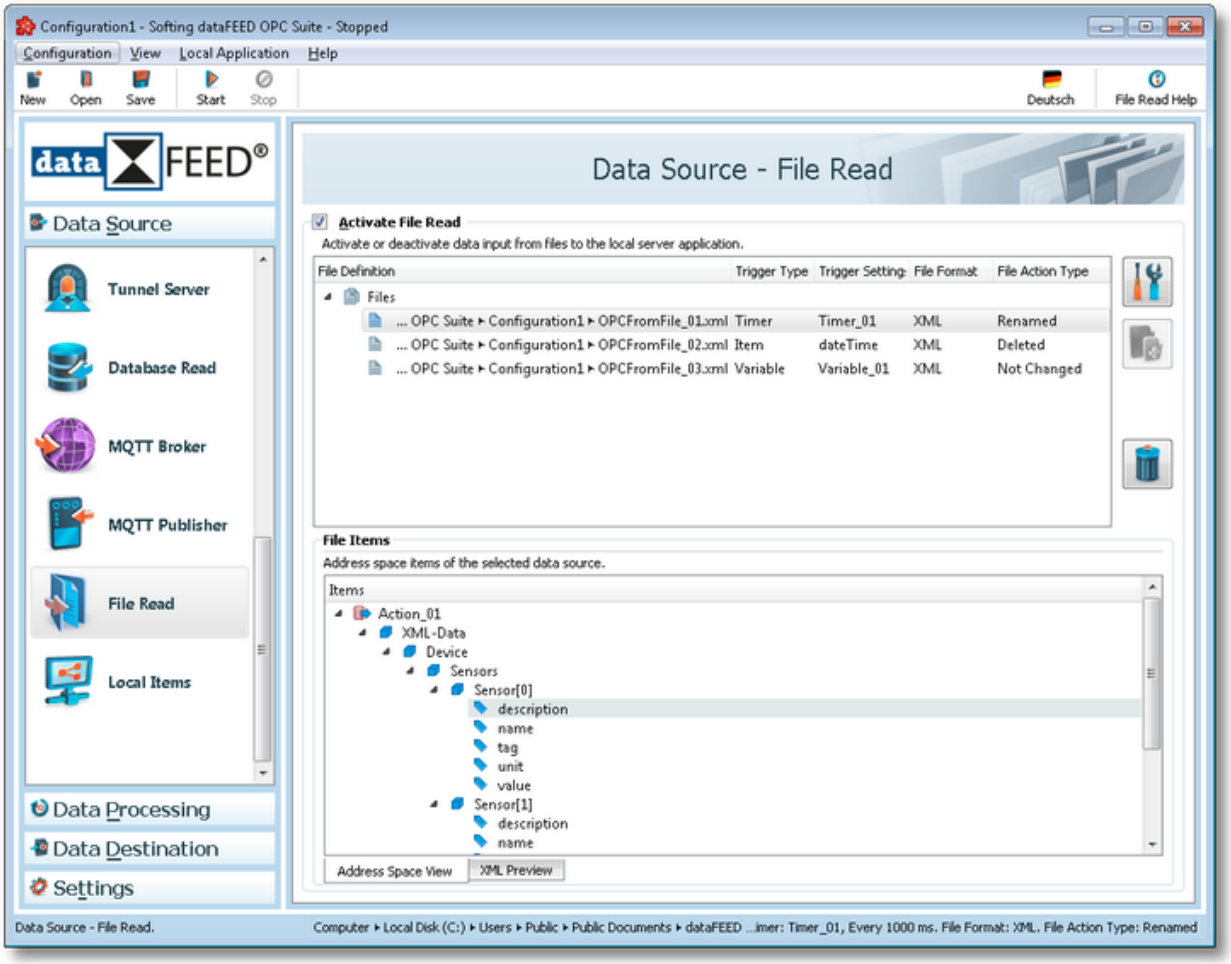
A detailed description on how to configure access to *File Read* external data file source can be found in the Configuration section.

7.3.1.13.1Configuration



In order to configure *File Read* data sources select *Data Source* in the left window part and then select the *File Read* icon. The *File Read* data source configuration overview is opened.

The image below illustrates the main user interface of the *Data Source File Read* add-on:



Add a File Read action

The user can define a new File Read action to a specific input data file, by using the button *Add*  or the menu command *Add* in the context menu. The File Read action wizard is opened.

Modify an existing File Read action

The configuration of an existing File Read action can be modified. To do so, select the line in the overview of the File Read actions of input data files, that contains the File Read action to be modified. Then you can use the button *Properties*  or the menu command *Properties* from the context menu to open the File Read action wizard.

Remove a File Read action

Delete the selected File Read action from the list by using the *Delete*  button, pressing the *DEL / DELETE* key on the keyboard or selecting *Delete* from the context menu.

© Softing Industrial Automation GmbH

Note: If no input data file read action is selected, the Add a new file read action button is the only option enabled.

A dialog will then prompt the user to confirm the deletion.

Attention: A File Read action cannot be deleted if at least one data item defined in its namespace is used.

Modify Trigger Type

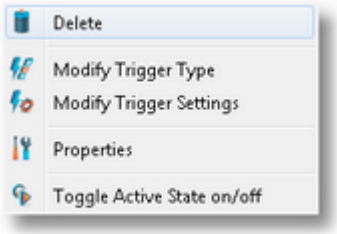
Modify the trigger type for the selected File Read action by selecting the *Modify Trigger Type* action from the context menu.

Modify Trigger Settings

Modify the trigger settings for the selected File Read action by selecting the *Modify Trigger Settings* action from the context menu. For more information see the Timer Trigger Wizard, the Item Trigger Wizard or the Variable Trigger Wizard.

These actions are also provided in the context menu of the File Read actions definition list. Which actions are available depends on the current selection in the list.

Following is an example of a context menu:

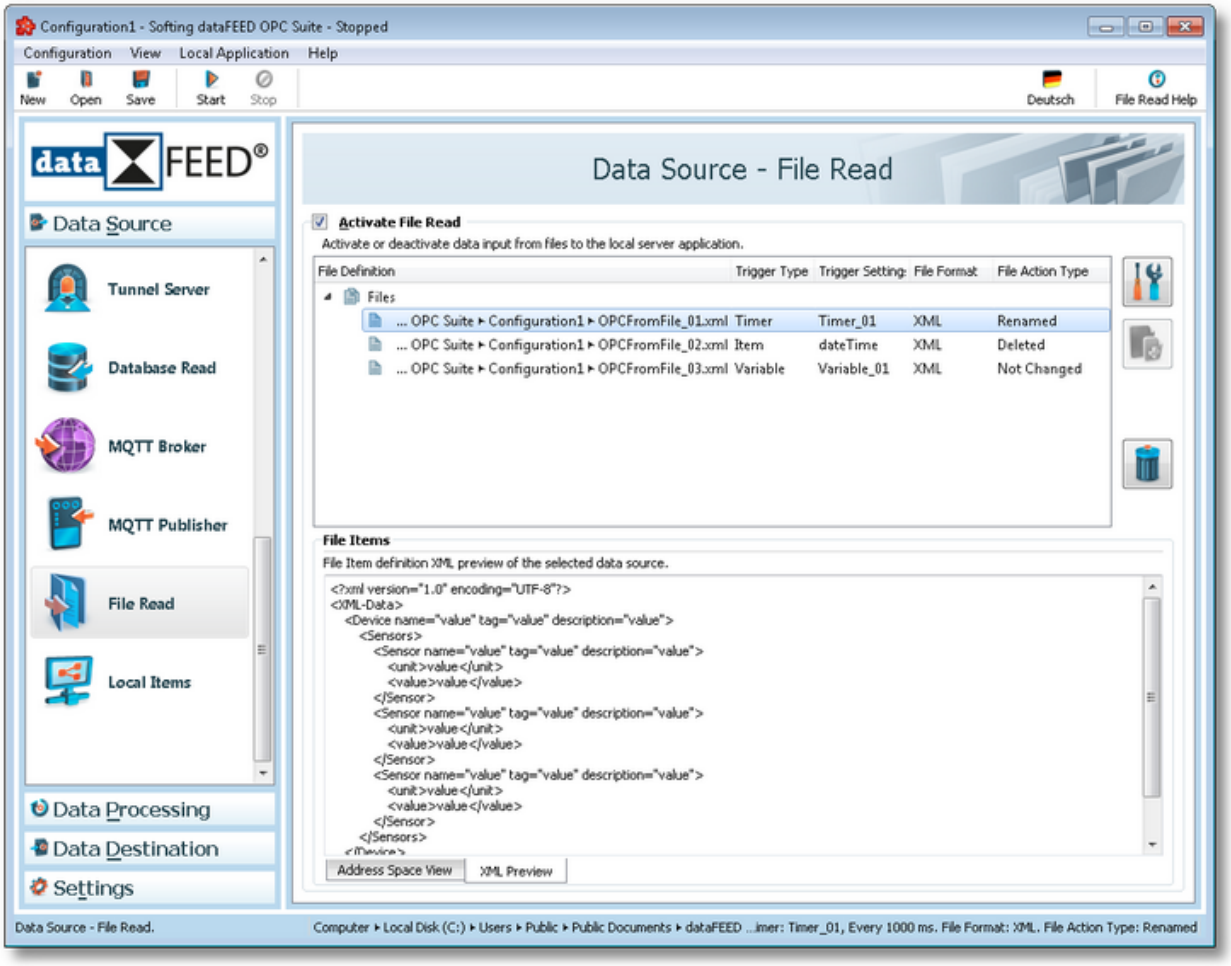


Toggle active state on/off

Toggle active state on/off  This action will enable / disable a File read action to an input data file. A disabled File Read action will be shown as grayed out in the File Read actions definition list and it will be ignored by the dataFEED OPC Suite application runtime.

File Items

The lower pane of the Data Source File Read add-on main user interface illustrates the Address Space View and the XML Preview.



The Address Space View and the XML Preview illustrate not editable views of the address space with the selected file data items, and the XML content form for a selected File Read action.

The File Read Action Wizard allows the user to configure a File Read action, to define manipulation options for the specified input data file, to import the file items from an external XML/XSD file, to select the data items that the user wants to add to the namespace of the dataFEED OPC Suite application, and to determine the File Read action's trigger properties.

7.3.1.13.2File Read Action Wizard

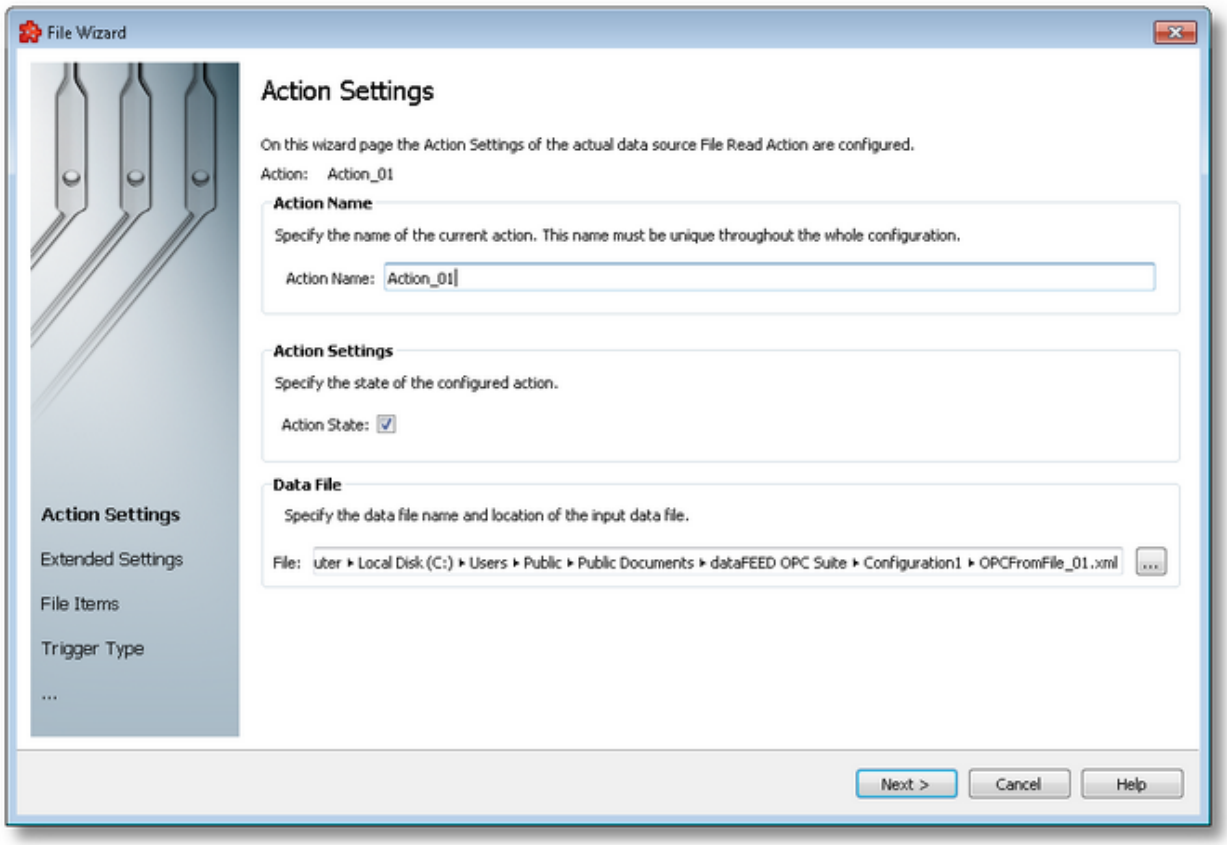
With the File Read Action Wizard you can create new external data file source definitions and modify the existing ones.

- Specify the data file source action settings @ File Read Action Wizard - Action Settings
- Specify the data file source extended settings @ File Read Action Wizard - Extended Settings
- Specify the data file source file items definition @ File Read Action Wizard - File Items

© Softing Industrial Automation GmbH

- Choose the file read action trigger type @ File Read Action Wizard - Trigger Type
- Select the timer trigger @ File Read Action Wizard - Timer Selection
- Define the timer identity @ File Read Action Wizard - Timer Identity
- Choose the timer type @ File Read Action Wizard - Timer Type
- Set the timer interval @ File Read Action Wizard - Interval Timer Settings
- Select the trigger item @ File Read Action Wizard - Trigger Item Selection
- Select the variable trigger @ File Read Action Wizard - Variable Trigger Selection
- Adjust schedule timer settings @ File Read Action Wizard - Schedule Timer Settings

The Action Settings wizard page allows you to specify the unique file read action name for identifying the action, to provide action state settings and to provide the file path and name for the input data source file.



Action Name

A unique identifier for the current action. In case the provided name is already used by another action, the user is not allowed to proceed to the next wizard page, until they provide a different name.

Action State

This flag specifies whether or not the current file read action will be executed.

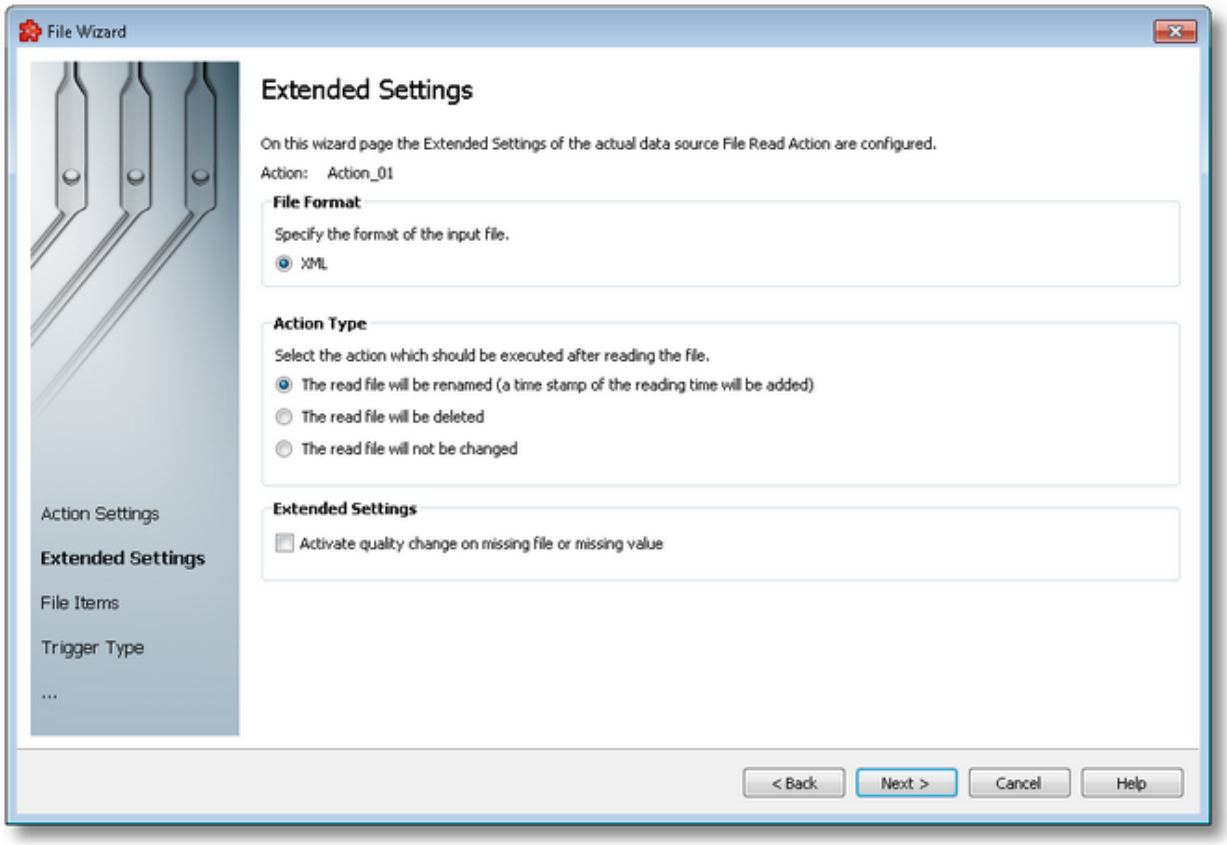
Data File

The path and name of the input data file can be typed in the provided text box. The input data file can also be selected by clicking the browse button located to the right of the *File* text box.

If the typed path does not exist, it will be automatically created by the dataFEED OPC Suite Configurator. With the help of the browse button, an existing file may be chosen or a new file may be specified.

File Read Action Wizard - Extended Settings >>>

The Extended Settings wizard page allows you to specify the format of the input data file, and to select the action which should be executed after reading the input data source file.



© Softing Industrial Automation GmbH

File Format

This flag specifies the format of the input data source file (XML).

Action Type

This flag specifies the action which should be executed after reading the input data source file:

- The read file will be renamed (a time stamp of the reading time will be added).
- The read file will be deleted.
- The read file will be not changed.

Extended Settings

This flag allows the activation of additional error handling features.

- Activate quality change on missing file or missing value.

With this option inactive, a positive handling is provided. In case of a missing file (file already deleted/renamed by a previous operation or just not written yet by the external tool), or a missing attribute from a file, the File Items preserve their previous value from their cache, and their quality remains unchanged.

With the option active, the positive handling becomes inactive. In case of a missing file, or missing attribute from a file, the quality of the File Items changes to Bad.

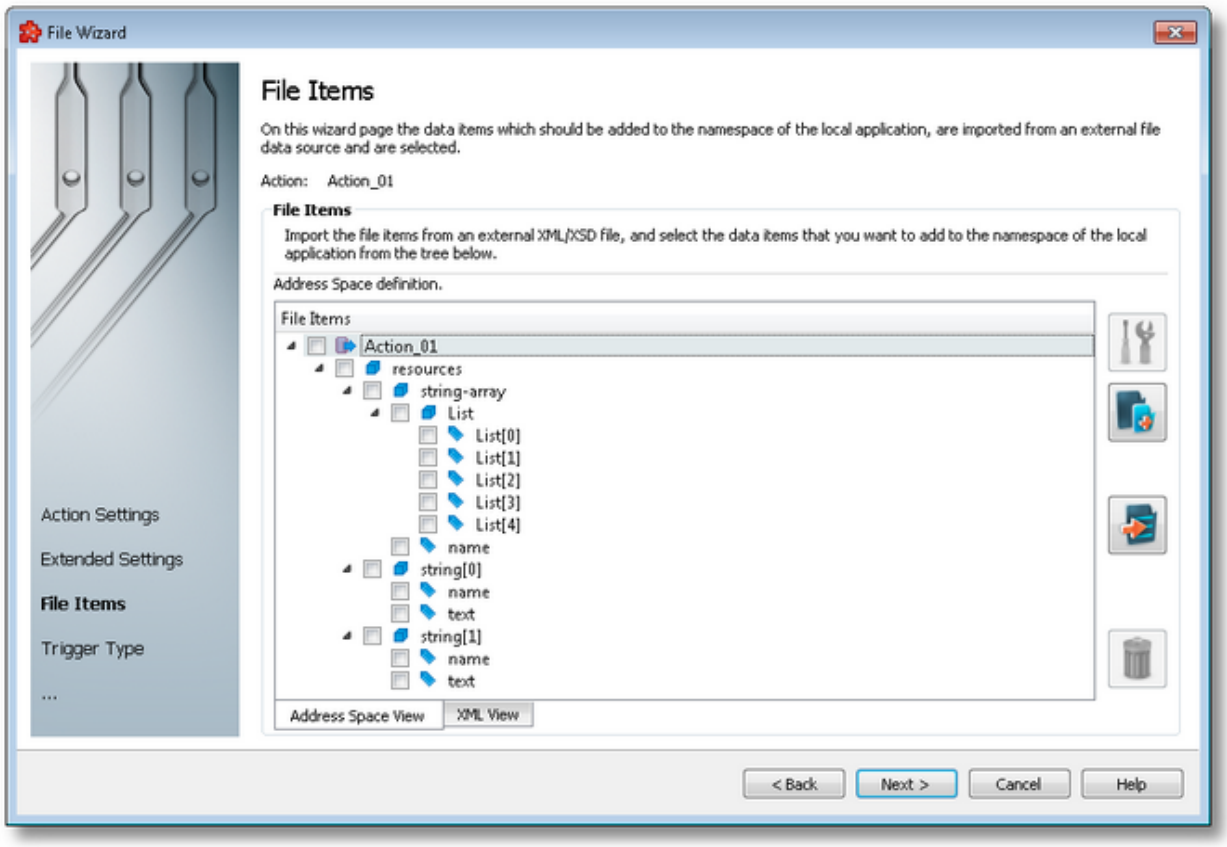
Use case description	**Outcome of option inactive (positive handling)**	**Outcome of option active (error handling feature enabled)**
Missing file	Quality unchanged, application log warning message occurs.	Quality changes to **BAD,** OPC_Quality_CommFailure (0x18).
Invalid file	Quality unchanged, application log warning message occurs.	Quality changes to **BAD,** OPC_Quality_DeviceFailure (0x0c).
Missing attribute from a file	Quality unchanged, application log warning message occurs.	Quality changes to **BAD,** OPC_Quality_Bad (0x00).
Invalid attribute value from a file	Quality unchanged, application log warning message occurs.	Quality changes to **BAD,** OPC_Quality_SensorFailure (0x10).

<<< File Read Action Wizard - Action Settings File Read Action Wizard - File Items >>>

The File Items wizard page allows the user to define the namespace of a File Read action. The wizard page contains two tab views, the *Address Space View* and the *XML View*.

Address Space View

The *Address Space View tab* allows the user to import the file items from an external XML or XSD input data files, to manually add new file items, and to select the data items that the user wants to add to the namespace of the local application. The user can also edit the properties, such as the item name or the source item ID of a selected data item.



There are three categories of data items that may be used in the namespace tree:

- **Root node** (having the name of the current file read action). This node is read only, the user has just the possibility to add or delete children of this node.

- **Nodes**: these elements are associated with branches in the tree. They can have child nodes.

© Softing Industrial Automation GmbH

Note: Arrays are represented as nodes with values in the namespace tree. In this case, the individual array elements are the tags of the parent array node.

- **Tags**: these elements are associated with leaves in the tree.

Note: The local application has read access rights to the file items from the namespace tree.

Operation

Depending on the current selection in the namespace tree, the following operations are possible using the buttons on the right, the context menu or the corresponding shortcuts:

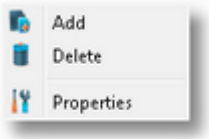
Properties of the selected item (Ctrl+P) 

Add a new item (Ctrl+A) 

Import from XML/XSD file (Ctrl+I) 

Delete data items (Del) 

An example for a context menu of the namespace tree is shown below:



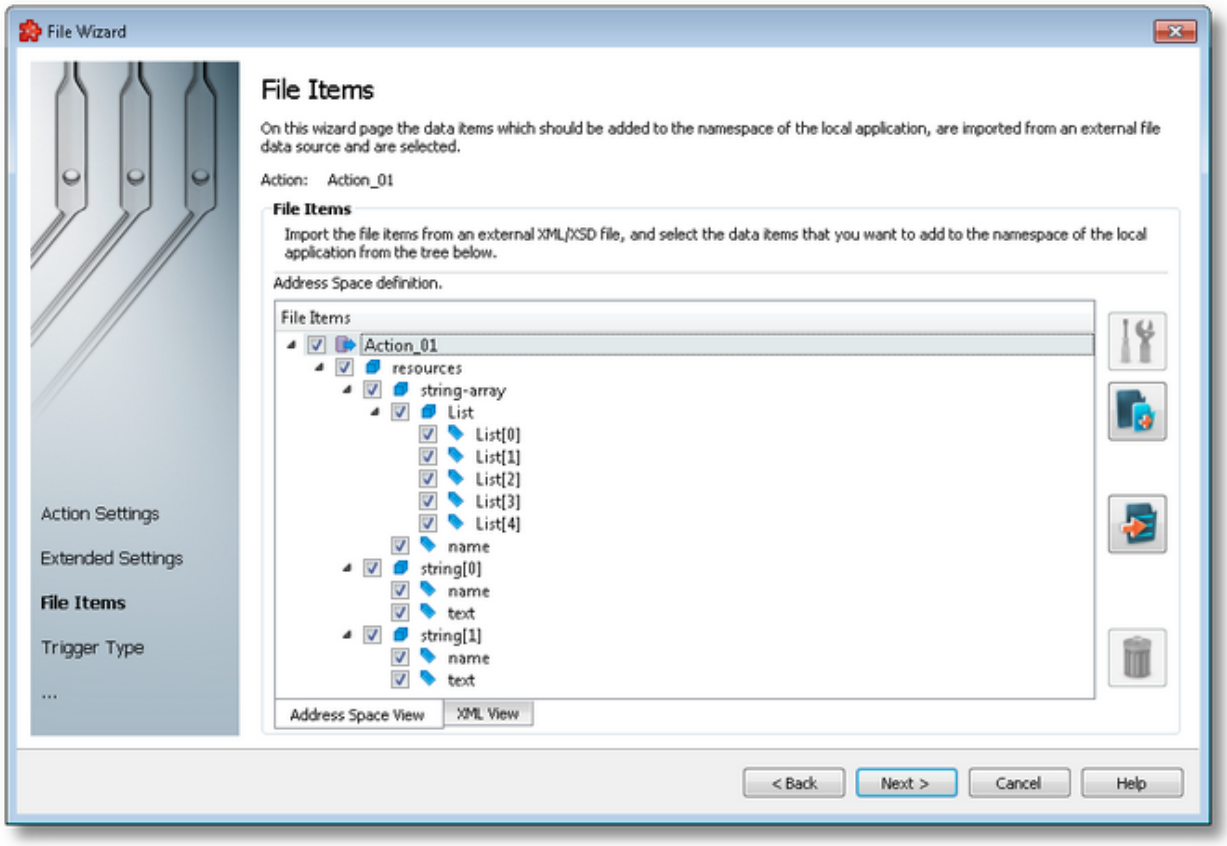
Delete data items

Triggering the *Delete data items*  action with the corresponding button on the right wizard side or *Delete* from the context menu will remove the currently file item from the namespace tree.

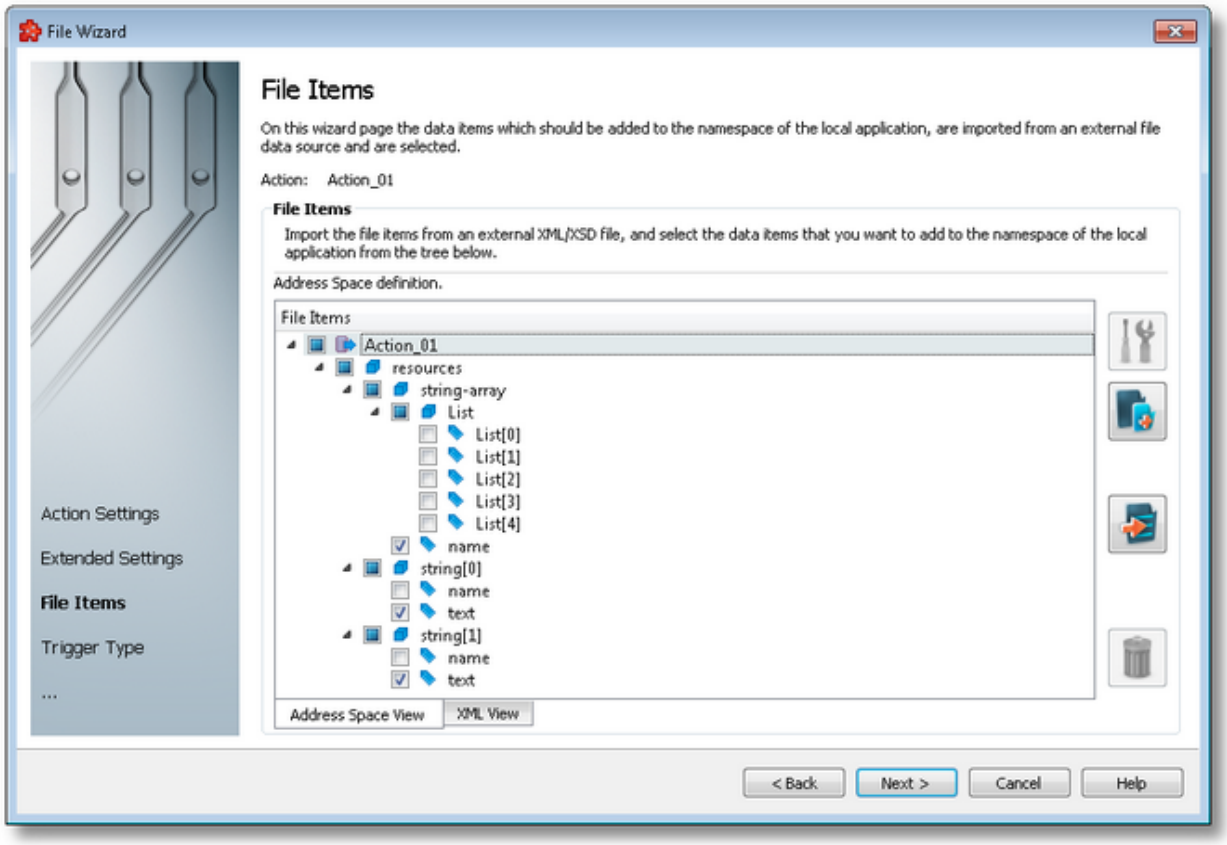
Note: The root node of the namespace tree can not be deleted.

Note: A fie item that is currently used in a data processing or data destination action can not be deleted.

Following the import of the file data items from external files or by manually adding the file items, the user will add the data items to the namespace of the local application by selecting them from the File items tree.

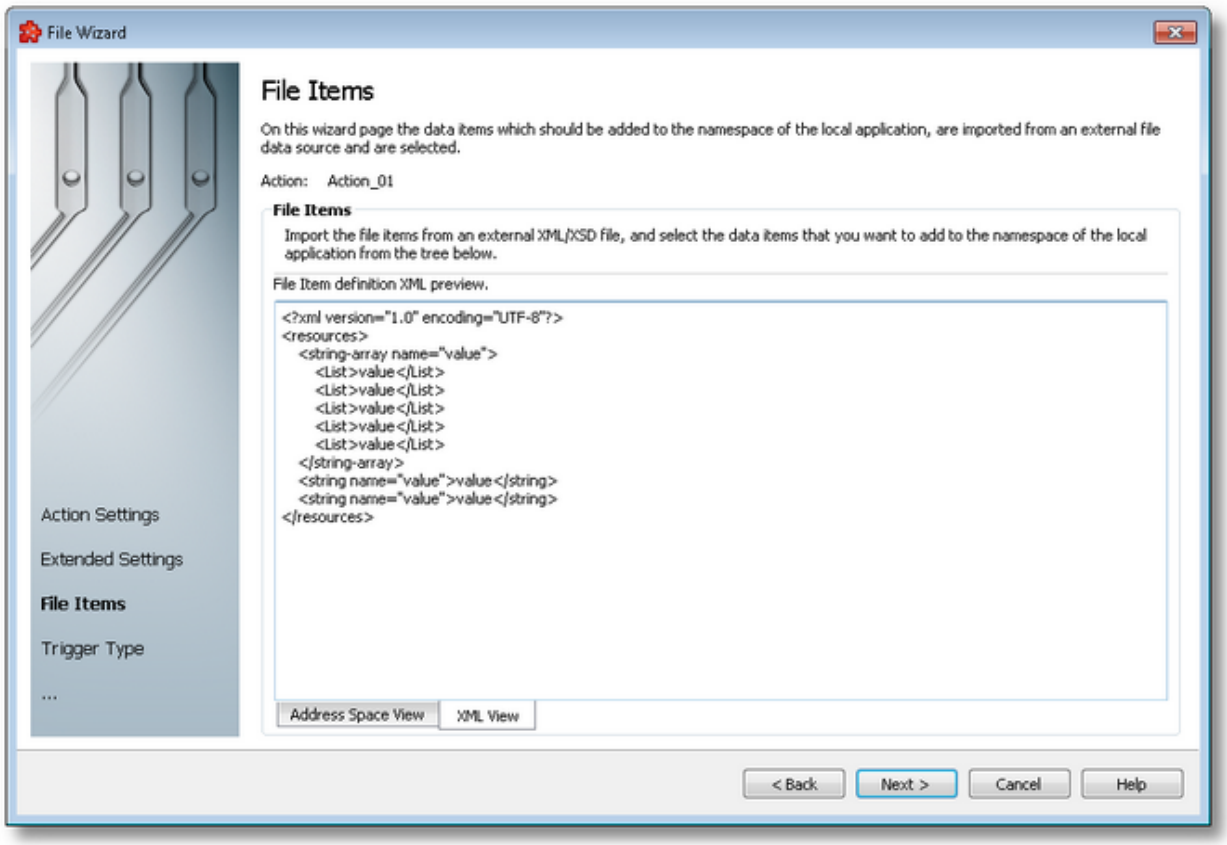


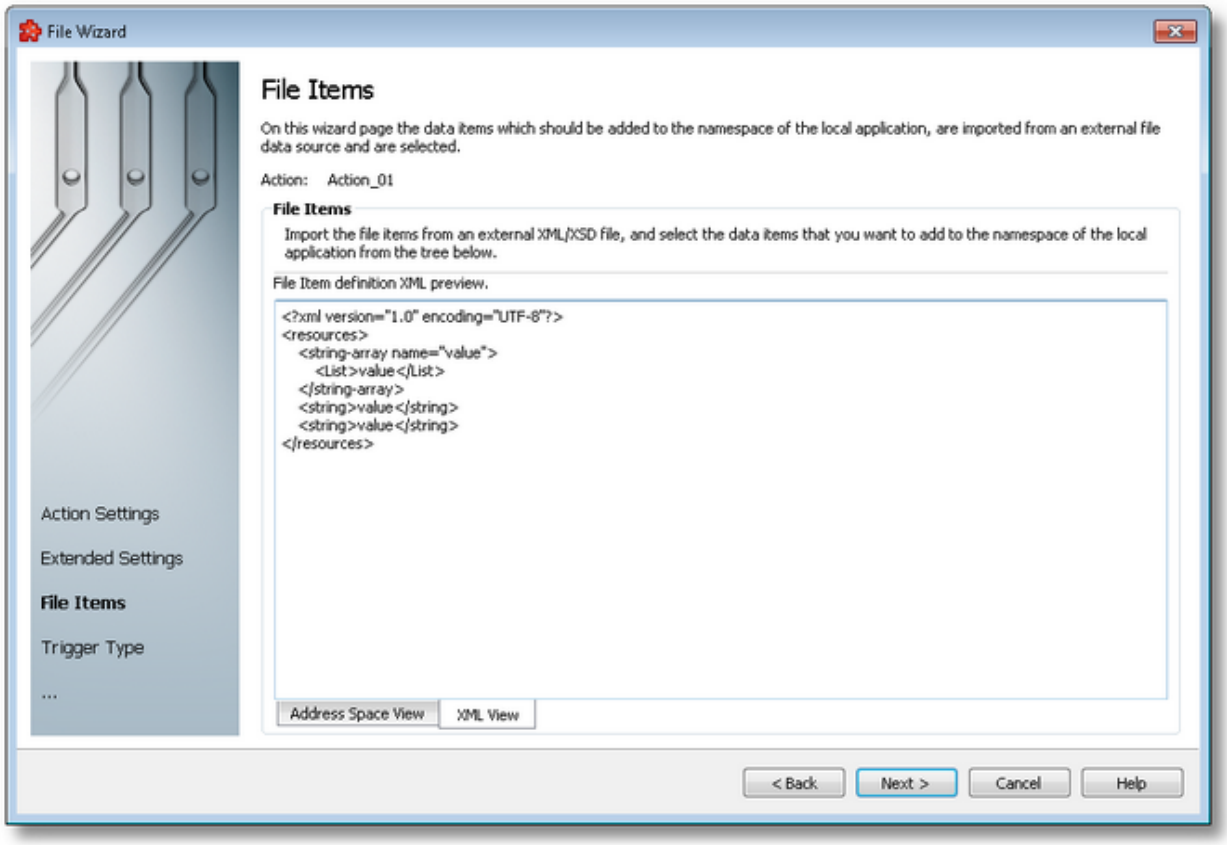
© Softing Industrial Automation GmbH



XML View

The *XML View tab* contains a read-only view, allowing the user to have an XML preview of the file data items added to the namespace of the local application.





© Softing Industrial Automation GmbH

<<< File Read Action Wizard - Extended Settings File Read Action Wizard - Trigger Type >>>

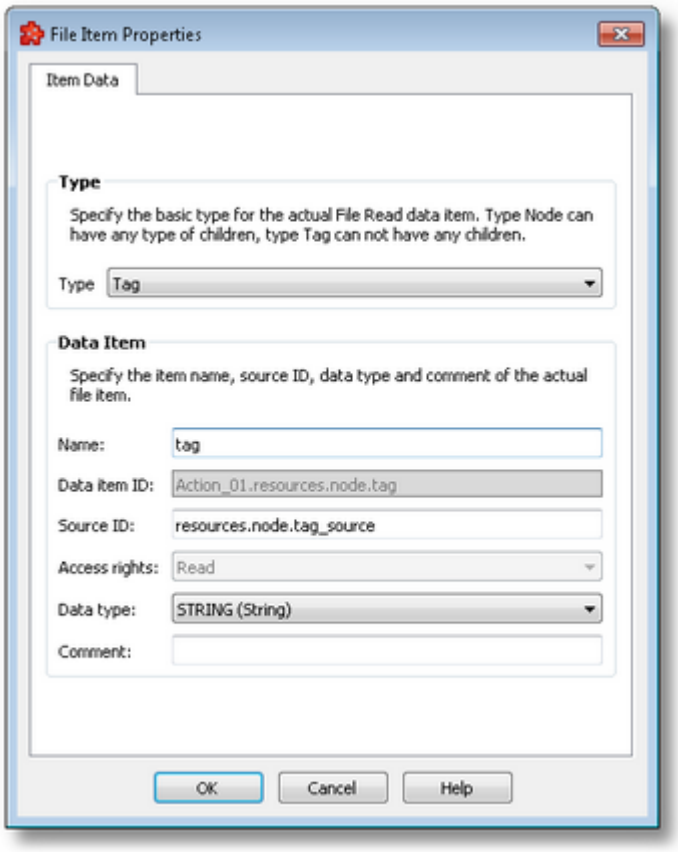
The File Items dialog allows adding new file items and modifying the existing ones.

Add a new File item

The add operation is triggered by selecting the parent node and then pressing the *Add a new item* button  , using the keyboard shortcut CTRL+A or selecting the *Add* action from the context menu. The Add action is available for the root node item and for other items of type Node.

Note: A tag cannot have any children, so Add is not available for tags.

Invoking the Add action brings up the following dialog:



The following file item properties can be selected in this dialog:

- the item class or type, which can be Node or Tag
- the file item name. It should be unique for its level in the tree. It is available for both items of type Node and Tag.

- the file item ID. It represents the full item path for the data item inside the dataFEED OPC Suite namespace.
- the source ID. It represents item ID for the file item in the external XML or XSD input data file, that is the XML tree path.
- the file item access rights. It is available only for items of type Tag, and array type items of type Node.
- the file item data type. It is available only for items of type Tag, and array type items of type Node.
- the file item comment. It can be any text. It is available only for items of type Tag.

Note: With the exception of array type file items, the items of type Node represent intermediate nodes in the tree, they are used for hierarchy purposes. The items of type Tag represent actual file items.

Note: For array type file items, the items of type Node represent the array type item containing all its array elements. The child items of the array type node parent, represent the individual array element item. This allows the user that following the import process, to add to the dataFEED OPC Suite namespace, the entire array type item or only a subset of its individual array element items.

The file items of the namespace can be of any data type covered by the OPC Specifications. The following table describes the available data types mapped to the corresponding VARIANT types from the OPC Specification.

Data type	VARIANT type	Description
I1	*VT_I1*	8 bit signed integer.
UI1	*VT_UI1*	8 bit unsigned integer.
I2	*VT_I2*	16 bit signed integer.
UI2	*VT_UI2*	16 bit unsigned integer.
I4	*VT_I4*	32 bit signed integer.
UI4	*VT_UI4*	32 bit unsigned integer.
I8	*VT_I8*	64 bit signed integer.
UI8	*VT_UI8*	64 bit unsigned integer.
R4	*VT_R4*	32 bit floating point.
R8	*VT_R8*	64 bit floating point.
BOOL	*VT_BOOL*	Boolean.
STRING	*VT_BSTR*	String.

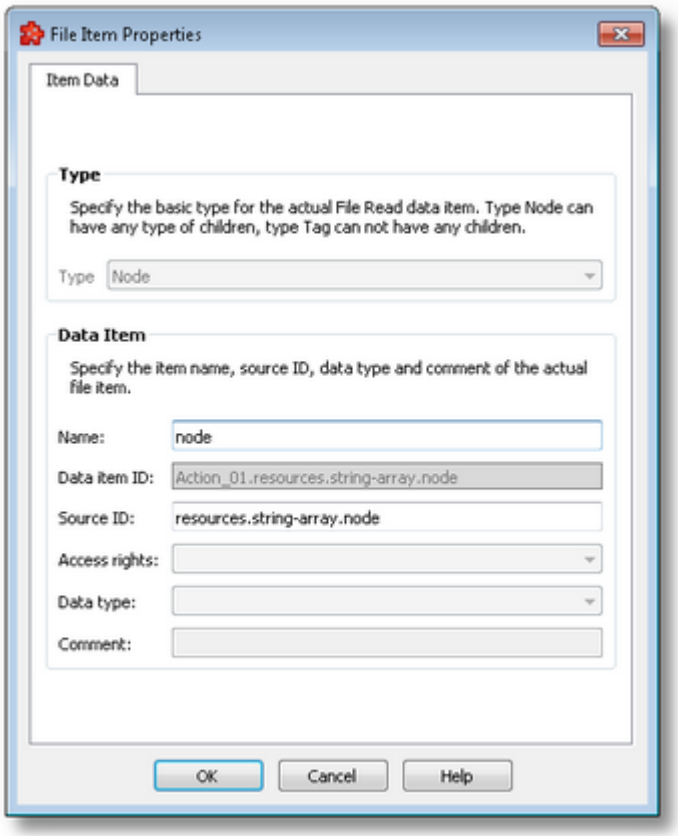
© Softing Industrial Automation GmbH

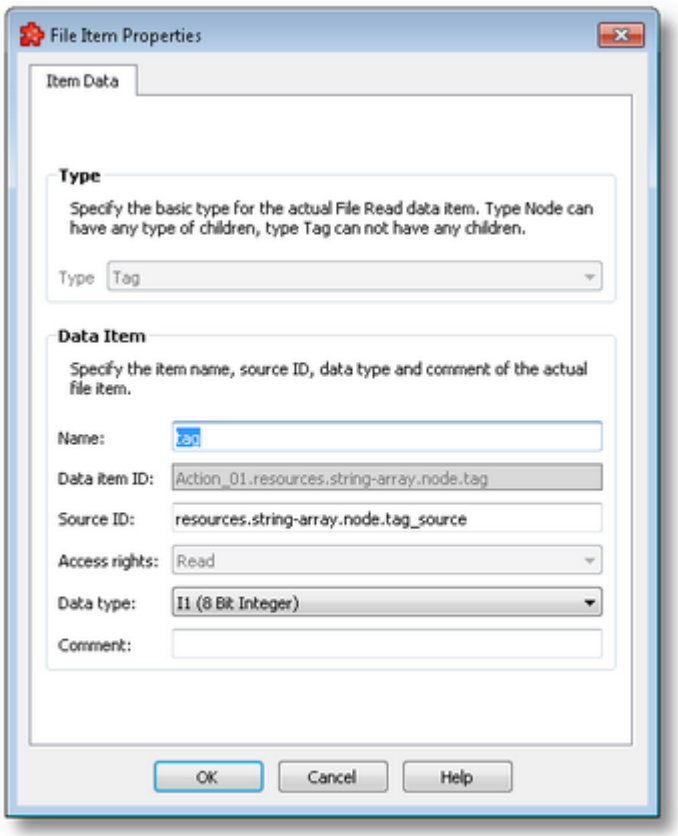
DATETIME	*VT_DATE*	Date and time.
DECIMAL	*VT_CY*	Decimal.
ANY	*VT_VARIANT*	Any data type.

Note: Arrays of fixed and variable length are also supported for these data types.

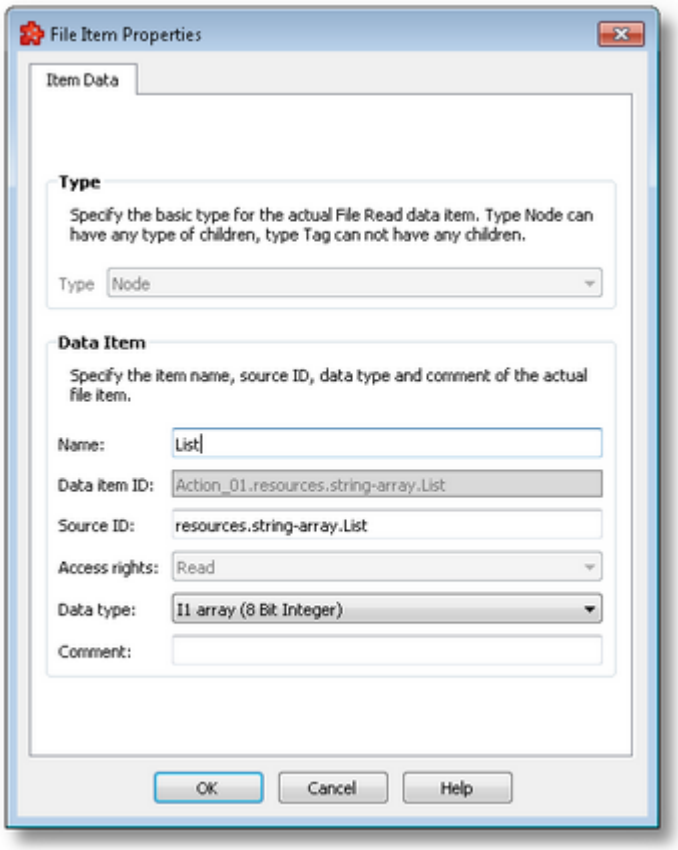
Modify a File item

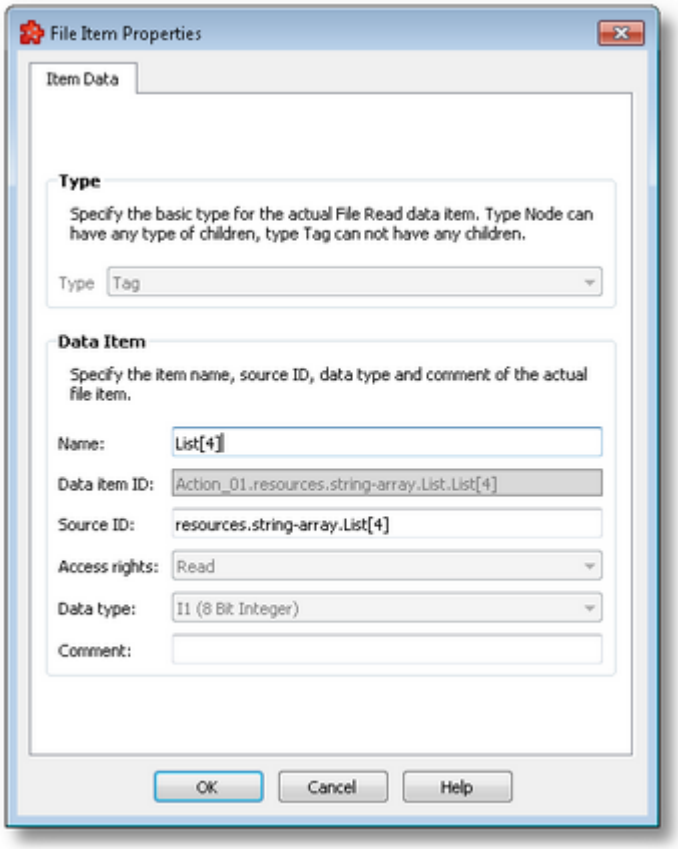
Modifying a file item is triggered by selecting the item and then pressing the Properties of the selected item button  , using the keyboard shortcut CTRL+P or selecting the Properties action from the context menu. Performing the Modify action displays the following dialog:





© Softing Industrial Automation GmbH





This dialog provides the same properties as the add item dialog.

Note: The file item class cannot be modified as the hierarchy of the items relies on it.

Note: The Modify action is triggered by double-clicking an item with no children.

Note: Node type array items are allowed to set array data type values only. If these items have child items, they will have the corresponding scalar data value set.

Note: The data type of the child items of node type array items is not allowed to be set. Their data type is set when assigning the array data type value to its parent item.

<<< File Read Action Wizard - File Items

The dataFEED OPC Suite Configurator allows the import of File Items from an external XML/XSD file to the root of the File Read data source address space.

The import action can be triggered by using the *Import from file* button  , by pressing the keyboard shortcut sequence CTRL+I or by selecting the *Import* action in the context sensitive menu.

Note: The import action is only available for the File Read data source address space root node.

Note: The names conflicts between the old and new items are automatically solved by renaming the newly imported items. They are renamed by appending a two-digit number at the end of the name of the first duplicated item.

When the import action is triggered the XML/XSD *File Item Import Wizard* is started.

▼ The File Items Import Wizard allows importing data items from XML and XSD files:

- Select the file to be imported @ Import XML/XSD Wizard - Import File Selection
- Examine the import results @ Import XML/XSD Wizard - Import Results

<<< File Read Action Wizard - File Items

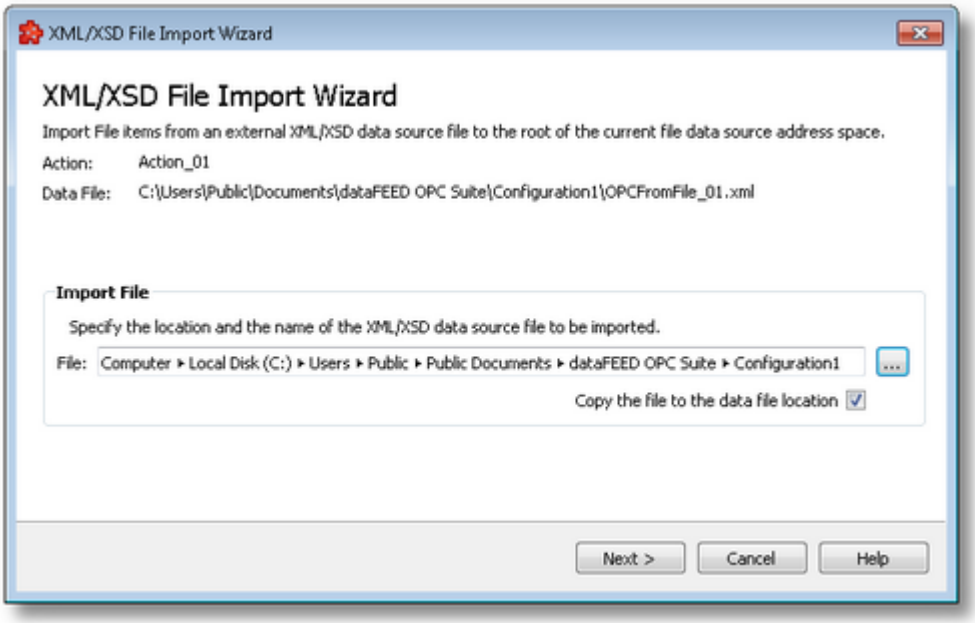
The Import File Selection wizard page allows you to choose an XML/XSD file to be imported.

© Softing Industrial Automation GmbH

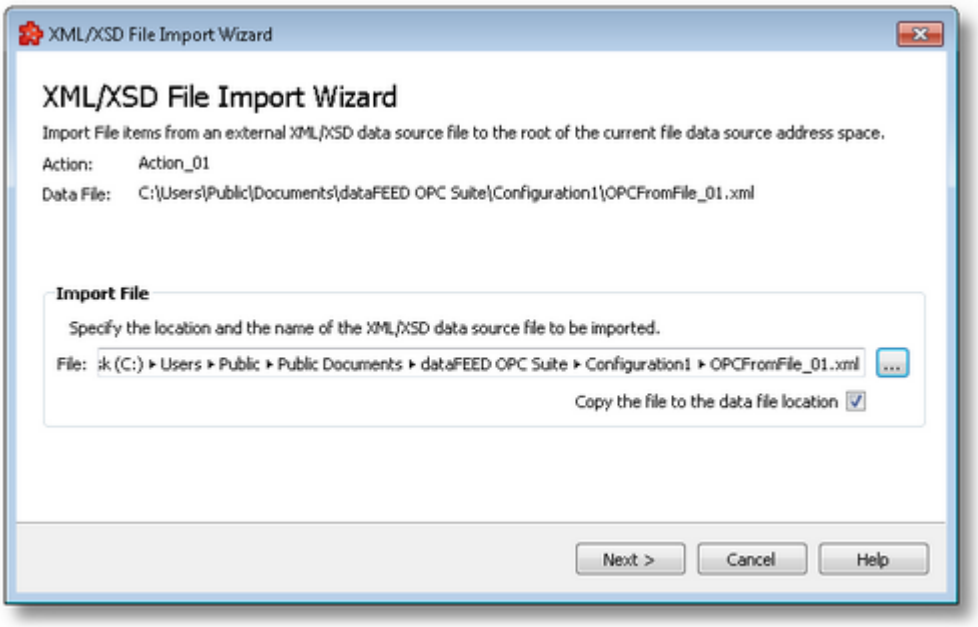
The wizard page displays the current File Read Action name along with the Data File path and name specified at the File Read Action Wizard Action Settings.

The file can be selected by manually typing its name in the *File* input box or by clicking the browse button located to the right of the input box (a standard file opening dialog will open).

The image below displays the case when the data file specified at the File Read Action Wizard Action Settings does not exist, the file path not including the name of the file. The user will need to select the XML/XSD input file. Depending on the "Copy the file to the data fie location" option activation, following the import process the XML file will be copied to the specified Data File. For XSD file import, the option of copying the file to the data file location is disabled and does not exist. The purpose of using an XSD file at import, is only for the creation of the file read data source items in the namespace of the local application.



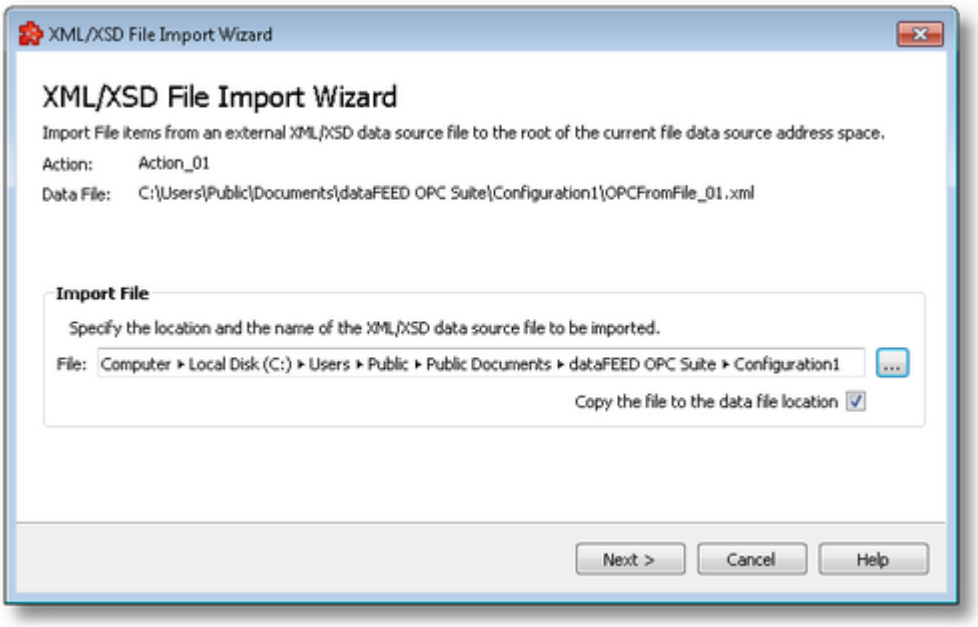
The image below displays the case when the data file specified at the File Read Action Wizard Action Settings exists, the file path including also the name of the file. The user can continue with the import of the the specified Data File by pressing the *Next* button.



The file path and name should be valid (they should point to an existing file). When you press the *Next* button the file path and name are verified. If the file identified by the specified file path and name does not exist an error message is displayed. The actual import begins if the selected file is valid. If the import takes longer to complete a waiting dialog is displayed.

<<< Import XML/XSD wizard Import XML/XSD Wizard - Import Results >>>

The Import Results wizard page displays the import results.



© Softing Industrial Automation GmbH

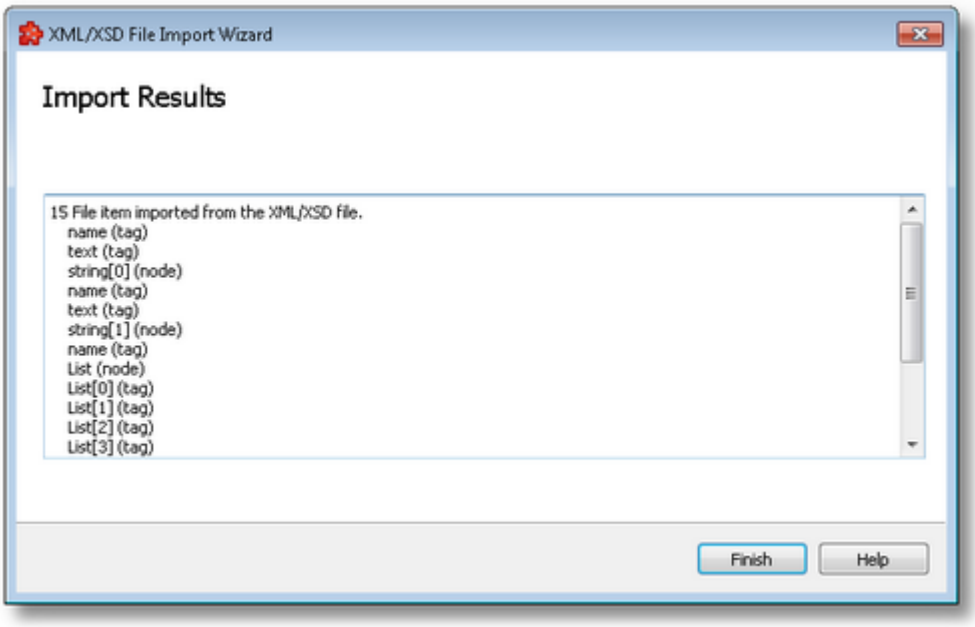
This page is the final page of the File Item Import Wizard. A back button is not provided as the import action cannot be undone. The import can be successful or it can fail:

1. After a successful import the results page contains the following information:

   - The absolute path of the imported file.
   - The number of the file items imported. This number includes only the items of type Tag, the items of type Node are not counted.
   - The names of the file items imported. The name is displayed for items of type Tag and Node.
   - The confirmation message that the import was successful.

2. After a failed import the results page contains the following information:

   - The invalid file line at which the import stopped.
   - The reason why the line is considered invalid.
   - The confirmation message that the import failed.



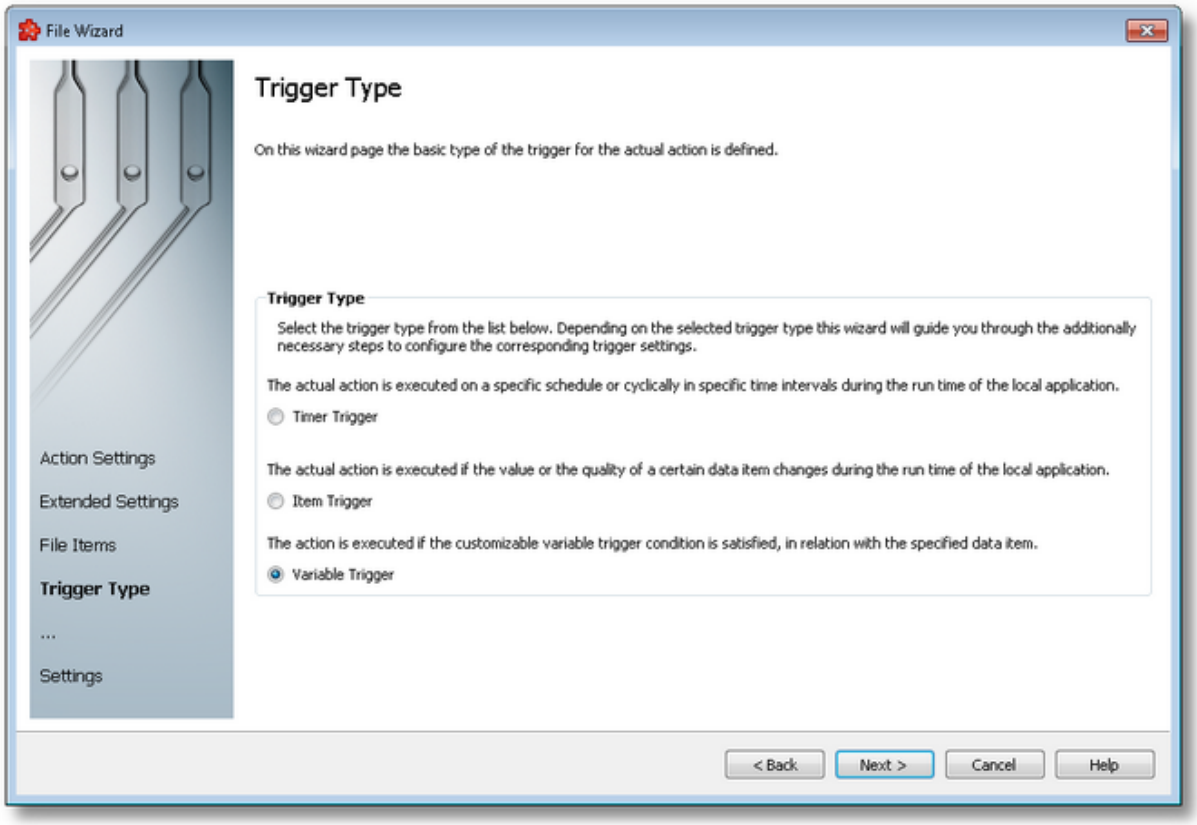
When the import wizard is closed after a successful import, the new items are added to the current File Read data source address space, as a branch of the root node item.

<<< Import XML/XSD Wizard - Import File Selection

The Trigger Type wizard page provides the possibility to select the trigger type for the actual action. The following trigger types are available:

- Timer Trigger - This trigger is fired at a specified time or interval.

- Item Trigger - This trigger is fired when a data change for the selected item occurred. A data change means that the value or the quality of the item changed.
- Variable Trigger - This trigger is fired when the variable trigger condition is satisfied, in relation with the specified data items.



To proceed with the wizard and continue to define the selected type of trigger use the *Next* button, to go back to the previous wizard page use the *Back* button, or discard the wizard by pressing the *Cancel* button.

Depending on the selected *Trigger Type* the next wizard page will be:

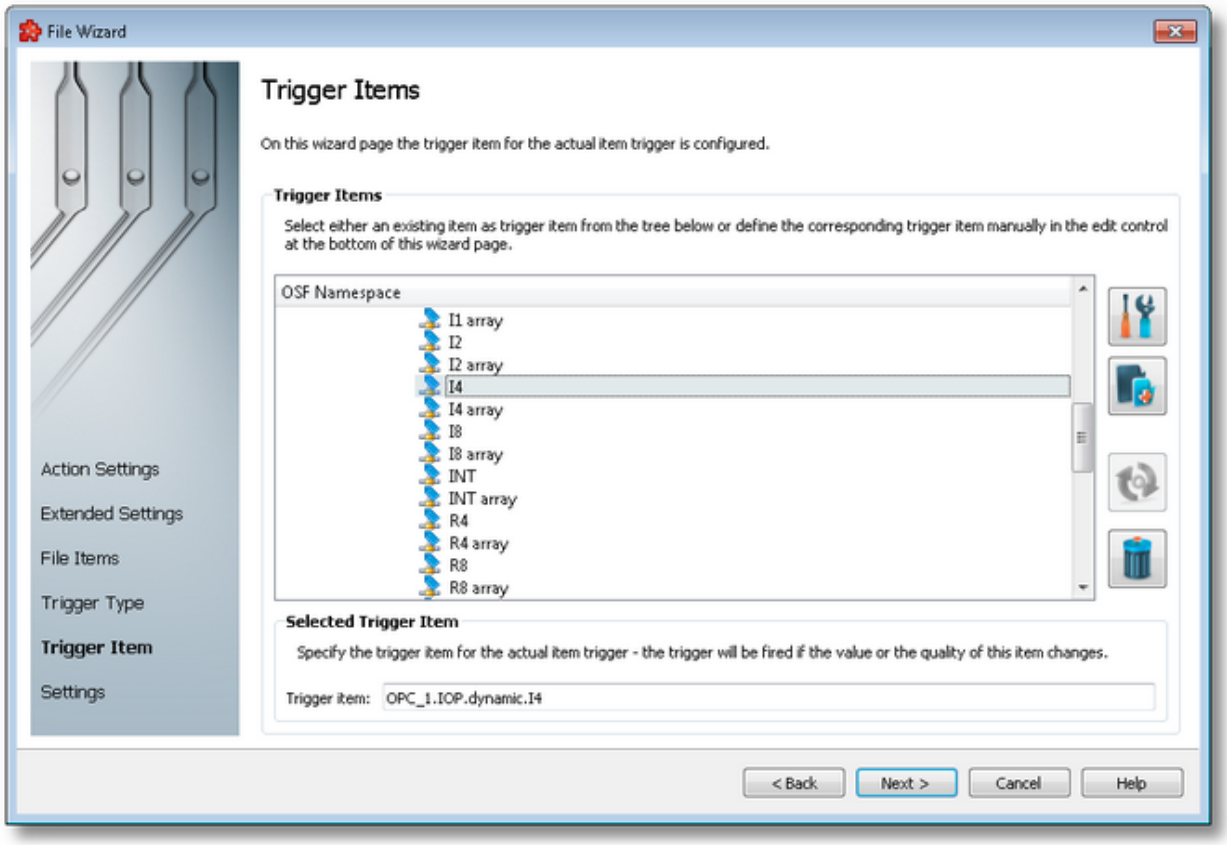
- File Read Action Wizard - Timer Selection - if the selected trigger type is *Timer Trigger*,
- File Action Wizard - Trigger Item - if the selected trigger type is *Item Trigger*,
- File Read Action Wizard - Variable Trigger - if the selected trigger type is *Variable Trigger*

The description of the File Read Action Wizard will continue with the wizard pages for configuring an *Item Trigger*.

<<< File Read Action Wizard - File Items File Read Action Wizard - Trigger Item >>>

© Softing Industrial Automation GmbH

The *Trigger Item* wizard page is shown if the selected trigger type is *Item Trigger*. Here you can configure the actual trigger item.



This wizard page contains two separate sections:

- *Trigger Items* section, which allows selecting an item trigger or managing the Local Namespace tree.
- *Selected Trigger Item* section, which shows the item ID of the selected trigger item and also allows typing an item ID, thereby selecting this item in the above list.

Trigger Items section

The entire namespace of the current configuration is shown as a namespace tree. The name of the root node is composed by appending the instance name to the *Local Namespace* keyword. The first level of the namespace contains:

- A node for each server connection, named by the server connection items prefix.
- A node for System Items.
- A node for Local Items.

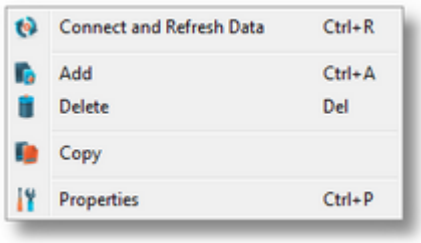
The local namespace tree offers the possibility to explore the entire namespace of the local dataFEED OPC Suite application in order to select a trigger item. After the selection is made, the *Trigger item* edit box is populated with the selected item ID.

The entire namespace of the local dataFEED OPC Suite application can be managed through this wizard page by performing certain operations using the buttons on the right, the context sensitive menu or the keyboard shortcuts:

- Open the item properties dialog using the *Properties of the selected item* button  , pressing the keyboard shortcut CTRL+P or selecting the *Properties* action from the context menu. Depending on the type of the selected item, a special Item Properties dialog opens:
    - If the selected item is under Local Items the Modify Local Item dialog is opened.
    - If the selected item is under a dynamic server connection the Modify Dynamic Server Item dialog is opened.
    - If the selected item is under a static server connection the Modify Static Server Item dialog is opened.
- Add a new item using the *Add a new item*  button, pressing the keyboard shortcut CTRL+A or selecting the *Add* action from the context menu. Depending on the type of the selected node, a special Add Item dialog opens:
    - If the selected node is under Local Items the Add Local Item dialog is opened.
    - If the selected node is under a dynamic server connection the Add Dynamic Server Item dialog is opened.
    - If the selected node is under a static server connection the Add Static Server Item dialog is opened.
- Copy the selected item to clipboard by selecting the *Copy* action from the context menu.
- Paste the previously copied item from clipboard by selecting the *Paste* action from the context menu. Items originating from the Local Items cannot be pasted under server connections, and vice-versa; an error message will be displayed in both cases. If an item is pasted in a static (filtered) server connection and the item does not exist on the server an error message is displayed.
- Connect to the server and refresh the data items by using the *Refresh data items*  button, pressing the keyboard shortcut CTRL+R or selecting the *Connect and Refresh Data* action from the context menu. The namespace of the selected server connection is recreated; if manually added items not referenced in the actual configuration exist, those items are deleted from the namespace.
- Delete the selected item and all of its child items by using the Delete data items button  , pressing the DELETE / DEL key or selecting the Delete action from the context menu. Items which are included in any Data Processing / Data Destination add-ons will not be deleted but an error message will be displayed instead.

An example of the context menu can be seen below:

© Softing Industrial Automation GmbH



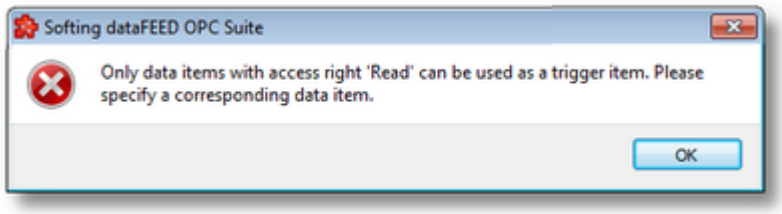
Selected Trigger Item section

The edit box in this section is populated with the item ID after an item from the Local Namespace tree is selected.

You can also manually type the ID of the item in the *Trigger item* edit box and thus have the above tree automatically select a valid server item.

Note: If the item ID is typed and the specified item exists in the Local Namespace, it will be used as the trigger item. The item will only be created if it could not be localized in the Local Namespace. In that case the newly created item will be used as the trigger item.

Note: If the item corresponding to the typed item ID could not be created, an error message will be displayed requesting the input of a new, valid item.

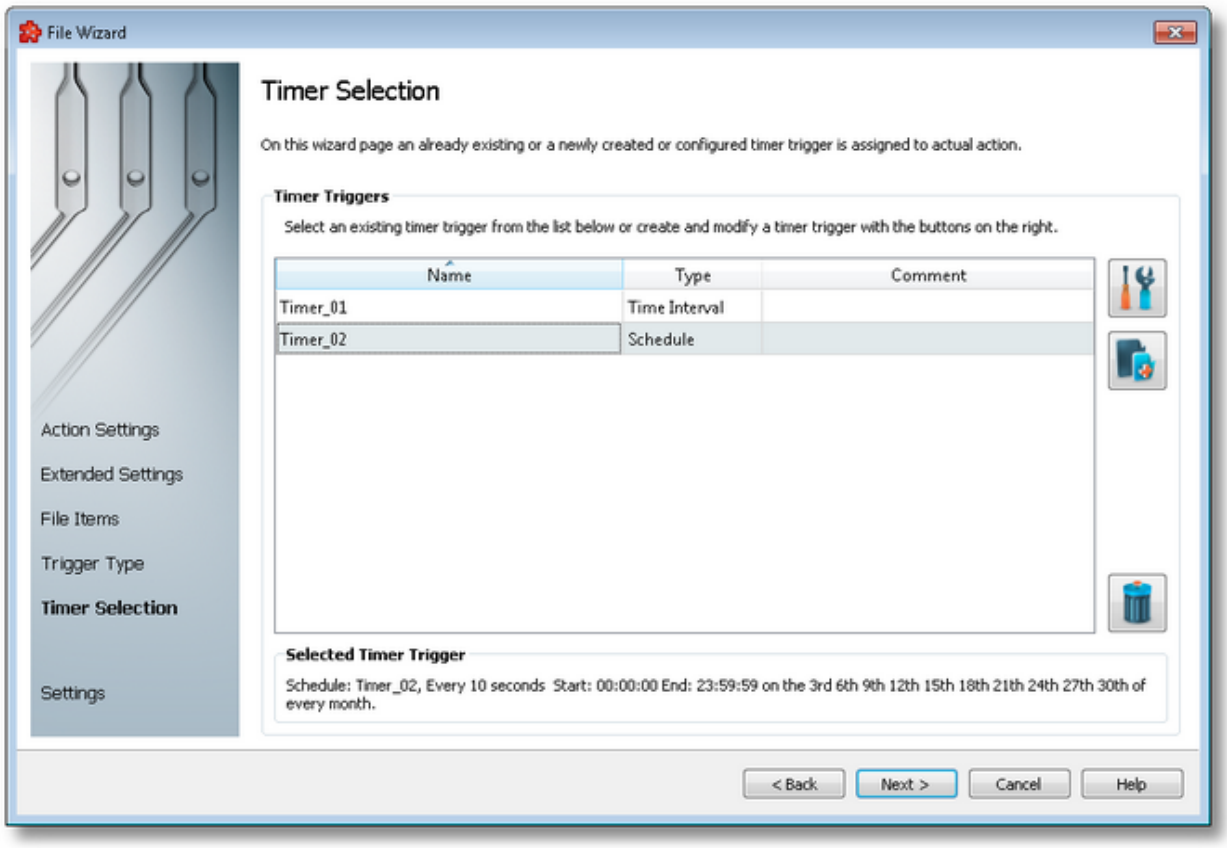


Attention: If the item corresponding to the typed item ID is valid and resides under a static server connection, the respective item will be created in the filter file definition.

To proceed with the wizard and continue to define the action settings use the *Next* button. To go back to the previous wizard page use the *Back* button, or exit the wizard and discard all the changes by clicking the *Cancel* button.

<<< File Read Action Wizard - Trigger Type File Read Action Wizard - Action Settings >>>

The Timer Selection wizard page is shown only if the selected trigger type is *Timer Trigger*. Here you can configure the actual timer trigger.

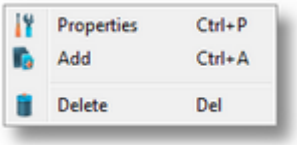


The current page allows selecting a previously defined timer trigger and managing the timer list by using the buttons on the right, the context sensitive menu or the keyboard shortcuts:

- Change the properties of the selected timer by using the *Properties* button  , pressing the keyboard shortcut CTRL+P or selecting the *Properties* action from the context menu.
- Add a new timer by using the Add timer trigger button  , pressing the keyboard shortcut CTRL+A or selecting the Add action from the context menu.
- Delete the selected timer trigger by using the Delete selected timer trigger button  , pressing the DELETE / DEL key or selecting the Delete action from the context menu.

Note: If a timer trigger has associated actions it cannot be deleted.

An example of the context sensitive menu can be seen below:



© Softing Industrial Automation GmbH

Note: Information about the selected timer is shown in the *Selected Timer Trigger* area at the bottom of the page.

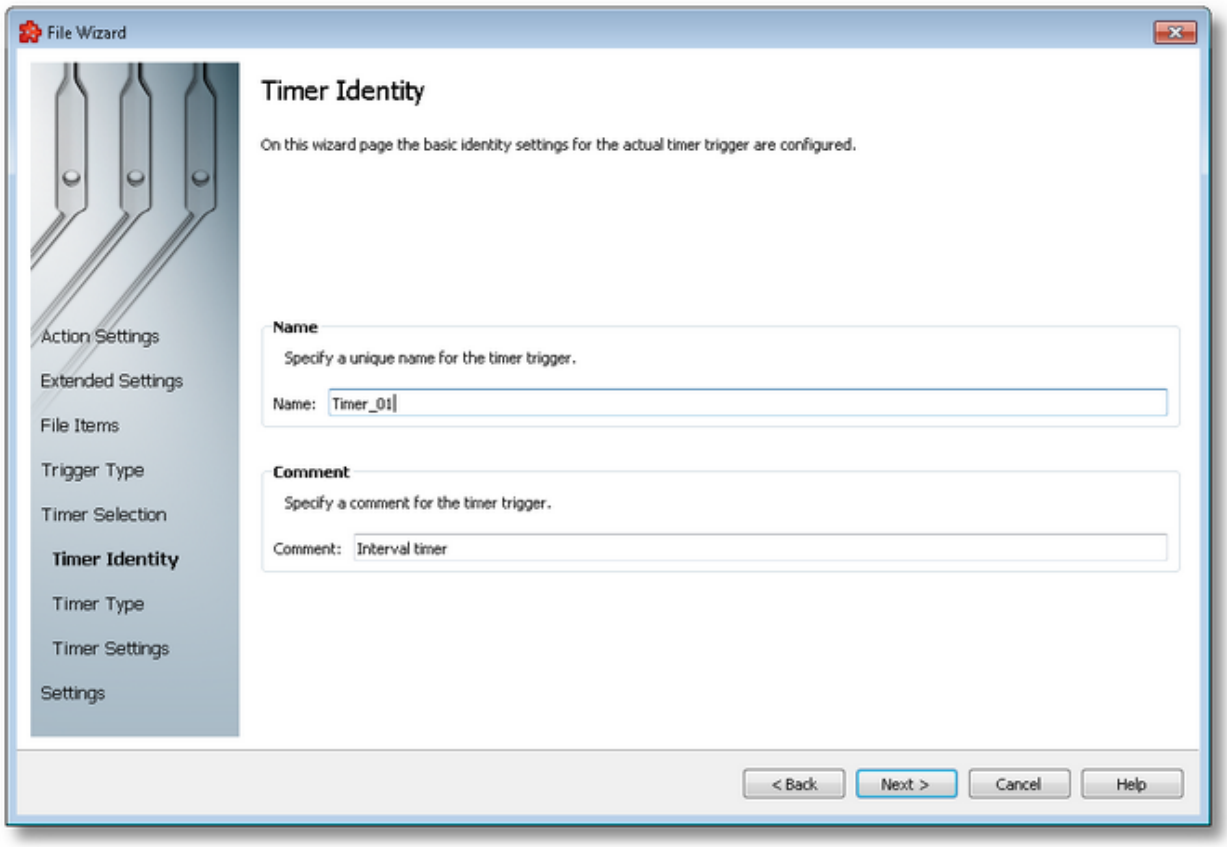
A sub-wizard that allows adding or modifying timer triggers can be launched by selecting to change an existing timer trigger or to add a new one. The first page of this sub-wizard is File Read Action Wizard - Timer Identity.

To proceed with the *File Action Wizard* and define the action settings press the *Next* button. To return to the previous wizard page use the *Back* button, or exit the wizard by clicking the *Cancel* button.

The wizard will continue straight to the *Action Settings* page as an already defined timer is selected.

<<< File Read Action Wizard - Trigger Type File Read Action Wizard - Action Settings >>>

The Timer Identity wizard page provides the possibility to assign a name and a comment to the timer trigger.

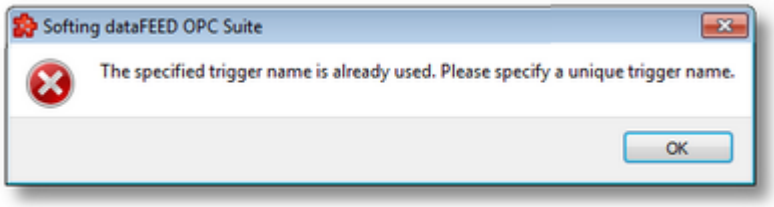


Note: When adding a new timer trigger, the timer name will be automatically generated according to the format: *Timer_number*, where the *number* part is an auto-incremented integer.

When editing an existing timer, the timer's defined name and comment will be filled in the two edit boxes allowing you to modify these attributes.

Attention: The name of the *Timer Trigger* must be unique among all triggers and cannot be empty.

If the newly added timer trigger has the same name as an existing trigger, an error message box will be displayed and a new trigger name must be provided:



The *Comment* field of the *Timer Identity* wizard page is optional; a Timer Trigger can also be added without any comment.

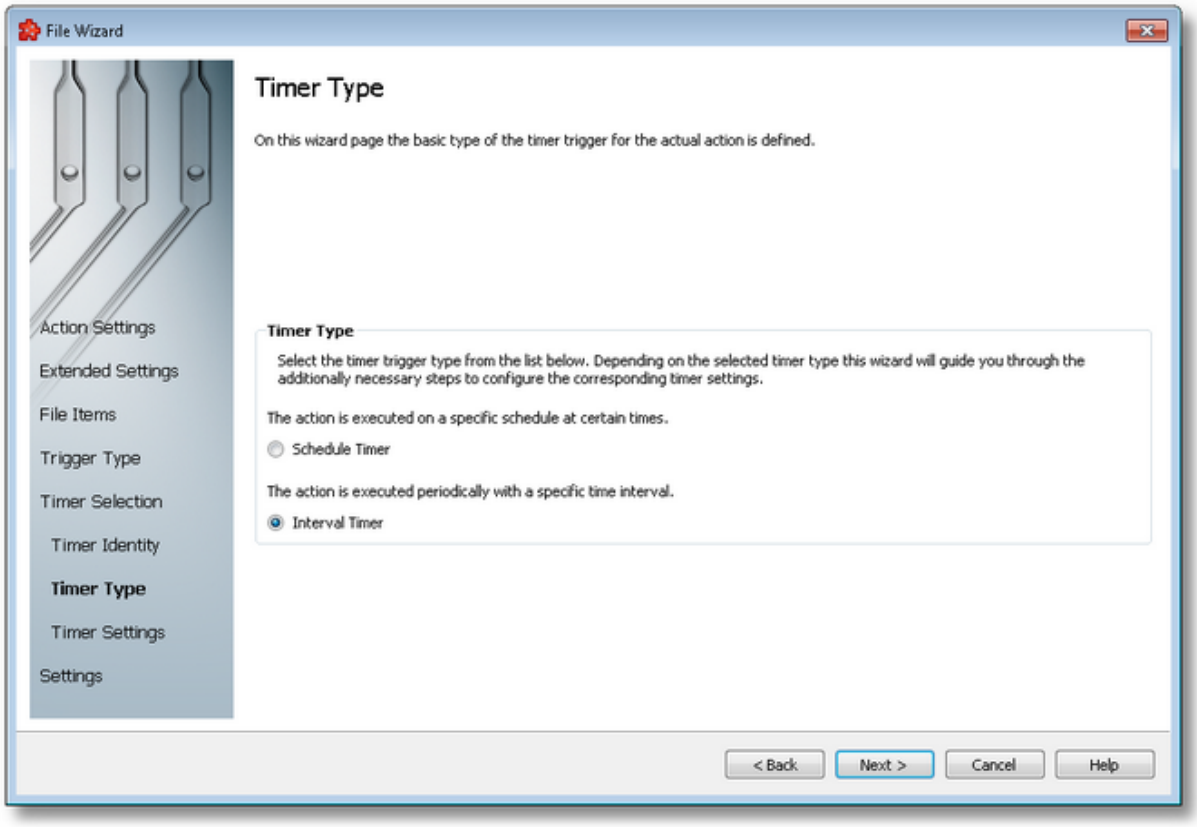
Continue to the next wizard page to choose the timer type by using the *Next* button, go back to the previous wizard page to select another timer trigger by pressing the *Back* button, or cancel the entire *File Action Wizard* by pressing the *Cancel* button.

<<< File Read Action Wizard - Timer Selection File Read Action Wizard - Timer Type >>>

This wizard page allows choosing the type of the timer trigger. The following types of timer are available:

- *Interval Timer*
- *Schedule Timer*

© Softing Industrial Automation GmbH



Interval Timer

The Interval Timer type allows defining a specific interval of time (in milliseconds) between the timer fire events. The time interval can be configured on the File Read Action Wizard - Interval Timer Settings wizard page.

Schedule Timer

The Schedule Timer type allows defining a specific date and time when the timer should fire. The schedule can be configured on the File Read Action Wizard - Schedule Timer Settings wizard page.

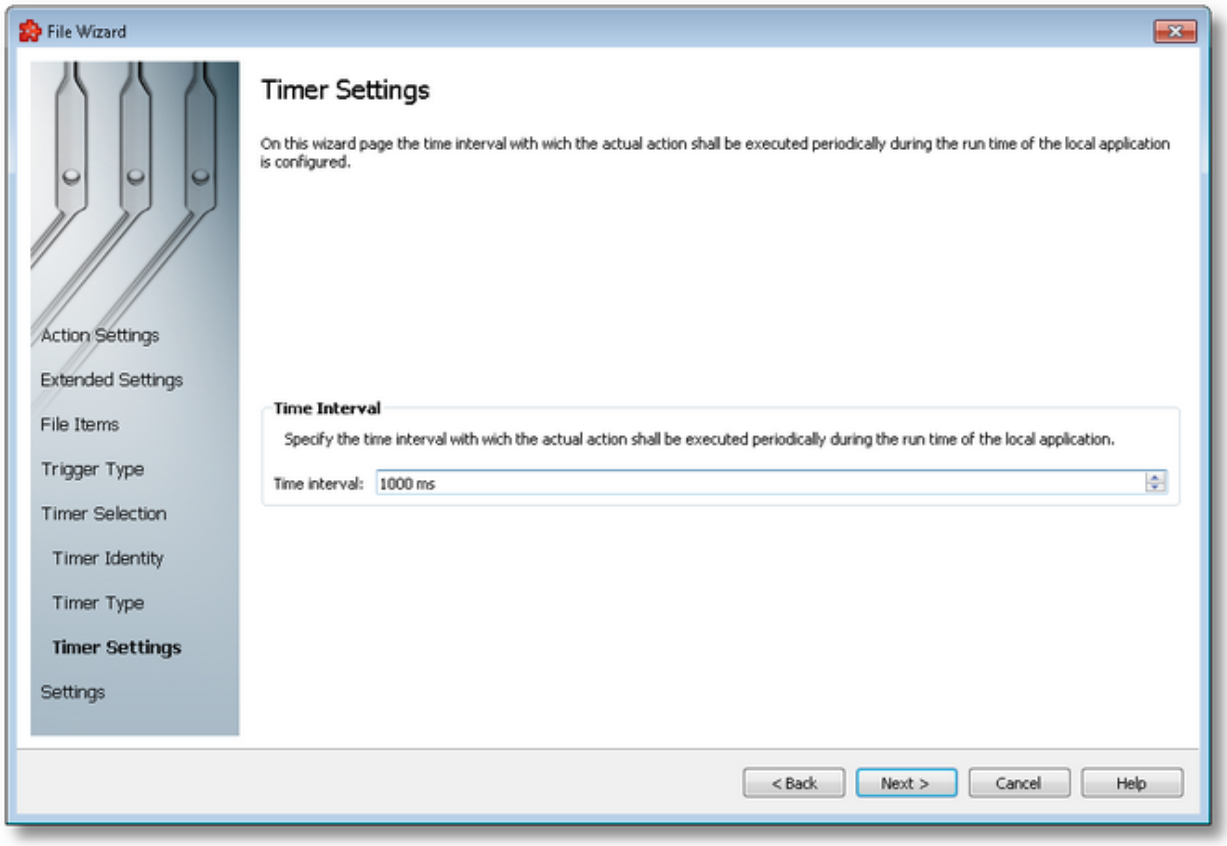
Note: The *Timer Type* of an existing *Timer Trigger* cannot be changed.

To continue defining the specific timer properties press the *Next* button, to return to the *Timer Identity* wizard page use the *Back* button, or exit this sub-wizard (thereby aborting the *File Action Wizard*) by pressing the *Cancel* button.

The description of the File Action Wizard will continue with the wizard pages for configuring a *Schedule Timer*.

<<< File Read Action Wizard - Timer Identity File Read Action Wizard - Schedule Timer Settings >>>

The Interval Timer Settings wizard page allows configuring the time interval at which the current action should be (periodically) executed.



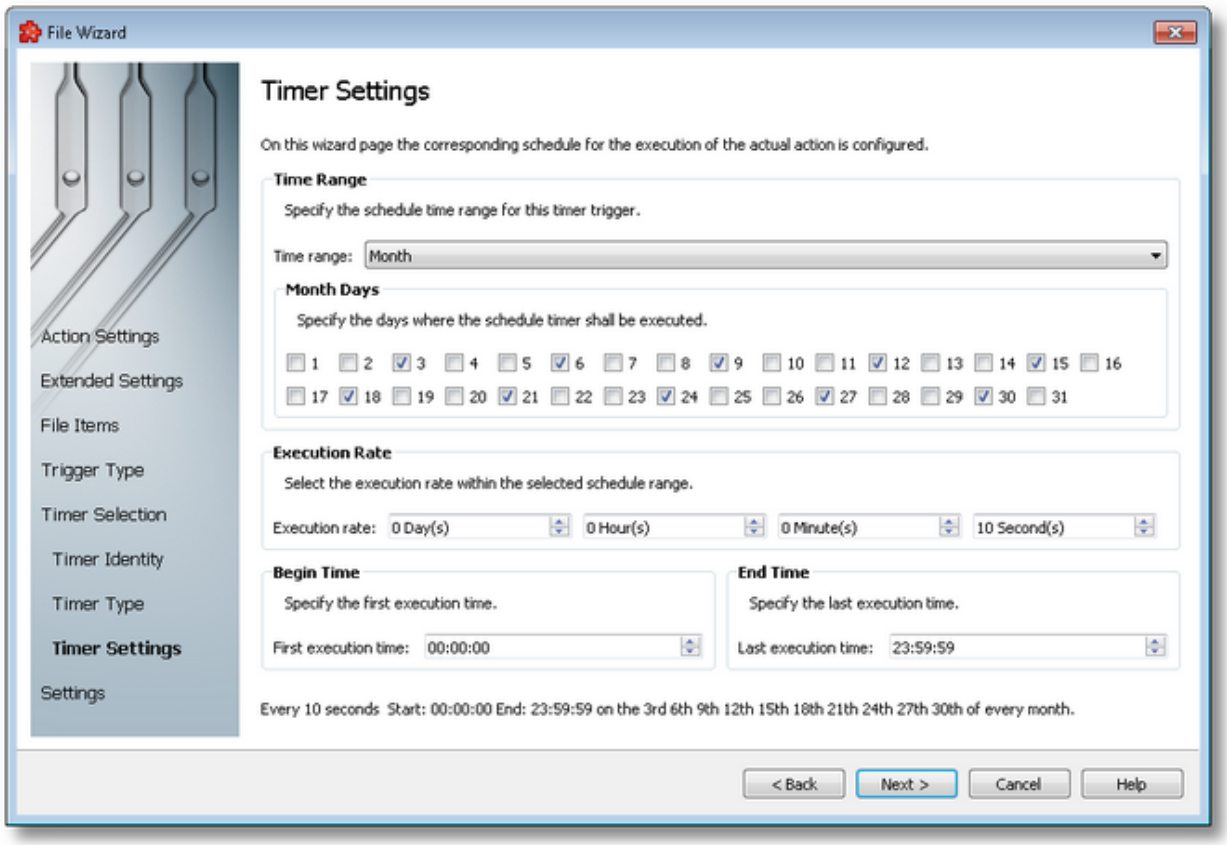
The time interval value is measured in milliseconds and ensures that the action will be triggered at the time interval defined in the spin-box.

Continue to the *Timer Selection* wizard page by pressing the *Next* button, return to the previous one by using the *Back* button, or exit the wizard by pressing the *Cancel* button.

<<< File Read Action Wizard - Timer Type File Read Action Wizard - Timer Selection >>>

The *Schedule Timer Settings* wizard page is a more complex page compared to the *Interval Timer Settings* wizard page and allows defining the corresponding schedule for the execution of the trigger action:

© Softing Industrial Automation GmbH



The page has four sections which identify the schedule:

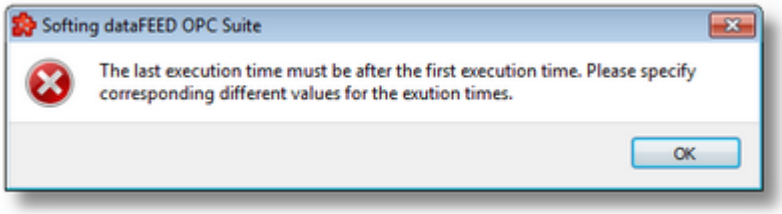
- *Time Range*: Provides the possibility to choose one of the following schedule ranges: *month*, *week*, *day*, *hour* or *minute*. When the *month* or the *week* ranges are selected the panel below the selection is populated with the month and week days, respectively.
- *Execution Rate*: Sets the execution rate of the current action in the previously-defined *Time Range* (e.g. every X day(s), every Y hour(s), every Z minute(s), every T second(s)).
- *Begin Time*: Sets the time the current action should begin execution.
- *End Time*: Sets the time the current action should stop execution.

According to the *Time Range* the *Begin Time* and the *End Time* fields will allow typing:

- Seconds (SS) if the *Time range* is *Minute*.
- Minutes and seconds (MM:SS) if the *Time range* is *Hour*.
- Hours, minutes and seconds (HH:MM:SS) if the *Time range* is *Day* or greater (*Week*, *Month*).

Note: A short review of the configured timer schedule is shown at the bottom of this wizard page.

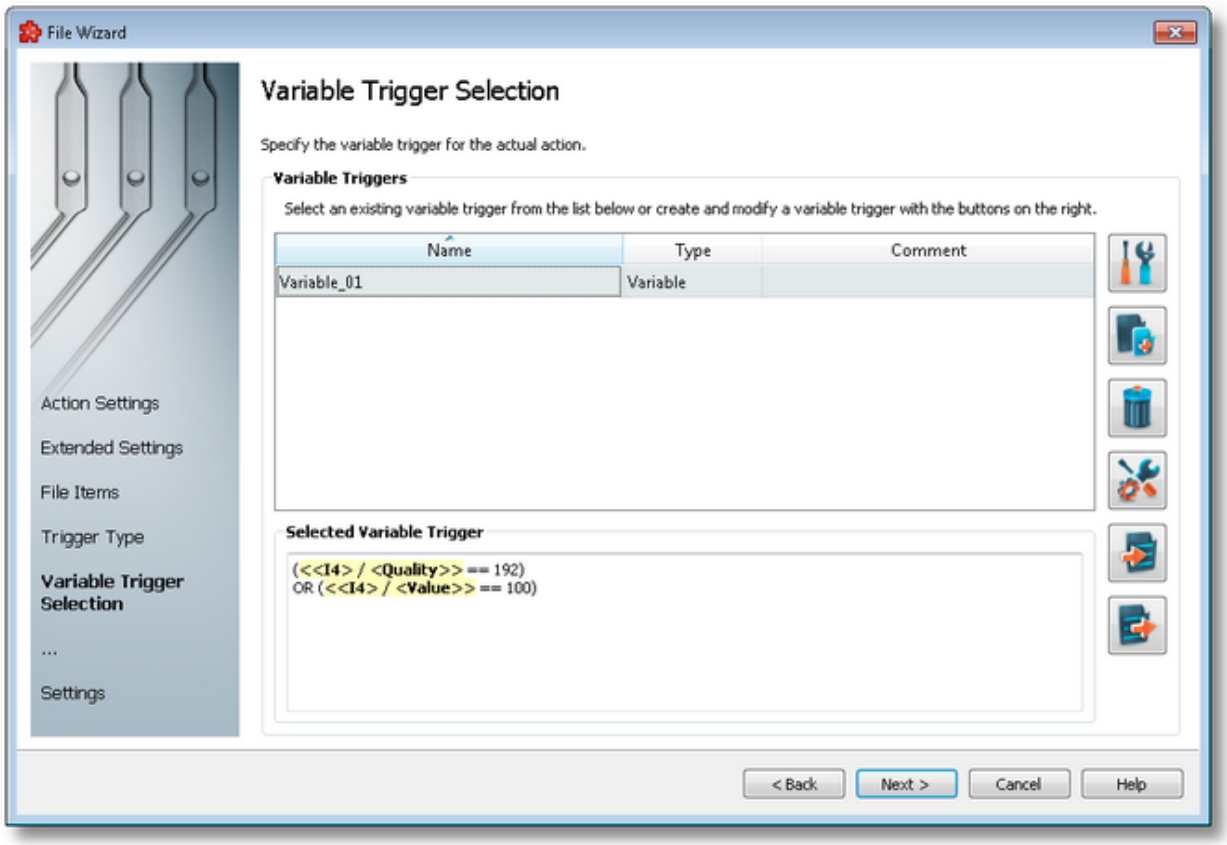
If the defined schedule is not valid, an error message dialog will appear naming the error that occurred.



Continue to the *Timer Selection* wizard page using the *Next* button, return to the previous wizard page using the *Back* button or exit the wizard pressing the *Cancel* button.

<<< File Read Action Wizard - Timer Type File Read Action Wizard - Timer Selection >>>

The Variable Trigger Selection wizard page is shown only if the selected trigger type is *Variable Trigger*. Here you can configure the actual variable trigger.



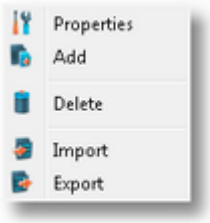
The current page allows selecting a previously defined variable trigger and managing the trigger list by using the buttons on the right, the context sensitive menu or the keyboard shortcuts:

© Softing Industrial Automation GmbH

- Change the properties of the selected variable trigger by using the *Properties* button  , pressing the keyboard shortcut CTRL+P or selecting the *Properties* action from the context menu.
- Add a new variable trigger by using the *Add new variable trigger* button  , pressing the keyboard shortcut CTRL+A or selecting the Add action from the context menu.
- Delete the selected variable trigger by using the *Delete selected variable trigger* button  , pressing the DELETE / DEL key or selecting the Delete action from the context menu.
- Modify the timestamp format settings for all condition expressions in the entire configuration using the *Timestamp format settings for all condition expressions* button  or pressing the keyboard shortcut CTRL+D.
- Import a new variable trigger by using the *Import new variable trigger* button  , or selecting the Import action from the context menu.
- Export a selected variable trigger by using the *Export selected variable trigger* button  , or selecting the Export action from the context menu.

Note: If a variable trigger has associated actions it cannot be deleted.

An example of the context sensitive menu can be seen below:



Note: Information about the selected variable trigger is shown in the *Selected Variable Trigger* area at the bottom of the page.

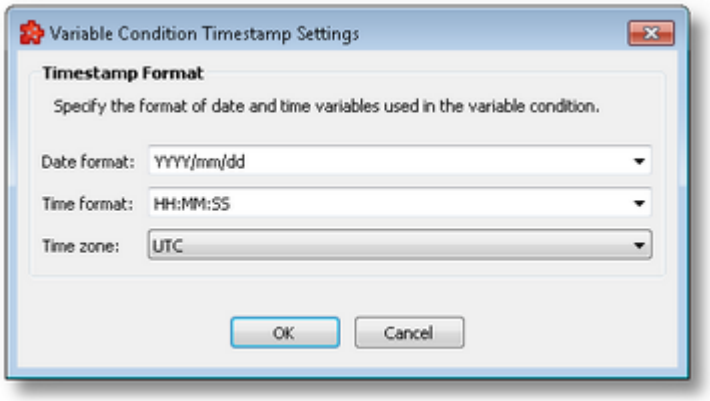
Variable Condition Timestamp Settings

Time format specifies the format in which the timestamps will be converted. Also, *Date format*, but for the date part of the timestamp values.

The default value for the date/time format is "YYYY-mm-dd HH:MM:SS" as seen in the screenshot above.

Note: At least one of the formats need to be specified. If both Time format and Date format values are empty, an error message will be displayed to the user and this dialog will not pass validation.

Time Zone, handles the source localization time: UTC (Universal Time Coordinated) or local time (local time localization). The default time zone is UTC.



These expressions may be used for the date and time format:

Expression	Description
YYYY	the year as four digit number
mm	the month as number with a leading zero (01-12)
dd	the day as number with a leading zero (01 to 31)
HH	the hour with a leading zero (00 to 23, even with AM/PM display)
MM	the minute with a leading zero (00 to 59)
SS	the whole second with a leading zero where applicable (00 to 59)
FFF	the fractional part of the second, to millisecond precision, including trailing zeroes where applicable (000 to 999)

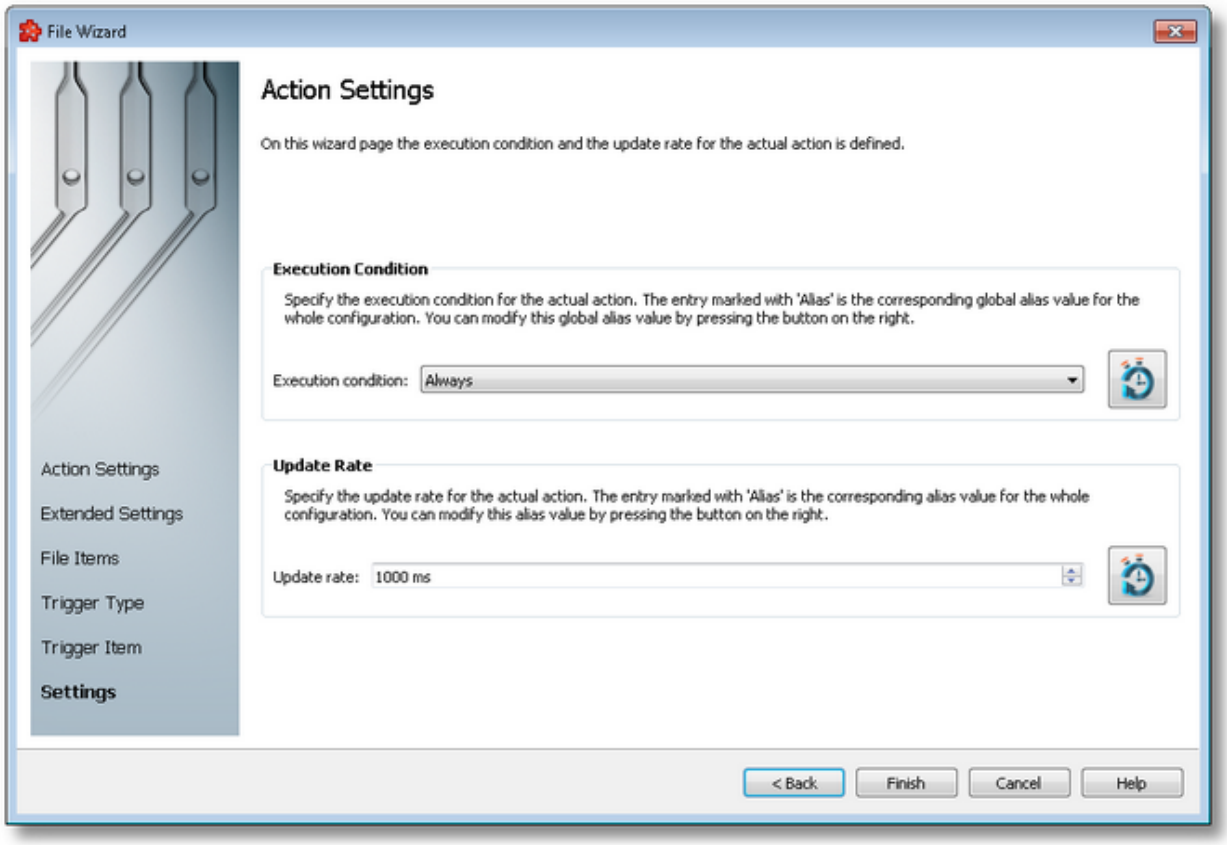
A sub-wizard that allows adding or modifying variable triggers can be launched by selecting to change an existing variable trigger or to add a new one. The first page of this sub-wizard is Variable Trigger Wizard - Trigger Identity.

Click the *Finish* button to finish the wizard, or use the *Back* button to return to the previous wizard page. To exit the wizard click the *Cancel* button.

<<< File Read Action Wizard - Trigger Type File Read Action Wizard - Action Settings >>>

© Softing Industrial Automation GmbH

The Action Settings wizard page allows defining the execution condition and the subscription update rate for the current file action.



The wizard page is divided into two sections: *Execution Condition* and *Update Rate*.

Execution Condition

Each statement has an execution condition which is evaluated by the local dataFEED OPC Suite application. The action is only performed when the evaluation is successful. If the evaluation fails then the statement is ignored.

The execution conditions allow reacting to server communication errors, executing special statements or preventing the execution of other statements.

The execution condition can be selected directly from the drop down combo box. The initial value is *Always*.

Changing the global default execution condition can be done using the Configure global alias and initial values button  or pressing the keyboard shortcut CTRL+D.

Update Rate

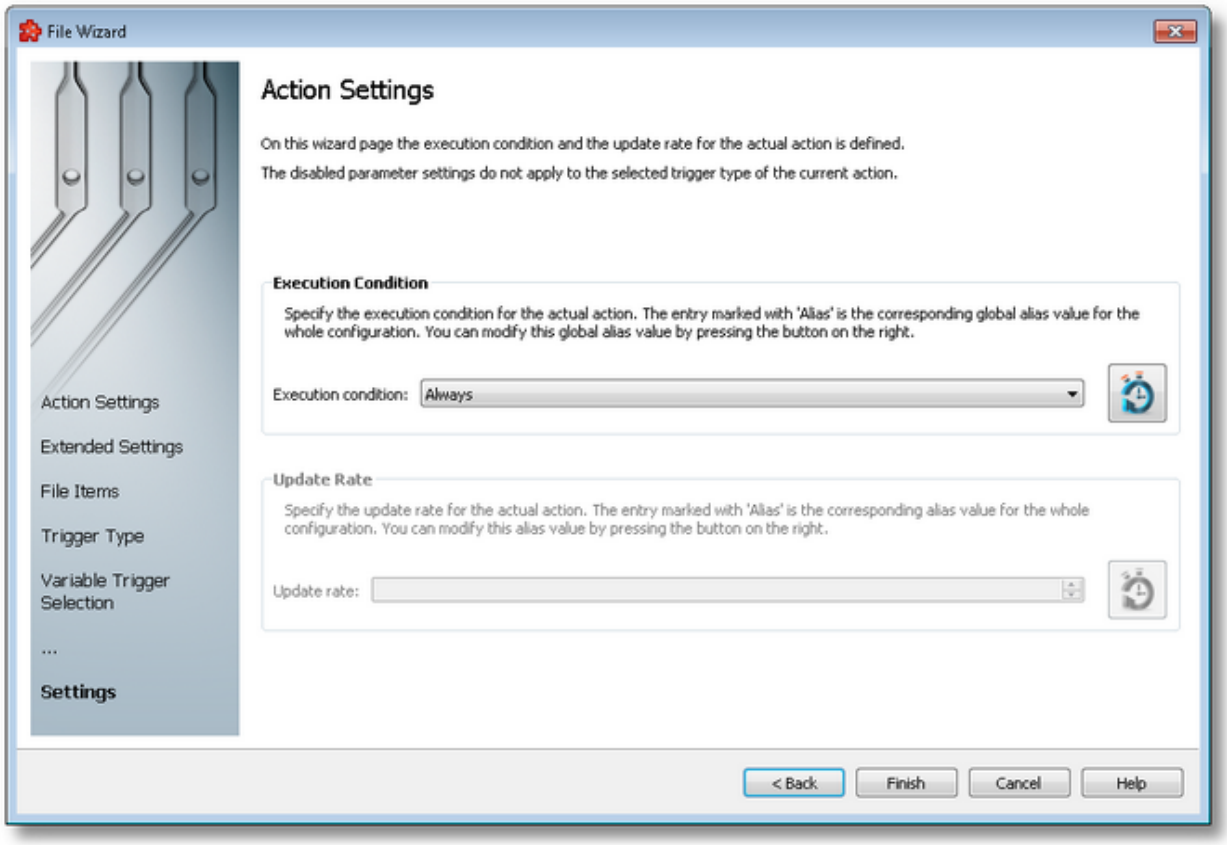
The *Update Rate* specifies the execution rate of the current file action.

The update rate can be typed or selected from the spin box. The initial value is *1000 ms*.

Changing the global default update rate can be done using the Configure global alias and initial values button  or pressing the keyboard shortcut CTRL+D.

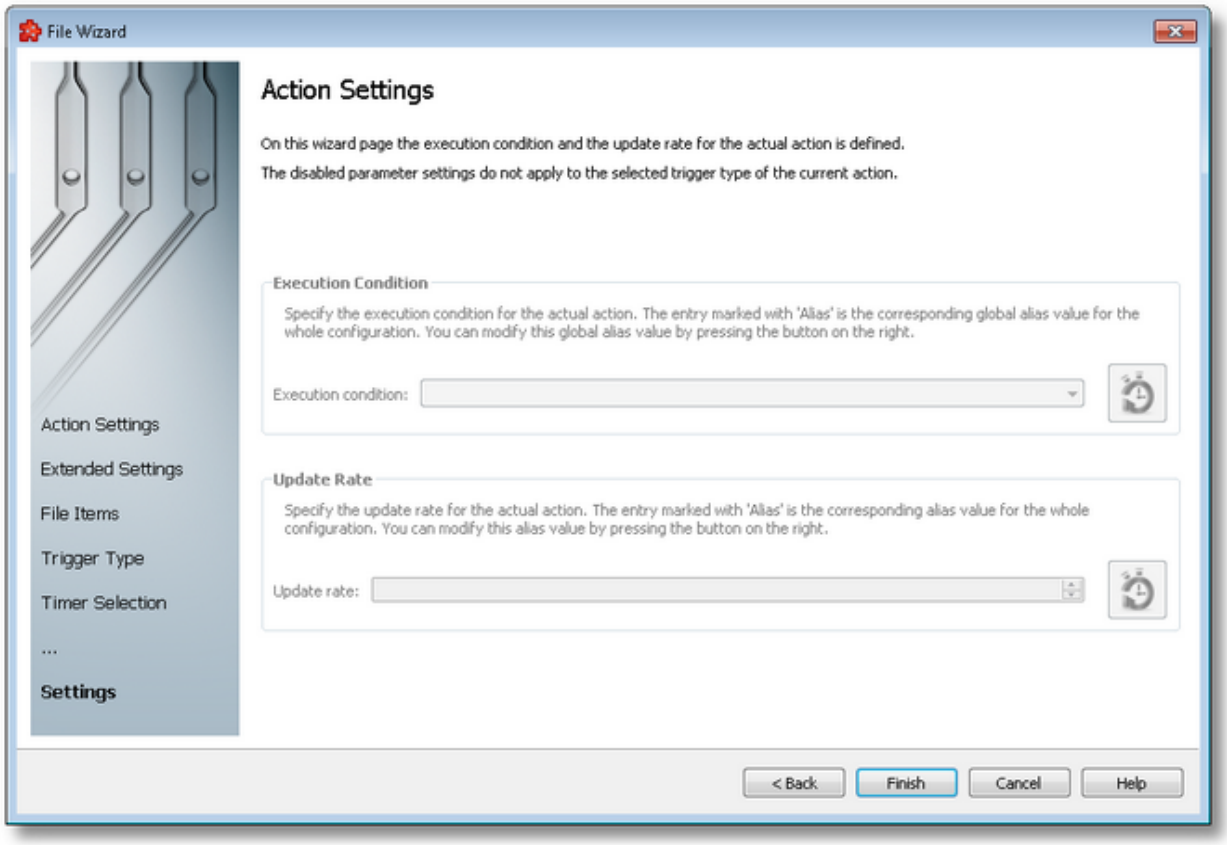
If the selected trigger type for the actual action is Item Trigger, both the Execution Condition and the Update Rate parameter settings are enabled and available to the user, as shown by the image above. Both parameter settings apply to the Item Trigger type triggered action.

If the selected trigger type for the actual action is Variable Trigger, the Execution Condition parameter setting is enabled and available to the user, and the Update Rate parameter setting is disabled, as shown by the image below. Only the Execution Condition parameter setting applies to the Variable Trigger type triggered action.



If the selected trigger type for the actual action is Timer Trigger, neither the Execution Condition nor the Update Rate parameter setting is enabled and available to the user, as shown by the image below. None of these parameter settings apply to the Timer Trigger type triggered action.

© Softing Industrial Automation GmbH

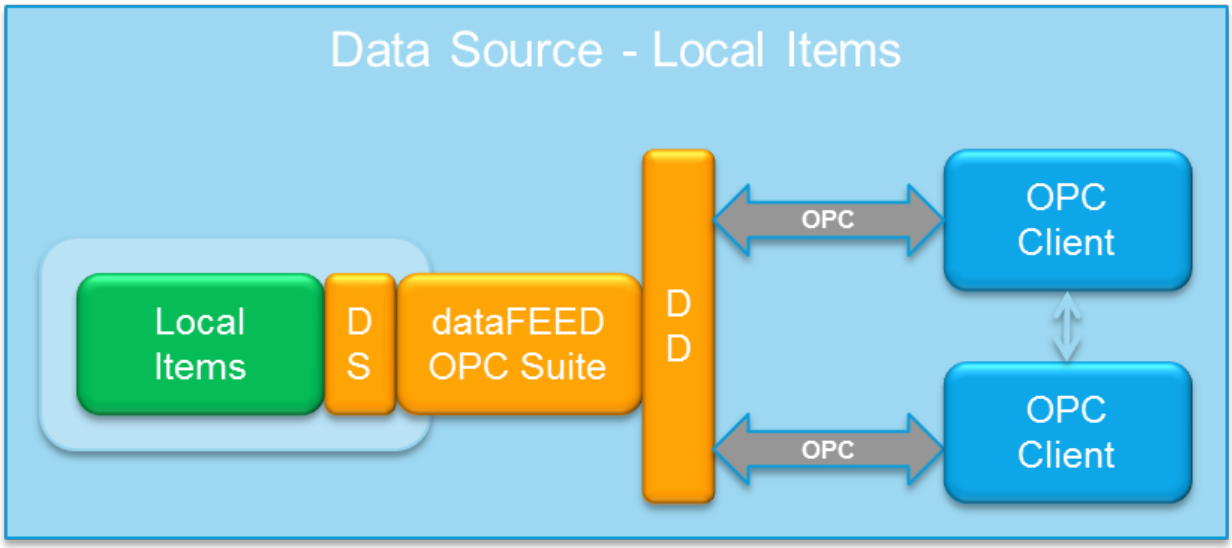


Complete the *File Action Wizard* pressing the *Finish* button, review the past wizard pages using the *Back* button, or cancel the wizard thus discarding all the changes by pressing the *Cancel* button.

<<< File Read Action Wizard - Trigger Type

7.3.1.14 Local Items

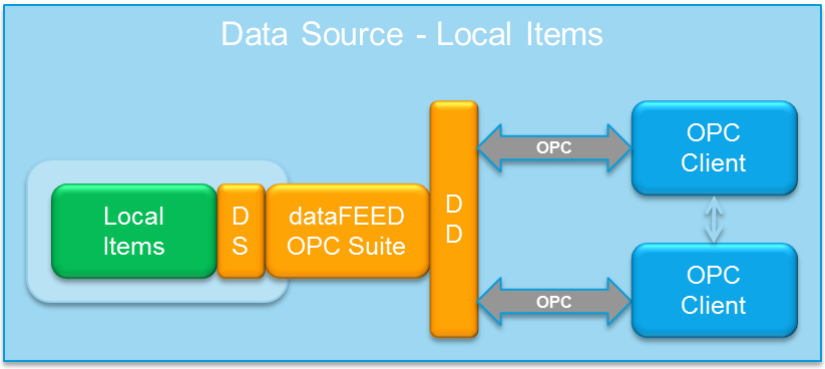
This section of the Softing dataFEED OPC Suite Online Help describes how to configure Local Items within Softing dataFEED OPC Suite.



Local Items are used to build up a static namespace for the OPC Client2Client functionality of Softing dataFEED OPC Suite, which allows direct communication between different OPC Clients without direct access to an OPC Server. See Introduction - OPC Client2Client for a detailed introduction to the OPC Client2Client functionality.

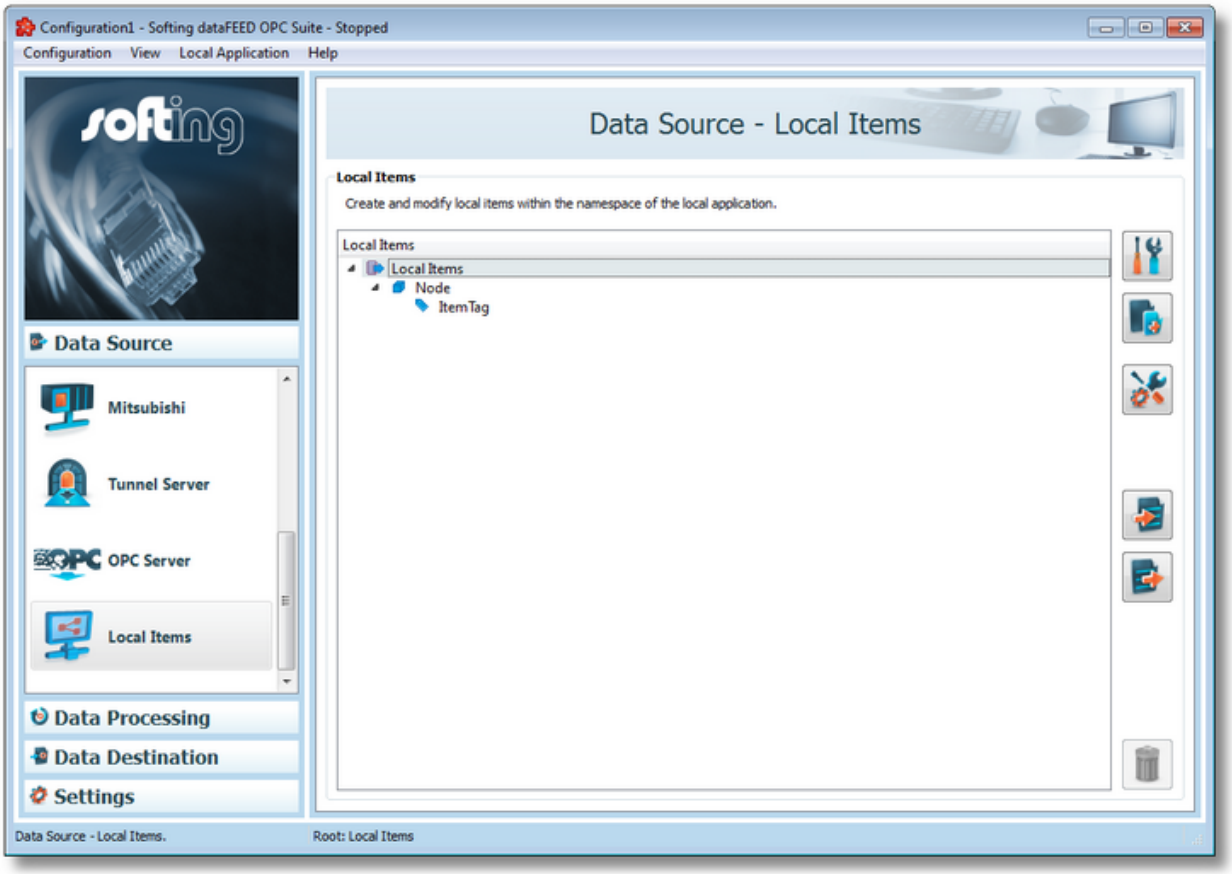
A detailed description of how to configure the OPC Client data destination functionality can be found in the Configuration: Data Source - Local Items section.

7.3.1.14.1Configuration



The Local Items namespace is configured in the *Local Items* main user interface. The configuration page can be reached by selecting *Local Items* from the *Data Source* section of the left navigation panel. The *Local Items* main configuration page will be displayed. For more details about the OPC Local Items see the Local Items page.

© Softing Industrial Automation GmbH

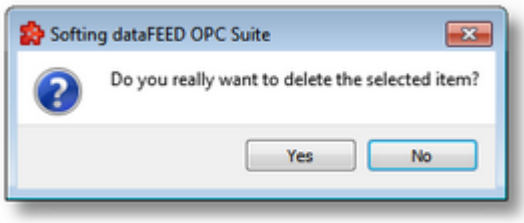


Local Items Tree Handling

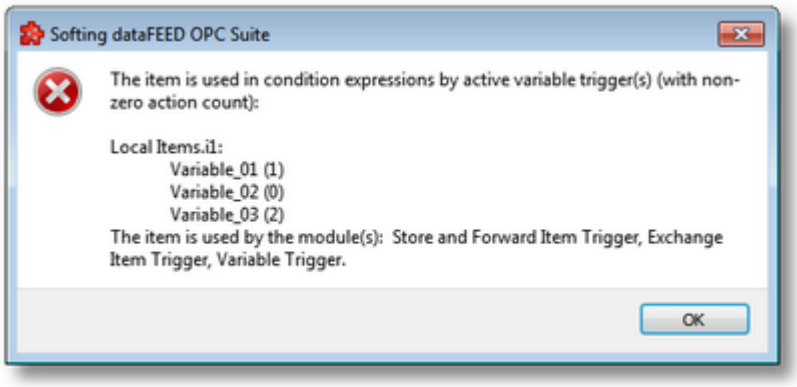
From the *Local Items* main user interface certain operations can be performed on the selected items by using the buttons on the right, accessing the context sensitive menu or by using the keyboard shortcuts. The following actions are available:

- Change the properties of the selected item (or change the properties of the root node) by using the *Properties* button  , pressing the keyboard shortcut CTRL+P or selecting the *Properties* action from the context menu.
- Add a new item under the selected node by using the *Add* button  , pressing the keyboard shortcut CTRL+A or selecting the *Add* action from the context menu.
- Copy and paste the local item and all of its children by selecting the *Copy* and *Paste* actions from the context menu.
- Change Advanced Settings by using the *Advanced settings* button  .
- Import from file by using the *Import from file* button  , pressing the keyboard shortcut CTRL+I or selecting the *Import* action from the root node's context menu.

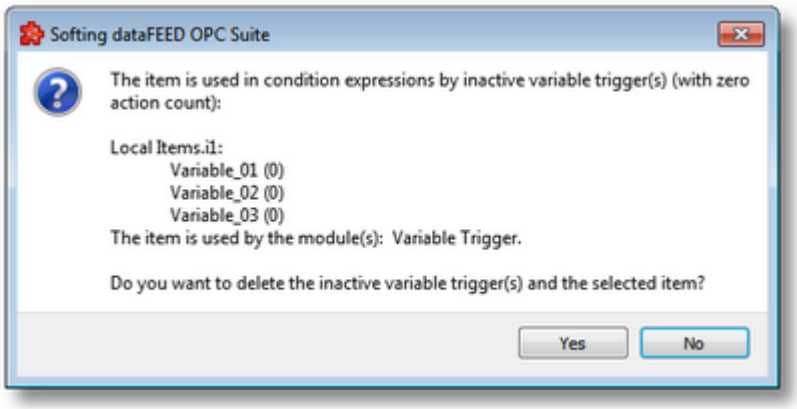
- Export to file by using the *Export to file* button  , pressing the keyboard shortcut CTRL+E or selecting the *Export* action from any node's context menu.
- Delete the selected local item using the *Delete data item* button  , pressing the DELETE key or selecting the *Delete* action from the context menu.



A dialog will inform the user that the item cannot be deleted, if the item is used by active variable triggers or other modules.

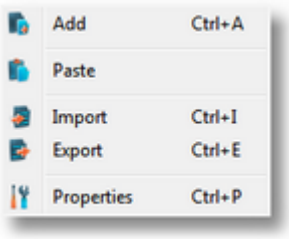


A dialog will prompt the user to confirm deletion of the item and inactive variable triggers, if the item is used only by inactive variable triggers.

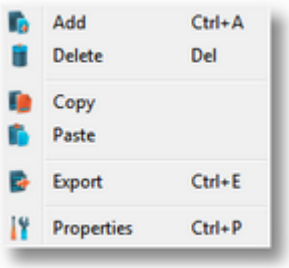


The available context menu options depend on the current selection in the tree. Multiple item selection is not allowed. An example of the root node context menu can be seen below:

© Softing Industrial Automation GmbH



An example of the context menu available for items of the Node type can be seen below:



Attention: The *Local Items* root node cannot be deleted. A local item which is being used in any consumer cannot be deleted, either.

Note: The copy/paste actions work on the selected node or on the whole selected branch. The copy action is not available for the *Local Items* root node, and the paste action is not available for items of type Tag because they cannot have any children.

Note: The *Export* action is not available for items of type Tag, and the *Import* action is only available for the *Local Items* root node.

Local Items Attributes

Each local item has an item ID and a data type.

Attribute	Description
ID	The fully qualified item ID (Local Item ID). This is the full path in the namespace of the Local Items tree.
Data Type and Access Rights	The data type and access rights of the item. The possible data type values are described below, mapped to the corresponding VARIANT types.

Note: All local items have the access rights "Read and Write" by default - which can be also configurable.

The status bar contains information about the selected local item: the item type, the item ID and for items of type Tag also the item data type and access rights.

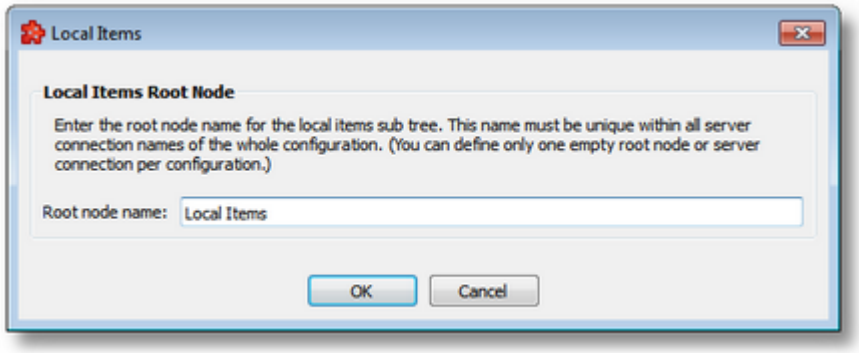
For a detailed description of how to add or modify local items see the Properties page.

7.3.1.14.2Local Items Root

The *Local Items* root node is created by default. This item cannot be deleted and the only available action for the *Local Items* root node is renaming it.

Note: The initial name of the root node is *Local Items*.

Its name can be changed using the *Properties of the selected item* button  , by using the CTRL+P keyboard shortcut, or by selecting the *Properties* context menu action. By triggering the properties action over the root node a dialog is displayed:



In this dialog the name of the root node can be changed.

Attention: The root node name should be unique throughout all configured data sources.

Note: The root node name can also be empty as long as it is unique.

Note: If the root name is empty and has only one child node (which may have any hierarchy below itself), the root node's unique child node becomes the new *Local Items* root node when you save and reopen the configuration.

See the Advanced Settings page for a description of general settings.

The Local Items list allows adding new local items and modifying the existing ones.

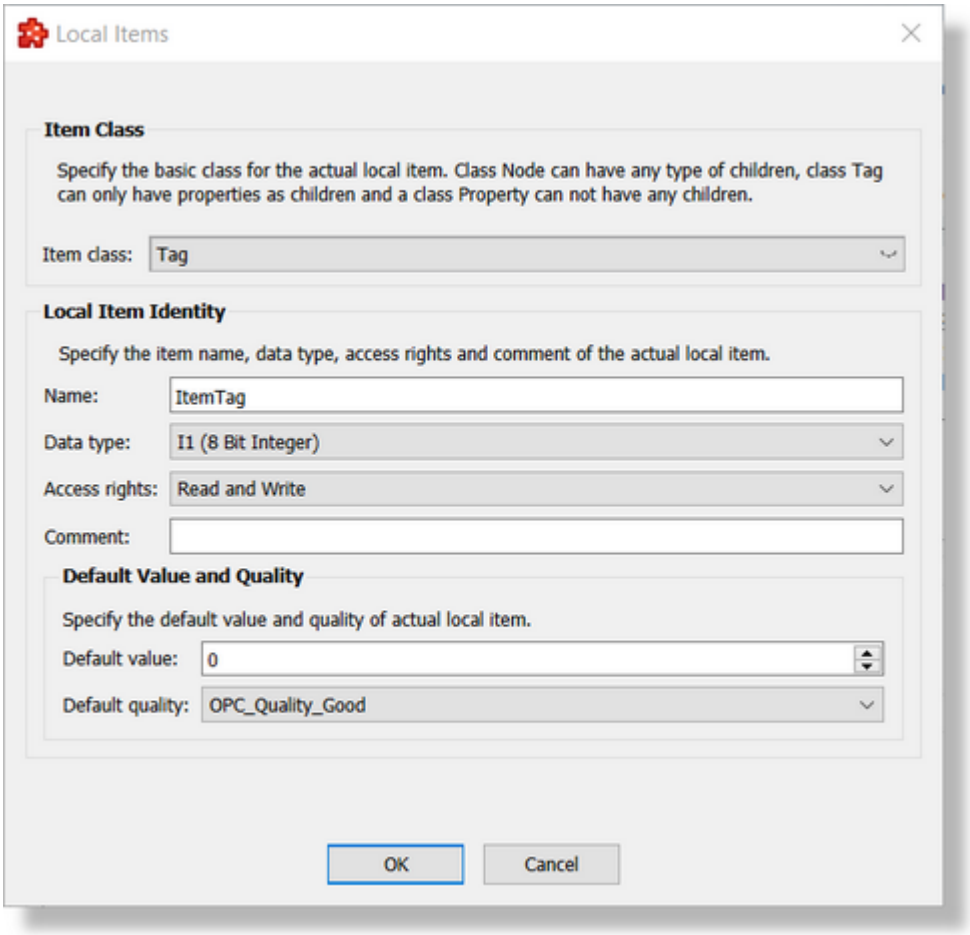
© Softing Industrial Automation GmbH

Add a New Local Item

The add operation is triggered by selecting the parent node and then pressing the *Add a new item* button  , using the keyboard shortcut CTRL+A or selecting the *Add* action from the context menu. The Add action is available for the root node item and for other items of type Node.

Note: A tag cannot have any children, so Add is not available for tags.

Invoking the Add action brings up the following dialog:



The following local item properties can be selected in this dialog:

- the item class or type, which can be Node or Tag
- the local item name. It should be unique for its level in the tree. It is available for both items of type Node and Tag.
- the local item data type. It is available only for items of type Tag.
- the local item access rights. It is available only for items of type Tag.
- the local item comment. It can be any text. It is available only for items of type Tag.

- the default/initialization OPC quality. It is available only for items of type Tag. By default this is set to OPC_Quality_BadWaitingForInitialData (0x20).
- the default/initialization OPC value. It is available only for items of type Tag having a non BAD (GOOD or UNCERTAIN) specified quality.
- the defaut/initialization array size. It is available only for items of type Tag having a non BAD specified quality and and ARRAY specified data type.

Note: The items of type Node represent intermediate nodes in the tree, they are used for hierarchy purposes. The items of type Tag represent actual local items.

Attention: The local items of type Node which do not have any children are not saved in the file. When you reopen the configuration they will disappear.

The local items of the namespace can be of any data type covered by the OPC Specifications. The following table describes the available data types mapped to the corresponding VARIANT types from the OPC Specification.

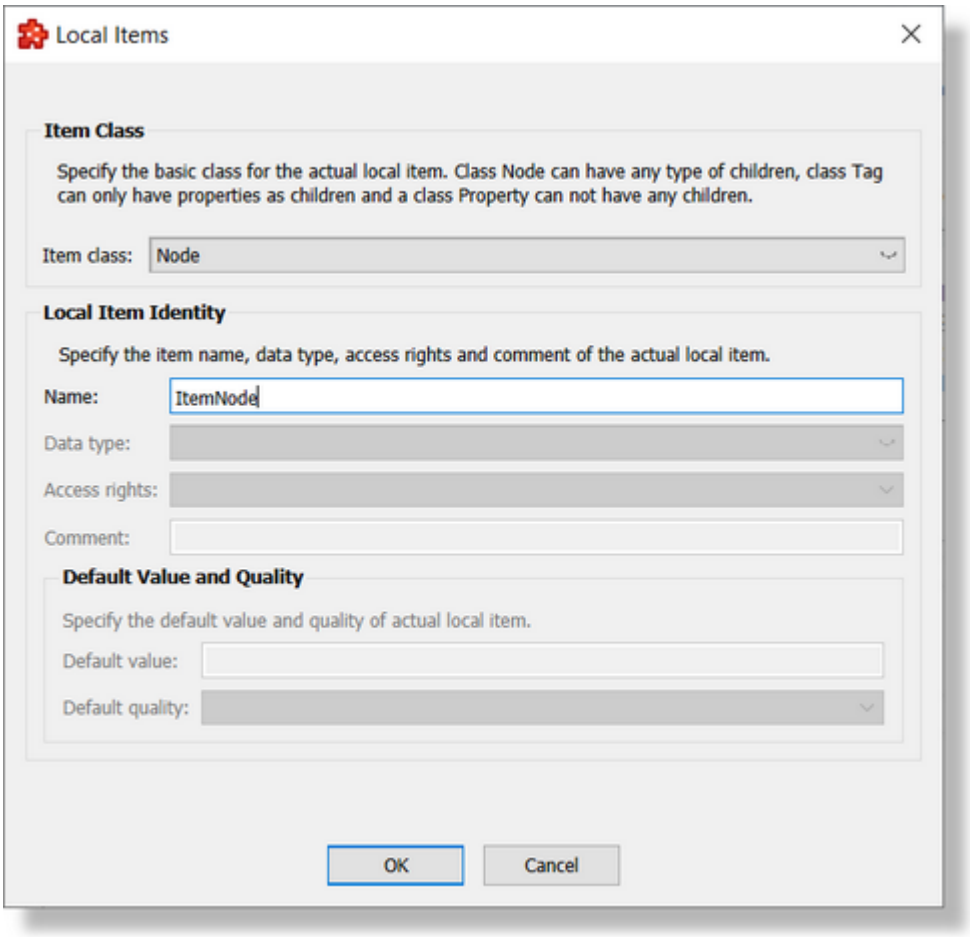
Data type	VARIANT type	Description
I1	*VT_I1*	8 bit signed integer.
UI1	*VT_UI1*	8 bit unsigned integer.
I2	*VT_I2*	16 bit signed integer.
UI2	*VT_UI2*	16 bit unsigned integer.
I4	*VT_I4*	32 bit signed integer.
UI4	*VT_UI4*	32 bit unsigned integer.
I8	*VT_I8*	64 bit signed integer.
UI8	*VT_UI8*	64 bit unsigned integer.
R4	*VT_R4*	32 bit floating point.
R8	*VT_R8*	64 bit floating point.
BOOL	*VT_BOOL*	Boolean.
STRING	*VT_BSTR*	String.
DATETIME	*VT_DATE*	Date and time.
DECIMAL	*VT_CY*	Decimal.
ANY	*VT_VARIANT*	Any data type.

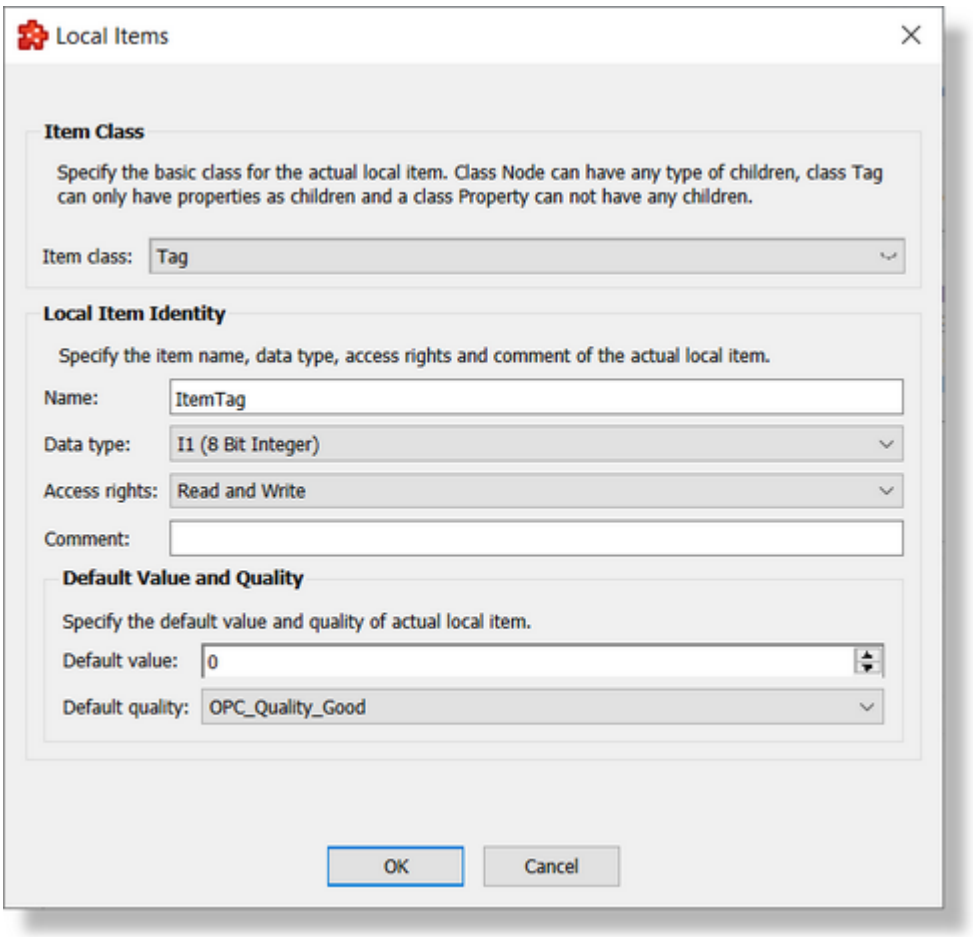
© Softing Industrial Automation GmbH

Note: Arrays of fixed and variable length are also supported for these data types.

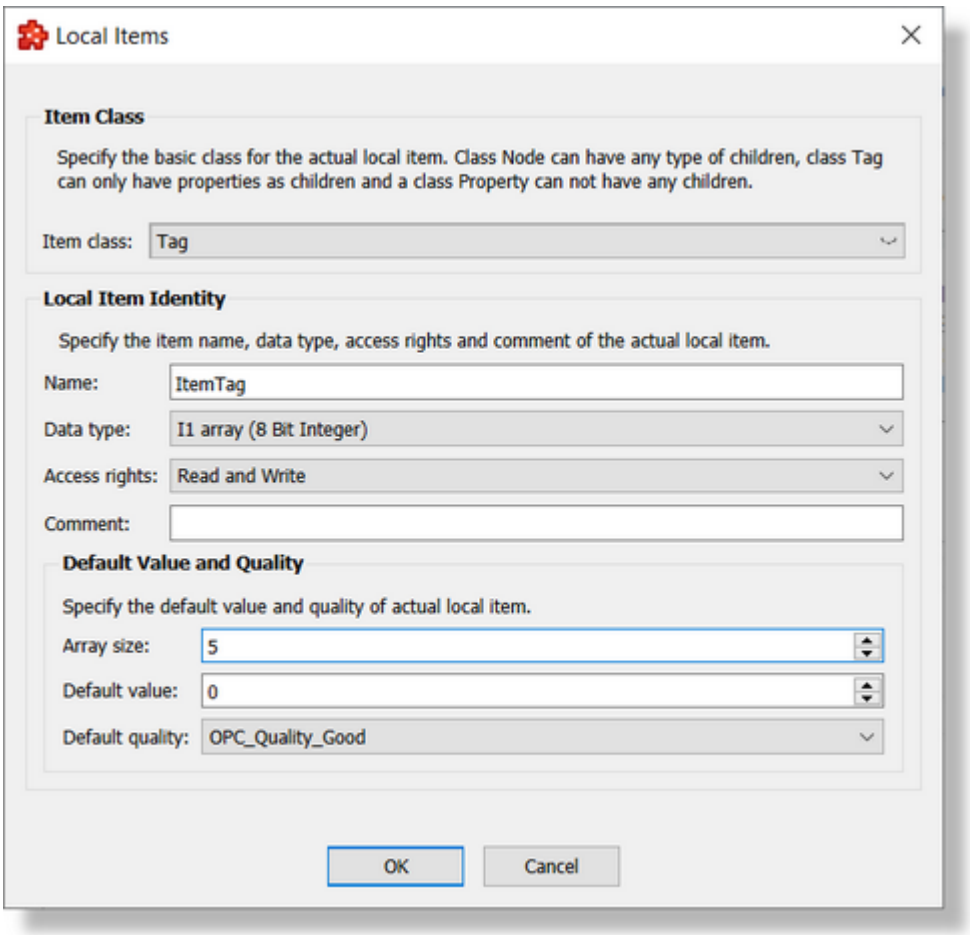
Modify a Local Item

Modifying a local item is triggered by selecting the item and then pressing the Properties of the selected item button  , using the keyboard shortcut CTRL+P or selecting the Properties action from the context menu. Performing the Modify action displays the following dialog:





© Softing Industrial Automation GmbH



This dialog allows provides the same properties as the add item dialog.

Note: The local item class cannot be modified as the hierarchy of the items relies on it.

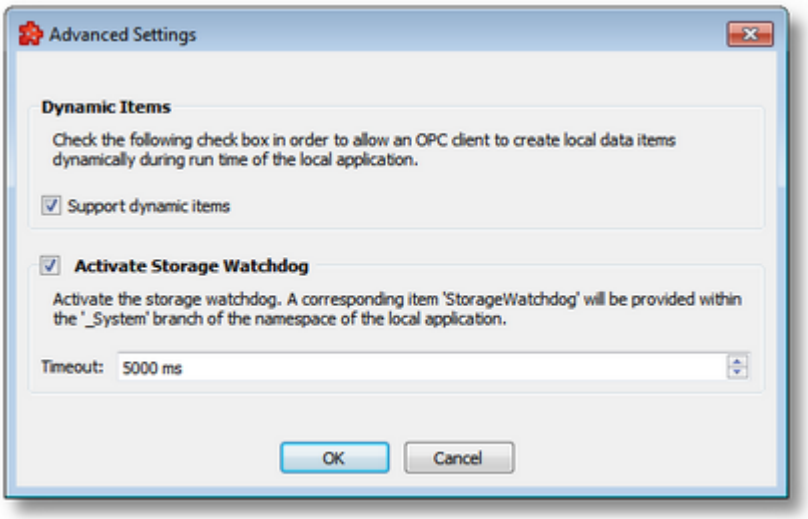
Note: The Modify action is triggered by double-clicking an item with no children.

By default all specified local item tags are initialized with NULL value and OPC_Quality_BadWaitingForInitialData (0x20) quality. This can be configured by specified a defined OPC quality, a specified value based on the native data type of the corresponding item tag and in case of being an array it can be specified also the array length - is is allowed only for non BAD specified qualities.

Modifying of the Local Items root node is described in the Local Items Root page.

7.3.1.14.3Advanced Settings

The *Advanced settings* page allows changing advanced properties of the Local Items. The action can be performed by clicking the *Advanced Settings* button  . The following dialog will be displayed:



In this dialog the following settings can be changed:

Attributes	Description
Support dynamic items	If this option is checked the OPC clients are allowed to create dynamic items during the run time of the local application in the local items tree. If it is unchecked the dynamic items creation is disabled. A dynamic item can be created by an OPC client that supports this feature. The syntax is the following: `<ItemName>%<DataType>`
Activate Storage Watchdog	A client that acts logically as an OPC server has to cyclically write to the Storage Watchdog. If the watchdog is no longer written to, the qualities of all items are set to *communication failure*. If this option is true, the *_System.Data_Source.LocalItems.LocalItemWatchdog* item is created in the *System items*.
Watchdog timeout	*Watchdog Timeout* is the time in milliseconds within which the watchdog has to be cyclically written to. If the watchdog item is provided, it must be written to by an OPC client at least once within the watchdog timeout. If the watchdog item is not provided, the quality changes to "communication failure" for all items. This mechanism is used for monitoring the OPC client that provides the values via the OPC Storage.

The Local Items tree hierarchy can be exported or imported. See the Import Wizard and Export Wizard pages for details.

© Softing Industrial Automation GmbH

7.3.1.14.4Local Items File

The Local Items tree is saved in a plain text file called *LocalItems.txt* which can be found in the *LocalItems* folder under the current configuration folder.

Note: The path and the name of the local items configuration file cannot be configured.

The filter files are plain text files, encoded in UTF-8 (thus providing full UNICODE compatibility) and they have a line-oriented structure. The elements of a line are tab-separated. For each OPC local item a line is added to the file.

The following line types can be found in the local items file:

- Local item line, defining a local item as described in the table below.
- Empty line.
- Comment line, starting with #; such a line is ignored during evaluation

```
<TAB>Local item ID<TAB>Data Type and Access Rights<TAB>Comment<TAB>Initial Quality[<T
```

The different elements of the line are described in the following table:

Element	Description
Local item ID	The full local item path from the Local Items root node, including the root node name.
Data Type and Access Rights	The data type and access rights of the data item. The possible values for data type are described in the Filter Item Properties page.
Initial Quality	The OPC quality of the item tag at the initialization time. If missing it is set to OPC_Quality_BadWaitingForInitialData. The specified quality values can be used: • OPC_Quality_Good (0xC0) • OPC_Quality_GoodLocalOverride (0xD8) • OPC_Quality_Uncertain (0x40) • OPC_Quality_UncertainLastUsable (0x44) • OPC_Quality_UncertainEngunitsExceeded (0x54) • OPC_Quality_Bad (0x0) • OPC_Quality_BadWaitingForInitialData (0x20) • OPC_Quality_BadConfigError (0x4) • OPC_Quality_BadNotConnected (0x8) • OPC_Quality_BadDeviceFailure (0xC) • OPC_Quality_BadSensorFailure (0x10)

	• OPC_Quality_BadCommunicationError (0x18) • OPC_Quality_BadOutOfService (0x1c)
Initial Value	The value of the item tag at the initialization time, this is available only for non BAD qualities.
Array Size of Initial Value	The array length of the initial value in case when the specified data type is an ARRAY.

Example:

```
<TAB>Local item ID<TAB>Data Type and Access Rights<TAB>Comment<TAB>Initial Quality[<T
#
<TAB>Local Items.Node1.Tag1<TAB>I1_RW<TAB><TAB>0x20
<TAB>Local Items.Node1.Tag2<TAB>I1_RW<TAB><TAB>0xC0<TAB>1
```

The configuration file can be managed with the dataFEED OPC Suite Configurator or it can be manually edited.

Note: The recommendation is to use only the dataFEED OPC Suite Configurator to edit these files. This ensures that all the definitions will be valid.

If the local items configuration file is manually edited try loading the entire configuration in the dataFEED OPC Suite Configurator as all the configuration files are automatically validated at load-time and, if there are any mistakes, an error message will be displayed. The dialog will contain an appropriate description and will point to the first invalid line of the local items configuration file.

7.3.1.14.5Import Wizard

The dataFEED OPC Suite Configurator allows the import of Local Items from an external file.

The external import file must be a plain text tab-separated file with the specific Local Items file format.

The import action can be triggered by using the *Import from file* button  , by pressing the keyboard shortcut sequence CTRL+I or by selecting the *Import* action in the context sensitive menu.

Note: The import action is only available for the *Local Items* root node.

Note: The names conflicts between the old and new items are automatically solved by renaming the newly imported items. They are renamed by appending a two-digit number at the end of the name of the first duplicated item.

© Softing Industrial Automation GmbH

When the import action is triggered the *Local Items Import Wizard* is started.

▾ The Local Items Import Wizard allows importing local items:

- Select the file to be imported @ Import Wizard - Import File Selection
- Examine the import results @ Import Wizard - Import Results

The Local Items can be imported from a text file with a specific structure. The file should have a line-oriented structure. The elements of a line must be tab-separated. For each line in the file a new OPC local item is created.

The following line types can be found in the local items file:

- Local item line, defining a local item as described in the below table.
- Empty line.
- Comment line, starting with #; such a line is ignored during evaluation

A local item line has the following format:

```
<TAB>Local item ID<TAB>Data Type and Access Rights<TAB>Comment<TAB>Initial Quality[<T
```

The different elements of the line are described in the following table:

Element	Description
Local item ID	The full local item path from the Local Items root node, including the root node name.
Data type and access rights	The data type and access rights of the data item. The possible values are described in the Filter Item Properties page.
Initial quality	The OPC quality of the item tag at the initialization time. If missing it is set to OPC_Quality_BadWaitingForInitialData.
Initial value	The value of the item tag at the initialization time only for non BAD qualities.
Array size of initial value	The array length of the initial value in case when the specified data type is an ARRAY.

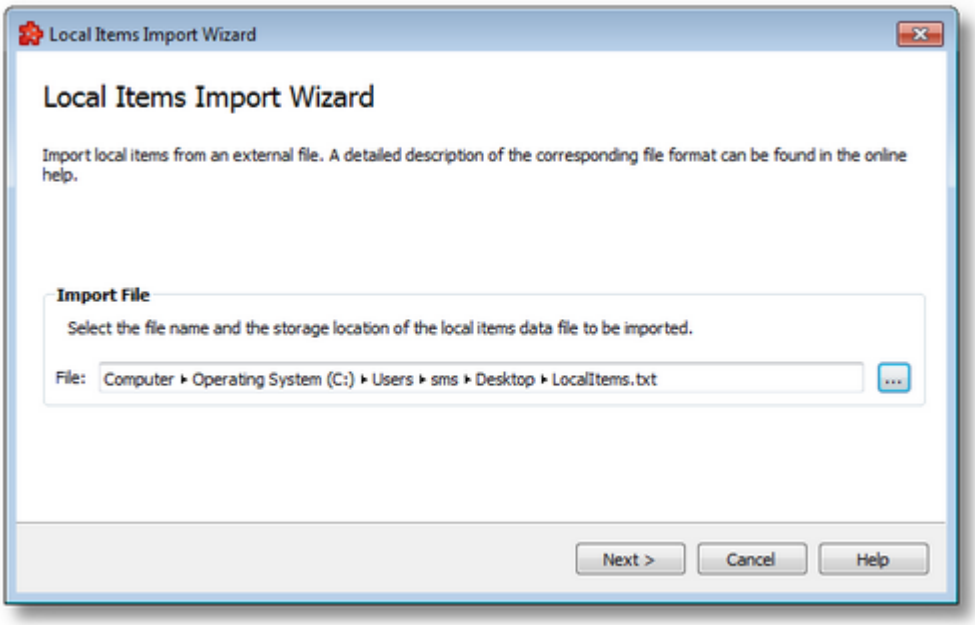
Example:

```
<TAB>Local item ID<TAB>Data type and Access Rights<TAB>Comment<TAB>Initial Quality[<T
#
```

```
<TAB>Local Items.Node1.Tag1<TAB>I1_RW<TAB><TAB>0x20
<TAB>Local Items.Node1.Tag2<TAB>I1_RW<TAB><TAB>0xC0<TAB>1
```

The Import File Selection wizard page allows you to choose a file to be imported.

The file can be selected by manually typing its name in the *File* input box or by clicking the browse button located to the right of the input box (a standard file opening dialog will open).

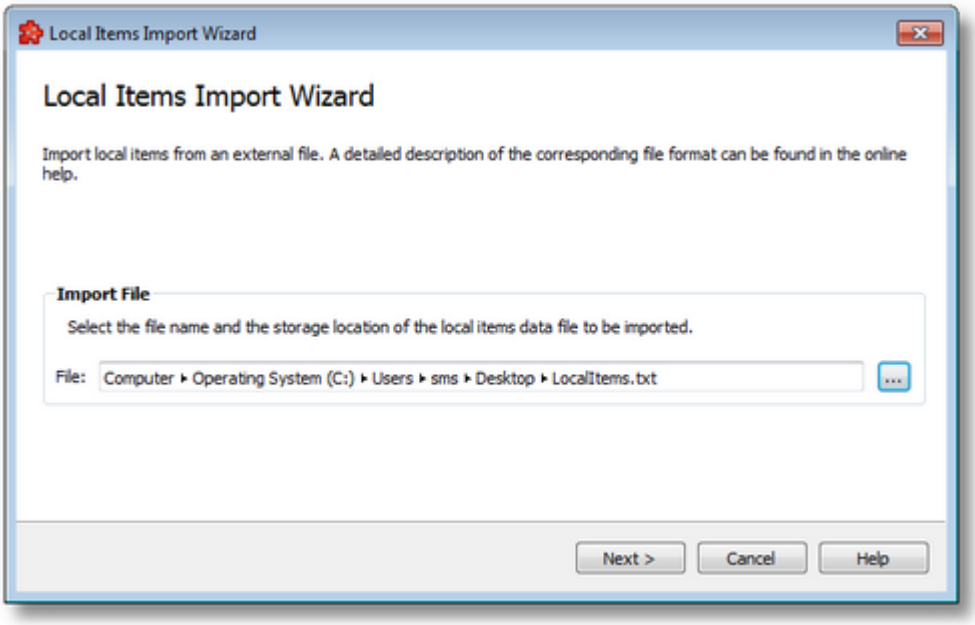


The file path and name should be valid (they should point to an existing file). When you press the *Next* button the file path and name are verified. If the file identified by the specified file path and name does not exist an error message is displayed. The actual import begins if the selected file is valid. If the import takes longer to complete a waiting dialog is displayed.

Import Wizard - Import Results >>>

© Softing Industrial Automation GmbH

The Import Results wizard page displays the import results.



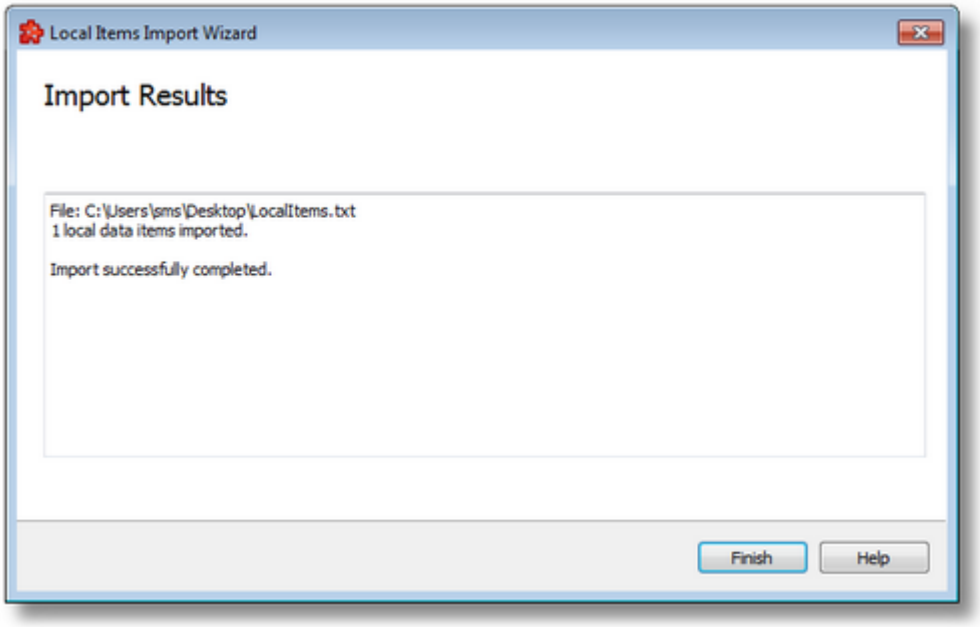
This page is the final page of the Filter Import Wizard. A back button is not provided as the import action cannot be undone. The import can be successful or it can fail:

1. After a successful import the results page contains the following information:

   - The absolute path of the imported file.
   - The number of local items imported. This number includes only the items of type Tag, the items of type Node are not counted.
   - The confirmation message that the import was successful.

2. After a failed import the results page contains the following information:

   - The invalid file line at which the import stopped.
   - The reason why the line is considered invalid.
   - The confirmation message that the import failed.



When the import wizard is closed after a successful import, the new items are added to the current Local Items tree as a branch of the root node item.

<<< Import Wizard - Import File Selection

7.3.1.14.6Export Wizard

The dataFEED OPC Suite Configurator allows the export of full branches from the Local Items tree. The entire Local Items tree or just a sub-branch of it can be exported.

The resulting external file will be a tab-separated file with the specific Local Items file format.

The export action can be triggered by pushing the *Export* button  , by pressing the keyboard shortcut sequence CTRL+E or by selecting the *Export* action in the context sensitive menu.

Note: The export action is available for the Local Items root and for items of type Node.

The exported file will contain all the items from the selected node's branch.

Note: The exported file will contain only the tags of the selected node with the complete node structure from the selected node to each of the tags.

Note: If the entire Local Items root was exported the resulting file should be identical with the configuration's Local Items file.

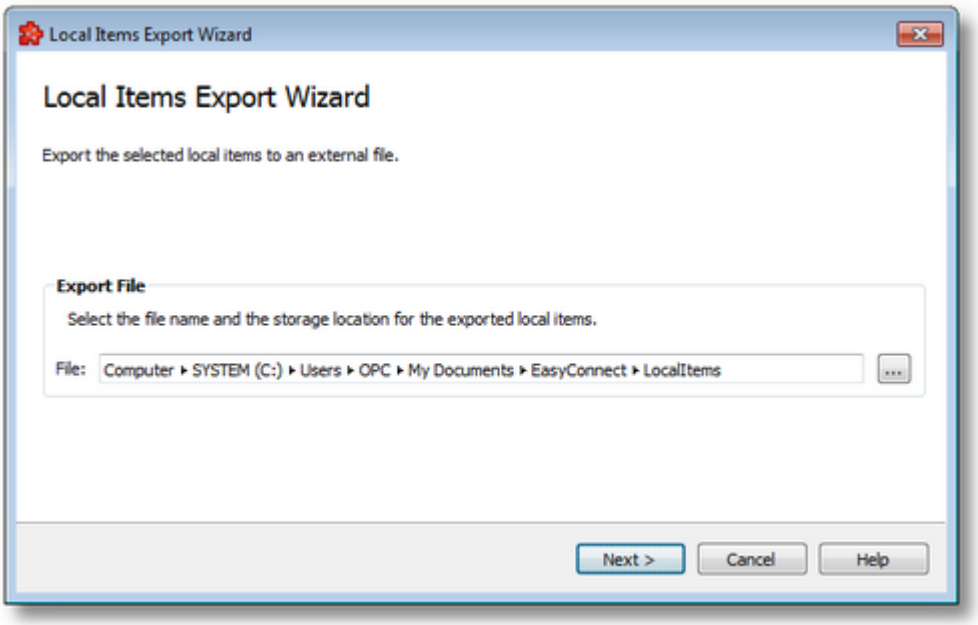
When the export action is triggered the *Local Items Export Wizard* is started.

© Softing Industrial Automation GmbH

▼ The Local Items Export Wizard allows exporting local items:

- Select the file the local items are to be exported to @ Export Wizard - Export File Selection
- Examine the export results @ Export Wizard - Export Results

The Export File Selection wizard page allows you to choose a file name and location for the exported file. The file can be selected by manually typing its path and name in the *File* input box or by clicking the browse button located to the right of the input box (a standard file saving dialog will open).

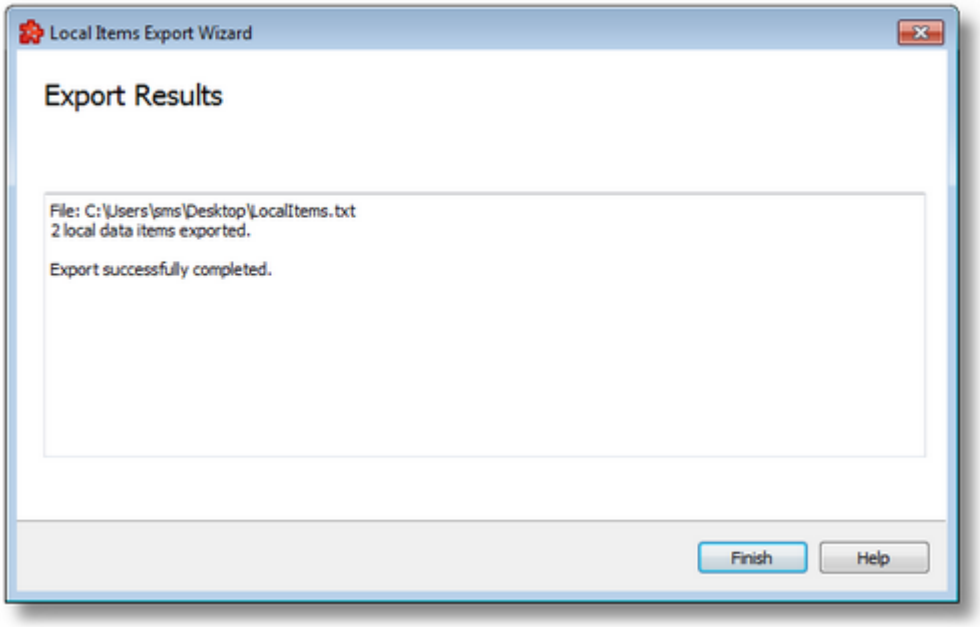


When the *Next* button is pressed the file path is verified and if the path does not exist, an error message is displayed (invalid path). If the specified file path is valid and a file with the specified name already exists, a message box will ask whether to overwrite the existing file or not (select another file). The actual local items export operation begins if the path to the file is valid and either a file with the specified name does not exist or it exists and overwrite has been chosen.

When the export operation is finished the *Export Results* page is displayed.

Export Wizard - Export Results >>>

The Export Results wizard page displays the export results as shown in the image below:



This page is the final page of the Local Items Export Wizard. A back button is not provided as the export action cannot be undone. The export can be successful or it can fail:

1. After a successful export the results page contains the following information:
   - The absolute path of the exported file.
   - The number of local items exported. This number includes only the items of type Tag, the items of type Node are not counted.
   - The confirmation message that the export was successful.

2. After a failed export the results page contains the following information:
   - The reason why the export failed.
   - The confirmation message that the export failed.

<<< Export Wizard - Export File Selection

7.3.1.15 Appendix

This section contains general information that applies to all or multiple controllers or device families.

Main sections are:

- S7 Symbol File Import
- Import Wizard

© Softing Industrial Automation GmbH

- Export Wizard
- Global Settings Wizard
- PLC Connection Item Properties
- Item Syntax
- TSAP information

7.3.1.15.1S7 Symbol File Import Wizard

The Softing dataFEED OPC Suite Configurator allows the import of S7 PLC connection items from an external STEP 7 symbol file or from a TIA symbol file generated with the Softing dataFEED Exporter.

Note: Make sure the Softing dataFEED Exporter application is installed on that PC where you have installed your **Siemens Totally Integrated Automtation Portal** version V13 or V13.1.

The import action can be triggered by using the *Import S7 symbol file* button  .

Attention: The import from S7 symbol file action applies on the entire address space of the current S7 PLC connection. Any existing items will be automatically removed from the address space prior to performing the import.

Attention: The option is not available for Siemens S7-2 connections (direct connections to Siemens S7-1200 or S7-1500 PLCs with Optimized Access).

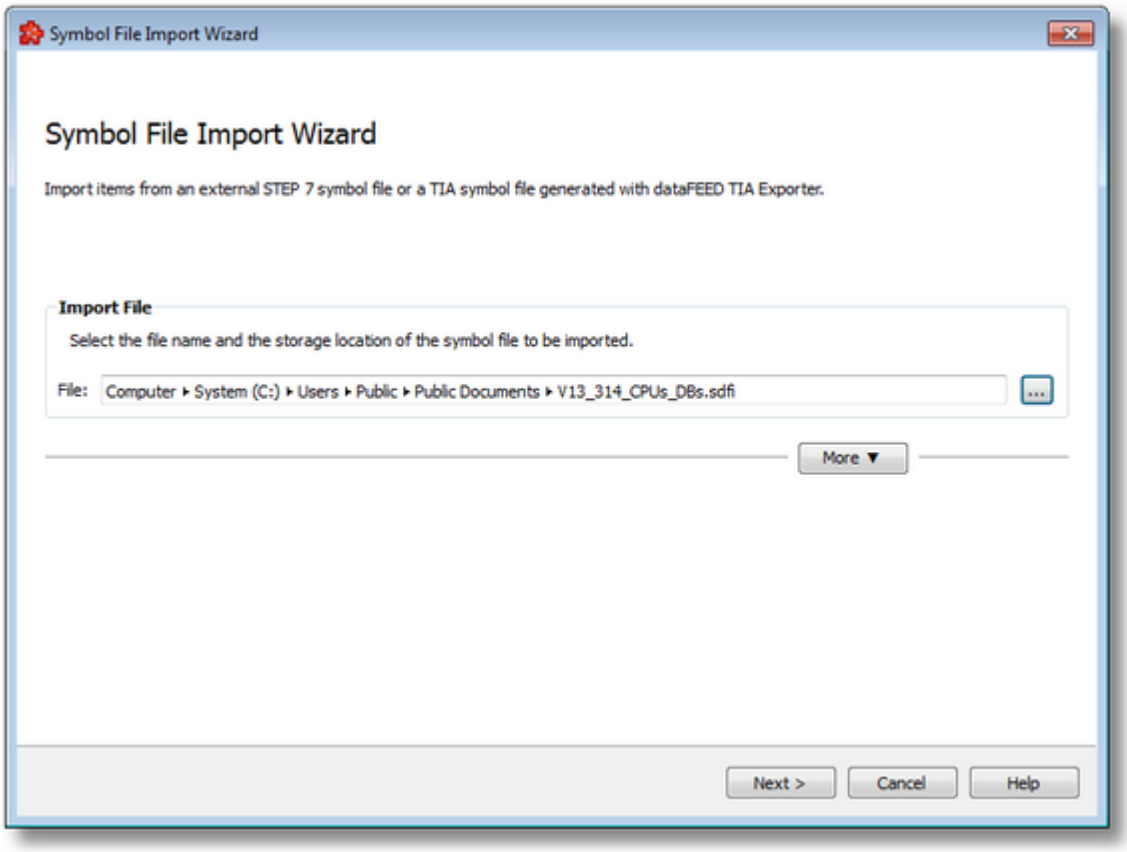
When this import action is triggered, the *Symbol File Import Wizard* is started.

▾ The Symbol File Import Wizard allows importing different data items:

- Select the file to be imported @ S7 Project Import Wizard - File Selection
- Select the PLC Program to use for import @ S7 Project Import Wizard - PLC Program Selection
- Select the Device to use for import @ S7 Project Import Wizard - Device Selection
- Select the PLC Blocks to import @ S7 Project Import Wizard - PLC Blocks Selection
- Examine the import results @ S7 Import Wizard - Import Results

The Symbol File Import wizard page allows you to choose a file to be imported.

The file can be selected by manually typing its name in the *File* input box or by clicking the browse button located to the right of the input box (a standard file selection dialog will open).



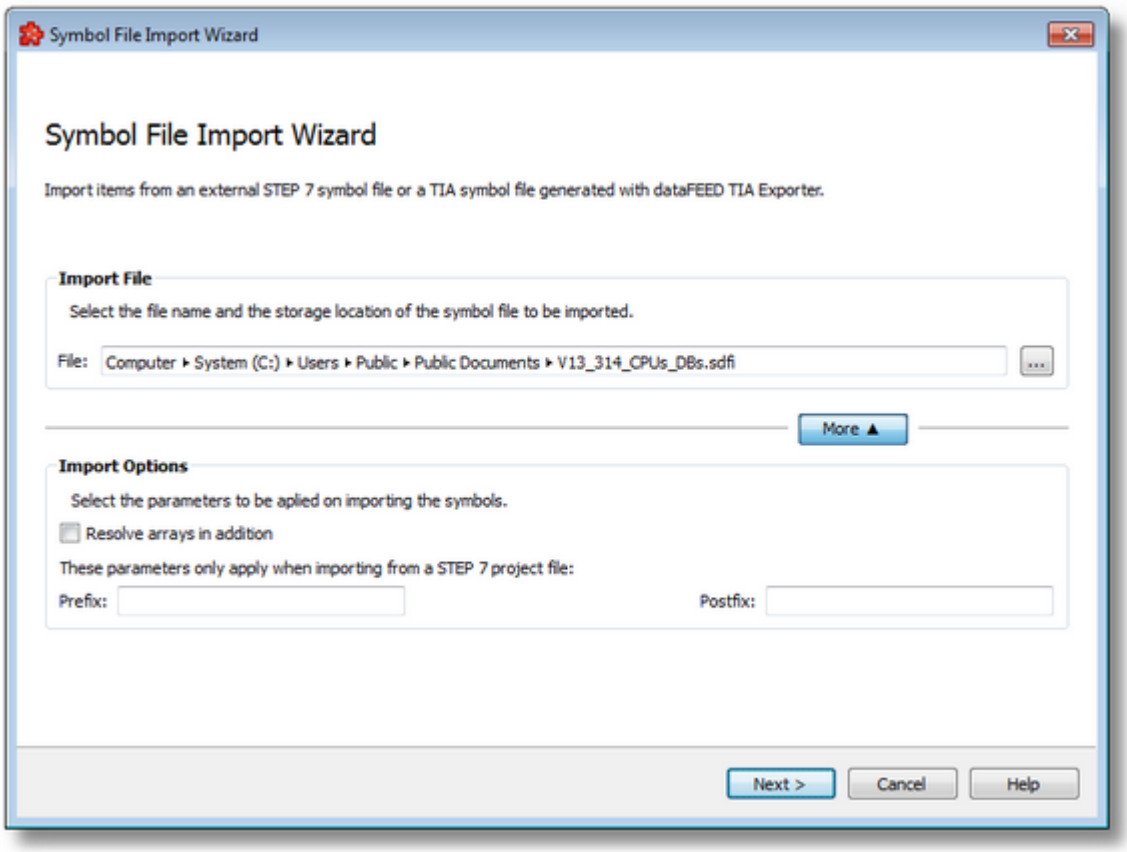
The file path and name should be valid (they should point to an existing file).

Note: When importing an **.sdfi* file from a TIA project, all symbols are imported but without their device definitions and settings.

When you press the *Next* button the file path and name are verified. If the file identified by the specified file path and name does not exist, an error message is displayed.

The import from a STEP 7 project file can be parameterized with some advanced options. For this, additional controls are presented to the user by pressing the *More* button, as shown in the picture below:

© Softing Industrial Automation GmbH



The available additional options in this wizard page are described in the table below:

Attribute	Description
Prefix	A prefix to be added to each data item which is created as result of the import (optional).
Postfix	A postfix to be added to each data item which is created as result of the import (optional).
Resolve arrays in addition	Every single element of an array is given its own icon name. Default value is switched off.

Note: The hierarchical separator character in the *Prefix* and *Postfix* fields is used to change/adapt the hierarchy of the imported data items. It is compulsory at the end of the *Prefix* and at the beginning of *Postfix*.

Note: The *Prefix* and *Postfix* parameters only apply when importing from a STEP 7 project file.

Note: The *Resolve arrays in addition* checkbox does not apply to DateTime, TimeStamp, S5Time, TOD (Time Of Day), Bool type elements. They will always be created, regardless of the checkbox setting.

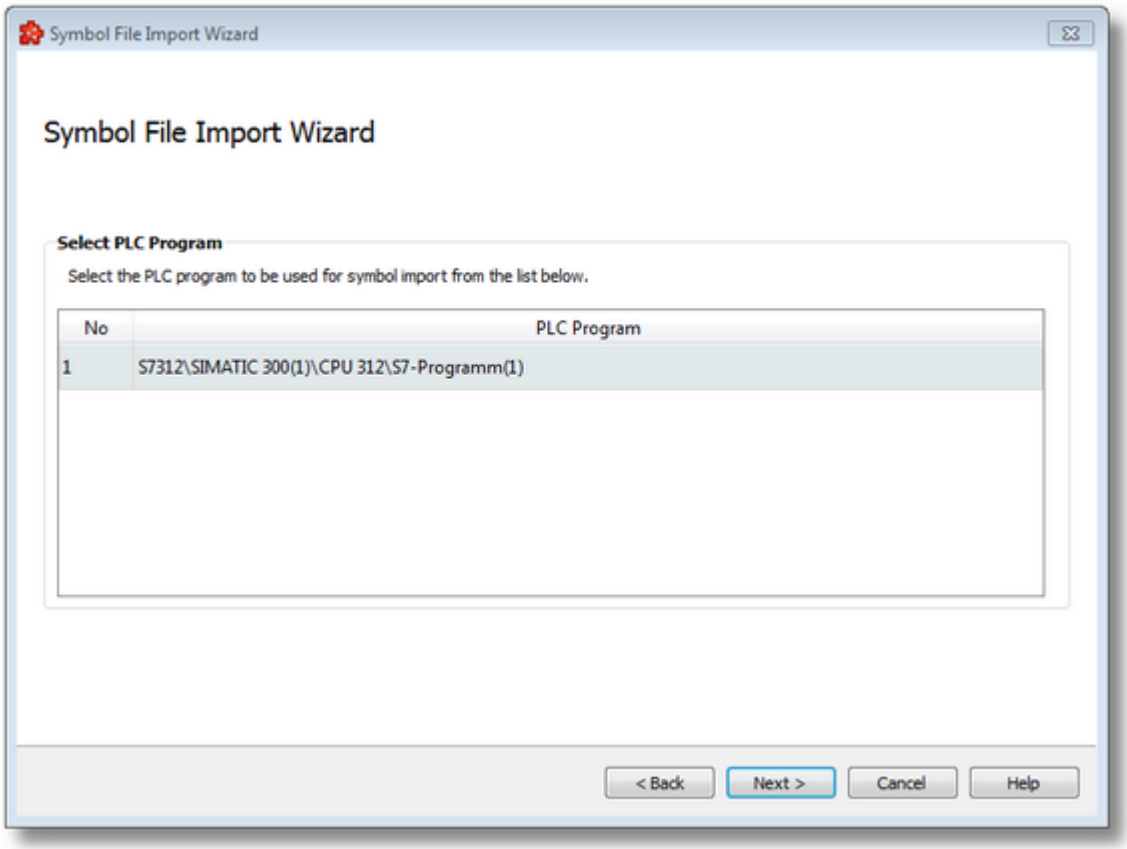
Attention: The usage of *Resolve arrays in addition* option may greatly enlarge the size of the resulting address space and this takes more time during program start.

Import Wizard - PLC Program Selection >>>

Import Wizard - Device Selection >>>

The second page in the STEP 7 Symbol File Import wizard allows you to choose a program inside the S7 project to be used for import.

A list with all available programs is displayed. Select the desired program within this list.

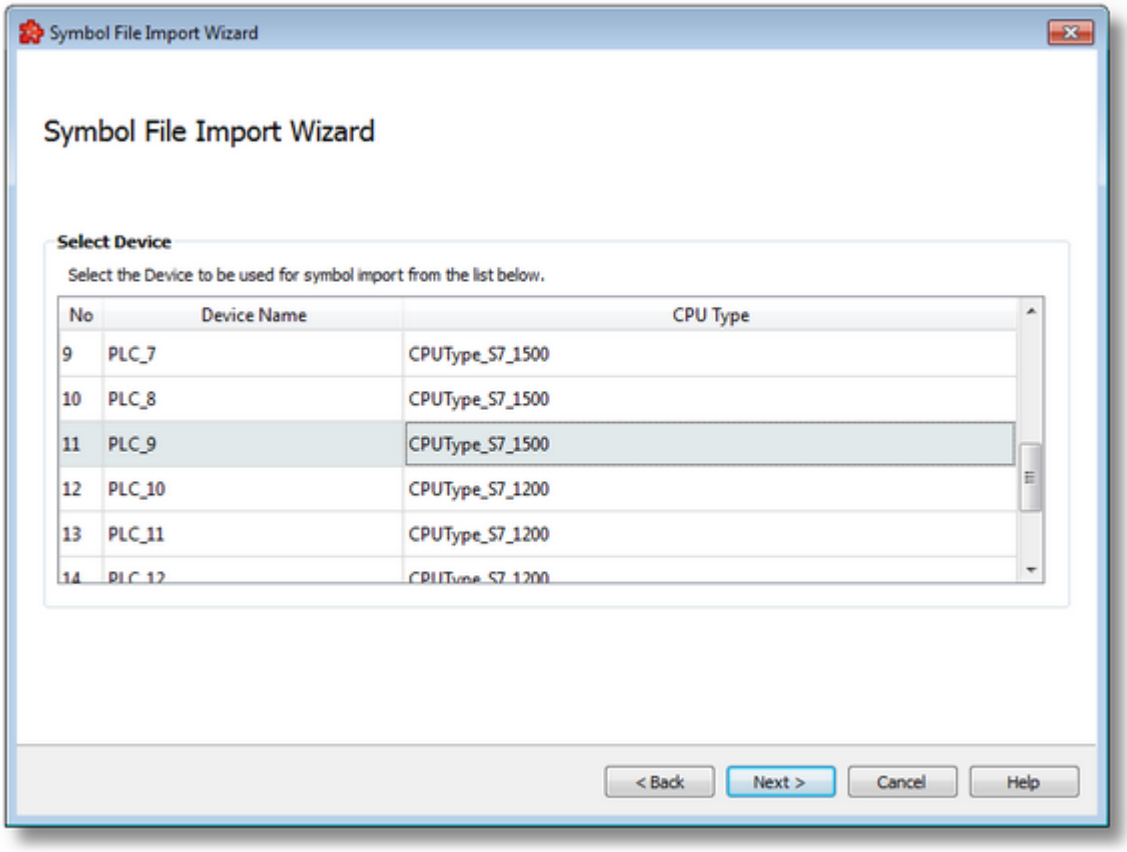


<<< Import - File Selection Import - PLC Blocks Selection >>>

The second page in the TIA Symbol File Import wizard allows you to choose a program inside the TIA project to be used for import.

A list with all available devices is displayed. Select the desired device within this list.

© Softing Industrial Automation GmbH



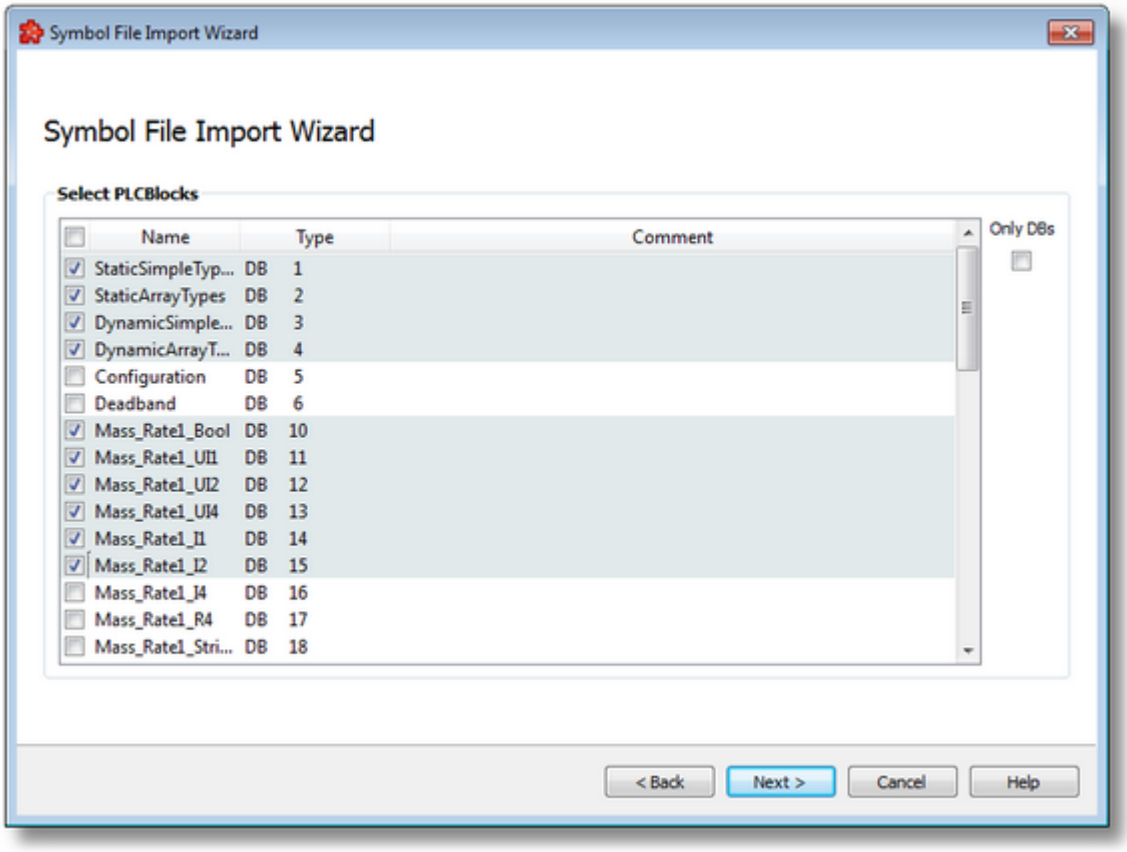
<<< Import - File Selection Import - PLC Blocks Selection >>>

The third page in the S7 Symbol File Import wizard allows the selection of individual data blocks that will be imported.

Note: We only support data blocks which are not running in the optimized mode.

With large PLC projects, we recommend only importing those symbols that are really required on the server.

Multiple selection is allowed, as well as *Select All* feature by using the checkbox in the table header or keyboard shortcut (Ctrl+A).



It is possible to filter the content of the table by using the *Only DBs* checkbox in the right side of the page. If this option is activated, only the data blocks are shown in the table. If not, all symbols are displayed.

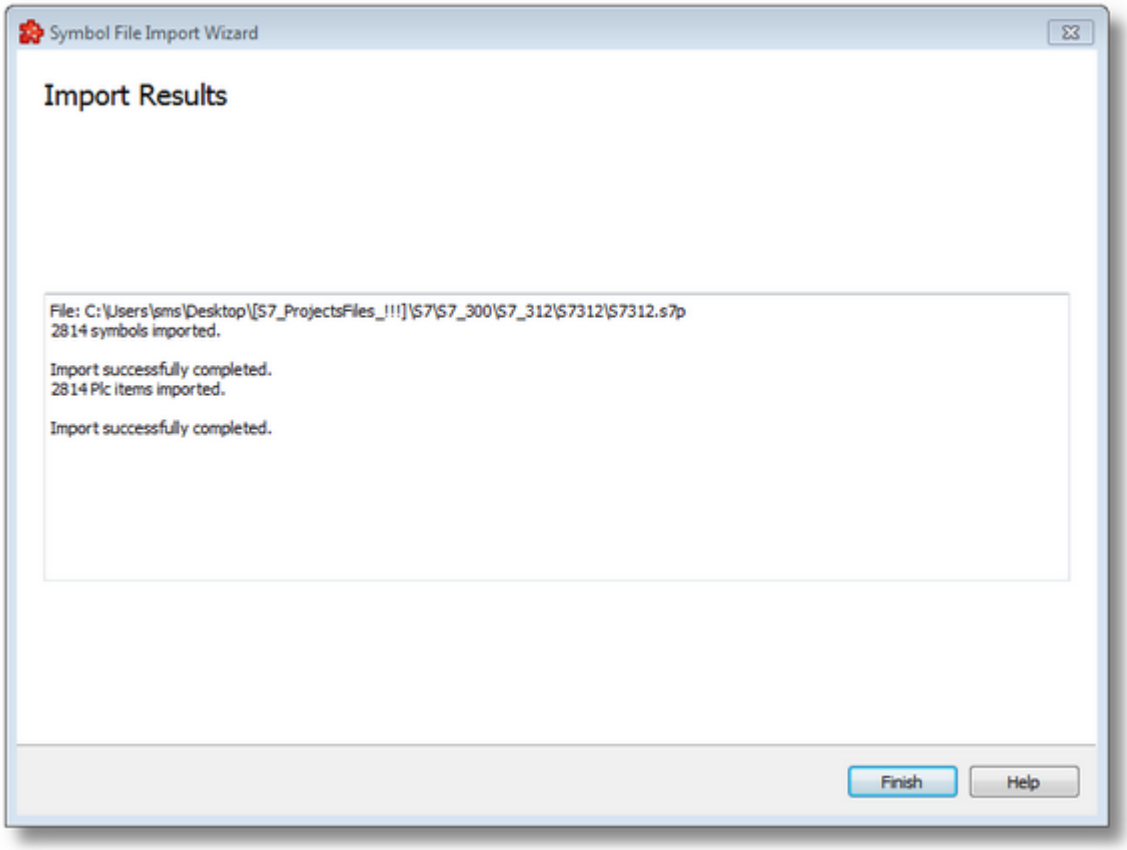
<<< Import - PLC Program Selection

<<< Import - Device Selection

Import - Results >>>

The Import Results wizard page displays the import results.

© Softing Industrial Automation GmbH



This page is the final page of the *Import S7 Symbol File Wizard*. A back button is not provided as the import action cannot be undone. The import can be successful or it can fail:

1. After a successful import the results page contains the following information:
   - The absolute path of the imported file.
   - The total number of imported symbols.
   - The confirmation message that the import was successful.
2. After a failed import the results page contains the following information:
   - The invalid line where the import failed.
   - The reason why the line is considered invalid.
   - The confirmation message that the import failed.

When the import wizard is closed after a successful import, the new items are added to the current PLC connection tree and all the previous data items are removed.

<<< Import - PLC Blocks Selection

Softing's dataFEED Exporter allows to convert TIA symbol files into Softing **.sdfi* files.

You can find the dataFEED Exporter application on your product CD-ROM or you can download the current version from the download area on the Softing website.

For detailed information about file conversion and export/import functionality refer to the online help of the dataFEED Exporter application.

Note: Make sure you have copied the **.sdfi* file to a location that can be accessed by the dataFEED Configurator application.

Note: Make sure the dataFEED Exporter application is installed on that PC where you have installed your **Siemens Totally Integrated Automtation Portal** version V13 or V13.1.

Note: When importing an **.sdfi* file from a TIA project, all symbols are imported but without their device definitions and settings.

1. Start the import process as described in S7 Symbol File Import Wizard.
2. Locate and import the **.sdfi* file.
3. Pass all steps from the Import Wizard.

When importing TIA projects, the data types and their corresponding OPC data types listed in the table below are supported.

The Struct or UDT types are supported as nodes with children sub-trees having the scalar leaves displayed as items of the corresponding data types.

Note: We only support data blocks which are not running in the optimized mode.

Note: Unsupported data types which appear in a structure or in a user defined data type (UDT) will not be imported and will not be displayed in the address tree.

TIA project data type	dataFEED OPC Suite data type	Syntax DB/DI	Syntax Item sample	Array Support	Suffix
Bool	VT_BOOL	X	DB1,X5.3	NO	None
Byte	VT_UI1	B	DB1,B5	YES	None
		Byte	DB1,Byte5		

© Softing Industrial Automation GmbH

TIA project data type	dataFEED OPC Suite data type	Syntax DB/DI	Syntax Item sample	Array Support	Suffix
Char	VT_CHAR	C	DB1,B5	YES	None
		Char	DB1,Char5		
Word	VT_UI2	W	DB1,W5	YES	None
		Word	DB1,Word5		
Int	VT_I2	I	DB1,I5	YES	None
		Int	DB1,Int5		
DWord	VT_UI4	DW	DB1,DW5	YES	None
		Dword	DB1,Dword5		
DInt	VT_I4	D	DB1,D5	YES	None
		DI	DB1,DI5		
		DInt	DB1,Dint5		
Real	VT_R4	R	DB1,R5	YES	None
		Real	DB1,Real5		
Date	VT_BSTR	W	DB1.W5D	NO	D
		Word			
Time_Of_Day	VT_UI4	DW	DW1,DW5TOD	NO	TOD
		Dword	DW1,Dword5TOD		
Time	VT_UI4	DW	DW1,DW5T	NO	T
		Dword	DW5,Dword5T		
S5Time	VT_UI2	W	DW1,W5S5T	NO	S5T
		Word	DW1,Word5S5T		
Date_And_Time	VT_UI2	W	DW1,W5DT	NO	DT
		Word	DW1,Word5DT		

TIA project data type	dataFEED OPC Suite data type	Syntax DB/DI	Syntax Item sample	Array Support	Suffix
String	VT_BSTR	S	DB1,S1	YES	None
		String	DB1,S1.2		
			DB1,S1.2.10	Array	
			DB1,String1		
			DB1,String1.2		
			DB1,String1.2.10	Array	
UInt	VT_UI2	W	DB1,W5	YES	None
		Word	DB1,Word5		
UDInt	VT_UI4	DW	DB1,DW5	YES	None
		Dword	DB1,Dword5		
SInt	VT_I2	C	DB1,C5	YES	None
		Char	DB1,Char5		
Struct	Not defined	Struct	An OPC Node is created for each structure defined and all its children will be parsed as child of this node.	YES	None

7.3.1.15.2Import Wizard

The Softing Configurator allows the import of PLC connection items from an external file.

The import action can be triggered by using the *Import from file* button  , by pressing the keyboard shortcut CTRL+I or by selecting the *Import* action in the context sensitive menu.

Note: The import action is available for the root item and for items of type *Node* from the namespace.

Attention: The import action is not available for items of type *Tag*.

© Softing Industrial Automation GmbH

Note: Name conflicts between the old and new data items are solved by renaming the newly imported items. They are renamed by appending a two-digit number at the end of the name of the first duplicated item.

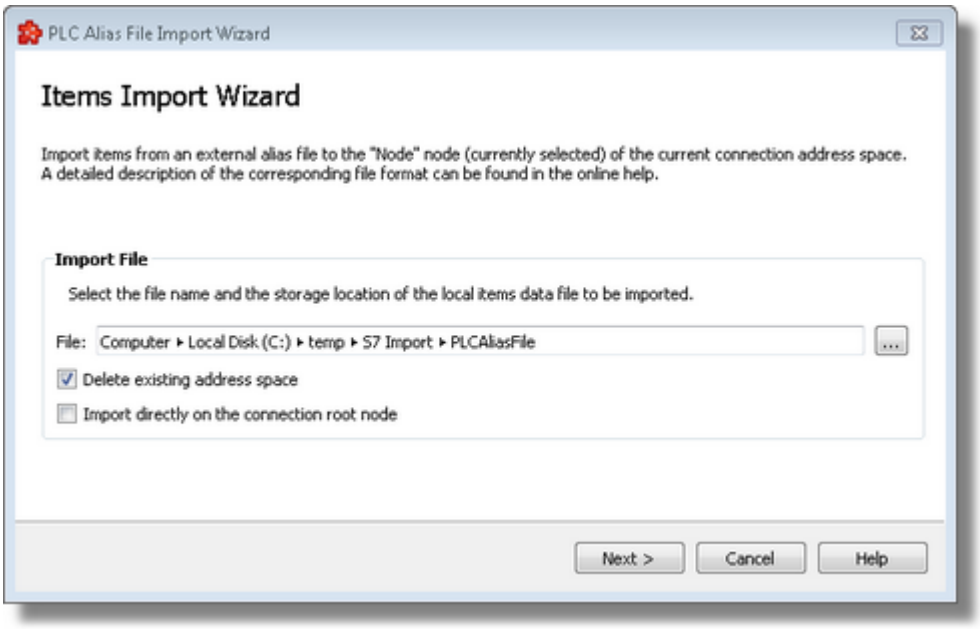
When the import action is triggered, the *Filter Import Wizard* is started.

▾ The Filter Import Wizard allows importing different data items:

- Select the file to be imported @ Import Wizard - Import File Selection
- Select the connection to import symbols from @ Import Wizard - Import Symbols Selection
- Examine the import results @ Import Wizard - Import Results

The Import File Selection wizard page allows you to choose a file to be imported.

The file can be selected by manually typing its name in the *File* input box or by clicking the browse button located to the right of the input box (a standard file selection dialog will open).



The file path and name should be valid (they should point to an existing file).

The already existing address space can be deleted. It is overwritten with the imported address space.

In addition it is allowed to merge the existing address space with the newly imported one. The merging conflicts are resolved by overwriting the existing identifiers with the newly imported ones.

Note: If item identifiers for items that are used as data sources or data destination items cause a conflict, the import operation will be aborted and the existing address space will be kept.

The delete address space option can not be enabled if at least one item is used as data source or data destination inside the starting point of the import operation.

In case when an import if alias file is requested on a node from the address space different the connection root node, the "***Import directly on the connection root node***" option is available. when this option is enabled, independent of the selected node, the import is executed directly on the connection root node.

When you press *Next*, the file path and name are verified. If the file identified by the specified file path and name does not exist, an error message is displayed.

The actual import begins if the selected file is valid.

After pressing *Next*, the file is analyzed and there are two paths to continue the import operation:

1. If the specified file contains only one single set of symbols, the wizard is continued with the import operation itself. If the import takes longer to complete, a waiting dialog is displayed. When the import operation is finished, the *Import Results* page is displayed.

2. If the specified file contains definitions of symbols for multiple connections (i.e. a Multiprotocol OPC Server symbols file was specified for import), the wizard is continued with the *Symbols Selection* page.

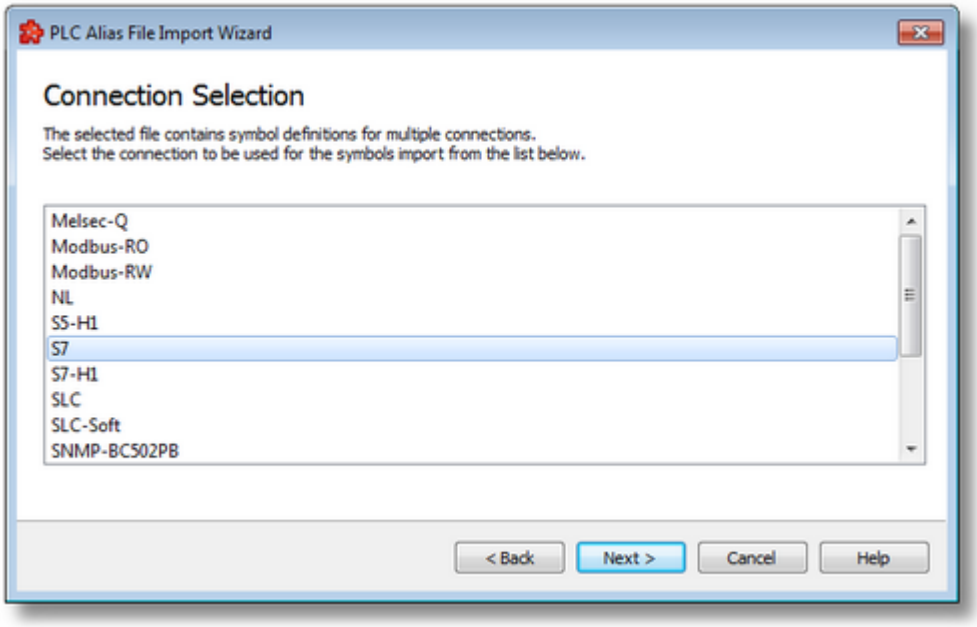
Import Wizard - Import Symbols Selection >>>

Import Wizard - Import Results >>>

The Symbols Selection wizard page allows you to choose a connection to import symbols from.

A list with all available PLC connections defining symbols in the selected file is presented and you must select the desired connections within it.

© Softing Industrial Automation GmbH

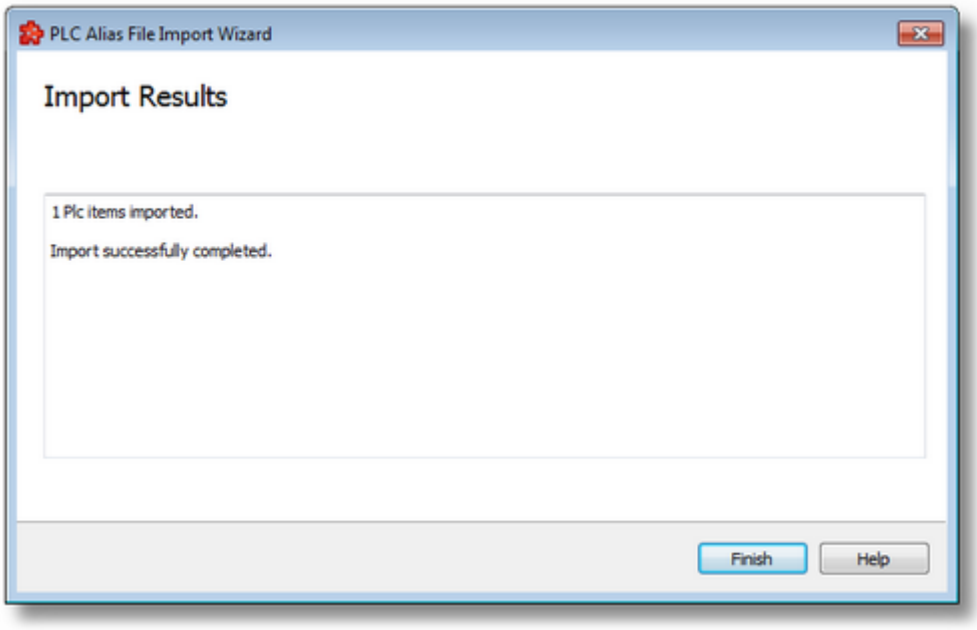


Note: It is user's responsibility to choose a proper connection (matching the protocol of the currently operated PLC connection).

The actual import begins after pressing *Next*. If the import takes longer to complete, a waiting dialog is displayed. When the import operation is finished, the *Import Results* page is displayed.

Import Wizard - Import Results >>>

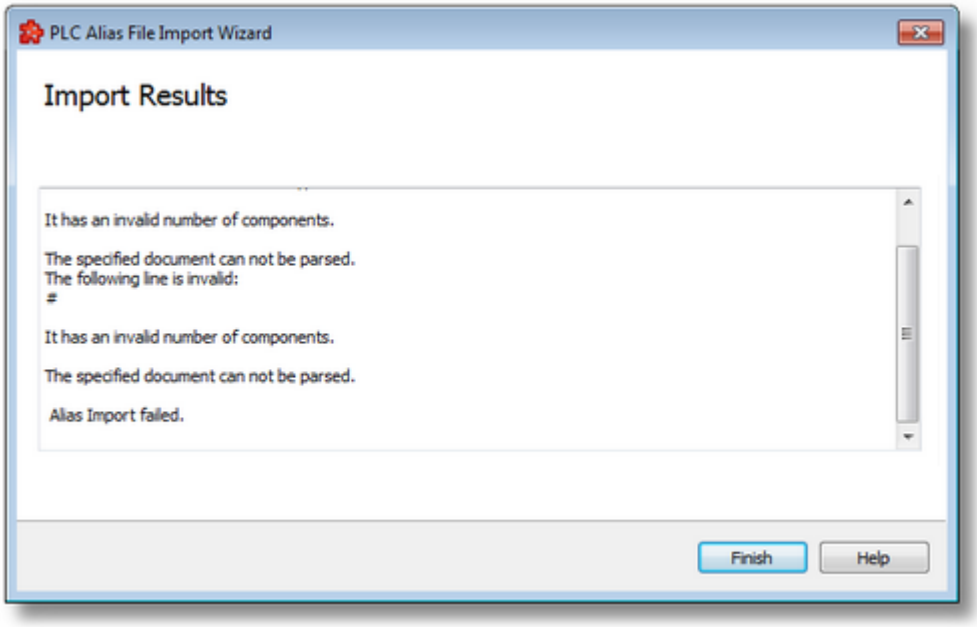
The Import Results wizard page displays the import results.



This page is the final page of the Filter Export Wizard. A back button is not provided as the import action cannot be undone. The import can be successful or it can fail:

1. After a successful import the results page contains the following information:
   - The absolute path of the imported file.
   - The number items imported. This number includes only the items of type Tag, the items of type Node are not counted.
   - The confirmation message that the import was successful.
2. After a successful import the results page contains the following information:
   - The invalid line where the import failed.
   - The reason why the line is considered invalid.
   - The confirmation message that the import failed.

An *Import Results* example for a failed import operation can be seen below:



When the import wizard is closed after a successful import, the new items are added to the current PLC connection tree as a branch of the currently selected node item.

<<< Import Wizard - Import File Selection

7.3.1.15.3Export Wizard

The dataFEED OPC Suite Configurator allows exporting the PLC connection items. The entire namespace tree or just a sub-branch of the PLC connection can be exported.

© Softing Industrial Automation GmbH

The export action can be triggered by pushing the *Export* button  , by pressing the keyboard shortcut CTRL+E or by selecting the *Export* action in the context sensitive menu. The export action is available for the root item and for any element of type *Node* of the namespace.

Attention: The export action is not available for nodes of type *Tag* or *Property*.

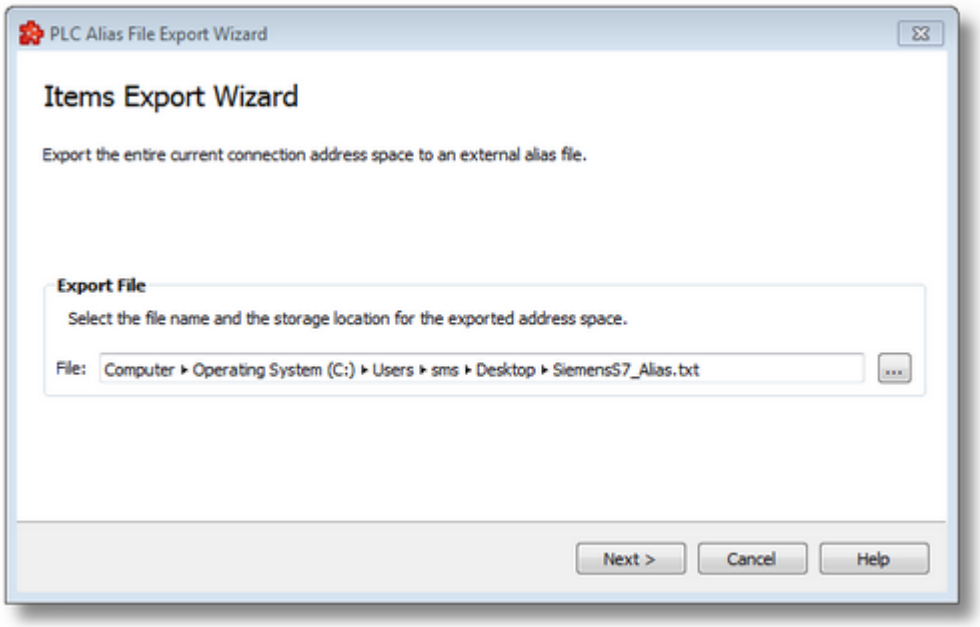
The exported file will contain all the items from the branch of the selected node.

When the export action is triggered the *Filter Export Wizard* is started.

▾ The Filter Export Wizard allows exporting different data items:

- Select the file the data items are to be exported to @ Export Wizard - Export File Selection
- Examine the export results @ Export Wizard - Export Results

The Export File Selection wizard page allows you to choose a file name and location for the exported file. The file can be selected by manually typing its path and name in the *File* input box or by clicking the browse button located to the right of the input box (a standard file saving dialog will open).



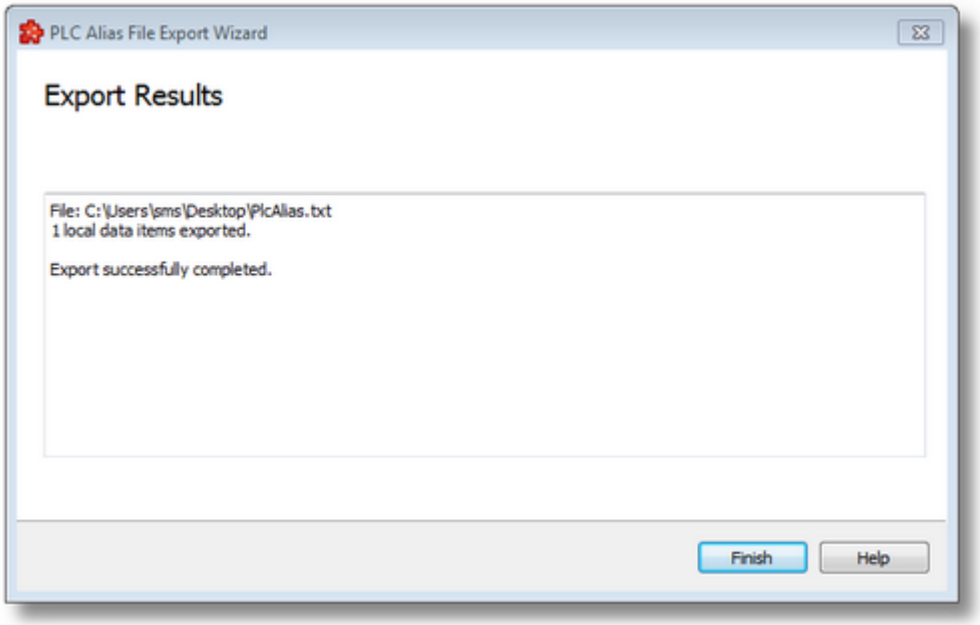
When the *Next* button is pressed the file path is verified and if the path does not exist, an error message is displayed (invalid path). If the specified file path is valid and a file with the specified name already exists, a message box will ask whether to overwrite the existing file or not (select

another file). The actual filter export operation begins if the path to the file is valid and either a file with the specified name does not exist or it exists and overwrite has been chosen.

When the export operation is finished the *Export Results* page is displayed.

Export Wizard - Export Results >>>

The Export Results wizard page displays the export results as shown in the image below:



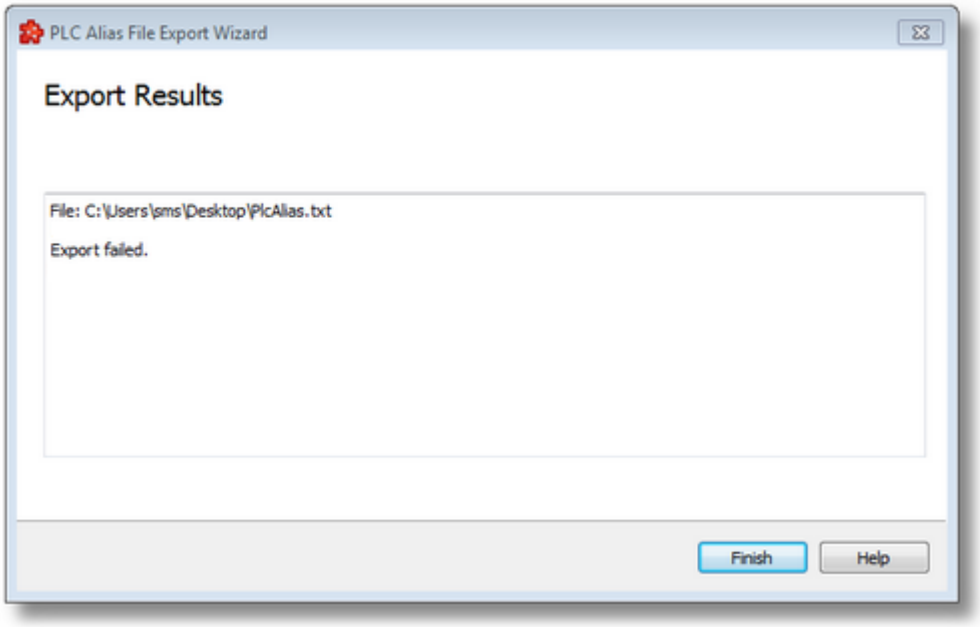
This page is the final page of the Filter Export Wizard. A back button is not provided as the export action cannot be undone. The export can be successful or it can fail:

1. After a successful export the results page contains the following information:
   - The absolute path of the exported file.
   - The number of filter items exported. This number includes only the items of type Tag and Property, the data items of type Node are not counted.
   - The confirmation message that the export was successful.

2. After a failed export the results page contains the following information:
   - The reason why the export failed.
   - The confirmation message that the export failed.

An *Export Results* example for a failed export operation can be seen below:

© Softing Industrial Automation GmbH



<<< Export Wizard - Export File Selection

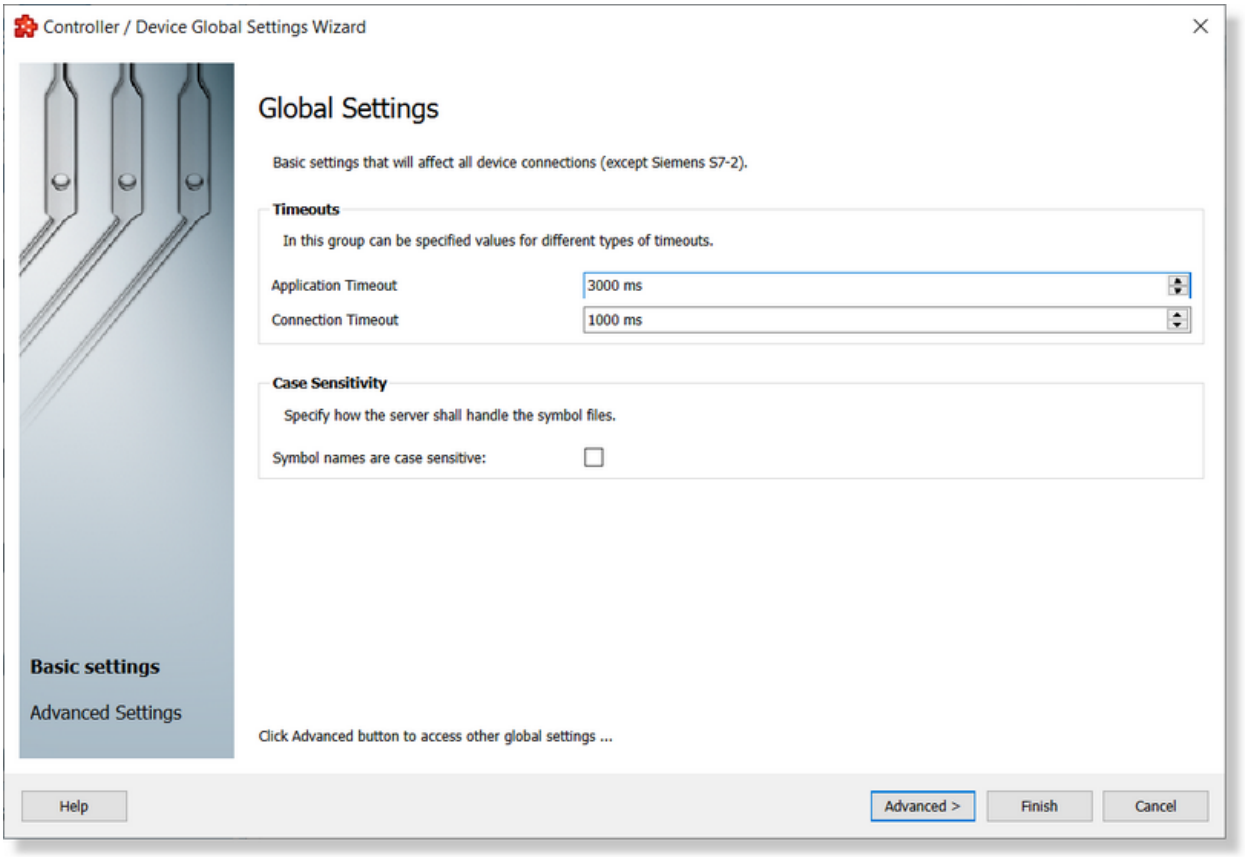
7.3.1.15.4Global Settings Wizard

With the Gloabal Settings Wizard you can modify connection settings that will be applied to all device connections:

- Specify basic settings @ Global Settings Wizard - Basic
- Specify advanced settings @ Global Settings Wizard - Advanced

In this page you can change settings related with the connections timeouts and symbols case sensitivity:

- Application timeout (time between the sending of a read or write to a PLC and receiving an answer from the PLC). When this timeout has expired and the PLC doesn't send an answer, the connection is reestablished.
- Connection timeout (in this time period the servers tries to build up a connection to the PLC, the value "0" disables the functionality). When a connection couldn't be established, the quality bad not connected is sent after this time period to the OPC client.
- How the server shall handle the symbol files: case sensitive or not (default no). The part of the symbol that addresses the connection is case sensitive anyway.

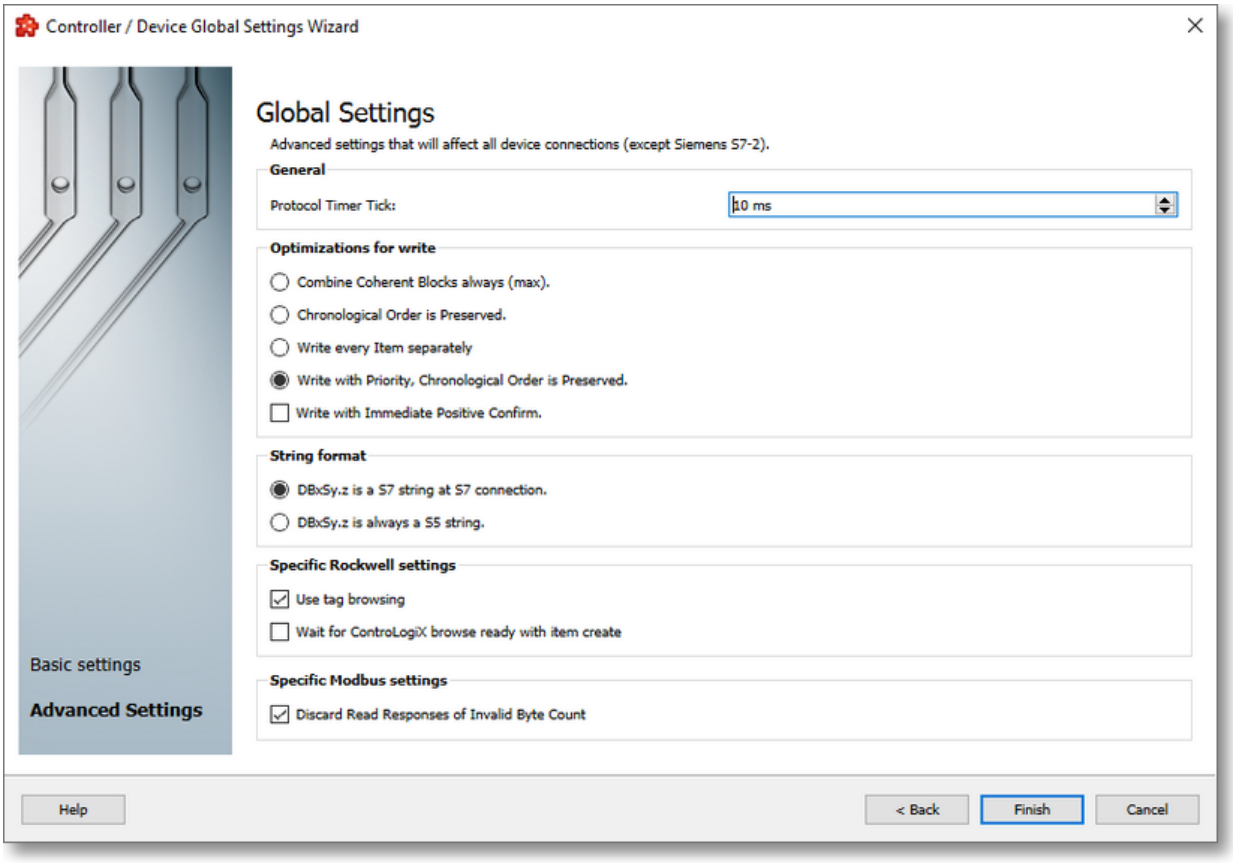


Press the *Advanced* button to proceed with the next page or *Finish* button to exit the wizard at this point. You can abort the operation with the *Cancel* button.

Global Settings Wizard - Advanced >>>

In this page you can change more advanced settings that will affect all or some specific device connections.

© Softing Industrial Automation GmbH



Protocol timer tick (interval)

At the time intervals specified here the CPU releases computing time for the OPC server. The OPC server then checks to determine whether data need to be processed. The default setting can usually be retained.

Attention: If the utilization load is very high, this can be reduced by assigning a greater value in the timer protocol.

Optimization for write

Write optimization takes effect differently depending whether the high performance protocol is enabled or not for the applicable access path.

Write Optimization	HPP = No	HPP = Yes
Related blocks always placed together (max.)	With multiple write accesses to an item, only the last value should be written.	No effect

Time sequence is retained.	Possible	If HPP=Yes, the time sequence should always be adhered to for all set write optimizations.
Prioritized write accesses, time sequence is retained.	Possible	Should not have any effect

The following tables depict all the optimization for write related configuration options, each with a brief description.

Attribute	**Description**
Combine Coherent Blocks always (max)	This maximum form of optimization causes fewer PLC frames to be transmitted which reduces the load. When a counter counts up very fast, the respective last value (or the current value at the moment) is written to the PLC.
Chronological Order is Preserved	The items that are written are arranged chronologically one behind the other and, after successfully read accessing of the applicable DBs, are written to the PLC. Related blocks are placed together. Overlapping items are written separately. Individual bits of a word are also written individually. This ensures that a bit that can also be set and deleted very quickly and can also be forwarded to the PLC. This ensures the signal flow.
Write every Item separately	Every item is transmitted individually. Nothing is optimized. The procedure is the same as with *Chronological Order is Preserved*.
Write with Priority, Chronological Order is Preserved	Write accesses are handled with a high priority. A read access that is currently running is finished first, however. Afterwards, the write access is then initiated immediately. If several items of a block are to be write-accessed, these are written one after the other in the order that they were received. The block is then read back so that the user receives an immediate response message. If items of another block are still to be written, the write procedure is initiated without delay and continued as described above. This write optimization may somewhat delay the entire write procedure since the area that was written must also be read out again.

© Softing Industrial Automation GmbH

Write with Immediate Positive Confirm	Default value: *Unchecked* Waiting for the acknowledgement of a write procedure on the PLC may block the client which can be a nuisance. In this case, we recommend "Write with immediate positive acknowledgement". When this optimization is selected, the user receives an immediate positive acknowledgement and not the actual result of the write access. The option can be selected to speed up synchronous write procedures on the PLC since you don't have to wait for the acknowledgement. This setting is intended for clients that cannot select asynchronous write accesses

String format

S5 and S7 strings are organized differently. S5 strings do not have length information while the first two bytes of an S7 string contain this information.
Since the item syntax for S5 and S7 strings is identical (DBxSy.z), you set here whether the string format is to be interpreted as an S7 string or an S5 string.
It is standard to interpret the string as an S5 string with an S5 access path and an S7 string with an S7 access path. This setting can be changed to interpret every string as an S5 string.

Tag Browsing

When an OPC server has a great number of data points in the icon table, these are transmitted to the OPC client on request. This requires working storage space and computing time. When the commissioning of OPC communication is concluded and this functionality is no longer needed, it can be switched off. This reduces the time that the server requires to boot after a new start.

Wait for ControLogix browse ready

After program start, the tags of a CLX connection are compared with the online PLC. Under some circumstances, this can take quite a while. If data points are registered for such a connection, this may cause the server to output an error message saying that the item is not available. This option can be enabled to keep this from happening. The registration of an item blocks the OPC client until CLX browsing has been concluded.

Discard Read Responses of Invalid Byte Count

Enabling this option, TCP retransmissions of Modbus Read responses will be discarded.

<<< Global Settings Wizard - Basic

7.3.1.15.5Item Syntax

This chapter describes the item syntax which can be used to set up items (tags) for the corresponding connection.

- S7 Item Syntax
- S5 Item Syntax
- Modbus Item Syntax
- PLC-5 / SLC Item Syntax
- MELSEC-Q Item Syntax
- Send/Receive Item Syntax

General aspects

Arrays

The word array means a series of equal elements (field, row, data area). An array combines several units of one data tpye in a field. To create an array, the length of the array is added to the standard syntax, separated by a period.

Arrays are NOT possible for: BOOL, DATE AND TIME, DATE, TIME, TIME OF DAY, STRINGS (is already an array of characters);

as well as for all suffixes which are string or bool to the client: BA (already an array of bits), KA, S5T, KT, D, T, DT, DU, TOD

Suffixes

Bit Masking

With a bit mask several bits of a data type can be read or written, by combining them to one decimal value. Bit maskign of arrays is possible as well.

Syntax

8 bits of data word 0 of data block 5 shall be read beginning with bit 2. Therefore the following syntax is used:

- db5.w0#2.8 oder

- db5.w0#2,8 oder

- db5.w0#2#8

Examples

© Softing Industrial Automation GmbH

```
DB10.W03.4
DB10.DWord2#20.10
DB10.DWord6#20.10KF
DB10.DInt10#20#10
```

Note: Does not work with Left Byte or Right Byte (DL, DR) and Suffix BA

Note: Do not mask bits of REAL or KG

The S7 item syntax is set up as shown below:

[Group.]<Area><Data type><Start address>[.Array size][Suffix]

If the data type is BOOL, the bit number is required.

[Group.]<Area><Data type><Start address><.Bit no.>[.Array size][Suffix]

Legend: <> Mandatory [ ] optional

<Area>

	Syntax	**Orientation1**	**>Access Rights**	**Notes**
Data block	DBx. V synonym DB1	BYTE	Read / write	With blocks, specification of a block number x is required (x = 1 to 65535). A period or a comma must appear after the block number.
Instance block	DIx.	BYTE	Read / write	
Flag	M or F	BYTE	Read / write	
Timer	T	WORD	Read	
Counter	Z or C	WORD	Read / write	

Input	E or I	BYTE	Read	
Output	A or O or Q	BYTE	Read / write	
I/O input	PE or PI	BYTE	Read	
I/O output	PA or PO or PQ	BYTE	Read / write	

[1] BYTE-oriented means that a byte is addressed for each physical address.

WORD-oriented means that a word (16 bits) is addressed for each physical address.

<Data type>

Type	**Syntax**	**DB / DI**	**F (M)**	**I/O (E/A)**	**PE /PA**	**C/T**	**With Array**	**Useful Suffix**	**OPC Data type**
BIT VT_BOOL	X	DB 5.X 4.3	MX 1.3	------	------	-------	------	------	-------
BIT VT_BOOL		DB 5.4.3	M1.3	E4.3 I4.3 A4.3 O4.3	PE 4.5 PI4.5 PA 1.3 PO 1.3	-------	------	------	-------
BYTE VT_UI1	B Byte	DB 5.B 2 DB 5.Byte 2	MB 4 MByte 4 FB 4 FByte 4	EB 4 EByte 4 IB4 IByte4	PEB4 PEByte4 PIB 4 PIByte 4	-------	DB 5.B 2.4 DB 5.Byte 2.4 MB 4.3	BA BCD RI RU	VT_BOOL | VT_ARRAY VT_I2 VT_R4 VT_R4

© Softing Industrial Automation GmbH

				AB5 AByte5 OB5 OByte5	PAB5 PAByte5 POB5 POByte5		MByte4.3 FB4.4 FByte4.4 POB5.3 etc.		
WORD VT_UI2	W Word	DB5.W3 DB5.Word3	MW4 MWord4 FW4 FWord4	EW4 EWord4 IW4 IWord4 AW5 AWord5 OW5 OWord5 QWord5	PEW4 PEWord4 PIW4 PIWord4 PAWord5 PAWord5 POW5 POWord5	C5 Z5 T5	DB5.W3.2 DB5.Word3.2 MW4.2 EWord4.2 PAWord5.3 C5.3 Z5.10 T5.2 etc.	BA BCD D DT DU KF KT RI RU S5T SWAB TR	VT_BOOL | VT_ARRAY VT_UI2 VT_BSTR VT_BSTR VT_BSTR VT_I2 VT_BSTR VT_R4 VT_R4 VT_BSTR VT_UI2 VT_R4

INT VT_I2	I Int	DB 5.I3 DB 5.Int3	MI4 MInt4 FI4 FInt4	EI4 EInt4 II4 IInt4 AI5 AInt5 OI5 OInt5	PEI4 PEInt4 PII4 PIInt4 PAI5 PAInt5 POI5 POInt5	-------	DB 5.I3.2 DB 5.Int3.2 MI4.4 FInt4.3 AInt5.3 OI5.2 OInt5.5 POInt5.4 etc.	BA BCD DT DU KT RI RU S5T SWAB TR	VT_BOOL | VT_ARRAY VT_I2 VT_BSTR VT_BSTR VT_BSTR VT_R4 VT_R4 VT_BSTR VT_I2 VT_R4
REAL VT_R4	R Real	DB 5.R2 DB 5.REAL2	MR4 MREAL4	ER4 EREAL4 IR4 IREAL4 AR4 AREAL4 OR4	PER4 PEREAL4 PIR4 PIREAL4 PAR4 PAREAL4	------	DB 5.R2.2 DB 5.REAL2.4 MR4.5 ER4.4 AREAL4.5 PER4.2	------	-------

© Softing Industrial Automation GmbH

				OREAL4	POR4 POREAL4		PAR4.7 POR4.3 etc.		
STRING VT_BSTR	S [3] String	DB5.S1.2 DB5.String1.2	MS2.3 MString2.3	------	------	-------	DB5.S1.2.5 DB5.String1.2.5	KA	VT_BSTR
S7-STRING VT_BSTR	G [3]	DB5.G2.2	MG2.3	------	------	-------	------	KA	VT_BSTR
DOUBLE WORD VT_UI4	DW Dword	DB5.DW3 DB5.Dword3	MDW4 MDWord4 FDW4 FDWord4	EDW4 EDWord4 IDW4 IDWord4 ADW5 ADWord5 ODW5 ODWord5	PEDW4 PEDWord4 PIDW4 PIDWord4 PADW5 PADWord5	-------	DB5.DW3 MDW4.2 FDW4.2 EDW4.4 ADWord5.8 PEDW4.8	BA BCD KF KG RI RU S5T SWAB T TOD	VT_BOOL | VT_ARRAY VT_I4 VT_I4 VT_R4 VT_R4 VT_R4 VT_BSTR VT_R4 VT_BSTR VT_BSTR

				QD W5	POD W5 POD Word5		PAD W5.4 POD W5.4 etc.		
DOUBLE INT VT_I4	D DI DInt	DB 5.D 3 [2] DB 5.DI3 DB 5.DInt3	MD I4 MD Int4 FDI 4 FDInt4	EDI 4 EDInt4 IDI 4 IDInt4 ADI 5 ADInt5 ODI5 ODInt5 QDI5	PEDI4 PEDInt4 PIDI4 PIDInt4 PADI5 PADInt5 PODI5 PODInt5	-------	DB 5.D 3.2 DB 5.DI3.2 DB 5.DInt3.4 MDI4.5 EDI 4.5 ADI 5.2 PEDI4.5 PADInt5.2 etc.	BA BCD KG RI RU S WAB T TOD	VT_BOOL | VT_ARRAY VT_I4 VT_R4 VT_R4 VT_R4 VT_I4 VT_BSTR VT_BSTR

[2] Caution! Danger of mixup with S5 syntax for bit (DB5D1.2)

[3] With SF you can force an S5 string. With SS you can force an S7.

Notes on counters and timers

Counters and timers are always addressed by words. For this reason, the specification of a data type is not required!

© Softing Industrial Automation GmbH

The start address directly follows the “T” or “C/Z” area. Timers can only be read! Counters can be read and written.

Timer values are indicated in seconds (e.g., T = 0.7 => T = 0.7 s = 700 ms).

Counters are represented in decimals (0 to 999).

<Start address>

The start address specifies the addresses starting at which can be read or written.

Example: DB5.DW6: Double word 6 of the data block 5 is the start address.

Example: MB17: Flag byte 17 is the start address.

If the start address is a certain bit, the bit number must also be specified.

<.Bit number>

The bit number must always be specified when the data type is BOOL.

Example: I4.3: bit 3 of input byte 4 – an input bit is addressed here.

Example: MX12.1: Bit 1 of flag byte 12 – a flag bit is addressed here.

[.Array size]

An array (i.e., field, row, data area) is a series of equal elements. An array combines several units of one data type into a field. If, for example, several words are read out from a data block, this is called an array of words. To create an array, the length of the array is added to the standard syntax separated by a dot.

Example: DB10.REAL2.5.

The S5 syntax has the following structure.

[Group.]<Area><Data type><Start address>[.Array size][Suffix]

The bit number must be specified if the data type is BOOL.

[Group.]<Area><Data type><Start address><.Bit no.>[.Array size][Suffix]

Legend: <> mandatory [ ] optional

[Group.]

Operand areas can be combined into groups. If there are gaps in the I/O (e.g., O10.1 and O10.3), these can be skipped by creating different groups. The gaps are then not requested.

The group name is always set in front of the item name and separated by a period (Groupname.Itemname). The group name consists of “G” and the number.

Example:

G1.O10.1

G2.O10.3

The gap is skipped. The data points are accessed separately and not combined.

<Area>

	Syntax	**Orientation 1**	**Access Rights**	**Notes**
Data block	DBx.	WORD	Read / write	With blocks, a block number x is required (x = 1 to 65535).
Expanded block	DXx.	WORD	Read / write	
Flag	M or F	BYTE	Read / write	
Timer	T	WORD	Read	
Counter	Z or C	WORD	Read	
Input	E or I	BYTE	Read	
Output	A or O or Q	BYTE	Read / write	
I/O	P	BYTE	Read / write	
Expanded I/O	OB	BYTE	Read / write	
System area		WORD	Read	
Absolute memory cells	AS	WORD	Read	

[1] BYTE-oriented means that a byte is addressed for each physical address.

WORD-oriented means that a word (16 bits) is addressed for each physical address.

<Data type> for data blocks and expanded data blocks

© Softing Industrial Automation GmbH

	Syntax	Example	with Array	Useful Suffix	OPC Data Type
BIT VT_BOOL	D	DB5 D4.12	-------	-------	-------
BYTE VT_UI1	DB	DB5 DB3	DB5 DB3.5	BA	VT_BOOL | VT_ARRAY
LEFT BYTE VT_UI1	DL	DB5 DL4	DB5 DL4.2	BA	VT_BOOL | VT_ARRAY
RIGHT BYTE VT_UI1	DR	DB5 DR2	DB5 DR2.5	BA	VT_BOOL | VT_ARRAY
WORD VT_UI2	DW	DB5 DW4	DB5 DW4.5	BA BCD D DT DU KF KT RI RU S5T SWAB TR	VT_BOOL | VT_ARRAY VT_UI2 VT_BSTR VT_BSTR VT_BSTR VT_I2 VT_BSTR VT_R4 VT_R4 VT_BSTR VT_UI2 VT_R4
DOUBLE WORD VT_UI4	DD	DB5 DD3	DB5 DD3.2	BA BCD IEEE KF	VT_BOOL | VT_ARRAY VT_I4 VT_R4 VT_I4

				KG RI RU SWAB T TOD	VT_R4 VT_R4 VT_R4 VT_I4 VT_BSTR VT_BSTR
STRING VT_BSTR	S	DB5 S2.3	-------	-------	-------

<Data type> for all other areas

	Syntax	**M (F)**	**I/O**	**P / OB**	**C / T / RS / AS**	**with Array**	**Sign. Suffix**	**OPC Data Type**
BIT VT_BOOL		M4.3 F4.3	E4.3 I4.3 A4.3 Q4.3	----------	----------	----------	-------	-------
BYTE VT_UI1	B	MB4 FB4	EB4 IB4 AB5 QB5	PB4 OB4	----------	MB4.3 FB4.4 EB4.4 IB4.5 AB5.2	-------	-------

© Softing Industrial Automation GmbH

						QB 5.5 PB 4.2 OB 4.3		
WORD VT_UI2	W	MW 4 FW 4	EW 4 IW4 AW 5 QW 5	PW 2 OW 2	C5 T5 RS 4 BS 4 AS5	MW 4.2 FW 4.4 EW 4.5 IW4 .2 AW 5.5 QW 5.2 PW 2.1 0 OW 2.3 C5. 5 T5. 3 RS 4.2 BS 4.7 AS5 .2	BA BCD D DT DU KF KT RI RU S5T SWAB TR	VT_BOOL | VT_ARRAY VT_UI2 VT_BSTR VT_BSTR VT_BSTR VT_I2 VT_BSTR VT_R4 VT_R4 VT_BSTR VT_UI2 VT_R4
DOUBLE WORD VT_UI4	D	MD 4	ED 4 ID4	PD 5	------	MD 4.4	BA BCD	VT_BOOL | VT_ARRAY VT_I4

		FD 4 AD 5 QD 5	OD 5		FD 4.2 ED 4.6 ID4.2 AD 5.6 QD 5.2 PD 5.4 OD 5.9	IEEE KF KG RI RU SWAB T TOD	VT_R4 VT_I4 VT_R4 VT_R4 VT_R4 VT_I4 VT_BSTR VT_BSTR
STRING VT_BSTR	S	MS 4.5 FS4.5	------	------	------	------		-------

Notes on counters and timers

Counters and timers are always addressed by words. For this reason, the specification of a data type is not required! The start address is directly connected to the “T” or “C/Z” area. Timers can only be read! Counters can be read and written. Timer values are indicated in seconds (e.g., T = 0.7 => T = 0.7 s = 700 ms).

Counters are represented with decimals (0 to 999).

<Start address>

The start address specifies the addresses starting at which can be read or written.

Example: DB5.DW6: Double word 6 of the data block 5 is the start address.

Example: MB17: Flag byte 17 is the start address.

If the start address is a certain bit, the bit number must also be specified.

<.Bit number>

The bit number must always be specified when the data type is BOOL.

Example: I4.3: bit 3 of input byte 4 – an input bit is addressed here.

Example: MX12.1: Bit 1 of flag byte 12 – a flag bit is addressed here.

© Softing Industrial Automation GmbH

[.Array size]

An array (i.e., field, row, data area) is a series of equal elements. An array combines several units of one data type into a field. If, for example, several words are read out from a data block, this is called an array of words. To create an array, the length of the array is added to the standard syntax separated by a dot.

Example: DB10DW3.4

[Suffix]

A value can be represented in another format with the aid of a suffix. The S5 syntax without suffix a data type without a sign. A data type with a sign requires a suffix.

Example: DB10DW3.4BCD.

The Modbus Item syntax is set up as shown below.

```
[Group.]<Area><Data type><Start address>[.Array size][Suffix]
```

If the data type is BOOL, this requires the bit number.

```
[Group.]<Area><Data type><Start address><.Bit number>[.Array size][Suffix]
```

If the connection "UnitID individual with item syntax" is selected, the following applies:

```
[UnitID.][Group.]<Area><Data type><Start address><.Bit number>[.Array size][Suffix]
```

Legend: <> Mandatory [ ] Optional

[Group.]

Operand areas can be combined into groups.

The area: 0 - 65535. If "UnitID individual with item syntax" is selected, the area is reduced to 0 - 255.

If there are gaps in the I/O, these can be skipped by creating different groups. The gap is then not requested.

The group name is placed in front of the item name and separated by a dot

Gruppenname.Itemname (Groupname.Itemname).

The group name is made up of G and a number.

Example:

G1.40001

G2.R2

G3.S20.30

[UnitID.]

The UnitId is specified for this item with the item syntax.

For this the Unitname is placed set in front of the item name and separated by a dot (Unitidname.Itemname).

The Unitidname is made up of the Id and the number.

The area: 0 - 255

When a group is specified, the area of the group numbers is reduced to 0 - 255.

If the UnitID is missing in the Itemsyntax, the parameterized UnitID is transferred to the PLC. Example:

Id1.40001

Id2.R2

Id3.S20.30

Id1.G2.R3 UnitId + Group

<Area>

	Syntax Abbrev.	**Abbrev. Number**	**Orientation 1**	**Access rights**
Discrete Inputs	I E DI DE	1xxxxx	BIT	Read
Discrete Outputs	A O Q DA DO DQ	0xxxxx	BIT	Read / write
Input Register	ER	3xxxxx	WORD	Read

© Softing Industrial Automation GmbH

	IR			
Register (Holding Register)	R HR	4xxxxx	WORD	Read / write
Discrete Inputs Octal [2]	J	-	BIT	Read
Discrete Outputs Octal [2]	P	-	BIT	Read / write

[1] BIT-oriented means that one bit is addressed for each physical address.

WORD-oriented means that one word (16 bits) is addressed for each physical address.

[2] Entry of the start address is octal and the numbers 8 and 9 are invalid characters. Internally, the address is handled decimally and must be considered for logger and status.

Discrete inputs and outputs occupy 1 bit on the PLC. With read and write accesses they are processed as 8-bit values. This means that before the write access, 8 bits are read. With High Performance Protocol enabled it is possible to write a single bit.

Areas can either be addressed via the above stated alphabetic sequence or via a number. This means that a discrete input can be addressed by the abbreviation “E” the same as with the number “1.”

<Data type>

	Syntax	**Syntax R**	**Syntax E / A**	**Syntax ER**	**with Array**	**Useful Suffix**	**OPC Data Type**
BIT VT_BOOL	X	RX5.2 HRX5.2 4X5.2	E255 I255 DE255 DI255 125543	----	----	----	----
INT VT_I2 (signed)	None I	R50 HR50	----	ER120 IR120	R50.2 HR50.2	BCD D WDT*	VT_I2 VT_BSTR VT_BSTR

		400050 RI50 HRI50 4I150		312034 ERI120 IRI120 3I12034	40005 0.2 ER120.2 IR120.2 31203 4.2	KT RI RU S5T SWAB BA	VT_BSTR VT_R4 VT_R4 VT_BSTR VT_I2 VT_BOOL|VT_ARRAY
WORD VT_UI2 (unsigned)	W	RW50 HRW50 4W50	----	ERW120 IRW120 3W12034	RW50.2 HRW50.2 4W00050.2 ERW120.2 IRW120.2 3W12034.2	BCD D WDT* KT RI RU S5T SWAB BA	VT_I2 VT_BSTR VT_BSTR VT_BSTR VT_R4 VT_R4 VT_BSTR VT_I2 VT_BOOL|VT_ARRAY
DOUBLE INT VT_I4 (signed)	D DI	RD50 HRD50 4D50 RDI50 HRDI50 4DI50	----	ERD120 IRD120 3D12034 ERDI120 IRDI120 3DI12034	RD50.2 HRD50.2 3D12034.2 ERDI120.2 IRDI120.2 3DI12034.2	BCD D WDT* KT RI RU S5T SWAB BA	VT_I2 VT_BSTR VT_BSTR VT_BSTR VT_R4 VT_R4 VT_BSTR VT_I2 VT_BOOL|VT_ARRAY
DOUBLE WORD VT_UI4 (unsigned)	DW	RDW50 HRDW50	----	ERDW120	RDW50.3 HRDW50.3	BCD SWAB KG	VT_I4 VT_I4 VT_R4

© Softing Industrial Automation GmbH

		4DW50		IRDW120 3DW12034	4DW50.3 ERDW120.3 IRDW120.3 3DW12034.3	IEEE TR RI RU T WDT [3] BA	VT_R4 VT_R4 VT_R4 VT_R4 VT_BSTR VT_BSTR VT_BOOL|VT_ARRAY
DOUBLE DOUBLE INT VT_I8 (signed)	DD DDI	RDD50 RDDI50	----	ERDD50 ERDDI50	RDD50.2 RDDI50.2 ERDD50.2 ERDDI50.2	BCD RI RU SWAB BA	VT_I8 VT_R8 VT_R8 VT_I8 VT_BOOL|VT_ARRAY
DOUBLE DOUBLE WORD VT_UI8 (unsigned)	DDW	RDDW50	----	ERDDW50	RDDW50.3 ERDDW50.3	BCD RI RU SWAB BA	VT_I8 VT_R8 VT_R8 VT_UI8 VT_BOOL|VT_ARRAY
REAL VT_R4	R	RR5	----	ERR5	RR5.3 ERR5.2	----	----
DOUBLE VT_R8	RD	RRD5	----	ERRD5	RRD5.10	----	----
STRING VT_BSTR	S	RS5.4	----	ERS5.4	----	KA SWAB	VT_BSTR VT_BSTR

[3] With the suffix WDT (Wago Date and Time) 4 registers (8 bytes) are read.

<Start address>

The start address specifies the address starting at which read or write accesses begin.

Example: ER120: input register 120

If the start address is a certain bit, the bit number must also be specified.

<.Bit number>

The bit number must always be specified when the data type is BOOL!

Example: HRX5.2: Bit 2 of holding register 5

[.Array size]

Arrays are created to combine several units of one data type together in one field.

Example: HRD50.3

[Suffix]

A suffix is used to represent a value in another format.

Example: HRD50.3WDT (representation in Wago Date and Time format)

The item syntax for PLC-5 and SLC is shown below.

<Area>[File number]<Start address>[.Array size][Suffix]

Legend: <> Mandatory [ ] Optional

<Area>

File Type	Syntax	Orientation 1	Access Rights	Default File number	Address
Output	O	BIT	Read / Write	0	octal
Input	I	BIT	Lesen	1	octal
Integer	N	WORD	Read / Write	7	decimal
Binary	B	WORD	Read / Write	3	decimal
Float	F	DOUBLE WORD	Read / Write	8	decimal

© Softing Industrial Automation GmbH

String	ST	SLC-String	Read / Write	3	decimal
SFC-Status	SC	WORD	Read / Write	3	decimal
Status	S	WORD	Read / Write	2	decimal
Timer	T	WORD	Read / Write	4	decimal
Counter	C	WORD	Read / Write	5	decimal
Control	R	WORD	Read / Write	6	decimal
ASCII	A	WORD	Read / Write	3	decimal

[1] BIT-oriented means that one bit is addressed for each physical address.
WORD-oriented means that one word (16 bits) is addressed for each physical address.
DOUBLE WORD-oriented means that one double word (32 bits) is addressed for each physical address.

[File numer]

Specification of the file number is optional. If it is not specified, the default file number is used. See column marked Default File Number in the table.

<Start address>

The start address specifies the address starting at which read or write accesses begin. The start address can consist of 2 pieces of information. Word (floating) number and when a single bit is accessed, then the bit number. The word number can be omitted with a bit. The word number or the floating number is introduced by a colon (:). The bit number is then introduced with a slash (/) <:word> or <:float> or </bit>. A format can be entered behind the colon. If a format is not entered, the data format in the Orientation column is used.

Format	**Syntax**	**Item Format**
String	S	STRING
High Byte	H	WORD
Low Byte	L	WORD
Byte	B	WORD

The address is octal for some areas. It is decimal for others. See column labeled Address.

Either the word number, the float number or the bit number is specified as the start address.

<:word> or <:float> or </bit>

The following syntax is used to address a certain bit within a word.

<:word/bit>

NOTE: When bits are write-accessed, the whole word is written!

Syntax	Explanation
O:0	Word 0 in output file 0
O:0/12	Bit 10 (12 octal = 10 decimal) in output file 0
O/12	Bit 10 (12 octal = 10 decimal) in output file 0
I:37	Word 31 (37 octal = 31 decimal) in input file 1
I4:37/2	Bit 2 in word 31 ((37 octal = 31 decimal) in input file 4
I:1/0	Bit 0 in word 1 in input file 1
B3/26	Bit 26 in binary file 3
B12:5.15	Array of 15 words in binary file 12, starting at word 5
B12:5/15	Bit 15 in word 5 of binary file 12
F8:0	Float 0
F9:10.16	Array of 16 double words of float file 9, starting at double word 10
N23:4	Word 4 of integer file 23
N23:4.10	Array of 10 words in integer file 23, starting at word 4
N23:4/2	Bit 2 of word 4 in integer file 23 = bit 66 in integer file 23
N23/66	Bit 66 in integer file 23

[.Array size]

Arrays are created to combine everal units of one data type in a field together. Arrays are only possible for word areas and float areas.

Examples: N23:4.10

© Softing Industrial Automation GmbH

[Suffix]

A suffix can be used to represent a value in another format. If no suffix is specified, the formats from the Orientation column are used.

Two syntax versions are available for setting up items.

1. Simple version:

[Group.]<Area><Start address>[.Array size][Suffix]

2. Expanded version:

[Group.]<Area><.Type><Start address>[.Array size][Suffix]

Legend: <> Mandatory [ ] Optional

REMEMBER :

- With the expanded version, a period or a comma is required between the <area> and the <type>. If the period or comma is omitted, the syntax uses the simple version. The <type> then corresponds to the default type (for bit area BIT, for word area WORD – see table <Bereich> (Area)).
- If the representation of the start address is HEX, all numbers for this area are also HEX.
- The HEX / decimal numbers can be changed with the following prefixes.

Conversion of HEX -> DEC: Input of 0d (number zero + the letter d) before the decimal address

Conversion of DEC -> HEX: Input of 0x (number 0 + the letter x) before the hexadecimal address

- When words, double words or strings are registered in a bit area, the start address is a bit address and only possible on word boundaries (dec. 0/16/32... or hex: 0/10/20...) (e.g., Y.D10.3).
- Bit masking is possible. The bit mask is specified before the suffix (see bit masking).
- Bit arrays in bit areas are not possible.
- Bit arrays in WORD areas with HEX representation, are not possible because of the suffix BA (BitArray) is interpreted as a hexadecimal number.

[Group.]

Areas can be separated into groups of related addresses.

If the I/O has gaps, these can be left out by creating groups for read and write accesses. The addresses before the gap are put in the one behind and the ones behind the gap in the other one. The gap is not requested.

The letter G and the group number are set in front of the item names and separated by a dot (GNr.Itemname).

Example:

G1.Y10

G2.Y30

Areas in different groups are queried individually and are not combined.

<Area>

	Syntax	**Orientation 1**	**Representation of the Start Address**
Special Relay	SM	BIT	Decimal
Special Register	SD	WORD	Decimal
Input Relay	X	BIT	HEX
Output Relay	Y	BIT	HEX
Internal Relay	M	BIT	Decimal
Latch Relay	L	BIT	Decimal
Annunciator	F	BIT	Decimal
Edge Relay	V	BIT	Decimal
Link Relay	B	BIT	HEX
Data Register	D	WORD	Decimal
Link Register	W	WORD	HEX
Timer Contact	TS	BIT	Decimal
Timer Coil	TC	BIT	Decimal
Timer Current Value	TN	WORD	Decimal

© Softing Industrial Automation GmbH

Retentive Timer Contact	SS	BIT	Decimal
Retentive Timer Coil	SC	BIT	Decimal
Retentive Timer Current Value	SN	WORD	Decimal
Counter Contact	CS	BIT	Decimal
Counter Coil	CC	BIT	Decimal
Counter Current Value	CN	WORD	Decimal
Special Link Relay	SB	BIT	HEX
Special Link Register	SW	WORD	HEX
Step Relay	S	BIT	Decimal
Direct Input	DX	BIT	HEX
Direct Output	DY	BIT	HEX
Index Register	Z	WORD	Decimal
File Register (Normal Access by block Switching)	R	WORD	Decimal
File Register (Serial No. Access)	ZR	WORD	Decimal

[1] BIT-oriented means that one bit is addressed for each physical address. WORD-oriented means that one word (16 bits) is addressed for each physical address.

<.Type> <,Type>

	Syntax	**Simple Syntax**	**Ex.: Bit Area**	**Ex.: Word Area**	**with Array Bit Area**	**with Array Word Area**	**Sign. Suffixes**	**OPC Data Type**

BIT VT_BOOL	X	DY 1	-----	D.X1.2	-----	-----	-----	-----
BIT [3] VT_BOOL	---- -	DY 1	-----	D.1.2	-----	-----	-----	-----
BYTE VT_UI2	B BY TE	---- -	Y.B1 0 Y.BY TE10	R.B1 R.BYTE 1	Y.B1 0.5 Y.BY TE10 .5	R.B1.3 R.BYTE 1.3	-----	-----
INT VT_I2	I IN T	Y.I 10 Y.I NT 10	Y.I10 Y.INT 10	R. I 2 R. INT 2	Y.I10 .3 Y.INT 10.3	R.I2.3 R.INT2.3	BCD D WDT* KT RI RU S5T SWAB BA	VT_I2 VT_BSTR VT_BSTR VT_BSTR VT_R4 VT_R4 VT_BSTR VT_I2 VT_BOOL|VT_ARRAY
WORD VT_UI2	W WO RD	R2 0	Y.W1 0 Y.W ORD 10	R.W2 R.WOR D2	Y.W 10.3 Y.W ORD 10.3	R.W2.3 R.WOR D2.3	BCD D WDT [2] KT RI RU S5T SWAB BA	VT_I2 VT_BSTR VT_BSTR VT_BSTR VT_R4 VT_R4 VT_BSTR VT_UI2 VT_BOOL|VT_ARRAY

© Softing Industrial Automation GmbH

DOUBLE WORD VT_UI4	D DW DWORD	-----	Y.D10 Y.DWORD10	R.D2 R.DWORD2	Y.D10.3 Y.DWORD10.3	R.D2.3 R.DWORD2.3	BCD SWAB KG IEEE TR RI RU T WDT [2] BA	VT_I4 VT_UI4 VT_R4 VT_R4 VT_R4 VT_R4 VT_R4 VT_BSTR VT_BSTR VT_BOOL|VT_ARRAY
DOUBLE INT VT_I4	DI DINT	-----	Y.DI10 Y.DINT10	R.DI2 R.DINT2	Y.DI10.3 Y.DINT10.3	R.DI2.3 R.DINT2.3	BCD SWAB KG IEEE TR RI RU T WDT [2] BA	VT_I4 VT_I4 VT_R4 VT_R4 VT_R4 VT_R4 VT_R4 VT_BSTR VT_BSTR VT_BOOL|VT_ARRAY
REAL VT_R4	R REAL	-----	Y.R10 Y.REAL10	R.R2 R.REAL2	Y.R10.3 Y.REAL10.3	R.R2.3 R.REAL2.3	-----	-----
STRING VT_BSTR	S STRING	-----	Y.S10.20 Y.STRING10.20	R.S2 R.STRING2.20	-----	-----	KA SWAB	VT_BSTR VT_BSTR

[2] For suffix WDT (Wago Date and Time) four registers (8 bytes) are read.

[3] Careful: With HEX addresses, it's better to select the version with the X.

[.Array size]

Arrays are created to combine several units of one data type into one field.

Examples:

D20.300

[Suffix]

Suffixes can be used to represent a value in another format.

Example: D20.300KF

The Send/Receive Item syntax is set up as follows.

<Job><Data type><Start address>[.Array size][Suffix]

If the data type is BOOL, this requires specification of the bit number.

<Job><Data type><Start address><.Bit number>[.Array size][Suffix]

Legend: <> Mandatory [ ] Optional

<Job>

	Syntax
SEND	**S**
RECEIVE	**R**

<Data type>

	Syntax	**Syntax Send**	**Syntax Receive**	**with Array**	**Useful Suffix**	**OPC Data Type**
BIT VT_BOOL	X	SX1.2	RX1.2	-----	-----	-----

© Softing Industrial Automation GmbH

BIT VT_BOOL	-----	S1.2	R1.2	-----	-----	-----
BYTE VT_UI1	B BYTE	SB4 SBYTE4	RB5 RBYTE5	SB4.4 SBYTE4.5 RB5.10 RBYTE5.3	BA BCD RI RU	VT_BOOL|VT_ARRAY VT_I2 VT_R4 VT_R4
CHAR VT_BSTR	C CHAR	SC4 SCHAR4	RC5 RCHAR5	SC4.2 SCHAR4.2 RC5.4 RCHAR5.4	DT BA	VT_BSTR VT_BOOL|VT_ARRAY
WORD VT_UI2	W WORD	SW10 SWORD10	RW10 RWORD10	SW10.2 SWORD10.2 RW10.4 RWORD10.4	BA BCD D KF KT RI RU S5T SWAB TR	VT_BOOL|VT_ARRAY VT_I2 VT_BSTR VT_I2 VT_BSTR VT_R4 VT_R4 VT_BSTR VT_UI2 VT_R4
INT	I	SI4	RI6	SI4.2	BA	VT_BOOL|VT_ARRAY

VT_I2	INT	SINT4	RINT6	SINT4.2 RI6.10 RINT6.10	BCD KT RI RU S5T SWAB TR	VT_I4 VT_BSTR VT_R4 VT_R4 VT_BSTR VT_I4 VT_R4
DOUBLE WORD VT_UI4	D DW DWORD	SD6 SDW6 SDWORD6	RD6 RDW6 RDWORD6	SD6.2 SDW6.2 SDWORD6.2 RD6.5 RDW6.5 RDWORD6.5	BA BCD KF KG RI RU SWAB T	VT_BOOL|VT_ARRAY VT_I4 VT_I4 VT_R4 VT_R4 VT_R4 VT_UI4 VT_BSTR
DOUBLE INT VT_I4	DI DINT	SDI6 SDINT6	RDI2 RDINT2	SDI6.2 SDINT6.2 RDI2.5 RDINT2.5	BA BCD KG RI RU SWAB T	VT_BOOL|VT_ARRAY VT_I4 VT_R4 VT_R4 VT_R4 VT_I4 VT_BSTR
REAL VT_R4	R REAL	SR2 SREAL2	RR2 RREAL2	SR2.2 SREAL2.2	----- 	-----

© Softing Industrial Automation GmbH

				RR2.5 RREAL2.5		
STRING VT_BSTR	S STRING	SS5.2 SSTRING5.2	RS5.2 RSTRING5.2	-----	KA DT	VT_BSTR VT_BSTR
S7-STRING VT_BSTR	G	SG5.2	RG5.2	-----	KA DT	VT_BSTR VT_BSTR

<Start address>

The start address specifies the byte address starting at which sending and receiving begins.

Example: SWORD10: Word 10 is the start address.

If the start address is a certain bit, specification of the bit number is also required.

<.Bit number>

When the data type is BOOL, the bit number must be specified!

Example: SX1.2: Bit 2 of byte 1 is the start address.

[.Array size]

Arrays are created to combine several units of one data type into a field.

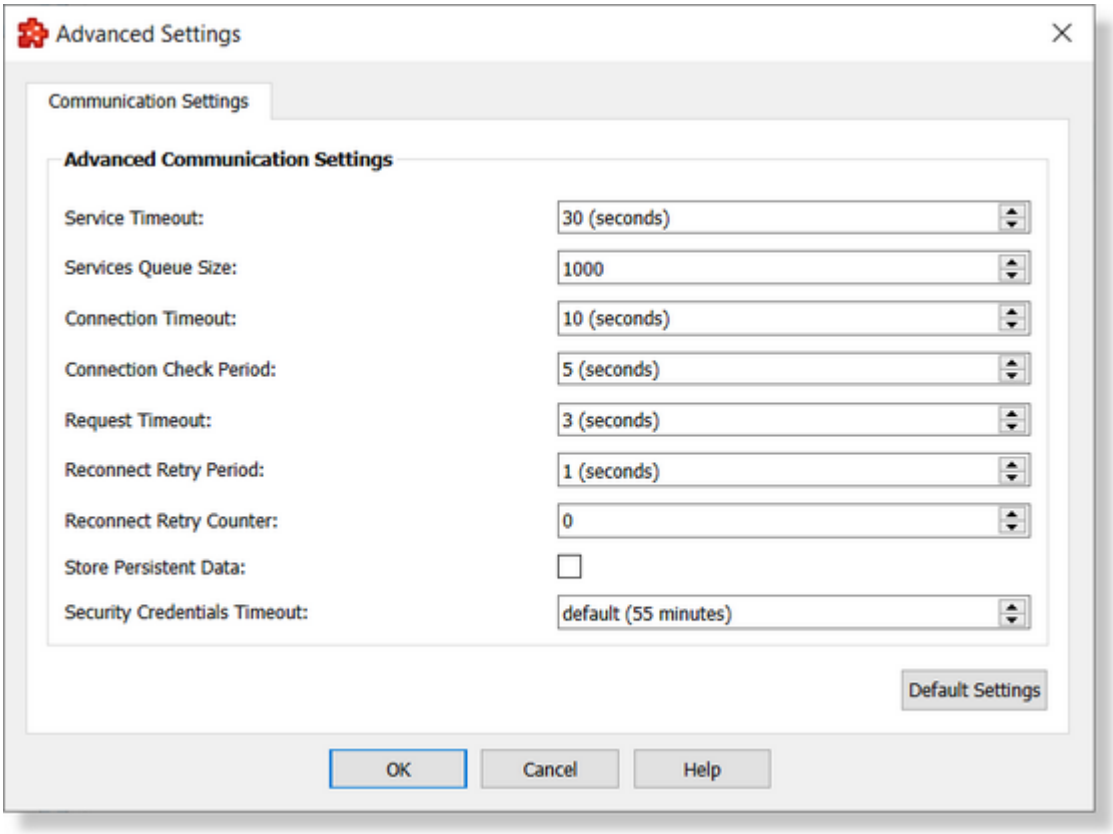
Example: RWORD10.4

[Suffix]

A suffix can be used to represent a value in another format.

7.3.1.15.6Siemens S7-2 Advanced Settings

Advanced settings dialog contain a set of parameters needed in order to create a connection to an Siemens S7-2 device. All parameters are connection specific.



Parameter Name	Description	Default value and value range (if applicable)
Service Timeout	Maximum time in seconds allowed for a service being completely executed. One service might be spitted into multiple protocol requests based on number of operations which should be processed in a single call - therefore it is advisable to have a larger value for this parameter.	Default value: 30 seconds (configurable at second range) Range: 1 to 600 seconds **Restriction**: The value should be at least equal (≥) to the double of Request Timeout value.
Services Queue Size	Maximum number of concurrent pending services for execution. Any additional service call made outside of the services queue is automatically rejected.	Default value: 1000 Range: 100 to 5000
Connection Timeout	Time in seconds the connection API waits for connection to be made. Also, it is used as watchdog operation for S7-2 connections. In case there is no traffic on the network (beside the	Default value: 10 seconds (configurable at second range). Range: 1 to 600 seconds

© Softing Industrial Automation GmbH

	keep alive mechanism) for an interval longer than the Connection Timeout value the connection will be automatically disconnect into the communication module. This should normally not happen as the application triggered keep alive mechanism is active.	**Restriction:** The value should be at least equal (≥) to the value of Connection Check Period multiplied with the maximum between 1 and Reconnect Retry Counter value.
Connection Check Period	Time in seconds used for monitoring the PLC connection and its availability on the network. The keep alive mechanism is handled sending a message for each Connection Check Period.	Default value: 5 seconds (configurable at seconds range). Range: 1 to 600 seconds **Restriction:** The value should be at least equal (≥) to the Request Timeout value. **Restriction:** The value should be at most equal (≤) to the Connection Timeout value divided with maximum between 1 and the Reconnect Retry Counter value.
Request Timeout	Time in seconds used by the protocol communication module to complete a request at the device layer. This is the maximum time allowed for the PLC to execute and confirm execution of a request. It should be adjustable based on network latency and PLC performance.	Default value: 3 seconds (configurable at second range). Range: 1 - 300 seconds **Restriction:** The value should be at most equal (≤) to the minimum between the Connection Check Period value and Service Timeout value divided by 2.
Reconnect Retry Period	Delay between consecutive reconnecting attempts to the PLC.	Default value: 1 second (configurable at second range). Range: 0 to 60 seconds
Reconnect Retry Counter	The number of connection attempts before to mark a connection as not accessible.	Default value: 0 (Referring to the default retries counter. In this case the communication module decides itself when

		and how many times a reconnection attempt is tried. Range: 0 to 60 **Restriction:** Thee value should be at most equal to the Connection Timeout value divided with the Connection Check Period value.
Store Persistent Data	By selecting this option data will be stored at the stopping time. Basically a snapshot of the application containing information about the current address space is stored at the configuration location and at the restart time this snapshot is loaded before the initialization of communication module. This will reduce the time needed for the first browse and validate operations.	
Security Connection Timeout	Siemens devices requires a periodically security token exchange. The connection being automatically disconnected of the security token exchange in not successfully executed. The actual firmware requires an update of the security token being hourly executed.	Default value is set to 55 minutes. **Attention:** using a value bigger then connection monitoring interval from the PLC can generate data loose between the moment when the PLC is cutting out the connection and the communication library is restoring it.

Note: All restrictions are automatically applied by the configurator. Values outside of these restrictions are not permitted.

© Softing Industrial Automation GmbH

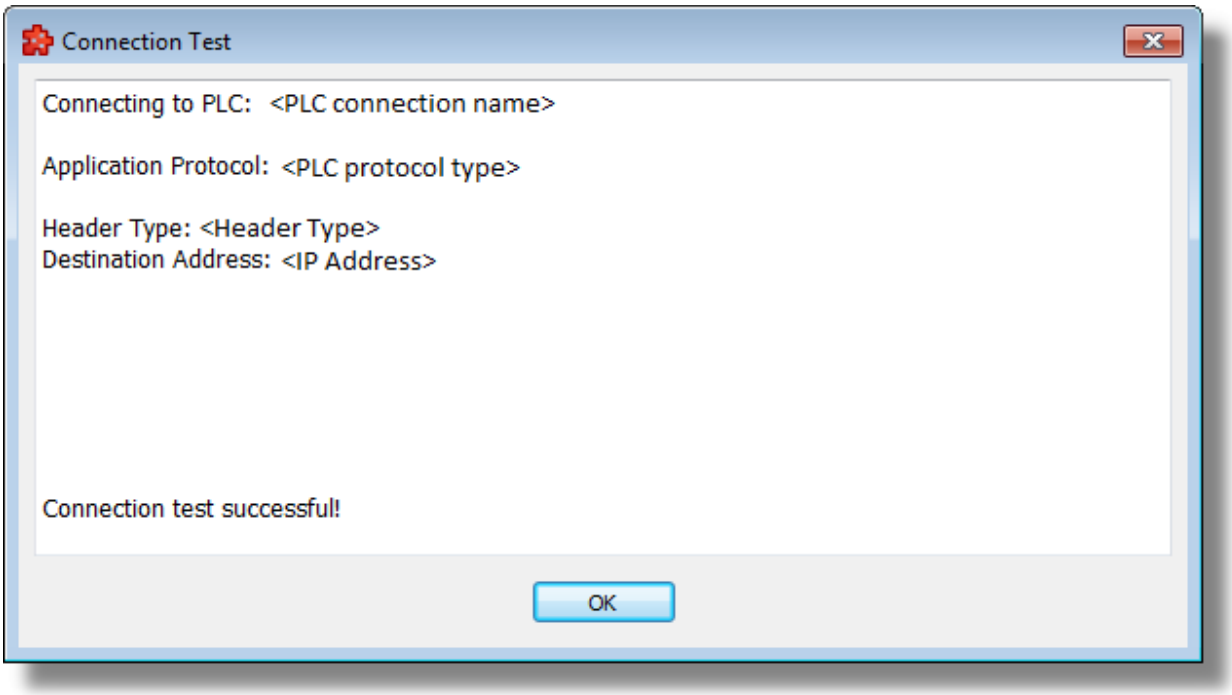
$$1000 <= \text{RequestTimeout} <= \text{ConnectionCheckPeriod} <= \frac{\text{ConnectionTimeout}}{\text{ReconnectRetryCounter}} \quad \text{(only if value > 0)}$$

$$1000 <= \text{RequestTimeout} <= \frac{\text{ServiceTimeout}}{2}$$

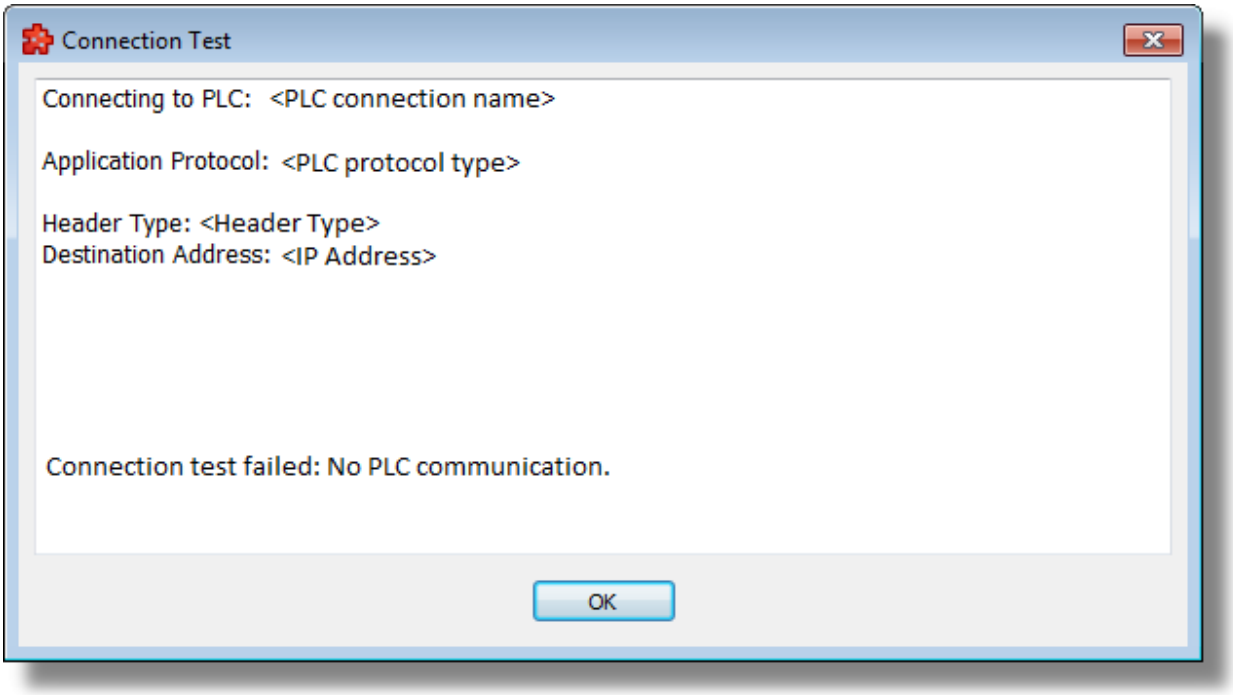
The default settings can be restored by using the **[Default Settings]** button.

7.3.1.15.7PLC Test Connection

The Test Connection button checks the PLC Device connection. The dataFEED OPC Suite Configurator tries to connect to the external PLC Device. A dialog is shown with the test results. If the connection was successful, the following dialog box is displayed:



If no connection could be established, the following dialog box is displayed:



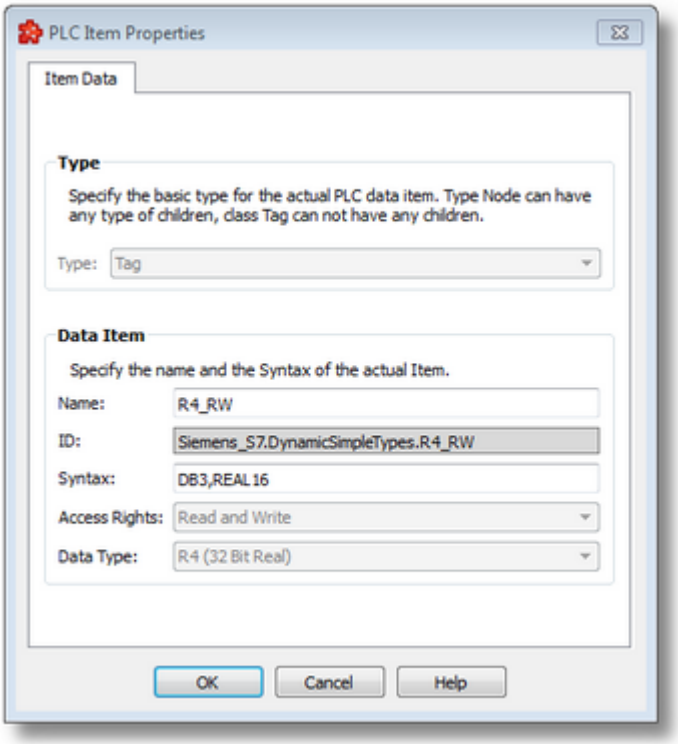
7.3.1.15.8Plc Connection Item Properties

The properties of the *Filter Item* can be set when adding Filter Items and can be modified when editing previously added Filter Items.

Add a New Plc Item

The add operation is triggered by selecting the parent node and then pressing the *Add a new item* button  , using the shortcut or selecting the *Add* action from the context menu. When adding a new Filter Item the following dialog is displayed:

© Softing Industrial Automation GmbH



Note: When you select items of type *Node*, the only configurable parameter is the *Name*.

Note: The *Access rights* and *Data type* parameters are read-only. The *Access Rights* reflects the access right which are set on the connection, while the *Data type* is automatically completed based on valid PLC data syntax which is provided by the user in the field *Syntax*.

The available configuration options in this dialog are described in the table below:

Attribute	Description
Type	The type of the data item. The possible types are: *Node* and *Tag*.
Name	The OPC item name identifying the corresponding data item inside the dataFEED OPC Suite namespace.
Id	The full OPC item path for the data item inside the dataFEED OPC Suite namespace.
Syntax	The PLC syntax for the data item. The PLC syntax is dependent of the protocol for which the current PLC Connection is configured. For more information please click here.

Access rights	The access rights for the data item. The possible values are: *Read*, *Write*, *Read and Write*.
Data type	The data type of the data item. The data type for the item is extracted out of a valid syntax provided by the user.

Note: Items of type *Node* may have child items of any existing type.

Note: Items of type *Tag* cannot have any children. Therefore, *Add* is not available for tags.

The OPC filter data items of the namespace may be of any data type covered by the OPC Specifications. The following table describes the available data types mapped to the corresponding VARIANT types from the OPC Specification.

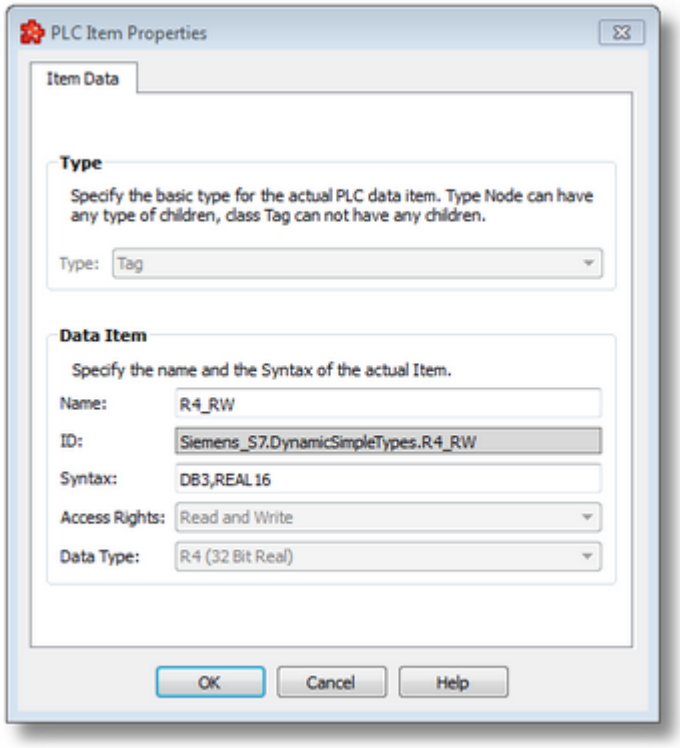
Data type	VARIANT type	Description
I1	*VT_I1*	8 bit signed integer.
UI1	*VT_UI1*	8 bit unsigned integer.
I2	*VT_I2*	16 bit signed integer.
UI2	*VT_UI2*	16 bit unsigned integer.
I4	*VT_I4*	32 bit signed integer.
UI4	*VT_UI4*	32 bit unsigned integer.
I8	*VT_I8*	64 bit signed integer.
UI8	*VT_UI8*	64 bit unsigned integer.
R4	*VT_R4*	32 bit floating point.
R8	*VT_R8*	64 bit floating point.
BOOL	*VT_BOOL*	Boolean.
STRING	*VT_BSTR*	String.
DATETIME	*VT_DATE*	Date and time.
DECIMAL	*VT_CY*	Decimal.
ANY	*VT_VARIANT*	Any data type.

Note: Arrays of fixed and variable length are also supported for these data types.

© Softing Industrial Automation GmbH

Modify a Plc Item

The modify operation is triggered by selecting the node to be modified and then pressing the *Properties* button  , using the shortcut or selecting the *Properties* action from the context menu. When modifying an existing Filter Item the following dialog is displayed:



The configuration options are the same as in the above described *Add a New Plc Item* dialog.

Note: The *Type* of the data item cannot be changed anymore.

Note: The *Modify* action is also triggered by double-clicking on tree items without children.

7.3.1.15.9TSAP

TSAP = Transport Service Access Point

ISO (H1) uses so-called TSAPs in the transport layer to address the applications. These connection points are required for both ISO (H1) connections and RFC1006 connections.

Parameter for RFC1006 Connections	
Station A	Station B

Local TSAP A	Local TSAP B
Remote TSAP B	Remote TSAP A
IP address A	IP address B

Before a connection can be established, the TSAPs must correspond crosswise (i.e., the “local TSAP” of communication partner A must correspond to the “remote TSAP” of communication partner B). In contrast, the “remote TSAP” of station A must correspond to the “local TSAP” of station B.

Local TSAP (station A) = remote TSAP (station B)

Remote TSAP (station A) = local TSAP (station B)

This requirement is easy to ad here to if you select identical values for “local TSAP” and “remote TSAP.” Local TSAP and remote TSAP may be identical.
If several connections are set up between 2 stations, the TSAPs of the individual connections must be different.

The combination of TSAP + IP address must be unambiguous so that a connection can be precisely specified.

Rules for entering the TSAPs for S7 connections

- TSAPs have a minimum length of 2 bytes and a maximum length of 8 bytes.
- TSAPs can be entered either as hex or as ASCII characters.
- The TSAPs for S7 connections have a special meaning. See TSAP for S7 connection. See TSAP for S7 connection.

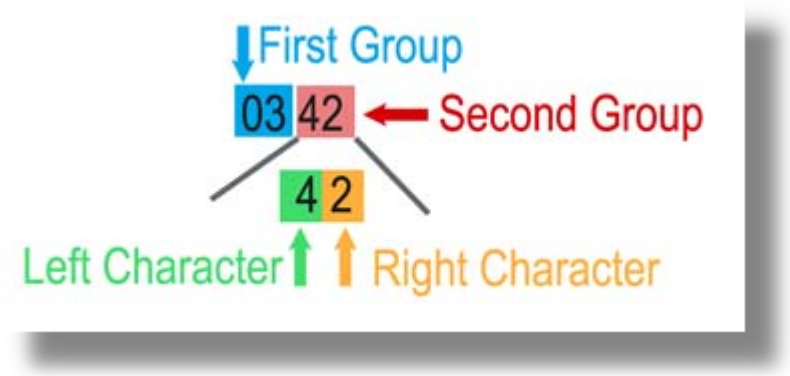
Rules for the entry of the TSAPs for S5 connections

- TSAPs have a minimum length of 2 bytes and a maximum length of 8 bytes for TCP/IP.
- TSAPs can be entered either as hex or as ASCII characters.
- The TSAPs for S5 connections have no special meaning. They are selected at random.

7.3.1.15. TSAP for S710

So-called standard TSAPs are used for connections that cannot be parameterized (i.e., standard connections). The following rules apply to these.

© Softing Industrial Automation GmbH



First group

Contains device identifiers for those that are provided in S7 resources.

01: PG or PC

02: OS (operator or monitoring device)

03: Other (e.g., OPC server, Simatic S7 PLC)

Second group

Contains the addresses of these components

Left character (bit 7...4):

Rack number multiplied by 2

Right character (bit 3...0):

Slot (< 16) of the CPU. Slot 2 is always used for the S7-300.

The standard TSAPs **MUST** be used on the controller side (remote TSAP of the OPC server station).

Although the own TSAP (local TSAP of the OPC server station) can be selected as desired, its should have the same format. We recommend using TSAP 01 01 as the own TSAP.

Examples:

01 01 PC running on the server; direct addressing

03 43 OPC communication with the S7 CPU on rack 2, slot 3

03 02 OPC communication with the S7 CPU on rack 0, slot 2

03 2E OPC communication with the S7-CPU on rack 1, slot 14

© Softing Industrial Automation GmbH

Note: The binary standard TSAPs are entered in the HEX window (left).

S7-300

The S7-300 is usually addressed with the TSAP 03 02 (binary).

S7-400

The S7-400 is usually addressed with the TSAP 03 03 (binary).

S7-1200

The S7-1200 is usually addressed with the TSAP 03 01 (binary).

S7-1500

The S7-1500 is usually addressed with the TSAP 03 01 (binary).

7.3.1.15. IP Address
11

The basics

In order to establish communication between two technical devices, each of the devices must be able to send data to the other device. To ensure that these data arrive at the right place on the other station, this station must be named (i.e., addressed) unambiguously. In IP networks this is done with an IP address.

An OPC server can address a PLC directly via its IP address (e.g., 192.168.1.20). A PLC can also be addressed via its name. This means that the name of the PLC must be stored with the applicable IP address and the domain server must be entered in the TCP/IP of the server. The server asks for a domain name (e.g., “PLC1”, the IP address for a name server) and then addresses the PLC directly via its IP address “192.168.1.20.”

IP address

Every IP data package begins with an information area for transportation through the IP layer – the IP header. This header also contains two fields in which the IP addresses of both the sender and the receiver are entered before the data package is sent. Relaying occurs in layer 3 of the OSI model – the relaying layer.

Setup

IPv4

The IPv4 addresses predominantly used since the introduction of version 4 of the Internet protocol consist of 32 bits (i.e., 4 octets – bytes). This means that 232 (i.e., 4,294,967,296) addresses can be represented. In dotted decimal notation, the 4 octets are written in decimal presentation as four whole numbers separated from each other by dots from 0 to 255.

Example: 192.168.1.20.

IPv6 – new version with larger address area

© Softing Industrial Automation GmbH

Due to the quickly increasing need for IP addresses, the day is coming when the usable address area of IPv4 will be exhausted.

The IANA address pool was used up on February 3, 2011. IPv6 was primarily developed for that reason. It uses 128 bits to store addresses which means that 2128 = 25616

(= 340.282.366.920.938.463.463.374.607.431.768.211.456 ≈ 3,4 · 1038) addresses can be represented. This number is sufficient to provide at least 665.570.793.348.866.944

(= 6,65 · 1017) IP addresses for every square millimeter of the earth's surface.

Since the decimal representation ddd.ddd.ddd.ddd.ddd.ddd.ddd.ddd.ddd.ddd.ddd.ddd.ddd.ddd.ddd.ddd would be confusing and difficult to handle, we usually represent IPv6 addresses as hexadecimal numbers. To further simplify this representation, two octets each of the address are combined and presented in groups separated by colons. XXXX:XXXX:XXXX:XXXX: XXXX:XXXX:XXXX:XXXX.

Example: 2001:0db8:85a3:0000:0000:8a2e:0370:7344

To further shorten the presentation, zeros can be left off at the beginning of a block.

A sequence of blocks which only consist of zeros is replaced by :: but only once per IPV6 address.

Example: 2001:db8:85a3::8a2e:370:7344

When IPV6 is used, usually names are specified instead of the IP addresses.

7.3.1.15. Suffixes

12

Suffixes can be used to represent a value in another format.

Suffix	Synt	Used for	Area	Data type	Variant Dat	Comments
BitArra	BA		No. of bits:			With the BA suffix, the data sotrage saved on the PLC are shown as an array of bits.
		Byte,	Size in bytes t	BOOLEAN	VT_BOOL	
		Word,	Size in words		VT_BOOL	
		Int,	Size in bytes t	BOOLEAN	VT_BOOL	
		DWord,	Size in bytes t	BOOLEAN	VT_BOOL	
		DInt,	Size in bytes t		VT_BOOL	
		DDWord,	Size in bytes t	BOOLEAN	VT_BOOL	
		DDInt	Size in bytes t	BOOLEAN	VT_BOOL	

					BOOLEAN BOOLEAN		
BCD	BCD	Byte, Word, Int, DWord, DInt, DDWord, DDInt	Byte: Word Int: DWo DInt: DDW DDIn	0 to 99 0 to 999 0 to 999 0 to 999 0 to 999 0 to 999 0 to 999	SHORT SHORT	VT_I1 VT_I2 VT_I2 VT_I4 VT_I4 VT_I8 VT_I8	With the BCD suffix, the data saved on the PLC are shown as non-signed, binary-coded values. For example, the decimal value "65535" is shown as "9999".
Date	D	Word	1990-01-01 to 2168-12-31		STRING	VT_DATE	The suffix D is used to show the data saved on the PLC as data type DATE. The DATE data type occupies one word. The content corresponds to the number of days since 01.01.1990. The representation contains the year, the day and the month, separated by a hyphen.September 1, 2006 is shown as 2006-01-09. The value range is from 0 (0 days since 01.01.1990: 1990-01-01) to 65378 (65378 days since 01.01.1990: 2168-12-31). Rules for use of suffix D: · The years 1990 up to and including 2089 can be specified with 2 or 4 positions.The years starting with 2090 must be entered with 4 positions. 90 to 99 ==> 1990 to 1999 · 00 to 89 ==> 2000 to 2089 · The months and days can be specified with either 1 or 2 positions.

© Softing Industrial Automation GmbH

						· Anything but numbers can be used as a separator (e.g., 89/09/17). Any number of separators can be used.
Date a	DT	Word, Int	1990-1-1-00:00:00.000 to 2089-12-31-23:59:59.999	STRING	VT_DATE	The DT suffix is used to show the data saved on the PLC as combined data type DATE_AND_TIME. The data type DATE_AND_TIME has 8 bytes (64 bits). The year, the month, the day, the hour, the minutes, the seconds and the milliseconds are included. Remember the separators (hyphen, colon and period)!
Date a	DU	Word, Int	1990-01-01 00:00:00.000 to 2089-12-31 23:59:59.999	STRING	VT_DATE	The DU suffix is used to show the data saved on the PLC as combined data type DATE_AND_TIME in SQL-compatible format. There is no hyphen in front of the time.
ASCII	KA	String, S7-String	HEX: 0 to 9, A to F	STRING	VT_BSTR	The KA suffix is used to show the data saved on the PLC in hex characters.
Signe	KF	Byte (LByte, RByte, S5), Word (S5), DWord (S5)	Byte: -128 to 127 Word: -32768 to 32767 DWord: -2147483648 to 2147483647	CHAR SHORT LONG	VT_I2 VT_I2 VT_I4	The KF suffix is used to show the data saved on the PLC as a signed, fixed-point number.
S5-KC	KG	DWord, DInt	0,1469368E-38 to 0,1701412E39 Caution: The KG area in the PLC is larger than on the PC!	REAL	VT_R4	The KG suffix is used to handle the data saved on the PLC as a 4-byte, floating point number.

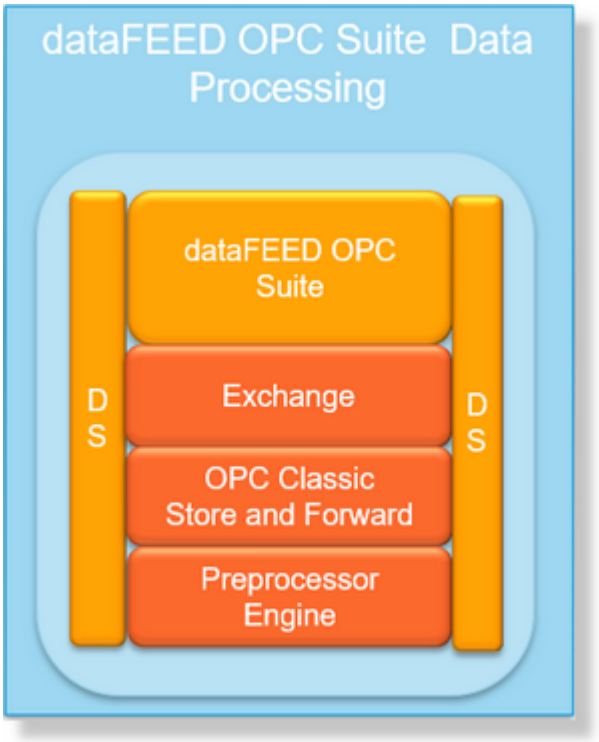
S5-KT	KT	Word, Int	000.0 to 999.3	STRING	VT_BSTR	The KT suffix is used to show the data saved on the PLC as a 2-byte time constant. The time basis is supplied in addition to the time value. The value range is from 000.0 to 999.3.
Signe	RI	Byte, Word, Int, DWord, DInt DDWord, DDInt	Converts item into a real value and rounds off the value during write-access.	REAL	VT_R4 VT_R8	The RI suffix is used to show the data saved on the PLC as a signed, REAL data type. When writing to the PLC, the value is rounded up. Example: db9dw0RI – the value 10.7 is rounded up to 11.
Unsig	RU	Byte, Word, Int, DWord, DInt DDWord, DDInt	Converts an item into a real value and rounds off the value during write-access	REAL	VT_R4 VT_R8	The RU suffix is used to show the data saved on the PLC as a non-signed, REAL data type. The value is rounded when written to the PLC.
S5 Tir	S5T	Word, Int	0ms to 2h46m30	STRING	VT_BSTR	The S5T suffix is used to show the data saved on the PLC as the Simatic time S5TIME. The data type S5TIME occupies a 16-bit word and is the product of time value and time frame. The time duration is specified in hours, minutes, seconds and milliseconds. The internal representation is a BCD number. The value range is from 0 ms to 2h46m30s. The smallest value is 10ms.
Swab	SWA	Word, Int, DWord, DInt, DDWord, DDInt	Switches high byte and low byte		VT_I2, UI2, VT_I4, VT_I4, VT_I4, VT_UI8, VT_I8	The SWAB suffix is used to switch high byte and low byte of the date stored on the PLC.
Time	T	DWord	-24D_20H_31M_23S_648MS to	STRING	VT_BSTR	The T suffix is used to show the data saved on the PLC as the data type TIME. The data type TIME occupies one double word (32

© Softing Industrial Automation GmbH

			24D_20H_31M_23S_647MS			bits). The representation contains the information for days (d), hours (h), minutes (m), seconds (s) and milliseconds (ms). Milliseconds can be omitted. The value range goes from -2147483648 (-24d20h31m23s648ms) to 2147483647 (24d20h31m23s647ms).
Time	TOD	DWord, DInt	0:0:0.0 to 23:59:59.999		VT_BSTR	The TOD suffix is used to show the data saved on the PLC as data type TIME_OF_DAY. The data type TIME_OF_DAY occupies one double word (32 bits). The representation contains the information for hours:minutes:seconds and .milliseconds. Milliseconds can be omitted. The value range goes from 0:0:0.0 to 23:59:59.999.
TimeR	TR	Word, Int	0.01 to 9990.0	REAL	VT_R4	The TR suffix is used to show the data saved on the PLC as the data type TIME REAL. The value range goes from 0.01 to 9990.0 (s).

7.3.2 Data Processing

This section of the Softing dataFEED OPC Suite Online Help describes the different dataFEED OPC Suite data processing modules.



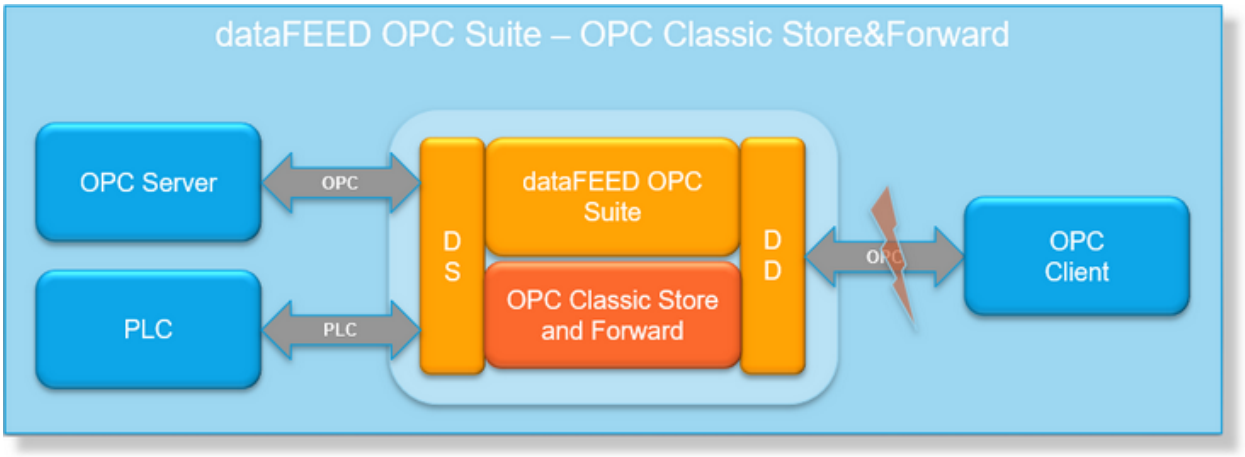
Softing dataFEED OPC Suite - Data Processing

- Protection against data loss @ Configuration: Data Processing - Classic OPC Store And Forward
- Exchange data between OPC servers @ Configuration: Data Processing - Exchange
- Processing data values on available data sources @ Configuration: Data Processing - Preprocessing Engine

Softing dataFEED OPC Suite V5.38
© 2024 Softing Industrial Automation GmbH

7.3.2.1 OPC Classic Store And Forward

This section of the Softing dataFEED OPC Suite Online Help describes how to configure OPC Classic Store And Forward within Softing dataFEED OPC Suite.



The OPC Classic Store And Forward functionality allows the data to be stored while a connection is interrupted and to be transferred when the connection resumes. The storage side of the OPC Classic Store And Forward functionality can be configured in the OPC Classic Store And Forward plug-in.

The OPC Classic Store And Forward needs a server which monitors the connection to a certain client and when this connection is offline it starts storing data with the defined settings. When the connection is back on the server starts forwarding the data to the client. The corresponding connection monitoring for Store And Forward can only be used for one special OPC or Tunnel client. This special monitoring client is, depending on the configuration, either identified by its name or always the first connected client.

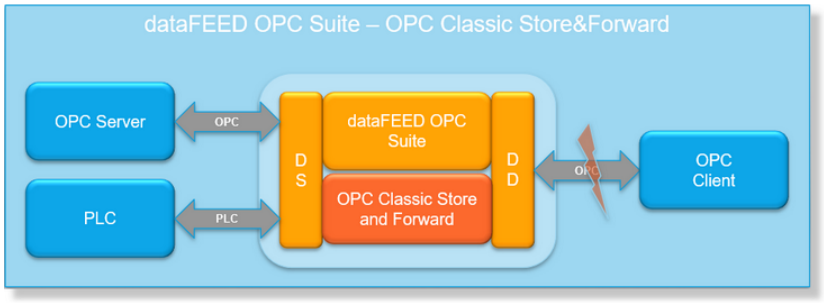
The special forward client can only access the data items explicitly defined as monitoring items on the server side. In the dataFEED OPC Suite Configurator all the data items from the server are always visible, however only the actual monitoring items can be used in the local dataFEED OPC Suite application during run time.

Such a special forward client can be configured from the dataFEED OPC Suite Configurator by activating the *Use as forward Client* option for the desired server connection. This option is found in the OPC Server Wizard - OPC Classic Store And Forward page.

The local dataFEED OPC Suite application ensures that no data is lost while the connection between the server and client is interrupted. Through the Store functionality dataFEED OPC Suite stores the values of included items, and through the Forward functionality it ensures that the stored data will be available for the client after the connection has been re-established.

Specific details about creating and handling the Store And Forward definitions can be found on the Configuration: Data Processing - OPC Classic Store And Forward page.

7.3.2.1.1 Configuration



To start configuring the OPC Classic Store And Forward functionality select *OPC Classic Store And Forward* from the *Data Processing* group in the navigator on the left.

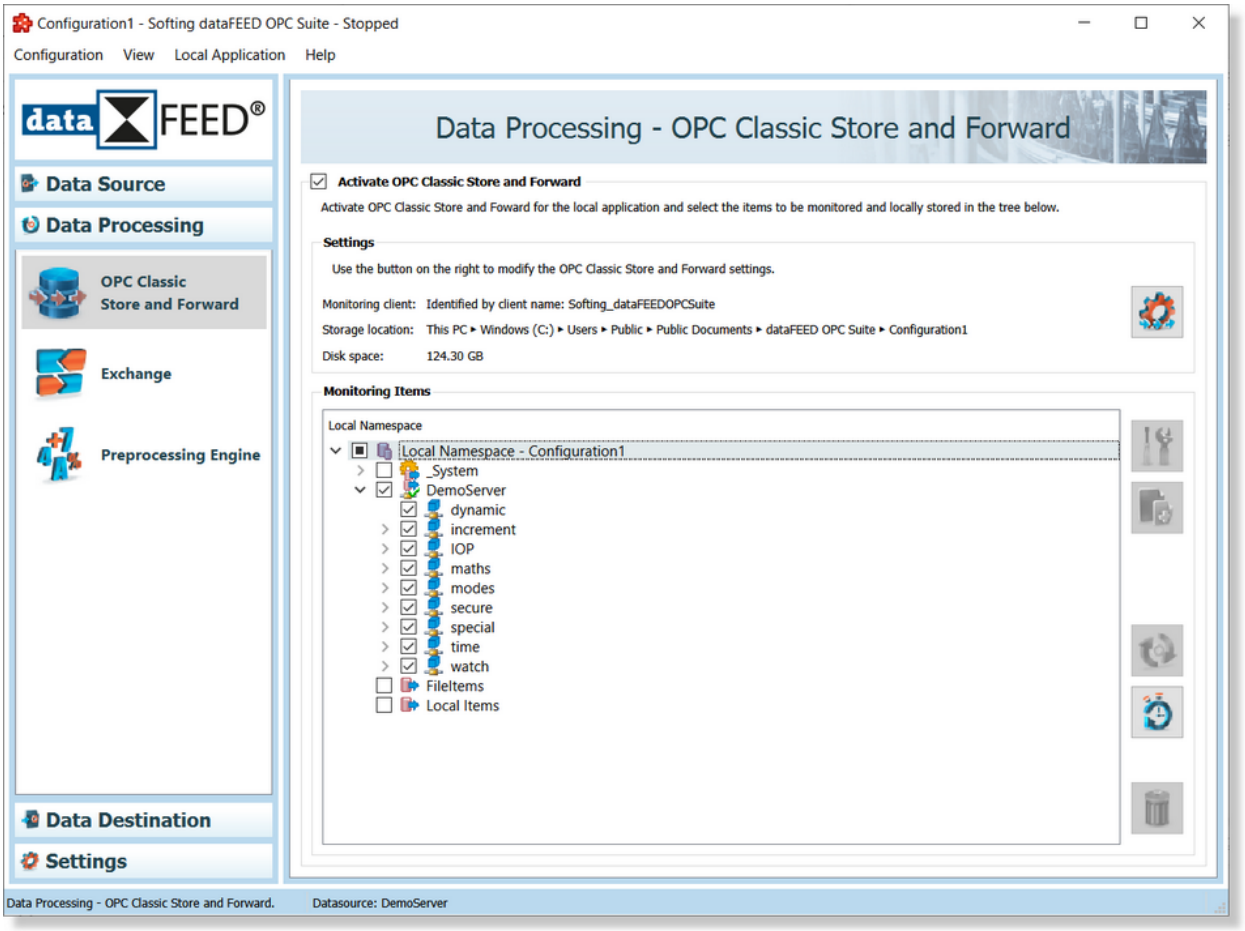
The OPC Classic Store And Forward main user interface will show up. The main elements of this page are the Store And Forward *Settings* and the *Monitoring Items*.

The checked items from the tree shown under *Monitoring Items* are the items selected for Store And Forward. When the connection to the name-identified client is lost – or when the last client disconnects if the name-identified client option is not set – only the selected items will be cached locally until the maximum number of data changes is reached.

Note: The *Monitoring Items* tree list is only visible if there is a configuration loaded.

Note: The *Monitoring Items* tree list is only editable if Store And Forward is activated.

© Softing Industrial Automation GmbH



The Store And Forward main user interface is divided into the following areas:

- The *Activate OPC Classic Store And Forward* checkbox
- The *Settings* group box
- The *Monitoring Items* group box

The Activate OPC Classic Store And Forward checkbox

OPC Classic Store And Forward can be activated by enabling the checkbox at the top of the main user interface. By default the Store And Forward add-on is disabled and because it is a costly add-on it should only be enabled when its functionality is required.

If OPC Classic Store And Forward is deactivated the OPC Classic Store And Forward add-on is not loaded by the local dataFEED OPC Suite application at run time.

When OPC Classic Store And Forward is deactivated its configuration data is not deleted but the add-on is only disabled. If activated again the existing data becomes available.

The Settings group box

The OPC Classic Store And Forward functionality is configured in the OPC Classic Store And Forward Settings Wizard.

In this wizard the following data can be configured:

- The local storage path and file name, the file size at which a warning is issued, the number of maximum data changes to be stored in the local data storage and whether to overwrite the data changes or not (when the maximum data changes has been reached).
- The store and forward monitoring client which can be identified as the first connecting client or a specific client identified by its name.
- The forward settings which include the forward method and the forward cycle time.
- The monitored storage items.

Some of the wizard values have defaults but it is highly advisable to verify them before proceeding to include items for monitoring in Store And Forward.

This wizard is opened when the *Activate OPC Classic Store And Forward* checkbox is selected or by using the *Modify OPC Classic Store And Forward Settings* button  .

The Monitoring Items group box

The *Monitoring Items* group box contains the entire dataFEED OPC Suite namespace tree. Each item in this tree is preceded by a checkbox. Only the items for which the corresponding checkbox is checked are monitored in Store And Forward during the local dataFEED OPC Suite application run time.

Operation

From the Store And Forward main user interface certain operations can be performed on the selected checked/unchecked items using the buttons on the right, accessing the context sensitive menu or pressing the keyboard shortcuts. The following actions are available:

- Include the selected item and all of its children in the Store And Forward monitoring by pressing the SPACE key on the keyboard (when the corresponding item checkbox is not checked) or by selecting *Include* from the context menu.
- Exclude the selected item and all of its children from the Store And Forward monitoring by pressing the SPACE key on the keyboard (when the corresponding item checkbox is checked) or by selecting *Exclude* from the context menu.
- Re-establish the server connection and refresh the data items using the *Refresh data items* button  , pressing the keyboard shortcut CTRL+R or selecting the *Connect and Refresh Data* action from the context menu.
- Access the Store And Forward Settings to modify tag update rate and execution condition by selecting the *Settings* action from the context menu.

© Softing Industrial Automation GmbH

- Add a new item under the selected node by using the *Add* button  , pressing the CTRL+A keyboard shortcut or selecting the *Add* action from the context menu. Depending on what node is selected a special Add Item dialog opens:
    - If the selected node is under Local Items the Add Local Item dialog is opened.
    - If the selected node is under a dynamic server connection the Add Dynamic Server Item dialog is opened.
    - If the selected node is under a static server connection the Add Static Server Item dialog is opened.
- Delete the selected Store And Forward item by using the *Delete* button  , pressing the DEL / DELETE key or selecting the *Delete* action from the context menu.
- Copy and paste the selected Store And Forward item and all of its children by selecting the *Copy* and *Paste* commands from the context menu. These operations are only allowed between items of the same data source type.
- Import Store And Forward items from an external file through the Import Wizard by selecting the *Import* action from the context menu. The imported items should exist in the configuration to which they are imported, or the data source should allow the creation of dynamic items.
- Export Store And Forward items to an external file through the Export Wizard by selecting the *Export* action from the context menu.
- Open the item properties dialog by using the *Properties* button  , pressing the keyboard shortcut CTRL+P or selecting the *Properties* action from the context menu. Depending on what item is selected a special Item Properties dialog opens:
    - If the selected item is under Local Items the Modify Local Item dialog is opened.
    - If the selected item is under a dynamic server connection the Modify Dynamic Server Item dialog is opened.
    - If the selected item is under a static server connection the Modify Static Server Item dialog is opened.
- Open the Global Alias Values dialog to configure global alias and initial values by using the *Configure global alias and initial values* button  or pressing the keyboard shortcut CTRL+D. Each Store And Forward monitored item is included with a default update rate and an execution condition. If desired the default value and the alias value can be changed through this dialog.
- Delete the selected item and all of its children by clicking the *Delete* button  , pressing the DEL / DELETE key or selecting the *Delete* action from the context menu.

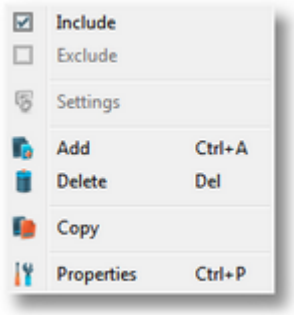
Note: The available context menu options depend on the selection in the tree. Multiple item selection is disabled in trees.

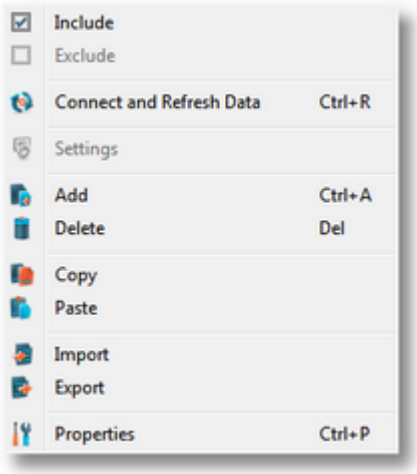
Note: The *Settings* context menu action is only available for items of type Tag.

Note: The *Add* action is only valid for items of type Node (including root node items).

Note: The *Import* action is only allowed for items of type Node (including root node items).

The following examples show the context menus for items of the Tag and Node types under a server connection:





Note: The tool tips on the tree also display item attributes.

Status Bar

The status bar contains the properties of the selected tree item:

- The item class (type).
- The item ID which actually is its hierarchical name in the dataFEED OPC Suite tree.
- If the selected item is of type Tag or Property then its monitoring status is also displayed. If the item is monitored for Store And Forward then its update rate and its execution condition

© Softing Industrial Automation GmbH

are displayed as well. If the item is not monitored then a message confirming this will be shown (e.g. *Not Monitored*).

The Store And Forward settings and items are stored in a separate file with a specific format which can be seen in the File Format page.

See the OPC Classic Store And Forward Wizard page for details about how to generally configure the Store And Forward add-on.

See the OPC Classic Store And Forward Items Settings page for details about how to configure specific Store And Forward items.

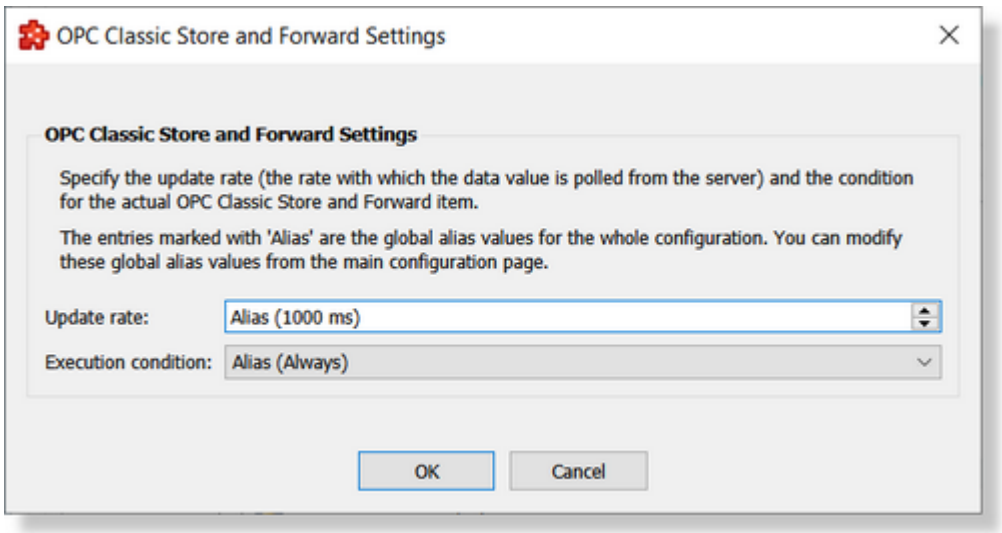
See the Store And Forward Import Wizard and Export Wizard for details about how to use the import and the export functionality of this add-on.

7.3.2.1.2 Store and Forward Item Settings

The *OPC Classic Store And Forward Settings* dialog allows changing the item's properties that are specific to Store And Forward.

This dialog can be accessed by selecting the *Settings* action from the context menu. The *Settings* context menu action is only available for items monitored for OPC Classic Store And Forward.

A dialog box is displayed:



In this dialog the specific Store And Forward item properties can be changed. These properties only apply when the Store And Forward add-on begins storing the data changes, that is when the last client disconnects (if client identification by name is not set) or (if client identification by name is set) when the specific client disconnects.

Attribute	Description

Update rate	The interval between the data polling for the specific item. This may be the *Global Alias Value* or a specific interval value in milliseconds.
Execution condition	The condition which decides whether the polled item value is stored or not. This value may be the *Global Alias Value* or one of the following: *AllGoodQuality*, *Always*, *ItemQualityChanged* or *OneBadQuality*.

By default the values of these attributes are the ones set in the Global Alias Values dialog. The Alias values represent global values for the entire application. When you change the Global Alias Values, items with *Alias* settings are automatically updated.

7.3.2.1.3 Store and Forward Wizard

This wizard allows the server side and storage side configuration of the local dataFEED OPC Suite application's OPC Classic Store And Forward functionality.

The wizard starts automatically when OPC Classic Store And Forward is activated. It can also be started using the *Modify the OPC Classic Store And Forward Settings* button  . The first time this wizard is displayed it adopts some default values.

This wizard allows the following:

- Configuration of the local storage properties.
- Configuration of the forward settings.
- Definition of the data items to be monitored and locally stored.

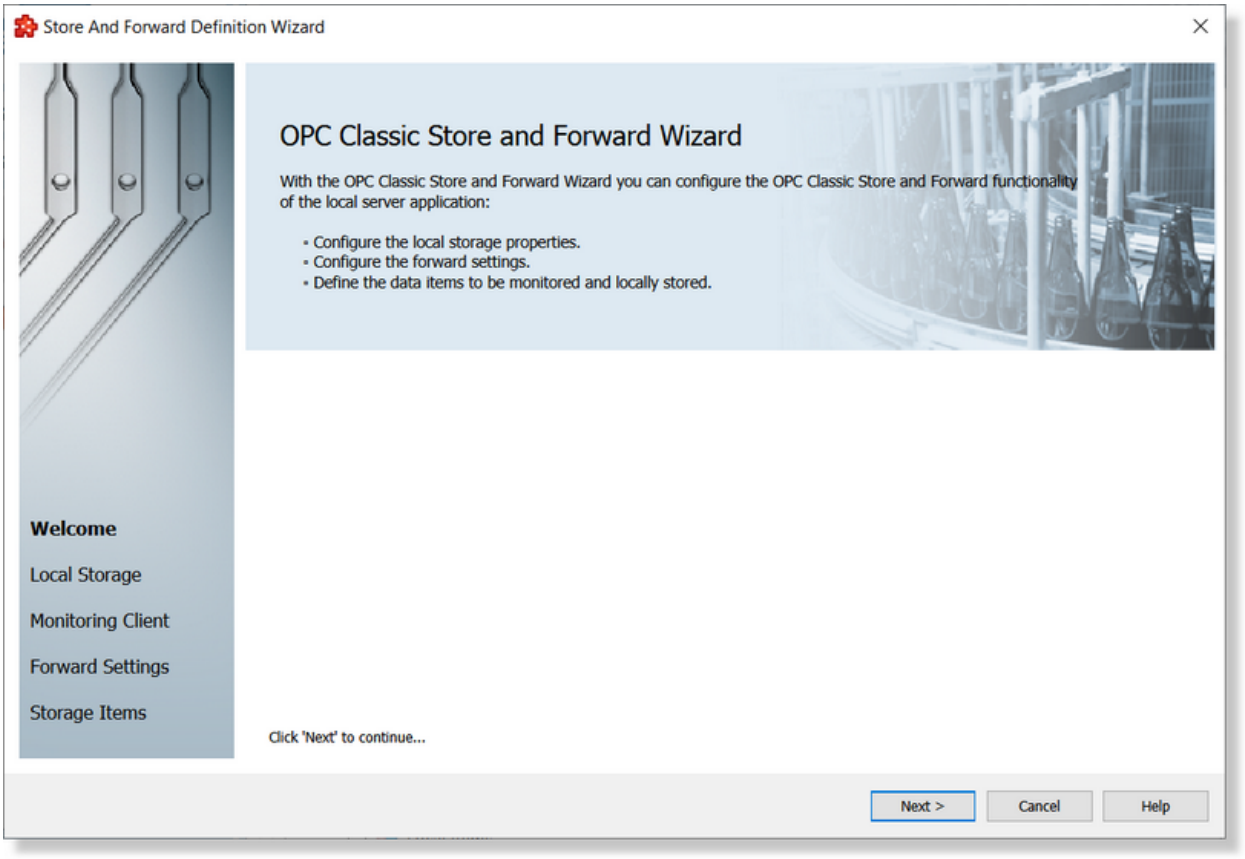
▾ With the OPC Classic Store And Forward Wizard you can configure the server respectively the storage side of the OPC Classic Store And Forward functionality of the Local Application:

- Obtain general information about the Store And Forward Wizard @ Store And Forward Wizard - Welcome
- Configure the local data storage @ Store And Forward Wizard - Local Storage
- Specify the external monitoring client identification @ Store And Forward Wizard - Monitoring Client
- Configure the data forwarding options @ Store And Forward Wizard - Forward Settings
- Define the storage items to be monitored and locally stored @ Store And Forward Wizard - Storage Items

© Softing Industrial Automation GmbH

This page provides a short overview of the OPC Classic Store And Forward Settings that can be configured on the following pages. This page requests no input as it only shows what the wizard allows:

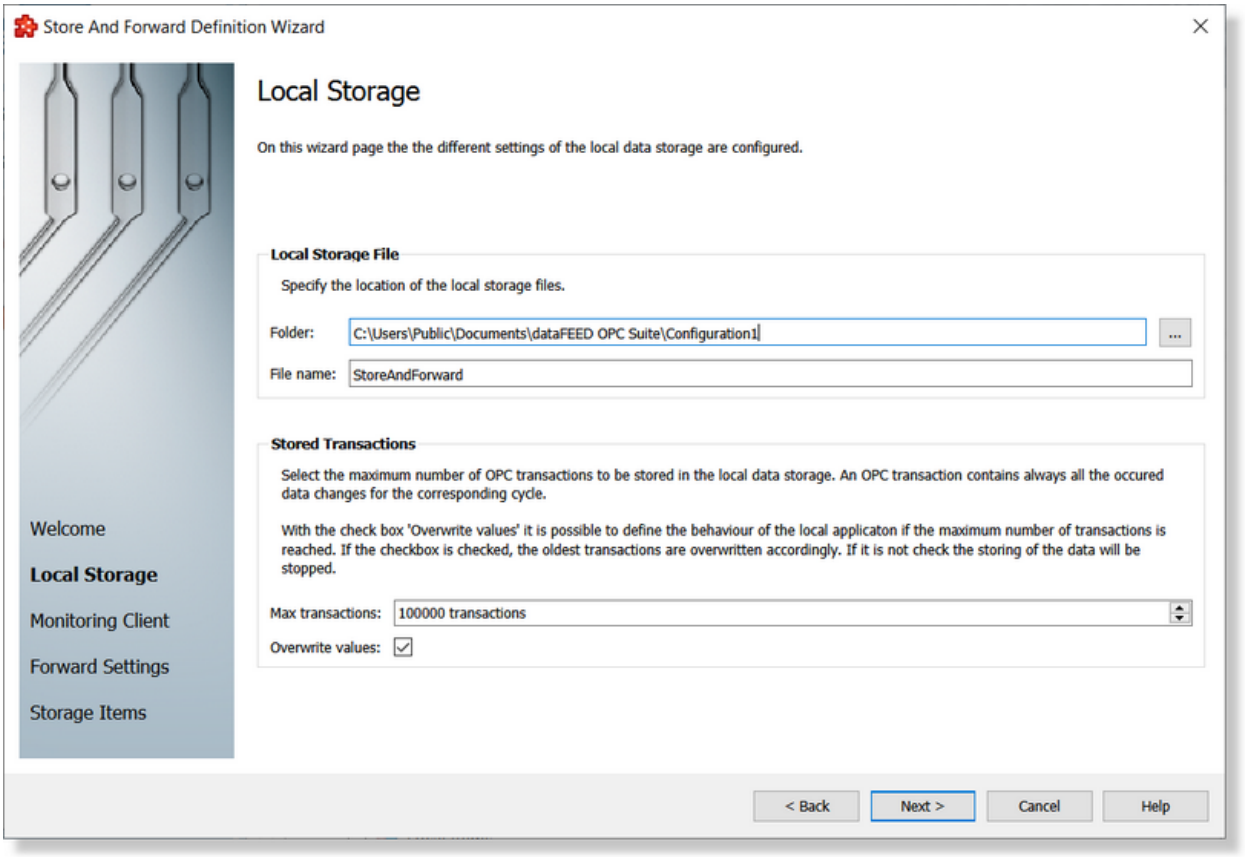
- Configuring the local storage properties.
- Configuring the forward settings.
- Defining the data items to be monitored and locally stored.



Press the *Next* button to proceed with the wizard or abort the operation with the *Cancel* button.

Store And Forward Wizard - Local Storage >>>

On the *Local Storage* wizard page the local storage settings can be configured: the *Local Storage File* settings and the *Stored Transactions* settings.



Local Storage File

In the *Local Storage File* group box the following settings can be configured:

- The path where the storage file should be located. If the path does not point to an existing folder then the corresponding path will be created either by the dataFEED OPC Suite Configurator when the *Next* button is pressed, or at run time by the local dataFEED OPC Suite application.
- The file name identifying the storage file. The file is created at the run time of the local dataFEED OPC Suite application. If the file exists it will be overwritten.

Stored Transactions

In the *Stored Transactions* group box the following settings can be configured:

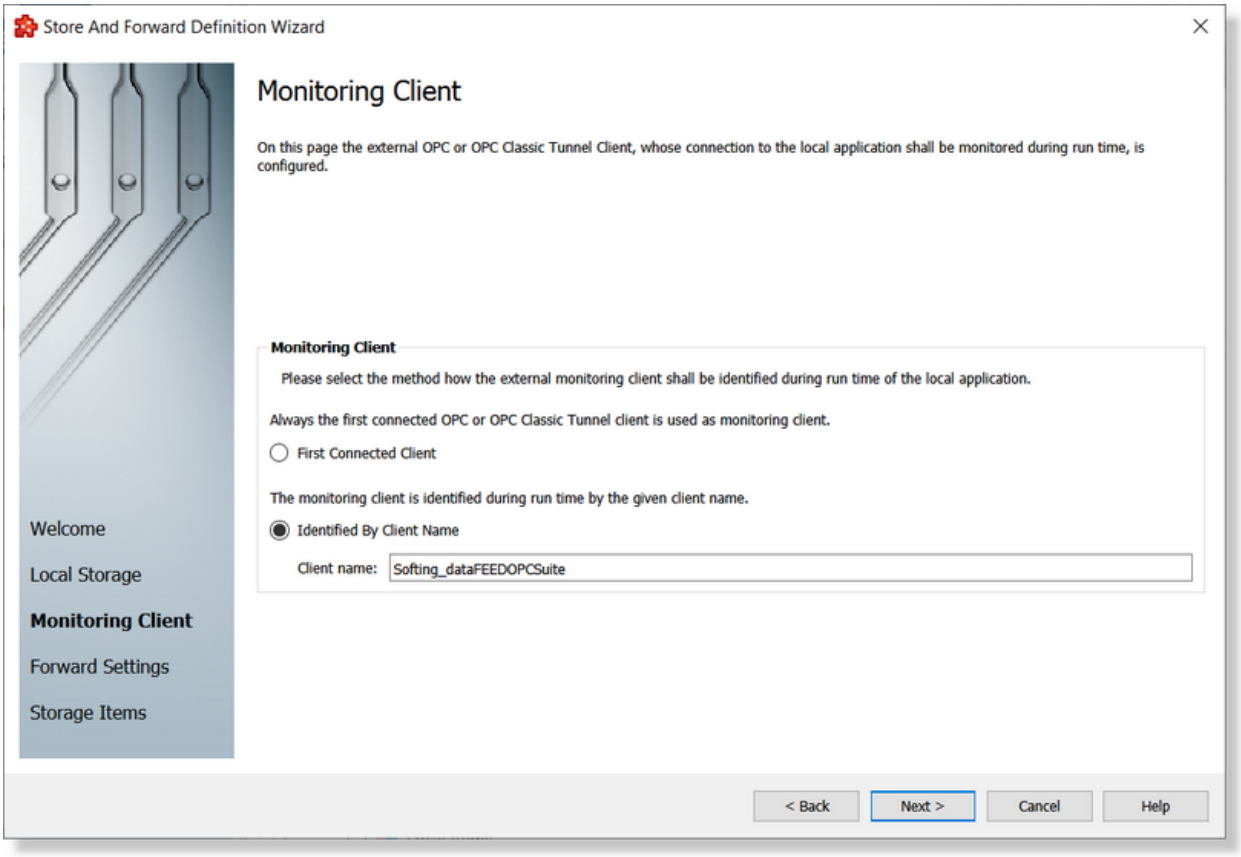
- The maximum number of transactions to be stored in the local data storage file. This value is entered in the *Max. transactions* input box.
- Whether to overwrite previously stored transactions when the above specified number has been reached. If the *Overwrite values* checkbox is not checked then no more data changes will be stored when the maximum number of transactions has been reached (this allows access to the first stored transactions). If the checkbox is checked the oldest transaction will be overwritten with the newest one when the maximum number of transactions has been reached (this allows access to the latest stored transactions).

© Softing Industrial Automation GmbH

Continue to the next wizard page by pressing the *Next* button, return to the previous one by using the *Back* button, or discard the changes made so far by clicking the *Cancel* button.

<<< Store And Forward Wizard - Welcome Store And Forward Wizard - Monitoring Client >>>

On the *Monitoring Client* wizard page you can configure the client monitoring settings for the local dataFEED OPC Suite application:



The monitoring client can be identified during the run time of the local dataFEED OPC Suite application either as the first connected client or by its name.

- If the *First Connected Client* option is selected, the stored data will be forwarded to the first client that connects. When the connected client disconnects from the server, the store mechanism is activated. The next client that successfully connects to the server after the initial client disconnected becomes the forward client, receiving the stored data. No normal clients (non-forward) may be connected to the server in this case.
- If the *Identified By Client Name* option is selected, the stored data will be forwarded to the client bearing a specific *Client name*. The input field for the client name must not be empty. When the name-identified forward client disconnects from the server, the store mechanism is activated. The stored data is forwarded to the next client of the specified name. Unlike the previous case, clients that do not have the specified name can connect to the server as

normal clients (they will receive data for their subscriptions); the forward client will still have to have the specified *Client Name*.

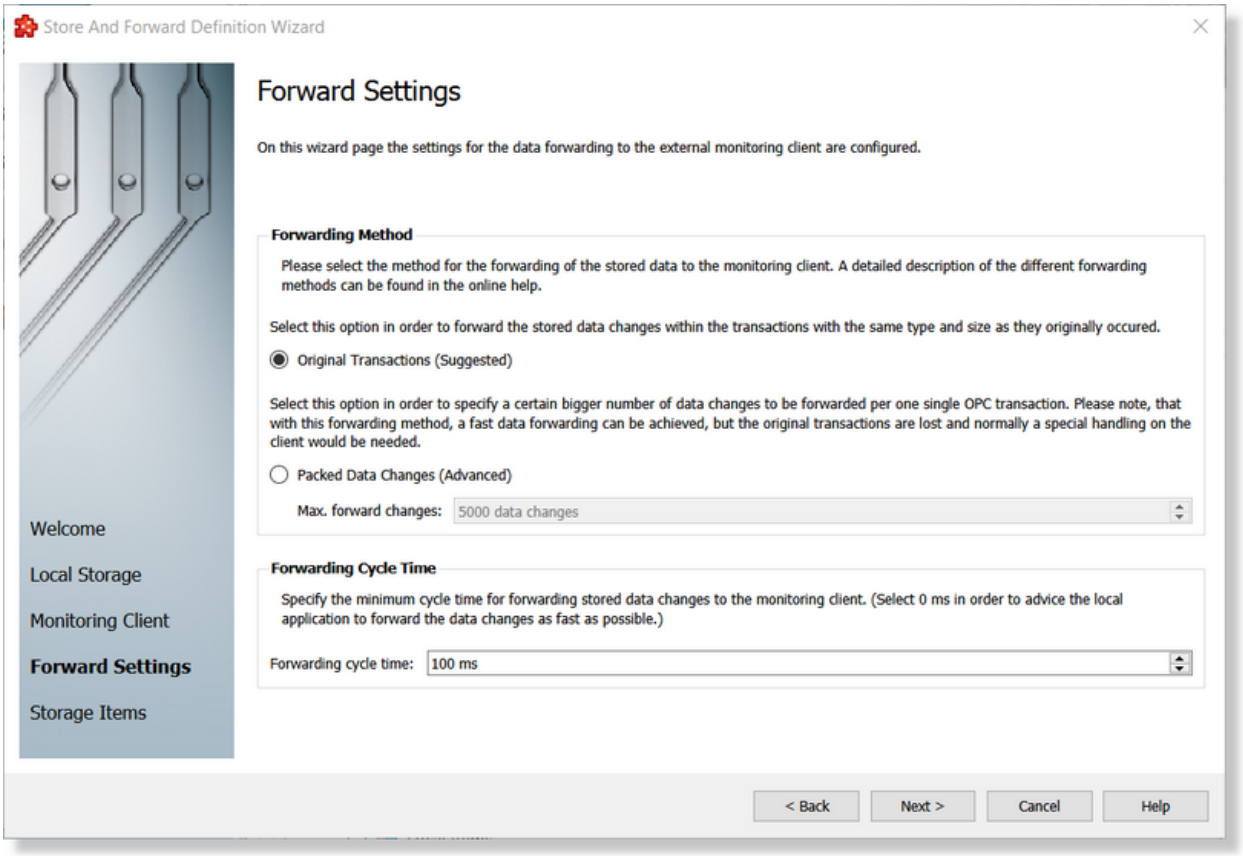
Note: In the *First Connected Client* mode only one client at a time can be connected to the server; no other clients are allowed to connect to the server after the first client has successfully connected.

Note: In the Identified By Client Name mode, if a client bearing a specific name connects to the server while another client with the same name is already connected, the previously connected client will be disconnected and the newly connected client will become the forward client. For a reliable functionality it is recommended to use only one client with the specific name as forwarding client.

Advance in the wizard by using the *Next* button, return to the *Local Storage* page by pressing the *Back* button or cancel the wizard with the *Cancel* button.

<<< Store And Forward Wizard - Local Storage Store And Forward Wizard - Forward Client >>>

On the *Forward Settings* wizard page the data forwarding settings can be configured:



© Softing Industrial Automation GmbH

Forwarding Method

In the Forwarding Method group you can choose between the following options:

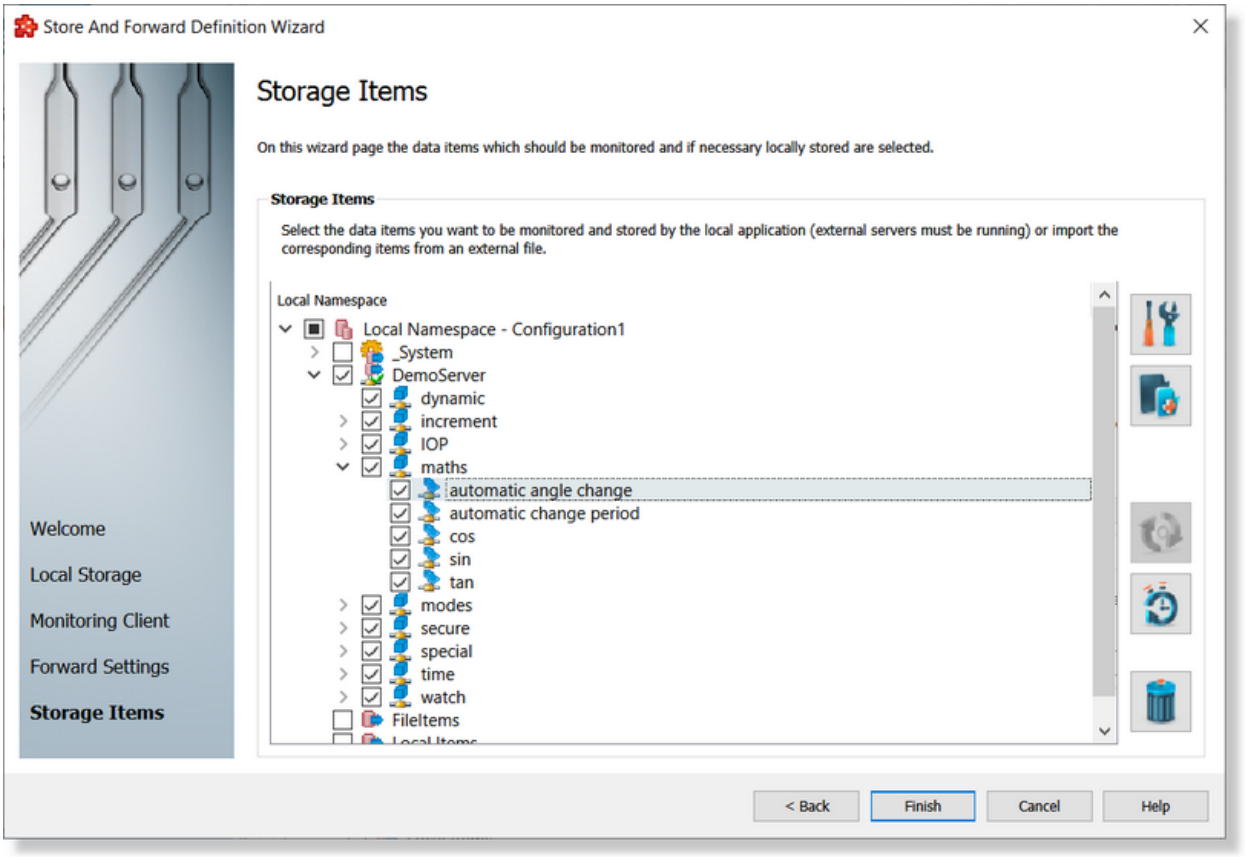
- If "Original Transactions" is selected, the stored data changes will be forwarded in transactions of the same type and size, in which they originally occurred. This is the recommended option.
- If "Packed Data Changes" is selected, the maximum number of forward changes also needs to be specified. This is an advanced feature. This option can be selected in order to specify a certain bigger number of data changes to be forwarded in a single OPC transaction. Note that with this forwarding method, a fast data forwarding can be achieved, but the original transactions are lost and normally a special handling on the client side would be needed.

Advance to the next wizard page by pressing the *Next* button, return to the previous one using the *Back* button or discard the changes made so far by clicking the *Cancel* button.

<<< Store And Forward Wizard - Monitoring Client Items

Store And Forward Wizard - Storage Items >>>

On the *Storage Items* wizard page you can select the items you want to monitor using the Store And Forward functionality:



This page is similar to the Configuration: Data Processing - Store And Forward page. There you will find a description of all the available actions.

Note: The expanded *Local Namespace* nodes will not maintain their folded/unfolded structure from the*Storage Items* wizard page to the *Data Processing - Store And Forward* main user interface page after pressing the *Finish* button. The selection of monitoring items will be retained, however.

Press the *Finish* button to save all the changes made in this wizard or discard all changes with the *Cancel* button.

<<< Store And Forward Wizard - Forward Client

7.3.2.1.4 File Format

The OPC Classic Store And Forward configuration settings are stored in two separate files:

- Task file which contains general descriptions for OPC Classic Store And Forward.
- Subscription file which contains all descriptions relating to the stored data:
    - which items are monitored for changes
    - the update rate and the execution condition for all the items.

© Softing Industrial Automation GmbH

Note: The file path and the file name of these configuration files are fixed and not configurable.

The task and subscription files are stored in the *Tasks* folder under the current configuration folder. Two files are created in the *Tasks* folder for the Store And Forward add-on: the *Tasks.txt* file which will hold the tasks and the *Subscriptions.txt* file which will contain the items selected for Store And Forward. The configuration file can be managed with the dataFEED OPC Suite Configurator or it can be manually edited.

Note: The recommendation is to use only the dataFEED OPC Suite Configurator to edit these files. This ensures that all the definitions will be valid.

If the OPC Classic Store And Forward configuration files are manually edited try loading the entire configuration in the dataFEED OPC Suite Configurator, as all the configuration files are automatically validated at load-time and, if there are any mistakes, an error message will be displayed. The dialog will contain an appropriate description and will point to the first invalid line of the local items configuration file.

The filter files are plain text files encoded in UTF-8 (thus providing full UNICODE compatibility) and they have a line-oriented structure. The elements of a line are tab-separated. Each line has the following format:

```
Line type<TAB>Line data
```

Task Configuration File

The following line types are defined in the Tasks file:

Line Type	Description
Store And Forward	Defines the OPC Classic Store And Forward task ID and eventually the forward client name.
Max. Item Changes	Defines the maximum number of data changes to be stored in the storage file.
Store Max. File Size	Defines the maximum size of the storage file which will raise an alarm if reached.
Store Folder	Defines the absolute path to the storage file.
Store File Name	Defines the name of the storage file.
Store Rollover Values	Defines the behavior of the store mechanism when the maximum data changes limit is reached: whether to overwrite the old data changes or whether to ignore the new data changes when the maximum number of data changes is reached and new data changes are available.
Forward Method	Defines when the forward process starts after a client is reconnected.

Forward Packed Data Changes	Defines the method used for forwarding the stored data to the monitoring client.
Forward Max. Data Changes	Defines the maximum number of data changes to be forwarded per transaction.
Forward Cycle Time	Defines the minimum interval of forwarding stored data changes to the monitoring client.
Commented Lines (#)	Any line that starts with the "#" character is considered a comment line which is ignored during evaluation.

The OPC Classic Store And Forward Line

The Store And Forward line defines the Store And Forward task ID and eventually the forward client name. A Store And Forward line has the following format:

```
STOREANDFORWARD<TAB>SF Task ID<TAB>CLIENTID<TAB>Client name
```

The OPC Classic Store And Forward line contains the following configurable parameters:

Parameter	Description
STOREANDFORWARD	The type of the line, predefined keyword.
SF Task ID	Store And Forward unique task identifier.
CLIENTID	Predefined keyword.
Client name	String defining the monitoring client name. If this value is an empty string then the data forwarding will start for the first connected client. The client name may contain any Unicode characters.

The Max. Item Changes Line

The Max. Item Changes line defines the maximum number of data changes to be stored in the storage file. A Max. Item Changes line has the following format:

```
STORE_MAXITEMCHANGES<TAB>SF Task ID<TAB>Data changes number
```

The Max. Item Changes line contains the following configurable parameters:

Parameter	Description

© Softing Industrial Automation GmbH

STORE_MAXITEMCHANGES	The type of the line, predefined keyword.
SF Task ID	Store And Forward unique task identifier.
Data changes number	Defines the maximum number of item data changes to be stored in the storage file.

The Store Max. File Size Line

The Store Max. File Size line defines the maximum size of the storage file which will raise an alarm if reached. A Store Max. File Size line has the following format:

```
STORE_MAXFILESIZE<TAB>SF Task ID<TAB>File Size Number<TAB>File Size Unit
```

The Store Max. File Size line contains the following configurable parameters:

Parameter	Description
STORE_MAXFILESIZE	The type of the line, predefined keyword.
SF Task ID	Store And Forward unique task identifier.
File Size Number	The number part of the maximum file size. Example: 200 if the maximum desired file size is 200MB.
File Size Unit	The unit part of the maximum file size. Allowed values are: KB, MB and GB. Example: MB if the maximum desired file size is 200MB.

The Store Folder Line

The Store Folder line defines the absolute path to the storage file. A Store Folder line has the following format:

```
STORE_FOLDER<TAB>SF Task ID<TAB>Folder Path
```

The Store Folder line contains the following configurable parameters:

Parameter	Description
STORE_FOLDER	The type of the line, predefined keyword.
SF Task ID	Store And Forward unique task identifier.

Folder Path	The absolute path for the Store And Forward file (see below).

The Store File Name Line

The Store File Name line defines the name of the storage file. A Store File Name line has the following format:

```
STORE_FILEBASENAME<TAB>SF Task ID<TAB>File Name
```

The Store File Name line contains the following configurable parameters:

Parameter	Description
STORE_FILEBASENAME	The type of the line, predefined keyword.
SF Task ID	Store And Forward unique task identifier.
File Name	The Store And Forward file name (which will be created in the above path).

The Store Rollover Values Line

The Store Rollover Values line defines the behavior of the store mechanism when the maximum data changes limit is reached: whether to overwrite the old data changes or whether to ignore the new data changes when the maximum number of data changes is reached and new data changes are available. A Store Rollover Values line has the following format:

```
STORE_ROLLOVERVALUES<TAB>SF Task ID<TAB>Rollover value
```

The Store Rollover Values line contains the following configurable parameters:

Parameter	Description
STORE_ROLLOVERVALUES	The type of the line, predefined keyword.
SF Task ID	Store And Forward unique task identifier.
Rollover value	The rollover value can be one of the following: *ON* or *OFF*. If rollover is *ON* and the maximum data changes limit is reached then the old values will be overwritten by the new ones; if it is *OFF* the new values will be ignored.

© Softing Industrial Automation GmbH

The Forward Monitoring Method Line

The Forward Monitoring Method line defines when the forward process starts after a client is reconnected. So far there is only one supported value: *OnClientConnected*. A Forward Monitoring Method line has the following format:

```
FORWARD<TAB>SF Task ID<TAB>Forward value
```

The Forward Monitoring Method line contains the following configurable parameters:

Parameter	Description
FORWARD	The type of the line, predefined keyword.
SF Task ID	Store And Forward unique task identifier.
Forward value	This value should always be *OnClientConnected*.

The Forward Packed Data Changes Line

The Forward Packed Data Changes line defines the method used for forwarding the stored data to the monitoring client. Softing dataFEED OPC Suite offers two forwarding methods:

- original transaction forwarding
- packed transaction forwarding

Original transaction forwarding forwards the initial transactions as they originally occurred; this method is recommended rather than packed transaction forwarding. Packed transaction forwarding forwards a specific bigger number of data changes (from more than one initial transaction) per single transaction. Define the maximum data changes to be forwarded per transaction in the Forward Max. Changes Line. A fast data forwarding can be achieved with this forwarding method but the original transactions are lost and therefore a special client side handling would be necessary.

A Forward Packed Data Changes line has the following format:

```
FORWARD_PACKED_DATA_CHANGES<TAB>SF Task ID<TAB>Packed Value
```

The Forward Packed Data Changes line contains the following configurable parameters:

Parameter	Description
FORWARD_PACKED_DATA_CHANGES	The type of the line, predefined keyword.

SF Task ID	Store And Forward unique task identifier.
Packed Value	This value can be *FALSE* or *TRUE*. *FALSE* means that the initial transactions will be forwarded as they were originally received. *TRUE* means that a specific greater number of data changes (from more than one initial transaction) will be packed in a forwarded transaction.

The Forward Max. Changes Line

The Forward Max. Changes line defines the maximum number of data changes to be forwarded per transaction. This line is only effective if the Forward Packed Data Changes line has the *Packed Value* set to *TRUE*. A Forward Max. Changes line has the following format:

```
FORWARD_MAXCHANGES<TAB>SF Task ID<TAB>Max. Changes Value
```

The Forward Max. Changes line contains the following configurable parameters:

Parameter	Description
FORWARD_MAXCHANGES	The type of the line, predefined keyword.
SF Task ID	Store And Forward unique task identifier.
Max. Changes Value	Defines the maximum number of item data changes to be forwarded per transaction.

The Forward Cycle Time Line

The Forward Cycle Time line defines the minimum interval of forwarding the stored data changes to the monitoring client. To advise the local dataFEED OPC Suite application to forward the data changes as fast as possible, the value *0* may be used. A Forward Cycle Time line has the following format:

```
FORWARD_CYCLE_TIME<TAB>SF Task ID<TAB>Cycle Time Value
```

The Forward Cycle Time line contains the following configurable parameters:

Parameter	Description
FORWARD_CYCLE_TIME	The type of the line, predefined keyword.
SF Task ID	Store And Forward unique task identifier.

© Softing Industrial Automation GmbH

Cycle Time Value	Defines the interval value expressed in milliseconds.

Task File - Example:

```
#     Store And Forward Add-on - configuration
STOREANDFORWARD   SFTask_ID   CLIENTID
STORE_MAXITEMCHANGES    SFTask_ID   100000
STORE_MAXFILESIZE SFTask_ID   1024  Mb
STORE_FOLDER      SFTask_ID   C:\Documents and Settings\All Users\Documents\dataFEED
STORE_FILEBASENAME      SFTask_ID   StoreAndForward
STORE_ROLLOVERVALUES    SFTask_ID   ON
FORWARD     SFTask_ID   OnClientConnected
FORWARD_MAXCHANGES      SFTask_ID   5000
FORWARD_PACKED_DATA_CHANGES   SFTask_ID   TRUE
FORWARD_CYCLE_TIME      SFTask_ID   100
```

Subscription Configuration File

The Data Processing - Store And Forward functionality generates multiple line types:

- common lines for all consumer add-ons: conditions, subscriptions and item trigger lines;
- specific lines for Store And Forward: STORE lines.

Common Lines:

The following lines are common lines and will be used to configure different functionalities:

Line Type	Description
Subscription	Defines the update rate with which the current action will be executed.
Condition	Defines the execution condition for the current action.
Item	This line defines a trigger which is fired if the item data has changed. The trigger data includes all other items which have changed at the same time (were reported in the same data change callback from the OPC server). An item data change consists of a different value or quality of the respective item compared to the previous one.
Commented Lines (#)	Any line that starts with the "#" character is considered a comment line which is ignored during evaluation.

The subscription file is divided into two sections:

- The global area:

The global area extends from the first subscription file line to the first trigger line (e.g.: ITEM line).

This area contains definitions valid for the entire subscription file.

- The local area:

  The local area covers all the actions between two trigger lines in the subscription file.

  The lines in this area are only executed if the parent trigger fires.

Note: In the common lines section there will always be two lines (a subscription line and a condition line) with empty IDs representing the global subscription and the global condition, respectively. The values in these lines can be accessed through the Global Alias And Initial Values dialog of the dataFEED OPC Suite Configurator.

Subscription Line

A subscription line has the following format:

```
SUBSCRIPTION<TAB>Subscription ID<TAB>UPDATERATE<TAB>Update rate value<TAB>ACTIVE<TAB>
```

The subscription line contains the following configurable parameters:

Parameter	Description
SUBSCRIPTION	The type of the line, predefined keyword.
Subscription ID	Unique subscription identifier.
UPDATERATE	Predefined keyword.
Update rate value	Defines the update rate of the current subscription.
ACTIVE	Predefined keyword.
Active value	Active value which may be *TRUE* or *FALSE*.

The subscription lines always contain the global subscription line as well as all subscriptions which are used at least once in consumer actions. The global subscription is represented by the subscription line with the empty *Subscription ID*. The default update rate value for the global subscription is *1000* milliseconds.

The value of the update rate is saved in milliseconds. This value may not be set to 0 inside the dataFEED OPC Suite Configurator but it can be manually set by editing the subscription file with an external text editor. An update rate value of 0 means *as fast as possible* but this is not recommended from a resource-related point of view.

Setting a subscription to be active or inactive is an advanced feature which cannot be configured from within the dataFEED OPC Suite Configurator. By default the active value is set to *TRUE* for every subscription (meaning that all subscriptions are active since creation). If a subscription's

© Softing Industrial Automation GmbH

active value is set to *FALSE* then the data changes of the items configured to be part of that subscription will be ignored.

Examples:

```
SUBSCRIPTION            UPDATERATE  1000  ACTIVE      TRUE
SUBSCRIPTION      Subscription_300  UPDATERATE  300   ACTIVE      TRUE
```

Condition Line

A condition line defines whether an action below an item trigger will be executed or not (when the item trigger is fired). This can be used in other configuration lines as well to condition the execution of an item action. A condition line has the following format:

```
CONDITION<TAB>Condition ID<TAB>Condition Statement
```

The condition line contains the following configurable parameters:

Parameter	Description
CONDITION	The type of the line, predefined keyword.
Condition ID	Unique condition identifier.
Condition statement	Single condition statement.

Conditions can be used to react to communication errors. Various actions may be executed depending on the quality of the item values. The following condition statements are supported:

Condition	Description
AllGoodQuality	The trigger actions are executed only if all items in the transaction have the quality *Good*.
Always	The trigger actions are executed always, ignoring the quality of the items in the incoming transaction.
ItemQualityChanged	The trigger actions are executed only if the quality of at least one item in the transaction has changed.
OneBadQuality	The trigger actions are executed if exactly one item in the transaction has the quality *Bad*.

The condition lines always contain the global condition line as well as all conditions which are used at least once in consumer actions. The global condition is represented by the condition line with the empty *Condition ID*. The default condition statement value for the global condition is

Always. As a general rule all the condition lines need to be declared in the global section of the subscription file. If no condition is defined the Always condition is applied as default.

Examples:

```
CONDITION          Always
CONDITION    AllGoodQuality     AllGoodQuality
```

Item Line

An item line is used to add an item to a subscription. The associated actions are executed each time the value or the quality of the specified data items changes during the run time of the local dataFEED OPC Suite application.

An item line has the following format:

```
ITEM<TAB>Subscription Item ID<TAB>Server Item ID<TAB>Subscription Id
```

The item line contains the following configurable parameters:

Parameter	Description
ITEM	The type of the line, predefined keyword.
Subscription Item ID	Unique subscription item identifier (implicitly created by concatenating the Server Item ID and the Subscription ID with the underscore character in between them).
Server Item ID	Item identifier within the local dataFEED OPC Suite application address namespace. This could be a server connection item ID or a local item ID.
Subscription ID	Subscription identifier (defined above, see Subscription Line).

This trigger is fired if the item data or quality has changed (also referred to as data change). The trigger data includes all other items which were included in the data change (items which had their data change reported in the same transaction).

Note: A single line consisting of the *Server Item ID* and *Subscription ID* is added to the subscription file for an item used several times with the same update rate in any consumer add-ons. This optimizes the local dataFEED OPC Suite application throughput.

Example:

```
ITEM  increment.I1_Subscription_300 increment.I1      Subscription_300
```

Note: The subscription item ID can be empty. In this case, however, only the Server Item ID can be used to identify the item.

© Softing Industrial Automation GmbH

Specific Lines:

The following line is Store And Forward specific:

Line Type	Description
STORE	The STORE line defines the Store And Forward action related to the item trigger.

A STORE line has the following format:

```
STORE<TAB>Condition ID<TAB>SF Task ID
```

The STORE line contains the following configurable parameters:

Parameter	Description
STORE	The type of the line, predefined keyword.
Condition ID	Condition identifier (defined above, see Condition Line).
SF Task ID	Store And Forward unique task identifier.
Subscription Item ID	Subscription item identifier (defined above, see Item Trigger Line).

Subscription File - Example:

```
CONDITION         Always
CONDITION   AllGoodQuality    AllGoodQuality
#
SUBSCRIPTION            UPDATERATE  1000  ACTIVE      TRUE
SUBSCRIPTION      Subscription_2000 UPDATERATE  2000  ACTIVE      TRUE
#
#
ITEM  OPC_1.increment.I1_Subscription_2000      OPC_1.increment.I1      Subscription_
STORE AllGoodQuality    SFTask_ID   OPC_1.increment.I1_Subscription_2000
#
ITEM  OPC_1.increment.I4_Subscription_2000      OPC_1.increment.I4      Subscription_
STORE AllGoodQuality    SFTask_ID   OPC_1.increment.I4_Subscription_2000
#
ITEM  OPC_1.maths.cos_  OPC_1.maths.cos
STORE       SFTask_ID   OPC_1.maths.cos_
#
ITEM  OPC_1.maths.sin_  OPC_1.maths.sin
STORE       SFTask_ID   OPC_1.maths.sin_
#
ITEM  OPC_1.watch.device 1.value 1_ OPC_1.watch.device 1.value 1
STORE       SFTask_ID   OPC_1.watch.device 1.value 1_
#
```

```
ITEM  OPC_1.watch.device 1.value 2_ OPC_1.watch.device 1.value 2
STORE       SFTask_ID   OPC_1.watch.device 1.value 2_
```

7.3.2.1.5 Import Wizard

The Softing Configurator allows the import of OPC Classic Store And Forward items from an external file.

The external file should be a plain-text tab-separated file with the specific OPC Classic Store And Forward File Format.

The *OPC Classic Store And Forward Import Wizard* can be triggered by selecting the *Import* action  from the *Monitoring Items* tree context sensitive menu. Depending on the current tree selection the import action may be allowed or not. After the import file is successfully parsed the selected tree item sub-branch is verified whether it contains the imported items or not. If the item sub-branch contains the imported items then these items are marked for monitoring. If the imported items do not exist in the current item's sub-branch they will be created and marked for monitoring (if the current item's server connection supports dynamic items). If the sub-branch of the selected item does not contain the imported items and the selected item's server connection does not support dynamic items then the non-existing items will not be created.

Note: If only part of the imported items are contained in the sub-branch of the currently selected item and the item's server connection does not support dynamic items then the import is done only partly: the existing items are marked for monitoring and the non-existing items are ignored.

When the import action is triggered the *OPC Classic Store And Forward Import Wizard* is started.

- The OPC Classi Store And Forward Import Wizard allows importing different OPC Classic Store And Forward items and actions:

  - Select the file to be imported @ Import Wizard - Import File Selection
  - Examine the import results @ Import Wizard - Import Results

The OPC Classic Store And Forward actions can be imported into the namespace of the OPC Classic Store And Forward add-on from a text file with a specific format. The external filter files should be plain text files and have a line-oriented structure. The elements of a line must be tab-separated. For each Store And Forward data item a line is added to the file. The number of elements of a specific line depends on the line type. For each item that should be marked for monitoring a Store And Forward action line (STORE) must exist in the file to be imported.

In an import file for Data Processing - OPC Classic Store And Forward, multiple line types can be found:

© Softing Industrial Automation GmbH

- common lines for all consumer add-ons: conditions, subscriptions and item trigger lines;
- specific lines for Store And Forward: STORE lines.

Common Lines:

The following lines are common lines and will be used to configure different functionalities:

Line Type	Description
Subscription	Defines the update rate with which the current action will be executed.
Condition	Defines the execution condition for the current action.
Item	This line defines a trigger which is fired if the item data has changed. An item data change consists of a different value or quality of the respective item compared to the previous one.
Commented Lines (#)	Any line that starts with the "#" character is considered a comment line which is ignored during evaluation.

The subscription file is divided into two sections:

- The global area

  The global area extends from the first subscription file line to the first trigger line (e.g.: ITEM line).

  This area contains definitions valid for the entire subscription file.

- The local area

  The local area covers all the actions between two trigger lines in the subscription file.

  The lines in this area are only executed if the parent trigger fires.

Note: In the common lines section there will always be two lines (a subscription line and a condition line) with empty IDs representing the global subscription and the global condition, respectively. The values in these lines can be accessed through the Global Alias And Initial Values dialog of the dataFEED OPC Suite Configurator.

Subscription Line

A subscription line has the following format:

```
SUBSCRIPTION<TAB>Subscription ID<TAB>UPDATERATE<TAB>Update rate value<TAB>ACTIVE<TAB>
```

The subscription line contains the following configurable parameters:

Parameter	Description
SUBSCRIPTION	The type of the line, predefined keyword.
Subscription ID	Unique subscription identifier.
UPDATERATE	Predefined keyword.
Update rate value	Defines the update rate of the current subscription.
ACTIVE	Predefined keyword.
Active value	Active value which may be *TRUE* or *FALSE*.

The subscription lines always contain the global subscription line as well as all subscriptions which are used at least once in consumer actions. The global subscription is represented by the subscription line with the empty *Subscription ID*. The default update rate value for the global subscription is *1000* milliseconds.

The value of the update rate is saved in milliseconds. This value may not be set to 0 inside the dataFEED OPC Suite Configurator but it can be manually set by editing the subscription file with an external text editor. An update rate value of 0 means *as fast as possible* but this is not recommended from a resource-related point of view.

Setting a subscription to be active or inactive is an advanced feature which cannot be configured from within the dataFEED OPC Suite Configurator. By default the active value is set to *TRUE* for every subscription (meaning that all subscriptions are active since creation). If a subscription's active value is set to *FALSE* then the data changes of the items configured to be part of that subscription will be ignored.

Examples:

```
SUBSCRIPTION            UPDATERATE  1000  ACTIVE      TRUE
SUBSCRIPTION      Subscription_300  UPDATERATE  300   ACTIVE      TRUE
```

Condition Line

A condition line defines whether an action below an item trigger will be executed or not (when the item trigger is fired). This can be used in other configuration lines as well to condition the execution of an item action. A condition line has the following format:

```
CONDITION<TAB>Condition ID<TAB>Condition Statement
```

The condition line contains the following configurable parameters:

Parameter	Description
CONDITION	The type of the line, predefined keyword.

© Softing Industrial Automation GmbH

Condition ID	Unique condition identifier.
Condition statement	Single condition statement.

Conditions can be used to react to communication errors. Various actions may be executed depending on the quality of the item values. The following condition statements are supported:

Condition	Description
AllGoodQuality	The trigger actions are executed only if all items in the transaction have the quality *Good*.
Always	The trigger actions are executed always, ignoring the quality of the items in the incoming transaction.
ItemQualityChanged	The trigger actions are executed only if the quality of at least one item in the transaction has changed.
OneBadQuality	The trigger actions are executed if exactly one item in the transaction has the quality *Bad*.

The condition lines always contain the global condition line as well as all conditions which are used at least once in consumer actions. The global condition is represented by the condition line with the empty *Condition ID*. The default condition statement value for the global condition is *Always*. As a general rule all the condition lines need to be declared in the global section of the subscription file. If no condition is defined the Always condition is applied as default.

Examples:

```
CONDITION          Always
CONDITION    AllGoodQuality     AllGoodQuality
```

Item Trigger Line

An item line is used to add an item to a subscription. The associated actions are executed each time the value or the quality of the specified data items changes during the run time of the local dataFEED OPC Suite application An item line has the following format:

```
ITEM<TAB>Subscription Item ID<TAB>Server Item ID<TAB><Subscription Id>
```

The item line contains the following configurable parameters:

Parameter	Description
ITEM	The type of the line, predefined keyword.

Subscription Item ID	Unique subscription item identifier (implicitly created by concatenating the Server Item ID and the Subscription ID with the underscore character in between them).
Server Item ID	Item identifier within the local dataFEED OPC Suite application address namespace. This could be a server connection item ID or a local item ID.
Subscription ID	Subscription identifier (defined above, see Subscription Line).

This trigger is fired if the item data or quality has changed (also referred to as data change). The trigger data includes all other items which were included in the data change (items which had their data change reported in the same transaction).

Note: A single line consisting of the *Server Item ID* and *Subscription ID* is added to the subscription file for an item used several times with the same update rate in any consumer add-ons. This optimizes the local dataFEED OPC Suite application throughput.

Example:

```
ITEM  increment.I1_Subscription_300 increment.I1      Subscription_300
```

Note: The subscription item ID can be empty. In this case, however, only the Server Item ID can be used to identify the item.

Specific Lines:

The following line is Store And Forward specific:

Line Type	Description
STORE	The STORE line defines the Store And Forward action related to the item trigger.

A STORE line has the following format:

```
STORE<TAB>Condition ID<TAB>SF Task ID
```

The STORE line contains the following configurable parameters:

Parameter	Description
STORE	The type of the line, predefined keyword.
Condition ID	Condition identifier (defined above, see Condition Line).

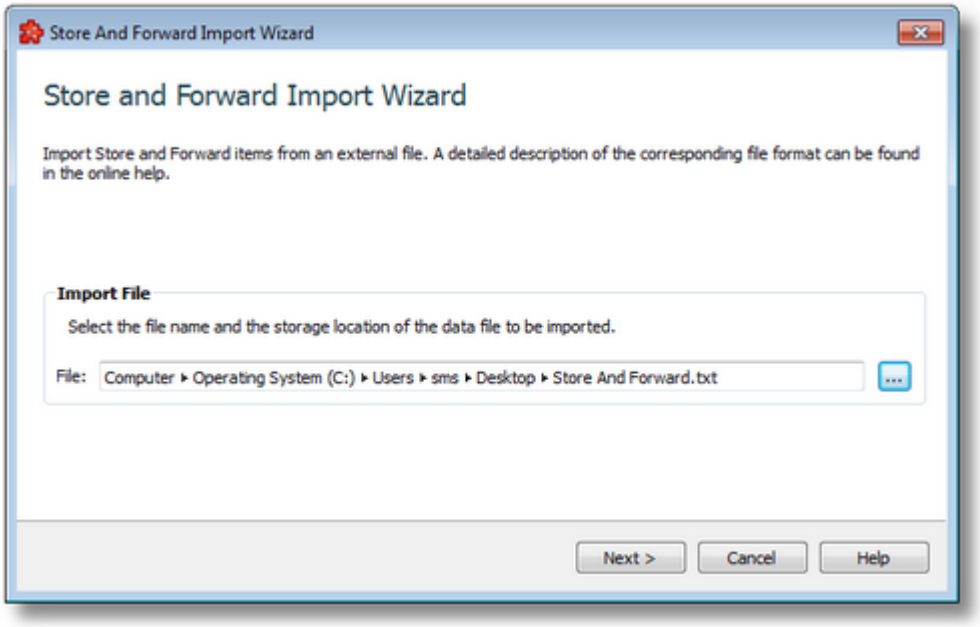
© Softing Industrial Automation GmbH

SF Task ID	Store And Forward unique task identifier.
Subscription Item ID	Subscription item identifier (defined above, see Item Trigger Line).

Subscription File - Example:

```
CONDITION         Always
CONDITION   AllGoodQuality    AllGoodQuality
#
SUBSCRIPTION            UPDATERATE  1000  ACTIVE      TRUE
SUBSCRIPTION      Subscription_2000 UPDATERATE  2000  ACTIVE      TRUE
#
#
ITEM  OPC_1.increment.I1_Subscription_2000      OPC_1.increment.I1      Subscription_
STORE AllGoodQuality    SFTask_ID   OPC_1.increment.I1_Subscription_2000
#
ITEM  OPC_1.increment.I4_Subscription_2000      OPC_1.increment.I4      Subscription_
STORE AllGoodQuality    SFTask_ID   OPC_1.increment.I4_Subscription_2000
#
ITEM  OPC_1.maths.cos_  OPC_1.maths.cos
STORE       SFTask_ID   OPC_1.maths.cos_
#
ITEM  OPC_1.maths.sin_  OPC_1.maths.sin
STORE       SFTask_ID   OPC_1.maths.sin_
#
ITEM  OPC_1.watch.device 1.value 1_ OPC_1.watch.device 1.value 1
STORE       SFTask_ID   OPC_1.watch.device 1.value 1_
#
ITEM  OPC_1.watch.device 1.value 2_ OPC_1.watch.device 1.value 2
STORE       SFTask_ID   OPC_1.watch.device 1.value 2_
```

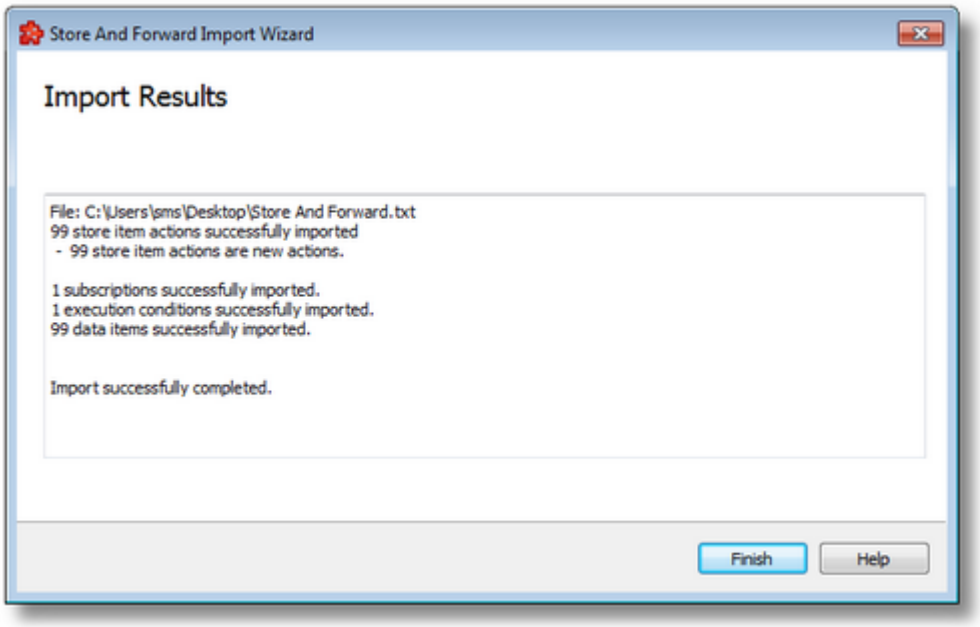
The Import File Selection wizard page allows you to choose a file to be imported. The file can be selected by manually typing its name in the *File* input box or by clicking the browse button to the right of the input box (a standard file selection dialog will open).



The file path and name should be valid (they should point to an existing file). When the *Next* button is pressed the file path and name are verified. If the file identified by the specified file path and name does not exist an error message is displayed. The actual import begins if the selected file is valid. If the import takes longer to complete a waiting dialog is displayed. When the import operation is finished the *Import Results* page is displayed.

Import Wizard - Import Results >>>

The Import Results wizard page displays the import results.



© Softing Industrial Automation GmbH

This page is the final page of the *Store And Forward Import Wizard*. A back button is not provided as the import action cannot be undone. The import can be successful or it can fail:

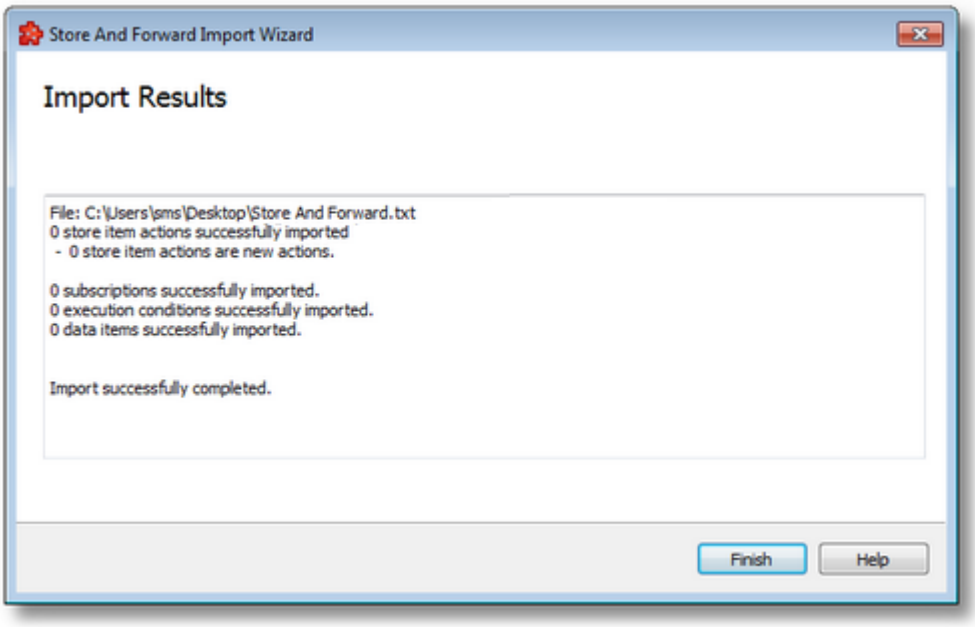
1. After a successful import the results page contains the following information:

   - The absolute path of the imported file.
   - The number of imported Store And Forward items (actions).
   - The number of newly imported (created) Store And Forward items.
   - The number of subscriptions imported.
   - The number of execution conditions imported.
   - The number of data items imported.
   - The confirmation message that the import was successful.

2. After a failed import the results page contains the following information:

   - The invalid line where the import failed.
   - The reason why the line is considered invalid.
   - The confirmation message that the import failed.

An *Import Results* example for a failed import operation can be seen below:



When the import wizard is closed after a successful import, the new items are marked (or, if the server connection supports dynamic items, they are added and then marked) for monitoring in the Store And Forward item tree.

<<< Import Wizard - Import File Selection

7.3.2.1.6 Export Wizard

The dataFEED OPC Suite Configurator allows the export of Store And Forward items and actions to an external file.

The external file will be a tab-separated file with the specific OPC Classic Store And Forward file format.

The export action can be triggered by selecting the *Export* action  from the *Monitoring Items* tree context sensitive menu. Depending on the current tree selection the export action may be allowed or not.

The exported file will contain all the items marked for Store And Forward monitoring, with all of the execution conditions, subscriptions and triggers.

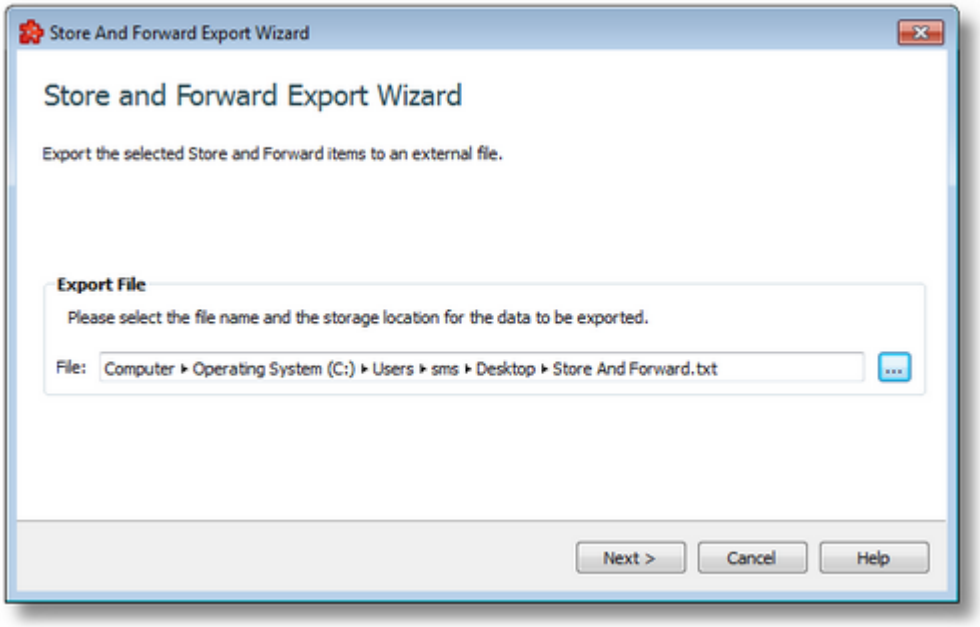
When the export action is triggered the *Store And Forward Export Wizard* is started.

- The OPC Classic Store And Forward Export Wizard allows exporting OPC Classic Store And Forward items and actions:

  - Select the file the Store And Forward items and actions are to be exported to @ Export Wizard - Export File Selection
  - Examine the export results @ Export Wizard - Export Results

The Export File Selection wizard page allows you to choose a file path and name for the exported file. The file can be selected by manually typing its path and name in the *File* input box or by clicking the browse button located to the right of the input box (a standard file saving dialog will open).

© Softing Industrial Automation GmbH

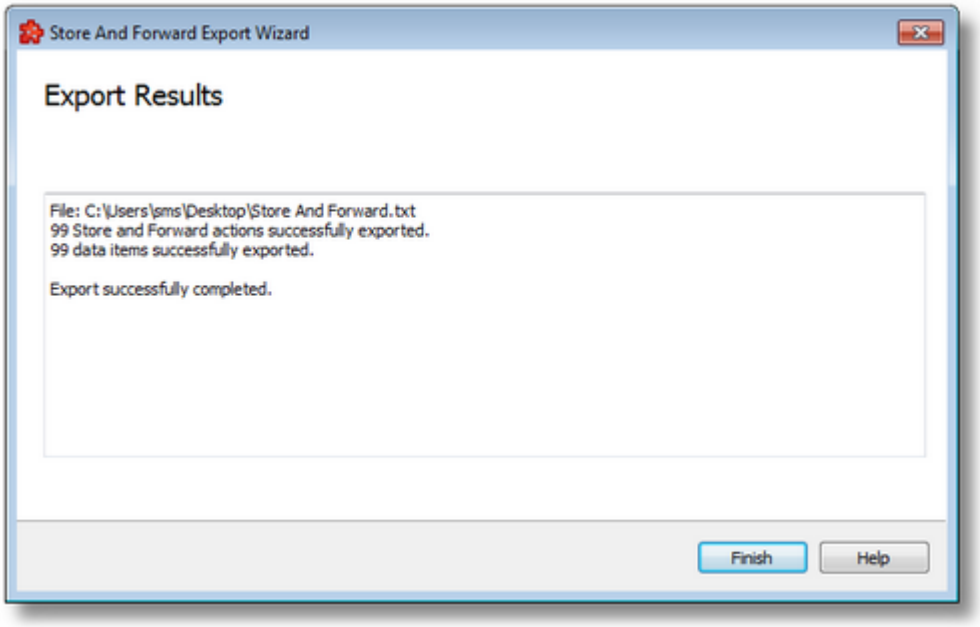


When the *Next* button is pressed the file path is verified and if the path does not exist, an error message is displayed (invalid path). If the specified file path is valid and a file with the specified name already exists, a message box will ask whether to overwrite the existing file or not (select another file). The actual Store And Forward export operation begins when the path to the file is valid and either a file with the specified name does not exist or it exists and overwrite has been chosen.

When the export operation is finished the *Export Results* page is displayed.

Export Wizard - Export Results >>>

The Export Results wizard page displays the export results.



This page is the final page of the Store And Forward Export Wizard. A back button is not provided as the export action cannot be undone. The export can be successful or it can fail:

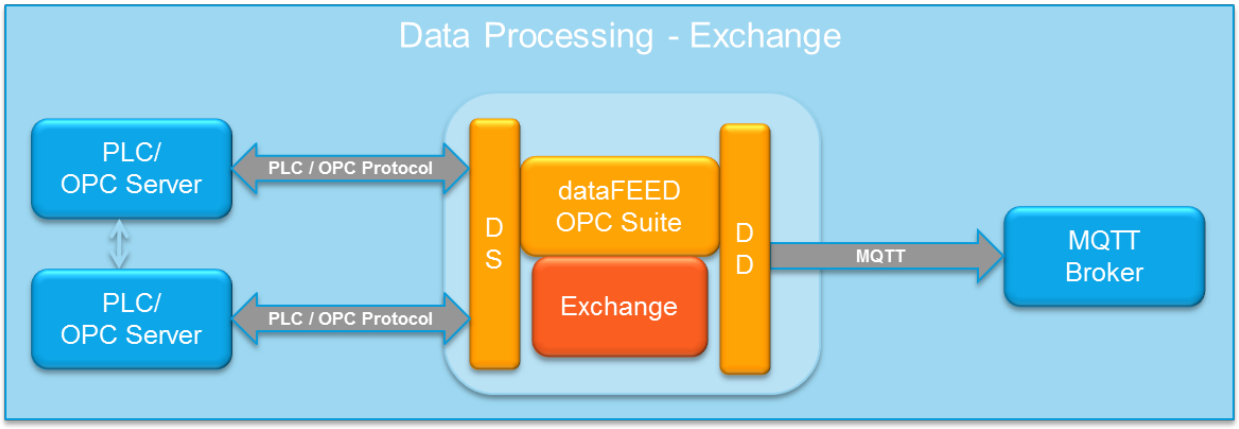
1. After a successful export the results page contains the following information:
   - The absolute path of the exported file.
   - The number of exported Store And Forward actions.
   - The number of exported Store And Forward data items.
   - The confirmation message that the export was successful.
2. After a failed export the results page contains the following information:
   - The reason why the export failed.
   - The confirmation message that the export failed.

<<< Export Wizard - Export File Selection

7.3.2.2 Exchange

This section of the Softing dataFEED OPC Suite Online Help describes how to configure the Exchange functionality within Softing dataFEED OPC Suite.

© Softing Industrial Automation GmbH



The OPC Exchange add-on is used for exchanging data between two OPC servers. The data change received from one item of the local dataFEED OPC Suite application's namespace will be written into another item of the local dataFEED OPC Suite application namespace normally residing on another server. It is also possible to exchange data within a single OPC server or between distinct Local Items. An exchange action always comprises a source item and a destination item whereby the source item value is written into the destination item.

Note: The source item of an exchange action should allow reading and the destination item should allow writing.

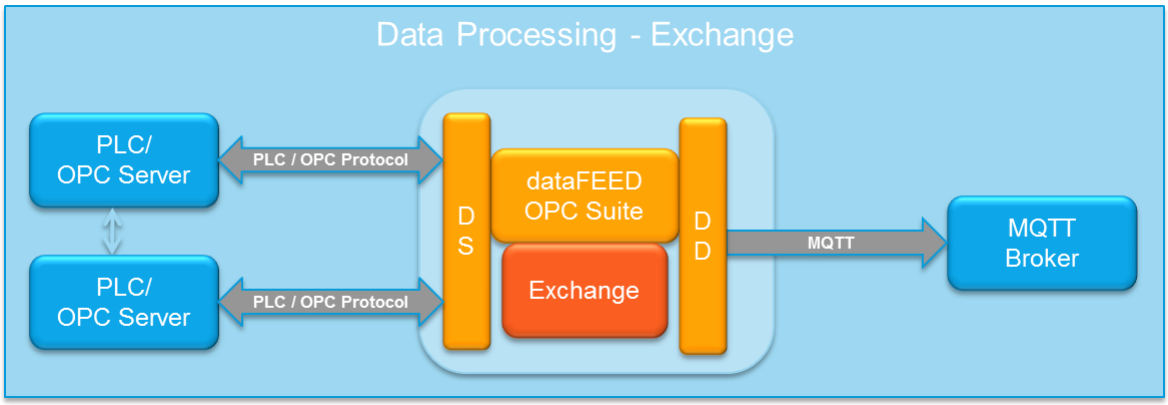
Exchange Action

An *Exchange Action* contains the following attributes:

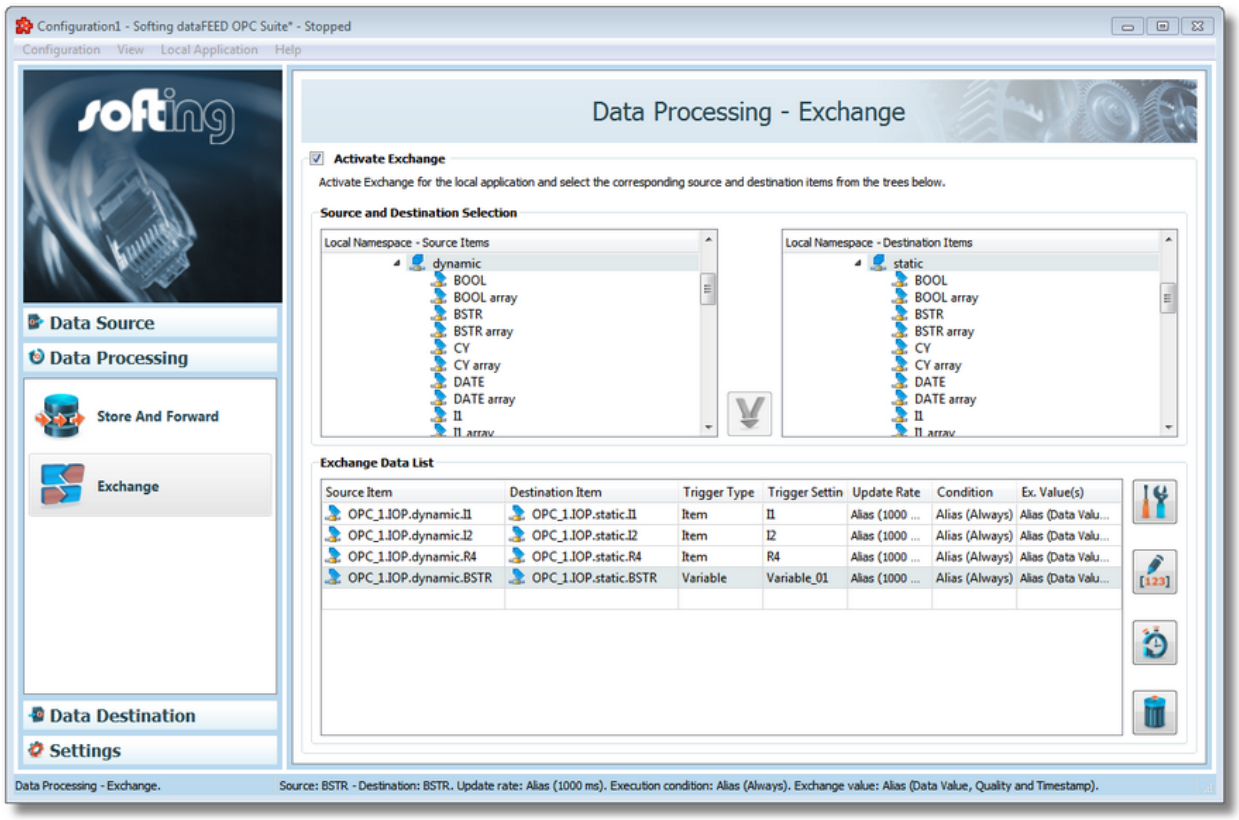
1. Source item
2. Destination item
3. Update rate
4. Condition
5. Exchange value(s)

An *Exchange Action* may be created simply by selecting items from the source and destination trees and adding them to the exchange actions table. When a new *Exchange Action* is created, it receives the default initial update rate, execution condition and exchange value(s), parameters that can be easily changed later. Specific details about how to create and handle the Exchange actions can be found on the Configuration: Data Processing - Exchange page. A description of the *Exchange* file format can be found on the Exchange File Format page.

7.3.2.2.1 Configuration



The *Exchange* add-on main user interface allows configuring the Exchange mechanism. This main user interface can be reached by selecting the Exchange option from the Data Processing section in the navigation panel on the left.



The main Exchange user interface has the following components:

- *Activate Exchange* checkbox
- *Source and Destination Selection* item trees
- Exchange *Data List* table.

The exchange actions can be configured by using the buttons, the context menu, the keyboard shortcuts or by performing drag and drop actions.

© Softing Industrial Automation GmbH

Activate Exchange

The Exchange add-on has to be activated before you can use it. If the Exchange add-on is not activated then the module is not loaded by the local dataFEED OPC Suite application at run time. When Exchange is deactivated the configured data is not deleted but only disabled. The data is available again when the Exchange add-on is reactivated.

Exchange Trees Handling

The source and destination trees contain the entire dataFEED OPC Suite namespace. Any exchange items of type Tag or Property which satisfy the access rights rule can be used as source or destination. The source items should at least have the access right Read and the destination items should at least have the access right Write.

Note: Items of type Node can also be used as exchange sources or destinations if they satisfy the access rights constraint.

The dataFEED OPC Suite root node and the data sources cannot be used as exchange sources or destinations. Items of type Tag or Property which are not valid cannot be used as exchange sources or destinations, therefore these items are disabled (grayed out).

Note: The same item cannot be used as source and destination in the same exchange action, therefore the *Insert a new Exchange action* button is disabled. If such an exchange action is created nevertheless – through the context menu or by performing a drag and drop operation – an error dialog will be displayed and the respective item will not be transferred to the current exchange action.

In both the source and the destination trees the context menu is dependent on the active tree selection. Multiple selection is disabled in both trees.

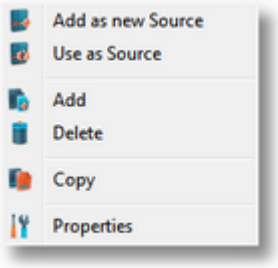
Operation

From the Exchange main user interface certain operations can be performed over the selected exchange action as well as for the selected source and destination items. The operations are easily accessible using the buttons on the right, accessing the context sensitive menu or by pressing the keyboard shortcuts. The following actions are available:

- Insert a new exchange action for the selected source and destination items using the Insert a new Exchange action button or pressing the keyboard shortcut CTRL+N.
- Change the properties of the selected Exchange action using the Properties of the selected Exchange action button, pressing the keyboard shortcut CTRL+P or selecting the *Properties* action from the context menu.
- Add a new Exchange action using the Add a new Exchange action button, pressing the keyboard shortcut CTRL+A or selecting the *Add a new Exchange action* action from the context menu.

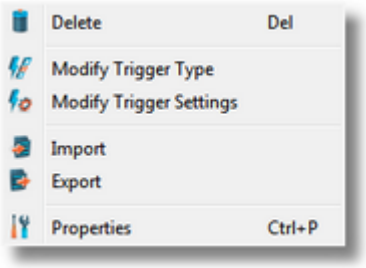
- Configure global alias and initial values using the Configure global alias and initial values button  or pressing the keyboard shortcut CTRL+D.
- Remove the selected Exchange action(s) using the Delete button  , pressing the DEL / DELETE key or selecting the *Delete* action from the context menu.
- Use the selected Source Item in the current Exchange action by selecting the Use As Source Item action from the Local Namespace - Source Items tree context menu.
- Use the selected Destination Item in the current Exchange action by selecting the Use As Destination Item action from the Local Namespace - Destination Items tree context menu.
- Use the selected Source Item in a new Exchange action by selecting the Insert New Source Item action from the Local Namespace - Source Items tree context menu.
- Use the selected Destination Item in a new Exchange action by selecting the Insert New Destination Item action from the Local Namespace - Destination Items tree context menu.
- Specific tree handling actions are available through the context menu for the Source and Destination Selection trees. For more details see the Tree Handling chapter.
- Import Exchange actions from an external file by selecting the Import action from the context menu.
- Export the selected Exchange actions to an external file by selecting the Export action from the context menu.
- Modify the trigger type for the selected exchange action selecting the Modify Trigger Type action from the context menu.
- Modify the trigger settings for the selected exchange action selecting the Modify Trigger Settings action from the context menu. For more information see the Item Trigger wizard or the Variable Trigger wizard page.
- Switch to Advanced Exchange action via Advance Exchange Action Editor by using Modify button  , pressing the keyboard shortcut CTRL+M or selecting the Switch to Advanced Exchange action from the context menu.

An example of the *Local Namespace - Source Items* context menu over a Tag item originating from a server connection can be seen below:



An example of the *Exchange Data List* context menu can be seen below:

© Softing Industrial Automation GmbH



Note: The item trees from the *Source and Destination Selection* section display the item's attributes as tool tips.

Note: The *Exchange Data List* allows multiple selection therefore the available actions depend on the active selection (empty line selected, single line selected or multiple lines selected). An Exchange line contains all the attributes of the respective Exchange action.

Exchange Action Attributes

Exchange Attribute	Description
Source item	The source item ID consisting of the item's complete hierarchical name.
Destination item	The destination item ID consisting of the item's complete hierarchical name.
Trigger Type	The trigger type of the actual exchange action. The default trigger type is *Item*.
Trigger Settings	The trigger of the actual exchange action.
Update rate	The rate at which the data values are polled from the source item and written to the destination item. The entries marked with *Alias* bear the global alias value of the configuration. The alias value can be changed through the Global Alias Values dialog.
Condition	The execution condition of the actual exchange action. The entries marked with *Alias* bear the global alias value of the configuration. The alias value can be changed through the Global Alias Values dialog.
Exchange value(s)	The exchange value(s) are the value(s) to be transferred from the source to destination items.

	The entry marked with "Alias" is the global alias value for the whole configuration. The alias value can be changed from the Global Alias Values dialog.

Add an Exchange Action

Exchange actions can be added to the Exchange Data List by:

- Performing drag and drop actions from the *Source and Destination Selection* trees into an empty line of the *Exchange Data List*, targeting the *Source Item* or *Destination Item* columns.
- Using the *Insert new Exchange action* button  when two valid items from the *Source Items* and *Destination Items* trees under the *Source and Destination Selection* section are selected.
- Consecutively selecting the *Insert New Source Item* (or the *Insert New Destination Item*) and the *Use as Destination Item* (or the *Use as Source Item*) actions from the context menu of the *Source Items* and *Destination Items* trees (no Exchange action must be selected).
- Using the *Add a new Exchange action* button  .
- Selecting the *Insert New Exchange Action* from the *Exchange Data List*'s context sensitive menu.

Global Alias and Initial Values

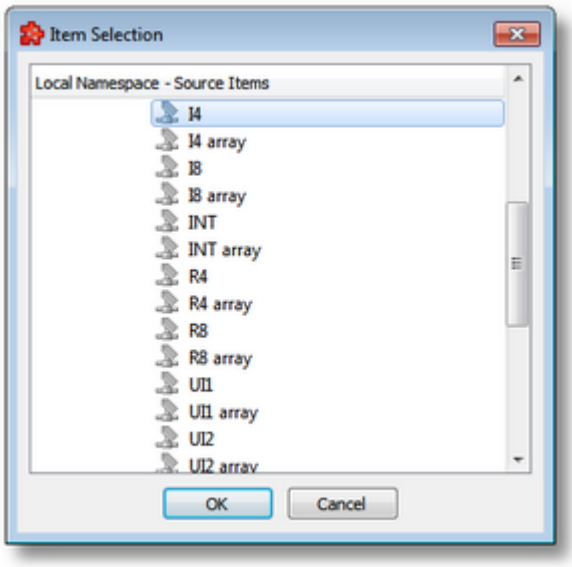
All the newly added exchange actions bear the default values for the update rate, execution condition and exchange value(s). These values are the ones set in the Global Alias Values and Global Initial Values dialog. The respective dialog can be reached using the *Configure global alias and initial values* button  .

Modify an Exchange Action

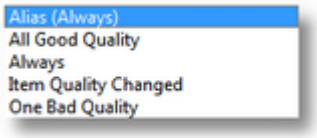
Exchange actions in the Exchange Data List can be modified by:

- Performing drag and drop actions from the *Source and Destination Selection* trees into an existing line of the *Exchange Data List*, targeting the *Source Item* or *Destination Item* columns.
- Selecting either the *Use as Source Item* or the *Use as Destination Item* action from the context menu of the *Source Items* and *Destination Items* trees (an Exchange action must be selected).
- Using the *Properties of the selected Exchange action* button  .
- Double-clicking an existing Exchange action in the Exchange Data List.
- Selecting the *Properties* action from the *Exchange Data List*'s context sensitive menu.

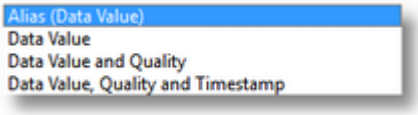
© Softing Industrial Automation GmbH



- Click on a selected line in the update rate cell. A spin box appears which allows modifying the update rate  .
- Click on a selected line in the condition cell. A drop down box appears which allows modifying the condition.



- Click on a selected line in the exchange value(s) cell. A drop down box appears which allows modifying the exchange value(s).



- The Properties button  on the right side of the table opens the Exchange Action dialog for the selected exchange action.
- Double clicking on an Exchange action row in the data list opens the Exchange Action dialog for that selected exchange action.
- "Properties" from the context sensitive menu opens the Exchange Action dialog for that selected exchange action.

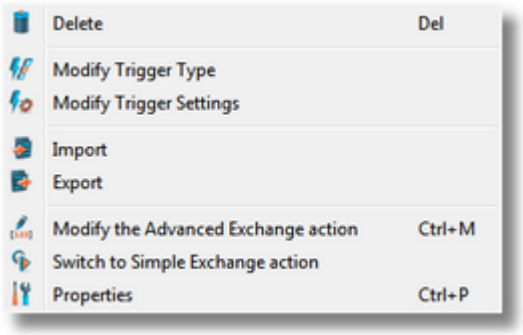
Advanced Exchange Action Editor

Attention: This functionality is applied only for **data destination items** with **STRING (String) data type** and **Plain Text** publish format (for MQTT destination tags).

This option allow you to format the data value destination by inserting text lines and placeholders:

- The *Modify the Advanced Exchange action* or button on the right side of the table opens the Advance Exchange Action Editor dialog.
- The *Switch to Simple Exchange action* allow to restore an *Advanced* action to *Simple* action.

An example of the *Advanced Exchange Data List* context menu can be seen below:



Delete an Exchange Action

To remove the desired Exchange action(s) from the *Exchange Data List*, select the Exchange action(s) to be deleted and click the *Delete the selected Exchange actions* button , press the DEL / DELETE key or select the *Delete* action from the context sensitive menu.

Import / export Exchange Action(s)

The selected action(s) in the *Exchange Data List* can be exported or external Exchange action(s) can be imported in order to enhance the configurability. The import / export wizards can be triggered by selecting the *Import* and *Export* context menu actions over the selected Exchange action(s). For more details concerning the import / export actions see the Import Wizard and Export Wizard pages.

Status Bar

The status bar contains the properties of the selected item from the control that has the focus.

If the focus is in one of the trees, the status bar contains the properties of the selected item:

- item class
- item ID which is its hierarchical name in the dataFEED OPC Suite tree
- whether the item is used in any Exchange action or not. If it is used in only one Exchange action, then the Exchange action information is also displayed: update rate and execution condition. If it is used in multiple Exchange actions, then the number of actions in which it is used is displayed.

© Softing Industrial Automation GmbH

If the focus is in the Exchange actions data list, then the properties of the selected Exchange action are displayed:

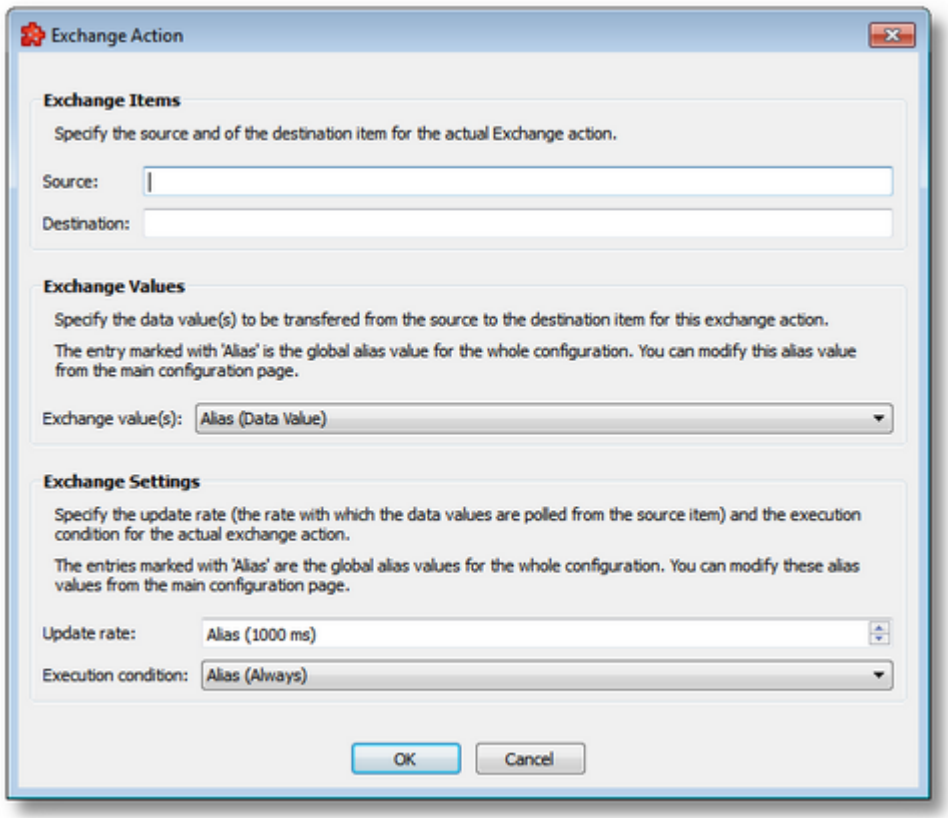
- source item name
- destination item name
- trigger type
- trigger settings
- update rate
- execution condition
- exchange value(s)

7.3.2.2.2 Exchange Action Properties

This page describes the process of adding and modifying an Exchange action.

Add a New Exchange Action

This operation can be triggered by performing various actions as described in the Configuration: Data Processing - Exchange page. The add *Exchange Action* dialog controls are described below:



Source item ID, destination item ID, update rate, execution condition and the exchange value(s) can be edited in this dialog.

The source item ID and destination item ID should be valid hierarchical names pointing to existing items in the dataFEED OPC Suite tree and must respect the access rights restrictions (the source item should be at least readable and the destination item should be at least writable). Items of the types Tag, Property or Node can be used as exchange sources or destinations as long as the access rights rule is respected.

The update rate, execution condition and the exchange value(s) initially have the values set in *Global Initial Values* which can easily be changed from the Global Alias Values and Global Initial Values dialog. More details about these attributes can be found in the Exchange Action Attributes section of the Configuration: Data Processing - Exchange page.

Note: If the source or destination item has a valid path in the dataFEED OPC Suite tree, but the item itself comes from a dynamic server connection and does not exist then it can be created from this dialog.

Note: The source and destination items of an Exchange action must be different items.

Send the source identifier for this exchange action

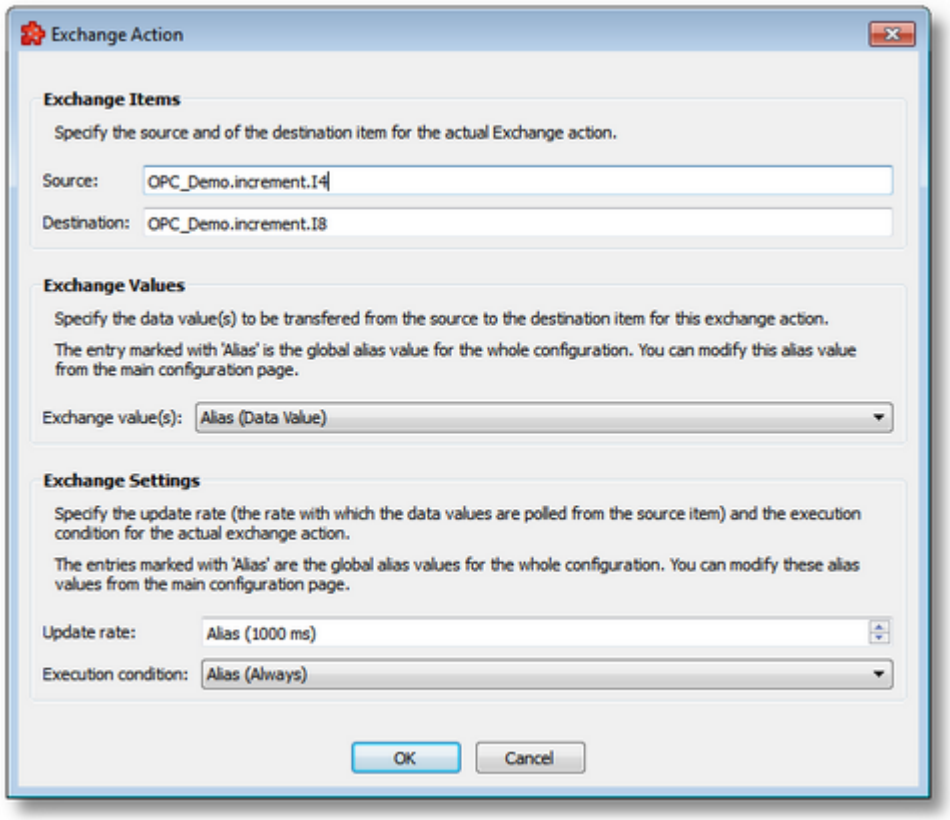
This setting is available only when the data destination is a MQTT topic with the publish format that contains information about Item Identifier.

By default, the checkbox is enabled and the identifier of the source item will be used. If the checkbox is disabled then the identifier of the destination item will be used.

Modify an Exchange Action

This operation can be triggered by performing various actions as described in the Configuration: Data Processing - Exchange page. The modify *Exchange Action* dialog controls are described in the above *Add a New Exchange Action* section. The modify Exchange Action dialog is shown below:

© Softing Industrial Automation GmbH



The dialog is populated with the attributes of the selected Exchange action.

7.3.2.2.3 Advance Exchange Action Editor

This page describes the process of using the *Advance Exchange Action Editor*.

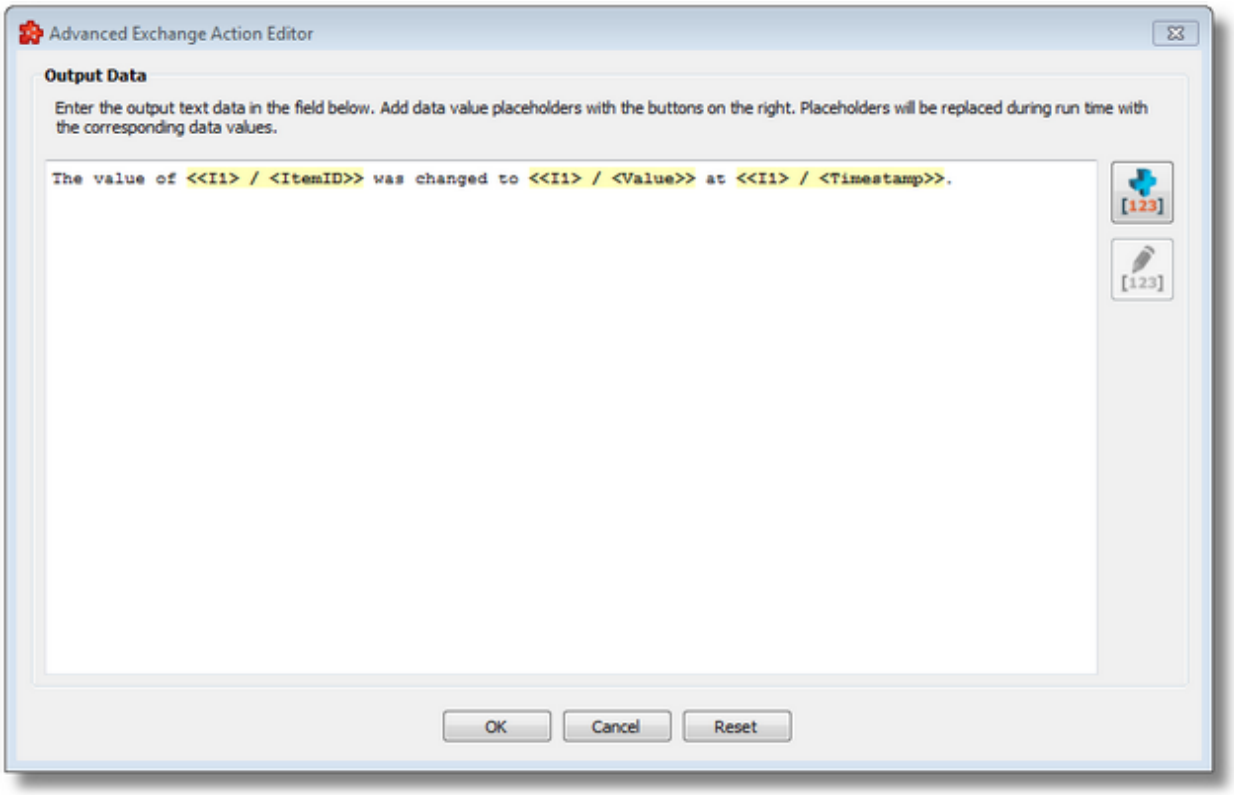
Attention: This functionality is applied only for **data destination items** with **STRING (String) data type** and **Plain Text** publish format (for MQTT destination tags).

Modify an Exchange Action

This option allow you to format the data value destination by inserting text lines and placeholders. Like File Action Editor the Advance Exchange Action Editor supports the Tab and New Line characters.

For example, the following statement writes the text to the file, replacing the placeholders accordingly:

```
TXT: The value of <<I1> / <ItemID>> was changed to <<I1> / <ItemValue>> at <<I1> / <I
XML: <value ItemID="<<I1> / <ItemID>>" ItemValue="<<I1> / <ItemValue>>" ItemTimestamp
TSV: <<I1> / <ItemID>>#%-TAB-%#<<I1> / <ItemValue>>#%-TAB-%#<<I1> / <ItemTimestampMS>
```

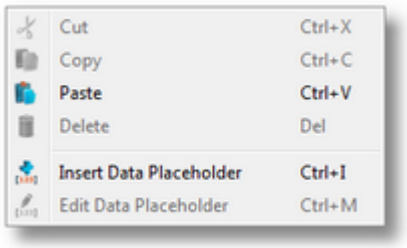


This wizard page allows defining the text for the current action using the action editor. Certain operations can be performed using the buttons on the right, the context sensitive menu or the keyboard shortcuts:

- Insert a data value placeholder using the *Insert Data Value Placeholder* button [123], pressing the keyboard shortcut: CTRL+I or selecting the *Insert Data Value Placeholder* action from the context menu.
- Modify an existing data value placeholder using the *Modify Data Value Placeholder* button [123], pressing the keyboard shortcut: CTRL+M or selecting the *Edit Data Value Placeholder* action from the context menu.
- Cut, copy, paste or delete text with or without placeholders selecting the *Cut*, *Copy*, *Paste* or *Delete* actions from the context menu or using the keyboard shortcuts associated with these actions CTRL+X, CTRL+C, CTRL+V respectively the DELETE / DEL keyboard key.

An example of the context sensitive menu is shown below:

© Softing Industrial Automation GmbH



Note: The *Cut*, *Copy* and *Paste* actions use the Windows Clipboard for data transfer thus allowing data to be transferred to/from other applications (e.g. the textual representation of the action editor content can be used in external applications and text from external application can be used in the action editor).

Attention: Please keep in mind that placeholders copied to clipboard (through the *Copy* or *Cut* commands) will only be pasted back as placeholders if the clipboard has not been tampered with, otherwise the placeholders will be pasted as plain text.

Attention: The placeholders visible in the action editor are aliases used only for displaying the placeholders. The real placeholder form is visible in the subscription file. Example: *<<R8> / <Value>>* placeholder from the action editor is serialized in the subscription file as: *#%-ItemValue-%increment.R8_%Current-%#*. Do not manually modify the subscription file by inserting placeholders in the action editor template.

To proceed forward and define the trigger type use the *Next* button, to go back to the previous wizard page use the *Back* button. Exit the wizard using the *Cancel* button.

7.3.2.2.4 File Format

The Exchange actions are stored in a single plain text configuration file called *Subscriptions.txt* which resides in the *Task* folder under the current configuration folder.

Note: The path and the name of the Exchange actions configuration cannot be configured.

The configuration file can be managed with the dataFEED OPC Suite Configurator or it can be manually edited.

Note: The recommendation is to use only the dataFEED OPC Suite Configurator to edit these files. This ensures that all the definitions will be valid.

If the Subscriptions configuration file is manually edited try loading the entire configuration in the dataFEED OPC Suite Configurator as all the configuration files are automatically validated at load-time and, if there are any mistakes, an error message will be displayed. The dialog will contain an appropriate description and will point to the first invalid line of the subscription configuration file.

The subscription file is a plain text file, encoded in UTF-8 (thus providing full UNICODE compatibility) and having a line-oriented structure. The elements of a line are tab-separated. The format of one line is the following:

```
Line type<TAB>Line data
```

Line Types

The Exchange add-on generates multiple line types:

- common lines which are specific to all the consumer add-ons: condition lines, subscription lines and item lines;
- specific lines: modifier lines and copy lines.

Common Lines

The following lines are common lines and will be used to configure different functionalities:

Line Type	Description
Subscription	Defines the update rate with which the current action will be executed.
Condition	Defines the execution condition for the current action.
Item	This line defines a trigger which is fired if the item data has changed. An item data change consists of a different value or quality of the respective item compared to the previous one.
Commented Lines (#)	Any line that starts with the "#" character is considered a comment line which is ignored during evaluation.

The subscription file is divided into two sections:

- The global area:

  The global area extends from the first subscription file line to the first trigger line (e.g: ITEM line).

  This area contains definitions valid for the entire subscription file.

- The local area:

  The local area covers all the actions between two trigger lines in the subscription file.

  The lines in this area are only executed if the parent trigger fires.

Note: In the common lines section there will always be two lines (a subscription line and a condition line) with empty IDs representing the global subscription and the global condition, respectively. The values in these lines can be accessed through the Global Alias And Initial Values dialog of the dataFEED OPC Suite Configurator.

© Softing Industrial Automation GmbH

Subscription Line

A subscription line has the following format:

```
SUBSCRIPTION<TAB>Subscription ID<TAB>UPDATERATE<TAB>Update rate value<TAB>ACTIVE<TAB>
```

The subscription line contains the following configurable parameters:

Parameter	Description
SUBSCRIPTION	The type of the line, predefined keyword.
Subscription ID	Unique subscription identifier.
UPDATERATE	Predefined keyword.
Update rate value	Defines the update rate of the current subscription.
ACTIVE	Predefined keyword.
Active value	Active value which may be *TRUE* or *FALSE*.

The subscription lines always contain the global subscription line as well as all subscriptions which are used at least once in consumer actions. The global subscription is represented by the subscription line with the empty *Subscription ID*. The default update rate value for the global subscription is *1000* milliseconds.

The value of the update rate is saved in milliseconds. This value may not be set to 0 inside the dataFEED OPC Suite Configurator but it can be manually set by editing the subscription file with an external text editor. An update rate value of 0 means *as fast as possible* but this is not recommended from a resource-related point of view.

Setting a subscription to be active or inactive is an advanced feature which cannot be configured from within the dataFEED OPC Suite Configurator. By default the active value for each and every subscription is set to *TRUE* (meaning that all subscriptions are active since creation). If a subscription's active value is set to *FALSE* then the data changes of the items configured to be part of that subscription will be ignored.

Examples:

```
SUBSCRIPTION            UPDATERATE  1000  ACTIVE      TRUE
SUBSCRIPTION      Subscription_300  UPDATERATE  300   ACTIVE      TRUE
```

Condition Line

A condition line defines whether an action below an item trigger will be executed or not (when the item trigger is fired). This can be used in other configuration lines as well to condition the execution of an item action. A condition line has the following format:

```
CONDITION<TAB>Condition ID<TAB>Condition Statement
```

The condition line contains the following configurable parameters:

Parameter	Description
CONDITION	The type of the line, predefined keyword.
Condition ID	Unique condition identifier.
Condition statement	Single condition statement

Conditions can be used to react to communication errors. Various actions may be executed depending on the quality of the item values. The following condition statements are supported:

Condition	Description
AllGoodQuality	The trigger actions are executed only if all items in the transaction have the quality *Good*.
Always	The trigger actions are executed always, ignoring the quality of the items in the incoming transaction.
ItemQualityChanged	The trigger actions are executed only if the quality of at least one item in the transaction has changed.
OneBadQuality	The trigger actions are executed if exactly one item in the transaction has the quality *Bad*.

The condition lines always contain the global condition line as well as all conditions which are used at least once in consumer actions. The global condition is represented by the condition line with the empty *Condition ID*. The default condition statement value for the global condition is *Always*. As a general rule all the condition lines need to be declared in the global section of the subscription file. If no condition is defined the Always condition is applied as default.

Examples:

```
CONDITION          Always
CONDITION    AllGoodQuality     AllGoodQuality
```

Item Line

An item line is used to add an item to a subscription. The associated actions are executed each time the value or the quality of the specified data items changes during the run time of the local dataFEED OPC Suite application.

An item line has the following format:

© Softing Industrial Automation GmbH

```
ITEM<TAB>Subscription Item ID<TAB>Server Item ID<TAB><Subscription Id>
```

The item line contains the following configurable parameters:

Parameter	Description
ITEM	The type of the line, predefined keyword.
Subscription Item ID	Unique subscription item identifier (implicitly created by joining the Server Item ID and the Subscription ID with the underscore character in between them).
Server Item ID	Item identifier within the local dataFEED OPC Suite application address namespace. This could be a server connection item ID or a local item ID.
Subscription ID	Subscription identifier (defined above, see Subscription Line).

This trigger is fired if the item data or quality has changed (also referred to as data change). The trigger data includes all other items which were included in the data change (items which had their data change reported in the same transaction).

Note: A single line consisting of the *Server Item ID* and *Subscription ID* is added to the subscription file for an item used several times with the same update rate in any consumer add-ons. This optimizes the local dataFEED OPC Suite application throughput.

Example:

```
ITEM  increment.I1_Subscription_300 increment.I1      Subscription_300
```

Note: The subscription item ID can be empty. In this case, however, only the Server Item ID can be used to identify the item.

Specific Lines

The following lines are Exchange specific:

Line type	Format
Modifier line	Defines the exchange value(s) to be used in an exchange action.
Copy line	Defines the items involved in the exchange process, denoting the source and destination.

Modifier Line

The modifier line names the exchange value(s) used in an exchange action. The modifier line has the following format:.

```
MODIFIER<TAB>Modifier ID<TAB>Modifier type
```

The modifier line contains the following configurable parameters:

Parameter	Description
MODIFIER	The type of the line, predefined keyword.
Modifier ID	Unique modifier identifier (implicitly created by cloning the type value it bears).
Modifier type	The type of the data to be exchanged. It may have any of the following values: *V*, *VQ*, *VQT*.

The possible modifier types are:

Condition	Description
Data Value (V)	This modifier denotes that only the value data attribute is exchanged.
Data Value and Quality (VQ)	This modifier denotes that the value and the quality data attributes are exchanged.
Data Value, Quality and Timestamp (VQT)	This modifier denotes that all of the data attributes (value, quality and the timestamp) are exchanged.

The modifier lines always contain the global condition and besides that contain all modifiers that are used at least once in exchange actions.

The default global modifier always has an empty *Modifier ID*. The default value for this modifier is *Value*.

Copy Line

The copy line is the essential exchange action line in the subscription file. It joins together the exchange condition, the source item along with its originating subscription, the destination item along with its source subscription and the modifier value. The copy line has the following format:.

```
COPY<TAB>Condition ID<TAB>Source Subscription Item ID<TAB>Destination Subscription It
```

The copy line contains the following configurable parameters:

© Softing Industrial Automation GmbH

Parameter	Description
COPY	The type of the line, predefined keyword.
Condition ID	Condition identifier (defined above, see Condition Line).
Source Subscription Item ID	Source subscription item identifier (defined above, see Item Line).
Destination Subscription Item ID	Destination subscription item identifier (defined above, see Item Line).
Modifier ID	Modifier identifier (defined above, see Modifier Line).

Deactivated Exchange

When the *Exchange* add-on is deactivated all the Exchange specific lines are pre-padded with the *#!* characters. This way the data remains saved in the configuration file but will be ignored by the local dataFEED OPC Suite application at run time.

Subscription File - Example:

```
CONDITION         Always
CONDITION   AllGoodQuality    AllGoodQuality
CONDITION   OneBadQuality     OneBadQuality
#
SUBSCRIPTION            UPDATERATE  1000  ACTIVE      TRUE
SUBSCRIPTION      Subscription_800  UPDATERATE  800   ACTIVE      TRUE
SUBSCRIPTION      Subscription_900  UPDATERATE  900   ACTIVE      TRUE
#
MODIFIER          V
MODIFIER    VQ    VQ
MODIFIER    VQT   VQT
#
#
ITEM  Local Items.Node1.Tag1_ Local Items.Node1.Tag1
#
ITEM  Local Items.Node1.Tag2_Subscription_900   Local Items.Node1.Tag2  Subscription_
#
ITEM  OPC_1.dynamic.myTag_Subscription_800      OPC_1.dynamic.myTag     Subscription_
#
ITEM  OPC_1.increment.BOOL_   OPC_1.increment.BOOL
COPY        OPC_1.increment.BOOL_   Local Items.Node1.Tag1_
#
ITEM  OPC_1.increment.BOOL array_Subscription_900     OPC_1.increment.BOOL array    S
COPY  AllGoodQuality    OPC_1.increment.BOOL array_Subscription_900     Local Items.N
#
ITEM  OPC_1.increment.CY_Subscription_800 OPC_1.increment.CY      Subscription_800
COPY  OneBadQuality     OPC_1.increment.CY_Subscription_800 OPC_1.dynamic.myTag_Subsc
```

7.3.2.2.5 Import Wizard

The dataFEED OPC Suite Configurator allows the import of Exchange actions from an external file. The external file should be a plain text tab-separated file with the specific Exchange File Format. The import action can be triggered by selecting the *Import* action from the *Exchange Data List* context sensitive menu. The import action is not dependent on the selection from the data list. The imported actions are always appended to the existing *Exchange Data List* list. When the import action is triggered the *Exchange Import Wizard* is started.

▾ The Exchange Import Wizard allows importing exchange actions:

- Select the file to be imported @ Import Wizard - Import File Selection
- Examine the import results @ Import Wizard - Import Results

The Exchange actions can be imported into the namespace of the Exchange add-on from a text file with a specific format. The external files should be plain text files with line-oriented structure. The elements of each line must be tab-separated. For each Copy line an Exchange action will be created.

Line Types

The Exchange add-on generates multiple line types:

- common lines which are specific to all the consumer add-ons: condition lines, subscription lines and item lines;
- specific lines: modifier lines and copy lines.

Common Lines

The following lines are common lines and will be used to configure different functionalities:

Line Type	Description
Subscription	Defines the update rate with which the current action will be executed.
Condition	Defines the execution condition for the current action.
Item	This line defines a trigger which is fired if the item data has changed. An item data change consists of a different value or quality of the respective item compared to the previous one.

© Softing Industrial Automation GmbH

Commented Lines (#)	Any line that starts with the "#" character is considered a comment line which is ignored during evaluation.

The subscription file is divided into two sections:

- The global area:

  The global area extends from the first subscription file line to the first trigger line (e.g: ITEM line).

  This area contains definitions valid for the entire subscription file.

- The local area:

  The local area covers all the actions between two trigger lines in the subscription file.

  The lines in this area are only executed if the parent trigger fires.

Note: In the common lines section there will always be two lines (a subscription line and a condition one) with empty IDs representing the global subscription respectively the global condition. The values in these lines can be accessed through the Global Alias And Initial Values dialog of the dataFEED OPC Suite Configurator.

Subscription Line

A subscription line has the following format:

```
SUBSCRIPTION<TAB>Subscription ID<TAB>UPDATERATE<TAB>Update rate value<TAB>ACTIVE<TAB>
```

The subscription line contains the following configurable parameters:

Parameter	Description
SUBSCRIPTION	The type of the line, predefined keyword.
Subscription ID	Unique subscription identifier.
UPDATERATE	Predefined keyword.
Update rate value	Defines the update rate of the current subscription.
ACTIVE	Predefined keyword.
Active value	Active value which may be *TRUE* or *FALSE*.

The subscription lines always contain the global subscription line as well as all subscriptions which are used at least once in consumer actions. The global subscription is represented by the subscription line with the empty *Subscription ID*. The default update rate value for the global subscription is *1000* milliseconds.

The value of the update rate is saved in milliseconds. This value may not be set to 0 inside the dataFEED OPC Suite Configurator but it can be manually set by editing the subscription file with

an external text editor. An update rate value of 0 means *as fast as possible* but this is not recommended from a resource-related point of view.

Setting a subscription to be active or inactive is an advanced feature which cannot be configured from within the dataFEED OPC Suite Configurator. By default the active value for each and every subscription is set to *TRUE* (meaning that all subscriptions are active since creation). If a subscription's active value is set to *FALSE* then the data changes of the items configured to be part of very subscription will be ignored.

Examples:

```
SUBSCRIPTION            UPDATERATE  1000  ACTIVE      TRUE
SUBSCRIPTION      Subscription_300  UPDATERATE  300   ACTIVE      TRUE
```

Condition Line

A condition line defines whether an action below an item trigger will be executed or not (when the item trigger is fired). This can be used in other configuration lines as well to condition the execution of an item action. A condition line has the following format:

```
CONDITION<TAB>Condition ID<TAB>Condition Statement
```

The condition line contains the following configurable parameters:

Parameter	Description
CONDITION	The type of the line, predefined keyword.
Condition ID	Unique condition identifier.
Condition statement	Single condition statement

Conditions can be used to react to communication errors. Various actions may be executed depending on the quality of the item values. The following condition statements are supported:

Condition	Description
AllGoodQuality	The trigger actions are executed only if all items in the transaction have the quality *Good*.
Always	The trigger actions are executed always, ignoring the quality of the items in the incoming transaction.
ItemQualityChanged	The trigger actions are executed only if the quality of at least one item in the transaction has changed.
OneBadQuality	The trigger actions are executed if exactly one item in the transaction has the quality *Bad*.

© Softing Industrial Automation GmbH

The condition lines always contain the global condition line as well as all conditions which are used at least once in consumer actions. The global condition is represented by the condition line with the empty *Condition ID*. The default condition statement value for the global condition is *Always*. As a general rule all the condition lines need to be declared in the global section of the subscription file. If no condition is defined the Always condition is applied as default.

Examples:

```
CONDITION          Always
CONDITION    AllGoodQuality     AllGoodQuality
```

Item Line

An item line is used to add an item to a subscription. The associated actions are executed each time the value or the quality of the specified data items changes during the run time of the local dataFEED OPC Suite application.

An item line has the following format:

```
ITEM<TAB>Subscription Item ID<TAB>Server Item ID<TAB><Subscription Id>
```

The item line contains the following configurable parameters:

Parameter	Description
ITEM	The type of the line, predefined keyword.
Subscription Item ID	Unique subscription item identifier (implicitly created by concatenating the Server Item Id and with Subscription ID with the underscore character in between them).
Server Item ID	Item identifier within the local dataFEED OPC Suite application address namespace. This could be a server connection item ID or a local item ID.
Subscription ID	Subscription identifier (defined above, see Subscription Line).

This trigger is fired if the item data or quality has changed (also referred to as data change). The trigger data includes all other items which were included in the data change (items which had their data change reported in the same transaction).

Note: A single line consisting of the *Server Item ID* and *Subscription ID* is added to the subscription file for an item used several times with the same update rate in any consumer add-ons. This optimizes the local dataFEED OPC Suite application throughput.

Example:

```
ITEM  increment.I1_Subscription_300 increment.I1      Subscription_300
```

Note: The subscription item ID can be empty. In this case, however, only the Server Item ID can be used to identify the item.

Specific Lines

The following lines are Exchange specific:

Line type	Format
Modifier line	Defines the exchange value(s) to be used in an exchange action.
Copy line	Defines the items involved in the exchange process, denoting the source and destination.

Modifier Line

The modifier line names the exchange value(s) used in an exchange action. The modifier line has the following format:.

```
MODIFIER<TAB>Modifier ID<TAB>Modifier type
```

The modifier line contains the following configurable parameters:

Parameter	Description
MODIFIER	The type of the line, predefined keyword.
Modifier ID	Unique modifier identifier (implicitly created by cloning the type value it bears).
Modifier type	The type of the data to be exchanged. It may have any of the following values: *V*, *VQ*, *VQT*.

The possible modifier types are:

Condition	Description
Data Value (V)	This modifier denotes that only the value data attribute is exchanged.
Data Value and Quality (VQ)	This modifier denotes that the value and the quality data attributes are exchanged.
Data Value, Quality and Timestamp (VQT)	This modifier denotes that all of the data attributes (value, quality and the timestamp) are exchanged.

© Softing Industrial Automation GmbH

The modifier lines always contain the global condition and besides that contain all modifiers that are used at least once in exchange actions.

The default global modifier always has an empty *Modifier ID*. The default value for this modifier is *Value*.

Copy Line

The copy line is the essential exchange action line in the subscription file. It joins together the exchange condition, the source item along with its originating subscription, the destination item along with its source subscription and the modifier value. The copy line has the following format:.

```
COPY<TAB>Condition ID<TAB>Source Subscription Item ID<TAB>Destination Subscription It
```

The copy line contains the following configurable parameters:

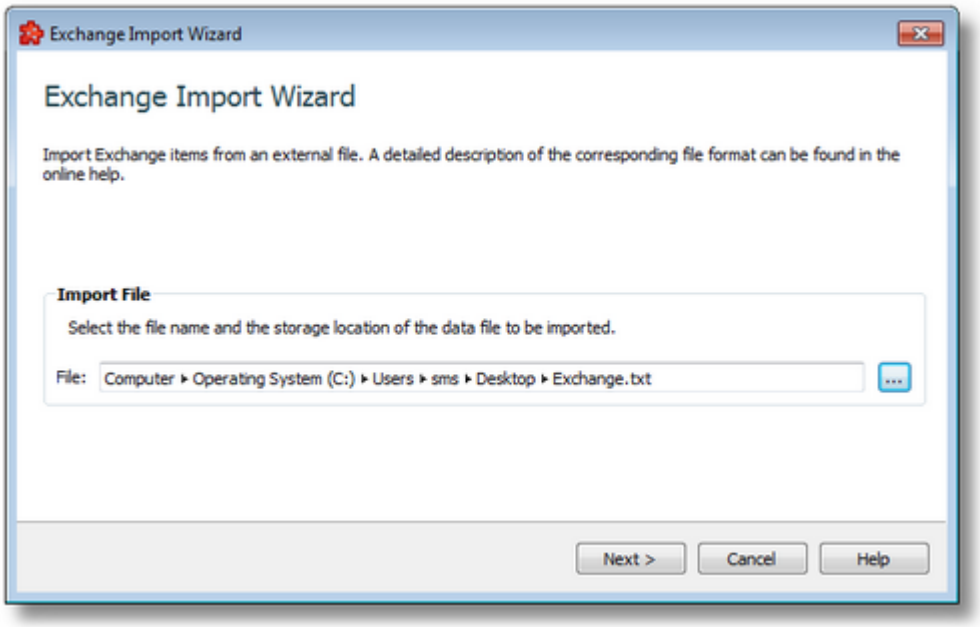
Parameter	Description
COPY	The type of the line, predefined keyword.
Condition ID	Condition identifier (defined above, see Condition Line).
Source Subscription Item ID	Source subscription item identifier (defined above, see Item Line).
Destination Subscription Item ID	Destination subscription item identifier (defined above, see Item Line).
Modifier ID	Modifier identifier (defined above, see Modifier Line).

Subscription Import File - Example:

```
CONDITION         Always
CONDITION   AllGoodQuality    AllGoodQuality
CONDITION   OneBadQuality     OneBadQuality
#
SUBSCRIPTION            UPDATERATE  1000  ACTIVE      TRUE
SUBSCRIPTION      Subscription_800  UPDATERATE  800   ACTIVE      TRUE
SUBSCRIPTION      Subscription_900  UPDATERATE  900   ACTIVE      TRUE
#
MODIFIER          V
MODIFIER    VQ    VQ
MODIFIER    VQT   VQT
#
#
ITEM  Local Items.Node1.Tag1_ Local Items.Node1.Tag1
#
ITEM  Local Items.Node1.Tag2_Subscription_900   Local Items.Node1.Tag2  Subscription_
#
```

```
ITEM  OPC_1.dynamic.myTag_Subscription_800      OPC_1.dynamic.myTag     Subscription_
#
ITEM  OPC_1.increment.BOOL_   OPC_1.increment.BOOL
COPY        OPC_1.increment.BOOL_   Local Items.Node1.Tag1_
#
ITEM  OPC_1.increment.BOOL array_Subscription_900     OPC_1.increment.BOOL array    S
COPY  AllGoodQuality    OPC_1.increment.BOOL array_Subscription_900     Local Items.N
#
ITEM  OPC_1.increment.CY_Subscription_800 OPC_1.increment.CY      Subscription_800
COPY  OneBadQuality     OPC_1.increment.CY_Subscription_800 OPC_1.dynamic.myTag_Subsc
```

The Import File Selection wizard page allows you to choose a file to be imported. The file can be selected by manually typing its name in the *File* input box or by clicking the browse button located to the right of the input box (a standard file opening dialog will open).

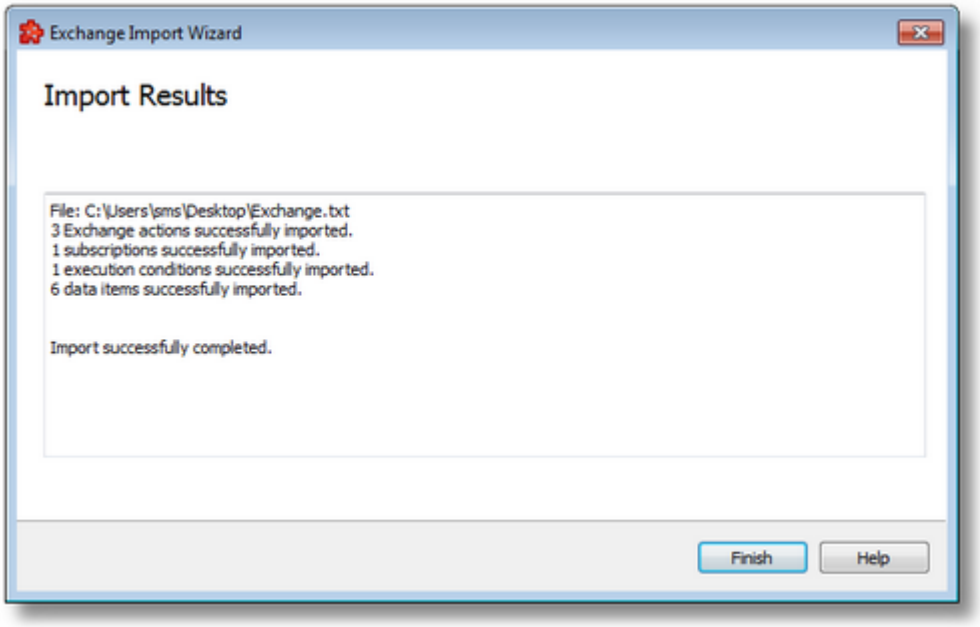


The file path and name should be valid (they should point to an existing file). When the *Next* button is pressed the file path and name are verified. If the file identified by the specified file path and name does not exist an error message is displayed. The actual import begins if the selected file is valid. If the import takes longer to complete a waiting dialog is displayed. When the import operation is finished the *Import Results* page is displayed.

Import Wizard - Import Results >>>

The Import Results wizard page displays the import results.

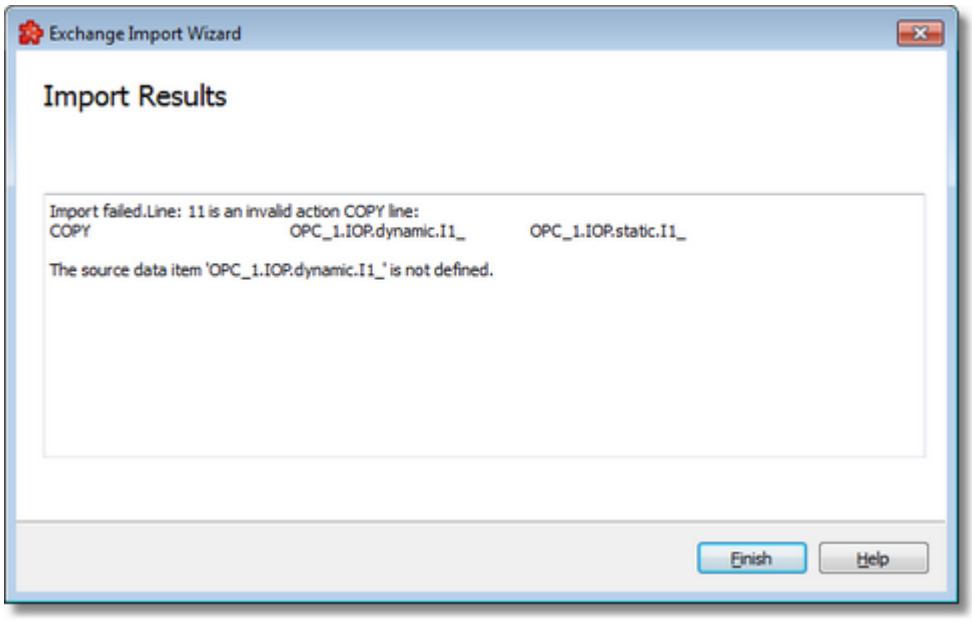
© Softing Industrial Automation GmbH



This page is the final page of the *Exchange Import Wizard*. A back button is not provided as the import action cannot be undone. The import can be successfully or it can fail.

1. After a successful import the results page contains the following information:

   - The absolute path of the imported file.
   - The number of imported Exchange actions.
   - The number of imported subscriptions.
   - The number of imported execution conditions.
   - The number of imported data items.
   - The confirmation message that the import was successful.



2. After a failed import the results page contains the following information:

   - The invalid line where the import failed.
   - The reason why the line is considered invalid.
   - The confirmation message that the import failed.

When the import wizard is closed after a successful import, the new items are appended to the Exchange data list.

<<< Import Wizard - Import File Selection

7.3.2.2.6 Export Wizard

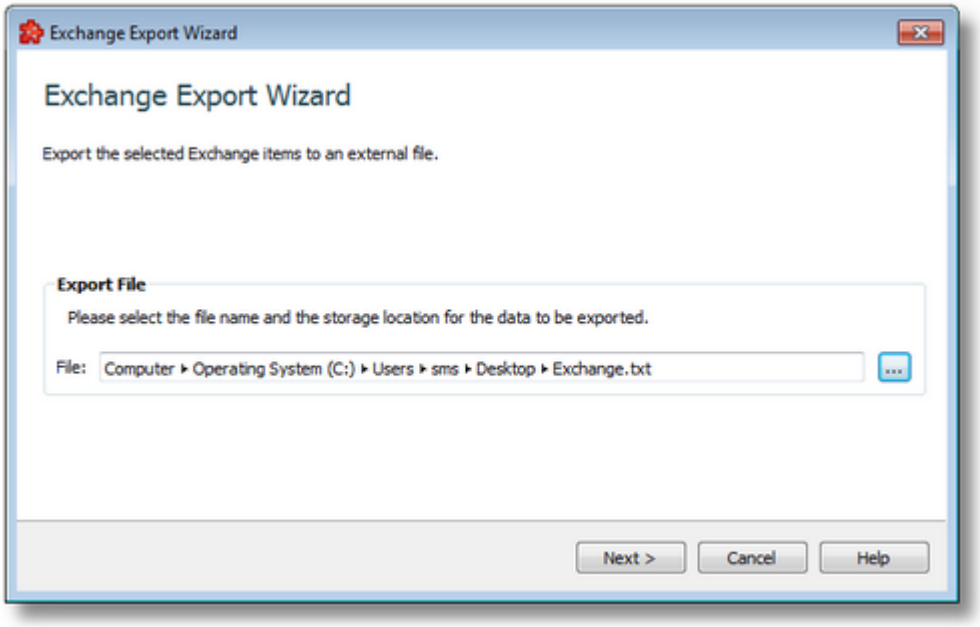
The dataFEED OPC Suite Configurator allows exporting Exchange actions to an external file. The external file will be a tab-separated file with the specific Exchange File Format. The export action can be triggered by selecting the *Export* entry from the *Exchange Data List* context sensitive menu. The exported file will contain all the selected Exchange actions, with all the execution conditions, subscriptions, modifiers and items used in them. When the export action is triggered the *Exchange Export Wizard* is started.

▾ The Exchange Export Wizard allows exporting Exchange actions:

- Select the file the Exchange actions to be exported to @ Export Wizard - Export File Selection
- Examine the export results @ Export Wizard - Export Results

The Export File Selection wizard page allows you to choose a file path and name for the exported file. The file can be selected by manually typing its path and name in the *File* input box or by clicking the browse button located to the right of the input box (a standard file saving dialog will open).

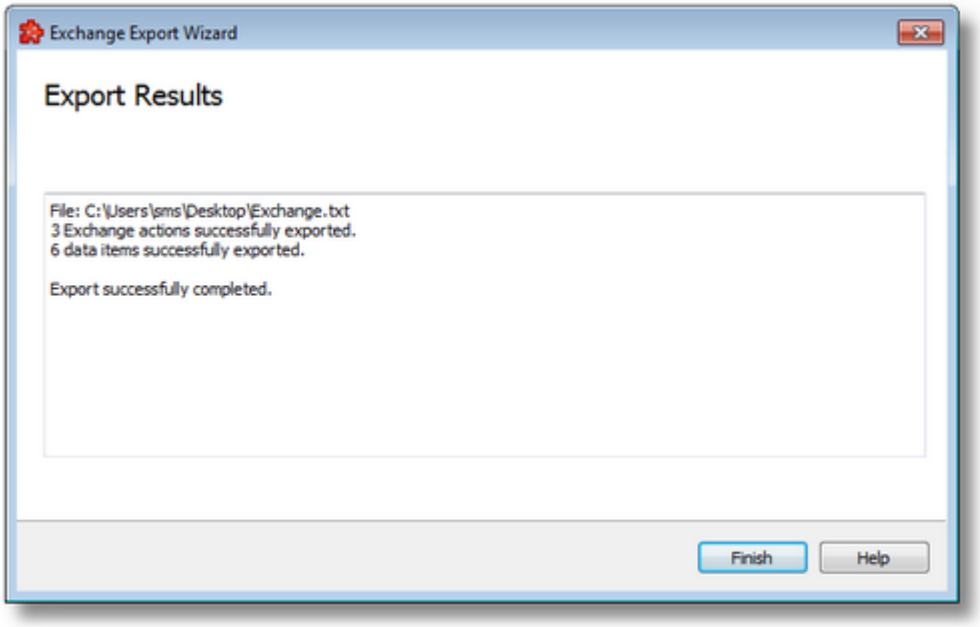
© Softing Industrial Automation GmbH



When the *Next* button is pressed the file path is verified and if the path does not exist, an error message is displayed (invalid path). If the specified file path is valid and a file with the specified name already exists, a message box will ask whether to overwrite the existing file or not (select another file). The actual filter export operation begins if the path to the file is valid and either a file with the specified name does not exist or it exists and overwrite has been chosen. When the export operation is finished the *Export Results* page is displayed.

Export Wizard - Export Results >>>

The Export Results wizard page displays the export results.



This page is the final page of the *Exchange Export Wizard*. A back button is not provided as the export action cannot be undone. The export can be successful or it can fail:

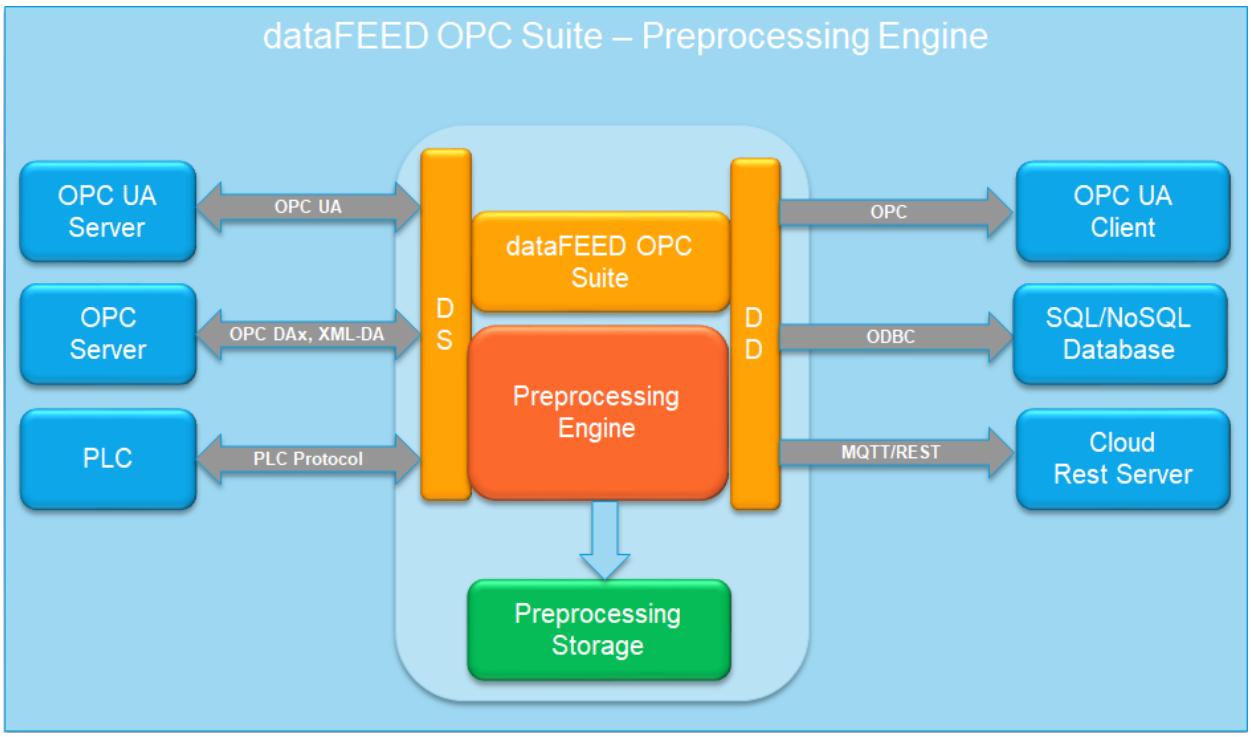
1. After a successful export the results page contains the following information:
   - The absolute path of the exported file.
   - The number of exported Exchange actions.
   - The number of exported Exchange data items.
   - The confirmation message that the export was successful.
2. After a failed export the results page contains the following information:
   - The reason why the export failed.
   - The confirmation message that the export failed.

<<< Export Wizard - Export File Selection

7.3.2.3 Preprocessing Engine

This section of the Softing dataFEED OPC Suite Online Help describes how to configure the Preprocessing Engine functionality within Softing dataFEED OPC Suite.

© Softing Industrial Automation GmbH

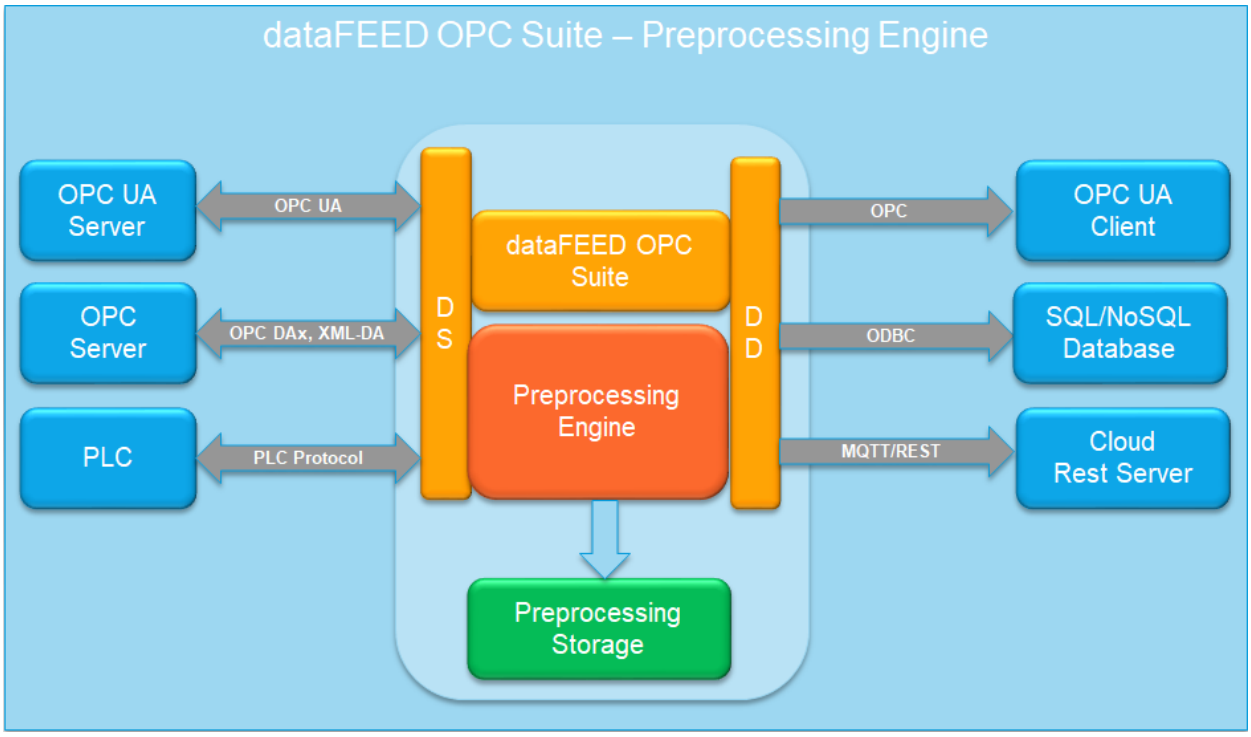


The Preprocessing Engine functionality is dynamically collecting data-values from all available data sources based on an internal subscription mechanism, applying a set of processing operations (based on mathematical and/or logical functions), data conversions and delivering the results back to a data source or to all available data destinations.

The processed data values can be directly written a data source or exposed on a Preprocessing Storage (based on Local Items) which later can be accessed as data destination by a connected OPC Client or by a delivery processor (ODBC, MQTT or REST) protocol.

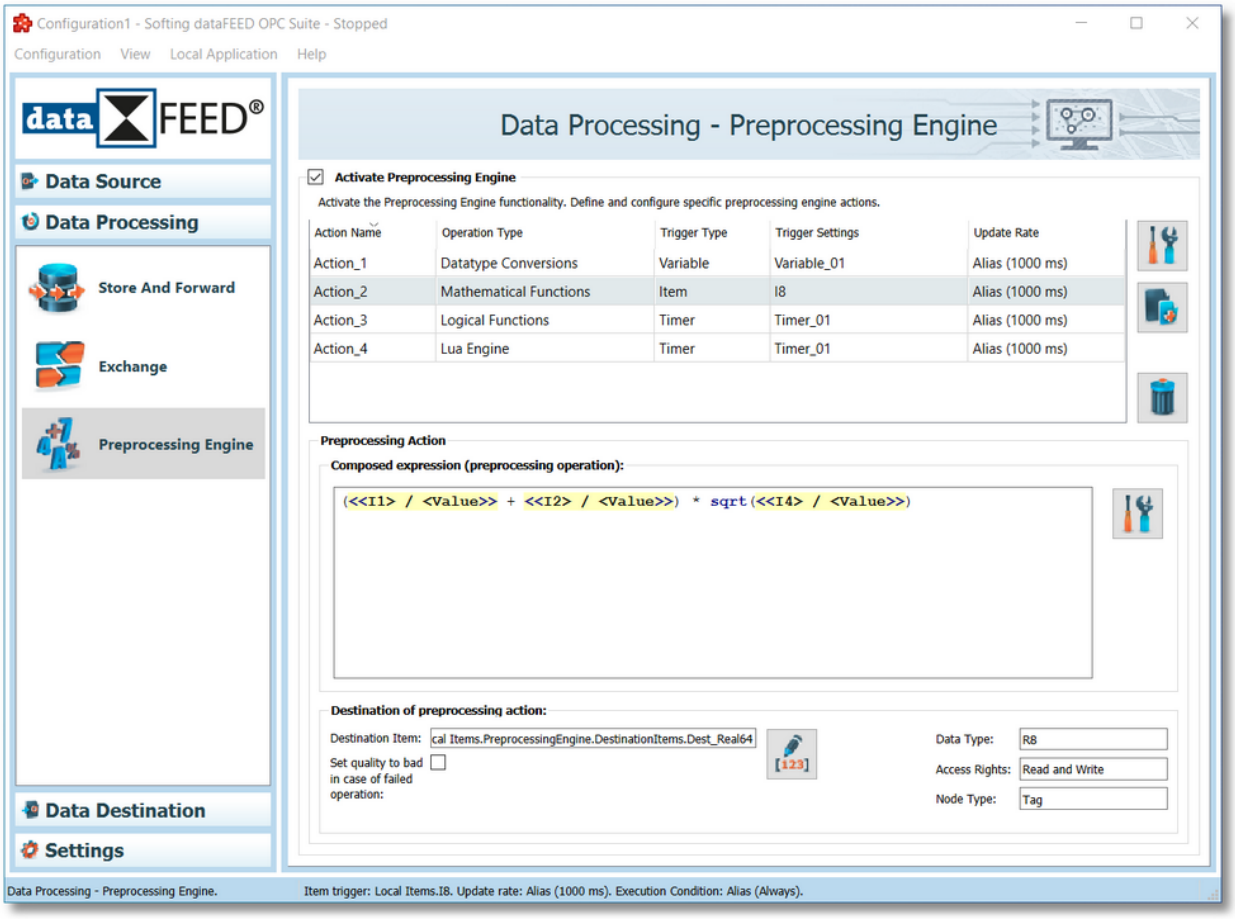
© Softing Industrial Automation GmbH

7.3.2.3.1 Configuration



The Preprocessing Engine main user interface allows to configure the preprocessing actions. The main user interface can be reached by selecting the Preprocessing Engine option from Data Preprocessing section on the navigator panel on the left side of dataFEED OPC Suite configuration tool.

© Softing Industrial Automation GmbH



The entire functionality can be enabled or disabled directly from the main page by using the **Activate Preprocessing Engine** check box. Disabling the functionality will completely deactivate the Preprocessing Engine, all the resources used at runtime will be not loaded at the initialization time; also all actions will be deactivated and not loaded at the initialization time.

Note: Disabling the functionality will not delete or modify the configured actions.

Enabling the Preprocessing Engine will activate the loading of specific module (including the preprocessing library - LUA) at the initialization time.

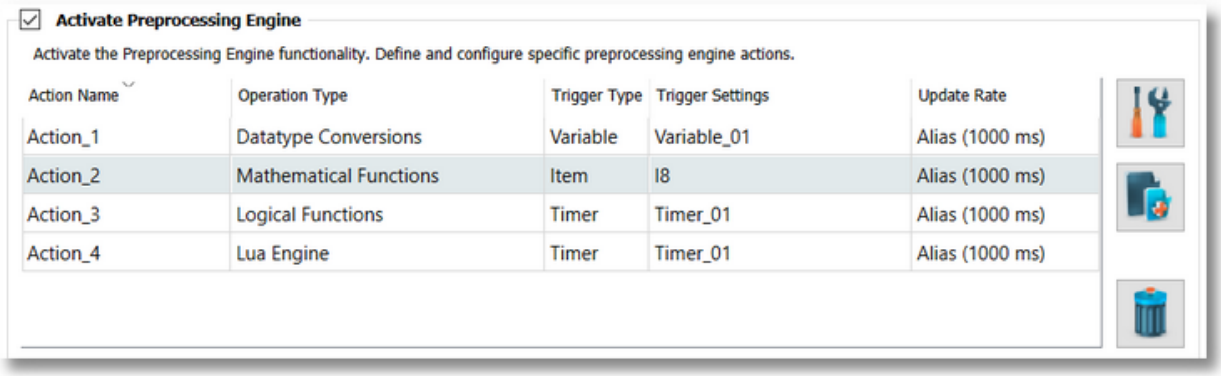
Preprocessing Engine - Main page components:

The main Preprocessing Engine user interface page contains the following components:

- Actions Table - containing basic information about all the defined actions. The following information being available:
  - Action Name: uniquely identifying a defined action. This information is needed specially for debug purposes or a later analysis of log files.
  - Operation Type: specifying the type of the defined operations on defined action. The supported operation types are: Mathematical Operations and Logical Operations. Based

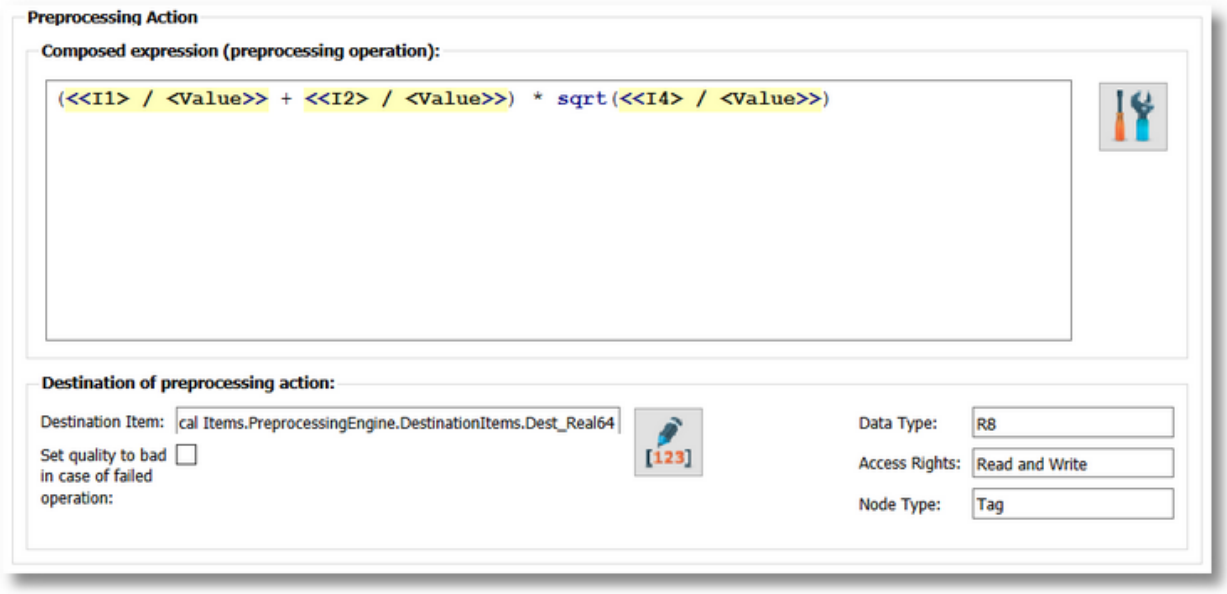
on configured operation type, a specific set of allowed operations on defined action is activated.

- Trigger Type: specifying the type of the trigger which required the execution of specified action. The supported trigger types are: Item, Timer and Variable Trigger.
- Trigger Settings: specifying a set of basic details about the configured trigger.
- Update Rate: specifying the rate of collecting data on items used as placeholders inside the defined action.



- Preprocessing Action Overview - containing basic information about a selected action. The following information being available:
  - *Composed Expression*: basic preview of configured action is available - making the identification of action possible. The preview is read only, but the composed expression of selected action can be modified by pressing the  button (it will be automatically redirected to the Expression Composer wizard page).
  - Destination of preprocessing action: specify the destination item where the result of the evaluated expression will be written. It is also specifying a set of basic information about the data destination item: Data Type, Access Rights and Node Type. The destination node id is not editable, but it can be edited by pressing the  button (it will be automatically redirected to the Destination Item wizard page).

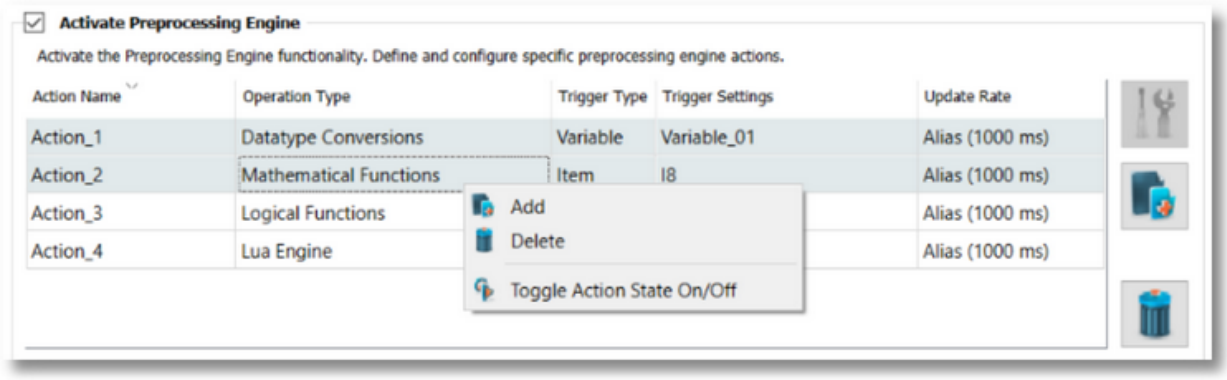
© Softing Industrial Automation GmbH



Actions Table:

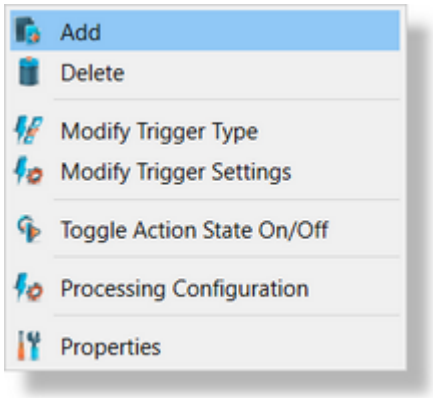
Allows to define new actions or to modify the major properties/settings of an already defined action.

The table supports multiple selection operation which will enable the common operation on multiple actions selection: Delete and Toggle Action State On/Off.



Each operation is applied to all selected actions. Basically deletion or activation/deactivation of multiple actions on a single operation is possible.

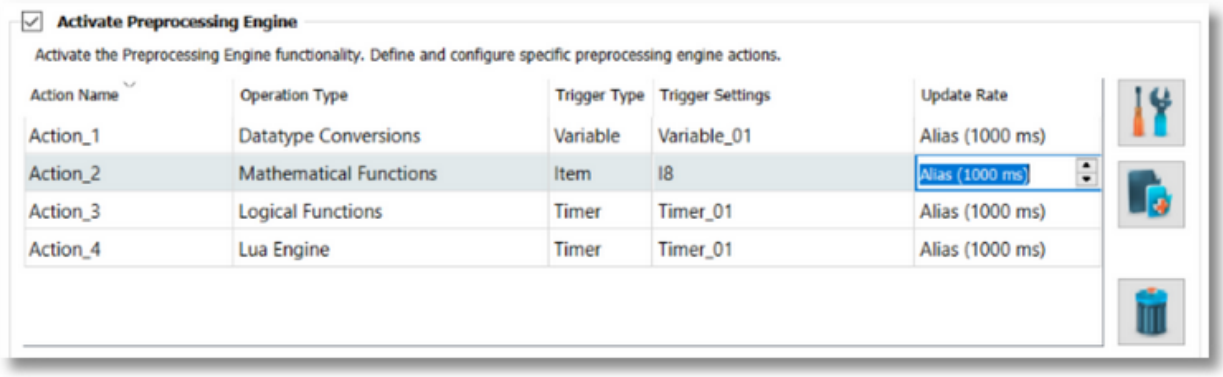
It is allowed to add or modify a preprocessing engine action by pressing the  or  buttons from right side of actions table or by using the context menu (modifying an action being possible only when a single action is selected).



Other available action for each selected action are: *Modify Trigger Type* or Settings, *Activation/Deactivation* of current action and *Processing Configuration* which open directly the Expression Composer wizard page for editing the action body.

The Actions Table supports also a direct shortcut for modifying the trigger type or trigger settings by double clicking on proper cell of selected item (this action is opening directly the proper trigger definition wizard page).

The update-rate used for internal subscription(s) which deserves the collecting data for all the placeholders used on a selected action can be also modified directly from the action table - this action is possible by double clicking on proper cell and then modifying the value of used update-rate.

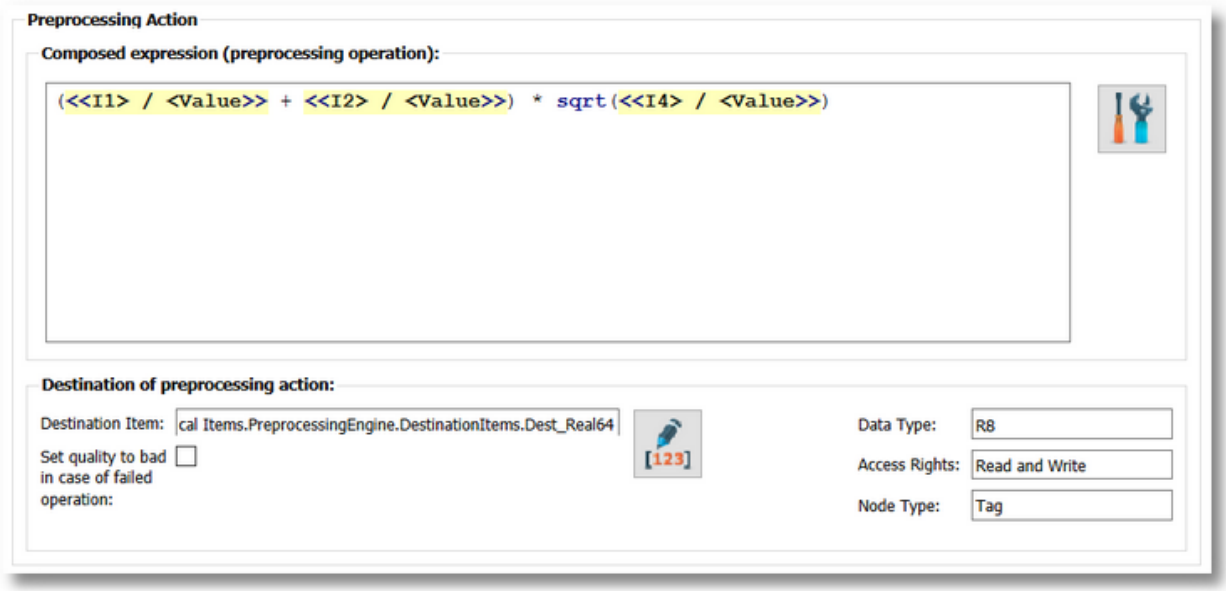


Deactivation of Preprocessing Engine functionality is automatically deactivating (at runtime - without changing the action state visible on configuration tool) all the defined preprocessing actions.

Preprocessing Action Overview:

The Preprocessing Action Overview is providing a set of information related to action body (composed expression) and destination item for a selected action. Basically, for a selected action, it can be visualized the main components of a preprocessing action.

© Softing Industrial Automation GmbH



It is composed from 2 individual components:

- *Composed Expression*: read only text editor - which provides information about the expression (action body) - specifying the formula used on preprocessing action (containing all the operations and operands (including the placeholders)).

Note: The text editor is read only, the composed expression is possible being modified by pressing the button which will directly open the Expression Composer wizard page which allow the modification of used formula (operations and operands including placeholders). Using this approach, the expression can be individually modified without following all the steps from action definition wizard.

- *Destination of Preprocessing Action:* read only text boxed - which provides information about the destination item:
  - Destination Item ID (the internal Item ID including the connection name)
  - Destination Item basic attribute: data type, access rights and node type.
  - Destination Item supports Bad Quality: if unchecked than Bad quality values are not written not written into destination item

Note: The Destination Item text editor is read only (only Copy operation being available), the destination item is possible being changed by pressing the button which will directly open Destination Definition wizard page which allows the changing of used destination item. Using it, the destination can be individually modified without following all the steps from action definition wizard.

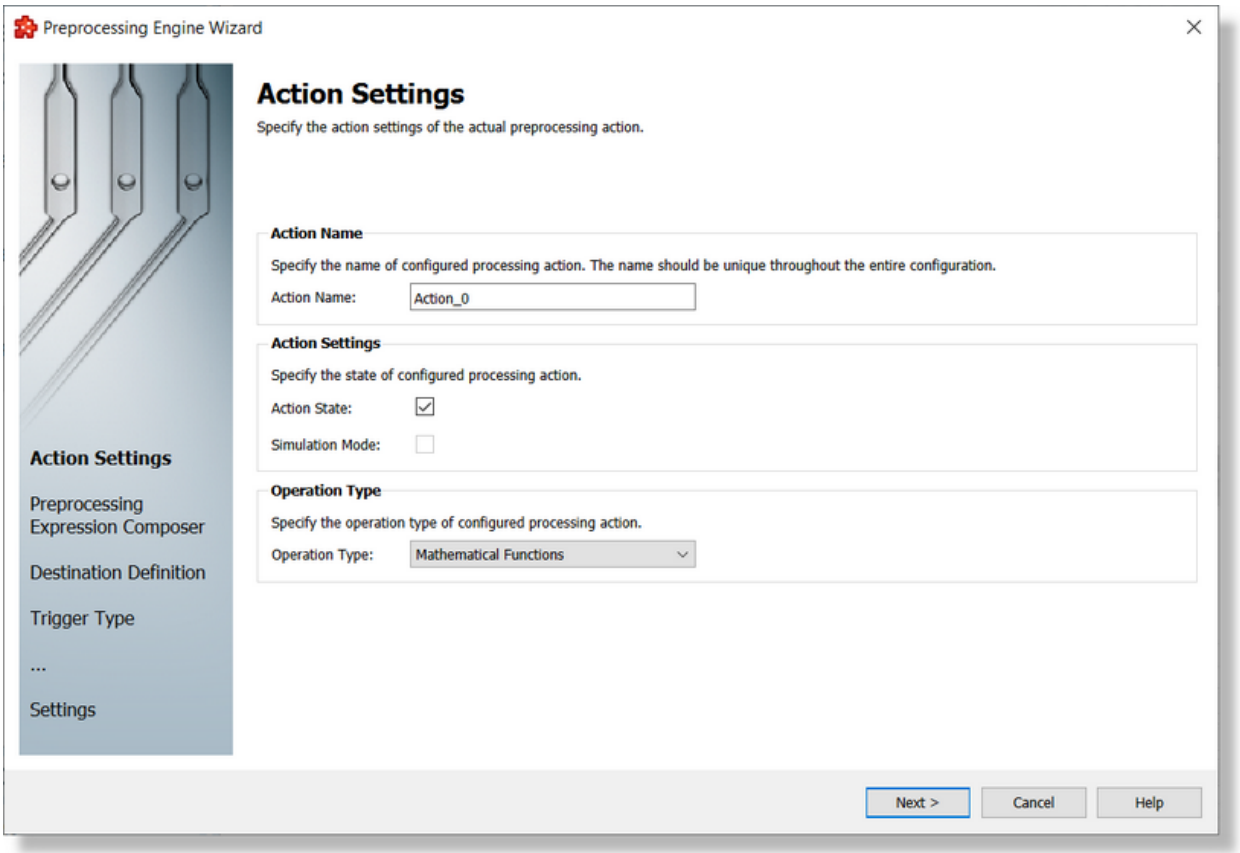
7.3.2.3.2 Preprocessing Action Wizard

▼ With the Preprocessing Engine Action Wizard you can create a new Preprocessing Action definition or modify an existing one:

- Specify the basic settings of preprocessing action @ Preprocession Engine Action Wizard - Action Settings
- Specify the expression formula containing the processing action @ Preprocessing Engine Action Wizard - Expression Composer
- Select the destination item @ Preprocessing Engine Action Wizard - Destination Definition
- Choose the trigger type @ Preprocessing Engine Action Wizard - Trigger Type
- Configure the execution condition and the update rate @ Preprocessing Engine Action Wizard - Action Settings

Note: A wizard flow will not include all the pages listed above. Depending on the selections performed in some of the pages, different flow paths will be followed.

On the **Action Settings** wizard page it is allowed to specify the following action settings:



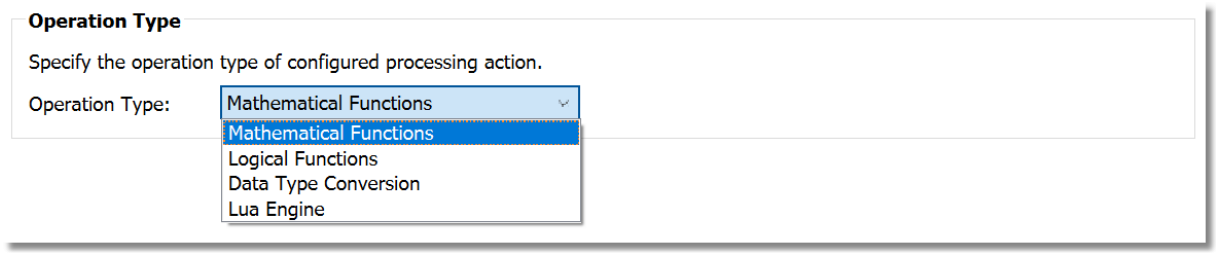
- **Action Name**: the action name is a unique action identifier, which uniquely identify a configured action - this identifier is useful specially on debug activities or analysis of log

© Softing Industrial Automation GmbH

files. Basically, it can be used to identify an action and all its paths visible on logs while actions are executed.

Note: Action Name is automatically generated in order to use unique action names. It can be modified to a configured value which should be uniquely used inside the Preprocessing Engine functionality.

- **Action State**: an action can be individually activated or deactivated. All the deactivated/disabled actions are ignored by the dataFEED OPC Suite runtime - being ignored at the initialization time - these actions can be configured, are accessible by the configuration tool but ignored by the application.
- **Operation Type**: the supported operation types on preprocessing engine actions are:
  - ***Mathematical Operations*** - allows usage of mathematical operations on defining an action body. Together with basic mathematical operators, it is possible to use the value of an item from the available application address space or a constant as operands.
  - ***Logical Operations*** - allows usage of logical operations on defining an action body. The value of an item from the available application address space or a constant can be used as operands.
  - ***Data Type Conversion Operations*** - allows usage of data type conversion operations on defining an action body. It allows to convert item from the available application address space to another item with different type.
  - ***Lua Engine Operations*** - allows usage of any Lua script as action body. The Lua script is evaluated and items placeholders from address space can be used in the expression.



Attention: the operation type cannot be modified for an already configured action. It is allowed being configured only for newly created actions - where the action body (composed expression) and destination item was not already configured.

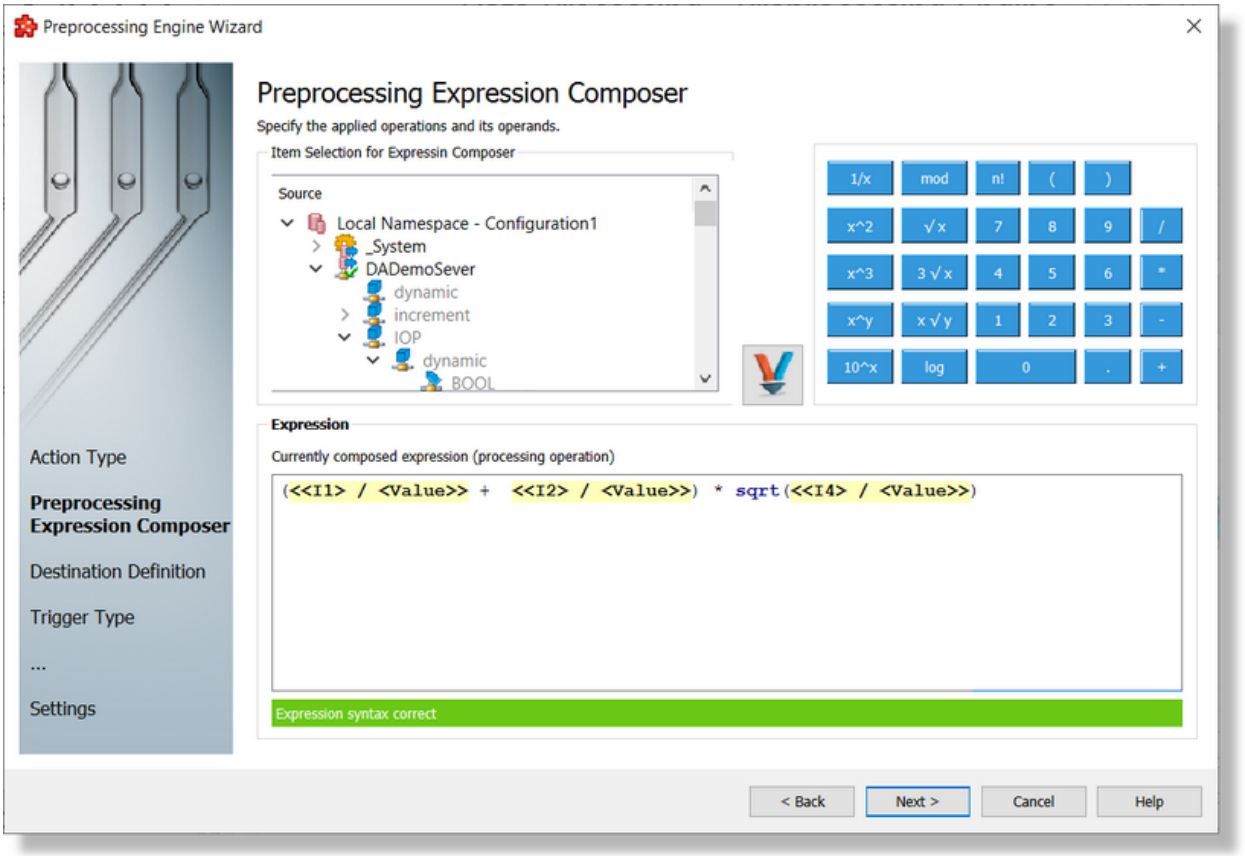
Basically, based on selected operation type, a set of specific operation are allowed being used for defining a preprocessing action.

Preprocessing Engine Action Wizard - Expression Composer >>>

On the Expression Composer wizard page - it can be specified the action body (composed expression) of the current action. Based on Operations Type (configured in previous wizard page) a set of operations and operands are available.

Mathematical Operations

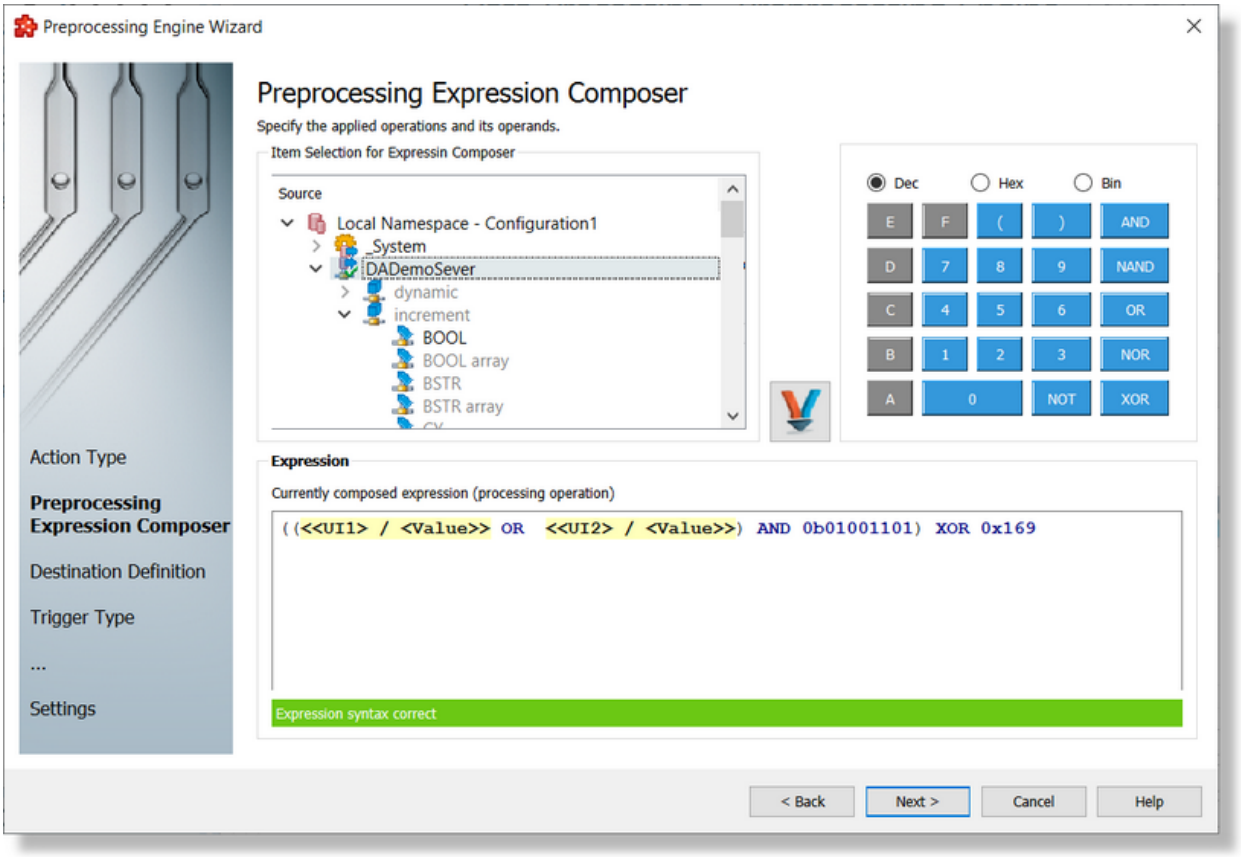
Mathematical Operations wizard page allows defining expressions containing a set of basic mathematical operations and its corresponding operands.



Logical Operations

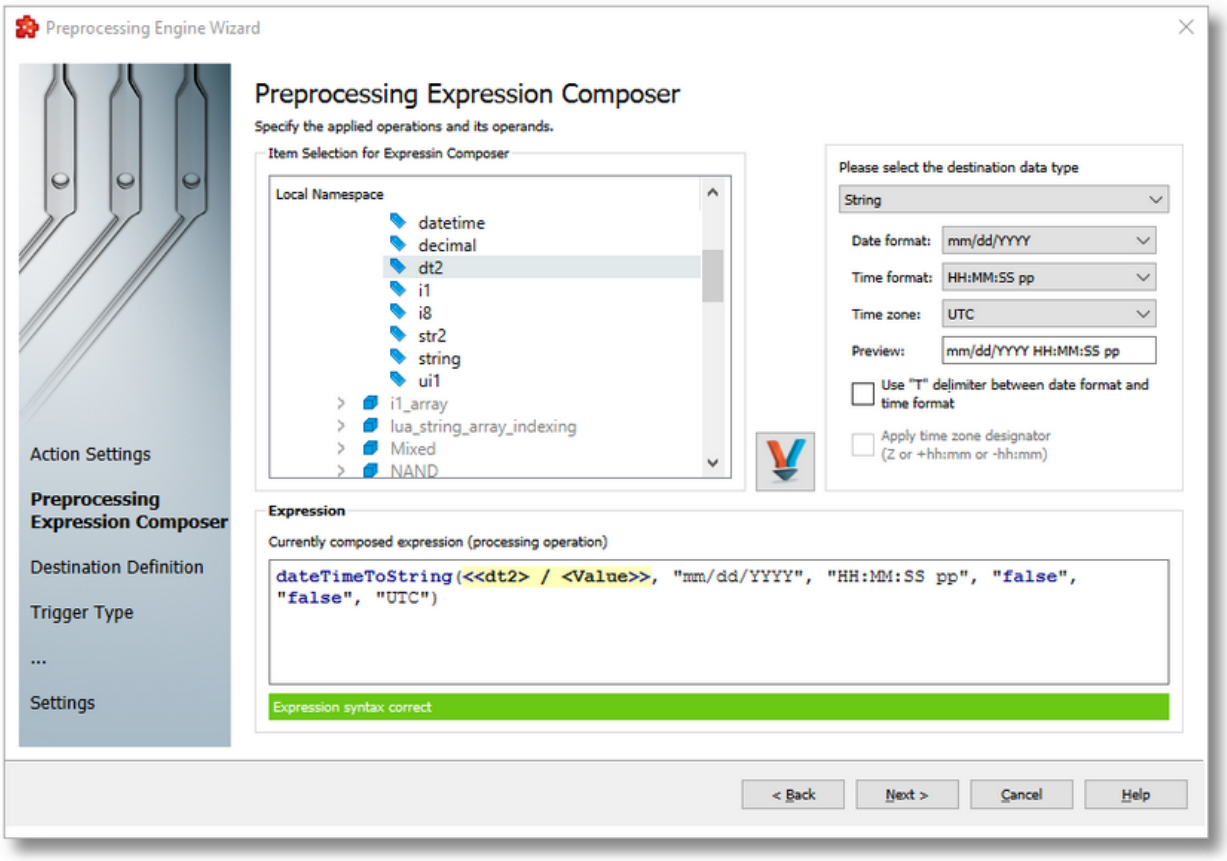
Logical Operations wizard page allows defining expressions containing a set of logical operations and its corresponding operands.

© Softing Industrial Automation GmbH



Data Type Conversions

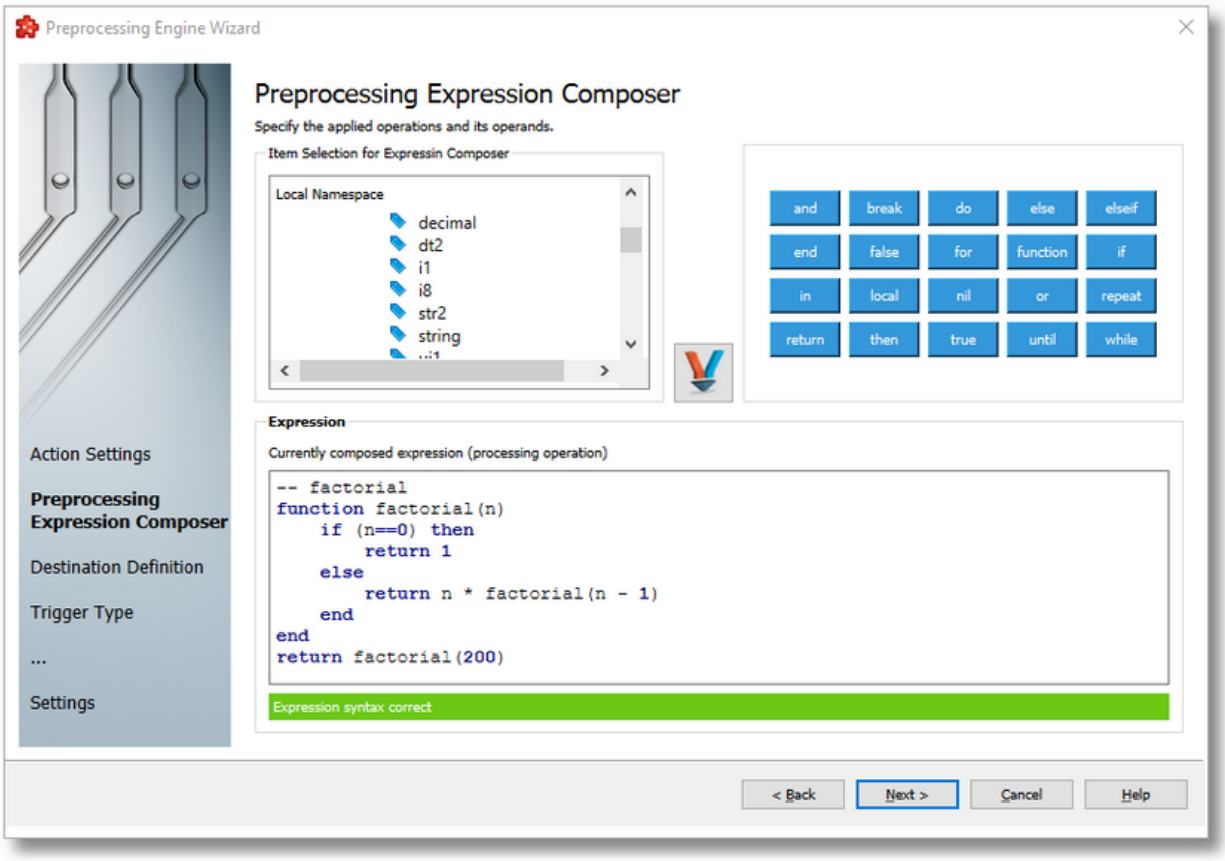
Data Type Conversions wizard page allows conversion between different data type items.



LUA Engine

LUA Engine Operations allows defining expressions containing almost any LUA script which after evaluation is returning a value.

© Softing Industrial Automation GmbH

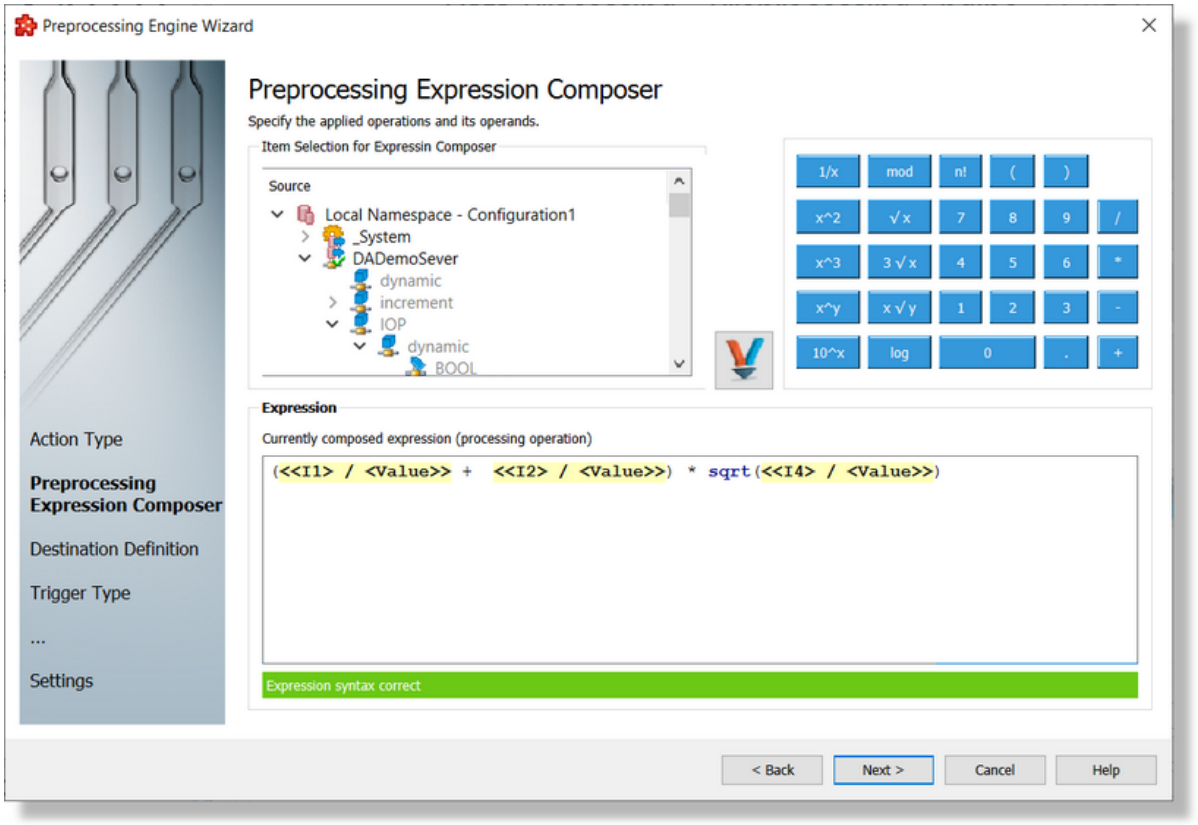


Note: The available operations and its corresponding operands are dependent about the operation type or current action. Also, the available items from the address space is filtered based the same settings.

Note: Detailed description of Expression Composer wizard page based on operation type is available as children of current topic.

<<< Preprocessing Engine Action Wizard - Action Type Preprocessing Engine Action Wizard - Destination Definition >>>

Mathematical Operation allows defining composed expressions containing a set of basic mathematical operations and its corresponding operands. Usage of multiple operations, grouping of multiple operations or usage of operations on different layers are also allowed.



Available mathematical operations:

- sum (+), difference (-), multiplication (*) and division (/) - this operations can be applied to a set of at least 2 operands.
- 1 divided by x (1/x) - this operation allows division of 1 to an operand.
- x at second power (x^2) - this operation allows calculation of the value of operand multiplied with the same operand.
- x at third power (x^3) - this operation allows calculation of the value of operand multiplied with the same operand for 3 times.
- x at y power (x^y) - this operation allows calculation of the value of operand multiplied with the same operand for a predefined number of times.
- 10 at x power (10^x) - this operation allows calculation of 10 multiplied with 10 for a predefined number of times.
- modulo (mod(x,y)) - this operation calculates for 2 given positive numbers: x and y, the remainder of the euclidean division, where x is a dividend and y is a divisor.
- square root (sqrt(x)) - this operation allows calculation of square root of a specified operand. Given a number x, the cube root of x is a number a such that $a^2 = x$.
- cube root (cuberoot(x)) - this operation allows calculation of square root of a specified operand. Given a number x, the cube root of x is a number a such that $a^3 = x$.
- nroot (nroot(x,y)) - this operation allows calculation of square root of a specified operand. Given a number x, the cube root of x is a number a such that $a^y = x$.

© Softing Industrial Automation GmbH

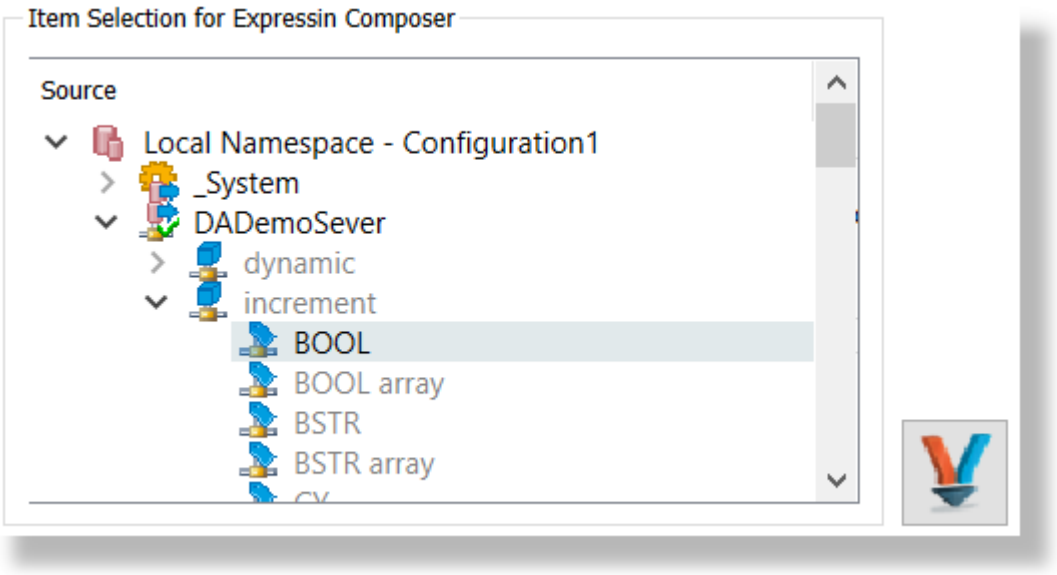
- logarithm (log(x)) - this operation allows calculation of logarithm of a specified operand.
- factorial (fact(x)) - this operation allows calculation of factorial of a positive integer, which is the product of all positive integers less than or equal with its value.

Available mathematical operands:

- variables: the value of a specified item available into the exposed address space - basically it can be selected an item from the available address space (this item will be part of an internal subscription (see Action Settings wizard page) which will update the internal cache of selected item - later on action execution it will be used the cached value).
- constants: the value should be specified by the user.

The Expression Composer wizard page contains the following components:

Address Space Tree View: this allows the selection of an item from the entire available address space. Browse operations are allowed for all the defined connections and for all available levels.



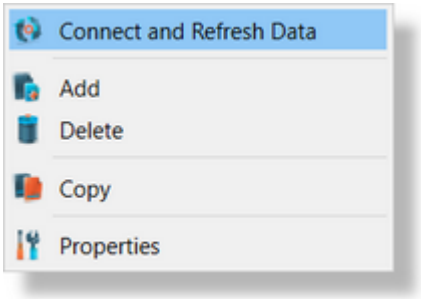
Also it is allowed to modify the address space by right clicking into the address space tree. The available operations being:

- Connect and Refresh Data - tries to re-establish the connection to the data source and re-browse it (this operation is needed specially when the data source changes its address space)
- Add/Delete - allows to add or delete an item from the available address space.

Attention: Adding a node requires to know the syntax used by the corresponding data source. Different data source types could require different syntax.

Attention: Deleting a node will deleted the node only from the exposed address space by the configuration tool. This operation will be visible from all the data destination and data processing plug-ins which has access to the address space.

- Copy/Paste (when clipboard is not empty) - allows to copy/paste a node or an item into the address space.
- Properties - allows to visualize or modify the properties of a selected item.



It is allowed to navigate into the address space and to select a variable. Once the selection is done - it can be decided to be used as operand into the expression composer for a defined operation by pressing the  button. Immediately, a corresponding placeholder is introduced into the Expression Composer Text Editor on current cursor position.

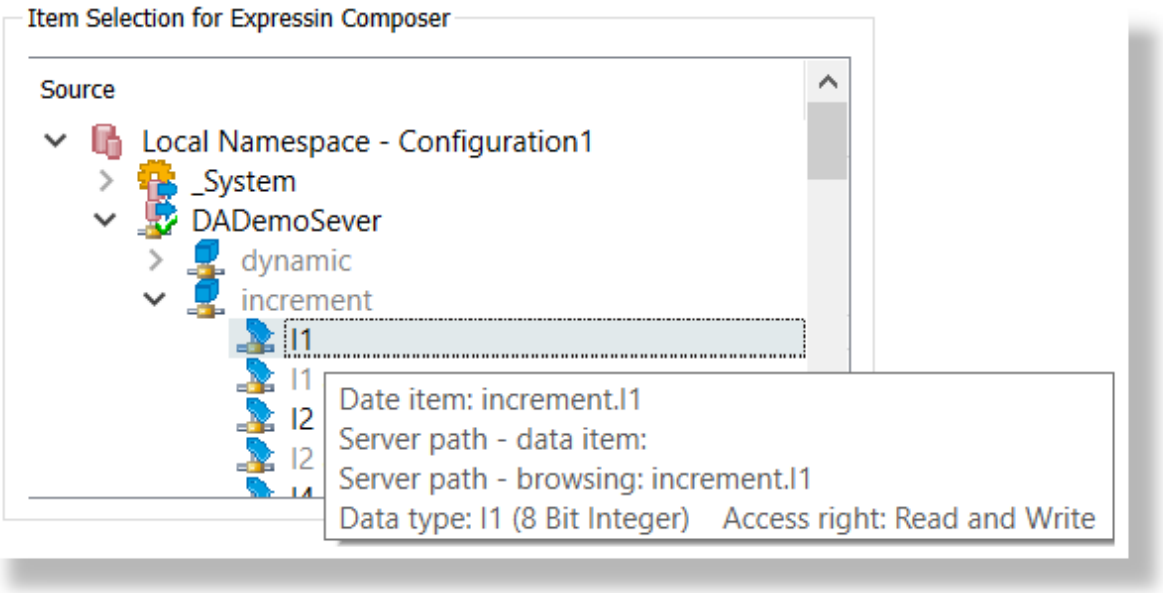
The access to the address space is limited to the item using the following data types:

- I1 (8 bit integer), I2 (16 bit integer), I4 (32 bit integer), I8 (64 bit integer),
- UI1 (8 bit unsigned integer), UI2 (16 bit unsigned integer), UI4 (32 bit unsigned integer), UI8 (64 bit unsigned integer),
- R4 (32 bit float), R8 (64 bit double)
- Decimal
- Variant (Any).

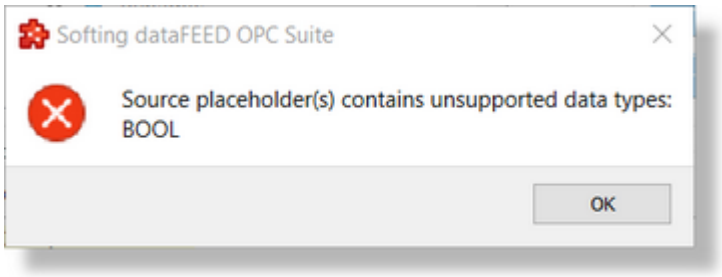
Note: Arrays are not supported being used as operands on preprocessing engine actions.

The items from the address space which does not fit with the defined filters are visible into the address space tree being grayed out. The items which are allowed being used for a specific preprocessing engine action are using a simple black color.

© Softing Industrial Automation GmbH

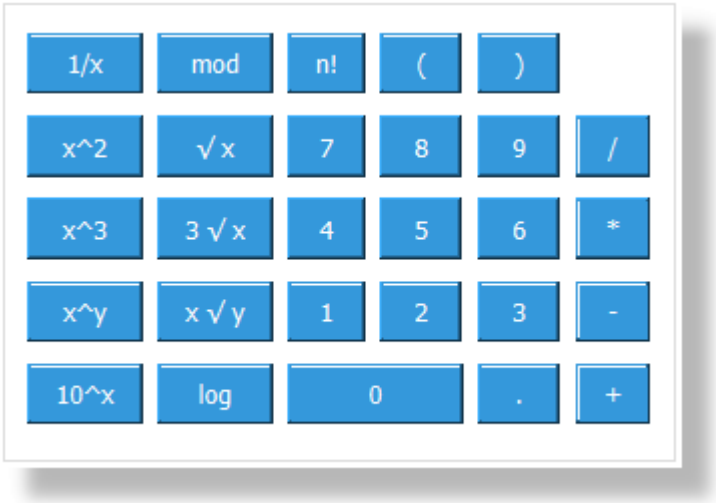


If an unsupported item is used as operand, on validate procedure the user is notified that the item is invalid based on an error message.



Operations Component: this component allows the user to select the operation(s) and operand(s) - if constant operands are used.

It allows usage of mathematical operation(s) and operand(s) - it can be used one of the existing operations or a constant value as operand. The operation can be selected using the mouse or the keyboard. Complex and composed operations are allowed. Also, it is allowed to group a set of operations (including its operands) using the opening and closing brackets.



The usage of Operation Component is similar with the usage of generic Calculator applications (based on usage of the mouse or specific shortcut keys). Immediately after an operation or an operand is selected it becomes available on text editor on the actual position of cursor.

Note: Usage of specific operands (constant or variables) can be used in parallel.

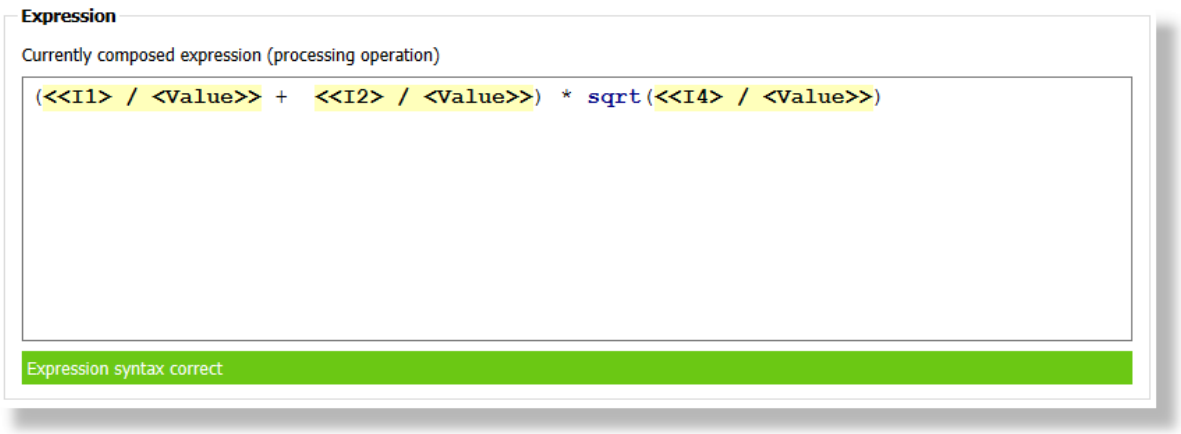
Attention: It is very important to correctly select the position of cursor before inserting a new operation or operand in order to generate correct LUA expressions.

Expression Composer Text Editor: this component is uses for visualizing and editing the composed expression.

This text editor is limited to operation/operands selection, cursor position management and a limited text editing on variant constants used as operands on current expression. Editing the operations (as text editor) is not possible - the operations being inserted by double clicking on proper operation on Operation Component or usage of corresponding shortcut key.

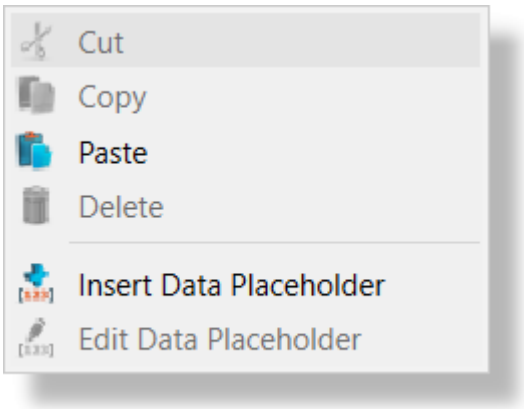
The placeholders used as operands can be inserted directly from Address Space Tree View by selecting an item and pressing the  button.

© Softing Industrial Automation GmbH

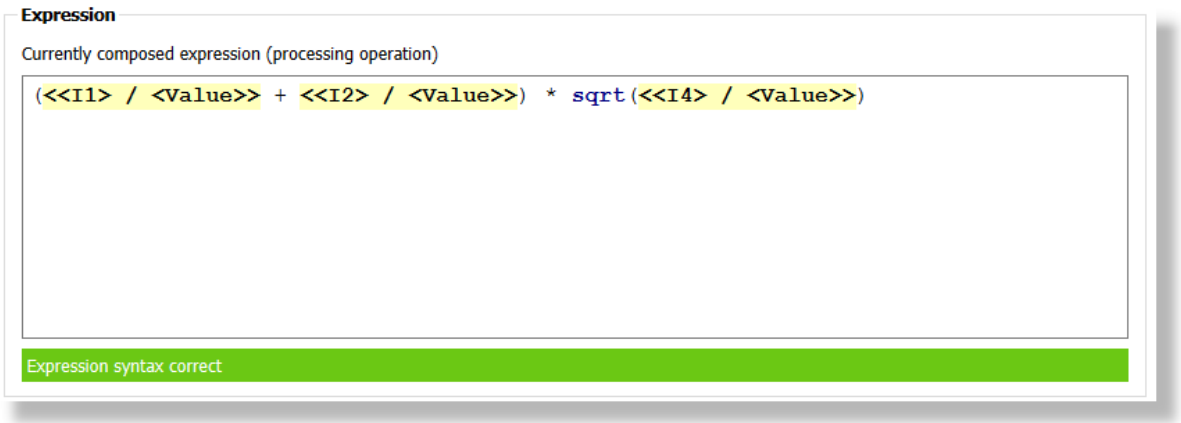


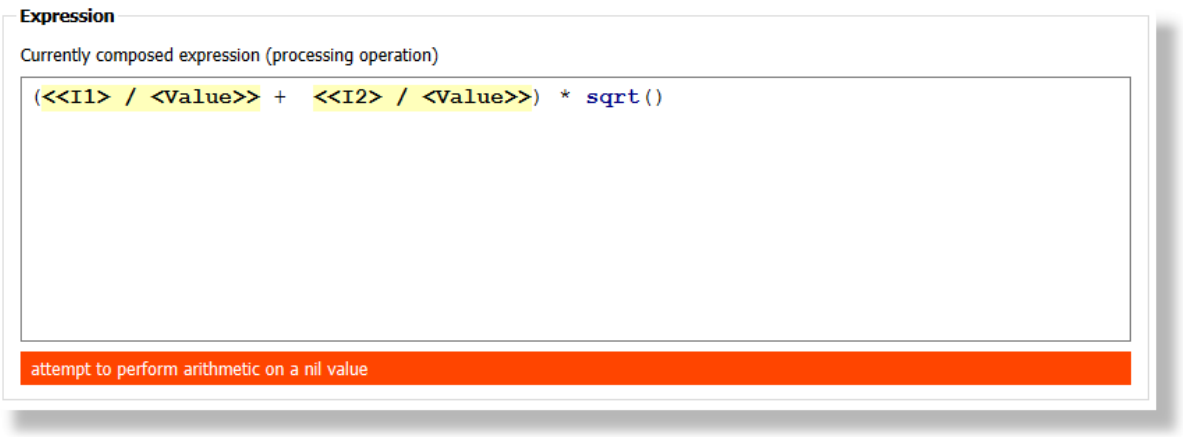
The complete action body (composed expression) is visible on text editor. Also, it contains a context menu which can be activated by right click and contains the following availabe options:

- Cut/Copy/Paste options - allowing to operate on a selected sequence of action body.
- Delete option - allowing the possibility to delete a sequence of action body.
- Insert/Edit Data Placeholder option - allowing the possibility to add/modify a placeholder.



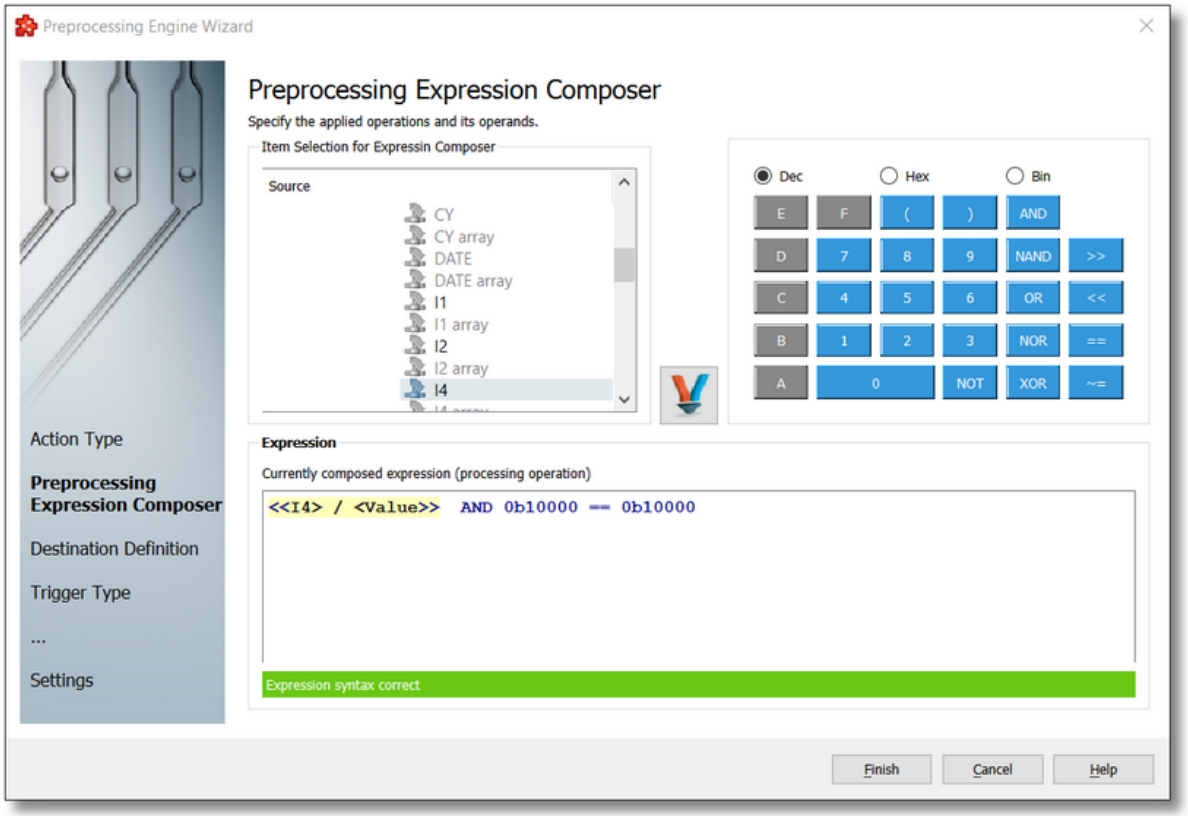
In addition, the text editor is validating the action on each operation - the result of being available on the bottom part of the text editor, signaling of if the expression is valid or not.





A final validation is also operated on finalizing the composed expression action body. In case of a negative result of global validation - an error message is shown and it make it impossible to continue the action validation (the only available option, before fixing the error, being closing the action editor).

Logical Operation allows defining expression containing a set of logical and binary operations and its corresponding operands. Usage of multiple operations, grouping of multiple operations or usage of operations on different layers are also allowed.



Available logical operations:

© Softing Industrial Automation GmbH

- AND, not AND (NAND) - logical/binary operations.
- OR, not OR (NOR) and exclusive OR (XOR) - logical/binary operations.
- == , ~= (not equal) - equivalence operators.
- >>, << - binary shift operations.
- NOT - binary operation

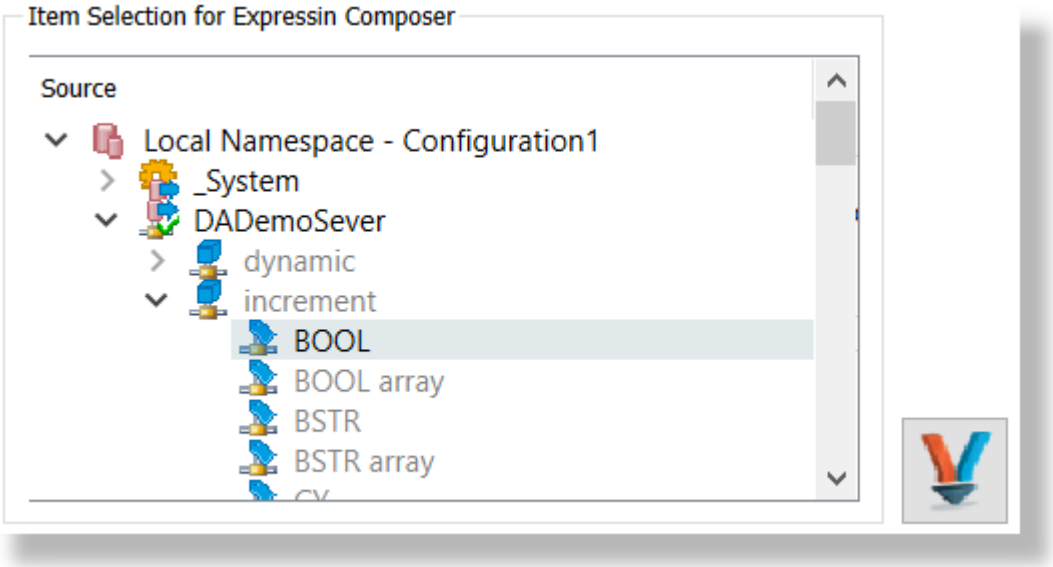
Attention: The binary NOT operation is bitwise NOT, or complement operation. (ex. NOT 0 => FFFFFFFFFFFFFFFF, NOT 1 => FFFFFFFFFFFFFFFE)

Available logical operands:

- variables: the value of a specified item available into the exposed address space - basically it can be selected an item from the available address space (this item will be part of an internal subscription (see Action Settings wizard page) which will update the internal cache of selected item - later on action execution it will be used the cached value).
- constants: the value should be specified by the user - this values are possible being introduced in various formats: Decimal (default), Hexadecimal and Binary.

The Expression Composer wizard page contains the following components:

Address Space Tree View: this allows the selection of an item from the entire available address space. Browse operations are allowed for all the defined connections and for all available levels.



Also it is allowed to modify the address space by right clicking into the address space tree. The available operations being:

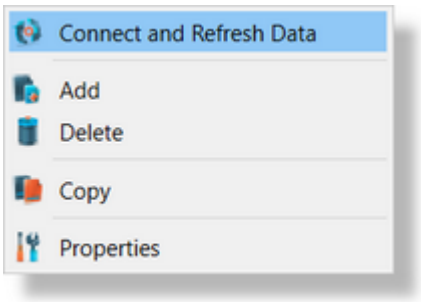
- Connect and Refresh Data - tries to re-establish the connection to the data source and re-browse it (this operation is needed specially when the data source changes its address space)

- Add/Delete - allows to add or delete an item from the available address space.

Attention: Adding a node requires to know the syntax used by the corresponding data source. Different data source types could require different syntax.

Attention: Deleting a node will deleted the node only from the exposed address space by the configuration tool. This operation will be visible from all the data destination and data processing plug-ins which has access to the address space.

- Copy/Paste (when clipboard is not empty) - allows to copy/paste a node or an item into the address space.
- Properties - allows to visualize or modify the properties of a selected item.



It is allowed to navigate into the address space and to select a variable. Once the selection is done - it can be decided to be used as operand into the expression composer for a defined operation by pressing the  button. Immediately, a corresponding placeholder is introduced into the Expression Composer Text Editor on current cursor position.

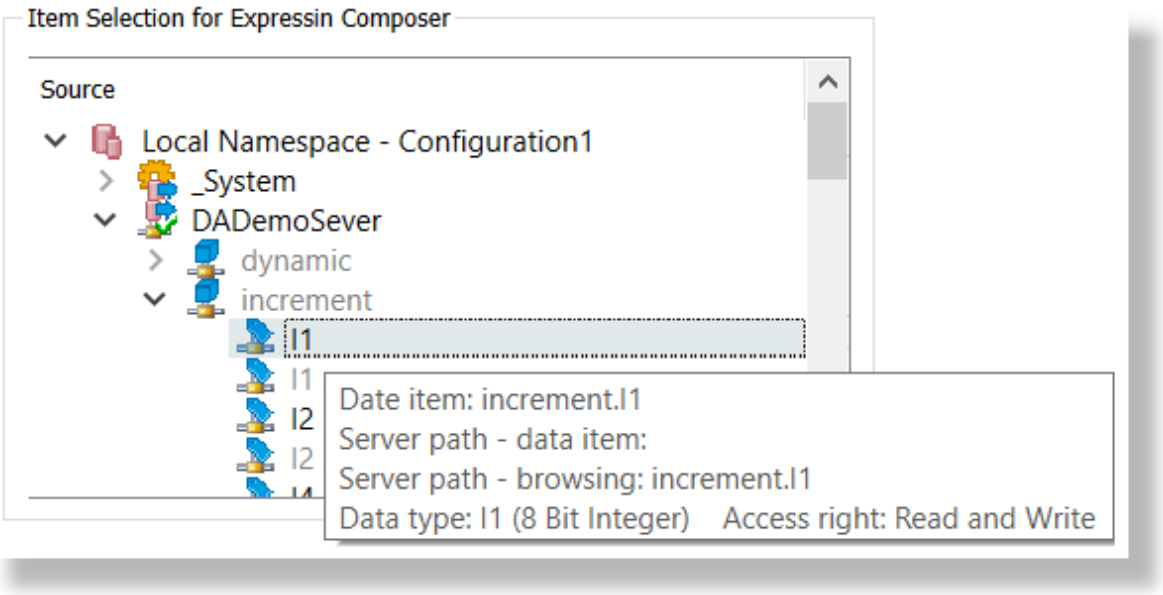
The access to the address space is limited to the item using the following data types:

- Bool (Boolean)
- I1 (8 bit integer), I2 (16 bit integer), I4 (32 bit integer), I8 (64 bit integer),
- UI1 (8 bit unsigned integer), UI2 (16 bit unsigned integer), UI4 (32 bit unsigned integer), UI8 (64 bit unsigned integer)
- Variant (Any).

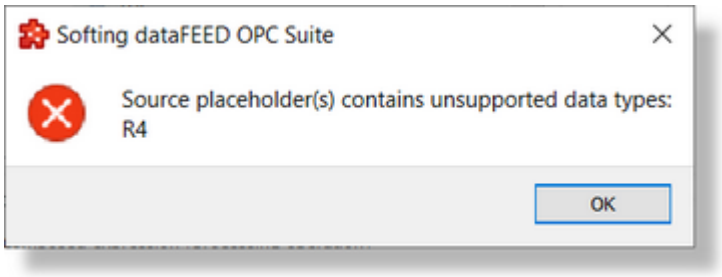
Note: Arrays are not supported being used as operands on preprocessing engine actions.

The items from the address space which does not fit with the defined filters are visible into the address space tree being grayed out. The items which are allowed being used for a specific preprocessing engine action are using a simple black color.

© Softing Industrial Automation GmbH



If an unsupported item is used as operand, on validate procedure the user is notified that the item is invalid based on an error message.



Operations Component: this component allows the user to select the operation(s) and operand(s) - if constant operands are used.

It allows allows usage of logical operation(s) and operand(s) - it can be used one of the existing operations or a constant value as operand. The operation can be selected using the mouse or the keyboard. Complex and composed operations are allowed. Also it is allowed to group a set of operations (including its operands) using the opening and closing brackets).



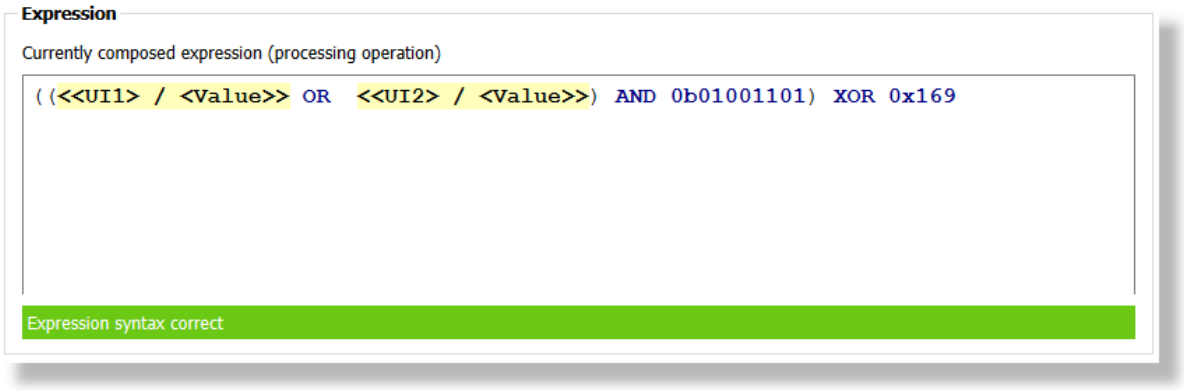
© Softing Industrial Automation GmbH

Note: Constant operations are supported being introduced or defined in binary, decimal or hexadecimal format.

Expression Composer Text Editor: this component is uses for visualizing and editing the composed expression.

This text editor is limited to operation/operands selection, cursor position management and a limited text editing on variant constants used as operands on current expression. Editing the operations (as text editor) is not possible - the operations being inserted by double clicking on proper operation on Operation Component or usage of corresponding shortcut key.

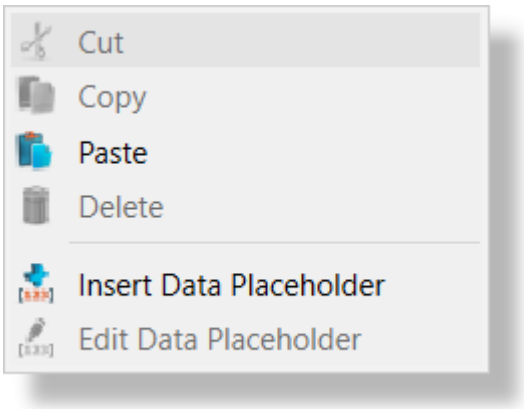
The placeholders used as operands can be inserted directly from Address Space Tree View by selecting an item and pressing the  button.



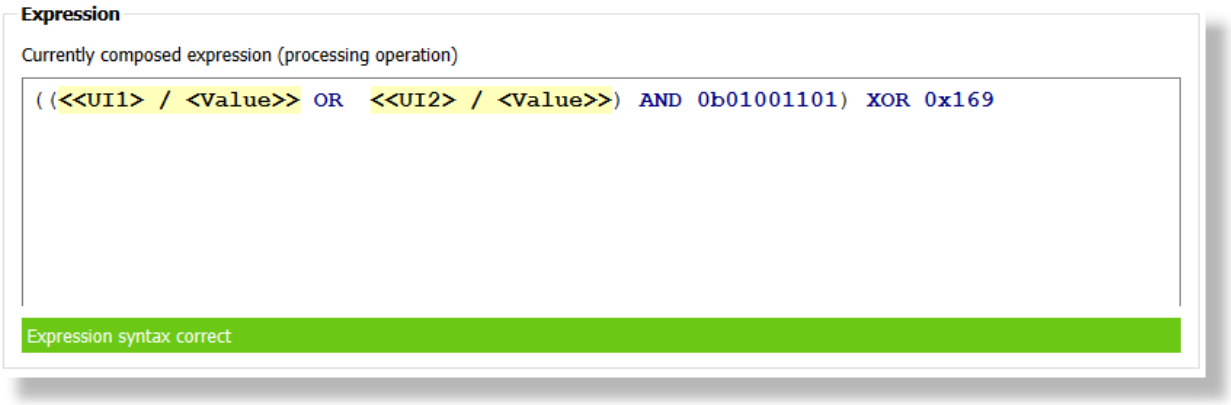
Note: Expressions with further Boolean destination item must have equivalence operators (==, ~=). Example: *<<I4> / <Value>> AND 0b10000 == 0b10000*

The complete action body (composed expression) is visible on text editor. Also, it contains a context menu which can be activated by right click and contains the following availabe options:

- Cut/Copy/Paste options - allowing to operate on a selected sequence of action body.
- Delete option - allowing the possibility to delete a sequence of action body.
- Insert/Edit Data Placeholder option - allowing the possibility to add/modify a placeholder.



In addition, the text editor is validating the action on each operation - the result of being available on the bottom part of the text editor, signaling of if the expression is valid or not.

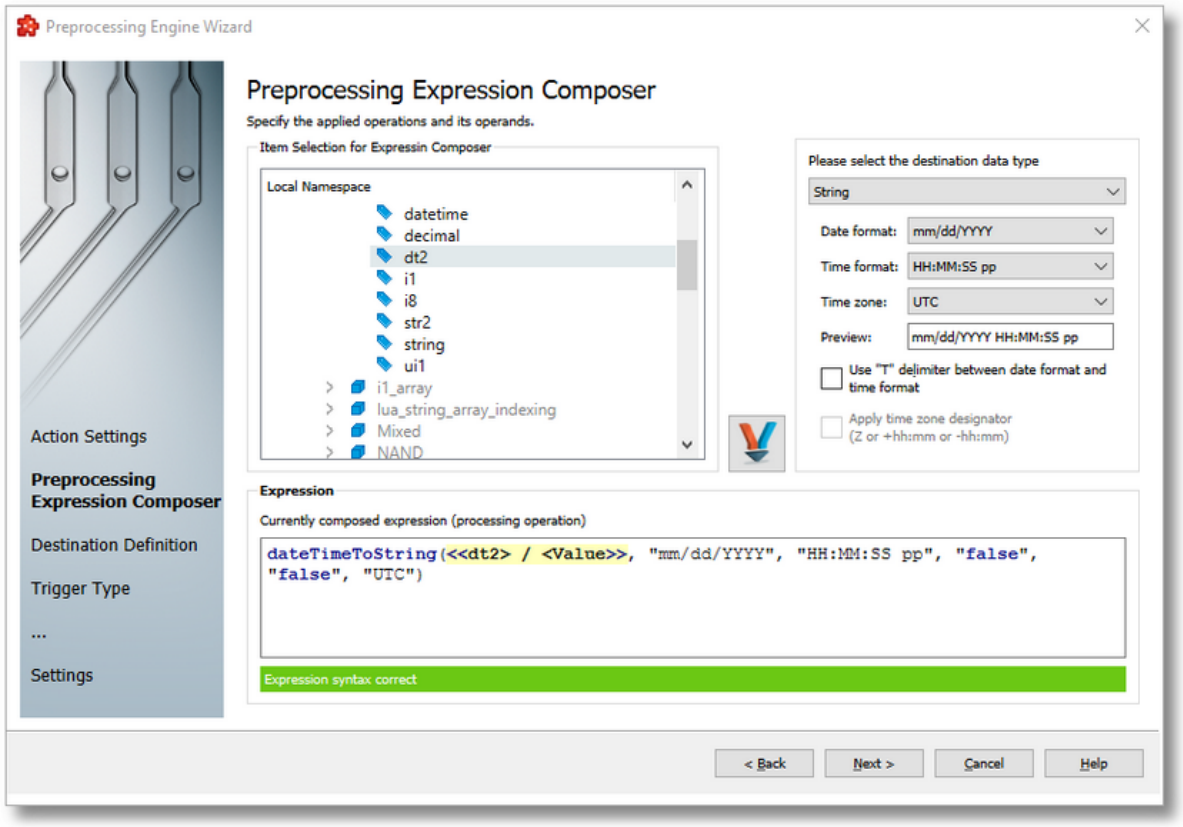




A final validation is also operated on finalizing the composed expression action body. In case of a negative result of global validation - an error message is shown and it make it impossible to continue the action validation (the only available option, before fixing the error, being closing the action editor).

Data Type Conversion Operation allows defining an expression to convert values between different types. Different conversion parameters can be set, depending on the source and destination data types.

© Softing Industrial Automation GmbH

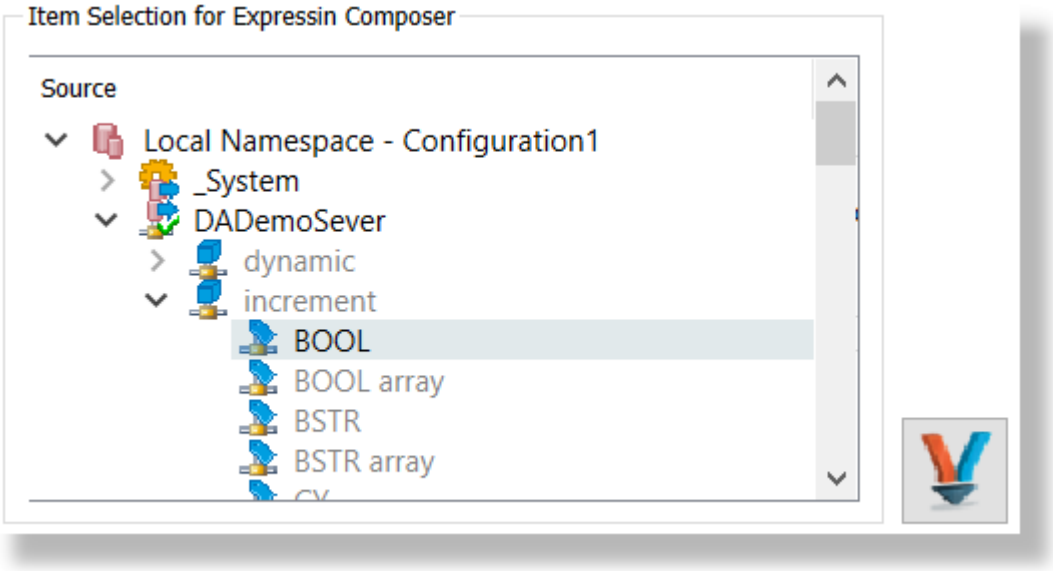


Available conversion operations:

- Integer/Real data types to String, Boolean, Integer/Real
- Boolean to Integer/Real, String conversion
- String to Integer/Real, Boolean, Date-time
- Date-time conversion to String

The Expression Composer wizard page contains the following components:

Address Space Tree View: this allows the selection of an item from the entire available address space. Browse operations are allowed for all the defined connections and for all available levels.



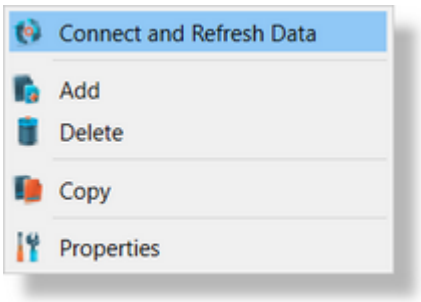
Also it is allowed to modify the address space by right clicking into the address space tree. The available operations being:

- Connect and Refresh Data - tries to re-establish the connection to the data source and re-browse it (this operation is needed specially when the data source changes its address space)
- Add/Delete - allows to add or delete an item from the available address space.

Attention: Adding a node requires to know the syntax used by the corresponding data source. Different data source types could require different syntax.

Attention: Deleting a node will delete the node only from the exposed address space by the configuration tool. This operation will be visible from all the data destination and data processing plug-ins which has access to the address space.

- Copy/Paste (when clipboard is not empty) - allows to copy/paste a node or an item into the address space.
- Properties - allows to visualize or modify the properties of a selected item.



It is allowed to navigate into the address space and to select a variable. Once the selection is done - it can be decided to be used as operand into the expression composer for a defined

© Softing Industrial Automation GmbH

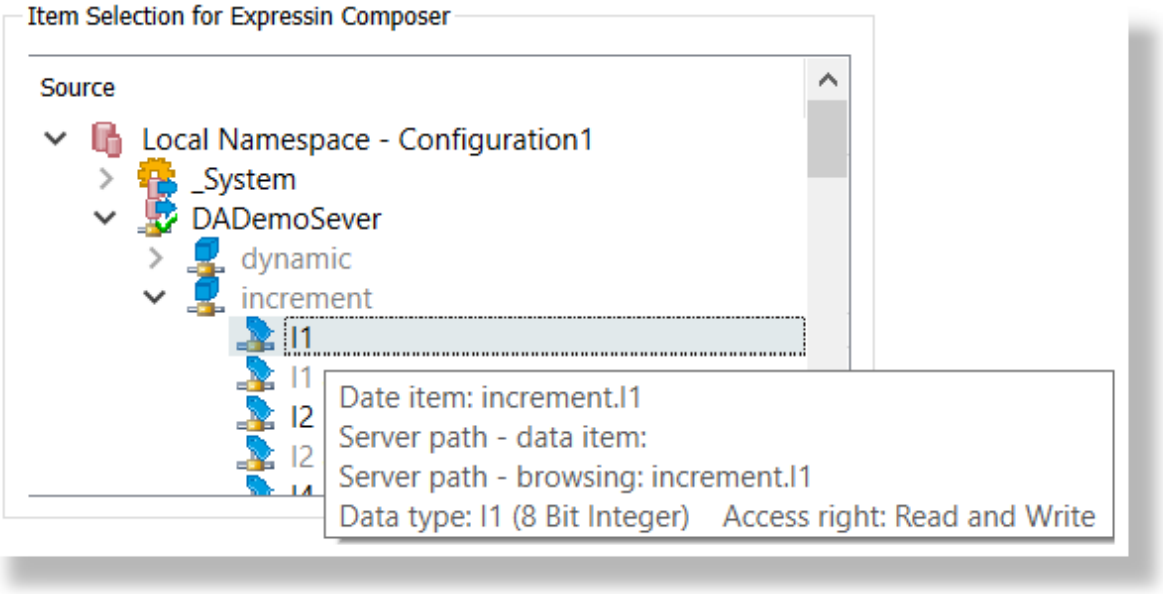
operation by pressing the  button. Immediately, a corresponding placeholder is introduced into the Expression Composer Text Editor on current cursor position.

The access to the address space is limited to the item using the following data types:

- Bool (Boolean)
- I1 (8 bit integer), I2 (16 bit integer), I4 (32 bit integer), I8 (64 bit integer),
- UI1 (8 bit unsigned integer), UI2 (16 bit unsigned integer), UI4 (32 bit unsigned integer), UI8 (64 bit unsigned integer)
- R4 (32 Bit Real), R8 (64 Bit Real)
- DATETIME (Date Time)
- STRING (String)

Note: Arrays are not supported to be used as operands in the conversion preprocessing actions.

The items from the address space which does not fit with the defined filters are visible into the address space tree being grayed out. The items which are allowed to be used for a specific preprocessing engine action are commonly displayed as available.

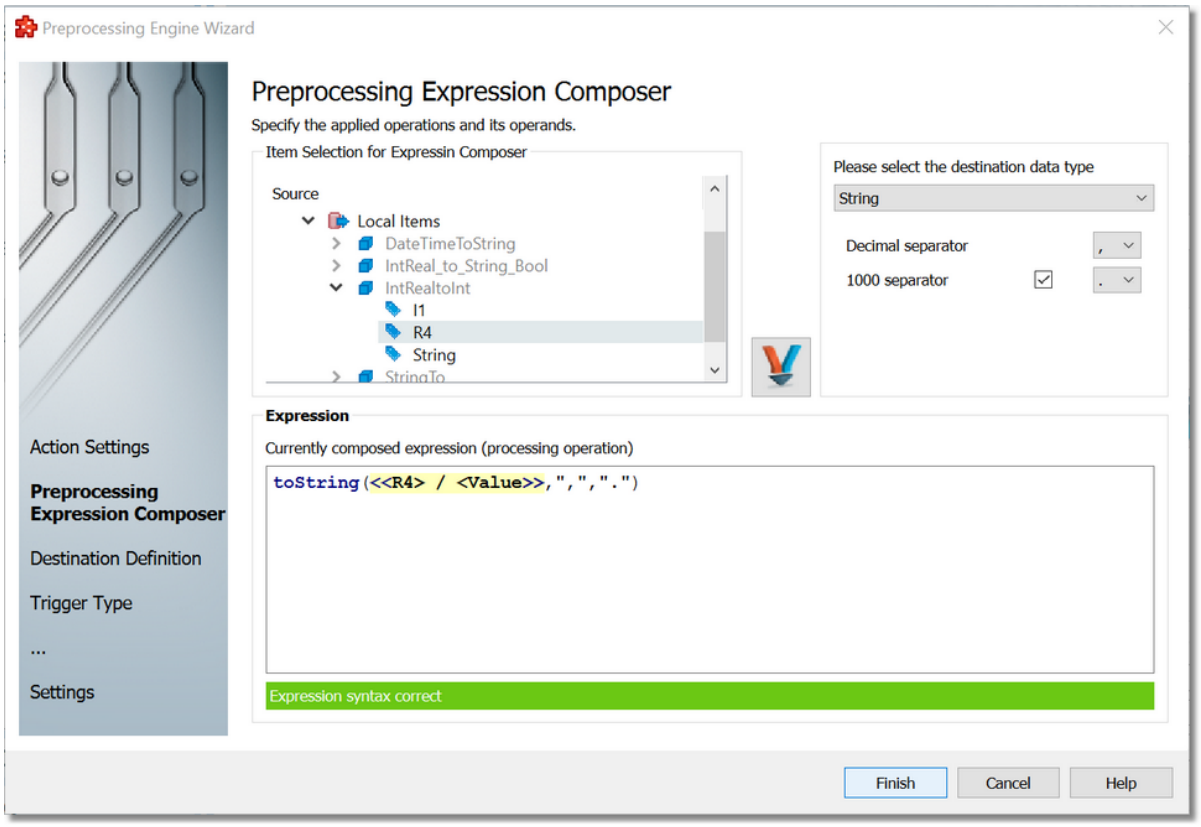


Operations Component: this component allows the user to select the destination data type and the conversion parameters.

Based on the source item data type and the selected destination data type different parameter are available.

Integer/Real data types to String

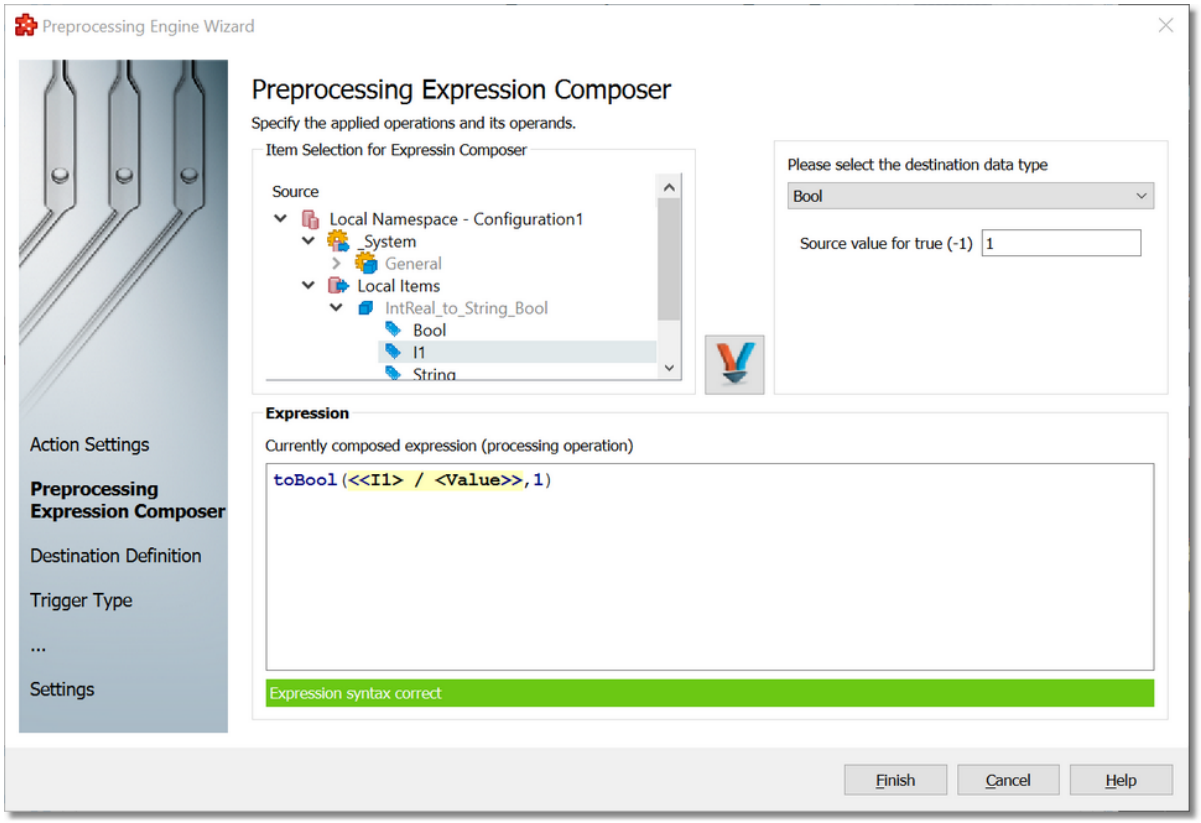
For String destination it is needed to specify the decimal separator and thousand separator which can be optional.



Integer/Real data types to Boolean

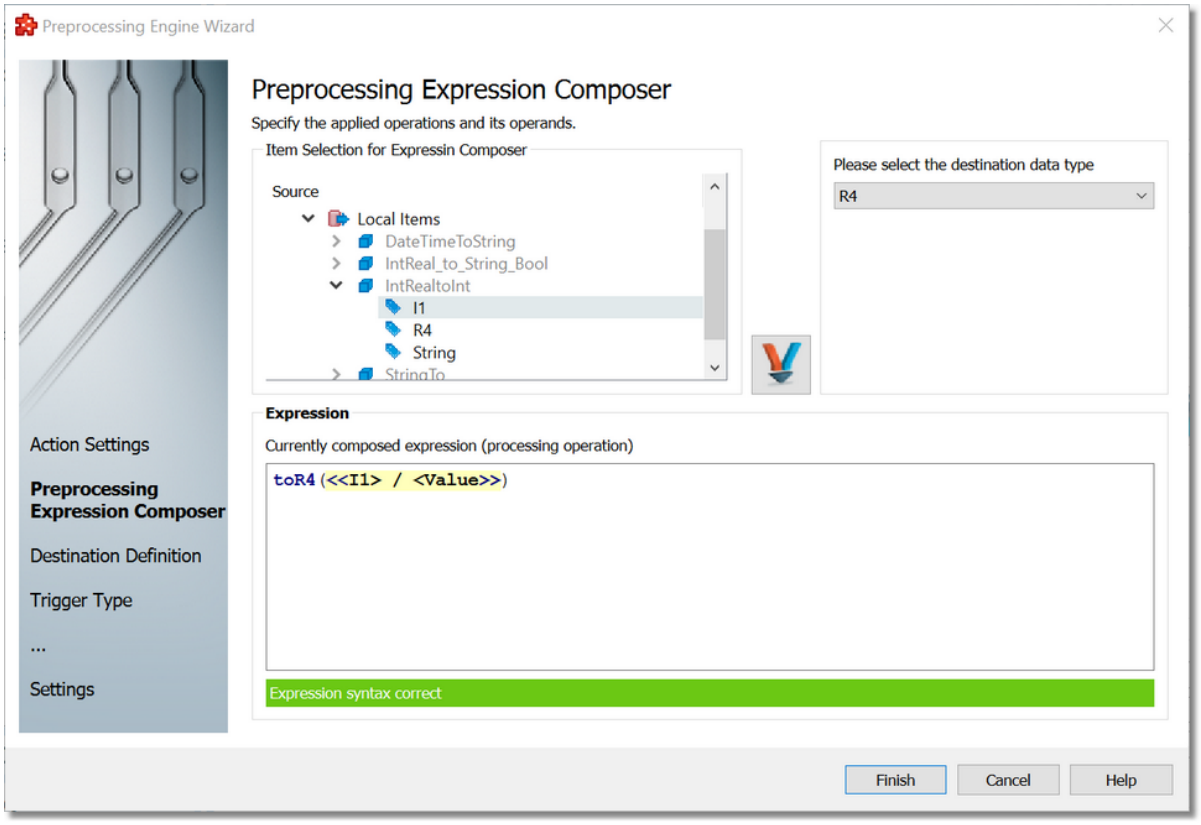
For destination Boolean it is needed to specify the source value for what the destination will be true.

© Softing Industrial Automation GmbH



Integer/Real data types to other Integer/Real

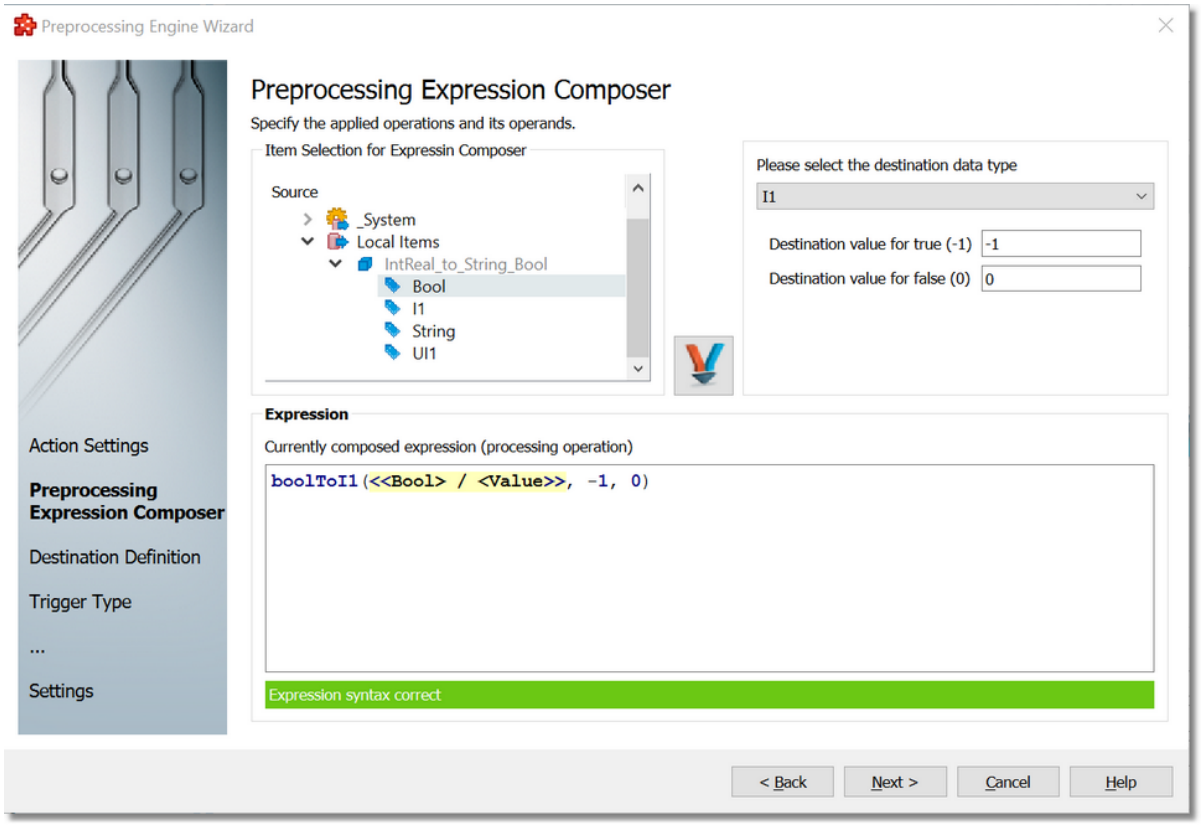
For this conversion there is no parameter to specify. The run-time will try to convert to the specified destination type if it is possible.



Boolean to Integer/Real, String types

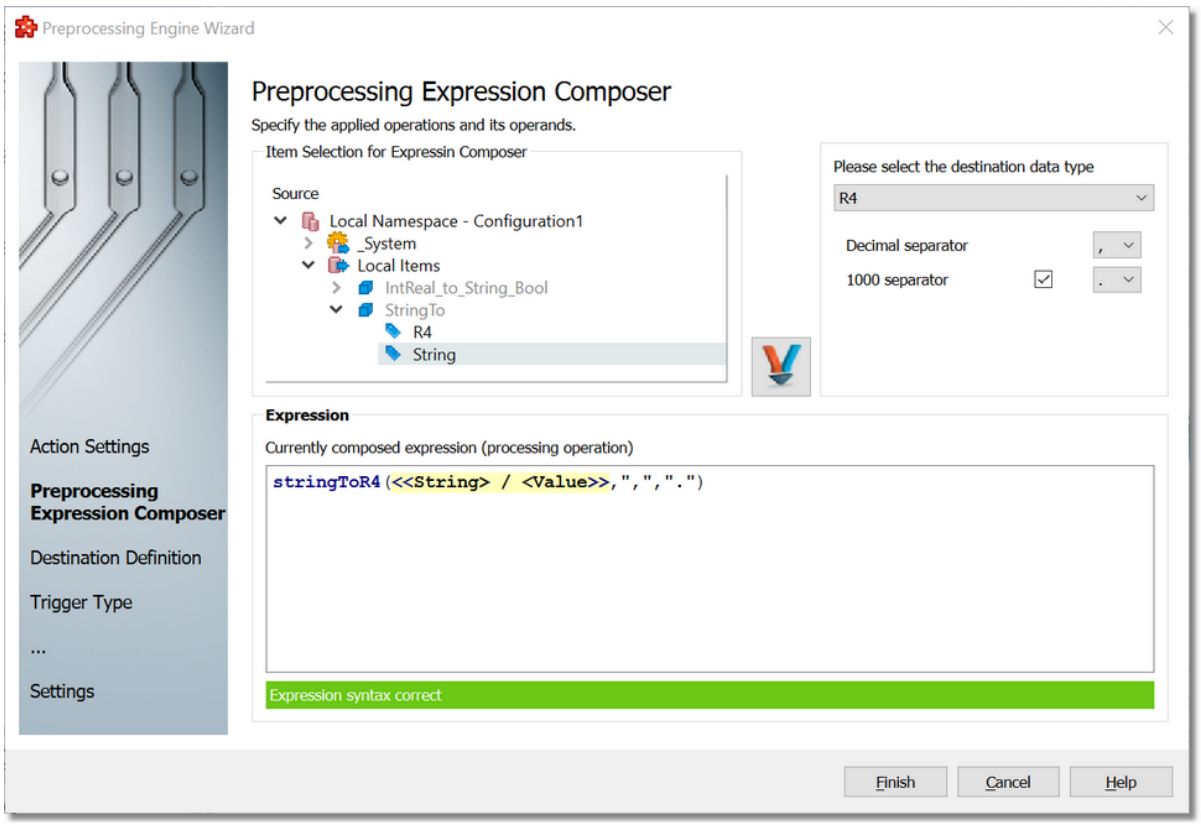
For this conversion "Destination value for true/false" is needed to be specified. This specified values will be written in the destination item based on the source item value.

© Softing Industrial Automation GmbH



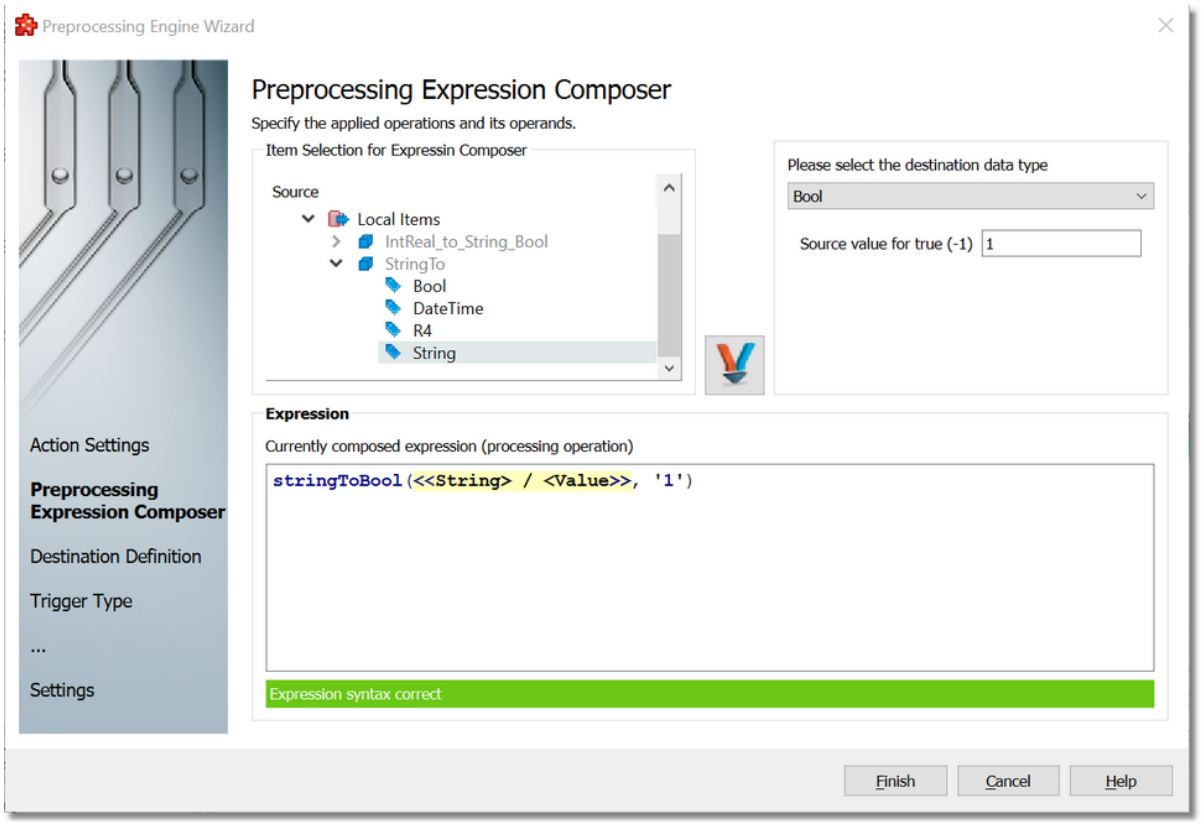
String to Integer/Real

The specified "Decimal separator" and thousand separator (if checked) will be used to parse the source String item and write the value to the destination item.



String to Boolean

If the source item contains the specified string for true than the destination value will be true otherwise it will be false.



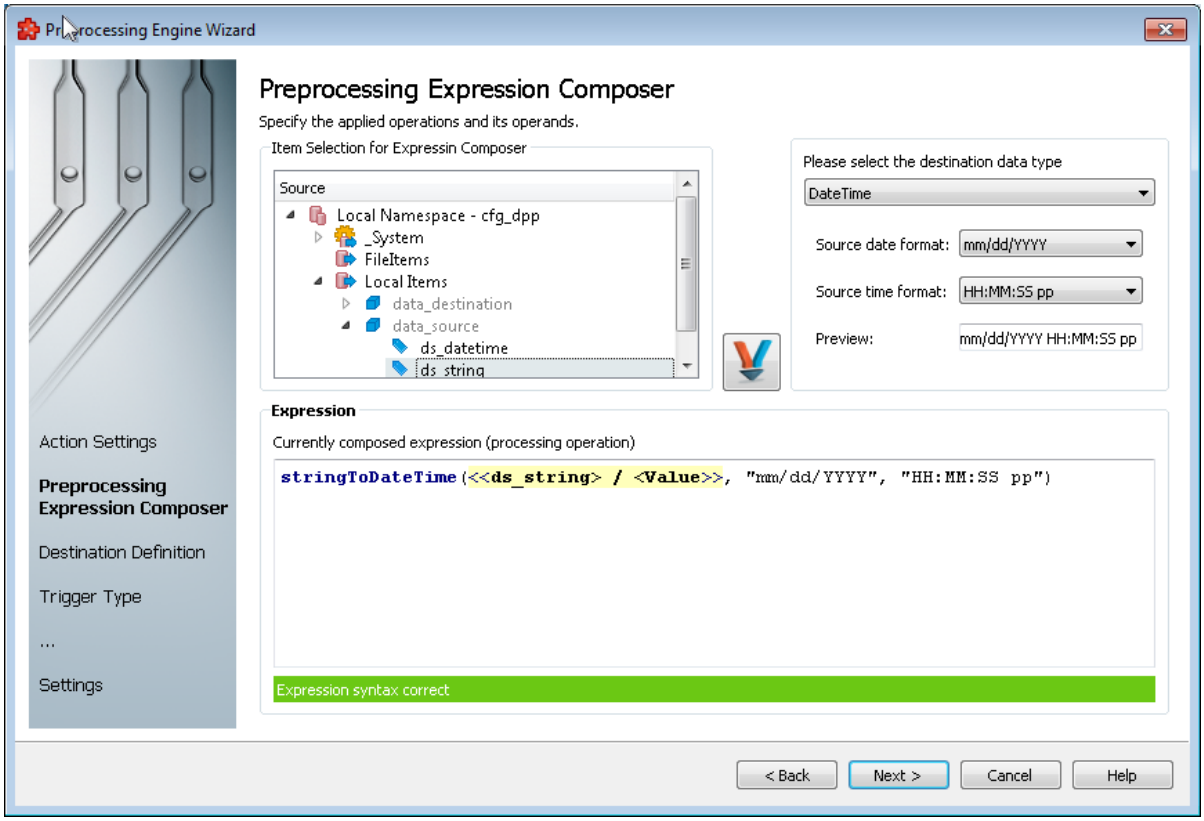
String to DateTime

For this conversion "Date format" and "Time format" parameters needs to be specified. These two parameters will define the format in which the source String must provide the input.

Possible date formats: `mm/dd/YYYY, dd-mm-YYYY, dd.mm.YYYY, dd/mm/YYYY, YYYY-mm-dd, mm-dd-YYYY`

Time formats: `HH:MM:SS pp, HH:MM:SS.FFF pp, HH:MM:SS.FFF, HH:MM:SS`

© Softing Industrial Automation GmbH



DateTime conversion to String

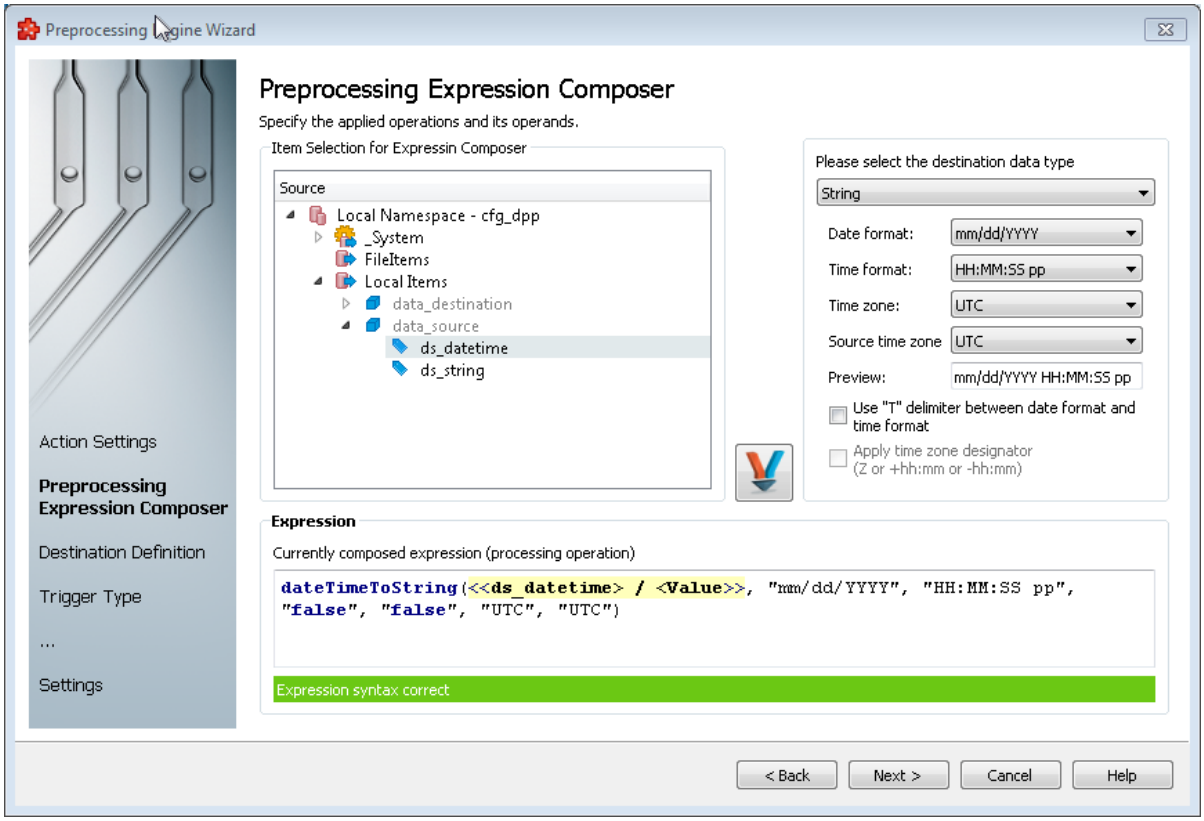
With this conversion it is possible to convert a DateTime source item value to a String destination item. It is necessary to specify the "Date format", "Time format", "Time zone", "Source time zone".

Possible date formats: `mm/dd/YYYY, dd-mm-YYYY, dd.mm.YYYY, dd/mm/YYYY, YYYY-mm-dd, mm-dd-YYYY`

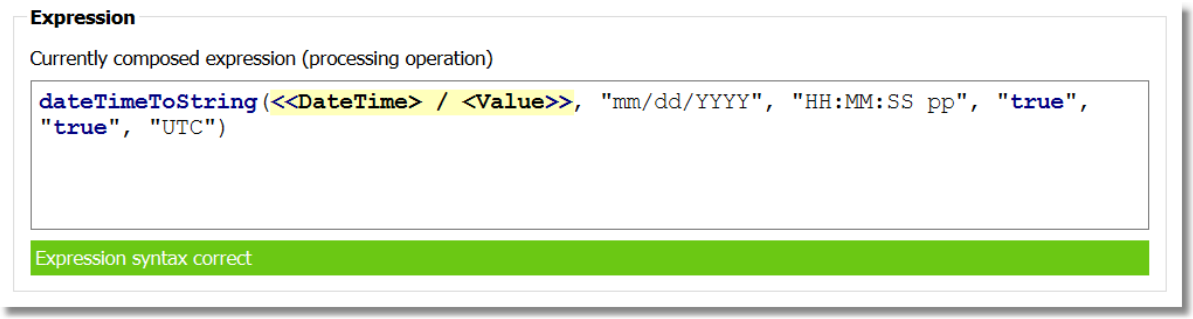
Time formats: `HH:MM:SS pp, HH:MM:SS.FFF pp, HH:MM:SS.FFF, HH:MM:SS`

Time zones: `UTC, Local`

Optionally it is possible to specify a "T" delimiter and also if a time zone designator is needed.



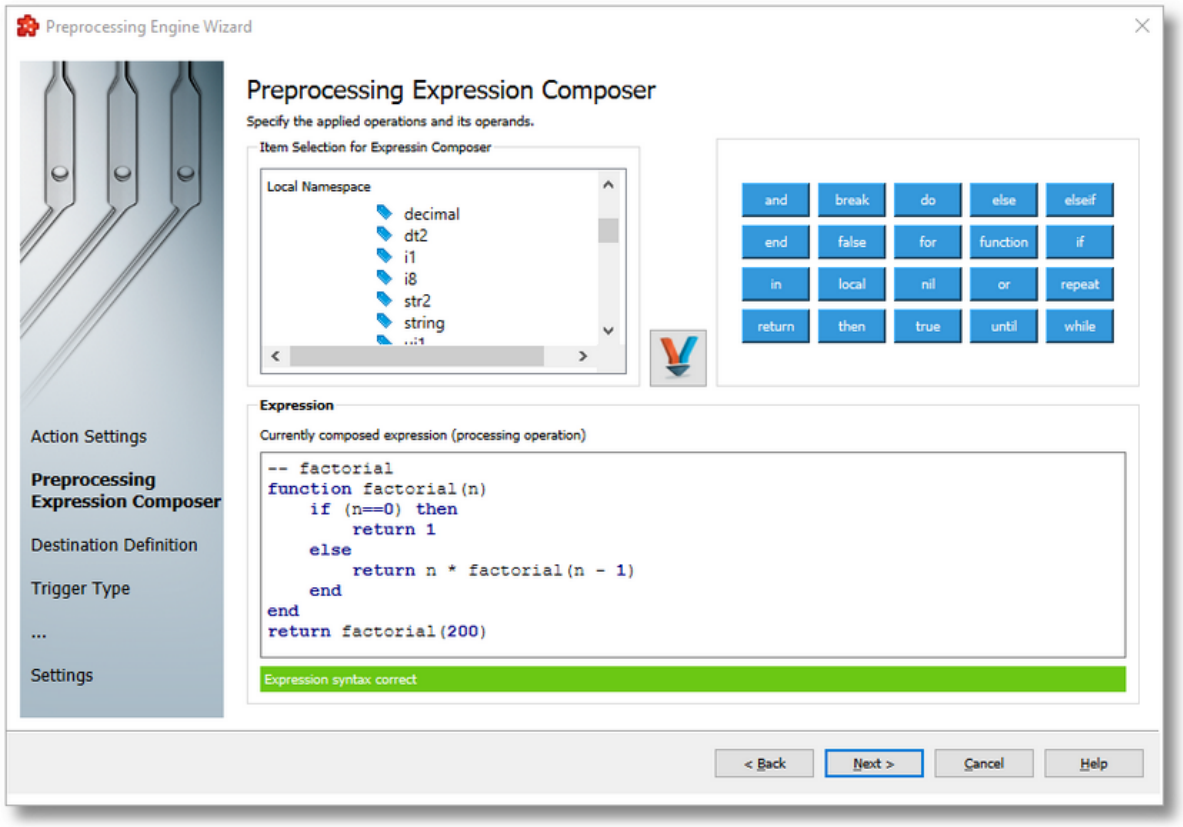
Expression Composer Text Editor: this component is used for visualizing the composed expression. It is only for preview purpose. It is not possible to edit or modify the content.



A final validation is also operated on finalizing the composed expression action body. In case of a negative result of global validation - an error message is shown, the action validation preventing to continue editing the action (the available options being to fix the error or cancel the action editing operation).

LUA Operation allows defining expression containing almost any LUA script which returns a value. The expression can contain any LUA keyword, function, expression but must return a value which will be written into the destination item.

© Softing Industrial Automation GmbH

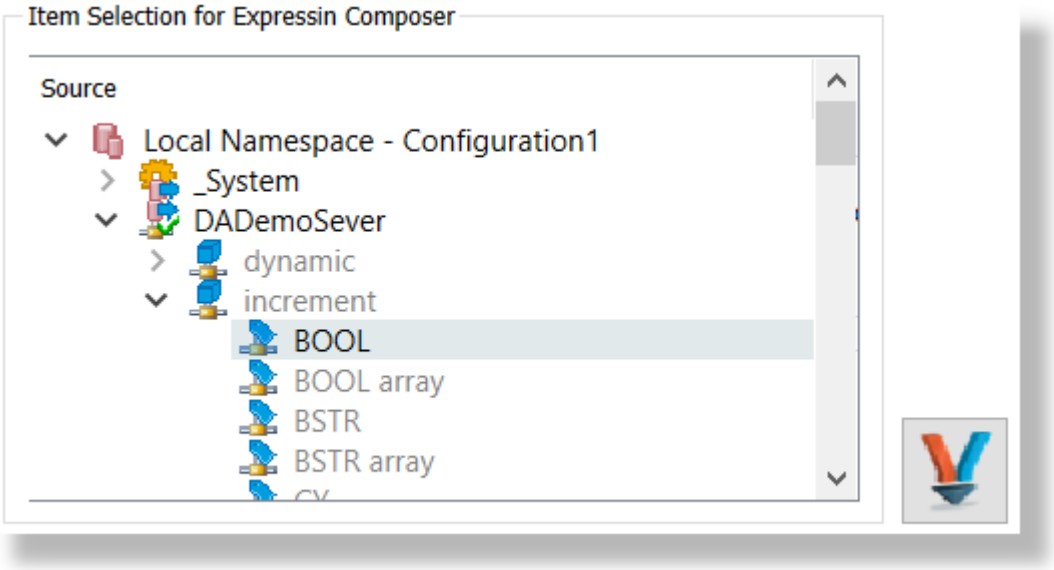


Available lexical conventions:

- The following keywords are reserved and cannot be used as names:
  - `and break do else elseif end false for function if in local nil not or repeat return then true until while`
- The following strings denote other tokens:
  - `+ - * / % ^ #`
    `== ~= <= >= < > =`
    `( ) { } [ ]`
    `; : , . .. ...`
- Numerical constant can be written with an optional decimal part and an optional decimal exponent:
  - `3 3.0 3.1416 314.16e-2 0.31416E1 0xff 0x56`
- Comment starts with a double hyphen (`--`) anywhere outside a string.

The Expression Composer wizard page contains the following components:

Address Space Tree View: this allows the selection of an item from the entire available address space. Browse operations are allowed for all the defined connections and for all available levels.



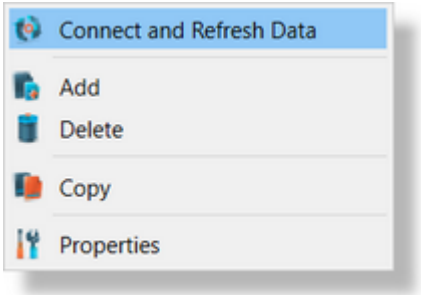
Also it is allowed to modify the address space by right clicking into the address space tree. The available operations being:

- Connect and Refresh Data - tries to re-establish the connection to the data source and re-browse it (this operation is needed specially when the data source changes its address space)
- Add/Delete - allows to add or delete an item from the available address space.

Attention: Adding a node requires to know the syntax used by the corresponding data source. Different data source types could require different syntax.

Attention: Deleting a node will delete the node only from the exposed address space by the configuration tool. This operation will be visible from all the data destination and data processing plug-ins which has access to the address space.

- Copy/Paste (when clipboard is not empty) - allows to copy/paste a node or an item into the address space.
- Properties - allows to visualize or modify the properties of a selected item.



© Softing Industrial Automation GmbH

It is allowed to navigate into the address space and to select a variable. Once the selection is done - it can be decided to be used as operand into the expression composer for a defined operation by pressing the  button. Immediately, a corresponding placeholder is introduced into the Expression Composer Text Editor on current cursor position.

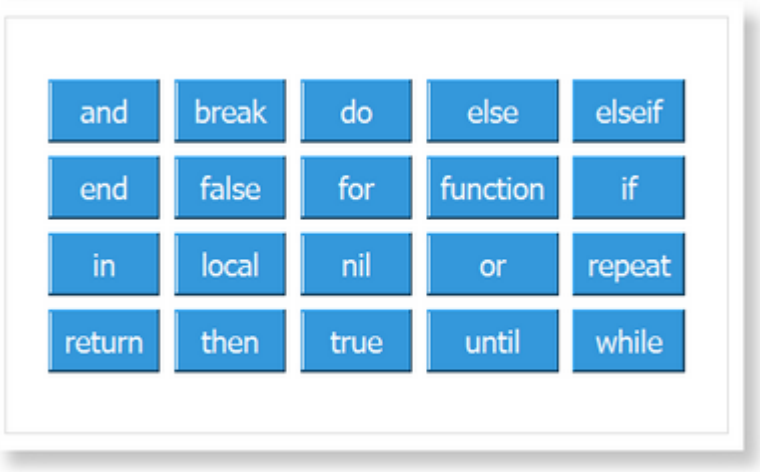
The access to the address space is limited to the item using the following data types:

- Bool (Boolean)
- I1 (8 bit integer), I2 (16 bit integer), I4 (32 bit integer), I8 (64 bit integer),
- UI1 (8 bit unsigned integer), UI2 (16 bit unsigned integer), UI4 (32 bit unsigned integer), UI8 (64 bit unsigned integer)
- R4 (32 Bit Real), R8 (64 Bit Real)
- DATETIME (Date Time)
- STRING (String)

Note: Arrays are supported being used as operands in LUA preprocessing engine actions (LUA Array Support).

Operations Component: this component allows the user to select from some of the most used keywords in LUA scripting.

The keywords can be selected from the operation component buttons, from where it will be inserted at the current cursor position in the Expression Composer.

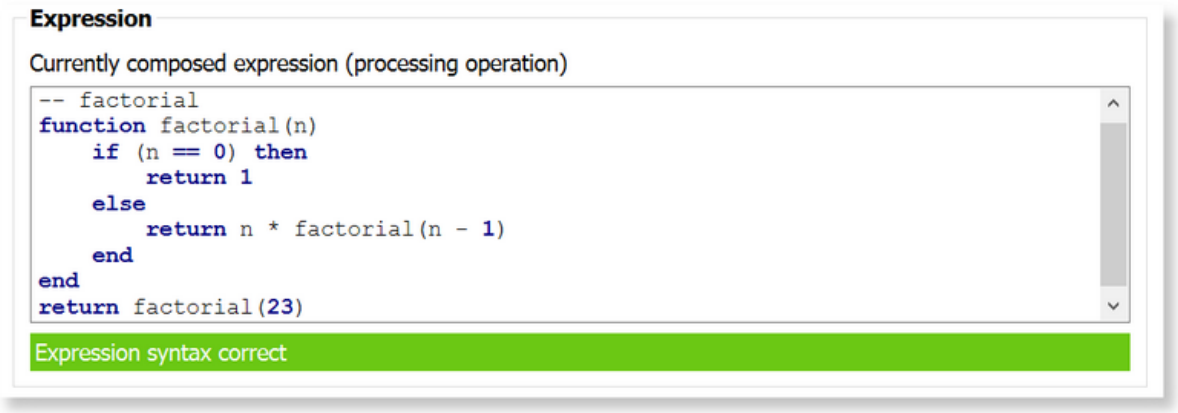


Note: Not only the predefined keywords can be inserted. Any valid LUA keyword, expression can be manually edited in the expression composer.

Expression Composer Text Editor: this component is uses for visualizing and editing the composed expression.

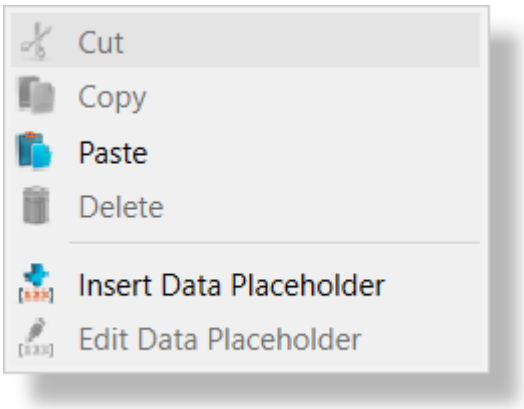
This text editor is not limited to operation/operands selection, cursor position management and a text editing are allowed. Composer expression is reevaluated at each key-press and the syntax correctness is displayed in the editor status line.

The placeholders used as operands can be inserted directly from Address Space Tree View by selecting an item and pressing the  button.



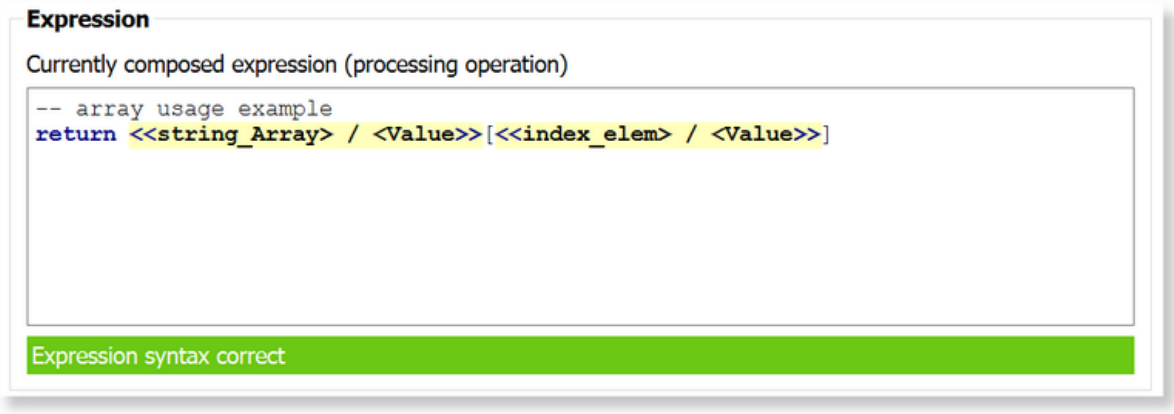
The complete action body (composed expression) is visible on text editor. Also, it contains a context menu which can be activated by right click and contains the following available options:

- Cut/Copy/Paste options - allowing to operate on a selected sequence of action body.
- Delete option - allowing the possibility to delete a sequence of action body.
- Insert/Edit Data Placeholder option - allowing the possibility to add/modify a placeholder.



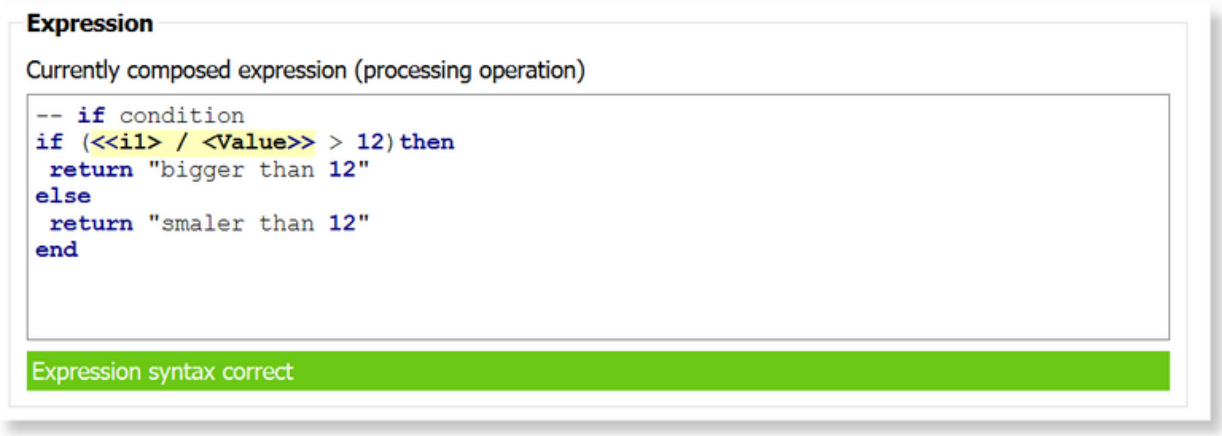
Array support is enabled in LUA Engine Operations. After an array item placeholder is it possible to specify the index of an element between the "[" and "]" brackets. It doesn't need to be a constant value it can be also an integer item placeholder.

© Softing Industrial Automation GmbH

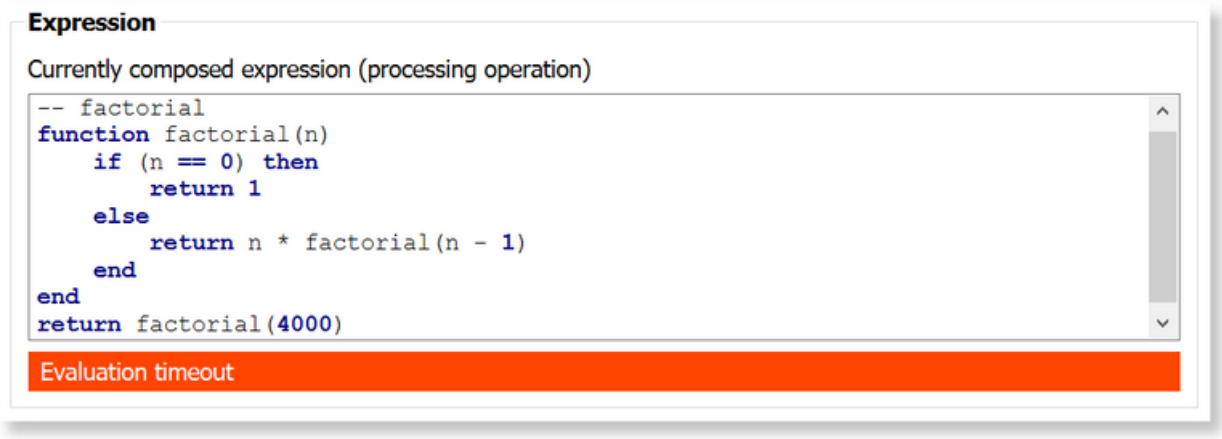


Note: LUA array indexing starts with 1. It means that the array starts with index 1.

In addition, the text editor is validating the action on each operation - the result of being available on the bottom part of the text editor, signaling of if the expression is valid or not.



Note: Expressions *can timeout (for example infinite loop) while editing.* But this is not an error, it just states that the current script (with dummy values for placeholders) times out, but the syntax is correct.



A final validation is also operated on finalizing the composed expression action body. In case of a negative result of global validation - an error message is shown, the action validation preventing to continue editing the action (the available options being to fix the error or cancel the action editing operation).

Example LUA scripts:

1. The LUA expression determines the greatest element for the values of an array source item.

```
local array = <<I1_Array>> / <Value>>
local max = 0

for key, value in ipairs(array) do
 local valueIsNil = (value == nil)
 local valueIsGreater = (value > max)
 if valueIsNil == false and valueIsGreater == true then
  max = value
 end
end

return max
```

   - Source item value: `[-48,55,-87,90,39,96,74,19]`
   - Destination item value: `96`

2. The LUA expression determines the standard deviation [1] for the values of an array source item.

```
local array = <<R4 array> / <Value>>

function mean(t)
  local val, sum = 0, 0
  for k, v in ipairs(t) do
    sum = sum + v
  end
  val = sum / #t
```


© Softing Industrial Automation GmbH

```
  return val
end

function stdev(t)
  local val, squares, avg = 0, 0, mean(t)
  for k, v in ipairs(t) do
    squares = squares + ((avg - v) ^ 2)
  end
  local variance = squares / #t
  val = math.sqrt(variance)
  return val
end

return stdev(array)
```

- Source item value: `[35.19,32.69,3.46,16.74,42.74,33.82,26.33,14.15,25.12,8.14,8.18,12.48,34.8,29.2,5.38,9.85]`
- Destination item value: `12.301129`

3. The LUA expression determines the cumulative standard deviation [2] for the values of an array source item.

```
local array = <<R4 array> / <Value>>

function stdev()
  local val, sum, sumsq, k = 0,0,0,0
  return function(n)
    sum, sumsq, k = sum + n, sumsq + n^2, k+1
    val = math.sqrt((sumsq / k) - (sum / k)^2)
    return string.format("%.2f", val)
  end
end

function createString(in1, out)
  if out == nil then
    out = string.format("%s", in1)
  else
    out = string.format("%s,%s", out, in1)
  end
  return out
end

ldev = stdev()

local output
for i, v in ipairs(array) do
  output = createString(ldev(v), output)
end

return output
```

- Source item value: `[29.37,37.41,8.44,15.35,30.73,11.24,8.35,37.23,9.07,6.73,8.14,40.73,48.24,39.07,49.01,6.51]`

- Destination item value: "0.00,4.02,12.21,11.38,10.68,10.89,11.17,11.88,11.95,12.09,11.98,13.03,14.57,14.68,15.54,15.72"

4. The LUA expression performs a circle sorting algorithm [3] on the values of an array source item.

```
local array = <<R4 array> / <Value>>

function innerCircle (t, lo, hi, swaps)
  if lo == hi then return swaps end
  local high, low, mid = hi, lo, math.floor((hi - lo) / 2)
  while lo < hi do
    if t[lo] > t[hi] then
      t[lo], t[hi] = t[hi], t[lo]
      swaps = swaps + 1
    end
    lo = lo + 1
    hi = hi - 1
  end
  if lo == hi then
    if t[lo] > t[hi + 1] then
      t[lo], t[hi + 1] = t[hi + 1], t[lo]
      swaps = swaps + 1
    end
  end
  swaps = innerCircle(t, low, low + mid, swaps)
  swaps = innerCircle(t, low + mid + 1, high, swaps)
  return swaps
end

function circleSort (t)
  while innerCircle(t, 1, #t, 0) > 0 do end
end

circleSort(array)

function createString(in1, out)
  if out == nil then
    out = string.format("%.2f", in1)
  else
    out = string.format("%s,%.2f", out, in1)
  end
  return out
end

local output
for i, v in pairs(array) do
  output = createString(v, output)
end

return output
```

- Source item value: [47.21,12.17,17.75,7.32,20.13,47.38,46.96,44.89,10.36,35.09,17.9,41.02,18.46,37.78,25.22,41.29]

© Softing Industrial Automation GmbH

- Destination item value: "7.32,10.36,12.17,17.75,17.90,18.46,20.13,25.22,35.09,37.78,41.02,41.29,44.89,46.96,47.21,47.38"

5. The LUA expression determines the prime divisors [4] for the value of a scalar source item.

```
local a = <<I1>> / <Value>>

function primdiv(n)
  if n < 0 then n = -n end
  local function f(var, v)
    if n > 1 then
      while n%v > 0 do
        v = v + (v == 2 and 1 or 2)
        if v*v > n then v = n end
      end
      n = n / v
      return v
    end
  end
  return f,nil,2
end

local output
function createString(in1, output)
  if output == nil or output == '' then
    output = string.format("%d", in1)
  else
    output = string.format("%s,%d", output, in1)
  end
  return output
end

for p in primdiv(a) do
 output = createString(p, output)
end

return output
```

- Source item value: 90
- Destination item value: "2,3,3,5"

6. String concatenation can be done by using the . . operator. Lua will automatically convert any numeric operand to string[5].

```
"inputs: " .. tostring(<<date_time_value>> / <Value>>) .. "," .. <<float_value>
```

- Source item value: "11/14/2023 11:22:33", 3.1415, 12345, "example"
- Destination item value: "inputs: 45244.473993056,3.1414999961853,12345,example"

References:

[1] Statistics/Basic, Rosetta Code, https://rosettacode.org/wiki/Statistics/Basic#Lua

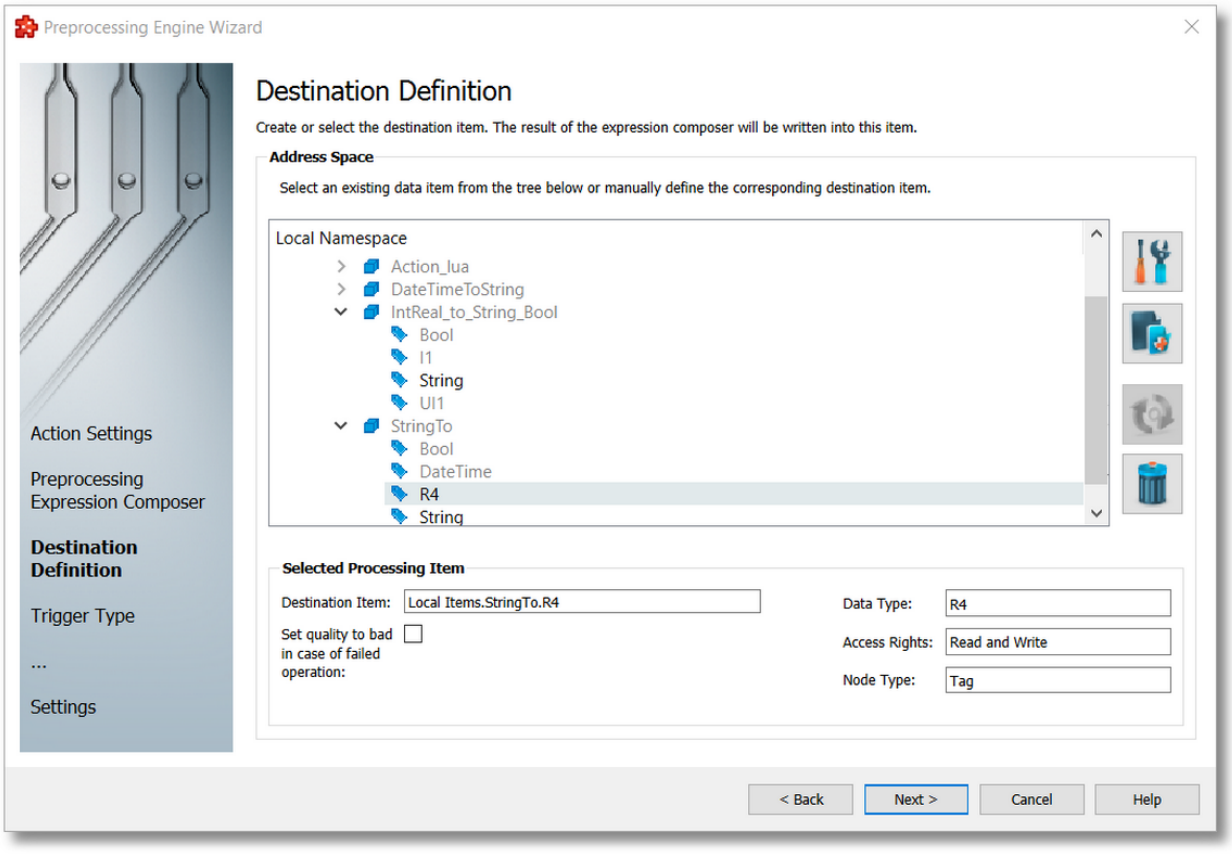
[2] Cumulative standard deviation, Rosetta Code, https://rosettacode.org/wiki/Cumulative_standard_deviation#Lua

[3] Sorting Algorithms/Circle Sort, Rosetta Code, http://rosettacode.org/wiki/Sorting_Algorithms/Circle_Sort#Lua

[4] Number Theory Example, Lua Users Wiki - Lua programming language information and resources, http://lua-users.org/wiki/IteratorsTutorial

[5] Programming in Lua - Concatenation, https://www.lua.org/pil/3.4.html

The result of the evaluated and processed expression could be stored into an internal local item (later being accessible from any available data destinations) or directly into a writable item from the available address space.



Each time when a preprocessing engine action is executed, the following process is done:

1. The placeholder is replaced with the cached value of each item used inside the action as placeholder (all these items are part of a set of internal subscriptions which collects data with requested frequency and store them into on the item cache storage.

2. The result expression (with the placeholders replaced with cached values) is validated by LUA library. If the validation fails then the complete action is marked as invalid, the information is logged into the logging file and execution of current action is stopped.

© Softing Industrial Automation GmbH

3. The validated expression is evaluated by the LUA library (based on LUA syntax). The evaluated result is stored under an internal cache of the actual action.

4. The result of the evaluated expression is tried being written into the selected destination item.

Attention: The write operation is based on OPC Technology, it can be the situation that the same value is tried being written for several times... this operation will succeed as write operation but will not generate a new data-change - which could be used in other destination. Basically, in case that the result of the evaluation remain constant than this is not generated a new data-change which and the action execution remain not visible to a connected OPC Client (including the internal OPC Client used by other data destination).

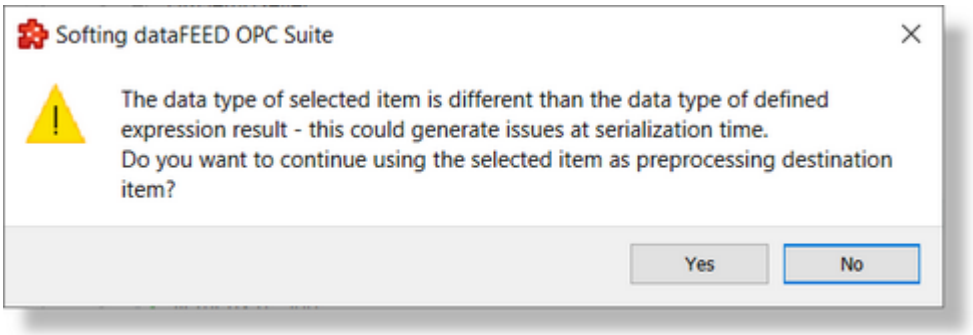
The usage of destination item is filtered based on data-types of items from the address space - each operation types supports (as destination item) a limited set of data-types:

- Mathematical Operation:
  - I1 (8 bit integer), I2 (16 bit integer), I4 (32 bit integer), I8 (64 bit integer)
  - UI1 (8 bit unsigned integer), UI2 (16 bit unsigned integer), UI4 (32 bit unsigned integer), UI8 (64 bit unsigned integer)
  - R4 (32 bit float), R8 (64 bit double)
  - Decimal
  - Variant (Any)
  - BSTR (String)
- Logical Operation:
  - Bool (Boolean)
  - I1 (8 bit integer), I2 (16 bit integer), I4 (32 bit integer), I8 (64 bit integer)
  - UI1 (8 bit unsigned integer), UI2 (16 bit unsigned integer), UI4 (32 bit unsigned integer), UI8 (64 bit unsigned integer)
  - Variant (Any)
  - BSTR (String)
- Data Type Conversions:
  - I1 (8 bit integer), I2 (16 bit integer), I4 (32 bit integer), I8 (64 bit integer)
  - UI1 (8 bit unsigned integer), UI2 (16 bit unsigned integer), UI4 (32 bit unsigned integer), UI8 (64 bit unsigned integer)
  - R4 (32 bit float), R8 (64 bit double)
  - Decimal
  - Variant (Any)
  - BSTR (String)
- LUA Engine:
  - I1 (8 bit integer), I2 (16 bit integer), I4 (32 bit integer), I8 (64 bit integer)

- UI1 (8 bit unsigned integer), UI2 (16 bit unsigned integer), UI4 (32 bit unsigned integer), UI8 (64 bit unsigned integer)
- R4 (32 bit float), R8 (64 bit double)
- Decimal
- Variant (Any)
- BSTR (String)

Attention: Boolean destination items for Logical Operations must have equivalence operators (==, ~=) in the expression.

The user could decide to ignore the filtering, in this situation the application will notify about the risks and will accept it - in this situation it will be tried a conversion to the item data type before trying to write the result into the specific destination item.



Attention: The result of the evaluated expression could be possible to not be convertible to destination item data-type. In this scenario, the write operation will failed and the action will be marked as not executed.

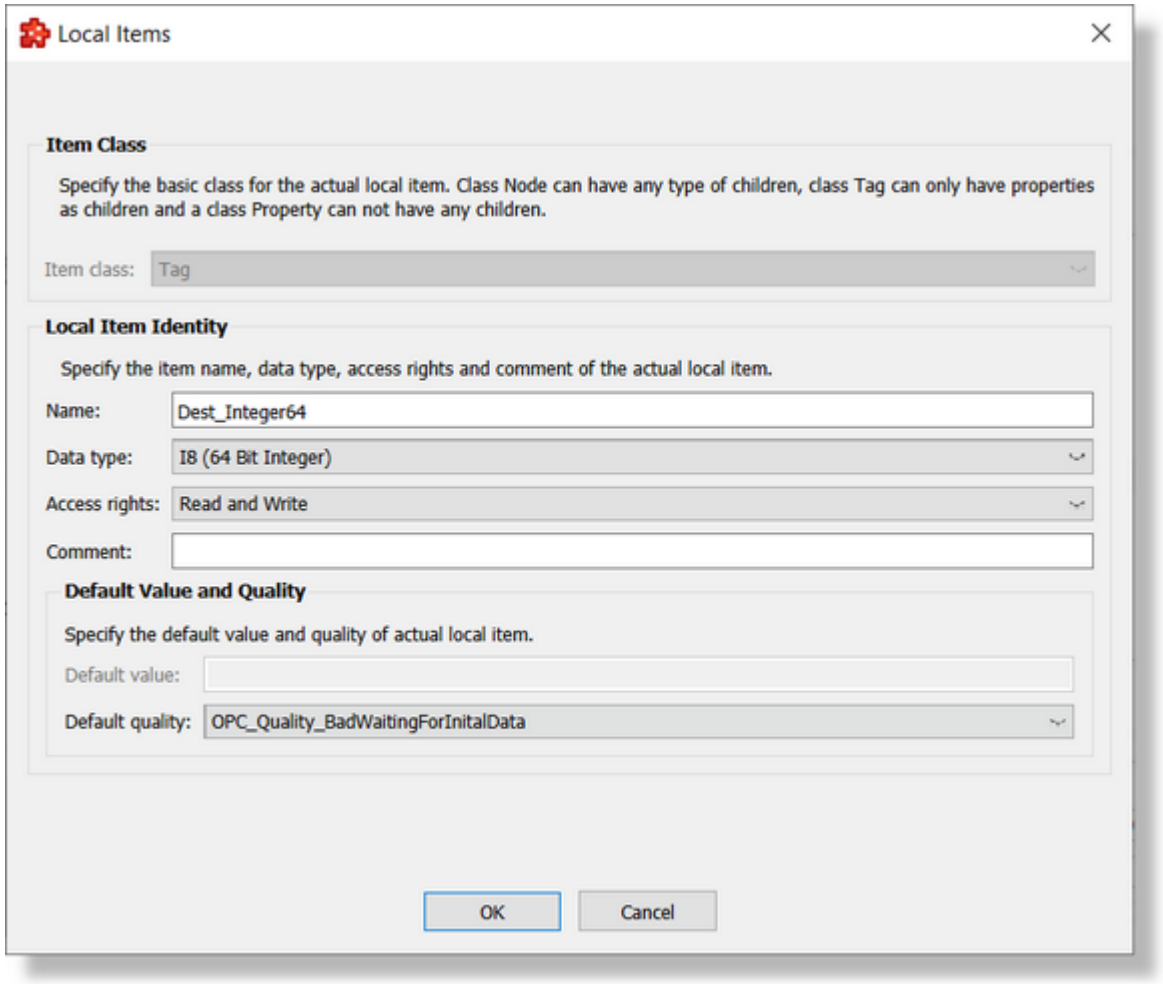
Attention: Even if all the variants (placeholders) contains items based on supported data types, the result of the evaluated expression could have a data-type different than the expected one which can generate failures on writing it to destination item. Sample: division of 2 unsigned integer values could have as result a float value or a not supported number (in case of division by 0 operation)... basically this kind of situations could not be statically evaluated at the configuration time due the reason that it is dependent of the value of of used placeholders which are evaluated based on current values at runtime.

Note: The entire evaluation of expression defined in the current action is done based on LUA library. The result of evaluated expression is predictable only of values involved in the composed expression are also predictable.

If the result of an expression evaluation is correct then the destination item is written using current time as timestamp and "StatusCode_Good/OPC Quality Good (192)" as status code/quality. If the expression evaluation failed and the option "Set quality to bad in case of failed operation" is checked then the "StatusCode_Bad/OPC Quality Bad (0)" is written.

© Softing Industrial Automation GmbH

By default, the proposed destination item from a preprocessing engine action is a local item. It is allowed to define new local items or to manipulate the local items address space directly from the address space tree visible current wizard page. Using the  ,  and  buttons it is allow to alter the address space. Also, the address space of other configured data sources can be changed directly from the address space tree used for selection of data destination item.

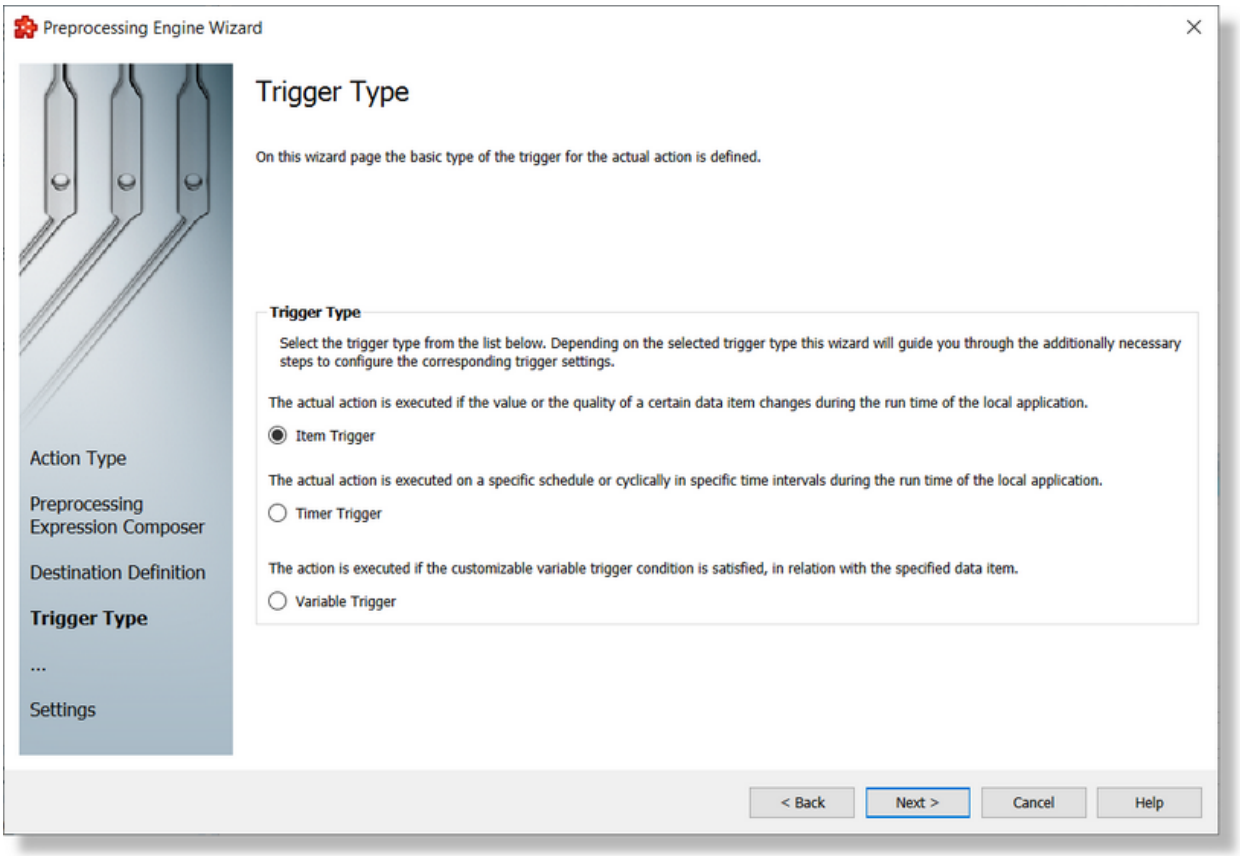


Attention: The destination item should have writable access rights. Also, please be aware about destination item data-type should should match with the possible data-type of evaluated results of expression composer. Defining a new destination item (as Local Item) be correctly defined (based on data-type and access rights).

<<< Preprocessing Engine Action Wizard - Expression Composer Preprocessing Engine Action Wizard - Trigger Type >>>

The Trigger Type wizard page provides the possibility to select the trigger type for the actual action. The following trigger types are available:

- **Item Trigger** - this trigger is fired when a data change for the selected item occurred. A data change means that the value or the quality of the item changed.
- **Timer Trigger** - this trigger is fired at a specified time or time interval.
- **Variable Trigger** - this trigger is fired when a data change for the items in the variable conditions occurs. A data change means that the value or the quality of the item changed.



To proceed with the wizard and continue to define the selected type of trigger use the *Next* button, to go back to the previous wizard page use the *Back* button, or discard the wizard by pressing the *Cancel* button.

Depending on the selected *Trigger Type* the next wizard page will be:

- Trigger Wizard - Trigger Item - if the selected trigger type is Item Trigger,
- Trigger Wizard - Timer Selection - if the selected trigger type is Timer Trigger
- Trigger Wizard - Variable Trigger - if the selected trigger type is Variable Trigger

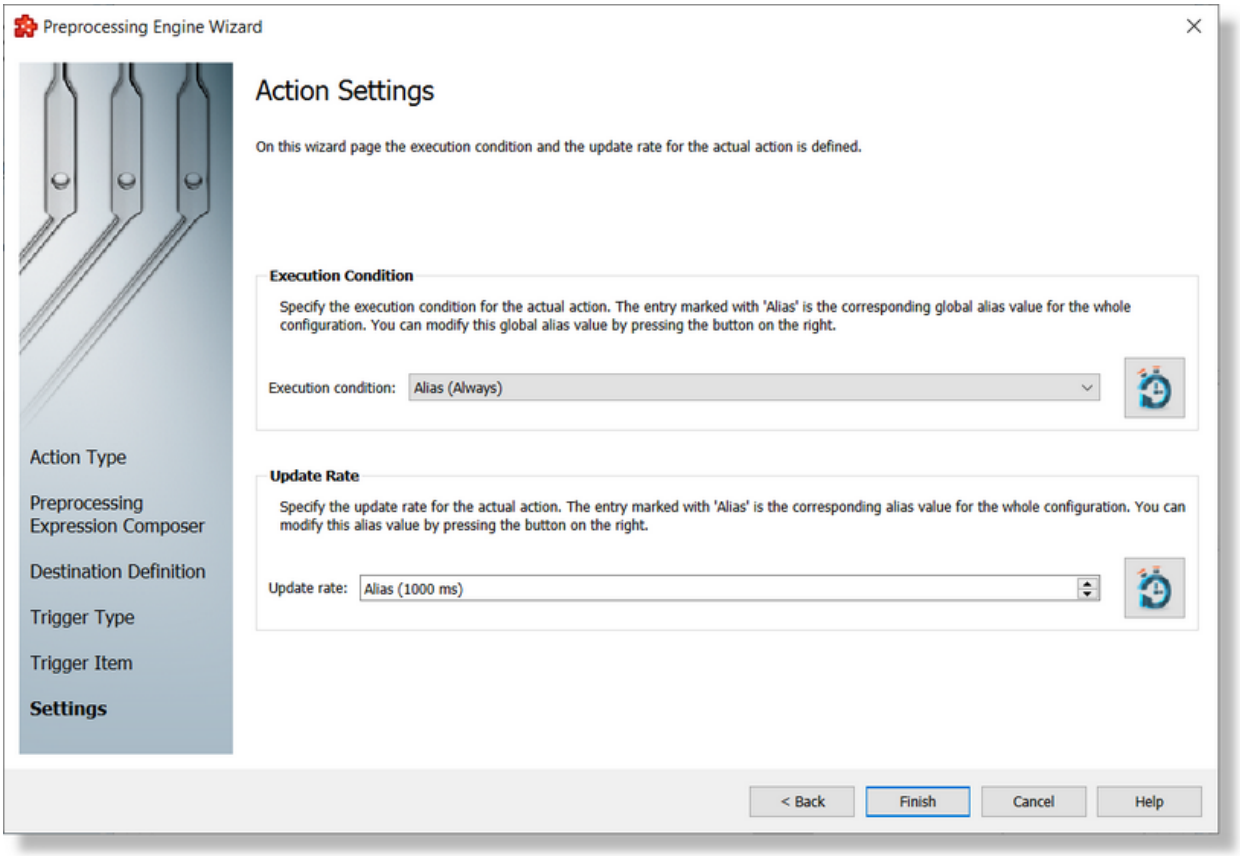
The description of the Trigger Wizard will continue with the wizard pages for configuring an *Item Trigger*.

Attention: The trigger only initiates the preprocessing action execution. The action will be subsequently validated, the expression itself will be also validated based on LUA syntax and validation/evaluation rules and later the evaluated result will be written into the destination item... in case that one of the previously described steps will be invalidated then the action could be invalidated and might not be executed after all.

© Softing Industrial Automation GmbH

<<< Preprocessing Engine Action Wizard - Destination Definition Preprocessing Engine Action Wizard - Action Settings >>>

The Action Settings wizard page allows defining the execution condition and the subscription update rate for the current preprocessing engine action.



This wizard page contains two sections: *Execution Condition* and *Update Rate*.

Execution Condition

Each statement has an execution condition which is evaluated by the local dataFEED OPC Suite application. The action is only performed when the evaluation is successful. If the evaluation fails then the statement is ignored.

The execution conditions allow reacting to server communication errors, executing special statements or preventing the execution of other statements.

The execution condition can be selected directly from the drop down combo box. The initial value is the global alias one. Changing the global default execution condition can be done using

the Configure global alias and initial values button  or pressing the keyboard shortcut Ctrl+D.

Update Rate

The *Update Rate* specifies the execution rate of the current preprocessing engine action.

The update rate can be typed or selected from the spin box. The initial value is the global alias one. Changing the global default update rate can be done using the Configure global alias and initial values button  or pressing the keyboard shortcut Ctrl+D.

Complete the *Preprocessing Engine Action Wizard* pressing the *Finish* button, review the past wizard pages using the *Back* button, or cancel the wizard thus discarding all the changes by pressing the *Cancel* button.

<<< Preprocessing Engine Action Wizard - Trigger Type

7.3.2.3.3 Data Values Placeholder Wizard

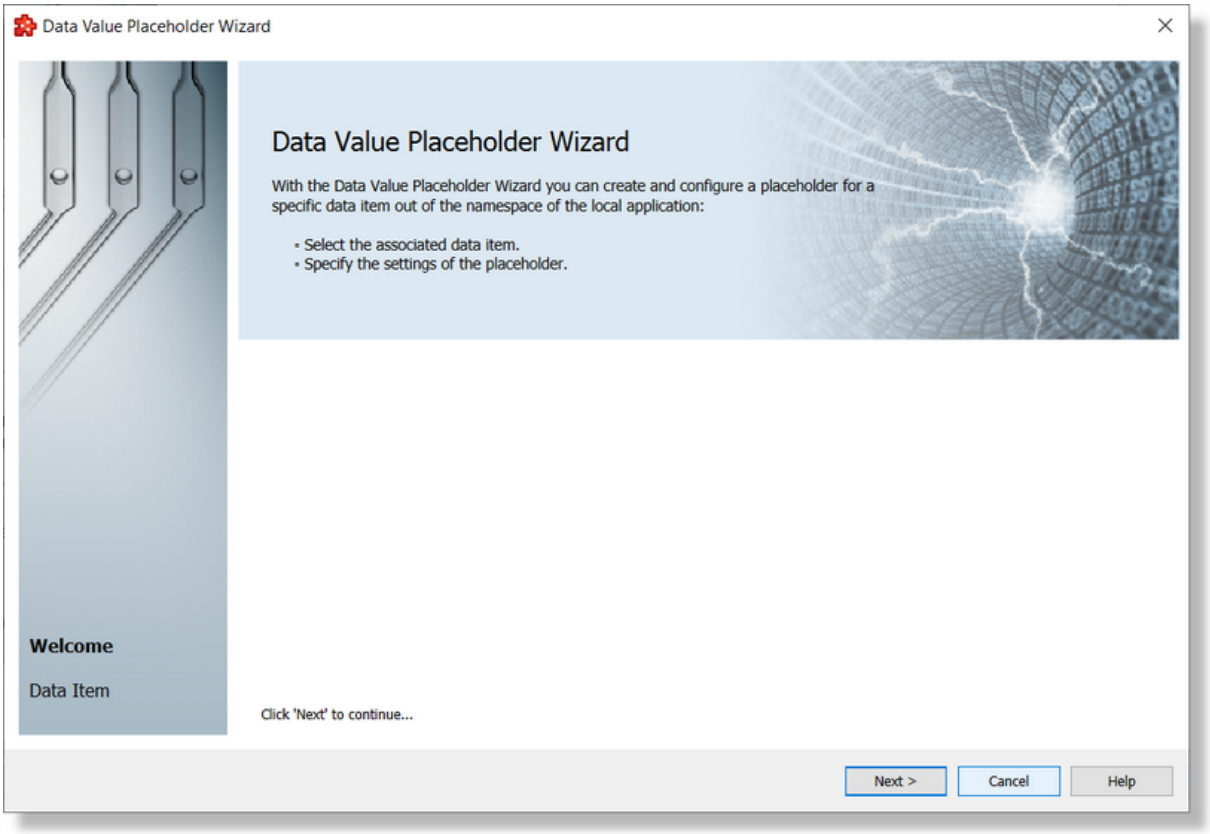
▾ With the Data Value Placeholder Wizard you can create a placeholder from a configured data source:

- Obtain a general description of the wizard @ Data Value Placeholder Wizard - Welcome
- Select a placeholder item @ Data Value Placeholder Wizard - Data Item

The Welcome wizard page describes the individual steps for defining the placeholder:

- Select an item from the local dataFEED OPC Suite application namespace or provide a valid wildcard.
- Choose the data value and the data source for the selected item.

© Softing Industrial Automation GmbH

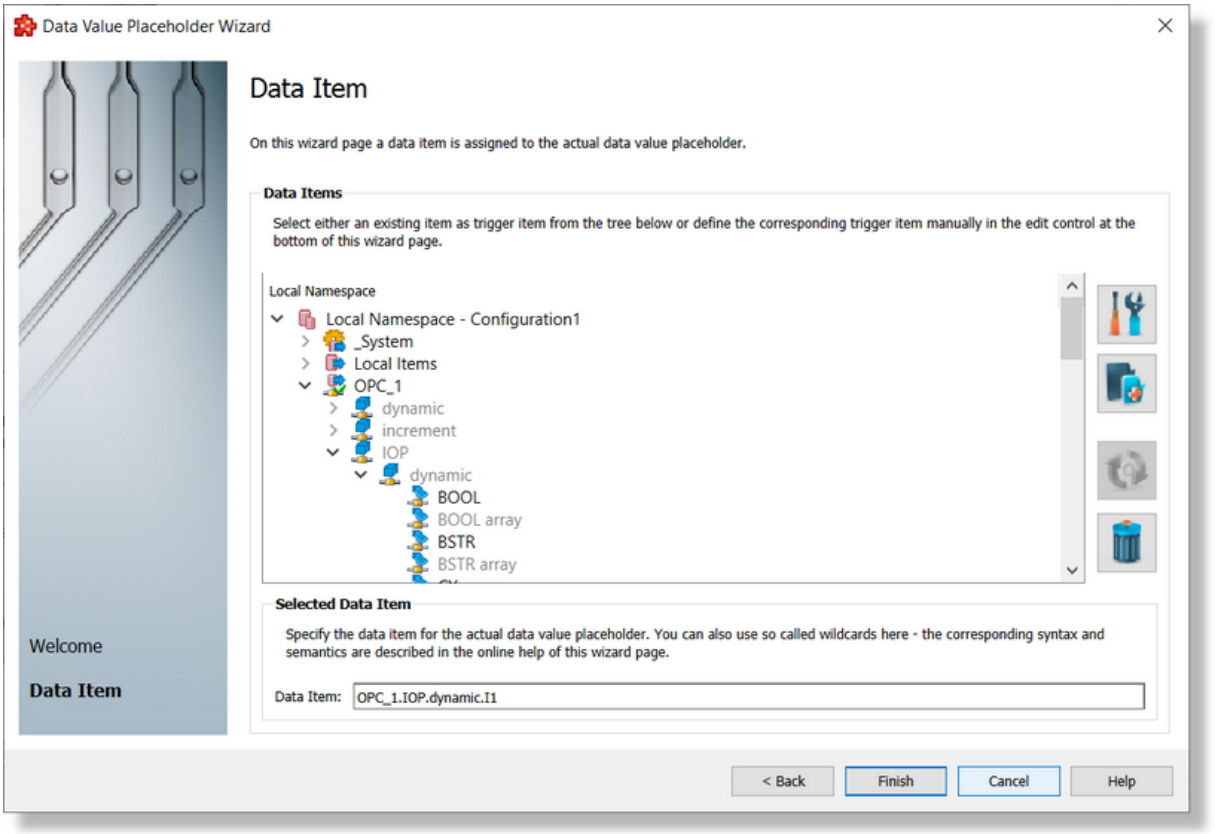


Continue to the next wizard page using the *Next* button or quit the wizard pressing the *Cancel* button.

Data Value Placeholder Wizard - Data Item >>>

The Data Item wizard page allows selecting the placeholder item. The placeholder item can be:

- Any item originating from a *Server Connection*, from the *Local Items* or from the *System Items* can be selected, choosing a tag from the *Local Namespace*.



Local Namespace Tree Handling

The *Local Namespace* tree offers the possibility to explore the entire namespace of the local dataFEED OPC Suite application in order to select a data item. After the selection is made, the *Data Item* edit box is populated with the selected item ID.

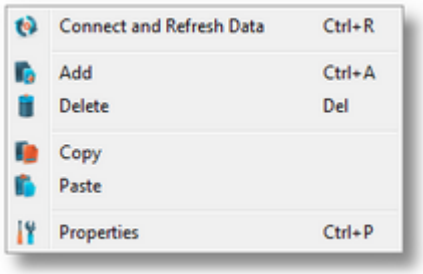
The entire namespace of the local dataFEED OPC Suite application can be managed through this wizard page by performing certain operations using the buttons on the right, the context sensitive menu or the keyboard shortcuts:

- Open the item properties dialog using the *Properties of the selected item* button  , pressing the keyboard shortcut CTRL+P or selecting the *Properties* action from the context menu. Depending on the type of the selected item, a special Item Properties dialog opens:
  - If the selected item is under Local Items the Modify Local Item dialog is opened.
  - If the selected item is under a dynamic server connection the Modify Dynamic Server Item dialog is opened.
  - If the selected item is under a static server connection the Modify Static Server Item dialog is opened.
- Add a new item using the *Add a new item*  button, pressing the keyboard shortcut CTRL+A or selecting the *Add* action from the context menu. Depending on the type of the selected node, a special Add Item dialog opens:

© Softing Industrial Automation GmbH

  - If the selected node is under Local Items the Add Local Item dialog is opened.
  - If the selected node is under a dynamic server connection the Add Dynamic Server Item dialog is opened.
  - If the selected node is under a static server connection the Add Static Server Item dialog is opened.
- Copy the selected item to clipboard by selecting the *Copy* action from the context menu.
- Paste the previously copied item from clipboard by selecting the *Paste* action from the context menu. Items originating from the Local Items cannot be pasted under server connections, and vice-versa; an error message will be displayed in both cases. If an item is pasted in a static (filtered) server connection and the item does not exist on the server an error message is displayed.
- Connect to the server and refresh the data items by using the *Refresh data items* button, pressing the keyboard shortcut CTRL+R or selecting the *Connect and Refresh Data* action from the context menu. The namespace of the selected server connection is recreated; if manually added items not referenced in the actual configuration exist, those items are deleted from the namespace.
- Delete the selected item and all of its child items by using the Delete data items button , pressing the DELETE / DEL key or selecting the Delete action from the context menu. Items which are included in any Data Processing / Data Destination add-ons will not be deleted but an error message will be displayed instead.

An example of the context menu can be seen below:



Attention: The *Local Namespace* root node, the *Local Items* root node and the server connection root nodes cannot be deleted. An item that is used in consumers or a system item cannot be deleted either.

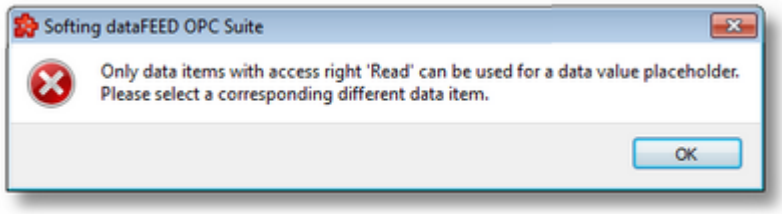
Note: The copy/paste actions work on the selected tag/property or on the whole selected branch. The copy action is not available for the Local Namespace root or for the server connection root nodes; the paste action is not available for items of type property (as they cannot have any children).

The entire namespace of the local Softing dataFEED OPC Suite application is exposed in this wizard page. The tree root node name is composed of the *Local Namespace* keyword followed by the instance name.

Selected Data Item Handling

When an item from the Local Namespace is selected, the *Data Item* edit box is populated with the item ID.

Note: Only tag and property items are allowed to be used in placeholder. If a node item is selected, an error message dialog will appear when the *Next* button is pressed.



Regular expressions with special characters (asterisk, question mark, square brackets) are not supported by this module.

Complete the wizard using the *Finish* button, go back to the previous wizard page using the *Back* button or discard any changes made in this wizard by pressing the *Cancel* button.

<<< Data Value Placeholder Wizard - Welcome

.

7.3.2.3.4 Data Values Placeholder

Placeholders can be used when configuring various preprocessing engine actions. These placeholders can be defined in the *Action Wizard,* the main user interface and are saved in the Subscription file. A placeholder is determined by the following attributes:

- Data Item - an item (tag or node with value) originating from a *Server Connection* or a *PLC Connection,* from the *Local Items or* from the *System Items*.
- Data Value - the actual data value for the associated data item which will be replaced during the run time of the local dataFEED OPC Suite application.
- Data Source - the source for the data value which will be replaced during the run time of the local dataFEED OPC Suite application.

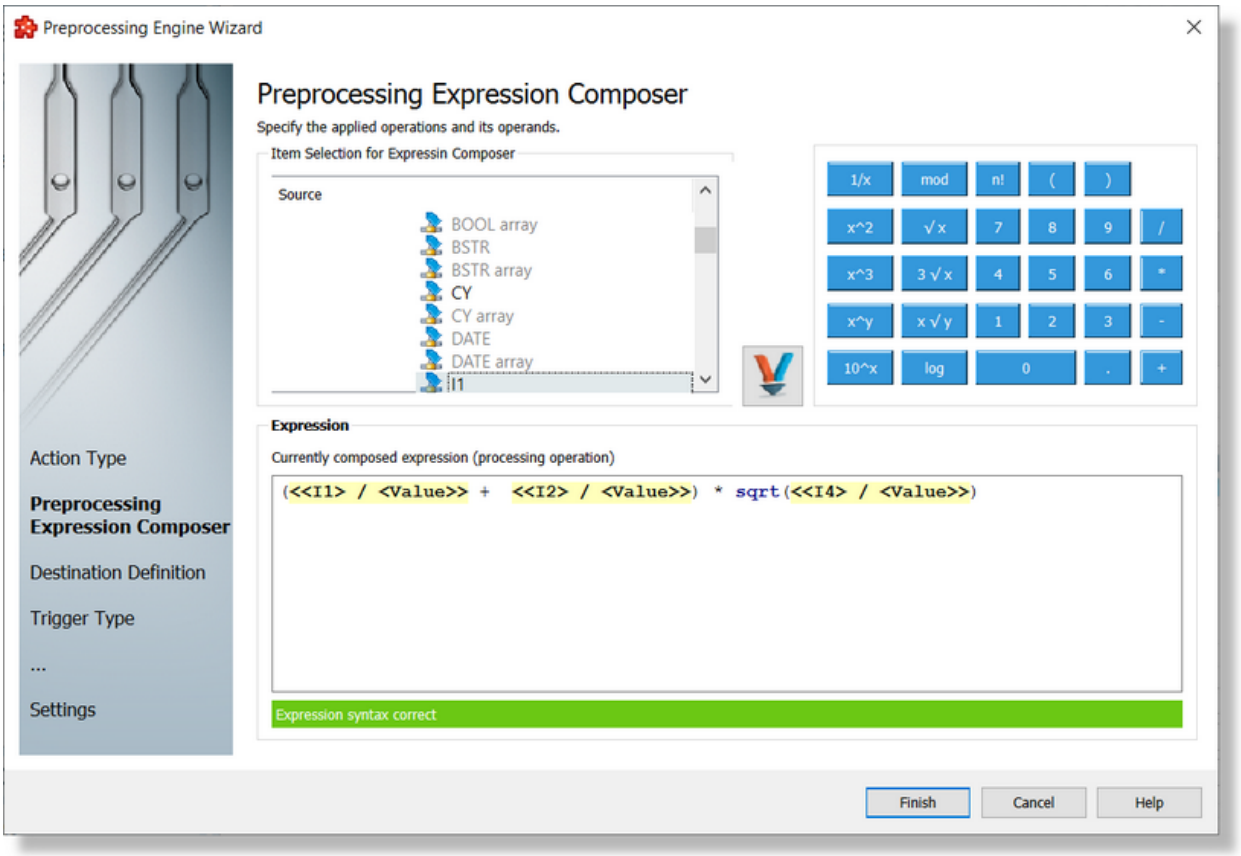
Placeholder in the user interface

A placeholder is used while configuring preprocessing actions to determine which item (available into the exposed address space) will be user at runtime as operand. On Expression Composer wizard page (while defining the action body), as operand it can be selected an item which will be stored into the action expression definition as placeholder (using the same syntax than all the other add-ons). The corresponding placeholder form is <Short Data Value> / <Short Data Item>>.

© Softing Industrial Automation GmbH

Sample:

```
<<I1> / <Value>>.
```



Placeholder in the subscription file

In the subscription file a placeholder has the following syntax:

```
#%-<Data Item>%<ItemID_SubscriptionID>%<DataSource>-%#
```

where:

Attribute	Description
#%- -%#	Represents the beginning and the end of a placeholder definition
Data Item	Represents the item data value.
%	Separates the Data Value, Data Item and Data Source fields in the placeholder.
ItemID_SubscriptionID	Represents the subscription item identifier.
Data source	Represents the item data source.

Example:

```
#%-ItemValue%OPC_1.increment.I1_Subscription_1000%Current-%#
```

Data value:

The following data value attributes can be used (they are supported by both dataFEED OPC Suite Configurator and local dataFEED OPC Suite application):

Attention: The configuration tool supports exclusively Data Value attribute. There is no possibility to select other attribute for a selected item. Usage of other attributes can be done only by manually editing the Subscription file.

Attribute	Subscription file form	Description
Data Value	ItemValue	The value of the data source server item.
Item Identified	ItemItemID	The ID of the data source server item.
Quality as Integer	ItemQuality	The data source server item quality as integer.
Quality as String	ItemQualityString	The data source server item quality as string.
Timestamp (without milliseconds)	ItemTimestamp	The data source server item timestamp as date time without milliseconds.
Timestamp (with milliseconds)	ItemTimestampMS	The data source server item timestamp as date time including milliseconds.
Timestamp (milliseconds part only)	ItemTimestampMSPart	The milliseconds part of the data source server item timestamp.
Internal Timestamp (without milliseconds)	ItemTimestampOCT	The timestamp of data change in the local dataFEED OPC Suite application as date time without milliseconds.
Internal Timestamp (with milliseconds)	ItemTimestampOCTMS	The timestamp of data change in the local dataFEED OPC Suite application as date time including milliseconds.
Internal Timestamp (milliseconds part only)	ItemTimestampOCTMSPart	The milliseconds part of the data change timestamp in the local dataFEED OPC Suite application.

The following data value attributes can be used (only supported by the local dataFEED OPC Suite application):

Attribute	Subscription file form	Description

© Softing Industrial Automation GmbH

Item Subscription ID	ItemSubscriptionID	The subscription ID of the item
Data Value	DataValue	The ID of the data source server item.
Data Quality as Integer	DataQuality	The data source server item quality as integer.
Data Quality as String	DataQualityString	The data source server item quality as string.
Timestamp (without milliseconds)	DataTimestamp	The data source server item timestamp as date time without milliseconds.
Timestamp (with milliseconds)	DataTimestampMS	The data source server item timestamp as date time including milliseconds.
Internal Timestamp (without milliseconds)	DataTimestampOCT	The timestamp of data change in the local dataFEED OPC Suite application as date time without milliseconds.
Internal Timestamp (with milliseconds)	DataTimestampOCTMS	The timestamp of data change in the local dataFEED OPC Suite application as date time including milliseconds.
Data Item ID	DataItemID	The value of the data source server item.

Attention: There is a significant difference between the Data Values in the tables above. The values in the second table are taken directly from dataFEED OPC Suite internal server while the values for the Data Values from the first table are taken directly from the server which holds the real Item. dataFEED OPC Suite holds only a copy of that Item.

Data Source or Item type:

One of the following Data Sources may be selected:

Attention: The configuration tool supports exclusively Current item type. There is no possibility to select other attribute for a selected item. Usage of other attributes can be done only by manually editing the Subscription file.

Item type	Description
Current	The item must be in the subscription item list. The value used is the current one, which means that if the value for the item is included in the current transaction, this value will be used. If the value is not included in the incoming transaction, the cached value will be used.
Cache	The item must be in the subscription item list. The value used is the cached one, even if that item is included in the incoming transaction.

Changed	The item must be in the subscription item list. The value used is the changed one, which means that the entire action will be invalidated if the item is not present in the incoming transaction.

Examples:

```
'#%-DataValue%OPC_1.increment.I8_Subscription_1000%Current-%#'
'#%-DataValue%OPC_1.increment.I8_Subscription_1000%Cache-%#'
'#%-DataValue%OPC_1.increment.I8_Subscription_1000%Changed-%#'
```

Attention: If the *Changed* Data Source is used in an action and the corresponding action trigger is a *Data Change* trigger, make sure that a transaction which contains data changes for all the items used in the current action will be generated by the server; otherwise, the entire action will be invalidated if an item has no associated data change in the incoming transaction.

7.3.2.3.5 Configuration Files

All the Preprocessing Engine configuration settings are stored in the following configuration file:

- Subscriptions File - which contains all the descriptions relating to the preprocessing action:
  - what data will be processed by processing engine.
  - what is the frequency of collecting data which will be used in action execution.
  - what is the frequency of executing preprocessing actions - basically when the execution of an action is triggered.
  - where to store the result of the evaluated expression from a preprocessing action.
  - convertible types and data formats.

Note: The path and file name of this configuration file is fixed and not configurable.

The subscription files are stored in the *Tasks* folder under the instance path. For subscription configuration file a text file called *Subscriptions.txt* will be created in the *Tasks* configuration folder. Although this files can be manually edited, using the dataFEED OPC Suite Configuration Tool is advisable.

Note: The recommendation is to use only the dataFEED OPC Suite Configurator to edit this file. This ensures that all the definitions will be valid.

If the configuration is manually edited try loading the configuration in the dataFEED OPC Suite Configurator as all the configuration files are automatically validated at load-time and, if there are any mistakes, an error message will be displayed. The dialog will contain an appropriate description and will point to the first invalid line of the local items configuration file.

The configuration file is plain text files encoded in UTF-8 (thus providing full UNICODE compatibility) and they have line-oriented structure. The elements of a line are tab-separated. The format of one line is the following:

```
Line type<TAB>Line data
```


© Softing Industrial Automation GmbH

Subscription File

The *Data Processing - Preprocessing Engine* generates multiple line types:

- common lines which are specific to all the consumer add-ons: condition lines, subscription lines and item lines.
- specific lines: processing action lines.

Common Lines:

The following line types are common lines and will be used to configure different functionalities:

Line Type	Description
Subscription	Defines the update rate with which the current action is collecting data (which later will be used on expression evaluation) and executing actions in case of usage of item triggers.
Condition	Defines the execution condition for the current action.
Item	This line defines a trigger which is fired if the item data has changed. The trigger data includes all other items which have changed at the same time (were reported in the same data change callback from the OPC server). An item data change consists of a different value or quality of the respective item compared to the previous one.
Timer	This line defines a trigger which is fired at specific time and date or at specific intervals.
Variable	This trigger is fired each time when the variable's condition is evaluated to a positive result.
Commented Lines (#)	Any line that starts with the "#" character is considered a comment line which is ignored during evaluation.

The subscription file is divided into two sections:

- The global area:

  The global area extends from the first subscription file line to the first trigger line (e.g: ITEM, TIMER or VARIABL lines).

  This area contains definitions valid for the entire subscription file.

- The local area:

  The local area covers all the actions between two trigger lines in the subscription file.

The lines in this area are only executed if the parent trigger fires.

Subscription Line

A subscription line has the following format:

```
SUBSCRIPTION<TAB>Subscription ID<TAB>UPDATERATE<TAB>Update rate value<TAB>ACTIVE<TAB>
```

The subscription line contains the following configurable parameters:

Parameter	Description
SUBSCRIPTION	The type of the line.
Subscription ID	It is used to identify the subscription and must be unique.
UPDATERATE	Predefined keyword.
Update rate value	Defines the update rate with which the current action will be executed.
ACTIVE	Predefined keyword.
Active value	Active value which can be TRUE or FALSE.

The subscription lines always contain the global subscription line as well as all subscriptions which are used at least once in consumer actions. The global subscription is represented by the subscription line with the empty *Subscription ID*. The default update rate value for the global subscription is *1000* milliseconds.

The value of the update rate is saved in milliseconds. This value may not be set to 0 inside the dataFEED OPC Suite Configurator but it can be manually set by editing the subscription file with an external text editor. An update rate value of 0 means *as fast as possible* but this is not recommended from a resource-related point of view.

Setting a subscription to be active or inactive is an advanced feature which cannot be configured from within the dataFEED OPC Suite Configurator. By default the active value for each and every subscription is set to *TRUE* (meaning that all subscriptions are active since creation). If a subscription's active value is set to *FALSE* then the data changes of the items configured to be part of that subscription will be ignored.

Attention: From a resource-related point of view it is recommended to remove the unused subscription definitions.

Examples:

```
SUBSCRIPTION<TAB><TAB>UPDATERATE<TAB>1000<TAB>ACTIVE<TAB>TRUE
SUBSCRIPTION<TAB>Subscription_300<TAB>UPDATERATE<TAB>300<TAB>ACTIVE<TAB>TRUE
```

Condition Line

© Softing Industrial Automation GmbH

A condition line defines whether an action below an item trigger will be executed or not (when the item trigger is fired). This can be used in other configuration lines as well to condition the execution of an item action. One example of this is the conditional execution of a SQL statement in the Database add-on. A condition line has the following format:

```
CONDITION<TAB>Condition ID<TAB>Condition Statement
```

The condition line contains the following configurable parameters:

Parameter	Description
CONDITION	The type of the line, predefined keyword.
Condition ID	Unique condition identifier.
Condition statement	Single condition statement.

Conditions can be used to react to communication errors. Various actions can be executed depending on the quality of items.

The following condition statements are supported:

Condition	Description
AllGoodQuality	The trigger actions are executed only if all items in the transaction have the quality *Good*.
Always	The trigger actions are executed always, ignoring the quality of the items in the incoming transaction.

The condition lines always contain the global condition line as well as all conditions which are used at least once in consumer actions. The global condition is represented by the condition line with the empty *Condition ID*. The default condition statement value for the global condition is *Always*. As a general rule all the condition lines need to be declared in the global section of the subscription file. If no condition is defined the Always condition is applied as default.

Examples:

```
CONDITION<TAB><TAB>Always
CONDITION<TAB>AllGoodQuality<TAB>AllGoodQuality
```

Item Line

An item line is used to add an item to a subscription. The associated actions are executed each time the value or the quality of the specified data items changes during the run time of the local dataFEED OPC Suite application.

An item line has the following format:

```
ITEM<TAB>Subscription Item ID<TAB>Server Item ID<TAB>Subscription ID
```

The item line contains the following configurable parameters:

Parameter	Description
ITEM	The type of the line, predefined keyword.
Subscription Item ID	Unique subscription item identifier (implicitly created by concatenating the Server Item ID and the Subscription ID with the underscore character in between them).
Server Item ID	Item identifier within the local dataFEED OPC Suite application address namespace. This could be a server connection item ID or a local item ID.
Subscription ID	Subscription identifier (defined above, see Subscription Line).

This trigger is fired if the item data or quality has changed (also referred to as data change). The trigger data includes all other items which were included in the data change (items which had their data change reported in the same transaction).

Note: A single line consisting of the *Server Item ID* and *Subscription ID* is added to the subscription file for an item used several times with the same update rate in any consumer add-ons. This optimizes the local dataFEED OPC Suite application throughput.

Example:

```
ITEM<TAB>increment.I1_Subscription_300<TAB>increment.I1<TAB>Subscription_300
```

Note: The subscription item ID can be empty. In this case, however, only the Server Item ID can be used to identify the item.

Timer Line

This trigger is fired at a specified time and date or at specific intervals. The associated actions are executed on a specific schedule or periodically during the run time of the local dataFEED OPC Suite application.

The format of the *Timer Line* is:

```
TIMER<TAB>Timer ID<TAB>INTERVAL<TAB>Rate in milliseconds
TIMER<TAB>Timer ID<TAB>SCHEDULE<TAB>Schedule range<TAB>First execution<TAB>Rate<TAB>L
```

The interval timer contains the following configurable parameters:

Parameter	Description

© Softing Industrial Automation GmbH

TIMER	The type of the line, predefined keyword.
Timer ID	Unique timer identifier.
INTERVAL	Denotes an interval timer trigger, predefined keyword.
Interval value	Defines the interval in milliseconds at which the timer will trigger the associated action(s).

The timer schedule contains the following configurable parameters:

Parameter	**Description**
TIMER	The type of the line, predefined keyword
Timer ID	Unique timer identifier.
SCHEDULE	Denotes a scheduled timer trigger, predefined keyword.
Schedule range	Defines the range of the schedule (see below the table).
First execution	Defines the time the timer should begin firing events (see below the table).
Rate	Defines the rate at which the timer should trigger the associated action(s).
Last execution	Defines the time the timer should stop firing events (see below the table).
List of Days	Should be used only with the Week and Month schedule. It must contain the days on which the execution is desired. If the list of days is not present, the obtained functionality is the same as by using the day marker in every day of the week/month.

The timer's *Schedule range* may have one of the following values:

- MINUTE
- HOUR
- DAY
- WEEK - List of days: 1d = Monday, 2d = Tuesday, …, 7d = Sunday.
- MONTH - List of days: 1d = First day of month, 2d = Second day of month, …, 30d = Thirtieth day of month, 31d = Thirty-first day of the month.

The *First execution* defines the time the timer should begin firing events.

Example:

```
1d13h10m15s
```

Note: If parts of the execution time are 0, they do not have to be in the string, e.g. 1d0h10m0s = 1d10m

The timer *Rate* defines the event firing rate. An empty timer *Rate* means a single execution.

Last Execution: The last execution time of the schedule. Could be empty.

List of Days: Shall be used only with the Week and Month schedule. It should contain the days on which execution is desired. If the list is not specified the same functionality can be achieved by using the day marker in the first and last execution time.

Note: If the List of Days and the day marker in first and last execution time are used in the same timer trigger the two setting types should match the marked days!

Examples:

```
# Following timer fires events every 250 milliseconds (4 times a second):
TIMER<TAB><TAB>INTERVAL<TAB>250
# Following timer fires events every 20 seconds of a minute starting at second 10 (eg
TIMER<TAB><TAB>SCHEDULE<TAB>MINUTE<TAB>10s<TAB>20s
# Following timer fires events every 4 hours of a day starting at 6:30, ending at 20:
# (e.g. 6:30, 10:30, 14:30, 18:30).
TIMER<TAB><TAB>SCHEDULE<TAB>DAY<TAB>6h30m<TAB>4h<TAB>20h
# Following timer fires events every 4 hours of a day starting at 6:30, ending at 20:
# Friday (e.g. 6:30, 10:30, 14:30, 18:30 of each day).
TIMER<TAB><TAB>SCHEDULE<TAB>WEEK<TAB>6h30m<TAB>4h<TAB>20h<TAB>1d2d3d4d5d
# Following timer fires events every 4 hours starting at 6:30, ending at 22:45 (e.g.
# 18:30, 22:30), on the 1th, 2th, 10th, 20th of every month.
TIMER<TAB><TAB>SCHEDULE<TAB>MONTH<TAB>6h30m<TAB>4h<TAB>22h45m<TAB>1d2d10d20d
# Following timer fires events every 1 hours, 3 minutes and 3 seconds starting at 00:
# 23:59:59 (e.g. 00:00:01, 01:03:04, 02:06:07 etc), on the 3rd, 6th, 9th, 12th, 15,th
TIMER<TAB>Timer_02<TAB>SCHEDULE<TAB>MONTH<TAB>1s<TAB>1h3m3s<TAB>23h59m59s<TAB>3d6d9d1
```

Specific Lines:

The following line types are Preprocessing Engine specific and will be used to configure the preprocessing engine functinality:

Line Type	Description
PROCESSING	Defines a preprocessing engine action.

The format of the PROCESSING line is the following:

```
PROCESSING<tab>ACTION_NAME<tab>ACTION_STATE<tab>CONDITION_NAME<tab>OPERATION_TYPE<tab
```

The PROCESSING line contains the following configurable parameters:

© Softing Industrial Automation GmbH

Parameter	Description
PROCESSING	The type of the line, predefined keyword.
ACTION NAME	Action name/id - string defining the name of the current action - it should be uniquely identified for all PROCESSING actions.
ACTION STATE	Action state defining if the action is active or not (inactive actions will be ignored by the runtime) Possible values: • Active (case insensitive) - default value. • Inactive (case insensitive).
CONDITION NAME	Condition name/id - string defining the condition of current action. The condition is applied to received values for all items used in expression (similar with condition evaluation of other plug-ins). Possible values: • Always (case insensitive) - default value. • AllGoodQuality (case insensitive).
OPERATION TYPE	Operation type - string defining the type of current operation. Possible values: • MathematicalFunctions (case insensitive) - default value. • LogicalFunctions (case insensitive). • BinaryFunctions (case insensitive) - not implemented on current version. • StringFunctions (case insensitive) - not implemented on current version. • DatatypeConversions (case insensitive) - not implemented on current version. • LimitFunctions (case insensitive) - not implemented on current version. • ComplexOrCombinedFunctions (case insensitive) - not implemented on current version.
EXPRESSION statement	String defining the action expression - containing operator ands operands (an operand being a constant or a defined placeholder). Validation of expression will be done by replacing the placeholder and validated it over the LUA programming language.

RESULT ON HIGH RESOLUTION	This string is specifying of a requested fallback mechanism user for avoiding limitations comming by usage of Double values on LUA algorithm should be used. Possible Values: • HighResolution (case insensitive) - fallback mechanism being applied base on additional library. • MediumResolution (case insensitive) - the result is generated based on pure LUA algorithm - default value.
DESTINATION ITEM ID	String defining the internal item id of destination. The result of the evaluated and calculated expression will be written into the destination item. Restriction: the item id should be an item already defined into the subscription file.
DESTINATION ITEM DATATYPE	String defining the native data type of the destination item. Possible Values (case insensitive): all available DA data types: "UI1", "UI2", "UI4", "UI8", "I1", "I2", "I4", "I8", "R4", "R8", "BOOL", "BSTR", "DATETIME", "DECIMAL" and "ANY"
CONVERT TO DESTINATION DATATYPE	String defining the appliance of conversion to destination native data-type (when the result of the data type of evaluated and calculated expression is different than the native data type of destination item). Possible values: • ConvertToDestinationDatatype (case insensitive) • DoNotConvertToDestinationDatatype (case insensitive) default value
LOCALIZATION	Application specific localization applied individually for each action. Possible value: • 1033 - English/International localization id (default locale id). • 1031 - German localization id.
TIME FORMAT	String defining the time format (appliance only for ComplexOrCombinedFunctions operation type) - if missing the default time format will be applied • UTC (case insensitive)- usage of UTC time format (default time format)

© Softing Industrial Automation GmbH

	• LOCAL (case insensitive) - usage of Local time format.
TIMESTAMP FORMAT	String defining the timestamp format (applicable only for ComplexOrCombinedFunctions operation type) - if missing then the default timestamp format will be applied.
TRIGGER ID	Sting defining the id of the trigger for current action - the trigger ID should be previously defined (used only by the configuration tool).
COMMENT	String defining a user comment - it should be started with "#" char.

The Expression Statement parameter configures the expression which will be evaluate by LUA when the corresponding action is executed. It contains free text and data placeholders on LUA format.

Example of PROCESSING configuration lines and the appropriate ITEM lines:

```
ITEM<TAB>DADemoSever.IOP.dynamic.I1_<TAB>DADemoSever.IOP.dynamic.I1<TAB><TAB><TAB>I1
ITEM<TAB>DADemoSever.IOP.dynamic.I2_<TAB>DADemoSever.IOP.dynamic.I2<TAB><TAB><TAB>I2
ITEM<TAB>DADemoSever.IOP.dynamic.I4_<TAB>DADemoSever.IOP.dynamic.I4<TAB><TAB><TAB>I4
#
ITEM<TAB>DADemoSever.IOP.dynamic.I8_<TAB>DADemoSever.IOP.dynamic.I8<TAB><TAB><TAB>I8
PROCESSING<TAB>Action_1<TAB>Active<TAB><TAB>MathematicalFunctions<TAB>(#%-ItemValue%D
```

In order to offer a flexible configuration for Expression Statements, placeholders can also be used. The placeholders are syntactical entities that are replaced with their associated values each time the statement is evaluated during the run time of the local dataFEED OPC Suite application (before the expression being evaluated by LUA).

Note: The *Condition ID* used in PROCESSING line definition must be defined in the current Subscription File; the default condition may also be used.

The Processing Statements will be executed on every data change event, however their execution will also depend on both the defined conditions and the type of the fired trigger.

If the statements are associated with triggers of type *Item*, the respective statements are evaluated on each item change, but are only executed if all of the following conditions are true:

- The subscription item ID of the statement has to be defined in the configuration.
- A change occurs for the respective item.
- The action condition is true.
- The placeholders in the statement can be replaced by their respective values (the changed data source may invalidate the action).

If the statements are associated with triggers of type *Timer*, the respective statements are evaluated and executed only if all of the following conditions are true:

- The subscription item ID of the statement has to be defined in the configuration.
- The action condition is true.
- The placeholders in the statement can be replaced by their respective values (the changed data source may invalidate the action).

If the statements are associated with triggers of type *Variable*, the respective statements are evaluated and executed only if all of the following conditions are true:

- The subscription item ID of the statement has to be defined in the configuration.
- The variable's condition is evaluated to a positive value.
- The action condition is true.
- The placeholders in the statement can be replaced by their respective values (the changed data source may invalidate the action).

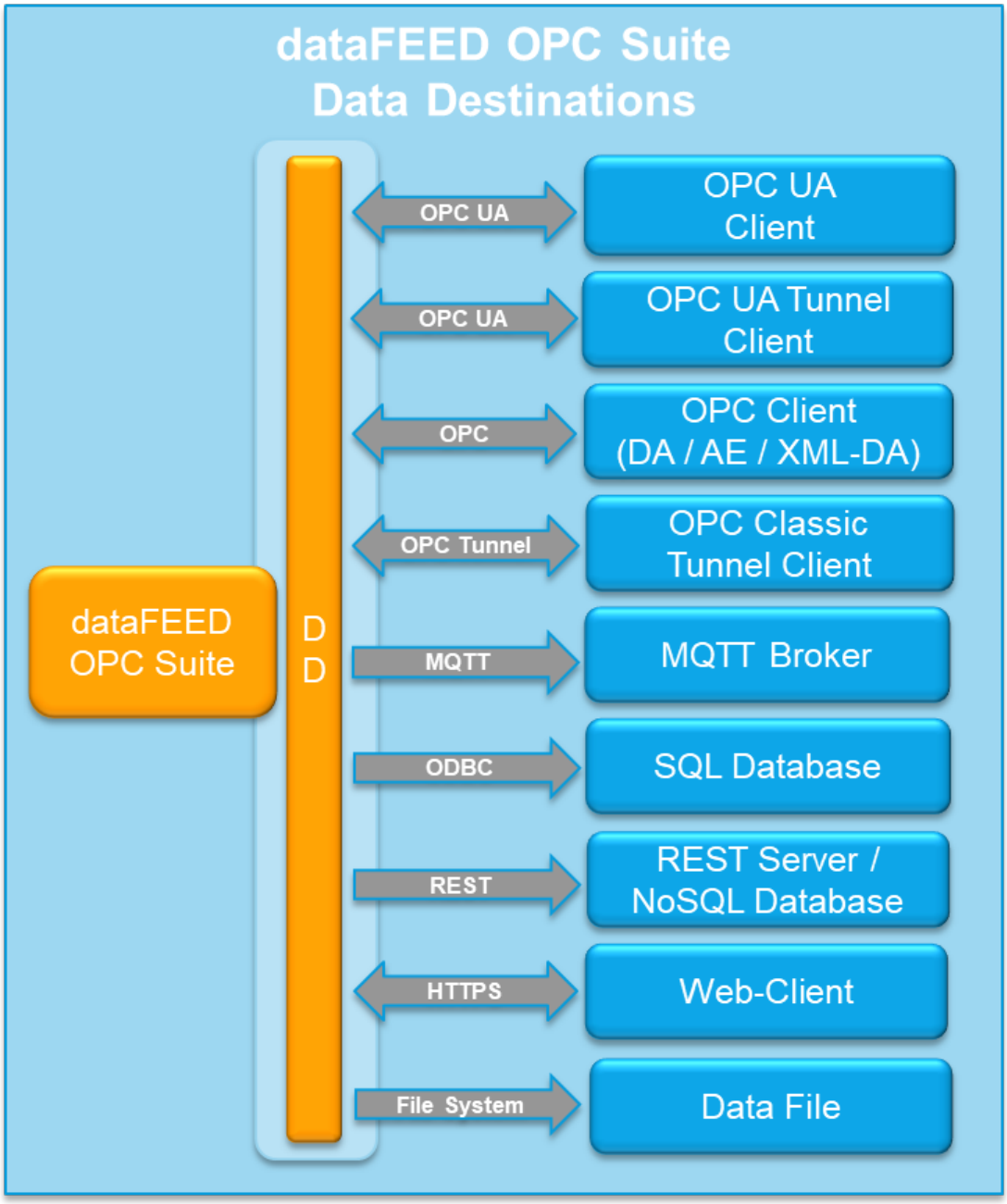
Subscription File - Example:

```
CONDITION<TAB><TAB>Always
#
SUBSCRIPTION<TAB><TAB>UPDATERATE<TAB>1000<TAB>ACTIVE<TAB>TRUE
#
MODIFIER<TAB><TAB>VQT
#
VARIABLE_TIMESTAMPFORMAT<TAB>YYYY/mm/dd HH:MM:SS
VARIABLE_TIMEFORMAT<TAB>utc
VARIABLE_LOCALE<TAB>1033
#
HYSTERESIS<TAB>DT
#
ITEM<TAB>DADemoSever.IOP.dynamic.I1_<TAB>DADemoSever.IOP.dynamic.I1<TAB><TAB><TAB>I1
ITEM<TAB>DADemoSever.IOP.dynamic.I2_<TAB>DADemoSever.IOP.dynamic.I2<TAB><TAB><TAB>I2
ITEM<TAB>DADemoSever.IOP.dynamic.I4_<TAB>DADemoSever.IOP.dynamic.I4<TAB><TAB><TAB>I4
ITEM<TAB>DADemoSever.IOP.dynamic.I8_<TAB>DADemoSever.IOP.dynamic.I8<TAB><TAB><TAB>I8
PROCESSING<TAB>Action_1<TAB>Active<TAB><TAB>MathematicalFunctions<TAB>(#%-ItemValue%D
#
ITEM<TAB>DADemoSever.IOP.dynamic.R4_<TAB>DADemoSever.IOP.dynamic.R4<TAB><TAB><TAB>R4
ITEM<TAB>DADemoSever.IOP.dynamic.R8_<TAB>DADemoSever.IOP.dynamic.R8<TAB><TAB><TAB>R8
ITEM<TAB>DADemoSever.IOP.dynamic.UI1_<TAB>DADemoSever.IOP.dynamic.UI1<TAB><TAB><TAB>U
ITEM<TAB>DADemoSever.IOP.dynamic.UI2_<TAB>DADemoSever.IOP.dynamic.UI2<TAB><TAB><TAB>U
#
ITEM<TAB>Local Items.PreprocessingEngine.DestinationItems.Dest_Real64_<TAB>Local Item
ITEM<TAB>Local Items.PreprocessingEngine.DestinationItems.Dest_UInteger64_<TAB>Local
#
TIMER<TAB>Timer_01<TAB>INTERVAL<TAB>5000
PROCESSING<TAB>Action_3<TAB>Active<TAB><TAB>LogicalFunctions<TAB>((#%-ItemValue%DADem
#
VARIABLE<TAB>Variable_01<TAB>(#%-ItemValue%DADemoSever.IOP.dynamic.R4_%Current-%# > 1
PROCESSING<TAB>Action_2<TAB>Active<TAB><TAB>MathematicalFunctions<TAB>#%-ItemValue%DA
```


© Softing Industrial Automation GmbH

7.3.3 Data Destinations

This section of the Softing dataFEED OPC Suite Online Help describes the different dataFEED OPC Suite data destinations.



▼ Softing dataFEED OPC Suite - Data Destinations

- Access of external OPC UA Clients (including OPC UA Tunneling) @ Configuration: Data Destination - OPC UA Client

- Access of external OPC Classic Clients (DA, AE or XML-DA) @ Configuration: Data Destination - OPC Client
- Access of external OPC Classic Tunnel Clients @ Configuration: Data Destination - OPC Classic Tunnel Client
- Output access to MOTT Brokers @ Configuration: Data Destination - MQTT Broker
- Archive data to SQL Databases @ Configuration: Data Destination - Database
- Archive data to REST Servers @ Configuration: Data Destination - REST Server
- Archive data to NoSQL Databases @ Configuration: Data Destination - NoSQL Database
- Archive data to a System File @ Configuration: Data Destination - File

Softing dataFEED OPC Suite V5.38
© 2024 Softing Industrial Automation GmbH

7.3.3.1 OPC UA Client

This chapter explains, which settings should be done for the data destination "OPC UA Client / OPC UA Tunnel".

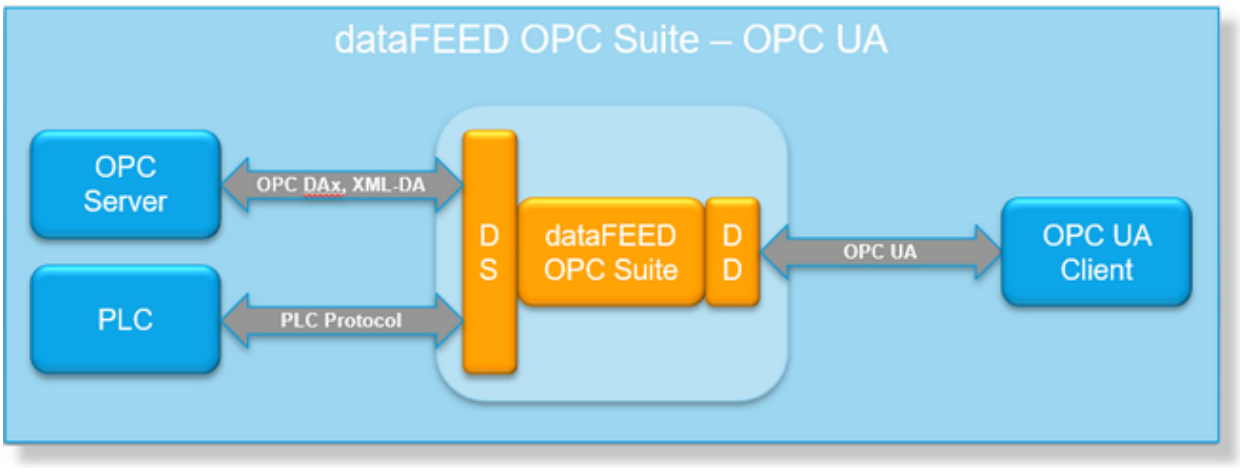
The dataFEED OPC Suite is exposing on a UA Server as data destination which enables different endpoints:

- a standard OPC UA Server Endpoint
- a customized OPC UA Server Endpoint which supports the "OPC UA Store&Forward" functionality
- an OPC UA Tunnel Server Endpoint designed for secured communication between multiple dataFEED OPC Suite instances (configurations).

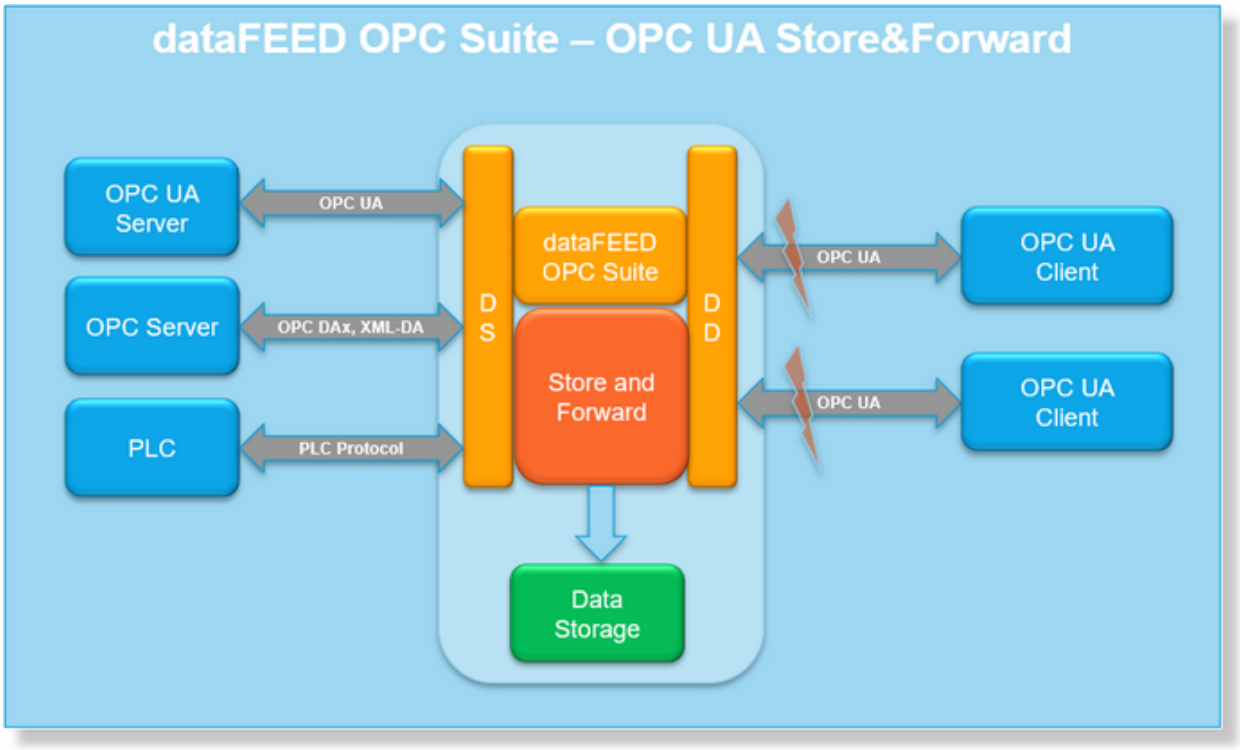
For building up a communication to an OPC UA Client at least one UA Server Endpoint should be activated (enabled).

Standard OPC UA Server Endpoint:

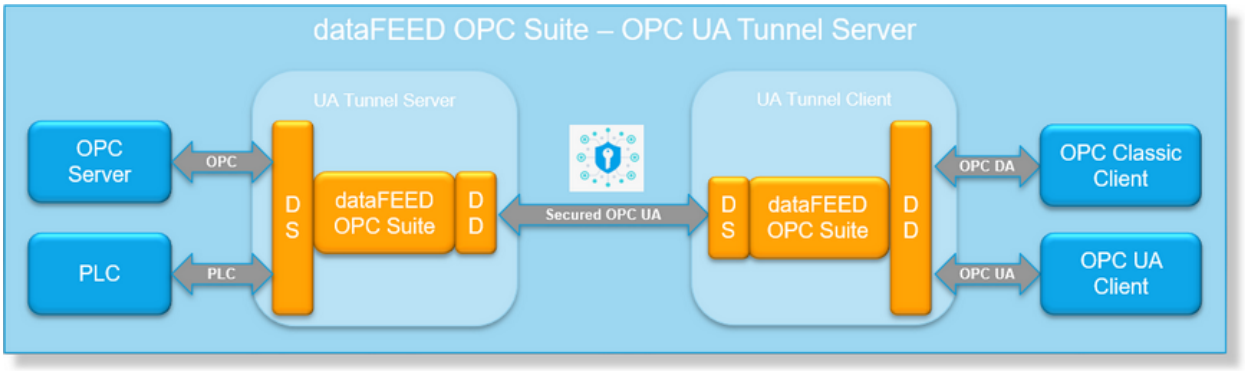
© Softing Industrial Automation GmbH



OPC UA Store&Forward Server Endpoint:



OPC UA Tunnel Server Endpoint:



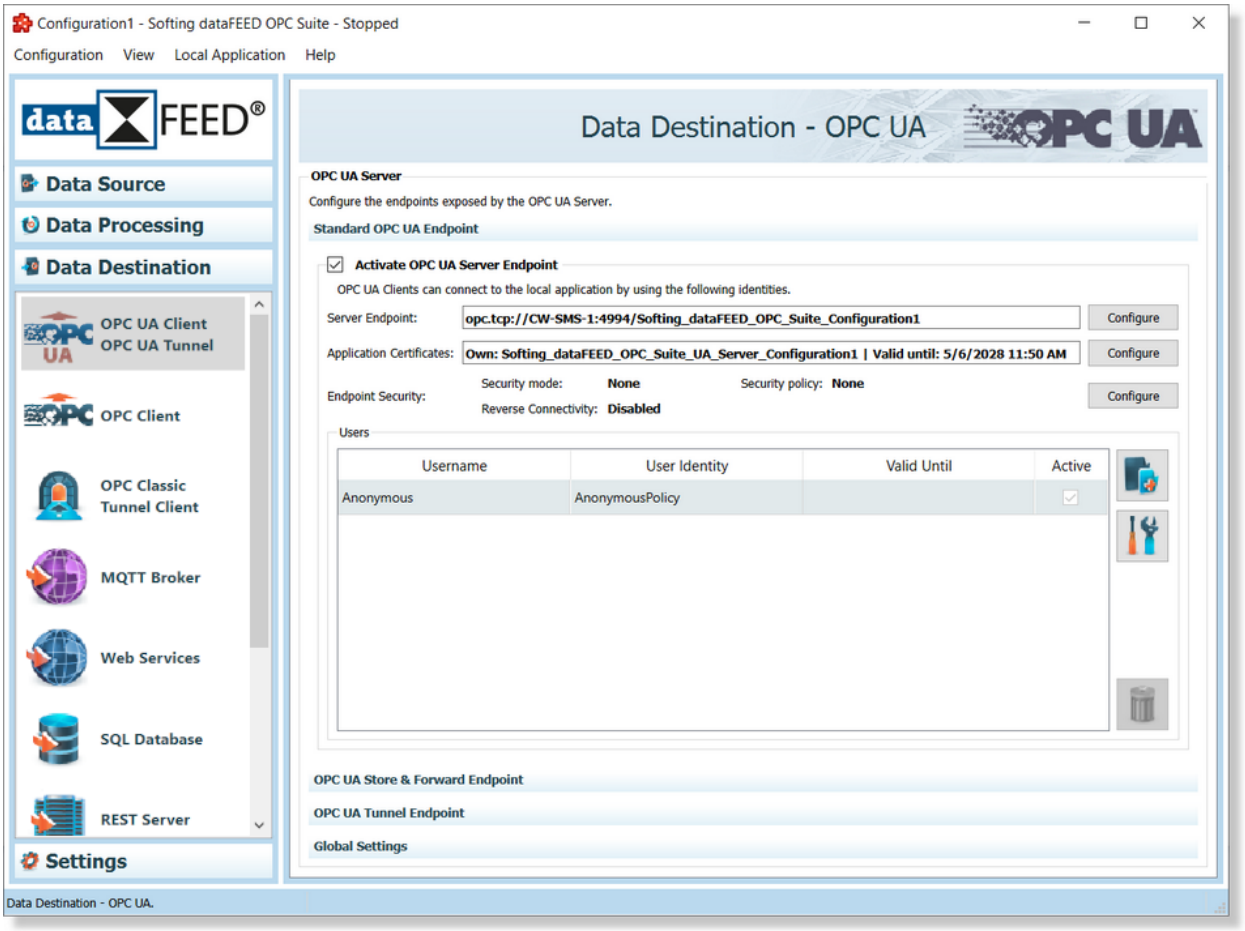
Enabling the OPC UA Client data destination functionality allows external OPC UA Clients to access the data items of the local dataFEED OPC Suite application via the OPC UA protocol.

A more detailed description of how to configure the OPC UA Client data destination functionality can be found in the Configuration Data Destination - OPC UA Client section.

7.3.3.1.1 Configuration

Note: The OPC UA functionality is deactivated by default for newly created configurations. If the OPC UA Client communication to the internal UA Server of the dataFEED OPC Suite product is intended being activated, then at least one of the exposed OPC UA Server endpoints should be activated (this can be done a enabling the "**Activate OPC UA Server Enpoint**", "**Activate OPC UA Store&Forward Server Endpoint**" or "**Activate OPC UA Tunnel Server Endpoint**" check-boxes).

© Softing Industrial Automation GmbH



The dataFEED OPC Suite UA Server is exposing (based on activated functionality) different OPC UA Server Endpoints:

- **Standard OPC UA Server Endpoint** which allows an OPC UA Client to communicate over standard (OPC UA Compliant) OPC UA Protocol with the application. The functionality is enabled by activating the "**OPC UA Server Endpoint**".

Note: This endpoint is recommended being used for Standard OPC UA Clients and standard OPC UA use-cases.

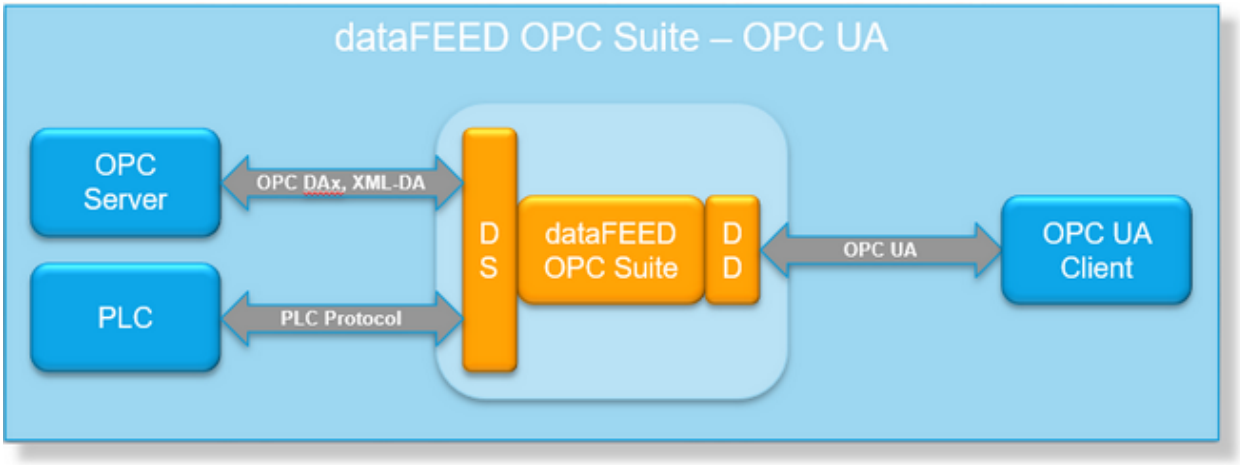
- **OPC UA Store&Forward Server Endpoint** which allows an OPC UA Client to communicate over OPC UA Protocol with the application. The communication is customized in order to be able to deliver stored data without requiring historical data access from the connected OPC UA Client. Also, this endpoint provides the capabilities to store data-values while the communication with the OPC UA Client was interrupted and delivering them later when the communication will be re-established.

Attention: This endpoint is enabling the Store&Forward functionality which is not OPC UA Compliant. Also, this endpoint enables a set of restrictions and limitations. It should be used only when the OPC UA Client is supporting the functionality and accepts the restrictions. It is not intended being used for OPC Clients which are handles online data values without interest on historical data values.

Attention: This endpoint is not supporting multiple and parallel OPC UA Client connections using the same user authentication settings.

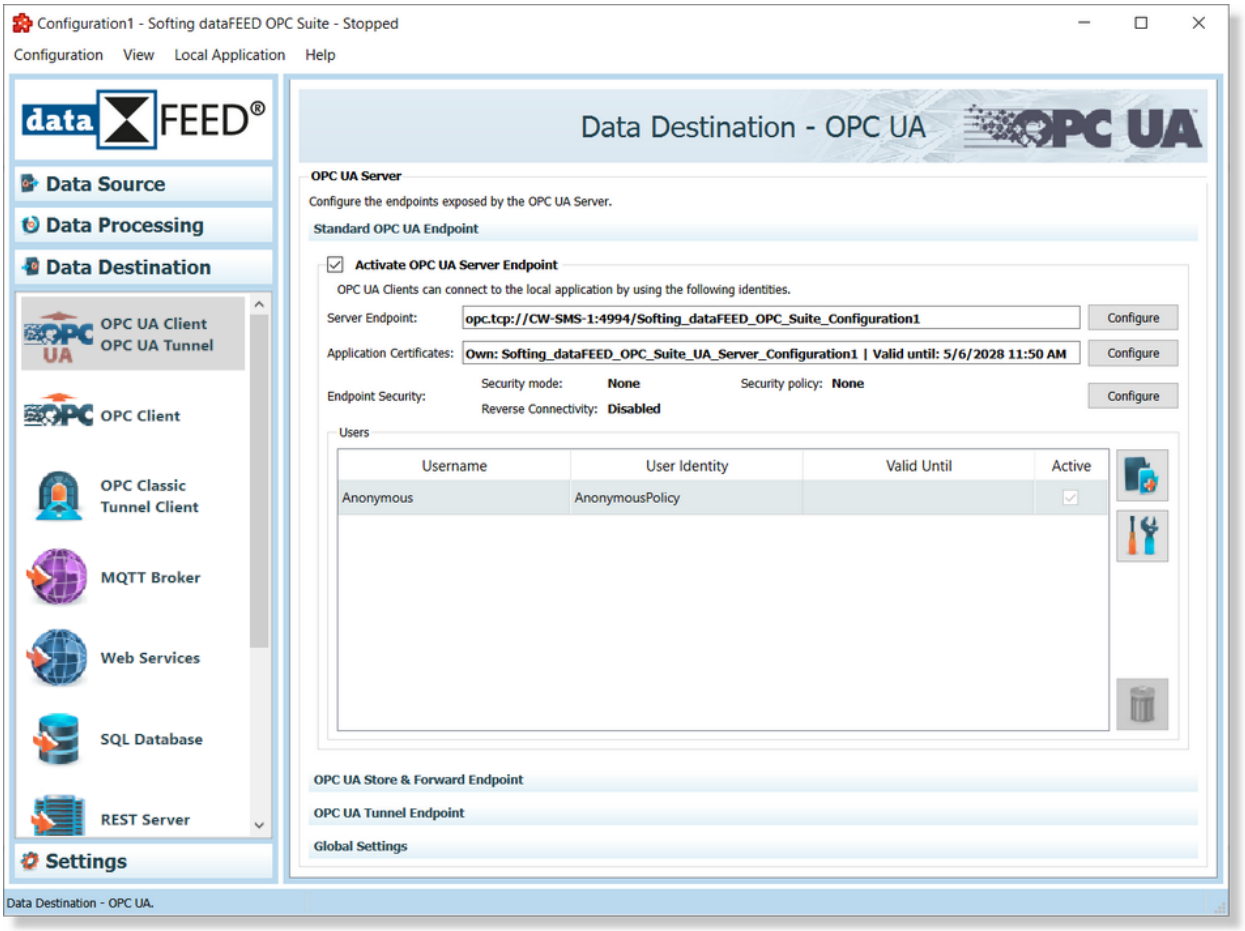
- **OPC UA Tunnel Server Endpoint** which allows an OPC UA Tunnel Client (other instance of dataFEED OPC Suite) to communicated over standard (OPC UA Compliant) OPC UA protocol with the application. The functionality is enabled by activating the "**OPC UA Tunnel Server Endpoint**".

Note: This endpoint is used for secured communication between 2 instances of dataFEED OPC Suite. The communication protocol used is standard OPC UA with a high degree of security.



The Standard OPC UA Server Endpoint is enabled by the activation of "**Activate OPC UA Server Endpoint**". By default it is disabled. Being enabled, the Standard OPC UA Server endpoint is open and accessible on the exposed OPC UA Server over the OPC UA protocol. Any OPC UA Clients will be able the establish connections to the exposed OPC Server over the exposed endpoint and operate on standard services.

© Softing Industrial Automation GmbH



Disabling all endpoints (***Standard OPC UA Server Endpoint***, ***OPC UA Store&Forward Endpoint*** and ***OPC UA Tunnel Server Endpoint***) will completely deactivate the OPC UA Server functionality.

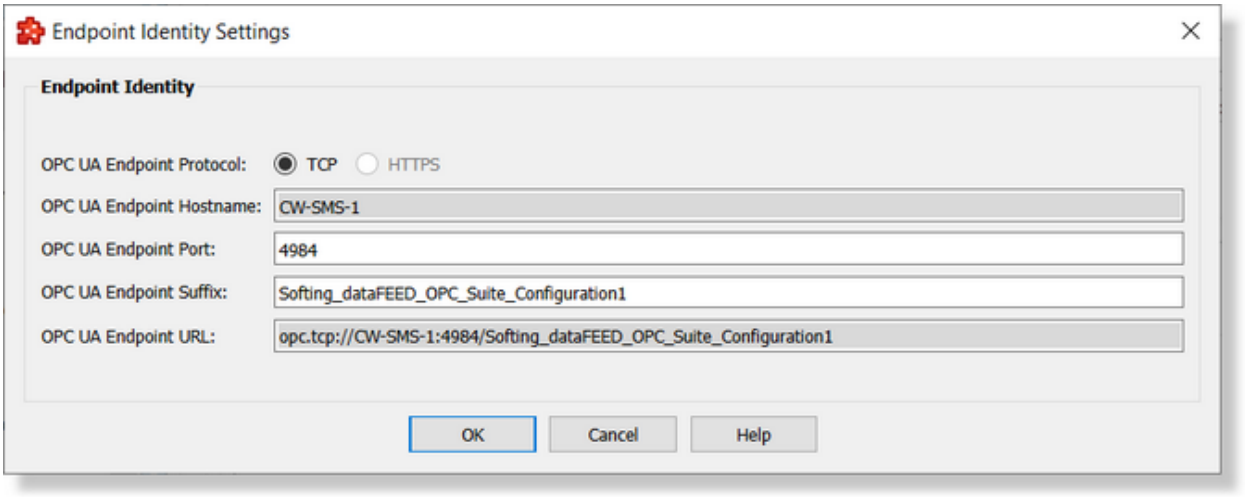
The following UA Server Endpoint specific settings are configurable:

- Server Endpoint identities
- Application Certificates
- Security and Reverse Connect Configuration settings
- User Authentication settings

OPC UA Server Endpoint Identities:

The auto-generated UA Server endpoint URL is accessible on main configuration page, the access rights to it is limited to read only (copy operations being available).

The server endpoint URL is used by the OPC UA Client for communication establishment and by the dataFEED OPC Suite to register to the LDS (Local Discovery Server). The server endpoint URL and other basic endpoint communication identities are configurable via "**Endpoint Identity Settings**" dialog which is activating by pressings the "**Configure**" button.



Attribute:	Description:
OPC UA Endpoint Protocol	Current version of dataFEED OPC Suite (version V5.38) is supporting only TCP protocol. HTTPs support being under development.
OPC UA Endpoint Hostname	Configured Hostname on current machine. By default, it is using the value from the Operating System. The hostname is configurable as global configurable settings common for both exposed endpoints.
OPC UA Endpoint Port	Configured port for TCP communication on local machine. The port will be used by the OPC UA Client for connection establishment. By default, the configured port is auto-generated using an unique value for each defined dataFEED OPC Suite configuration.
	Attention: Please ensure that the configured port is not used by other running instance (in case that it was manually configured) or by other running application on current machine. Usage of same port on TCP communication by 2 different processes is not allowed by the Operating System.
OPC UA Endpoint Suffix	Suffix is auto-generated and used only for an easy identification of the application based on the exposed URL. It has no functional usage on the communication establishment (using it into the UA Client connection being optional).
OPC UA Endpoint URL	The endpoint URL is auto-generated using the previously specified parameters. It is read only - the adjustments being done by modifying the other parameters.

Application Certificates

Each defined endpoint (enabled or not) has assigned an own Public Key Infrastructure (PKI) store containing:

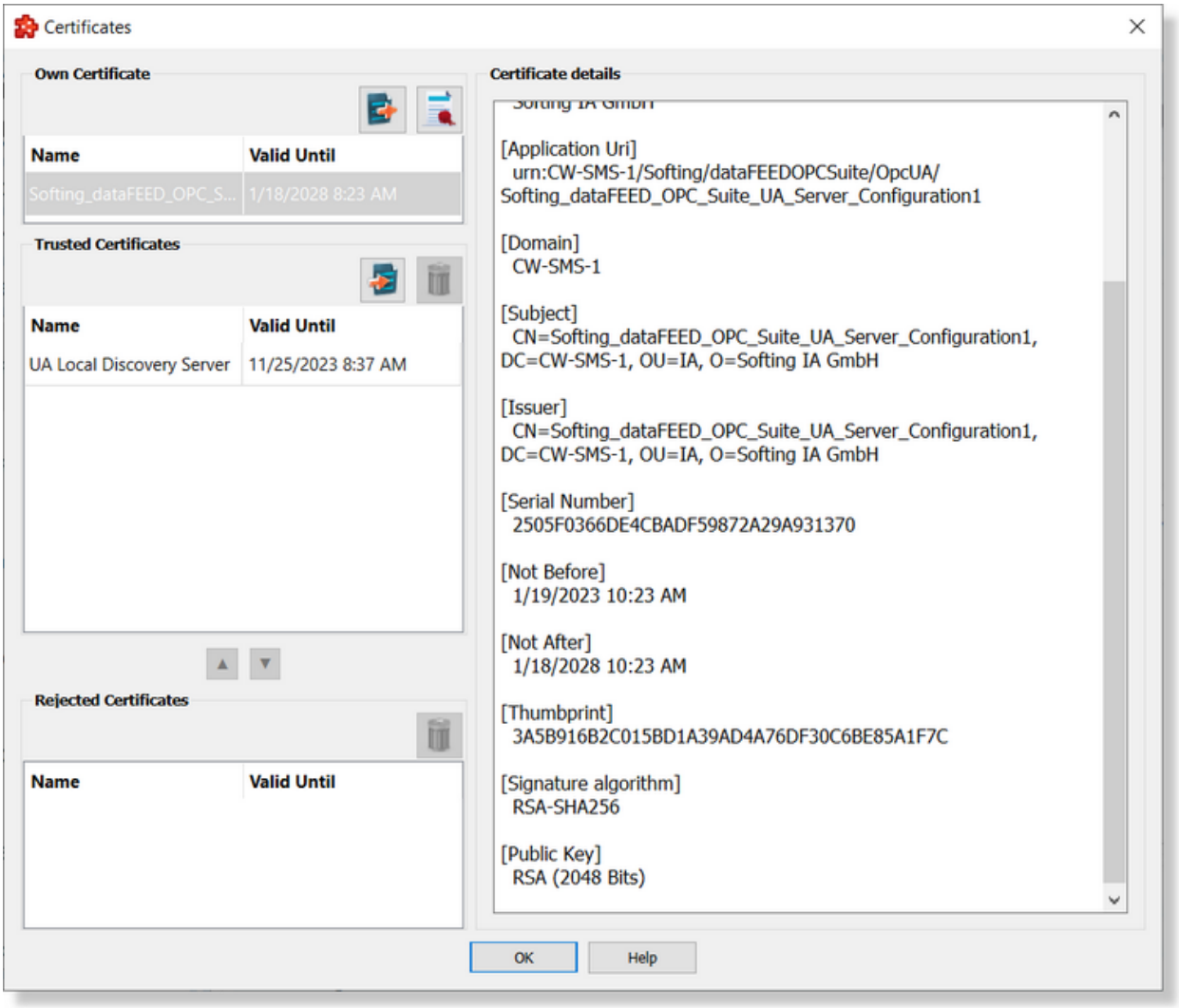
© Softing Industrial Automation GmbH

- Own Self Signed Certificate including a private key (format used: Distinguished Encoding Rules (.der) with Privacy Enhanced Mail Certificate (.pem)).
- Trusted and Rejected List locations.

The PKI Store is auto-generated at the creation time of current configuration (for default configuration is it created at the installation time). The validity of the own certificate is 5 years. Also, the trusted relation with LDS (Local Discovery Service) is automatically established.

Attention: The application certificate is validated by the OPC UA Client at the connection establishment time. All the validation rules accordingly with the OPC UA Specification V3.0 should pass. Due this reason, the own certificate is auto regenerated when an internal validation is failing (sample: changing the Hostname on working machine required a change of the used hostname on own certificate - re-generation automatically executed).

The PKI store can be manipulated (re-generation of own certificate or manipulation of trusted/rejected certificates) by pressing the corresponding "**Configure**" button on Application Certificates.



Left column contains *Own* certificate, *Trusted* certificates and *Rejected* certificates lists while the right column is populated with the details of the selected certificate.

Own Certificate.

This list contains only one certificate and offers the possibility to be exported.

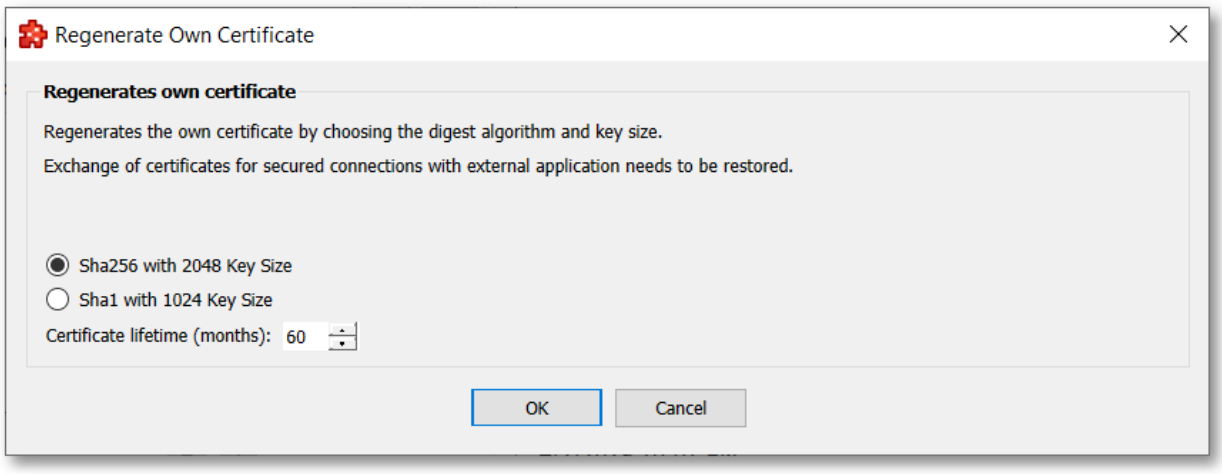
By using the export button the user is prompted to select a destination folder where to save dataFEED OPC Suite UA Server application certificate. This functionality is particularly useful when the Message security mode is other than "None".

The regeneration of own certificate is allowed by using the button. The own certificate can be generated by specifying the digest algorithm and encryption key size. By default the own certificate is regenerated using SHA256 digest algorithm and 2048 encryption key size independent of the settings from the previous certificate. Be aware that after a this operation the application own certificate was regenerated and all secured connections to the application should be reconfigured - the trust relation between application and connected servers/clients should be renew.

Also, In case of activation of Basic256Sha256 security policy - the regeneration of certificates using Sha1 digest algorithm with 1028 key size is forbidden.

Regeneration of certificates using Sha1 digest algorithm with 1024 key size will disable the Basic256Sha256 security endpoint.

It is also possible to set the certificate lifetime in months (default value is 60 months).



Attention: The re-generation of certificates without saving the configuration is corrupting the actual configuration (a regeneration of certificates is overwriting the existing certificates - but the thumbprint is different (without saving the configuration, the application will try to use a certificate which was deleted).

The back-up/restore configuration operations are providing the possibility to operate also over the PKI store.

© Softing Industrial Automation GmbH

Attention: Restoring a configuration on a different machine, is requiring also a re-generation of own certificates due the reason that the identities (including hostname) were changed. The lifetime of automatic regenerated certificate will be the default 60 month (5 year) not the lifetime specified at manual regeneration.

Trusted Certificates.

This group contains the list of the currently trusted certificates. The validity date can be easily checked on the "Valid Until" column. Certificates from this list can be dragged and dropped in the rejected certificate list.

By using the import button  the user is prompted to select a server application certificate (.der format) that will be added to the trusted folder. This operation basically can substitute Receive server certificate operation.

One certificate can be deleted using  button.

Rejected Certificates.

Contains the list of the rejected certificates. Each certificate can be dragged into trusted or deleted.

The up and down arrow buttons can be used to easily move certificates from trusted to rejected locations and vice-verse.

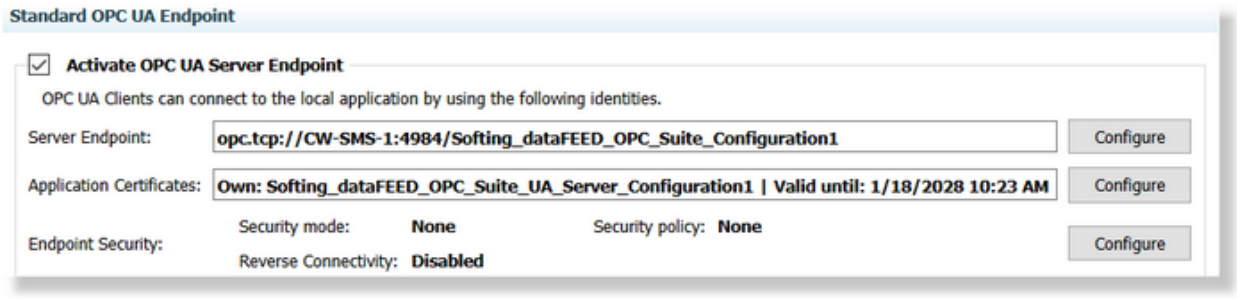
Certificate details.

In this area shows some details of the selected certificate like: location (on the disk), Application Name, Application Uri, Organization, Subject, Issuer, Validity dates and so on. The details area is read-only.

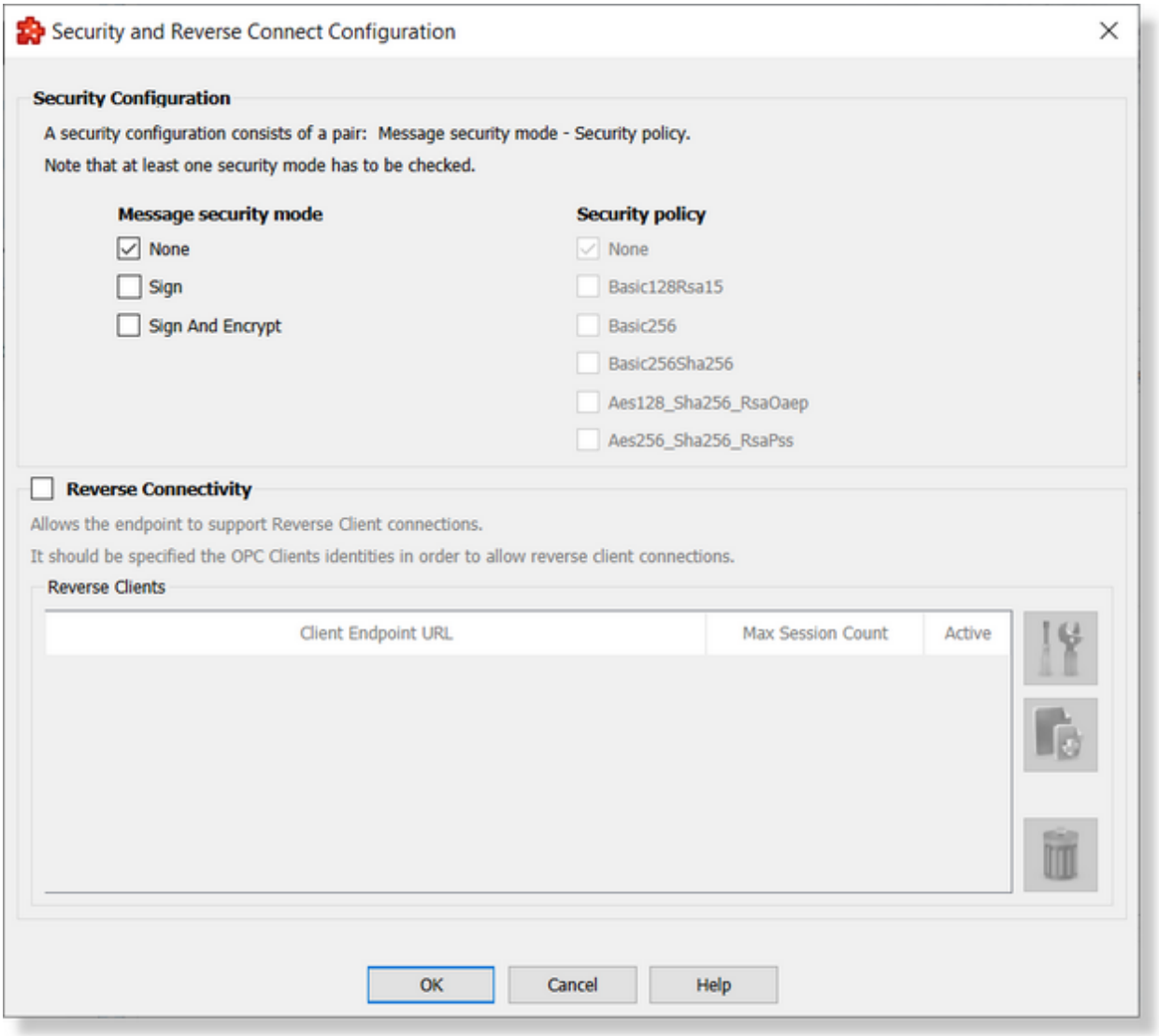
Security and Reverse Connect Settings

Allows the user to specify endpoint settings for the "**Message Security Mode**" and "**Security Policy**". Also, it is allowed to activate and configure the "**Reverse Connectivity**". This settings will be used by the OPC UA Client for connection establishment - specially for defining the security degree of communication.

On the main OPC UA Client data destination page is visible the lowest configured security mode. The Reverse Connectivity status is also visible.



Pressing the "**Endpoint Security**" corresponding "**Configure**" button it is opening the "**Security and Reverse Connect Configuration**" dialog which allows you to specify the desired degree of security and the settings for reverse connectivity (if enabled).



Security Settings

© Softing Industrial Automation GmbH

At least one message security mode and one security policy have to be selected. It is also possible, to have multiple security configurations. That means, that more than one message security mode and security policy is selected.

As message security mode you can choose between none, a signed and a signed and encrypted mode.

The security policy can be not encrypted or you can choose between different security algorithms.

Attention: The configured security configuration should be synchronized with the OPC UA Client capabilities.

Note: The default security configuration is set to unsecured ("None" as Message Security Mode and Security Policy).

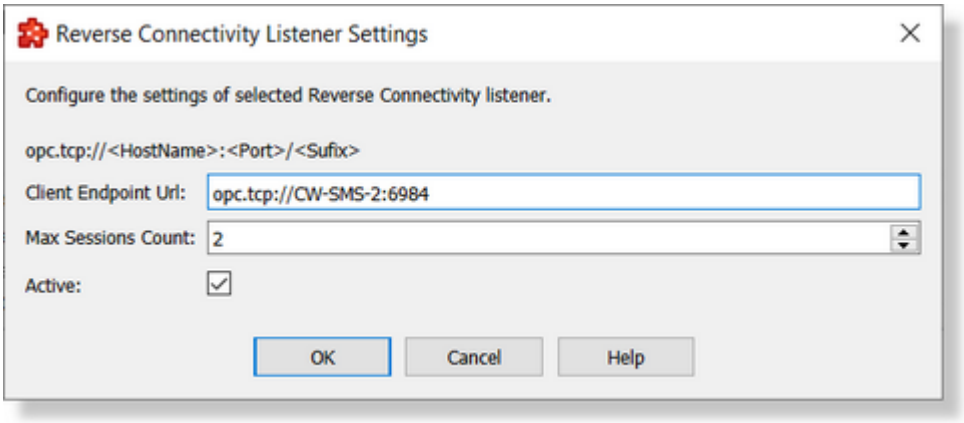
Also, in case of a usage of a secured configuration, the trusted relation between dataFEED OPC Suite and OPC UA Client should be manually configured (the OPC UA Client's own certificate should be imported into the application trust list location and the application's own **certificate** should be exported into the trusted list location of the OPC UA Client).

Reverse Connect Settings

This settings allows the user to activate the **Reverse Connectivity** functionality.

The Reverse Connect feature, allows a server to initiate the connection to a client. The server tries cyclically to open a physical connections (sockets) to the client and send the "Reverse Hello message". A client (which supports Reverse Connect), is listening for such a socket and "Reverse Hello" messages will use this to create the "secured channel".

The functionality can be enabled by enabling the "Reverse Connectivity" check-box - it can be used only with at least one definition of a client listener. Defining or modifying the settings for a client listener can be done by pressing the "*Add a new Reverse Client*"  button or *"Properties of selected Reverse Client"*  button - immediately after pressing the buttons, the "Reverse Connectivity Listener Settings" dialog will be open allowing to specify the Reverse Listener settings.



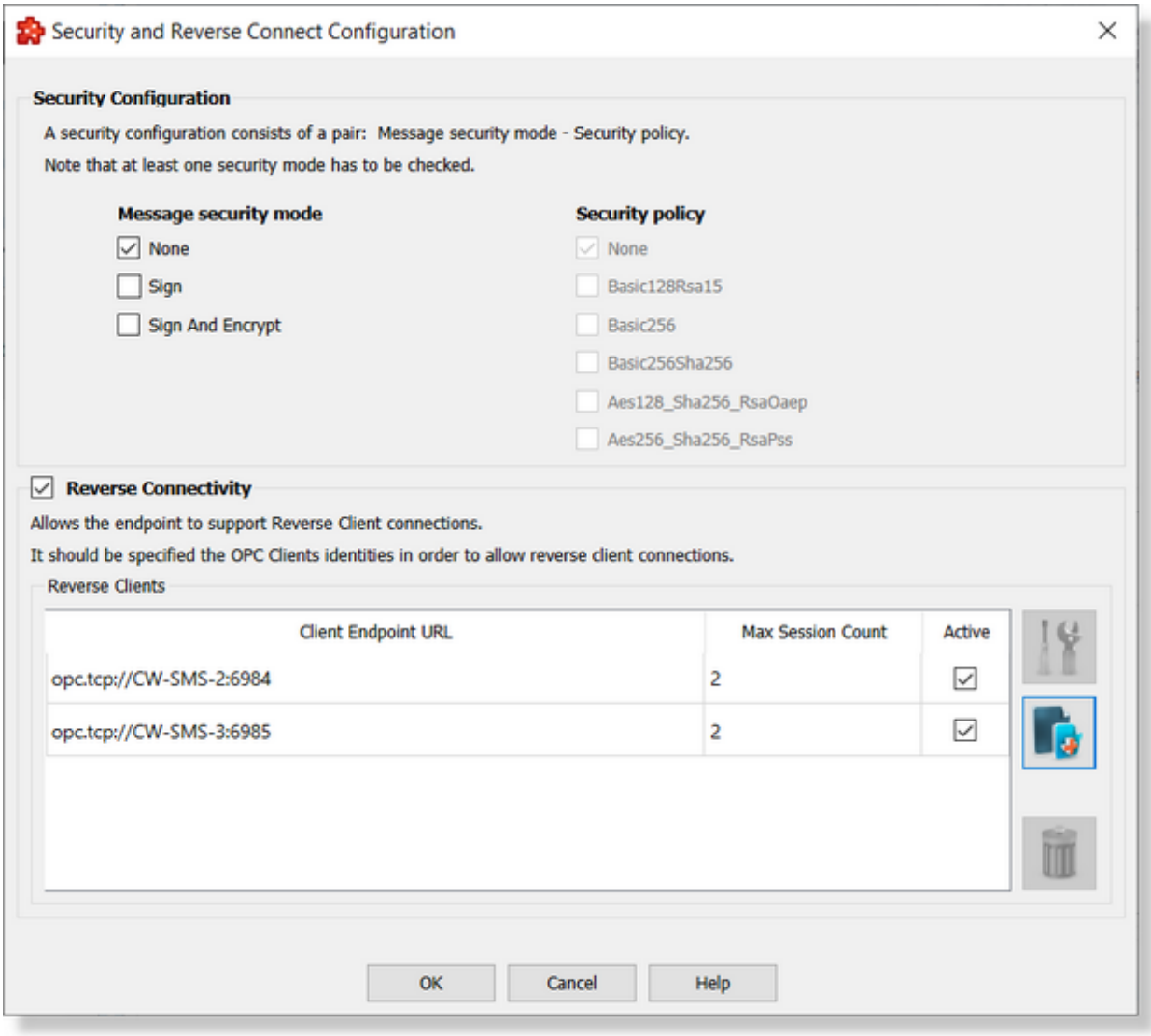
Attribute	Description
Client Endpoint URL	The URL where the OPC Client listens for reverse connections. The URL expression must comply with the OPC UA specifications. The port number must be an integer between 1024 and 65535.
Max Session Count	Maximum number of connections allowed for configured OPC Client.
Active	State of configured OPC Client

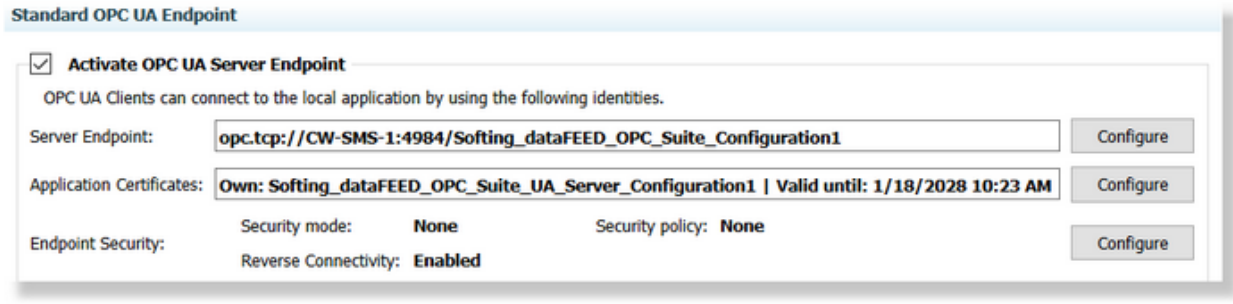
Deleting an defined OPC Client Listener can be done by pressing the "*Delete the Selected Reverse Client*"  button.

The Standard Endpoint supports maximum 50 active individual OPC Client connections.

Attention: The maximum number of the sockets which can be open by the application is 106. Due this limitation, the maximum number of parallel connected OPC Clients is limited to 106 even if the sum of Max Session Count for all defined (max 50) OPC Clients is bigger. After the number of 106 open sockets is reached - no other additional connection can be established.

© Softing Industrial Automation GmbH





Note: Enabling the Reverse Connections is not deactivating the Forward Connections.

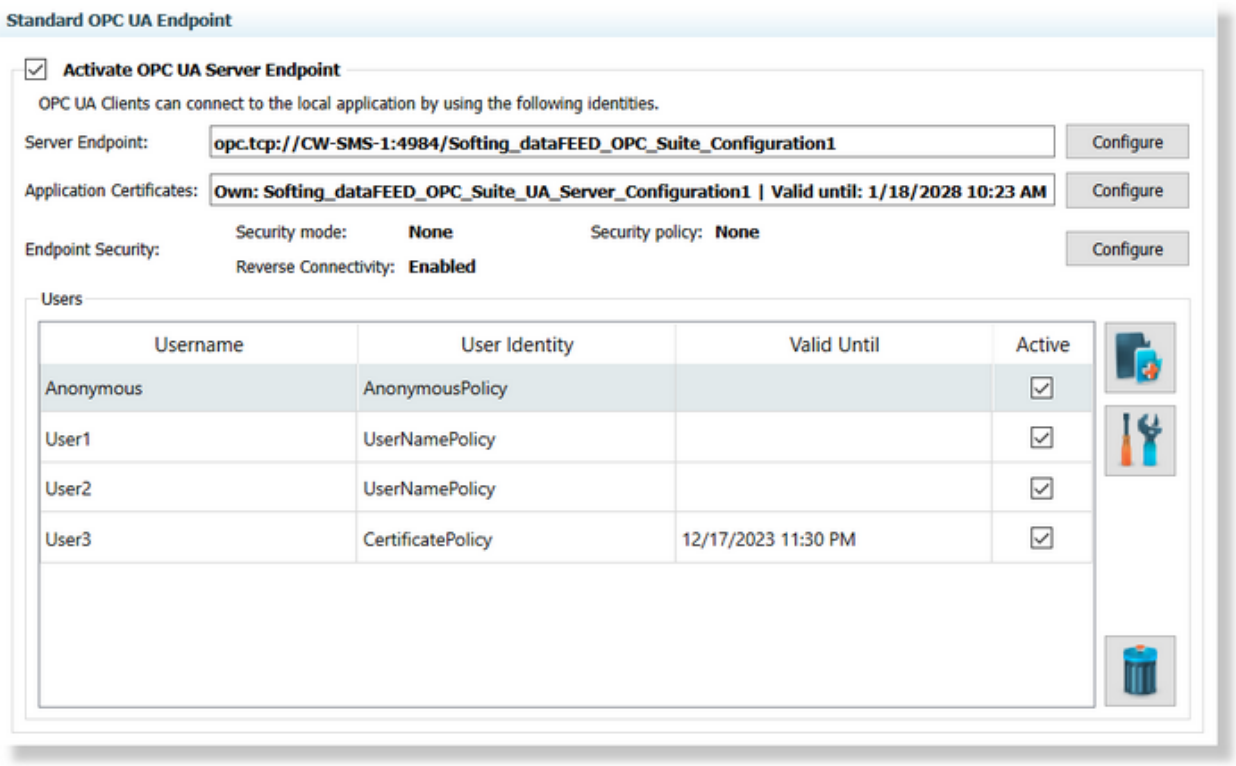
Note: All the settings applicable a a session created on a Forward Connection are applicable for Reverse Connections too.

User Authentication Settings

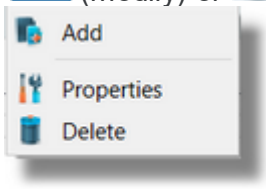
The user authentication settings are used by OPC UA Clients for building up a communication to dataFEED OPC Suite Standard Endpoint. The access to dataFEED OPC Suite can be protected by using a user name and password, as well as by using a certificate.

By enabling anonymous policy, there is no authentication necessary. By default only the anonymous policy is enabled.

The application is supporting up to 5 configured active users. At least one active user should be configured - the application not allowing the deletion of the last active user.



Only the active users are used by the application, all the other ones being ignored at the application start up time.

The application allow the user to add, modify or remove an authentication user. This operations are available based on right buttons:  (add),  (modify) or  (remove) - or based on a context menu (right click on user definition table) 

.

© Softing Industrial Automation GmbH

Adding/Modifying a user is operated by a designed user definition wizard page - responsible for user identification settings.

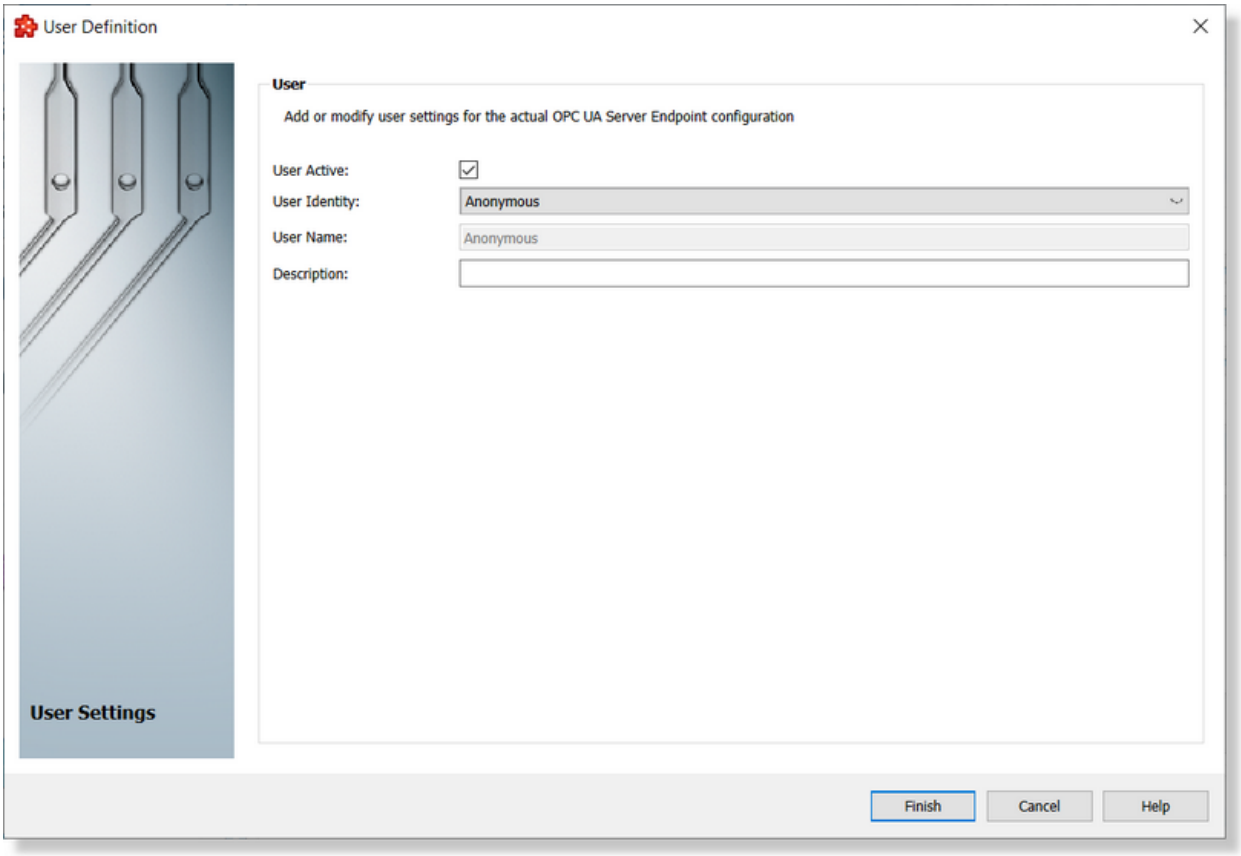
The application is supporting the following user authentication setting:

- **Anonymous** (no authentication being necessary).
- **Username and Password** (the authentication is executed based on a set of credentials (username/password) - which is specified by the OPC UA Client - the matching between the active user (including the corresponding passwords) and the credentials specified by the OPC UA Client being mandatory).

Attention: The authentication based on username and password is using case sensitive policy.

- **Certificate** (the authentication is executed based on a user certificate - which is delivered by the OPC UA Client via OPC UA Protocol - the matching between the certificates of the active configured users and the certificate deliver by the OPC UA Client being mandatory).

Anonymous user



The defined user can be activate/deactivated (all the inactive users are ignored). By default, on a clean configuration, the Anonymous User is defined and active.

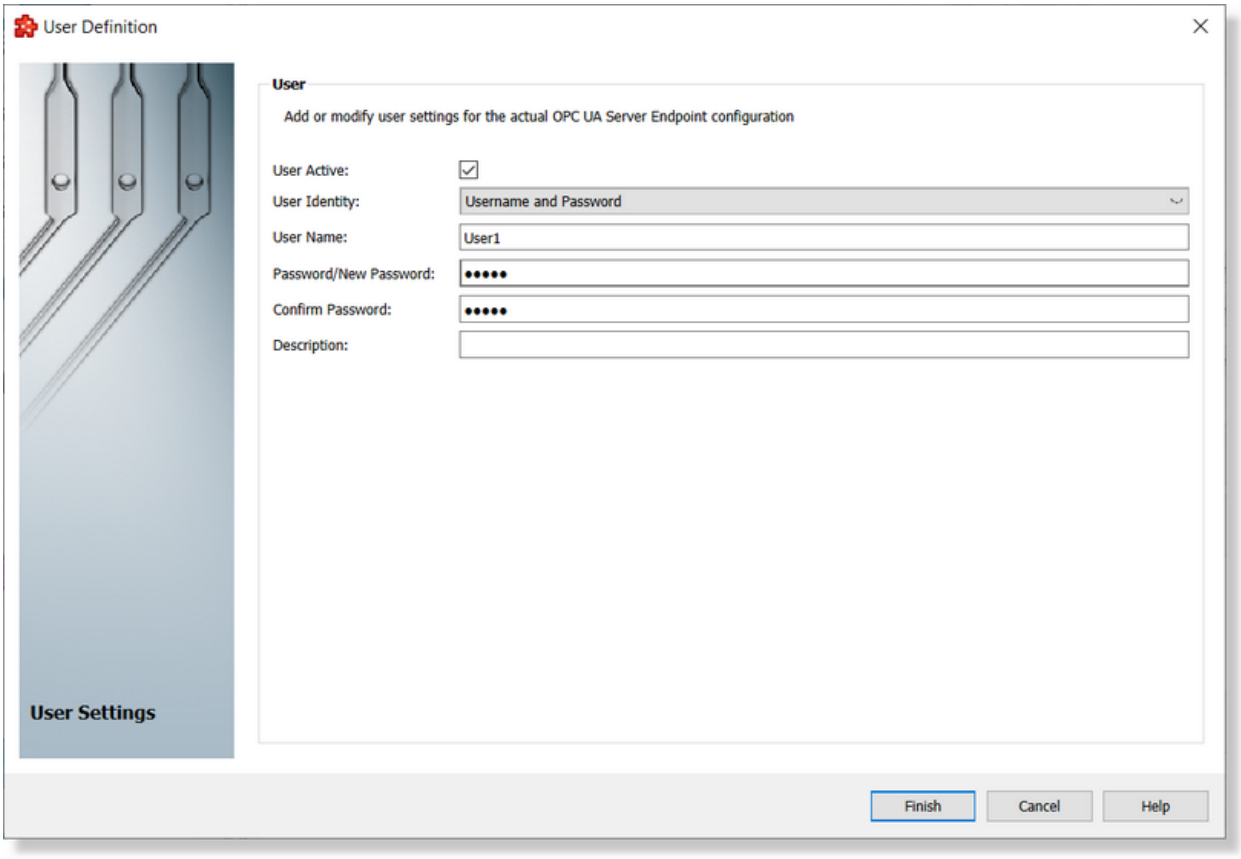
Note: The Anonymous User is intended being used only while unsecured policy is configured.

Note: Only one Anonymous User is possible being defined.

The "User Identity" type can be changed to:

- Username and Password
- Certificate

User identified by a set of credentials (user-name and password).

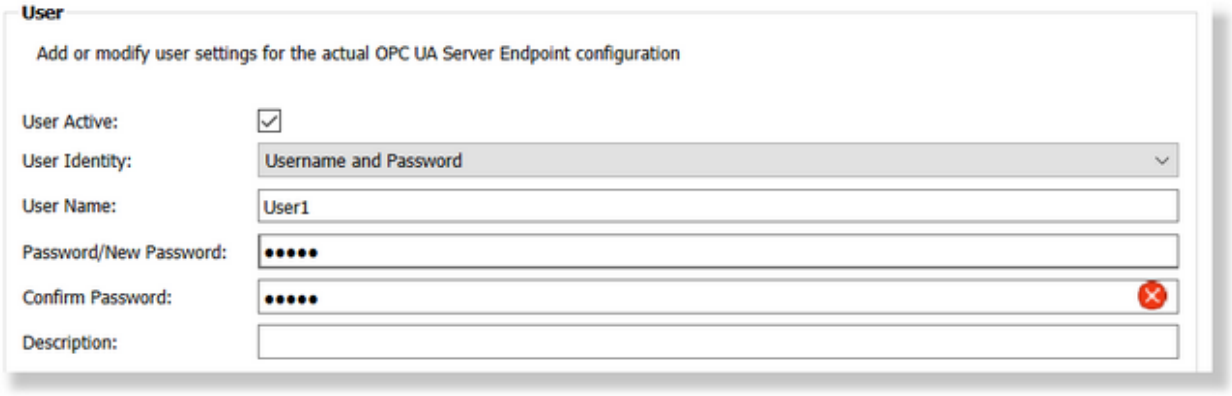


The user can be activate/deactivated (all the inactive users are ignored).

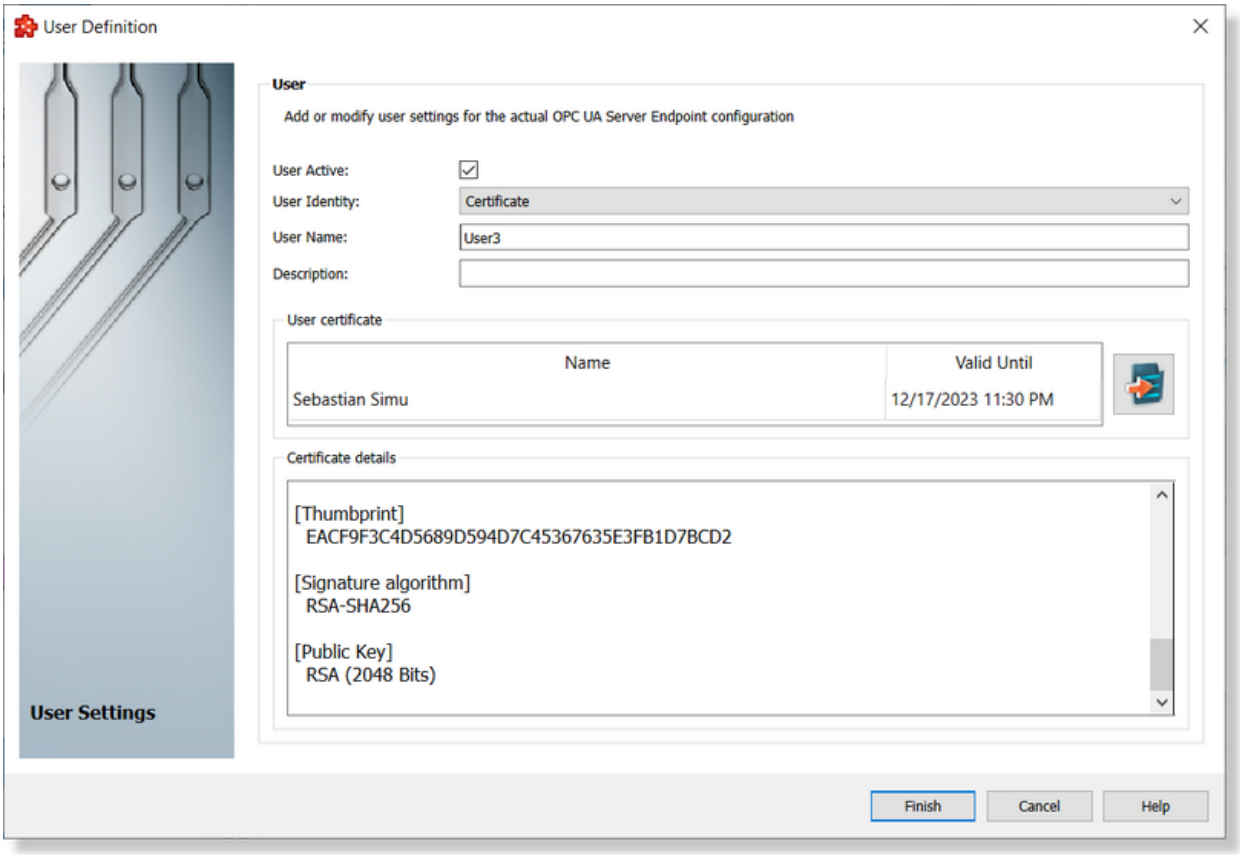
The username should be unique (it is not allowed to define duplicate users even if the specified password or identity type is different).

The confirmation of the password is mandatory - in case of failure, the user definition cannot be saved and the reason is visible on the wizard page:

© Softing Industrial Automation GmbH



User identified by a specified certificate

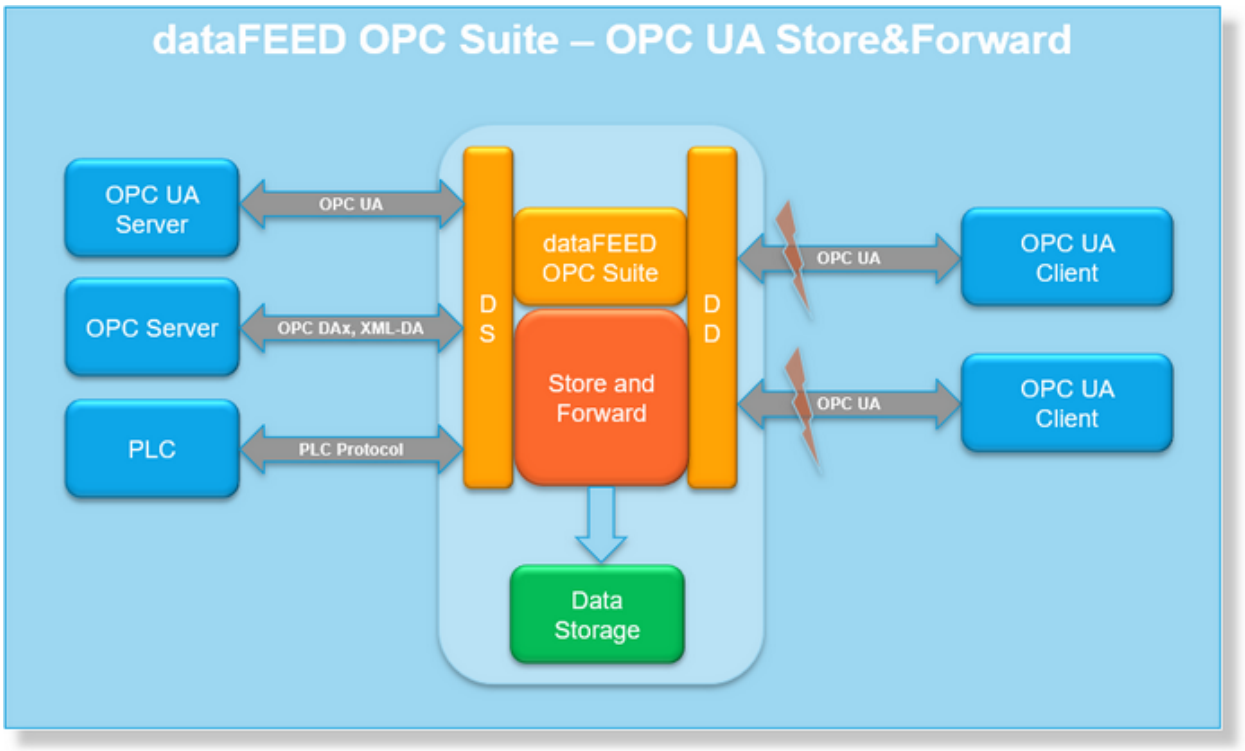


The user can be activate/deactivated (all the inactive users are ignored).

Importing a user certificate is mandatory. The supported certificates formats are: .der and .pfx. After a successfully import, the details of the imported certificates are visible on the wizard page.

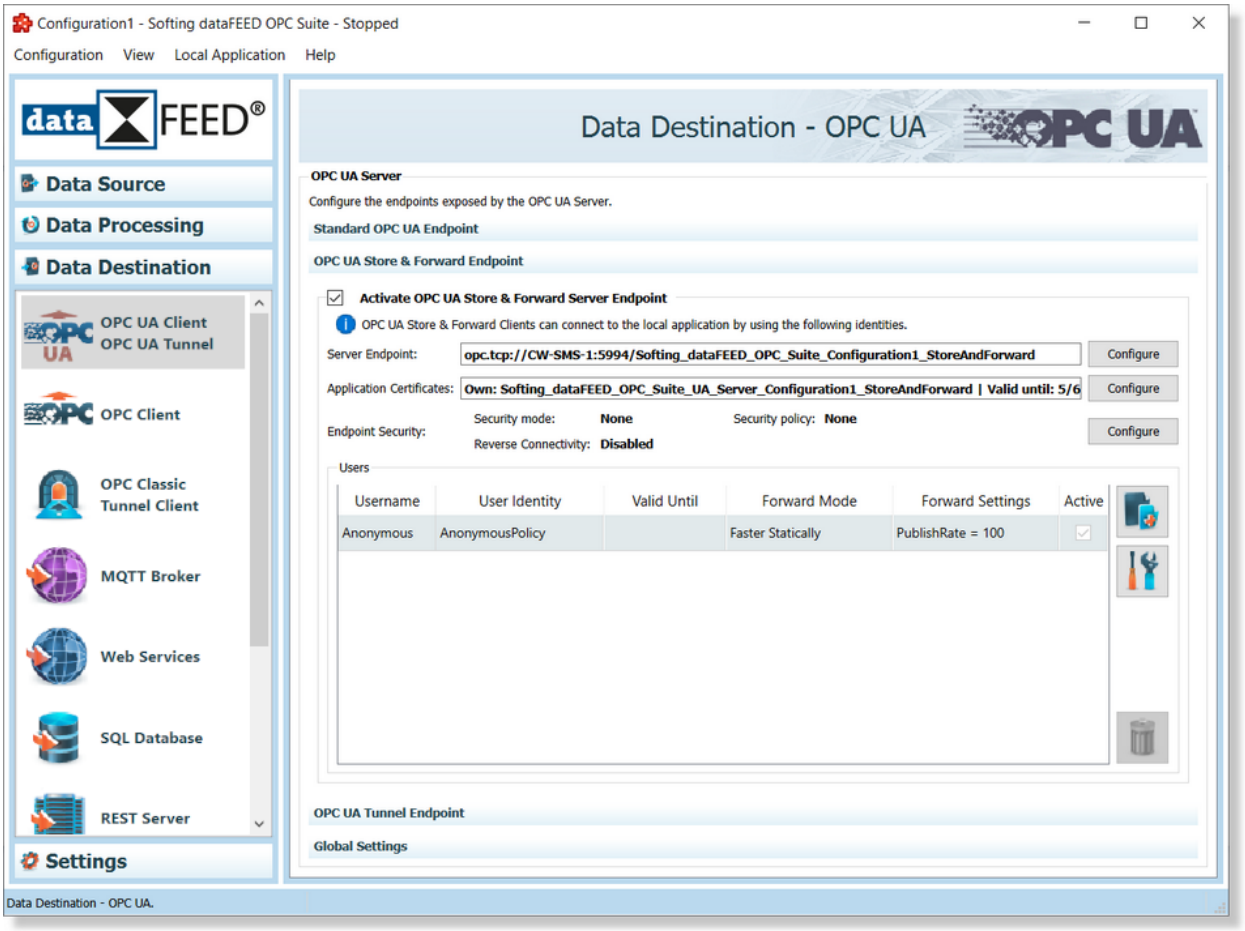
Licensing

The functionality is licensed by **dataFEED OPC Organizer UA** or **dataFEED OPC Suite/dataFEED OPC Suite Extended** licenses. *Demo Mode* is working on full functionality for 72 hours.



The OPC UA Store&Forward Server endpoint is enabled by the activation of "**Activate OPC UA Store & Forward Server Endpoint**". By default it is disabled. Being enabled, the OPC UA Store&Forward Server endpoint is open and accessible on the exposed OPC UA Server over the OPC UA protocol. Any OPC UA Clients will be able the establish connections to the exposed OPC Server over the exposed endpoint and operate on standard services.

© Softing Industrial Automation GmbH



Disabling all endpoints (***Standard OPC UA Server Endpoint***, ***OPC UA Store&Forward Endpoint*** and ***OPC UA Tunnel Server Endpoint***) will completely deactivate the OPC UA Server functionality.

OPC UA Store&Forward Functionality

The OPC UA Server is exposing a dedicated server endpoint which allows the OPC UA Clients to connect and perform any supported operations. For each defined session, the Store&Forward module is creating a mirroring of all created subscriptions (including the defined monitored items), it is storing all the received data-values and it is continuously trying to deliver them to the proper OPC UA Client - the data-values being released only after the confirmations is received from the OPC UA Client.

Basically, in case of any unattended network interruptions or client disruptions (coming after an application crashed on OPC UA Client side or any failures at the OPC UA Client level) the application is trying preserving the subscriptions (including the monitored items), is still keeping to collect data-values and try forwarding them the OPC UA Client. Immediately after the OPC UA Client is reconnected to the application - the forward mechanism is start forwarding all the stored data-values (preserving the order and all data-changes identities (Value, Source and Server Timestamps and StatusCodes)) to the re-connected OPC UA Client.

Attention: The functionality comes with a set of limitations, overhead on data processing operations at OPC UA Client level and possible performance penalties. Please read the documentation, analyze the functionality in your environment and decide if it is not affecting your behavior before decide to enable it.

Attention: The functionality is not OPC Compliant accordingly with the OPC Specs 3.0.

General Overview on Store&Forward functionality

The main functionality is to ensure that a communication interruption (based on a network interruption or a OPC UA Client disruption) - without an intended disconnect operation executed - is not loosing collected data changes at the OPC UA Client side. The functionality is supporting short and medium (up to 1 hour) communication interruptions.

The OPC UA Server is able to identify a session, to mirror all its subscriptions and all its monitored items and to publish the collected data-changes for each monitored items when the OPC UA Client connection is established. Also, all the collected data-changes are stored at the application level, published to corresponding OPC UA Client and released only when the confirmation for a data-change is received from the OPC UA Client side. In case of a network interruption or an unattended OPC UA Client disruption (crash or functionality disruption at the OPC Client side) the application is able to detect it, the server session is not affected at the application level and all the monitored items remain active (still collecting data-changes from corresponding data sources and store them at the application level) - when the communication with OPC UA Client is re-established, then the application is able to detect it and start forwarding the stored data-values to the corresponding OPC UA Client (the data-values are forwarded with a bigger frequency in order to deliver all the stored data-values in a limited period of time). Finally the stored data-values are released when the confirmation is received from OPC UA Client side.

The functionality is based on 3 main mechanisms:

- OPC UA Client identification mechanism - the application should be able to detect that a session (including its subscriptions and monitored items) is overwriting or taking ownership of stored data of a previously connected session - this is needed in order to start forwarding data after a network interruption or an OPC UA Client communication disruption.
- Storing data values for all active monitored items - the application is storing all the received (collected) data values for all active monitored nodes - the data values are released only when they are acknowledged by the connected OPC UA Client or releasing them is required by the OPC UA Client (the session being intentionally disconnected by the OPC UA Client or the monitored items being intentionally deleted).
- Forwarding data values for all active monitored items - the application is continuously forwarding collected and stored data values to the connected OPC UA Client until the confirmation is received. Same data values can be forwarded (published) for multiple times until the OPC UA Client received and acknowledged them.

All supported OPC UA Services (except Monitoring Items) are not affected.

Attention: The dataFEED OPC Suite UA Store&Forward Server cannot be used as dataFEED OPC Suite UA Server data source into other configuration due the reason that the internal UA

© Softing Industrial Automation GmbH

Client is from the aggregation sever is not supporting this functionality. In case that a connection is tried being established then the UA Store&Forward Server is rejecting it with "BadNotSupported" status code.

Client (Session) Identification procedure

Attention: For each defined user it is allowed to have only one active session.

The main functionality of the Store&Forward UA Server Endpoint is to ensure that no data-values are lost while a the communication to the OPC UA Client was lost - basically the application is able to detect that the communication with the OPC UA Client was lost - start to collect and store all the data-values received for the existing monitored items and immediately after the communication was re-established to start delivering stored data-values (preserving the data order and data integrity).

Note: The session is kept online and active at the server side (internal OPC UA Server of dataFEED OPC Suite application) for period up 1 hour even if there is no communication with the OPC UA Client (including Publish services on Keep Alive calls).

A communication interruption, usually means:

- network interruptions (problems appearing at the network infrastructure level).
- communication disruptions at the connected OPC UA Client side - the OPC UA Client being crashed, hangup or the OPC UA Client being not anymore able to collect and process data-values.

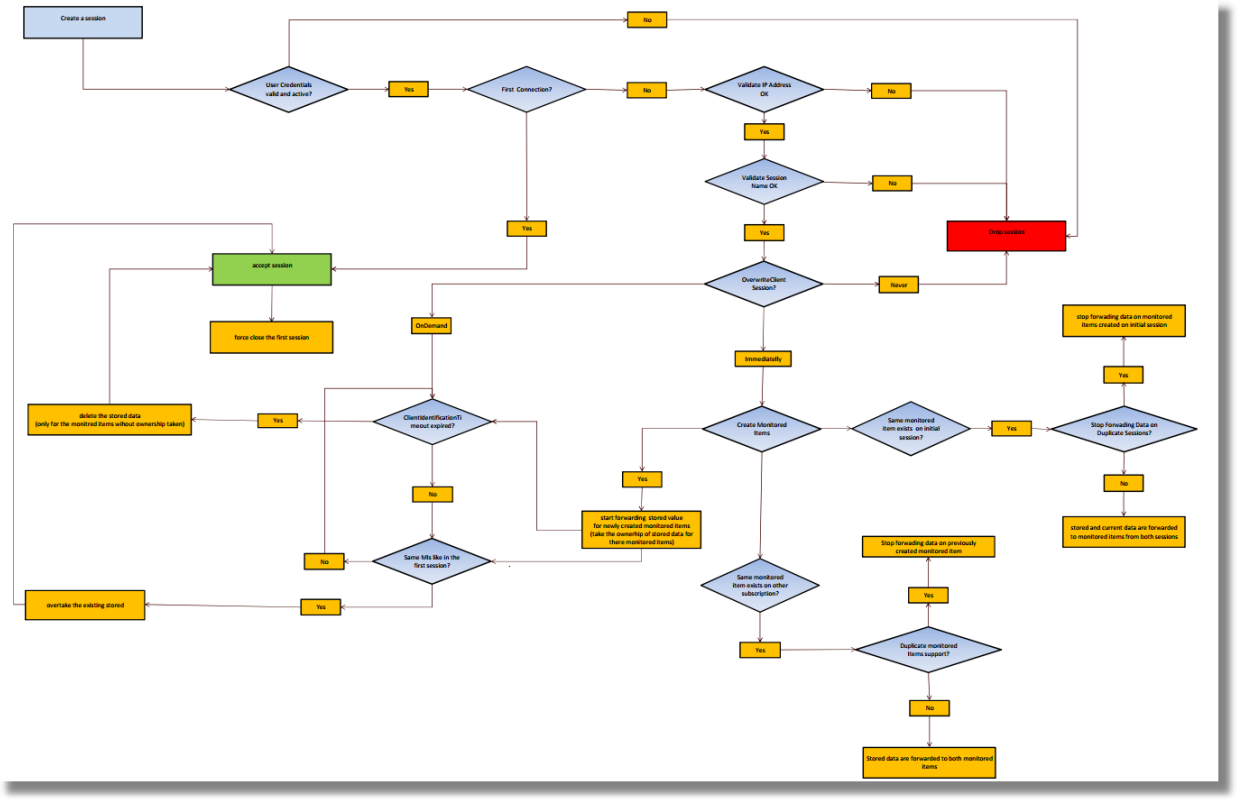
Communication re-establishment can be operated in based on following session handling:

- The existing session is used - basically the existing session is kept active at the OPC UA Client side (this is the responsibility of the OPC UA Client) for the entire network interruption - no changes being operated at the session level and immediately after the communication is re-established, the application is detecting that the existing session become accessible, identify it as "**Forward Client**" and starts forwarding the stored data-values.
- A new session is created - basically the old session was disconnected or deleted while the network was interrupted or due an OPC UA Client disruption (usually a crash or application hangs at the OPC UA Client side). Immediately after the network re-establishment or an application restart at the client side - the OPC UA Client is creating a new session trying to recover the previous one. The initial session is kept alive at the OPC Server side and when the second session is tried being created (using the same user authentication identities) then the second session enters in conflicts with the initial session - the application being responsible to identify if the second session is accepted as "**Forward Client**" or not.

  Based on the settings described in Override Client Session for a configurable Client identification Timeout the application (based on configured settings) is evaluating if the second session should be accepted and should take the ownership of stored data-values or not. In case that it is identified as "**Forward Client**" the ownership of stored data-values will be transferred to the second session otherwise all the stored data-values will be released supposing that the second session intends to not override the previous one.

Attention: Accepting the second session (based on configuration) will force closing the previous one (with or without transferring the stored data-values to the second session) after the Client Identification process is completed (with or without identifying the second session as "Forward Session"). Due this reason, it is very important to not create and delete test or control sessions while the initial session was created, activated and started collecting data-values.

The next schema will describe the creation and acceptance of a new session (more details about each configurable settings will be described later on configuration chapter):



Data Collector and Storage

The application is collecting data-values from all supported data-sources immediately after a monitored item is created - the sampling interval of the monitored item being used as collecting data-values rate. In case when same variables are monitored using multiple monitored items, the application is using an optimization mechanism which optimizes the collecting data-values frequency rates - basically only one monitored item is created at the data source level using the lowest sampling interval.

All the collected data-values are received by all defined monitored items (on the data storing mechanism) which is storing them in order to be transferred later to the OPC UA Client. The received data-values are stored temporary in a memory buffer (limited to 100 data-values) which later on is stored in a persistent SQLite database. The buffer being used for very fast access to the data-values on forward process. The path to the persistent storage is configurable and can be set by the user at the configuration time as global setting for Store&Forward endpoint.

© Softing Industrial Automation GmbH

Attention: The Data-Storing while OPC UA Client connection is interrupted is limited to 60 minutes (the timeout on the server session being revised to 60 minutes). In case that the communication is interrupted for a period bigger than 60 minutes, then all the stored data-values corresponding for current session are released (deleted) and the session is automatically closed and destroyed.

Note: A stored data-value is released only after the confirmation from OPC UA Client is received (the acknowledge response on corresponding used sequence number is received by the application).

Deletion of a monitored item (or a subscription which contains a set of monitored items) - requested by the connected OPC UA Client will release all the stored data-values for the corresponding deleted monitored items. Also, deletion of a session (requested by the OPC UA Client via a disconnection service) will delete all the stored data-values on monitored items corresponding to deleted session.

Attention: The persisted data-values are not persistent over application restart operations. Restarting or Stopping the application will also release all the stored data values.

Attention: The persistent SQLite database should not be edited or modified using external tools while application is running. This will generate data lost and possible wrong functionality at the application level.

Data Forwarder

The application is storing data-values while the connection to the OPC UA Client interrupted. Immediately after the OPC UA Client re-connection (after a network re-establishment or OPC UA Client restart) the application will start forwarding the stored data-values. The forwarding process should ensure that the data-values are forwarded with a faster rate than collecting new data (this is mandatory - and the configured settings should reflect it in order to not accumulate more data-values than delivering).

Attention: The collecting data-values frequency is based on used sampling interval of each monitored item. Due the reason that under communication interruption time, the application is collecting data and in a reasonable time it should be able to deliver all collected data, the forward settings should ensure that this is possible.

Attention: The forwarding data-values frequency (rates) is configurable at the configuration time by the Configuration Tool, but the collecting data-values frequency is an OPC UA Client specific settings which is handled at the running/operating time... it is the user responsibility to synchronize the rates (frequencies).

The forwarding process is operated based on Publish and Republish OPC UA Services.

The dataFEED OPC Suite Store&Forward is providing 2 modes (possibilities) for forwarding data values after a communication interruption:

- **Publish Faster** - the application is publishing data-values with a faster rate than the one required by the OPC UA Client at the session connection time. Basically, the requested publish interval (requested by the OPC UA Client at the session creation time) is revised to a configured value (configured by the user at the configuration time).

  Recommendation: it is recommended to adapt the forward rate (the value will be used as revised publish interval) to the value which will be intended being used as publish interval by the OPC UA Client.

Note: On this forward mode, the application is responsible to publish a data-values on each publish request (revised publish request - configured by the user) - the monitored item queue size being revised to 1.

Attention: Revising the publish interval by the OPC UA Server (internal OPC Server of dataFEED OPC Suite application) should be supported by the used OPC UA Client. Also, the OPC UA Client should be able to adapt its functionality to a revised publish interval (using a value lower than the requested one) - otherwise the OPC UA Client will not be able to consume the delivered data values.

Attention: The delivery data-values without any OPC UA Client interruptions (classic OPC UA Communication scenarios) can be affected by the revised publish interval based the report between requested publish interval and sampling interval - as result the application will deliver in some scenarios more data values than the OPC UA Client expects.

The minimum accepted value for forward rate is 10ms. The revised publish interval is handled on CreateSession service requested by the OPC UA Client.

- **Publish More Data** - The application is publishing more data-values on the same publish service than the one required by the OPC UA Client at monitored item creation time. Basically, the requested queue size for each created monitored item (requested by the OPC UA Client at monitored item creation time) is revised to a configured value (configured by at the configuration time).

  Recommendation: it is recommended to adapt the forward queue size (the value will be used as revised queue size) to the value which will be intended being used as monitored item queue size by the OPC UA Client.

  On each publish service the application is trying to deliver (in case that the application is containing enough stored data-values) a complete queue (the number of deliverable data-values being equal with the revised queue size). If the number of stored data-values is less than the revised queue size, then the application is delivering them completely on the same publish request.

Note: On this forward mode the rates used for publishing operations (publish intervals) are preserved from the ones requested by the connected OPC UA Client.

Attention: Revising the queue size by the OPC UA Server (internal OPC Server of dataFEED OPC Suite application) should be supported by the used OPC UA Client. Also, the OPC UA Client should be able to adapt its functionality to the revised queue size (using a bigger

© Softing Industrial Automation GmbH

queue size than the requested one) - otherwise the OPC Ua client will not be able to consume the delivered data value.

Attention: The delivery data-values without any OPC UA Client interruption (classic OPC UA Communication scenarios) can be affected by the revised queue size based on the report between publish intervals and sampling intervals - as results the application will deliver in some scenario more data than the OPC UA Client expects.

Sample: on a requested publish interval set to 10s with a sampling interval set to 500ms and with a queue size of 10 (discard older value set to false) it is expected to deliver each 10s maximum 10 values (supposing that the source is changing its values with rate faster than 500ms) - but now in case that the revised value of queue size is 100, then the application will deliver maximum 20 values on the same supposition.

The forward settings are configurable for each defined user. Basically there can be used different settings for different sessions.

The forward mechanism is activated only when a new session is identify as "Forward Session" (see Client Identification description) - the forwarding data being started on each existing monitored item with the configured value. Activation/Deactivation of forwarding mechanism is not affecting at all the data delivery on other existing sessions.

Note: The application is delivering the next data-value only after the confirmation on the previous sent data-value is received (the acknowledge being received for the corresponding sequence number which contains the previously sent data-value on the same monitored item.

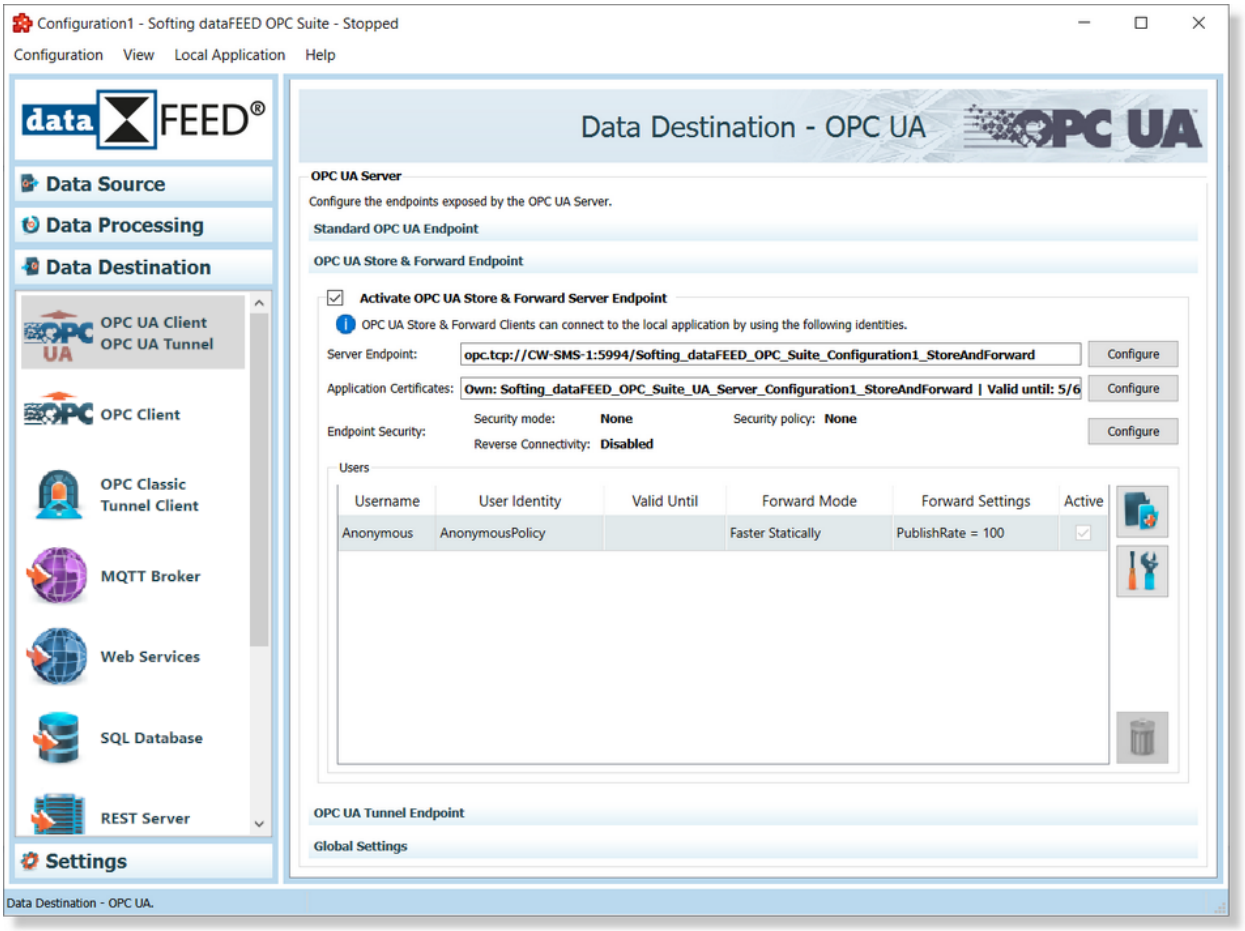
Note: A data-value is always re-sent when the confirmation from OPC UA Client is not received inside a timeout which bigger that actual publish interval multiplied with the value of "**Client Acknowledge Response on Publish Request**" parameter.

Attention: The Store&Forward functionality is not compatible with OPC UA Clients which are not acknowledging the publish operations.

Configuration of Store&Forward Endpoint

The OPC UA Store&Forward functionality is accessible from the application by opening a dedicated UA Server Endpoint which is activated by "**Activate OPC UA Store&Forward Endpoint**".

The configurable settings are accessible on the main configuration page by selecting **OPC UA Store & Forward Server Endpoint** navigator option. Immediately after, the specific settings for OPC UA Store&Forward endpoint are made available on the main configuration page.



Disabling both endpoints will completely deactivate the OPC UA Server functionality.

The following OPC UA Store&Forward Server Endpoint specific settings are configurable:

- Server Endpoint Identities and Global Store&Forward Settings
- Application Certificates
- Security and Reverse Connect Configuration Settings
- User Authentication and Forward Settings

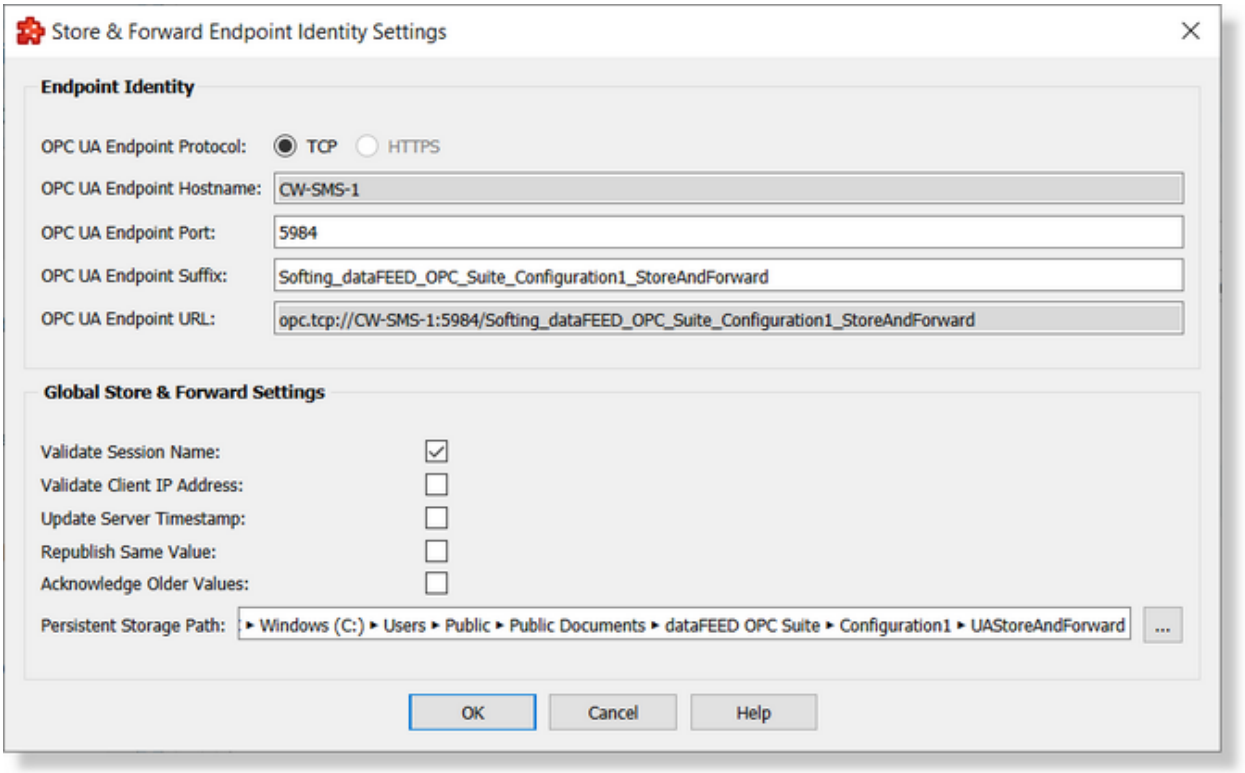
Server Endpoint Identities and Global Store&Forward Settings

The auto-generated OPC UA Store&Forward Server Endpoint URL is accessible on main configuration page, the access rights to it is limited to read only (copy operations being available).

The special OPC UA Store&Forward Server Endpoint URL is used by the OPC UA Client for communication establishment and by the dataFEED OPC Suite to register to the LDS (local discovery server). The server endpoint URL and other basic endpoint communication settings can be configured from "**Store & Forward Endpoint Identity Settings**" dialog which is activating by pressings the "**Configure**" button.

Also, the Global Store&Forward settings are configurable on same dialog.

© Softing Industrial Automation GmbH



OPC UA Server Endpoint Identities:

Attribute:	Description:
OPC UA Endpoint Protocol	Current version of dataFEED OPC Suite (version V5.38) is supporting only TCP protocol. HTTPs support being under development.
OPC UA Endpoint Hostname	Configured Hostname on current machine. By default, it is using the value from the Operating System. The hostname is configurable as global configurable settings common for both exposed endpoints.
OPC UA Endpoint Port	Configured port for TCP communication on local machine. The port will be used by the OPC UA Client for connection establishment. By default, the configured port is auto-generated using the an unique value for each defined dataFEED OPC Suite configuration.
	Attention: Please ensure that the configured port is not used by other running instance (in case that it was manually configured) or by other running application on current machine. Usage of same port on TCP communication by 2 different processes is not allowed by the Operating System.

OPC UA Endpoint Suffix	Suffix is auto-generated and used only for an easy identification of the application based on the exposed URL. It has no functional usage on the communication establishment (usage of it into the UA Client connection is optional).
OPC UA Endpoint URL	The endpoint URL is auto-generated using the previously specified parameters. It is read only - the adjustments being done by modifying the other parameters.

Global Store&Forward Settings:

Attribute:	**Description:**
Validate Session Name	Validates the Session Name on Client Identification Process. After a network interruption or client communication lost - while a second session is tried being created using the same user authentication identities than the main session (the session created before communication lost) - the session name is validated against the previous session name - in case that the session name differs, then the second session is not accepted.
	Attention: The OPC UA Client used should have the possibility to set the session name and not generate a unique session name on each client connection try - otherwise the Client Identification will always fail while this setting is active/enabled.
	Note: On OPC UA Clients which did not allow configuration of session name on Client Connections, it is recommended to disable the option.
	The validation of session name is tried being verified when a session is tried being established (connected) on an active user which already has assigned a previously created (and not disconnected yet) session - basically the storing mechanism being active on actual user. By default, the validation of session name is enabled.
Validate Client IP Address	Validates the IP Address of the OPC UA Client which tries a connection re-establishment after a connection lost (due a network interruption or a client disruption). Basically, while a second session is tried being created using the same user authentication identities than the main session (the session which was created before communication lost) - the IP Address of OPC Client which tries a connection establishment is compared with the IP Address of the OPC Client which created the main session.
	Attention: There a OPC UA Clients which did not specify on connection establishment the IP Address of owning machine. Before activating it, ensure that the OPC Client is specifying the IP Address on Connect and Reconnect operations.

© Softing Industrial Automation GmbH

	By default, the validation of IP Address is disabled.
Update Server Timestamp	The server timestamp is set to actual time on each data-value when is received - basically, on each received data-value, internally the server timestamp is set to actual time. Enabling this option, will update the server timestamp with the actual time, each time when the server is trying to deliver data-value to the connected OPC UA Client.
	Attention: Each time when the OPC UA Client tries to deliver the same data-value, the server timestamp is updated with the actual time - the OPC Client can receive the same data-value for multiple times (when the confirmation is not received in time by the application) using different server timestamps.
	By default, the update of server timestamp is disabled.
Republish Same Value	The Republish service can be disabled, by disabling this option - basically the application ensure that all the values are received using only Publish services by the Store&Forward mechanism. In order to ensure that the same value is not received twice (via different services - Publish and Republish) the Republish service can be disabled.
	Note: The Republish service is still supported due the reason that non store and forward monitored items should benefit on this service - disabling it will only not sent the same value on a monitored item for a Republish operation on same sequence number.
	By default, the republish same value option is disabled.
Acknowledge Older Values	The received values are stored on the persistent storage and continuously tried being delivered until the confirmation is received from OPC UA Client side (immediately after the confirmation is received, the data-value is released and deleted) - enabling this option, when a confirmation is received for a sent data-value, all the older data-values which are not confirmed yet are released.
	Attention: Enabling this option, the data-integrity will not be guaranteed. This option should be enabled only when the OPC Client is more interested on preserving the data order (delivering the data-values on the same order than those data-values are received) than the data-integrity.
	By default, the acknowledge older values option is disabled.
Persistent Storage Path	The path to the persistent storage can be specified using this option.

Those settings are globally applied for all the configured users - being not uniquely configurable per each defined user (OPC UA Client session).

Application Certificates

Each defined endpoint (enabled or not) has assigned an own Public Key Infrastructure (PKI) store containing:

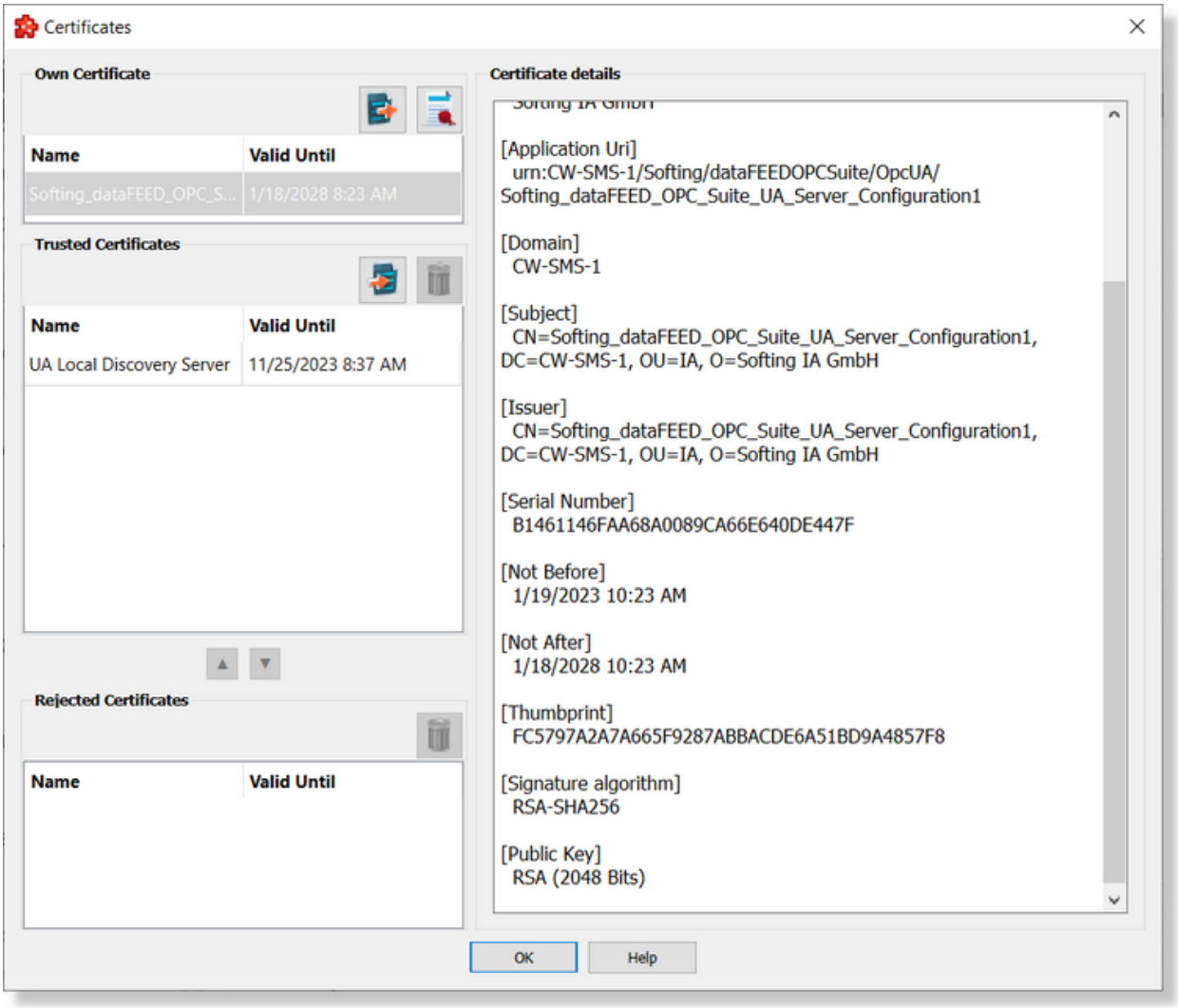
- Own Self Signed certificate including a private key (format used: Distinguished Encoding Rules (.der) with Privacy Enhanced Mail Certificate (.pem)).
- Trusted and Rejected List locations.

The PKI Store is auto-generated at the creation time of current configuration (for default configuration is it created at the installation time). The validity of the own certificate is 5 years. Also, the trusted relation with LDS (local discovery service) is automatically established.

Attention: The validation of the application certificate is operated by the OPC UA Client at the connection establishment time. All the validation rules accordingly with the OPC UA Specification V3.0 should pass. Due this reason, the own certificate is auto regenerated when an internal validation is failing (sample: changing the Hostname on working machine required a change of the used hostname on own certificate - re-generation automatically executed).

The PKI store can be manipulated (re-generation of own certificate or manipulation of trusted/rejected certificates) by pressing the corresponding "Configure" button on Application Certificates.

© Softing Industrial Automation GmbH



Left column contains Own certificate, Trusted certificates and Rejected certificates lists while the right column is populated with the details of the selected certificate.

Own Certificate.

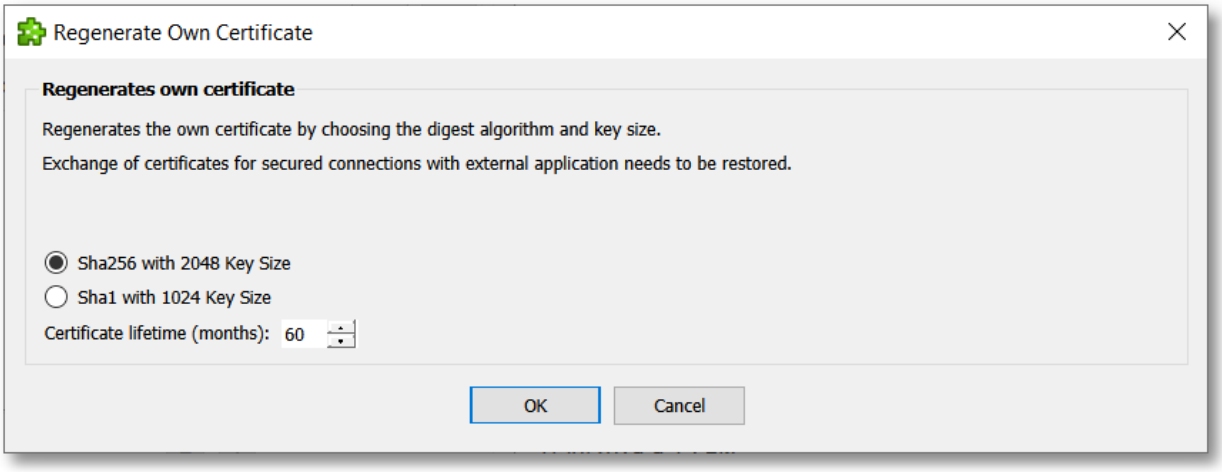
This list contains only one certificate and offers the possibility to be exported.

By using the export button the user is prompted to select a destination folder where to save dataFEED OPC Suite UA Server application certificate. This functionality is particularly useful when the Message security mode is other than "None".

The regeneration of own certificate is allowed by using the button. The own certificate can be generated by specifying the digest algorithm and encryption key size. By default the own certificate is regenerated using SHA256 digest algorithm and 2048 encryption key size independent of the settings from the previous certificate. Be aware that after a this operation the application own certificate was regenerated and all secured connections to the application should be reconfigured - the trust relation between application and connected servers/clients should be renew.

Also, In case of activation of Basic256Sha256 security policy - the regeneration of certificates using Sha1 digest algorithm with 1028 key size is forbidden.

Regeneration of certificates using Sha1 digest algorithm with 1024 key size will disable the Basic256Sha256 security endpoint.



Attention: The re-generation of certificates without saving the configuration is corrupting the actual configuration (a regeneration of certificates is overwriting the existing certificates - but the thumbprint is different (without saving the configuration, the application will try to use a certificate which was deleted).

The back-up/restore configuration operations are providing the possibility to operate also over the PKI store.

Attention: Restoring a configuration on a different machine, is requiring also a re-generation of own certificates due the reason that the identities (including hostname) were changed.

Trusted Certificates.

This group contains the list of the currently trusted certificates. The validity date can be easily checked on the "Valid Until" column. Certificates from this list can be dragged and dropped in the rejected certificate list.

By using the import button the user is prompted to select a server application certificate (.der format) that will be added to the trusted folder. This operation basically can substitute Receive server certificate operation.

One certificate can be deleted using button.

Rejected Certificates.

© Softing Industrial Automation GmbH

Contains the list of the rejected certificates. Each certificate can be dragged into trusted or deleted.

The up and down arrow buttons can be used to easily move certificates from trusted to rejected locations and vice-verse.

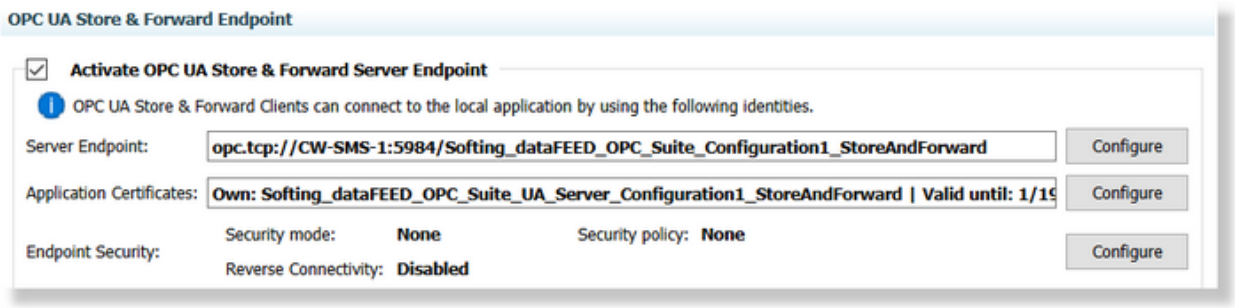
Certificate details.

In this area shows some details of the selected certificate like: location (on the disk), Application Name, Application Uri, Organization, Subject, Issuer, Validity dates and so on. The details area is read-only.

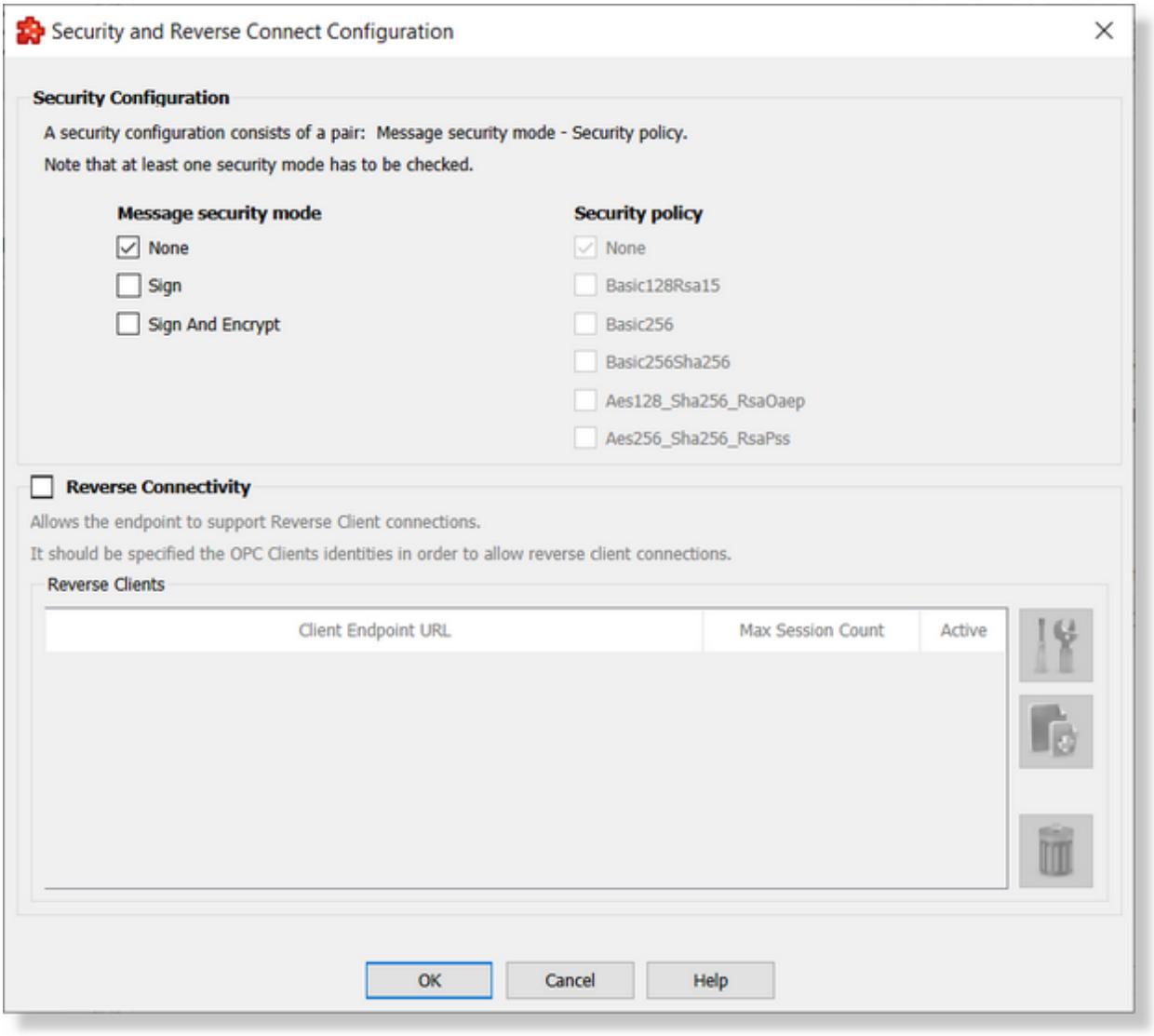
Security and Reverse Connect Configuration Settings

Allows the user to specify endpoint settings for the "**Message Security Mode**" and "**Security Policy**". Also, it is allowed to activate and configure the "**Reverse Connectivity**". This settings will be used by the OPC UA client for connection establishment - specially for defining the security degree of communication.

On the main OPC UA Client data destination page is it visible the lowest configured security mode.



Pressing the "**Endpoint Security**" corresponding "**Configure**" button is opening the "**Security and Reverse Connect Configuration**" dialog which allow you to specify the desired degree of security and the settings for reverse connectivity (if enabled).



Security Settings

At least one message security mode and one security policy have to be selected. It is also possible, to have multiple security configurations. That means, that more than one message security mode and security policy is selected.

As message security mode you can choose between none, a signed and a signed and encrypted mode.

The security policy can be not encrypted or you can choose between different security algorithms.

Attention: The configured security configuration should be synchronized with the OPC UA Client capabilities.

© Softing Industrial Automation GmbH

Note: The default security configuration is set to unsecured ("None" as Message Security Mode and Security Policy).

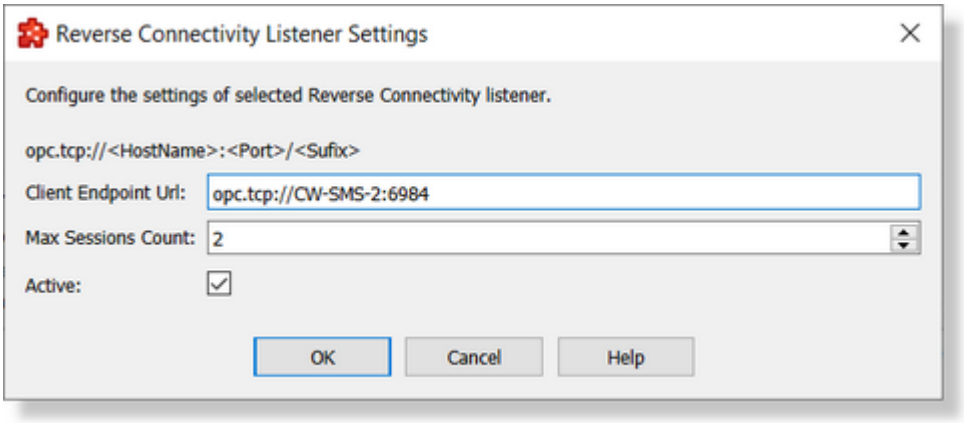
Also, in case of a usage of a secured configuration, the trusted relation between dataFEED OPC Suite and OPC UA client should be manually configured (the OPC UA Client's own certificate should be imported into the application trust list location and the application's own certificate should be exported into the trusted list location of the OPC UA Client).

Reverse Connect Settings

This settings allows the user to activate the **Reverse Connectivity** functionality.

The Reverse Connect feature, allows a server to initiate the connection to a client. The server tries cyclically to open a physical connections (sockets) to the client and send the "Reverse Hello message". A client (which supports Reverse Connect), is listening for such a socket and "Reverse Hello" messages will use this to create the "secured channel".

The functionality can be enabled by enabling the "Reverse Connectivity" check-box - it can be used only with at least one definition of a client listener. Defining or modifying the settings for a client listener can be done by pressing the "*Add a new Reverse Client*"  button or *"Properties of selected Reverse Client"*  button - immediately after pressing the buttons, the "Reverse Connectivity Listener Settings" dialog will be open allowing to specify the Reverse Listener settings.

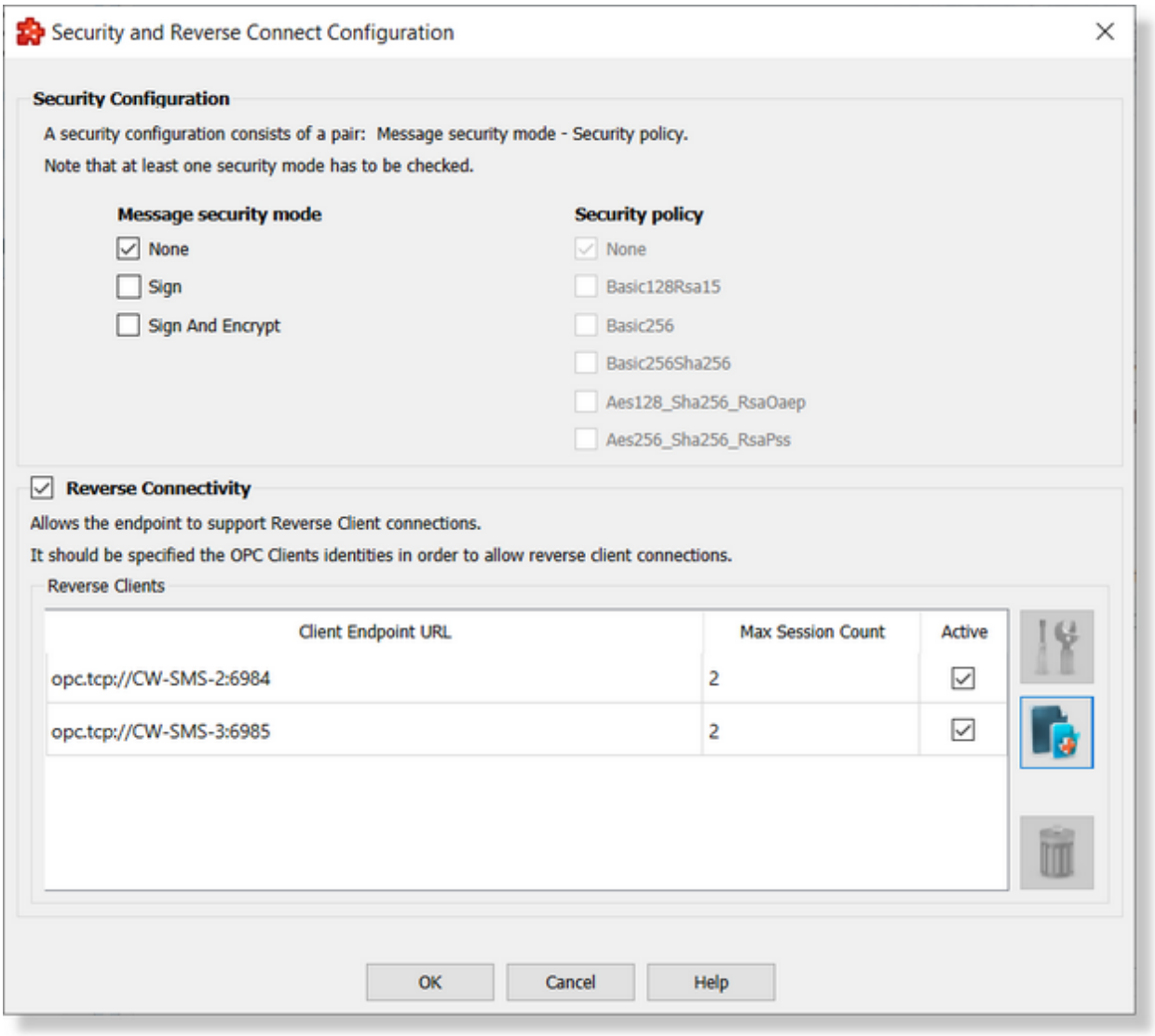


Attribute	Description
Client Endpoint URL	The URL where the OPC Client listens for reverse connections. The URL expression must comply with the OPC UA specifications. The port number must be an integer between 1024 and 65535.
Max Session Count	Maximum number of connections allowed for configured OPC Client.

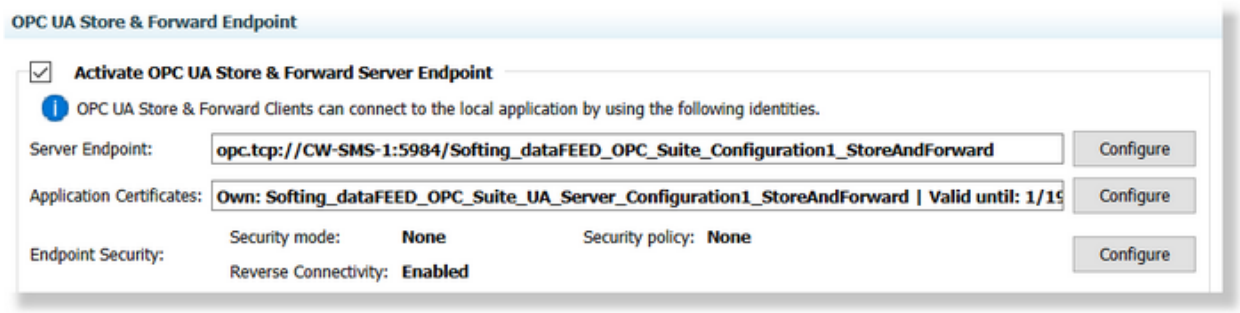
Active	State of configured OPC Client

Deleting an defined OPC Client Listener can be done by pressing the "*Delete the Selected Reverse Client*"  button.

The Store&Forward Endpoint supports maximum 5 active individual OPC Client connections.



© Softing Industrial Automation GmbH



Note: Enabling the Reverse Connections is not deactivating the Forward Connections.

Note: All the settings applicable a a session created on a Forward Connection are applicable for Reverse Connections too.

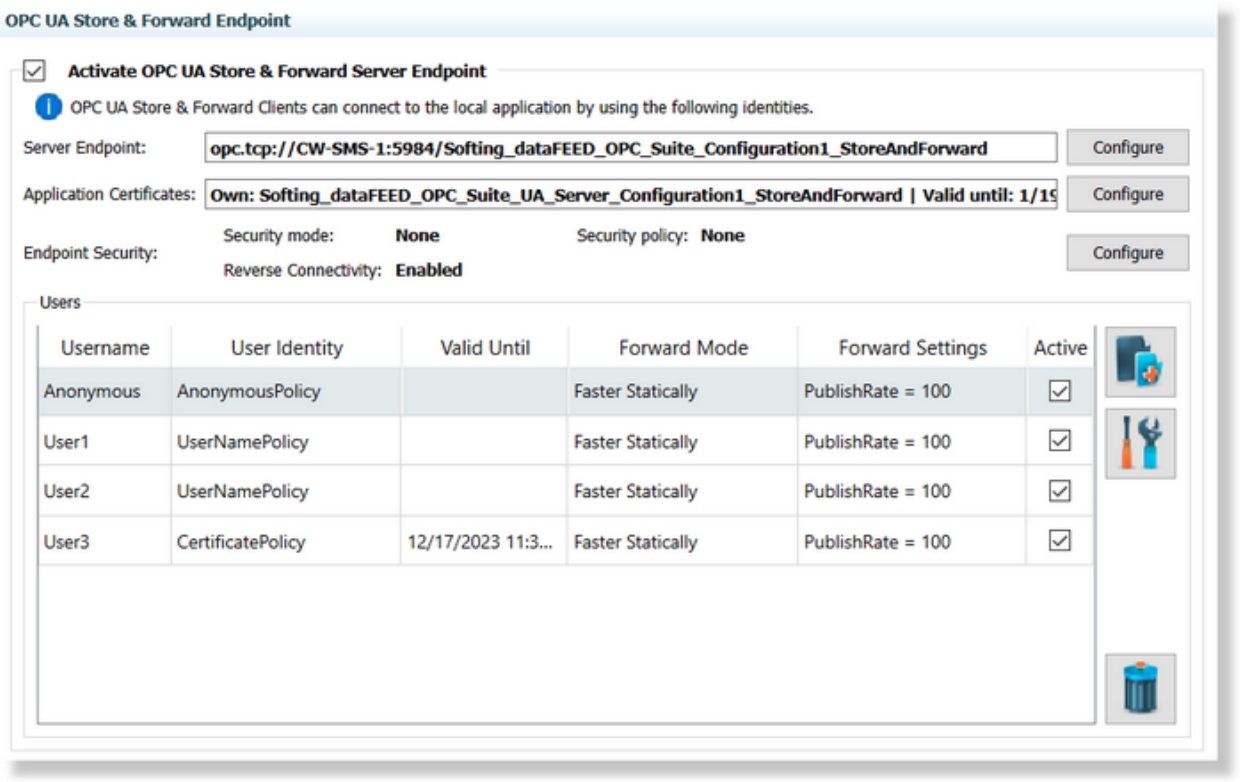
User Authentication Settings and Forward Settings

The user authentication settings are used by OPC UA Clients for building up a communication to dataFEED OPC Suite Standard Endpoint. The access to dataFEED OPC Suite can be protected by using a user name and password, as well as by using a certificate.

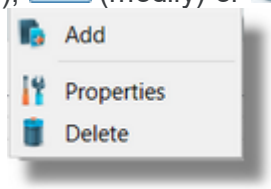
The Store&Forward functionality is limiting the parallel connected session to one for each defined user. Basically on each defined user, an OPC UA Client can establish only one connection (this connection/session becoming the the main session on corresponding user) and a second try on connection establishment will directly enter into the Client Identification Process - parallel connections being possible only inside Client Identification Timeout - immediately after the first session being forced closed.

The application is supporting up to 5 configured active users - together with this, the maximum number of parallel connections/sessions being limited to 5. At least one active user should be configured - the application not allowing the deletion of latest active user.

By enabling anonymous policy, there is no authentication necessary. By default only the anonymous policy enabled.



Only the active users are used by the application, all the other ones being ignored at the application start up time.

The application allow the user to add, modify or remove an authentication user. This operations are available based on right buttons:  (add),  (modify) or  (remove) - or based on a define menu (right click on user definition table) 

.

Each User Definition configures 2 distinctive set of settings

- User Authentication Settings
- Store&Forward Settings configurable for each defined user

User Authentication Settings:

Adding/Modifying a user operated by one designed user definition wizard page - responsible for user identification settings.

The application is supporting the following user authentication setting:

- Anonymous (no authentication being necessary).

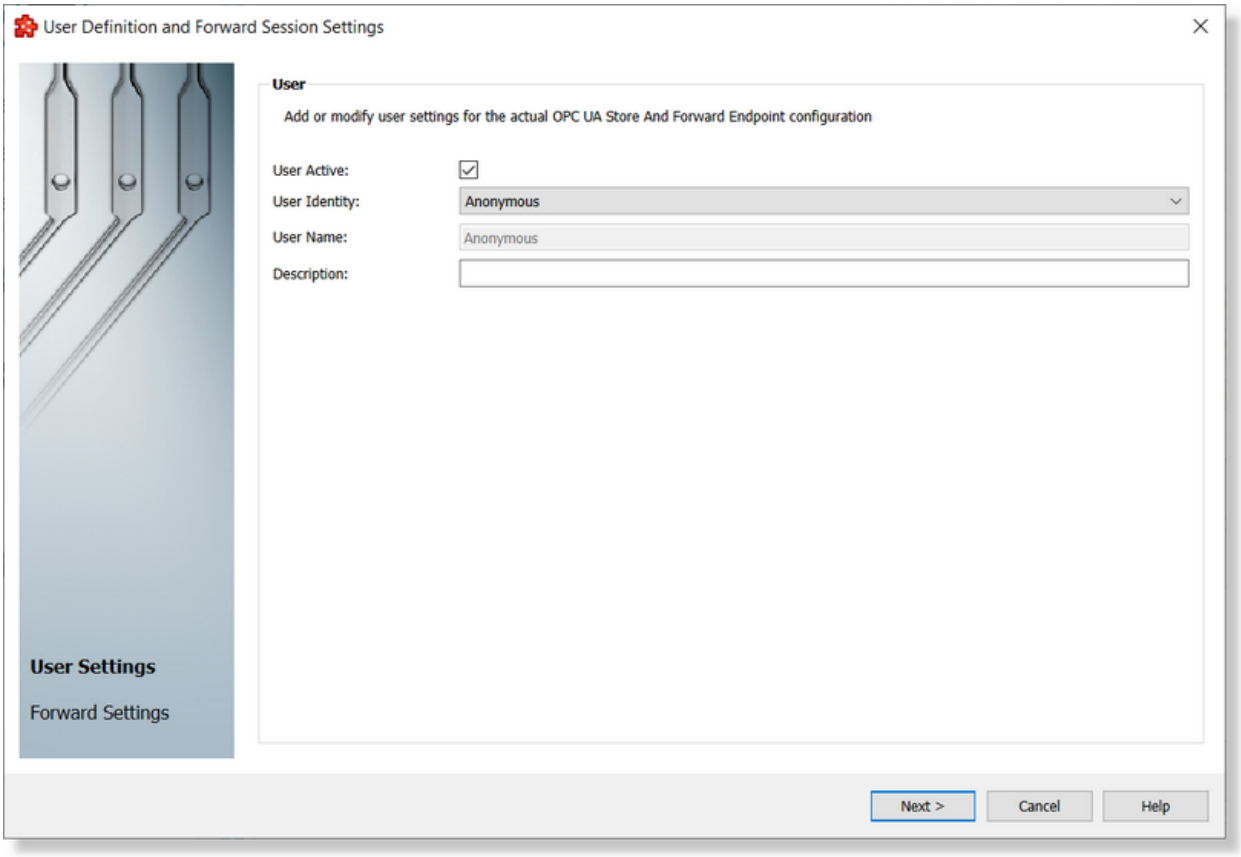
© Softing Industrial Automation GmbH

- Username and Password (the authentication is executed based on a set of credentials (username/password) - which is specified by the OPC UA Client - the matching between the active user (including the corresponding passwords) and the specified credentials by the OPC UA Client being mandatory).

Attention: The authentication based on username and password is using case sensitive policy.

- Certificate (the authentication is executed based on a user certificate - which is delivered by the OPC UA Client via OPC UA Protocol - the matching between the certificate of the active user and the certificate deliver by the OPC Client being mandatory).

Anonymous user



The user can be activate/deactivated (all the inactive users are ignored). By default, on a clean configuration, the Anonymous User is defined and active.

Note: The Anonymous User is intended being used only while unsecured policy is configured.

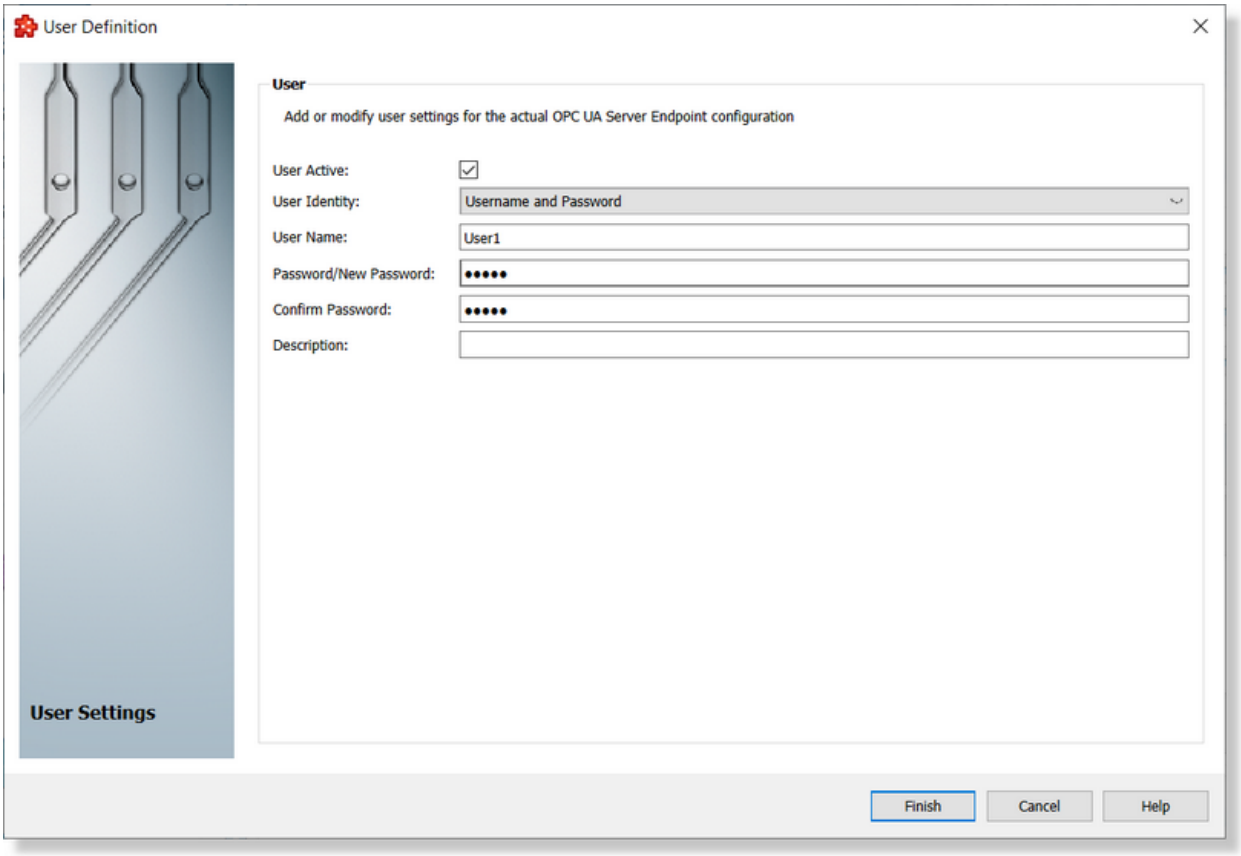
Note: Only one Anonymous User is possible being defined.

The "User Identity" type can be changed to:

- Username and Password

- Certificate

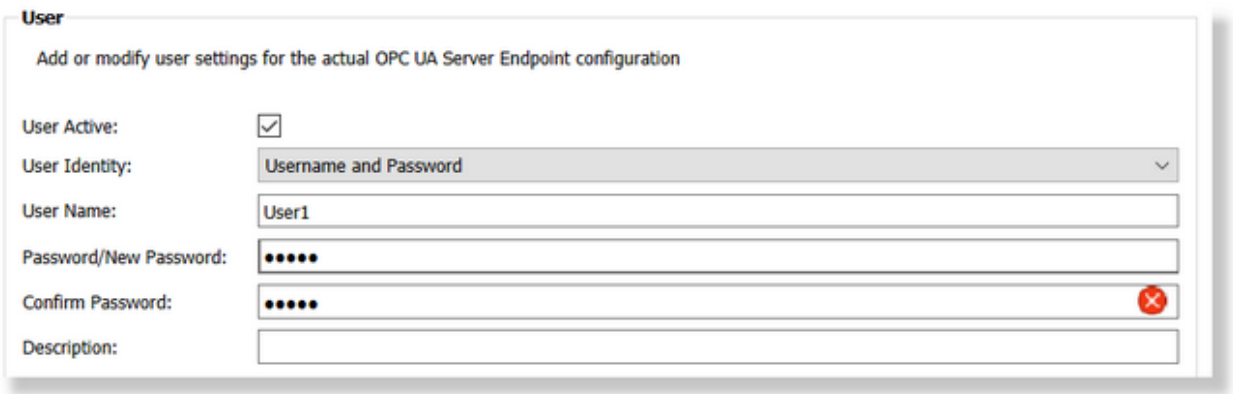
User identified by a set of credentials (user-name and password).



The user can be activate/deactivated (all the inactive users are ignored).

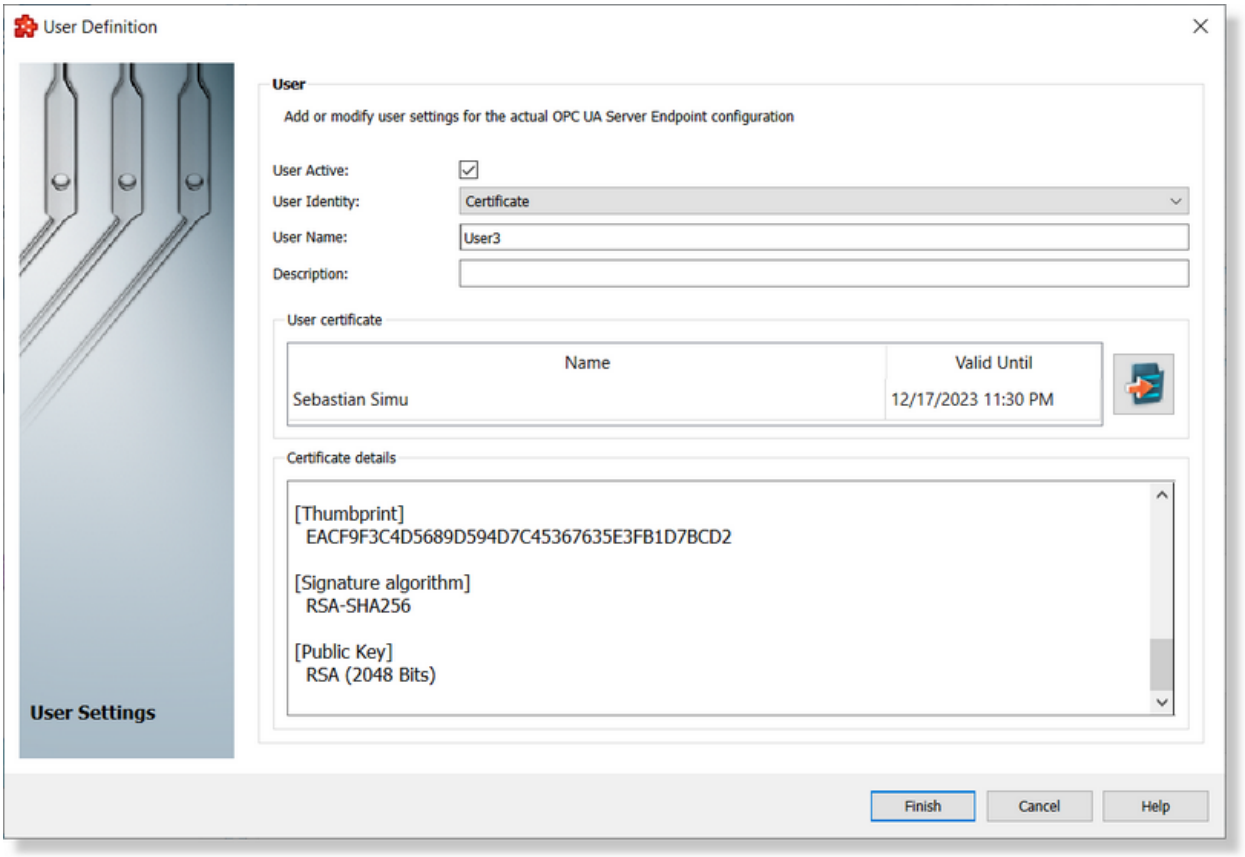
The username should be unique (it is not allowed to define duplicate users even if the specified password to identity type is different).

The confirmation of the password is mandatory - in case of failure, the user definition cannot be saved and the reason is visible on the wizard page:



© Softing Industrial Automation GmbH

User identified by a specified certificate



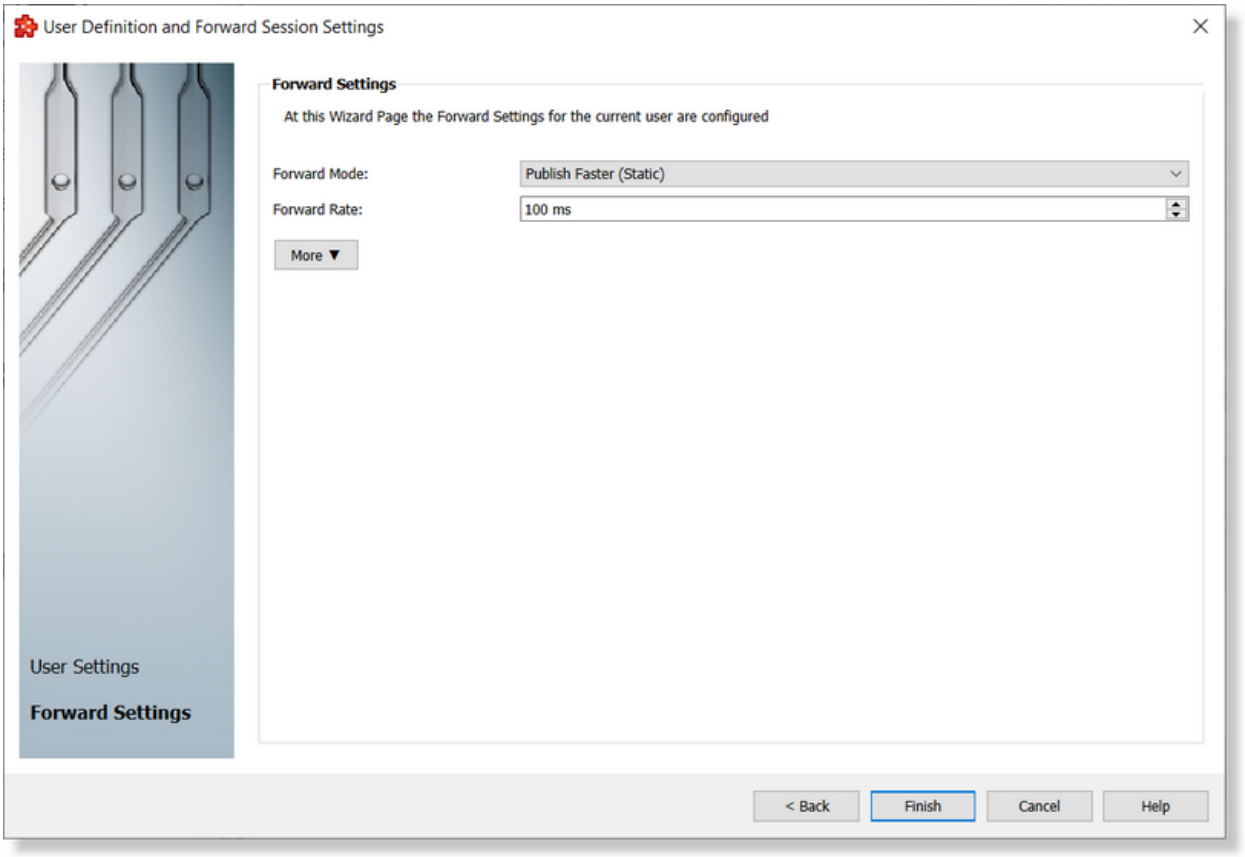
The user can be activate/deactivated (all the inactive users are ignored).

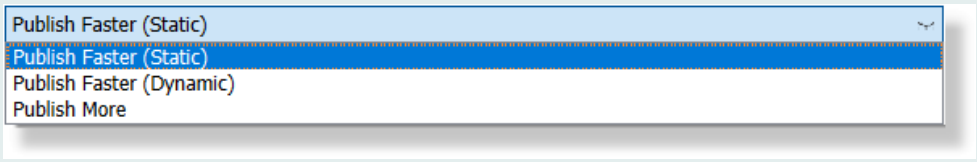
Importing a user certificate is mandatory. The supported certificates formats are: .der and .pfx. After a successfully import, the details of the imported certificates are visible on the wizard page.

Forward Settings:

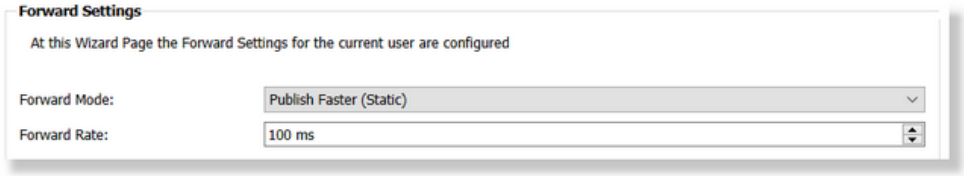
On the Authentication Settings (User Definition) second page - the main forward settings are configurable: Forward Type and Forward Rate (based on forward type is can be defined the forwarding rate or the forward queue size).

The settings are applied to the Data Forwarder (the mechanism responsible on delivering the stored data-values).



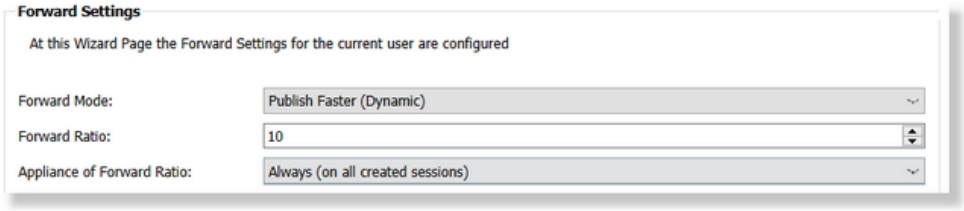
Attribute:	Description:
Forward Mode	The setting is specifying the forward mode, following options being available:  • Publish Faster (Static) - the Data Forwarder is publishing data-values with a faster frequency (the frequency being statically configurable as "Forward Rate". • Publish Faster (Dynamic) - the Data Forwarder is publishing data-values with a faster frequency (the frequency being dynamically configurable as "Forward Ratio". • Publish More - the Data Forwarder is publishing data-values in a bigger volume with the same frequency than the connected client is requiring. The volume is specified based on "Queue Size".
Forward Rate / Forward Ratio / Queue Size	• **Forward Rate** - the option is available only for Forward Mode - Publish Faster (Static), it is specifying the frequency of publish operations - basically the value is statically configured (using millisecond as measurement unit). The publish interval of a created session is automatically revised to configured "Forward Rate" value.

© Softing Industrial Automation GmbH



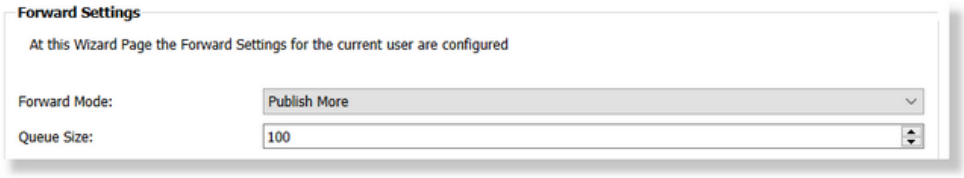
- **Forward Ratio** - the option is available only for Forward Mode - Publish Faster (Dynamic), it is specifying the the frequency of publish operations based on a division ratio of requested Publish Interval - basically the value is dynamically applied based on requested Publish Interval (by the connected OPC Ua Client) which is revised to the required value divided by the ratio.

Sample: Requested Publish Interval - 1000ms, Forward Ratio - 10 then the Revised Publish Interval is set to 1000ms/10 = 100ms.

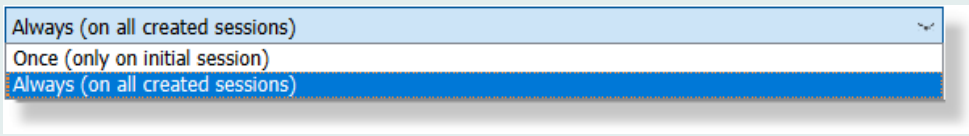


The forward ratio is applied until the minimum supported value for Publish Interval is reached (10ms).

- **Queue Size** - the option is available only for Forward Mode - Publish More, it is specifying the volume of the data-values sent on the same publish operation on a monitored item. Basically, while forwarding data-values process is running, the values is used as revised monitored item queue size - each publish operation delivering a specified number of stored data-values (if exists).



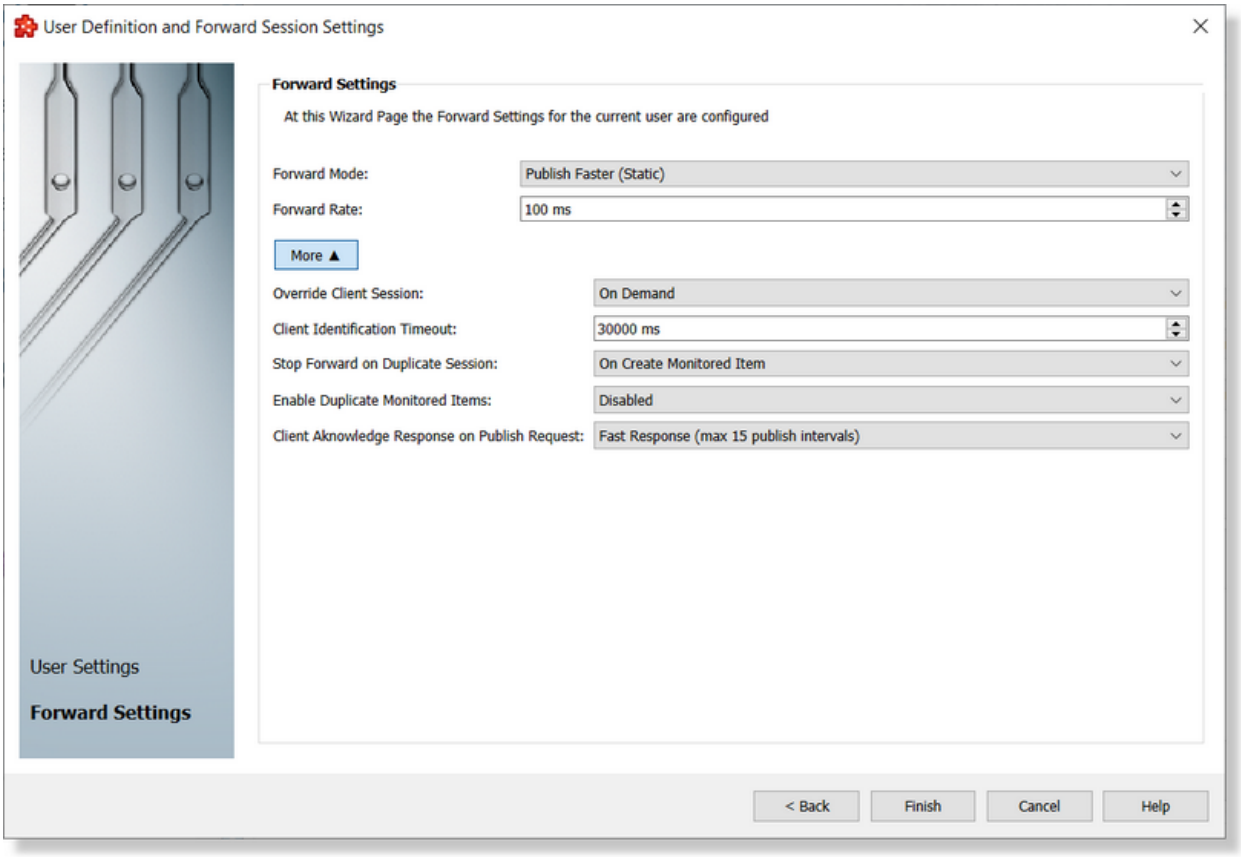
Note: on this forward mode, the frequency of forwarding data is not adapted, being used the one required by the OPC UA Client.

Appliance of Forward Ratio	This option is available only for Forward Mode - Publish Faster (Dynamic). It is specifying when to apply the division ratio to the Publish Interval.  The available options are: • Once (only on initial session) - applies the forward ratio (revises the publish interval to the initial value divided by the ratio) only while there is no active session on current authentication user - on reconnect routine the division being not applied.

	Attention: In case that the OPC UA Client is preserving the revised rate (on re-connect or on next session creation) then the ratio should be applied only on initial session. This option should be used in order to void recurrence appliance of division ratio. • Always (on all created sessions) - applies the forward ratio (revises the publish interval to the initial value divided by the ratio) on all created session. **Attention:** This option should be used for OPC UA Clients which are not preserving the revised publish interval.



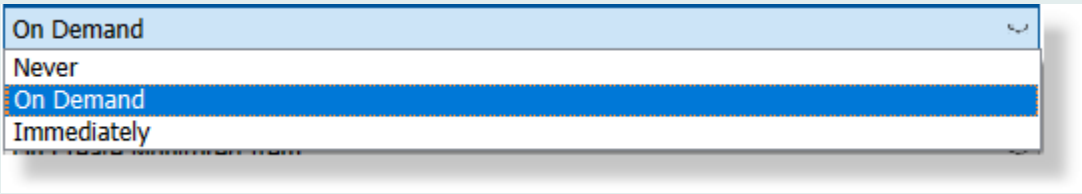
The **Advanced Store&Forward Settings** are shown by pressing the "More" button.



The following configuration settings are accessible and configurable:

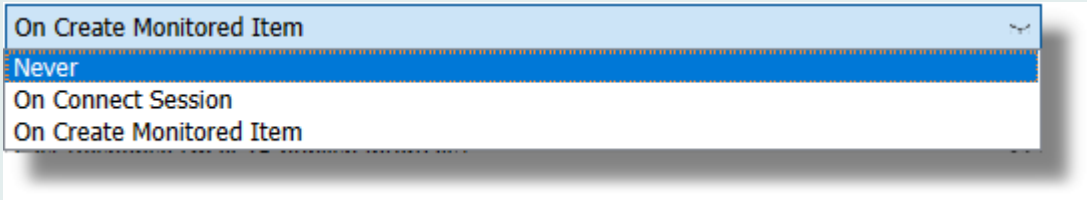
Attribute:	Description:

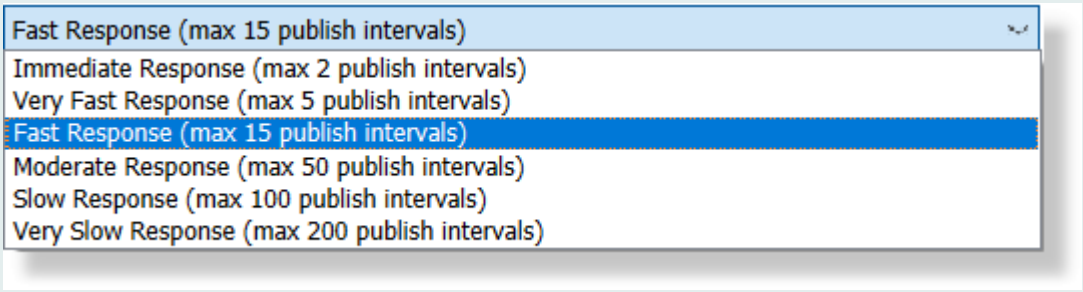
© Softing Industrial Automation GmbH

Override Client Session	The setting is specifying when a second session using the same user authentication identities than an existing one (usually this should happen after a unattended client disconnection based on a network interruption or client disruption) is accepted as potential "Forward" OPC UA Client (the ownership of stored data on previous session being transferred to actual session). The following options are available:  • **Never** - it is never accepted a creation of a session using an user authentication identities while the same user was used for a previously created session and the session was not disconnected by the OPC UA Client or the session timeout expiration. Basically, until the initial session is disconnected by the OPC Client which connected it or there is no communication inside the session timeout interval - the second session using the same user authentication identities is not accepted - the initial session being never overwritten by a new session. **Attention:** Enabling this setting will remove the possibility to reconnect an OPC UA Client in case of any failure at the OPC UA Client level inside the Session Timeout interval (60 minutes) - no other sessions being allowed on current user. • **On Demand** - the second session using the same user authentication identities than an existing one at the server side is accepted - but its identification as "**Forward Client**" will be done later when the second session will be validated as mirroring of initial session (the same subscriptions with the same monitored items will be identified for the second session that the one from the initial session). After the second session will be identified as "Forward Session" - the application will start forwarding stored data-values - the second session taking the ownership of stored data-values. In case that the Client Identification cannot be performed inside the Client Identification Timeout then ownership of stored data will be lost and the stored data-values released. **Note:** While the Client Identification mechanism is in progress, all created monitored items on the second session will receive a data-change with "Bad_WaitingForInitialDatachange" as status code. **Note:** The Client Identification mechanism will be started immediately after first monitored item will be created on the second session. After the Client Identification is finalized (successfully or not) *the initial session is forced closed and the ownership of stored data-values will be transferred or lost*. **Attention:** Immediately after the second session is created and accepted as potential "Forward Client" - the Client Identification process is started - the final result (inside the Client Identification timeout) being irreversible.

	Attention: The second session is identified as "Forward Client" only after the same subscriptions and monitored items will be created - in case that this is not done inside the Client Identification timeout then all the stored data-values will be lost, the second session becoming the main session without any references to the previous session.
	• **Immediately** - the second session using the same user authentication identities than the existing one at the server side is automatically accepted and identified as "Forward Client". Immediately after a monitored item is created, the stored data-values (if exists) are started being transferred (forwarded).
	Note: The monitored items created which were not part of previous session will receive immediately current data-values (those are generated by the corresponding data source).
	Note: The initial session is automatically force closed when the second session is identified as mirroring of initial session or the Client Identification timeout expired.
	Attention: The stored data-values for monitored items which are not created inside the Client Identification timeout are automatically released - being impossible their recovery.
Client Identification Timeout	The maximum time used for the application to identify that a second session using the same user authentication settings than an existing one is accepted as "Forward Session". A session is accepted as forward session based on the value of Override Client Session attribute: • **Never** - never identify as forward session. • **On Demand** - only after a second session becomes a mirroring of initial session (all the subscriptions and monitored items from initial session being created and activated on the second session). No additional subscriptions or monitored items being accepted. • **Immediately** - immediately after a second session is created.
	Note: The second session should be created using the same user authentication identities than an existing session and all the validations being passed.
	Attention: After the expiration of Client Identification timeout, all the stored data-values on monitored items which are not transferred to the second session are released. After this operation being lost without any recovery possibilities.
Stop Forward on	This option is specifying when to stop forwarding data-values on existing monitored items from a session when a second session using the same user authentication settings is created and accepted - basically starting with this point, on the same

© Softing Industrial Automation GmbH

Duplicate Session	defined user exits 2 active sessions (the initial one being force closed after the identification of the second one as "Forward Client" or the expiration of Client Identification timeout.
	Note: This setting is applied only for the time interval between the acceptance of second session and its identification (successfully or not) as "Forward Client" - basically for an interval which is maximum equal with the Client Identification timeout.
	The following options are available  • **Never** - the data-values (stored or current) are continuously delivered to the initial session until the session is force closed.
	Attention: sending data-values in parallel to 2 open and active sessions (the initial session to the OPC UA Client being still open and active) on the same monitored items (when this are created) will have as result that some store data-values will be sent to initial session and others to second session - basically both session losing data. The option should be used only the it is sure that the initial session was lost (disconnected required by the OPC Client while the network was interrupted).
	• **On Session Connect** - the data-values (stored or current) are stopped being delivered to the initial session immediately after a second session is created and accepted using the same user authentication identities than the current one. All monitored items from current session will received a last data-value with current timestamp and "Bad_Disconnected" as status code - no other data-values being received anymore on this session. • **On Create Monitored Item** - the data-values (stored or current) are stopped being delivered to the initial session only after at least one monitored item is created on a second session which was previously created and accepted on the same user authentication identities than the current one. All monitored items from current session will received a last data-value with current timestamp and "Bad_Disconnected" as status code - no other data-values being received anymore on this session.
Enable Duplicate Monitored Items	The option is enabling the possibility to create duplicate monitored items o the same session. By default, the application is not supported creation of duplicate monitored items on the same session - the creation of a duplicated monitored item being rejected at the creation time with "Bad_TooManyMonitoredItems" status code. Enabling the support for duplicate monitored items will allow the creation of the duplicated monitored item - but for the initial monitored item it will be send a final data-change with "Bad_Disconnected" - not other data-values being sent anymore.

	Note: The option is needed when an OPC UA Client is operating delete monitored items service while the connection to the server was interrupted - immediately after the communication reestablishment without creating a second session - the OPC UA Client can try to re-create the subscriptions and monitored items (at that time the monitored items are not deleted at the server side - the deletion of monitored items being done without active communication).
Client Acknowledge Response on Publish Request	The application is continuously sending the same data-value on a monitored item until the confirmation is received from the connected OPC UA Client (the confirmation is received based on acknowledgment on sequence number used on Publish services while a data-value is delivered to the connected OPC UA Client). Immediately after the confirmation is received for a data-value the application is starting delivering the next data-value. The option is specifying how many cycles (Publish Intervals) to wait for a confirmation until trying to re-sent the same data-value. The available options are:  • Immediate Response (max 2 publish intervals) • Very Fast Response (max 5 publish intervals) • Fast Response (max 15 publish intervals) • Moderate Response (max 50 publish intervals) • Slow Response (max 100 publish intervals • Very Slow Response (max 200 publish intervals)
	Note: The setting is also responsible with the moment when the first data-value is tried being sent after a communication re-establishment - Basically the first data-value will be sent with a delay of maximum "Client Acknowledge Response on Publish Request" Publish Intervals.
	Note: Using a reduce value for this setting will increased the probability to send duplicates (same data-value received by the OPC UA Client for multiple times).
	The value should reflect the OPC UA Client capabilities reported to the number of monitored items on the same session.

© Softing Industrial Automation GmbH

Limitations and Corner Cases

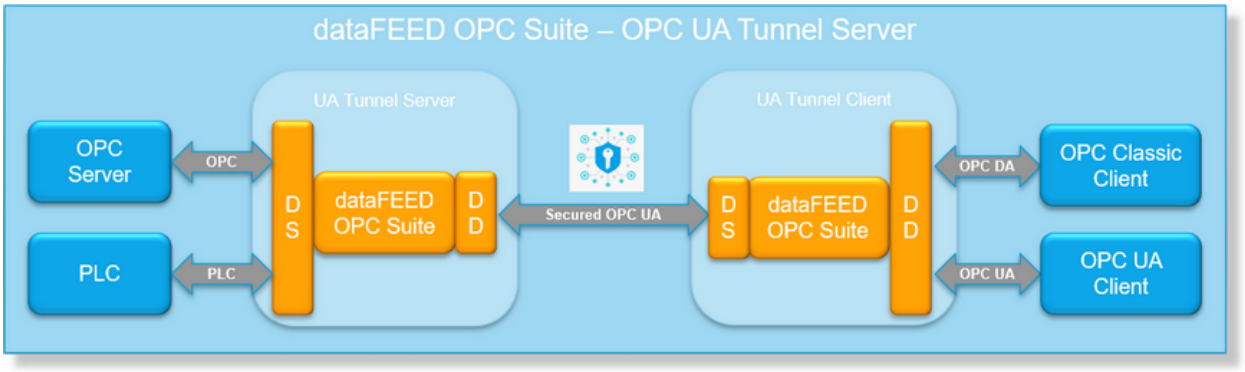
The Store&Forward UA Server endpoint is exposing a UA Server with a set of limitations - the decision on usage it, should be highly analyzed before being used.

Limitations:
Limited to only one active session for each defined user authentication settings - it is not allow the have active multiple parallel sessions (beside the time while the Client Identification mechanism is in progress) - for each defined user authentication settings only one active session can be used.
Maximum number of allowed user authentication definitions limited to 5 - the application is limiting the number of active users to 5, based on this and the fact that only one active session is allowed for each defined use - the maximum number of allowed parallel active sessions will be limited to 5 too.
Maximum number of monitored item per session is limited to 1000 - the maximum number of allowed monitored items is limited to 1000 per defined/created session. When the limit is reach, each Create Monitored Item service call will failed with "Bad_TooManyMonitoredItems" status code.
Duplicate monitored items are not allowed or will be not deserved with data-values - The creation of a duplicate monitored items (same item used twice on the same subscription with the same major settings) is not allowed or deserved with data-values.
The Store&Forward functionality is not compatible with OPC UA Clients which does not acknowledge the publish operations - the entire forward mechanism is delivering data based on the received acknowledge on previously sequence number from the previously executed Publish service - without receiving the confirmation on a sent data-value - the same data-value will be sent forever to the connection OPC UA Client.
The maximum supported network interruption is 1 hours - the revised session timeout being set to 1 hour.
Performance penalties on sending data-value - this performance penalties comes due the reason that processed operation on this functionality is time consuming (each received data-value being stored and continuously re-sent until the confirmation is received from OPC UA Client side.

Licensing

The functionality is licensed exclusively by **dataFEED OPC Suite Extended license**. *Demo Mode* is working on full functionality for 72 hours.

Attention: The **dataFEED OPC Suite license** is not licensing the OPC UA Store&Forward functionality (without **dataFEED OPC Suite Extended license** - the OPC UA Store&Forward Endpoint will be not open).

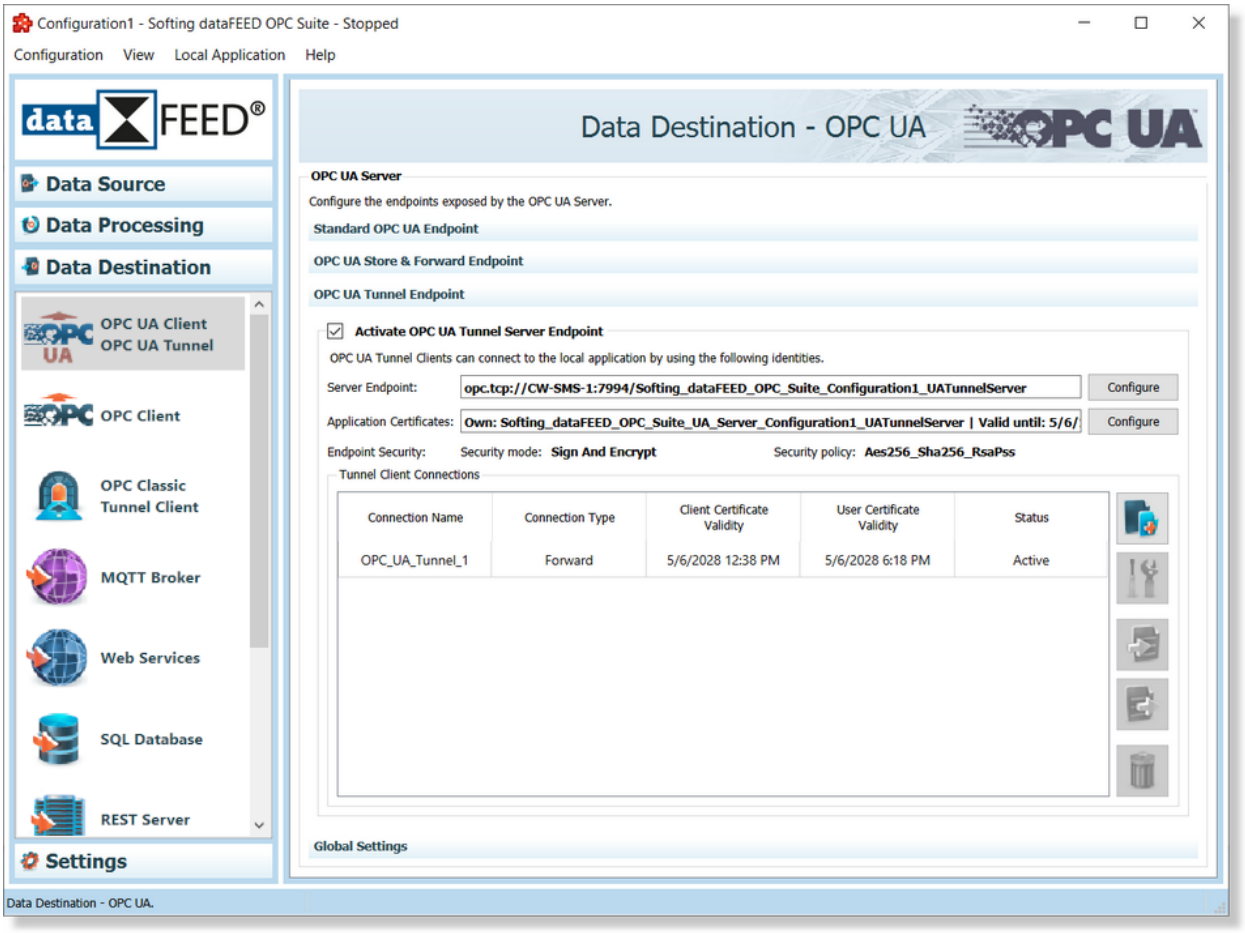


The OPC UA Tunnel Functionality allows 2 instances of Softing dataFEED OPC Suite to communicate over standard OPC UA protocol with a high degree of security.

The OPC UA Tunnel Server endpoint is enabled by the activation of "**Activate OPC UA Tunnel Server Endpoint**". By default it is disabled. Being enabled, the OPC UA Tunnel Server endpoint is open and accessible on the exposed OPC UA Server over the OPC UA protocol. Any OPC UA Tunnel Clients (other instances of dataFEED OPC Suite product) will be able the establish connections to the exposed OPC Server over the exposed endpoint and operate on standard OPC UA services.

Attention: Enabling the OPC UA Tunnel endpoint should be followed by the definition of at least one active OPC UA Tunnel connection. The endpoint is automatically disabled when no active and valid OPC UA Tunnel connections are defined.

© Softing Industrial Automation GmbH



Note: The OPC UA Tunnel functionality is designed to replace the OPC Classical Tunnel functionality operating over the secured OPC UA protocol. The purpose of OPC UA Tunnel Server Endpoint is to allow a secured communication between multiple instances of dataFEED OPC Suite running on different machines and avoiding the potential security communication issues.

Basically, the exposed endpoint is providing only a secured communication to other dataFEED OPC Suite instances.

Note: Configuration of security mode and policy is disabled. Security mode **Sign and Encrypt** and security policy **Aes256_Sha256_RsaPss** are the only available options.

Disabling all endpoints (***Standard OPC UA Server Endpoint***, ***OPC UA Store&Forward Endpoint*** and ***OPC UA Tunnel Server Endpoint***) will completely deactivate the OPC UA Server functionality.

The following UA Server Endpoint specific settings are configurable:

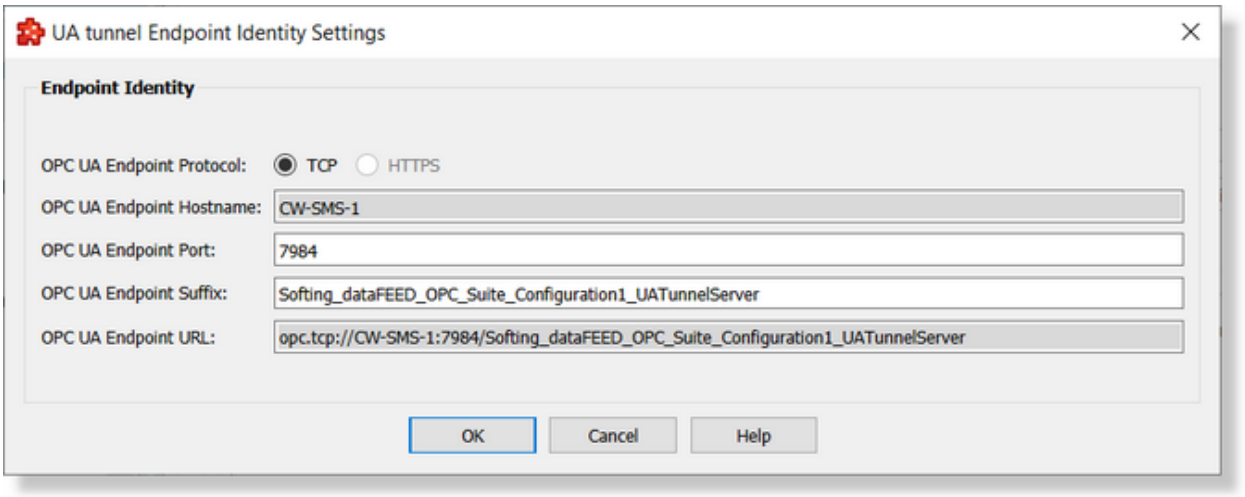
- Server Endpoint identities
- Application Certificates

- Security Configuration settings
- Tunnel Client Connections

OPC UA Server Endpoint Identities:

The auto-generated UA Tunnel Server endpoint URL is accessible on main configuration page, the access rights to it is limited to read only (copy operations being available).

The server endpoint URL is used by the internal OPC UA Tunnel Client of dataFEED OPC Suite for communication establishment. The server endpoint URL and other basic endpoint communication identities are configurable via "**Endpoint Identity Settings**" dialog which is activating by pressings the "**Configure**" button.



Attribute:	Description:
OPC UA Endpoint Protocol	Current version of dataFEED OPC Suite (version V5.38) is supporting only TCP protocol. HTTPs support being under development.
OPC UA Endpoint Hostname	Configured Hostname on current machine. By default, it is using the value from the Operating System. The Hostname is configurable as global configurable settings common for both exposed endpoints.
OPC UA Endpoint Port	Configured port for TCP communication on local machine. The port will be used by the internal OPC UA Client of dataFEED OPC Suite for connection establishment. By default, the configured port is auto-generated using an unique value for each defined dataFEED OPC Suite configuration.
	Attention: Please ensure that the configured port is not used by other running instance (in case that it was manually configured) or by other running application on current machine. Usage of same port on TCP communication by 2 different processes is not allowed by the Operating System.

© Softing Industrial Automation GmbH

OPC UA Endpoint Suffix	Suffix is auto-generated and used only for an easy identification of the application based on the exposed URL. It has no functional usage on the communication establishment (using it into the UA Client connection being optional).
OPC UA Endpoint URL	The endpoint URL is auto-generated using the previously specified parameters. It is read only - the adjustments being done by modifying the other parameters.

Application Certificates

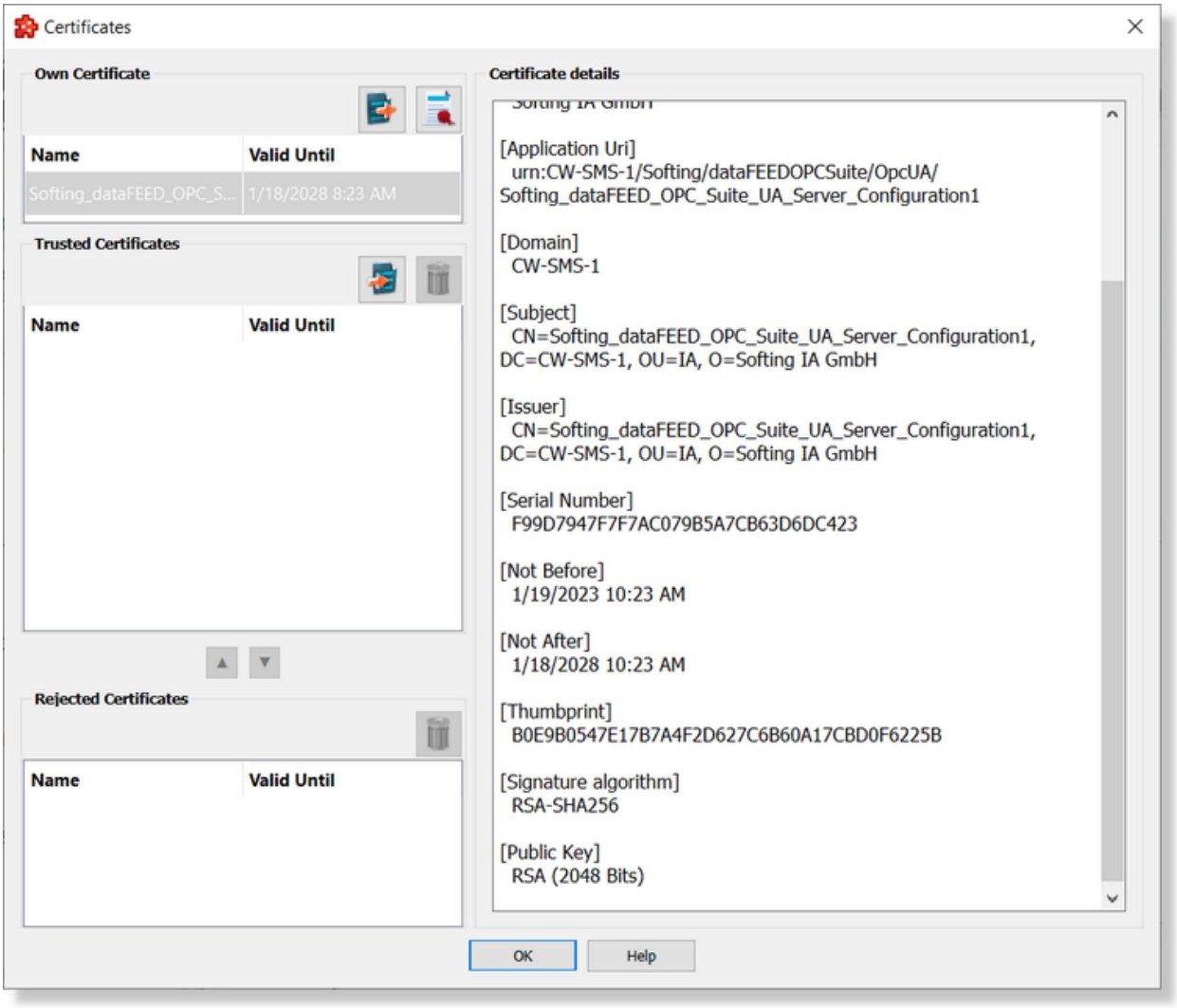
Each defined endpoint (enabled or not) has assigned an own Public Key Infrastructure (PKI) store containing:

- Own Self Signed Certificate including a private key (format used: Distinguished Encoding Rules (.der) with Privacy Enhanced Mail Certificate (.pem)).
- Trusted and Rejected List locations.

The PKI Store is auto-generated at the creation time of current configuration (for default configuration is it created at the installation time). The validity of the own certificate is 5 years.

Attention: The application certificate is validated by the OPC UA Tunnel Client at the connection establishment time. All the validation rules accordingly with the OPC UA Specification V3.0 should pass. Due this reason, the own certificate is auto regenerated when an internal validation is failing (sample: changing the Hostname on working machine required a change of the used hostname on own certificate - re-generation automatically executed).

The PKI store can be manipulated (re-generation of own certificate or manipulation of trusted/rejected certificates) by pressing the corresponding "**Configure**" button on Application Certificates.



Left column contains *Own* certificate, *Trusted* certificates and *Rejected* certificates lists while the right column is populated with the details of the selected certificate.

Own Certificate.

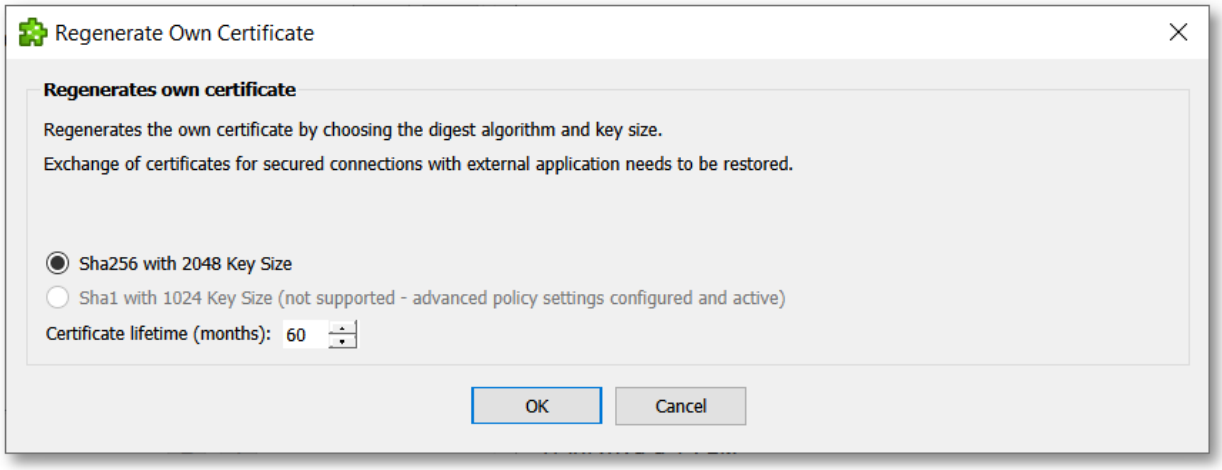
This list contains only one certificate and offers the possibility to be exported.

By using the export button the user is prompted to select a destination folder where to save dataFEED OPC Suite UA Server application certificate. This functionality is particularly useful when the Message security mode is other than "None".

The regeneration of own certificate is allowed by using the button. The own certificate can be generated by specifying the digest algorithm and encryption key size. By default the own certificate is regenerated using SHA256 digest algorithm and 2048 encryption key size independent of the settings from the previous certificate. Be aware that after this operation the application own certificate was regenerated and all secured connections to the application should be reconfigured - the trust relation between application and connected servers/clients should be renew.

© Softing Industrial Automation GmbH

Regeneration of certificates using Sha1 digest algorithm with 1024 key size is disabled due the reason that only secured connections based on **Aes256_Sha256_RsaPSS** security policy is allowed.



Attention: The re-generation of certificates without saving the configuration is corrupting the actual configuration (a regeneration of certificates is overwriting the existing certificates - but the thumbprint is different (without saving the configuration, the application will try to use a certificate which was deleted).

The back-up/restore configuration operations are providing the possibility to operate also over the PKI store.

Attention: Restoring a configuration on a different machine, is requiring also a re-generation of own certificates due the reason that the identities (including hostname) were changed.

Trusted Certificates.

This group contains the list of the currently trusted certificates. The validity date can be easily checked on the "Valid Until" column. Certificates from this list can be dragged and dropped in the rejected certificate list.

By using the import button the user is prompted to select a server application certificate (.der format) that will be added to the trusted folder. This operation basically can substitute Receive server certificate operation.

One certificate can be deleted using button.

Rejected Certificates.

Contains the list of the rejected certificates. Each certificate can be dragged into trusted or deleted.

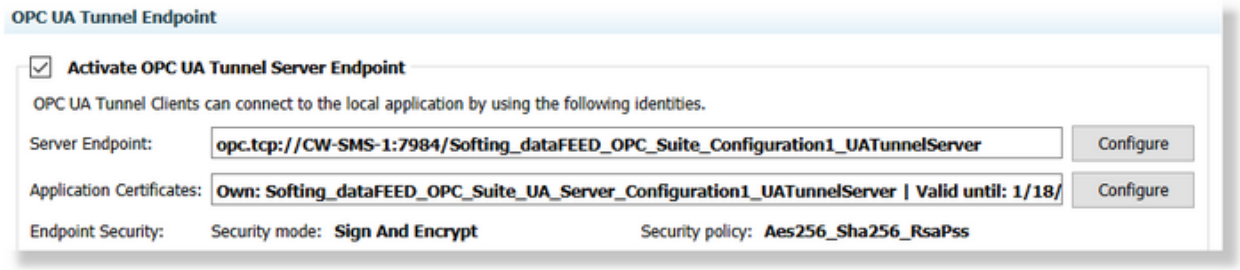
The up and down arrow buttons can be used to easily move certificates from trusted to rejected locations and vice-verse.

Certificate details.

In this area shows some details of the selected certificate like: location (on the disk), Application Name, Application Uri, Organization, Subject, Issuer, Validity dates and so on. The details area is read-only.

Security Settings

Allows the user to visualize the endpoint settings for the "**Security Mode**" and "**Security Policy**". This settings will be used by the OPC UA Client for connection establishment - specially for defining the security degree of communication.



Attention: Configuration of security mode and policy is disabled. Security mode **Sign and Encrypt** and security policy **Aes256_Sha256_RsaPss** are the only available options.

Tunnel Client Connections

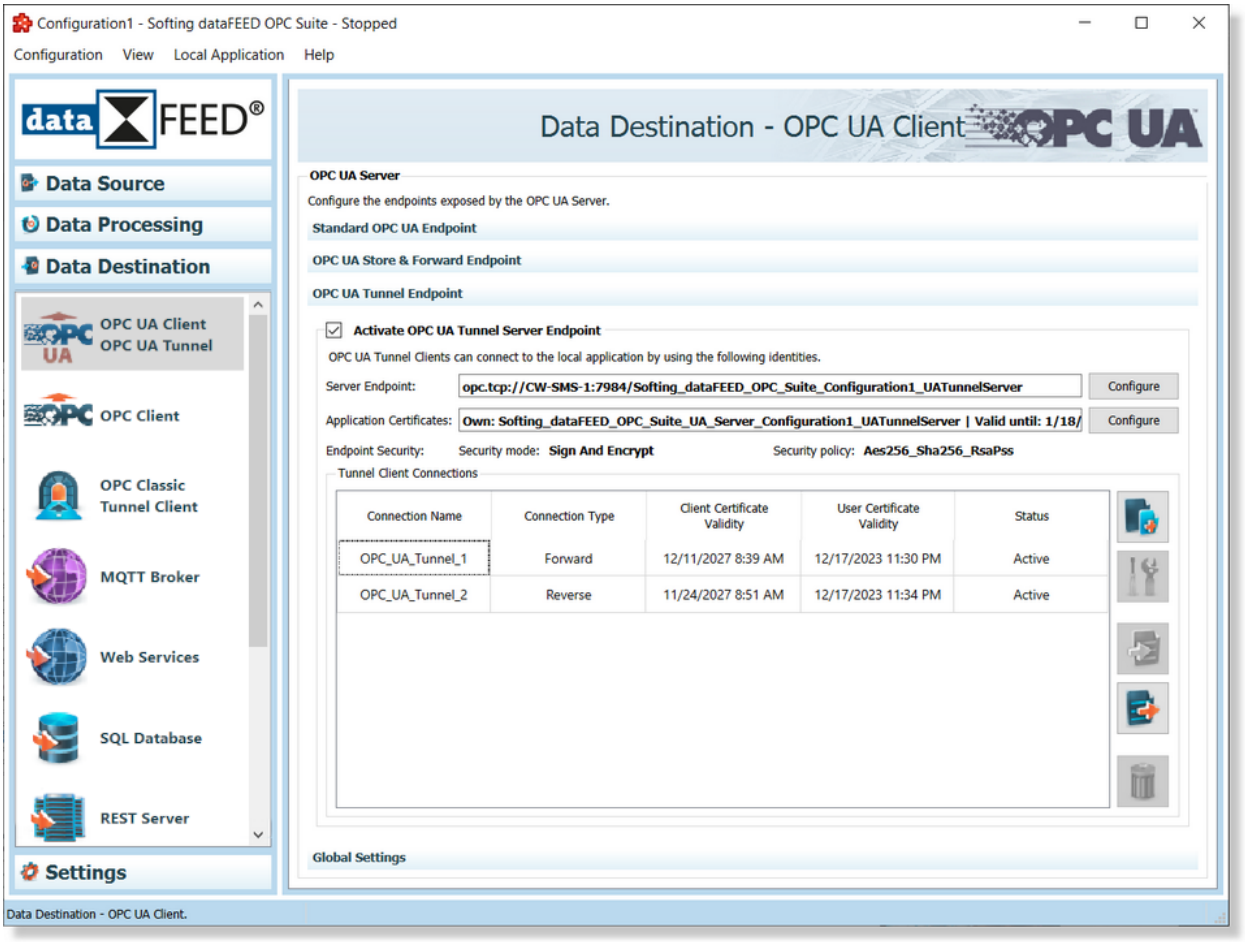
The definition of UA Tunnel connections allows to define, synchronize and ensure the trust relation (used for secured communication) between dataFEED UA Tunnel Server side (as data destination) and dataFEED UA Tunnel Client side (as data source). The connection parameters and security settings (including the certificates exchange) should be configure on the next steps.

For each defined connection, it is specified the validity of **Client Application** and **User Authentication** certificates

Note: In case of missing of certificates this is signaled on the Tunnel Client Connections table.

In addition, the *Connection Name, Connection Type and Connection Status* for each defined connections are visible on the Tunnel Client Connections table.

© Softing Industrial Automation GmbH



Attention: Having no active connections will not open the OPC UA Tunnel Endpoint at run-time. At least one UA Tunnel connection should be defined and activated in order to open the UA Tunnel Endpoint while running the dataFEED OPC Suite data destination.

The Connection Type (Forward or Reverse) is possible being configured for each connection.

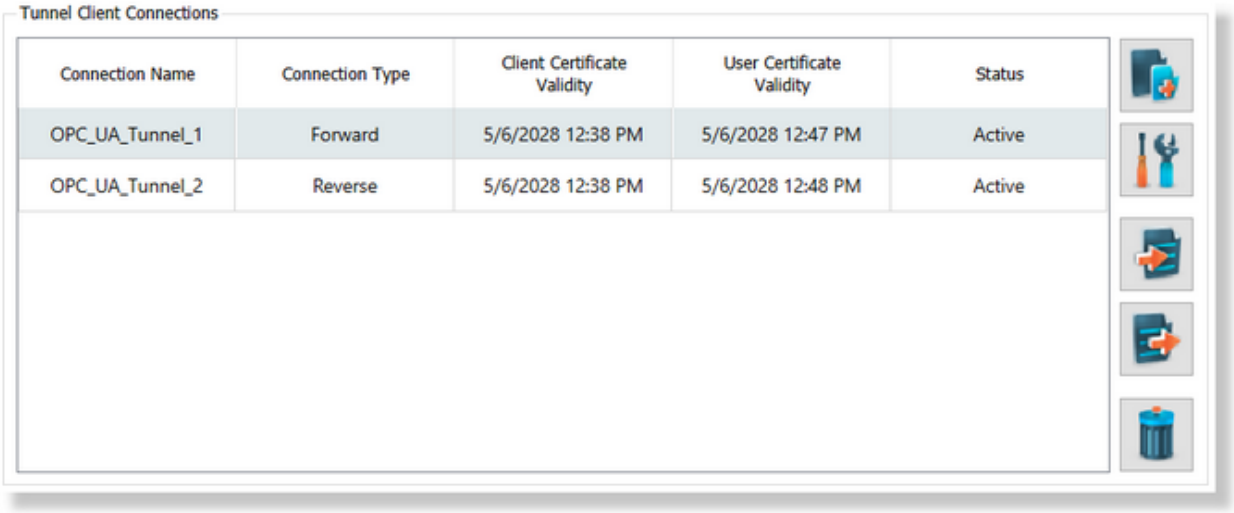
The Connection Status can be individually changed for each defined connection.

Manipulation of the connections is possible being done directly from the UA Tunnel Connections table by using of specific buttons or context menu. Based on actual selection, a set of actions become available:

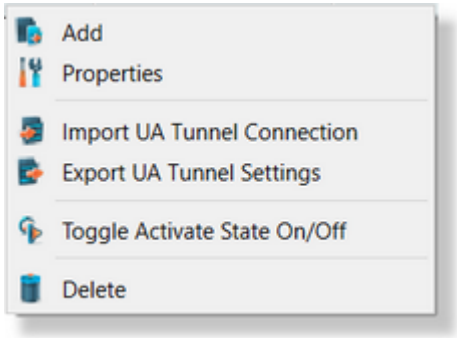
- Adding a new connection: operation executed by pressing the  button or context menu option.
- Modifying an existing connection: operation executed by pressing the  button or context menu option.
- Importing the settings for a defined connection (as data source): operation executed by pressing the  button or context menu option.

- Exporting the UA Tunnel Endpoint Settings (as data destination): operation executed by pressing the button or context menu option.
- Deleting an existing connection: operation executed by pressing the button or context menu option.

In addition, an existing connection can be activated/deactivated by using the context menu option.

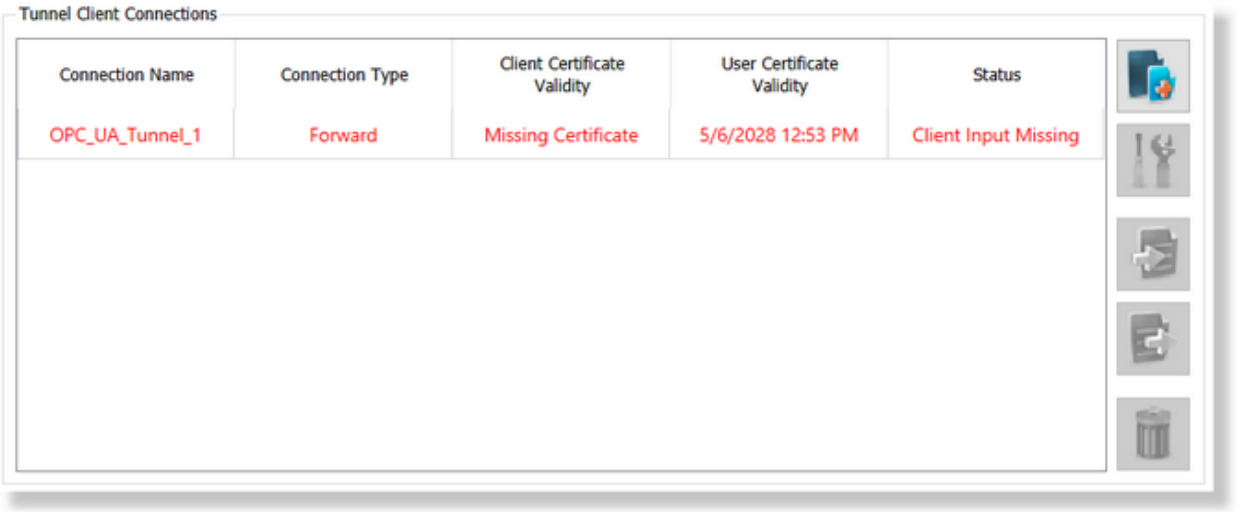


The context menu can be activated by selecting one or multiple connections and right clicking them.



Attention: Application Certificate (due an incomplete defined connection) missing is signaled directly on the connection table. This issue should be solved, otherwise the connection will be marked as invalid and ignored at run-time.

© Softing Industrial Automation GmbH



Add/Modify UA Tunnel Connection

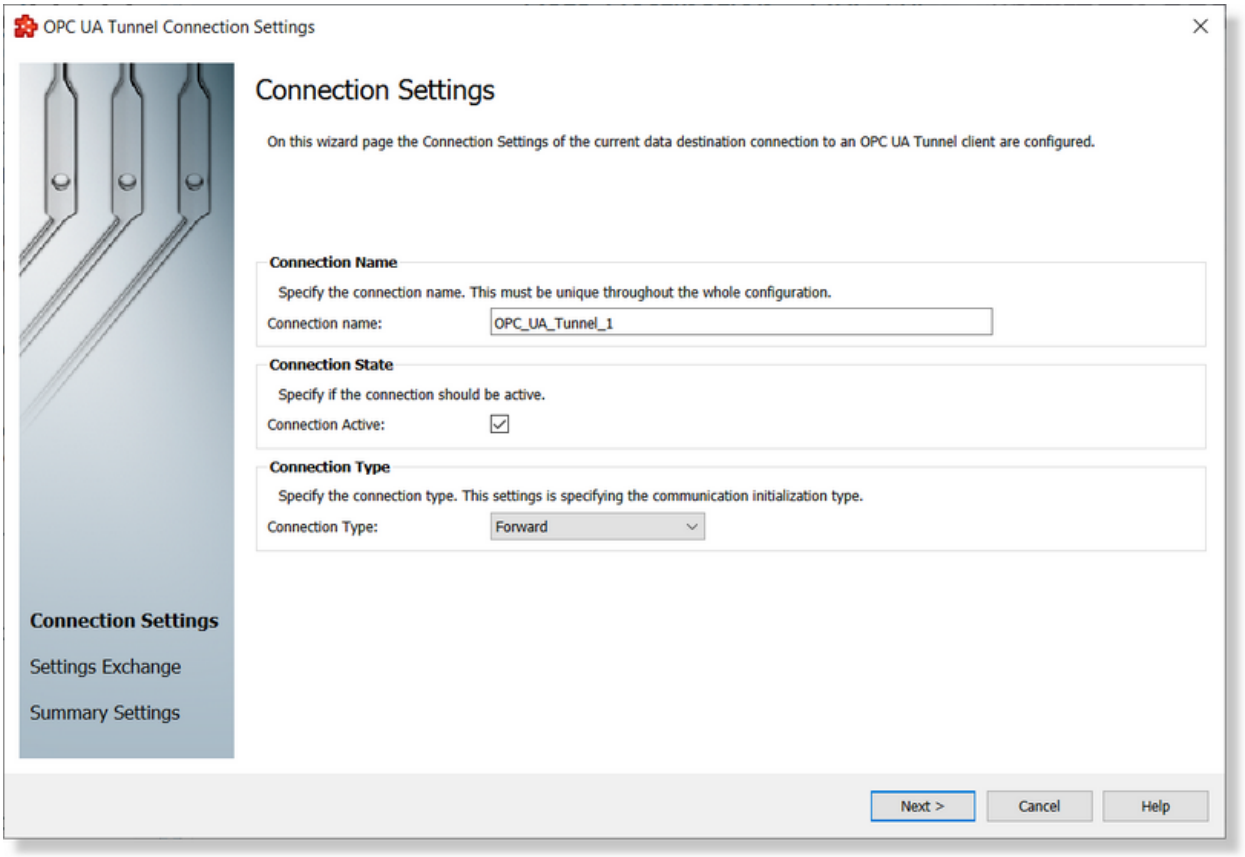
Adding/Modifying an OPC UA Tunnel connection is handled by pressing the / button or context menu option on the Tunnel Client Connections table. This operation will open the definition of UA Tunnel Client connection wizard.

On the OPC UA Tunnel connection wizard, the following operations should be executed:

- Definition of **Connection Settings**: Connection Name, Connection State and Connection Type.
- **OPC UA Tunnel Settings Exchange**: The trust relation between OPC UA Tunnel server and OPC UA Tunnel client is established and the user authentication certificate is generated. This operation is completed by exchanging a set of files between OPC UA Tunnel server and OPC UA Tunnel client. This operation is mandatory, otherwise the connection definition is not complete and the connection is ignored by the dataFEED OPC Suite Runtime.
- **Connection Settings Summary**: A summary of the connection settings defined on previous pages are available.

This operation allows to add a new connection definition or to modify an existing one.

Connection Settings Wizard page:



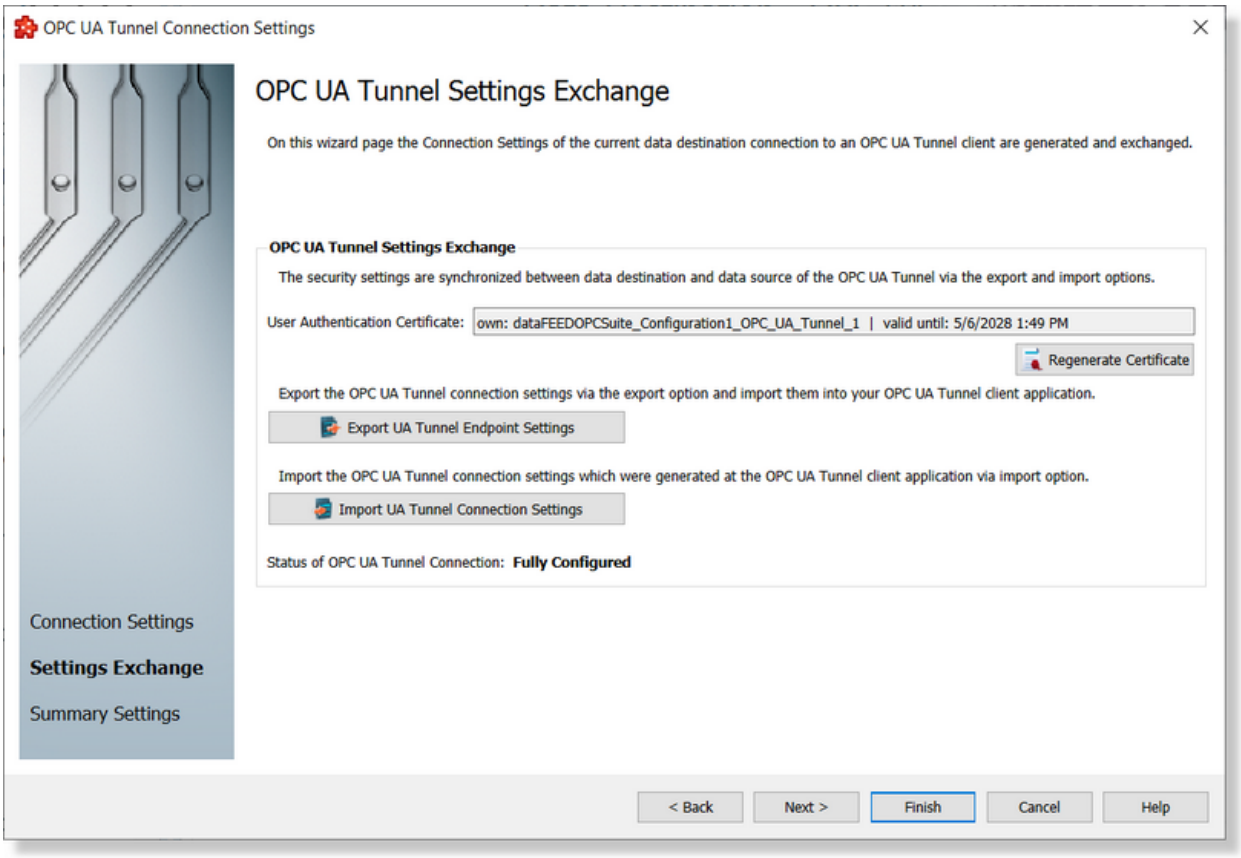
On the Connection Settings page the following identities are configurable:

Attribute	Description
Connection Name	The name of the defined connection. This is used to identify the connection - due this reason it should be unique. It is auto-completed with a predefined value, but can be also manually adjusted. The connection name is used also to identify the connection on the logging mechanism.
Connection State	The state of the defined connection. It is specifying the connection state: **active** or **not active.** Only the active connections are used by the dataFEED OPC Suite - all the not active connections being ignored at the initialization time.
	Attention: A **not active** connection is completely ignored by the dataFEED OPC Suite Runtime. When no active connections are defined the OPC UA Tunnel Endpoint is not activated and open - no connections being possible to be establish.

© Softing Industrial Automation GmbH

Connection Type	The type of the communication initialization. The following options are available: **Forward** and **Reverse**. The Reverse Connect feature, allows a server to initiate the connection to a client. The server tries cyclically to open a physical connections (sockets) to the client and send the "Reverse Hello message". A client (which supports Reverse Connect), is listening for such a socket and "Reverse Hello" messages will use this to create the "secured channel". The default connection type is **Forward**.

OPC UA Tunnel Settings Exchange page



On this page, the user authentication settings and the trust relation between OPC UA Tunnel Server (data destination) and OPC UA Tunnel Client (data source) is established by exporting the OPC UA Tunnel Endpoint settings (needed on the data source side) and importing the OPC UA Tunnel Connection settings (defined by the data source side).

The following options/functionalities are available:

- **Regenerate Certificate**: this option allows the user to regenerate the certificate used for user authentication. The name and the validity time of the certificate is available on the page.



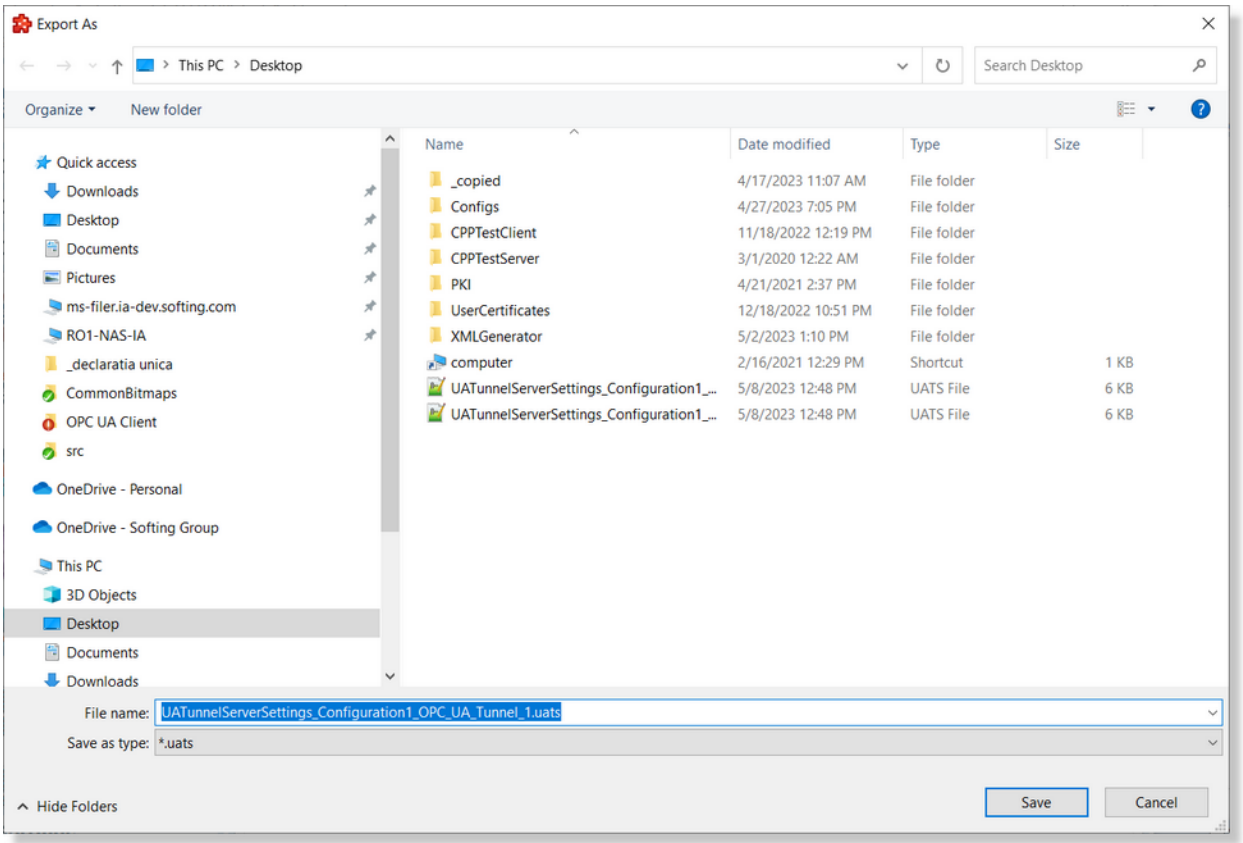
By pressing the "Regenerate Certificate" button, the user authentication certificate is re-generated.

The user authentication certificate is needed to authenticate and authorize the communication between the OPC UA Tunnel client and OPC UA Tunnel server. This user certificate is generated, managed and own by the OPC UA Tunnel server (data destination) and used by the OPC UA Tunnel client (data source) on the authentication procedure. Basically both OPC UA Tunnel server (data destination) and client (data source) should be aware about it and have access to it.

Attention: Regeneration of user authentication certificate should be followed by a synchronization between OPC UA Tunnel server and client - otherwise the conflict between instances will make impossible the authentication and authorization of communication.

- **Export UA Tunnel Endpoint Settings**: this option allows the user to export the configured OPC UA Tunnel Endpoint settings by pressing the "**Export UA Tunnel Endpoint Settings**" button. Those settings will be later imported on the OPC UA Tunnel connection definition on the OPC UA Tunnel Client (data source).

The export option is executed by selecting a destination file and exporting the settings.



© Softing Industrial Automation GmbH

Saving the exported settings will serialize the settings into an "*.uats" file which later can be imported on the connection definition of OPC UA Tunnel client configuration.

This file contains:

- Information about the OPC UA Tunnel Endpoint:
  - Server Application Name
  - Server Endpoint URL
  - Server Application URI
  - Security Connection Parameters
- Information about defined OPC UA Tunnel connection:
  - Connection Name
  - Connection State
  - Connection Type
- OPC UA Tunnel Endpoint Certificate (used for establishing the trust relation thru the OPC UA Tunnel client)
- User Authentication Certificate (used for authenticating and authorization of the commutation between OPC UA Tunnel server and client.

Note: It is mandatory to export those settings and to import them into a defined OPC UA Tunnel connection as data source - without this operation the communication cannot be establish between OPC UA Tunnel server and client.

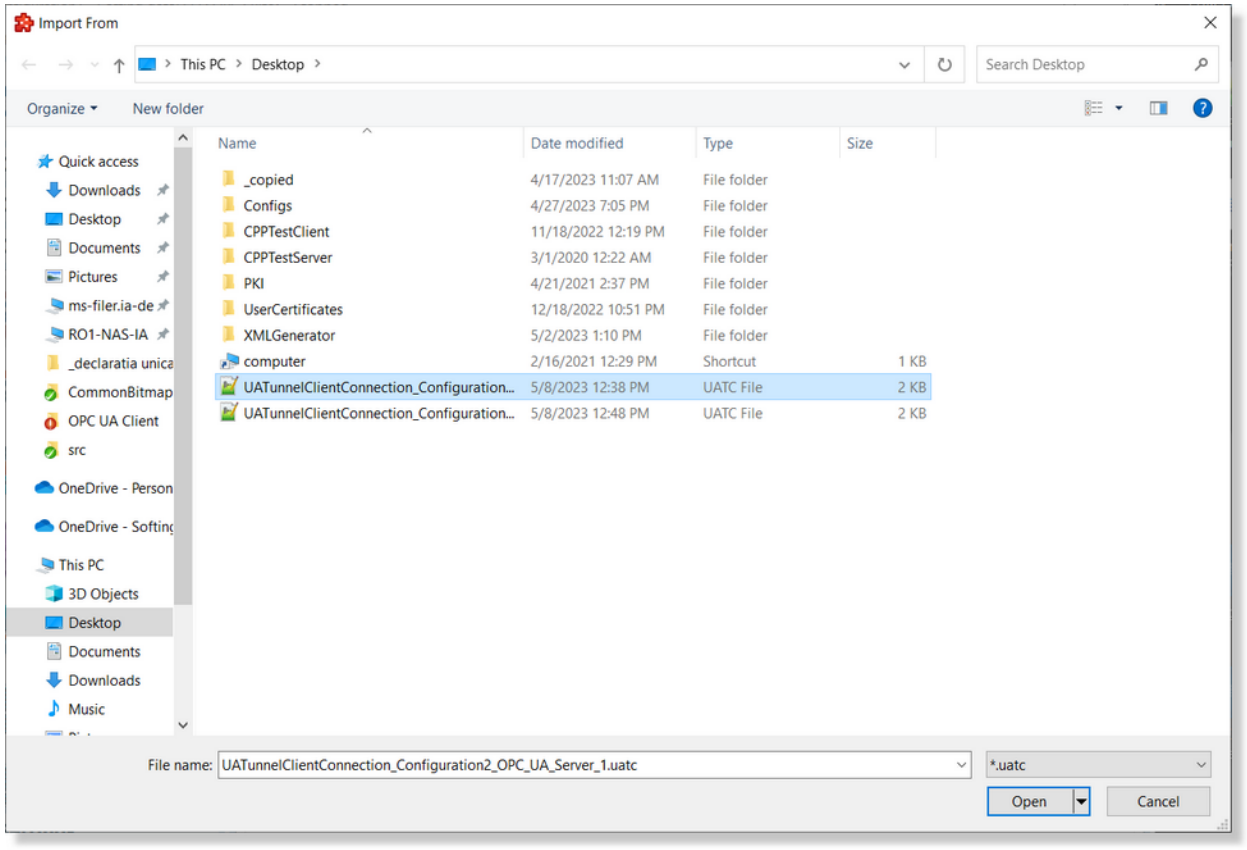
Attention: The exported files (*.uatc) cannot be modified with an external tool - in this case the import operation of the file will be not allowed (being rejected by the dataFEED OPC Suite Configurator application).

Attention: Regeneration of user authentication certificate is requiring a re-synchronization of the authentication settings between OPC UA Tunnel server and client - in this scenario, the export operation is mandatory to be re-executed.

The export operation can be executed on local driver or over the network (if access is allowed for local user).

- **Import UA Tunnel Connection Settings:** this option allows the user to import the settings defined on OPC UA Tunnel client configuration - needed to obtain the connection parameters for an OPC UA Tunnel connection (including *Reverse Connectivity* parameters if connection type is *Reverse*) and to establish the trust relation to the OPC UA Tunnel client. This operation is performed by pressing the "**Import UA Tunnel Connection Settings**" button.

The import option is executed by selecting a source file and importing the connection settings.



The imported file (*.uatc) contains the following information:

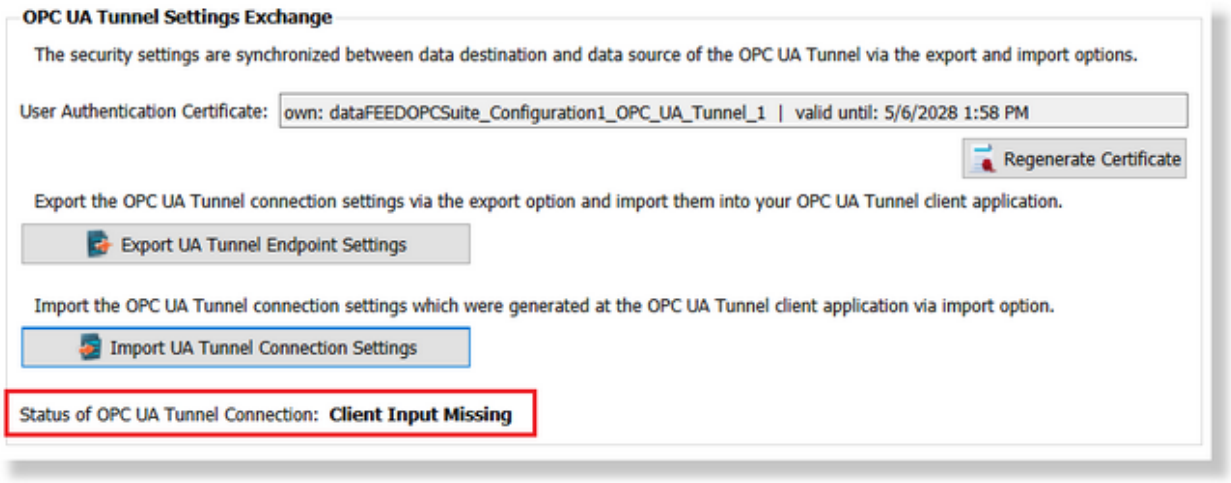
- Information about OPC UA Tunnel Client:
  - Client Application Name
  - Client Endpoint URL (only for reverse connections)
  - Security Connection Parameters
- OPC UA Tunnel Client application certificate (used for establishing the trust relation between OPC UA Tunnel Server Endpoint and OPC UA Tunnel Client)

Attention: This operation is mandatory to completely define an OPC UA Tunnel connection - without it the OPC UA Tunnel Client is not trusted for the current OPC UA Tunnel Endpoint (as effect: the communication cannot be established).

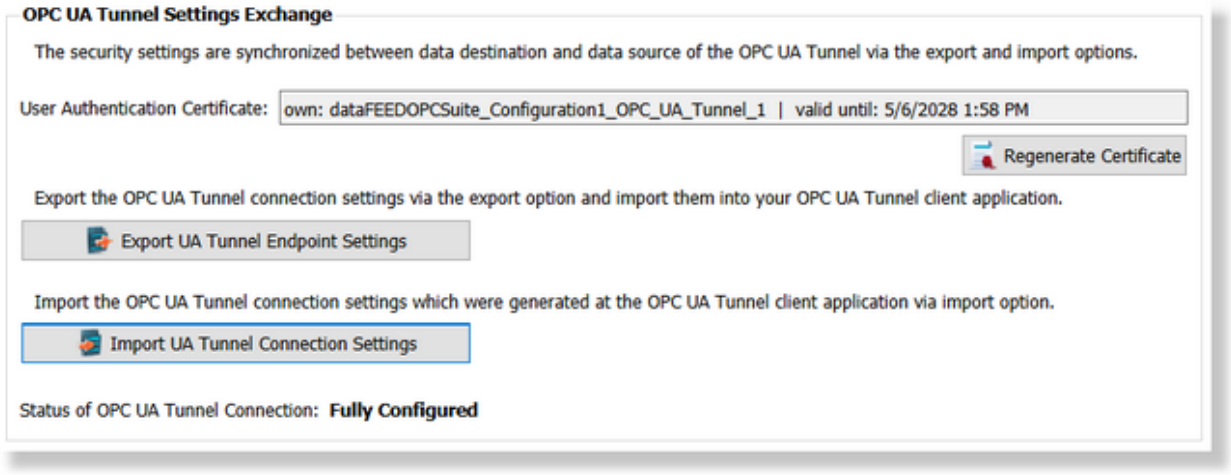
Attention: Any regeneration of application certificates on the OPC UA Tunnel client, requires a re-synchronization with the OPC UA Tunnel server endpoint - otherwise the trust relation between the OPC UA Tunnel Server and Client is not anymore valid.

A connection can be defined without importing the OPC UA Tunnel Connection settings (at this step) but it will be ignored by the dataFEED OPC Suite Runtime, being handled as an incomplete connection definition. This will be visible on the status of the connection:

© Softing Industrial Automation GmbH



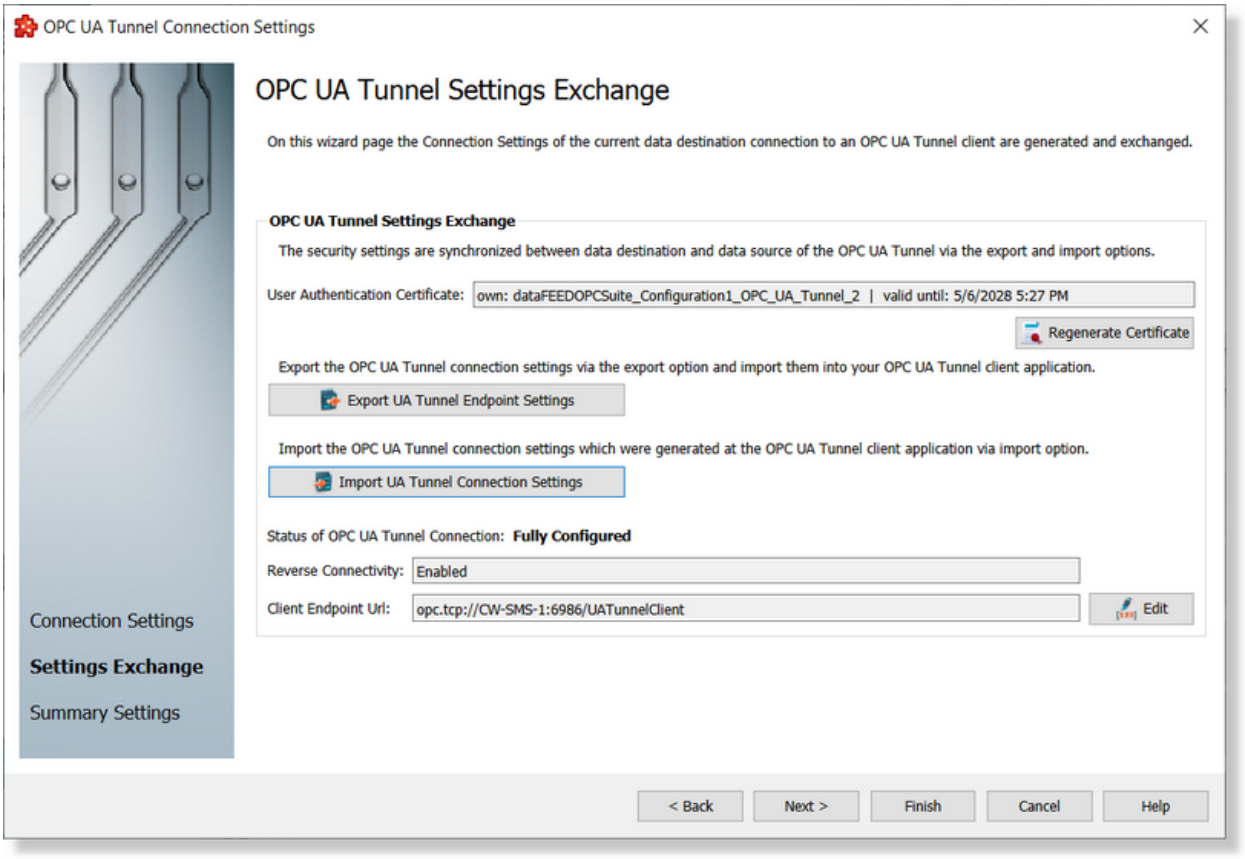
Immediately after the OPC UA Tunnel Connection settings are imported, the Status of the OPC UA Tunnel connection is changed from "**Client Input Missing**" to "**Fully Configured**".



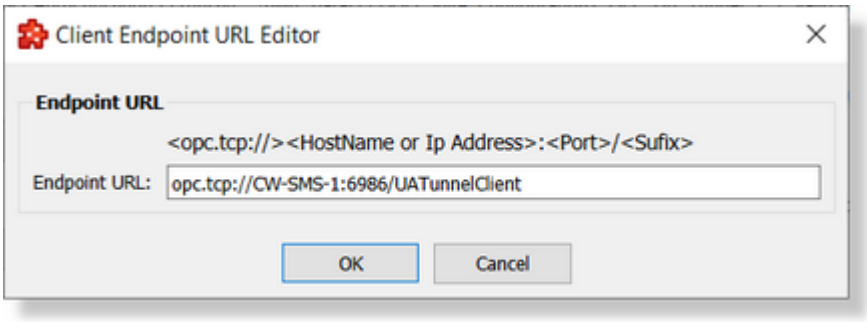
This operation is mandatory in order to establish the trust relation and the authentication and authorization mechanism.

Reverse Connection

When connection type is "**Reverse**", additional set of parameters (needed for **Reverse Connectivity**) are configurable. These settings become available after importing the OPC UA Tunnel connection is completed.



The **Client Endpoint URL** can be edited (modified) by pressing the "**Edit**" button. Then, a dialog is open allowing it to edit the Client Endpoint URL settings.

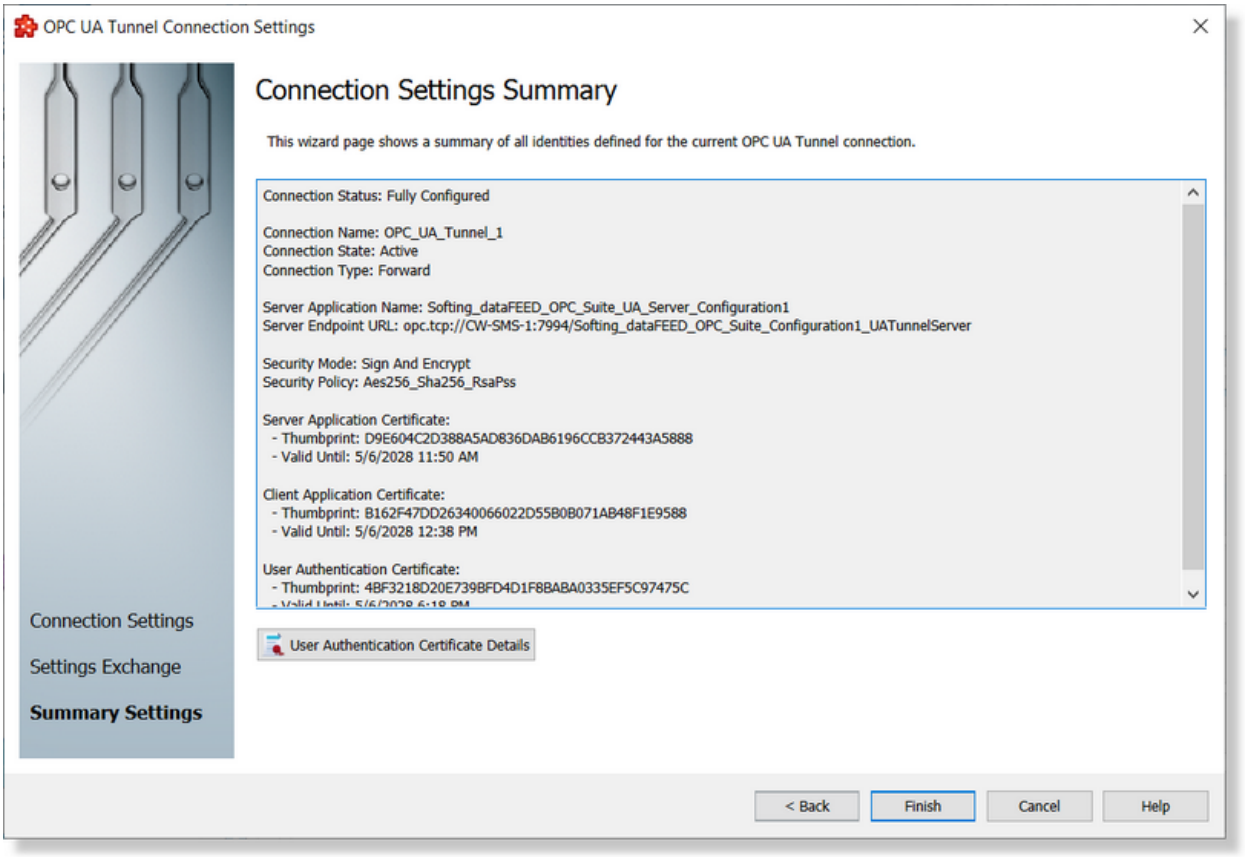


Attention: A similar Client Endpoint URL should be used on the data source side too - otherwise the server will be not able to open the unsecured channel to the OPC UA Tunnel client.

Note: This configurable setting is used for modifying the Client Endpoint URL in order to use an IP based URL instead of one based on Hostname when the Hostname cannot be solved by the DHCP server.

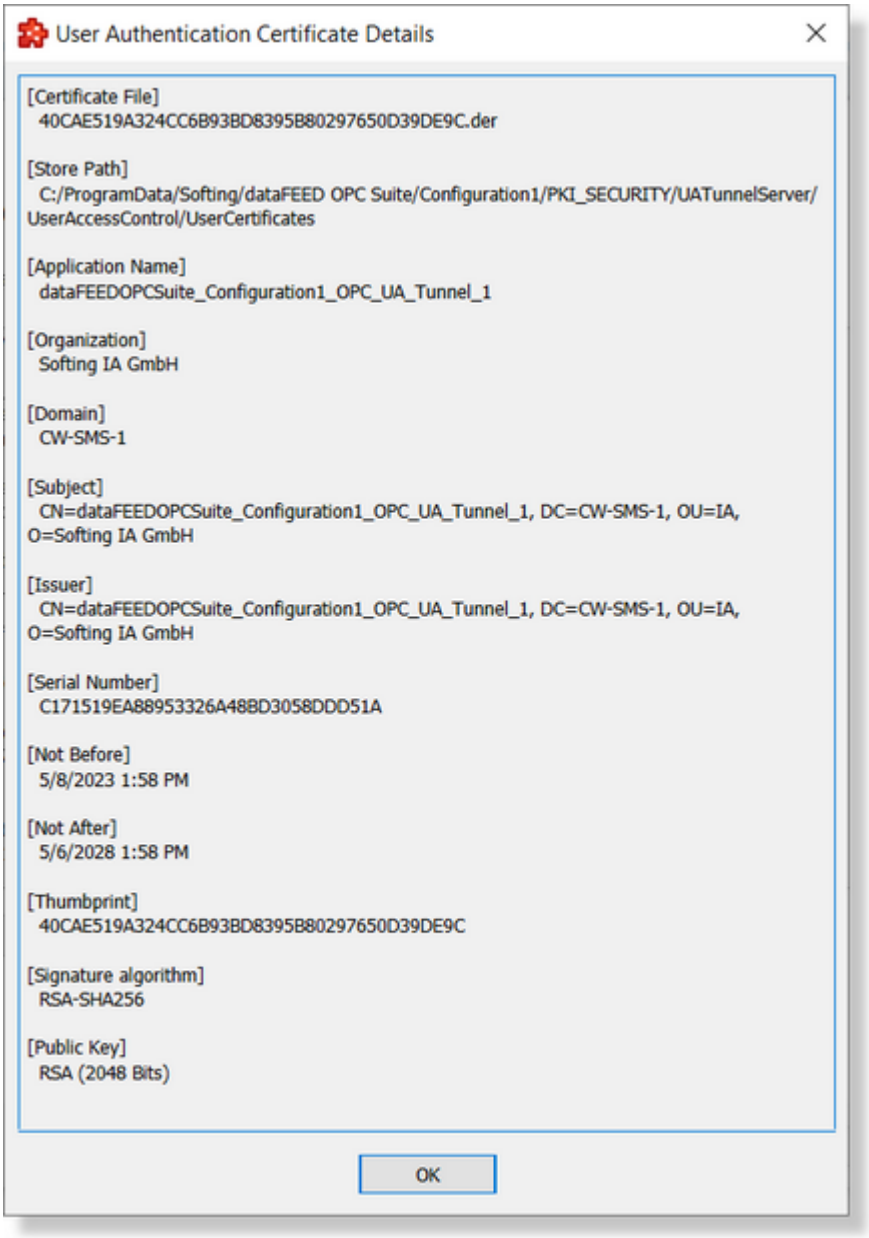
Connection Settings Summary page

© Softing Industrial Automation GmbH



On this wizard page, it is shown a summary of the configured settings (this includes the imported settings) for a specified OPC UA Tunnel connection. The detailed settings are available on the page on read only format.

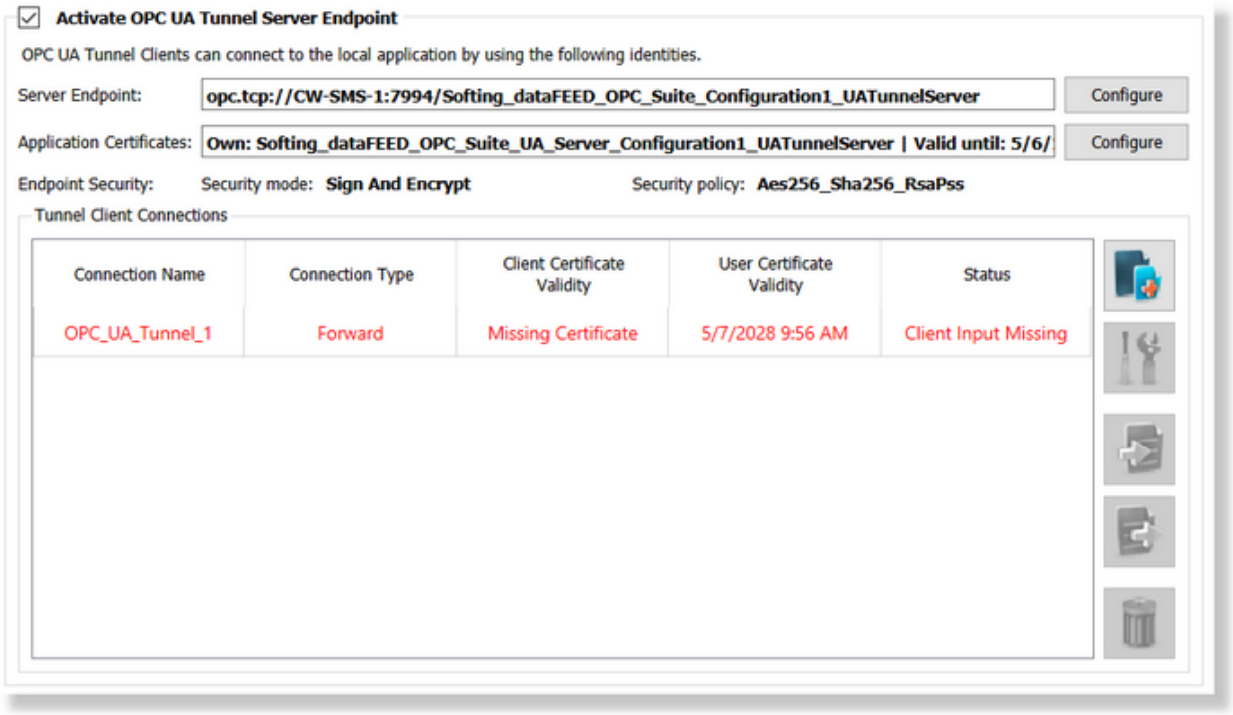
Also, it is allowed to visualize the User Authentication Certificates details by pressing the  "User Authentication Certificate Details" button. This operation opens a dialog containing all the details of actual user authentication certificate.



Pressing "**Finish**" button is finalizing the definition of an OPC UA Tunnel connection and the connection became visible on the Data Destination OPC UA Client / OPC UA Tunnel - OPC UA Tunnel Server Endpoint connections definition table. Canceling the wizard is discarding all the defined parameters of actual connection.

In case when a connection is not completely defined (the OPC UA Tunnel Connection Settings were not imported), this is visible on the Tunnel Client Connections table (as Connection Status):

© Softing Industrial Automation GmbH



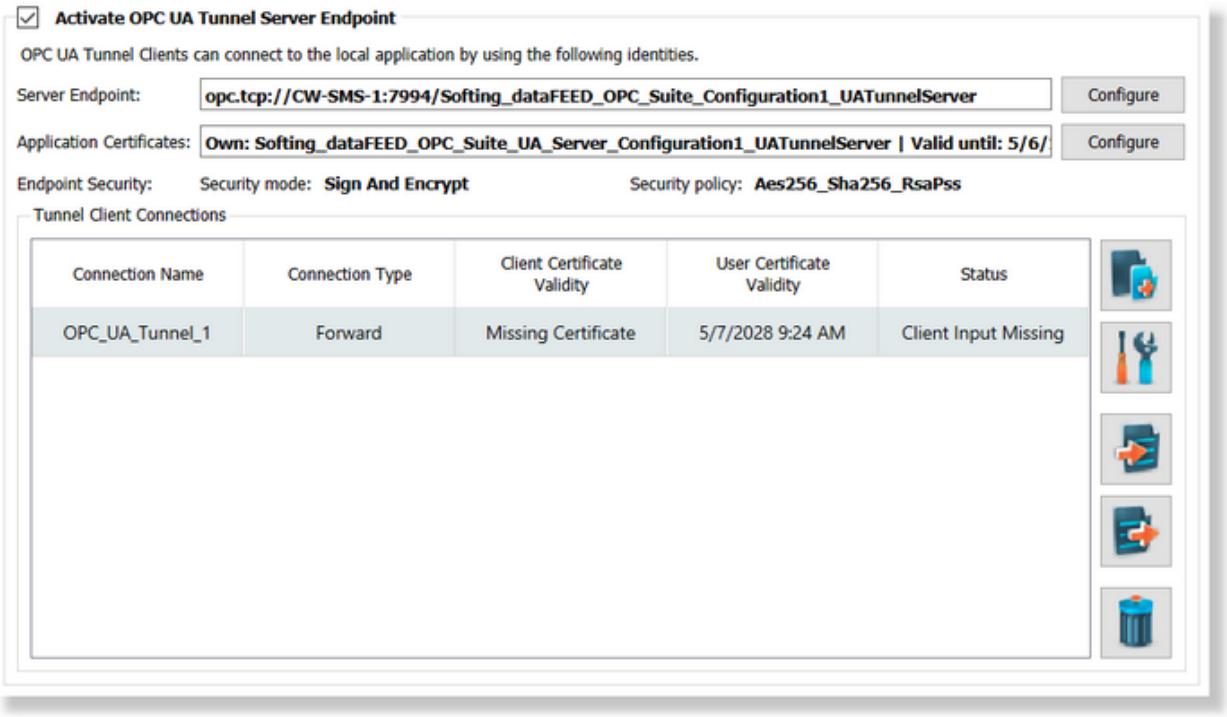
Attention: All the connections which are not completely defined (the status of connection being "**Client Input Missing**") are ignored by the dataFEED OPC Suite Runtime. Those connection are available exclusively on the dataFEED OPC Suite Configurator Tool for being defined later purposes.

Attention: The OPC UA Tunnel Server Endpoint is open only when at least one active and valid (Fully Configured) connection is defined.

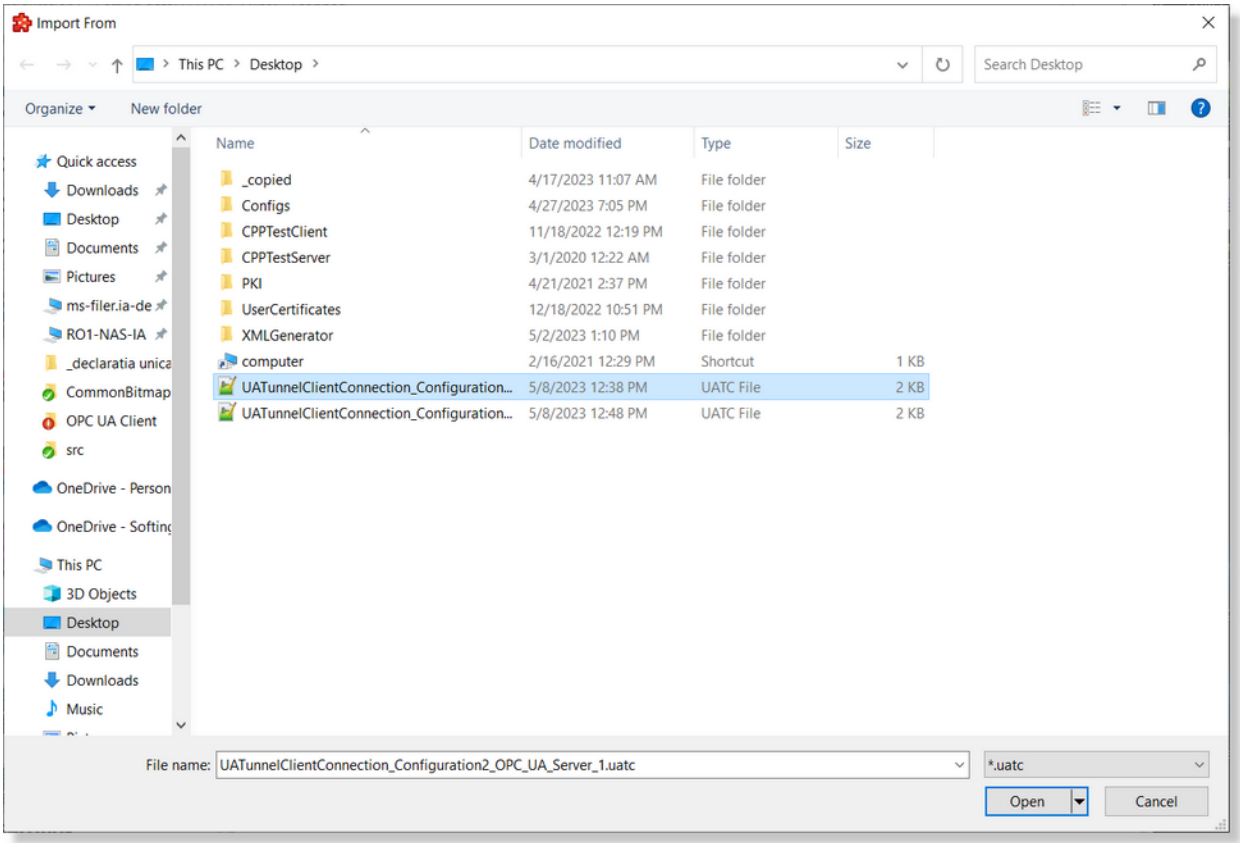
Note: The dataFEED OPC Suite Configurator allows definition of maximum 50 active OPC UA Tunnel connections. It can be defined more than 50 OPC UA Tunnel connections but maximum 50 can be activated.

Import UA Tunnel Connection Settings

Importing the settings of a selected OPC UA Tunnel connection is also possible by pressing the  button or context menu option on the Tunnel Client Connections table.

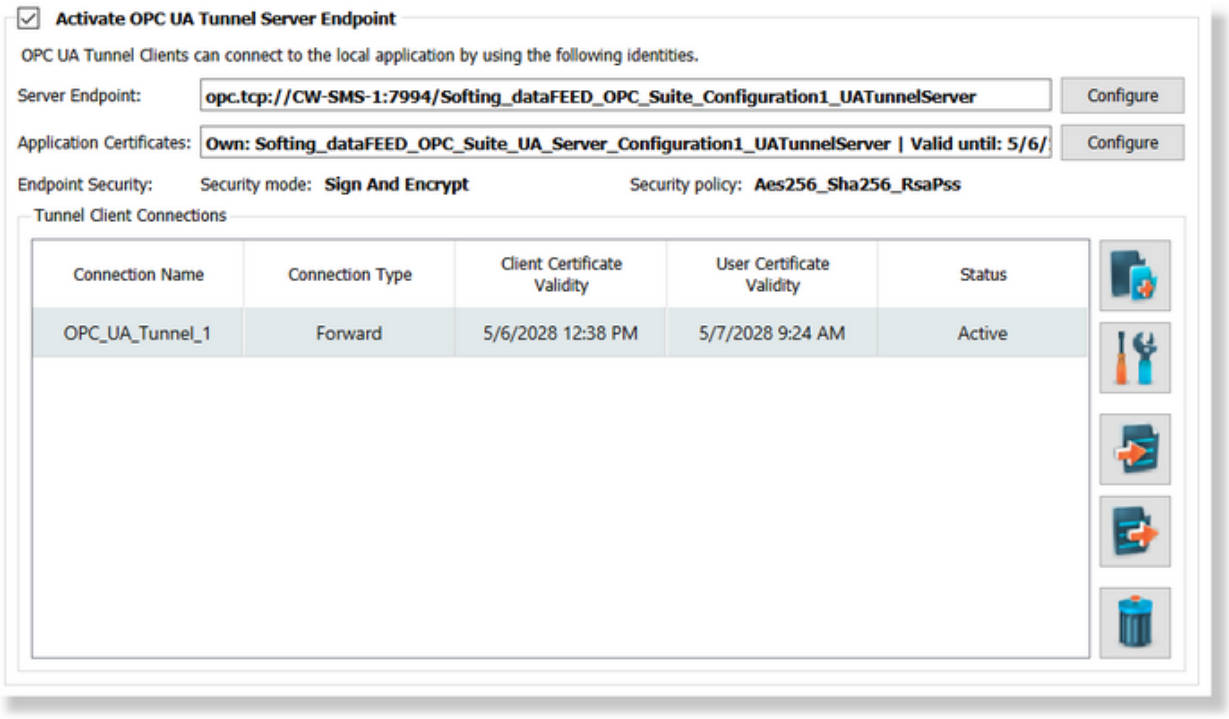


Immediately after the button is pressed, it is allowed to select the *.uatc file which will be imported:

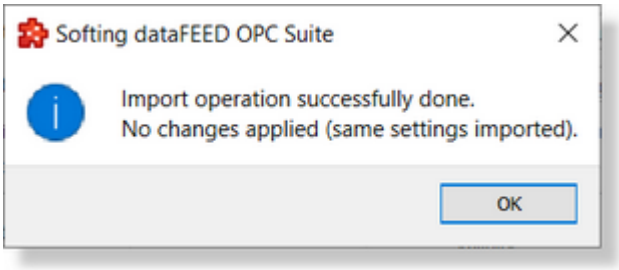


After the import operation is succeeded, the connection parameters (including the client application certificate) are overwritten with the settings from the imported file.

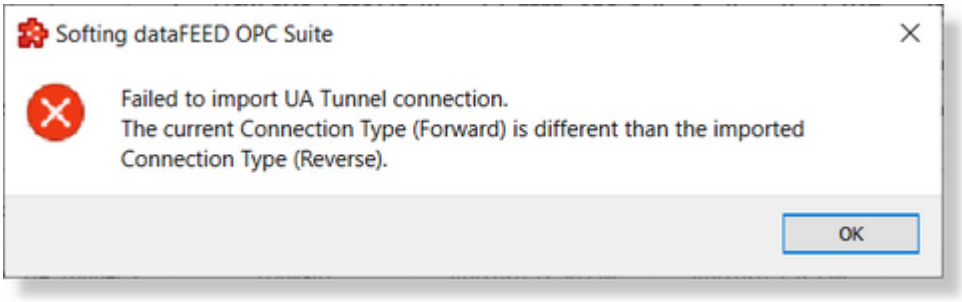
© Softing Industrial Automation GmbH



Note: Importing the same settings (re-importing the same *.uatc file) is not affecting the current OPC UA Tunnel Connection. A corresponding message is shown after the import operation.



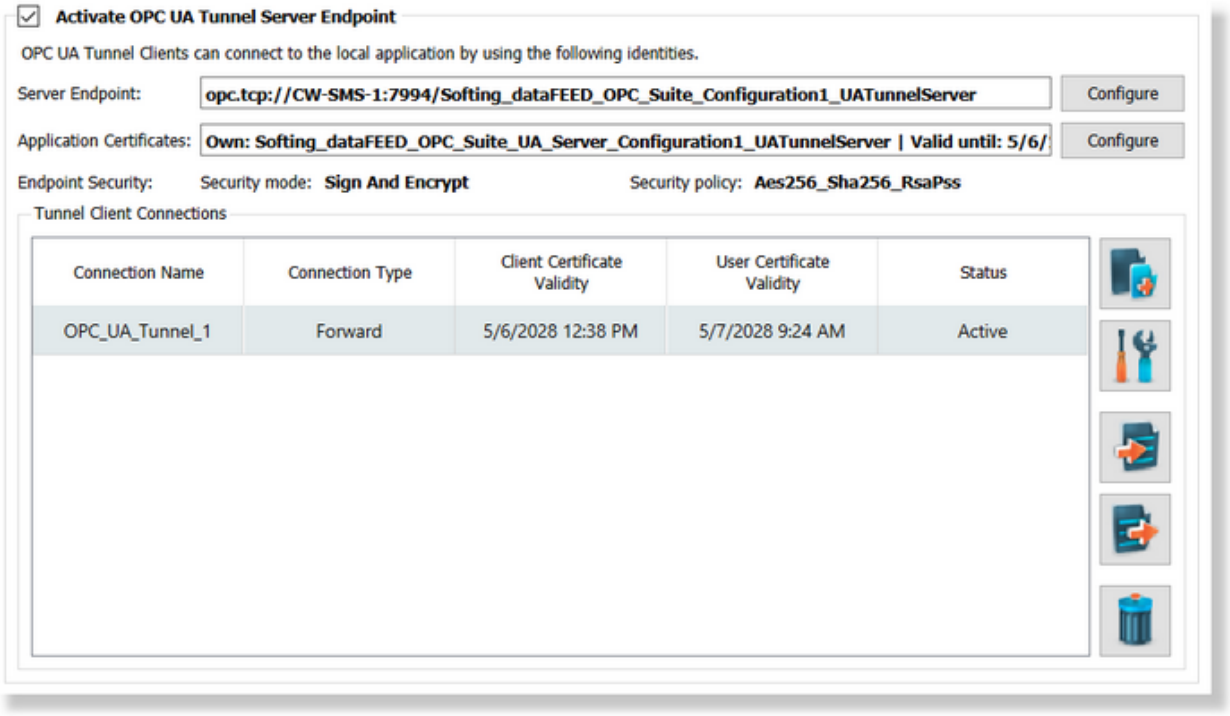
Note: Trying to import a file which was defined for connection with a different connection type than connection type of actual connection is detected by the import mechanism and the import operation is rejected with a proper error message.



Importing the OPC UA Tunnel Connection parameters (operation executed on the Connection Definition wizard or on the OPC UA Tunnel Connections table) is mandatory in order to complete a connection definition - otherwise the connection is not complete and will be ignored by the dataFEED OPC Suite Runtime.

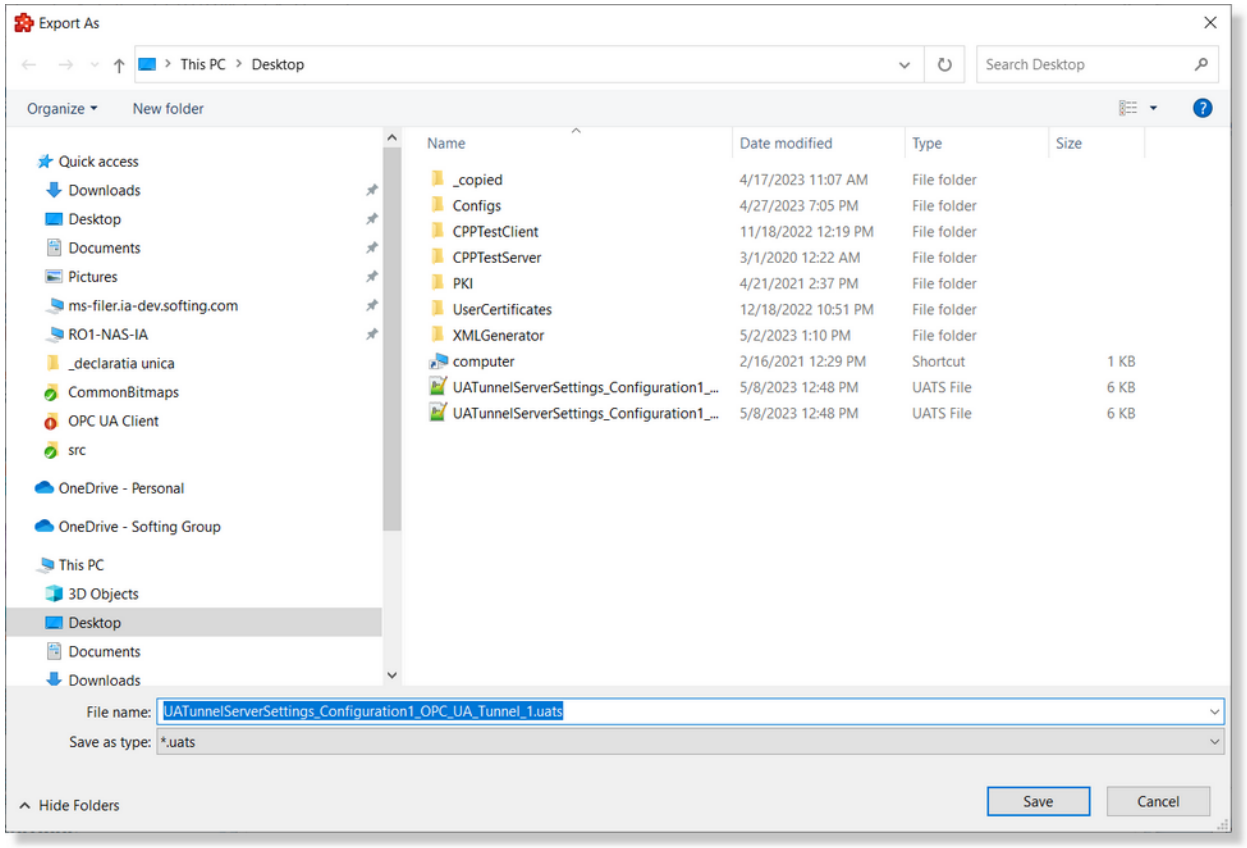
Export UA Tunnel Endpoint Server Settings

Exporting the settings of the OPC UA Tunnel Server endpoint applied for a selected OPC UA Tunnel connection is also possible by pressing the  button or context menu option from the Tunnel Client Connections table.



Immediately after the button is pressed, it is allowed to select the *.uats file which will be exported:

© Softing Industrial Automation GmbH



When the export functionality is completed, a proper message is signaling that the file was exported and can be used on the definition of connections on data source side.

Note: The export operation is mandatory in order to be possible the definition of OPC UA Tunnel connection on the OPC UA Tunnel Client (data source) side.

Note: The connection definition on both OPC UA Tunnel server and client can be done exclusively by import/export operations. This operation is also responsible for ensuring the trust relation between OPC UA Tunnel server and client.

The export functionality can be operated on the local drive and also over the network (when access of current user to the network shared location is allowed).

Remove UA Tunnel Connection

Removing a set of defined connections is possible by pressing the  button or context menu option. This operation is allowed for one or multiple OPC UA Tunnel connections. Immediately after the operation is confirmed, the selected OPC UA Tunnel connections are deleted.

Attention: It is mandatory to have defined at least one active and valid OPC UA Tunnel connection - otherwise the OPC UA Tunnel endpoint will be not open.

Licensing

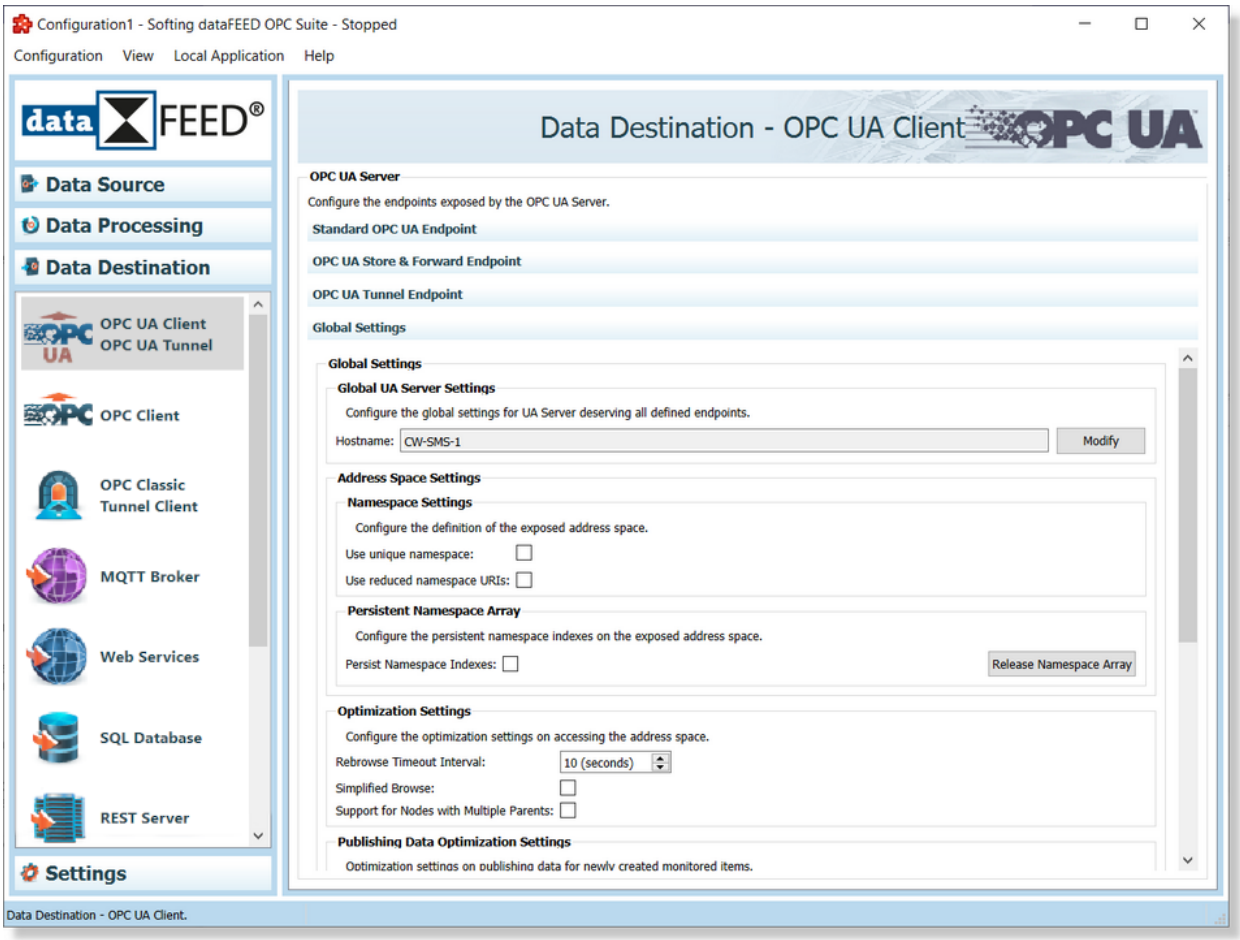
The functionality is licensed exclusively by **dataFEED OPC UA Tunnel** license. This license is included into to **dataFEED OPC Suite Extended** license.

Attention: The **dataFEED OPC Suite** license is not including the **OPC UA Tunneling** (both **OPC UA Tunnel Server** and **Client**) functionality.

Demo Mode is working on full functionality for 72 hours.

Note: Without a license (or Demo Mode) the functionality is disabled (the OPC UA Tunnel endpoint being disabled at initialization time).

The Global UA Settings are intend being directly over the exposed OPC UA Server by the dataFEED OPC Suite. Basically these settings are applied on all server endpoints without a possibility to apply different changes on different server endpoints.



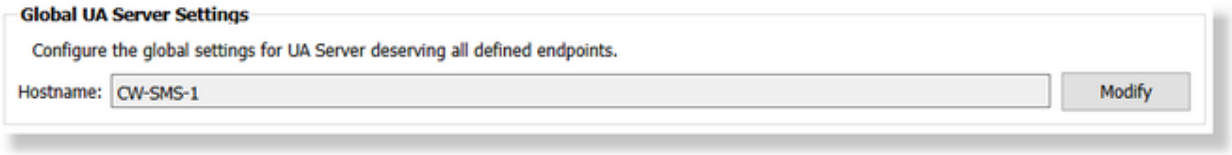
The following global settings are configurable:

- Hostname as Global UA Server Settings.

© Softing Industrial Automation GmbH

- Persistance of Namespace Array.
- Optimization Settings.
- Publishing Data Optimization Settings
- Output Data Access
- LDS (Local Discovery Settings) Registration.
- Diagnostics Informations

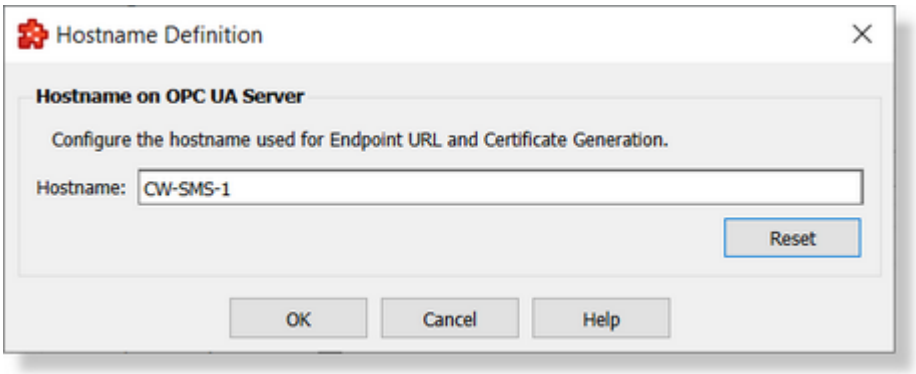
Hostname



By default the application is using the current machine hostname obtained from the Operating System.

This can be changed to a user defined hostname (it can be used the network domain as prefix or a totally different hostname).

The hostname can be changed by opening the **Hostname Definition** dialog:

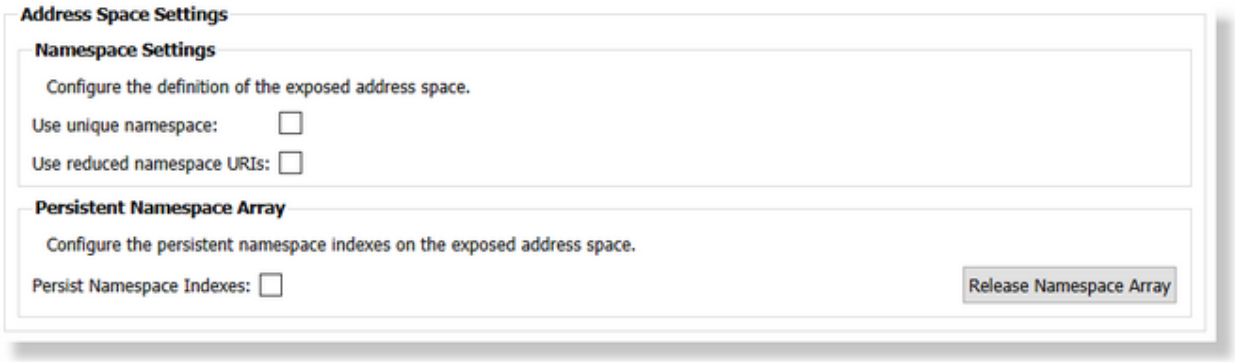


The machine hostname can be received from the operating system by pressing **Reset** button.

Attention: Changing the hostname used by the OPC UA Server is requiring a re-generation of all application certificates (operation which is done automatically). This operation is requiring (where needed) a redone of trust relation between the exposed UA Server and OPC UA Clients (this operation should be done manually by exporting/importing application certificates).

Attention: The local host name should contain exclusively ASCII characters. Usage of special/language related characters (sample: Ä,Ö, Ü, ? , ä, ö, ü, ß) are recommended being not used as host name by the operation system (Microsoft Windows 7 or later) and not allowed being used inside the application certificates by the OPC UA Specifications.

Address Space Definition Settings



The address space definition can be adjusted based on **Address Space Seetings** which allows to change a set of settings on manipulation of the address space. Those settings allows to:

- Define the way how the namespace is defined.
- Persist the namespace indexes over different configuration changes.

Namespace Settings



On the main UA Client data destination view, it is allow to specify the address space definition settings:

Use unique namespace:

By default each defined data source connection is added into the address space using its own namespace, enabling this functionality will result in aggregating all the data source connections into only one namespace inside the exposed address space.

Basically the first namespace (namespace index 0) is reserved for OPC Foundation stack items, the second namespace (namespace index 1) is used by the OPC Toolkit internal items and the other ones can be used for by the application. With this function enabled, all the connections will be part of the third namespace, using namespace index 2) instead of being defined an own namespace for each connection.

The creation of the address space is done at the initialization time of the configuration.

© Softing Industrial Automation GmbH

Attention: Operating on the defined data-sources by adding or removing them, can change the allocated namespace index for a defined connection. It can be the situation that, after a configuration changed, followed by a restart of the configuration, the corresponding index for a defined connection being changed. This can be avoided by enabling the "Persistent Namespace Array" functionality.

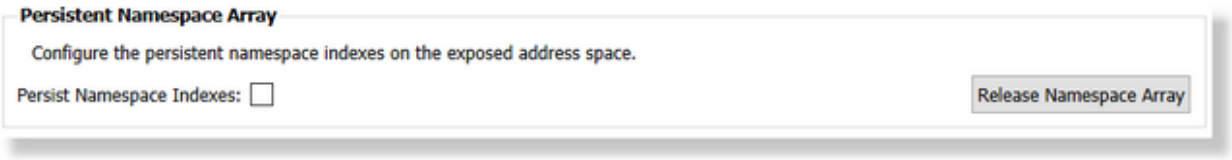
Use reduced namespace URIs:

The pattern used on defining a namespace URI is: "urn:dataFEEDOPCSuite.<Connection Name>", where <Connection Name> is the data source connection name.

Enabling the usage of reduced namespace URIs will change the pattern on defining the namespace URIs to use only <Connection Name>.

Attention: usage of reduced namespace URIs is not OPC compliant accordingly with the OPC UA Specs 3.0.

Persistent Namespace Array



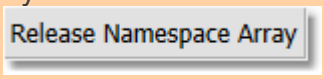
The allocation of namespace indexes for each defined data source connection is done by the application at the start-up (initialization) time - basically for each defined connection is it allocated an individual namespace index (the allocation of indexes is using consecutive values and no gaps are allowed by the OPC Specs). The sequence/order of namespace indexes allocation for each defined connection being done based on the type of data source.

Operating on the defined data source connections by adding or removing data sources has as impact a change on the distribution of indexes on defined data source connection. Even if the data source family is not changed (the newly introduce connections being added for other data source family) - the allocation of the indexes on current data source family can be changed.

Enabling the functionality, each time when the application is started it is allocated the same namespace index for a defined connection - with this functionality the distribution of namespaces over the defined data source connections being preserved.

Attention: Each time when a new data source connection is identified at application startup time, it is allocated the next available namespace index (for older data source connections being allocated the previously used namespace index).

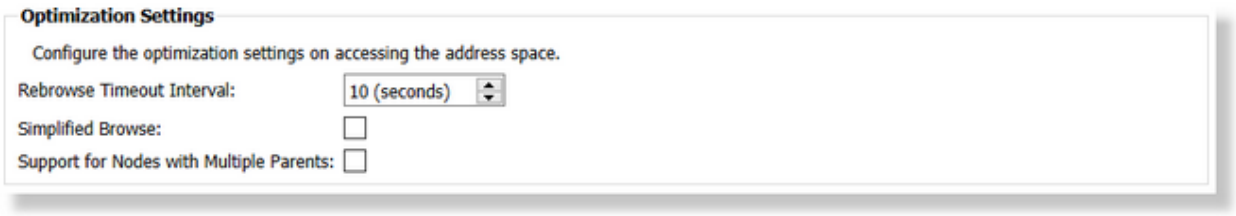
Attention: Removing a data source connection will not delete the previously used namespace - basically the historical data source connections are visible at the namespace array level. The

only way to release an allocated namespace being by accessing the "Release Namespace Array" 

functionality.

Note: The persistent namespace array option make sens only while the usage of unique namespace is disabled - otherwise the namespace array is persisted by default (always the second namespace being used for all defined connections).

Optimization Settings



This settings is configuring the optimization settings on accessing the OPC Address space specifying the interval for a requiring a device browse.

Rebrowse Timeout Interval:

Basically depending on this setting on a browse service call the response is provided directly from the internal address space of OPC UA Toookit or the service call is forwarded to corresponding data source.

When a browse is performed on the data source the timer is reinitialized - in case on a new browse depending on time elapsed between operation (bigger or lower then the specified value) the call is forwarded or not to the data source.

The value is specified in seconds and the default value is 10 seconds.

Simplified Browse:

By default, a browse operation is adding the provided results to the internal address space of the exposed endpoint (UA Server). Basically, the internal address space is synchronized with the results of any browse operations - the internal address space reflecting the results of any browse operations. Due this reason, a browse operation, basically is performing additional operations in order to have enough information to add the result items to the address space:

- Reading a set of properties for each received node/variable as browse result: additional properties (access rights, data type and node type) are mandatory properties needed to add a node/variable to the internal address space.
- Identification of leafs inside the address space: this is needed in order to determine that a node has forward hierarchical references or not (contains children or not). This information is obtained by browsing the node/variable filtered for a single hierarchical reference.

© Softing Industrial Automation GmbH

This additional operations executed together with a browse operation, are time consuming operations and also could put some pressure at the data source side (being executed on chunks based on operational limits settings).

Also, this behavior introduce a set of benefits: the items being already part of the internal address space, the next operations executed on them: read, write, subscribe and others does not required an additional validation on first execution and some properties (item identified, data type, access rights, node type and others) could be provided directly from the internal address space without interfering with the data source - due this, it is much faster and reduce the traffic to the data source.

Enabling the **Simplified Browse**, a browse operation is not synchronizing the results with the address space. Basically, a pure browse operation is executed thru the data source and the results are provided back to the OPC Client which required it. The results are not processed. This behavior, reduced the amount of additional operations executed on the data source to zero, reducing the time needed to perform a browse operation. Also, the data source is not stressed by the additional operations on each received browse result.

Still, enabling this functionality will reduce the performance on next IO/Subscribe operations - the initial operation requiring in this context an additional node/variable validation at the data source level.

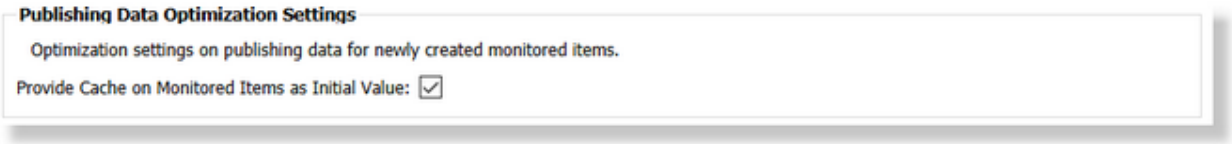
This option is recommended to data sources with limited resources or capabilities (Nano/Embedded UA/DA Servers) or data sources with flat address spaces.

By default this option is disabled. Enabling it, the **Rebrowse Timeout Interval** settings has no meaning.

Support for Nodes with Multiple Parents:

This options allows to create multiple hierarchical references to a variable when this defined into the data source. Disabling it, will create only the first hierarchical reference to variable which was obtained after a browse operation.

Publishing Data Optimization Settings



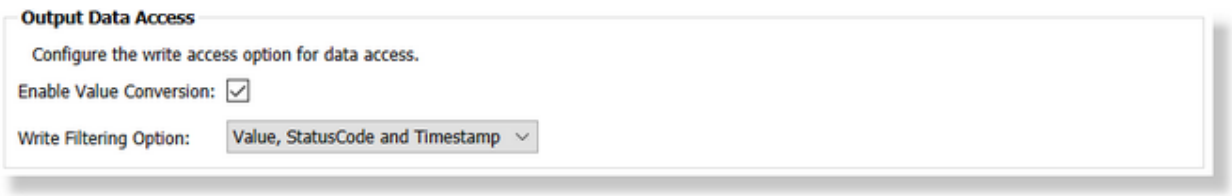
This settings specifies the source that the server uses for providing the initial values on a newly created subscriptions (including its corresponding monitored items). Basically, when an OPC UA Client creates monitored items or changes their monitoring mode from "Disabled" to "Sampling", the server application needs to provide the initial values for the monitored attributed of each created nodes. Depending on this setting, the application may decided to either use the cache attribute value (if exists) or to wait until the corresponding data source from the underlying system sent a proper data update notification.

Enabling the "Provide Cache on Monitored Items as Initial Value" will force the application to send the cache value (if exists) on all newly created o activated monitored items.

Disabling the functionality, will adjust the application to wait until the data source will subscribe the monitored item and deliver an initial datachange in order to prepare the initial publish on all monitored items.

Attention: Disabling this option could generate the situation that the initial datachange provided by the application is not sent immediately - changing the application behavior in a non-compliant way (based on OPC Specification).

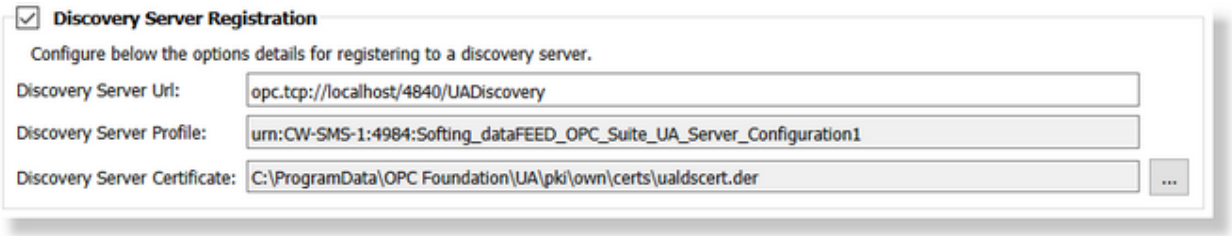
Output Data Access



This settings is configuring the output access to data - basically specifying the filters/validation of data on write operations.

- **Enable Value Conversion** - This setting is enabling value conversions on write requests for OPC UA Server. When this setting is enabled a value conversion is tried on write requests with values having data-type different the the native data-type of corresponding variable. The conversion is tried only for convertible data types.
- **Write Filtering Option** - This setting is enabling the filtering on attributes required to be written by a write request on OPC UA Server. Basically on a write request, depending on the client implementation the following attributes can be required to be written: Value, Quality (status code) and Source/Server Timestamp. With this setting there is implemented the possibility to ignore or to set to default some of these properties.The following values are allowed:
  - VQT - value, quality and timestamp.
  - VQ - value and quality - in case when timestamp is part of the request this is ignored.
  - VT - value and timestamp
  - V - only value.

Discovery Server Registration



© Softing Industrial Automation GmbH

dataFEED OPC Suite registers at the Local Discovery Server by using the default settings (see picture above) and configured security and authentication settings for each enabled server endpoint.

Furthermore, you can register dataFEED OPC Suite with its endpoint URL at an already existing Local Discovery Server. dataFEED OPC Suite is then identified with the endpoint URL that has been specified by the user beforehand.

Note: The trust relation between the dataFEED OPC Suite UA Server and Local Discovery Server is automatically done by the application configuration tool.

Server Diagnostics Information



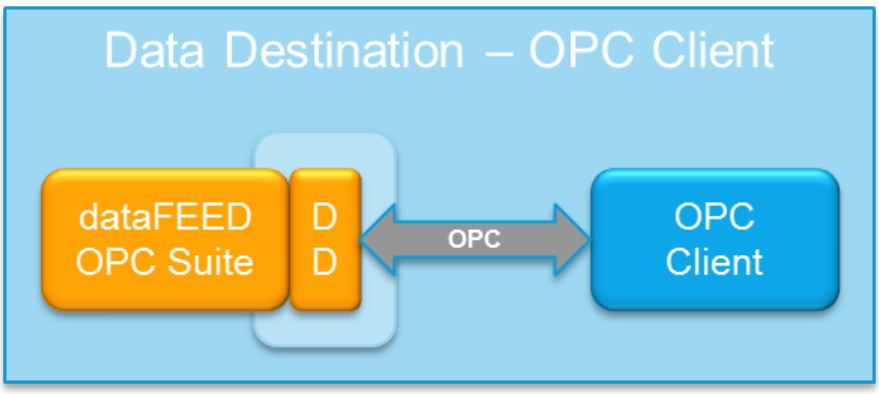
dataFEED OPC Suite UA Server allows collecting diagnostics data based on created sessions and subscriptions (by default being deactivated). This information can be used by an external connected OPC UA Client to analyze the load on the OPC UA Server:

- number of active connected sessions.
- number of active subscriptions and monitored items.

Disabling the functionality is not allowing for a connecting client to activate and read the diagnostics information.

7.3.3.2 OPC Client

This section of the Softing dataFEED OPC Suite Online Help describes how to configure the OPC client data destination functionality of Softing dataFEED OPC Suite.

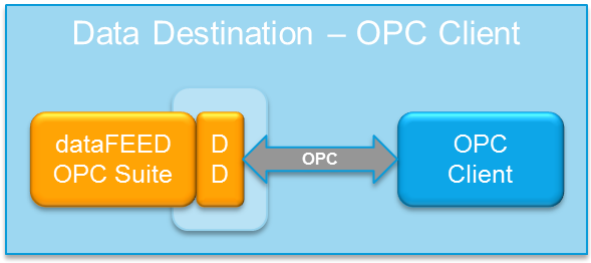


Enabling the OPC client data destination functionality allows external OPC clients to access the data and alarm items of the local dataFEED OPC Suite application via one of the supported

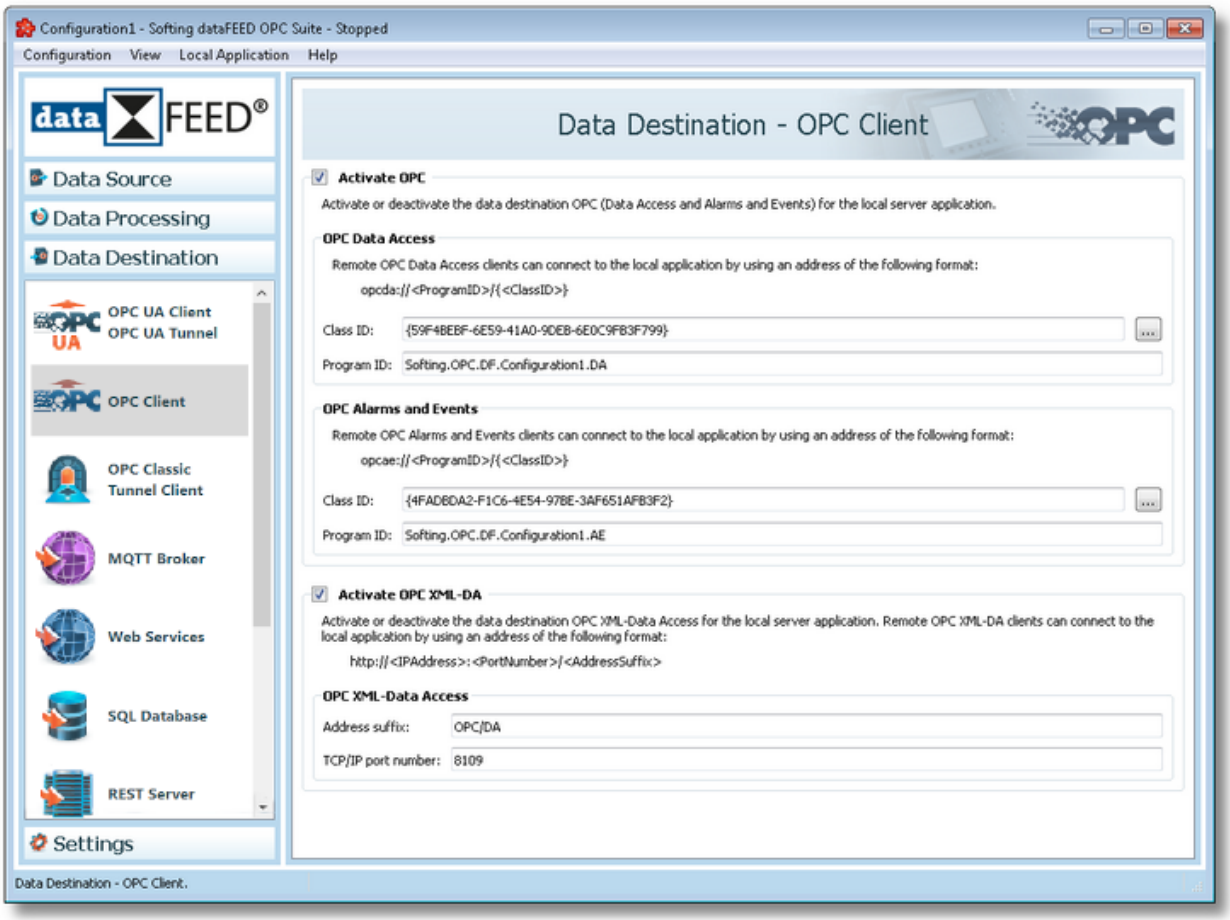
OPC protocols. It is possible to separately activate, deactivate and configure the *OPC Data Access*, the *OPC Alarms and Events* and the *OPC XML Data Access* protocols.

A more detailed description of how to configure the OPC client data destination functionality can be found in the Configuration: Data Destination - OPC Client section.

7.3.3.2.1 Configuration



Enabling the data destination client functionality allows external OPC clients to access the data and alarm items of the local dataFEED OPC Suite application via one of the supported OPC protocols. It is possible to separately activate, deactivate and configure the *OPC Data Access*, the *OPC Alarms and Events* and the *OPC XML Data Access* protocols. This configuration page contains all the OPC identity settings for the three different OPC protocols.



© Softing Industrial Automation GmbH

Attention: All the identity settings (Class ID, Program ID and TCP/IP port number) of the current configuration must be unique within all configured local dataFEED OPC Suite applications on the computer.

Attention: If the current configuration has a Class ID or a Program ID which is already being used by another configuration (the other configuration being registered in the system with the respective IDs) then when the current configuration gets saved (the moment the system registration is performed) these IDs will no longer identify the previous configuration but the current one.

Attention: Two instances of the local dataFEED OPC Suite application for different configurations must not run on the same TCP/IP port.

Note: It is not recommended to manually modify the Class IDs or Program IDs of the current local dataFEED OPC Suite application configuration.

Note: This section is disabled when the corresponding local dataFEED OPC Suite application is running.

OPC Data Access and OPC Alarms and Events

By checking the *Activate OPC* option, the OPC client access to the local dataFEED OPC Suite application is enabled. This includes both OPC Data Access and OPC Alarms and Events clients.

A new, unique Class ID can be created by pressing the *Generate new Class ID* button . Remote OPC Data Access or OPC Alarms and Events clients can connect to the local application by using an address with the following format:

```
<ConnectionType>://<HostName>/<ProgramID>/{ClassID}
```

or

```
<ConnectionType>://<IPAddress>/<ProgramID>/{ClassID}
```

The <ConnectionType> placeholder is a pre-defined string, the allowed values are shown in the table below:

<ConnectionType>	**Description**
opcda	Connection from an external OPC Data Access client.
opcae	Connection from an external OPC Alarms & Events client.

The *<ProgramID>* and *<ClassID>* placeholders are the OPC / DCOM identities of the corresponding internal OPC server of the local dataFEED OPC Suite application. The *<HostName>* and *<IPAddress>* placeholders specify the host name and the IP address of the PC on which the local dataFEED OPC Suite application is running.

Note: The OPC Data Access and OPC Alarms and Events functionalities are enabled by default. You should only disable them if you don't want to grant OPC clients access to the namespace of the local dataFEED OPC Suite application.

Attention: For correct OPC DA/AE functionality (DCOM based protocols) on PCs running Windows 10, please ensure that the 1803 OS version is used (build 17134.48).

OPC XML Data Access

By checking the Activate OPC XML-DA option, the OPC XML Data Access client access to the local dataFEED OPC Suite application is enabled. A corresponding unique TCP/IP port number needs to be provided for OPC XML-DA communication. The Address suffix setting is configurable but cannot be empty.

After saving the configuration and starting the local dataFEED OPC Suite application, remote OPC XML-DA clients can connect to the running local dataFEED OPC Suite application by using an address of the following format:

```
http://<HostName>:<PortNumber>/<AddressSuffix>
```

or

```
http://<IPAddress>:<PortNumber>/<AddressSuffix>
```

The <HostName> and <IPAddress> placeholders specify the host name and the IP address of the PC on which the local dataFEED OPC Suite application is running. The <AddressSuffix> placeholder is mandatory.

Note: The TCP/IP port number for OPC XML-DA client access is the same as the TCP/IP port number for the integrated web server, which can be configured on the Configuration: Settings - Web Server page.

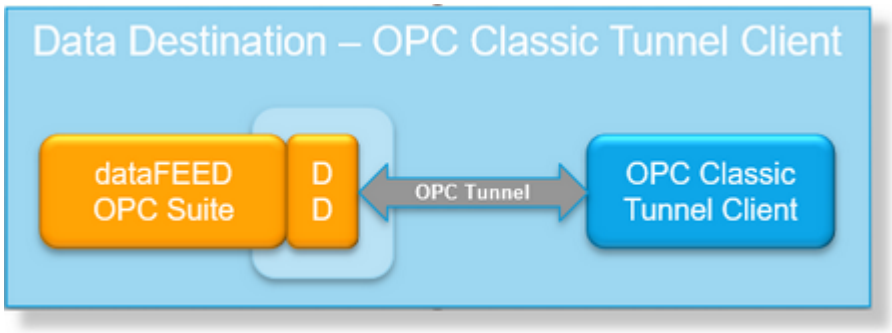
Note: Each local dataFEED OPC Suite application running on the same machine must use a unique TCP/IP port number. This port number should not be used by other applications running on the same machine.

Some additional options of the Data Destination OPC Client can be configured under Configuration: Settings - Advanced.

7.3.3.3 OPC Classic Tunnel Client

This section of the Softing dataFEED OPC Suite Online Help describes how to configure the OPC Classic Tunnel client data destination functionality of Softing dataFEED OPC Suite.

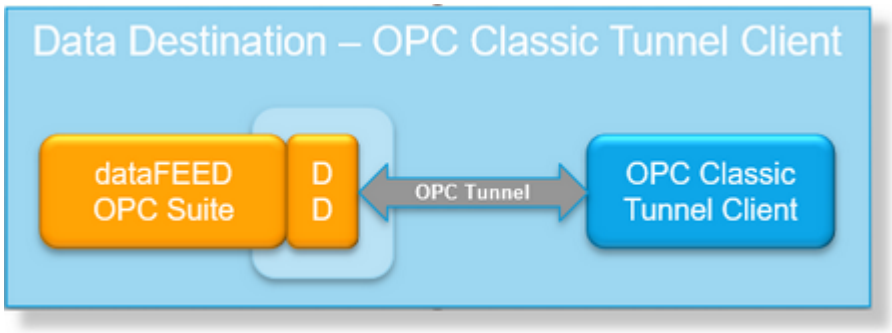
© Softing Industrial Automation GmbH



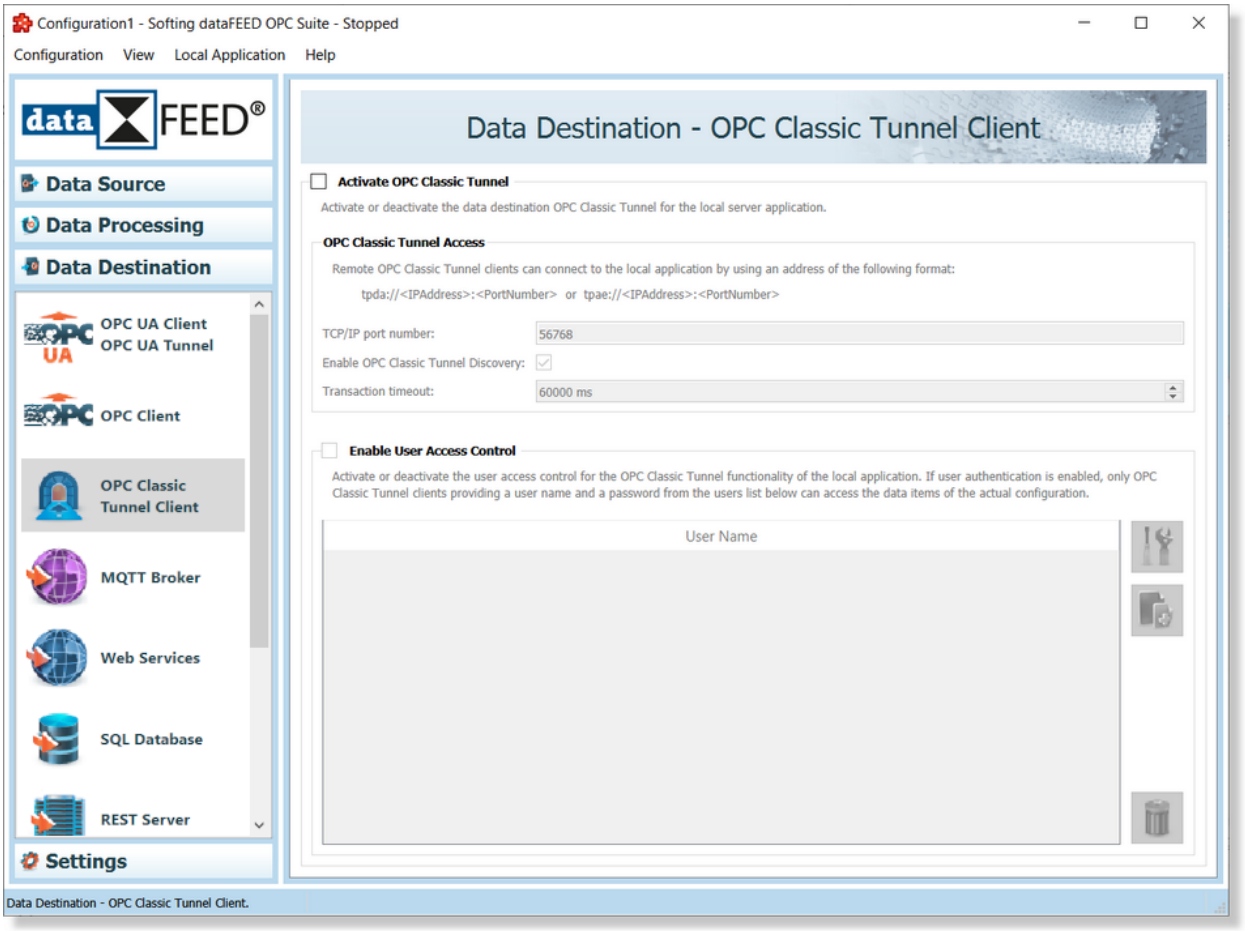
Enabling the OPC Classic Tunnel client data destination functionality allows external OPC Classic Tunnel and ABB Freelance Tunnel clients to access the data and alarm items of the local dataFEED OPC Suite application via the Softing OPC Tunnel protocol.

General information on the Softing OPC Classic Tunnel is provided under Introduction - OPC Classic Tunnel. A more detailed description of how to configure the OPC Classic Tunnel client data destination functionality can be found in the Configuration: Data Destination - OPC Classic Tunnel Client section.

7.3.3.3.1 Configuration

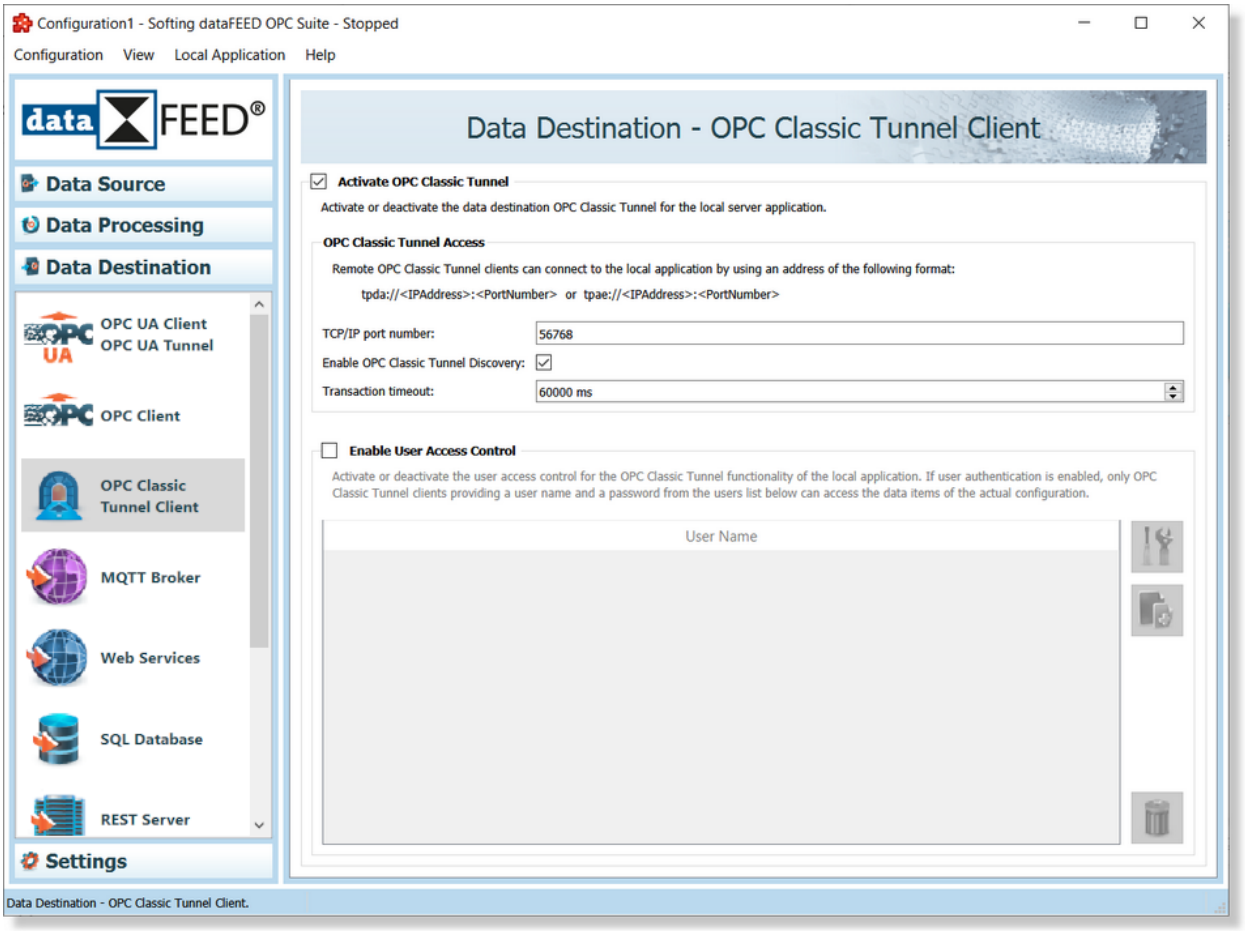


The OPC Classic Tunnel Client functionality is disabled by default.



This functionality can be enabled by checking the Activate Tunnel check-box on the main Tunnel Client configuration page. This page also allows modifying various settings of the Tunnel Client. The OPC Classic Tunnel Client main configuration page can be accessed by selecting *OPC Classic Tunnel and ABB Freelance Tunnel* clients from the *Data Destination* group in the navigation panel on the left.

© Softing Industrial Automation GmbH



Note: The OPC ClassicTunnel functionality is disabled by default. Enabling it will allow tunnel clients access to the namespace of the local dataFEED OPC Suite application.

The OPC Classic Tunnel Client configuration window contains editable fields for the basic settings needed by an OPC Classic Tunnel client to access the local dataFEED OPC Suite application address space. The table below provides a short description of the individual configuration options:

Attribute	Description
Activate OPC Classic Tunnel	Enables or disables access to the local address space for OPC Classic Tunnel clients.
TCP/IP port number	Port number used by the Tunnel client to communicate with the local dataFEED OPC Suite application running for the current configuration.
Enable OPC Classic Tunnel Discovery	Enables or disables the registration of the local dataFEED OPC Suite application with the *OPC Classic Tunnel Discovery Server*.

Transaction timeout	Defines the OPC Classic Tunnel Client transaction timeout.
Enable User Access Control	Restricts access of remote Tunnel clients by requesting authentication. Several users and passwords can be defined in the *User Name* list of the *User Access Control* section.
User Name list	Shows the users allowed to connect to the namespace of the local dataFEED OPC Suite application through the Tunnel interface. The list can be edited with the buttons provided on the right side of the main Tunnel Client configuration page.

Note: The default port define for ABB Freelance Tunnel Client is 9950. Still, please have in mind that Softing dataFEED OPC Suite is running in a multiple instance mode - this limits the usage of a unique port to a single instance.

Enable User Access Control

In the *Enable User Access Control* section you can activate or deactivate the user access control (also referred to as user authentication) and define the corresponding users which may access the local namespace through a Tunnel client.

The user name list and the buttons are only enabled if the encapsulating checkbox is marked.

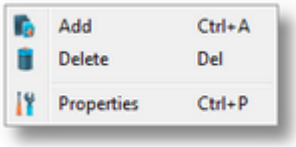
Note: If no user is specified (although the *Enable User Access Control* check-box is enabled) or if the user access control is disabled, access will be granted to all connecting Tunnel clients without any restrictions.

From the *Tunnel Client* main user interface certain operations can be performed (on the selected user) by using the buttons on the right, accessing the context sensitive menu or using the keyboard shortcuts. The following actions are available:

- Access the properties of the currently selected user (change the user password) by using the *Properties* button  , pressing the keyboard shortcut CTRL+P or selecting the *Properties* action from the context menu.
- Add a new user by using the Add button  , pressing the keyboard shortcut CTRL+A or selecting the Add action from the context menu.
- Delete the currently selected user by using the *Delete* button  , pressing the keyboard key DEL / DELETE or selecting the *Delete* action from the context menu.

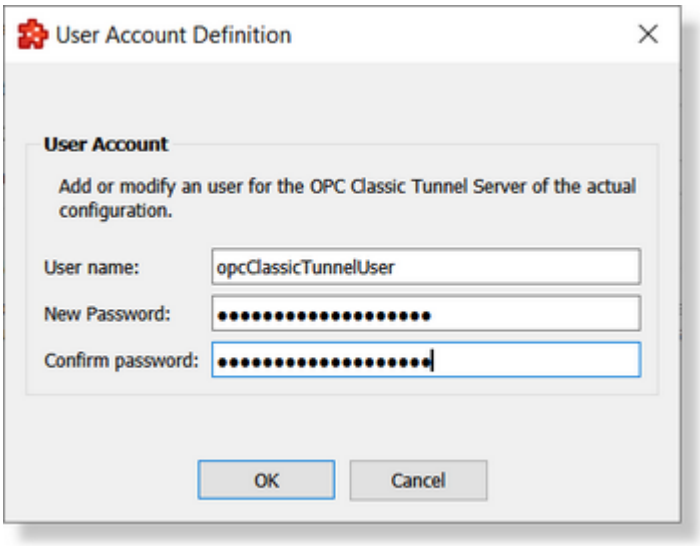
An example of the context sensitive menu can be seen below:

© Softing Industrial Automation GmbH

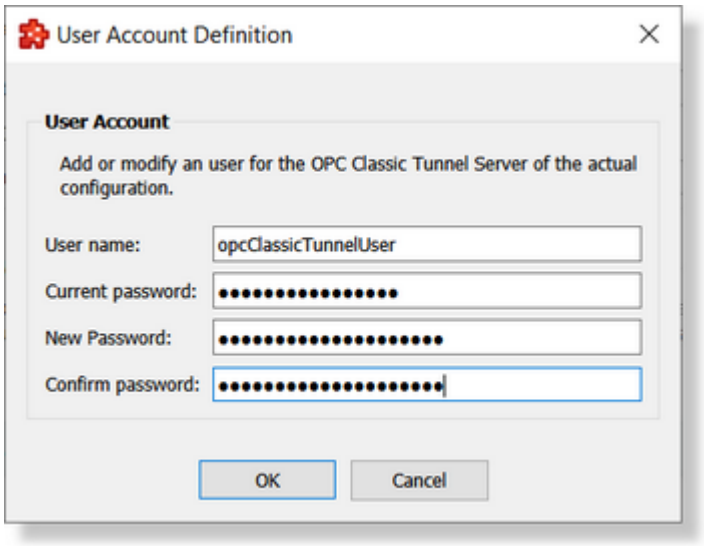


User Account Definition

When adding a new user to the list, the system requires the user name and the password. For this purpose the *User Account Definition* dialog is shown:



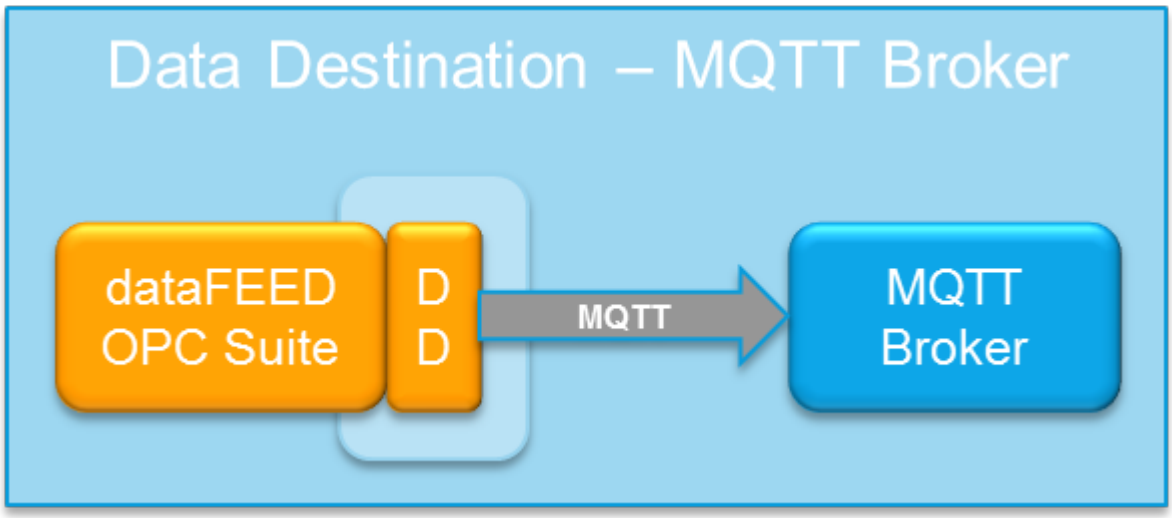
When modifying an existing user account, you can change the user name and password. However, to do so you need to enter the old password as well.



In any of the above dialogs (add / modify user) the changes can be applied by pressing the *OK* button – or they can be discarded by pressing the *Cancel* button.

7.3.3.4 MQTT Broker

This section of the Softing dataFEED OPC Suite online help explains how an MQTT Broker is configured as data destination by using the functionality provided by Softing dataFEED OPC Suite.



The MQTT protocol knows the following subscribers:

1. MQTT Broker: The MQTT Broker is the central instance in an MQTT network. The other subscribers establish each a TCP or SLL/TLS connection to the broker. Depending on the MQTT Broker configuration an authentication with user name and passwords or an SSL certificate is required. The MQTT Broker receives data from the MQTT Publishers. If a certain publisher does not have the required write permissions, the corresponding date is discarded by the broker. Otherwise the broker provides the data to all MQTT Subscribers that have subscribed to this data.

2. MQTT Publisher: The MQTT Publisher create the MQTT address space (topics) and fill this space with content (data). The MQTT Publisher sends this data to the MQTT Broker. This is exactly the role of the dataFEED MQTT Publisher Module.

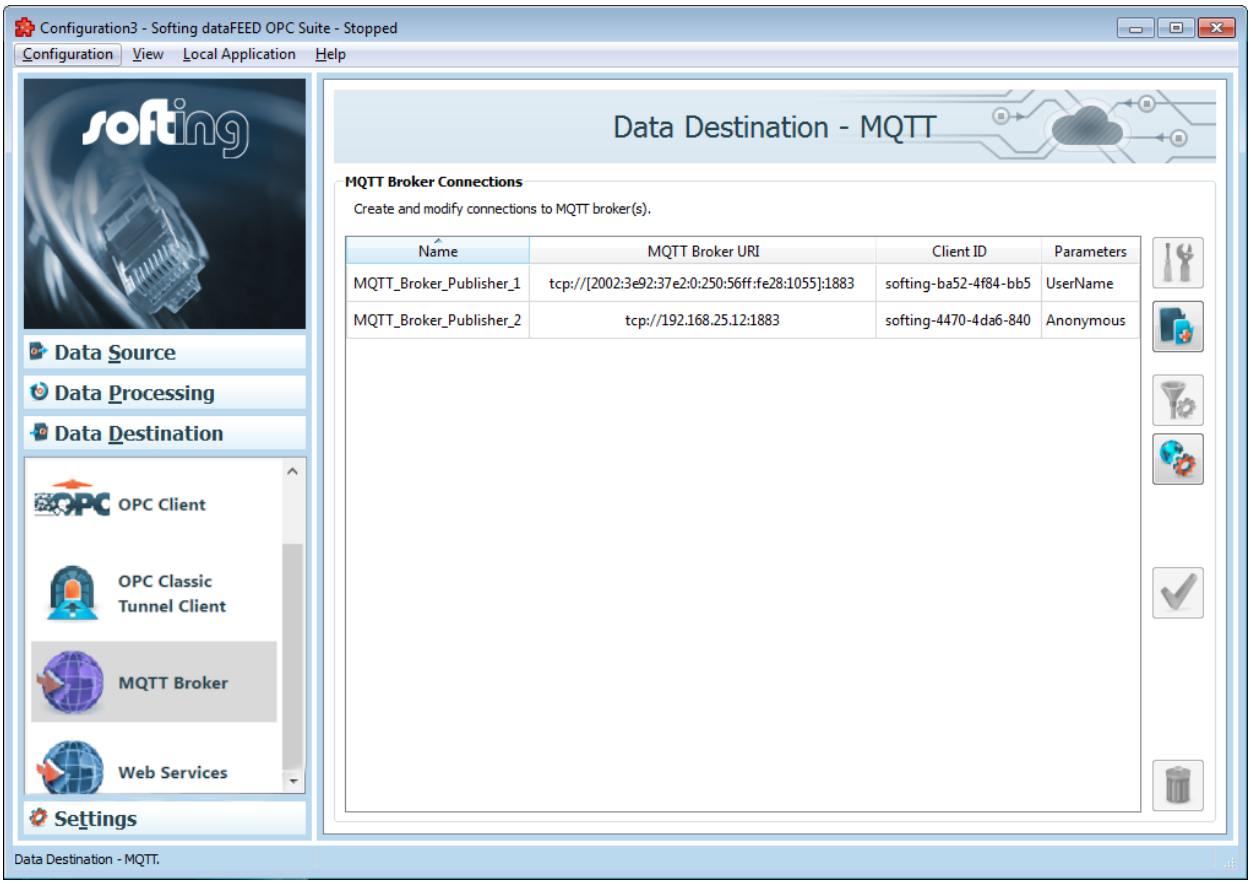
3. MQTT Subscriber: The MQTT Subscribers subscribe to MQTT Topics. In this context they are allowed to use the wild card characters '+' and '#'. Thus, an MQTT Subscriber that only uses the character '#' for topic subscription, subscribes **all** data from a broker.

The data format is not specified by the MQTT protocol specification but can be specified by the respective MQTT publisher. The dataFEED MQTT Publisher Module uses UTF 8-coded strings as data format.

© Softing Industrial Automation GmbH

7.3.3.4.1 Configuration

In order to configure connections to MQTT Brokers select **Data Destination** in the left window part and then select the MQTT Broker icon. The MQTT connection overview is opened.



Add an MQTT connection

You can establish a connection to an MQTT Broker by using the button *Add*  or the menu command "Add" in the context menu. The MQTT connection wizard is opened.

Modify an existing MQTT connection

The configuration of an existing connection to an MQTT Broker can be modified. To do so, select the line in the connection overview that contains the connection to be modified. Then you can use the button *Properties*  or the menu command "Properties" from the context menu to open the MQTT connection wizard.

Modify the address space of an existing MQTT connection

The address space of an MQTT connection is composed of "topics". You can specify these topics within the MQTT connection wizard.

An existing connection can be modified directly in the Address Space Definition page of the wizard by selecting the connection in the connection overview. Then you can or press the address space definition button  or the corresponding context menu entry.

Connection test

Click test connection button  to check the availability of specified server or the correctness of the authentication and security settings. If the backup broker is enabled, the application performs a connection test on both brokers and displays a combined result page containing also the correctness of the backup broker configuration.

Remove a connection

Delete the selected data source by using the *Delete*  button, pressing the DEL / DELETE key on the keyboard or selecting *Delete* from the context menu.

Note: If no connection is selected, the *Add a new data source* button and Global settings are the only options enabled.

The *Delete* button  will remove the selected device connections from the list.

The user can select more than one server connection with the help of the CTRL or SHIFT keys.

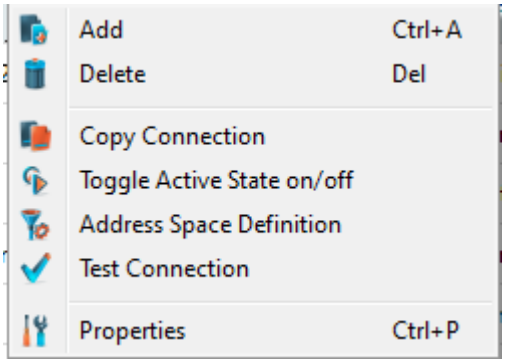
A dialog will then prompt the user to confirm the deletion.

Attention: A device connection cannot be deleted if at least one data item defined in its namespace is used in a Processing or Data Destination action.

These actions are also provided in the context menu of the External device list. Which actions are available depends on the current selection in the list.

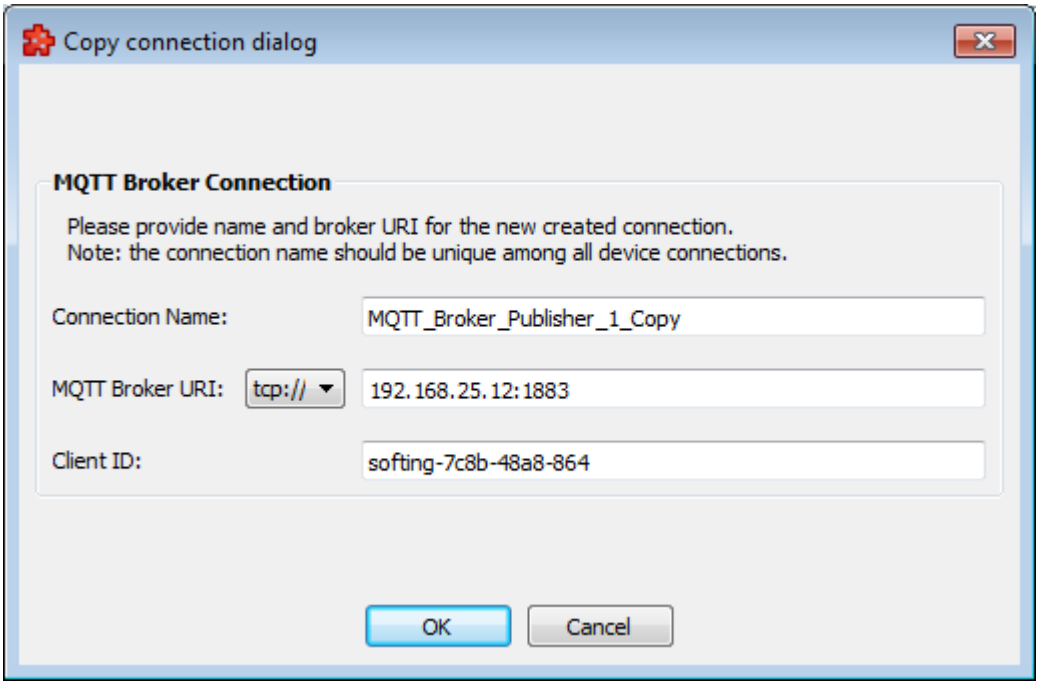
Following an example of a context menu:

© Softing Industrial Automation GmbH



Copy Connection

 . This function is available only in the context menu and opens a dialog where you can enter name and Broker URI.



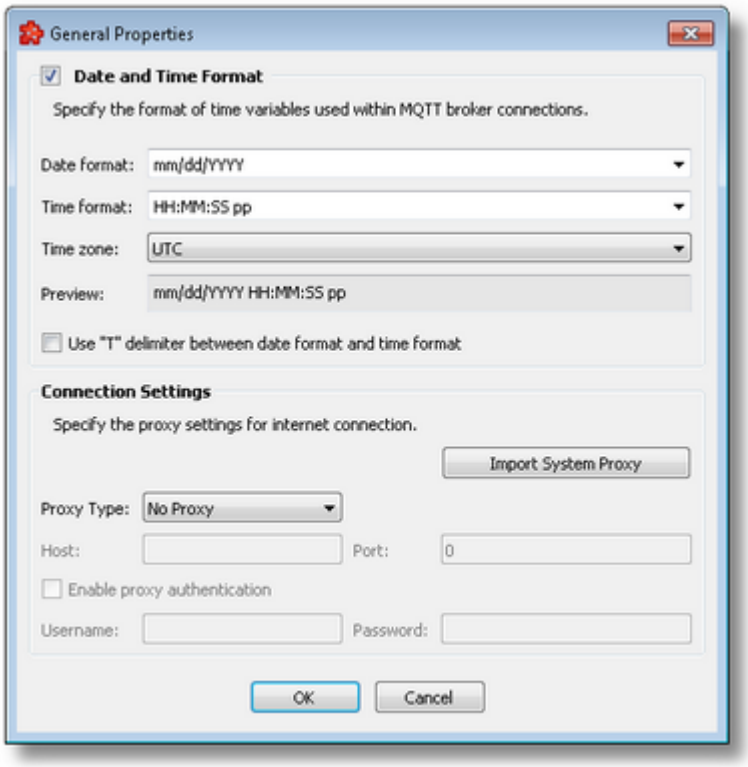
The proposed name is the original connection's name post fixed with "_Copy". The endpoint URL is copied from the source connection.

Note: The connection name should be unique among all device connections names. In addition, the client ID for the respective MQTT Broker must be unique.

Toggle active state on/off

Toggle active state on/off  . This action will enable / disable a device's connection. A disabled connection will be shown as grayed out in the device connection list and it will be ignored by runtime.

General Properties



The **Date and Time Format** section handles the different formats which can be specified when items of date and time are used in actions, also the source localization time: UTC (Universal Time Coordinated) or local time (local time localization). The default time zone is UTC.

The option *Use "T" delimiter between date format and time format* allows to obtain custom formats such as `YYYY-MM-DDTHH:MM:SS.FFFTZD`, by inserting the "T" delimiter between the selected date and time format values, and appending the `TZD` time zone designator expression.

For UTC time zone selected the `TZD` has the value `Z` (`YYYY-MM-DDTHH:MM:SS.FFFZ`), and the for Local time zone selected has the value `+hh:mm` indicating a time zone offset ahead of UTC (`YYYY-MM-DDTHH:MM:SS.FFF+hh:mm`), or `-hh:mm` indicating a time zone offset behind UTC (`YYYY-MM-DDTHH:MM:SS.FFF-hh:mm`).

The Date and Time format settings apply to item data values of data type date and time (DATETIME), and to item data values configured to the data value types: Timestamp (without milliseconds), Timestamp (with milliseconds), Internal Timestamp (without milliseconds), or Internal Timestamp (with milliseconds). For more details on the latter data value attributes, please see the description at Data Values Placeholder.

The default values are `mm/dd/YYYY` for the *Date format and* `HH:MM:SS pp` for the *Time format*. Other defined values for Time and Date format are:

© Softing Industrial Automation GmbH

- *Date format*: `mm/dd/YYYY, dd-mm-YYYY, dd.mm.YYYY, dd/mm/YYYY, YYYY-mm-dd, mm-dd-YYYY`
- *Time format*: `HH:MM:SS pp, HH:MM:SS.FFF pp, HH:MM:SS.FFF, HH:MM:SS`

Note: Item data values of data type date and time (DATETIME) that have a time format setting containing the milliseconds also (`FFF`), will display zero milliseconds value.

Note: Values for Time and Date format can also be defined manually but in this case special care must be taken to ensure that the definitions are recognized by the local dataFEED OPC Suite application.

In addition to the default values listed in the format selection combo boxes, the user can add his/her own custom format. Custom Date and Time format strings are accepted as valid, when the format strings contain in their composition the following accepted format substrings and format delimiters. These formats allow the inclusion of specific locale information for the regional settings.

- *Valid Date format strings*: `AAAAAA, aaa, BBBBBBBB, bbb, ccccccccccccccccccc, dd, jjj, mm, UU, ww, xxxxxxxxxx, yy, YYYY`
- *Valid Time format strings*: `II, HH, MM, SS, pp, XXXXXXXX, FFFFFF, FFFFF, FFFF, FFF ,FF, F`
- *Valid Date and Time format delimiters*: `<dot>, <colon>, <slash>, <comma>, <backslash>, <vertical bar>, <semicolon>, <space>, <tab>, <minus>`

The **Connection Settings** section allows for the customization of proxy settings. The user can either import system proxy settings automatically (*Import System Proxy* button), or specify manually the host, port and authentication settings.

These settings will be applied to all connections in the plugin, and the following parts of the functionality will be affected: test connection, getting the server certificate and action execution. This means that if the network, on which the dataFEED is configured, is behind a proxy server and dataFEED is not configured with a proxy, like in the image above, the mentioned pieces of functionality will fail. In order for them to work correctly, the user needs to specify the type of proxy, the host name of the server, a port number and user credentials, if necessary.

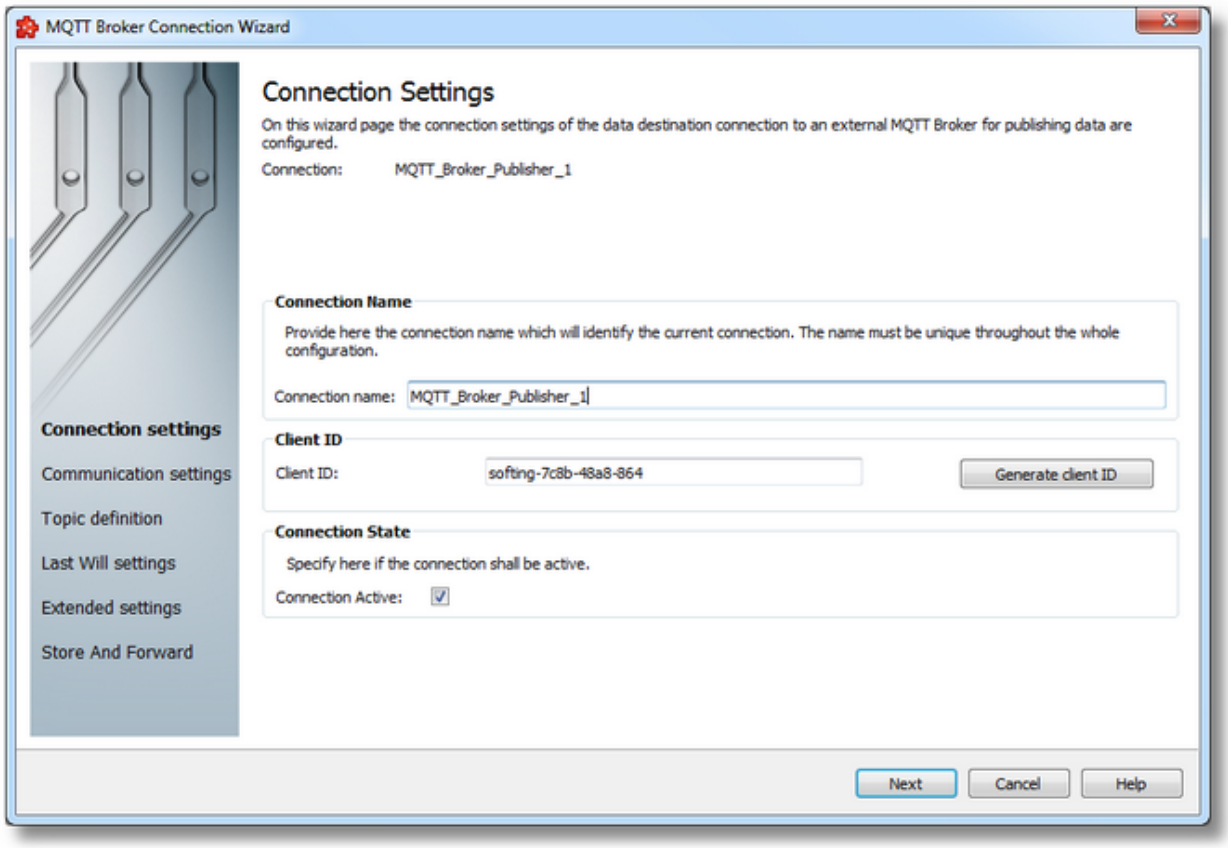
7.3.3.4.2 MQTT Connection Wizard

With the MQTT Connection Wizard you can create and modify a data destination connection to an MQTT Broker.

- Connection Settings
- Communication Settings
- Address Space Definition
- "Last Will" Settings
- Extended Settings

- Store And Forward

This page allows to specify connection settings such as e.g. the unique connection name for identifying the connection.



Connection Name

A unique identifier for the current connection. In case the provided name is already used by other connection, the user use is not allowed to proceed to the next wizard page.

Client ID

The client ID must be unique for the respective MQTT Broker. If two MQTT clients are using the identical client ID, the broker is disconnected. The client (as e.g. dataFEED) cannot recognize that the client ID is already used by another client.

The MQTT Broker can be configured in a way that it only accepts connections with specific client IDs.

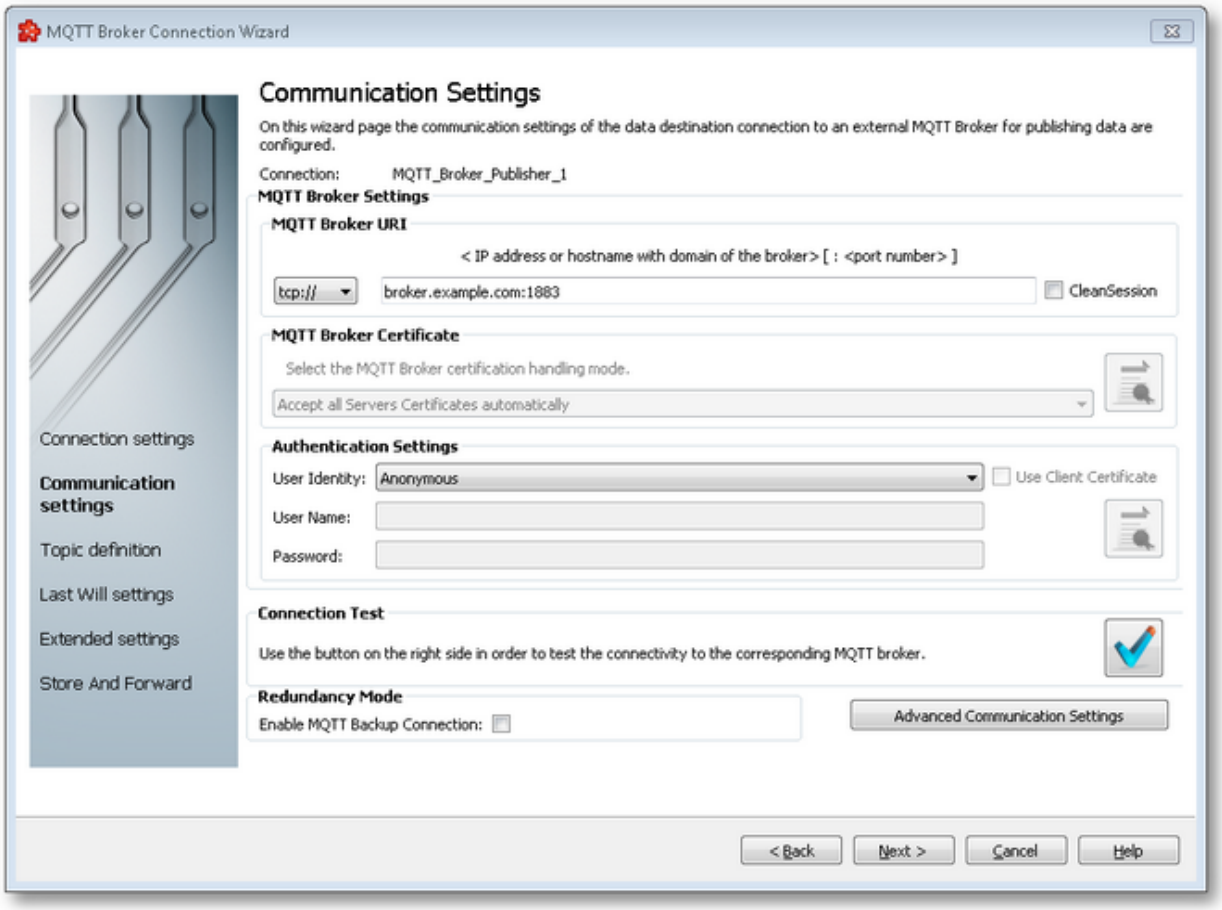
Connection active

© Softing Industrial Automation GmbH

This flag specifies if the current connection will be activated or not.

Connection Wizard - Comunication Settings >>>

This page allows to parameterize the communication and authentication of the MQTT Broker.



Broker URI

The Broker URI is composed of the transport protocol, the Full Qualified Domain name (FQDN, consisting of host name and domain name [see wikipedia]) optionally followed by the port number separated by a colon.

For the transport protocol, the following variants could be chosen:

1. tcp:// MQTT uses the plain TCP. MQTT brokers usually offer this service at port 1883.
2. ssl:// MQTT uses SSL or TLS security upon TCP. MQTT brokers usually offer this service at port 8883.
3. ws:// MQTT over WebSockets. MQTT brokers usually offer this service at port 8000.

4. wss:// MQTT over secure WebSockets. MQTT brokers usually offer this service at port 443.

The FQDNS can be replaced by the corresponding IPv4- or IPv6-address.

MQTT Broker Certificate

If the transport protocol ssl:// has been chosen, you may configure here how the identity of the MQTT broker should be verified by the dataFEED MQTT publisher. You have the choice between the following options:

1. Accept all Server certificates axiomatically: This option disables verification of the broker identity. The broker just needs to provide some valid certificate.

2. Accept Trusted Certificates only: Connections are only completely established, if the broker proofs its identity with the certificate chain, which has been assigned to the connection. This option is only available, if you have assigned a broker certificate chain for this dataFEED connection with the Certificates dialog.

Authentication settings

Depending on the MQTT Broker configuration dataFEED may need to identify itself as an MQTT client with user name and password or by a client certificate.

In this case you need to select the entry "Username and Password" in "User Identity" Then fill out the input fields for user name and password accordingly. If needed activate the "Use Client Certificate" check box and assign a corresponding client certificate file through the MQTT Client Certificate dialog.

If no identification is required, leave the entry "Anonymous" in "User Identity".

Note: dataFEED OPC Suite does not use a strong encryption to protect the passwords contained in your configuration files. Therefore we recommend, to protect the PC where dataFEED OPC Suite is installed against unauthorized access. In addition you can use a hard disk encryption for protection purposes.

Connection test

Use the button Connection Test  to allow dataFEED OPC Suite establishing a connection to the MQTT Broker with the specified parameters for test purposes.

Redundancy mode

This checkbox enables an alternative broker, which is activated as soon as the connection to the original broker gets lost. Enabling this option automatically activates the Store and Forward functionality and disables the Clean Session flag.

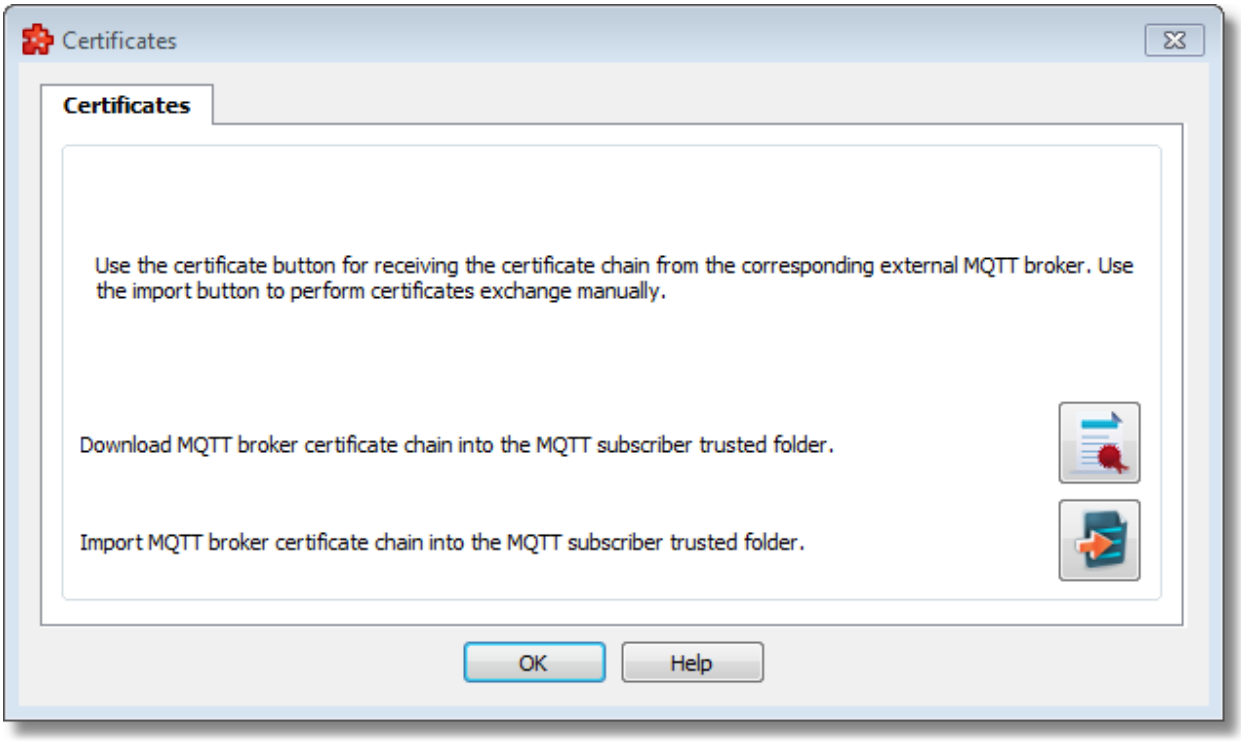
© Softing Industrial Automation GmbH

Advanced Communication Settings

Specific communication protocol settings related to the MQTT Broker connection can be provided through the Advanced Communication Settings dialog.

<<< Connection Wizard - Connection Settings Connection Wizard - Backup Broker Connection >>>

This dialog provides options to facilitate the secure communication between client and external broker by certificate handling.



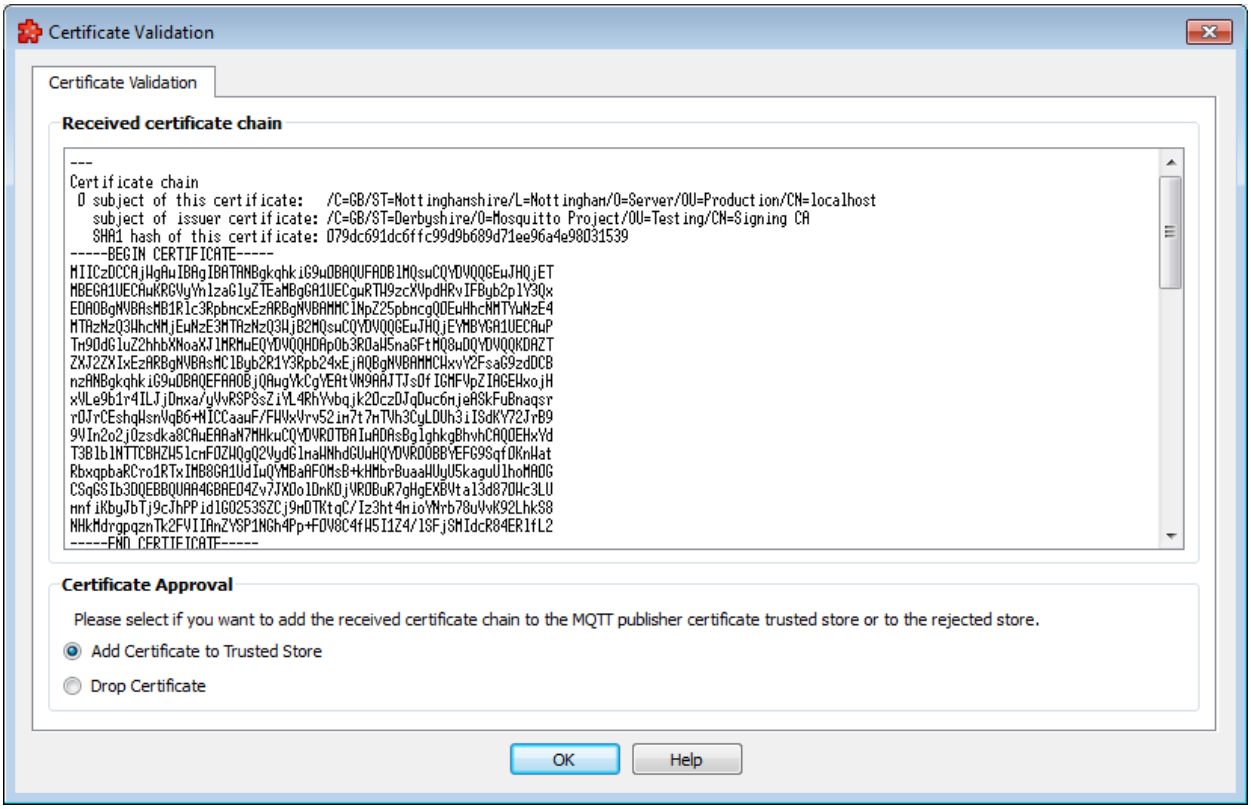
Download MQTT broker Certificate chain into the MQTT subscriber trusted folder

By pressing the receive broker certificate button  the user initiates a server connection and implicitly a certificate exchange. Once the server certificate chain is obtained it is presented to the user into certificate validation to be investigated. If the user accepts the certificate chain it would then added to the trusted folder and assigned to the connection. Otherwise the certificate chain is removed from the file system.

Import MQTT broker Certificate chain into the MQTT subscriber trusted folder

By using the import button  the user is prompted to select a broker certificate chain (*.crt or .pem* format) that will be added to the trusted folder. This operation basically can substitute Receive server certificate operation.

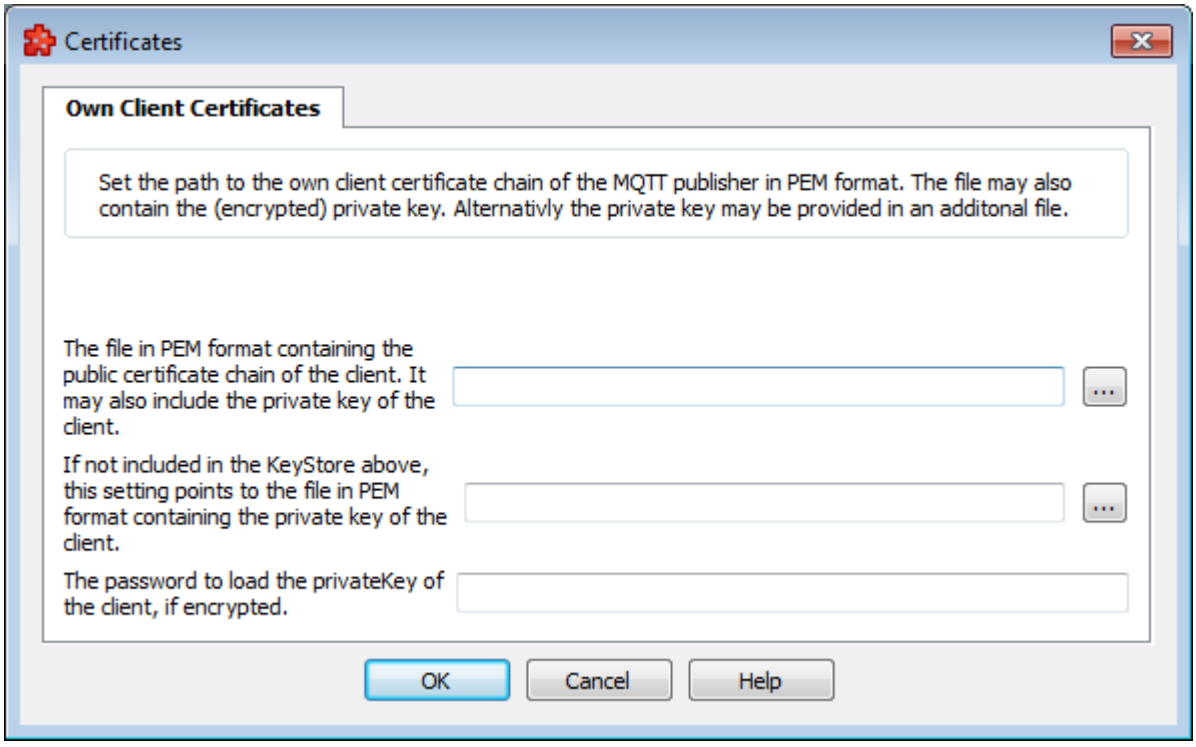
This dialog displays the received certificate chain of the MQTT broker.



The user may accept or discard the received certificate chain. If the user accepts the certificate chain it would then added to the trusted folder and assigned to the connection. Otherwise the certificate chain is removed from the file system.

This dialog allows to select the client certificate, which is intended to identify the client by the MQTT broker.

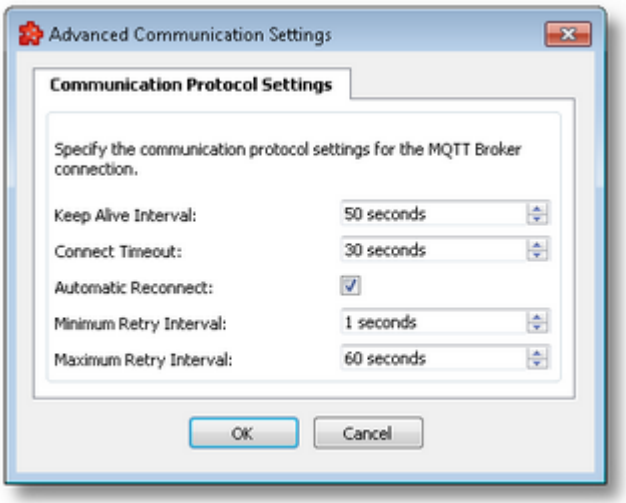
© Softing Industrial Automation GmbH



This dialog provides the following input fields:

1. The file in PEM format containing the public certificate chain of the client. It may also include the private key of the client.
2. If not included in the KeyStore above, this setting points to the file in PEM format containing the private key of the client.
3. The password to load the privateKey of the client, if encrypted.

The dialog allows to set specific communication protocol settings for the MQTT Broker connection.



Keep Alive Interval:

- The maximum time that should pass without communication between the client and the server. Set to 0 if no keep alive processing is needed (default value 50 seconds).

Connect Timeout:

- The time interval in seconds to allow a connect to complete (default value 30 seconds).

Automatic Reconnect:

- Reconnect automatically in the case of a connection being lost (default value enabled).

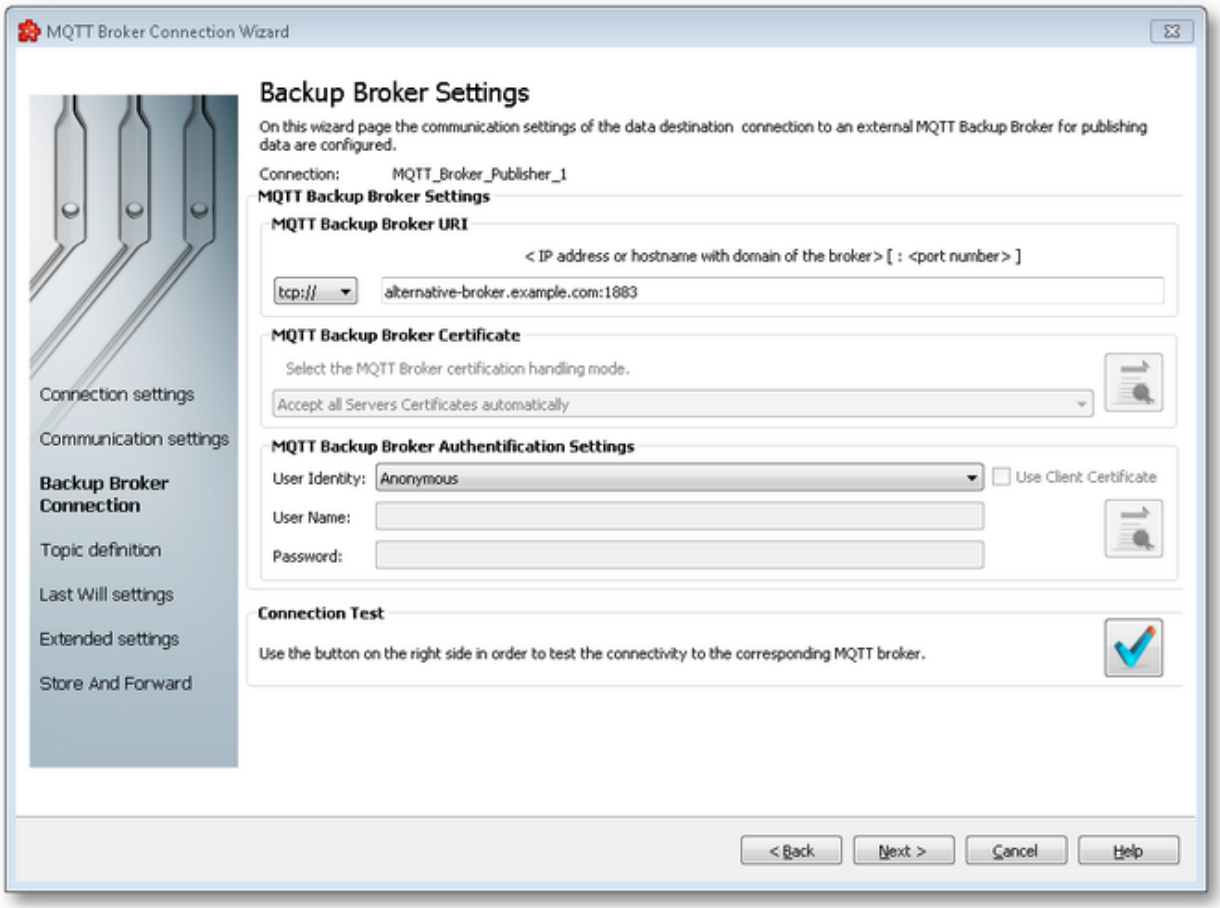
Minimum Retry Interval:

- Minimum retry interval in seconds. Doubled on each failed retry (default value 1 second).

Maximum Retry Interval:

- Maximum retry interval in seconds. The doubling stops here on failed retries (default value 60 seconds).

This page allows to parameterize the communication and authentication of the Alternative MQTT Broker.



© Softing Industrial Automation GmbH

Broker URI

The Broker URI is composed of the transport protocol, the Full Qualified Domain name (FQDN, consisting of host name and domain name [see wikipedia]) optionally followed by the port number separated by a colon.

For the transport protocol, the following variants could be chosen:

1. tcp:// MQTT uses the plain TCP. MQTT brokers usually offer this service at port 1883.
2. ssl:// MQTT uses SSL or TLS security upon TCP. MQTT brokers usually offer this service at port 8883.
3. ws:// MQTT over WebSockets. MQTT brokers usually offer this service at port 8000.
4. wss:// MQTT over secure WebSockets. MQTT brokers usually offer this service at port 443.

The FQDNS can be replaced by the corresponding IPv4- or IPv6-address.

MQTT Broker Certificate

If the transport protocol ssl:// has been chosen, you may configure here how the identity of the MQTT broker should be verified by the dataFEED MQTT publisher. You have the choice between the following options:

1. Accept all Server certificates axiomatically: This option disables verification of the broker identity. The broker just needs to provide some valid certificate.
2. Accept Trusted Certificates only: Connections are only completely established, if the broker proofs its identity with the certificate chain, which has been assigned to the connection. This option is only available, if you have assigned a broker certificate chain for this dataFEED connection with the Certificates dialog.

Authentication settings

Depending on the MQTT Broker configuration dataFEED may need to identify itself as an MQTT client with user name and password or by a client certificate.

In this case you need to select the entry "Username and Password" in "User Identity" Then fill out the input fields for user name and password accordingly. If needed activate the "Use Client Certificate" check box and assign a corresponding client certificate file through the MQTT Client Certificate dialog.

If no identification is required, leave the entry "Anonymous" in "User Identity".

Note: dataFEED OPC Suite does not use a strong encryption to protect the passwords contained in your configuration files. Therefore we recommend, to protect the PC where

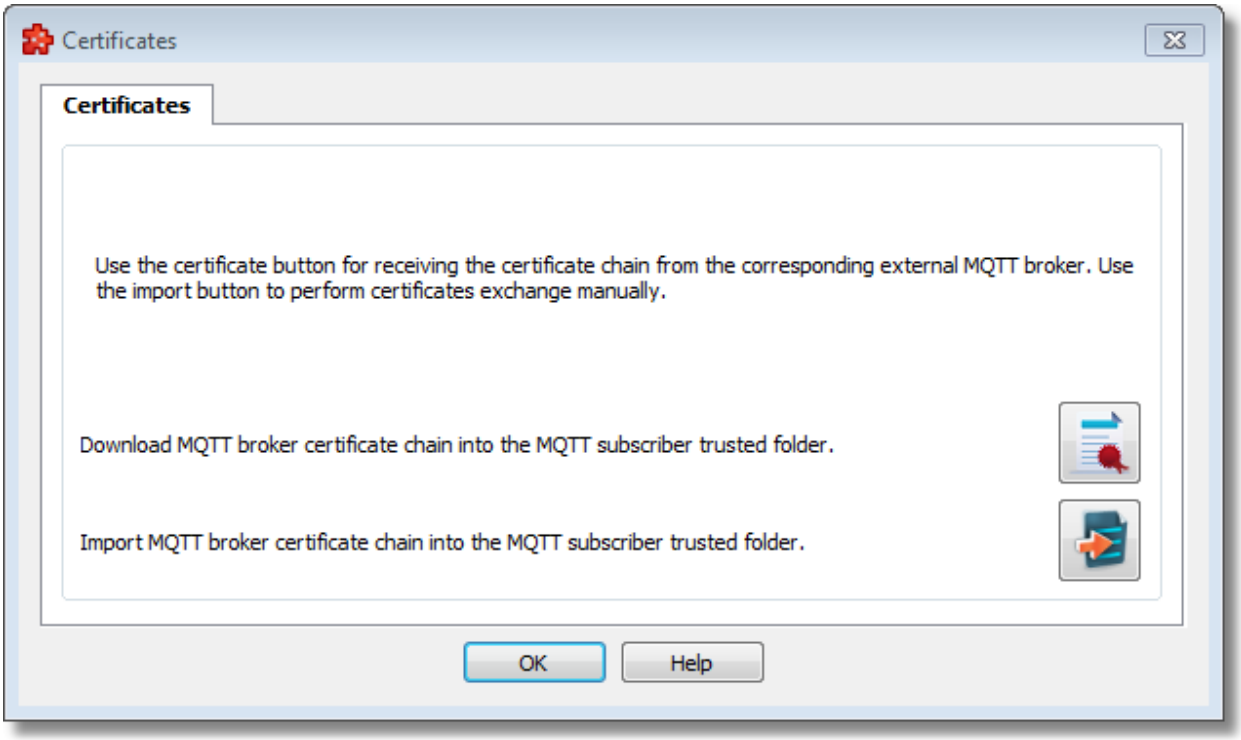
dataFEED OPC Suite is installed against unauthorized access. In addition you can use a hard disk encryption for protection purposes.

Connection test

Use the button Connection Test  to allow dataFEED OPC Suite establishing a connection to the Alternative MQTT Broker with the specified parameters for test purposes.

<<< Connection Wizard - Communication Settings Connection Wizard - Topic Definition >>>

This dialog provides options to facilitate the secure communication between client and external broker by certificate handling.



Download MQTT broker Certificate chain into the MQTT subscriber trusted folder

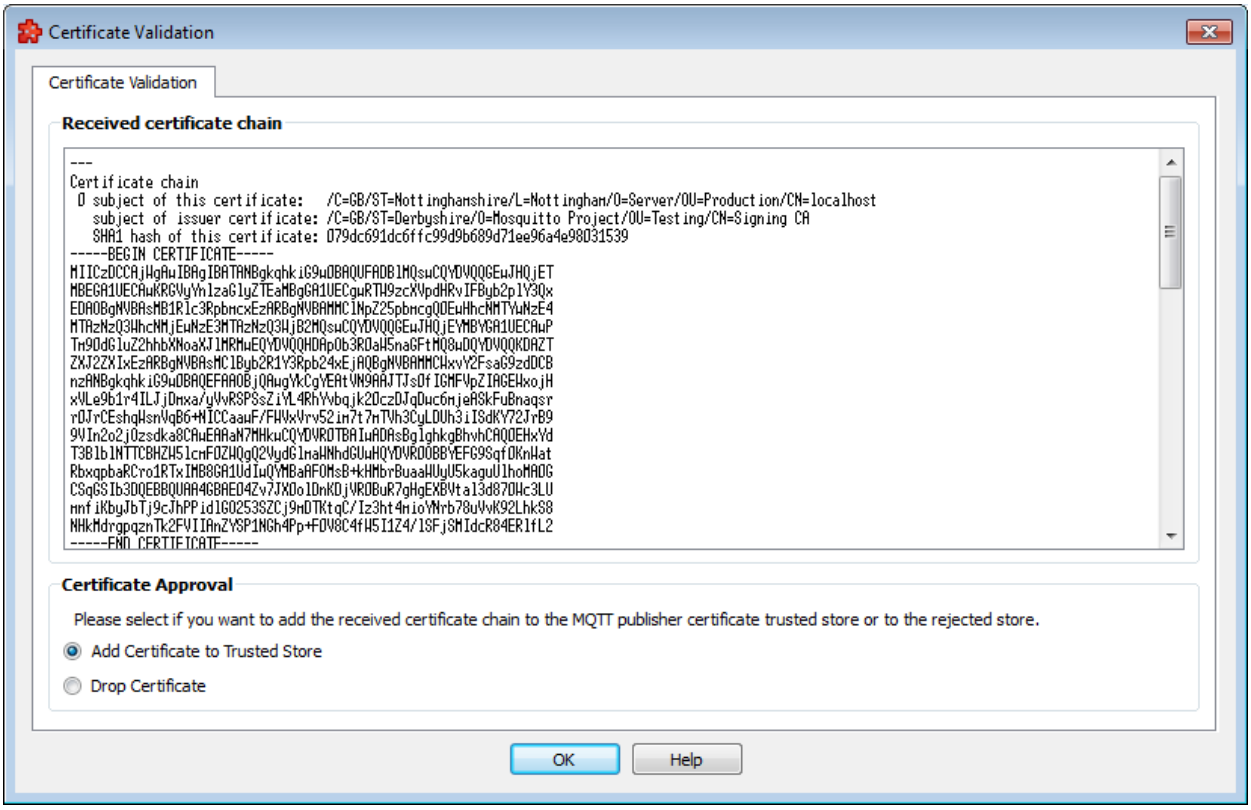
By pressing the receive broker certificate button  the user initiates a server connection and implicitly a certificate exchange. Once the server certificate chain is obtained it is presented to the user into certificate validation to be investigated. If the user accepts the certificate chain it would then added to the trusted folder and assigned to the connection. Otherwise the certificate chain is removed from the file system.

Import MQTT broker Certificate chain into the MQTT subscriber trusted folder

© Softing Industrial Automation GmbH

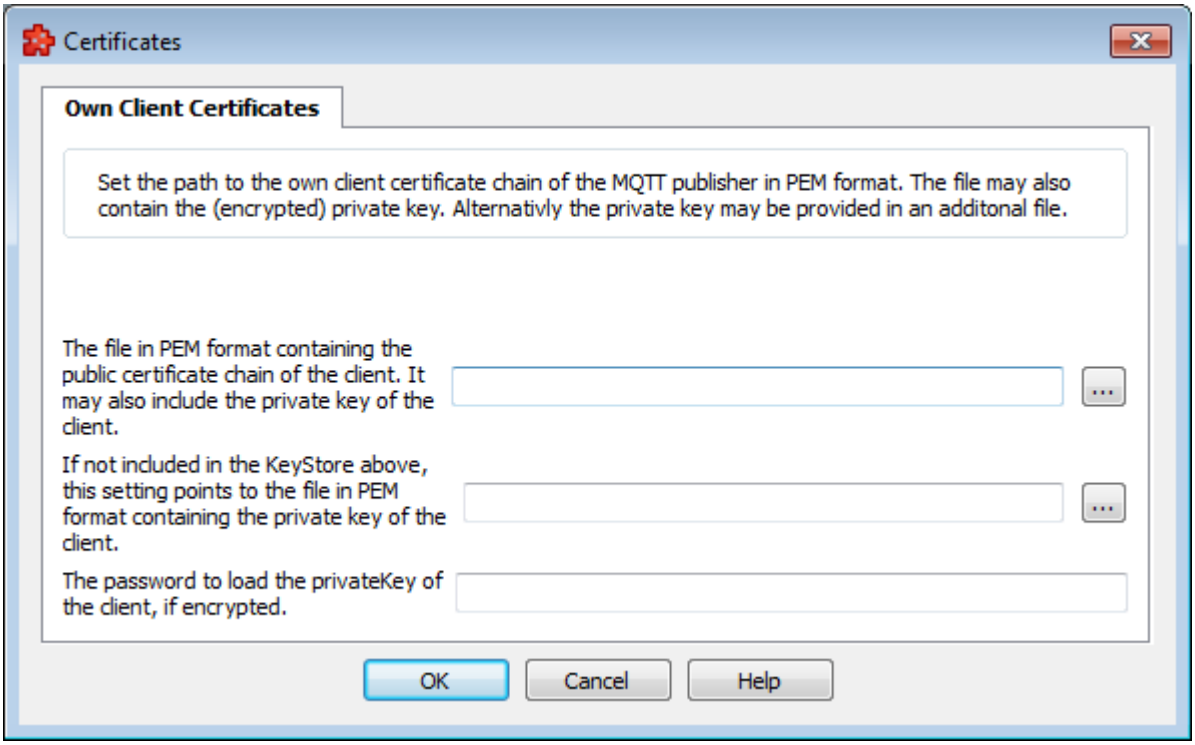
By using the import button  the user is prompted to select a broker certificate chain (*.crt or .pem* format) that will be added to the trusted folder. This operation basically can substitute Receive server certificate operation.

This dialog displays the received certificate chain of the Backup MQTT broker.



The user may accept or discard the received certificate chain. If the user accepts the certificate chain it would then added to the trusted folder and assigned to the connection. Otherwise the certificate chain is removed from the file system.

This dialog allows to select the client certificate, which is intended to identify the client by the MQTT broker.



This dialog provides the following input fields:

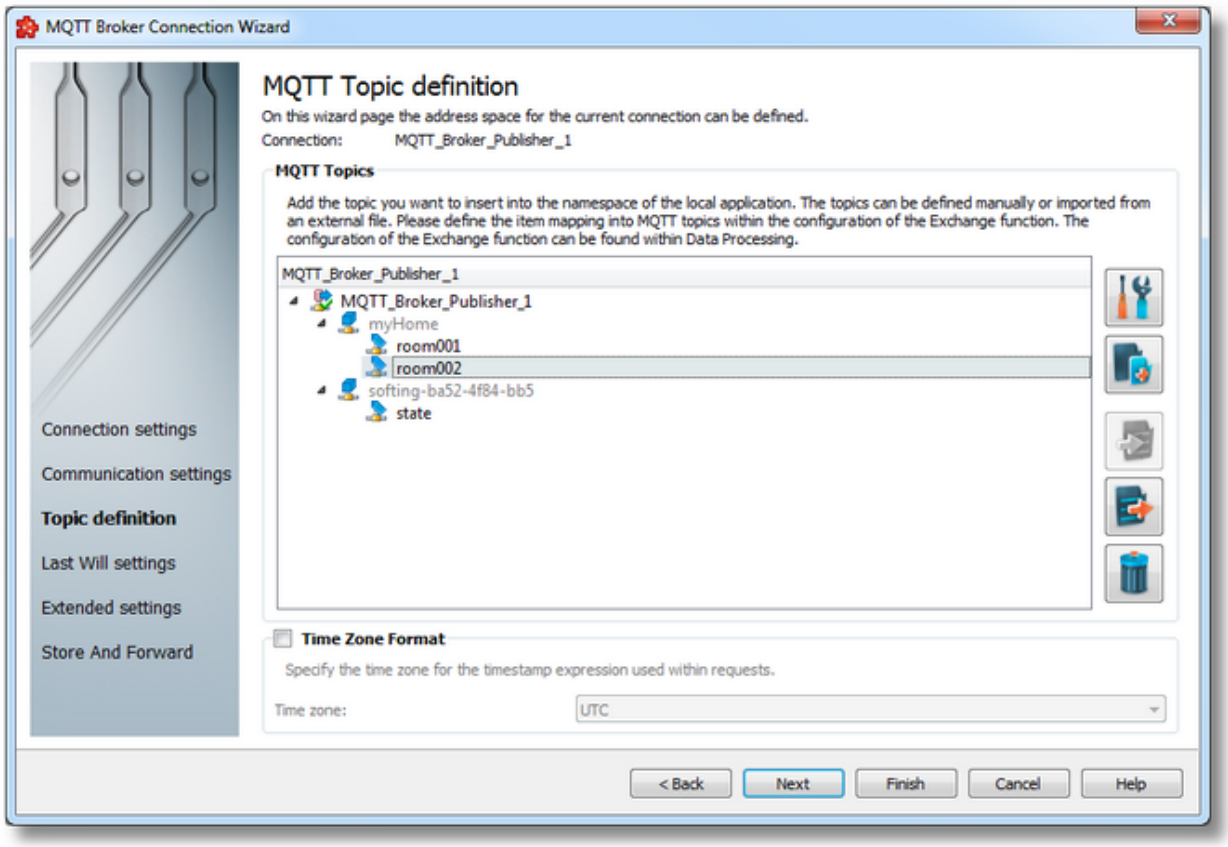
1. The file in PEM format containing the public certificate chain of the client. It may also include the private key of the client.

2. If not included in the KeyStore above, this setting points to the file in PEM format containing the private key of the client.

3. The password to load the privateKey of the client, if encrypted.

This page displays the MQTT address space of the MQTT Publisher and allows to configure the address space. The page also allows to specify an additional connection setting such as the time zone format.

© Softing Industrial Automation GmbH



MQTT Topics

- The MQTT address space is composed of topics.
- The individual hierarchy levels are separated by the '/' character. This character is inserted automatically by the address space wizard and must therefore not be entered manually.
- In addition you must not user the wildcard characters '+' and '#' of the MQTT Publisher. They are reserved for MQTT Subscribers.
- An editable tree view where the user can change the address space structure or its content by adding, deleting or modifying topics.

There are three categories of data items that may be used in the tree:

- **Root node** (having the name of the current connection). This node is read only, the user has just the possibility to add or delete children of this node.
- **Nodes:** these elements are associated with branches in tree. They can have child nodes. But no value can be written to the nodes.

Note: The items of type Node represent intermediate nodes in the tree, they are used for hierarchy purposes.

- **Tags:** these elements are associated with leaves in the tree.

Note: In MQTT the Publisher defines the address space; therefore adding a sub topic in MQTT Publisher results in changing a tag into a node.

Operation

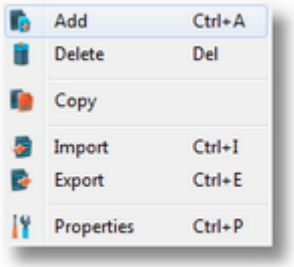
Depending on the current selection in the topic-tree, the following operations are possible using the buttons on the right, the context menu or the corresponding shortcuts:

- Topic Properties (Ctrl+P) 
- Add new topic(Ctrl+A) 
- Import from file (Ctrl+I) 
- Export to file (Ctrl+E) 
- Delete data items(Del) 

Two more operations are supported that are only accessible from the context menu:

- CopyData item(s) 
- Pasteto data item(s) 

An example for a context menu of the topic tree is shown below:



Delete data items

Triggering the *Delete data items*  action with the corresponding button on the right wizard side or *Delete* from the context menu will remove the currently selected topic from the Topictree.

© Softing Industrial Automation GmbH

Attention: The root node of the namespace tree can not be deleted.

Attention: A node that is currently used in a data processing or data destination action can not be deleted.

Copy

The *Copy*  command is available in the context menu of the Topic *tree*. Triggering this action will copy the selected topic to the local clipboard. All topic properties are also copied. Even if the topic contains child topics, the complete branch is copied.

Note: The root node of the namespace tree does not provide a *Copy* action.

Paste

The *Paste*  command is available in the *context menu of the Topic tree*. This function will add the topic currently stored in the local clipboard as a child of the currently selected topic. Whether the action is available in the context menu depends on the type of the currently selected node and the type of the node in the clipboard.

Note: The *Paste* action is only available if a topic was previously copied to the clipboard.

Time Zone Format

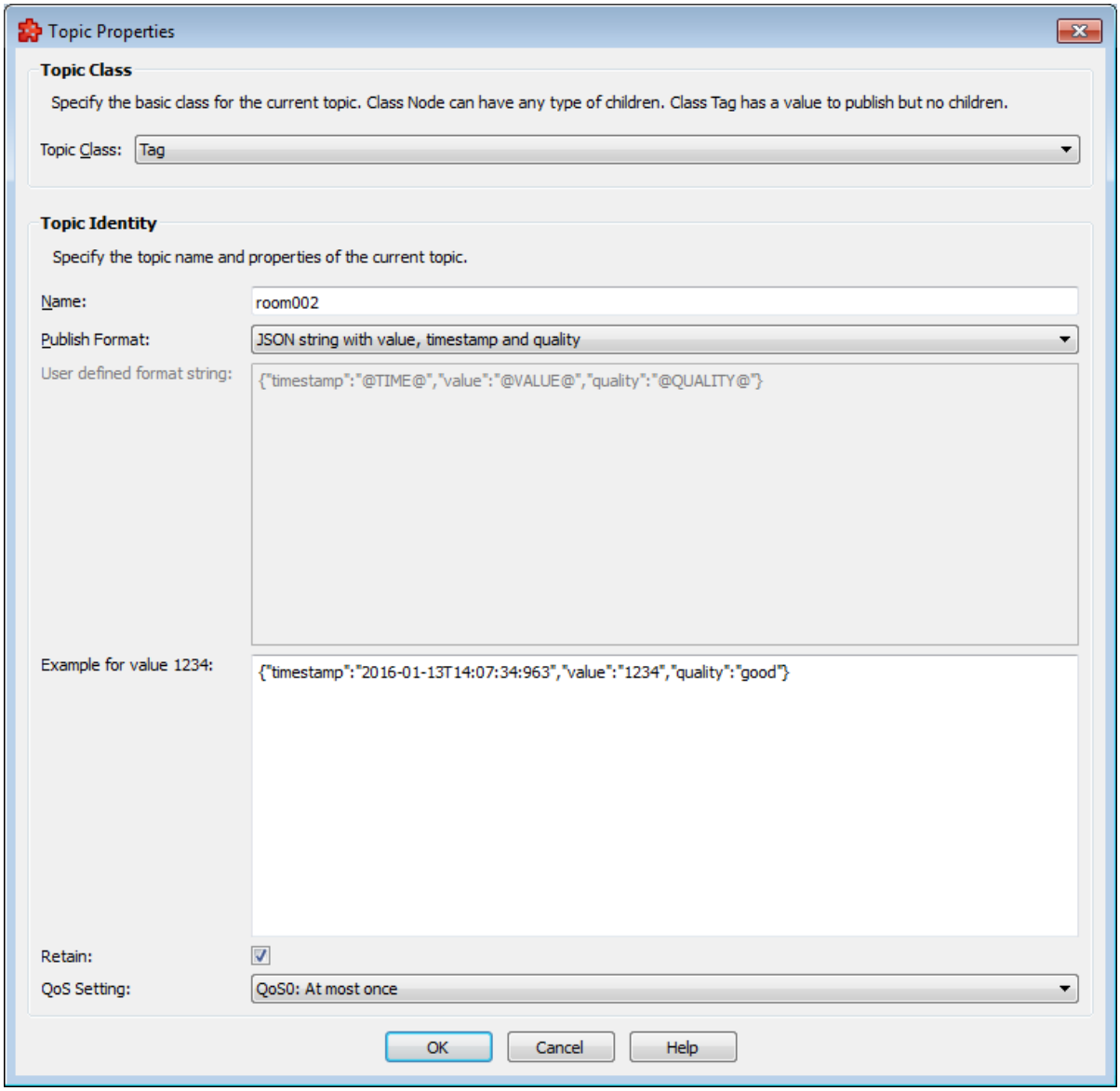
This section handles the time zone format that is to be used for the timestamp expression within the MQTT publisher's write transactions.

Time Zone handles the source localization time: UTC (Universal Time Coordinated) or local time (local time localization). The default time zone is UTC.

When the *Time Zone Format* option is not selected the default value is used. If this is modified, it applies to the currently selected MQTT connection.

<<< Connection Wizard - Communication Settings Connection Wizard - Last Will Settings >>>

This page allows you to specify the topic name as well as dataFEED OPC Suite-specific topic properties.



Topic Class

This list box allows to specify the topic class. You can choose between node and tag.

Topic Identity

Topic Name

Here you can enter the topic name for the current hierarchy level. The name you are using must be unique for the corresponding parent node. The name may only consist of printable ASCII characters. Special characters such as '.', '/', '+' and '#' from MQTT and dataFEED must not be used.

© Softing Industrial Automation GmbH

Format

The MQTT Publisher Module writes tags in UTF8-coded strings. You can specify for any topic the string formatting by using the drop-down list box:

1. JSON formats strings (as is usual in web applications) composed of time stamp, value, and quality attribute.
2. Bluemix JSON format string composed of time stamp, value, and quality attribute, suitable for IBM Bluemix cloud.
3. Plain text: a string that only contains a data value without supplements.
4. User defined format: The user could define the format string. The following, casesensitive keywords will be replaced with the corresponding data:
   a. @VALUE@: The value of the dataFEED item.
   b. @TIME@: The timestamp of the dataFEED item.
   c. @QUALITY@: The quality attribute of the dataFEED item.

Depending on your selection an example string for data value 1234 is displayed.

Note: Within the configuration of the exchange functionality the user is allowed to define the exchange values as a combination of value, time stamp and/or quality. This setting is ignored by the MQTT Publisher. The MQTT Publisher uses the format described above in "Format".

Retain

By setting the Retain flag the MQTT Broker is instructed to save the most recent data value for the topic. (Depending on the configuration the broker saves the data into the RAM or persistently into the file system/data base.) Data values without Retain flag are only transferred from the MQTT Broker to those MQTT Subscribers that are registered in that moment by the corresponding broker and have subscribed the corresponding topic.

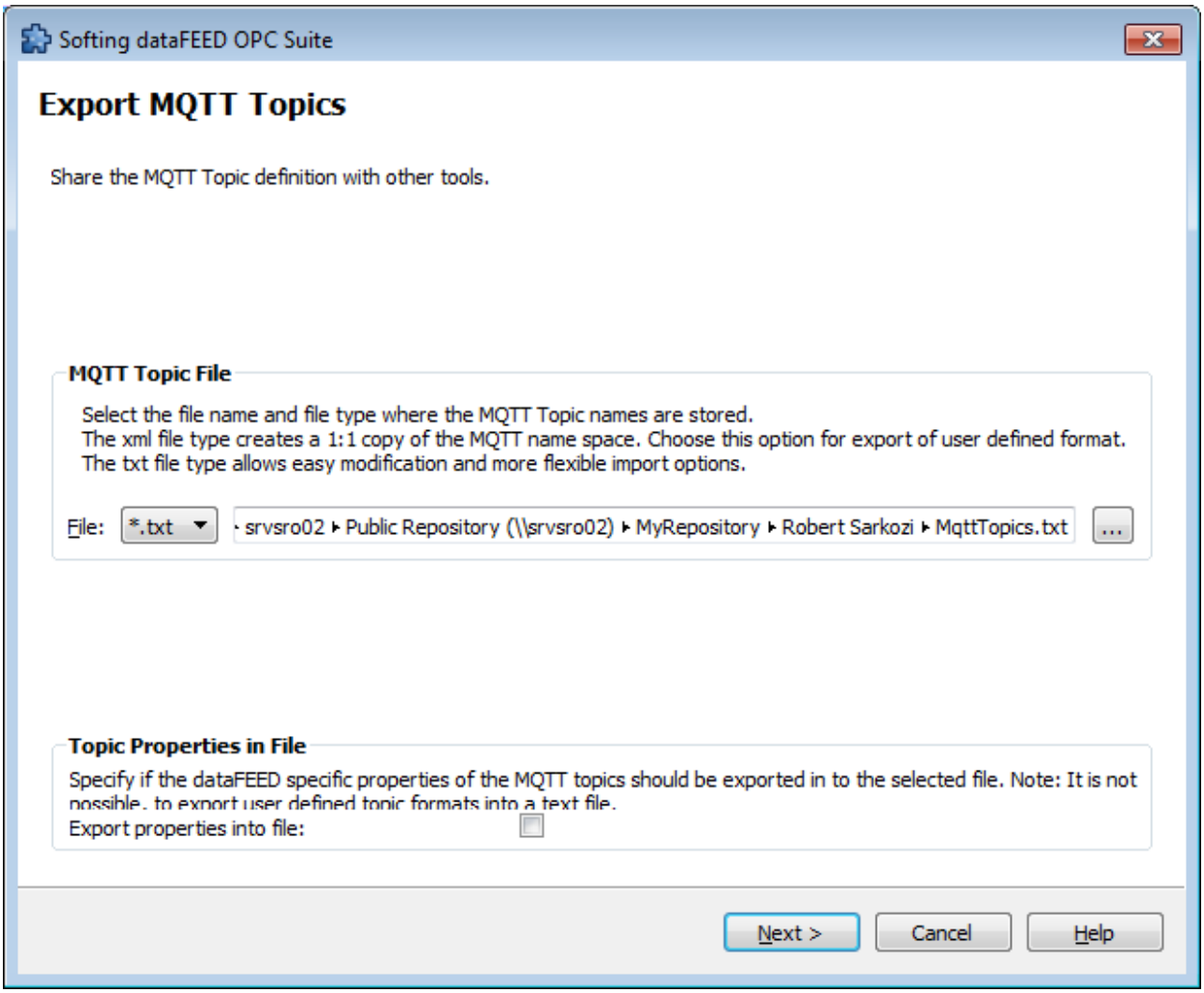
This check box allows to specify whether the MQTT Publisher Module sets the Retain flag for the respective topic or not.

QoS Setting

The QoS setting is used to control the delivery of MQTT messages in case of TCP connection loss. The following settings are possible:

1. QoS0: The MQTT message is published with the safety of the current TCP connection.
2. QoS1: A MQTT messge is repeated in the next TCP connection, if the delivery of the message in the current TCP connection fails.
3. QoS2: MQTT protocol uses confirmation of confirmations to ensure that a message is delivered exactly once.

This page allows to export the topics of a tree or sub tree into a text or XML file.



The check box **Properties in File** specifies whether dataFEED OPC Suite-specific topic properties are written into the file or whether not. It is not possible to export all complex properties like the user defined topic formats into simple text files. Therefore the user has the opportunity to choose a XML file as target. The XML file will contain the whole name space of the connection with all topic properties.

The export allows the following objectives:

1. The listed topics in the created text file allow to configure the MQTT subscriber so that the MQTT subscriber subscribe for the topics created by the dataFEED MQTT publisher. You can find detailed information in the description of the respective MQTT subscriber. Deactivate the check box **Topic properties in file** for this use case.

2. The file created allows to copy topics into another dataFEED instance. See also Import MQTT topics. Activate the check box **Topic properties in file** or choose a xml file for this use case.

© Softing Industrial Automation GmbH

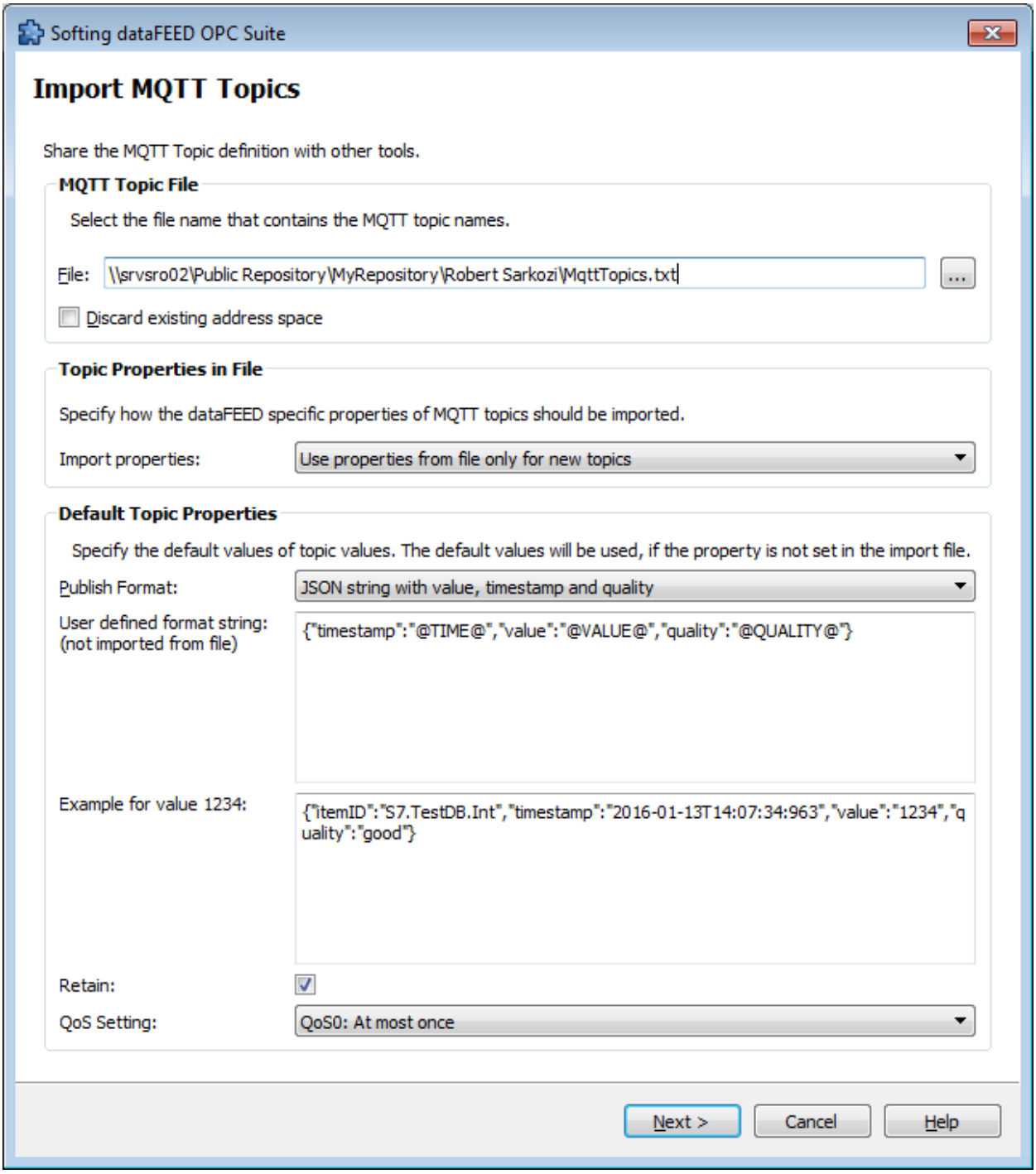
The Export File Selection wizard page allows you to choose a file name and location for the exported file. The file can be selected by manually typing its path and name in the *File* input box or by clicking the **[Browse]** button located to the right of the input box. A standard file saving dialog is opened.

When the **[Next]** button is pressed the file path is verified and if the path does not exist, an error message is displayed (invalid path). If the specified file path is valid and a file with the specified name already exists, a message box will ask whether to overwrite the existing file or not (select another file). The actual filter export operation begins if the path to the file is valid and either a file with the specified name does not exist or it exists and **Overwrite** has been chosen.

When the export operation is finished the **Export Results** page is displayed.

Export Wizard - Export Results >>>

Use this page to import topic from a file and to define their properties.



The file can be selected by manually typing its path and name in the **File** input box or by clicking the **[Browse]** button located to the right of the input box. A standard file selection dialog is opened. If an import file in XML format has been chosen, all topics of the current connection will be replaced by the topics from the import file, with their properties.

- The file path and name should be valid (they should point to an existing file).
- The file must list the topic names. Parent nodes that are required are created automatically. If the import file in text format contains for example the following line:

```
myCastle/northTower/floor1/room001
```


© Softing Industrial Automation GmbH

the following topics are created for the specified address space (if not already existing):

- myCastle
- myCastle/northTower
- myCastle/northTower/floor1
- myCastle/northTower/floor1/room001

The address space that already exists can be deleted alternatively. It is overwritten with the address space that has been imported.

In addition it is possible to enhance the existing address space with the newly imported one.

You can influence the topic properties of the topics to be imported by using a combo box. You have the following selection options:

1. Use properties from file only for new topics: For topics that are newly created during import, the property values specified in the import file are used. If for a specific topic in the import file no property is specified, default values are used (see below).

2. Ignore properties in file (use default values for new topics): The property values from the import file are ignored. Instead default values (see below) for newly created topics are used.

3. Overwrite existing properties with properties from file: For newly created topics and for already existing topics the properties specified in the import file are used. If for a specific topic in the import file no property is specified, default values are used (see below). Topics that are not existing in the import file are not modified.

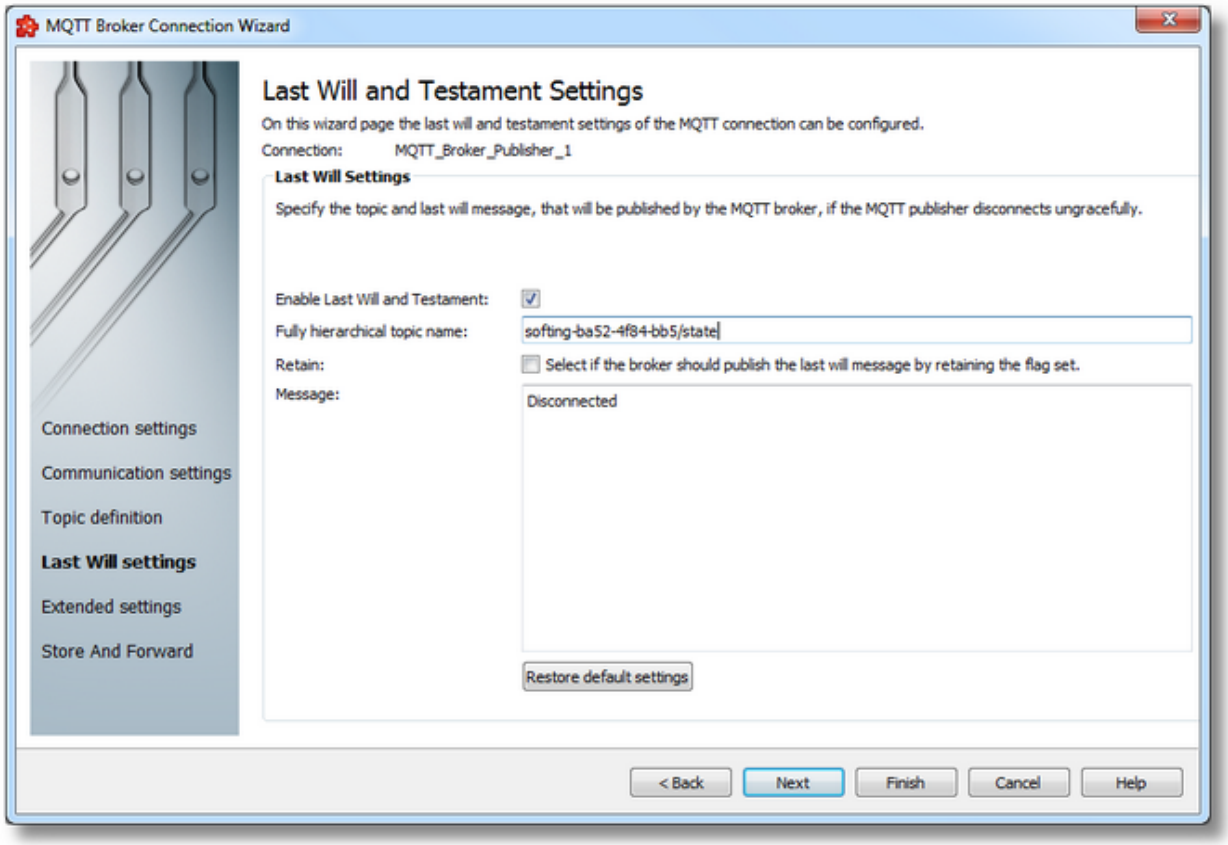
4. Overwrite existing properties with default properties: The property values from the import file are ignored. For newly created topics and for already existing topics the default properties are used (see below). Topics that are not existing in the import file are not modified.

In cases where dataFEED OPC Suite-specific properties are ignored or do not exist (partly) in the import file you can specify standard values for these properties. See also Add or modify MQTT Topic.

When you press the **[Next]** button, the file name and file path are verified. If the file identified by the specified file path and name does not exist, an error message is displayed.

Import Wizard - Import Results >>>

This page allows to perform the "Last Will and Testament" settings of the MQTT connection.



When establishing a connection, MQTT clients such as the dataFEED MQTT Publisher can transfer a topic name and a message.

If the Broker detects that the connection to the client does not exist any longer (e.g. because a network switch has failed), then the Broker publishes this message within the topic.

A possible scenario is that MQTT clients use for example the topic <client id>/state to inform other clients about their connection status. (The <client id> part is being replaced by the respective client ID) When establishing the connection, the clients make use of this topic as Last-Will-Topic and as "Disconnected" notification. After successfully establishing the connection they publish the notification "Connected" to this topic. If a client tries to disconnect, it publishes "Disconnected" as notification to this topic.

Using Last Will and Testament

This check box allows to specify whether the "Last Will and Testament" MQTT connection options are used or not. If they are used, additional configuration fields are available.

Complete name of the Last Will topic including all hierarchy levels.

© Softing Industrial Automation GmbH

The individual hierarchy levels are separated by the '/' character. They must be entered for the Last-Will topic. In order to use this topic in the Exchange module, it is added to the address space of the connection (see note below).

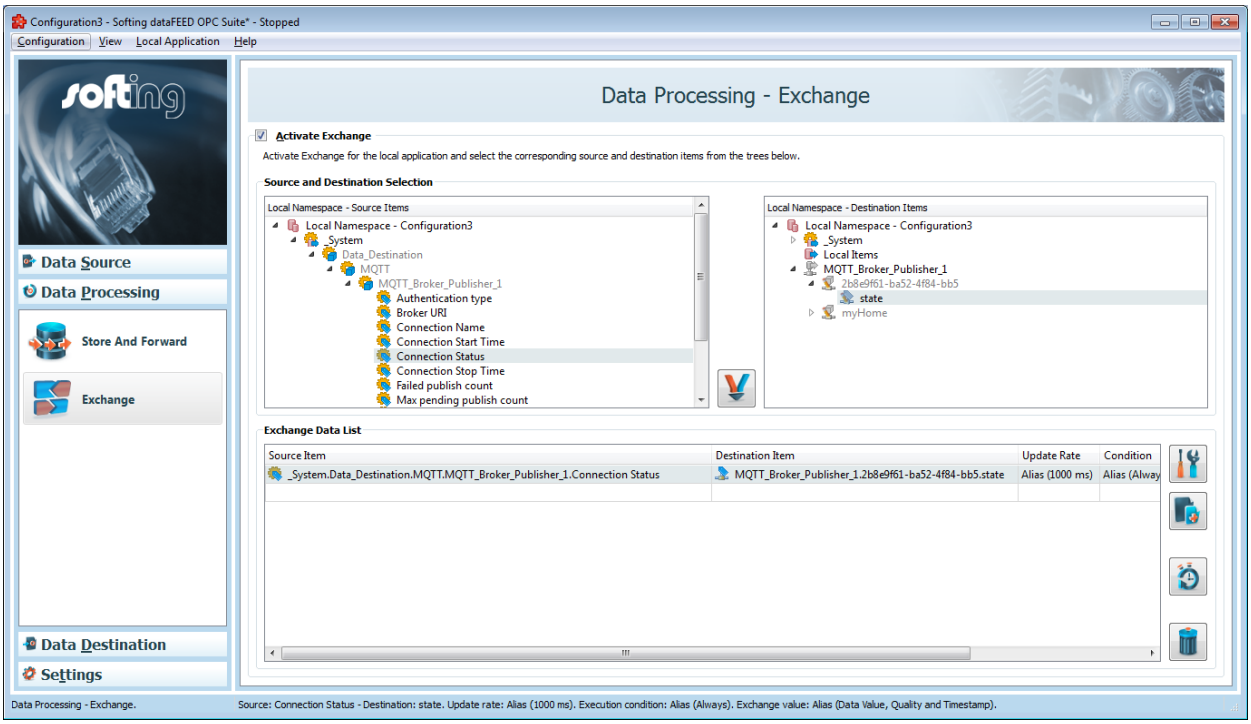
Retain

By setting the Retain flag the MQTT Broker is instructed to save the most recent data value for the topic. Depending on the Broker configuration it saves the data in RAM or persistently in the file system/database. Data values without Retain flag are only transferred from the MQTT Broker to those MQTT Subscribers that are registered in that moment by the corresponding broker and have subscribed the corresponding topic.

Message

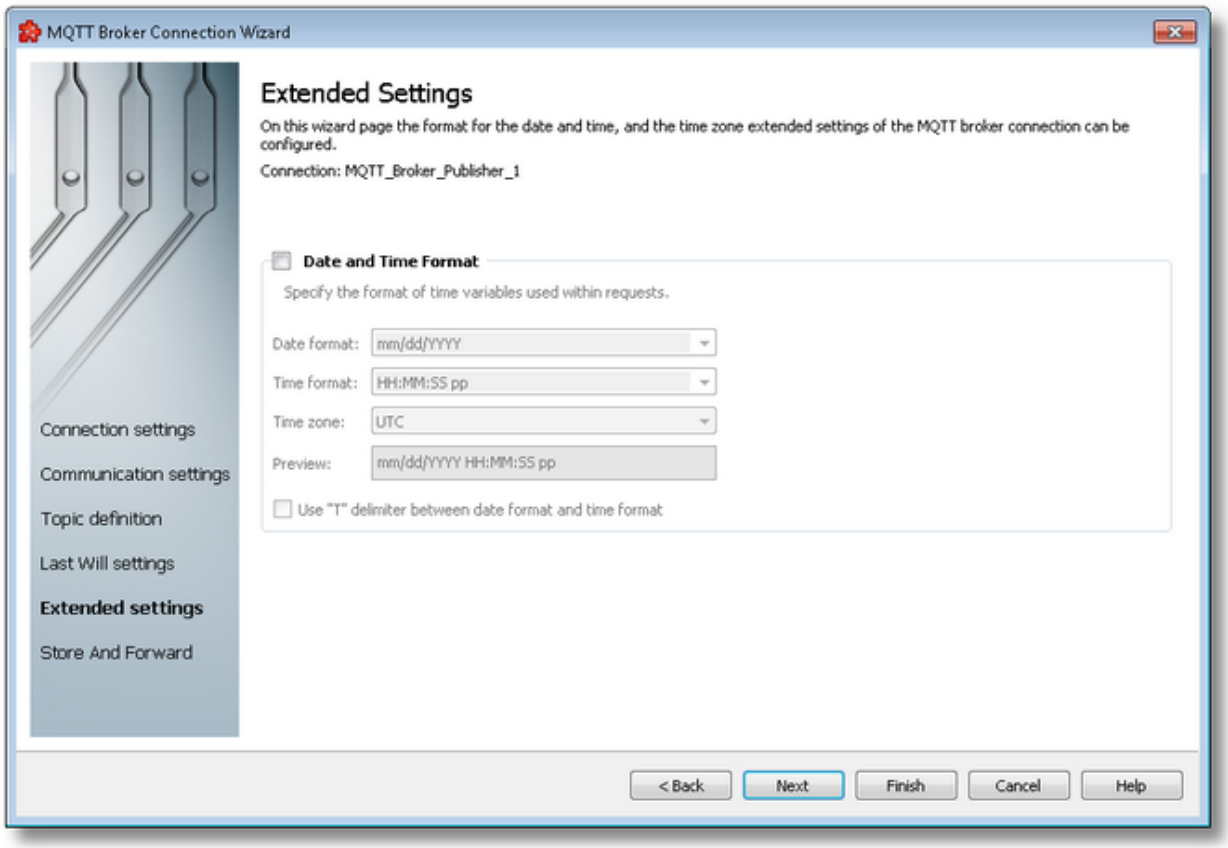
Enter the message into this filed that is to be published by the Broker as testament.

Note: You can use the Exchange module to publish the connection status following a successful connection establishment. To do so open the configuration of the exchange module after completing the connection wizard. Then select the "Connection Status" system item of the MQTT connection as data source and the last will topic as data destination as shown in the figure below:



<<< Connection Wizard - Topic Definition Connection Wizard - Extended Settings >>>

This page allows to specify the extended connection settings such as time and date format.



Date and Time Format

This section handles the time and date formats that are to be used when the timestamp placeholders are expanded and their values written to the MQTT requests.

The Date and Time format settings apply to item data values of data type date and time (DATETIME), and to item data values configured to the data value types: Timestamp (without milliseconds), Timestamp (with milliseconds), Internal Timestamp (without milliseconds), or Internal Timestamp (with milliseconds). For more details on the latter data value attributes, please see the description at Data Values Placeholder.

The default values are `mm/dd/YYYY` for the *Date format and* `HH:MM:SS pp` for the *Time format*. Other defined values for Time and Date format are:

- *Date format*: `mm/dd/YYYY`, `dd-mm-YYYY`, `dd.mm.YYYY`, `dd/mm/YYYY`, `YYYY-mm-dd`, `mm-dd-YYYY`
- *Time format*: `HH:MM:SS pp`, `HH:MM:SS.FFF pp`, `HH:MM:SS.FFF`, `HH:MM:SS`

Note: Item data values of data type date and time (DATETIME) that have a time format setting containing the milliseconds also (`FFF`), will display zero milliseconds value.

© Softing Industrial Automation GmbH

Note: Values for Time and Date format can also be defined manually but in this case special care must be taken to ensure that the definitions are recognized by the local dataFEED OPC Suite application.

In addition to the default values listed in the format selection combo boxes, the user can add his/her own custom format. Custom Date and Time format strings are accepted as valid, when the format strings contain in their composition the following accepted format substrings and format delimiters. These formats allow the inclusion of specific locale information for the regional settings.

- *Valid Date format strings*: `AAAAAA, aaa, BBBBBBBB, bbb, ccccccccccccccccccc, dd, jjj, mm, UU, ww, xxxxxxxxxx, yy, YYYY`
- *Valid Time format strings*: `II, HH, MM, SS, pp, XXXXXXXX, FFFFFF, FFFFF, FFFF, FFF ,FF, F`
- *Valid Date and Time format delimiters*: `<dot>, <colon>, <slash>, <comma>, <backslash>, <vertical bar>, <semicolon>, <space>, <tab>, <minus>`

Time Zone handles the source localization time: UTC (Universal Time Coordinated) or local time (local time localization). The default time zone is UTC.

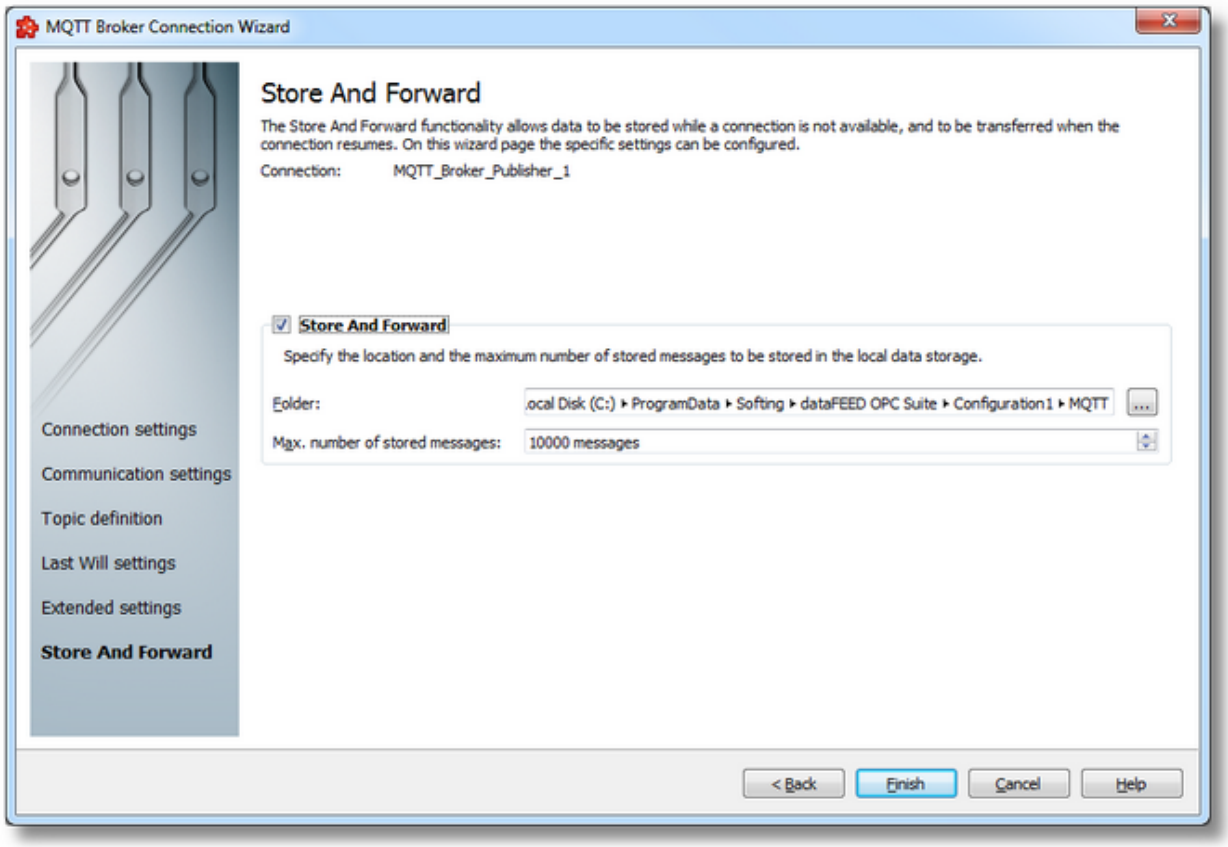
The option *Use "T" delimiter between date format and time format* allows to obtain custom formats such as `YYYY-MM-DDTHH:MM:SS.FFFTZD`, by inserting the "T" delimiter between the selected date and time format values, and appending the `TZD` time zone designator expression.

For UTC time zone selected the `TZD` has the value `Z` (`YYYY-MM-DDTHH:MM:SS.FFFZ`), and the for Local time zone selected has the value `+hh:mm` indicating a time zone offset ahead of UTC (`YYYY-MM-DDTHH:MM:SS.FFF+hh:mm`), or `-hh:mm` indicating a time zone offset behind UTC (`YYYY-MM-DDTHH:MM:SS.FFF-hh:mm`).

When the *Time, Date Format and Time Zone* option is not selected the default values are used. If they are modified, they apply to all actions of the currently selected MQTT broker connection.

<<< Connection Wizard - Last Will Settings Connection Wizard - Store And Forward >>>

This page allows to perform the "Store and Forward" settings of the MQTT connection in case that the connection to the broker is unavailable.



This functionality can be turned on and off by using the Store And Forward checkbox. If it is off (default), then dataFEED will not persist the publisher messages that are not delivered to the broker when the connection is interrupted.

Note: The Store and Forward functionality is active only if ***clean session*** flag is *false*. When the ***clean session*** flag is set to *true*, the client does not want a persistent session. If the client disconnects for any reason, all information and messages that are queued from a previous persistent session are lost.

Folder

The path where the storage files should be located. If the path does not point to an existing folder then the corresponding path will be created by the dataFEED OPC Suite Configurator when the Next button is pressed, and user confirms.

Maximum number of stored messages

Maximum number of messages stored used to limit the messages saved in the storage directory. Default is 10000.

© Softing Industrial Automation GmbH

<<< Connection Wizard - Extended Settings

7.3.3.4.3 Connect to Azure IoT Hub via SSL and WSS

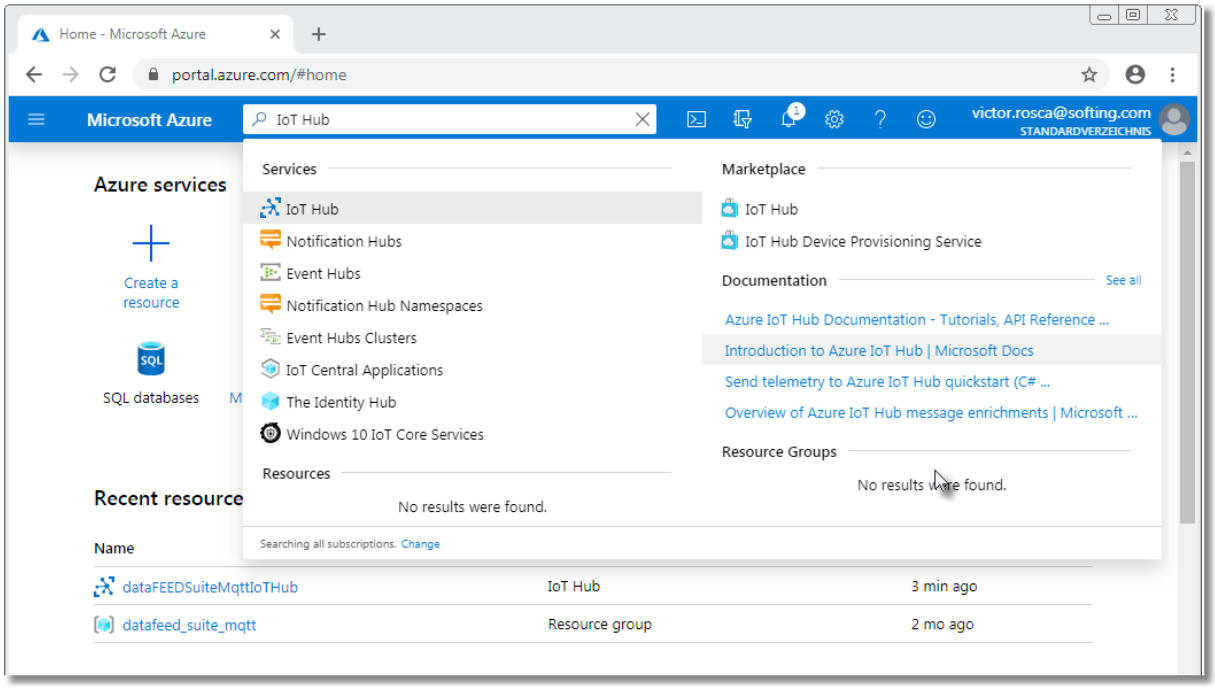
The dataFEED OPC Suite application supports connections with *Azure IoT Hub*. Devices can communicate with *IoT Hub in Azure* using *SSL* and *WSS* protocols, *SSL* via port *8883* and *WSS* via port *443*. In order to connect to *Azure IoT Hub* from *Data Destination / MQTT Broker*, please follow the next steps.

Download *Device Explorer*

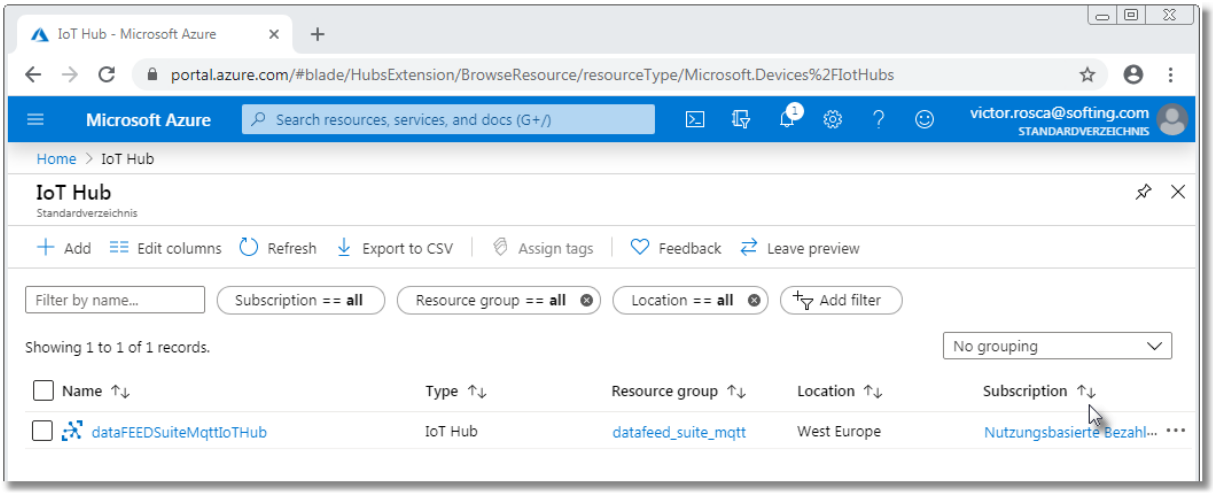
- Download the latest *Device Explorer* version form: https://github.com/Azure/azure-iot-explorer/releases

Connect *Device Explorer* to *Azure IoT Hub*

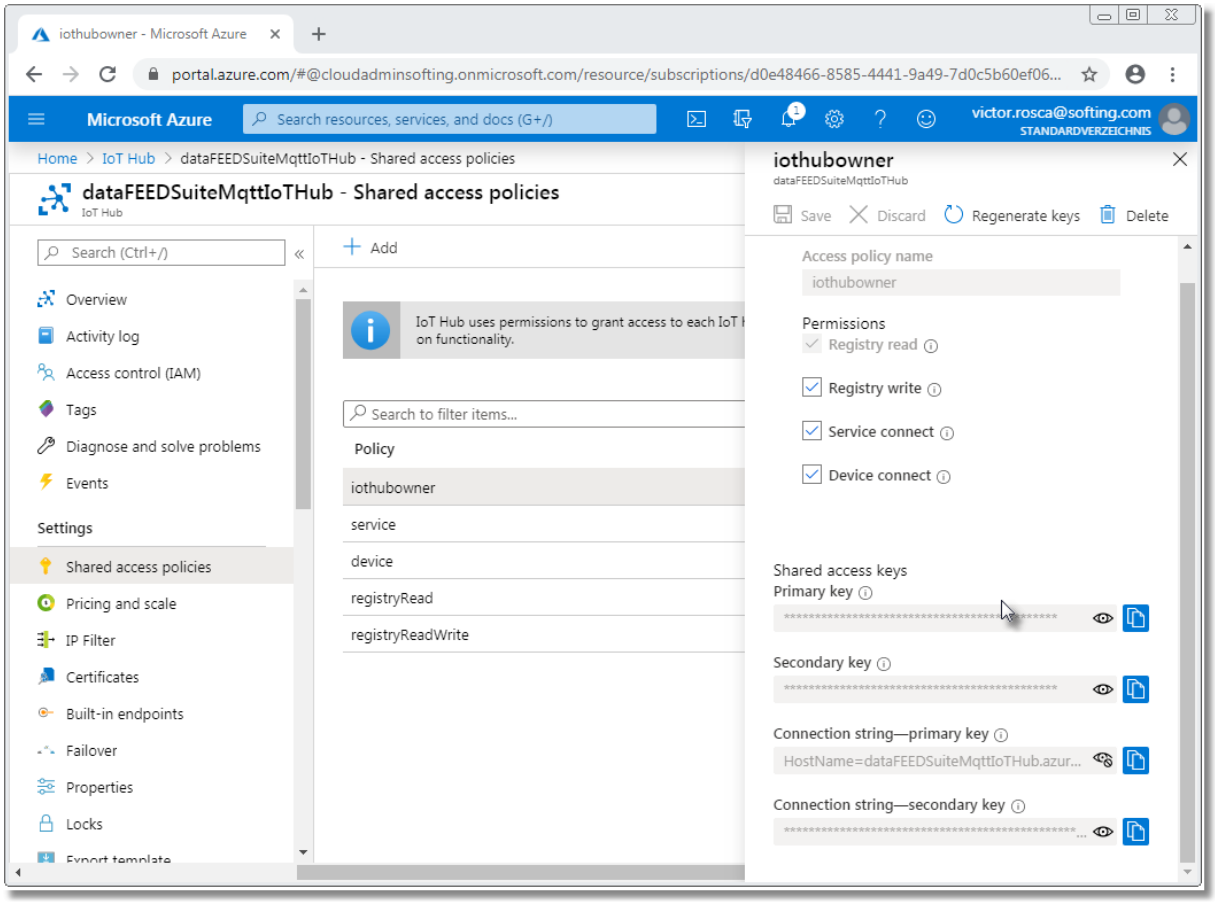
- In *Microsoft Azure Portal* search for *IoT Hub*



- Select *IoT Hub* to be connected or create a new one

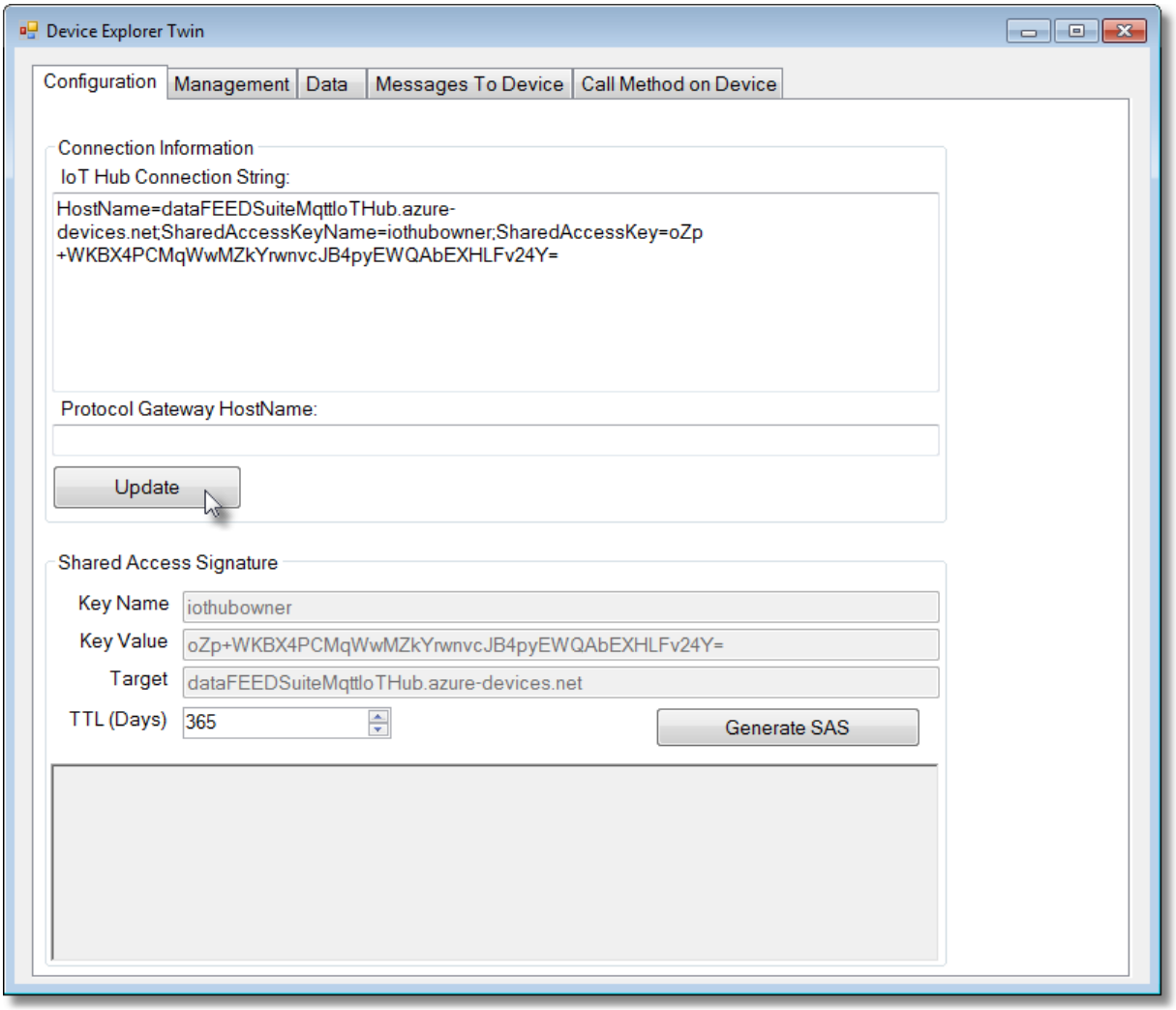


- Select *Shared access policies* Settings



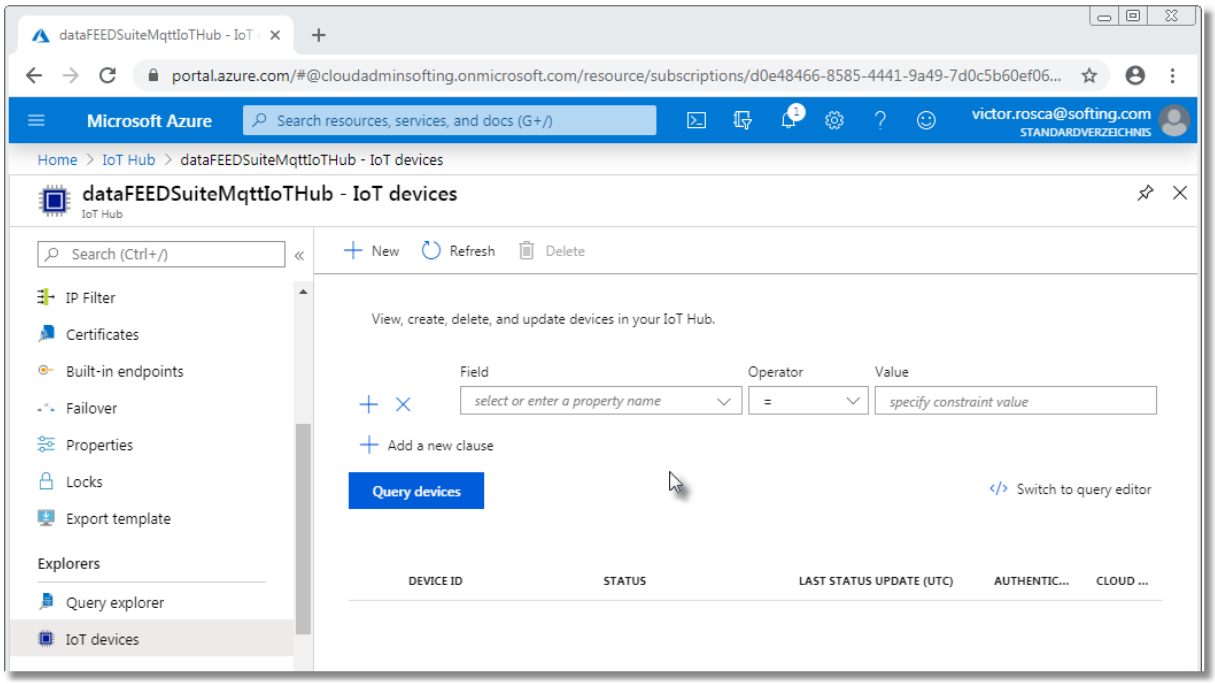
- Select ***iothubowner*** *Policy and* copy *Connection string - primary key*
- In *Device Explorer*, paste connection string to *IoT Hub Connection String* field and press *Update* button

© Softing Industrial Automation GmbH

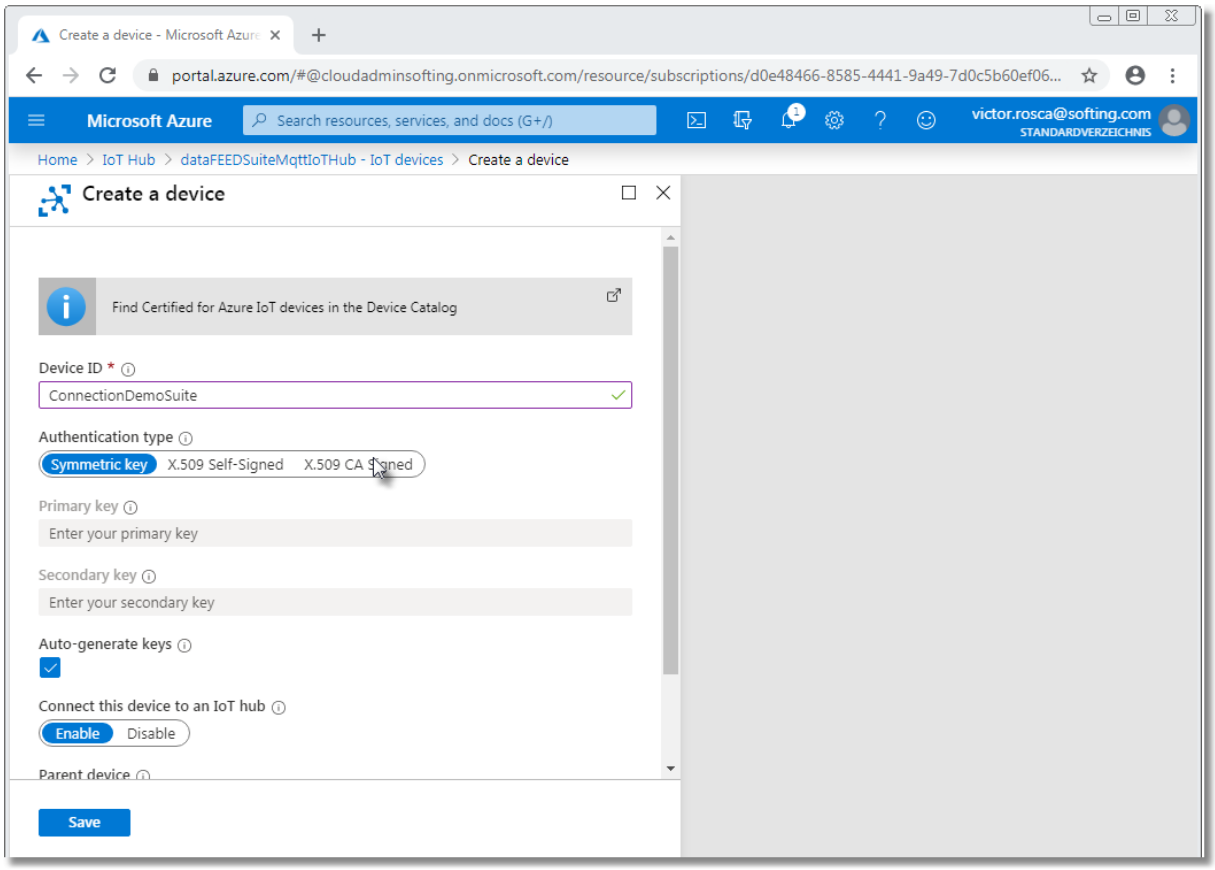


Create MQTT Connection in Azure IoT Hub

- In *Azure Iot Hub* select *IoT devices* and *Add* a new device

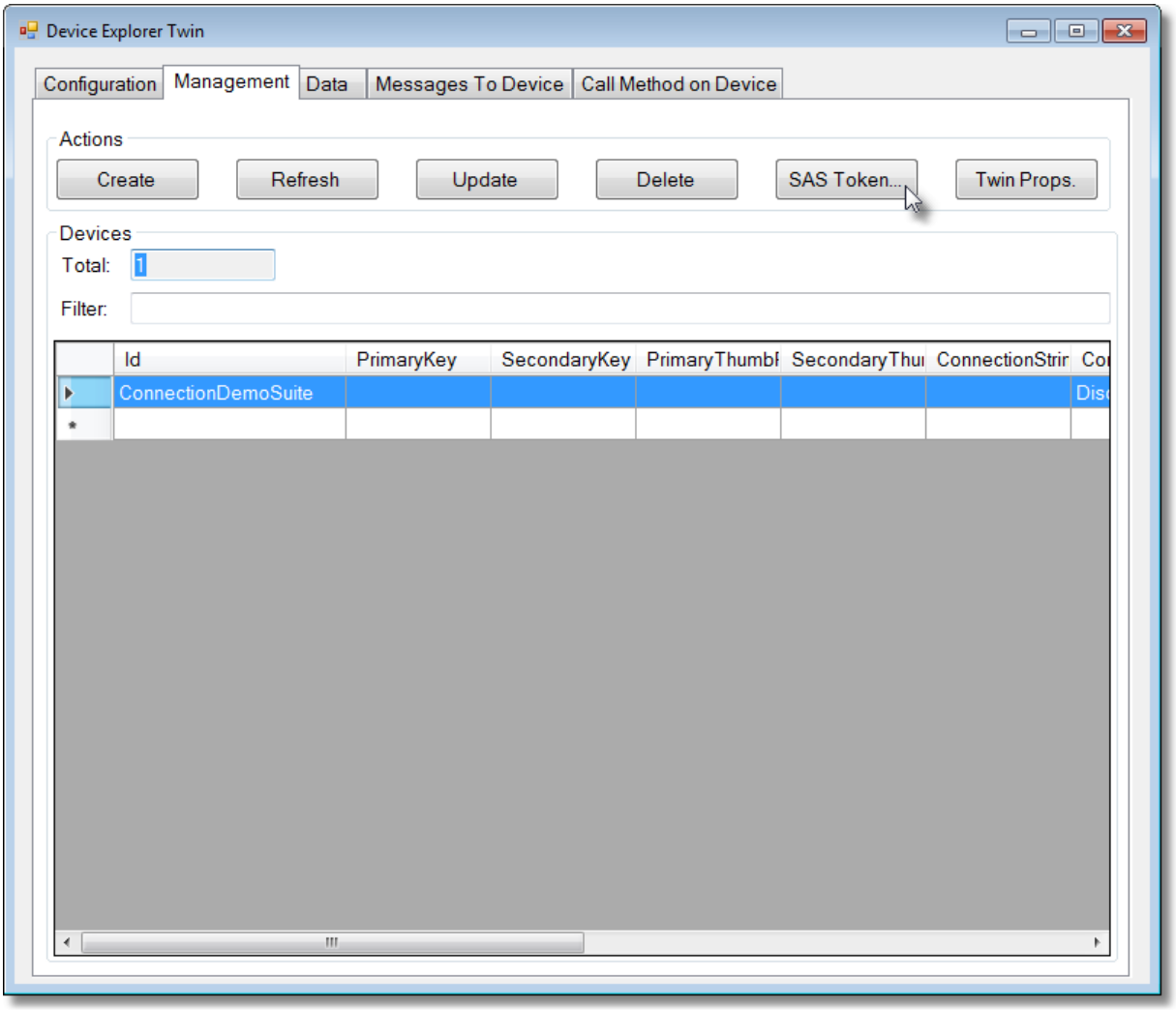


- Define unique *Device ID* (23 characters maximum)
- Do not modify default settings for rest of the fields and press *Save* button

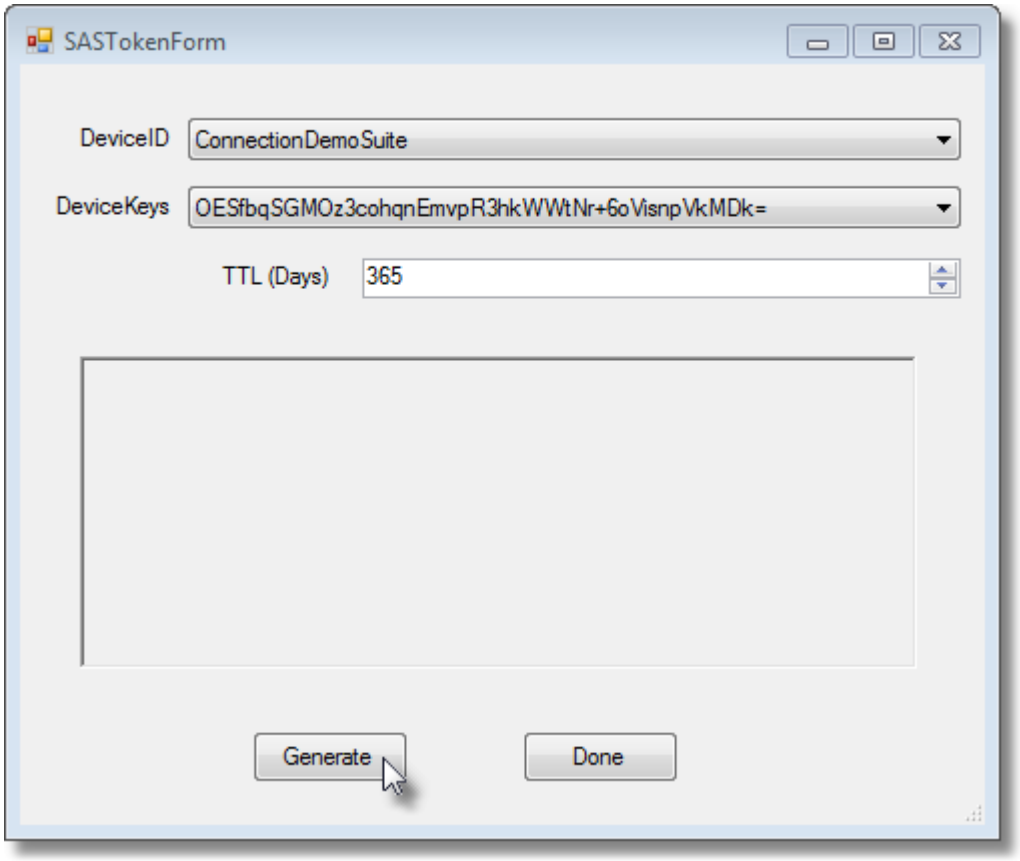


- The added device will appear in *Device Explorer*

© Softing Industrial Automation GmbH



- Select *Management* tab
- Select *IoT Hub* device
- Press *SAS Token...* button

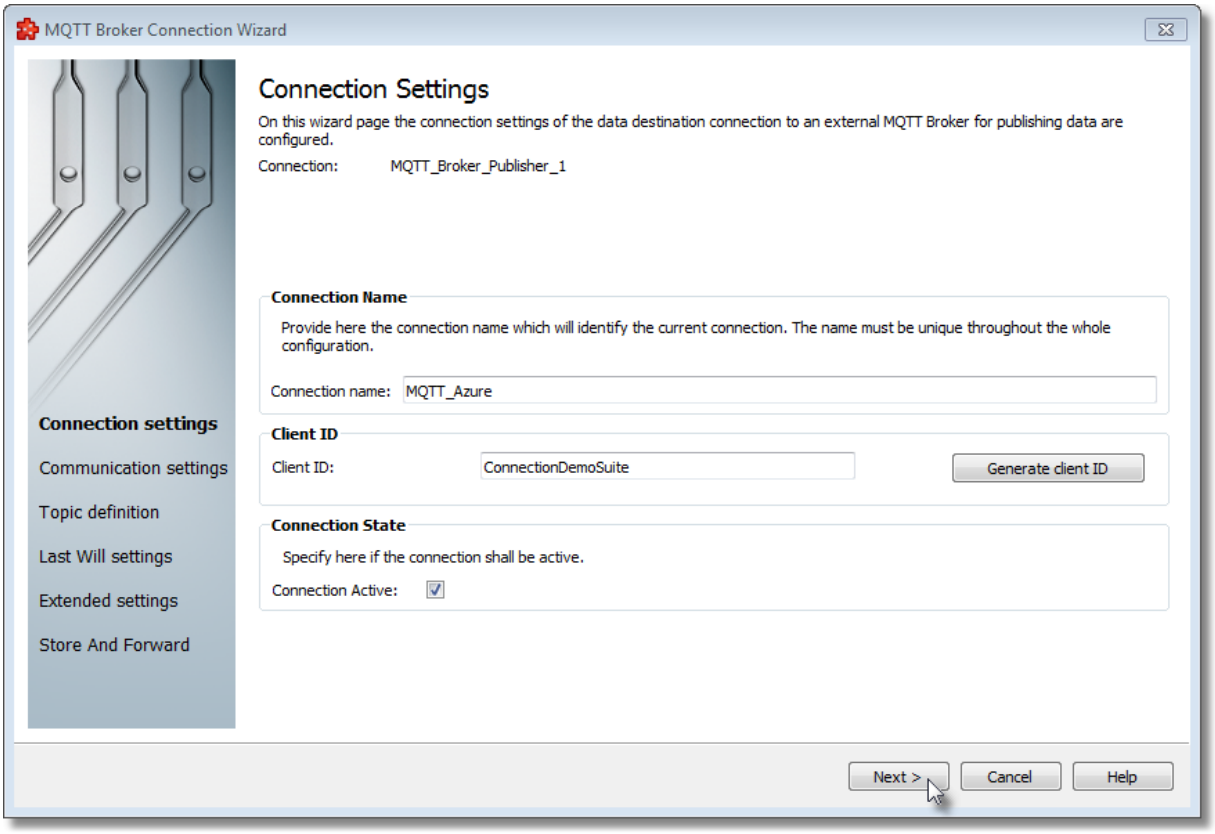


- Define SAS duration in TTL (Days) (e.g. to 365 days)
- Press Generate button
- Copy generated *SAS token* string for later use

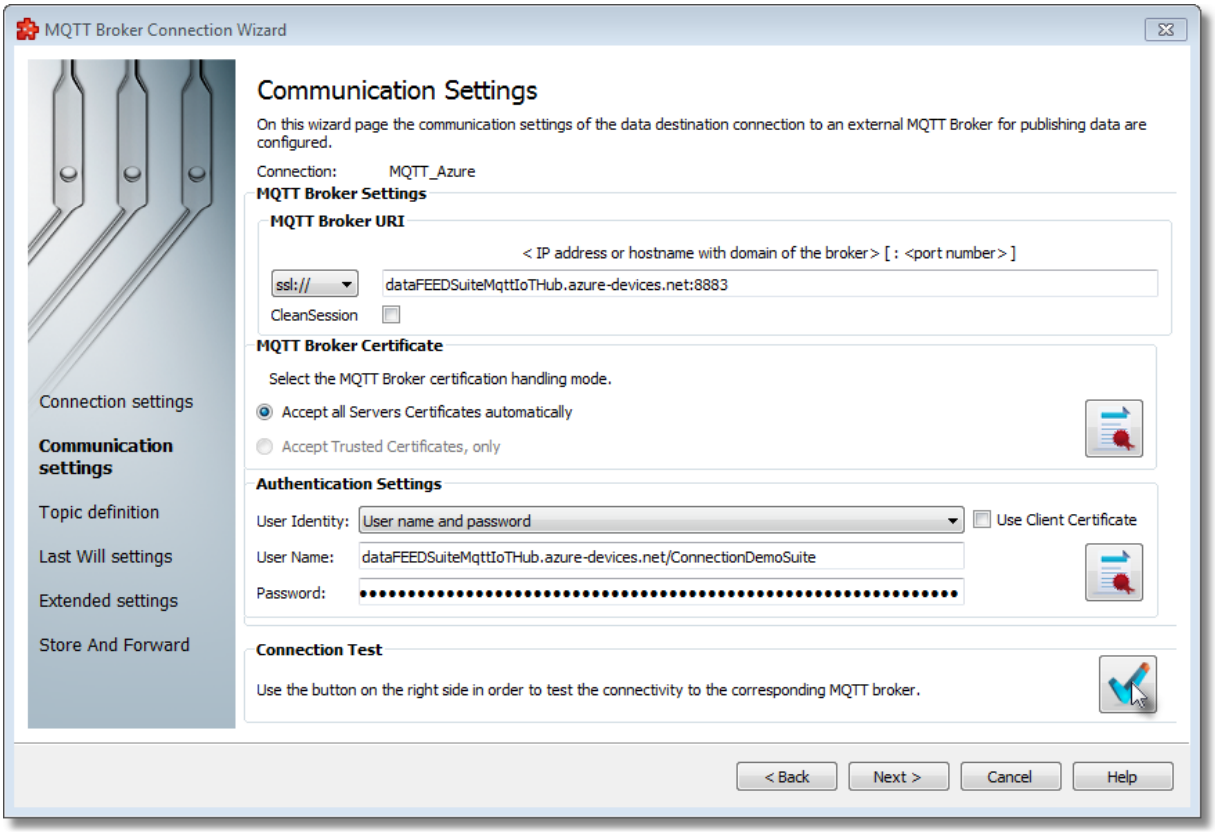
Connect *dataFEED OPC Suite* to *Azure IoT Hub*

- Start *dataFEED OPC Suite* configuration
- Select *Data Destination / MQTT Broker*

© Softing Industrial Automation GmbH

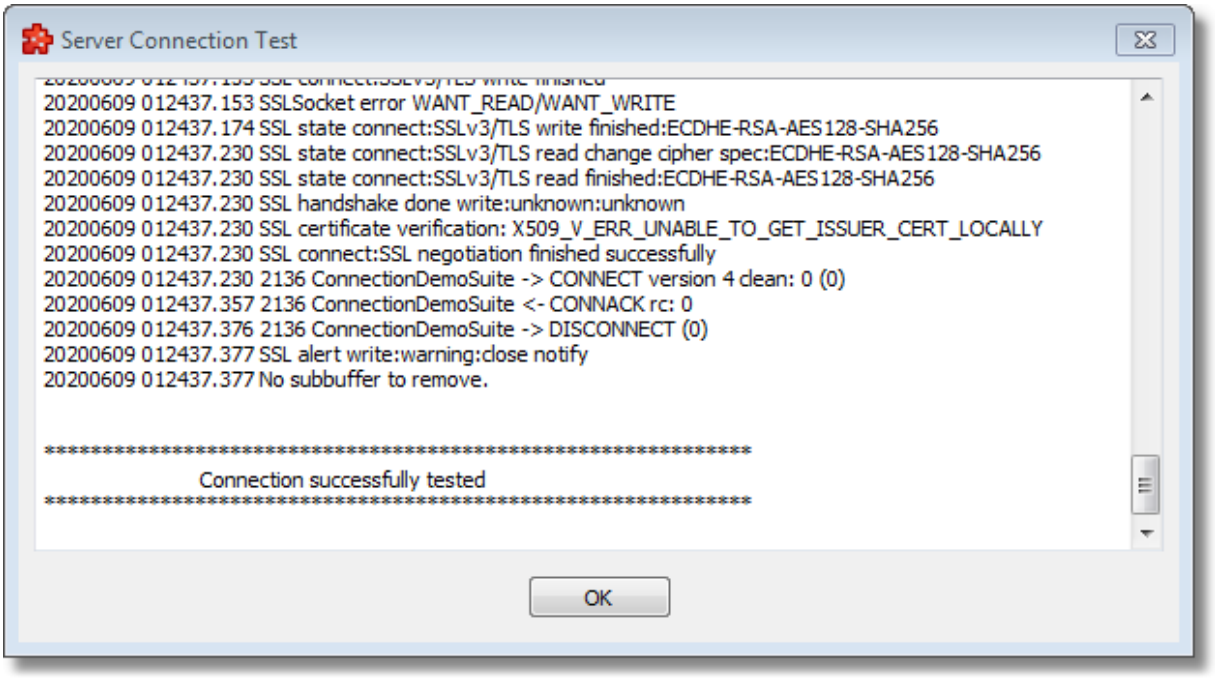


- Enter connection name in *Connection Name* field
- Enter device ID of created *Azure IoT Hub* device in *Client ID* field
- Press *Next* button



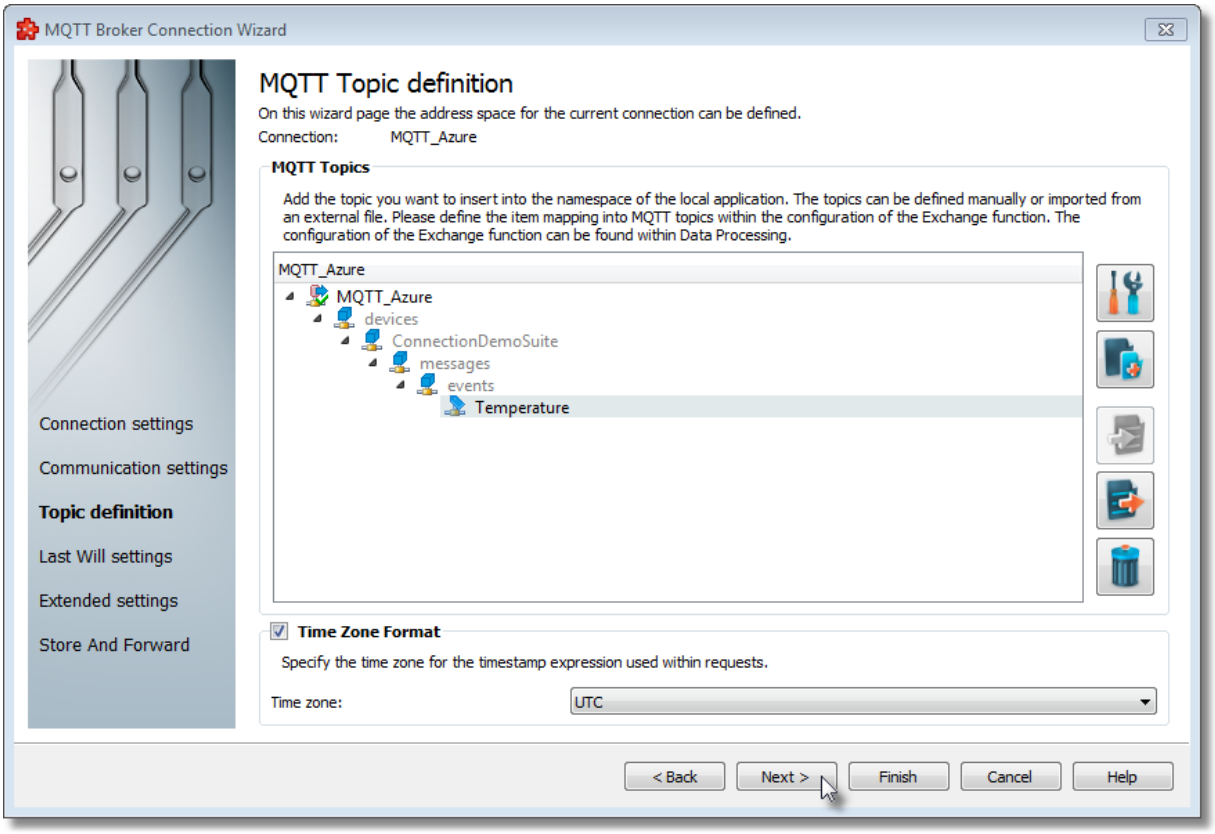
- Set *MQTT Broker URI* to *ssl://<IoT Hub Name>.azure-devices.net:8883*
- Set *User Identity* in *Authentication Settings* to *User name and password*
- Set *User Name* in *Authentication Settings* to *<IoT Hub Name>.azure-devices.net/<DeviceID*
- Set *Password* in *Authentication Settings* to part of generated *SAS Token* after *SharedAccessSignature=string* (e.g. string beginning with *SharedAccessSignature sr=Softing-IoT-Hub-SalesDemo.azure-devices.net*)
- Press Connection Test button to verify connection

© Softing Industrial Automation GmbH



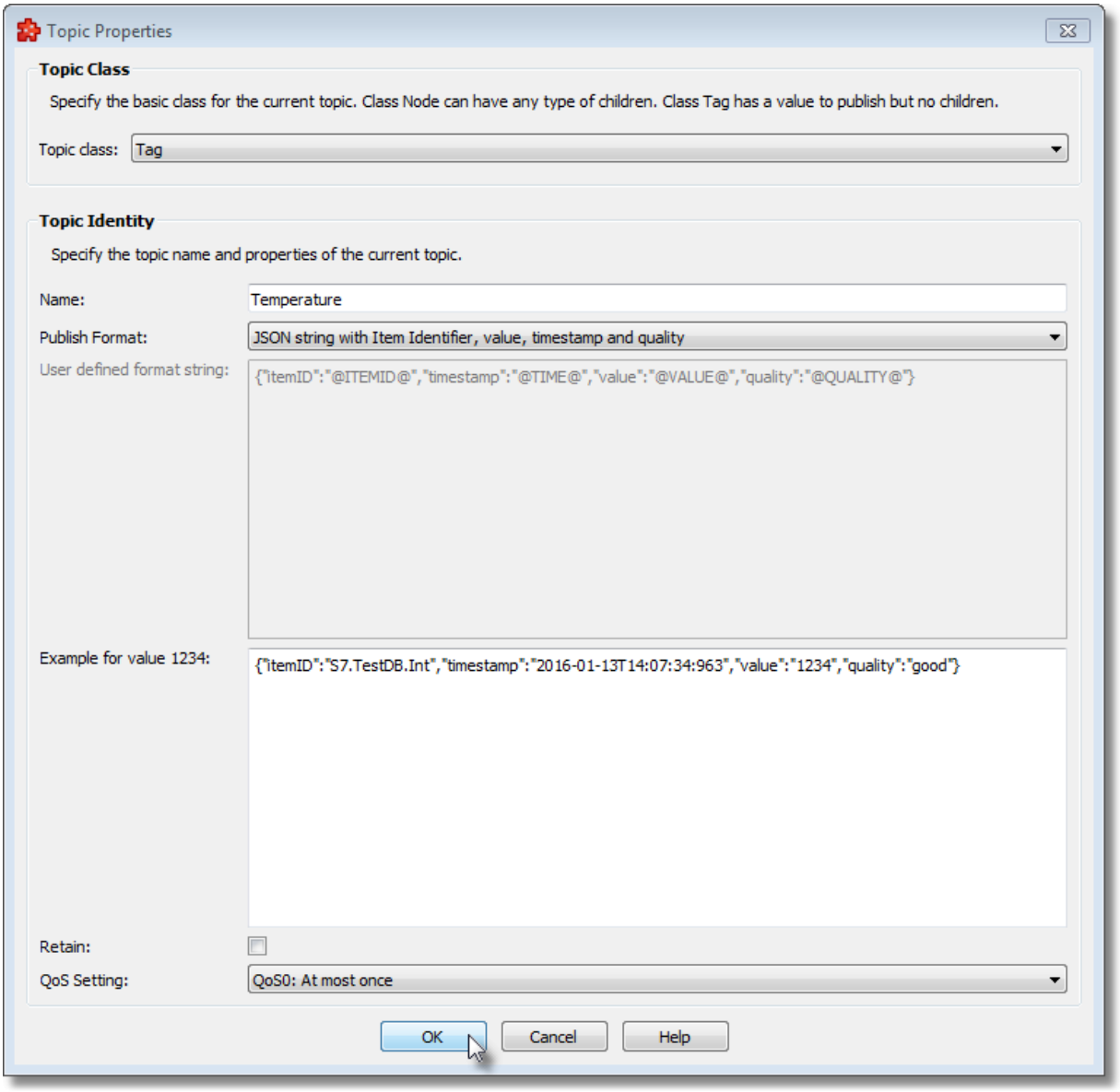
Note: In order to perform a connection via *WSS* the *MQTT Broker URI* should be: ***<IoT Hub>.azure-devices.net:443/$iothub/websocket***
E.g.: *dataFEEDSuiteMqttIoTHub.azure-devices.net:443/$iothub/websocket*

- Press *Next* button
- Define *devices / <Device ID> / messages / events* hierarchy
- Add topics for data to be sent below:

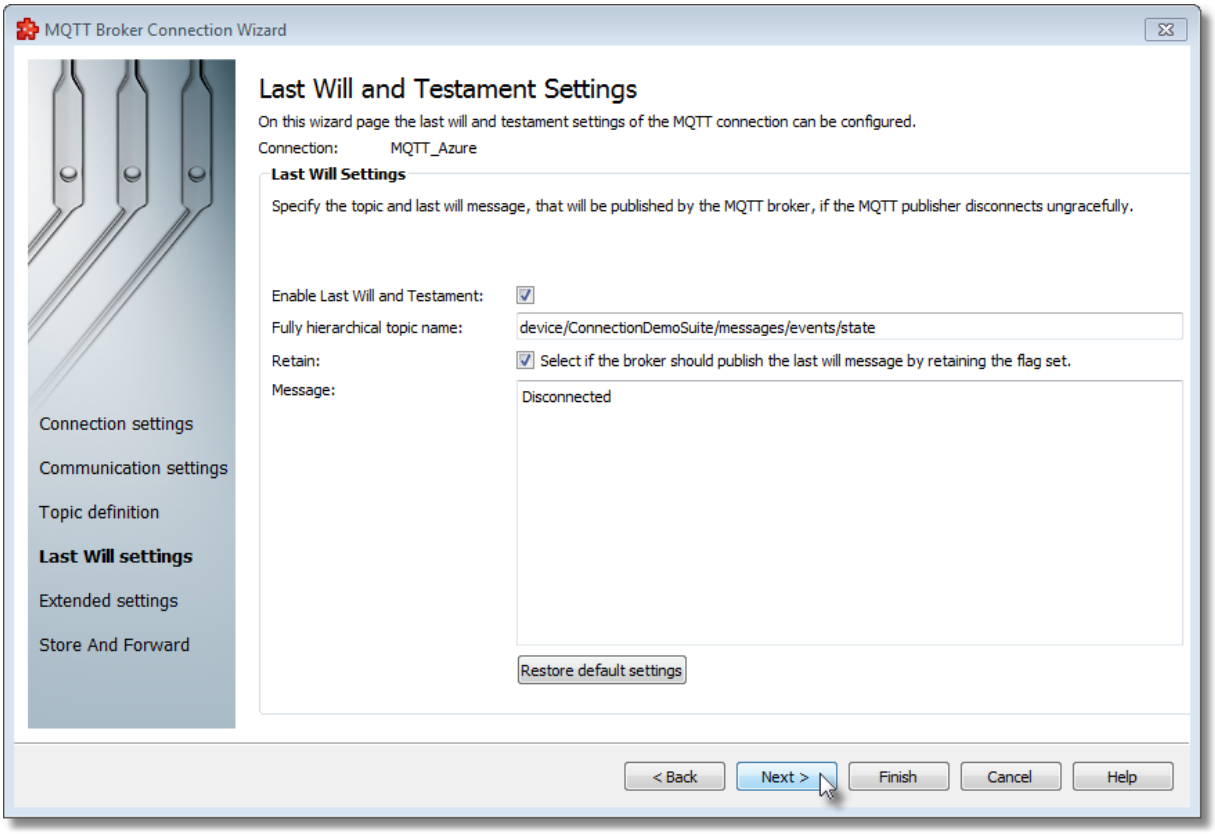


- Select topic and press *Properties* button

© Softing Industrial Automation GmbH



- Define Publish Format
- Press OK button
- Press Next button at MQTT Topic definition page
- Select Enable Last Will and Testament checkbox, if required

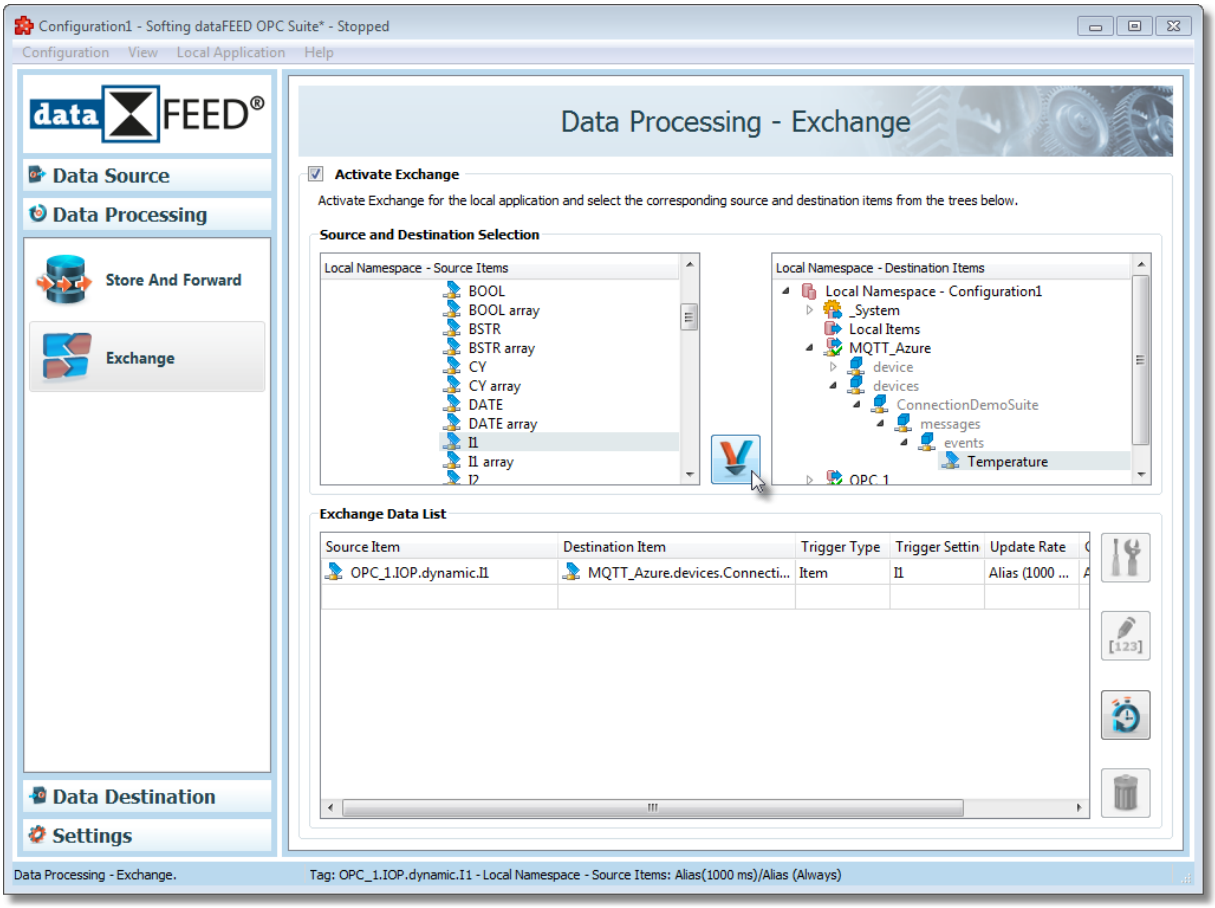


- Enter devices/<Device ID>/messages/events/state in Fully hierarchical topic name field
- Select Retain checkbox
- Leave Disconnected in Message field
- Press Finish button

Configure Data Exchange

- Select Data Processing / Exchange in dataFEED OPC Suite configuration

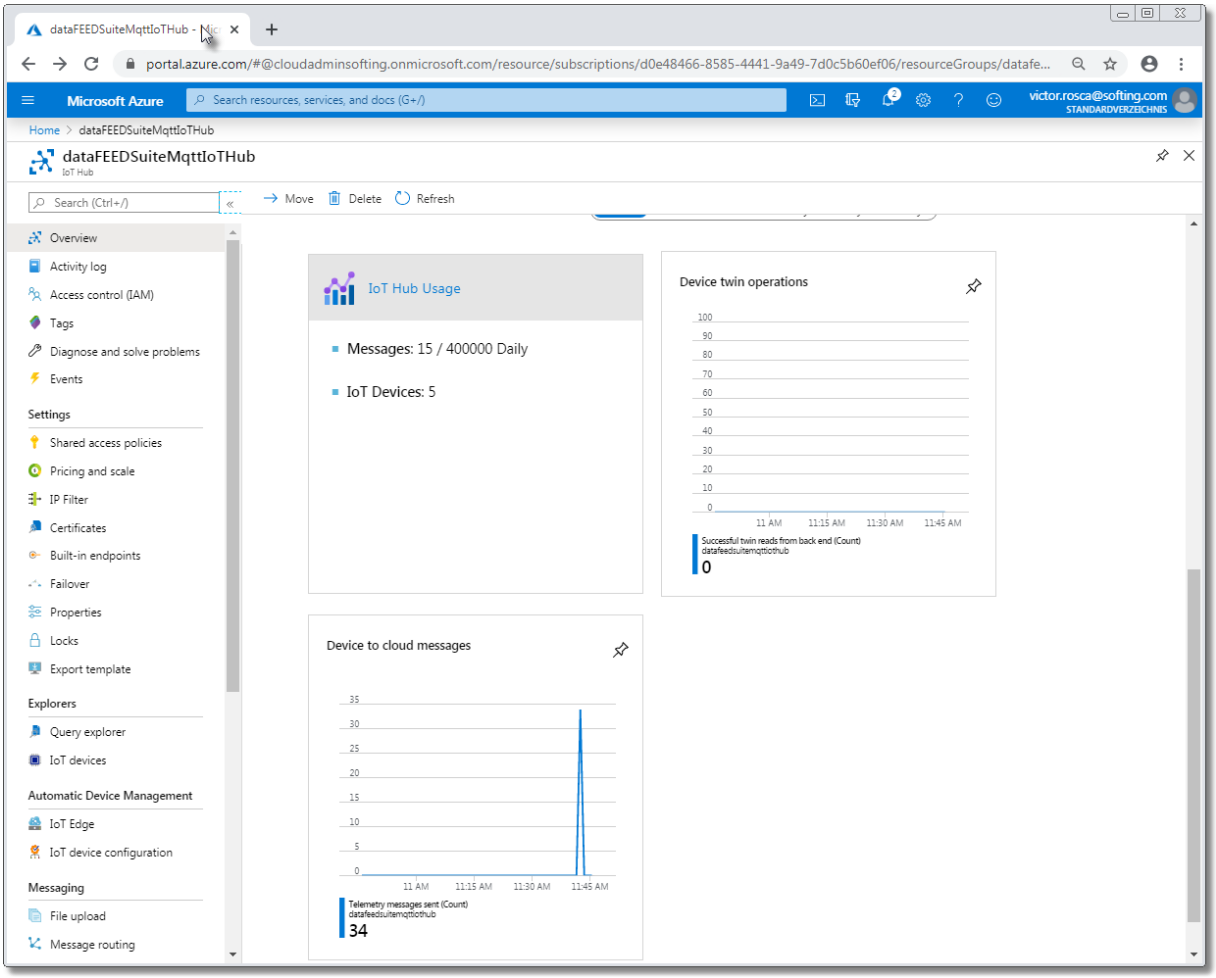
© Softing Industrial Automation GmbH



- Select individual data source item to be sent in Local Namespace - Source Items section
- Select individual MQTT item to be used for data exchange in Local Namespace - Destination Items section
- Click Create Exchange Action button for each individual data exchange item pair

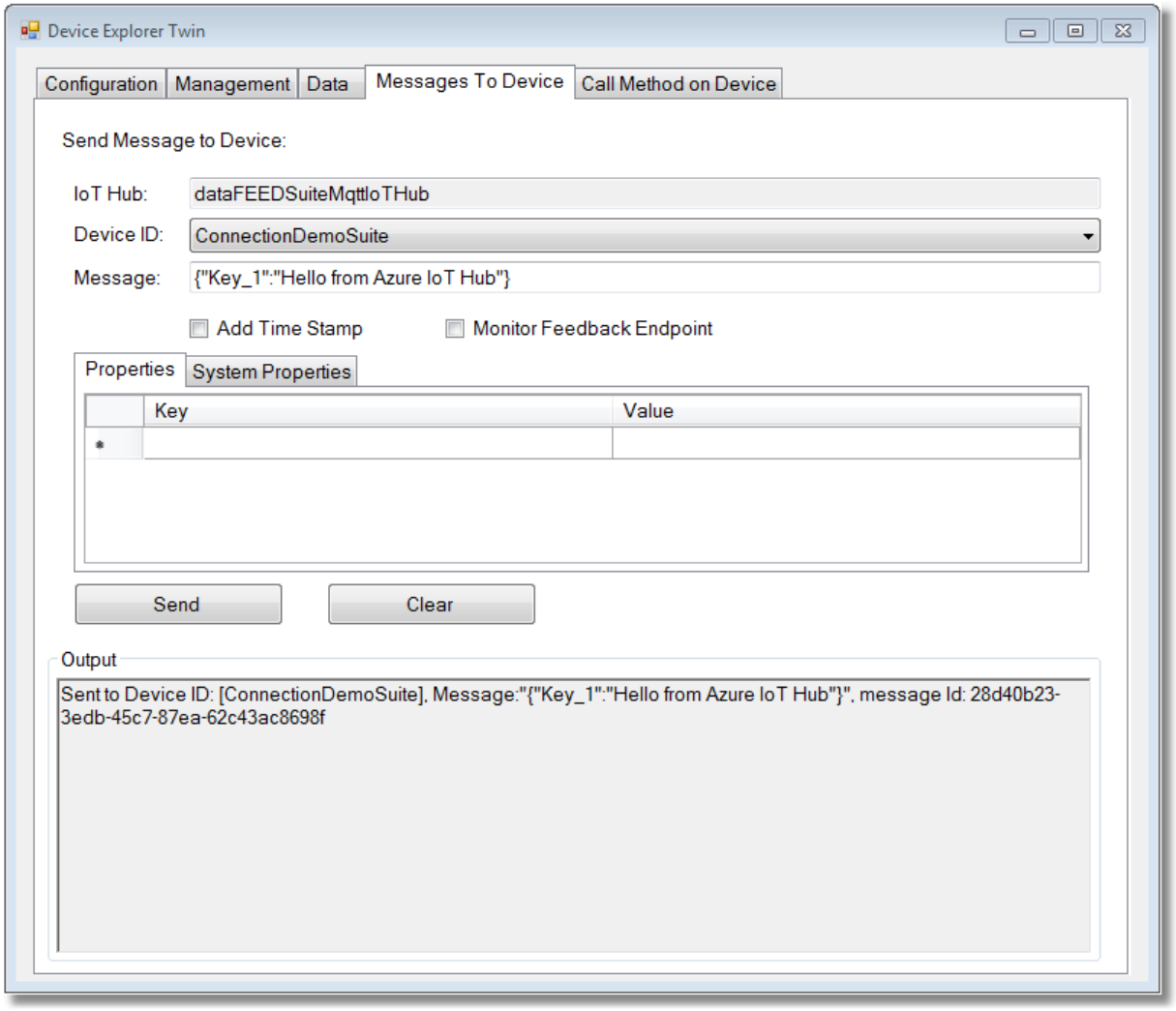
Test MQTT Connection and Data Exchange

- Start dataFEED OPC Suite application
- MQTT messages then are sent to Azure IoT Hub
- Select Overview page



- In Device Explorer user can monitor the received data after connecting to *IoT Hub*

© Softing Industrial Automation GmbH



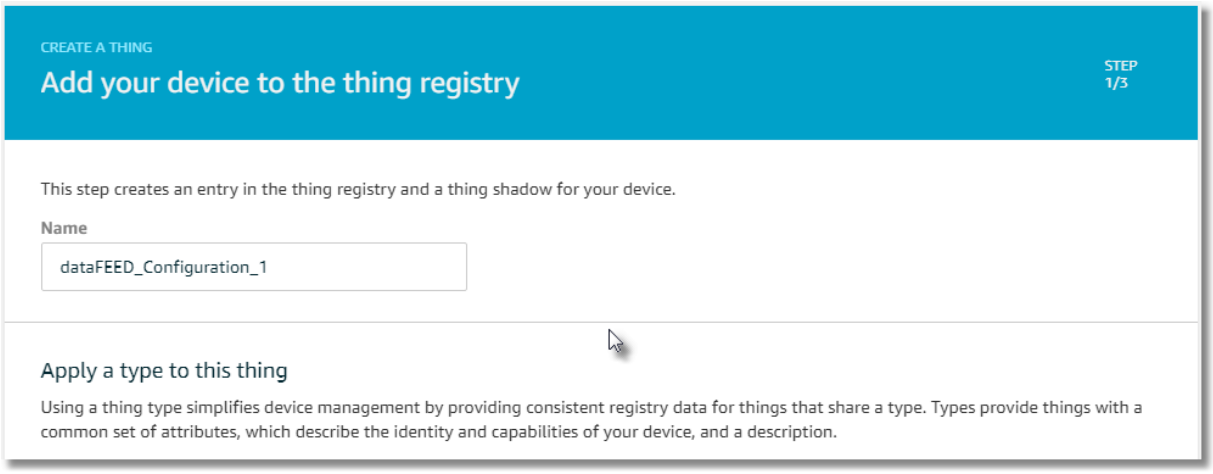
7.3.3.4.4 Connect to AWS IoT Device via SSL

The dataFEED OPC Suite application supports connections with *AWS IoT Device*. This topic describes how to configure dataFEED OPC Suite as *AWS IoT* device (*"Thing"*).

From the *AWS IoT* point of view dataFEED OPC Suite is referred to in this manual as *gateway*. It is based on the the *AWS Developer Guide*, from: https://docs.aws.amazon.com/iot/latest/developerguide/what-is-aws-iot.html.

AWS Preparation Steps

- Register Gateway as Device in the Registry. Follow the instructions given at: https://docs.aws.amazon.com/iot/latest/developerguide/register-device.html.
- In step 4. define unique gateway name



Create and Activate Certificate for Gateway

- Follow the instructions given at https://docs.aws.amazon.com/iot/latest/developerguide/create-device-certificate.html.
- In step 2. download certificate and key files. These files need to be uploaded as *MQTT Client* certificate into the gateway in a later step.

Create AWS IoT Policy for MQTT Client Certificate

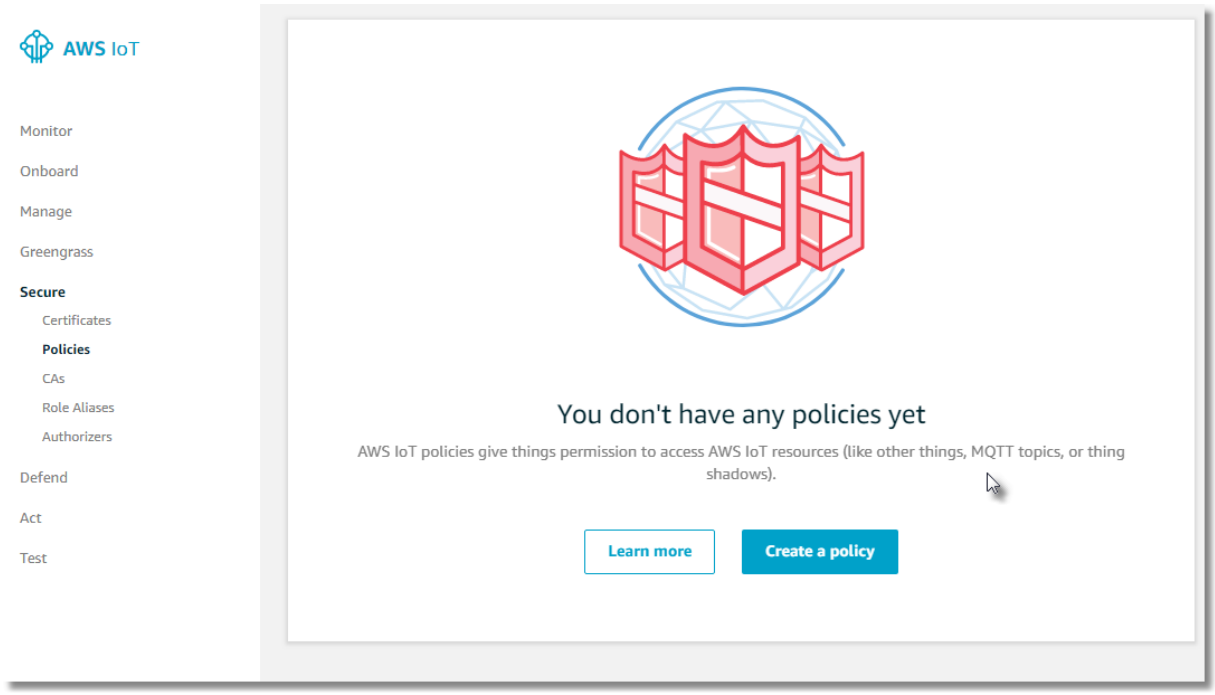
- Follow the instructions given at https://docs.aws.amazon.com/iot/latest/developerguide/create-iot-policy.html.
- Define gateway permission for ***iot:Connect*** and ***iot:Publish*** actions.

NOTE: For some unknown reasons the attachment of policies as described in step 3. does not work. Instead, it looks like the policy must be created before attaching it to a certificate. However, the certificate dialog does not trigger the creation of a policy.

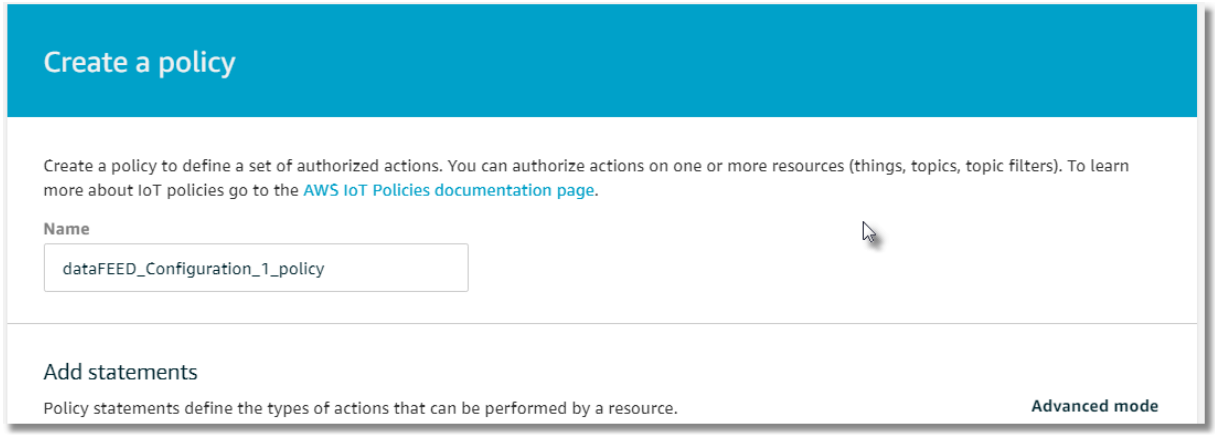
It is possible to create and attach a policy after the creation of a certificate by following these steps:

- Return to *AWS IoT Console* main screen and navigate to *Secure/Policies*

© Softing Industrial Automation GmbH

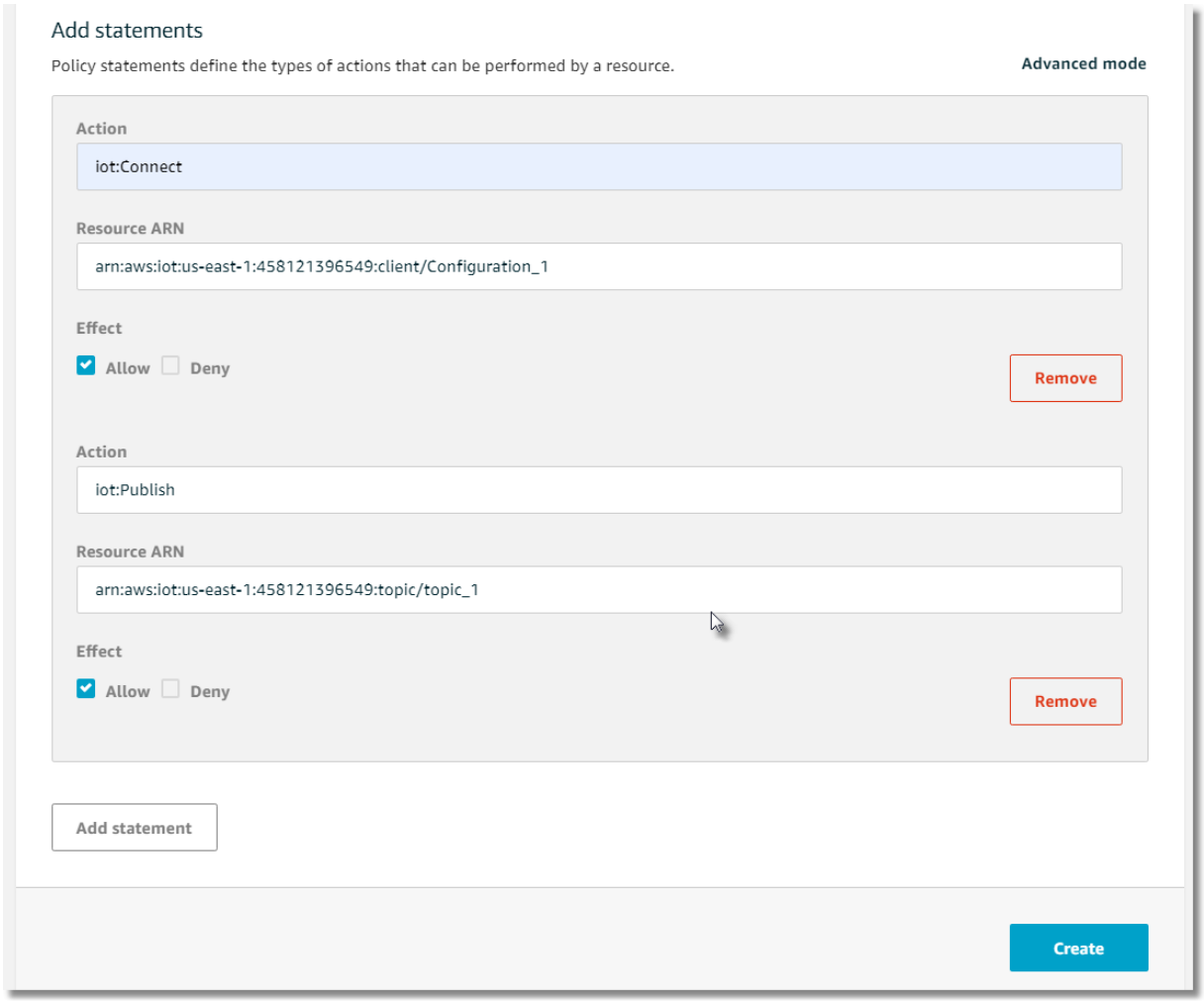


- Press *Create* a policy button
- Define policy name



- Define *gateway* permission for ***iot:Connect*** action
- Press *Add Statement* button
- Define *gateway* permission for ***iot:Publish*** action
- Define user-specific *ARN* topic. Remember this topic for a later step.

NOTE: The placeholder "*" allows publishing to any topic.

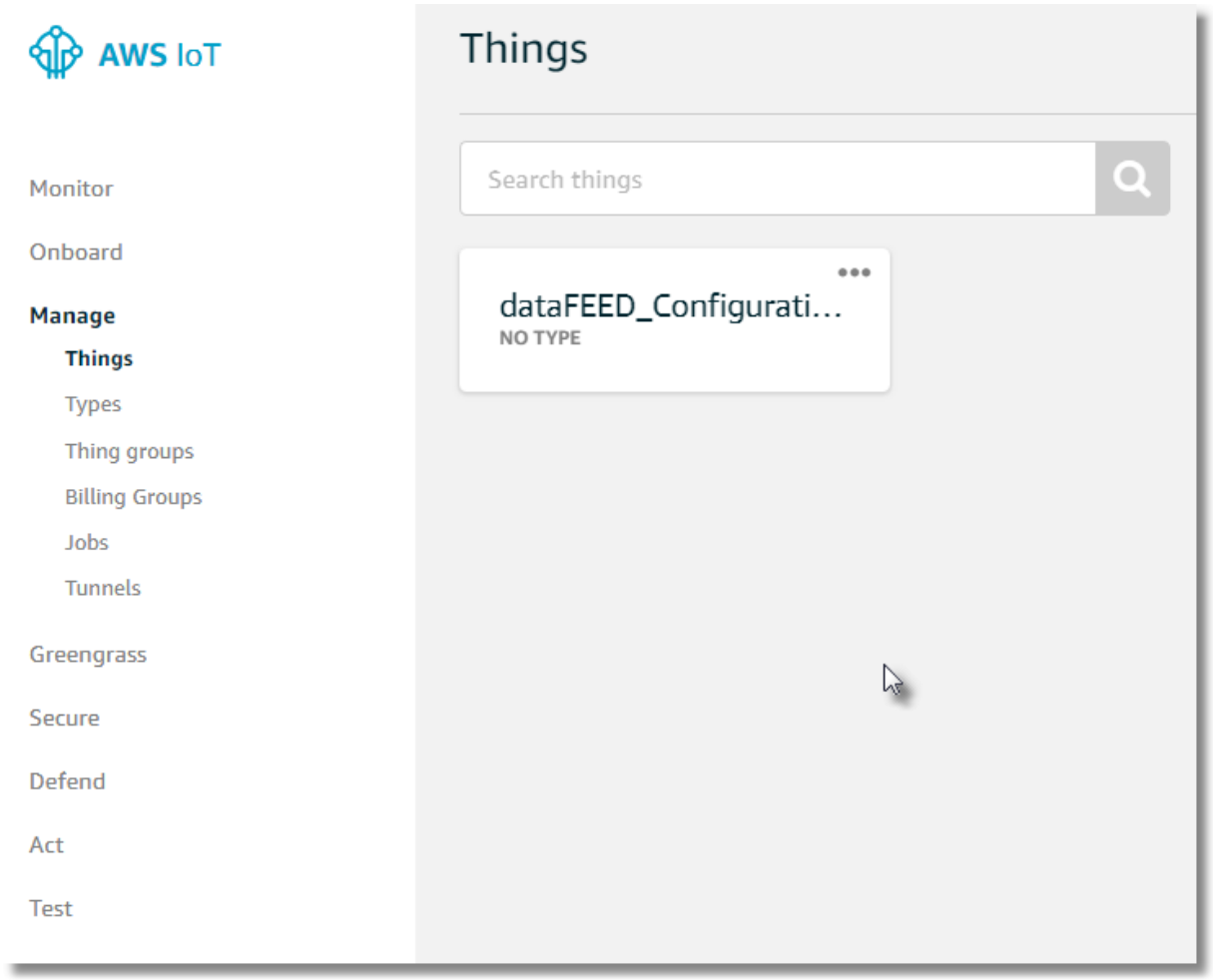


- Press *Create* button

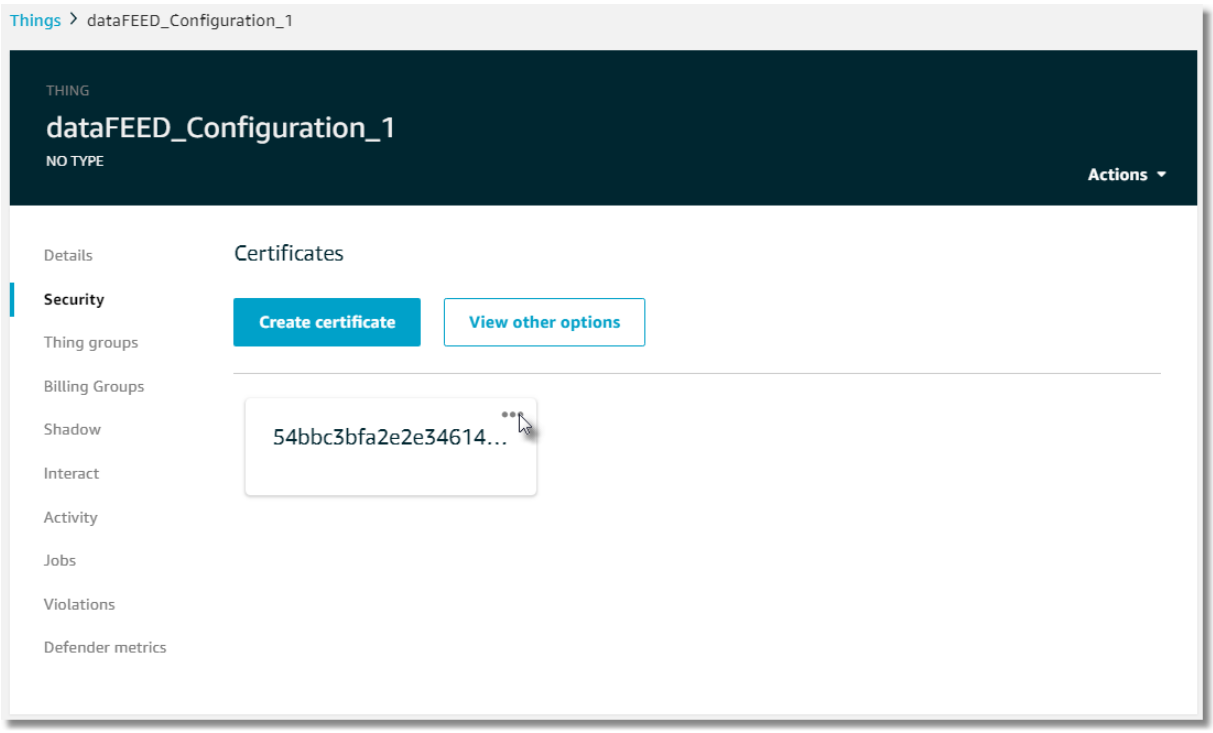
Attach *AWS IoT* Policy to Gateway

- Return to *AWS IoT Console* main screen and navigate to *Manage/Things* page
- Click on device name of gateway

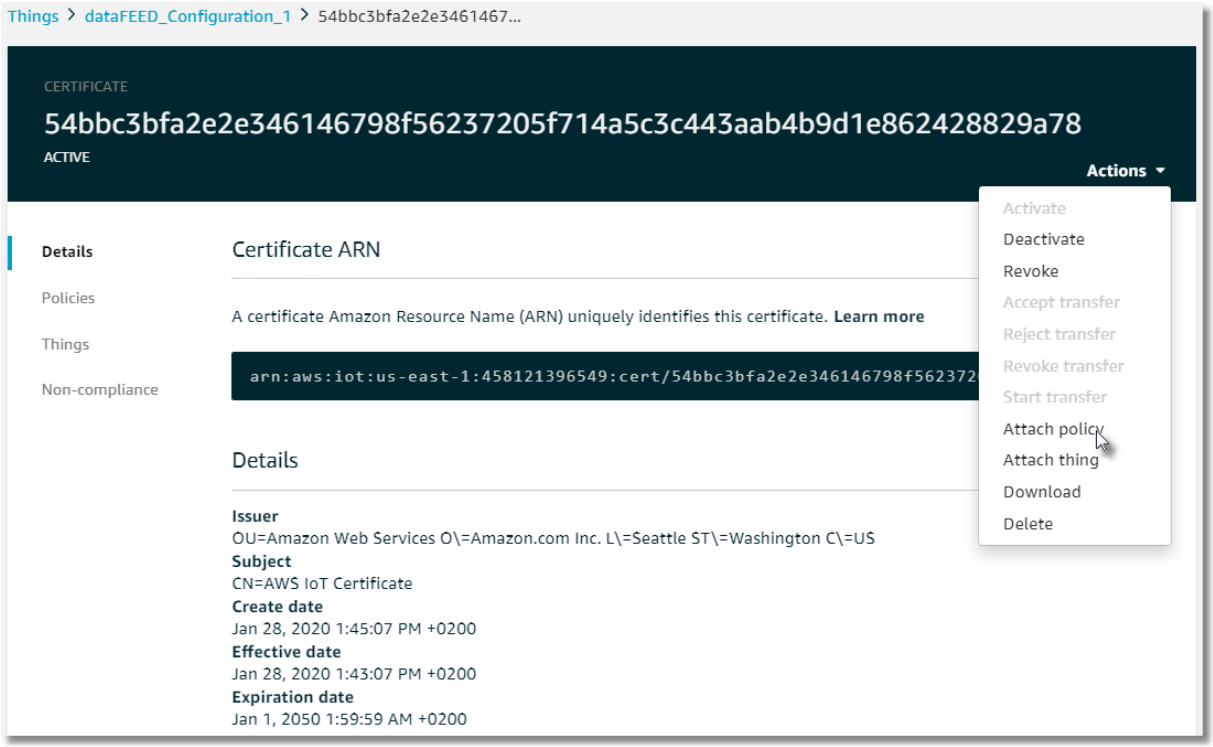
© Softing Industrial Automation GmbH



- Click on *Security*
- Click on hash value of created device certificate

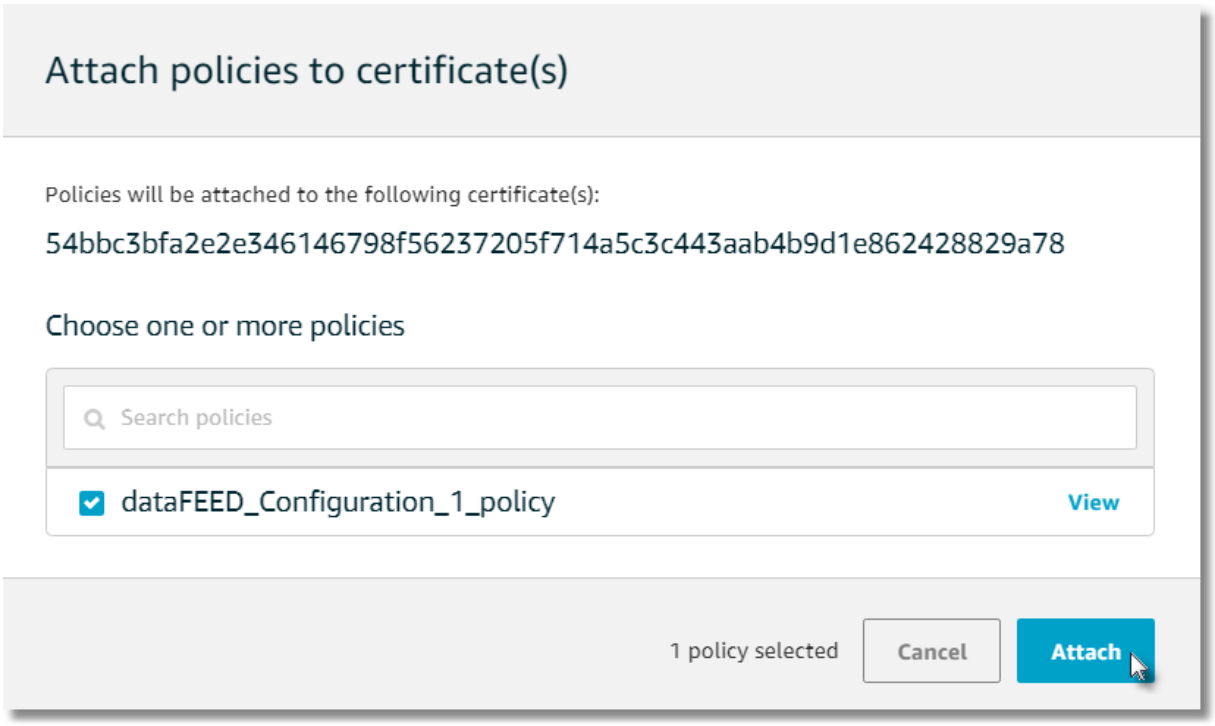


- Select *Attach policy* in *Actions* menu



- Attach created policy to certificate

© Softing Industrial Automation GmbH



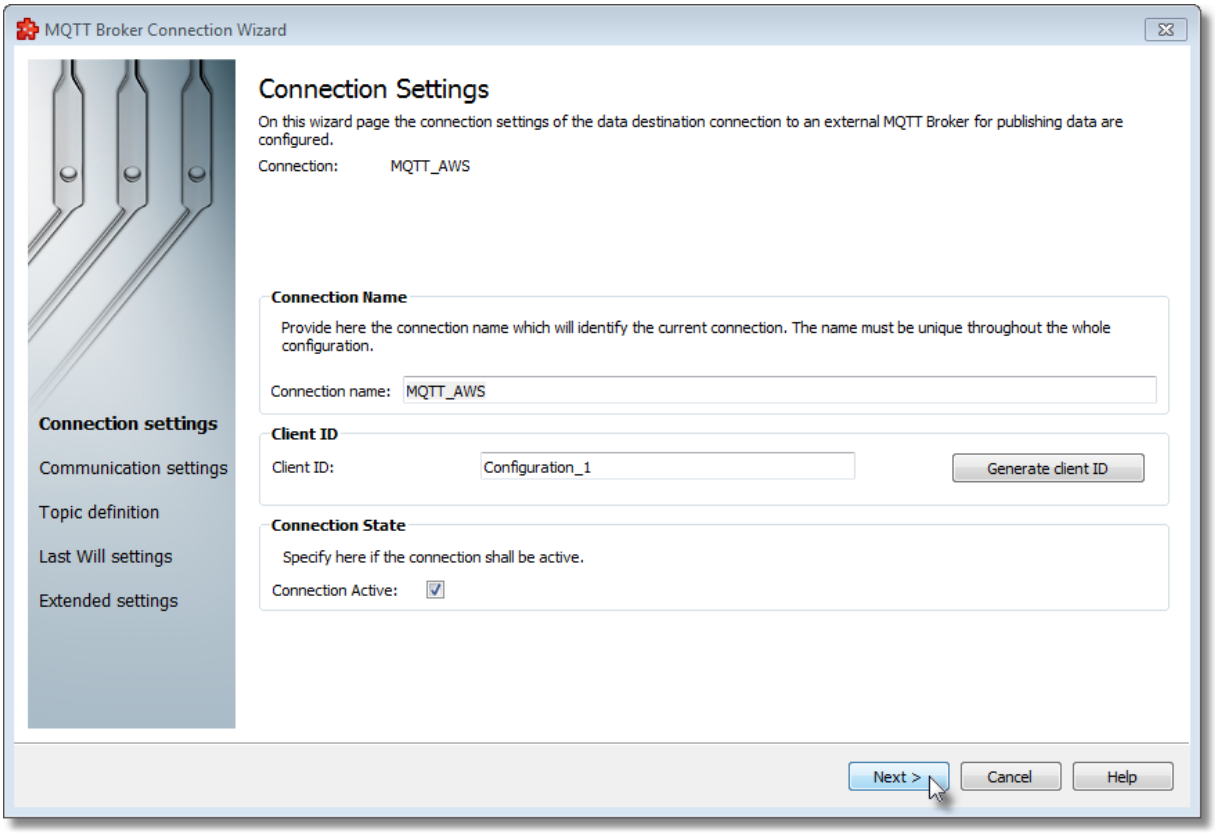
Determine AWS MQTT Broker Address

- Return to *AWS IoT Console* main screen and navigate to *Settings*
- *MQTT Broker* address can be found in Custom Endpoint field

NOTE: The address looks similar to "*a23px2f5602wdn-ats.iot.us-east-1.amazonaws.com*"

Define MQTT Broker Connection

- Start dataFEED OPC Suite Configurator
- Navigate to *Data Destination/MQTT Broker* page
- Add a new data destination (*Ctrl+A*)

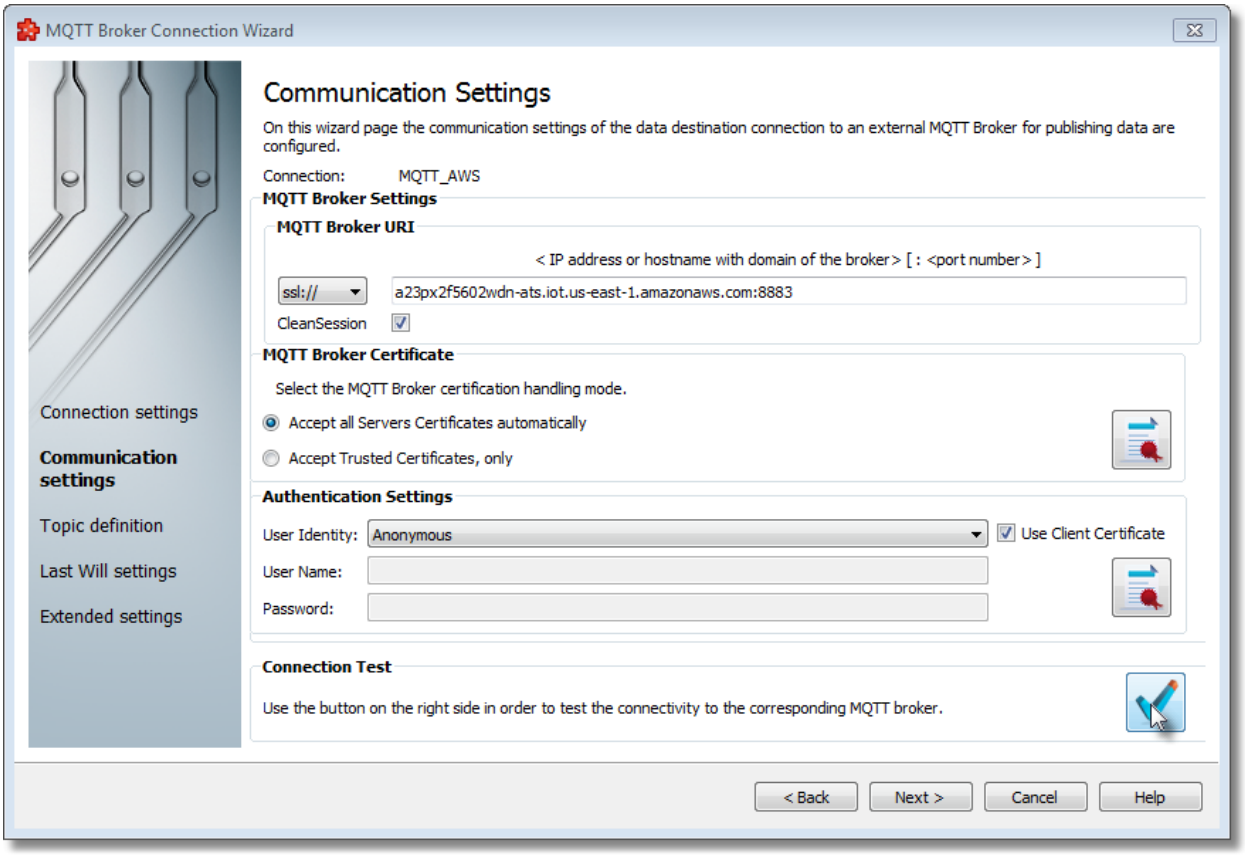


- Define *MQTT Broker Connection Name* and unique *Client ID*

NOTE: *Device ID* has to be allowed in ***iot:Connect*** policy (see section *AWS Preparation Steps*)

- Navigate to *Communication Settings* page (page 2) of *MQTT Broker Connection Wizard*

© Softing Industrial Automation GmbH



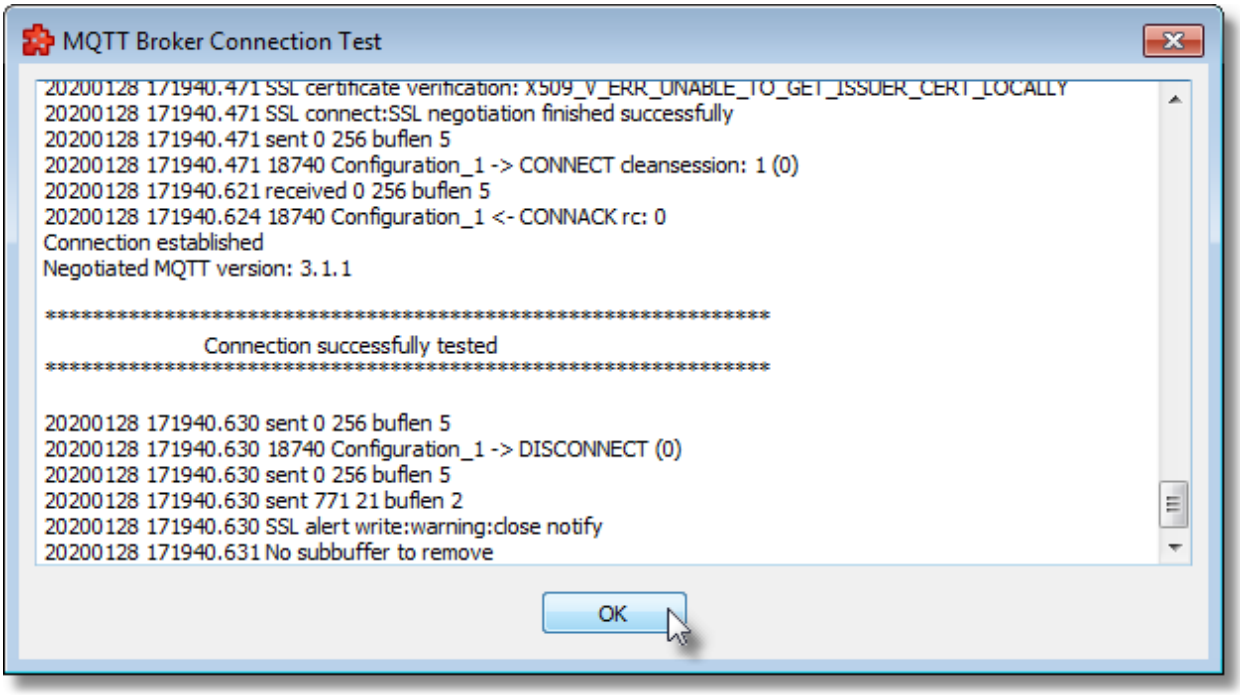
- Set *MQTT Broker URI* to *ssl://*
- Enter *AWS MQTT* broker address as prefix in *< IP address or hostname with domain of the broker > [: <port number>]* field
- Enter port number *"8883"* as postfix in *< IP address or hostname with domain of the broker > [: <port number>]* field
- Enable *MQTT Clean Session* flag

NOTE: *AWS Broker* disconnects, if flag is not set (see https://docs.aws.amazon.com/iot/latest/developerguide/protocols.html)

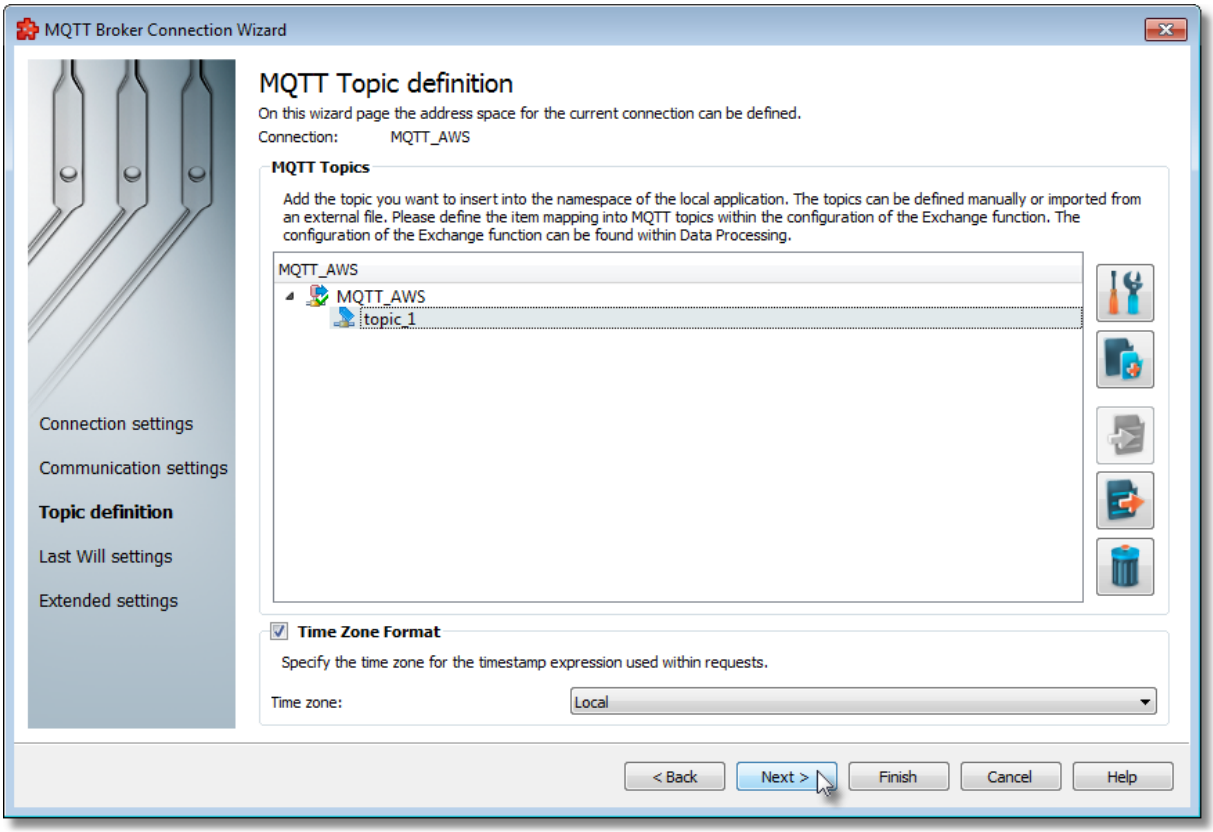
- Set Authentication Settings/User Identity to Anonymous
- Activate *Use Client Certificate* checkbox
- Select the downloaded certificate "**-certificate.pem"* and private key "**-private.pem.key*"

NOTE: Before the private key can be selected the certificate file extension has to be changed to ".pem"

- Press Connection Test button to verify connection



- Navigate to *MQTT Topic* definition page (page 3) of *MQTT Broker Connection Wizard*

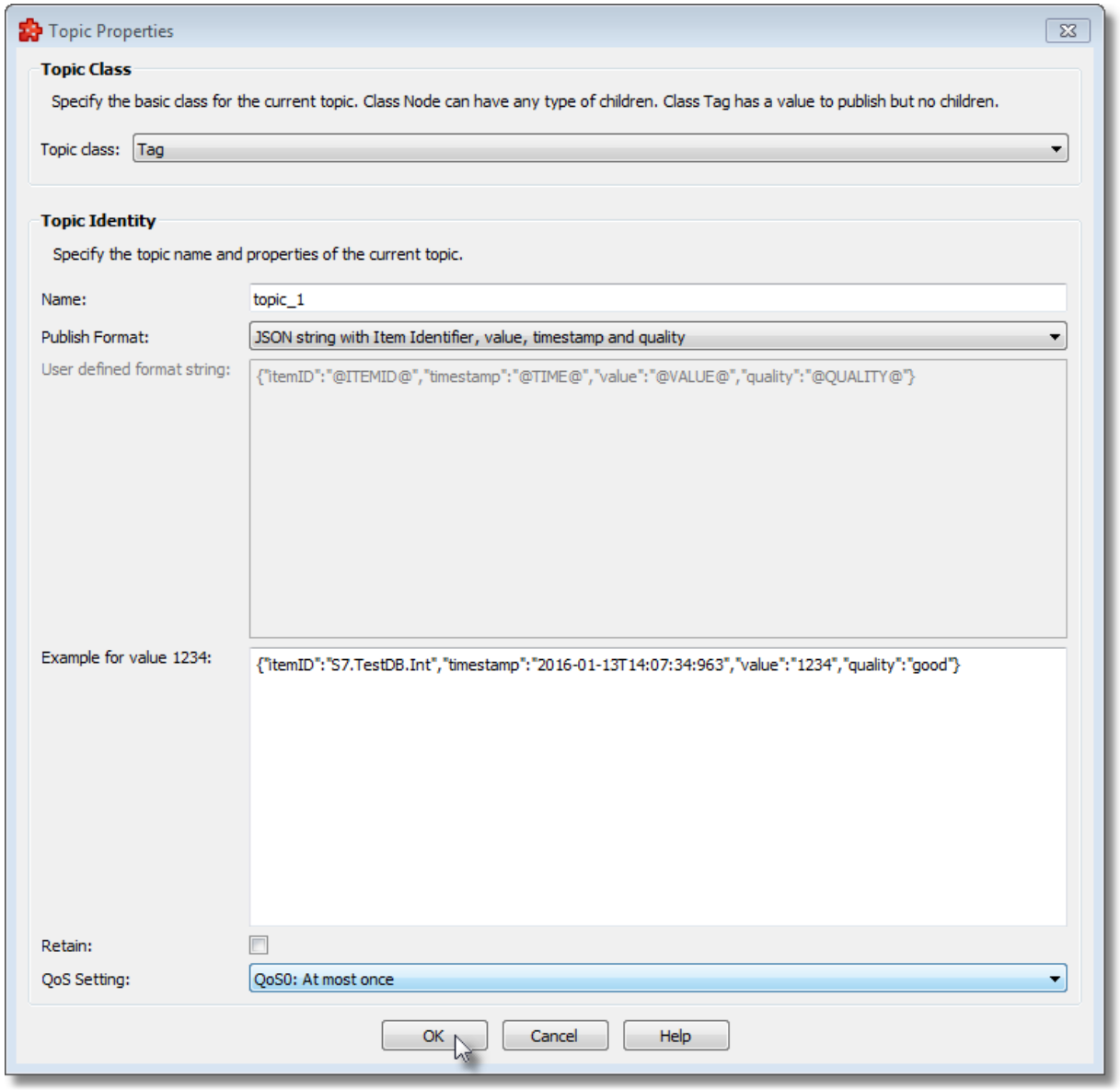


- Add topics

NOTE: Topics have to be allowed in ***iot:Publish*** policy (see section *AWS Preparation Steps*)

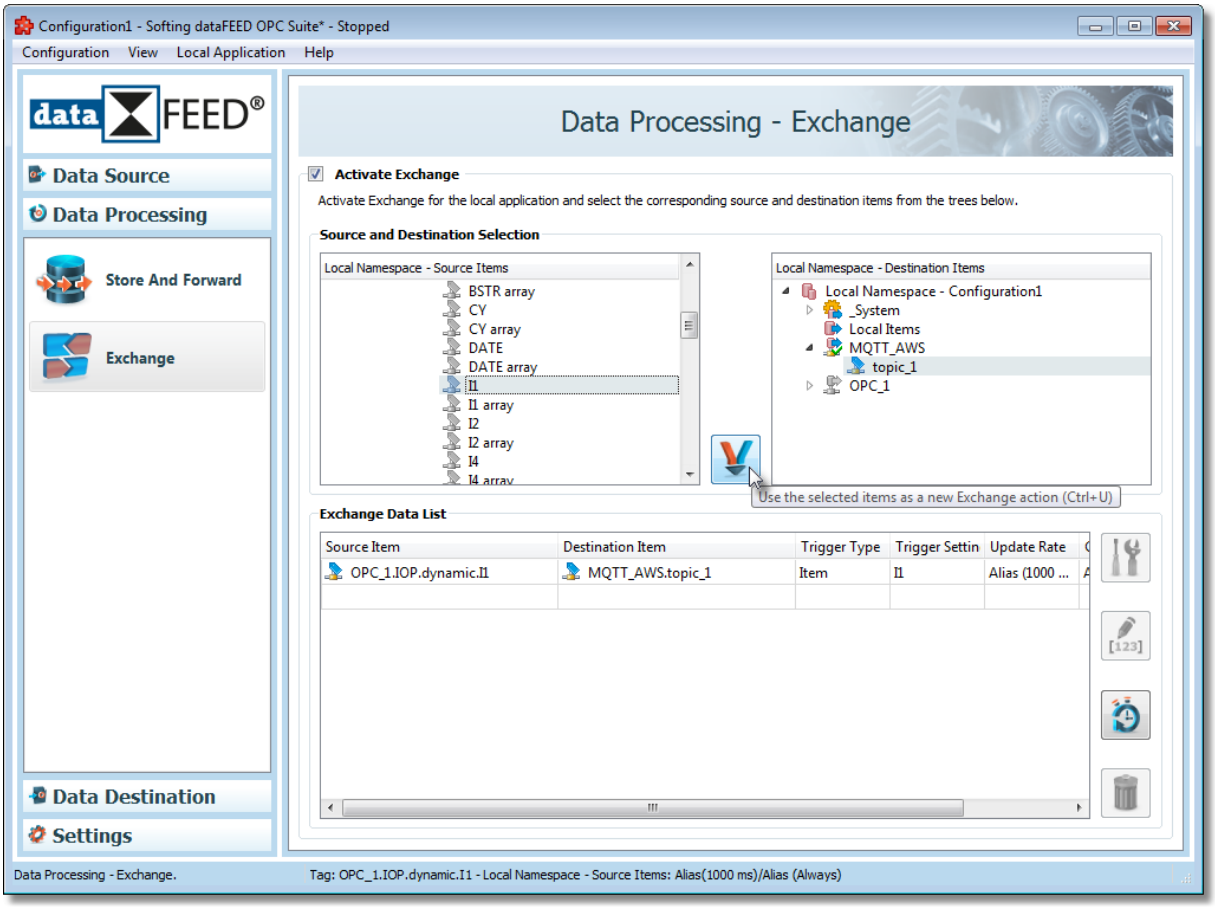
© Softing Industrial Automation GmbH

- Deactivate *Retain* flag
- Press *Save* button



Navigate to Data Processing/Exchange page

- Add exchange task for copying data source values to *MQTT* topic



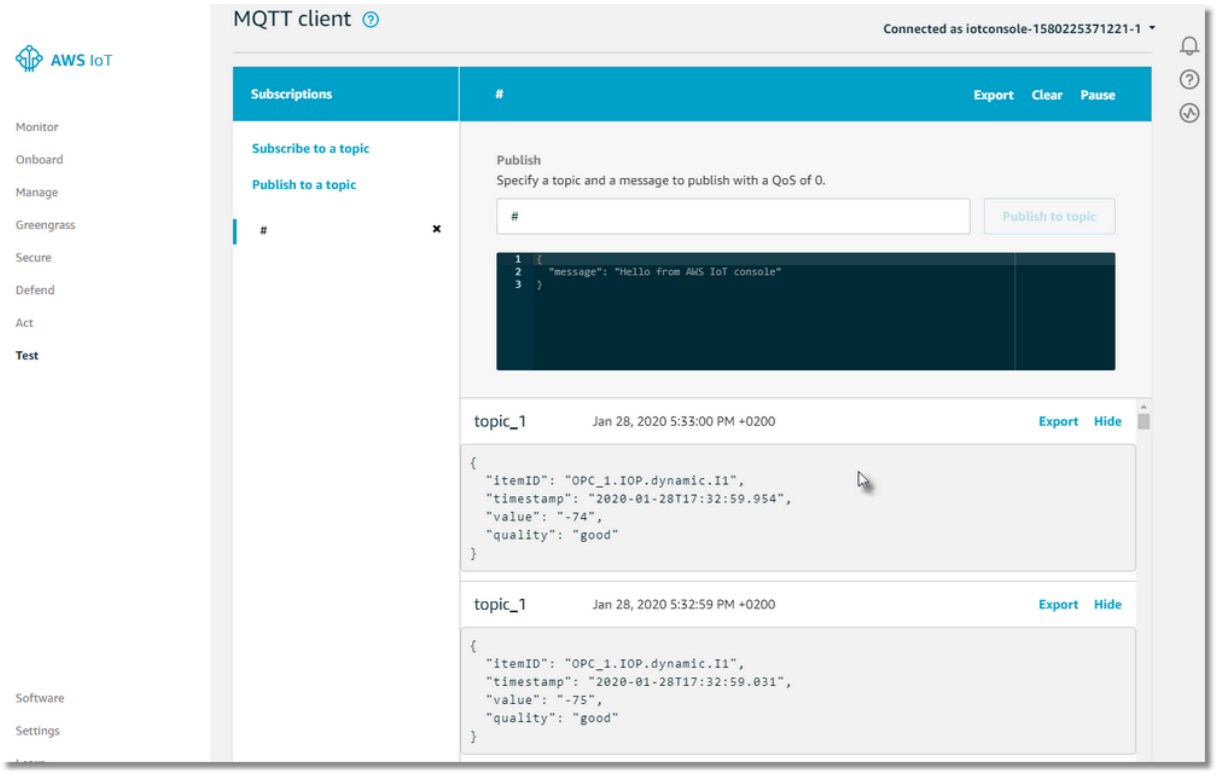
Finish dataFEED OPC Suite Configuration

- dataFEED OPC Suite configuration is finished
- Save configuration and start dataFEED OPC Suite

Monitor Values in AWS

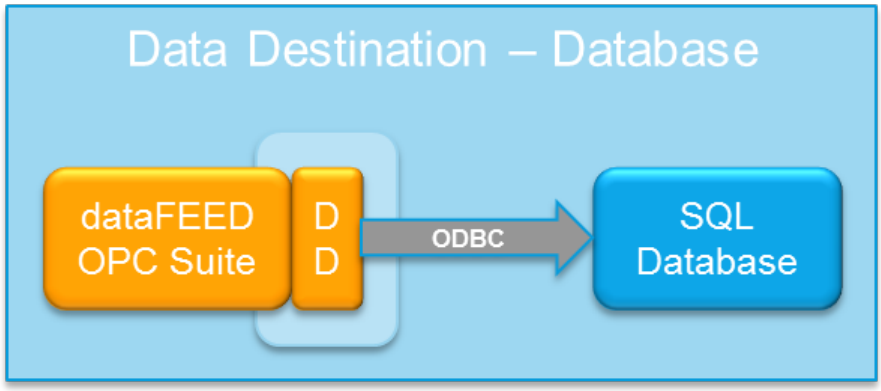
- Return to *AWS IoT Console* main screen and navigate to *Test*
- Subscribe to all topics using the wildcard *"#"*
- Received *MQTT* messages look different, depending on topic format settings and options for the collected *OPC* / *PLC* values

© Softing Industrial Automation GmbH



7.3.3.5 Database

This section of the Softing dataFEED OPC Suite Online Help describes how to configure the database data destination functionality of Softing dataFEED OPC Suite.



OPC2Database

OPC to Database stores OPC data from any Data Access or XML DA server in databases. It supports any type of database to which an ODBC connection can be established through a Data Service Name (DSN). The data read by Softing dataFEED OPC Suite is stored into databases via simple SQL statements.

Softing dataFEED OPC Suite allows the following operations to be done:

- Configure the connection to the desired database
- Define INSERT INTO statements as actions to be triggered
- Define general SQL statements as actions to be triggered
- Transfer of OPC data from any DA or XML DA server to a Database server
- Add placeholders from the server items to be stored in database

The Database add-on can be activated or deactivated by selecting / deselecting the *Activate Database* checkbox. When deactivating the Database add-on, the configured database connections and actions are not lost.

Database connection is done through *Open Database Connectivity* (ODBC) which provides a standard software interface for accessing the most common types of databases like: Microsoft SQL Server, MySQL, Oracle, PostgreSQL, IBM DB2. Thus, dataFEED OPC Suite uses ODBC to execute SQL commands on a database. It accomplishes this by using a driver as a translation layer between dataFEED OPC Suite and database management system (DBMS). Each driver is configured with a DSN that defines the connection to the target database.

The DSN connection string must contain the following:

- Database location: a host name or IP address and optionally a database name and port number
- Logon information, if are required by the database
- Other properties such as: connection pooling, buffer size, read/write state.

A more detailed description can be found on the Configuration: Data Destination - Database page.

Database tree

The main control of the *Database* add-on is the *Databases tree*. This tree contains database actions grouped under database connections which are displayed below the root node *Databases*.

Various operations are allowed on the tree items based on their type:

- for the root node Database the following operations are allowed: *Add Database Definition*, *Properties*.
- for database definitions the following operations are allowed: *Add Database Action*, *Delete Database Definition*, *Properties*.
- for database actions the following operations are allowed: *Delete Database Action*, *Modify Trigger Type*, *Modify Trigger Settings*, *Properties*.

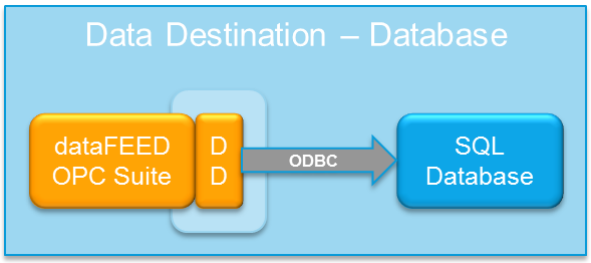
Each database connection contains a list of database actions which, when selected, will populate the *Database Action* section of the add-on main page with the respective expanded action (so that it will be editable).

Specific details about how to create and handle items in the Database tree can be found on the Configuration: Data Destination - Database page.

© Softing Industrial Automation GmbH

A description of the configuration file format can be found on the Configuration Files page.

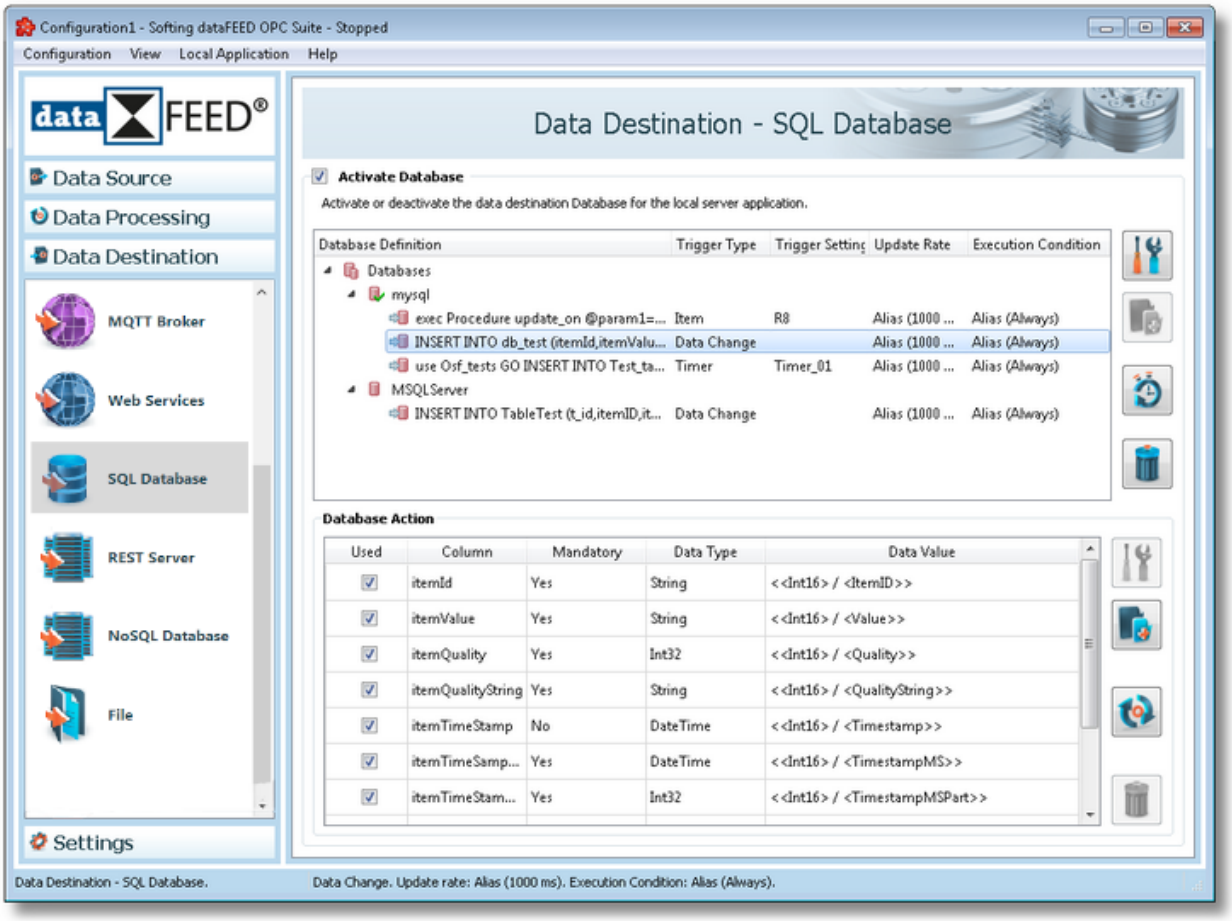
7.3.3.5.1 Configuration



The OPC2Database functionality is configured in the Data Destination - Database main user interface. The main add-on UI page can be reached by choosing the *Database* option from the *Data Destination* section in the navigation panel on the left. The Database Action section provides different controls depending on what type of database action is selected in the database tree. If a database connection or the root *Databases* node is selected then this section will be disabled. Three types of database actions exist:

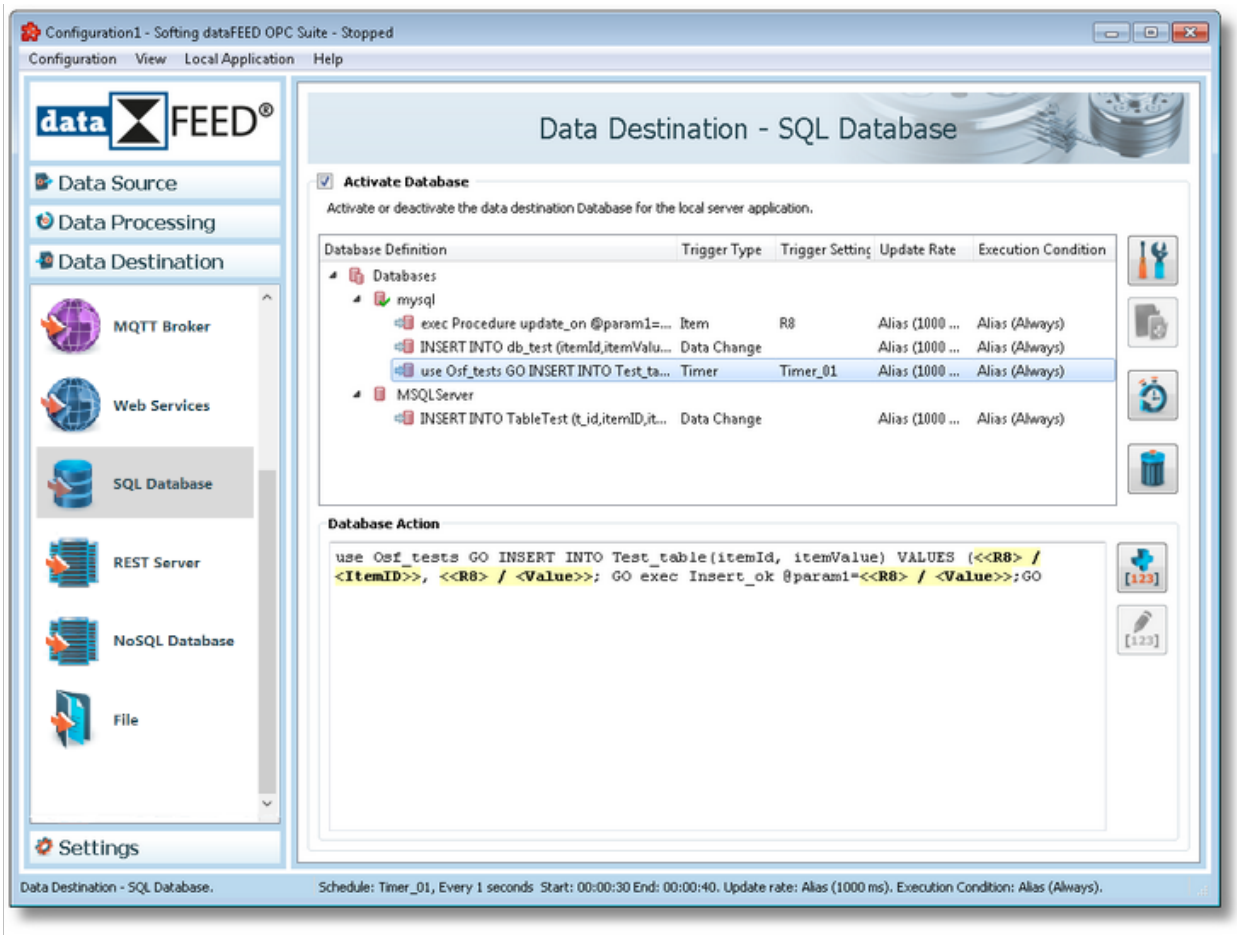
- *INSERT INTO* SQL statement which defines an Insert Into action in a user friendly way.
- *Stored Procedure* is an action editor which allows typing calls to SQL stored procedures.
- *General SQL Statement* is an action editor which allows typing any SQL query code.

The image below illustrates the main user interface of the Database add-on (Insert Into database action selected):



The image below represents a general SQL statement database action (the Stored Procedure action appears in an action editor as well):

© Softing Industrial Automation GmbH



For more details about the OPC2Database functionalities see the Database page.

The entire *Database* add-on functionality can be activated or deactivated by selecting / deselecting the *Activate Database* checkbox.

Note: If database definitions or database actions are defined and the add-on functionality is deactivated, these definitions will not be deleted from the configuration files (task and subscription files). Instead they are marked as deactivated - prefixed with the deactivated mark: *#!* These lines are then ignored by the local dataFEED OPC Suite application.

By default the OPC2Database functionality is disabled. It needs to be enabled first by selecting the *Activate Database* checkbox at the top of the *Data Destination - Database* main UI page in order to use it. When this add-on is enabled the Database Wizard will be shown.

The *Data Destination - Database* main view consists of:

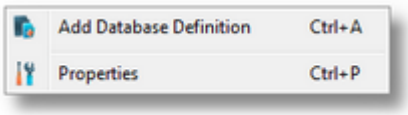
- The *Activate Database* checkbox which allows enabling / disabling this add-on.
- A tree displaying the database definitions and database actions related to them.
- An action editor which offers support in editing the selected database action.

Database definition handling

From the Data Destination - Database main user interface, certain operations can be performed using the buttons on the right, the keyboard shortcuts or the corresponding context menu actions:

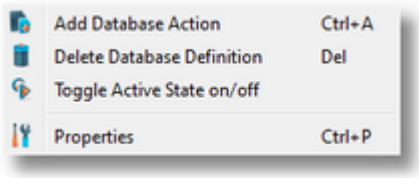
- Change the properties of the selected database definition using the *Properties* button  , pressing the keyboard shortcut CTRL+P or selecting the *Properties* action from the context menu.
- Change the properties of the selected database action using the *Properties* button  , pressing the keyboard shortcut CTRL+P or selecting the *Properties* action from the context menu.
- Add a new database definition selecting the *Databases* root element and using the *Add* button  , pressing the keyboard shortcut CTRL+A or selecting the *Add* action from the root element context menu.
- Add a new action for the selected database definition using the Add button  , pressing the keyboard shortcut CTRL+A or selecting the Add action from the database definition context menu.
- Remove the selected database definition / database action using the *Delete* button  , pressing the DELETE / DEL keyboard key or selecting the *Delete* action from the context menu.
- Modify the trigger type for the selected database action by selecting the *Modify Trigger Type* action from the context menu.
- Modify the trigger settings for the selected database action by selecting the *Modify Trigger Settings* action from the context menu. If the trigger type is *Data Change* this context menu action is disabled. For more information see the Item Trigger Wizard or the Timer Trigger Wizard.
- Specify the global alias values for the entire configuration using the *Configure global alias and initial values* button  or pressing the keyboard shortcut CTRL+D.

The context menu of the *Databases* root node looks like this:

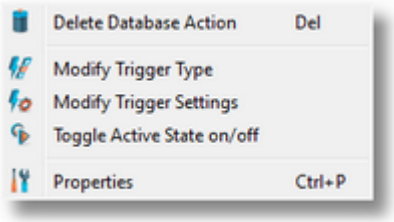


The *Database Definition* context menu looks like this:

© Softing Industrial Automation GmbH



The *Database Action* context menu looks like this:



Each Database connection and action can be activated or deactivated separately using Toggle Active State on/off option. Once deactivated the item is grayed out inside the view.

For database action definitions in the Database tree the trigger type, the update rate and the execution condition can be defined. These values can be edited by clicking in the corresponding cell.

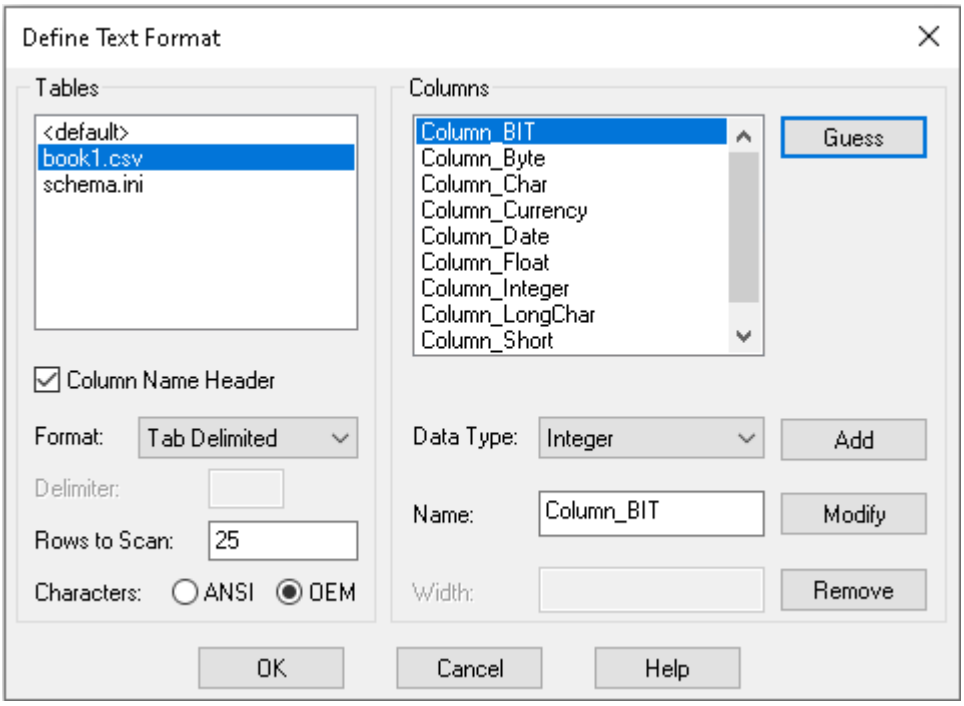
Connecting to .txt /.csv files through Open Database Connectivity (ODBC) 32-bit.

Writing .txt / .csv files is possible based on ODBC drivers - acting similar as any other supported database.

To configure the ODBC connection follow the steps:

- From System DSN, create a new Data Source by selecting for example the default Microsoft Text Driver
- Add a Data Source Name and select the location of the source .txt / .csv file
- Press the Options and Define Format buttons to open the Define Text Format dialog
- Continue by defining the source file specific settings: usage of Column Name Header, Format (tab ,comma delimited), data-types of the columns, etc

See below an ODBC definition example to a local .csv file TAB separated, containing columns with all supported Microsoft ODBC Text Driver data types.



The schema.ini file will contain:

ColNameHeader=True

Format=TabDelimited

MaxScanRows=25

CharacterSet=OEM

Col1=COLUMN_BIT Bit

Col2=COLUMN_BYTE Byte

Col3=COLUMN_CHAR Char Width 255

Col4=COLUMN_CURRENCY Currency

Col5=COLUMN_DATE Date

Col6=COLUMN_FLOAT Float

Col7=COLUMN_INTEGER Integer

Col8=COLUMN_LONGCHAR LongChar

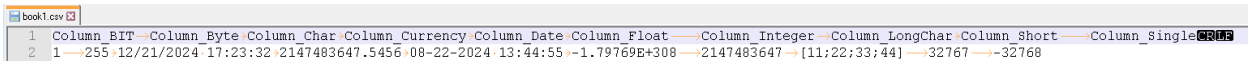
Col9=COLUMN_SHORT Short

Col10=COLUMN_SINGLE Single

Note: For German OS, if the .csv is comma separated, then the schema.ini file must contain Format=CSV

Example of book1.csv TAB separated file:

© Softing Industrial Automation GmbH



Example of values in the .csv file:

Column_BIT	Column_Byte	Column_Char	Column_Currency	Column_Date	Column_Float	Column_Integer	Column_LongChar	Column_Short	Column_Single
1	1	strArray0;strArray1;strArray2	1234567890123.45	08-22-2024 13:44:55	1.7976E+308	-2147483648	[-128;255;127;0]	-32768	-32768
0	255	12/20/2024 17:23:32	2147483647.5456	08-22-2024 13:44:56	-1.7976 9E+308	2147483647	strArray0; strArray1; strArray2; strArray3	32767	32767
1	0	11;22;33;44	-12345678901234.5	08-22-2024 13:44:57	1.7976 9E+308	0	String	1	1

Note: The columns with data-type Char and LongChar can also contain Arrays and DateTime values (the string value must match the configured DateTime format in dataFEED OPC Suite)

The functional limitations for .txt / .csv data Read and possible conversion failures are described below:

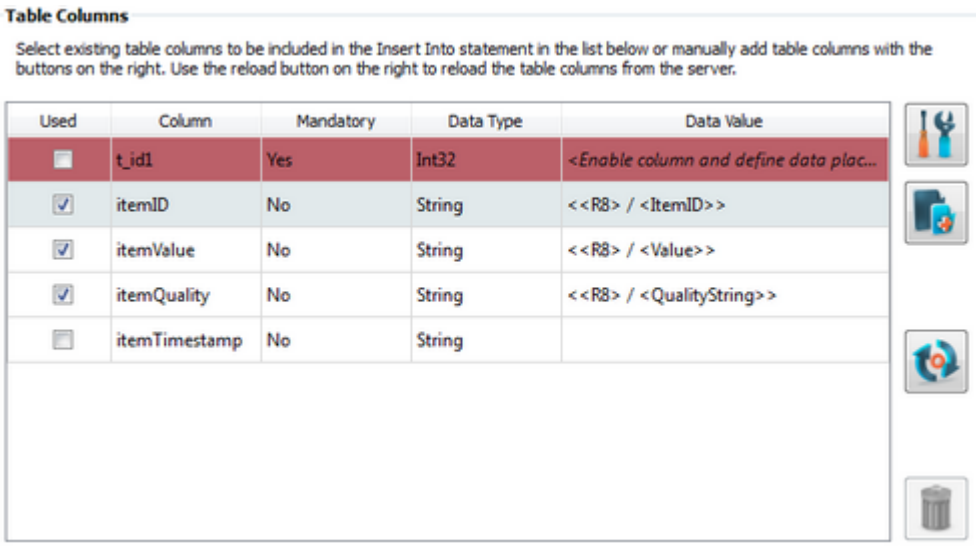
- The .txt / .csv columns values must have the correct data-type matching the schema.ini file and the dataFEED destination variable types.
- For writing into DATETIME variable from a column value as a DateTime but with the data-type as CHAR, the column-value must match the "Date format" defined in Database Read Definition Wizard. If the column data-type is DATE, then the column-value must match the ODBC driver required format; example for Microsoft Text Driver, the Date format is: mm-dd-yyyy hh:mm:ss.
- For writing into ARRAY variables, the columns data-types must be CHAR with the correct size set. The index separator must match the "Array elements separator" defined in Database Read Definition Wizard; example1: column value: 1;2;3;4 can be written in ARRAY variable, output value being: [0,3][1,2,3,4]; example2: column value: [1;2;3;4] can only be written in ARRAY variable of type INT.

Action editor handling for INSERT INTO actions

The INSERT INTO action editor offers a convenient way to edit an insert into SQL statement by displaying an interactive table grid containing the table column names as rows with their respective data associations. The Insert Into action editor table has the following fields:

- *Used* - column which determines whether the respective column will receive any data from the incoming transaction. When you select the checkbox in this cell the Data Value Placeholder Wizard will open. In this wizard you can associate the table column name with a placeholder. When the checkbox is unchecked this association will be lost and the defined placeholder will automatically be deleted.
- *Column* - column name in the selected database definition for storing the item/placeholder value.
- *Mandatory* - indicates whether the respective table column is compulsory or not. It can contain two values - *Yes* or *No* - denoting whether an insert action will fail or not. When the *Used* column for the respective row is not selected and the value of this column is *Yes*, the background color of the Database Action row will be red, emphasizing that an insert into action will fail if the respective table column is not included in the transaction. The background color will not be altered if the value of this field is *No* thus proving that the table column is not necessary. The row's background color will not be altered, either, if the field is mandatory and the *Used* column is selected.
- *Data Type* - column displaying the data type of the database table column.
- *Data Value* - column in the interactive table which displays the user defined placeholder. If the column *Used* is not checked this column will contain no placeholder reference.

An example of a complex action editor containing selected and unselected rows of both mandatory and non-mandatory types can be seen below:



For more details see the Database Action Wizard - Table Columns page.

Note: dataFEED OPC Suite displays texts encoded in UTF-8. The Unicode characters are thus fully supported for Database names or table column names.

© Softing Industrial Automation GmbH

Action editor handling for General SQL Statement actions

The *General SQL Statement* editor offers the possibility to edit a general SQL statement or a SQL stored procedure by inserting free SQL text and placeholders.

For example, the following statement first runs an initialization statement, then it creates an Insert Into action and executes them on a database. dataFEED OPC Suite replaces the placeholders accordingly:

```
use Test_Database GO INSERT INTO Test_Table (itemId,itemValue)  VALUES (<<R8> / <Item
```

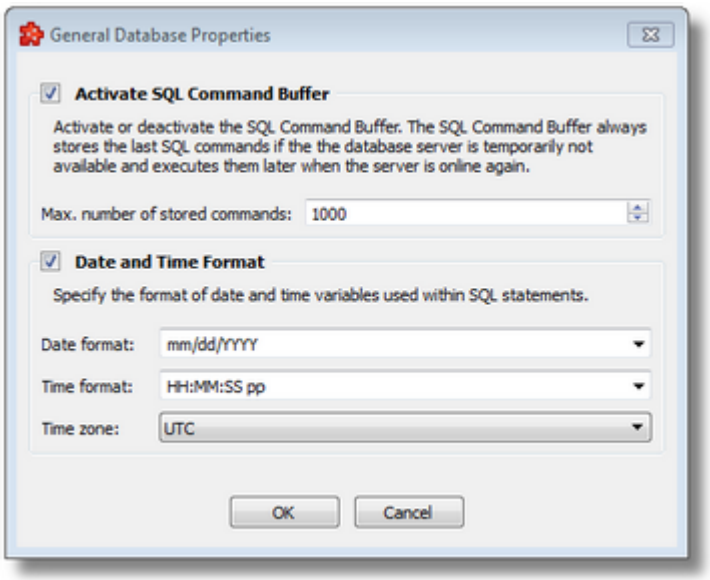
For more details see the Database Action Wizard - SQL Statement page.

Note: Configuration files are encoded in UTF-8, thus providing full support of Unicode characters.

7.3.3.5.2 Database Properties

When the *Databases* root node is selected, invoking the properties action by any means (button, context menu or keyboard shortcut) displays a dialog that allows the user to activate or modify some advanced settings. The *General Database Properties* dialog can be opened by using the *Properties* button  , by pressing the keyboard shortcut CTRL+P or by selecting the *Properties* action from the context menu.

Note: The *General Database Properties* dialog will not open unless the *Databases* root node is selected.



This dialog contains two sections that can be independently enabled or disabled.

The first section, *Activate SQL Command Buffer*, controls whether to store the SQL statements in a file if the database connection has been lost for any reason. By selecting this checkbox the functionality is activated and SQL statements will be stored, starting with the first one that could not be executed.

The maximum number of stored SQL statements can be configured through the *Max. number of stored commands* edit box. If this number is reached and the connection to the database has not been reestablished yet, the storage of SQL statements stops and the subsequent commands are lost.

When the connection is reestablished, the stored SQL statements are executed. The storage file is removed after all the commands have been executed. For each database definition in the database tree a stored SQL statements file is created in the current configuration folder (when needed).

By default this checkbox is disabled and SQL statements are not stored.

Note: If *Activate SQL Command Buffer* is not activated or if the value for *Max. number of stored commands* is too small, several values could be lost.

The second section, *Date And Time Format*, handles the time and date formats that are to be used when the timestamp placeholders are expanded and their values written to the database. The default values are `HH:MM:SS pp` for the *Time format* and `mm/dd/YYYY` for the *Date format*. Other defined values for Time and Date format are:

- *Time format*: `HH:MM:SS.FFF pp, HH:MM:SS.FFF, HH:MM:SS`
- *Date format*: `dd-mm-YYYY, dd.mm.YYYY, dd/mm/YYYY, YYYY-mm-dd, mm-dd-YYYY`

Values for Time and Date format can also be defined manually but in this case special care must be taken to ensure that the definitions are recognized by the local dataFEED OPC Suite application.

In addition to the default values listed in the format selection combo boxes, the user can add his/her own custom format. Custom Date and Time format strings are accepted as valid, when the format strings contain in their composition the following accepted format substrings and format delimiters. These formats allow the inclusion of specific locale information for the regional settings.

- *Valid Date format strings*: `AAAAAA, aaa, BBBBBBBB, bbb, ccccccccccccccccccc, dd, jjj, mm, UU, ww, xxxxxxxxxx, yy, YYYY`
- *Valid Time format strings*: `II, HH, MM, SS, pp, XXXXXXXX, FFFFFF, FFFFF, FFFF, FFF ,FF, F`
- *Valid Date and Time format delimiters*: `<dot>, <colon>, <slash>, <comma>, <backslash>, <vertical bar>, <semicolon>, <space>, <tab>, <minus>`

Time Zone, handles the source localization time: UTC (Universal Time Coordinated) or local time (local time localization). The default time zone is UTC.

© Softing Industrial Automation GmbH

When the *Time,Date Format and Time Zone* option is not selected the default values are used.

Note: Settings that are modified in this section apply to all database definitions and their corresponding actions. This could cause some limitation when using different database types or different settings for the same database type.

Attention: If the date and time format configured in Configurator does not match with the format in which the destination database stores such values, it may happen that the values will not be stored in the database or truncated/changed in some way. This is because the ODBC drives does not always report such errors from the database. It may also happen that the database will not report such errors at all.

It is recommended that the user check the dataFEED log files for error messages if they find the sent values are not correct. Also, manually executing the action defined in dataFEED against the database and see the actual error message to be able to change the action so that values are stored correctly.

This is a good approach for other data types also, like text, not only date and time, when the log files do not give enough information.

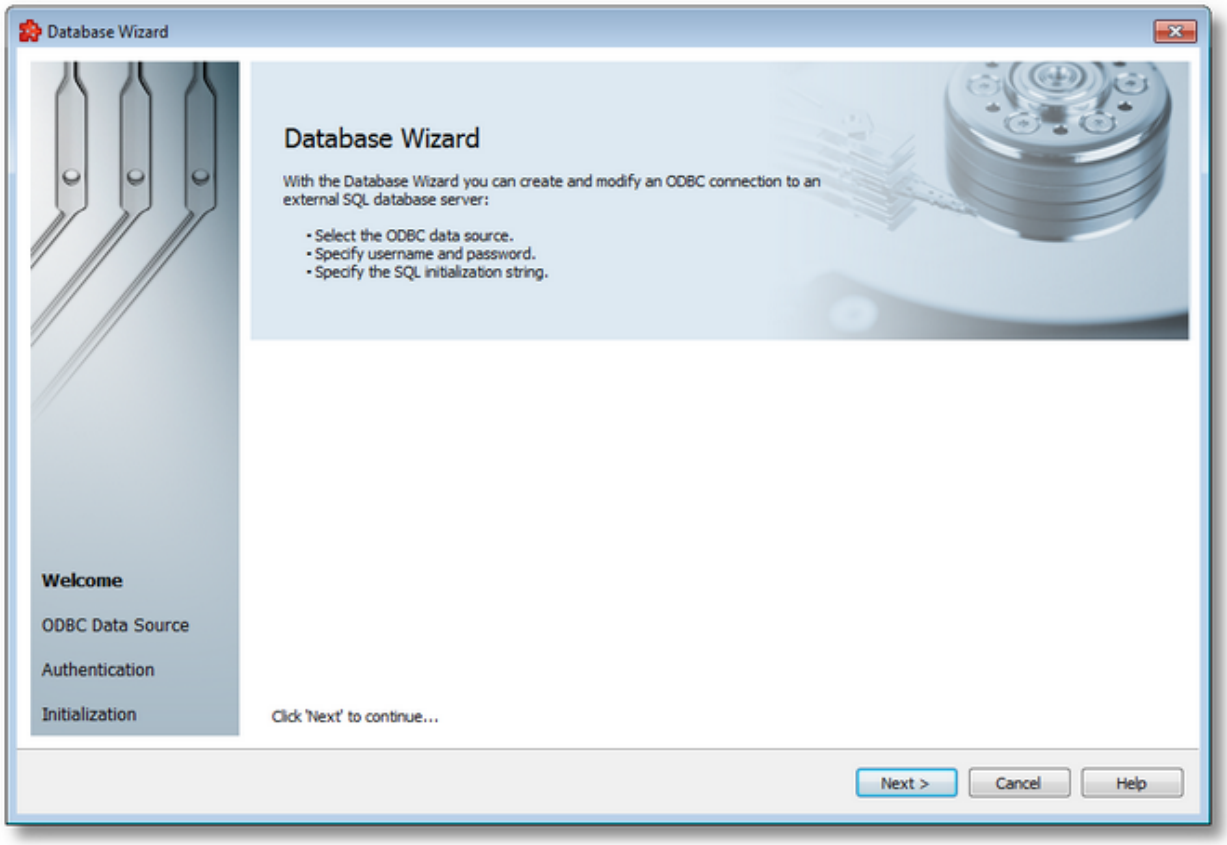
7.3.3.5.3 Database Wizard

- With the Database Wizard you can create and modify an ODBC connection to an external SQL database server:

  - Obtain general information about the Database Wizard @ Database Wizard - Welcome
  - Select the ODBC data source for the database connection @ Database Wizard - ODBC Data Source
  - Specify the user name and the password for the database connection @ Database Wizard - Authentication
  - Specify an SQL initialization statement for the database server @ Database Wizard - Initialization Statement

The Overview wizard page describes the basic steps for completing the database definition:

- Select an already configured ODBC data source connection.
- Define credentials for connecting to the selected ODBC connection.
- Define a database initialization statement that is to be executed the first time the local dataFEED OPC Suite application starts.

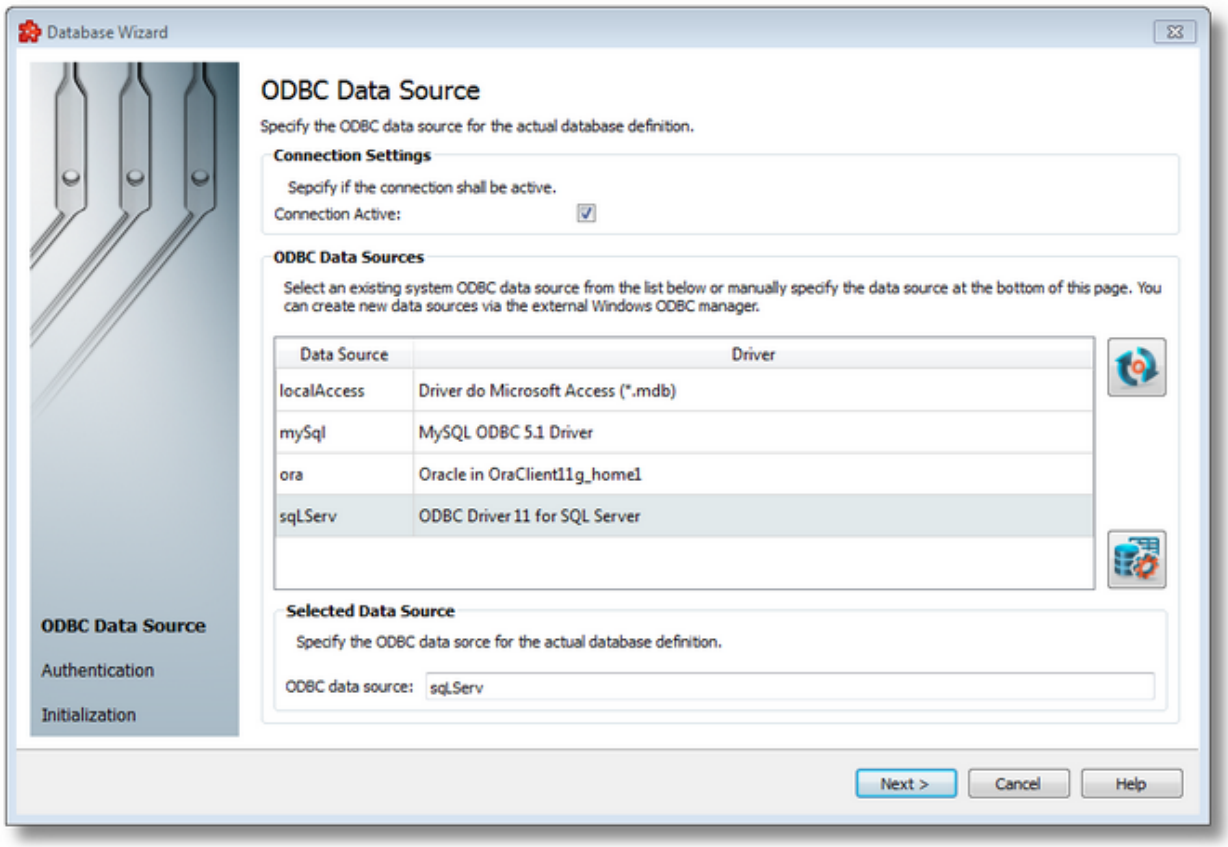


From this wizard page, you can continue with the database definition by pressing the *Next* button, or quit the wizard by pressing the *Cancel* button.

Database Wizard - ODBC Data Source >>>

The role of the *ODBC Data Source* wizard page is to provide an intuitive and simple way for configuring the database connections. It contains two sections: *ODBC Data Sources* and *Selected Data Source*:

© Softing Industrial Automation GmbH



The first section allows selecting an already defined ODBC connection from the System DSN. If the desired connection does not exist in the list, a new ODBC connection can be defined by using the *Open the Windows ODBC data source manager* button  .

Note: Only the System DSN data sources are displayed in the *ODBC Data Sources* table.

Using the *Reload ODBC data sources* button  or pressing the keyboard shortcut CTRL+R updates the available data source list. After a new ODBC database definition is added to the Windows System DSN, an update needs to be done in order to see the latest addition(s). The ODBC Data Source table contains 2 columns:

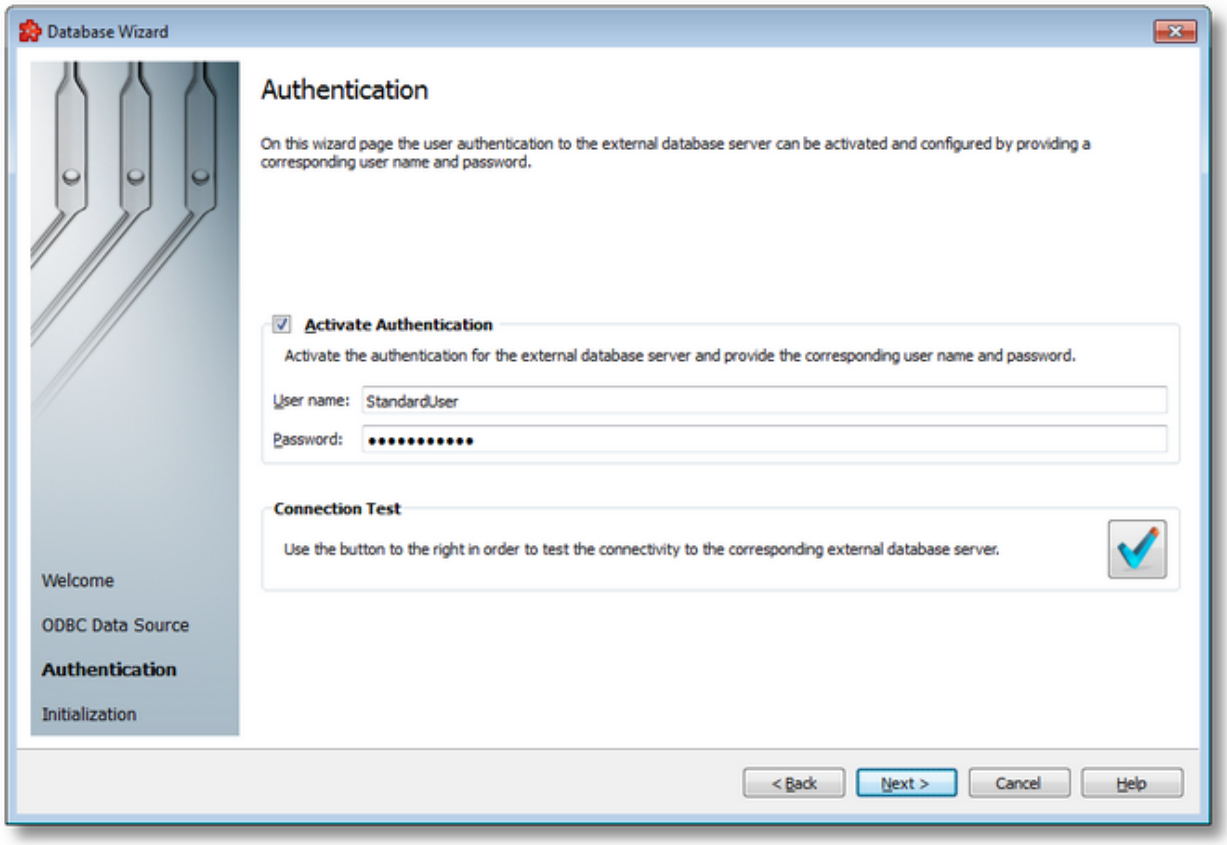
- *Data Source* - contains the ODBC Data Source name. This name will also be used as the database connection name in the *Databases* tree.
- *Driver* - contains the name of ODBC data source driver used for connecting to the database.

The second section contains an editable text box that is automatically populated with the selected DSN name. The name of the data source can also be manually written in case of an offline configuration when the DSN will be created later.

Note: It is mandatory to provide an *ODBC data source* name through which the database connection will be established, therefore it is not possible to continue the wizard without providing one.

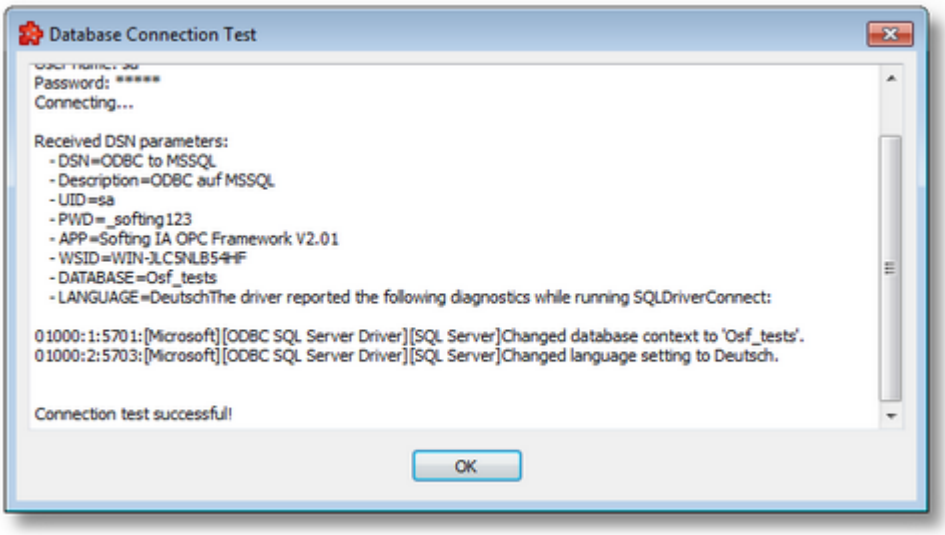
<<< Database Wizard - Welcome Database Wizard - Authentication >>>

The next step in configuring a database definition is to provide (if necessary) the authentication information for connecting to the ODBC. This is done in the Authentication page, which is organized as follows:

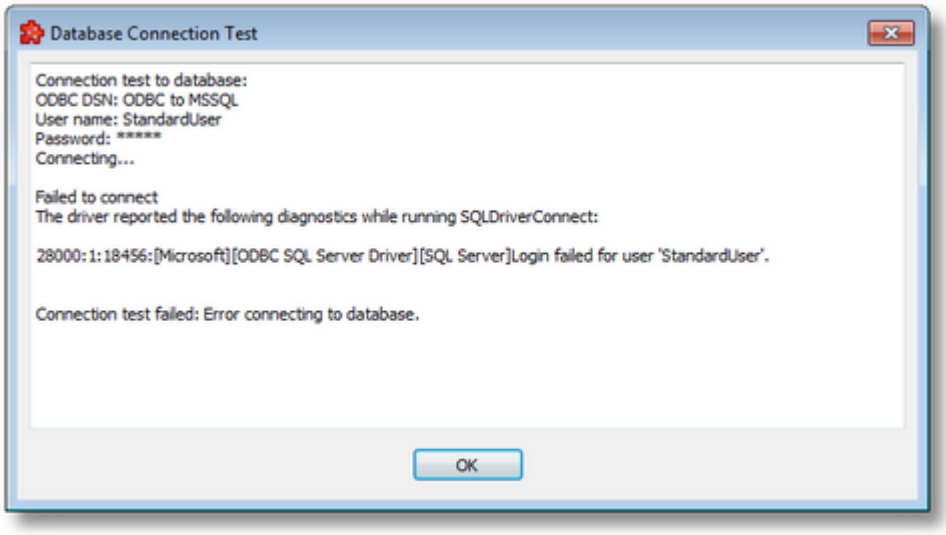


- A short description of this page's functionality is shown at the top of the wizard page.
- The *Activate Authentication* section has an associated checkbox for activating or deactivating this part of the database definition process. When active, the two fields for *User name* and *Password* are enabled and the corresponding data can be filled in. By default the Activate Authentication checkbox is disabled and the *User name* and *Password* fields are empty.
- The last section, *Connection Test*, allows testing the current configuration (including the non-mandatory user name and password). To perform the connection test, click the *Connection test* button  or press the keyboard shortcut CTRL+T. The *Database Connection Test* dialog appears where you can see the connection test results. A successful database connection test result is shown below:

© Softing Industrial Automation GmbH



Various information about the server is also displayed in this window. If the test connection failed, the database connection test result will look like this:

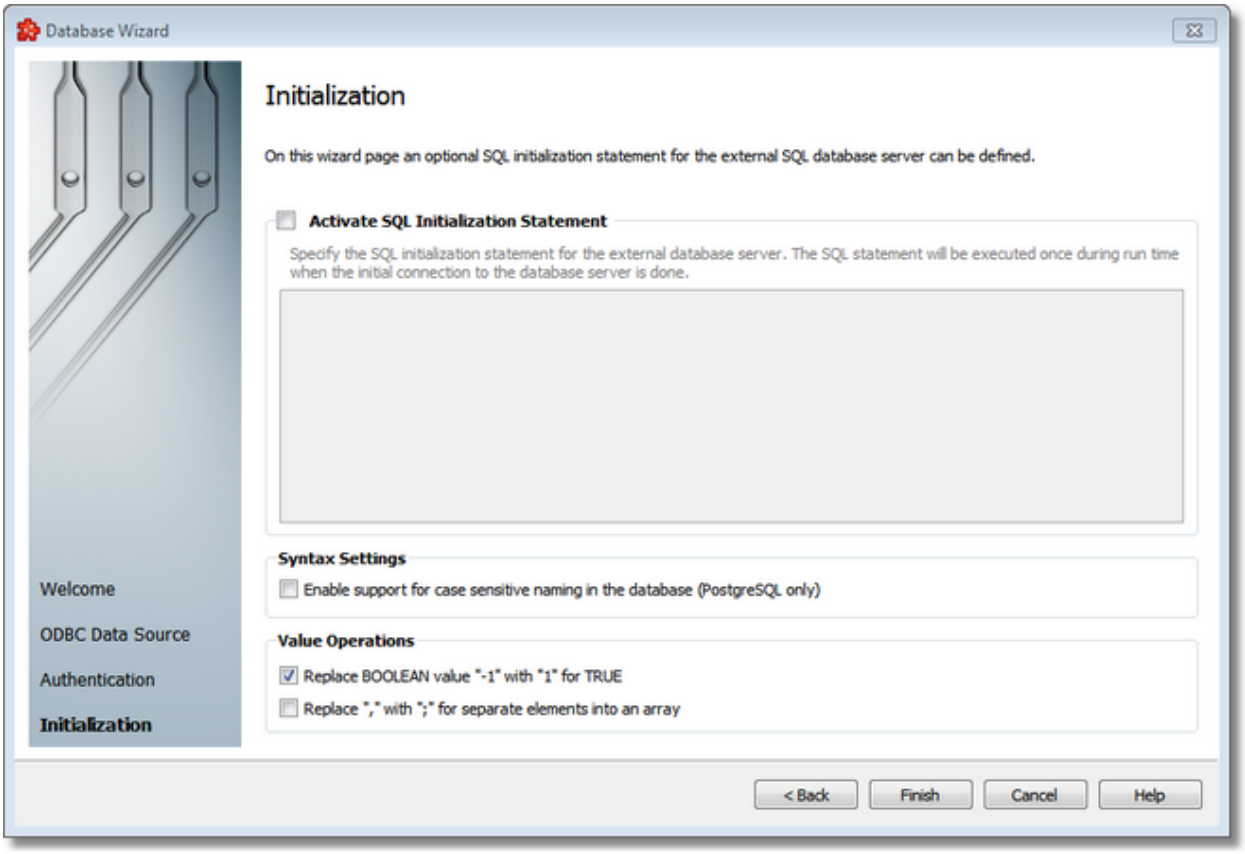


To proceed with the wizard and continue to define the database initialization statement press the *Next* button. To go back to the previous page (to change the DSN) use the *Back* button. Quit the *Database Wizard* by pressing the *Cancel* button.

<<< Database Wizard - ODBC Data Source Database Wizard - Initialization Statement >>>

The last page of the database definition wizard is the initialization page which allows defining an initialization statement for the external SQL database. It consists of a single section with an associated checkbox: *Activate SQL Initialization Statement*. By default this checkbox is not

selected, therefore no SQL initialization command will be executed. The wizard page is shown below:



When the *Activate SQL Initialization Statement* ckeckbox is selected, the free form SQL text editor is enabled where you can define a SQL statement:

```
ALTER SESSION SET NLS_LANG = JAPANESE_JAPAN.JA16EUC;
```

Note: This SQL initialization statement will be executed each time the local dataFEED OPC Suite application connects to the database, that is when the local dataFEED OPC Suite application starts and also when the local dataFEED OPC Suite application reconnects to the database. It will not be executed when adding new Database Actions in the dataFEED OPC Suite Configurator. The default database specified in the ODBC manager will be used in the Configurator and tables from that database will be automatically displayed in the Database Action Wizard - Database Table page.

For better **PostgreSQL support** (see also section below) the user needs to add a special command to the Windows ODBC Driver settings or to the initialization script on this wizard page, when using tables in actions:

```
SET SEARCH_PATH to public, custom_schema_1,..., custom_schema_n;
```

for the ODBC driver, and

```
SET SEARCH_PATH = public, custom_schema_1,..., custom_schema_n;
```


© Softing Industrial Automation GmbH

for the wizard page.

Setting the schema list only in the ODBC driver is advised.

Note: If the above schema list is given only in the wizard page, its settings will not be applied until the Runtime is started, so the Configurator will not take advantage of it. If the schema list is specified in the ODBC driver settings, then it will be applied as soon as the settings are saved in the driver. Thus, the Configurator can show database content (tables, stored procedures) only from the specified schemas.

Please read the PostgreSQL documentation regarding schemas.

This command specifies to PostgreSQL which schema it will search for the tables in, because dataFEED OPC Suite is not reading or saving this information into the configuration and **PostgreSQL** requires them by default.

Syntax Settings

This section contains a setting introduced for better support of PostgreSQL case sensitive naming convention, regarding how database tables and their columns are used.

Note: In PostgreSQL table and column names can be specified surrounded with quotation marks ("), which results in them being considered as case sensitive entities. This means that a table created with the name *"ExampleTable"* will not be the same table as another one created without quotation marks, i.e. Example*Table*. In fact, the latter one will be created as *exampletable*. The same rule applies for the column names of a table. Also, a table can have column names specified with and without quotation marks when creating it.

When enabled, the Enable support for case sensitive naming in database (PostgreSQL only) option enables the application to be more compliant with this feature in the database, by using the following rules:

- when listing the tables from the database, dataFEED OPC Suite will list both case sensitive and case insensitive tables and their columns
- as noted above, there may be tables with the same name, but with different-case letters (they will be regarded as different tables)
- when saving the configuration, the dataFEED OPC Suite will save the table and column names surrounded by quotation marks for the Runtime to be able to correctly specify to PostgreSQL which table it is using in an action
- when loading the configuration, the application will remove the quotation marks internally so that the user will not see them on the user interface
- e.g.: INSERT INTO "tb_0_OPCDAV3" ("ID_OPCDAV3","INFO","BIT","BYTE","CHAR","Word","DWORD","Integer","DInteger","REAL","STRING","UI8","I8","R8") VALUES (...) - this is how a saved command will be executed when the option is enabled, and the table was created containing lower and upper-case letters in its name and its columns

- the option is only applied to Insert command action; the user needs to add quotation marks when creating a Stored procedure action or a general SQL command action

Other rules:

- when the option is disabled (by default) the application may not display all the tables and their columns correctly
- the list of tables may be incomplete
- the list of columns may be incomplete or empty
- it is not advised to use connections to PostgreSQL with the option disabled
- it is not advised to use other types of database connections with the option enabled (PostgreSQL only)

Value Operations

The Database Connection may be configured to replace all TRUE BOOLEAN values with "1" instead of "-1" by marking the *Replace BOOLEAN value "-1" with "1" for TRUE* checkbox in the *Value Operations* group box.

Note: The *Replace BOOLEAN value "-1" with "1" for TRUE* checkbox is marked by default for new Database Definitions created with dataFEED OPC Suite v4.11 and later, and unmarked for existing Database Definitions from configurations created with earlier dataFEED OPC Suite versions.

The Database Connection may also be configured to replace "," with ";" for separate elements into an array. The default separator is comma.

Note: The *Replace "," with ";" for separate elements into an array* checkbox is unmarked for all existing Database Definitions versions.

Press *Finish* to complete the wizard or quit the wizard by pressing the *Cancel* button.

<<< Database Wizard - Authentication

7.3.3.5.4 Database Action Wizard

With the Database Action Wizard you can create a new Database Action definition or modify an existing one:

© Softing Industrial Automation GmbH

- Obtain general information about the Database Action Wizard @ Database Action Wizard - Welcome
- Define the database action type @ Database Action Wizard - Action Type
- Select the database table the action will be executed on @ Database Action Wizard - Database Table
- Add data sources to the table columns involved in the transaction @ Database Action Wizard - Table Columns
- Define the SQL stored procedure call statement @ Database Action Wizard - Stored Procedure
- Define the SQL statement to be executed @ Database Action Wizard - SQL Statement
- Choose the database action trigger type @ Database Action Wizard - Trigger Type
- Pick the database action trigger item @ Database Action Wizard - Trigger Item
- Select the timer trigger @ Database Action Wizard - Timer Selection
- Configure basic timer identity settings @ Database Action Wizard - Timer Identity
- Decide whether the timer should be interval-based or scheduled @ Database Action Wizard - Timer Type
- Specify the timer interval @ Database Action Wizard - Interval Timer Settings
- Define the timer schedule @ Database Action Wizard - Schedule Timer Settings
- Configure the execution condition and the update rate @ Database Action Wizard - Action Settings
- Select the variable trigger @ Database Action Wizard - Variable Trigger Selection

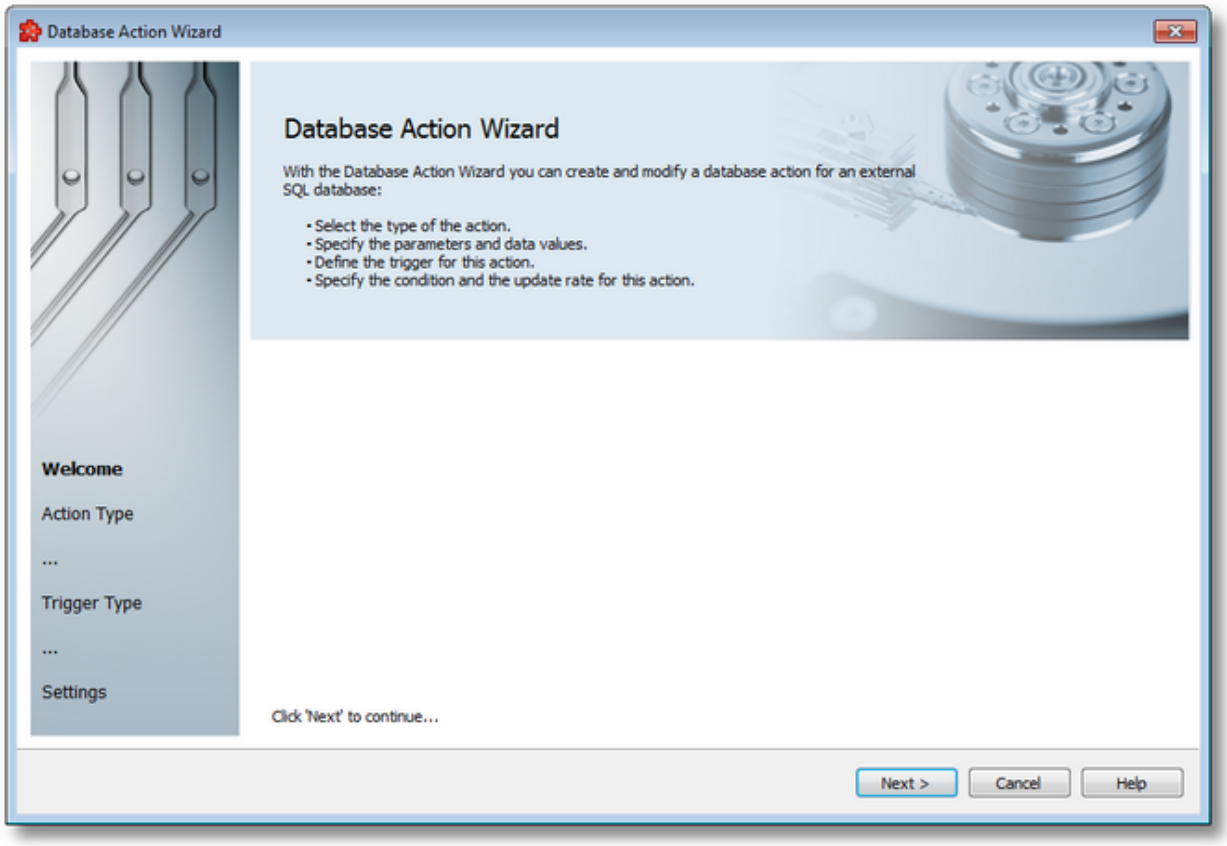
Note: A Database Action Wizard flow will not include all the pages listed above. A specific branch will be followed during the wizard, according to the selections performed in some of the pages.

The Welcome wizard page describes the procedure for creating a new Database Action definition.

The steps to be performed in the following pages are:

- select one of three possible actions that can be defined for a database: *Insert Into*, *SQL Stored Procedure* and *General SQL Statement*.

- specify the statements of the entire action using free texts and placeholders for *SQL Stored Procedures* and *General SQL Statements* or associations between placeholders and table columns for Insert Into actions.
- define the trigger type and trigger settings for the current action.
- specify the subscription update rate and the execution condition of the current action on the action settings page.

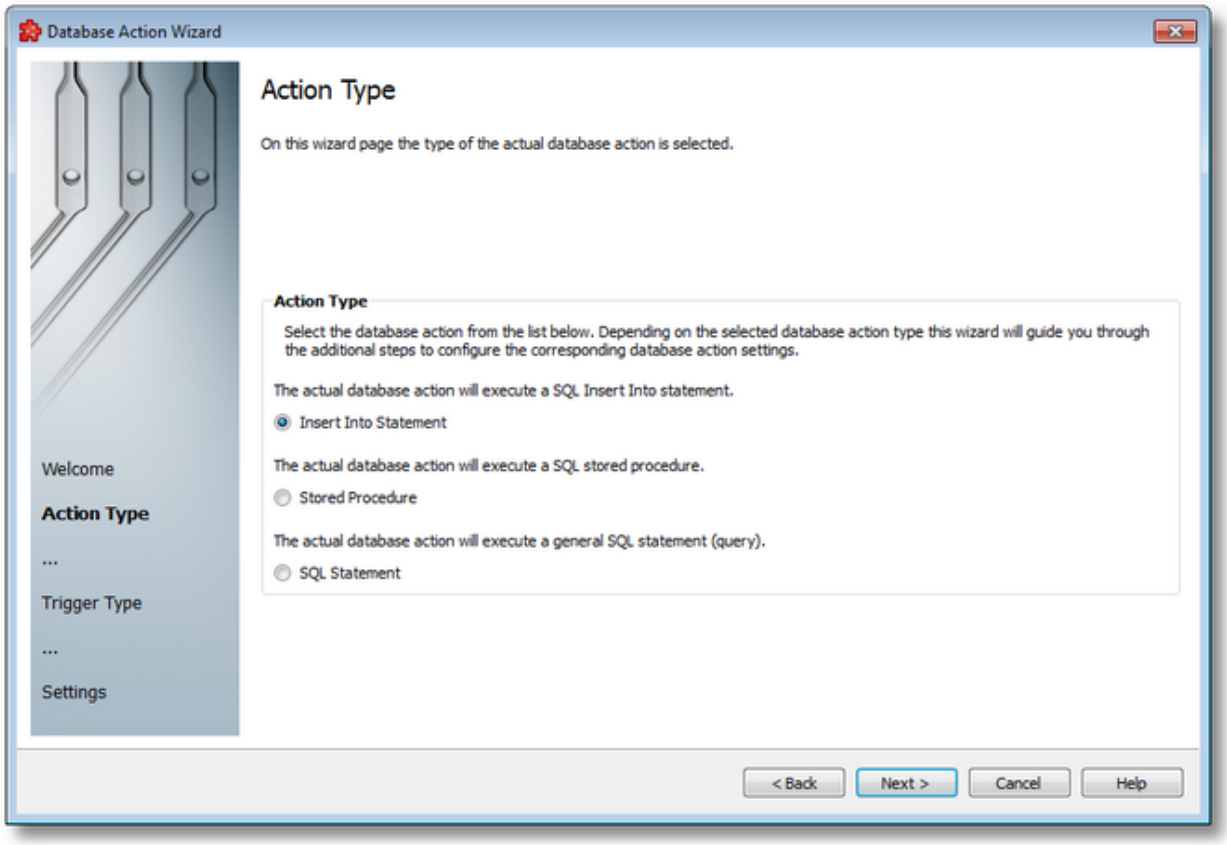


To continue with the database definition use the *Next* button or abort the operation by pressing the *Cancel* button.

Database Action Wizard - Action Type >>>

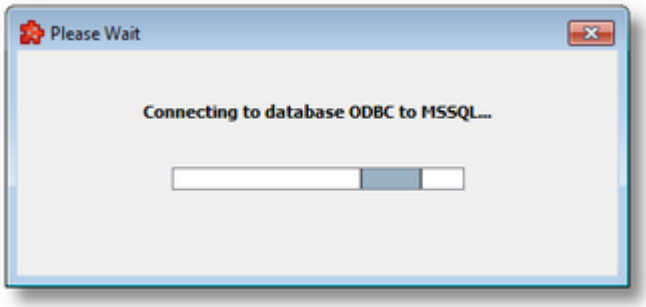
The Action Type wizard page describes the different actions you can choose when defining a new Database Action. It contains three radio buttons for the three types of database actions: Insert Into, SQL Store Statement and General SQL Statement plus a short description of each.

© Softing Industrial Automation GmbH



Choose the type of action you want to perform on the database:

- *Insert Into* - only *SQL Insert Into* statements can be created if this option is selected. It offers a two-step interactive method for setting up all the necessary configuration. The first step allows selecting the table name and the second step offers the possibility to associate the OPC data placeholders with column names inside database tables. When you press the *Next* button, a database connection will be initiated to read the table columns; a waiting dialog will be displayed until the database query completes:



- *Stored Procedure* - allows executing the created stored SQL procedures with some parameters which can be defined as placeholders. For this purpose a free form SQL text editor is provided.
- *SQL Statement* - allows defining a specific SQL statement by adding specific SQL text in the above-mentioned free form SQL text editor.

Note: It is possible that some of the defined database actions might appear as different ones after the configuration is reloaded into dataFEED OPC Suite Configurator (for example a General SQL action acting as an insert into statement will appear as an Insert Into statement).

To proceed with the wizard and continue to define the selected type of database action use the *Next* button, to go back to the previous wizard page use the *Back* button, or quit the wizard and discard the changes by pressing the *Cancel* button.

Depending on the selected *Action Type* the next wizard page will be:

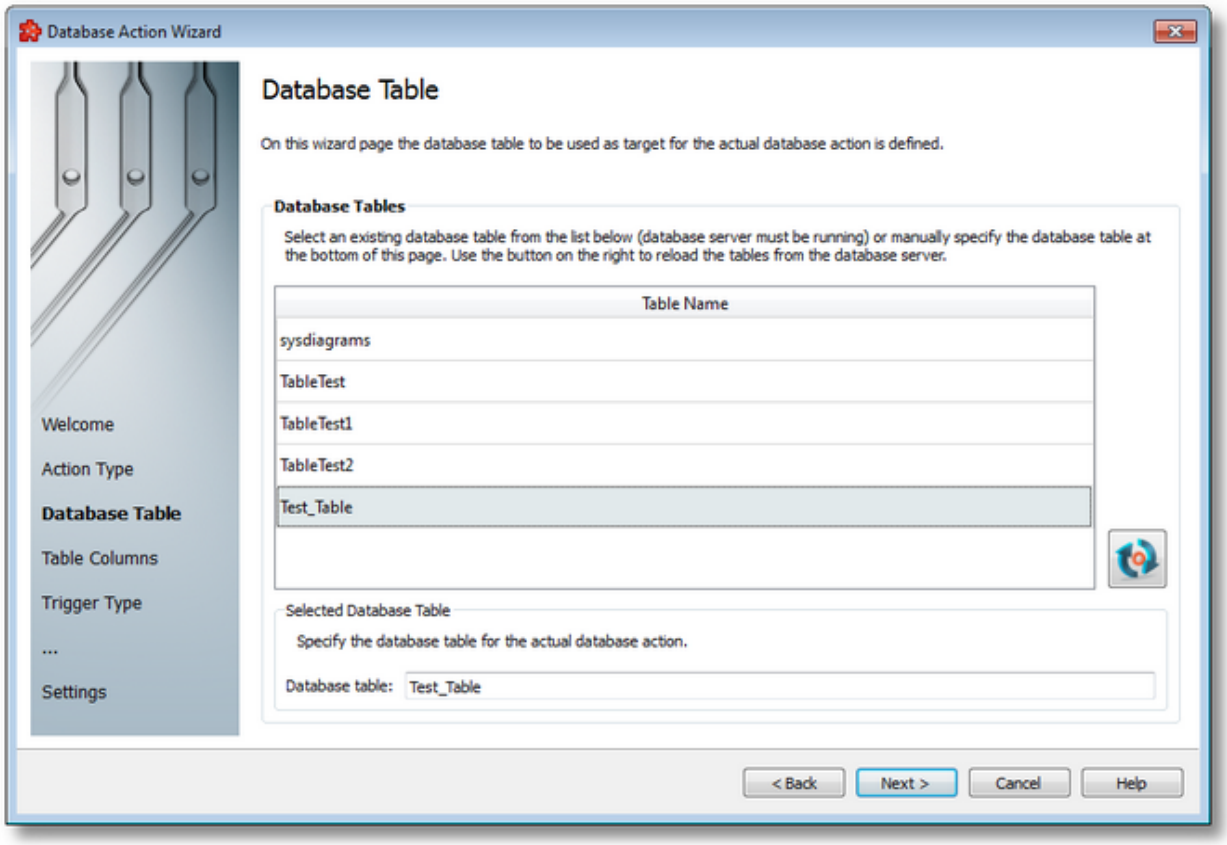
- Database Action Wizard - Database Table - if the selected action is *Insert Into Statement*,
- Database Action Wizard - Stored Procedure - if the selected action is *Stored Procedure*,
- Database Action Wizard - SQL Statement - if the selected action is *SQL Statement*.

The description of the Database Action Wizard will continue with the wizard pages for defining an *Insert Into Statement*.

<<< Database Action Wizard - Welcome Database Action Wizard - Database Table >>>

The *Database Table* page allows selecting the desired database table in order to create the new *Insert Into* action. When this page is shown a *Refresh* operation is automatically performed on the database for updating the *Table Name* list with the latest information from the database.

© Softing Industrial Automation GmbH

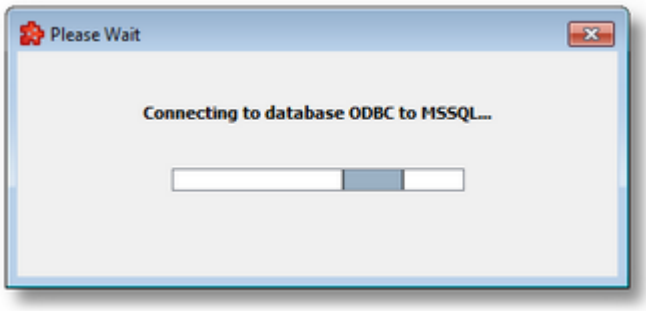


The *Database Table* section contains:

- the *Table Name* list containing the tables queried from the database
- the *Database table* editable text field which is automatically populated with the selected table name. The desired table name can also be manually typed. The typed table name will be automatically selected in the above list. Before the next page is displayed after the *Next* button is pressed, the selected database table will be queried for its column names, data types and whether they are mandatory or not.
- the *Reconnect* button  re-queries the database for retrieving the latest modifications regarding the existing tables.

Note: dataFEED OPC Suite displays the table names encoded in UTF-8. The Unicode characters are thus fully supported for Database table names.

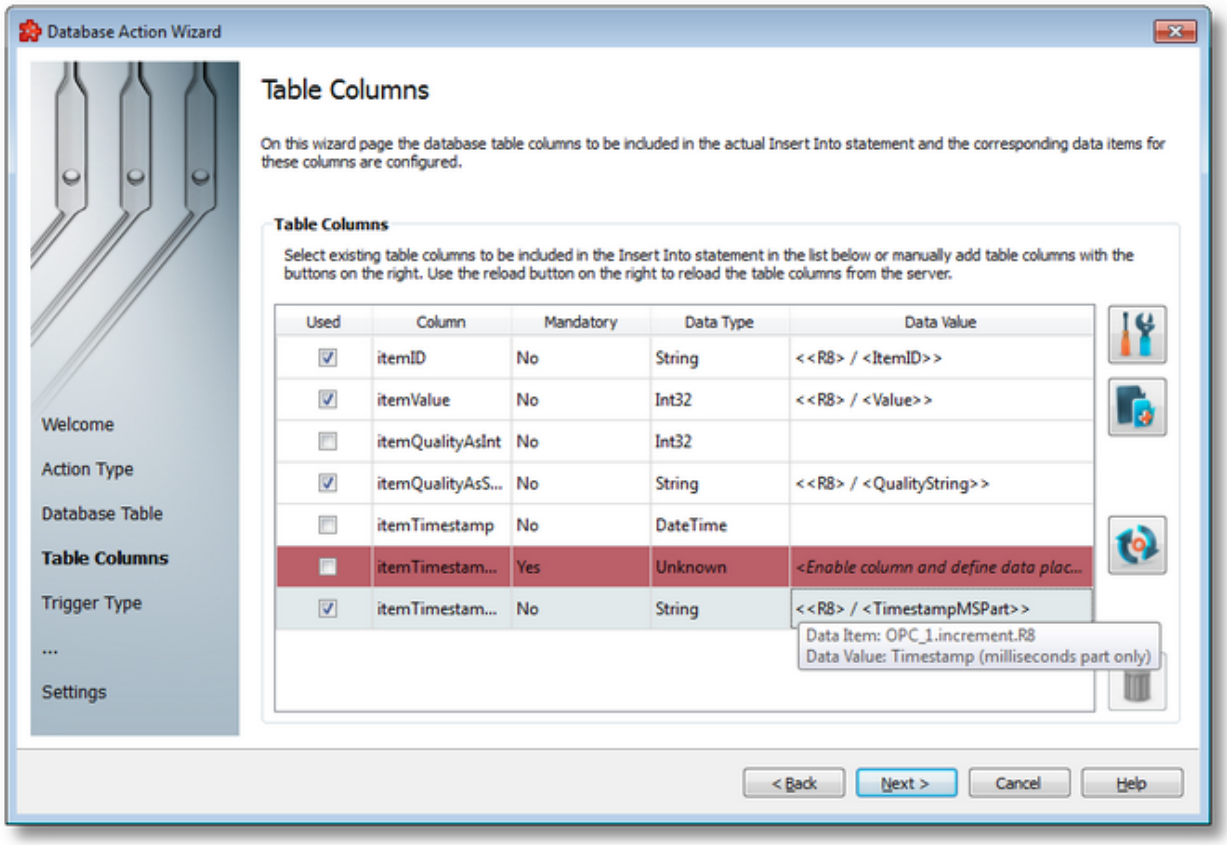
To advance in the wizard and define the association between OPC data placeholders and the corresponding table columns press the *Next* button. A database query will be performed to retrieve the selected table's column names:



To go back to previous wizard page and select a different database action type use the *Back* button. To quit the wizard and discard all the changes press the *Cancel* button.

<<< Database Action Wizard - Action Type Database Action Wizard - Table Columns >>>

The *Table Column* wizard page allows selecting the desired table columns that compose the *Insert Into* SQL statement action. A database query will be performed to retrieve the selected table's column names, data types and whether they are mandatory or not.



© Softing Industrial Automation GmbH

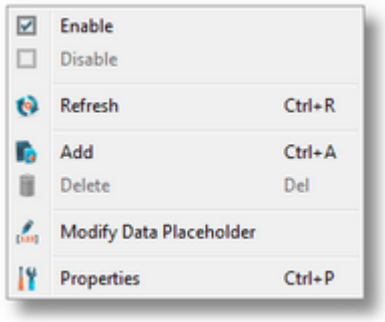
The INSERT INTO action editor offers a convenient way to edit an insert into SQL statement. It displays an interactive table grid containing the table column names as rows with their respective data associations. The insert into action editor table has the following fields:

- *Used* - column which determines whether the respective column will receive any data from the incoming transaction. When you select the checkbox in this cell the Data Value Placeholder Wizard will open. In this wizard you can associate the table column name with a placeholder. When the checkbox is unchecked this association will be lost and the defined placeholder will automatically be deleted.
- *Column* - column name in the selected database definition for storing the item/placeholder value.
- *Mandatory* - indicates whether the respective table column is compulsory or not. It can contain two values - *Yes* or *No* - denoting whether an insert action will fail or not. When the *Used* column for the respective row is not selected and the value of this column is *Yes*, the background color of the Database Action row will be red, emphasizing that an insert into action will fail if the respective table column is not included in the transaction. The background color will not be altered if the value of this field is *No* thus proving that the table column is not necessary. The row's background color will not be altered, either, if the field is mandatory and the *Used* column is selected.
- *Data Type* - column displaying the data type of the database table column.
- *Data Value* - column in the interactive table which displays the user defined placeholder. If the *Used* column is not checked this column will contain no placeholder reference.

The following operations can be performed on the *Table Columns* by using the buttons on the right, the keyboard shortcuts or the context sensitive menu:

- Properties of the selected column using the *Properties of the selected database column* button  , pressing the keyboard shortcut CTRL+P or selecting the *Properties* action from the context menu.
- Add a new table column using the *Add Database Column* button  , pressing the keyboard shortcut CTRL+A or selecting the *Add* action from the context menu.
- Retrieve the latest column names from the database table by pressing the *Refresh Database Columns* button  , pressing the keyboard shortcut CTRL+R or selecting the *Refresh* action from the context menu.
- Remove the selected table column using the *Delete Selected Database Column* button  , pressing the DELETE / DEL key or selecting the *Delete* action from the context menu.
- Enable the desired table column by checking the checkbox in the *Used* column of the respective *Database Action* row or selecting *Enable* action from the context menu.
- Disable the desired table column by unchecking the checkbox in the *Used* column of the respective *Database Action* row or selecting *Disable* action from the context menu.
- Modify the data placeholder by selecting the *Modify Data Placeholder* action from the context menu.

The available context menu is shown below:



The text and background colors for *Database Action* rows comply the following rules:

- The selected column's background color is always gray.
- An unselected column's background color is red if this column is not included in the action (the checkbox in the *Used* field is unchecked) and the value of the *Mandatory* field is *Yes* (the database table column must exist in the insert action).
- The unselected online columns always have a white background color and a black text color.
- The unselected offline columns always have a white background color and a red text color.

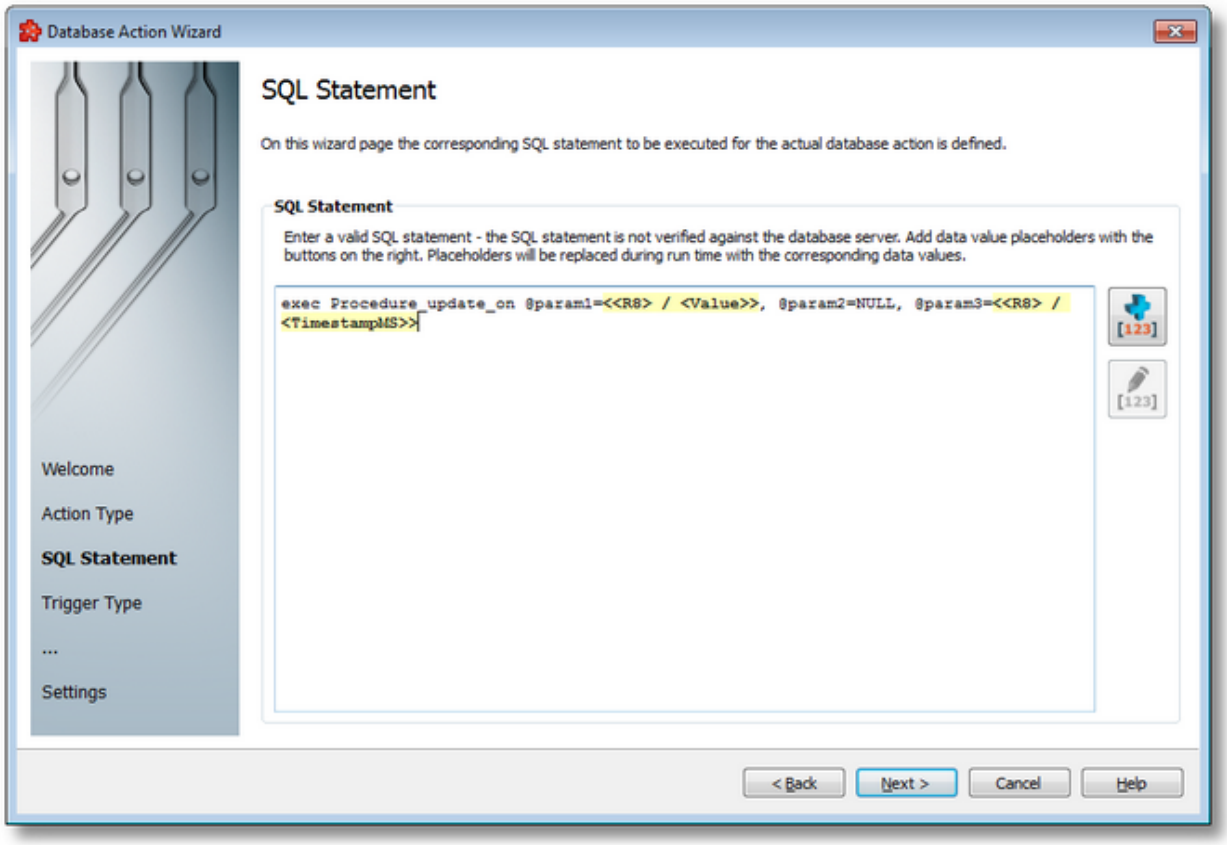
To proceed forward and define the trigger type use the *Next* button, to go back to the previous wizard page and select a different type of database action use the *Back* button. To exit the wizard click the *Cancel* button.

<<< Database Action Wizard - Database Table Type Database Action Wizard - Trigger >>>

This wizard page allows defining a SQL Stored Procedure call statement using the action editor. Certain operations are available using the buttons on the right, the context sensitive menu or the keyboard shortcuts:

- Insert a data value placeholder using the *Insert Data Value Placeholder* button  , pressing the keyboard shortcut CTRL+I or selecting the *Insert Data Placeholder* action from the context menu.
- Modify an existing data value placeholder using the *Modify Data Value Placeholder* button  , pressing the keyboard shortcut CTRL+M or selecting the *Edit Data Placeholder* action from the context menu.
- Cut, copy, paste or delete text with or without placeholders by selecting the *Cut*, *Copy*, *Paste* or *Delete* actions from the context menu or using the respective keyboard shortcuts CTRL+X, CTRL+C, CTRL+V and the DELETE / DEL key.

© Softing Industrial Automation GmbH



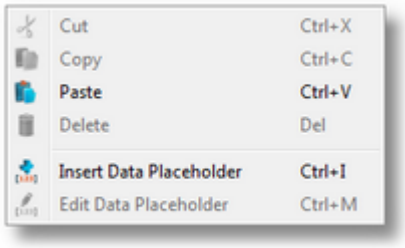
Free text blocks can be added in the action editor, including special characters. The database-specific action editor does not support new line and tab characters, however. The above screenshot shows a stored procedure call where the parameters @param1 and @param2 each have a placeholder as their value:

```
exec Procedure_update_on @param1=<<R8> / <Value>>, @param2=NULL, @param3=<<R8>
```

Note: Please keep in mind that the above example is SQL server specific syntax (Microsoft SQL Server). In the action editor you need to enter for each server in database specific syntax. Or you can use the "Database Read" stored procedure functionality which is more advanced and has drag-drop functionality for stored procedure call composition.

Or you can use the “Database Read” store procedure functionality which is more advanced and has drag-drop functionality for stored procedure call composition.

An example of the context sensitive menu is shown below:



Note: The *Cut*, *Copy* and *Paste* actions use the Windows Clipboard for data transfer, thus allowing data to be transferred to/from other applications (e.g. the textual representation of the action editor content can be used in external applications and text from external application can be used in the action editor).

Attention: Please keep in mind that placeholders copied to clipboard (through the *Copy* or *Cut* commands) will only be pasted back as placeholders if the clipboard has not been tampered with, otherwise the placeholders will be pasted as plain text.

Attention: The placeholders visible in the action editor are aliases used only for displaying the placeholders. The real placeholder form is visible in the subscription file. Example: *<<R8> / <Value>>* placeholder from the action editor is serialized in the subscription file as: *#%-ItemValue-%increment.R8_%Current-%#*. Do not manually modify the subscription file by inserting placeholders in the action editor template.

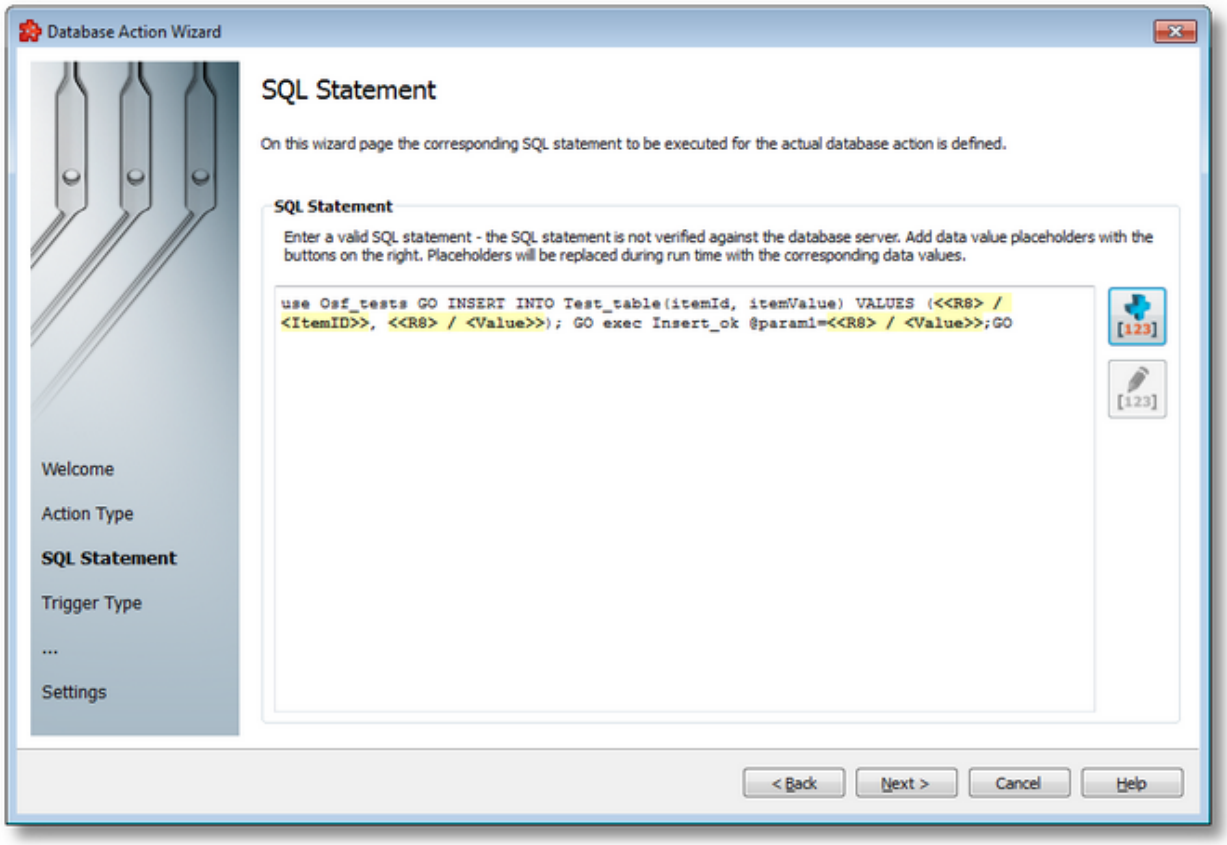
To proceed forward and define the trigger type use the *Next* button, to go back to the previous wizard page and select a different type of database action use the *Back* button. To exit the wizard click the *Cancel* button.

<<< Database Action Wizard - Action Type Database Action Wizard - Trigger Type >>>

This wizard page allows defining the SQL statement text for the current database action with the help of the action editor. Certain operations are available using the buttons on the right, the context sensitive menu or the keyboard shortcuts:

- Insert a data value placeholder using the *Insert Data Value Placeholder* button  , pressing the keyboard shortcut CTRL+I or selecting the *Insert Data Placeholder* action from the context menu.
- Modify an existing data value placeholder using the *Modify Data Value Placeholder* button  , pressing the keyboard shortcut CTRL+M or selecting the *Edit Data Placeholder* action from the context menu.
- Cut, copy, paste or delete text with or without placeholders by selecting the *Cut*, *Copy*, *Paste* or *Delete* actions from the context menu or using the respective keyboard shortcuts CTRL+X, CTRL+C, CTRL+V and the DELETE / DEL key.

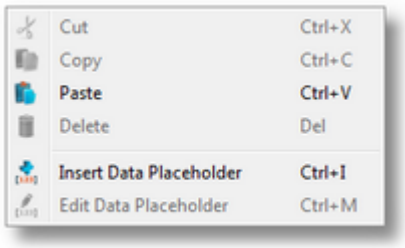
© Softing Industrial Automation GmbH



Free text blocks can be added in the action editor, including special characters. The database-specific action editor does not support new line and tab characters, however. The above example shows a call to an initialization statement followed by an Insert Into and then by a SQL stored procedure call:

```
use Osf_tests GO INSERT INTO Test_Table (itemId,itemValue)  VALUES (<<R8> / <ItemID>>
GO exec Insert_ok @param1=<<R8> / <Value>>; GO
```

An example of the context sensitive menu is presented below:



Note: The *Cut*, *Copy* and *Paste* actions use the Windows Clipboard for data transfer, thus allowing data to be transferred to/from other applications (e.g. the textual representation of the action editor content can be used in external applications and text from external application can be used in the action editor).

Attention: Please keep in mind that placeholders copied to clipboard (through the *Copy* or *Cut* commands) will only be pasted back as placeholders if the clipboard has not been tampered with, otherwise the placeholders will be pasted as plain text.

Attention: The placeholders visible in the action editor are aliases used only for displaying the placeholders. The real placeholder form is visible in the subscription file. Example: *<<R8> / <Value>>* placeholder from the action editor is serialized in the subscription file as: *#%-ItemValue-%increment.R8_%Current-%#*. Do not manually modify the subscription file by inserting placeholders in the action editor template.

To proceed forward and define the trigger type use the *Next* button, to go back to the previous wizard page and select a different type of database action use the *Back* button. To exit the wizard click the *Cancel* button.

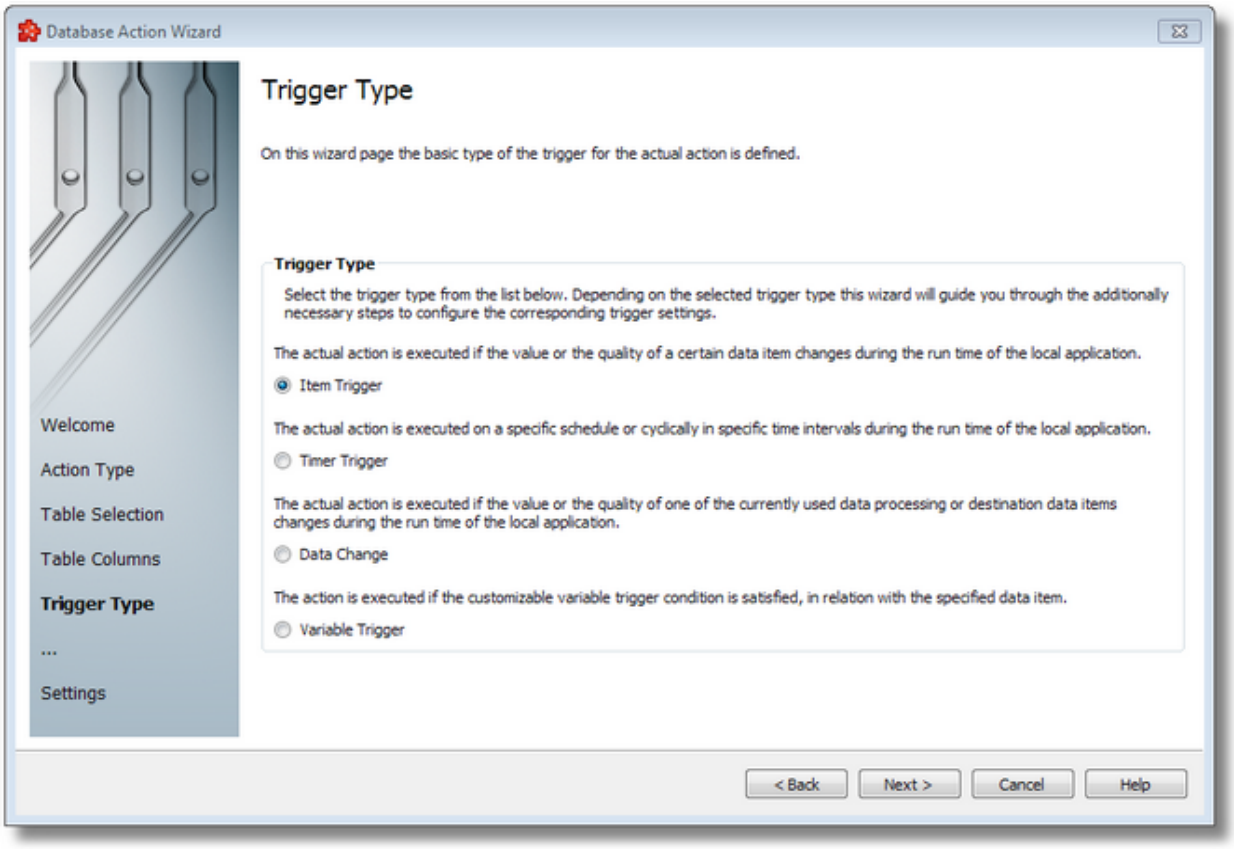
<<< Database Action Wizard - Action Type Database Action Wizard - Trigger Type >>>

The Trigger Type wizard page provides the possibility to select the trigger type for the actual action. The following trigger types are available:

- *Item Trigger* - This trigger is fired when a data change for the selected item occurred. A data change means that the value or the quality of the item changed.
- *Timer Trigger* - This trigger is fired at a specified time or interval.
- *Data Change* - This trigger is fired when a data change occurred in at least one of the items used in any of the consumer add-ons.
- *Variable Trigger* - This trigger is fired when a data change for the items in the variable conditions occurs. A data change means that the value or the quality of the item changed.

Attention: The trigger only initiates the database action execution. The action will be subsequently validated, however, and might not be executed after all.

© Softing Industrial Automation GmbH



To proceed with the wizard and continue to define the selected type of trigger (or to advance to the last wizard page when the *Data Change* trigger type is selected) use the *Next* button, to go back to the previous wizard page use the *Back* button, or discard the wizard by pressing the *Cancel* button.

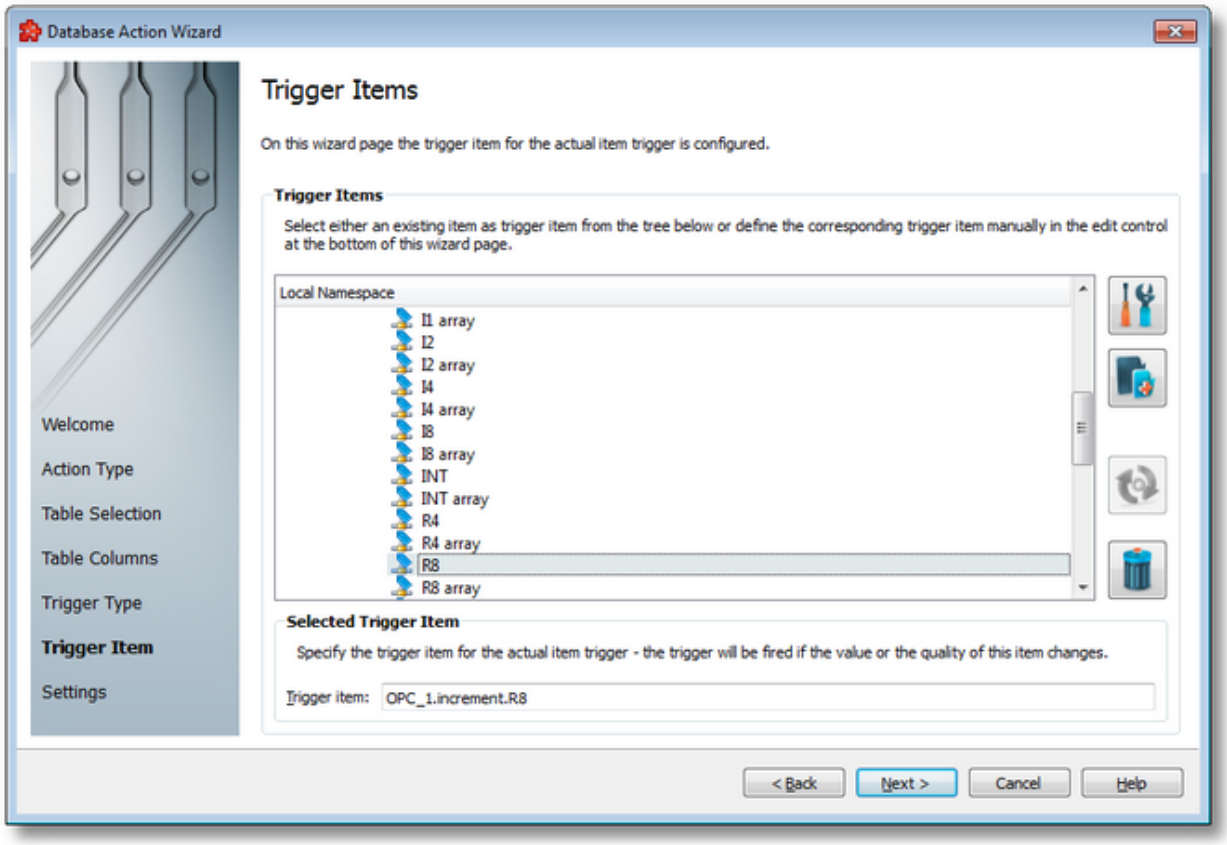
Depending on the selected *Trigger Type* the next wizard page will be:

- Database Action Wizard - Trigger Item - if the selected trigger type is *Item Trigger*,
- Database Action Wizard - Timer Selection - if the selected trigger type is *Timer Trigger*,
- Database Action Wizard - Action Settings - if the selected trigger type is *Data Change* (unlike the above options, no other parameters need to be configured here).
- Database Action Wizard - Variable Trigger - if the selected trigger type is *Variable Trigger*

The description of the Database Action Wizard will continue with the wizard pages for configuring an *Item Trigger*.

<<< Database Action Wizard - Table Columns Database Action Wizard - Trigger Item >>>

The *Trigger Item* wizard page is shown if the selected trigger type is *Item Trigger*. Here you can configure the actual trigger item:



This wizard page contains two separate sections:

- *Trigger Items* section, which allows selecting an item trigger or managing the Local Namespace tree.
- *Selected Trigger Item* section, which shows the item ID of the selected trigger item and also allows typing an item ID, thereby selecting this item in the above list.

Trigger Items section

The entire namespace of the current configuration is shown as a namespace tree. The name of the root node is composed by appending the instance name to the *Local Namespace* keyword. The first level of the namespace contains:

- A node for each server connection, named by the server connection items prefix.
- A node for System Items.
- A node for Local Items.

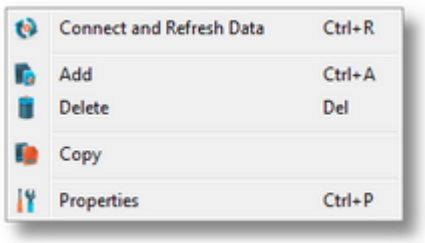
The local namespace tree offers the possibility to explore the entire namespace of the local dataFEED OPC Suite application in order to select a trigger item. After the selection is made, the *Trigger item* edit box is populated with the selected item ID.

The entire namespace of the local dataFEED OPC Suite application can be managed through this wizard page by performing certain operations using the buttons on the right, the context sensitive menu or the keyboard shortcuts:

© Softing Industrial Automation GmbH

- Open the item properties dialog using the *Properties of the selected item* button  , pressing the keyboard shortcut CTRL+P or selecting the *Properties* action from the context menu. Depending on the type of the selected item, a special Item Properties dialog opens:
    - If the selected item is under Local Items the Modify Local Item dialog is opened.
    - If the selected item is under a dynamic server connection the Modify Dynamic Server Item dialog is opened.
    - If the selected item is under a static server connection the Modify Static Server Item dialog is opened.
- Add a new item using the *Add a new item*  button, pressing the keyboard shortcut CTRL+A or selecting the *Add* action from the context menu. Depending on the type of the selected node, a special Add Item dialog opens:
    - If the selected node is under Local Items the Add Local Item dialog is opened.
    - If the selected node is under a dynamic server connection the Add Dynamic Server Item dialog is opened.
    - If the selected node is under a static server connection the Add Static Server Item dialog is opened.
- Copy the selected item to clipboard by selecting the *Copy* action from the context menu.
- Paste the previously copied item from clipboard by selecting the *Paste* action from the context menu. Items originating from the Local Items cannot be pasted under server connections, and vice-versa; an error message will be displayed in both cases. If an item is pasted in a static (filtered) server connection and the item does not exist on the server an error message is displayed.
- Connect to the server and refresh the data items by using the *Refresh data items*  button, pressing the keyboard shortcut CTRL+R or selecting the *Connect and Refresh Data* action from the context menu. The namespace of the selected server connection is recreated; if manually added items not referenced in the actual configuration exist, those items are deleted from the namespace.
- Delete the selected item and all of its child items by using the Delete data items button  , pressing the DELETE / DEL key or selecting the Delete action from the context menu. Items which are included in any Data Processing / Data Destination add-ons will not be deleted but an error message will be displayed instead.

An example of the context menu can be seen below:



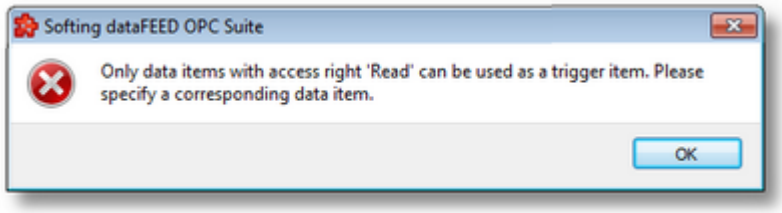
Selected Trigger Item section

The edit box in this section is populated with the item ID after an item from the Local Namespace tree is selected.

You can also manually type the ID of the item in the *Trigger item* edit box and thus have the above tree automatically select a valid server item.

Note: If the item ID is typed and the specified item exists in the Local Namespace, it will be used as the trigger item. The item will only be created if it could not be localized in the Local Namespace. In that case the newly created item will be used as the trigger item.

Note: If the item corresponding to the typed item ID could not be created, an error message will be displayed requesting the input of a new, valid item.



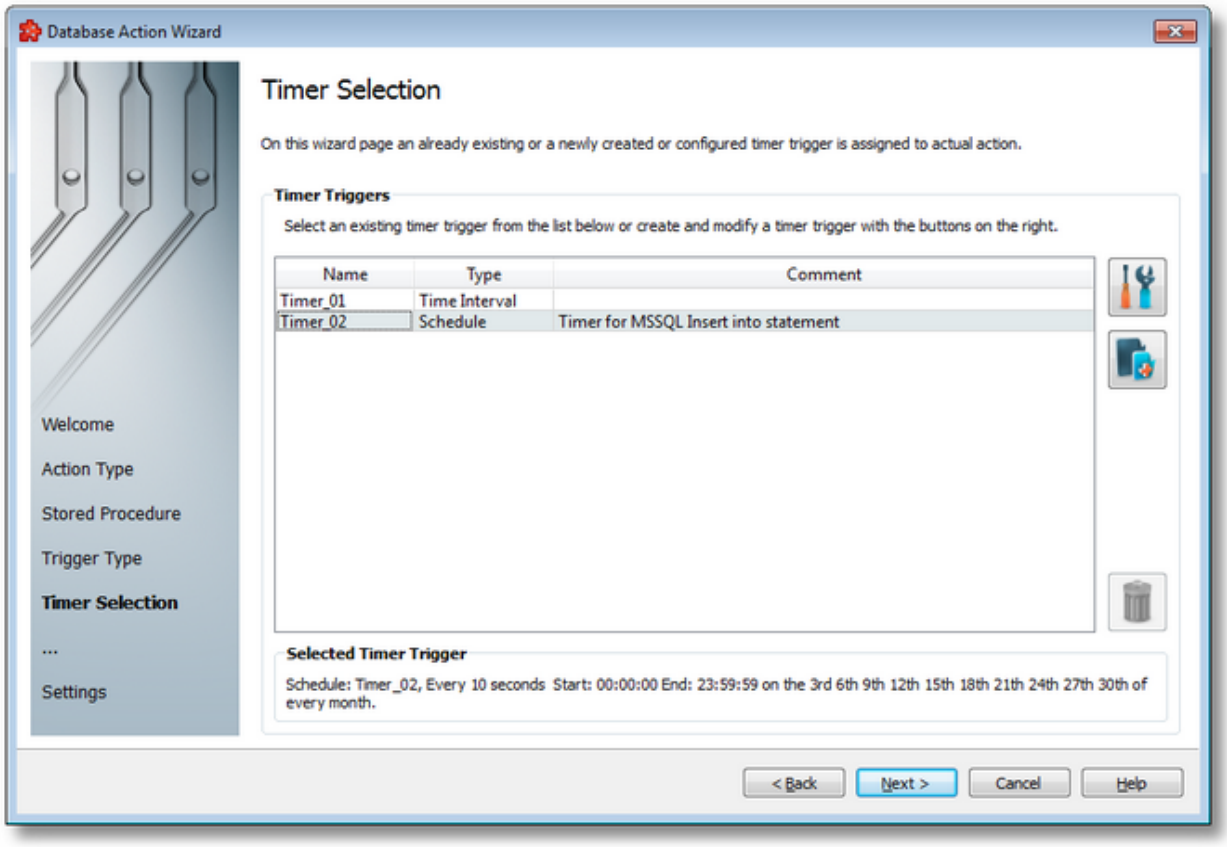
Attention: If the item corresponding to the typed item ID is valid and resides under a static server connection, the respective item will be created in the filter file definition.

To proceed with the wizard and continue to define the action settings use the *Next* button. To go back to the previous wizard page use the *Back* button, or exit the wizard and discard all the changes by clicking the *Cancel* button.

<<< Database Action Wizard - Trigger Type Database Action Wizard - Action Settings >>>

The Timer Selection wizard page is shown only if the selected trigger type is Timer Trigger. Here you can configure the actual timer trigger.

© Softing Industrial Automation GmbH

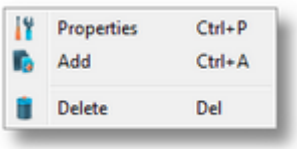


The current page allows selecting a previously defined timer trigger and managing the timer list by using the buttons on the right, the context sensitive menu or the keyboard shortcuts:

- Change the properties of the selected timer by using the *Properties* button  , pressing the keyboard shortcut CTRL+P or selecting the *Properties* action from the context menu.
- Add a new timer by using the Add timer trigger button  , pressing the keyboard shortcut CTRL+A or selecting the Add action from the context menu.
- Delete the selected timer trigger by using the Delete selected timer trigger button  , pressing the DELETE / DEL key or selecting the Delete action from the context menu.

Note: If a timer trigger has associated actions it cannot be deleted.

An example of the context sensitive menu can be seen below:



Note: Information about the selected timer is shown in the *Selected Timer Trigger* area at the bottom of the page.

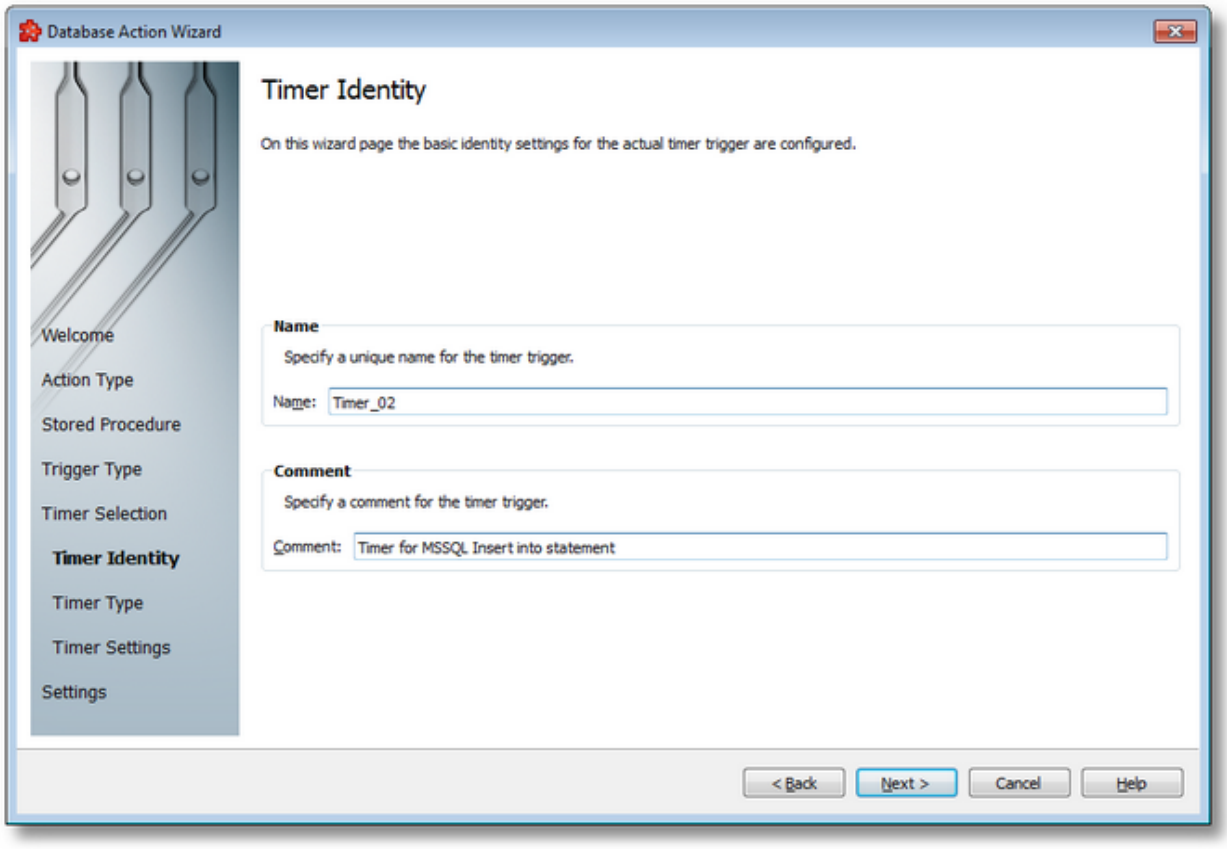
A sub-wizard that allows adding or modifying timer triggers can be launched by selecting to change an existing timer trigger or to add a new one. The first page of this sub-wizard is Database Action Wizard - Timer Identity.

To proceed with the *Database Action Wizard* and define the action settings press the *Next* button. To return to the previous wizard page use the *Back* button, or exit the wizard by clicking the *Cancel* button.

The wizard will continue straight to the *Action Settings* page as the selected timer has already been defined.

<<< Database Action Wizard -Trigger Type Database Action Wizard - Action Settings >>>

The Timer Identity wizard page provides the possibility to assign a name and a comment to the timer trigger.



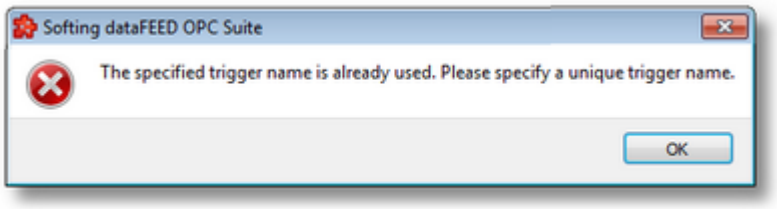
Note: When adding a new timer trigger, the timer name will be automatically generated according to the format: *Timer_number*, where the *number* part is an auto-incremented integer.

When editing an existing timer, the timer's defined name and comment will be filled in the two edit boxes allowing you to modify these attributes.

© Softing Industrial Automation GmbH

Attention: The name of the *Timer Trigger* must be unique among all triggers and cannot be empty.

If the newly added timer trigger has the same name as an existing trigger, an error message box will be displayed and a new trigger name must be provided:



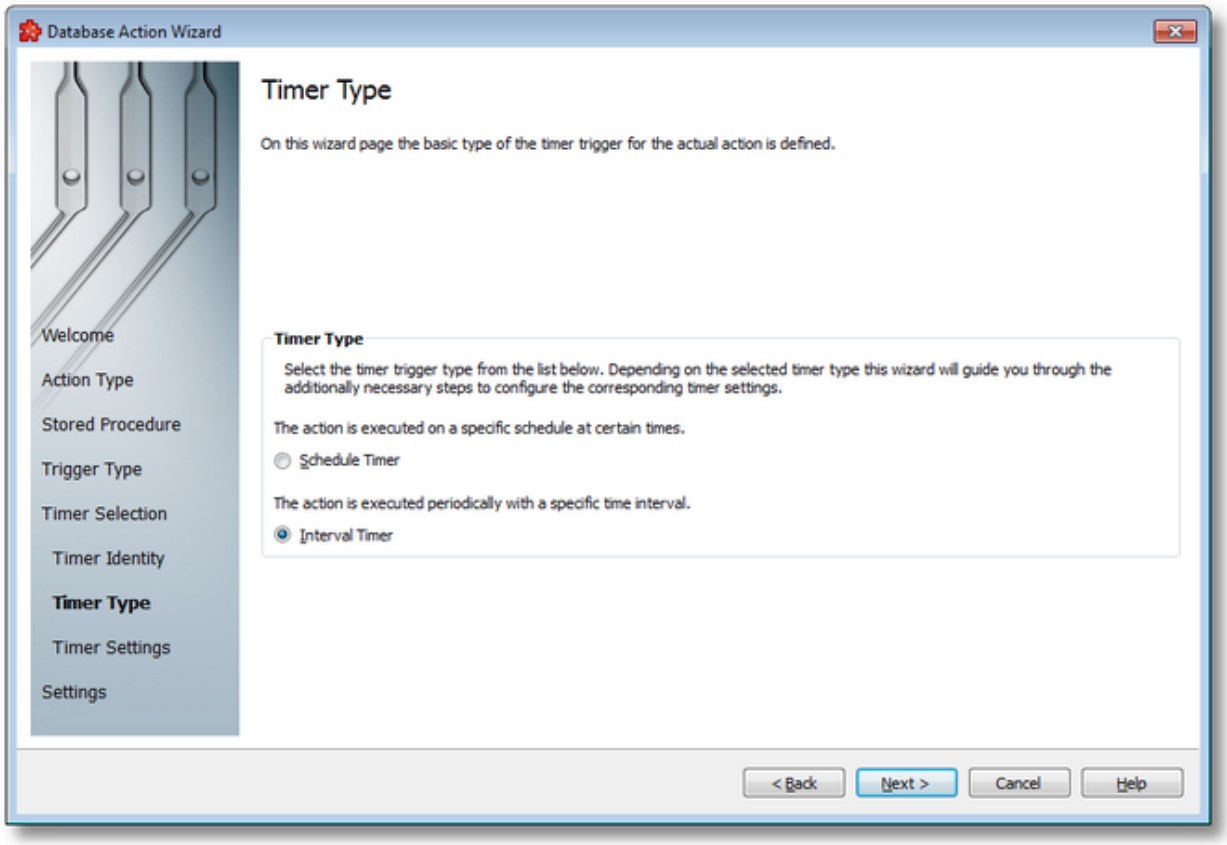
The *Comment* field of the *Timer Identity* wizard page is optional; a Timer Trigger can also be added without any comment.

Continue to the next wizard page to choose the timer type by using the *Next* button, go back to the previous wizard page to select another timer trigger by pressing the *Back* button, or cancel the entire *Database Action Wizard* by pressing the *Cancel* button.

<<< Database Action Wizard - Timer Selection Database Action Wizard - Timer Type >>>

This wizard page allows choosing the type of the timer trigger. The following types of timer are available:

- *Interval Timer*
- *Schedule Timer*



Interval Timer

The Interval Timer type allows defining a specific interval of time (in milliseconds) between the timer fire events. The time interval can be configured on the Database Action Wizard - Interval Timer Settings wizard page.

Schedule Timer

The Schedule Timer type allows defining a specific date and time when the timer should fire. The schedule can be configured on the Database Action Wizard - Schedule Timer Settings wizard page.

Note: The *Timer Type* of an existing *Timer Trigger* cannot be changed.

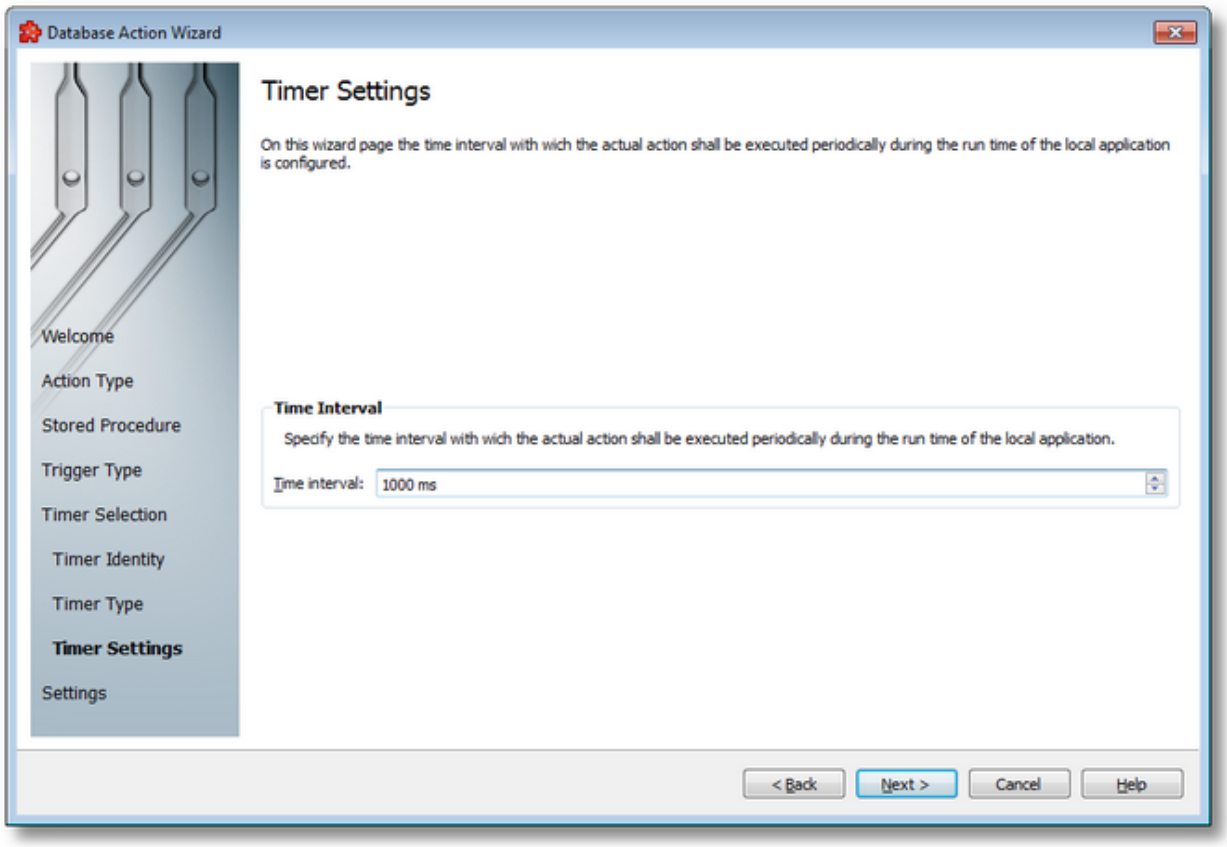
To continue defining the specific timer properties press the *Next* button, to return to the *Timer Identity* wizard page use the *Back* button, or exit this sub-wizard (thereby aborting the *Database Action Wizard*) by pressing the *Cancel* button.

The description of the Database Action Wizard will continue with the wizard pages for configuring a *Schedule Timer*.

© Softing Industrial Automation GmbH

<<< Database Action Wizard - Timer Identity
Database Action Wizard - Schedule Timer >>>

The Interval Timer Settings wizard page allows configuring the time interval at which the current action should be (periodically) executed.

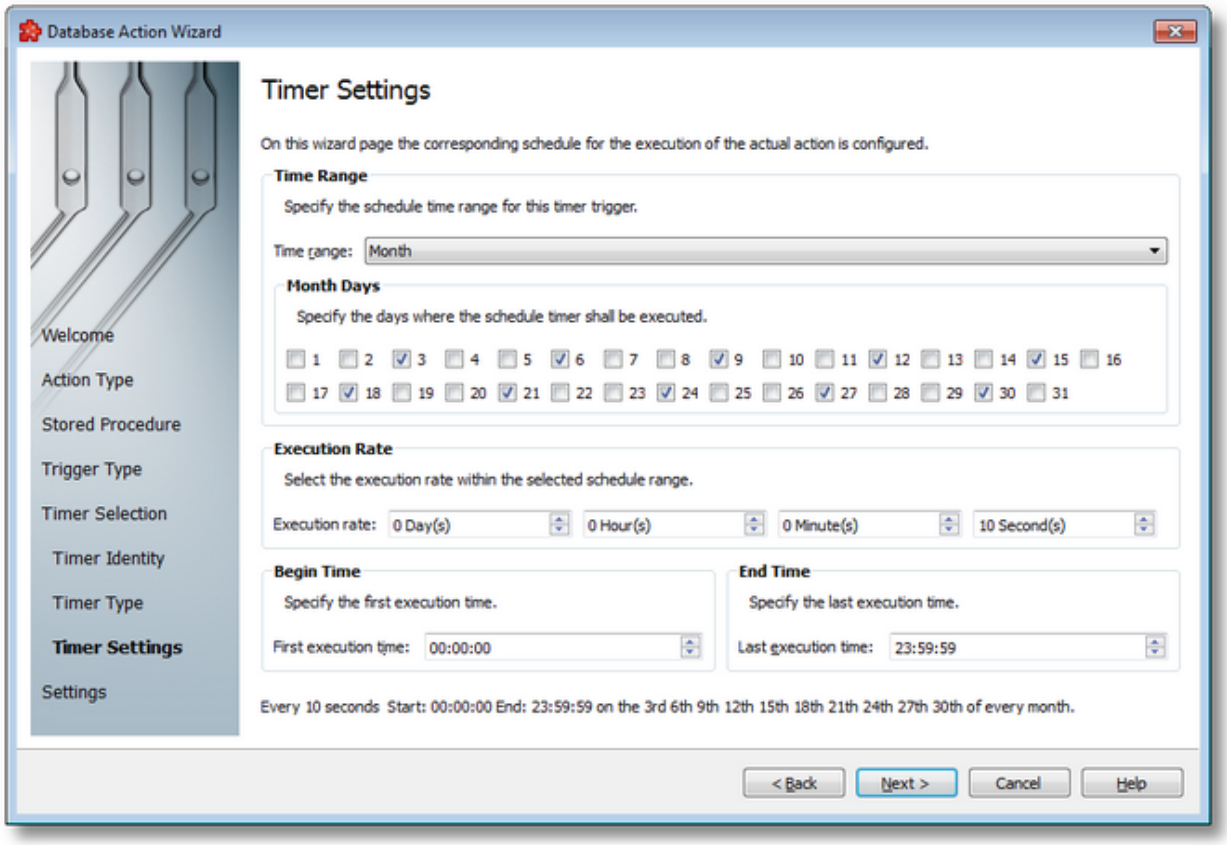


The time interval value is measured in milliseconds and ensures that the action will be triggered at the time interval defined in the spin-box.

Press the *Next* button to proceed to the *Timer Selection* wizard page, return to the previous one using the *Back* button or exit the wizard pressing the *Cancel* button.

<<< Database Action Wizard - Timer Type
Database Action Wizard - Timer Selection >>>

The *Schedule Timer Settings* wizard page is a more complex page compared to the *Interval Timer Settings* wizard page and allows defining the corresponding schedule for the execution of the trigger action:



The page has four sections which identify the schedule:

- *Time Range*: Provides the possibility to choose one of the following schedule ranges: *month*, *week*, *day*, *hour* or *minute*. When the *month* or the *week* ranges are selected the panel below the selection is populated with the month and week days, respectively.
- *Execution Rate*: Sets the execution rate of the current action in the previously-defined *Time Range* (e.g. every X day(s), every Y hour(s), every Z minute(s), every T second(s)).
- *Begin Time*: Sets the time the current action should begin execution.
- *End Time*: Sets the time the current action should stop execution.

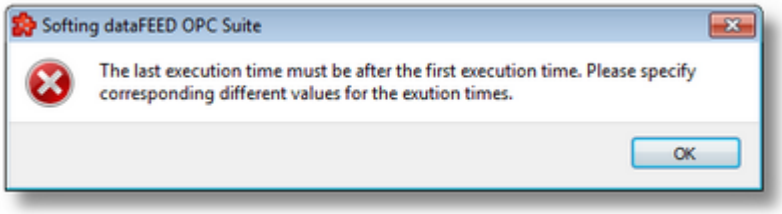
According to the *Time Range* the *Begin Time* and the *End Time* fields will allow typing:

- Seconds (SS) if the *Time range* is *Minute*.
- Minutes and seconds (MM:SS) if the *Time range* is *Hour*.
- Hours, minutes and seconds (HH:MM:SS) if the *Time range* is *Day* or greater (*Week*, *Month*).

Note: A short review of the configured timer schedule is shown at the bottom of this wizard page.

If the defined schedule is not valid, an error message dialog will appear naming the error that occurred.

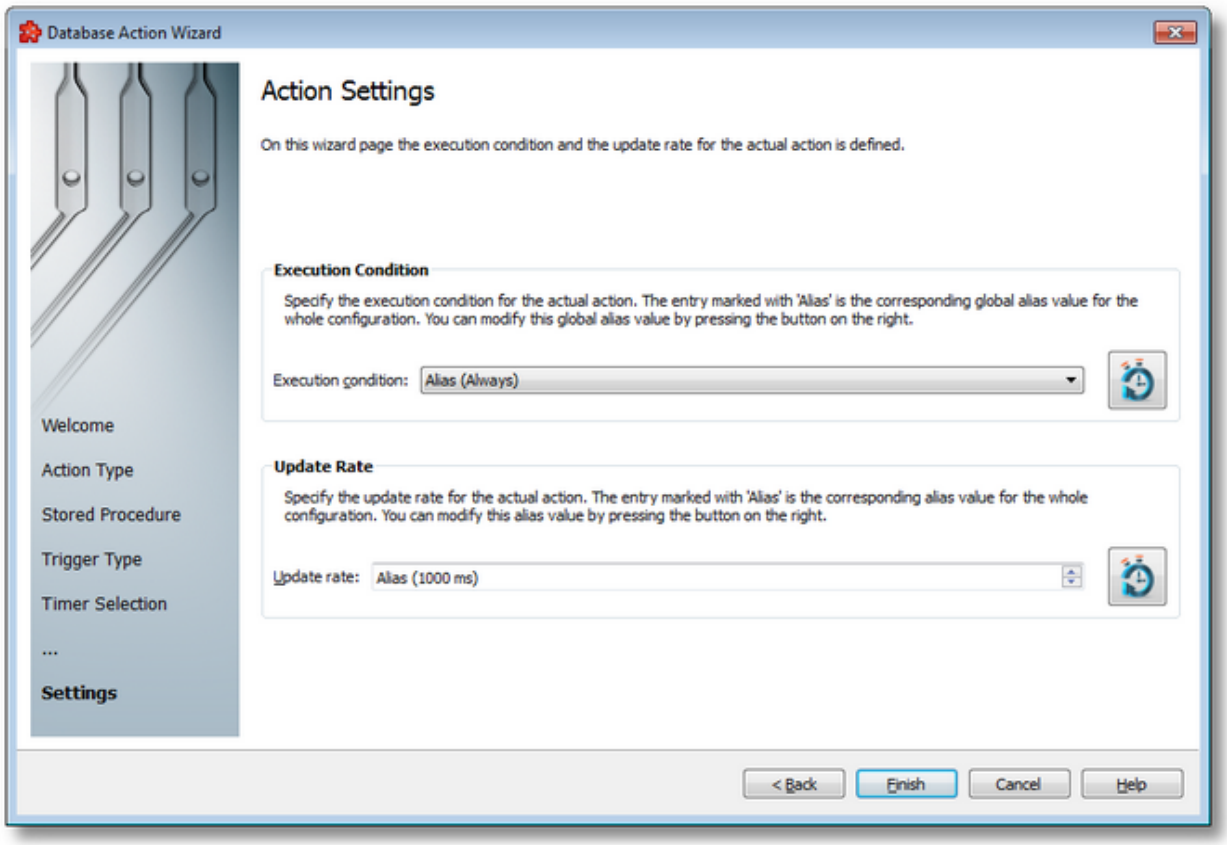
© Softing Industrial Automation GmbH



Continue to the *Timer Selection* wizard page using the *Next* button, return to the previous wizard page using the *Back* button or exit the wizard pressing the *Cancel* button.

<<< Database Action Wizard - Interval Timer Settings Database Action Wizard - Timer Selection >>>

The Action Settings wizard page allows defining the execution condition and the subscription update rate for the current database action.



The wizard page is divided into two sections: *Execution Condition* and *Update Rate*.

Execution Condition

Each statement has an execution condition which is evaluated by the local dataFEED OPC Suite application. The action is only performed when the evaluation is successful. If the evaluation fails then the statement is ignored.

The execution conditions allow reacting to server communication errors, executing special statements or preventing the execution of other statements.

The execution condition can be selected directly from the drop down combo box. The initial value is the global alias one. Changing the global default execution condition can be done using the Configure global alias and initial values button  or pressing the keyboard shortcut CTRL+D.

Update Rate

The *Update Rate* specifies the execution rate of the current database action.

The update rate can be typed or selected from the spin box. The initial value is the global alias one. Changing the global default update rate can be done using the Configure global alias and initial values button  or pressing the keyboard shortcut CTRL+D.

Complete the *Database Action Wizard* pressing the *Finish* button, review the past wizard pages using the *Back* button, or cancel the wizard thus discarding all the changes by pressing the *Cancel* button.

<<< Database Action Wizard -Trigger Type

7.3.3.5.5 Table Column Wizard

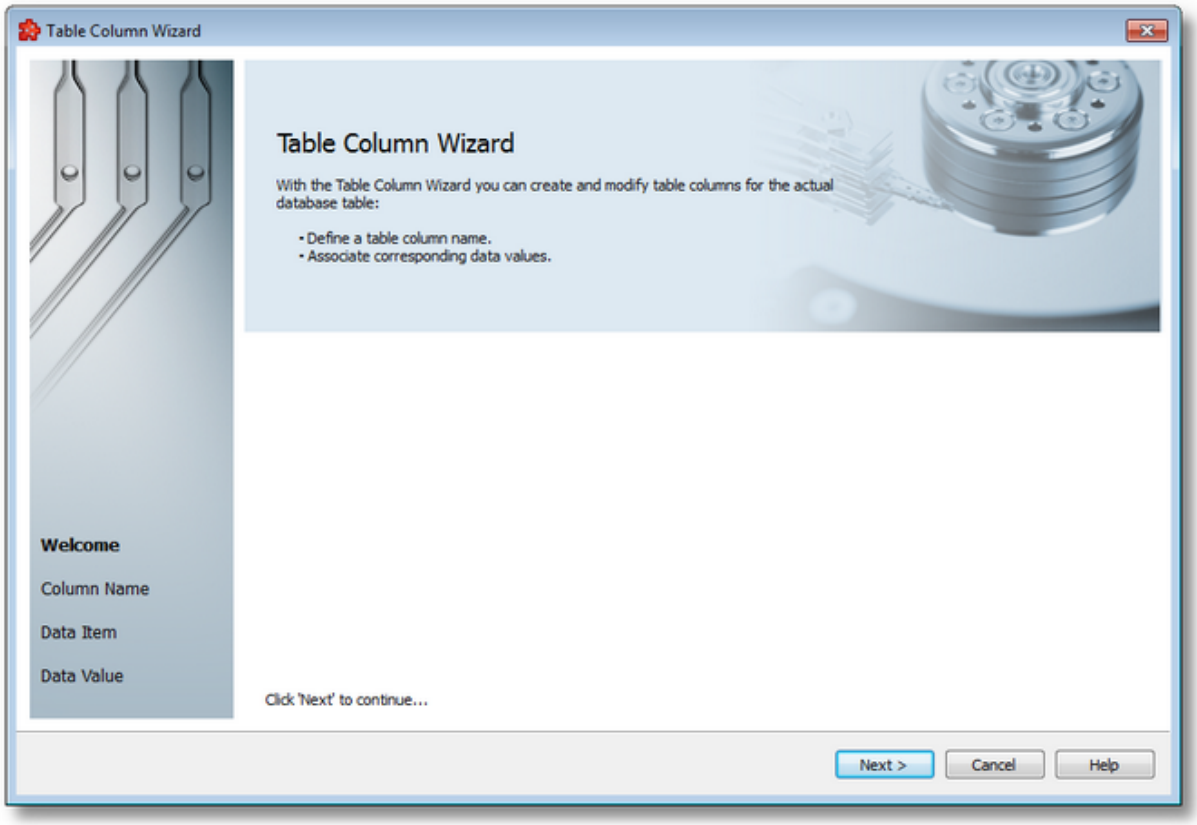
- An offline table column can be added through the Table Column Wizard:

  - Obtain general information about the Table Column Wizard @ Table Column Wizard - Welcome
  - Define a new table column @ Table Column Wizard - Column Name
  - Associate a server item with the newly created table column @ Table Column Wizard - Data Item
  - Select the data value and source @ Table Column Wizard - Data Value

© Softing Industrial Automation GmbH

The individual steps in this wizard are described in the Welcome page:

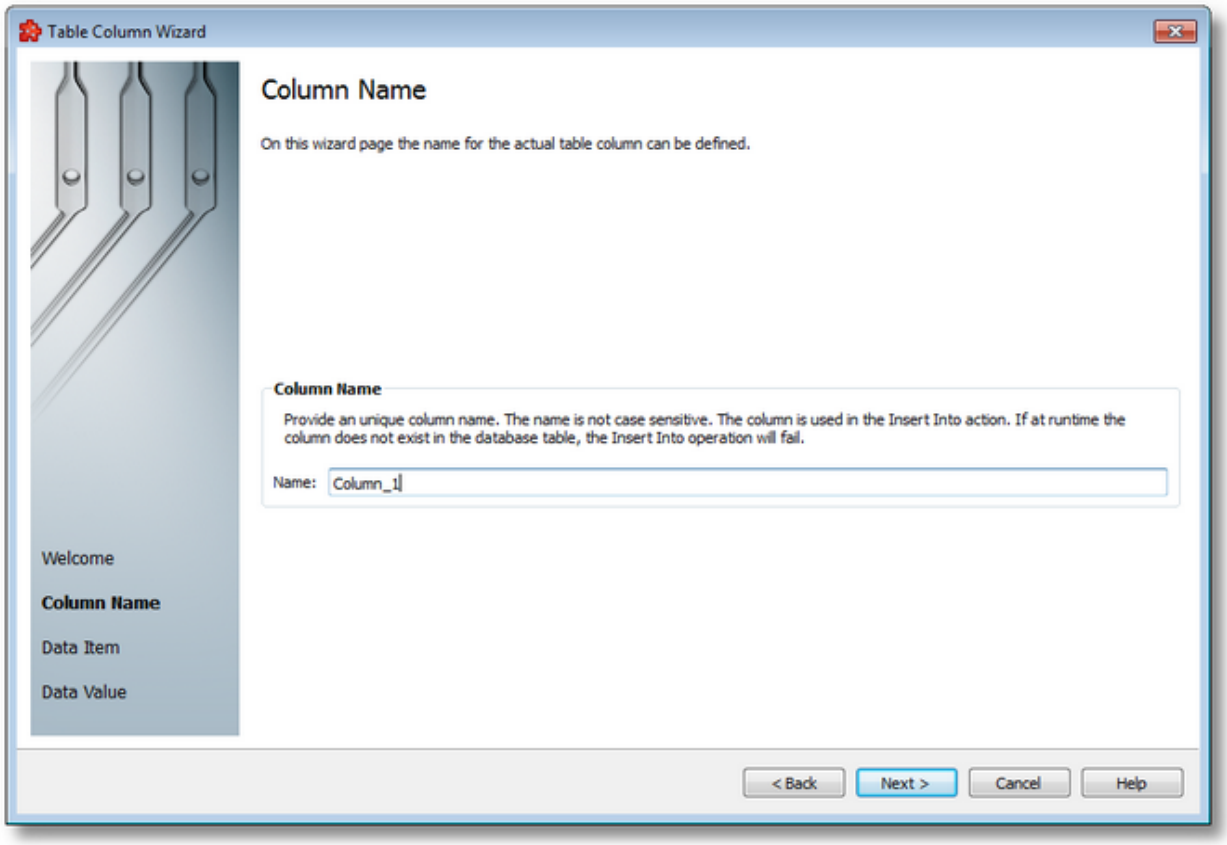
- Provide a name for the new table column in the Column Name page.
- Associate an item or a valid wildcard from the local dataFEED OPC Suite application namespace in the Data Item page.
- Select the data value and the data source for the selected item in the Data Value page.



Proceed to the next page by pressing the *Next* button or exit the wizard by pressing the *Cancel* button.

Table Column Wizard - Column Name >>>

The Column Name page allows selecting the name of the table column that will be added as part of the Insert Into action. No default value is available for the column name.



The column name must be manually typed. After adding this column, it will be displayed like an online column if it exists in the database table.

Note: dataFEED OPC Suite supports column names encoded in UTF-8, thus supporting any possible column name. However, this does not guarantee that the column name will also be supported by the ODBC driver or by the database itself. Additional changes might be necessary on the ODBC or database side to support unicode.

Continue to the next wizard page to associate a data placeholder using the *Next* button, go back to the previous wizard page using the *Back* button or quit the wizard by pressing the *Cancel* button.

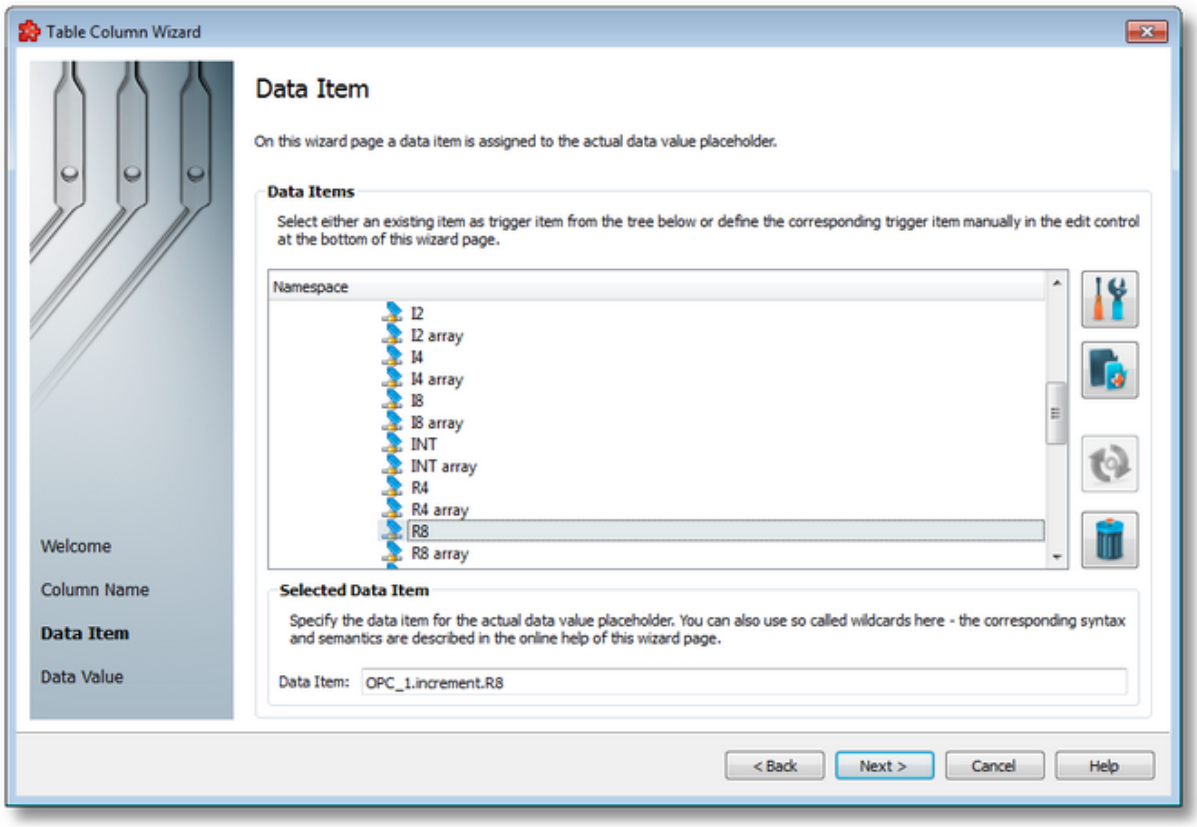
<<< Table Column Wizard - Welcome Table Column Wizard - Data Item >>>

The *Data Item* wizard page provides the possibility to select the placeholder to be associated with the new table column. The placeholder item can be:

- Any item originating from a *Server Connection*, from the *Local Items* or from the *System Items* can be selected, choosing a tag from the Local Namespace.

© Softing Industrial Automation GmbH

- A valid wildcard can be manually created by typing its regular expression in the *Data Item* edit box.



Local Namespace Tree Handling

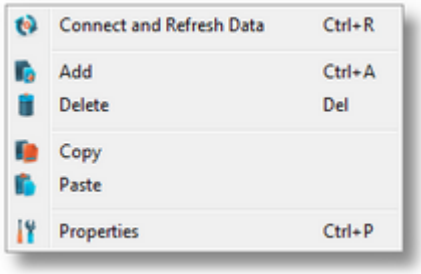
The *Local Namespace* tree offers the possibility to explore the entire namespace of the local dataFEED OPC Suite application in order to select a data item. After the selection is made, the *Data Item* edit box is populated with the selected item ID.

The entire namespace of the local dataFEED OPC Suite application can be managed through this wizard page by performing certain operations using the buttons on the right, the context sensitive menu or the keyboard shortcuts:

- Open the item properties dialog using the *Properties of the selected item* button  , pressing the keyboard shortcut CTRL+P or selecting the *Properties* action from the context menu. Depending on the type of the selected item, a special Item Properties dialog opens:
    - If the selected item is under Local Items the Modify Local Item dialog is opened.
    - If the selected item is under a dynamic server connection the Modify Dynamic Server Item dialog is opened.
    - If the selected item is under a static server connection the Modify Static Server Item dialog is opened.

- Add a new item using the *Add a new item*  button, pressing the keyboard shortcut CTRL+A or selecting the *Add* action from the context menu. Depending on the type of the selected node, a special Add Item dialog opens:
  - If the selected node is under Local Items the Add Local Item dialog is opened.
  - If the selected node is under a dynamic server connection the Add Dynamic Server Item dialog is opened.
  - If the selected node is under a static server connection the Add Static Server Item dialog is opened.
- Copy the selected item to clipboard by selecting the *Copy* action from the context menu.
- Paste the previously copied item from clipboard by selecting the *Paste* action from the context menu. Items originating from the Local Items cannot be pasted under server connections, and vice-versa; an error message will be displayed in both cases. If an item is pasted in a static (filtered) server connection and the item does not exist on the server an error message is displayed.
- Connect to the server and refresh the data items by using the *Refresh data items*  button, pressing the keyboard shortcut CTRL+R or selecting the *Connect and Refresh Data* action from the context menu. The namespace of the selected server connection is recreated; if manually added items not referenced in the actual configuration exist, those items are deleted from the namespace.
- Delete the selected item and all of its child items by using the Delete data items button  , pressing the DELETE / DEL key or selecting the Delete action from the context menu. Items which are included in any Data Processing / Data Destination add-ons will not be deleted but an error message will be displayed instead.

An example of the context menu can be seen below:



Attention: The *Local Namespace* root node, the *Local Items* root node and the server connection root nodes cannot be deleted. An item that is used in consumers or a system item cannot be deleted either.

Note: The copy/paste actions work on the selected tag/property or on the whole selected branch. The copy action is not available for the Local Namespace root or for the server connection root nodes; the paste action is not available for items of type property (as they cannot have any children).

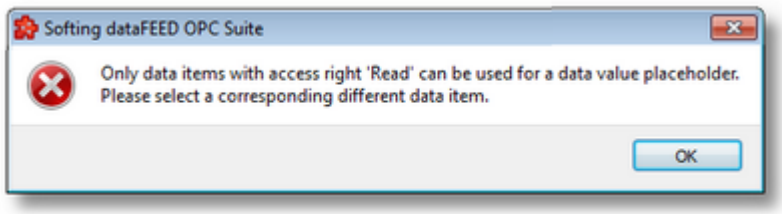
© Softing Industrial Automation GmbH

The entire namespace of the local Softing dataFEED OPC Suite application is exposed in this wizard page. The tree root node name is composed of the *Local Namespace* keyword followed by the instance name.

Selected Data Item Handling

When an item from the Local Namespace is selected, the *Data Item* edit box is populated with the item ID.

Note: Only tag and property items are allowed to be used in placeholder. If a node item is selected, an error message dialog will appear when the *Next* button is pressed.



Note: The *Data Item* edit box in the *Selected Data Item* section of this wizard page is editable. The text typed in this text area is considered a wildcard.

Regular expressions with special characters (asterisk, question mark, square brackets) can be used in this field.

The asterisk wildcard (*) substitutes any sequence of characters of any length:

- SID* - replaces any string of at least 3 characters that starts with SID
- S*D – replaces a string of at least 2 characters that starts with S and ends with D

The question mark wildcard (?) substitutes a single character:

- SID? – replaces any string of 4 characters that starts with SID.
- S?D – replaces any string of 3 characters that starts with S and ends with D.

The range wildcard ([ ]) substitutes any single character in the range specified between the square brackets:

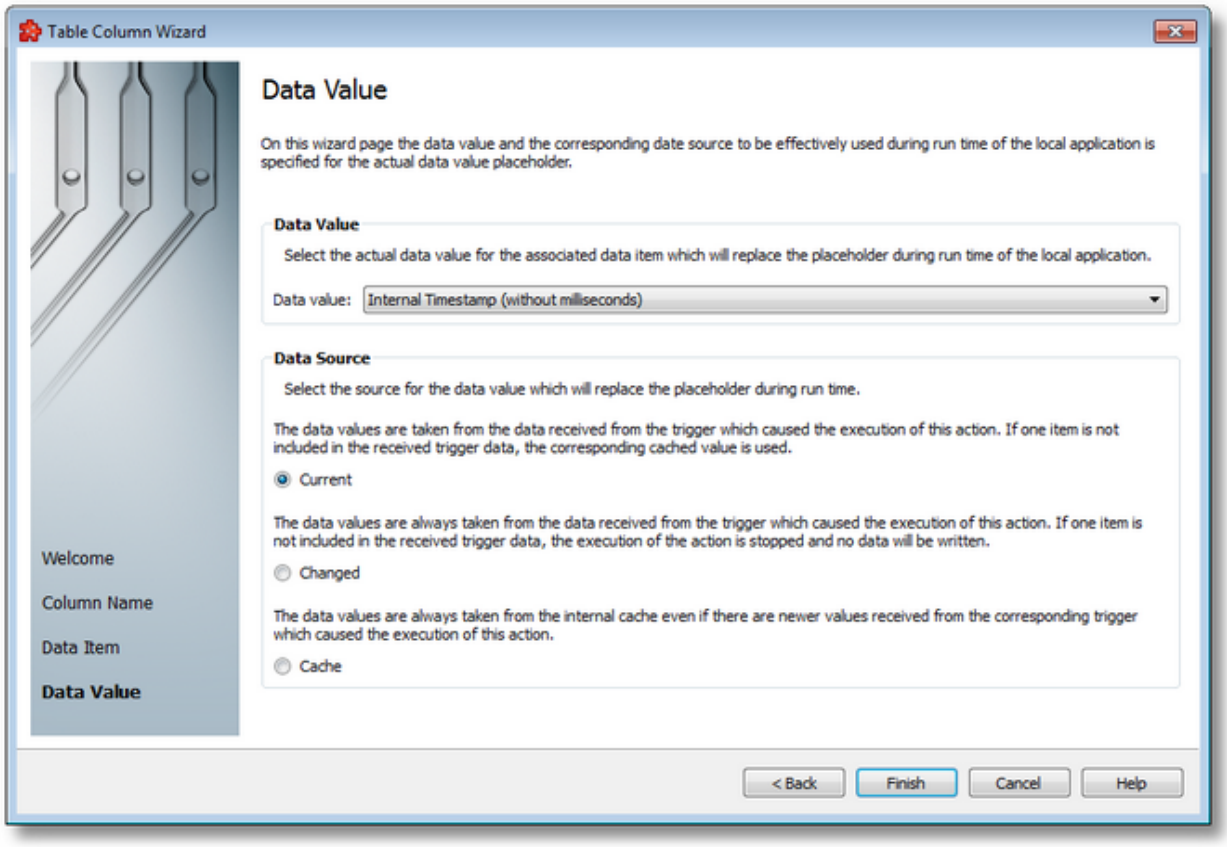
- S[A-Z]D – replaces any string of 3 characters that starts with S, ends with D and contains any character in the range A to Z.
- SID[0-9] – replaces any string of 4 characters that starts with SID and ends with any number in the range 0 to 9.

Proceed to the next page to set the item data value and source by using the *Next* button, return to the previous wizard page by using *Back* button, or quit the wizard by pressing the *Cancel* button.

<<< Table Column Wizard - Column Name Table Column Wizard - Data Value >>>

The Data Value wizard page provides the possibility to set the properties of the placeholder:

- Data Value - select the actual data value for the associated data item which will be replaced during the run time of the local dataFEED OPC Suite application.
- Data Source - select the source for data value which will be replaced during the run time of the local dataFEED OPC Suite application.



Data Value

The following data value attributes may be used:

Attribute	Description
Data Value	The value of the data source server item.
Item Identified	The ID of the data source server item.
Quality as Integer	The data source server item quality as integer.

© Softing Industrial Automation GmbH

Quality as String	The data source server item quality as string.
Timestamp (without milliseconds)	The data source server item timestamp as date time without milliseconds.
Timestamp (with milliseconds)	The data source server item timestamp as date time including milliseconds.
Timestamp (milliseconds part only)	The milliseconds part of the data source server item timestamp.
Internal Timestamp (without milliseconds)	The timestamp of data change in the local dataFEED OPC Suite application as date time without milliseconds.
Internal Timestamp (with milliseconds)	The timestamp of data change in the local dataFEED OPC Suite application as date time including milliseconds.
Internal Timestamp (milliseconds part only)	The milliseconds part of timestamp of data change in the local dataFEED OPC Suite application.

Data Source

The following data sources may be selected:

Data Source	Description
Current	The value used is the current one. This means that if the item is included in the transaction, the received value is used. If the item is not included, the cached value is used.
Cache	The cached value is always used, even if the item is present in the incoming transaction.
Changed	The item must be included in the incoming transaction as the current value will be used.

Attention: If the action trigger is set to *Data Change* and the corresponding action contains at least one placeholder having *Changed* as the *Data Source*, that specific item must exist in the incoming transaction in order for the action to be executed. When the item is not present in the transaction, the action is invalidated.

Complete the wizard using the *Finish* button, go back to the previous wizard page using the *Back* button, or discard any changes made in this wizard by pressing the *Cancel* button.

<<< Table Column Wizard - Data Item

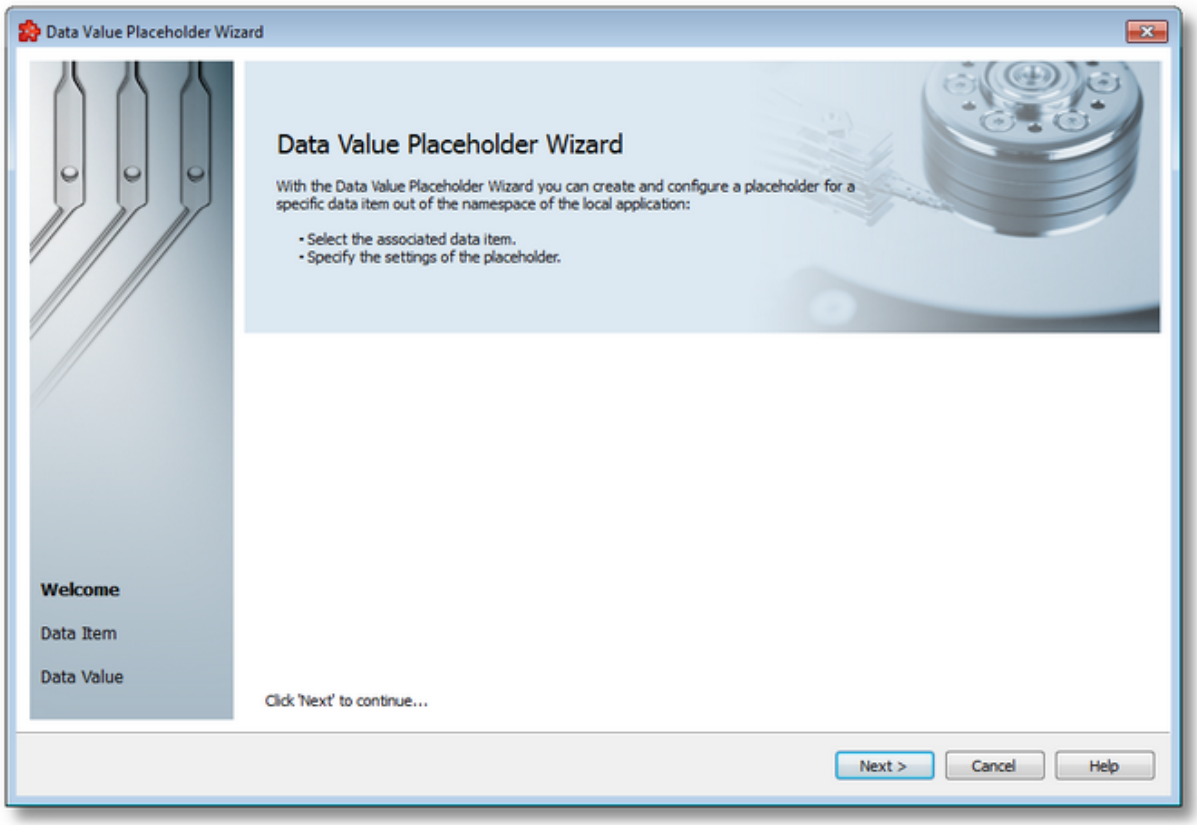
7.3.3.5.6 Data Value Placeholder Wizard

With the Data Value Placeholder Wizard you can create a placeholder from a configured data source or from a valid wildcard:

- Obtain a general description of the wizard @ Data Value Placeholder Wizard - Welcome
- Select a placeholder item @ Data Value Placeholder Wizard - Data Item
- Select the data value and source @ Data Value Placeholder Wizard - Data Value

The Welcome wizard page describes the individual steps for defining the placeholder:

- Select an item from the local dataFEED OPC Suite application namespace or provide a valid wildcard.
- Choose the data value and the data source for the selected item.



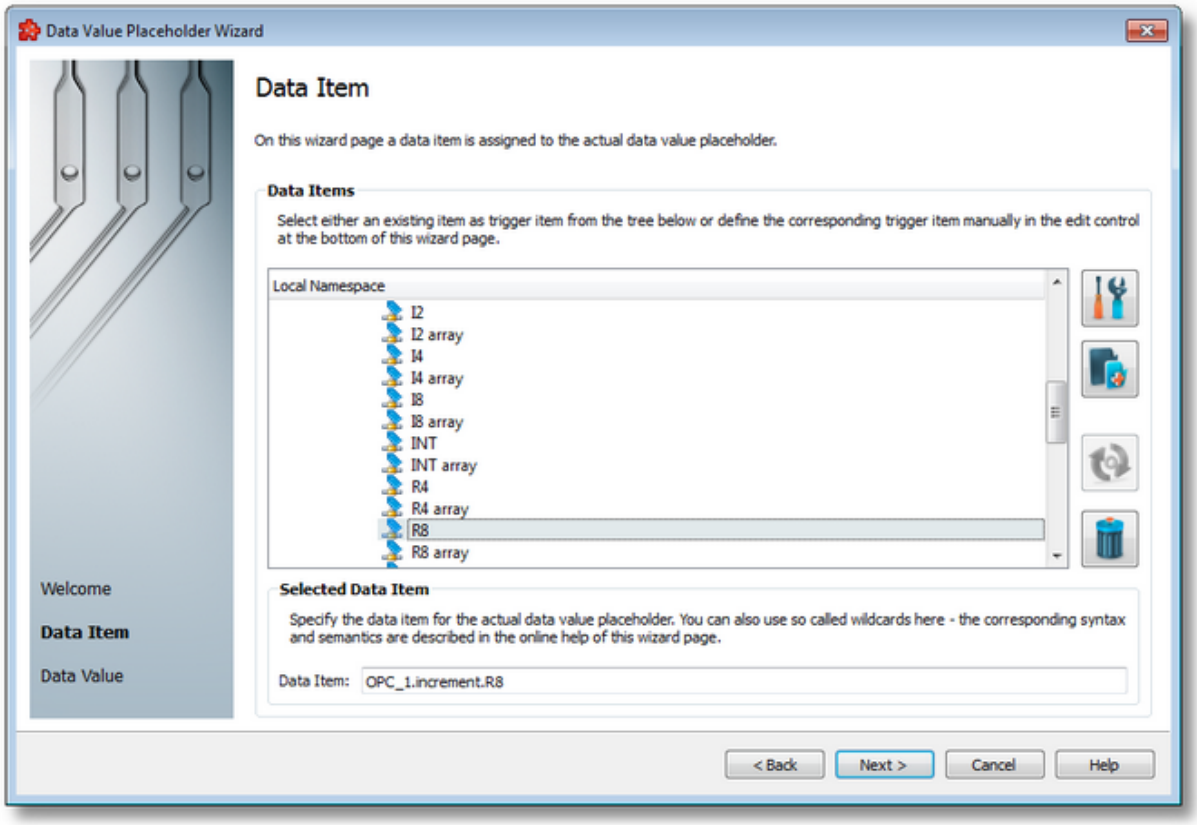
Continue to the next wizard page using the *Next* button or quit the wizard pressing the *Cancel* button.

© Softing Industrial Automation GmbH

Data Value Placeholder Wizard - Data Item >>>

The Data Item wizard page allows selecting the placeholder item. The placeholder item can be:

- Any item originating from a *Server Connection*, from the *Local Items* or from the *System Items* can be selected, choosing a tag from the *Local Namespace*.
- A valid wildcard can be manually created by typing its regular expression in the *Data Item* edit box.



Local Namespace Tree Handling

The *Local Namespace* tree offers the possibility to explore the entire namespace of the local dataFEED OPC Suite application in order to select a data item. After the selection is made, the *Data Item* edit box is populated with the selected item ID.

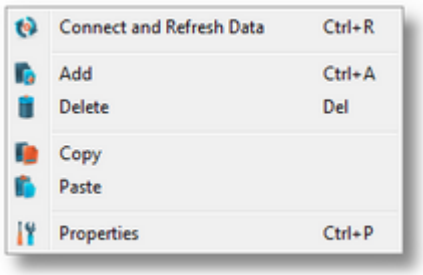
The entire namespace of the local dataFEED OPC Suite application can be managed through this wizard page by performing certain operations using the buttons on the right, the context sensitive menu or the keyboard shortcuts:

- Open the item properties dialog using the *Properties of the selected item* button  , pressing the keyboard shortcut CTRL+P or selecting the *Properties* action from the context menu. Depending on the type of the selected item, a special Item Properties dialog opens:

- If the selected item is under Local Items the Modify Local Item dialog is opened.
- If the selected item is under a dynamic server connection the Modify Dynamic Server Item dialog is opened.
- If the selected item is under a static server connection the Modify Static Server Item dialog is opened.

- Add a new item using the *Add a new item*  button, pressing the keyboard shortcut CTRL+A or selecting the *Add* action from the context menu. Depending on the type of the selected node, a special Add Item dialog opens:
  - If the selected node is under Local Items the Add Local Item dialog is opened.
  - If the selected node is under a dynamic server connection the Add Dynamic Server Item dialog is opened.
  - If the selected node is under a static server connection the Add Static Server Item dialog is opened.
- Copy the selected item to clipboard by selecting the *Copy* action from the context menu.
- Paste the previously copied item from clipboard by selecting the *Paste* action from the context menu. Items originating from the Local Items cannot be pasted under server connections, and vice-versa; an error message will be displayed in both cases. If an item is pasted in a static (filtered) server connection and the item does not exist on the server an error message is displayed.
- Connect to the server and refresh the data items by using the *Refresh data items*  button, pressing the keyboard shortcut CTRL+R or selecting the *Connect and Refresh Data* action from the context menu. The namespace of the selected server connection is recreated; if manually added items not referenced in the actual configuration exist, those items are deleted from the namespace.
- Delete the selected item and all of its child items by using the Delete data items button  , pressing the DELETE / DEL key or selecting the Delete action from the context menu. Items which are included in any Data Processing / Data Destination add-ons will not be deleted but an error message will be displayed instead.

An example of the context menu can be seen below:



© Softing Industrial Automation GmbH

Attention: The *Local Namespace* root node, the *Local Items* root node and the server connection root nodes cannot be deleted. An item that is used in consumers or a system item cannot be deleted either.

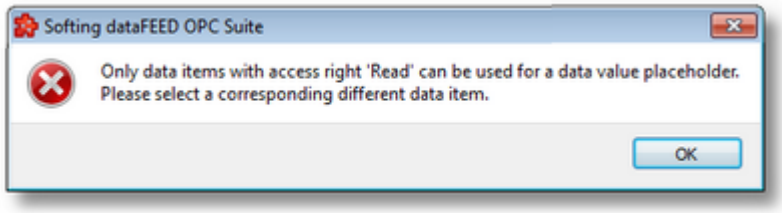
Note: The copy/paste actions work on the selected tag/property or on the whole selected branch. The copy action is not available for the Local Namespace root or for the server connection root nodes; the paste action is not available for items of type property (as they cannot have any children).

The entire namespace of the local Softing dataFEED OPC Suite application is exposed in this wizard page. The tree root node name is composed of the *Local Namespace* keyword followed by the instance name.

Selected Data Item Handling

When an item from the Local Namespace is selected, the *Data Item* edit box is populated with the item ID.

Note: Only tag and property items are allowed to be used in placeholder. If a node item is selected, an error message dialog will appear when the *Next* button is pressed.



Note: The *Data Item* edit box in the *Selected Data Item* section of this wizard page is editable. The text typed in this text area is considered as wildcard.

Regular expressions with special characters (asterisk, question mark, square brackets) can be used in this field.

The asterisk wildcard (*) substitutes any sequence of characters of any length:

- SID* - replaces any string of at least 3 characters that starts with SID
- S*D – replaces a string of at least 2 characters that starts with S and ends with D

The question mark wildcard (?) substitutes a single character:

- SID? – replaces any string of 4 characters that starting with SID.
- S?D – replaces any string of 3 characters that starts with S and ends with D.

The range wildcard ([ ]) substitutes any single character in the range specified between the square brackets:

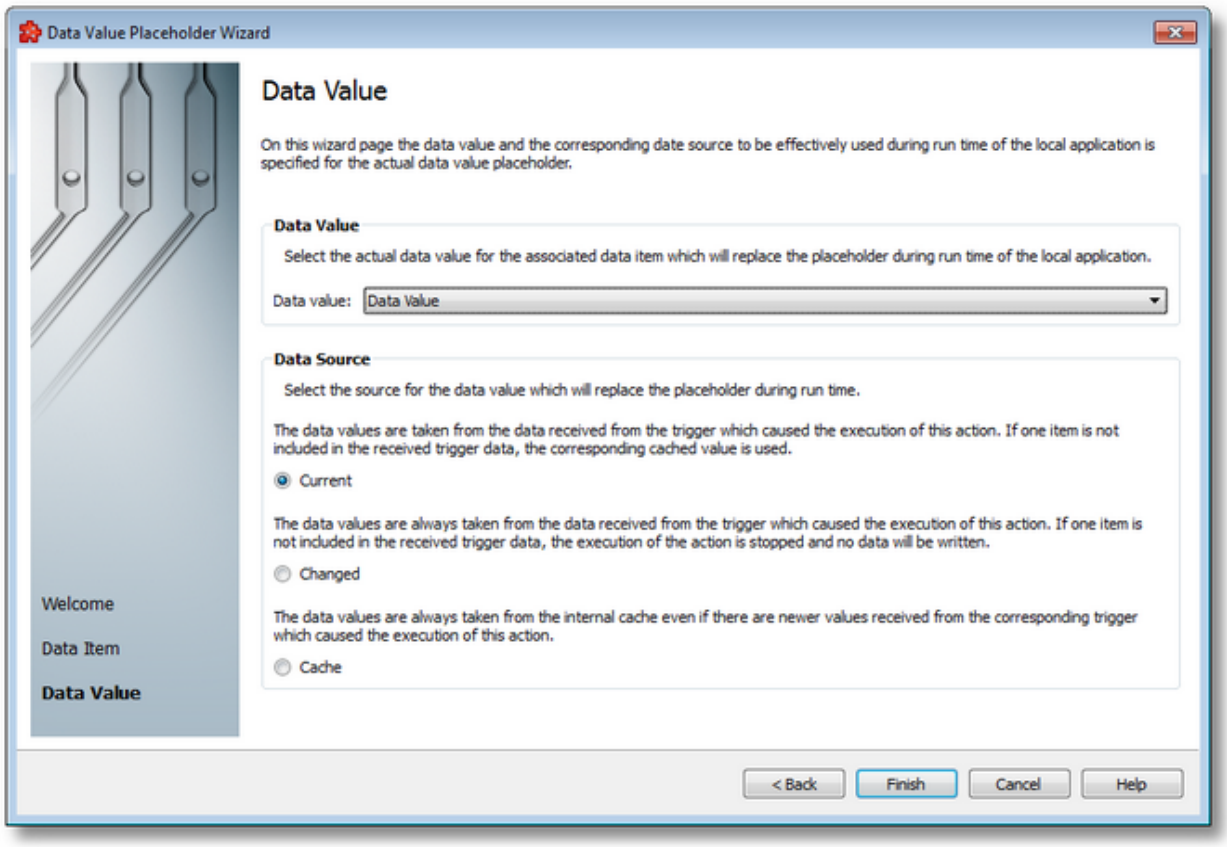
- S[A-Z]D – replaces any string of 3 characters that starts with S, ends with D and contains any character in the range A to Z.
- SID[0-9] – replaces any string of 4 characters that starts with SID and ends with any number in the range 0 to 9.

Proceed to the next page to set the item data value and source by using the *Next* button, return to the previous wizard page by using *Back* button or quit the wizard by pressing the *Cancel* button.

<<< Data Value Placeholder Wizard - Welcome Data Value Placeholder Wizard - Data Value >>>

The Data Value wizard page provides the possibility to set the properties of the placeholder:

- Data Value - select the actual data value for the associated data item which will be replaced during the run time of the local dataFEED OPC Suite application.
- Data Source - select the source for data value which will be replaced during the run time of the local dataFEED OPC Suite application.



Data Value

© Softing Industrial Automation GmbH

The following data value attributes may be used:

Attribute	Description
Data Value	The value of the data source server item.
Item Identified	The ID of the data source server item.
Quality as Integer	The data source server item quality as integer.
Quality as String	The data source server item quality as string.
Timestamp (without milliseconds)	The data source server item timestamp as date time without milliseconds.
Timestamp (with milliseconds)	The data source server item timestamp as date time including milliseconds.
Timestamp (milliseconds part only)	The milliseconds part of the data source server item timestamp.
Internal Timestamp (without milliseconds)	The timestamp of data change in the local dataFEED OPC Suite application as date time without milliseconds.
Internal Timestamp (with milliseconds)	The timestamp of data change in the local dataFEED OPC Suite application as date time including milliseconds.
Internal Timestamp (milliseconds part only)	The milliseconds part of timestamp of data change in the local dataFEED OPC Suite application.

Data Source

The following data sources may be selected:

Data Source	Description
Current	The value used is the current one. This means that if the item is included in the transaction, the received value is used. If the item is not included, the cached value is used.
Cache	The cached value is always used, even if the item is present in the incoming transaction.
Changed	The item must be included in the incoming transaction as the current value will be used.

Attention: If the action trigger is set to *Data Change* and the corresponding action contains at least one placeholder having *Changed* as the *Data Source*, that specific item must exist in the incoming transaction in order for the action to be executed. When the item is not present in the transaction, the action is invalidated.

Complete the wizard using the *Finish* button, go back to the previous wizard page using the *Back* button, or discard any changes made in this wizard by pressing the *Cancel* button.

<<< Data Value Placeholder Wizard - Data Item

7.3.3.5.7 Data Values Placeholder

Placeholders can be used when configuring various actions (SQL, TEXT and other statements). These placeholders can be defined in the *Database Action Wizard*, *File Action Wizard*, *Database* or *File* main user interface and are saved in the Subscription file. A placeholder is determined by the following attributes:

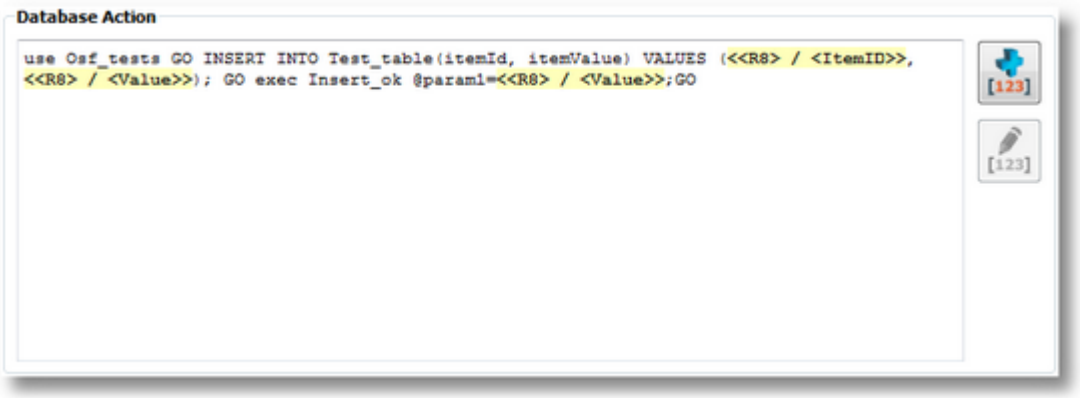
- Data Item - an item (tag, property or node with value) originating from a *Server Connection*, from the *Local Items or* from the *System Items*, or a valid wildcard.
- Data Value - the actual data value for the associated data item which will be replaced during the run time of the local dataFEED OPC Suite application.
- Data Source - the source for the data value which will be replaced during the run time of the local dataFEED OPC Suite application.

Placeholder in the user interface

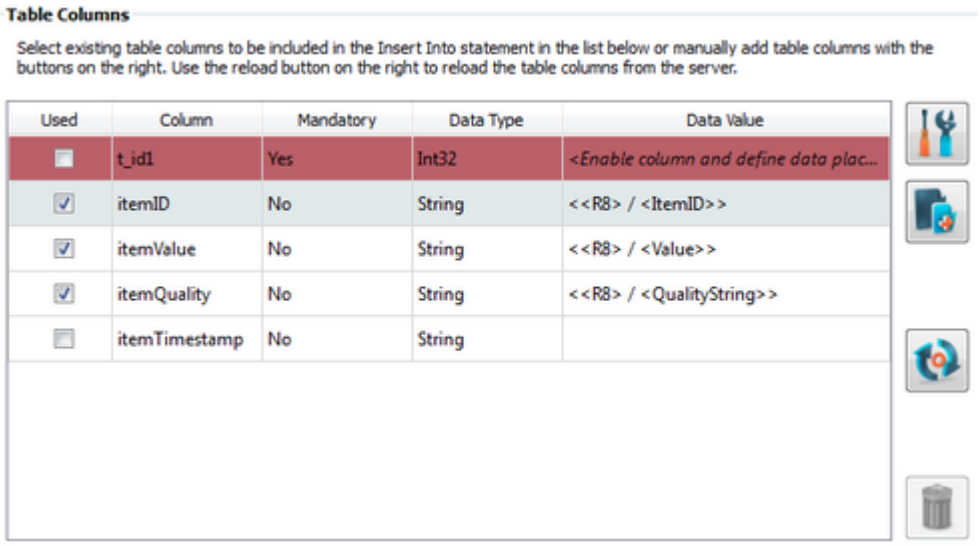
Two types of placeholder editors are implemented in the Softing Configurator. One is the *Text Action Editor* found in the Stored Procedure and SQL Statement wizard pages and also in the Configuration: Data Destination - Database main user interface page of the Database add-on (when either a *Stored Procedure* or a *SQL Statement* action is selected); the corresponding placeholder form is: <<Short Data Value> / <Short Data Item>>:

```
<<R8> / <ItemID>>
<<R8> / <Value>>
<<R8> / <QualityString>>
<<R8> / <TimestampMS>>
```


© Softing Industrial Automation GmbH



The other placeholder editor is the one implemented in the Table Columns wizard page and also in the Configuration: Data Destination - Database main user interface page of the Database add-on (when an *Insert Into Statement* action is selected):



The placeholders are saved in the subscription files between a pair of simple quote marks to fix possible SQL issues:

```
'#%-ItemTimestampOCT%OPC_1.increment.R8_Subscription_222%Current-%#'
```

Placeholder in the subscription file

In the subscription file a placeholder has the following syntax:

```
#%-<Data Item>%<ItemID_SubscriptionID>%<DataSource>-%#
```

where:

Attribute	Description

#%- -%#	Represents the beginning and the end of a placeholder definition
Data Item	Represents the item data value.
%	Separates the Data Value, Data Item and Data Source fields in the placeholder.
ItemID_SubscriptionID	Represents the subscription item identifier.
Data source	Represents the item data source.

Example:

```
#%-ItemValue%OPC_1.increment.R8_Subscription_222%Current-%#
```

Data value:

The following data value attributes can be used (they are supported by both dataFEED OPC Suite Configurator and local dataFEED OPC Suite application):

Attribute	**Subscription file form**	**Description**
Data Value	ItemValue	The value of the data source server item.
Item Identified	ItemItemID	The ID of the data source server item.
Quality as Integer	ItemQuality	The data source server item quality as integer.
Quality as String	ItemQualityString	The data source server item quality as string.
Timestamp (without milliseconds)	ItemTimestamp	The data source server item timestamp as date time without milliseconds.
Timestamp (with milliseconds)	ItemTimestampMS	The data source server item timestamp as date time including milliseconds.
Timestamp (milliseconds part only)	ItemTimestampMS Part	The milliseconds part of the data source server item timestamp.
Internal Timestamp (without milliseconds)	ItemTimestampOC T	The timestamp of data change in the local dataFEED OPC Suite application as date time without milliseconds.
Internal Timestamp (with milliseconds)	ItemTimestampOC TMS	The timestamp of data change in the local dataFEED OPC Suite application as date time including milliseconds.

© Softing Industrial Automation GmbH

Internal Timestamp (milliseconds part only)	ItemTimestampOCTMSPart	The milliseconds part of the data change timestamp in the local dataFEED OPC Suite application.

The following data value attributes can be used (only supported by the local dataFEED OPC Suite application):

Attribute	**Subscription file form**	**Description**
Item Subscription ID	ItemSubscriptionID	The subscription ID of the item
Data Value	DataValue	The ID of the data source server item.
Data Quality as Integer	DataQuality	The data source server item quality as integer.
Data Quality as String	DataQualityString	The data source server item quality as string.
Timestamp (without milliseconds)	DataTimestamp	The data source server item timestamp as date time without milliseconds.
Timestamp (with milliseconds)	DataTimestampMS	The data source server item timestamp as date time including milliseconds.
Internal Timestamp (without milliseconds)	DataTimestampOCT	The timestamp of data change in the local dataFEED OPC Suite application as date time without milliseconds.
Internal Timestamp (with milliseconds)	DataTimestampOCTMS	The timestamp of data change in the local dataFEED OPC Suite application as date time including milliseconds.
Data Item ID	DataItemID	The value of the data source server item.

Attention: There is a significant difference between the Data Values in the tables above. The values in the second table are taken directly from the dataFEED OPC Suite internal server while the values for the Data Values from the first table are taken directly from the server that holds the real Item. dataFEED OPC Suite holds only a copy of that Item.

Data Source or Item type:

One of the following Data Sources can be selected:

Item type	**Description**
Current	The item must be in the subscription item list. The value used is the current one, which means that if the value for the item is included in the current transaction, this value will be used. If the

	value is not included in the incoming transaction, the cached value will be used.
Cache	The item must be in the subscription item list. The value used is the cached one, even if that item is included in the incoming transaction.
Changed	The item must be in the subscription item list. The value used is the changed one, which means that the entire action will be invalidated if the item is not present in the incoming transaction.

Examples:

```
'#%-ItemTimestampOCTMSPart%OPC_1.increment.R8_Subscription_222%Current-%#'
'#%-ItemTimestampOCTMS%OPC_1.increment.R8_Subscription_222%Changed-%#'
'#%-ItemTimestampOCT%OPC_1.increment.R8_Subscription_222%Cache-%#'
```

Attention: If the *Changed* Data Source is used in an action and the corresponding action trigger is a *Data Change* trigger, make sure that a transaction which contains data changes for all the items used in the current action will be generated by the server; otherwise, the entire action will be invalidated if an item has no associated data change in the incoming transaction.

Wildcards

A placeholder can also contain a wildcard at the position of ItemID_SubscriptionID, which is matched by the local dataFEED OPC Suite application against the items defined in the subscription file:

```
#%-ItemValue%Wildcard_02%Current-%#
#%-ItemQualityString%Wildcard_02%Cache-%#
#%-ItemTimestamp%Wildcard_02%Changed-%#
#%-ItemItemID%Wildcard_02%Current-%#
```

where Wildcard_02 is the wildcard ID of a wildcard previously defined in the subscription file.

Below you can find examples of possible placeholder definition forms. Please note that not all of them are supported by the dataFEED OPC Suite Configurator:

Placeholder Form	**Description**
#%-ItemValue%<Subscription Item ID>%item type-%#	The item value
#%-ItemQuality%<Subscription Item ID>%item type-%#	The item quality as integer
#%-ItemTimestamp%<Subscription Item ID>% item type-%#	The item timestamp as date time

© Softing Industrial Automation GmbH

#%-ItemTimestampMS%<Subscription Item ID>%item type-%#	The milliseconds part of the item timestamp
#%-ItemTimestampOCTMS%<Subscription Item ID>%item type-%#	The timestamp of data change in the local dataFEED OPC Suite application (milliseconds part)
#%-DataValue%<OCT Item ID>%item type%% data type-%#	The item value
#%-DataQuality%<OCT Item ID>%item type-%#	The item quality as integer
#%-DataQualityString%<OCT Item ID>%item type-%#	The item quality as readable string
#%-DataTimestamp%<OCT Item ID>%item type-%#	The item timestamp as date time
#%-DataTimestampOCT%<OCT Item ID>% item type-%#	The timestamp of data change in the local dataFEED OPC Suite application as date time
#%-TAB-%#	The tabulator character
#%-CRLF-%#	Windows-specific line separator (Carriage return + Line feed)
#%-LF-%#	Linux-specific line separator (Line feed)
#%-CR-%#	Macintosh-specific line separator (Carriage return)

Examples of wildcards used in Database Actions can be found below:

```
CONDITION<TAB><TAB>Always
CONDITION<TAB>AllGoodQuality<TAB>AllGoodQuality
#
SUBSCRIPTION<TAB><TAB>UPDATERATE<TAB>1000<TAB>ACTIVE<TAB>TRUE
SUBSCRIPTION<TAB>Subscription_222<TAB>UPDATERATE<TAB>222<TAB>ACTIVE<TAB>TRUE
#
WILDCARD<TAB>Wildcard_02<TAB>Current<TAB>OPC_1.increment.*8
#
ITEM<TAB>OPC_1.increment.R8_Subscription_222<TAB>OPC_1.increment.R8<TAB>Subscription_
#
ITEM<TAB>OPC_3.increment.R8_<TAB>OPC_3.increment.R8<TAB>
#
ITEM<TAB>OPC_3.increment.R8_Subscription_222<TAB>OPC_3.increment.R8<TAB>Subscription_
SQL<TAB>AllGoodQuality<TAB>MyDatabase_0<TAB>INSERT INTO Test_Table1 (itemId,itemQuali
('#%-ItemItemID%OPC_3.increment.R8_Subscription_222%Current-%#','#%-ItemValue%Wildcar
#
DATACHANGE
SQL<TAB><TAB>MyDatabase_0<TAB>use Osf_tests GO #%-CRLF-%# INSERT INTO Test_Table (ite
(#%-ItemItemID%OPC_3.increment.R8_%Current-%#, #%-ItemValue%OPC_3.increment.R8_%Curre
```

```
#
#!TIMER<TAB>Timer_01<TAB>INTERVAL<TAB>1000
#
TIMER<TAB>Timer_02<TAB>SCHEDULE<TAB>MONTH<TAB>1s<TAB>1h3m3s<TAB>23h59m59s<TAB>3d6d9d1
SQL<TAB><TAB>MyDatabase_0<TAB>exec Procedure_update_on
@param1=#%-ItemValue%OPC_3.increment.R8_%Current-%#, @param2=NULL, @param3=#%-ItemTim
```

7.3.3.5.8 Configuration Files

All the OPC2Database configuration settings are stored in two separated files:

- Tasks file - contains general descriptions of the Database definitions.
- Subscriptions file - which contains all the descriptions relating to the stored data:
    - what data will be stored.
    - what is the frequency of storing the data.
    - where and when will the data be stored.

Note: The path and file name of these configuration files are fixed and not configurable.

The task and subscription files are stored in the *Tasks* folder under the instance path. For task configuration a text file called *Tasks.txt* and for subscription configuration a text file called *Subscriptions.txt* will be created in the *Tasks* configuration folder. Although these files can be manually edited, using the dataFEED OPC Suite Configurator is advisable.

Note: The recommendation is to use only the dataFEED OPC Suite Configurator to edit these files. This ensures that all the definitions will be valid.

If the configuration is manually edited try loading the configuration in the dataFEED OPC Suite Configurator as all the configuration files are automatically validated at load-time and, if there are any mistakes, an error message will be displayed. The dialog will contain an appropriate description and will point to the first invalid line of the local items configuration file.

Both configurations files are plain text files encoded in UTF-8 (thus providing full UNICODE compatibility) and they have line-oriented structure. The elements of a line are tab-separated. The format of one line is the following:

```
Line type<TAB>Line data
```

Task Configuration File

The following line types are defined in the Tasks file:

Line Type	Description

© Softing Industrial Automation GmbH

SQL_COMMANDSTORE	The maximum number of stored SQL statements is saved under this type of line. It holds the value of the *Max. number of stored commands* field from the General Database Properties window in the dataFEED OPC Suite Configurator. This line is optional and can be omitted.
SQL_TIMESTAMPFORMAT	Sets the time format that is to be used when the timestamp placeholders are expanded and their values written to the database. It holds the value of the *Time format* and *Date format* fields from the General Database Properties window in the dataFEED OPC Suite Configurator. This line is optional and can be omitted.
DB	Defines the DSN parameters used in the ODBC connection: database server address, ODBC connection name and authentication credentials.
INIT	Used to specify an initialization SQL command which is executed only once after the local dataFEED OPC Suite application starts up (database selection etc).
Commented Lines (#)	Any line that starts with the "#" character is considered a comment line which is ignored during evaluation.

The SQL_COMMANDSTORE line

The maximum number of stored SQL statements (when database connection is lost) can be configured through this line. Any SQL statements arriving after the respective limit is reached will be discarded if the database connection has not resumed yet:

```
SQL_COMMANDSTORE<TAB>Value
```

Parameter	Description
SQL_COMMANDSTORE	The type of the line, predefined keyword
Value	A positive integer representing the desired number of SQL commands that may be stored.

The SQL_TIMESTAMPFORMAT line

The SQL_TIMESTAMPFORMAT allows defining the time and date formats that are to be used when the timestamp placeholders are expanded and their values written to the database:

```
SQL_TIMESTAMPFORMAT<TAB>DateFormat<SPACE>TimeFormat
```

Parameter	Description
SQL_TIMESTAMPFORMAT	The type of the line, predefined placeholder.
DateFormat<SPACE>TimeFormat	This line section allows defining the actual date and time formats. This row can be any combination of the following values: • *Time format*: `HH:MM:SS pp, HH:MM:SS.FFF pp, HH:MM:SS.FFF, HH:MM:SS` • *Date format*: `mm/dd/YYYY, dd-mm-YYYY, dd.mm.YYYY, dd/mm/YYYY, YYYY-mm-dd, mm-dd-YYYY`

The DB line

The DB line describes the DSN parameters used in the ODBC connection. The DB line has the following format:

```
DB<TAB>Database ID<TAB>DSN<TAB>IP Address or Host name<TAB>ODBC connection name<TAB>U
```

Parameter	Description
DB	The type of the line, predefined keyword.
Database ID	Unique database identifier within the task file (used for matching the INIT line and the statements in the corresponding subscription file).
DSN	The type of the database driver, predefined keyword if ODBC connection.
IP Address or Host name	Specifies the location of the ODBC connection. Generally this parameter will have the *localhost* value.
ODBC Connection name	The name of ODBC connection as it is saved in the ODBC Manager.
User name	Specifies what user name to use for connecting to the DSN connection. This parameter holds the value of

© Softing Industrial Automation GmbH

	the *User Name* field in the Database Wizard - Authentication page.
Password	Specifies what password to use for connecting to the DSN connection. This parameter holds the value of the *Password* field in the Database Wizard - Authentication page.

Examples:

```
DB    MyDatabase_0      DSN   localhost   ODBC to MSSQL     sa    sa
DB    MyDatabase_1      DSN   localhost   ODBC to MySQL
DB    MyDatabase_2      DSN   localhost   ODBC to Oracle    TESTADMIN   TESTADMIN
DB    MyDatabase_3      DSN   localhost   ODBC to PostgreSQL      postgres    postgre
DB    MyDatabase_4      DSN   localhost   ODBC to Oracle    SYSADMIN    SYSADMIN
```

Note: If the SQL_COMMANDSTORE value is not null, then SQL files are created for each DB line from the Tasks file at the start-up of the local dataFEED OPC Suite application.

The INIT line

The INIT line can be used to specify an initialization SQL command for any of the database connections. This statement will be executed once, immediately after the local dataFEED OPC Suite starts:

```
INIT<TAB>Database ID<TAB>SQL statement
```

Parameter	Description
INIT	The type of the line, predefined keyword.
Database ID	Unique database identifier within the task file (used for matching the DB line and the statements in the corresponding subscription file).
SQL statement	A database initialization SQL command that is executed once, immediately after the startup of the local dataFEED OPC Suite application.

Examples:

```
INIT<TAB>MyDatabase_4<TAB>ALTER SESSION SET NLS_LANG =
JAPANESE_JAPAN.JA16EUC;
```

Note: The *Database ID* must be defined in the previous DB definition line.

Task File - Example:

```
#     Database Add-on - configuration
SQL_COMMANDSTORE  41
SQL_TIMESTAMPFORMAT     mm-dd-YYYY HH:MM:SS
#
DB    MyDatabase_0      DSN   localhost   ODBC to MSSQL     sa    sa
DB    MyDatabase_1      DSN   localhost   ODBC to MySQL
DB    MyDatabase_2      DSN   localhost   ODBC to Oracle    TESTADMIN   TESTADMIN
DB    MyDatabase_3      DSN   localhost   ODBC to PostgreSQL      postgres    postgre
DB    MyDatabase_4      DSN   localhost   ODBC to Oracle    SYSADMIN    SYSADMIN
INIT  MyDatabase_4      ALTER SESSION SET NLS_LANG = JAPANESE_JAPAN.JA16EUC;
#
```

Subscription Configuration File

The *Data Destination - Database* generates multiple line types:

- common lines which are specific to all the consumer add-ons: condition lines, subscription lines and item lines;
- specific lines: modifier lines and copy lines.

Common Lines:

The following line types are common lines and will be used to configure different functionalities:

Line Type	Description
Subscription	Defines the update rate with which the current action will be executed.
Condition	Defines the execution condition for the current action.
Wildcard	Defines a regular expression which is matched against the items defined in the subscription file.
Item	This line defines a trigger which is fired if the item data has changed. The trigger data includes all other items which have changed at the same time (were reported in the same data change callback from the OPC server). An item data change consists of a different value or quality of the respective item compared to the previous one.
Timer	This line defines a trigger which is fired at specific time and date or at specific intervals.

© Softing Industrial Automation GmbH

Datachange	This trigger is fired each time an item data change occurs (for each incoming transaction). At most one Datachange trigger may exist per subscription file. If an internal action has a Datachange trigger, it will receive the data changes of all subscriptions of the local dataFEED OPC Suite application. This includes dataFEED OPC Suite internal subscriptions and subscriptions created by the connected OPC clients.
Commented Lines (#)	Any line that starts with the "#" character is considered a comment line which is ignored during evaluation.

The subscription file is divided into two sections:

- The global area:

  The global area extends from the first subscription file line to the first trigger line (e.g: ITEM, TIMER or DATACHANGE lines).

  This area contains definitions valid for the entire subscription file.

- The local area:

  The local area covers all the actions between two trigger lines in the subscription file.

  The lines in this area are only executed if the parent trigger fires.

Subscription Line

A subscription line has the following format:

```
SUBSCRIPTION<TAB>Subscription ID<TAB>UPDATERATE<TAB>Update rate value<TAB>ACTIVE<TAB>
```

The subscription line contains the following configurable parameters:

Parameter	Description
SUBSCRIPTION	The type of the line.
Subscription ID	It is used to identify the subscription and must be unique.
UPDATERATE	Predefined keyword.
Update rate value	Defines the update-rate with which the current action will be executed.
ACTIVE	Predefined keyword.

Active value	Active value which can be TRUE or FALSE.

The subscription lines always contain the global subscription line as well as all subscriptions which are used at least once in consumer actions. The global subscription is represented by the subscription line with the empty *Subscription ID*. The default update rate value for the global subscription is *1000* milliseconds.

The value of the update rate is saved in milliseconds. This value may not be set to 0 inside the dataFEED OPC Suite Configurator but it can be manually set by editing the subscription file with an external text editor. An update rate value of 0 means *as fast as possible* but this is not recommended from a resource-related point of view.

Setting a subscription to be active or inactive is an advanced feature which cannot be configured from within the dataFEED OPC Suite Configurator. By default the active value for each and every subscription is set to *TRUE* (meaning that all subscriptions are active since creation). If a subscription's active value is set to *FALSE* then the data changes of the items configured to be part of that subscription will be ignored.

Attention: From a resource-related point of view it is recommended to remove the unused subscription definitions if the DATACHANGE trigged is to be used.

Examples:

```
SUBSCRIPTION            UPDATERATE  1000  ACTIVE      TRUE
SUBSCRIPTION      Subscription_300  UPDATERATE  300   ACTIVE      TRUE
```

Condition Line

A condition line defines whether an action below an item trigger will be executed or not (when the item trigger is fired). This can be used in other configuration lines as well to condition the execution of an item action. One example of this is the conditional execution of a SQL statement in the Database add-on. A condition line has the following format:

```
CONDITION<TAB>Condition ID<TAB>Condition Statement
```

The condition line contains the following configurable parameters:

Parameter	Description
CONDITION	The type of the line, predefined keyword.
Condition ID	Unique condition identifier.
Condition statement	Single condition statement.

Conditions can be used to react to communication errors. Various actions can be executed depending on the quality of items.

The following condition statements are supported:

© Softing Industrial Automation GmbH

Condition	Description
AllGoodQuality	The trigger actions are executed only if all items in the transaction have the quality *Good*.
Always	The trigger actions are executed always, ignoring the quality of the items in the incoming transaction.
ItemQualityChanged	The trigger actions are executed only if the quality of at least one item in the transaction has changed.
OneBadQuality	The trigger actions are executed if exactly one item in the transaction has the quality *Bad*.

The condition lines always contain the global condition line as well as all conditions which are used at least once in consumer actions. The global condition is represented by the condition line with the empty *Condition ID*. The default condition statement value for the global condition is *Always*. As a general rule all the condition lines need to be declared in the global section of the subscription file. If no condition is defined the Always condition is applied as default.

Examples:

```
CONDITION          Always
CONDITION    AllGoodQuality     AllGoodQuality
```

Wildcard Line

The Subscription Item ID, Item ID and Event Source strings can include wildcards (* .? and []) as parts of the string.

A wildcard must be previously defined in a *Wildcard Line* before it can be used in the Action.

A wildcard line has the following format:

```
WILDCARD<TAB>Wildcard ID<TAB>Item type<TAB>Wildcard string
```

The wildcard line contains the following configurable parameters:

Parameter	Description
WILDCARD	The type of the line, predefined keyword
Wildcard ID	Unique wildcard identifier.
Item type	The item type can be used to specify which item should be used.

Wildcard string	The regular expression.

The following item types are allowed:

Item type	Description
Current	The item must be in the subscription item list. The value used is the current one, which means that if the value for the item is included in the current transaction, this value will be used. If the value is not included in the incoming transaction, the cached value will be used.
Changed	The item must be in the subscription item list. The value used is the changed one, which means that the entire action will be invalidated if the item is not present in the incoming transaction.
Cache	The item must be in the subscription item list. The value used is the cached one, even if that item is included in the incoming transaction.

Note: The data value defined for wildcards in the wildcard lines overrides the functionality of the data value defined in the item placeholders found in the subscription file.

The asterisk wildcard (*) substitutes any sequence of characters of any length:

- SID* - replaces any string of at least 3 characters that starts with SID
- S*D – replaces a string of at least 2 characters that starts with S and ends with D

The question mark wildcard (?) substitutes a single character:

- SID? – replaces any string of 4 characters that starts with SID.
- S?D – replaces any string of 3 characters that starts with S and ends with D.

The range wildcard ([ ]) substitutes any single character in the range specified between the square brackets:

- S[A-Z]D – replaces any string of 3 characters that starts with S, ends with D and contains any character in the range A to Z.
- SID[0-9] – replaces any string of 4 characters that starts with SID and ends with any number in the range 0 to 9.

Note: The format of the range wildcard is the following: [rangeStart-rangeEnd]. Please note that the delimiter is the hyphen-minus (-) character.

© Softing Industrial Automation GmbH

Note: If the percent sign (%) is to be included in a statement, note that %% is used as substitute for a single percent sign %.

Example:

```
WILDCARD    All_Integer_Increments  Current     Increment.I*
```

Attention: The wildcard can only replace the items defined in the subscription file.

Item Line

An item line is used to add an item to a subscription. The associated actions are executed each time the value or the quality of the specified data items changes during the run time of the local dataFEED OPC Suite application.

An item line has the following format:

```
ITEM<TAB>Subscription Item ID<TAB>Server Item ID<TAB>Subscription ID
```

The item line contains the following configurable parameters:

Parameter	Description
ITEM	The type of the line, predefined keyword.
Subscription Item ID	Unique subscription item identifier (implicitly created by concatenating the Server Item ID and the Subscription ID with the underscore character in between them).
Server Item ID	Item identifier within the local dataFEED OPC Suite application address namespace. This could be a server connection item ID or a local item ID.
Subscription ID	Subscription identifier (defined above, see Subscription Line).

This trigger is fired if the item data or quality has changed (also referred to as data change). The trigger data includes all other items which were included in the data change (items which had their data change reported in the same transaction).

Note: A single line consisting of the *Server Item ID* and *Subscription ID* is added to the subscription file for an item used several times with the same update rate in any consumer add-ons. This optimizes the local dataFEED OPC Suite application throughput.

Example:

```
ITEM  increment.I1_Subscription_300 increment.I1      Subscription_300
```

Note: The subscription item ID can be empty. In this case, however, only the Server Item ID can be used to identify the item.

Timer Line

This trigger is fired at a specified time and date or at specific intervals. The associated actions are executed on a specific schedule or periodically during the run time of the local dataFEED OPC Suite application.

The format of the *Timer Line* is:

```
TIMER<TAB>Timer ID<TAB>INTERVAL<TAB>Rate in milliseconds
```

or

```
TIMER<TAB>Timer ID<TAB>SCHEDULE<TAB>Schedule range<TAB>First execution<TAB>Rate<TAB>L
```

The interval timer contains the following configurable parameters:

Parameter	Description
TIMER	The type of the line, predefined keyword.
Timer ID	Unique timer identifier.
INTERVAL	Denotes an interval timer trigger, predefined keyword.
Interval value	Defines the interval in milliseconds at which the timer will trigger the associated action(s).

The timer schedule contains the following configurable parameters:

Parameter	Description
TIMER	The type of the line, predefined keyword
Timer ID	Unique timer identifier.
SCHEDULE	Denotes a scheduled timer trigger, predefined keyword.
Schedule range	Defines the range of the schedule (see below the table).
First execution	Defines the time the timer should begin firing events (see below the table).

© Softing Industrial Automation GmbH

Rate	Defines the rate at which the timer should trigger the associated action(s).
Last execution	Defines the time the timer should stop firing events (see below the table).
List Of Days	Should be used only with the Week and Month schedule. It must contain the days on which the execution is desired. If the list of days is not present, the obtained functionality is the same as by using the day marker in every day of the week/month.

The timer's *Schedule range* may have one of the following values:

- MINUTE
- HOUR
- DAY
- WEEK - List of days: 1d = Monday, 2d = Tuesday, …, 7d = Sunday.
- MONTH - List of days: 1d = First day of month, 2d = Second day of month, …, 30d = Thirtieth day of month, 31d = Thirty-first day of the month.

The *First execution* defines the time the timer should begin firing events.

Example:

```
1d13h10m15s
```

Note: If parts of the execution time are 0, they do not have to be in the string, e.g. 1d0h10m0s = 1d10m

The timer *Rate* defines the event firing rate. An empty timer *Rate* means a single execution.

Last Execution: The last execution time of the schedule. Could be empty.

List of Days: Shall be used only with the Week and Month schedule. It should contain the days on which execution is desired. If the list is not specified the same functionality can be achieved by using the day marker in the first and last execution time.

Note: If the List of Days and the day marker in first and last execution time are used in the same timer trigger the two setting types should match the marked days!

Examples:

```
# Following timer fires events every 250 milliseconds (4 times a second):
TIMER       INTERVAL    250
# Following timer fires events every 20 seconds of a minute starting at second 10 (eg
TIMER       SCHEDULE    MINUTE      10s   20s
# Following timer fires events every 4 hours of a day starting at 6:30, ending at 20:
# (e.g. 6:30, 10:30, 14:30, 18:30).
TIMER       SCHEDULE    DAY   6h30m 4h    20h
```

```
# Following timer fires events every 4 hours of a day starting at 6:30, ending at 20:
# Friday (e.g. 6:30, 10:30, 14:30, 18:30 of each day).
TIMER       SCHEDULE    WEEK  6h30m 4h    20h   1d2d3d4d5d
# Following timer fires events every 4 hours starting at 6:30, ending at 22:45 (e.g.
# 18:30, 22:30), on the 1th, 2th, 10th, 20th of every month.
TIMER       SCHEDULE    MONTH 6h30m 4h    22h45m      1d2d10d20d
# Following timer fires events every 1 hours, 3 minutes and 3 seconds starting at 00:
# 23:59:59 (e.g. 00:00:01, 01:03:04, 02:06:07 etc), on the 3rd, 6th, 9th, 12th, 15,th
TIMER Timer_02    SCHEDULE    MONTH 1s    1h3m3s      23h59m59s   3d6d9d12d15d18d21d
```

Datachange Line

This trigger is fired on every data change of an item, regardless of the subscription the incoming transaction belongs to. The format of a Datachange line is the following:

```
DATACHANGE
```

Note: A single Datachange trigger may exist per subscription file.

The associated actions are executed if the value or the quality of one of the currently used data processing or destination data items changes during the run time of the local dataFEED OPC Suite application.

Attention: The Datachange trigger will be fired for each configured subscription rate (when an incoming transaction of at least one item arrives).

Note: Make sure that only used subscription definitions are present in the configuration file (resource-related point of view).

Specific Lines:

The following line types are Database specific lines and will be used to configure different functionalities:

Line Type	Description
Database (SQL) line	Defines a SQL statement.

Database (SQL) Line

The format of the SQL line is the following:

```
SQL<TAB>Condition ID<TAB>Database ID<TAB>SQL statement
```


© Softing Industrial Automation GmbH

The SQL line contains the following configurable parameters:

Parameter	Description
SQL	The type of the line, predefined keyword.
Condition ID	Condition identifier (defined above, see Condition Line).
Database ID	Database definition identifier (defined above, see DB Line).
SQL statement	A SQL statement (can be an *Insert Into* statement, a *Stored Procedure* call or any *General SQL* statement).
Trigger ID	Associated trigger identifier (one of the above defined: Item Trigger, Timer Trigger or Datachange Trigger)

The SQL statement parameter defines the SQL instruction which will be executed on the associated database when the trigger fires the event. It may contain plain text and also data placeholders (see the Data Values Placeholder chapter for mode details).

An example of SQL configuration lines and the appropriate item lines is presented below:

```
# Insert statements with the default condition
ITEM  OPC_3.increment.R8_     OPC_3.increment.R8
SQL         MyDatabase_0      INSERT INTO Test_Table1 (itemId,itemValue,itemQuality)
ITEM  OPC_3.increment.R8_     OPC_3.increment.R8
SQL         MyDatabase_0      use Osf_tests GO INSERT INTO Test_Table (itemId,itemVal
# Insert statement with condition AllGoodQuality (not defined in the current example)
ITEM  OPC_3.increment.R8_     OPC_3.increment.R8
SQL   AllGoodQuality    MyDatabase_0      INSERT INTO Test_Table1 (itemId,itemValue,i
```

When executed, these statements will insert the ID, the value and the quality of the subscription item *increment.R8* from a *OPC_3* prefixed server connection into the configured database.

In order to offer a flexible configuration for SQL statements, placeholders can also be used. The placeholders are syntactical entities that are replaced with their associated values each time the statement is evaluated during the run time of the local dataFEED OPC Suite application (before the instruction is sent to the database).

Note: The *Condition ID* used in SQL line definition must be defined in the current Subscription File; the default condition may also be used.

Note: The *Database ID* used in a SQL line definition must be defined in the associated Task File.

The SQL statements will be executed on every data change event, however their execution will also depend on both the defined conditions and the type of the fired trigger.

If the statements are associated with triggers of type *Datachange*, the respective statements are evaluated on every data change and are only executed if all of the following conditions are satisfied:

- At least one item from the SQL statement changes.
- The action condition is true.
- The placeholders in the statement can be replaced by their respective values (the changed data source may invalidate the action).

If the statements are associated with triggers of type *Item*, the respective statements are evaluated on each item change, but are only executed if all of the following conditions are true:

- The subscription item ID of the statement has to be defined in the configuration.
- A change occurs for the respective item.
- The action condition is true.
- The placeholders in the statement can be replaced by their respective values (the changed data source may invalidate the action).

If the statements are associated with triggers of type *Timer*, the respective statements are evaluated and executed only if all of the following conditions are true:

- The subscription item ID of the statement has to be defined in the configuration.
- The action condition is true.
- The placeholders in the statement can be replaced by their respective values (the changed data source may invalidate the action).

As a general rule, if a statement includes placeholders with wildcards, at least one of the wildcards must match one of the items that changed. If a statement that includes items which are not in the data changed list should be executed, the values of these items are retrieved from the cache.

Deactivated Database functionality

When the Database functionality is deactivated, all the lines that are used exclusively by the Database add-on will be preceded by the *#!* characters. Therefore, the database configuration is not deleted but only disabled and will be ignored by the local dataFEED OPC Suite application at run time. When the Database functionality is reactivated, the *#!* character groups will be removed and the database lines will not be ignored anymore by the local dataFEED OPC Suite application.

Example:

```
CONDITION         Always
CONDITION   AllGoodQuality    AllGoodQuality
#
SUBSCRIPTION            UPDATERATE  1000  ACTIVE      TRUE
#!SUBSCRIPTION    Subscription_222  UPDATERATE  222   ACTIVE      TRUE
#
WILDCARD    Wildcard_01 Current     OPC_1.increment.*
#
```


© Softing Industrial Automation GmbH

```
CONDITION         Always
CONDITION   AllGoodQuality    AllGoodQuality
#
SUBSCRIPTION            UPDATERATE  1000  ACTIVE      TRUE
#!SUBSCRIPTION    Subscription_222  UPDATERATE  222   ACTIVE      TRUE
#
MODIFIER          V
#
#
#!ITEM      OPC_3.increment.R8_Subscription_222 OPC_3.increment.R8      Subscription_
#!SQL AllGoodQuality    MyDatabase_0      INSERT INTO Test_Table1 (itemId,itemValue,i
('#%-ItemItemID%OPC_3.increment.R8_Subscription_222%Current-%#','#%-ItemValue%OPC_3.i
'#%-ItemValue%Wildcard_01%Current-%#')    OPC_3.increment.R8_Subscription_222
#
#!SQL       MyDatabase_0      use Osf_tests GO INSERT INTO Test_Table (itemId,itemVal
(#%-ItemItemID%OPC_3.increment.R8_%Current-%#, #%-ItemValue%OPC_3.increment.R8_%Curre
#
#!TIMER     Timer_01    INTERVAL    1000
#
#!TIMER     Timer_02    SCHEDULE    MONTH 1s    1h3m3s      23h59m59s   3d6d9d12d15d1
#!SQL       MyDatabase_0      exec Procedure_update_on @param1=#%-ItemValue%OPC_3.inc
 @param3=#%-ItemTimestampMS%OPC_3.increment.R8_%Current-%#; Timer_02
```

Subscription File - Example:

```
CONDITION         Always
CONDITION   AllGoodQuality    AllGoodQuality
#
SUBSCRIPTION            UPDATERATE  1000  ACTIVE      TRUE
SUBSCRIPTION            Subscription_222  UPDATERATE  222   ACTIVE      TRUE
#
ITEM  OPC_1.increment.R8_     OPC_1.increment.R8
#
ITEM  OPC_1.increment.R8_Subscription_222 OPC_1.increment.R8      Subscription_222
#
ITEM  OPC_3.increment.R8_Subscription_222 OPC_3.increment.R8      Subscription_222
SQL   AllGoodQuality    MyDatabase_0      INSERT INTO Test_Table1 (itemId,itemValue,i
#
DATACHANGE
SQL         MyDatabase_0      use Osf_tests GO INSERT INTO Test_Table (itemId,itemVal
#
TIMER Timer_02    SCHEDULE    MONTH 1s    1h3m3s      23h59m59s   3d6d9d12d15d18d21d
SQL         MyDatabase_1      exec Procedure_update_on @param1=#%-ItemValue%OPC_1.inc
```

7.3.3.5.9 Multiple Configuration Files

This section handles special cases of multiple configuration files – *Subscription* and *Task* files – which were fully supported in the previous version. The V5.38 version of the dataFEED OPC Suite Configurator does not support multiple subscription or task files and will therefore only load the default Task and Subscription files. This is specified in the *OSF_Configuration.xml* configuration file:

```
<Tasks>
```

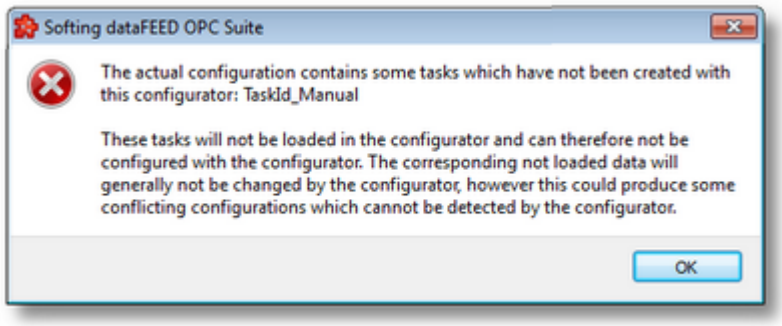
```
  <Task ID="TaskId_Default">
    <TaskFile Path=".\Tasks\Tasks.txt" />
    <Libraries>
      <Library Path=".\OSF_AO_Database.dll" Type="Database"
      ID="LibraryID_Database" />
    </Libraries>
    <Subscriptions>
      <Subscription Type="DA" ID="SubscriptionId_Default">
        <SubscriptionFile Path=".\Tasks\Subscriptions.txt" />
      </Subscription>
    </Subscriptions>
  </Task>
</Tasks>
```

It contains the default task ID *TaskId_Default* and the default subscription ID *SubscriptionId_Default*. When multiple task or subscription files are needed, they can be defined in the <Tasks> section of the XML configuration file by creating a new <Task> section and assigning it another ID, dissimilar to the default one. The same applies to the subscription files.

```
<Tasks>
  <Task ID="TaskId_Default">
    <TaskFile Path=".\Tasks\Tasks.txt" />
    <Libraries>
      <Library Path=".\OSF_AO_Database.dll" Type="Database"
      ID="LibraryID_Database" />
    </Libraries>
    <Subscriptions>
      <Subscription Type="DA" ID="SubscriptionId_Default">
        <SubscriptionFile Path=".\Tasks\Subscriptions.txt" />
      </Subscription>
    </Subscriptions>
  </Task>
  <Task ID="TaskId_01">
    <TaskFile Path=".\Tasks\Task_01.txt" />
    <Libraries>
      <Library Path=".\OSF_AO_Database.dll" Type="Database"
      ID="LibraryID_Database" />
    </Libraries>
    <Subscriptions>
      <Subscription Type="DA" ID="SubscriptionId_01">
        <SubscriptionFile Path=".\Tasks\Subscription_01.txt" />
      </Subscription>
    </Subscriptions>
  </Task>
</Tasks>
```

When loading such a configuration file into the dataFEED OPC Suite Configurator a specific warning message will be displayed. This is not a load failure but informs you that the unloaded tasks and subscriptions will not be configured by the configurator:

© Softing Industrial Automation GmbH



Note: When multiple tasks and subscriptions exist in one configuration file, proper measures must be taken so that the default task and subscription are valid in the default context and can be executed correctly by the local dataFEED OPC Suite application.

Support of Old Database Connection Definitions

The *Multiple Configuration Files* case is a solution when direct database connection is desired instead of using a DSN connection. In such cases a new task file and a new subscription file must be manually defined and the direct database connections must be defined inside the task file:

```
SQL_COMMANDSTORE<TAB>100
#
DB<TAB>MyDB1<TAB>MySQL ODBC 5.1 Driver<TAB>192.168.150.110<TAB>Opc_test<TAB>root<TAB>
INIT<TAB>MyDB1<TAB>USE opctablett
#
DB<TAB>MyDB2<TAB>SQL Server Native Client 10.0<TAB>192.168.150.110<TAB>Osf_tests<TAB>
INIT<TAB>MyDB2<TAB>USE Osf_tests
#
DB<TAB>MyDB3<TAB>Oracle in OraClient11g_home1<TAB>192.168.150.110<TAB><TAB>TESTADMIN<
INIT<TAB>MyDB3<TAB>USE Table_complex
```

The subscription file must contain the required database actions:

```
ITEM<TAB>OPC_1.increment.I1_Subscription_1000<TAB>OPC_1.increment.I1<TAB>Subscription
SQL<TAB>AllGoodQuality<TAB>MyDB2<TAB>INSERT INTO opctablett ( ItemValue, ItemQualityS
( '#%-ItemValue%OPC_1.increment.I1_Subscription_1000%Current-%#',
  '#%-ItemQuality%OPC_1.increment.I1_Subscription_1000%Current-%#',
  '#%-ItemTimestampMS%OPC_1.increment.I1_Subscription_1000%Current-%#')<TAB>OPC_1.inc

SQL<TAB>ItemQualityChanged<TAB>MyDB3<TAB>INSERT INTO opctablett ( ItemValue, ItemQual
( '#%-ItemValue%OPC_1.increment.I4_Subscription_300%Current-%#',
  '#%-ItemQuality%OPC_1.increment.I4_Subscription_300%Current-%#',
  '#%-ItemTimestampMS%OPC_1.increment.I4_Subscription_300%Current-%#')<TAB>OPC_1.incr
```

MyDB2 and *MyDB3* are defined in the above task file.

In this particular case nothing must be changed for the SQL line. Changes appear only in the newly created task file, in the DB line, which will be defined based on the old principles used in versions V1.4x. A short description of that format is shown below:

```
DB<TAB>Database ID<TAB>Driver<TAB>IP Address or Host name<TAB>Database Name<TAB>User
```

Parameter	Description
DB	The type of the line, predefined keyword.
Database ID	Unique database identifier within the task file (used for matching the INIT line and the statements in the corresponding subscription file).
Driver	Driver used for database connection. Examples: Oracle in OraClient11g_home1, SQL Server Native Client 10.0 or MySQL ODBC 5.1 Driver
IP Address or Host name	Specifies the database location. In general this parameter can be an IP Address or a Host Name.
Database Name	The name of the database.
User Name	Specifies what user name to use for connecting to the database server.
Password	Specifies the password for the corresponding user name.

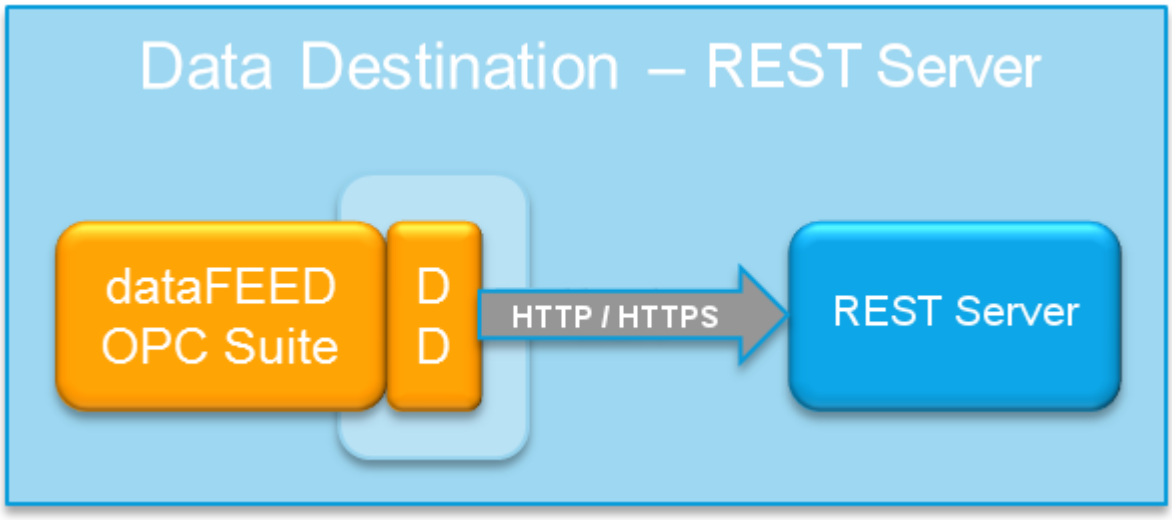
Examples:

```
DB<TAB>MySQL_DB<TAB>MySQL ODBC 5.1 Driver<TAB>192.168.150.110<TAB>Opc_test<TAB>root<T
DB<TAB>MMSQL_DB<TAB>SQL Server Native Client 10.0<TAB>192.168.150.110<TAB>Osf_tests<T
DB<TAB>Oracle_DB<TAB>Oracle in OraClient11g_home1<TAB>192.168.150.110<TAB><TAB>SYSADM
```

7.3.3.6 REST Server

This section of the Softing dataFEED OPC Suite online help explains how a REST Client can be configured in order to send data to a REST Server using HTTP and HTTPS protocols.

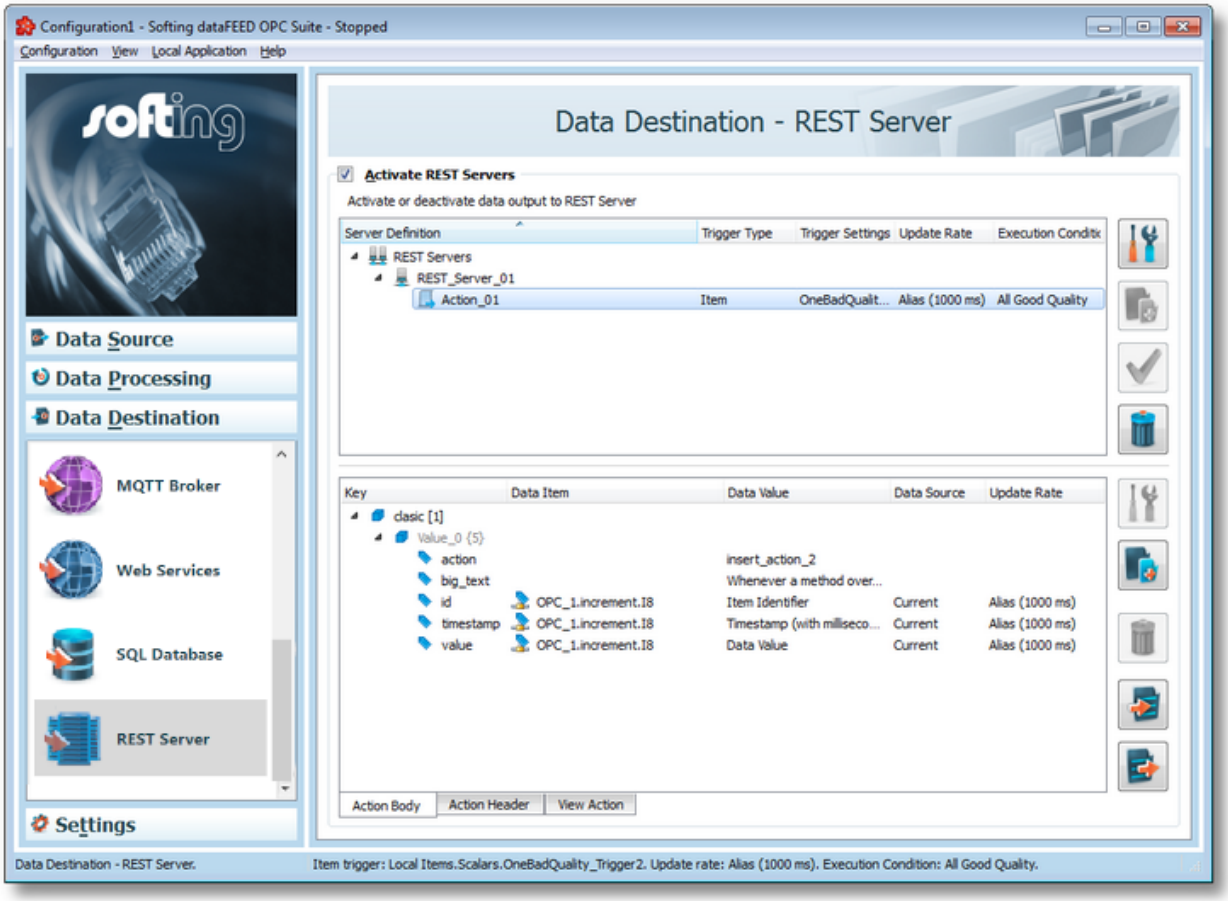
© Softing Industrial Automation GmbH



A server built using REST (Representational State Transfer) architecture most commonly uses the HTTP/HTTPS protocol to communicate with clients. GET, POST, PUT and other HTTP operations are used to query, create and manipulate resources and content on the server.

7.3.3.6.1 Configuration

In order to configure connections to REST servers, expand the **Data Destination** section in the left panel and then select the REST Server icon. The REST Server connection overview will be displayed.



Add a REST Server connection

A new REST Server connection can be defined by using the *Add* button or the menu command "Add REST Server Definition" in the context menu. The REST Server connection wizard will show up. The Ctrl+A keyboard shortcut has the same effect.

Modify an existing REST Server connection

An existing connection to a REST Server can be modified. To do so, select the line in the connection overview that contains the connection to be modified. Then the *Properties* button or the "Properties" command from the context menu can be used to open the REST Server connection wizard.

Add an action definition to an existing REST Server connection

Adding a new action to a selected server connection is accomplished by pressing the *Add* button or the "Add REST Server Action" context menu item. Also, the Ctrl+A key combination is

© Softing Industrial Automation GmbH

available for this operation. The REST Action wizard will be opened where you can go through different wizard pages to specify action details.

Modify the action settings of an existing REST Server connection

To modify an existing action, select one from the list and by double clicking it, or pressing the *Properties* button, the REST Action wizard will be opened, and you will be able to go through the same wizard pages as when the action was created.

Connection test

Click the *Connection Test* button to check the availability of the selected server or the correctness of the authentication and security settings. If the connection cannot be established the error message and code returned by the network or the server will be shown in the dialog.

Remove a connection/action

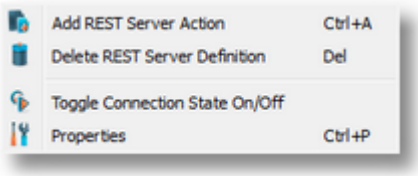
Delete the selected server connection or action definition by using the *Delete* button, pressing the DEL / DELETE key on the keyboard or selecting the appropriate delete operation from the context menu.

Note: If no connection is selected, the *Add a HTTP REST Server definition* and *General Settings* buttons are the only options enabled.

The *Delete* button will remove the selected connection or action from the list.

A dialog will then prompt the user to confirm the deletion.

The following image shows an example of a context menu:



Action body, header and action overview

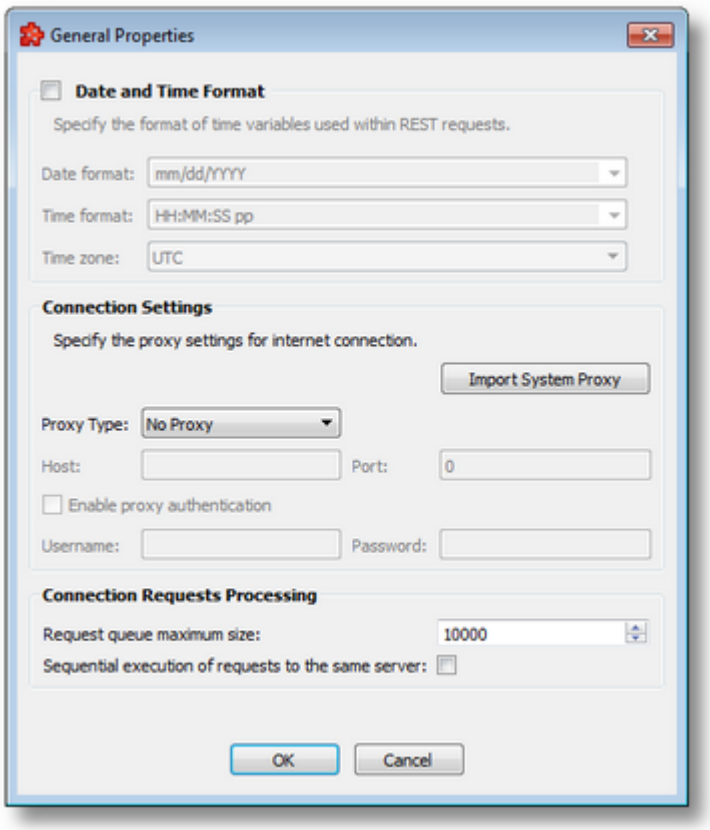
When an action is selected, the main view page can also show the action body and header definition, as well as an overview of the final action content with item placeholders, which the Runtime will replace when it's run.

The action body, header and overview can be seen in the three different tabs in the main screen, below the definition list: *Action Body, Action Header, View Action*.

This lower part of the main view contains tab pages which can also be found in the action wizard and allow easier access for the user to change the action body and header contents without going through the wizard pages. More information can be found on the specific help pages.

Time Format and Connection Settings

When the *REST Servers* root node is selected, invoking the properties action by any means (button, context menu or keyboard shortcut) displays a dialog that allows the user to modify the time format setting. The *General Properties* dialog can be opened by using the *Properties* button  , by pressing the keyboard shortcut CTRL+P or by selecting the *Properties* action from the context menu.



The **Date and Time Format** section handles the different formats which can be specified when items of date and time are used in actions, also the source localization time: UTC (Universal Time Coordinated) or local time (local time localization). The default time zone is UTC.

The default values are `mm/dd/YYYY` for the *Date format and* `HH:MM:SS pp` for the *Time format*. Other defined values for Time and Date format are:

- *Date format*: `mm/dd/YYYY`, `dd-mm-YYYY`, `dd.mm.YYYY`, `dd/mm/YYYY`, `YYYY-mm-dd`, `mm-dd-YYYY`

© Softing Industrial Automation GmbH

- *Time format*: `HH:MM:SS pp, HH:MM:SS.FFF pp, HH:MM:SS.FFF, HH:MM:SS`

In addition to the default values listed in the format selection combo boxes, the user can add his/her own custom format. Custom Date and Time format strings are accepted as valid, when the format strings contain in their composition the following accepted format substrings and format delimiters. These formats allow the inclusion of specific locale information for the regional settings.

- *Valid Date format strings*: `AAAAAA, aaa, BBBBBBBB, bbb, ccccccccccccccccccc, dd, jjj, mm, UU, ww, xxxxxxxxxx, yy, YYYY`

- *Valid Time format strings*: `II, HH, MM, SS, pp, XXXXXXXX, FFFFFF, FFFFF, FFFF, FFF ,FF, F`

- *Valid Date and Time format delimiters*: `<dot>, <colon>, <slash>, <comma>, <backslash>, <vertical bar>, <semicolon>, <space>, <tab>, <minus>`

The **Connection Settings** section allows for the customization of proxy settings. The user can either import system proxy settings automatically (*Import System Proxy* button), or specify manually the host, port and authentication settings.

These settings will be applied to all connections in the plugin, and the following parts of the functionality will be affected: test connection, getting the server certificate and action execution. This means that if the network, on which the dataFEED is configured, is behind a proxy server and dataFEED is not configured with a proxy, like in the image above, the mentioned pieces of functionality will fail. In order for them to work correctly, the user needs to specify the type of proxy, the host name of the server, a port number and user credentials, if necessary.

The **Connection Requests Processing** section contains settings for the internal request queue used by dataFEED to manage requests. The maximum queue size and execution type can be specied. The default queue size is 10000 (minimum is 100, maximum is 60000) and execution type is parallel (checkbox unchecked). The user can change these values by entering different numbers for the queue size and by enabling sequential execution using the checkbox. When the queue starts filling up, memory consumption of the application grows and when the application is able to consume requests from the queue quickly, memory consumption decreases.

The queue is used to hold requests which are still waiting for execution because actions are executed faster than the destination server can handle. In some cases, proxy servers may cause communication lags through the network, which may also slow down server replies, thus leading to the queue filling up with pending requests.

Special system items are available for the user to keep track of the number of sent requests, failed requests, pending requests in the queue, number of stored requests in case the Store and Forward mechanism is enabled, and dropped requests, in case server certificates are invalid, rejected or for some reason the server cannot accept the sent content. System items are available per action and also totals are calculated per connections.

REST status codes

After the *CouchDB* server received a request, it will either respond with a status code if the request was successfully executed or with an error code if the request was not executed. If the dataFEED OPC Suite application receives a non-success error/status code, then it will retry or

drop the request depending on the received error/status code.

The following table contains all the possible error/status codes with a brief description and the action that dataFEED OPC Suite application will perform.

Constant	Value	Description	Operation
ConnectionRefusedError	1	the remote server refused the connection (the server is not accepting requests)	Retried
RemoteHostClosedError	2	the remote server closed the connection prematurely, before the entire reply was received and processed	Droped
HostNotFoundError	3	the remote host name was not found (invalid hostname)	Droped
TimeoutError	4	the connection to the remote server timed out	Retried
OperationCanceledError	5	the operation was canceled via calls to abort or close before it was finished.	Droped
SslHandshakeFailedError	6	the SSL/TLS handshake failed and the encrypted channel could not be established.	Droped
TemporaryNetworkFailureError	7	the connection was broken due to disconnection from the network, however the system has initiated roaming to another access point. The request should be resubmitted and will be processed as soon as the connection is re-established.	Retried
NetworkSessionFailedError	8	the connection was broken due to disconnection from the network or failure to start the network	Droped
BackgroundRequestNotAllowedError	9	the background request is not currently allowed due to platform policy	Droped
ProxyConnectionRefusedError	101	the connection to the proxy server was refused (the proxy server is not accepting requests)	Droped
ProxyConnectionClosedError	102	the proxy server closed the connection prematurely, before the entire reply was received and processed	Droped
ProxyNotFoundError	103	the proxy host name was not found (invalid proxy hostname)	Droped
ProxyTimeoutError	104	the connection to the proxy timed out or the proxy did not reply in time to the request sent	Droped

© Softing Industrial Automation GmbH

ProxyAuthenticationRequiredError	105	the proxy requires authentication in order to honour the request but did not accept any credentials offered (if any)	Dropped
ContentAccessDenied	201	the access to the remote content was denied (similar to HTTP error 403)	Dropped
ContentOperationNotPermittedError	202	the operation requested on the remote content is not permitted	Dropped
ContentNotFoundError	203	the remote content was not found at the server (similar to HTTP error 404)	Retried
AuthenticationRequiredError	204	the remote server requires authentication to serve the content but the credentials provided were not accepted (if any)	Dropped
ContentReSendError	205	the request needed to be sent again, but this failed for example because the upload data could not be read a second time	Retried
ContentConflictError	206	the request could not be completed due to a conflict with the current state of the resource	Dropped
ContentGoneError	207	the requested resource is no longer available at the server	Dropped
InternalServerError	401	the server encountered an unexpected condition which prevented it from fulfilling the request	Dropped
OperationNotImplementedError	402	the server does not support the functionality required to fulfill the request	Dropped
ProtocolUnknownError	301	the server is unable to handle the request at this time	Dropped
ProtocolInvalidOperationError	302	the requested operation is invalid for this protocol	Dropped
UnknownNetworkError	99	an unknown network-related error was detected	Retried
UnknownProxyError	199	an unknown proxy-related error was detected	Retried
UnknownContentError	299	an unknown error related to the remote content was detected	Dropped
ProtocolFailure	399	a breakdown in protocol was detected (parsing error, invalid or unexpected responses, etc.)	Dropped
UnknownServerError	499	an unknown error related to the server response was detected	Dropped

The number of dropped and of pending request can be visible in the system items.

7.3.3.6.2 Connection Wizard

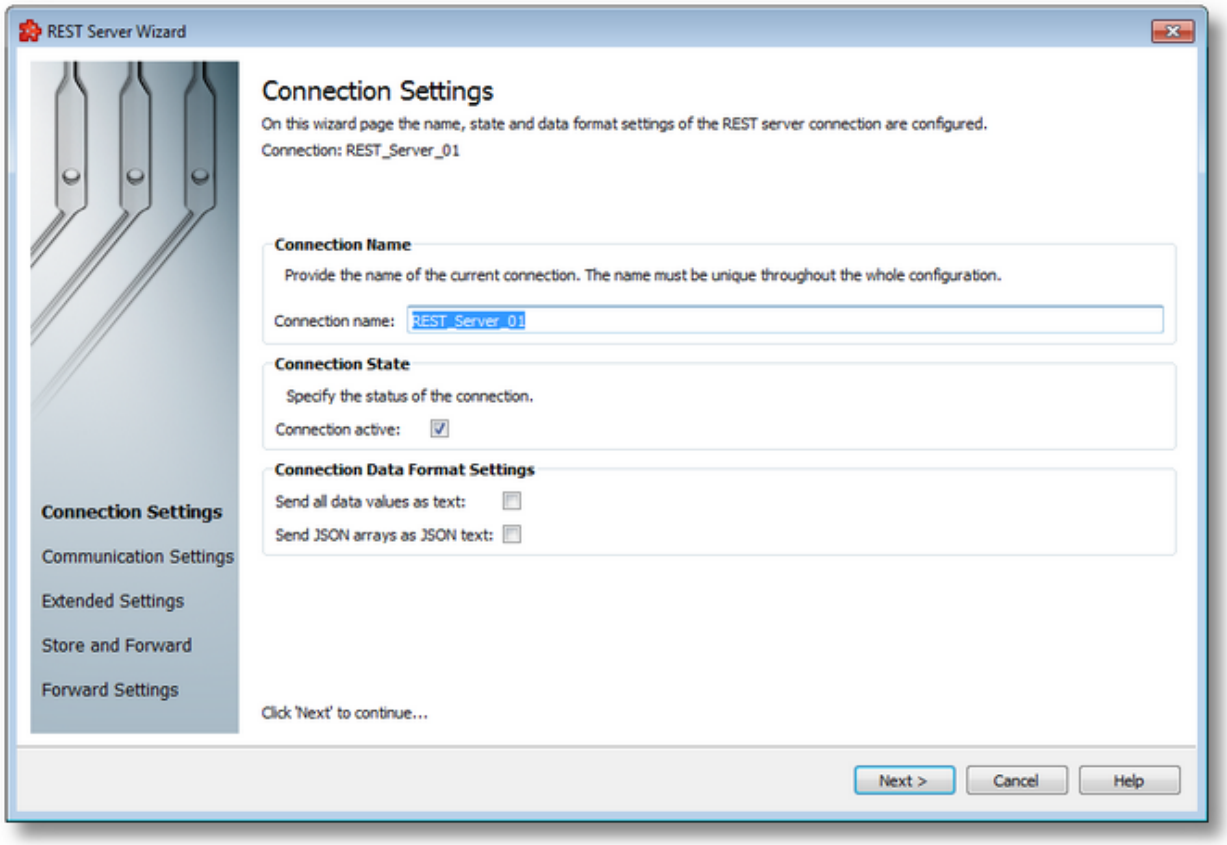
- With the Connection Wizard you can create a new connection definition or modify an existing one:

  - Define REST Connection settings @ Connection Wizard - Connection Settings
  - Define settings related to communication with server @ Connection Wizard - Communication Settings
  - Define additional settings for the server connection @ Connection Wizard - Extended Settings
  - Define store and forward settings on connection @ Connection Wizard - Store and Forward Settings
  - Define additional forward method settings on connection @ Connection Wizard - Forward Settings

Note: A wizard flow will not include all the pages listed above. Depending on the selections performed in some of the pages, different flow paths will be followed.

This page allows specifying connection settings such as a unique connection name and state for identifying the connection.

© Softing Industrial Automation GmbH



Connection Name

A unique identifier for the current connection. In case the provided name is already used by another connection, the user is not allowed to proceed to the next wizard page, until they provide a different name.

Connection State

This flag specifies if the current connection will be activated or not.

Connection Data Format Settings

The "*Send all data values as text*" checkbox enables the Runtime to surround all data item values with quotation marks ("). So, if this is checked integer values, doubles, etc. will be surrounded with quotes: "1" instead of 1, "3.14" instead of 3.14, and so on. String values will not be surrounded with an extra set of quotes, they will be sent simply as for example "apples and oranges", not ""apples and oranges"".

The *"Send JSON arrays as JSON text"* checkbox specifies whether or not the arrays should be surrounded with quotation marks after their placeholders are replaced by the runtime. It is important to note that in this case the whole array is sent as text/string, not its individual elements.

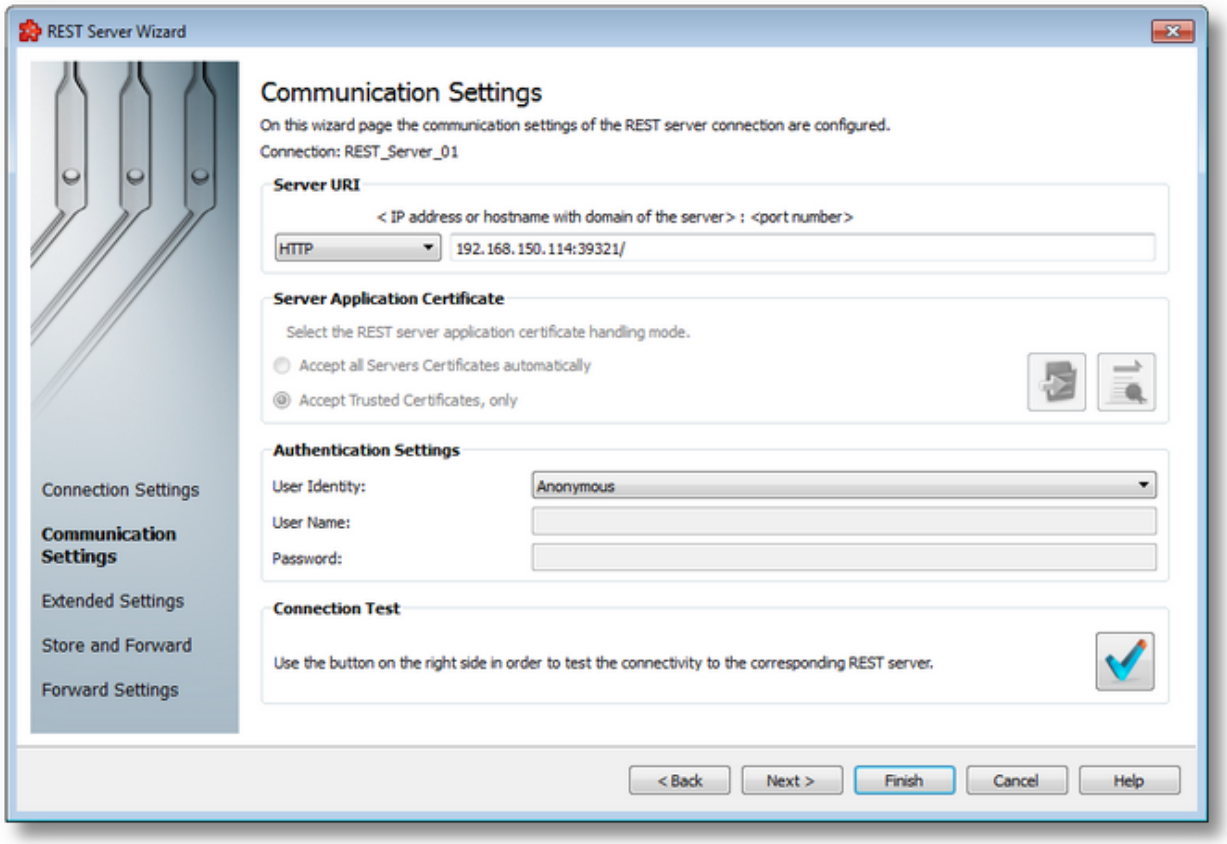
Example: if the checkbox is checked an array will be sent like this "[element1, element2, ..., elementN]", otherwise like this [element1, element2, ..., elementN].

These settings may be necessary for certain server which expect values as text or string.

The default values are unchecked, so no changes are made to the item values.

Connection Wizard - Communication Settings >>>

This page allows to specify communications settings such protocol type, host/IP, port number, certificate settings for HTTPS connections and authentication settings.



Server URI

In this section it is possible to specify the protocol type, choosing from TCP, HTTP and HTTPS. IPv4 and IPv6 are both supported. The port number is mandatory, and the user will be prompted if it is not provided.

The default port numbers per protocol type are the following:

- TCP: 8080

© Softing Industrial Automation GmbH

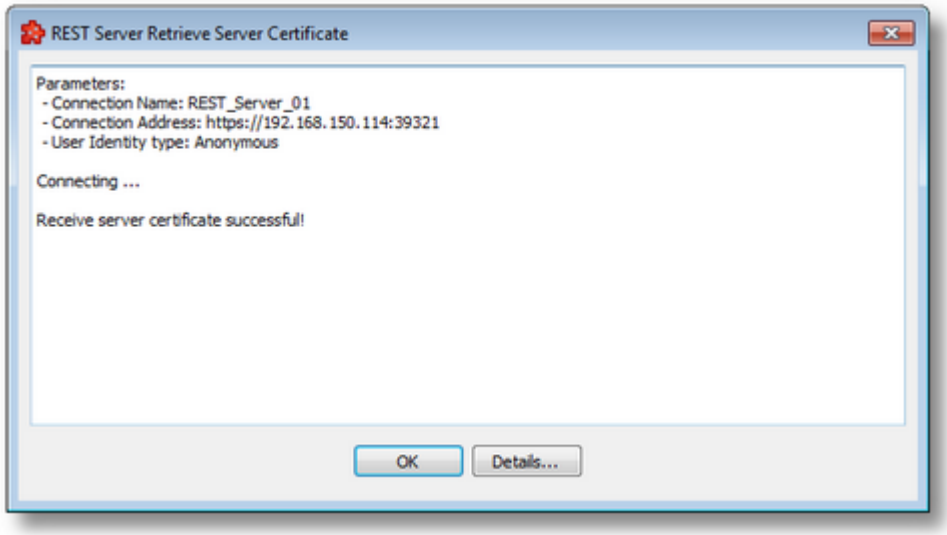
- HTTP: 80
- HTTPS: 443

REST Server Application Certificate

This section is only enabled if the protocol type for the connection is set to HTTPS. In this case there are two additional settings available.

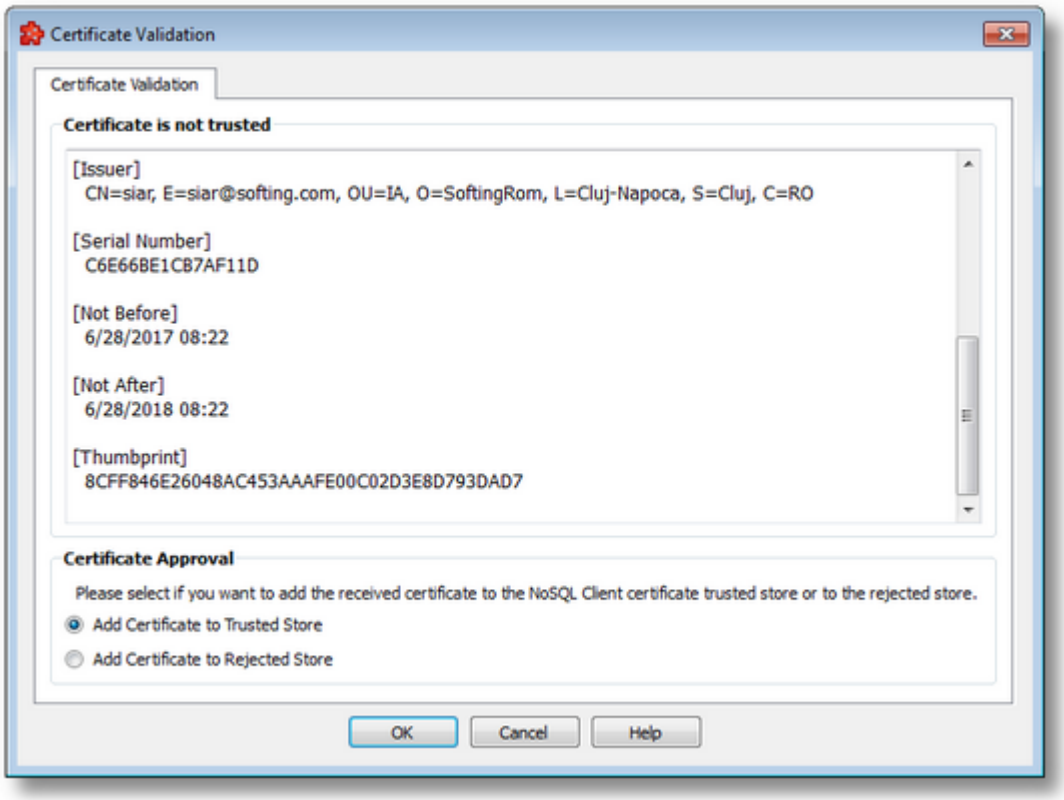
- *Accept all Server Certificates automatically* - if this option is selected, dataFEED will accept any certificate the server sends
- *Accept Trusted Certificates, only* - in this case, dataFEED needs to have the server certificate in its local certificate store, and will validate the server certificate against the one stored there at a certain interval, when executing actions related to this connection. If the stored certificate is in the trusted store and the certificate sent by the server match, then the actions are executed. If the server certificate is in the rejected store, then no actions of this connection will be executed.

To get the server certificate and save it in the local certificate store the user needs to use the *Receive server certificate* button next to the described settings above.



By pressing the *Details...* button the user can view extra information sent by the server, if available.

By pressing *OK,* the certificate content is presented and the user can choose where to store it in the local certificate store.



To view and manage all the stored certificates use the *Manage Certificates* button, which opens the Certificate Manager.

Authentication Settings

In this section the user can specify the two different authentication modes for TCP and HTTP, and three for HTTPS.

TCP & HTTP:

- Anonymous - no information is given with this option
- User name and password

HTTPS:

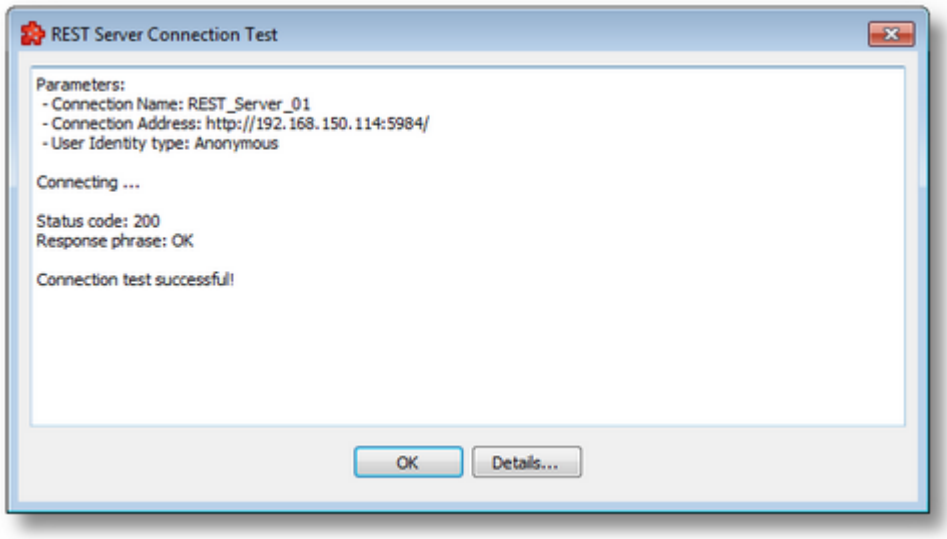
- All of the above
- Certificates - in this case a separate certificate and private key file can be provided as well as a private key password. The certificate file is mandatory, the other two are not. In some cases the certificate file may contain the private key as well.

Note: Because of the version of Qt used (5.2.1), dataFEED currently only supports certificates in .pem and .der formats. No other certificate file formats can be used and conversions need to be made by the user manually, for example using the openSSL tool, downloaded separately.

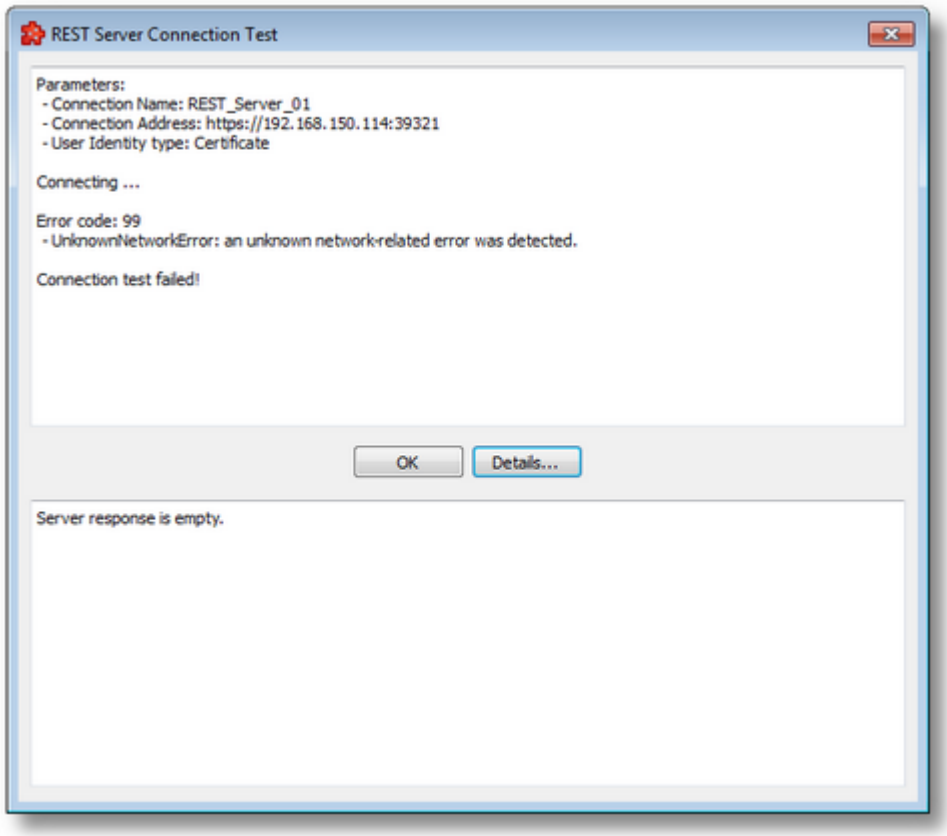
© Softing Industrial Automation GmbH

Test Connection

Finally, the *Connection Test* button can be used to check if the configured information lead to a successful connection with the server. If the test is successful actions will be executed successfully (if configured correctly themselves), otherwise the user needs to tweak these settings or check the server availability and configuration.



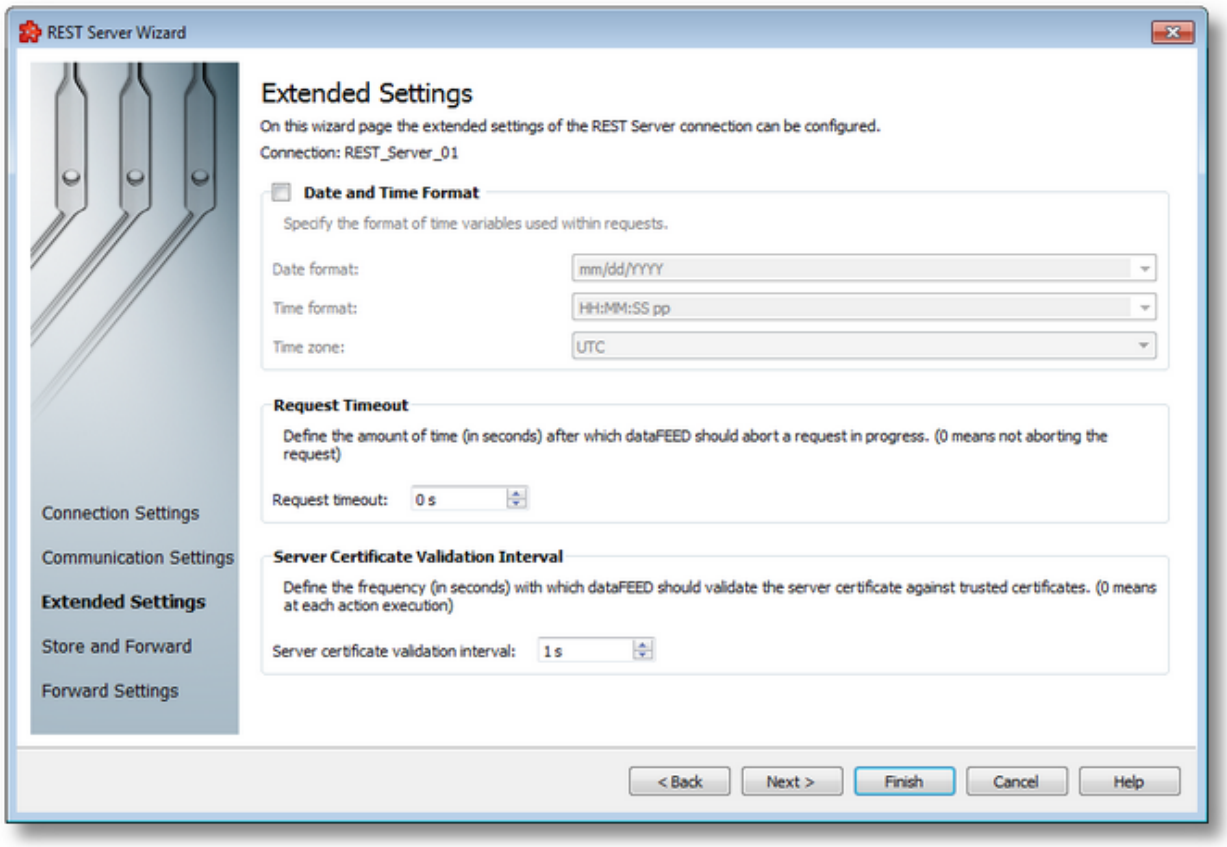
A successful connection test looks like above.



In any case, after pressing the button, a dialog will be displayed either prompting the user that the connection is successful, or showing error information, like in the image above.

<<< Connection Wizard - Connection Settings Connection Wizard - Extended Settings >>>

This page allows to specify the extended connection settings such as time and date format and request timeout.



Date and Time Format

This section handles the time and date formats that are to be used when the timestamp placeholders are expanded and their values written to the HTTP requests. The default values are `mm/dd/YYYY` for the *Date format and* `HH:MM:SS pp` for the *Time format*. Other defined values for Time and Date format are:

- *Date format*: `mm/dd/YYYY, dd-mm-YYYY, dd.mm.YYYY, dd/mm/YYYY, YYYY-mm-dd, mm-dd-YYYY`
- *Time format*: `HH:MM:SS pp, HH:MM:SS.FFF pp, HH:MM:SS.FFF, HH:MM:SS`

© Softing Industrial Automation GmbH

Values for Time and Date format can also be defined manually but in this case special care must be taken to ensure that the definitions are recognized by the local dataFEED OPC Suite application.

In addition to the default values listed in the format selection combo boxes, the user can add his/her own custom format. Custom Date and Time format strings are accepted as valid, when the format strings contain in their composition the following accepted format substrings and format delimiters. These formats allow the inclusion of specific locale information for the regional settings.

- *Valid Date format strings*: `AAAAAA, aaa, BBBBBBBB, bbb, ccccccccccccccccccc, dd, jjj, mm, UU, ww, xxxxxxxxxx, yy, YYYY`
- *Valid Time format strings*: `II, HH, MM, SS, pp, XXXXXXXX, FFFFFF, FFFFF, FFFF, FFF ,FF, F`
- *Valid Date and Time format delimiters*: `<dot>, <colon>, <slash>, <comma>, <backslash>, <vertical bar>, <semicolon>, <space>, <tab>, <minus>`

Time Zone handles the source localization time: UTC (Universal Time Coordinated) or local time (local time localization). The default time zone is UTC.

When the *Time, Date Format and Time Zone* option is not selected the default values are used. If they are modified, they apply to all actions of the currently selected REST connection.

Request Timeout

This value, expressed in seconds, is used when a request takes a long time to complete. If the user specifies a non-zero value, it means that dataFEED will wait for that amount of time before aborting the request. If the value is zero, then no aborting will be done, dataFEED will wait for the request to complete successfully or with some kind of error.

Default is 0.

Server Certificate Validation Interval

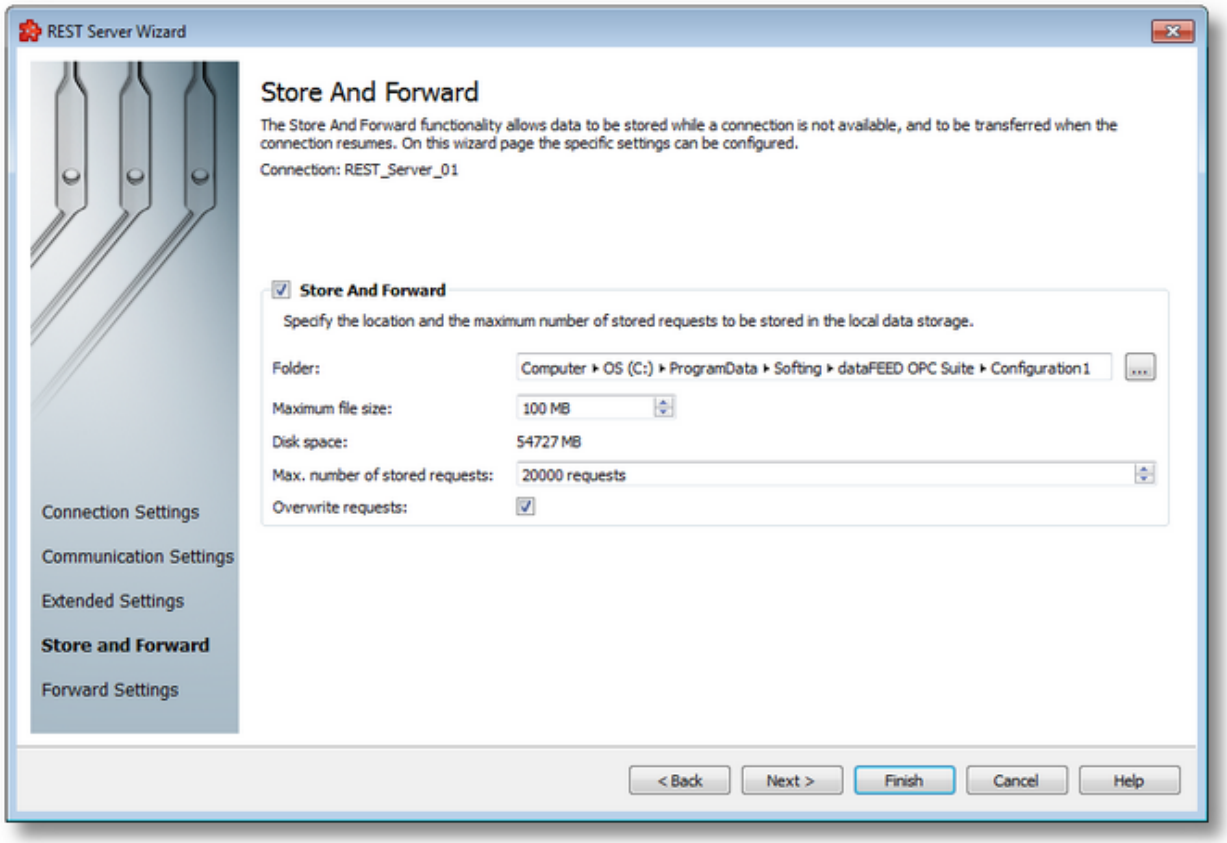
This value, expressed in seconds, is available only when the connection type is HTTPS and *Accept Trusted Certificates, only* is selected on the previous wizard page.

The value specifies after how much time, from the first action execution, will dataFEED validate the server certificate. The default value is 60 seconds, and 0 seconds means that it will validate the certificate sent by the server at each action execution.

Setting this value close to 0 is not advisable as it may add a performance penalty to action execution, if the server is slow in providing certificates or otherwise. Setting a higher value is better, as certificates do not change very often.

<<< Connection Wizard - Communication Settings Connection Wizard - Store and Forward Settings >>>

This page allows to specify the local storage settings in case that the server is unavailable.



This functionality can be turned on and off by using the *Store And Forward* checkbox. If it is off (default), it instructs dataFEED to drop any requests that are executed after the internal request queue is filled. If it is on, the application will store the requests lining up after the queue fills up.

The options that can be configured on this page and which change the mechanism's behavior are the following:

Folder

The path where the storage file should be located. If the path does not point to an existing folder then the corresponding path will be created by the dataFEED OPC Suite Configurator when the Next button is pressed, and user confirms.

The file is created at the run time of the local dataFEED OPC Suite application. If the file exists it will be overwritten. The name of the connection will be the name of the file as well, with .bin extension.

Maximum file size

© Softing Industrial Automation GmbH

Maximum file size is used to limit the space used on the hard drive. Minimum is 1 MB and maximum is 500 MB.

When the file capacity is reached requests will be dropped. This behavior is influenced by the *Overwrite requests* option described below.

Maximum number of stored requests

Maximum number of requests stored used to limit the requests stored in the file. Default is 20000.

Note: Requests are dropped starting either when the file size limit or the requests limit is reached first.

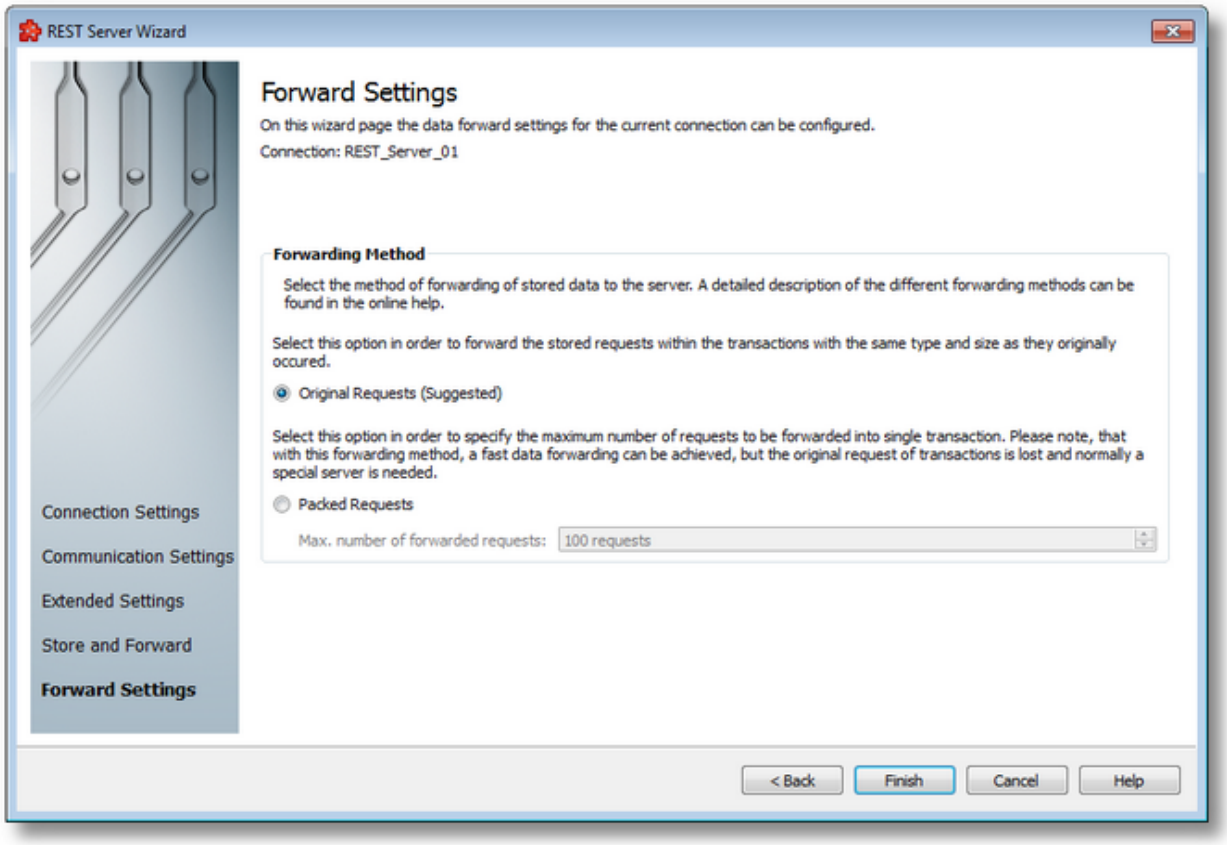
Overwrite requests

Whether to overwrite previously stored requests when the above specified number has been reached.

If the checkbox is not checked then no more data changes will be stored when the maximum number of requests has been reached (this allows access to the first stored requests). If the checkbox is checked the oldest request will be overwritten with the newest one when the maximum number of requests has been reached (this allows access to the latest stored requests).

<<< Connection Wizard - Extended Settings Connection Wizard - Forward Settings >>>

This page allows to specify the local storage settings in case that the server is available after a while.



Forwarding Method

This page is only available if the *Store And Forward* functionality is enabled in the previous wizard page.

In the Forwarding Method group you can choose between the following options:

- If "Original Requests" is selected, the stored data changes will be forwarded in requests of the same type and size, in which they originally occurred. This is the recommended option.
- If "Packed Requests" is selected, the maximum number of forward requests also needs to be specified. This is an advanced feature. This option can be selected in order to specify a certain bigger number of data changes to be forwarded in a single requests. Note that with this forwarding method, a fast data forwarding can be achieved, but the original requests are lost and normally a special handling on the server side would be needed.

<<< Connection Wizard - Store and Forward Settings

7.3.3.6.3 Action Wizard

▼ With the REST Action Wizard you can create a new REST Action definition or modify an existing one:

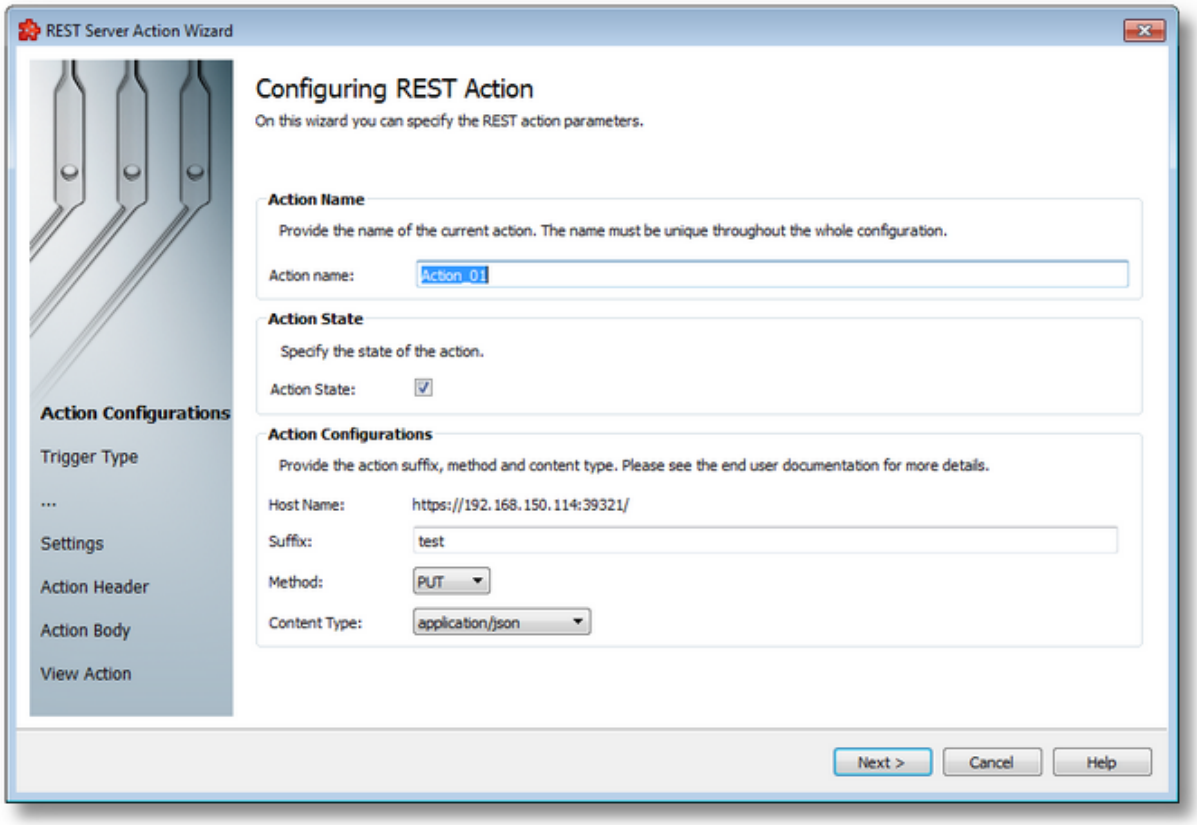
© Softing Industrial Automation GmbH

- Obtain general information about the REST Action Wizard @ REST Action Wizard - Configuring REST Action
- Choose the action trigger type @ REST Action Wizard - Trigger Type
- Pick the item trigger @ REST Action Wizard - Trigger Item
- Select the timer trigger @ REST Action Wizard - Timer Selection
- Define the timer identity @ REST Action Wizard - Timer Identity
- Choose the timer type @ REST Action Wizard - Timer Type
- Adjust schedule timer settings @ REST Action Wizard - Schedule Timer Settings
- Set the timer interval @ REST Action Wizard - Interval Timer Settings
- Configure the execution condition and the update rate @ REST Action Wizard - Action Settings
- Define the header for the current action @ REST Action Wizard - Action Header
- Define the body for the current action @ REST Action Wizard - Action Body Definition
- See the final text output for the current action @ REST Action Wizard - Action Overview

Note: A wizard flow will not include all the pages listed above. Depending on the selections performed in some of the pages, different flow paths will be followed.

Softing dataFEED OPC Suite V5.38
© 2024 Softing Industrial Automation GmbH

The first wizard page provides the first set of action properties to complete.



Action Name

A unique identifier for the current action. In case the provided name is already used by another action, the user is not allowed to proceed to the next wizard page, until they provide a different name.

Action State

This flag specifies whether or not the current connection will be executed.

Action Configurations

- Host Name - shows the URI from the connection definition to which this action is related
- Suffix - the user can specify additional URI data that will be used to manipulate specific data and resources on the server by this action: for example "resource_name/property_name"
- Method - the user can select from POST (for creation resources) and PUT (for updating resources) HTTP verbs
- Content Type - the content type of the action body is selected from this list of options: application/json for JSON format, application/xml, application/xhtml+xml and text/html for XML format, for application/x-www-form-urlencoded the body of the message sent to the server is essentially one query string -- name/value pairs are separated by the ampersand (&), and

© Softing Industrial Automation GmbH

names are separated from values by the equals symbol (=), and text/plain for plain text format without any validation in the application.

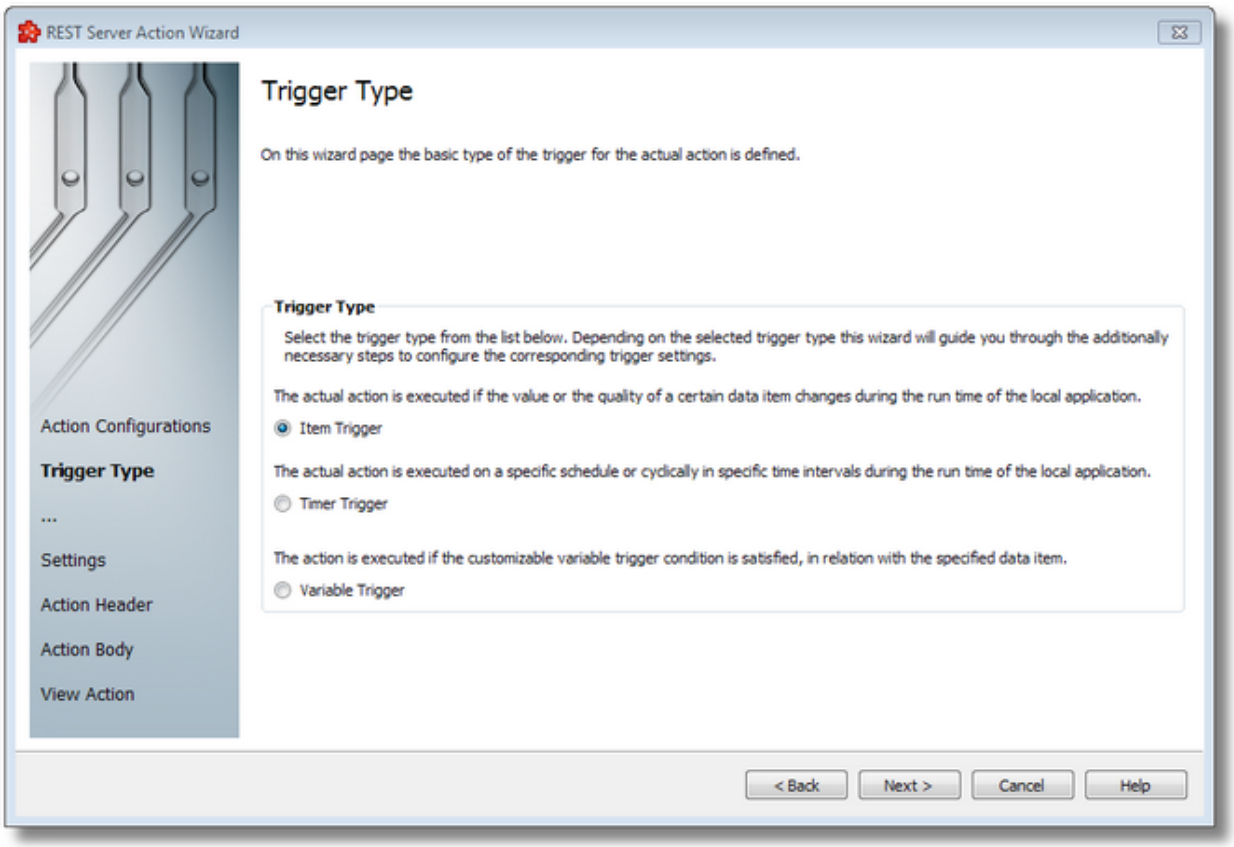
For JSON and XML formats the Configurator offers validation of the content, and reports errors to the user if for some reason the content is not in a valid form. For plain text cases, the user has more flexibility in creating the body content in the action, but no validation can be done.

Note: Conversions between these content types are also possible, if the user changes the content type and then goes through the wizard before saving.

REST Action Wizard - Trigger Type >>>

The Trigger Type wizard page provides the possibility to select the trigger type for the actual action. The following trigger types are available:

- *Item Trigger* - This trigger is fired when a data change for the selected item occurred. A data change means that the value or the quality of the item changed.
- *Timer Trigger* - This trigger is fired at a specified time or interval.
- *Variable Trigger* - This trigger is fired when a data change for the items in the variable conditions occurs. A data change means that the value or the quality of the item changed.



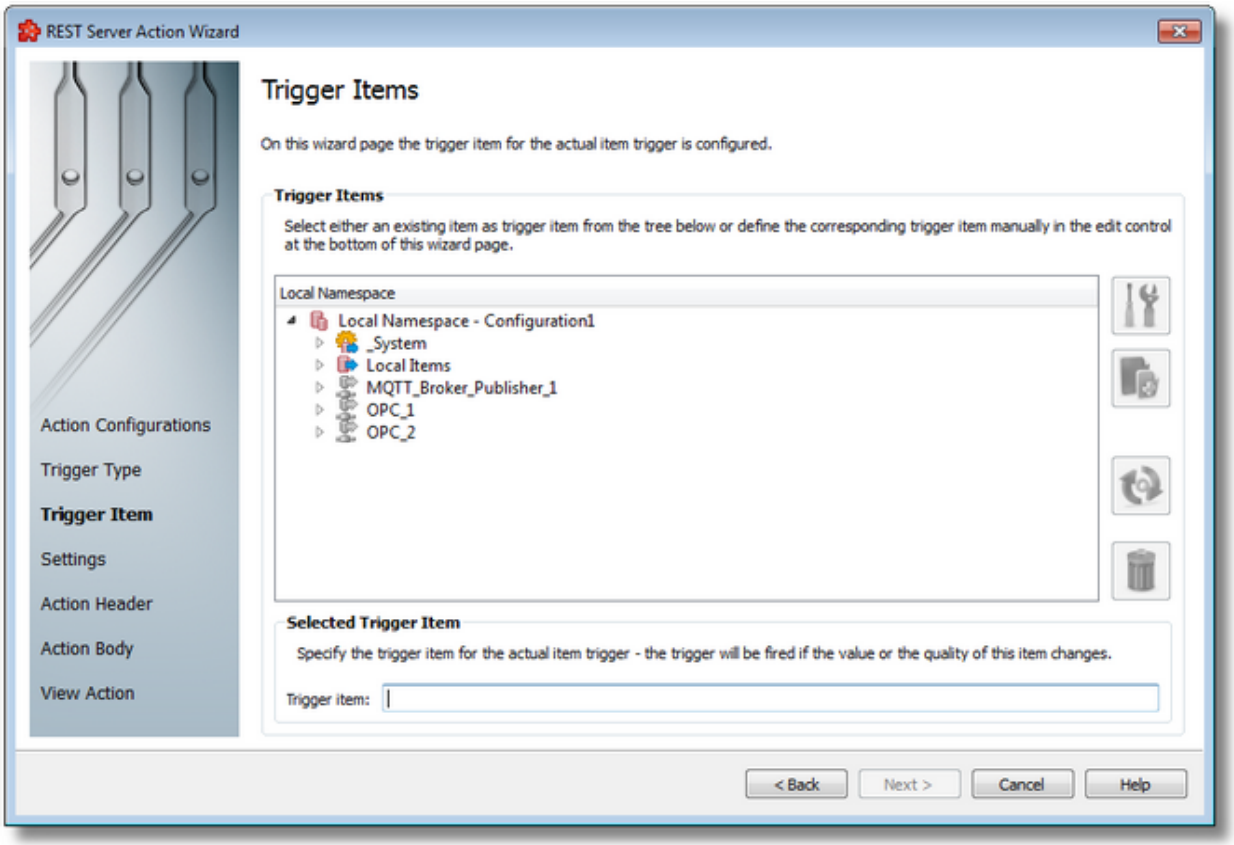
To proceed with the wizard and continue to define the selected type of trigger (or to advance to the last wizard page when the *Data Change* trigger type is selected) use the *Next* button, to go back to the previous wizard page use the *Back* button, or discard the wizard by pressing the *Cancel* button.

Depending on the selected *Trigger Type* the next wizard page will be:

- REST Action Wizard - Trigger Item - if the selected trigger type is *Item Trigger*,
- REST Action Wizard - Timer Selection - if the selected trigger type is *Timer Trigger*
- REST Action Wizard - Variable Trigger - if the selected trigger type is *Variable Trigger*

<<< REST Action Wizard - Configuring REST Action REST Action Wizard - Trigger Item >>>

The *Trigger Item* wizard page is shown if the selected trigger type is *Item Trigger*. Here you can configure the actual trigger item.



This wizard page contains two separate sections:

- *Trigger Items* section, which allows selecting an item trigger or managing the Local Namespace tree.

© Softing Industrial Automation GmbH

- *Selected Trigger Item* section, which shows the item ID of the selected trigger item and also allows typing an item ID, thereby selecting this item in the above list.

Trigger Items section

The entire namespace of the current configuration is shown as a namespace tree. The name of the root node is composed by appending the instance name to the *Local Namespace* keyword. The first level of the namespace contains:

- A node for each server connection, named by the server connection items prefix.
- A node for System Items.
- A node for Local Items.

The local namespace tree offers the possibility to explore the entire namespace of the local dataFEED OPC Suite application in order to select a trigger item. After the selection is made, the *Trigger item* edit box is populated with the selected item ID.

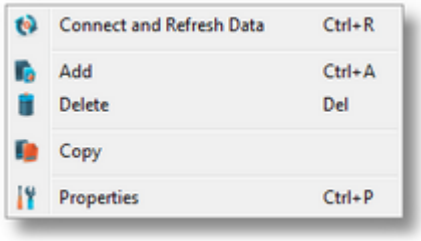
The entire namespace of the local dataFEED OPC Suite application can be managed through this wizard page by performing certain operations using the buttons on the right, the context sensitive menu or the keyboard shortcuts:

- Open the item properties dialog using the *Properties of the selected item* button  , pressing the keyboard shortcut CTRL+P or selecting the *Properties* action from the context menu. Depending on the type of the selected item, a special Item Properties dialog opens:
    - If the selected item is under Local Items the Modify Local Item dialog is opened.
    - If the selected item is under a dynamic server connection the Modify Dynamic Server Item dialog is opened.
    - If the selected item is under a static server connection the Modify Static Server Item dialog is opened.
- Add a new item using the *Add a new item*  button, pressing the keyboard shortcut CTRL+A or selecting the *Add* action from the context menu. Depending on the type of the selected node, a special Add Item dialog opens:
    - If the selected node is under Local Items the Add Local Item dialog is opened.
    - If the selected node is under a dynamic server connection the Add Dynamic Server Item dialog is opened.
    - If the selected node is under a static server connection the Add Static Server Item dialog is opened.
- Copy the selected item to clipboard by selecting the *Copy* action from the context menu.
- Paste the previously copied item from clipboard by selecting the *Paste* action from the context menu. Items originating from the Local Items cannot be pasted under server connections, and vice-versa; an error message will be displayed in both cases. If an item is pasted in a static (filtered) server connection and the item does not exist on the server an error message is displayed.

- Connect to the server and refresh the data items by using the *Refresh data items* button, pressing the keyboard shortcut CTRL+R or selecting the *Connect and Refresh Data* action from the context menu. The namespace of the selected server connection is recreated; if manually added items not referenced in the actual configuration exist, those items are deleted from the namespace.

- Delete the selected item and all of its child items by using the Delete data items button , pressing the DELETE / DEL key or selecting the Delete action from the context menu. Items which are included in any Data Processing / Data Destination add-ons will not be deleted but an error message will be displayed instead.

An example of the context menu can be seen below:



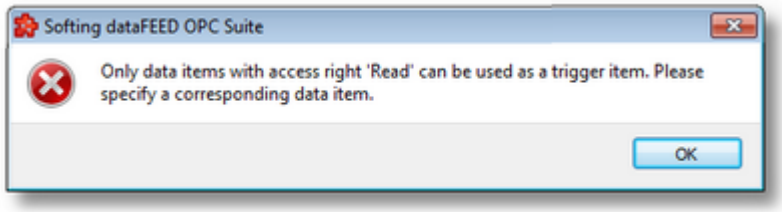
Selected Trigger Item section

The edit box in this section is populated with the item ID after an item from the Local Namespace tree is selected.

You can also manually type the ID of the item in the *Trigger item* edit box and thus have the above tree automatically select a valid server item.

Note: If the item ID is typed and the specified item exists in the Local Namespace, it will be used as the trigger item. The item will only be created if it could not be localized in the Local Namespace. In that case the newly created item will be used as the trigger item.

Note: If the item corresponding to the typed item ID could not be created, an error message will be displayed requesting the input of a new, valid item.

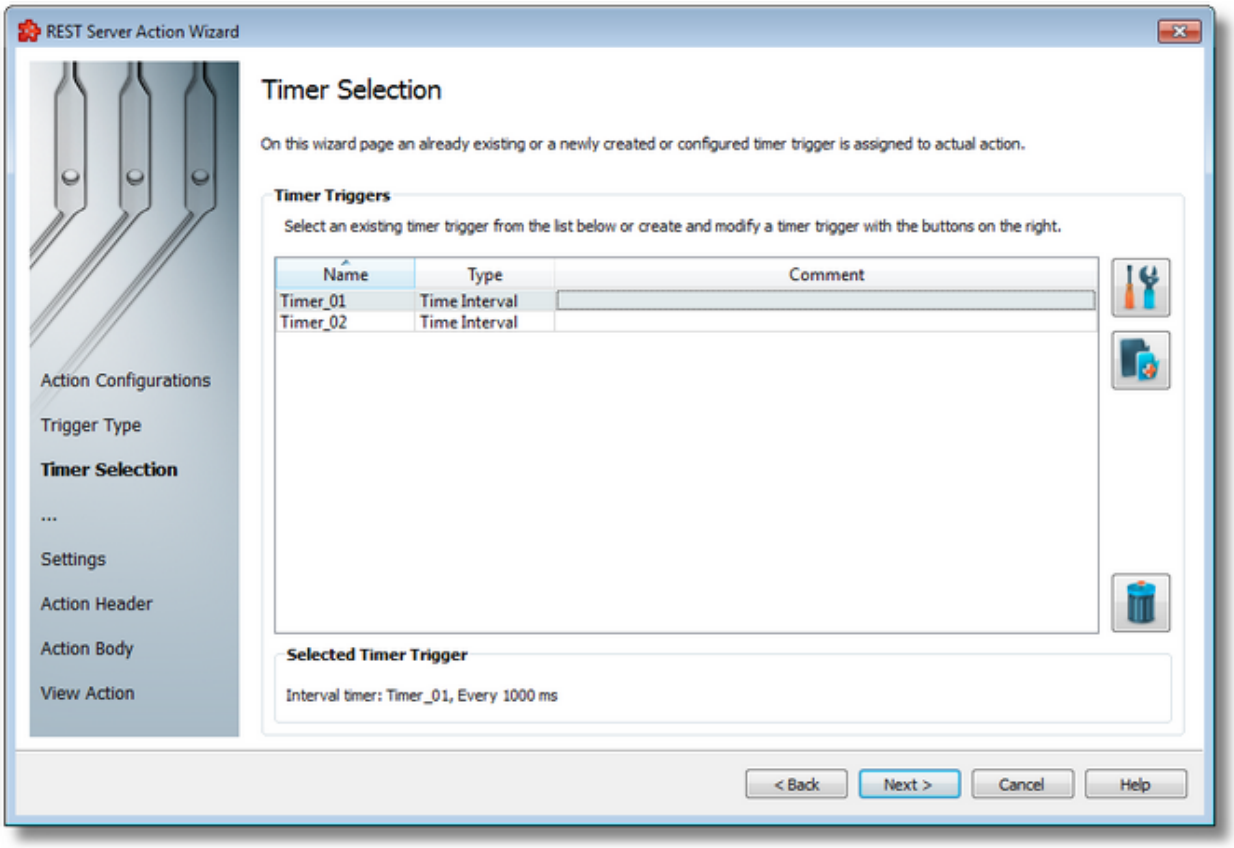


Attention: If the item corresponding to the typed item ID is valid and resides under a static server connection, the respective item will be created in the filter file definition.

© Softing Industrial Automation GmbH

<<< REST Action Wizard - Trigger Type REST Action Wizard - Action Settings >>>

The Timer Selection wizard page is shown only if the selected trigger type is *Timer Trigger*. Here you can configure the actual timer trigger.

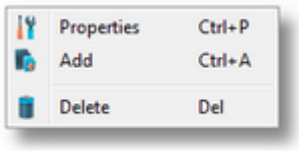


The current page allows selecting a previously defined timer trigger and managing the timer list by using the buttons on the right, the context sensitive menu or the keyboard shortcuts:

- Change the properties of the selected timer by using the *Properties* button  , pressing the keyboard shortcut CTRL+P or selecting the *Properties* action from the context menu.
- Add a new timer by using the Add timer trigger button  , pressing the keyboard shortcut CTRL+A or selecting the Add action from the context menu.
- Delete the selected timer trigger by using the Delete selected timer trigger button  , pressing the DELETE / DEL key or selecting the Delete action from the context menu.

Note: If a timer trigger has associated actions it cannot be deleted.

An example of the context sensitive menu can be seen below:



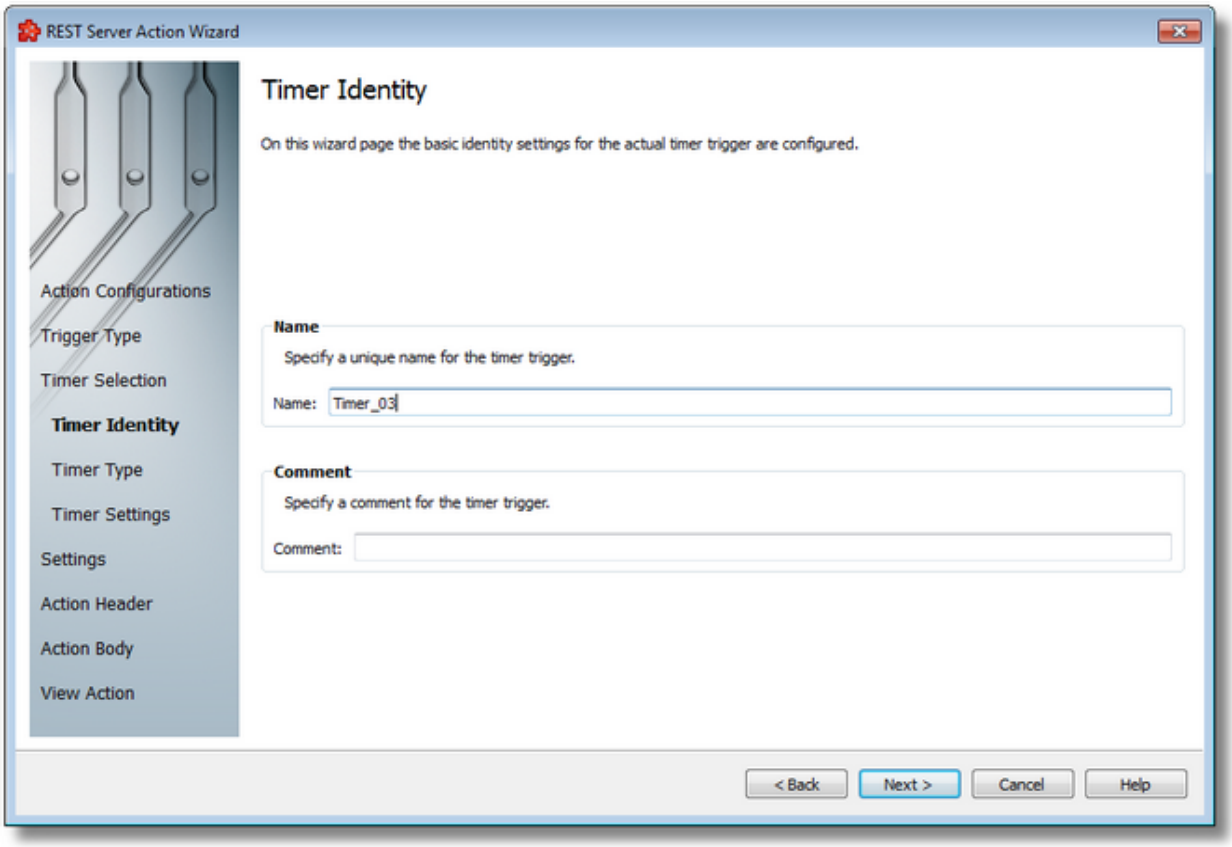
Note: Information about the selected timer is shown in the *Selected Timer Trigger* area at the bottom of the page.

A sub-wizard that allows adding or modifying timer triggers can be launched by selecting to change an existing timer trigger or to add a new one. The first page of this sub-wizard is REST Action Wizard - Timer Identity.

The wizard will continue straight to the *Action Settings* page as an already defined timer is selected.

<<< REST Action Wizard - Trigger Type REST Action Wizard - Action Settings >>>

The Timer Identity wizard page provides the possibility to assign a name and a comment to the timer trigger.



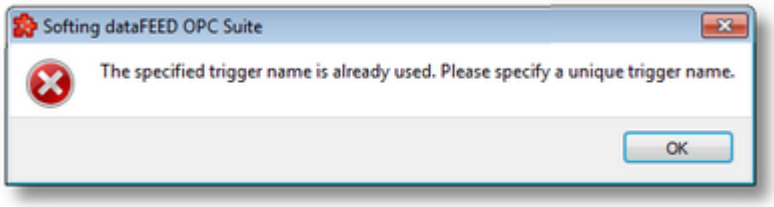
Note: When adding a new timer trigger, the timer name will be automatically generated according to the format: *Timer_number*, where the *number* part is an auto-incremented integer.

© Softing Industrial Automation GmbH

When editing an existing timer, the timer's defined name and comment will be filled in the two edit boxes allowing you to modify these attributes.

Attention: The name of the *Timer Trigger* must be unique among all triggers and cannot be empty.

If the newly added timer trigger has the same name as an existing trigger, an error message box will be displayed and a new trigger name must be provided:

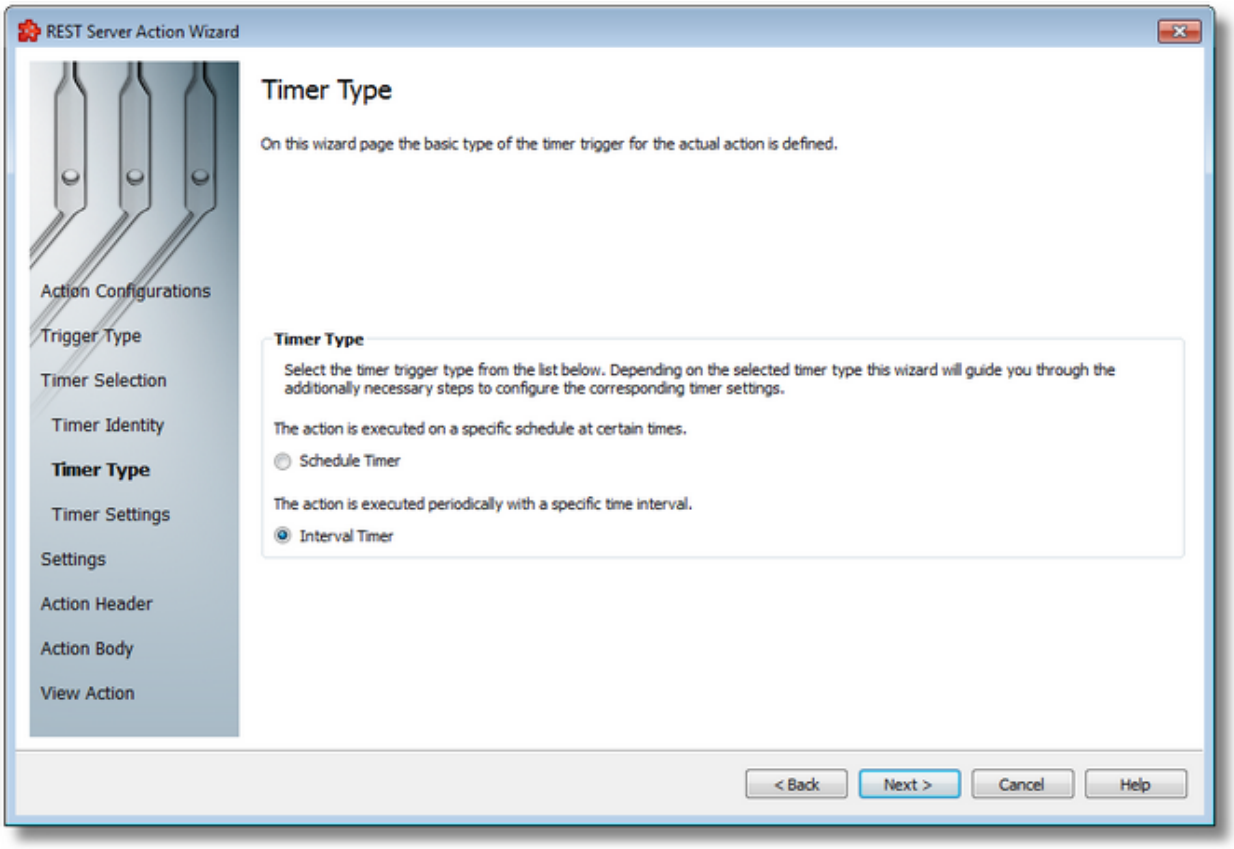


The *Comment* field of the *Timer Identity* wizard page is optional; a Timer Trigger can also be added without any comment.

<<< REST Action Wizard - Timer Selection REST Action Wizard - Timer Type >>>

This wizard page allows choosing the type of the timer trigger. The following types of timer are available:

- *Interval Timer*
- *Schedule Timer*



Interval Timer

The Interval Timer type allows defining a specific interval of time (in milliseconds) between the timer fire events. The time interval can be configured on the REST Action Wizard - Interval Timer Settings wizard page.

Schedule Timer

The Schedule Timer type allows defining a specific date and time when the timer should fire. The schedule can be configured on the REST Action Wizard - Schedule Timer Settings wizard page.

Note: The *Timer Type* of an existing *Timer Trigger* cannot be changed.

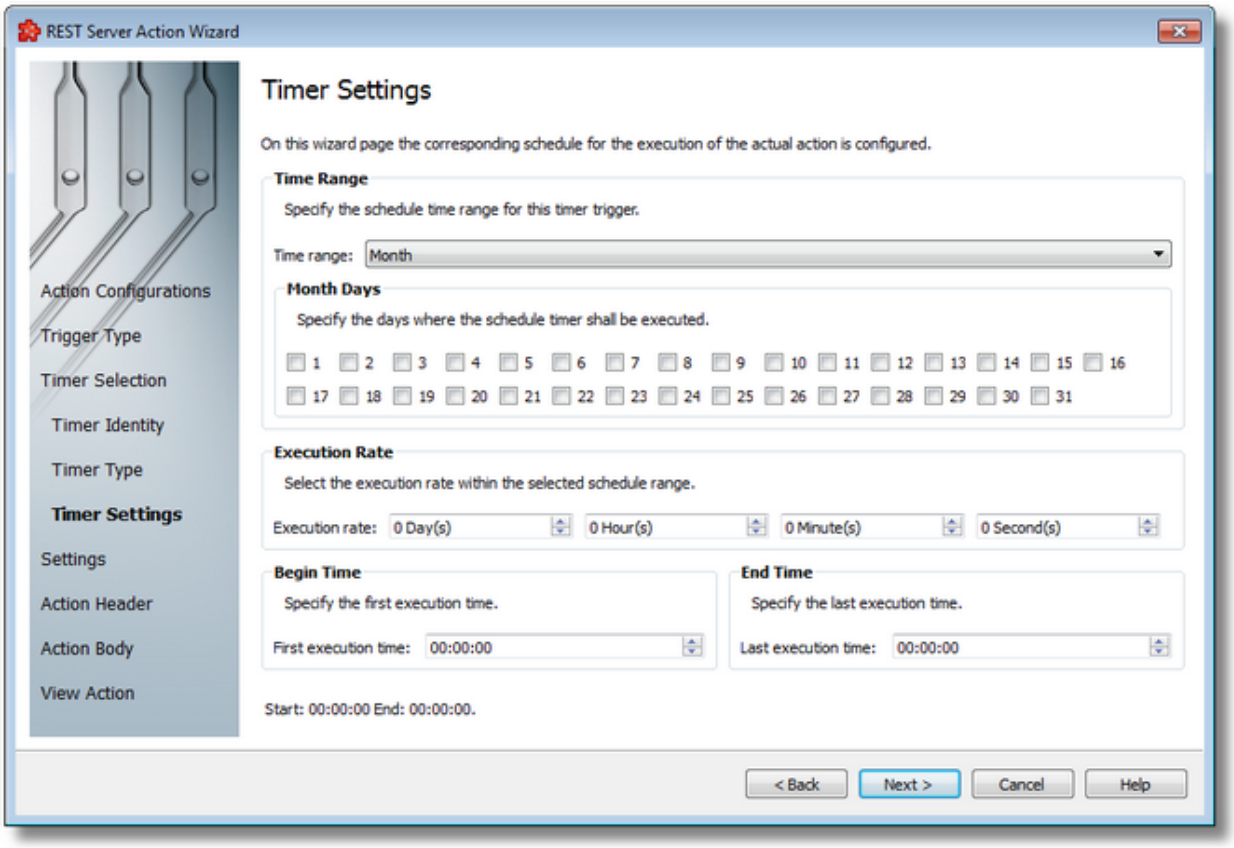
To continue defining the specific timer properties press the *Next* button, to return to the *Timer Identity* wizard page use the *Back* button, or exit this sub-wizard (thereby aborting the *Action Wizard*) by pressing the *Cancel* button.

The description of the Action Wizard will continue with the wizard pages for configuring a *Schedule Timer*.

<<< REST Action Wizard - Timer Identity REST Action Wizard - Schedule Timer Settings >>>

© Softing Industrial Automation GmbH

The *Schedule Timer Settings* wizard page is a more complex page compared to the *Interval Timer Settings* wizard page and allows defining the corresponding schedule for the execution of the trigger action:



The page has four sections which identify the schedule:

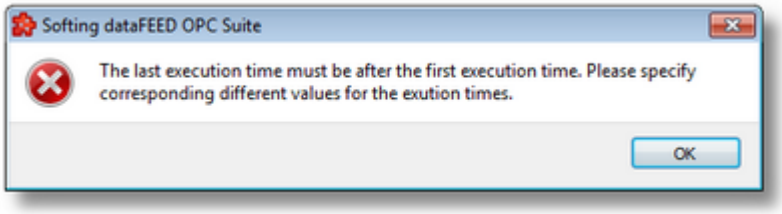
- *Time Range*: Provides the possibility to choose one of the following schedule ranges: *month*, *week*, *day*, *hour* or *minute*. When the *month* or the *week* ranges are selected the panel below the selection is populated with the month and week days, respectively.
- *Execution Rate*: Sets the execution rate of the current action in the previously-defined *Time Range* (e.g. every X day(s), every Y hour(s), every Z minute(s), every T second(s)).
- *Begin Time*: Sets the time the current action should begin execution.
- *End Time*: Sets the time the current action should stop execution.

According to the *Time Range* the *Begin Time* and the *End Time* fields will allow typing:

- Seconds (SS) if the *Time range* is *Minute*.
- Minutes and seconds (MM:SS) if the *Time range* is *Hour*.
- Hours, minutes and seconds (HH:MM:SS) if the *Time range* is *Day* or greater (*Week*, *Month*).

Note: A short review of the configured timer schedule is shown at the bottom of this wizard page.

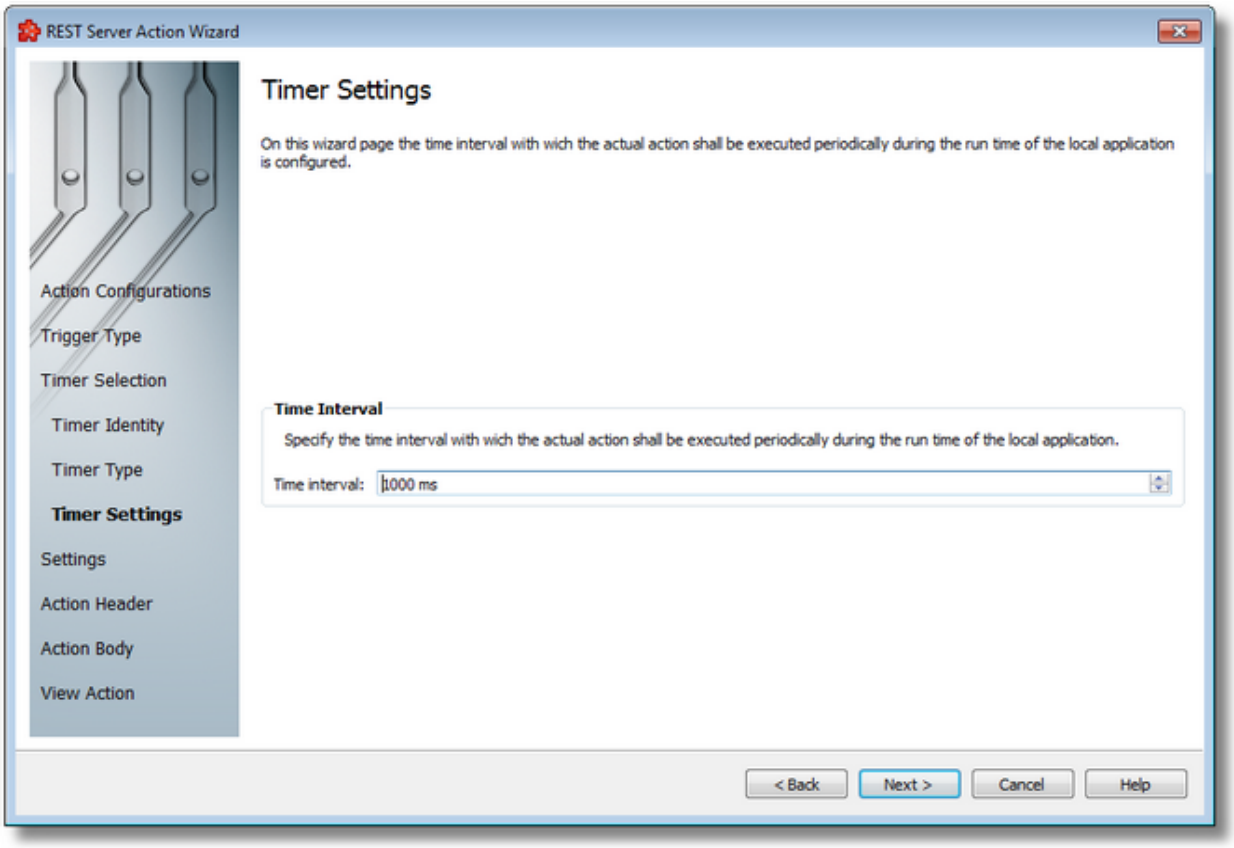
If the defined schedule is not valid, an error message dialog will appear naming the error that occurred.



Continue to the *Timer Selection* wizard page using the *Next* button, return to the previous wizard page using the *Back* button or exit the wizard pressing the *Cancel* button.

<<< REST Action Wizard - Timer Type REST Action Wizard - Timer Selection >>>

The Interval Timer Settings wizard page allows configuring the time interval at which the current action should be (periodically) executed.



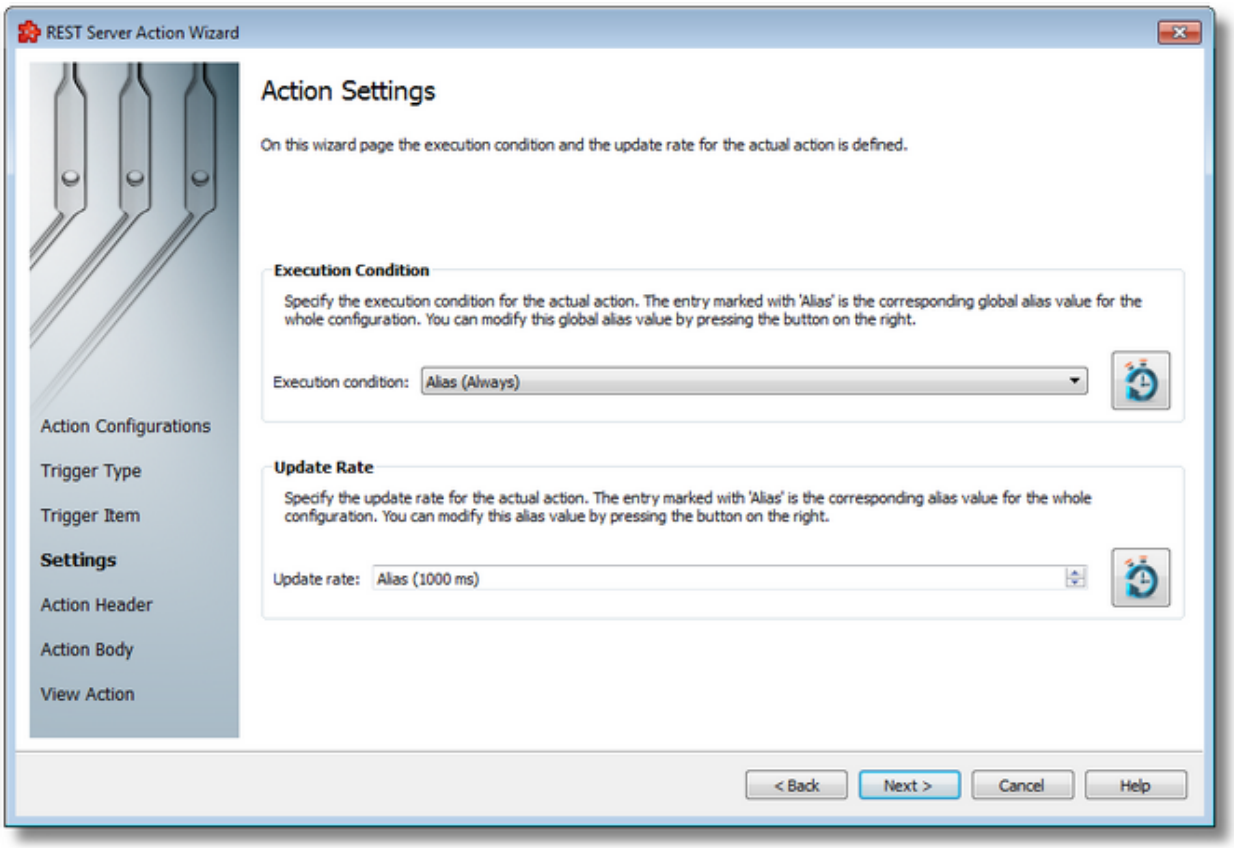
© Softing Industrial Automation GmbH

The time interval value is measured in milliseconds and ensures that the action will be triggered at the time interval defined in the spin-box.

Continue to the *Timer Selection* wizard page by pressing the *Next* button, return to the previous one by using the *Back* button, or exit the wizard by pressing the *Cancel* button.

<<< REST Action Wizard - Timer Type REST Action Wizard - Timer Selection >>>

The Action Settings wizard page allows defining the execution condition and the subscription update rate for the current action.



The wizard page is divided into two sections: *Execution Condition* and *Update Rate*.

Execution Condition

Each statement has an execution condition which is evaluated by the local dataFEED OPC Suite application. The action is only performed when the evaluation is successful. If the evaluation fails then the statement is ignored.

The execution conditions allow reacting to server communication errors, executing special statements or preventing the execution of other statements.

The execution condition can be selected directly from the drop down combo box. The initial value is the global alias one. Changing the global default execution condition can be done using the Configure global alias and initial values button  or pressing the keyboard shortcut CTRL+D.

Update Rate

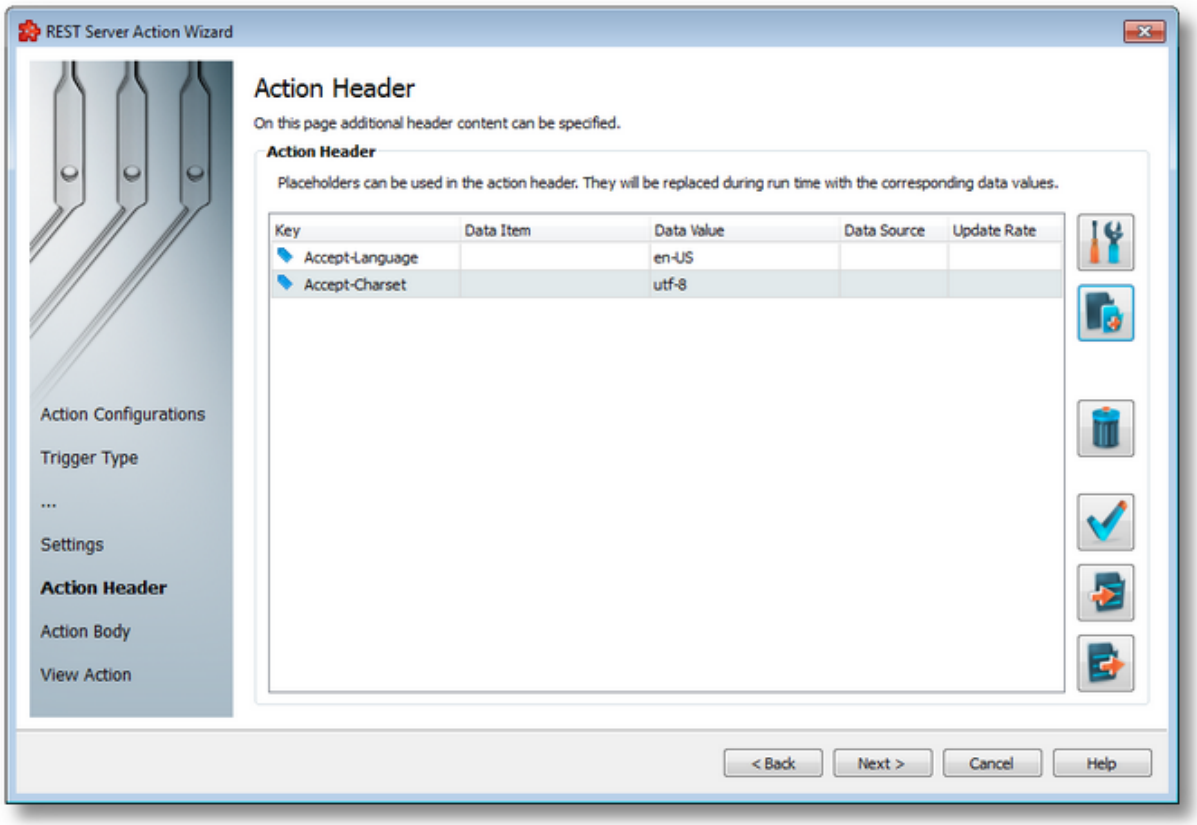
The *Update Rate* specifies the execution rate of the current action.

The update rate can be typed or selected from the spin box. The initial value is the global alias one. Changing the global default update rate can be done using the Configure global alias and initial values button  or pressing the keyboard shortcut CTRL+D.

Complete the *Action Wizard* pressing the *Finish* button, review the past wizard pages using the *Back* button, or cancel the wizard thus discarding all the changes by pressing the *Cancel* button.

<<< REST Action Wizard - Trigger Type REST Action Wizard - Action Header >>>

On this page additional header information can be provided for the current action.



© Softing Industrial Automation GmbH

By using the *Add*  button, the user will be prompted with the *Parameter Properties* dialog in which data coming from items as well as manually entered data can be added to the header section of the HTTP request.

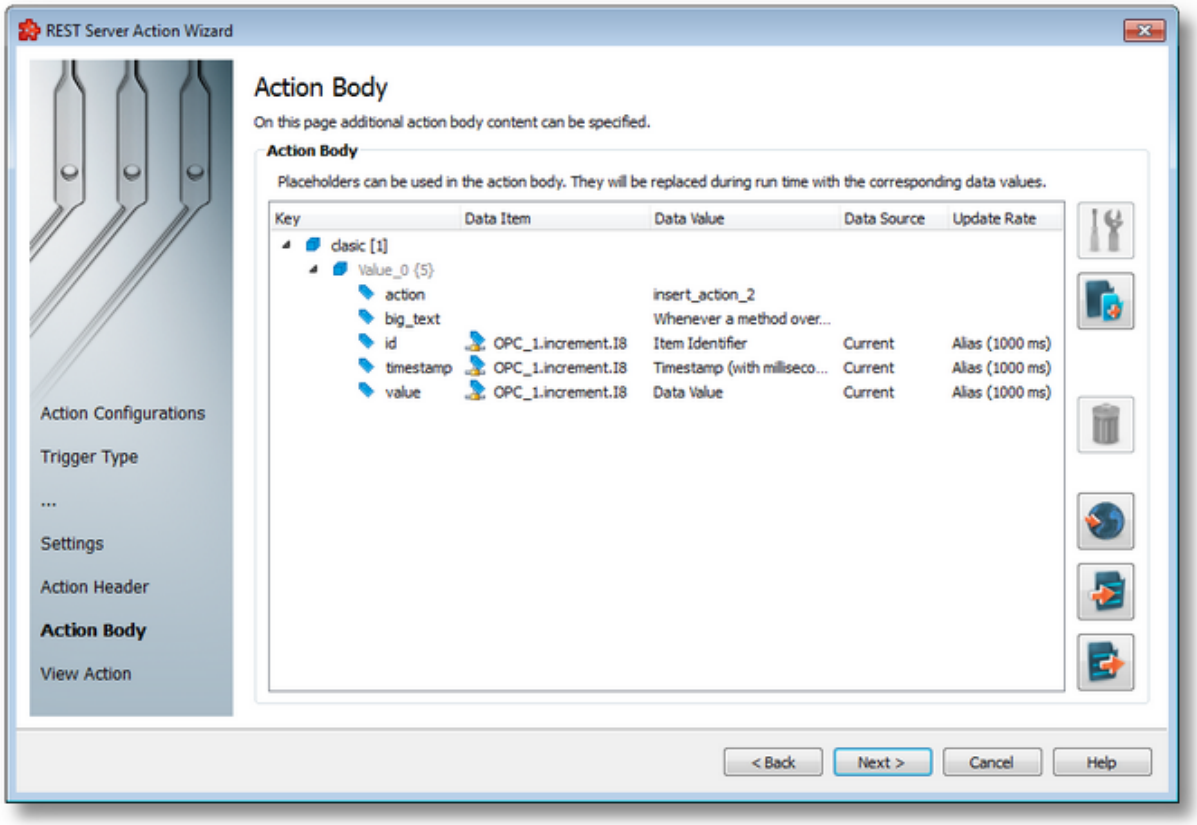
Note: See the Parameter Properties, Import and Export dialogs on the REST Action Wizard - Action Body Definition page.

By default the *Content-Length*, *Content-Type* and *Authorization* header attributes are set by the application, if their values are specified. These values are not visible on this wizard page, but they can be overridden by the user if they are specified again and new values are provided.

The user can view the final form of the header, as it will be sent by pressing the *View* button. It is possible to Import and Export these values by using the two buttons on the left side, bottom.

<<< REST Action Wizard - Action Settings REST Action Wizard - Action Body Definition >>>

On this page additional body information can be provided for the current action.

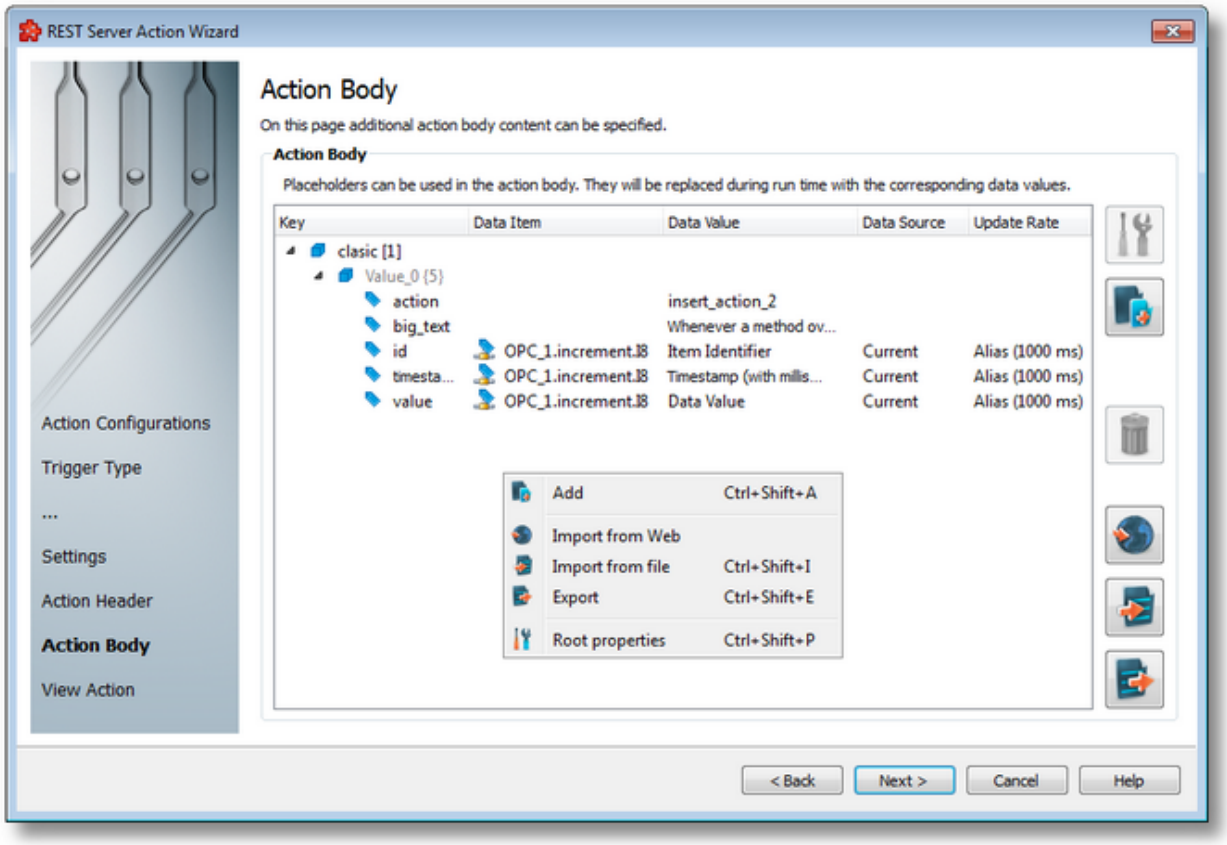


The most important part of the wizard flow is the action body editing page. This is where the actual content of the action request is specified in the selected content format, on the first

wizard page. The user needs to create the server specific payload structure in this wizard for the server to be able to accept it.

Similarly to the header page, the user has the possibility to add data placeholders coming from items from the address space, or statically defined data values, in different types.

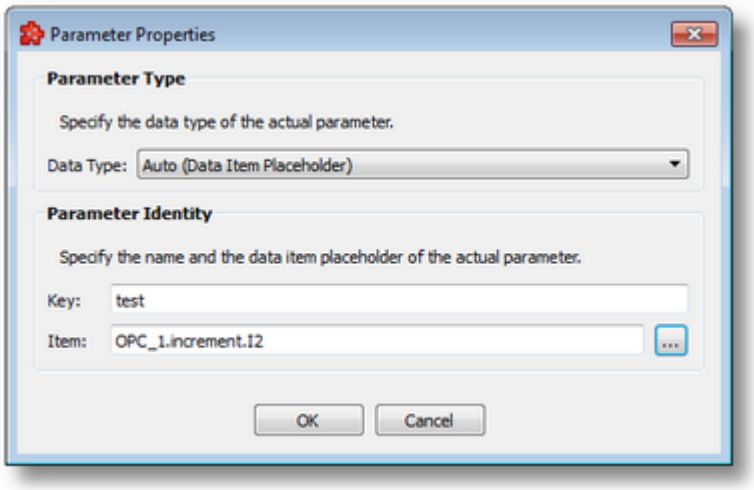
The user can right-click in the tree view above and access all the available options from a context-menu as well:



By using the *Properties*  button, any of the elements in the structure can be modified.

By using the *Add*  button, the user will be prompted with the *Parameter Properties* dialog in which this is configurable.

© Softing Industrial Automation GmbH



The image above shows the dialog set with data coming from an item. A key needs to be given in all cases, then the value in the box below the key. The user is able to specify other types as well, by selecting on of the types from the *Data Type* list.

For more information refer to the Parameter Properties page.

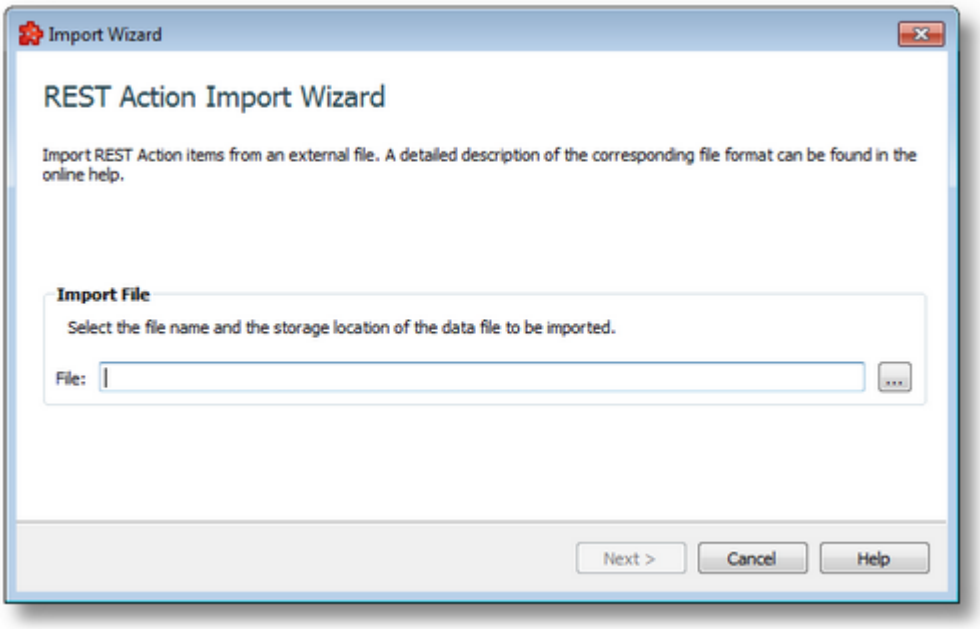
Note: Due to performance penalty reasons the maximum number of items that can be included in the header or body data of the action is limited to 10000. The maximum number of nodes in the structure of either of them is 100000 nodes. If these limits are reached or exceeded the contents will be truncated to fit. For the plain text format the maximum allowed character number in the body content is 100000 characters.

Note: Different update rates can be specified for each action item compared to that of the action, but they will only work if the user specifies the "Current" and "Cache" settings for the item. It will not work with the "Changed" setting.

Import and Export

The user has the possibility to import and export body content to be able to reuse it for other action definitions that only need small changes.

To do this, the *Import* and *Export* buttons can be used, which allow the user to specify a source file from which the content will be imported, and destination file to which it will be saved. Also, a *Import from Web* button is available to be able to get content from the Web as well.



Note: When using the *Import from Web* functionality with HTTPS URLs, no certificates or other types of authentication data can be provided.

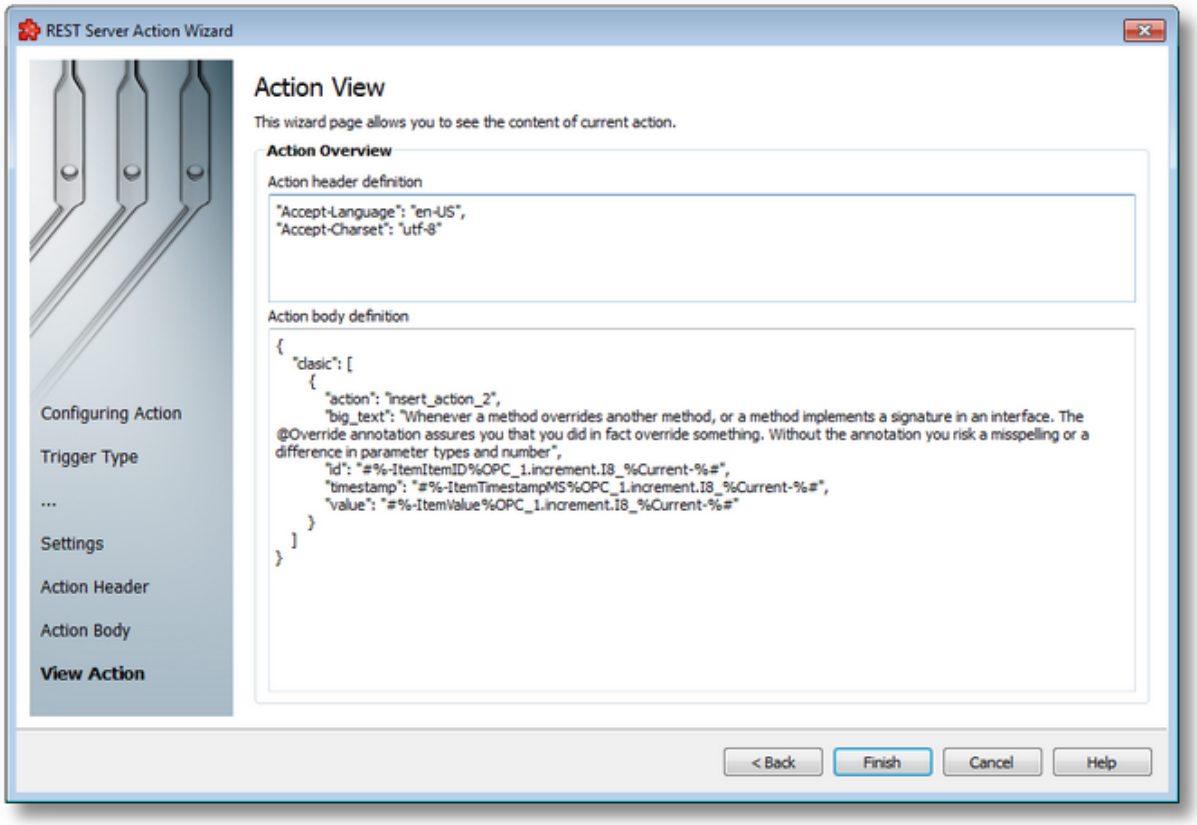
After a failed import or export operation the results page may contain the following information:

- The invalid line where the operation failed.
- The reason why the line is considered invalid.

<<< REST Action Wizard - Action Header REST Action Wizard - Action Overview >>>

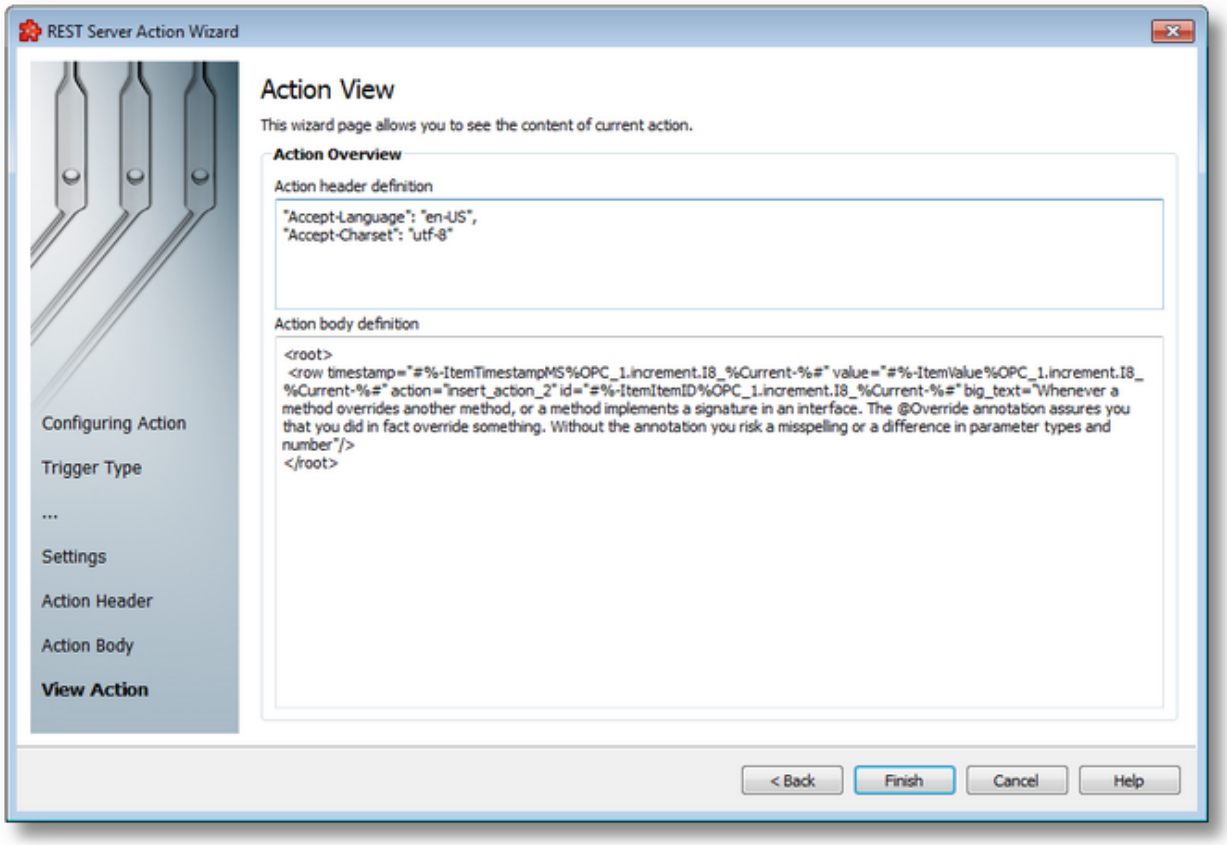
On this wizard page the user is presented with the final form of the content of the action, both header and body values.

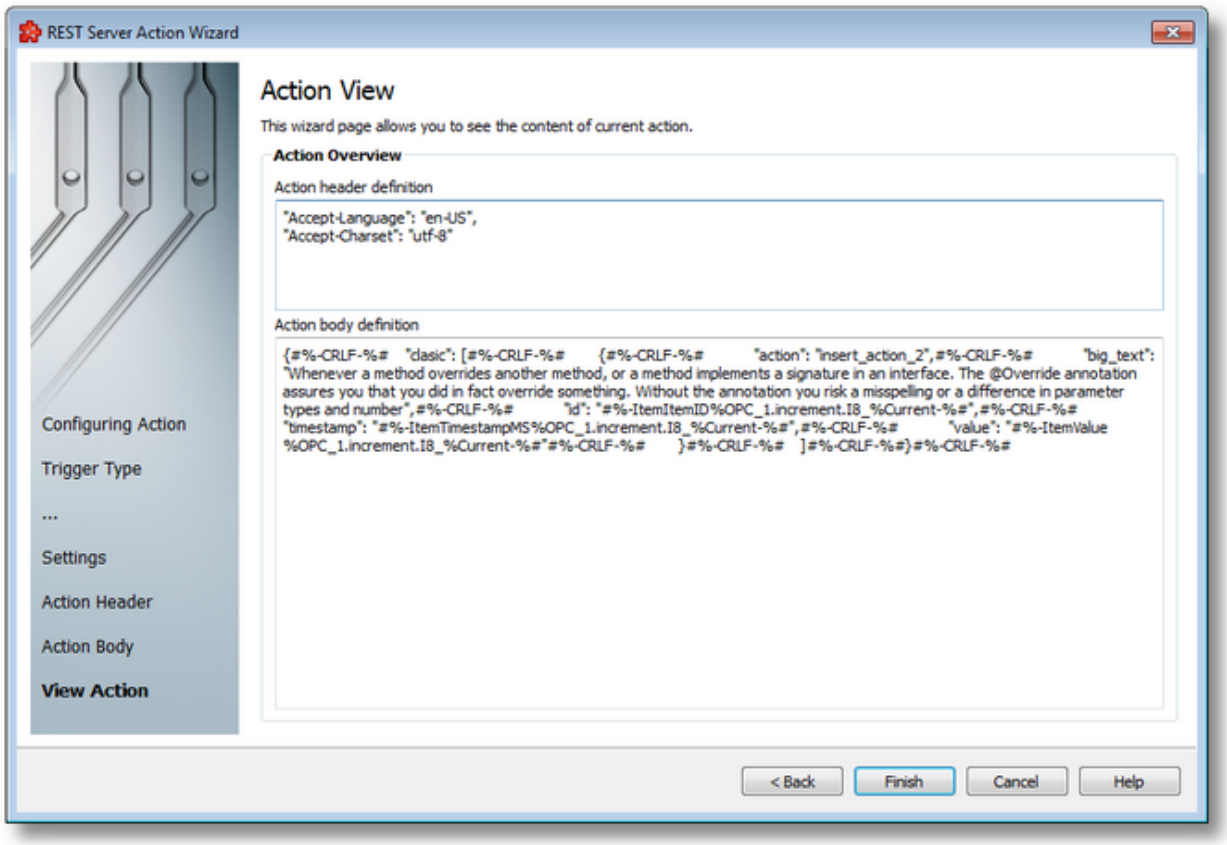
© Softing Industrial Automation GmbH



This is a read-only view, used just for reviewing the correctness of the content. The user can only copy the content from both boxes.

Changing the action body content type results in different final forms of the action body. The image above shows JSON format, the images below show XML and plain text.





© Softing Industrial Automation GmbH

In case of plain text TAB and ENTER characters are saved using special placeholders, which the application will replace at runtime and thus the original form of the body content will be restored.

<<< REST Action Wizard - Action Body Definition

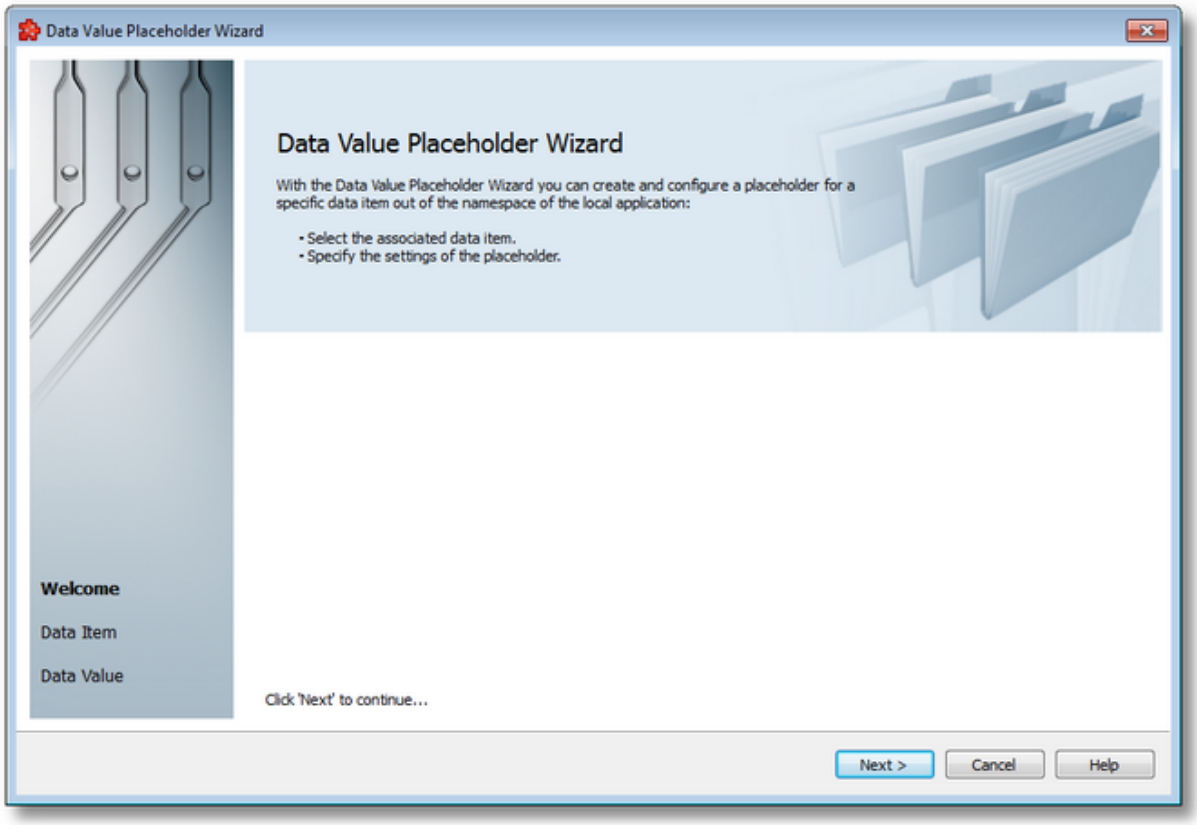
7.3.3.6.4 Data Value Placeholder Wizard

▾ With the Data Value Placeholder Wizard you can create a placeholder from a configured data source:

- Obtain a general description of the wizard @ Data Value Placeholder Wizard - Welcome
- Select a placeholder item @ Data Value Placeholder Wizard - Data Item
- Select the data value and source @ Data Value Placeholder Wizard - Data Value

The Welcome wizard page describes the individual steps for defining the placeholder:

- Select an item from the local dataFEED OPC Suite application namespace or provide a valid wildcard.
- Choose the data value and the data source for the selected item.



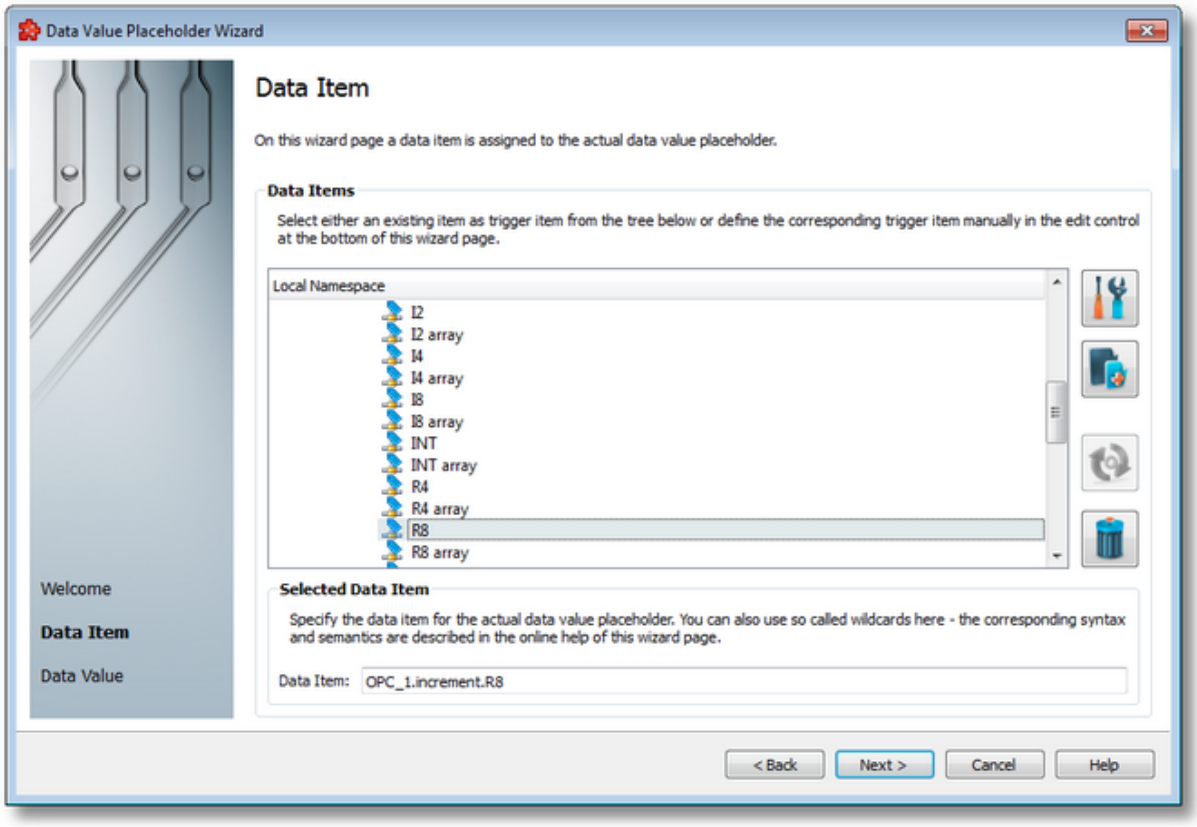
Continue to the next wizard page using the *Next* button or quit the wizard pressing the *Cancel* button.

Data Value Placeholder Wizard - Data Item >>>

The Data Item wizard page allows selecting the placeholder item. The placeholder item can be:

- Any item originating from a *Server Connection*, from the *Local Items* or from the *System Items* can be selected, choosing a tag from the *Local Namespace*.

© Softing Industrial Automation GmbH



Local Namespace Tree Handling

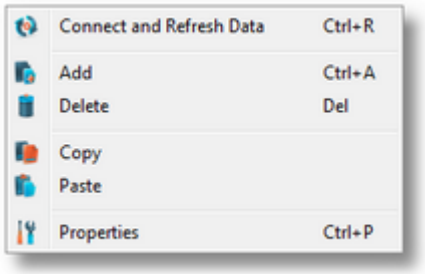
The *Local Namespace* tree offers the possibility to explore the entire namespace of the local dataFEED OPC Suite application in order to select a data item. After the selection is made, the *Data Item* edit box is populated with the selected item ID.

The entire namespace of the local dataFEED OPC Suite application can be managed through this wizard page by performing certain operations using the buttons on the right, the context sensitive menu or the keyboard shortcuts:

- Open the item properties dialog using the *Properties of the selected item* button  , pressing the keyboard shortcut CTRL+P or selecting the *Properties* action from the context menu. Depending on the type of the selected item, a special Item Properties dialog opens:
    - If the selected item is under Local Items the Modify Local Item dialog is opened.
    - If the selected item is under a dynamic server connection the Modify Dynamic Server Item dialog is opened.
    - If the selected item is under a static server connection the Modify Static Server Item dialog is opened.
- Add a new item using the *Add a new item*  button, pressing the keyboard shortcut CTRL+A or selecting the *Add* action from the context menu. Depending on the type of the selected node, a special Add Item dialog opens:

  - If the selected node is under Local Items the Add Local Item dialog is opened.
  - If the selected node is under a dynamic server connection the Add Dynamic Server Item dialog is opened.
  - If the selected node is under a static server connection the Add Static Server Item dialog is opened.
- Copy the selected item to clipboard by selecting the *Copy* action from the context menu.
- Paste the previously copied item from clipboard by selecting the *Paste* action from the context menu. Items originating from the Local Items cannot be pasted under server connections, and vice-versa; an error message will be displayed in both cases. If an item is pasted in a static (filtered) server connection and the item does not exist on the server an error message is displayed.
- Connect to the server and refresh the data items by using the *Refresh data items* button, pressing the keyboard shortcut CTRL+R or selecting the *Connect and Refresh Data* action from the context menu. The namespace of the selected server connection is recreated; if manually added items not referenced in the actual configuration exist, those items are deleted from the namespace.
- Delete the selected item and all of its child items by using the Delete data items button , pressing the DELETE / DEL key or selecting the Delete action from the context menu. Items which are included in any Data Processing / Data Destination add-ons will not be deleted but an error message will be displayed instead.

An example of the context menu can be seen below:



Attention: The *Local Namespace* root node, the *Local Items* root node and the server connection root nodes cannot be deleted. An item that is used in consumers or a system item cannot be deleted either.

Note: The copy/paste actions work on the selected tag/property or on the whole selected branch. The copy action is not available for the Local Namespace root or for the server connection root nodes; the paste action is not available for items of type property (as they cannot have any children).

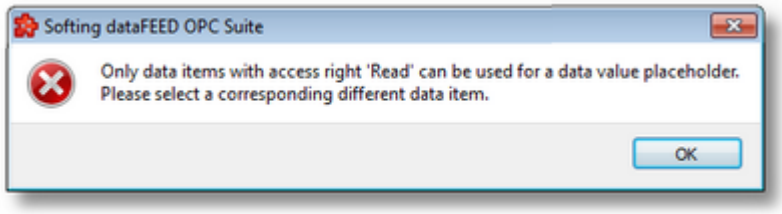
The entire namespace of the local Softing dataFEED OPC Suite application is exposed in this wizard page. The tree root node name is composed of the *Local Namespace* keyword followed by the instance name.

© Softing Industrial Automation GmbH

Selected Data Item Handling

When an item from the Local Namespace is selected, the *Data Item* edit box is populated with the item ID.

Note: Only tag and property items are allowed to be used in placeholder. If a node item is selected, an error message dialog will appear when the *Next* button is pressed.



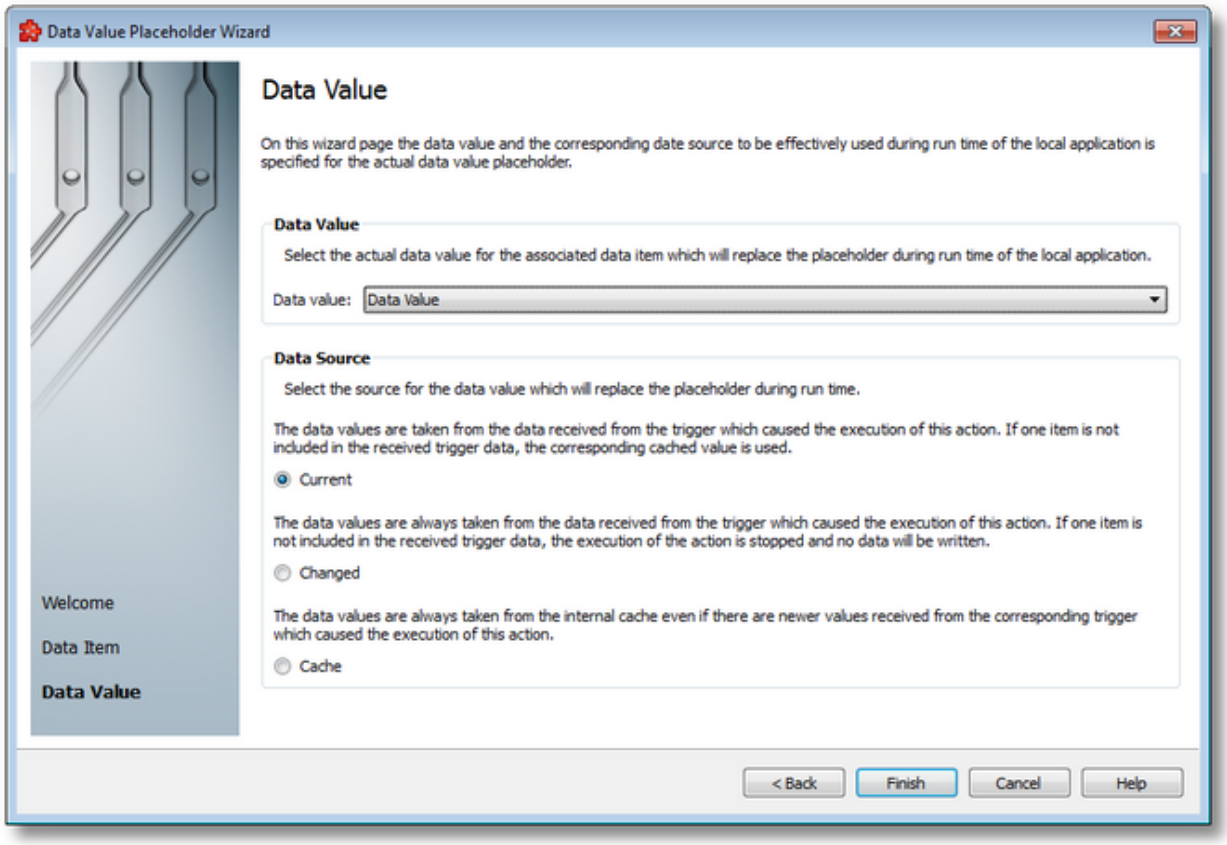
Regular expressions with special characters (asterisk, question mark, square brackets) are not supported by this module.

Proceed to the next page to set the item data value and source by using the *Next* button, return to the previous wizard page by using *Back* button, or quit the wizard by pressing the *Cancel* button.

<<< Data Value Placeholder Wizard - Welcome Data Value Placeholder Wizard - Data Value >>>

The Data Value wizard page provides the possibility to set the properties of the placeholder:

- Data Value - select the actual data value for the associated data item which will be replaced during the run time of the local dataFEED OPC Suite application.
- Data Source - select the source for data value which will be replaced during the run time of the local dataFEED OPC Suite application.



Data Value

The following data value attributes may be used:

Attribute	Description
Data Value	The value of the data source server item.
Item Identified	The ID of the data source server item.
Quality as Integer	The data source server item quality as integer.
Quality as String	The data source server item quality as string.
Timestamp (without milliseconds)	The data source server item timestamp as date time without milliseconds.
Timestamp (with milliseconds)	The data source server item timestamp as date time including milliseconds.
Timestamp (milliseconds part only)	The milliseconds part of the data source server item timestamp.
Internal Timestamp (without milliseconds)	The timestamp of data change in the local dataFEED OPC Suite application as date time without milliseconds.

© Softing Industrial Automation GmbH

Internal Timestamp (with milliseconds)	The timestamp of data change in the local dataFEED OPC Suite application as date time including milliseconds.
Internal Timestamp (milliseconds part only)	The milliseconds part of timestamp of data change in the local dataFEED OPC Suite application.

Data Source

The following data sources can be selected:

Data Source	Description
Current	The value used is the current one. This means that if the item is included in the transaction, the received value is used. If the item is not included, the cached value is used.
Cache	The cached value is always used, even if the item is present in the incoming transaction.
Changed	The item must be included in the incoming transaction as the current value will be used.

Attention: If the action trigger is set to *Data Change* and the corresponding action contains at least one placeholder having *Changed* as the *Data Source*, that specific item must exist in the incoming transaction in order for the action to be executed. When the item is not present in the transaction, the action is invalidated.

Complete the wizard using the *Finish* button, go back to the previous wizard page using the *Back* button or discard any changes made in this wizard by pressing the *Cancel* button.

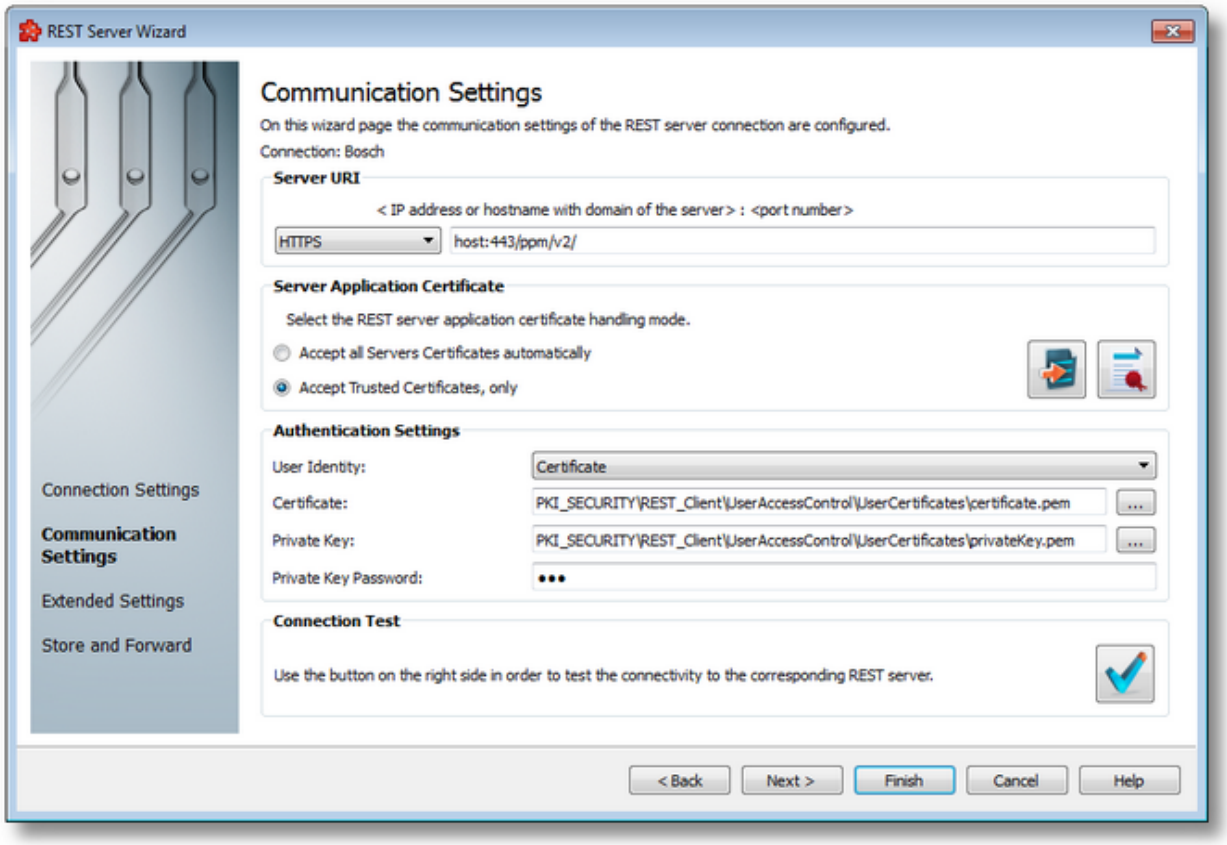
<<< Data Value Placeholder Wizard - Data Item

7.3.3.6.5 Bosch PPM Example

The dataFEED OPC Suite application supports connections with the Bosch PPM server (Production Performance Manager).

Currently, the Bosch PPM requires HTTPS authentication using Bosch certificates in order to allow the dataFEED OPC Suite application to send data to it.

An example connection configuration looks like this:

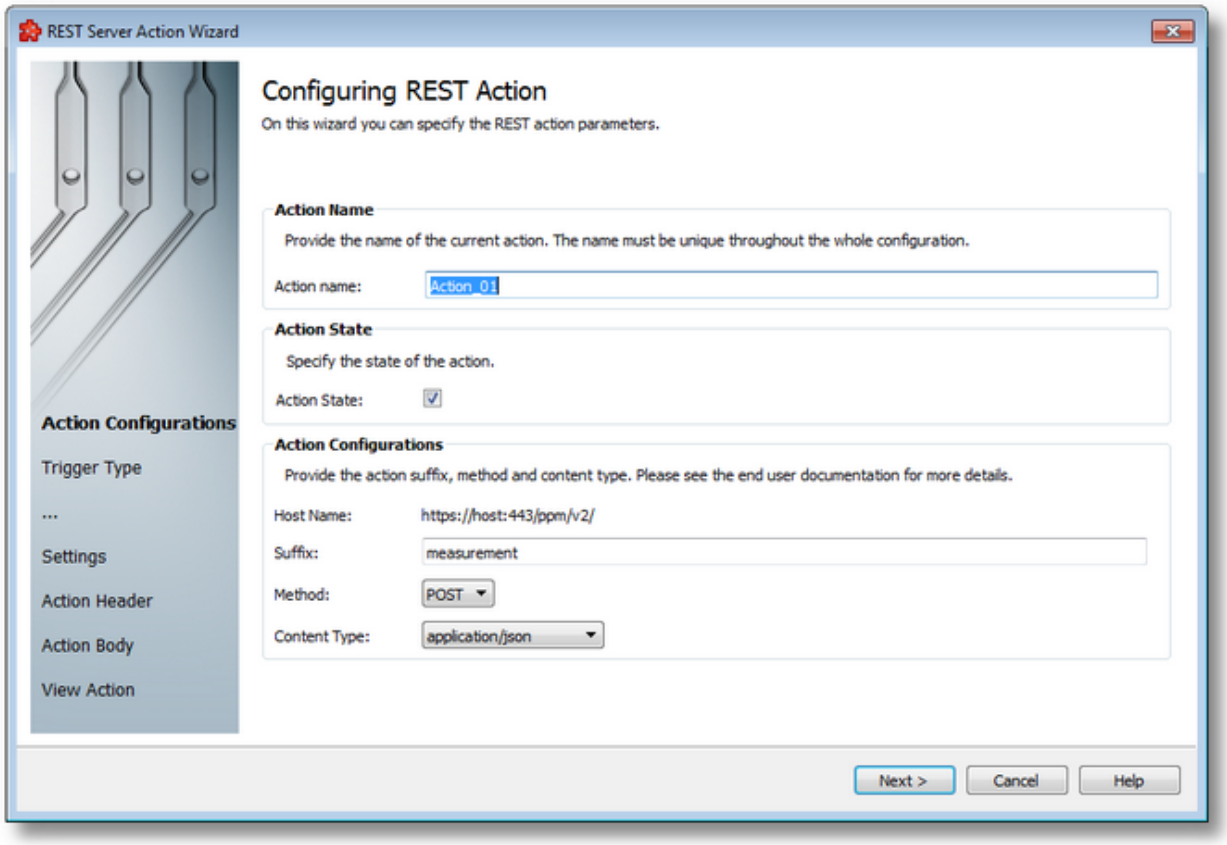


On the connection wizard page the user needs to specify the URL in the following form: host:443*/ppm/v2/*, where the host part needs to be adapted to the specific server name or IP. Default port is 443 for HTTPS.

The important part is the */ppm/v2/* suffix.

When configuring an action, it needs to look like this:

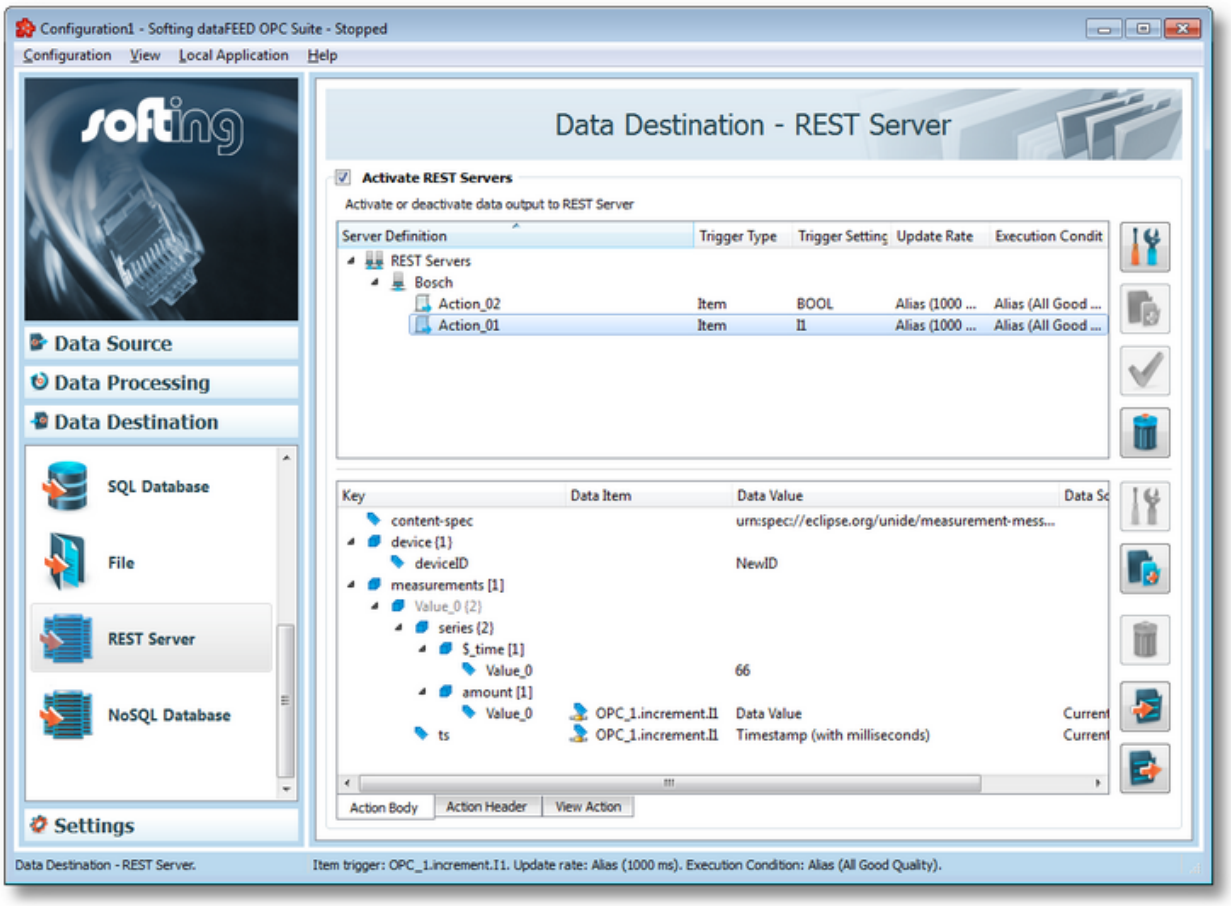
© Softing Industrial Automation GmbH



The action suffix must contain the ***measurement*** Bosch PPM endpoint.

When the action is executed the Host Name part will be concatenated with the Suffix part, thus forming the valid URL for the Bosch PPM to receive data.

The next important part is the action body itself. This is where the user can send different *Measurement Points*, to which different types of data values can be sent.



As the Bosch PPM only accepts a specific kind of JSON payload, the above example is demonstrated below:

```
{
    "content-spec": "urn:spec://eclipse.org/unide/measurement-message",
    "device": {
        "deviceID": "NewID"
    },
    "measurements": [
        {
            "series": {
                "$_time": [
                    66
                ],
                "amount": [
                    "#%-ItemValue%OPC_1.increment.I1_%Current-%#"
                ]
            },
            "ts": "#%-ItemTimestampMS%OPC_1.increment.I1_%Current-%#"
        }
    ]
}
```

Short description of the elements:

- DeviceID - the unique ID of the device to which the dataFEED OPC Suite application wants to send data
- Measurements - is an array of JSON elements, which will contain the important data
- amount - this is the most important piece of data: the measurement point.

© Softing Industrial Automation GmbH

- ts - is a timestamp value (in the example above, it will contain the timestamp of the OPC_1.increment.I1

The user can configure the other major endpoint of the Bosch PPM Server: the ***message*** endpoint.



The raw payload in this case looks like this, and it contains system diagnostics information like an alarm c

```
{
    "content-spec": "urn:spec://eclipse.org/unide/machine-message",
    "device": {
        "deviceID": "SoftingDeviceID"
    },
    "messages": [
        {
            "code": "190ABT",
            "description": "My description",
            "hint": "My hint",
            "metaData": {
                "firmware": "20130304_22.020"
            },
            "origin": "sensor-id-992.2393.22",
            "severity": "HIGH",
            "title": "My Title",
            "ts": "2017-02-21T09:30:10.123+02:00",
            "type": "DEVICE"
        },
        {
            "code": "33-01",
            "description": "Disk size has reached limit. Unable to write log files.",
            "severity": "HIGH",
```

```
            "title": "Disk size limit reached",
            "ts": "2017-02-21T01:10:10.125+02:00",
            "type": "TECHNICAL_INFO"
        }
    ]
}
```

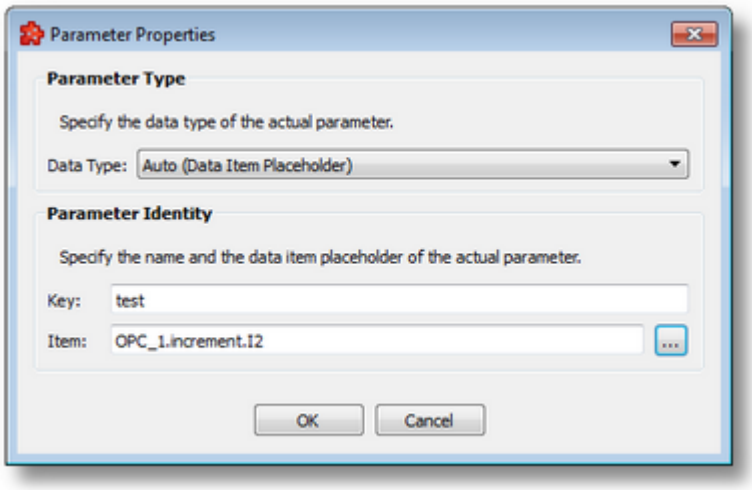
The payload usually contains a code, a description of the problem, some metadata, severity, title, type, etc who will handle these issues, based on severity, type and such.

This payload can be also parameterized with dataFEED OPC Suite placeholders of items, to transfer live

For more information about configuration details and available measurement points refer to the Bosch PP

7.3.3.6.6 Parameter Properties

The *Parameter Properties* window offers the possibility to create or modify a *JSON* data type that is the basic element of the action body's definition.



The following parameter properties can be selected in this window:

- the parameter data type: it can be auto, object, array, string, boolean, and double;
- the parameter key: it should be unique for its level in the tree;
- the parameter value: It is available only for the data types auto, string, boolean, and double.

If the action content type is *application/xml* or *application/xhtml+xml* then the parameter key should respect the following rules:

- it must start with a letter or underscore
- it cannot start with the letters xml (or XML, or Xml, etc)
- it can contain letters, digits, hyphens, underscores, and periods
- it cannot contain spaces.

© Softing Industrial Automation GmbH

Data Types

The supported data types are:

- *Auto (Data Item Placeholder)* - is an automatic type and depends on the placeholder definition
- *Object* - an unordered collection of key-and-value pairs
- *Array* - an ordered sequence of values
- *String* - double-quoted *Unicode* with backslash escape
- Boolean - true or false
- Double - double precision floating-point format

Note: The default data type is *Auto (Data Item Placeholder)*.

7.3.3.6.7 Data Values Placeholder

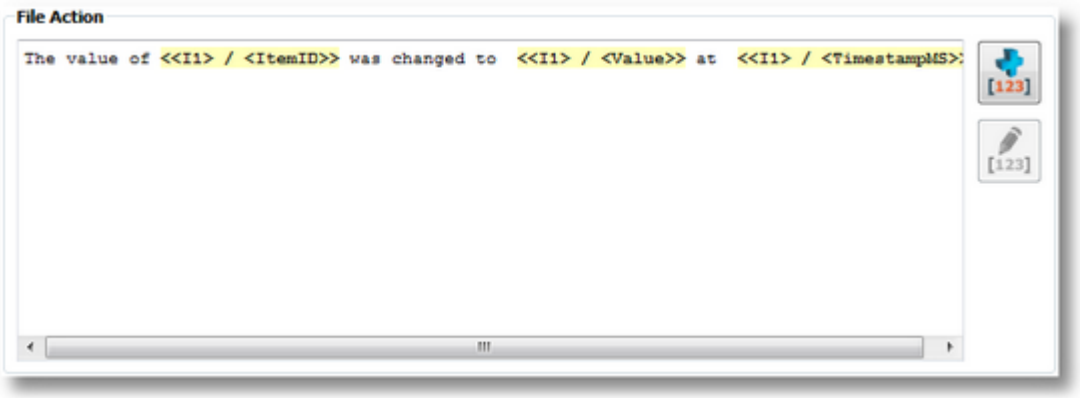
Placeholders can be used when configuring various actions (SQL, TEXT, REST_REQ and other statements). These placeholders can be defined in the *Action Wizard,* the main user interface and are saved in the Subscription file. A placeholder is determined by the following attributes:

- Data Item - an item (tag, property or node with value) originating from a *Server Connection*, from the *Local Items or* from the *System Items*.
- Data Value - the actual data value for the associated data item which will be replaced during the run time of the local dataFEED OPC Suite application.
- Data Source - the source for the data value which will be replaced during the run time of the local dataFEED OPC Suite application.

Placeholder in the user interface

Two types of placeholder editors are implemented in the Softing Configurator. Only one type of action editor is used in the REST add-on, that is the *Text Action Editor*. This placeholder editor is found in REST Action wizard page and also in the Configuration: Data Destination - REST Server main user interface page of the File add-on; the corresponding placeholder form is: <<Short Data Value> / <Short Data Item>>:

```
<<R8> / <ItemID>>
<<R8> / <Value>>
<<R8> / <QualityString>>
<<R8> / <TimestampMS>>
```



Placeholder in the subscription file

In the subscription file a placeholder has the following syntax:

```
#%-<Data Item>%<ItemID_SubscriptionID>%<DataSource>-%#
```

where:

Attribute	Description
#%- -%#	Represents the beginning and the end of a placeholder definition
Data Item	Represents the item data value.
%	Separates the Data Value, Data Item and Data Source fields in the placeholder.
ItemID_SubscriptionID	Represents the subscription item identifier.
Data source	Represents the item data source.

Example:

```
#%-ItemValue%OPC_1.increment.R8_Subscription_222%Current-%#
```

Data value:

The following data value attributes can be used (they are supported by both dataFEED OPC Suite Configurator and local dataFEED OPC Suite application):

Attribute	Subscription file form	Description
Data Value	ItemValue	The value of the data source server item.

© Softing Industrial Automation GmbH

Item Identified	ItemItemID	The ID of the data source server item.
Quality as Integer	ItemQuality	The data source server item quality as integer.
Quality as String	ItemQualityString	The data source server item quality as string.
Timestamp (without milliseconds)	ItemTimestamp	The data source server item timestamp as date time without milliseconds.
Timestamp (with milliseconds)	ItemTimestampMS	The data source server item timestamp as date time including milliseconds.
Timestamp (milliseconds part only)	ItemTimestampMSPart	The milliseconds part of the data source server item timestamp.
Internal Timestamp (without milliseconds)	ItemTimestampOCT	The timestamp of data change in the local dataFEED OPC Suite application as date time without milliseconds.
Internal Timestamp (with milliseconds)	ItemTimestampOCTMS	The timestamp of data change in the local dataFEED OPC Suite application as date time including milliseconds.
Internal Timestamp (milliseconds part only)	ItemTimestampOCTMSPart	The milliseconds part of the data change timestamp in the local dataFEED OPC Suite application.

The following data value attributes can be used (only supported by the local dataFEED OPC Suite application):

Attribute	**Subscription file form**	**Description**
Item Subscription ID	ItemSubscriptionID	The subscription ID of the item
Data Value	DataValue	The ID of the data source server item.
Data Quality as Integer	DataQuality	The data source server item quality as integer.
Data Quality as String	DataQualityString	The data source server item quality as string.
Timestamp (without milliseconds)	DataTimestamp	The data source server item timestamp as date time without milliseconds.
Timestamp (with milliseconds)	DataTimestampMS	The data source server item timestamp as date time including milliseconds.
Internal Timestamp (without milliseconds)	DataTimestampOCT	The timestamp of data change in the local dataFEED OPC Suite application as date time without milliseconds.

Internal Timestamp (with milliseconds)	DataTimestampOCTMS	The timestamp of data change in the local dataFEED OPC Suite application as date time including milliseconds.
Data Item ID	DataItemID	The value of the data source server item.

Attention: There is a significant difference between the Data Values in the tables above. The values in the second table are taken directly from dataFEED OPC Suite internal server while the values for the Data Values from the first table are taken directly from the server which holds the real Item. dataFEED OPC Suite holds only a copy of that Item.

Data Source or Item type:

One of the following Data Sources may be selected:

Item type	Description
Current	The item must be in the subscription item list. The value used is the current one, which means that if the value for the item is included in the current transaction, this value will be used. If the value is not included in the incoming transaction, the cached value will be used.
Cache	The item must be in the subscription item list. The value used is the cached one, even if that item is included in the incoming transaction.
Changed	The item must be in the subscription item list. The value used is the changed one, which means that the entire action will be invalidated if the item is not present in the incoming transaction.

Examples:

```
'#%-ItemTimestampOCTMSPart%OPC_1.increment.R8_Subscription_222%Current-%#'
'#%-ItemTimestampOCTMS%OPC_1.increment.R8_Subscription_222%Changed-%#'
'#%-ItemTimestampOCT%OPC_1.increment.R8_Subscription_222%Cache-%#'
```

Attention: If the *Changed* Data Source is used in an action and the corresponding action trigger is a *Data Change* trigger, make sure that a transaction which contains data changes for all the items used in the current action will be generated by the server; otherwise, the entire action will be invalidated if an item has no associated data change in the incoming transaction.

7.3.3.6.8 Configuration Files

All the REST configuration settings are stored in two separated files:

- Tasks file - contains general descriptions of the File definitions.

© Softing Industrial Automation GmbH

- Subscriptions file - which contains all the descriptions related to the stored data:
    - what data will be stored.
    - what is the frequency of storing the data.
    - where and when will the data be stored.

Note: The path and file name of these configuration files are fixed and not configurable.

The task and subscription files are stored in the *Tasks* folder under the instance path. For task configuration a text file called *Tasks.txt* and for subscription configuration a text file called *Subscriptions.txt* will be created in the *Tasks* configuration folder. Although these files can be manually edited, using the dataFEED OPC Suite Configurator is advisable.

Note: The recommendation is to use only the dataFEED OPC Suite Configurator to edit these files. This ensures that all the definitions will be valid.

If the configuration is manually edited try loading the configuration in the dataFEED OPC Suite Configurator as all the configuration files are automatically validated at load-time and, if there are any mistakes, an error message will be displayed. The dialog will contain an appropriate description and will point to the first invalid line of the local items configuration file.

Both configurations files are plain text files encoded in UTF-8 (thus providing full UNICODE compatibility) and they have line-oriented structure. The elements of a line are tab-separated. The format of one line is the following:

```
Line type<TAB>Line data
```

Task Configuration File

The following line types will be defined:

Line Type	Description
REST_TIMESTAMPFORMAT	The format of the global time and date values
REST_TIMEFORMAT	The global time zone value
REST_PROXY	Parameters defined for a proxy connection
REST_REQ_PROC	Parameters used for the internal queue and processing settings
REST	Describes the parameters of a server connection.
Commented Lines (#)	Any line that starts with the "#" character is considered a comment line which is ignored during evaluation.

The REST_TIMESTAMPFORMAT line

The REST_TIMESTAMPFORMAT line allows defining parameters needed by the REST add-on to format time and date values, globally:

```
REST_TIMESTAMPFORMAT<TAB>Format
```

Example:

```
REST_TIMESTAMPFORMAT    mm/dd/YYYY HH:MM:SS pp
```

The REST_TIMEFORMAT line

The REST_TIMEFORMAT line allows defining the time zone, globally (utc or local):

```
REST_TIMEFORMAT<TAB>Value
```

Examples:

```
REST_TIMEFORMAT   utc
REST_TIMEFORMAT   local
```

The REST_PROXY line

The REST_PROXY line allows defining all the parameters needed by a proxy connection:

```
REST_PROXY<TAB>Proxy Type<TAB>Host/IP<TAB>Port<TAB>Authentication Type<TAB>User Crend
```

Examples:

```
REST_PROXY  HttpProxy   192.168.150.171   3128  UsingProxyAuth    admin C122ED818D7C5
REST_PROXY  HttpProxy   192.168.150.171   3128  NotUsingProxyAuth
REST_PROXY  NoProxy
```

The REST_REQ_PROC line

The REST_REQ_PROC line allows defining all the parameters needed by the internal request queue, globally:

```
REST_REQ_PROC<TAB>Maximum queue size per connection<TAB>Execution type
```

Maximum queue size refers to the number of allowed request the internal request queue can hold for one connection. Each connection queue will be able to hold that number of requests.

Execution type specifies whether or not the execution of requests should be parallel (currently 6 at a time, because of Qt limitations) or sequential (one at a time).

Examples:

```
REST_REQ_PROC     10000 Parallel
REST_REQ_PROC     100   Sequential
```


© Softing Industrial Automation GmbH

The REST line

The REST line allows defining all the parameters needed by the REST add-on to send data to a REST Server:

```
REST<TAB>Connection ID<TAB>Connection Name<TAB>Status<TAB>Protocol Type<TAB>Host Name
```

Parameter	Description
REST	The type of the line, predefined keyword.
Connection ID	Unique identifier within the task file (used for matching the statements in the corresponding subscription file).
Connection Name	The name of the connection visible in the Configurator
Status	The status of the connection: Active or Inactive
Protocol Type	TCP, HTTP or HTTPS
Host Name	The server address, or IP. A '/' character is compulsory at the end
Authentication Type	Anonymous, Username and Password, or Certificate (only for HTTPS). The parameters that follow may differ based on selection.
Send all values as text	Send all data as text: SendText or DontSendText
Send arrays in quotes	Send arrays in quotation marks: SendQuotes or DontSendQuotes
Request Timeout	Value in seconds of the time to wait for a request to be aborted. 0 means waiting until it completes
Store and Forward status	Whether or not Store and Forward is active: Active or Inactive
Store and Forward folder path	The path of the folder where the file will be created in.
Store And Forward file size	Maximum amount of space used for the file
Store And Forward requests size	Maximum number of requests stored in the file
Overwrite requests	Whether or not to overwrite older requests if the storage gets full: ON of OFF
Packed requests	Whether or not the send original requests or packed in bigger payloads: TRUE or FALSE

Max. number of forwarded requests	How many requests to pack together in one
Time zone	UTC or Local
Date-time format	The format of the time and date items
Autoaccept certificates	TRUE or FALSE
Certificate retrieval interval	Value in seconds of the interval after which the application will validate the server certificate again at action execution

Example:

```
REST  REST_01     REST_Server_01    Active      HTTPS 192.168.150.114:39321/  Anonymo
REST  REST_02     REST_Server_02    Active      HTTP  127.0.0.1:80/     UsernameAndPa
REST  REST_03     REST_Server_03    Active      HTTPS 127.0.0.1:80/     Certificate P
```

Task File - Example:

```
#
#     REST - configuration
#
REST_TIMEFORMAT   utc
REST_PROXY  NoProxy
#
REST_REQ_PROC     10000 Parallel
#
REST  REST_01     REST_Server_01    Active      HTTPS 192.168.150.114:39321/  Anonymo
#
REST  REST_02     REST_Server_02    Active      HTTP  127.0.0.1:80/     UsernameAndPa
#
REST  REST_03     REST_Server_03    Active      HTTPS 127.0.0.1:80/     Certificate P
#
```

Subscription Configuration File

The *Data Destination - REST Server* generates multiple line types:

- common lines which are specific to all the consumer add-ons: condition lines, subscription lines and item lines;
- specific lines: modifier lines and copy lines.

Common Lines:

The following line types are common lines and will be used to configure different functionalities:

© Softing Industrial Automation GmbH

Line Type	Description
Subscription	Defines the update rate with which the current action will be executed.
Condition	Defines the execution condition for the current action.
Item	This line defines a trigger which is fired if the item data has changed. The trigger data includes all other items which have changed at the same time (were reported in the same data change callback from the OPC server). An item data change consists of a different value or quality of the respective item compared to the previous one.
Timer	This line defines a trigger which is fired at specific time and date or at specific intervals.
Commented Lines (#)	Any line that starts with the "#" character is considered a comment line which is ignored during evaluation.

The subscription file is divided into two sections:

- The global area:

  The global area extends from the first subscription line to the first trigger line (e.g: ITEM, TIMER lines).

  This area contains definitions valid for the entire subscription file.

- The local area:

  The local area covers all the actions between two trigger lines in the subscription file.

  The lines in this area are only executed if the parent trigger fires.

Subscription Line

A subscription line has the following format:

```
SUBSCRIPTION<TAB>Subscription ID<TAB>UPDATERATE<TAB>Update rate value<TAB>ACTIVE<TAB>
```

The subscription line contains the following configurable parameters:

Parameter	Description
SUBSCRIPTION	The type of the line.
Subscription ID	It is used to identify the subscription and must be unique.

UPDATERATE	Predefined keyword.
Update rate value	Defines the update rate with which the current action will be executed.
ACTIVE	Predefined keyword.
Active value	Active value which can be TRUE or FALSE.

The subscription lines always contain the global subscription line as well as all subscriptions which are used at least once in consumer actions. The global subscription is represented by the subscription line with the empty *Subscription ID*. The default update rate value for the global subscription is *1000* milliseconds.

The value of the update rate is saved in milliseconds. This value may not be set to 0 inside the dataFEED OPC Suite Configurator but it can be manually set by editing the subscription file with an external text editor. An update rate value of 0 means *as fast as possible* but this is not recommended from a resource-related point of view.

Setting a subscription to be active or inactive is an advanced feature which cannot be configured from within the dataFEED OPC Suite Configurator. By default the active value for each and every subscription is set to *TRUE* (meaning that all subscriptions are active since creation). If a subscription's active value is set to *FALSE* then the data changes of the items configured to be part of that subscription will be ignored.

Examples:

```
SUBSCRIPTION<TAB><TAB>UPDATERATE<TAB>1000<TAB>ACTIVE<TAB>TRUE
SUBSCRIPTION<TAB>Subscription_300<TAB>UPDATERATE<TAB>300<TAB>ACTIVE<TAB>TRUE
```

Condition Line

A condition line defines whether an action below an item trigger will be executed or not (when the item trigger is fired). This can be used in other configuration lines as well to condition the execution of an item action. One example of this is the conditional execution of a SQL statement in the Database add-on. A condition line has the following format:

```
CONDITION<TAB>Condition ID<TAB>Condition Statement
```

The condition line contains the following configurable parameters:

Parameter	**Description**
CONDITION	The type of the line, predefined keyword.
Condition ID	Unique condition identifier.
Condition statement	Single condition statement.

© Softing Industrial Automation GmbH

Conditions can be used to react to communication errors. Various actions can be executed depending on the quality of items.

The following condition statements are supported:

Condition	Description
AllGoodQuality	The trigger actions are executed only if all items in the transaction have the quality *Good*.
Always	The trigger actions are executed always, ignoring the quality of the items in the incoming transaction.
ItemQualityChanged	The trigger actions are executed only if the quality of at least one item in the transaction has changed.
OneBadQuality	The trigger actions are executed if exactly one item in the transaction has the quality *Bad*.

The condition lines always contain the global condition line as well as all conditions which are used at least once in consumer actions. The global condition is represented by the condition line with the empty *Condition ID*. The default condition statement value for the global condition is *Always*. As a general rule all the condition lines need to be declared in the global section of the subscription file. If no condition is defined the Always condition is applied as default.

Examples:

```
CONDITION<TAB><TAB>Always
CONDITION<TAB>AllGoodQuality<TAB>AllGoodQuality
```

Item Line

An item line is used to add an item to a subscription. The associated actions are executed each time the value or the quality of the specified data items changes during the run time of the local dataFEED OPC Suite application.

An item line has the following format:

```
ITEM  Local Items.Scalars.Always_Trigger_ Local Items.Scalars.Always_Trigger
```

The item line contains the following configurable parameters:

Parameter	Description
ITEM	The type of the line, predefined keyword.
Subscription Item ID	Unique subscription item identifier (implicitly created by concatenating the Server Item ID and the

	Subscription ID with the underscore character in between them).
Server Item ID	Item identifier within the local dataFEED OPC Suite application address namespace. This could be a server connection item ID or a local item ID.
Subscription ID	Subscription identifier (defined above, see Subscription Line).

This trigger is fired if the item data or quality has changed (also referred to as data change). The trigger data includes all other items which were included in the data change (items which had their data change reported in the same transaction).

Note: A single line consisting of the *Server Item ID* and *Subscription ID* is added to the subscription file for an item used several times with the same update rate in any consumer add-ons. This optimizes the local dataFEED OPC Suite application throughput.

Example:

```
ITEM<TAB>increment.I1_Subscription_300<TAB>increment.I1<TAB>Subscription_300
```

Note: The subscription item ID can be empty. In this case, however, only the Server Item ID can be used to identify the item.

Timer Line

This trigger is fired at a specified time and date or at specific intervals. The associated actions are executed on a specific schedule or periodically during the run time of the local dataFEED OPC Suite application.

The format of the *Timer Line* is:

```
TIMER<TAB>Timer ID<TAB>INTERVAL<TAB>Rate in milliseconds
```

or

```
TIMER<TAB>Timer ID<TAB>SCHEDULE<TAB>Schedule range<TAB>First execution<TAB>Rate<TAB>L
```

The interval timer contains the following configurable parameters:

Parameter	Description
TIMER	The type of the line, predefined keyword.
Timer ID	Unique timer identifier.

© Softing Industrial Automation GmbH

INTERVAL	Denotes an interval timer trigger, predefined keyword.
Interval value	Defines the interval in milliseconds at which the timer will trigger the associated action(s).

The timer schedule contains the following configurable parameters:

Parameter	**Description**
TIMER	The type of the line, predefined keyword
Timer ID	Unique timer identifier.
SCHEDULE	Denotes a scheduled timer trigger, predefined keyword.
Schedule range	Defines the range of the schedule (see below the table).
First execution	Defines the time the timer should begin firing events (see below the table).
Rate	Defines the rate at which the timer should trigger the associated action(s).
Last execution	Defines the time the timer should stop firing events (see below the table).
List Of Days	Should be used only with the Week and Month schedule. It must contain the days on which the execution is desired. If the list of days is not present, the obtained functionality is the same as by using the day marker in every day of the week/month.

The timer's *Schedule range* may have one of the following values:

- MINUTE
- HOUR
- DAY
- WEEK - List of days: 1d = Monday, 2d = Tuesday, …, 7d = Sunday.
- MONTH - List of days: 1d = First day of month, 2d = Second day of month, …, 30d = Thirtieth day of month, 31d = Thirty-first day of the month.

The *First execution* defines the time the timer should begin firing events.

Example:

```
1d13h10m15s
```

Note: If parts of the execution time are 0, they do not have to be in the string, e.g. 1d0h10m0s = 1d10m

The timer *Rate* defines the event firing rate. An empty timer *Rate* means a single execution.

Last Execution: The last execution time of the schedule. Could be empty.

List of Days: Shall be used only with the Week and Month schedule. It should contain the days on which execution is desired. If the list is not specified the same functionality can be achieved by using the day marker in the first and last execution time.

Note: If the List of Days and the day marker in first and last execution time are used in the same timer trigger the two setting types should match the marked days!

Examples:

```
# Following timer fires events every 250 milliseconds (4 times a second):
TIMER<TAB><TAB>INTERVAL<TAB>250
# Following timer fires events every 20 seconds of a minute starting at second 10 (eg
TIMER<TAB><TAB>SCHEDULE<TAB>MINUTE<TAB>10s<TAB>20s
# Following timer fires events every 4 hours of a day starting at 6:30, ending at 20:
# (e.g. 6:30, 10:30, 14:30, 18:30).
TIMER<TAB><TAB>SCHEDULE<TAB>DAY<TAB>6h30m<TAB>4h<TAB>20h
# Following timer fires events every 4 hours of a day starting at 6:30, ending at 20:
# Friday (e.g. 6:30, 10:30, 14:30, 18:30 of each day).
TIMER<TAB><TAB>SCHEDULE<TAB>WEEK<TAB>6h30m<TAB>4h<TAB>20h<TAB>1d2d3d4d5d
# Following timer fires events every 4 hours starting at 6:30, ending at 22:45 (e.g.
# 18:30, 22:30), on the 1th, 2th, 10th, 20th of every month.
TIMER<TAB><TAB>SCHEDULE<TAB>MONTH<TAB>6h30m<TAB>4h<TAB>22h45m<TAB>1d2d10d20d
# Following timer fires events every 1 hours, 3 minutes and 3 seconds starting at 00:
# 23:59:59 (e.g. 00:00:01, 01:03:04, 02:06:07 etc), on the 3rd, 6th, 9th, 12th, 15,th
TIMER<TAB>Timer_02<TAB>SCHEDULE<TAB>MONTH<TAB>1s<TAB>1h3m3s<TAB>23h59m59s<TAB>3d6d9d1
```

Note: Make sure that only used subscription definitions are present in the configuration file (resource-related point of view).

Specific Lines:

The following line types are REST Server specific and will be used to configure different functionalities:

Line Type	Description
REST_REQ	Defines a REST Server action statement.

The format of the REST_REQ line is the following:

```
REST_REQ<TAB>Condition ID<TAB>Connection ID<TAB>Action Name<TAB>Action Status<TAB>Suf
```

Example:

© Softing Industrial Automation GmbH

```
REST_REQ    AllGoodQuality    REST_01    Action_01   Active      test  PUT   {"Accep
```

The REST_REQ line contains the following configurable parameters:

Parameter	Description
REST_REQ	The type of the line, predefined keyword.
Condition ID	Condition identifier (defined above, see Condition Line).
Connection ID	REST connection ID
Action name	The name of the action visible in the Configurator
Action Status	Specifies whether or not the action will be executed: Active or Inactive
Suffix	The resource name or part of the URL that will be concatenated with the server URL defined in the connections. No '/' is needed at the beginning
HTTP method	POST or PUT
Header	The serialized form of the action header
Body	The serialized form of the action body
Content Type	The HTTP content type of the body
Trigger ID	Associated trigger identifier (one of the above defined: Item Trigger, Timer Trigger)

The action body parameter configures the text which will be sent to the REST server when the corresponding action is executed.

Example of REST Server configuration lines and the appropriate ITEM lines:

```
ITEM  Local Items.Scalars.OneBadQuality_Trigger2_     Local Items.Scalars.OneBadQuali
REST_REQ    AllGoodQuality    REST_01    Action_01   Active      test  PUT   {"Accep
```

In order to offer a flexible configuration for REST_REQ statements, placeholders can also be used. Placeholders are syntactical entities that are replaced with their associated values each time the statement is evaluated during the run time of the local dataFEED OPC Suite application (before the request is sent to the server).

Note: The *Condition ID* used in REST_REQ line definition must be defined in the current Subscription File; the default condition may also be used.

Note: The *Connections ID* used in a REST_REQ line definition must be defined in the associated Task File.

The REST_REQ actions will be executed on every data change event, however their execution will also depend on both the defined conditions and the type of the fired trigger.

If the actions are associated with triggers of type *Item*, the respective actions are evaluated on each item change, but are only executed if all of the following conditions are true:

- The subscription item ID of the action has to be defined in the configuration.
- A change occurs for the respective item.
- The action condition is true.
- The placeholders in the action can be replaced by their respective values (the changed data source may invalidate the action).

If the actions are associated with triggers of type *Timer*, the respective actions are evaluated and executed only if all of the following conditions are true:

- The subscription item ID of the action has to be defined in the configuration.
- The action condition is true.
- The placeholders in the action can be replaced by their respective values (the changed data source may invalidate the action).

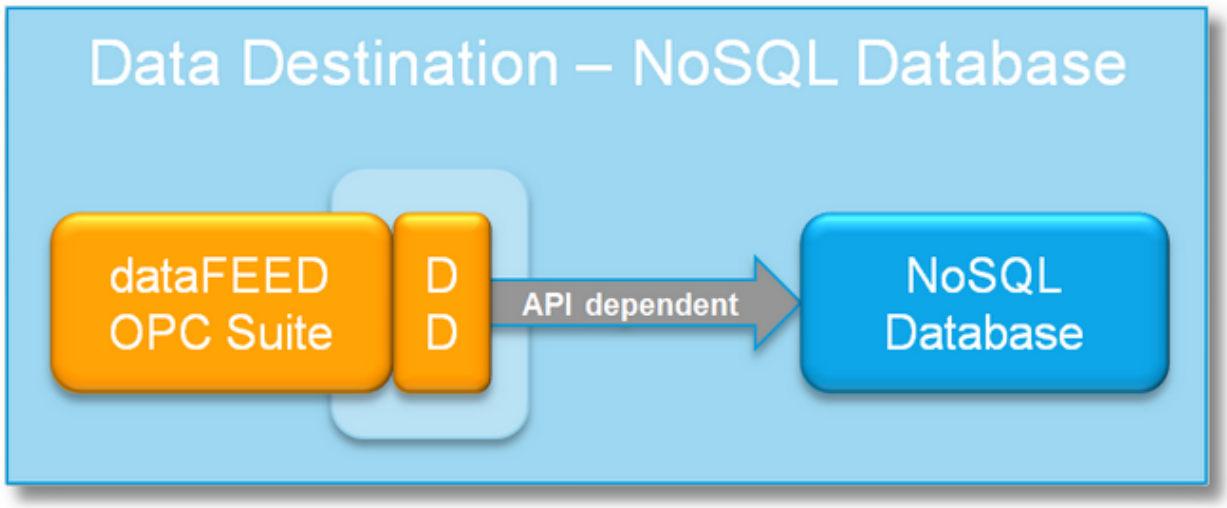
Subscription File - Example:

```
CONDITION          AllGoodQuality
CONDITION    AllGoodQuality     AllGoodQuality
#
SUBSCRIPTION             UPDATERATE  1000  ACTIVE      TRUE
#
MODIFIER           V
#
#
ITEM  Local Items.Scalars.OneBadQuality_Trigger2_      Local Items.Scalars.OneBadQuali
REST_REQ     AllGoodQuality     REST_01      Action_01    Active       test  PUT    {"Accep
#
ITEM  Local Items.Scalars.i1_ Local Items.Scalars.i1               I1
REST_REQ           REST_01      Action_02    Active       suffix       POST  {}     {}     a
#
ITEM  OPC_1.increment.I8_      OPC_1.increment.I8                  I8
```

7.3.3.7 NoSQL Database

This section of the Softing dataFEED OPC Suite online help explains how a NoSQL Client is configured as data destination in order to send data to specific NoSQL database servers using HTTP/HTTPS protocol or proprietary interfaces of the supported NoSQL Databases.

© Softing Industrial Automation GmbH

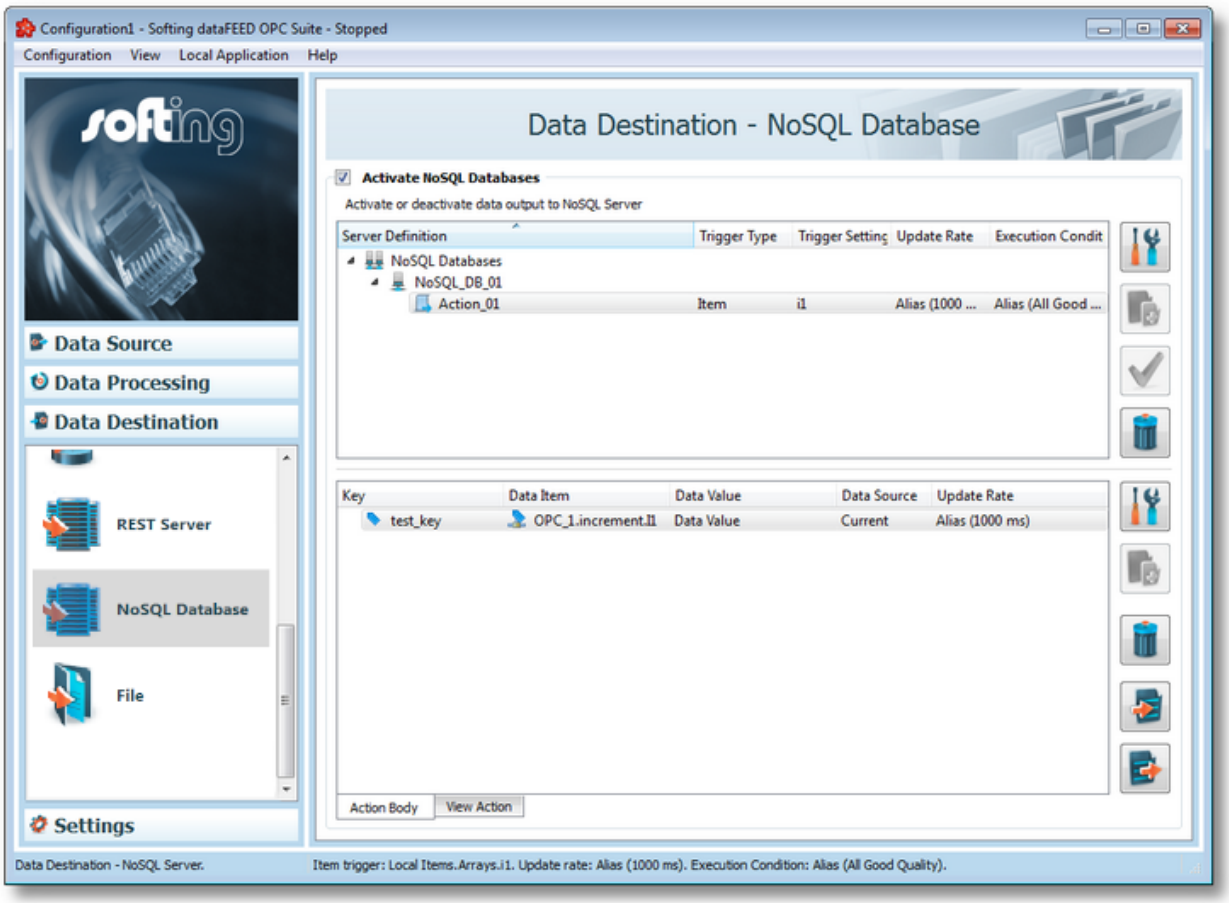


In case of NoSQL databases with REST interfaces using the HTTP/HTTPS protocol to communicate with clients, GET, POST, PUT and other HTTP operations are used to query, create and manipulate resources and content on the server. In this case it is possible to specify whether to create new database (POST) entries or to update existing ones (PUT).

Databases that have their own proprietary interfaces, the communication is done through that API, and the operation used cannot be configured.

7.3.3.7.1 Configuration

In order to configure connections to NoSQL databases expand the **Data Destination** section in the left panel and then select the NoSQL Database icon. The NoSQL Database connection overview will be displayed.



Add a NoSQL Database connection

A new NoSQL Database connection can be defined by using the *Add* button or the menu command "Add NoSQL DB Connection" in the context menu. The NoSQL Database connection wizard will show up. The Ctrl+A keyboard shortcut has the same effect.

Modify an existing NoSQL Database connection

An existing connection to a NoSQL Database can be modified. To do so, select the line in the connection overview that contains the connection to be modified. Then the *Properties* button or the "Properties" command from the context menu can be used to open the NoSQL Database connection wizard.

Add an action definition to an existing NoSQL Database connection

Adding a new action to a selected server connection is accomplished by pressing the *Add* button or the "Add REST Server Action" context menu item. Also, the Ctrl+A key combination is

© Softing Industrial Automation GmbH

available for this operation. The NoSQL Action wizard will be opened where you can go through different wizard pages to specify action details.

Modify the action settings of an existing NoSQL Database connection

To modify an existing action, select one from the list and by double clicking it, or pressing the *Properties* button, the NoSQL Action wizard will be opened, and you will be able to go through the same wizard pages as when the action was created.

Connection test

Click the *Connection Test* button to check the availability of the selected server or the correctness of the authentication and security settings. If the connection cannot be established the error message and code returned by the network or the server will be shown in the dialog.

Remove a connection/action

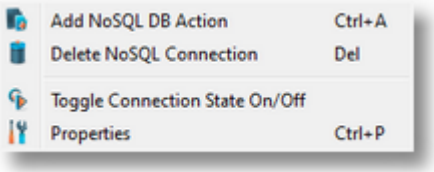
Delete the selected server connection or action definition by using the *Delete* button, pressing the DEL / DELETE key on the keyboard or selecting the appropriate delete operation from the context menu.

Note: If no connection is selected, the *Add a NoSQL Connection* and *Properties* buttons are the only options enabled.

The *Delete* button will remove the selected connection or action from the list.

A dialog will then prompt the user to confirm the deletion.

The following image shows an example of a context menu:



Action body, header and action overview

When an action is selected, the main view page can also show the action body and header definition, as well as an overview of the final action content with item placeholders, which the Runtime will replace when it's run.

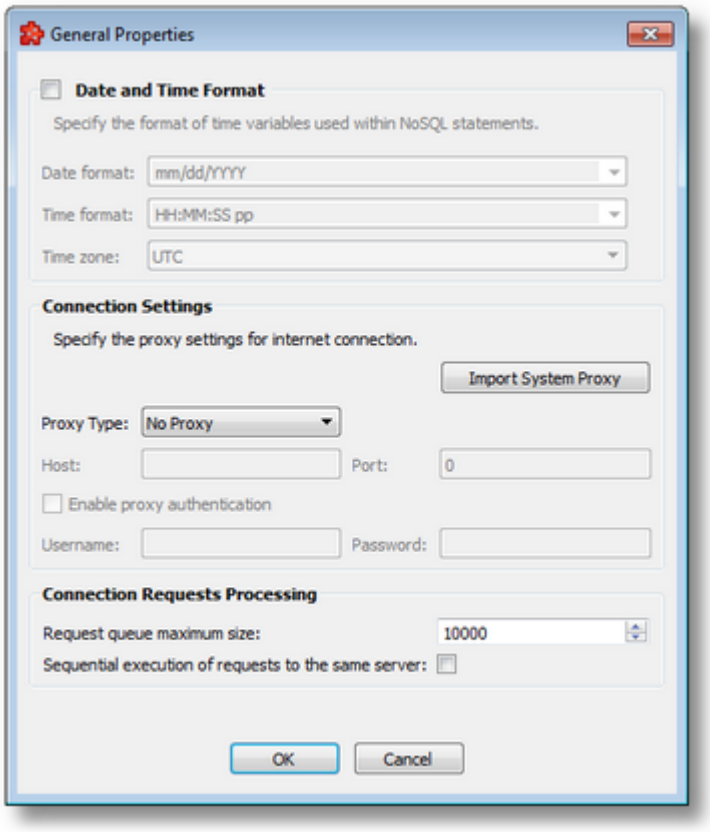
Note: The action header is only available for databases with REST interfaces, like CouchDb, so it will not be always visible. MongoDb's API does not support this.

The action body, header and overview can be seen in the three different tabs in the main screen, below the definition list: *Action Body, Action Header, View Action*.

This lower part of the main view contains tab pages which can also be found in the action wizard and allow easier access for the user to change the action body and header contents without going through the wizard pages. More information can be found on the specific help pages.

Time Format and Connection Settings

When the *NoSQL Databases* root node is selected, invoking the properties action by any means (button, context menu or keyboard shortcut) displays a dialog that allows the user to modify the time format setting. The *General Properties* dialog can be opened by using the *Properties* button  , by pressing the keyboard shortcut CTRL+P or by selecting the *Properties* action from the context menu.



The **Date and Time Format** section handles the different formats which can be specified when items of date and time are used in actions, also the source localization time: UTC (Universal Time Coordinated) or local time (local time localization). The default time zone is UTC.

© Softing Industrial Automation GmbH

The default values are `mm/dd/YYYY` for the *Date format and* `HH:MM:SS pp` for the *Time format*. Other defined values for Time and Date format are:

- *Date format*: `mm/dd/YYYY, dd-mm-YYYY, dd.mm.YYYY, dd/mm/YYYY, YYYY-mm-dd, mm-dd-YYYY`
- *Time format*: `HH:MM:SS pp, HH:MM:SS.FFF pp, HH:MM:SS.FFF, HH:MM:SS`

In addition to the default values listed in the format selection combo boxes, the user can add his/her own custom format. Custom Date and Time format strings are accepted as valid, when the format strings contain in their composition the following accepted format substrings and format delimiters. These formats allow the inclusion of specific locale information for the regional settings.

- *Valid Date format strings*: `AAAAAA, aaa, BBBBBBBB, bbb, ccccccccccccccccccc, dd, jjj, mm, UU, ww, xxxxxxxxxx, yy, YYYY`
- *Valid Time format strings*: `II, HH, MM, SS, pp, XXXXXXXX, FFFFFF, FFFFF, FFFF, FFF ,FF, F`
- *Valid Date and Time format delimiters*: `<dot>, <colon>, <slash>, <comma>, <backslash>, <vertical bar>, <semicolon>, <space>, <tab>, <minus>`

The **Connection Settings** section allows for the customization of proxy settings. The user can either import system proxy settings automatically (*Import System Proxy* button), or specify manually the host, port and authentication settings.

These settings will be applied to all connections in the plugin, and the following parts of the functionality will be affected: test connection, getting the server certificate and action execution. This means that if the network, on which the dataFEED is configured, is behind a proxy server and dataFEED is not configured with a proxy, like in the image above, the mentioned pieces of functionality will fail. In order for them to work correctly, the user needs to specify the type of proxy, the host name of the server, a port number and user credentials, if necessary.

Note: MongoDb connections do not support proxy settings at the moment, because of the MongoC driver. No configured proxy settings will apply to such connections. Test connections, getting the server certificates and executing actions will be executed regardless of proxy requirements.

The **Connection Requests Processing** section contains settings for the internal request queue used by dataFEED to manage requests. The maximum queue size and execution type can be specied. The default queue size is 10000 (minimum is 100, maximum is 60000) and execution type is parallel (checkbox unchecked). The user can change these values by entering different numbers for the queue size and by enabling sequential execution using the checkbox. When the queue starts filling up, memory consumption of the application grows and when the application is able to consume requests from the queue quickly, memory consumption decreases.

The queue is used to hold requests which are still waiting for execution because actions are executed faster than the destination server can handle. In some cases, proxy servers may cause communication lags through the network, which may also slow down server replies, thus leading to the queue filling up with pending requests.

Special system items are available for the user to keep track of the number of sent requests, failed requests, pending requests in the queue, number of stored requests in case the Store and Forward mechanism is enabled, and dropped requests, in case server certificates are invalid, rejected or for some reason the server cannot accept the sent content. System items are available per action and also totals are calculated per connections.

REST status codes

After the *CouchDB* server received a request, it will either respond with a status code if the request was successfully executed or with an error code if the request was not executed. If the dataFEED OPC Suite application receives a non-success error/status code, then it will retry or drop the request depending on the received error/status code.

The following table contains all the possible error/status codes with a brief description and the action that dataFEED OPC Suite application will perform.

Constant	Value	Description	Operation
ConnectionRefusedError	1	the remote server refused the connection (the server is not accepting requests)	Retried
RemoteHostClosedError	2	the remote server closed the connection prematurely, before the entire reply was received and processed	Dropped
HostNotFoundError	3	the remote host name was not found (invalid hostname)	Dropped
TimeoutError	4	the connection to the remote server timed out	Retried
OperationCanceledError	5	the operation was canceled via calls to abort or close before it was finished.	Dropped
SslHandshakeFailedError	6	the SSL/TLS handshake failed and the encrypted channel could not be established.	Dropped
TemporaryNetworkFailureError	7	the connection was broken due to disconnection from the network, however the system has initiated roaming to another access point. The request should be resubmitted and will be processed as soon as the connection is re-established.	Retried
NetworkSessionFailedError	8	the connection was broken due to disconnection from the network or failure to start the network	Dropped
BackgroundRequestNotAllowedError	9	the background request is not currently allowed due to platform policy	Dropped

© Softing Industrial Automation GmbH

ProxyConnectionRefusedError	101	the connection to the proxy server was refused (the proxy server is not accepting requests)	Droped
ProxyConnectionClosedError	102	the proxy server closed the connection prematurely, before the entire reply was received and processed	Droped
ProxyNotFoundError	103	the proxy host name was not found (invalid proxy hostname)	Droped
ProxyTimeoutError	104	the connection to the proxy timed out or the proxy did not reply in time to the request sent	Droped
ProxyAuthenticationRequiredError	105	the proxy requires authentication in order to honour the request but did not accept any credentials offered (if any)	Droped
ContentAccessDenied	201	the access to the remote content was denied (similar to HTTP error 403)	Droped
ContentOperationNotPermittedError	202	the operation requested on the remote content is not permitted	Droped
ContentNotFoundError	203	the remote content was not found at the server (similar to HTTP error 404)	Retried
AuthenticationRequiredError	204	the remote server requires authentication to serve the content but the credentials provided were not accepted (if any)	Droped
ContentReSendError	205	the request needed to be sent again, but this failed for example because the upload data could not be read a second time	Retried
ContentConflictError	206	the request could not be completed due to a conflict with the current state of the resource	Droped
ContentGoneError	207	the requested resource is no longer available at the server	Droped
InternalServerError	401	the server encountered an unexpected condition which prevented it from fulfilling the request	Droped
OperationNotImplementedError	402	the server does not support the functionality required to fulfill the request	Droped
ProtocolUnknownError	301	the server is unable to handle the request at this time	Droped
ProtocolInvalidOperationError	302	the requested operation is invalid for this protocol	Droped

UnknownNetworkError	99	an unknown network-related error was detected	Retried
UnknownProxyError	199	an unknown proxy-related error was detected	Retried
UnknownContentError	299	an unknown error related to the remote content was detected	Dropped
ProtocolFailure	399	a breakdown in protocol was detected (parsing error, invalid or unexpected responses, etc.)	Dropped
UnknownServerError	499	an unknown error related to the server response was detected	Dropped

The number of dropped and of pending request can be visible in the system items.

7.3.3.7.2 Connection Wizard

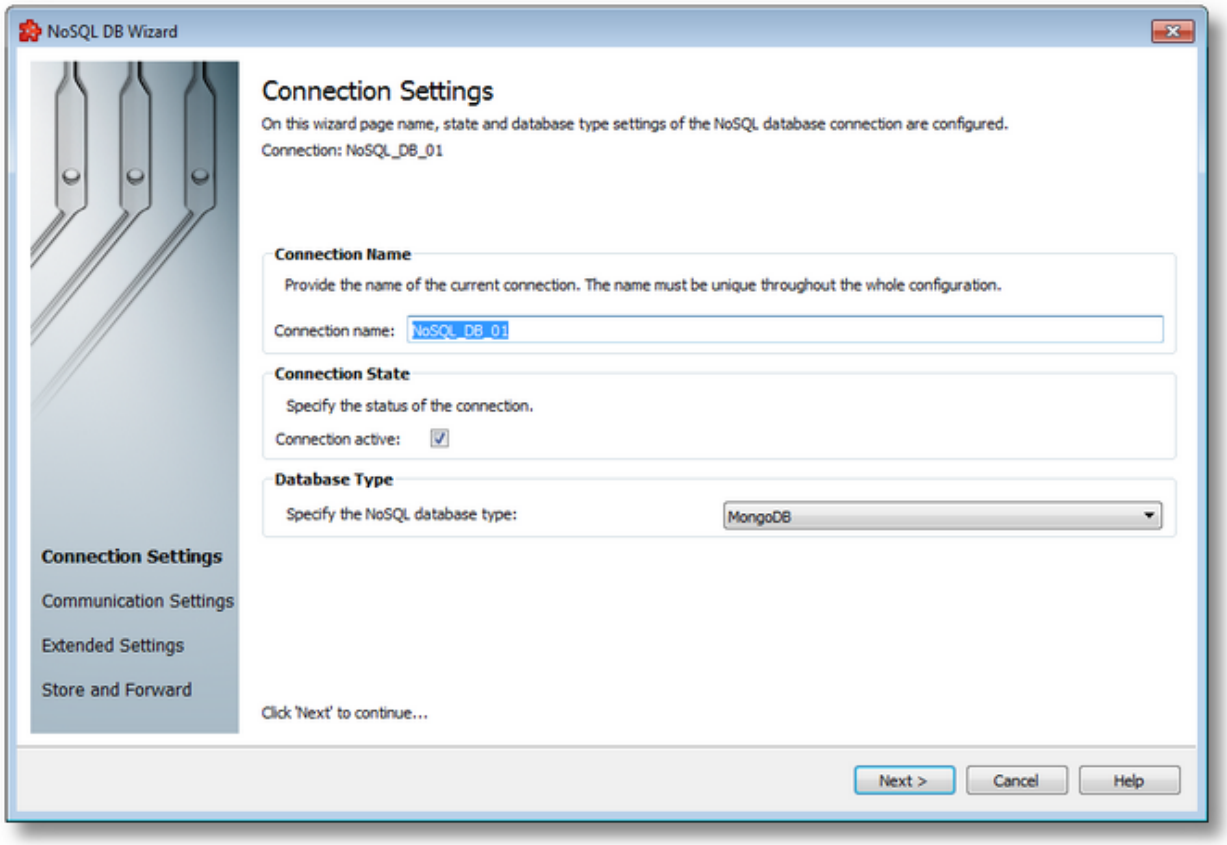
- With the Connection Wizard you can create a new connection definition or modify an existing one:

- Define NoSQL Connection settings @ Connection Wizard - Connection Settings
- Define settings related to communication with server @ Connection Wizard - Comminication Settings
- Define additional settings for the server connection @ Connection Wizard - Extended Settings
- Define store and forward settings on connection @ Connection Wizard - Store and Forward Settings

Note: A wizard flow will not include all the pages listed above. Depending on the selections performed in some of the pages, different flow paths will be followed.

This page allows specifying connection settings such as a unique connection name and state for identifying the connection.

© Softing Industrial Automation GmbH



Connection Name

A unique identifier for the current connection. In case the provided name is already used by another connection, the user is not allowed to proceed to the next wizard page, until they provide a different name.

Connection State

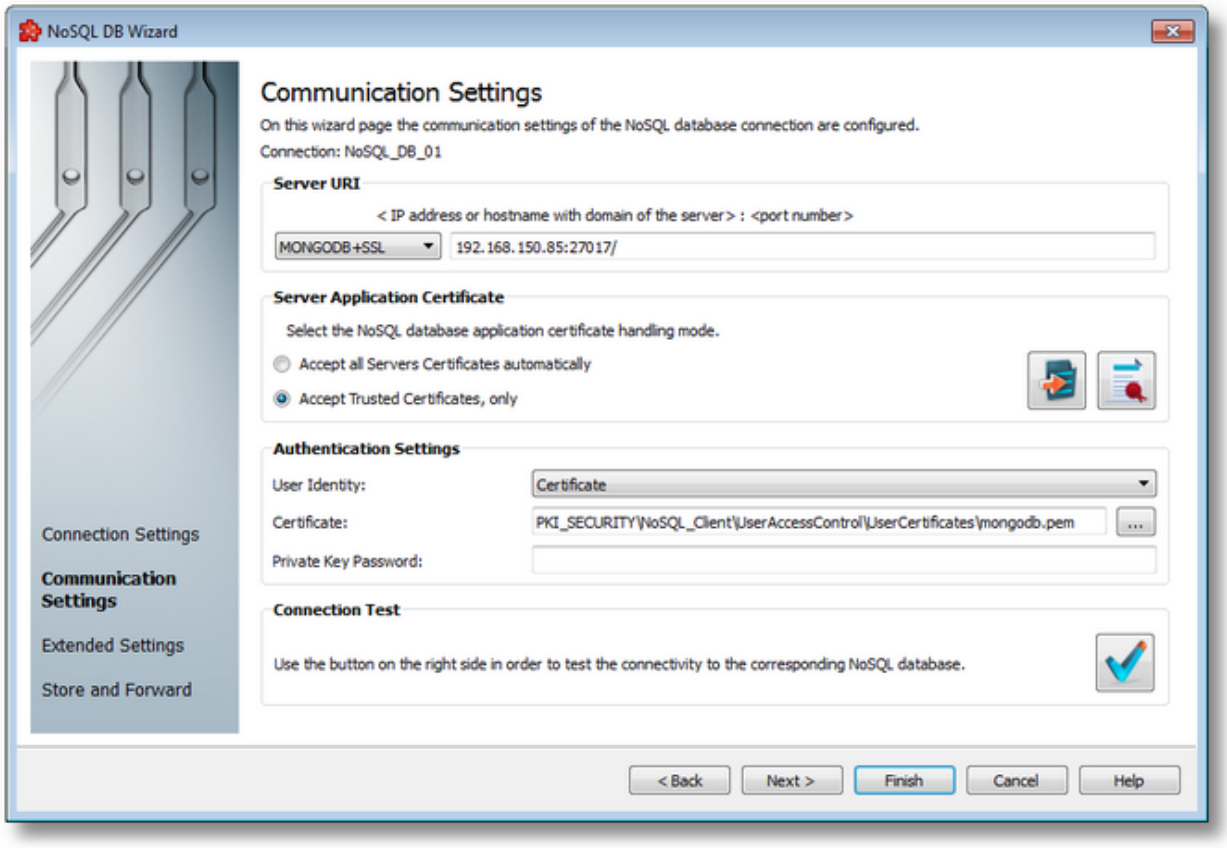
This flag specifies if the current connection will be activated or not.

Database Type

The desired database type can be selected from the list displayed in this section. Currently MongoDb, CouchDb and InfluxDb are supported.

Connection Wizard - Communication Settings >>>

This page allows to specify communications settings such protocol type, host/IP, port number, certificate settings for secure connections and authentication settings.



Server URI

In this section it is possible to specify the protocol type, choosing from TCP, HTTP and HTTPS, MONGODB and MONGODB+SSL. IPv4 and IPv6 are both supported. The port number is mandatory, and the user will be prompted if it is not provided.

The default port numbers per database type are the following:

- MongoDb: 27017
- CouchDb: 5984
- InfluxDb: 8086

NoSQL Server Application Certificate

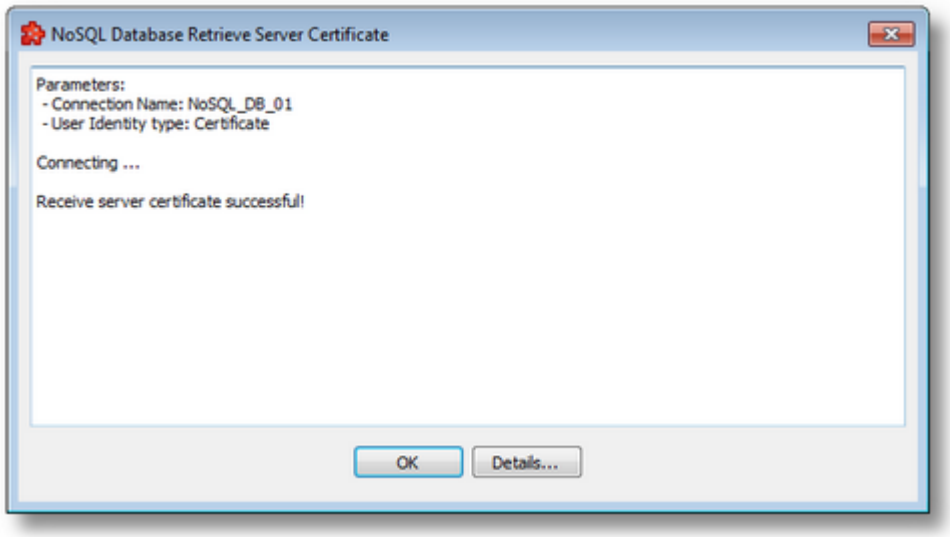
This section is only enabled if the protocol type for the connection is set to HTTPS. In this case there are two additional settings available.

© Softing Industrial Automation GmbH

- *Accept all Server Certificates automatically* - if this option is selected, dataFEED will accept any certificate the server sends

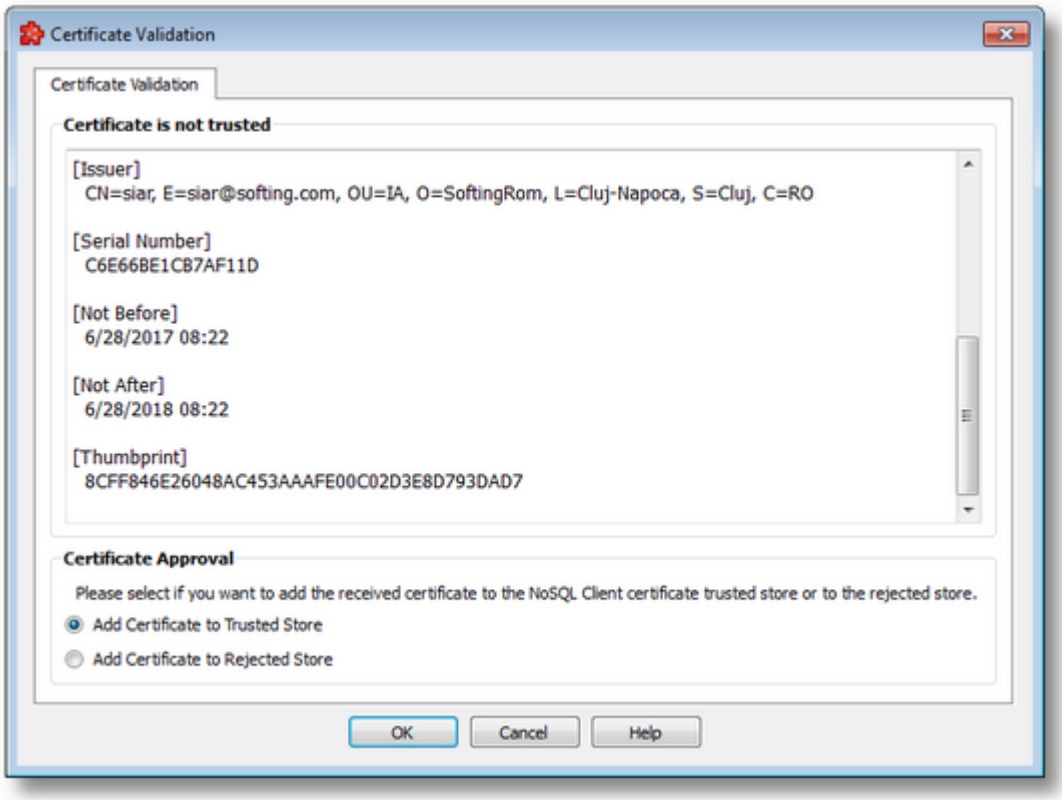
- *Accept Trusted Certificates, only* - in this case, dataFEED needs to have the server certificate in its local certificate store, and will validate the server certificate against the one stored there at a certain interval, when executing actions related to this connection. If the stored certificate is in the trusted store and the certificate sent by the server match, then the actions are executed. If the server certificate is in the rejected store, then no actions of this connection will be executed.

To get the server certificate and save it in the local certificate store the user needs to use the *Receive server certificate* button next to the described settings above.



By pressing the *Details...* button the user can view extra information sent by the server, if available.

By pressing *OK,* the certificate content is presented and the user can choose where to store it in the local certificate store.



To view and manage all the stored certificates use the *Manage Certificates* button, which opens the Certificate Manager.

Authentication Settings

In this section the user can specify the two different authentication modes for TCP, HTTP and MONGODB, and three for HTTPS and MONGODB+SSL.

TCP & HTTP:

- Anonymous - no information is given with this option
- User name and password

HTTPS:

- All of the above
- Certificates - in this case a separate certificate and private key file can be provided as well as a private key password. The certificate file is mandatory, the other two are not. In some cases the certificate file may contain the private key as well.

Note: Because of the version of Qt used (5.2.1), dataFEED currently only supports certificates in .pem and .der formats. No other certificate file formats can be used and conversions need to be made by the user manually, for example using the openSSL tool, downloaded separately.

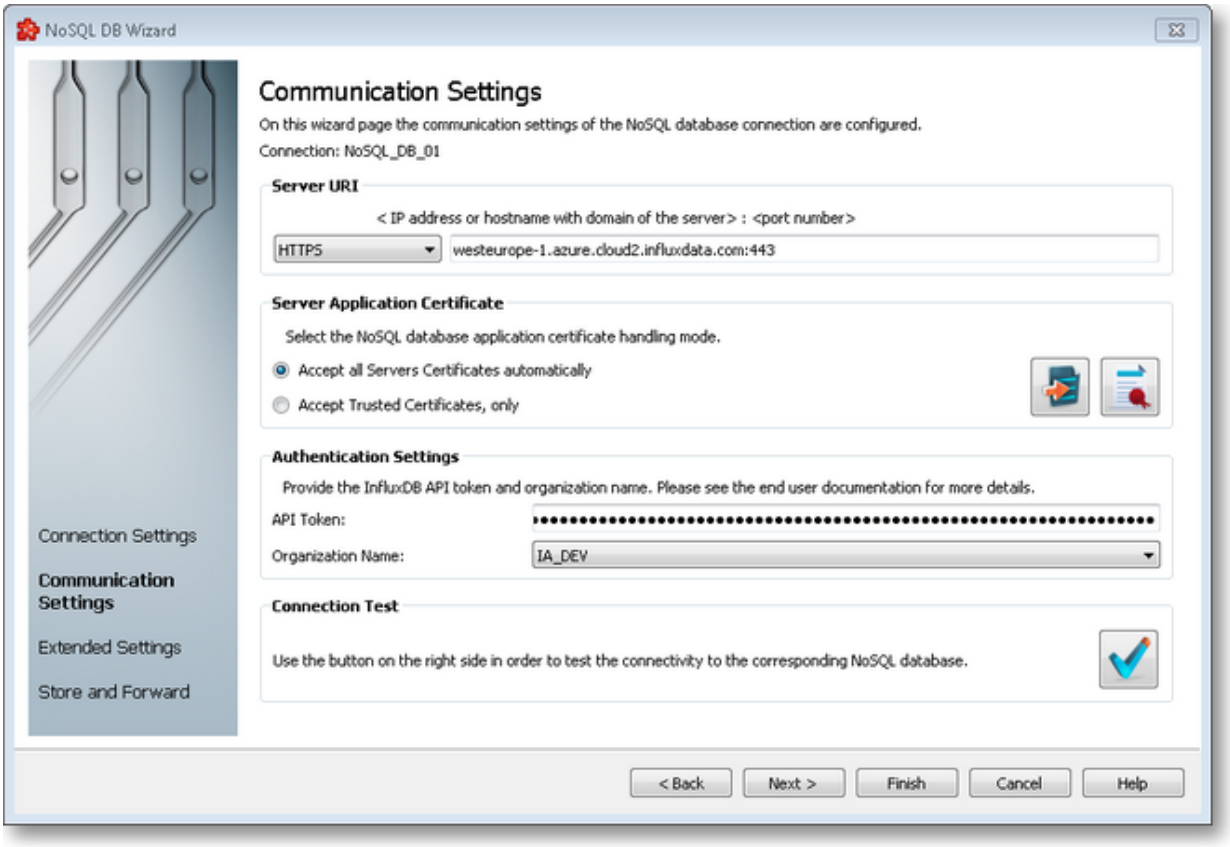
© Softing Industrial Automation GmbH

Note: MongoDb only requires and accepts one single certificate file which must contain the private key as well. They cannot be specified separately. In case of CouchDb, on the first image, both certificate and private key fields will be available.

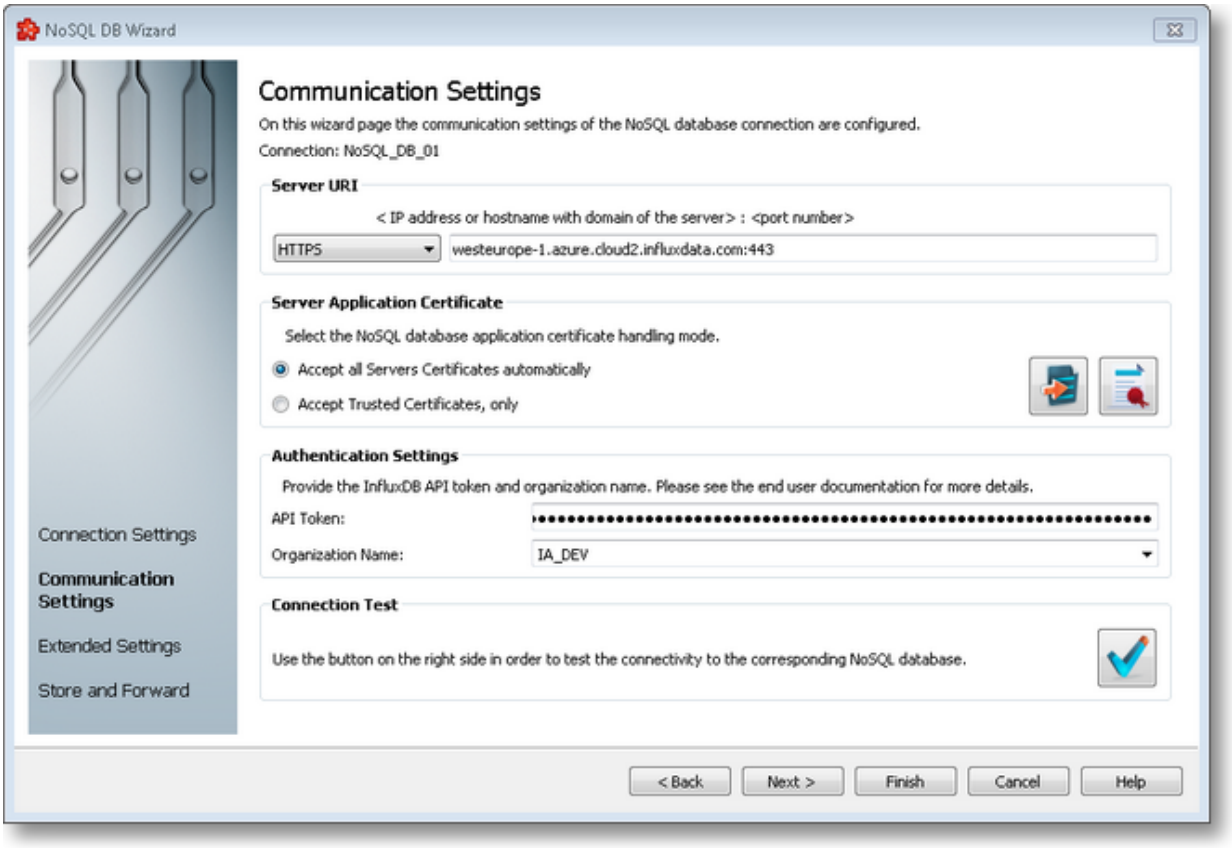
Additional InfluxDB Settings

the InfluxDB communication settings need to be configured differently: it requires in addition the API Token and the Organization parameter.

The InfluxDB v2 API requires a Token authentication scheme to authenticate to the InfluxDB API.



The list of organizations is retrieved at wizard page initialization. If retrieving the list of organizations is successful, the user needs to select the Organization Name from the combobox provided values, as shown in the image above.

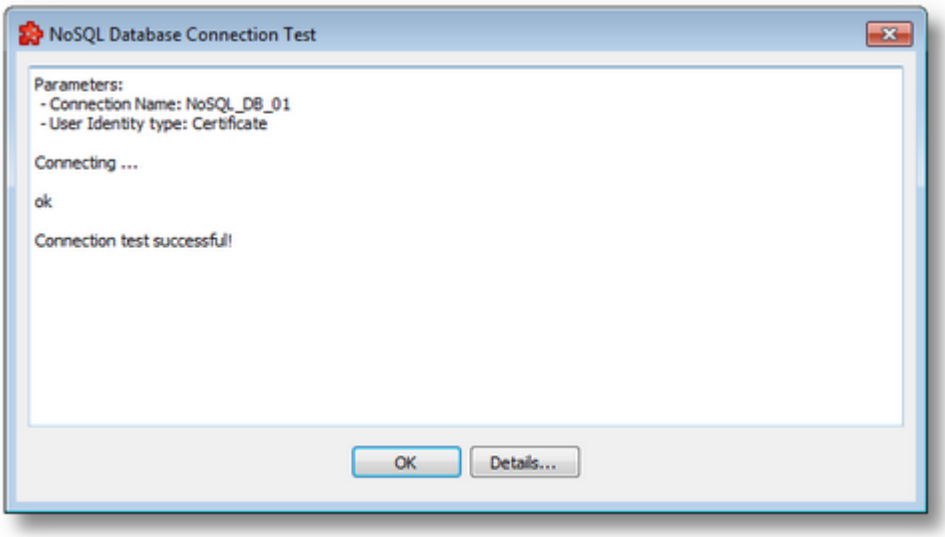


If the list of organizations can't be retrieved, it is the responsibility of the user to write in the combobox editable field an existing Organization Name from the InfluxDB database, as shown in the image above.

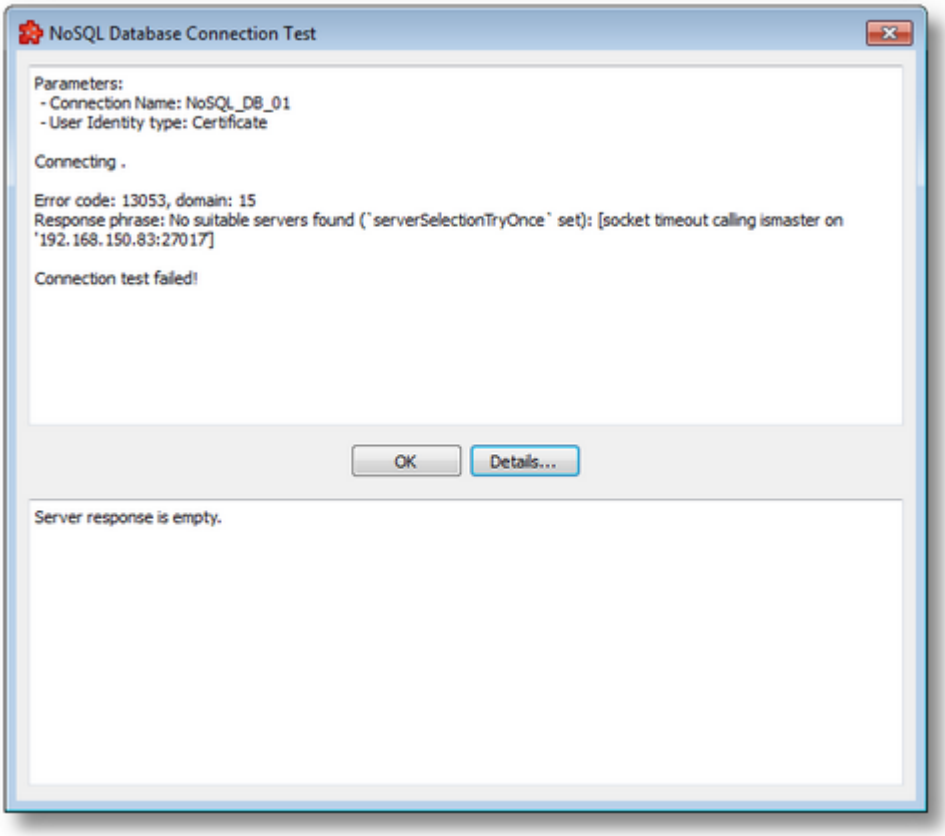
Test Connection

Finally, the *Connection Test* button can be used to check if the configured information lead to a successful connection with the server. If the test is successful actions will be executed successfully (if configured correctly themselves), otherwise the user needs to tweak these settings or check the server availability and configuration.

© Softing Industrial Automation GmbH



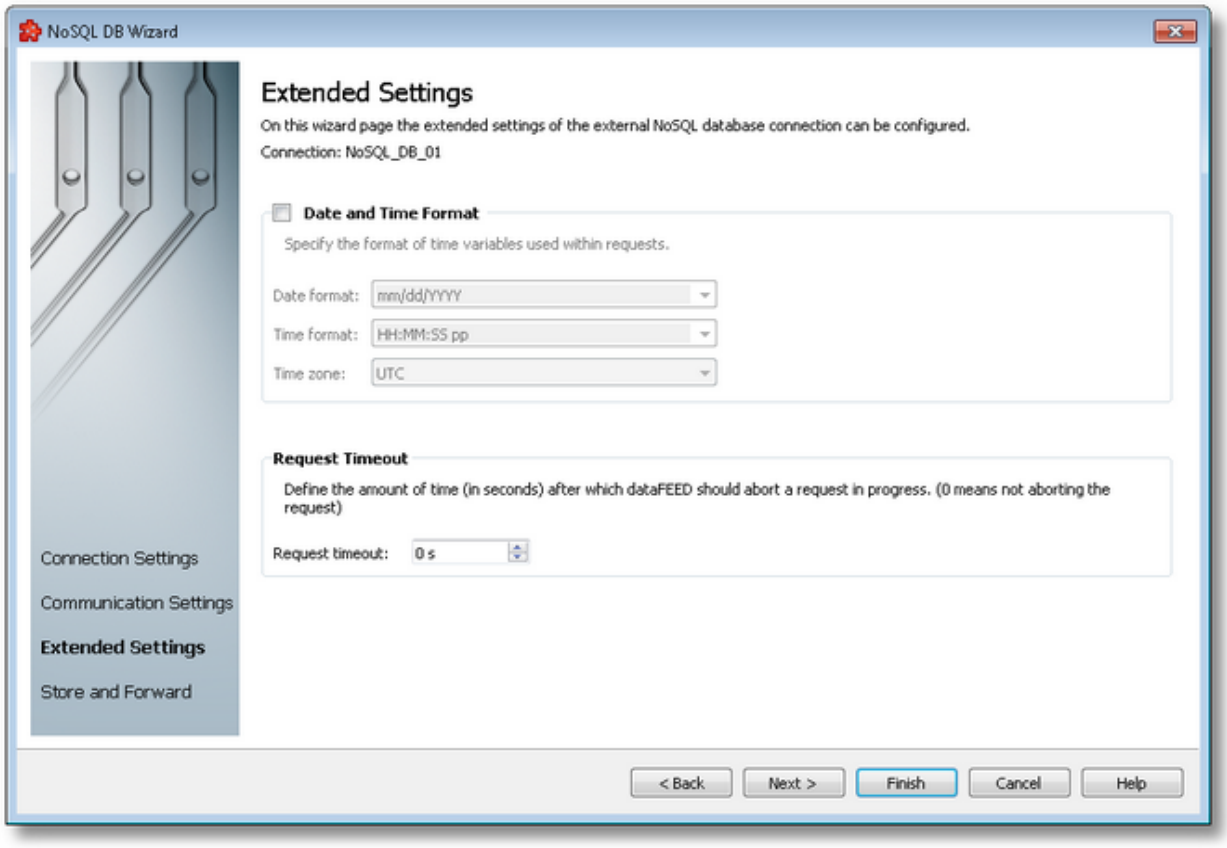
A successful connection test looks like above.



In any case, after pressing the button, a dialog will be displayed either prompting the user that the connection is successful, or showing error information, like in the image above.

<<< Connection Wizard - Connection Settings Connection Wizard - Extended Settings >>>

This page allows to specify the extended connection settings such as time and date format and request timeout.



Date and Time Format

This section handles the time and date formats that are to be used when the timestamp placeholders are expanded and their values written to database actions. The default values are `mm/dd/YYYY` for the *Date format and* `HH:MM:SS pp` for the *Time format*. Other defined values for Time and Date format are:

- *Date format*: `mm/dd/YYYY`, `dd-mm-YYYY`, `dd.mm.YYYY`, `dd/mm/YYYY`, `YYYY-mm-dd`, `mm-dd-YYYY`
- *Time format*: `HH:MM:SS pp`, `HH:MM:SS.FFF pp`, `HH:MM:SS.FFF`, `HH:MM:SS`

Values for Time and Date format can also be defined manually but in this case special care must be taken to ensure that the definitions are recognized by the local dataFEED OPC Suite application.

In addition to the default values listed in the format selection combo boxes, the user can add his/her own custom format. Custom Date and Time format strings are accepted as valid, when the format strings contain in their composition the following accepted format substrings and format delimiters. These formats allow the inclusion of specific locale information for the regional settings.

© Softing Industrial Automation GmbH

- *Valid Date format strings*: `AAAAAA, aaa, BBBBBBBB, bbb, ccccccccccccccccccc, dd, jjj, mm, UU, ww, xxxxxxxxxx, yy, YYYY`
- *Valid Time format strings*: `II, HH, MM, SS, pp, XXXXXXXX, FFFFFF, FFFFF, FFFF, FFF ,FF, F`
- *Valid Date and Time format delimiters*: `<dot>, <colon>, <slash>, <comma>, <backslash>, <vertical bar>, <semicolon>, <space>, <tab>, <minus>`

Time Zone handles the source localization time: UTC (Universal Time Coordinated) or local time (local time localization). The default time zone is UTC.

When the *Time, Date Format and Time Zone* option is not selected the default values are used. If they are modified, they apply to all actions of the currently selected REST connection.

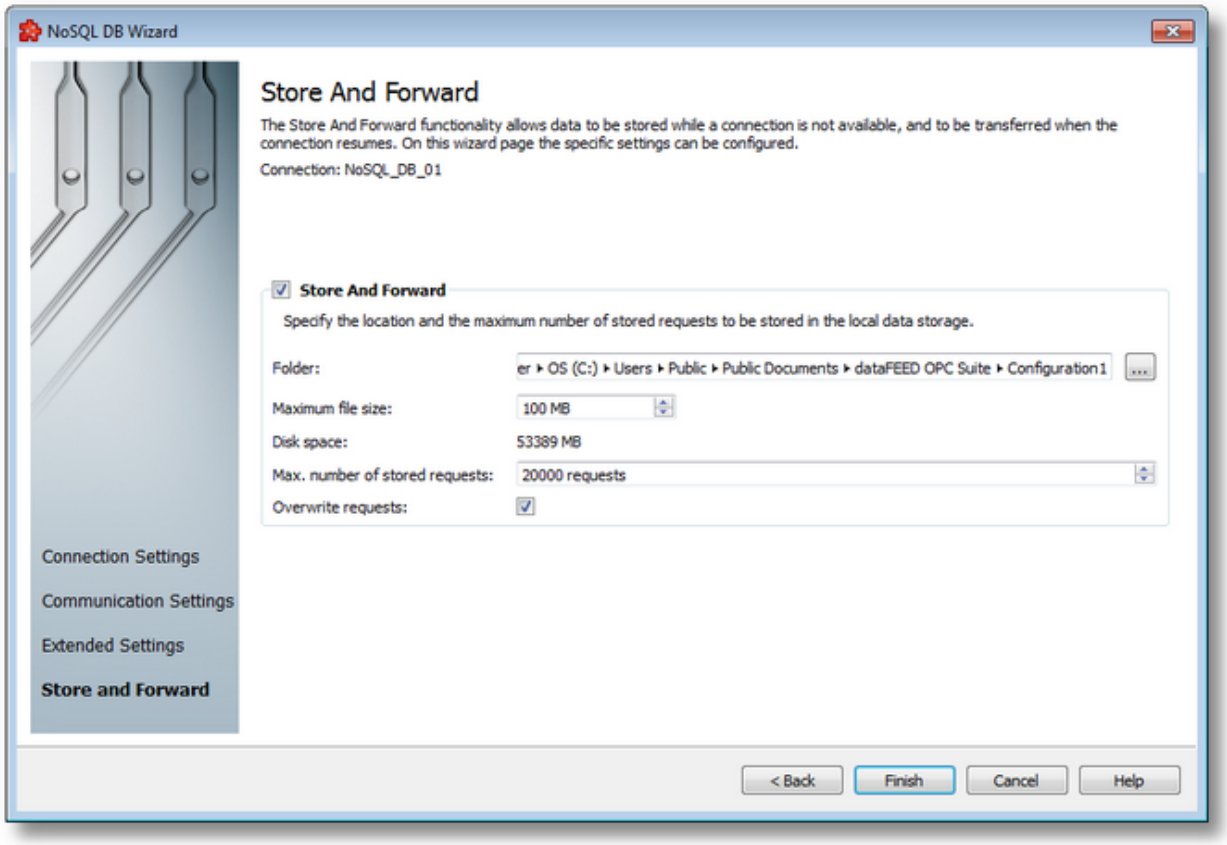
Request Timeout

This value, expressed in seconds, is used when a request takes a long time to complete. If the user specifies a non-zero value, it means that dataFEED OPC Suite will wait for that amount of time before aborting the request. If the value is zero, then no aborting will be done, dataFEED OPC Suite will wait for the request to complete successfully or with some kind of error.

Default is 0.

<<< Connection Wizard - Communication Settings Connection Wizard - Store and Forward Settings >>>

This page allows to specify the local storage settings in case that the server is unavailable.



This functionality can be turned on and off by using the *Store And Forward* checkbox. If it is off (default), it instructs dataFEED OPC Suite to drop any requests that are executed after the internal request queue is filled. If it is on, the application will store the requests lining up after the queue fills up.

The options that can be configured on this page and which change the mechanism's behavior are the following:

Folder

The path where the storage file should be located. If the path does not point to an existing folder then the corresponding path will be created by the dataFEED OPC Suite Configurator when the Next button is pressed, and user confirms.

The file is created at the run time of the local dataFEED OPC Suite application. If the file exists it will be overwritten. The name of the connection will be the name of the file as well, with .bin extension.

Maximum file size

Maximum file size is used to limit the space used on the hard drive. Minimum is 1 MB and maximum is 500 MB.

© Softing Industrial Automation GmbH

When the file capacity is reached requests will be dropped. This behavior is influenced by the *Overwrite requests* option described below.

Maximum number of stored requests

Maximum number of requests stored used to limit the requests stored in the file. Default is 20000.

Note: Requests are dropped starting either when the file size limit or the requests limit is reached first.

Overwrite requests

Whether to overwrite previously stored requests when the above specified number has been reached.

If the checkbox is not checked then no more data changes will be stored when the maximum number of requests has been reached (this allows access to the first stored requests). If the checkbox is checked the oldest request will be overwritten with the newest one when the maximum number of requests has been reached (this allows access to the latest stored requests).

<<< Connection Wizard - Extended Settings

7.3.3.7.3 Action Wizard

- With the NoSQL Action Wizard you can create a new NoSQL Action definition or modify an existing one:

  - Obtain general information about the NoSQL Action Wizard @ NoSQL Action Wizard - Configuring NoSQL Action
  - Choose the action trigger type @ NoSQL Action Wizard - Trigger Type
  - Pick the item trigger @ NoSQL Action Wizard - Trigger Item
  - Select the timer trigger @ NoSQL Action Wizard - Timer Selection
  - Define the timer identity @ NoSQL Action Wizard - Timer Identity
  - Choose the timer type @ NoSQL Action Wizard - Timer Type

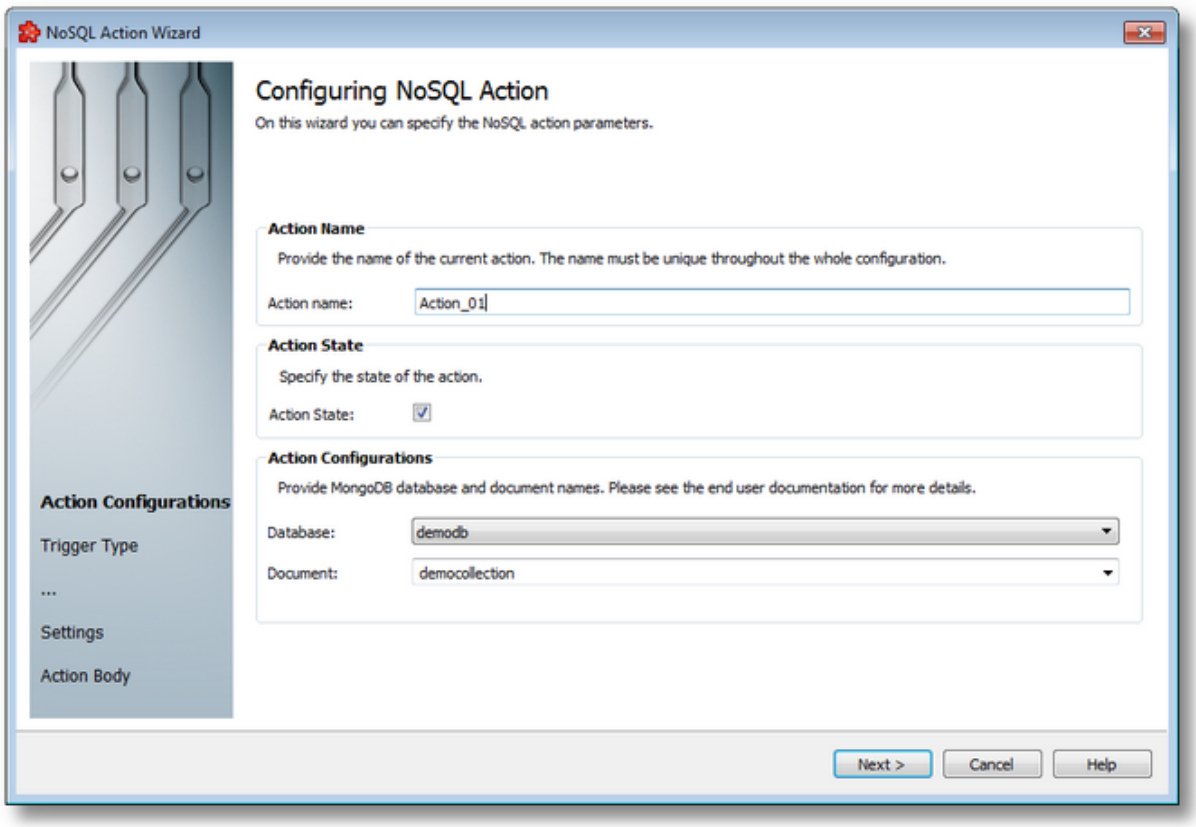
- Adjust schedule timer settings @ NoSQL Action Wizard - Schedule Timer Settings
- Set the timer interval @ NoSQL Action Wizard - Interval Timer Settings
- Configure the execution condition and the update rate @ NoSQL Action Wizard - Action Settings
- Define the header for the current action @ NoSQL Action Wizard - Action Header
- Define the body for the current action @ NoSQL Action Wizard - Action Definition

Note: A wizard flow will not include all the pages listed above. Depending on the selections performed in some of the pages, different flow paths will be followed.

The first wizard page provides the first set of action properties to complete.

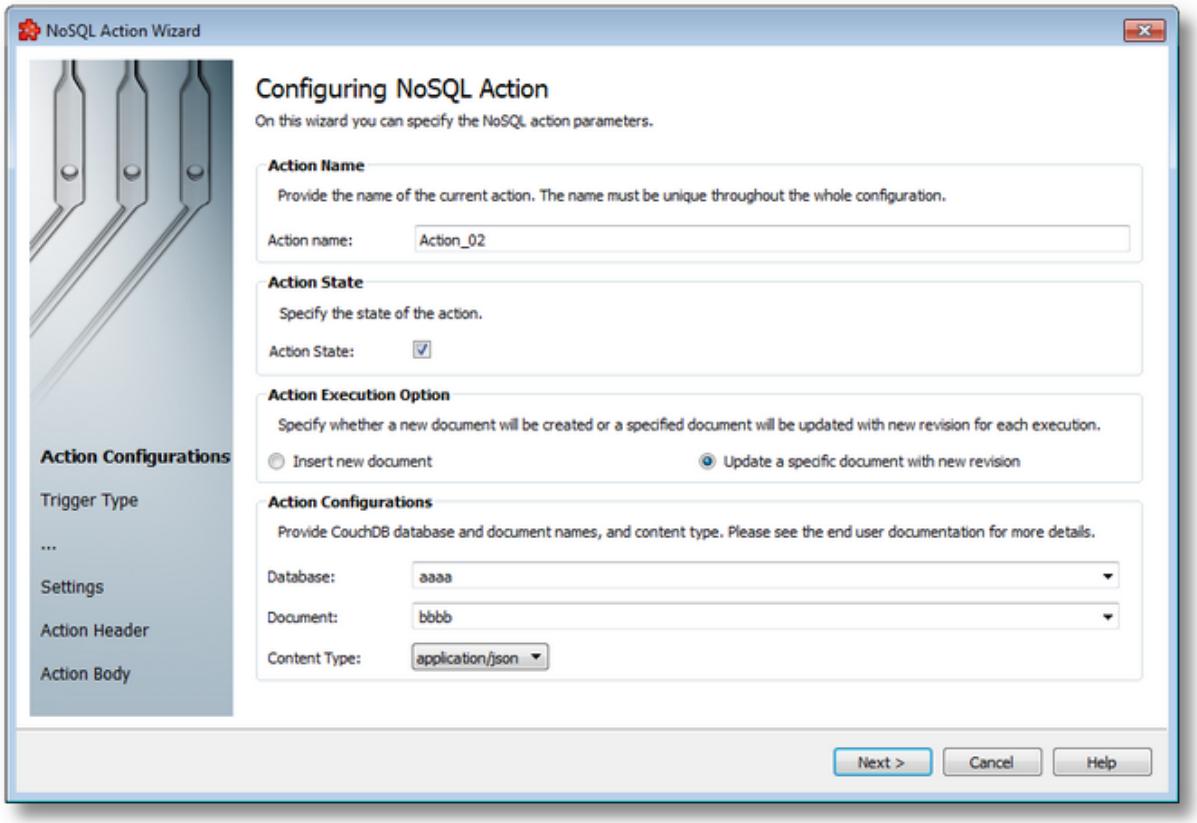
MongoDb and CouchDb actions need to be configured differently: MongoDb only requires the database and document name; CouchDb expects the operation on the document as well (insert or update), and content type.

MongoDb Action page:

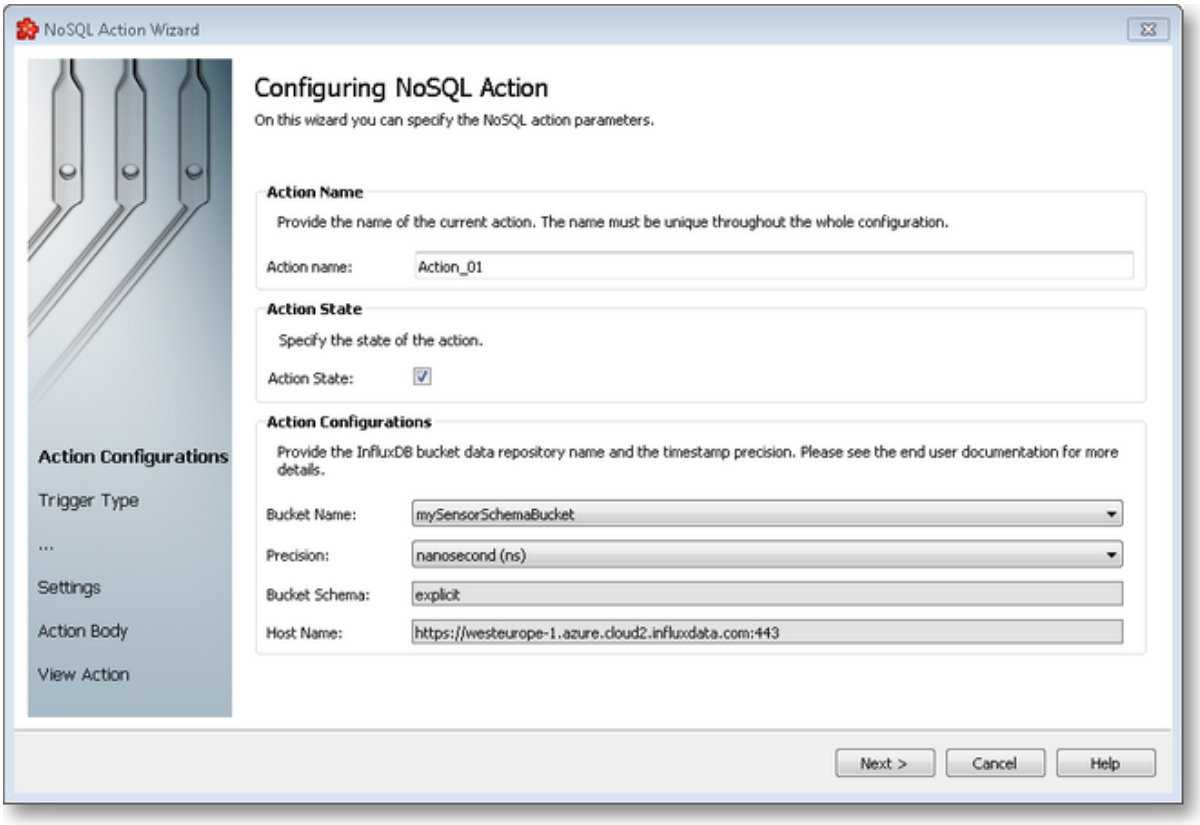


© Softing Industrial Automation GmbH

CouchDb Action page:



InfluxDb Action page:



Action Name

A unique identifier for the current action. In case the provided name is already used by another action, the user is not allowed to proceed to the next wizard page, until they provide a different name.

Action State

This flag specifies whether or not the current connection will be executed.

Action Execution Option

For CouchDb the operation type must be specified. If inserting a new document is chosen, only the *Database* field will be visible and document creation will be left to CouchDb. If updating is chosen like in the image above, all three fields need to be specified.

Database name is read-only, meaning that it cannot be edited by the user, while the document name can be either selected from the list or a new name can be specified. In the latter case, the document will be created when the action will be executed for the first time.

Action Configurations

© Softing Industrial Automation GmbH

- Database - the name of the database
- Document - the name of the document
- Content Type - the content type of the action body is selected from this list of options: application/json for JSON format, application/xml, application/xhtml+xml and text/html for XML format, and text/plain for plain text format without any validation in the application.

For JSON and XML formats the Configurator offers validation of the content, and reports errors to the user if for some reason the content is not in a valid form. For plain text cases, the user has more flexibility in creating the body content in the action, but no validation can be done.

InfluxDb specific action configurations are provided with the following parameters. Please see the end user documentation for more details.

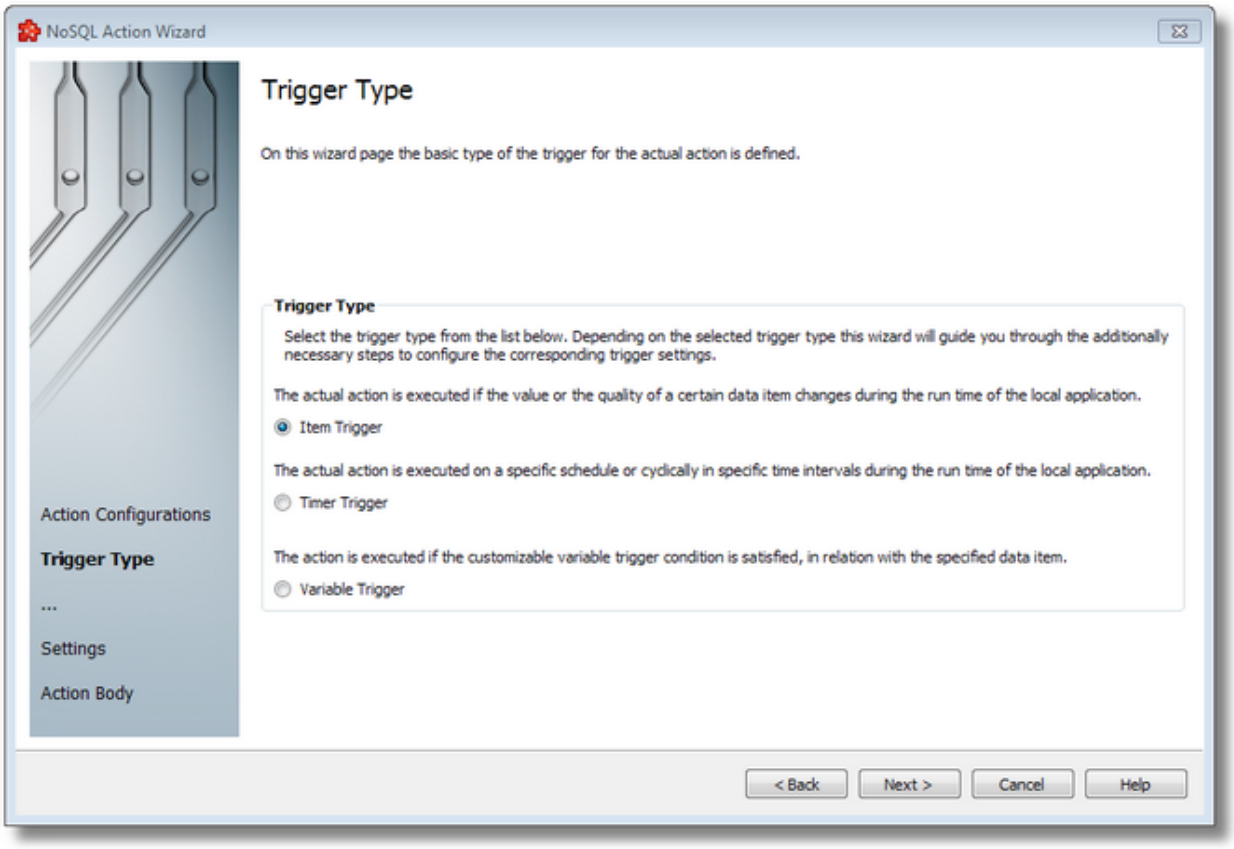
- Bucket Name - the name of the bucket data repository (InfluxDb database existing bucket name)
- Precision - the precision of the timestamp
- Bucket Schema - the type of the bucket schema (read-only)
- Host name - the InfluxDB database host name (read-only)

Note: Conversions between these content types are also possible, if the user changes the content type and then goes through the wizard before saving.

NoSQL Action Wizard - Trigger Type >>>

The Trigger Type wizard page provides the possibility to select the trigger type for the actual action. The following trigger types are available:

- *Item Trigger* - This trigger is fired when a data change for the selected item occurred. A data change means that the value or the quality of the item changed.
- *Timer Trigger* - This trigger is fired at a specified time or interval.
- *Variable Trigger* - This trigger is fired when a data change for the items in the variable conditions occurs. A data change means that the value or the quality of the item changed.



To proceed with the wizard and continue to define the selected type of trigger (or to advance to the last wizard page when the *Data Change* trigger type is selected) use the *Next* button, to go back to the previous wizard page use the *Back* button, or discard the wizard by pressing the *Cancel* button.

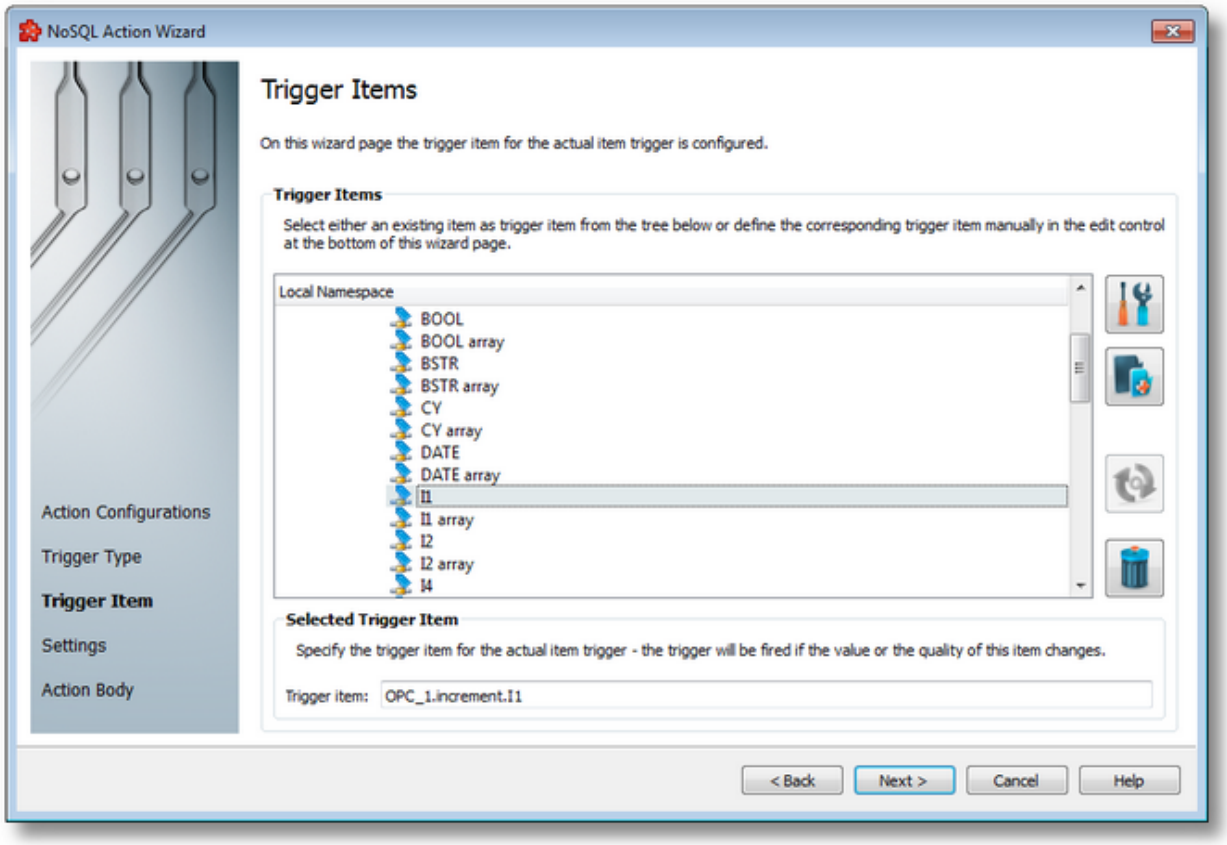
Depending on the selected *Trigger Type* the next wizard page will be:

- NoSQL Action Wizard - Trigger Item - if the selected trigger type is *Item Trigger*,
- NoSQL Action Wizard - Timer Selection - if the selected trigger type is *Timer Trigger*
- Variable Trigger Wizard - Trigger Selection - if the selected trigger type is *Variable Trigger*

<<< NoSQL Action Wizard - Configuring NoSQL Action NoSQL Action Wizard - Trigger Item >>>

The *Trigger Item* wizard page is shown if the selected trigger type is *Item Trigger*. Here you can configure the actual trigger item.

© Softing Industrial Automation GmbH



This wizard page contains two separate sections:

- *Trigger Items* section, which allows selecting an item trigger or managing the Local Namespace tree.
- *Selected Trigger Item* section, which shows the item ID of the selected trigger item and also allows typing an item ID, thereby selecting this item in the above list.

Trigger Items section

The entire namespace of the current configuration is shown as a namespace tree. The name of the root node is composed by appending the instance name to the *Local Namespace* keyword. The first level of the namespace contains:

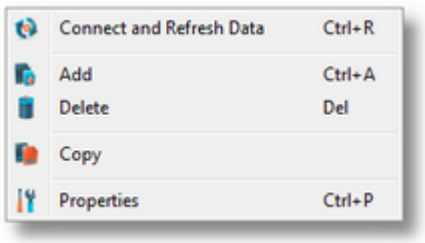
- A node for each server connection, named by the server connection items prefix.
- A node for System Items.
- A node for Local Items.

The local namespace tree offers the possibility to explore the entire namespace of the local dataFEED OPC Suite application in order to select a trigger item. After the selection is made, the *Trigger item* edit box is populated with the selected item ID.

The entire namespace of the local dataFEED OPC Suite application can be managed through this wizard page by performing certain operations using the buttons on the right, the context sensitive menu or the keyboard shortcuts:

- Open the item properties dialog using the *Properties of the selected item* button  , pressing the keyboard shortcut CTRL+P or selecting the *Properties* action from the context menu. Depending on the type of the selected item, a special Item Properties dialog opens:
    - If the selected item is under Local Items the Modify Local Item dialog is opened.
    - If the selected item is under a dynamic server connection the Modify Dynamic Server Item dialog is opened.
    - If the selected item is under a static server connection the Modify Static Server Item dialog is opened.
- Add a new item using the *Add a new item*  button, pressing the keyboard shortcut CTRL+A or selecting the *Add* action from the context menu. Depending on the type of the selected node, a special Add Item dialog opens:
    - If the selected node is under Local Items the Add Local Item dialog is opened.
    - If the selected node is under a dynamic server connection the Add Dynamic Server Item dialog is opened.
    - If the selected node is under a static server connection the Add Static Server Item dialog is opened.
- Copy the selected item to clipboard by selecting the *Copy* action from the context menu.
- Paste the previously copied item from clipboard by selecting the *Paste* action from the context menu. Items originating from the Local Items cannot be pasted under server connections, and vice-versa; an error message will be displayed in both cases. If an item is pasted in a static (filtered) server connection and the item does not exist on the server an error message is displayed.
- Connect to the server and refresh the data items by using the *Refresh data items*  button, pressing the keyboard shortcut CTRL+R or selecting the *Connect and Refresh Data* action from the context menu. The namespace of the selected server connection is recreated; if manually added items not referenced in the actual configuration exist, those items are deleted from the namespace.
- Delete the selected item and all of its child items by using the Delete data items button  , pressing the DELETE / DEL key or selecting the Delete action from the context menu. Items which are included in any Data Processing / Data Destination add-ons will not be deleted but an error message will be displayed instead.

An example of the context menu can be seen below:



© Softing Industrial Automation GmbH

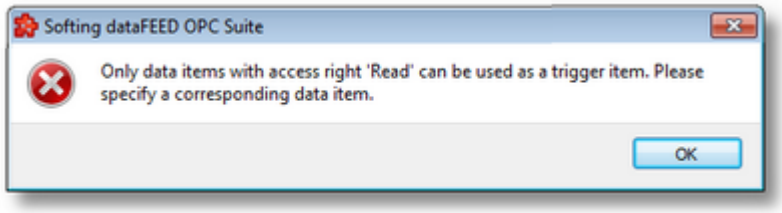
Selected Trigger Item section

The edit box in this section is populated with the item ID after an item from the Local Namespace tree is selected.

You can also manually type the ID of the item in the *Trigger item* edit box and thus have the above tree automatically select a valid server item.

Note: If the item ID is typed and the specified item exists in the Local Namespace, it will be used as the trigger item. The item will only be created if it could not be localized in the Local Namespace. In that case the newly created item will be used as the trigger item.

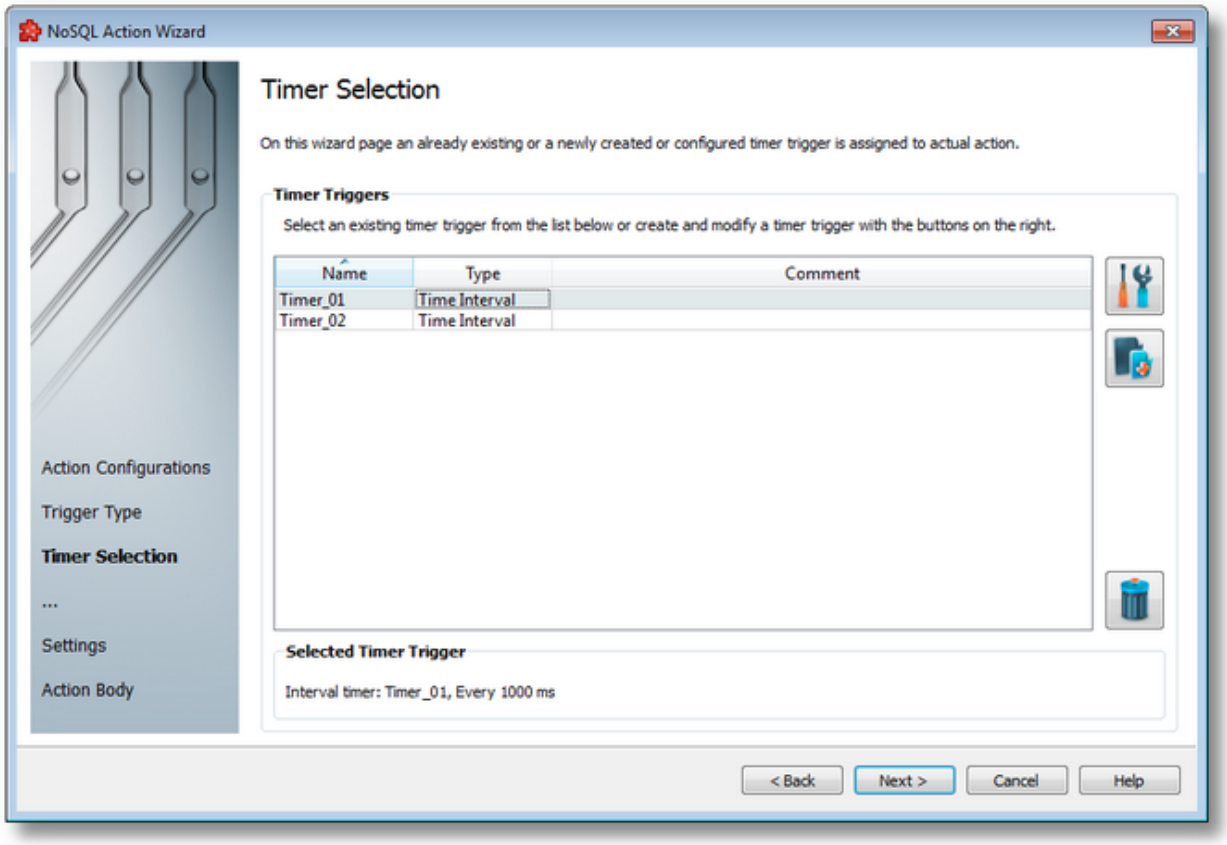
Note: If the item corresponding to the typed item ID could not be created, an error message will be displayed requesting the input of a new, valid item.



Attention: If the item corresponding to the typed item ID is valid and resides under a static server connection, the respective item will be created in the filter file definition.

<<< NoSQL Action Wizard - Trigger Type NoSQL Action Wizard - Action Settings >>>

The Timer Selection wizard page is shown only if the selected trigger type is *Timer Trigger*. Here you can configure the actual timer trigger.

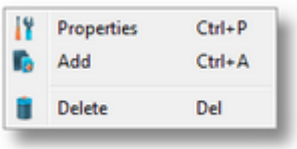


The current page allows selecting a previously defined timer trigger and managing the timer list by using the buttons on the right, the context sensitive menu or the keyboard shortcuts:

- Change the properties of the selected timer by using the *Properties* button  , pressing the keyboard shortcut CTRL+P or selecting the *Properties* action from the context menu.
- Add a new timer by using the Add timer trigger button  , pressing the keyboard shortcut CTRL+A or selecting the Add action from the context menu.
- Delete the selected timer trigger by using the Delete selected timer trigger button  , pressing the DELETE / DEL key or selecting the Delete action from the context menu.

Note: If a timer trigger has associated actions it cannot be deleted.

An example of the context sensitive menu can be seen below:



Note: Information about the selected timer is shown in the *Selected Timer Trigger* area at the bottom of the page.

© Softing Industrial Automation GmbH

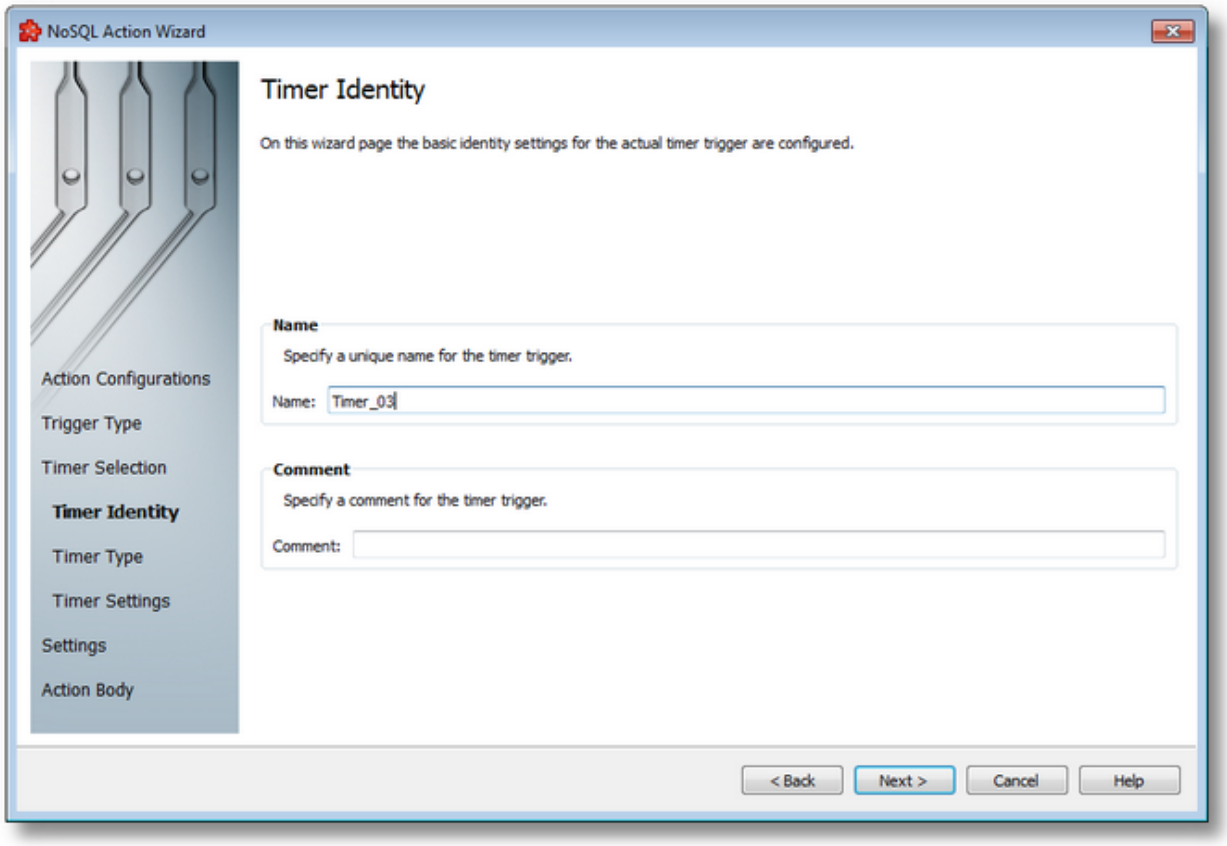
A sub-wizard that allows adding or modifying timer triggers can be launched by selecting to change an existing timer trigger or to add a new one. The first page of this sub-wizard is NoSQL Action Wizard - Timer Identity.

To proceed with the *NoSQL Action Wizard* and define the action settings press the *Next* button. To return to the previous wizard page use the *Back* button, or exit the wizard by clicking the *Cancel* button.

The wizard will continue straight to the *Action Settings* page as an already defined timer is selected.

<<< NoSQL Action Wizard - Trigger Type NoSQL Action Wizard - Action Settings >>>

The Timer Identity wizard page provides the possibility to assign a name and a comment to the timer trigger.

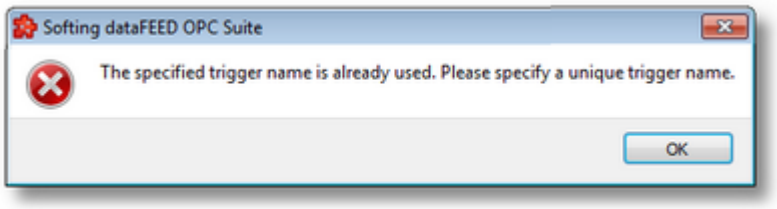


Note: When adding a new timer trigger, the timer name will be automatically generated according to the format: *Timer_number*, where the *number* part is an auto-incremented integer.

When editing an existing timer, the timer's defined name and comment will be filled in the two edit boxes allowing you to modify these attributes.

Attention: The name of the *Timer Trigger* must be unique among all triggers and cannot be empty.

If the newly added timer trigger has the same name as an existing trigger, an error message box will be displayed and a new trigger name must be provided:



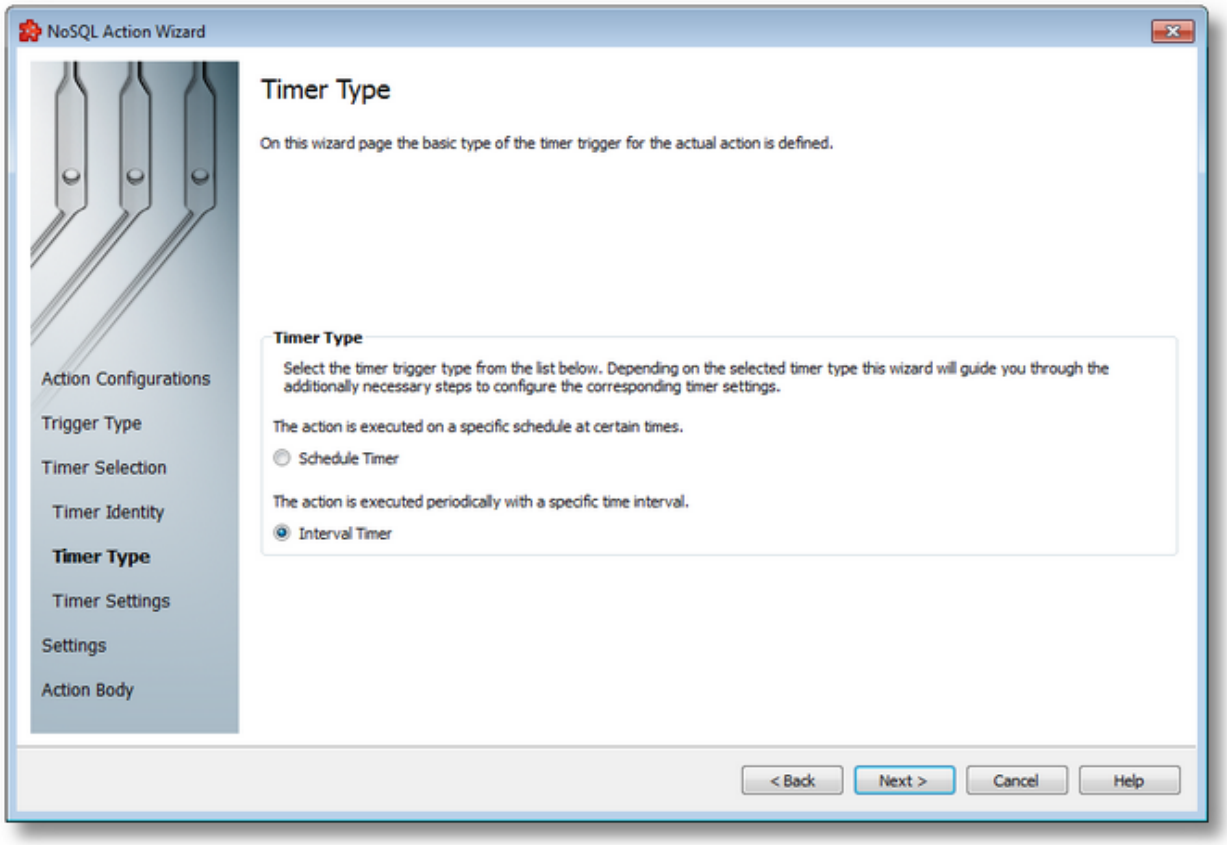
The *Comment* field of the *Timer Identity* wizard page is optional; a Timer Trigger can also be added without any comment.

<<< NoSQL Action Wizard - Timer Selection NoSQL Action Wizard - Timer Type >>>

This wizard page allows choosing the type of the timer trigger. The following types of timer are available:

- *Interval Timer*
- *Schedule Timer*

© Softing Industrial Automation GmbH



Interval Timer

The Interval Timer type allows defining a specific interval of time (in milliseconds) between the timer fire events. The time interval can be configured on the NoSQL Action Wizard - Interval Timer Settings wizard page.

Schedule Timer

The Schedule Timer type allows defining a specific date and time when the timer should fire. The schedule can be configured on the NoSQL Action Wizard - Schedule Timer Settings wizard page.

Note: The *Timer Type* of an existing *Timer Trigger* cannot be changed.

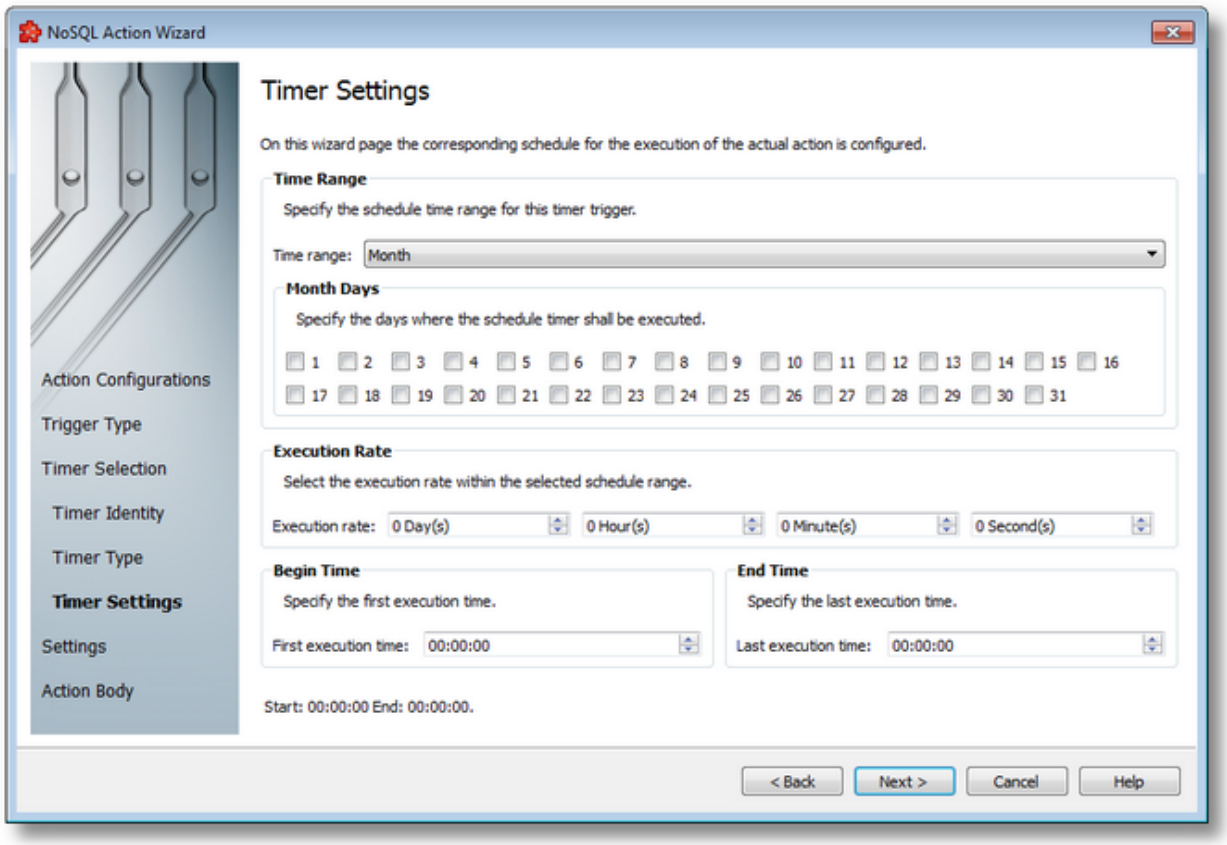
To continue defining the specific timer properties press the *Next* button, to return to the *Timer Identity* wizard page use the *Back* button, or exit this sub-wizard (thereby aborting the *Action Wizard*) by pressing the *Cancel* button.

The description of the Action Wizard will continue with the wizard pages for configuring a *Schedule Timer*.

<<< NoSQL Action Wizard - Timer Identity Settings

NoSQL Action Wizard - Schedule Timer >>>

The *Schedule Timer Settings* wizard page is a more complex page compared to the *Interval Timer Settings* wizard page and allows defining the corresponding schedule for the execution of the trigger action:



The page has four sections which identify the schedule:

- *Time Range*: Provides the possibility to choose one of the following schedule ranges: *month*, *week*, *day*, *hour* or *minute*. When the *month* or the *week* ranges are selected the panel below the selection is populated with the month and week days, respectively.
- *Execution Rate*: Sets the execution rate of the current action in the previously-defined *Time Range* (e.g. every X day(s), every Y hour(s), every Z minute(s), every T second(s)).
- *Begin Time*: Sets the time the current action should begin execution.
- *End Time*: Sets the time the current action should stop execution.

According to the *Time Range* the *Begin Time* and the *End Time* fields will allow typing:

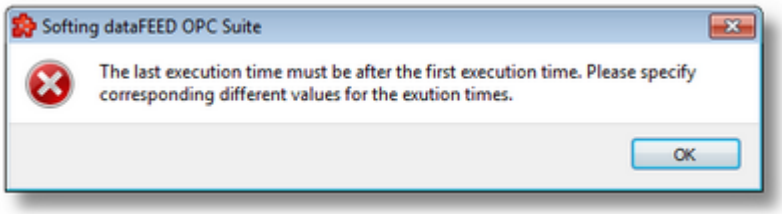
- Seconds (SS) if the *Time range* is *Minute*.
- Minutes and seconds (MM:SS) if the *Time range* is *Hour*.

© Softing Industrial Automation GmbH

- Hours, minutes and seconds (HH:MM:SS) if the *Time range* is *Day* or greater (*Week*, *Month*).

Note: A short review of the configured timer schedule is shown at the bottom of this wizard page.

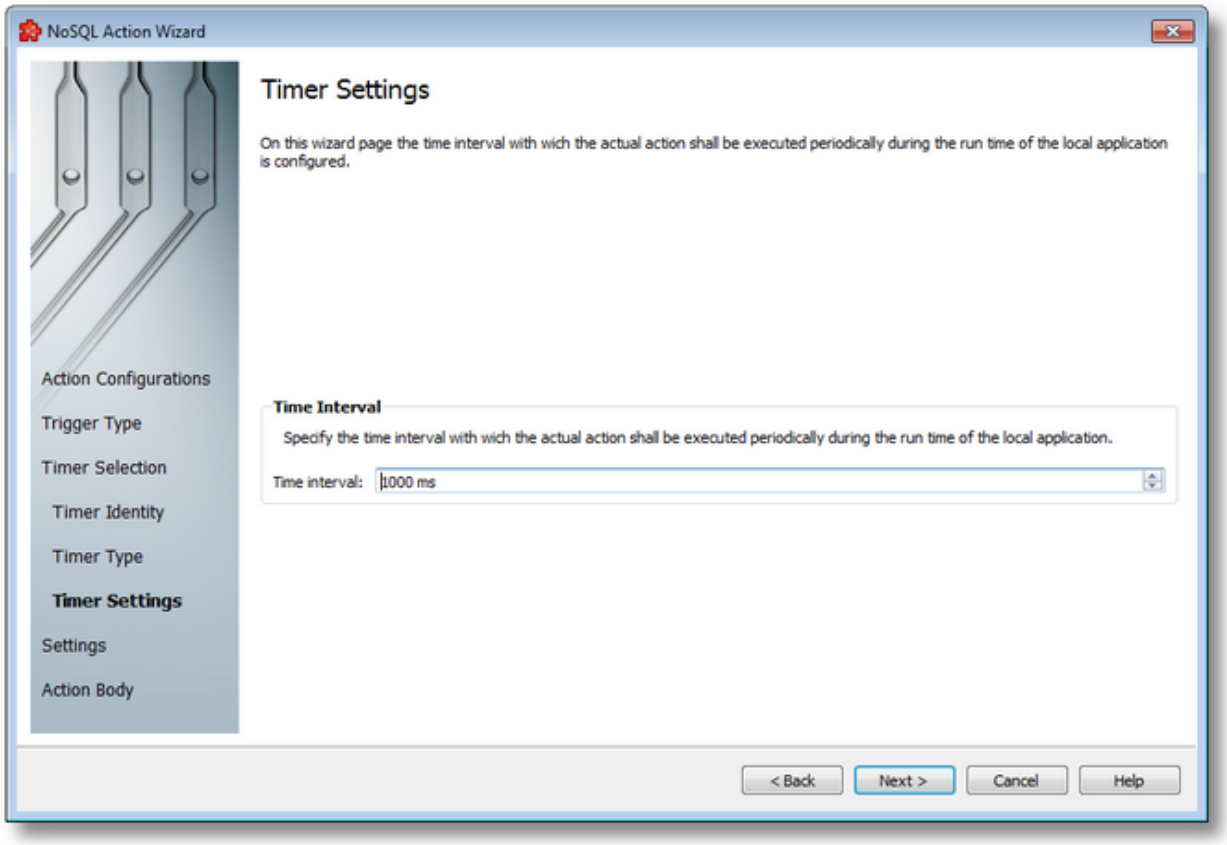
If the defined schedule is not valid, an error message dialog will appear naming the error that occurred.



Continue to the *Timer Selection* wizard page using the *Next* button, return to the previous wizard page using the *Back* button or exit the wizard pressing the *Cancel* button.

<<< NoSQL Action Wizard - Timer Type NoSQL Action Wizard - Timer Selection >>>

The Interval Timer Settings wizard page allows configuring the time interval at which the current action should be (periodically) executed.



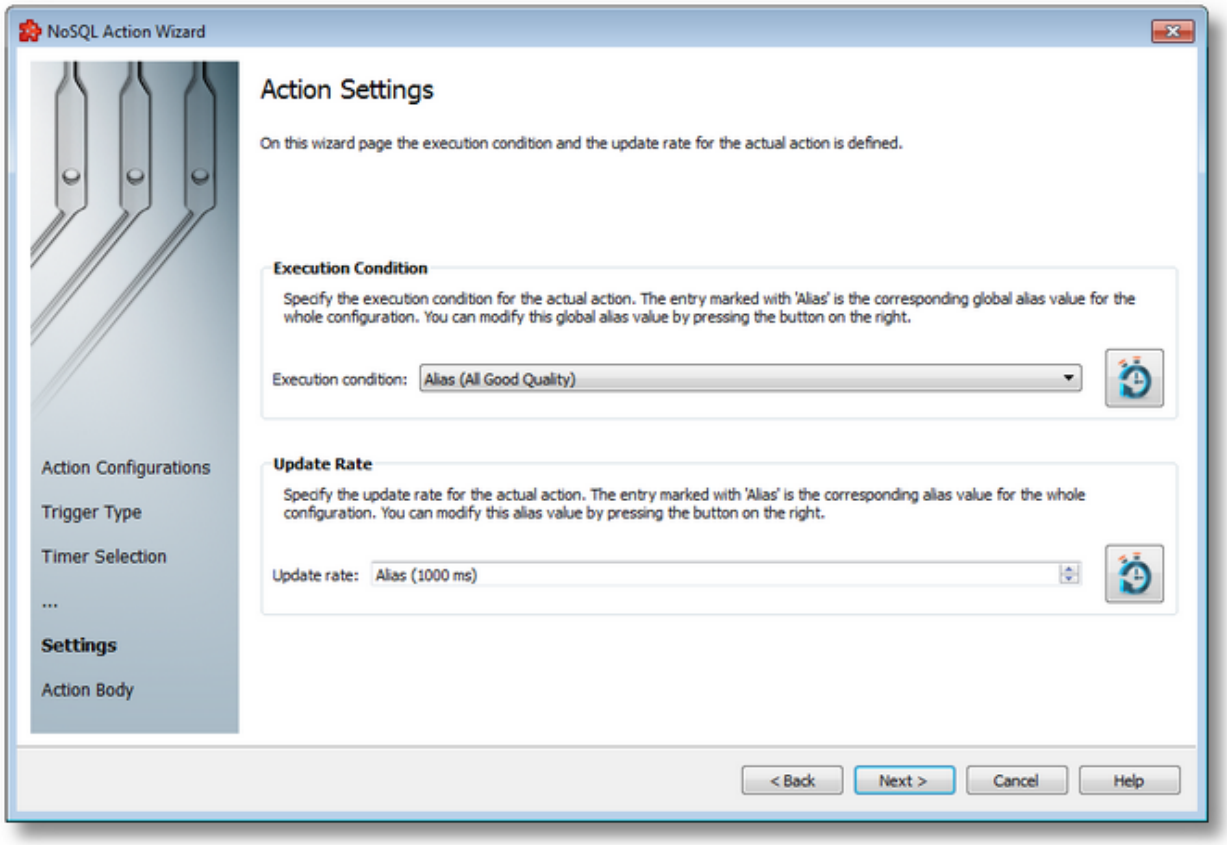
The time interval value is measured in milliseconds and ensures that the action will be triggered at the time interval defined in the spin-box.

Continue to the *Timer Selection* wizard page by pressing the *Next* button, return to the previous one by using the *Back* button, or exit the wizard by pressing the *Cancel* button.

<<< NoSQL Action Wizard - Timer Type NoSQL Action Wizard - Timer Selection >>>

The Action Settings wizard page allows defining the execution condition and the subscription update rate for the current action.

© Softing Industrial Automation GmbH



The wizard page is divided into two sections: *Execution Condition* and *Update Rate*.

Execution Condition

Each statement has an execution condition which is evaluated by the local dataFEED OPC Suite application. The action is only performed when the evaluation is successful. If the evaluation fails then the statement is ignored.

The execution conditions allow reacting to server communication errors, executing special statements or preventing the execution of other statements.

The execution condition can be selected directly from the drop down combo box. The initial value is the global alias one. Changing the global default execution condition can be done using the Configure global alias and initial values button  or pressing the keyboard shortcut CTRL+D.

Update Rate

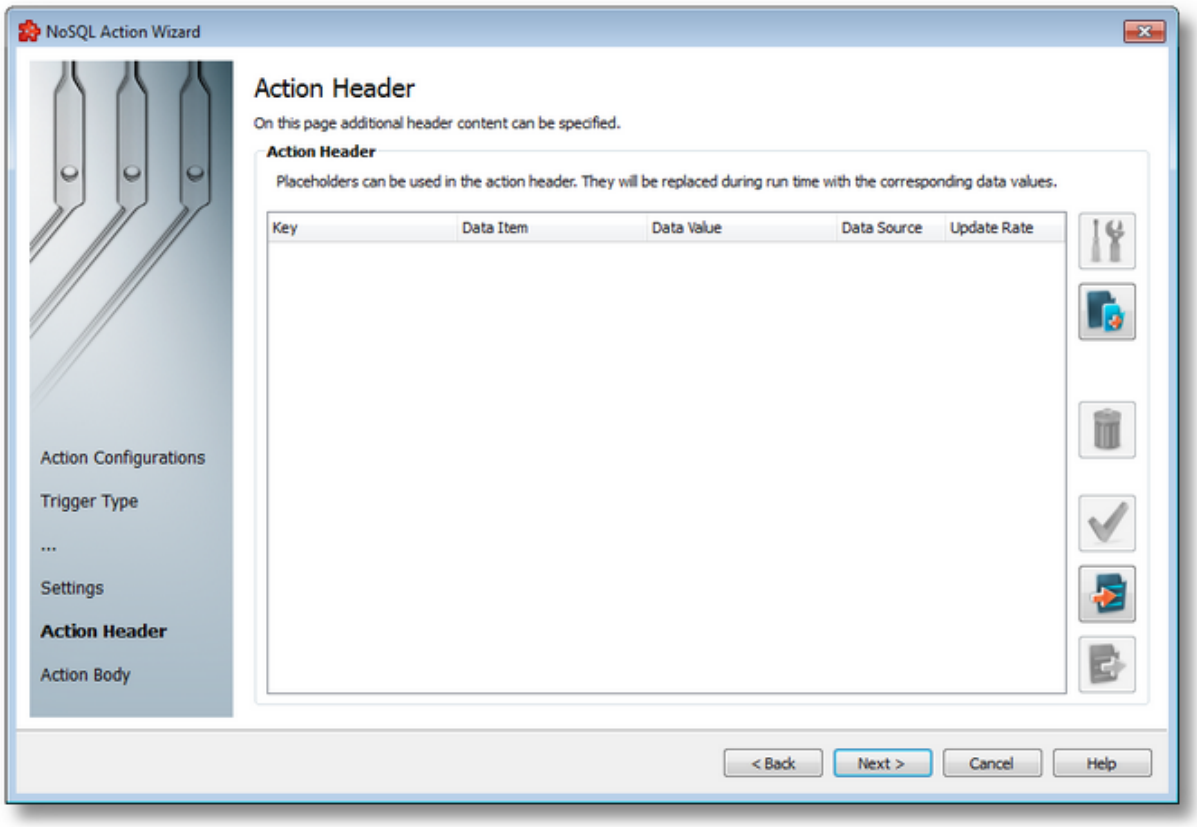
The *Update Rate* specifies the execution rate of the current action.

The update rate can be typed or selected from the spin box. The initial value is the global alias one. Changing the global default update rate can be done using the Configure global alias and initial values button  or pressing the keyboard shortcut CTRL+D.

Complete the *NoSQL Action Wizard* pressing the *Finish* button, review the past wizard pages using the *Back* button, or cancel the wizard thus discarding all the changes by pressing the *Cancel* button.

<<< NoSQL Action Wizard - Trigger Type NoSQL Action Wizard - Action Header >>>

On this page additional header information can be provided for the current action.



By using the *Add*  button, the user will be prompted with the *Parameter Properties* dialog in which data coming from items as well as manually entered data can be added to the header section of the HTTP request.

Note: See the Parameter Properties, Import and Export dialogs on the NoSQL Action Wizard - Action Body Definition page.

By default the *Content-Length*, *Content-Type* and *Authorization* header attributes are set by the application, if their values are specified. These values are not visible on this wizard page, but they can be overridden by the user if they are specified again and new values are provided.

The user can view the final form of the header, as it will be sent by pressing the *View* button. It is possible to Import and Export these values by using the two buttons on the left side, bottom.

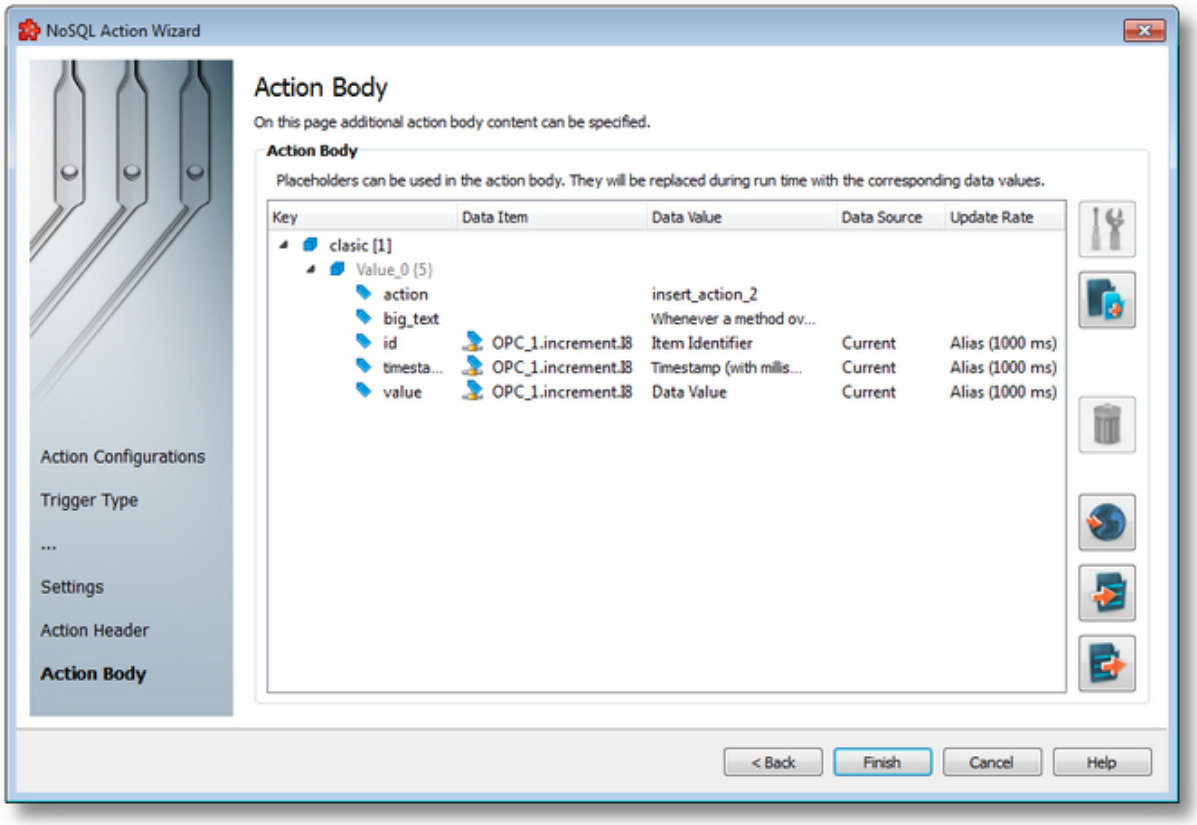
© Softing Industrial Automation GmbH

<<< NoSQL Action Wizard - Action Settings NoSQL Action Wizard - Action Definition >>>

On this page additional body information can be provided for the current action.

MongoDb, CouchDb and InfluxDB actions body needs to be configured differently: MongoDb and CouchDb has a similar way of defining the content of the action request corresponding to the specified content format; InfluxDb uses a more specific payload structure for the server to be able to accept it.

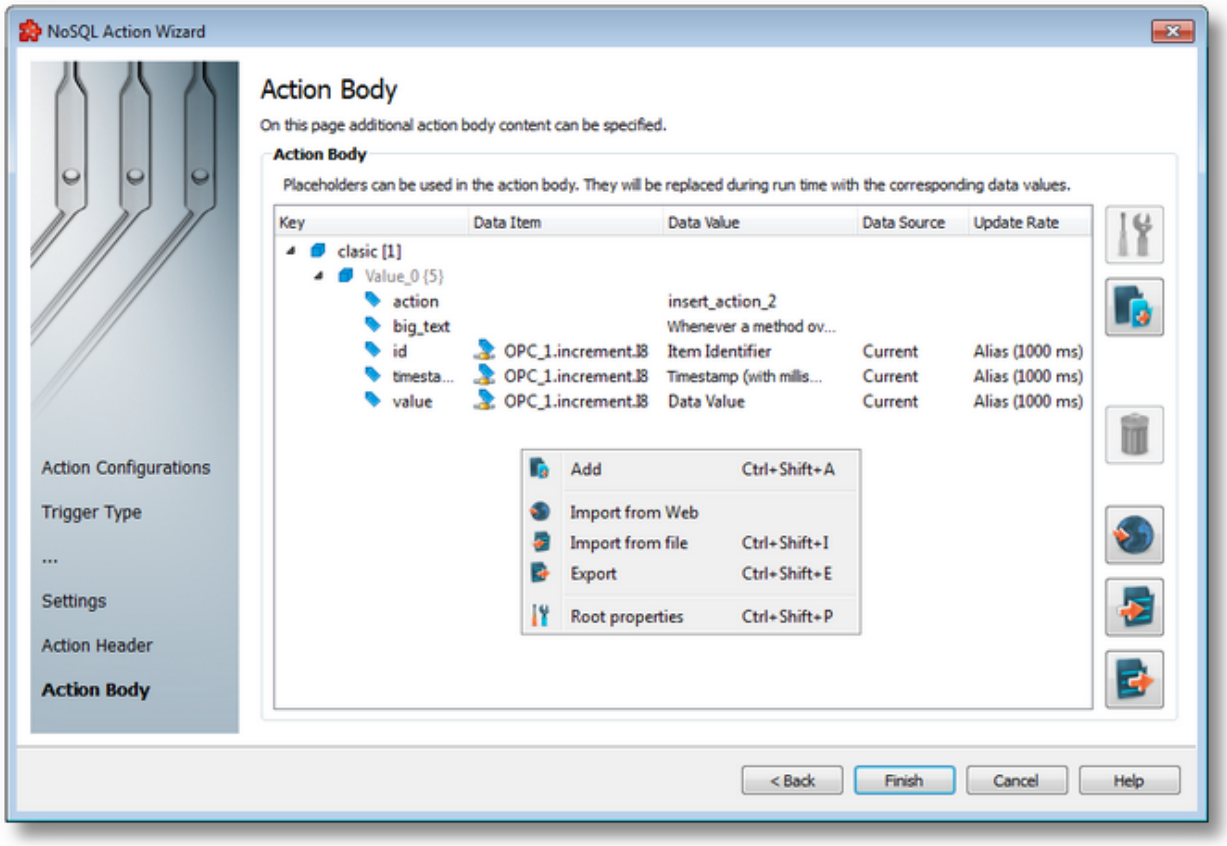
MongoDb and CouchDb page:



The most important part of the wizard flow is the action body editing page. This is where the actual content of the action request is specified in the selected content format, on the first wizard page. The user needs to create the server specific payload structure in this wizard for the server to be able to accept it.

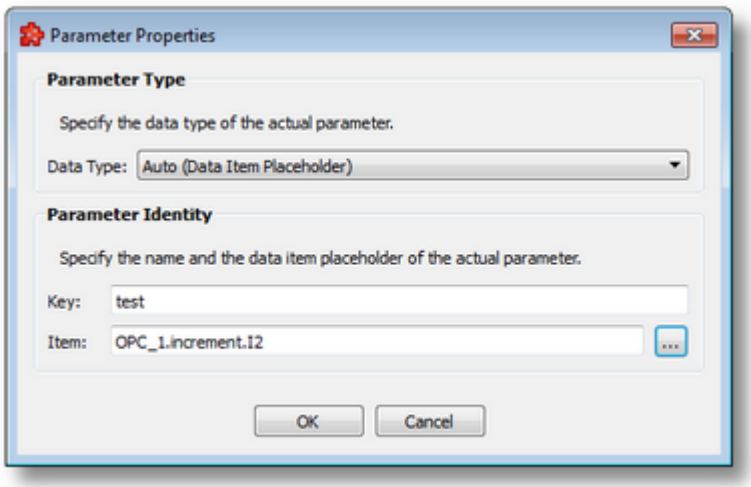
Similarly to the header page, the user has the possibility to add data placeholders coming from items from the address space, or statically defined data values, in different types.

The user can right-click in the tree view above and access all the available options from a context-menu as well:



By using the *Properties*  button, any of the elements in the structure can be modified.

By using the *Add*  button, the user will be prompted with the *Parameter Properties* dialog in which this is configurable.



© Softing Industrial Automation GmbH

The image above shows the dialog set with data coming from an item. A key needs to be given in all cases, then the value in the box below the key. The user is able to specify other types as well, by selecting on of the types from the *Data Type* list.

For more information refer to the Parameter Properties page.

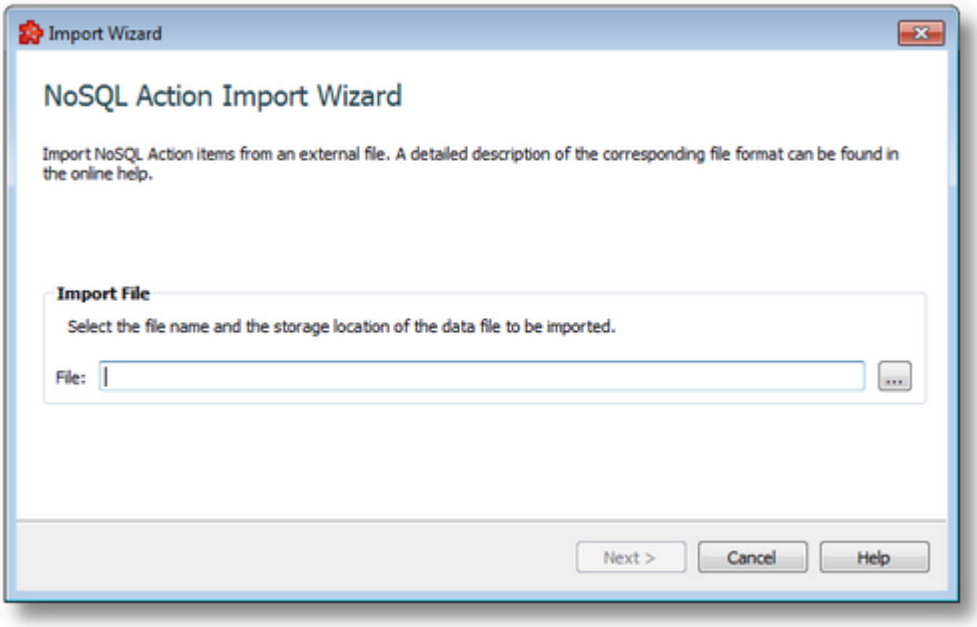
Note: Due to performance penalty reasons the maximum number of items that can be included in the header or body data of the action is limited to 10000. The maximum number of nodes in the structure of either of them is 100000 nodes. If these limits are reached or exceeded the contents will be truncated to fit. For the plain text format the maximum allowed character number in the body content is 100000 characters.

Note: Different update rates can be specified for each action item compared to that of the action, but they will only work if the user specifies the "Current" and "Cache" settings for the item. It will not work with the "Changed" setting.

Import and Export

The user has the possibility to import and export body content to be able to reuse it for other action definitions that only need small changes.

To do this, the *Import* and *Export* buttons can be used, which allow the user to specify a source file from which the content will be imported, and destination file to which it will be saved. Also, a *Import from Web* button is available to be able to get content from the Web as well.



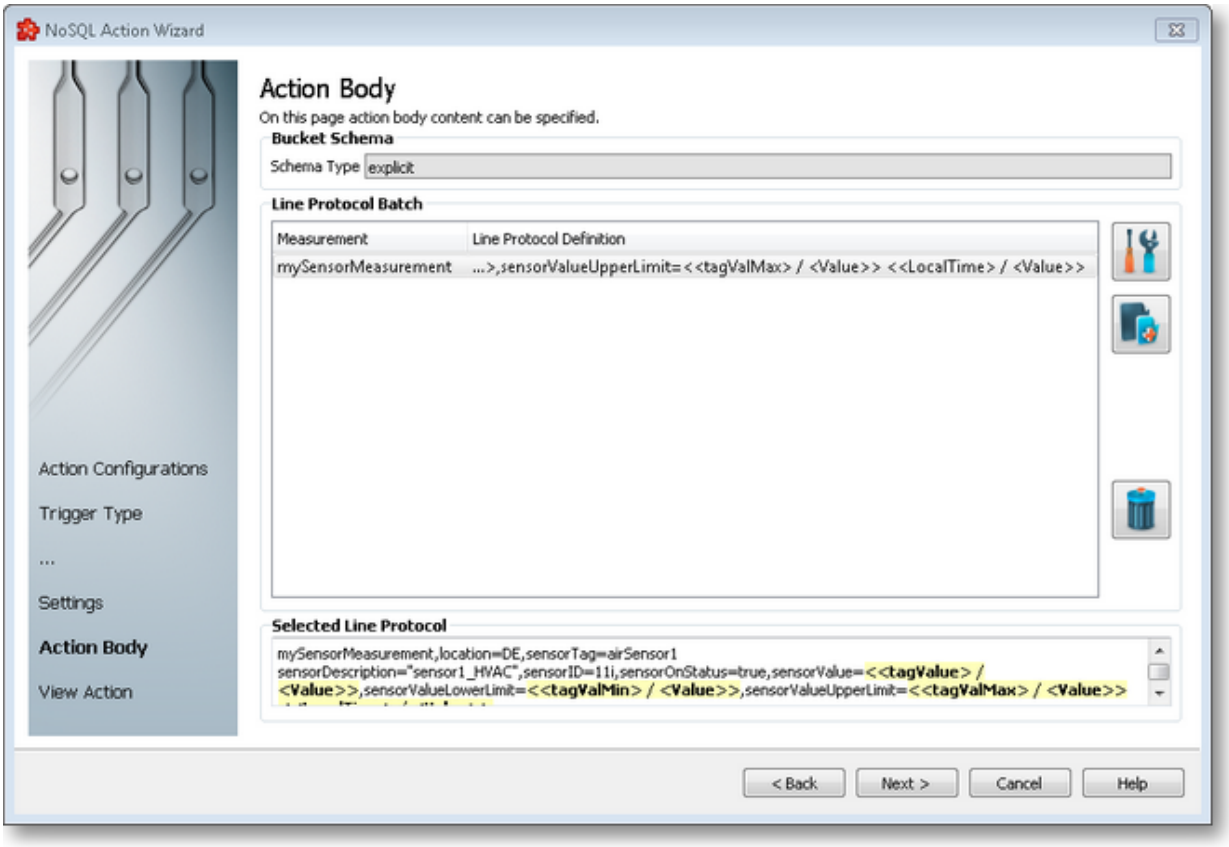
Note: When using the *Import from Web* functionality with HTTPS URLs, no certificates or other types of authentication data can be provided.

After a failed import or export operation the results page may contain the following information:

- The invalid line where the operation failed.
- The reason why the line is considered invalid.

InfluxDb page:

InfluxDb uses a more specific payload structure in the content of the action request. The Action Body wizard page allows to create Line Protocol definitions and to add them to a Line Protocol Batch.



Prior to opening the InfluxDB Action Body wizard page, the bucket measurement schema is attempted to be retrieved. Please see the end user documentation for more details.

For an explicit bucket schema type, the explicit measurement schema is retrieved.

For an implicit bucket schema type, the measurement tag keys and fields are retrieved. If the default retention policy is not set for the bucket that is configured in the NoSQL Action Wizard, a default "autogen" database retention policy mapping is added to the bucket, in order to be able to retrieve the implicit measurement tag and field keys.

By using the *Add*  button, the user can create a new line protocol definition. The user will be prompted with the *Line Protocol Composer* dialog in which the needed parameters are configurable.

© Softing Industrial Automation GmbH

With the *Properties*  button, any of the Line Protocol entries in the Line Protocol Batch table is allowed to be modified.

For the creation of the line protocol definition, the Line Protocol Composer dialog uses the retrieved measurement schema information of tag and field keys, and usable tag and field value data types.

For more information refer to the Line Protocol Composer page.

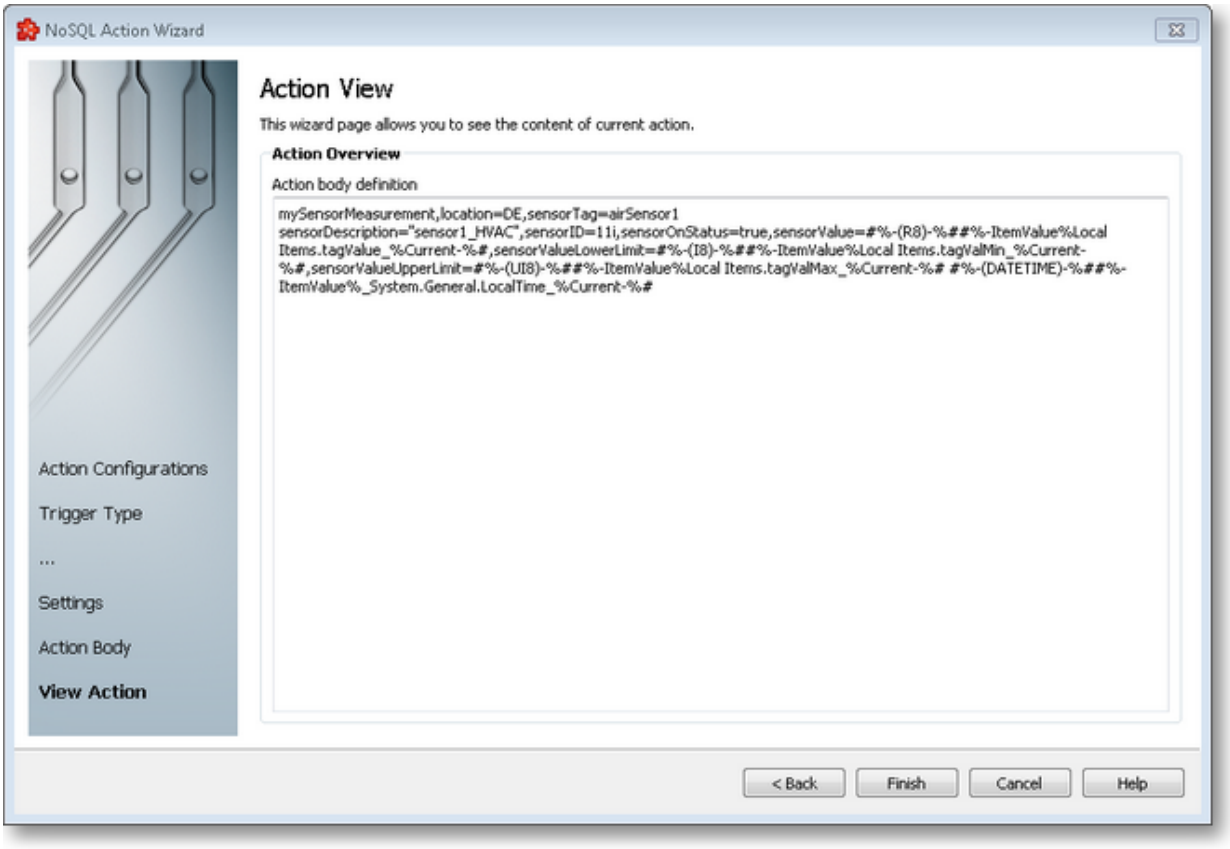
<<< NoSQL Action Wizard - Action Header NoSQL Action Wizard - Action Overview >>>

On this wizard page the user is presented with the final form of the content of the action body value.

The InfluxDB action configuration wizard includes the Action Overview wizard page also, while this is not included for the MongoDb and CouchDb actions.

The Action Overview wizard page displays the resulted batch of individual Line Protocol formats.

If the tag / field parameter values are item values of data placeholders corresponding to address space items, the item's data values placeholder format is detailed (see Data Values Placeholder), preceded by the conversion expression to the *InfluxDB* tag / field destination data type.



The Action Overview wizard page contains a read-only view, used just for reviewing the correctness of the content. The user can only select and copy the content from the box.

In case of plain text ENTER character, this is saved using a special placeholder, which the application will replace at runtime and thus the original form of the body content will be restored.

<<< NoSQL Action Wizard - Action Definition

7.3.3.7.4 Data Value Placeholder Wizard

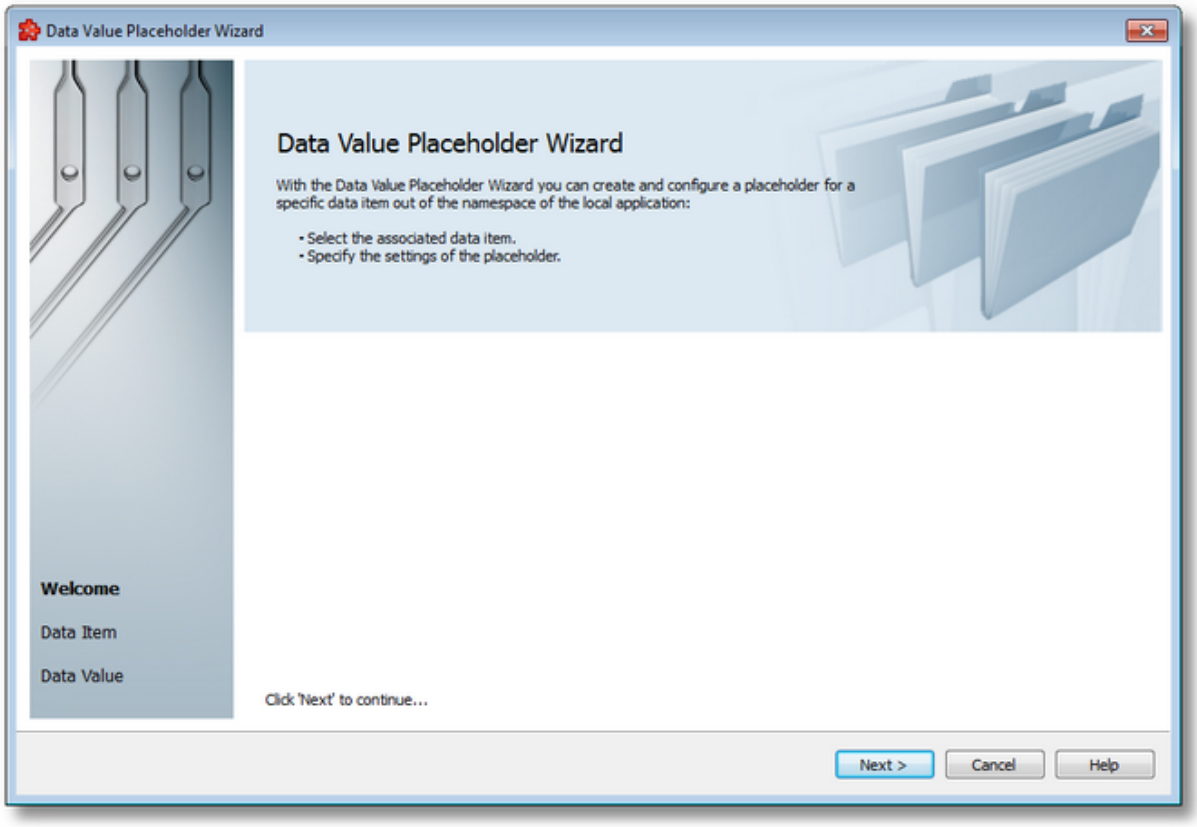
▼ With the Data Value Placeholder Wizard you can create a placeholder from a configured data source:

- Obtain a general description of the wizard @ Data Value Placeholder Wizard - Welcome
- Select a placeholder item @ Data Value Placeholder Wizard - Data Item
- Select the data value and source @ Data Value Placeholder Wizard - Data Value

The Welcome wizard page describes the individual steps for defining the placeholder:

© Softing Industrial Automation GmbH

- Select an item from the local dataFEED OPC Suite application namespace.
- Choose the data value and the data source for the selected item.

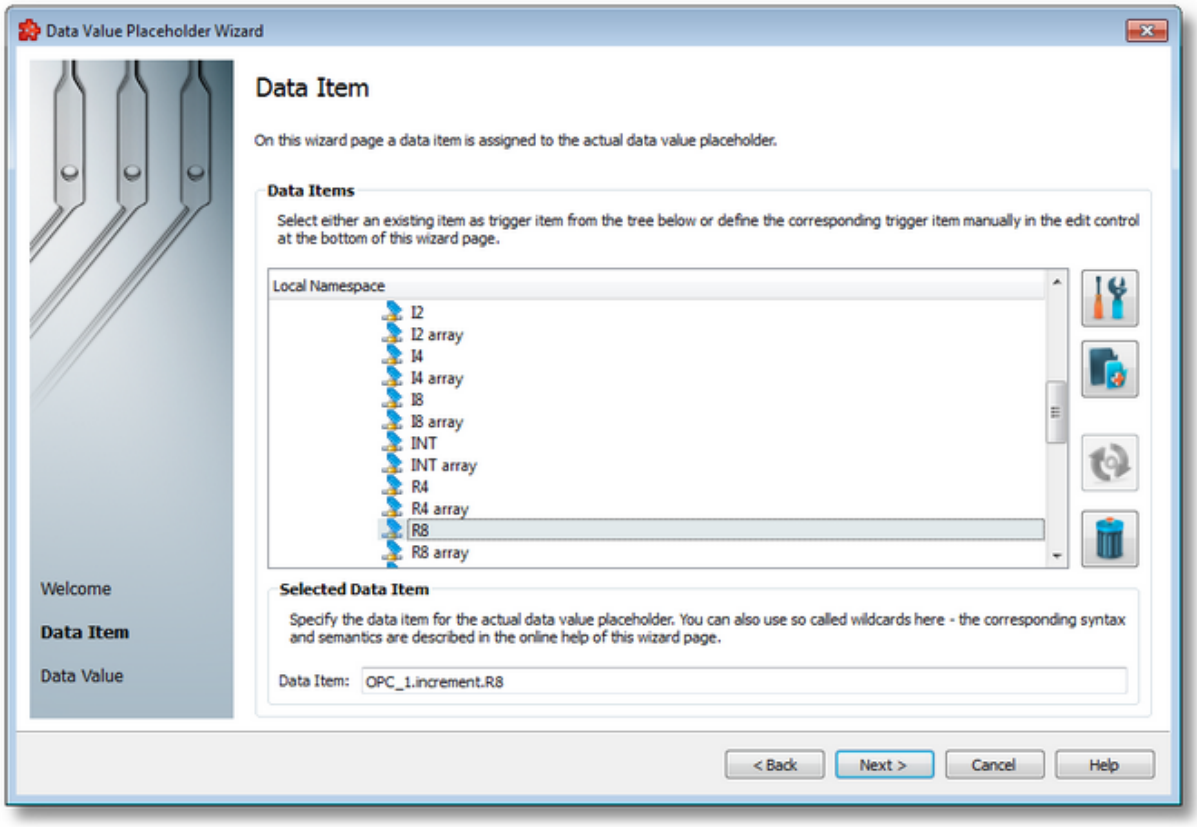


Continue to the next wizard page using the *Next* button or quit the wizard pressing the *Cancel* button.

Data Value Placeholder Wizard - Data Item >>>

The Data Item wizard page allows selecting the placeholder item. The placeholder item can be:

- Any item originating from a *Server Connection*, from the *Local Items* or from the *System Items* can be selected, choosing a tag from the *Local Namespace*.



Local Namespace Tree Handling

The *Local Namespace* tree offers the possibility to explore the entire namespace of the local dataFEED OPC Suite application in order to select a data item. After the selection is made, the *Data Item* edit box is populated with the selected item ID.

The entire namespace of the local dataFEED OPC Suite application can be managed through this wizard page by performing certain operations using the buttons on the right, the context sensitive menu or the keyboard shortcuts:

- Open the item properties dialog using the *Properties of the selected item* button  , pressing the keyboard shortcut CTRL+P or selecting the *Properties* action from the context menu. Depending on the type of the selected item, a special Item Properties dialog opens:
    - If the selected item is under Local Items the Modify Local Item dialog is opened.
    - If the selected item is under a dynamic server connection the Modify Dynamic Server Item dialog is opened.
    - If the selected item is under a static server connection the Modify Static Server Item dialog is opened.
- Add a new item using the *Add a new item*  button, pressing the keyboard shortcut CTRL+A or selecting the *Add* action from the context menu. Depending on the type of the selected node, a special Add Item dialog opens:

© Softing Industrial Automation GmbH

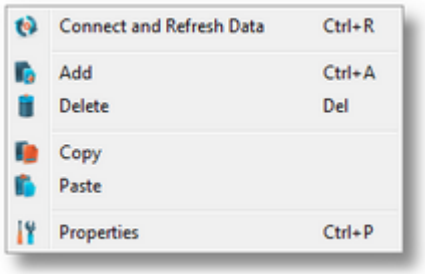
- If the selected node is under Local Items the Add Local Item dialog is opened.
- If the selected node is under a dynamic server connection the Add Dynamic Server Item dialog is opened.
- If the selected node is under a static server connection the Add Static Server Item dialog is opened.

- Copy the selected item to clipboard by selecting the *Copy* action from the context menu.
- Paste the previously copied item from clipboard by selecting the *Paste* action from the context menu. Items originating from the Local Items cannot be pasted under server connections, and vice-versa; an error message will be displayed in both cases. If an item is pasted in a static (filtered) server connection and the item does not exist on the server an error message is displayed.

- Connect to the server and refresh the data items by using the *Refresh data items* button, pressing the keyboard shortcut CTRL+R or selecting the *Connect and Refresh Data* action from the context menu. The namespace of the selected server connection is recreated; if manually added items not referenced in the actual configuration exist, those items are deleted from the namespace.

- Delete the selected item and all of its child items by using the Delete data items button , pressing the DELETE / DEL key or selecting the Delete action from the context menu. Items which are included in any Data Processing / Data Destination add-ons will not be deleted but an error message will be displayed instead.

An example of the context menu can be seen below:



Attention: The *Local Namespace* root node, the *Local Items* root node and the server connection root nodes cannot be deleted. An item that is used in consumers or a system item cannot be deleted either.

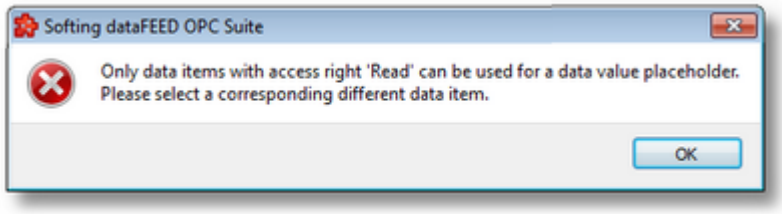
Note: The copy/paste actions work on the selected tag/property or on the whole selected branch. The copy action is not available for the Local Namespace root or for the server connection root nodes; the paste action is not available for items of type property (as they cannot have any children).

The entire namespace of the local Softing dataFEED OPC Suite application is exposed in this wizard page. The tree root node name is composed of the *Local Namespace* keyword followed by the instance name.

Selected Data Item Handling

When an item from the Local Namespace is selected, the *Data Item* edit box is populated with the item ID.

Note: Only tag and property items are allowed to be used in placeholder. If a node item is selected, an error message dialog will appear when the *Next* button is pressed.



Regular expressions with special characters (asterisk, question mark, square brackets) are not supported by this module.

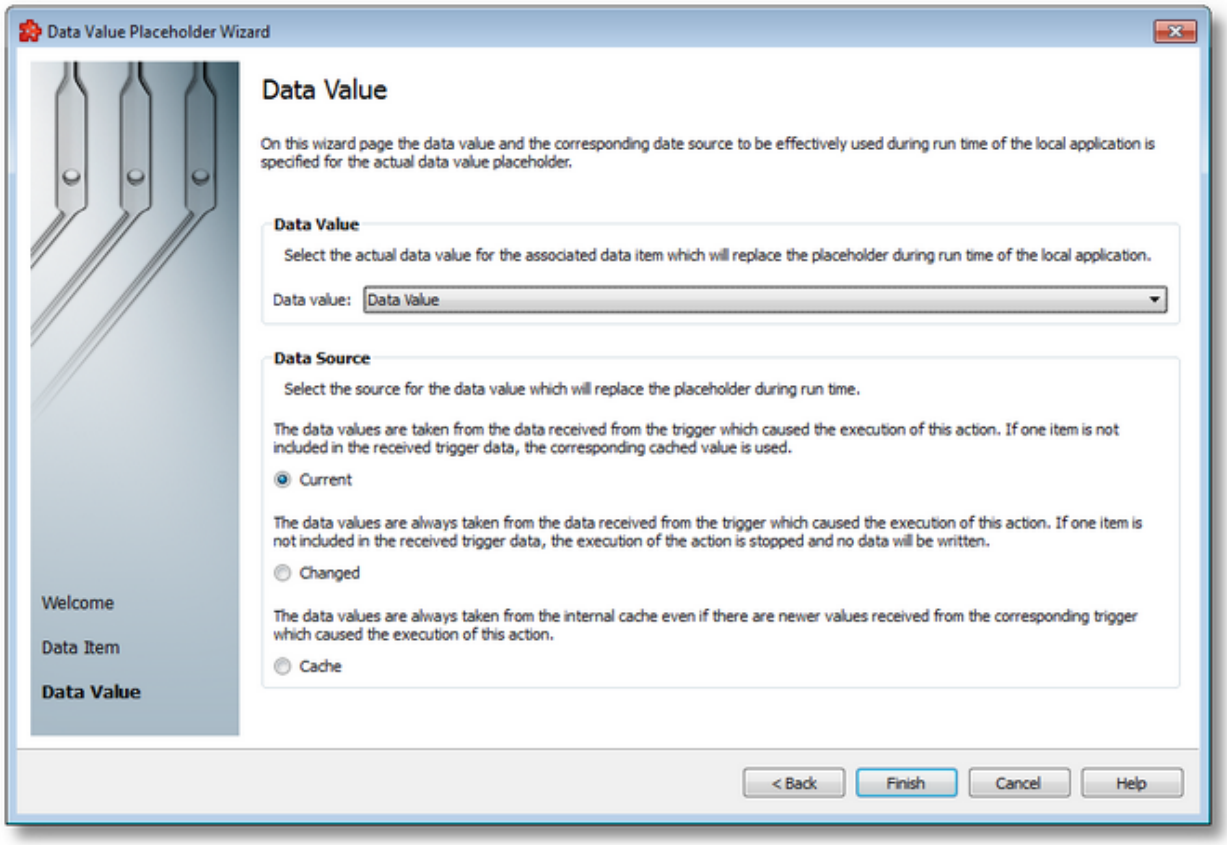
Proceed to the next page to set the item data value and source by using the *Next* button, return to the previous wizard page by using *Back* button, or quit the wizard by pressing the *Cancel* button.

<<< Data Value Placeholder Wizard - Welcome Data Value Placeholder Wizard - Data Value >>>

The Data Value wizard page provides the possibility to set the properties of the placeholder:

- Data Value - select the actual data value for the associated data item which will be replaced during the run time of the local dataFEED OPC Suite application.
- Data Source - select the source for data value which will be replaced during the run time of the local dataFEED OPC Suite application.

© Softing Industrial Automation GmbH



Data Value

The following data value attributes may be used:

Attribute	Description
Data Value	The value of the data source server item.
Item Identified	The ID of the data source server item.
Quality as Integer	The data source server item quality as integer.
Quality as String	The data source server item quality as string.
Timestamp (without milliseconds)	The data source server item timestamp as date time without milliseconds.
Timestamp (with milliseconds)	The data source server item timestamp as date time including milliseconds.
Timestamp (milliseconds part only)	The milliseconds part of the data source server item timestamp.
Internal Timestamp (without milliseconds)	The timestamp of data change in the local dataFEED OPC Suite application as date time without milliseconds.

Internal Timestamp (with milliseconds)	The timestamp of data change in the local dataFEED OPC Suite application as date time including milliseconds.
Internal Timestamp (milliseconds part only)	The milliseconds part of timestamp of data change in the local dataFEED OPC Suite application.

Data Source

The following data sources can be selected:

Data Source	Description
Current	The value used is the current one. This means that if the item is included in the transaction, the received value is used. If the item is not included, the cached value is used.
Cache	The cached value is always used, even if the item is present in the incoming transaction.
Changed	The item must be included in the incoming transaction as the current value will be used.

Attention: If the action trigger is set to *Data Change* and the corresponding action contains at least one placeholder having *Changed* as the *Data Source*, that specific item must exist in the incoming transaction in order for the action to be executed. When the item is not present in the transaction, the action is invalidated.

Complete the wizard using the *Finish* button, go back to the previous wizard page using the *Back* button or discard any changes made in this wizard by pressing the *Cancel* button.

<<< Data Value Placeholder Wizard - Data Item

7.3.3.7.5 Supported Databases

This section of the Softing dataFEED OPC Suite online help describes the supported NoSQL databases. Please see the topics within for more details about every specific database.

- Define a MongoDB configuration @ MongoDB

© Softing Industrial Automation GmbH

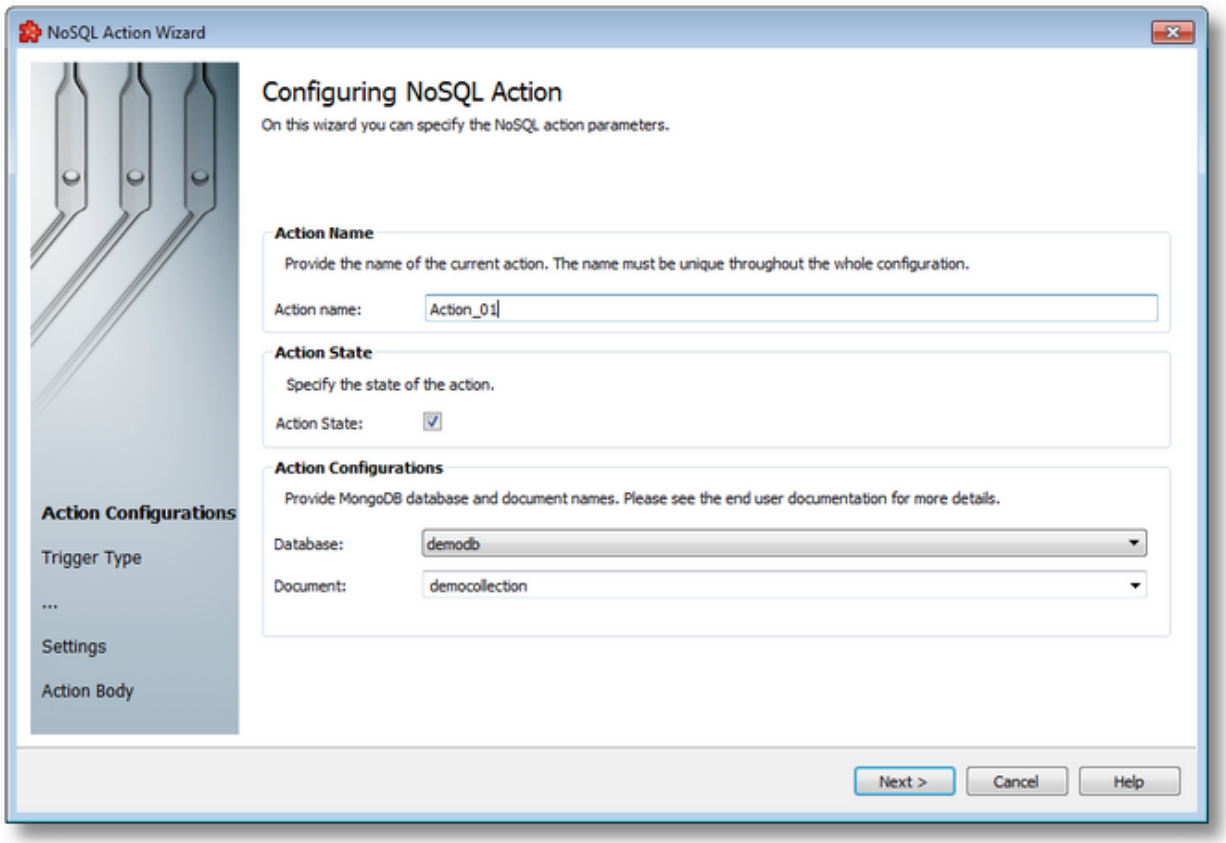
- Define an Apache CouchDB configuration @ CouchDB
- Define an InfluxDB configuration @ InfluxDB

dataFEED OPC Suite can connect to and insert data into Mongo databases. The *MongoDB* server consists of named databases which contain named collections. Every triggered action will be recorded in the *MongoDB* as a new document of a specified collection from a selected database. By default, named collections which do not already exist in the *MongoDB* database will be automatically created.

The *MongoDB* data destination plugin connects to the *MongoDB* server with or without authentication. Username/password or X.509 certificates can be used to authenticate to the server. Both secured and unsecured server connections are supported. Replica sets are currently not supported.

Insert Operation

For every triggered *MongoDB* action a new document is created and inserted in the selected database/collection. The server guarantees atomicity at collection level over the entire insert operation.



For every insert operation or ping command the dataFEED OPC Suite will advertise its name to the MongoDB server, either as the "Softing dataFEED OPC Suite - Configurator" for operations

conducted by the Configurator itself (e.g. database retreival, collection retreival, certificate retreival, ping/connection test) or as the "Softing dataFEED OPC Suite - <Configuration name>" for operations conducted by the runtime (e.g. ping/connection test, insert operations).

The dataFEED OPC Suite runtime will only ping the default "admin" database of every action when the service/application is started. The databases containing the used collections will be pinged for every error occurred while inserting a document in order to better diagnose connection problems.

Softing dataFEED OPC Suite employs a request retrying mechanism for most of the errors that can occur. The requests will be retried in such a manner that the order of the data will be preserved. For some errors however, there is no point in retrying the request as the very same error will occur again and the request queue will not advance.

The following errors will cause a request to be dropped and not retried if it fails once:

Write concern error

Request too big

Request too small (empty request)

Invalid request body

Invalid namespace (invalid collection name)

Collection insert failed (empty insert operation)

Duplicate key

If a request fails with any other different error, it will be retried until it succeeds.

Note: The default namespace delimiter - that is the dot/period character (**.**) - cannot be used in the field names of the MongoDB JSON body as this character is reserved for the MongoDB dot query notation. The field names are not allowed to start with the dollar sign neither (**$**).

The driver used for connecting to the MongoDB server is MongoC version 1.6.3 and it is build against (it uses) the OpenSSL version 1.0.1.

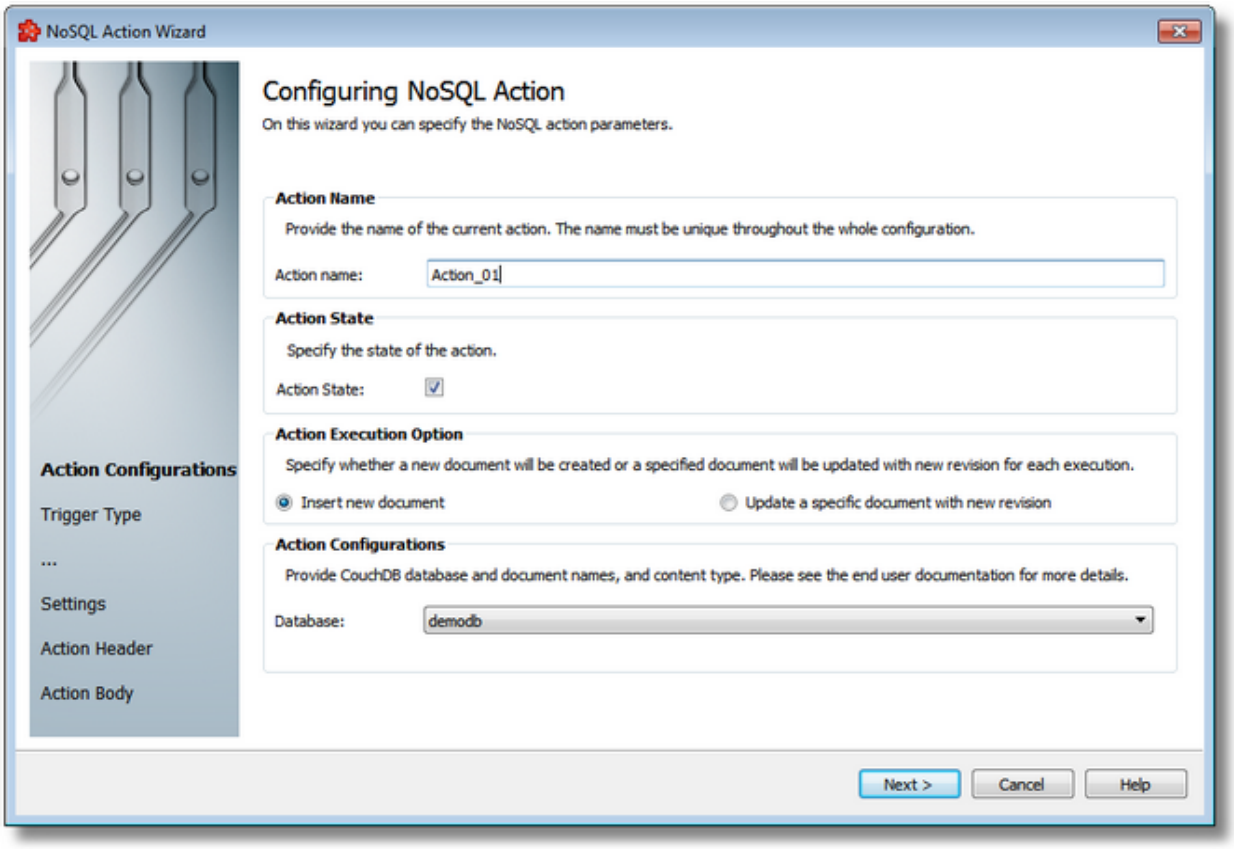
Note: Specific to MongoDb connections is that they do not support proxy settings at the moment, because of MongoC driver limitations.

CouchDB allows to perform two types of operations, insert a new document and to update an existing one. A *CouchDB* server hosts named databases, which store documents. Each document is uniquely named in the database, having a unique *ID* and *revision*. The *ID* and revision are provided by *CouchDB* server. The *CouchDB* file layout and commitment system features all *Atomic Consistent Isolated Durable* (ACID) properties.

Insert Operation

If inserting operation is chosen, only the *Database* field will be visible and document creation will be left to *CouchDB*. In this case a new document will be created every time when the action will be triggered.

© Softing Industrial Automation GmbH

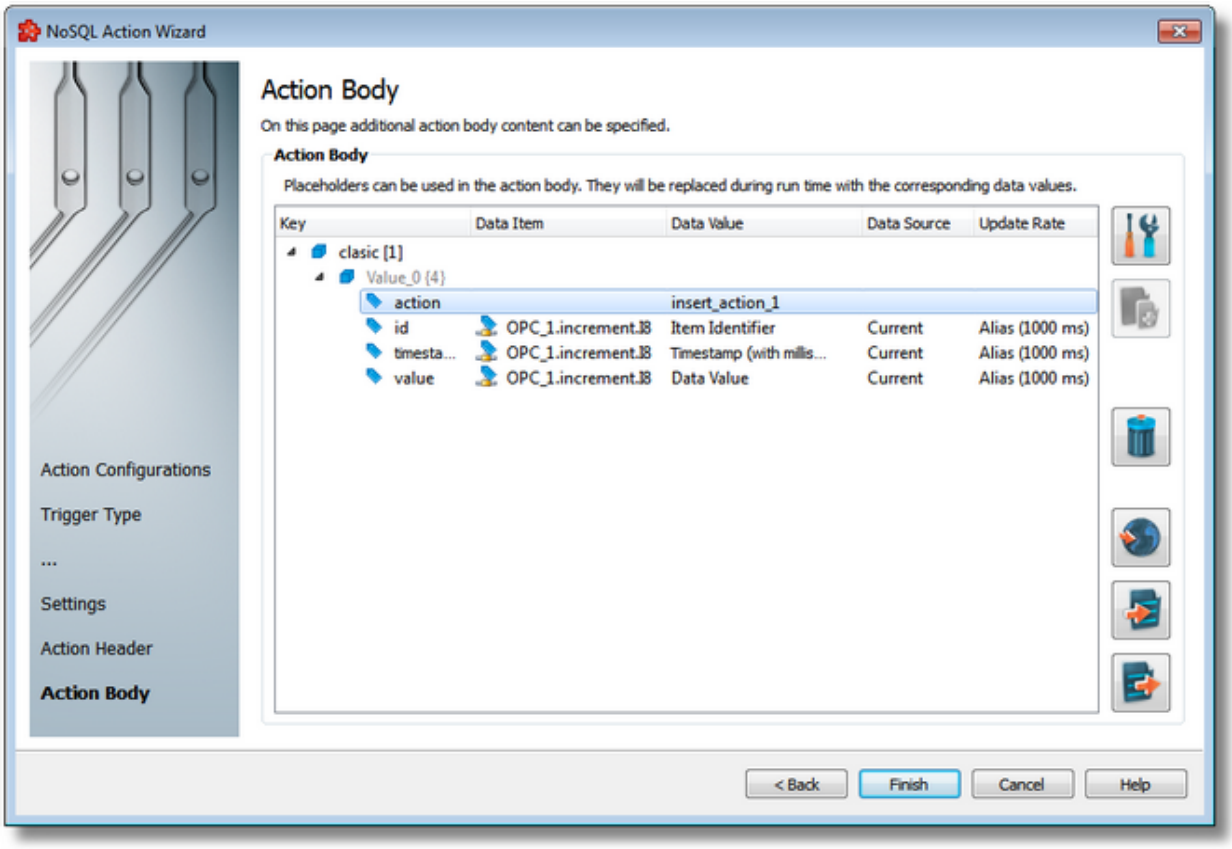


CouchDB doesn't have the concept of collections, like MongoBD.

The document unique *ID*'s will be generated by *CouchDB* server. A new document will be created every time when action will be triggered.

It is possible to execute actions on the same database or to have separate database for each action. In the second case, it is recommended to have a field in the action body to be able to filter the documents for that action. The *CouchDB* official documentation contains more details about *Views*.

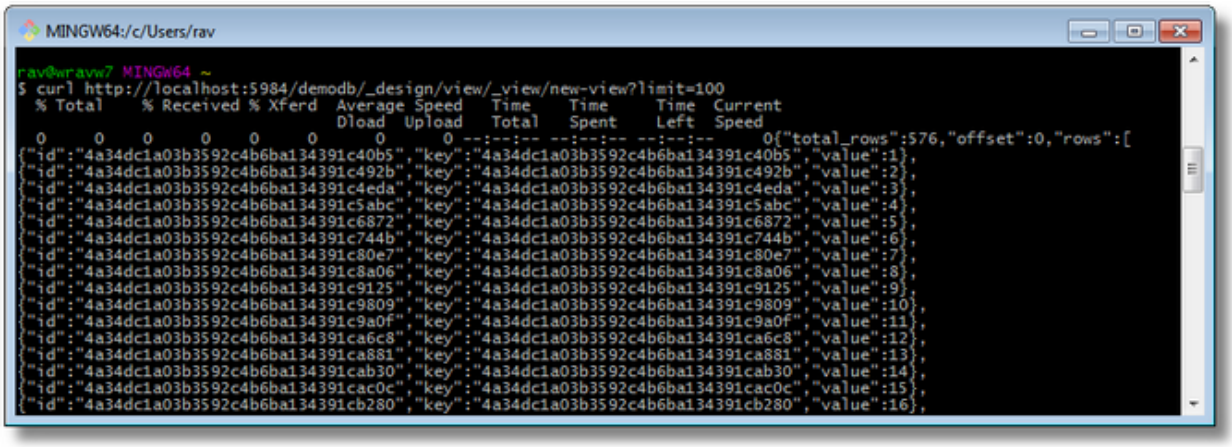
The following image contains a body that has defined a type identifier named "action".



A query used by a *View* that will show only the values of the item *OPC_1.increment.I8* for all documents generated by *Action_01* can be written like in the following example:

```
function (doc) {
  if (doc.clasic[0].action == "insert_action_1" && doc.clasic[0].id == "OPC_1.increme
  {
     emit(doc._id, doc.clasic[0].value);
  }
}
```

This image shows the results of executing of the *View* above: only the documents with the action field equaling "*insert_action_1*" are visible in the output.

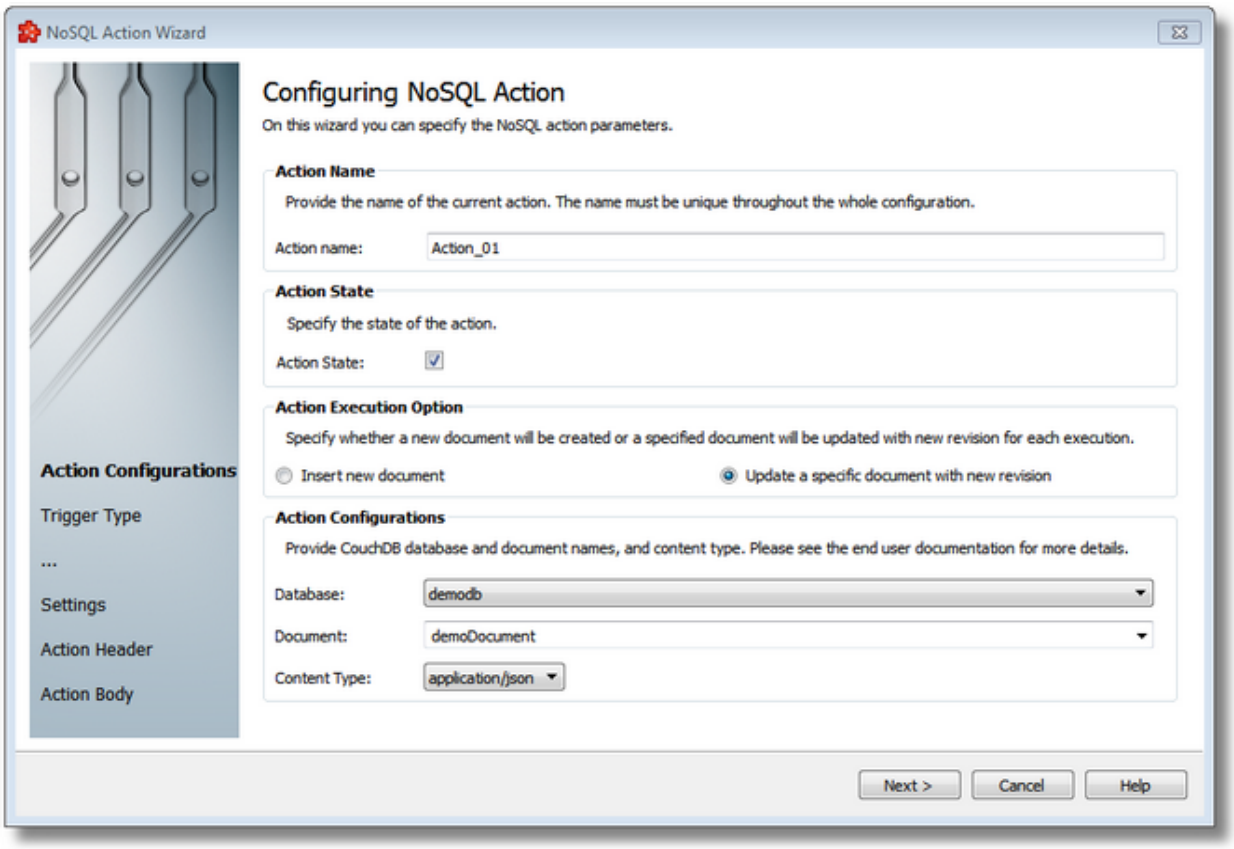


© Softing Industrial Automation GmbH

Attention: Isn't recommended to overwrite the *Content-Type* parameter in *Action Header* wizard page because the *CouchDB* server accepts for this type of operation only *application/json* content type, that is by default.

Update Operation

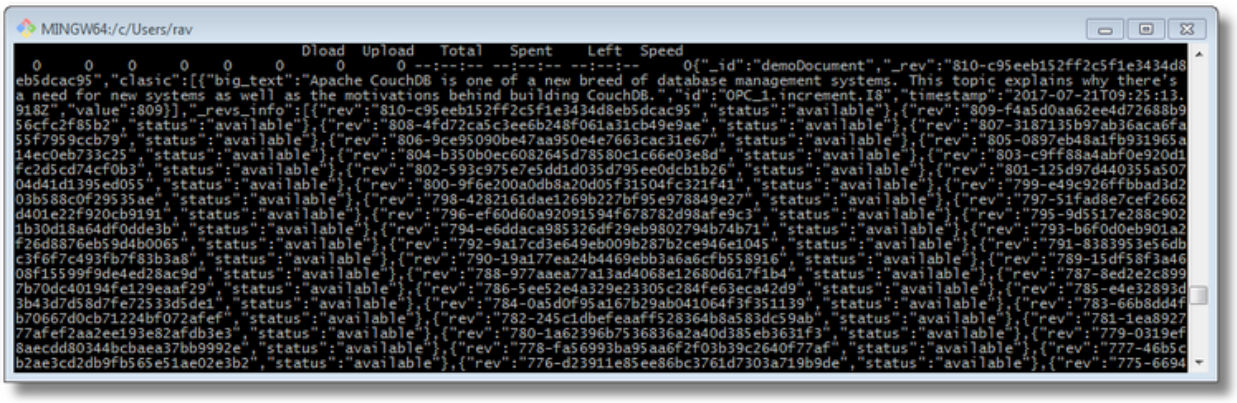
If updating operation is chosen then the *Database*, *Document* and *Content Type* fields will need to be specified. In this case a new revision of the specified document will be created every time when the action will be triggered.



CouchDB has a limit of the stored revisions, by default it is set to 1000. This means that only the last 1000 revisions will be stored. This limit can be changed by issuing an HTTP PUT command:

```
curl -X PUT -d "10000" http://localhost:5984/demodb/_revs_limit
```

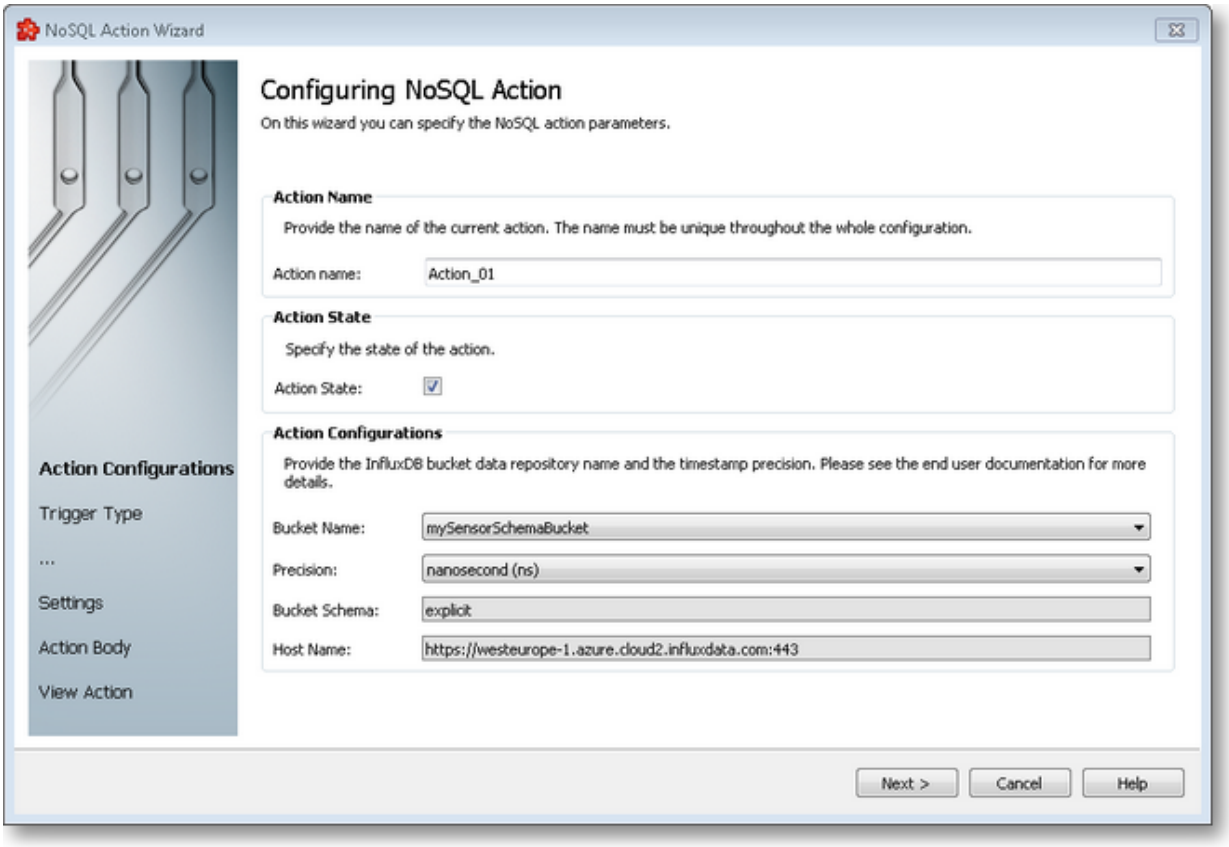
You can fetch a document's revision information by adding *?revs_info=true* to the get request, like in following image.



Note: After the *CouchDB* server receives a request, it will either respond with a status code if the request was successfully executed or with an error code if the request was not executed. If the dataFEED OPC Suite application receives a non-success error/status code, then it will retry or drop the request depending on the received error/status code. The REST Configuration topic contains more details about all the error/status codes.

dataFEED OPC Suite allows to connect and to write data points to *InfluxDB* databases. Optimized writes can be performed by writing data in batches of Line Protocols.

For every triggered *InfluxDB* action a new Line Protocol Batch is written in the selected bucket.



© Softing Industrial Automation GmbH

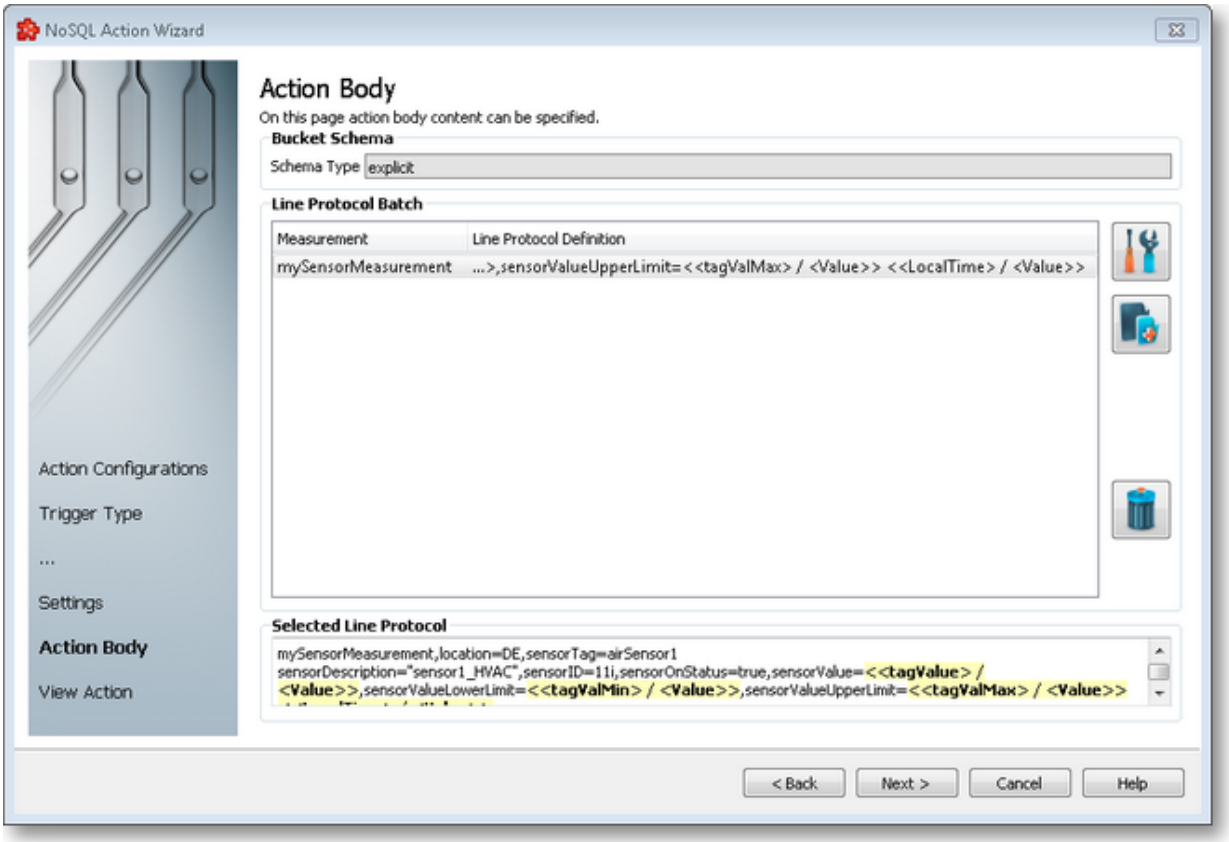
InfluxDb uses a more specific payload structure in the content of the action request. The Action Body wizard page allows to create Line Protocol definitions and to add them to a Line Protocol Batch.

Prior to opening the InfluxDB Action Body wizard page, the bucket measurement schema is attempted to be retrieved. Please see the end user documentation for more details.

For an explicit bucket schema type, the explicit measurement schema is retrieved.

For an implicit bucket schema type, the measurement tag keys and fields are retrieved. If the default retention policy is not set for the bucket that is configured in the NoSQL Action Wizard, a default "autogen" database retention policy mapping is added to the bucket, in order to be able to retrieve the implicit measurement tag and field keys.

On the InfluxDB Action Body wizard page the user can create a new Line Protocol definition and add it to the the Line Protocol Batch table. The user can also modify any of the existing Line Protocol entries in the Line Protocol Batch table.



For the creation of the line protocol definition, the Line Protocol Composer dialog uses the retrieved measurement schema information of tag and field keys, and usable tag and field value data types.

The *Line Protocol Composer* allows to create the required text based specific syntax format for writing points to *InfluxDB* databases.

A point in *InfluxDB* represents a single data record. A point has a measurement, tag set (collection of tag keys and tag values on a point), field set (collection of field keys and field values on a point), and a timestamp.

Please see the end user documentation for more details.

For more information refer to the Line Protocol Composer page.

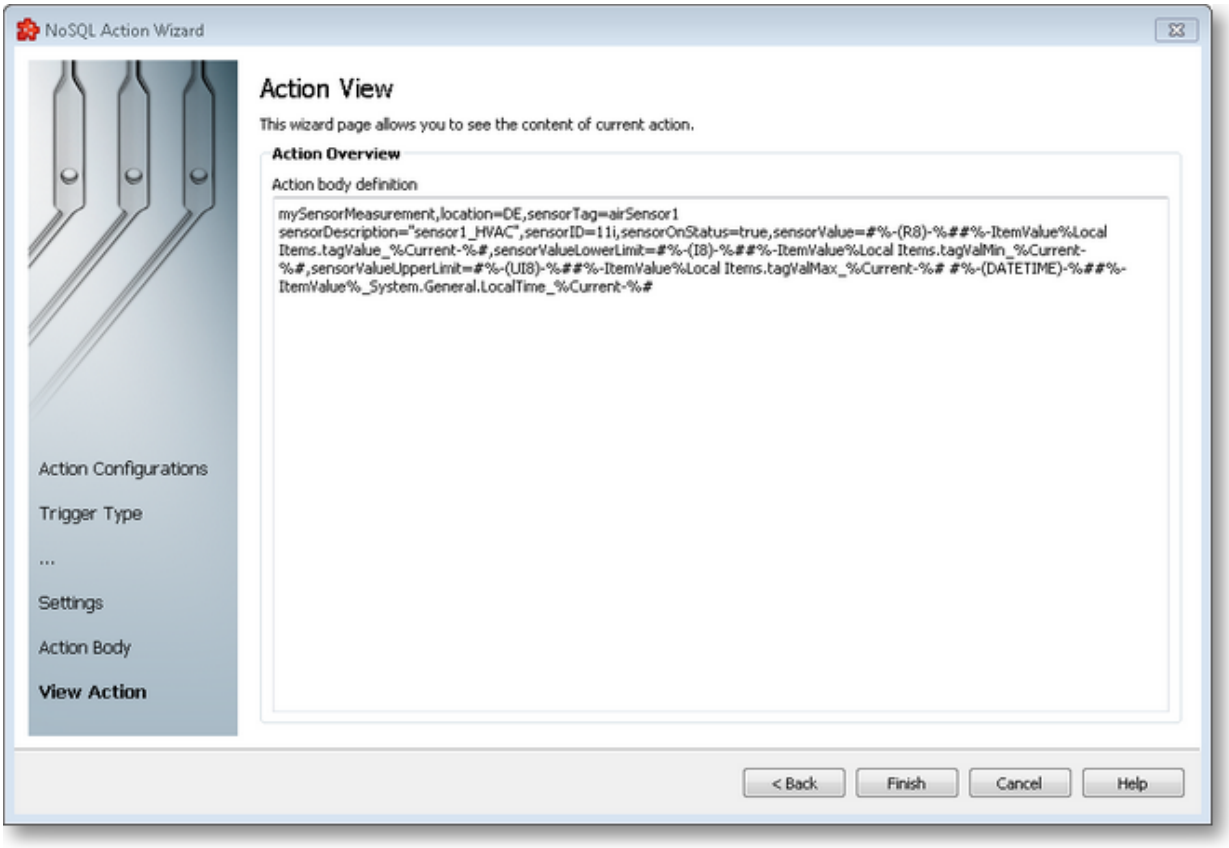
The tag set and field set is created by defining each individual tag / field key-value pair.

On the line Protocol Composer page, the user can create a new tag / field key-value pair definition and can modify existing individual tag / field key-value pair definitions with the use of the *Tag* / *Field Parameter Properties* dialog.

For more information refer to the Tag Field Parameter Properties page.

The *InfluxDB* action configuration wizard concludes with the Action Overview wizard page, that displays the resulted batch of individual Line Protocol formats.

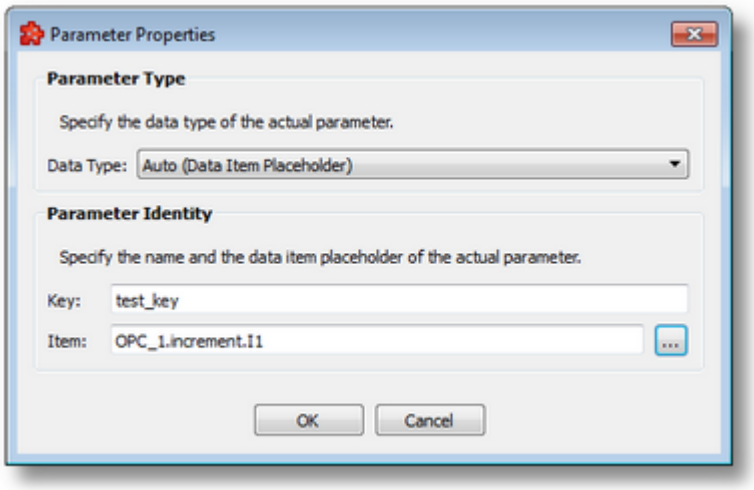
If the tag / field parameter values are item values of data placeholders corresponding to address space items, the item's data values placeholder format is detailed (see Data Values Placeholder), preceded by the conversion expression to the *InfluxDB* tag / field destination data type.



7.3.3.7.6 Parameter Properties

The *Parameter Properties* window offers the possibility to create or modify a *JSON* data type that is the basic element of the action body's definition.

© Softing Industrial Automation GmbH



The following parameter properties can be selected in this window:

- the parameter data type: it can be auto, object, array, string, boolean, and double;
- the parameter key: it should be unique for its level in the tree;
- the parameter value: It is available only for the data types auto, string, boolean, and double.

Data Types

The supported data types are:

- *Auto (Data Item Placeholder)* - is an automatic type and depends on the placeholder definition
- *Object* - an unordered collection of key-and-value pairs
- *Array* - an ordered sequence of values
- *String* - double-quoted *Unicode* with backslash escape
- Boolean - true or false
- Double - double precision floating-point format

Note: The default data type is *Auto (Data Item Placeholder)*.

7.3.3.7.7 Line Protocol Composer

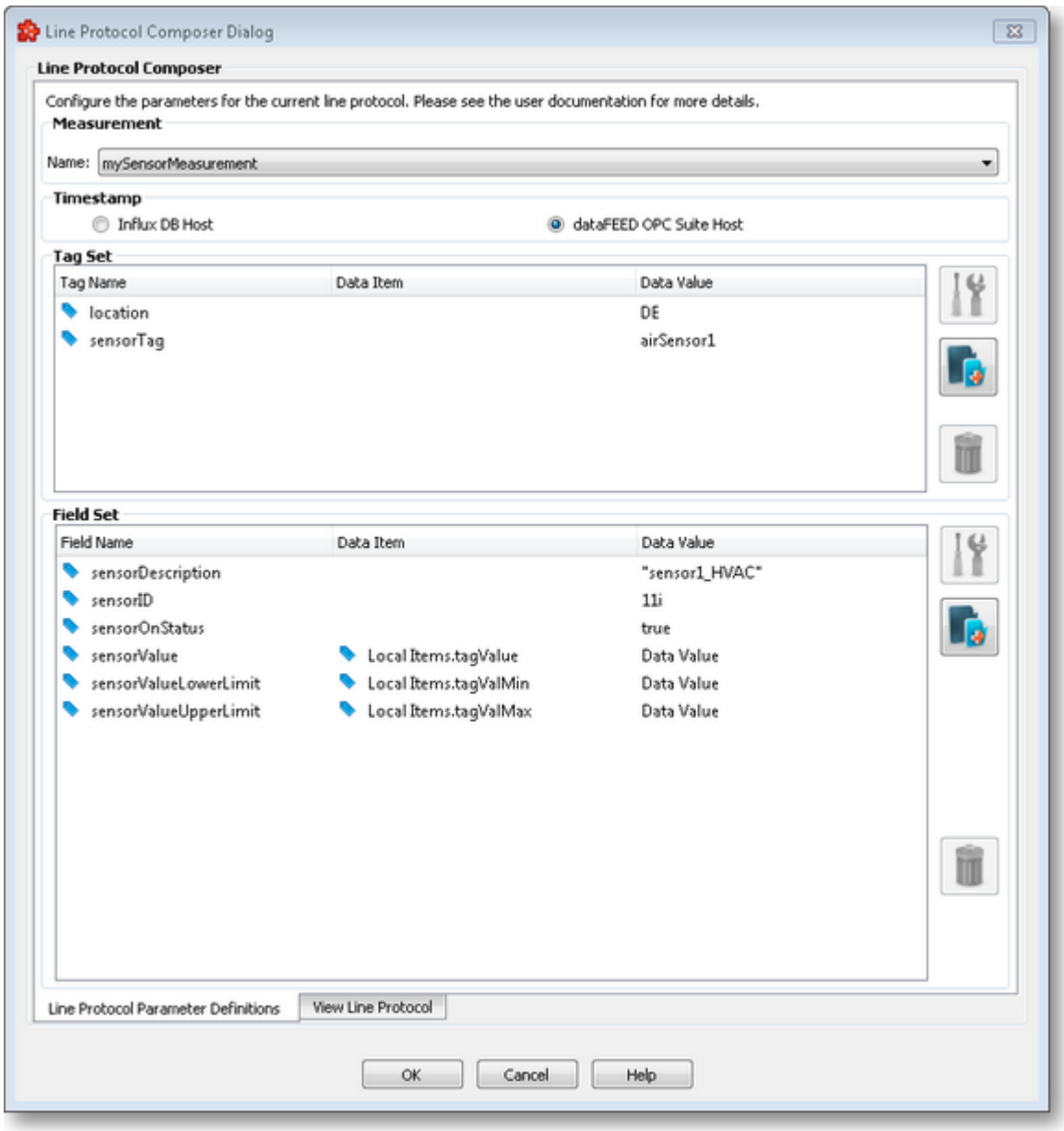
The *Line Protocol Composer* allows to create the required text based specific syntax format for writing points to InfluxDB databases.

A point in InfluxDB represents a single data record. A point has a measurement, tag set (collection of tag keys and tag values on a point), field set (collection of field keys and field values on a point), and a timestamp.

Please see the end user documentation for more details.

The *Line Protocol Composer* dialog has two tab displays.

The *Line Protocol Composer* dialog's Definition tab allows to define the required components for the line protocol format.



For InfluxDB, the measurement name parameter provides the description of the data that are stored in the associated database fields.

© Softing Industrial Automation GmbH

How the measurement name parameter is provided, depends by the type of the measurement schema:

- *explicit type* - the retrieved measurement values are populated into combobox fields and selectable.
- *implicit type* - existing retrieved measurement values are populated into combobox fields and are selectable, while new measurement values need to be provided in the editable combobox field.

The timestamp parameter is necessary to provide the settings for the date and time associated with a point.

The following settings are available:

- *InfluxDB Host* - The timestamp is not included in the Line Protocol, the InfluxDB will use its own timestamp.
- *dataFEED OPC Suite Host* - The dataFEED OPC Suite local time is used and included in the Line Protocol.

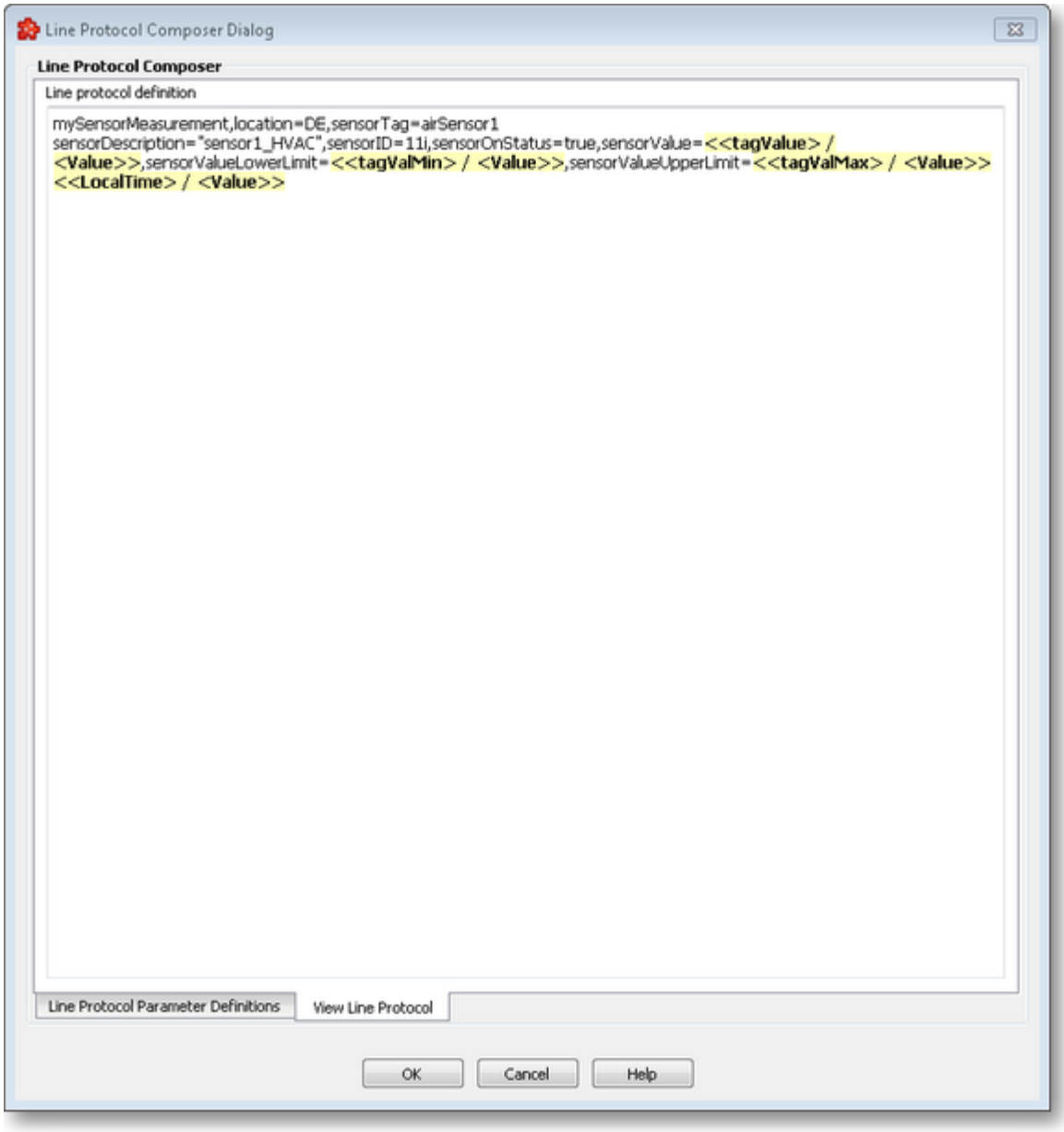
The tag set and field set is created by defining each individual tag / field key-value pair.

By using the *Add*  button, the user can create a new key-value pair definition. The user will be prompted with the *Tag* / *Field Parameter Properties* dialog.

With the *Properties*  button, the individual parameter key-value pair definitions can be modified.

For more information refer to the Tag Field Parameter Properties page.

The *Line Protocol Composer* dialog's overview tab displays the resulted line protocol format.

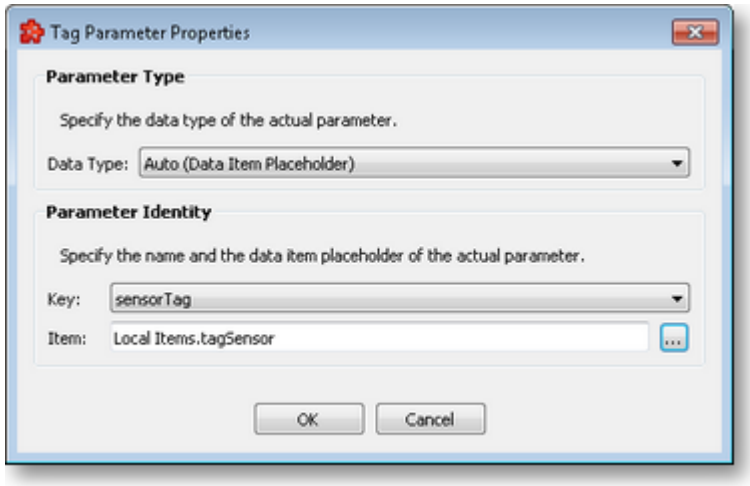


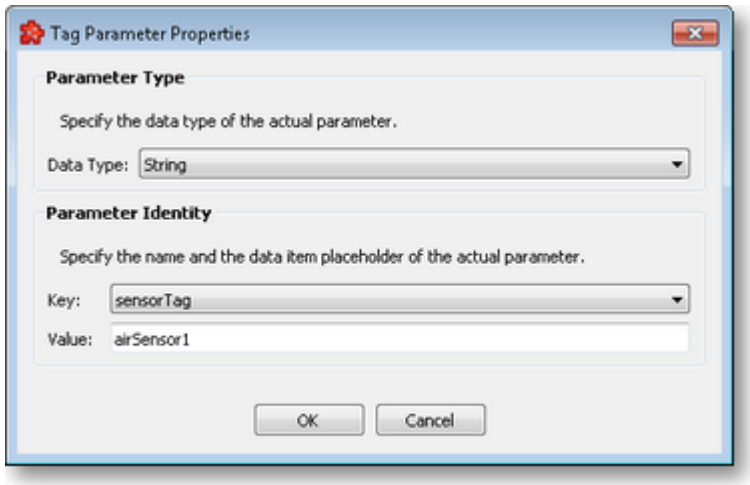
<<< NoSQL Action Wizard - Action Definition

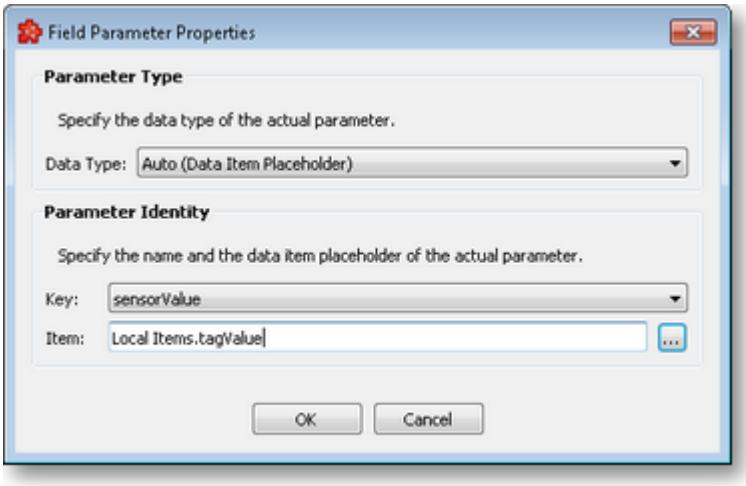
7.3.3.7.8 Tag Field Parameter Properties

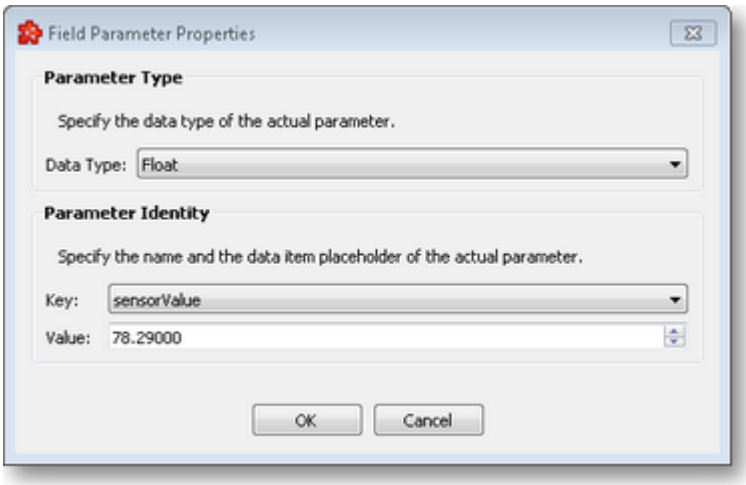
The *Tag Parameter Properties* and *Field Parameter Properties* window offers the possibility to create or modify a tag / field key-value pairs, needed for the tag / field set definitions.

© Softing Industrial Automation GmbH

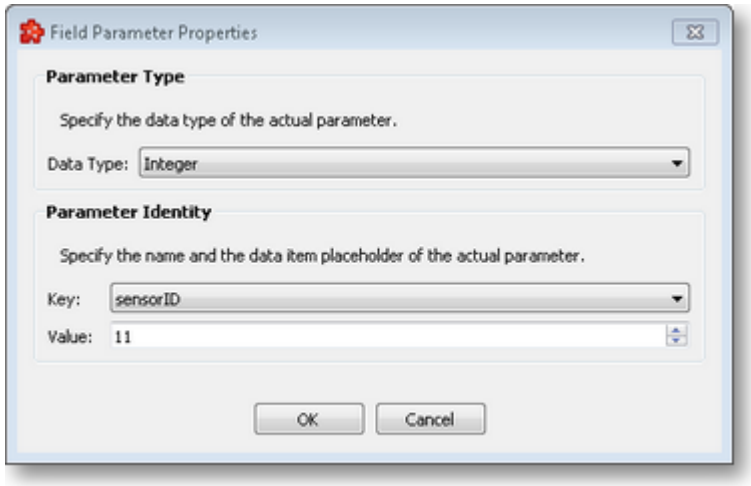

Tag Parameter Properties
Parameter Type
Specify the data type of the actual parameter.
Data Type:
Auto (Data Item Placeholder)
Parameter Identity
Specify the name and the data item placeholder of the actual parameter.
Key:
sensorTag
Item:
Local Items.tagSensor
OK
Cancel

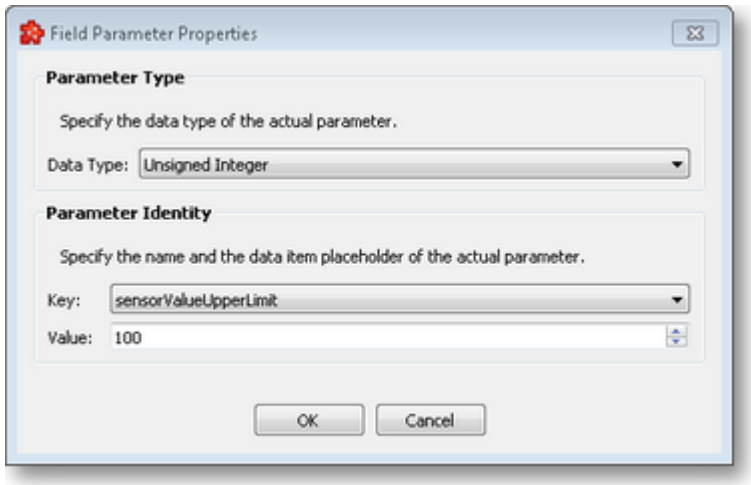

Tag Parameter Properties
Parameter Type
Specify the data type of the actual parameter.
Data Type:
String
Parameter Identity
Specify the name and the data item placeholder of the actual parameter.
Key:
sensorTag
Value:
airSensor1
OK
Cancel

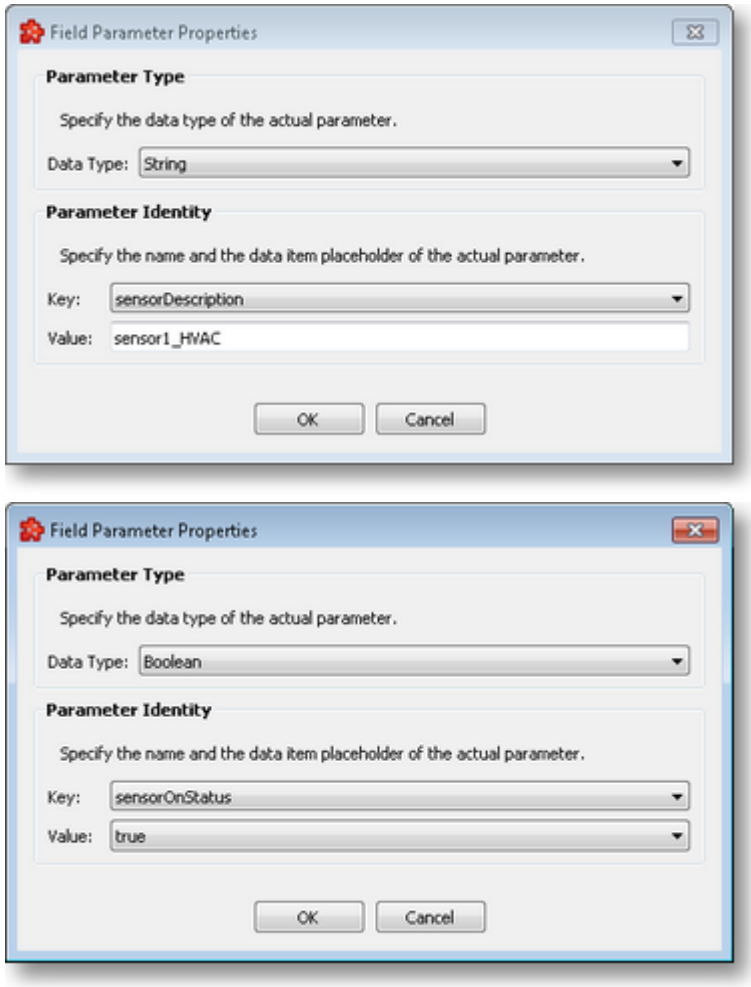




© Softing Industrial Automation GmbH


Field Parameter Properties
Parameter Type
Specify the data type of the actual parameter.
Data Type: Integer
Parameter Identity
Specify the name and the data item placeholder of the actual parameter.
Key: sensorID
Value: 11
OK
Cancel


Field Parameter Properties
Parameter Type
Specify the data type of the actual parameter.
Data Type: Unsigned Integer
Parameter Identity
Specify the name and the data item placeholder of the actual parameter.
Key: sensorValueUpperLimit
Value: 100
OK
Cancel



The following tag / field parameter properties can be selected in this window:

- *Data type* - defines the data type of the parameter value
  - for tag parameters it can be Auto (Data Item Placeholder) or String;
  - for field parameters it can be Auto (Data Item Placeholder) or Float, Integer, Unsigned Integer, String, Boolean;
- *Key* - statically defined string data value;
- *Value / Item* - statically defined Float, Integer, Unsigned Integer, String, Boolean data values, or item value of a data placeholder corresponding to address space items.

The definition of the tag / field key depends by the type of the measurement schema:

- *explicit type* - the retrieved values of the keys are populated into combobox fields and are selectable.
- *implicit type* - existing retrieved key values are populated into combobox fields and are selectable, while new values for the keys need to be provided in the editable combobox field.

© Softing Industrial Automation GmbH

Data Types

The supported data types are:

- *Auto (Data Item Placeholder)* - is an automatic type and depends on the placeholder definition;
- *Float* - double precision floating-point format;
- *Integer* - signed 64-bit integers;
- *Unsigned Integer* - unsigned 64-bit integers;
- *String* - double-quoted *Unicode* with backslash escape;
- *Boolean* - true or false.

Note: The default tag parameter data type is *String*. The default field parameter data type is *Auto (Data Item Placeholder)*.

<<< NoSQL Database - Line Protocol Composer

7.3.3.7.9 Data Values Placeholder

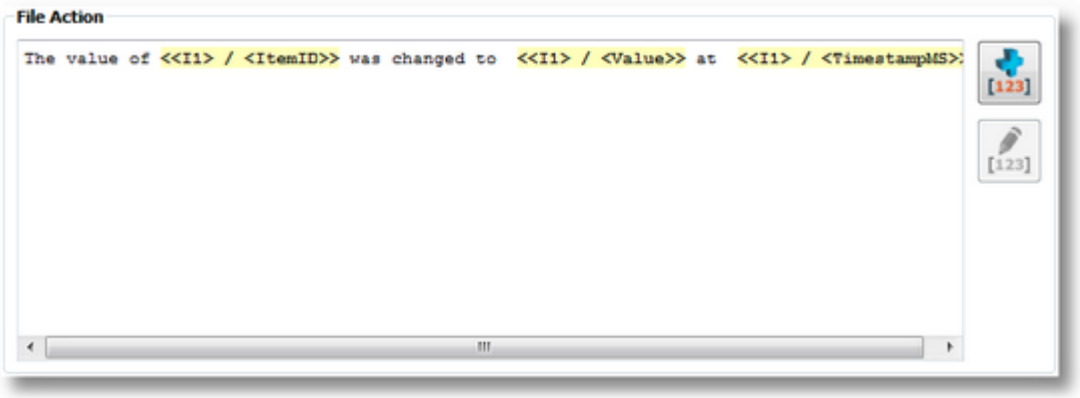
Placeholders can be used when configuring various actions (SQL, TEXT, NOSQL_REQ and other statements). These placeholders can be defined in the *Action Wizard,* the main user interface and are saved in the Subscription file. A placeholder is determined by the following attributes:

- Data Item - an item (tag, property or node with value) originating from a *Server Connection*, from the *Local Items or* from the *System Items*.
- Data Value - the actual data value for the associated data item which will be replaced during the run time of the local dataFEED OPC Suite application.
- Data Source - the source for the data value which will be replaced during the run time of the local dataFEED OPC Suite application.

Placeholder in the user interface

Two types of placeholder editors are implemented in the Softing Configurator. Only one type of action editor is used in the NoSQL add-on, that is the *Text Action Editor*. This placeholder editor is found in NoSQL Action wizard page and also in the Configuration: Data Destination - NoSQL Server main user interface page of the File add-on; the corresponding placeholder form is: <<Short Data Value> / <Short Data Item>>:

```
<<R8> / <ItemID>>
<<R8> / <Value>>
<<R8> / <QualityString>>
<<R8> / <TimestampMS>>
```



Placeholder in the subscription file

In the subscription file a placeholder has the following syntax:

```
#%-<Data Item>%<ItemID_SubscriptionID>%<DataSource>-%#
```

where:

Attribute	Description
#%- -%#	Represents the beginning and the end of a placeholder definition
Data Item	Represents the item data value.
%	Separates the Data Value, Data Item and Data Source fields in the placeholder.
ItemID_SubscriptionID	Represents the subscription item identifier.
Data source	Represents the item data source.

Example:

```
#%-ItemValue%OPC_1.increment.R8_Subscription_222%Current-%#
```

Data value:

The following data value attributes can be used (they are supported by both dataFEED OPC Suite Configurator and local dataFEED OPC Suite application):

Attribute	Subscription file form	Description
Data Value	ItemValue	The value of the data source server item.

© Softing Industrial Automation GmbH

Item Identified	ItemItemID	The ID of the data source server item.
Quality as Integer	ItemQuality	The data source server item quality as integer.
Quality as String	ItemQualityString	The data source server item quality as string.
Timestamp (without milliseconds)	ItemTimestamp	The data source server item timestamp as date time without milliseconds.
Timestamp (with milliseconds)	ItemTimestampMS	The data source server item timestamp as date time including milliseconds.
Timestamp (milliseconds part only)	ItemTimestampMSPart	The milliseconds part of the data source server item timestamp.
Internal Timestamp (without milliseconds)	ItemTimestampOCT	The timestamp of data change in the local dataFEED OPC Suite application as date time without milliseconds.
Internal Timestamp (with milliseconds)	ItemTimestampOCTMS	The timestamp of data change in the local dataFEED OPC Suite application as date time including milliseconds.
Internal Timestamp (milliseconds part only)	ItemTimestampOCTMSPart	The milliseconds part of the data change timestamp in the local dataFEED OPC Suite application.

The following data value attributes can be used (only supported by the local dataFEED OPC Suite application):

Attribute	**Subscription file form**	**Description**
Item Subscription ID	ItemSubscriptionID	The subscription ID of the item
Data Value	DataValue	The ID of the data source server item.
Data Quality as Integer	DataQuality	The data source server item quality as integer.
Data Quality as String	DataQualityString	The data source server item quality as string.
Timestamp (without milliseconds)	DataTimestamp	The data source server item timestamp as date time without milliseconds.
Timestamp (with milliseconds)	DataTimestampMS	The data source server item timestamp as date time including milliseconds.
Internal Timestamp (without milliseconds)	DataTimestampOCT	The timestamp of data change in the local dataFEED OPC Suite application as date time without milliseconds.

Internal Timestamp (with milliseconds)	DataTimestampOCTMS	The timestamp of data change in the local dataFEED OPC Suite application as date time including milliseconds.
Data Item ID	DataItemID	The value of the data source server item.

Attention: There is a significant difference between the Data Values in the tables above. The values in the second table are taken directly from dataFEED OPC Suite internal server while the values for the Data Values from the first table are taken directly from the server which holds the real Item. dataFEED OPC Suite holds only a copy of that Item.

Data Source or Item type:

One of the following Data Sources may be selected:

Item type	Description
Current	The item must be in the subscription item list. The value used is the current one, which means that if the value for the item is included in the current transaction, this value will be used. If the value is not included in the incoming transaction, the cached value will be used.
Cache	The item must be in the subscription item list. The value used is the cached one, even if that item is included in the incoming transaction.
Changed	The item must be in the subscription item list. The value used is the changed one, which means that the entire action will be invalidated if the item is not present in the incoming transaction.

Examples:

```
'#%-ItemTimestampOCTMSPart%OPC_1.increment.R8_Subscription_222%Current-%#'
'#%-ItemTimestampOCTMS%OPC_1.increment.R8_Subscription_222%Changed-%#'
'#%-ItemTimestampOCT%OPC_1.increment.R8_Subscription_222%Cache-%#'
```

Attention: If the *Changed* Data Source is used in an action and the corresponding action trigger is a *Data Change* trigger, make sure that a transaction which contains data changes for all the items used in the current action will be generated by the server; otherwise, the entire action will be invalidated if an item has no associated data change in the incoming transaction.

7.3.3.7.10Configuration Files

All the NoSQL configuration settings are stored in two separated files:

- Tasks file - contains general descriptions of the File definitions.

© Softing Industrial Automation GmbH

- Subscriptions file - which contains all the descriptions related to the stored data:
    - what data will be stored.
    - what is the frequency of storing the data.
    - where and when will the data be stored.

Note: The path and file name of these configuration files are fixed and not configurable.

The task and subscription files are stored in the *Tasks* folder under the instance path. For task configuration a text file called *Tasks.txt* and for subscription configuration a text file called *Subscriptions.txt* will be created in the *Tasks* configuration folder. Although these files can be manually edited, using the dataFEED OPC Suite Configurator is advisable.

Note: The recommendation is to use only the dataFEED OPC Suite Configurator to edit these files. This ensures that all the definitions will be valid.

If the configuration is manually edited try loading the configuration in the dataFEED OPC Suite Configurator as all the configuration files are automatically validated at load-time and, if there are any mistakes, an error message will be displayed. The dialog will contain an appropriate description and will point to the first invalid line of the local items configuration file.

Both configurations files are plain text files encoded in UTF-8 (thus providing full UNICODE compatibility) and they have line-oriented structure. The elements of a line are tab-separated. The format of one line is the following:

```
Line type<TAB>Line data
```

Task Configuration File

The following line types will be defined:

Line Type	Description
NOSQL_TIMESTAMPFORMAT	The format of the global time and date values
NOSQL_TIMEFORMAT	The global time zone value
NOSQL_PROXY	Parameters defined for a proxy connection
NOSQL_REQ_PROC	Parameters used for the internal queue and processing settings
NOSQL	Describes the parameters of a server connection.
Commented Lines (#)	Any line that starts with the "#" character is considered a comment line which is ignored during evaluation.

The NOSQL_TIMESTAMPFORMAT line

The NOSQL_TIMESTAMPFORMAT line allows defining parameters needed by the NoSQL add-on to format time and date values, globally:

```
NOSQL_TIMESTAMPFORMAT<TAB>Format
```

Example:

```
NOSQL_TIMESTAMPFORMAT   mm/dd/YYYY HH:MM:SS pp
```

The NOSQL_TIMEFORMAT line

The NOSQL_TIMEFORMAT line allows defining the time zone, globally (utc or local):

```
NOSQL_TIMEFORMAT<TAB>Value
```

Examples:

```
NOSQL_TIMEFORMAT  utc
NOSQL_TIMEFORMAT  local
```

The NOSQL_PROXY line

The NOSQL_PROXY line allows defining all the parameters needed by a proxy connection:

```
NOSQL_PROXY<TAB>Proxy Type<TAB>Host/IP<TAB>Port<TAB>Authentication Type<TAB>User Cren
```

Examples:

```
NOSQL_PROXY HttpProxy   192.168.150.171   3128  UsingProxyAuth    admin C122ED818D7C5
NOSQL_PROXY HttpProxy   192.168.150.171   3128  NotUsingProxyAuth
NOSQL_PROXY NoProxy
```

The NOSQL_REQ_PROC line

The NOSQL_REQ_PROC line allows defining all the parameters needed by the internal request queue, globally:

```
NOSQL_REQ_PROC<TAB>Maximum queue size per connection<TAB>Execution type
```

Maximum queue size refers to the number of allowed request the internal request queue can hold for one connection. Each connection queue will be able to hold that number of requests.

Execution type specifies whether or not the execution of requests should be parallel (currently 6 at a time, because of Qt limitations) or sequential (one at a time).

Examples:

```
NOSQL_REQ_PROC    10000 Parallel
NOSQL_REQ_PROC    100   Sequential
```


© Softing Industrial Automation GmbH

The NOSQL line

The NOSQL line allows defining all the parameters needed by the NOSQL add-on to send data to a NoSQL Database:

```
NOSQL<TAB>Connection ID<TAB>Connection Name<TAB>Status<TAB>Database Type<TAB>Protocol
```

Parameter	Description
NOSQL	The type of the line, predefined keyword.
Connection ID	Unique identifier within the task file (used for matching the statements in the corresponding subscription file).
Connection Name	The name of the connection visible in the Configurator
Status	The status of the connection: Active or Inactive
Protocol Type	TCP, HTTP or HTTPS; MONGODB or MONGODB+SSL
Host Name	The server address, or IP. A '/' character is compulsory at the end
Authentication Type	Anonymous, Username and Password, or Certificate (only for secure connections). The parameters that follow may differ based on selection.
Request Timeout	Value in seconds of the time to wait for a request to be aborted. 0 means waiting until it completes
Store and Forward status	Whether or not Store and Forward is active: Active or Inactive
Store and Forward folder path	The path of the folder where the file will be created in.
Store And Forward file size	Maximum amount of space used for the file
Store And Forward requests size	Maximum number of requests stored in the file
Overwrite requests	Whether or not to overwrite older requests if the storage gets full: ON of OFF
Packed requests	Currently reserved and disabled for NoSQL
Max. number of forwarded requests	How many requests to pack together in one
Time zone	UTC or Local

Date-time format	The format of the time and date items
Autoaccept certificates	TRUE or FALSE
Certificate retrieval interval	Value in seconds of the interval after which the application will validate the server certificate again at action execution

Example:

```
NOSQL NoSQL_01    NoSQL_DB_01 Active      MongoDB     MONGODB_SSL 192.168.150.85:2701
NOSQL NoSQL_02    NoSQL_DB_02 Active      CouchDB     HTTP  127.0.0.1:5984/   Anonymo
```

Task File - Example:

```
#
#     NoSQL - configuration
#
NOSQL_TIMESTAMPFORMAT   mm/dd/YYYY HH:MM:SS pp
NOSQL_TIMEFORMAT  utc
#
NOSQL_PROXY NoProxy
#
NOSQL_REQ_PROC    10000 Parallel
#
NOSQL NoSQL_01    NoSQL_DB_01 Active      MongoDB     MONGODB_SSL 192.168.150.85:2701
#
NOSQL NoSQL_02    NoSQL_DB_02 Active      CouchDB     HTTP  127.0.0.1:5984/   Anonymo
```

Subscription Configuration File

The *Data Destination - NoSQL Database* generates multiple line types:

- common lines which are specific to all the consumer add-ons: condition lines, subscription lines and item lines;
- specific lines: modifier lines and copy lines.

Common Lines:

The following line types are common lines and will be used to configure different functionalities:

Line Type	Description
Subscription	Defines the update rate with which the current action will be executed.
Condition	Defines the execution condition for the current action.

© Softing Industrial Automation GmbH

Item	This line defines a trigger which is fired if the item data has changed. The trigger data includes all other items which have changed at the same time (were reported in the same data change callback from the OPC server). An item data change consists of a different value or quality of the respective item compared to the previous one.
Timer	This line defines a trigger which is fired at specific time and date or at specific intervals.
Commented Lines (#)	Any line that starts with the "#" character is considered a comment line which is ignored during evaluation.

The subscription file is divided into two sections:

- The global area:

  The global area extends from the first subscription line to the first trigger line (e.g: ITEM, TIMER lines).

  This area contains definitions valid for the entire subscription file.

- The local area:

  The local area covers all the actions between two trigger lines in the subscription file.

  The lines in this area are only executed if the parent trigger fires.

Subscription Line

A subscription line has the following format:

```
SUBSCRIPTION<TAB>Subscription ID<TAB>UPDATERATE<TAB>Update rate value<TAB>ACTIVE<TAB>
```

The subscription line contains the following configurable parameters:

Parameter	Description
SUBSCRIPTION	The type of the line.
Subscription ID	It is used to identify the subscription and must be unique.
UPDATERATE	Predefined keyword.
Update rate value	Defines the update rate with which the current action will be executed.
ACTIVE	Predefined keyword.

Active value	Active value which can be TRUE or FALSE.

The subscription lines always contain the global subscription line as well as all subscriptions which are used at least once in consumer actions. The global subscription is represented by the subscription line with the empty *Subscription ID*. The default update rate value for the global subscription is *1000* milliseconds.

The value of the update rate is saved in milliseconds. This value may not be set to 0 inside the dataFEED OPC Suite Configurator but it can be manually set by editing the subscription file with an external text editor. An update rate value of 0 means *as fast as possible* but this is not recommended from a resource-related point of view.

Setting a subscription to be active or inactive is an advanced feature which cannot be configured from within the dataFEED OPC Suite Configurator. By default the active value for each and every subscription is set to *TRUE* (meaning that all subscriptions are active since creation). If a subscription's active value is set to *FALSE* then the data changes of the items configured to be part of that subscription will be ignored.

Examples:

```
SUBSCRIPTION<TAB><TAB>UPDATERATE<TAB>1000<TAB>ACTIVE<TAB>TRUE
SUBSCRIPTION<TAB>Subscription_300<TAB>UPDATERATE<TAB>300<TAB>ACTIVE<TAB>TRUE
```

Condition Line

A condition line defines whether an action below an item trigger will be executed or not (when the item trigger is fired). This can be used in other configuration lines as well to condition the execution of an item action. One example of this is the conditional execution of a SQL statement in the Database add-on. A condition line has the following format:

```
CONDITION<TAB>Condition ID<TAB>Condition Statement
```

The condition line contains the following configurable parameters:

Parameter	**Description**
CONDITION	The type of the line, predefined keyword.
Condition ID	Unique condition identifier.
Condition statement	Single condition statement.

Conditions can be used to react to communication errors. Various actions can be executed depending on the quality of items.

The following condition statements are supported:

Condition	**Description**

© Softing Industrial Automation GmbH

AllGoodQuality	The trigger actions are executed only if all items in the transaction have the quality *Good*.
Always	The trigger actions are executed always, ignoring the quality of the items in the incoming transaction.
ItemQualityChanged	The trigger actions are executed only if the quality of at least one item in the transaction has changed.
OneBadQuality	The trigger actions are executed if exactly one item in the transaction has the quality *Bad*.

The condition lines always contain the global condition line as well as all conditions which are used at least once in consumer actions. The global condition is represented by the condition line with the empty *Condition ID*. The default condition statement value for the global condition is *Always*. As a general rule all the condition lines need to be declared in the global section of the subscription file. If no condition is defined the Always condition is applied as default.

Examples:

```
CONDITION<TAB><TAB>Always
CONDITION<TAB>AllGoodQuality<TAB>AllGoodQuality
```

Item Line

An item line is used to add an item to a subscription. The associated actions are executed each time the value or the quality of the specified data items changes during the run time of the local dataFEED OPC Suite application.

An item line has the following format:

```
ITEM  Local Items.Scalars.Always_Trigger_ Local Items.Scalars.Always_Trigger
```

The item line contains the following configurable parameters:

Parameter	**Description**
ITEM	The type of the line, predefined keyword.
Subscription Item ID	Unique subscription item identifier (implicitly created by concatenating the Server Item ID and the Subscription ID with the underscore character in between them).
Server Item ID	Item identifier within the local dataFEED OPC Suite application address namespace. This could be a server connection item ID or a local item ID.

Subscription ID	Subscription identifier (defined above, see Subscription Line).

This trigger is fired if the item data or quality has changed (also referred to as data change). The trigger data includes all other items which were included in the data change (items which had their data change reported in the same transaction).

Note: A single line consisting of the *Server Item ID* and *Subscription ID* is added to the subscription file for an item used several times with the same update rate in any consumer add-ons. This optimizes the local dataFEED OPC Suite application throughput.

Example:

```
ITEM<TAB>increment.I1_Subscription_300<TAB>increment.I1<TAB>Subscription_300
```

Note: The subscription item ID can be empty. In this case, however, only the Server Item ID can be used to identify the item.

Timer Line

This trigger is fired at a specified time and date or at specific intervals. The associated actions are executed on a specific schedule or periodically during the run time of the local dataFEED OPC Suite application.

The format of the *Timer Line* is:

```
TIMER<TAB>Timer ID<TAB>INTERVAL<TAB>Rate in milliseconds
```

or

```
TIMER<TAB>Timer ID<TAB>SCHEDULE<TAB>Schedule range<TAB>First execution<TAB>Rate<TAB>L
```

The interval timer contains the following configurable parameters:

Parameter	Description
TIMER	The type of the line, predefined keyword.
Timer ID	Unique timer identifier.
INTERVAL	Denotes an interval timer trigger, predefined keyword.
Interval value	Defines the interval in milliseconds at which the timer will trigger the associated action(s).

The timer schedule contains the following configurable parameters:

© Softing Industrial Automation GmbH

Parameter	Description
TIMER	The type of the line, predefined keyword
Timer ID	Unique timer identifier.
SCHEDULE	Denotes a scheduled timer trigger, predefined keyword.
Schedule range	Defines the range of the schedule (see below the table).
First execution	Defines the time the timer should begin firing events (see below the table).
Rate	Defines the rate at which the timer should trigger the associated action(s).
Last execution	Defines the time the timer should stop firing events (see below the table).
List Of Days	Should be used only with the Week and Month schedule. It must contain the days on which the execution is desired. If the list of days is not present, the obtained functionality is the same as by using the day marker in every day of the week/month.

The timer's *Schedule range* may have one of the following values:

- MINUTE
- HOUR
- DAY
- WEEK - List of days: 1d = Monday, 2d = Tuesday, …, 7d = Sunday.
- MONTH - List of days: 1d = First day of month, 2d = Second day of month, …, 30d = Thirtieth day of month, 31d = Thirty-first day of the month.

The *First execution* defines the time the timer should begin firing events.

Example:

```
1d13h10m15s
```

Note: If parts of the execution time are 0, they do not have to be in the string, e.g. 1d0h10m0s = 1d10m

The timer *Rate* defines the event firing rate. An empty timer *Rate* means a single execution.

Last Execution: The last execution time of the schedule. Could be empty.

List of Days: Shall be used only with the Week and Month schedule. It should contain the days on which execution is desired. If the list is not specified the same functionality can be achieved by using the day marker in the first and last execution time.

Note: If the List of Days and the day marker in first and last execution time are used in the same timer trigger the two setting types should match the marked days!

Examples:

```
# Following timer fires events every 250 milliseconds (4 times a second):
TIMER<TAB><TAB>INTERVAL<TAB>250
# Following timer fires events every 20 seconds of a minute starting at second 10 (eg
TIMER<TAB><TAB>SCHEDULE<TAB>MINUTE<TAB>10s<TAB>20s
# Following timer fires events every 4 hours of a day starting at 6:30, ending at 20:
# (e.g. 6:30, 10:30, 14:30, 18:30).
TIMER<TAB><TAB>SCHEDULE<TAB>DAY<TAB>6h30m<TAB>4h<TAB>20h
# Following timer fires events every 4 hours of a day starting at 6:30, ending at 20:
# Friday (e.g. 6:30, 10:30, 14:30, 18:30 of each day).
TIMER<TAB><TAB>SCHEDULE<TAB>WEEK<TAB>6h30m<TAB>4h<TAB>20h<TAB>1d2d3d4d5d
# Following timer fires events every 4 hours starting at 6:30, ending at 22:45 (e.g.
# 18:30, 22:30), on the 1th, 2th, 10th, 20th of every month.
TIMER<TAB><TAB>SCHEDULE<TAB>MONTH<TAB>6h30m<TAB>4h<TAB>22h45m<TAB>1d2d10d20d
# Following timer fires events every 1 hours, 3 minutes and 3 seconds starting at 00:
# 23:59:59 (e.g. 00:00:01, 01:03:04, 02:06:07 etc), on the 3rd, 6th, 9th, 12th, 15,th
TIMER<TAB>Timer_02<TAB>SCHEDULE<TAB>MONTH<TAB>1s<TAB>1h3m3s<TAB>23h59m59s<TAB>3d6d9d1
```

Note: Make sure that only used subscription definitions are present in the configuration file (resource-related point of view).

Specific Lines:

The following line types are REST Server specific and will be used to configure different functionalities:

Line Type	Description
NOSQL_REQ	Defines a NoSQL Database action statement.

The format of the NOSQL_REQ line is the following:

CouchDb:

```
NOSQL_REQ<TAB>Condition ID<TAB>Connection ID<TAB>Action Name<TAB>Action Status<TAB>Da
```

MongoDb:

```
NOSQL_REQ<TAB>Condition ID<TAB>Connection ID<TAB>Action Name<TAB>Action
Status<TAB>Database Name<TAB>Document Name<TAB>Body<TAB>Content
Type<TAB>Trigger
```


© Softing Industrial Automation GmbH

Examples:

CouchDb:

```
NOSQL_REQ         NoSQL_02    Action_02   Active      aaaa        POST  {}
      {"clasic": [{"action": "insert_action_2","big_text": "Whenever a method
overrides another method, or a method implements a signature in an interface.
The @Override annotation assures you that you did in fact override something.
Without the annotation you risk a misspelling or a difference in parameter
types    and    number","id":    "#%-ItemItemID%OPC_1.increment.I8_%Current-%
#","timestamp":   "#%-ItemTimestampMS%OPC_1.increment.I8_%Current-%#","value":
"#%-ItemValue%OPC_1.increment.I8_%Current-%#"}]}      application/json  Local
Items.Arrays.i4_
```

MongoDb:

```
NOSQL_REQ         NoSQL_01    Action_01   Active      demodb
democollection    {"test_key":  "#%-ItemValue%OPC_1.increment.I1_%Current-%#"}
      Local Items.Arrays.i1_
```

The NOSQL_REQ line contains the following configurable parameters:

Parameter	**Description**
NOSQL_REQ	The type of the line, predefined keyword.
Condition ID	Condition identifier (defined above, see Condition Line).
Connection ID	NoSQL connection ID
Action name	The name of the action visible in the Configurator
Action Status	Specifies whether or not the action will be executed: Active or Inactive
Database name	The name of the database
Document name	The name of the document
HTTP method	POST or PUT - only available for databases with REST interfaces
Header	The serialized form of the action header - only available for databases with REST interfaces
Body	The serialized form of the action body
Content Type	The HTTP content type of the body - only available for databases with REST interfaces

Trigger ID	Associated trigger identifier (one of the above defined: Item Trigger, Timer Trigger)

The action body parameter configures the text which will be sent to the NoSQL server when the corresponding action is executed.

Example of NoSQL database configuration lines and the appropriate ITEM lines:

```
ITEM  Local Items.Arrays.i1_  Local Items.Arrays.i1              I1[]
NOSQL_REQ         NoSQL_01    Action_01   Active      demodb      democollection    {

ITEM  Local Items.Arrays.i4_  Local Items.Arrays.i4              I4[]
NOSQL_REQ         NoSQL_02    Action_02   Active      aaaa        POST  {}    {"clasi
```

In order to offer a flexible configuration for NOSQL_REQ statements, placeholders can also be used. Placeholders are syntactical entities that are replaced with their associated values each time the statement is evaluated during the run time of the local dataFEED OPC Suite application (before the request is sent to the server).

Note: The *Condition ID* used in NOSQL_REQ line definition must be defined in the current Subscription File; the default condition may also be used.

Note: The *Connections ID* used in a NOSQL_REQ line definition must be defined in the associated Task File.

The NOSQL_REQ actions will be executed on every data change event, however their execution will also depend on both the defined conditions and the type of the fired trigger.

If the actions are associated with triggers of type *Item*, the respective actions are evaluated on each item change, but are only executed if all of the following conditions are true:

- The subscription item ID of the action has to be defined in the configuration.
- A change occurs for the respective item.
- The action condition is true.
- The placeholders in the action can be replaced by their respective values (the changed data source may invalidate the action).

If the actions are associated with triggers of type *Timer*, the respective actions are evaluated and executed only if all of the following conditions are true:

- The subscription item ID of the action has to be defined in the configuration.
- The action condition is true.
- The placeholders in the action can be replaced by their respective values (the changed data source may invalidate the action).

Subscription File - Example:

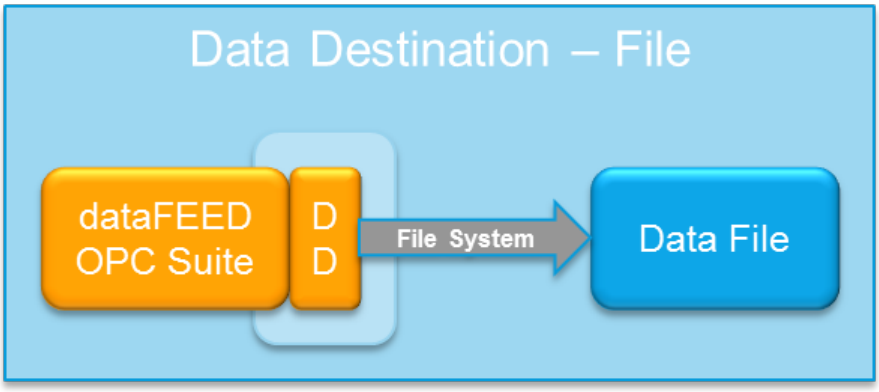
```
CONDITION         AllGoodQuality
```


© Softing Industrial Automation GmbH

```
#
SUBSCRIPTION            UPDATERATE  1000  ACTIVE      TRUE
#
MODIFIER          V
#
#
ITEM  Local Items.Arrays.i1_  Local Items.Arrays.i1              I1[]
NOSQL_REQ         NoSQL_01    Action_01   Active      demodb      democollection    {
#
ITEM  Local Items.Arrays.i4_  Local Items.Arrays.i4              I4[]
NOSQL_REQ         NoSQL_02    Action_02   Active      aaaa        POST  {}    {"clasi
#
ITEM  OPC_1.increment.I1_     OPC_1.increment.I1                I1
#
ITEM  OPC_1.increment.I8_     OPC_1.increment.I8                I8
```

7.3.3.8 File

This section of the Softing dataFEED OPC Suite Online Help describes how to configure the file data destination functionality of Softing dataFEED OPC Suite.



OPC2File

OPC to File stores OPC data from any *Data Access* or *XML DA* server in text files (.txt, .xml or .html). The data read by Softing dataFEED OPC Suite is archived in files via simple TEXT statements.

The OPC2File has the following functionalities:

- Transfer of OPC data from any DA or XML DA server to a file.
- Output of information to plain text files, xml or html files.
- Comes with its own Web server diagnostic interface.

The OPC File add-on is the product name of the Softing's file writer add-on for dataFEED OPC Suite, therefore the add-on library will have the name OSF_AO_File.dll and will be present in the installation folder.

The dataFEED OPC Suite *OPC2File* add-on can be configured using the configuration files (tab separated values) and the XML configuration elements. The respective configuration files can be manually created and edited or they can be managed with the help of by dataFEED OPC Suite Configurator.

Note: This add-on retrieves the values of items from OPC servers originating from server connections and writes them to a text file using standard streaming (currently the plain text, xml and html syntaxes are supported). For each data change received the file task performs the configured text statements.

The following item information can be serialized:

- the item value
- the item quality as integer or as readable string.
- the item timestamp as date time with or without milliseconds
- the data change timestamp as date time with or without milliseconds
- the internal subscription item ID
- the item ID

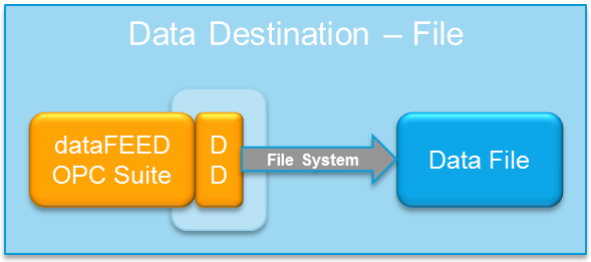
All this information can be configured from the dataFEED OPC Suite Configurator, yet advanced information carried for each item received in a data change is only configurable through the subscription file (manually editing this file):

- the internal item value.
- the internal item quality as integer or as readable string.
- the internal item timestamp with or without milliseconds.
- the internal data change timestamp as date time with or without milliseconds
- the internal item ID.

Specific details about how to create and handle the OPC2File definitions can be found on the Configuration: Data Destination - File page.

A detailed configuration file format description can be found on the Configuration Files page.

7.3.3.8.1 Configuration

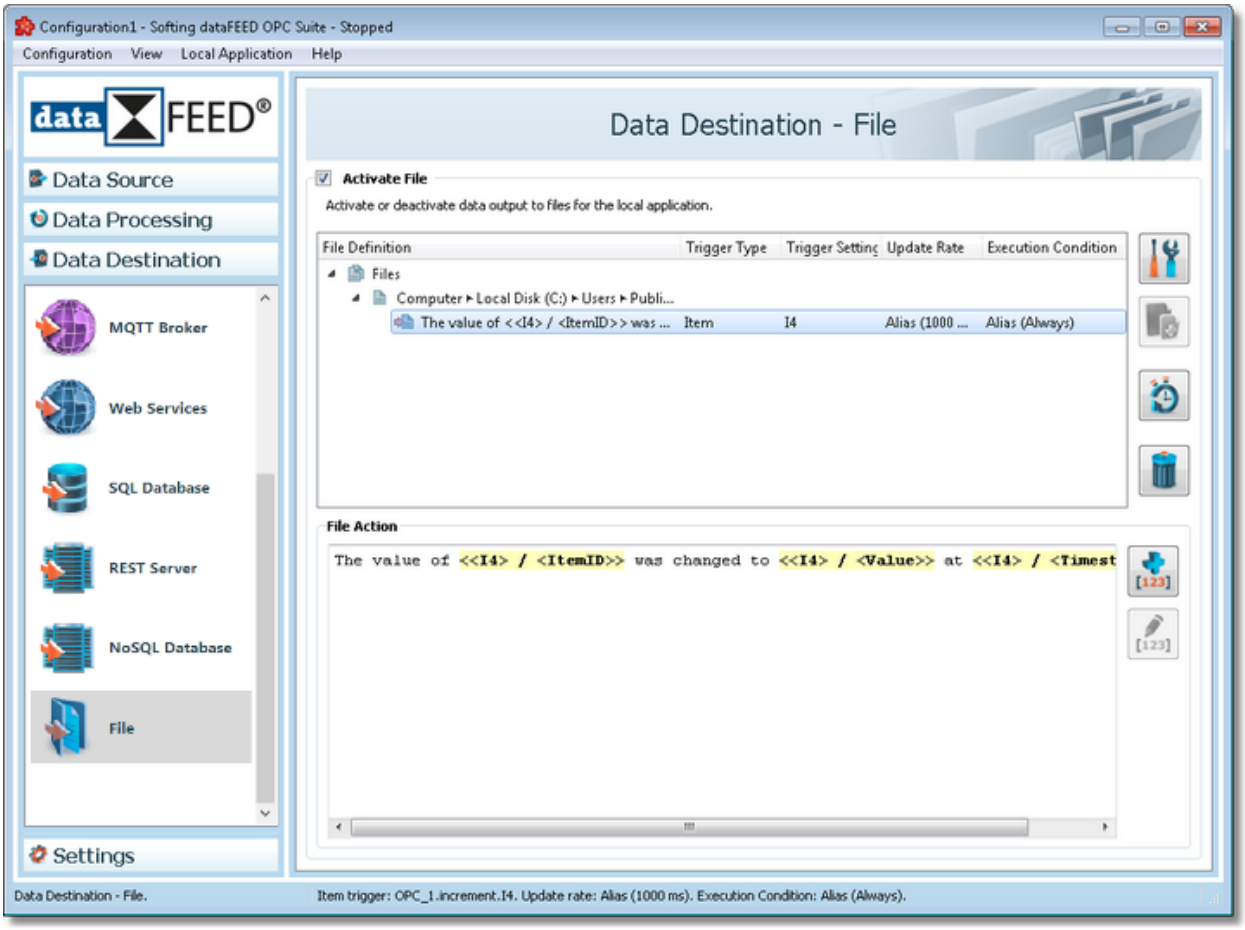


Main user interface

© Softing Industrial Automation GmbH

Configuration of the OPC2File functionality is done in the Data Destination - File main user interface.The main add-on UI page can be reached by choosing the *File* option from the *Data Destination* section in the left side navigation panel.

The below image illustrates the main user interface of the File add-on:



For more details about the OPC2File functionalities see the File page.

The entire *File* add-on functionality can be activated or deactivated by selecting respectively deselecting the *Activate File* checkbox.

Note: If file definitions or file actions are defined and the add-on functionality is deactivated, this definitions will not be deleted from the configuration files (task and subscription files). Instead they are marked as deactivated - prefixed with the deactivated mark: *#!* - which lines are ignored by the local dataFEED OPC Suite application.

By default the OPC2File functionality is disabled and it must be first enabled by selecting the *Activate File* check-box at the top of the *Data Destination - File* main UI page in order to use it. When this add-on is enabled the File Wizard will be shown.

The *Data Destination - File* main view consists of:

- The *Activate File* checkbox which allows enabling / disabling this add-on.
- A tree displaying the file definitions and file actions related to them.

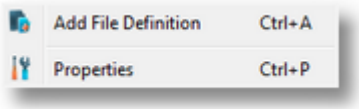
- An action editor which offers support in editing the selected file action.

File definition handling

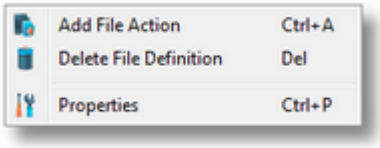
From the Data Destination - File main user interface, certain operations can be performed using the buttons on the right, the keyboard shortcuts or the corresponding context menu actions:

- Change the properties of the selected file definition using the *Properties* button  , pressing the keyboard shortcut CTRL+P or selecting the *Properties* action from the context menu.
- Change the properties of the selected file action using the *Properties* button  , pressing the keyboard shortcut CTRL+P or selecting the *Properties* action from the context menu.
- Add a new file definition selecting the Files root element and using the *Add* button  , pressing the keyboard shortcut CTRL+A or selecting the *Add* action from the root element context menu.
- Add a new action for the selected file definition using the *Add* button  , pressing the keyboard shortcut CTRL+A or selecting the *Add* action from the file definition context menu.
- Remove the selected file definition / file action using the *Delete* button  , pressing the DELETE / DEL keyboard key or selecting the *Delete* action from the context menu.
- Modify the trigger type for the selected file action selecting the *Modify Trigger Type* action from the context menu.
- Modify the trigger settings for the selected file action selecting the *Modify Trigger Settings* action from the context menu. If the trigger type is Datachange this context menu action is disabled. For more information see the Item Trigger wizard or the Timer Trigger wizard page.
- Specify the global alias values for the entire configuration using the *Configure global alias and initial values* button  or pressing the keyboard shortcut CTRL+D.

The *Files* root node's context menu can be seen below:

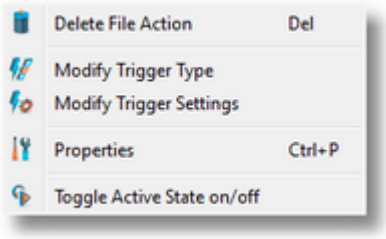


The *File Definition* context menu can be seen below:



The *File Action* context menu can be seen below:

© Softing Industrial Automation GmbH



Each action can be activated or deactivated separately using Toggle Active State on/off option. Inactive actions are grayed out in the tree view.

For file action definitions in the File tree the trigger type, the update rate and the execution condition can be defined. These values can be edited by clicking in the corresponding cell.

Action editor handling

The File Action Editor offers the possibility to edit a selected file action by inserting text lines and placeholders. Unlike the Database Action Editor, the File Action Editor supports the Tab and New Line characters.

For example, the following statement writes the text to the file, replacing the placeholders accordingly:

```
TXT: The value of <<I1> / <ItemID>> was changed to <<I1> / <ItemValue>> at <<I1> / <I
XML: <value ItemID="<<I1> / <ItemID>>" ItemValue="<<I1> / <ItemValue>>" ItemTimestamp
TSV: <<I1> / <ItemID>>#%-TAB-%#<<I1> / <ItemValue>>#%-TAB-%#<<I1> / <ItemTimestampMS>
```

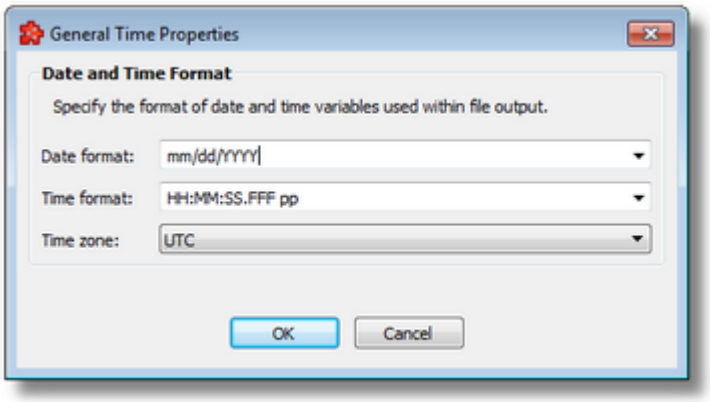
Note: This add-on offers the possibility to include a header file and a footer file at the beginning and end of the output files. The full paths to these two files must be specified or the corresponding text must be provided.

Note: Configuration files are encoded in UTF-8, thus providing full support of Unicode characters.

Time Format

When the *File* root node is selected, invoking the properties action by any means (button, context menu or keyboard shortcut) displays a dialog that allows the user to modify the time format setting. The *General Time Properties* dialog can be opened by using the *Properties* button  , by pressing the keyboard shortcut CTRL+P or by selecting the *Properties* action from the context menu.

Note: The *General Time Properties* dialog will not open unless the *File* root node is selected.



Time format specifies the format in which the item data values will be converted. Also, *Date format*, but for the date part of the item data values.

The Date and Time format settings apply to item data values of data type date and time (DATETIME), and to item data values configured to the data value types: Timestamp (without milliseconds), Timestamp (with milliseconds), Internal Timestamp (without milliseconds), or Internal Timestamp (with milliseconds). For more details on the latter data value attributes, please see the description at Data Values Placeholder.

The default value for the date/time format is "mm/dd/YYYY HH:MM:SS.FFF pp" as seen in the screenshot above. Other defined values for Time and Date format are:

- *Date format*: `mm/dd/YYYY, dd-mm-YYYY, dd.mm.YYYY, dd/mm/YYYY, YYYY-mm-dd, mm-dd-YYYY`
- *Time format*: `HH:MM:SS pp, HH:MM:SS.FFF pp, HH:MM:SS.FFF, HH:MM:SS`

Values for Time and Date format can also be defined manually but in this case special care must be taken to ensure that the definitions are recognized by the local dataFEED OPC Suite application.

In addition to the default values listed in the format selection combo boxes, the user can add his/her own custom format. Custom Date and Time format strings are accepted as valid, when the format strings contain in their composition the following accepted format substrings and format delimiters. These formats allow the inclusion of specific locale information for the regional settings.

- *Valid Date format strings*: `AAAAAA, aaa, BBBBBBBB, bbb, ccccccccccccccccccc, dd, jjj, mm, UU, ww, xxxxxxxxxx, yy, YYYY`
- *Valid Time format strings*: `II, HH, MM, SS, pp, XXXXXXXX, FFFFFF, FFFFF, FFFF, FFF ,FF, F`
- *Valid Date and Time format delimiters*: `<dot>, <colon>, <slash>, <comma>, <backslash>, <vertical bar>, <semicolon>, <space>, <tab>, <minus>`

Note: At least one of the formats need to be specified. If both Time format and Date format values are empty, an error message will be displayed to the user and this dialog will not pass validation.

© Softing Industrial Automation GmbH

Time Zone, handles the source localization time: UTC (Universal Time Coordinated) or local time (local time localization). The default time zone is UTC.

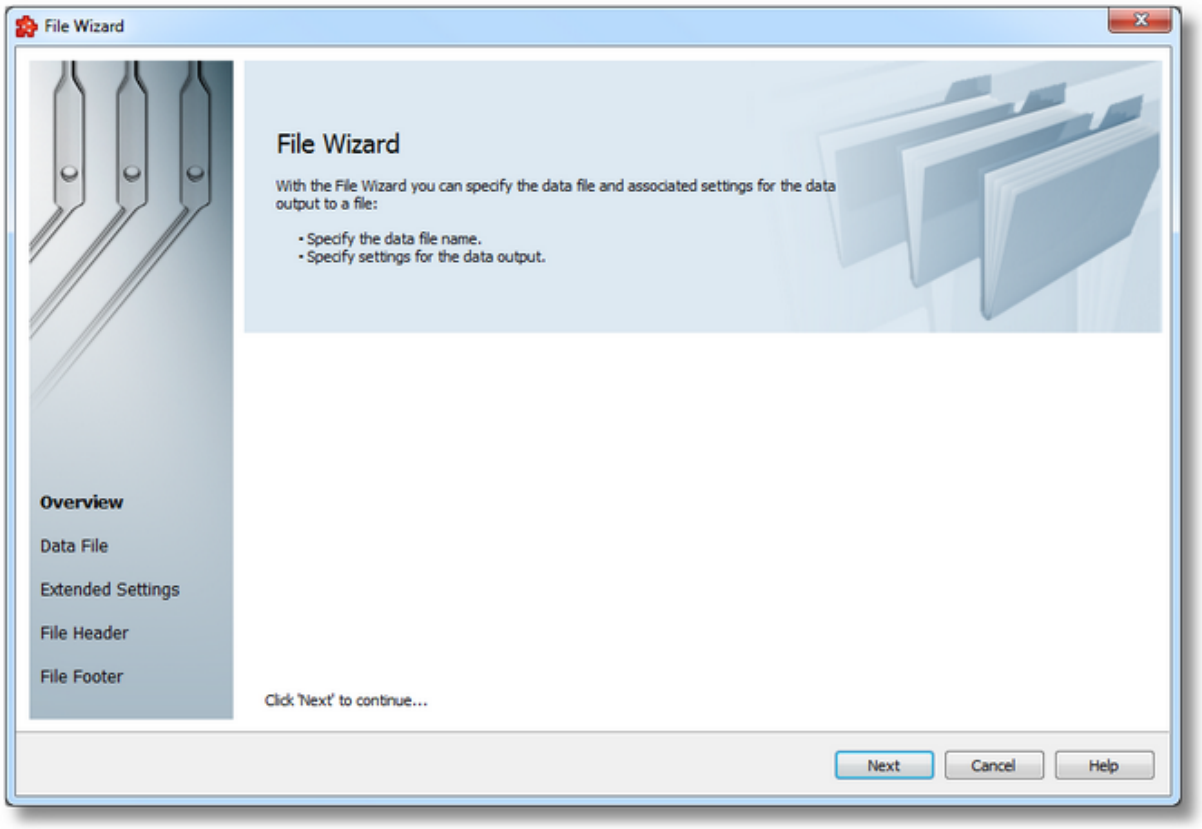
7.3.3.8.2 File Wizard

With the File Wizard you can create new file definitions and modify the existing ones:

- Obtain general information about the File Wizard @ File Wizard - Welcome
- Select the data file and decide whether to use multiple files or not @ File Wizard - Data File
- Define the extended date and time format settings @ File Wizard - Extended Settings
- Define the file header @ File Wizard - File Header
- Define the file footer @ File Wizard - File Footer

The Welcome wizard page describes the basic steps for completing the file definition:

- define file's name and path
- define the file's output rate
- define the file's header
- define the file's footer

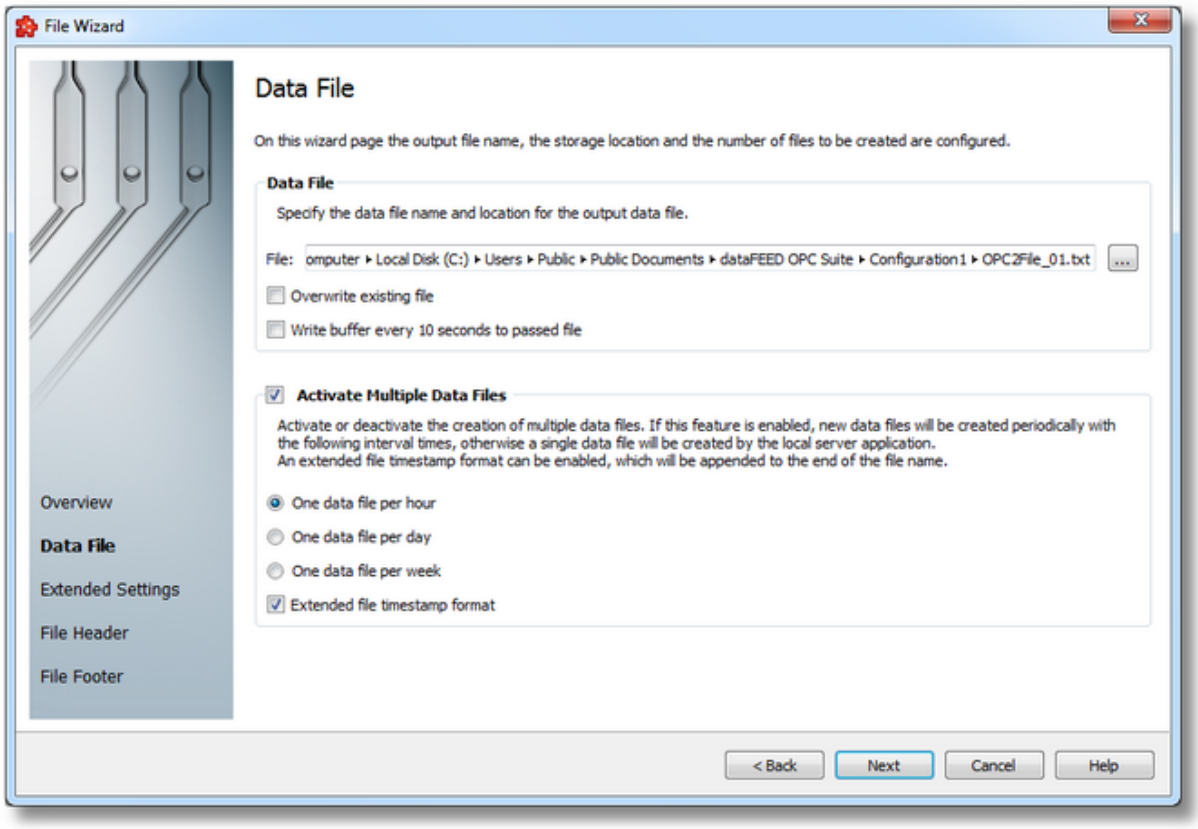


Continue to the next wizard page by pressing the *Next* button, or quit the wizard by clicking the *Cancel* button.

File Wizard - Data File >>>

The Data File wizard page allows you to provide the file path and name for the output data file as well as specify whether multiple data files should be used or not.

© Softing Industrial Automation GmbH



Data File

The path and name of the file can be typed in the provided text box. The data file can also be selected by clicking the browse button located to the right of the *File* text box.

If the typed path does not exist, it will be automatically created by the dataFEED OPC Suite Configurator. If an existing file is chosen with the help of the browse button, a confirmation message will be shown asking whether to overwrite the file or not.

The file will be overwritten on each restart of the application if the "*Overwrite existing file*" check box is enabled. By default the option is inactive and the new data is appended to an existing file or a new file is being created.

The write buffer is flushed to the specified file every time when its size is bigger than 4Kb. By checking the "Commit write buffer to file every 10 seconds" the write buffer is flushed every 10 seconds independent of its size.

Attention: Make sure that the location path is valid and that you have writing rights on it.

Note: The path and the file name are mandatory; the wizard cannot be continued without providing this data.

Multiple Data Files

The *Activate Multiple Data Files* section allows choosing how often new data files should be created:

- hourly - one output file is created each hour.
- daily - one output file is created each day.
- weekly - one output file is created each week.

There is another setting, called "Extended file timestamp format", which makes the Runtime append the _YYYYMMDDHHMMSS timestamp format to the file name, instead of the standard -YYYYMMDD-HH format. The last one is appended when the check box is unchecked.

This functionality can be activated or deactivated by checking / unchecking the *Activate Multiple Data Files* checkbox. When this feature is not active, the entire local dataFEED OPC Suite application output is written to a single file.

Note: Make sure the operating system supports large files when this option is not active or when the created output is large enough.

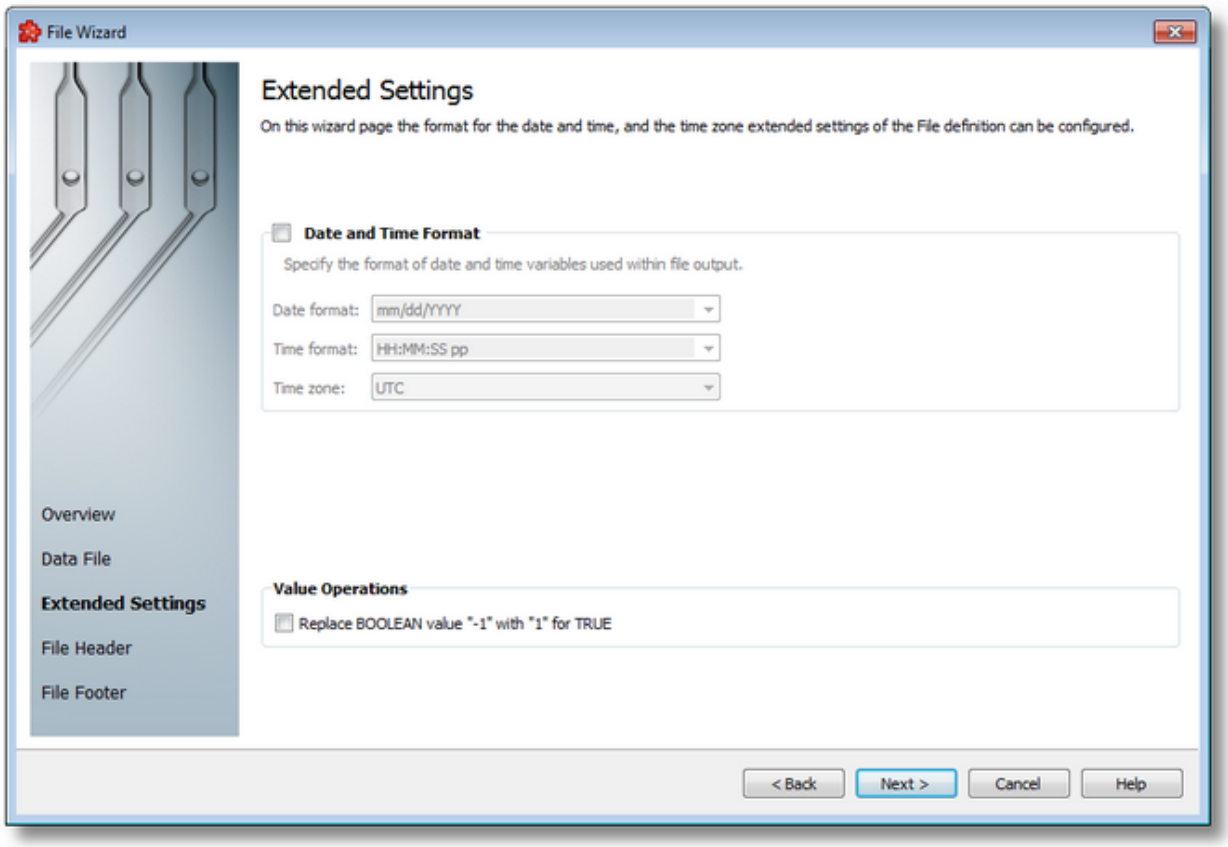
Note: Make sure that there is enough free space available on the target drive to write the output files. If there is not enough free space the OPC2File functionality will not add any data to the corresponding file (when the drive is full).

To proceed with the wizard and continue to define the file's header press the *Next* button. To go to the previous wizard pages use the *Back* button. Exit the wizard by pressing the *Cancel* button.

<<< File Wizard - Welcome File Wizard - Extended Settings >>>

This page allows to specify the extended connection settings such as time and date format.

© Softing Industrial Automation GmbH



Date and Time Format

This section handles the time and date formats that are to be used when the timestamp placeholders are expanded and their values written to the file output.

The Date and Time format settings apply to item data values of data type date and time (DATETIME), and to item data values configured to the data value types: Timestamp (without milliseconds), Timestamp (with milliseconds), Internal Timestamp (without milliseconds), or Internal Timestamp (with milliseconds). For more details on the latter data value attributes, please see the description at Data Values Placeholder.

The default values are `mm/dd/YYYY` for the *Date format and* `HH:MM:SS pp` for the *Time format*. Other defined values for Time and Date format are:

- *Date format*: `mm/dd/YYYY, dd-mm-YYYY, dd.mm.YYYY, dd/mm/YYYY, YYYY-mm-dd, mm-dd-YYYY`
- *Time format*: `HH:MM:SS pp, HH:MM:SS.FFF pp, HH:MM:SS.FFF, HH:MM:SS`

Note: Values for Time and Date format can also be defined manually but in this case special care must be taken to ensure that the definitions are recognized by the local dataFEED OPC Suite application.

In addition to the default values listed in the format selection combo boxes, the user can add his/her own custom format. Custom Date and Time format strings are accepted as valid, when

the format strings contain in their composition the following accepted format substrings and format delimiters. These formats allow the inclusion of specific locale information for the regional settings.

- *Valid Date format strings*: `AAAAAA, aaa, BBBBBBBB, bbb, ccccccccccccccccccc, dd, jjj, mm, UU, ww, xxxxxxxxxx, yy, YYYY`
- *Valid Time format strings*: `II, HH, MM, SS, pp, XXXXXXXX, FFFFFF, FFFFF, FFFF, FFF ,FF, F`
- *Valid Date and Time format delimiters*: `<dot>, <colon>, <slash>, <comma>, <backslash>, <vertical bar>, <semicolon>, <space>, <tab>, <minus>`

Time Zone handles the source localization time: UTC (Universal Time Coordinated) or local time (local time localization). The default time zone is UTC.

When the *Time, Date Format and Time Zone* option is not selected the default values are used. If they are modified, they apply to all actions of the currently selected File definition.

Value Operations

The File Definition may be configured to replace all TRUE BOOLEAN values with "1" instead of "-1" by marking the *Replace BOOLEAN value "-1" with "1" for TRUE* checkbox in the *Value Operations* group box.

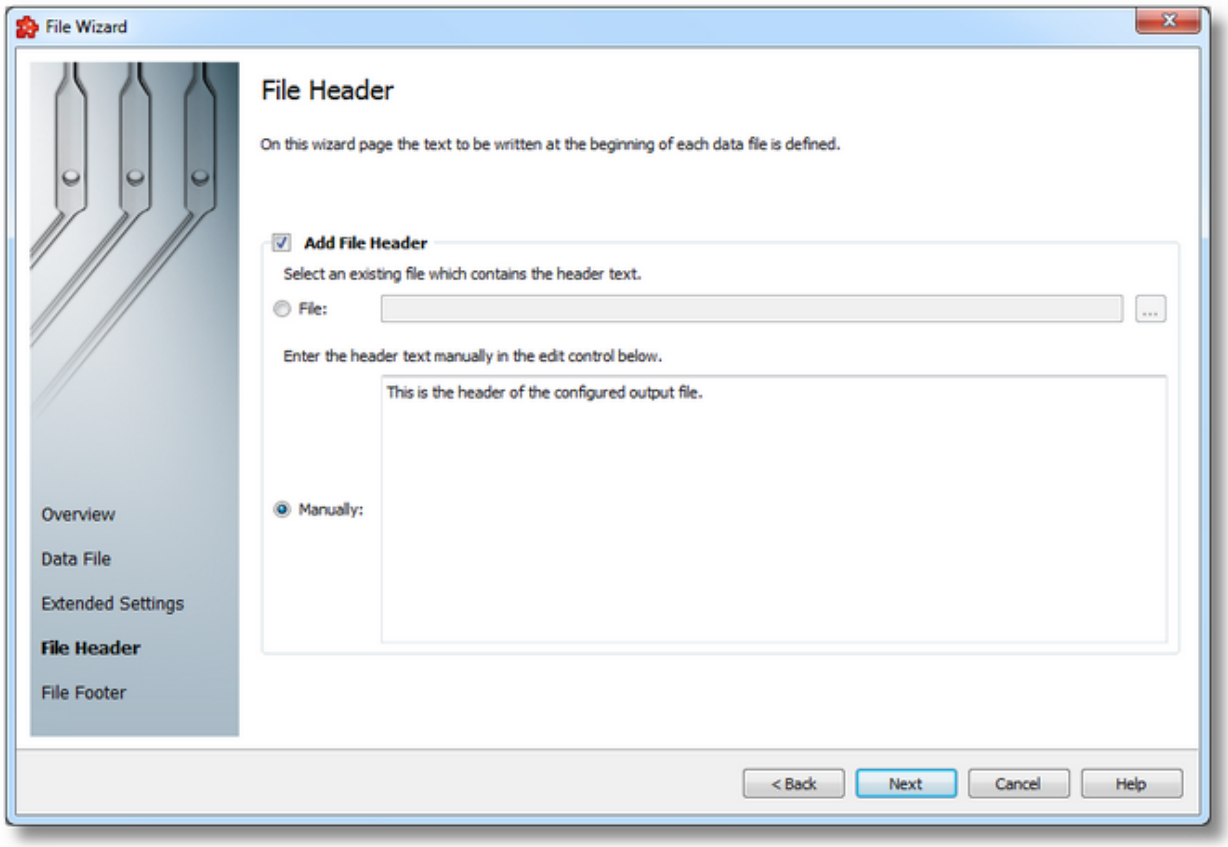
The *Replace BOOLEAN value "-1" with "1" for TRUE* checkbox is unmarked by default for new File Definitions.

To proceed with the wizard and continue to define the file's header press the *Next* button. To go to the previous wizard pages use the *Back* button. Exit the wizard by pressing the *Cancel* button.

<<< File Wizard - Data File File Wizard - File Header >>>

The *File Header* wizard page allows defining the optional file header. The file header is inserted in each output file created by this add-on. The header may be an existing file but it may also be manually typed in.

© Softing Industrial Automation GmbH



By default this option is disabled but it can be enabled by marking the *Add File Header* checkbox.

When enabled, the header can be defined in two ways:

- select an existing file and use the file contents as the file header.
- manually type the file header in the provided text area.

From File

When the File option is selected, the path and file name pointing to an external file can be typed into the corresponding text box. The file can also be selected by clicking the browse button to the right of the file input box.

Note: If this option is selected, it is mandatory to provide a valid path and file name wherefrom the header will be imported. It is not possible to continue the wizard without properly providing/selecting this file.

Manually

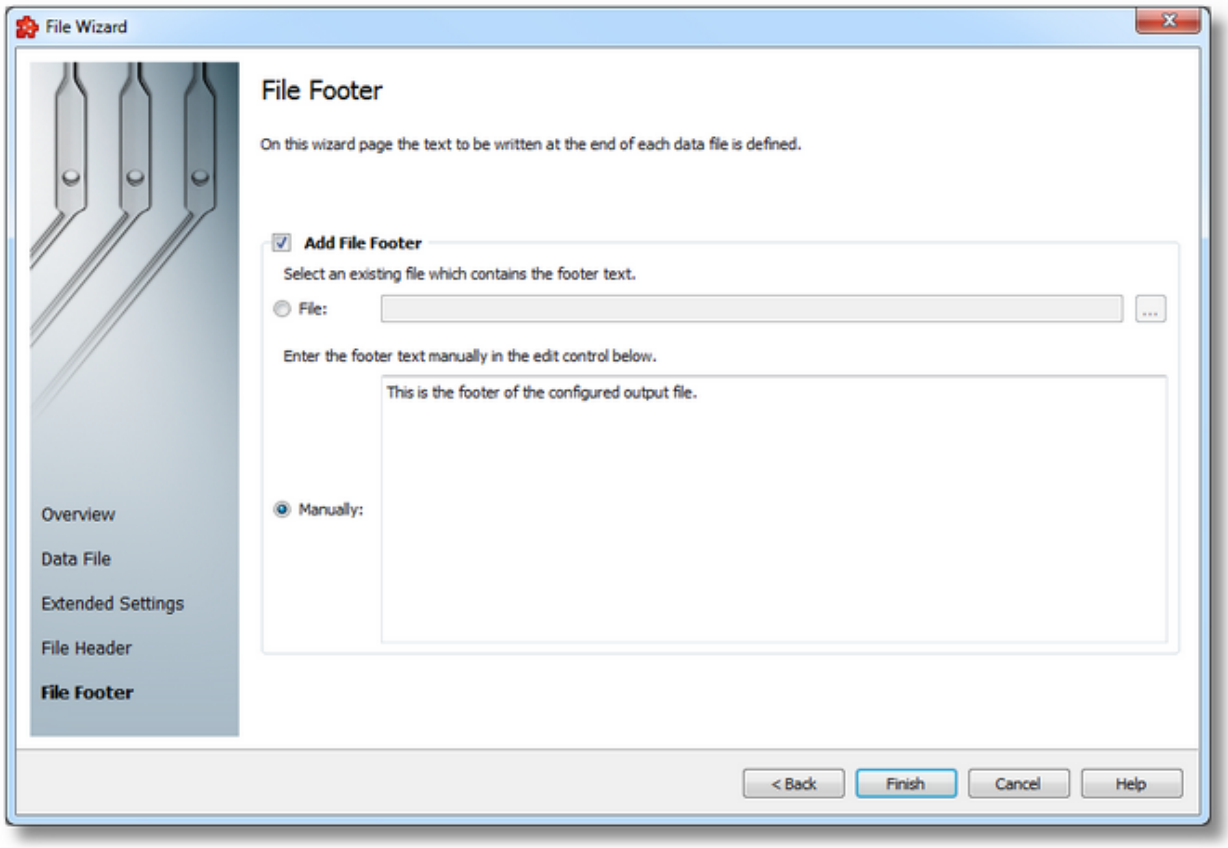
When this option is selected, it is possible to add a file header by manually typing the text in the provided text area.

Note: If this option is selected, it is mandatory to provide the header otherwise the wizard cannot continue.

To proceed forward and define the file footer use the *Next* button, to go back and modify the *Data File* options use the *Back* button. Exit the wizard using the *Cancel* button.

<<< File Wizard - Extended Settings File Wizard - File Footer >>>

The *File Footer* wizard page allows defining the optional file footer. The file footer is inserted in each output file created by this add-on. The footer may be an existing file but it may also be manually typed in.



By default this option is disabled but it can be enabled by marking the *Add File Footer* checkbox.

When enabled, the footer can be defined in two ways:

- select an existing file and use the file contents as the file footer.
- manually type the file footer in the provided text area.

© Softing Industrial Automation GmbH

From File

When the File option is selected, the path and file name pointing to an external file can be typed into the corresponding text box. The file can also be selected by clicking the browse button to the right of the file input box.

Note: If this option is selected, it is mandatory to provide a valid path and file name wherefrom the footer will be imported. It is not possible to continue the wizard without properly providing/selecting this file.

Manually

When this option is selected, it is possible to add a file footer by manually typing the text in the provided text area.

Note: If this option is selected, it is mandatory to provide the footer otherwise the wizard cannot continue.

To complete the *File Wizard* use the *Finish* button, to go back and modify the file header use the *Back* button. Exit the wizard using the *Cancel* button.

<<< File Wizard - File Header

7.3.3.8.3 File Action Wizard

- With the File Action Wizard you can create a new File Action definition or modify an existing one:

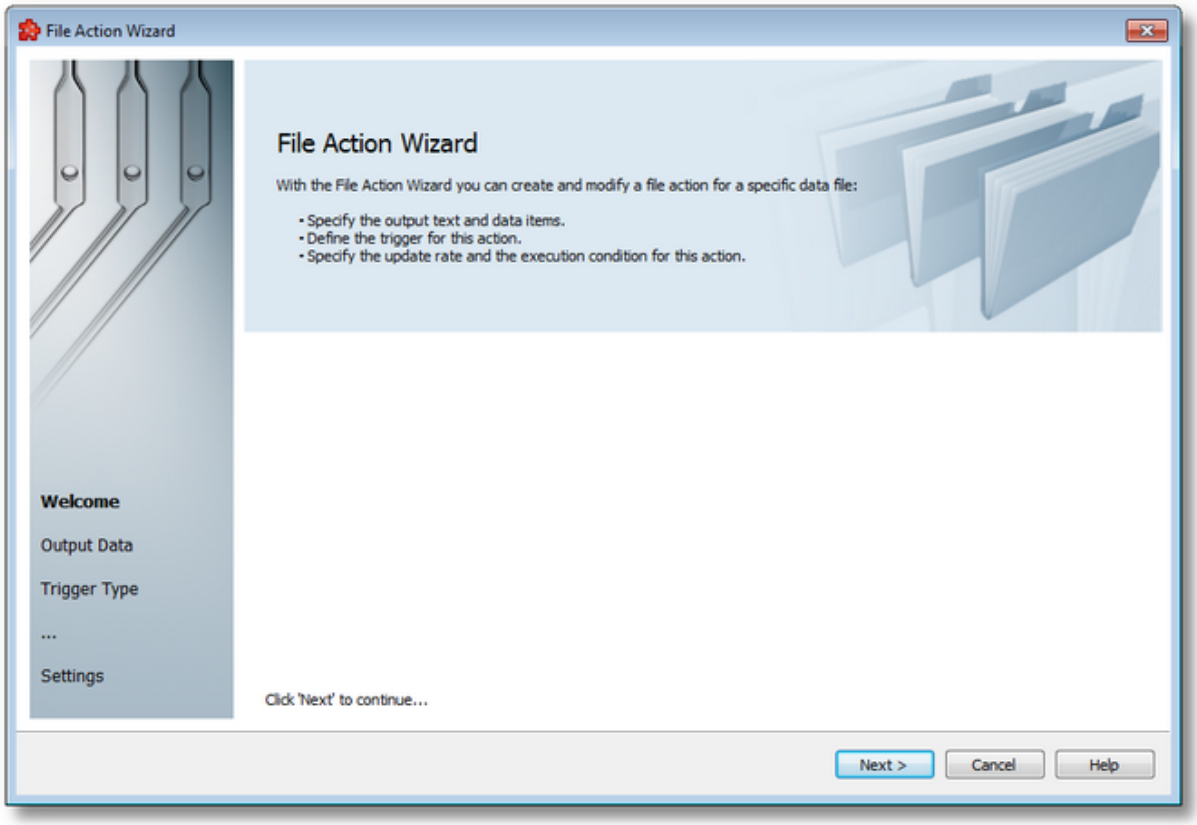
  - Obtain general information about the File Action Wizard @ File Action Wizard - Welcome
  - Define the text output for the actual file action @ File Action Wizard - Output Data
  - Choose the file action trigger type @ File Action Wizard - Trigger Type
  - Pick the item trigger @ File Action Wizard - Trigger Item
  - Select the timer trigger @ File Action Wizard - Timer Selection
  - Define the timer identity @ File Action Wizard - Timer Identity

- Choose the timer type @ File Action Wizard - Timer Type
- Adjust schedule timer settings @ File Action Wizard - Schedule Timer Settings
- Set the timer interval @ File Action Wizard - Interval Timer Settings
- Select the variable trigger @ File Action Wizard - Variable Trigger Selection
- Configure the execution condition and the update rate @ File Action Wizard - Action Settings

Note: A File Action Wizard flow will not include all the pages listed above. A specific branch will be followed during the wizard according to the selections performed in some of the pages.

The Welcome wizard page describes the basic steps for completing the file action definition:

- specify the text statement of the current action using text and placeholders.
- define the trigger settings of the current action.
- specify the subscription update rate and the execution condition of the current action.



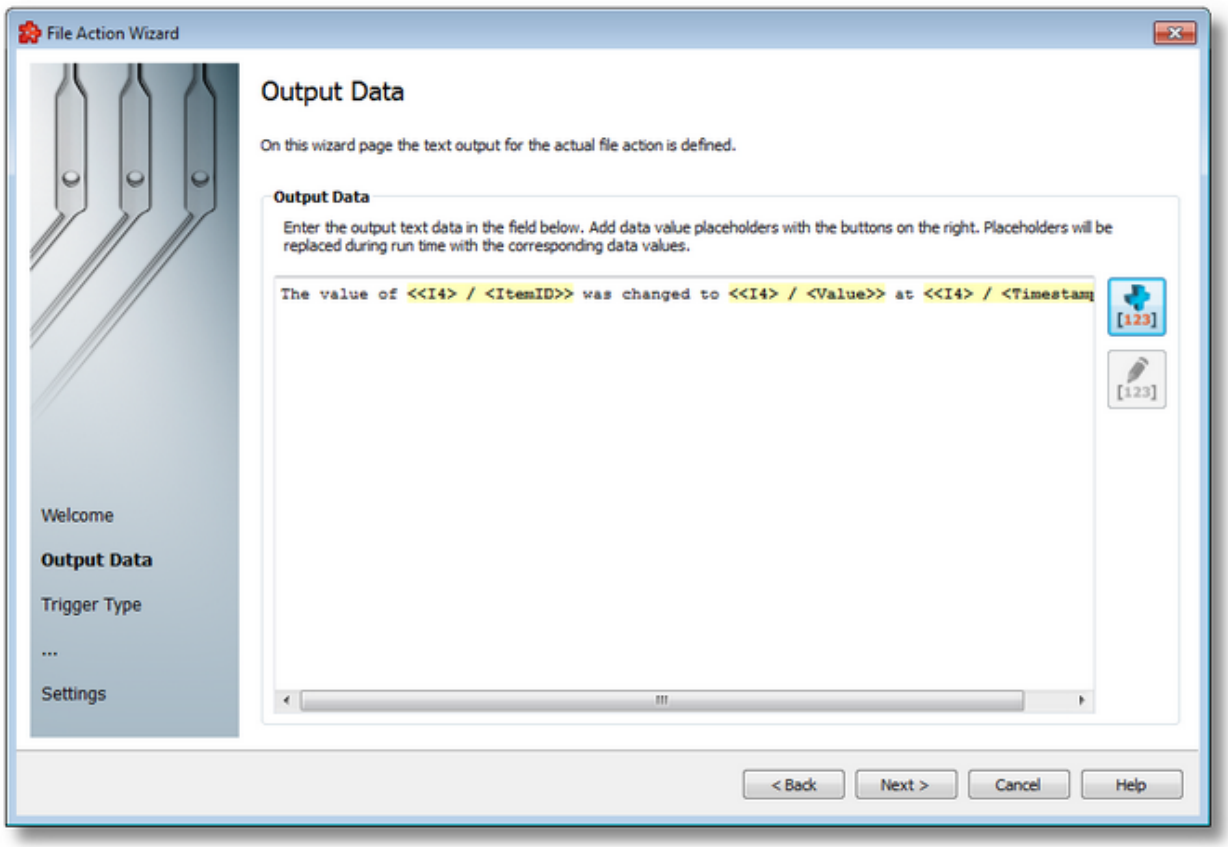
Proceed to the next wizard page pressing the *Next* button, exit the wizard pressing the *Cancel* button.

© Softing Industrial Automation GmbH

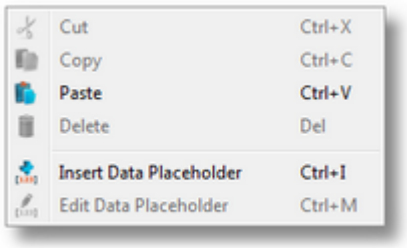
File Action Wizard - Output Data >>>

This wizard page allows defining the text for the current action using the action editor. Certain operations can be performed using the buttons on the right, the context sensitive menu or the keyboard shortcuts:

- Insert a data value placeholder using the *Insert Data Value Placeholder* button  , pressing the keyboard shortcut: CTRL+I or selecting the *Insert Data Value Placeholder* action from the context menu.
- Modify an existing data value placeholder using the *Modify Data Value Placeholder* button  , pressing the keyboard shortcut: CTRL+M or selecting the *Edit Data Value Placeholder* action from the context menu.
- Cut, copy, paste or delete text with or without placeholders selecting the *Cut*, *Copy*, *Paste* or *Delete* actions from the context menu or using the keyboard shortcuts associated with these actions CTRL+X, CTRL+C, CTRL+V respectively the DELETE / DEL keyboard key.



An example of the context sensitive menu is shown below:



Note: The *Cut*, *Copy* and *Paste* actions use the Windows Clipboard for data transfer thus allowing data to be transferred to/from other applications (e.g. the textual representation of the action editor content can be used in external applications and text from external application can be used in the action editor).

Attention: Please keep in mind that placeholders copied to clipboard (through the *Copy* or *Cut* commands) will only be pasted back as placeholders if the clipboard has not been tampered with, otherwise the placeholders will be pasted as plain text.

Attention: The placeholders visible in the action editor are aliases used only for displaying the placeholders. The real placeholder form is visible in the subscription file. Example: *<<R8> / <Value>>* placeholder from the action editor is serialized in the subscription file as: *#%-ItemValue-%increment.R8_%Current-%#*. Do not manually modify the subscription file by inserting placeholders in the action editor template.

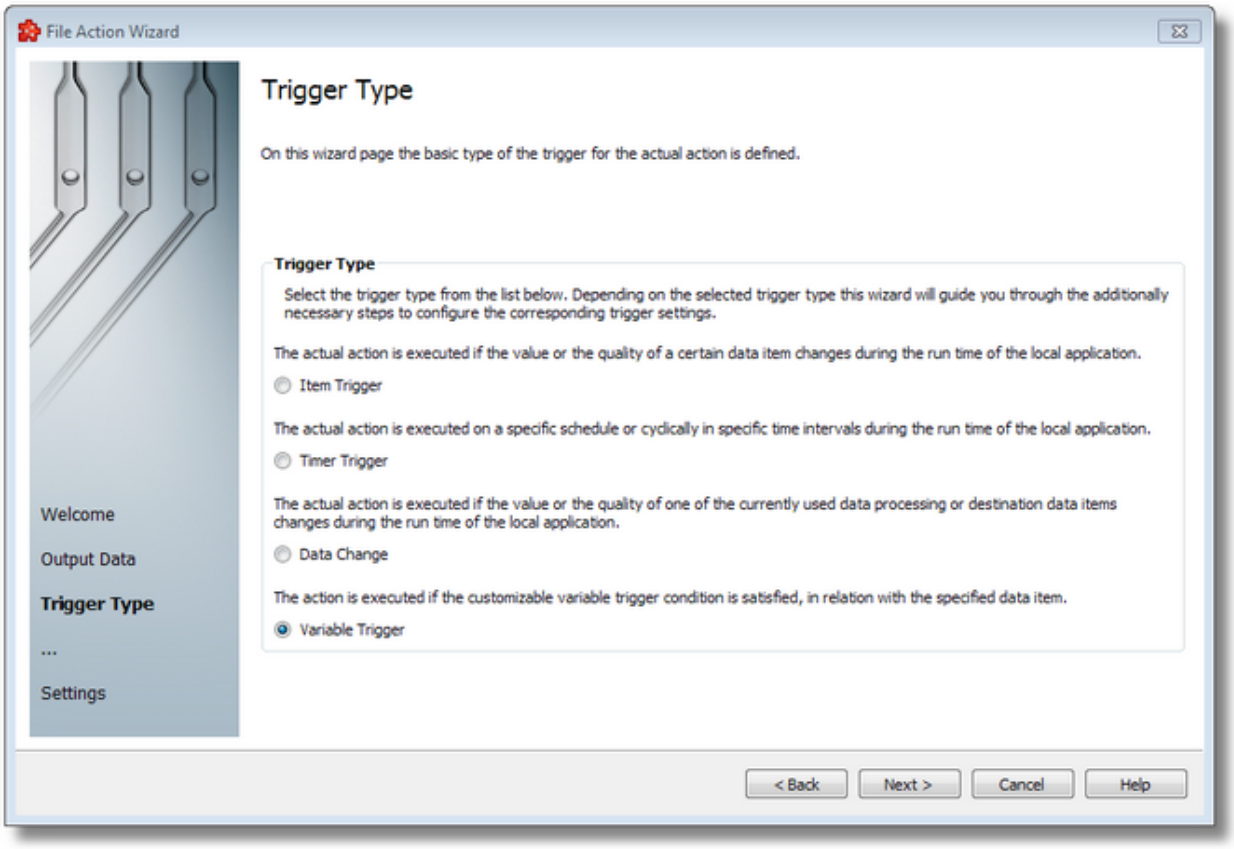
To proceed forward and define the trigger type use the *Next* button, to go back to the previous wizard page use the *Back* button. Exit the wizard using the *Cancel* button.

<<< File Action Wizard - Welcome File Action Wizard - Trigger Type >>>

The Trigger Type wizard page provides the possibility to select the trigger type for the actual action. The following trigger types are available:

- Item Trigger - This trigger is fired when a data change for the selected item occurred. A data change means that the value or the quality of the item changed.
- Timer Trigger - This trigger is fired at a specified time or interval.
- Data Change - This trigger is fired when a data change occurred in at least one of the items used in any of the consumer add-ons.
- Variable Trigger - This trigger is fired when the variable trigger condition is satisfied, in relation with the specified data items.

© Softing Industrial Automation GmbH



To proceed with the wizard and continue to define the selected type of trigger (or to advance to the last wizard page when the *Data Change* trigger type is selected) use the *Next* button, to go back to the previous wizard page use the *Back* button, or discard the wizard by pressing the *Cancel* button.

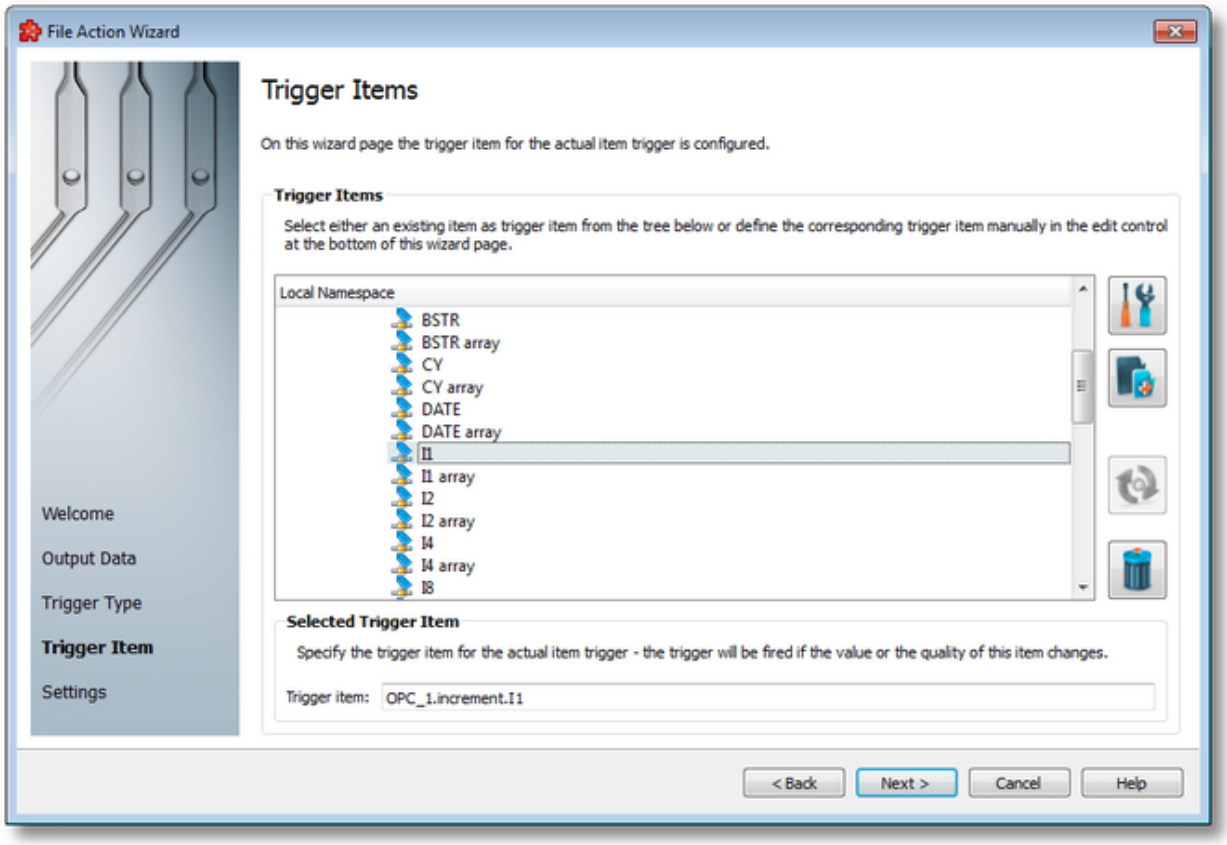
Depending on the selected *Trigger Type* the next wizard page will be:

- File Action Wizard - Trigger Item - if the selected trigger type is *Item Trigger*,
- File Action Wizard - Timer Selection - if the selected trigger type is *Timer Trigger*,
- File Action Wizard - Action Settings - if the selected trigger type is *Data Change* (unlike the above options, no other parameters need to be configured here),
- File Action Wizard - Variable Trigger - if the selected trigger type is *Variable Trigger*

The description of the File Action Wizard will continue with the wizard pages for configuring an *Item Trigger*.

<<< File Action Wizard - Output Data File Action Wizard - Trigger Item >>>

The *Trigger Item* wizard page is shown if the selected trigger type is *Item Trigger*. Here you can configure the actual trigger item.



This wizard page contains two separate sections:

- *Trigger Items* section, which allows selecting an item trigger or managing the Local Namespace tree.
- *Selected Trigger Item* section, which shows the item ID of the selected trigger item and also allows typing an item ID, thereby selecting this item in the above list.

Trigger Items section

The entire namespace of the current configuration is shown as a namespace tree. The name of the root node is composed by appending the instance name to the *Local Namespace* keyword. The first level of the namespace contains:

- A node for each server connection, named by the server connection items prefix.
- A node for System Items.
- A node for Local Items.

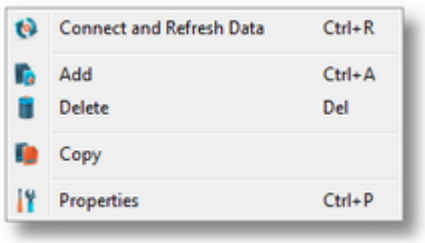
The local namespace tree offers the possibility to explore the entire namespace of the local dataFEED OPC Suite application in order to select a trigger item. After the selection is made, the *Trigger item* edit box is populated with the selected item ID.

The entire namespace of the local dataFEED OPC Suite application can be managed through this wizard page by performing certain operations using the buttons on the right, the context sensitive menu or the keyboard shortcuts:

© Softing Industrial Automation GmbH

- Open the item properties dialog using the *Properties of the selected item* button  , pressing the keyboard shortcut CTRL+P or selecting the *Properties* action from the context menu. Depending on the type of the selected item, a special Item Properties dialog opens:
    - If the selected item is under Local Items the Modify Local Item dialog is opened.
    - If the selected item is under a dynamic server connection the Modify Dynamic Server Item dialog is opened.
    - If the selected item is under a static server connection the Modify Static Server Item dialog is opened.
- Add a new item using the *Add a new item*  button, pressing the keyboard shortcut CTRL+A or selecting the *Add* action from the context menu. Depending on the type of the selected node, a special Add Item dialog opens:
    - If the selected node is under Local Items the Add Local Item dialog is opened.
    - If the selected node is under a dynamic server connection the Add Dynamic Server Item dialog is opened.
    - If the selected node is under a static server connection the Add Static Server Item dialog is opened.
- Copy the selected item to clipboard by selecting the *Copy* action from the context menu.
- Paste the previously copied item from clipboard by selecting the *Paste* action from the context menu. Items originating from the Local Items cannot be pasted under server connections, and vice-versa; an error message will be displayed in both cases. If an item is pasted in a static (filtered) server connection and the item does not exist on the server an error message is displayed.
- Connect to the server and refresh the data items by using the *Refresh data items*  button, pressing the keyboard shortcut CTRL+R or selecting the *Connect and Refresh Data* action from the context menu. The namespace of the selected server connection is recreated; if manually added items not referenced in the actual configuration exist, those items are deleted from the namespace.
- Delete the selected item and all of its child items by using the Delete data items button  , pressing the DELETE / DEL key or selecting the Delete action from the context menu. Items which are included in any Data Processing / Data Destination add-ons will not be deleted but an error message will be displayed instead.

An example of the context menu can be seen below:



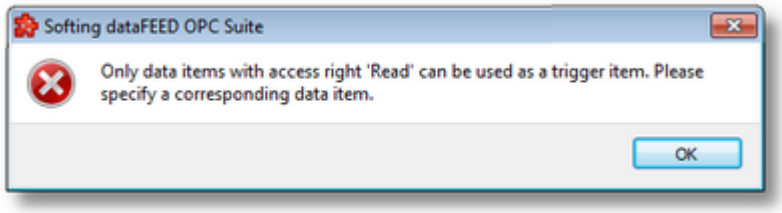
Selected Trigger Item section

The edit box in this section is populated with the item ID after an item from the Local Namespace tree is selected.

You can also manually type the ID of the item in the *Trigger item* edit box and thus have the above tree automatically select a valid server item.

Note: If the item ID is typed and the specified item exists in the Local Namespace, it will be used as the trigger item. The item will only be created if it could not be localized in the Local Namespace. In that case the newly created item will be used as the trigger item.

Note: If the item corresponding to the typed item ID could not be created, an error message will be displayed requesting the input of a new, valid item.



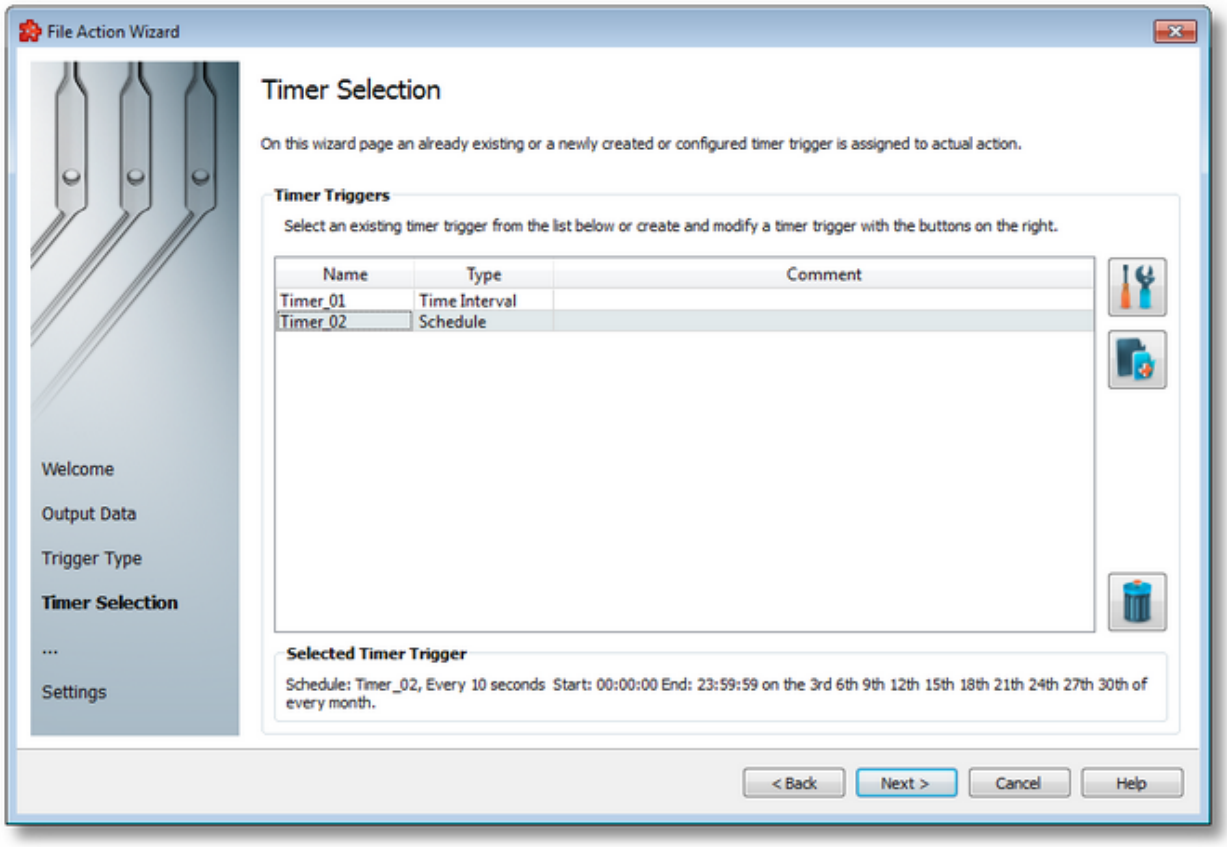
Attention: If the item corresponding to the typed item ID is valid and resides under a static server connection, the respective item will be created in the filter file definition.

To proceed with the wizard and continue to define the action settings use the *Next* button. To go back to the previous wizard page use the *Back* button, or exit the wizard and discard all the changes by clicking the *Cancel* button.

<<< File Action Wizard - Trigger Type File Action Wizard - Action Settings >>>

The Timer Selection wizard page is shown only if the selected trigger type is *Timer Trigger*. Here you can configure the actual timer trigger.

© Softing Industrial Automation GmbH

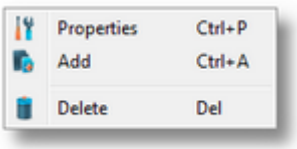


The current page allows selecting a previously defined timer trigger and managing the timer list by using the buttons on the right, the context sensitive menu or the keyboard shortcuts:

- Change the properties of the selected timer by using the *Properties* button  , pressing the keyboard shortcut CTRL+P or selecting the *Properties* action from the context menu.
- Add a new timer by using the Add timer trigger button  , pressing the keyboard shortcut CTRL+A or selecting the Add action from the context menu.
- Delete the selected timer trigger by using the Delete selected timer trigger button  , pressing the DELETE / DEL key or selecting the Delete action from the context menu.

Note: If a timer trigger has associated actions it cannot be deleted.

An example of the context sensitive menu can be seen below:



Note: Information about the selected timer is shown in the *Selected Timer Trigger* area at the bottom of the page.

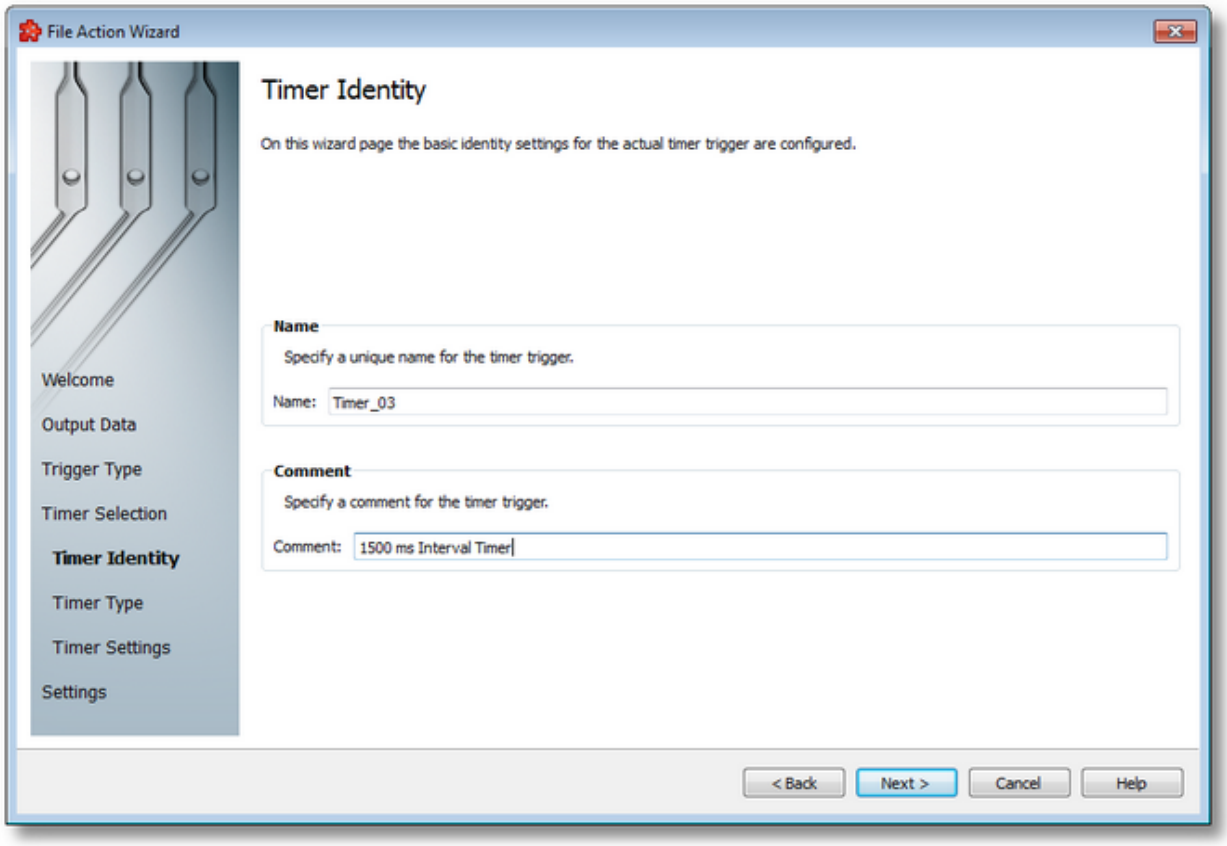
A sub-wizard that allows adding or modifying timer triggers can be launched by selecting to change an existing timer trigger or to add a new one. The first page of this sub-wizard is File Action Wizard - Timer Identity.

To proceed with the *File Action Wizard* and define the action settings press the *Next* button. To return to the previous wizard page use the *Back* button, or exit the wizard by clicking the *Cancel* button.

The wizard will continue straight to the *Action Settings* page as an already defined timer is selected.

<<< File Action Wizard - Trigger Type File Action Wizard - Action Settings >>>

The Timer Identity wizard page provides the possibility to assign a name and a comment to the timer trigger.



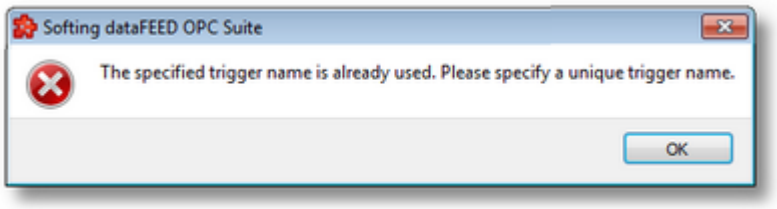
Note: When adding a new timer trigger, the timer name will be automatically generated according to the format: *Timer_number*, where the *number* part is an auto-incremented integer.

When editing an existing timer, the timer's defined name and comment will be filled in the two edit boxes allowing you to modify these attributes.

© Softing Industrial Automation GmbH

Attention: The name of the *Timer Trigger* must be unique among all triggers and cannot be empty.

If the newly added timer trigger has the same name as an existing trigger, an error message box will be displayed and a new trigger name must be provided:



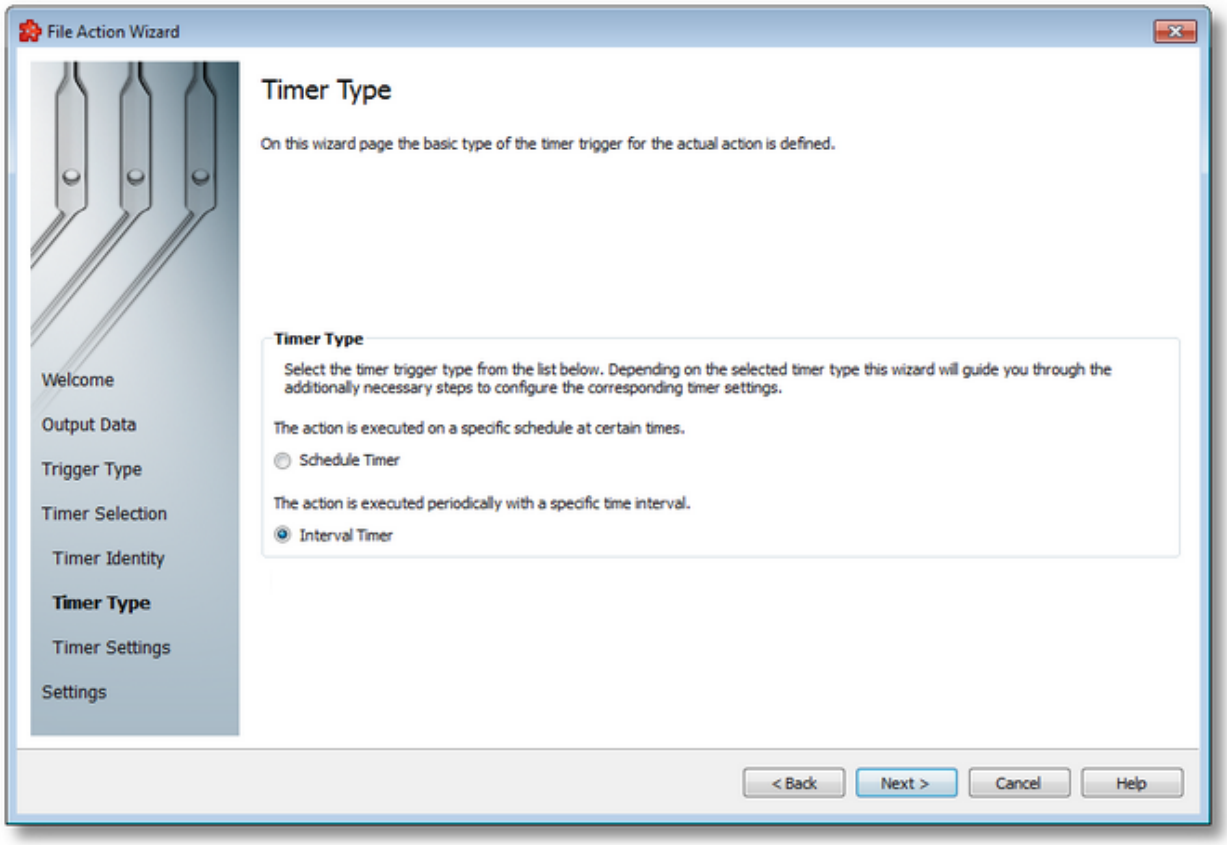
The *Comment* field of the *Timer Identity* wizard page is optional; a Timer Trigger can also be added without any comment.

Continue to the next wizard page to choose the timer type by using the *Next* button, go back to the previous wizard page to select another timer trigger by pressing the *Back* button, or cancel the entire *File Action Wizard* by pressing the *Cancel* button.

<<< File Action Wizard - Timer Selection File Action Wizard - Timer Type >>>

This wizard page allows choosing the type of the timer trigger. The following types of timer are available:

- *Interval Timer*
- *Schedule Timer*



Interval Timer

The Interval Timer type allows defining a specific interval of time (in milliseconds) between the timer fire events. The time interval can be configured on the File Action Wizard - Interval Timer Settings wizard page.

Schedule Timer

The Schedule Timer type allows defining a specific date and time when the timer should fire. The schedule can be configured on the File Action Wizard - Schedule Timer Settings wizard page.

Note: The *Timer Type* of an existing *Timer Trigger* cannot be changed.

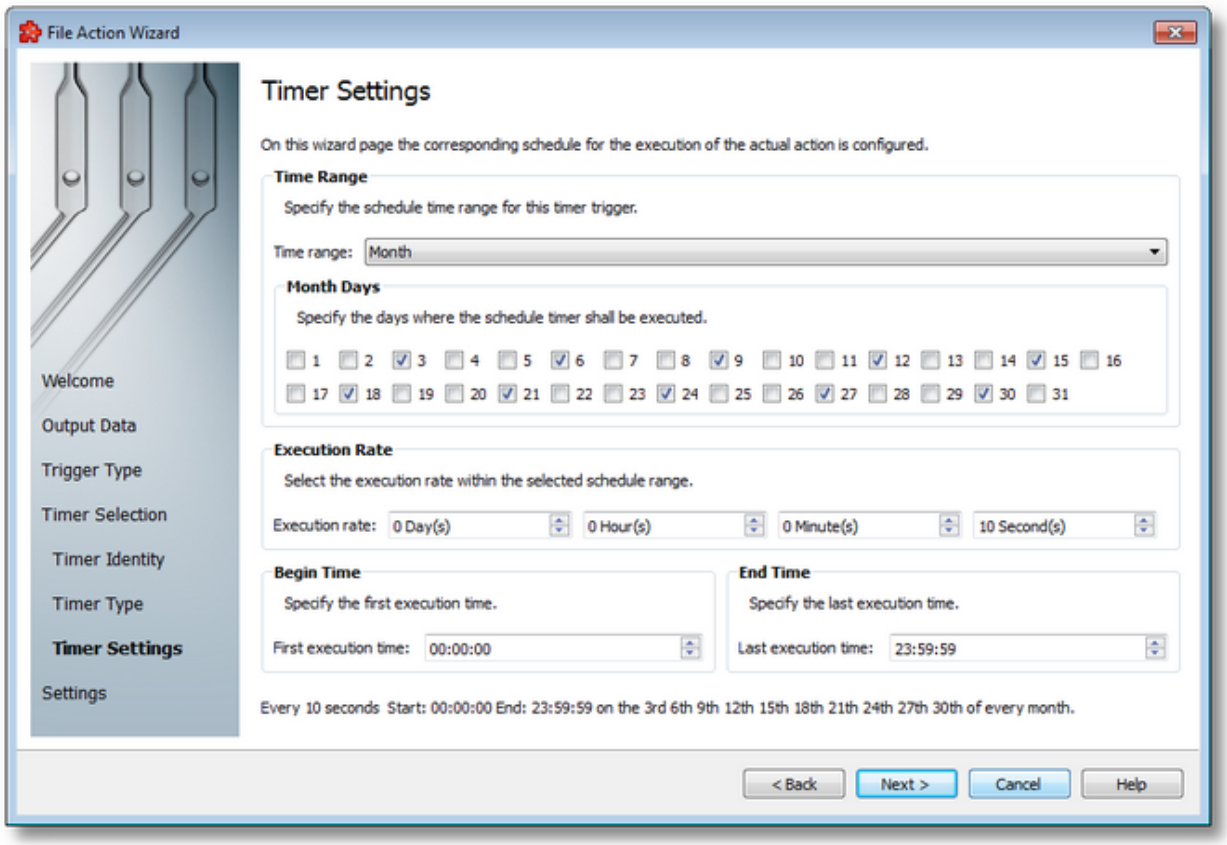
To continue defining the specific timer properties press the *Next* button, to return to the *Timer Identity* wizard page use the *Back* button, or exit this sub-wizard (thereby aborting the *File Action Wizard*) by pressing the *Cancel* button.

The description of the File Action Wizard will continue with the wizard pages for configuring a *Schedule Timer*.

© Softing Industrial Automation GmbH

<<< File Action Wizard - Timer Identity Settings >>> File Action Wizard - Schedule Timer

The *Schedule Timer Settings* wizard page is a more complex page compared to the *Interval Timer Settings* wizard page and allows defining the corresponding schedule for the execution of the trigger action:



The page has four sections which identify the schedule:

- *Time Range*: Provides the possibility to choose one of the following schedule ranges: *month*, *week*, *day*, *hour* or *minute*. When the *month* or the *week* ranges are selected the panel below the selection is populated with the month and week days, respectively.
- *Execution Rate*: Sets the execution rate of the current action in the previously-defined *Time Range* (e.g. every X day(s), every Y hour(s), every Z minute(s), every T second(s)).
- *Begin Time*: Sets the time the current action should begin execution.
- *End Time*: Sets the time the current action should stop execution.

According to the *Time Range* the *Begin Time* and the *End Time* fields will allow typing:

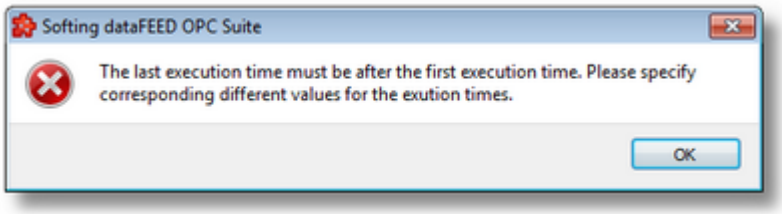
- Seconds (SS) if the *Time range* is *Minute*.
- Minutes and seconds (MM:SS) if the *Time range* is *Hour*.

© Softing Industrial Automation GmbH

- Hours, minutes and seconds (HH:MM:SS) if the *Time range* is *Day* or greater (*Week*, *Month*).

Note: A short review of the configured timer schedule is shown at the bottom of this wizard page.

If the defined schedule is not valid, an error message dialog will appear naming the error that occurred.

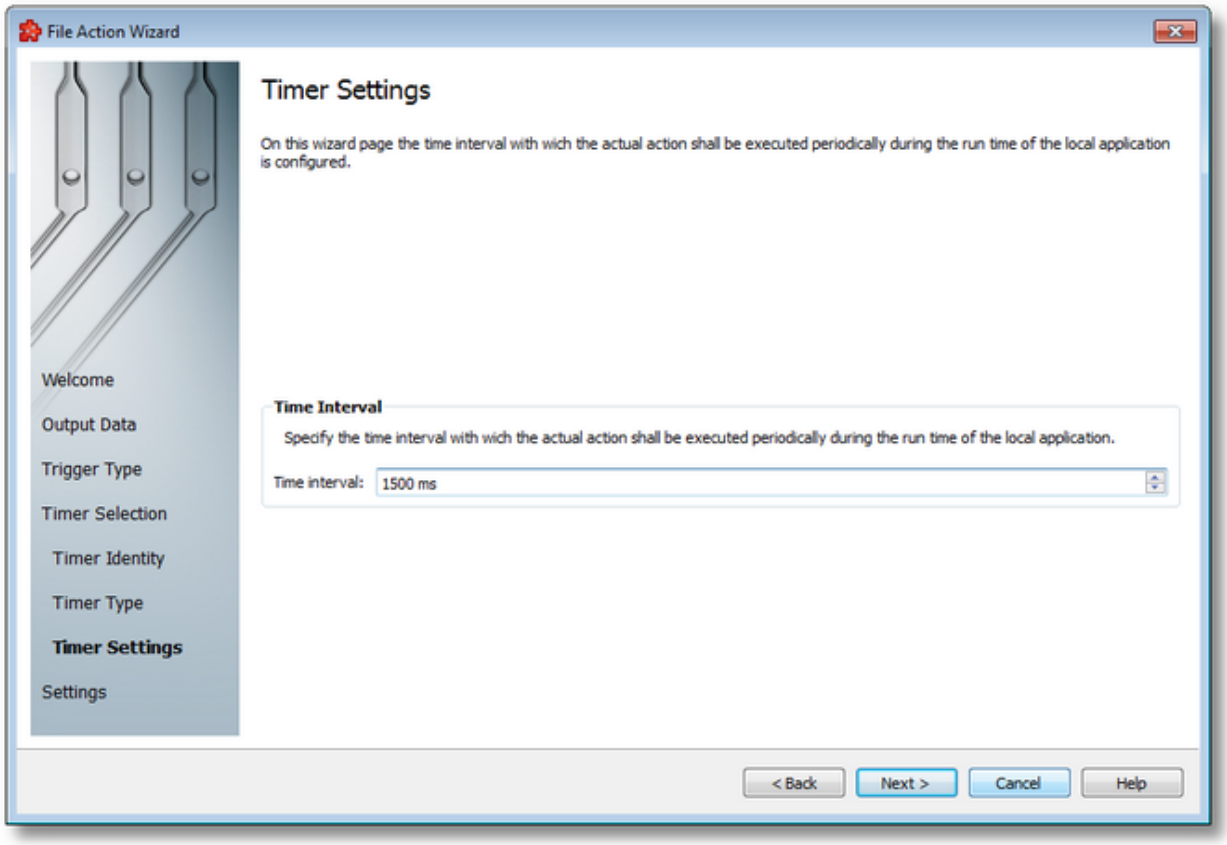


Continue to the *Timer Selection* wizard page using the *Next* button, return to the previous wizard page using the *Back* button or exit the wizard pressing the *Cancel* button.

<<< File Action Wizard - Timer Type File Action Wizard - Timer Selection >>>

The Interval Timer Settings wizard page allows configuring the time interval at which the current action should be (periodically) executed.

© Softing Industrial Automation GmbH

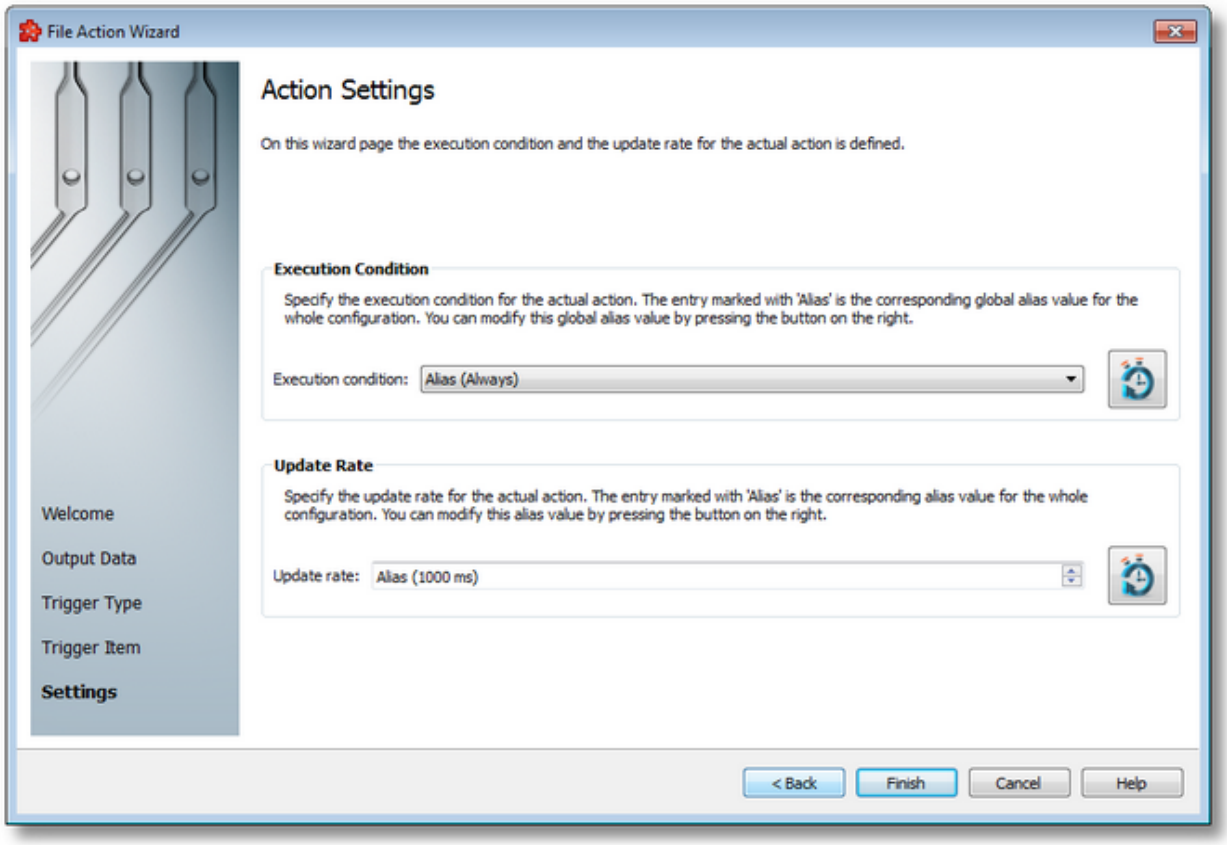


The time interval value is measured in milliseconds and ensures that the action will be triggered at the time interval defined in the spin-box.

Continue to the *Timer Selection* wizard page by pressing the *Next* button, return to the previous one by using the *Back* button, or exit the wizard by pressing the *Cancel* button.

<<< File Action Wizard - Timer Type File Action Wizard - Timer Selection >>>

The Action Settings wizard page allows defining the execution condition and the subscription update rate for the current file action.



The wizard page is divided into two sections: *Execution Condition* and *Update Rate*.

Execution Condition

Each statement has an execution condition which is evaluated by the local dataFEED OPC Suite application. The action is only performed when the evaluation is successful. If the evaluation fails then the statement is ignored.

The execution conditions allow reacting to server communication errors, executing special statements or preventing the execution of other statements.

The execution condition can be selected directly from the drop down combo box. The initial value is the global alias one. Changing the global default execution condition can be done using the Configure global alias and initial values button  or pressing the keyboard shortcut CTRL+D.

Update Rate

The *Update Rate* specifies the execution rate of the current file action.

The update rate can be typed or selected from the spin box. The initial value is the global alias one. Changing the global default update rate can be done using the Configure global alias and initial values button  or pressing the keyboard shortcut CTRL+D.

© Softing Industrial Automation GmbH

Complete the *File Action Wizard* pressing the *Finish* button, review the past wizard pages using the *Back* button, or cancel the wizard thus discarding all the changes by pressing the *Cancel* button.

<<< File Action Wizard - Trigger Type

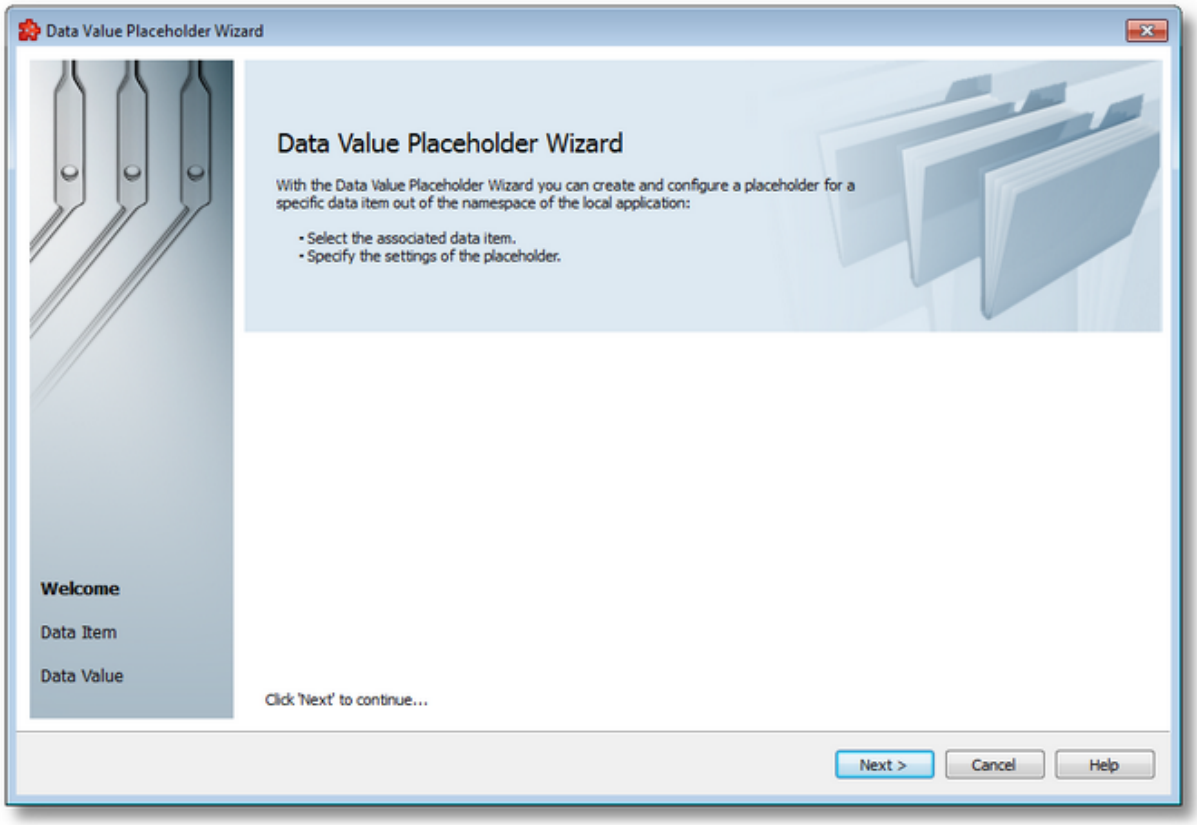
7.3.3.8.4 Data Value Placeholder Wizard

▾ With the Data Value Placeholder Wizard you can create a placeholder from a configured data source or from a valid wildcard:

- Obtain a general description of the wizard @ Data Value Placeholder Wizard - Welcome
- Select a placeholder item @ Data Value Placeholder Wizard - Data Item
- Select the data value and source @ Data Value Placeholder Wizard - Data Value

The Welcome wizard page describes the individual steps for defining the placeholder:

- Select an item from the local dataFEED OPC Suite application namespace or provide a valid wildcard.
- Choose the data value and the data source for the selected item.



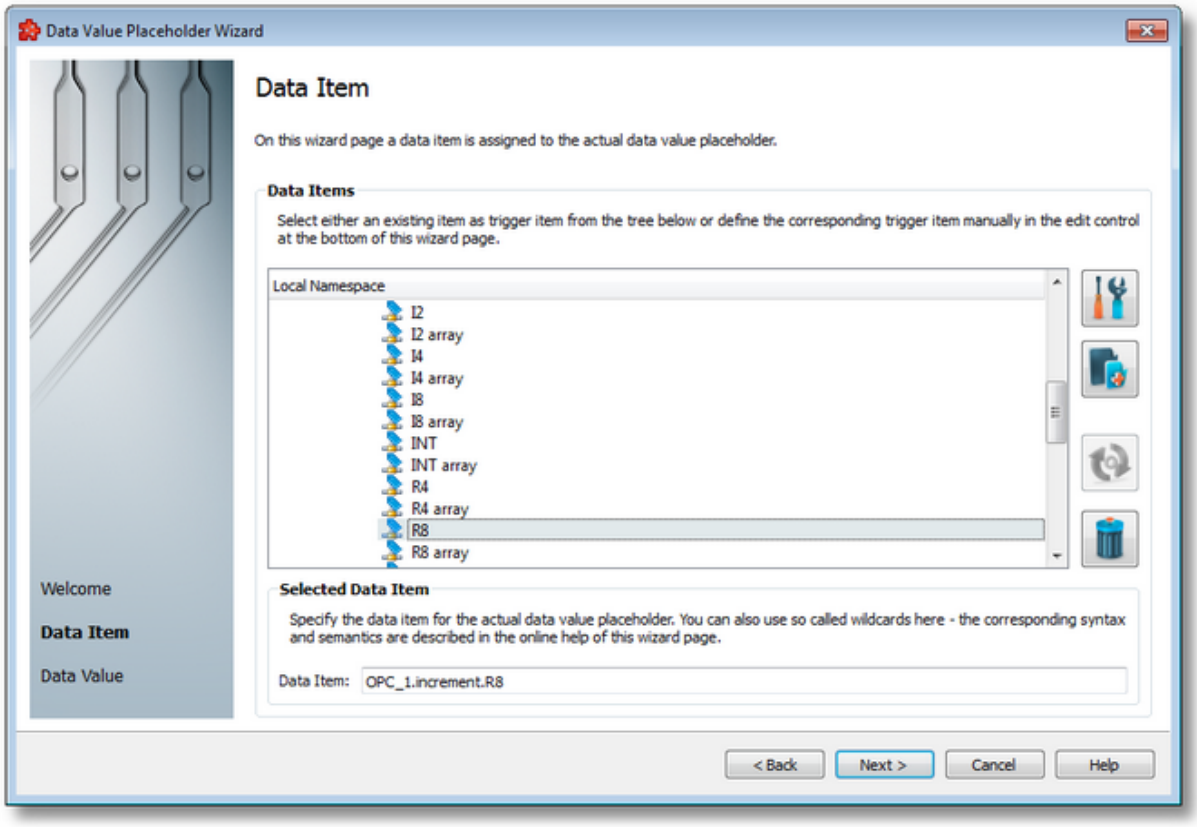
Continue to the next wizard page using the *Next* button or quit the wizard pressing the *Cancel* button.

Data Value Placeholder Wizard - Data Item >>>

The Data Item wizard page allows selecting the placeholder item. The placeholder item can be:

- Any item originating from a *Server Connection*, from the *Local Items* or from the *System Items* can be selected, choosing a tag from the *Local Namespace*.
- A valid wildcard can be manually created by typing its regular expression in the *Data Item* edit box.

© Softing Industrial Automation GmbH



Local Namespace Tree Handling

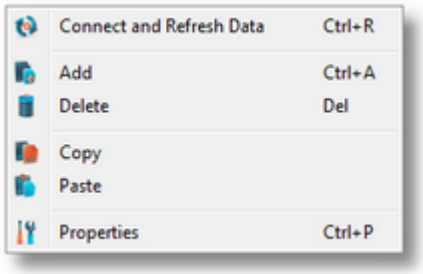
The *Local Namespace* tree offers the possibility to explore the entire namespace of the local dataFEED OPC Suite application in order to select a data item. After the selection is made, the *Data Item* edit box is populated with the selected item ID.

The entire namespace of the local dataFEED OPC Suite application can be managed through this wizard page by performing certain operations using the buttons on the right, the context sensitive menu or the keyboard shortcuts:

- Open the item properties dialog using the *Properties of the selected item* button  , pressing the keyboard shortcut CTRL+P or selecting the *Properties* action from the context menu. Depending on the type of the selected item, a special Item Properties dialog opens:
    - If the selected item is under Local Items the Modify Local Item dialog is opened.
    - If the selected item is under a dynamic server connection the Modify Dynamic Server Item dialog is opened.
    - If the selected item is under a static server connection the Modify Static Server Item dialog is opened.
- Add a new item using the *Add a new item*  button, pressing the keyboard shortcut CTRL+A or selecting the *Add* action from the context menu. Depending on the type of the selected node, a special Add Item dialog opens:

  - If the selected node is under Local Items the Add Local Item dialog is opened.
  - If the selected node is under a dynamic server connection the Add Dynamic Server Item dialog is opened.
  - If the selected node is under a static server connection the Add Static Server Item dialog is opened.
- Copy the selected item to clipboard by selecting the *Copy* action from the context menu.
- Paste the previously copied item from clipboard by selecting the *Paste* action from the context menu. Items originating from the Local Items cannot be pasted under server connections, and vice-versa; an error message will be displayed in both cases. If an item is pasted in a static (filtered) server connection and the item does not exist on the server an error message is displayed.
- Connect to the server and refresh the data items by using the *Refresh data items* button, pressing the keyboard shortcut CTRL+R or selecting the *Connect and Refresh Data* action from the context menu. The namespace of the selected server connection is recreated; if manually added items not referenced in the actual configuration exist, those items are deleted from the namespace.
- Delete the selected item and all of its child items by using the Delete data items button , pressing the DELETE / DEL key or selecting the Delete action from the context menu. Items which are included in any Data Processing / Data Destination add-ons will not be deleted but an error message will be displayed instead.

An example of the context menu can be seen below:



Attention: The *Local Namespace* root node, the *Local Items* root node and the server connection root nodes cannot be deleted. An item that is used in consumers or a system item cannot be deleted either.

Note: The copy/paste actions work on the selected tag/property or on the whole selected branch. The copy action is not available for the Local Namespace root or for the server connection root nodes; the paste action is not available for items of type property (as they cannot have any children).

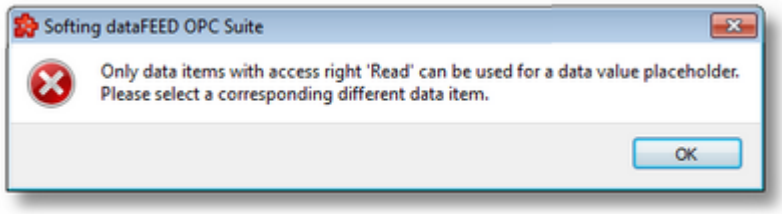
The entire namespace of the local Softing dataFEED OPC Suite application is exposed in this wizard page. The tree root node name is composed of the *Local Namespace* keyword followed by the instance name.

© Softing Industrial Automation GmbH

Selected Data Item Handling

When an item from the Local Namespace is selected, the *Data Item* edit box is populated with the item ID.

Note: Only tag and property items are allowed to be used in placeholder. If a node item is selected, an error message dialog will appear when the *Next* button is pressed.



Note: The *Data Item* edit box in the *Selected Data Item* section of this wizard page is editable. The text typed in this text area is considered as wildcard.

Regular expressions with special characters (asterisk, question mark, square brackets) can be used in this field.

The asterisk wildcard (*) substitutes any sequence of characters of any length:

- SID* - replaces any string of at least 3 characters that starts with SID
- S*D – replaces a string of at least 2 characters that starts with S and ends with D

The question mark wildcard (?) substitutes a single character:

- SID? – replaces any string of 4 characters that starting with SID.
- S?D – replaces any string of 3 characters that starts with S and ends with D.

The range wildcard ([ ]) substitutes any single character in the range specified between the square brackets:

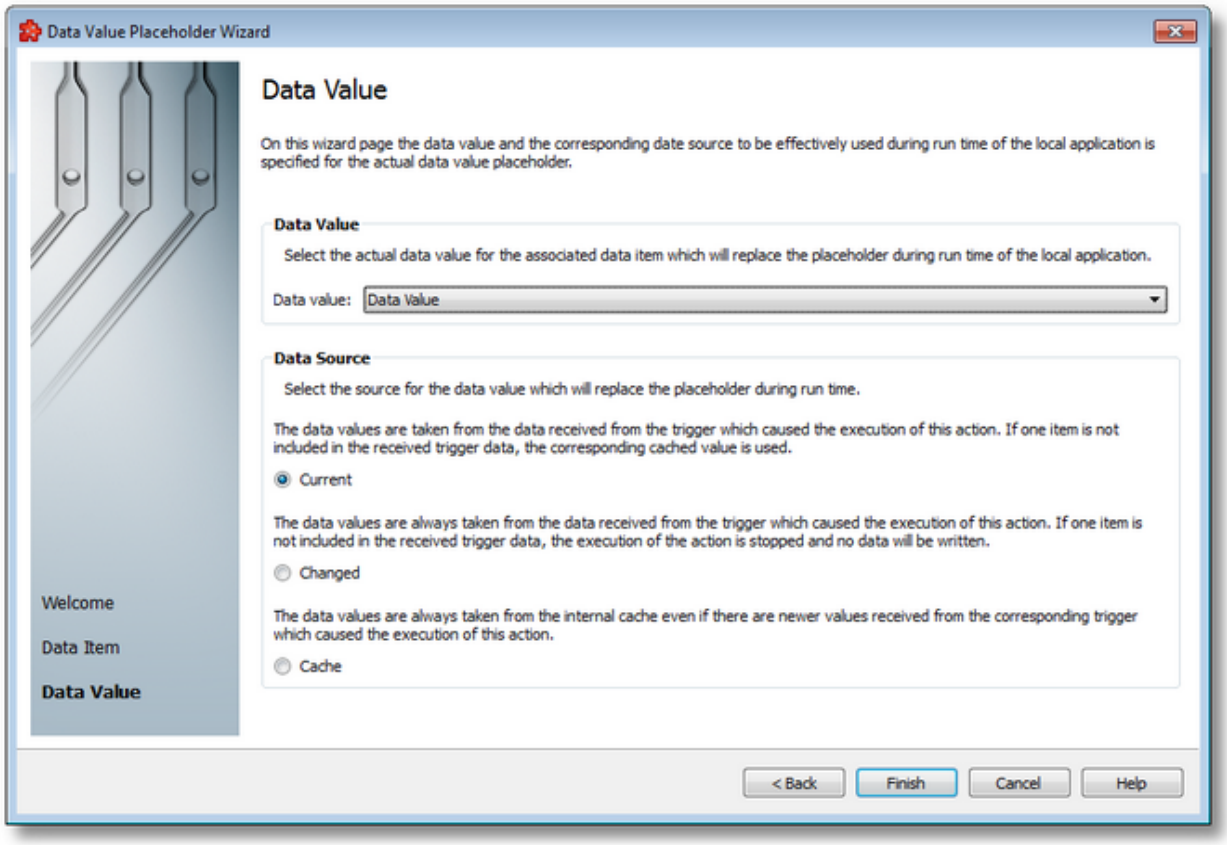
- S[A-Z]D – replaces any string of 3 characters that starts with S, ends with D and contains any character in the range A to Z.
- SID[0-9] – replaces any string of 4 characters that starts with SID and ends with any number in the range 0 to 9.

Proceed to the next page to set the item data value and source by using the *Next* button, return to the previous wizard page by using *Back* button, or quit the wizard by pressing the *Cancel* button.

<<< Data Value Placeholder Wizard - Welcome Data Value Placeholder Wizard - Data Value >>>

The Data Value wizard page provides the possibility to set the properties of the placeholder:

- Data Value - select the actual data value for the associated data item which will be replaced during the run time of the local dataFEED OPC Suite application.
- Data Source - select the source for data value which will be replaced during the run time of the local dataFEED OPC Suite application.



Data Value

The following data value attributes may be used:

Attribute	Description
Data Value	The value of the data source server item.
Item Identified	The ID of the data source server item.
Quality as Integer	The data source server item quality as integer.
Quality as String	The data source server item quality as string.
Timestamp (without milliseconds)	The data source server item timestamp as date time without milliseconds.

© Softing Industrial Automation GmbH

Timestamp (with milliseconds)	The data source server item timestamp as date time including milliseconds.
Timestamp (milliseconds part only)	The milliseconds part of the data source server item timestamp.
Internal Timestamp (without milliseconds)	The timestamp of data change in the local dataFEED OPC Suite application as date time without milliseconds.
Internal Timestamp (with milliseconds)	The timestamp of data change in the local dataFEED OPC Suite application as date time including milliseconds.
Internal Timestamp (milliseconds part only)	The milliseconds part of timestamp of data change in the local dataFEED OPC Suite application.

Data Source

The following data sources can be selected:

Data Source	Description
Current	The value used is the current one. This means that if the item is included in the transaction, the received value is used. If the item is not included, the cached value is used.
Cache	The cached value is always used, even if the item is present in the incoming transaction.
Changed	The item must be included in the incoming transaction as the current value will be used.

Attention: If the action trigger is set to *Data Change* and the corresponding action contains at least one placeholder having *Changed* as the *Data Source*, that specific item must exist in the incoming transaction in order for the action to be executed. When the item is not present in the transaction, the action is invalidated.

Complete the wizard using the *Finish* button, go back to the previous wizard page using the *Back* button or discard any changes made in this wizard by pressing the *Cancel* button.

<<< Data Value Placeholder Wizard - Data Item

7.3.3.8.5 Data Values Placeholder

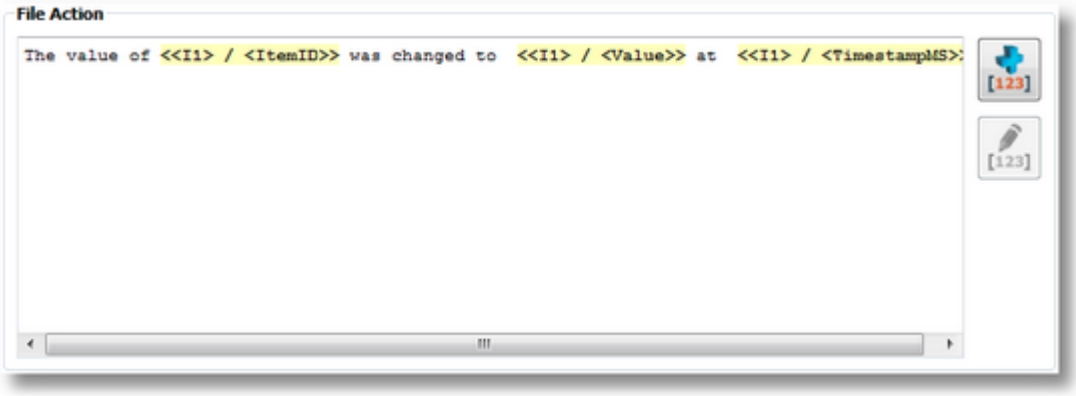
Placeholders can be used when configuring various actions (SQL, TEXT and other statements). These placeholders can be defined in the *File Action Wizard, Database Action Wizard*, *File* or *Database* main user interface and are saved in the Subscription file. A placeholder is determined by the following attributes:

- Data Item - an item (tag, property or node with value) originating from a *Server Connection*, from the *Local Items or* from the *System Items*, or a valid wildcard.
- Data Value - the actual data value for the associated data item which will be replaced during the run time of the local dataFEED OPC Suite application.
- Data Source - the source for the data value which will be replaced during the run time of the local dataFEED OPC Suite application.

Placeholder in the user interface

Two types of placeholder editors are implemented in the Softing Configurator. Only one type of action editor is used in the File add-on, that is the *Text Action Editor*. This placeholder editor is found in File Action Wizard - Output Data wizard page and also in the Configuration: Data Destination - File main user interface page of the File add-on; the corresponding placeholder form is: <<Short Data Value> / <Short Data Item>>:

```
<<R8> / <ItemID>>
<<R8> / <Value>>
<<R8> / <QualityString>>
<<R8> / <TimestampMS>>
```



Placeholder in the subscription file

In the subscription file a placeholder has the following syntax:

```
#%-<Data Item>%<ItemID_SubscriptionID>%<DataSource>-%#
```

where:

© Softing Industrial Automation GmbH

Attribute	Description
#%- -%#	Represents the beginning and the end of a placeholder definition
Data Item	Represents the item data value.
%	Separates the Data Value, Data Item and Data Source fields in the placeholder.
ItemID_SubscriptionID	Represents the subscription item identifier.
Data source	Represents the item data source.

Example:

```
#%-ItemValue%OPC_1.increment.R8_Subscription_222%Current-%#
```

Data value:

The following data value attributes can be used (they are supported by both dataFEED OPC Suite Configurator and local dataFEED OPC Suite application):

Attribute	Subscription file form	Description
Data Value	ItemValue	The value of the data source server item.
Item Identified	ItemItemID	The ID of the data source server item.
Quality as Integer	ItemQuality	The data source server item quality as integer.
Quality as String	ItemQualityString	The data source server item quality as string.
Timestamp (without milliseconds)	ItemTimestamp	The data source server item timestamp as date time without milliseconds.
Timestamp (with milliseconds)	ItemTimestampMS	The data source server item timestamp as date time including milliseconds.
Timestamp (milliseconds part only)	ItemTimestampMS Part	The milliseconds part of the data source server item timestamp.
Internal Timestamp (without milliseconds)	ItemTimestampOC T	The timestamp of data change in the local dataFEED OPC Suite application as date time without milliseconds.
Internal Timestamp (with milliseconds)	ItemTimestampOC TMS	The timestamp of data change in the local dataFEED OPC Suite application as date time

		including milliseconds.
Internal Timestamp (milliseconds part only)	ItemTimestampOCTMSPart	The milliseconds part of the data change timestamp in the local dataFEED OPC Suite application.

The following data value attributes can be used (only supported by the local dataFEED OPC Suite application):

Attribute	**Subscription file form**	**Description**
Item Subscription ID	ItemSubscriptionID	The subscription ID of the item
Data Value	DataValue	The ID of the data source server item.
Data Quality as Integer	DataQuality	The data source server item quality as integer.
Data Quality as String	DataQualityString	The data source server item quality as string.
Timestamp (without milliseconds)	DataTimestamp	The data source server item timestamp as date time without milliseconds.
Timestamp (with milliseconds)	DataTimestampMS	The data source server item timestamp as date time including milliseconds.
Internal Timestamp (without milliseconds)	DataTimestampOCT	The timestamp of data change in the local dataFEED OPC Suite application as date time without milliseconds.
Internal Timestamp (with milliseconds)	DataTimestampOCTMS	The timestamp of data change in the local dataFEED OPC Suite application as date time including milliseconds.
Data Item ID	DataItemID	The value of the data source server item.

Attention: There is a significant difference between the Data Values in the tables above. The values in the second table are taken directly from dataFEED OPC Suite internal server while the values for the Data Values from the first table are taken directly from the server which holds the real Item. dataFEED OPC Suite holds only a copy of that Item.

Data Source or Item type:

One of the following Data Sources may be selected:

Item type	**Description**

© Softing Industrial Automation GmbH

Current	The item must be in the subscription item list. The value used is the current one, which means that if the value for the item is included in the current transaction, this value will be used. If the value is not included in the incoming transaction, the cached value will be used.
Cache	The item must be in the subscription item list. The value used is the cached one, even if that item is included in the incoming transaction.
Changed	The item must be in the subscription item list. The value used is the changed one, which means that the entire action will be invalidated if the item is not present in the incoming transaction.

Examples:

```
'#%-ItemTimestampOCTMSPart%OPC_1.increment.R8_Subscription_222%Current-%#'
'#%-ItemTimestampOCTMS%OPC_1.increment.R8_Subscription_222%Changed-%#'
'#%-ItemTimestampOCT%OPC_1.increment.R8_Subscription_222%Cache-%#'
```

Attention: If the *Changed* Data Source is used in an action and the corresponding action trigger is a *Data Change* trigger, make sure that a transaction which contains data changes for all the items used in the current action will be generated by the server; otherwise, the entire action will be invalidated if an item has no associated data change in the incoming transaction.

Wildcards

A placeholder can also contain a wildcard at the position of ItemID_SubscriptionID, which is matched by the local dataFEED OPC Suite application against the items defined in the subscription file:

```
#%-ItemValue%Wildcard_02%Current-%#
#%-ItemQualityString%Wildcard_02%Cache-%#
#%-ItemTimestamp%Wildcard_02%Changed-%#
#%-ItemItemID%Wildcard_02%Current-%#
```

where Wildcard_02 is the wildcard ID of a wildcard previously defined in the subscription file.

Below you can find examples of possible placeholder definition forms. Please note that not all of them are supported by the dataFEED OPC Suite Configurator:

Placeholder Form	**Description**
#%-ItemValue%<Subscription Item ID>%item type-%#	The item value
#%-ItemQuality%<Subscription Item ID>%item type-%#	The item quality as integer

#%-ItemTimestamp%<Subscription Item ID>% item type-%#	The item timestamp as date time
#%-ItemTimestampMS%<Subscription Item ID> %item type-%#	The milliseconds part of the item timestamp
#%-ItemTimestampOCTMS%<Subscription Item ID>%item type-%#	The timestamp of data change in the local dataFEED OPC Suite application (milliseconds part)
#%-DataValue%<OCT Item ID>%item type%% data type-%#	The item value
#%-DataQuality%<OCT Item ID>%item type-%#	The item quality as integer
#%-DataQualityString%<OCT Item ID>%item type-%#	The item quality as readable string
#%-DataTimestamp%<OCT Item ID>%item type-%#	The item timestamp as date time
#%-DataTimestampOCT%<OCT Item ID>% item type-%#	The timestamp of data change in the local dataFEED OPC Suite application as date time
#%-TAB-%#	The tabulator character
#%-CRLF-%#	Windows-specific line separator (Carriage return + Line feed)
#%-LF-%#	Linux-specific line separator (Line feed)
#%-CR-%#	Macintosh-specific line separator (Carriage return)

Examples of wildcards used in File Actions can be found below:

```
CONDITION<TAB><TAB>Always
CONDITION<TAB>AllGoodQuality<TAB>AllGoodQuality
CONDITION<TAB>Always<TAB>Always
CONDITION<TAB>ItemQualityChanged<TAB>ItemQualityChanged
#
SUBSCRIPTION<TAB><TAB>UPDATERATE<TAB>1000<TAB>ACTIVE<TAB>TRUE
SUBSCRIPTION<TAB>Subscription_500<TAB>UPDATERATE<TAB>500<TAB>ACTIVE<TAB>TRUE
#
ITEM<TAB>increment.I1_Subscription_500<TAB>increment.I1<TAB>Subscription_500
TEXT<TAB>AllGoodQuality<TAB>File_01<TAB>The value of #%-ItemItemID%increment.I1_Subsc
#
ITEM<TAB>increment.I2_<TAB>increment.I2<TAB>
#
ITEM<TAB>increment.I4_Subscription_500<TAB>increment.I4<TAB>Subscription_500
#
```


© Softing Industrial Automation GmbH

```
DATACHANGE
TEXT<TAB>Always<TAB>File_01<TAB>The value of #%-ItemItemID%increment.I4_Subscription_
#
TIMER<TAB>Timer_02<TAB>INTERVAL<TAB>1000
TEXT<TAB>ItemQualityChanged<TAB>File_01<TAB>The value of #%-ItemItemID%increment.I2_%
```

7.3.3.8.6 Configuration Files

All the OPC2File configuration settings are stored in two separated files:

- Tasks file - contains general descriptions of the File definitions.
- Subscriptions file - which contains all the descriptions relating to the stored data:
    - what data will be stored.
    - what is the frequency of storing the data.
    - where and when will the data be stored.

Note: The path and file name of these configuration files are fixed and not configurable.

The task and subscription files are stored in the *Tasks* folder under the instance path. For task configuration a text file called *Tasks.txt* and for subscription configuration a text file called *Subscriptions.txt* will be created in the *Tasks* configuration folder. Although these files can be manually edited, using the dataFEED OPC Suite Configurator is advisable.

Note: The recommendation is to use only the dataFEED OPC Suite Configurator to edit these files. This ensures that all the definitions will be valid.

If the configuration is manually edited try loading the configuration in the dataFEED OPC Suite Configurator as all the configuration files are automatically validated at load-time and, if there are any mistakes, an error message will be displayed. The dialog will contain an appropriate description and will point to the first invalid line of the local items configuration file.

Both configurations files are plain text files encoded in UTF-8 (thus providing full UNICODE compatibility) and they have line-oriented structure. The elements of a line are tab-separated. The format of one line is the following:

```
Line type<TAB>Line data
```

Task Configuration File

The following line types will be defined:

Line Type	Description
File	Describes the parameters needed to write to a file.

FileHeader	Used to specify a free text that is inserted immediately after the startup of the application.
FileFooter	Used to specify a free text that is inserted immediately before the closing of the application.
Commented Lines (#)	Any line that starts with the "#" character is considered a comment line which is ignored during evaluation.

The FILE line

The FILE line allows defining all the parameters needed by the File add-on to write in a file:

```
FILE<TAB>File ID<TAB>Absolute File Path<TAB>File Options
```

Parameter	Description
FILE	The type of the line, predefined keyword.
File ID	Unique file identifier within the task file (used for matching the FILEHEADER, FILEFOOTER lines and the statements in the corresponding subscription file).
Absolute File Path	Absolute file path, including drive letter, path and file name.
File Options	File options used to specify how often data files are to be created: every hour, every day etc.

Example:

```
FILE<TAB>File_01<TAB>C:\Documents and settings\user\My Documents\opcfile1.txt<TAB>
FILE<TAB>File_02<TAB>C:\Documents and settings\user\My Documents\opcfile2.txt<TAB>One
FILE<TAB>File_03<TAB>C:\Documents and settings\user\My Documents\opcfile3.xml<TAB>One
FILE<TAB>File_04<TAB>C:\Documents and settings\user\My Documents\opcfile4.txt<TAB>One
```

The File options can have one of the following values:

- OneFilePerHour
- OneFilePerDay
- OneFilePerWeek

If any of the above File Options is specified, the *OPC2File* add-on will generate a new file for each hour, day and week, respectively. The date and time of the file creation will be added to the file name with the format: <Absolute file path>-YYMMDD-HH. If the file specified in the task file already exists, the OPC2File add-on will append the new data to the file.

© Softing Industrial Automation GmbH

Note: If no File Options are specified, the file is created without the suffix -YYMMDD-HH.

Note: If the output files already exist, they will be overwritten. Backup or rename the output files of the previous session if this data is still needed.

The FILEHEADER line

The FILEHEADER line can be used to specify a text block that is to be inserted at the beginning of each output file. The syntax of this line is the following:

```
FILEHEADER<TAB>File ID<TAB>Absolute File Path
```

Parameter	Description
FILEHEADER	The type of the line, predefined keyword.
File ID	Unique file identifier within the task file (used for matching the FILE line and the statements in the corresponding subscription file).
Absolute path	Absolute path to the file containing the header.

Example:

```
FILEHEADER<TAB>File_01<TAB>C:\Documents and settings\user\My Documents\headerfile.txt
```

Note: The File ID used must be previously defined in a FILE definition.

The FILEFOOTER line

The FILEFOOTER line can be used to specify a text block that is to be inserted at the end of each output file. The syntax of this line is the following:

```
FILEFOOTER<TAB>File ID<TAB>Absolute File Path
```

Parameter	Description
FILEFOOTER	The type of the line, predefined keyword.
File ID	Unique file identifier within the task file (used for matching the FILE line and the statements in the corresponding subscription file).

Absolute path	Absolute path to the file containing the footer.

Example:

```
FILEFOOTER File_01 C:\Documents and settings\user\My Documents\footerfile.txt
```

Note: The File ID used must be previously defined in a FILE definition.

Task File - Example:

```
#<TAB>File Add-on - configuration
FILE<TAB>File_01<TAB>C:\Documents and Settings\All Users\Documents\dataFEED OPC Suite
FILEHEADER<TAB>File_01<TAB>C:\Documents and Settings\All Users\Documents\dataFEED OPC
FILEFOOTER<TAB>File_01<TAB>C:\Documents and Settings\All Users\Documents\dataFEED OPC
```

Subscription Configuration File

The *Data Destination - File* generates multiple line types:

- common lines which are specific to all the consumer add-ons: condition lines, subscription lines and item lines;
- specific lines: modifier lines and copy lines.

Common Lines:

The following line types are common lines and will be used to configure different functionalities:

Line Type	Description
Subscription	Defines the update rate with which the current action will be executed.
Condition	Defines the execution condition for the current action.
Wildcard	Defines a regular expression which is matched against the items defined in the subscription file.
Item	This line defines a trigger which is fired if the item data has changed. The trigger data includes all other items which have changed at the same time (were reported in the same data change callback from the OPC server). An item data change consists of a different value or quality of the respective item compared to the previous one.

© Softing Industrial Automation GmbH

Timer	This line defines a trigger which is fired at specific time and date or at specific intervals.
Datachange	This trigger is fired each time an item data change occurs (for each incoming transaction). At most one Datachange trigger may exist per subscription file. If an internal action has a Datachange trigger, it will receive the data changes of all subscriptions of the local dataFEED OPC Suite application. This includes dataFEED OPC Suite internal subscriptions and subscriptions created by the connected OPC clients.
Commented Lines (#)	Any line that starts with the "#" character is considered a comment line which is ignored during evaluation.

The subscription file is divided into two sections:

- The global area:

  The global area extends from the first subscription file line to the first trigger line (e.g: ITEM, TIMER or DATACHANGE lines).

  This area contains definitions valid for the entire subscription file.

- The local area:

  The local area covers all the actions between two trigger lines in the subscription file.

  The lines in this area are only executed if the parent trigger fires.

Subscription Line

A subscription line has the following format:

```
SUBSCRIPTION<TAB>Subscription ID<TAB>UPDATERATE<TAB>Update rate value<TAB>ACTIVE<TAB>
```

The subscription line contains the following configurable parameters:

Parameter	Description
SUBSCRIPTION	The type of the line.
Subscription ID	It is used to identify the subscription and must be unique.
UPDATERATE	Predefined keyword.
Update rate value	Defines the update rate with which the current action will be executed.

ACTIVE	Predefined keyword.
Active value	Active value which can be TRUE or FALSE.

The subscription lines always contain the global subscription line as well as all subscriptions which are used at least once in consumer actions. The global subscription is represented by the subscription line with the empty *Subscription ID*. The default update rate value for the global subscription is *1000* milliseconds.

The value of the update rate is saved in milliseconds. This value may not be set to 0 inside the dataFEED OPC Suite Configurator but it can be manually set by editing the subscription file with an external text editor. An update rate value of 0 means *as fast as possible* but this is not recommended from a resource-related point of view.

Setting a subscription to be active or inactive is an advanced feature which cannot be configured from within the dataFEED OPC Suite Configurator. By default the active value for each and every subscription is set to *TRUE* (meaning that all subscriptions are active since creation). If a subscription's active value is set to *FALSE* then the data changes of the items configured to be part of that subscription will be ignored.

Attention: From a resource-related point of view it is recommended to remove the unused subscription definitions if the DATACHANGE trigged is to be used.

Examples:

```
SUBSCRIPTION<TAB><TAB>UPDATERATE<TAB>1000<TAB>ACTIVE<TAB>TRUE
SUBSCRIPTION<TAB>Subscription_300<TAB>UPDATERATE<TAB>300<TAB>ACTIVE<TAB>TRUE
```

Condition Line

A condition line defines whether an action below an item trigger will be executed or not (when the item trigger is fired). This can be used in other configuration lines as well to condition the execution of an item action. One example of this is the conditional execution of a SQL statement in the Database add-on. A condition line has the following format:

```
CONDITION<TAB>Condition ID<TAB>Condition Statement
```

The condition line contains the following configurable parameters:

Parameter	Description
CONDITION	The type of the line, predefined keyword.
Condition ID	Unique condition identifier.
Condition statement	Single condition statement.

© Softing Industrial Automation GmbH

Conditions can be used to react to communication errors. Various actions can be executed depending on the quality of items.

The following condition statements are supported:

Condition	Description
AllGoodQuality	The trigger actions are executed only if all items in the transaction have the quality *Good*.
Always	The trigger actions are executed always, ignoring the quality of the items in the incoming transaction.
ItemQualityChanged	The trigger actions are executed only if the quality of at least one item in the transaction has changed.
OneBadQuality	The trigger actions are executed if exactly one item in the transaction has the quality *Bad*.

The condition lines always contain the global condition line as well as all conditions which are used at least once in consumer actions. The global condition is represented by the condition line with the empty *Condition ID*. The default condition statement value for the global condition is *Always*. As a general rule all the condition lines need to be declared in the global section of the subscription file. If no condition is defined the Always condition is applied as default.

Examples:

```
CONDITION<TAB><TAB>Always
CONDITION<TAB>AllGoodQuality<TAB>AllGoodQuality
```

Wildcard Line

The Subscription Item ID, Item ID and Event Source strings can include wildcards (* .? and []) as parts of the string.

A wildcard must be previously defined in a *Wildcard Line* before it can be used in the Action.

A wildcard line has the following format:

```
WILDCARD<TAB>Wildcard ID<TAB>Item type<TAB>Wildcard string
```

The wildcard line contains the following configurable parameters:

Parameter	Description
WILDCARD	The type of the line, predefined keyword

Wildcard ID	Unique wildcard identifier.
Item type	The item type can be used to specify which item should be used.
Wildcard string	The regular expression.

The following item types are allowed:

Item type	**Description**
Current	The item must be in the subscription item list. The value used is the current one, which means that if the value for the item is included in the current transaction, this value will be used. If the value is not included in the incoming transaction, the cached value will be used.
Changed	The item must be in the subscription item list. The value used is the changed one, which means that the entire action will be invalidated if the item is not present in the incoming transaction.
Cache	The item must be in the subscription item list. The value used is the cached one, even if that item is included in the incoming transaction.

Note: The data value defined for wildcards in the wildcard lines overrides the functionality of the data value defined in the item placeholders found in the subscription file.

The asterisk wildcard (*) substitutes any sequence of characters of any length:

- SID* - replaces any string of at least 3 characters that starts with SID
- S*D – replaces a string of at least 2 characters that starts with S and ends with D

The question mark wildcard (?) substitutes a single character:

- SID? – replaces any string of 4 characters that starts with SID.
- S?D – replaces any string of 3 characters that starts with S and ends with D.

The range wildcard ([ ]) substitutes any single character in the range specified between the square brackets:

- S[A-Z]D – replaces any string of 3 characters that starts with S, ends with D and contains any character in the range A to Z.
- SID[0-9] – replaces any string of 4 characters that starts with SID and ends with any number in the range 0 to 9.

© Softing Industrial Automation GmbH

Note: The format of the range wildcard is the following: [rangeStart-rangeEnd]. Please note that the delimiter is the hyphen-minus (-) character.

Note: If the percent sign (%) is to be included in a statement, note that %% is used as substitute for a single percent sign %.

Example:

```
WILDCARD<TAB>All_Integer_Increments<TAB>Current<TAB>Increment.I*
```

Attention: The wildcard can only replace the items defined in the subscription file.

Item Line

An item line is used to add an item to a subscription. The associated actions are executed each time the value or the quality of the specified data items changes during the run time of the local dataFEED OPC Suite application.

An item line has the following format:

```
ITEM<TAB>Subscription Item ID<TAB>Server Item ID<TAB>Subscription ID
```

The item line contains the following configurable parameters:

Parameter	Description
ITEM	The type of the line, predefined keyword.
Subscription Item ID	Unique subscription item identifier (implicitly created by concatenating the Server Item ID and the Subscription ID with the underscore character in between them).
Server Item ID	Item identifier within the local dataFEED OPC Suite application address namespace. This could be a server connection item ID or a local item ID.
Subscription ID	Subscription identifier (defined above, see Subscription Line).

This trigger is fired if the item data or quality has changed (also referred to as data change). The trigger data includes all other items which were included in the data change (items which had their data change reported in the same transaction).

Note: A single line consisting of the *Server Item ID* and *Subscription ID* is added to the subscription file for an item used several times with the same update rate in any consumer add-ons. This optimizes the local dataFEED OPC Suite application throughput.

Example:

```
ITEM<TAB>increment.I1_Subscription_300<TAB>increment.I1<TAB>Subscription_300
```

Note: The subscription item ID can be empty. In this case, however, only the Server Item ID can be used to identify the item.

Timer Line

This trigger is fired at a specified time and date or at specific intervals. The associated actions are executed on a specific schedule or periodically during the run time of the local dataFEED OPC Suite application.

The format of the *Timer Line* is:

```
TIMER<TAB>Timer ID<TAB>INTERVAL<TAB>Rate in milliseconds
```

or

```
TIMER<TAB>Timer ID<TAB>SCHEDULE<TAB>Schedule range<TAB>First execution<TAB>Rate<TAB>L
```

The interval timer contains the following configurable parameters:

Parameter	Description
TIMER	The type of the line, predefined keyword.
Timer ID	Unique timer identifier.
INTERVAL	Denotes an interval timer trigger, predefined keyword.
Interval value	Defines the interval in milliseconds at which the timer will trigger the associated action(s).

The timer schedule contains the following configurable parameters:

Parameter	Description
TIMER	The type of the line, predefined keyword
Timer ID	Unique timer identifier.

© Softing Industrial Automation GmbH

SCHEDULE	Denotes a scheduled timer trigger, predefined keyword.
Schedule range	Defines the range of the schedule (see below the table).
First execution	Defines the time the timer should begin firing events (see below the table).
Rate	Defines the rate at which the timer should trigger the associated action(s).
Last execution	Defines the time the timer should stop firing events (see below the table).
List Of Days	Should be used only with the Week and Month schedule. It must contain the days on which the execution is desired. If the list of days is not present, the obtained functionality is the same as by using the day marker in every day of the week/month.

The timer's *Schedule range* may have one of the following values:

- MINUTE
- HOUR
- DAY
- WEEK - List of days: 1d = Monday, 2d = Tuesday, …, 7d = Sunday.
- MONTH - List of days: 1d = First day of month, 2d = Second day of month, …, 30d = Thirtieth day of month, 31d = Thirty-first day of the month.

The *First execution* defines the time the timer should begin firing events.

Example:

```
1d13h10m15s
```

Note: If parts of the execution time are 0, they do not have to be in the string, e.g. 1d0h10m0s = 1d10m

The timer *Rate* defines the event firing rate. An empty timer *Rate* means a single execution.

Last Execution: The last execution time of the schedule. Could be empty.

List of Days: Shall be used only with the Week and Month schedule. It should contain the days on which execution is desired. If the list is not specified the same functionality can be achieved by using the day marker in the first and last execution time.

Note: If the List of Days and the day marker in first and last execution time are used in the same timer trigger the two setting types should match the marked days!

Examples:

```
# Following timer fires events every 250 milliseconds (4 times a second):
TIMER<TAB><TAB>INTERVAL<TAB>250
# Following timer fires events every 20 seconds of a minute starting at second 10 (eg
TIMER<TAB><TAB>SCHEDULE<TAB>MINUTE<TAB>10s<TAB>20s
# Following timer fires events every 4 hours of a day starting at 6:30, ending at 20:
# (e.g. 6:30, 10:30, 14:30, 18:30).
TIMER<TAB><TAB>SCHEDULE<TAB>DAY<TAB>6h30m<TAB>4h<TAB>20h
# Following timer fires events every 4 hours of a day starting at 6:30, ending at 20:
# Friday (e.g. 6:30, 10:30, 14:30, 18:30 of each day).
TIMER<TAB><TAB>SCHEDULE<TAB>WEEK<TAB>6h30m<TAB>4h<TAB>20h<TAB>1d2d3d4d5d
# Following timer fires events every 4 hours starting at 6:30, ending at 22:45 (e.g.
# 18:30, 22:30), on the 1th, 2th, 10th, 20th of every month.
TIMER<TAB><TAB>SCHEDULE<TAB>MONTH<TAB>6h30m<TAB>4h<TAB>22h45m<TAB>1d2d10d20d
# Following timer fires events every 1 hours, 3 minutes and 3 seconds starting at 00:
# 23:59:59 (e.g. 00:00:01, 01:03:04, 02:06:07 etc), on the 3rd, 6th, 9th, 12th, 15,th
TIMER<TAB>Timer_02<TAB>SCHEDULE<TAB>MONTH<TAB>1s<TAB>1h3m3s<TAB>23h59m59s<TAB>3d6d9d1
```

Datachange Line

This trigger is fired on every data change of an item, regardless of the subscription the incoming transaction belongs to. The format of a Datachange line is the following:

```
DATACHANGE
```

Note: A single Datachange trigger may exist per subscription file.

The associated actions are executed if the value or the quality of one of the currently used data processing or destination data items changes during the run time of the local dataFEED OPC Suite application.

Attention: The Datachange trigger will be fired for each configured subscription rate (when an incoming transaction of at least one item arrives).

Note: Make sure that only used subscription definitions are present in the configuration file (resource-related point of view).

Specific Lines:

The following line types are OPC2File specific and will be used to configure different functionalities:

Line Type	Description

© Softing Industrial Automation GmbH

TEXT	Defines a free text statement.

The format of the TEXT line is the following:

```
TEXT<TAB>Condition ID<TAB>File ID<TAB>TEXT statement
```

The TEXT line contains the following configurable parameters:

Parameter	**Description**
TEXT	The type of the line, predefined keyword.
Condition ID	Condition identifier (defined above, see Condition Line).
File ID	File definition identifier (defined above, see File Line).
TEXT statement	A free text statement (can be XML, tab-separated, etc).
Trigger ID	Associated trigger identifier (one of the above defined: Item Trigger, Timer Trigger or Datachange Trigger)

The text statement parameter configures the text which will be inserted in the specified file when the corresponding action is executed. It contains free text and data placeholders.

Example of TEXT configuration lines and the appropriate ITEM lines:

```
# Insert statement without condition (i.e. default condition considered)
#
ITEM<TAB>increment.I1_<TAB>increment.I1<TAB>
TEXT<TAB><TAB>File_TEXT<TAB>The value of #%-ItemItemID%increment.I1_%Current-%# chang
TEXT<TAB><TAB>File_XML<TAB><value ItemID=”#%-ItemItemID%increment.I1_%Current-%#”>#%-
TEXT<TAB><TAB>File_TAB<TAB>#%-ItemItemID%increment.I1_%Current-%##%-TAB-%##%-ItemValu
#
# Insert statement with condition AllGoodQuality (needs to be defined in the conditio
#
ITEM<TAB>increment.I1_<TAB>increment.I1<TAB>
TEXT<TAB>AllGoodQuality<TAB>File_Text<TAB>The value of #%-ItemItemID%increment.I1_%Cu
```

When executed, these statements will add the value and quality of the subscription item "increment.I1" from an empty prefixed server connection to the configured file: File_TEXT, File_XML and File_TAB, respectively.

In order to offer a flexible configuration for TEXT statements, placeholders can also be used. The placeholders are syntactical entities that are replaced with their associated values each time the statement is evaluated during the run time of the local dataFEED OPC Suite application (before the text is written to the file).

Note: The *Condition ID* used in TEXT line definition must be defined in the current Subscription File; the default condition may also be used.

Note: The *File ID* used in a TEXT line definition must be defined in the associated Task File.

The TEXT statements will be executed on every data change event, however their execution will also depend on both the defined conditions and the type of the fired trigger.

If the statements are associated with triggers of type *Datachange*, the respective statements are evaluated on every data change and are only executed if all of the following conditions are satisfied:

- At least one item from the TEXT statement changes.
- The action condition is true.
- The placeholders in the statement can be replaced by their respective values (the changed data source may invalidate the action).

If the statements are associated with triggers of type *Item*, the respective statements are evaluated on each item change, but are only executed if all of the following conditions are true:

- The subscription item ID of the statement has to be defined in the configuration.
- A change occurs for the respective item.
- The action condition is true.
- The placeholders in the statement can be replaced by their respective values (the changed data source may invalidate the action).

If the statements are associated with triggers of type *Timer*, the respective statements are evaluated and executed only if all of the following conditions are true:

- The subscription item ID of the statement has to be defined in the configuration.
- The action condition is true.
- The placeholders in the statement can be replaced by their respective values (the changed data source may invalidate the action).

As a general rule, if a statement includes placeholders with wildcards, at least one of the wildcards must match one of the items that changed. If a statement that includes items which are not in the data changed list should be executed, the values of these items are retrieved from the cache.

Subscription File - Example:

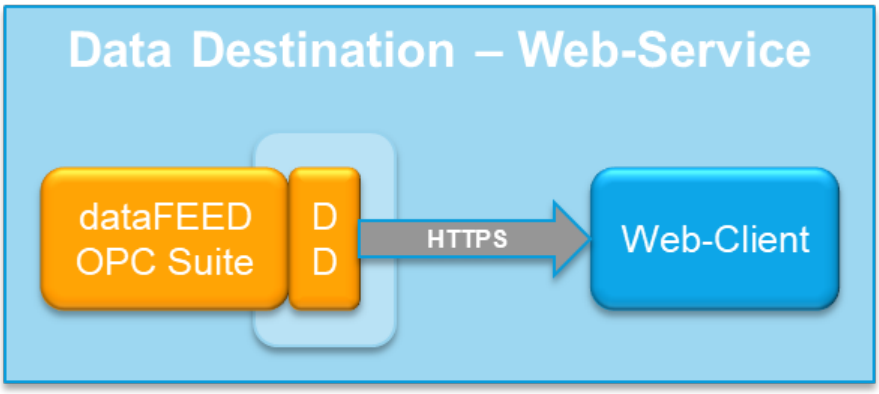
```
CONDITION<TAB><TAB>Always
CONDITION<TAB>AllGoodQuality<TAB>AllGoodQuality
CONDITION<TAB>Always<TAB>Always
CONDITION<TAB>ItemQualityChanged<TAB>ItemQualityChanged
#
SUBSCRIPTION<TAB><TAB>UPDATERATE<TAB>1000<TAB>ACTIVE<TAB>TRUE
SUBSCRIPTION<TAB>Subscription_500<TAB>UPDATERATE<TAB>500<TAB>ACTIVE<TAB>TRUE
```


© Softing Industrial Automation GmbH

```
#
ITEM<TAB>increment.I1_Subscription_500<TAB>increment.I1<TAB>Subscription_500
TEXT<TAB>AllGoodQuality<TAB>File_01<TAB>The value of #%-ItemItemID%increment.I1_Subsc
#
ITEM<TAB>increment.I2_<TAB>increment.I2<TAB>
#
ITEM<TAB>increment.I4_Subscription_500<TAB>increment.I4<TAB>Subscription_500
#
DATACHANGE
TEXT<TAB>Always<TAB>File_01<TAB>The value of #%-ItemItemID%increment.I4_Subscription_
#
TIMER<TAB>Timer_02<TAB>INTERVAL<TAB>1000
TEXT<TAB>ItemQualityChanged<TAB>File_01<TAB>The value of #%-ItemItemID%increment.I2_%
```

7.3.3.9 Web Service

This section of the Softing dataFEED OPC Suite online help explains how to configure Web Services in order to server clients using the HTTPS protocols.

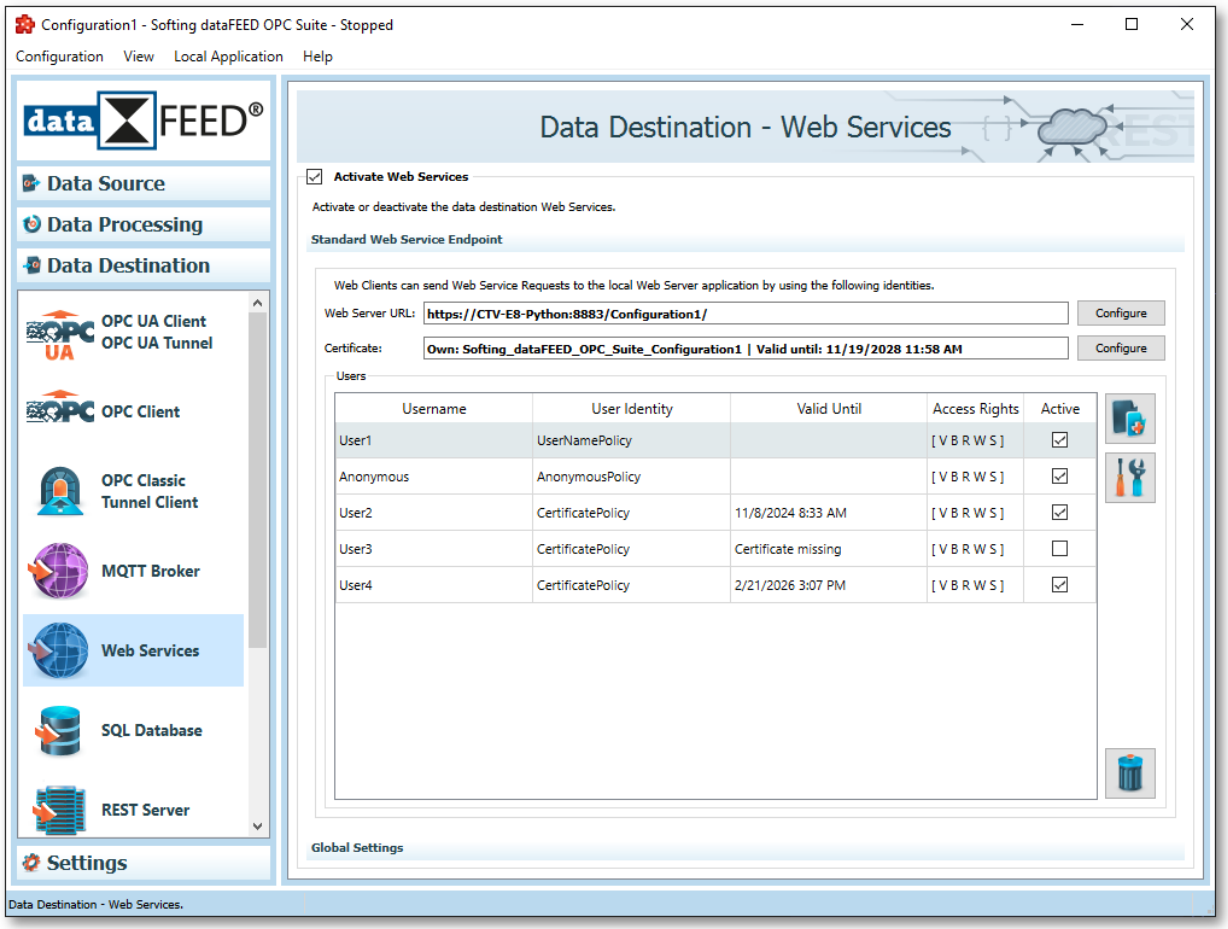


Enabling the Web Services data destination functionality allows external Web Clients to access the data items of the local dataFEED OPC Suite application via the HTTPS protocol.

7.3.3.9.1 Configuration

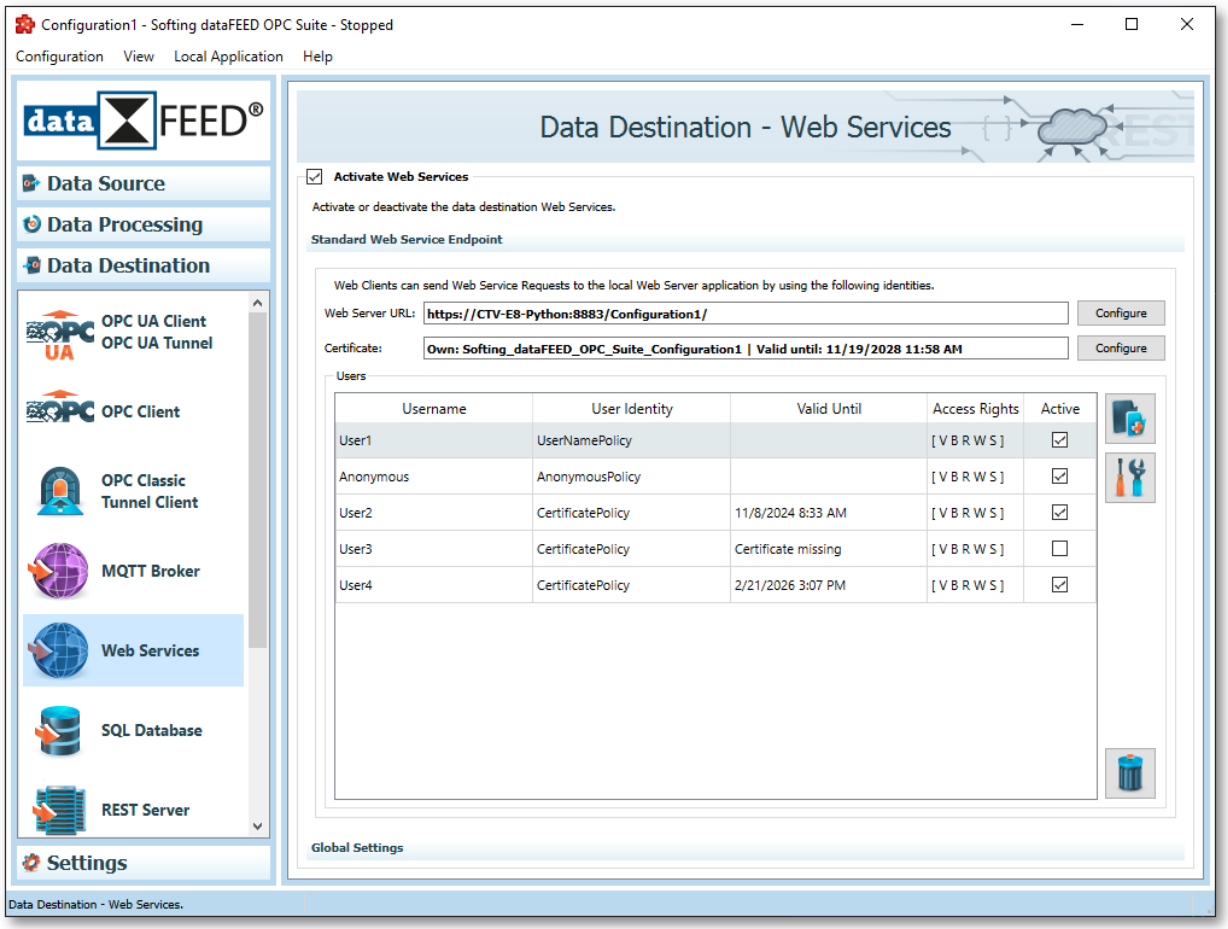
Enabling the data destination Web Services functionality allows external HTTPS clients to access the data items of the local dataFEED OPC Suite application. Only HTTPS clients are supported. No unsecured HTTP connections are available.

To enable this functionality the "Activate Web Services" check-box needs to be enabled/checked.



The Web Services Endpoint is enabled by the activation of "**Activate Web Services**". By default it is disabled. Being enabled, the server endpoint is open and accessible over the HTTPS protocol. Any HTTPS Clients will be able the establish connections to the exposed Server over the exposed endpoint and operate on standard services.

© Softing Industrial Automation GmbH



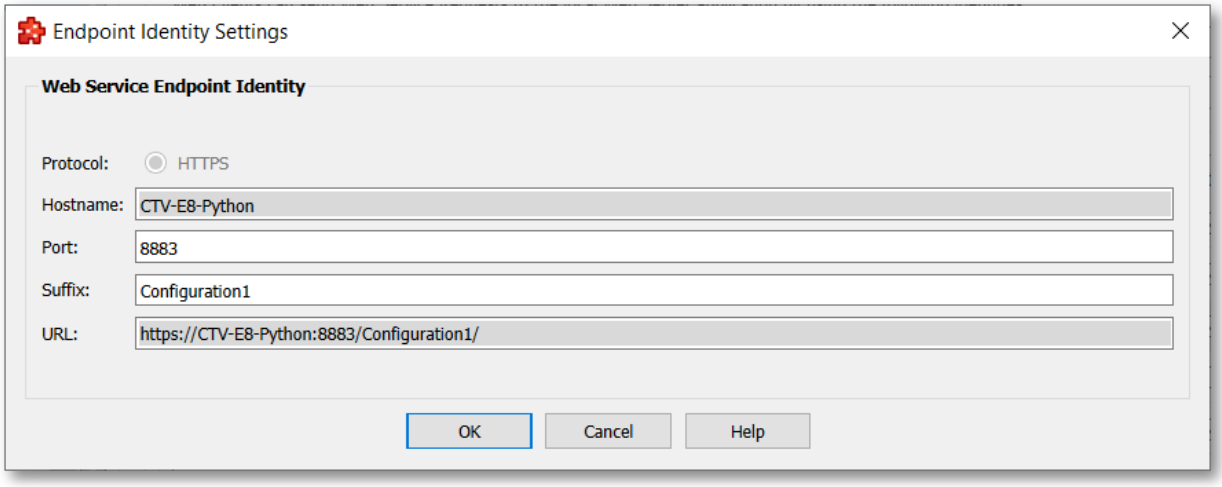
The following Web Services Endpoint specific settings are configurable:

- Server Endpoint identities
- Application Certificates
- User Authentication settings
- User Access Control Settings

Web Services Endpoint Identities:

The auto-generated Web Service endpoint URL is accessible on main configuration page, the access rights to it is limited to read only (copy operations being available).

The server endpoint URL is used by the HTTPS Client for communication establishment. The server endpoint URL and other basic endpoint communication identities are configurable via "**Endpoint Identity Settings**" dialog which is activating by pressings the "**Configure**" button.



Attribute:	Description:
Protocol	Current version of dataFEED OPC Suite (version V5.38) is supporting only HTTPS protocol. Unsecured HTTP is not supported.
Hostname	Configured Hostname on current machine. By default, it is using the value from the Operating System. The hostname is configurable at global configurable settings.
Port	Configured port for TCP communication on local machine. The port will be used by the HTTPS Client for connection establishment. By default, the configured port is auto-generated using an unique value for each defined dataFEED OPC Suite configuration.
	Attention: Please ensure that the configured port is not used by other running instance (in case that it was manually configured) or by other running application on current machine. Usage of same port on TCP communication by 2 different processes is not allowed by the Operating System.
Suffix	Suffix is auto-generated and used only for an easy identification of the application based on the exposed URL. It is an active part of the exposed functional requests.
URL	The endpoint URL is auto-generated using the previously specified parameters. It is read only - the adjustments being done by modifying the other parameters.

Application Certificates

The Web Service endpoint has assigned an own Public Key Infrastructure (PKI) store containing:

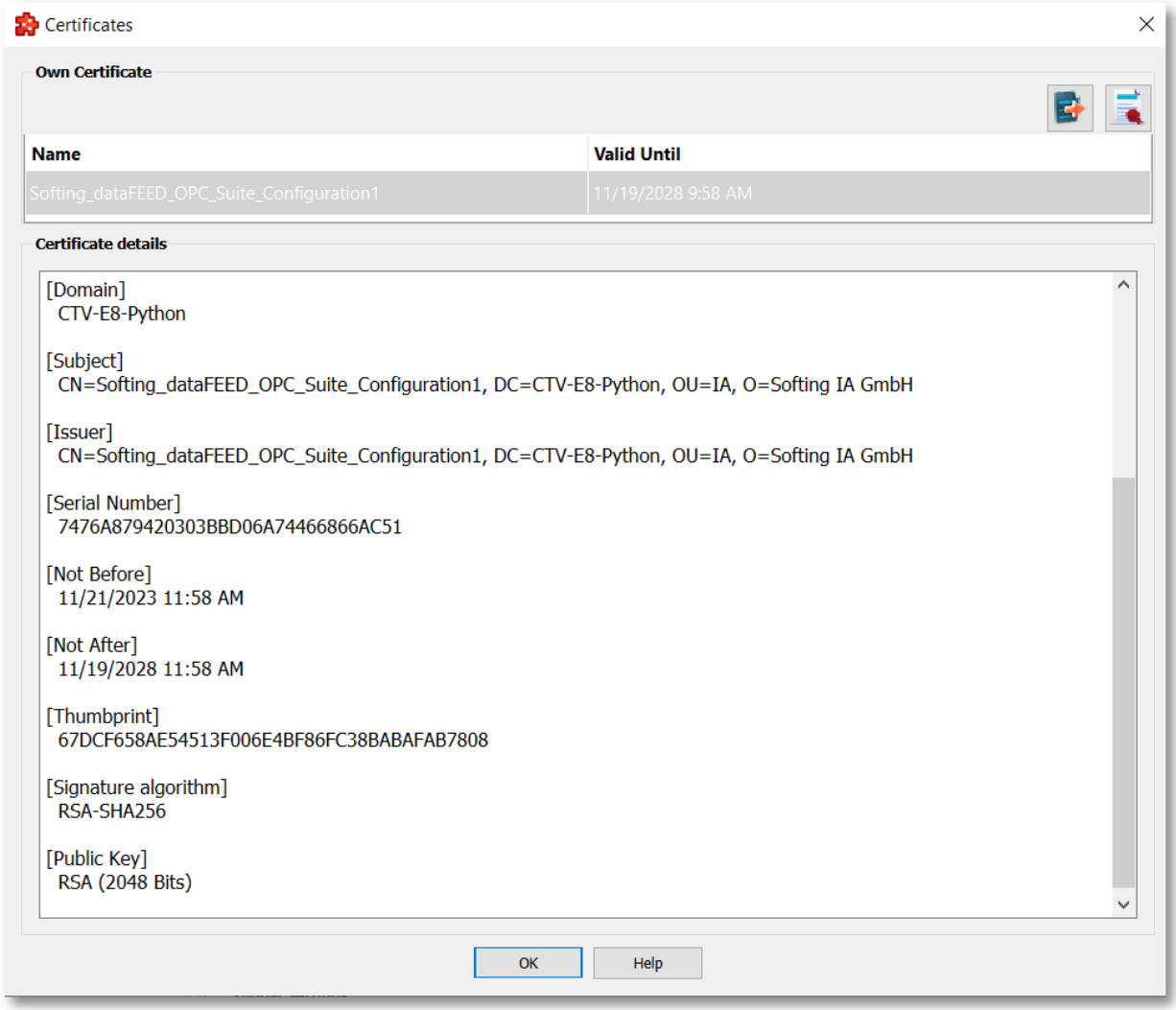
© Softing Industrial Automation GmbH

- Own Self Signed Certificate including a private key (format used: Distinguished Encoding Rules (.der) with Privacy Enhanced Mail Certificate (.pem)).

The PKI Store is auto-generated at the creation time of current configuration (for default configuration is it created at the installation time). The validity of the own certificate is 5 years.

Attention: The application certificate can be validated by the HTTPS Client at the connection establishment time. The own certificate is auto regenerated when an internal validation is failing (sample: changing the Hostname on working machine required a change of the used hostname on own certificate - re-generation automatically executed).

The PKI store can be manipulated (re-generation of own certificate) by pressing the corresponding "**Configure**" button on Certificates.



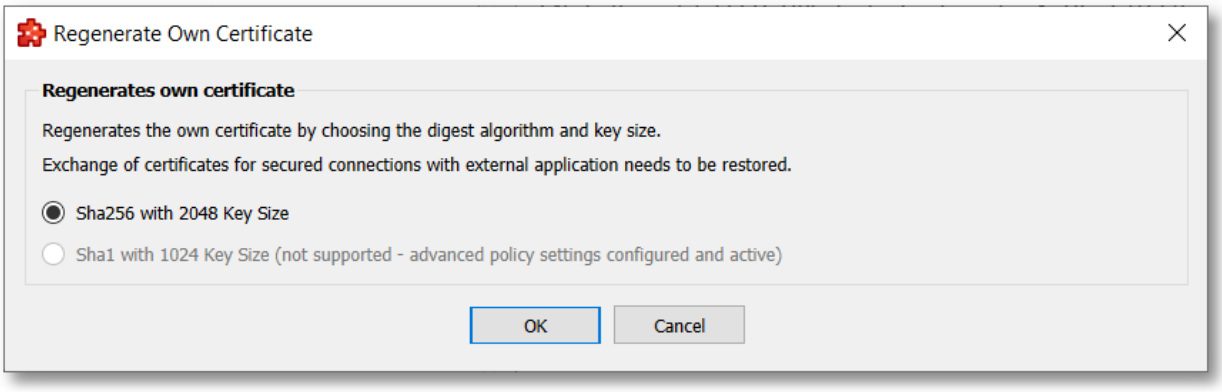
Main column is populated with the details of the own certificate.

Own Certificate.

This list contains only one certificate and offers the possibility to be exported.

By using the export button the user is prompted to select a destination folder where to save dataFEED OPC Suite Web Services application certificate. This functionality is particularly useful when the clients needs the server certificate to be validated.

The regeneration of own certificate is allowed by using the button. The own certificate can be generated by specifying the digest algorithm and encryption key size. The own certificate is regenerated using SHA256 digest algorithm and 2048 encryption key size independent of the settings from the previous certificate. Be aware, after this operation, the application own certificate was regenerated and all secured connections to the application should be reconfigured - the trust relation between application and connected servers/clients should be renewed.



Attention: The re-generation of certificates without saving the configuration is corrupting the actual configuration (a regeneration of certificates is overwriting the existing certificates - but the thumbprint is different (without saving the configuration, the application will try to use a certificate which was deleted).

The back-up/restore configuration operations are providing the possibility to operate also over the PKI store.

Attention: Restoring a configuration on a different machine, is requiring also a re-generation of own certificates due the reason that the identities (including hostname) were changed.

Certificate details.

In this area shows some details of the selected certificate like: location (on the disk), Application Name, Application Uri, Organization, Subject, Issuer, Validity dates and so on. The details area is read-only.

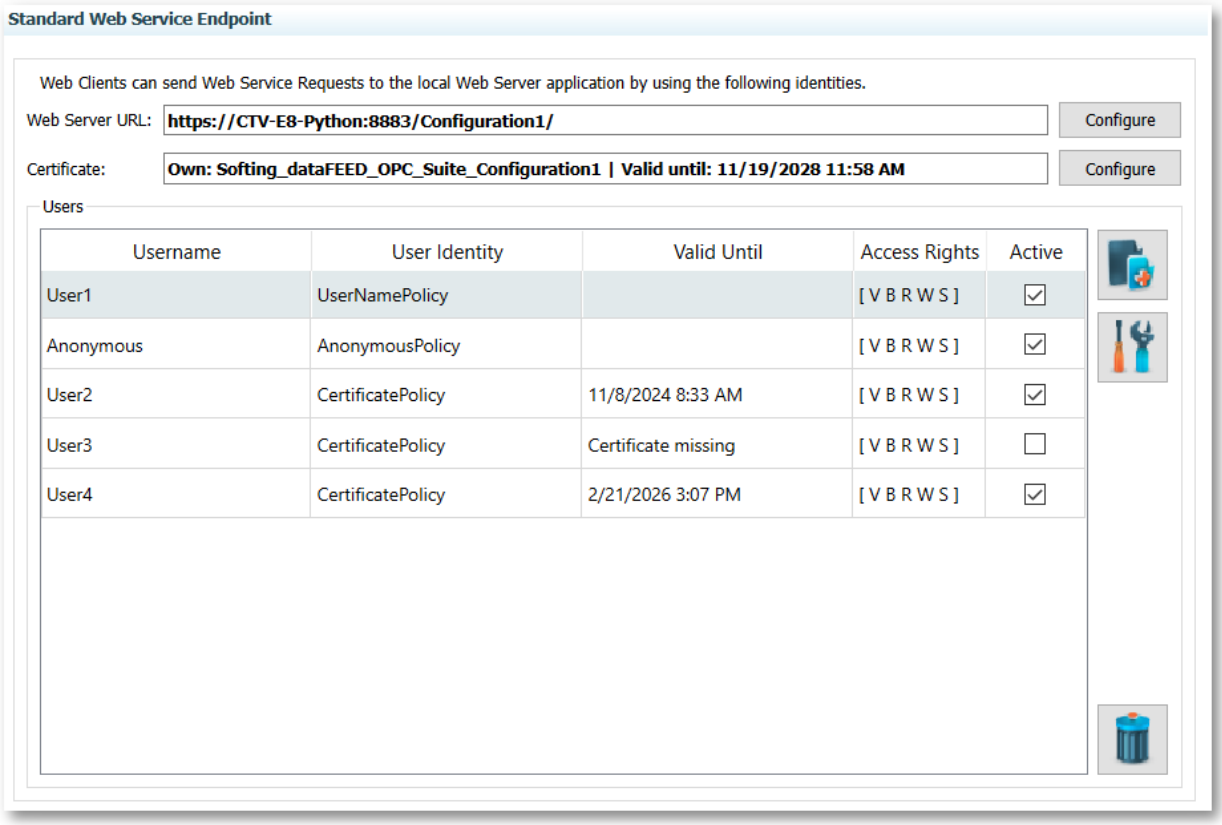
User Authentication Settings

© Softing Industrial Automation GmbH

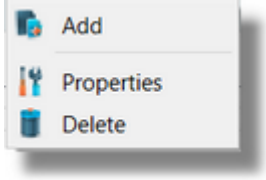
The user authentication settings are used by HTTPS Clients for building up a communication to dataFEED OPC Suite Endpoint. The access to dataFEED OPC Suite can be Anonymous or protected by using a user name and password, as well as by using a certificate.

By enabling anonymous policy, there is no authentication necessary. By default only the anonymous policy is enabled.

The application is supporting up to 50 configured active users. At least one active user should be configured - the application not allowing the deletion of the last active user.



Only the active users are used by the application, all the other ones being ignored at the application start up time.

The application allow the user to add, modify or remove an authentication user. This operations are available based on right buttons:  (add),  (modify) or  (remove) - or based on a context menu (right click on user definition table) 

.

Adding/Modifying a user is operated by a designed user definition wizard page - responsible for user identification settings.

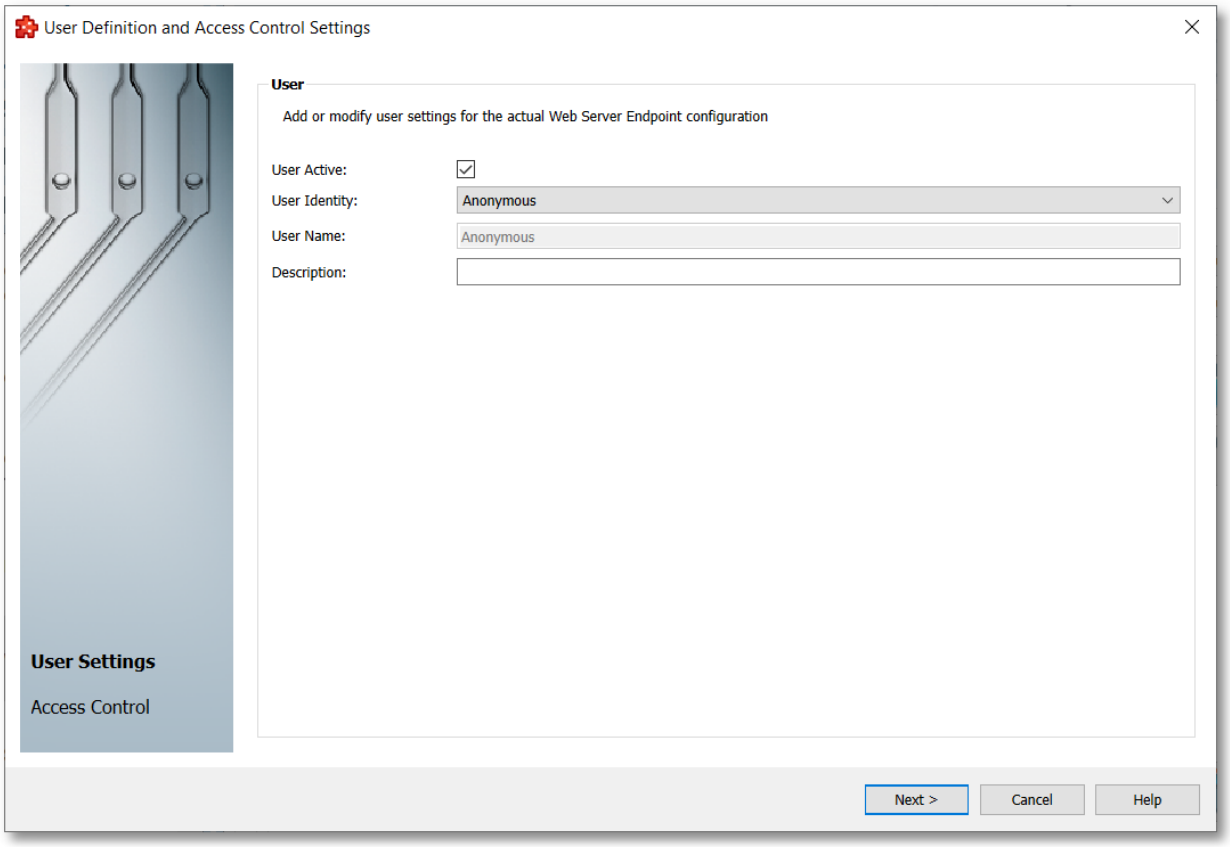
The application is supporting the following user authentication setting:

- **Anonymous** (no authentication being necessary).
- **Username and Password** (the authentication is executed based on a set of credentials (user-name/password)
  - for Username/Password authentication dataFEED uses the HTTP "Basic" authentication scheme.
  - the "Basic" HTTP authentication scheme is defined in RFC 7617, which transmits credentials as user ID/password pairs, encoded using base64.

Attention: The authentication based on user-name and password is using case sensitive policy.

- **Certificate** (the authentication is executed based on a user certificate sent from client with the requests).

Anonymous user



The defined user can be activate/deactivated (all the inactive users are ignored). By default, on a clean configuration, the Anonymous User is defined and active.

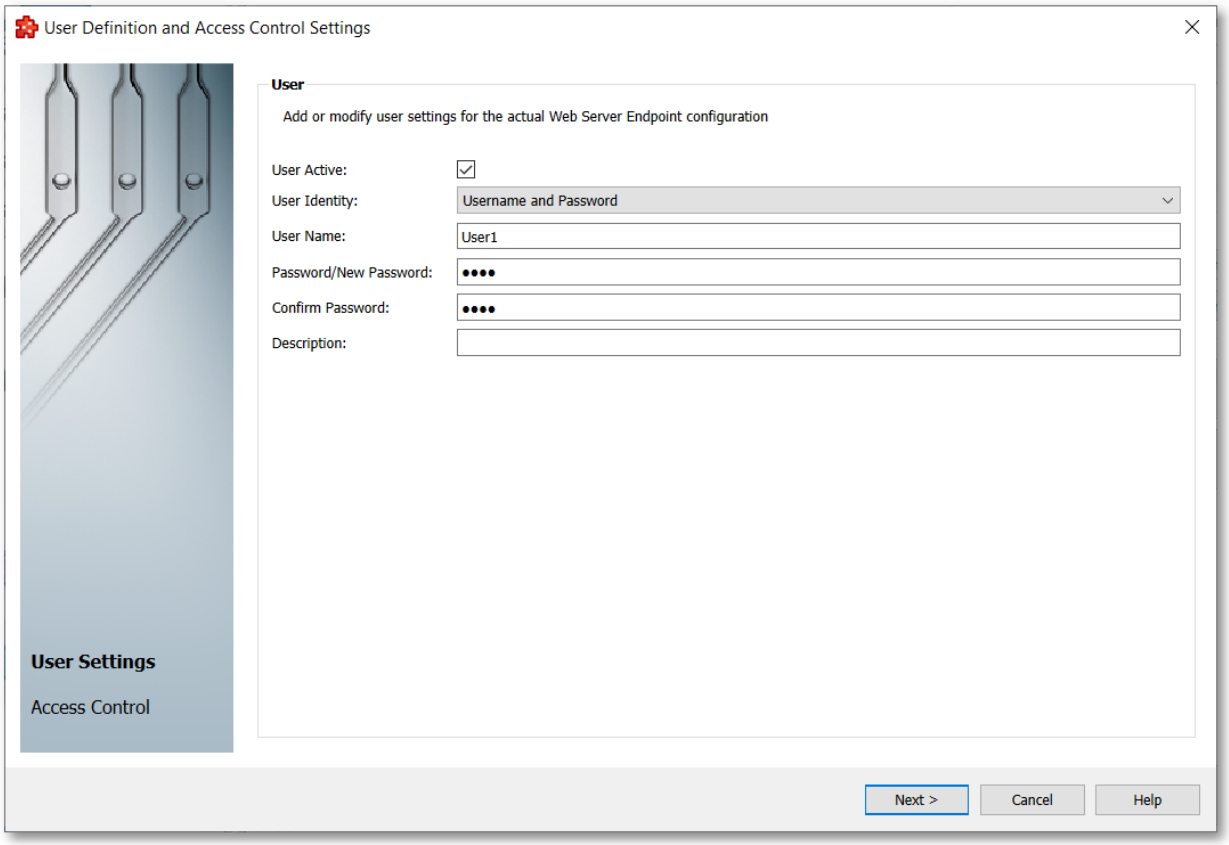
Note: Only one Anonymous User is possible being defined.

The "User Identity" type can be changed to:

© Softing Industrial Automation GmbH

- Username and Password
- Certificate

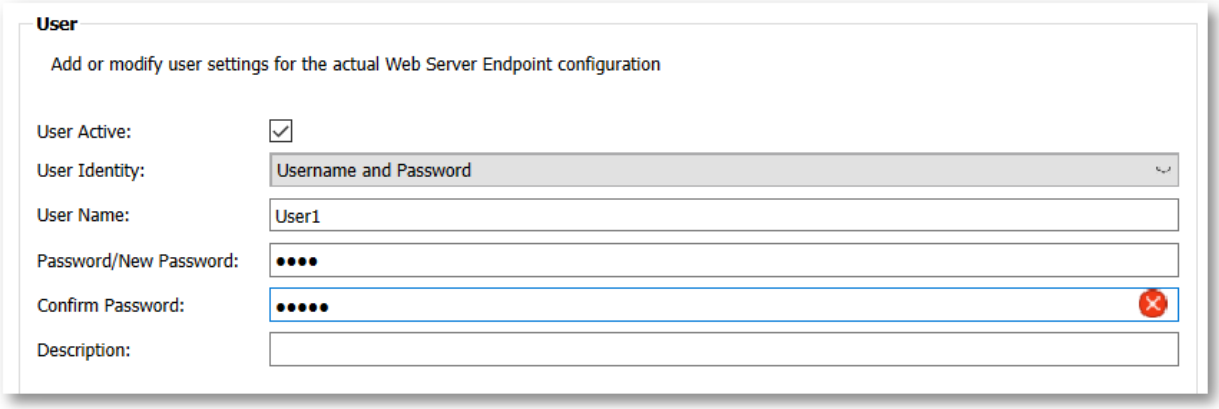
User identified by a set of credentials (user-name and password).



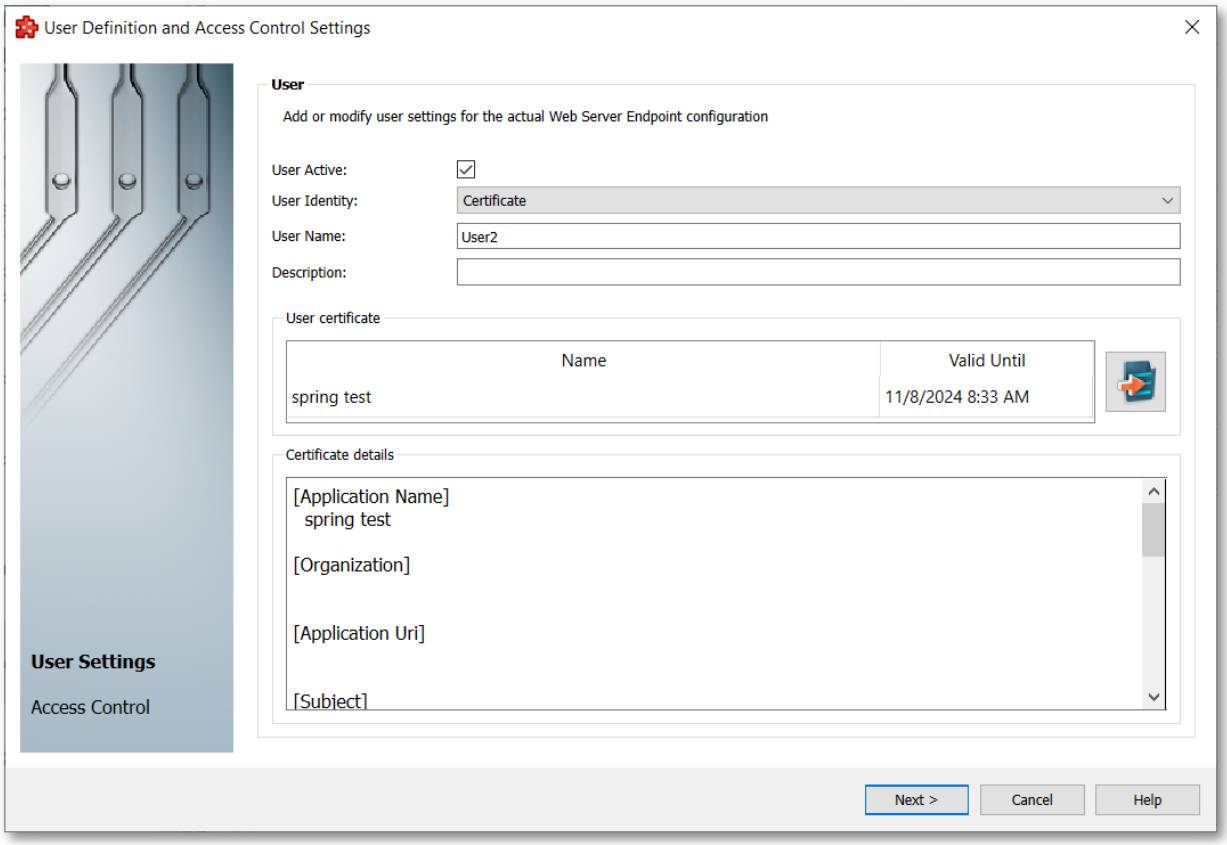
The user can be activate/deactivated (all the inactive users are ignored).

The user-name should be unique (it is not allowed to define duplicate users even if the specified password or identity type is different).

The confirmation of the password is mandatory - in case of failure, the user definition cannot be saved and the reason is visible on the wizard page:



User identified by a specified certificate



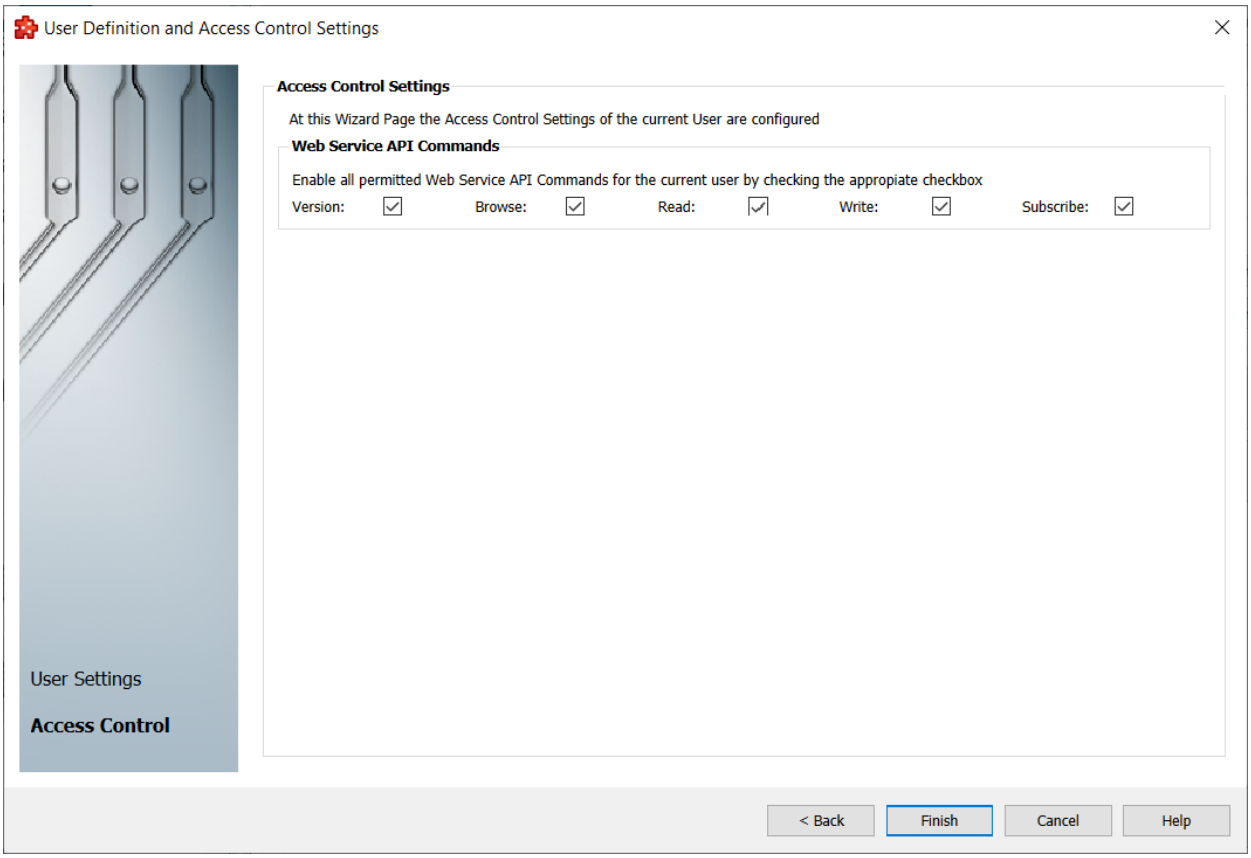
The user can be activate/deactivated (all the inactive users are ignored).

Importing a user certificate is mandatory. The supported certificates formats are: .der and .pfx. After a successfully import, the details of the imported certificates are visible on the wizard page.

User Access Control Settings

© Softing Industrial Automation GmbH

In this wizard page the specified user access control settings can be set, each functionality can be enabled or disabled separately.

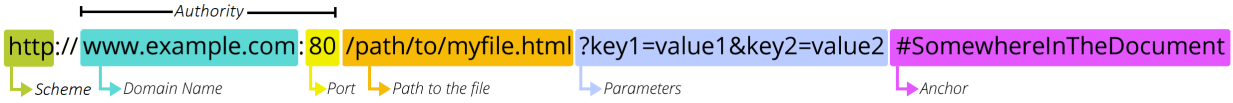


At least one service must be enabled. All functions can't be disabled in the same time.

Licensing

The functionality is licensed by **dataFEED Web Service** or **dataFEED OPC Suite Extended** licenses. *Demo Mode* is working on full functionality for 72 hours.

General:



1. Scheme: support only **HTTPS**

2. Authority
   a. The domain indicates which Web server is being requested. Usually this is a domain name, but an IP address may also be used.

   b. The port indicates the technical "gate" used to access the resources on the web server. It is usually omitted if the web server uses the standard ports of the HTTP protocol (80 for HTTP and 443 for HTTPS) to grant access to its resources. Otherwise it is mandatory. In dataFEED OPC Suite case the ports are the endpoints which are configured in Runtime.xml.

3. Path to resource

   a. For the API functions it contains the Configuration name then /api/function. For example: **https://localhost:8882/Configuration1/api/version**

4. Parameters

   a. In general URIs as defined by RFC 3986 (see Section 2: Characters) may contain any of the following 84 characters: ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789-._~:/?#[]@!$&'()*+,;= Note that this list doesn't state where in the URI these characters may occur. Any other character needs to be encoded with the percent-encoding (%hh). Each part of the URI has further restrictions about what characters need to be represented by an percent-encoded word.

   b. Some API functions can be used with request bodies where bigger request size is needed. These request bodies are JSON encoded and are limited to 2MB in size.

5. Anchor

   a. Not used in dataFEED OPC Suite Web API

Functions summary:

- Version
- Browse
- Read
- Write
- Subscribe
- Unsubscribe

Version (GET request):

https://localhost:8882/Configuration1/api/version

© Softing Industrial Automation GmbH

Returns the current running version of dataFEED OPC Suite.

Browse (GET request):

1. The browsed item syntax is given in the URL parameter query (syntax parameter)
2. The browse result is returned as a JSON object
3. Example:
   a. Browse root:
      i. https://127.0.0.1:443/Configuration1/api/browse?syntax=
         1. response:


```
{
   "status_code":"Good",
   "nodes":[
      {
         "node_id":"Local Items",
         "data_type":"Null",
         "array_type":"Scalar",
         "description":"",
         "access_rights":"None",
         "node_type":"Branch"
      },
      {
         "node_id":"MQTTTestWandAnonym",
         "data_type":"Null",
         "array_type":"Scalar",
         "description":"",
         "access_rights":"None",
         "node_type":"Branch"
      },
      {
         "node_id":"Siemens_1",
         "data_type":"Null",
         "array_type":"Scalar",
         "description":"",
         "access_rights":"None",
         "node_type":"Branch"
      }
   ]
}
```


      ii. https://127.0.0.1:443/Configuration1/api/browse?syntax=Siemens_1
         1. response:

```
{
   "status_code":"Good",
   "nodes":[
      {
         "node_id":"Siemens_1.Inputs",
         "data_type":"Null",
         "array_type":"Scalar",
         "description":"",
         "access_rights":"None",
         "node_type":"Branch"
      },
      {
         "node_id":"Siemens_1.Outputs",
         "data_type":"Null",
         "array_type":"Scalar",
         "description":"",
         "access_rights":"None",
         "node_type":"Branch"
      },
      {
         "node_id":"Siemens_1.Counters",
         "data_type":"Null",
         "array_type":"Scalar",
         "description":"",
         "access_rights":"None",
         "node_type":"Branch"
      },
      {
         "node_id":"Siemens_1.Timers",
         "data_type":"Null",
         "array_type":"Scalar",
         "description":"",
         "access_rights":"None",
         "node_type":"Branch"
      },
      {
         "node_id":"Siemens_1.Memory",
         "data_type":"Null",
         "array_type":"Scalar",
         "description":"",
         "access_rights":"None",
         "node_type":"Branch"
      },
      {
         "node_id":"Siemens_1.DataBlocks",
         "data_type":"Null",
         "array_type":"Scalar",
         "description":"",
         "access_rights":"None",
         "node_type":"Branch"
      }
   ]
}
```


© Softing Industrial Automation GmbH

Read (GET/POST request):

1. Read item(s) from address-space
2. It supports also with POST request because in browsers mainly it is not possible to send GET requests with body
3. The Item syntax and index range(for arrays, if not specified the full array is retrieved) can be specified in the URL or in the request body
4. Example:
   a. Simple read of an item:
      - https://127.0.0.1:443/Configuration1/api/read?items[0][syntax]=Siemens_1.DataBlocks.Datatypes_All.Word0
      - response:

```
[
   {
      "node_id":"Siemens_1.DataBlocks.Datatypes_All.Word0",
      "data_type":"UInt16",
      "value":0,
      "status_code":"Good",
      "timestamp":"2023-07-27T09:03:01.009653993+03:00"
   }
]
```

   b. Read array items the index is given in the URL parameter query (index parameter):
      - https://127.0.0.1:443/Configuration1/api/read?items[0][syntax]=Siemens_1.DataBlocks.Datatypes_All_Arrays.Date_And_Time&items[0][index][start]=0&items[0][index][end]=5
      - response:

```
[
   {
      "node_id":"Siemens_1.DataBlocks.Datatypes_All_Arrays.Date_And_Time",
      "data_type":"DateTime",
      "value":[
         "1990-01-01T01:02:03+00:00",
         "1990-01-01T00:00:00.001230+00:00",
         "1990-01-01T00:00:00+00:00",
         "1990-01-01T00:00:00+00:00",
         "1990-01-01T00:00:00+00:00",
         "1990-01-01T00:00:00+00:00"
      ],
      "status_code":"Good",
      "timestamp":"2023-07-27T09:18:00.007191553+03:00"
   }
]
```

c. For large number of items the query part is limited by the client, for this reason the items are passed as a JSON object in the request body:

- https://127.0.0.1:443/Configuration1/api/read
  - request body:

```
{
   "items":[
      {
         "syntax":"Siemens_1.DataBlocks.Datatypes_All_Arrays.Word0",
         "index":null
      },
      {
         "syntax":"Siemens_1.DataBlocks.Datatypes_All_Arrays.Word1",
         "index":{
            "start":1,
            "end":2
         }
      },
      {
         "syntax":"Siemens_1.DataBlocks.Datatypes_All_Arrays.Byte",
         "index":null
      }
   ]
}
```

- response:

```
[
   {
       "node_id":"Siemens_1.DataBlocks.Datatypes_All_Arrays.Date_And_T
ime",
      "data_type":"DateTime",
      "value":[
         "1990-01-01T01:02:03+00:00",
         "1990-01-01T00:00:00.001230+00:00",
         "1990-01-01T00:00:00+00:00",
         "1990-01-01T00:00:00+00:00",
         "1990-01-01T00:00:00+00:00",
         "1990-01-01T00:00:00+00:00"
      ],
      "status_code":"Good",
      "timestamp":"2023-07-27T10:08:07.008169808+03:00"
   },
   {
      "node_id":"Siemens_1.DataBlocks.Datatypes_All_Arrays.Word0",
      "data_type":"UInt16",
      "value":[
         0,
         0,
```


© Softing Industrial Automation GmbH

```
                .....
               0,
               0,
               0,
               0,
               0,
               0
            ],
            "status_code":"Good",
            "timestamp":"2023-07-27T10:08:07.008169808+03:00"
         },
         {
            "node_id":"Siemens_1.DataBlocks.Datatypes_All_Arrays.Word1",
            "data_type":"UInt16",
            "value":[
               0,
               0
            ],
            "status_code":"Good",
            "timestamp":"2023-07-27T10:08:07.008169808+03:00"
         },
         {
            "node_id":"Siemens_1.DataBlocks.Datatypes_All_Arrays.Byte",
            "data_type":"UInt8",
            "value":[
               0,
               1,
                ..........
               2,
               3,
               0,
               0,
               0,
               0
            ],
            "status_code":"Good",
            "timestamp":"2023-07-27T10:08:07.008169808+03:00"
         }
      ]
```


Write (POST request):

1. Write item(s) value(s) into address-space
2. Values can be written by URL query or with JSON body requests
3. Examples:
   - https://127.0.0.1:8883/Configuration1/api/write?items[0][syntax]=OPC_1.IOP.static.I1&items[0][value]=12
   - response:

```
[
    {
        "node_id": "OPC_1.IOP.static.I1",
        "data_type": "Int8",
        "status_code": "Good",
        "data_value": 12
    }
]
```


- write with body: https://127.0.0.1:8883/Configuration1/api/write
- body:


```
{
    "items": [
        {
            "syntax": "OPC_1.maths.sin",
            "index": null,
            "value": 3.533694e+72
        }
    ]
}
```


- response:


```
[
    {
        "node_id": "OPC_1.maths.sin",
        "data_type": "Double",
        "status_code": "BadInternalError",
        "data_value": 3.5336940000e72
    }
]
```


Subscribe (GET/POST request) :

1. For data-change notification we use server-sent events notifications from server side: https://developer.mozilla.org/en-US/docs/Web/API/Server-sent_events/Using_server-sent_events
2. in the request we need to pass the "items" with "syntax" and "sampling_rate" parameters
3. The event responses are received under the "datachange" event id
4. After closing the connection the subscribed items are automatically unsubscribed
5. Example:
   - https://127.0.0.1:8881/Configuration1/api/subscribe?items[0][syntax]=Siemens_S71511-1PN.DataBlocks.Datatypes_All_Arrays.WString&items[0][sampling_rate]=1000
   - response:

© Softing Industrial Automation GmbH

```
[
   {
      "node_id":"Siemens_S71511-1PN.DataBlocks.Datatypes_All_Arrays.WString",
      "data_type":"String",
      "value":[
         "item1",
         "item2",
         "item3",
         "",
         ""
      ],
      "status_code":"Good",
      "timestamp":"2023-08-08T15:06:27.003211828+03:00"
   }
]
```


6. Subscription request information can be sent also in the body of the request:


```
{
   "items":[
      {
         "syntax":"Siemens_1.Demo.Dynamic.Scalar.Int32",
         "sampling_rate":1000
      },
      {
         "syntax":"Siemens_1.Demo.Dynamic.Scalar.Int64",
         "sampling_rate":1000
      }
   ]
}
```


Unsubscribe (GET request):

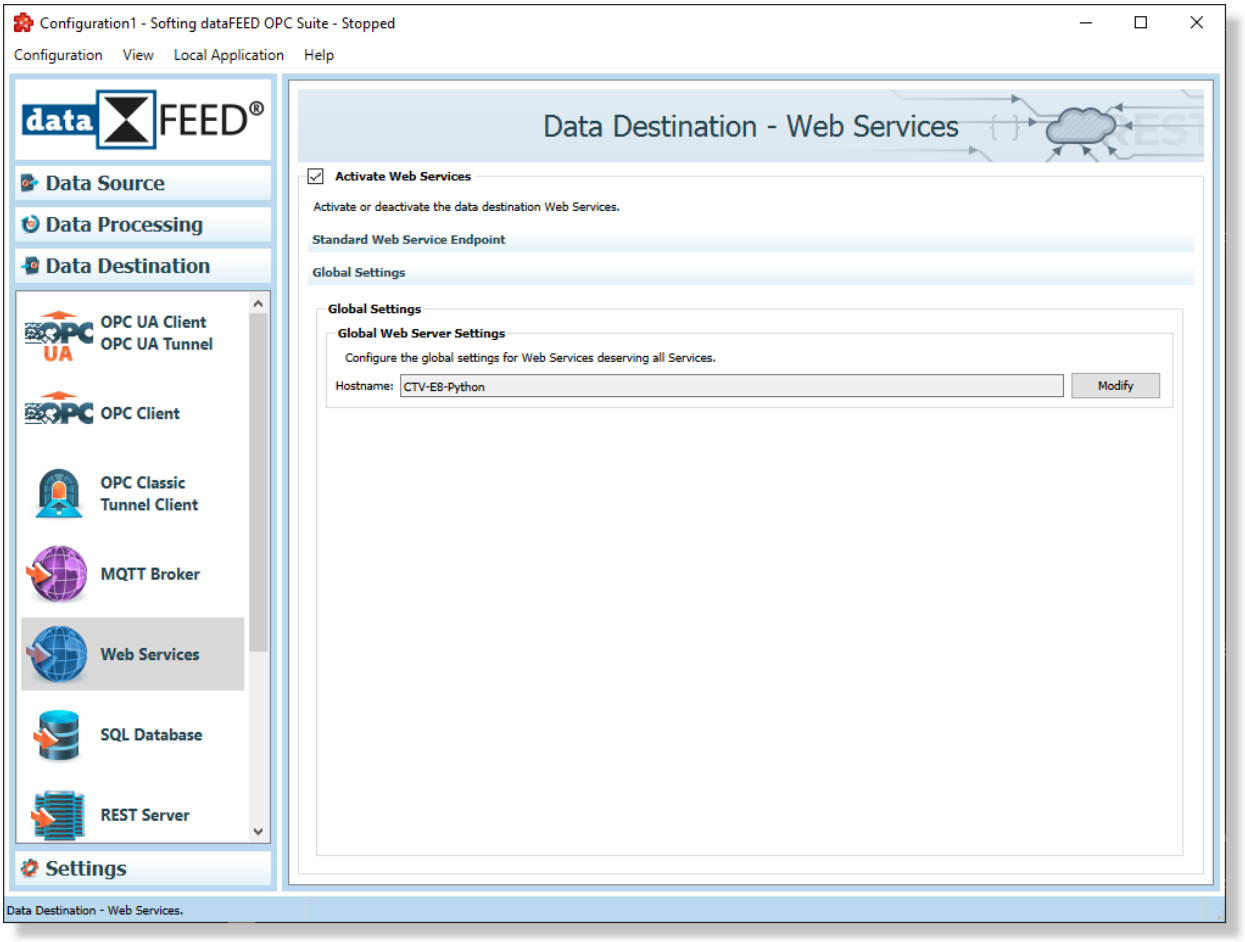
1. To manually unsubscribe from data-change notifications we use the "unsubscribe" endpoint with the "items"and "syntax" parameter

2. Example: https://127.0.0.1:8881/Configuration1/api/unsubscribe?items[0][syntax]=Siemens_S71511-1PN.DataBlocks.Datatypes_All_Arrays.WString

   - response:


```
{
   "status_code":"GoodCompletesAsynchronously",
   "nodes":[
      {
         "node_id":"Siemens_S71511-1PN.DataBlocks.Datatypes_All_Arrays.WString",
         "status_code":"Good"
      }
   ]
}
```

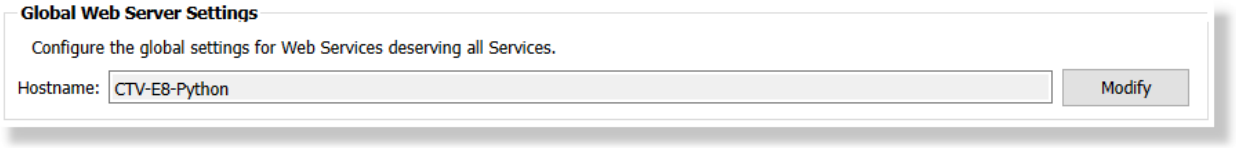

Basically these settings are applied on all server endpoints without a possibility to apply different changes on different server endpoints.



The following global settings are configurable:

- Hostname as Global Web Services Settings.

Hostname

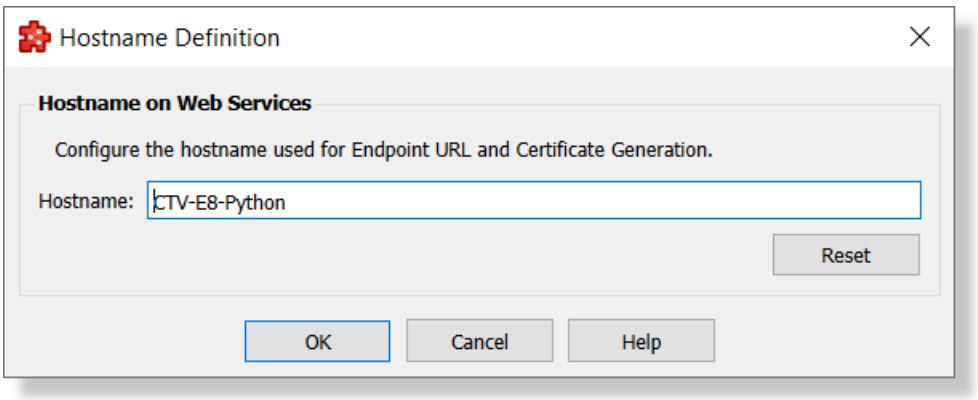


By default the application is using the current machine hostname obtained from the Operating System.

This can be changed to a user defined hostname (it can be used the network domain as prefix or a totally different hostname).

The hostname can be changed by opening the **Hostname Definition** dialog:

© Softing Industrial Automation GmbH



The machine hostname can be received from the operating system by pressing **Reset** button.

Attention: Changing the hostname used by the Web Services is requiring a re-generation of all application certificates (operation which is done automatically). This operation is requiring (where needed) a redone of trust relation between the exposed Web Services and HTTPS Clients (this operation should be done manually by exporting/importing application certificates).

Attention: The local host name should contain exclusively ASCII characters. Usage of special/language related characters (sample: Ä,Ö, Ü, ? , ä, ö, ü, ß) are recommended being not used as host name by the operation system (Microsoft Windows 7 or later)

7.3.3.9.2 Samples

At this topic there are samples for different web service clients (Python, Java Spring Framework), command line utilities (Curl) , GUI web clients (Postman) and Web Browsers.

Details and descriptions of the samples can be found at dataFEED OPC Suite install location under **"samples\WebServices"** directory (ex. "**C:\Program Files (x86)\Softing\dataFEED OPC Suite\samples\WebServices**").

Edge Browser

Chrome Browser

Postman

Python

Curl

Spring Framework

Steps to connect with Edge Web Client to dataFEED OPC Suite **Web server:**

Preconditions:

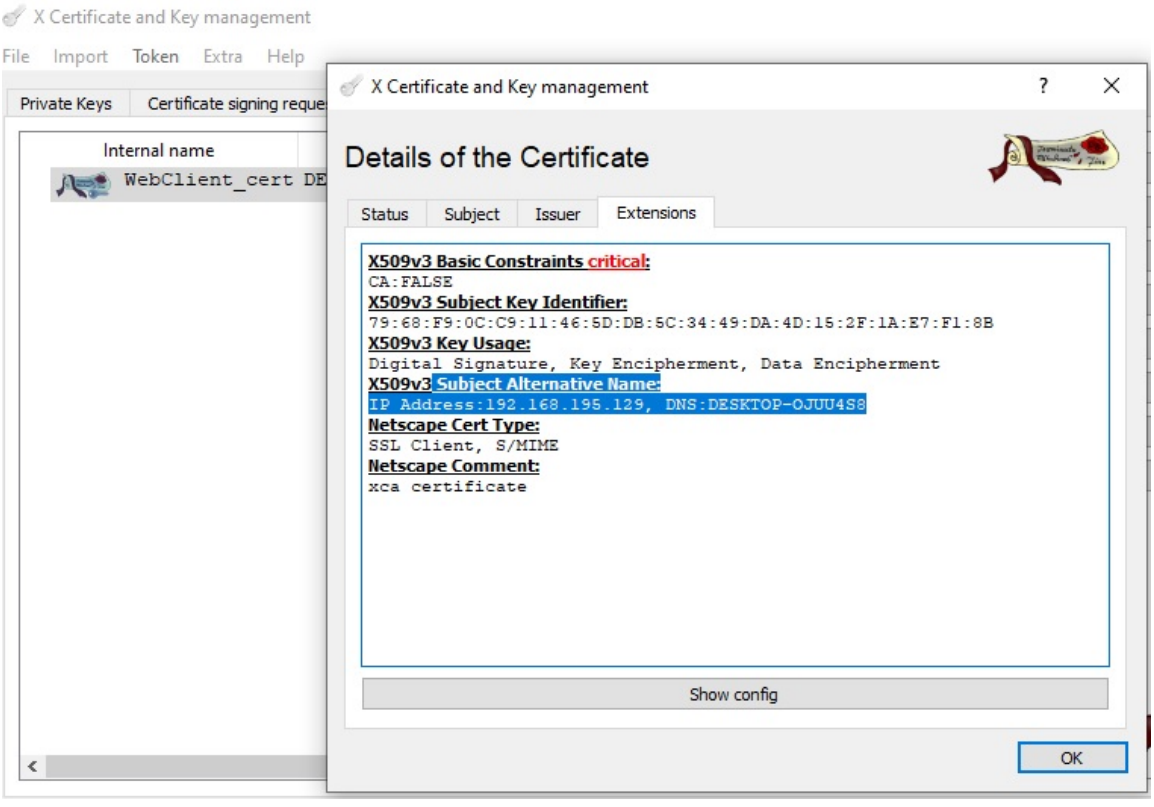
- Install/Update Edge browser

- Install extension: Boomerang - SOAP & REST Client for edge browser
- Install dataFEED OPC Suite V5.35 or newer version
- Generate .pfx Client certificate
- Optionally restore WebServices_Test.osf configuration from EdgeWebServicesTests.zip file (found at dataFEED OPC Suite install location under **"samples\WebServices"** directory (ex. "**C:\Program Files (x86)\Softing\dataFEED OPC Suite\samples\WebServices**").

Generate .pfx Client certificate

Before sending the requests using Certificate user, the client certificates should be generated. Client certificate should contain **Subject Alternative Name** with host IP and DNS – Host PC Name:

IP Address:127.0.0.1, DNS:pc_name



Create **.pfx** client certificate using the generated client certificate and client key. The following openssl command can be used:

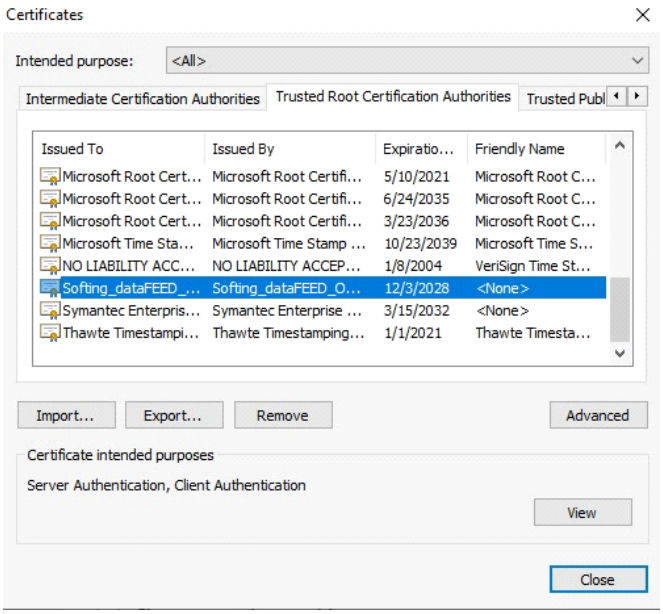
openssl pkcs12 -export -out WebClient_cert.pfx -inkey WebClient_cert.pem -in WebClient_cert.crt

© Softing Industrial Automation GmbH

Note: .pfx client certificate should be used also by dataFEED web service user with certificate authentication.

Configure dataFEED OPC Suite Web Services:

1. Open dataFEED OPC Suite Configurator and go to Data Destination Web Services
2. Enable Data Destination Web Services and add users with all authentication types: Anonymous, Username and Password, Certificate
3. Add Data Source OPC Server connection (or any other data source with an address space)
4. Start dataFEED OPC Suite Runtime



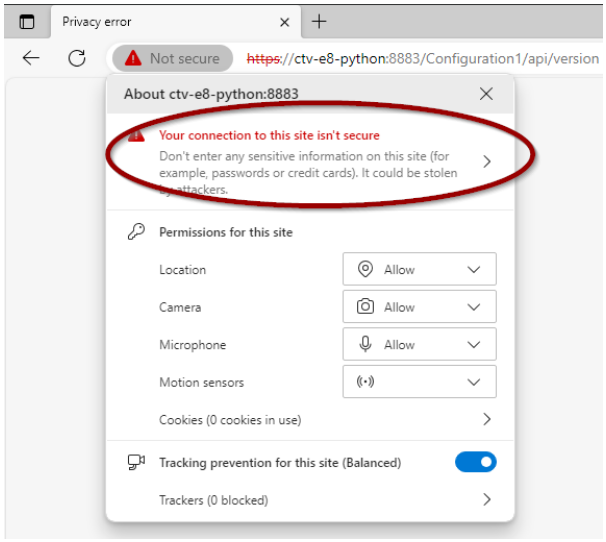
Add dataFEED Web Server Certificate into Edge Trusted Root Certification Authorities:

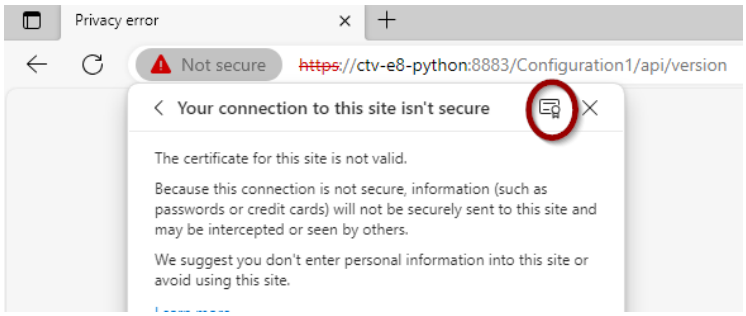
1. Open Edge Browser and go to Settings -> Manage Certificates
2. Open Trusted Root Certification Authorities tab
3. Import the Web Server own certificate in Trusted Root Certification Authorities

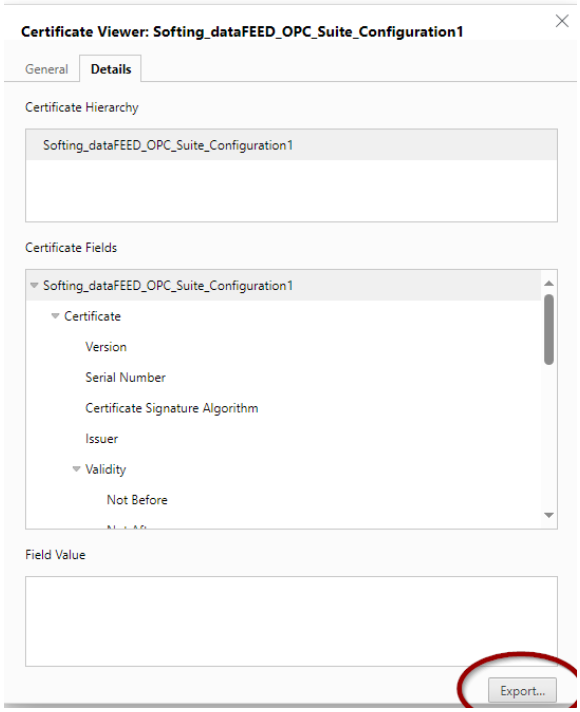
Note: Web Server own certificate can be found in path:

```
C:\ProgramData\Softing\dataFEED OPC Suite\ConfigurationName\PKI_SECURITY\WebServer\own
```

Or it can be saved from a connection to the server:







For executing complex (Json body encoded, user authentication) https request with a browser we implemented some sample web page which contains JavaScript which do the https calls. Also, some third-party browser extensions can be used to execute queries, for example (Boomerang - SOAP & REST Client).

© Softing Industrial Automation GmbH

Sample HTML pages with JavaScript are archived with current documents and can be found at dataFEED OPC Suite install location under "samples\WebServices\EdgeWebServicesTests.zip" directory (ex. "c:\Program Files (x86)\Softing\dataFEED OPC Suite\samples\WebServices\EdgeWebServicesTests.zip").

- **Send requests using Anonymous user:**

- Run the following JavaScript: RunRequests_Anonymous.html
  - It contains samples for retrieving the Version, Browse the address-space, Read, Write and Subscribe items.

- **Send requests using Username and Password user:**

- For Username/Password authentication dataFEED uses the HTTP "Basic" authentication scheme.
- The "Basic" HTTP authentication scheme is defined in RFC 7617, which transmits credentials as user ID/password pairs, encoded using base64.
- Examples are simplified in the following HTML/JavaScript: RunRequests_UserPass.html
  - It contains samples for retrieving the Version, Browse the address-space, Read, Write and Subscribe items.

For authentication the following header should be added to the requests: Authorization: Basic <credentials>

For the examples which use the htmx library the authentication is specified in the following way: *hx-headers='{"Authorization": "Basic VXNlcjE6cGFzcw=="}*

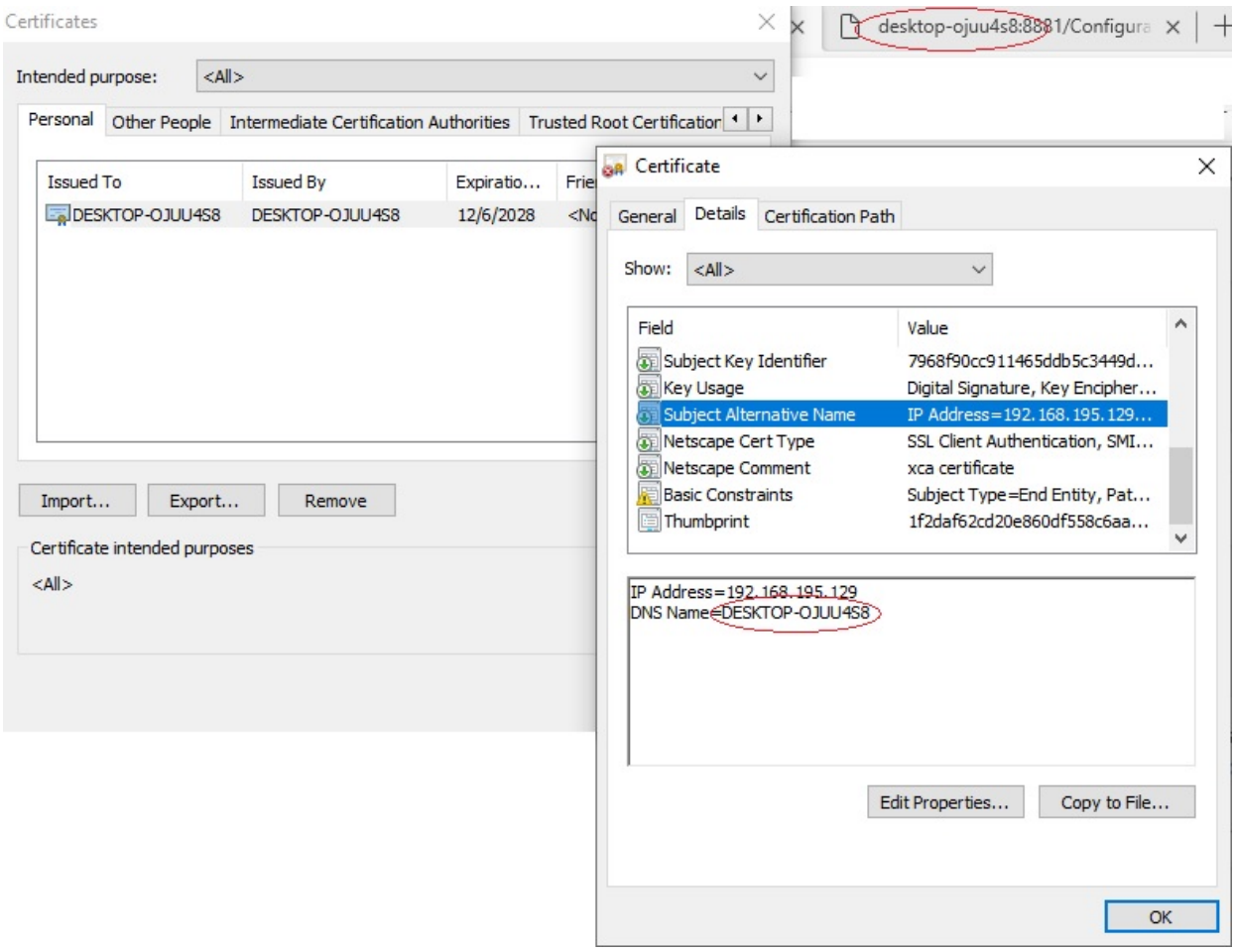
The Authorization data need to be base64 encoded (ex. Online encode/decode https://www.base64decode.org/)

- **Send requests using Certificate user:**

Add Client Certificate into Edge Personal certificates:

- Open Edge Browser and go to Settings -> Manage Certificates
- Open Personal tab
- Import the .pfx web client certificate

Note: The DNS from client certificate should be the same with the host from URL requests.



- From edge browser send the following request:

https://desktop-ojuu4s8:8881/Configuration1/api/version

- Press the padlock from left side of request
- Select Your certificate choices -> Change certificate
- Select the client certificate for authentication and press OK button

The request should be sent successfully with certificates.

- Run the following JavaScript: RunRequests_Anonymous.html
  - It contains samples for retrieving the Version, Browse the address-space, Read, Write and Subscribe items.

Useful steps for deleting Certificates from Settings -> Manage Certificates tabs :

- Go to Manage computer certificates
- Delete certificates from Personal -> Cerificates
- Delete certificates from Other People -> Cerificates

© Softing Industrial Automation GmbH

Steps to connect with Chrome Web Client to dataFEED OPC Suite **Web server:**

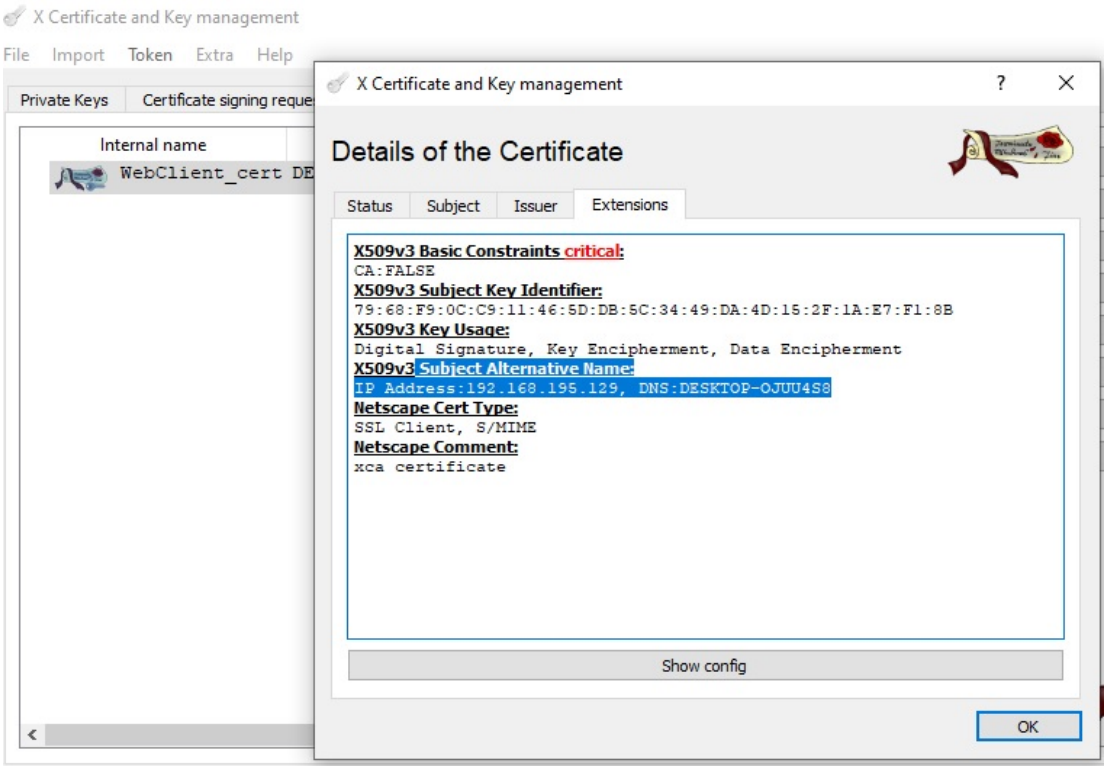
Preconditions:

- Install/Update Chrome browser
- Install extension: ex. Boomerang - SOAP & REST Client for Chrome browser
- Install dataFEED OPC Suite V5.35 or newer version
- Generate .pfx Client certificate
- Optionally restore WebServices_Test.osf configuration from ChromeWebServicesTests.zip file (found at dataFEED OPC Suite install location under **"samples\WebServices"** directory (ex. "**C:\Program Files (x86)\Softing\dataFEED OPC Suite\samples\WebServices**").

Generate .pfx Client certificate

Before sending the requests using Certificate user, the client certificates should be generated. Client certificate should contain **Subject Alternative Name** with host IP or DNS – Host PC Name:

IP Address:127.0.0.1, DNS:pc_name

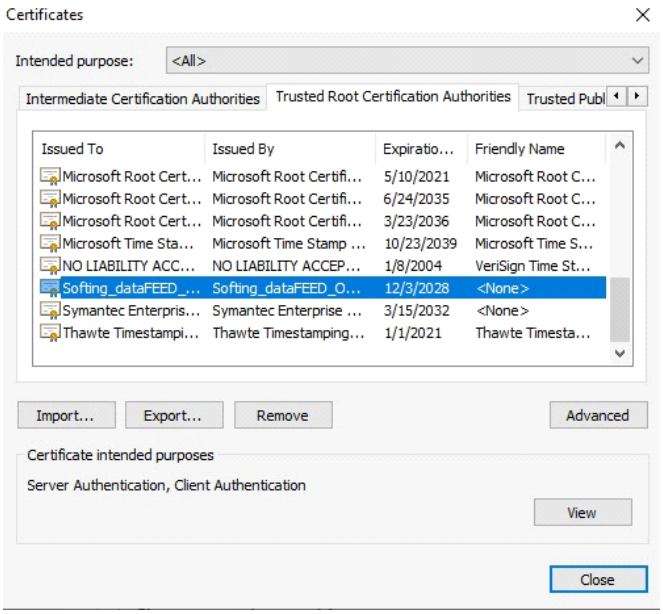


Create **.pfx** client certificate using the generated client certificate and client key. The following openssl command can be used:

openssl pkcs12 -export -out WebClient_cert.pfx -inkey WebClient_cert.pem -in WebClient_cert.crt

Configure dataFEED OPC Suite Web Services:

1. Open dataFEED OPC Suite Configurator and go to Data Destination Web Services
2. Enable Data Destination Web Services and add users with all authentication types: Anonymous, Username and Password, Certificate (use .pfx certificate)
3. Add Data Source OPC Server connection (or any other data source with an address space)
4. Start dataFEED OPC Suite Runtime



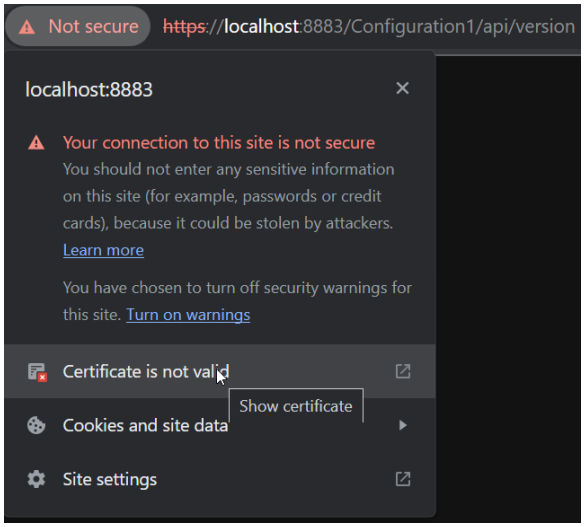
Add dataFEED Web Server Certificate into Chrome Trusted Root Certification Authorities:

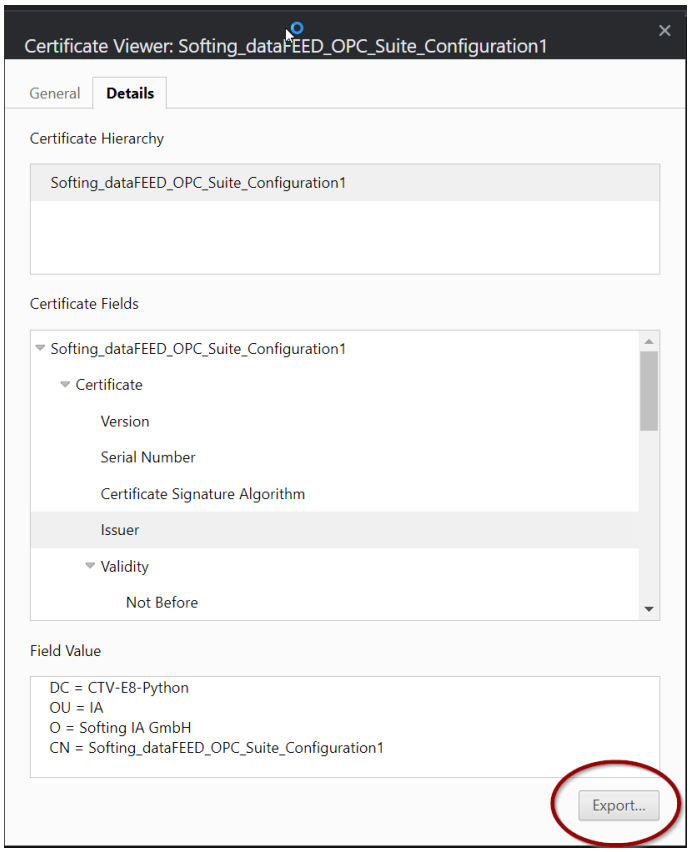
1. Open Chrome Browser and go to Settings -> Privacy and security -> Security -> Manage certificates
2. Open Trusted Root Certification Authorities tab
3. Import the Web Server own certificate in Trusted Root Certification Authorities

Note: Web Server own certificate can be found in path: C:\ProgramData\Softing\dataFEED OPC Suite\ConfigurationName\PKI_SECURITY\WebServer\own

Or it can be saved from a connection to the server:

© Softing Industrial Automation GmbH





For executing complex (Json body encoded, user authentication) https request with a browser we implemented some sample web page which contains JavaScript which do the https calls. Also, some third-party browser extensions can be used to execute queries, for example (Boomerang - SOAP & REST Client).

Sample HTML pages with JavaScript are archived with current document and can be found at dataFEED OPC Suite install location under "samples\WebServices\ChromeWebServicesTests.zip" directory (ex. "c:\Program Files (x86)\Softing\dataFEED OPC Suite\samples\WebServices\ChromeWebServicesTests.zip").

- **Send requests using Anonymous user:**

- Open Boomerang extension and send a HTTP Request

Ex: https://DESKTOP-OJUU4S8:8881/Configuration1/api/version

- Or run the following JavaScript: RunRequests_Anonymous.html
- It contains samples for retrieving the Version, Browse the address-space, Read, Write and Subscribe items.

- **Send requests using Username and Password:**
  - For Username/Password authentication dataFEED uses the HTTP "Basic" authentication scheme.
  - The "Basic" HTTP authentication scheme is defined in RFC 7617, which transmits credentials as user ID/password pairs, encoded using base64.
  - Examples are simplified in the following HTML/JavaScript: RunRequests_UserPass.html
    - It contains samples for retrieving the Version, Browse the address-space, Read, Write and Subscribe items.

For authentication the following header should be added to the requests: Authorization: Basic <credentials>

For the examples which use the htmx library the authentication is specified in the following way: *hx-headers='{"Authorization": "Basic VXNlcjE6cGFzcw=="}*

The Authorization data need to be base64 encoded (ex. Online encode/decode https://www.base64decode.org/)

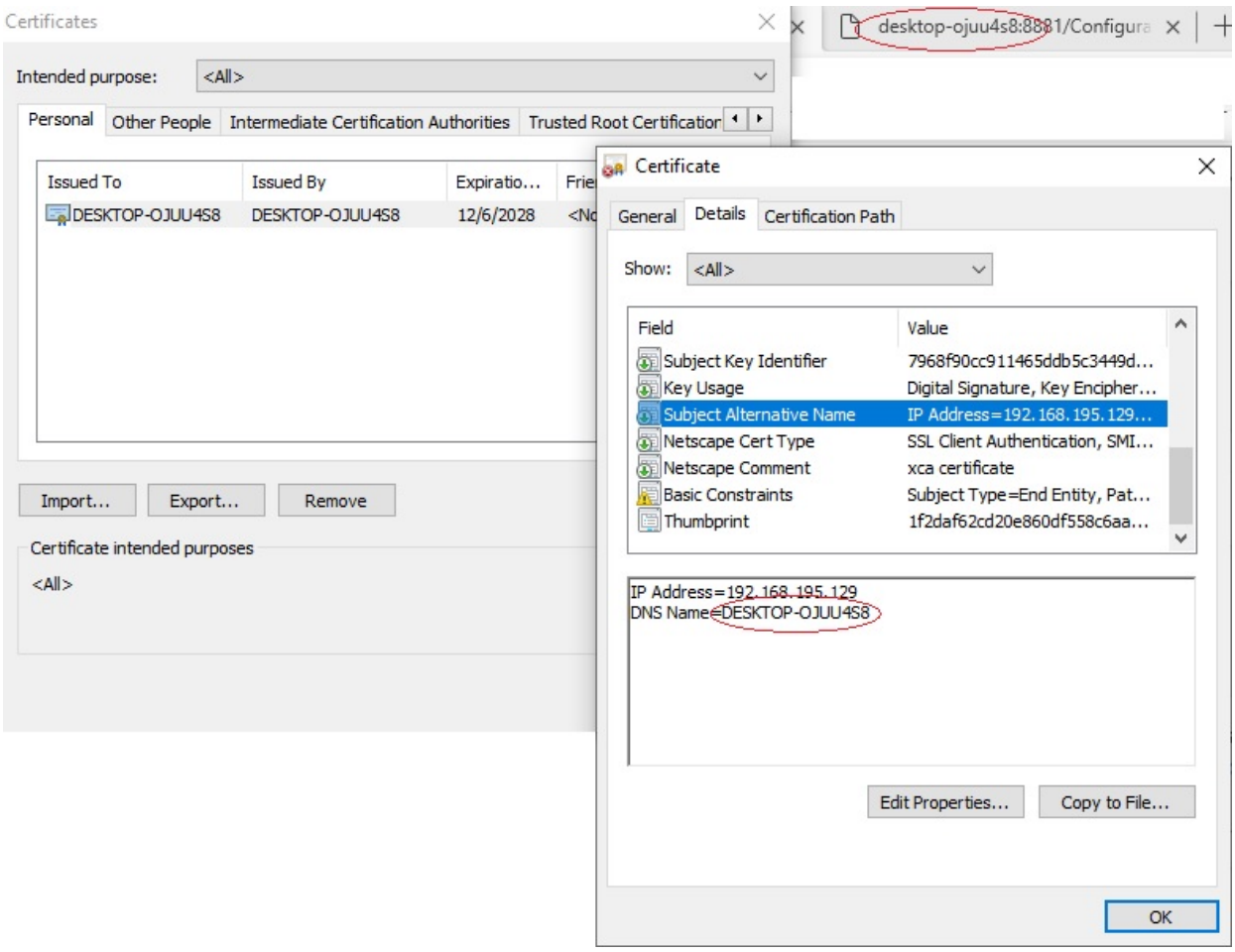
- **Send requests using Certificate user authentication:**

Add Client Certificate into Chrome Personal certificates:

- Open Chrome Browser and go to Settings -> Privacy and security -> Security -> Manage Certificates
- Open Personal tab
- Import the .pfx web client certificate

Note: The DNS from client certificate should be the same with the host from URL requests.

© Softing Industrial Automation GmbH



- Open Boomerang extension and send a HTTP Request

Ex: https://DESKTOP-OJUU4S8:8881/Configuration1/api/version

- Accept the client certificate that is required to authenticate to the host

https://desktop-ojuu4s8:8881/Configuration1/api/version

- The request should be sent successfully with certificates.
- Or run the following JavaScript: RunRequests_Anonymous.html
- It contains samples for retrieving the Version, Browse the address-space, Read, Write and Subscribe items.

Useful steps for deleting Certificates from Settings -> Manage Certificates tabs :

- Go to Manage computer certificates
- Delete certificates from Personal -> Cerificates
- Delete certificates from Other People -> Cerificates

Steps to connect with Postman Web Client to dataFEED OPC Suite **Web server:**

Preconditions:

- Install dataFEED OPC Suite V5.35 or newer version
- Generate .pfx Client certificate
- Optionally restore WebServices_Test.osf configuration from PostmanWebServicesTests.zip file (found at dataFEED OPC Suite install location under **"samples\WebServices"** directory (ex. "**C:\Program Files (x86)\Softing\dataFEED OPC Suite\samples\WebServices**").

Configure dataFEED OPC Suite Web Services:

1. Open dataFEED OPC Suite Configurator and go to Data Destination Web Services
2. Enable Data Destination Web Services and add users with all authentication types: Anonymous, Username and Password, Certificate (use .pfx certificate)
3. Add Data Source OPC Server connection (or any other data source with an address space)
4. Start dataFEED OPC Suite Runtime

Postman requests examples:

1. Version:

   - Returns the current running version of dataFEED OPC Suite.

   GET:

   https://localhost:8881/Configuration1/api/version

2. Browse:

   - Browse the address-space, item "*syntax*" specified as URL parameter

   - Retrieves the child nodes of the OPC_1 node (from OPC Demo Server)

   GET:

   https://localhost:8881/Configuration1/api/browse?syntax=OPC_1

© Softing Industrial Automation GmbH

3. Read (URL):

- Read item value, item "*syntax*" and optional "*index*" (start, end) specified in URL query or body of the request.

- Read one node (in *items* array only one node is specified), with syntax OP_1.IOP.dynamic.I1

GET/POST:

https://localhost:8881/Configuration1/api/read?items[0][syntax]=OPC_1.IOP.dynamic.I1

4. Read (JSON Body):

- Same as above but query parameters are specified in the body of the request as JSON.

- Read multiple items specified as JSON array (OPC_1.IOP.static.BOOL, OPC_1.IOP.static.I1, OPC_1.IOP.static.I2),

the *index* range is not specified so if the item is an array then all the elements are retrieved)

GET/POST:

https://localhost:8881/Configuration1/api/read

Body – raw JSON:

{"items": [{"syntax": "OPC_1.IOP.static.BOOL", "index": null}, {"syntax": "OPC_1.IOP.static.I1", "index": null}, {"syntax": "OPC_1.IOP.static.I2", "index": null}]}

5. Write (URL):

- Write item value, item id specified in "*syntax*" element. Optional "*index*" (start, end) for array range. Item value is set in "*value*" parameter.

- One item (with *syntax* element) is specified with one value (*value* element) to be written.

POST:

https://localhost:8881/Configuration1/api/write?items[0][syntax]=OPC_1.IOP.static.I1&items[0][value]=12

6. Write (JSON Body):

- Same parameters as above just the values are specified in the POST body as JSON object.

- Multiple elements to be written is specified each as separate JSON object as *"items"* array elements. Each object has "*syntax*", "*index*" and "*value*" specified.

POST:

https://localhost:8881/Configuration1/api/write

Body – raw JSON:

{"items": [{"syntax": "OPC_1.IOP.static.BOOL", "index": null, "value": false}, {"syntax": "OPC_1.IOP.static.I1", "index": null, "value": 50}, {"syntax": "OPC_1.IOP.static.I2", "index": null, "value": 50}]}

7. Subscribe (URL):

- Subscribe to items specified in "*syntax*" value. Also monitoring "*sampling_rate*" can be specified.

GET/POST:

https://localhost:8881/Configuration1/api/subscribe?items[0][syntax]
=OPC_1.IOP.dynamic.I1&items[0][sampling_rate]=1000

8. Subscribe (JSON Body)

- In the following example the subscription parameters are set in the body of the request.

- Items to be subscribed must be specified as JSON objects (with "*syntax*", "*sampling_rate*" values) in the "*items*" array.

GET/POST:

https://localhost:8881/Configuration1/api/subscribe

Body – raw JSON:

{"items": [{"syntax": "OPC_1.IOP.dynamic.BOOL", "sampling_rate": 1000}, {"syntax": "OPC_1.IOP.dynamic.I1", "sampling_rate": 1000}, {"syntax": "OPC_1.IOP.dynamic.I2", "sampling_rate": 1000}]}

The requests above can be sent with Anonymous, Username/Password or Certificate authentication:

- Send requests using Anonymous authentication:

© Softing Industrial Automation GmbH

- Authorization: No Auth / Inherit auth from parent
- Certificates: No certificate added in Settings -> Certificates tab

- Send requests using Username and Password authentication:
  - Authorization: Basic Auth
  - Certificates: No certificate added in Settings -> Certificates tab
- Send requests using Certificate authentication:
  - Authorization: No Auth / Inherit auth from parent
  - Certificates: Web services client certificate (.pfx) is added in Settings -> Certificates tab

Note: Don't forget to Clear All Cookies when sending multiple requests with different authentication types.

Note: The JSON Body has a size limit of 2MB.

Configure dataFEED OPC Suite Web Services

- Open dataFEED OPC Suite Configurator and go to Data Destination Web Services
- Enable Data Destination Web Services and add users with all authentication types: Anonymous, Username and Password, Certificate (use .pfx certificate)
- Add Data Source OPC Server connection (or any other data source with an address space)
- Start dataFEED OPC Suite Runtime
- Optionally restore WebServices_PythonTest.osf configuration from PythonWebServicesTests.zip file (found at dataFEED OPC Suite install location under **"samples\WebServices"** directory (ex. "**C:\Program Files (x86)\Softing\dataFEED OPC Suite\samples\WebServices**").

Python samples for *AnonymousPolicy*:

- In the following example the Version, Browse, Read, Write, Subscribe functionality is implemented with Anonymous user policy.

```
#!/usr/bin/python3.9
# -*- coding: utf-8 -*-
"""
Module for WEB Server Access
"""
# Python Imports
import json
import pprint
import sseclient
import sys
import requests
import urllib.parse
import urllib3
urllib3.disable_warnings()

class clsWebServices:
    def get(self, api_url, params=None, jsonData=None):
        """************************************************************
        *Function    :get
        *Parameter   :api_url = the Web Server URL
        *             params = A dictionary, list of tuples or bytes to send as a que
        *             jsonData = a valid JSON that will be sent in request body
        *RetVal      :
        *Description :Retrieve an existing resource.
        ************************************************************"""
        try:
            # In the following examples SSL certificate validation is avoided in the
            # For "requests" to verify certificates you should follow the correspondi
            # https://docs.python-requests.org/en/latest/user/advanced/#ssl-cert-veri
            response = requests.get(api_url, params=params, data=jsonData, verify=Fal
            return response
        except Exception as e:
            print(
                "Error in %s in function %s: %s" % (self.__class__.__name__, sys._get
            return -1


    def post(self, api_url, jsonData=None):
        """************************************************************
        *Function    :post
        *Parameter   :api_url = the Web Server URL
        *            jsonData = a valid JSON that will be sent in request body
        *RetVal      :
        *Description :Create a new resource.
        ************************************************************"""
        try:
            response = requests.post(api_url, jsonData, verify=False)
            return response
        except Exception as e:
            print("Error in %s in function %s: %s" % (self.__class__.__name__,sys._ge
            return -1

def main():

    # initialize WebServices object
    oWebClient = clsWebServices()
    webServerURL_version = "https://127.0.0.1:8881/Configuration1/api/version"
    webServerURL_browse = "https://127.0.0.1:8881/Configuration1/api/browse?syntax="
```


© Softing Industrial Automation GmbH

```
webServerURL_read = 'https://127.0.0.1:8881/Configuration1/api/read'
webServerURL_subscribe = "https://127.0.0.1:8881/Configuration1/api/subscribe"
webServerURL_write = "https://127.0.0.1:8881/Configuration1/api/write"

# Anonymous Authentication
# Get Version
print('>>> 1. Version Request')
getDFVersion = oWebClient.get(webServerURL_version)
print('Send GET request:', getDFVersion.url)
# Status code 200: The requested action was successful
print('Status code of get request:', getDFVersion.status_code)
# GET request Result
print('Result of get request:', getDFVersion.text)

# Browse
print('>>> 2. Browse Request')
# encode URLs that contain special chars
browseNode = 'OPC_1'
browseNodeEncoded = urllib.parse.quote_plus(browseNode)

getBrowseResult = oWebClient.get(webServerURL_browse + browseNodeEncoded)
print('Send GET request:', getBrowseResult.url)
# Status code 200: The requested action was successful
print('Status code of get request:', getBrowseResult.status_code)
# GET request Result
print('Result of get request:', getBrowseResult.text)

# Read Url
print('>>> 3. Read URL Request')
payload = 'items[0][syntax]=OPC_1.IOP.dynamic.I1'
getReadResult = oWebClient.get(webServerURL_read, params=payload)
print('Send GET request:', getReadResult.url)
# Status code 200: The requested action was successful
print('Status code of get request:', getReadResult.status_code)
# GET request Result
print('Result of get request:', getReadResult.text)

# Read JSON
print('>>> 4. Read JSON Body Request')
jsonData= {"items": [{"syntax": "OPC_1.IOP.dynamic.BOOL", "index": None}, {"synta
getReadJSONBodyResult = oWebClient.get(webServerURL_read, jsonData=json.dumps(jso
print('JSON Data: %s', str(jsonData))
print('Send GET request:', getReadJSONBodyResult.url)
# Status code 200: The requested action was successful
print('Status code of get request:', getReadJSONBodyResult.status_code)
# GET request Result
print('Result of get request:', getReadJSONBodyResult.text)

# Subscribe URL
# This is a Python client library for iterating over http Server Sent Event (SSE)
# The SSEClient class accepts a url on init, and is then an iterator over message
print('>>> 5. Subscribe URL Request')
params = "?items[0][syntax]=OPC_1.IOP.dynamic.I1&items[0][sampling_rate]=1000"
getSubscribeResult = requests.get(webServerURL_subscribe + params, stream=True, v
client = sseclient.SSEClient(getSubscribeResult)
count = 0
for event in client.events():
```

```
            pprint.pprint(json.loads(event.data))
            count = count + 1
            if count == 5:
                break

        # Subscribe JSON Body
        # This is a Python client library for iterating over http Server Sent Event (SSE)
        # The SSEClient class accepts a url on init, and is then an iterator over message
        print('>>> 6. Subscribe JSON Body Request')
        jsonData = {"items": [{"syntax": "OPC_1.IOP.dynamic.BOOL", "sampling_rate": 1000}
        getSubscribeJSONBodyResult = requests.get(webServerURL_subscribe, data=json.dumps
        client = sseclient.SSEClient(getSubscribeJSONBodyResult)
        count = 0
        for event_2 in client.events():
            pprint.pprint(json.loads(event_2.data))
            count = count + 1
            if count == 5:
                break

        # Write Url
        print('>>> 7. Write URL Request')
        item = "?items[0][syntax]=OPC_1.IOP.static.I1&items[0][value]=12"
        # Send POST write request to dataFEED data destination web server
        getWriteResult = oWebClient.post(webServerURL_write + item)
        print('Send GET request:', getWriteResult.url)
        # Status code 200: The requested action was successful
        print('Status code of get request:', getWriteResult.status_code)
        # GET request Result
        print('Result of get request:', getWriteResult.text)

        # Write JSON Body
        print('>>> 8. Write JSON Body Request')
        jsonData = {"items": [{"syntax": "OPC_1.IOP.static.I1", "index": None, "value": 1
        # Send POST write request to dataFEED data destination web server
        getWriteResult = oWebClient.post(webServerURL_write, jsonData=json.dumps(jsonData
        print('Send GET request:', getWriteResult.url)
        # Status code 200: The requested action was successful
        print('Status code of get request:', getWriteResult.status_code)
        # GET request Result
        print('Result of get request:', getWriteResult.text)

if __name__ == "__main__":
    sys.exit(main())
```

Python samples for *UserNamePolicy*:

- In the following example the Version, Browse, Read, Write, Subscribe functionality is implemented with User Name and Password policy.

© Softing Industrial Automation GmbH

```
#!/usr/bin/python3.9
# -*- coding: utf-8 -*-
"""
Module for WEB Server Access
"""
# Python Imports
import json
import pprint
import sseclient
import sys
import requests
import urllib.parse
import urllib3
urllib3.disable_warnings()

class clsWebServices:

    def get(self, api_url, params=None, authentication=None, jsonData=None):
        """************************************************************
        *Function    :get
        *Parameter   :api_url = the Web Server URL
        *            params = A dictionary, list of tuples or bytes to send as a quer
        *            jsonData = a valid JSON that will be sent in request body
        *            authentication = Username and password used for authentication
        *RetVal      :
        *Description :Retrieve an existing resource.
        ************************************************************"""
        try:
            # In the following examples SSL certificate validation is avoided in the
            # For "requests" to verify certificates you should follow the correspondi
            # https://docs.python-requests.org/en/latest/user/advanced/#ssl-cert-veri
            response = requests.get(api_url, params=params, data=jsonData, auth=authe
            return response
        except Exception as e:
            print(
                "Error in %s in function %s: %s" % (self.__class__.__name__, sys._get
            return -1


    def post(self, api_url, jsonData = None, authentication = None):
        """************************************************************
        *Function    :post
        *Parameter   :api_url = the Web Server URL
        *            jsonData = a valid JSON that will be sent in request body
        *            authentication = Username and password used for authentication
        *RetVal      :
        *Description :Create a new resource.
        ************************************************************"""
        try:
            # In the following examples SSL certificate validation is avoided in the
            # For "requests" to verify certificates you should follow the correspondi
            # https://docs.python-requests.org/en/latest/user/advanced/#ssl-cert-veri
            response = requests.post(api_url, jsonData, auth=authentication, verify=F
            return response
        except Exception as e:
            print("Error in %s in function %s: %s" % (self.__class__.__name__,sys._ge
            return -1
```

```
def main():

    # initialize WebServices object
    oWebClient = clsWebServices()
    webServerURL_version = "https://127.0.0.1:8881/Configuration1/api/version"
    webServerURL_browse = "https://127.0.0.1:8881/Configuration1/api/browse?syntax="
    webServerURL_read = 'https://127.0.0.1:8881/Configuration1/api/read'
    webServerURL_subscribe = "https://127.0.0.1:8881/Configuration1/api/subscribe"
    webServerURL_write = "https://127.0.0.1:8881/Configuration1/api/write"

    #Set username and password
    username = 'User1'
    password = 'qaqaqa'

    # Username and Password Authentication
    # Get Version
    print('>>> 1. Version Request')
    getDFVersion = oWebClient.get(webServerURL_version, authentication=(username, pas
    print('Send GET request:', getDFVersion.url)
    # Status code 200: The requested action was successful
    print('Status code of get request:', getDFVersion.status_code)
    # GET request Result
    print('Result of get request:', getDFVersion.text)

    # Browse
    print('>>> 2. Browse Request')
    # encode URLs that contain special chars
    browseNode = 'OPC_1'
    browseNodeEncoded = urllib.parse.quote_plus(browseNode)

    getBrowseResult = oWebClient.get(webServerURL_browse + browseNodeEncoded, authent
    print('Send GET request:', getBrowseResult.url)
    # Status code 200: The requested action was successful
    print('Status code of get request:', getBrowseResult.status_code)
    # GET request Result
    print('Result of get request:', getBrowseResult.text)

    # Read Url
    print('>>> 3. Read URL Request')
    payload = 'items[0][syntax]=OPC_1.IOP.dynamic.I1'
    getReadResult = oWebClient.get(webServerURL_read, params=payload, authentication=
    print('Send GET request:', getReadResult.url)
    # Status code 200: The requested action was successful
    print('Status code of get request:', getReadResult.status_code)
    # GET request Result
    print('Result of get request:', getReadResult.text)

    # Read JSON
    print('>>> 4. Read JSON Body Request')
    jsonData= {"items": [{"syntax": "OPC_1.IOP.dynamic.BOOL", "index": None}, {"synta
    getReadJSONBodyResult = oWebClient.get(webServerURL_read, jsonData=json.dumps(jso
    print('JSON Data: %s', str(jsonData))
    print('Send GET request:', getReadJSONBodyResult.url)
    # Status code 200: The requested action was successful
    print('Status code of get request:', getReadJSONBodyResult.status_code)
    # GET request Result
    print('Result of get request:', getReadJSONBodyResult.text)
```


© Softing Industrial Automation GmbH

```
        # Subscribe URL
        # This is a Python client library for iterating over http Server Sent Event (SSE)
        # The SSEClient class accepts a url on init, and is then an iterator over message
        print('>>> 5. Subscribe URL Request')
        params = "?items[0][syntax]=OPC_1.IOP.dynamic.I1&items[0][sampling_rate]=1000"
        getSubscribeResult = requests.get(webServerURL_subscribe + params, stream=True, v
        client = sseclient.SSEClient(getSubscribeResult)
        count = 0
        for event in client.events():
            pprint.pprint(json.loads(event.data))
            count = count + 1
            if count == 5:
                break

        # Subscribe JSON Body
        # This is a Python client library for iterating over http Server Sent Event (SSE)
        # The SSEClient class accepts a url on init, and is then an iterator over message
        print('>>> 6. Subscribe JSON Body Request')
        jsonData = {"items": [{"syntax": "OPC_1.IOP.dynamic.BOOL", "sampling_rate": 1000}
        getSubscribeJSONBodyResult = requests.get(webServerURL_subscribe, data=json.dumps
        client = sseclient.SSEClient(getSubscribeJSONBodyResult)
        count = 0
        for event_2 in client.events():
            pprint.pprint(json.loads(event_2.data))
            count = count + 1
            if count == 5:
                break

        # Write Url
        print('>>> 7. Write URL Request')
        item = "?items[0][syntax]=OPC_1.IOP.static.I1&items[0][value]=12"
        # Send POST write request to dataFEED data destination web server
        getWriteResult = oWebClient.post(webServerURL_write + item, authentication=(usern
        print('Send GET request:', getWriteResult.url)
        # Status code 200: The requested action was successful
        print('Status code of get request:', getWriteResult.status_code)
        # GET request Result
        print('Result of get request:', getWriteResult.text)

        # Write JSON Body
        print('>>> 8. Write JSON Body Request')
        jsonData = {"items": [{"syntax": "OPC_1.IOP.static.I1", "index": None, "value": 1
        # Send POST write request to dataFEED data destination web server
        getWriteResult = oWebClient.post(webServerURL_write, jsonData=json.dumps(jsonData
        print('Send GET request:', getWriteResult.url)
        # Status code 200: The requested action was successful
        print('Status code of get request:', getWriteResult.status_code)
        # GET request Result
        print('Result of get request:', getWriteResult.text)

if __name__ == "__main__":
    sys.exit(main())
```

Python samples for *CertificatePolicy*:

- In the following example the Version, Browse, Read, Write, Subscribe functionality is implemented with Certificate based policy.

© Softing Industrial Automation GmbH

```
#!/usr/bin/python3.9
# -*- coding: utf-8 -*-
"""
Module for WEB Server Access
"""
# Python Imports
import os
import json
import pprint
import sseclient
import sys
import requests
import urllib.parse
import urllib3
urllib3.disable_warnings()

class clsWebServices:

    def get(self, api_url, params=None, certificate=None, jsonData=None):
        """************************************************************
        *Function    :get
        *Parameter   :api_url = the Web Server URL
        *             params = A dictionary, list of tuples or bytes to send as a que
        *            jsonData = a valid JSON that will be sent in request body
        *            certificate = certificate used for authentication
        *RetVal      :
        *Description :Retrieve an existing resource.
        ************************************************************"""
        try:
            # In the following examples SSL certificate validation is avoided in the
            # For "requests" to verify certificates you should follow the correspondi
            # https://docs.python-requests.org/en/latest/user/advanced/#ssl-cert-veri
            response = requests.get(api_url, params=params, data=jsonData, cert=certi
            return response
        except Exception as e:
            print(
                "Error in %s in function %s: %s" % (self.__class__.__name__, sys._get
            return -1


    def post(self, api_url, jsonData = None, certificate=None):
        """************************************************************
        *Function    :post
        *Parameter   :api_url = the Web Server URL
        *            jsonData = a valid JSON that will be sent in request body
        *            certificate = certificate used for authentication
        *RetVal      :
        *Description :Create a new resource.
        ************************************************************"""
        try:
            response = requests.post(api_url, jsonData, cert=certificate, verify=Fals
            return response
        except Exception as e:
            print("Error in %s in function %s: %s" % (self.__class__.__name__,sys._ge
            return -1

def main():
```

```
# initialize WebServices object
oWebClient = clsWebServices()
webServerURL_version = "https://127.0.0.1:8881/Configuration1/api/version"
webServerURL_browse = "https://127.0.0.1:8881/Configuration1/api/browse?syntax="
webServerURL_read = 'https://127.0.0.1:8881/Configuration1/api/read'
webServerURL_subscribe = "https://127.0.0.1:8881/Configuration1/api/subscribe"
webServerURL_write = "https://127.0.0.1:8881/Configuration1/api/write"

# Set certificate
cert_file_path_1 = os.path.join('C:\Projects\Test-Cases\Test-Cases\Testtrack-Proj
key_file_path_1 = os.path.join('C:\Projects\Test-Cases\Test-Cases\Testtrack-Proje
certPath = (cert_file_path_1,key_file_path_1)

# Username and Password Authentication
# Get Version
print('>>> 1. Version Request')
getDFVersion = oWebClient.get(webServerURL_version, certificate=certPath)
print('Send GET request:', getDFVersion.url)
# Status code 200: The requested action was successful
print('Status code of get request:', getDFVersion.status_code)
# GET request Result
print('Result of get request:', getDFVersion.text)

# Browse
print('>>> 2. Browse Request')
# encode URLs that contain special chars
browseNode = 'OPC_1'
browseNodeEncoded = urllib.parse.quote_plus(browseNode)

getBrowseResult = oWebClient.get(webServerURL_browse + browseNodeEncoded, certifi
print('Send GET request:', getBrowseResult.url)
# Status code 200: The requested action was successful
print('Status code of get request:', getBrowseResult.status_code)
# GET request Result
print('Result of get request:', getBrowseResult.text)

# Read Url
print('>>> 3. Read URL Request')
payload = 'items[0][syntax]=OPC_1.IOP.dynamic.I1'
getReadResult = oWebClient.get(webServerURL_read, params=payload, certificate=cer
print('Send GET request:', getReadResult.url)
# Status code 200: The requested action was successful
print('Status code of get request:', getReadResult.status_code)
# GET request Result
print('Result of get request:', getReadResult.text)

# Read JSON
print('>>> 4. Read JSON Body Request')
jsonData= {"items": [{"syntax": "OPC_1.IOP.dynamic.BOOL", "index": None}, {"synta
getReadJSONBodyResult = oWebClient.get(webServerURL_read, jsonData=json.dumps(jso
print('JSON Data: %s', str(jsonData))
print('Send GET request:', getReadJSONBodyResult.url)
# Status code 200: The requested action was successful
print('Status code of get request:', getReadJSONBodyResult.status_code)
# GET request Result
print('Result of get request:', getReadJSONBodyResult.text)
```


© Softing Industrial Automation GmbH

```
        # Subscribe URL
        # This is a Python client library for iterating over http Server Sent Event (SSE)
        # The SSEClient class accepts a url on init, and is then an iterator over message
        print('>>> 5. Subscribe URL Request')
        params = "?items[0][syntax]=OPC_1.IOP.dynamic.I1&items[0][sampling_rate]=1000"
        getSubscribeResult = requests.get(webServerURL_subscribe + params, stream=True, v
        client = sseclient.SSEClient(getSubscribeResult)
        count = 0
        for event in client.events():
            pprint.pprint(json.loads(event.data))
            count = count + 1
            if count == 5:
                break

        # Subscribe JSON Body
        # This is a Python client library for iterating over http Server Sent Event (SSE)
        # The SSEClient class accepts a url on init, and is then an iterator over message
        print('>>> 6. Subscribe JSON Body Request')
        jsonData = {"items": [{"syntax": "OPC_1.IOP.dynamic.BOOL", "sampling_rate": 1000}
        getSubscribeJSONBodyResult = requests.get(webServerURL_subscribe, data=json.dumps
        client = sseclient.SSEClient(getSubscribeJSONBodyResult)
        count = 0
        for event_2 in client.events():
            pprint.pprint(json.loads(event_2.data))
            count = count + 1
            if count == 5:
                break

        # Write Url
        print('>>> 7. Write URL Request')
        item = "?items[0][syntax]=OPC_1.IOP.static.I1&items[0][value]=12"
        # Send POST write request to dataFEED data destination web server
        getWriteResult = oWebClient.post(webServerURL_write + item, certificate=certPath)
        print('Send GET request:', getWriteResult.url)
        # Status code 200: The requested action was successful
        print('Status code of get request:', getWriteResult.status_code)
        # GET request Result
        print('Result of get request:', getWriteResult.text)

        # Write JSON Body
        print('>>> 8. Write JSON Body Request')
        jsonData = {"items": [{"syntax": "OPC_1.IOP.static.I1", "index": None, "value": 1
        # Send POST write request to dataFEED data destination web server
        getWriteResult = oWebClient.post(webServerURL_write, jsonData=json.dumps(jsonData
        print('Send GET request:', getWriteResult.url)
        # Status code 200: The requested action was successful
        print('Status code of get request:', getWriteResult.status_code)
        # GET request Result
        print('Result of get request:', getWriteResult.text)

    if __name__ == "__main__":
        sys.exit(main())
```

Steps to connect with Curl Web Client to dataFEED OPC Suite **Web server:**

Preconditions:

- Install curl
- Install dataFEED OPC Suite V5.35 or newer version
- Generate .pfx Client certificate
- Optionally restore DD_WebServices_Read.osf configuration from CurlWebServicesTests.zip file (found at dataFEED OPC Suite install location under **"samples\WebServices"** directory (ex. "**C:\Program Files (x86)\Softing\dataFEED OPC Suite\samples\WebServices**").

Install and configure curl application

Download application from path: https://curl.se/windows/ and place it in proper directory.

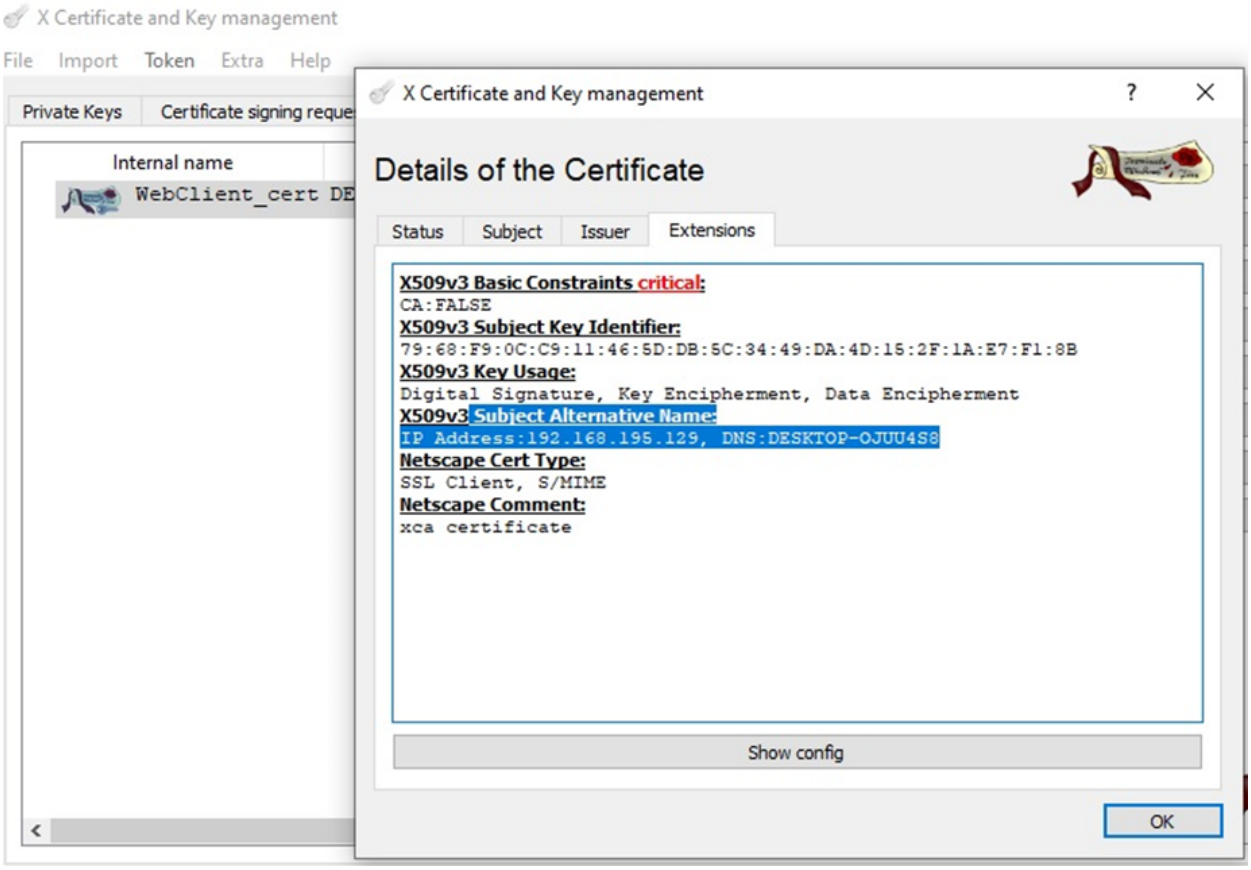
Generate .pfx Client certificate

Before sending the requests using Certificate user, the client certificates should be generated.

Client certificate should contain Subject Alternative Name with host IP and DNS – Host PC Name:

IP Address:127.0.0.1, DNS:pc_name

© Softing Industrial Automation GmbH



Create .pfx client certificate using the generated client certificate and client key. The following openssl command can be used:

```
openssl pkcs12 -export -out WebClient_cert.pfx -inkey WebClient_cert.pem -in WebClien
```

Copy the .pfx client certificate in the same folder where .bat files are stored.

Configure dataFEED OPC Suite Web Services:

1. Open dataFEED OPC Suite Configurator and go to Data Destination Web Services
2. Enable Data Destination Web Services, change port to 8881 and add users with all authentication types: Anonymous, Username and Password, Certificate (use .pfx certificate)
3. Add Data Source OPC Server connection (or any other data source with an address space)
4. Start dataFEED OPC Suite Runtime

Run the following scripts .bat, using CMD:

Run Command Prompt as administrator and open the folder where .bat files are stored. Run the following .bat files using curl:

- curl_test_GetVersion.bat - Version
- curl_test_Browse.bat - Browse
- curl_test_Read_URL.bat - Read
- curl_test_Read_Body.bat - Read
- curl_test_Write_URL.bat - Write
- curl_test_Write_Body.bat - Write
- curl_test_Subscribe_URL.bat - Subscribe
- curl_test_Subscribe_Body.bat - Subscribe

Note: Samples can be found at dataFEED OPC Suite install location under "samples\WebServices\CurlWebServicesTests.zip" directory (ex. "c:\Program Files (x86) \Softing\dataFEED OPC Suite\samples\WebServices\CurlWebServicesTests.zip").

Note: The curl.exe path, username and password and certificate name and password should be updated in each .bat file.

Steps to connect with Spring Framework to dataFEED OPC Suite **Web server:**

Preconditions:

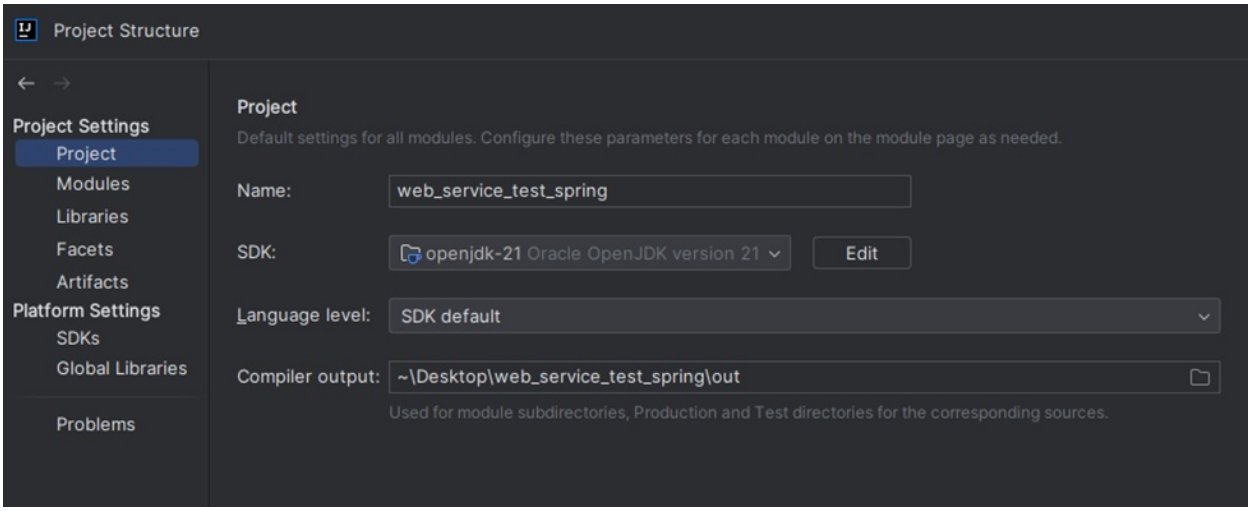
- In this example: Install IntelliJ IDEA
- Install dataFEED OPC Suite V5.35 or newer version
- Generate .pfx Client certificate
- Optionally restore DD_WebServices_Read.osf configuration from SpringWebServicesTests.zip file (found at dataFEED OPC Suite install location under **"samples\WebServices"** directory (ex. "**C:\Program Files (x86)\Softing\dataFEED OPC Suite\samples\WebServices**").

Install and configure IntelliJ IDEA application

Download application from path: https://www.jetbrains.com/idea/download/?fromIDE=&section=windows

Open IntelliJ IDEA integrated development environment (IDE) and also the Spring Project. Go to Settings -> Project Structure and configure the project like is showing in the following picture:

© Softing Industrial Automation GmbH

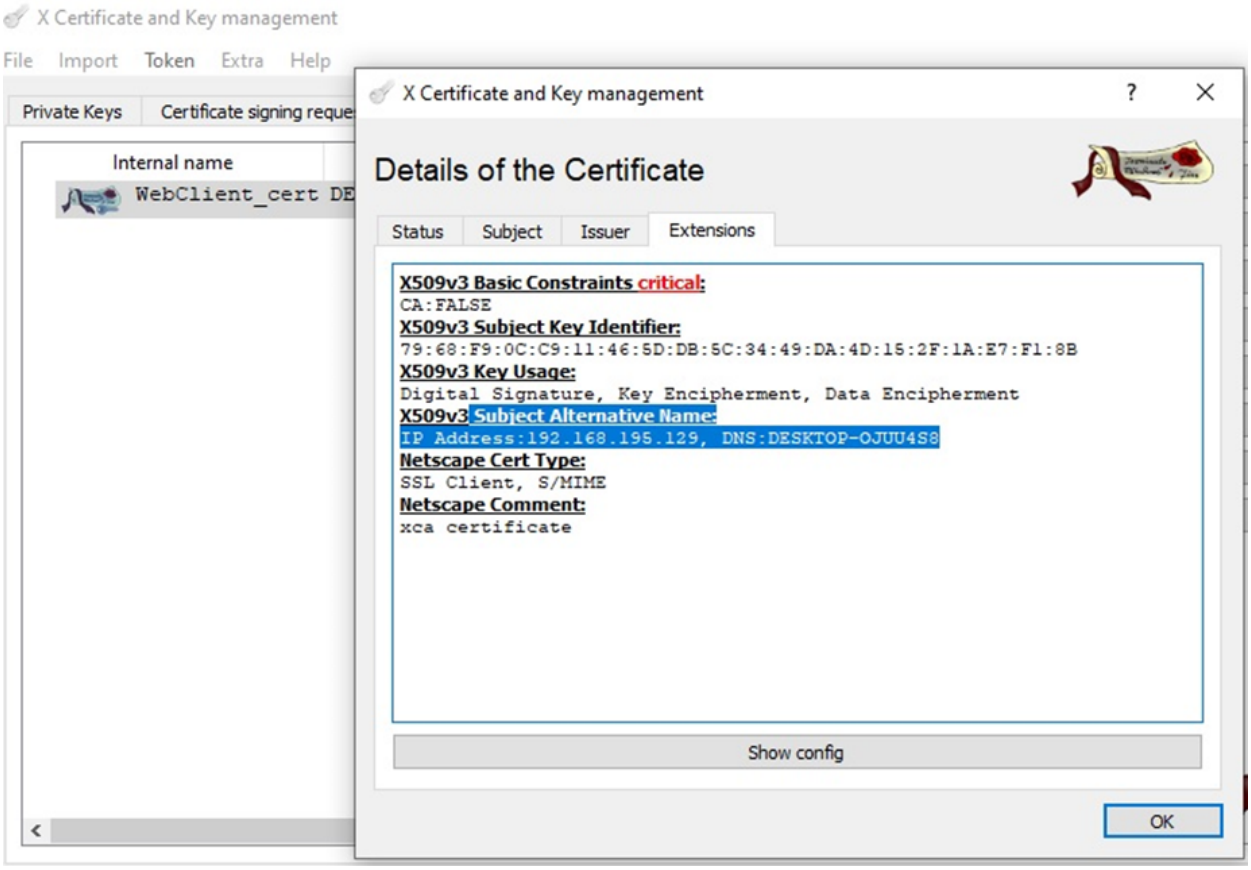


Fix all the Problems from Settings -> Project Structure -> Problems

Generate .pfx Client certificate

Before sending the requests using Certificate user, the client certificates should be generated. Client certificate should contain Subject Alternative Name with host IP and DNS – Host PC Name:

IP Address:127.0.0.1, DNS:pc_name



Create .pfx client certificate using the generated client certificate and client key. The following openssl command can be used:

```
openssl pkcs12 -export -out WebClient_cert.pfx -inkey WebClient_cert.pem -in WebClien
```

Create a folder _certs on Desktop: “c:/Users/sfadm/Desktop/_certs” and copy there the .pfx client certificate.

Configure dataFEED OPC Suite Web Services:

1. Open dataFEED OPC Suite Configurator and go to Data Destination Web Services
2. Enable Data Destination Web Services, change port to 8883 and add users with all authentication types: Anonymous, Username and Password, Certificate (use .pfx certificate)
3. Add Data Source OPC Server connection (or any other data source with an address space)
4. Start dataFEED OPC Suite Runtime

© Softing Industrial Automation GmbH

Run the following scripts from src -> main -> java -> com.softing:

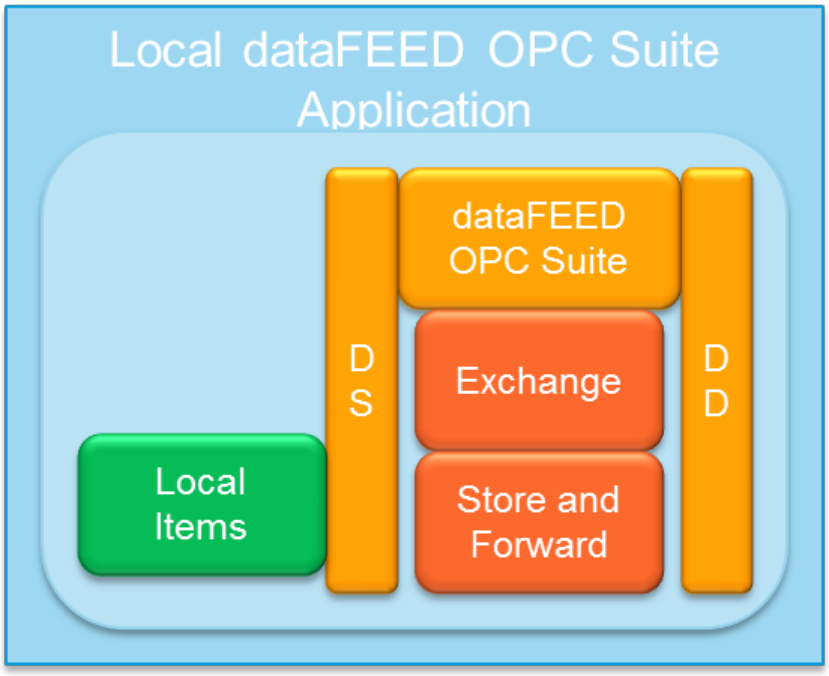
- WebClientTestAnonymous.java
- WebClientTestUser.java
- WebClientTestCertificate.java
- All the samples contains examples for retrieving the Version, Browse the address-space, Read, Write and Subscribe items.

Note: Samples can be found at dataFEED OPC Suite install location under "samples\WebServices\SpringWebServicesTests.zip" directory (ex. "c:\Program Files (x86) \Softing\dataFEED OPC Suite\samples\WebServices\SpringWebServicesTests.zip").

Note: The certificate name and password should be updated in WebClientTestCertificate.java script.

7.3.4 Settings

This section of the Softing dataFEED OPC Suite Online Help describes some global settings for the local dataFEED OPC Suite application.



▾ Softing dataFEED OPC Suite - Settings

- Configure diagnostics trace outputs to file @ Configuration: Settings - Diagnostics
- Enable the integrated web server @ Configuration: Settings - Web Server
- Advanced configuration settings @ Configuration: Settings - Advanced
- Discover the currently installed licenses and the license items used @ Configuration: Settings - Licenses

Softing dataFEED OPC Suite V5.38
© 2024 Softing Industrial Automation GmbH

7.3.4.1 Diagnostic

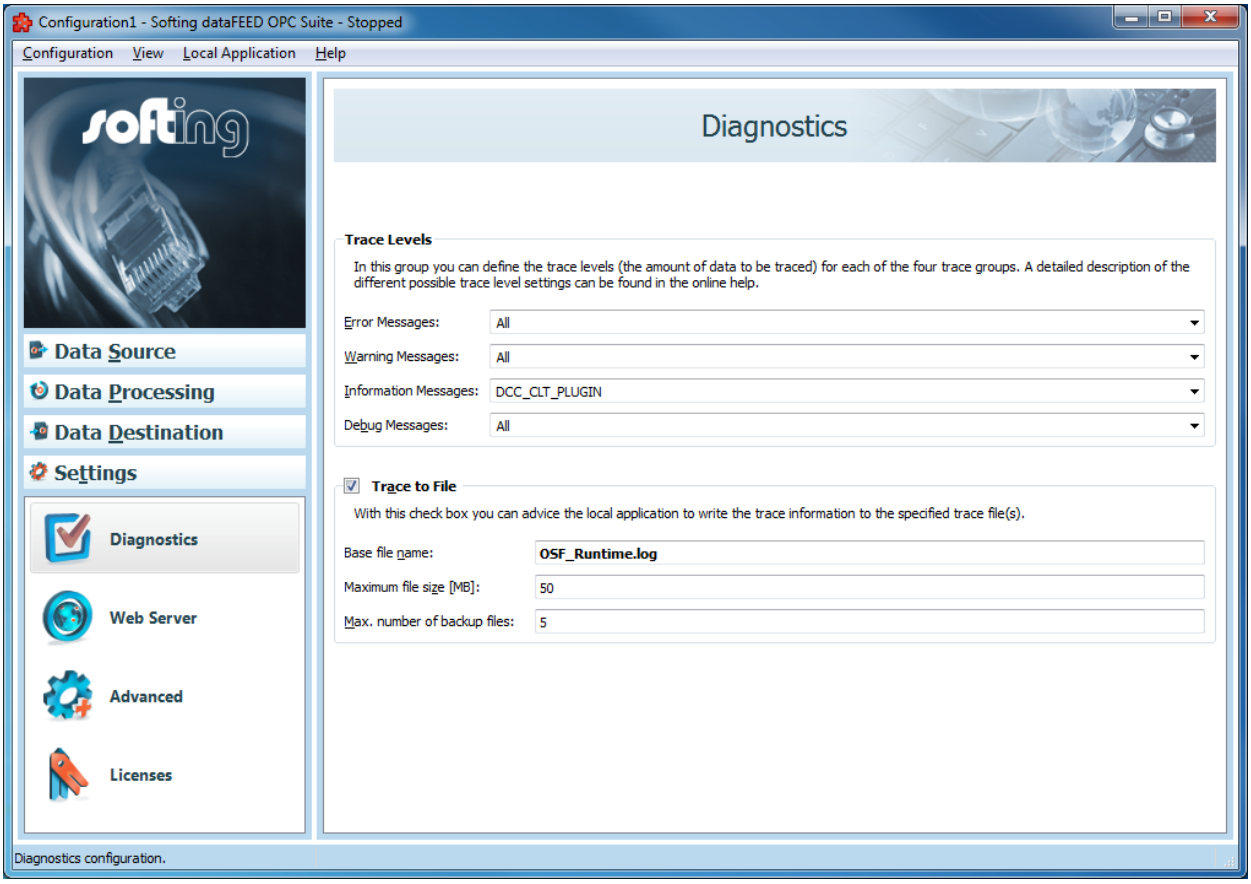
This section allows defining the trace output to be written for each of the four trace groups. The section allows:

- choosing what traces to include in the logs
- enabling or disabling file tracing
- choosing a name for the trace file
- defining the maximum file size
- setting the maximum number of trace file backups

For a more detailed description see the Configuration: Settings - Diagnostics page.

7.3.4.1.1 Configuration

This section describes how to configure the diagnostics output of the local dataFEED OPC Suite application.



Trace Levels

In the *Trace Levels* group box the different trace levels (the amount of data to be traced) can be defined for each of the four trace groups. The following trace groups are available:

Trace Group	Description
Error Messages	Trace group containing severe error messages which could occur during the run time of the local dataFEED OPC Suite application.
Warning Messages	Trace group with additional warning messages (non-critical error messages).
Information Messages	Trace group containing additional information messages.
Debug Messages	Trace group with additional debug messages (changing the value of this group from *Nothing* will cause the trace files size to increase dramatically over a short period).

The modules for each of the above trace groups can be selected from the drop down list. The drop-down list can also be manually edited providing a way for typing a hexadecimal representation of different combinations:

- *All:* All trace messages will be logged (OPC, Server, Client, OCT).
- *Nothing:* No trace messages will be logged for the selected group.
- *OPC:* Trace messages from the OPC side will be logged.
- *Server:* Trace messages from the server side will be logged.
- *Client:* Trace messages from the client side will be logged.
- *OCT:* Trace messages from Softing dataFEED OPC Suite will be logged.

Trace To File

In the bottom group box, you can enable the logging of trace data and specify the trace file you want to use.

To enable tracing to a file the *Trace to File* checkbox must be checked.

If the checkbox is not checked, the section requesting input of the name and size of the trace files will be disabled (grayed out).

Attribute	Description
Base file name	Name of the trace file.
Maximum file size [MB]	Maximum size (in Megabytes) of each of the trace files.
Max. number of backup files	The number of desired backup files.

Note: When the local dataFEED OPC Suite application is restarted the contents of the trace files are backed up. The backup is only performed for the previous log files when a new session of the local dataFEED OPC Suite application is started and not when the log files reach their maximum size during the application runtime.

If asked for the trace files in a support case, you can find them in the following folder:

```
%ALLUSERSPROFILE%\Softing\dataFEED OPC Suite\<ConfigurationName>\
```

Please note that this folder can vary depending on the Microsoft Windows operating system and the language version used.

The backed up trace files have the following names:

```
OSF_Runtime_backup#<number>A.log
OSF_Runtime_backup#<number>B.log
```

The actual trace files have the following names:

```
OSF_Runtime_currentA.log
OSF_Runtime_currentB.log
```


© Softing Industrial Automation GmbH

Note: The tracing mechanism is circular meaning that it writes data to two separate files. When the first file reaches the *Maximum file size* the trace mechanism overwrites the second file and so on.

Note: When the local dataFEED OPC Suite application starts it automatically renames the two trace files from the previous session as the most recent backup files. If *Number of backup files* is set to 0 then the trace files from the last session will not be backed up.

Attention: Please make sure that you have enough free space for the ongoing trace files (e.g. twice the *Maximum File Size*) plus additional space for the backup files (if *Number of backup* files is greater than 0).

7.3.4.2 Web Server

Softing dataFEED OPC Suite features an integrated web server, which also allows you to provide your own web pages for displaying and setting values. The architecture and performance of the web server have been designed for small-scale visualizations.

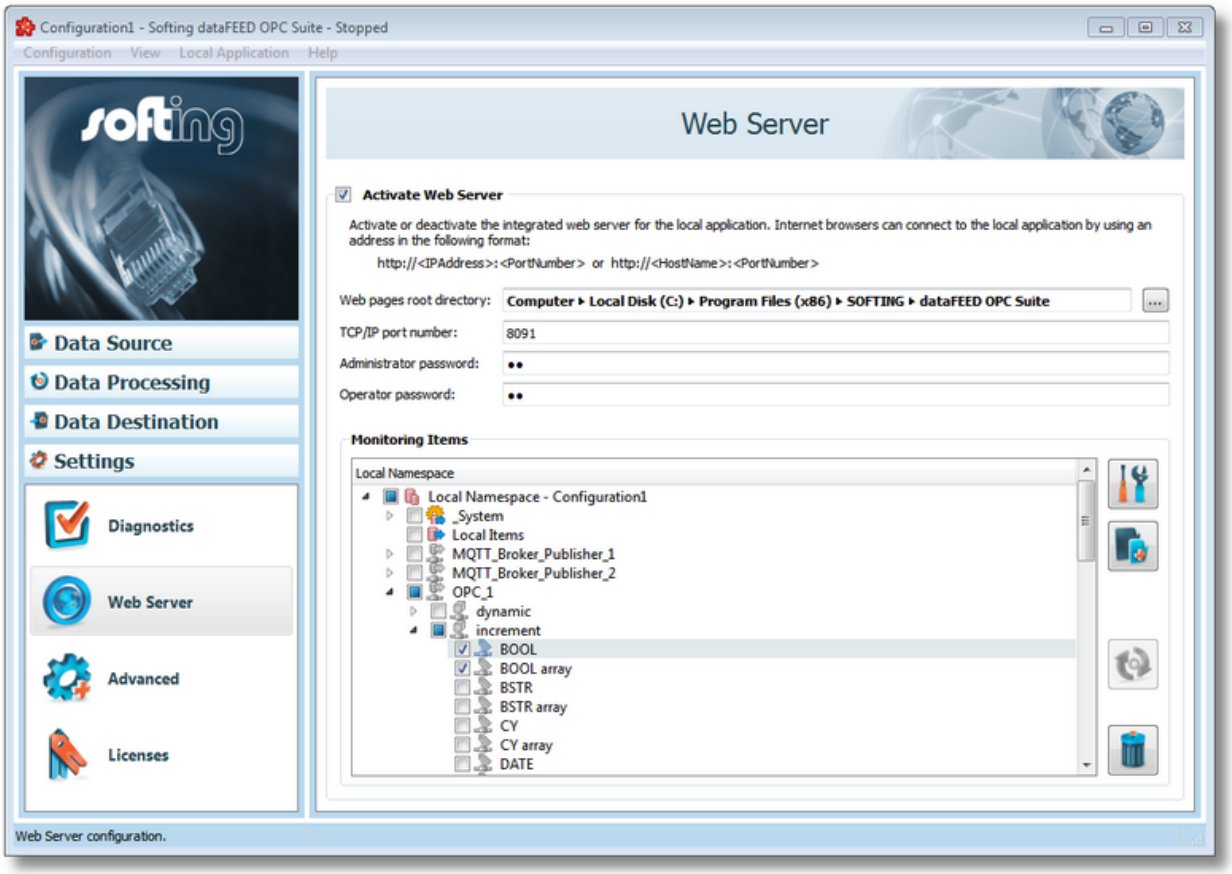
The web pages of the local dataFEED OPC Suite application provide run time information about the current licenses and license items, version information, diagnostics information and contact data for requesting technical support for Softing dataFEED OPC Suite. They are described in detail under dataFEED OPC Suite Web Pages.

More detailed information on the dataFEED OPC Suite web server configuration can be found on the Configuration: Settings - Web Server page.

Attention: When the 72-hours demonstration mode expires, the local dataFEED OPC Suite application is generally stopped. As a result, the corresponding web pages will not be available anymore, either.

7.3.4.2.1 Configuration

In this configuration section the integrated web server for the local dataFEED OPC Suite application can be activated and configured.



The TCP/IP port number used for the data destination OPC XML-DA functionality and the integrated web server is always the same.

Note: Each dataFEED OPC Suite configuration running on the same workstation must use a unique TCP/IP port number. This port number must not be used by other applications running on the same PC.

If a new configuration is created, a unique port number is automatically assigned to the new configuration. If the TCP/IP port number is manually changed, additional care must be taken that the corresponding port number is not already used on the target machine.

The attributes for the integrated web server are described in the table below:

Attribute	Description
Web pages root directory	The directory where the web pages are located on the local host. The default value is the installation path. The path must be updated if the web pages are moved from the installation path.
TCP/IP port number	The web server and OPC XML-DA server TCP/IP port number.
Administrator password	The password for the administrator login.

© Softing Industrial Automation GmbH

Operator password	The password for the operator login.

The administrator and the operator users have the following default passwords:

User	Password
Operator	op
Administrator	ad

The passwords can be changed by editing the *Administrator password* and/or *Operator password* edit boxes or through the web interface.

Note: If the administrator or operator password is changed through the web interface and the corresponding configuration is currently loaded in Softing dataFEED OPC Suite, the password edit boxes will not be updated until the current configuration is reloaded.

In the **Monitoring Items** tree the user can select readable items that are added in the web server pages under Diagnostics / Process Values once the runtime configuration is restarted.

Both nodes and tag items can be selected but in the web server only the tags are shown.

7.3.4.3 Advanced

Advanced global settings which affect the entire local dataFEED OPC Suite application can be configured in this section.

There are two settings groups, which are displayed in the following group boxes:

- Global Application Settings
- Global DS-OPC and DS-Tunnel Settings

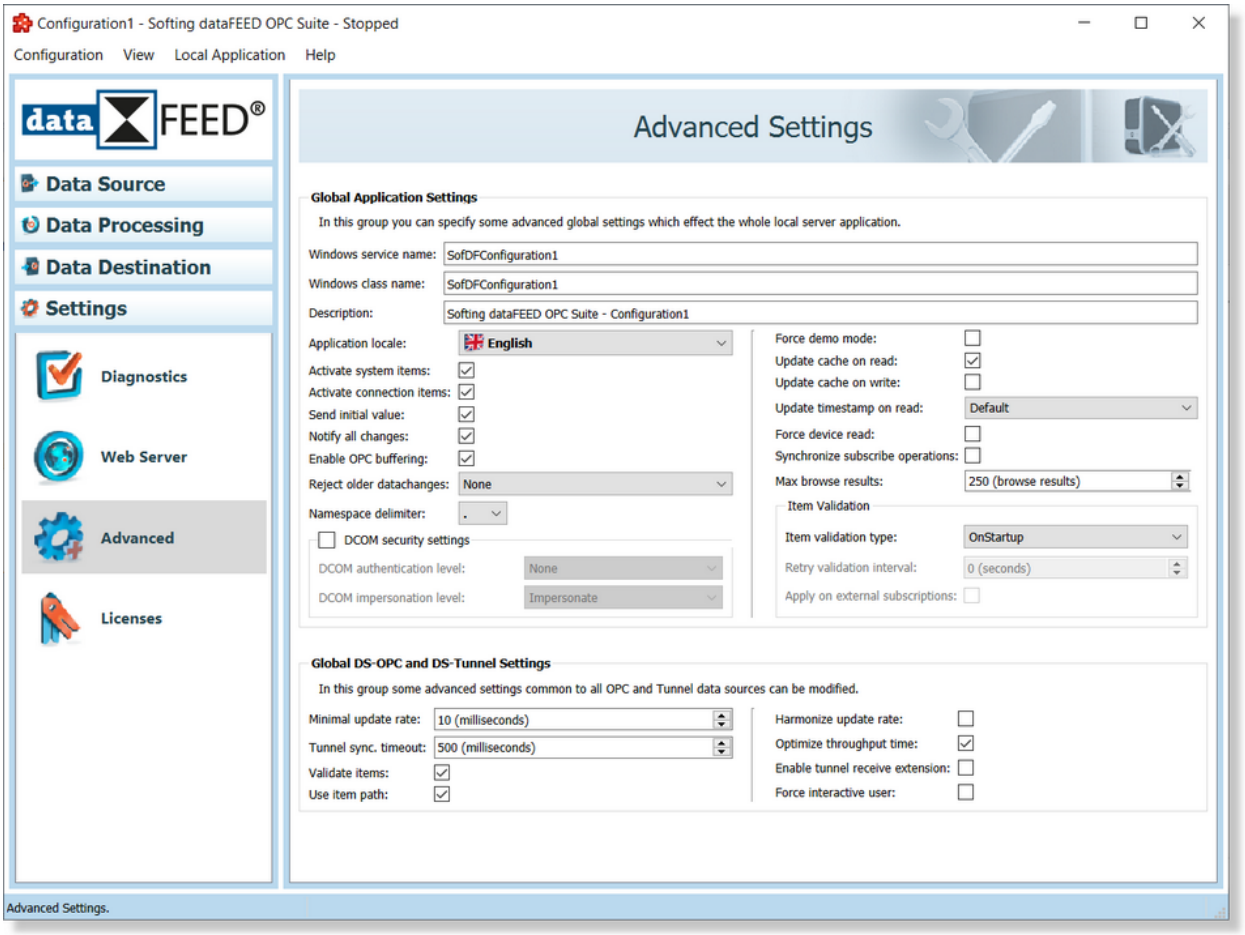
The following settings can be configured in the *Global Application Settings* section: the service name, the Windows class name, the description, the application locale, whether the local dataFEED OPC Suite application should expose the system and the connection items or not, and whether to force the demo mode if a license is installed or not. Also, it is allowed to set the advanced communication and processing settings.

The following settings can be configured in the *Global DS-OPC and DS-Tunnel Settings* section: the minimum client update rate, the tunnel synchronization timeout value, whether to validate items, to harmonize update rate and to optimize the throughput time or not.

A more detailed description can be found on the Configuration: Settings - Advanced page.

7.3.4.3.1 Configuration

This configuration section allows configuring advanced global settings for the local dataFEED OPC Suite application.



Global Application Settings

The meanings of the *Windows service name*, *Windows class name*, *Description*, *Application Locale*, *Activate system items*, *Activate connection items* and *Force demo mode* configuration fields are explained in the table below:

Attribute	Description
Windows service name	Service name used to register the local dataFEED OPC Suite application with the Windows Service Manager. This name must be unique for all local dataFEED OPC Suite application configurations running on the computer.
Windows class name	Name of the Windows class used for the local dataFEED OPC Suite application. This name must be unique for all local dataFEED OPC Suite application configurations running on the computer.
Description	Service description used to register the local dataFEED OPC Suite application with the Windows Service Manager.
Application Locale	This option influences the formatting of the output data that the local dataFEED OPC Suite application writes to the file or database. This setting

© Softing Industrial Automation GmbH

	usually affects the number format (decimal separators, etc), the date/time format, the currency format, etc.
Activate system items	If this attribute is checked, system items will be provided for the server connections. A new node with specific items will be created for each server connection. The following items are provided: • _System.General.HeartBeat (32-bit Unsigned Integer, read-only) - This item increments on each read that a client performs on it (reads on other items do not trigger the increment). The initial value is 0. • _System.General.SessionID (32-bit Unsigned Integer, read-only) - Contains a unique session identifier (random number updated on each server start, kept for as long as the server is running). • _System.General.LocalTime (DateTime, read-only) - Provides the operating system local current date time. If *Store And Forward* is also activated (see the Data Processing - Store And Forward chapter for details) you will see three more items under the *_System* node (see the Store And Forward Wizard - Local Storage page of the Store And Forward Wizard to identify the below mentioned (bolded) controls): • _System.Data_Processing.StoreAndForward.ExceededMaxSize (Boolean, read-only) - Returns true when the Max. size specified for the local storage file has been exceeded (but the Max. data changes limit has not been reached). The warning is generated but the Store And Forward plug-in continues to write data values to the file while the Max. data changes limit is not reached. • _System.Data_Processing.StoreAndForward.FileLoadPercentage (64-bit Real, read-only) - Provides the load percentage of the Store And Forward buffer (related to Max. data changes; the Max. size of the local file buffer is not relevant as the _System.Data_Processing.StoreAndForward.ExceededMaxSize warning is generated when it is exceeded). • _System.Data_Processing.StoreAndForward.OverWritingStarted (Boolean, read-only) - This item returns whether overwriting of the stored item values began or not. This item is always false unless the Overwrite values checkbox is checked. • _System.Data_Processing.StoreAndForward.StoredTransactions (64-bit Unsigned Integer, read-only) - This item returns the number of stored (and not forwarded yet) transaction. Each time when a transaction is stored it will be increased and when a transaction is forwarded to the connected OPC Client and the confirmation is received it will be decreased.
Activate connection items	When checked a new node named Data_Source will appear under the *_System* node. The node will only show up if at least one external server connection exists in the *External Tunnel Servers* or *External OPC Servers* or at least one PLC connection is defined. The *Data_Source* node has a child node for each data source connection provided in any of the above

	lists. For each defined data-source connection a set of special items are predefined in the internal address space. Based on data-source type a specific group of system items are created. All connections are grouped under a node based on its data-source types.. **Siemens, Rockwell, Modbus or Mitsubishi data-source type:** • _System.Data_Source.<DataSource>.<ConnectionName>.Connected (Boolean, read-only) - The item specifies the connection establishment between dataFEED OPC Suite and corresponding configured PLC. • _System.Data_Source.<DataSource>.<ConnectionName>.Status (32-bit unsigned integer, read-only) - The item returns the connection status between dataFEED OPC Suite and corresponding configured PLC. • _System.Data_Source.<DataSource>.<ConnectionName>.ServerCycle (32-bit unsigned integer, read-only) - The special item ServerCycle shows the number of complete passes of the server for a connection. • _System.Data_Source.<DataSource>.<ConnectionName>.MaxInterval (32-bit integer, read-only) - The special item MaxInterval is used to access the measured Maximum Update Interval (in milliseconds) of all items of a corresponding connection. This value refers to the last complete polling cycle. • _System.Data_Source.<DataSource>.<ConnectionName>.ItemCount (32-bit unsigned integer, read-only) - The special ItemCount item is used to access the number of active items of the corresponding connection. This value is also indicated in the status window under the heading “Items.” • _System.Data_Source.<DataSource>.<ConnectionName>.ErrorCount (32-bit unsigned integer, read-only) - The special ErrorCount item is used to access the number of all active, erroneous items of the applicable Access Path. If the connection is in a “bad” state, all items have errors (i.e., the ItemCount item corresponds to the ErrorCount item). • _System.Data_Source.<DataSource>.<ConnectionName>.IpAddress (String, read-only) - The special item IpAddress is used to access the target IP address of the applicable Access Path. • _System.Data_Source.<DataSource>.<ConnectionName>.PollingInterval (32-bit unsigned integer, read-only) - The special item Poll Interval is used to access the current poll interval. The current poll interval is indicated in milliseconds. A negative value indicates that polling was disabled. In this case, the last entered positive value remains on the display. • _System.Data_Source.<DataSource>.<ConnectionName>.PLCStatus (32-bit unsigned integer, read-only) - The special Item PLCStatus is used to indicate the status of an S7 CPU. With S7 connections and HPP switched on, it can be registered (applicable only for Siemens connections). • _System.Data_Source.<DataSource>.<ConnectionName>.BrowsingReady (Boolean, read-only) - The special Item BrowsingReady is used to indicate the when the browse operations are available of dataFEED OPC

© Softing Industrial Automation GmbH

	Suite. Basically this is done after the initialization of all Rockwell defined connections (applicable only for Rockwell connections). **B&R data-source type:** • _System.Data_Source.<DataSource>.<ConnectionName>.Connected (Boolean, read-only) - The item specifies the connection establishment between dataFEED OPC Suite and corresponding configured B&R device. • _System.Data_Source.<DataSource>.<ConnectionName>.DeviceIPAddress (String, read-only) - The item specifies the IP address of the target station (IP address of the PLC) or the DNS name of the target station is entered. • _System.Data_Source.<DataSource>.<ConnectionName>.PVIIPAddress (String, read-only) - The item specifies the IP address of configured PVI Manager. • _System.Data_Source.<DataSource>.<ConnectionName>.ItemCount (32-bit unsigned integer, read-only) -The special ItemCount item is used to access the number of active items of the corresponding connection. This value is also indicated in the status window under the heading “Items.” • _System.Data_Source.<DataSource>.<ConnectionName>.ErrorCount (32-bit unsigned integer, read-only) - The special ErrorCount item is used to access the number of all active, erroneous items of the applicable Access Path. If the connection is in a “bad” state, all items have errors (i.e., the ItemCount item corresponds to the ErrorCount item). **OPCServer or Tunnel Server data-source type:** • _System.Data_Source.<DataSource>.<ConnectionName>.Connected (Boolean, read-only) - The Boolean value of this system item shows if there is currently a connection to the corresponding server or not. • _System.Data_Source.<DataSource>.<ConnectionName>.ProvideItems (Boolean, read-only) - This item returns the Provide Items attribute of the corresponding server connection as a string. • _System.Data_Source.<DataSource>.<ConnectionName>.URL (String, read-only) - This item returns the associated connection URL of the corresponding server connection. • _System.Data_Source.<DataSource>.<ConnectionName>..ItemPrefix (String, read-only) - This item returns the configured item prefix (i.e. connection name) of the corresponding server connection. This item is only provided if an item prefix / connection name has been configured. **Local Items data-source type** • _System.Data_Source.<DataSource>.<ConnectionName>.LocalItemsWatchdog (32-bit unsigned integer, read-only) - A client that acts logically as the OPC server has to cyclically write to the LocalItemsWatchdog.
Send initial value	Each subscribed item on classical interfaces (DA2.0, DA3.0, Tunnel and XML-DA) is internally initialized with quality BadWaitingForInitialValue

	creating its internal cache. Inside the update-rate interval the data source should provide the initial value of the monitored item - updating the internal initialized cache with the value, quality and timestamp from the connected data source. In case the the connection establishment is needed more time than the update-rate interval the internal cache initialization is sent to the connected client. With this setting disabled, the notification of the connected client with the internal initialization is interrupted. **Attention:** Disabling this option will make the product not OPC Compliant - in case that the initial read is needed more time than the update-rate interval then the initial data-change is not sent to the connected client.
Notify all changes	Basically for a monitored/subscribed node accordingly with the OPC specifications there should be send at least one data-change inside the update-rate interval in case of value/quality changed. This settings is allowing the possibility to send multiple notification in inside the update rate for a monitored item - each notification is sent in a dedicated call in case when OPC Buffering is disabled. The default value of these attribute is: "true". **Attention:** Disabling this setting will have impact on the Store and Forward functionality - not all the stored values will be sent to the Forward Client in case when *Packed Data Changes* is set as Forwarding Method.
Enable OPC buffering	The setting is enabling the creation of a buffer containing a queue of data-changes for the same subscribed item. This is is very important in context of using store and forward mechanism where in the same transaction as sent multiple data-changes for the same item at the forward time. Default value for this attribute is: "true". **Attention:** Disabling this setting will will have impact on the Store and Forward functionality - not all the stored values will be sent to the Forward Client in case when *Original Transaction* is set as Forwarding Method.
Reject Older Datachanges	This setting enables/disables the cache update operation when an older datachange is received than current cached value. When the option is enabled on each received datachange is done a comparison between current timestamp (based on internal cache) and the received timestamp - in case that the current datachange is older than the cache then the datachange will be ignored.

© Softing Industrial Automation GmbH

	Attention: When the cache contains only the internal initialization (no real values being received from the corresponding data-source on current item) - the validation of timestamp is not activated. The following options are allowed: • *None* - no timestamp validation being performed. • *Older Than Timestamp* - older values received are ignored only if those values are not older than the subscription rate (sampling interval). • *All Older Values* - all older values are ignored. By default this setting is set to None (older values are never ignored).
Namespace Delimiter	This setting allows the possibility to specify the namespace delimiter for the address space exposed by the dataFEED OPC Suite application. Basically the item ID for all the items from the exposed address space are created being composed from multiple tokens delimited by this namespace delimiter. Default value of the namespace delimiter is "." This a very advanced setting which should be changed only by advanced users. **Attention:** This setting is applied for all import/export operations. Importing an address space from a file which was exported from a configuration using a different namespace delimiter then the actual used is generating a address space different then the expected one - always in this situation the imported address space will be in flat format, the entire hierarchy being not preserved.
DCOM security settings	Enables the possibility to customize the DCOM security settings (authentication level and impersonation level). Disabling it (default value) is using the default values on Classical SDK used by the application. The "None" authentication level and "Impersonate" impersonation level are used on ::CoInitializeSecurity application initialization function over the RPC. (The ::CoCreateInstanceEx function over the RPC is using no security while establishing connections over DCOM). **Attention:** Enabling the security settings, the configured values are use exclusively by the application. Still, the communication are influenced by the global (machine specific) DCOM Settings. By default it is disabled.

DCOM authentication level	Registers the authentication level on DCOM security values for the process. These values specify an authentication level, which indicates the amount of authentication provided to help protect the integrity of the data. Each level includes the protection provided by the previous levels. The following options are allowed: • Default (**RPC_C_AUTHN_LEVEL_DEFAULT - 0**): Tells DCOM to choose the authentication level using its normal security blanket negotiation algorithm. • None (**RPC_C_AUTHN_LEVEL_NONE - 1**): Performs no authentication. • Connect (**RPC_C_AUTHN_LEVEL_CONNECT - 2**): Authenticates the credentials of the client only when the client establishes a relationship with the server. Datagram transports always use RPC_AUTHN_LEVEL_PKT instead. • Call (**RPC_C_AUTHN_LEVEL_CALL - 3**): Authenticates only at the beginning of each remote procedure call when the server receives the request. Datagram transports use RPC_C_AUTHN_LEVEL_PKT instead. • PKT (**RPC_C_AUTHN_LEVEL_PKT - 4**): Authenticates that all data received is from the expected client. • PKT Integrity (**RPC_C_AUTHN_LEVEL_PKT_INTEGRITY - 5**): Authenticates and verifies that none of the data transferred between client and server has been modified. • PKT Privacy (**RPC_C_AUTHN_LEVEL_PKT_PRIVACY - 6**): Authenticates all previous levels and encrypts the argument value of each remote procedure call. A complete description of available parameters based on operating system can be found: https://docs.microsoft.com/en-us/windows/win32/com/com-authentication-level-constants
DCOM impersonation level	Specifies an impersonation level, which indicates the amount of authority given to the server when it is impersonating the client. The following options are allowed: • Default (**RPC_C_IMP_LEVEL_DEFAULT - 0**): DCOM can choose the impersonation level using its normal security blanket negotiation algorithm. • Anonymous (**RPC_C_IMP_LEVEL_ANONYMOUS - 1**): The client is anonymous to the server. The server process can impersonate the client, but the impersonation token will not contain any information and cannot be used. • Identify (**RPC_C_IMP_LEVEL_IDENTIFY - 2**): The server can obtain the client's identity. The server can impersonate the client for ACL checking, but it cannot access system objects as the client.

© Softing Industrial Automation GmbH

	• Impersonate (**RPC_C_IMP_LEVEL_IMPERSONATE - 3**): The server process can impersonate the client's security context while acting on behalf of the client. This level of impersonation can be used to access local resources such as files. When impersonating at this level, the impersonation token can only be passed across one machine boundary. The Schannel authentication service only supports this level of impersonation. • Delegate (**RPC_C_IMP_LEVEL_DELEGATE - 4**): The server process can impersonate the client's security context while acting on behalf of the client. The server process can also make outgoing calls to other servers while acting on behalf of the client, using cloaking. The server may use the client's security context on other machines to access local and remote resources as the client. When impersonating at this level, the impersonation token can be passed across any number of computer boundaries. A complete description of available parameters based on operating system can be found: https://docs.microsoft.com/en-us/windows/win32/com/com-impersonation-level-constants
Force demo mode	With this setting, you can force Softing dataFEED OPC Suite to run in demo mode. This is very useful if you have licensed individual features on a computer and want to test other tools later. When a feature has been licensed, the demo functionality is disabled for the other features.
Update cache on read	This setting allows the possibility to update the internal cache of a read node after a successful read operation executed. **Note:** The result of the read operation is not affected at all by this setting - the setting being applied only for the internal cache which later can deserve other operations - Cache Read or Datachange value preparation. **Attention:** Updating the cache value is updating for the current node the following properties: Value, Status Code and Timestamp(s). This settings is changing the behavior of the application for the next executed operations if those required access to the cache value. By default this setting is enabled.
Update cache on write	This setting allows the possibility to update the internal cache of a written node after a successful write operation executed.

	Note: The result of the write operation is not affected at all by this setting - the setting being applied only for the internal cache which later can deserve other operations - Cache Read or Datachange value preparation. **Attention:** Updating the cache value is updating for the current node the following properties: Value, Status Code and Timestamp(s). This settings is changing the behavior of the application for the next executed operations if those required access to the cache value. By default this setting is disabled.
Update timestamp on read	This setting enables the update of timestamp on cache value after a successful read operation executed. The following options are allowed: • *Default* - the timestamp is updated only when this is required by the connected client (timestamp property being required by the read operation). • *On Device Read Request* - the timestamp on cache is updated only in case of a Device Read operation successfully executed. • *On Every Read Requests* - the timestamp on cache is updated on every read operations successfully executed. **Attention:** this settings is not applicable for UA Data Destination.
Force device read	This setting is enabling the forcing of a Device Read on every Read requests. Basically, a Cache Read operation required by a connected client is converted (if the setting is enabled) to a Device Read request to the data source. **Note:** The cache is also updated accordingly the the specified settings even if this settings is enable and the Cache Read request is replaced by the Device Read request. **Note:** Recommendation is to not enable this settings for standard OPC Clients. By default this settings is disabled.
Synchronize subscribe	This setting enables the synchronization between on going subscribe and unsubscribe operation even if asynchronous subscribe/unsubscribe

© Softing Industrial Automation GmbH

operations	operations are required. Basically, when this settings is enabled it is not allowed to perform parallel subscribe/unsubscribe operations - the operations being executed sequentially in the same order than those were received from the OPC Client - each on progress operation waiting for the finalization of the previous one. **Attention:** this setting is introducing delays when parallel subscribe/unsubscribe operations are required being executed. The recommendation is to not be used except very specific situations. By default this settings is disabled.
Max browse results	Specifies the maximum number of results required on a browse operation. **Note:** In case when the data source node contains more children than this value, the additional children will be obtain based on Browse Next operations and handling of correspondent continuation points.
Item validation type	This setting enables the possibility to operate a continuous validation of items used on internal subscriptions (subscription operated by the configured functionality on the configuration tool - items used as data destinations or data processing items). By default (used as exclusive option on previous versions) at the initialization time, the items used on data destinations and data processing functionality were validated exclusively at the initialization time - in case of failure all the actions referenced by these items were invalidated and deactivated. The new functionality allows the user to activate a continuous validation of items while running. The following options being available: • *OnStartup* - the validation is done only once at the initialization time - in case of failure the actions referenced by these items being disabled. • *OnDemand* - the validation is done continuously until the corresponding connection is enabled - basically the application can be started without the source connection being activated yet, immediately after the connection is established, the validation and subscribe operation is executed... in case of failure the corresponding action being disabled. • *Continuously* - the validation is tried continuously while the application is running for the entire life time (until a successful result is received).

	Attention: enabling a continuous validation items is affecting the performance of the application in case while the items are not valid at the initialization phase - this can introduce overhead and additional communication with the data source. By default the validation of items option is set to OnStartup.
Retry validation interval	This settings is specifying the interval for continuous validation mechanism (if the continuous validation mechanism is enabled). When the continuous validation mechanism is enabled, the validation of items is executed continuously (until succeeded) with a periodicity of the specified interval. The default value of this setting is 30 seconds.
Apply on external subscriptions:	The "*Item Validation type*" setting is apply by default exclusively on internal subscriptions (used by internal OPC Client deserving the data destinations and processing functionality). Enabling the "*Apply on external subscriptions*" is activating the configured option for "*Item Validation type*" on subscriptions created by external OPC Clients connected to the exposed OPC Endpoint. **Note:** The option is available only for "OnDemand" and "Continuously" item validation types. By default this option is deactivated.

Global DS-OPC and DS-Tunnel Settings

The meanings of *Minimal update rate*, *Tunnel sync. timeout*, *Validate items*, *Harmonize update rate* and *Optimize throughput time* are explained in the table below:

Attribute	Description
Minimal update rate	This attribute can be used to set the minimum refresh rate granted to an OPC client. This reduces the communication load generated by an OPC client for an OPC server.
Tunnel sync. timeout	This property is specific to the Softing OPC Tunnel protocol. It defines the maximum time which the client/server will wait for a request acknowledgement from the other party before

© Softing Industrial Automation GmbH

	declaring the respective tunnel connection as disconnected. This value should be adjusted to suit the communication medium.
Validate items	Decides whether the item IDs from a dynamic server connection subscription should be validated or not. If no validation is performed all item IDs from the subscription file will be accepted by the local dataFEED OPC Suite application for a dynamic source. Leaving out validation increases the item adding speed. See the Tunnel Server Wizard - Filter page and the OPC Server Wizard - Filter page on how to configure the server connection: dynamic or static.
Use Item Path	This feature allows a special usage of the Item Path attribute when accessing items via the classical OPC DA interfaces: if this option is enabled, an external client can use the Item Path when accessing an item to specify the connection name the item is linked to. In this case the Item ID will provide only the remaining part of the fully qualified Item ID. Providing the fully qualified Item ID without the usage of the Access Path is supported in parallel. The feature can be used by the following interfaces: OPC-DA, Tunnel-DA and XML-DA. It can be used with any item in the address space, independent of the data source the item is retrieved through: Controller and Devices connections, OPC Server connections, Tunnel Server connections, Local Items or system items.
Harmonize update rate	Decides whether the groups with similar refresh rates should be combined in one group for the server connection. If this attribute is checked, two groups with a refresh rate of e.g. 520 ms and 490 ms will be combined in one with 500 ms, for instance.
Optimize throughput time	This feature allows data throughput through the local dataFEED OPC Suite application to be as fast as possible. If groups containing items from multiple server connections are defined in an OPC client, multiple data changes can be transmitted in one update cycle.
Enable tunnel receive extension	This feature enables the local dataFEED OPC Suite application to receive faster the tunnel communication data, by increasing the size of the local tunnel receive buffer. **Note:** Activating this feature will increase the data communication volume. Its usage is recommended only for this case.

	Note: To use this feature, please enable the option at both the Tunnel Server and Tunnel Client dataFEED OPC Suite configurations.
Force interactive user	This feature allows the classical DA/AE dataFEED OPC Suite interface to be registered under the Windows "Interactive User". This is applicable only while running in Application Mode and OPC data destination is activated.
	Note: By default, without activating this feature the dataFEED OPC Suite is running under the "SoftingOPCUser."
	Note: If the dataFEED OPC Suite is running in the System Mode, this setting has no effect. The service is always running under user "System".

7.3.4.4 Licenses

Softing OPC products and their scalable features are licensed and managed following a uniform concept. The scaling is done via product features, e.g. number of devices, number of connections or number of data items.

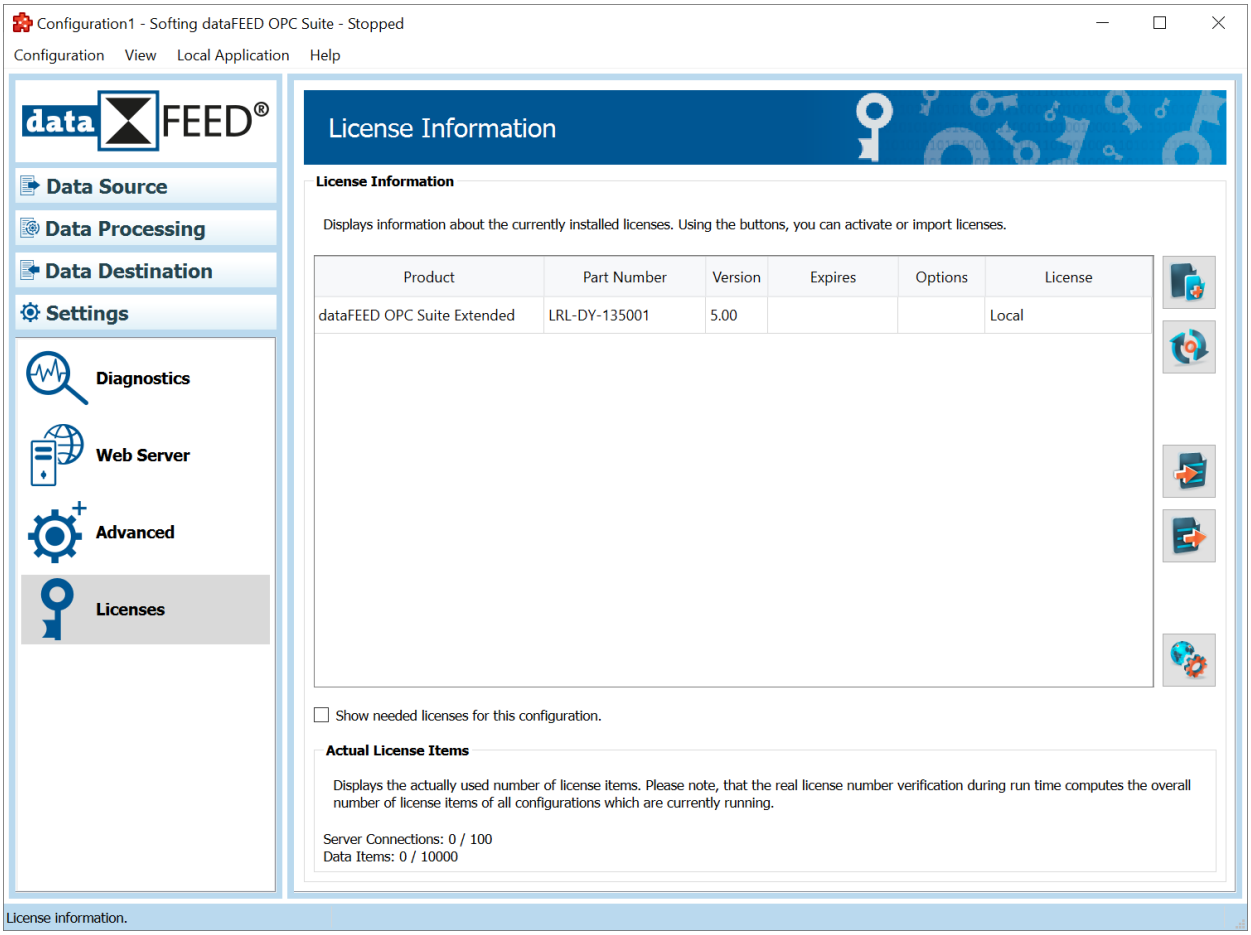
More details about the different licenses and the corresponding license activation can be found in the Licensing of Softing dataFEED OPC Suite chapter.

A more detailed description of the currently installed licenses can be found on the Configuration: Settings - Licenses page.

7.3.4.4.1 Configuration

The installed licenses and the license items currently used are exposed through the *Settings - License* add-on. Licenses can be installed and activated on this page, using the buttons on the right.

© Softing Industrial Automation GmbH



The *License Information* group box provides information about the corresponding product license, the part number, version, expiration date, options and license type.

If *Show needed licenses for this configuration* checkbox is checked then the *Actual License Items* group box will also display the information about needed licenses, besides information about the license items currently used.

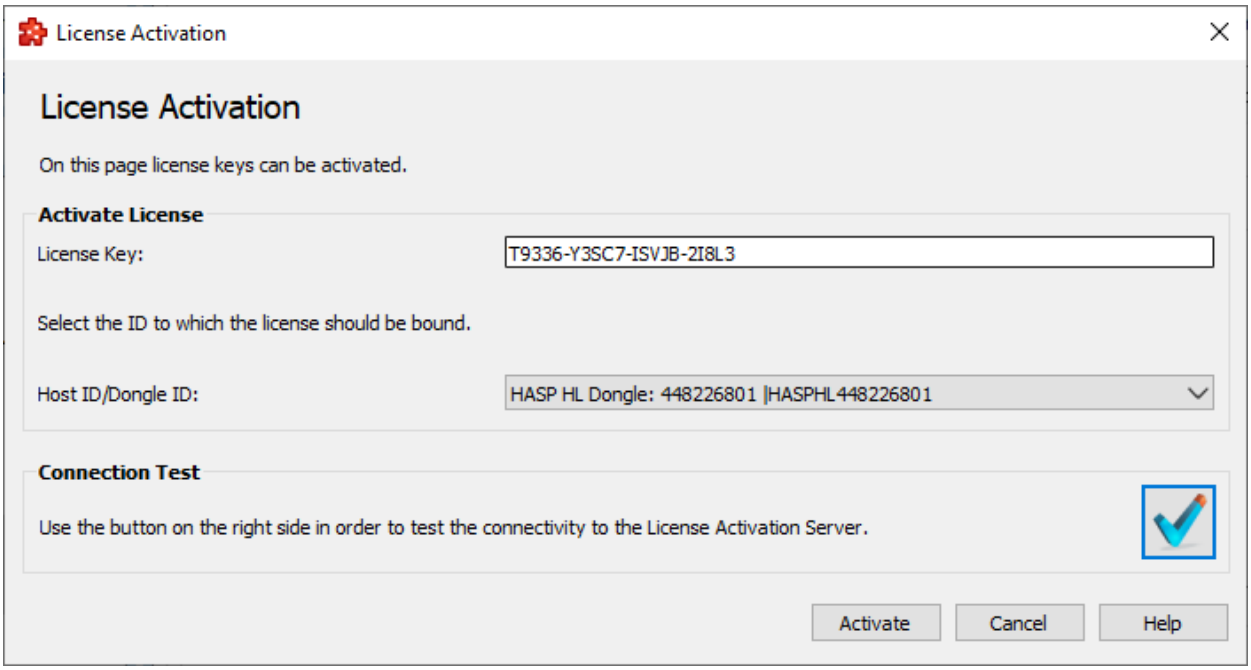
Attention: The computation of the actual license items here in the dataFEED OPC Suite Configurator only considers the license items of the corresponding configuration. The real number of license items is verified by the local dataFEED OPC Suite application during run time by computing the overall number of license items of all currently running local dataFEED OPC Suite applications.

Attention: The version of the licensed feature should be compatible with the version of the product, otherwise the feature is not activated by license. For example, the dataFEED OPC Suite version 5.xx is activated only by version 5.00 license keys (4.00 license keys not activating features on dataFEED OPC Suite version 5.00).

License Activation

To activate a license press the  button on the main view. The License Activation dialog will appear, where you can enter a valid license key. Then, press the **Activate** button at the bottom.

Note: The license keys are validated on this dialog. They must be in a specific format, no other format will be accepted. The generic format is XXXXX-XXXXX-XXXXX-XXXXX, containing alpha-numeric characters only, lower- and/or uppercase. An error message will be displayed when the license key is empty or has a different format and activation will not proceed.



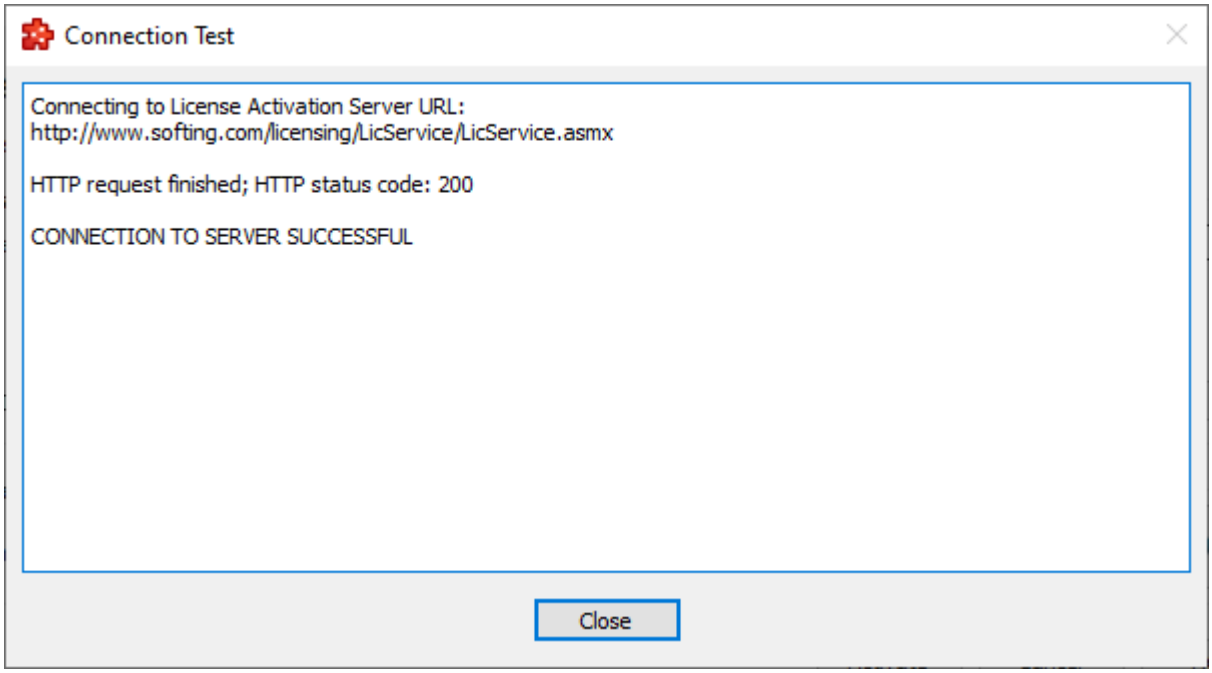
If successful, the activation process will store the received license file and the main view will show the new license in the list as soon as the dialog closes. In case of an unsuccessful activation an error message will be shown.

Different Host IDs can be selected from the drop-down. BIOS ID is the default selection, and if HASP driverless dongles are present in the machine USB interfaces, the IDs of those dongles will be shown as first entries in the list.

License Activation Server Connection Test

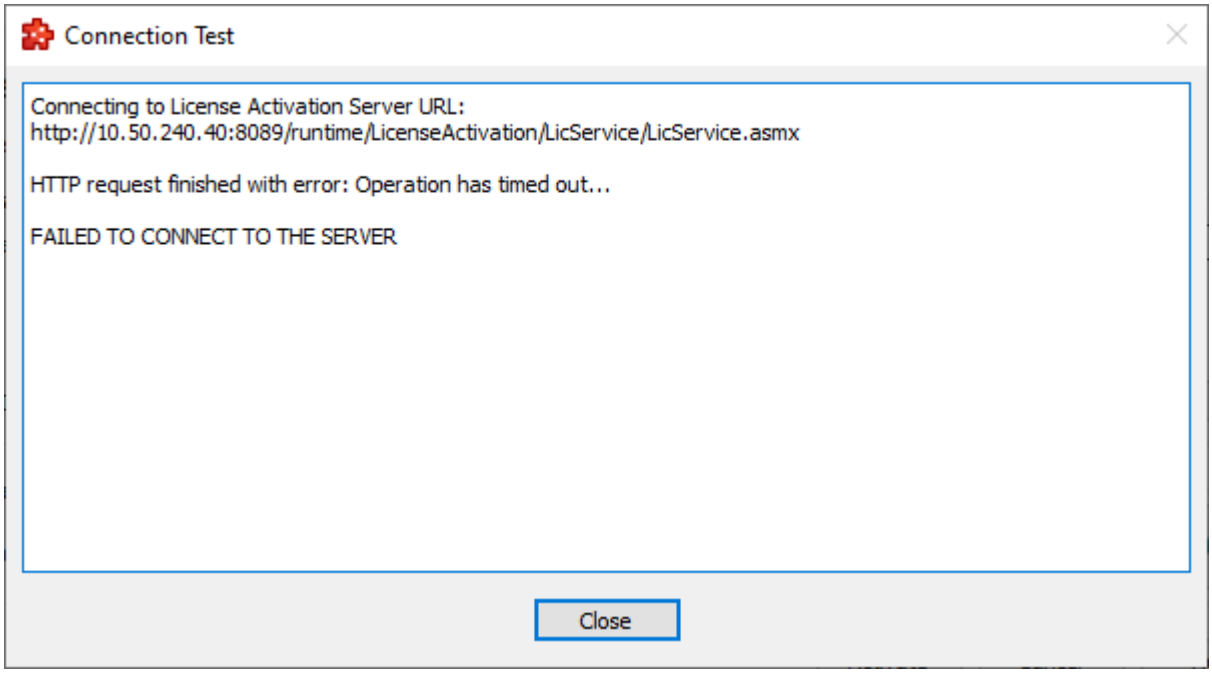
When the  button is pressed a connection test is performed to the Softing License Activation Server. The server is always online, when an internet connection exits. If not, a request time-out of 15 seconds will be applied when performing the connection test. This means the operation will be halted automatically if the server does not respond in that amount of time.

© Softing Industrial Automation GmbH



If the request fails when the server is available, the resulting HTTP status code is displayed, with a failed message.

A timed-out request look like this:

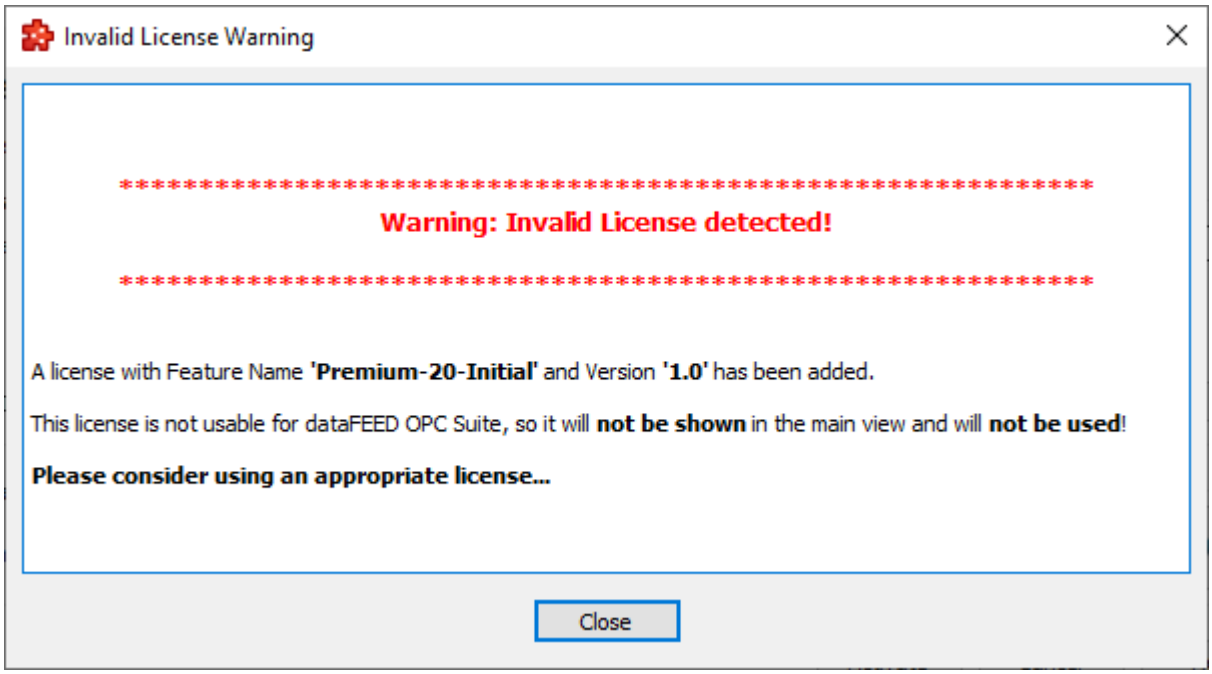


Invalid License Warning

Note: If a license key is activated or imported which represents a license of another Softing product, that license will not be used or shown in the licenses list. The activation process will be successful but the license of another product will be considered invalid. A dialog will be shown if

such a license is detected, notifying the user that an appropriate license must be activated.

Licenses corresponding to the application version must be used, e.g. 5.0, 5.3.



Licensed Status Update

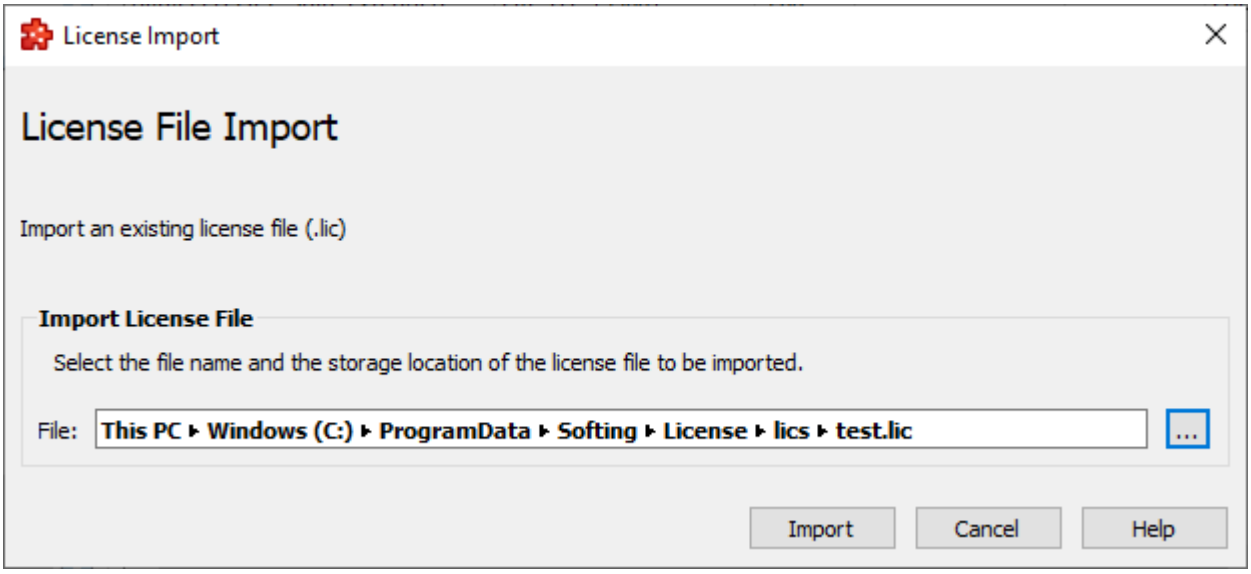
The **Refresh**  button updates the licensed status of the application after a license activation has completed, a dongle has been added/removed, a license has been lost, etc.

License File Import

The **Import License** button allows users to import an existing license file, which will be copied and stored in the default licenses directory. The extension of the license file must be **.lic**, otherwise they cannot be selected using the file open dialog.

When the button is pressed the following dialog will appear:

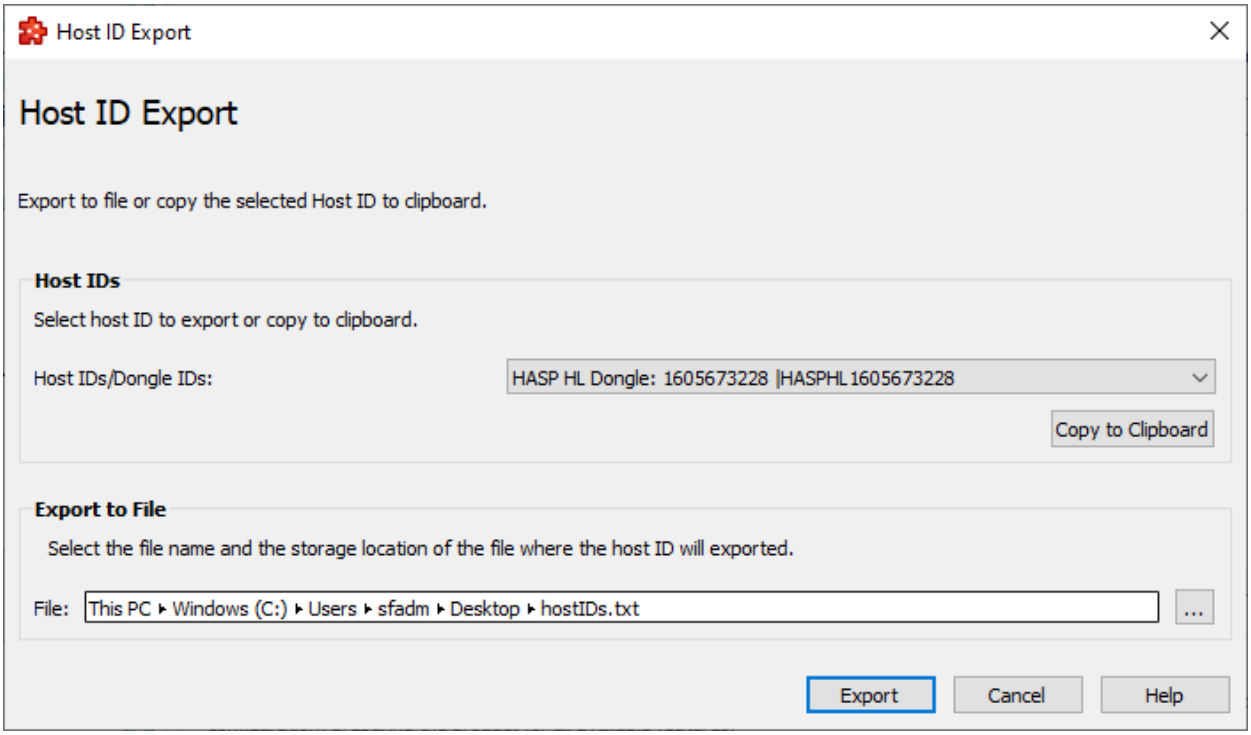
© Softing Industrial Automation GmbH



Press the **Import** button to start the import process. An appropriate message will appear depending on the success of the upload process. If successful, the dialog will close and the main view licenses list will be refreshed. Canceling the dialog will just close the dialog, without any effect.

Host ID Export

The **Export Host IDs**  button allows users to export a selected Host ID to a file. The dialog that is shown also allows to copy a Host ID to the clipboard.



Select a Host ID and press the **Copy to Clipboard** button to copy the host ID. After that the value can be pasted somewhere else.

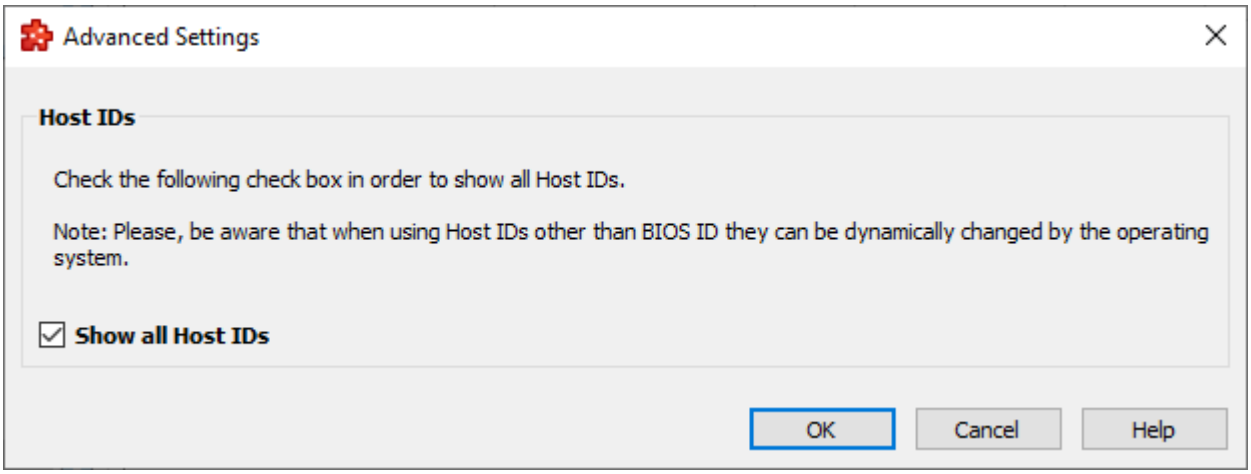
When a file path is selected or the proposed file path is used the **Export** button will export the selected Host ID to the specified file. Only one Host ID is exported, no multiple selection is possible.

A file can be specified for export by manually typing its path and name in the *File* input box or by clicking the browse button located to the right of the input box (a standard file saving dialog will open). The file will be created if its path was typed in and it doesn't exist. If an existing file is provided its contents will be overwritten. The type of the file will be a .txt file for easier opening after export.

Advanced Settings

When pushing the  button, the settings dialog will appear. It contains an option which changes which types of host IDs are retrieved from the system for license activation.

By default it is turned off, the "Show all Host IDs" check box is not checked. This means that only BIOS ID and HASP dongle IDs will be shown in the list on the License Activation and the Host ID Export dialog. When checking the check box, all host IDs will be shown: Dongle IDs (if any is plugged into the computer), BIOS ID and HardDisk IDs as well.



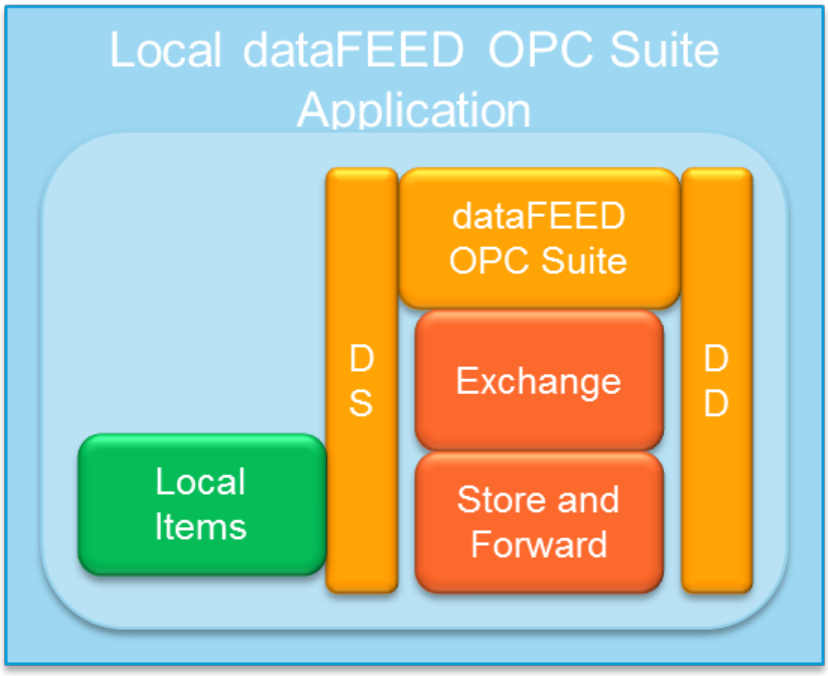
When pressing OK, the state of the option will be saved and loaded next time. It will instantly affect how host IDs are retrieved from the system.

For more details regarding the different licenses see the License Model of Softing dataFEED OPC Suite chapter.

7.3.5 General

This section of the Softing dataFEED OPC Suite Online Help describes some global general settings for the local dataFEED OPC Suite application.

© Softing Industrial Automation GmbH



▼ Softing dataFEED OPC Suite - General

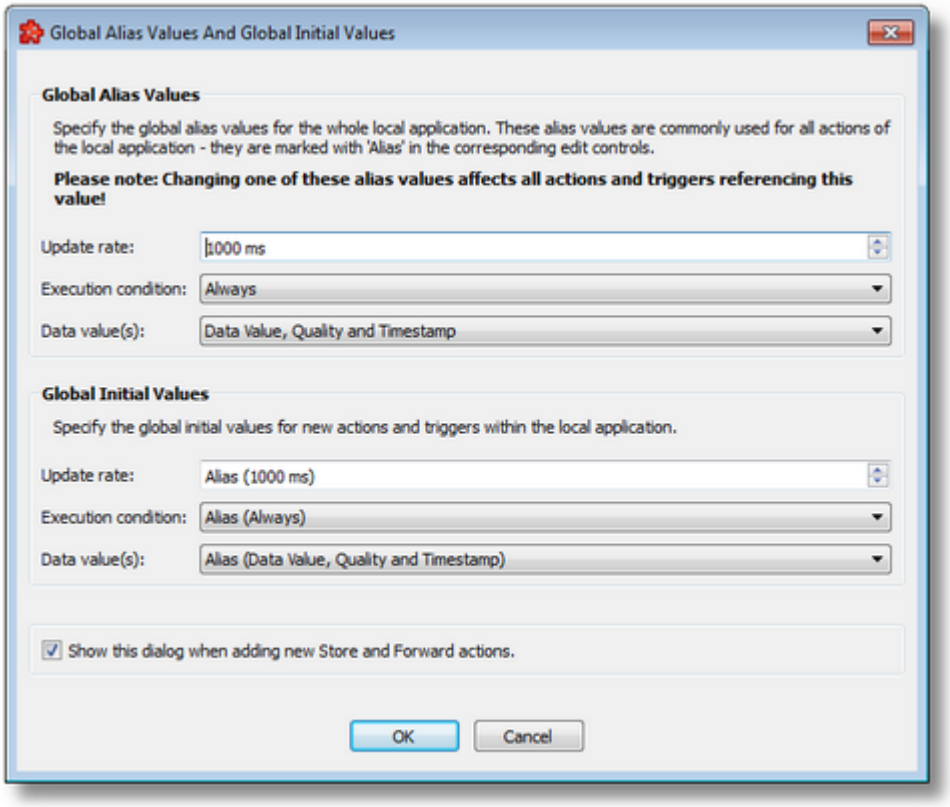
- Maintain global initial and alias values for actions of the local application @ Configuration: Global Initial and Alias Values
- Add or modify Variable Trigger type triggers to actions @ Configuration: Variable Trigger

Softing dataFEED OPC Suite V5.38
© 2024 Softing Industrial Automation GmbH

7.3.5.1 Global Initial and Alias Values

The *Global Alias And Initial Values* dialog allows changing the global alias values and the initial values for update rates, execution conditions and Exchange data values.

It can be triggered by the *Configure global alias and initial values* button  on several configuration pages of the Softing dataFEED OPC Suite Configurator:



In the *Global Alias And Initial Values* dialog the following settings can be changed:

- *Global Alias Values*: The alias values are special placeholders commonly used for all actions in the Local Application. The dataFEED OPC Suite Configurator properties bearing these alias values are marked with the *Alias* keyword next to the value in the corresponding edit control. By changing the *Global Alias Values* in this window all the items bearing alias values will have their corresponding values updated.

- *Global Initial Values:* The initial values are practically the default values for all new actions and triggers created within the dataFEED OPC Suite Configurator. The *Global Initial Values* initially point to the *Global Alias Values*.

The following values can be changed for the global alias and global initial values:

Attribute	Description
Update rate	Specifies the update rate for the corresponding action. The default value is 1000 milliseconds.
Execution condition	Specifies the execution condition for the corresponding action. The following execution conditions are currently defined: • *Always*: Execute the corresponding action without any restrictions, each time the action's trigger is fired. • *All Good Quality*: Execute the corresponding action only if all the items in the incoming transaction have their quality *Good*.

© Softing Industrial Automation GmbH

	• *Item Quality Changed*: Execute the corresponding action only if the quality of at least one of the data items in the incoming transaction has changed. • *One Quality Bad*: Execute the corresponding action only if at least one of the items in the incoming transaction has the quality *Bad*. The default execution condition is *Always*.
Data value(s)	This value is only used for Exchange actions. It defines the OPC data values that should be transferred from the associated source to the associated destination data item by the corresponding Exchange action. The following data values can be selected: • *Data Value*: Only the current data value is transferred from the source to the destination data item. • *Data Value and Quality*: The current data value and the current quality are transferred from the source to the destination data item. • *Data Value, Quality and Timestamp*: The current data value, the current quality and the current timestamp are transferred from the source to the destination data item. The default setting is *Data Value, Quality and Timestamp*. **Attention:** Sending the quality and/or timestamp is not always allowed by all the data destinations. A data destination can forbid the writing of the quality and/or timestamp for an item - in this situation the specifier should be changed to a supported one.

Hysteresis value since last

A separate setting exits for changing the behavior of the Hysteresis parameter of Leading/Falling Edge or Changed variable trigger functions. This setting in the *Global Alias Values* section allows the following values: *Data Changed* (default), *Fired Trigger*.
It is applied globally to all existing expressions of Leading/Falling Edge or Changed functions in all variable triggers. It can be overridden for each expression individually when configuring it on the *Variable Condition Expression Composer Wizard* page for Leading Edge, Falling Edge and Changed functions.

See links to these pages above for more information on how to override this setting.

Behavior:

If the *Hysteresis Value Since Last* parameter has the value *Data Changed* the currently received value is compared against the value the last data change provided, independently of whether the trigger was fired or not.

Subsequently, if the selected value is *Fired Trigger*, the currently received value is compared against the last value that issued the triggering of the variable (note: see also the *Retrigger* setting of the Variable trigger on the Trigger Identity page).

7.3.5.2 Trigger Wizard

- The Trigger Wizard allows modifying the trigger for the current action:

- Obtain general information about the Trigger Wizard @ Trigger Wizard - Welcome
- Pick a trigger type @ Trigger Wizard - Trigger Type
- Select the trigger item @ Trigger Wizard - Trigger Item
- Assign an existing timer trigger @ Trigger Wizard - Timer Selection
- Name the newly created timer trigger @ Trigger Wizard - Timer Identity
- Choose the timer type @ Trigger Wizard - Timer Type
- Configure the corresponding timer schedule @ Trigger Wizard - Schedule Timer Settings
- Define the timer interval @ Trigger Wizard - Interval Timer Settings
- Select the variable trigger @ File Action Wizard - Variable Trigger Selection

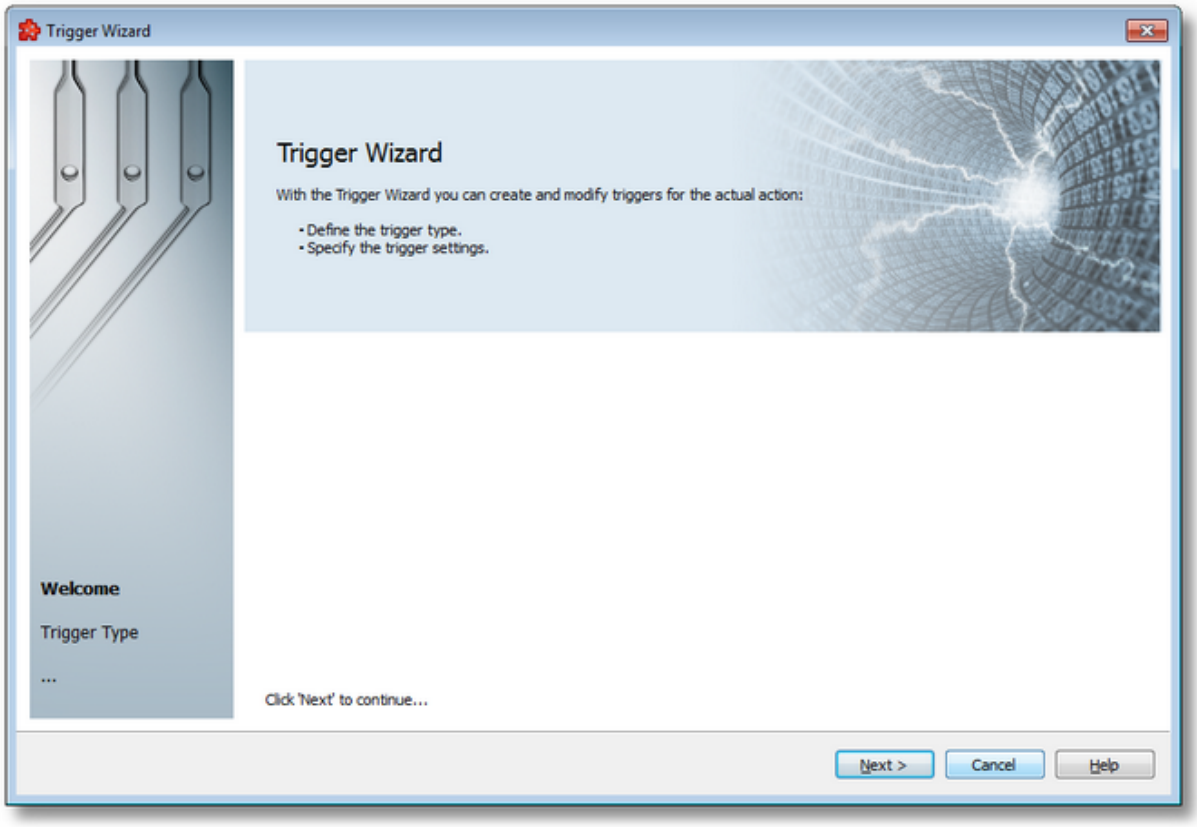
Note: A Trigger Wizard flow will not include all the pages listed above. A specific branch will be followed during the wizard according to the selections performed in some of the pages.

7.3.5.2.1 Welcome

The Welcome wizard page describes the main steps for defining a trigger:

- Choosing a trigger type.
- Configuring the respective trigger settings.

© Softing Industrial Automation GmbH



To proceed with the trigger creation use the *Next* button, or quit the wizard by pressing the *Cancel* button.

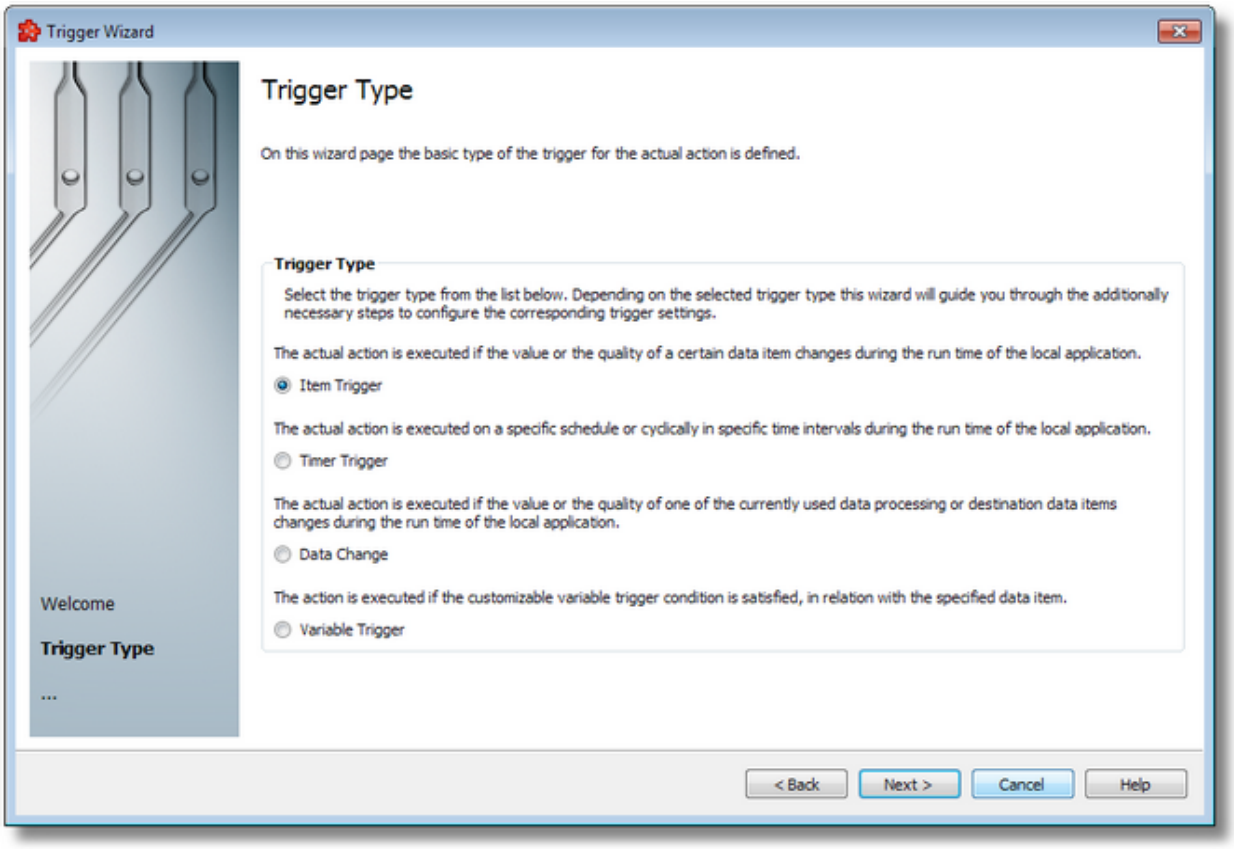
Trigger Wizard - Trigger Type >>>

7.3.5.2.2 Trigger Type

The *Trigger Type* wizard page provides the possibility to select the trigger type for the actual action. The following trigger types are available:

- *Item Trigger* - This trigger is fired when a data change for the selected item occurred. A data change means that the value or the quality of the item changed.
- *Timer Trigger* - This trigger is fired at a specified time or interval.
- *Data Change* - This trigger is fired when a data change occurred in at least one of the items used in any of the consumer add-ons.
- *Variable Trigger* - This trigger is fired when a data change for the items in the variable conditions occurs. A data change means that the value or the quality of the item changed.

Attention: The trigger only initiates the file action execution. The action will be subsequently validated, however, and might not be executed after all.



To proceed with the wizard and continue to define the selected type of trigger (or to advance to the last wizard page when the *Data Change* trigger type is selected) use the *Next* button, to go back to the previous wizard page use the *Back* button, or discard the wizard by pressing the *Cancel* button.

Depending on the selected *Trigger Type* the next wizard page will be:

- Trigger Wizard - Trigger Item - if the selected trigger type is *Item Trigger*,
- Trigger Wizard - Timer Selection - if the selected trigger type is *Timer Trigger*
- Trigger Wizard - Variable Trigger - if the selected trigger type is *Variable Trigger*
- If the selected action is *Data Change* this will be the last wizard page, therefore the *Next* button will become *Finish*.

The description of the Trigger Wizard will continue with the wizard pages for configuring an *Item Trigger*.

<<< Trigger Wizard - Welcome Trigger Wizard - Trigger Item >>>

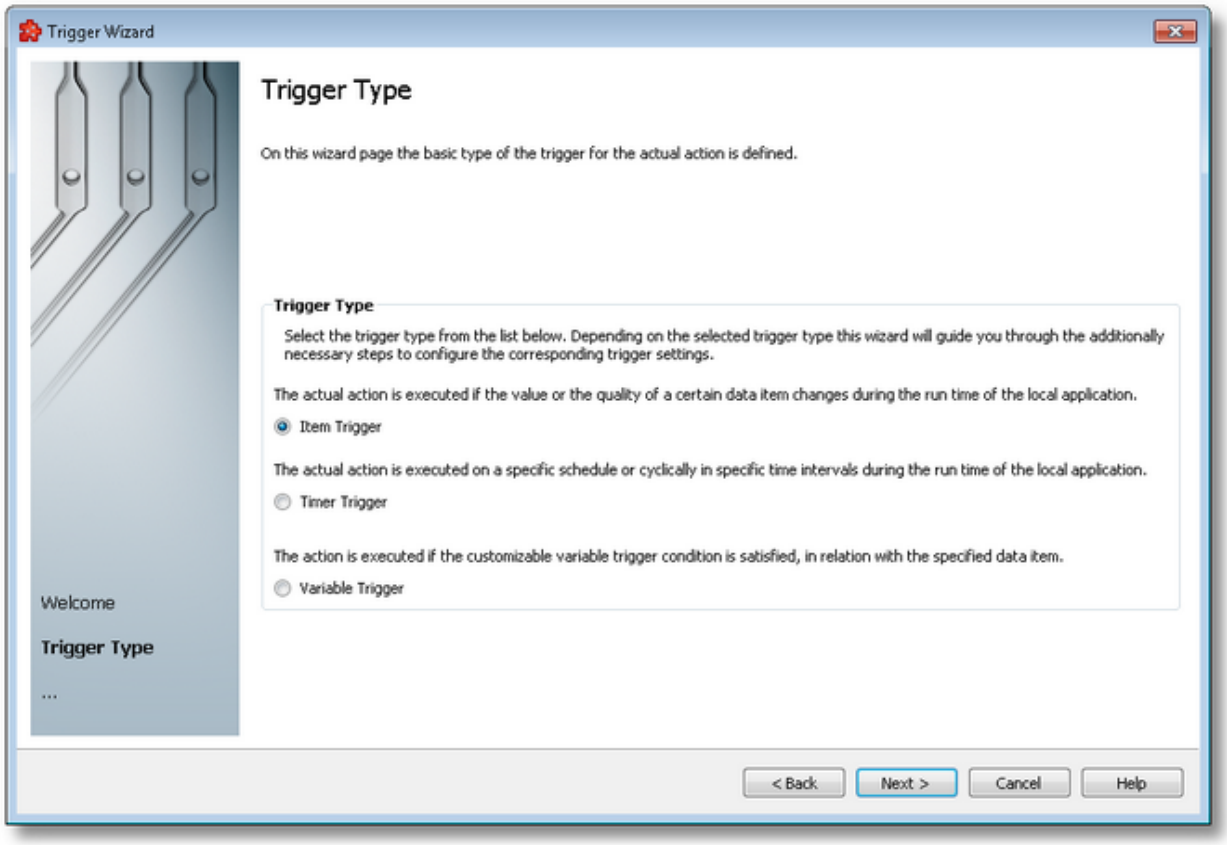
7.3.5.2.3 Trigger Type (Item, Timer, Variable)

The *Trigger Type* wizard page provides the possibility to select the trigger type for the actual action. The following trigger types are available:

© Softing Industrial Automation GmbH

- *Item Trigger* - This trigger is fired when a data change for the selected item occurred. A data change means that the value or the quality of the item changed.
- *Timer Trigger* - This trigger is fired at a specified time or interval.
- *Variable Trigger* - This trigger is fired when a data change for the items in the variable conditions occurs. A data change means that the value or the quality of the item changed.

Attention: The trigger only initiates the file action execution. The action will be subsequently validated, however, and might not be executed after all.



To proceed with the wizard and continue to define the selected type of trigger (or to advance to the last wizard page when the *Data Change* trigger type is selected) use the *Next* button, to go back to the previous wizard page use the *Back* button, or discard the wizard by pressing the *Cancel* button.

Depending on the selected *Trigger Type* the next wizard page will be:

- Trigger Wizard - Trigger Item - if the selected trigger type is *Item Trigger*,
- Trigger Wizard - Timer Selection - if the selected trigger type is *Timer Trigger*
- Trigger Wizard - Variable Trigger - if the selected trigger type is *Variable Trigger*

The description of the Trigger Wizard will continue with the wizard pages for configuring an *Item Trigger*.

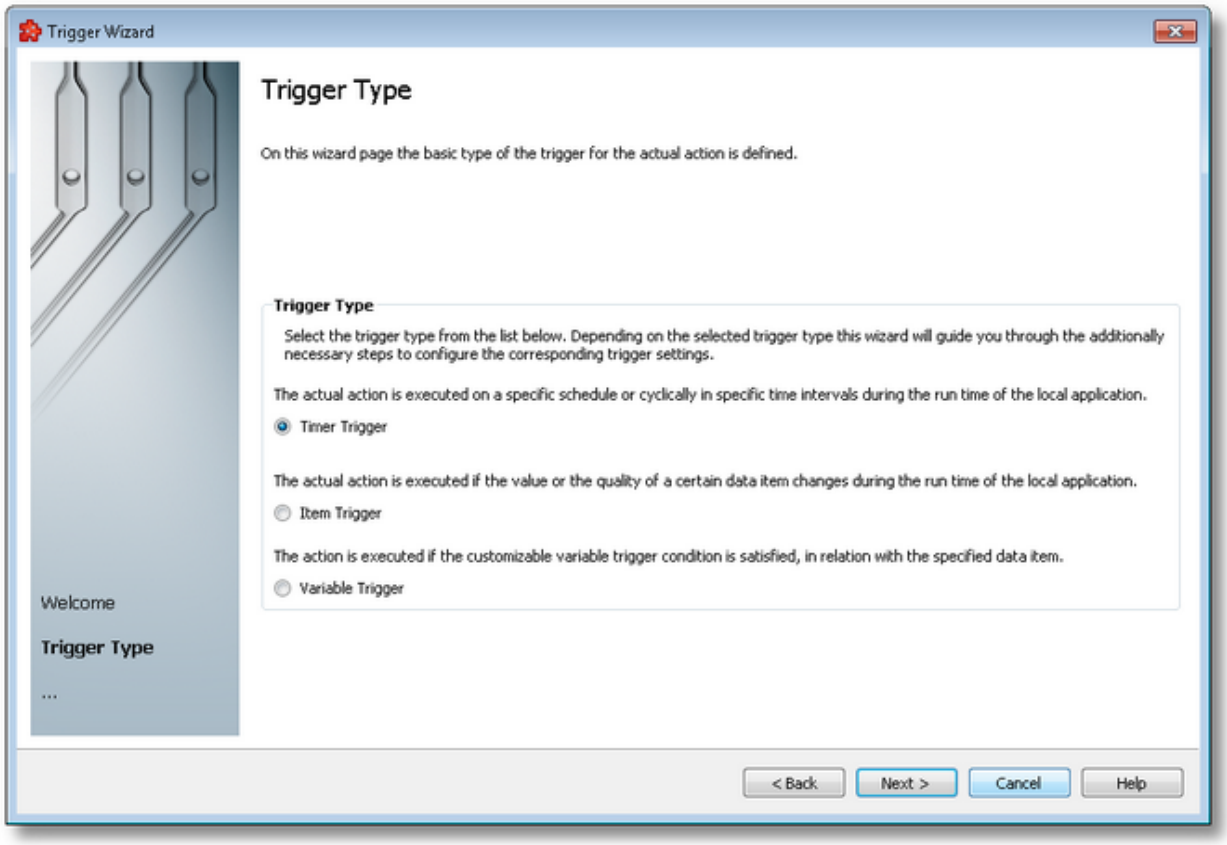
<<< Trigger Wizard - Welcome Trigger Wizard - Trigger Item >>>

7.3.5.2.4 Trigger Type (Timer, Item, Variable)

The *Trigger Type* wizard page provides the possibility to select the trigger type for the actual action. The following trigger types are available:

- *Timer Trigger* - This trigger is fired at a specified time or interval.
- *Item Trigger* - This trigger is fired when a data change for the selected item occurred. A data change means that the value or the quality of the item changed.
- *Variable Trigger* - This trigger is fired when a data change for the items in the variable conditions occurs. A data change means that the value or the quality of the item changed.

Attention: The trigger only initiates the file action execution. The action will be subsequently validated, however, and might not be executed after all.



To proceed with the wizard and continue to define the selected type of trigger (or to advance to the last wizard page when the *Data Change* trigger type is selected) use the *Next* button, to go back to the previous wizard page use the *Back* button, or discard the wizard by pressing the *Cancel* button.

Depending on the selected *Trigger Type* the next wizard page will be:

- Trigger Wizard - Timer Selection - if the selected trigger type is *Timer Trigger*

© Softing Industrial Automation GmbH

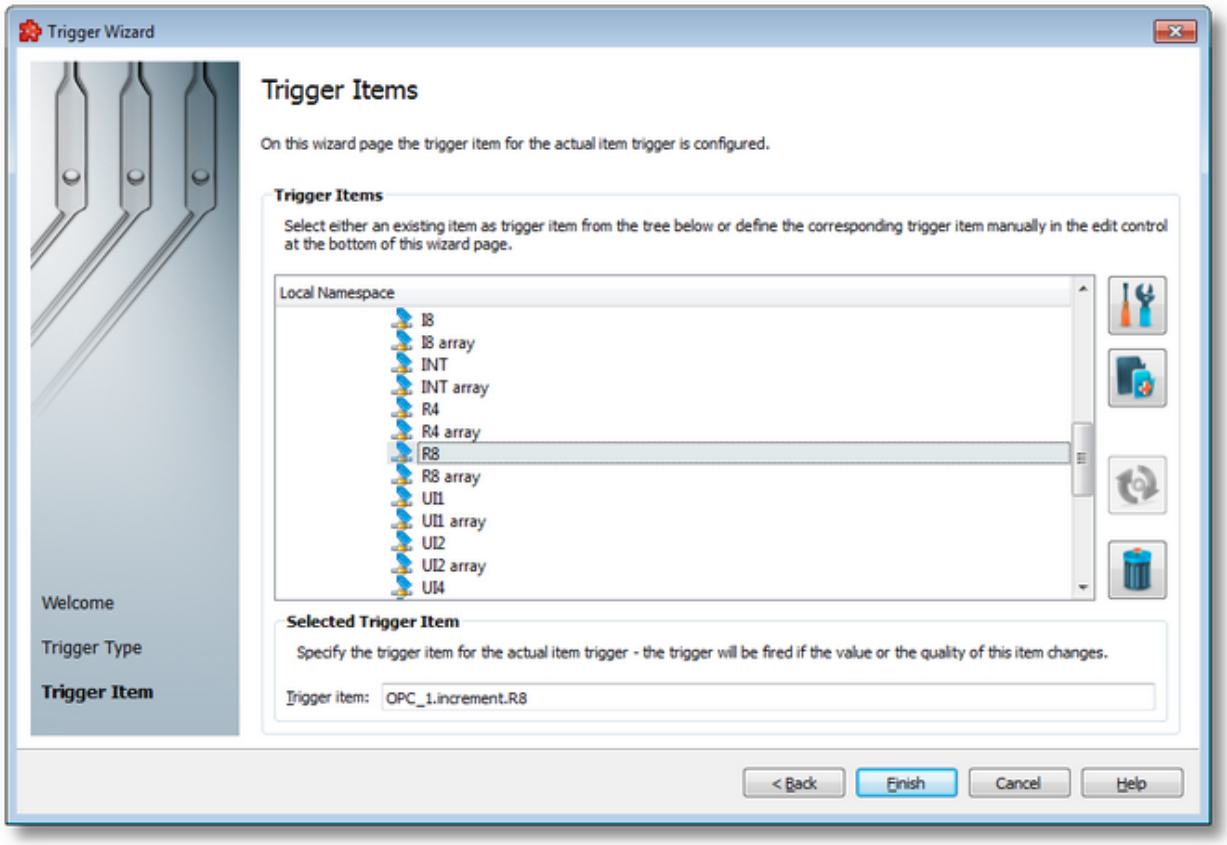
- Trigger Wizard - Trigger Item - if the selected trigger type is *Item Trigger*,
- Trigger Wizard - Variable Trigger - if the selected trigger type is *Variable Trigger*

The description of the Trigger Wizard will continue with the wizard pages for configuring an *Item Trigger*.

<<< Trigger Wizard - Welcome Trigger Wizard - Trigger Item >>>

7.3.5.2.5 Trigger Item

The *Trigger Item* wizard page is shown if the selected trigger type is *Item Trigger*. Here you can configure the actual trigger item.



This wizard page contains two separate sections:

- *Trigger Items* section, which allows selecting an item trigger or managing the Local Namespace tree.
- *Selected Trigger Item* section, which shows the item ID of the selected trigger item and also allows typing an item ID, thereby selecting this item in the above list.

Trigger Items section

The entire namespace of the current configuration is shown as a namespace tree. The name of the root node is composed by appending the instance name to the *Local Namespace* keyword. The first level of the namespace contains:

- A node for each server connection, named by the server connection items prefix.
- A node for System Items.
- A node for Local Items.

The local namespace tree offers the possibility to explore the entire namespace of the local dataFEED OPC Suite application in order to select a trigger item. After the selection is made, the *Trigger item* edit box is populated with the selected item ID.

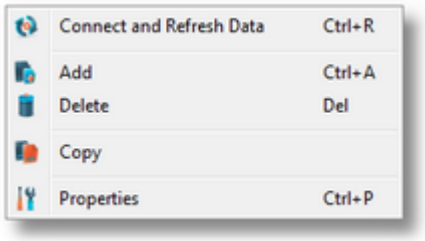
The entire namespace of the local dataFEED OPC Suite application can be managed through this wizard page by performing certain operations using the buttons on the right, the context sensitive menu or the keyboard shortcuts:

- Open the item properties dialog using the *Properties of the selected item* button  , pressing the keyboard shortcut CTRL+P or selecting the *Properties* action from the context menu. Depending on the type of the selected item, a special Item Properties dialog opens:
  - If the selected item is under Local Items the Modify Local Item dialog is opened.
  - If the selected item is under a dynamic server connection the Modify Dynamic Server Item dialog is opened.
  - If the selected item is under a static server connection the Modify Static Server Item dialog is opened.
- Add a new item using the *Add a new item*  button, pressing the keyboard shortcut CTRL+A or selecting the *Add* action from the context menu. Depending on the type of the selected node, a special Add Item dialog opens:
  - If the selected node is under Local Items the Add Local Item dialog is opened.
  - If the selected node is under a dynamic server connection the Add Dynamic Server Item dialog is opened.
  - If the selected node is under a static server connection the Add Static Server Item dialog is opened.
- Copy the selected item to clipboard selecting the *Copy* action from the context menu.
- Paste the previously copied item from clipboard by selecting the *Paste* action from the context menu. Items originating from the Local Items cannot be pasted under server connections, and vice-versa; an error message will be displayed in both cases. If an item is pasted in a static (filtered) server connection and the item does not exist on the server an error message is displayed.
- Connect to the server and refresh the data items by using the *Refresh data items*  button, pressing the keyboard shortcut CTRL+R or selecting the *Connect and Refresh Data* action from the context menu. The namespace of the selected server connection is recreated; if manually added items not referenced in the actual configuration exist, those items are deleted from the namespace.

© Softing Industrial Automation GmbH

- Delete the selected item and all of its child items by using the Delete data items button  , pressing the DELETE / DEL key or selecting the Delete action from the context menu. Items which are included in any Data Processing / Data Destination add-ons will not be deleted but an error message will be displayed instead.

An example of the context menu can be seen below:



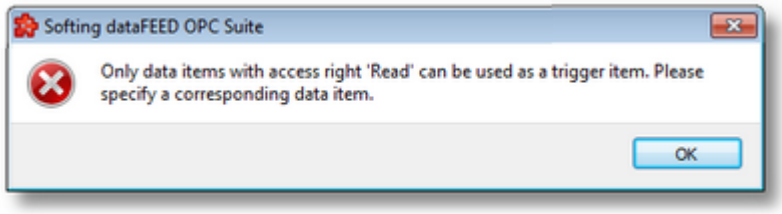
Selected Trigger Item section

The edit box in this section is populated with the item ID after an item from the Local Namespace tree is selected.

You can also manually type the ID of the item in the *Trigger item* edit box and thus have the above tree automatically select a valid server item.

Note: If the item ID is typed and the specified item exists in the Local Namespace, it will be used as the trigger item. The item will only be created if it could not be localized in the Local Namespace. In that case the newly created item will be used as the trigger item.

Note: If the item corresponding to the typed item ID could not be created, an error message will be displayed requesting the input of a new, valid item.



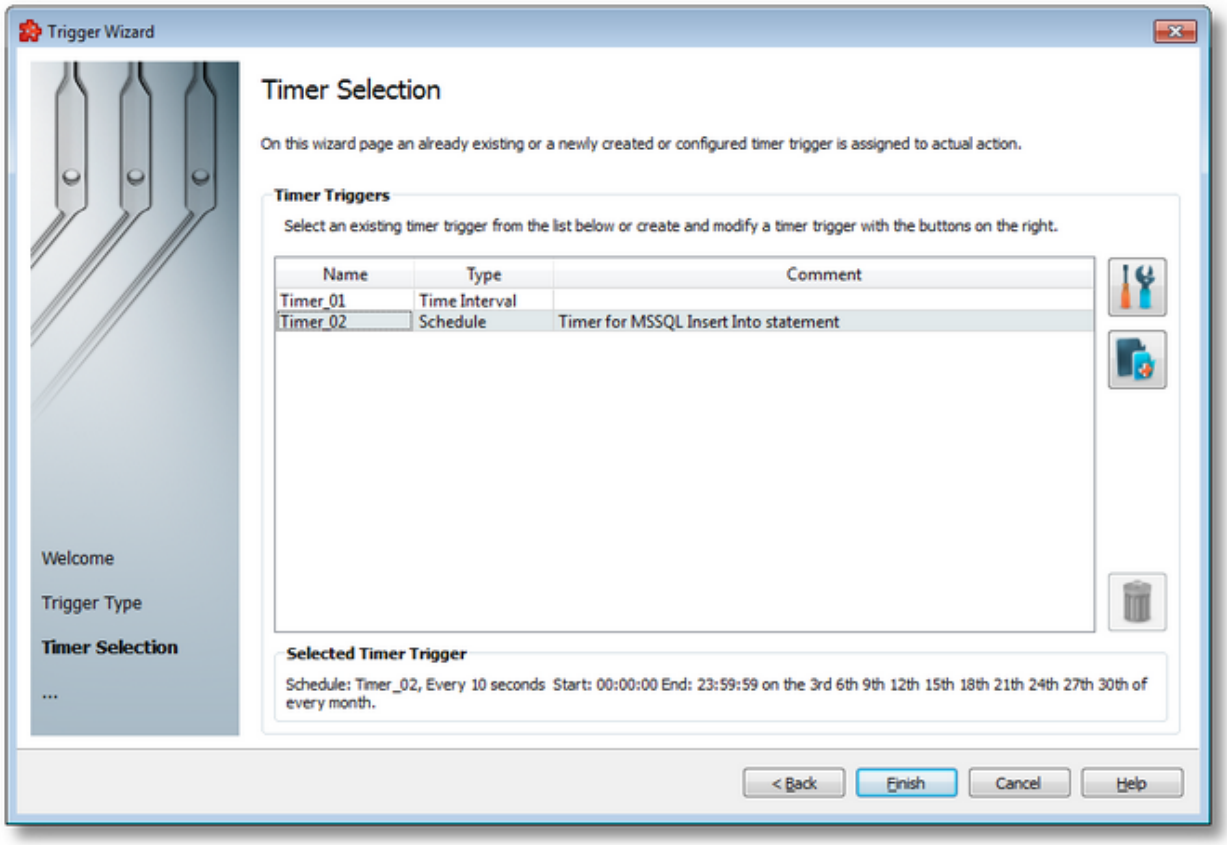
Attention: If the item corresponding to the typed item ID is valid and resides under a static server connection, the respective item will be created in the filter file definition.

Complete the wizard using the *Finish* button or return to the previous wizard page using the *Back* button. To exit the wizard and discard all the changes click the *Cancel* button.

<<< Trigger Wizard - Trigger Type

7.3.5.2.6 Timer Selection

The *Timer Selection* wizard page is shown only if the selected trigger type is *Timer Trigger*. Here you can configure the actual timer trigger.



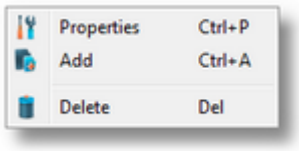
The current page allows selecting a previously defined timer trigger and managing the timer list by using using the buttons on the right, the context sensitive menu or the keyboard shortcuts:

- Change the properties of the selected timer by using the *Properties* button  , pressing the keyboard shortcut CTRL+P or selecting the *Properties* action from the context menu.
- Add a new timer by using the Add timer trigger button  , pressing the keyboard shortcut CTRL+A or selecting the Add action from the context menu.
- Delete the selected timer trigger by using the Delete selected timer trigger button  , pressing the DELETE / DEL key or selecting the Delete action from the context menu.

Note: If a timer trigger has associated actions it cannot be deleted.

An example of the context sensitive menu can be seen below:

© Softing Industrial Automation GmbH



Note: Information about the selected timer is shown in the *Selected Timer Trigger* area at the bottom of the page.

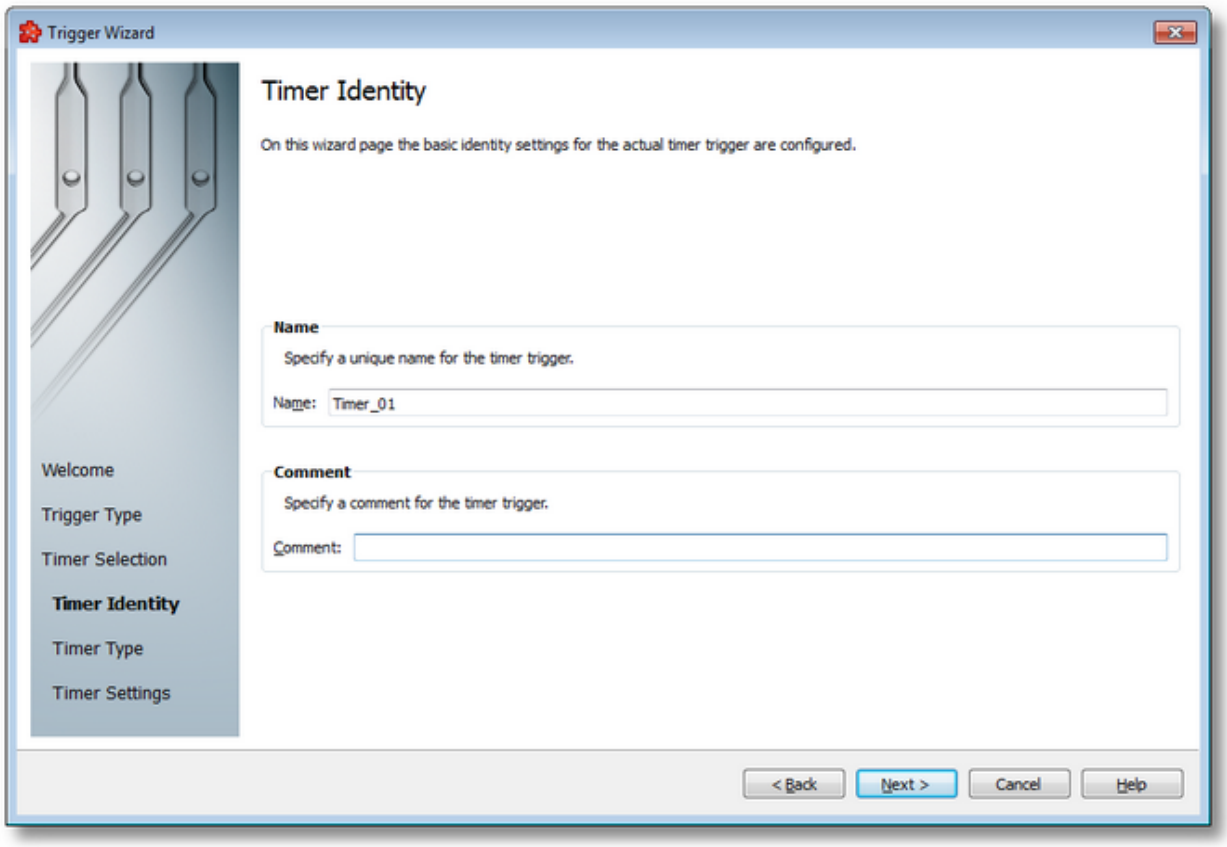
A sub-wizard that allows adding or modifying timer triggers can be launched by selecting to change an existing timer trigger or to add a new one. The first page of this sub-wizard is Trigger Wizard - Timer Identity.

Click the *Finish* button to finish the wizard, or use the *Back* button to return to the previous wizard page. To exit the wizard click the *Cancel* button.

<<< Trigger Wizard - Trigger Item

7.3.5.2.7 Timer Identity

The *Timer Identity* wizard page provides the possibility to assign a name and a comment to the timer trigger.

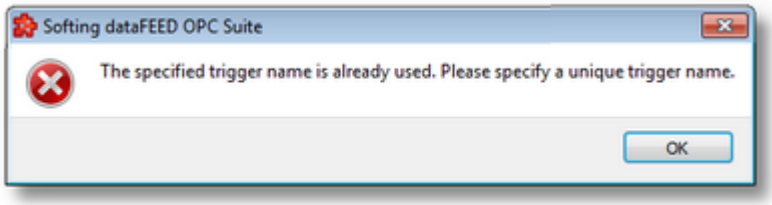


Note: When adding a new timer trigger, the timer name will be automatically generated according to the format: *Timer_number*, where the *number* part is an auto-incremented integer.

When editing an existing timer, the timer's defined name and comment will be filled in the two edit boxes allowing you to modify these attributes.

Attention: The name of the *Timer Trigger* must be unique among all triggers and cannot be empty.

If the newly added timer trigger has the same name as an existing trigger, an error message box will be displayed and a new trigger name must be provided:



The *Comment* field of the *Timer Identity* wizard page is optional; a Timer Trigger can also be added without any comment.

Continue to the next wizard page to choose the timer type by using the *Next* button, go back to the previous wizard page to select another timer trigger by pressing the *Back* button, or cancel the entire *Trigger Wizard* by pressing the *Cancel* button.

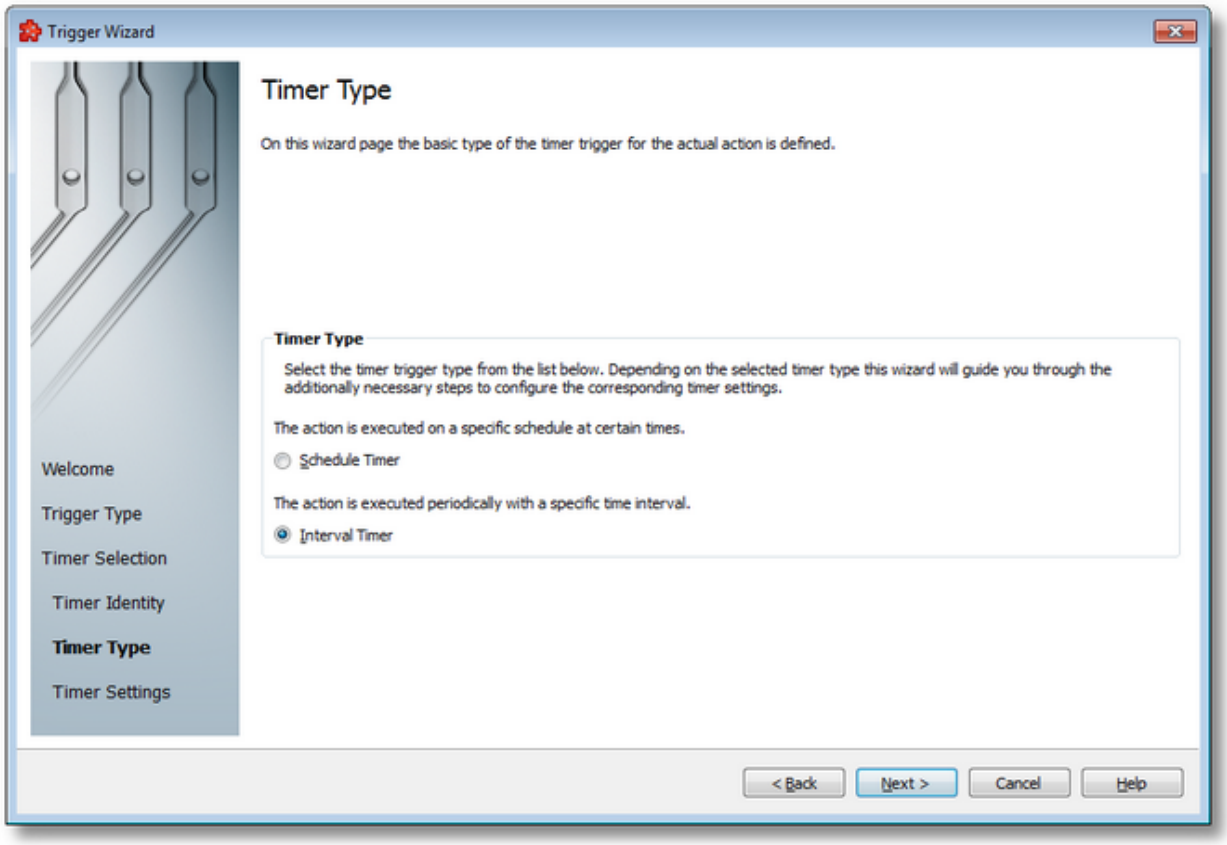
<<< Trigger Wizard - Timer Selection Trigger Wizard - Timer Type >>>

7.3.5.2.8 Timer Type

This wizard page allows choosing the type of the timer trigger. The following types of timer are available:

- *Interval Timer*
- *Schedule Timer*

© Softing Industrial Automation GmbH



Interval Timer

The Interval Timer type allows defining a specific interval of time (in milliseconds) between the timer fire events. The time interval can be configured on the Trigger Wizard - Interval Timer Settings wizard page.

Schedule Timer

The Schedule Timer type allows defining a specific date and time when the timer should fire. The schedule can be configured on the Trigger Wizard - Schedule Timer Settings wizard page.

Note: The *Timer Type* of an existing *Timer Trigger* cannot be changed.

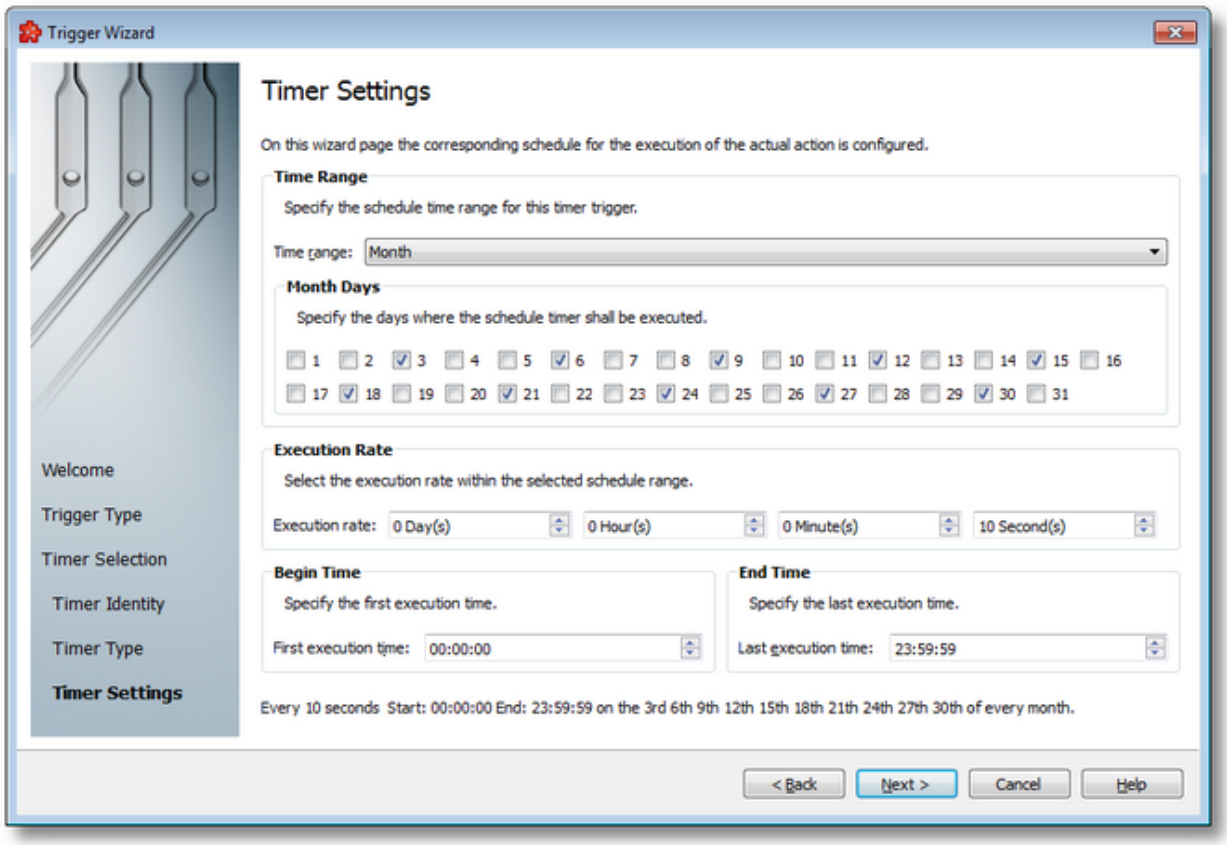
To continue defining the specific timer properties press the *Next* button, to return to the *Timer Identity* wizard page use the *Back* button, or exit this sub-wizard (thereby aborting the *Trigger Wizard*) by pressing the *Cancel* button.

The description of the Trigger Wizard will continue with the wizard pages for configuring a *Schedule Timer*.

<<< Trigger Wizard - Timer Identity Trigger Wizard - Schedule Timer Settings >>>

7.3.5.2.9 Schedule Timer Settings

The *Schedule Timer Settings* wizard page is a more complex page compared to the *Interval Timer Settings* wizard page and allows defining the corresponding schedule for the execution of the trigger action:



The page has four sections which identify the schedule:

- *Time Range*: Provides the possibility to choose one of the following schedule ranges: *month*, *week*, *day*, *hour* or *minute*. When the *month* or the *week* ranges are selected the panel below the selection is populated with the month and week days, respectively.
- *Execution Rate*: Sets the execution rate of the current action in the previously-defined *Time Range* (e.g. every X day(s), every Y hour(s), every Z minute(s), every T second(s)).
- *Begin Time*: Sets the time the current action should begin execution.
- *End Time*: Sets the time the current action should stop execution.

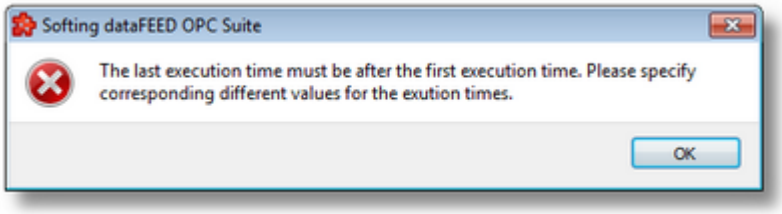
According to the *Time Range* the *Begin Time* and the *End Time* fields will allow typing:

- Seconds (SS) if the *Time range* is *Minute*.
- Minutes and seconds (MM:SS) if the *Time range* is *Hour*.
- Hours, minutes and seconds (HH:MM:SS) if the *Time range* is *Day* or greater (*Week*, *Month*).

© Softing Industrial Automation GmbH

Note: A short review of the configured timer schedule is shown at the bottom of this wizard page.

If the defined schedule is not valid, an error message dialog will appear naming the error that occurred.

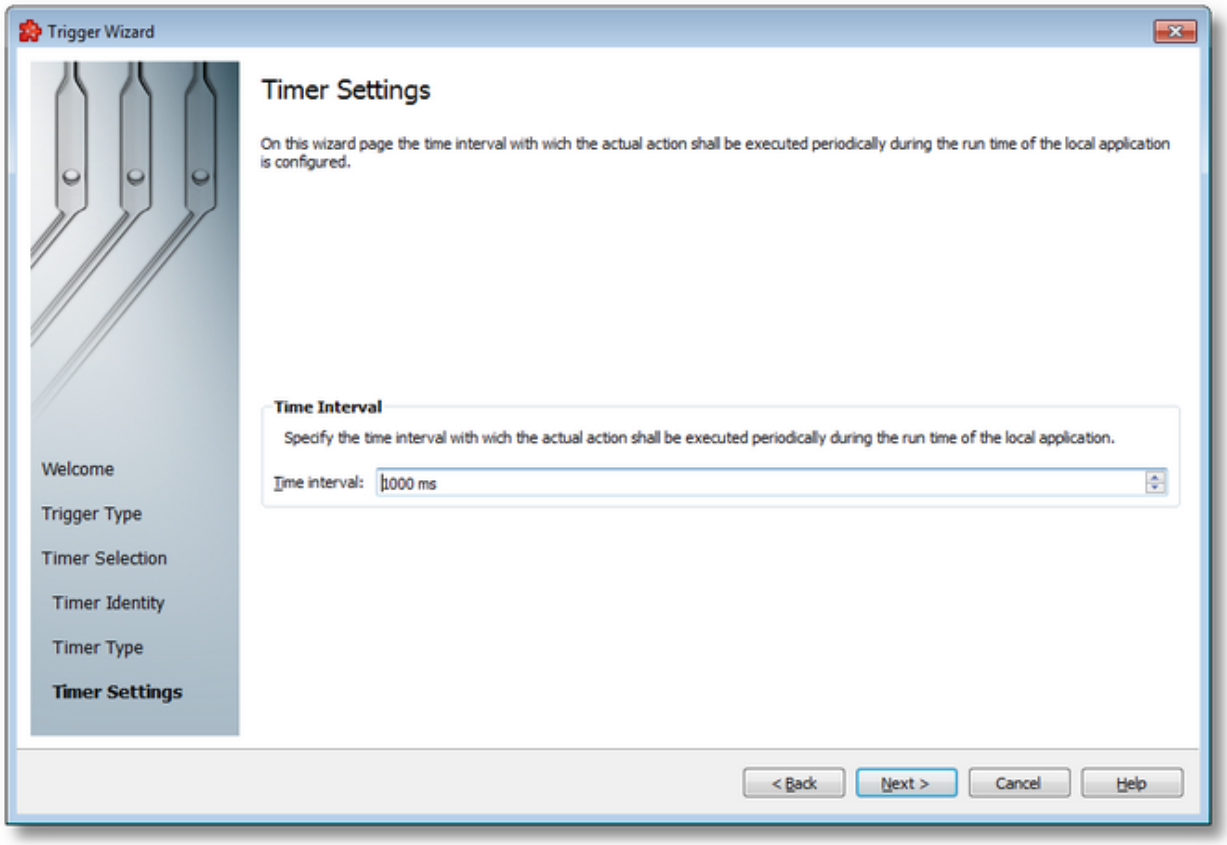


Continue to the *Timer Selection* wizard page using the *Next* button, return to the previous wizard page using the *Back* button or exit the wizard pressing the *Cancel* button.

<<< Trigger Wizard - Timer Type Trigger Wizard - Timer Selection >>>

7.3.5.2.10Interval Timer Settings

The Interval Timer Settings wizard page allows configuring the time interval at which the current action should be (periodically) executed.



The time interval value is measured in milliseconds and ensures that the action will be triggered at the time interval defined in the spin-box.

Continue to the *Timer Selection* wizard page by pressing the *Next* button, return to the previous one by using the *Back* button, or exit the wizard by pressing the *Cancel* button.

<<< Trigger Wizard - Timer Type Trigger Wizard - Timer Selection >>>

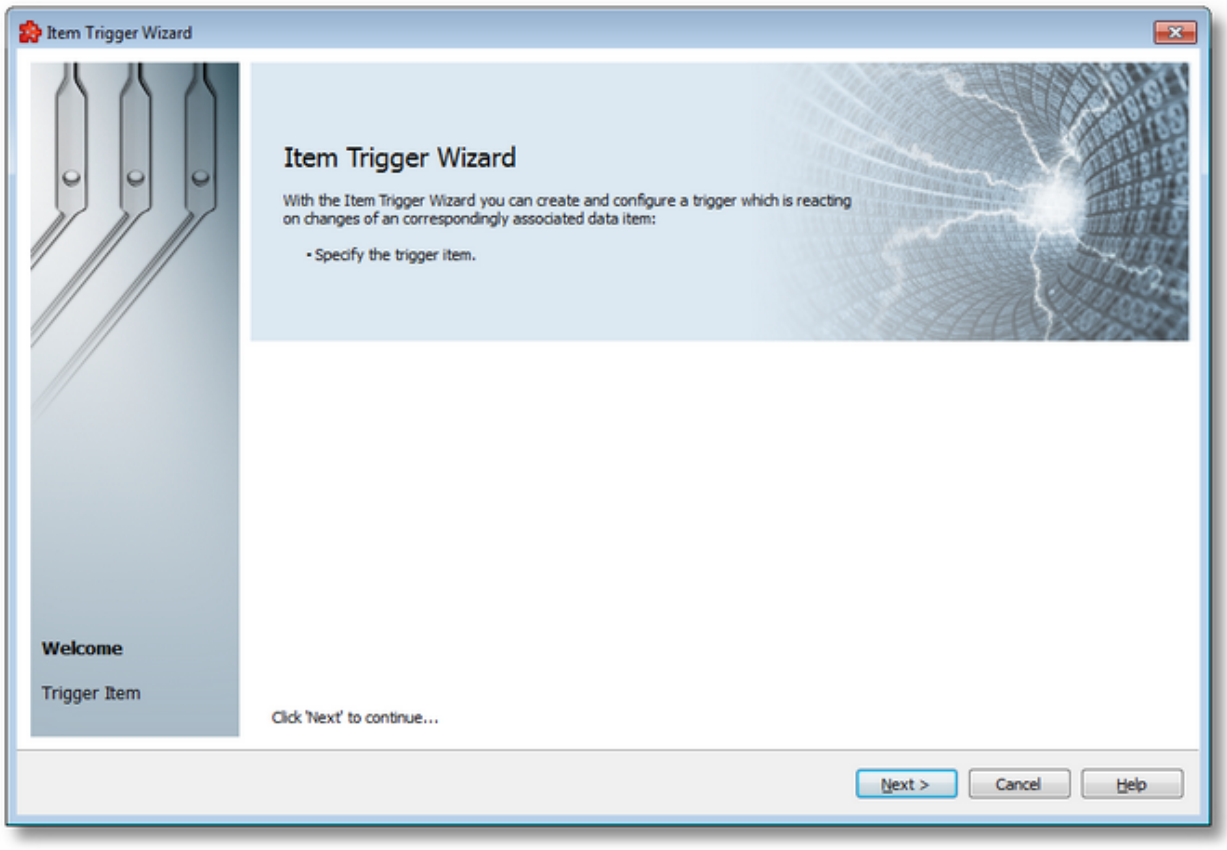
7.3.5.3 Item Trigger Wizard

- The Item Trigger Wizard allows modifying the item trigger for the current action:

- Obtain general information about the Item Trigger Wizard @ Item Trigger Wizard - Welcome
- Configure the trigger item @ Item Trigger Wizard - Trigger Item

7.3.5.3.1 Welcome

The Welcome page describes the basic steps for completing the Item Trigger definition.



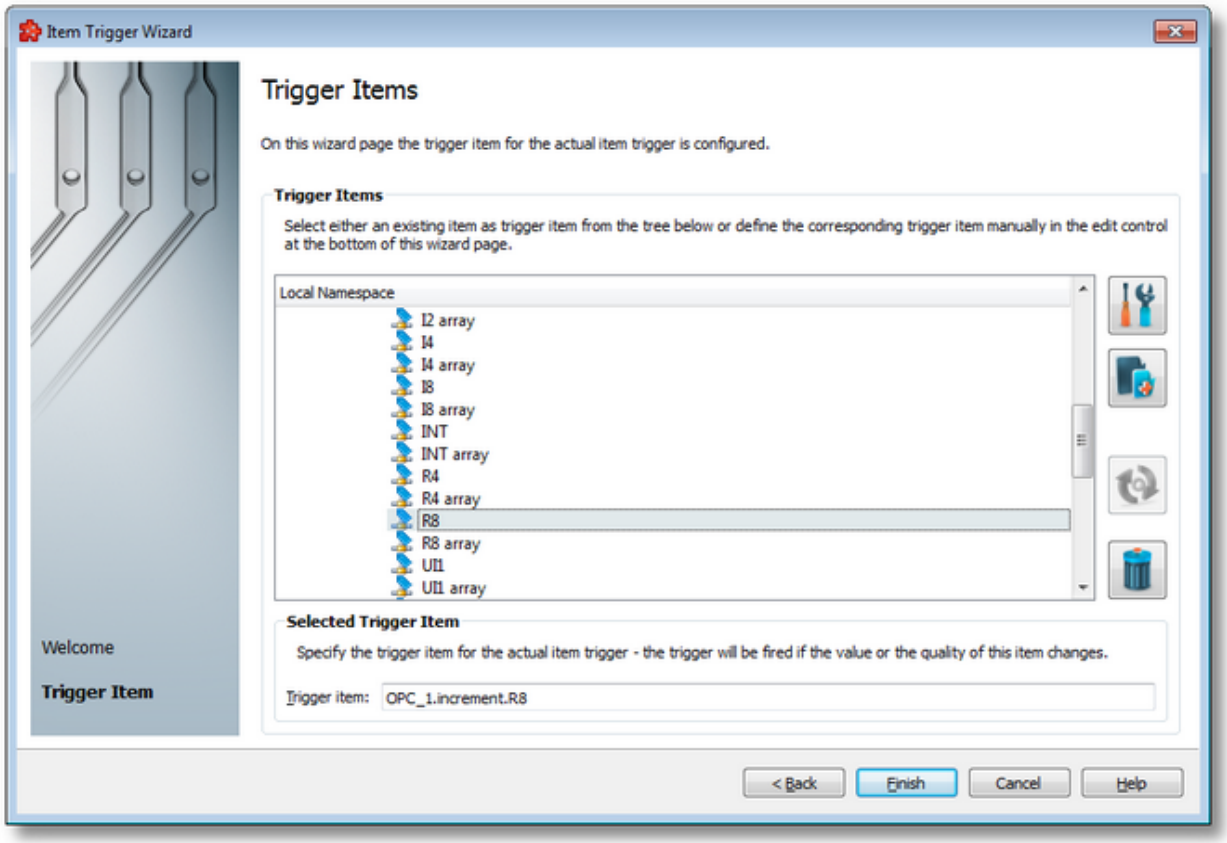
© Softing Industrial Automation GmbH

Proceed to the next page by pressing the *Next* button or exit the wizard by pressing the *Cancel* button.

Item Trigger Wizard - Trigger Item >>>

7.3.5.3.2 Trigger Item

On the Trigger Item wizard page you can configure the trigger item you want to use.



This wizard page contains two separate sections:

- *Trigger Items* section, which allows selecting an item trigger or managing the Local Namespace tree.
- *Selected Trigger Item* section, which shows the item ID of the selected trigger item and also allows typing an item ID, thereby selecting this item in the above list.

Trigger Items section

The entire namespace of the current configuration is shown as a namespace tree. The name of the root node is composed by appending the instance name to the *Local Namespace* keyword. The first level of the namespace contains:

- A node for each server connection, named by the server connection items prefix.
- A node for System Items.
- A node for Local Items.

The local namespace tree offers the possibility to explore the entire namespace of the local dataFEED OPC Suite application in order to select a trigger item. After the selection is made, the *Trigger item* edit box is populated with the selected item ID.

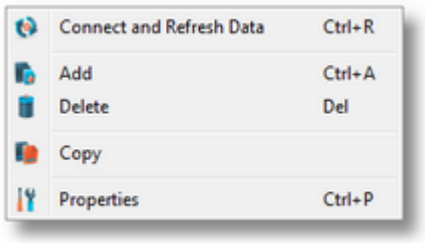
The entire namespace of the local dataFEED OPC Suite application can be managed through this wizard page by performing certain operations using the buttons on the right, the context sensitive menu or the keyboard shortcuts:

- Open the item properties dialog using the *Properties of the selected item* button  , pressing the keyboard shortcut CTRL+P or selecting the *Properties* action from the context menu. Depending on the type of the selected item, a special Item Properties dialog opens:
  - If the selected item is under Local Items the Modify Local Item dialog is opened.
  - If the selected item is under a dynamic server connection the Modify Dynamic Server Item dialog is opened.
  - If the selected item is under a static server connection the Modify Static Server Item dialog is opened.
- Add a new item using the *Add a new item*  button, pressing the keyboard shortcut CTRL+A or selecting the *Add* action from the context menu. Depending on the type of the selected node, a special Add Item dialog opens:
  - If the selected node is under Local Items the Add Local Item dialog is opened.
  - If the selected node is under a dynamic server connection the Add Dynamic Server Item dialog is opened.
  - If the selected node is under a static server connection the Add Static Server Item dialog is opened.
- Copy the selected item to clipboard selecting the *Copy* action from the context menu.
- Paste the previously copied item from clipboard by selecting the *Paste* action from the context menu. Items originating from the Local Items cannot be pasted under server connections, and vice-versa; an error message will be displayed in both cases. If an item is pasted in a static (filtered) server connection and the item does not exist on the server an error message is displayed.
- Connect to the server and refresh the data items by using the *Refresh data items*  button, pressing the keyboard shortcut CTRL+R or selecting the *Connect and Refresh Data* action from the context menu. The namespace of the selected server connection is recreated; if manually added items not referenced in the actual configuration exist, those items are deleted from the namespace.
- Delete the selected item and all of its child items by using the Delete data items button  , pressing the DELETE / DEL key or selecting the Delete action from the context menu.

© Softing Industrial Automation GmbH

Items which are included in any Data Processing / Data Destination add-ons will not be deleted but an error message will be displayed instead.

An example of the context menu can be seen below:



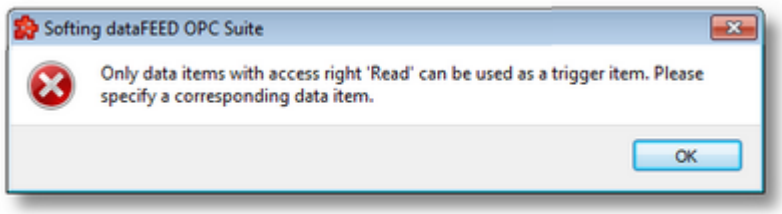
Selected Trigger Item section

The edit box in this section is populated with the item ID after an item from the Local Namespace tree is selected.

You can also manually type the ID of the item in the *Trigger item* edit box and thus have the above tree automatically select a valid server item.

Note: If the item ID is typed and the specified item exists in the Local Namespace, it will be used as the trigger item. The item will only be created if it could not be localized in the Local Namespace. In that case the newly created item will be used as the trigger item.

Note: If the item corresponding to the typed item ID could not be created, an error message will be displayed requesting the input of a new, valid item.



Attention: If the item corresponding to the typed item ID is valid and resides under a static server connection, the respective item will be created in the filter file definition.

Complete the wizard using the *Finish* button or return to the previous wizard page using the *Back* button. To exit the wizard and discard all the changes click the *Cancel* button.

<<< Item Trigger Wizard - Welcome

7.3.5.4 Timer Trigger Wizard

The Timer Trigger Wizard allows modifying the timer trigger for the current action:

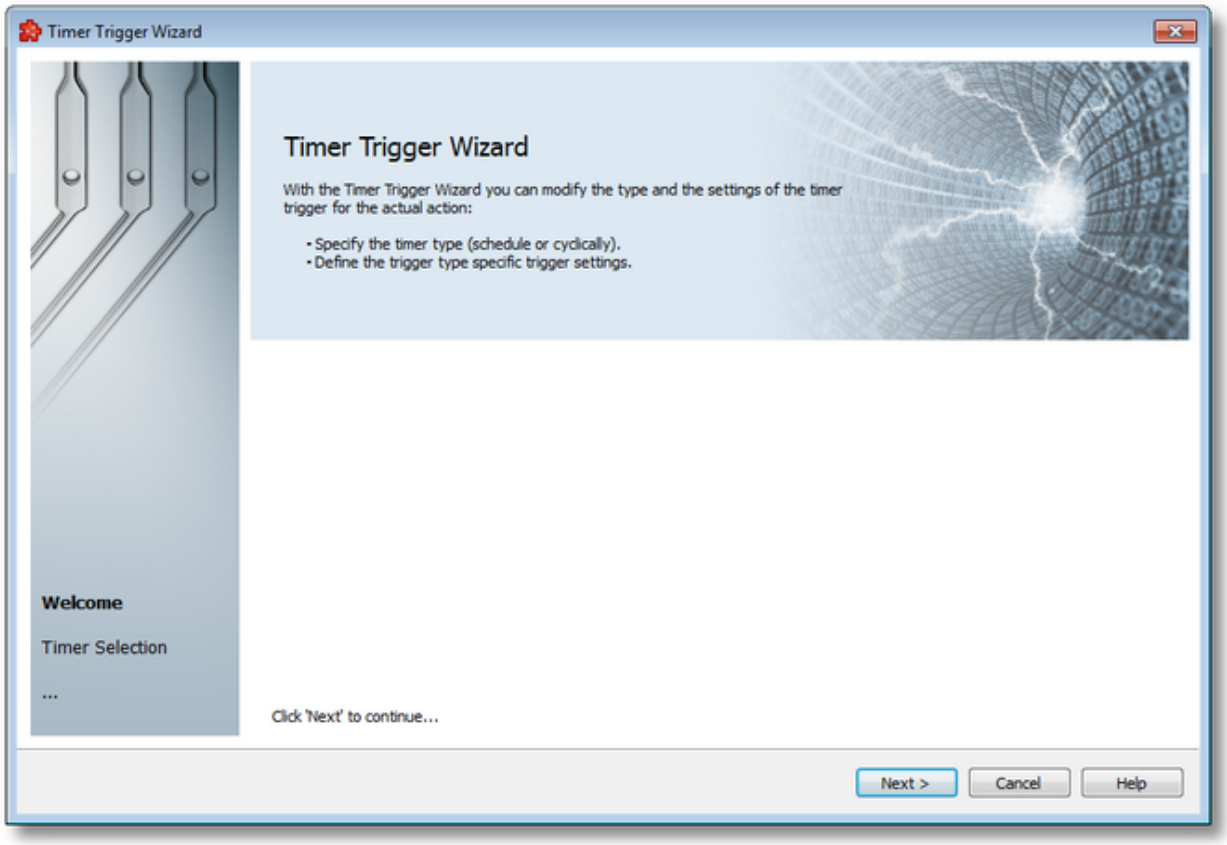
- Obtain general information about the Timer Trigger @ Timer Trigger Wizard - Welcome
- Select an existing timer trigger @ Timer Trigger Wizard - Timer Selection
- Name the newly created timer trigger @ Timer Trigger Wizard - Timer Identity
- Choose the timer type @ Timer Trigger Wizard - Timer Type
- Configure the corresponding timer schedule @ Timer Trigger Wizard - Schedule Timer Settings
- Define the timer interval @ Timer Trigger Wizard - Interval Timer Settings

Note: A Timer Trigger Wizard flow will not include all the pages listed above. A specific branch will be followed during the wizard according to the selections performed in some of the pages.

7.3.5.4.1 Welcome

The *Welcome* wizard page describes the basic steps for completing the timer definition.

© Softing Industrial Automation GmbH

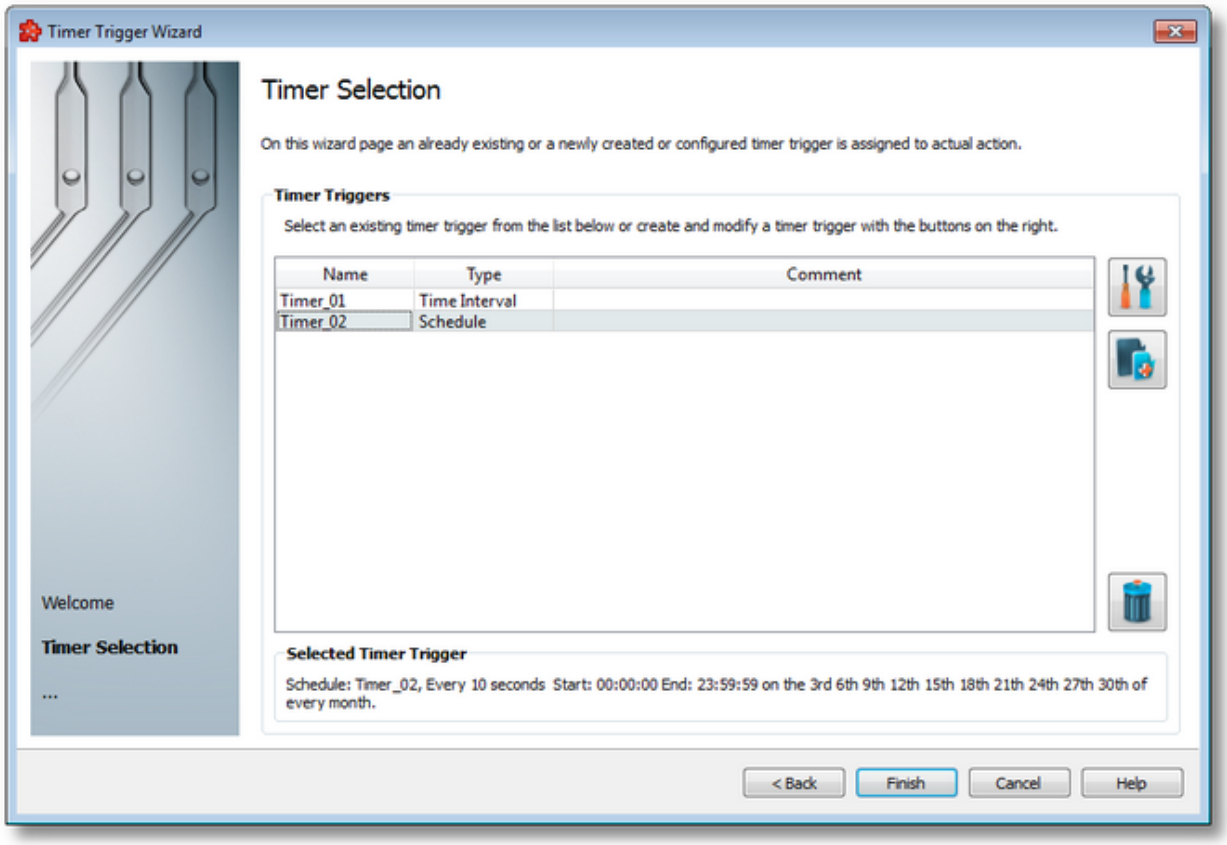


Continue to the next wizard page by using the *Next* button or quit the wizard by pressing the *Cancel* button.

Timer Trigger Wizard - Timer Selection >>>

7.3.5.4.2 Timer Selection

The Timer Selection wizard page allows configuring the actual timer trigger.

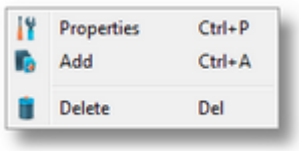


The current page allows selecting a previously defined timer trigger and managing the timer list by using the buttons on the right, the context sensitive menu or the keyboard shortcuts:

- Change the properties of the selected timer by using the *Properties* button  , pressing the keyboard shortcut CTRL+P or selecting the *Properties* action from the context menu.
- Add a new timer by using the Add timer trigger button  , pressing the keyboard shortcut CTRL+A or selecting the Add action from the context menu.
- Delete the selected timer trigger by using the Delete selected timer trigger button  , pressing the DELETE / DEL key or selecting the Delete action from the context menu.

Note: If a timer trigger has associated actions it cannot be deleted.

An example of the context sensitive menu can be seen below:



Note: Information about the selected timer is shown in the *Selected Timer Trigger* area at the bottom of the page.

© Softing Industrial Automation GmbH

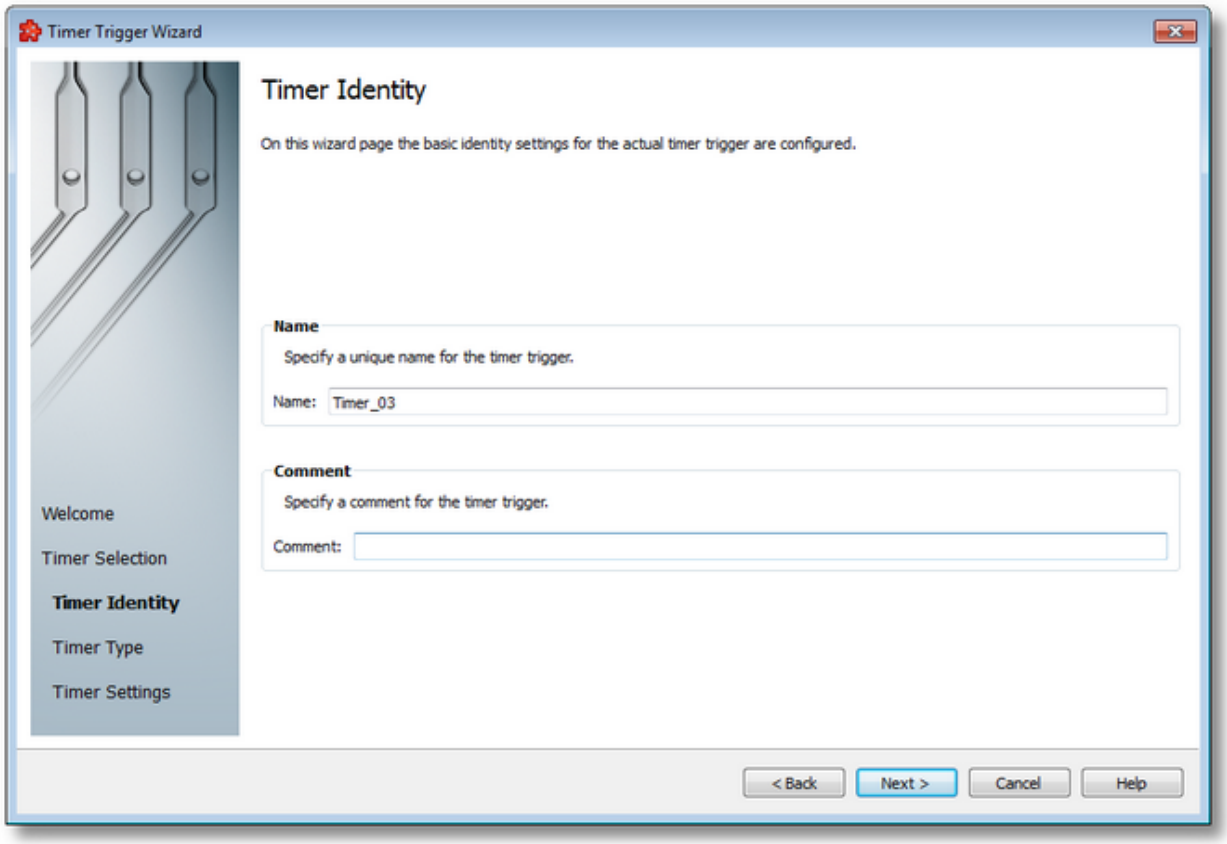
A sub-wizard that allows adding or modifying timer triggers can be launched by selecting to change an existing timer trigger or to add a new one. The first page of this sub-wizard is Trigger Wizard - Timer Identity.

Click the *Finish* button to finish the wizard, or use the *Back* button to return to the previous wizard page. To exit the wizard click the *Cancel* button.

<<< Timer Trigger Wizard - Welcome

7.3.5.4.3 Timer Identity

The Timer Identity wizard page provides the possibility to assign a name and a comment to the timer trigger.

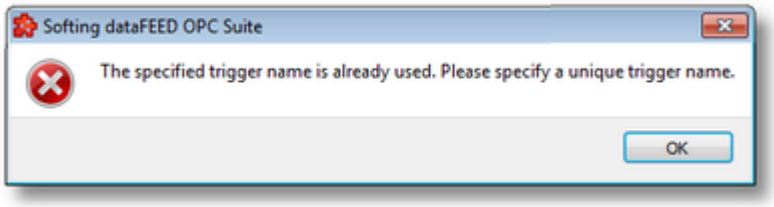


Note: When adding a new timer trigger, the timer name will be automatically generated according to the format: *Timer_number*, where the *number* part is an auto-incremented integer.

When editing an existing timer, the timer's defined name and comment will be filled in the two edit boxes allowing you to modify these attributes.

Attention: The name of the *Timer Trigger* must be unique among all triggers and cannot be empty.

If the newly added timer trigger has the same name as an existing trigger, an error message box will be displayed and a new trigger name must be provided:



The *Comment* field of the *Timer Identity* wizard page is optional; a Timer Trigger can also be added without any comment.

Continue to the next wizard page to choose the timer type by using the *Next* button, go back to the previous wizard page to select another timer trigger by pressing the *Back* button, or cancel the entire *Timer Trigger Wizard* by pressing the *Cancel* button.

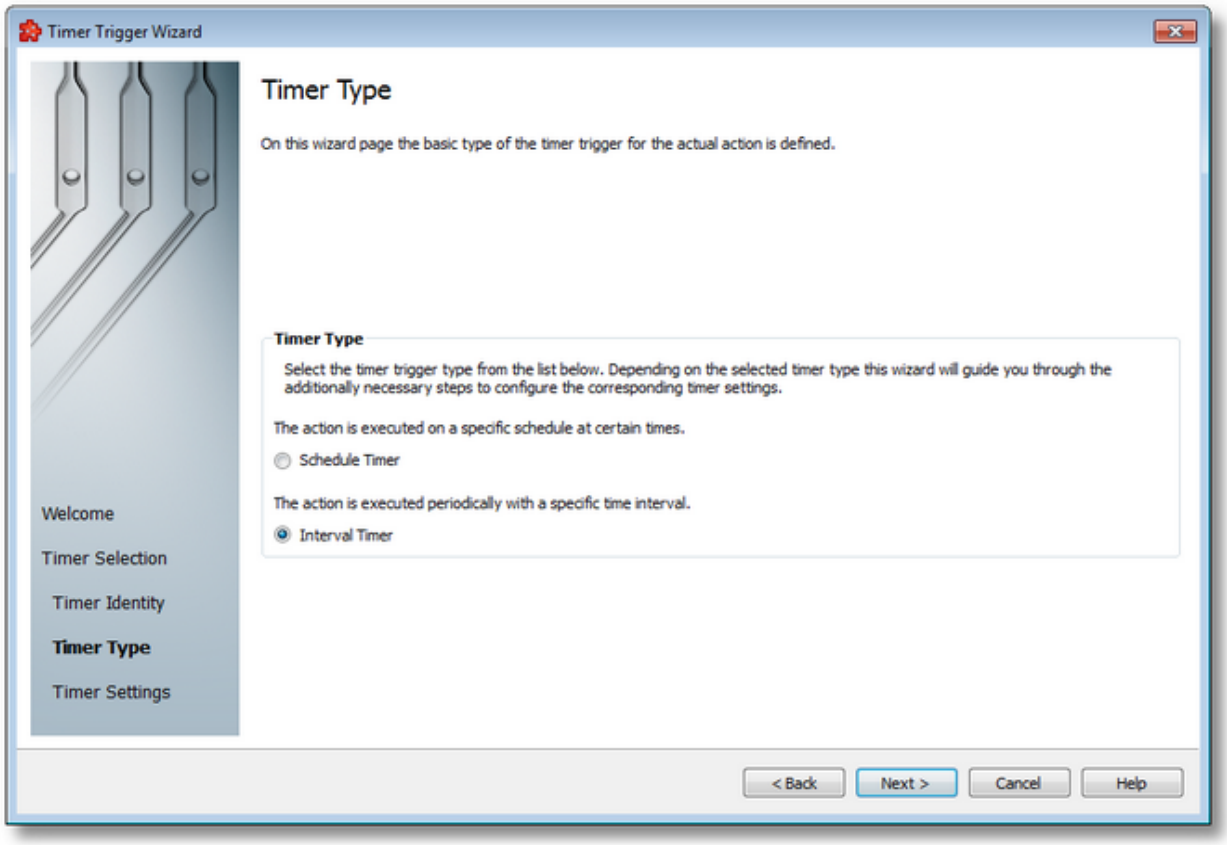
<<< Timer Trigger Wizard - Timer Selection Timer Trigger Wizard - Timer Type >>>

7.3.5.4.4 Timer Type

This wizard page allows choosing the type of the timer trigger. The following types of timer are available:

- *Interval Timer*
- *Schedule Timer*

© Softing Industrial Automation GmbH



Interval Timer

The Interval Timer type allows defining a specific interval of time (in milliseconds) between the timer fire events. The time interval can be configured on the Timer Trigger Wizard - Interval Timer Settings wizard page.

Schedule Timer

The Schedule Timer type allows defining a specific date and time when the timer should fire. The schedule can be configured on the Timer Trigger Wizard - Schedule Timer Settings wizard page.

Note: The *Timer Type* of an existing *Timer Trigger* cannot be changed.

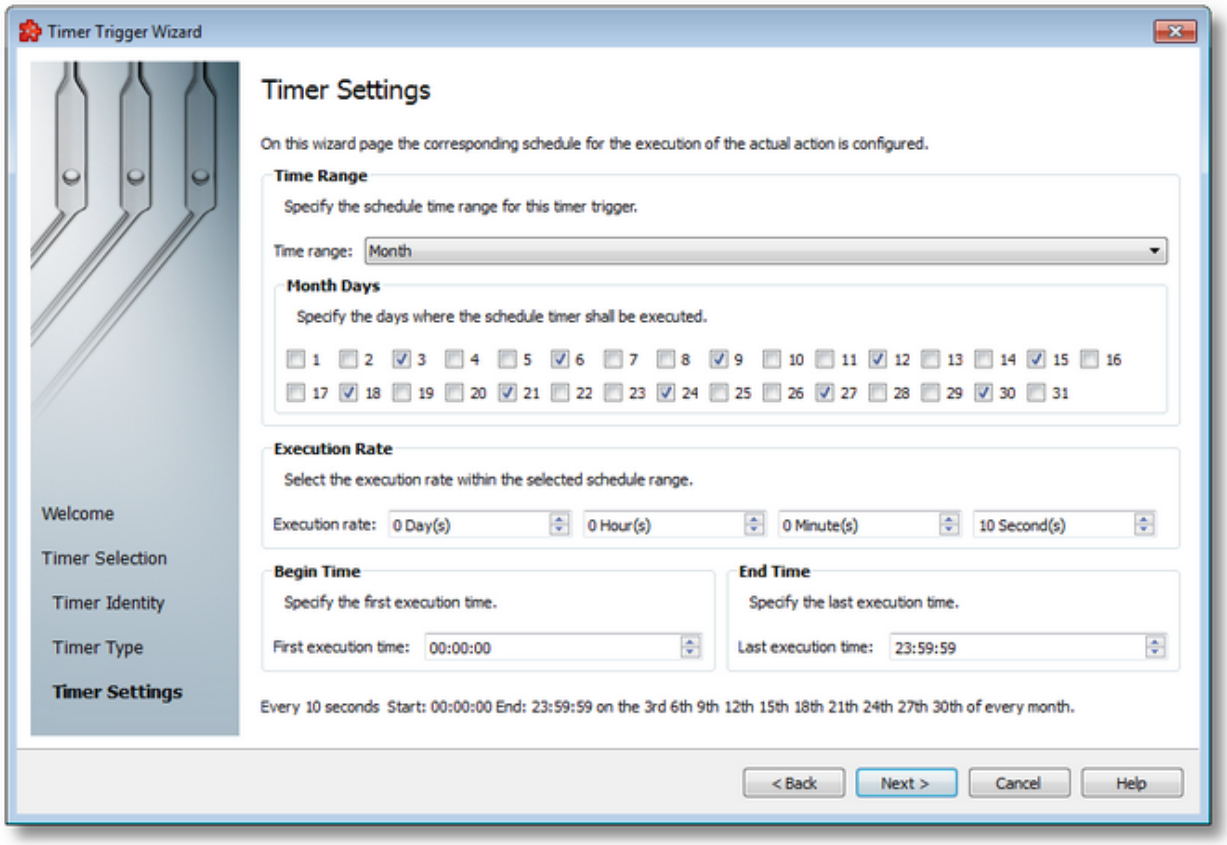
To continue defining the specific timer properties press the *Next* button, to return to the *Timer Identity* wizard page use the *Back* button, or exit this sub-wizard (thereby aborting the *Timer Trigger Wizard*) by pressing the *Cancel* button.

The description of the Timer Trigger Wizard will continue with the wizard pages for configuring a *Schedule Timer*.

<<< Timer Trigger Wizard - Timer Identity Settings >>> Timer Trigger Wizard - Schedule Timer

7.3.5.4.5 Schedule Timer Settings

The Schedule Timer Settings wizard page is a more complex page compared to the Interval Timer Settings wizard page and allows defining the corresponding schedule for the execution of the trigger action:



The page has four sections which identify the schedule:

- *Time Range*: Provides the possibility to choose one of the following schedule ranges: *month*, *week*, *day*, *hour* or *minute*. When the *month* or the *week* ranges are selected the panel below the selection is populated with the month and week days, respectively.
- *Execution Rate*: Sets the execution rate of the current action in the previously-defined *Time Range* (e.g. every X day(s), every Y hour(s), every Z minute(s), every T second(s)).
- *Begin Time*: Sets the time the current action should begin execution.
- *End Time*: Sets the time the current action should stop execution.

According to the *Time Range* the *Begin Time* and the *End Time* fields will allow typing:

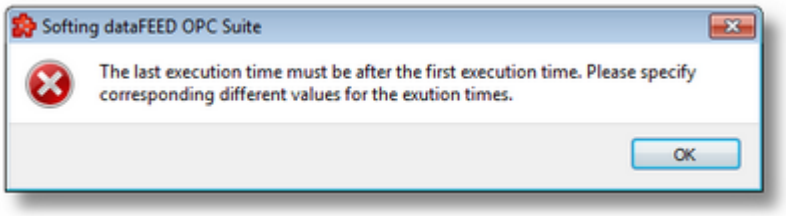
- Seconds (SS) if the *Time range* is *Minute*.
- Minutes and seconds (MM:SS) if the *Time range* is *Hour*.

© Softing Industrial Automation GmbH

- Hours, minutes and seconds (HH:MM:SS) if the *Time range* is *Day* or greater (*Week*, *Month*).

Note: A short review of the configured timer schedule is shown at the bottom of this wizard page.

If the defined schedule is not valid, an error message dialog will appear naming the error that occurred.

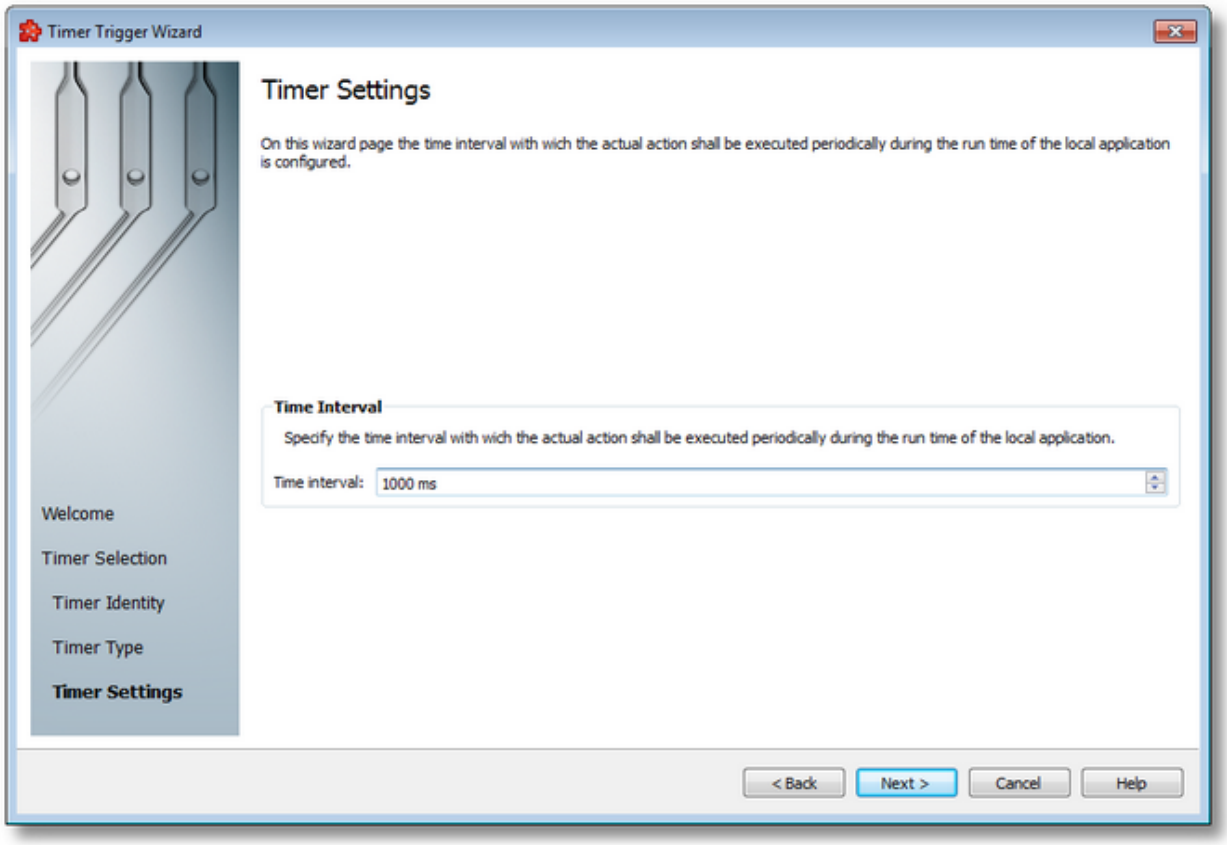


Continue to the *Timer Selection* wizard page using the *Next* button, return to the previous wizard page using the *Back* button or exit the wizard pressing the *Cancel* button.

<<< Timer Trigger Wizard - Timer Type Timer Trigger Wizard - Timer Selection >>>

7.3.5.4.6 Interval Timer Settings

The Interval Timer Settings wizard page allows configuring the time interval at which the current action should be (periodically) executed.



The time interval value is measured in milliseconds and ensures that the action will be triggered at the time interval defined in the spin-box.

Continue to the *Timer Selection* wizard page by pressing the *Next* button, return to the previous one by using the *Back* button, or exit the wizard by pressing the *Cancel* button.

<<< Trigger Wizard - Timer Type Timer Trigger Wizard - Timer Selection >>>

7.3.5.5 Variable Trigger

This section of the Softing dataFEED OPC Suite Online Help describes general settings for the Variable Trigger type for the local dataFEED OPC Suite application.

A variable trigger is a type of trigger, which is fired when a data change for the items used in the variable trigger condition occurs, and conditions defined in the trigger evaluate to true. A data change means that the value or the quality of the item changed.

Conditions can be defined on the Condition Editor and Condition Expression Composer wizard pages, where items can be used the logical expressions to define a condition. Conditions can be linked with logical AND, OR, XOR operators, as well as negated with the NOT operator. Grouping of conditions is also possible for a more clear definition and for better readability. In the Advanced Condition Editor wizard page the user has the possibility to freely refine and better

© Softing Industrial Automation GmbH

customize the syntax of the condition that is currently being edited. A limited set of operators and custom functions are supported, which are listed in the Condition Editor page.

The application has a built-in expression validator, which will prompt the user if there are syntactic errors, meaning that the conditions are not correctly defined. This validator works on every wizard page mentioned above, and also validates the conditions at Runtime, the live values of the items being substituted in that case. Validation is not passed until the condition is not correct syntactically, because otherwise a bad syntax would lead to the action not being executed at Runtime.

Attention: See known limitations in the Condition Expression Composer Wizard - Expression Composer page, Known Limitations section.

The following sections describe how to set up and compose variable triggers:

- Create new variable triggers, select them with the Variable Trigger Wizard @ Variable Trigger Wizard
- Refine the details in a selected variable trigger, use predefined operators and functions in the Condition Expression Composer Wizard @ Condition Expression Composer Wizard

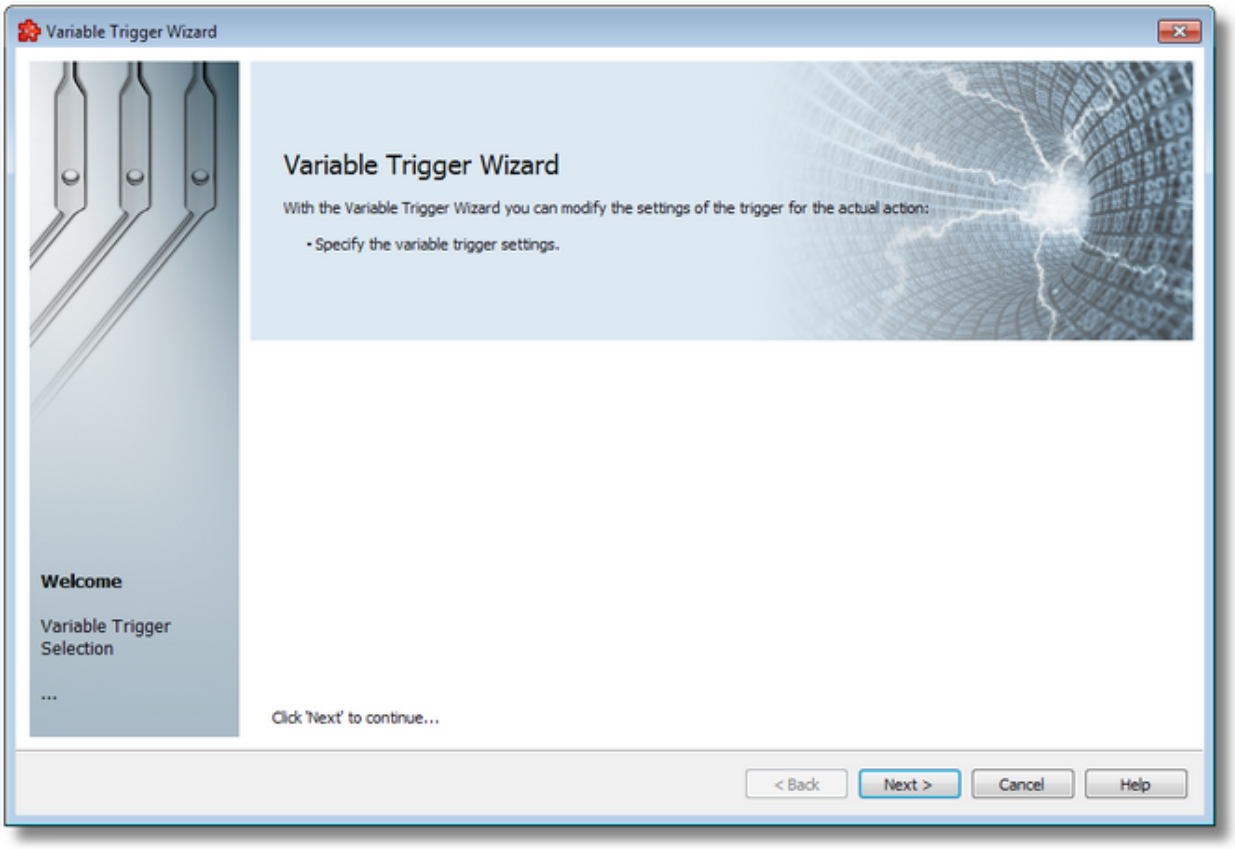
7.3.5.5.1 Variable Trigger Wizard

The Variable Trigger Wizard allows modifying the variable trigger for the current action:

- Obtain general information about the variable trigger @ Variable Trigger Wizard - Welcome
- Select an existing variable trigger @ Variable Trigger Wizard - Trigger Selection
- Name the newly created variable trigger @ Variable Trigger Wizard - Trigger Identity
- Edit the variable trigger condition @ Variable Trigger Wizard - Condition Editor
- Advanced editor for the variable trigger condition @ Variable Trigger Wizard - Advanced Condition Editor

Note: In the following sections example screen shots are given based on the OPC2File add-on in the Data Destination - File main user interface.© **2024 Softing Industrial Automation GmbH**

The *Welcome* wizard page describes the basic steps for completing the variable definition.

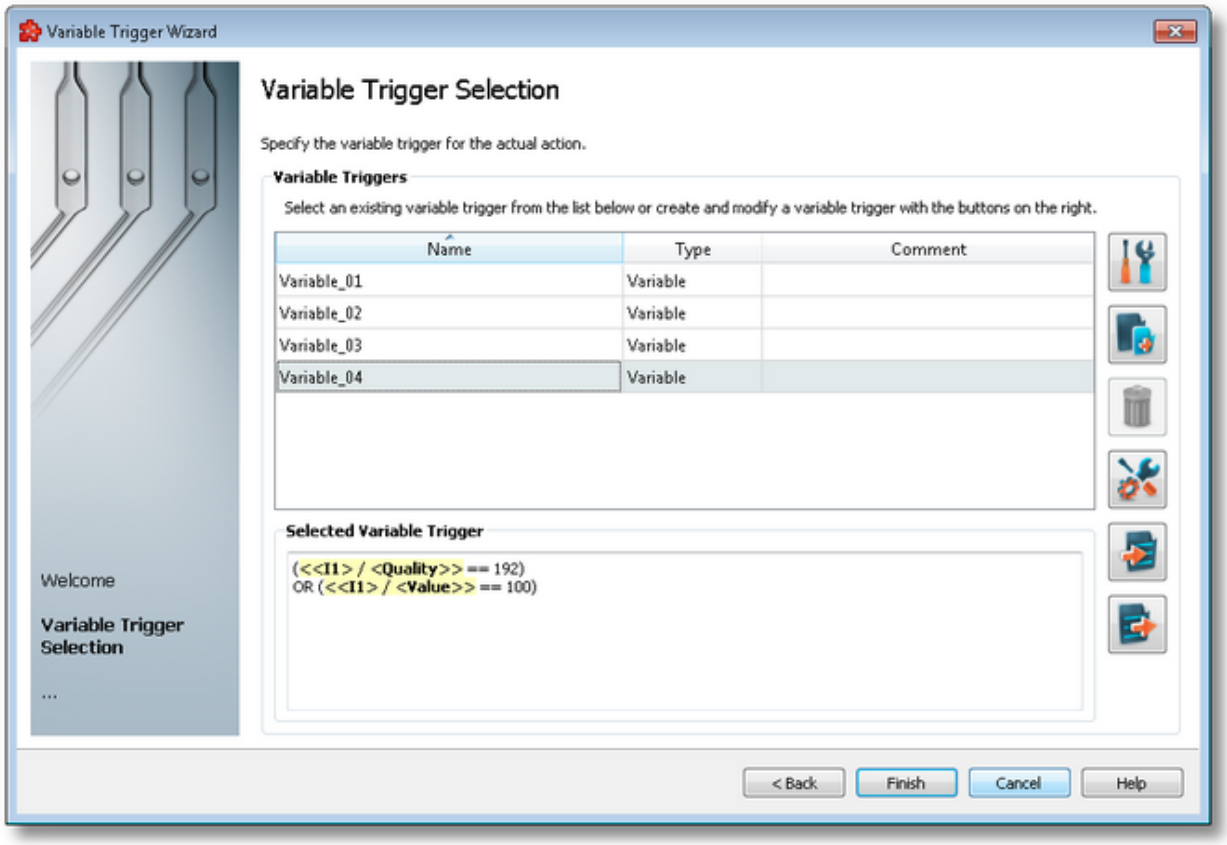


Continue to the next wizard page by using the *Next* button or quit the wizard by pressing the *Cancel* button.

Variable Trigger Wizard - Trigger Selection >>>

The Variable Trigger Selection wizard page is shown only if the selected trigger type is *Variable Trigger*. Here you can configure the actual variable trigger.

© Softing Industrial Automation GmbH

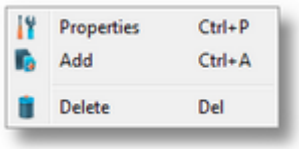


The current page allows selecting a previously defined variable trigger and managing the trigger list by using the buttons on the right, the context sensitive menu or the keyboard shortcuts:

- Change the properties of the selected variable trigger by using the *Properties* button  , pressing the keyboard shortcut CTRL+P or selecting the *Properties* action from the context menu.
- Add a new variable trigger by using the *Add new variable trigger* button  , pressing the keyboard shortcut CTRL+A or selecting the Add action from the context menu.
- Delete the selected variable trigger by using the *Delete selected variable trigger* button  , pressing the DELETE / DEL key or selecting the Delete action from the context menu.
- Modify the timestamp format settings for all condition expressions in the entire configuration using the *Timestamp format settings for all condition expressions* button  or pressing the keyboard shortcut CTRL+D.
- Import a new variable trigger by using the *Import new variable trigger* button  , or selecting the Import action from the context menu.
- Export a selected variable trigger by using the *Export selected variable trigger* button  , or selecting the Export action from the context menu.

Note: If a variable trigger has associated actions it cannot be deleted.

An example of the context sensitive menu can be seen below:



Note: Information about the selected variable trigger is shown in the *Selected Variable Trigger* area at the bottom of the page.

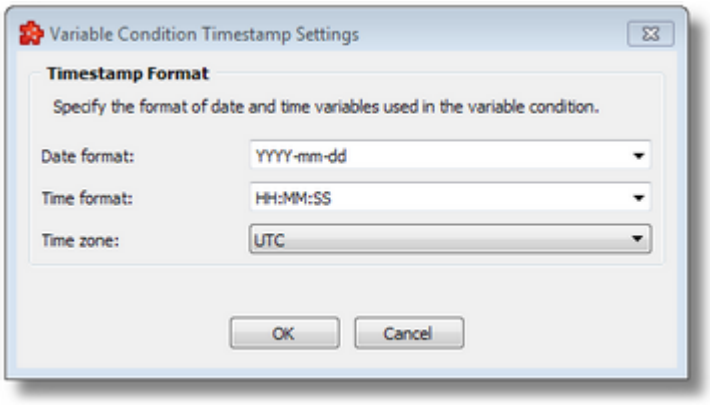
Variable Condition Timestamp Settings

Time format specifies the format in which the timestamps will be converted. Also, *Date format*, but for the date part of the timestamp values.

The default value for the date/time format is "YYYY-mm-dd HH:MM:SS" as seen in the screenshot above.

Note: At least one of the formats need to be specified. If both Time format and Date format values are empty, an error message will be displayed to the user and this dialog will not pass validation.

Time Zone, handles the source localization time: UTC (Universal Time Coordinated) or local time (local time localization). The default time zone is UTC.



These expressions may be used for the date and time format:

Expression	Description
YYYY	the year as four digit number
mm	the month as number with a leading zero (01-12)

© Softing Industrial Automation GmbH

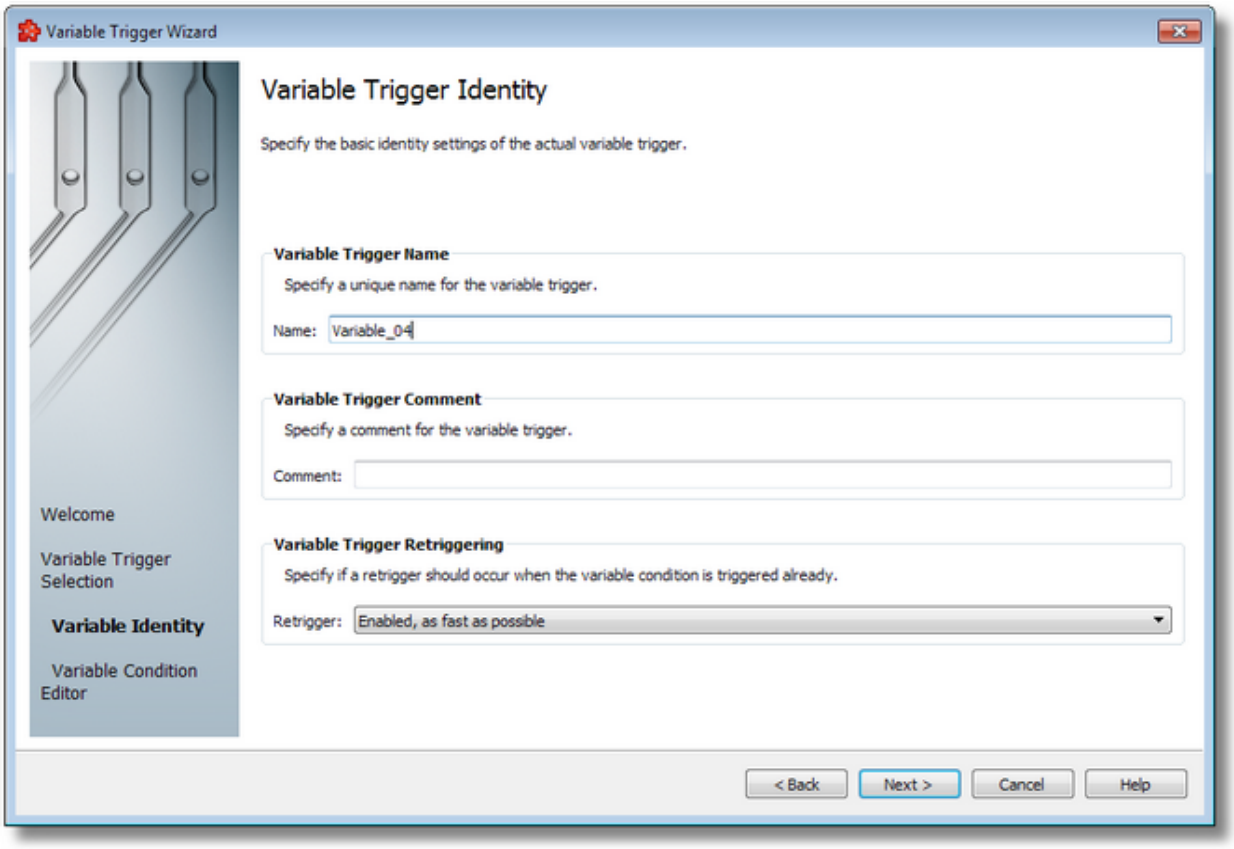
dd	the day as number with a leading zero (01 to 31)
HH	the hour with a leading zero (00 to 23, even with AM/PM display)
MM	the minute with a leading zero (00 to 59)
SS	the whole second with a leading zero where applicable (00 to 59)
FFF	the fractional part of the second, to millisecond precision, including trailing zeroes where applicable (000 to 999)

A sub-wizard that allows adding or modifying variable triggers can be launched by selecting to change an existing variable trigger or to add a new one. The first page of this sub-wizard is Variable Trigger Wizard - Trigger Identity.

Click the *Finish* button to finish the wizard, or use the *Back* button to return to the previous wizard page. To exit the wizard click the *Cancel* button.

<<< Variable Trigger Wizard - Welcome

The Trigger Identity wizard page provides the possibility to assign a name, a comment and a retriggering option to the variable trigger.

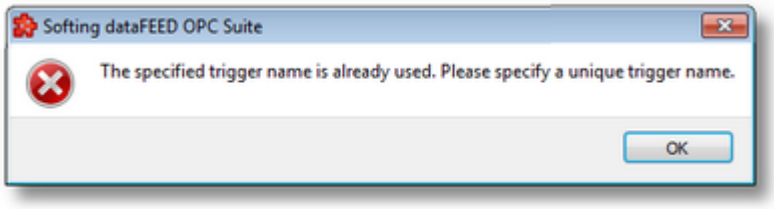


Note: When adding a new variable trigger, the trigger name will be automatically generated according to the format: *Variable_number*, where the *number* part is an auto-incremented integer.

When editing an existing variable trigger, the trigger's defined name and comment will be filled in the two edit boxes allowing you to modify these attributes.

Attention: The name of the *Variable Trigger* must be unique among all triggers and cannot be empty.

If the newly added variable trigger has the same name as an existing variable trigger, an error message box will be displayed and a new trigger name must be provided:



The *Comment* field of the *Variable Identity* wizard page is optional; a Variable Trigger can also be added without any comment.

© Softing Industrial Automation GmbH

Variable Trigger Retriggering

The following retriggering options can be selected:

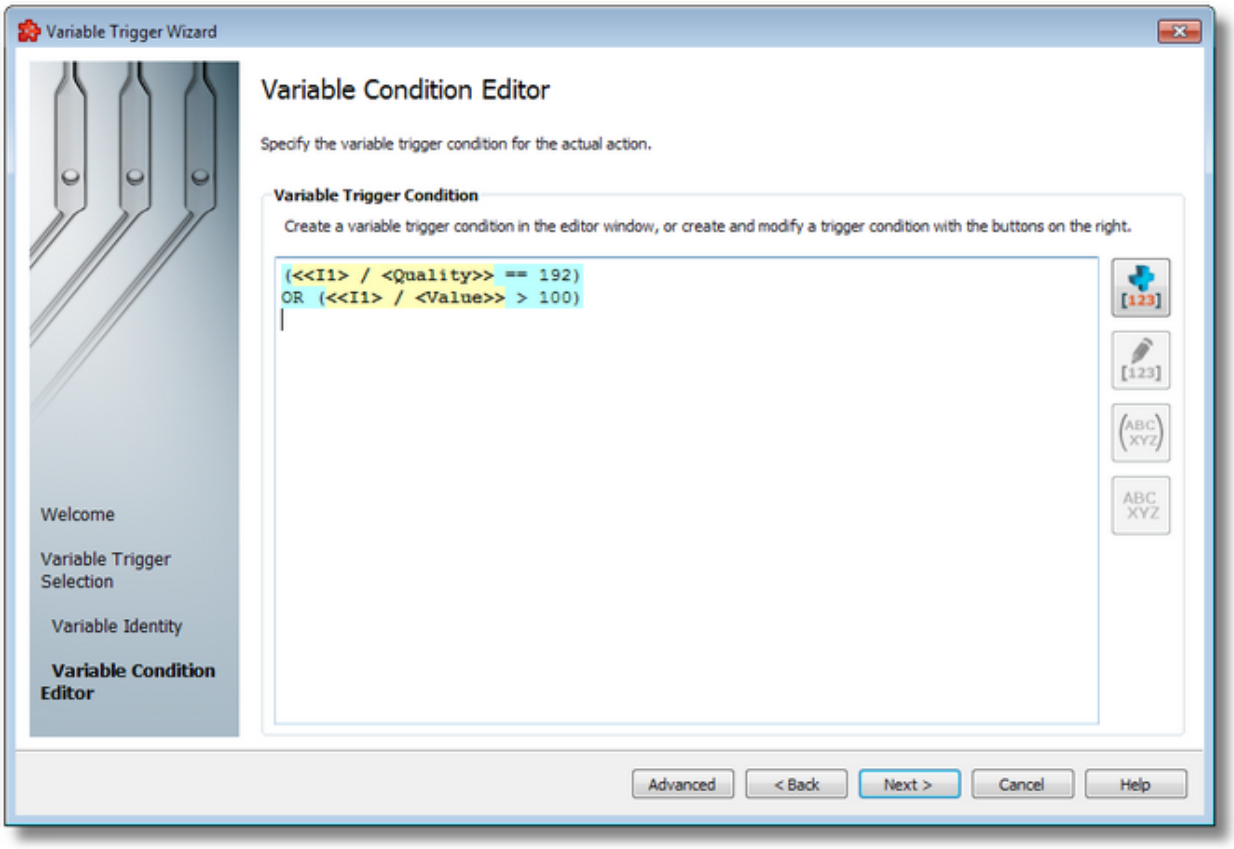
Variable Trigger Retriggering	Description
Disabled	The trigger is fired only at the first time when the variable trigger condition is satisfied. If the variable trigger condition will be evaluated to false for a period of time, the trigger will be fired again, only for one time, when the condition will be satisfied.
Enabled, as fast as possible	The trigger is fired every time when the variable trigger condition is satisfied.
Enabled, at least the specified time interval apart	The trigger is fired when the variable trigger condition is satisfied and at the time interval defined in the *Value* edit-box.

The default *Variable Trigger Retriggering* option is *Enabled, as fast as possible*.

Continue to the next wizard page to specify the trigger condition by using the *Next* button, go back to the previous wizard page to select another variable trigger by pressing the *Back* button, or cancel the entire *Variable Trigger Wizard* by pressing the *Cancel* button.

<<< Variable Trigger Wizard - Variable Trigger Selection Variable Trigger Wizard - Variable Condition Editor >>>

The Condition Editor wizard page allows configuring the condition expression at which the current action should be executed only if the condition expression is true.

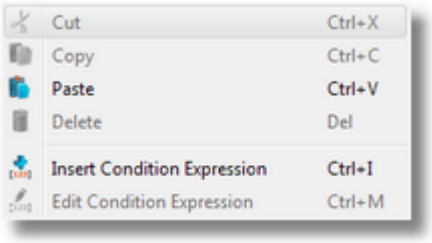


The current page allows the creating or editing of the variable trigger condition by using the buttons on the right, the context sensitive menu or the keyboard shortcuts:

- Insert a condition expression using the *Insert Condition Expression* button  , pressing the keyboard shortcut: CTRL+I or selecting the *Insert Condition Expression* action from the context menu.
- Modify an existing condition expression using the *Modify Condition Expression* button  , pressing the keyboard shortcut: CTRL+M or selecting the *Modify Condition Expression* action from the context menu.
- Group two or more condition expressions using the *Group Selected Condition* button  .
- Ungroup a group of the condition expressions using the *Ungroup Selected Condition* button  .
- Cut, copy, paste or delete conditions. Cut, Copy, Paste or Delete actions from the context menu or using the keyboard shortcuts associated with these actions CTRL+X, CTRL+C, CTRL+V respectively the DELETE / DEL keyboard key.

An example of the context sensitive menu can be seen below:

© Softing Industrial Automation GmbH



Condition Operators

Boolean comparison:

- *booleanItem.value* is *true*;
- *booleanItem.value* is *false*;

Equal, greater, smaller, greater or equal, smaller or equal, not equal:

- *Item.value* == 1 (equal to);
- *Item.value* > 2;
- *Item.value* >= 3;
- *Item.value* < 4;
- *Item.value* <= 5;
- *Item.value* != 6 (not equal to);
- *Item1.value* == *Item2.value*;
- *Item1.value* > *Item2.value*;
- *Item1.value* >= *Item2.value*;
- *Item1.value* < *Item2.value*;
- *Item1.value* <= *Item2.value*;
- *Item1.value* != *Item2.value*;

Note: The following operators are custom functions supported by the expression validator.

Check if the value of a numerical data point has increased/rising edge (with/without hysteresis, from 0/other value):

- *LeadingEdge*(*numericItem.value*, 5, 0) – leading edge from 5, no hysteresis;
- *LeadingEdge*(*numericItem.value*, 2) – leading edge from any value, hysteresis of 2;

Check if the value of a numerical data point has decreased/falling edge (with/without hysteresis, from 0/other value):

- *FallingEdge*(*numericItem.value*, 0, 1) – falling edge from 0, hysteresis of 1;
- *FallingEdge*(*numericItem.value*, 0) – falling edge from any value, no hysteresis;

Check if the value of a numerical data point has changed (with/without hysteresis, from 0/other value):

- *Changed*(*numericItem.value*, "F", 0, "H%", 10)
- *Changed*(*numericItem.value*, "F", 0)
- *Changed*(*numericItem.value*, "F", 10, "H", 5)

Checking if the timestamp or the value of a date/time variable is older than a specific value:

- TimestampOlder("mm/dd/YYYY HH:MM:SS pp", "utc", datetimeItem.value, "d", 30, "h", 1, "m", 0, "s", 0, "x", 0) – if the value of the Item is older than "30 01:00:00", converted to milliseconds;

Note: Values specified as 0 value are not to be taken into consideration when the custom function is evaluated at Runtime. In this case the value of the item needs to be a date time value larger than 30 days and 1 hours expressed in milliseconds.

Negating a condition (not):

- *Not* (*FallingEdge*(numericItem.value, 0)) – if the value is not decreasing (some packing process got jammed);
- *Not* (*Item1.value* == "Normal") – if the value of the *Item1* is not equal to constant string "Normal"; in this case using a != operator yields better performance, the example is given just to show the syntax;

Create complex conditions between two or more data points: Boolean logic – AND/OR/XOR:

```
(booleanItem.value == true)
OR (numericItem.value > 10)
```

Attention: The conditions need to be written each on a separate line in the *Variable Condition Editor*.

Grouping of conditions should be easily done by selecting more conditions and clicking the *Group Selected Condition* button  .
Ungrouping a expression by selecting that expression and by pressing the *Ungroup Selected Condition* button  .

```
((booleanItem.value == true)
  AND (numericItem.value > 10))
OR ((booleanItem.value == true)
  AND (stringItem.value == "Normal"))
```


© Softing Industrial Automation GmbH

Attention: Editing a condition inside a group will not be possible. Ungrouping will be required.

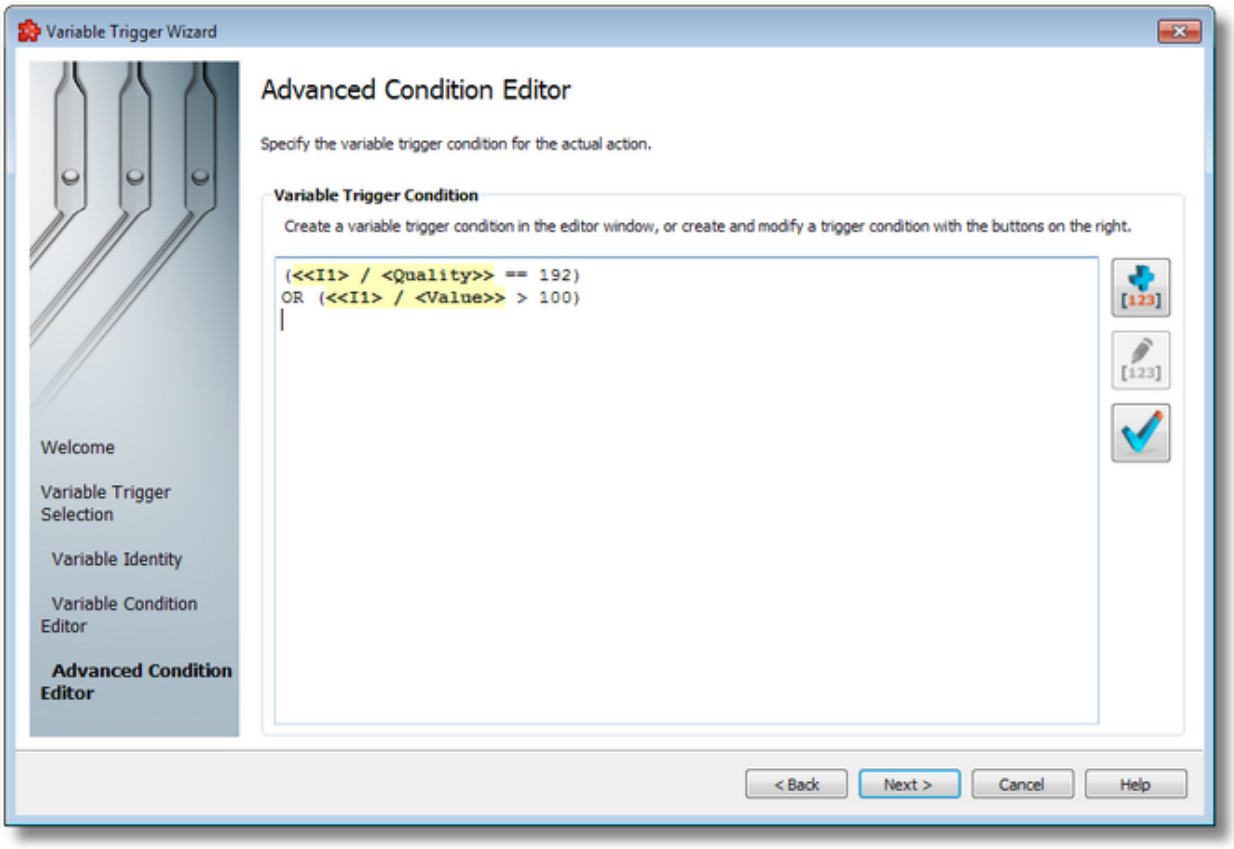
Attention: Please keep in mind that conditions copied to clipboard (through the *Copy* or *Cut* commands) will only be pasted back as conditions if the clipboard has not been tampered with, otherwise the conditions will be pasted as plain text.

Attention: The placeholders visible in the action editor are aliases used only for displaying the placeholders. The real placeholder form is visible in the subscription file. Example: <<*R8*> / <*Value*>> placeholder from the action editor is serialized in the subscription file as: *#%-ItemValue-%increment.R8_%Current-%#*. Do not manually modify the subscription file by inserting placeholders in the action editor template.

Continue to the *Trigger Selection* wizard page by pressing the *Next* button, edit using the Variable Trigger Wizard - Variable Advanced Condition Editor page by pressing *Advanced*, return to the previous one by using the *Back* button, or exit the wizard by pressing the *Cancel* button.

<<<Variable Trigger Wizard - Trigger Identity Variable Trigger Wizard - Variable Trigger Selection >>>

The Condition Editor wizard page allows configuring the condition expression at which the current action should be executed only if the condition expression is true.

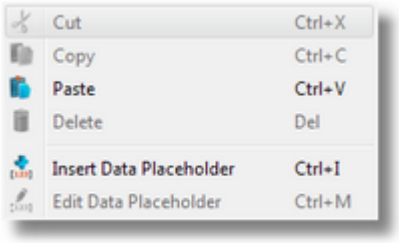


The current page allows the creating or editing of the variable trigger condition by using the buttons on the right, the context sensitive menu or the keyboard shortcuts:

- Insert a data value placeholder using the *Insert Data Value Placeholder* button  , pressing the keyboard shortcut: CTRL+I or selecting the *Insert Data Value Placeholder* action from the context menu.
- Modify an existing data value placeholder using the *Modify Data Value Placeholder* button  , pressing the keyboard shortcut: CTRL+M or selecting the *Edit Data Value Placeholder* action from the context menu.
- Validate the condition using the *Validate Condition* button  , that allow to validate the condition according to the syntax specifications.
- Cut, copy, paste or delete text with or without placeholders selecting the Cut, Copy, Paste or Delete actions from the context menu or using the keyboard shortcuts associated with these actions CTRL+X, CTRL+C, CTRL+V respectively the DELETE / DEL keyboard key.

An example of the context sensitive menu can be seen below:

© Softing Industrial Automation GmbH



For more information about condition syntax and operators check Variable Trigger Wizard - Variable Condition Editor topic.

Note: The *Cut*, *Copy* and *Paste* actions use the Windows Clipboard for data transfer thus allowing data to be transferred to/from other applications (e.g. the textual representation of the action editor content can be used in external applications and text from external application can be used in the action editor).

Attention: Please keep in mind that placeholders copied to clipboard (through the *Copy* or *Cut* commands) will only be pasted back as placeholders if the clipboard has not been tampered with, otherwise the placeholders will be pasted as plain text.

Attention: The placeholders visible in the action editor are aliases used only for displaying the placeholders. The real placeholder form is visible in the subscription file. Example: *<<R8> / <Value>>* placeholder from the action editor is serialized in the subscription file as: *#%-ItemValue-%increment.R8_%Current-%#*. Do not manually modify the subscription file by inserting placeholders in the action editor template.

Continue to the *Trigger Selection* wizard page by pressing the *Next* button, return to the previous one by using the *Back* button, or exit the wizard by pressing the *Cancel* button.

<<< Variable Trigger Wizard - Variable Condition Editor Variable Trigger Wizard - Variable Trigger Selection >>>

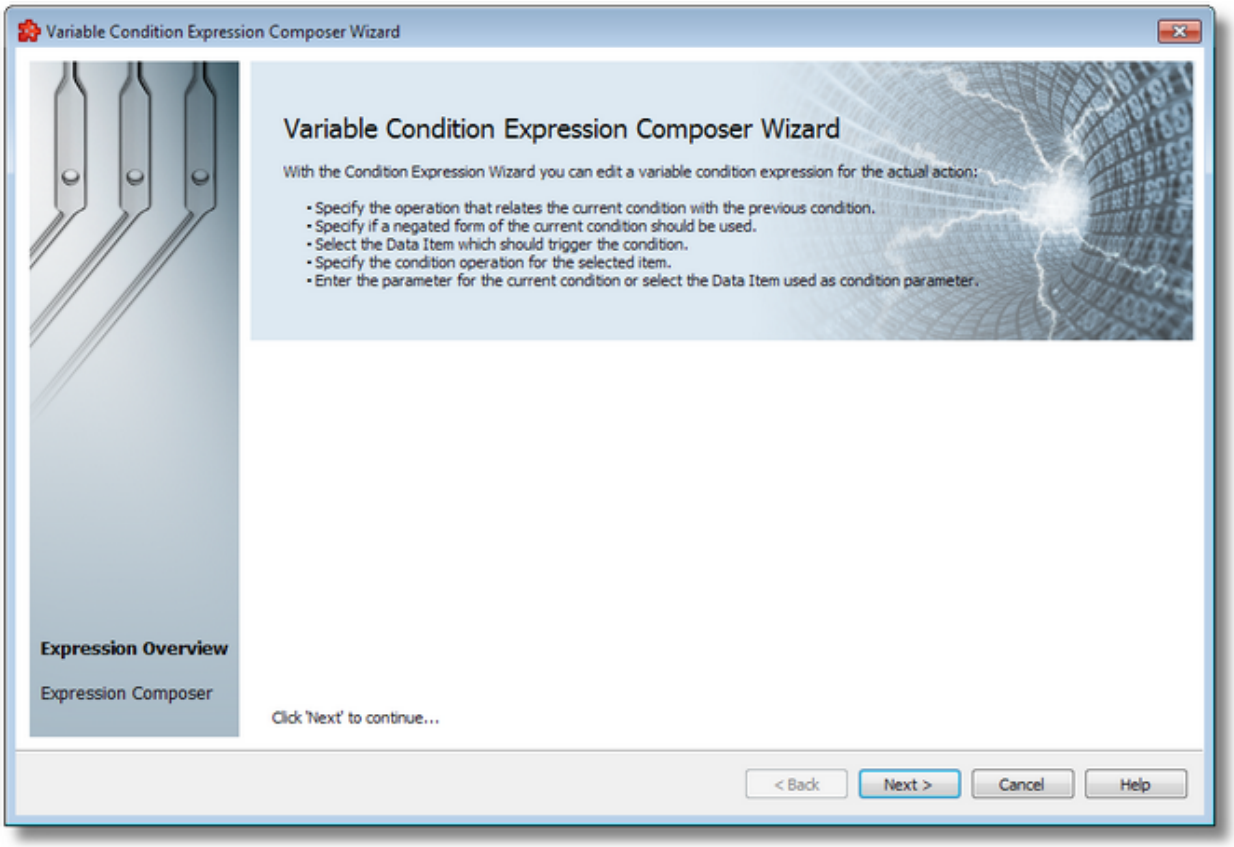
7.3.5.5.2 Condition Expression Composer Wizard

▼ The Condition Expression Composer Wizard allows modifying the condition expressions for the current variable trigger:

- Obtain general information about the condition expression composer wizard @ Condition Expression Composer Wizard - Expression Overview

- Edit a variable condition expression for the actual action @ Condition Expression Composer Wizard - Expression Composer
- Specify the parameters for the *Timestamp Older* condition operator @ Condition Expression Composer Wizard - Timestamp Older Settings
- Specify the parameters for the *Leading Edge* condition operator @ Condition Expression Composer Wizard - Leading Edge Settings
- Specify the parameters for the *Falling Edge* condition operator @ Condition Expression Composer Wizard - Falling Edge Settings
- Specify the parameters for the *Changed* condition operator @ Condition Expression Composer Wizard - Changed Settings

The *Welcome* wizard page describes the basic steps for completing the variable condition expression.

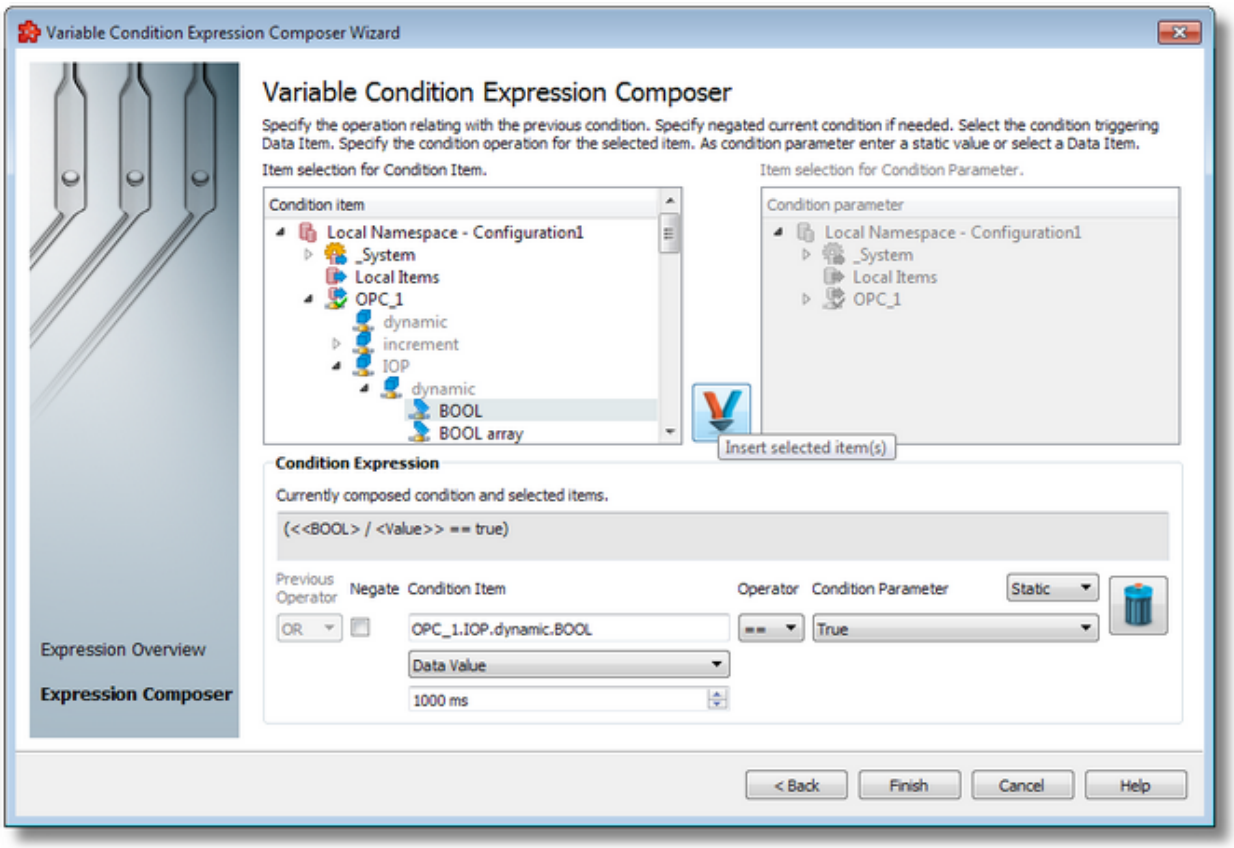


Continue to the next wizard page by using the *Next* button or quit the wizard by pressing the *Cancel* button.

Condition Expression Composer Wizard - Expression Composer >>>

© Softing Industrial Automation GmbH

The Expression Composer wizard page provides the possibility to specify the condition items and operator, the operator relating with the previous condition and negated current condition if needed.



Condition Composing Elements

Previous Operator - allows selecting between operators *OR* / *AND* / *XOR,* which will link different conditions.

Negate Condition - allows the negation of the condition expression.

Condition Item - allows selecting the first parameter of the condition and specifying the *Data Item*, *Data Value* and *Update Rate* properties.

Condition Operator - allows the selecting between operations:

- equal, greater, smaller, greater or equal, smaller or equal, not equal;
- *TO* (Timestamp Older), only for *DATE* data type condition items;
- *LE* (Leading Edge), *FE* (Falling Edge) and *CHD* (Changed) only for numeric and DATE data type condition items; CHD (Changed) is also available for string data type condition items

Condition Parameter - allows the selecting the second condition item between types *Static Value* or other *Data Items.*

Delete the selected condition expression by using the *Delete the selected expression* button  or pressing the DELETE / DEL key.

For more information about condition syntax and operators check Variable Trigger Wizard - Condition Editor topic.

Composing Conditions

This wizard page makes it possible to compose conditions very easily, by dragging and dropping items from the left and right address space views, selecting operators from a pre-populated list, and by correctly being able to specify static value for the right part of conditions.

The first element to be specified is the *Previous Operator*, which links the edited condition to any existing conditions in the current variable trigger. They can be seen in the Condition Editor wizard page, after exiting this composer. By default this element is OR, but AND or XOR (eXclusive OR) can be selected.

Second is the negation operator, which will result in the opposite of the current condition. By default it is not applied. Check the *Negate Condition* checkbox to enable it.

The *Condition Item* tree view allows to specify the left part of the condition, which must always be an item. The item can be added by drag and drop, or by manually writing out its whole hierarchical name. Any of its VQT elements can be used in the condition, and an update rate can also be specified. Array elements can also be used here, but only indexed elements, not the whole item, whose data type is array. Ex.: Bool[0], I4[3]. Indexing is possible on the user interface, when such an item is dragged.

Next, the condition operator needs to be selected, depending on the *Condition Item's* data type. Some operators will not be available for all data types.

According to the selected condition item, the input field in *Condition Parameter* is going to change to allow and validate the entry of values of that specific data type. For example, if the data type of the condition item is float/double, the entry field will change so that decimal values can be entered. If it is boolean, then it will change into a dropdown box and will allow on True or False values to be selected. The default here is *Static Value*, but if the user switches to *Item Value*, then the condition parameter can be an item from the address space. The right tree view is going to become enabled and items can be selected, or drag and dropped, similarly to the condition item.

When a match is valid between the selected condition item and condition parameter item, the button in the middle can also be pressed to populate the lower section with the item information, the condition thus becoming valid.

After pressing the *Finish* button, the validated condition will be added to the Condition Editor on a new line, separated by a new line from the existing conditions. If this is the first condition added, the *Previous Operator* is going to be omitted.

© Softing Industrial Automation GmbH

Known Limitations

Attention: When using very large or very small UI4, I8, UI8, R4 or R8 values in conditions, which are close or equal to the maximums or minimums of these data types, precision losses may occur.

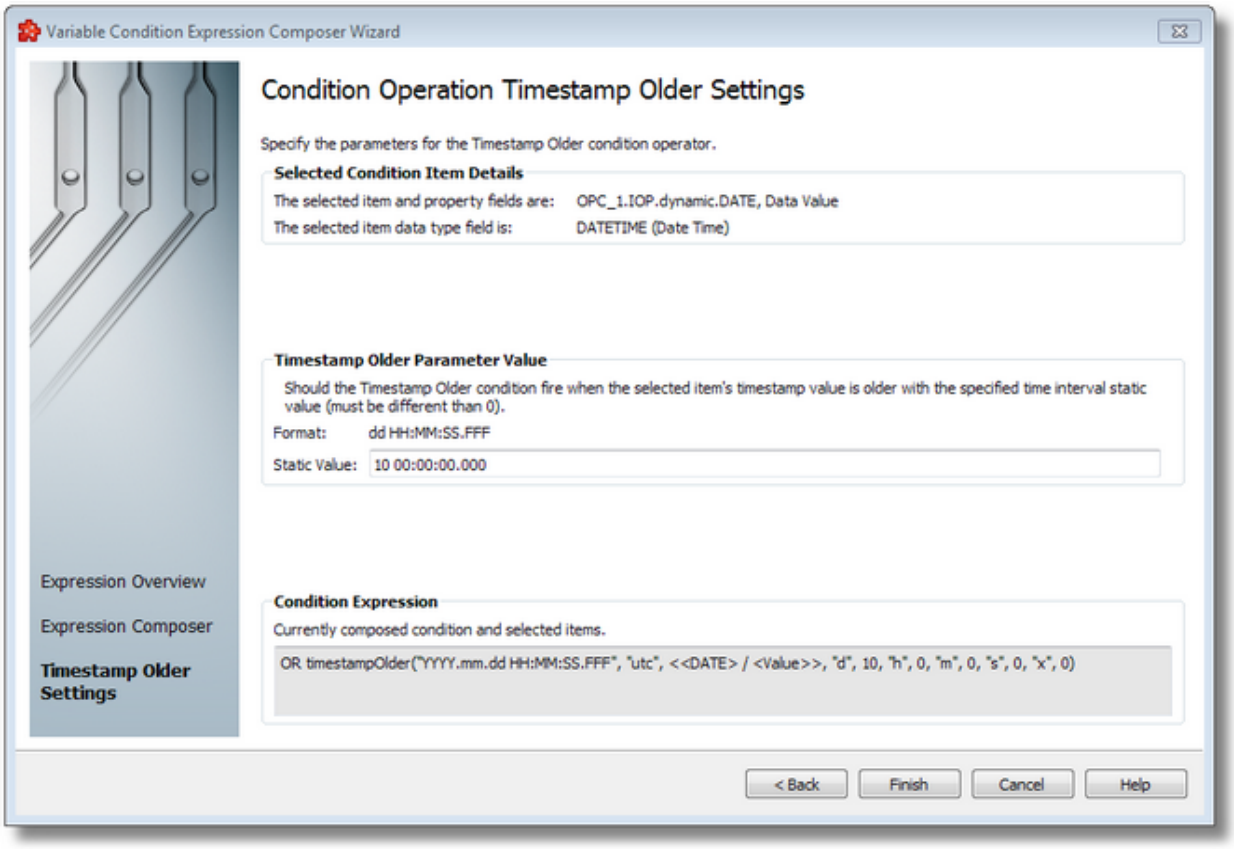
Due to limitations in the representation of floating-point values in the condition evaluation library, values that are close to the extremes of these types are going to lose precision.

When comparing large integer values of R4, R8, I8 and UI8 data types, especially with *equality (==)* operator, conditions with such values may not evaluate correctly.

Such comparisons are done with an epsilon of 1e-15, by calculating the difference between the two values and comparing to this value. If the difference is smaller, then the two values are considered equal.

<<< Condition Expression Composer Wizard - Expression Overview

The Timestamp Older Settings wizard page allows configuring the Timestamp Older operator.



Selected Condition Item Details - show details about selected condition item.

Timestamp Older Parameter Value - allows to specify the static value that is used at timestamp comparison. This is a value that needs to be entered similarly to a date with all components, but it is converted to milliseconds.

Not all elements need to be specified, those that are not desired have 0 as default value, which does not change the final value expressed in milliseconds.

The maximum time value that can be entered is 49 days, 17 hours, 2 minutes, 47 seconds and 295 milliseconds (or 4294967295 milliseconds).

Note: The used timestamp format is the general format for variable trigger conditions and can be modified in the Variable Condition Timestamp Settings dialog.

At least one component needs to be specified, otherwise the wizard will prompt the user to provide a value.

Parameters

- timestamp format: this is the format in which the function will interpret the date-time value of the item at parameter 3
- time zone: either "utc" or "local" as text

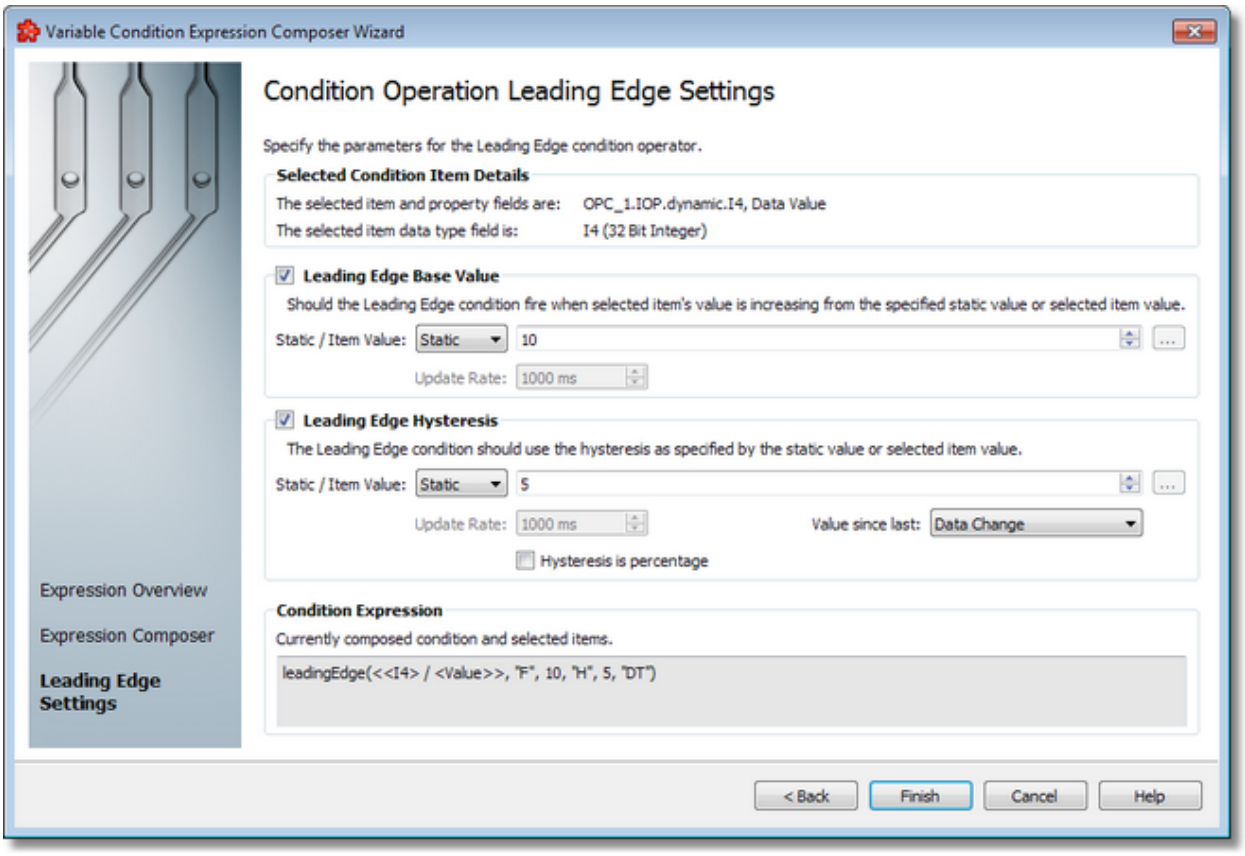
© Softing Industrial Automation GmbH

- the item, whose value must be of date type date-time
- 5 pairs of parameters, which specify the different components of the time value with which the item value will be compared

Continue to the *Condition Editor* wizard page by pressing the *Finish* button, return to the previous one by using the *Back* button, or exit the wizard by pressing the *Cancel* button.

<<< Condition Expression Composer Wizard - Expression Composer

The Leading Edge Settings wizard page allows configuring the Leading Edge operator.



Selected Condition Item Details - show details about selected condition item.

Leading Edge Base Value - allows to specify a static value or an item value to actual condition

Leading Edge Hysteresis - allows to specify a static value or an item value to actual condition; for DATE data type, hysteresis can be only static timestamp value and has the same constraints like Timestamp older settings.

Both *Base Value* and *Hysteresis* are optional parameters.

Note: *Leading Edge* is the opposite of *Falling Edge.*

This custom function evaluates to true when the value of the item specified as first parameter is larger than its previous value.

Moreover, the following additional conditioning may be used:

- Base Value: the condition only evaluates to true if the item's previous value exists, and that value equals the base value
- Base Value/Value Since Last: if the *Global*(...) value is selected, then the global default is used specified in parenthesis. Otherwise, the selected value will override the global default for this expression only (see explanation on the Global Initial and Alias Values page)
- Hysteresis: the condition evaluates to true, when the amount the item's value increases with is larger than the previous value, with at least the amount of hysteresis expressed as number or as a percentage

Examples:

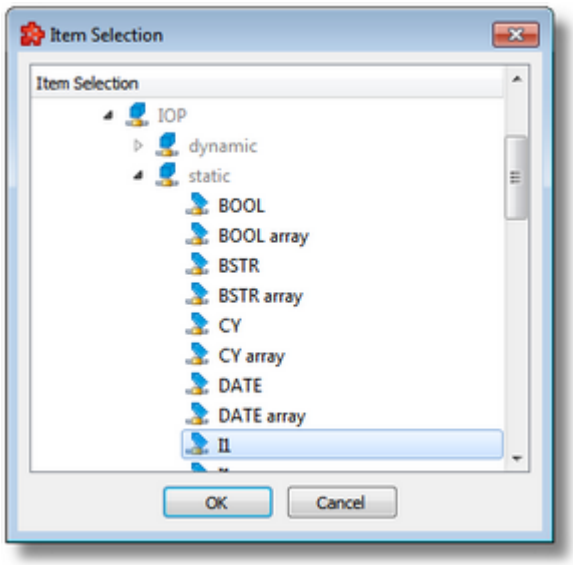
- leadingEdge(item) - item current value is greater than item previous value;
- leadingEdge(item, "f", itemFrom) - current item value is greater than previous item value and previous item value was itemFrom value; *Global* (...) parameter is applied with the selected global default
- leadingEdge(item, "f", itemFrom, "DT") - current item value is greater than previous item value and previous item value was itemFrom value; *Data Changed* parameter is applied
- leadingEdge(item, "F", 123.456) - current item value is greater than previous item value and previous item value was 123.456; *Global* (...) parameter is applied with the selected global default
- leadingEdge(item, "F", 123.456, "TR") - current item value is greater than previous item value and previous item value was 123.456; *Fired Trigger* parameter is applied
- leadingEdge(item, "h", itemHysteresis) - current item value is greater than previous item value with at least itemHysteresis value;
- leadingEdge(item, "H%", 12.34) - current item value is greater than previous item value with at least 12.34 percent of previous item value;
- leadingEdge(item, "F", itemFrom, "H%", 5.4321098) - current item value is greater than the previous item value with at least 5.4321098 percent of previous item value and previous item value was the itemFrom value; *Global* (...) parameter is applied with the selected global default
- leadingEdge("D", "YYYY/mm/dd HH:MM:SS", "utc", datetime_item): current date time item is newer than it's previous value

© Softing Industrial Automation GmbH

- leadingEdge("D", "YYYY/mm/dd HH:MM:SS", "utc", datetime_item, "F", itemFrom) current date time item is newer than itemFrom
- leadingEdge("D", "YYYY/mm/dd HH:MM:SS", "utc", datetime_item, "F", "2024/02/21 13:24:59") current date time item is newer than 21st February 2024, 13:24:59
- leadingEdge("D", "YYYY/mm/dd HH:MM:SS", "utc", datetime_item, "H", "d", 1, "h", 0, "m", 2, "s", 1, "x", 0) current item is newer than it's previous value by 1 day, 2 minutes and 1 second

Note: If the *Leading Edge Base Value* or *Hysteresis* are item values then the *Update Rate* parameter can be configured for every item.

An example of the item value selection can be seen below:

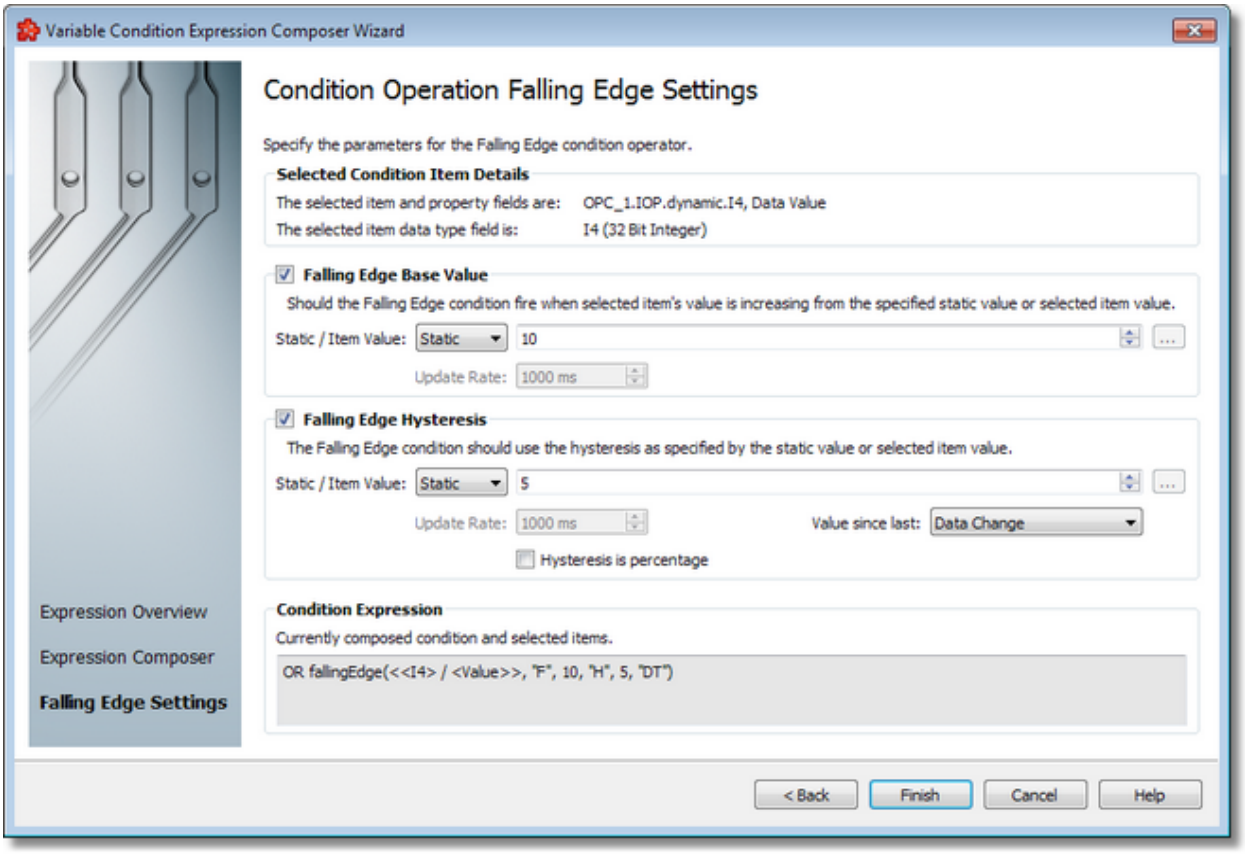


Note: Information about the actual condition is shown in the *Condition Expression* area at the bottom of this wizard page.

Continue to the *Condition Editor* wizard page by pressing the *Finish* button, return to the previous one by using the *Back* button, or exit the wizard by pressing the *Cancel* button.

<<< Condition Expression Composer Wizard - Expression Composer

The Falling Edge Settings wizard page allows configuring the Falling Edge operator.



Selected Condition Item Details - show details about selected condition item.

Falling Edge Base Value - allows to specify a static value or an item value to actual condition

Falling Edge Hysteresis - allows to specify a static value or an item value to actual condition; for DATE data type, hysteresis can be only static timestamp value and has the same constraints like Timestamp older settings.

Both *Base Value* and *Hysteresis* are optional parameters.

Note: *Falling Edge* is the opposite of *Leading Edge.*

This custom function evaluates to true when the value of the item specified as first parameter is lower than its previous value.

Moreover, the following additional conditioning may be used:

- Base Value: the condition only evaluates to true if the item's previous value exists, and that value equals the base value
- Base Value/Value Since Last: if the *Global*(...) value is selected, then the global default is used specified in parenthesis. Otherwise, the selected value will override the global default for this expression only (see explanation on the Global Initial and Alias Values page)

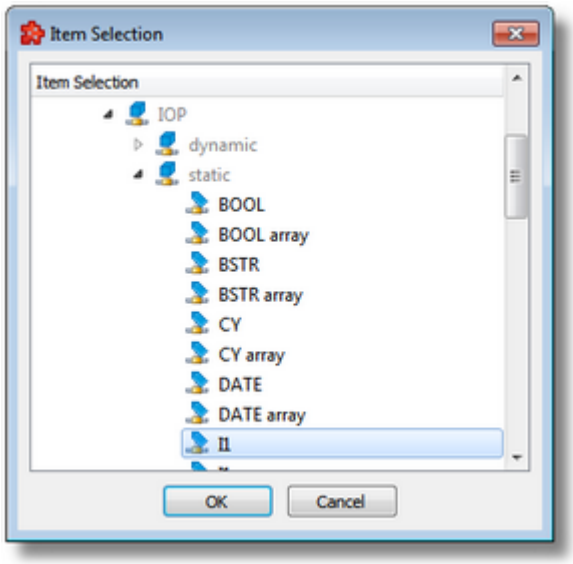
© Softing Industrial Automation GmbH

- Hysteresis: the condition evaluates to true, when the amount the item's value decreases with is lower than the previous value, with at least the amount of hysteresis expressed as number, a timestamp or as a percentage

The same examples apply here as well as for *Leading Edge*, but the condition is that the current value of the item is lower than the previous.

Note: If the *Falling Edge Base Value* or *Hysteresis* are item values then the *Update Rate* parameter can be configured for every item.

An example of the item value selection can be seen below:

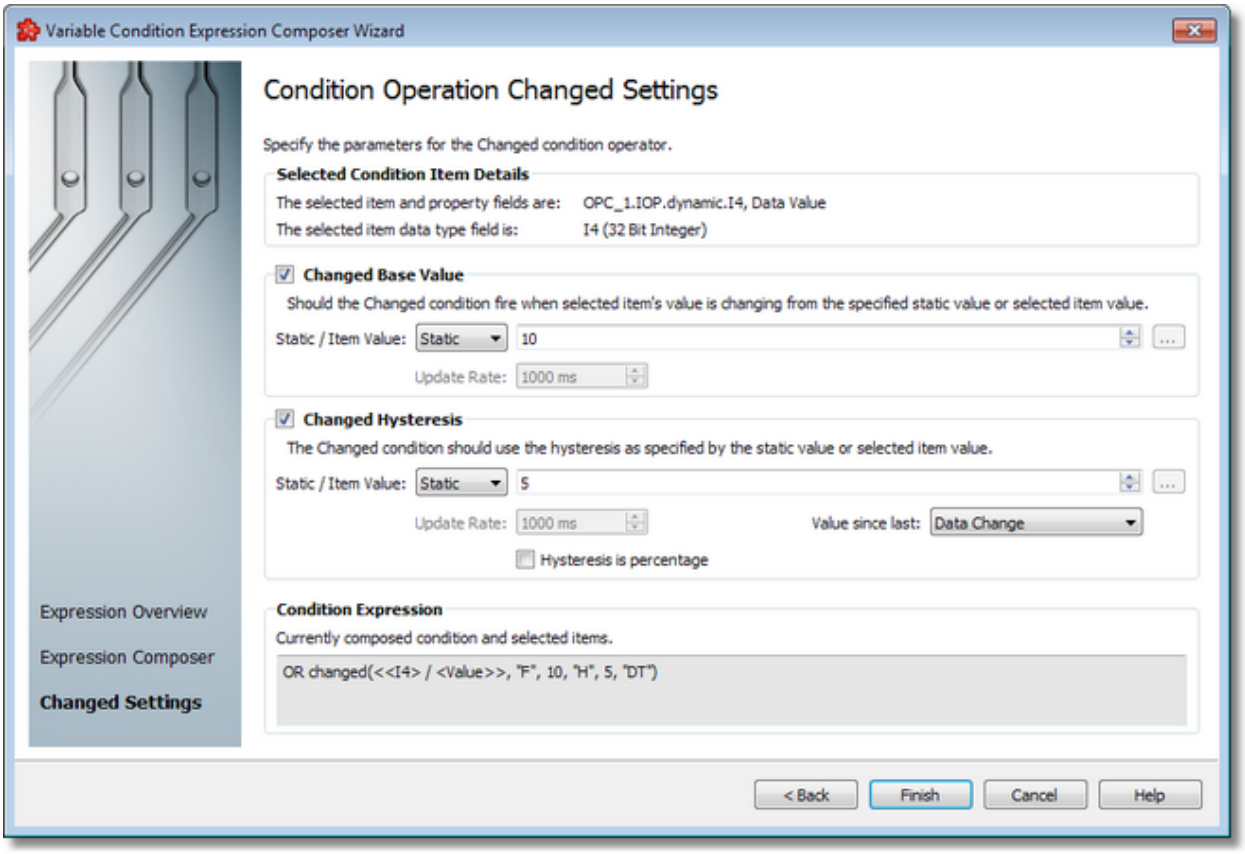


Note: Information about the actual condition is shown in the *Condition Expression* area at the bottom of this wizard page.

Continue to the *Condition Editor* wizard page by pressing the *Finish* button, return to the previous one by using the *Back* button, or exit the wizard by pressing the *Cancel* button.

<<< Condition Expression Composer Wizard - Expression Composer

The Changed Settings wizard page allows configuring the Changed operator.



Selected Condition Item Details - show details about selected condition item.

Changed Base Value - allows to specify a static value or an item value to actual condition

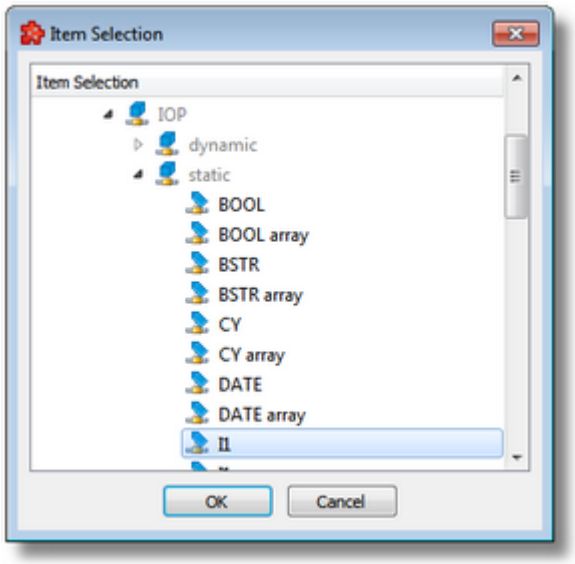
Changed Hysteresis - allows to specify a static value or an item value to actual condition; for DATE data type, hysteresis can be only static timestamp value and has the same constraints like Timestamp older settings.

Note: If the *Changed Base Value* or *Hysteresis* are item values then the *Update Rate* parameter can be configured for every item.

This custom function evaluates to true when *Leading Edge* OR *Falling Edge* is true for the same values.

An example of the item value selection can be seen below:

© Softing Industrial Automation GmbH



Note: Information about the actual condition is shown in the *Condition Expression* area at the bottom of this wizard page.

Continue to the *Condition Editor* wizard page by pressing the *Finish* button, return to the previous one by using the *Back* button, or exit the wizard by pressing the *Cancel* button.

<<< Condition Expression Composer Wizard - Expression Composer

This page is intentionally left blank.

Copyright 2024 Softing Industrial Automation GmbH

Softing Industrial Automation GmbH

Richard-Reitzner-Allee 6
85540 Haar / Germany
http://industrial.softing.com

Tel: + 49 89 45 656-0
Fax: + 49 89 45 656-488
info.automation@softing.com